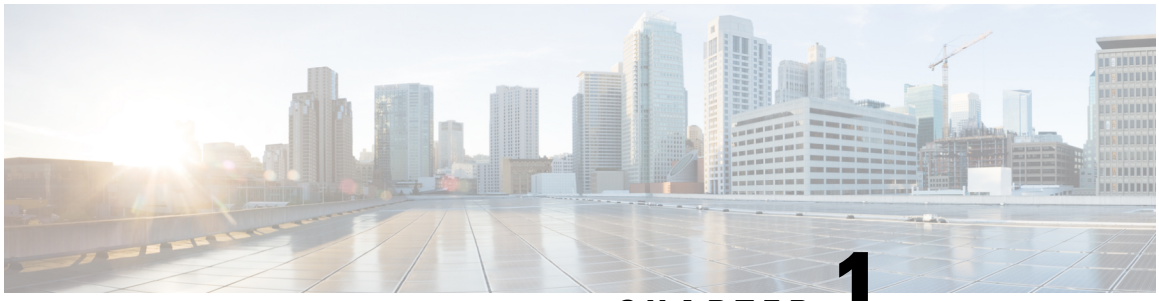
Cisco Adaptive Security Virtual Appliance (ASAv) Getting Started Guide, 9.17

First Published: 2021-06-21

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Introduction to the ASAv

The Adaptive Security Virtual Appliance (ASAv) brings full firewall functionality to virtualized environments to secure data center traffic and multitenant environments.

You can manage and monitor the ASAv using ASDM or CLI. Other management options may be available.

Hypervisor Support

For hypervisor support, see Cisco ASA Compatibility.

Licensing for the ASAv

The ASAv uses Cisco Smart Software Licensing. For complete information, see Smart Software Licensing.

Note You must install a smart license on the ASAv. Until you install a license, throughput is limited to 100 Kbps so you can perform preliminary connectivity tests. A smart license is required for regular operation.

Beginning with 9.13(1), any ASAv license can be used on any supported ASAv vCPU/memory configuration. This allows you to deploy an ASAv on a wide variety of VM resource footprints. Session limits for AnyConnect Client and TLS Proxy are determined by the ASAv platform entitlement installed rather than a platform limit tied to a model type.

See the following sections for information about ASAv licensing entitlements and resource specifications for the supported private and public deployment targets.

About Smart License Entitlements

Any ASAv license can be used on any supported ASAv vCPU/memory configuration. This allows you to run the ASAv on a wide variety of VM resource footprints. This also increases the number of supported AWS and Azure instances types. When configuring the ASAv machine, the maximum supported number of vCPUs is 16 (ASAv100); and the maximum supported memory is 64GB for ASA virtual deployed on all platforms other than AWS and OCI. For ASA virtual deployed on AWS and OCI, the maximum supported memory is 128GB.

Important It is not possible to change the resource allocation (memory, CPUs, disk space) of an ASAv instance once it is deployed. If you need to increase your resource allocations for any reason, for example to change your licensed entitlement from the ASAv30/2Gbps to the ASAv50/10Gbps, you need to create a new instance with the necessary resources.

- vCPUs—The ASAv supports 1 to 16 vCPUs.
- Memory—The ASAv supports 2GB to 64GB of RAM for ASA virtual deployed on all platforms other than AWS and OCI. For ASA virtual deployed on AWS and OCI, the maximum supported memory is 128GB.
- Disk storage—The ASAv supports a minimum virtual disk of 8GB by default. Depending on the type of platform, the virtual disk support varies between 8GB to 10GB. Keep this in mind when you provision your VM resources.

Important The minimum memory requirement for the ASAv is 2 GB. If your current ASAv runs with less than 2 GB of memory, you cannot upgrade to version 9.13(1) or greater from an earlier version without increasing the memory of your ASAv machine. You can also redeploy a new ASAv machine with the latest version.

The minimum memory requirement for deploying ASAv with more than 1 vCPU is 4 GB.

For upgrading from ASAv version 9.14 and later to a latest version, the ASA virtual machine requires a minimum memory of 4 GB and 2 vCPU.

Session Limits for Licensed Features

Session limits for AnyConnect Client and TLS Proxy are determined by the installed ASAv platform entitlement tier, and enforced via a rate limiter. The following table summarizes the session limits based on the entitlement tier and rate limiter.

Table 1: ASAv Session Limits by Entitlement

Entitlement	AnyConnect Client Premium Peers	Total TLS Proxy Sessions	Rate Limiter
Standard Tier, 100M	50	500	150 Mbps
Standard Tier, 1G	250	500	1 Gbps
Standard Tier, 2G	750	1000	2 Gbps

Entitlement	AnyConnect Client Premium Peers	Total TLS Proxy Sessions	Rate Limiter
Standard Tier, 10G	10,000	10,000	10 Gbps
Standard Tier, 20G	20,000	20,000	20 Gbps

The session limits granted by an entitlement, as shown in the previous table, cannot exceed the session limits for the platform. The platform session limits are based on the amount of memory provisioned for the ASAv.

Table 2: ASAv Session Limits by Memory Requirement

Provisioned Memory	AnyConnect Client Premium Peers	Total TLS Proxy Sessions
2 GB to 7.9 GB	250	500
8 GB to 15.9 GB	750	1000
16 GB - 31.9 GB	10,000	10,000
32 GB to 64 GB	20,000	20,000
64 GB to 128 GB	20,000	20,000

Platform Limits

Firewall connections, concurrent and VLANs are platform limits based on the ASAv memory.

Note We limit the firewall connections to 100 when the ASAv is in an unlicensed state. Once licensed with any entitlement, the connections go to the platform limit. The minimum memory requirement for the ASAv is 2GB.

Table 3: Platform Limits

ASAv Memory	Firewall Conns, Concurrent	VLANs
2 GB to 7.9 GB	100,000	50
8 GB to 15.9 GB	500,000	200
16 GB to 31.9	2,000,000	1024
32 GB to 64 GB	4,000,000	1024

ASAv Private Cloud Entitlements (VMware, KVM, Hyper-V)

Because any ASAv license can be used on any supported ASAv vCPU/memory configuration, you have greater flexibility when you deploy the ASAv in a private cloud environment (VMware, KVM, Hyper-V).

Note ASAv50 and ASAv100 are not supported on HyperV.

Session limits for AnyConnect Client and TLS Proxy are determined by the installed ASAv platform entitlement tier, and enforced via a rate limiter. The following table summarizes the session limits based on the entitlement tier for the ASAv deployed to a private cloud environment, with the enforced rate limiter.

Note ASAv session limits are based on the amount of memory provisioned for the ASAv; see Table 2: ASAv Session Limits by Memory Requirement, on page 3.

Table 4: ASAv on VMware/KVM/HyperV Private Cloud - Licensed Feature Limits Based on Entitlement

RAM (GB)		Entitlement Support*				
Min	Max	Standard Tier, 100M	Standard Tier, 1G	Standard Tier, 2G	Standard Tier, 10G	Standard Tier, 20G
2	7.9	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500/20G
8	15.9	50/500/100M	250/500/1G	750/1000/2G	750/1000/10G	750/1000/20G
16	31.9	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	10K/10K/20G
32	64	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	20K/20K/20G
*AnyConnect Client Sessions / TLS Proxy Sessions / Rate Limiter per entitlement/instance.						

ASAv Public Cloud Entitlements (AWS)

Because any ASAv license can be used on any supported ASAv vCPU/memory configuration, you can deploy the ASAv on a wide variety AWS instances types. Session limits for AnyConnect Client and TLS Proxy are determined by the installed ASAv platform entitlement tier, and enforced via a rate limiter.

The following table summarizes the session limits and rate limiter based on the entitlement tier for AWS instance types. See "About ASAv Deployment On the AWS Cloud" for a breakdown of the AWS VM dimensions (vCPUs and memory) for the supported instances.

Table 5: ASAv on AWS - Licensed Feature Limits Based on Entitlement

Instance	BYOL Entitlement Support*				PAYG**
	Standard Tier, 100M	Standard Tier, 1G	Standard Tier, 2G	Standard Tier, 10G	
c5.xlarge	50/500/100M	250/500/1G	750/1000/2G	750/1000/10G	750/1000
c5.2xlarge	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	10K/10K
c4.large	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500

Instance	BYOL Entitlement Support*				PAYG**
	Standard Tier, 100M	Standard Tier, 1G	Standard Tier, 2G	Standard Tier, 10G	
c4.xlarge	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500
c4.2xlarge	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	750/1000
c3.large	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500
c3.xlarge	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500
c3.2xlarge	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	750/1000
m4.large	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500
m4.xlarge	50/500/100M	250/500/1G	250/500/2G	250/500/10G	10K/10K
m4.2xlarge	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	10K/10K
*AnyConnect Client Sessions / TLS Proxy Sessions / Rate Limiter per entitlement/instance. **AnyConnect Client Sessions / TLS Proxy Sessions. The Rate Limiter is not employed in PAYG mode.					

Pay-As-You-Go (PAYG) Mode

The following table summarizes the Smart Licensing entitlements for each tier for the hourly billing (PAYG) mode, which is based on the allocated memory.

Table 6: ASAv on AWS - Smart License Entitlements for PAYG

RAM (GB)	Hourly Billing Mode Entitlement
< 2 GB	Standard Tier, 100M (ASAv5)
2 GB to < 8 GB	Standard Tier, 1G (ASAv10)
8 GB to < 16 GB	Standard Tier, 2G (ASAv30)
16 GB < 32 GB	Standard Tier, 10G (ASAv50)
30 GB and higher	Standard Tier, 20G (ASAv100)

ASAv Public Cloud Entitlements (Azure)

Because any ASAv license can be used on any supported ASAv vCPU/memory configuration, you can deploy the ASAv on a wide variety Azure instances types. Session limits for AnyConnect Client and TLS Proxy are determined by the installed ASAv platform entitlement tier, and enforced via a rate limiter.

The following table summarizes the session limits and rate limiter based on the entitlement tier for the Azure instance types. See "About ASAv Deployment On the Microsoft Azure Cloud" for a breakdown of the Azure VM dimensions (vCPUs and memory) for the supported instances.

Note Pay-As-You-Go (PAYG) Mode is currently not supported for the ASAv on Azure.

Table 7: ASAv on Azure - Licensed Feature Limits Based on Entitlement

Instance	BYOL Entitlement Support*				
	Standard Tier, 100M	**Standard Tier, 1G**	**Standard Tier, 2G**	**Standard Tier, 10G**	**Standard Tier, 20G**
D1, D1_v2DS1, DS1_v2	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500/20G
D2, D2_v2, DS2, DS2_v2	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500/20G
D3, D3_v2, DS3, DS3_v2	50/500/100M	250/500/1G	750/1000/2G	750/1000/10G	750/1000/20G
D4, D4_v2, DS4, DS4_v2	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	10K/10K/20G
D5, D5_v2, DS5, DS5_v2	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	10K/20K/20G
D2_v3	50/500/100M	250/500/1G	750/1000/2G	750/1000/10G	750/1000/20G
D4_v3	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	10K/10K/20G
D8_v3	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	10K/10K/20G
F4, F4s	50/500/100M	250/500/1G	750/1000/2G	750/1000/10G	750/1000/20G
F8, F8s	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	10K/20K/20G
F16, F16s	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	10K/20K/20G
*AnyConnect Client Sessions / TLS Proxy Sessions / Rate Limiter per entitlement/instance.					

Guidelines and Limitations

The ASAv firewall functionality is very similar to the ASA hardware firewalls, but with the following guidelines and limitations.

Guidelines and Limitations for the ASAv (all entitlements)

Smart Licensing Guidelines

- The maximum supported number of vCPUs is 16. The maximum supported memory is 64GB for ASA virtual deployed on all platforms other than AWS and OCI. For ASA virtual deployed on AWS and OCI,

the maximum supported memory is 128GB. Any ASAv license can be used on any supported ASAv vCPU/memory configuration.

- Session limits for licensed features and unlicensed platform capabilities are set based on the amount of VM memory.
- Session limits for AnyConnect Client and TLS Proxy are determined by the ASAv platform entitlement; session limits are no longer associated with an ASAv model type (ASAv5/10/30/50/100).
- Session limits have a minimum memory requirement; in cases where the VM memory is below the minimum requirement, the session limits will be set for the maximum number supported by the amount of memory.
- There are no changes to existing entitlements; the entitlement SKU and display name will continue to include the model number (ASAv5/10/30/50/100).
- The entitlement sets the maximum throughput via a rate limiter.
- There is no change to customer ordering process.

Disk Storage

The ASAv supports a maximum virtual disk of 8 GB by default. You cannot increase the disk size beyond 8 GB. Keep this in mind when you provision your VM resources.

Context Mode Guidelines

Supported in single context mode only. Does not support multiple context mode.

Failover for High Availability Guidelines

For failover deployments, make sure that the standby unit has the same license entitlement; for example, both units should have the 2Gbps entitlement.

Important When creating a high availability pair using ASAv, it is necessary to add the data interfaces to each ASAv in the same order. If the exact same interfaces are added to each ASAv, but in different order, errors may be presented at the ASAv console. Failover functionality may also be affected.

Unsupported ASA Features

The ASAv does not support the following ASA features:

- Clustering (for all entitlements, except KVM and VMware)
- Multiple context mode
- Active/Active failover
- EtherChannels
- Shared AnyConnect Premium Licenses

Limitations

- The ASAv is not compatible with the 1.9.5 i40en host driver for the x710 NIC. Older or newer driver versions will work. (VMware only)

Guidelines and Limitations for the 1 GB Entitlement

Performance Guidelines

- Jumbo frame reservation on the 1 GB platform with 9 or more configured e1000 interfaces may cause the device to reload. If **jumbo-frame reservation** is enabled, reduce the number of interfaces to 8 or less. The exact number of interfaces will depend on how much memory is needed for the operation of other features configured, and could be less than 8.

Guidelines and Limitations for the 10 GB Entitlement

Performance Guidelines

- Supports 10Gbps of aggregated traffic.
- Supports the following practices to improve ASAv performance:
  - Numa nodes
  - Multiple RX queues
  - SR-IOV provisioning
  - See Performance Tuning, on page 34 and Performance Tuning, on page 53 for more information.
- CPU pinning is recommended to achieve full throughput rates; see Increasing Performance on ESXi Configurations, on page 34 and Increasing Performance on KVM Configurations, on page 53.
- Jumbo frame reservation with a mix of e1000 and i40e-vf interfaces may cause the i40e-vf interfaces to remain down. If **jumbo-frame reservation** is enabled, do not mix interface types that use e1000 and i40e-vf drivers.

Limitations

- Transparent mode is not supported.
- The ASAv is not compatible with the 1.9.5 i40en host driver for the x710 NIC. Older or newer driver versions will work. (VMware only)
- Not supported on Hyper-V.

Guidelines and Limitations for the 20 GB Entitlement

Performance Guidelines

- Supports 20Gbps of aggregated traffic.
- Supports the following practices to improve ASAv performance:
  - Numa nodes
  - Multiple RX queues
  - SR-IOV provisioning
  - See Performance Tuning, on page 34 and Performance Tuning, on page 53 for more information.
- CPU pinning is recommended to achieve full throughput rates; see Increasing Performance on ESXi Configurations, on page 34 and Increasing Performance on KVM Configurations, on page 53.

Limitations

- The ASAv is not compatible with the 1.9.5 i40en host driver for the x710 NIC. Older or newer driver versions will work. (VMware only)
- Transparent mode is not supported.
- Not supported on Amazon Web Services (AWS) and Hyper-V.

ASAv Interfaces and Virtual NICs

As a guest on a virtualized platform, the ASAv uses the network interfaces of the underlying physical platform. Each ASAv interface maps to a virtual NIC (vNIC).

- ASAv Interfaces
- Supported vNICs

ASAv Interfaces

The ASAv includes the following Gigabit Ethernet interfaces:

- Management 0/0

  For AWS and Azure, Management 0/0 can be a traffic-carrying “outside” interface.
- GigabitEthernet 0/0 through 0/8. Note that the GigabitEthernet 0/8 is used for the failover link when you deploy the ASAv as part of a failover pair.

Note To simply configuration migration, Ten GigabitEthernet interfaces, like those available on the VMXNET3 driver, are labeled GigabitEthernet. This has no impact on the actual interface speed and is cosmetic only.

The ASAv defines GigabitEthernet interfaces using the E1000 driver as 1Gbps links. Note that VMware no longer recommends using the E1000 driver.

- Hyper-V supports up to eight interfaces. Management 0/0 and GigabitEthernet 0/0 through 0/6. You can use GigabitEthernet 0/6 as a failover link.

Supported vNICs

The ASAv supports the following vNICs. Mixing vNICs, such as e1000 and vmxnet3, on the same ASAv is not supported.

Table 8: Supported vNics

vNIC Type	Hypervisor Support		ASAv Version	Notes
	VMware	KVM		
vmxnet3	Yes	No	9.9(2) and later	VMware default When using vmxnet3, you need to disable Large Receive Offload (LRO) to avoid poor TCP performance. See Disable LRO for VMware and VMXNET3, on page 10.
e1000	Yes	Yes	9.2(1) and later	Not recommended by VMware.
virtio	No	Yes	9.3(2.200) and later	KVM default
ixgbe-vf	Yes	Yes	9.8(1) and later	AWS default; ESXi and KVM for SR-IOV support.
i40e-vf	No	Yes	9.10(1) and later	KVM for SR-IOV support.

Disable LRO for VMware and VMXNET3

Large Receive Offload (LRO) is a technique for increasing inbound throughput of high-bandwidth network connections by reducing CPU overhead. It works by aggregating multiple incoming packets from a single stream into a larger buffer before they are passed higher up the networking stack, thus reducing the number of packets that have to be processed. However, LRO can lead to TCP perfomance problems where network packet delivery may not flow consistently and could be "bursty" in congested networks.

Important VMware enables LRO by default to increase overall throughput. It is therefore a requirement to disable LRO for ASAv deployments on this platform.

You can disable LRO directly on the ASAv machine. Power off the virtual machine before you make any configuration changes.

1. Find the ASAv machine in the vSphere Web Client inventory.
   a. To find a virtual machine, select a data center, folder, cluster, resource pool, or host.
   b. Click the **Related Objects** tab and click **Virtual Machines**.
2. Right-click the virtual machine and select **Edit Settings**.
3. Click **VM Options**.
4. Expand **Advanced**.
5. Under Configuration Parameters, click the **Edit Configuration** button.
6. Click **Add Parameter** and enter a name and value for the LRO parameters:
   - Net.VmxnetSwLROSL | 0
   - Net.Vmxnet3SwLRO | 0
   - Net.Vmxnet3HwLRO | 0
   - Net.Vmxnet2SwLRO | 0
   - Net.Vmxnet2HwLRO | 0

Note Optionally, if the LRO parameters exist, you can examine the values and change them if needed. If a parameter is equal to 1, LRO is enabled. If equal to 0, LRO is disabled.

7. Click **OK** to save your changes and exit the **Configuration Parameters** dialog box.
8. Click **Save**.

See the following VMware support articles for more information:

- VMware KB 1027511
- VMware KB 2055140

ASAv and SR-IOV Interface Provisioning

Single Root I/O Virtualization (SR-IOV) allows multiple VMs running a variety of guest operating systems to share a single PCIe network adapter within a host server. SR-IOV allows a VM to move data directly to and from the network adapter, bypassing the hypervisor for increased network throughput and lower server

CPU burden. Recent x86 server processors include chipset enhancements, such as Intel VT-d technology, that facilitate direct memory transfers and other operations required by SR-IOV.

The SR-IOV specification defines two device types:

- Physical Function (PF)—Essentially a static NIC, a PF is a full PCIe device that includes SR-IOV capabilities. PFs are discovered, managed, and configured as normal PCIe devices. A single PF can provide management and configuration for a set of virtual functions (VFs).
- Virtual Function (VF)—Similar to a dynamic vNIC, a VF is a full or lightweight virtual PCIe device that provides at least the necessary resources for data movements. A VF is not managed directly but is derived from and managed through a PF. One or more VFs can be assigned to a VM.

SR-IOV is defined and maintained by the Peripheral Component Interconnect Special Interest Group (PCI SIG), an industry organization that is chartered to develop and manage the PCI standard. For more information about SR-IOV, see PCI-SIG SR-IOV Primer: An Introduction to SR-IOV Technology.

Provisioning SR-IOV interfaces on the ASAv requires some planning, which starts with the appropriate operating system level, hardware and CPU, adapter types, and adapter settings.

Guidelines and Limitations for SR-IOV Interfaces

The specific hardware used for ASAv deployment can vary, depending on size and usage requirements. Licensing for the ASAv, on page 1 explains the compliant resource scenarios that match license entitlement for the different ASAv platforms. In addition, SR-IOV Virtual Functions require specific system resources.

Host Operating System and Hypervisor Support

SR-IOV support and VF drivers are available for:

- Linux 2.6.30 kernel or later

The ASAv with SR-IOV interfaces is currently supported on the following hypervisors:

- VMware vSphere/ESXi
- QEMU/KVM
- AWS

Hardware Platform Support

Note You should deploy the ASAv on any *server class* x86 CPU device capable of running the supported virtualization platforms.

This section describes hardware guidelines for SR-IOV interfaces. Although these are guidelines and not requirements, using hardware that does not meet these guidelines may result in functionality problems or poor performance.

A server that supports SR-IOV and that is equipped with an SR-IOV-capable PCIe adapter is required. You must be aware of the following hardware considerations:

- The capabilities of SR-IOV NICs, including the number of VFs available, differ across vendors and devices.
- Not all PCIe slots support SR-IOV.
- SR-IOV-capable PCIe slots may have different capabilities.

Note You should consult your manufacturer's documentation for SR-IOV support on your system.

- For VT-d enabled chipsets, motherboards, and CPUs, you can find information from this page of virtualization-capable IOMMU supporting hardware. VT-d is a required BIOS setting for SR-IOV systems.
- For VMware, you can search their online Compatibility Guide for SR-IOV support.
- For KVM, you can verify CPU compatibility. Note that for the ASAv on KVM we only support x86 hardware.

Note We tested the ASAv with the Cisco UCS C-Series Rack Server. Note that the Cisco UCS-B server does not support the ixgbe-vf vNIC.

Supported NICs for SR-IOV

- Intel Ethernet Network Adapter X710

Attention The ASAv is not compatible with the 1.9.5 i40en host driver for the x710 NIC. Older or newer driver versions will work. (VMware only)

- Intel Ethernet Server Adapter X520 - DA2

CPUs

- x86_64 multicore CPU

  Intel Sandy Bridge or later (Recommended)

Note We tested the ASAv on Intel's Broadwell CPU (E5-2699-v4) at 2.3GHz.

- Cores
  - Minimum of 8 physical cores per CPU socket
  - The 8 cores must be on a single socket.

Note CPU pinning is recommended to achieve full throughput rates on the ASAv50 and ASAv100; see Increasing Performance on ESXi Configurations, on page 34 and Increasing Performance on KVM Configurations, on page 53.

BIOS Settings

SR-IOV requires support in the BIOS as well as in the operating system instance or hypervisor that is running on the hardware. Check your system BIOS for the following settings:

- SR-IOV is enabled
- VT-x (Virtualization Technology) is enabled
- VT-d is enabled
- (Optional) Hyperthreading is disabled

We recommend that you verify the process with the vendor documentation because different systems have different methods to access and change BIOS settings.

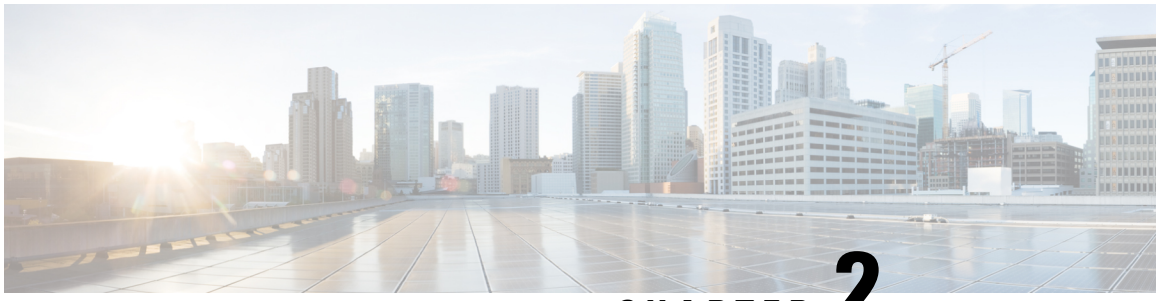
Limitations

Be aware of the following limitations when using ixgbe-vf interfaces:

- The guest VM is not allowed to set the VF to promiscuous mode. Because of this, transparent mode is not supported when using ixgbe-vf.
- The guest VM is not allowed to set the MAC address on the VF. Because of this, the MAC address is not transferred during HA like it is done on other ASA platforms and with other interface types. HA failover works by transferring the IP address from active to standby.

Note This limitation is applicable to the i40e-vf interfaces too.

- The Cisco UCS-B server does not support the ixgbe-vf vNIC.
- In a failover setup, when a paired ASAv (primary unit) fails, the standby ASAv unit takes over as the primary unit role and its interface IP address is updated with a new MAC address of the standby ASAv unit. Thereafter, the ASAv sends a gratuitous Address Resolution Protocol (ARP) update to announce the change in MAC address of the interface IP address to other devices on the same network. However, due to incompatibility with these types of interfaces, the gratuitous ARP update is not sent to the global IP address that is defined in the NAT or PAT statements for translating the interface IP address to global IP addresses.



CHAPTER 2

Deploy the ASAv Using VMware

You can deploy the ASAv on any *server class* x86 CPU device that is capable of running VMware ESXi.

Important The minimum memory requirement for the ASAv is 2GB. If your current ASAv runs with less than 2GB of memory, you cannot upgrade to 9.13(1)+ from an earlier version without increasing the memory of your ASAv machine. You can also redeploy a new ASAv machine with the latest version.

Guidelines and Limitations

You can create and deploy multiple instances of the ASAv on an ESXi server. The specific hardware used for ASAv deployments can vary, depending on the number of instances deployed and usage requirements. Each virtual appliance you create requires a minimum resource allocation—memory, number of CPUs, and disk space—on the host machine.

Important The ASAv deploys with a disk storage size of 8GB. It is not possible to change the resource allocation of the disk space.

Review the following guidelines and limitations before you deploy the ASAv.

ASAv on VMware ESXi System Requirements

Make sure to conform to the specifications below to ensure optimal performance. The ASAvASAv has the following requirements:

- The host CPU must be a *server class* x86-based Intel or AMD CPU with virtualization extension.

  For example, ASAv performance test labs use as minimum the following: Cisco Unified Computing System™ (Cisco UCS®) C series M4 server with the Intel® Xeon® CPU E5-2690v4 processors running at 2.6GHz.

- ASAv supports ESXi version 6.0, 6.5, 6.7, 7.0, 7.0 Upgrade 1, 7.0 Upgrade 2, and 7.0 Upgrade 3.

Recommended vNICs

The following vNICs are recommended in order of optimum performance.

- i40e in PCI passthrough—Dedicates the server's physical NIC to the VM and transfers packet data between the NIC and the VM via DMA (Direct Memory Access). No CPU cycles are required for moving packets.
- i40evf/ixgbe-vf—Effectively the same as above (DMAs packets between the NIC and the VM) but allows the NIC to be shared across multiple VMs. SR-IOV is generally preferred because it has more deployment flexibility. See Guidelines and Limitations, on page 38
- vmxnet3—This is a para-virtualized network driver that supports 10Gbps operation but also requires CPU cycles. This is the VMware default.

  When using vmxnet3, you need to disable Large Receive Offload (LRO) to avoid poor TCP performance.

Performance Optimizations

To achieve the best performance out of the ASAv, you can make adjustments to the both the VM and the host. See Performance Tuning, on page 34 for more information.

- **NUMA**—You can improve performance of the ASAv by isolating the CPU resources of the guest VM to a single non-uniform memory access (NUMA) node. See NUMA Guidelines, on page 34 for more information.
- **Receive Side Scaling**—The ASAv supports Receive Side Scaling (RSS), which is a technology utilized by network adapters to distribute network receive traffic to multiple processor cores. Supported on Version 9.13(1) and later. See Multiple RX Queues for Receive Side Scaling (RSS), on page 36 for more information.
- **VPN Optimization**—See VPN Optimization, on page 58 for additional considerations for optimizing VPN performance with the ASAv.

Clustering

Starting from version 9.17, clustering is supported on ASA virtual instances deployed on VMware. See ASA Cluster for the ASAv for more information.

OVF File Guidelines

The selection of the asav-vi.ovf or asav-esxi.ovf file is based on the deployment target:

- asav-vi—For deployment on vCenter
- asav-esxi—For deployment on ESXi (no vCenter)
- The ASAv OVF deployment does not support localization (installing the components in non-English mode). Be sure that the VMware vCenter and the LDAP servers in your environment are installed in an ASCII-compatible mode.
- You must set your keyboard to United States English before installing the ASAv and for using the VM console.
- When the ASAv is deployed, two different ISO images are mounted on the ESXi hypervisor:
  - The first drive mounted has the OVF environment variables generated by vSphere.
  - The second drive mounted is the day0.iso.

Attention You can unmount both drives after the ASAv machine has booted. However, Drive 1 (with the OVF environment variables) will always be mounted every time the ASAv is powered off/on, even if **Connect at Power On** is unchecked.

Export OVF Template Guidelines

The Export OVF Template in vSphere helps you export an existing ASAv instance package as an OVF template. You can use an exported OVF template for deploying the ASAv instance in the same or different environment. Before deploying the ASAv instance using an exported OVF template on vSphere, you must modify the configuration details in the OVF file to prevent deployment failure.

To modify the exported OVF file of ASAv.

1. Log in to the local machine where you have exported the OVF template.
2. Browse and open the OVF file in a text editor.
3. Ensure that the tag `<vmw:ExtraConfig vmw:key="monitor_control.pseudo_perfctr" vmw:value="TRUE"></vmw:ExtraConfig>` is present.
4. Delete the tag `<rasd:ResourceSubType>vmware.cdrom.iso</rasd:ResourceSubType>`.

   Or

   Replace the tag `<rasd:ResourceSubType>vmware.cdrom.iso</rasd:ResourceSubType>` with `<rasd:ResourceSubType>vmware.cdrom.remotepassthrough</rasd:ResourceSubType>`.

   See the Deploying an OVF fails on vCenter Server 5.1/5.5 when VMware tools are installed (2034422) published by VMware for more information.
5. Enter the property values for `UserPrivilege`, `OvfDeployment`, and `ControllerType`.

   For example:

```
- <Property ovf:qualifiers="ValueMap{"ovf", "ignore", "installer"}" ovf:type="string"
ovf:key="OvfDeployment">
+ <Property ovf:qualifiers="ValueMap{"ovf", "ignore", "installer"}" ovf:type="string"
ovf:key="OvfDeployment" ovf:value="ovf">

- <Property ovf:type="string" ovf:key="ControllerType">
+ <Property ovf:type="string" ovf:key="ControllerType" ovf:value="ASAv">
```

```
- <Property ovf:qualifiers="MinValue(0) MaxValue(255)" ovf:type="uint8"
ovf:key="UserPrivilege">
+ <Property ovf:qualifiers="MinValue(0) MaxValue(255)" ovf:type="uint8"
ovf:key="UserPrivilege" ovf:value="15">
```

6. Save the OVF file.
7. Deploy the ASAv using the OVF template. See, Deploy the ASAv Using the VMware vSphere Web Client.

Failover for High Availability Guidelines

For failover deployments, make sure that the standby unit has the same license entitlement; for example, both units should have the 2Gbps entitlement.

Important When creating a high availability pair using ASAv, it is necessary to add the data interfaces to each ASAv in the same order. If the exact same interfaces are added to each ASAv, but in different order, errors may be presented at the ASAv console. Failover functionality may also be affected.

For the ESX port group used for ASAv Inside interface or ASAv failover high availability link, configure the ESX port group failover order with two virtual NICs – one as active uplink and the other as standby uplink. This is necessary for the two VMs to ping each other or ASAv high availability link to be up.

vMotion Guidelines

- VMware requires that you only use shared storage if you plan to use vMotion. During ASAv deployment, if you have a host cluster you can either provision storage locally (on a specific host) or on a shared host. However, if you try to vMotion the ASAv to another host, using local storage will produce an error.

Memory and vCPU Allocation for Throughput and Licensing

- The memory allocated to the ASAv is sized specifically for the throughput level. Do not change the memory setting or any vCPU hardware settings in the Edit Settings dialog box unless you are requesting a license for a different throughput level. Under-provisioning can affect performance.

Note If you need to change the memory or vCPU hardware settings, use only the values documented in Licensing for the ASAv, on page 1. Do not use the VMware-recommended memory configuration minimum, default, and maximum values.

CPU Reservation

- By default the CPU reservation for the ASAv is 1000 MHz. You can change the amount of CPU resources allocated to the ASAv by using the shares, reservations, and limits settings (Edit Settings > Resources > CPU). Lowering the CPU Reservation setting from 1000 Mhz can be done if the ASAv can perform its required purpose while under the required traffic load with the lower setting. The amount of CPU

used by an ASAv depends on the hardware platform it is running on as well as the type and amount of work it is doing.

You can view the host's perspective of CPU usage for all of your virtual machines from the CPU Usage (MHz) chart, located in the Home view of the Virtual Machine Performance tab. Once you establish a benchmark for CPU usage when the ASAv is handling typical traffic volume, you can use that information as input when adjusting the CPU reservation.

See the CPU Performance Enhancement Advice published by VMware for more information.

- You can use the ASAv **show vm** and **show cpu** commands or the ASDM **Home** > **Device Dashboard** > **Device Information** > **Virtual Resources** tab or the **Monitoring** > **Properties** > **System Resources Graphs** > **CPU** pane to view the resource allocation and any resources that are over- or under-provisioned.

Transparent Mode on UCS B Series Hardware Guidelines

MAC flaps have been observed in some ASAv configurations running in transparent mode on Cisco UCS B Series hardware. When MAC addresses appear from different locations you will get dropped packets.

The following guidelines help prevent MAC flaps when you deploy the ASAv in transparent mode in VMware environments:

- VMware NIC teaming—If deploying the ASAv in transparent mode on UCS B Series, the Port Groups used for the Inside and Outside interfaces must have only 1 Active Uplink, and that uplink must be the same. You configure VMware NIC teaming in vCenter.

  See the VMware documentation for complete information on how to configure NIC teaming.

- ARP inspection—Enable ARP inspection on the ASAv and statically configure the MAC and ARP entry on the interface you expect to receive it on. See the Cisco ASA Series General Operations Configuration Guide for information about ARP inspection and how to enable it.

Additional Guidelines and Limitations

- The ASA Virtual boots without the two CD/DVD IDE drives if you are running ESXi 6.7, vCenter 6.7, ASA Virtual 9.12 and above.
- The vSphere Web Client is not supported for ASAv OVF deployment; use the vSphere client instead.

VMware Feature Support for the ASAv

The following table lists the VMware feature support for the ASAv.

Table 9: VMware Feature Support for the ASAv

Feature	Description	Support (Yes/No)	Comment
Cold Clone	The VM is powered off during cloning.	Yes	–

Feature	Description	Support (Yes/No)	Comment
DRS	Used for dynamic resource scheduling and distributed power management.	Yes	See VMware guidelines.
Hot add	The VM is running during an addition.	No	–
Hot clone	The VM is running during cloning.	No	–
Hot removal	The VM is running during removal.	No	–
Snapshot	The VM freezes for a few seconds.	Yes	Use with care. You may lose traffic. Failover may occur.
Suspend and resume	The VM is suspended, then resumed.	Yes	–
vCloud Director	Allows automatic deployment of VMs.	No	–
VM migration	The VM is powered off during migration.	Yes	–
vMotion	Used for live migration of VMs.	Yes	Use shared storage. See vMotion Guidelines, on page 18.
VMware FT	Used for HA on VMs.	No	Use ASAv failover for ASAv machine failures.
VMware HA	Used for ESXi and server failures.	Yes	Use ASAv failover for ASAv machine failures.
VMware HA with VM heartbeats	Used for VM failures.	No	Use ASAv failover for ASAv machine failures.
VMware vSphere Standalone Windows Client	Used to deploy VMs.	Yes	–
VMware vSphere Web Client	Used to deploy VMs.	Yes	–

Prerequisites

You can deploy the ASAv using the VMware vSphere Web Client, vSphere standalone client, or the OVF tool. See Cisco ASA Compatibility for system requirements.

Security Policy for a vSphere Standard Switch

For a vSphere switch, you can edit Layer 2 security policies and apply security policy exceptions for port groups used by the ASAv interfaces. See the following default settings:

- Promiscuous Mode: **Reject**
- MAC Address Changes: **Accept**
- Forged Transmits: **Accept**

You may need to modify these settings for the following ASAv configurations. See the vSphere documentation for more information.

Table 10: Port Group Security Policy Exceptions

	Routed Firewall Mode		Transparent Firewall Mode	
Security Exception	**No Failover**	**Failover**	**No Failover**	**Failover**
Promiscuous Mode	<any>	<any>	Accept	Accept
MAC Address Changes	<any>	Accept	<any>	Accept
Forged Transmits	<any>	Accept	Accept	Accept

Unpack the ASAv Software and Create a Day 0 Configuration File

You can prepare a Day 0 configuration file before you launch the ASAv. This file is a text file that contains the ASAv configuration to be applied when the ASAv is launched. This initial configuration is placed into a text file named “day0-config” in a working directory you chose, and is manipulated into a day0.iso file that is mounted and read on first boot. At the minimum, the Day 0 configuration file must contain commands to activate the management interface and set up the SSH server for public key authentication, but it can also contain a complete ASA configuration. A default day0.iso containing an empty day0-config is provided with the release. The day0.iso file (either your custom day0.iso or the default day0.iso) must be available during first boot.

Before you begin

We are using Linux in this example, but there are similar utilities for Windows.

- To automatically license the ASAv during initial deployment, place the Smart Licensing Identity (ID) Token that you downloaded from the Cisco Smart Software Manager in a text file named 'idtoken' in the same directory as the Day 0 configuration file.
- If you want to access and configure the ASAv from the serial port on the hypervisor instead of the virtual VGA console, you should include the **console serial** setting in the Day 0 configuration file to use the serial port on first boot.
- If you want to deploy the ASAv in transparent mode, you must use a known running ASA config file in transparent mode as the Day 0 configuration file. This does not apply to a Day 0 configuration file for a routed firewall.
- See the OVF file guidelines in Guidelines and Limitations, on page 15 for additional information about how the ISO images are mounted on the ESXi hypervisor.

Step 1 Download the ZIP file from Cisco.com, and save it to your local disk:

https://www.cisco.com/go/asa-software

Note A Cisco.com login and Cisco service contract are required.

Step 2 Unzip the file into a working directory. Do not remove any files from the directory. The following files are included:

- asav-vi.ovf—For vCenter deployments.
- asav-esxi.ovf—For non-vCenter deployments.
- boot.vmdk—Boot disk image.
- disk0.vmdk—ASAv disk image.
- day0.iso—An ISO containing a day0-config file and optionally an idtoken file.
- asav-vi.mf—Manifest file for vCenter deployments.
- asav-esxi.mf—Manifest file for non-vCenter deployments.

Step 3 Enter the CLI configuration for the ASAv in a text file called "day0-config." Add interface configurations for the three interfaces and any other configuration you want.

The fist line should begin with the ASA version. The day0-config should be a valid ASA configuration. The best way to generate the day0-config is to copy the desired parts of a running config from an existing ASA or ASAv. The order of the lines in the day0-config is important and should match the order seen in an existing **show running-config** command output.

We provide two examples of the day0-config file. The first example shows a day0-config when deploying an ASAv with Gigabit Ethernet interfaces. The second example shows a day0-config when deploying an ASAv with 10 Gigabit Ethernet interfaces. You would use this day0-config to deploy an ASAv with SR-IOV interfaces; see Guidelines and Limitations, on page 38.

Example:

```
ASA Version 9.4.1
!
console serial
interface management0/0
nameif management
security-level 100
```

```
ip address 192.168.1.1 255.255.255.0
no shutdown
interface gigabitethernet0/0
nameif inside
security-level 100
ip address 10.1.1.2 255.255.255.0
no shutdown
interface gigabitethernet0/1
nameif outside
security-level 0
ip address 198.51.100.2 255.255.255.0
no shutdown
http server enable
http 192.168.1.0 255.255.255.0 management
crypto key generate rsa modulus 1024
username AdminUser password paSSw0rd
ssh 192.168.1.0 255.255.255.0 management
aaa authentication ssh console LOCAL
call-home
http-proxy 10.1.1.1 port 443
license smart
feature tier standard
throughput level 2G
```

Example:

```
ASA Version 9.8.1
!
console serial
interface management 0/0
management-only
nameif management
security-level 0
ip address 192.168.0.230 255.255.255.0
!
interface GigabitEthernet0/0
nameif inside
security-level 100
ip address 10.10.10.10 255.255.255.0
!
interface GigabitEthernet0/1
nameif outside
security-level 0
ip address 10.10.20.10 255.255.255.0
!
route management 0.0.0.0 0.0.0.0 192.168.0.254
!
username cisco password cisco123 privilege 15
!
aaa authentication ssh console LOCAL
ssh 0.0.0.0 0.0.0.0 management
ssh timeout 60
ssh version 2
!
http 0.0.0.0 0.0.0.0 management
!
logging enable
logging timestamp
logging buffer-size 99999
logging buffered debugging
logging trap debugging
!
dns domain-lookup management
DNS server-group DefaultDNS
name-server 64.102.6.247
```

```
!
license smart
feature tier standard
throughput level 10G
!
crypto key generate rsa modulus 2048
```

Step 4 (Optional) Download the Smart License identity token file issued by the Cisco Smart Software Manager to your PC.

Step 5 (Optional) Copy the ID token from the download file and put it in a text file named 'idtoken' that only contains the ID token.

The Identity Token automatically registers the ASAv with the Smart Licensing server.

Step 6 Generate the virtual CD-ROM by converting the text file to an ISO file:

Example:

```
stack@user-ubuntu:-/KvmAsa$ sudo genisoimage -r -o day0.iso day0-config idtoken
I: input-charset not specified, using utf-8 (detected in locale settings)
Total translation table size: 0
Total rockridge attributes bytes: 252
Total directory bytes: 0
Path table size (byptes): 10
Max brk space used 0
176 extents written (0 MB)
stack@user-ubuntu:-/KvmAsa$
```

Step 7 Compute a new SHA1 value on Linux for the day0.iso:

Example:

```
openssl dgst -sha1 day0.iso
SHA1(day0.iso)= e5bee36e1eb1a2b109311c59e2f1ec9f731ecb66 day0.iso
```

Step 8 Include the new checksum in the asav-vi.mf file in the working directory and replace the day0.iso SHA1 value with the newly generated one.

Example:

```
SHA1(asav-vi.ovf)= de0f1878b8f1260e379ef853db4e790c8e92f2b2
SHA1(disk0.vmdk)= 898b26891cc68fa0c94ebd91532fc450da418b02
SHA1(boot.vmdk)= 6b0000ddebfc38ccc99ac2d4d5dbfb8abfb3d9c4
SHA1(day0.iso)= e5bee36e1eb1a2b109311c59e2f1ec9f731ecb66
```

Step 9 Copy the day0.iso file into the directory where you unzipped the ZIP file. You will overwrite the default (empty) day0.iso file.

When any VM is deployed from this directory, the configuration inside the newly generated day0.iso is applied.

Deploy the ASAv Using the VMware vSphere Web Client

This section describes how to deploy the ASAv using the VMware vSphere Web Client. The Web Client requires vCenter. If you do not have vCenter, see Deploy the ASAv Using the VMware vSphere Standalone Client and Day 0 Configuration, or Deploy the ASAv Using the OVF Tool and Day 0 Configuration.

- Access the vSphere Web Client and Install the Client Integration Plug-In, on page 25
- Deploy the ASAv Using the VMware vSphere Web Client, on page 24

Access the vSphere Web Client and Install the Client Integration Plug-In

This section describes how to access the vSphere Web Client. This section also describes how to install the Client Integration Plug-In, which is required for ASAv console access. Some Web Client features (including the plug-in) are not supported on the Macintosh. See the VMware website for complete client support information.

Step 1 Launch the VMware vSphere Web Client from your browser:

https://*vCenter_server:port***/vsphere-client/**

By default, the port is 9443.

Step 2 (One time only) Install the Client Integration Plug-in so that you can access the ASAv console.

a. In the login screen, download the plug-in by clicking **Download the Client Integration Plug-in**.

b. Close your browser and then install the plug-in using the installer.

c. After the plug-in installs, reconnect to the vSphere Web Client.

Step 3 Enter your username and password, and click **Login**, or check the **Use Windows session authentication** check box (Windows only).

Deploy the ASAv Using the VMware vSphere Web Client

To deploy the ASAv, use the VMware vSphere Web Client (or the vSphere Client) and a template file in the open virtualization format (OVF). You use the Deploy OVF Template wizard in the vSphere Web Client to deploy the Cisco package for the ASAv. The wizard parses the ASAv OVF file, creates the virtual machine on which you will run the ASAv, and installs the package.

Most of the wizard steps are standard for VMware. For additional information about the Deploy OVF Template, see the VMware vSphere Web Client online help.

Before you begin

You must have at least one network configured in vSphere (for management) before you deploy the ASAv.

Step 1 Download the ASAv ZIP file from Cisco.com, and save it to your PC:

http://www.cisco.com/go/asa-software

Note A Cisco.com login and Cisco service contract are required.

Step 2 In the vSphere Web Client **Navigator** pane, click **vCenter**.

Step 3 Click **Hosts and Clusters**.

Step 4 Right-click the data center, cluster, or host where you want to deploy the ASAv, and choose **Deploy OVF Template**. The **Deploy OVF Template** wizard appears.

Step 5 Follow the wizard screens as directed.

Step 6 In the **Setup networks** screen, map a network to each ASAv interface that you want to use.

The networks may not be in alphabetical order. If it is too difficult to find your networks, you can change the networks later from the Edit Settings dialog box. After you deploy, right-click the ASAv instance, and choose **Edit Settings** to access the **Edit Settings** dialog box. However that screen does not show the ASAv interface IDs (only Network Adapter IDs). See the following concordance of Network Adapter IDs and ASAv interface IDs:

Network Adapter ID	ASAv Interface ID
Network Adapter 1	Management 0/0
Network Adapter 2	GigabitEthernet 0/0
Network Adapter 3	GigabitEthernet 0/1
Network Adapter 4	GigabitEthernet 0/2
Network Adapter 5	GigabitEthernet 0/3
Network Adapter 6	GigabitEthernet 0/4
Network Adapter 7	GigabitEthernet 0/5
Network Adapter 8	GigabitEthernet 0/6
Network Adapter 9	GigabitEthernet 0/7
Network Adapter 10	GigabitEthernet 0/8

You do not need to use all ASAv interfaces; however, the vSphere Web Client requires you to assign a network to all interfaces. For interfaces you do not intend to use, you can simply leave the interface disabled within the ASAv configuration. After you deploy the ASAv, you can optionally return to the vSphere Web Client to delete the extra interfaces from the Edit Settings dialog box. For more information, see the vSphere Web Client online help.

Note For failover/HA deployments, GigabitEthernet 0/8 is preconfigured as the failover interface.

Step 7 If your network uses an HTTP proxy for Internet access, you must configure the proxy address for smart licensing in the **Smart Call Home Settings** area. This proxy is also used for Smart Call Home in general.

Step 8 For failover/HA deployments, in the Customize template screen, configure the following:

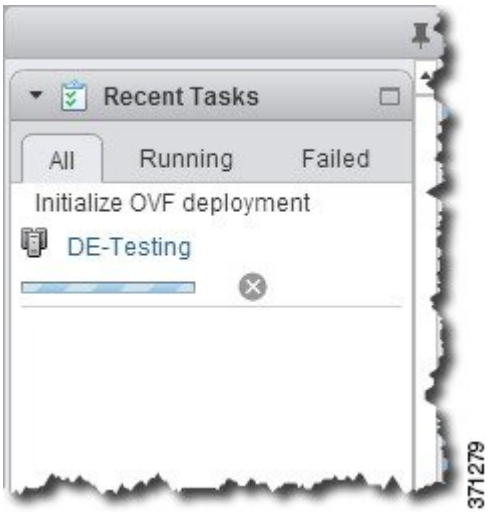
- Specify the standby management IP address.

  When you configure your interfaces, you must specify an active IP address and a standby IP address on the same network. When the primary unit fails over, the secondary unit assumes the IP addresses and MAC addresses of the primary unit and begins passing traffic. The unit that is now in a standby state takes over the standby IP addresses and MAC addresses. Because network devices see no change in the MAC to IP address pairing, no ARP entries change or time out anywhere on the network.

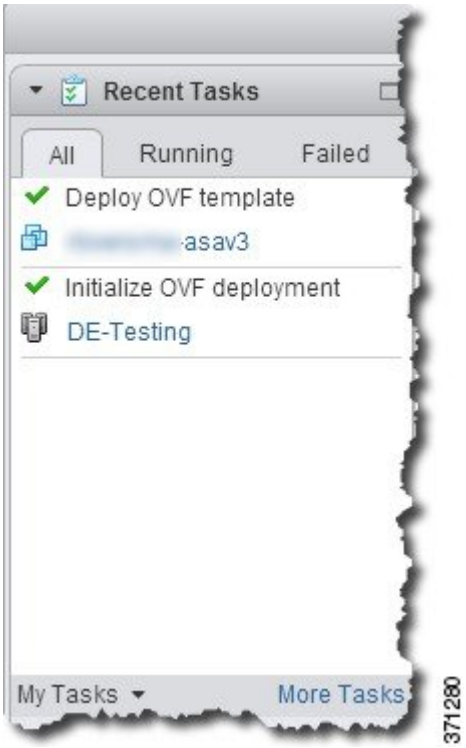
- Configure the failover link settings in the **HA Connection Settings** area.

  The two units in a failover pair constantly communicate over a failover link to determine the operating status of each unit. GigabitEthernet 0/8 is preconfigured as the failover link. Enter the active and standby IP addresses for the link on the same network.

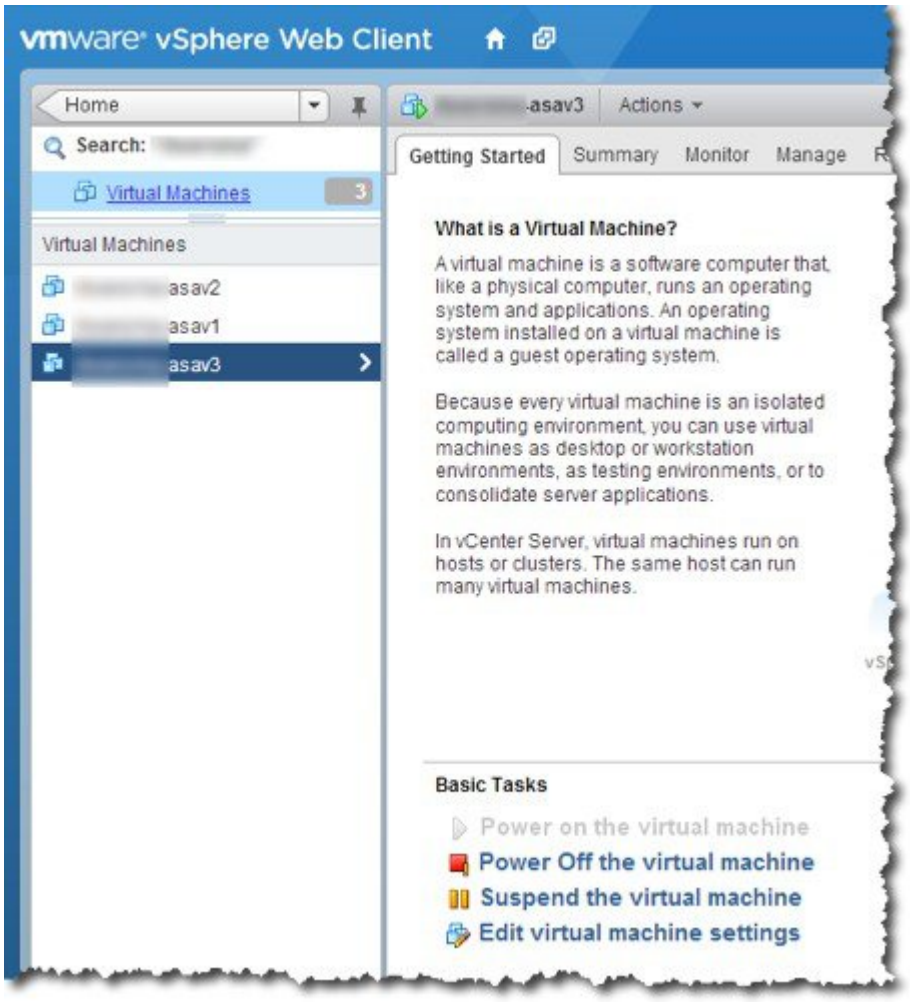
Step 9 After you complete the wizard, the vSphere Web Client processes the VM; you can see the "Initialize OVF deployment" status in the **Global Information** area **Recent Tasks** pane.



When it is finished, you see the Deploy OVF Template completion status.



The ASAv machine instance then appears under the specified data center in the Inventory.



Step 10 If the ASAv machine is not yet running, click **Power On the virtual machine**.

Wait for the ASAv to boot up before you try to connect with ASDM or to the console. When the ASAv starts up for the first time, it reads parameters provided through the OVF file and adds them to the ASAv system configuration. It then automatically restarts the boot process until it is up and running. This double boot process only occurs when you first deploy the ASAv. To view bootup messages, access the ASAv console by clicking the **Console** tab.

Step 11 For failover/HA deployments, repeat this procedure to add the secondary unit. See the following guidelines:

- Set the same throughput level as the primary unit.
- Enter the *exact same IP address settings* as for the primary unit. The bootstrap configurations on both units are identical except for the parameter identifying a unit as primary or secondary.

What to do next

To successfully register the ASAv with the Cisco Licensing Authority, the ASAv requires Internet access. You might need to perform additional configuration after deployment to achieve Internet access and successful license registration.

Deploy the ASAv Using the VMware vSphere Standalone Client and Day 0 Configuration

To deploy the ASAv, use the VMware vSphere Client and the open virtualization format (OVF) template file (asav-vi.ovf for a vCenter deployment or asav-esxi.ovf for a non-vCenter deployment). You use the Deploy OVF Template wizard in the vSphere Client to deploy the Cisco package for the ASAv. The wizard parses the ASAv OVF file, creates the virtual machine on which you will run the ASAv, and installs the package.

Most of the wizard steps are standard for VMware. For additional information about the Deploy OVF Template wizard, see the VMware vSphere Client online help.

Before you begin

- You must have at least one network configured in vSphere (for management) before you deploy the ASAv.
- Follow the steps in Unpack the ASAv Software and Create a Day 0 Configuration File, on page 21 to create the Day 0 configuration.

Step 1 Launch the VMware vSphere Client and choose **File** > **Deploy OVF Template**.

The Deploy OVF Template wizard appears.

Step 2 Browse to the working directory where you unzipped the asav-vi.ovf file and select it.

Step 3 The OVF Template details are shown. Proceed through the following screens. You do not have to change any configuration if you choose to use a custom Day 0 configuration file.

Step 4 A summary of the deployment settings is shown in the last screen. Click **Finish** to deploy the VM.

Step 5 Power on the ASAv, open the VMware console, and wait for the second boot.

Step 6 SSH to the ASAv and complete your desired configuration. If you do not have all the configuration that you wanted in the Day 0 configuration file, open a VMware console and complete the necessary configuration.

The ASAv is now fully operational.

Deploy the ASAv Using the OVF Tool and Day 0 Configuration

This section describes how to deploy the ASAv using the OVF tool, which requires a day 0 configuration file.

Before you begin

- The day0.iso file is required when you are deploying the ASAv using the OVF tool. You can use the default empty day0.iso file provided in the ZIP file, or you can use a customized Day 0 configuration file that you generate. See Unpack the ASAv Software and Create a Day 0 Configuration File, on page 21 for creating a Day 0 configuration file.
- Make sure the OVF tool is installed on a Linux or Windows PC and that it has connectivity to your target ESXi server.

Step 1 Verify the OVF tool is installed:

Example:

```
linuxprompt# which ovftool
```

Step 2 Create a .cmd file with the desired deployment options:

Example:

```
linuxprompt# cat launch.cmd
ovftool \
--name="asav-941-demo" \
--powerOn \
--deploymentOption=4Core8GB \
--diskMode=thin \
--datastore=datastore1 \
--acceptAllEulas \
--net:Management0-0="Portgroup_Mgmt" \
--net:GigabitEthernet0-1="Portgroup_Inside" \
--net:GigabitEthernet0-0="Portgroup_Outside" \
--prop:HARole=Standalone \
asav-esxi.ovf \
vi://root@10.1.2.3/
```

Step 3 Execute the cmd file:

Example:

```
linuxprompt# ./launch.cmd
```

The ASAv is powered on; wait for the second boot.

Step 4 SSH to the ASAv to complete configuration as needed. If more configuration is required, open the VMware console to the ASAv and apply the necessary configuration.

The ASAv is now fully operational.

Access the ASAv Console

In some cases with ASDM, you may need to use the CLI for troubleshooting. By default, you can access the built-in VMware vSphere console. Alternatively, you can configure a network serial console, which has better capabilities, including copy and paste.

- Use the VMware vSphere Console

- Configure a Network Serial Console Port

Note If you deploy the ASAv using a Day 0 configuration file, you can include the **console serial** setting in the configuration file to use the serial port on first boot instead of the virtual VGA console; see Unpack the ASAv Software and Create a Day 0 Configuration File, on page 21.

Use the VMware vSphere Console

For initial configuration or troubleshooting, access the CLI from the virtual console provided through the VMware vSphere Web Client. You can later configure CLI remote access for Telnet or SSH.

Before you begin

For the vSphere Web Client, install the Client Integration Plug-In, which is required for ASAv console access.

Step 1 In the VMware vSphere Web Client, right-click the ASAv instance in the Inventory, and choose **Open Console**. Or you can click **Launch Console** on the Summary tab.

Step 2 Click in the console and press **Enter**. Note: Press **Ctrl + Alt** to release the cursor.

If the ASAv is still starting up, you see bootup messages.

When the ASAv starts up for the first time, it reads parameters provided through the OVF file and adds them to the ASAv system configuration. It then automatically restarts the boot process until it is up and running. This double boot process only occurs when you first deploy the ASAv.

Note Until you install a license, throughput is limited to 100 Kbps so that you can perform preliminary connectivity tests. A license is required for regular operation. You also see the following messages repeated on the console until you install a license:

```
Warning: ASAv platform license state is Unlicensed.
Install ASAv platform license for full functionality.
```

You see the following prompt:

```
ciscoasa>
```

This prompt indicates that you are in user EXEC mode. Only basic commands are available from user EXEC mode.

Step 3 Access privileged EXEC mode:

Example:

```
ciscoasa> enable
```

The following prompt appears:

```
Password:
```

Step 4 Press the **Enter** key to continue. By default, the password is blank. If you previously set an enable password, enter it instead of pressing Enter.

The prompt changes to:

```
ciscoasa#
```

All nonconfiguration commands are available in privileged EXEC mode. You can also enter configuration mode from privileged EXEC mode.

To exit privileged mode, enter the **disable**, **exit**, or **quit** command.

Step 5 Access global configuration mode:

```
ciscoasa# configure terminal
```

The prompt changes to the following:

```
ciscoasa(config)#
```

You can begin to configure the ASAv from global configuration mode. To exit global configuration mode, enter the **exit**, **quit**, or **end** command.

Configure a Network Serial Console Port

For a better console experience, you can configure a network serial port singly or attached to a virtual serial port concentrator (vSPC) for console access. See the VMware vSphere documentation for details about each method. On the ASAv, you must send the console output to a serial port instead of to the virtual console. This procedure describes how to enable the serial port console.

Step 1 Configure a network serial port in VMware vSphere. See the VMware vSphere documentation.

Step 2 On the ASAv, create a file called "use_ttyS0" in the root directory of disk0. This file does not need to have any contents; it just needs to exist at this location:

disk0:/use_ttyS0

- From ASDM, you can upload an empty text file by that name using the **Tools** > **File Management** dialog box.
- At the vSphere console, you can copy an existing file (any file) in the file system to the new name. For example:

```
ciscoasa(config)# cd coredumpinfo
ciscoasa(config)# copy coredump.cfg disk0:/use_ttyS0
```

Step 3 Reload the ASAv.

- From ASDM, choose **Tools** > **System Reload**.
- At the vSphere console, enter **reload**.

  The ASAv stops sending to the vSphere console, and instead sends to the serial console.

Step 4 Telnet to the vSphere host IP address and the port number you specified when you added the serial port; or Telnet to the vSPC IP address and port.

Upgrade the vCPU or Throughput License

The ASAv uses a throughput license, which affects the number of vCPUs you can use.

If you want to increase (or decrease) the number of vCPUs for your ASAv, you can request a new license, apply the new license, and change the VM properties in VMware to match the new values.

Note The assigned vCPUs must match the ASAv CPU license or Throughput license. The RAM must also be sized correctly for the vCPUs. When upgrading or downgrading, be sure to follow this procedure and reconcile the license and vCPUs immediately. The ASAv does not operate properly when there is a persistent mismatch.

Step 1 Request a new license.

Step 2 Apply the new license. For failover pairs, apply new licenses to both units.

Step 3 Do one of the following, depending on whether you use failover:

- Failover—In the vSphere Web Client, power off the standby ASAv. For example, click the ASAv and then click **Power Off the virtual machine**, or right-click the ASAv and choose **Shut Down Guest OS**.
- No Failover—In the vSphere Web Client, power off the ASAv. For example, click the ASAv and then click **Power Off the virtual machine**, or right-click the ASAv and choose **Shut Down Guest OS**.

Step 4 Click the ASAv and then click **Edit Virtual machine settings** (or right-click the ASAv and choose **Edit Settings**).

The **Edit Settings** dialog box appears.

Step 5 Refer to the CPU and memory requirements in Licensing for the ASAv, on page 1 to determine the correct values for the new vCPU license.

Step 6 On the **Virtual Hardware** tab, for the **CPU**, choose the new value from the drop-down list.

Step 7 For the **Memory**, enter the new value for the RAM.

Step 8 Click **OK**.

Step 9 Power on the ASAv. For example, click **Power On the Virtual Machine**.

Step 10 For failover pairs:

a. Open a console to the active unit or launch ASDM on the active unit.

b. After the standby unit finishes starting up, fail over to the standby unit:

- ASDM: Choose **Monitoring** > **Properties** > **Failover** > **Status**, and click **Make Standby**.
- CLI: **failover active**

c. Repeat Steps 3 through 9 for the active unit.

What to do next

See Licensing for the ASAv, on page 1 for more information.

Performance Tuning

Increasing Performance on ESXi Configurations

You can increase the performance for an ASAv in the ESXi environment by tuning the ESXi host CPU configuration settings. The Scheduling Affinity option gives you control over how virtual machine CPUs are distributed across the host's physical cores (and hyperthreads if hyperthreading is enabled). By using this feature, you can assign each virtual machine to processors in the specified affinity set.

See the following VMware documents for more information:

- The *Administering CPU Resources* chapter of vSphere Resource Management.
- Performance Best Practices for VMware vSphere.
- The vSphere Client online help.

NUMA Guidelines

Non-Uniform Memory Access (NUMA) is a shared memory architecture that describes the placement of main memory modules with respect to processors in a multiprocessor system. When a processor accesses memory that does not lie within its own node (remote memory), data must be transferred over the NUMA connection at a rate that is slower than it would be when accessing local memory.

The x86 server architecture consists of multiple sockets and multiple cores within a socket. Each CPU socket along with its memory and I/O is referred to as a NUMA node. To efficiently read packets from memory, guest applications and associated peripherals (such as the NIC) should reside within the same node.

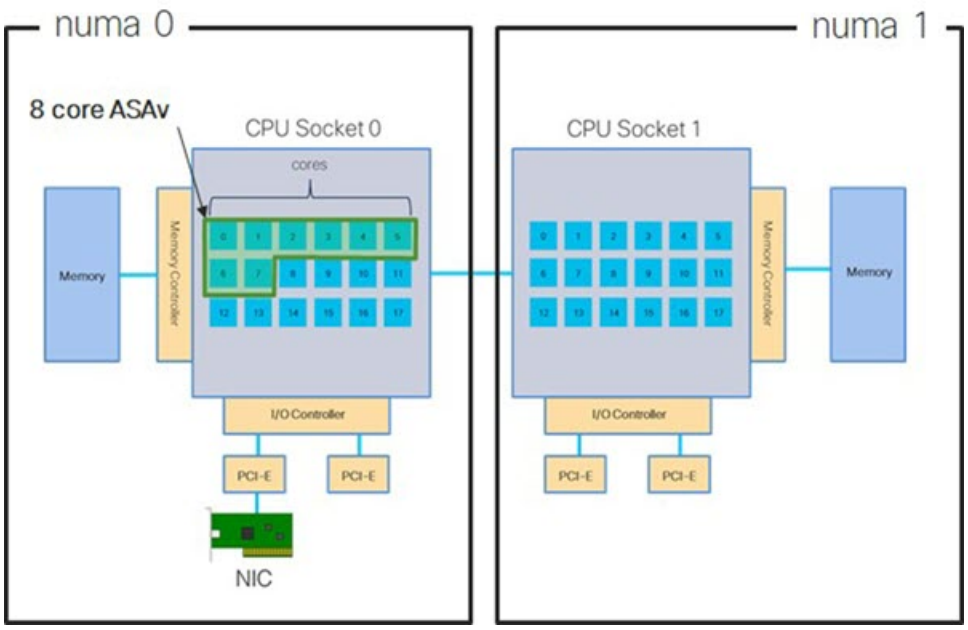
For optimum ASAv performance:

- The ASAv machine must run on a single numa node. If a single ASAv is deployed so that is runs across 2 sockets, the perfomance will be significantly degraded.
- An 8-core ASAv (Figure 1: 8-Core NUMA Architecture Example, on page 35) requires that each socket on the host CPU have a minimum of 8 cores per socket. Consideration must be given to other VMs running on the server.
- A 16-core ASAv (Figure 2: 16-Core ASAv NUMA Architecture Example, on page 35) requires that each socket on the host CPU have a minimum of 16 cores per socket. Consideration must be given to other VMs running on the server.
- The NIC should be on same NUMA node as ASAv machine.

Note ASAvdoes not support multi-Non-uniform memory access (NUMA) nodes and multiple CPU sockets for physical cores.

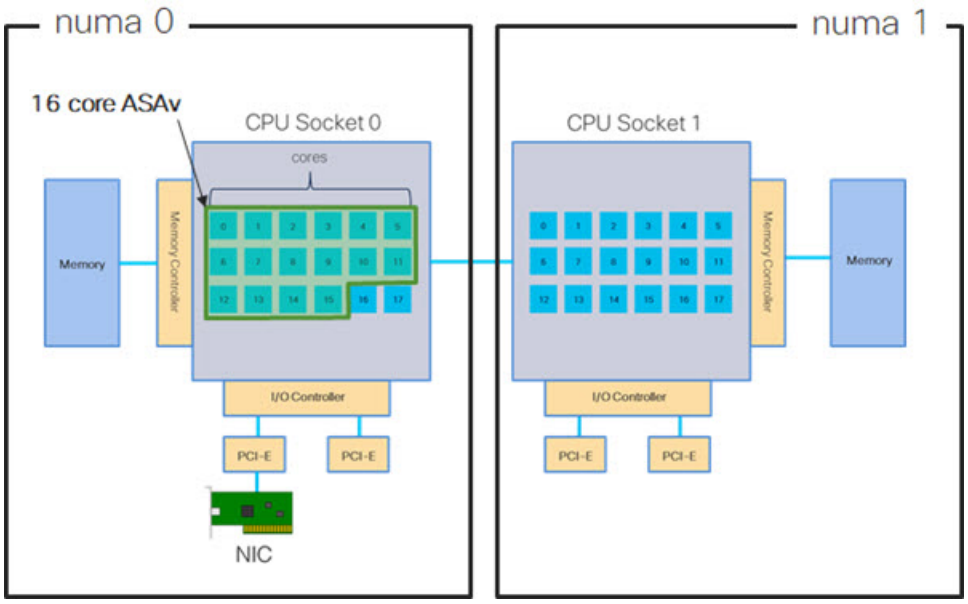
The following figure shows a server with two CPU sockets with each CPU having 18 cores. The 8-core ASAv requires that each socket on the host CPU have a minimum of 8 cores.

Figure 1: 8-Core NUMA Architecture Example



The following figure shows a server with two CPU sockets with each CPU having 18 cores. The 16-core ASAv requires that each socket on the host CPU have a minimum of 16 cores.

Figure 2: 16-Core ASAv NUMA Architecture Example



More information about using NUMA systems with ESXi can be found in the VMware document *vSphere Resource Management* for your VMware ESXi version. To check for more recent editions of this and other relevant documents, see http://www.vmware.com/support/pubs

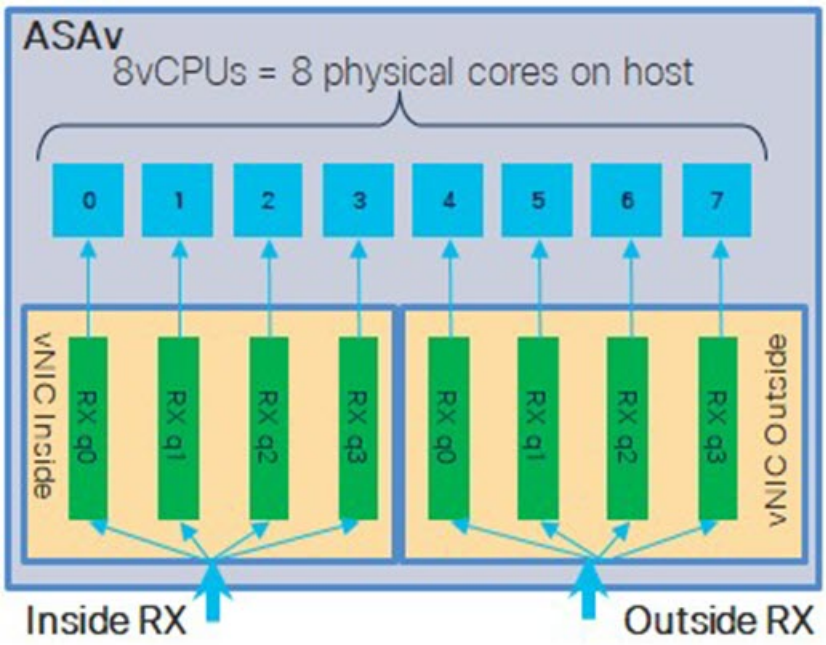
Multiple RX Queues for Receive Side Scaling (RSS)

The ASAv supports Receive Side Scaling (RSS), which is a technology utilized by network adapters to distribute network receive traffic in parallel to multiple processor cores. For maximum throughput, each vCPU (core) must have its own NIC RX queue. Note that a typical RA VPN deployment might use a single inside/outside pair of interfaces.

Important You need ASAv Version 9.13(1) or greater to use multiple RX queues.

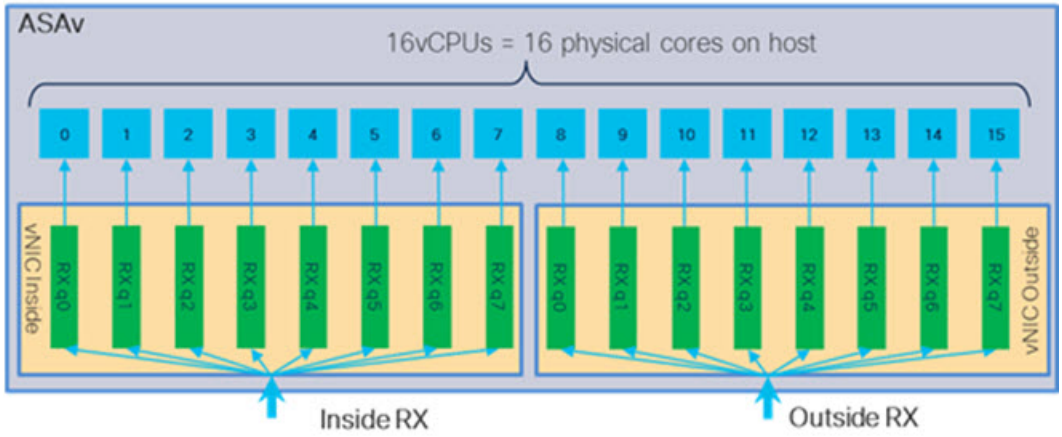
For an 8-core VM with an inside/outside pair of interfaces, each interface will have 4 RX queues, as shown in Figure 3: 8-Core ASAv RSS RX Queues, on page 36.

Figure 3: 8-Core ASAv RSS RX Queues



For a 16-core VM with an inside/outside pair of interfaces, each interface will have 8 RX queues, as shown in Figure 4: 16-Core ASAv RSS RX Queues, on page 37.

Figure 4: 16-Core ASAv RSS RX Queues



The following table presents the ASAv's vNICs for VMware and the number of supported RX queues. See Recommended vNICs, on page 16 for descriptions of the supported vNICs.

Table 11: VMware Recommended NICs/vNICs

NIC Card	vNIC Driver	Driver Technology	Number of RX Queues	Performance
x710*	i40e	PCI Passthrough	8 max	PCI Passthrough offers the highest performance of the NICs tested. In passthrough mode the NIC is dedicated to the ASAv and is not an optimal choice for virtual.
	i40evf	SR-IOV	4	SR-IOV with the x710 NIC has lower throughput (~30%) than PCI Passthrough. i40evf on VMware has a maximum of 4 RX queues per i40evf . 8 RX queues are needed for maximum throughput on a 16 core VM.
x520	ixgbe-vf	SR-IOV	2	—
	ixgbe	PCI Passthrough	6	The ixgbe driver (in PCI Passthrough mode) has 6 RX queues. Performance is on par with i40evf (SR-IOV).
N/A	vmxnet3	Para-virtualized	8 max	Not recommended for ASAv100.
N/A	e1000	Not recommended by VMware.		
*The ASAv is not compatible with the 1.9.5 i40en host driver for the x710 NIC. Older or newer driver versions will work. See Identify NIC Drivers and Firmware Versions, on page 38 for information on ESXCLI commands to identify or verify NIC driver and firmware versions.				

Identify NIC Drivers and Firmware Versions

If you need to identify or verify your specific firmware and driver version information, it is possible to find that data using ESXCLI commands.

- To get a list of the installed NICs, SSH to the pertinent host and run the `esxcli network nic list` command. This command should provide you with a record of devices and general information.
- After you have a list of the installed NICs, you can pull detailed configuration information. Run the `esxcli network nic get` command specifying the name of the NIC necessary: `esxcli network nic get -n <nic name>`.

Note General network adapter information can also be viewed from the VMware vSphere Client. The adapter and driver are found under **Physical Adapters** within the **Configure** tab.

SR-IOV Interface Provisioning

SR-IOV allows multiple VMs to share a single PCIe network adapter inside a host. SR-IOV defines these functions:

- Physical function (PF)—PFs are full PCIe functions that include the SR-IOV capabilities. These appear as regular static NICs on the host server.
- Virtual function (VF)—VFs are lightweight PCIe functions that help in data transfer. A VF is derived from, and managed through, a PF.

VFs are capable of providing up to 10 Gbps connectivity to ASAv machine within a virtualized operating system framework. This section explains how to configure VFs in a KVM environment. SR-IOV support on the ASAv is explained in ASAv and SR-IOV Interface Provisioning, on page 11.

Guidelines and Limitations

Guidelines for SR-IOV Interfaces

VMware vSphere 5.1 and later releases support SR-IOV in an environment with specific configurations only. Some features of vSphere are not functional when SR-IOV is enabled.

In addition to the system requirements for the ASAv and SR-IOV as described in Guidelines and Limitations for SR-IOV Interfaces, on page 12, you should review the Supported Configurations for Using SR-IOV in the VMware documentation for more information about requirements, supported NICs, availability of features, and upgrade requirements for VMware and SR-IOV.

This section shows various setup and configuration steps for provisioning SR-IOV interfaces on a VMware system. The information in this section was created from devices in a specific lab environment, using VMware ESXi 6.0 and vSphere Web Client, a Cisco UCS C Series server, and an Intel Ethernet Server Adapter X520 - DA2.

Limitations for SR-IOV Interfaces

When the ASAv is booted, be aware that SR-IOV interfaces can show up in reverse order when compared to the order presented in ESXi. This could cause interface configuration errors that result in a lack of network connectivity for a particular ASAv machine.

Caution It is important that you verify the interface mapping before you begin configuring the SR-IOV network interfaces on the ASAv. This ensures that the network interface configuration will apply to the correct physical MAC address interface on the VM host.

After the ASAv boots, you can confirm which MAC address maps to which interface. Use the **show interface** command to see detailed interface information, including the MAC address for an interface. Compare the MAC address to the results of the **show kernel ifconfig** command to confirm the correct interface assignment.

Check the ESXi Host BIOS

To deploy the ASAv with SR-IOV interfaces on VMware, virtualization needs to be supported and enabled. VMware provides several methods of verifying virtualization support, including their online Compatibility Guide for SR-IOV support as well as a downloadable CPU identification utility that detects whether virtualization is enabled or disabled.

You can also determine if virtualization is enabled in the BIOS by logging into the ESXi host.

Step 1 Log in to the ESXi Shell using one of the following methods:

- If you have direct access to the host, press Alt+F2 to open the login page on the machine's physical console.
- If you are connecting to the host remotely, use SSH or another remote console connection to start a session on the host.

Step 2 Enter a user name and password recognized by the host.

Step 3 Run the following command:

Example:

```
esxcfg-info|grep "\----\HV Support"
```

The output of the HV Support command indicates the type of hypervisor support available. These are the descriptions for the possible values:

0 - VT/AMD-V indicates that support is not available for this hardware.

1 - VT/AMD-V indicates that VT or AMD-V might be available but it is not supported for this hardware.

2 - VT/AMD-V indicates that VT or AMD-V is available but is currently not enabled in the BIOS.

3 - VT/AMD-V indicates that VT or AMD-V is enabled in the BIOS and can be used.

Example:

```
~ # esxcfg-info|grep "\----\HV Support"
         |----HV Support.........................3
```

The value 3 indicates the virtualization is supported and enabled.

What to do next

- Enable SR-IOV on the host physical adapter.

Enable SR-IOV on the Host Physical Adapter

Use the vSphere Web Client to enable SR-IOV and set the number of virtual functions on your host. You cannot connect virtual machines to virtual functions until you do so.

Before you begin

- Make sure you have an SR-IOV-compatible network interface card (NIC) installed; see Supported NICs for SR-IOV, on page 13.

Step 1 In the vSphere Web Client, navigate to the ESXi host where you want to enable SR-IOV.

Step 2 On the **Manage** tab, click **Networking** and choose **Physical adapters**.

You can look at the SR-IOV property to see whether a physical adapter supports SR-IOV.

Step 3 Select the physical adapter and click **Edit adapter settings**.

Step 4 Under SR-IOV, select **Enabled** from the **Status** drop-down menu.

Step 5 In the **Number of virtual functions** text box, type the number of virtual functions that you want to configure for the adapter.

Note For ASAv50, we recommend that you **DO NOT** use more than 1 VF per interface. Performance degradation is likely to occur if you share the physical interface with multiple virtual functions.

Step 6 Click **OK**.

Step 7 Restart the ESXi host.

The virtual functions become active on the NIC port represented by the physical adapter entry. They appear in the PCI Devices list in the **Settings** tab for the host.

What to do next

- Create a standard vSwitch to manage the SR-IOV functions and configurations.

Create a vSphere Switch

Create a vSphere switch to manage the SR-IOV interfaces.

Step 1 In the vSphere Web Client, navigate to the ESXi host.

Step 2 Under **Manage** select **Networking**, and then select **Virtual switches**.

Step 3 Click the **Add host networking** icon, which is the green globe icon with the plus (+) sign.

Step 4 Select a **Virtual Machine Port Group for a Standard Switch** connection type and click **Next**.

Step 5 Choose **New standard switch** and click **Next**.

Step 6 Add physical network adapters to the new standard switch.

a) Under Assigned adapters, click the green plus (+) sign to **Add adapters**.

b) Select the corresponding network interface for SR-IOV from the list. For example, Intel(R) 82599 10 Gigabit Dual Port Network Connection.

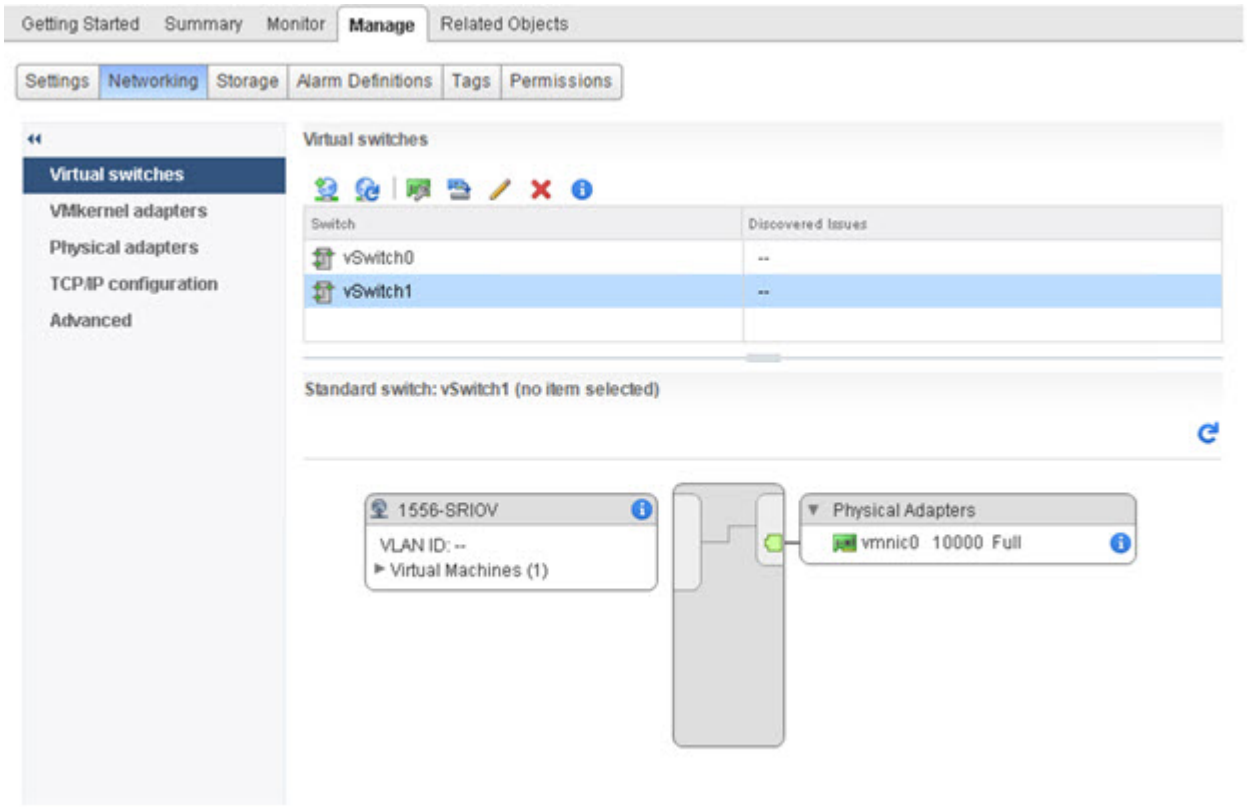
c) From the **Failover order group** drop-down menu, select from the **Active adapters**.

d) Click **OK**.

Step 7 Enter a **Network label** for the SR-IOV vSwitch and click **Next**.

Step 8 Review your selections on the **Ready to complete** page, then click **Finish**.

Figure 5: New vSwitch with an SR-IOV Interface attached



What to do next

- Review the compatibility level of your virtual machine.

Upgrade the Compatibility Level for Virtual Machines

The compatibility level determines the virtual hardware available to the virtual machine, which corresponds to the physical hardware available on the host machine. The ASAv machine needs to be at Hardware Level 10 or higher. This will expose the SR-IOV passthough feature to the ASAv. This procedure upgrades the ASAv to the latest supported virtual hardware version immediately.

For information about virtual machine hardware versions and compatibility, see the vSphere Virtual Machine Administration documentation.

Step 1 Log in to the vCenter Server from the vSphere Web Client.

Step 2 Locate the ASAv machine you wish to modify.

a) Select a datacenter, folder, cluster, resource pool, or host and click the **Related Objects** tab.

b) Click **Virtual Machines** and select the ASAv machine from the list.

Step 3 Power off the selected virtual machine.

Step 4 Right-click on the ASAv and select **Actions** > **All vCenter Actions** > **Compatibility** > **Upgrade VM Compatibility**.

Step 5 Click **Yes** to confirm the upgrade.

Step 6 Choose the **ESXi 5.5 and later** option for the virtual machines compatiblity.

Step 7 (Optional) Select **Only upgrade after normal guest OS shutdown**.

The selected virtual machine is upgraded to the corresponding hardware version for the Compatibility setting that you chose, and the new hardware version is updated in the Summary tab of the virtual machine.

What to do next

- Associate the ASAv with a virtual function through an SR-IOV passthrough network adapter.

Assign the SR-IOV NIC to the ASAv

To ensure that the ASAv machine and the physical NIC can exchange data, you must associate the ASAv with one or more virtual functions as SR-IOV passthrough network adapters. The following procedure explains how to assign the SR-IOV NIC to the ASAv machine using the vSphere Web Client.

Step 1 Log in to the vCenter Server from the vSphere Web Client.

Step 2 Locate the ASAv machine you wish to modify.

a) Select a datacenter, folder, cluster, resource pool, or host and click the **Related Objects** tab.

b) Click **Virtual Machines** and select the ASAv machine from the list.

Step 3 On the **Manage** tab of the virtual machine, select **Settings** > **VM Hardware**.

Step 4 Click **Edit** and choose the **Virtual Hardware** tab.

Step 5 From the **New device** drop-down menu, select **Network** and click **Add**.

A **New Network** interface appears.

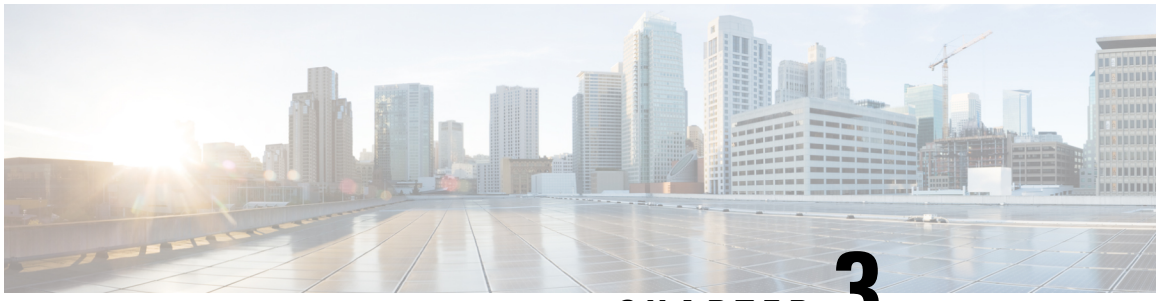
Step 6 Expand the **New Network** section and select an available SRIOV option.

Step 7 From the **Adapter Type** drop-down menu, select **SR-IOV passthrough**.

Step 8 From the **Physical function** drop-down menu, select the physical adapter that corresponds to the passthrough virtual machine adapter.

Step 9 Power on the virtual machine.

When you power on the virtual machine, the ESXi host selects a free virtual function from the physical adapter and maps it to the SR-IOV passthrough adapter. The host validates all properties of the virtual machine adapter and the underlying virtual function.



CHAPTER 3

Deploy the ASAv Using KVM

You can deploy the ASAv on any *server class* x86 CPU device that is capable of running the Kernel-based Virtual Machine (KVM).

Important The minimum memory requirement for the ASAv is 2GB. If your current ASAv runs with less than 2GB of memory, you cannot upgrade to 9.13(1)+ from an earlier version without increasing the memory of your ASAv machine. You can also redeploy a new ASAv machine with the latest version.

Guidelines and Limitations

The specific hardware used for ASAv deployments can vary, depending on the number of instances deployed and usage requirements. Each virtual appliance you create requires a minimum resource allocation—memory, number of CPUs, and disk space—on the host machine.

Important The ASAv deploys with a disk storage size of 8GB. It is not possible to change the resource allocation of the disk space.

Review the following guidelines and limitations before you deploy the ASAv.

ASAv on KVM System Requirements

Make sure to conform to the specifications below to ensure optimal performance. The ASAv has the following requirements:

- The host CPU must be a *server class* x86-based Intel or AMD CPU with virtualization extension.

For example, ASAv performance test labs use as minimum the following: Cisco Unified Computing System™ (Cisco UCS®) C series M4 server with the Intel® Xeon® CPU E5-2690v4 processors running at 2.6GHz.

Recommended vNICs

The following vNICs are recommended in order of optimum performance.

- i40e in PCI passthrough—Dedicates the server's physical NIC to the VM and transfers packet data between the NIC and the VM via DMA (Direct Memory Access). No CPU cycles are required for moving packets.
- i40evf/ixgbe-vf—Effectively the same as above (DMAs packets between the NIC and the VM) but allows the NIC to be shared across multiple VMs. SR-IOV is generally preferred because it has more deployment flexibility. See
- virtio—This is a para-virtualized network driver that supports 10Gbps operation but also requires CPU cycles.

Note ASAv instance running on KVM system might encounter data connectivity issues with the SR-IOV interface using the vNIC driver i40e version 2.11.25. We recommend you upgrade this vNIC version to other versions as a workaround to fix this issue.

Performance Optimizations

To achieve the best performance out of the ASAv, you can make adjustments to the both the VM and the host. See Performance Tuning, on page 53 for more information.

- **NUMA**—You can improve performance of the ASAv by isolating the CPU resources of the guest VM to a single non-uniform memory access (NUMA) node. See NUMA Guidelines, on page 54 for more information.
- **Receive Side Scaling**—The ASAv supports Receive Side Scaling (RSS), which is a technology utilized by network adapters to distribute network receive traffic to multiple processor cores. See Multiple RX Queues for Receive Side Scaling (RSS), on page 56 for more information.
- **VPN Optimization**—See VPN Optimization, on page 58 for additional considerations for optimizing VPN performance with the ASAv.

Clustering

Starting from version 9.17, clustering is supported on ASA virtual instances deployed on KVM. See ASA Cluster for the ASAv for more information.

CPU Pinning

CPU pinning is required for the ASAv to function in a KVM environment; see Enable CPU Pinning, on page 53.

Failover for High Availability Guidelines

For failover deployments, make sure that the standby unit has the same license entitlement; for example, both units should have the 2Gbps entitlement.

Important When creating a high availability pair using ASAv, it is necessary to add the data interfaces to each ASAv in the same order. If the exact same interfaces are added to each ASAv, but in different order, errors may be presented at the ASAv console. Failover functionality may also be affected.

ASAv on Proxmox VE

Proxmox Virtual Environment (VE) is an open-source server virtualization platform that can manage KVM virtual machines. Proxmox VE also provides a web-based management interface.

When you deploy the ASAv on Proxmox VE, you need to configure the VM to have an emulated serial port. Without the serial port, the ASAv will go into a loop during the bootup process. All management tasks can be done using the Proxmox VE web-based management interface.

Note For advanced users who are used to the comfort of the Unix shell or Windows Powershell, Proxmox VE provides a command line interface to manage all the components of your virtual environment. This command line interface has intelligent tab completion and full documentation in the form of UNIX man pages.

To have the ASAv boot properly the VM needs to have a serial device configured:

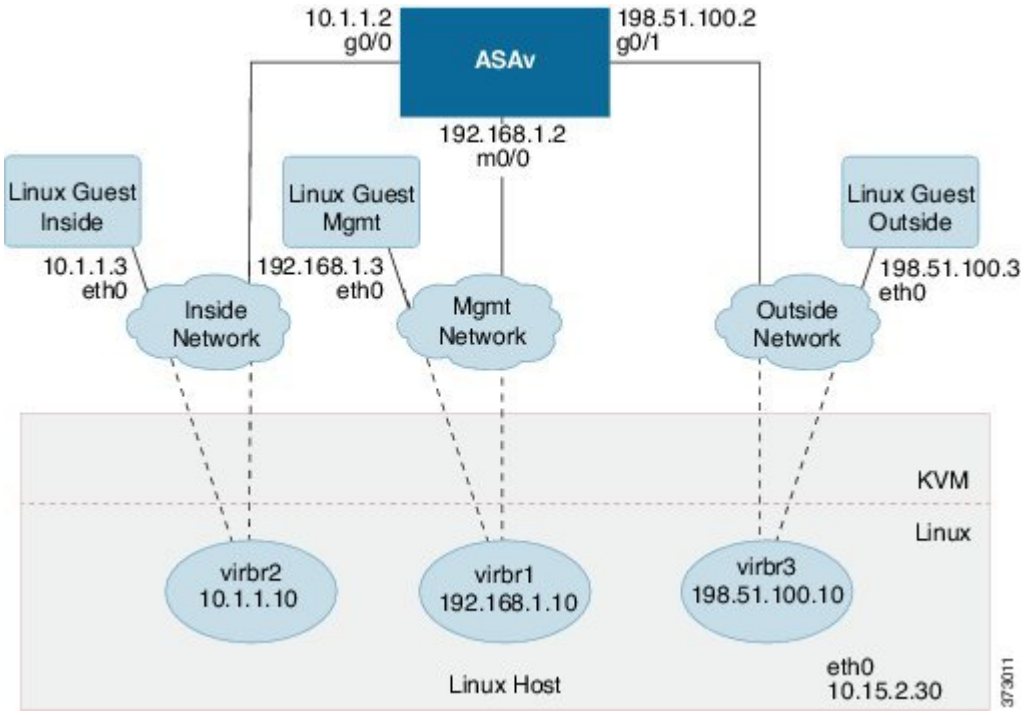
1. In the main management center, select the ASAv machine in the left navigation tree.
2. Power off the virtual machine.
3. Choose **Hardware** > **Add** > **Network Device** and add a serial port.
4. Power on the virtual machine.
5. Access the ASAv machine using Xterm.js.

See the Proxmox Serial Terminal page for information on how to setup and activate the terminal on the guest/server.

Overview

The following figure shows a sample network topology with ASAv and KVM. The procedures described in this chapter are based on the sample topology. The ASAv acts as the firewall between the inside and outside networks. A separate management network is also configured.

Figure 6: Sample ASAv Deployment Using KVM



Prerequisites

- Download the ASAv qcow2 file from Cisco.com and put it on your Linux host:

  http://www.cisco.com/go/asa-software

Note A Cisco.com login and Cisco service contract are required.

- For the purpose of the sample deployment in this document, we are assuming you are using Ubuntu 18.04 LTS. Install the following packages on top of the Ubuntu 18.04 LTS host:
  - qemu-kvm
  - libvirt-bin
  - bridge-utils
  - virt-manager
  - virtinst
  - virsh tools
  - genisoimage

- Performance is affected by the host and its configuration. You can maximize the throughput of the ASAv on KVM by tuning your host. For generic host-tuning concepts, see NFV Delivers Packet Processing Performance with Intel.
- Useful optimizations for Ubuntu 18.04 include the following:
  - macvtap—High performance Linux bridge; you can use macvtap instead of a Linux bridge. Note that you must configure specific settings to use macvtap instead of the Linux bridge.
  - Transparent Huge Pages—Increases memory page size and is on by default in Ubuntu 18.04.

    Hyperthread disabled—Reduces two vCPUs to one single core.
  - txqueuelength—Increases the default txqueuelength to 4000 packets and reduces drop rate.
  - pinning—Pins qemu and vhost processes to specific CPU cores; under certain conditions, pinning is a significant boost to performance.
- For information on optimizing a RHEL-based distribution, see Red Hat Enterprise Linux 7 Virtualization Tuning and Optimization Guide.
- For ASA software and ASAv hypervisor compatibility, see Cisco ASA Compatibility.

Prepare the Day 0 Configuration File

You can prepare a Day 0 configuration file before you launch the ASAv. This file is a text file that contains the ASAv configuration applied when the ASAv is launched. This initial configuration is placed into a text file named “day0-config” in a working directory you chose, and is manipulated into a day0.iso file that is mounted and read on first boot. At the minimum, the Day 0 configuration file must contain commands to activate the management interface and set up the SSH server for public key authentication, but it can also contain a complete ASA configuration.

The day0.iso file (either your custom day0.iso or the default day0.iso) must be available during first boot:

- To automatically license the ASAv during initial deployment, place the Smart Licensing Identity (ID) Token that you downloaded from the Cisco Smart Software Manager in a text file named ‘idtoken’ in the same directory as the Day 0 configuration file.
- If you want to access and configure the ASAv from the **serial port** on the hypervisor instead of the virtual VGA console, you should include the console serial setting in the Day 0 configuration file to use the serial port on first boot.
- If you want to deploy the ASAv in transparent mode, you must use a known running ASA config file in transparent mode as the Day 0 configuration file. This does not apply to a Day 0 configuration file for a routed firewall.

Note We are using Linux in this example, but there are similar utilities for Windows.

Step 1 Enter the CLI configuration for the ASAv in a text file called “day0-config.” Add interface configurations for the three interfaces and any other configuration you want.

The fist line should begin with the ASA version. The day0-config should be a valid ASA configuration. The best way to generate the day0-config is to copy the relevant parts of a running config from an existing ASA or ASAv. The order of the lines in the day0-config is important and should match the order seen in an existing **show running-config** command output.

Example:

```
ASA Version 9.4.1
!
console serial
interface management0/0
nameif management
security-level 100
ip address 192.168.1.2 255.255.255.0
no shutdown
interface gigabitethernet0/0
nameif inside
security-level 100
ip address 10.1.1.2 255.255.255.0
no shutdown
interface gigabitethernet0/1
nameif outside
security-level 0
ip address 198.51.100.2 255.255.255.0
no shutdown
http server enable
http 192.168.1.0 255.255.255.0 management
crypto key generate rsa modulus 1024
username AdminUser password paSSw0rd
ssh 192.168.1.0 255.255.255.0 management
aaa authentication ssh console LOCAL
```

Step 2 (Optional) For automated licensing during initial ASAv deployment, make sure the following information is in the day0-config file:

- Management interface IP address
- (Optional) HTTP proxy to use for Smart Licensing
- A **route** command that enables connectivity to the HTTP proxy (if specified) or to tools.cisco.com
- A DNS server that resolves tools.cisco.com to an IP address
- Smart Licensing configuration specifying the ASAv license you are requesting
- (Optional) A unique host name to make the ASAv easier to find in CSSM

Step 3 (Optional) Download the Smart License identity token file issued by the Cisco Smart Software Manager to your computer, copy the ID token from the download file, and put it a text file named ‘idtoken’ that only contains the ID token.

Step 4 Generate the virtual CD-ROM by converting the text file to an ISO file:

Example:

```
stack@user-ubuntu:-/KvmAsa$ sudo genisoimage -r -o day0.iso day0-config idtoken
I: input-charset not specified, using utf-8 (detected in locale settings)
Total translation table size: 0
Total rockridge attributes bytes: 252
Total directory bytes: 0
Path table size (byptes): 10
Max brk space used 0
176 extents written (0 MB)
```

```
stack@user-ubuntu:-/KvmAsa$
```

The Identity Token automatically registers the ASAv with the Smart Licensing server.

Step 5 Repeat Steps 1 through 5 to create separate default configuration files with the appropriate IP addresses for each ASAv you want to deploy.

Prepare the Virtual Bridge XML Files

You need to set up virtual networks that connect the ASAv guests to the KVM host and that connect the guests to each other.

Note This procedure does not establish connectivity to the external world outside the KVM host.

Prepare the virtual bridge XML files on the KVM host. For the sample virtual network topology described in Prepare the Day 0 Configuration File, on page 49, you need the following three virtual bridge files: virbr1.xml, virbr2.xml, and virbr3.xml (you must use these three filenames; for example, virbr0 is not allowed because it already exists). Each file has the information needed to set up the virtual bridges. You must give the virtual bridge a name and a unique MAC address. Providing an IP address is optional.

Step 1 Create three virtual network bridge XML files. For example, virbr1.xml, virbr2.xml, and virbr3.xml:

Example:

```
<network>
<name>virbr1</name>
<bridge name='virbr1' stp='on' delay='0' />
<mac address='52:54:00:05:6e:00' />
<ip address='192.168.1.10' netmask='255.255.255.0' />
</network>
```

Example:

```
<network>
<name>virbr2</name>
<bridge name='virbr2' stp='on' delay='0' />
<mac address='52:54:00:05:6e:01' />
<ip address='10.1.1.10' netmask='255.255.255.0' />
</network>
```

Example:

```
<network>
<name>virbr3</name>
<bridge name='virbr3' stp='on' delay='0' />
<mac address='52:54:00:05:6e:02' />
<ip address='198.51.100.10' netmask='255.255.255.0' />
</network>
```

Step 2 Create a script that contains the following (in our example, we name the script virt_network_setup.sh):

```
virsh net-create virbr1.xml
virsh net-create virbr2.xml
virsh net-create virbr3.xml
```

Step 3 Run this script to set up the virtual network. The script brings up the virtual networks. The networks stay up as long as the KVM host is running.

```
stack@user-ubuntu:-/KvmAsa$ virt_network_setup.sh
```

Note If you reload the Linux host, you must rerun the virt_network_setup.sh script. It does not persist over reboots.

Step 4 Verify that the virtual networks were created:

```
stack@user-ubuntu:-/KvmAsa$ brctl show
bridge name bridge id STP enabled Interfaces
virbr0 8000.0000000000000 yes
virbr1 8000.5254000056eed yes virb1-nic
virbr2 8000.5254000056eee yes virb2-nic
virbr3 8000.5254000056eec yes virb3-nic
stack@user-ubuntu:-/KvmAsa$
```

Step 5 Display the IP address assigned to the virbr1 bridge. This is the IP address that you assigned in the XML file.

```
stack@user-ubuntu:-/KvmAsa$ ip address show virbr1
S: virbr1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
link/ether 52:54:00:05:6e:00 brd ff:ff:ff:ff:ff:ff
inet 192.168.1.10/24 brd 192.168.1.255 scope global virbr1
valid_lft forever preferred_lft forever
```

Deploy the ASAv

Use a virt-install based deployment script to launch the ASAv.

Step 1 Create a virt-install script called “virt_install_asav.sh.”

The name of the ASAv machine must be unique across all other VMs on this KVM host.

The ASAv supports up to 10 networks. This example uses three networks. The order of the network bridge clauses is important. The first one listed is always the management interface of the ASAv (Management 0/0), the second one listed is GigabitEthernet 0/0 of the ASAv, and the third one listed is GigabitEthernet 0/1 of the ASAv, and so on up through GigabitEthernet 0/8. The virtual NIC must be Virtio.

Example:

```
virt-install \
--connect=qemu:///system \
--network network=default,model=virtio \
--network network=default,model=virtio \
--network network=default,model=virtio \
--name=asav \
--cpu host \
--arch=x86_64 \
--machine=pc-1.0 \
--vcpus=1 \
--ram=2048 \
```

```
--os-type=linux \
--virt-type=kvm \
--import \
--disk path=/home/kvmperf/Images/desmo.qcow2,format=qcow2,device=disk,bus=virtio,cache=none \
--disk path=/home/kvmperf/asav_day0.iso,format=iso,device=cdrom \
--console pty,target_type=virtio \
--serial tcp,host=127.0.0.1:4554,mode=bind,protocol=telnet
```

Step 2 Run the virt_install script:

Example:

```
stack@user-ubuntu:-/KvmAsa$ ./virt_install_asav.sh

Starting install...
Creating domain...
```

A window appears displaying the console of the VM. You can see that the VM is booting. It takes a few minutes for the VM to boot. Once the VM stops booting you can issue CLI commands from the console screen.

Performance Tuning

Increasing Performance on KVM Configurations

You can increase the performance for an ASAv in the KVM environment by changing settings on the KVM host. These settings are independent of the configuration settings on the host server. This option is available in Red Hat Enterprise Linux 7.0 KVM.

You can improve performance on KVM configurations by enabling CPU pinning.

Enable CPU Pinning

ASAv requires that you use the KVM CPU affinity option to increase the performance of the ASAv in KVM environments. Processor affinity, or CPU pinning, enables the binding and unbinding of a process or a thread to a central processing unit (CPU) or a range of CPUs, so that the process or thread will execute only on the designated CPU or CPUs rather than any CPU.

Configure host aggregates to deploy instances that use CPU pinning on different hosts from instances that do not, to avoid unpinned instances using the resourcing requirements of pinned instances.

Attention Do not deploy instances with NUMA topology on the same hosts as instances that do not have NUMA topology.

To use this option, configure CPU pinning on the KVM host.

Step 1 In the KVM host environment, verify the host topology to find out how many vCPUs are available for pinning:

Example:

```
virsh nodeinfo
```

Step 2 Verify the available vCPU numbers:

Example:

```
virsh capabilities
```

Step 3 Pin the vCPUs to sets of processor cores:

Example:

```
virsh vcpupin <vm-name> <vcpu-number> <host-core-number>
```

The **virsh vcpupin** command must be executed for each vCPU on your ASAv. The following example shows the KVM commands needed if you have an ASAv configuration with four vCPUs and the host has eight cores:

```
virsh vcpupin asav 0 2
virsh vcpupin asav 1 3
virsh vcpupin asav 2 4
virsh vcpupin asav 3 5
```

The host core number can be any number from 0 to 7. For more information, see the KVM documentation.

Note When configuring CPU pinning, carefully consider the CPU topology of the host server. If using a server configured with multiple cores, do not configure CPU pinning across multiple sockets.

The downside of improving performance on KVM configuration is that it requires dedicated system resources.

NUMA Guidelines

Non-Uniform Memory Access (NUMA) is a shared memory architecture that describes the placement of main memory modules with respect to processors in a multiprocessor system. When a processor accesses memory that does not lie within its own node (remote memory), data must be transferred over the NUMA connection at a rate that is slower than it would be when accessing local memory.

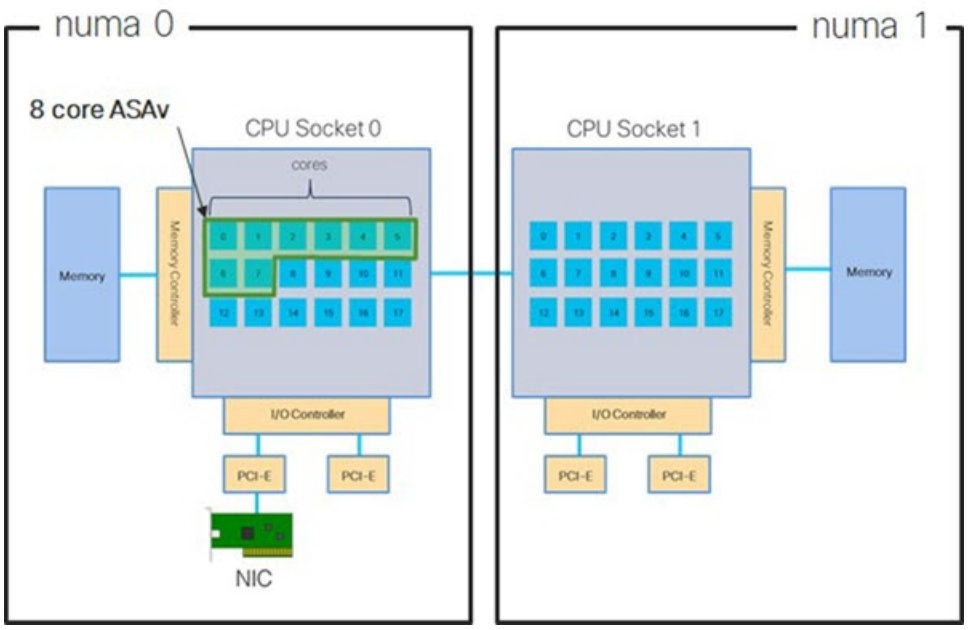
The x86 server architecture consists of multiple sockets and multiple cores within a socket. Each CPU socket along with its memory and I/O is referred to as a NUMA node. To efficiently read packets from memory, guest applications and associated peripherals (such as the NIC) should reside within the same node.

For optimum ASAv performance:

- The ASAv machine must run on a single numa node. If a single ASAv is deployed so that is runs across 2 sockets, the perfomance will be significantly degraded.
- An 8-core ASAv (Figure 7: 8-Core ASAv NUMA Architecture Example, on page 55) requires that each socket on the host CPU have a minimum of 8 cores per socket. Consideration must be given to other VMs running on the server.
- A 16-core ASAv (Figure 8: 16-Core ASAv NUMA Architecture Example, on page 55) requires that each socket on the host CPU have a minimum of 16 cores per socket. Consideration must be given to other VMs running on the server.
- The NIC should be on same NUMA node as ASAv machine.

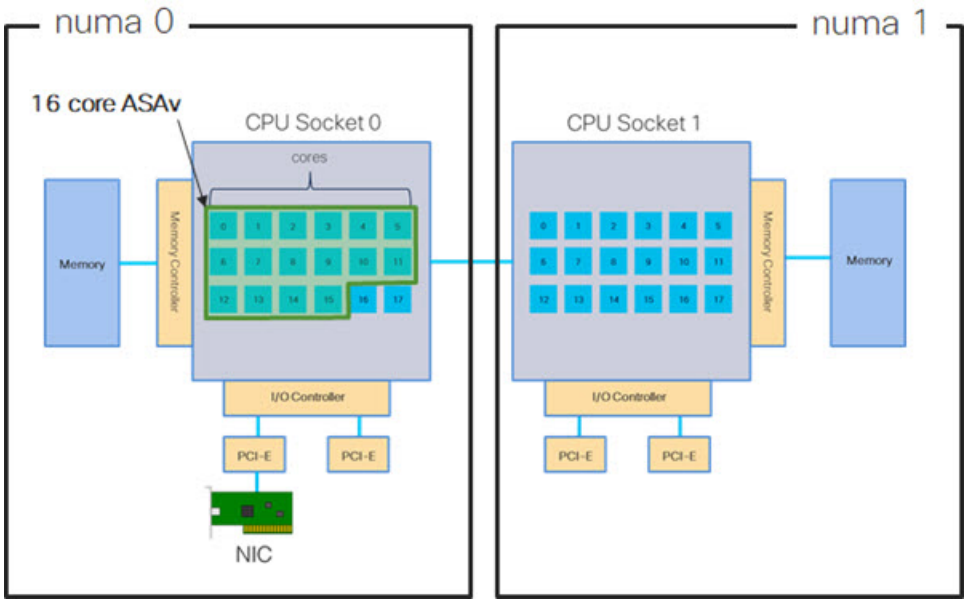
The following figure shows a server with two CPU sockets with each CPU having 18 cores. The 8-core ASAv requires that each socket on the host CPU have a minimum of 8 cores.

Figure 7: 8-Core ASAv NUMA Architecture Example



The following figure shows a server with two CPU sockets with each CPU having 18 cores. The 16-core ASAv requires that each socket on the host CPU have a minimum of 16 cores.

Figure 8: 16-Core ASAv NUMA Architecture Example



NUMA Optimization

Optimally, the ASAv machine should run on the same numa node that the NICs are running on. To do this:

1. Determine which node the NICs are on by using "lstopo" to show a diagram of the nodes. Locate the NICs and take note to which node they are attached.
2. At the KVM Host, use `virsh list` to find the ASAv.
3. Edit the VM by: `virsh edit <VM Number>`.
4. Align ASAv on the chosen node. The following examples assume 18-core nodes.

   Align onto Node 0:

   ```
   <vcpu placement='static' cpuset='0-17'>16</vcpu>
   <numatune>
       <memory mode='strict' nodeset='0'/>
   </numatune>
   ```

   Align onto Node 1:

   ```
   <vcpu placement='static' cpuset='18-35'>16</vcpu>
   <numatune>
       <memory mode='strict' nodeset='1'/>
   </numatune>
   ```

5. Save the .xml change and power cycle the ASAv machine.
6. To ensure your VM is running on the desired node, perform a `ps aux | grep <name of your ASAv VM>` to get the process ID.
7. Run `sudo numastat -c <ASAv VM Process ID>` to see if the ASAv machine is properly aligned.

More information about using NUMA tuning with KVM can be found in the RedHat document 9.3. libvirt NUMA Tuning.

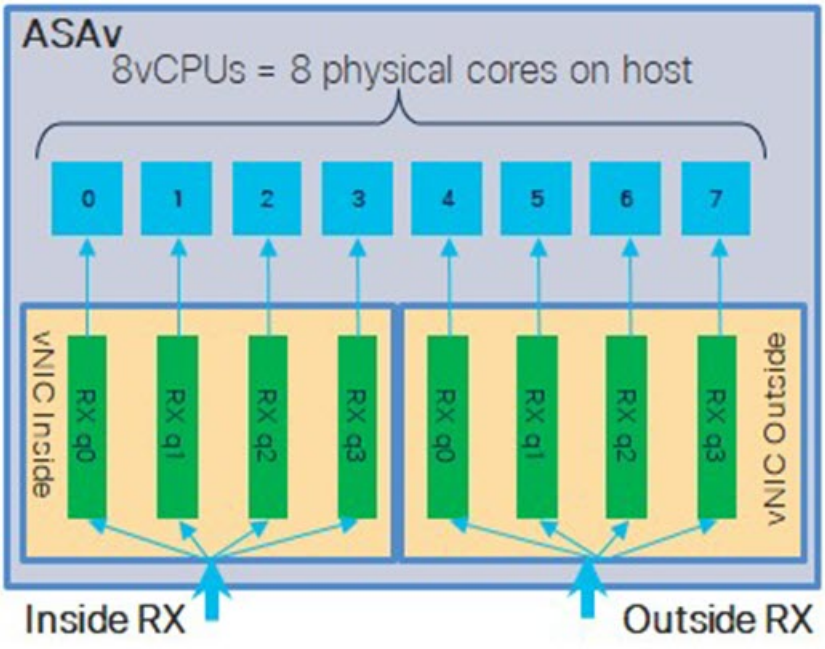
Multiple RX Queues for Receive Side Scaling (RSS)

The ASAv supports Receive Side Scaling (RSS), which is a technology utilized by network adapters to distribute network receive traffic in parallel to multiple processor cores. For maximum throughput, each vCPU (core) must have its own NIC RX queue. Note that a typical RA VPN deployment might use a single inside/outside pair of interfaces.

Important You need ASAv Version 9.13(1) or greater to use multiple RX queues. For KVM, the *libvirt* version needs to be a minimum of 1.0.6.

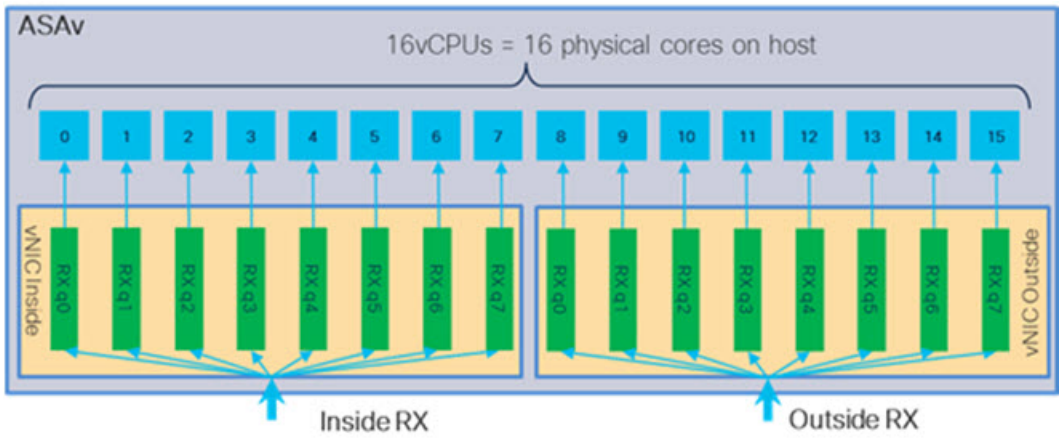
For an 8-core VM with an inside/outside pair of interfaces, each interface will have 4 RX queues, as shown in Figure 9: 8-Core ASAv RSS RX Queues, on page 57.

Figure 9: 8-Core ASAv RSS RX Queues



For a 16-core VM with an inside/outside pair of interfaces, each interface will have 8 RX queues, as shown in Figure 10: 16-Core ASAv RSS RX Queues, on page 57.

Figure 10: 16-Core ASAv RSS RX Queues



The following table presents the ASAv's vNICs for KVM and the number of supported RX queues. See Recommended vNICs, on page 46 for descriptions of the supported vNICs.

Table 12: KVM Recommended NICs/vNICs

NIC Card	vNIC Driver	Driver Technology	Number of RX Queues	Performance
x710	i40e	PCI Passthrough	8 maximum	PCI Passthrough and SR-IOV modes for the x710 offer the best performance. SR-IOV is typically preferred for virtual deployments because the NIC can be shared across multiple VMs.
	i40evf	SR-IOV	8	
x520	ixgbe	PCI Passthrough	6	The x520 NIC performs 10 to 30% lower than the x710. PCI Passthrough and SR-IOV modes for the x520 offer similar performance. SR-IOV is typically preferred for virtual deployments because the NIC can be shared across multiple VMs.
	ixgbe-vf	SR-IOV	2	
N/A	virtio	Para-virtualized	8 maximum	Not recommended for ASAv100. For other deployments, see Enable Multiqueue Support for Virtio on KVM, on page 58.

Enable Multiqueue Support for Virtio on KVM

The following example shows to configure the number of Virtio NIC RX queues to 4 using virsh to edit the libvirt xml:

```
<interface type='bridge'>
 <mac address='52:54:00:43:6e:3f'/>
 <source bridge='clients'/>
 <model type='virtio'/>
 <driver name='vhost' queues='4'/>
 <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
</interface>
```

Important The *libvirt* version needs to be a minimum of 1.0.6 to support multiple RX queues.

VPN Optimization

These are some additional considerations for optimizing VPN performance with the ASAv.

- IPSec has higher throughput than DTLS.
- Cipher - GCM has about 2x the throughput of CBC.

SR-IOV Interface Provisioning

SR-IOV allows multiple VMs to share a single PCIe network adapter inside a host. SR-IOV defines these functions:

- Physical function (PF)—PFs are full PCIe functions that include the SR-IOV capabilities. These appear as regular static NICs on the host server.
- Virtual function (VF)—VFs are lightweight PCIe functions that help in data transfer. A VF is derived from, and managed through, a PF.

VFs are capable of providing up to 10 Gbps connectivity to ASAv machine within a virtualized operating system framework. This section explains how to configure VFs in a KVM environment. SR-IOV support on the ASAv is explained in ASAv and SR-IOV Interface Provisioning, on page 11.

Requirements for SR-IOV Interface Provisioning

If you have a physical NIC that supports SR-IOV, you can attach SR-IOV-enabled VFs, or Virtual NICs (vNICs), to the ASAv instance. SR-IOV also requires support in the BIOS as well as in the operating system instance or hypervisor that is running on the hardware. The following is a list of general guidelines for SR-IOV interface provisioning for the ASAv running in a KVM environment:

- You need an SR-IOV-capable physical NIC in the host server; see Guidelines and Limitations for SR-IOV Interfaces, on page 12.
- You need virtualization enabled in the BIOS on your host server. See your vendor documentation for details.
- You need IOMMU global support for SR-IOV enabled in the BIOS on your host server. See your hardware vendor documentation for details.

Modify the KVM Host BIOS and Host OS

This section shows various setup and configuration steps for provisioning SR-IOV interfaces on a KVM system. The information in this section was created from devices in a specific lab environment, using Ubuntu 14.04 on a Cisco UCS C Series server with an Intel Ethernet Server Adapter X520 - DA2.

Before you begin

- Make sure you have an SR-IOV-compatible network interface card (NIC) installed.
- Make sure that the Intel Virtualization Technology (VT-x) and VT-d features are enabled.

Note Some system manufacturers disable these extensions by default. We recommend that you verify the process with the vendor documentation because different systems have different methods to access and change BIOS settings.

- Make sure all Linux KVM modules, libraries, user tools, and utilities have been installed during the operation system installation; see Prerequisites, on page 48.
- Make sure that the physical interface is in the UP state. Verify with ifconfig <*ethname*>.

Step 1 Log in to your system using the "root" user account and password.

Step 2 Verify that Intel VT-d is enabled.

Example:

```
kvmuser@kvm-host:/$ dmesg | grep -e DMAR -e IOMMU
[    0.000000] ACPI: DMAR 0x000000006F9A4C68 000140 (v01 Cisco0 CiscoUCS 00000001 INTL 20091013)
[    0.000000] DMAR: IOMMU enabled
```

The last line indicates that VT-d is enabled.

Step 3 Activate Intel VT-d in the kernel by appending the *intel_iommu=on* parameter to the GRUB_CMDLINE_LINUX entry in the */etc/default/grub* configuration file.

Example:

```
# vi /etc/default/grub
...
GRUB_CMDLINE_LINUX="nofb splash=quiet console=tty0 ... intel_iommu=on"
...
```

Note If you are using an AMD processor, append *amd_iommu=on* to the boot parameters instead.

Step 4 Reboot the server for the iommu change to take effect.

Example:

```
> shutdown -r now
```

Step 5 Create VFs by writing an appropriate value to the *sriov_numvfs* parameter via the *sysfs* interface using the following format:

```
#echo n > /sys/class/net/device name/device/sriov_numvfs
```

To ensure that the desired number of VFs are created each time the server is power-cycled, you append the above command to the *rc.local* file, which is located in the */etc/rc.d/* directory. The Linux OS executes the *rc.local* script at the end of the boot process.

For example, the following shows the creation of one VF per port. The interfaces for your particular setup will vary.

Example:

```
echo '1' > /sys/class/net/eth4/device/sriov_numvfs
echo '1' > /sys/class/net/eth5/device/sriov_numvfs
echo '1' > /sys/class/net/eth6/device/sriov_numvfs
echo '1' > /sys/class/net/eth7/device/sriov_numvfs
```

Step 6 Reboot the server.

Example:

```
> shutdown -r now
```

Step 7 Verify that the VFs have been created using *lspci*.

Example:

```
 > lspci | grep -i "Virtual Function"
kvmuser@kvm-racetrack:~$ lspci | grep -i "Virtual Function"
0a:10.0 Ethernet controller: Intel Corporation 82599 Ethernet Controller Virtual Function (rev 01)
0a:10.1 Ethernet controller: Intel Corporation 82599 Ethernet Controller Virtual Function (rev 01)
0a:10.2 Ethernet controller: Intel Corporation 82599 Ethernet Controller Virtual Function (rev 01)
0a:10.3 Ethernet controller: Intel Corporation 82599 Ethernet Controller Virtual Function (rev 01)
```

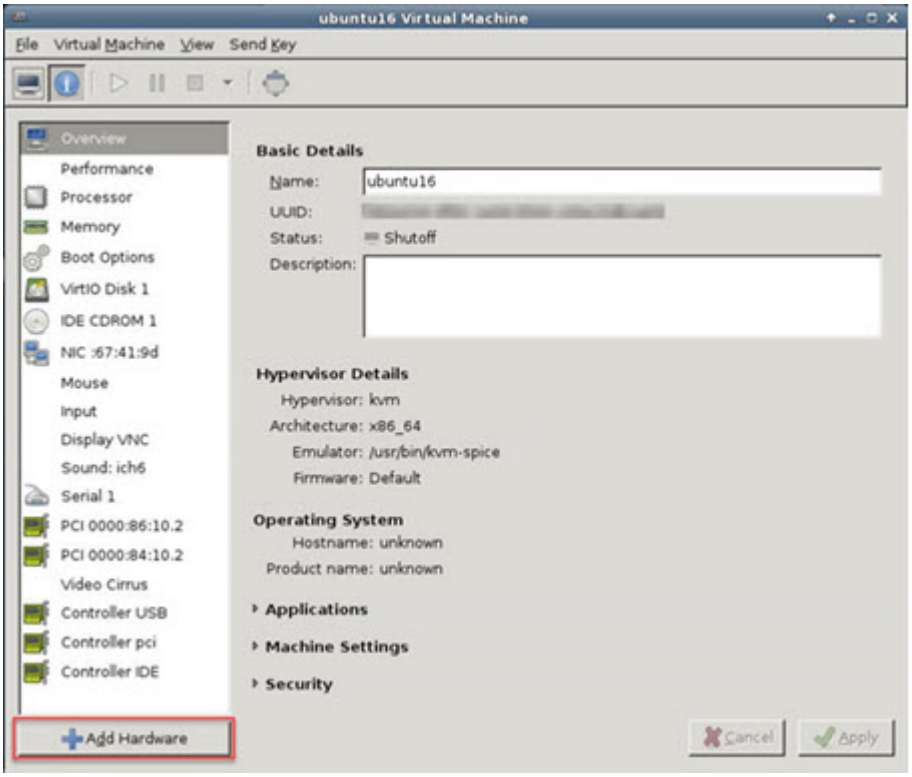
Note You will see additional interfaces using the **ifconfig** command.

Assign PCI Devices to the ASAv

Once you create VFs, you can add them to the ASAv just as you would add any PCI device. The following example explains how to add an Ethernet VF controller to an ASAv using the graphical **virt-manager** tool.

Step 1 Open the ASAv click the **Add Hardware** button to add a new device to the virtual machine.

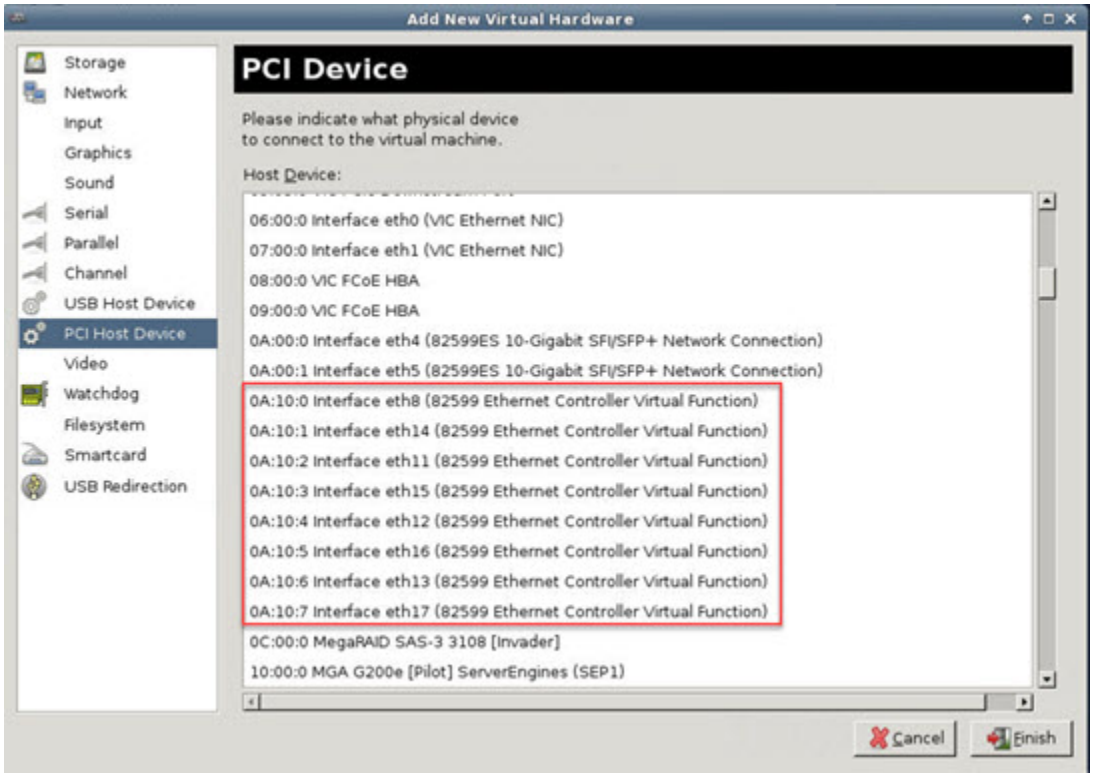
Figure 11: Add Hardware



Step 2 Click **PCI Host Device** from the **Hardware** list in the left pane.

The list of PCI devices, including VFs, appears in the center pane.

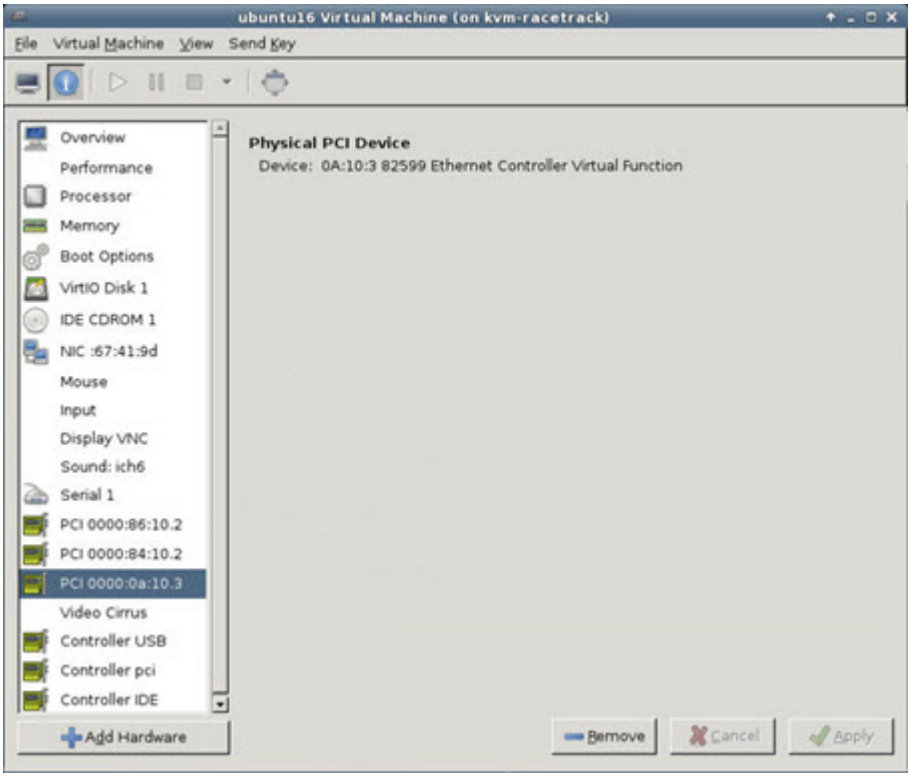
Figure 12: List of Virtual Functions



Step 3 Select one of the available Virtual Functions and click **Finish**.

The PCI Device shows up in the Hardware List; note the description of the device as Ethernet Controller Virtual Function.

Figure 13: Virtual Function added



What to do next

- Use the **show interface** command from the ASAv command line to verify newly configured interfaces.
- Use the interface configuration mode on the ASAv to configure and enable the interface for transmitting and receiving traffic; see the *Basic Interface Configuration* chapter of the Cisco ASA Series General Operations CLI Configuration Guide for more information.

CPU Usage and Reporting

The CPU Utilization report summarizes the percentage of the CPU used within the time specified. Typically, the Core operates on approximately 30 to 40 percent of total CPU capacity during nonpeak hours and approximately 60 to 70 percent capacity during peak hours.

Important Beginning with 9.13(1), any ASA Virtual license now can be used on any supported ASA Virtual vCPU/memory configuration. This allows ASA Virtual customers to run on a wide variety of VM resource footprints.

vCPU Usage in the ASA Virtual

The ASA virtual vCPU usage shows the amount of vCPUs used for the data path, control point, and external processes.

The vSphere reported vCPU usage includes the ASA virtual usage as described plus:

- ASA virtual idle time
- %SYS overhead used for the ASA virtual machine
- Overhead of moving packets between vSwitches, vNICs, and pNICs. This overhead can be quite significant.

CPU Usage Example

The **show cpu usage** command can be used to display CPU utilization statistics.

Example

```
Ciscoasa#show cpu usage

CPU utilization for 5 seconds = 1%; 1 minute: 2%; 5 minutes: 1%
```

The following is an example in which the reported vCPU usage is substantially different:

- ASA Virtual reports: 40%
- DP: 35%
- External Processes: 5%
- ASA (as ASA Virtual reports): 40%
- ASA idle polling: 10%
- Overhead: 45%

The overhead is used to perform hypervisor functions and to move packets between NICs and vNICs using the vSwitch.

KVM CPU Usage Reporting

The

```
virsh cpu-stats domain --total start count
```

command provides the CPU statistical information on the specified guest virtual machine. By default, it shows the statistics for all CPUs, as well as a total. The `--total` option will only display the total statistics. The `--count` option will only display statistics for *count* CPUs.

Tools like OProfile, top etc. give the total CPU usage of a particular KVM VM which includes the CPU usage of both the hypervisor as well as VM. Similarly, tools like XenMon which are specific to Xen VMM gives total CPU usage of Xen hypervisor i.e Dom 0 but don’t separate it into hypervisor usage per VM.

Apart from this, certain tools exist in cloud computing frameworks like OpenNebula which only provides coarse grained information of percentage of Virtual CPU used by a VM.

ASA Virtual and KVM Graphs

There are differences in the CPU % numbers between the ASA Virtual and KVM:

- The KVM graph numbers are always higher than the ASA Virtual numbers.
- KVM calls it %CPU usage; the ASA Virtual calls it %CPU utilization.

The terms “%CPU utilization” and “%CPU usage” mean different things:

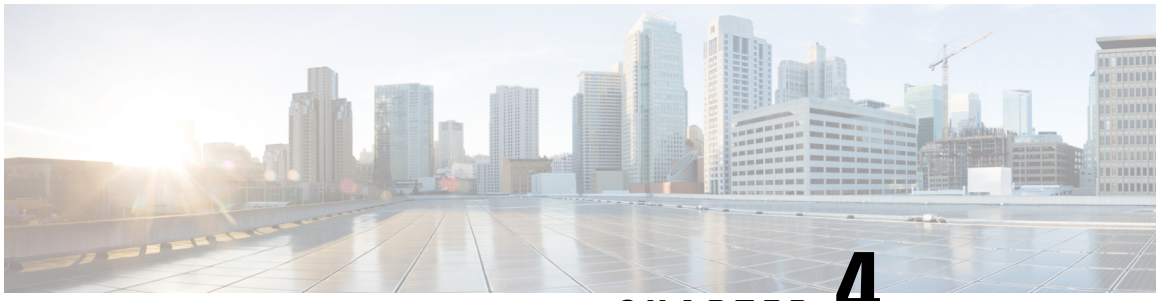
- CPU utilization provides statistics for physical CPUs.
- CPU usage provides statistics for logical CPUs, which is based on CPU hyperthreading. But because only one vCPU is used, hyperthreading is not turned on.

KVM calculates the CPU % usage as follows:

Amount of actively used virtual CPUs, specified as a percentage of the total available CPUs

This calculation is the host view of the CPU usage, not the guest operating system view, and is the average CPU utilization over all available virtual CPUs in the virtual machine.

For example, if a virtual machine with one virtual CPU is running on a host that has four physical CPUs and the CPU usage is 100%, the virtual machine is using one physical CPU completely. The virtual CPU usage calculation is Usage in MHz / number of virtual CPUs x core frequency



CHAPTER 4

Deploy the ASAv On the AWS Cloud

You can deploy the ASAv on the Amazon Web Services (AWS) cloud.

Important Beginning with 9.13(1), any ASAv license now can be used on any supported ASAv vCPU/memory configuration. This allows the ASAv customers to run on a wide variety of VM resource footprints. This also increases the number of supported AWS instances types.

Overview

The ASAv runs the same software as physical ASAs to deliver proven security functionality in a virtual form factor. The ASAv can be deployed in the public AWS cloud. It can then be configured to protect virtual and physical data center workloads that expand, contract, or shift their location over time.

The ASAv support the following AWS instance types.

Table 13: AWS Supported Instance Types

Instance	Attributes		Maximum Number of Interfaces
	vCPUs	Memory (GB)	
c3.large	2	3.75	3
c3.xlarge	4	7.5	4
c3.2xlarge	8	15	4
c4.large	2	3.75	3

Instance	Attributes		Maximum Number of Interfaces
	vCPUs	Memory (GB)	
c4.xlarge	4	7.5	4
c4.2xlarge	8	15	4
c5.large	2	4	3
c5.xlarge	4	8	4
c5.2xlarge	8	16	4
c5.4xlarge	16	32	8
c5a.large	2	4	3
c5a.xlarge	4	8	4
c5a.2xlarge	8	16	4
c5a.4xlarge	16	32	8
c5ad.large	2	4	3
c5ad.xlarge	4	8	4
c5ad.2xlarge	8	16	4
c5ad.4xlarge	16	32	8
c5d.large	2	4	3
c5d.xlarge	4	8	4
c5d.2xlarge	8	16	4
c5d.4xlarge	16	32	8
c5n.large	2	5.3	3
c5n.xlarge	4	10.5	4
c5n.2xlarge	8	21	4
c5n.4xlarge	16	42	8
m4.large	2	8	2
m4.xlarge	4	16	4
m4.2xlarge	8	32	4
m5n.large	2	8	3
m5n.xlarge	4	16	4

Instance	Attributes		Maximum Number of Interfaces
	vCPUs	Memory (GB)	
m5n.2xlarge	8	32	4
m5n.4xlarge	16	64	8
m5zn.large	2	8	3
m5zn.xlarge	4	16	4
m5zn.2xlarge	8	32	4

Tip If you are using M4 or C4 instance type, then we recommend that you migrate to M5 or C5 instance type that uses Nitro hypervisor and Elastic Network Adapter (ENA) interface drivers for improved performance.

Tip If you are using C4 instance type, then we recommend that you migrate to C5 instance type that uses Nitro hypervisor and Elastic Network Adapter (ENA) interface drivers for improved performance.

Table 14: ASAv Licensed Feature Limits Based on Entitlement

Performance Tier	Instance Type (Core/RAM)	Rate Limit	RA VPN Session Limit
ASAv5	c5.large 2 core/4 GB	100 Mbps	50
ASAv10	c5.large 2 core/4 GB	1 Gbps	250
ASAv30	c5.xlarge 4 core/8 GB	2 Gbps	750
ASAv50	c5.2xlarge 8 core/16 GB	10 Gbps	10,000
ASAv100	c5n.4xlarge 16 core/42 GB	16 Gbps	20,000

You create an account on AWS, set up the ASAv using the AWS Wizard, and chose an Amazon Machine Image (AMI). The AMI is a template that contains the software configuration needed to launch your instance.

Important The AMI images are not available for download outside of the AWS environment.

Prerequisites

- Create an account on aws.amazon.com.
- License the ASAv. Until you license the ASAv, it will run in degraded mode, which allows only 100 connections and throughput of 100 Kbps. See Licensing for the ASAv, on page 1.
- Interface requirements:
  - Management interface
  - Inside and outside interfaces
  - (Optional) Additional subnet (DMZ)
- Communications paths:
  - Management interface—Used to connect the ASAv to the ASDM; can't be used for through traffic.
  - Inside interface (required)—Used to connect the ASAv to inside hosts.
  - Outside interface (required)—Used to connect the ASAv to the public network.
  - DMZ interface (optional)—Used to connect the ASAv to the DMZ network when using the c3.xlarge interface.
- For ASAv system requirements, see Cisco ASA Compatibility.

Guidelines and Limitations

Supported Features

The ASAv on AWS supports the following features:

- Support for Amazon EC2 C5 instances, the next generation of the Amazon EC2 Compute Optimized instance family.
- Deployment in the Virtual Private Cloud (VPC)
- Enhanced networking (SR-IOV) where available
- Deployment from Amazon Marketplace
- User deployment of L3 networks
- Routed mode (default)
- Amazon CloudWatch

Unsupported Features

The ASAv on AWS does not support the following:

- Console access (management is performed using SSH or ASDM over network interfaces)

- VLAN
- Promiscuous mode (no sniffing or transparent mode firewall support)
- Multiple context mode
- Clustering
- ASAv native HA
- EtherChannel is only supported on direct physical interfaces
- VM import/export
- Hypervisor agnostic packaging
- VMware ESXi
- Broadcast/multicast messages

  These messages are not propagated within AWS so routing protocols that require broadcast/multicast do not function as expected in AWS. VXLAN can operate only with static peers.
- Gratuitous/unsolicited ARPs

  These ARPS are not accepted within AWS so NAT configurations that require gratuitous ARPs or unsolicited ARPs do not function as expected.
- IPv6

Configuration Migration and SSH Authentication

Upgrade impact when using SSH public key authentication—Due to updates to SSH authentication, additional configuration is required to enable SSH public key authentication; as a result, existing SSH configurations using public key authentication no longer work after upgrading. Public key authentication is the default for the ASAv on Amazon Web Services (AWS), so AWS users will see this issue. To avoid loss of SSH connectivity, you can update your configuration before you upgrade. Or you can use ASDM after you upgrade (if you enabled ASDM access) to fix the configuration.

The following is a sample original configuration for a username "admin":

```
username admin nopassword privilege 15
username admin attributes
  ssh authentication publickey 55:06:47:eb:13:75:fc:5c:a8:c1:2c:bb:
  07:80:3a:fc:d9:08:a9:1f:34:76:31:ed:ab:bd:3a:9e:03:14:1e:1b hashed
```

To use the **ssh authentication** command, before you upgrade, enter the following commands:

```
aaa authentication ssh console LOCAL
username admin password <password> privilege 15
```

We recommend setting a password for the username as opposed to keeping the **nopassword** keyword, if present. The **nopassword** keyword means that any password can be entered, not that no password can be entered. Prior to 9.6(2), the **aaa** command was not required for SSH public key authentication, so the **nopassword** keyword was not triggered. Now that the **aaa** command is required, it automatically also allows regular password authentication for a **username** if the **password** (or **nopassword**) keyword is present.

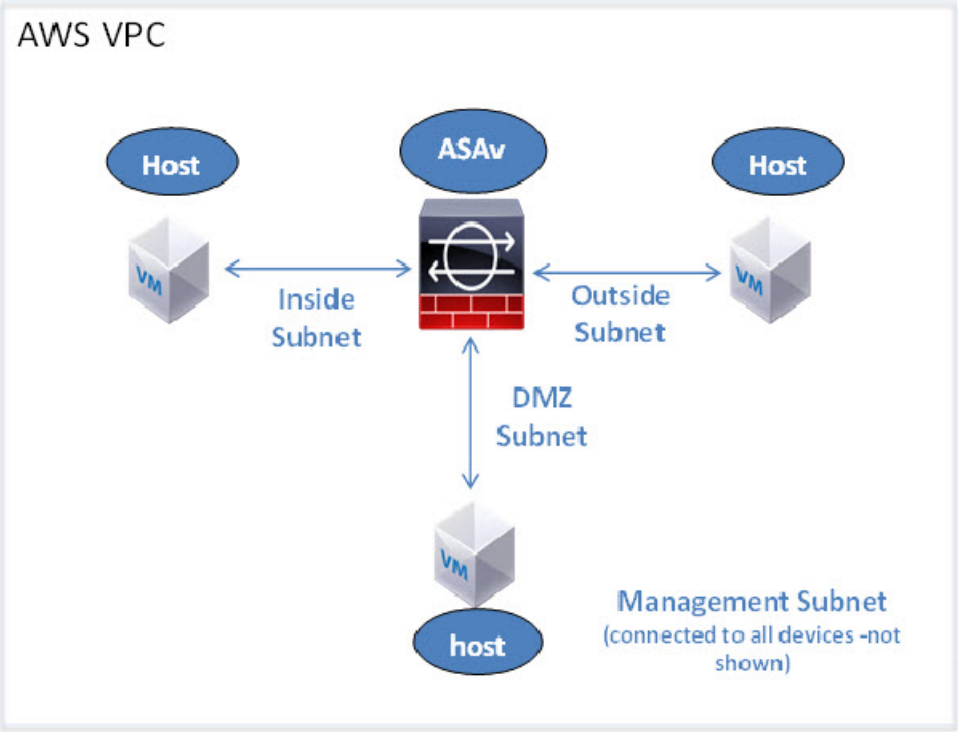
After you upgrade, the **username** command no longer requires the **password** or **nopassword** keyword; you can require that a user cannot enter a password. Therefore, to force public key authentication only, re-enter the **username** command:

```
username admin privilege 15
```

Sample Network Topology

The following figure shows the recommended topology for the ASAv in Routed Firewall Mode with four subnets configured in AWS for the ASAv (management, inside, outside, and DMZ).

Figure 14: Sample ASAv on AWS Deployment



Deploy the ASAv

The following procedure is a top-level list of steps to set up AWS on the ASAv. For detailed steps for setup, see Getting Started with AWS.

Step 1 Log into aws.amazon.com and choose your region.

Note AWS is divided into multiple regions that are isolated from each other. The region is displayed in the upper right corner of your screen. Resources in one region do not appear in another region. Check periodically to make sure you are in the intended region.

Step 2 Click **My Account** > **AWS Management Console**, and under Networking, click **VPC** > **Start VPC Wizard**, and create your VPC by choosing a single public subnet, and set up the following (you can use the default settings unless otherwise noted):

- Inside and outside subnet—Enter a name for the VPC and the subnets.
- Internet Gateway—Enables direct connectivity over the Internet (enter the name of the Internet gateway).
- Outside table—Add entry to enable outbound traffic to the Internet (add 0.0.0.0/0 to Internet Gateway).

Step 3 Click **My Account** > **AWS Management Console** > **EC2**, and then click **Create an Instance**.

- Select your AMI (for example Ubuntu Server 14.04 LTS).

  Use the AMI identified in the your image delivery notification.
- Choose the instance type supported by the ASAv (for example, c3.large).
- Configure the instance (CPUs and memory are fixed).
- Expand the **Advanced Details** section and in the **User data** field you can optionally enter a Day 0 configuration, which is text input that contains the ASAv configuration applied when the ASAv is launched. For more information on how to configure the Day 0 configuration with more information, such as Smart Licensing, see Prepare the Day 0 Configuration File.
  - **Management interface** - If you choose to provide a Day 0 configuration, you **must** provide management interface details, which should be configured to use DHCP.
  - **Data interfaces** - IP addresses for the data interfaces will be assigned and configured only if you provide that information as part of the Day 0 configuration. Data interfaces can be configured to use DHCP or, if the network interfaces to be attached are already created and the IP addresses are known, you can provide the IP details in the Day 0 configuration.
  - **Without Day 0 Configuration** - If you deploy the ASAv **without** providing the Day 0 configuration, the ASAv applies the default ASAv configuration where it fetches the IPs of the attached interfaces from the AWS metadata server and allocates the IP addresses (the data interfaces will get the IPs assigned but the ENIs will be down). Management0/0 interface will be up and gets the IP configured with DHCP address. See IP Addressing in your VPC for information about Amazon EC2 and Amazon VPC IP addressing.
- **Sample Day 0 Configuration** -

```
! ASA Version 9.x.1.200
!
interface management0/0
management-only
nameif management
security-level 100
ip address dhcp setroute

no shutdown
!
crypto key generate rsa modulus 2048
ssh 0 0 management
ssh ::/0 management
ssh timeout 60
ssh version 2
username admin password Q1w2e3r4 privilege 15
username admin attributes
service-type admin
```

```
aaa authentication ssh console LOCAL
!
same-security-traffic permit inter-interface
same-security-traffic permit intra-interface
access-list allow-all extended permit ip any any
access-list allow-all extended permit ip any6 any6
access-group allow-all global
!
interface G0/0
nameif outside
ip address dhcp setroute

no shutdown
!
interface G0/1
nameif inside
ip address dhcp

no shutdown
!
```

- Storage (accept the defaults).
- Tag Instance—You can create a lot of tags to classify your devices. Give it a name you can use to find it easily.
- Security Group—Create a security group and name it. The security group is a virtual firewall for an instance to control inbound and outbound traffic.

  By default the Security Group is open to all addresses. Change the rules to only allow SSH in from addresses used to access your ASAv.

  For information on how the security group controls the traffic, refer to AWS documentation - Control traffic to your AWS resources using security groups.
- Review your configuration and then click **Launch**.

Step 4 Create a Key Pair.

Caution Give the key pair a name you will recognize and download the key to a safe place; the key can never be downloaded again. If you lose the key pair, you must destroy your instances and redeploy them again.

Step 5 Click **Launch Instance** to deploy your ASAv.

Step 6 Click **My Account** > **AWS Management Console** > **EC2** > **Launch an Instance** > **My AMIs**.

Step 7 Make sure that the Source/Destination Check is disabled per interface for the ASAv.

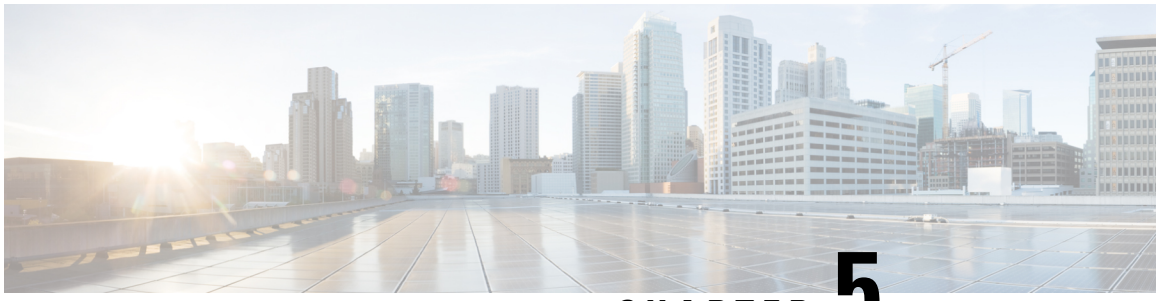
AWS default settings only allow an instance to receive traffic for its IP address (IPv4) and only allow an instance to send traffic from its own IP address (IPv4) . To enable the ASAv to act as a routed hop, you must disable the Source/Destination Check on each of the ASAv's traffic interfaces (inside, outside, and DMZ).

Performance Tuning

VPN Optimization

The AWS c5 instances offer much higher performance than the older c3, c4, and m4 instances. The approximate RA VPN throughput (DTLS using 450B TCP traffic with AES-CBC encryption) on the c5 instance family should be:

- 0.5Gbps on c5.large
- 1Gbps on c5.xlarge
- 2Gbps on c5.2xlarge



CHAPTER 5

Deploy the ASAv Auto Scale Solution on AWS

Auto Scale Solution for the FTDv ASAv on AWS

The following sections describe how the components of the auto scale solution work for the ASAv on AWS.

Overview

Cisco provides CloudFormation Templates and scripts for deploying an auto-scaling group of ASAv firewalls using several AWS services, including Lambda, auto scaling groups, Elastic Load Balancing (ELB), Amazon S3 Buckets, SNS, and CloudWatch.

The ASAv auto scale in AWS is a complete serverless implementation (i.e. no helper VMs involved in the automation of this feature) that adds horizontal auto scaling capability to ASAv instances in the AWS environment. Starting from version 6.4, the auto scale solution is supported on managed by FMC.

The ASAv auto scale solution is a CloudFormation template-based deployment that provides:

- Completely automated configuration automatically applied to scaled-out ASAv instances.
- Support for Load Balancers and multi-availability zones.
- Support for enabling and disabling the auto scale feature.

Auto Scale Use Case

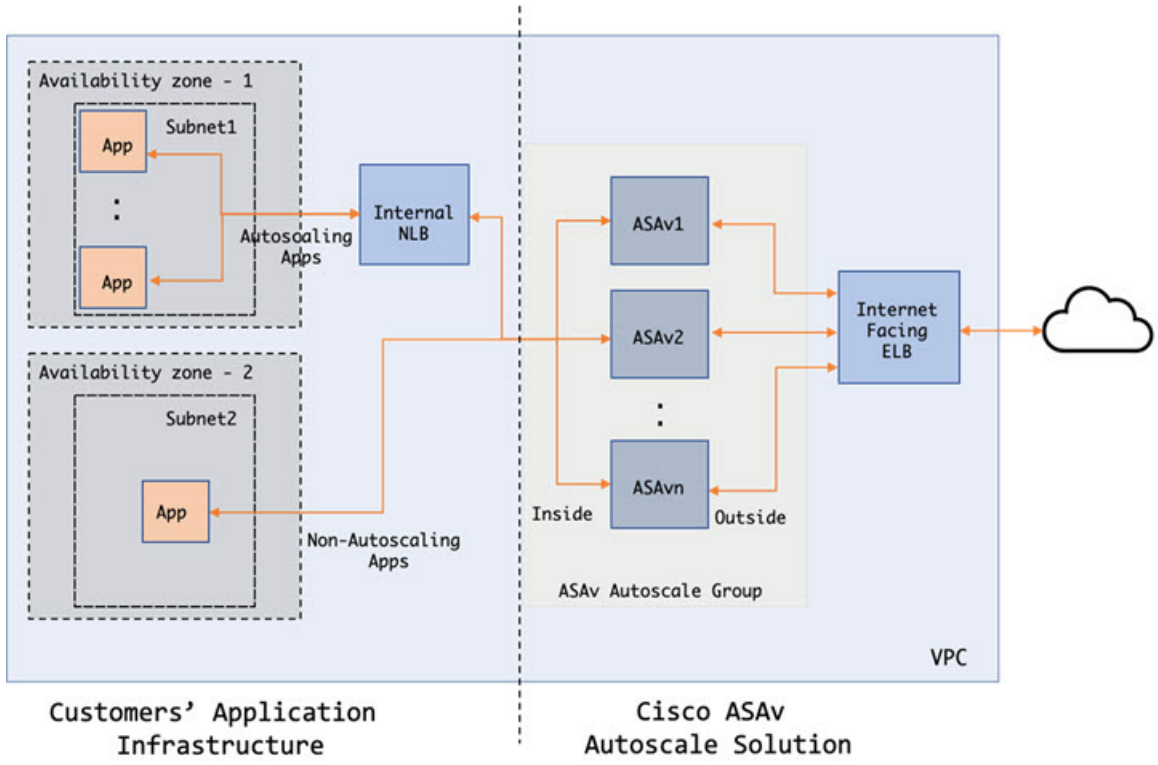
The Use Case for this ASAv AWS auto scale Solution is shown in the use case diagram. Because the AWS Load Balancer allows only Inbound-initiated connections, only externally generated traffic is allowed to pass inside via the ASAv firewall.

Note Secured ports need an SSL/TLS certificate, as described SSL Server Certificate, on page 82 in the Prerequisites.

The Internet-facing load balancer can be a Network Load Balancer or an Application Load Balancer. All of the AWS requirements and conditions hold true for either case. As indicated in the Use Case diagram, the right side of the dotted line is deployed via the ASAv templates. The left side is completely user-defined.

Note Application-initiated outbound traffic will not go through the ASAv.

Figure 15: ASAv Auto Scale Use Case Diagram



Port-based bifurcation for traffic is possible. This can be achieved via NAT rules. For example, traffic on Internet-facing LB DNS, Port: 80 can be routed to Application-1; Port: 88 traffic can be routed to Application-2.

How the Auto Scale Solution Works

To scale the ASAv instances in and out, an external entity called the Auto Scale Manager monitors metrics, commands an auto scale group to add or delete the ASAv instances, and configures the ASAv instances.

The Auto Scale Manager is implemented using AWS Serverless architecture and communicates with AWS resources and the ASAv. We provide CloudFormation templates to automate the deployment of Auto Scale Manager components. The template also deploys other resources required for complete solution to work.

Note Serverless auto scale scripts are only invoked by CloudWatch events, hence they only run when an instance is launched.

Auto Scale Solution Components

The following components make up the auto scale solution.

CloudFormation Template

The CloudFormation template is used to deploy resources required by auto scale solution in AWS. The template consists of:

- Auto Scale Group, Load Balancer, Security Groups, and other miscellaneous components.
- The template takes user input to customize the deployment.

Note The template has limitations in validating user input, hence it is the user's responsibility to validate input during deployment.

Lambda Functions

The auto scale solution is a set of Lambda functions developed in Python, which gets triggered from Lifecycle hooks, SNS, CloudWatch event/alarm events. The basic functionality includes:

- Add/Remove Gig0/0, and Gig 0/1 interfaces to instance.
- Register Gig0/1 interface to Load Balancer's Target Groups.
- Configure and deploy a new ASAv with the ASA configuration file.

Lambda Functions are delivered to customer in the form of a Python package.

Lifecycle Hooks

- Lifecycle hooks are used to get lifecycle change notification about an instance.
- In the case of instance launch, a Lifecycle hook is used to trigger a Lambda function which can add interfaces to an ASAv instance, and register outside interface IPs to target groups.
- In the case of instance termination, a Lifecycle hook is used to trigger a Lambda function to deregister an ASAv instance from the target group.

Simple Notification Service (SNS)

- Simple Notification Service (SNS) from AWS is used to generate events.
- Due to the limitation that there is no suitable orchestrator for Serverless Lambda functions in AWS, the solution uses SNS as a kind of function chaining to orchestrate Lambda functions based on events.

Prerequisites

Download Deployment Files

Download the files required to launch the ASAv auto scale for AWS solution. Deployment scripts and templates for your ASA version are available in the GitHub repository.

Attention Note that Cisco-provided deployment scripts and templates for auto scale are provided as open source examples, and are not covered within the regular Cisco TAC support scope. Check GitHub regularly for updates and ReadMe instructions.

Infrastructure Configuration

In a cloned/downloaded GitHub repository, the **infrastructure.yaml** file can be found in the template folder. This CFT can be used to deploy VPCs, subnets, routes, ACLs, security groups, VPC end-points, and S3 buckets with bucket policies. This CFT can be modified to fit your requirements.

The following sections provide more information about these resources and their use in auto scale. You can manually deploy these resources and also use them in auto scale.

Note The **infrastructure.yaml** template deploys VPCs, subnets, ACLs, security groups, S3 buckets, and VPC end-points only. It does not create the SSL certificate, Lambda layer, or KMS key resources.

VPC

You should create the VPC as required for your application requirements. It is expected that the VPC have an Internet gateway with at least one subnet attached with a route to the Internet. Refer to the appropriate sections for the requirements for Security Groups, Subnets, etc.

Subnets

Subnets can be created as needed for the requirements of the application. The ASAv machine requires 3 subnets for operation as shown in the Use Case.

Note If multiple availability zone support is needed, then subnets are needed for each zone as subnets are zonal properties within the AWS Cloud

Outside Subnet

The Outside subnet should have a default route with '0.0.0.0/0' to the Internet gateway. This will contain the Outside interface of the ASAv, and also the Internet-facing NLB will be in this subnet.

Inside Subnet

This can be similar to the Application subnets, with or without NAT/Internet gateway. Please note that for the ASAv health probes, it should be possible to reach the AWS Metadata Server (169.254.169.254) via port 80.

Note In this AutoScale solution, Load Balancer health probes are redirected to the AWS Metadata Server via inside/Gig0/0 interface. However, you can change this with your own application serving the health probe connections sent to the ASAv from the Load Balancer. In that case, you need to replace the AWS Metadata Server object to the respective application IP address to provide the health probes response.

Management Subnet

This subnet includes the ASAv Management interface. It's optional for you to have a default route.

Lambda Subnets

The AWS Lambda function requires two subnets having the NAT gateway as the default gateway. This makes the Lambda function private to the VPC. Lambda subnets do not need to be as wide as other subnets. Please refer to AWS documentation for best practices on Lambda subnets.

Application Subnets

There is no restriction imposed on this subnet from the auto scale solution, but in case the application needs Outbound connections outside the VPC, there should be respective routes configured on the subnet. This is because outbound-initiated traffic does not pass through Load Balancers. See the AWS Elastic Load Balancing User Guide.

Security Groups

All connections are allowed in the provided Auto Scale Group template. You need only the following connections for the auto scale Solution to work.

Table 15: Required Ports

Port	Usage	Subnet
Health Probe port (default: 8080)	Internet-facing Load Balancer health probes	Outside, Inside Subnets
Application ports	Application data traffic	Outside, Inside Subnets

Amazon S3 Bucket

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. You can place all the required files for both the firewall template and the application template in the S3 bucket.

When templates are deployed, Lambda functions get created referencing Zip files in the S3 bucket. Hence the S3 bucket should be accessible to the user account.

SSL Server Certificate

If the Internet-facing Load Balancer has to support TLS/SSL, a Certificate ARN is required. Refer to the following links for more information:

- Working with Server Certificates
- Create a Private Key and Self-Signed Certificate for Testing
- Create AWS ELB with Self-Signed SSL Cert (Third-party link)

Example of ARN: arn:aws:iam::[AWS Account]:server-certificate/[Certificate Name]

Lambda Layer

The ***autoscale_layer.zip*** can be created in a Linux environment, such as Ubuntu 18.04 with Python 3.9 installed.

```
#!/bin/bash
mkdir -p layer
virtualenv -p /usr/bin/python3.9 ./layer/
source ./layer/bin/activate
pip3 install cffi==1.15.1
pip3 install cryptography==2.9.1
pip3 install paramiko==2.7.1
pip3 install requests==2.23.0
pip3 install scp==0.13.2
pip3 install jsonschema==3.2.0
pip3 install pycryptodome==3.15.0
echo "Copy from ./layer directory to ./python\n"
cp -r ./layer/lib/python3.9/site-packages/* ./python/
zip -r autoscale_layer.zip ./python
```

The resultant ***autoscale_layer.zip*** file should be copied to the *lambda-python-files* folder.

KMS Master Key

This is required if the ASAv passwords are in encrypted format. Otherwise this component is not required. Passwords should be encrypted using only the KMS provided here. If KMS ARN is entered on CFT, then passwords have to be encrypted. Otherwise passwords should be plain text.

For more information about master keys and encryption, see the AWS document Creating keys and the AWS CLI Command Reference about password encryption and KMS.

Example:

```
$ aws kms encrypt --key-id <KMS-ARN> --plaintext 'MyC0mplIc@tedProtect1oN'
{
    "KeyId": "KMS-ARN",
    "CiphertextBlob":
"AQICAHgcQFAGtz/hvaxMtJvY/x/rfHnKI3clFPpSXUU7HQRnCAFwfXhXHJAHL8tcVmDqurALAAAAajBoBgkqhki
G9w0BBwagWzBZAgEAMFQGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQM45AIkTqjSekX2mniAgEQgCcOav6Hhol
+wxpWKtXY4y1Z1d0z1P4fx0jTdosfCbPnUExmNJ4zdx8="
}
$
```

The value of *CiphertextBlob* key should be used as a password.

Python 3 Environment

A ***make.py*** file can be found in the cloned repository top directory. This will Zip the python files into a Zip file and copy to a target folder. In order to do these tasks, the Python 3 environment should be available.

Deploy the Auto Scale Solution

Preparation

It is expected that the Application is either deployed or its deployment plan is available.

Input Parameters

The following input parameters should be collected prior to deployment.

Note For AWS Gateway Load Balancer (GWLB), the **LoadBalancerType**, **LoadBalancerSG**, **LoadBalancerPort**, and **SSLcertificate** parameters are not applicable.

Table 16: Auto Scale Input Parameters

Parameter	Allowed Values/Type	Description
PodNumber	String Allowed Pattern: '^\d{1,3}$'	This is the pod number. This will be suffixed to the Auto Scale Group name (ASAv-Group-Name). For example, if this value is '1', then the group name will be *ASAv-Group-Name-1*. It should be at least 1 numerical digit but not more than 3 digits. Default: 1
AutoscaleGrpNamePrefix	String	This is the Auto Scale Group Name Prefix. The pod number will be added as a suffix. Maximum: 18 characters Example: Cisco-ASAv-1
NotifyEmailID	String	Auto Scale events will be sent to this email address. You need to accept a subscription email request. Example: admin@company.com

Parameter	Allowed Values/Type	Description
VpcId	String	The VPC ID in which the device needs to be deployed. This should be configured as per AWS requirements. Type: AWS::EC2::VPC::Id If the "*infrastructure.yaml*" file is used to deploy the infrastructure, the output section of the stack will have this value. Please use that value.
LambdaSubnets	List	The subnets where Lambda functions will be deployed. Type: List<AWS::EC2::Subnet::Id> If the "*infrastructure.yaml*" file is used to deploy the infrastructure, the output section of the stack will have this value. Please use that value.
LambdaSG	List	The Security Groups for Lambda functions. Type: List<AWS::EC2::SecurityGroup::Id> If the "*infrastructure.yaml*" file is used to deploy the infrastructure, the output section of the stack will have this value. Please use that value.
S3BktName	String	The S3 bucket name for files. This should be configured in your account as per AWS requirements. If the "*infrastructure.yaml*" file is used to deploy the infrastructure, the output section of the stack will have this value. Please use that value.
LoadBalancerType	String	The type of Internet-facing Load Balancer, either “application” or “network”. Example: application
LoadBalancerSG	String	The Security Groups for the Load Balancer. In the case of a network load balancer, it won't be used. But you should provide a Security Group ID. Type: List<AWS::EC2::SecurityGroup::Id> If the "*infrastructure.yaml*" file is used to deploy the infrastructure, the output section of the stack will have this value. Please use that value.
LoadBalancerPort	Integer	The Load Balancer port. This port will be opened on LB with either HTTP/HTTPS or TCP/TLS as the protocol, based on the chosen Load Balancer type. Make sure the port is a valid TCP port, it will be used to create the Load Balancer listener. Default: 80

Parameter	Allowed Values/Type	Description
SSLcertificate	String	The ARN for the SSL certificate for secured port connections. If not specified, a port opened on the Load Balancer will be TCP/HTTP. If specified, a port opened on the Load Balancer will be TLS/HTTPS.
TgHealthPort	Integer	This port is used by the Target group for health probes. Health probes arriving at this port on the ASAv will be routed to the AWS Metadata server and should not be used for traffic. It should be a valid TCP port. If you want your application itself to reply to health probes, then accordingly NAT rules can be changed for the ASAv. In such a case, if the application does not respond, the ASAv will be marked as unhealthy and deleted due to the Unhealthy instance threshold alarm. Example: 8080
AssignPublicIP	Boolean	If selected as "true" then a public IP will be assigned. In case of a BYOL-type ASAv, this is required to connect to https://tools.cisco.com. Example: TRUE
ASAvInstanceType	String	The Amazon Machine Image (AMI) supports different instance types, which determine the size of the instance and the required amount of memory. Only AMI instance types that support the ASAv should be used. Example: c4.2xlarge
ASAvLicenseType	String	The ASAv license type, either BYOL or PAYG. Make sure the related AMI ID is of the same licensing type. Example: BYOL
ASAvAmiId	String	The ASAv AMI ID (a valid Cisco ASAv AMI ID). Type: AWS::EC2::Image::Id Please choose the correct AMI ID as per the region and desired version of the image.

Parameter	Allowed Values/Type	Description
ConfigFileURL	String	The HTTP URL for the ASAv configuration files. Configuration files for each AZs should be available in the URL. The Lambda function will take care of choosing the correct file. You can deploy an HTTP server to host configuration files, or you can use the AWS S3 static web-hosting facility. **Note** The last "/" is also needed as configuration file names will be appeneded to the URL at the time of import. If the "*infrastructure.yaml*" file is used to deploy the infrastructure, the output section of the stack will have this value. Please use that value. Example: https://myserver/asavconfig/asaconfig .txt/
NoOfAZs	Integer	The number of availability zones that the ASAv should span across, between 1 and 3. In the case of an ALB deployment, the minimum value is 2, as required by AWS. Example: 2
ListOfAzs	Comma separated string	A comma-separated list of zones in order. **Note** The order in which these are listed matters. Subnet lists should be given in the same order. If the "*infrastructure.yaml*" file is used to deploy the infrastructure, the output section of the stack will have this value. Please use that value. Example: us-east-1a, us-east-1b, us-east-1c
ASAvMgmtSubnetId	Comma separated list	A comma-separated list of management subnet-ids. The list should be in the same order as the corresponding availability zones. Type: List<AWS::EC2::SecurityGroup::Id> If the "*infrastructure.yaml*" file is used to deploy the infrastructure, the output section of the stack will have this value. Please use that value.

Parameter	Allowed Values/Type	Description
ASAvInsideSubnetId	Comma separated list	A comma-separated list of inside/Gig0/0 subnet-ids. The list should be in the same order as the corresponding availability zones. Type: List<AWS::EC2::SecurityGroup::Id> If the "*infrastructure.yaml*" file is used to deploy the infrastructure, the output section of the stack will have this value. Please use that value.
ASAvOutsideSubnetId	Comma separated list	A comma-separated list of outside/Gig0/1 subnet-ids. The list should be in the same order as the corresponding availability zones. Type: List<AWS::EC2::SecurityGroup::Id> If the "*infrastructure.yaml*" file is used to deploy the infrastructure, the output section of the stack will have this value. Please use that value.
KmsArn	String	The ARN of an existing KMS (AWS KMS key to encrypt at rest). If specified, the ASAv passwords should be encrypted. The password encryption should be done using only the specified ARN. Generating Encrypted Password Example: " aws kms encrypt --key-id <KMS ARN> --plaintext <password> ". Please used such generated passwords as shown. Example: arn:aws:kms:us-east-1:[AWS Account]:key/7d586a25-5875-43b1-bb68-a452e2f6468e
CpuThresholds	Comma separated integers	The lower CPU threshold and the upper CPU threshold. The minimum value is 0 and maximum value is 99. Defaults: 10, 70 Please note that the lower threshold should be less than the upper threshold. Example: 30,70

Update the ASA Configuration Files

You prepare ASA configuration files and store them in a http/https server accessible by an ASAv instance. This is a standard ASA configuration file format. A scaled-out ASAv will download a configuration file and update its configuration.

The following sections provide examples on how the ASA configuration file could be modified for the Auto Scale solution.

Objects, Device Groups, NAT Rules, and Access Policies

See the following for an example of objects, route, and NAT rules for the Load Balancer health probes for the ASAv configuration.

```
! Load Balancer Health probe Configuration
object network aws-metadata-server
host 169.254.169.254
object service aws-health-port
service tcp destination eq 7777
object service aws-metadata-http-port
service tcp destination eq 80
route inside 169.254.169.254 255.255.255.255 10.0.100.1 1
nat (outside,inside) source static any interface destination static interface
aws-metadata-server service aws-health-port aws-metadata-http-port
!
```

Note The health probe connections above should be allowed on your access policy.

See the following for an example of the data-plane configuration for an ASAv configuration.

```
! Data Plane Configuration
route inside 10.0.0.0 255.255.0.0 10.0.100.1 1
object network http-server-80
host 10.0.50.40
object network file-server-8000
host 10.0.51.27
object service http-server-80-port
service tcp destination eq 80
nat (outside,inside) source static any interface destination static interface http-server-80
 service http-server-80-port http-server-80-port
object service file-server-8000-port
service tcp destination eq 8000
nat (outside,inside) source static any interface destination static interface file-server-8000
 service file-server-8000-port file-server-8000-port
object service https-server-443-port
service tcp destination eq 443
nat (outside,inside) source static any interface destination static interface http-server-80
 service https-server-443-port http-server-80-port
!
```

Configuration File Updates

The ASAv configuration should be updated in the *az1-connfiguration.txt*, *az2-configuration.txt*, and *az3-configuration.txt* files.

Note Having three configuration files allows you to modify the configuration based on the Availability Zone (AZ). For example, the static route to the aws-metadata-server will have a different gateway in each AZ.

Template Updates

The *deploy_autoscale.yaml* template should be modified carefully. You should modify the *UserData* field of the ***LaunchTemplate***. The *UserData* can be updated as needed. The *name-server* should be updated accordingly;

for example, it can be the VPC DNS IP. Where your licensing is BYOL, the licensing *idtoken* should be shared here.

```
!
dns domain-lookup management
DNS server-group DefaultDNS
name-server <VPC DNS IP>
!
! License configuration
        call-home
        profile License
        destination transport-method http
        destination address http <url>
        license smart
        feature tier standard
        throughput level <entitlement>
        license smart register idtoken <token>
```

Upload Files to Amazon Simple Storage Service (S3)

All the files in the *target* directory should be uploaded to the Amazon S3 bucket. Optionally, you can use the CLI to upload all of the files in the *target* directory to the Amazon S3 bucket.

```
$ cd ./target
$ aws s3 cp . s3://<bucket-name> --recursive
```

Deploy Stack

After all of the prerequisites are completed for deployment, you can create the AWS CloudFormation stack.

Use the *deploy_autoscale.yaml* file in the *target* directory.

Use the *deploy_ngfw_autoscale_with_gwlb.yaml* file in the *target* directory for Geneve Autoscale.

Note Before you deploy *deploy_ngfw_autoscale_with_gwlb.yaml* file, you must deploy **infrastructure_gwlb.yaml** file for AWS GWLB auto scale solution.

You must create the Gateway Loadbalancer Endpoint (GWLB-E) by choosing the GWLB that is created during *deploy_autoscale_with_gwlb.yaml* template deployment. After creating the GWLBe, you must update the default route to use GWLBe for Application Subnet and default Route table.

For more information, see https://docs.amazonaws.cn/en_us/vpc/latest/privatelink/create-endpoint-service-gwlbe.html.

Provide the parameters as collected in Input Parameters, on page 83.

Validate Deployments

Once the template deployment is successful, you should validate that the Lambda functions and the CloudWatch events are created. By default, the Auto Scale Group has the minimum and maximum number of instances as zero. You should edit the Auto Scale group in the AWS EC2 console with how many instances you want. This will trigger the new ASAv instances.

We recommend that you launch only one instance and check its workflow and validate its behavior as to whether it is working as expected. Post that actual requirements of the ASAv can be deployed, they can also be verified for the behavior. The minimum number of ASAv instances can be marked as Scale-In protected to avoid their removal by AWS Scaling policies.

Maintenance Tasks

Scaling Processes

This topic explains how to suspend and then resume one or more of the scaling processes for your Auto Scale group.

Start and Stop Scale Actions

To start and stop scale out/in actions, follow these steps.

- For AWS Dynamic Scaling—Refer to the following link for information to enable or disable scale out actions:

  Suspending and Resuming Scaling Processes

Health Monitor

Every 60 minutes, a CloudWatch Cron job triggers the Auto Scale Manager Lambda for the Health Doctor module:

- If there are unhealthy IPs which belong to a valid ASAv VM, that instance gets deleted if the ASAv is more than an hour old.
- If those IPs are not from a valid ASAv machine, then only IPs are removed from the Target Group.

Disable Health Monitor

To disable a health monitor, in ***constant.py*** make the constant as “True”.

Enable Health Monitor

To enable a health monitor, in ***constant.py*** make the constant as “False”.

Disable Lifecycle Hooks

In the unlikely event that Lifecycle hook needs to be disabled, if disabled it won’t add additional interfaces to Instances. It can also cause a series of failed deployment of the ASAv instances.

Disable Auto Scale Manager

To disable Auto Scale Manager, respective CloudWatch Events “notify-instance-launch” and “notify-instance-terminate” should be disabled. Disabling this won’t trigger Lambda for any new events. But

already executing Lambda actions will continue. There is no abrupt stop of Auto Scale Manager. Trying abrupt stopping by stack deletion or deleting resources can cause an indefinite state.

Load Balancer Targets

Because the AWS Load Balancer does not allow instance-type targets for instances having more than one network interface, the Gigabit0/1 interface IP is configured as a target on Target Groups. As of now however, the AWS Auto Scale health checks work only for instance-type targets, not IPs. Also, these IPs are not automatically added or removed from target groups. Hence our Auto Scale solution programmatically handles both of these tasks. But in the case of maintenance or troubleshooting, there could be a situation demanding manual effort to do so.

Register a Target to a Target Group

To register the ASAv instance to the Load Balancer, its Gigabit0/1 instance IP (outside subnet) should be added as a target in Target Group(s). See Register or Deregister Targets by IP Address.

Deregister a Target from a Target Group

To deregister the ASAv instance to the Load Balancer, its Gigabit0/1 instance IP (outside subnet) should be deleted as a target in Target Group(s). See Register or Deregister Targets by IP Address.

Instance Stand-by

AWS does not allow instance reboot in the Auto Scale group, but it does allow a user to put an instance in Stand-by and perform such actions. However, this works best when the Load Balancer targets are instance-type. However, the ASAv machines cannot be configured as instance-type targets, because of multiple network interfaces.

Put an Instance in Stand-by

If an instance is put into stand-by, its IP in Target Groups will still continue to be in the same state until the health probes fail. Because of this, it is recommended to deregister respective IPs from the Target Group before putting the instance into stand-by state; see Deregister a Target from a Target Group, on page 91 for more information.

Once the IPs are removed, see Temporarily Removing Instances from Your Auto Scaling Group.

Remove an Instance from Stand-by

Similarly you can move an instance from stand-by to running state. After removal from stand-by state, the instance's IP should be registered to Target Group targets. See Register a Target to a Target Group, on page 91.

For more information about how to put instances into stand-by state for troubleshooting or maintenance, see the AWS News Blog.

Remove/Detach Instance from Auto Scale Group

To remove an instance from the Auto Scale group, first it should be moved to stand-by state. See "Put Instances on Stand-by". Once the instance is in the stand-by state it can be removed or detached. See Detach EC2 Instances from Your Auto Scaling Group.

Terminate an Instance

To terminate an instance it should be put into stand-by state; see Instance Stand-by, on page 91. Once the instance is in stand-by, you can proceed to terminate.

Instance Scale-In Protection

To avoid an accidental removal of any particular instance from the Auto Scale group, it can be made as Scale-In protected. If an instance is Scale-In protected, it won't be terminated due to a Scale-In event.

Please refer to the following link to put an instance into Scale-In protected state.

https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-instance-termination.html

Important It is recommended to make the minimum number of instances which are healthy (the target IP should be healthy, not just the EC2 instance) as Scale-In protected.

Changes to Configuration

Any changes in configuration won't be automatically reflected on already running instances. Changes will be reflected on upcoming devices only. Any such changes should be manually pushed to already existing devices.

If you are facing issues while manually updating the configuration on existing instances, we recommend removing these instances from the Scaling Group and replacing them with new instances.

Change the ASAv Admin Password

A change to the ASAv password requires the user to change it on each device manually for running instances. For new ASAv devices to be onboarded, the ASAv password will be taken from the Lambda environment variables. See Using AWS Lambda Environment Variables.

Changes to AWS Resources

You can change many things in AWS post deployment, such as the Auto Scale Group, Launch Configuration, CloudWatch events, Scaling Policies etc. You can import your resources into a CloudFormation stack or create a new stack from your existing resources.

See Bringing Existing Resources Into CloudFormation Management for more information about how to manage changes performed on AWS resources.

Collect and Analyze CloudWatch Logs

In order to export CloudWatch logs please refer to Export Log Data to Amazon S3 Using the AWS CLI.

Troubleshooting and Debugging

AWS CloudFormation Console

You can verify the input parameters to your CloudFormation stack in the AWS CloudFormation Console, which allows you to create, monitor, update and delete stacks directly from your web browser.

Navigate to the required stack and check the parameter tab. You can also check inputs to Lambda Functions on the Lambda Functions environment variables tab.

To learn more about the AWS CloudFormation console, see the *AWS CloudFormation User Guide*.

Amazon CloudWatch Logs

You can view logs of individual Lambda functions. AWS Lambda automatically monitors Lambda functions on your behalf, reporting metrics through Amazon CloudWatch. To help you troubleshoot failures in a function, Lambda logs all requests handled by your function and also automatically stores logs generated by your code through Amazon CloudWatch Logs.

You can view logs for Lambda by using the Lambda console, the CloudWatch console, the AWS CLI, or the CloudWatch API. To learn more about log groups and accessing them through the CloudWatch console, see the Monitoring system, application, and custom log files in the *Amazon CloudWatch User Guide*.

Load Balancer Health Check Failure

The load balancer health check contains information such as the protocol, ping port, ping path, response timeout, and health check interval. An instance is considered healthy if it returns a 200 response code within the health check interval.

If the current state of some or all your instances is `OutOfService` and the description field displays the message that the `Instance has failed at least the Unhealthy Threshold number of health checks consecutively`, the instances have failed the load balancer health check.

You should check the health probe NAT rule in the ASA configuration. For more information, see Troubleshoot a Classic Load Balancer: Health checks.

Traffic Issues

To troubleshoot traffic issues with your ASAv instances, you should check the Load Balancer rules, the NAT rules, and the static routes configured in the ASAv instances.

You should also check the AWS virtual network/subnets/gateway details provided in the deployment template, including security group rules, etc. You can also refer to AWS documentation, for example, Troubleshooting EC2 instances.

ASAv Failed to Configure

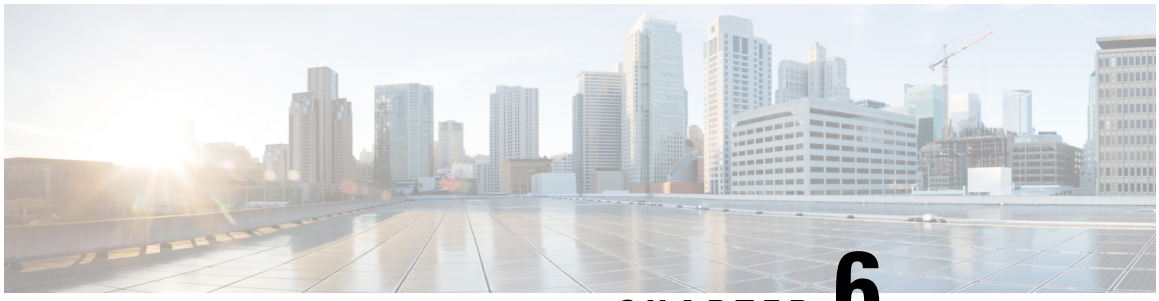
If the ASAv fails to configure, you should check the connectivity to the Amazon S3 static HTTP webserver hosting configuration. See Hosting a static website on Amazon S3 for more information.

ASAv Failed to License

If the ASAv fails to license, you should check the connectivity to the CSSM server, check the ASAv Security Group configuration, and check the Access Control Lists.

Unable to SSH into the ASAv

If you are unable to SSH into the ASAv, check to see if the complex password was passed to the ASAv via the template.



CHAPTER 6

Deploy the ASAv On the Microsoft Azure Cloud

You can deploy the ASAv on the Microsoft Azure cloud.

Important Beginning with 9.13(1), any ASAv license now can be used on any supported ASAv vCPU/memory configuration. This allows the ASAv customers to run on a wide variety of VM resource footprints. This also increases the number of supported Azure instances types.

Overview

Select the Azure virtual machine tier and size to meet your ASAv needs. Any ASAv license can be used on any supported ASAv vCPU/memory configuration. This allows you to run the ASAv on a wide variety Azure instances types.

Table 17: Azure Supported Instance Types

Instance	Attributes		Interfaces
	vCPUs	Memory (GB)	
D3, D3_v2, DS3, DS3_v2	4	14	4
D4, D4_v2, DS4, DS4_v2	8	28	8

Instance	Attributes		Interfaces
	vCPUs	Memory (GB)	
D5, D5_v2, DS5, DS5_v2	16	56	8
D8_v3	8	32	4
D16_v3	16	64	4
D8s_v3	8	32	4
D16s_v3	16	64	8
F4, F4s	4	8	4
F8, F8s	8	16	8
F16, F16s	16	32	8
F8s_v2	8	16	4
F16s_v2	16	32	8

Table 18: ASAv Licensed Feature Limits Based on Entitlement

Performance Tier	Instance Type (Core/RAM)	Rate Limit	RA VPN Session Limit
ASAv5	D3_v2 4 core/14 GB	100 Mbps	50
ASAv10	D3_v2 4 core/14 GB	1 Gbps	250
ASAv30	D3_v2 4 core/14 GB	2 Gbps	750
ASAv50	D4_v2 8 core/28 GB	5.5 Gbps	10,000
ASAv100	D5_v2 16 core/56 GB	11 Gbps	20,000

You can deploy the ASAv on Microsoft Azure:

- As a stand-alone firewall using the Azure Resource Manager on the standard Azure public cloud and the Azure Government environments
- As an integrated partner solution using the Azure Security Center
- As a high availability (HA) pair using the Azure Resource Manager on the standard Azure public cloud and the Azure Government environments

See Deploy the ASAv from Azure Resource Manager, on page 104. Note that you can deploy the ASAv HA configuration on the standard Azure public cloud and the Azure Government environments.

Prerequisites

- Create an account on Azure.com.

  After you create an account on Microsoft Azure, you can log in, choose the ASAv in the Microsoft Azure Marketplace, and deploy the ASAv.

- License the ASAv.

  Until you license the ASAv, it will run in degraded mode, which allows only 100 connections and throughput of 100 Kbps. See Smart Software Licensing for the ASAv.

Note The ASAv defaults to the 2Gbps entitlement when deployed on Azure. The use of the 100Mbps and 1Gbps entitlement is allowed. However, the throughput level must be explicitly configured to use the 100Mbps or 1Gbps entitlement.

- Interface requirements:

  You must deploy the ASAv with four interfaces on four networks. You can assign a public IP address to any interface; see Public IP addresses for Azure's guidelines regarding public IPs, including how to create, change, or delete a public IP address.

- Management interface:

  In Azure, the first defined interface is always the Management interface.

- Communications paths:

  - Management interface—Used for SSH access and to connect the ASAv to the ASDM.

Note Azure accelerated networking is not supported on the Management interface.

  - Inside interface (required)—Used to connect the ASAv to inside hosts.
  - Outside interface (required)—Used to connect the ASAv to the public network.
  - DMZ interface (optional)—Used to connect the ASAv to the DMZ network when using the Standard_D3 interface.

- For ASAv hypervisor and virtual platform support information, see Cisco ASA Compatibility.

Guidelines and Limitations

Supported Features

- Deployment from Microsoft Azure Cloud
- Azure Accelerated Networking (AN)
- Maximum of 16 vCPUs, based on the selected instance type

Note Azure does not provide configurable L2 vSwitch capability.

- Public IP address on any interface

  You can assign a public IP address to any interface; see Public IP addresses for Azure's guidelines regarding public IPs, including how to create, change, or delete a public IP address.

- Routed firewall mode (default)

Note In routed firewall mode the ASAv is a traditional Layer 3 boundary in the network. This mode requires an IP address for each interface. Because Azure does not support VLAN tagged interfaces, the IP addresses must be configured on non-tagged, non-trunk interfaces.

Known Issues

Idle Timeout

The ASAv on Azure has a configurable *idle timeout* on the VM. The minimum setting is 4 minutes and the maximum setting is 30 minutes. However, for SSH sessions the minimum setting is 5 minutes and the maximum setting is 60 minutes.

Note Be aware that the ASAv's idle timeout always overrides the SSH timeout and disconnects the session. You can choose to match the VM's idle timeout to the SSH timeout so that the session does not timeout from either side.

Failover from Primary ASAv to Standby ASAv

When an Azure upgrade occurs on an ASAv HA in Azure deployment, a failover may occur from the primary ASAv to the standby ASAv. An Azure upgrade causes the primary ASAv to enter a pause state. The standby ASAv does not receive any hello packets when the primary ASAv is paused. If the standby ASAv does not receive any hello packets beyond the failover hold time, a failover to the standby ASAv occurs.

There is also the possibility of a failover occurring even if the failover hold time has not been exceeded. Consider a scenario in which the primary ASAv resumes 19 seconds after entering the pause state. The failover hold time is 30 seconds. But, the standby ASAv does not receive hello packets with the right timestamp

because the clock is synchronized every ~2 minutes. This causes a failover from the primary ASAv to the standby ASAv.

Note This feature supports IPv4 only, ASA Virtual HA is not supported for IPv6 configuration.

Unsupported Features

- Console access (management is performed using SSH or ASDM over network interfaces)
- VLAN tagging on user instance interfaces
- Jumbo frames
- Proxy ARP for an IP address that the device does not own from an Azure perspective
- Promiscuous mode (no sniffing or transparent mode firewall support)

Note Azure policy prevents the ASAv from operating in transparent firewall mode because it doesn't allow interfaces to operate in promiscuous mode.

- Multi-context mode
- Clustering
- ASAv native HA.

Note You can deploy ASAv on Azure in a stateless Active/Backup high availability (HA) configuration.

- VM import/export
- By default, FIPS mode is not enabled on the ASAv running in the Azure cloud.

Note If you enable FIPS mode, you must change the Diffie-Helman key exchange group to a stronger key by using the **ssh key-exchange group dh-group14-sha1** command. If you don’t change the Diffie-Helman group, you will no longer be able to SSH to the ASAv, and that is the only way to initially manage the ASAv.

- IPv6
- Gen 2 VM generation on Azure
- Re-sizing the VM after deployment
- Migration or update of the Azure Storage SKU for the OS Disk of the VM from premium to standard SKU and vice versa

Azure DDoS Protection Feature

Azure DDoS Protection in Microsoft Azure is an additional feature implemented at the forefront of ASAv. In a virtual network, when this feature is enabled it helps to defend applications against common network layer attacks depending on the packet per second of a network's expected traffic. You can customize this feature based on the network traffic pattern.

For more information about the Azure DDoS Protection feature, see Azure DDoS Protection Standard overview.

Resources Created During Deployment

When you deploy the ASAv in Azure the following resources are created:

- The ASAv machine
- A resource group (unless you chose an existing resource group)

  The ASAv resource group must be the same resource group used by the Virtual Network and the Storage Account.
- Four NICS named vm name-Nic0, vm name-Nic1, vm name-Nic2, vm name-Nic3

  These NICs map to the ASAv interfaces Management 0/0, GigabitEthernet 0/0, GigabitEthernet 0/1, and GigabitEthernet 0/2 respectively.

Note Based on the requirement, you can create Vnet with IPv4 only .

- A security group named vm name-SSH-SecurityGroup

  The security group will be attached to the VM's Nic0, which maps to ASAv Management 0/0.

  The security group includes rules to allow SSH and UDP ports 500 and UDP 4500 for VPN purposes. You can modify these values after deployment.
- Public IP addresses (named according to the value you chose during deployment)

  You can assign a public IP address (IPv4 only).

  to any interface; see Public IP addresses for Azure's guidelines regarding public IPs, including how to create, change, or delete a public IP address.
- A Virtual Network with four subnets (unless you chose an existing network)
- A Routing Table for each subnet (updated if it already exists)

  The tables are named subnet name-ASAv-RouteTable.

  Each routing table includes routes to the other three subnets with the ASAv IP address as the next hop. You may chose to add a default route if traffic needs to reach other subnets or the Internet.
- A boot diagnostics file in the selected storage account

  The boot diagnostics file will be in Blobs (binary large objects).
- Two files in the selected storage account under Blobs and container VHDs named *vm name*-disk.vhd and *vm name*-<uuid>.status

- A Storage account (unless you chose an existing storage account)

Note When you delete a VM, you must delete each of these resources individually, except for any resources you want to keep.

Azure Routing

Routing in an Azure Virtual Network is determined by the Virtual Network's Effective Routing Table. The Effective Routing Table is a combination of an existing System Routing Table and the User Defined Routing Table.

Note The ASAv cannot use dynamic interior routing protocols like EIGRP and OSPF due to the nature of Azure cloud routing. The Effective Routing Table determines the next hop, regardless of whether a virtual client has any static/dynamic route configured.

Currently you cannot view either the Effective Routing Table or the System Routing Table.

You can view and edit the User Defined Routing table. When the System table and the User Defined tables are combined to form the Effective Routing Table, the most specific route wins and ties go to the User Defined Routing table. The System Routing Table includes a default route (0.0.0.0/0) pointing to Azure's Virtual Network Internet Gateway. The System Routing Table also includes specific routes to the other defined subnets with the next-hop pointing to Azure's Virtual Network infrastructure gateway.

To route traffic through the ASAv, the ASAv deployment process adds routes on each subnet to the other three subnets using the ASAv as the next hop. You may also want to add a default route (0.0.0.0/0) that points to the ASAv interface on the subnet. This will send all traffic from the subnet through the ASAv, which may require that ASAv policies be configured in advance to handle that traffic (perhaps using NAT/PAT).

Because of the existing specific routes in the System Routing Table, you must add specific routes to the User Defined Routing table to point to the ASAv as the next-hop. Otherwise, a default route in the User Defined table would lose to the more specific route in the System Routing Table and traffic would bypass the ASAv.

Routing Configuration for VMs in the Virtual Network

Routing in the Azure Virtual Network depends on the Effective Routing Table and not the particular gateway settings on the clients. Clients running in a Virtual Network may be given routes by DHCP that are the .1 address on their respective subnets. This is a place holder and serves only to get the packet to the Virtual Network's infrastructure virtual gateway. Once a packet leaves the VM it is routed according to the Effective Routing Table (as modified by the User Defined Table). The Effective Routing Table determines the next hop regardless of whether a client has a gateway configured as .1 or as the ASAv address.

Azure VM ARP tables will show the same MAC address (1234.5678.9abc) for all known hosts. This ensures that all packets leaving an Azure VM will reach the Azure gateway where the Effective Routing Table will be used to determine the path of the packet.

Note The ASAv cannot use dynamic interior routing protocols like EIGRP and OSPF due to the nature of Azure cloud routing. The Effective Routing Table determines the next hop, regardless of whether a virtual client has any static/dynamic route configured.

IP Addresses

The following information applies to IP addresses in Azure:

- You should use DHCP to set the IP addresses of ASAv interfaces.

  The Azure infrastructure ensures that the ASAv interfaces are assigned the IP addresses set in Azure.

- Management 0/0 is given a private IP address in the subnet to which it is attached.

  A public IP address may be associated with this private IP address and the Azure Internet gateway will handle the NAT translations.

- You can assign a public IP address to any interface.
- Public IP addresses that are dynamic may change during an Azure stop/start cycle. However, they are persistent during Azure restart and during ASAv reload.
- Public IP addresses that are static won't change until you change them in Azure.

DNS

All Azure virtual networks have access to a built-in DNS server at 168.63.129.16 that you can use as follows:

```
configure terminal
dns domain-lookup management
dns server-group DefaultDNS
 name-server 168.63.129.16
end
```

You can use this configuration when you configure Smart Licensing and you don’t have your own DNS Server set up.

Accelerated Networking (AN)

Azure's Accelerated Networking (AN) feature enables single root I/O virtualization (SR-IOV) to a VM, which accelerates networking by allowing VM NICs to bypass the hypervisor and go directly to the PCIe card underneath. AN significantly enhances the throughput performance of the VM and also scales with additional cores (i.e. larger VMs).

AN is disabled by default. Azure supports enabling AN on pre-provisioned virtual machines. You simply have to stop VM in Azure and update the network card property to set the *enableAcceleratedNetworking* parameter to true. See the Microsoft documentation Enable accelerated networking on existing VMs. Then restart the VM.

Support for Mellanox Hardware

Microsoft Azure cloud has two types of hardware that support the AN functionality: Mellanox 4 (MLX4) and Mellanox 5 (MLX5). ASAv supports AN for Mellanox hardware for the following instances from Release 9.15:

- D3, D3_v2, DS3, DS3_v2
- D4, D4_v2, DS4, DS4_v2
- D5, D5_v2, DS5, DS5_v2
- D8_v3, D8s_v3
- D16_v3, D16s_v3
- F4, F4s
- F8, F8s, F8s_v2
- F16, F16s, F16s_v2

Note MLX4 (Mellanox 4) is also referred to as connectx3 = cx3, and MLX5 (Mellanox 5) is also referred as connectx4 = cx4.

You cannot specify which NIC Azure uses MLX4 or MLX5 for your VM deployment. Cisco recommends that you upgrade to ASAv 9.15 version or later to use the accelerated networking functionality.

Deploy the ASAv

You can deploy the ASAv on Microsoft Azure.

- Deploy the ASAv as a stand-alone firewall using the Azure Resource Manager on the standard Azure public cloud and the Azure Government environments. See Deploy the ASAv from Azure Resource Manager.
- Deploy the ASAv as an integrated partner solution within Azure using the Azure Security Center. Security-conscious customers are offered the ASAv as a firewall option to protect Azure workloads. Security and health events are monitored from a single integrated dashboard. See Deploy the ASAv from Azure Security Center.
- Deploy an ASAv High Availablity pair using the Azure Resource Manager. To ensure redundancy, you can deploy the ASAv in an Active/Backup high availability (HA) configuration. HA in the public cloud implements a stateless Active/Backup solution that allows for a failure of the active ASAv to trigger an automatic failover of the system to the backup ASAv. See Deploy the ASAv for High Availability from Azure Resource Manager, on page 107.
- Deploy the ASAv or an ASAv High Availablity pair with a custom template using a Managed Image from a VHD (available from cisco.com). Cisco provides a compressed virtual hard disk (VHD) that you can upload to Azure to simplify the process of deploying the ASAv. Using a Managed Image and two JSON files (a Template file and a Parameter File), you can deploy and provision all the resources for the ASAv in a single, coordinated operation. To use the custom template, see Deploy the ASAv from Azure Using a VHD and Resource Template, on page 108.

Deploy the ASAv from Azure Resource Manager

The following procedure is a top-level list of steps to set up Microsoft Azure on the ASAv. For detailed steps for Azure setup, see Getting Started with Azure.

When you deploy the ASAv in Azure it automatically generates various configurations, such as resources, public IP addresses, and route tables. You can further manage these configurations after deployment. For example, you may want to change the Idle Timeout value from the default, which is a low timeout.

Step 1 Log into the Azure Resource Manager (ARM) portal.

The Azure portal shows virtual elements associated with the current account and subscription regardless of data center location.

Step 2 Search Marketplace for Cisco ASAv, and then click on the ASAv you would like to deploy.

Step 3 Configure the basic settings.

a) Enter a name for the virtual machine. This name should be unique within your Azure subscription.

Important If your name is not unique and you reuse an existing name, the deployment will fail.

b) Enter your username.

c) Choose an authentication type, either **Password** or **SSH public key**.

If you choose **Password**, enter a password and confirm.

d) Choose your subscription type.

e) Choose a **Resource group**.

The resource group should be the same as the virtual network's resource group.

f) Choose your location.

The location should be the same as for your network and resource group.

g) Click **OK**.

Step 4 Configure the ASAv settings.

a) Choose the virtual machine size.

b) Choose a storage account.

You can use an existing storage account or create a new one. The location of the storage account should be the same as for the network and virtual machine.

c) Request a public IP address by entering a label for the IP address in the Name field, and then click **OK**.

Azure creates a dynamic public IP by default, which may change when the VM is stopped and restarted. If you prefer a fixed IP address, you can open the public-ip in the portal and change it from a dynamic to a static address.

d) Add a DNS label if desired.

The fully qualified domain name will be your DNS label plus the Azure URL: *<dnslabel>.<location>.cloupapp.azure.com*

e) Choose an existing virtual network or create a new one.

f) Configure the four subnets that the ASAv will deploy to, and then click **OK**.

Important Each interface must be attached to a unique subnet.

g) Click **OK**.

Step 5 View the configuration summary, and then click **OK**.

Step 6 View the terms of use and then click **Create**.

What to do next

- Continue configuration using CLI commands available for input via SSH or use ASDM. See Start ASDM for instructions for accessing the ASDM.

Deploy the ASAv from Azure Security Center

The Microsoft Azure Security Center is a security solution for Azure that enables customers to protect, detect, and mitigate security risks for their cloud deployments. From the Security Center dashboard, customers can set security policies, monitor security configurations, and view security alerts.

Security Center analyzes the security state of Azure resources to identify potential security vulnerabilities. A list of recommendations guides customers through the process of configuring needed controls, which can include deployment of the ASAv as a firewall solution to Azure customers.

As an integrated solution in Security Center, you can rapidly deploy the ASAv in just a few clicks and then monitor security and health events from a single dashboard. The following procedure is a top-level list of steps to deploy the ASAv from Security Center. For more detailed information, see Azure Security Center.

Step 1 Log into the Azure portal.

The Azure portal shows virtual elements associated with the current account and subscription regardless of data center location.

Step 2 From the Microsoft Azure menu, choose **Security Center**.

If you are accessing Security Center for the first time, the **Welcome** blade opens. Choose **Yes! I want to Launch Azure Security Center** to open the **Security Center** blade and to enable data collection.

Step 3 On the **Security Center** blade, choose the **Policy** tile.

Step 4 On the **Security policy** blade, choose **Prevention policy**.

Step 5 On the **Prevention policy** blade, turn on the recommendations that you want to see as part of your security policy.

a) Set **Next generation firewall** to **On**. This ensures that the ASAv is a recommended solution in Security Center.

b) Set any other recommendations as needed.

Step 6 Return to the **Security Center** blade and the **Recommendations** tile.

Security Center periodically analyzes the security state of your Azure resources. When Security Center identifies potential security vulnerabilities, it shows recommendations on the **Recommendations** blade.

Step 7 Select the **Add a Next Generation Firewall** recommendation on the **Recommendations** blade to view more information and/or to take action to resolve the issue.

Step 8 Choose **Create New** or **Use existing solution**, and then click on the ASAv you would like to deploy.

Step 9 Configure the basic settings.

a) Enter a name for the virtual machine. This name should be unique within your Azure subscription.

Important If your name is not unique and you reuse an existing name, the deployment will fail.

b) Enter your username.
c) Choose an authorization type either password or SSH key.

If you choose password, enter a password and confirm.

d) Choose your subscription type.
e) Choose a resource group.

The resource group should be the same as the virtual network's resource group.

f) Choose your location.

The location should be the same as for your network and resource group.

g) Click **OK**.

Step 10 Configure the ASAv settings.

a) Choose the virtual machine size.

The ASAv supports Standard D3 and Standard D3_v2.

b) Choose a storage account.

You can use an existing storage account or create a new one. The location of the storage account should be the same as for the network and virtual machine.

c) Request a public IP address by entering a label for the IP address in the Name field, and then click **OK**.

Azure creates a dynamic public IP by default, which may change when the VM is stopped and restarted. If you prefer a fixed IP address, you can open the public-ip in the portal and change it from a dynamic to a static address.

d) Add a DNS label if desired.

The fully qualified domain name will be your DNS label plus the Azure URL: *<dnslabel>.<location>.cloupapp.azure.com*

e) Choose an existing virtual network or create a new one.
f) Configure the four subnets that the ASAv will deploy to, and then click **OK**.

Important Each interface must be attached to a unique subnet.

g) Click **OK**.

Step 11 View the configuration summary, and then click **OK**.

Step 12 View the terms of use and then click **Create**.

What to do next

- Continue configuration using CLI commands available for input via SSH or use ASDM. See Start ASDM for instructions for accessing the ASDM.
- If you need more information on how the recommendations in Security Center help you protect your Azure resources, see the documentation available from Security Center.

Deploy the ASAv for High Availability from Azure Resource Manager

The following procedure is a top-level list of steps to set up a High Availability (HA) ASAv pair on Microsoft Azure. For detailed steps for Azure setup, see Getting Started with Azure.

ASAv HA in Azure deploys two ASAvs into an Availability Set, and automatically generates various configurations, such as resources, public IP addresses, and route tables. You can further manage these configurations after deployment.

Step 1 Log into the Azure portal.

The Azure portal shows virtual elements associated with the current account and subscription regardless of data center location.

Step 2 Search Marketplace for **Cisco ASAv**, and then click on the **ASAv 4 NIC HA** to deploy a failover ASAv configuration.

Step 3 Configure the **Basics** settings.

a) Enter a prefix for the ASAv machine names. The ASAv names will be 'prefix'-A and 'prefix'-B.

Important Make sure you do not use an existing prefix or the deployment will fail.

b) Enter a **Username**.

This will be the administrative username for both Virtual Machines.

Important The username **admin** is not allowed in Azure.

c) Choose an authentication type for both Virtual Machines, either **Password** or **SSH public key**.

If you choose **Password**, enter a password and confirm.

d) Choose your subscription type.

e) Choose a **Resource group**.

Choose **Create new** to create a new resource group, or **Use existing** to select an existing resource group. If you use an existing resource group, it must be empty. Otherwise you should create a new resource group.

f) Choose your **Location**.

The location should be the same as for your network and resource group.

g) Click **OK**.

Step 4 Configure the **Cisco ASAv settings**.

a) Choose the Virtual Machine size.

b) Choose **Managed** or **Unmanaged OS disk** storage.

Important ASA HA mode always uses **Managed**.

Step 5 Configure the **ASAv-A** settings.

a) (Optional) Choose **Create new** to request a public IP address by entering a label for the IP address in the Name field, and then click **OK**. Choose **None** if you do not want a public IP address.

Note Azure creates a dynamic public IP by default, which may change when the VM is stopped and restarted. If you prefer a fixed IP address, you can open the public-ip in the portal and change it from a dynamic to a static address.

b) Add a DNS label if desired.

The fully qualified domain name will be your DNS label plus the Azure URL: *<dnslabel>.<location>.cloupapp.azure.com*

c) Configure the required settings for the storage account for the ASAv-A boot diagnostics.

Step 6 Repeat the previous steps for the **ASAv-B** settings.

Step 7 Choose an existing virtual network or create a new one.

a) Configure the four subnets that the ASAv will deploy to, and then click **OK**.

Important Each interface must be attached to a unique subnet.

b) Click **OK**.

Step 8 View the **Summary** configuration, and then click **OK**.

Step 9 View the terms of use and then click **Create**.

What to do next

- Continue configuration using CLI commands available for input via SSH or use ASDM. See Start ASDM for instructions for accessing the ASDM.
- See the 'Failover for High Availability in the Public Cloud' chapter in the ASA Series General Operations Configuration Guide for more information about ASAv HA configuration in Azure.

Deploy the ASAv from Azure Using a VHD and Resource Template

You can create your own custom ASAv images using a compressed VHD image available from Cisco. To deploy using a VHD image, you must upload the VHD image to your Azure storage account. Then, you can create a managed image using the uploaded disk image and an Azure Resource Manager template. Azure templates are JSON files that contain resource descriptions and parameter definitions.

Before you begin

- You need the JSON template and corresponding JSON parameter file for your ASAv template deployment. You can download template files from the GitHub repository at:

  https://github.com/CiscoDevNet/cisco-asav/tree/master/deployment-templates/azure

- For instructions on how to build a template and a parameter file, see Appendix — Azure Resource Template Example, on page 111.
- This procedure requires an existing Linux VM in Azure. We recommended you use a temporary Linux VM (such as Ubuntu 16.04) to upload the compressed VHD image to Azure. This image will require about 50G of storage when unzipped. Also, your upload times to Azure storage will be faster from a Linux VM in Azure.

  If you need to create a VM, use one of the following methods:

  - Create a Linux virtual machine with the Azure CLI
  - Create a Linux virtual machine in the Azure portal

- In your Azure subscription, you should have a storage account available in the Location in which you want to deploy the ASAv.

Step 1 Download the ASAv compressed VHD image from the https://software.cisco.com/download/home page:

a) Navigate to **Products** > **Security** > **Firewalls** > **Adaptive Security Appliances (ASA)** > **Adaptive Security Appliance (ASA) Software**.

b) Click **Adaptive Security Virtual Appliance (ASAv)**.

Follow the instructions for downloading the image.

For example, asav9-14-1.vhd.bz2

Step 2 Copy the compressed VHD image to your Linux VM in Azure.

There are many options that you can use to move files up to Azure and down from Azure. This example shows SCP or secure copy:

```
# scp /username@remotehost.com/dir/asav9-14-1.vhd.bz2 <linux-ip>
```

Step 3 Log in to the Linux VM in Azure and navigate to the directory where you copied the compressed VHD image.

Step 4 Unzip the ASAv VHD image.

There are many options that you can use to unzip or decompress files. This example shows the Bzip2 utility, but there are also Windows-based utilities that would work.

```
# bunzip2 asav9-14-1.vhd.bz2
```

Step 5 Upload the VHD to a container in your Azure storage account. You can use an existing storage account or create a new one. The storage account name can only contain lowercase letters and numbers.

There are many options that you can use to upload a VHD to your storage account, including AzCopy, Azure Storage Copy Blob API, Azure Storage Explorer, Azure CLI, or the Azure Portal. We do not recommend using the Azure Portal for a file as large as the ASAv.

The following example shows the syntax using Azure CLI:

```
azure storage blob upload \
       --file <unzipped vhd> \
       --account-name <azure storage account> \
       --account-key yX7txxxxxxxx1dnQ== \
       --container <container> \
       --blob <desired vhd name in azure> \
       --blobtype page
```

Step 6 Create a Managed Image from the VHD:

a) In the Azure Portal, select **Images**.

b) Click **Add** to create a new image.

c) Provide the following information:

- **Subscription**—Choose a subscription from the drop-down list.
- **Resource group**—Choose an existing resource group or create a new one.
- **Name**—Enter a user-defined name for the managed image.
- **Region**—Choose the region in which the VM Is deployed.
- **OS type**—Choose **Linux** as the OS type.

- **VM generation**—Choose **Gen 1**.

  Note **Gen 2** is not supported.

- **Storage blob**—Browse to the storage account to select the uploaded VHD.
- **Account type**—As per your requirement, choose Standard HDD, Standard SSD, or Premium SSD, from the drop-down list.

  When you select the VM size planned for deployment of this image, ensure that the VM size supports the selected account type.
- **Host caching**—Choose Read/write from the drop-down list.
- **Data disks**—Leave at default; don't add a data disk.

d) Click **Create**.

Wait for the **Successfully created image** message under the **Notifications** tab.

Note Once the Managed Image has been created, the uploaded VHD and upload Storage Account can be removed.

Step 7 Acquire the Resource ID of the newly created Managed Image.

Internally, Azure associates every resource with a Resource ID. You'll need the Resource ID when you deploy new ASAv firewalls from this managed image.

a) In the Azure Portal, select **Images**.
b) Select the managed image created in the previous step.
c) Click **Overview** to view the image properties.
d) Copy the **Resource ID** to the clipboard.

The **Resource ID** takes the form of:

/subscriptions/<subscription-id>/resourceGroups/<resourceGroup>
/providers/Microsoft.Compute/<container>/ <vhdname>

Step 8 Build an ASAv firewall using the managed image and a resource template:

a) Select **New**, and search for **Template Deployment** until you can select it from the options.
b) Select **Create**.
c) Select **Build your own template in the editor**.

You have a blank template that is available for customizing. See Create a Resource Template, on page 112 for an example of how to create a template

d) Paste your customized JSON template code into the window, and then click **Save**.
e) Choose a **Subscription** from the drop-down list.
f) Choose an existing **Resource group** or create a new one.
g) Choose a **Location** from the drop-down list.
h) Paste the Managed Image **Resource ID** from the previous step into the **Vm Managed Image Id** field.

Step 9 Click **Edit parameters** at the top of the **Custom deployment** page. You have a parameters template that is available for customizing.

a) Click **Load file** and browse to the customized ASAv parameter file. See Create a Parameter File, on page 120 for an example of how to create a parameter template.

b) Paste your customized JSON parameters code into the window, and then click **Save**.

Step 10 Review the Custom deployment details. Make sure that the information in **Basics** and **Settings** matches your expected deployment configuration, including the **Resource ID**.

Step 11 Review the Terms and Conditions, and check the **I agree to the terms and conditions stated above** check box.

Step 12 Click **Purchase** to deploy an ASAv firewall using the managed image and a custom template.

If there are no conflicts in your template and parameter files, you should have a successful deployment.

The Managed Image is available for multiple deployments within the same subscription and region.

What to do next

- Continue configuration using CLI commands available for input via SSH or use ASDM. See Start ASDM for instructions for accessing the ASDM.

Appendix — Azure Resource Template Example

This section describes the structure of an Azure Resource Manager template you can use to deploy the ASAv. An Azure Resource Template is a JSON file. To simplify the deployment of all the required resources, this example includes two JSON files:

- **Template File**—This is the main resources file that deploys all the components within the resource group.
- **Parameter File**—This file includes the parameters required to successfully deploy the ASAv. It includes details such as the subnet information, virtual machine tier and size, username and password for the ASAv, the name of the storage container, etc. You can customize this file for your Azure Stack Hub deployment environment.

Template File Format

This section describes the structure of an Azure Resource Manager template file. The following example shows a collapsed view of a template file and presents the different sections of a template.

Azure Resource Manager JSON Template File

```
{
    "$schema":
"http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "",
    "parameters": { },
    "variables": { },
    "resources": [ ],
    "outputs": { }
}
```

The template consists of JSON and expressions that you can use to construct values for your ASAv deployment. In its simplest structure, a template contains the following elements:

Table 19: Azure Resource Manager JSON Template File Elements Defined

Element	Required	Description
$schema	Yes	Location of the JSON schema file that describes the version of the template language. Use the URL shown in the preceding figure.
contentVersion	Yes	Version of the template (such as 1.0.0.0). You can provide any value for this element. When deploying resources using the template, this value can be used to make sure that the right template is being used.
parameters	No	Values that are provided when deployment is executed to customize resource deployment. Parameters allow for inputting values at the time of deployment. They are not absolutely required, but without them the JSON template will deploy the resources with the same parameters each time.
variables	No	Values that are used as JSON fragments in the template to simplify template language expressions.
resources	Yes	Resource types that are deployed or updated in a resource group.
outputs	No	Values that are returned after deployment.

You can make use of JSON templates to not only declare the resource types to be deployed, but also their related configuration parameters. The following example shows a template that deploys a new ASAv.

Create a Resource Template

You can use the example below to create your own deployment template using a text editor.

Step 1 Copy the text in the following example.

Example:

```
{
    "$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "vmName": {
            "type": "string",
            "defaultValue": "ngfw",
            "metadata": {
                "description": "Name of the NGFW VM"
            }
        },
        "vmManagedImageId": {
            "type": "string",
            "defaultValue":
"/subscriptions/{subscription-id}/resourceGroups/myresourcegroup1/providers/Microsoft.Compute/images/myImage",

            "metadata": {
                "description": "The ID of the managed image used for deployment.
```

```
/subscriptions/{subscription-id}/resourceGroups/myresourcegroup1/providers/Microsoft.Compute/images/myImage"

            }
        },
        "adminUsername": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "Username for the Virtual Machine. admin, Administrator among other
values are disallowed - see Azure docs"
            }
        },
        "adminPassword": {
            "type": "securestring",
            "defaultValue" : "",
            "metadata": {
                "description": "Password for the Virtual Machine. Passwords must be 12 to 72 chars
and have at least 3 of the following: Lowercase, uppercase, numbers, special chars"
            }
        },
        "vmStorageAccount": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
              "description": "A storage account name (boot diags require a storage account). Between
 3 and 24 characters. Lowercase letters and numbers only"
            }
        },
        "virtualNetworkResourceGroup": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "Name of the virtual network's Resource Group"
            }
        },
        "virtualNetworkName": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "Name of the virtual network"
            }
        },
        "mgmtSubnetName": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "The FTDv management interface will attach to this subnet"
            }
        },
        "mgmtSubnetIP": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "NGFW IP on the mgmt interface (example: 192.168.0.10)"
            }
        },
        "diagSubnetName": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "The FTDv diagnostic0/0 interface will attach to this subnet"
            }
        },
        "diagSubnetIP": {
```

```
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "NGFW IP on the diag interface (example: 192.168.1.10)"
            }
        },
        "gig00SubnetName": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "The FTDv Gigabit 0/0 interface will attach to this subnet"
            }
        },
        "gig00SubnetIP": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "The IP on the Gigabit 0/0 interface (example: 192.168.2.10)"
            }
        },
        "gig01SubnetName": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "The FTDv Gigabit 0/1 interface will attach to this subnet"
            }
        },
        "gig01SubnetIP": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "The IP on the Gigabit 0/1 interface (example: 192.168.3.5)"
            }
        },
        "VmSize": {
            "type": "string",
            "defaultValue": "Standard_D3_v2",
            "allowedValues": [ "Standard_D3_v2" , "Standard_D3" ],
            "metadata": {
                "description": "NGFW VM Size (Standard_D3_v2 or Standard_D3)"
            }
        }
    },
    "variables": {

        "virtualNetworkID":
"[resourceId(parameters('virtualNetworkResourceGroup'),'Microsoft.Network/virtualNetworks',
parameters('virtualNetworkName'))]",

        "vmNic0Name":"[concat(parameters('vmName'),'-nic0')]",
        "vmNic1Name":"[concat(parameters('vmName'),'-nic1')]",
        "vmNic2Name":"[concat(parameters('vmName'),'-nic2')]",
        "vmNic3Name":"[concat(parameters('vmName'),'-nic3')]",

        "vmNic0NsgName":"[concat(variables('vmNic0Name'),'-NSG')]",

        "vmMgmtPublicIPAddressName": "[concat(parameters('vmName'),'nic0-ip')]",
        "vmMgmtPublicIPAddressType": "Static",
        "vmMgmtPublicIPAddressDnsName": "[variables('vmMgmtPublicIPAddressName')]"
    },
    "resources": [
        {
            "apiVersion": "2017-03-01",
            "type": "Microsoft.Network/publicIPAddresses",
```

```
            "name": "[variables('vmMgmtPublicIPAddressName')]",
            "location": "[resourceGroup().location]",
            "properties": {
              "publicIPAllocationMethod": "[variables('vmMgmtPublicIpAddressType')]",
              "dnsSettings": {
                "domainNameLabel": "[variables('vmMgmtPublicIPAddressDnsName')]"
              }
            }
        },
        {
            "apiVersion": "2015-06-15",
            "type": "Microsoft.Network/networkSecurityGroups",
            "name": "[variables('vmNic0NsgName')]",
            "location": "[resourceGroup().location]",
            "properties": {
                "securityRules": [
                    {
                        "name": "SSH-Rule",
                        "properties": {
                            "description": "Allow SSH",
                            "protocol": "Tcp",
                            "sourcePortRange": "*",
                            "destinationPortRange": "22",
                            "sourceAddressPrefix": "Internet",
                            "destinationAddressPrefix": "*",
                            "access": "Allow",
                            "priority": 100,
                            "direction": "Inbound"
                        }
                    },
                    {
                        "name": "SFtunnel-Rule",
                        "properties": {
                            "description": "Allow tcp 8305",
                            "protocol": "Tcp",
                            "sourcePortRange": "*",
                            "destinationPortRange": "8305",
                            "sourceAddressPrefix": "Internet",
                            "destinationAddressPrefix": "*",
                            "access": "Allow",
                            "priority": 101,
                            "direction": "Inbound"
                        }
                    }
                ]
            }
        },
        {
            "apiVersion": "2017-03-01",
            "type": "Microsoft.Network/networkInterfaces",
            "name": "[variables('vmNic0Name')]",
            "location": "[resourceGroup().location]",
            "dependsOn": [
                "[concat('Microsoft.Network/networkSecurityGroups/',variables('vmNic0NsgName'))]",
                "[concat('Microsoft.Network/publicIPAddresses/',
variables('vmMgmtPublicIPAddressName'))]"
            ],
            "properties": {
                "ipConfigurations": [
                    {
                        "name": "ipconfig1",
                        "properties": {
                            "privateIPAllocationMethod": "Static",
                            "privateIPAddress" : "[parameters('mgmtSubnetIP')]",
```

```
                            "subnet": {
                                "id": "[concat(variables('virtualNetworkID'),'/subnets/',
parameters('mgmtSubnetName'))]"
                            },
                            "publicIPAddress":{
                                "id": "[resourceId('Microsoft.Network/publicIPAddresses/',
variables('vmMgmtPublicIPAddressName'))]"
                            }
                        }
                    }
                ],
                "networkSecurityGroup": {
                    "id": "[resourceId('Microsoft.Network/networkSecurityGroups',
variables('vmNic0NsgName'))]"
                },
                "enableIPForwarding": true
            }
        },
        {
            "apiVersion": "2017-03-01",
            "type": "Microsoft.Network/networkInterfaces",
            "name": "[variables('vmNic1Name')]",
            "location": "[resourceGroup().location]",
            "dependsOn": [
            ],
            "properties": {
                "ipConfigurations": [
                    {
                        "name": "ipconfig1",
                        "properties": {
                            "privateIPAllocationMethod": "Static",
                            "privateIPAddress" : "[parameters('diagSubnetIP')]",
                            "subnet": {
                                "id": "[concat(variables('virtualNetworkID'),'/subnets/',
parameters('diagSubnetName'))]"
                            }                        }
                    }
                ],
                "enableIPForwarding": true
            }
        },
        {
            "apiVersion": "2017-03-01",
            "type": "Microsoft.Network/networkInterfaces",
            "name": "[variables('vmNic2Name')]",
            "location": "[resourceGroup().location]",
            "dependsOn": [
            ],
            "properties": {
                "ipConfigurations": [
                    {
                        "name": "ipconfig1",
                        "properties": {
                            "privateIPAllocationMethod": "Static",
                            "privateIPAddress" : "[parameters('gig00SubnetIP')]",
                            "subnet": {
                                "id": "[concat(variables('virtualNetworkID'),'/subnets/',
parameters('gig00SubnetName'))]"
                            }                        }
                    }
                ],
                "enableIPForwarding": true
            }
        },
```

```
        {
            "apiVersion": "2017-03-01",
            "type": "Microsoft.Network/networkInterfaces",
            "name": "[variables('vmNic3Name')]",
            "location": "[resourceGroup().location]",
            "dependsOn": [
            ],
            "properties": {
                "ipConfigurations": [
                    {
                        "name": "ipconfig1",
                        "properties": {
                            "privateIPAllocationMethod": "Static",
                            "privateIPAddress" : "[parameters('gig01SubnetIP')]",
                            "subnet": {
                                "id": "[concat(variables('virtualNetworkID'),'/subnets/',
parameters('gig01SubnetName'))]"

                            }                        }
                    }
                ],
                "enableIPForwarding": true
            }
        },
        {
            "type": "Microsoft.Storage/storageAccounts",
            "name": "[concat(parameters('vmStorageAccount'))]",
            "apiVersion": "2015-06-15",
            "location": "[resourceGroup().location]",
            "properties": {
              "accountType": "Standard_LRS"
            }
        },
        {
            "apiVersion": "2017-12-01",
            "type": "Microsoft.Compute/virtualMachines",
            "name": "[parameters('vmName')]",
            "location": "[resourceGroup().location]",
            "dependsOn": [
                "[concat('Microsoft.Storage/storageAccounts/', parameters('vmStorageAccount'))]",
                "[concat('Microsoft.Network/networkInterfaces/',variables('vmNic0Name'))]",
                "[concat('Microsoft.Network/networkInterfaces/',variables('vmNic1Name'))]",
                "[concat('Microsoft.Network/networkInterfaces/',variables('vmNic2Name'))]",
                "[concat('Microsoft.Network/networkInterfaces/',variables('vmNic3Name'))]"
            ],
            "properties": {
                "hardwareProfile": {
                    "vmSize": "[parameters('vmSize')]"
                },
                "osProfile": {
                    "computername": "[parameters('vmName')]",
                    "adminUsername": "[parameters('AdminUsername')]",
                    "adminPassword": "[parameters('AdminPassword')]"
                },
                "storageProfile": {
                    "imageReference": {
                        "id": "[parameters('vmManagedImageId')]"
                    },
                    "osDisk": {
                        "osType": "Linux",
                        "caching": "ReadWrite",
                        "createOption": "FromImage"
                    }
                },
```

```
                "networkProfile": {
                    "networkInterfaces": [
                        {
                            "properties": {
                                "primary": true
                            },
                            "id": "[resourceId('Microsoft.Network/networkInterfaces',
variables('vmNic0Name'))]"
                        },
                        {
                            "properties": {
                                "primary": false
                            },
                            "id": "[resourceId('Microsoft.Network/networkInterfaces',
variables('vmNic1Name'))]"
                        },
                        {
                            "properties": {
                                "primary": false
                            },
                            "id": "[resourceId('Microsoft.Network/networkInterfaces',
variables('vmNic2Name'))]"
                        },
                        {
                            "properties": {
                                "primary": false
                            },
                            "id": "[resourceId('Microsoft.Network/networkInterfaces',
variables('vmNic3Name'))]"
                        }
                    ]
                },
                "diagnosticsProfile": {
                    "bootDiagnostics": {
                        "enabled": true,
                        "storageUri":
"[concat('http://',parameters('vmStorageAccount'),'.blob.core.windows.net')]"
                    }
                }
            }
        }
    ],
    "outputs": {  }
}
```

Step 2 Save the file locally as a JSON file; for example, **azureDeploy.json**.

Step 3 Edit the file to create a template to suit your deployment parameters.

Step 4 Use this template to deploy the ASAv as described in Deploy the ASAv from Azure Using a VHD and Resource Template, on page 108.

Parameter File Format

When you start a new deployment, you have parameters defined in your resource template. These need to be entered before the deployment can start. You can manually enter the parameters that you have defined in your resource template, or you can put the parameters in a template parameters JSON file.

The parameter file contains a value for each parameter shown in the parameters example in Create a Parameter File, on page 120. These values are automatically passed to the template during deployment. You can create multiple parameter files for different deployment scenarios.

For the ASAv template in this example, the parameter file must have the following parameters defined:

Table 20: ASAv Parameter Definitions

Field	Description	Example
vmName	The name the ASAv machine will have in Azure.	cisco-asav
vmManagedImageId	The ID of the managed image used for deployment. Internally, Azure associates every resource with a Resource ID.	/subscriptions/73d2537e-ca44-46aa-beb2-74ff1dd61b41/ resourceGroups/ew ManagedImages-rg/providers/Microsoft .Compute/ images/ASAv910-Managed-I mage
adminUsername	The username for logging into the ASAv. This cannot be the reserved name ‘admin’.	jdoe
adminPassword	The admin password. This must be 12 to 72 characters long, and include three of the following: 1 lower case, 1 upper case, 1 number, 1 special character.	Pw0987654321
vmStorageAccount	Your Azure storage account. You can use an existing storage account or create a new one. The storage account name must be between 3 and 24 characters, and can only contain lowercase letters and numbers.	ciscoasavstorage
virtualNetworkResourceGroup	The name of the virtual network's Resource Group. The ASAv is always deployed into a new Resource Group.	ew-west8-rg
virtualNetworkName	The name of the virtual network.	ew-west8-vnet
mgmtSubnetName	The management interface will attach to this subnet. This maps to Nic0, the first subnet. Note, this must match an existing subnet name if joining an existing network.	mgmt
mgmtSubnetIP	The Management interface IP address.	10.8.0.55

Field	Description	Example
gig00SubnetName	The GigabitEthernet 0/0 interface will attach to this subnet. This maps to Nic1, the second subnet. Note, this must match an existing subnet name if joining an existing network.	inside
gig00SubnetIP	The GigabitEthernet 0/0 interface IP address. This is for the ASAv's first data interface.	10.8.2.55
gig01SubnetName	The GigabitEthernet 0/1 interface will attach to this subnet. This maps to Nic2, the third subnet. Note, this must match an existing subnet name if joining an existing network.	outside
gig01SubnetIP	The GigabitEthernet 0/1 interface IP address. This is for ASAv's second data interface.	10.8.3.55
gig02SubnetName	The GigabitEthernet 0/2 interface will attach to this subnet. This maps to Nic3, the fourth subnet. Note, this must match an existing subnet name if joining an existing network.	dmz
gig02SubnetIP	The GigabitEthernet 0/2 interface IP address. This is for ASAv's third data interface.	10.8.4.55
vmSize	The VM size to use for the ASAv VM. Standard_D3_V2 and Standard_D3 are supported. Standard_D3_V2 is the default.	Standard_D3_V2 or Standard_D3

Create a Parameter File

You can use the example below to create your own parameter file using a text editor.

Note The following example is for IPV4 only.

Step 1 Copy the text in the following example.

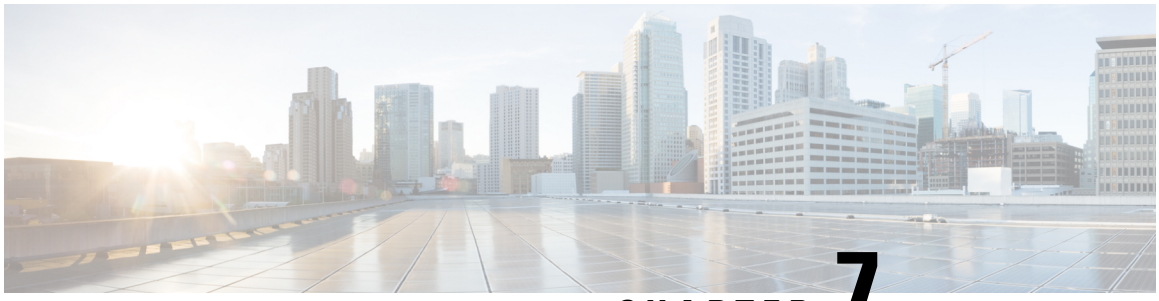
Example:

```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "vmName": {
      "value": "cisco-asav1"
    },
    "vmManagedImageId": {
      "value":
"/subscriptions/33d2517e-ca88-46aa-beb2-74ff1dd61b41/resourceGroups/ewManagedImages-rg/providers/Microsoft.Compute/images/ASAv-9.10.1-81-Managed-Image"

    },
    "adminUsername": {
      "value": "jdoe"
    },
    "adminPassword": {
      "value": "Pw0987654321"
    },
    "vmStorageAccount": {
      "value": "ciscoasavstorage"
    },
    "virtualNetworkResourceGroup": {
      "value": "ew-west8-rg"
    },
    "virtualNetworkName": {
      "value": "ew-west8-vn"
    },
    "mgmtSubnetName": {
      "value": "mgmt"
    },
    "mgmtSubnetIP": {
      "value": "10.8.3.77"
    },
    "gig00SubnetName": {
      "value": "inside"
    },
    "gig00SubnetIP": {
      "value": "10.8.2.77"
    },
    "gig01SubnetName": {
      "value": "outside"
    },
    "gig01SubnetIP": {
      "value": "10.8.1.77"
    },
    "gig02SubnetName": {
      "value": "dmz"
    },
    "gig02SubnetIP": {
      "value": "10.8.0.77"
    },
    "VmSize": {
      "value": "Standard_D3_v2"
    }
  }
}
```

Step 2 Save the file locally as a JSON file; for example, **azureParameters.json**.

Step 3 Edit the file to create a template to suit your deployment parameters.

Step 4 Use this parameter template to deploy the ASAv as described in Deploy the ASAv from Azure Using a VHD and Resource Template, on page 108.



CHAPTER 7

Deploy the ASAv Auto Scale Solution on Microsoft Azure

Auto Scale Solution for the ASA Virtual on Azure

Overview

The ASAv auto scale for Azure is a complete serverless implementation which makes use of serverless infrastructure provided by Azure (Logic App, Azure Functions, Load Balancers, Security Groups, Virtual Machine Scale Set, etc.).

Some of the key features of the ASAv auto scale for Azure implementation include:

- Azure Resource Manager (ARM) template-based deployment.
- Support for scaling metrics based on CPU.

Note See Auto Scale Logic, on page 148 for more information.

- Support for ASAv deployment and multi-availability zones.
- Completely automated configuration automatically applied to scaled-out ASAv instances.
- Support for Load Balancers and multi-availability zones.

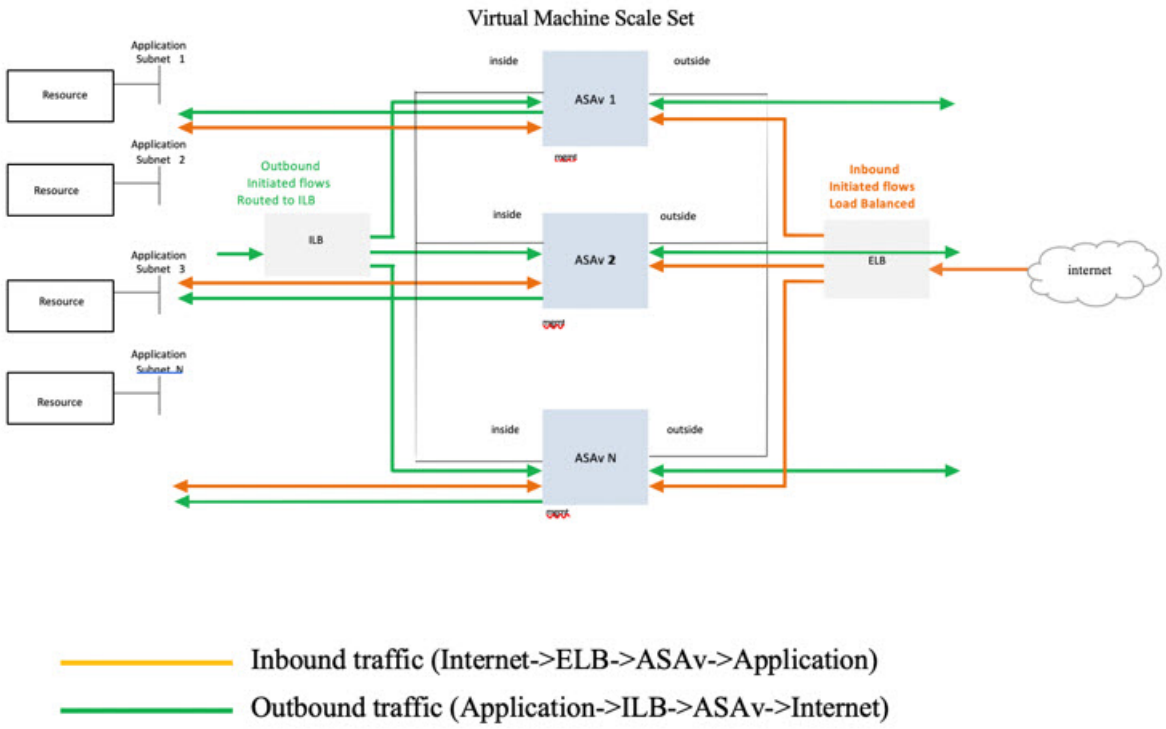
- Support for enabling and disabling the auto scale feature.
- Cisco provides an auto scale for Azure deployment package to facilitate the deployment.

Auto Scale Use Case

The ASAv auto scale for Azure is an automated horizontal scaling solution that positions an ASAv scale set sandwiched between an Azure Internal load balancer (ILB) and an Azure External load balancer (ELB).

- The ELB distributes traffic from the Internet to ASAv instances in the scale set; the firewall then forwards traffic to application.
- The ILB distributes outbound Internet traffic from an application to ASAv instances in the scale set; the firewall then forwards traffic to Internet.
- A network packet will never pass through both (internal & external) load balancers in a single connection.
- The number of ASAv instances in the scale set will be scaled and configured automatically based on load conditions.

Figure 16: ASAv Auto Scale Use Case



Scope

This document covers the detailed procedures to deploy the serverless components for the ASAv auto scale for Azure solution.

Important

- Read the entire document before you begin your deployment.
- Make sure the prerequisites are met before you start deployment.
- Make sure you follow the steps and order of execution as described herein.

Download the Deployment Package

The ASAv auto scale for Azure solution is an Azure Resource Manager (ARM) template-based deployment which makes use of the serverless infrastructure provided by Azure (Logic App, Azure Functions, Load Balancers, Virtual Machine Scale Set, and so on).

Download the files required to launch the ASAv auto scale for Azure solution. Deployment scripts and templates for your version are available in the GitHub repository.

Attention Note that Cisco-provided deployment scripts and templates for auto scale are provided as open source examples, and are not covered within the regular Cisco TAC support scope. Check GitHub regularly for updates and ReadMe instructions.

See Build Azure Functions from Source Code, on page 150 for instructions on how to build the *ASM_Function.zip* package.

Auto Scale Solution Components

The following components make up the ASAv auto scale for Azure solution.

Azure Functions (Function App)

The Function App is a set of Azure functions. The basic functionality includes:

- Communicate/Probe Azure metrics periodically.
- Monitor the ASAv load and trigger Scale In/Scale Out operations.

These functions are delivered in the form of compressed Zip package (see Build the Azure Function App Package, on page 128). The functions are as discrete as possible to carry out specific tasks, and can be upgraded as needed for enhancements and new release support.

Orchestrator (Logic App)

The Auto Scale Logic App is a workflow, i.e. a collection of steps in a sequence. Azure functions are independent entities and cannot communicate with each other. This orchestrator sequences the execution of these functions and exchanges information between them.

- The Logic App is used to orchestrate and pass information between the auto scale Azure functions.
- Each step represents an auto scale Azure function or built-in standard logic.

- The Logic App is delivered as a JSON file.
- The Logic App can be customized via the GUI or JSON file.

Virtual Machine Scale Set (VMSS)

The VMSS is a collection of homogeneous virtual machines, such as ASAv devices.

- The VMSS is capable of adding new identical VMs to the set.
- New VMs added to the VMSS are automatically attached with Load Balancers, Security Groups, and network interfaces.
- The VMSS has a built–in auto scale feature which is disabled for ASAv for Azure.
- You should not add or delete ASAv instances in the VMSS manually.

Azure Resource Manager (ARM) Template

ARM templates are used to deploy the resources required by the ASAv auto scale for Azure solution.

auto scale for Azure - The ARM template **azure_asav_autoscale.json** provides input for the Auto Scale Manager components including:

- Azure Function App
- Azure Logic App
- The Virtual Machine Scale Set (VMSS)
- Internal/External load balancers.
- Security Groups and other miscellaneous components needed for deployment.

Important The ARM template has limitations with respect to validating user input, hence it is your responsibility to validate input during deployment.

Prerequisites

Azure Resources

Resource Group

An existing or newly created Resource Group is required to deploy all the components of this solution.

Note Record the Resource Group name, the Region in which it is created, and the Azure Subscription ID for later use.

Networking

Make sure a virtual network is available or created. An auto scale deployment does not create, alter, or manage any networking resources.

The ASAv requires three network interfaces, thus your virtual network requires three subnets for:

1. Management traffic
2. Inside traffic
3. Outside traffic

The following ports should be open in the Network Security Group to which the subnets are connected:

- SSH(TCP/22)

  Required for the Health probe between the Load Balancer and ASAv.

  Required for communication between the Serverless functions and ASAv.

- Application-specific protocol/ports

  Required for any user applications (for example, TCP/80, etc.).

Note Record the virtual network name, the virtual network CIDR, the names of the 3 subnets, and the Gateway IP addresses of the outside and inside subnets.

Prepare the ASA Configuration File

Prepare an ASAv configuration file and store in a http/https server accessible by the ASAv instance. This is a standard ASA configuration file format. A scaled-out ASAv will download this file and update its configuration.

The ASA configuration file should have the following (at a minimum):

- Set DHCP IP assignment to all the interfaces.
- GigabitEthernet0/1 should be the ‘inside’ interface.
- GigabitEthernet0/0 should be the ‘outside’ interface.
- Set the gateway to the inside and outside interface.
- Enable SSH on the inside and outside interface from Azure utility IP (for health probe).
- Create a NAT configuration to forward traffic from the outside to the inside interface.
- Create an access policy to allow desired traffic.
- License the configuration. PAYG billing is not supported.

Note There is no need to specifically configure the Management interface.

The following is a sample ASA configuration file.

```
ASA Version 9.13(1)
!
interface GigabitEthernet0/1
nameif inside
security-level 100
ip address dhcp setroute
!
interface GigabitEthernet0/0
nameif outside
security-level 0
ip address dhcp setroute
!
route outside 0.0.0.0 0.0.0.0 10.12.3.1 2
!
route inside 0.0.0.0 0.0.0.0 10.12.2.1 3
!
ssh 168.63.129.0 255.255.255.0 outside
!
ssh 168.63.129.0 255.255.255.0 inside
!
object network webserver
host 10.12.2.5
object service myport
service tcp source range 1 65535 destination range 1 65535
access-list outowebaccess extended permit object myport any any log disable
access-group outowebaccess in interface outside
object service app
service tcp source eq www
nat (inside,outside) source static webserver interface destination static interface any
service app app
object network obj-any
subnet 0.0.0.0 0.0.0.0
nat (inside,outside) source dynamic obj-any interface destination static obj-any obj-any
configure terminal
dns domain-lookup management
policy-map global_policy
class inspection_default
inspect icmp
call-home
profile License
destination transport-method http
destination address http https://tools.cisco.com/its/service/oddce/services/DDCEService
license smart
feature tier standard
throughput level 2G
license smart register idtoken <TOKEN>
: end
```

Build the Azure Function App Package

The ASAv auto scale solution requires that you build an archive file: *ASM_Function.zip*. which delivers a set of discrete Azure functions in the form of a compressed ZIP package.

See Build Azure Functions from Source Code, on page 150 for instructions on how to build the *ASM_Function.zip* package.

These functions are as discrete as possible to carry out specific tasks, and can be upgraded as needed for enhancements and new release support.

Input Parameters

The following table defines the template parameters and provides an example. Once you decide on these values, you can use these parameters to create the ASAv device when you deploy the ARM template into your Azure subscription. See Deploy the Auto Scale ARM Template, on page 133.

Table 21: Template Parameters

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
resourceNamePrefix	String* (3-10 characters)	All the resources are created with name containing this prefix. Note: Use only lowercase letters. Example: asav	New
virtualNetworkRg	String	The virtual network resource group name. Example: cisco-virtualnet-rg	Existing
virtualNetworkName	String	The virtual network name (already created). Example: cisco-virtualnet	Existing
mgmtSubnet	String	The management subnet name (already created). Example: cisco-mgmt-subnet	Existing
insideSubnet	String	The inside Subnet name (already created). Example: cisco-inside-subnet	Existing
internalLbIp	String	The internal load balancer IP address for the inside subnet (already created). Example: 1.2.3.4	Existing
outsideSubnet	String	The outside subnet name (already created). Example: cisco-outside-subnet	Existing
softwareVersion	String	The ASAv Version (selected from drop-down during deployment). Default: 914.1.0Allowed: 914.1.0, 913.1.0	Existing
vmSize	String	Size of ASAv instance (selected from drop-down during deployment).	N/A

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
asaAdminUserName	String*	User name for the ASAv 'admin' user. Passwords must be 12 to 72 characters long, and must have: lowercase, uppercase, numbers, and special characters; and must have no more than 2 repeating characters. This **cannot** be ‘admin’. See Azure for VM administrator user name guidelines. **Note** There is no compliance check for this in the template.	New
asaAdminUserPassword	String*	Password for the ASAv administrator user. Passwords must be 12 to 72 characters long, and must have: lowercase, uppercase, numbers, and special characters; and must have no more than 2 repeating characters. **Note** There is no compliance check for this in the template.	New
scalingPolicy	POLICY-1 / POLICY-2	**POLICY-1**: Scale-Out will be triggered when the average load of any ASAv goes beyond the Scale Out threshold for the configured duration. **POLICY-2**: Scale-Out will be triggered when average load of all the ASAv devices in the auto scale group goes beyond the Scale Out threshold for the configured duration. In both cases Scale-In logic remains the same: Scale-In will be triggered when average load of all the ASAv devices comes below the Scale In threshold for the configured duration.	N/A

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
scalingMetricsList	String	Metrics used in making the scaling decision. Allowed: CPU Default: CPU	N/A
scaleInThreshold	String	The Scale-In threshold in percent. Default: 10 When the ASAv metric goes below this value the Scale-In will be triggered. See Auto Scale Logic, on page 148.	N/A
scaleOutThreshold	String	The Scale-Out threshold in percent. Default: 80 When the ASAv metric goes above this value, the Scale-Out will be triggered. The 'scaleOutThreshold' should always be **greater** than the 'scaleInThreshold'. See Auto Scale Logic, on page 148.	N/A
minAsaCount	Integer	The minimum ASAv instances available in the scale set at any given time. Example: 2	N/A
maxAsaCount	Integer	The maximum ASAv instances allowed in the Scale set. Example: 10 **Note** The Auto Scale logic will not check the range of this variable, hence fill this carefully.	N/A

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
metricsAverageDuration	Integer	Select from the drop-down. This number represents the time (in minutes) over which the metrics are averaged out. If the value of this variable is 5 (i.e. 5min), when the Auto Scale Manager is scheduled it will check the past 5 minutes average of metrics and based on this it will make a scaling decision. **Note** Only numbers 1, 5, 15, and 30 are valid due to Azure limitations.	N/A
initDeploymentMode	BULK / STEP	Primarily applicable for the first deployment, or when the Scale Set does not contain any ASAv instances. BULK: The Auto Scale Manager will try to deploy 'minAsaCount' number of ASAv instances in parallel at one time. STEP: The Auto Scale Manager will deploy the 'minAsaCount' number of ASAv devices one by one at each scheduled interval.	
configurationFile	String	The file path to the ASAv configuration file. Example: https://myserver/asavconfig/asaconfig.txt	N/A
*Azure has restrictions on the naming convention for new resources. Review the limitations or simply use all lowercase. **Do not use spaces or any other special characters**.			

Deploy the Auto Scale Solution

Deploy the Auto Scale ARM Template

Use the ARM template **azure_asav_autoscale.json** to deploy the resources required by the ASAv auto scale for Azure. Within a given resource group, the ARM template deployment creates the following:

- Virtual Machine Scale Set (VMSS)
- External Load Balancer
- Internal Load Balancer
- Azure Function App
- Logic App
- Security groups (For Data and Management interfaces)

Before you begin

- Download the ARM templates from the GitHub repository (https://github.com/CiscoDevNet/cisco-asav/tree/master/autoscale/azure).

Step 1 If you need to deploy the ASAv instances in multiple Azure zones, edit the ARM template based on the zones available in the Deployment region.

Example:

```
"zones": [
    "1",
    "2",
    "3"
],
```

This example shows the “Central US” region which has 3 zones.

Step 2 Edit the traffic rules required in External Load Balancer. You can add any number of rules by extending this ‘json’ array.

Example:

```
{
 "type": "Microsoft.Network/loadBalancers",
 "name": "[variables('elbName')]",
 "location": "[resourceGroup().location]",
 "apiVersion": "2018-06-01",
 "sku": {
   "name": "Standard"
 },
 "dependsOn": [
   "[concat('Microsoft.Network/publicIPAddresses/', variables('elbPublicIpName'))]"
```

```
        ],
        "properties": {
          "frontendIPConfigurations": [
            {
              "name": "LoadBalancerFrontEnd",
                "properties": {
                  "publicIPAddress": {
                    "id": "[resourceId('Microsoft.Network/publicIPAddresses/',
variables('elbPublicIpName'))]"
                  }
                }
            }
          ],
          "backendAddressPools": [
            {
              "name": "backendPool"
            }
          ],
          "loadBalancingRules": [
            {
              "properties": {
                "frontendIPConfiguration": {
                 "Id": "[concat(resourceId('Microsoft.Network/loadBalancers', variables('elbName')),
 '/frontendIpConfigurations/LoadBalancerFrontend')]"
                },
                "backendAddressPool": {
                "Id": "[concat(resourceId('Microsoft.Network/loadBalancers', variables('elbName')),
 '/backendAddressPools/BackendPool')]"
                },
                "probe": {
                "Id": "[concat(resourceId('Microsoft.Network/loadBalancers', variables('elbName')),
 '/probes/lbprobe')]"
                },
                "protocol": "TCP",
                "frontendPort": "80",
                "backendPort": "80",
                "idleTimeoutInMinutes": "[variables('idleTimeoutInMinutes')]"
              },
              "Name": "lbrule"
            }
          ],
```

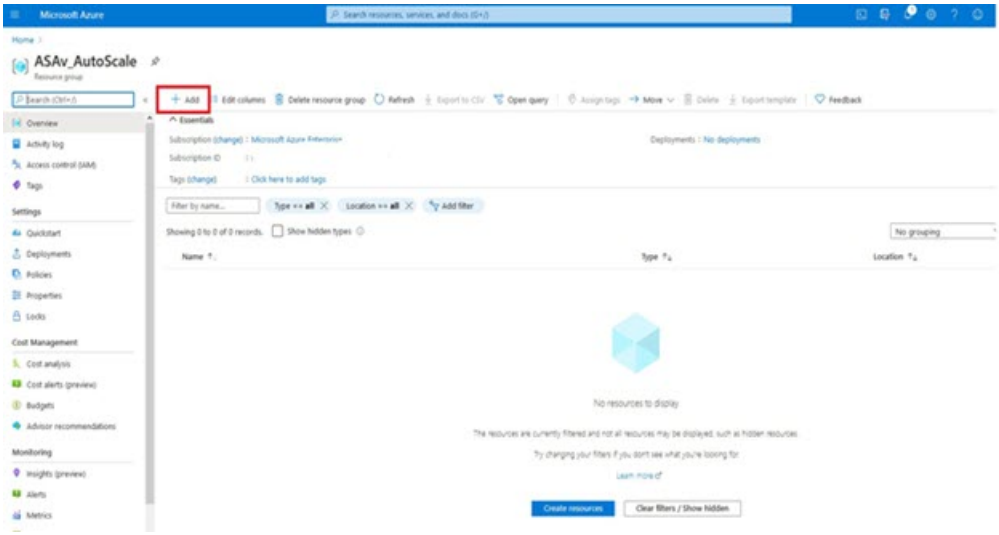
Note You can also edit this from the Azure portal post-deployment if you prefer not to edit this file.

Step 3 Log in to the Microsoft Azure portal using your Microsoft account username and password.

Step 4 Click **Resource groups** from the menu of services to access the Resource Groups blade. You will see all the resource groups in your subscription listed in the blade.

Create a new resource group or select an existing, empty resource group; for example, *ASAv_AutoScale*.

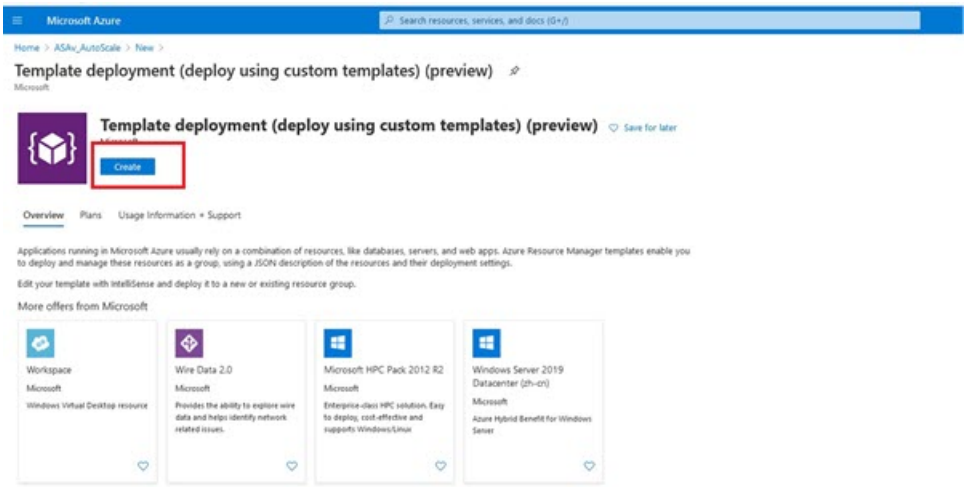
Figure 17: Azure Portal



Step 5 Click **Create a resource (+)** to create a new resource for template deployment. The Create Resource Group blade appears.

Step 6 In **Search the Marketplace**, type **Template deployment (deploy using custom templates)**, and then press **Enter**.

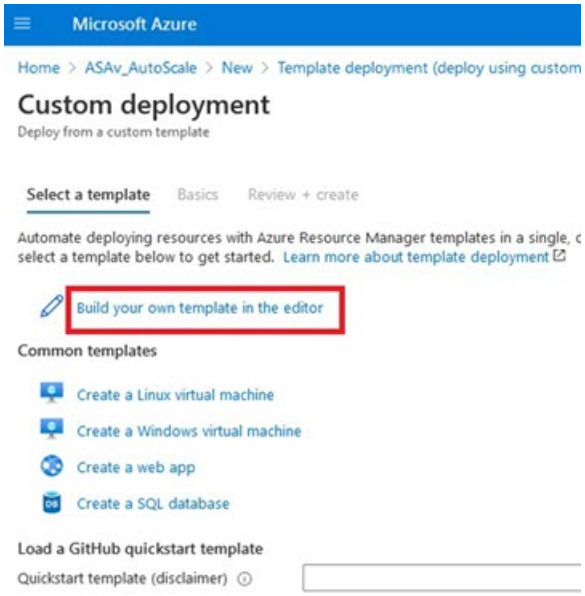
Figure 18: Custom Template Deployment



Step 7 Click **Create**.

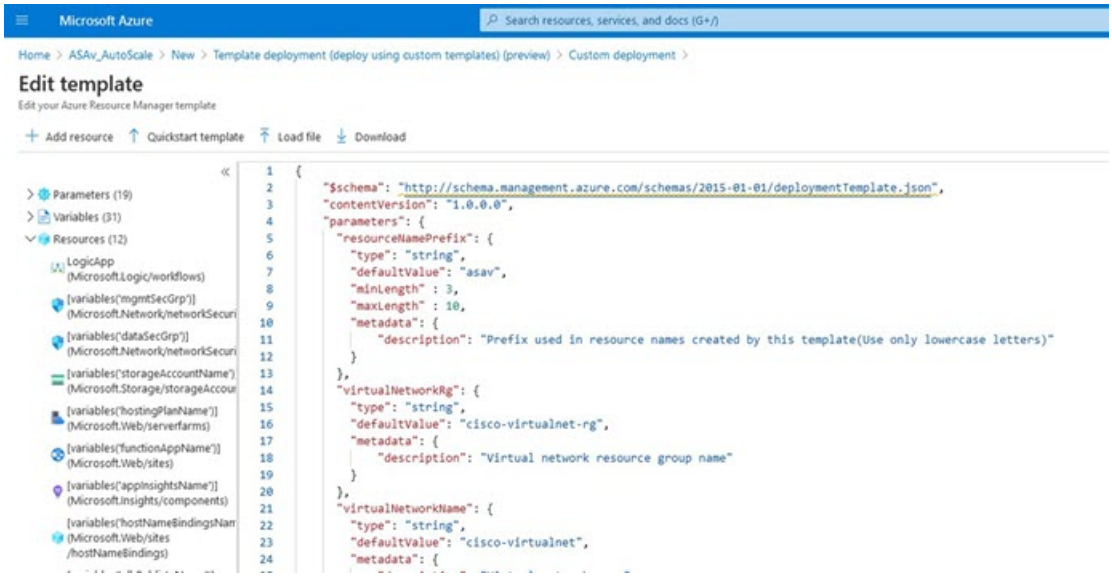
Step 8 There are several options for creating a template. Choose **Build your own template in editor**.

Figure 19: Build Your Own Template



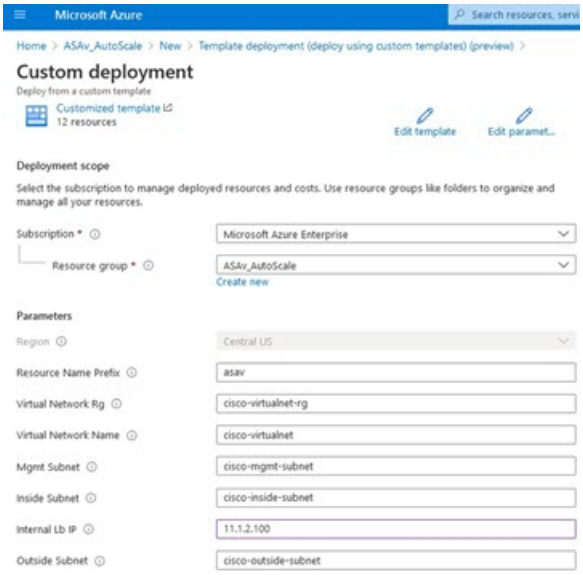
Step 9 In the **Edit template** window, delete all the default content and copy the contents from the updated *azure_asav_autoscale.json* and click **Save**.

Figure 20: Edit Template



Step 10 In next section, fill all the parameters. Refer to Input Parameters, on page 129 for details about each parameter, then click **Purchase**.

Figure 21: ARM Template Parameters

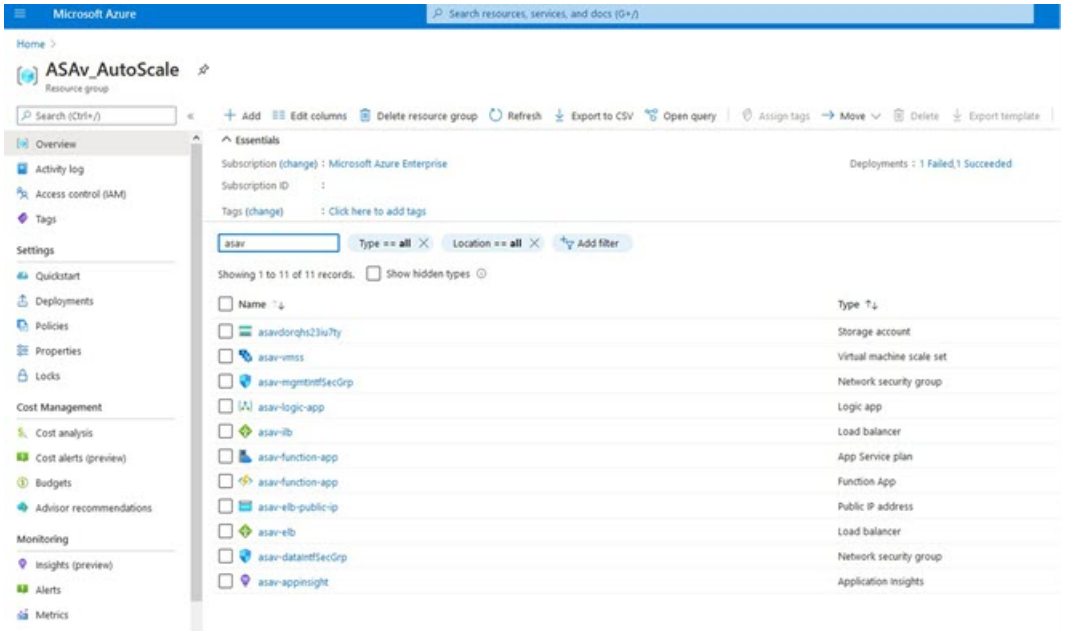


Note You can also click **Edit Parameters** and edit the JSON file or upload pre-filled contents.

The ARM template has limited input validation capabilities, hence it is your responsibility to validate the input.

Step 11 When a template deployment is successful, it creates all the required resources for the ASAv auto scale for Azure solution. See the resouses in the following figure. The Type column describes each resource, including the Logic App, VMSS, Load Balancers, Public IP address, etc.

Figure 22: ASAv Auto Scale Template Deployment



Deploy the Azure Function App

When you deploy the ARM template, Azure creates a skeleton Function App, which you then need to update and configure manually with the functions required for the Auto Scale Manager logic.

Before you begin

- Build the *ASM_Function.zip* package. See Build Azure Functions from Source Code, on page 150.

Step 1 Go to the Function App you created when you deployed the ARM template, and verify that no functions are present. In a browser go to this URL:

https://<Function App Name>.scm.azurewebsites.net/DebugConsole

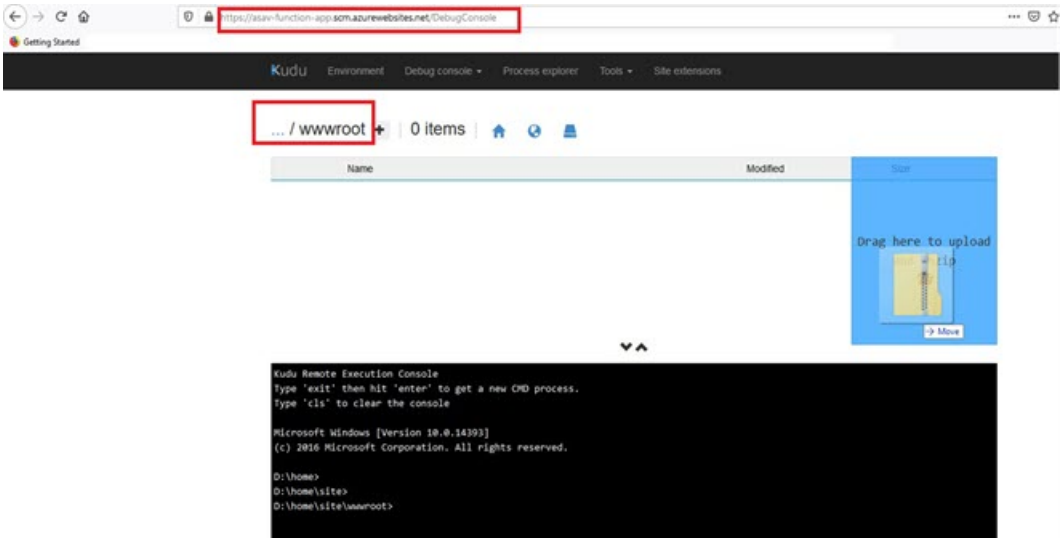
For the example in Deploy the Auto Scale ARM Template, on page 133:

https://asav-function-app.scm.azurewebsites.net/DebugConsole

Step 2 In the file explorer navigate to **site/wwwroot**.

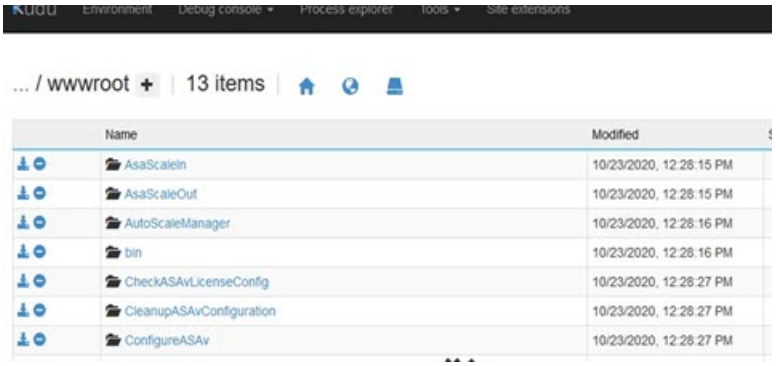
Step 3 Drag-and-drop the **ASM_Function.zip** to the right side corner of the file explorer.

Figure 23: Upload the ASAv Auto Scale Functions



Step 4 Once the upload is successful, all of the serverless functions should appear.

Figure 24: ASAv Serverless Functions

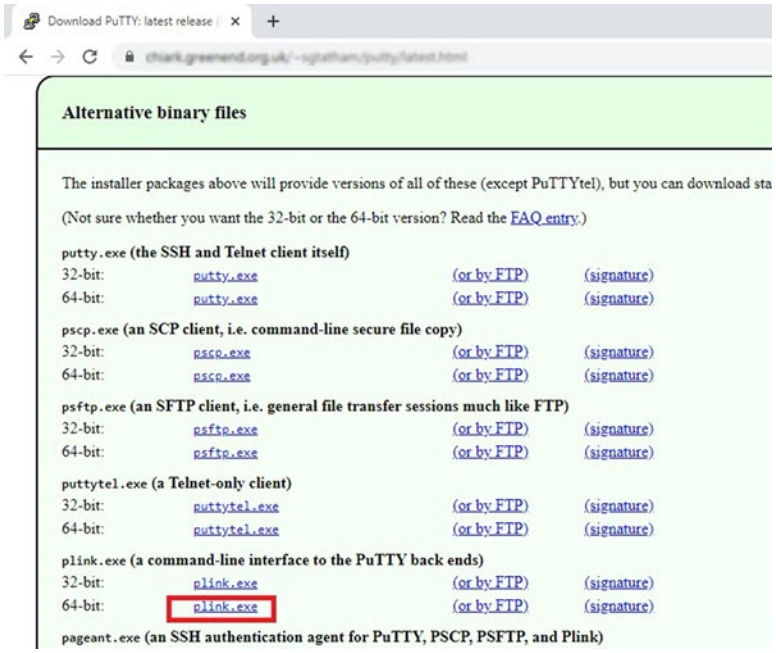


Step 5 Download the PuTTY SSH client.

Azure functions need to access the ASAv via an SSH connection. However, the opensource libraries used in the serverless code do not support the SSH key exchange algorithms used by the ASAv. Hence you need to download a pre-built SSH client.

Download the PuTTY command-line interface to the PuTTY back end (*plink.exe*) from www.putty.org.

Figure 25: Download PuTTY



Step 6 Rename the SSH client executable file **plink.exe** to **asassh.exe**.

Step 7 Drag-and-drop the **asassh.exe** to the right side corner of the file explorer, to the location where **ASM_Function.zip** was uploaded in the previous step.

Step 8 Verify the SSH client is present with the function application. Refresh the page if necessary.

Fine Tune the Configuration

There are a few configurations available to fine tune the Auto Scale Manager or to use in debugging. These options are not exposed in the ARM template, but you can edit them under the Function App.

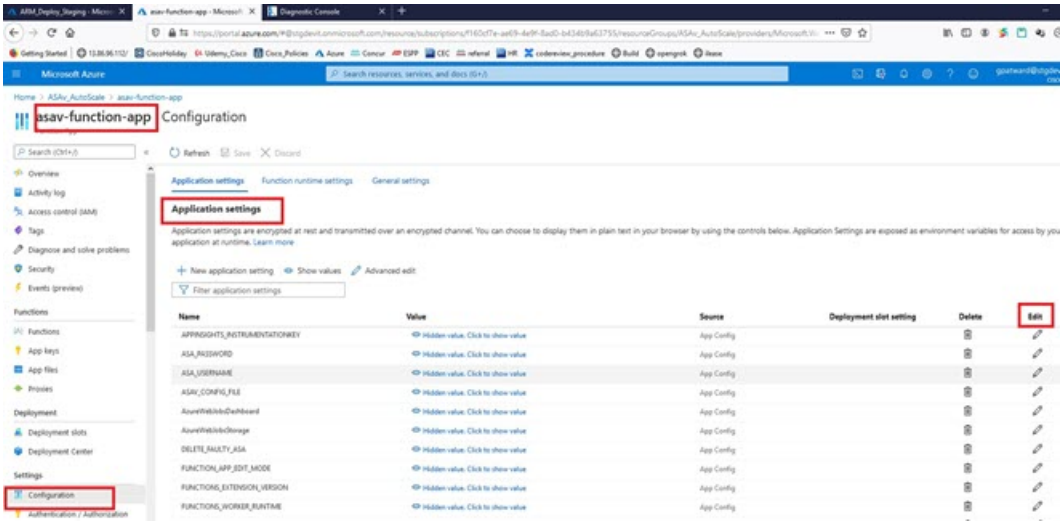
Before you begin

Note This can be edited at any time. Follow this sequence to edit the configurations.

- Disable the Function App.
- Wait for existing scheduled task to finish.
- Edit and save the configuration.
- Enable the Function App.

Step 1 In the Azure portal, search for and select the ASAv function application.

Figure 26: ASAv Function Application



Step 2 Configurations passed via the ARM template can also be edited here. Variable names may appear different from the ARM template, but you can easily identify the purpose of these variables from their name.

Most of the options are self-explanatory from the name. For example:

- Configuration Name: “DELETE_FAULTY_ASA” (Default value : YES)

  During Scale-Out, a new ASAv instance is launched and configured via the configuration file. In case the configuration fails, based on this option, Auto Scale Manager will decide to keep that ASAv instance or delete it. (YES : Delete faulty ASAv / NO : Keep the ASAv instance even if the configuration fails).

- In the Function App settings, all the variables (including variables containing a secure string like ‘password’) can be seen in clear text format by users that have access to the Azure subscription.

  If users have any security concerns with this (for example, if an Azure subscription is shared among users with lower privilages within the organization), a user can make use of Azure’s *Key Vault* service to protect passwords. Once this is configured, instead of providing a clear text ‘password’ in function settings, a user has to provide a secure identifier generated by the key vault where the password is stored.

  Note Search the Azure documentation to find the best practices to secure your application data.

Configure the IAM Role in the Virtual Machine Scale Set

Azure Identity and Access Management (IAM) is used as a part of Azure Security and Access Control to manage and control a user’s identity. Managed identities for Azure resources provides Azure services with an automatically managed identity in Azure Active Directory.

This allows the Function App to control the Virtual Machine Scale Sets (VMSS) without explicit authentication credentials.

Step 1 In the Azure portal, go to the VMSS.

Step 2 Click **Access control (IAM)**.

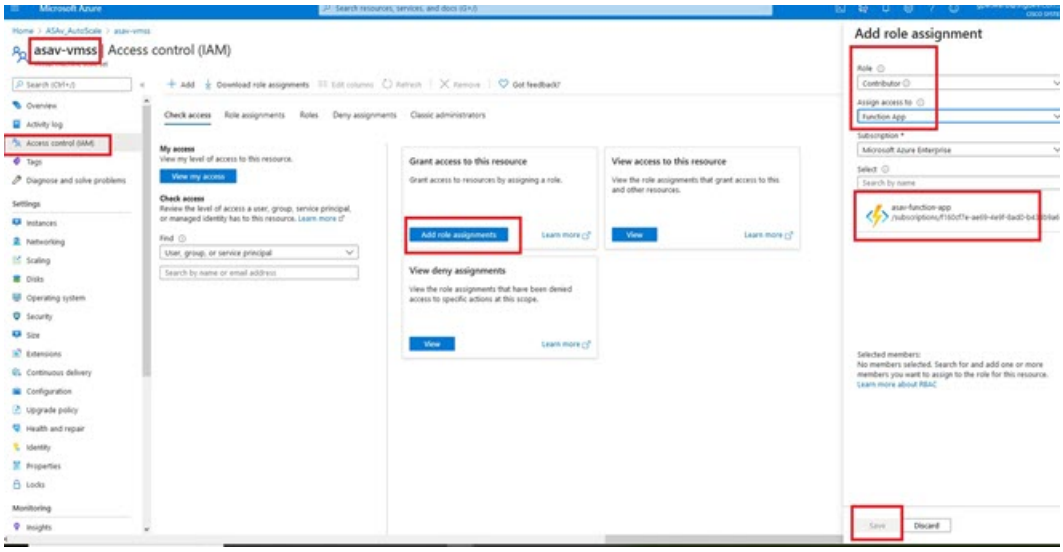
Step 3 Click **Add** to add a role assignment

Step 4 From the **Add role assignment** drop-down, choose **Contributor**.

Step 5 From the **Assign access to** drop-down, choose **Function App**.

Step 6 Select the ASAv function application.

Figure 27: AIM Role Assignment



Step 7 Click **Save**.

Note You should also verify that there are no ASAv instances launched yet.

Update Security Groups

The ARM template creates two security groups, one for the Management interface, and one for data interfaces. The Management security group will allow only traffic required for ASAv management activities. However, the data interface security group will allow all traffic.

Fine tune the security group rules based on the topology and application needs of your deployments.

Note The data interface security group should allow, at a minimum, SSH traffic from the load balancers.

Update the Azure Logic App

The Logic App acts as the orchestrator for the Autoscale functionality. The ARM template creates a skeleton Logic App, which you then need to update manually to provide the information necessary to function as the auto scale orchestrator.

Step 1 From the repository, retrieve the file *LogicApp.txt* to the local system and edit as shown below.

Important Read and understand all of these steps before proceeding.

These manual steps are not automated in the ARM template so that only the Logic App can be upgraded independently later in time.

a) Required: Find and replace all the occurrences of "SUBSCRIPTION_ID" with your subscription ID information.
b) Required: Find and replace all the occurrences of "RG_NAME" with your resource group name.
c) Required: Find and replace all of the occurrences of "FUNCTIONAPPNAME" to your function app name.

The following example shows a few of these lines in the *LogicApp.txt* file:

```
  "AutoScaleManager": {
      "inputs": {
          "function": {
              "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/AutoScaleManager"

          }
.
.
                      },
                      "Deploy_Changes_to_ASA": {
                          "inputs": {
                              "body": "@body('AutoScaleManager')",
                              "function": {
                                  "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/DeployConfiguration"

                              }
.
.
                      "DeviceDeRegister": {
                          "inputs": {
                              "body": "@body('AutoScaleManager')",
                              "function": {
                                  "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/DeviceDeRegister"

                              }
                          },
                          "runAfter": {
                              "Delay_For_connection_Draining": [
```

d) (Optional) Edit the trigger interval, or leave the default value (5). This is the time interval at which the Autoscale functionality is periodically triggered. The following example shows these lines in the *LogicApp.txt* file:

```
        "triggers": {
            "Recurrence": {
                "conditions": [],
                "inputs": {},
                "recurrence": {
                    "frequency": "Minute",
                    "interval": 5
                },
```

e) (Optional) Edit the time to drain, or leave the default value (5). This is the time interval to drain existing connections from the ASAv before deleting the device during the Scale-In operation. The following example shows these lines in the *LogicApp.txt* file:

```
"actions": {
     "Branch_based_on_Scale-In_or_Scale-Out_condition": {
         "actions": {
             "Delay_For_connection_Draining": {
                 "inputs": {
                     "interval": {
                         "count": 5,
                         "unit": "Minute"
                     }
```

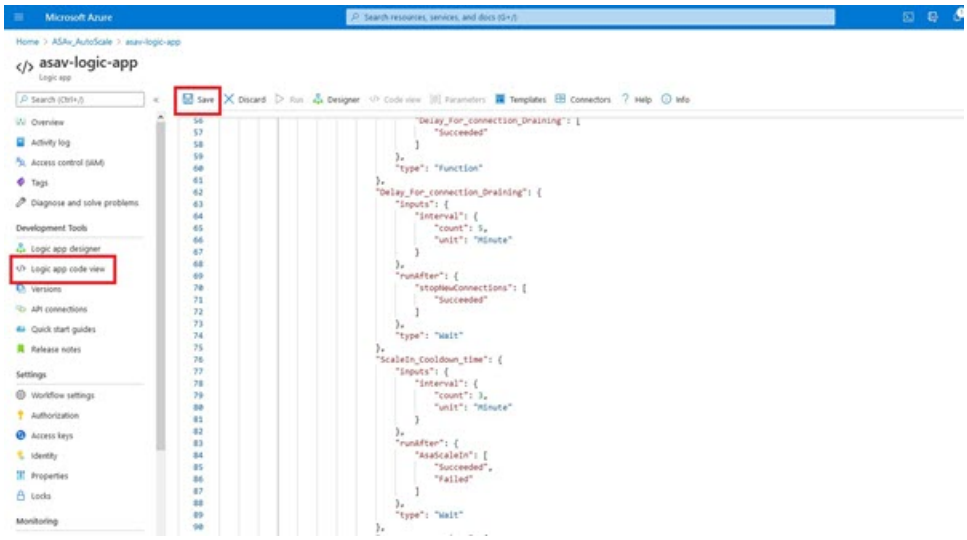
f) (Optional) Edit the cool down time, or leave the default value (10). This is the time to perform NO ACTION after the Scale-Out is complete. The following example shows these lines in the *LogicApp.txt* file:

```
"actions": {
    "Branch_based_on_Scale-Out_or_Invalid_condition": {
        "actions": {
            "Cooldown_time": {
                "inputs": {
                    "interval": {
                        "count": 10,
                        "unit": "Second"
                    }
```

Note These steps can also be done from the Azure portal. Consult the Azure documentation for more information.

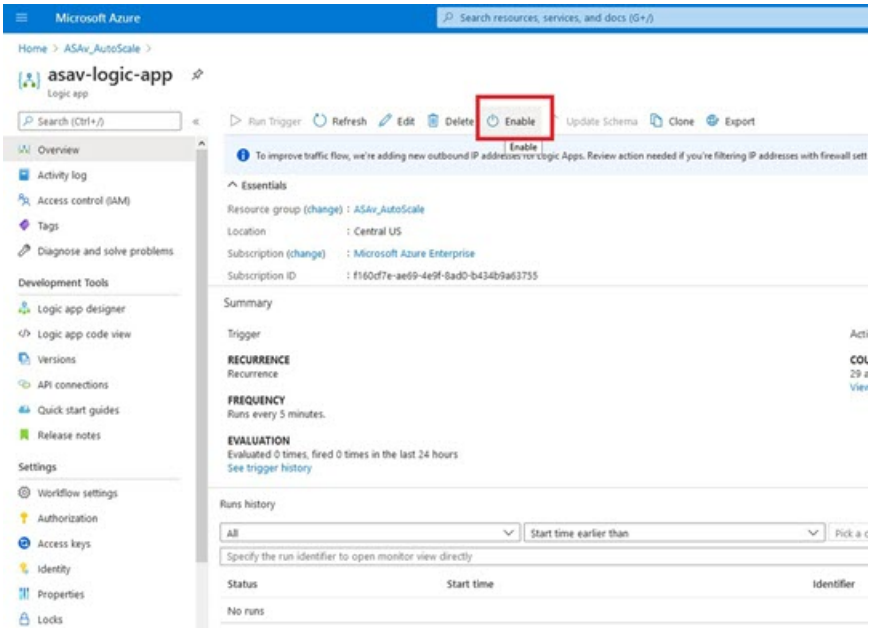
Step 2 Go to the **Logic App code view**, delete the default contents and paste the contents from the edited *LogicApp.txt* file, and click **Save**.

Figure 28: Logic App Code View



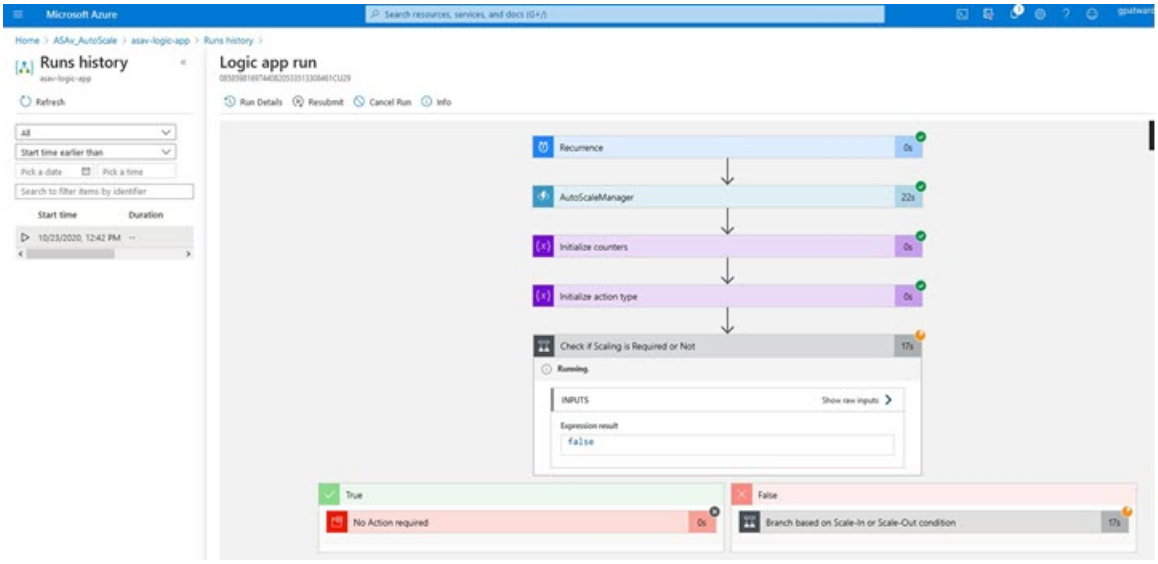
Step 3 When you save the Logic App, it is in a 'Disabled' state. Click **Enable** when you want to start the Auto Scale Manager.

Figure 29: Enable Logic App



Step 4 Once enabled, the tasks start running. Click the 'Running' status to see the activity.

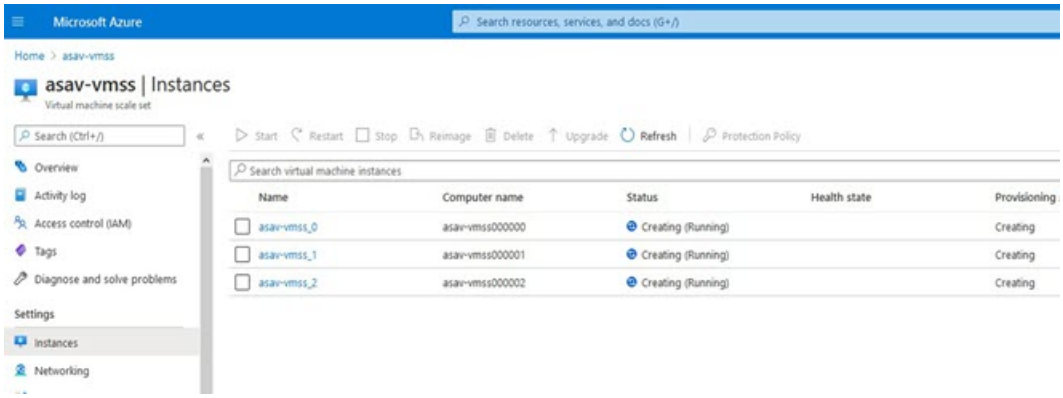
Figure 30: Logic App Running Status



Step 5 Once the Logic App starts, all the deployment-related steps are complete.

Step 6 Verify in the VMSS that ASAv instances are being created.

Figure 31: ASAv Instances Running



In this example, three ASAv instances are launched because 'minAsaCount' was set to '3' and 'initDeploymentMode' was set to 'BULK' in the ARM template deployment.

Upgrade the ASAv

The ASAv upgrade is supported only in the form of an image upgrade of virtual machine scale set (VMSS). Hence, you upgrade the ASAv through the Azure REST API interface.

Note You can use any REST client to upgrade the ASAv.

Before you begin

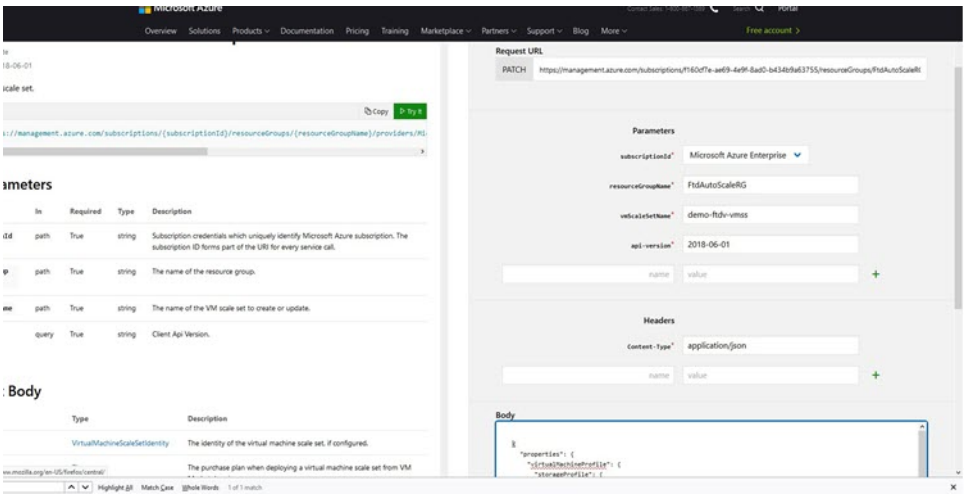
- Obtain the new ASAv image version available in market place (example: 914.001).
- Obtain the SKU used to deploy original scale set (example: asav-azure-byol).
- Obtain the Resource Group and the virtual machine scale set name.

Step 1 In a browser go to the following URL:

https://docs.microsoft.com/en-us/rest/api/compute/virtualmachinescalesets/update#code-try-0

Step 2 Enter the details in the Parameters section.

Figure 32: Upgrade the ASAv



Step 3 Enter the JSON input containing the new ASAv image version, SKU, and trigger RUN in the **Body** section.

```
{
 "properties": {
        "virtualMachineProfile": {
              "storageProfile": {
                      "imageReference": {
                           "publisher": "cisco",
                           "offer": "cisco-asav",
                           "sku": "asav-azure-byol",
                            "version": "650.32.0"
                      }
                  },
            }
      }
}
```

Step 4 A successful response from Azure means that the VMSS has accepted the change.

The new image will be used in the new ASAv instances which will get launched as part of Scale-Out operation.

- Existing ASAv instances will continue to use the old software image while they exist in a scale set.
- You can override the above behavior and upgrade the existing ASAv instances manually. To do this, click the **Upgrade** button in the VMSS. It will reboot and upgrade the selected ASAv instances. You must reregister and reconfigure these upgraded ASAv instances manually. **Note that this method is NOT recommended.**

Auto Scale Logic

Scale-Out Logic

- **POLICY-1**: Scale-Out will be triggered when the average load of **any** ASAv goes beyond the Scale-Out threshold for the configured duration.
- **POLICY-2**: Scale-Out will be triggered when average load of **all** of the ASAv devices go beyond Scale-Out threshold for the configured duration.

Scale-In Logic

- If the CPU utilization of **all** of the ASAv devices goes below the configured Scale-In threshold for the configured duration.

Notes

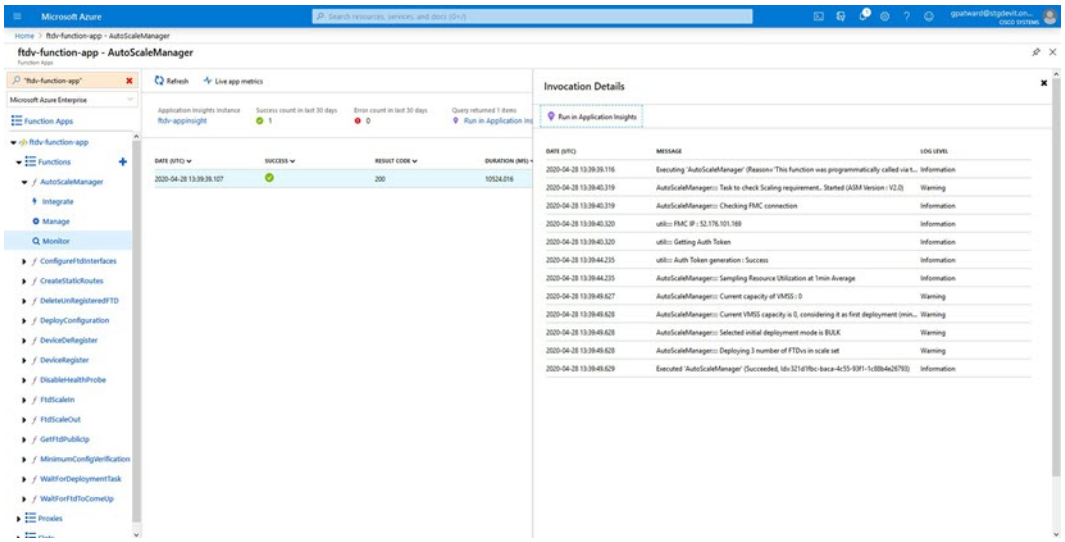
- Scale-In/Scale-Out occurs in steps of 1 (i.e. only 1 ASAv will be scaled in/out at a time).
- The above logic is based on the assumption that the load balancer will try to equally distribute connections across all ASAv devices, and on an average all ASAv devices should be loaded equally.

Auto Scale Logging and Debugging

Each component of the serverless code has its own logging mechanism. In addition, logs are published to application insight.

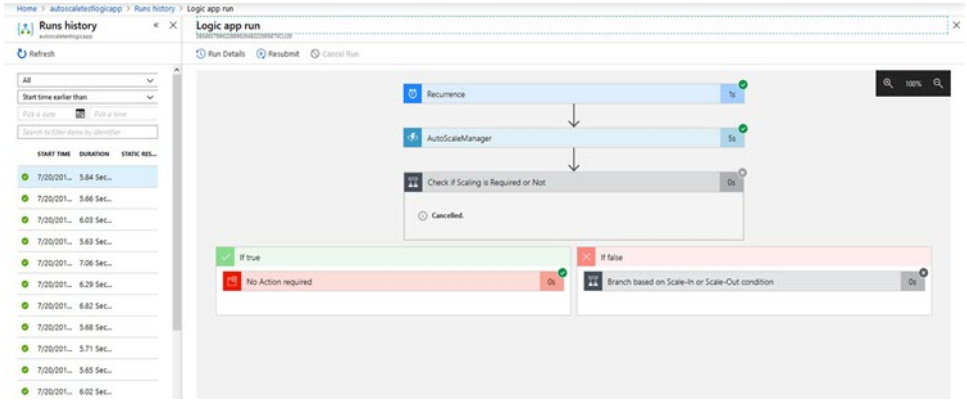
- Logs of individual Azure functions can be viewed.

Figure 33: Azure Function Logs



- Similar logs for each run of the Logic App and its individual components can be viewed.

Figure 34: Logic App Run Logs



- If needed, any running task in the Logic App can be stopped/terminated at any time. However, currently running ASAv devices getting launched/terminated will be in an inconsistent state.
- The time taken for each run/individual task can be seen in the Logic App.
- The Function App can be upgraded at any time by uploading a new zip. Stop the Logic App and wait for all tasks to complete before upgrading the Function App.

Auto Scale Guidelines and Limitations

Be aware of the following guidelines and limitations when deploying the ASAv auto scale for Azure:

- Scaling decisions are based on CPU utilization.
- The ASAv Management interface is configured to have public IP address.
- Only IPv4 is supported.
- The ARM template has limited input validation capabilities, hence it is your responsibility to provide the correct input validation.
- The Azure administrator can see sensitive data (such as admin login credentials and passwords) in plain text format inside Function App environment. You can use the *Azure Key Vault* service to secure sensitive data.
- Any changes in configuration won't be automatically reflected on already running instances. Changes will be reflected on upcoming devices only. Any such changes should be manually pushed to already existing devices.
- If you are facing issues while manually updating the configuration on existing instances, we recommend removing these instances from the Scaling Group and replacing them with new instances.

Troubleshooting

The following are common error scenarios and debugging tips for the ASAv auto scale for Azure:

- Unable to SSH into the ASAv: Check if a complex password is passed to the ASAv via the template; check if Security Groups allow SSH connections.
- Load Balancer Health check failure: Check if the ASAv responds to SSH on data interfaces; check Security Group settings.
- Traffic issues: Check Load Balancer rules, NAT rules / Static routes configured in ASAv; check Azure virtual network / subnets / gateway details provided in the template and Security Group rules.
- Logic App failed to access VMSS: Check if the IAM role configuration in VMSS is correct.
- Logic App runs for very long time: Check SSH access on scaled-out ASAv devices; check the state of the ASAv devices in Azure VMSS.
- Azure Function throwing error related to subscription ID : Verify that you have a default subscription selected in your account.
- Failure of Scale-In operation: Sometimes, Azure takes a considerably long time to delete an instance in such situations, Scale-in operation may time out and report an error; but eventually the instance, will get deleted.
- Before doing any configuration change, make sure to disable the logic application and wait for all the running tasks to complete.

Build Azure Functions from Source Code

System Requirements

- Microsoft Windows desktop/laptop.
- Visual Studio (tested with Visual studio 2019 version 16.1.3)

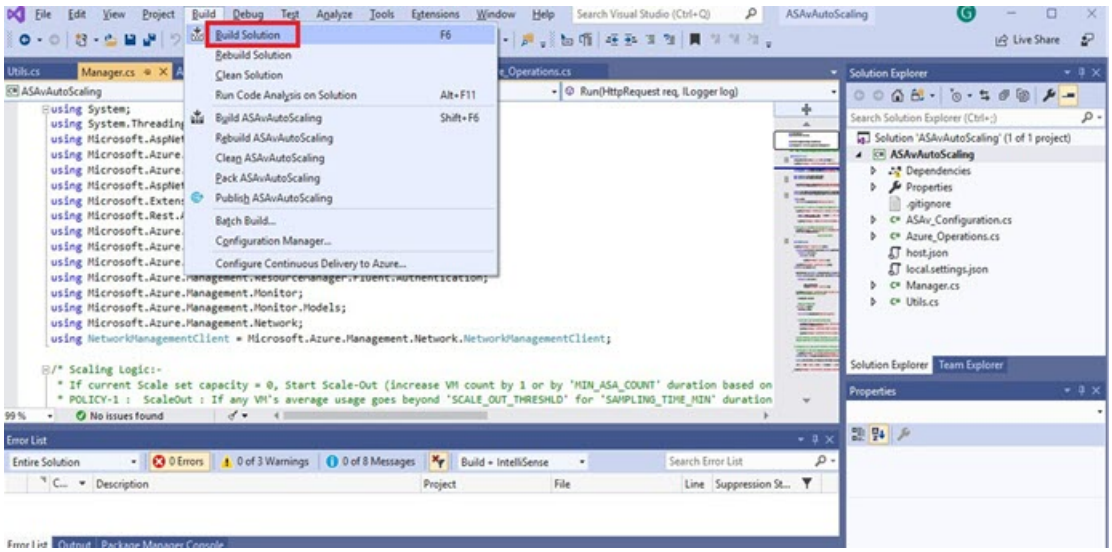
Note Azure functions are written using C#.

- The "Azure Development" workload needs to be installed in Visual Studio.

Build with Visual Studio

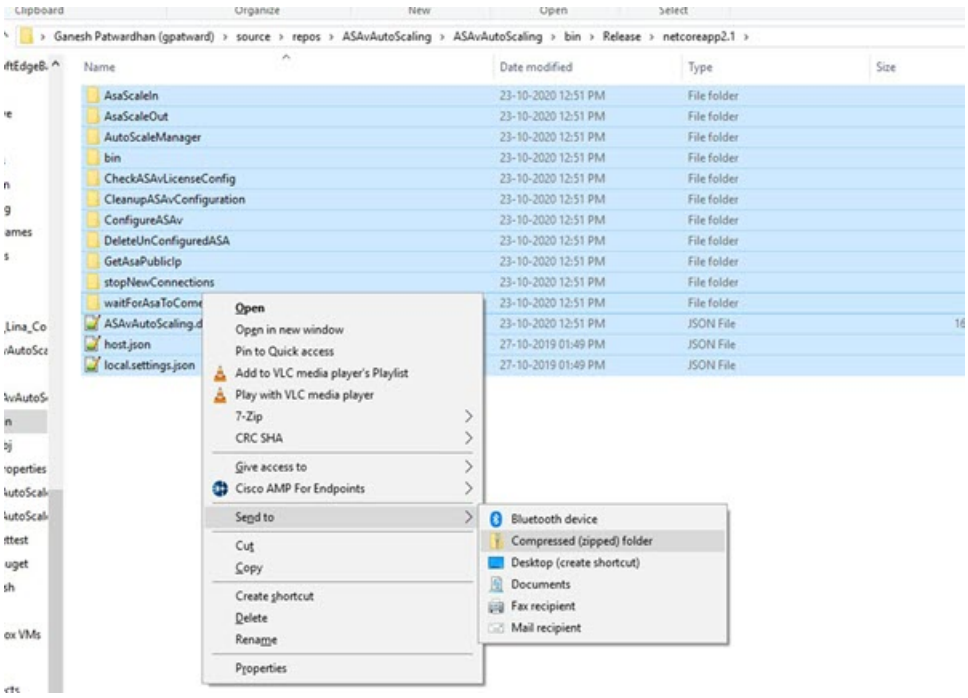
1. Download the 'code' folder to the local machine.
2. Navigate to the folder 'ASAAutoScaling'.
3. Open the project file 'ASAAutoScaling.csproj' in Visual Studio.
4. Use Visual Studio standard procedure to Clean and Build.

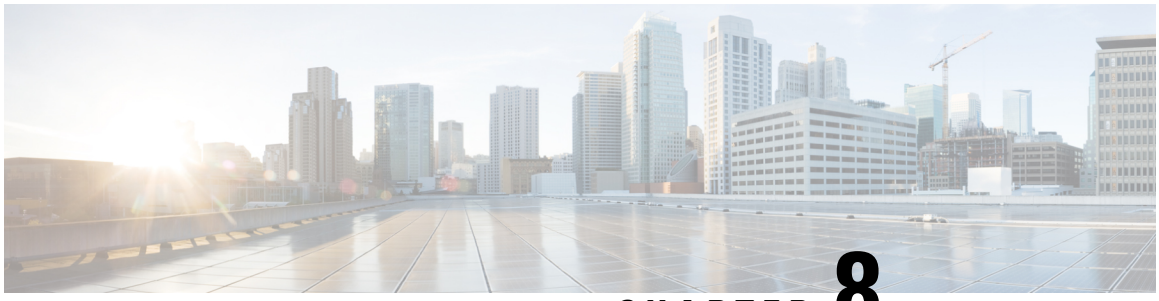
Figure 35: Visual Studio Build



5. Once the build is compiled successfully, navigate to the **\bin\Release\netcoreapp2.1** folder.
6. Select all the contents, click **Send to** > **Compressed (zipped) folder**, and save the ZIP file as *ASM_Function.zip*.

Figure 36: Build ASM_Function.zip





CHAPTER 8

Deploy the ASAv On the Rackspace Cloud

You can deploy the ASAv on the Rackspace cloud.

Important Beginning with 9.13(1), any ASAv license now can be used on any supported ASAv vCPU/memory configuration. This allows ASAv customers to run on a wide variety of VM resource footprints.

Overview

Rackspace is a leading provider of expertise and managed services across all the major public and private cloud technologies. The Rackspace Cloud is a set of cloud computing products and services billed on a utility computing basis.

You can deploy the ASAv for Rackspace as a virtual appliance in the Rackspace cloud. This chapter explains how to install and configure a single instance ASAv appliance.

Instance types in the Rackspace Cloud are referred to as *flavors*. The term flavor refers to a server's combination of RAM size, vCPUs, network throughput (RXTX factor), and disk space. The following table lists Rackspace flavors suitable for ASAv deployment.

Table 22: Rackspace Supported Flavors

Flavor	Attributes		Aggregate Bandwidth
	vCPUs	Memory (GB)	
general 1-2	2	2	400 Mbps
general 1-4	4	4	800 Mbps
general 1-8	8	8	1.6 Gbps

Flavor	Attributes		Aggregate Bandwidth
	vCPUs	Memory (GB)	
compute 1-4	2	3.75	312.5 Mbps
compute 1-8	4	7.5	625 Mbps
compute 1-15	8	15	1.3 Gbps
memory 1-15	2	15	625 Mbps
memory 1-15	4	30	1.3 Gbps
memory 1-15	8	60	2.5 Gbps

About Rackspace Flavors

Rackspace Virtual Cloud Server Flavors fall into the following classes:

- **General Purpose v1**
  - Useful for a range of use cases, from general-purpose workloads to high performance websites.
  - The vCPUs are oversubscribed and “burstable”; in other words, there are more vCPUs allocated to the Cloud Servers on a physical host than there are physical CPU threads.
- **Compute v1**
  - Optimized for web servers, application servers, and other CPU-intensive workloads.
  - The vCPUs are “reserved”; in other words, there are never more vCPUs allocated to the Cloud Servers on a physical host than there are physical CPU threads on that host.
- **Memory v1**
  - Recommended for memory-intensive workloads.
- **I/O v1**
  - Ideal for high performance applications and databases that benefit from fast disk I/O.

Prerequisites

- Create a Rackspace account.

  All Rackspace Public Cloud accounts are set to the Managed Infrastructure service level by default. You can upgrade to the Managed Operations service level inside the Cloud Control Panel. At the top of the Cloud Control Panel, click your account username and then select Upgrade Service Level.
- License the ASAv. Until you license the ASAv, it will run in degraded mode, which allows only 100 connections and throughput of 100 Kbps. See Licensing for the ASAv, on page 1.
- Interface requirements:

- Management interface
- Inside and outside interfaces
- (Optional) Additional subnet (DMZ)

- Communications paths:
  - Management interface—Used to connect the ASAv to the ASDM; can't be used for through traffic.
  - Inside interface (required)—Used to connect the ASAv to inside hosts.
  - Outside interface (required)—Used to connect the ASAv to the public network.
  - DMZ interface (optional)—Used to connect the ASAv to the DMZ network.
- For ASA and ASAv system compatibility and requirements, see Cisco ASA Compatibility.

Rackspace Cloud Network

Your cloud configuration can include several kinds of networks, connected as appropriate for your needs. You can manage the networking capabilities of your cloud servers in many of the same ways you manage your other networks. Your ASAv deployment will interact primarily with three types of virtual networks in the Rackspace Cloud:

- PublicNet—Connects cloud infrastructure components such as cloud servers, cloud load balancers, and network appliances to the Internet.
  - Use PublicNet to connect the ASAv to the Internet.
  - The ASAv attaches to this network via the Management0/0 interface.
  - PublicNet is dual-stacked for IPv4 and IPv6. When you create a server with PublicNet, your server receives an IPv4 address and an IPv6 address by default.
- ServiceNet—An internal, IPv4-only multi-tenant network within each Rackspace cloud region.
  - ServiceNet is optimized to carry traffic across servers within your configuration (east-west traffic).
  - It provides servers with no-cost access to regionalized services such as Cloud Files, Cloud Load Balancers, Cloud Databases, and Cloud Backup.
  - The networks 10.176.0.0/12 and 10.208.0.0/12 are reserved for ServiceNet. Any servers that have ServiceNet connectivity will be provisioned with an IP address from one of these networks.
  - The ASAv attaches to this network via the Gigabit0/0 interface.
- Private Cloud Networks—Cloud Networks lets you create and manage secure, isolated networks in the cloud.
  - These networks are fully single tenant, and you have complete control over the network topology, IP addressing (IPv4 or IPv6), and which Cloud Servers are attached.
  - Cloud Networks are regional in scope, and you can attach them to any of your Cloud Servers in a given region.

- You can create and manage Cloud Networks via an API or by using the Rackspace Cloud Control Panel.

  The ASAv attaches to these networks via Gigabit0/1 through Gigabit0/8 interfaces.

Rackspace Day 0 Configuration

When a VM is deployed in the Rackspace Cloud, a CD-ROM device containing files with Rackspace provisioning information is attached to the VM. The provisioning information includes:

- The hostname
- IP addresses for required interfaces
- Static IP routes
- Username and password (Optional SSH public key)
- DNS servers
- NTP servers

These files are read during the initial deployment and ASA configuration is generated.

ASAv Hostname

By default, the ASAv hostname is the name you assign to your cloud server when you begin to build your ASAv.

```
hostname rackspace-asav
```

The ASA hostname configuration will only accept a hostname that complies with RFCs 1034 and 1101:

- Must start and end with a letter or digit.
- Interior characters must be a letter, a digit or a hyphen.

Note The ASAv will alter the cloud server name to comply with these rules while making it as close as possible to the original cloud server name. It will drop special characters from the beginning and end of the cloud server name, and replace non-compliant interior characters with a hyphen.

For example, a cloud server named **ASAv-9.13.1.200** will have hostname **ASAv-9-13-1-200**.

Interfaces

Interfaces are configured in the following manner:

- Management0/0
  - Named ‘outside’ because it is connected to the PublicNet.
  - Rackspace assigns both IPv4 and IPv6 public addresses to the PublicNet interface.

- Gigabit0/0
  - Named ‘management’ since it is connected to the ServiceNet.
  - Rackspace assigns an IPv4 address from the ServiceNet subnet for the Rackspace region.
- Gigabit0/1 through Gigabit0/8
  - Named ‘inside’, ‘inside02’, ‘inside03’, etc. because they are connected to private Cloud Networks.
  - Rackspace assigns an IP address from the Cloud Network subnet.

The interface configuration for an ASAv with 3 interfaces would look something like this:

```
interface GigabitEthernet0/0
 nameif management
 security-level 0
 ip address 10.176.5.71 255.255.192.0
!
interface GigabitEthernet0/1
 nameif inside
 security-level 100
 ip address 172.19.219.7 255.255.255.0
!
interface Management0/0
 nameif outside
 security-level 0
 ip address 162.209.103.109 255.255.255.0
 ipv6 address 2001:4802:7800:1:be76:4eff:fe20:1763/64
```

Static Routes

Rackspace provisions the following static IP routes:

- Default IPv4 route via PublicNet interface (**outside**).
- Default IPv6 route via PublicNet interface.
- Infrastructure subnet routes on ServiceNet interface (**management**).

```
route outside 0.0.0.0 0.0.0.0 104.130.24.1 1
ipv6 route outside ::/0 fe80::def
route management 10.176.0.0 255.240.0.0 10.176.0.1 1
route management 10.208.0.0 255.240.0.0 10.176.0.1 1
```

Login Credentials

A username ‘admin’ is created with a password created by Rackspace. A public key for user ‘admin’ is created if the cloud server is deployed with a Rackspace Public Key.

```
username admin password <admin_password> privilege 15
username admin attributes
 ssh authentication publickey <public_key>
```

The Day0 SSH configuration:

- SSH via PublicNet interface (**outside**) is enabled for IPv4 and IPv6.
- SSH via ServiceNet interface (**management**) is enabled for IPv4 .
- Configure stronger key exchange group at request of Rackspace.

```
aaa authentication ssh console LOCAL
ssh 0 0 management
ssh 0 0 outside
ssh ::0/0 outside
ssh version 2
ssh key-exchange group dh-group14-sha1
```

DNS and NTP

Rackspace provides two IPv4 service addresses to be used for DNS and NTP.

```
dns domain-lookup outside
dns server-group DefaultDNS
 name-server 69.20.0.164
 name-server 69.20.0.196

ntp server 69.20.0.164
ntp server 69.20.0.196
```

Deploy the ASAv

You can deploy the ASAv as a virtual appliance in the Rackspace Cloud. This procedure shows you how to install a single instance ASAv appliance.

Before you begin

Review the Rackspace Day 0 Configuration, on page 156 topic for a description of the configuration parameters that the Rackspace Cloud enables for a successful ASAv deployment, including hostname requirement, interface provisioning, and networking information.

Step 1 On the Rackspace mycloud portal, go to **SERVERS** > **CREATE RESOURCES** > **Cloud Server**.

Step 2 On the **Create Server** page, enter your **Server Details**:

a) Enter the name for your ASAv machine in the **Server Name** field.

b) Choose your region from the **Region** drop-down list.

Step 3 Under **Image**, choose **Linux/Appliances** > **ASAv** > **Version**.

Note You would typically choose the most recent supported version when deploying a new ASAv.

Step 4 Under **Flavor**, choose a **Flavor Class** that fits your resource needs; see Table 22: Rackspace Supported Flavors, on page 153 for a list of suitable VMs.

Important Beginning with 9.13(1), the minimum memory requirement for the ASAv is 2GB. When deploying an ASAv with more than 1 vCPU, the minimum memory requirement for the ASAv is 4GB.

Step 5 (Optional) Under **Advanced Options**, configure an SSH key.

See Managing access with SSH keys for complete information on SSH keys in the Rackspace Cloud.

Step 6 Review any applicable **Recommended Installs** and **Itemized Charges** for your ASAv, then click **Create Server**.

The root admin password displays. Copy the password, then dismiss the dialog.

Step 7 After you create the server, the server details page displays. Wait for the server to show an active status. This usually takes a few minutes.

What to do next

- Connect to the ASAv.
- Continue configuration using CLI commands available for input via SSH or use ASDM. See Start ASDM, on page 281 for instructions for accessing the ASDM.

CPU Usage and Reporting

The CPU Utilization report summarizes the percentage of the CPU used within the time specified. Typically, the Core operates on approximately 30 to 40 percent of total CPU capacity during nonpeak hours and approximately 60 to 70 percent capacity during peak hours.

vCPU Usage in the ASA Virtual

The ASA virtual vCPU usage shows the amount of vCPUs used for the data path, control point, and external processes.

The Rackspace reported vCPU usage includes the ASA virtual usage as described plus:

- ASA virtual idle time
- %SYS overhead used for the ASA virtual machine
- Overhead of moving packets between vSwitches, vNICs, and pNICs. This overhead can be quite significant.

CPU Usage Example

The **show cpu usage** command can be used to display CPU utilization statistics.

Example

```
Ciscoasa#show cpu usage

CPU utilization for 5 seconds = 1%; 1 minute: 2%; 5 minutes: 1%
```

The following is an example in which the reported vCPU usage is substantially different:

- ASA Virtual reports: 40%

- DP: 35%
- External Processes: 5%
- ASA (as ASA Virtual reports): 40%
- ASA idle polling: 10%
- Overhead: 45%

The overhead is used to perform hypervisor functions and to move packets between NICs and vNICs using the vSwitch.

Rackspace CPU Usage Reporting

In addition to viewing CPU, RAM, and disk space configuration information for available Cloud Servers, you can also view disk, I/O, and networking information. Use this information to help you decide which Cloud Server is right for your needs. You can view the available servers through either the command-line nova client or the **Cloud Control Panel** interface.

On the command line, run the following command:

```
nova flavor-list
```

All available server configurations are displayed. The list contains the following information:

- ID - The server configuration ID
- Name - The configuration name, labeled by RAM size and performance type
- Memory_MB - The amount of RAM for the configuration
- Disk - The size of the disk in GB (for general purpose Cloud Servers, the size of the system disk)
- Ephemeral - The size of the data disk
- Swap - The size of the swap space
- VCPUs - The number of virtual CPUs associated with the configuration
- RXTX_Factor - The amount of bandwidth, in Mbps, allocated to the PublicNet ports, ServiceNet ports, and isolated networks (cloud networks) attached to a server
- Is_Public - Not used

ASA Virtual and Rackspace Graphs

There are differences in the CPU % numbers between the ASA Virtual and Rackspace:

- The Rackspace graph numbers are always higher than the ASA Virtual numbers.
- Rackspace calls it %CPU usage; the ASA Virtual calls it %CPU utilization.

The terms “%CPU utilization” and “%CPU usage” mean different things:

- CPU utilization provides statistics for physical CPUs.

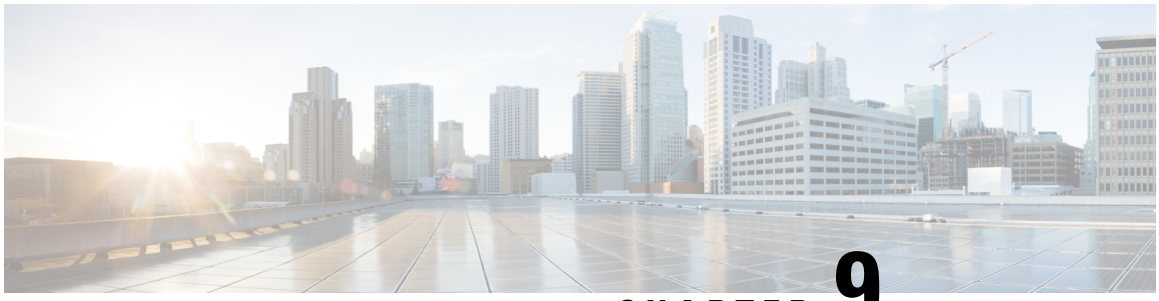
- CPU usage provides statistics for logical CPUs, which is based on CPU hyperthreading. But because only one vCPU is used, hyperthreading is not turned on.

Rackspace calculates the CPU % usage as follows:

Amount of actively used virtual CPUs, specified as a percentage of the total available CPUs

This calculation is the host view of the CPU usage, not the guest operating system view, and is the average CPU utilization over all available virtual CPUs in the virtual machine.

For example, if a virtual machine with one virtual CPU is running on a host that has four physical CPUs and the CPU usage is 100%, the virtual machine is using one physical CPU completely. The virtual CPU usage calculation is Usage in MHz / number of virtual CPUs x core frequency



CHAPTER 9

Deploy the ASAv Using Hyper-V

You can deploy the ASAv using Microsoft Hyper-V.

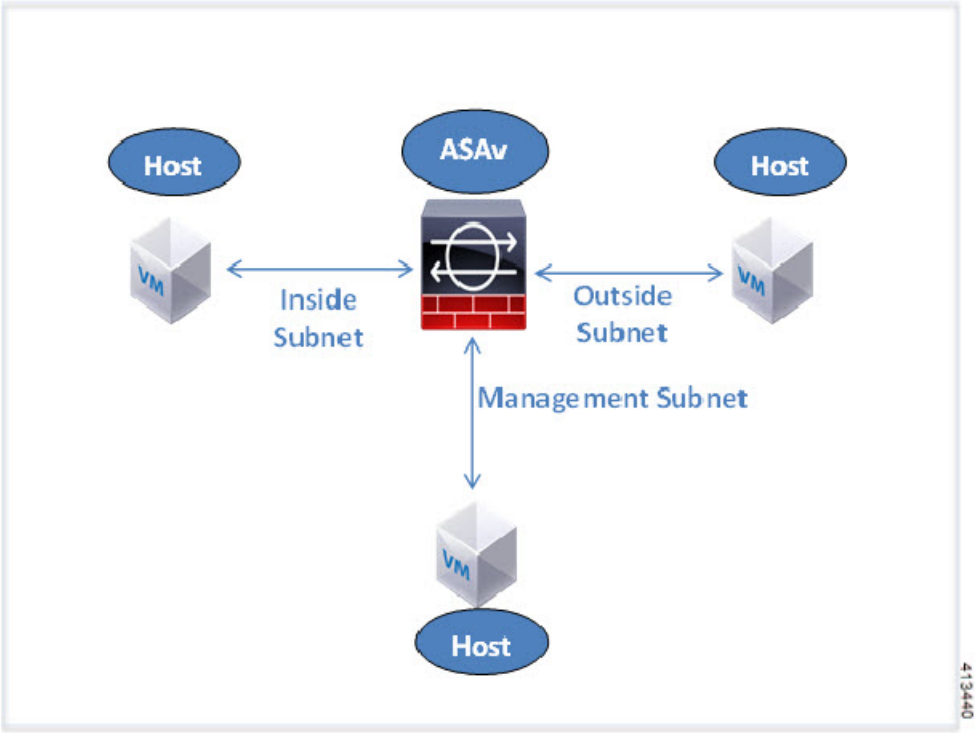
Important Beginning with 9.13(1), the minimum memory requirement for the ASAv is 2GB. If your current ASAv runs with less than 2GB of memory, you cannot upgrade to 9.13(1) from an earlier version without increasing the memory of your ASAv machine. You can also redeploy a new ASAv machine with version 9.13(1).

Overview

You can deploy Hyper-V on a standalone Hyper-V server or through the Hyper-V Manager. For instructions to install using the Powershell CLI commands, see Install the ASAv on Hyper-V Using the Command Line, page 46. For instructions to install using the Hyper-V Manager, see Install the ASAv on Hyper-V Using the Hyper-V Manager, page 46. Hyper-V does not provide a serial console option. You can manage Hyper-V through SSH or ASDM over the management interface. See Configuring SSH, page 54 for information to set up SSH.

The following figure shows the recommended topology for the ASAv in Routed Firewall Mode. There are three subnets set up in Hyper-V for the ASAv—management, inside, and outside.

Figure 37: Recommended Topology for the ASAv in Routed Firewall Mode



Guidelines and Limitations

- Platform Support
  - Cisco UCS B-Series servers
  - Cisco UCS C-Series servers
  - Hewlett Packard Proliant DL160 Gen8
- OS Support
  - Windows Server 2012
  - Native Hyper-V

Note The ASAv should run on most modern, 64-bit high-powered platforms used for virtualization today.

- File format

  Supports the VHDX format for initial deployment of the ASAv on Hyper-V.
- Day 0 configuration

You create a text file that contains the ASA CLI configuration commands that you need. See Prepare the Day 0 Configuration File for the procedure.

- Firewall Transparent Mode with Day 0 configuration

  The configuration line 'firewall transparent' must be at the top of the day 0 configuration file; if is appears anywhere else in the file, you could experience erratic behavior. See Prepare the Day 0 Configuration File for the procedure.

- Failover

  The ASAv on Hyper-V supports Active/Standby failover. For Active/Standby failover in both routed mode and transparent mode you must enable MAC Address spoofing on all the virtual network adapters. See Configure MAC Address Spoofing Using the Hyper-V Manager. For transparent mode in the standalone ASAv, the management interface should not have the MAC address spoofing enabled because the Active/Standby failover is not supported.

- Hyper-V supports up to eight interfaces. Management 0/0 and GigabitEthernet 0/0 through 0/6. You can use GigabitEthernet as a failover link.

- VLANs

  Use the **Set-VMNetworkAdapterVLan** Hyper-V Powershell command to set VLANs on an interface in trunk mode. You can set the NativeVlanID for the management interface as a particular VLAN or '0' for no VLAN. Trunk mode is not persistent across Hyper-V host reboots. You must reconfigure trunk mode after every reboot.

- Legacy network adapters are not supported.
- Generation 2 virtual machines are not supported.
- Microsoft Azure is not supported.

Prerequisites

- Install Hyper-V on MS Windows 2012.
- Create the Day 0 configuration text file if you are using one.

  You must add the Day 0 configuration before the ASAv is deployed for the first time; otherwise, you must perform a write erase from the ASAv to use the Day 0 configuration. See Prepare the Day 0 Configuration File for the procedure.

- Download the ASAv VHDX file from Cisco.com.

  http://www.cisco.com/go/asa-software

Note A Cisco.com login and Cisco service contract are required.

- Hyper-V switch configured with at least three subnets/VLANs.
- For Hyper-V system requirements, see Cisco ASA Compatibility.

Prepare the Day 0 Configuration File

You can prepare a Day 0 configuration file before you launch the ASAv. This file is a text file that contains the ASAv configuration that will be applied when the ASAv is launched. This initial configuration is placed into a text file named “day0-config” in a working directory you chose, and is manipulated into a day0.iso file that is mounted and read on first boot. At the minimum, the Day 0 configuration file must contain commands that will activate the management interface and set up the SSH server for public key authentication, but it can also contain a complete ASA configuration. The day0.iso file (either your custom day0.iso or the default day0.iso) must be available during first boot.

Before you begin

We are using Linux in this example, but there are similar utilities for Windows.

- To automatically license the ASAv during initial deployment, place the Smart Licensing Identity (ID) Token that you downloaded from the Cisco Smart Software Manager in a text file named ‘idtoken’ in the same directory as the Day 0 configuration file.
- If you want to deploy the ASAv in transparent mode, you must use a known running ASA config file in transparent mode as the Day 0 configuration file. This does not apply to a Day 0 configuration file for a routed firewall.
- You must add the Day 0 configuration file before you boot the ASAv for the first time. If you decide you want to use a Day 0 configuration after you have initially booted the ASAv, you must execute a **write erase** command, apply the day 0 configuration file, and then boot the ASAv.

Step 1 Enter the CLI configuration for the ASAv in a text file called “day0-config”. Add interface configurations for the three interfaces and any other configuration you want.

The fist line should begin with the ASA version. The day0-config should be a valid ASA configuration. The best way to generate the day0-config is to copy the desired parts of a running config from an existing ASA or ASAv. The order of the lines in the day0-config is important and should match the order seen in an existing show run command output.

Example:

```
ASA Version 9.5.1
!
interface management0/0
nameif management
 security-level 100
 ip address 192.168.1.2 255.255.255.0
 no shutdown
interface gigabitethernet0/0
 nameif inside
 security-level 100
 ip address 10.1.1.2 255.255.255.0
 no shutdown
interface gigabitethernet0/1
 nameif outside
 security-level 0
 ip address 198.51.100.2 255.255.255.0
 no shutdown
http server enable
http 192.168.1.0 255.255.255.0 management
crypto key generate rsa modulus 1024
username AdminUser password paSSw0rd
```

```
ssh 192.168.1.0 255.255.255.0 management
aaa authentication ssh console LOCAL
```

Step 2 (Optional) Download the Smart License identity token file issued by the Cisco Smart Software Manager to your computer.

Step 3 (Optional) Copy the ID token from the download file and put it a text file that only contains the ID token.

Step 4 (Optional) For automated licensing during initial ASAv deployment, make sure the following information is in the day0-config file:

- Management interface IP address
- (Optional) HTTP proxy to use for Smart Licensing
- A route command that enables connectivity to the HTTP proxy (if specified) or to tools.cisco.com
- A DNS server that resolves tools.cisco.com to an IP address
- Smart Licensing configuration specifying the ASAv license you are requesting
- (Optional) A unique host name to make the ASAv easier to find in CSSM

Step 5 Generate the virtual CD-ROM by converting the text file to an ISO file:

```
stack@user-ubuntu:-/KvmAsa$ sudo genisoimage -r -o day0.iso day0-config idtoken
I: input-charset not specified, using utf-8 (detected in locale settings)
Total translation table size: 0
Total rockridge attributes bytes: 252
Total directory bytes: 0
Path table size (bytes): 10
Max brk space used 0
176 extents written (0 MB)
stack@user-ubuntu:-/KvmAsa$
```

The Identity Token automatically registers the ASAv with the Smart Licensing server.

Step 6 Repeat Steps 1 through 5 to create separate default configuration files with the appropriate IP addresses for each ASAv you want to deploy.

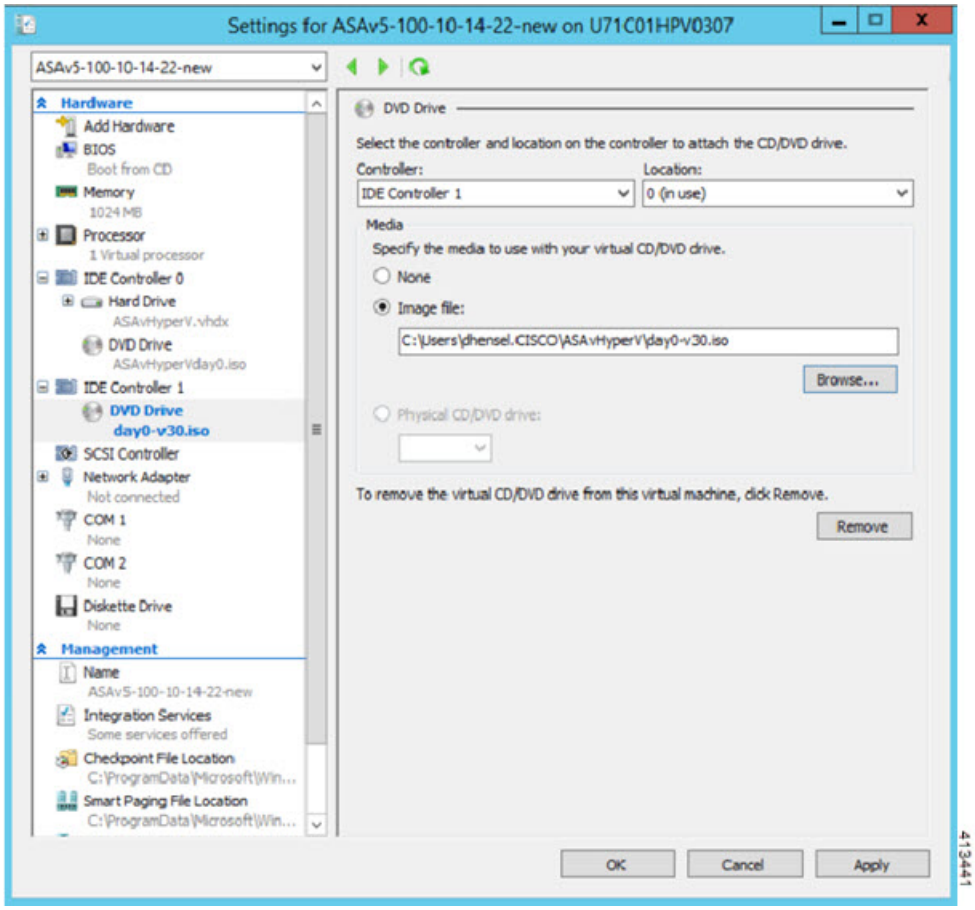
Deploy the ASAv with the Day 0 Configuration File Using the Hyper-V Manager

After you set up the Day 0 configuration file (Prepare the Day 0 Configuration File), you can deploy it using the Hyper-V Manager.

Step 1 Go to **Server Manager** > **Tools** > **Hyper-V Manager**.

Step 2 Click **Settings** on the right side of the Hyper-V Manager. The Settings dialog box opens. Under **Hardware** on the left, click **IDE Controller 1**.

Figure 38: Hyper-V Manager



Step 3 Under **Media** in the right pane, select the **Image file** radio button, and then browse to the directory where you keep your Day 0 ISO configuration file, and then click **Apply**. When you boot up your ASAv for the first time, it will be configured based on what is in the Day 0 configuration file.

Deploy the ASAv on Hyper-V Using the Command Line

You can install the ASAv on Hyper-V through the Windows Powershell command line. If you are on a standalone Hyper-V server, you must use the command line to install Hyper-V.

Step 1 Open a Windows Powershell.

Step 2 Deploy the ASAv:

Example:

```
new-vm -name $fullVMName -MemoryStartupBytes $memorysize -Generation 1 -vhdpath
C:\Users\jsmith.CISCO\ASAvHyperV\$ImageName.vhdx -Verbose
```

Step 3 Depending on your ASAv model, change the CPU count from the default of 1.

Example:

```
set-vm -Name $fullVMName -ProcessorCount 4
```

Step 4 (Optional) Change the interface name to something that makes sense to you.

Example:

```
Get-VMNetworkAdapter -VMName $fullVMName -Name "Network Adapter" | Rename-vmNetworkAdapter -NewName
mgmt
```

Step 5 (Optional) Change the VLAN ID if your network requires it.

Example:

```
Set-VMNetworkAdapterVlan -VMName $fullVMName -VlanId 1151 -Access -VMNetworkAdapterName "mgmt"
```

Step 6 Refresh the interface so that Hyper-V picks up the changes.

Example:

```
Connect-VMNetworkAdapter -VMName $fullVMName -Name "mgmt" -SwitchName 1151mgmtswitch
```

Step 7 Add the inside interface.

Example:

```
Add-VMNetworkAdapter -VMName $fullVMName -name "inside" -SwitchName 1151mgmtswitch
Set-VMNetworkAdapterVlan -VMName $fullVMName -VlanId 1552 -Access -VMNetworkAdapterName "inside"
```

Step 8 Add the outside interface.

Example:

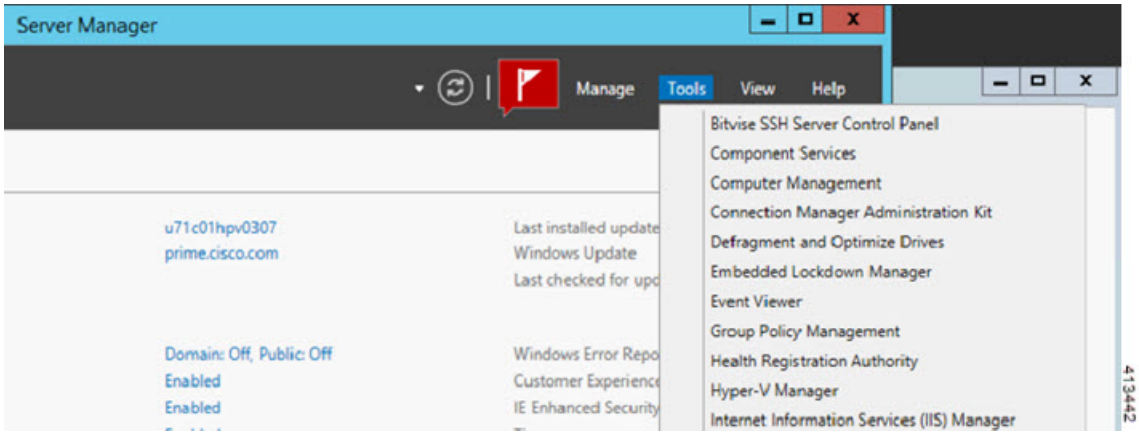
```
Add-VMNetworkAdapter -VMName $fullVMName -name "outside" -SwitchName 1151mgmtswitch
Set-VMNetworkAdapterVlan -VMName $fullVMName -VlanId 1553 -Access -VMNetworkAdapterName "outside"
```

Deploy the ASAv on Hyper-V Using the Hyper-V Manager

You can use the Hyper-V Manager to install the ASAv on Hyper-V.

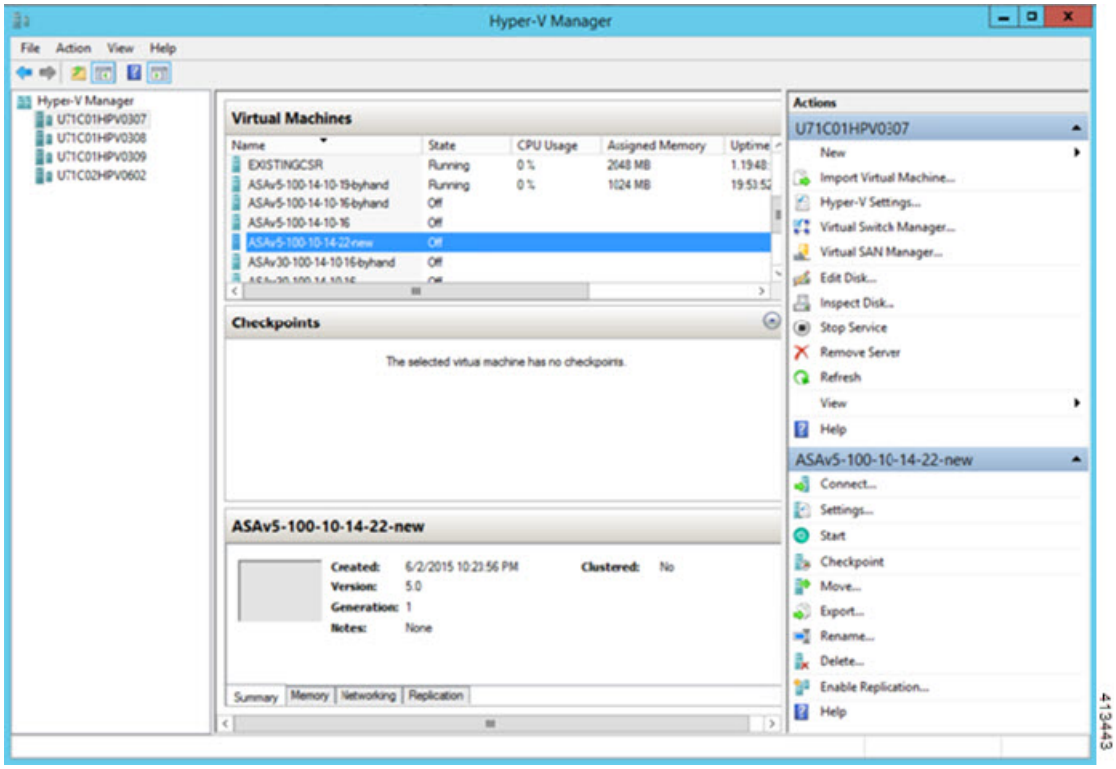
Step 1 Go to **Server Manager** > **Tools** > **Hyper-V Manager**.

Figure 39: Server Manager



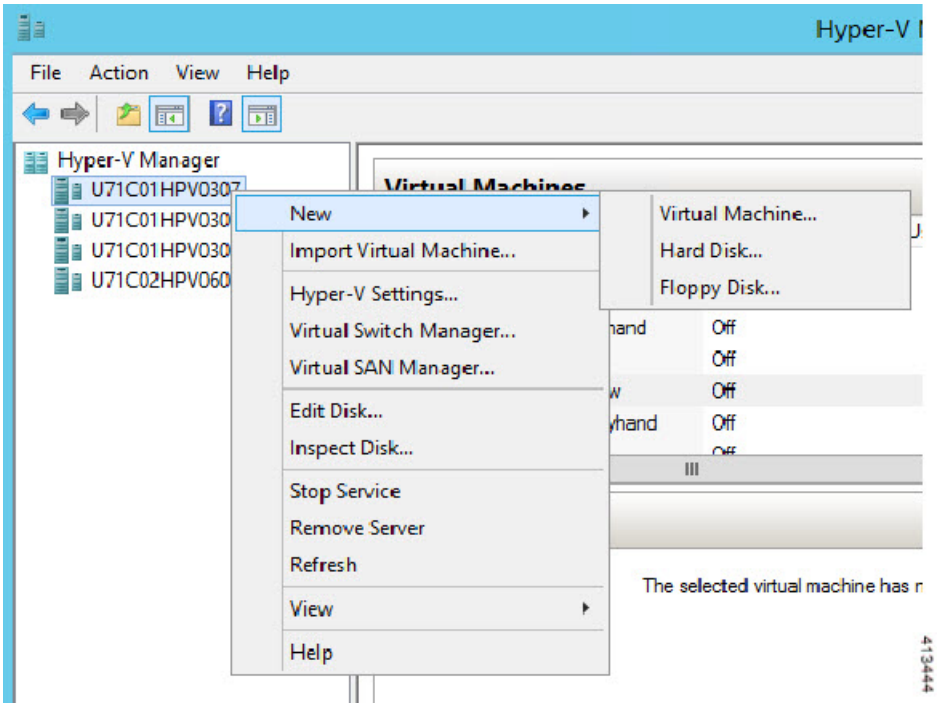
Step 2 The Hyper-V Manager appears.

Figure 40: Hyper-V Manager



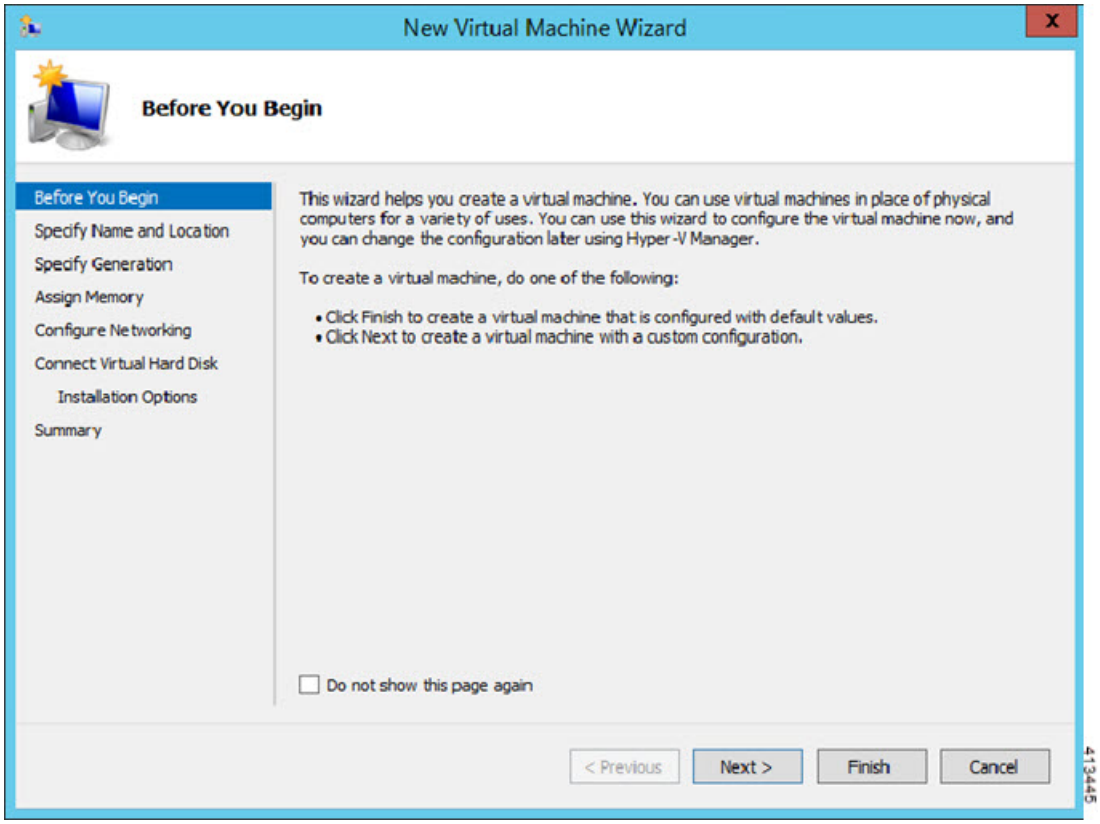
Step 3 From the list of hypervisors on the right, right-click the desired Hypervisor in the list and choose **New** > **Virtual Machine**.

Figure 41: Launch New Virtual Machine



Step 4 The New Virtual Machine Wizard appears.

Figure 42: New Virtual Machine Wizard



Step 5 Working through the wizard, specify the following information:

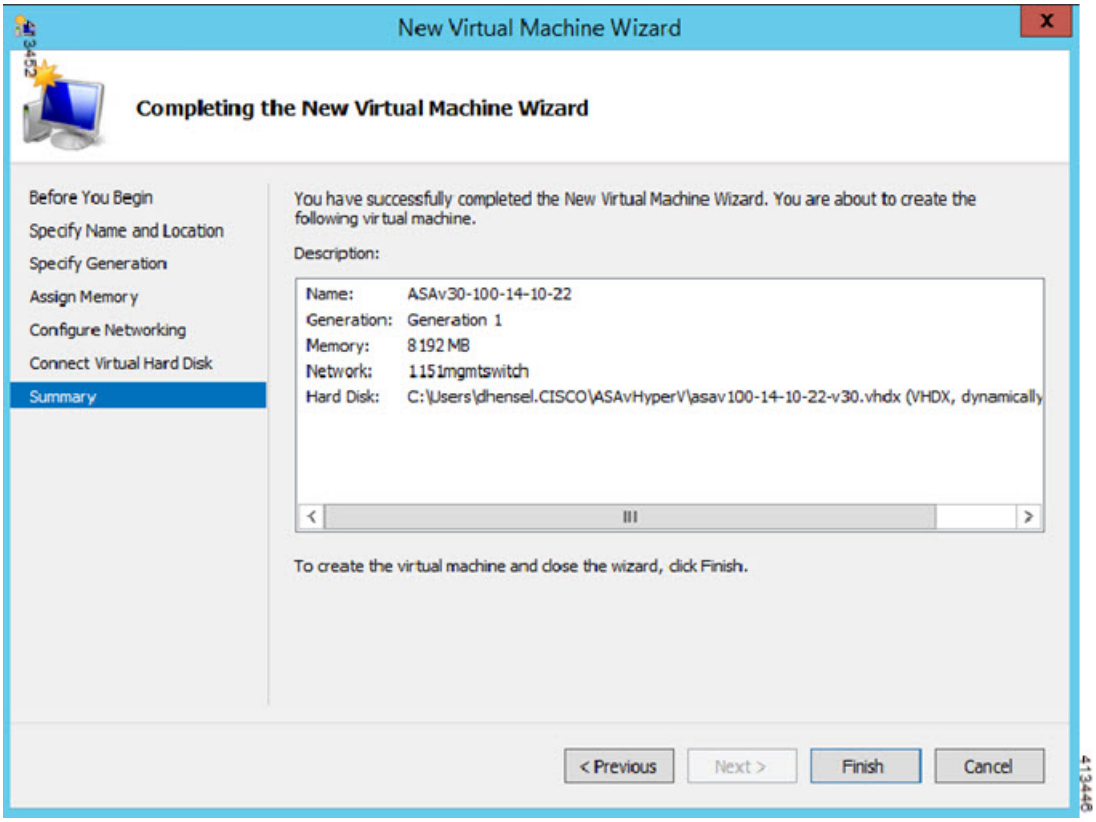
- Name and location of your ASAv
- Generation of your ASAv

  The only Generation supported for the ASAv is **Generation 1**.
- Amount of memory for your ASAv (1024 MB for 100Mbps, 2048 MB for 1Gbps, 8192 MB for 2Gbps)
- Network adapter (connect to the virtual switch you have already set up)
- Virtual hard disk and location

  Choose **Use an existing virtual hard disk** and browse to the location of your VHDX file.

Step 6 Click Finish and a dialog box appears showing your ASAv configuration.

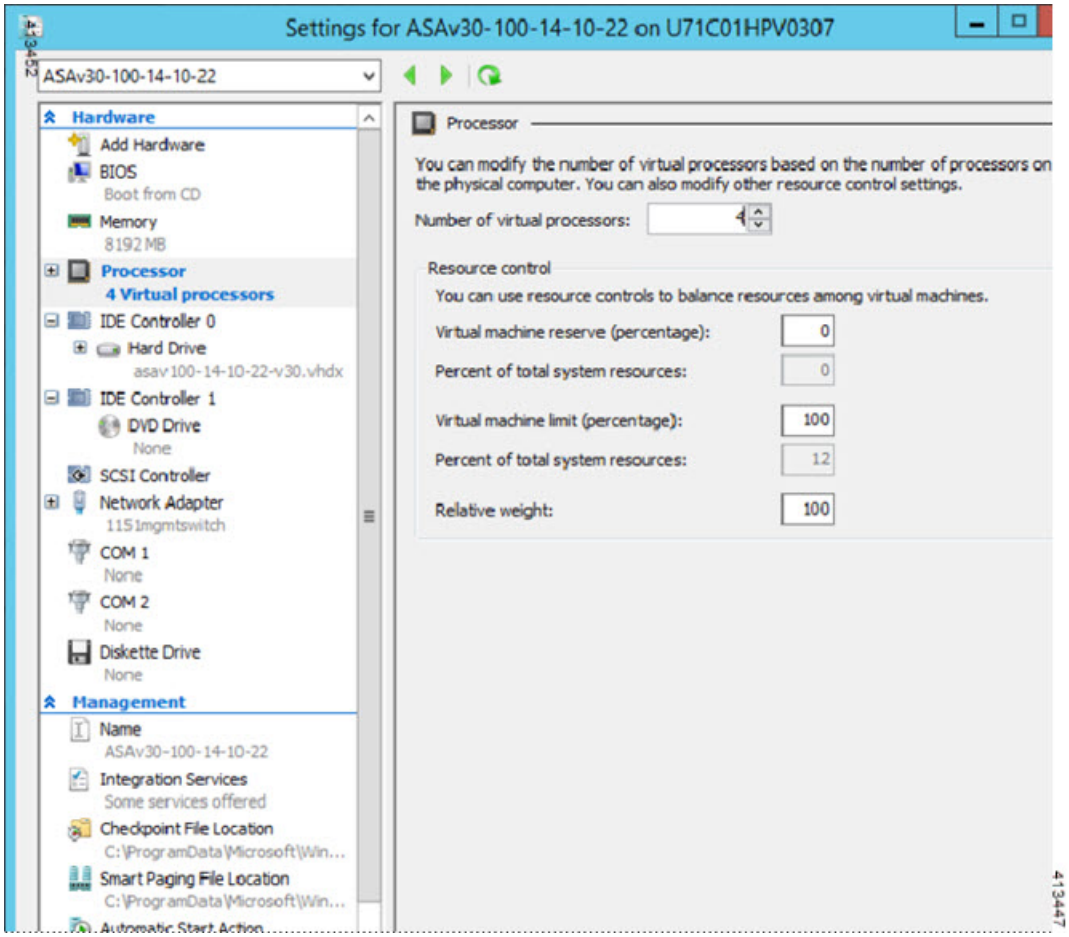
Figure 43: New Virtual Machine Summary



Step 7 If your ASAv has four vCPUs, you must modify the vCPU value before starting up your ASAv. Click **Settings** on the right side of the Hyper-V Manager. The Settings dialog box opens. Under the Hardware menu on the left, click **Processor** to get to the Processor pane. Change the **Number of virtual processors** to 4.

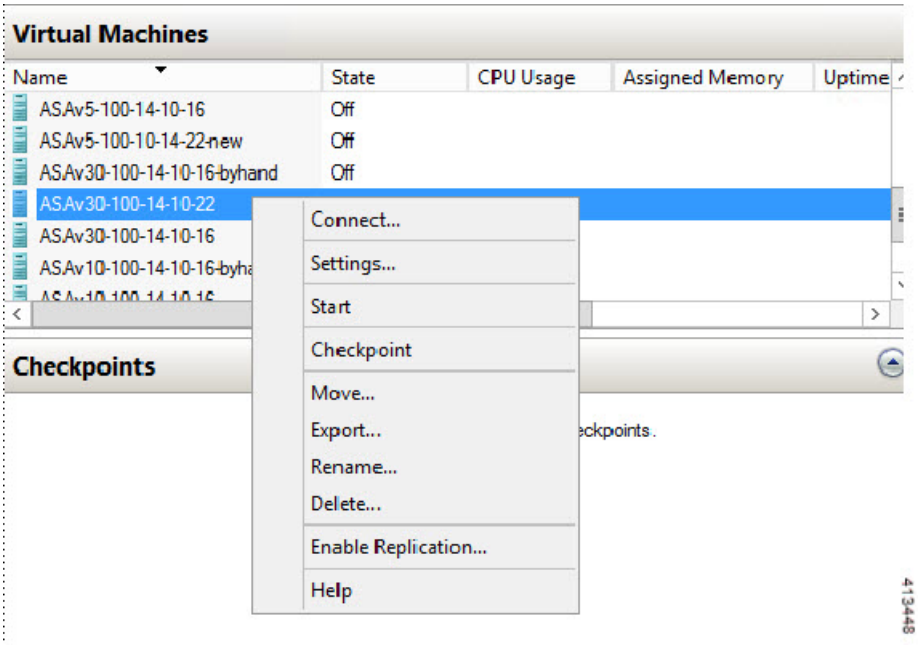
The 100Mbps and 1Gbps entitlements have one vCPU, and the 2Gbps entitlement has four vCPUs. The default is 1.

Figure 44: Virtual Machine Processor Settings



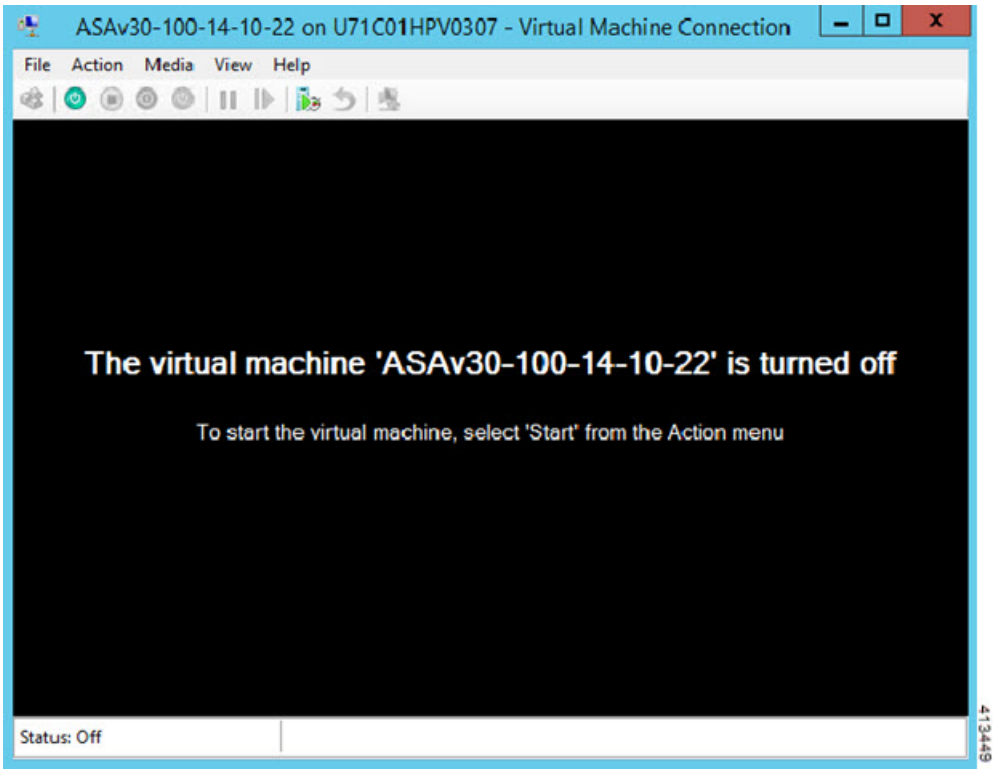
Step 8 In the Virtual Machines menu, connect to your ASAv by right-clicking on the name of the ASAv in the list and clicking **Connect**. The console opens with the stopped ASAv.

Figure 45: Connect to the Virtual Machine



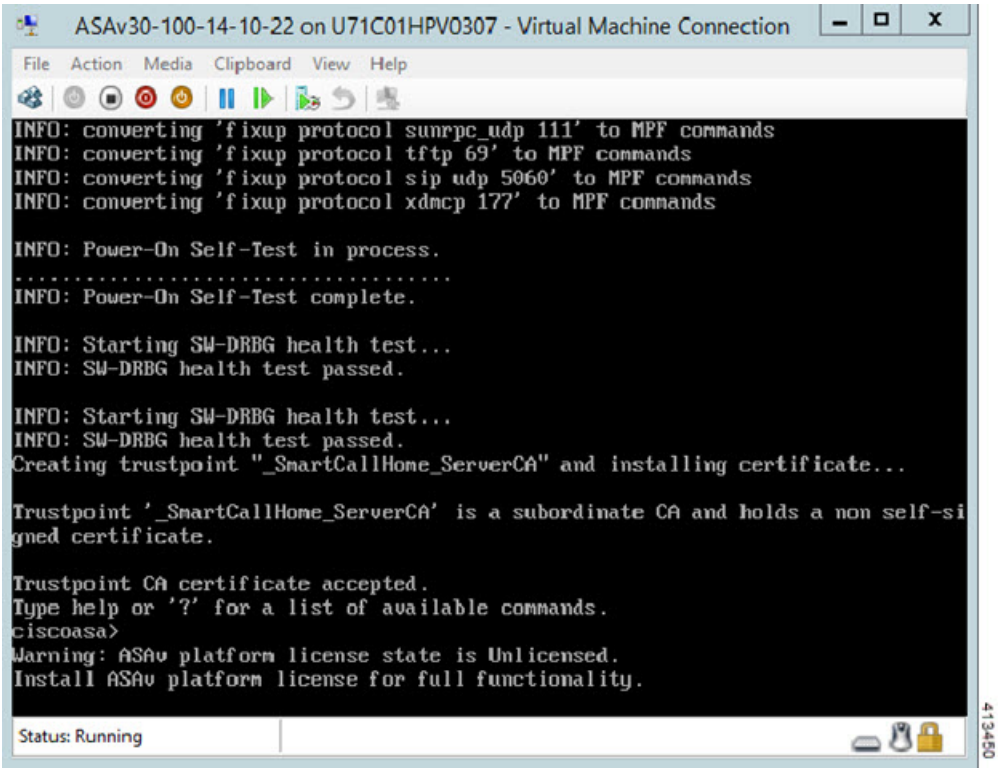
Step 9 In the Virtual Machine Connection console window, click the turquoise Start button to start the ASAv.

Figure 46: Start the Virtual Machine



Step 10 The boot progress of the ASAv is shown in the console.

Figure 47: Virtual Machine Boot Progress



Add a Network Adapter from the Hyper-V Manager

A newly deployed ASAv has only one network adapter. You need to add at least two more network adapters. In this example, we are adding the inside network adapter.

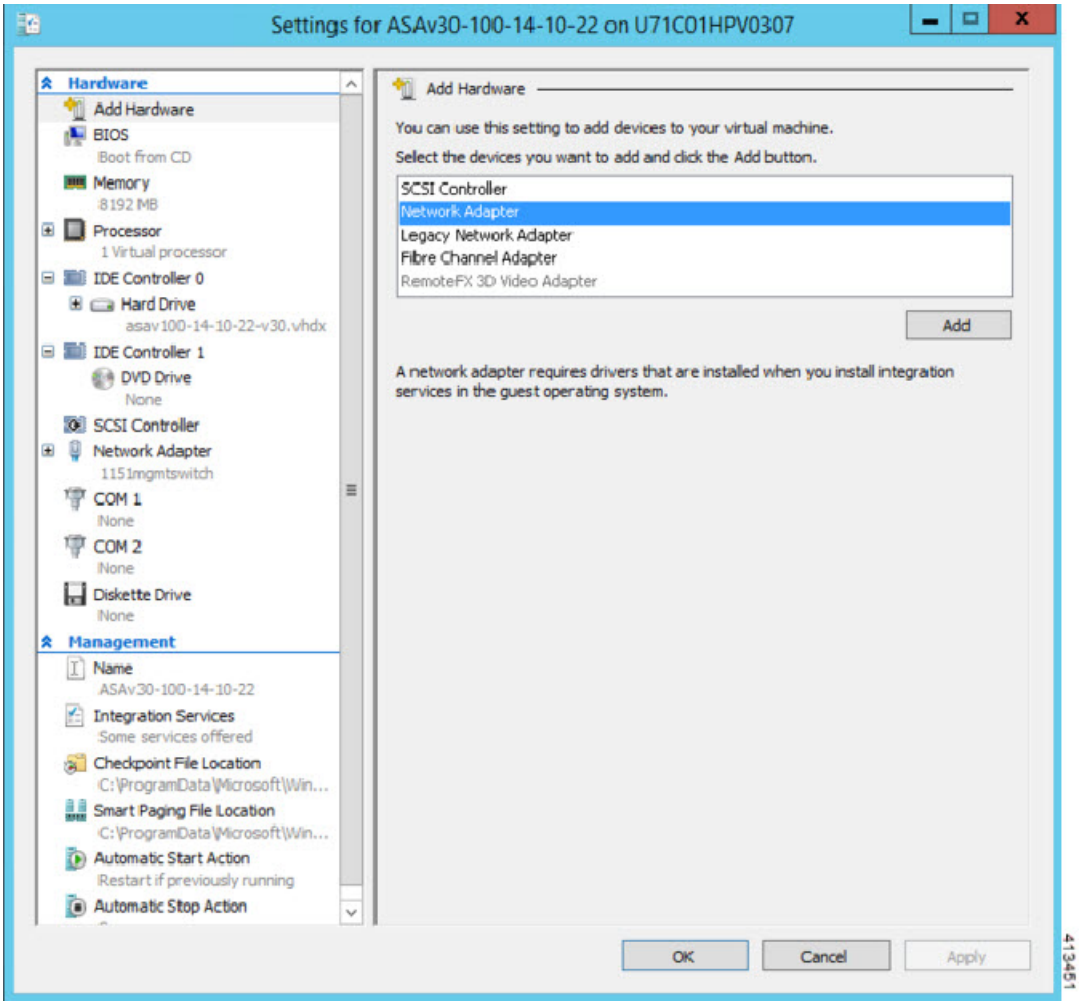
Before you begin

- The ASAv must be in the off state.

Step 1 Click **Settings** on the right side of the Hyper-V Manager. The Settings dialog box opens. Under the Hardware menu on the left, click **Add Hardware**, and then click **Network Adapter**.

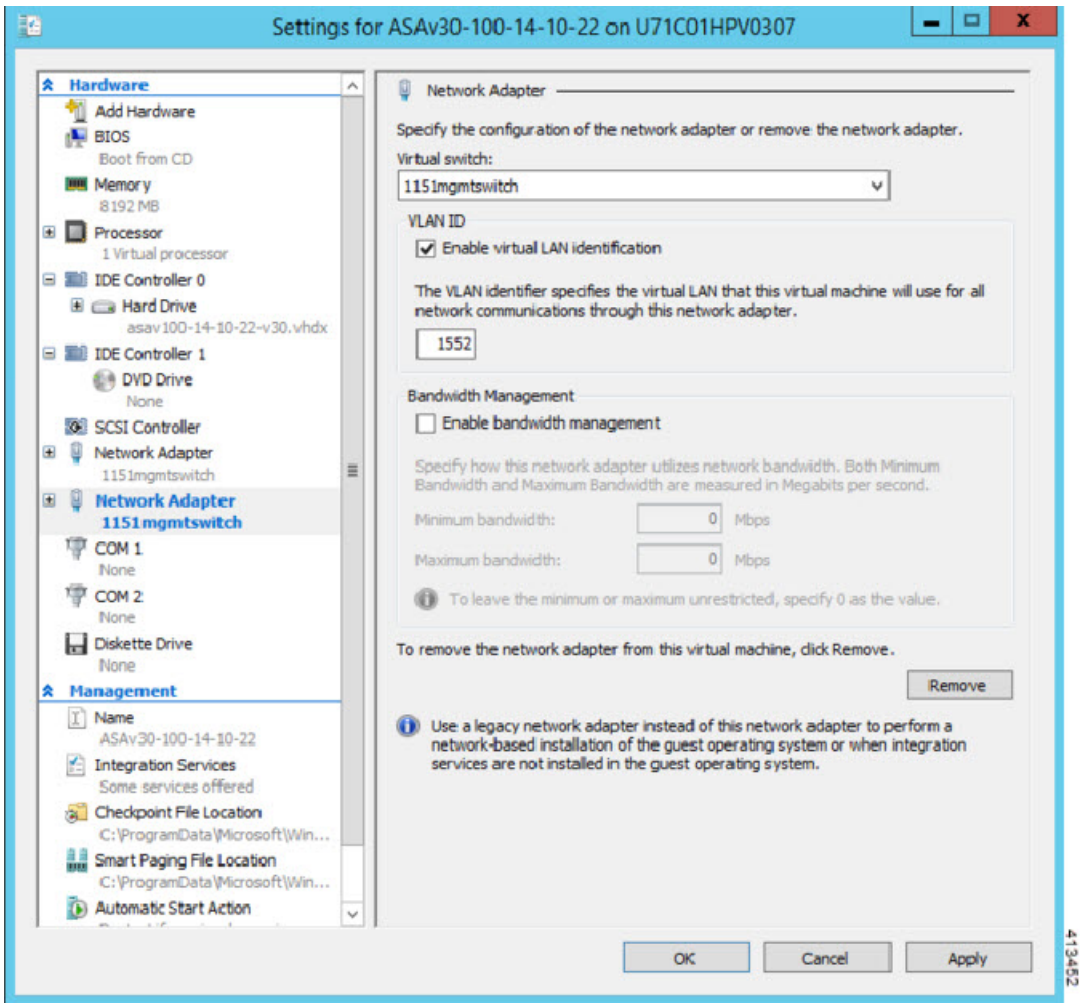
Note Do NOT use the Legacy Network Adapter.

Figure 48: Add Network Adapter



Step 2 After the network adapter has been added, you can modify the virtual switch and other features. You can also set the VLAN ID here if needed.

Figure 49: Modify Network Adapter Settings



Modify the Network Adapter Name

In Hyper-V, a generic network interface name is used, 'Network Adapter.' This can be confusing if the network interfaces all have the same name. You cannot modify the name using the Hyper-V Manager. You must modify it using the Windows Powershell commands.

Step 1 Open a Windows Powershell.

Step 2 Modify the network adapters as needed.

Example:

```
$NICRENAME= Get-VMNetworkAdapter -VMName 'ASAvVM' -Name "Network Adapter"
rename-VMNetworkAdapter -VMNetworkAdapter $NICRENAME[0] -newname inside
rename-VMNetworkAdapter -VMNetworkAdapter $NICRENAME[1] -newname outside
```

MAC Address Spoofing

For the ASAv to pass packets in transparent mode and for HA Active/Standby failover, you must turn on MAC address spoofing for ALL interfaces. You can do this in the Hyper-V Manager or using Powershell commands.

Configure MAC Address Spoofing Using the Hyper-V Manager

You can use the Hyper-V Manager to configure MAC spoofing on Hyper-V.

Step 1 Go to **Server Manager** > **Tools** > **Hyper-V Manager**.

The Hyper-V Manager appears.

Step 2 Click **Settings** on the right side of the Hyper-V Manager to open the settings dialog box.

Step 3 Under the **Hardware** menu on the left:

a. Click **Inside** and expand the menu.

b. Click **Advanced Features** to get to the MAC address option.

c. Click the **Enable MAC address spoofing** radio button.

Step 4 Repeat for the Outside interface.

Configure MAC Address Spoofing Using the Command Line

You can use the the Windows Powershell command line to configure MAC spoofing on Hyper-V.

Step 1 Open a Windows Powershell.

Step 2 Configure MAC address spoofing.

Example:

```
Set-VMNetworkAdapter -VMName $vm_name\
-ComputerName $computer_name -MacAddressSpoofing On\
-VMNetworkAdapterName $network_adapter\r"
```

Configure SSH

You can configure the ASAv for SSH access over the management interface from the Virtual Machine Connection in the Hyper-V Manager. If you are using a Day 0 configuration file, you can add SSH access to it. See Prepare the Day 0 Configuration File for more information.

Step 1 Verify that the RSA key pair is present:

Example:

```
asav# show crypto key mypubkey rsa
```

Step 2 If there is no RSA key pair, generate the RSA key pair:

Example:

```
asav(conf t)# crypto key generate rsa modulus 2048

username test password test123 privilege 15
aaa authentication ssh console LOCAL
ssh 10.7.24.0 255.255.255.0 management
ssh version 2
```

Step 3 Verify that you can access the ASAv using SSH from another PC.

CPU Usage and Reporting

The CPU Utilization report summarizes the percentage of the CPU used within the time specified. Typically, the Core operates on approximately 30 to 40 percent of total CPU capacity during nonpeak hours and approximately 60 to 70 percent capacity during peak hours.

vCPU Usage in the ASA Virtual

The ASA virtual vCPU usage shows the amount of vCPUs used for the data path, control point, and external processes.

The Hyper-V reported vCPU usage includes the ASA virtual usage as described plus:

- ASA Virtual idle time
- %SYS overhead used for the ASA virtual machine

CPU Usage Example

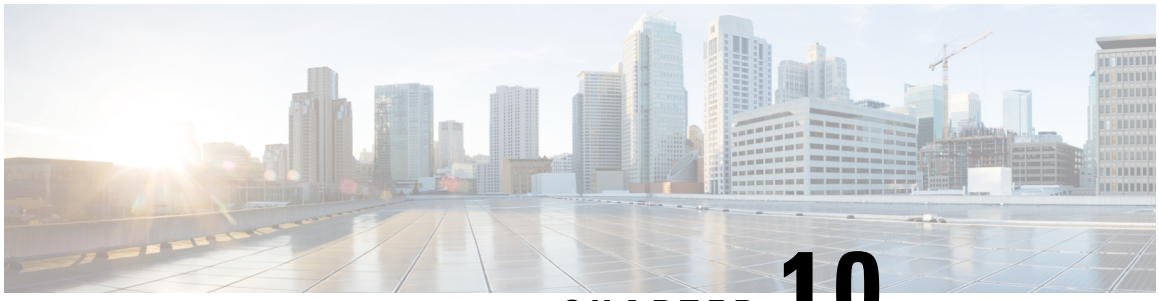
The **show cpu usage** command can be used to display CPU utilization statistics.

Example

```
Ciscoasa#show cpu usage

CPU utilization for 5 seconds = 1%; 1 minute: 2%; 5 minutes: 1%
```

The following is an example in which the reported vCPU usage is substantially different:

- ASA Virtual reports: 40%
- DP: 35%
- External Processes: 5%
- ASA (as ASA Virtual reports): 40%
- ASA idle polling: 10%
- Overhead: 45%



CHAPTER 10

Deploy the ASAv on Oracle Cloud Infrastructure

You can deploy the ASAv on the Oracle Cloud Infrastructure (OCI).

Overview

OCI is a public cloud computing service that enables you to run your applications in a highly-available, hosted environment offered by Oracle.

The ASAv runs the same software as physical ASAvs to deliver proven security functionality in a virtual form factor. The ASAv can be deployed in the public OCI. It can then be configured to protect virtual and physical data center workloads that expand, contract, or shift their location over time.

OCI Compute Shapes

A shape is a template that determines the number of CPUs, amount of memory, and other resources that are allocated to an instance. The ASAv supports the following *Standard – General purpose* OCI shape types:

Table 23: Supported Compute Shapes for ASAv

OCI Shape	Supported ASAv Version	Attributes		Interfaces
		oCPUs	RAM (GB)	
Intel VM.Standard2.4	9.15 and later	4	60	Minimum 4, Maximum 4
IntelVM.Standard2.8	9.15 and later	8	120	Minimum 4, Maximum 8

- The ASAv requires a minimum of 3 interfaces.
- In OCI, 1 oCPU is equal to 2 vCPUs.

- The maximum supported vCPUs is 16 (8 oCPUs).

You create an account on OCI, launch a compute instance using the Cisco ASA virtual firewall (ASAv) offering on the Oracle Cloud Marketplace, and choose an OCI shape.

Prerequisites

- Create an account on https://www.oracle.com/cloud/sign-in.html.
- License the ASAv. Until you license the ASAv, it will run in degraded mode, which allows only 100 connections and throughput of 100 Kbps. See Licenses: Smart Software Licensing .
- Interface requirements:
  - Management interface
  - Inside and outside interfaces
  - (Optional) Additional subnet (DMZ)
- Communications paths:
  - Management interface—Used to connect the ASAv to the ASDM; can't be used for through traffic.
  - Inside interface (required)—Used to connect the ASAv to inside hosts.
  - Outside interface (required)—Used to connect the ASAv to the public network.
  - DMZ interface (optional)—Used to connect the ASAv to the DMZ network.
- For ASAv system requirements, see Cisco ASA Compatibility.

Guidelines and Limitations

Supported Features

The ASAv on OCI supports the following features:

- Deployment in the OCI Virtual Cloud Network (VCN)
- Maximum of 16 vCPUs (8 oCPUs) per instance
- Routed mode (default)
- Licensing – Only BYOL is supported
- Single Root I/O Virtualization (SR-IOV) is supported

Performance Tiers for ASAv Smart Licensing

The ASAv supports performance-tiered licensing that provides different throughput levels and VPN connection limits based on deployment requirements.

Performance Tier	Instance Type (Core/RAM)	Rate Limit	RA VPN Session Limit
ASAv5	VM.Standard2.4 4 core/60 GB	100 Mbps	50
ASAv10	VM.Standard2.4 4 core/60 GB	1 Gbps	250
ASAv30	VM.Standard2.4 4 core/60 GB	2 Gbps	750
ASAv50	VM.Standard2.8 8 core/120 GB	NA	10,000
ASAv100	VM.Standard2.8 8 core/120 GB	NA	20,000

Unsupported Features

The ASAv on OCI does not support the following:

- ASAv native HA
- Transparent/inline/passive modes
- Multi-context mode
- IPv6

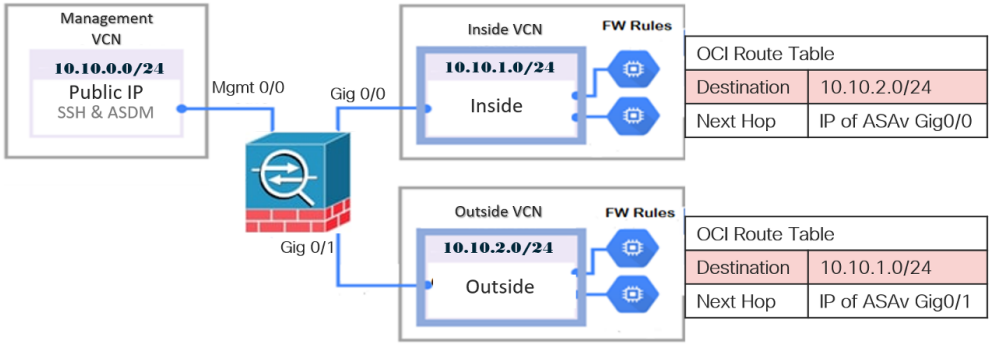
Limitations

- ASAv deployment on OCI does not support Mellanox 5 as vNICs in the SR-IOV mode.
- Separate routing rules required for ASAv for both static and DHCPconfiguration.

Sample Network Topology

The following figure shows the recommended network topology for the ASAv in Routed Firewall Mode with 3 subnets configured in OCI for the ASAv (management, inside, and outside).

Figure 50: Sample ASAv on OCI Deployment



Deploy the ASAv

The following procedures describe how to prepare your OCI environment and launch the ASAv instance. You log into the OCI portal, search the OCI Marketplace for the Cisco ASA virtual firewall (ASAv) offering, and launch the compute instance. After launching the ASAv, you must configure route tables to direct traffic to the firewall depending on the traffic's source and destination.

Create the Virtual Cloud Network (VCN)

You configure the Virtual Cloud Network (VCN) for your ASAv deployment. At a minimum, you need three VCNs, one for each interface of the ASAv.

You can continue with the following procedures to complete the Management VCN. Then you return to **Networking** to create VCNs for the inside and outside interfaces.

Before you begin

Note After you select a service from the navigation menu, the menu on the left includes the compartments list. Compartments help you organize resources to make it easier to control access to them. Your root compartment is created for you by Oracle when your tenancy is provisioned. An administrator can create more compartments in the root compartment and then add the access rules to control which users can see and take action in them. See the Oracle document "Managing Compartments" for more information.

Step 1 Log into OCI and choose your region.

OCI is divided into multiple regions that are isolated from each other. The region is displayed in the upper right corner of your screen. Resources in one region do not appear in another region. Check periodically to make sure you are in the intended region.

Step 2 Choose **Networking** > **Virtual Cloud Networks** and click Create Virtual Cloud Networks.

Step 3 Enter a descriptive **Name** for your VCN, for example *ASAvManagement*.

Step 4 Enter a **CIDR block** for your VCN.

Step 5 Click **Create VCN**.

Create the Network Security Group

A network security group consists of a set of vNICs and a set of security rules that apply to those vNICs.

Step 1 Choose **Networking** > **Virtual Cloud Networks** > **Virtual Cloud Network Details** > **Network Security Groups**, and click **Create Network Security Group**.

Step 2 Enter a descriptive **Name** for your Network Security Group, for example *ASAv-Mgmt-Allow-22-443*.

Step 3 Click **Next**.

Step 4 Add your security rules:

a) Add a rule to allow TCP port 22 for SSH Access to ASAv console.

b) Add a rule to allow TCP port 443 for HTTPS Access to ASDM.

The ASAv can be managed via ASDM, which requires port 443 to be opened for HTTPS connections.

Step 5 Click **Create**.

Create the Internet Gateway

An Internet gateway is required to make your management subnet publicly accessible.

Step 1 Choose **Networking** > **Virtual Cloud Networks** > **Virtual Cloud Network Details** > **Internet Gateways**, and click **Create Internet Gateway**.

Step 2 Enter a descriptive **Name** for your Internet gateway, for example, *ASAv-IG*.

Step 3 Click **Create Internet Gateway**.

Step 4 Add the route to the Internet Gateway:

a) Choose **Networking** > **Virtual Cloud Networks** > **Virtual Cloud Network Details** > **Route Tables**.

b) Click on the link for your default route table to add route rules.

c) Click **Add Route Rules**.

d) From the **Target Type** drop-down, select **Internet Gateway**.

e) Enter the Destination IPv4 CIDR Block, for example 0.0.0.0/0.

f) From the **Target Internet Gateway** drop-down, select the gateway you created.

g) Click **Add Route Rules**.

Create the Subnet

Each VCN will have one subnet, at a minimum. You'll create a Management subnet for the Management VCN. You'll also need an Inside subnet for the Inside VCN, and an Outside subnet for the Outside VCN.

Step 1 Choose **Networking** > **Virtual Cloud Networks** > **Virtual Cloud Network Details** > **Subnets**, and click **Create Subnet**.

Step 2 Enter a descriptive **Name** for your subnet, for example, *Management*.

Step 3 Select a **Subnet Type** (leave the recommended default of **Regional**).

Step 4 Enter a **CIDR Block**, for example 10.10.0.0/24. The internal (non-public) IP address for the subnet is taken from this CIDR block.

Step 5 Select one of the route tables you created previously from the **Route Table** drop-down.

Step 6 Select the **Subnet Access** for your subnet.

For the Management subnet, this must be **Public Subnet**.

Step 7 Select the **DHCP Option**.

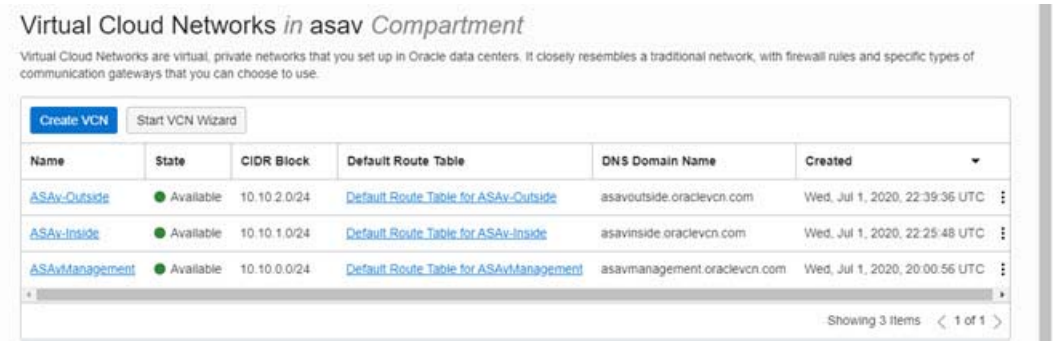
Step 8 Select a **Security List** that you created previously.

Step 9 Click **Create Subnet**.

What to do next

After you configure your VCNs (Management, Inside, Outside) you are ready to launch the ASAv. See the following figure for an example of the ASAv VCN configuration.

Figure 51: ASAv Cloud Networks



Create the ASAv Instance on OCI

You deploy the ASAv on OCI via a Compute instance using the Cisco ASA virtual firewall (ASAv) offering on the Oracle Cloud Marketplace. You select the most appropriate machine shape based on characteristics such as the number of CPUs, amount of memory, and network resources.

Step 1 Log into the OCI portal.

The region is displayed in the upper right corner of your screen. Make sure you are in the intended region.

Step 2 Choose **Marketplace** > **Applications**.

Step 3 Search Marketplace for “Cisco ASA virtual firewall (ASAv)” and choose the offering.

Step 4 Review the Terms and Conditions, and check the **I have reviewed and accept the Oracle Terms of Use and the Partner terms and conditions.**check box.

Step 5 Click **Launch Instance**.

Step 6 Enter a descriptive **Name** for your instance, for example, *ASAv-9-15*.

Step 7 Click **Change Shape** and select the shape with the number of oCPUs, the amount of RAM, and the number of interfaces required for the ASAv; for example, VM.Standard2.4 (see Table 23: Supported Compute Shapes for ASAv, on page 183).

Step 8 From the **Virtual Cloud Network** drop-down, choose the Management VCN.

Step 9 From the **Subnet** drop-down, choose the Management subnet if it's not autopopulated.

Step 10 Check **Use Network Security Groups to Control Traffic** and choose the security group you configured for the Management VCN.

Step 11 Click the **Assign a Public Ip Address** radio button.

Step 12 Under **Add SSH keys**, click the **Paste Public Keys** radio button and paste the SSH key.

Linux-based instances use an SSH key pair instead of a password to authenticate remote users. A key pair consists of a private key and public key. You keep the private key on your computer and provide the public key when you create an instance. See Managing Key Pairs on Linux Instances for guidelines.

Step 13 Click the **Show Advanced Options** link to expand the options.

Step 14 (Optional) Under **Initialization Script**, click the **Paste Cloud-Init Script** radio button to provide a day0 configuration for the ASAv. The day0 configuration is applied when the ASAv is launched.

The following example shows a sample day0 configuration you can copy and paste in the **Cloud-Init Script** field:

See the ASA Configuration Guides and the ASA Command Reference for complete information on the ASA commands.

Important When you copy text from this example, you should validate the script in a third-party text editor or validation engine to prevent format errors and remove invalid Unicode characters.

```
!ASA Version 9.18.1
interface management0/0
management-only
nameif management
security-level 100
ip address dhcp setroute

no shut
!
same-security-traffic permit inter-interface
same-security-traffic permit intra-interface
!
crypto key generate rsa modulus 2048
ssh 0 0 management
ssh timeout 60
ssh version 2
username admin nopassword privilege 15
username admin attributes
service-type admin
http server enable
http 0 0 management
aaa authentication ssh console LOCAL
```

Step 15 Click **Create**.

What to do next

Monitor the ASAv instance, which shows the state as Provisioning after you click the **Create** button.

Important It's important to monitor the status. As soon as the ASAv instance goes from Provisioning to Running state you need to attach the VNICs as required before the ASAv boot completes.

Attach the Interfaces

The ASAv enters the Running state with one VNIC attached (see **Compute** > **Instances** > **Instance Details** > **Attached VNICs**). This is referred to as the Primary VNIC, and maps to the Management VCN. Before the ASAv completes the first boot, you need to attach the VNICs for the other VCN subnets you created previously (inside, outside) so that the VNICs are correctly detected on ASAv.

Step 1 Select your newly launched ASAv instance.

Step 2 Choose **Attached VNICs** > **Create VNIC**.

Step 3 Enter a descriptive **Name** for your VNIC, for example *Inside*.

Step 4 Select the VCN from the **Virtual Cloud Network** drop-down.

Step 5 Select your subnet from the **Subnet** drop-down.

Step 6 Check **Use Network Security Groups to Control Traffic** and choose the security group you configured for the selected VCN.

Step 7 Check **Skip Source Destination Check** Network Security Groups to Control Traffic.

Step 8 (Optional) Specify a **Private IP Address**. This is only required if you want to choose a particular IP for the VNIC.

If you do not specify an IP, OCI will assign an IP address from the CIDR block you assigned to the subnet.

Step 9 Click **Save Changes** to create the VNIC.

Step 10 Repeat this procedure for each VNIC your deployment requires.

Add Route Rules for the Attached VNICs

Add route table rules to the inside and outside route tables.

Step 1 Choose **Networking** > **Virtual Cloud Networks** > and click the default route table associated with the VCN (inside or outside).

Step 2 Click **Add Route Rules**.

Step 3 From the **Target Type** drop-down, select **Private IP**.

Step 4 From the **Destination Type** drop-down, select **CIDR Block**.

Step 5 Enter the **Destination IPv4 CIDR Block**, for example, 0.0.0.0/0.

Step 6 Enter the private IP address of the VNIC in the **Target Selection** field.

If you did not explicitly assign an IP address to the VNIC, you can find the auto-assigned IP address from the VNIC details (**Compute** > **Instances** > **Instance Details** > **Attached VNICs**).

Step 7 Click **Add Route Rules**.

Step 8 Repeat this procedure for each VNIC your deployment requires.

Note Separate routing rules required for ASA Virtual (Static and DHCP) configuration.

Access the ASAv Instance on OCI

You can connect to a running instance by using a Secure Shell (SSH) connection.

- Most UNIX-style systems include an SSH client by default.
- Windows 10 and Windows Server 2019 systems should include the OpenSSH client, which you'll need if you created your instance using the SSH keys generated by Oracle Cloud Infrastructure.
- For other Windows versions you can download PuTTY, the free SSH client from http://www.putty.org.

Prerequisites

You'll need the following information to connect to the instance:

- The public IP address of the instance. You can get the address from the Instance Details page in the Console. Open the navigation menu. Under **Core Infrastructure**, go to **Compute** and click **Instances**. Then, select your instance. Alternatively, you can use the Core Services API ListVnicAttachments and GetVnic operations.
- The username and password of your instance.
- The full path to the private key portion of the SSH key pair that you used when you launched the instance. For more information about key pairs, see Managing Key Pairs on Linux Instances.

Note You can log in to the ASAv instance using the credentials specified in the day0 configuration, or by using the SSH key pair you created during the instance launch.

Connect to the ASAv Instance Using SSH

To connect to the ASAv instance from a Unix-style system, log in to the instance using SSH.

Step 1 Use the following command to set the file permissions so that only you can read the file:

```
$ chmod 400 <private_key>
```

Where:

`<private_key>` is the full path and name of the file that contains the private key associated with the instance you want to access.

Step 2 Use the following SSH command to access the instance.

```
$ ssh -i <private_key> <username>@<public-ip-address>
```

Where:

<private_key> is the full path and name of the file that contains the private key associated with the instance you want to access.

<username> is the username for the ASAv instance.

<public-ip-address> is your instance IP address that you retrieved from the Console.

Connect to the ASAv Instance Using OpenSSH

To connect to the ASAv instance from a Windows system, log in to the instance using OpenSSH.

Step 1 If this is the first time you are using this key pair, you must set the file permissions so that only you can read the file.

Do the following:

a) In Windows Explorer, navigate to the private key file, right-click the file, and then click **Properties**.
b) On the **Security** tab, click **Advanced**.
c) Ensure that the **Owner** is your user account.
d) Click **Disable Inheritance**, and then select **Convert inherited permissions into explicit permissions on this object**.
e) Select each permission entry that is not your user account and click **Remove**.
f) Ensure that the access permission for your user account is **Full control**.
g) Save your changes.

Step 2 To connect to the instance, open Windows PowerShell and run the following command:

```
$ ssh –i <private_key> <username>@<public-ip-address>
```

Where:

<private_key> is the full path and name of the file that contains the private key associated with the instance you want to access.

<username> is the username for the ASAv instance.

<public-ip-address> is your instance IP address that you retrieved from the Console.

Connect to the ASAv Instance Using PuTTY

To connect to the ASAv instance from a Windows system using PuTTY:

Step 1 Open PuTTY.

Step 2 In the **Category** pane, select **Session** and enter the following:

- **Host Name (or IP address):**

  <username>@<public-ip-address>

  Where:

  <username> is the username for the ASAv instance.

`<public-ip-address>` is your instance public IP address that you retrieved from the Console.

- **Port:** 22
- **Connection type:** SSH

Step 3 In the **Category** pane, expand **Window**, and then select **Translation**.

Step 4 In the **Remote character set** drop-down list, select **UTF-8**.

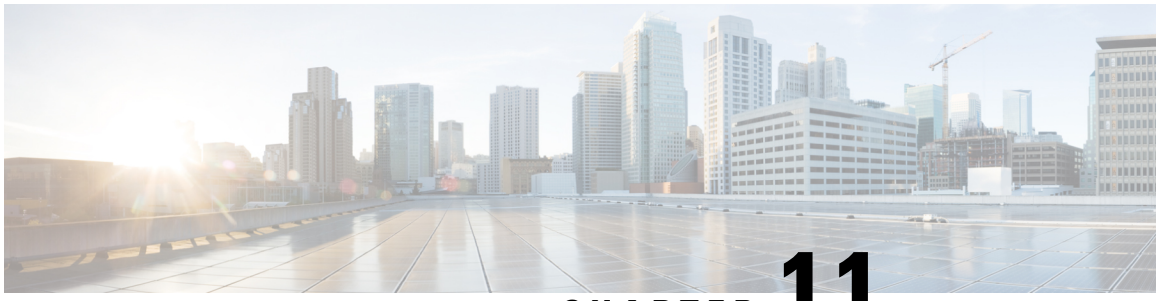
The default locale setting on Linux-based instances is UTF-8, and this configures PuTTY to use the same locale.

Step 5 In the **Category** pane, expand **Connection**, expand **SSH**, and then click **Auth**.

Step 6 Click **Browse**, and then select your private key.

Step 7 Click **Open** to start the session.

If this is your first time connecting to the instance, you might see a message that the server's host key is not cached in the registry. Click **Yes** to continue the connection.



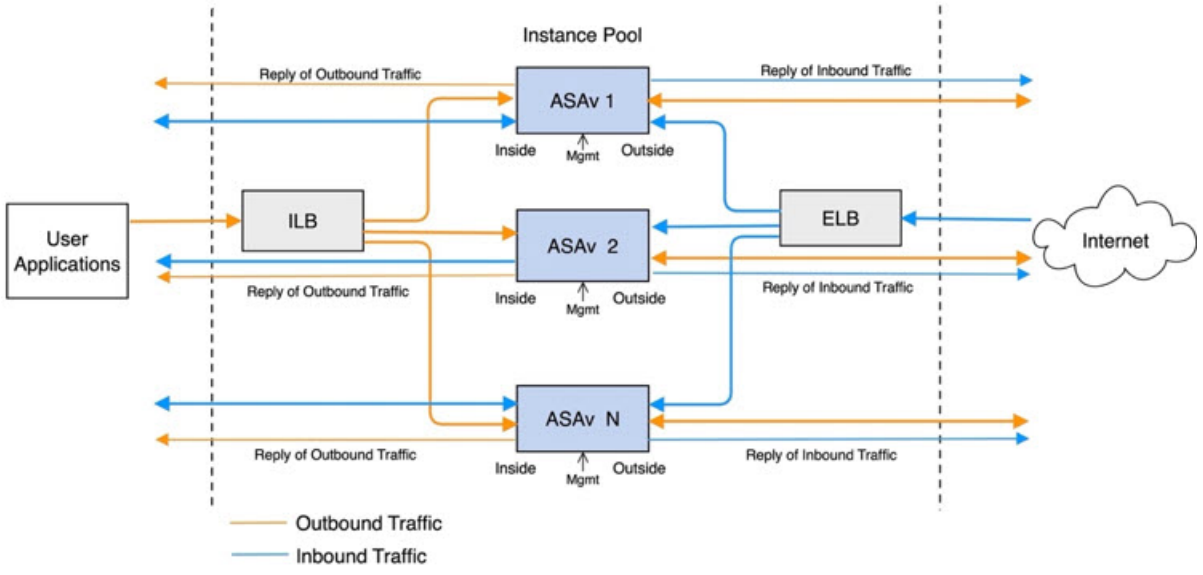
CHAPTER 11

Deploy the ASAv Auto Scale Solution on OCI

Use Case

The use case for this ASAv – OCI Autoscale solution is shown in the Use Case diagram. Internet-facing load balancer has public ip address with ports enabled using Listener and Target Group combination.

Figure 52: Use Case Diagram



Port based bi-furcation can be implemented for network traffic. This can be achieved through NAT rules. This configuration example is explained in the following sections.

Prerequisites

Permission and Policies

Following are the OCI permissions and policies that you require to implement the solution:

1. **Users and Group**

Note You must be an OCI User or a Tenancy Administrator to create the Users and Groups.

Create Oracle Cloud Infrastructure user accounts and a group to which the user accounts belong. If the relevant group with user accounts exist, you need not create them. For instructions on creating users and groups, see Creating Groups and Users.

2. **Group Policies**

You need to create the policies and then map them to the group. To create the policies, go to **OCI** > **Identity & Security** > **Policies** > **Create Policy**. Create and add the following policies to the desired group:

- Allow group <*Group_Name*> to use metrics in compartment <*Compartment_Name*>
- Allow group <*Group_Name*> to manage alarms in compartment <*Compartment_Name*>
- Allow group <*Group_Name*> to manage ons-topics in compartment <*Compartment_Name*>
- Allow group <*Group_Name*> to inspect metrics in compartment <*Compartment_Name*>
- Allow group <*Group_Name*> to read metrics in compartment <*Compartment_Name*>
- Allow group <*Group_Name*> to use tag-namespaces in compartment <*Compartment_Name*>
- Allow group <*Group_Name*> to read log-groups in compartment <*Compartment_Name*>
- Allow group <*Group_Name*> to use instance-pools compartment <*Compartment_Name*>
- Allow group <*Group_Name*> to use cloud-shell in tenancy
- Allow group <*Group_Name*> to read objectstorage-namespace in tenancy
- Allow group <*Group_Name*> to manage repos in tenancy

Note You can create policies at tenancy level as well. It is at your discretion how you want to provide all the permissions.

3. **Permission for Oracle Functions**

To enable a Oracle-Function to access another Oracle Cloud Infrastructure resource, include the function in a dynamic group, and then create a policy to grant the dynamic group access to that resource.

4. **Create Dynamic Group**

To create dynamic groups, go to **OCI** > **Identity & Security** > **Dynamic Group** > **Create Dynamic Group**

Specify the following rule while creating the dynamic group:

```
ALL {resource.type = 'fnfunc', resource.compartment.id = '<Your_Compartment_OCID>'}
```

For more details on dynamic groups, see:

- https://docs.oracle.com/en-us/iaas/Content/Functions/Tasks/functionsaccessingociresources.htm
- https://docs.oracle.com/en-us/iaas/Content/Identity/Tasks/managingdynamicgroups.htm

5. **Create Policy for Dynamic Group**

   To add policy, go to **OCI** > **Identity & Security** > **Policies** > **Create Policy**. Add the following policy to the group:

   ```
   Allow dynamic-group <Dynamic_Group_Name> to manage all-resources in compartment
   <Compartment_OCID>
   ```

Download files from GitHub

ASAv – OCI Autoscale solution is delivered as a GitHub repository. You can pull or download the files from the repository.

Python3 Environment

A *make.py* file can be found in the cloned repository. This program compresses the oracle functions and template files into a Zip file; copy them to a target folder. In order to do these tasks, the Python 3 environment should be configured.

Note This python script can be used only on Linux environment.

Infrastructure Configuration

The following must be configured:

1. **VCN**

   Create VCN as required for your ASAv application. Create VCN with the Internet Gateway having at least one of the subnet attached with route to internet.

   For information on creating VCN, see https://docs.oracle.com/en-us/iaas/Content/GSG/Tasks/creatingnetwork.htm.

2. **Application Subnets**

   Create subnets as required for your ASAv application. To implement the solution as per this use case, ASAv instance requires 3 subnets for its operation.

   For information on creating subnet, see https://docs.oracle.com/en-us/iaas/Content/Network/Tasks/managingVCNs_topic-Overview_of_VCNs_and_Subnets.htm#.

3. **Outside Subnet**

Subnet should have route with '0.0.0.0/0' to Internet Gateway. This subnet contains the Outside interface of Cisco ASAv and the Internet-facing Load balancer. Ensure that the NAT Gateway is added for outbound traffic.

For more information, see the following documents:

- https://docs.oracle.com/en-us/iaas/Content/Network/Tasks/managingIGs.htm
- https://docs.oracle.com/en-us/iaas/Content/Network/Tasks/NATgateway.htm#To_create_a_NAT_gateway

4. **Inside Subnet**

   This is similar to the Application Subnets, with or without NAT/Internet gateway.

Note For ASAv health probes, you can reach the metadata server (169.254.169.254) through Port 80.

5. **Management Subnet**

   Management subnet should be public so that it supports SSH accessibility to the ASAv.

6. **Security Groups- Network Security Group for ASAv Instance**

   Configure the security group for ASAv instances that addresses the following requirements:

   - The Oracle Functions(in same VCN) perform SSH connections to ASAv's management address.
   - Admin hosts might need SSH access to ASAv instances.
   - ASAv initiates communication with CSSM/Satellite servers for licensing.

7. **Object Storage Namespace**

   This object storage namespace is used for hosting static website, having configuration.txt file. You must create a pre-authenticated requests for the configuration.txt file. This pre-authenticated URL is used during the template deployment.

Note Ensure that the following configurations that are uploaded are accessible by the ASAv instances through HTTP URL.

When ASAv is booted, it executes the following command`$ copy /noconfirm <configuration.txt file's pre-authenticated request URL > disk0:Connfiguration.txt`

This command enables ASAv launch to be configured with configuration.txt file.

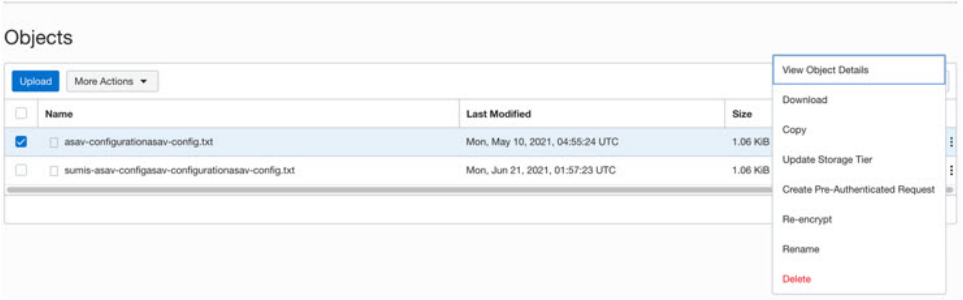
8. **Upload configuration.txt file**

   To create a pre-authenticated request URL of the ASAv config file:

   a. Click **Buckets** > **Create Bucket**.

   b. Click **Upload**.

   c. When the config file is uploaded, choose **Create Pre-Authenticated Request** as shown in the figure below.



Note The config file should be accessible from the oracle-function now.

Network Configuration

1. **Inbound traffic**

   Make sure that <*Application VM IP*> address is correct in configuration.txt as mentioned in Step 2.

2. **Outbound Traffic**

   - Make sure that <*External Server IP*> address is correct in configuration.txt. as mentioned in Step 2.
   - Make sure there is one NAT Gateway in your Outside VCN.
   - Make sure to add same <*External Server IP*> address in route table of your Outside VCN, destined through NAT gateway, as shown in the example figure below:

Destination	Target Type	Target
0.0.0.0/0	Internet Gateway	outside-ig
8.8.8.8/32	NAT Gateway	nat-gw

Encrypt Password

Note For more information on this procedure, see Create Vaults and Secrets.

Password for ASAv is used to configure all the ASAv instances being used while autoscaling and it is used to retrieve the CPU usage data of the ASAv instances.

Therefore, you need to save and process the password every now and then. Owing to the frequent changes and vulnerability, editing or saving the password in the plain-text format is not allowed. Password must be in an encrypted format only.

To obtain password in encrypted form:

Step 1 Create Vault.

OCI Vault provides services to create and save master encryption keys safely, and methods for encryption and decryption in using them. So Vault should be created (if not having already) in the same compartment as the rest of the autoscale solution.

Go to **OCI** > **Identity & Security** > **Vault** > **Choose or Create New Vault**

.

Step 2 Create Master Encryption Key.

One master encryption key is needed to encrypt the plain text password.

Go to **OCI** > **Identity & Security** > **Vault** > **Choose or Create Key**

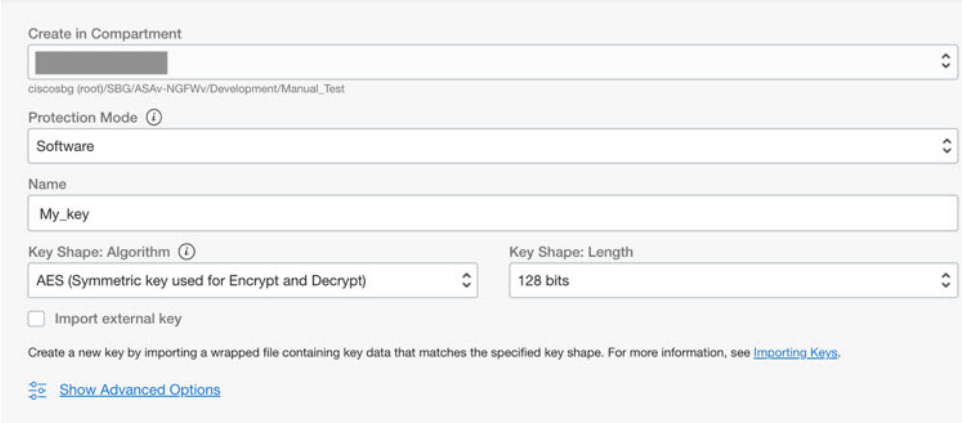
Choose any of the keys from any of the given algorithm with any bit of length.

a. AES – 128, 192, 256

b. RSA – 2048, 3072, 4096

c. ECDSA – 256, 384, 521

Figure 53: Create Key



Step 3 Create encrypted password.

a. Go to **OCI** > **Open CloudShell (OCI Cloud Terminal)**

b. Execute following command by replacing <*Password*> as your password.

```
echo -n '<Password>' | base64
```

c. From the selected Vault, copy cryptographic endpoint and master encryption key OCID. Replace the following values, and then execute the encrypt command:

- KEY_OCID with Your key's OCID
- Cryptographic_Endpoint_URL with Your vault's cryptographic endpoint URL
- Password with Your password

Encrypt Command

```
oci kms crypto encrypt --key-id Key_OCID --endpoint

Cryptographic_Endpoint_URL --plaintext <base64-value-of-password>
```

d. Copy ciphertext from output of the above command and use it as required.

Preparation of the ASA Configuration File

Ensure that the Application is either deployed or its deployment plan is available.

Step 1 Collect the following input parameters before deployment:

Parameter	Data Type	Description
tenancy_ocid	String	OCID of the tenancy to which your account belongs. To know how to find your tenancy OCID, see here. The tenancy OCID looks something like this - `ocid1.tenancy.oc1..<unique_ID>`
compartment_id	String	The OCID of the compartment in which to create the resources. Example: `ocid1.compartment.oc1..<unique_ID>`
compartment_name	String	Name of the compartment
region	String	The unique identifier of the region in which you want the resources to be created. Example: `us-phoenix-1, us-ashburn-1`
lb_size	String	A template that determines the total pre-provisioned bandwidth (ingress plus egress) of the external and internal load balancer. Supported values: 100Mbps, 10Mbps, 10Mbps-Micro, 400Mbps, 8000Mbps Example: 100Mbps

Parameter	Data Type	Description
availability_domain	Comma separated value	Example: Tpeb:PHX-AD-1 **Note** Execute **oci iam availability-domain list** command in the Cloud Shell to get the availability domain names.
min_and_max_instance_count	comma separated value	The minimum and the maximum number of instances that you would want to retain in the instance pool. Example: 1,5
autoscale_group_prefix	String	The prefix to be used to name all the resources that are created using the template. For example, if the resource prefix is given as 'autoscale', all the resources are named as follows - autoscale_resource1, autoscale_resource2 etc.
asav_config_file_url	URL	The URL of the configuration file uploaded to the object storage to be used to configure the ASAv. **Note** Pre-Authenticated Request URL of the configuration file has to be given Example: https://objectstorage.<region-name>.oraclecloud.com/<object-storage-name>/oci-asav-configuration.txt
mgmt_subnet_ocid	String	OCID of the Management subnet that is to be used.
inside_subnet_ocid	String	OCID of the Inside subnet that is to be used.
outside_subnet_ocid	String	OCID of the Outside subnet that is to be used.
mgmt_nsg_ocid	String	OCID of the Management subnet network security group that is to be used.
inside_nsg_ocid	String	OCID of the Inside subnet network security group that is to be used.

Parameter	Data Type	Description
outside_nsg_ocid	String	OCID of the Outside subnet network security group that is to be used.
elb_listener_port	comma separated Values	List of the communication ports for the external load balancer listener. Example: 80
ilb_listener_port	comma separated Values	List of the communication ports for the internal load balancer listener. Example: 80
health_check_port	String	The backend server port of load balancer against which the health check is executed. Example: 8080
instance_shape	String	The shape of the instance to be created. The shape determines the number of CPUs, amount of memory, and other resources allocated to the instance. Supported shapes :"VM.Standard2.4" & "VM.Standard2.8"
lb_bs_policy	String	The load balancer policy to be used for the internal and external load balancer's backend set. To know more about how load balancer policies work, see here. Supported values: "ROUND_ROBIN", "LEAST_CONNECTIONS", "IP_HASH"
image_name	String	The name of the marketplace image to be used for creating the instance configuration. Default value : "Cisco ASA virtual firewall (ASAv)" **Note** If the user wants to deploy custom image, user has to configure the custom_image_ocid parameter.

Parameter	Data Type	Description
image_version	String	The Version of the ASAv image available in OCI Marketplace to be used. Currently, 9.15.1.15 and 9.16.1 versions are available. Default value : "Cisco ASA virtual firewall (ASAv)"
scaling_thresholds	Comma separated value	The CPU usage thresholds to be used for scale-in and scale-out. Specify the scale-in and scale-out threshold values as comma separated input. Example : 15,50 where, 15 is the scale-in threshold and 50 is the scale-out threshold.
custom_image_ocid	String	OCID of the custom image to be used to create instance configuration if the marketplace image is not to be used. **Note** custom_image_ocid is optional parameter
asav_password	String	The password for ASAv in the encrypted form, to SSH into the ASAv for configuration. Use configuration guide for the instructions on how to encrypt password or see here.
cryptographic_endpoint	String	Cryptographic endpoint is a URL, that is used for decrypting password. It can be found in the Vault.
master_encryption_key_id	String	The OCID of key with which the password was encrypted. It can be found in the Vault.
Profile Name		It is the User's profile name in OCI. It can be found under profile section of the user. Example: oracleidentitycloudservice/<user>@<mail>.com
Object Storage Namespace		It is unique identifier created at the time of Tenancy creation. You can find this value in **OCI** > **Administration** > **Tenancy Details**

Parameter	Data Type	Description
Authorization Token		This is used as password for docker login which authorizes to push Oracle-Functions into the OCI container registry. To procure the token, go to **OCI** > **Identity** > **Users** > **User Details** > **Auth Tokens** > **Generate Token**.

Step 2 Configure Objects, Licensing, NAT rule for Load Balancer health probes and Access Policies.

```
! Default route via outside
route outside 0.0.0.0 0.0.0.0 <Outside Subnet gateway> 2

! Health Check Configuration
object network metadata-server
host 169.254.169.254
object service health-check-port
service tcp destination eq <health-check-port>
object service http-port
service tcp destination eq <traffic port>
route inside 169.254.169.254 255.255.255.255 <Inside Subnet GW> 1

! Health check NAT
nat (outside,inside) source static any interface destination static interface metadata-server service
 health-check-port http-port
nat (inside,outside) source static any interface destination static interface metadata-server service
 health-check-port http-port

! Outbound NAT
object network inside-subnet
subnet <Inside Subnet> <Inside Subnet Gateway>
object network external-server
host <External Server IP>
nat (inside,outside) source static inside-subnet interface destination static interface external-server


! Inbound NAT
object network outside-subnet
subnet <Outside Subnet> <Outside Subnet GW>
object network http-server-80
host <Application VM IP>
nat (outside,inside) source static outside-subnet interface destination static interface http-server-80


!
dns domain-lookup outside
DNS server-group DefaultDNS

! License Configuration
call-home
profile license
destination transport-method http
destination address http <URL>
debug menu license 25 production
license smart
feature tier standard
throughput level <Entitlement>
licence smart register idtoken <License token> force
!
```

These health probe connections and data plane configuration should be allowed on Access policy.

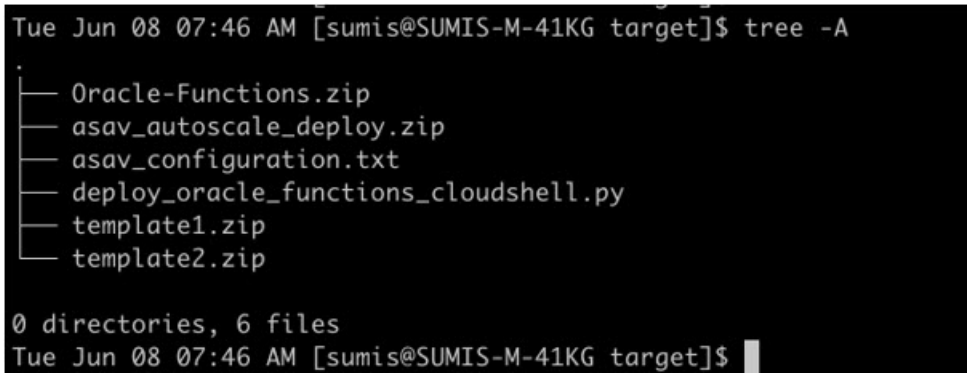
Step 3 Update *configuration.txt* file with the configuration details.

Step 4 Upload *configuration.txt* file to the user created object storage space and create the pre-authenticated request for the uploaded file.

Note Ensure that pre-authenticated request URL of configuration.txt is used in the stack deployment.

Step 5 Create Zip files.

A *make.py* file can be found in the cloned repository. Execute the `python3 make.py build` command to create the zip files. The target folder has the following files.



Note If you are deploying the autoscale solution using cloud shell, update the *easy_deploy/deployment_parameters.json* file before executing the `python3 make.py build`. For updating, refer Step 1 and Deploy Oracle Functions.

Deploy the Auto Scale Solution

After completing the pre-requisite steps for deployment, start creating the OCI stack. You can perform a Manual Deployment or perform Deploy Autoscale Using Cloud Shell. Deployment scripts and templates for your version are available in the GitHub repository.

Manual Deployment

End-to-end Autoscale solution deployment consist of three steps: Deploy Terraform Template-1 Stack , Deploy Oracle Functions, and then Deploy Terraform Template-2.

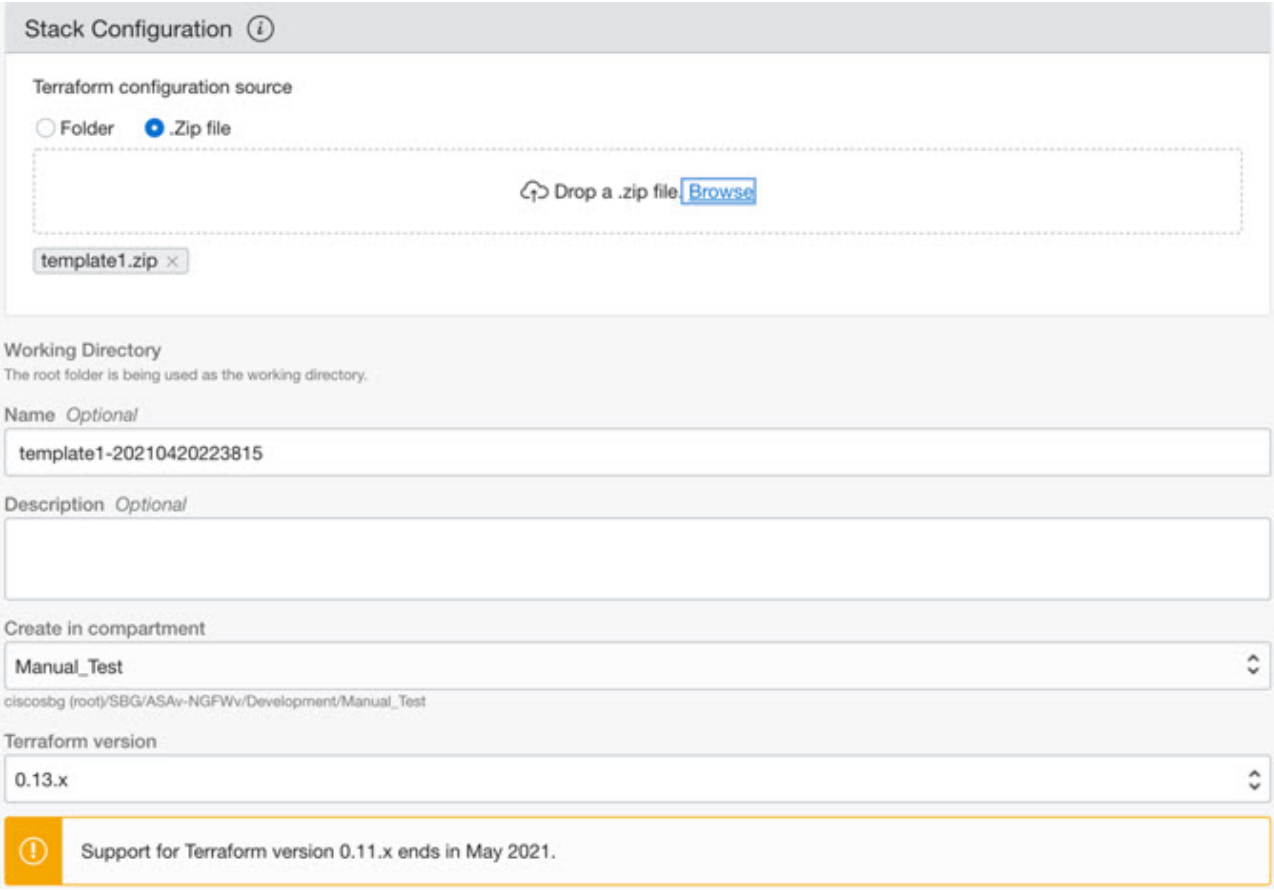
Deploy Terraform Template-1 Stack

Step 1 Log into the OCI portal.

The region is displayed in the upper right corner of your screen. Make sure you are in the intended region.

Step 2 Choose **Developer Service** > **Resource Manager** > **Stack** > **Create Stack**

Choose **My Configuration**, and then select the *Terraform template1.zip* file in the target folder as Terraform Configuration Source as shown in the figure below.



Step 3 In the **Transform version** drop-down list, select 0.13.x or 0.14.x.

Step 4 In the next step, enter all the details as collected in Step 1.

Note Enter valid input parameters, otherwise stack deployment may fail in further steps.

Step 5 In the next step, choose **Terraform Actions** > **Apply**.

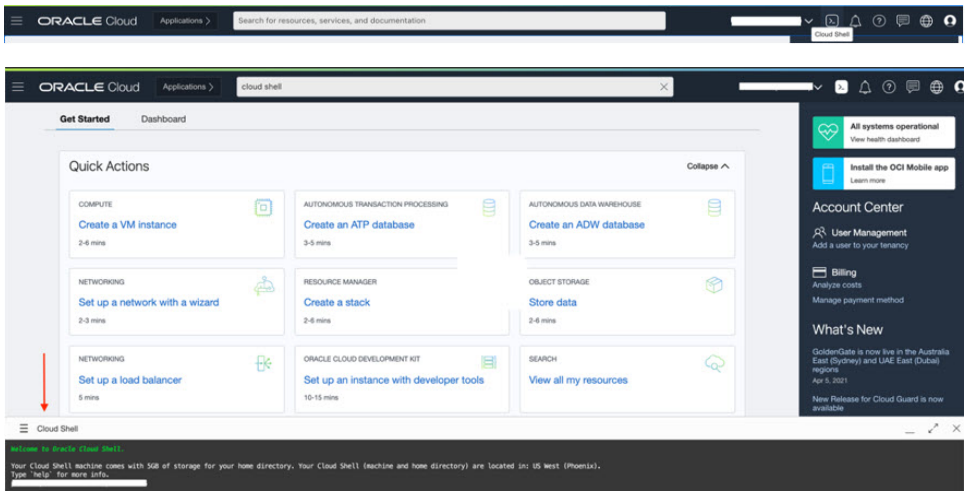
Post successful deployment, proceed to deploy the Oracle functions.

Deploy Oracle Functions

Note *This step must be performed only after successful Terraform Template-1 deployment.*

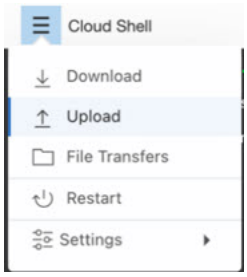
In OCI, Oracle Functions are uploaded as Docker Images, which are saved into the OCI container registry. Oracle Functions are needed to be pushed into one of the OCI Application (created in Terraform Template-1) at the time of deployment.

Step 1 Open OCI Cloud Shell.

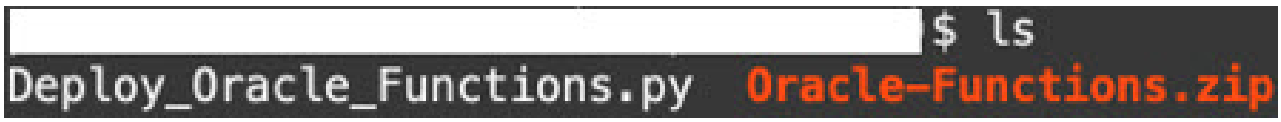


Step 2 Upload *deploy_oracle_functions_cloudshell.py* and *Oracle-Functions.zip*.

From the Cloud Shell's hamburger menu, choose **Upload**.



Step 3 Verify files using the **ls** command.



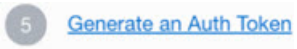
Step 4 Run `python3 Deploy_Oracle_Functions.py -h`. The `deploy_oracle_functions_cloudshell.py` script requires some input parameters whose details can be found using help argument, as shown in figure below.

```
                                $ python3 Deploy_Oracle_Functions.py -h
usage: Deploy_Oracle_Functions.py [-h] -a  -r  -p  -c  -o  -t

*** Script to deploy Oracle Function for OCI ASAv Autoscale Solution ***

Instruction to find values of required arguments:
Application Name: Name of Application created by first Terraform Template
Region Identifier: OCI -> Administration -> Region Management
Profile Name: OCI -> Profile
Compartment OCID: OCI -> Identity -> Compartment -> Compartment Details
Object Storage Namespace: OCI -> Administration -> Tenancy Details
Authorization Token: OCI -> Identity -> Users -> User Details -> Auth Tokens -> Generate Token

optional arguments:
  -h, --help  show this help message and exit
  -a          Name of Application in OCI to which functions will be deployed
  -r          Region Identifier
  -p          Profile Name of User
  -c          Compartment OCID
  -o          Object Storage Namespace
  -t          Authorization Token for Docker Login (*Please Put in Quotes)
```

To run the script pass the following arguments:

Table 24: Arguments and Details

Argument	Particulars
Application Name	It is the name of OCI Application created by Terraform Template-1 deployment. Its value is obtained by combining **"autoscale_group_prefix"** given in Template-1 and suffix **"_application".**
Region Identifier	Region identifier is the region codeword fixed in the OCI for different regions. Example: 'us-phoenix-1' for Phoenix or "ap-melbourne-1" for Melbourne. To get the list of all region with their region identifiers, go to **OCI** > **Administration** > **Region Management**.
Profile Name	It is simple User's profile name in OCI. Example: *oracleidentitycloudservice/<user>@<mail>.com* The name can be found under profile section of the user.
Compartment OCID	It is the compartment's OCID (Oracle Cloud Identifier). Compartment OCID where user have the OCI Application. Go to **OCI** > **Identity** > **Compartment** > **Compartment Details**.
Object Storage Namespace	It is unique identifier created at the time of Tenancy creation. Go to **OCI** > **Administration** > **Tenancy Details**.

Argument	Particulars
Authorization Token	This is used as password for docker login which authorizes it to push Oracle-Functions into the OCI container registry. Specify the token in quotes in the deployment script. Go to **OCI** > **Identity** > **Users** > **User Details** > **Auth Tokens** > **Generate Token**. For some reason, if you are not able to see User Details then click **Developer services** > **Functions**. Go to the application created by Terraform Template-1. Click **Getting Started**, and choose Cloud Shell Setup and among the steps you will find the link to generate auth token as shown below. 

Step 5 Run the `python3 Deploy_Oracle_Functions.py` command by passing valid input arguments. It will take some time to deploy all the functions. You can then remove the file and close the Cloud Shell.

Deploy Terraform Template-2

Template 2 deploys the resources related to alarm creation, including alarms, ONS topics for invoking function. The deployment of template 2 is similar to Terraform Template-1 deployment.

Step 1 Log into the OCI portal.

The region is displayed in the upper right corner of your screen. Make sure you are in the intended region.

Step 2 Choose **Developer Service** > **Resource Manager** > **Stack** > **Create Stack**.

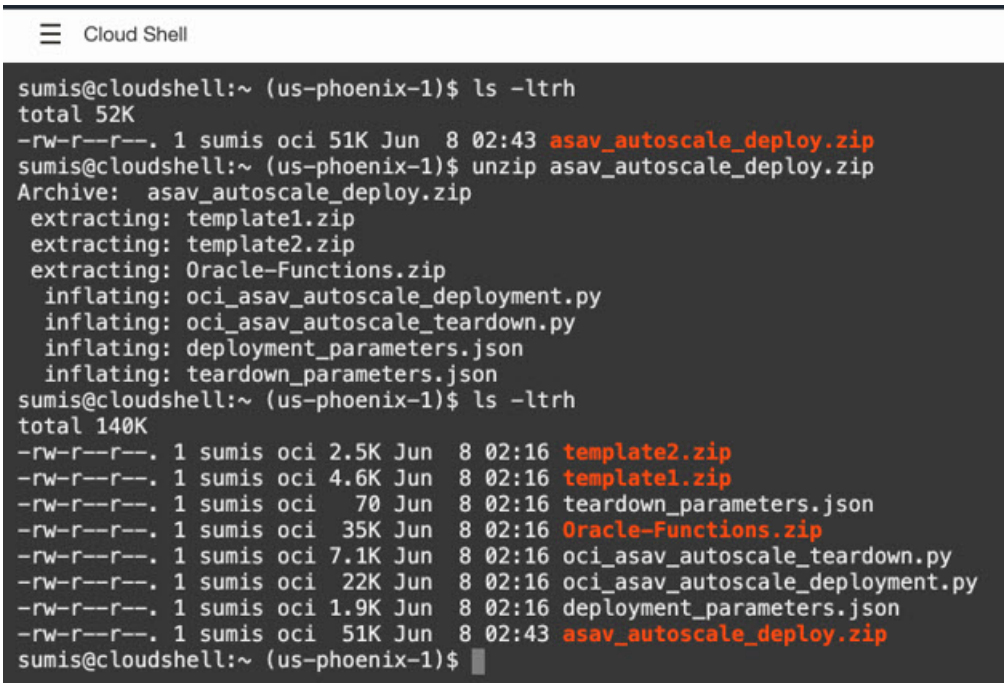
Select *Terraform template template2.zip* in the target folder as source of Terraform configuration.

Step 3 In next step, click **Terraform Actions** > **Apply**.

Deploy Autoscale Using Cloud Shell

To avoid the deployment overhead, you can invoke the easy, end-to-end deployment script to deploy the autoscale solution (terraform template1, template2 and oracle functions).

Step 1 Upload the *asav_autoscale_deploy.zip* file in the target folder to the cloud shell and extract the files.



Step 2 Make sure you have updated the input parameters in the *deployment_parameters.json* before executing the python3 `make.py` build command.

Step 3 To start the autoscale solution deployment, run the `python3 oci_asav_autoscale_deployment.py` command on the cloud shell.

It will take approximately 10-15 minutes for the solution deployment to complete.

If there is any error during the solution deployment, error log is saved.

Validate Deployment

Validate if all resources are deployed and the Oracle Functions are connected with Alarm & Events. By default, instance pool has minimum and maximum number of instances as zero. You can edit the instance pool in OCI UI with the minimum and maximum number that you want. This will trigger new ASAv instances.

We recommend that you launch only one instance and check for its workflow, validate its behaviour to ensure that it is working as it is expected. Post this validation, you can deploy the actual requirements of ASAv.

Note Specify the minimum number of ASAv instances as **Scale-In protected** to avoid their removal by OCI scaling policies.

Upgrade

Upgrade Autoscale Stack

No support for upgrade in this release. Stacks should be re-deployed.

Upgrade ASAv VMs

No support for upgrade for ASAv VMs in this release. The Stack should be re-deployed with the required ASAv image.

Instance Pool

1. To change minimum and maximum number of instances in the Instance Pool:

   Click **Developer Services** > **Function** > **Application Name(created by Terraform Template 1)** > **Configuration**.

   Change the min_instance_count and max_instance_count respectively.

2. Deletion/Termination of Instance is not equal to Scale-in. If any instance in the Instance Pool is deleted/terminated due to external action and not the scale-in action, instance pool automatically initiates a new instance to recover.

3. Max_instance _count defines threshold limit for Scale-out action, but it can be surpassed by changing the instance count of the Instance Pool through the UI. Ensure that the instance count from UI is less than max_instance_count set in OCI Application. Else, increase the threshold accordingly.

4. Reducing the count of instances in Instance Pool directly from the application does not perform the clean-up actions set programmatically. Due to which backends will not be drained and removed from both the load balancers, if ASAv has license, it will be lost.

5. Due to some reasons, if ASAv instance is unhealthy, not responding and unreachable through SSH for some definite period of time, instance is removed from the instance pool forcefully, any license may be lost.

Oracle Functions

- Oracle Functions are actually docker images. These images are saved into root directory of OCI Container registry. These images should not be deleted as it will also delete the function that are used in the Autoscale solution.
- OCI Application created by Terraform template-1, contains crucial environmental variables, which are required by Oracle Functions to work properly. Neither the value nor the format of these environment variables should be changed, unless it is mandated. Any changes made are reflected with new instances only.

Load Balancer Backend Sets

In OCI, Load Balancer attachment to instance pool is only supported using primary interface that is configured as management interface in ASAv. Hence, inside interface is connected to Internal Load Balancer's backend set; outside interface is connected to External load balancer's backend set. These IPs are not automatically added or removed from backend set. The Autoscale solution programmatically handles both of this task. But

in case of any external action, maintenance or troubleshooting, there could be situation demanding manual effort to operate on them.

As per requirements, more ports can be opened on Load Balancer using listener and backend sets. Upcoming instances IPs are automatically added to the backend set, however already existing instances IPs should be manually added.

Adding Listener in Load Balancer

To add some port as listener in Load Balancer, go to **OCI** > **Networking** > **Load Balancer** > **Listener** > **Create Listener**.

Register a backend to Backend Set

In order to register an ASAv instance to Load Balancer, ASAv instance Outside interface IP should be configured as a backend in the Backend Set of External Load Balancer. Inside interface IP should be configured as backend in Backend set of Internal Load Balancer. Ensure that the port you are using has been added into the listener.

Delete Autoscale Configuration from OCI

Stacks deployed using terraform can be deleted in the same manner, using Resource Manager in OCI. Deletion of stack removes all the resources created by it and all the information associated with these resources are removed permanently.

Note In case of stack deletion, it is recommended to make the Minimum number of instances in Instance pool to 0, wait for instances to be terminated. This will help removal of all instances and won't leave any residue.

You can perform a Manual Deletion or use Delete Autoscale Using Cloud Shell .

Manual Deletion

The end-to-end autoscale solution deletion consist of three steps: Delete Terraform Template-2 Stack, Delete Oracle-Functions, and then Delete Terraform Template-1 Stack .

Delete Terraform Template-2 Stack

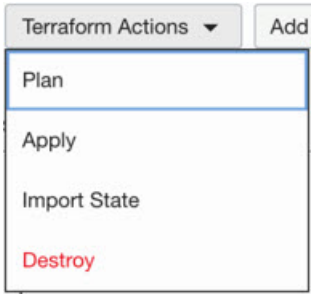
To delete the Autoscale configuration, you must begin with Terraform Template-2 stack deletion.

Step 1 Log into the OCI portal.

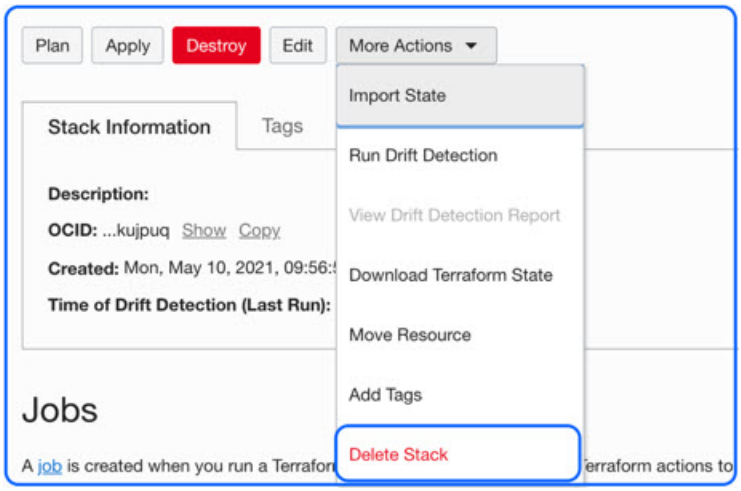
The region is displayed in the upper right corner of your screen. Make sure you are in the intended region.

Step 2 Choose **Developer Services** > **Resource Manager** > **Stack**.

Step 3 Select the stack created by Terraform Template-2, then select **Destroy** in **Terraform Actions** drop-down menu as shown in the figure below:



Destroy Job is created which takes some time to remove resources one after another. You can delete the stack after the destroy job is completed. as shown in the figure below:



Step 4 Proceed to delete the Oracle functions.

Delete Oracle-Functions

The Oracle-Function deployment is not a part of Terraform Template Stack deployment, it is uploaded separately using Cloud Shell. Hence, its deletion is also not supported by Terraform Stack deletion. You must delete all the Oracle-Functions inside the OCI application created by Terraform Template-1.

Step 1 Log into the OCI portal.

The region is displayed in the upper right corner of your screen. Make sure you are in the intended region.

Step 2 Choose **Developer Services** > **Functions**. Choose the application name that was created in Template-1 stack.

Step 3 Inside this application visit each function and delete it.

Delete Terraform Template-1 Stack

Note Template-1 Stack deletion will only succeed after deleting all Oracle-Functions.

Same as Terraform Template-2 Deletion.

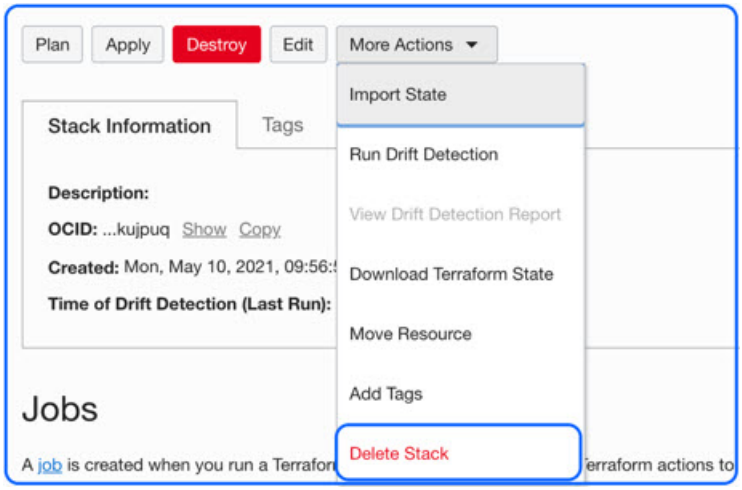
Step 1 Log into the OCI portal.

The region is displayed in the upper right corner of your screen. Make sure you are in the intended region.

Step 2 Choose **Developer Services** > **Resource Manager** > **Stack**.

Step 3 Select the stack created by Terraform Template-2, then click **Destroy** in Terraform **Actions** drop-down menu. Destroy Job will be created which will take some time to remove resources one after another.

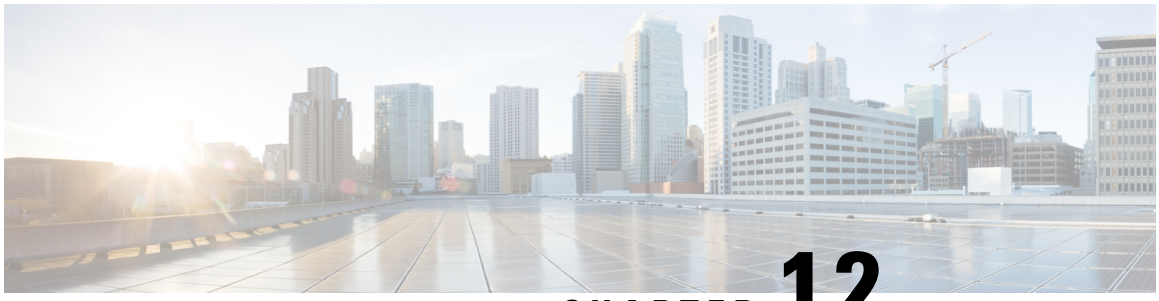
Step 4 After the destroy job is completed, you can delete the stack from **More Actions** drop-down menu as shown in the figure below.



Post successful deletion of Terraform Template-1 stack, you must verify whether all the resources are deleted and there is no residue of any kind.

Delete Autoscale Using Cloud Shell

User can use the script to delete the stacks and oracle functions by executing the `python3 oci_asav_autoscale_teardown.py` command in the cloud shell. If the stacks are deployed manually, update the stack id of the stack1 and stack2, and update the application id in the *teardown_parameters.json* file.



CHAPTER 12

Deploy the ASAv on Google Cloud Platform

You can deploy the ASAv on the Google Cloud Platform (GCP).

Overview

GCP lets you build, deploy, and scale applications, websites, and services on the same infrastructure as Google.

The ASAv runs the same software as physical ASAs to deliver proven security functionality in a virtual form factor. The ASAv can be deployed in the public GCP. It can then be configured to protect virtual and physical data center workloads that expand, contract, or shift their location over time.

GCP Machine Type Support

Select the Google virtual machine type and size to meet your ASAv needs.

The ASAv supports the following *General-purpose N1, N2* and *Compute-optimized C2* GCP machine types:

Table 25: Supported Compute-Optimized Machine Types

Compute-Optimized Machine Types	Attributes	
	vCPUs	Memory (GB)
c2-standard-4	4	16
c2-standard-8	8	32
c2-standard-16	16	64

Table 26: Supported General Purpose Machine Types

Machine Type	Attributes	
	vCPUs	Memory (GB)
n1-standard-4	4	15
n1-standard-8	8	30
n1-standard-16	16	60
n2-standard-4	4	16
n2-standard-8	8	32
n2-standard-16	16	64
n1-highcpu-8	8	7.2
n1-highcpu-16	16	14.4
n2-highcpu-8	8	8
n2-highcpu-16	16	16
n2-highmem-4	4	32
n2-highmem-8	8	64

- The ASAv requires a minimum of 3 interfaces.
- The maximum supported vCPUs is 16.
- The Memory-Optimized machine type is not supported

You create an account on GCP, launch an ASAv instance using the ASA virtual firewall (ASAv) offering on the GCP Marketplace, and choose a GCP machine type.

C2 Compute-Optimized Machine Type Limitations

The Compute-Optimized C2 machine types have the following restrictions:

- You cannot use regional persistent disks with compute-optimized machine types. For more information, see the Google documentation Adding or resizing regional persistent disks.
- Subject to different disk limits than general-purpose and memory-optimized machine types. For more information, see the Google documentation Block storage performance.
- Available only in select zones and regions. For more information, see the Google documentation Available regions and zones.
- Available only on select CPU platforms. For more information, see the Google documentation CPU platforms.

Performance Tiers for ASAv

The ASAv supports performance-tiered licensing that provides different throughput levels and VPN connection limits based on deployment requirements.

Performance Tier	Instance Type (Core/RAM)	Rate Limit	RA VPN Session Limit
ASAv5	c2-standard-4 4 core/16 GB	100 Mbps	50
ASAv10	c2-standard-4 4 core/16 GB	1 Gbps	250
ASAv30	c2-standard-4 4 core/16 GB	2 Gbps	750
ASAv50	c2-standard-8 8 core/32 GB	7.6 Gbps	10,000
ASAv100	c2-standard-16 16 core/64 GB	16 Gbps	20,000

Prerequisites

- Create a GCP account at https://cloud.google.com.
- Create your GCP project. See the Google documentation, Creating Your Project.
- License the ASAv. Until you license the ASAv, it will run in degraded mode, which allows only 100 connections and throughput of 100 Kbps. See Licenses: Smart Software Licensing.
- Interface requirements:
  - Management interface—Used to connect the ASAv to the ASDM; can't be used for through traffic.
  - Inside interface—Used to connect the ASAv to inside hosts.
  - Outside interface—Used to connect the ASAv to the public network.
- Communications paths:
  - Public IPs for access into the ASAv.
- For ASAv system requirements, see Cisco ASA Compatibility.

Guidelines and Limitations

Supported Features

The ASAv on GCP supports the following features:

- Deployment in the GCP Virtual Private Cloud (VPC)
- Maximum of 16 vCPUs per instance
- Routed mode (default)
- Licensing – Only BYOL is supported

Unsupported Features

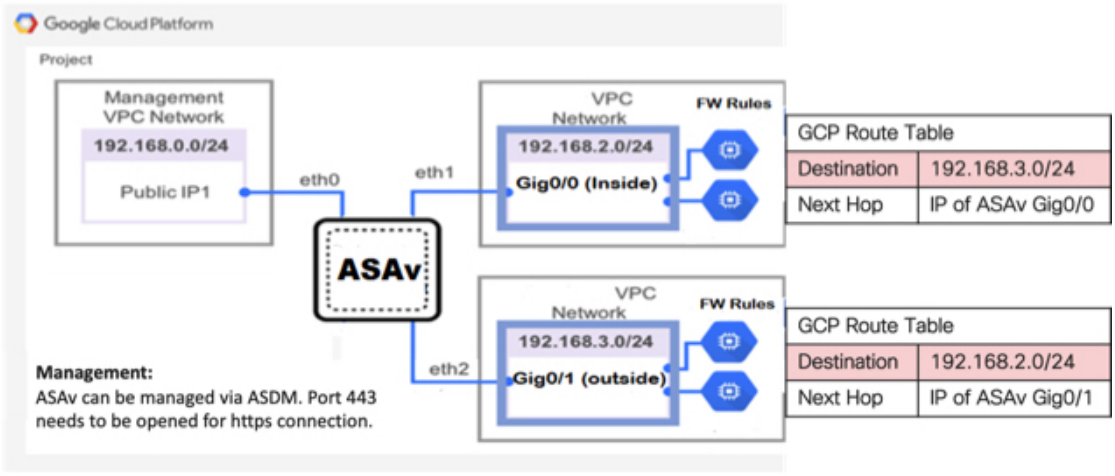
The ASAv on GCP does not support the following:

- IPv6
  - Instance-level IPv6 setting is not supported on GCP
  - Only the load balancer can accept IPv6 connections, and proxy them over IPv4 to GCP Instances
- Jumbo Frames
- ASAv native HA
- Autoscale
- Transparent/inline/passive modes

Sample Network Topology

The following figure shows the recommended network topology for the ASAv in Routed Firewall Mode with 3 subnets configured in GCP for the ASAv (management, inside, and outside).

Figure 54: Sample ASAv on GCP Deployment



Deploy the ASAv on Google Cloud Platform

You can deploy the ASAv on the Google Cloud Platform (GCP).

Create VPC Networks

Before you begin

The ASAv deployment requires three networks which you must create prior to deploying the ASAv. The networks are as follows:

- Management VPC for the management subnet.
- Inside VPC for the inside subnet.
- Outside VPC for the outside subnet.

Additionally, you set up the route tables and GCP firewall rules to allow traffic flow through the ASAv. The route tables and firewall rules are separate from those that are configured on the ASAv itself. Name the GCP route tables and firewall rules according to associated network and functionality. See Sample Network Topology, on page 220.

Step 1 In the GCP console, choose **Networking** > **VPC network** > **VPC networks** , then click **Create VPC Network**.

Step 2 In the **Name** field, enter the descriptive name for your VPC network, for example, *vpc-asiasouth-mgmt*.

Step 3 From the **Subnet creation mode**, click **Custom**.

Step 4 In the **Name** field under **New subnet**, enter the desired name, for example, *vpc-asiasouth-mgmt*.

Step 5 From the **Region** drop-down list, select the region appropriate for your deployment. All three networks must be in the same region.

Step 6 In the **IP address range** field, enter the first network's subnet in CIDR format, such as 10.10.0.0/24.

Step 7 Accept the defaults for all other settings, then click **Create**.

Step 8 Repeat steps 1-7 to create the remaining two networks in your VPC.

Create the Firewall Rules

You apply the firewall rules for the management interface (to allow SSH and HTTPS connections) while deploying the ASAv instance, see Create the ASAv Instance on GCP, on page 222. According to your requirements, you can also create firewall rules for the inside and outside interfaces.

Step 1 In the GCP console, choose **Networking** > **VPC network** > **Firewall**, then click **Create Firewall Rule**.

Step 2 In the **Name** field, enter a descriptive name for your firewall rule, for example, *vpc-asiasouth-inside-fwrule*.

Step 3 From the **Network** drop-down list, select the name of the VPC network for which you are creating the firewall rule, for example, *asav-south-inside*.

Step 4 From the **Targets** drop-down list, select the option applicable for your firewall rule, for example, **All instances in the network**.

Step 5 In the **Source IP ranges** field, enter the source IP address ranges in CIDR format, for example, 0.0.0.0/0.

Traffic is only allowed from sources within these IP address ranges.

Step 6 Under **Protocols and ports**, select **Specified protocols and ports**.

Step 7 Add your security rules.

Step 8 Click **Create**.

Create the ASAv Instance on GCP

Complete the following steps to deploy an ASAv instance using the Cisco ASA virtual firewall (ASAv) offering from the GCP Marketplace.

Step 1 Log into to the GCP Console.

Step 2 Click **Navigation menu** > **Marketplace**.

Step 3 Search the Marketplace for “Cisco ASA virtual firewall (ASAv)” and choose the offering.

Step 4 Click **Launch**.

Step 5 Add a unique **Deployment name** for the instance.

Step 6 Select the **Zone** where you want to deploy the ASAv.

Step 7 Select the appropriate **Machine type**. For a list of supported machine types, see Overview, on page 217.

Step 8 (Optional) Paste the public key from the SSH key pair under **SSH key (optional)**.

The key pair consists of a public key that GCP stores and a private key file that the user stores. Together they allow you to connect to your instance securely. Be sure to save the key pair to a known location, as it will be required to connect to the instance.

Step 9 Choose whether to allow or block the project-wide SSH keys to access this instance. See the Google documentation Allowing or blocking project-wide public SSH keys from a Linux instance.

Step 10 (Optional) Under **Startup script**, provide the day0 configuration for your ASAv. The day0 configuration is applied during the firstboot of the ASAv.

The following example shows a sample day0 configuration you can copy and paste in the **Startup script** field:

See the ASA Configuration Guides and the ASA Command Reference for complete information on the ASA commands.

Important When you copy text from this example, you should validate the script in a third-party text editor or validation engine to prevent format errors and remove invalid Unicode characters.

```
!ASA Version 9.15.1

interface management0/0

management-only
nameif management
security-level 100
ip address dhcp setroute
no shut
!
same-security-traffic permit inter-interface
same-security-traffic permit intra-interface
!
crypto key generate rsa modulus 2048
ssh 0 0 management
ssh timeout 60
ssh version 2
username admin password cisco123 privilege 15
username admin attributes
service-type admin
! required config end
dns domain-lookup management
dns server-group DefaultDNS
name-server 8.8.8.8
```

Step 11 Keep the default **Boot disk type** and **Boot disk size in GB** for the provisioned disk space.

Step 12 Configure the interfaces under **Network interfaces**.

- management
- inside
- outside

Note You cannot add interfaces to an instance after you create it. If you create the instance with an improper interface configuration, you must delete the instance and recreate it with the proper interface configuration.

a) From the **Network** drop-down list, select a VPC network, for example, *vpc-asiasouth-mgmt*.
b) From the **External IP** drop-down list, select the appropriate option.

For the management interface, select the **External IP** to **Ephemeral**. This is optional for inside and outside interfaces.

c) Click **Done**.

Step 13 Apply the firewall rules under **Firewall**.

- Check the **Allow TCP port 22 traffic from the Internet (SSH access)** check box to allow SSH.
- Check the **Allow HTTPS traffic from the Internet (ASDM access)** check box to allow HTTPS connections.

Step 14 Click **More** to expand the view and make sure that **IP Forwarding** is set to **On**.

Step 15 Click **Deploy**.

View the instance details from the VM instance page of the GCP console. You'll find the internal IP address, external IP address, and controls to stop and start the instance. You need to stop the instance if you need to edit it.

Access the ASAv Instance on GCP

Make sure that you have already enabled a firewall rule to allow SSH (TCP connections through port 22) during deployment. See Create the ASAv Instance on GCP, on page 222 for more information.

This firewall rule enables access to the ASAv instance and allows you to connect to the instance using the following methods.

- External IP
  - Any other SSH client or third-party tools
- Serial console
- Gcloud command line

See the Google documentation, Connecting to instances for more information.

Note You can log in to the ASAv instance using the credentials specified in the day0 configuration, or by using the SSH key pair you created during the instance launch.

Connect to the ASAv Instance Using an External IP

The ASAv instance is assigned with an internal IP and an external IP. You can use the external IP to access the ASAv instance.

Step 1 In the GCP console, choose **Compute Engine** > **VM instances**.

Step 2 Click the ASAv instance name to open the **VM instance details** page.

Step 3 Under the **Details** tab, click the drop-down menu for the **SSH** field.

Step 4 Select the desired option from the **SSH** drop-down menu.

You can connect to the ASAv instance using the following method.

- Any other SSH client or third-party tools—See the Google documentation, Connecting using third-party tools for more information.

Note You can log in to the ASAv instance using the credentials specified in the day0 configuration, or by using the SSH key pair you created during the instance launch.

Connect to the ASAv Instance Using SSH

To connect to the ASAv instance from a Unix-style system, log in to the instance using SSH.

Step 1 Use the following command to set the file permissions so that only you can read the file:

```
$ chmod 400 <private_key>
```

Where:

`<private_key>` is the full path and name of the file that contains the private key associated with the instance you want to access.

Step 2 Use the following SSH command to access the instance.

```
$ ssh -i <private_key> <username>@<public-ip-address>
```

Where:

`<private_key>` is the full path and name of the file that contains the private key associated with the instance you want to access.

`<username>` is the username for the ASAv instance.

`<public-ip-address>` is your instance IP address that you retrieved from the Console.

Connect to the ASAv Instance Using the Serial Console

Step 1 In the GCP console, choose **Compute Engine** > **VM instances**.

Step 2 Click the ASAv instance name to open the **VM instance details** page.

Step 3 Under the **Details** tab, click **Connect to serial console**.

See the Google documentation, Interacting with the serial console for more information.

Connect to the ASAv Instance Using Gcloud

Step 1 In the GCP console, choose **Compute Engine** > **VM instances**.

Step 2 Click the ASAv instance name to open the **VM instance details** page.

Step 3 Under the **Details** tab, click the drop-down menu for the **SSH** field.

Step 4 Click **View gcloud command** > **Run in Cloud Shell**.

The Cloud Shell terminal window opens. See the Google documentation, gcloud command-line tool overview, and gcloud compute ssh for more information.

CPU Usage and Reporting

The CPU Utilization report summarizes the percentage of the CPU used within the time specified. Typically, the Core operates on approximately 30 to 40 percent of total CPU capacity during nonpeak hours and approximately 60 to 70 percent capacity during peak hours.

vCPU Usage in the ASA Virtual

The ASA virtual vCPU usage shows the amount of vCPUs used for the data path, control point, and external processes.

The GCP reported vCPU usage includes the ASA virtual usage as described:

- ASA Virtual idle time
- %SYS overhead used for the ASA virtual machine
- Overhead of moving packets between vSwitches, vNICs, and pNICs. This overhead can be quite significant.

CPU Usage Example

The **show cpu usage** command can be used to display CPU utilization statistics.

Example

```
Ciscoasa#show cpu usage

CPU utilization for 5 seconds = 1%; 1 minute: 2%; 5 minutes: 1%
```

The following is an example in which the reported vCPU usage is substantially different:

- ASA Virtual reports: 40%
- DP: 35%
- External Processes: 5%

- ASA (as ASA Virtual reports): 40%
- ASA idle polling: 10%
- Overhead: 45%

The overhead is used to perform hypervisor functions and to move packets between NICs and vNICs using the vSwitch.

GCP CPU Usage Reporting

Click the instance name on GCP Console and then click on the tab **Monitoring**. You will be able to see the CPU usage percentage.

Compute Engine lets you export detailed reports of your Compute Engine usage to a Google Cloud Storage bucket using the usage export feature. Usage reports provide information about the lifetime of your resources. For example, you can see how many VM instances in your project are running an `n2-standard-4` machine type and how long each instance has been running. You can also review the storage space of a persistent disk, and information about other Compute Engine features.

ASA Virtual and GCP Graphs

There are differences in the CPU % numbers between the ASA Virtual and GCP:

- The GCP graph numbers are always higher than the ASA Virtual numbers.
- GCP calls it %CPU usage; the ASA Virtual calls it %CPU utilization.

The terms “%CPU utilization” and “%CPU usage” mean different things:

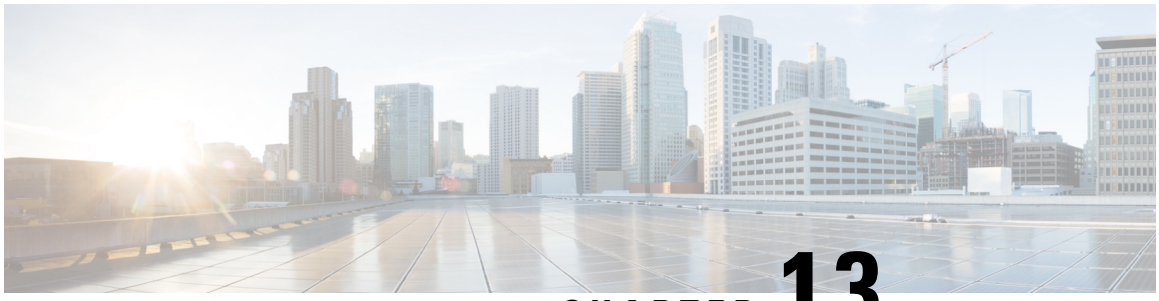
- CPU utilization provides statistics for physical CPUs.
- CPU usage provides statistics for logical CPUs, which is based on CPU hyperthreading. But because only one vCPU is used, hyperthreading is not turned on.

GCP calculates the CPU % usage as follows:

Amount of actively used virtual CPUs, specified as a percentage of the total available CPUs

This calculation is the host view of the CPU usage, not the guest operating system view, and is the average CPU utilization over all available virtual CPUs in the virtual machine.

For example, if a virtual machine with one virtual CPU is running on a host that has four physical CPUs and the CPU usage is 100%, the virtual machine is using one physical CPU completely. The virtual CPU usage calculation is Usage in MHz / number of virtual CPUs x core frequency



CHAPTER 13

Deploy the ASAv Auto Scale Solution on GCP

Overview

The following sections describe how the components of the auto scale solution work for the ASAv on GCP.

About the Auto Scale Solution

ASAv auto scale for GCP is a complete serverless implementation that makes use of serverless infrastructure provided by GCP (Cloud Functions, Load Balancers, Pub/Sub, Instance Groups, etc.).

Some of the key features of the ASAv auto scale for GCP implementation include:

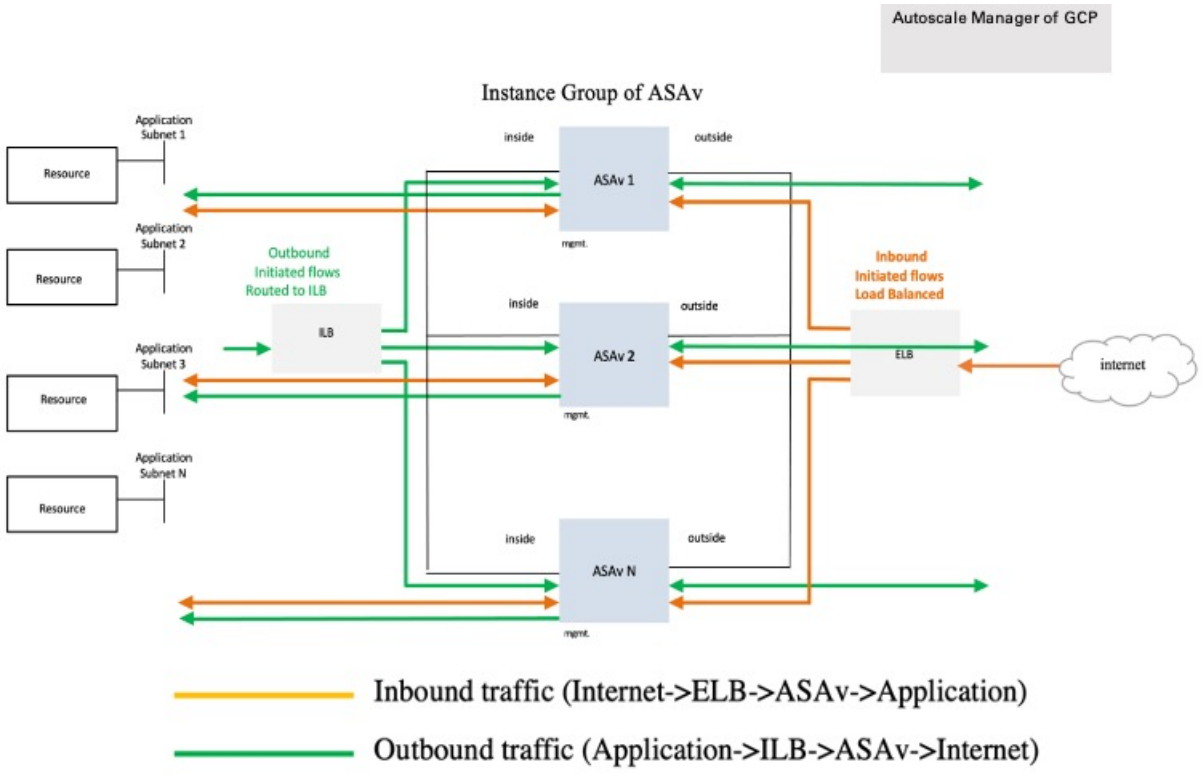
- GCP Deployment Manager template-based deployment.
- Support for scaling metrics based on CPU.
- Support for ASAv deployment and multi-availability zones.
- Completely automated configuration automatically applied to scaled-out ASAv instances.
- Support for Load Balancers and multi-availability zones.
- Cisco provides an auto scale for GCP deployment package to facilitate the deployment.

Auto Scale Use Case

The ASAv auto scale for GCP is an automated horizontal scaling solution that positions an ASAv instance group sandwiched between a GCP Internal load balancer (ILB) and a GCP External load balancer (ELB).

- The ELB distributes traffic from the Internet to ASAv instances in the instance group; the firewall then forwards traffic to the application.
- The ILB distributes outbound Internet traffic from an application to ASAv instances in the instance group; the firewall then forwards traffic to the Internet.
- A network packet will never pass through both (internal & external) load balancers in a single connection.
- The number of ASAv instances in the scale set will be scaled and configured automatically based on load conditions.

Figure 55: ASAv Auto Scale Use Case



Scope

This document covers the detailed procedures to deploy the serverless components for the ASAv Auto Scale for GCP solution.

Important

- Read the entire document before you begin your deployment.
- Make sure the prerequisites are met before you start deployment.
- Make sure you follow the steps and order of execution as described herein.

Download the Deployment Package

The ASAv Auto Scale for GCP solution is a GCP Deployment Manager template-based deployment that makes use of the serverless infrastructure provided by GCP (Cloud Functions, Load Balancers, Pub/Sub, Instance Groups, etc.).

Download the files required to launch the ASAv auto scale for GCP solution. Deployment scripts and templates for your ASA version are available in the GitHub repository.

Attention Note that Cisco-provided deployment scripts and templates for auto scale are provided as open source examples, and are not covered within the regular Cisco TAC support scope.

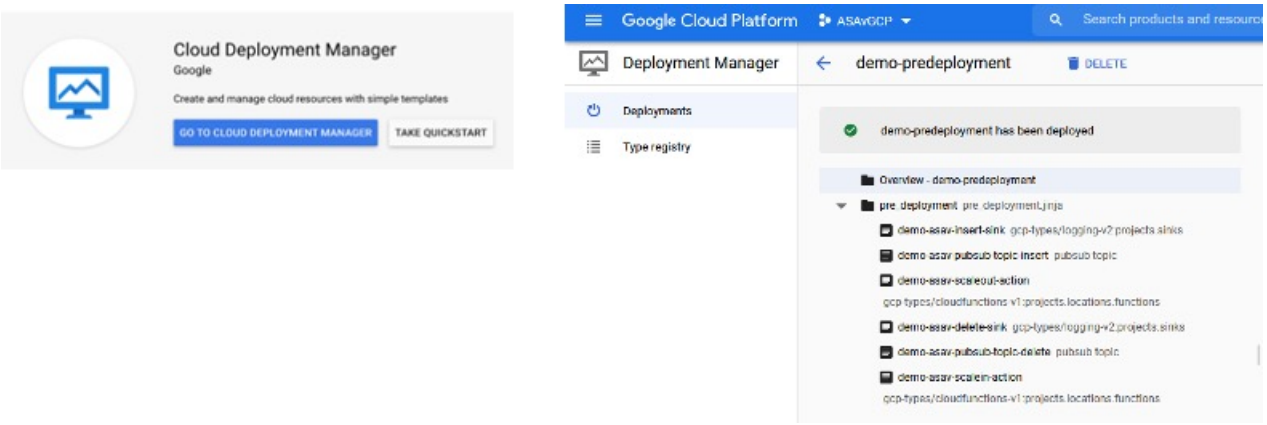
Auto Scale Solution Components

The following components make up the ASAv auto scale for GCP solution.

Deployment Manager

- Treat your configuration as code and perform repeatable deployments. Google Cloud Deployment Manager allows you to specify all the resources needed for your application in a declarative format using YAML. You can also use Python or Jinja2 templates to parameterize the configuration and allow the reuse of common deployment paradigms.
- Create configuration files that define the resources. The process of creating those resources can be repeated over and over with consistent results. See https://cloud.google.com/deployment-manager/docs for more information.

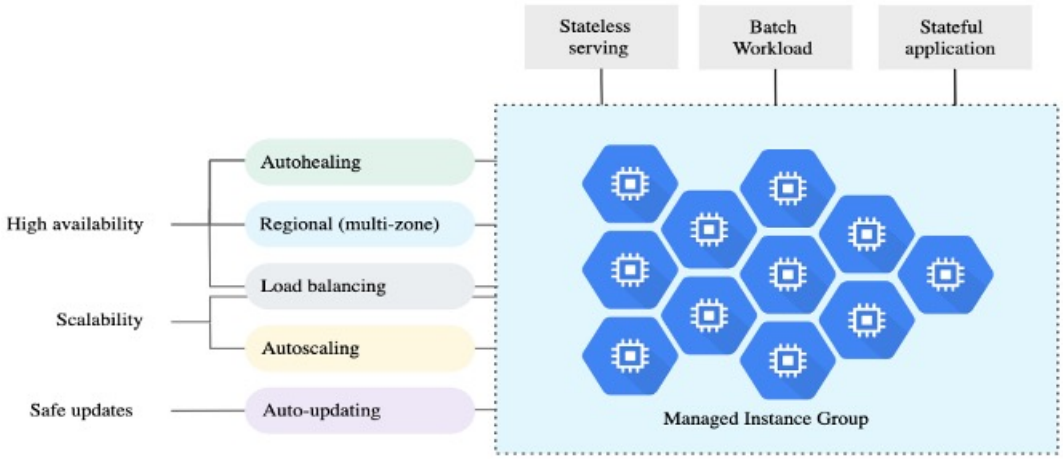
Figure 56: Deployment Manager View



Managed Instance Group in GCP

A Managed Instance Group (MIG) creates each of its managed instances based on the instance template and optional stateful configuration that you specify. See https://cloud.google.com/compute/docs/instance-groups for more information.

Figure 57: Instance Group Features

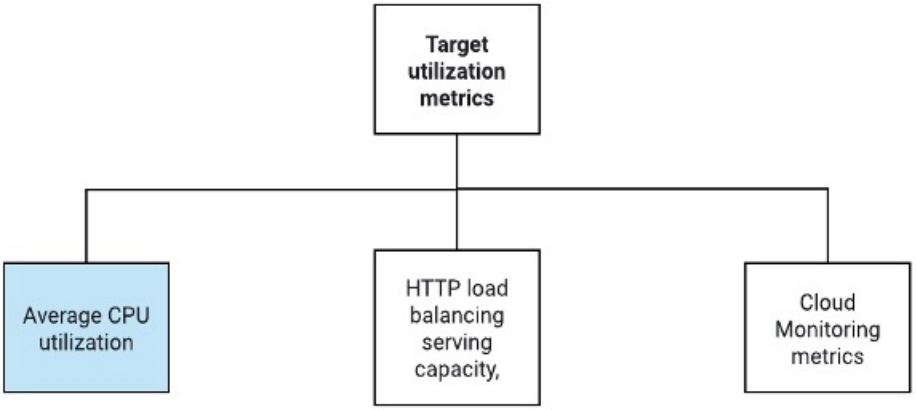


Target Utilization Metrics

- The following diagram alongside shows the target utilization metrics. Only average CPU utilization metrics are used in making autoscaling decisions.
- The autoscaler continuously collects usage information based on the selected utilization metric, compares actual utilization to your desired target utilization, and uses this information to determine whether the group needs to remove instances (Scale In) or add instances (Scale Out).
- The target utilization level is the level at which you want to maintain your virtual machine (VM) instances. For example, if you scale based on CPU utilization, you can set your target utilization level at 75% and

the autoscaler will maintain the CPU utilization of the specified group of instances at or close to 75%. The utilization level for each metric is interpreted differently based on the autoscaling policy. See https://cloud.google.com/compute/docs/autoscaler for more information.

Figure 58: Target Utilization Metrics



Serverless Cloud Functions

You use serverless Google Cloud functions for setting the SSH Password, enable Password, and Changing the Hostname when the instance comes up in the Instance Group Manager.

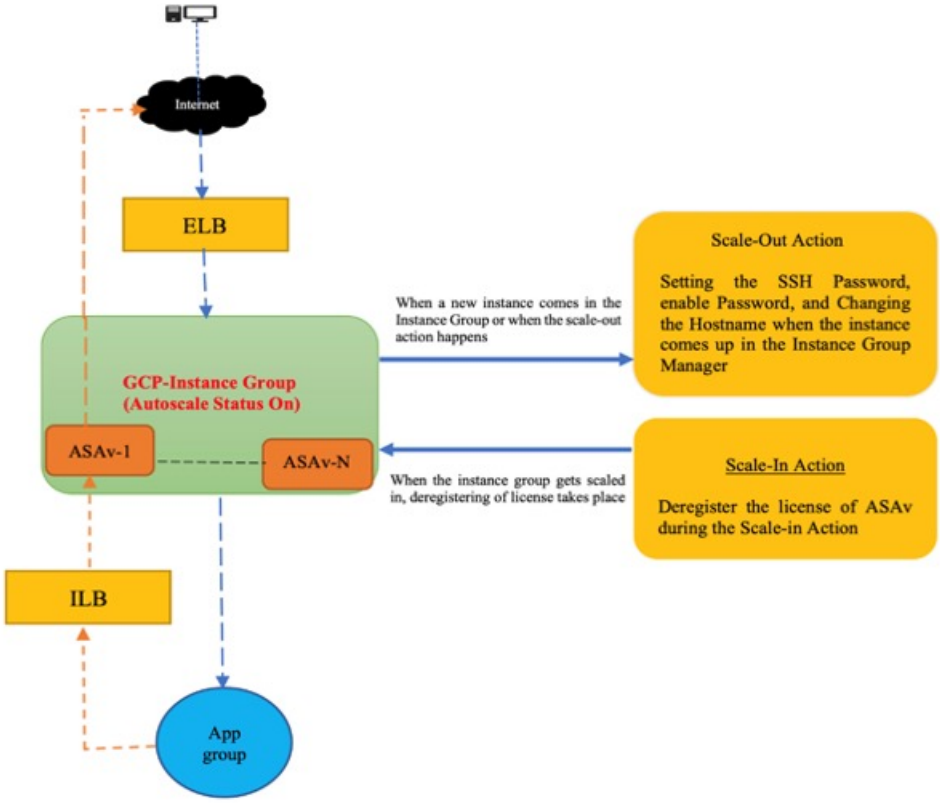
- When a new ASAv instance comes up in the instance group during Scale Out, you need to set the SSH Password, enable Password, and change the Hostname because you cannot always monitor the Scale Out process.
- Cloud functions are triggered through a Cloud Pub/Sub Topic during the Scale Out process. You also have a Log Sink with a filter that is exclusive to the addition of instances while Scale Out.

Serverless License Deregistering using Cloud Functions

- While the instances are getting deleted during Scale In, you need to deregister the license from the ASAv instance.
- Cloud functions are triggered through a Cloud Pub/Sub Topic. Particularly for the deletion process, you have a Log Sink with a filter that is exclusive to the deletion of instances while Scale In.
- Cloud Function, when triggered, will SSH into the deleting ASAv instance and run the command for license deregistration.

High-Level Overview of Autoscale Solution

Figure 59: Autoscale Solution Overview



Prerequisites

GCP Resources

GCP Project

An existing or newly created project is required to deploy all the components of this solution.

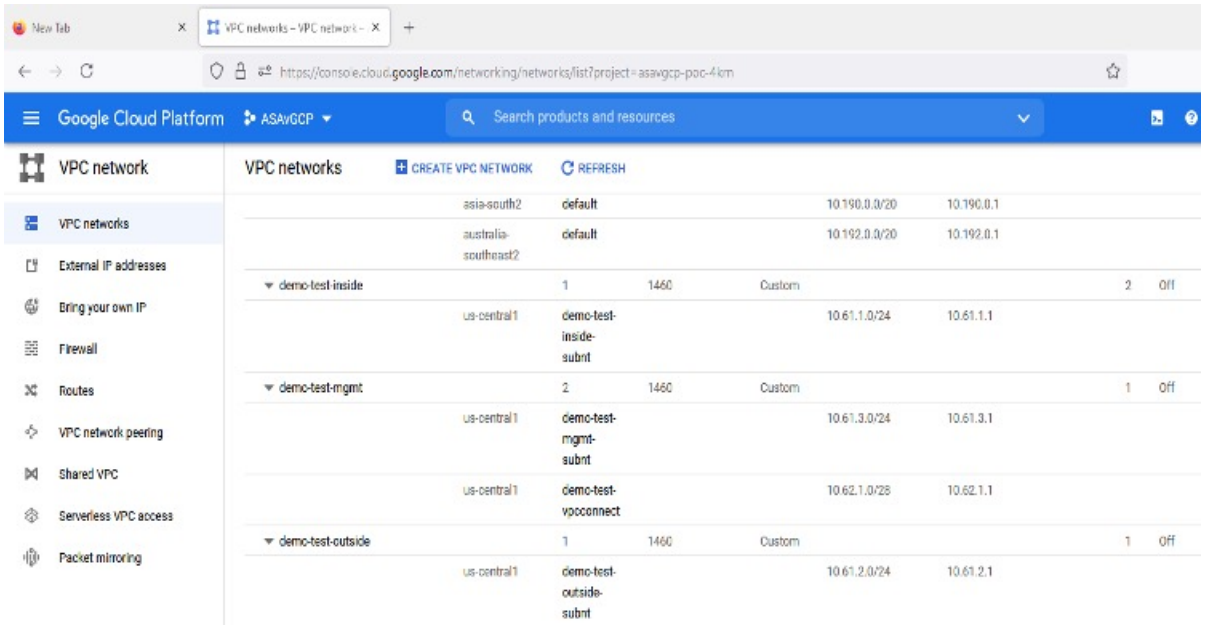
Networking

Make sure three VPCs are available/created. An auto scale deployment will not create, alter, or manage any networking resources.

The ASAv requires 3 network interfaces, thus your virtual network requires 3 subnets for:

- Management traffic
- Inside traffic
- Outside traffic

Figure 60: VPC Network View



Firewall

Firewall rules that allow inter VPC communication and also allow health probes are required to be created. You must note the firewall tags which are used later in the deployment manager template.

The following ports should be open in the Network Security Group to which the subnets are connected:

- SSH(TCP/22) — Required for the health probe between the Load Balancer and ASAv. Required for communication between the serverless functions and ASAv.
- Application-specific protocol/ports — Required for any user applications (for example, TCP/80, etc.).

Prepare the ASA Configuration File

Prepare an ASAv configuration file which will be put into the deployment manager jinja configuration file. This configuration will be used as a startup script in the instance template for ASAv in the project.

The configuration file should have the following (at a minimum):

- Set DHCP IP assignment to all the interfaces.
- Nic0 should be marked as ‘outside’ because GCP Load Balancer forwards traffic only to nic0.
- Nic0 will be used to SSH to ASAv as it only supports IP forwarding.
- Enable SSH on the outside interface in ASA configuration.
- Create NAT configuration to forward traffic from outside to inside interface.
- Create Access policy to allow desired traffic.
- For the health status of resources, their health probes should be redirected to the metadata server using proper NAT rules.

The following is a sample ASA configuration file for reference only.

```
!ASA Version 9.15.1.10
!Interface Config
interface G0/0
nameif inside
security-level 100
ip address dhcp setroute
no shutdown


interface G0/1
nameif management
security-level 50
ip address dhcp setroute
no shutdown

interface M0/0
no management-only
nameif outside
security-level 0
ip address dhcp setroute
no shutdown
!
same-security-traffic permit inter-interface
!
!Due to some constraints in GCP,
!"GigabitEthernet0/0" will be used as a Management interface
!"Management0/0" will be used as a data interface
crypto key generate rsa modulus 2048
ssh 0.0.0.0 0.0.0.0 management
ssh version 2
ssh timeout 60
aaa authentication ssh console LOCAL
ssh authentication publickey {{ properties["publicKey"] }}
username admin privilege 15
username admin attributes
service-type admin

! required config end
dns domain-lookup management
dns server-group DefaultDNS
name-server 8.8.8.8
!
access-list all extended permit ip any any
access-list out standard permit any4
access-group all global
! Objects
object network metadata
host 169.254.169.254
object network ilb
host $(ref.{{ properties["resourceNamePrefix"] }}-ilb-ip.address)
object network hc1
subnet 35.191.0.0 255.255.0.0
object network hc2
subnet 130.211.0.0 255.255.63.0
object network elb
host $(ref.{{ properties["resourceNamePrefix"] }}-elb-ip.address)
object network appServer
host 10.61.2.3
object network defaultGateway
subnet 0.0.0.0 0.0.0.0
! Nat Rules
nat (inside,outside) source dynamic hc1 ilb destination static ilb metadata
```

```
nat (inside,outside) source dynamic hc2 ilb destination static ilb metadata
nat (inside,outside) source dynamic defaultGateway interface
 !
 object network appServer
nat (inside,outside) static $(ref.{{ properties["resourceNamePrefix"] }}-elb-ip.address)
object network defaultGateway
nat (outside,inside) dynamic interface
! Route Add
route inside 0.0.0.0 0.0.0.0 10.61.1.1 2
route management 0.0.0.0 0.0.0.0 10.61.3.1 3
license smart register idtoken <licenseIDToken>
```

Build the GCP Cloud Function Package

The ASAv GCP auto scale solution requires that you build two archive files that deliver the cloud functions in the form of a compressed ZIP package.

- `scalein-action.zip`
- `scaleout-action.zip`

See the auto scale deployment instructions for information on how to build the `scalein-action.zip` and `scaleout-action.zip` packages.

These functions are as discrete as possible to carry out specific tasks and can be upgraded as needed for enhancements and new release support.

Input Parameters

The following table defines the template parameters and provides an example. Once you decide on these values, you can use these parameters to create the ASAv device when you deploy the GCP Deployment Manager template into your GCP project.

Table 27: Template Parameters

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
resourceNamePrefix	String	All the resources are created with name containing this prefix. Example: demo-test	New
region	Valid regions supported by GCP [String]	Name of the region where project will be deployed. Example: us-central1	
serviceAccountMailId	String [Email Id]	Email address that identifies the service account.	

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
vpcConnectorName	String	Name of the connector that handles the traffic between your serverless environment and your VPC network. Example: demo-test-vpc-connector	
bucketName	String	Name of the GCP storage bucket where the cloud function ZIP package will be uploaded. Example: demo-test-bkt	
cpuUtilizationTarget	Decimal (0,1]	The average CPU utilization of the VMs in the instance group the autoscaler should maintain. Example: 0.5	
healthCheckFirewallRuleName	String	Tag of the firewall rule that allows packets from health check probe IP ranges. Example: demo-test-healthallowall	Existing
insideFirewallRuleName	String	Tag of the firewall rules that allows communication in Inside VPC. Example: demo-test-inside-allowall	Existing
insideVPCName	String	Name of Inside VPC. Example: demo-test-inside	Existing
insideVPCSubnet	String	Name of Inside subnet. Example: demo-test-inside-subnt	Existing
machineType	String	Machine type for the ASAv VM. Example: e2-standard-4	

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
maxASACount	Integer	The maximum ASAv instances allowed in the instance group. Example: 3	
mgmtFirewallRuleName	String	Tag of the firewall rules which allows communication in Management VPC. Example: demo-test-mgmt-allowall	
mgmtVPCName	String	Name of Management VPC. Example: demo-test-mgmt	
mgmtVPCSubnet	String	Name of Management Subnet. Example: demo-test-mgmt-subnt	
minASACount	Integer	The minimum ASAv instances available in the Instance Group at any given time. Example: 1	
outsideFirewallRuleName	String	Tag of the firewall rules which allows communication in outside VPC. Example: demo-test-outside-allowall	
outsideVPCName	String	Name of Outside VPC. Example: demo-test-outside	
outsideVPCSubnet	String	Name of Outside Subnet. Example: demo-test-outside-subnt	
publicKey	String	SSH key of the ASAv VM.	

Parameter Name	Allowed Values/Type	Description	Resource Creation Type
sourceImageURL	String	Image of the ASAv which is to be used in the project. Example: https://www.googleapis.com/compute/v1/projects/cisco-public/global/images/cisco-asav-9-15-1-15	
Application server IP address	String	Internal IP address of the inside Linux machine. Example: 10.61.1.2	
Inside VPC Gateway IP address	String	Gateway of Inside VPC. Example: 10.61.1.1	
Management VPC Gateway IP address	String	Gateway of Management VPC. Example: 10.61.3.1	

Deploy the Auto Scale Solution

Step 1 Clone the Git repository to a local folder.

```
git clone git_url -b branch_name
```

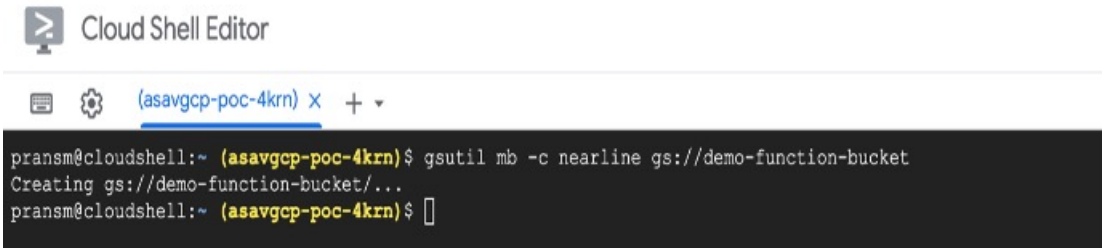
Example:

```
Last login: Thu Jun  3 13:01:32 on ttys002
(base) pransm@PRANSM-M-F9KA ~ % git clone https://bitbucket-eng-bgl1.cisco.com/bitbucket/scm/vcb/cloud_autoscale.git -b saaarwar_asa_autoscale_public_key
Cloning into 'cloud_autoscale'...
remote: Enumerating objects: 1604, done.
remote: Counting objects: 100% (1604/1604), done.
remote: Compressing objects: 100% (1507/1507), done.
remote: Total 1604 (delta 759), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (1604/1604), 58.35 MiB | 8.54 MiB/s, done.
Resolving deltas: 100% (759/759), done.
(base) pransm@PRANSM-M-F9KA ~ %
```

Step 2 Create the bucket in gcloud CLI.

```
gsutil mb -c nearline gs://bucket_name
```

Example:



Step 3 Build compressed Zip packages:

a) Create compressed Zip packages consisting of the following files from the folders `scalein_action` and `scaleout_action`.

- main.py
- basic_functions.py

  requirements.txt

b) Rename the compressed Zip packages to `scaleout-action.zip` and `scalein-action.zip`.

Note Navigate inside the folder, select the files, right-click, and select 'compress | archive' to make a .zip that GCP can read.

Step 4 Upload the compressed Zip packages (`scaleout-action.zip` and `scalein-action.zip`) to the Cloud Editor workspace.

Step 5 Upload the following files from the deployment manager template to the Cloud Editor workspace.

- asav_autoscale.jinja
- asav_autoscale_params.yaml
- pre_deployment.jinja
- pre_deployment.yaml

Step 6 Copy the compressed Zip packages to the Bucket Storage.

- `gsutil cp scaleout-action.zip gs://bucket_name`
- `gsutil cp scalein-action.zip gs://bucket_name`

Example:

```
pransm@cloudshell:~ (asavgcp-poc-4krn)$ gsutil cp scaleout-action.zip gs://demo-function-bucket
Copying file://scaleout-action.zip [Content-Type=application/zip]...
/ [1 files][  3.3 KiB/  3.3 KiB]
Operation completed over 1 objects/3.3 KiB.
pransm@cloudshell:~ (asavgcp-poc-4krn)$ gsutil cp scalein-action.zip gs://demo-function-bucket
Copying file://scalein-action.zip [Content-Type=application/zip]...
/ [1 files][  3.3 KiB/  3.3 KiB]
Operation completed over 1 objects/3.3 KiB.
pransm@cloudshell:~ (asavgcp-poc-4krn)$
```

Step 7 Create VPC and Subnet for inside, outside, and management interfaces.

In the management VPC, you need to have /28 subnet, for example, 10.8.2.0/28.

Step 8 You need three firewall rules for the interfaces inside, outside, and management. Also, you should have a firewall rule to allow the health check probes.

Step 9 Update the parameters in the Jinja and YAML files for the Pre-Deployment and ASAv Autoscale deployment.

a) Open the `asav_autoscale_params.yaml` file and update the following parameters:

- **resourceNamePrefix**: <resourceNamePrefix>
- **region**: <region>
- **serviceAccountMailId**: <serviceAccountMailId>
- **publicKey**: <publicKey>
- **insideVPCName**: <Inside-VPC-Name>
- **insideVPCSubnet**: <Inside-VPC-Subnet>
- **outsideVPCName**: <Outside-VPC-Name>
- **outsideVPCSubnet**: <Outside-VPC-Subnet>
- **mgmtVPCName**: <Mgmt-VPC-Name>
- **mgmtVPCSubnet**: <Mgmt-VPC-Subnet>
- **insideFirewallRuleName**: <Inside-Network-Firewall-Tag>
- **outsideFirewallRuleName**: <Outside-Network-Firewall-Tag>
- **mgmtFirewallRuleName**: <Mgmt-Network-Firewall-Tag>
- **healthCheckFirewallRuleName**: <HealthCheck-IP-Firewall-Tag>
- **machineType**: <machineType>

Note For the ASAv auto scale, the **cpuUtilizationTarget: 0.5** parameter is set and you can edit it according to your requirements.

This value signifies 50% CPU usage of all the ASAv Instance Group.

b) Open the `asav_autoscale.jinja` file and update the following parameters.

- **host**: <Application server IP address>
- **route inside 0.0.0.0 0.0.0.0**: <Inside VPC Gateway IP address> 2
- **route management 0.0.0.0 0.0.0.0**: <Management VPC Gateway IP address> 3
- **license smart register idtoken**: <licenseIDToken>

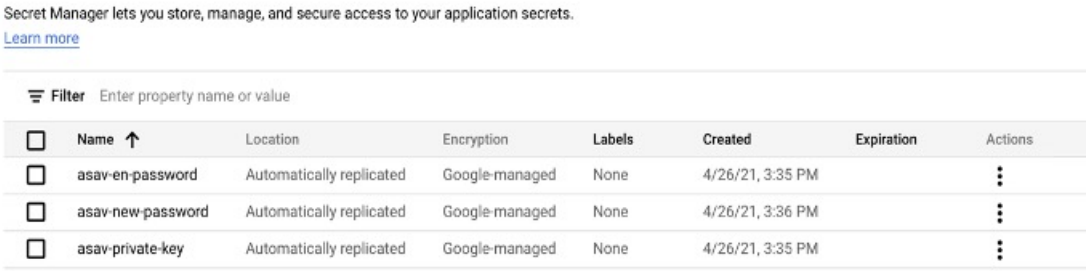
c) Open the `pre_deployment.yaml` file and update the following parameters.

- **resourceNamePrefix**: <resourceNamePrefix>
- **region**: <region>
- **serviceAccountMailId**: <serviceAccountMailId>
- **vpcConnectorName**: <VPC-Connector-Name>

- **bucketName**: <bucketName>

Step 10 Create three secrets for the following using the Secret Manager GUI. See https://console.cloud.google.com/security/secret-manager.

- asav-en-password
- asav-new-password
- asav-private-key



Step 11 Create the VPC connector.

```
gcloud beta compute networks vpc-access connectors create <vpc-connector-name>
--region <region> --subnet=</28 subnet name>
```

Example:

```
gcloud beta compute networks vpc-access connectors create demo-vpc-connector
--region us-central1 --subnet=outside-connect-28
Create request issued for: [demo-vpc-connector]
Waiting for operation [projects/asavgcp-poc-4krn/locations/us-central1/operations/
10595de7-837f-4c19-9396-0c22943ecf15] to complete...done.
Created connector [demo-vpc-connector].
```

Step 12 Deploy the pre-deployment YAML configuration.

```
gcloud deployment-manager deployments create <pre-deployment-name>
--config pre_deployment.yaml
```

Example:

```
gcloud deployment-manager deployments create demo-predeployment
--config pre_deployment.yaml
The fingerprint of the deployment is b'9NOy0gsTPgg16SqUEVsBjA=='
Waiting for create [operation-1624383045917-5c55e266e596d-4979c5b6-66d1025c]...done.
Create operation operation-1624383045917-5c55e266e596d-4979c5b6-66d1025c
completed successfully
```

NAME	TYPE	STATE
demo-asav-delete-sink	gcp-types/logging-v2:projects.sinks	COMPLETED
demo-asav-insert-sink	gcp-types/logging-v2:projects.sinks	COMPLETED
demo-asav-pubsub-topic-delete	pubsub.v1.topic	COMPLETED
demo-asav-pubsub-topic-insert	pubsub.v1.topic	COMPLETED
demo-asav-scalein-action	gcp-types/cloudfunctions-v1:projects.locations.functions	COMPLETED
demo-asav-scaleout-action	gcp-types/cloudfunctions-v1:projects.locations.functions	COMPLETED

Step 13 Create the ASAv auto scale deployment.

```
gcloud deployment-manager deployments create <deployment-name>
--config asav_autoscale_params.yaml
```

Example:

```
gcloud deployment-manager deployments create demo-asav-autoscale
--config asav_autoscale_params.yaml
The fingerprint of the deployment is b'1JCQi7Il-laWOY7vOLza0g=='
Waiting for create [operation-1624383774235-5c55e51d79d01-1a3acf92-4f3daf16]...done.
Create operation operation-1624383774235-5c55e51d79d01-1a3acf92-4f3daf16
completed successfully.
```

NAME	TYPE	STATE
demo-asav-autoscaler	compute.v1.regionAutoscaler	COMPLETED
demo-asav-backend-service-elb	compute.v1.regionBackendService	COMPLETED
demo-asav-backend-service-ilb	compute.v1.regionBackendService	COMPLETED
demo-asav-fr-elb	compute.v1.forwardingRule	COMPLETED
demo-asav-fr-ilb	compute.v1.forwardingRule	COMPLETED
demo-asav-hc-elb	compute.v1.regionHealthChecks	COMPLETED
demo-asav-hc-ilb	compute.v1.healthCheck	COMPLETED
demo-asav-health-check	compute.v1.healthCheck	COMPLETED
demo-asav-instance-group	compute.v1.regionInstanceGroupManager	COMPLETED
demo-asav-instance-template	compute.v1.instanceTemplate	COMPLETED
demo-elb-ip	compute.v1.address	COMPLETED

Step 14 Create a route for ILB to forward the packets from the inside application to the Internet.

```
gcloud beta compute routes create <ilb-route-name>
--network=<inside-vpc-name> --priority=1000 --destination-range=0.0.0.0/0
--next-hop-ilb=<ilb-forwarding-rule-name> --next-hop-ilb-region=<region>
```

Example:

```
gcloud beta compute routes create demo-ilb --network=sdt-test-asav-inside
--priority=1000 --destination-range=0.0.0.0/0 --next-hop-ilb=demo-asav-fr-ilb
--next-hop-ilb-region=us-central1
Created [https://www.googleapis.com/compute/beta/projects/asavgcp-poc-4krn/global
/routes/demo-ilb].
```

NAME	NETWORK	DEST_RANGE	NEXT_HOP	PRIORITY
demo-ilb	sdt-test-asav-inside	0.0.0.0/0	10.7.1.60	1000

Step 15 Create Cloud Router and Cloud NAT.

```
gcloud compute routers create <cloud-router-name>
--project=<project-name> --region <region> --network=<outside-vpc-name>
--advertisement-mode=custom
```

```
gcloud compute routers nats create <cloud-nat-name>
--router=<cloud-router-name> --nat-all-subnet-ip-ranges --auto-allocate-nat-external-ips
--region=<region>
```

Example:

```
gcloud compute routers create demo-cloud-router --project=asavgcp-poc-4krn
--region us-central1 --network=sdt-test-asav-outside --advertisement-mode=custom
Creating router [demo-cloud-router]...done.
```

NAME	REGION	NETWORK
demo-cloud-router	us-central1	sdt-test-asav-outside

```
gcloud compute routers nats create demo-cloud-nat
--router=demo-cloud-router --nat-all-subnet-ip-ranges
```

```
--auto-allocate nat-external-ips --region=us-central1
Creating NAT [demo-cloud-nat] in router [demo-cloud-router]...done.
```

Auto Scale Logic

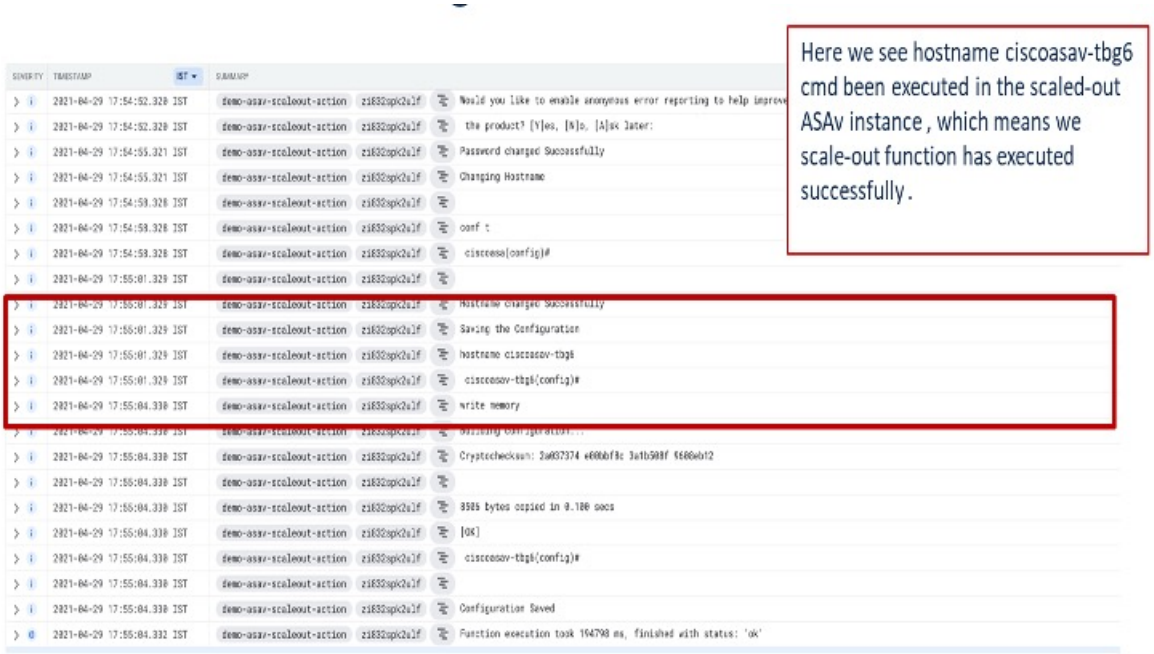
- The autoscaler treats the target CPU utilization level as a fraction of the average use of all vCPUs over time in the instance group.
- If the average utilization of your total vCPUs exceeds the target utilization, the autoscaler adds more VM instances. If the average utilization of your total vCPUs is less than the target utilization, the autoscaler removes instances.
- For example, setting a 0.75 target utilization tells the autoscaler to maintain an average utilization of 75% among all vCPUs in the instance group.
- Only CPU utilization metrics are used in scaling decisions.
- This logic is based on the assumption that load balancer will try to equally distribute connections across all ASAs, and on average, all ASAs should be loaded equally.

Logging and Debugging

Logs of cloud functions can be viewed as follows.

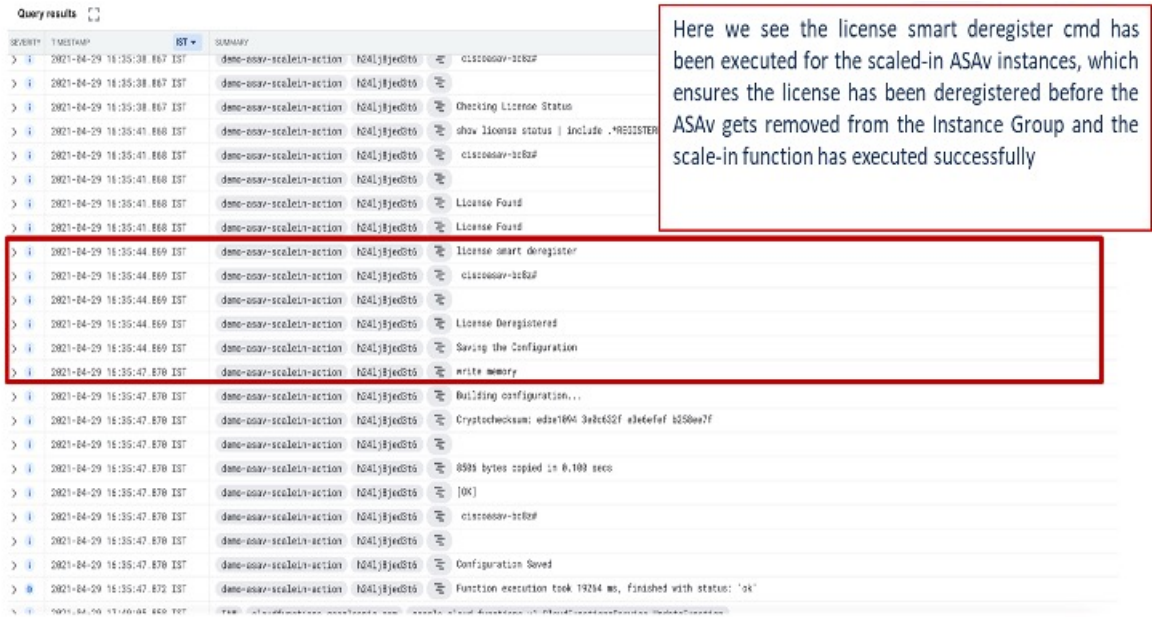
- Scale Out function logs

Figure 61: Scale Out Function Logs



- Scale In function log

Figure 62: Scale In Function Log



Guidelines and Limitations

- Only IPv4 is supported.
- Supported licensing is BYOL only. PAYG is not available for ASAv on GCP.
- The external Load Balancer is created by the template, and therefore, any specific DNS requirements for Load Balancer's public IP are out of the scope.
- An assumption is made that the application is behind a user-created Load Balancer, and the ASAv will route all traffic to this Load Balancer (instead of directly sending traffic to a specific application IP).
- Details about the need for TAGs, redundancy, and Load Balancer affinity configurations are not considered.
- ASAv credentials are visible to you as:
  - Clear text in the serverless code.
  - In all the instances in the Instance Group.
  - In the Instance Template, if you are using a shared GCP account.

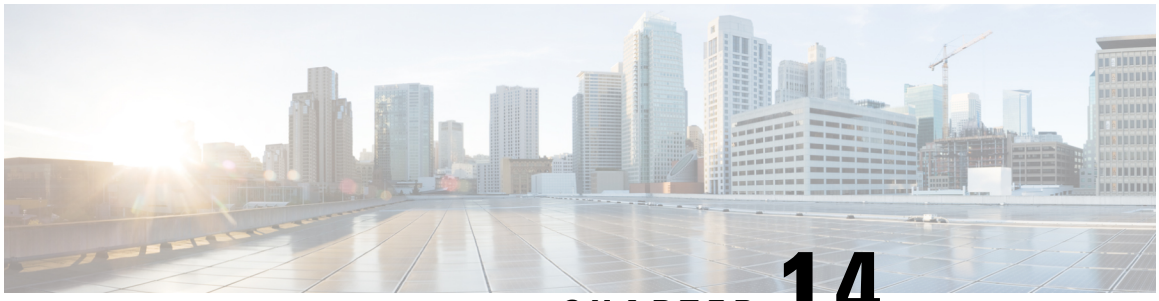
  Such sensitive data can be protected using the public key service in GCP.

Important Cisco recommends tracking the ASAv registration with the licensing server periodically to check if the scaled-out ASAs are registering with the licensing server as expected, and scaled-in ASAv instances are getting removed from the license server).

Troubleshooting

The following are common error scenarios and debugging tips for ASAv auto scale for GCP:

- `main.py` not found—Make sure that the Zip package is made only from the files. You can go to cloud functions and check the file tree. There should not be any folder.
- Error while deploying the template—Make sure that all the parameters values within "<>" are filled in. jinja and .yaml as well, or the deployment by the same name exists already.
- Google Function cannot reach ASAv—Make sure that the VPC connector is created and the same name is mentioned in the YAML parameter file.
- Authentication Failed while SSH-ing ASAv—Make sure that the Public and Private key pair is correct.
- License Registration Failed—Make sure that the License ID token is correct. Also, ensure that the Cloud NAT is created and ASAv is able to reach tools.cisco.com.



CHAPTER 14

Deploy the ASAv on OpenStack

You can deploy the ASAv on OpenStack.

Overview

You can deploy the ASAv in an OpenStack environment. OpenStack is a set of software tools for building and managing cloud computing platforms for public and private clouds, and is tightly integrated with the KVM hypervisor.

Enabling OpenStack platform support for ASAv allows you to run ASAv on open source cloud platforms. OpenStack uses a KVM hypervisor to manage virtual resources. ASAv devices are already supported on KVM hypervisor. Therefore, there is no extra addition of kernel packages or drivers to enable OpenStack support.

Prerequisites for the ASAv and OpenStack

- Download the ASAv qcow2 file from software.cisco.com and put it on your Linux host:

  http://www.cisco.com/go/asa-software

- ASAv supports deployment on opensource OpenStack environment and Cisco VIM managed OpenStack environment.

  Set up the OpenStack environment according to the OpenStack guidelines.

  - See the opensource OpenStack document:

    Stein Release - https://docs.openstack.org/project-deploy-guide/openstack-ansible/stein/overview.html

    Queens Release - https://docs.openstack.org/project-deploy-guide/openstack-ansible/queens/overview.html

    - See the Cisco Virtualized Infrastructure Manager (VIM) OpenStack document: Cisco Virtualized Infrastructure Manager Documentation, 3.4.3 to 3.4.5

- License the ASAv. Until you license the ASAv, it will run in degraded mode, which allows only 100 connections and throughput of 100 Kbps. See Licenses: Smart Software Licensing.
- Interface requirements:
    - Management interface
    - Inside and outside interfaces
- Communications paths:
    - Management interface—Used to connect the ASAv to the ASDM; can't be used for traffic.
    - Inside interface (required)—Used to connect the ASAv to inside hosts.
    - Outside interface (required)—Used to connect the ASAv to the public network.
- Communications paths:
    - Floating IPs for access into the ASAv.
- Minimum supported ASAv version:
    - ASA 9.16.1
- For OpenStack requirements, see System Requirements.
- For ASAv system requirements, see Cisco ASA Compatibility.

Guidelines and Limitations

Supported Features

The ASAv on OpenStack supports the following features:

- Deployment of ASAv on the KVM hypervisor running on a compute node in your OpenStack environment.
- OpenStack CLI
- Heat template-based deployment
- OpenStack Horizon dashboard
- Routed mode (default)
- Licensing – Only BYOL is supported
- ASAv management using the CLI and ASDM
- Drivers - VIRTIO, VPP, and SRIOV
- IPv6 (version 9.19 and later)

Unsupported Features

The ASAv on OpenStack does not support the following:

- Autoscale
- OpenStack releases other than the OpenStack Stein and Queens releases
- Operating systems other than the Ubuntu 18.04 version and Red Hat Enterprise Linux (RHEL) 7.6

System Requirements

The OpenStack environment must conform to the following supported hardware and software requirements.

Table 28: Hardware and Software Requirements

Category	Supported Versions	Notes
Server	UCS C240 M5	2 UCS servers are recommended, one each for os-controller and os-compute nodes.
Driver	VIRTIO, IXGBE, I40E	These are the supported drivers.
Operating System	Ubuntu Server 18.04	This is the recommended OS on UCS servers.
OpenStack Version	Stein release	Details of the various OpenStack releases are available at: https://releases.openstack.org/

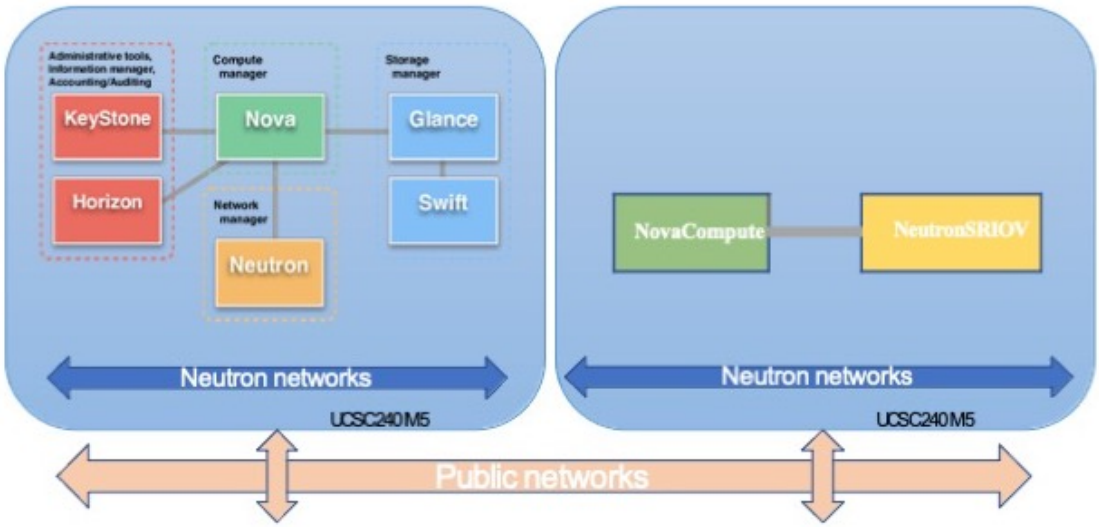
Table 29: Hardware and Software Requirements for Cisco VIM Managed OpenStack

Category	Supported Versions	Notes
Server Hardware	UCS C220-M5/UCS C240-M4	5 UCS servers are recommended, three each for os-controller and Two or more for os-compute nodes.
Drivers	VIRTIO, SRIOV, and VPP	These are the supported drivers.

Category	Supported Versions	Notes
Cisco VIM Version	Cisco VIM 3.4.4 Supported on: • Operating System - Red Hat Enterprise Linux 7.6 • OpenStack version - OpenStack 13.0 (Queens Release)	See Cisco Virtualized Infrastructure Manager Documentation, 3.4.3 to 3.4.5 for more information. Details of the various OpenStack releases are available at https://releases.openstack.org/.
	Cisco VIM 4.2.1 Supported on: • Operating System - Red Hat Enterprise Linux 8.2 • OpenStack version - OpenStack 16.1 (Train Release)	See Cisco Virtualized Infrastructure Manager Documentation, 4.2.1 for more information. Details of the various OpenStack releases are available at https://releases.openstack.org/.

Figure 63: OpenStack Platform Topology

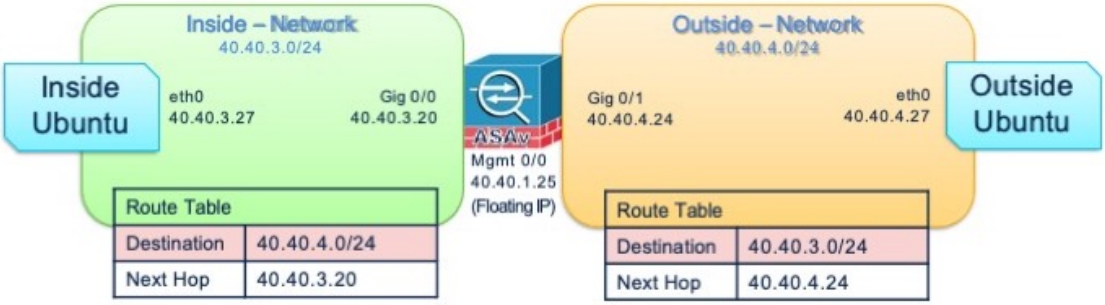
OpenStack platform topology shows the general OpenStack setup on two UCS servers.



Sample Network Topology

The following figure shows the recommended network topology for the ASAv in Routed Firewall Mode with 3 subnets configured in OpenStack for the ASAv (management, inside, and outside).

Figure 64: Sample ASAv on OpenStack Deployment



Deploy the ASAv

Cisco provides sample heat templates for deploying the ASAv. Steps for creating the OpenStack infrastructure resources are combined in a heat template (`deploy_os_infra.yaml`) file to create networks, subnets, and router interfaces. At a high-level, the ASAv deployment steps are categorized into the following sections.

- Upload the ASAv qcow2 image to the OpenStack Glance service.
- Create the network infrastructure.
  - Network
  - Subnet
  - Router interface
- Create the ASAv instance.
  - Flavor
  - Security Groups
  - Floating IP
  - Instance

You can deploy the ASAv on OpenStack using the following steps.

Upload the ASAv Image to OpenStack

Copy the qcow2 image (`asav-<version>.qcow2`) to the OpenStack controller node, and then upload the image to the OpenStack Glance service.

Before you begin

Download the ASAv qcow2 file from Cisco.com and put it on your Linux host:

http://www.cisco.com/go/asa-software

Note A Cisco.com login and Cisco service contract are required.

Step 1 Copy the qcow2 image file to the OpenStack controller node.

Step 2 Upload the ASAv image to the OpenStack Glance service.

```
root@ucs-os-controller:$ openstack image create <image_name> --public --disk-
format qcow2 --container-format bare --file ./<asav_qcow2_file>
```

Step 3 Verify if the ASAv image upload is successful.

root@ucs-os-controller:$ openstack image list

Example:

```
root@ucs-os-controller:$ openstack image
list+--------------------------------------+-------------------+--------+
| ID                                   | Name              | Status
|+--------------------------------------+-------------------+--------+
| 06dd7975-0b6e-45b8-810a-4ff98546a39d | asav-<version>-image | active
|+--------------------------------------+-------------------+--------+
```

The uploaded image and its status is displayed.

What to do next

Create the network infrastructure using the `deploy_os_infra.yaml` template.

Create the Network Infrastructure for OpenStack and ASAv

Before you begin

Heat template files are required to create the network infrastructure and the required components for ASAv, such as flavor, networks, subnets, router interfaces, and security group rules:

- `deploy_os_infra.yaml`
- `env.yaml`

Templates for your ASAv version are available from the GitHub repository at:

- https://github.com/CiscoDevNet/cisco-asav

Important Note that Cisco-provided templates are provided as open source examples, and are not covered within the regular Cisco TAC support scope. Check GitHub regularly for updates and ReadMe instructions.

Step 1 Deploy the infrastructure heat template file.

root@ucs-os-controller:$ openstack stack create <stack-name> -e <environment files name> -t <deployment file name>

Example:

```
root@ucs-os-controller:$ openstack stack create infra-stack -e env.yaml -t deploy_os_infra.yaml
```

Step 2 Verify if the infrastructure stack is created successfully.

root@ucs-os-controller:$ openstack stack list

What to do next

Create the ASAv instance on OpenStack.

Create the ASAv Instance on OpenStack

Use the sample ASAv heat template to deploy ASAv on OpenStack.

Before you begin

A heat template is required to deploy the ASAv on OpenStack:

- `deploy_asav.yaml`

Templates for your ASAv version are available from the GitHub repository at:

- https://github.com/CiscoDevNet/cisco-asav

Important Note that Cisco-provided templates are provided as open source examples, and are not covered within the regular Cisco TAC support scope. Check GitHub regularly for updates and ReadMe instructions.

Step 1 Deploy the ASAv heat template file (`deploy_asav.yaml`) to create the ASAv instance.

root@ucs-os-controller:$ openstack stack create asav-stack -e env.yaml-t deploy_asav.yaml

Example:

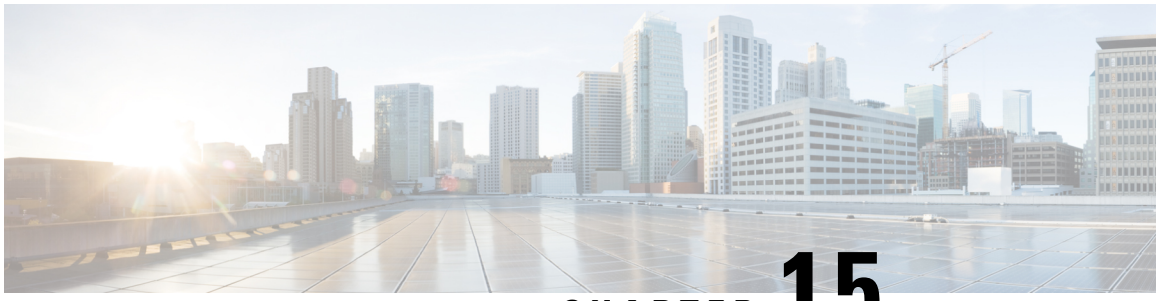
```
+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| id                  | 14624af1-e5fa-4096-bd86-c453bc2928ae |
| stack_name          | asav-stack                           |
| description         | ASAvtemplate                         |
| creation_time       | 2020-12-07T14:55:05Z                 |
| updated_time        | None                                 |
| stack_status        | CREATE_IN_PROGRESS                   |
| stack_status_reason | Stack CREATE started                 |
+---------------------+--------------------------------------+
```

Step 2 Verify that your ASAv stack is created successfully.

root@ucs-os-controller:$ openstack stack list

Example:

```
+--------------------------------------+------------+----------------------------------+--------------+-----------------+------------+
| ID                                   | Stack Name | Project                          | Stack Status
    | Creation Time    | Updated Time |
+--------------------------------------+------------+----------------------------------+--------------+-----------------+------------+
| 14624af1-e5fa-4096-bd86-c453bc2928ae | asav-stack | 13206e49b48740fdafca83796c6f4ad5 |
CREATE_COMPLETE | 2020-12-07T14:55:05Z | None         |
| 198336cb-1186-45ab-858f-15ccd3b909c8 | infra-stack | 13206e49b48740fdafca83796c6f4ad5 |
CREATE_COMPLETE | 2020-12-03T10:46:50Z | None         |
+--------------------------------------+------------+----------------------------------+--------------+-----------------+------------+
```



CHAPTER 15

Deploy the ASAv on Nutanix

This chapter describes the procedures to deploy the ASAv to a Nutanix environment.

Overview

The Cisco Adaptive Security Virtual Appliance (ASAv) brings full firewall functionality to virtualized environments to secure data center traffic and multitenant environments.

You can deploy the ASAv on Nutanix.

Guidelines and Limitations

Important The ASAv deploys with a disk storage size of 8 GB. It is not possible to change the resource allocation of the disk space.

Review the following guidelines and limitations before you deploy the ASAv.

Recommended vNIC

The following vNIC is recommended for optimum performance.

VirtIO—A para-virtualized network driver that supports 10 Gbps operation but also requires CPU cycles.

CPU Pinning

CPU pinning is required for the ASAv to function in a Nutanix environment; see Enable CPU Pinning, on page 53.

Failover for High Availability

For failover deployments, make sure that the standby unit has the same license entitlement; for example, both units should have 2 Gbps entitlement.

Important You must add the data interfaces to each ASAv in the same order when creating a high availability pair. If the exact same interfaces are added to each ASAv, but in a different order, you may see errors at the ASAv console, which could impact the failover functionality.

General Guidelines

- The maximum number of interfaces supported is ten. You will receive an error message if you attempt to add more than ten interfaces

Note
- By default the ASAv configures the management interface and inside interface on the same subnet.
- When you are modifying the network interfaces, you must turn off the ASAv device.

- By default, the ASAv assumes that you configured both the management and inside interfaces on the **different subnet**. The management interface has "IP address DHCP setroute" and the Default Gateway is provided by DHCP.
- The ASAv must be powered up on first boot with at least three interfaces. Your system will not deploy without three interfaces.
- The ASAv supports a total of 10 interfaces—one management interface (nic0) and a maximum of nine network interfaces (nic1-9) for data traffic. The network interfaces for data traffic can follow any order.

Note The minimum number of network interfaces for ASAv are three data interfaces.

- For the console access, terminal server is supported through telnet.
- The following are the supported vCPU and memory parameters:

CPUs	Memory	ASAv Platform Size	License Type
1	2 GB	1vCPU/2 GB (default)	1G (ASAv10)
4	8 GB	4vCPU/8 GB	2G (ASAv30)
8	16 GB	8vCPU/16 GB	10G (ASAv50)
16	32 GB	16vCPU/32 GB	20G (ASAv100)

Supported Features

- Routed mode (Default)
- Transparent mode

Note Service chain in a multi-node cluster is not supported in transparent mode.

See the following concordance of Network Adapters, Source Networks, and Destination Networks for ASAv interfaces:

Network Adapter	Source Network	Destination Network	Function
vnic0	Management0-0	Management0/0	Management
vnic1	GigabitEthernet0-1	GigabitEthernet0/1	Outside
vnic2	GigabitEthernet0-2	GigabitEthernet0/2	Inside
vnic3-9	Data	Data	Data

ASAv on Proxmox VE

Proxmox Virtual Environment (VE) is an open-source server virtualization platform that can manage Nutanix virtual machines. Proxmox VE also provides a web-based management interface.

When you deploy the ASAv on Proxmox VE, you need to configure the VM to have an emulated serial port. Without the serial port, the ASAv will go into a loop during the startup process. All management tasks can be done using the Proxmox VE web-based management interface.

Note For advanced users who are used to the comfort of the Unix shell or Windows Powershell, Proxmox VE provides a command line interface to manage all the components of your virtual environment. This command line interface has intelligent tab completion and full documentation in the form of UNIX man pages.

To have the ASAv start properly, the VM needs to have a serial device configured:

1. In the main management center, select the ASAv VM in the left navigation tree.
2. Power off the virtual machine.
3. Choose **Hardware** > **Add** > **Network Device** and add a serial port.
4. Power on the virtual machine.
5. Access the ASAv VM using Xterm.js.

See the Proxmox Serial Terminal page for information on how to setup and activate the terminal on the guest/server.

Unsupported Features

- ASAv on Nutanix AHV does not support hot-plugging of interface. Do not try to add or remove interfaces when the ASAv is powered on.
- Nutanix AHV does not support Single Root I/O Virtualization (SR-IOV) or Data Plane Development Kit-Open vSwitch (DPDK-OVS).

Note Nutanix AHV supports in-guest DPDK using VirtIO. For more information, refer to DPDK support on AHV.

Related Documentation

- Nutanix Release Notes
- Nutanix Field Installation Guide
- Hardware Support on Nutanix
- Virtio-Net Multi-Queue support on Nutanix AHV

System Requirements

ASA Version

9.16.2

ASAv Memory, vCPU, and Disk Sizing

The specific hardware used for ASAv deployments can vary, depending on the number of instances deployed and usage requirements. Each instance of the ASAv requires a minimum resource allocation—amount of memory, number of CPUs, and disk space—on the server.

ASAv Licenses

- Configure all license entitlements for the security services from the ASAv CLI.
- See *ASAv: Configure Smart Software Licensing* in the Cisco ASA Configuration Guide for more information about how to manage licenses.

Nutanix Components and Versions

Component	Version
Nutanix Acropolis Operating System (AOS)	5.15.5 LTS and later
Nutanix Cluster Check (NCC)	4.0.0.1
Nutanix AHV	20201105.12 and later

How to Deploy the ASAv on Nutanix

Step	Task	More Information
1	Review the prerequisites.	Prerequisites, on page 261

Step	Task	More Information
2	Upload the ASAv qcow2 file to the Nutanix environment.	Upload the QCOW2 File to Nutanix, on page 261
3	Prepare a Day 0 configuration file with the initial configuration data that gets applied at the time of deploying a virtual machine.	Prepare the Day 0 Configuration File, on page 262
4	Deploy the ASAv on Nutanix.	Deploy the ASAv, on page 263
5	Launch the ASAv.	Launch the ASAv , on page 265

Prerequisites

- Download the ASAv qcow2 file from Cisco.com and put it on your Linux host:

  http://www.cisco.com/go/asa-software

Note A Cisco.com login and Cisco service contract are required.

- For ASA software and ASAv HyperFlex compatibility, see Cisco ASA Compatibility.

Upload the QCOW2 File to Nutanix

To deploy ASAv to the Nutanix environment, you must create an image from the qcow2 disk file in the Prism Web Console.

Before you begin

Download the qcow2 disk file from Cisco.com: https://software.cisco.com/download/navigator.html

Step 1 Log in to the Nutanix Prism Web Console.

Step 2 Click the gear icon to open the **Settings** page.

Step 3 Click **Image Configuration** from the left pane.

Step 4 Click **Upload Image**.

Step 5 Create the image.

a. Enter a name for the image.

b. From the **Image Type** drop-down list, choose **DISK**.

c. From the **Storage Container** drop-down list, choose the desired container.

d. Specify the location of the qcow2 disk file.

You can either specify a URL (to import the file from a web server) or upload the file from your workstation.

e. Click **Save**.

Step 6 Wait until the new image appears in the **Image Configuration** page.

Prepare the Day 0 Configuration File

You can prepare a Day 0 configuration file before you deploy the ASAv. This file is a text file that contains the initial configuration data that gets applied at the time a virtual machine is deployed.

If you deploy with a Day 0 configuration file, the process allows you to perform the entire initial setup for ASAv appliance.

In the file, you can specify the following:

- A hostname for the system.
- A new administrator username and password for the admin account.
- The initial firewall mode; sets the initial firewall mode, either **routed** or **transparent**.

  If you plan to manage your deployment using the local, you can only enter **routed** for the firewall mode. You cannot configure transparent firewall mode interfaces using the ASAv device manager.

- ASDM to enable:
  - **http server enable**
  - **access-group all global**
  - **http 0.0.0.0 0.0.0.0 management**
- Access List
- Name-Server
- Network settings that allow the appliance to communicate on your management network.

Note You can either upload the Day 0 configuration file or copy and paste the content in the text box provided.

Step 1 Create a new text file using a text editor of your choice.

Step 2 Enter the configuration details in the text file as shown in the following sample:

Example:

```
ASA Version 9.16.2
!
console serial
interface management0/0
nameif management
security-level 100
ip address 192.168.1.2 255.255.255.0
no shutdown
interface gigabitethernet0/0
nameif inside
security-level 100
```

```
ip address 10.1.1.2 255.255.255.0
no shutdown
interface gigabitethernet0/1
nameif outside
security-level 0
ip address 198.51.100.2 255.255.255.0
no shutdown
http server enable
http 192.168.1.0 255.255.255.0 management
crypto key generate rsa modulus 1024
username AdminUser password paSSw0rd
ssh 192.168.1.0 255.255.255.0 management
aaa authentication ssh console LOCAL
```

The first line should begin with the ASA version. The day0-config should be a valid ASA configuration. The best way to generate the day0-config is to copy the relevant parts of a running config from an existing ASA or ASAv. The order of the lines in the day0-config is important and should match the order seen in an existing **show running-config** command output.

Day0-config possible configuration:

- Hostname
- Domain name
- Administrative password
- Interfaces
- IP addresses
- Static routes
- DHCP server
- Network address translation rules

Note The content of the Day 0 configuration file must be in JSON format. You must validate the text using a JSON validator tool.

Step 3 Save the file as `day0-config.txt`.

Step 4 Select the **Custom Script** option.

Step 5 Either you upload the `day0-config.txt` file or copy and paste the file in the text box provided.

Step 6 Repeat steps 1–3 to create unique default configuration files for each ASAv that you want to deploy.

Deploy the ASAv

Before you begin

Ensure that the image of the ASAv that you plan to deploy is appearing on the **Image Configuration** page.

Step 1 Log in to the Nutanix Prism Web Console.

Step 2 From the main menu bar, click the **View** drop-down list, and choose **VM**.

Step 3 On the VM Dashboard, click **Create VM**.

Step 4 Do the following:

a. Enter a name for the ASAv instance.

b. (Optional) Enter a description for the ASAv instance.

c. Select the timezone that you want the ASAv instance to use.

Step 5 Enter the compute details.

a. Enter the number of virtual CPUs to allocate to the ASAv instance.

b. Enter the number of cores that must be assigned to each virtual CPU.

c. Enter the amount of memory (in GB) to allocate to the ASAv instance.

Step 6 Attach a disk to the ASAv instance.

a. Under **Disks**, click **Add New Disk**.

b. From the **Type** drop-down list, choose **DISK**.

c. From the **Operation** drop-down list, choose **Clone from Image Service**.

d. From the **Bus Type** drop-down list, choose **SATA**.

e. From the **Image** drop-down list, choose the image that you want to use.

f. Click **Add**.

Step 7 Configure at least three virtual network interfaces.

Under **Network Adapters (NIC)**, click **Add New NIC**, select a network, and click **Add**.

Repeat this process to add more network interfaces.

The ASAv on Nutanix supports a total of ten interfaces—One management interface and a maximum of nine network interfaces for data traffic. The interface-to-network assignments must be ordered as follows:

- vnic0—Management interface (required)
- vnic1—Outside interface (required)
- vnic2—Inside interface (required)
- vnic3-9—Data interface (optional)

Step 8 Configure affinity policy for the ASAv.

Under **VM Host Affinity**, click **Set Affinity**, select the hosts, and click **Save**.

Select more than one host to ensure that the VM can run even if there is a node failure.

Step 9 If you have prepared a Day 0 configuration file, do the following:

a. Select **Custom Script**.

b. Click **Upload A File**, and choose the Day 0 configuration file `day0-config.txt` or copy and paste the content into a text box.

Note All the other custom script options are not supported in release.

Step 10 Click **Save** to deploy the ASAv instance. The instance appears in the VM table view.

Step 11 In the VM table view, select the newly created instance, and click **Power On**.

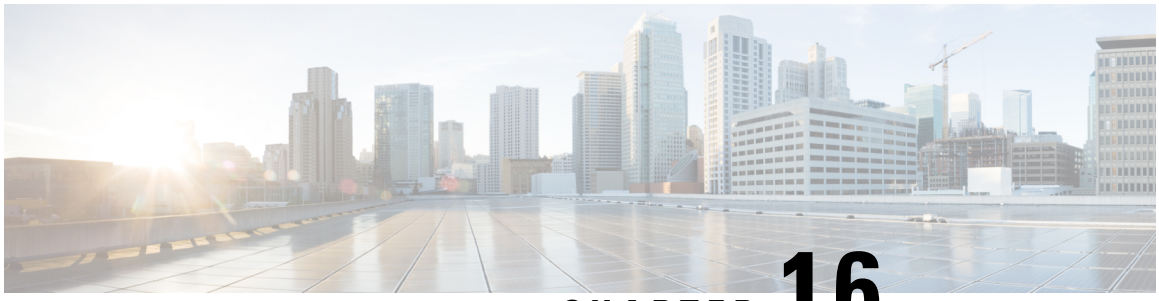
Launch the ASAv

Once the VM is powered on, select the **ASAv-VM** > **Launch Console** with predefined username and password using `day0-config` file for you to access it.

Note To change any of these settings for a virtual device after you complete the initial setup, you must use the CLI.

Step 1 Click on **Launch Console** to access the deployed ASAv.

Step 2 At the **asav login** prompt, log in with the `day0-config` **username** and the **password**.



CHAPTER 16

Deploy the ASAv on Cisco HyperFlex

HyperFlex systems deliver hyperconvergence for any application, and anywhere. HyperFlex with Cisco Unified Computing System (Cisco UCS) technology that is managed through the Cisco Intersight cloud operations platform can power applications and data anywhere, optimize operations from a core datacenter to the edge and into public clouds, and therefore increase agility through accelerating DevOps practices.

This chapter describes how the ASAv functions within a Cisco HyperFlex environment, including feature support, system requirements, guidelines, and limitations.

Important The minimum memory requirement for the ASAv is 2GB. If your current ASAv runs with less than 2GB of memory, you cannot upgrade to 9.13(1)+ from an earlier version without increasing the memory of your ASAv VM. You can also redeploy a new ASAv VM with the latest version.

Guidelines and Limitations

You can create and deploy multiple instances of the ASAv Cisco HyperFlex on a VMware vCenter server. The specific hardware used for ASAv deployments can vary, depending on the number of instances deployed and usage requirements. Each virtual appliance you create requires a minimum resource allocation—memory, number of CPUs, and disk space—on the host machine.

Important The ASAv deploys with a disk storage size of 8 GB. It is not possible to change the resource allocation of the disk space.

Review the following guidelines and limitations before you deploy the ASAv.

Recommended vNICs

For optimal performance, we recommend that you use vmxnet3 vNIC. This vNIC is a para-virtualized network driver that supports 10 Gbps operation but also requires CPU cycles. In addition, when using vmxnet3, disable Large Receive Offload (LRO) to avoid poor TCP performance.

OVF File Guidelines

- asav-vi.ovf—For deployment on vCenter
- The ASAv OVF deployment does not support localization (installing the components in non-English mode). Be sure that the VMware vCenter and the LDAP servers in your environment are installed in an ASCII-compatible mode.
- You must set your keyboard to United States English before installing the ASAv and for using the VM console.

Failover for High Availability Guidelines

For failover deployments, make sure that the standby unit has the same license entitlement; for example, both units should have the 2 Gbps entitlement.

Important When creating a high availability pair using ASAv, you must add the data interfaces to each ASAv in the same order. If you have added the exact same interfaces are added to each ASAv, but in different order, you might see errors at the ASAv console. Failover functionality may also be affected.

IPv6 Guidelines

You cannot specify IPv6 addresses for the management interface when you first deploy the ASAv OVF file using the VMware vSphere Web Client; you can later add IPv6 addressing using ASDM or the CLI.

vMotion Guidelines

- VMware requires you to use only shared storage if you are using vMotion. During ASAv deployment, if you have a host cluster you can either provision storage locally (on a specific host) or on a shared host. However, if you try to vMotion the ASAv to another host, using local storage will produce an error.

Memory and vCPU Allocation for Throughput and Licensing

- The memory allocated to the ASAv is sized specifically for the throughput level. Do not change the memory setting or any vCPU hardware settings in the **Edit Settings** dialog box unless you are requesting a license for a different throughput level. Under-provisioning can affect performance.

Note If you need to change the memory or vCPU hardware settings, use only the values documented in Licensing for the ASAv, on page 1. Do not use the VMware-recommended memory configuration minimum, default, and maximum values.

CPU Reservation

- By default the CPU reservation for the ASAv is 1000 MHz. You can change the amount of CPU resources allocated to the ASAv by using the shares, reservations, and limits settings. **Edit Settings** > **Resources** > **CPU**. Lowering the CPU Reservation setting from 1000 MHz can be done if the ASAv can perform its required purpose while under the required traffic load with the lower setting. The amount of CPU used by an ASAv depends on the hardware platform it is running on as well as the type and amount of work it is doing.

  You can view the host's perspective of CPU usage for all of your virtual machines from the CPU Usage (MHz) chart, located in the Home view of the Virtual Machine Performance tab. Once you establish a benchmark for CPU usage when the ASAv is handling typical traffic volume, you can use that information as input when adjusting the CPU reservation.

  For More information, see the link CPU Performance Enhancement Advice
- You can view the resource allocation and any resources that are over- or under-provisioned using the ASAv**show vm** > **show cpu**

  commands or the ASDM

  Home > **Device Dashboard** > **Device Information** > **Virtual Resources**

  tab or the

  Monitoring > **Properties** > **System Resources Graphs** > **CPU** pane

Transparent Mode on UCS B and C Series Hardware Guidelines

MAC flaps have been observed in some ASAv configurations running in transparent mode on Cisco UCS B (Compute Nodes) and C (Converged Nodes) Series hardware. When MAC addresses appear from different locations you will get dropped packets.

The following guidelines help to prevent MAC flaps when you deploy the ASAv in transparent mode in VMware environments:

- VMware NIC teaming—If deploying the ASAv in transparent mode on UCS B or C Series, the port groups used for the inside and outside interfaces must have only 1 Active uplink, and that uplink must be the same. Configure VMware NIC teaming in vCenter.
- ARP inspection—Enable ARP inspection on the ASAv and statically configure the MAC and ARP entry on the interface that you expect to receive it on. See the Cisco ASA Series General Operations Configuration Guide for information about ARP inspection and how to enable it.

System Requirements

Configurations and Clusters for HyperFlex HX-Series

Configurations	Clusters
HX220c converged nodes	• Flash cluster • Minimum of 3-node Cluster (Databases, VDI, VSI)

Configurations	Clusters
HX240c converged nodes	• Flash cluster • Minimum of 3-node Cluster (VSI: IT/Biz Apps, Test/Dev)
HX220C and Edge (VDI, VSI, ROBO) HX240C (VDI, VSI, Test/Dev)	• Hybrid cluster • Minimum of 3-node Cluster
B200 + C240/C220	Compute bound apps/VDI

Deployment options for the HyperFlex HX-Series:

- Hybrid Cluster
- Flash Cluster
- HyperFlex HX Edge
- SED drives
- NVME Cache
- GPUs

For HyperFlex HX cloud powered management option, refer to the *Deploying HyperFlex Fabric Interconnect-attached Clusters* section in the Cisco HyperFlex Systems Installation Guide.

HyperFlex Components and Versions

Component	Version
VMware vSphere	7.0.2-18426014
HyperFlex Data Platform	4.5.2a-39429

Supported Features

- Deployment Modes—Routed (Standalone), Routed (HA), and Transparent
- ASAv native HA
- Jumbo frames
- VirtIO
- HyperFlex Data Center Clusters (excluding Stretched Clusters)
- HyperFlex Edge Clusters
- HyperFlex All NVMe, All Flash and Hybrid converged nodes
- HyperFlex Compute-only Nodes

Unsupported Features

ASAv running with SR-IOV has not been qualified with HyperFlex.

Note HyperFlex supports SR-IOV, but requires a PCI-e NIC in addition to the MLOM VIC

Deploy the ASAv

Step	Task	More Information
1	Review the Guidelines and Limitations.	Guidelines and Limitations, on page 267
2	Review the prerequisites.	Prerequisites for the ASAv and Cisco HyperFlex, on page 271
3	Download the OVF file from cisco.com.	Download and Unpack the ASAv Software, on page 272
4	Deploy the ASAv on Cisco HyperFlex.	Deploy the ASAv on Cisco HyperFlex to vSphere vCenter, on page 272
5	Access the ASAv Console.	Access the ASAv Console, on page 275

Prerequisites for the ASAv and Cisco HyperFlex

You can deploy the ASAv on Cisco HyperFlex using the VMware vSphere Web Client, vSphere standalone client, or the OVF tool. See Cisco ASA Compatibility for system requirements.

Security Policy for a vSphere Standard Switch

For a vSphere switch, you can edit Layer 2 security policies and apply security policy exceptions for port groups used by the ASAv interfaces. See the following default settings:

- Promiscuous Mode: **Reject**
- MAC Address Changes: **Accept**
- Forged Transmits: **Accept**

You may need to modify these settings for the following ASAv configurations. See the vSphere documentation for more information.

Table 30: Port Group Security Policy Exceptions

Security Exception	Routed Firewall Mode		Transparent Firewall Mode	
	No Failover	Failover	No Failover	Failover
Promiscuous Mode	<any>	<any>	Accept	Accept

Security Exception	Routed Firewall Mode		Transparent Firewall Mode	
	No Failover	Failover	No Failover	Failover
MAC Address Changes	<any>	Accept	<any>	Accept
Forged Transmits	<any>	Accept	Accept	Accept

Download and Unpack the ASAv Software

Before you begin

You must have at least one network configured in vSphere (for management) before you deploy the ASAv.

Step 1 Download the ZIP file from Cisco.com, and save it to your local disk:

https://www.cisco.com/go/asa-software

Note A Cisco.com login and Cisco service contract are required.

Step 2 Unzip the file into a working directory. Do not remove any files from the directory. The following files are included:

- asav-vi.ovf—For vCenter deployments.
- boot.vmdk—Boot disk image.
- disk0.vmdk—ASAv disk image.
- day0.iso—An ISO containing a day0-config file and optionally an idtoken file.
- asav-vi.mf—Manifest file for vCenter deployments.

Deploy the ASAv on Cisco HyperFlex to vSphere vCenter

Use this procedure to deploy the ASAv on HyperFlex to VMware vSphere vCenter. You can use the VMware Web Client (or vSphere Client) to deploy and configure virtual machines.

Before you begin

You must have at least one network configured in vSphere (for management) before you deploy the ASAv on HyperFlex.

Before ASAv to be installed on the HyperFlex cluster, the HyperFlex cluster and shared datastore must be created. See the HyperFlex configuration guide for more information.

Step 1 Log in to the vSphere Web Client.

Step 2 Using the vSphere Web Client, deploy the OVF template file that you downloaded earlier by clicking **ACTIONS** > **Deploy OVF Template**.

The Deploy OVF Template wizard appears.

Step 3 Browse your file system for the OVF template source location, and then click **NEXT**.

Step 4 Review the **OVF Template Details** page and verify the OVF template information (product name, version, vendor, download size, size on disk, and description), and then click **NEXT**.

Step 5 The **End User License Agreement** page appears. Review the license agreement packaged with the OVF template (VI templates only), click **Accept** to agree to the terms of the licenses and click **NEXT**.

Step 6 On the **Name and Location** page, enter a name for this deployment and select a location in the inventory (Shared datastore or cluster) on which you want to deploy the HyperFlex, and then click **NEXT**. The name must be unique within the inventory folder and can contain up to 80 characters.

The vSphere Web Client presents the organizational hierarchy of managed objects in inventory views. Inventories are the hierarchal structure used by vCenter Server or the host to organize managed objects. This hierarchy includes all of the monitored objects in vCenter Server.

Step 7 Navigate to, and select the resource pool where you want to run the ASAv HyperFlex and click **NEXT**.

Note This page appears only if the cluster contains a resource pool. For the compute resource pool, we only recommend the cluster for best performance

Step 8 Select a **Deployment Configuration**. Choose one of the three supported vCPU/memory values from the **Configuration** drop-down list, and click **NEXT**.

Step 9 Select a **Storage** location to store the virtual machine files, and then click **NEXT.**

On this page, select the datastores (The datastore is HX cluster shared datastore that created with HX connect) that is already configured on the destination cluster. The virtual machine configuration file and virtual disk files are stored on the datastore. Select a datastore large enough to accommodate the virtual machine and all of its virtual disk files.

Step 10 On the **Network Mapping** page, map the networks specified in the OVF template to networks in your inventory, and then select **NEXT**.

Ensure the Management0-0 interface is associated with a VM Network that is reachable from the Internet. Non-management interfaces are configurable from either a ASAv Mangement Centre or a ASAv Device Manager, depending on your management mode.

Important ASAv on HyperFlex now defaults to vmxnet3 interfaces when you create a virtual device. Previously, the default was e1000. If you are using e1000 interfaces, we **strongly recommend** you to switch. The vmxnet3 device drivers and network processing are integrated with the HyperFlex, so they use fewer resources and offer better network performance.

The networks may not be in alphabetical order. If it is too difficult to find your networks, you can change the networks later from the **Edit Settings** dialog box. After you deploy, right-click the instance, and choose **Edit Settings**. However, the network mapping page does not show the IDs (only Network Adapter IDs).

See the following concordance of Network Adapter, and Source Networks and Destination Networks for interfaces (note these are the default vmxnet3 interfaces):

Table 31: Source to Destination Network Mapping—VMXNET3

Network Adapter ID	ASAv Interface ID
Network Adapter 1	Management 0/0
Network Adapter 2	GigabitEthernet 0/0

Network Adapter ID	ASAv Interface ID
Network Adapter 3	GigabitEthernet 0/1
Network Adapter 4	GigabitEthernet 0/2
Network Adapter 5	GigabitEthernet 0/3
Network Adapter 6	GigabitEthernet 0/4
Network Adapter 7	GigabitEthernet 0/5
Network Adapter 8	GigabitEthernet 0/6
Network Adapter 9	GigabitEthernet 0/7
Network Adapter 10	GigabitEthernet 0/8

You can have a total of 10 interfaces when you deploy the ASAv. For data interfaces, make sure that the Source Networks map to the correct Destination Networks, and that each data interface maps to a unique subnet or VLAN. You do not have to use all interfaces; for interfaces you do not intend to use, they can remain disabled within the configuration.

Step 11 On the **Properties** page, set the user-configurable properties packaged with the OVF template (VI templates only):

- **Password**—Set the password for admin access.
- **Network**—Set the network information, including the Fully Qualified Domain Name (FQDN), DNS, search domain, and network protocol (IPv4 or IPv6).
- **Management Interface**—Set the management configuration and then click the drop down select **DHCP/Manual** and set the ip configuration for the management interface.
- **Firewall Mode**—Set the initial firewall mode. Click the drop-down arrow for **Firewall Mode** and choose one of the two supported modes, either **Routed** or **Transparent.**

Step 12 Click **NEXT**. In the **Ready to Complete** section, review and verify the displayed information. To begin the deployment with these settings, click **Finish**. To make any changes, click **Back** to navigate back the previous dialog boxes.

(Optional) check the **Power on after deployment** option to power on the VM, then click **Finish**.

After you complete the wizard, the vSphere Web Client processes the virtual machine; you can see the Initialize OVF deployment status in the **Global Information** area **Recent Tasks** pane.

When it is finished, you see the Deploy OVF Template completion status.

The ASAv instance appears under the specified data center in the Inventory. Starting the new VM could take up to 30 minutes.

Note You require Internet access to successfully register the ASAv HyperFlex with the Cisco Licensing Authority. You might need to perform additional configuration after deployment to achieve Internet access and successful license registration.

Access the ASAv Console

In some cases with ASDM, you may need to use the CLI for troubleshooting. By default, you can access the built-in VMware vSphere console. Alternatively, you can configure a network serial console, which has better capabilities, including copy and paste.

- Use the VMware vSphere Console
- Configure a Network Serial Console Port

Use the VMware vSphere Console

For initial configuration or troubleshooting, access the CLI from the virtual console provided through the VMware vSphere Web Client. You can later configure CLI remote access for Telnet or SSH.

Before you begin

For the vSphere Web Client, install the Client Integration Plug-In, which is required for ASAv console access.

Step 1 In the VMware vSphere Web Client, right-click the ASAv instance in the Inventory, and choose **Open Console**. Or you can click **Launch Console** on the Summary tab.

Step 2 Click in the console and press **Enter**. Note: Press **Ctrl + Alt** to release the cursor.

If the ASAv is still starting up, you see bootup messages.

When the ASAv starts up for the first time, it reads parameters provided through the OVF file and adds them to the ASAv system configuration. It then automatically restarts the boot process until it is up and running. This double boot process only occurs when you first deploy the ASAv.

Note Until you install a license, throughput is limited to 100 Kbps so that you can perform preliminary connectivity tests. A license is required for regular operation. You also see the following messages repeated on the console until you install a license:

```
Warning: ASAv platform license state is Unlicensed.
Install ASAv platform license for full functionality.
```

You see the following prompt:

```
ciscoasa>
```

This prompt indicates that you are in user EXEC mode. Only basic commands are available from user EXEC mode.

Step 3 Access privileged EXEC mode:

Example:

```
ciscoasa> enable
```

The following prompt appears:

```
Password:
```

Step 4 Press the **Enter** key to continue. By default, the password is blank. If you previously set an enable password, enter it instead of pressing Enter.

The prompt changes to:

```
ciscoasa#
```

All nonconfiguration commands are available in privileged EXEC mode. You can also enter configuration mode from privileged EXEC mode.

To exit privileged mode, enter the **disable**, **exit**, or **quit** command.

Step 5 Access global configuration mode:

```
ciscoasa# configure terminal
```

The prompt changes to the following:

```
ciscoasa(config)#
```

You can begin to configure the ASAv from global configuration mode. To exit global configuration mode, enter the **exit**, **quit**, or **end** command.

Configure a Network Serial Console Port

For a better console experience, you can configure a network serial port singly or attached to a virtual serial port concentrator (vSPC) for console access. See the VMware vSphere documentation for details about each method. On the ASAv, you must send the console output to a serial port instead of to the virtual console. This procedure describes how to enable the serial port console.

Step 1 Configure a network serial port in VMware vSphere. See the VMware vSphere documentation.

Step 2 On the ASAv, create a file called "use_ttyS0" in the root directory of disk0. This file does not need to have any contents; it just needs to exist at this location:

disk0:/use_ttyS0

- From ASDM, you can upload an empty text file by that name using the **Tools** > **File Management** dialog box.
- At the vSphere console, you can copy an existing file (any file) in the file system to the new name. For example:

```
ciscoasa(config)# cd coredumpinfo
ciscoasa(config)# copy coredump.cfg disk0:/use_ttyS0
```

Step 3 Reload the ASAv.

- From ASDM, choose **Tools** > **System Reload**.
- At the vSphere console, enter **reload**.

  The ASAv stops sending to the vSphere console, and instead sends to the serial console.

Step 4 Telnet to the vSphere host IP address and the port number you specified when you added the serial port; or Telnet to the vSPC IP address and port.

Upgrade the vCPU or Throughput License

The ASAv uses a throughput license, which affects the number of vCPUs you can use.

If you want to increase (or decrease) the number of vCPUs for your ASAv, you can request a new license, apply the new license, and change the VM properties in VMware to match the new values.

Note The assigned vCPUs must match the ASAv Virtual CPU license or throughput license. The RAM must also be sized correctly for the vCPUs. When upgrading or downgrading, be sure to follow this procedure and reconcile the license and vCPUs immediately. The ASAv does not operate properly when there is a persistent mismatch.

Step 1 Request a new ASAv Virtual CPU license or throughput license.

Step 2 Apply the new license. For failover pairs, apply new licenses to both units.

Step 3 Do one of the following, depending on whether you use failover:

- Failover—In the vSphere Web Client, power off the standby ASAv. For example, click the ASAv and then click **Power Off the virtual machine**, or right-click the ASAv and choose **Shut Down Guest OS**.
- No Failover—In the vSphere Web Client, power off the ASAv. For example, click the ASAv and then click **Power Off the virtual machine**, or right-click the ASAv and choose **Shut Down Guest OS**.

Step 4 Click the ASAv, and then click **Edit Virtual machine settings** (or right-click the ASAv and choose **Edit Settings**).

The **Edit Settings** dialog box appears.

Step 5 Refer to the CPU and memory requirements in Licensing for the ASAv, on page 1 to determine the correct values for the new vCPU license.

Step 6 On the **Virtual Hardware** tab, for the **CPU**, choose the new value from the drop-down list.

Step 7 For the **Memory**, enter the new value for the RAM.

Step 8 Click **OK**.

Step 9 Power on the ASAv. For example, click **Power On the Virtual Machine**.

Step 10 For failover pairs:

a. Open a console to the active unit or launch ASDM on the active unit.

b. After the standby unit finishes starting up, failover to the standby unit:

- ASDM: Choose **Monitoring** > **Properties** > **Failover** > **Status**, and click **Make Standby**.
- CLI: **failover active**

c. Repeat Steps 3 through 9 for the active unit.

What to do next

See Licensing for the ASAv, on page 1 for more information.

Performance Tuning

The ASAv is a high-performance appliance but may require tuning of the Cisco HyperFlex to achieve the best results.

The following is the best practice and recommendation for facilitating the best performance of the ASAv in a HyperFlex environment.

Enabling Jumbo Frames

A larger MTU allows you to send larger packets. Larger packets might be more efficient for your network. See the following guidelines:

- Matching MTUs on the traffic path—We recommend that you set the MTU on all ASAv interfaces and other device interfaces along the traffic path to be the same. Matching MTUs prevents intermediate devices from fragmenting the packets.
- Accommodating jumbo frames—You can set the MTU up to 9198 bytes. The maximum is 9000 for the ASAv.

This procedure explains how to enable jumbo frames in the following environment:

HyperFlex Cluster on the vSphere 7.0.1 > **VMware vSphere vSwitch**> **Cisco UCS Fabric Interconnects (FI).**

Step 1 Change the MTU settings of the ASAv host where you have deployed the ASAv.

a. Connect to the vCenter Server using the vSphere Web Client.

b. In the **Advanced System Settings** of your HyperFlex host, set the value of the configuration parameter—`Net.Vmxnet3NonTsoPacketGtMtuAllowed` to `1`.

c. Save the changes and reboot the host.

For more information, see https://kb.vmware.com/s/article/1038578.

Step 2 Change the MTU settings of the VMware vSphere vSwitch.

a. Connect to the vCenter Server using the vSphere Web Client.

b. Edit the properties of the VMware vSphere vSwitch, and set the value of **MTU** to `9000`.

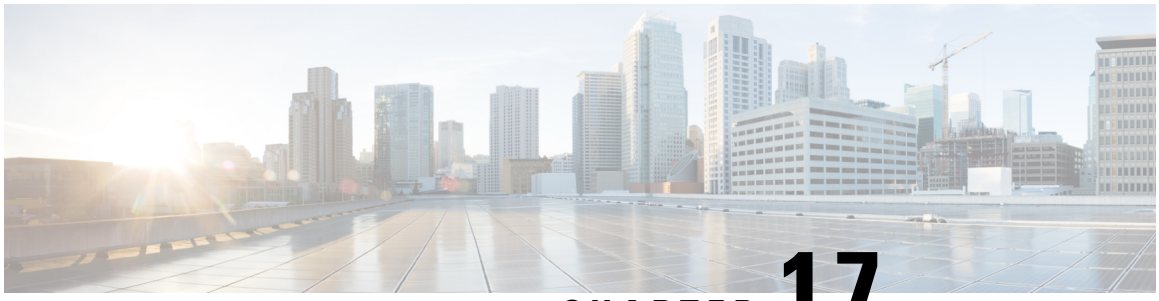
Step 3 Change the MTU settings of the Cisco UCS Fabric Interconnects (FI).

a. Log in to the Cisco UCS Management console.

b. To Edit QoS System Class, choose **LAN** > **LAN Cloud** > **QoS System Class.** Under the **General** tab, set the value of **MTU** to 9216.

c. To edit your vNIC, choose **LAN** > **Policies** > **root** > **Sub-Organizations**

<your-hyperflex-org>**vNIC Templates** <your-vnic>. Under the **General** tab, set the value of **MTU** to 9000.



CHAPTER 17

Configure the ASAv

The ASAv deployment preconfigures ASDM access. From the client IP address you specified during deployment, you can connect to the ASAv management IP address with a web browser. This chapter also describes how to allow other clients to access ASDM and also how to allow CLI access (SSH or Telnet). Other essential configuration tasks covered in this chapter include the license installation and common configuration tasks provided by wizards in ASDM.

Start ASDM

Step 1 On the PC that you specified as the ASDM client, enter the following URL:

https://*asa_ip_address*/**admin**

The ASDM launch window appears with the following buttons:

- **Install ASDM Launcher and Run ASDM**
- **Run ASDM**
- **Run Startup Wizard**

Step 2 To download the Launcher:

a) Click **Install ASDM Launcher and Run ASDM**.

b) Leave the username and password fields empty (for a new installation), and click **OK**. With no HTTPS authentication configured, you can gain access to ASDM with no username and the **enable** password, which is blank by default. If you enabled HTTPS authentication, enter your username and associated password.

c) Save the installer to your PC, and then start the installer. The ASDM-IDM Launcher opens automatically after installation is complete.

d) Enter the management IP address, leave the username and password blank (for a new installation), and then click **OK**. If you enabled HTTPS authentication, enter your username and associated password.

Step 3 To use Java Web Start:

a) Click **Run ASDM** or **Run Startup Wizard**.

b) Save the shortcut to your computer when prompted. You can optionally open it instead of saving it.

c) Start Java Web Start from the shortcut.
d) Accept any certificates according to the dialog boxes that appear. The Cisco ASDM-IDM Launcher appears.
e) Leave the username and password blank (for a new installation), and then click **OK**. If you enabled HTTPS authentication, enter your username and associated password.

Perform Initial Configuration Using ASDM

You can perform initial configuration using the following ASDM wizards and procedures.

- Run the Startup Wizard
- (Optional) Allow Access to Public Servers Behind the ASAv
- (Optional) Run VPN Wizards
- (Optional) Run Other Wizards in ASDM

For CLI configuration, see the Cisco ASA Series CLI configuration guides.

Run the Startup Wizard

Run the **Startup Wizard** to customize the security policy to suit your deployment.

Step 1 Choose **Wizards** > **Startup Wizard**.

Step 2 Customize the security policy to suit your deployment. You can set the following:

- Hostname
- Domain name
- Administrative passwords
- Interfaces
- IP addresses
- Static routes
- DHCP server
- Network address translation rules
- and more ...

(Optional) Allow Access to Public Servers Behind the ASAv

The **Configuration** > **Firewall** > **Public Servers** pane automatically configures the security policy to make an inside server accessible from the Internet. As a business owner, you might have internal network services,

such as a web and FTP server, that need to be available to an outside user. You can place these services on a separate network behind the ASAv, called a demilitarized zone (DMZ). By placing the public servers on the DMZ, any attacks launched against the public servers do not affect your inside networks.

(Optional) Run VPN Wizards

You can configure VPN using the following wizards (**Wizards** > **VPN Wizards**):

- Site-to-Site VPN Wizard—Creates an IPsec site-to-site tunnel between the ASAv and another VPN-capable device.
- AnyConnect VPN Wizard—Configures SSL VPN remote access for the Cisco AnyConnect VPN client. AnyConnect Client provides secure SSL connections to the ASA for remote users with full VPN tunneling to corporate resources. You can configure the ASA policy to download the AnyConnect Client to remote users when they initially connect through a browser. With AnyConnect Client 3.0 and later, the client can run either the SSL or IPsec IKEv2 VPN protocol.
- Clientless SSL VPN Wizard—Configures clientless SSL VPN remote access for a browser. Clientless, browser-based SSL VPN lets users establish a secure, remote-access VPN tunnel to the ASA using a web browser. After authentication, users access a portal page and can access specific, supported internal resources. The network administrator provides access to resources by users on a group basis. ACLs can be applied to restrict or allow access to specific corporate resources.
- IPsec (IKEv1 or IKEv2) Remote Access VPN Wizard—Configures IPsec VPN remote access for the Cisco IPsec client.

For information on how to configure an ASAv IPsec Virtual Tunnel Interface (VTI) connection to Azure, see Configure ASA IPsec VTI Connection to Azure.

(Optional) Run Other Wizards in ASDM

You can run other wizards in ASDM to configure failover with high availability, VPN cluster load balancing, and packet capture.

- High Availability and Scalability Wizard—Configure failover or VPN load balancing.
- Packet Capture Wizard—Configure and run packet capture. The wizard runs one packet capture on each of the ingress and egress interfaces. After capturing packets, you can save the packet captures to your PC for examination and replay in the packet analyzer.

Advanced Configuration

To continue configuring your ASAv, see Navigating the Cisco ASA Series Documentation.