Securing Applications in AWS

REFERENCE ARCHITECTURE GUIDE

JUNE 2020

Table of Contents

Preface 1

Purpose of This Guide 3

- Audience 3
- Related Documentation 4

Introduction 5

Public Cloud Concepts 6

- Scaling Methods 6
- Reduced Time to Deployment 6
- Security Integration 7
- Cloud Infrastructure Protection 8

AWS Concepts and Services 10

- AWS Top-Level Concepts 10
- Virtual Private Cloud 11
- Virtual Compute 15
- Accessing AWS Virtual Networks 16
- Connecting VPCs 20
- Resiliency Constructs 23

Palo Alto Networks Design Details 25

- VM-Series Firewall on AWS 25
- VM-Series Firewall Integration to AWS 28
- Networking in the VPC 33
- AWS Traffic Flows 37

Design Models 47

- Choosing a Design Model 47
- Single VPC Design Model 48
- Transit Gateway Design Model 52

Summary 64

Preface

GUIDE TYPES

Overview guides provide high-level introductions to technologies or concepts.

Reference architecture guides provide an architectural overview for using Palo Alto Networks® technologies to provide visibility, control, and protection to applications built in a specific environment. These guides are required reading prior to using their companion deployment guides.

Deployment guides provide decision criteria for deployment scenarios, as well as procedures for combining Palo Alto Networks technologies with third-party technologies in an integrated design.

DOCUMENT CONVENTIONS

Notes provide additional information.

Cautions warn about possible data loss, hardware damage, or compromise of security.

Blue text indicates a configuration variable for which you need to substitute the correct value for your environment.

In the **IP** box, enter `10.5.0.4/24`, and then click **OK**.

Bold text denotes:

- Command-line commands.

```
# show device-group branch-offices
```

- User-interface elements.

  In the **Interface Type** list, choose **Layer 3**.

- Navigational paths.

  Navigate to **Network > Virtual Routers**.

- A value to be entered.

  Enter the password **admin**.

Italic text denotes the introduction of important terminology.

> An *external dynamic list* is a file hosted on an external web server so that the firewall can import objects.

Highlighted text denotes emphasis.

> Total valid entries: 755

ABOUT PROCEDURES

These guides sometimes describe other companies' products. Although steps and screen-shots were up-to-date at the time of publication, those companies might have since changed their user interface, processes, or requirements.

GETTING THE LATEST VERSION OF GUIDES

We continually update reference architecture and deployment guides. You can access the latest version of this and all guides at this location:

https://www.paloaltonetworks.com/referencearchitectures

WHAT'S NEW IN THIS RELEASE

Palo Alto Networks made the following changes since the last version of this guide:

- Modified the VM-Series firewall and Panorama™ instance sizes tables
- Removed the Transit VPC design model
- Updated PAYG information for multiple VM-Series capacity licenses
- Updated the guide for brand changes and readability

Purpose of This Guide

This guide describes reference architectures for securing applications using Palo Alto Networks VM-Series virtual next-generation firewalls on Amazon Web Services (AWS).

This guide:

- Provides an architectural overview for using VM-Series firewalls to provide visibility, control, and protection to your applications built on AWS.
- Links the technical design aspects of AWS and the Palo Alto Networks solutions and then explores several technical design models. The design models include two options that span the scale of enterprise-level operational environments.
- Provides a framework for architectural discussions between Palo Alto Networks and your organization.
- Is required reading prior to using the Palo Alto Networks deployment guides for AWS. The deployment guides provide decision criteria for deployment scenarios, as well as procedures for enabling features of the AWS and the Palo Alto Networks VM-Series firewalls in order to achieve an integrated design.

AUDIENCE

This guide is written for technical readers, including system architects and design engineers, who want to deploy the Palo Alto Networks VM-Series firewalls and Panorama within a public cloud data center infrastructure. It assumes the reader is familiar with the basic concepts of applications, networking, virtualization, security, and high availability, as well as a basic understanding of network and data center architectures.

To be successful, you must have a working knowledge of networking and policy in PAN-OS®.

RELATED DOCUMENTATION

The following documents support this guide:

- Securing Applications in AWS Single VPC: Deployment Guide—Details the deployment of the Single Virtual Private Cloud (VPC) design model, which is well-suited for initial deployments and proof of concepts of Palo Alto Networks VM-Series firewalls on AWS. This guide describes deploying VM-Series firewalls to provide visibility and protection for the VPC's inbound and outbound traffic.

- Securing Applications in AWS Transit Gateway: Deployment Guide—Details the deployment of the Transit Gateway design model. This model provides a hub-and-spoke design for centralized and scalable firewall services for resilient inbound, outbound, and east-west traffic flows. This guide describes deploying the VM-Series firewalls to provide protection and visibility for traffic flowing through the transit gateway.

- Panorama on AWS: Deployment Guide—Details the deployment of Palo Alto Networks Panorama management nodes in the AWS VPC. This guide includes setup of Panorama in a high-availability configuration and setup of Cortex™ Data Lake.

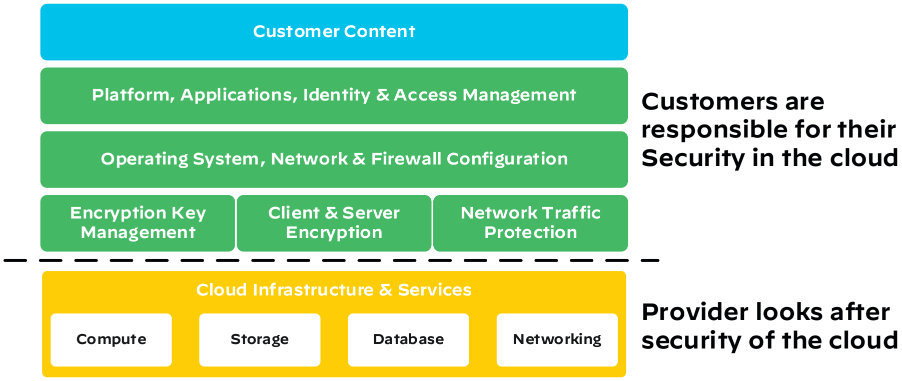
Introduction

Organizations are adopting AWS in order to deploy applications and services on a public cloud infrastructure for a variety of reasons, including:

- **Business agility**—Infrastructure resources are available when and where they are needed, minimizing IT staffing requirements and providing faster, predictable time-to-market. Virtualization in both public and private cloud infrastructures has permitted IT to respond to business requirements within minutes instead of days or weeks.
- **Better use of resources**—Projects are more efficient, and there are fewer operational issues, permitting employees to spend more time adding business value. Employees have the resources needed to bring value to the organization.
- **Operational vs capital expenditure**—Costs are aligned directly with usage, providing a utility model for IT infrastructure that requires little-to-no capital expense. Gone are the large capital expenditures and time delays associated with building private data center infrastructure.

Although Infrastructure as a Service (IaaS) providers are responsible for ensuring the security and availability of their infrastructure, ultimately, organizations are still responsible for the security of their applications and data. The security requirements are similar to on-premises deployments, but the specific implementation details of how to properly architect security technology in a public cloud environment, such as Google Cloud Platform, are different.

Figure 1 Security responsibility in the IaaS environment


Customer Content
Platform, Applications, Identity & Access Management
Operating System, Network & Firewall Configuration
Encryption Key Management
Client & Server Encryption
Network Traffic Protection
Customers are responsible for their Security in the cloud
Cloud Infrastructure & Services
Compute
Storage
Database
Networking
Provider looks after security of the cloud

The VM-Series firewall is an integral security enforcement and intelligence gathering component of the Palo Alto Networks security solutions. The Palo Alto Networks VM-Series firewall deployed on AWS has the same features, benefits, and management as the PA-Series next-generation firewalls you might have deployed elsewhere in your organization. First and foremost, the application control and threat prevention capabilities of the VM-Series firewalls protect your AWS deployments from threats, data loss, and business disruption. Any observed and collected threat intelligence information is shared across other Palo Alto Networks solutions in order to improve threat prevention capabilities collectively and continually.

Public Cloud Concepts

Organizations generally move to the public cloud with the goals of increasing scale and reducing time to deployment. Achieving these goals requires application architectures that are built specifically for the public cloud. Before you can architect for the public cloud, you must understand how it is different from traditional on-premises environments.

SCALING METHODS

Traditionally, organizations scale on-premises deployments through the purchase of devices that have increased performance capacity. Scaling up an on-premises deployment in this method makes sense because organizations typically purchase the devices in order to satisfy the performance requirements during the devices' lifetime.

Public cloud environments focus on scaling out the deployment instead of scaling up. This architectural difference stems primarily from the capability of public cloud environments to dynamically increase or decrease the number of resources allocated to your environment. In the public cloud, infrastructure used to satisfy performance requirements can have a lifetime in minutes instead of years. Instead of purchasing extra capacity for use at some time in the future, the dynamic nature of the public cloud allows you to allocate just the right amount of resources required to service the application.

In practice, to architect an application for the cloud, you need to distribute functionality, and you should build each functional area to scale out as necessary. Typically, this means a load balancer distributes traffic across a pool of identically configured resources. When changes occur in the application traffic, the number of resources you have allocated to the pool can dynamically increase or decrease. This design method provides scale and resiliency. However, the application architecture must take into account that the resources are transient. For example, you should not store the application state in the networking infrastructure or in the frontend application servers. Instead, store state information on the client or persistent storage services.

The ability to scale a cloud architecture extends not only to the capacity of an application but also the capacity to deploy applications globally. Scaling an application to a new region in a traditional on-premises deployment requires significant investment and planning. Public cloud architectures are location-agnostic, and you can deploy them globally in a consistent amount of time.

REDUCED TIME TO DEPLOYMENT

To achieve the goal of reduced time to deployment, you must have a development and deployment process that is repeatable and reacts to changes quickly. DevOps workflows are the primary method for implementing this process. DevOps workflows are highly dependent on the ability to automate, as much as possible, the process of deploying a resource or application. In practice, this means you must be able to programmatically bootstrap, configure, update, and destroy the cloud infrastructure, as well as the resources running on it. Compared to traditional on-premises deployments where device deployment, configuration, and operation happen manually, automated workflows in a public cloud environment can significantly reduce time to deployment.

Automation is so core to cloud design that many cloud application architectures deploy new capabilities through the automated build-out of new resources instead of updating the existing ones. This type of cloud architecture provides several benefits, including the ability to phase in the changes to a subset of the traffic and the ability to quickly roll back the changes by redirecting traffic from the new resources to the old.

SECURITY INTEGRATION

VM-Series firewalls enable you to securely implement scalable cloud architectures and reduce time to deployment. You leverage the following capabilities of VM-Series firewalls in order to achieve this:

- **Application visibility**—VM-Series firewalls natively analyze all traffic in a single pass to determine the application, content, and user identity. The application, content, and user are core elements of your security policy that are used for visibility, reporting, and incident investigation.
- **Advanced attack prevention at the application level**—Attacks, much like many applications, can use any port, rendering traditional prevention mechanisms ineffective. VM-Series firewalls allow you to use Threat Prevention and the WildFire® cloud-based threat analysis services. These services apply application-specific threat prevention policies that block exploits, malware, and previously unknown threats from infecting your cloud.
- **Consistent policy and management**—Panorama network security management enables you to manage your VM-Series firewall deployments across multiple cloud environments, along with your physical security appliances, thereby ensuring policy consistency and cohesiveness. Rich, centralized logging and reporting capabilities provide visibility into virtualized applications, users, and content.
- **Automation features that reduce time to deployment**—VM-Series firewalls include management features that enable you to integrate security into your public cloud development projects. You can use bootstrapping to automatically deploy firewalls. After bootstrapped firewalls deploy, Panorama instances can configure the firewall and keep the firewall policy up to date. Alternatively, you can use automation tools, such as Terraform and Ansible, to deploy and configure the VM-Series firewalls and the AWS resources. You can use firewall performance metrics and health information to create automated actions based on performance and usage patterns. By using the fully documented XML API and dynamic address groups, which allow VM-Series firewalls to consume external data in the form of tags, you can automate policy updates when workloads change. The result is that you can deploy new applications and next-generation security simultaneously in an automated manner.

CLOUD INFRASTRUCTURE PROTECTION

AWS provides basic infrastructure components and has a responsibility to ensure that each customer's workloads are appropriately isolated and ensure that the underlying infrastructure and physical environment are secure. However, the customer has the responsibility to securely configure the instances, operating systems, and any necessary applications, as well as maintain the integrity of the data each virtual machine processes and stores. This shared-responsibility model is often a point of confusion for consumers of cloud services.

Services have default configurations that might be secure upon implementation, but to ensure the integrity of the data itself, it is up to the customer to make the assessment and lock those service configurations down.

Security and compliance risks in cloud computing threaten an organization's ability to drive digital business. The dynamic nature of the cloud, the potential complexity of having multiple cloud service providers in the environment, and the massive volume of cloud workloads makes security and compliance cumbersome.

Public cloud environments use a decentralized administration framework that often suffers from a corresponding lack of any centralized visibility. Additionally, compliance within these environments is complex to manage. Incident response requires the ability to rapidly detect and respond to threats. However, public cloud capabilities are limited in these areas.

Prisma™ Cloud offers comprehensive and consistent cloud infrastructure protection that enables organizations to effectively transition to the public cloud by managing security and compliance risks within their public cloud infrastructure.

Prisma Cloud threat defense enables your organization to:

- Improve the visibility of assets and applications.
- Provide security and compliance posture reporting.
- Enforce DevOps best practices, implemented using policy guardrails.
- Implement DevOps threat monitoring, which identifies risky configurations, network intrusions, and host vulnerabilities for the management plane. This complements the capabilities of the VM-Series firewall to secure the in-line data plane.
- Perform anomaly detection to identify compromised accounts and insider threats.
- Gain forensic capabilities that permit the investigation of current threats or past incidents to quickly determine the root cause.
- Use contextual alerting in order to prioritize issues and respond appropriately.

By utilizing industry best practices for proactive security assessment and configuration management, Prisma Cloud makes cloud-computing assets harder to exploit. Prisma Cloud enables organizations to implement continuous monitoring of the AWS infrastructure and provides an essential, automated, up-to-date status of security posture that they can use to make cost-effective, risk-based decisions about service configuration and vulnerabilities inherent in cloud deployments.

Organizations can also use Prisma Cloud to prevent the AWS infrastructure from falling out of compliance. Visibility into the actual security posture of the cloud prevents failed audits and the subsequent fines associated with data breaches and non-compliance.

AWS Concepts and Services

When deployed on AWS, VM-Series firewalls rely upon underlying AWS services and functionality to integrate into the application traffic flow and protect the workload. The concepts covered in this section give an overview of AWS services relevant to VM-Series firewalls. AWS documentation is the definitive source of information on these topics and should be consulted for additional detail.

AWS TOP-LEVEL CONCEPTS

Public cloud architectures must consider where to locate computing resources, how to design for communicating effectively, and how to provide effective isolation such that an outage in one part of the cloud does not impact the entire cloud compute environment. The AWS global platform offers the ability to architect a cloud environment with scale, resilience, and flexibility.

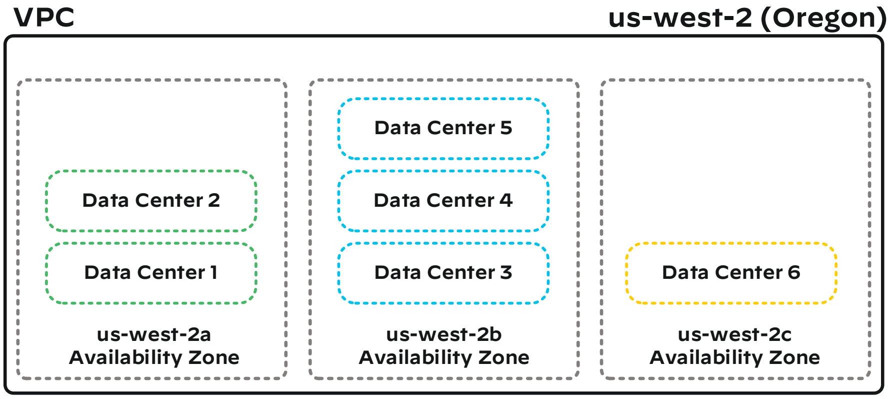
Regions

Regions enable you to place services and applications in proximity to your customers, as well as to meet government regulatory requirements for customer data residency. Regions represent AWS physical data center locations distributed around the globe. A region consists of several physically separate and co-located data center buildings, which provides maximum fault-tolerance and stability. The AWS backbone that provides redundant encrypted transport provides a path for communications between regions.

Availability Zones

Availability zones, or *zones*, provide a logical data center within a region. They consist of one or more physical data centers that interconnect with low-latency network links and have separate cooling and power. No two availability zones share a common facility. Availability zones provide inherent fault tolerance, and you distribute well-architected applications across multiple zones within a region.

Figure 2 Example of availability zones within a region



VIRTUAL PRIVATE CLOUD

An AWS *Virtual Private Cloud* (VPC) is a logically segmented network within AWS that allows connected resources to communicate with each other. VPCs are associated with a specific region and span that region's availability zones.

When deploying a new VPC, you specify a classless inter-domain routing (CIDR) IPv4 address block that you can then divide into subnets. VPC IP address blocks are reachable only within the VPC or through services connected to the VPC, such as a VPN. Because AWS isolates VPCs from each other, you can overlap IP address blocks across VPCs. IPv4 address blocks can be any valid IPv4 address range with a network prefix in the range of /16 (65,535 hosts) to /28 (16 hosts). The actual number of host addresses available to you on any subnet is less than the maximum because AWS reserves some addresses for services. You cannot change a VPC's original address block, but you can add secondary address blocks. It's recommended you choose a CIDR prefix that exceeds your anticipated address space requirements for the lifetime of the VPC. There are no costs associated with a VPC CIDR address block, and your VPC is only visible to you.

The primary considerations when choosing a VPC CIDR address block are the same as with any enterprise network:

- Anticipated number of IP addresses needed within a VPC
- IPv4 connectivity requirements across all VPCs
- IP address overlap in your entire organization—that is, between your AWS environment and your organization on-premises IP addressing or other IaaS clouds that you might use

Unlike enterprise networks that are mostly static and where network addressing changes can be difficult, cloud infrastructure tends to be highly dynamic, which minimizes the need to anticipate growth requirements far into the future. Instead of upgrading the resources in a VPC, many cloud deployments build new resources for an upgrade and then delete the old ones. Regardless of network address size, the general requirement for communications across the enterprise network is for all network addresses to be unique. The same requirement applies across your VPCs. When you deploy new VPCs, consider using a unique network address space for each to ensure maximum communications compatibility between VPCs and back to your organization.

Most VPC IP address ranges fall within the private IP address ranges specified in RFC 1918. However, you can use publicly routable CIDR address blocks for your VPC. Regardless of the IP address range of your VPC, AWS does not support direct access to the internet from your VPC's CIDR address block, including a publicly routable CIDR address block. You must set up internet access through a gateway service from AWS or a VPN connection to your organization's network.

Subnets

A *subnet* identifies a portion of its parent VPC CIDR address block as belonging to an availability zone. A subnet is unique to an availability zone and cannot span multiple zones. However, an availability zone can have many subnets. To associate resources to an availability zone, the zone must have a subnet. At the time of creating a resource, you assign it to an availability zone by associating it to a subnet within that zone. A subnet prefix length can be as large as the VPC CIDR address block (VPC with one subnet and one availability zone) or as small as a /28 prefix length. Subnets within a single VPC cannot overlap.

Subnets are either *public subnets*, which means they are associated with a route table that has internet connectivity via an internet gateway (IGW), or they are *private subnets* that have no route to the internet. Newly deployed subnets are associated with the main route table of your VPC. In Figure 3, subnets 1 and 2 are public subnets, and subnets 3 and 4 are private subnets.

Route Tables

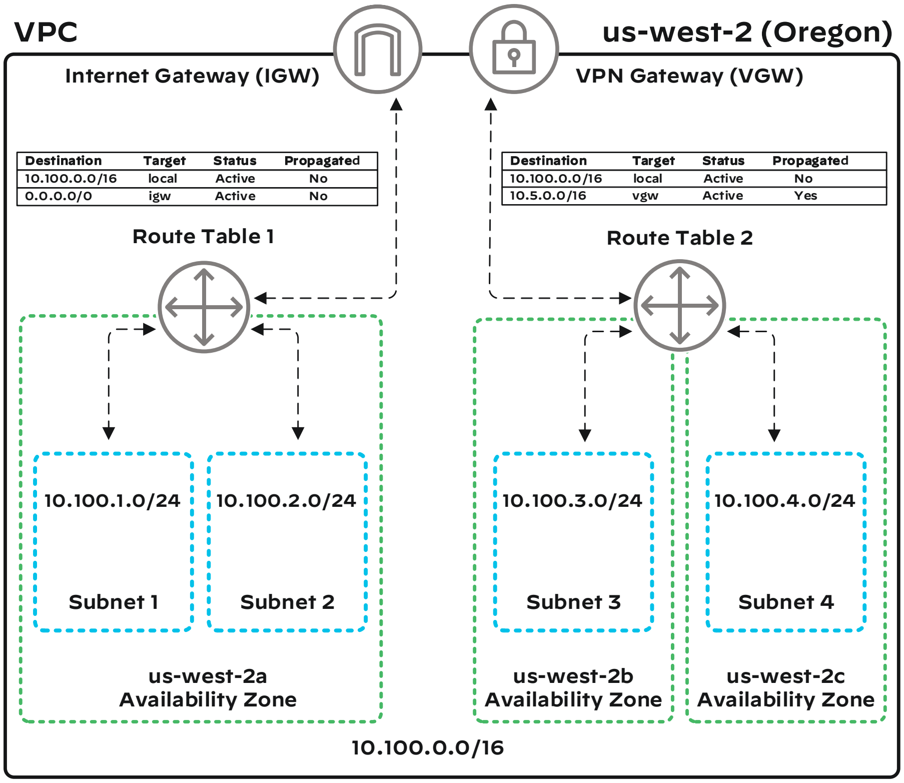
Route tables provide source-based control of Layer 3 forwarding within a VPC, which is different than traditional networking where routing information is bidirectional and might lead to asymmetric routing paths. Subnets are associated with route tables, and subnets receive their Layer 3 forwarding policy from their associated route table. A route table can have many subnets attached, but AWS only allows one route table to attach to a subnet. All route tables contain an entry for the entire VPC CIDR address block in which they reside. Any instance within the VPC has direct Layer 3 reachability to any other instance within the same VPC. This behavior has implications for network segmentation because route tables cannot contain more specific routes than the VPC CIDR address block. Any instance within a VPC can communicate directly with any other instance, and traditional network segmentation by subnets is not an option. An instance references the route table associated with its subnet for the default gateway but only for destinations outside the VPC. Host routing changes on instances are not necessary to direct traffic to a default gateway, because this is part of route table configuration. Routes external to your VPC can have a destination that directs traffic to a gateway or another instance.

Route tables can contain dynamic routing information learned from Border Gateway Protocol (BGP). Routes learned dynamically show in a route table as Propagated=YES.

A cursory review of route tables might give the impression of functionality similar to virtual routing and forwarding (VRF), but this is not the case. All route tables contain a route to the entire VPC address space and do not permit the segmentation of routing information less than the entire VPC CIDR address space within a VPC. Traditionally you must configure a device on a network with a default gateway in order to provide a path outside the local network. In AWS, route tables provide a similar function without the need to change instance configuration to redirect traffic.

You also use route tables to direct traffic to VM-Series firewalls. Note in Figure 3 that both route tables 1 and 2 contain the entire VPC CIDR address block entry. Route table 1 has a default route pointing to an internet gateway (IGW), and route table 2 has no default route. A route to 10.5.0.0/16 was learned via BGP, which is reachable via its virtual private gateway (VGW). Subnets 1 and 2 are assigned to availability zone 2a, subnet 3 is assigned to availability zone 2b, and subnet 4 is assigned to availability zone 2c.

Figure 3 Subnets and route tables



Destination	Target	Status	Propagated
10.100.0.0/16	local	Active	No
0.0.0.0/0	igw	Active	No

Destination	Target	Status	Propagated
10.100.0.0/16	local	Active	No
10.5.0.0/16	vgw	Active	Yes

There are limits to how many routes can be in a route table. The default limit of non-propagated routes in the table is 50, and you can increase to a limit of 1000. However, this might impact network performance. The limit to routes advertised by BGP into the VPC is 100, and you cannot increase this limit. Use IP address summarization upstream or a default route to address scenarios where more than 100 propagated routes might occur.

Network Access Control Lists

Because every route table contains a route to the entire VPC, you must use network access control lists (ACLs) to restrict traffic between subnets within your VPC. *Network ACLs* provide Layer 4 control of source/destination IP addresses and ports, inbound and outbound from subnets. When you deploy a VPC, there is a default network ACL associated with it, which permits all traffic. Network ACLs do not provide control of traffic to Amazon-reserved addresses (the first four addresses of a subnet) or of link-local networks (169.254.0.0/16), which AWS uses for VPN tunnels.

Network ACLs:

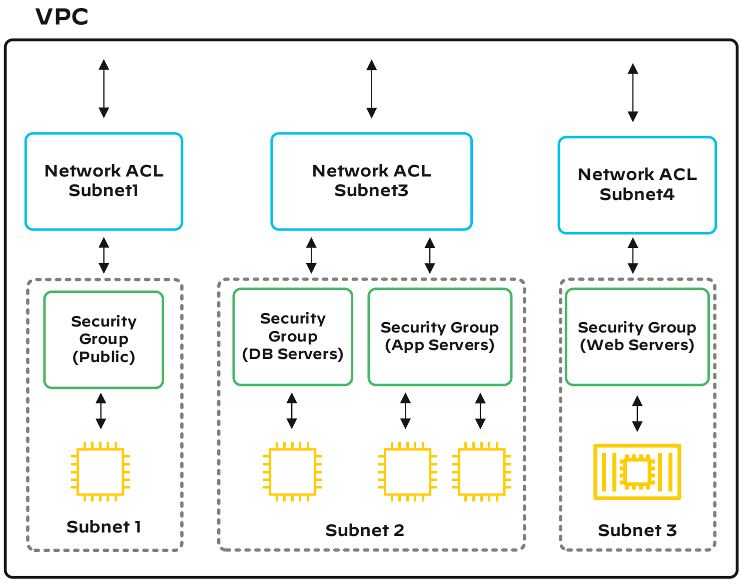
- Are applied at the subnet level.
- Have separate inbound and outbound policies.
- Have allow and deny action rules.
- Are stateless—bidirectional traffic must be permitted in both directions.
- Are order dependent—the first match rule (allow/deny) applies.
- Apply to all instances in the subnet.
- Do not filter traffic between instances within the same subnet.

Security Groups

Security groups (SGs) provide a Layer 4 stateful firewall for control of the source/destination IP addresses and ports that are permitted. You apply SGs to an instance's network interface. Up to five SGs can be associated with a network interface. Amazon Machine Images, which are available in the AWS Marketplace, have a default SG associated with them.

SGs define the network traffic that should be explicitly permitted and deny any traffic not explicitly permitted. They have separate rules for inbound and outbound traffic from an instance network interface. SGs are *stateful*, meaning that they also permit return traffic associated with permitted rules. SGs can control traffic on any protocol that has a standard protocol number. A VM-Series firewall in the traffic path is required to enforce traffic at the application layer. When you deploy a new SG, the default setting contains no inbound rule, and the outbound rule permits all traffic. The effect of this default is to permit all outbound network traffic originating from your instance and its associated return traffic and to deny external traffic that is inbound to an instance. Figure 4 illustrates how you can apply network ACLs to traffic between subnets and SGs to network interfaces.

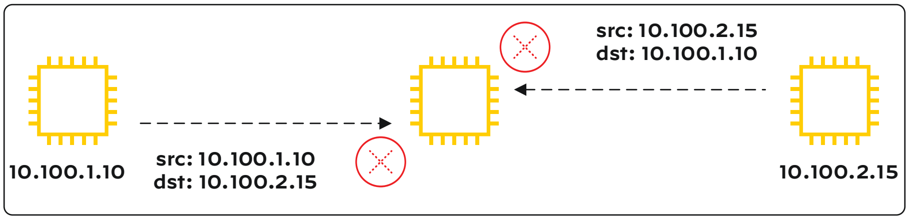
Figure 4 Security groups and network access control lists



Source and Destination Check

AWS enables source and destination checks by default on all network interfaces. The Source/Dest Check feature validates whether traffic is destined to, or originates from, an instance and prevents any traffic that does not meet this validation. A network device, like a firewall, that wishes to forward traffic between its network interfaces within a VPC must have the Source/Dest Check feature disabled on all interfaces that are capable of forwarding traffic.

Figure 5 Source and destination check



Source/Dest. Check – Enabled (default)

VIRTUAL COMPUTE

Virtual machines and related resources like storage are resources that you can deploy or terminate on demand. You configure a virtual machine's location by associating it to a region and availability zone. Most organizations deploy resources based upon an application's access and availability requirements.

Amazon Machine Image

Amazon Machine Images (AMIs) are virtual machine images available in the Amazon Marketplace. AWS publishes many AMIs that contain standard software configurations for public use. Also, members of the AWS developer community have published their own custom AMIs. You can also create your own custom AMIs. Doing so enables you to quickly and easily start new instances that have everything you need.

Instance

An *instance*, also known as *Elastic Compute Cloud* (EC2), represents a virtual machine. Much like their physical counterparts, instances have various performance characteristics, such as the number of CPUs, memory, and number of interfaces. You can change the instance type for instances that are in the stopped state. AWS bases the hourly operating costs on instance type and region.

Elastic Network Interface

Elastic network interfaces (ENIs) are virtual network interfaces that you attach to instances and appear as network interfaces on the instance. Every instance type has a maximum number of network interfaces. As an example, the c5.xlarge instance commonly used for a Palo Alto Networks VM-Series firewall supports 4 network interfaces.

Instance network interfaces receive IP addresses, default gateways, and DNS servers from the AWS DHCP service. By default, when you start an instance, DHCP dynamically assigns the first available IP address in the subnet to the instance's interface. Stopping an instance releases any IP addresses allocated to it. The AWS DHCP service does not retain the IP address reservation until the lease time expires. The next time you start an instance, DHCP again assigns the first available IP address in the subnet to the instance's interface. In environments where instances change state often, you should not expect consistent IP address assignment.

Static IP addressing is available when you require a consistent IP address. When there is only one IP address on an interface, you do not need to configure static IP addresses in the operating system running on the instance. Instead, you set up that static IP address in AWS. When you configure a static IP address, the instance still receives the IP address through DHCP. However, unlike dynamic IP address allocation, when started, the instance uses the configured IP address, and when stopped, the instance does not release the IP address. The next time the instance starts, the IP address remains the same.

An elastic network interface can include the following attributes:

- A primary private IPv4 address from the address range of your VPC
- One dynamic or elastic public IPv4 address per private IP address
- One or more IPv6 addresses
- One or more SGs
- Source/Dest Check

Within the same availability zone, you can detach and reattach ENIs to another instance up to the maximum number of interfaces supported by the instance type. The ENI characteristics are then associated with its newly attached instance.

Elastic IP Address

Elastic IP addresses are public IP addresses that belong to AWS or a customer-owned IP address pool. Public IP addresses are associated with the network interface of an instance. After they are associated, AWS configures network address translation in the VPC IGW that provides a 1:1 translation between the public IP address and the network interface's private IP address. When an instance is in the stopped state, the public IP address remains associated with the instance unless you intentionally move or delete it.

ACCESSING AWS VIRTUAL NETWORKS

Many use-cases require access to and from the AWS VPC. For example, instances in the VPC need to download their applications, patches, and data from existing customer data centers or vendor sites on the internet. Users might also need inbound access from the internet or need remote private network access to instances in AWS that provide application services. There are many access methods available with AWS; the below sections cover the most used.

Internet Gateway

An *internet gateway* (IGW) provides a mapping of an internal VPC IP address to a public IP address owned by AWS. The IGW maps an IP address to an instance for inbound and outbound network access. The public IP address can be:

- Random and dynamic, which means that AWS assigns the IP address to an instance at startup and returned to the pool when you stop the instance. Every time you start the instance, AWS assigns a new address from its pool.
- Random and assigned to an instance as part of a process, which means that the IP address stays with the instance unless you intentionally assign it to another instance or delete it and return it to the pool. This type of public IP address is known as an *Elastic IP address*.

This 1:1 private-to-public IP address mapping is part of a network interface configuration of each instance. After deploying a network interface, you can then associate a dynamic or an Elastic IP address to create the 1:1 IP address translation between public and VPC private IP addresses.

For internet connectivity to your VPC, the VPC must have an associated IGW. The IGW is a horizontally scaled, redundant, and highly available service. After it is associated, the IGW resides in all availability zones of your VPC, available to map to any route table or subnet where direct internet access is required.

Note

For an instance to have direct connectivity to the internet, it must have a public IP address associated with its interface, an IGW in the VPC, and a route for the internet traffic pointing to the IGW.

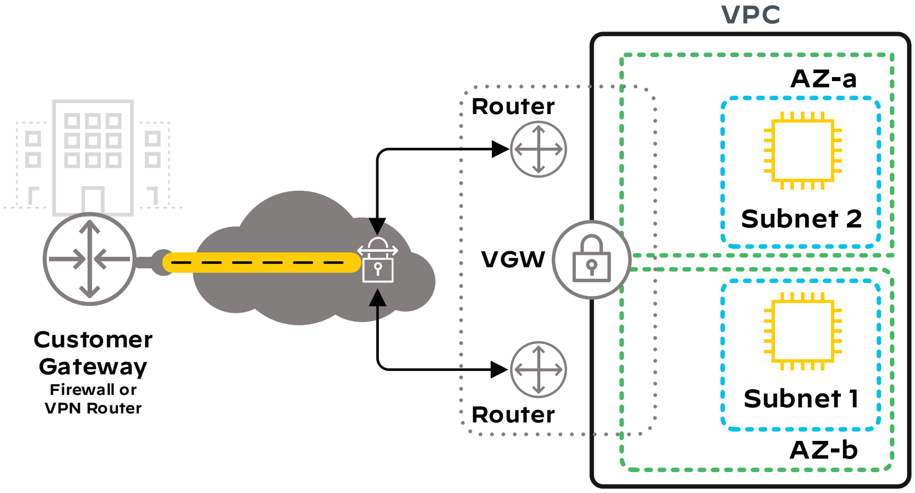
Virtual Private Gateway

A *virtual private gateway* (VGW) provides a VPN service to a VPC for the termination of IPSec tunnels. The tunnels provide confidentiality of traffic in transit and support peering to almost any device capable of supporting IPSec. Like with IGWs, the VGW resides in all availability zones of your VPC, available to map to any route table where VPN network access is required. You can map to the remote-site routes in the route table statically, or the VGW can learn them dynamically.

A *customer gateway* (CGW) identifies the target IP address of a peer device that terminates IPSec tunnels from the VGW. The customer gateway is typically a firewall or a router and must be capable of supporting an IPSec tunnel with required cryptographic algorithms.

VPN connections are the IPSec tunnels between your VGW and CGW. VPN connections represent two redundant IPSec tunnels from a single CGW to two public IP addresses of the VGW in your subscriber VPC.

Figure 6 VPN connections



AWS Direct Connect

AWS Direct Connect allows you to connect your network infrastructure directly to your AWS infrastructure by using private, dedicated bandwidth. You can connect from your data center or office via a dedicated link from a telecommunications provider, or you can connect directly in a colocation facility where AWS has a presence. This direct connection provides some advantages:

- Support for physical firewall hardware
- Higher-bandwidth network connections
- Lower-bandwidth costs
- Consistent network-transport performance
- Arbitrary inter-VPC traffic inspection and enforcement

AWS Direct Connect requires a network device to terminate the network connection from the AWS backbone network. This same device also terminates your carrier connection, completing the path between your private network and your AWS infrastructure. Your firewalls exchange BGP routing information with the AWS network infrastructure. Static routing is not available.

AWS Direct Connect Gateway

The AWS Direct Connect gateway complements AWS Direct Connect by allowing you to connect one or more of your VPCs to your on-premises network, whether those VPCs are in the same or different AWS regions.

The Direct Connect gateway:

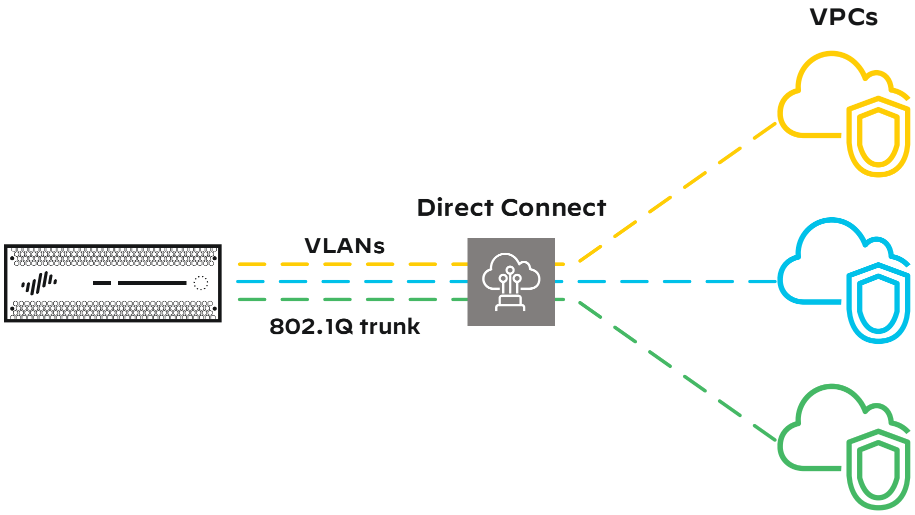
- Can be deployed in any public region.
- Can be accessed from most public regions.
- Is a globally available resource.
- Uses AWS Direct Connect for connectivity to your on-premises network.

The Direct Connect gateway connects on-premises networks to the VPCs. VPCs connected to the gateway cannot communicate directly with each other. You configure the on-premise gateway to create a BGP peer connection to the Direct Connect gateway and not to every VPC. This connection at the gateway eliminates the configuration and monitoring tasks that would otherwise be required to support BGP peering for each VPC.

As demand for bandwidth to your on-premises network grows, AWS Direct Connect provides the ability to extend your private VPC and publicly available AWS services directly to your data center infrastructure by using dedicated network bandwidth. You can place your network equipment directly in an AWS regional location, with a direct cross-connect to their backbone network, or you can use a network provider service, such as LAN extension or MPLS, to extend your AWS infrastructure to your network.

AWS extends the private virtual interfaces configured on the Direct Connect port over 802.1Q trunk links to your on-premises firewall, one VLAN for each VPC to which you are mapping, as shown in Figure 7.

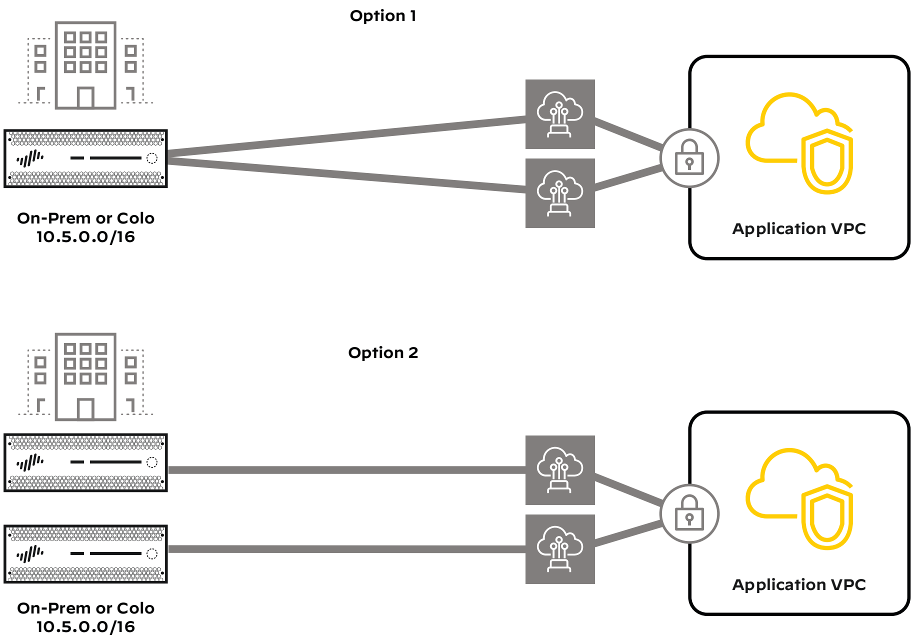
Figure 7 Direct Connect private virtual interfaces



Direct Connect provides many options for device and location redundancy. Figure 8 illustrates two of these options. The first option provides redundant connections from a single firewall to your VPC. When you require multiple instances of Direct Connect, AWS distributes them across redundant AWS backbone infrastructure. The BGP peering IP addresses must be unique across all virtual interfaces connected to a VGW. The VLAN is unique to each AWS Direct Connect port because it is only locally significant, so virtual interfaces can share a common VLAN.

The second option illustrates redundant firewalls distributed across two separate AWS locations servicing the same region. This option provides redundancy for devices, geographic location, and service providers.

Figure 8 Direct Connect redundancy options



CONNECTING VPCS

The use of multiple VPCs allows you to separate workloads across functional environments or administrative domains. You can connect VPCs through the use of VPC peering or an AWS transit gateway.

VPC Peering

VPC peering allows you to connect VPCs logically and is a native capability of AWS for creating direct two-way peer relationships between two VPCs within the same region. The peer relationship permits traffic only directly between the two peers and does not provide for any transit capabilities from one peer VPC through another to an external destination. VPC peering is a two-way agreement between member VPCs. It’s initiated by one VPC to another target VPC, and the target VPC must accept the VPC peering relationship. The VPCs in a peering relationship can be in the same AWS account or different accounts, and a VPC can be in multiple VPC peering relationships. After you establish the VPC peering relationship, there is two-way network connectivity between the entire IP address block of both VPCs.

VPC peering ensures that traffic traversing the peering connection has source and destination IP addresses of the directly peered VPCs. AWS drops any packets with a source or destination IP address outside of the two peered VPCs.

VPC peering architecture uses network policy to permit traffic only between two directly adjacent VPCs. The following are two example scenarios:

- **Hub-and-spoke model**—In a hub-and-spoke model, subscriber VPCs (spokes) use VPC peering with the central VPC (hub) to provide direct communications between the instances in the subscriber VPCs and the instances in the central VPC. The subscriber VPCs are unable to communicate with each other because this would require transit connectivity through the central VPC, which is not a capability supported by VPC peering. You can configure additional direct VPC peering relationships to permit communication between subscriber VPCs as required. Figure 9 illustrates how a hub and spoke model of VPC peering connections could operate.
- **Multi-tiered application model**—For multi-tiered applications, you can use VPC peering to restrict communication to only directly adjacent application tiers. A typical three-tier application might use VPC peering to connect frontend web servers in a public-facing VPC to a VPC containing the application tier. The application-tier VPC would have another VPC peering relationship with a third VPC containing the database tier. VPC peering provides no external connectivity directly to the application or database tiers. Figure 10 illustrates how a central VPC with multi-tier VPC connections could operate.

Figure 9 Hub-and-spoke model for VPC peering

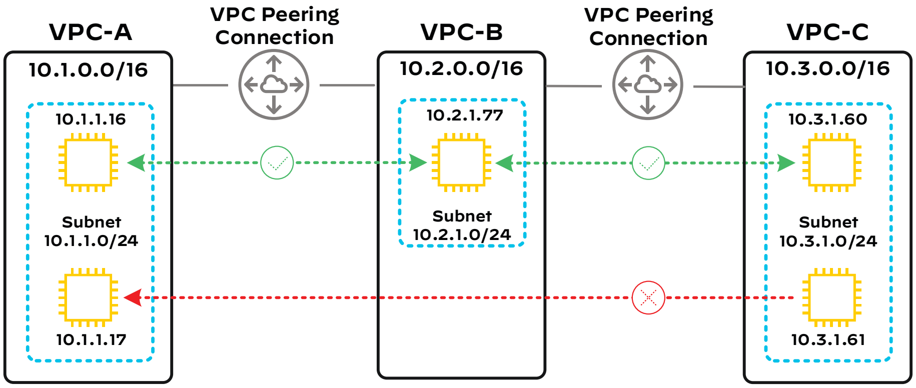
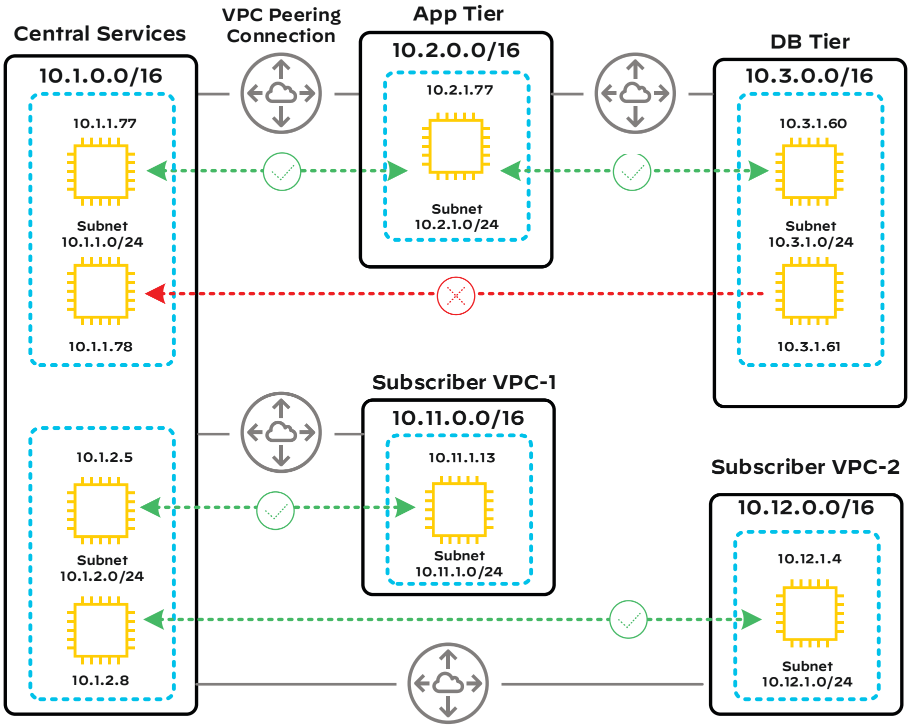


Figure 10 Multi-tiered application model for VPC peering



Transit Gateway

The AWS transit gateway (TGW) service enables you to scale connectivity across thousands of VPCs, AWS accounts, and on-premises networks. TGW enables you to control communications between your VPCs and to connect to your on-premises networks via a single gateway. In contrast to VPC peering, which interconnects two VPCs only, TGWs can act as a hub in a hub-and-spoke model for interconnecting VPCs. The spokes peer only to the gateway, which simplifies design and management overhead. You can add new spokes to the gateway incrementally as your deployment grows.

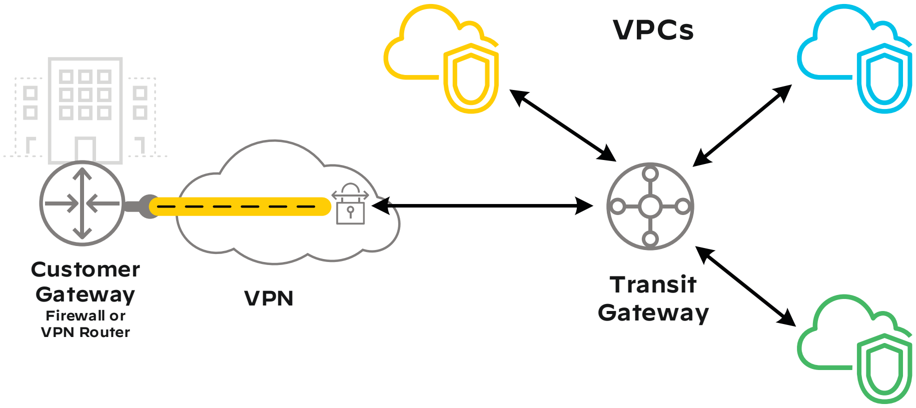
TGWs provide the ability to centrally manage the connectivity and routing between VPCs and from the VPCs to any on-premises connections via VPN or Direct Connect. TGWs allow central control of spoke-to-spoke communication and routing. TGWs support dynamic and static Layer 3 routing between VPCs and VPNs. Routes determine the next hop depending on the destination IP address of the packet, and they can point to a VPC or a VPN connection.

TGWs support the attachment of thousands of VPCs. There are two types of attachments, VPC attachments and VPN attachments.

VPC attachments are attachments from a VPC to the TGW. As part of the VPC attachment, the TGW deploys an ENI interface in each of the VPCs in-use availability zones. The TGW deploys the ENIs to subnets that you create in the attached VPCs, one subnet per availability zone. VPC attachments support only static routing and do not support Equal Cost Multipath (ECMP).

VPN attachments are either VPCs attached via VPN or VPN attached remote sites. A VPN attachment consists of two IPSec VPN tunnels per attachment to a CGW. VPN attachments have the advantage of supporting dynamic routing and ECMP.

Figure 11 AWS transit gateway



You can configure connections between a TGW and on-premises gateways by using a VPN or Direct Connect. Because a TGW supports ECMP, you can increase bandwidth to on-premises networks by:

- Deploying multiple VPN connections between the TGW and the on-premises firewall.
- Using BGP to announce the same prefixes over each path.
- Enabling ECMP on both ends of the connections to load-balance traffic over the multiple paths.

You can use TGWs to direct inbound, outbound, and east-west flows through centralized firewalls.

RESILIENCY CONSTRUCTS

Traditional data centers provide network resilience through alternate transport paths and high-availability platforms like switches, routers, and firewalls. The high-availability platforms either had redundancy mechanisms within the chassis or between chassis, which often introduced cost and complexity in the design. As networks scaled to meet higher throughput demands of modern data centers, more chassis had to be included in the high-availability group to provide more throughput, further complicating designs and increasing costs. The move to web-based, multi-tier applications allows network designs to scale more gracefully and provide resiliency to the overall application through a stateless scale out of resources.

Availability Zones

As discussed earlier in this guide, AWS provides resilience within regions by grouping data center facilities into availability zones. For resilience, applications and services you deploy in AWS should reside in at least two availability zones. AWS services like IGW and VGW are resilient in multiple zones.

Load Balancers

Load balancers distribute traffic inbound to an application across a set of resources based on IP traffic criteria such as the DNS, Layer 4, or Layer 7 information. Load balancers check the traffic targets for health and remove unhealthy resources, thereby enhancing resiliency of the application. *Targets* can be instances, containers, and VPC IP addresses.

AWS offers several types of load balancers. AWS recommends using the Application Load Balancer for Layer 7 and Network Load Balancer for Layer 4. Like other AWS services, the load balancers exist in multiple availability zones for resiliency and scale horizontally for growth.

Application Load Balancer

The *Application Load Balancer* (ALB) uses HTTP and HTTPS information to distribute traffic to targets.

Traffic destined for the ALB uses DNS names and not a discrete IP address. AWS assigns a fully qualified domain name (FQDN) when it deploys the load balancer. The FQDN maps to IP addresses in each availability zone. If the ALB in one zone fails or has no healthy targets and if it is tied to an external DNS like the Amazon Route 53 cloud-based DNS, then traffic is directed to an alternate ALB in the other zone.

You can configure an ALB with its frontend facing the internet and load balance by using public IP addressing, or you can configure it to be internal only and load balance by using IP addresses from the VPC. The ALB targets can be any HTTP or HTTPS applications that are reachable from within the VPC.

The ALB offers content-based routing of connection requests based on either the host field or the URL of the HTTP header of the client request. ALB uses a round-robin algorithm to distribute traffic and supports a slow start when adding targets to the pool in order to avoid overwhelming the application target. The ALB determines which targets are healthy based on IP probes and HTTP error codes.

The ALB supports terminating HTTPS between the client and the load balancer, and it can manage SSL certificates.

Network Load Balancer

The *Network Load Balancer* (NLB) uses TCP/UDP information to distribute traffic to targets.

On a single, static IP address per availability zone, the NLB accepts incoming traffic from clients and distributes the traffic to targets within the same availability zone. Monitoring the health of the targets, NLB ensures that only healthy targets get traffic. If all targets in an availability zone are unhealthy and if you have set up targets in another zone, then the NLB automatically fails-over to the healthy targets in the other availability zones.

The NLB supports a static IP address assignment, including an Elastic IP address, for the frontend of the load balancer, making it ideal for services that do not use DNS and rely on the IP address to route connections. If the NLB has no healthy targets and if it is tied to Amazon Route 53 cloud-based DNS, then traffic is directed to an alternate NLB in another region. You can load balance to any IP address target that is reachable from within the VPC. This allows the NLB to load balance to any IP address and any interface on an instance.

Palo Alto Networks Design Details

VM-SERIES FIREWALL ON AWS

The Palo Alto Networks VM-Series firewall is the virtual form factor of a next-generation firewall. It can be deployed in a range of private and public cloud computing environments. The VM-Series firewall on AWS enables you to securely implement a cloud-first methodology while transforming your data center into a hybrid architecture that combines the scalability and agility of AWS with your on-premises resources. This allows you to move your applications and data to AWS while maintaining a security posture that is consistent with the one you might have established on your physical network. The VM-Series firewall on AWS natively analyzes all traffic in a single pass to determine application, content, and user identity. The application, content, and user are core elements of your security policy and for visibility, reporting, and incident investigation.

VM-Series Firewall Models

VM-Series firewalls on AWS are available in four primary models: VM-100, VM-300, VM-500, VM-700. Varying only by capacity, all of the firewall models use the same image. A *capacity license* configures the firewall with a model number and associated capacity limits.

Table 1 VM-Series firewall capacities and system requirements

	VM-100	VM-300	VM-500	VM-700
Capacities				
AWS instance size tested (recommended)	m5.xlarge	m5.xlarge	m5.2xlarge	m5.4xlarge
Firewall throughput (App-ID™ enabled)	2.7Gbps	4.2Gbps	7.8Gbps	7.9Gbps
Threat Prevention throughput	1.4Gbps	2.4Gbps	5.3Gbps	7.0Gbps
IPSec VPN throughput	0.9Gbps	1.2Gbps	2.2Gbps	4.4Gbps

Although the capacity license sets the VM-Series firewalls limits, the size of the instance on which you deploy the firewall determines the firewall's performance and functional capacity. In Table 1, the mapping of the VM-Series firewall to AWS instance size is based on VM-Series model requirements for CPU, memory, disk capacity, and network interfaces. When deployed on an instance that provides more CPU than the model supports, VM-Series firewalls, using BYOL, do not use the additional CPU cores. Conversely, when you deploy a large VM-Series model, using BYOL, on an instance that meets the minimum CPU requirements, it effectively performs the same as a lower model VM-Series firewall.

In smaller VM-Series firewall models, it might seem that an instance size smaller than those listed in Table 1 would be appropriate; however, smaller instance sizes do not have enough network interfaces. AWS provides instances with two, three, four, eight, or fifteen network interfaces. Because VM-Series firewalls reserve an interface for management functionality, two-interface instances are not a viable option. Four-interface instances meet the minimum requirement of a management, public, and private interface. You can configure the fourth interface as a security interface for optional services such as VPN or a demilitarized zone (DMZ).

Although larger models of VM-Series firewalls offer increased capacities, on AWS, some throughput is limited, and a larger number of cores helps with scale more than throughput. For the latest detailed information, see the VM-Series for Amazon Web Services document. Many factors affect performance, and Palo Alto Networks recommends you do additional testing in your environment to ensure the deployment meets your performance and capacity requirements. In general, public cloud environments are more efficient when scaling out the number of resources versus scaling up to a larger instance size.

License Options

You purchase licenses for VM-Series firewalls on AWS through the AWS Marketplace or through traditional Palo Alto Networks channels.

Note

Whichever licensing model you chose is permanent. After you deploy them, VM-Series firewalls cannot switch between the pay-as-you-go (PAYG) and bring-your-own-license (BYOL) licensing models. Switching between licensing models requires deploying a new firewall and migrating the configuration. In the BYOL model, you can migrate between licensing agreements, including evaluation, regular, and enterprise, because they are all part of the same licensing model.

PAYG

A *pay-as-you-go* (PAYG) license model is also called a *usage-based* or *pay-per-use* license. You can purchase this type of license from the AWS Marketplace, and you are billed hourly.

With the PAYG license, a VM-Series firewall is licensed and ready for use as soon as you deploy it. You do not receive a license authorization code. When the firewall is stopped or terminated in AWS, the usage-based licenses are suspended or terminated.

PAYG licenses support the VM-100, VM-300, VM-500, VM-700 capacity licenses. A PAYG license applies a VM-Series capacity license based on the hardware allocated to the instance. The PAYG instance checks the amount of hardware resources available to the instance and applies the largest VM-Series firewall capacity license allowed for the resources available. For example, if the instance has 2 vCPUs and 16GB of memory, a VM-100 capacity license is applied based on the number of vCPUs. However, if the instance has 16 vCPUs and 16GB of memory, a VM-500 license is applied based on the amount of memory.

PAYG licenses are available in the following bundles:

- **Bundle 1**—Includes the VM-Series capacity license, Threat Prevention license (IPS, AV, malware prevention), and a premium support entitlement
- **Bundle 2**—Includes the VM-Series capacity license, Threat Prevention license (IPS, AV, malware prevention), DNS Security license, GlobalProtect™ license, WildFire license, PAN-DB URL Filtering license, and a premium support entitlement

BYOL and VM-Series ELA

A *bring-your-own license* (BYOL) model allows you to purchase a license from a partner, reseller, or directly from Palo Alto Networks. In a BYOL model, VM-Series firewalls support all capacities, support entitlements, and subscription licenses.

When using your own licenses, you license VM-Series firewalls like a traditionally deployed appliance, and you must apply a license authorization code. After you apply the code to the device, the device registers with the Palo Alto Networks support portal and obtains information about its capacity and subscriptions. Subscription licenses include Threat Prevention, PAN-DB URL Filtering, AutoFocus™, GlobalProtect, and WildFire.

To accelerate firewall deployment, the VM-Series enterprise licensing agreement (ELA) provides a fixed-price licensing option that allows unlimited deployment of VM-Series firewalls with BYOL. Palo Alto Networks offers licenses in one- and three-year term agreements with no true-up at the end of the term.

The VM-Series ELA includes four components:

- A license token pool that allows you to deploy any model of the VM-Series firewall. Depending on the firewall model and the number of firewalls that you deploy, you deduct a specified number of tokens from your available license token pool. All of your VM-Series ELA deployments use a single license authorization code, which allows for easier automation and simplifies the deployment of firewalls.
- Threat Prevention, WildFire, GlobalProtect, DNS Security, and PAN-DB subscriptions for every VM-Series firewall deployed as part of the VM-Series ELA.
- Unlimited deployments of Panorama as a virtual appliance.
- Support that covers all the components deployed as part of the VM-Series ELA.

VM-SERIES FIREWALL INTEGRATION TO AWS

Launching a VM-Series Firewall on AWS

The Amazon AWS Marketplace provides a wide variety of Linux, Windows, and specialized machine images, like a Palo Alto Networks VM-Series firewall. There, you can find AMIs for Palo Alto Networks VM-Series firewall with various licensing options. After you select one, the AMI launch instance workflow provides a step-by-step guided workflow for all IP addressing, network settings, and storage requirements. You can provide your own custom AMIs to suit your design needs. Automation scripts for building out large-scale environments usually include AMI programming.

Bootstrapping

At deployment, VM-Series firewalls have the factory default configuration and a base software image that varies based on which deployment method you have chosen. You can manually upgrade the software and update the configuration after deploying the instance, or if you are using a BYOL licensing model, you can use bootstrapping to license, configure, and update the firewall software at boot time.

Bootstrapping allows you to create a repeatable process of deploying VM-Series firewalls through a bootstrap package. The package can contain everything required to make the firewall ready for production or just enough information to get the firewall operational and connected to Panorama. In AWS, you implement the bootstrap package through an AWS S3 file share that contains directories for configuration, content, license, and software. On the first boot, VM-Series firewalls mount the file share and use the information in the directories to configure and upgrade the firewall. After the firewall is out of the factory default state, it stops looking for a bootstrap package.

One of the fundamental design differences between traditional and public-cloud deployments is the lifetime of resources. One method of achieving resiliency in public cloud deployments is through the quick deployment of new resources and the quick destruction of failed resources. One of the requirements for achieving quick resource build-out and tear-down is current and readily available configuration information for the resource to use during initial deployment. When the configuration is static, the simplest method of achieving this for VM-Series firewalls is to use bootstrapping to configure the firewall policies during firewall deployment.

Management

The first interface attached to the instance (eth0) is the firewall's management interface. In most templates, this interface has an Elastic IP address and DNS hostname associated with it in addition to the internal IP address in the VPC. The firewall's management interface obtains its internal IP address through DHCP. The IGW translates the internal IP address to the public IP address when the traffic leaves the VPC. Because IP addresses might change, use an FQDN to manage the firewall.

Note

If you assign a dynamic public IP address on the instance eth0 interface and later assign an Elastic IP address to any interface on the same instance, upon reboot, AWS also replaces the dynamic IP address on eth0 with an Elastic IP address. The best way to predictably assign a public IP address to your eth0 management interface is to associate an Elastic IP address to the eth0 interface.

If you are managing the firewalls with Panorama, you might not need an Elastic IP address on the firewall's management interface. If Panorama is in the AWS environment or deployed on-premises with VPN connectivity to the VPC, Panorama can use internal IP addresses to manage the VM-Series firewalls.

Managing AWS Deployments with Panorama

The best method for ensuring up-to-date firewall configuration is to use Panorama for the central management of firewall policies. Panorama simplifies consistent policy configuration across multiple independent firewalls through its device group and template stack capabilities. When multiple firewalls are part of the same device group, they receive a common ruleset. Because Panorama enables you to control all of your firewalls—whether they are on-premises or in the public cloud or whether they are a physical or virtual appliance—device groups also provide configuration hierarchy. With device group hierarchy, lower-level groups include the policies of the higher-level groups. Configuration hierarchy allows you to configure consistent rulesets that apply to all firewalls, as well as consistent rulesets that apply to specific firewall deployment locations such as the public cloud.

As bootstrapped firewalls deploy, they can also automatically pull configuration information from Panorama. VM-Series firewalls use a VM authorization key and Panorama IP address in the bootstrap package to authenticate and register to Panorama on its initial boot. You must generate the VM authorization key in Panorama before creating the bootstrap package. If you provide a device group and template in the bootstrap package's basic configuration file, Panorama assigns the firewall to the appropriate device group and template so that the relevant rulesets are applied, and you can manage the device in Panorama going forward.

You can deploy Panorama in your on-premises data center or in a public cloud environment such as AWS. When deployed in your on-premises data center, Panorama can manage all the PA-Series and VM-Series firewalls in your organization. If you want a dedicated instance of Panorama for the VM-Series firewalls deployed on AWS, deploy Panorama on AWS.

When you have an existing Panorama deployment on-premises for firewalls in your data center and internet edge, you can use it to manage the VM-Series firewalls in AWS. Beyond management, you need to consider your firewall log collection and retention. Log collection, storage, and analysis is an important cybersecurity best practice that organizations perform to correlate potential threats and prevent successful cyber breaches.

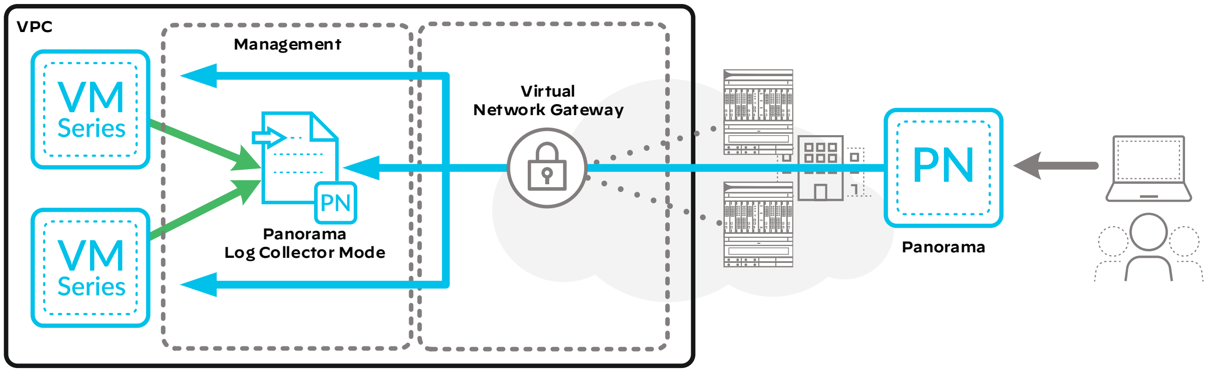
The following three deployment mode options are available for Panorama, which, if necessary, allows for the separation of management and log collection:

- **Log Collector mode**—One or more log collectors collect and manage logs from the managed devices. This assumes that another deployment of Panorama is operating in Management-Only mode.
- **Management-Only mode**—Panorama manages configurations for the managed devices but does not collect or manage logs.
- **Panorama mode**—Panorama controls both policy and log management functions for all the managed devices.

On-Premises Panorama with Dedicated Log Collectors in the Cloud

Sending logging data back to the on-premises Panorama can be inefficient, costly, and pose data privacy and residency issues in some regions. An alternative to sending the logging data black to your on-premises Panorama is to deploy Panorama dedicated log collectors on AWS and use the on-premises Panorama for management. Deploying a dedicated log collector on AWS reduces the amount of logging data that leaves the cloud but still allows your on-premises Panorama to manage the VM-Series firewalls in AWS and have full visibility to the logs as needed.

Figure 12 Panorama Log Collector mode in AWS

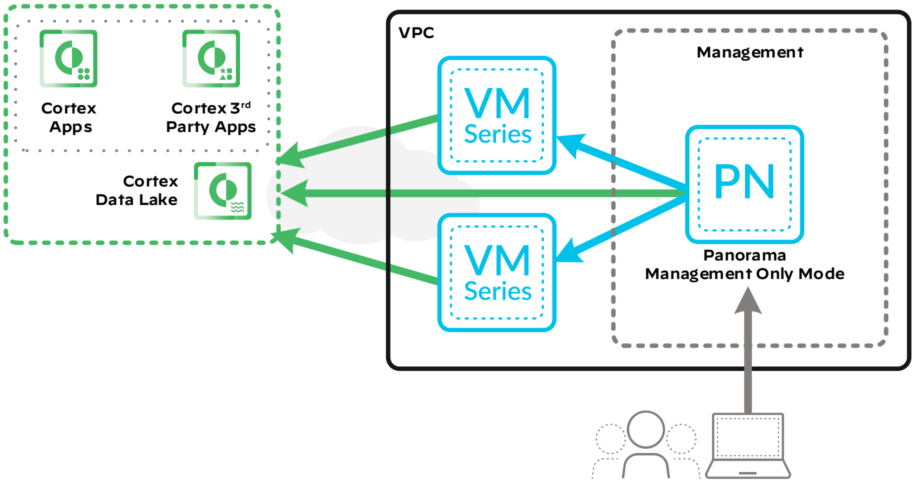


Panorama Management in AWS with Cortex Data Lake

There are two design options when deploying Panorama management on AWS. First, you can use Panorama for management only and use Palo Alto Networks Cortex Data Lake to store the logs generated by the VM-Series firewalls. *Cortex Data Lake* is a cloud-based log collector service that provides resilient storage and fast search capabilities for large amounts of logging data. Cortex Data Lake emulates a traditional log collector. The VM-Series firewalls encrypt the logs and then send them to the Cortex Data Lake over TLS/SSL connections. Cortex Data Lake allows you to scale your logging storage as your AWS deployment scales because Cortex bases licensing on storage capacity and not the number of devices sending log data.

The benefit of using Cortex Data Lake goes well beyond scale and convenience when tied into the Palo Alto Networks Cortex AI-based continuous security platform. Cortex is a scalable ecosystem of security applications that can apply advanced analytics in concert with Palo Alto Networks enforcement points to prevent the most advanced attacks. Palo Alto Networks analytics applications such as Cortex XDR and AutoFocus, as well as third-party analytics applications that you choose, use Cortex Data Lake as the primary data repository for all of Palo Alto Networks offerings.

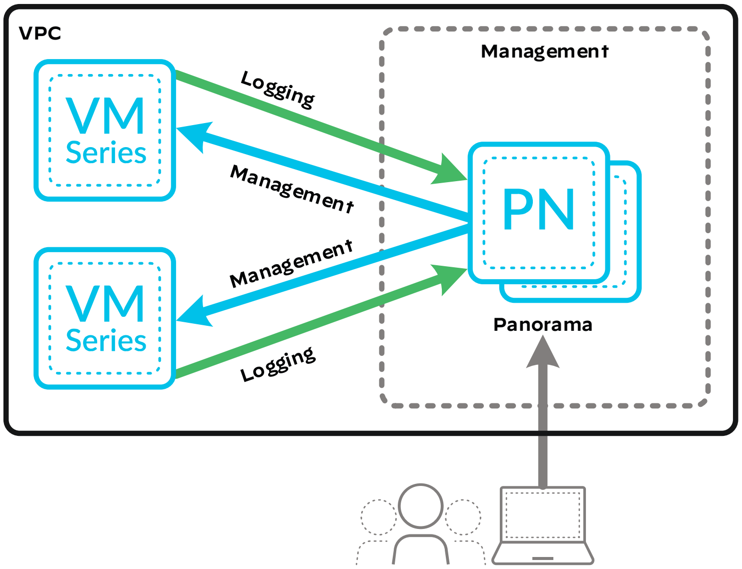
Figure 13 Panorama management and Cortex Data Lake



Panorama Management and Log Collection in AWS

Second, you can use Panorama for both management and log collection. You can deploy the management and log collection functionality as a shared virtual appliance or on dedicated virtual appliances. For smaller deployments, you can deploy Panorama and the log collector as a single virtual appliance. For larger deployments, a dedicated log collector per region allows traffic to stay within the region and reduce outbound data transfers.

Figure 14 Panorama management and log collection in AWS



Panorama is available as a virtual appliance for deployment on AWS and supports Management-Only mode, Panorama mode, and Log Collector mode with the system requirements defined in Table 2. Panorama on AWS is only available with a BYOL licensing model.

Table 2 Panorama Virtual Appliance on AWS

	Management-Only	Panorama	Log Collector
Minimum system requirements	16 CPUs 32GB memory 81GB system disk	16 CPUs 32GB memory 2TB to 24TB log storage capacity	16 CPUs 32GB memory 2TB to 24TB log storage capacity

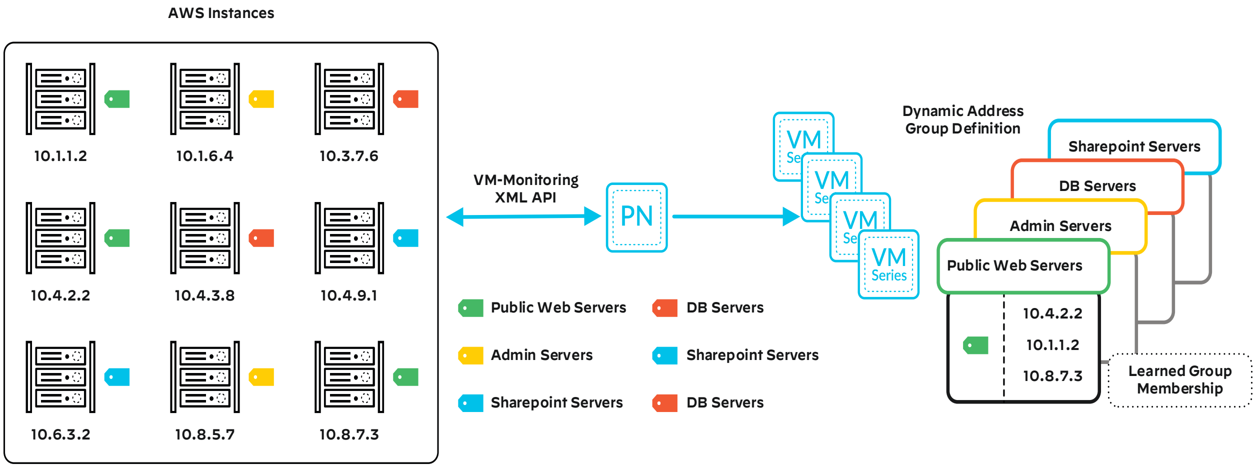
Security Policy Automation with VM Monitoring

Organizations typically build public cloud application environments around an agile application development process. In the agile environment, you deploy applications quickly and build new environments to accommodate a revised application versus trying to upgrade the existing operational environment. When the new environment goes online, you remove the now-unused, older application environment. This amount of change presents a challenge to enforcing security policy unless your security environment is compatible with an agile development environment.

Palo Alto Networks firewalls, including the VM-Series, support dynamic address groups. Dynamic address groups allow you to create policy that automatically adapts to instance additions, moves, or deletions. They also enable the flexibility to apply a security policy to the device based on its role.

A dynamic address group uses tags as a filtering criterion in order to determine its members. You can define tags statically or register them dynamically. You can dynamically register the IP address and associated tags for AWS instances by using VM monitoring on the VM-Series firewalls or the Panorama plugin for AWS.

Figure 15 VM monitoring of AWS tag to dynamic address group mappings



If you enable VM monitoring on the firewall, you can poll up to 10 VPCs. However, each firewall polls AWS independently, limiting the flexibility and scale. The Panorama plugin for AWS allows you to monitor up to 100 VPCs on AWS and over 6000 hosts. Panorama relays the dynamic address group mapping to the VM-Series firewalls, providing scale and flexibility.

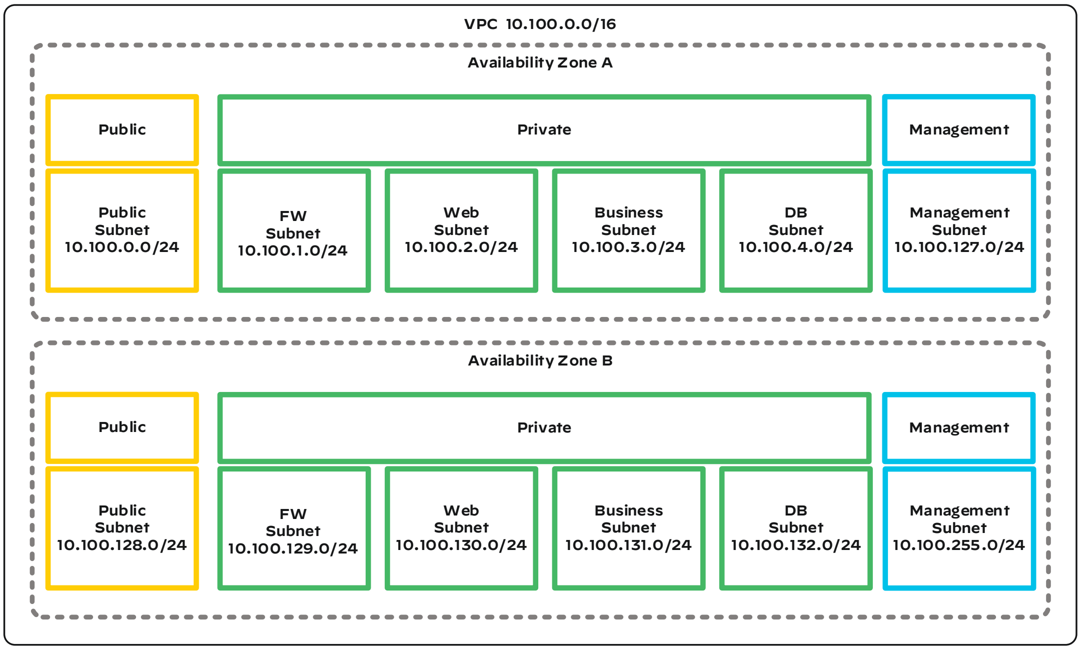
NETWORKING IN THE VPC

When deploying a VPC, configure it with an IP address block that is appropriately sized for your organization's needs. Avoid IP address overlap within AWS and with the rest of your organization. IP address overlap prevents you from peering VPCs and might require address translation at the borders.

Consider breaking your subnet ranges into functional blocks that you can summarize in ACLs and routing tables. Consider using three ranges: one each for the management network, public network, and private network.

For resiliency, use multiple availability zones. Duplicate the subnets in the additional availability zones, but configure unique IP address ranges from your VPC address block to each subnet. By default, all resources attached to the VPC, regardless of their availability zone or subnet, can communicate directly.

Figure 16 VPC IP addressing


VPC 10.100.0.0/16
Availability Zone A
Public
Private
Management
Public Subnet 10.100.0.0/24
FW Subnet 10.100.1.0/24
Web Subnet 10.100.2.0/24
Business Subnet 10.100.3.0/24
DB Subnet 10.100.4.0/24
Management Subnet 10.100.127.0/24
Availability Zone B
Public
Private
Management
Public Subnet 10.100.128.0/24
FW Subnet 10.100.129.0/24
Web Subnet 10.100.130.0/24
Business Subnet 10.100.131.0/24
DB Subnet 10.100.132.0/24
Management Subnet 10.100.255.0/24

Firewall Interfaces

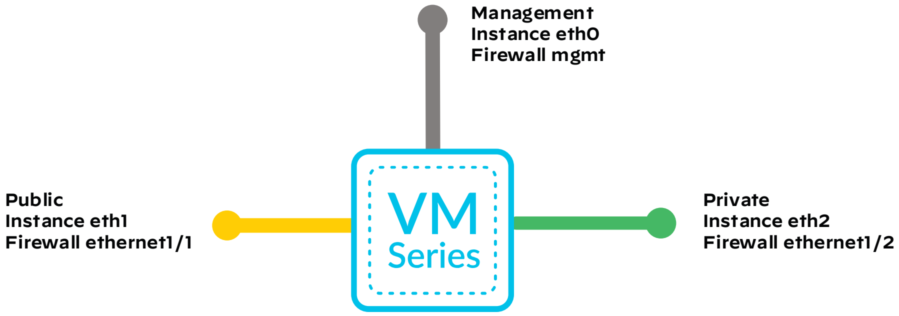
Although the VM-Series firewall supports multiple interface deployment configurations such as virtual wire, Layer 2, and tap mode, VM-Series firewall interfaces on AWS are always Layer 3 interfaces because of AWS networking requirements.

In a Layer 3 deployment, you must assign each interface an IP address. In AWS, you should always configure VM-Series firewall interfaces to obtain their IP address through DHCP. Even though the firewall receives its interface IP address assignments through DHCP, you should statically assign the addresses for routing next-hop assignments and other firewall functions. You can configure static IP addresses in the AWS console.

The VM-Series firewall requires three Ethernet interfaces: a management interface, a public-facing interface, and a compute or private-facing interface. When you deploy a VM-Series instance from the AWS Marketplace, by default, it has a single interface. You have to create the additional interfaces and associate them to the VM-Series instance. You also need to create two Elastic IP addresses. Assign one to the management interface so you can manage the firewall. Assign the second to the public-facing interface for inbound internet traffic and outbound address translation.

In order to provide inbound and outbound internet access, you must also deploy an IGW to the VPC. The IGW performs network address translation of the instance's private IP address to their associated public IP address.

Figure 17 Firewall interfaces



Caution

If you assign a dynamic public IP address to the management interface when deploying the instance, any Elastic IP assigned to the instance later results in AWS returning the dynamic address to the pool. The management interface will not have a public IP address assigned anymore. You must create another Elastic IP address and assign it to the management interface.

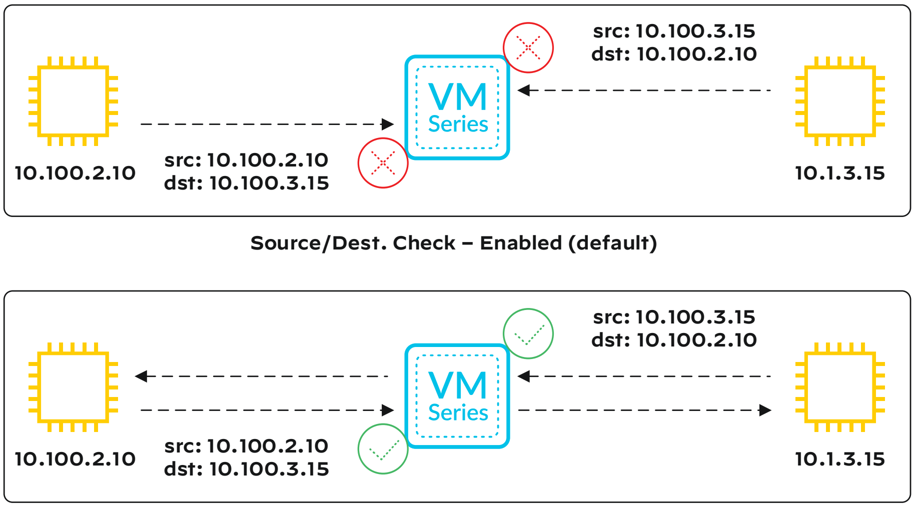
Source and Destination Check

AWS enables source and destination checks by default on all network interfaces within your VPC. The check validates whether traffic is destined to or originates from an instance, and it prevents any traffic that does not meet this validation. An instance that wishes to forward traffic between its network interfaces must have this check disabled on all forwarding interfaces. You must disable source and destination checking on all of the VM-Series firewall dataplane interfaces.

Note

You can change the SGs and Source/Dest Check feature at the instance level; however, these changes apply only to the first interface of the instance. For a VM-Series firewall, the first interface represents the management interface. To avoid ambiguity, you should apply SGs and disable the Source/Dest Check feature on the individual network interfaces (management, public, and private).

Figure 18 Source and destination check



Source/Dest. Check – Enabled (default)

Source/Dest. Check – Disabled

Route Tables

Route tables allow you to determine what external resources an instance can reach based on what routes or services you configure within that route table. AWS applies route tables to subnets. Consider deploying separate route tables for management, public, and private subnets as follows:

- The management route table has the management subnets assigned to it and a default route to the IGW for internet access.
- The public route table has the public subnets assigned to it and a default route to the IGW for internet access.
- Each availability zone has a private route table. Each route table has the private subnets in their availability zone assigned to it. They do not contain a route to the IGW. After you deploy the firewalls, you configure a default route pointing to the ENI of the VM-Series firewall in their availability zone.

Route tables do not affect the ability of instances in the same VPC to communicate.

Figure 19 Example route tables

Management	Destination	Target	Subnets Assigned
	10.100.0.0/16	Local	10.100.127.0 10.100.255.0
	0.0.0.0/0	Igw-991233991ab	

Public	Destination	Target	Subnets Assigned
	10.100.0.0/16	Local	10.100.0.0 10.100.128.0
	0.0.0.0/0	Igw-991233991ab	

Private AZ-a	Destination	Target	Subnets Assigned
	10.100.0.0/16	Local	10.100.2.0
	0.0.0.0/0	Eni-1182230012a	

Private AZ-b	Destination	Target	Subnets Assigned
	10.100.0.0/16	Local	10.100.130.0
	0.0.0.0/0	Eni-1182230012b	

AWS TRAFFIC FLOWS

There are three traffic-flow types that you might wish to inspect and secure:

- **Inbound**—Traffic originating externally and destined to the instances in your VPC
- **Outbound**—Traffic originating from your instance and destined to external resources
- **East-west**—Traffic between instances within your VPC

Inbound Traffic from the Internet

Inbound traffic originates outside the VPC and is destined to services hosted within your VPC, such as web servers. To allow a client on the internet to communicate with an instance behind the firewall, associate a public IP address with the VM-Series firewall. Although it is possible to associate public IP addresses directly to instances in the private subnets, for the firewall to be able to protect those instances, you must associate the public IP address with the firewall. The firewall then translates the destination IP address to the appropriate private resource.

There are two options for providing inbound inspection for multiple applications through the firewall:

- Using a single Elastic IP address with a unique port per application
- Use secondary IP addresses and multiple Elastic IP addresses per application

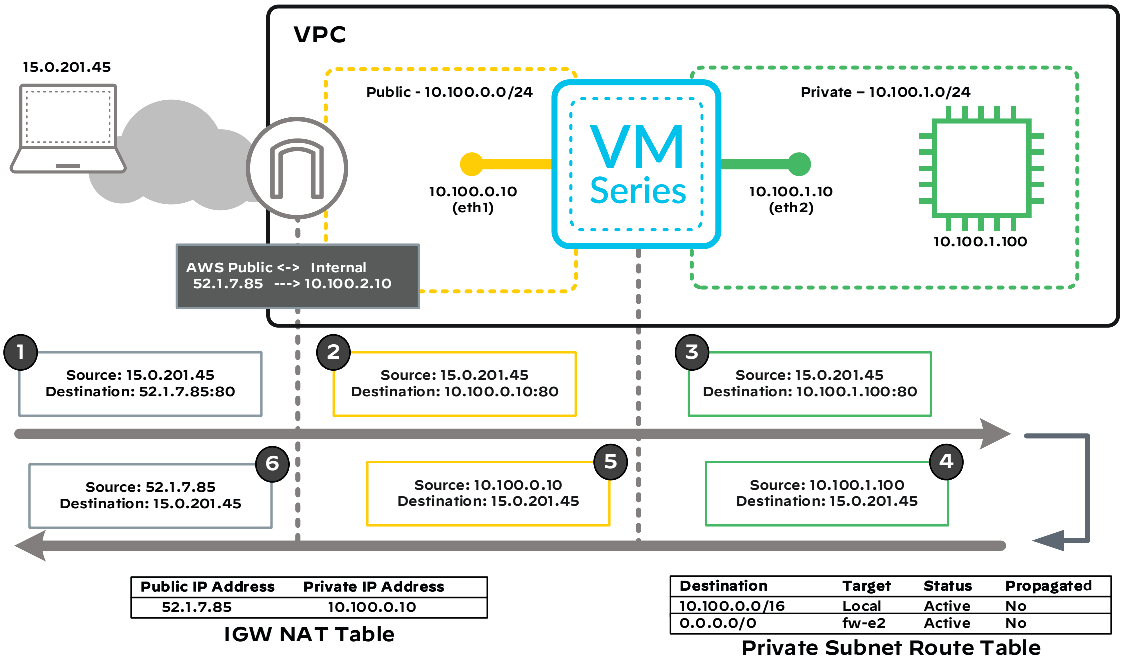
The first option minimizes Elastic IP use, increases the complexity of destination IP address translation on the firewall, and might confuse potential end-users when they are accessing multiple inbound services using the same FQDN or IP address with a different port representing different applications.

A single service port (for example, SSL) might have its external IP address mapped to a single instance's internal IP address that provides the service. The firewall represents additional instances that provide the same service on the same external IP address and uses the service port to differentiate between applications.

Figure 20 illustrates the inbound and return address translation:

1. The client sends traffic to the public IP address of the firewall.
2. The IGW translates the packet's destination address.
3. The firewall translates the packet's destination IP address (and optionally port).
4. The return traffic routes to the firewall based on the subnet's route table.
5. The firewall translates the source IP address to the firewall internet interface in order to match the IGW NAT.
6. The IGW translates the source IP address to the external IP address.

Figure 20 Inbound traffic inspection using destination IP address translation



Public IP Address	Private IP Address
52.1.7.85	10.100.0.10

IGW NAT Table

Destination	Target	Status	Propagated
10.100.0.0/16	Local	Active	No
0.0.0.0/0	fw-e2	Active	No

Private Subnet Route Table

Outbound Traffic Inspection

Outbound traffic originates from within the VPC and is destined to an external resource, typically the internet. Outbound inspection is useful for ensuring that instances are connecting to permitted services (such as Windows Update) and permitted URL categories, as well as for preventing data exfiltration of sensitive information.

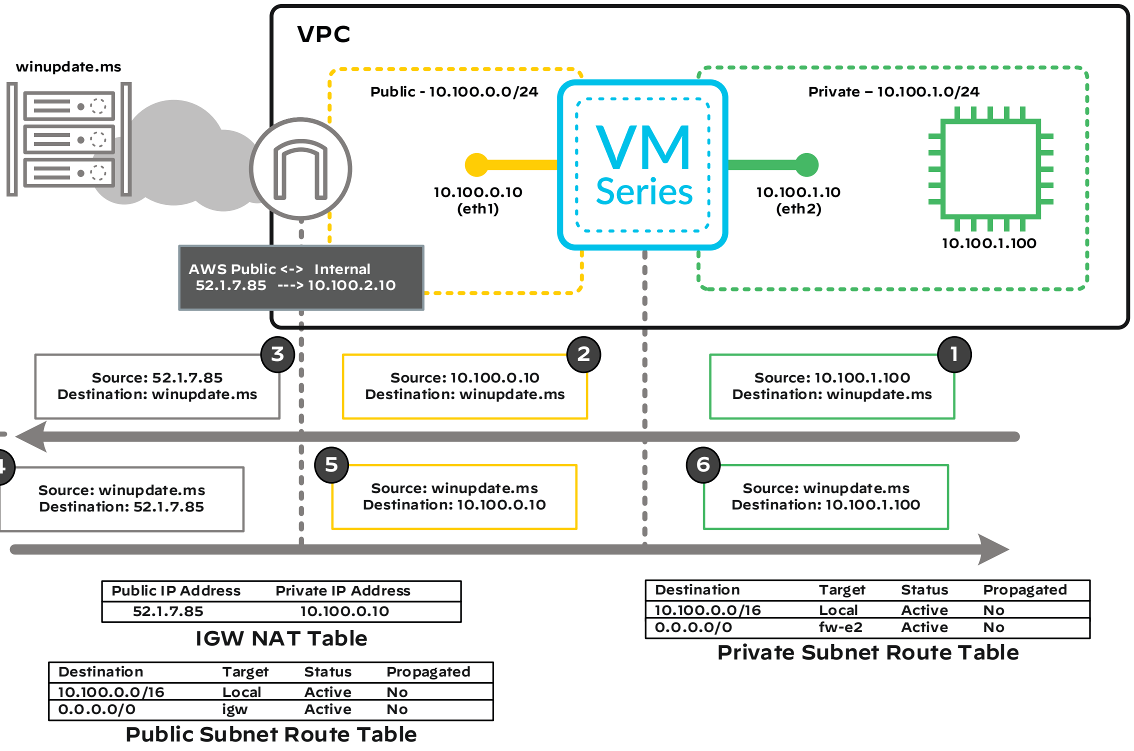
For instances behind the VM-Series firewall to communicate to resources on the internet, the firewall must translate the source IP address of the outbound traffic to its public-facing interface IP address. The IGW then translates the source IP address again as the outbound traffic leaves the VPC.

Outbound traffic through the VM-Series firewall uses source IP address translation on the firewall, ensuring symmetric traffic and firewall inspection of all traffic.

Traffic that originates from an instance on a private subnet and is destined to the internet routes to the firewall through the route table applied to the instance's subnet. Directing outbound traffic to the firewall does not require any changes to the instance. You configure the route table with a default route that points to the firewall's private-facing interface.

Figure 21 illustrates outbound traffic flow. Packet address view 1 has its source IP address modified to that of the firewall interface in packet address view 2. Note that the private subnet route table has a default route pointing to the firewall.

Figure 21 Outbound traffic inspection using source NAT



Public IP Address	Private IP Address
52.1.7.85	10.100.0.10

IGW NAT Table

Destination	Target	Status	Propagated
10.100.0.0/16	Local	Active	No
0.0.0.0/0	igw	Active	No

Public Subnet Route Table

Destination	Target	Status	Propagated
10.100.0.0/16	Local	Active	No
0.0.0.0/0	fw-e2	Active	No

Private Subnet Route Table

East-West Traffic Inspection within a VPC

VPC networking provides direct reachability between all instances within a VPC, regardless of IP address and subnet allocation. All instances within a VPC can reach any other instance within the same VPC, regardless of AWS route tables. You can use the AWS ACLs to permit or deny traffic into or out of an instance or group of instances.

East-West Traffic between VPCs

To provide inspection of east-west traffic, you use multiple VPCs. Group instances with similar security policy requirements in a VPC and inspect inter-VPC traffic. Traffic that originates from an instance in one VPC and is destined to an instance in a different VPC routes to the VM-Series firewall through a transit gateway. The Transit Gateway design model presented later in this guide provides a scalable design for inspection and control of east-west, inbound, and outbound traffic.

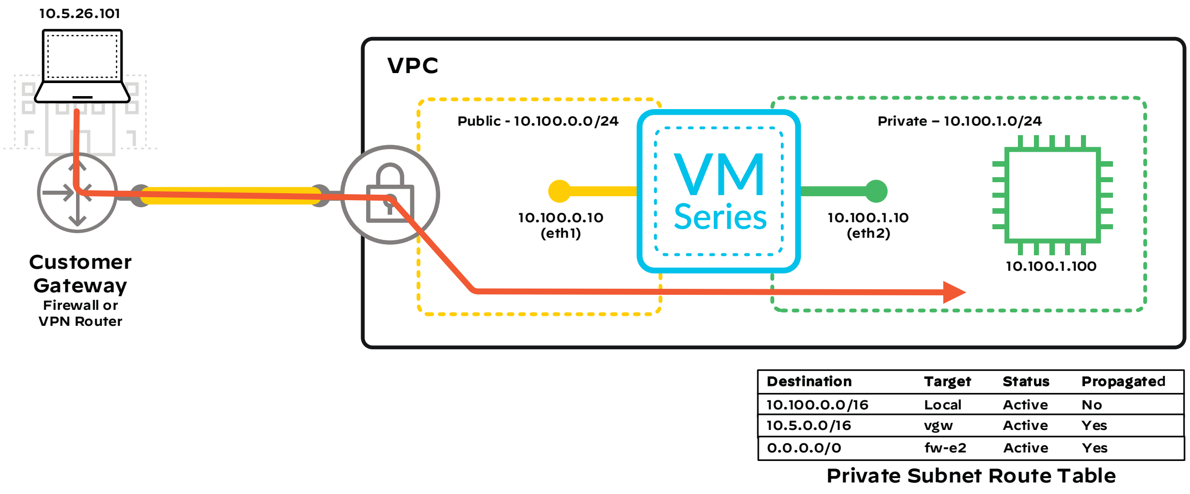
Connections to On-Premises Networks

It can be convenient to have direct connectivity between the private IP addresses of hosts in your on-premises networks to the private IP addresses of your instances in AWS. Direct connectivity also helps administrators and developers to reach resources that do not have public IP address access. You should control access between the on-premises networks and the instances in AWS by protecting the connection with VM-Series or PA-Series firewalls.

Two options when configuring the AWS side of the VPN connection between AWS and your on-premises networks are to use a VGW or to use the VM-Series firewall.

A challenge of using a VGW is that there is no easy way to have the VM-Series firewall control traffic between the VPC and the on-premises networks. Inbound packets that enter through the VGW communicate directly with the instances, bypassing the firewall. If the VPN gateway at the on-premises location is not a firewall, then you have open access from the on-premises network to the instances in the VPC, as shown in Figure 22.

Figure 22 VPN connections to the VGW bypass the VPC firewall

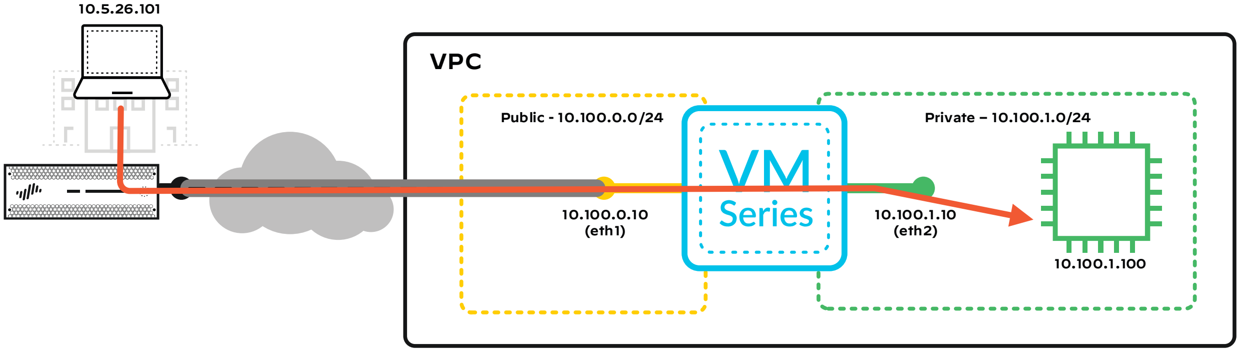


Destination	Target	Status	Propagated
10.100.0.0/16	Local	Active	No
10.5.0.0/16	vgw	Active	Yes
0.0.0.0/0	fw-e2	Active	Yes

Private Subnet Route Table

The preferred design in Figure 23 uses VPN connections between the VM-Series firewalls in AWS and your on-premises networks. With this design, the firewall can inspect and control all inbound VPN traffic. This design removes the risk that the on-premises VPN peer is not a firewall. You can use static or dynamic routing for populating the route tables on the firewalls. The instances in the VPC can remain pointed to the firewalls for their default gateway.

Figure 23 VPN connections to the VM-Series firewalls in the VPC



Destination	Target	Status	Propagated
10.100.0.0/16	Local	Active	No
0.0.0.0/0	fw-e2	Active	Yes

Private Subnet Route Table

AWS allows you to assign IP address subnets to VPN connection tunnels manually, or you can have AWS automatically assign them. Use manual assignment in order to prevent subnet collisions on the firewall interface. When using manual assignment, you must use addressing in the link-local reserved range 169.254.0.0/16 (RFC 3927) and typically use a /30 mask. When using a manual assignment, you should keep track of allocated addresses and avoid duplicate address ranges on the same firewall.

You should take care if you choose automatic assignment, as AWS randomly assigns IP address subnets to VPN connection tunnels from 256 available /30 subnets of the link-local reserved range 169.254.0.0/16 (RFC 3927). For each tunnel subnet, AWS assigns the first available address to the VGW side and the second available address to the CGW. Because the subnet assignments are random, an increasing number of VPN connections from subscriber VGWs results in a greater likelihood of a tunnel subnet collision on the CGW. AWS guidance indicates that at 15 VPN connections (two tunnels for each), the probability of any two tunnel subnets colliding is 50%, and at 25 tunnels, the probability of subnet collision increases to 85%. VM-Series firewalls do not support the assignment of the same IP address on multiple interfaces. You must terminate overlapping tunnel subnets on different firewalls. Because tunnel subnet assignment is random, if you experience a tunnel subnet collision during the creation of a new VPN connection, you have the option to delete the VPN connection and create a new one. The likelihood of subnet collisions continues to increase as the number of VPN connections increases.

Resiliency

Traditionally, you achieve firewall resiliency through a high-availability configuration on the firewall. In a high-availability configuration, a pair of firewalls shares configuration and state information that allows the second firewall to take over for the first if a failure occurs. Although you can configure high availability so that both firewalls are passing traffic, in the majority of deployments, the firewalls operate as an active/passive pair where only one firewall is passing traffic at a time. The VM-Series firewall on AWS does support stateful high availability in active/passive mode for traditional data center-style deployments in the cloud. However, both VM-Series firewalls must exist in the same availability zone, and it can take 60 seconds or longer for the failover to take place due to infrastructure interactions beyond the control of the firewall.

Unlike traditional implementations, you can achieve VM-Series firewall resiliency in AWS through the use of native cloud services. The benefits of configuring resiliency through native public cloud services instead of firewall high availability are faster failover and the ability to scale out the firewalls as needed. However, in a public cloud resiliency model, the firewalls do not share configuration and state information. Applications typically deployed in a public cloud infrastructure, such as web- and service-oriented architectures, do not rely on the network infrastructure to track session state. Instead, they track session data within the application infrastructure, which allows the application to scale out and be resilient independent of the network infrastructure.

The AWS resources and services used to achieve resiliency for the application and firewall include:

- **Availability zones**—Ensure that a failure or maintenance event in an AWS VPC does not affect all VM-Series firewalls at the same time.
- **ECMP**—Available with the AWS transit gateway service, using VPN attachments and dynamic routing with BGP.
- **Load balancers**—Distribute traffic across two or more independent firewalls that are members of a common target group. Every firewall in the load balancer's pool of resources actively passes traffic, allowing firewall capacity to scale out as required. The load balancer monitors the availability of the firewalls through TCP or HTTP probes and updates the pool of resources, as necessary.

AWS load balancers use availability zones for resilient operation as follows:

- The load-balancer frontend can live in multiple zones. This prevents an outage of one zone from completely taking down application access.
- The load balancer can address targets in multiple availability zones. This allows upgrades and migrations to happen without shutting down complete sections.

Another way that firewall resiliency in AWS differs from traditional firewall high availability is that in AWS, you do not implement firewall resiliency at a device level. Instead, you implement firewall resiliency based on the direction of the traffic.

Resiliency for Inbound Application Traffic

You can implement resiliency in AWS for inbound applications through the use of an AWS load-balancer sandwich design. This design uses a resilient, public-facing load balancer and a second resilient load balancer on the private side of the firewalls.

Traffic routing services such as domain name servers and firewall next-hop configurations use FQDN to resolve load-balancer IP addresses versus hard-coded IP addresses. Using FQDN allows the load balancers to be dynamic and scale up or down in size, as well as remain resilient; one load balancer might go down, and the other can continue feeding sessions to the application instances.

To analyze the pieces of the load balancer design, you can walk through the steps in Figure 24. This scenario illustrates a user accessing an application example.lb.aws.com located on the instances:

1. The URL request from the end user is directed toward example.lb.aws.com. This request is sent to a DNS. In this case, the AWS Route 53 cloud-based DNS resolves to an A record for the public load balancers.

2. The DNS resolves to one of two public IP addresses for the public load balancers. There are two IP addresses, one for each of the public load balancers, which are located in separate availability zones in order to provide resiliency.

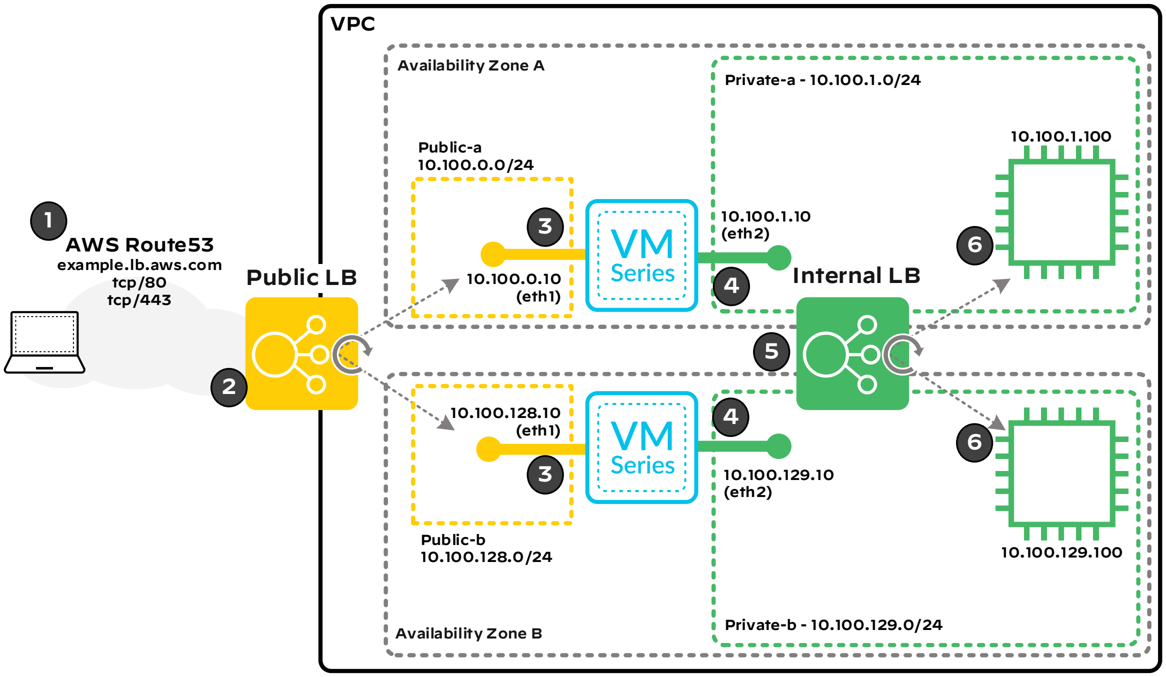
3. The public load balancer has two targets for the next hop. The two targets are the private IP addresses for the public-facing interface on each of the VM-Series firewalls. IP addresses 10.100.10.10 and 10.100.110.10 provide two paths for the incoming connection. The public load balancer translates the packet's destination address to one of the two VM-Series firewall target address and translates the source IP address to the private IP address for the public load balancer so that traffic returns to it on the return flow.

4. Each firewall translates the IP addresses on the incoming HTTP requests. The firewalls change the source IP address to the IP address of the firewall's private interface so that the return response traffic from the instance travels back through the same firewall in order to maintain state and translation tables. The firewall changes the destination IP address to the IP address of the internal load balancer.

5. The firewall learns the IP addresses for the redundant internal load balancers by sending a DNS request for the FQDN assigned to the load balancers. This requests one of two IP addresses, one for each of the internal load balancers, which are located in separate availability zones.

6. The internal load balancers have instances in both availability zones and do a round-robin load balancing to the active instances in the target list.

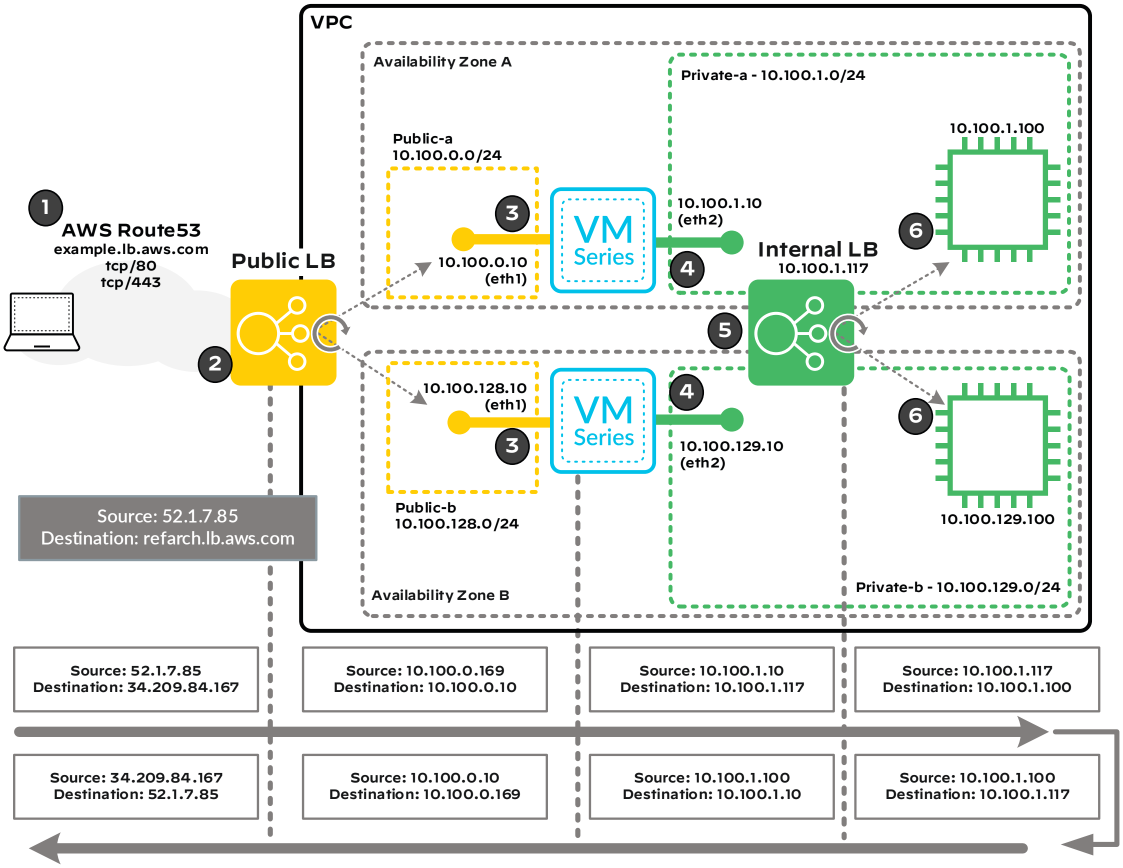
Figure 24 Inbound resiliency



The redundant load-balancer design uses health probes to make sure that each path is operational and that instances are operational. This liveliness check also serves to guarantee the return path is operational as the public load balancer probes through the firewall to the internal load balancer. The internal load balancers probe the instances in their target group to make sure they are operational.

Figure 25 shows the network address translations as the packet moves from the outside to the inside of the VPC and then returns.

Figure 25 Load-balancer design with address translation

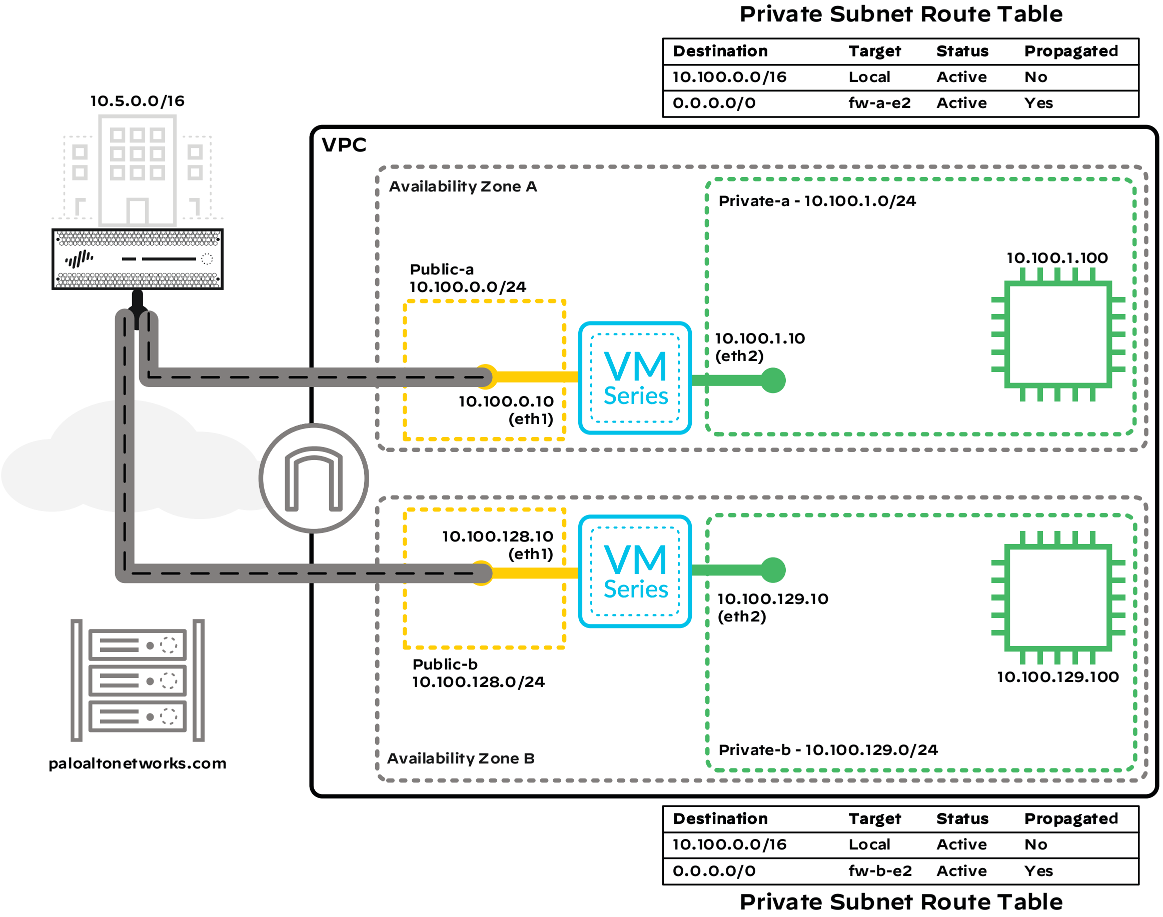


Resiliency for Outbound Traffic and Traffic Bound to On-Premises Networks

The private subnet route table has a default route pointing to the firewall's private interface. The firewall provides the path and the address translation required to reach the internet. The firewall protects the outbound traffic flows and associated return traffic against threats. The instances in the first availability zone exit via the firewall in the first availability zone. The instances in the second availability zone exit the firewall in the second availability zone. This configuration provides resilience on an availability zone basis.

The VPN links terminate on the firewalls in each availability zone. The firewalls route the on-premises traffic to the private instances. This design has the same resilience as outbound traffic, because the instances use the default route to get to anything other than other instances in the VPC. The Transit Gateway design model, covered later in this guide, offers a fully resilient outbound, inbound, and east-west design.

Figure 26 Outbound and on-premises traffic flow



Private Subnet Route Table

Destination	Target	Status	Propagated
10.100.0.0/16	Local	Active	No
0.0.0.0/0	fw-a-e2	Active	Yes

Destination	Target	Status	Propagated
10.100.0.0/16	Local	Active	No
0.0.0.0/0	fw-b-e2	Active	Yes

Private Subnet Route Table

Design Models

There are many ways to use the concepts discussed in the previous sections to build a secure architecture for application deployment in AWS. The design models in this section offer example architectures for centralized management and securing inbound and outbound application traffic flows, communication between private instances, and the connection to your on-premises networks.

As part of the overall AWS architecture, you use a separate management VPC to create a centralized management location so that a single Panorama deployment can manage VM-Series firewalls deployed across all of your organization's VPCs. Panorama streamlines and consolidates core tasks and capabilities, enabling you to view all your firewall traffic, manage all aspects of device configuration, push global policies, and generate reports on traffic patterns or security incidents. You deploy Panorama in Management-Only mode and securely access it over the public internet. The VM-Series firewalls encrypt and send all firewall logs to Cortex Data Lake over TLS/SSL connections.

The design models presented here differ in how they provide resiliency, scale, and services for the design. The design models in this reference design are:

- **Single VPC**—Proof-of-concept or small-scale, multipurpose design
- **Transit Gateway**—High-performance solution for connecting large quantities of VPCs, with a scalable solution to support inbound, outbound, and east-west traffic flows through separate dedicated security VPCs

CHOOSING A DESIGN MODEL

When choosing a design model, consider the following factors:

- **Scale**—Is this deployment an initial move into the cloud and a proof of concept? Will the application load need to scale quickly and modularly? Are there requirements for inbound, outbound, and east-west flows? The Single VPC design model provides inbound traffic control and scale, outbound control, and outbound scale on a per-availability-zone basis. The Transit Gateway design model offers the benefits of a highly scalable design for multiple VPCs connecting to a central hub for inbound, outbound, and VPC-to-VPC traffic control and visibility.
- **Resilience and availability**—What are the application requirements for availability? The Single VPC model provides a robust inbound design with load balancers to spread the load, detect outages, and route traffic to operational firewalls and instances. The Transit Gateway model provides a highly resilient and available architecture for inbound, outbound, and east-west traffic flows.
- **Complexity**—Understanding application flows and how to scale and troubleshoot is important to the design. Placing all services in a single VPC might seem efficient but could be costly in design complexity. Beyond the initial implementation, consider the Transit Gateway design model for a more intuitive and scalable design.

SINGLE VPC DESIGN MODEL

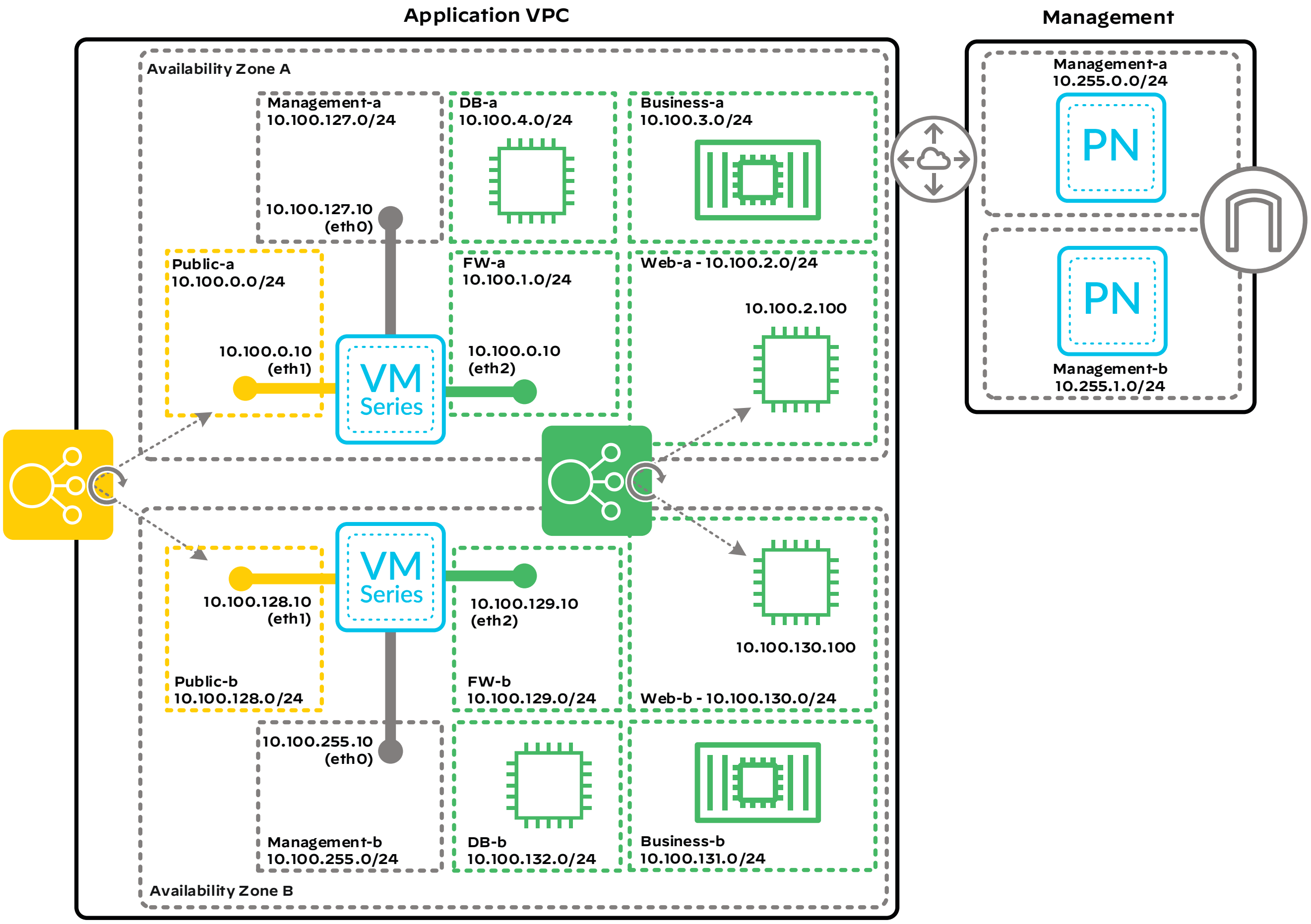
A single standalone VPC might be appropriate for small AWS deployments that:

- Provide the initial move to the cloud for an organization.
- Require a starting deployment that they can build on for a multi-VPC design.

For application resiliency, the architecture consists of a pair of VM-Series firewalls, one in each availability zone within your VPC. You sandwich the firewalls between load balancers for resilient inbound web application traffic and the return traffic. The firewalls are capable of inbound and outbound traffic inspection that is easy to support and transparent to DevOps teams.

You can use security groups and network access control lists to further restrict traffic to individual instances and between subnets. This design model provides the foundation for other architectures in this guide.

Figure 27 Single VPC design model



Inbound Traffic

For resiliency, you deploy two VM-Series firewalls, each in separate availability zones.

There are two options for load balancing inbound traffic:

- **Network Load Balancer**—Choose this option if you require load balancing only at Layer 4 (TCP/UDP). Health checks monitor the application instances through TCP or web server responses.
- **Application Load Balancer**—Choose this option if you require load balancing at Layer 7 (the application layer) for HTTP and HTTPS. The application load-balancer capabilities include host- and path-based routing as well as SSL offloading. Health checks in this design directly monitor the health of the target web server instances.

Inbound Traffic with a Network Load Balancer

For inbound traffic, a Network Load Balancer distributes inbound traffic to the VM-Series firewalls. The NLB is associated with the availability zones that contain VM-Series firewalls. Because the NLB proxies the inbound traffic, you can use security group rules on the public interfaces of the firewalls to allow only inbound traffic from other IP addresses on the public subnets.

The NLB forwards traffic destined to the load balancer's FQDN and port pair to the VM-Series firewalls in the target pool. Common ports required for inbound traffic include TCP port 80 (HTTP) and TCP port 443 (HTTPS). The load balancer distributes traffic between the VM-Series firewalls based on the traffic *5-tuple*, which is the source zone, source IP address, destination zone, destination IP address, and destination port defined in the security policy. The public load balancer's health checks monitor target instance availability through the VM-Series firewalls to the private instances.

ACLs block all inbound traffic to the private instances except for TCP 80 and 443 traffic that traverses through the VM-Series firewall. This approach ensures that internet traffic can communicate with private instances only through the firewall.

The VM-Series firewall applies both a destination and source IP address translation to inbound traffic. The firewall translates the destination IP address from the private IP address of the firewall's public interface to the private instance or load balancer in the private subnets. The firewall translates the source IP address to the IP address of the private interface of the firewall, ensuring return traffic flows symmetrically.

The firewall security policy allows appropriate application traffic to the instances in the private subnets while firewall security profiles prevent known malware and vulnerabilities from entering the network in traffic allowed by the security policy.

Inbound Traffic with Application Load Balancer

For inbound traffic, the Application Load Balancer terminates incoming connections to its frontend and initiates corresponding new connections to the VM-Series firewalls in the target pool. The ALB is associated with the availability zones that contain VM-Series firewalls. If you configure the ALB for multiple web applications that are behind the same set of VM-Series firewalls, you must define unique target pools for each application. Each target pool contains the same VM-Series firewall instance groups but has unique TCP ports assigned.

AWS sources all new connections from the Application Load Balancer interfaces in the public subnets. The destination IP address is the private IP address of the VM-Series firewall's public interface. Health checks monitor back-end availability on all specified HTTP and HTTPS ports.

Destination IP address translation rules on the VM-Series firewalls map incoming traffic from the ALB frontend to the private instance or internal load balancer. The VM-Series firewall also applies a source IP address translation to inbound traffic. The firewall translates the source IP address to the IP address of the private interface of the firewall, ensuring return traffic flows symmetrically.

The VM-Series firewall security policy allows HTTP and HTTPS application traffic from the load balancer to the private instances, and VM-Series firewall security profiles prevent known malware and vulnerabilities from entering the network in traffic allowed by the security policy. If you want to support the use of HTTP and HTTPS back ends on ports other than 80 or 443, you should configure the services of the security policy rules to include the specific service ports in use instead of *application-default*.

Outbound Traffic

The VM-Series firewalls protect outbound traffic flows and associated return traffic against threats. Configure the route tables for the private subnets so that their default route points to the VM-Series firewall's private interface in their availability zone. You configure the subnets associated with the first availability zone to exit the VPC through the firewall in the first availability zone, and you configure the subnets in the second availability zone to point to the second firewall. This configuration provides resilience for outbound and return traffic on an availability-zone basis.

You use VM-Series firewall security policies to limit what applications and resources the private instances can reach. In most designs, the VM-Series firewall does not need to translate the destination IP address. The VM-Series firewall must translate the source IP address to the IP address of the VM-Series firewall's public interface. Without this source NAT, traffic might not return to the firewall. The default route in the public subnets route table directs traffic from the VM-Series firewall to the IGW. When the outbound traffic leaves the public VPC network, the IGW translates the source address to the public IP address associated to the VM-Series firewall's public interface.

The VM-Series firewall security policy allows appropriate application traffic from private instances to the internet. You should implement the outbound security policy by using positive security policies (*whitelisting*). Security profiles prevent known malware and vulnerabilities from entering the network in return traffic allowed by the security policy. URL filtering, file blocking, and data filtering protect against data exfiltration.

East-West Traffic

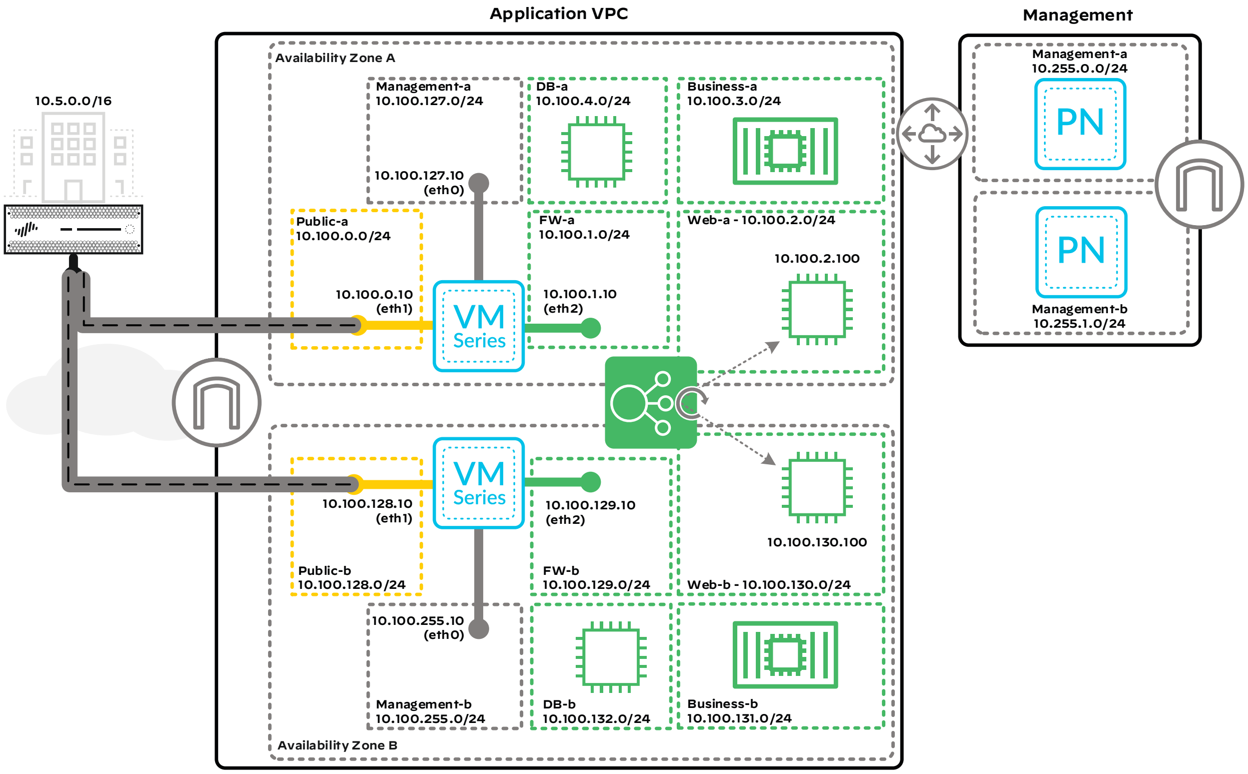
East-west traffic, or traffic between subnets within a VPC, always goes directly between instances. AWS route tables cannot override this behavior, and a limitation of the Single VPC design model is that the VM-Series firewall cannot have control over or visibility to east-west traffic. You can use network ACLs to restrict traffic between subnets. Still, they are not a replacement for the visibility and control provided by the VM-Series firewalls if you need to segment out instances within a VPC. Consider the Transit Gateway design model if you need visibility and control over east-west traffic flows.

Backhaul or Management Traffic

To get traffic from on-premises resources to private instances, VPN connections from on-premises gateways connect to the VM-Series firewalls. Depending on the resiliency required, one or more IPSec tunnels should connect from each of the AWS VM-Series firewalls to the on-premises gateways. The default route configuration for outbound traffic in the private subnets provides the path for traffic from the private instances to reach on-premises resources through the VM-Series firewalls and vice versa. Backhaul traffic has the same resilience characteristics as the outbound traffic flows

The IPSec tunnels terminate on the public interface of the VM-Series firewall. The VPN tunnel interfaces on the VM-Series firewalls are part of a VPN security zone so that you can configure a policy for VPN connectivity that is separate from the outbound public network traffic. Security policies on the VM-Series firewalls only allow required applications through the dedicated connection from the on-premises resources in the VPN security zone.

Figure 28 Single VPC design model—VPN connection



TRANSIT GATEWAY DESIGN MODEL

This guide describes two designs for providing a scalable, secure architecture for the TGW:

- Multiple security VPCs with VPC-only attachments
- Multiple security VPCs with VPC and VPN attachments

This guide briefly covers the first design and then provides more detail about the second, which is the recommended approach because it routes around failures faster by using dynamic routing and ECMP.

In both designs, you connect the spoke VPCs with a VPC attachment. The spoke VPCs can scale up to thousands of VPCs.

What differs between the designs is how you attach the VPCs that contain the VM-Series firewalls to the TGW. You deploy three security VPCs, each with a pair of VM-Series firewalls. Each security VPC controls a specific traffic flow: inbound traffic, outbound traffic, and east-west traffic. Even though you could deploy one security VPC with a pair of firewalls for all traffic, separating the security for each traffic flow allows you to scale up that security function when needed. For example, you might need more firewalls for inbound and its return traffic than for outbound or east-west traffic.

For resiliency, deploy the firewalls in different availability zones.

You can connect your on-premises networks via AWS Direct Connect, VPNs, or both.

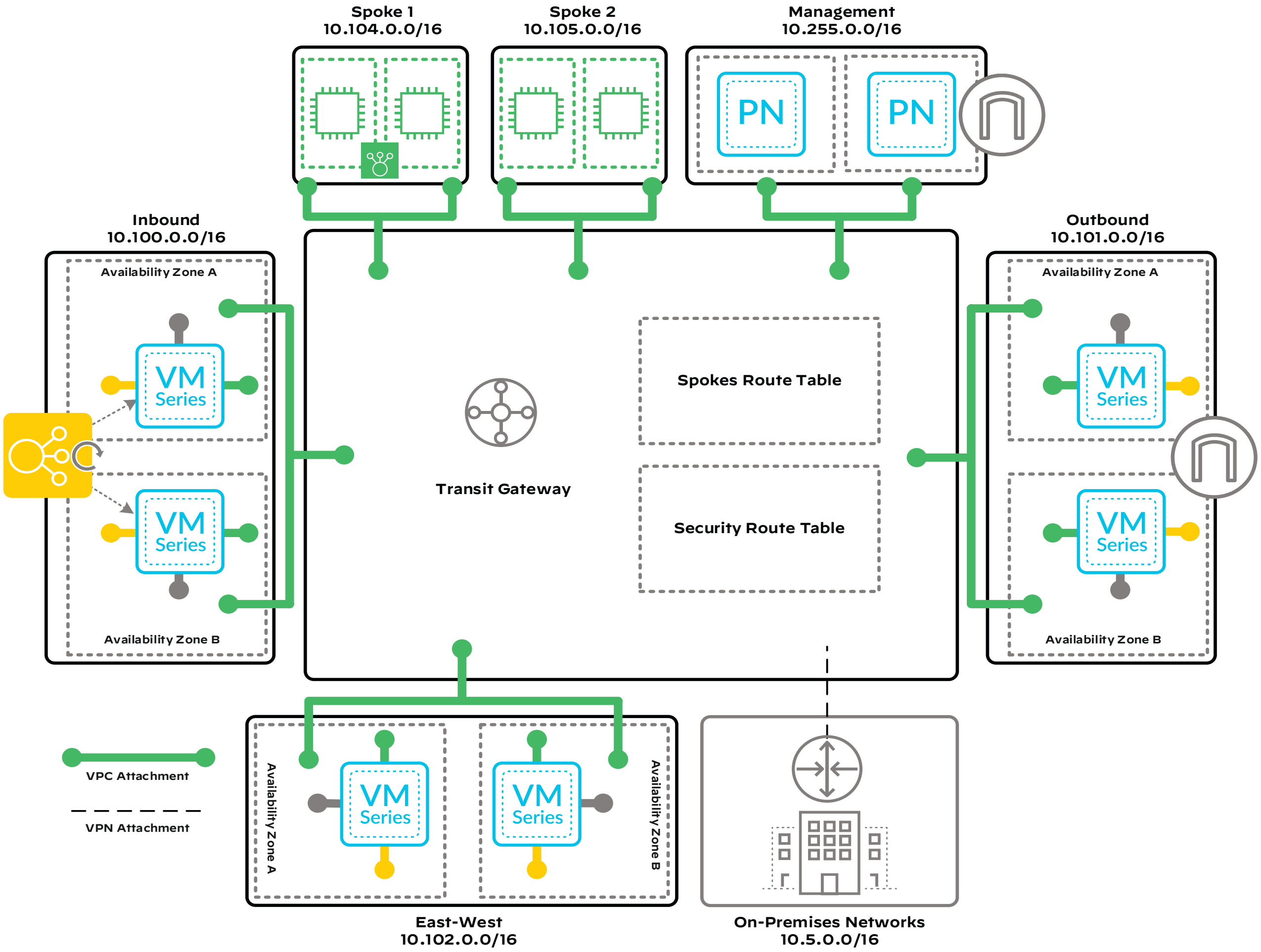
A VPN connection has a limit of 1.25Gbps. To overcome the VPN bandwidth limitation, you can use ECMP routing to aggregate multiple VPN connections. These designs allocate one subnet per availability zone for the network interfaces of the VPC attachment.

Multiple Security VPCs with VPC-Only Attachments

In this design, you attach the three security VPCs (Security-In, Security-Out, and Security-East-West) to the TGW with VPC attachments. With the VPC-only attachment method for the outbound and east-west security VPCs, AWS limits you to static routing; there is no ECMP support. During an outage, you must reconfigure the static routes to an alternative firewall's network interface. You can do this manually or automate it by using AWS CloudWatch, AWS Lambda, and an AWS CloudFormation template script for detection and failover of the firewalls. This automation can take minutes, which is challenging for many customers.

The advantage of VPC attachment is a simple, high-bandwidth design with no VPN tunnels. The disadvantages are the lack of support for ECMP and the inability to provide fast, automatic failover for the firewalls securing outbound and east-west traffic flows.

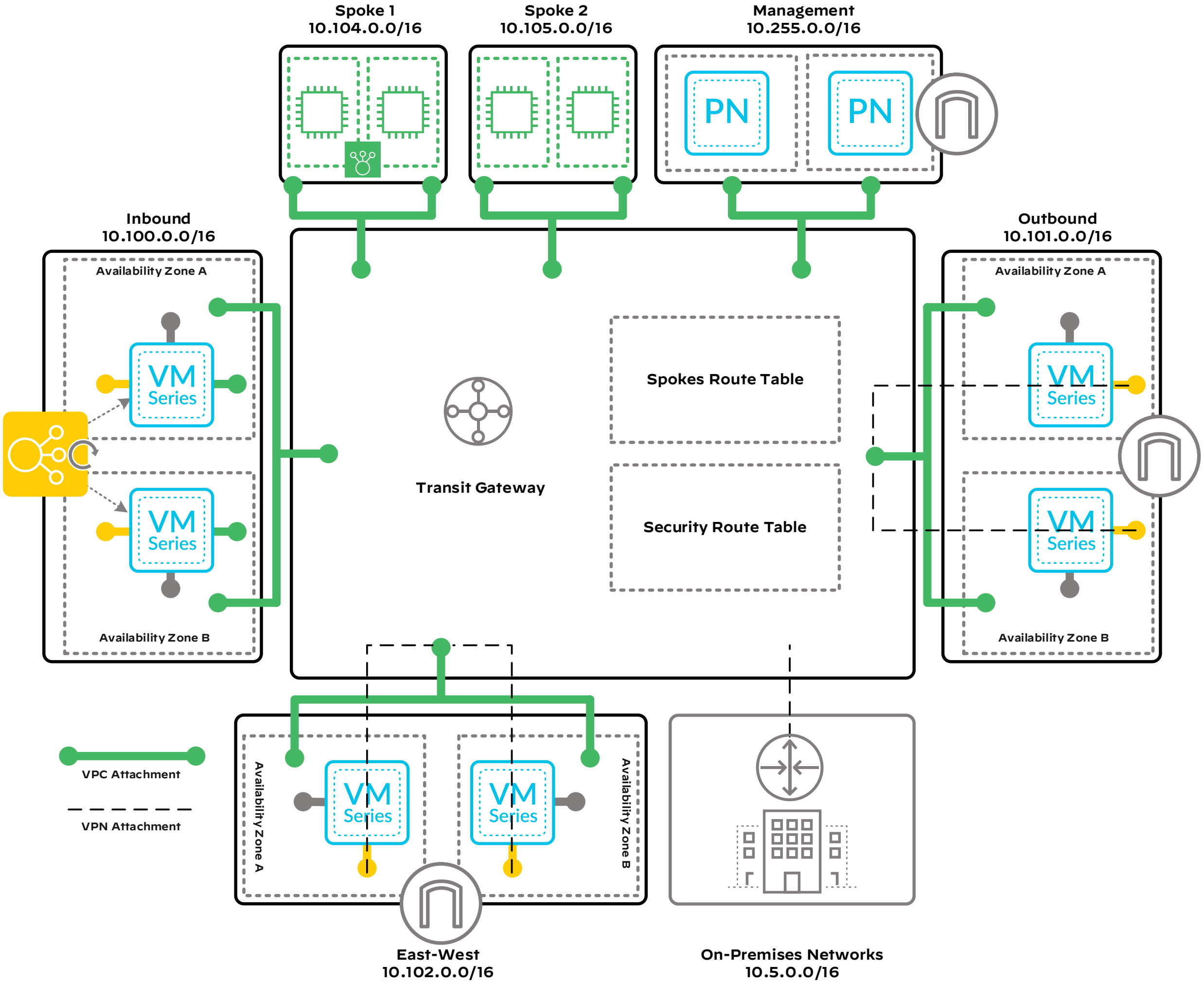
Figure 29 Multiple security VPCs with VPC-only attachments



Multiple Security VPCs with VPC and VPN Attachments

This design is recommended over the VPC-only design because it quickly detects and corrects path failures in the outbound and east-west security VPCs by using ECMP and dynamic routing. In this design, the inbound security VPC uses a VPC attachment type because the load balancers in the inbound security VPC ensure path connectivity and rapid failover. In this design, the outbound and east-west security VPCs connect to the TGW via VPN attachment for data traffic and via a VPC attachment for the firewall management traffic. This ensures a fast recovery time during an outage in the availability zone or VM-Series firewall.

Figure 30 Multiple security VPCs with both VPC and VPN attachments



TGW Design

The TGW design uses two route tables (spokes and security) with all the necessary propagated routes. The TGW in this scenario has:

- VPC attachments for each spoke VPC.
- One VPC attachment for the inbound security VPC.
- Two VPN attachments for the outbound security VPC (two IPSec VPNs from each firewall to the TGW).
- Two VPN attachments for the east-west security VPC (two IPSec VPNs from each firewall to the TGW).
- One VPC attachment for the outbound security VPC, for firewall management in the event that the VPN tunnels are down.
- One VPC attachment for the east-west security VPC, for firewall management in the event that the VPN tunnels are down.

Routing

TGW route tables behave like route domains. You can achieve segmentation of the network by deploying multiple route tables on the TGW and associating VPCs and VPNs to them. You can create isolated networks, allowing you to steer and control traffic flow between VPCs and on-premises connections. This design uses two TGW route tables: security and spokes. The security route table on the TGW has all of the routes propagated to it so that the VM-Series firewall can reach all the VPCs. The spokes route table on the TGW has routes to all the security VPCs but does not have direct routes to other spokes. Only including the routes to the VM-Series firewalls ensure spoke-to-spoke communication can only occur through the VM-Series firewalls in the east-west security VPC.

On the spoke VPCs, VPC route tables route traffic to the TGW. After traffic reaches the TGW, TGW route tables route the traffic to the destination VPC. TGW attachments are associated with a single TGW route table. Each table can have multiple attachments.

You can configure static routes within the TGW route table, or you can use the TGW attachments to propagate routes into the TGW route table. Routes that propagate across a VPN connection with BGP support ECMP.

VPC attachments don't support ECMP. Static routes allow only a single route of the same destination, pointing to a single next hop. This means you can't configure two default routes in the same route table in order to separate next hops.

Note

Even though the TGW route tables can support up to 10,000 routes, the BGP prefix limitation is 100 prefixes per virtual gateway.

Spoke VPCs

Private instances are distributed across the spoke VPCs. The spoke VPCs support direct connection to individual instances or to internal load balancers that distribute traffic between instances within the VPC. In the TGW, don't propagate the spoke VPC routes in the spokes routing table, only propagate it to the security routing table. This routing design ensures east-west traffic between spoke VPCs flows to the VM-Series firewalls in the east-west security VPC.

In this design, each spoke VPC has a default route in the VPC routing table pointing to the TGW as the next hop. The TGW route table for the spoke VPCs has the routes mentioned previously in the Routing section, to allow the spoke VPCs to reach the security VPCs and the on-premises network. For routing between the spoke VPCs, they have to route via the firewalls in the east-west security VPC.

Inbound Traffic

This design deploys a VPC dedicated to inbound security with VM-Series firewalls. The inbound security VPC attaches to the TGW through a VPC attachment. The VPC attachment terminates into the inbound security VPC in a dedicated subnet, one per availability zone.

You deploy the two VM-Series firewalls in separate availability zones and deploy an IGW and ALB to distribute incoming traffic to the firewalls. Each firewall's public-facing, private-facing, and management interfaces attach to separate subnets. Each type of subnet has a separate route table as follows:

- The management route table has the management subnets assigned to it, a default route to the IGW for internet access, and a route to the TGW for access to Panorama.
- The public route table has the public subnets assigned to it and a default route to the IGW for internet access.
- The private route table has the private subnets assigned to it and a static route to the TGW for access to the other VPCs attached to the TGW. To make configuration easier, try to use easily summarized IP address blocks for the spoke VPCs.

You do not need to modify the default routing of the subnets dedicated to the TGW attachment. By default, they can reach all the IP addresses within the inbound security VPC.

The VM-Series firewalls have static routes for all internal networks reachable through the TGW, while the VM-Series firewall's public interface obtains a default route through DHCP.

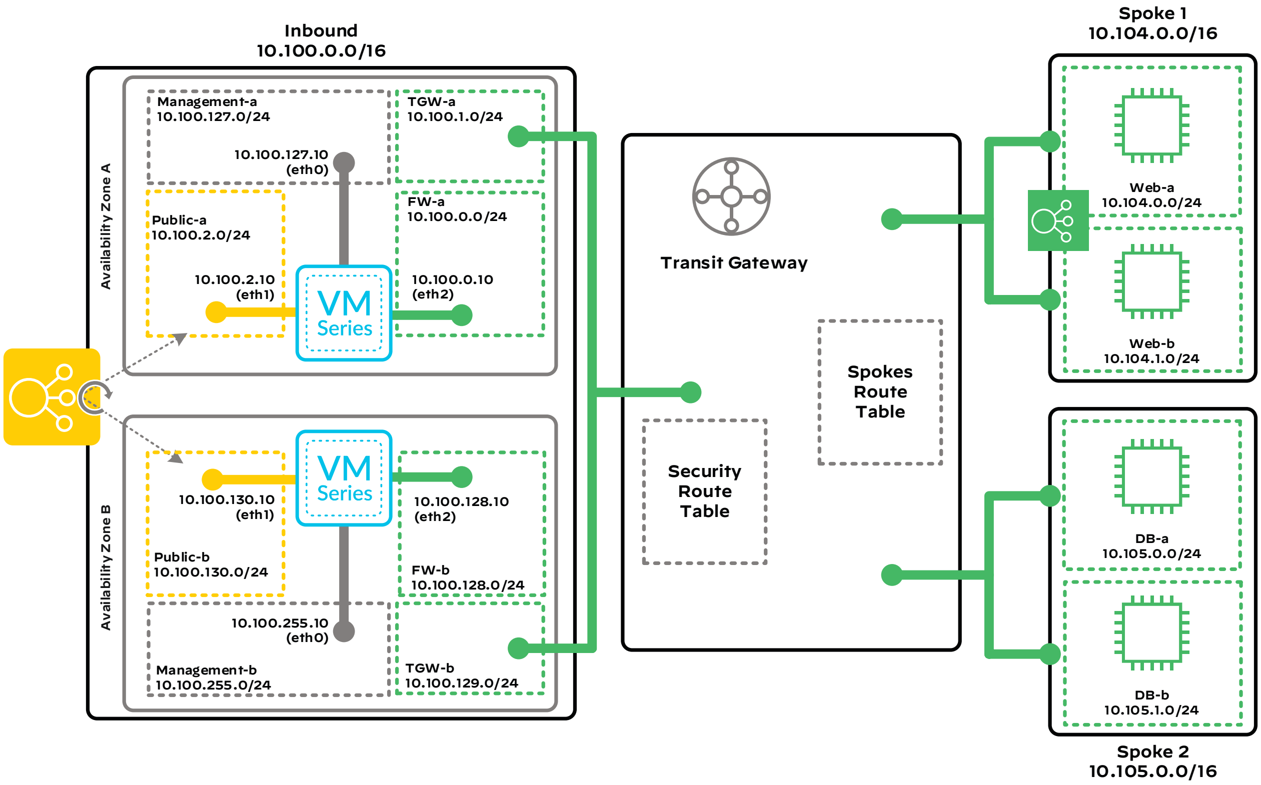
For inbound traffic, the ALB terminates incoming connections to its frontend and initiates corresponding new connections to the VM-Series firewalls in the target pool. The ALB is associated with the availability zones that contain VM-Series firewalls. If you configure the ALB for multiple web applications that are behind the same set of VM-Series firewalls, you must define unique target pools for each application. Each target pool contains the same VM-Series firewall instance groups but has unique TCP ports assigned.

AWS sources all new connections from the Application Load Balancer interfaces in the public subnets. The destination IP address is the private IP address of the VM-Series firewall's public interface. Health checks monitor back-end availability on all specified HTTP and HTTPS ports.

Destination IP address translation rules on the VM-Series firewalls map incoming traffic from the ALB frontend to the private instance or internal load balancer. The VM-Series firewall also applies source IP address translation to inbound traffic. The firewall translates the source IP address to the IP address of the private interface of the firewall, ensuring return traffic flows symmetrically.

The VM-Series firewall security policy allows HTTP and HTTPS application traffic from the load balancer to the private instances, and VM-Series firewall security profiles prevent known malware and vulnerabilities from entering the network in traffic allowed by the security policy. If you want to support the use of HTTP and HTTPS back ends on ports other than 80 or 443, you should configure the services of the security policy rules to include the specific service ports in use instead of *application-default*.

Figure 31 Inbound security



After inbound traffic egresses the VM-Series firewall, the private route table directs the traffic to the TGW. The TGW uses the security route table to direct traffic to the correct spoke VPC. The TGW attachment in the spoke VPC communicates directly with the instance or load balancer in the VPC.

Return traffic follows a default route to the TGW. The TGW uses the spokes route table to direct traffic to the inbound security VPC. The TGW attachment in the inbound security VPC communicates directly with the VM-Series firewall instance, which returns the traffic towards the internet.

Outbound Traffic

This design deploys a VPC dedicated to outbound security with VM-Series firewalls. You connect the VPC to the TGW through two VPN attachments that connect to the two firewalls deployed in the outbound security VPC. Each firewall has two IPSec tunnels, one to each VPN attachment.

The VPC also attaches to the TGW through a VPC attachment for management. The VPC attachment terminates into the outbound security VPC in a dedicated subnet, one per availability zone.

You deploy the two VM-Series firewalls in separate availability zones and deploy an IGW for connectivity to the internet. Each firewall's public-facing and management interfaces are attached to separate subnets. Each type of subnet has a separate route table as follows:

- The management route table has the management subnets assigned to it, a default route to the IGW for internet access, and a route to the TGW for access to Panorama. Use the VPC attachment for this route.
- The public route table has the public subnets assigned to it and a default route to the IGW for internet access and connectivity to the TGW through VPN.

You do not need to modify the default routing of the subnets dedicated to the TGW attachment. By default, they can reach all the IP addresses within the inbound security VPC.

Because the connectivity to the TGW is across an IPSec tunnel, you do not need to configure a private interface on the firewall. The VM-Series firewalls peer to the TGW using eBGP and advertise a default route across their IPSec tunnels. ECMP and dynamic routing using BGP provide peer detection and failover.

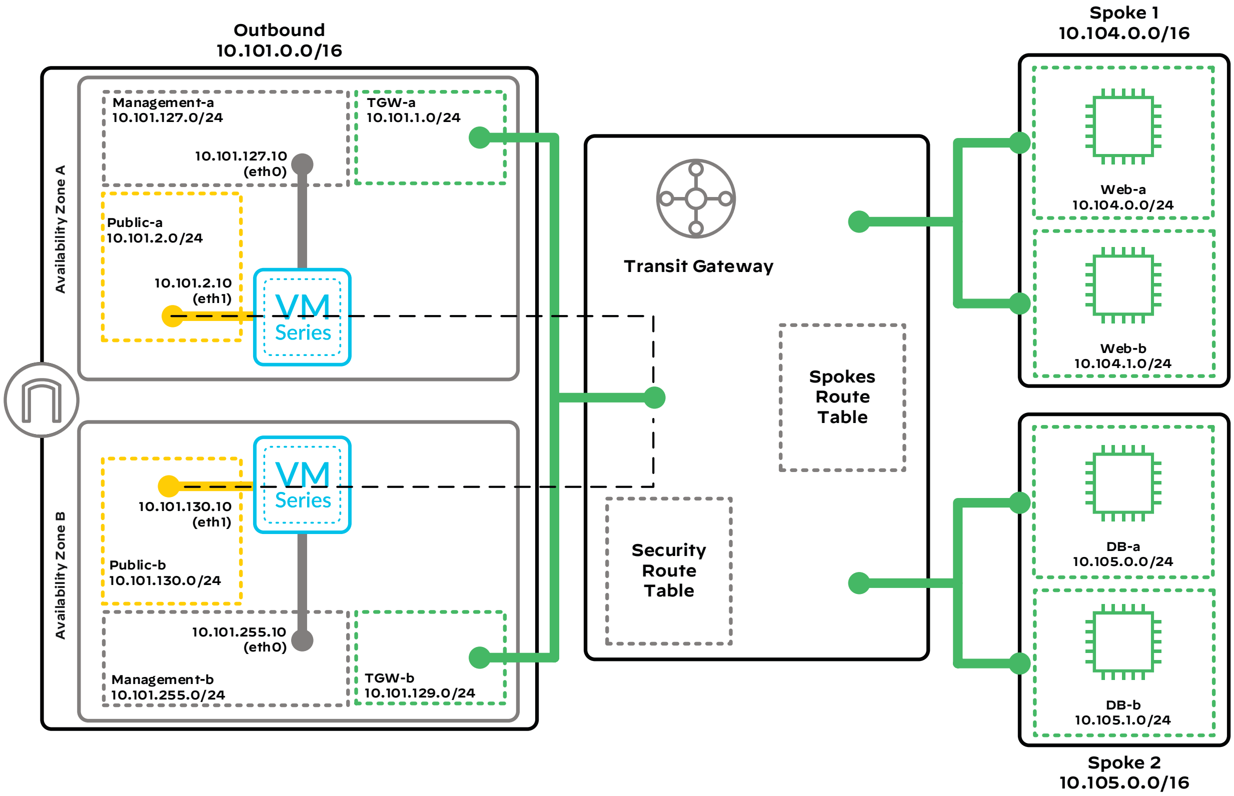
Default routes in the spoke VPCs direct outbound traffic to the TGW. In order to direct traffic to one of the VM-Series firewalls in the outbound security VPC, the TGW uses the spokes route table and the default route learned from the VPN attachments to the VM-Series firewalls.

You use VM-Series firewall security policies to limit what applications and resources the private instances can reach. In most designs, the VM-Series firewall does not need to translate the destination IP address. The VM-Series firewall must translate the source IP address to the IP address of the VM-Series firewall's public interface. The default route in the public subnet's route table directs traffic from the VM-Series firewall to the IGW. When the outbound traffic leaves the VPC network, the IGW translates the source address to the public IP address associated with the VM-Series firewall's public interface.

The VM-Series firewall security policy allows appropriate application traffic from private instances to the internet. You should implement the outbound security policy by using positive security policies (*whitelisting*). Security profiles prevent known malware and vulnerabilities from entering the network in return traffic allowed by the security policy. URL filtering, file blocking, and data filtering protect against data exfiltration.

Return traffic follows the spoke routes learned from the TGW. The TGW uses the security route table to direct traffic to the correct spoke VPC. The TGW attachment in the spoke VPC communicates directly with the instance in the VPC.

Figure 32 Outbound security



East-West Traffic

This design deploys a VPC dedicated to east-west security with VM-Series firewalls. You connect the VPC to the TGW through two VPN attachments that connect to the two firewalls deployed in the east-west security VPC. Each firewall has two IPSec tunnels, one to each VPN attachment.

The VPC also attaches to the TGW through a VPC attachment for management. The VPC attachment terminates into the east-west security VPC in a dedicated subnet, one per availability zone.

You deploy the two VM-Series firewalls in separate availability zones and deploy an IGW for connectivity to the internet. Each firewall's public-facing and management interfaces are attached to separate subnets.

Each type of subnet has a separate route table as follows:

- The management route table has the management subnets assigned to it, a default route to the IGW for internet access, and a route to the TGW for access to Panorama. Use the VPC attachment for this route.
- The public route table has the public subnets assigned to it and a default route to the IGW for connectivity to the TGW through VPN.

You do not need to modify the default routing of the subnets dedicated to the TGW attachment. By default, they can reach all the IP addresses within the inbound security VPC.

Because the connectivity to the TGW is across an IPSec tunnel, you do not need to configure a private interface on the firewall. The VM-Series firewalls peer to the TGW using eBGP and advertise a route that summarizes all IP address blocks in the spoke VPCs, such as 10.0.0.0/8, across their IPSec tunnels. ECMP and dynamic routing using BGP provide peer detection and failover.

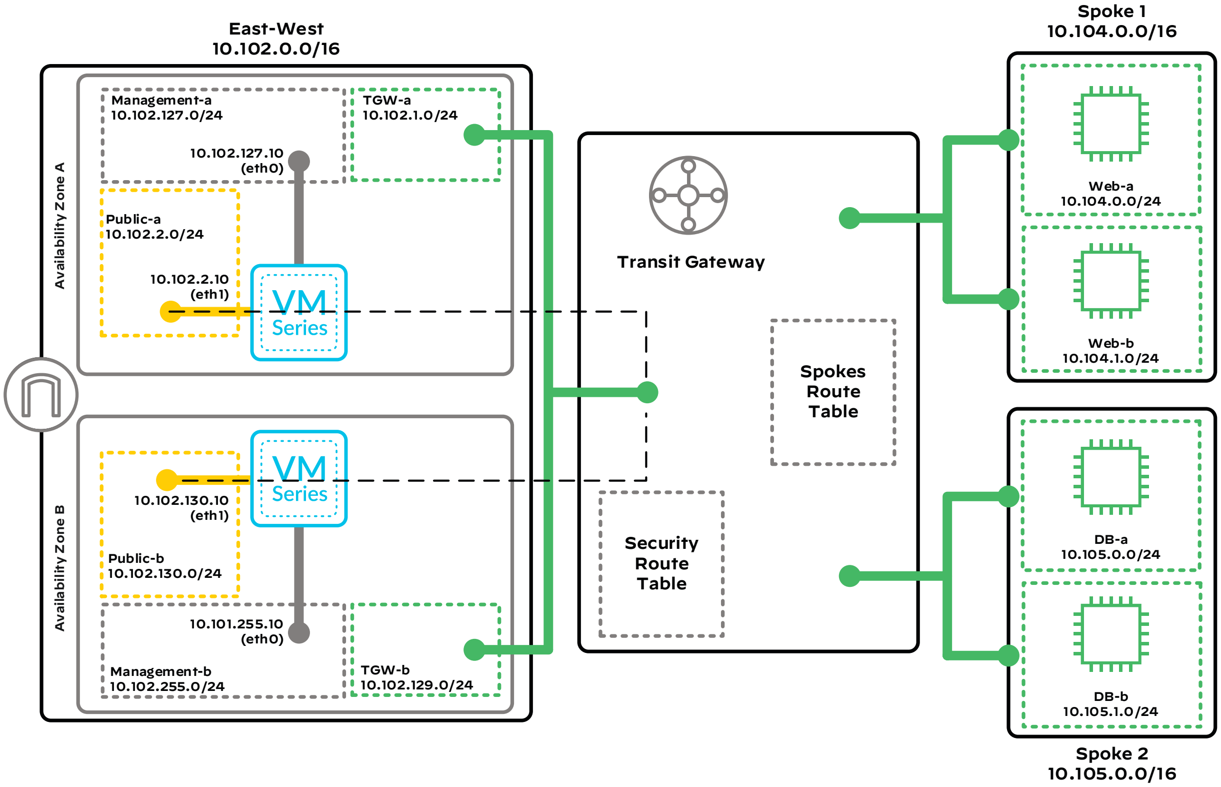
The VPN tunnels advertise 10.0.0.0/8 routes per VPN tunnel into the two TGW routing tables.

This design provides east-west traffic control without address translation. Address translation for east-west traffic flow is challenging to application and database teams. To ensure traffic uses the same firewall in order to avoid asymmetric routing, the east-west firewalls use an active-standby design, which you configure by using the BGP AS-Path attribute. It forces the firewall in the second availability zone to have a longer BGP autonomous system (AS) path than the firewall in the first availability zone. This allows the TGW to prefer the firewall in the first availability zone and failover to the second if there are connectivity issues.

Default routes in the spoke VPCs direct east-west traffic to the TGW. In order to direct traffic to the primary VM-Series firewall in the east-west security VPC, the TGW uses the spokes route table and the summarized internal route learned from the VPN attachments to the east-west VM-Series firewalls.

You use VM-Series firewall security policies to limit what applications and resources the private instances can reach. The VM-Series firewall does not need to perform address translation. The firewall uses the spoke routes learned from the TGW in order to direct traffic to the TGW. The TGW uses the security route table to direct traffic to the correct spoke VPC. The TGW attachment in the spoke VPC communicates directly with the instance in the VPC. Return traffic follows the default route in the spoke VPC to the TGW.

Figure 33 East-west security



Backhaul to On-Premises Traffic

To get traffic from on-premises resources to private instances, you can use VPN connections or AWS Direct Connect. VPN connections from on-premises gateways connect to the TGW as a VPN attachment. Multiple tunnels and ECMP provide resiliency. The default route in the spokes route table provides the path that allows traffic from instances in the spoke VPCs to reach on-premises resources.

You can backhaul to the TGW with Direct Connect either directly in a colocation facility or from on-premises as a service through a WAN provider. Dual connectivity is recommended for resiliency.

Figure 34 Backhaul with Direct Connect gateway

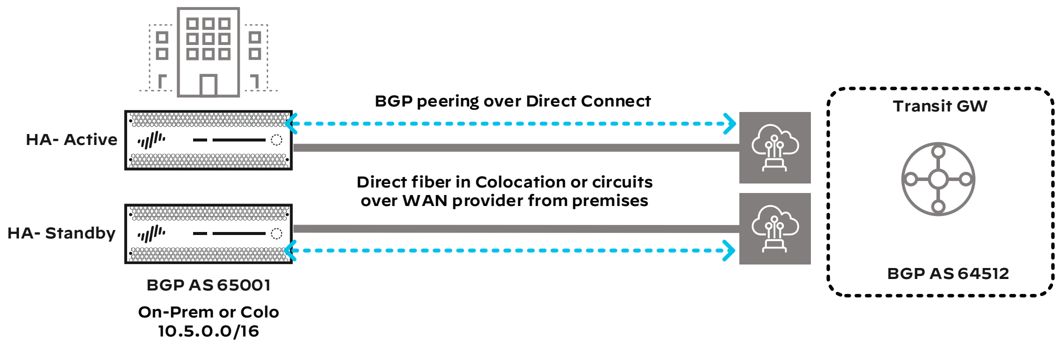
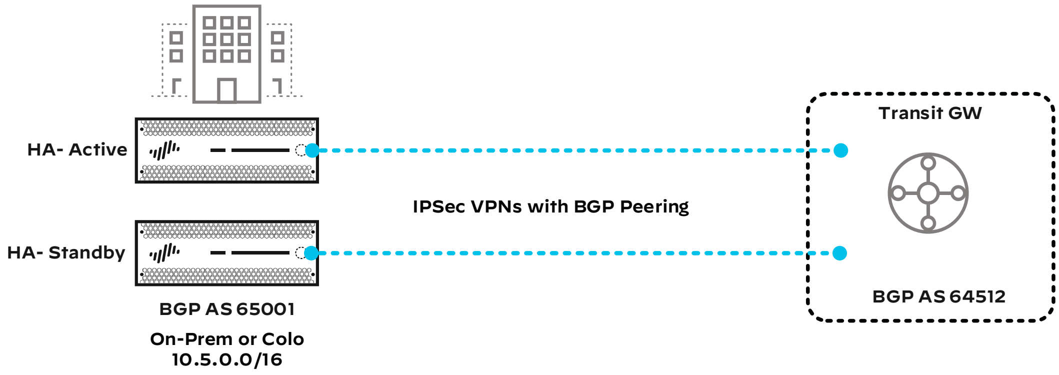

HA- Active
HA- Standby
BGP peering over Direct Connect
Direct fiber in Colocation or circuits over WAN provider from premises
Transit GW
BGP AS 64512
BGP AS 65001
On-Prem or Colo
10.5.0.0/16

Figure 35 shows VPN connectivity from on-premises gateways to the TGW via a VPN attachment. There is a VPN attachment for each CGW, and each attachment is made of two tunnels.

Figure 35 Backhaul with VPN



Management Traffic

This design uses Panorama for the management of the firewalls and uses Cortex Data Lake for logging. You deploy Panorama in an active/standby configuration in a separate, dedicated VPC. You deploy the firewalls with a management interface that routes to Panorama and the internet for software and content updates. The firewalls also need connectivity to subscription services and Cortex Data Lake for logging.

This design connects the dedicated Panorama VPC to the firewalls via a VPC attachment. It is not recommended that you use the VPN attachment because in the event that the VPN tunnels are down, you would lose connectivity to the firewalls.

Scaling

You can scale TGW with thousands of connected VPCs. You can also deploy multiple TGWs per region.

You scale the security solution for each traffic type as follows:

- **Inbound security**—Add additional VM-Series firewalls. The load balancer distributes traffic to the additional firewalls, and source address translation provides for return traffic. You can deploy the firewalls in additional availability zones or within the two existing availability zones.
- **Outbound security**—You can add additional firewalls because you have deployed source address translation and VPN attachments that support ECMP and dynamic routing. You can deploy the firewalls in additional availability zones or within the two existing availability zones.
- **East-west security**—Because address translation is not in use on east-west traffic flows, adding a firewall requires careful planning. You must understand your IP address subnet allocation and understand which spoke IP prefixes you can summarize uniquely in the additional firewalls. The additional firewalls advertise the summarized prefixes for subsets of the spoke VPCs that need east-west inspection.

Summary

Moving applications to the cloud requires the same enterprise-class security as your private network. The shared-security model in cloud deployments places the burden of protecting applications and data on your organization. Deploying Palo Alto Networks VM-Series firewalls in your AWS infrastructure provides a scalable infrastructure with the same protections from known and unknown threats, complete application visibility, a common security policy, and native cloud automation support. Your ability to move applications to the cloud securely helps you to meet challenging business requirements.

The Transit Gateway design model presented in this guide builds upon the initial Single VPC design model in the same way that an organization's environment expands. You build the first VPC as a proof of concept, and as your environment grows, it evolves into a more modular design where you can purpose-build VPCs for the application tier that it houses. Reuse the Single VPC design model for a resilient inbound design, and use the Transit Gateway design model to scale your environment with more visibility and less complexity.

You can use the feedback form to send comments about this guide.

HEADQUARTERS

Palo Alto Networks
3000 Tannery Way
Santa Clara, CA 95054, USA
http://www.paloaltonetworks.com

Phone: +1 (408) 753-4000
Sales: +1 (866) 320-4788
Fax: +1 (408) 753-4001
info@paloaltonetworks.com