Reference Architecture Guide for Google Cloud Platform

JANUARY 2020

Table of Contents

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:

- Retired the single project model.
- Removed the GKE East-West inter-node-pool traffic flows, because these traffic flows were applicable only in the single project model context with a routes-based GKE cluster deployment.
- Updated the shared virtual private cloud (VPC) model to include VPC Network Peering and internal load balancer next-hop. The updated design removes the requirement to have a VM-Series firewall interface in each Shared VPC service project.
- Introduced a new VPC Network Peering model.
- Updated Panorama™ deployment to use a separate centralized management project.
- Changed phrasing, terminology, and diagrams for clarity.

Comprehensive revision history for this guide

Purpose of This Guide

This reference architecture guide describes how your organization can use Palo Alto Networks VM-Series firewalls to bring visibility, control, and protection to applications built on the Google Cloud Platform (GCP).

This guide:

- Links the technical design aspects of GCP and the Palo Alto Networks solutions and then explores several design models. The design models include two variations of multiple project, enterprise-level environments.
- Provides an overview of how Prisma™ Cloud (formerly *RedLock*®) helps organizations manage security risks and compliance in a public-cloud infrastructure.
- Provides a framework for architectural discussions between Palo Alto Networks and your organization.
- Is required reading prior to using the GCP deployment guide series. The deployment guides provide decision criteria for deployment scenarios, as well as procedures for enabling features of GCP and the Palo Alto Networks VM-Series firewalls in order to achieve an integrated design.

AUDIENCE

This design guide is 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 design guide:

- Palo Alto Networks Security Operating Platform Overview—Introduces the components of the Security Operating Platform® and describes the roles they can serve in various designs.
- Securing Data in the Private Data Center and Public Cloud with Zero Trust—Describes how your organization can use the Palo Alto Networks Security Operating Platform in the design of a Zero Trust security policy in order to protect your sensitive and critical data, applications, endpoints, and systems.
- Deployment Guide for Panorama on GCP—Details the deployment of Palo Alto Networks Panorama management on GCP.
- Deployment Guide for GCP—Shared VPC Design Model—Details deployment scenarios and provides step-by-step guidance for the Shared VPC design model on GCP.
- Deployment Guide for GCP—VPC Network Peering Design Model—Details deployment scenarios and provides step-by-step guidance for the VPC Network Peering design model on GCP.

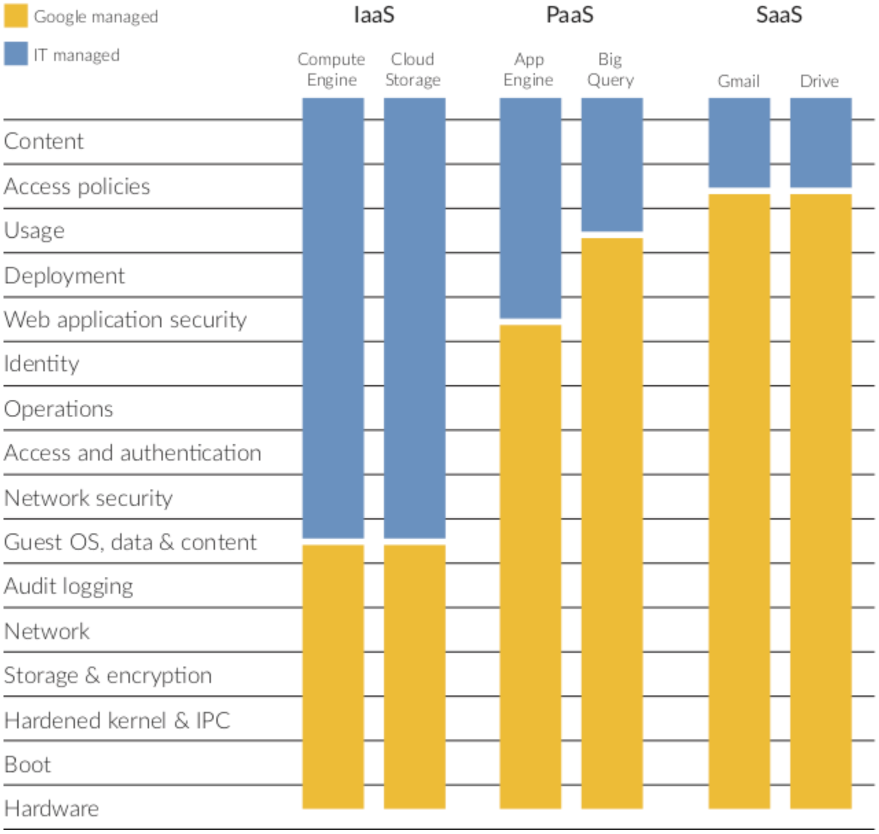
Introduction

Organizations are deploying applications and services on the GCP public-cloud infrastructure for a variety of reasons including:

- **Business agility**—Infrastructure resources are available when and where you need them, minimizing IT staffing requirements and providing faster, predictable time-to-market. Virtualization in both public and private cloud infrastructure 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 they need when they need to bring value to the organization.
- **Operational vs. capital expenditure**—Costs are aligned directly with usage, providing a utility model for IT infrastructure requiring 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, 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



The VM-Series firewall is an integral security enforcement and intelligence gathering component of the Palo Alto Networks Security Operating Platform. The Palo Alto Networks VM-Series firewall deployed on GCP has the same features, benefits, and management as the physical next-generation firewalls you might have deployed elsewhere in your organization. First and foremost, the VM-Series application control and threat prevention capabilities protect your GCP deployments from threats, data loss, and business disruption. Any observed and collected threat intelligence information is shared across other Security Operating Platform components 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 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 to last for multiple years and must size the devices to satisfy the performance requirements during their lifetime.

Public-cloud environments can also scale out the deployment in addition to scaling up. This architectural difference stems primarily from the capability of public-cloud environments to dynamically increase or decrease the number of resources you have allocated. 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.

What this means in practice is that to architect an application for the cloud, you need to distribute functionality and build each functional area to scale out as necessary. Typically, this means a load balancer distributes traffic across a pool of identically configured resources. As application traffic volume changes, the number of resources you have allocated to a pool can be increased or decreased dynamically. This design methodology provides scalability and resiliency. However, the application architecture must consider that the resources are transient. For example, the application should not store state information 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 public-cloud architecture extends not only to the capacity of an application but also 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 goals of a 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 the cloud infrastructure, as well as the resources running on it, have the ability to be bootstrapped, configured, updated, and destroyed programmatically. Compared to a traditional on-premises deployment, 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, not the least of which is the ability to phase in the changes to a subset of the traffic, as well as 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. VM-Series firewalls' capabilities leveraged to achieve this include:

- Application visibility—VM-Series firewalls natively analyze all traffic in a single pass to determine the application, content, and user identity. You use the application, content, and user identity as core elements of your security policy and for visibility, reporting, and incident investigation.

- Prevent advanced attacks 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 service to 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 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 to 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 as well as the GCP project resources. You can use firewall performance metrics and health information to create automated actions based on performance and usage patterns. You can automate policy updates when workloads change by using the fully documented XML API and dynamic address groups to allow VM-Series firewalls to consume external data in the form of tags. The result is that you can deploy new applications and next-generation security simultaneously in an automated manner.

CLOUD INFRASTRUCTURE PROTECTION

GCP provides basic infrastructure components with a responsibility to ensure that the customer's workloads are appropriately isolated from other workloads and that the underlying infrastructure and physical environment are secure. However, the customer has the responsibility for securely configuring the instances, operating systems, and any necessary applications, as well as maintaining the integrity of the data processed and stored by each virtual machine. 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 it is up to the customer to make the assessment and lock those service configurations down to ensure the integrity of the data itself.

Security and compliance risks in cloud computing threaten an organization's ability to drive digital business. The dynamic nature of the cloud, coupled with 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 by using policy guardrails.
- Implement DevOps threat monitoring, which identifies risky configurations, network intrusions, and host vulnerabilities for the management plane. This feature complements the capabilities of the VM-Series 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 determine root cause quickly.
- Prioritize issues and respond appropriately by using contextual alerting.

Through proactive security assessment and configuration management that uses industry best practices, Prisma Cloud makes cloud-computing assets harder to exploit. Prisma Cloud enables organizations to implement continuous monitoring of the GCP infrastructure. It provides an essential, automated, up-to-date status of their security posture that organizations 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 GCP infrastructure from falling out of compliance and to provide visibility into the actual security posture of the cloud to avoid failed audits and the subsequent fines associated with data breaches and non-compliance.

Google Cloud Platform Concepts and Services

When deployed on GCP, VM-Series firewalls rely upon underlying GCP resources and functionality to integrate into the application traffic flow and protect the workload. The concepts covered in this section give an overview of the GCP services relevant to VM-Series firewalls. For additional information, see the GCP documentation, the definitive source of information on these topics.

CONSOLE

GCP provides a variety of interface options for deploying and managing resources. The GCP console provides a graphical front-end as well as command-line control through Google Cloud Shell.

You use the GCP console to deploy, manage, and monitor resources. *Resources* are the components used to build applications and services. Resources include, but are not limited to, virtual machine instances, virtual private clouds, load balancers, and storage services.

For those who prefer programmatic interaction, GCP also provides extensive REST APIs. Those who frequently work with on-site equipment might feel more familiar with the direct deployment of resources through the GCP console.

PROJECTS AND IAM

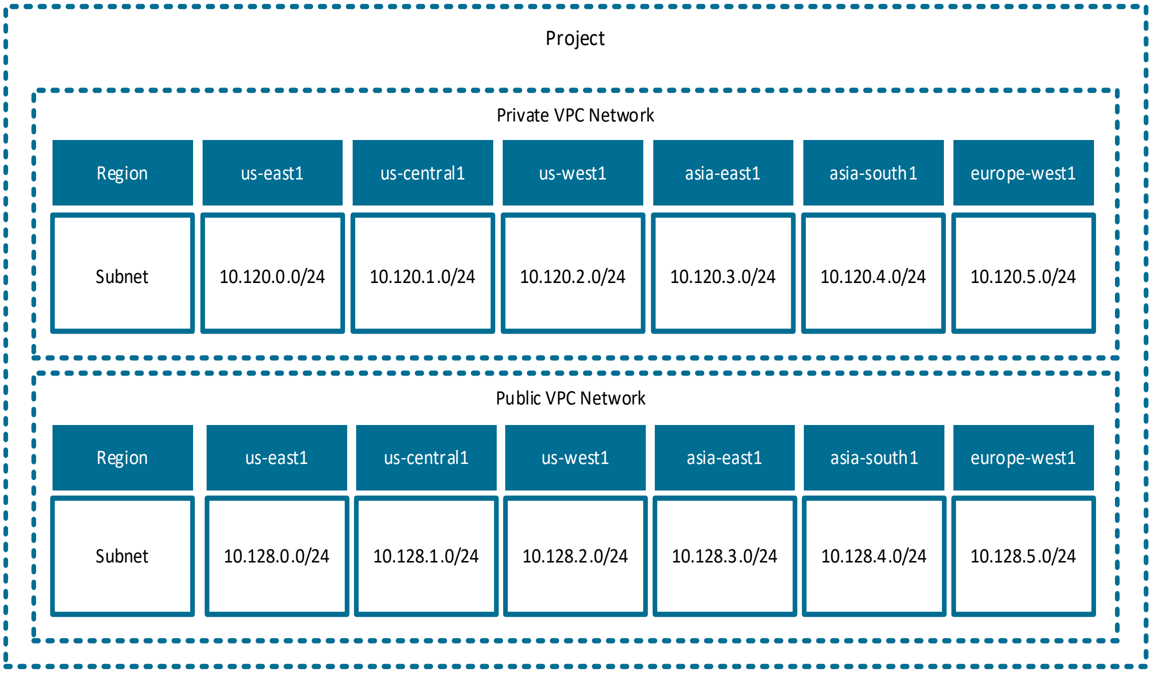
Projects are a way of organizing what you are building. They define settings, who can manage and control resources, and other metadata that describe your applications. All resources (virtual machine instances, VPC networks, load balancers, etc.) belong to a project. Importantly, though, resources can only belong to one project. Proof-of-concepts and personal labs often use a single project that contains all of the needed resources. When deployments grow, most organizations want to implement role-based access control (RBAC) to control who can create and manage resources. Projects provide an easy assignment of RBAC. You define roles in Cloud Identity and Access Management (Cloud IAM) and associate users to roles in the Cloud IAM policy for a project or individual resource.

To provide further configuration flexibility, GCP uses resource hierarchy. Projects inherit settings either directly from their parent organization or from a parent folder that inherits settings from the organization. Every organization has unique requirements for determining how to separate resources, but a common technique is to group resources based on function (network, application).

VIRTUAL PRIVATE CLOUD

A *virtual private cloud* is a logically segmented global network within GCP that allows connected resources to communicate with each other. VPC networks contain one or more private IP subnets, each assigned to a GCP region. A subnet lives in only one region, but all subnets within a VPC network are reachable to the connected resources regardless of their location within GCP or their project membership.

Figure 2 VPC networks



Project

Private VPC Network

Region	us-east1	us-central1	us-west1	asia-east1	asia-south1	europe-west1
Subnet	10.120.0.0/24	10.120.1.0/24	10.120.2.0/24	10.120.3.0/24	10.120.4.0/24	10.120.5.0/24

Public VPC Network

Region	us-east1	us-central1	us-west1	asia-east1	asia-south1	europe-west1
Subnet	10.128.0.0/24	10.128.1.0/24	10.128.2.0/24	10.128.3.0/24	10.128.4.0/24	10.128.5.0/24

A project can have multiple VPC networks. When you want communication between resources in separate VPC networks, you can connect them if there is no overlap in the IP network definition. Virtual machine network interfaces receive IP addresses, default gateways, and DNS servers through DHCP. By default, when you start a virtual machine instance, GCP assigns the first available IP address in the subnet. When you stop an instance, GCP releases any IP addresses allocated to it.. The next time you start the instance, it again obtains the first available IP address in the subnet. In environments where instances often change state, obtaining a consistent IP address is not likely.

When a resource requires it, you can configure a static IP address reservation. This means you do not need to configure static IP addresses in the instance operating system. Instead, GCP reserves the IP address so the instance always receives the same IP address through DHCP. However, unlike dynamic IP address allocation, the IP address remains reserved even when the instance is not running.

An alternative to static IP addressing for consistent connectivity is the use of name resolution to communicate between resources within a VPC network. By default, resources receive the GCP DNS servers through DHCP. The GCP DNS server provides both public name resolution and internal name resolution within the project. The addition or state change of an instance automatically updates the GCP DNS service. When an instance has multiple private IP addresses, its name resolves to its primary private IP address or the IP address assigned to the first interface on the instance.

Although VPC networks do not contain publicly routable IP addresses, each instance's network interface can have a public IP address in addition to a private IP address. Like private IP addresses, public IP addresses can change as an instance changes state. You should configure a static public IP address if you need public IP addressing that doesn't change. You can also dynamically obtain IP address information through the GCP API so that management or automation tools can reflect IP address changes automatically.

If you need multiple private IP addresses on a single network interface, you can configure a network interface with an IP address alias range. You must configure IP addresses from the alias range in the instance's operating system because GCP does not distribute them through DHCP.

Traffic Forwarding

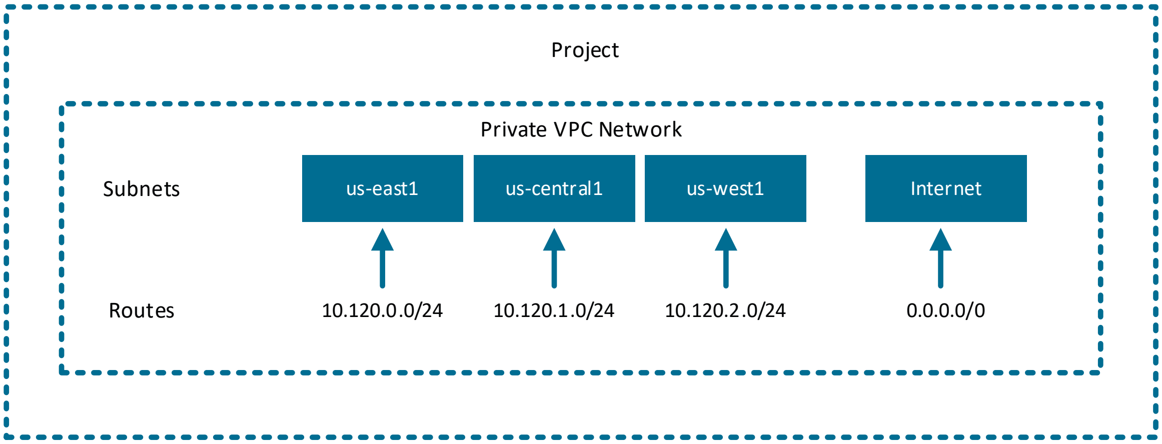
It is important to understand the differences between how GCP forwards traffic in a VPC network and how a traditional Layer 2/Layer 3 network forwards traffic. In traditional networking, when there are two devices that need to communicate, they either communicate directly or through a switch or router depending on whether they are on the same IP subnet. In GCP, all network communication happens at Layer 3. Communication happens at Layer 3 because GCP forwards traffic by using host routes, even when the instances are part of a subnet that has a smaller subnet mask, such as /24. The reduced mask implies that all instances within the VPC network must communicate via a router. However, the intra-VPC traffic never transits the router. Instead, the router responds with a proxy address resolution protocol that tells the device how to communicate directly with the destination device.

By default, all instances connected to the VPC network communicate directly, even when they are part of different subnets. When you add a subnet, GCP automatically generates routes that facilitate communication within the VPC network as well as to the internet. These routes are known as *system-generated* routes. A system-generated route is always the route chosen when the destination is part of a VPC network. Although you can create custom routes within the VPC network, they cannot be equal to or more specific than the system-generated subnet routes.

You can use custom static routes in place of the default route that defines the path out of the VPC network or for subnets that are not part of the VPC. If there are multiple routes to a destination with the same prefix and length, GCP uses route priority to choose a route, where the lower the value, the higher the priority. GCP prefers higher priority routes, and if more than one route has the highest priority, GCP load shares traffic between the routes.

The effect of GCP's traffic forwarding capabilities is that a resource, such as a firewall, can't be inserted in the middle of intra-VPC traffic. However, you can insert one or more firewalls at the ingress/egress of the VPC network.

Figure 3 GCP networking default behavior



Firewall Rules

GCP firewall rules can filter traffic in and out of virtual machine instances. A GCP firewall rule is part of the VPC network configuration and applies to either inbound or outbound traffic. Firewall rules are *stateful*, meaning that they permit return traffic associated with permitted inbound or outbound traffic rules. You can associate GCP firewall rules to all instances in a VPC network, instances with specific tags, or instances with specific services accounts. By default, the VPC networks have two implied rules: one that denies all inbound traffic and one that permits all outbound traffic. You cannot remove these default rules. The effect of this default is to permit all network traffic originating from your instance, along with its associated return traffic, and to permit no traffic inbound to the instance. To change the default behavior for inbound traffic, you create rules that permit traffic, and for outbound traffic, you create rules that deny traffic.

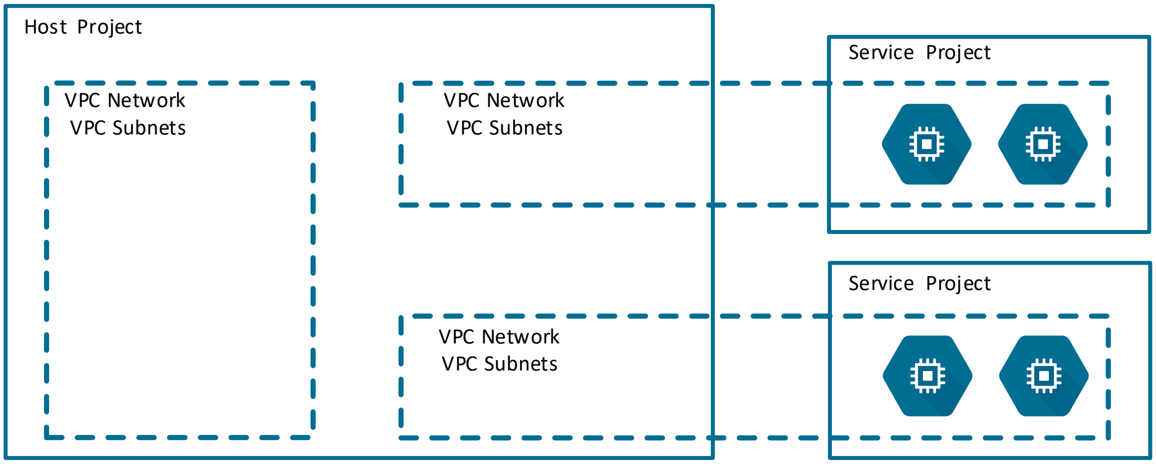
You define and match firewall rules by the traffic source, destination, port, and protocol. Depending on the direction of the rule, the virtual machine instance to which the rule is applied defines either the source or destination. For inbound rules, the destination is the instance, and for outbound rules, the source is the instance. In addition to IP addressing, you can set the source of an ingress rule through GCP tags.

GCP applies firewall rules based on their priority. Priority is part of the rule definition, where the lower the value, the higher the priority. GCP applies the highest priority rule that matches the traffic and ignores any lower-priority rules.

Shared VPC

Shared VPC allows one project to share its VPC networks with one or more projects. Shared VPC is useful in situations where you want one set of administrators to control the networks, traffic forwarding, and security into and out of an application project, and a second set of administrators to control the application project resources.

Figure 4 Shared VPC



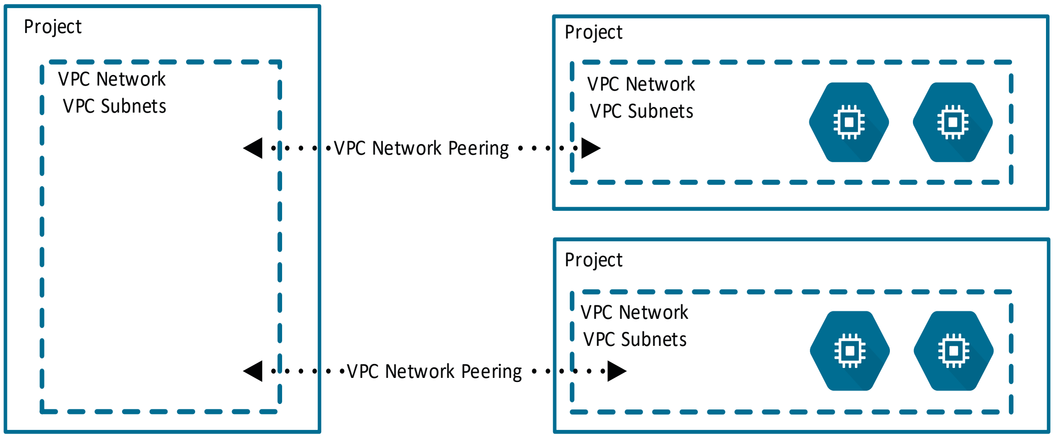
When you use the GCP Shared VPC capability, you designate a project as a *host project*. The host project shares its VPC networks with one or more *service projects*. You can choose the VPC networks that the host project shares with the service projects, from sharing everything to specific subnets. You share the VPCs from a host project with a service project by attaching the service projects to the host project and sharing the VPC networks with the administrators of the service project. GCP uses IAM roles to define which users can administer the host project and its VPC networks and which can administer the application resources in the service projects.

Beyond the separation of network administration and application administration, Shared VPC is an efficient means for different projects to communicate as well as share load balancers and network security infrastructure. If multiple service projects have access to the same Shared VPC network, then they can communicate directly. If each service project uses a unique Shared VPC network, then they can't directly communicate but can share resources deployed in the host project, such as a network firewall that might allow communication through it.

VPC Network Peering

VPC Network Peering allows connectivity across two VPC networks regardless of whether they belong to the same project or the same organization. The traffic stays within the Google Cloud Platform global network and doesn't traverse the public internet. If you have multiple network administrative domains within your organization, VPC Network Peering allows you to make services available across all the domains. If you offer services to other organizations, VPC Network Peering allows you to make those services available to those organizations. The ability to offer services across organizations is useful within your own enterprise if you have several distinct organization nodes due to your own structure or because of mergers or acquisitions, and it is useful if you want to offer services to other enterprises.

Figure 5 VPC Network Peering



Peered VPC networks remain separate administratively. Each project administrator configures routes, firewalls, VPNs, and other traffic management tools separately in each of the VPC networks. You set up each side of a VPC Network Peering association independently. Peering is active and operational only when the configuration from both sides matches. At the time of peering, GCP checks to see if there are any subnets with overlapping IP ranges between the two VPC networks or in any of their other peered networks. If there is an overlap, GCP does not establish a peering relationship, as this would cause routing issues. When you use VPC Network Peering, either project administrator can choose to remove the peering association at any time.

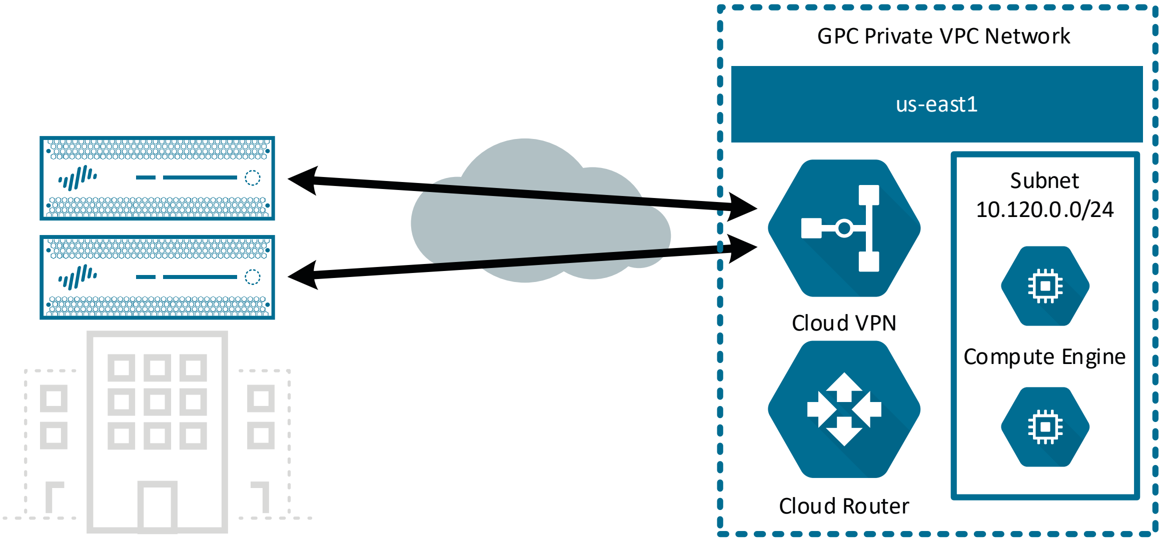
VPC network peers exchange all VPC subnet routes by default, and there is no capability to filter these routes. You can also exchange custom routes, such as static and dynamic routes, if you have configured the peering setup to import or export them. VPC network peers learn routing information dynamically from their peered networks. For example, if a custom route in a peered VPC network changes, your VPC network automatically receives and uses the updated custom route without requiring any action from you.

On-Premises Network Connectivity

The Google Cloud Interconnect and Cloud VPN provide connectivity between a GCP VPC network and your on-premises networks and data centers. Cloud VPN provides connectivity through IPSec VPN, and Cloud Interconnect provides connectivity through either a dedicated private connection or through a service provider.

A Cloud VPN gateway coupled with a Cloud Router enables static or dynamic IP routing between GCP and the on-premises network. For increased connection resiliency, Cloud VPN supports configurations with multiple tunnels to a single location for deployments with resilient on-premises VPN devices. Active/active configuration is also possible if you deploy multiple Cloud VPN gateways.

Figure 6 Cloud VPN gateway



Connections through Cloud Interconnect do not go over the public internet. Instead, you can access GCP instances directly through colocation facilities or dedicated connections. Google recommends Cloud Interconnect connections for all enterprise customer connectivity and to support a range of bandwidth options from 50 Mbps to 10 Gbps.

RESILIENCY CONSTRUCTS

Regions and Zones

GCP consists of a set of physical assets that are contained in Google's data centers around the globe. Each data center location is in a global region. To ensure that maintenance and failures within GCP do not affect the availability of an application or service, you can place instances used for load-sharing and resiliency into different regions and zones. GCP *regions* are separate geographical locations and have one or more zones. Zone names consist of a letter identifier combined with the name of the region. For example, us-central1-a identifies zone a in the US Central region. Zones within a region do not share power, physical switching, or other GCP data center infrastructure. Deploying resilient instances in multiple zones limits the number of instances that a hardware or software failure in GCP can affect. This distribution of resources provides several benefits, including redundancy and reduced latency, by locating resources close to clients.

Note

You can configure a region and zone on an instance only during its initial deployment.
You can't modify the region or the zone after deployment.

Load Balancing

GCP offers several load balancers that distribute traffic to a set of instances based on the type of traffic.

Network Load Balancer

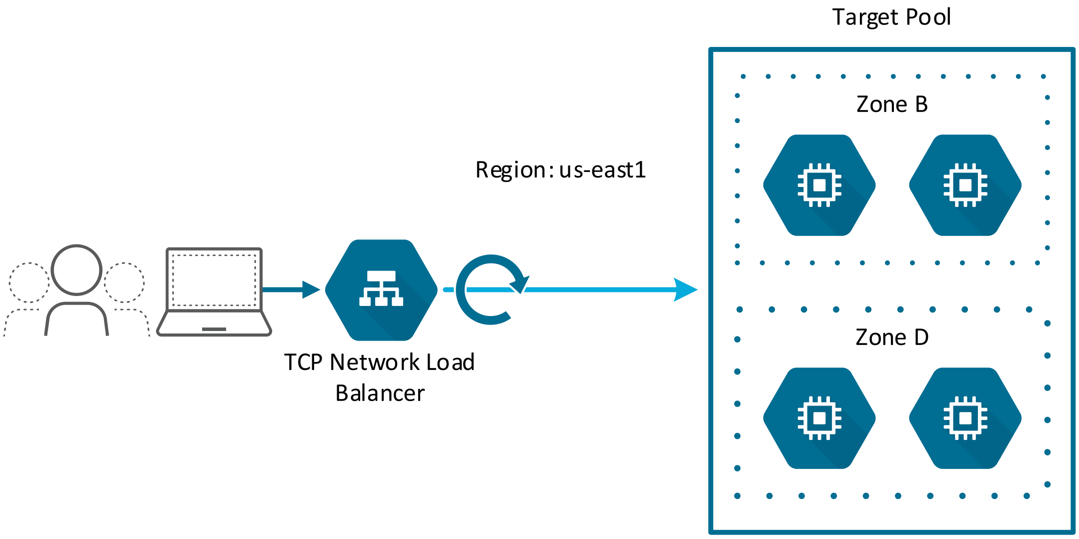
A network load balancer distributes TCP or UDP flows that arrive on the load balancer's frontend to target pool instances in the backend, allowing you to scale applications and provide high availability for services. The network load balancer is a regional network service.

The network load balancer distributes traffic based on TCP/UDP information. It listens on one or more frontend IP addresses. You configure rules defined by a protocol and port number to distribute traffic to healthy instances in a target pool. The load-balancing algorithm uses a 5-tuple hash consisting of source IP address and port number, destination IP address and port number, and protocol.

The load balancer comes in two types: internal and network. The difference between the two types is the source of the traffic. The internal load balancer only supports traffic originating from within the VPC network or coming across a VPN terminating within GCP. The network load balancer is reachable from any device on the internet. Network load balancers are not members of VPC networks. Instead, like public IP addresses attached directly to an instance, GCP translates inbound traffic to the public frontend IP address directly to the private IP address of the instance. Because the public network load balancer is not attached to a specific VPC network, any instance in the project that is in the region can be part of the target pool (regardless of the VPC network) to which the backend instance is attached.

The backend target pools of the network load balancer are composed of instances within the GCP region. For highest availability, the instances should be in different zones. After the network load balancer picks a backend instance from the target pool, the network load balancer sends the traffic directly to the instance. The network load balancer does not translate the destination IP address to the IP address of the backend instance. Instead, it sends the traffic to the backend instance with the original destination. The backend instance can use both the IP and the port to differentiate between applications, allowing port reuse. This configuration requires that the backend instances listen for the frontend IP address of the network load balancer in addition to its own IP address.

Figure 7 Network load balancer



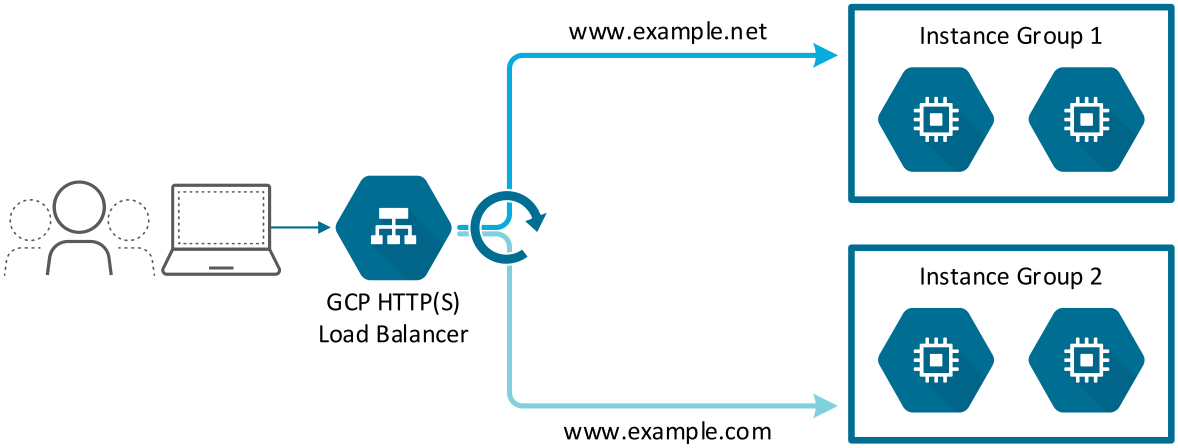
To provide resiliency in case of an instance failure, the network load balancer monitors instances by using HTTP health checks. If the network load balancer does not receive the expected response of HTTP 200, it removes the instance from the target pool. The network load balancer uses HTTP health checks regardless of the TCP/UDP port being load-balanced. Like traffic through the load balancer, health checks sent to backend instances are not destined to the instance IP address, but use the frontend IP address of the network load balancer instead. The network load balancer continues to monitor unhealthy instances so that when they return to a healthy state, they return to the pool of available instances.

HTTP(S) Load Balancer

The HTTP load balancer distributes traffic across multiple instances. You can deploy the HTTP load balancer either as a global load balancer or a region-specific load balancer. When you deploy it as a global load balancer, GCP uses Anycast to advertise the frontend IP address from multiple regions. Traffic enters GCP at the region closet to the client. When you deploy it as a regional load balancer, GCP advertises the frontend IP address from a single region.

HTTP load balancers have one or more frontend public IP addresses. Regardless of the number of frontend public IP addresses, using host rules, HTTP load balancers support multiple websites, each with a backend service. Primarily, HTTP load balancers rely on HTTP host headers to differentiate between websites. When the frontend supports HTTPS, the HTTP load balancer also uses server-name indication to distinguish between websites.

Figure 8 HTTP load balancer



Also, for each website, path rules allow you to select a backend service to serve content based on the URL path. For example, the HTTP load balancer can service the URLs www.example.com/images/ and www.example.com/video/ on two different backend services even though it is a single website.

Backend services are composed of one or more backends. A backend is either an instance group or network endpoint group.

- Instance groups—A set of virtual machine instances, either managed or unmanaged. When managed, GCP can automatically scale the instances on demand.
- Network endpoint groups—A set of IP addresses and ports. Network endpoint groups are useful when a single instance supports multiple applications or containers.

Note

All members of an instance group or network endpoint group must be in the same zone.

When deployed globally, the HTTP load balancer chooses a backend based on the region in which the traffic entered GCP. When deployed regionally or when there are multiple backends within a region, the HTTP load balancer picks the first backend that has healthy instances. After the load balancer chooses a backend, it distributes traffic in a round-robin distribution to all of the available instances in the backend.

Traffic from the load balancer to your instances has a source IP address in the ranges of 130.211.0.0/22 and 35.191.0.0/16. The destination IP address is the private IP address of the backend instance.

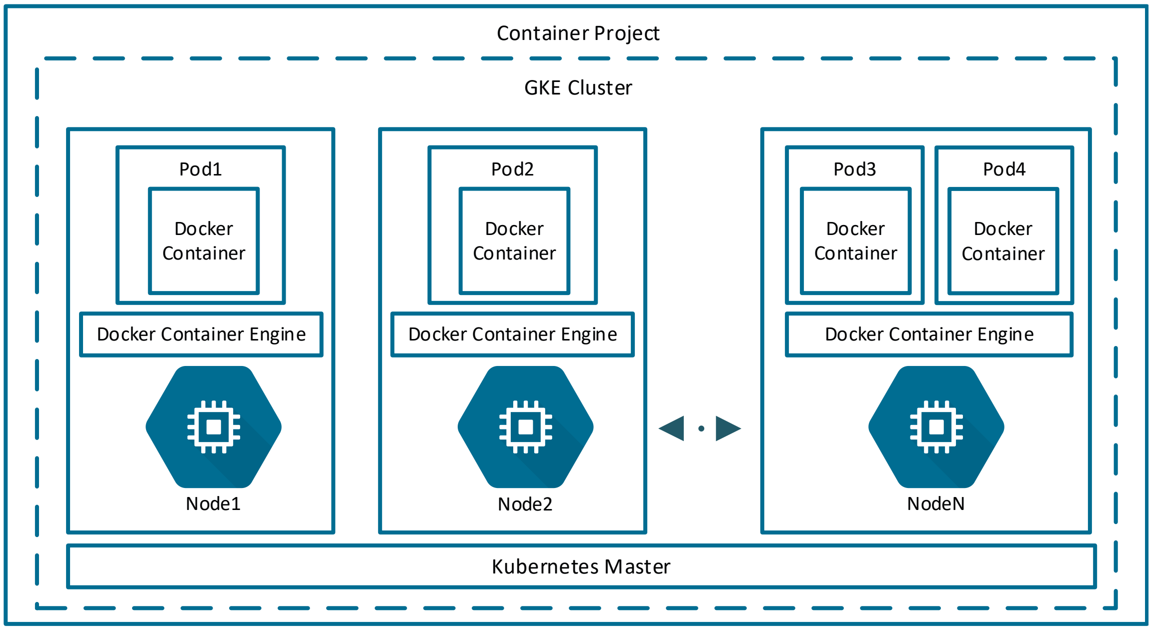
To provide resiliency in case of failure, the HTTP load balancer monitors the health of the instances through HTTP(S) requests. If the HTTP load balancer does not receive the expected response of HTTP 200, GCP removes the instance from the pool of available instances. HTTP load balancer continues to monitor unhealthy instances so that when they return to a healthy state, they return to the pool of available instances.

GOOGLE KUBERNETES ENGINE

GCP offers several products to support specific application deployments. Google Kubernetes Engine (GKE) is a GCP environment for deploying and managing containerized applications. Application containerization is a virtualization method for deploying and running distributed applications without launching an entire virtual machine (VM) for each application. Software developers use GKE to create and test new enterprise applications. GKE eliminates the need to install, manage, and operate your own dedicated Kubernetes clusters because you can easily create them in a GCP project.

GKE is composed of a group of Compute Engine instances that form a cluster. The cluster runs a software package called Kubernetes. *Kubernetes* is an open-source container-orchestration system for automating application deployment, scaling, and management. A Kubernetes master node manages the cluster of Docker containers and runs a Kubernetes API server that communicates with the cluster and performs various tasks, such as servicing API requests and scheduling containers. Every node in the cluster runs a Docker runtime and a Kubernetes node agent (AKA kubelet) to manage the Docker containers.

Figure 9 GKE cluster



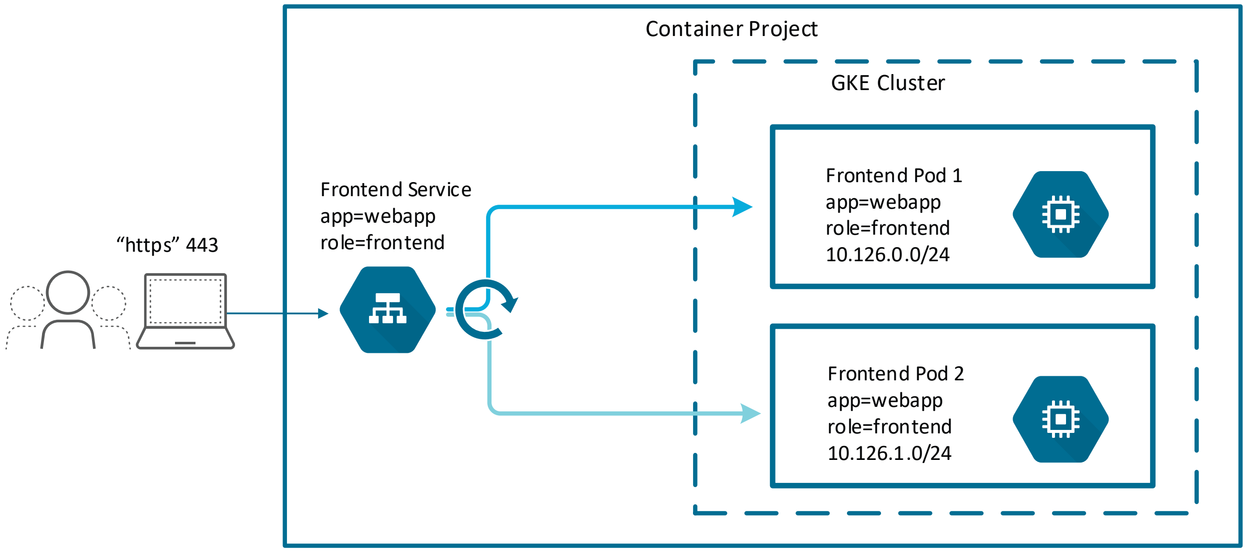
When you deploy a GKE cluster within a project, you get the benefit of some key GCP features. These features include:

- Load-balancing for compute engine instances.
- Node pools to designate subsets of nodes within a cluster.
- Automatic scaling of your cluster's node instance count.

Services

A *service* is a grouping of pods running on the cluster. The idea of a service is to group a set of pods into a single resource. You can have many services running within the GKE cluster. Services provide key features used across the cluster, such as load-balancing, service discovery, and support for zero-downtime application deployments.

Figure 10 GKE service and pods



Pod

A *pod* is a container or group of containers deployed on the same host. Pods operate at one level higher than individual containers because it is common to have a group of containers work together to process a set of work.

Pods typically run a single container but can include more containers in advanced deployments. In this single container per pod model, you can imagine the pod as a wrapper around the container. Kubernetes manages the pods rather than directly managing the containers.

Kubernetes allocates a unique IP address to each pod. Every container in a pod shares the network namespace, including the IP address space and network ports. When containers in a pod communicate with entities outside the pod, they must coordinate how they use the shared network resources. When a pod sends a packet to another pod on the same node, the packet leaves the node and is processed by the GCP network. Next, the packet returns to the same sending node before going to the destination pod.

The GKE default settings configure each node in the cluster to run no more than 110 pods. Kubernetes assigns each node a range of IP addresses so that each pod can have a unique IP address. With the default maximum of 110 pods per node, Kubernetes assigns a /24 classless inter-domain routing block (256 addresses) to each of the nodes in the cluster. By having approximately twice as many available IP addresses as possible pods, Kubernetes can mitigate IP address reuse as you add or remove pods from a node.

Node Pools

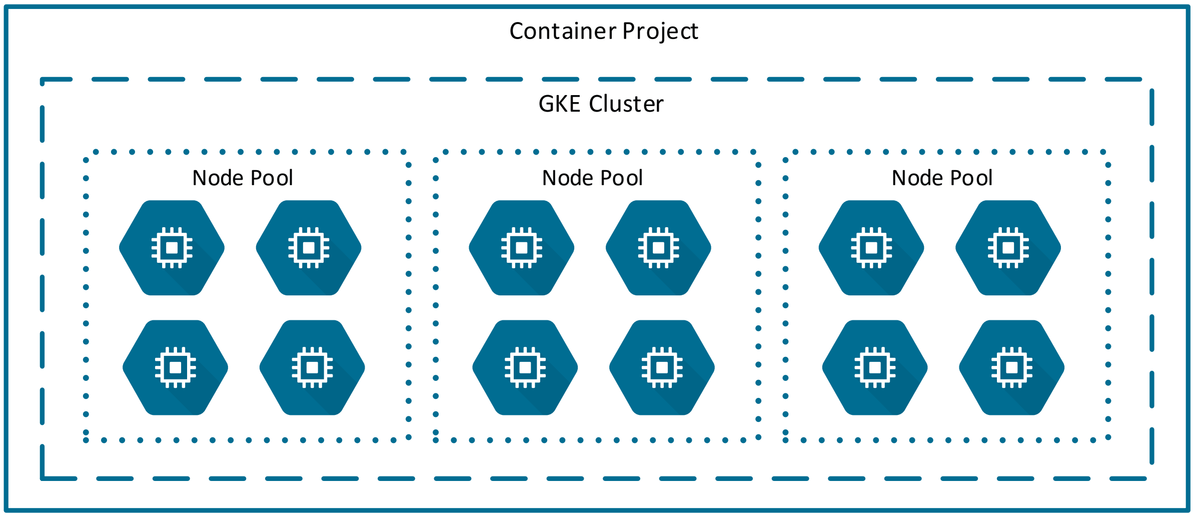
A *node pool* is a group of nodes within a GKE cluster that all have the same configuration. For example, you might create a node pool with a specific instance size or running image. A node pool can vary in size from a single node to many nodes.

When you create a GKE cluster, the number and type of nodes that you specify becomes the default node pool. You can configure the default node pool and add additional node pools or remove node pools in your cluster. Each node in the node pool has a Kubernetes node label associated to it.

When you define a service, you can indirectly control deployment on a specific node pool. The node pool is not dependent on the configuration of the service itself but on the configuration of the pods.

- You can explicitly force a pod to deploy to a specific node pool.
- You can specify resource requests for the containers. The pod only runs on nodes that satisfy the resource requests.

Figure 11 GKE node pools



Palo Alto Networks Design Details

VM-SERIES FIREWALL ON GCP

The Palo Alto Networks VM-Series firewall is the virtualized form factor of our next-generation firewall that can be deployed in a range of private and public-cloud computing environments. VM-Series firewalls protect public-cloud deployments by using application-enablement policies while simultaneously preventing known and unknown threats.

VM-Series Firewall Models

VM-Series firewalls on GCP are available in five primary models: VM-100, VM-300, VM-500, and VM-700. Varying only in capacity, all 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 requirements

	VM-100	VM-300	VM-500	VM-700
Capacities				
Maximum sessions	250,000	800,000	2,000,000	10,000,000
Security rules	1,500	10,000	10,000	20,000
Security zones	40	40	200	200
IPSec VPN tunnels	1000	2000	4000	8000
SSL VPN tunnels	500	2000	6000	12,000
Requirements				
CPU cores (minimum/maximum)	4	4	8	16
Minimum memory	6.5GB	9GB	16GB	56GB
Minimum disk capacity	60GB	60GB	60GB	60GB
Licensing models	BYOL	BYOL/PAYG	BYOL	BYOL

Although the capacity license sets the VM-Series firewalls limits, the size of the virtual machine you deploy the firewall on determines its performance and functional capacity. Table 2 maps the VM-Series firewall to GCP virtual machine size based on VM-Series model requirements for CPU, memory, disk capacity, and network interfaces. When deployed on a virtual machine that provides more CPU than the model supports, VM-Series firewalls do not use the additional CPU cores. Conversely, when you deploy a large VM-Series model on a virtual machine that meets the minimum CPU requirements, it effectively performs the same as a lower model VM-Series.

Table 2 VM-Series mapping to GCP virtual machine sizes

Virtual machine specifications	VM-100	VM-300	VM-500	VM-700
n1-standard-4 *4 interfaces* *4 vCPU* *15GB memory*	Recommended	Recommended	—	—
n1-standard-8 *8 interfaces* *8 vCPU* *30GB memory*	Supported	Supported	Recommended	—
n1-standard-16 *8 interfaces* *16 vCPU* *60GB memory*	Supported	Supported	Supported	Recommended

In smaller VM-Series firewall models, it might seem that a virtual machine size smaller than those listed in Table 2 would be appropriate; however, smaller virtual machine sizes do not have enough network interfaces. GCP provides virtual machines with two, four, or eight network interfaces. GCP virtual machine sizes such as the n1-standard-2 might work if CPU, memory, and disk capacity were the only concern, but they are limited by having only two network interfaces. Because VM-Series firewalls reserve an interface for management functionality, two interface virtual machines are not a viable option. Four interface virtual machines meet the minimum requirement of a management, public, and private interface. You can configure the fourth interface as a security interface in an additional VPC or a network partition like a DMZ.

When deployed on larger GCP virtual machine sizes, all VM-Series firewall models support eight network interfaces. The number of network interfaces is relevant for GCP deployments because you might attach an interface to each VPC network in the deployment. Like a physical deployment, each interface can have a unique security zone allowing the VM-Series to leverage zone information in the security policy.

For the latest detailed information on the throughput of the VM-Series on GCP, see the VM-Series on Google Cloud Platform datasheet. Many factors affect performance, and Palo Alto Networks recommends you test in your environment to ensure the deployment meets your performance and capacity requirements. In general, public-cloud architectures are more efficient when scaling out the number of resources versus scaling up to larger virtual machine sizes.

License Models

You can license VM-Series firewalls on GCP with licenses purchased through the GCP Marketplace or regular Palo Alto Networks channels.

> **Note**
>
> Whichever licensing model you choose 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 VM-Series firewall and migrating the configuration. Migration between an evaluation license, a regular license, and an enterprise license agreement (ELA) is possible because they are all part of the BYOL licensing model.

PAYG

A PAYG license is a *usage-based* or *pay-per-use* license. You can purchase a PAYG license from the GCP Marketplace. Google bills hourly for the GCP PAYG licenses.

With a PAYG license, Google includes licenses for the VM-Series firewalls to be used in GCP once you purchase a PAYG license, so the firewalls are ready for use as soon as you deploy them. You do not need a separate license authorization code for the VM-Series firewalls. When you stop or terminate the firewall in GCP, Google suspends or terminates the usage-based licenses.

PAYG licenses are available in the following bundles:

- Bundle 1—Includes a VM-300 capacity license, Threat Prevention license—intrusion prevention system (IPS), antivirus (AV), and malware prevention—and a premium support entitlement.
- Bundle 2—Includes a VM-300 capacity license, Threat Prevention license—IPS, AV, and malware prevention—GlobalProtect™, WildFire, DNS Security, PAN-DB URL Filtering licenses, and a premium support entitlement.

BYOL and VM-Series ELA

You purchase this license from a partner, reseller, or directly from Palo Alto Networks. VM-Series firewalls support all capacity, support, and subscription licenses in BYOL.

When using your own licenses, you license VM-Series firewalls like a traditionally deployed appliance, and you must apply the license authorization code provided by Palo Alto Networks. 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 end-user license 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, Palo Alto Networks deducts 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 VM-Series 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 GCP

This section describes the interaction between VM-Series firewalls and GCP. It begins with a single firewall deployment and then expands with a discussion of resilient deployments.

There are many ways that you can deploy VM-Series firewalls on GCP. Predefined VM-Series solutions are available through the GCP Marketplace as well as code repositories such as GitHub. The solutions available in the marketplace allow you to define settings including the administrator information, number of interfaces, VPC network, subnet, and virtual machine size. The solutions available in the marketplace default to three interfaces (management, untrusted, and trusted) and allow you to add additional interfaces to the virtual machine, up to the limit of 8 total interfaces.

Note

Regardless of the deployment method, you cannot add additional interfaces to a virtual machine instance after deployment.

Deployment and Management

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 rapid deployment of new resources and the destruction of failed resources. A requirement for achieving rapid resource build-out and tear-down is current and readily available configuration information for the resource to use during initial deployment.

Bootstrapping Deployment

At deployment, VM-Series firewalls have a base software-image and factory-default configuration. You can manually upgrade the software and update the configuration after deploying the virtual machine, or you can use bootstrapping to license (if using BYOL), configure, and update the firewall software at boot time. When the configurations are static across similar resources, the simplest method of achieving this for VM-Series firewalls is to use bootstrapping to configure the firewall's networking and policies during deployment. Bootstrapping speeds up the process of configuring, licensing, and making the firewall operational on the network.

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 GCP, you implement the bootstrap package through a GCP storage bucket that contains directories for configuration, content, license, and software. On the first boot, the VM-Series firewall accesses the file share and uses 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.

Management Interface

By default, the first interface attached to the instance is the firewall's management interface. In most instance templates, this interface has a public IP address and a private IP address in the VPC. The firewall's management interface obtains its private IP address through DHCP. If you configure the management interface with a public internet address, GCP networking translates the private IP address to the public IP address when the traffic leaves the VPC.

Note

Because ephemeral public IP addresses might change when instances change state, use a static public IP address in the management VPC if you are managing the VM-Series firewalls over the internet.

You can use GCP firewall rules to restrict access to the management interface of the VM-Series firewall. You can modify GCP firewall rules even when the VM-Series firewall isn't operational. If you have a VPN connection from the VPC back to your on-premises networks using Cloud Interconnect or a Cloud VPN connection, it might be best to manage the firewall through the private IP address on the management VPC instead of the public IP address. However, consider keeping the public IP address on the management interface to provide a second method of connecting to the firewall in case of a configuration error or a failure of the VPN connection back to your network.

Managing Deployments with Panorama

The best method for ensuring an up-to-date VM-Series firewall configuration is to use Panorama for central management of the firewall policies. Panorama simplifies 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 be on-premises or in the public cloud, a physical appliance or virtual—you can use device groups to provide configuration hierarchy. With device group hierarchy, lower-level groups inherit the policies of the higher-level groups. This inheritance model allows you to configure common rulesets that apply to all firewalls, as well as consistent rulesets that apply to specific firewall deployment locations, such as the public cloud.

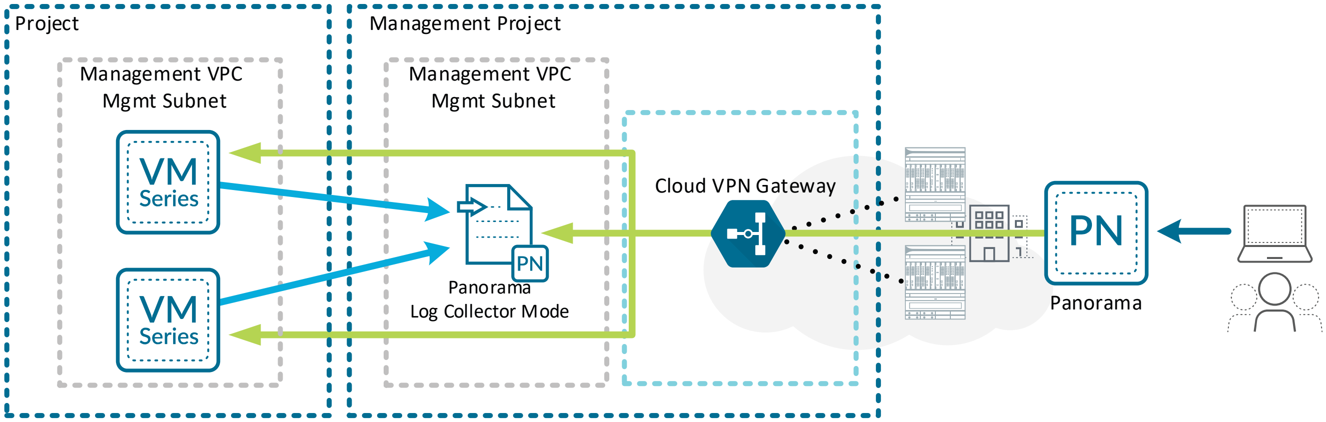
As bootstrapped firewalls deploy, they can also automatically pull configuration information from Panorama. VM-Series firewalls use a VM authorization key and a Panorama IP address in the bootstrap package to authenticate and register to Panorama on its initial boot. You must generate the VM authorization 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 it applies the relevant rulesets, and you can manage the device in Panorama going forward.

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

When you have an existing Panorama deployment on-premises for firewalls in your data center and internet perimeter, you can use Panorama to manage the VM-Series firewalls in GCP. However, sending logging data back to the on-premises Panorama can be inefficient and costly, and it can pose data privacy and residency issues in some regions. An alternative to sending the logging data back to your on-premises Panorama instance is to deploy Panorama-dedicated log collectors on GCP and use the on-premises Panorama for management.

Deploying a dedicated log collector on GCP reduces the amount of logging data that leaves the cloud but still allows your on-premises Panorama to manage the VM-Series firewalls in GCP and have full visibility to the logs as needed.

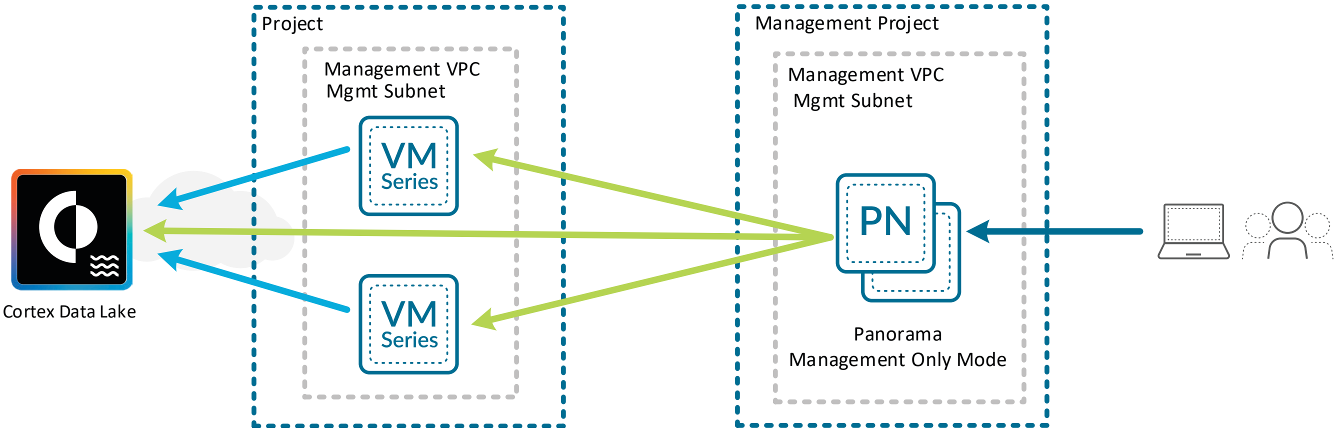
Figure 12 Panorama log collector mode on GCP



There are two design options when deploying Panorama management on GCP. First, you can use Panorama for management only and use Palo Alto Networks Cortex™ Data Lake (formerly *logging service*) 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 their logs and send them to Cortex Data Lake over TLS/SSL connections. Cortex Data Lake allows you to scale your logging storage as your GCP deployment scales, because Cortex Data Lake bases licensing on storage capacity and not the number of devices sending log data.

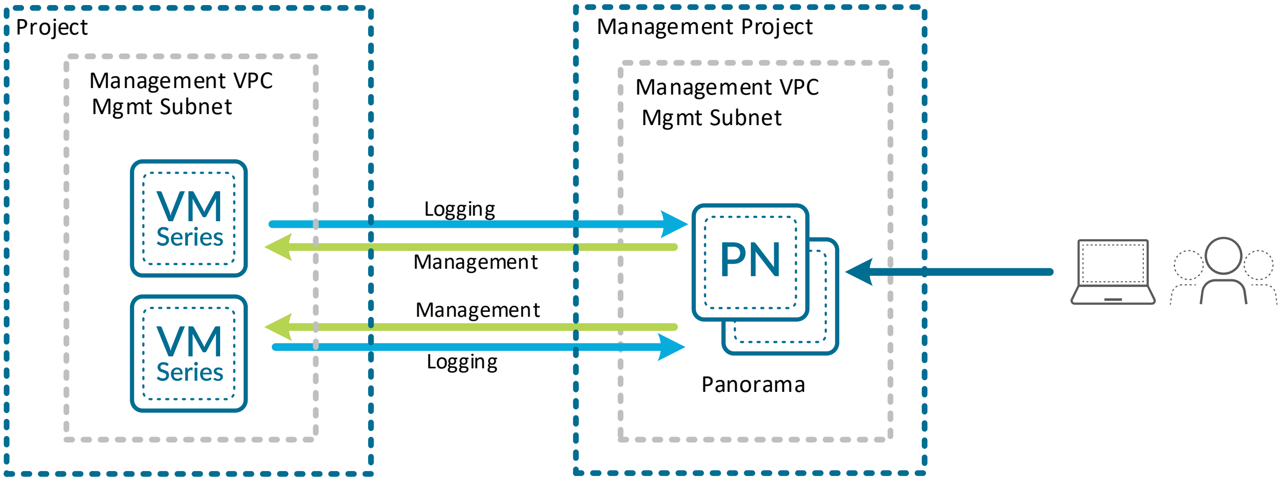
The benefit of using Cortex Data Lake goes well beyond scale and convenience when tied into Cortex. Cortex is a scalable ecosystem of security applications that can apply advanced analytics in concert with Palo Alto Networks enforcement points to prevent the most advanced attacks. Palo Alto Networks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

Figure 13 Panorama management-only mode and Cortex Data Lake



Second, you can use Panorama for both management and log collection. Panorama on GCP supports high-availability deployment if both virtual appliances are in the same VPC network. 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 for each VPC network allows traffic to stay within the VPC and reduce outbound data transfers.

Figure 14 Panorama management and log collection on GCP



Panorama is available as a virtual appliance for deployment on GCP and supports Log Collector mode, Management Only mode, and Panorama mode with the system requirements defined in Table 3. Panorama on GCP is only available with a BYOL licensing model.

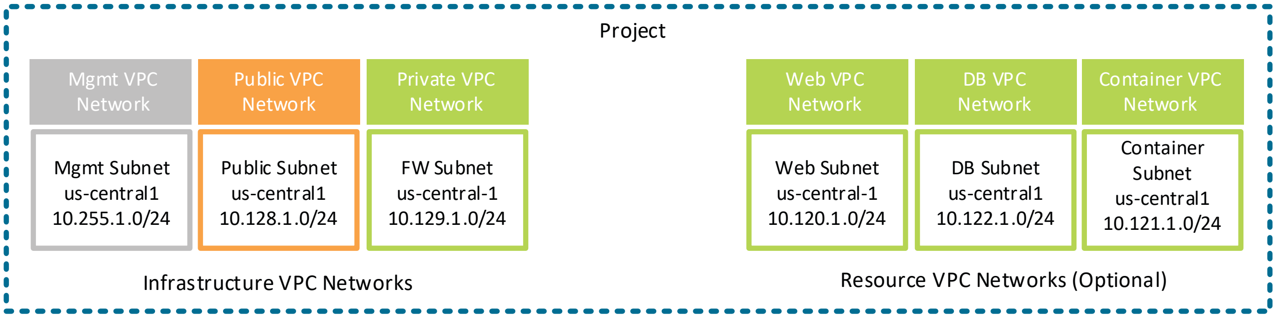
Table 3 Panorama virtual appliance on GCP

	Log collector	Management only	Panorama
Minimum system requirements	16 CPUs 32 GB memory 2TB to 24 TB log storage capacity	4 CPUs 8 GB memory 81 GB system disk	8 CPUs 32 GB memory 2TB to 24TB log storage capacity
GCP sizing	n1-standard-16	n1-standard-4	n1-standard-8

VPC Networking

Although a VPC network supports multiple subnets, by creating separate VPCs for public, management, and private resources, you simplify the configuration for traffic forwarding and security policy. At the most basic, consider using three subnet ranges: one each for the management network, public network, and private network.

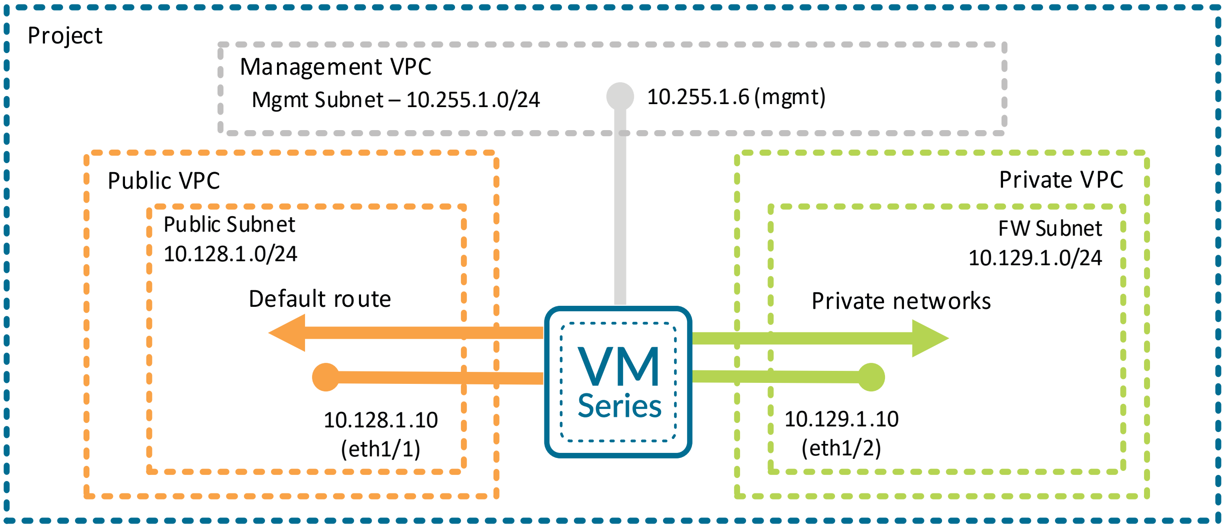
Figure 15 VPC network IP address ranges and subnets


Project
Mgmt VPC Network
Public VPC Network
Private VPC Network
Web VPC Network
DB VPC Network
Container VPC Network
Mgmt Subnet us-central1 10.255.1.0/24
Public Subnet us-central1 10.128.1.0/24
FW Subnet us-central-1 10.129.1.0/24
Web Subnet us-central-1 10.120.1.0/24
DB Subnet us-central1 10.122.1.0/24
Container Subnet us-central1 10.121.1.0/24
Infrastructure VPC Networks
Resource VPC Networks (Optional)

Although VM-Series firewalls support a range of interface deployment options, such as virtual wire, Layer 2, and Layer 3 on GCP, the firewall interfaces are always configurated as Layer 3 because of GCP's networking requirements.

In a Layer 3 deployment, you must assign each interface an IP address. In GCP, you configure VM-Series firewall interfaces to obtain their IP address through DHCP. By default, when a VM-Series firewall interface obtains a default gateway from DHCP, it installs a default route. To ensure proper traffic flow, you should modify the firewall configuration so that it does not obtain default routes through DHCP on the private interface(s). To allow the firewall to reach instances and services within the private VPC networks, set up static routes to the private networks on the firewall's private interface. GCP reserves the first address in the subnet (Example: .1 in a /24) as the subnet's default router address.

Figure 16 Firewall IP routing



GCP Firewall Rules

GCP firewalls rules block inbound traffic to an instance by default, so you must create GCP firewall rules that allow traffic into the VM-Series instance. You define GCP firewall rules at the VPC network level that apply to the instances deployed in the VPC. To simplify the GCP firewall rules, you can assign a network tag to the VM-Series instances and then target that tag when creating the GCP firewall rules. Create GCP firewall rules that:

- Allow TCP/22 and 443 traffic into the VM-Series management interface from the public IP address or range from which you are managing the firewall.
- Allow all traffic into the VM-Series management interface from the private IP range where you deployed Panorama.
- Allow application traffic into the VM-Series instance on the public VPC network from all or a select set of sources. The GCP firewall rule can be broad, allowing all traffic or allowing specific ports and protocols, such as TCP/80 and 443.
- Allow traffic into the VM-Series instance on the private VPC network from all IP subnets in the private VPCs.

Custom Routing

Custom default routes created in the private VPC network direct traffic to a VM-Series instance or an internal network load balancer. The VM-Series instance or the internal network load balancer then forwards the traffic to the available VM-Series firewalls in the backend group. This custom routing replaces the default behavior of traffic flowing from a VPC directly to the internet by using the default route in the VPC. GCP custom routes cannot direct traffic between instances in the same VPC to the VM-Series firewall. Instances in the same VPC network can always communicate directly if you have configured the GCP firewall rules to allow this type of communication.

Because the public, private, and management networks are in separate VPC networks, they are isolated and cannot communicate with each other by default. You do not need to define routes or adjust GCP firewall rules to control traffic between these networks because the VM-Series next-generation firewalls direct and control the traffic flows.

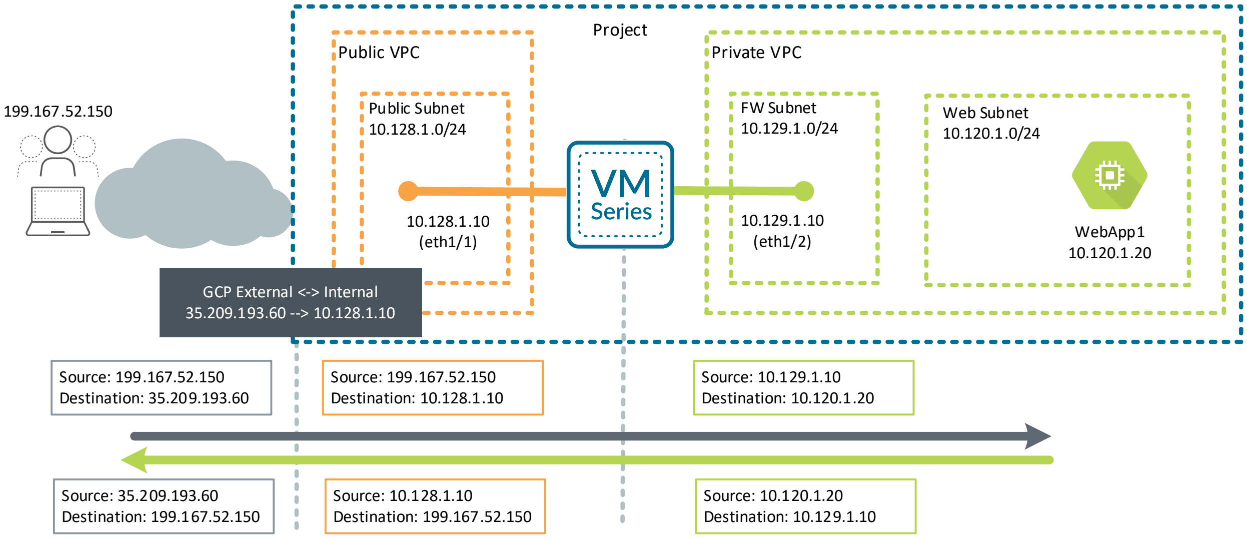
Traffic Flows

Inbound Internet Traffic

To allow a client on the internet to communicate with an instance or application behind the VM-Series firewall, you must associate a public IP address with the instance. Although it is possible to associate public IP addresses directly to instances in the private VPC network, you must associate the public IP address with the VM-Series firewall in order to protect the private instances. The VM-Series firewall then translates the destination IP address to the appropriate private instance.

Because GCP networking translates the destination IP address from the public to the private IP address when the traffic enters the VPC, you must use the private IP addresses of the public interface in the VM-Series firewall's security and NAT policies. You can associate one public IP address to each VM-Series interface, which means the supporting applications that use the same port and protocol require port translation.

Figure 17 Inbound IP address translation

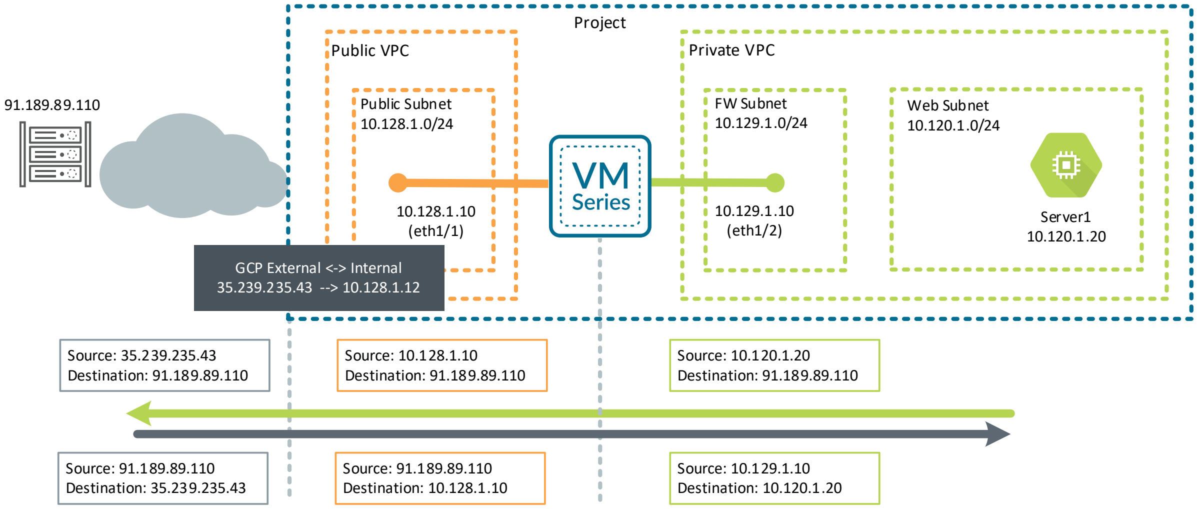


Outbound Internet Traffic

Traffic that originates from an instance on a private VPC network and is destined to the internet, routes to the firewall through a custom default route in the private VPC network. GCP firewall rules block all inbound traffic by default, even intra-VPC traffic, so you must apply a GCP firewall rule to the VM-Series private interface that allows inbound traffic.

For instances behind the VM-Series firewall to communicate to devices on the internet, the public interface on the VM-Series instance must have a public IP assigned in the GCP console. When the VM-Series firewall receives traffic destined to the internet, it translates the source IP address of the outbound traffic to the private IP address assigned to its public interface. GCP then translates the source IP address from the private IP address to the public IP address as the outbound traffic leaves the VPC.

Figure 18 Outbound IP address translation



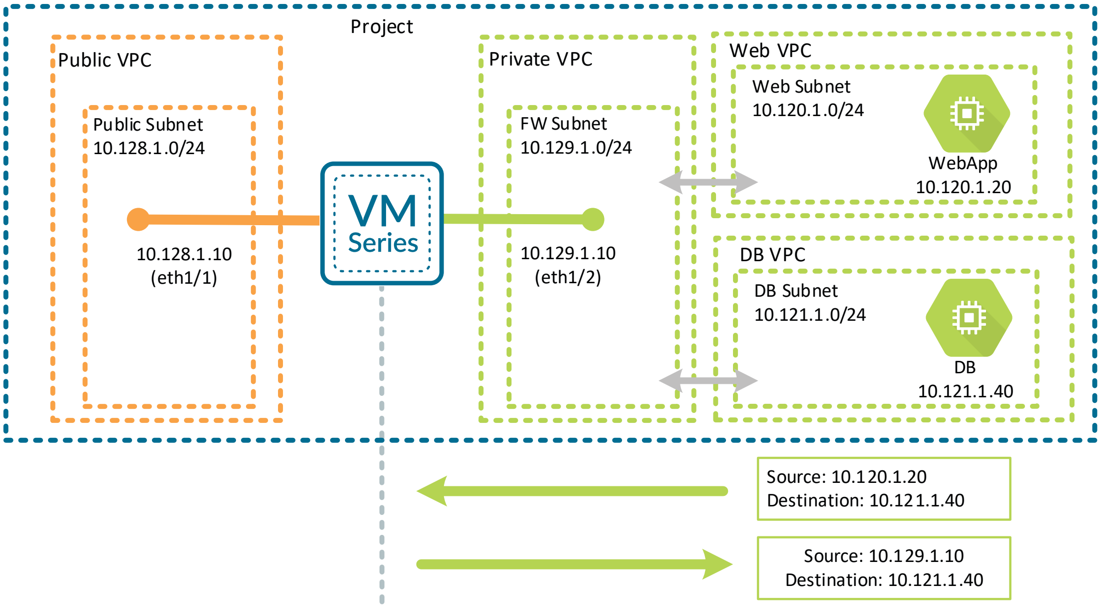
East-West Traffic within a VPC

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 GCP custom routes. You can use the GCP firewall rules 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 custom default route.

Figure 19 East-West IP address translation



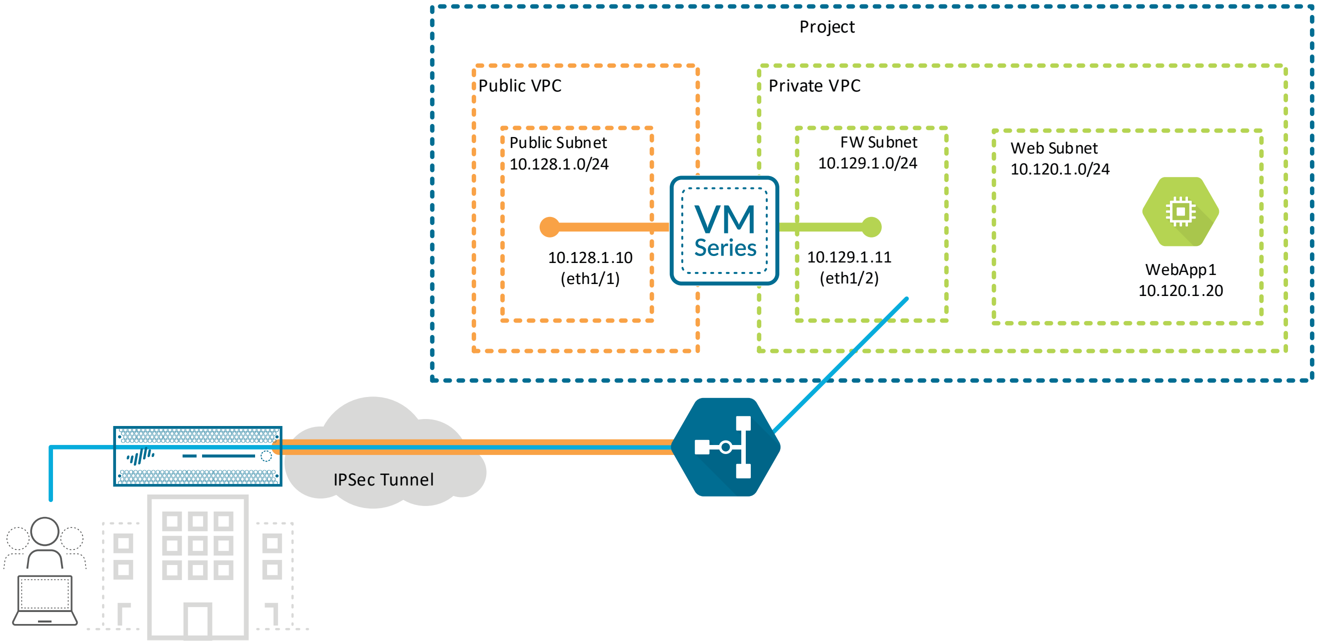
GCP to On-Premises Networks Using Cloud VPN Gateway

When migrating workloads to GCP, it is typically more convenient to create direct connectivity between servers in your data center to the private IP addresses of your instances in GCP. Direct connectivity also helps the network and system administrators to reach instances that do not have public IP access. You should still control traffic between the on-premises connections and the instances in GCP. Protect at least one end of the connection with firewalls.

To access the instances in a VPC, you can either create the VPN directly to/from the VM-Series firewall in the VPC or to a Google Cloud VPN gateway. When creating a VPN connection, you have the option of running a dynamic routing protocol (BGP) over the tunnels or using static routes.

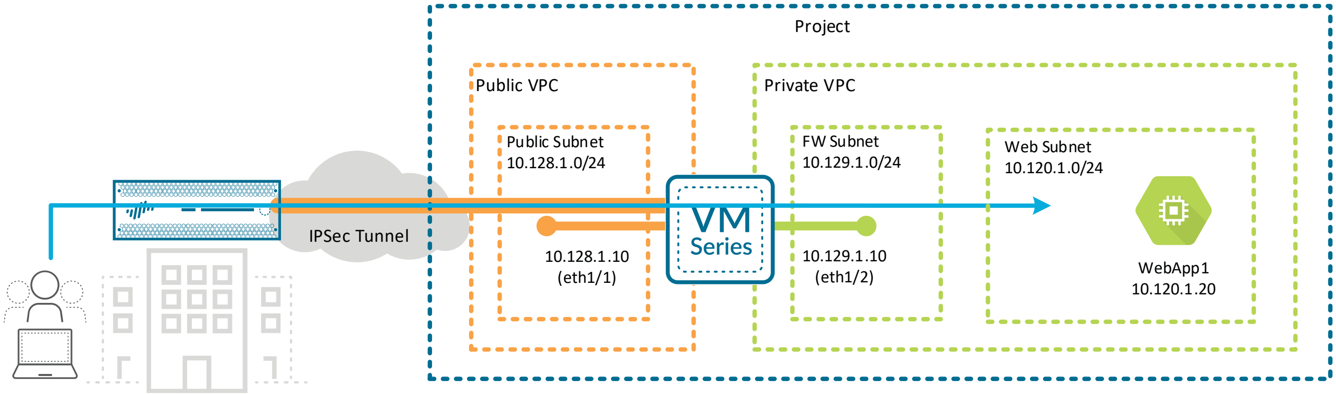
The challenge of using a Cloud VPN gateway is that there is no easy way to force traffic coming into the VPC network through the VM-Series firewall for traffic visibility and protection. Inbound traffic to the VPC follows the VPC route and goes directly to the instance, bypassing the VM-Series firewall. If the VPN gateway at the on-premises location is not a firewall, then you have uncontrolled access from the on-premises network to the instances in the VPC, as shown in Figure 20.

Figure 20 VPN connections with Cloud VPN bypass firewalls



The preferred design uses a VPN connection between your VM-Series firewalls in GCP to an on-premises device. With this design, the VM-Series firewall inspects and controls all inbound VPN traffic as shown in Figure 21. This design covers the scenario where the on-premises VPN peer is not a firewall. You can use static or dynamic routing to advertise reachability information. The instances in the VPC still use the private custom GCP routing and the VM-Series firewalls for their default gateway.

Figure 21 VPN connections to the VM-Series firewall



As environments grow to multiple VPCs, you might consider a dedicated centralized pair of VM-Series firewalls to provide VPN access in and out of the VPCs.

Resiliency

In a traditional high availability deployment, a pair of firewalls shares configuration and state information that allows the second firewall to take over for the first when a failure occurs. Although you can configure high availability so that both firewalls are passing traffic, the firewalls typically operate as an active/passive pair where only one firewall is passing traffic at a time.

You achieve VM-Series resiliency in GCP by using native cloud services. The benefits of configuring resiliency through native public-cloud services include faster failover and the ability to scale out the firewalls as needed. However, in a public-cloud resiliency model, VM-Series 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 provides application resiliency and allows scale out independent of the network infrastructure.

The GCP resources and services used to achieve resiliency for the VM-Series firewall include:

- **Load balancers and target pools**—Distribute traffic across two or more independent VM-Series firewalls in the same GCP region but in different GCP zones. Every VM-Series firewall in the load balancer's backend target pool actively passes traffic, allowing firewall capacity to scale out as required. The load balancer monitors the availability of the web server backend through HTTP health checks and updates the target pool as necessary.
- **HTTP(S) load balancer and instance groups**—Distribute traffic across two or more instance groups, each with one or more independent VM-Series firewalls. The HTTP load balancer monitors the availability of the web server backend through HTTP(S) health checks and updates the backend service as necessary.

Another way that VM-Series firewall resiliency in GCP differs from traditional firewall high availability is that in GCP, you do not implement firewall resiliency at a device level. Instead, you can optionally implement VM-Series firewall resiliency based on the direction of the traffic. For example, it is possible to configure resiliency for inbound traffic from the internet and its return traffic separately from outbound internet, east-west inter-VPC traffic, and backhaul VPN traffic. In fact, the resiliency for inbound traffic can completely differ from both outbound and east-west traffic flows.

Inbound Traffic

You implement resiliency for inbound traffic from the internet by using either a GCP network load balancer or a GCP HTTP load balancer.

Resiliency for Inbound Traffic with GCP Network Load Balancers

GCP network load balancers have one or more public IP addresses configured on the frontend and a target pool associated with the VM-Series firewall instances. You do not need to pick the VM-Series interface or IP address, because the load balancer always sends traffic to the first interface of the VM-Series firewall.

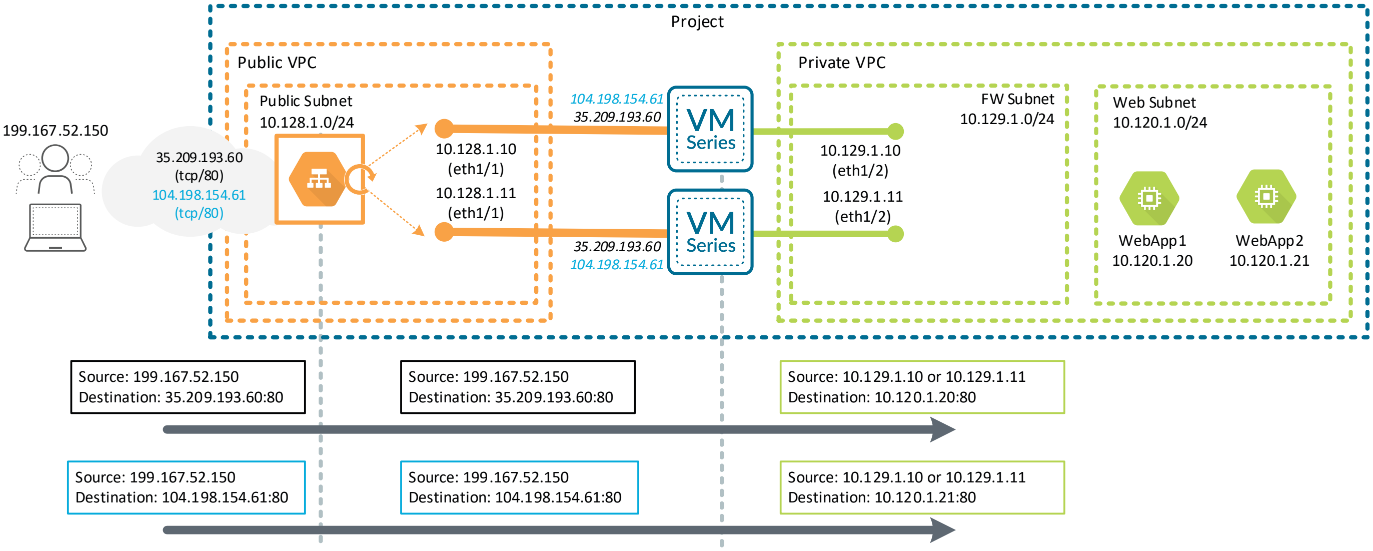
> **Note**
>
> By default, the VM-Series firewall assigns the first interface to the management interface. To support load balancing, you must swap the first and second interfaces on the VM-Series by enabling the **mgmt-interface-swap** metadata field.

Load-balancing rules direct traffic to the VM-Series firewalls based on the destination IP address and TCP or UDP port numbers. The load balancer does not translate the destination IP address before sending the traffic to VM-Series instances in the target pool.

To get the traffic to an instance in the private zone, a NAT policy rule on the VM-Series firewall must translate the destination IP address from the public IP address assigned to the load balancer to the private IP address. Because there is no DNS name associated with the frontend IP address, you must use a static public IP address. To ensure traffic symmetry, or that return traffic from the instance leaves through the same VM-Series firewall that processed the incoming traffic, the VM-Series firewall must also translate the source IP address to the IP address of its private interface. Without this source NAT, default routing selects the path out of the VPC network.

Inbound resiliency with a network load balancer supports multiple applications that use the same destination port number, because the VM-Series firewall can use the unique frontend IP addresses to differentiate applications. This also makes VM-Series firewall policy configuration consistent across firewalls as the IP addresses used in VM-Series policy do not include firewall-specific IP addresses.

Figure 22 Inbound traffic flow with network load balancer



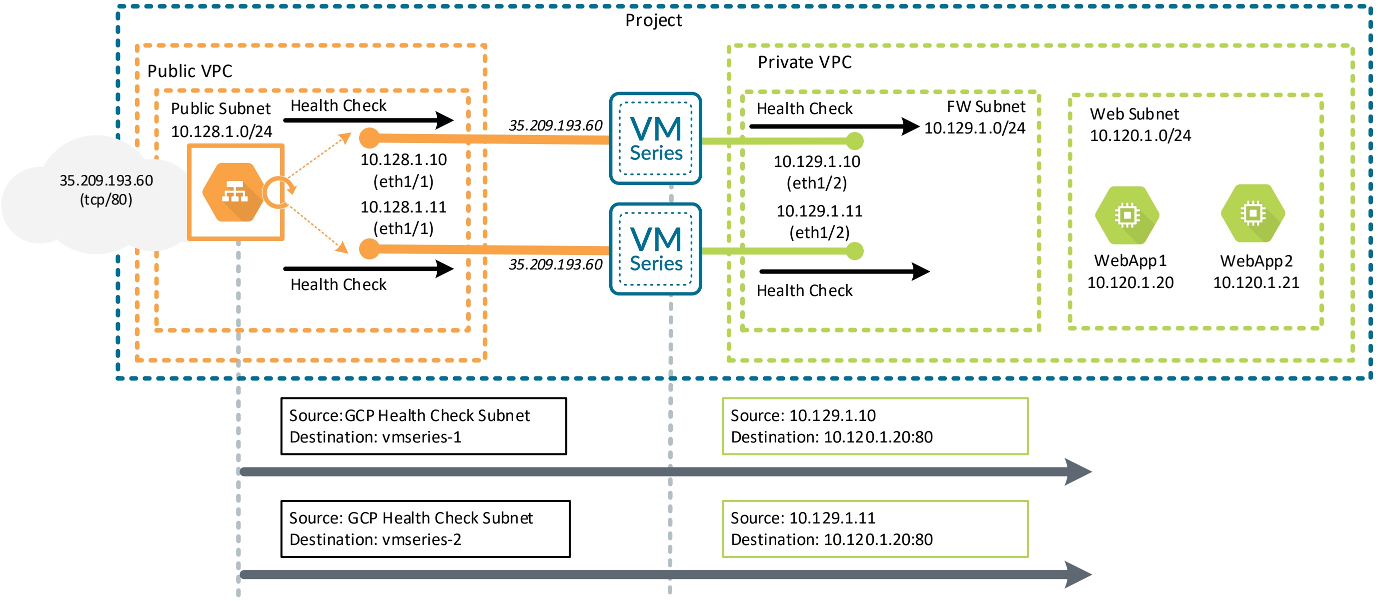
Health checks determine the health of the VM-Series firewalls in the backend pool. The load balancer sends health checks to the instances defined in the target pool, but like regular traffic through the load balancer, the destination IP address is the frontend IP address. This allows the health checks to monitor the full path to the private instances as the VM-Series firewall's NAT and security policies use the public interface IP address.

The health check is always an HTTP check, even when the application is not a web server. In the instances where the backend application is not a web application, you must deploy a web server instance that the health check can query through the VM-Series firewall.

Note

GCP sends HTTP health checks from the IP ranges 209.85.152.0/22, 209.85.204.0/22, and 35.191.0.0/16.

Figure 23 Inbound network load balancer health checks



Resiliency for Inbound Traffic with GCP HTTP(S) Load Balancer

HTTP load balancers have one or more frontends, each configured with a public IP address, protocol (HTTP or HTTPS), and Port (80,8080,443). The HTTP load balancer's backend service is comprised of one or more instance groups associated to the public interfaces of the VM-Series firewalls. GCP requires multiple instance groups when the firewalls are in separate zones, as all the members of an unmanaged instance group must be in the same zone.

The HTTP load balancer is a proxy. By default, a single host and path rule forwards all traffic, regardless of the frontend, to the backend service. The VM-Series firewalls select the backend service based on the host and path rules and does not take into consideration the frontend.

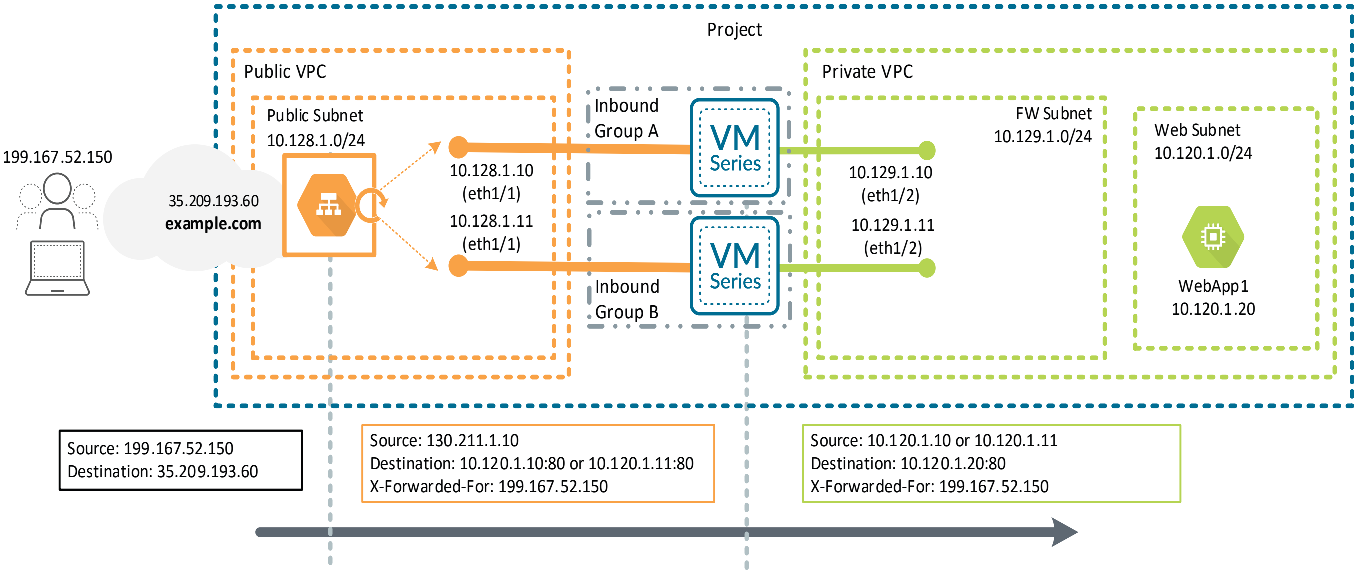
The HTTP load balancer initiates connections to the backend instances sourced from 130.211.0.0/22 and 35.191.0.0/16. 1

Note

The original source IP address of the web client is part of the HTTP packet header in the X-Forwarded-For (XFF) HTTP header field. The VM-Series firewall logs the XFF information, in addition to other session data, to retain information about the original source IP address for each session.

To get the traffic to a private instance, a NAT policy rule on the VM-Series firewall must translate the destination address from the firewall's public interface IP address to the backend instance IP address for traffic sourced from the HTTP load balancer. The private destination might be a server instance or the frontend IP of an internal load balancer. To ensure traffic symmetry, the VM-Series firewall must also translate the source IP address to the IP address of its private interface.

Figure 24 Inbound traffic flow with HTTP(S) load balancer

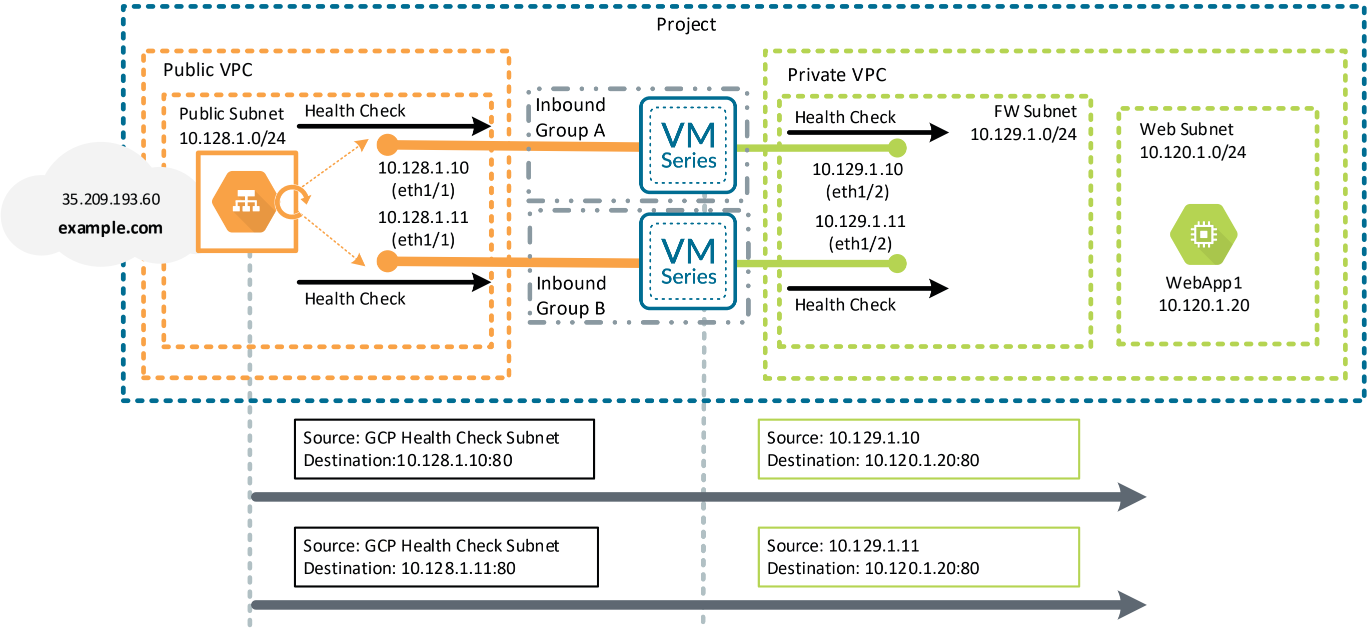


Health checks for the HTTP load balancer determine the health of the actual backend instances. GCP sources the checks from the same subnets as all inbound traffic from the load balancer that is destined to the backend service instances. The same NAT and security policies that permit application traffic permit the checks to pass directly to the private instances. A successful health check verifies that both the VM-Series firewall and the private instances for the application are available.

Note

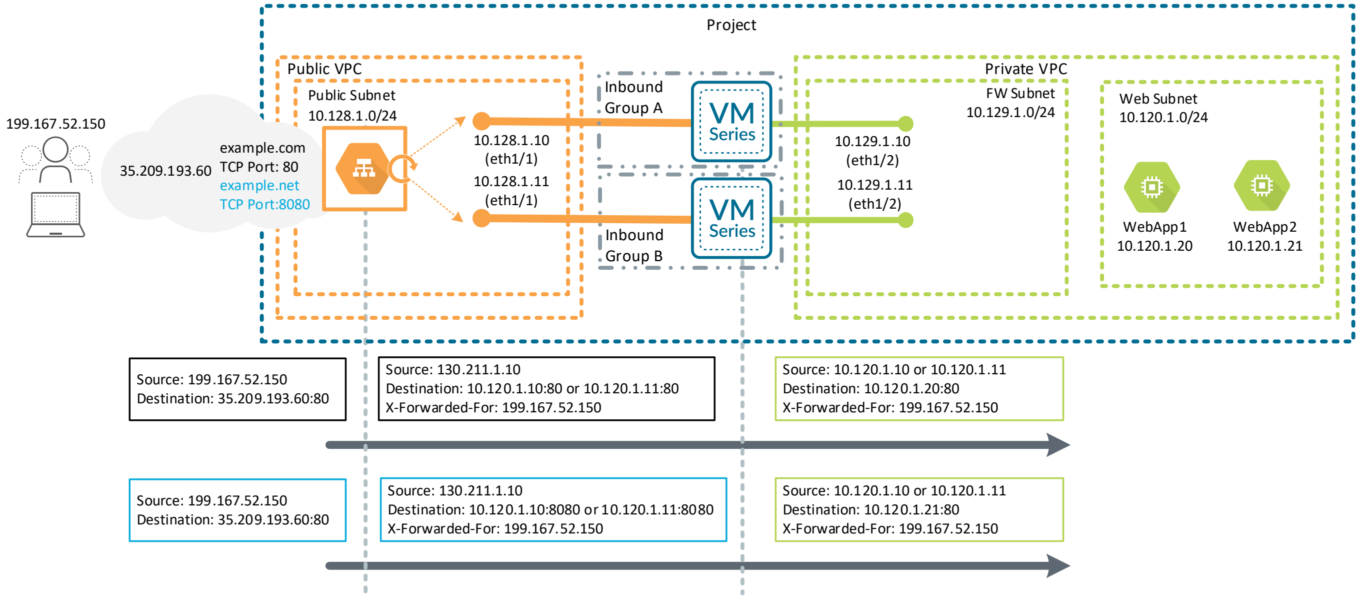
If you configure the VM-Series firewall with a management profile on the public interface for HTTP, HTTPS, SSH, or Telnet, then traffic to TCP/80, TCP/443, TCP/22, or TCP/23 does not pass to the backend instances. Do not use those ports for the backend with the HTTP load balancer. The health checks might succeed if the VM-Series firewall permits load balancer source IP addresses in its management profile, but this does not verify the health of the backend instances.

Figure 25 Inbound HTTP load balancer health checks



Because the backend service for the HTTP load balancer includes only the public interfaces of the VM-Series firewalls, supporting multiple web applications requires the firewalls to serve in multiple backend services. VM-Series firewalls can serve multiple backend services using the destination port as the differentiator between applications. The VM-Series firewalls have a NAT policy that directs incoming traffic to the correct private instance based on the destination port.

Figure 26 Inbound HTTP(S) load balancer with multiple applications



Outbound and East-West Inter-VPC Traffic

For outbound and east-west VM-Series firewall resiliency, use a GCP internal network load balancer to distribute those traffic flows to the outbound and east-west firewall group. The GCP internal network load balancer has a private IP address configured on the frontend and a target pool associated with the outbound and east-west VM-Series firewall instances. The internal load balancer configuration forwards all traffic to the VM-Series in the backend group as opposed to the protocol/port configuration that the inbound network or HTTP load balancers use. You do not need to pick a VM-Series interface or IP address, because the load balancer always sends traffic to the first interface of the VM-Series firewall.

Note

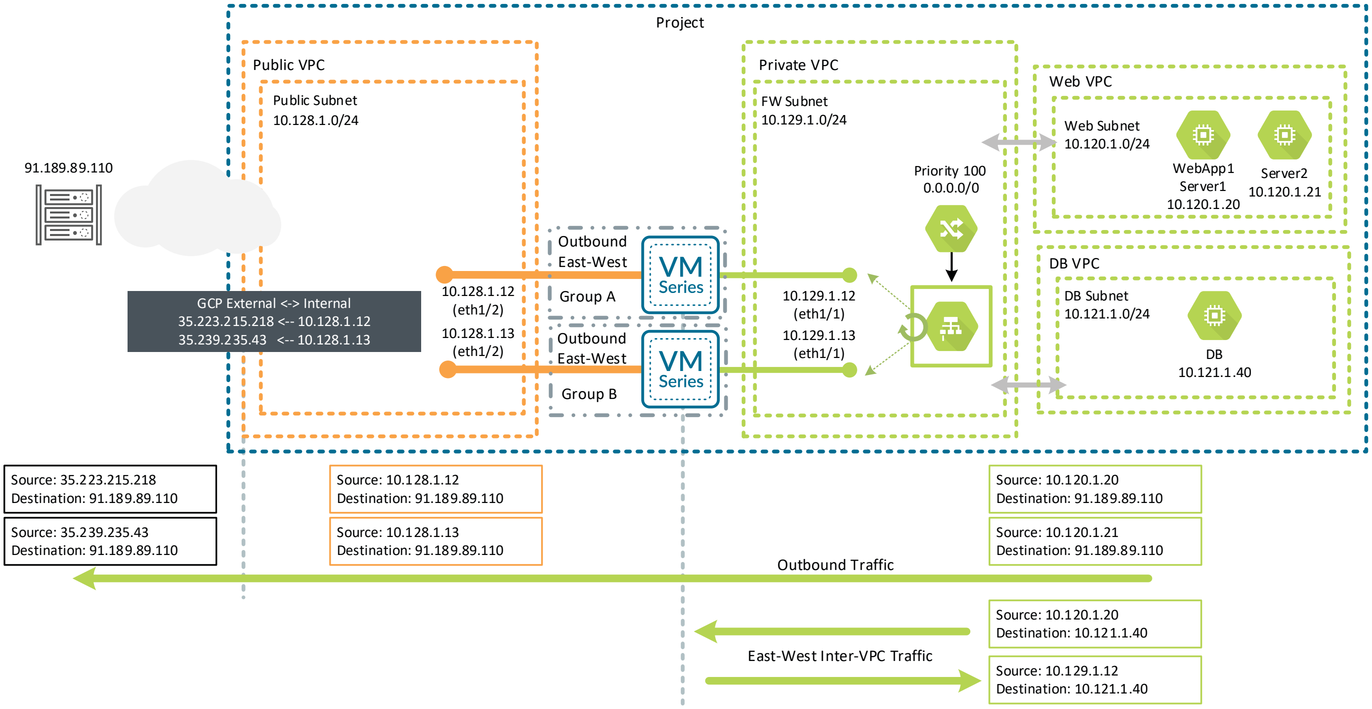
By default, the VM-Series firewall assigns the first interface to the management interface. To support load balancing of outbound and east-west traffic to the VM-Series firewall, you must swap the first and second interfaces. You can swap these interfaces by enabling the mgmt-interface-swap metadata field. To enable a GCP internal network load balancer to send traffic to a VM-Series firewall, you must deploy the firewall with the first interface in the private VPC.

A GCP custom static, default route directs all outbound internet traffic, east-west inter-VPC traffic, and backhaul traffic from on-premises networks to the internal network load balancer frontend. The network load balancer then directs the traffic to the available backend group's VM-Series instances. The load balancer does not translate the source or destination IP address before sending the traffic to the VM-Series instances in the target pool. The network load balancer uses SSH health checks to verify the operational status of the VM-Series firewalls in the backend group. If a health check fails, the network load balancer removes the VM-Series instance from the pool and forwards the traffic to the available VM-Series firewalls.

The security policies on the VM-Series firewall should limit what applications and resources the private instances can access. In most designs, the firewall does not need to translate the destination IP address. For outbound internet and east-west inter-VPC traffic, the firewalls must translate the source IP address to the IP address of the VM-Series firewall's egress interface to ensure traffic symmetry. Without this source NAT, traffic might not return to the same firewall. In the case of outbound internet traffic, the VM-Series firewall that the flow traversed changes the source address to the public interface IP of the VM-Series firewall. For east-west inter-VPC traffic, the VM-Series firewall that the east-west flow traversed changes the source address to the private interface IP of the VM-Series firewall.

In an east-west flow, the traffic enters and exits the same private interface of the firewall. Since this interface is in a single zone, the security policy of the firewall should also include source and destination subnet addresses.

Figure 27 Outbound and East-West traffic—default routes

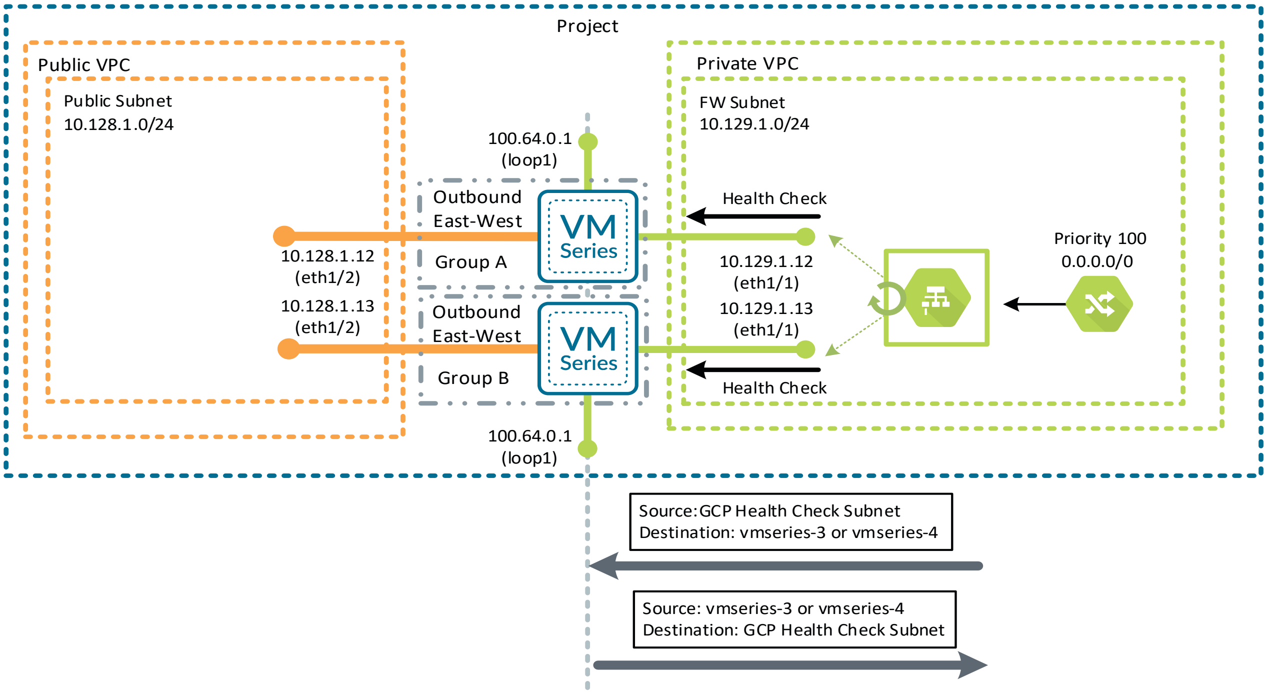


Health checks determine the health of the VM-Series firewalls in the load balancer backend pool. The load balancer sends health checks to the instances defined in the backend configuration. The health check configuration allows for several different protocol options. To ensure the availability of the outbound and east-west VM-Series instances, the internal load balancer sends a simple SSH health check. This setup requires you to configure a loopback interface and a destination NAT policy on the VM-Series firewalls. You apply a management profile to the loopback interface that allows only SSH packets sourced from the GCP health check IP subnets.

The VM-Series firewall receives the health check and the NAT policy changes the destination to the loopback interface. This flow allows the firewall to process the health check and respond back to the initial connection request.

> **Note**
>
> GCP sends health checks to the internal network load balancer from the 130.211.0.0/22 and 35.191.0.0/16 subnets.

Figure 28 Outbound and east-west network load balancer health checks



PRISMA CLOUD FOR GCP

Prisma Cloud is a cloud infrastructure security solution that provides complete visibility and control over risks within your public-cloud infrastructure. This service continuously monitors your cloud environments to help ensure that your cloud infrastructure is protected from security threats.

You can purchase Prisma Cloud directly from the Google Cloud Marketplace. Within 24 hours of purchase, you'll get access to the Prisma Cloud tenant that Palo Alto Networks and Google provisioned for you. Just go to the GCP Marketplace and search for Prisma Cloud.

Prisma Cloud provides cloud infrastructure protection across the following areas:

- Multi-cloud security—Provides a consistent implementation of security best practices across GCP, AWS, and Azure. Prisma Cloud requires no agents, proxies, software, or hardware for deployment and integrates with a variety of threat intelligence feeds. Prisma Cloud includes pre-packaged policies to secure multiple public-cloud environments.
- Continuous compliance—Maintain continuous compliance across CIS, NIST, PCI, FedRAMP, GDPR, ISO, and SOC 2 by monitoring API-connected cloud resources across multiple cloud environments in real time. Prisma Cloud can generate compliance documentation with one-click exportable, fully prepared reports.
- Cloud forensics—Go back in time to the moment a resource was first created and see when every change was made chronologically and by whom. Prisma Cloud provides forensic investigation and auditing capabilities of potentially compromised resources across your GCP environment, as well as other public-cloud environments. Historical information extends back to initial creation of each resource and the detailed change records includes who made each change.
- DevOps and automation—Enable secure DevOps without adding friction by setting architecture standards that provide prescribed policy guardrails. This methodology permits agile development teams to maintain their focus on developing and deploying apps to support business requirements.

Prisma Cloud connects to your cloud via APIs and aggregates raw configuration data, user activities, and network traffic to analyze and produce concise actionable insights.

Prisma Cloud performs a five-stage assessment of your cloud workloads. Contributions from each stage progressively improve the overall security posture for your organization:

- Discovery—Prisma Cloud continuously aggregates configuration, user activity, and network traffic data from disparate cloud APIs. It automatically discovers new workloads as soon as they are created.
- Contextualization—Prisma Cloud correlates the data and applies machine learning to understand the role and behavior of each cloud workload.
- Enrichment—External data sources—such as vulnerability scanners, threat intelligence tools, and SIEMs—further enrich the correlated data to deliver critical insights.
- Risk assessment—Prisma Cloud scores each cloud workload for risk, based on the severity of business risks, policy violations, and anomalous behavior. Aggregated risk scores enable you to benchmark and compare risk postures across different departments and across the entire environment.
- Visualization—An interactive dependency map shows the entire cloud infrastructure environment, providing context beyond the raw data.

Threat Defense

Prisma Cloud enables you to visualize your entire GCP environment, including every component within the environment. The platform dynamically discovers cloud resources and applications by continuously correlating configuration, user activity, and network traffic data. Combining this deep understanding of the GCP environment with data from external sources, such as threat intelligence feeds and vulnerability scanners, enables Prisma Cloud to produce context around risks.

The Prisma Cloud platform includes policies that adhere to industry-standard best practices right out-of-the-box. You can also create custom policies based on your organization's specific needs. The platform continuously monitors for violations of these policies by existing resources as well any new resources that are dynamically created. You can easily report on the compliance posture of your GCP environment to auditors.

Prisma Cloud automatically detects user and entity behavior within the GCP infrastructure and management plane. The platform establishes behavior baselines, and it flags any deviations. The platform computes *risk scores*—similar to credit scores—for every resource, based on the severity of business risks, violations, and anomalies. The risk score helps you to quickly identify the riskiest resources and enables you to quantify your overall security posture.

Prisma Cloud reduces investigation-time from weeks or months to seconds. You can use the platform's graph analytics to quickly pinpoint issues and perform upstream and downstream impact analysis. The platform provides you with a DVR-like capability to view time-serialized activity for any given resource. You can review the history of changes for a resource and better understand the root cause of an incident, past or present.

Prisma Cloud enables you to quickly respond to an issue based on contextual alerts. Alerts are triggered based on a risk-scoring methodology and provide context on all risk factors associated with a resource. This feature makes it simple to prioritize the most important issues first. When a resource has a high risk score, you can choose to send alerts, orchestrate policy, or perform auto-remediation. Prisma Cloud can also send alerts to third-party tools such as Slack, Splunk, ServiceNow, and Demisto® to remediate the issue.

Prisma Cloud provides the following visibility, detection, and response capabilities:

- Host and container security—Configuration monitoring and vulnerable image detection.
- Network security—Real-time network visibility and incident investigations. Suspicious/malicious traffic detection.
- User and credential protection—Account and access key compromise detection. Anomalous insider activity detection. Privileged activity monitoring.
- Configurations and control plane security—Compliance scanning. Storage, snapshots, and image configuration monitoring. Security group and firewall configuration monitoring. IP address management configuration monitoring.

Continuous Monitoring

The dynamic nature of the cloud creates challenges for risk and compliance professionals tasked with measuring and demonstrating adherence to security and privacy controls. With the Prisma Cloud portal, you can view the collected, continuous security-monitoring data collected by Prisma Cloud and verify compliance of your resources to CIS v1.0, CSA CCM v3.0.1, GDPR, HIPAA, ISO 27001:2013, NIST 800.53 R4, PCI DSS v3.2, and SOC2 standards. This capability eliminates the manual component of compliance assessment.

Prisma Cloud provides security and compliance teams with a view into the risks across all their cloud accounts, services, and regions by automating monitoring, inspection, and assessment of your cloud infrastructure services. With real-time visibility into the security posture of your environment, you can identify issues that do not comply with your organization's required controls and settings and send automated alerts.

Scanning API

The Prisma Cloud Scanning API is a public API services designed to help developers, DevOps, and security teams detect and address security issues with their containerized applications as early as possible in their software development lifecycle to reduce the overall attack surface of their applications and potential runtime security issues.

There are two types of services:

- **Prisma Cloud Vulnerability Scan API**—This service identifies vulnerabilities in packages used in container images as well as VMs.
- **Prisma Cloud Infrastructure-as-Code (IaC) Scan API**—This service identifies insecure configurations in common IaC (e.g. HashiCorp Terraform templates and Kubernetes App Deployment YAML files).

Design Models

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

As part of the overall GCP architecture, you use a separate management project to create a centralized management location so that a single Panorama deployment can manage VM-Series firewalls deployed across all of your organization's GCP projects. 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 slightly in how they offer administrative boundaries and networking services. You can combine the design concepts to offer services like load balancing for the inbound applications in one project and common outbound services in another. The design models highlighted as part of this reference architecture are:

- Shared VPC Model—A flexible model suitable for production deployments where separation of administration and resources is necessary. This model distributes compute resources across multiple service projects. The host project administrators create and control the VPC networks to which the service project resources attach. The service project administrators control the compute resources. The addition of VPC Network Peering allows the service project VPC networks to peer with and share routing information with the private VPC network deployed in the host project. This version of the Shared VPC design model differs from previous versions as it does not require you to deploy a VM-Series interface in each service project VPC. The result is a larger scaled hub-and-spoke topology that leverages VM-Series firewall groups to secure inbound and outbound traffic flows and traffic flows between service project VPCs.

- VPC Network Peering Model—A flexible model suitable for production deployments where you do not require separation of administration and resources. In this model, you can deploy compute resources in the same project as the VM-Series firewall or distribute compute resources across multiple projects with unique VPC networks and administrative domains. Each separate project VPC network peers to and shares routing information with the private VPC network deployed in the security project creating a larger scale hub-and-spoke topology leveraging VM-Series firewall groups to secure inbound and outbound traffic flows and traffic flows between project VPCs.

CHOOSING A DESIGN MODEL

Consider which model best fits your requirements and use it as a starting point for your design. When choosing a design model, consider the following factors:

- **Operational Structure**—Beyond the initial implementation, consider the Shared VPC design for a more structured administrative and operational design. If your requirements are to provide security to existing disparate projects, consider using the VPC Network Peering model instead.

- **Administrative Boundaries**—Both models provide inbound, outbound, and east-west traffic control as well as scale and resiliency. The Shared VPC design offers the benefits of a modular and scalable design where network and security resources are centrally managed and shared with the service projects. The VPC Network Peering design centralizes security resource management but requires administrators of peered projects to manage their network resources independently.

SHARED VPC DESIGN MODEL

As the number of projects in your GCP environment grows, you might have a requirement to segment applications from each other for security or administrative reasons, prompting the question of how to best secure all projects. One option is to use a design with dedicated VM-Series firewalls deployed inside each project. With this option, the cost and management complexity of the VM-Series firewalls required grows linearly with the increase in the number of projects. Another option is to centralize the security services by using the Shared VPC design model. This design provides VM-Series security capabilities while minimizing the cost and complexity of deploying VM-Series firewalls in every project.

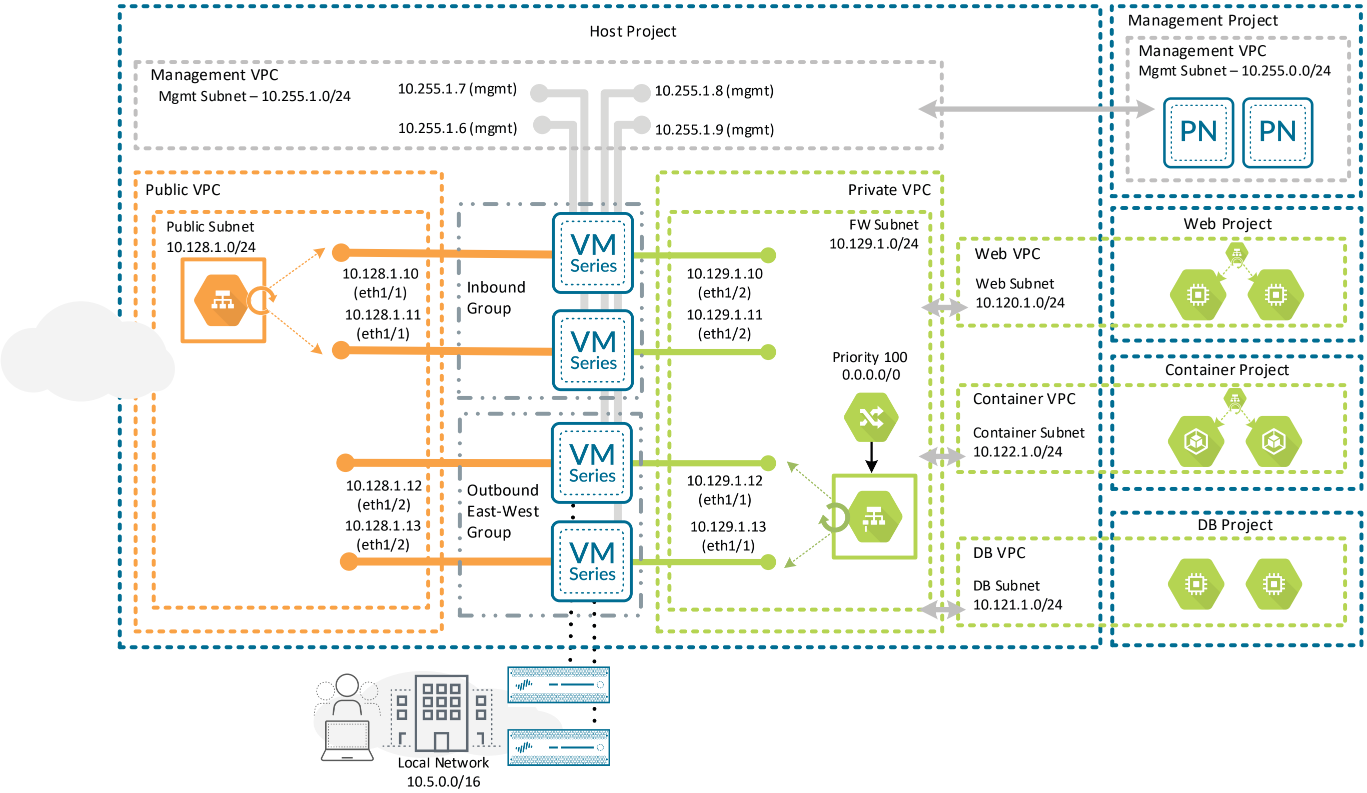
The Shared VPC design provides a common centralized security infrastructure and policy for all traffic flows (inbound, outbound, east-west, and backhaul) for the instances in the service projects. The Shared VPC design uses a hub-and-spoke network architecture consisting of a host project as the hub (which contains the VM-Series firewalls and all the VPCs) and service projects as spokes.

The Shared VPC design easily accommodates organizations that want to distribute project ownership across different business functions (engineering, sales, marketing, etc.) while maintaining common infrastructure and security policies. Within a service project, the business or DevOps teams can move rapidly and independently. When traffic needs to leave the service project VPC, you can apply common or per-VPC policies in the Shared VPC.

To provide VM-Series security to the service projects, you create a VPC network per service project in the host project and share them to the service project owners. In the host project, a group of VM-Series firewalls secure inbound traffic flows. You must deploy the VM-Series firewalls in the same GCP region, but in separate zones, to avoid downtime caused by GCP infrastructure maintenance or failure. Another group of VM-Series firewalls, deployed in the same fashion, secure outbound, east-west, and backhaul traffic. Each VM-Series firewall has an interface in the public, private, and management VPC networks. The VM-Series firewalls have static routes for all internal subnets deployed within the private and service project VPC networks while the VM-Series public interface obtains a default route through DHCP.

You can automate the VM-Series configuration management by using Panorama in the centralized management project to provide a secure, scalable, and automated architecture. Bootstrapping speeds up the process of configuring and licensing the firewall and making it operational on the network. This process allows you to deploy the firewall with only a basic configuration so that it can connect to Panorama and obtain the complete operational configuration.

Figure 29 Shared VPC model



Inbound Traffic

There are two options for load balancing inbound traffic:

- **GCP Network Load Balancer**—Choose this option if you require load balancing only at Layer 4 (TCP/UDP). Health checks monitor the application instances through backend web server responses even when the application being load balanced is not a web application.
- **GCP HTTP(S) Load Balancer**—Choose this option if you require load balancing at Layer 7 (the application layer) for HTTP and HTTPS. The GCP HTTP(S) 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 backend web server instances.

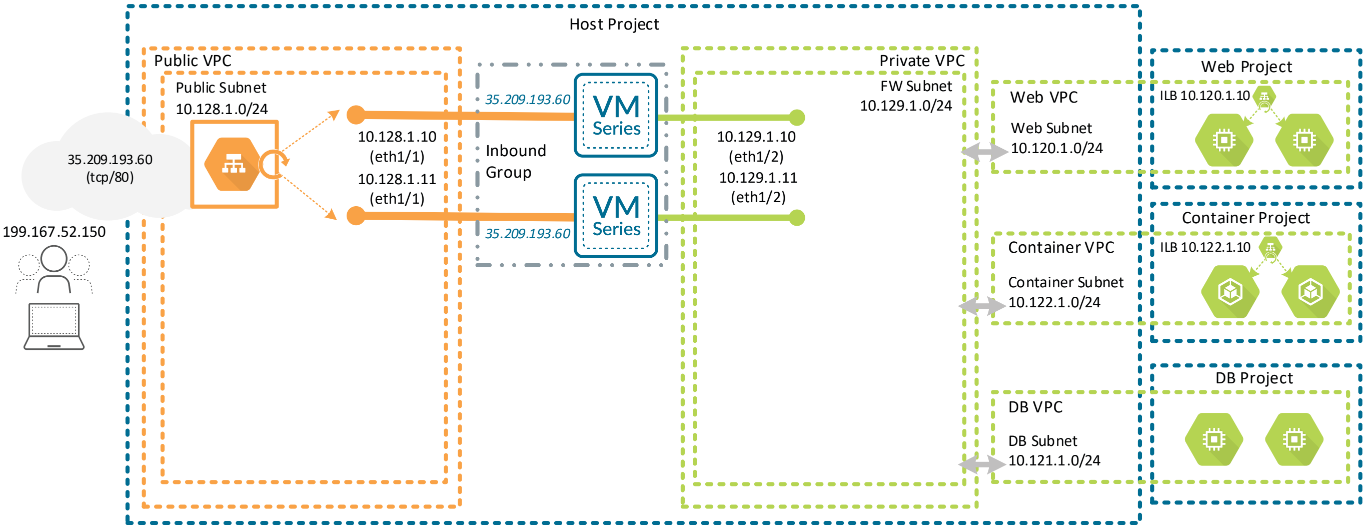
Inbound Traffic with GCP Network Load Balancer

For inbound traffic, a network load balancer distributes inbound traffic to the VM-Series firewalls in the host project. GCP firewall rules in the host project allow all inbound traffic from the internet to reach the VM-Series firewall instances in the public VPC network.

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

GCP firewall rules in the host project block all inbound traffic to the private instances of the service project 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.

Figure 30 Shared VPC—inbound traffic with network load balancer



The VM-Series firewall applies both a destination and source NAT to inbound traffic. The destination NAT translates the static public IP address associated with the network load balancer frontend to the private instance or load balancer in the service project. The source NAT translates the source to be the IP address of the private interface of the firewall, ensuring return traffic flows symmetrically.

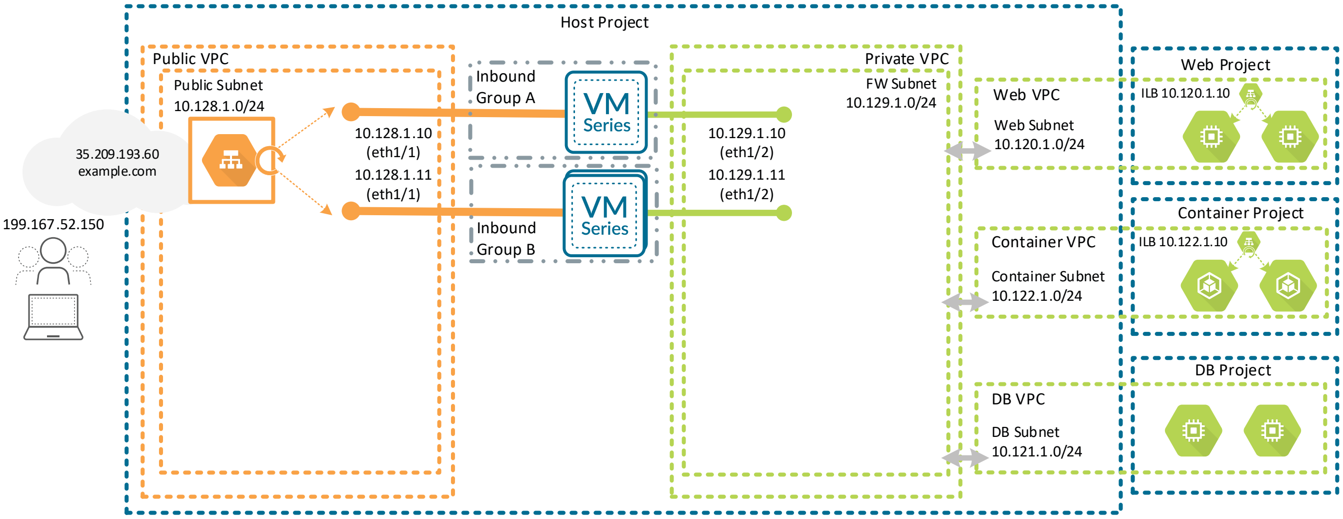
The firewall security policy allows appropriate application traffic to the instances in the private network while firewall security profiles prevent known malware and vulnerabilities from entering the network in traffic allowed by the security policy.

Inbound Traffic with GCP HTTP(S) Load Balancer

For inbound traffic, the HTTP load balancer terminates incoming connections to its frontend and initiates corresponding new connections to the VM-Series firewalls in the backend. If you configure the HTTP load balancer for multiple web applications that are behind the same set of VM-Series firewalls, you must define unique backend services for each application. Each backend service contains the same VM-Series firewall instance groups but has unique TCP ports assigned.

If the VM-Series firewalls are in separate GCP zones for resiliency, the VM-Series firewalls must be in separate unmanaged instance groups. If the HTTP load balancer has multiple instance groups in the backend service, it distributes traffic among the instance groups. If all the VM-Series firewalls are in a single zone, use a single instance group to distribute traffic amongst the VM-Series firewalls.

Figure 31 Shared VPC—inbound traffic with HTTP(S) load balancer



GCP sources all new connections from the HTTP load balancer from the IP ranges of 130.211.0.0/22 and 35.191.0.0/16. The destination IP address is the private IP address of the VM-Series public interface. Health checks monitor backend availability on all specified HTTP/HTTPS ports.

Destination NAT rules on the VM-Series firewalls map incoming traffic from the HTTP load balancer frontend to the private instance or internal load balancer in the service project. Because the destination instance or internal load balancer and the VM-Series firewalls are in different projects, they cannot use GCP internal DNS to define the destination and must use static internal IP addresses. The VM-Series firewall also applies a source NAT to inbound traffic. The source NAT translates the source to be the IP address of the private interface of the firewall, ensuring return traffic flows symmetrically.

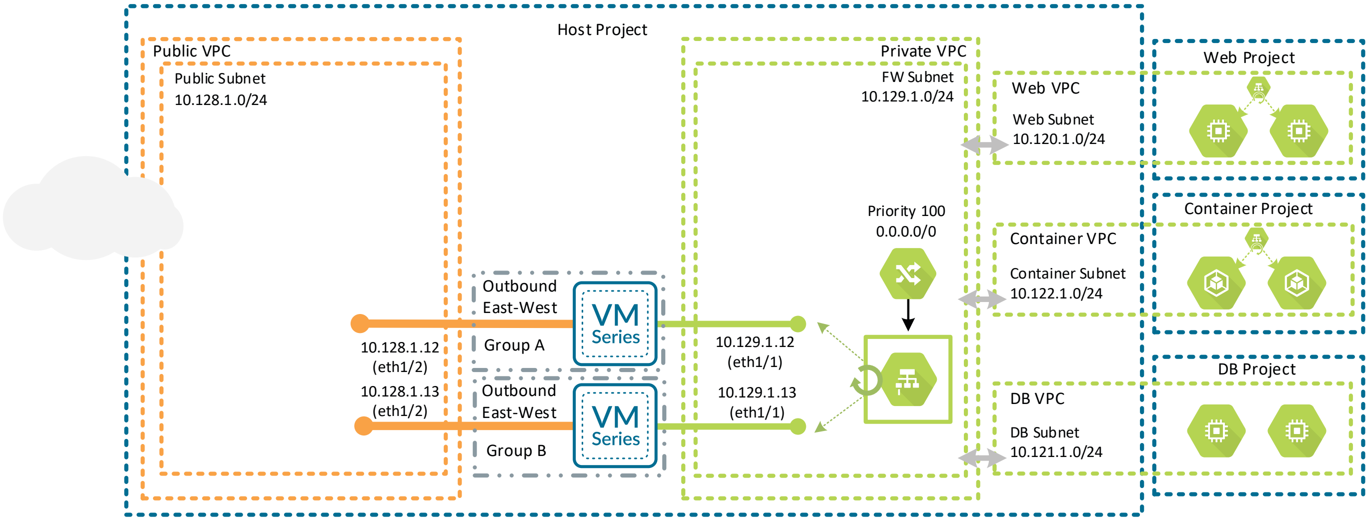
The VM-Series firewall security policy allows HTTP/HTTPS application traffic from the load balancer to the instances in the service project, and VM-Series firewall security profiles prevent known malware and vulnerabilities from entering the network in traffic allowed by the security policy. To support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

Outbound Traffic

For outbound traffic from instances in the service projects, a GCP internal network load balancer deployed in the private VPC distributes outbound traffic to the outbound VM-Series firewall group in the host project. You can configure the internal load balancer to forward all traffic to the VM-Series in the backend group as opposed to the protocol/port configuration that the inbound network or HTTP load balancers use. A GCP custom default route uses the internal network load balancer as a next-hop. VPC Network Peering exports the custom default route to all peered service projects. GCP firewall rules in the host project allow all outbound traffic to the internet to reach the VM-Series firewall instances in the private VPC network.

You use VM-Series firewall security policies to limit what applications and resources the private instances in the services that projects 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. Because the VM-Series firewall's public interface is in a different VPC network than the private interface, the system's default route in the public VPC network determines the path traffic from the VM-Series firewall takes to the internet. When the outbound traffic leaves the public VPC network, GCP translates the source address to the public IP address assigned to the VM-Series firewall's public interface.

Figure 32 Shared VPC—outbound traffic



The VM-Series firewall security policy allows appropriate application traffic from the private instances in the service projects 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

Functional segmentation by application tiers—such as web, database, container, or business units—are all done by placing the instances for each group in separate VPC networks. In the Shared VPC design, each service project should have a unique VPC network. In this deployment, east-west traffic, or traffic between service project VPCs, flows through the VM-Series firewalls in the host project. This spoke-to-spoke east-west traffic, or traffic between service projects, follows the same custom default route that forwards outbound traffic to the internal network load balancer and then to the VM-Series firewalls.

As with outbound traffic, you use the VM-Series firewall security policies to limit what applications and resources the private instances in the service project can reach. Because each VM-Series interface should have a unique security zone applied, you should use zone information as you define NAT and security policies to differentiate traffic from a private zone destined to the internet, versus traffic from a private zone VPC network destined to another private zone VPC network. In most designs, the VM-Series firewall does not need to translate the destination IP address. For east-west inter-VPC traffic, the VM-Series firewall must translate the source IP address to the IP address of the VM-Series firewall's private egress interface. Without this source NAT, traffic might not return to the firewall.

Note

Because bi-directional NAT matches traffic on any zone, do not enable bi-directional NAT in NAT policy rules. If you enable bi-directional NAT, the NAT policy might incorrectly translate east-west traffic.

The VM-Series firewall security policy allows appropriate application traffic from the private instances in the spoke projects to the internet. You should implement the east-west inter-VPC security policies by whitelisting.

Backhaul Traffic

To get traffic from on-premises resources to private instances in the service projects, 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 GCP VM-Series firewalls to the on-premises gateways. The custom default route configuration for outbound traffic in the host project provides the path for traffic from instances in the service project to reach on-premises resources through the VM-Series firewalls and vice versa.

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

VPC NETWORK PEERING DESIGN MODEL

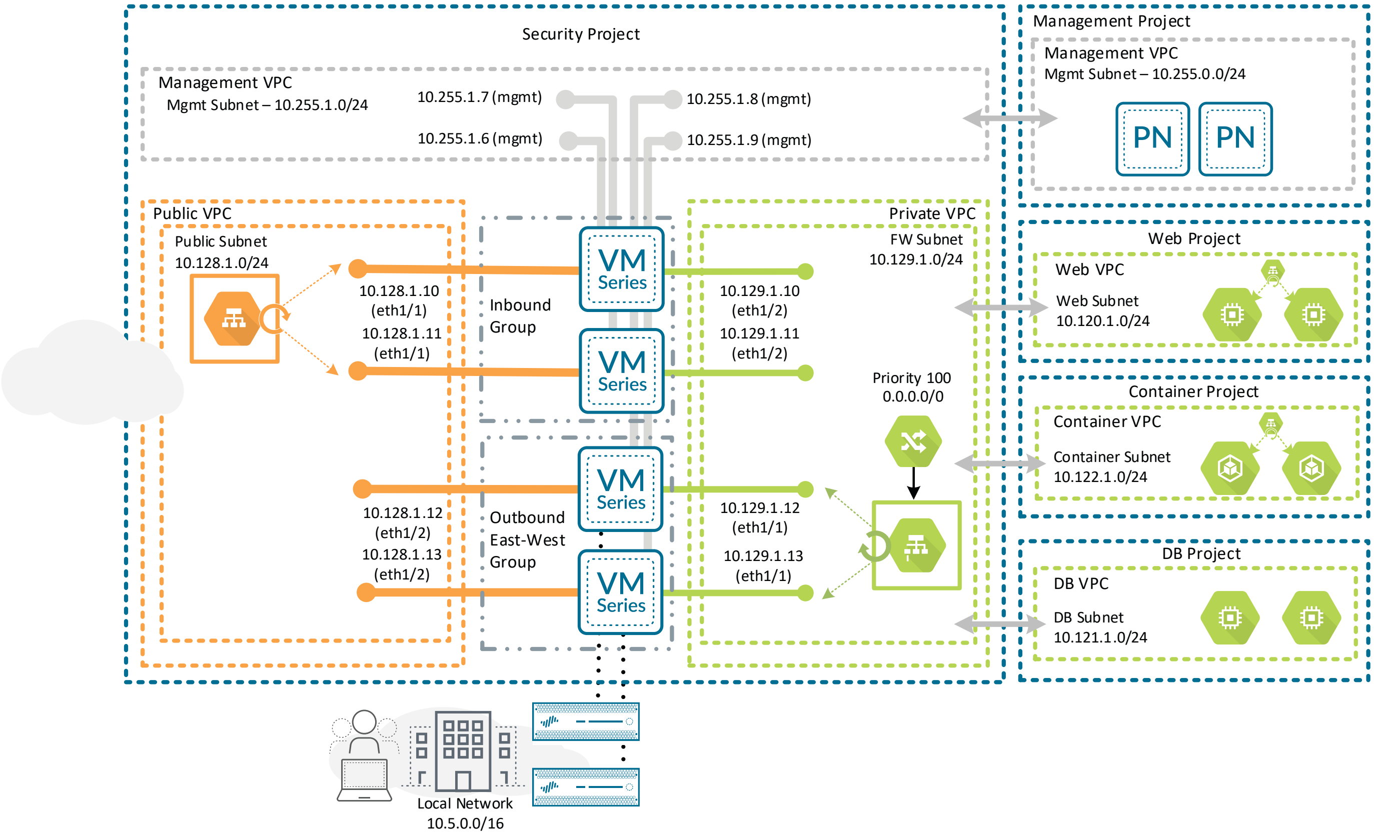
Another design option is the VPC Network Peering design model. The VPC Network Peering design also uses a hub-and-spoke architecture consisting of a hub security project (which contains the VM-Series firewalls and public, private, and management VPCs) and peered projects as spokes. The main difference between the two models is how the VPC Network Peering model implements the administrative boundaries for network and compute resource control. The VPC Network Peering model has some inherent flexibility that allows you to also deploy compute resources in the security project.

The VPC Network Peering model accommodates organizations that wish to consolidate ad hoc projects and VPC infrastructure within a flexible architecture. In the VPC Network Peering model, the security project and peered project administrators are individually responsible for VPC network configuration, peering relationships, and resource management within their respective projects. The VPC Network Peering design provides common security infrastructure and policy for all traffic flows (inbound, outbound, east-west, and backhaul) for the instances in the peered projects.

To provide VM-Series security to the peered projects, you peer a hub project private VPC with a resource VPC in the spoke project. In the hub project, a group of VM-Series firewalls secure inbound traffic flows. You must deploy the VM-Series firewalls in the same GCP region, but in separate zones, to avoid downtime caused by GCP infrastructure maintenance or failure. Another group of VM-Series firewalls, deployed in the same fashion, secure outbound, east-west, and backhaul traffic. Each VM-Series firewall has an interface in the public, private, and management VPC networks. The VM-Series firewalls have static routes for all internal subnets deployed within the private and peered project VPC networks, while the VM-Series public interface obtains a default route through DHCP.

You can automate the VM-Series configuration management by using Panorama in the centralized management project to provide a secure, scalable, and automated architecture. Bootstrapping speeds up the process of configuring and licensing the firewall and making it operational on the network. This process allows deployment of the firewall with only a basic configuration so that it can connect to Panorama and obtain the complete operational configuration.

Figure 33 VPC Network Peering model



Inbound Traffic

There are two options for load-balancing inbound traffic:

- GCP Network Load Balancer—Choose this option if you require load balancing only at Layer 4 (TCP/UDP). Health checks monitor the application instances through backend web server responses even when the application being load balanced is not a web application.
- GCP HTTP(S) Load Balancer—Choose this option if you require load balancing at Layer 7 (the application layer) for HTTP and HTTPS. The GCP HTTP(S) 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 backend web server instances.

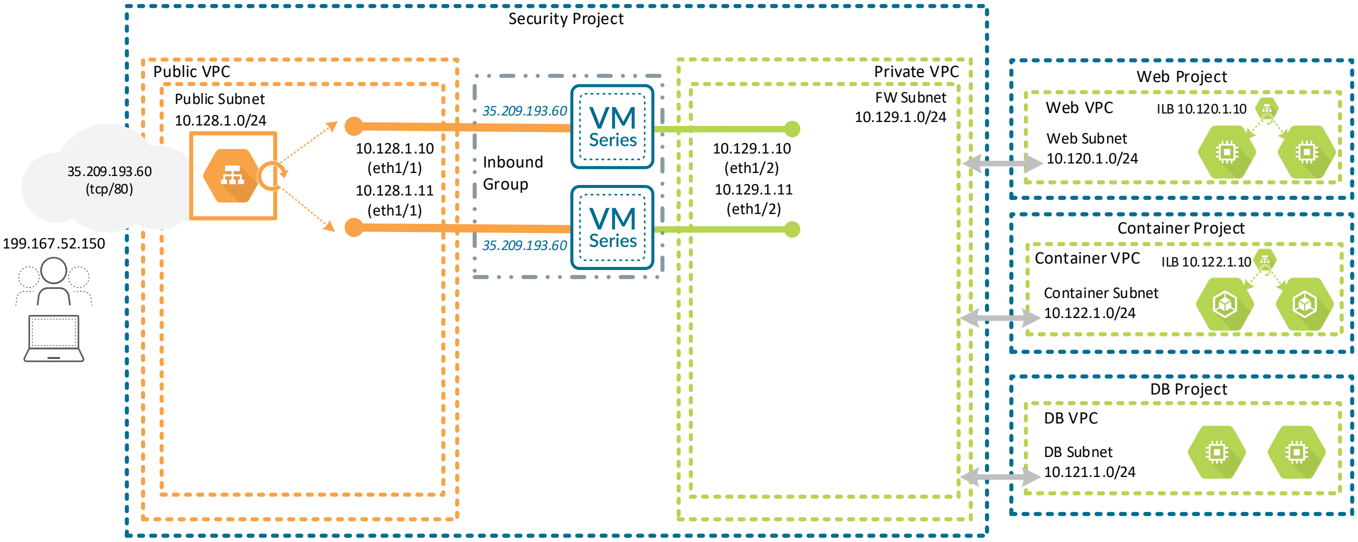
Inbound Traffic with GCP Network Load Balancer

For inbound traffic, a network load balancer distributes inbound traffic to the VM-Series firewalls in the security project. GCP firewall rules in the security project allow all inbound traffic from the internet to reach the VM-Series firewall instances in the public VPC network.

The network load balancer forwards traffic destined to the load balancer's frontend IP address and port pair to the VM-Series firewalls in the target pool. Common ports required for inbound traffic include TCP/80 (HTTP) and TCP/443 (HTTPS). The load balancer distributes traffic among the VM-Series firewalls based on the traffic 5-tuple. The public load balancer uses health checks to monitor backend instance availability through the VM-Series firewalls to the private instances.

GCP firewall rules in the security project block all inbound traffic to the peered project's 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.

Figure 34 VPC Network Peering—inbound traffic with network load balancer



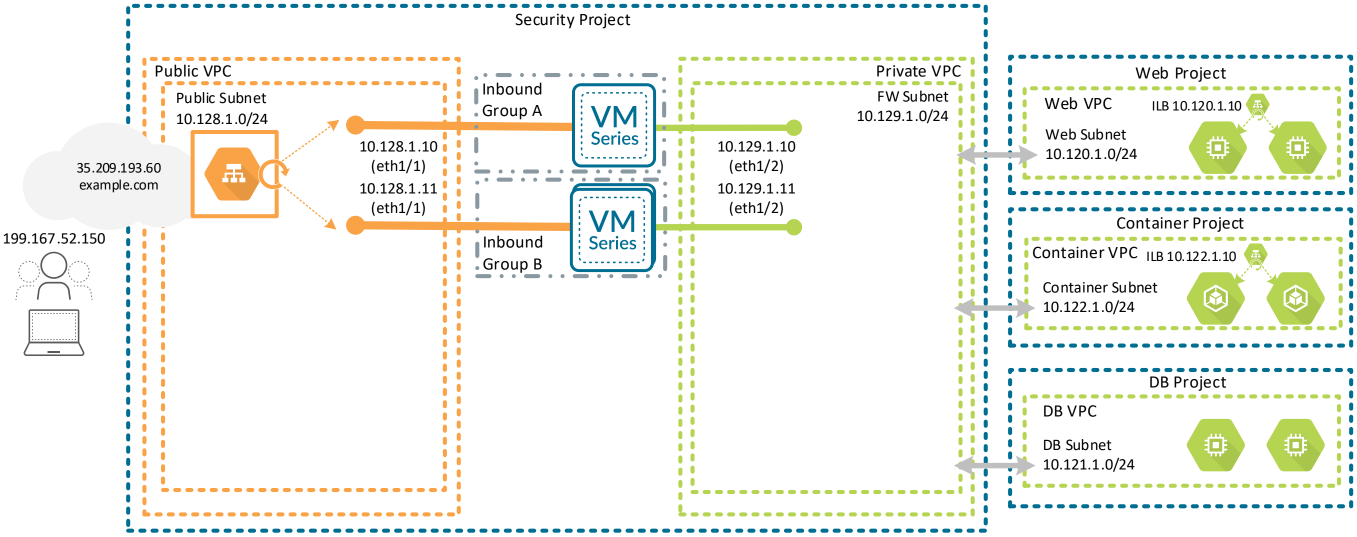
The VM-Series firewall applies both a destination and source NAT to inbound traffic. The destination NAT translates the static public IP address associated with the network load balancer frontend to the private instance or load balancer in the peered project. The source NAT translates the source to be 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 network, while firewall security profiles prevent known malware and vulnerabilities from entering the network in traffic allowed by the security policy.

Inbound Traffic with GCP HTTP(S) Load Balancer

For inbound traffic, the HTTP load balancer terminates incoming connections to its frontend and initiates corresponding new connections to the VM-Series firewalls in the backend. If you configure the HTTP load balancer for multiple web applications that are behind the same set of VM-Series firewalls, you must define unique backend services for each application. Each backend service contains the same VM-Series firewall instance groups but has unique TCP ports assigned to them.

If the VM-Series firewalls are in separate GCP zones for resiliency, the VM-Series firewalls must be in separate unmanaged instance groups. If the HTTP load balancer has multiple instance groups in the backend service, it distributes traffic among the instance groups. If all the VM-Series firewalls are in a single zone, you should use a single instance group to distribute traffic amongst the VM-Series firewalls.

Figure 35 VPC Network Peering—inbound traffic with HTTP(S) load balancer



GCP sources all new connections from the HTTP load balancer from the IP ranges of 130.211.0.0/22 and 35.191.0.0/16. The destination IP address is the private IP address of the VM-Series public interface. Health checks monitor backend availability on all specified HTTP/HTTPS ports.

Destination NAT rules on the VM-Series firewalls map incoming traffic from the HTTP load balancer frontend to the private instance or internal load balancer in the peered project. Because the destination instance or internal load balancer and the VM-Series firewalls are in different projects, they cannot use GCP internal DNS to define the destination and must use static internal IP addresses. The VM-Series firewall also applies a source NAT to inbound traffic. The source NAT translates the source to be the IP address of the private interface of the firewall, ensuring return traffic flows symmetrically.

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

Outbound Traffic

For outbound traffic from instances in the peered projects, a GCP internal network load balancer deployed in the private VPC of the security project distributes outbound traffic to the outbound VM-Series firewall group. You can configure the internal load balancer to forward all traffic to the VM-Series in the backend group as opposed to the protocol/port configuration that the inbound network or HTTP load balancers use. A GCP custom default route uses the internal network load balancer as a next-hop. VPC Network Peering exports the custom default route to all peered peered projects. The GCP firewall rules in the security project allow all outbound traffic to the internet to reach the

VM-Series firewall outbound instances in the private VPC network.

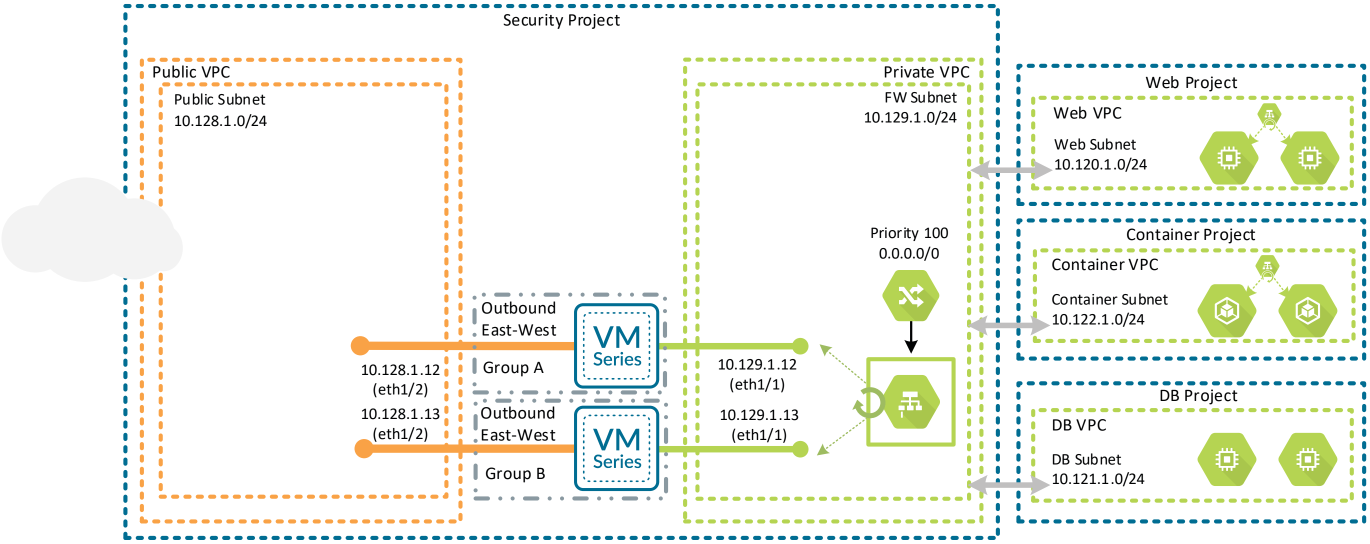
You use the VM-Series firewall security policies to limit what applications and resources the private instances in the peered projects can reach. In most designs, the VM-Series firewall does not need to translate the destination IP address. The VM-Series firewall must translate the source IP address to the IP address of the VM-Series firewall's public interface. Without this source NAT, traffic might not return to the firewall. Because the VM-Series firewall's public interface is in a separate VPC network than the private interface, the system default route in the public VPC network determines the path traffic takes from the VM-Series firewall to the internet.

When the outbound traffic leaves the public VPC network, GCP translates the source address to the public IP address assigned to the VM-Series firewall's public interface.

Note

Because bi-directional NAT matches traffic on any zone, do not enable bi-directional NAT in NAT policy rules. If you enable bi-directional NAT, the NAT policy might incorrectly translate east-west traffic.

Figure 36 VPC Network Peering—outbound and east-west inter-VPC traffic



The VM-Series firewall security policy allows appropriate application traffic from the private instances in the peered projects to the internet. You should implement the outbound security policy by 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

Functional segmentation by application tiers—such as web, database, container, or business units—are all done by placing the instances for each group in separate projects. In the VPC Network Peering design, each peered project has one or more unique VPC networks. In this deployment, east-west traffic, or traffic between peered project VPCs, flows through the VM-Series firewalls in the security project. This spoke-to-spoke east-west traffic, or traffic between peered projects, follows the same custom default route that forwards outbound traffic to the internal network load balancer in the security project and then to the outbound VM-Series firewalls.

As with the outbound traffic, you use VM-Series firewall security policies to limit what applications and resources the private instances in the each peered project can reach. Because each VM-Series interface has a unique security zone applied, you should use zone information as you define NAT and security policies to differentiate traffic from a private zone destined to the internet, versus traffic from a private zone VPC network destined to another private zone VPC network. In most designs, the VM-Series firewall does not need to translate the destination IP address. For east-west inter-VPC traffic, the VM-Series firewall must translate the source IP address to the IP address of the VM-Series firewall's private egress interface. Without this source NAT, traffic might not return to the firewall.

Note

Because bi-directional NAT matches traffic on any zone, do not enable bi-directional NAT in NAT policy rules. If you enable bi-directional NAT, the NAT policy might incorrectly translate east-west traffic.

The VM-Series firewall security policy allows appropriate application traffic from the private instances in the peered projects to the internet. You should implement the east-west inter-VPC security policies by whitelisting.

Backhaul Traffic

To get traffic from on-premises resources to private instances in the peered projects, 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 GCP VM-Series firewalls to the on-premises gateways. The custom default route configuration for outbound traffic in the security project provides the path for traffic from instances in the peered project to reach on-premises resources through the VM-Series firewalls and vice versa.

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

Summary

Moving applications to the public cloud requires the same enterprise-class security as your private network. The shared-security model in public-cloud deployments places the burden of protecting the applications and data on the customer. Deploying Palo Alto Networks VM-Series firewalls on GCP provides a scalable security infrastructure that includes native cloud automation support in addition to protection from known and unknown threats, complete application visibility, and a security policy similar to your private network security policy .

The design models presented in this guide provide you with enterprise production-level designs. If you have advanced project requirements or they evolve over time, a more modular design might be necessary. The Shared VPC model centralizes infrastructure control and security policy while delegating administrative responsibilities as the number of projects in your organization increases. The VPC Network Peering model provides a centralized security infrastructure and policy control while providing flexible project peering options.

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