Google Cloud and Hybrid Network Architecture

Philipp Maier
Course Developer, Google Cloud

In this module, we discuss Google Cloud network architectures, including hybrid architectures.

Learning objectives

- Design VPC networks to optimize for cost, security, and performance.
- Configure global and regional load balancers to provide access to services.
- Leverage Cloud CDN to provide lower latency and decrease network egress.
- Evaluate network architecture using the Network Intelligence Center.
- Connect networks using peering, VPNs and Cloud Interconnect

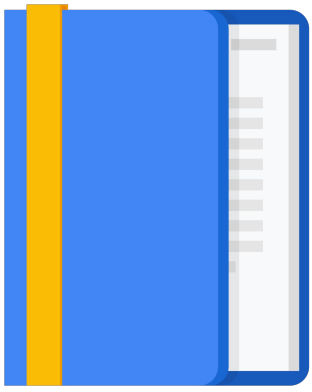
We will start by talking about how to design VPC networks to optimize for cost, security, and performance. Then, we'll cover the configuration of global and regional load balancers to provide access to services.

As part of the load balancer configuration, you can enable Cloud CDN to provide lower latency and decrease network egress, which ultimately decreases your networking costs. We will also introduce the Network Intelligence Center to evaluate your network's architecture and go over the network connection options, including peering, VPN, and Cloud Interconnect.

Agenda

Designing Google Cloud Networks

Connecting Networks



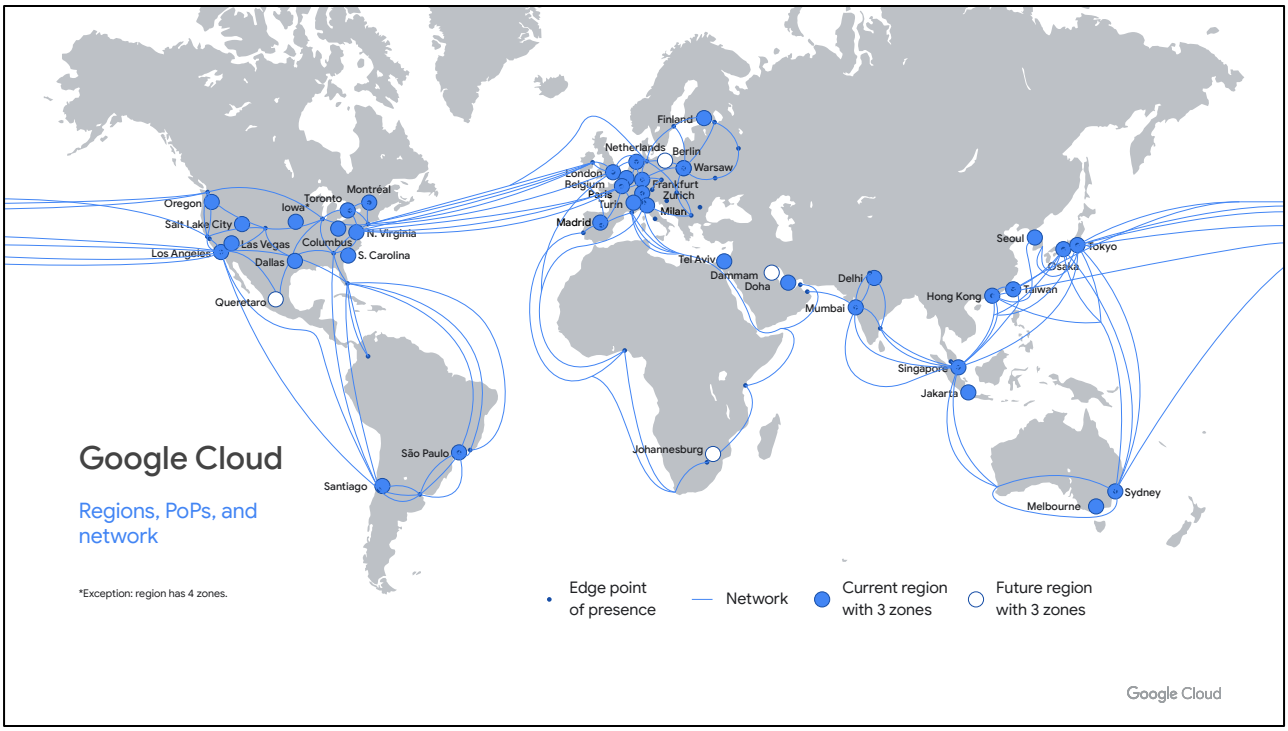
Let’s get started by designing Google Cloud networks and load balancers.

Google runs a worldwide network that connects regions all over the world

Design your networks based on location, number of users, scalability, fault tolerance, and other service requirements.

Google runs a worldwide network that connects regions all over the world. You can use this high-bandwidth infrastructure to design your cloud networks to meet your requirements such as location, number of users, scalability, fault tolerance, and latency.

Let’s take a closer look at Google Cloud’s network.



This map represents Google Cloud's reach. On a high level, Google Cloud consists of regions, which are the icons in blue; points of presence, or PoPs, which are the dots in grey; a global private network, which is represented by the blue lines; and services.

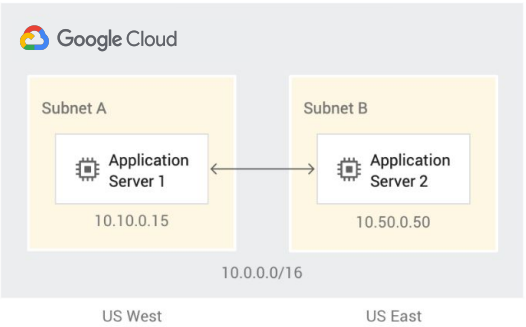
A region is a specific geographical location where you can run your resources. This map shows several regions that are currently operating, as well as future regions and their zones.

The PoPs are where Google's network is connected to the rest of the internet. Google Cloud can bring its traffic closer to its peers because it operates an extensive global network of interconnection points. This reduces costs and provides users with a better experience.

The network connects regions and PoPs and is composed of a global network of fiber optic cables with several submarine cable investments.

In Google Cloud, VPC networks are global

- When creating networks, create subnets for the regions you want to operate in.
- Resources across regions can reach each other without any added interconnect.
- If you are a global company, choose regions around the world.
- If your users are close together, choose the region closest to them plus a backup region.
- A project can have multiple networks.



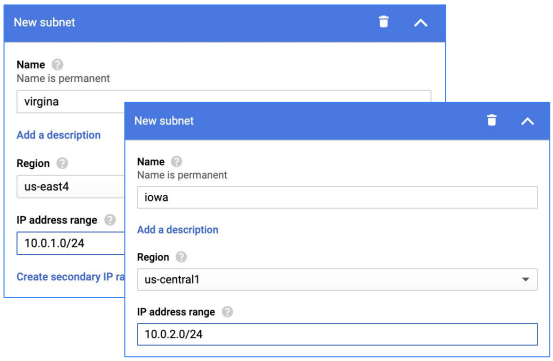
In Google Cloud, VPC networks are global, and you can either create auto mode networks that have one subnet per region or create your own custom mode network where you get to specify which region to create a subnet in.

Resources across regions can communicate using their internal IP addresses without any added interconnect. For example, the diagram on the rights shows two subnets in different regions with a server on each subnet. They can communicate with each other using their internal IP addresses because they are connected to the same VPC network.

Selecting which regions to create subnets in depends on your requirements. For example, if you are a global company, you will most likely create subnetworks in regions across the world. If users are within a particular region, it may be suitable to select just one subnet in a region closest to these users and maybe a backup region close by. Also, you can have multiple networks per project. These networks are just a collection of regional subnetworks or subnets.

When creating custom subnets, specify the region and the internal IP address range

- IP address ranges cannot overlap.
- Machines in the same VPC can communicate via their internal IP address regardless of the subnet region.
- Subnets don't need to be derived from a single CIDR block.
- Subnets are expandable without down time.
- IP Aliasing or Secondary range can be set on the subnet.

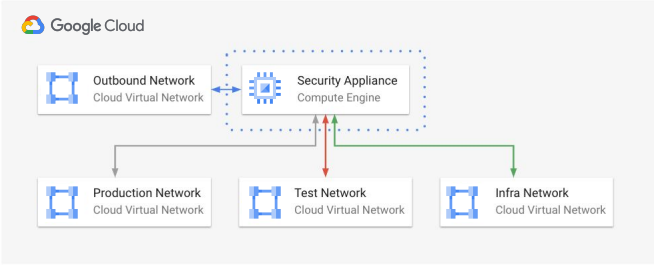


To create custom subnets you specify the region and the internal IP address range, as illustrated in the screenshots on the right. The IP ranges of these subnets don't need to be derived from a single CIDR block, but they cannot overlap with other subnets of the same VPC network. This applies to primary and secondary ranges. Secondary ranges allow you to define alias IP addresses.

Also, you can expand the primary IP address space of any subnets without any workload shutdown or downtime. Once you defined your subnets, machines in the same VPC network can communicate with each other through their internal IP address regardless of the subnet they are connected to.

A single VM can have multiple network interfaces connecting to different networks

- Each network must have a subnet in the region the VM is created in.
- Each interface must be attached to a different VPC.
- Maximum of 8 interfaces per VM.



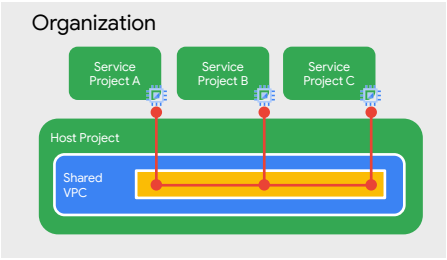
Now, a single VM can have multiple network interfaces connecting to different VPC networks. This graphic illustrates an example of a Compute Engine instance connected to four different networks covering production, test, infra, and an outbound network.

A VM must have at least one network interface but can have up to 8, depending on the instance type and the number of vCPUs. A general rule is that with more vCPUs, more network interfaces are possible. All of the network interfaces must be created when the instance is created, and each interface must be attached to a different network.

A Shared VPC is created in one project, but can be shared and used by other projects

Requires an organization

- Create the VPC in the host project.
- Shared VPC admin shares the VPC with other service projects.



Allows centralized control over network configuration

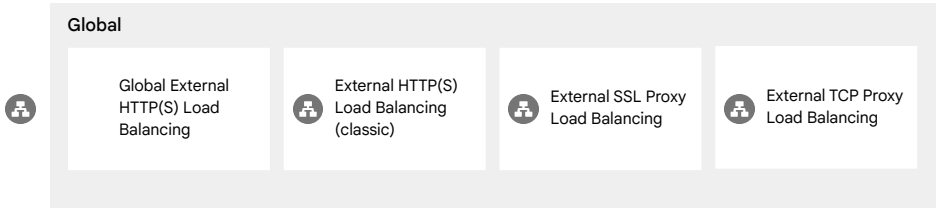
- Network admins configure subnets, firewall rules, routes, etc.
- Remove network admin rights from developers.
- Developers focus on machine creation and configuration in the shared network.
- Disable the creation of the default network using an organizational policy.

Shared VPC allows an organization to connect resources from multiple projects of a single organization to a common VPC network. This allows the resources to communicate with each other securely and efficiently using internal IPs from that network.

This graphic shows a scenario where a shared VPC is used by three other projects, namely service projects A, B, and C. Each of these projects has a VM instance that is attached to the Shared VPC.

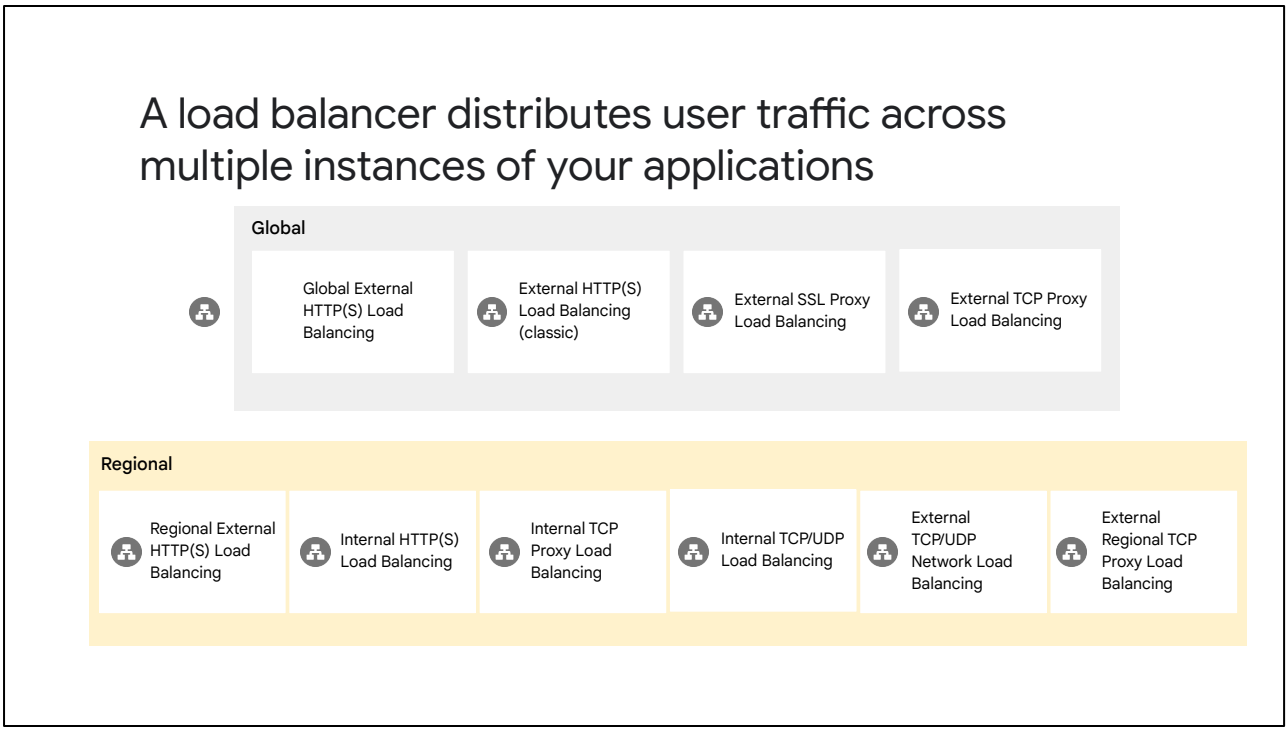
Shared VPC is a centralized approach to multi-project networking, because security and network policy occurs in a single designated VPC network. This allows for network administrator rights to be removed from developers so they can focus on what they do best. Meanwhile, organization network administrators maintain control of resources such as subnets, firewall rules, and routes while delegating the control of creating resources such as instances to service project administrators or developers.

A load balancer distributes user traffic across multiple instances of your applications



Google Cloud offers different types of load balancers that can be divided into two categories: global and regional.

The **global load balancers** are the HTTP(S), SSL proxy, and TCP proxy load balancers. These load balancers leverage Google's frontends, which are software-defined, distributed systems that sit in Google's points of presence and are distributed globally. Therefore, you want to use a global load balancer when your users and instances are globally distributed, your users need access to the same applications and content, and you want to provide access using a single anycast IP address.



Regional load balancers are external and internal HTTP(S), and TCP Proxy. There are also internal TCP/UDP, and external TCP/UDP Network load balancers.

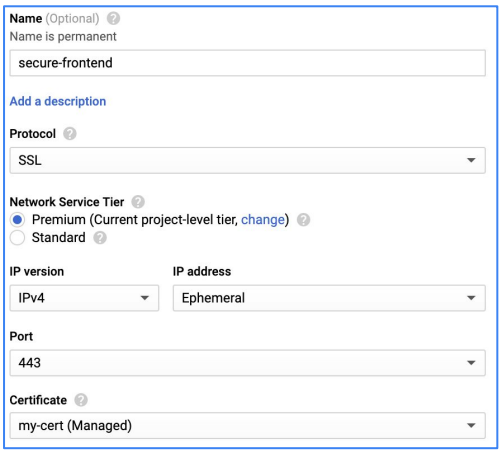
The internal and network load balancers, distribute traffic to instances that are in a single Google Cloud region. The internal load balancer uses Andromeda, which is a Google Cloud's software-defined network virtualization stack, and the network load balancers use Maglev, which is a large, distributed software system.

The internal load balancer for HTTP(S) traffic, is a proxy-based, regional Layer 7 load balancer that enables you to run and scale your services behind a private load balancing IP address that is accessible only to the load balancer's region in your VPC network.

For more information on Load Balancers, please refer to the Cloud Load Balancing Overview.

If your load balancers have public IPs, secure them using SSL

- Supported by HTTP and TCP load balancers
- Self-managed and Google-managed SSL certificates



If your load balancers have public IP addresses, traffic will likely be traversing the internet. I recommend securing this traffic with SSL, which is available for HTTP and TCP load balancers as shown in the screenshot on the right.

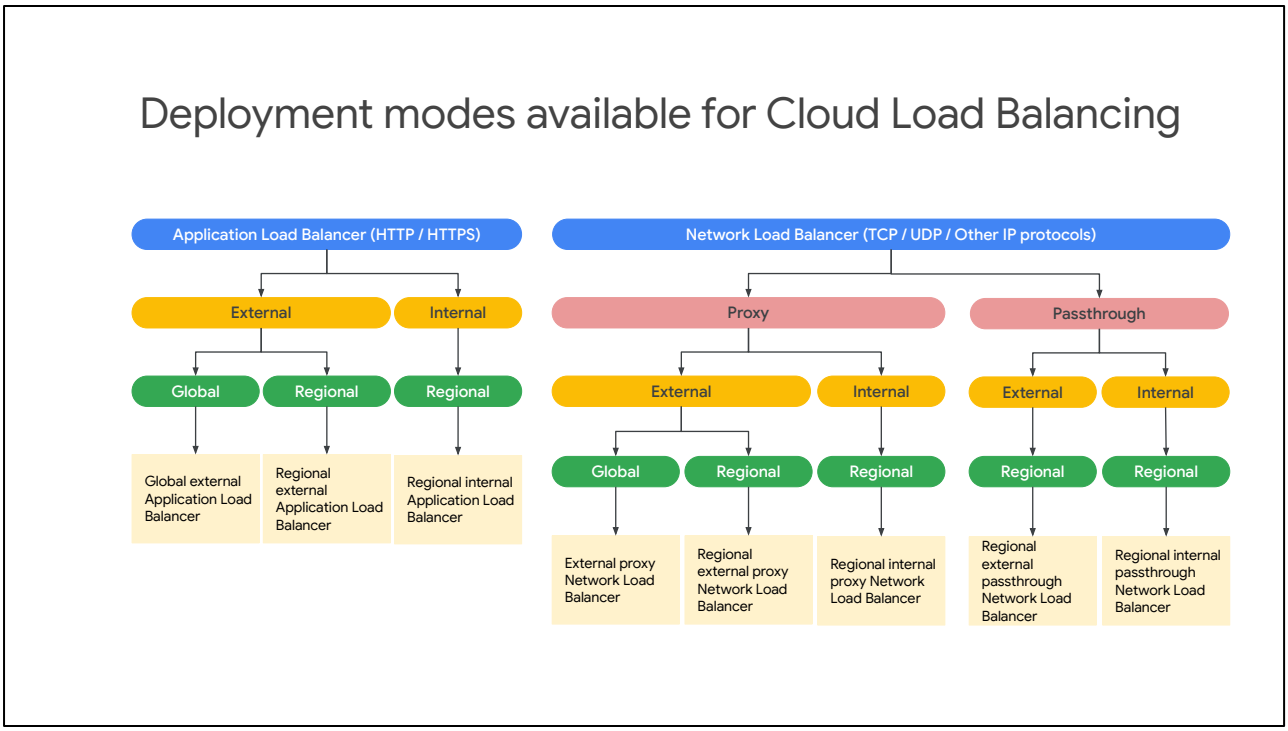
You can use either self-managed SSL certificates or Google-managed SSL certificates when using SSL.

For lower-latency and decreased egress cost leverage Cloud CDN

- Can be enabled when configuring the HTTP global load balancer.
- Caches static content worldwide using Google Cloud edge-caching locations.
- Cache static data from web servers in Compute Engine instances, GKE pods, or Cloud Storage buckets.

Now, if you are using HTTP(S) load balancing, you should leverage Cloud CDN to achieve lower latency and decreased egress costs. You can enable Cloud CDN by simply checking a box when configuring a HTTP(S) global load balancer. Cloud CDN caches content across the world using Google Cloud's edge-caching locations. This means that the content is cached closest to the users making the requests.

The data that is cached can be from a variety of sources, including Compute Engine instances, GKE pods, or Cloud Storage buckets.



To determine which Cloud Load Balancing product to use, you must first determine what traffic type your load balancers must handle. As a general rule, you'd choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP(S) traffic. You'd choose a proxy Network Load Balancer to implement TLS offload, TCP proxy, or support for external load balancing to backends in multiple regions. You'd choose a passthrough Network Load Balancer to preserve client source IP addresses, avoid the overhead of proxies, and to support additional protocols like UDP, ESP, and ICMP. UDP, or if you need to expose client IP addresses to your applications.

You can further narrow down your choices depending on your application's requirements: whether your application is external (internet-facing) or internal and whether you need backends deployed globally or regionally.

Summary of Google Cloud load balancers

Load balancer	Deployment mode	Traffic type	Network Service Tier	Load-balancing scheme
Application Load Balancers	**Global external**	HTTP or HTTPS	Premium	EXTERNAL_MANAGED
	Regional external	HTTP or HTTPS	Standard	EXTERNAL_MANAGED
	Classic	HTTP or HTTPS	Global in Premium Regional in Standard	EXTERNAL
	Internal Always regional	HTTP or HTTPS	Premium	INTERNAL_MANAGED
Proxy Network Load Balancers	**Global external**	TCP with optional SSL offload	Global in Premium Regional in Standard	EXTERNAL
	Regional external	TCP	Standard only	EXTERNAL_MANAGED
	Internal Always regional	TCP without SSL offload	Premium only	INTERNAL_MANAGED
Passthrough Network Load Balancers	**External** Always regional	TCP, UDP, ESP, GRE, ICMP, and ICMPv6	Premium or Standard	EXTERNAL
	Internal Always regional	TCP or UDP	Premium only	INTERNAL

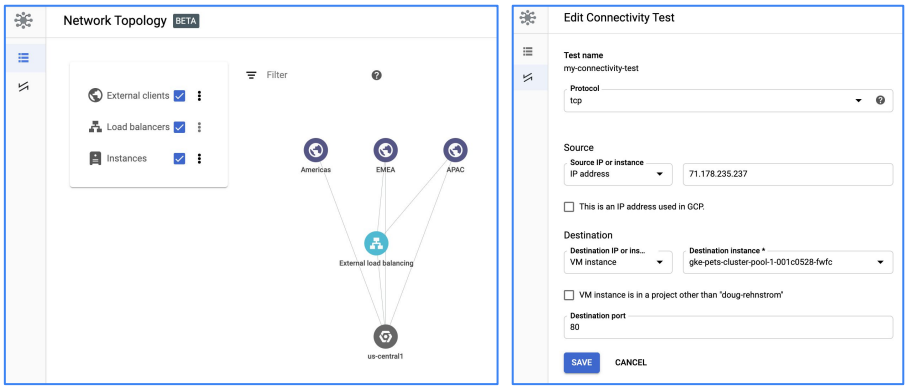
If you prefer a table over a flow chart, we recommend this summary table.

The load-balancing scheme is an attribute on the forwarding rule and the backend service of a load balancer and indicates whether the load balancer can be used for internal or external traffic.

The term *_MANAGED in the load-balancing scheme indicates that the load balancer is implemented as a managed service either on Google Front Ends (GFEs) or on the open source Envoy proxy. In a load-balancing scheme that is *_MANAGED, requests are routed to either the GFE or to the Envoy proxy.

For more information on Network Service Tiers, refer to the documentation: https://cloud.google.com/network-tiers/docs/overview.

Network Intelligence Center can be used to visualize network topology and test network connectivity



As part of our discussion in designing VPC networks, I also want to mention the Network Intelligence Center. The Network Intelligence Center is a Google Cloud service that can be used to visualize your VPC's network topology and test network connectivity.

The left-hand graphic shows a simple network topology visualization with external clients in three different regions and traffic routed through an external load balancer to resources in us-central1. This facility is extremely valuable for confirming the network topology when configuring a network or performing diagnostics. The right-hand graphic shows the configuration of a connectivity test between a source and destination along with a protocol and port. The following tests can be performed:

- Between the source and destination endpoints in your (VPC) network
- From your VPC network to and from the internet
- From your VPC network to and from your on-premises network

Activity 8: Defining network characteristics

Refer to your Design and Process Workbook.

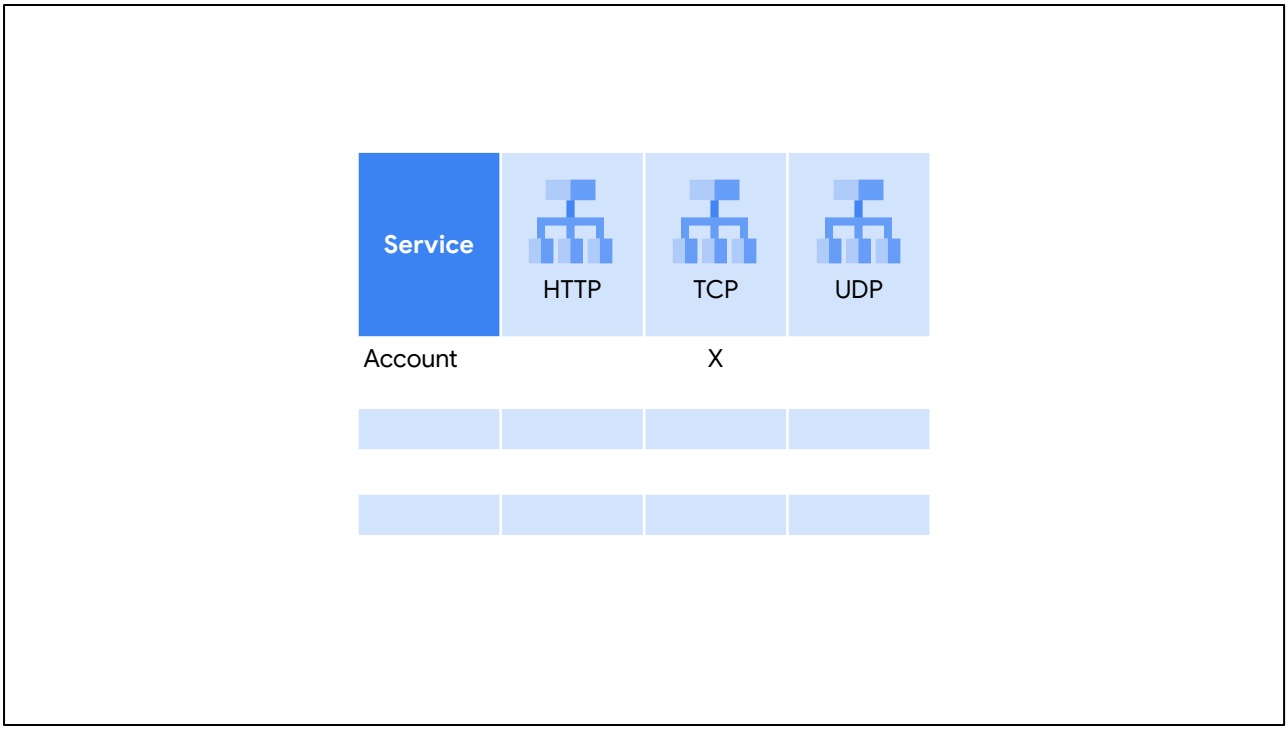
- Specify the network characteristics for your case study VPC.
- Choose the type of load balancer required for each service.

In this design activity, you specify the network characteristics for your case study and select the type of load balancer required for each service.

Service	Internet facing or Internal only	HTTP	TCP	UDP	Multiregional?
account	*Internal only*		*Yes*		*No*

In the first part of this activity, describe the network characteristics of each of your services by filling out this table.

The example shown here is for the account service. Because this is a backend service, it will only be accessed internally using TCP, and we don't plan to deploy this service in multiple regions.



Then, based on the network characteristics for each of your services, select the right load balancer using this table. Based on the parameters from the last slide, we will use the regional TCP load balancer.

Refer to activities 8a and 8b in your design workbook to fill out similar tables for your services, and feel free to explore Cloud CDN to decrease latency and network egress costs.

Review Activity 8: Defining network characteristics

- Specify the network characteristics for your case study VPC.
- Choose the type of load balancer required for each service.

In this activity, you were asked to specify the network characteristics of each of your services and choose the appropriate load balancer for each one.

Service	Internet facing or Internal only	HTTP	TCP	UDP	Multiregional?
Search	Internet facing	X			Yes
Inventory	Internal		X		No
Analytics	Internet facing	X			No
Web UI	Internet facing	X			Yes
Orders	Internal		X		No

Here's a completed example for our online travel portal, ClickTravel.

The inventory and orders service are internal and regional using TCP. The other services need to be facing the internet using HTTP. We decided to deploy these to multiple regions for lower latency, higher performance, and high availability to our users who are in multiple countries around the world.

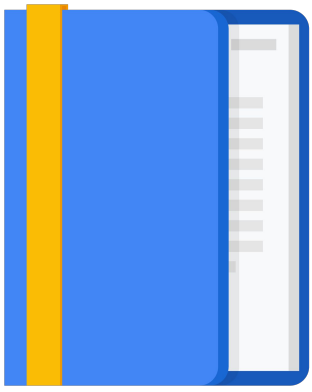
Service	HTTP	TCP	UDP
Search	X		
Inventory		X	
Analytics	X		
Web UI	X		
Orders		X	

Based on those network characteristics, we chose the global HTTP load balancer for our public-facing services and the internal TCP load balancer for our internal-facing services.

Agenda

Designing Google Cloud Networks

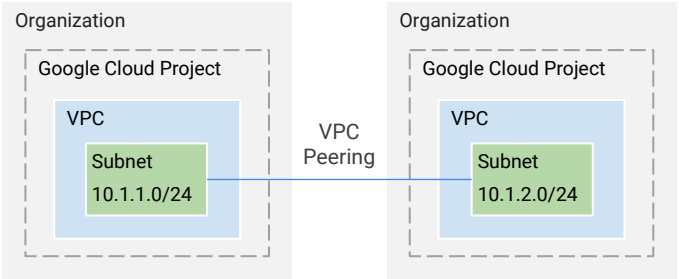
Connecting Networks



Let’s focus our attention on Google Cloud’s network connectivity products, which are Peering, Cloud VPN, and Cloud Interconnect.

Use VPC peering to connect networks when they are both in Google Cloud

- Can be the same or different organizations.
- Subnet ranges cannot overlap.
- Network admins for each VPC must approve the peering requests.

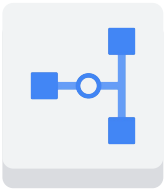


If you're trying to connect two VPC networks, you might want to consider VPC peering. VPC Peering allows private RFC 1918 connectivity across two VPC networks, regardless of whether they belong to the same project or the same organization. Now, remember that each VPC network will have firewall rules that define what traffic is allowed or denied between the networks.

This diagram shows a VPC peering connection between two networks belonging to different projects and different organizations. You might notice that the subnet ranges do not overlap. This is a requirement for a connection to be established. Speaking of the connection, network administrators for each VPC network must configure a VPC peering request for a connection to be established.

Cloud VPN securely connects your on-premises network to your Google Cloud VPC network

- Useful for low-volume data connections
- Classic VPN: 99.9% SLA
- High-availability (HA) VPN: 99.99% SLA
- Supports:
  - Site-to-site VPN
  - Static routes (Classic VPN only)
  - Dynamic routes (Cloud Router)
  - IKEv1 and IKEv2 ciphers



Cloud VPN

Cloud VPN securely connects your on-premises network to your Google Cloud VPC network through an IPsec VPN tunnel. Traffic traveling between the two networks is encrypted by one VPN gateway, then decrypted by the other VPN gateway. This protects your data as it travels over the public internet, and that's why Cloud VPN is useful for low-volume data connections.

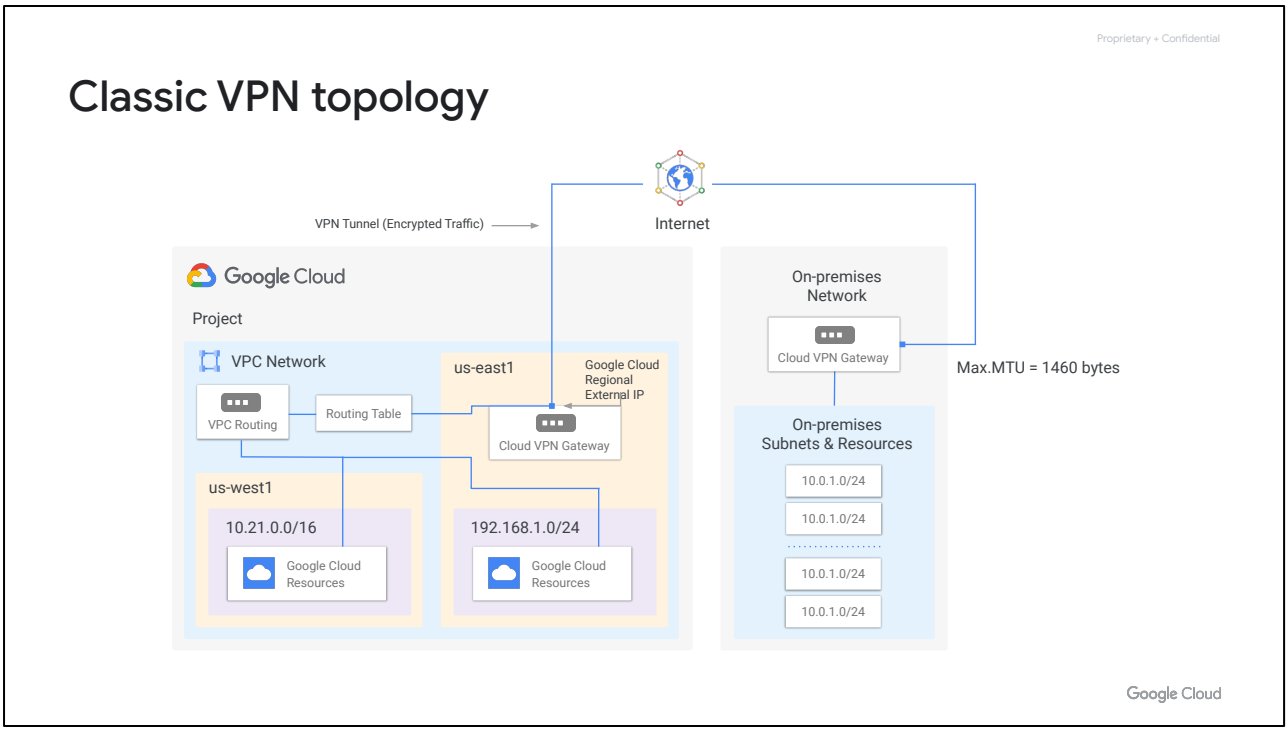
As a managed service, Cloud VPN provides an SLA of 99.9% monthly uptime for the Classic VPN configuration, and 99.99% monthly uptime for the High-availability (HA) VPN configuration. The Classic VPN gateways have a single interface and a single external IP address whereas high-availability VPN gateways have two interfaces with two external IP addresses (one for each gateway). The choice of VPN gateway comes down to your SLA requirement and routing options.

Cloud VPN supports site-to-site VPN, static routes and dynamic routes using Cloud Router and IKEv1 and IKEv2 ciphers. However, static routes are only supported by Classic VPN.

Also, Cloud VPN doesn't support use cases where client computers need to "dial in" to a VPN using client VPN software.

For more information on the SLA and these features, refer to the documentation:

https://cloud.google.com/vpn/docs/concepts/overview



Let’s walk through an example of Cloud VPN. This diagram shows a Classic VPN connection between your VPC and on-premises network. Your VPC network has subnets in us-east1 and us-west1, with Google Cloud resources in each of those regions.

These resources are able to communicate using their internal IP addresses because routing within a network is automatically configured (assuming that firewall rules allow the communication).

Now, in order to connect to your on-premises network and its resources, you need to configure your Cloud VPN gateway, on-premises VPN gateway, and two VPN tunnels. The Cloud VPN gateway is a regional resource that uses a regional external IP address.

You on-premises VPN gateway can be a physical device in your data center or a physical or software-based VPN offering in another cloud provider's network. This VPN gateway also has an external IP address.

A VPN tunnel then connects your VPN gateways and serves as the virtual medium through which encrypted traffic is passed. In order to create a connection between two VPN gateways, you must establish two VPN tunnels. Each tunnel defines the

connection from the perspective of its gateway, and traffic can only pass when the pair of tunnels is established.

Now, one thing to remember when using Cloud VPN is that the maximum transmission unit (MTU) for your on-premises VPN gateway cannot be greater than 1460 bytes. This is because of the encryption and encapsulation of packets. For more information on this MTU consideration, refer to the documentation https://cloud.google.com/vpn/docs/concepts/mtu-considerations.

In addition to Classic VPN, Google Cloud also offers a second type of Cloud VPN gateway, HA VPN.

HA VPN overview

- Provides 99.99% service availability.
- Google Cloud automatically chooses two external IP addresses.
  - Supports multiple tunnels
  - VPN tunnels connected to HA VPN gateways must use dynamic (BGP) routing
- Supports site-to-site VPN for different topologies/configuration scenarios:
  - An HA VPN gateway to peer VPN devices
  - An HA VPN gateway to an Amazon Web Services (AWS) virtual private gateway
  - Two HA VPN gateways connected to each other

HA VPN is a high availability Cloud VPN solution that lets you securely connect your on-premises network to your Virtual Private Cloud (VPC) network through an IPsec VPN connection in a single region. HA VPN provides an SLA of 99.99% service availability. To guarantee a 99.99% availability SLA for HA VPN connections, you must properly configure two or four tunnels from your HA VPN gateway to your peer VPN gateway or to another HA VPN gateway.

When you create an HA VPN gateway, Google Cloud automatically chooses two external IP addresses, one for each of its fixed number of two interfaces. Each IP address is automatically chosen from a unique address pool to support high availability.

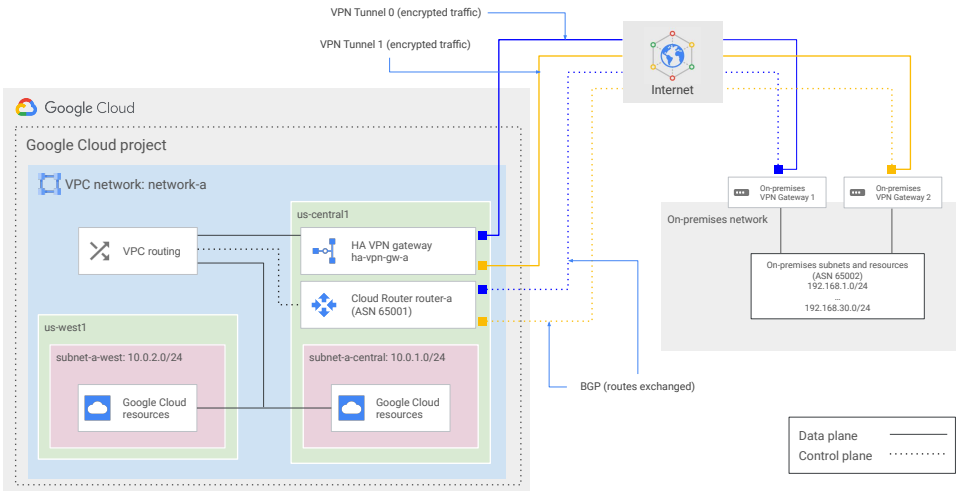
Each of the HA VPN gateway interfaces supports multiple tunnels. You can also create multiple HA VPN gateways. When you delete the HA VPN gateway, Google Cloud releases the IP addresses for reuse. You can configure an HA VPN gateway with only one active interface and one external IP address; however, this configuration does not provide a 99.99% service availability SLA. VPN tunnels connected to HA VPN gateways must use dynamic (BGP) routing. Depending on the way that you configure route priorities for HA VPN tunnels, you can create an active/active or active/passive routing configuration.

HA VPN supports site-to-site VPN in one of the following recommended topologies or configuration scenarios:

- An HA VPN gateway to peer VPN devices
- An HA VPN gateway to an Amazon Web Services (AWS) virtual private gateway
- Two HA VPN gateways connected to each other

Let's explore these configurations in a bit more detail.

HA VPN to peer VPN gateway topology



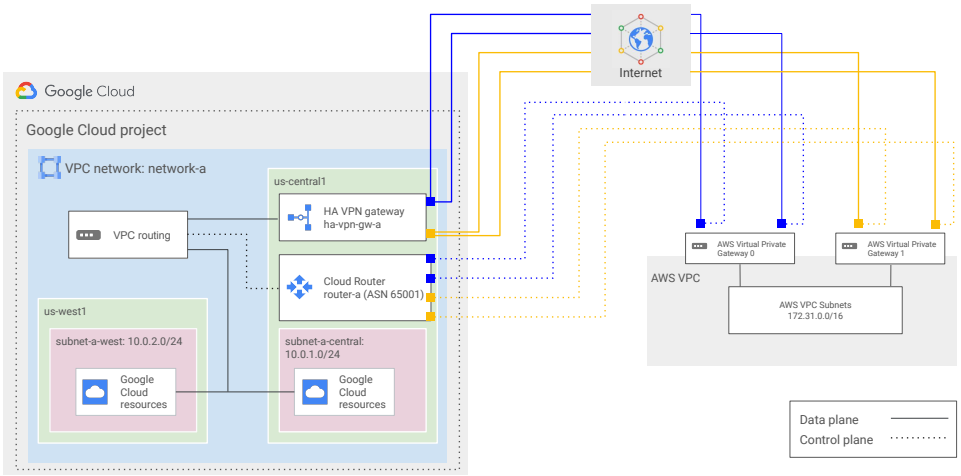
There are three typical peer gateway configurations for HA VPN. An HA VPN gateway to two separate peer VPN devices, each with its own IP address, an HA VPN gateway to one peer VPN device that uses two separate IP addresses and an HA VPN gateway to one peer VPN device that uses one IP address.

Let's walk through an example. In this topology, one HA VPN gateway connects to two peer devices. Each peer device has one interface and one external IP address. The HA VPN gateway uses two tunnels, one tunnel to each peer device. If your peer-side gateway is hardware-based, having a second peer-side gateway provides redundancy and failover on that side of the connection.

A second physical gateway lets you take one of the gateways offline for software upgrades or other scheduled maintenance. It also protects you if there is a failure in one of the devices.

In Google Cloud, the REDUNDANCY_TYPE for this configuration takes the value TWO_IPS_REDUNDANCY. The example shown here provides 99.99% availability.

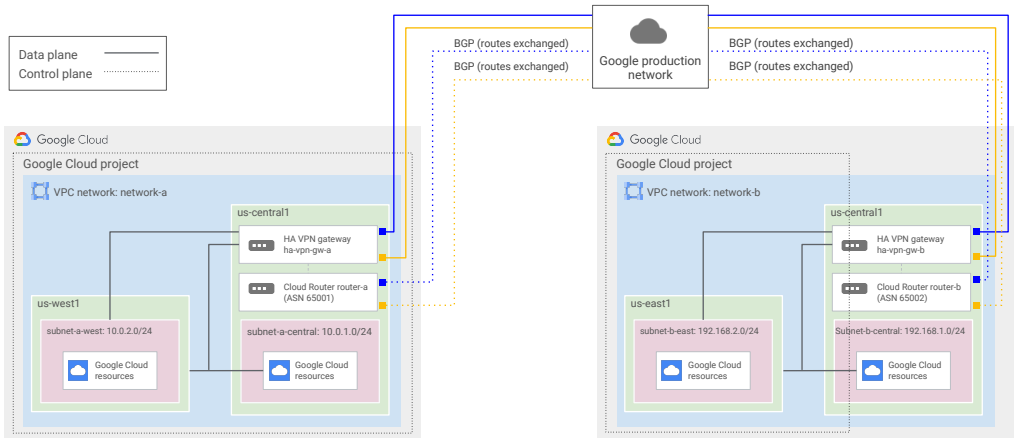
HA VPN to AWS peer gateway topology



When configuring an HA VPN external VPN gateway to Amazon Web Services (AWS), you can use either a transit gateway or a virtual private gateway. Only the transit gateway supports equal-cost multipath (ECMP) routing. When enabled, ECMP equally distributes traffic across active tunnels. Let's walk through an example.

In this topology, there are three major gateway components to set up for this configuration. An HA VPN gateway in Google Cloud with two interfaces, two AWS virtual private gateways, which connect to your HA VPN gateway, and an external VPN gateway resource in Google Cloud that represents your AWS virtual private gateway. This resource provides information to Google Cloud about your AWS gateway. The supported AWS configuration uses a total of four tunnels. Two tunnels from one AWS virtual private gateway to one interface of the HA VPN gateway, and two tunnels from the other AWS virtual private gateway to the other interface of the HA VPN gateway.

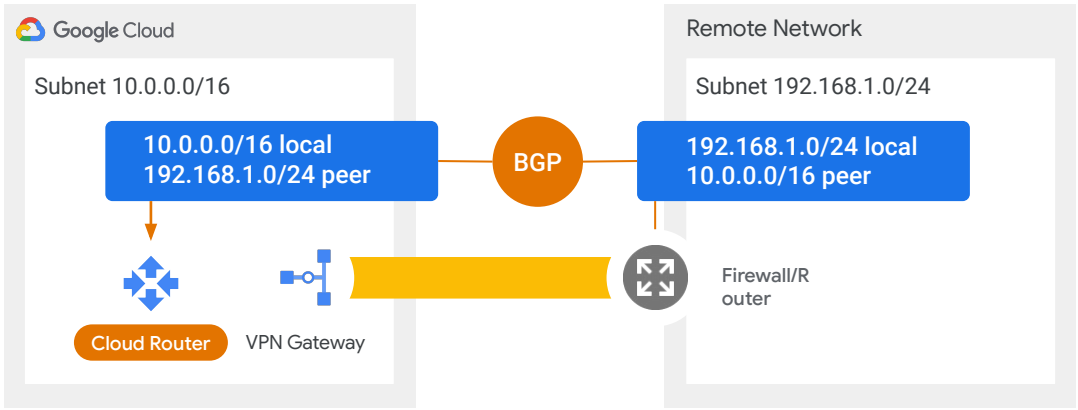
HA VPN to peer VPN gateway topology



You can connect two Google Cloud VPC networks together by using an HA VPN gateway in each network. The configuration shown provides 99.99% availability. From the perspective of each HA VPN gateway you create two tunnels. You connect `interface 0` on one HA VPN gateway to `interface 0` on the other HA VPN, and `interface 1` on one HA VPN gateway to `interface 1` on the other HA VPN.

For more information on HA VPN, refer to the documentation Cloud VPN topologies. For information on moving to HA VPN, see Moving to HA VPN.

Cloud Router enables dynamic discovery of routes between connected networks



I mentioned earlier that Cloud VPN supports both static and dynamic routes. In order to use dynamic routes, you need to configure Cloud Routers. A Cloud Router can manage routes for a Cloud VPN tunnel using Border Gateway Protocol, or BGP. This routing method allows for routes to be updated and exchanged without changing the tunnel configuration.

This allows for new subnets like staging in the VPC network and Rack 30 in the peer network to be seamlessly advertised between networks.

Use Cloud Interconnect when a dedicated high- speed connection is required between networks

- Dedicated Interconnect provides a direct connection to a colocation facility.
  - From 10 to 200 Gbps
- Partner Interconnect provides a connection through a service provider.
  - Can purchase less bandwidth from 50 Mbps
- Allows access to VPC resources using internal IP address space.
- Private Google Access allows on-premises hosts to access Google services using private IPs.

If you need a dedicated high speed connection between networks, consider using Cloud Interconnect. Cloud Interconnect has two options for extending on-premises networks: Dedicated Interconnect and Partner Interconnect.

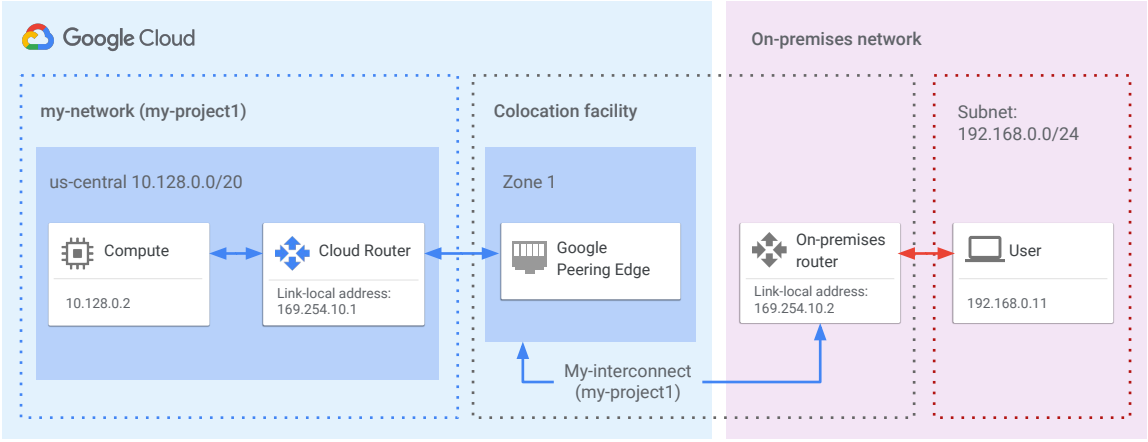
Dedicated Interconnect provides a direct connection to a colocation facility. The colocation facility must support either 10 Gbps or 100 Gbps circuits, and a dedicated connection can bundle up to eight 10 Gbs connections or two 100 Gbps for a maximum of 200 Gbps.

Partner Interconnect provides a connection through a service provider. This can be useful for lower bandwidth requirements starting from 50 Mbps.

In both cases, Cloud Interconnect allows access to VPC resources using an internal IP address space.

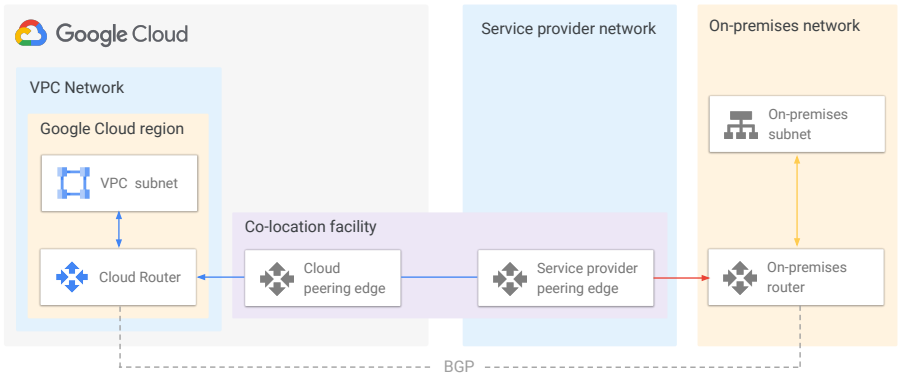
You can even configure Private Google Access for on-premises hosts to allow them to access Google services using private IP addresses.

Dedicated Interconnect provides direct physical connections



In order to use Dedicated Interconnect, you need to provision a cross connect between the Google network and your own router in a common colocation facility, as shown in this diagram. To exchange routes between the networks, you configure a BGP session over the interconnect between the Cloud Router and the on-premises router. This will allow user traffic from the on-premises network to reach Google Cloud resources on the VPC network, and vice versa.

Partner Interconnect provides connectivity through a supported service provider



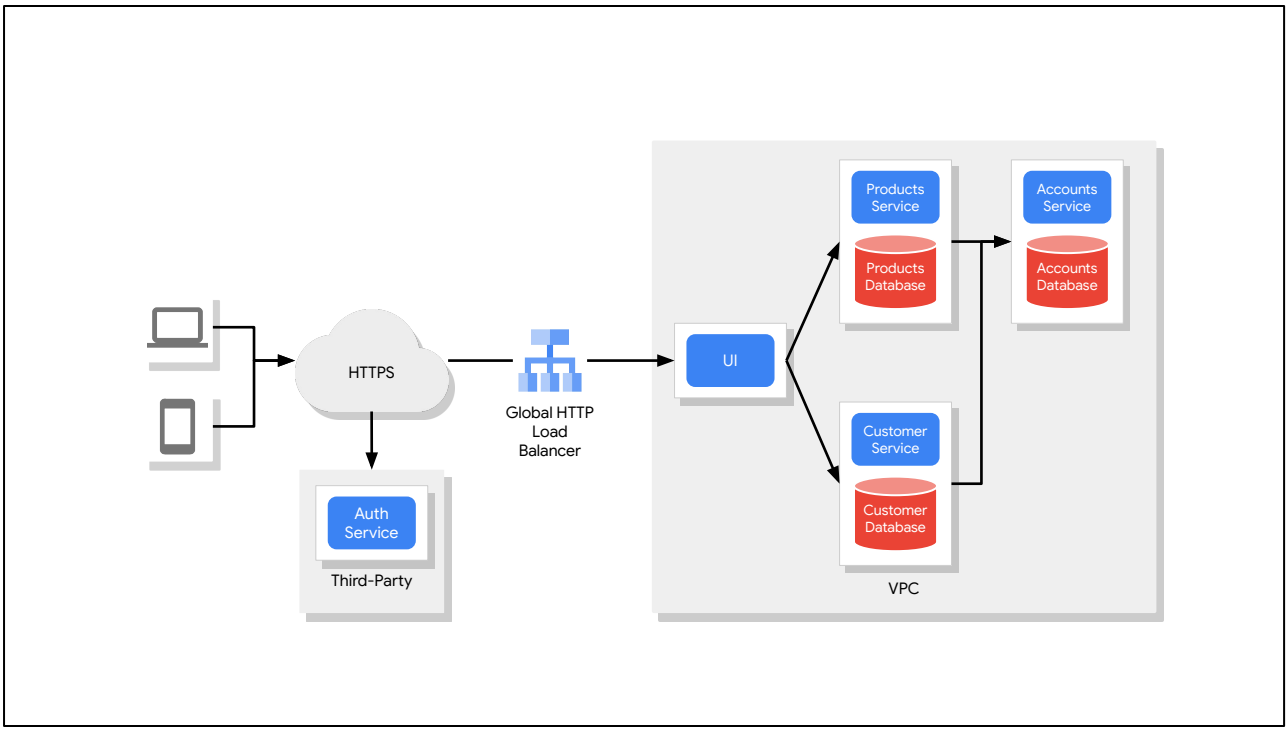
Partner Interconnect provides connectivity between your on-premises network and your VPC network through a supported service provider. This is useful if your data center is in a physical location that cannot reach a Dedicated Interconnect colocation facility or if your data needs don't warrant a Dedicated Interconnect.

Activity 9: Diagramming your network

Refer to your Design and Process Workbook.

- Draw a diagram that depicts your network requirements.

In this design activity, you draw a diagram that depicts the network requirements of your case study. Let me show you a simple example.



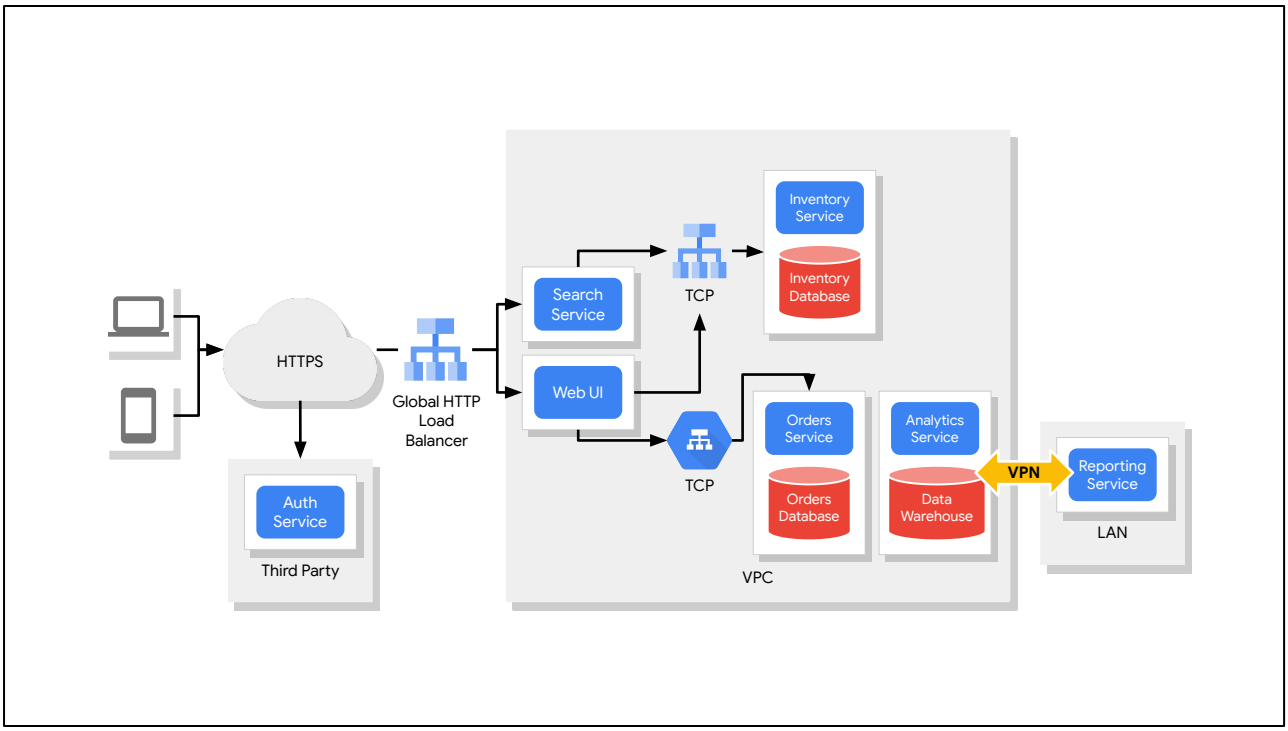
This network diagram shows where the network boundaries are and how traffic is served from our users through a load balancer to our backend. We could also include the use of Cloud CDN, Cloud VPN or any Cloud Interconnect services that are relevant to our network design.

Refer to activity 9 in your workbook to create a similar network diagram for your services.

Review Activity 9: Diagramming your network

- Draw a diagram that depicts your network requirements.

In this activity, you were ask to create a diagram that depicts the network requirements of your application.



Here's an example for our online travel portal, ClickTravel.

User traffic from mobile and web will first be authenticated using a third-party service. Then a Global HTTP Load Balancer directs traffic to our public facing Search and web UI services. From there, regional TCP load balancers direct traffic to the internal inventory and orders services.

The analytics service could leverage BigQuery as the data warehouse with an on-prem reporting service that accesses the analytics service over a VPN. This might be good enough to start, and we could refine this once we start implementing it.

Review

Google Cloud and Hybrid Network Architecture

In this module, you learned about Google Cloud networking and how to design networks that meet your application's security, performance, reliability, and scalability requirements.

We also covered the different options to connect networks using peering, VPN and Cloud Interconnect.