



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

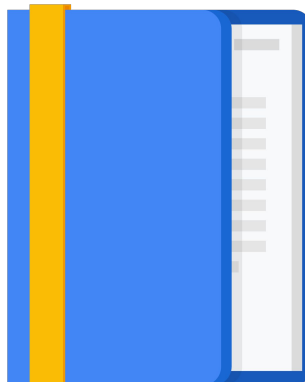
We will start by talking about how to design VPC networks to optimize for cost, security, and performance. Then, we'll cover the configuration of global and regional load balancers to provide access to services.

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.

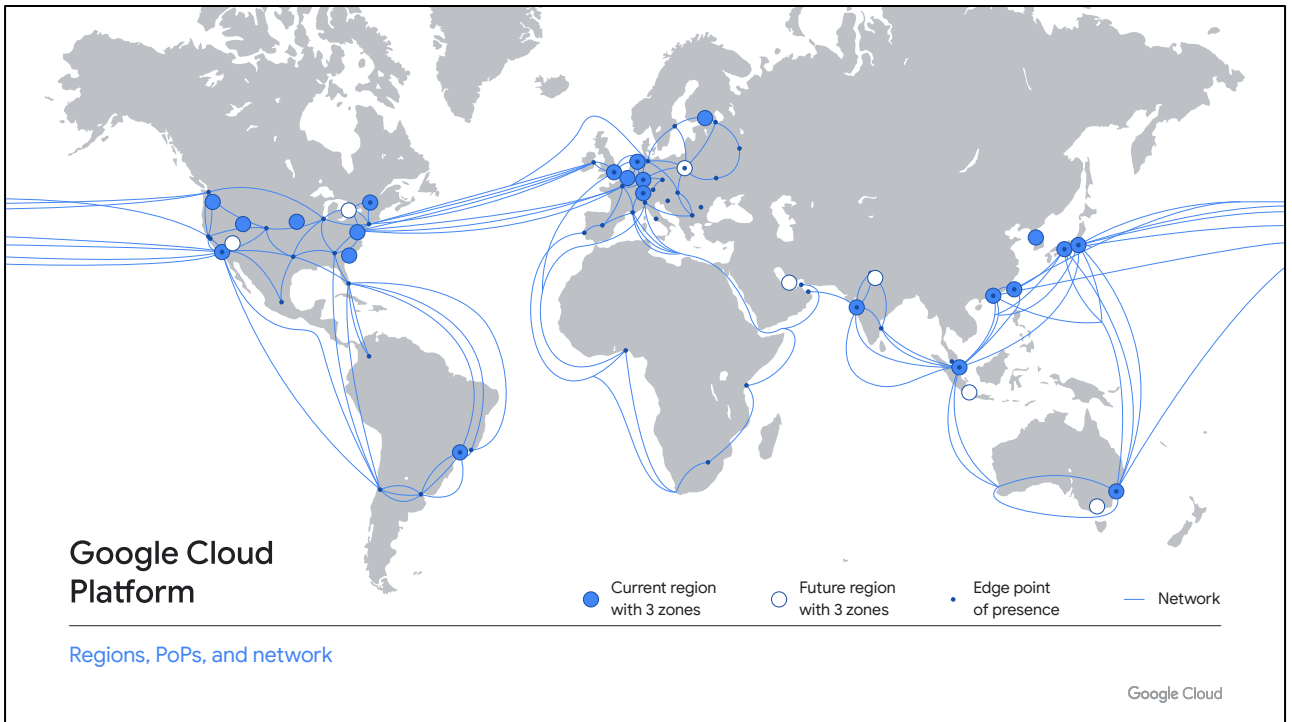
Google runs a worldwide network that connects regions all over the world

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.

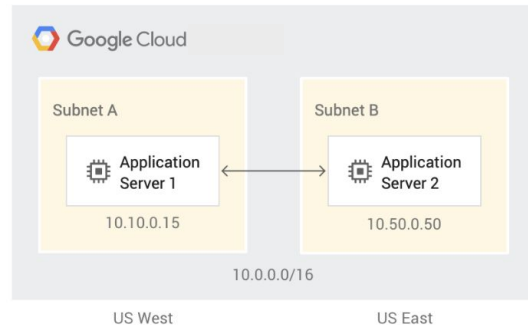
A region is a specific geographical location where you can run your resources. This map shows several regions that are currently operating, as well as future regions and their zones. As of this recording, there are 21 regions and 64 zones.

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



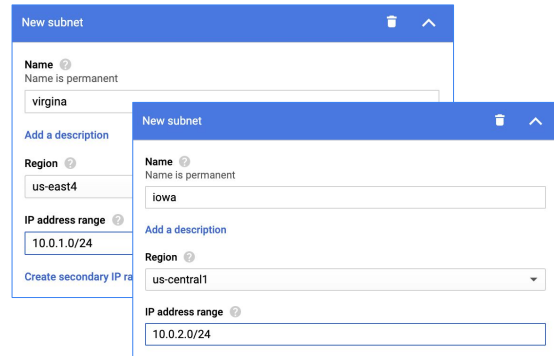
In Google Cloud, VPC networks are global, and you can either create auto mode networks that have one subnet per region or create your own custom mode network where you get to specify which region to create a subnet in.

Resources across regions can communicate using their internal IP addresses without any added interconnect. For example, the diagram on the right 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.



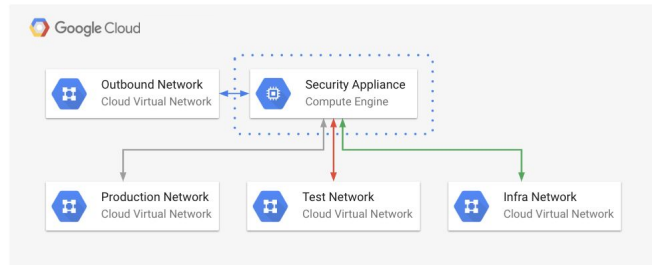
The image shows two overlapping screenshots of the AWS Management Console 'New subnet' page. The background screenshot shows a subnet named 'virginia' in the 'us-east4' region with an IP address range of '10.0.1.0/24'. The foreground screenshot shows a subnet named 'iowa' in the 'us-central1' region with an IP address range of '10.0.2.0/24'. Both screenshots show the 'Name', 'Region', and 'IP address range' fields, along with a link to 'Add a description' and a 'Create secondary IP range' button.

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.



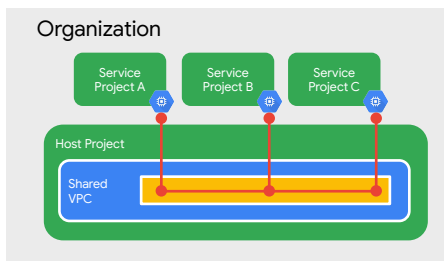
Now, a single VM can have multiple network interfaces connecting to different VPC networks. This graphic illustrates an example of a Compute Engine instance connected to four different networks covering production, test, infra, and an outbound network.

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.

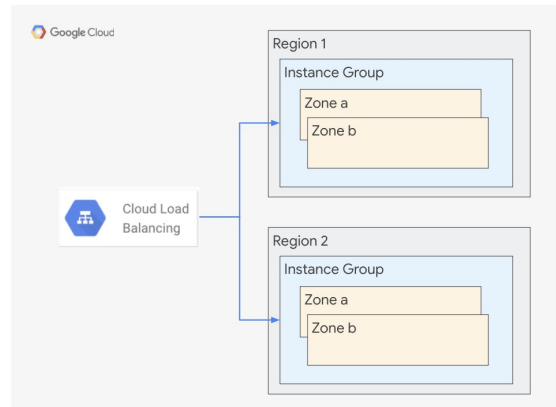
Shared VPC allows an organization to connect resources from multiple projects of a single organization to a common VPC network. This allows the resources to communicate with each other securely and efficiently using internal IPs from that network.

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.

Use a global load balancer to provide access to services deployed in multiple regions

- Global load balancing supported by HTTP load balancer and TCP and SSL proxies.
- HTTP load balancer routes requests to the region closest to the user.
 - Uses a global, anycast IP address.

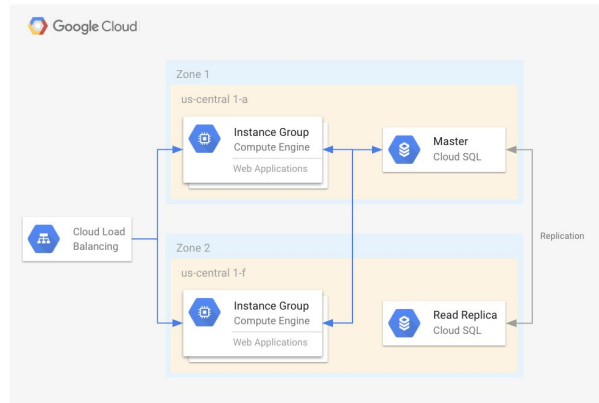


Let's talk about load balancers. Global load balancers provide access to services deployed in multiple regions. For example, the load balancer shown on this slide has a backend with two instance groups deployed in different regions. Cloud Load Balancing is used to distribute the load among these instance groups.

Global load balancing is supported by HTTP load balancers and TCP and SSL proxies in Google Cloud. For an HTTP load balancer, a global anycast IP address can be used, simplifying DNS lookup. By default, requests are routed to the region closest to the requestor.

Use a regional load balancer to provide access to services deployed in a single region

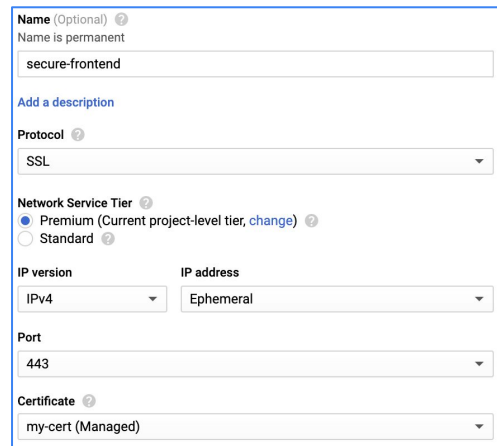
- Supported by HTTP, TCP, and UDP load balancers.
- Can have a public or private IP address.
- Can use any TCP or UDP port.



For services deployed in a single region, use a regional load balancer. This graphic illustrates resources deployed within a single region and Cloud Load Balancing routing requests to those resources. Regional load balancers support HTTP(S) and any TCP or UDP port.

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



The screenshot shows the configuration for a 'secure-frontent' load balancer. The 'Name' field is 'secure-frontent' and is marked as permanent. The 'Protocol' is set to 'SSL'. The 'Network Service Tier' is set to 'Premium (Current project-level tier, change)'. The 'IP version' is 'IPv4' and the 'IP address' is 'Ephemeral'. The 'Port' is '443'. The 'Certificate' is 'my-cert (Managed)'.

Name (Optional) ⓘ
Name is permanent

secure-frontent

[Add a description](#)

Protocol ⓘ
SSL

Network Service Tier ⓘ
☒ Premium (Current project-level tier, [change](#)) ⓘ
☐ Standard ⓘ

IP version IP address
IPv4 Ephemeral

Port
443

Certificate ⓘ
my-cert (Managed)

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 an HTTP(S) global load balancer. Cloud CDN caches content across the world using Google Cloud's edge-caching locations. This means that 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.

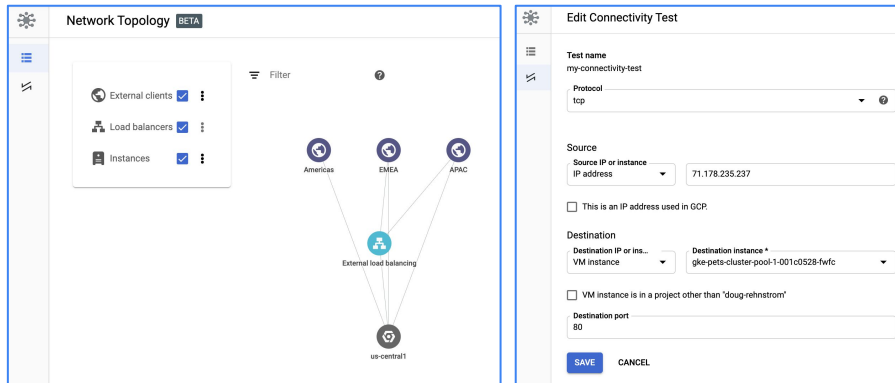
Google Cloud load balancer types and capabilities

HTTP(S) Load Balancing	TCP Load Balancing	UDP Load Balancing
Layer 7 load balancing for HTTP and HTTPS applications Learn more	Layer 4 load balancing or proxy for applications that rely on TCP/SSL protocol Learn more	Layer 4 load balancing for applications that rely on UDP protocol Learn more
Configure HTTP LB HTTPS LB (includes HTTP/2 LB)	Configure TCP LB SSL Proxy TCP Proxy	Configure UDP LB
Options Internet-facing or internal Single or multi-region	Options Internet-facing or internal Single or multi-region	Options Internet-facing or internal Single-region
Start configuration	Start configuration	Start configuration

Let me recap this discussion on Google Cloud load balancers and summarize the types and their capabilities. At a high level, load balancers can work with internal or external IP addresses. We refer to external IP addresses as *internet-facing*. The load balancers can be regional or multi-regional, and finally, they support different traffic types: HTTP, TCP, and UDP. Let's review these, starting with traffic type.

- HTTP(S) load balancing is a layer 7 load balancer. Support is provided for HTTP and HTTPS including HTTP/2. The load balancing supports both internet-facing and internal load balancing, as well as regional or global.
- TCP load balancing provides layer 4 balancing or proxy for applications that require the TCP/SSL protocol. You can configure a TCP load balancer or a TCP or SSL proxy. TCP load balancing supports both internet-facing and internal load balancing as well as regional and global.
- UDP load balancing is for those applications that rely on UDP as a protocol. The UDP load balancer supports both internet-facing and internal load balancing but only regional traffic.

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



As part of our discussion of 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 networks 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 when 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 source and destination endpoints in your Virtual Private Cloud (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?
<i>account</i>	<i>Internal only</i>		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.

Service	HTTP	TCP	UDP
Account		X	

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	Internal 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.

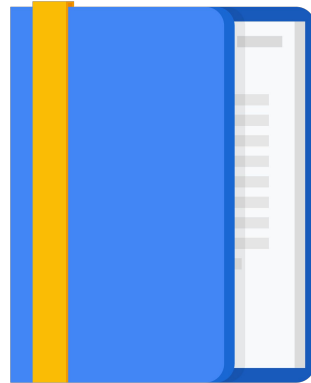
Service	HTTP	TCP	UDP
Search	X		
Inventory		X	
Analytics	X		
Web UI	X		
Orders		X	

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

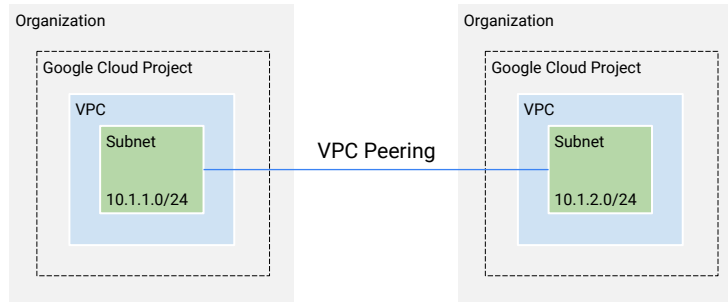
Connecting Networks



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

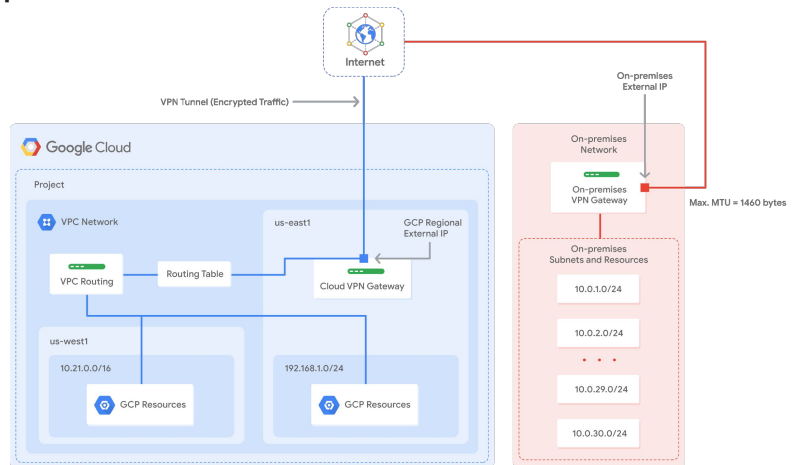


If you're trying to connect two VPC networks, you might want to consider VPC peering. VPC Peering allows private RFC 1918 connectivity across two VPC networks, regardless of whether they belong to the same project or the same organization. Now, remember that each VPC network will have firewall rules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

Use Cloud VPN to connect a Google Cloud network to a network on-premises or in another cloud

- 99.9% SLA
- For low-volume data connections
- Can configure static or dynamic routes using BGP (Border Gateway Protocol)



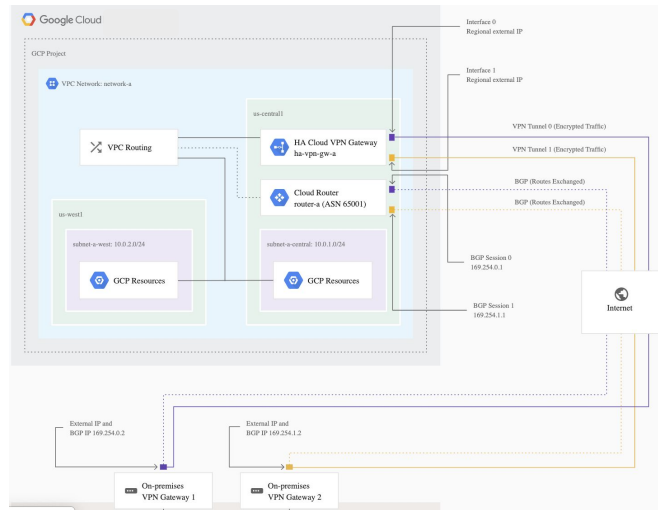
If you're trying to connect a VPC network with an on-premises network or with another cloud network, you might want to consider Cloud VPN. Cloud VPN securely connects two networks through an IPsec VPN tunnel. Traffic traveling between the two networks is encrypted by one VPN gateway, then decrypted by 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, specifically up to 3 Gbps.

This diagram shows a simple 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).

This VPN setup is referred to as Classic VPN and it has a 99.9% monthly uptime SLA. Classic VPN gateways have a single interface, a single external IP address, and support tunnels using static routing or dynamic routing like BGP.

High availability VPN ensures 99.99% availability

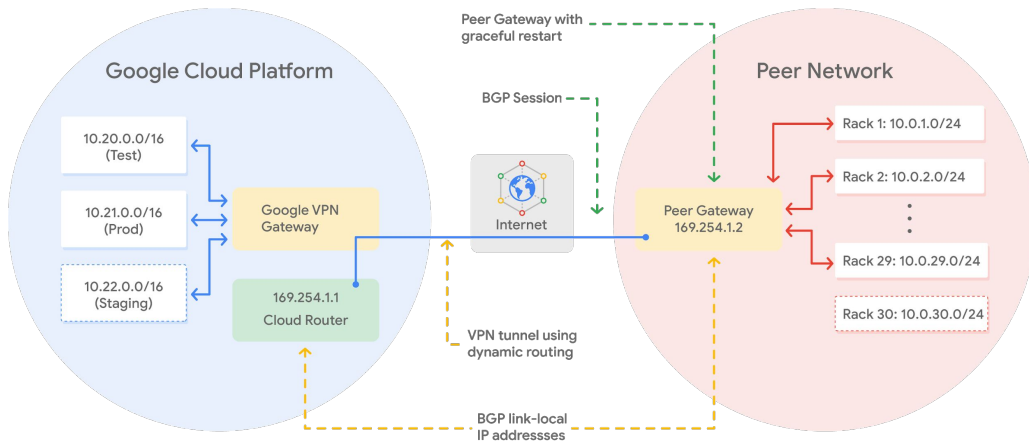
- VPN gateway has 2 network interfaces.
- Creates two IP addresses.
- Each gateway supports multiple VPN tunnels.



In order to ensure a 99.99% monthly uptime SLA, you can also configure a high-availability (HA) VPN.

For a HA VPN connection, two network interfaces and two external IP addresses are required on-premises, as illustrated on this slide. In this topology, one HA Cloud VPN gateway connects to two peer devices. Each peer device has one interface and one public IP address. The HA VPN gateway uses two tunnels: one tunnel to each peer device. This protects against failure of one device and also allows upgrade of a device individually.

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

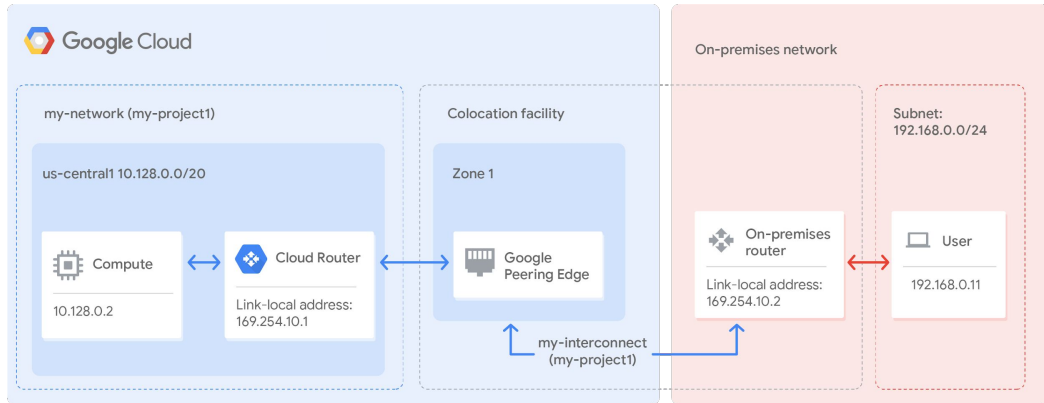
- Dedicated Interconnect provides a direct connection to a colocation facility.
 - From 10 to 200 Gbps
- Partner Interconnect provides a connection through a service provider.
 - Can purchase less bandwidth from 50 Mbps
- Allows access to VPC resources using internal IP address space.
- Private Google 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 10Gbps or 100Gbps circuits, and a dedicated connection can bundle up to eight 10Gbs connections or two 100Gbps for a maximum of 200Gbps. Partner Interconnect provides a connection through a service provider. This can be useful for lower bandwidth requirements starting from 50Mbps. In both cases, Cloud Interconnect allows access to VPC resources using an internal IP address space.

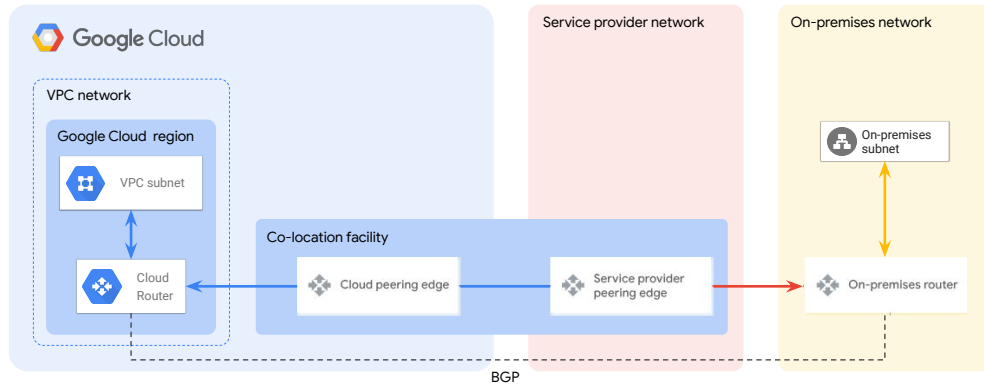
You can even configure Private Google Access for on-premises hosts to allow them to access Google services using private IP addresses.

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.

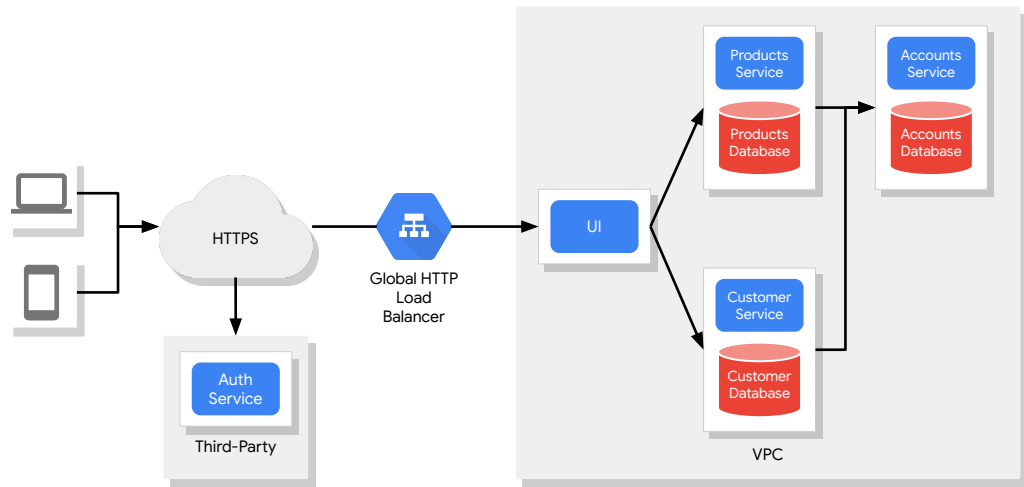
Activity 9: Diagramming your network

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.

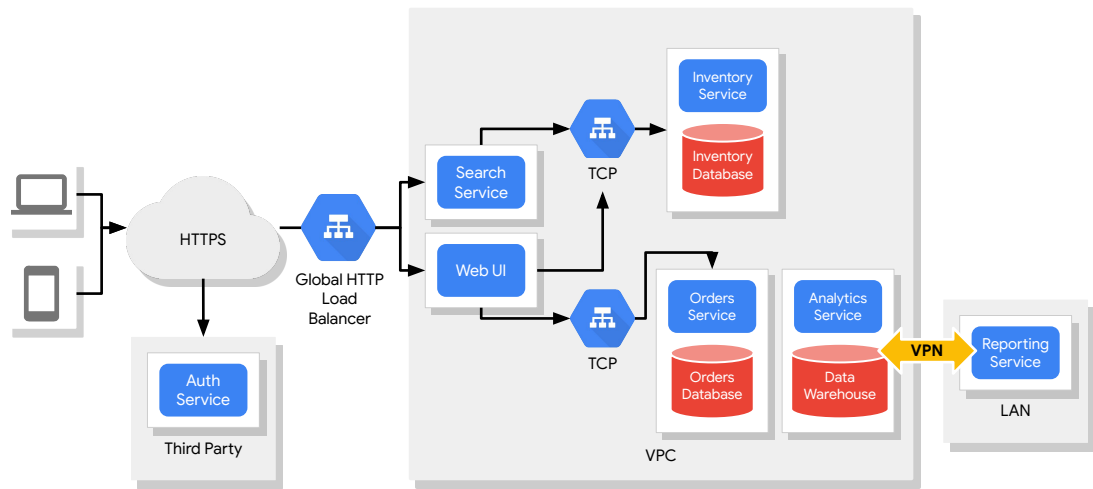
Refer to activity 9 in your workbook to create a similar network diagram for your services.

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.