Welcome to the Virtual Machines and Networks in the Cloud module

Learn how to...

Create virtual private networks

Control access with firewall rules

Create and manage virtual machines

Connect source environment to
Your virtual private cloud

In this module, you will learn how to create a Virtual Private Cloud, your network hosted on Google Cloud's infrastructure. You will also learn how to control access to your network with firewall rules and how to create subnets. You will then learn how to create and manage Virtual Machines in Compute Engine, choose the right configuration, and understand Compute Engine's pricing model. Lastly, you will learn how to create a connection between your source environment and your Virtual Private Cloud.

Agenda

Regions and Zones

Virtual Private Cloud (VPC) Network

Lab

Compute Engine Virtual Machines

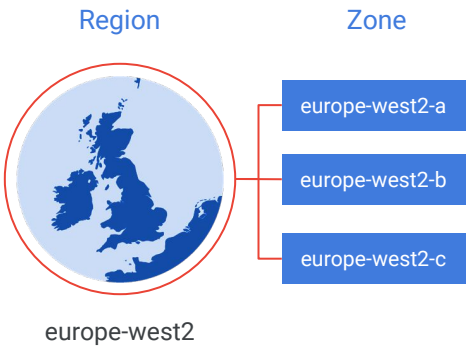
Persistent Disks and NICs

Lab

Interconnecting Networks

In this video, you will learn about Google Cloud's physical geographic distribution.

Google Cloud is organized into regions and zones

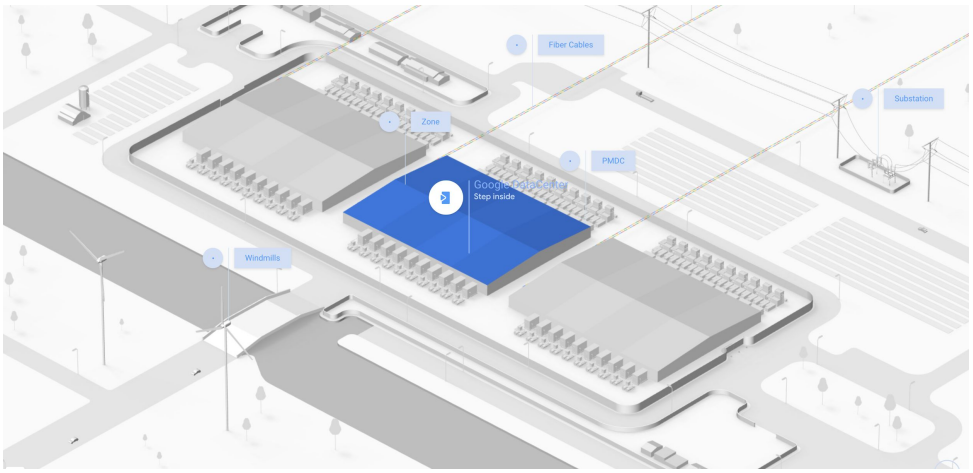


Regions are independent geographic areas that consist of zones. A region is usually referred to by a continent, a cardinal direction and a number. for example, Europe West 2 is the region in London.

A Zone is a single, physical data center. Most regions have 3 or more zones to ensure redundancy.

A regional resource like an external IP address is available to all the zones in a region and benefits from a higher degree of redundancy.

Spreading resources across zones in a region



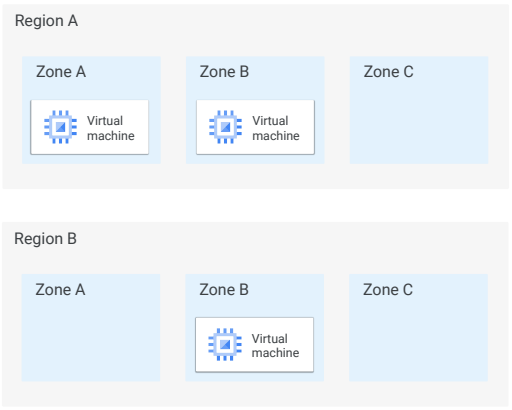
A zone should be considered a single failure domain within a region. Google designs zones to be independent from each other: A zone has power, cooling, networking, and control planes that are isolated from other zones, which ensures a higher level of resiliency.

Putting resources in different zones in a region provides isolation from most types of physical infrastructure and control plane failures. Putting resources in different regions provides an even higher degree of failure independence.

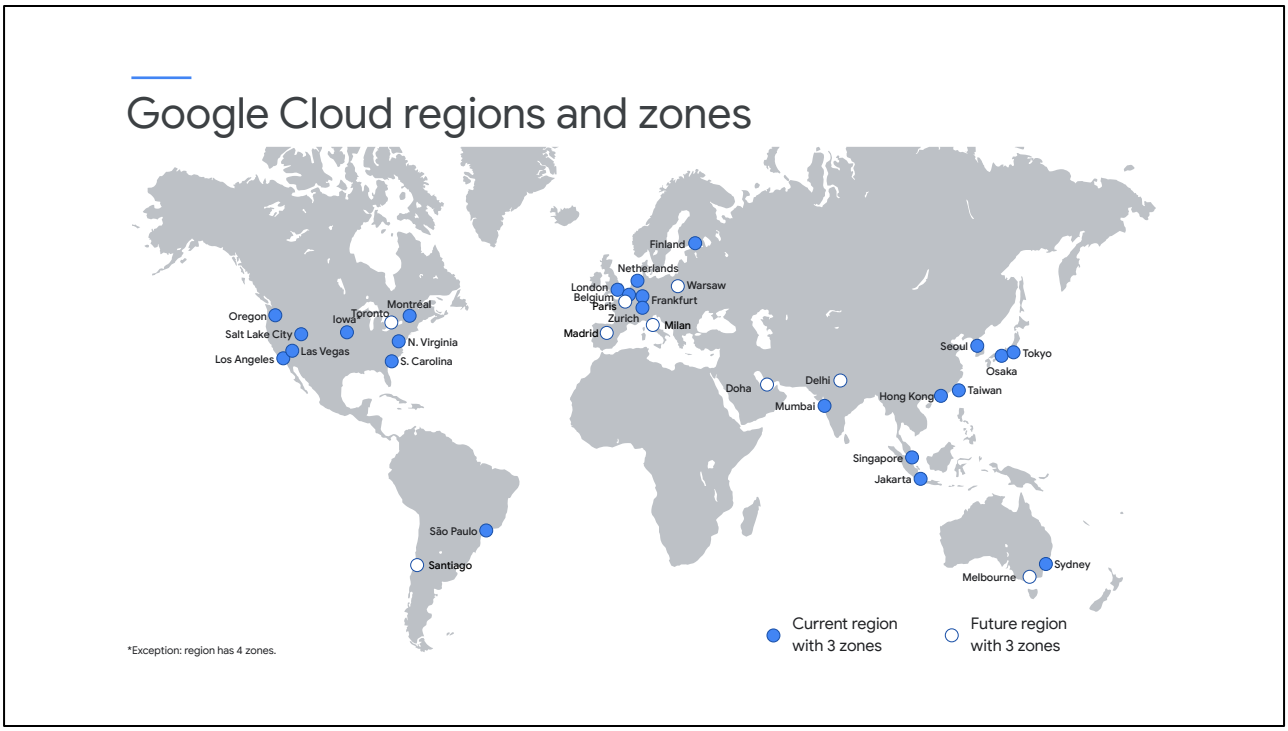
This allows you to design robust systems with resources spread across different failure domains. Because zones have high-bandwidth, low-latency network connections to other zones, you are free to create a highly resilient topography with minimal compromise on performance.

An example of a zonal resource is a Compute Engine virtual machine.

Regions and zones



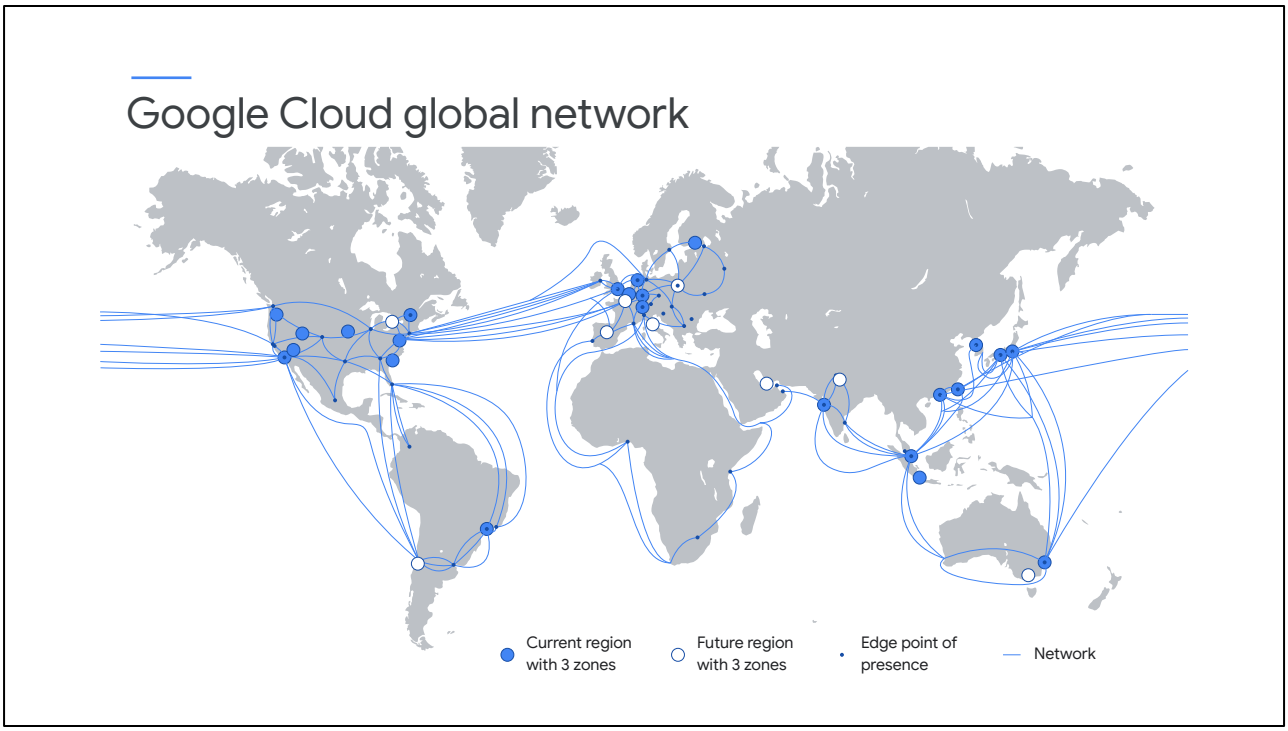
In this example, we have 3 virtual machines--2 in Region A and one in Region B--that all service end user traffic. This design ensures a higher level of resiliency because the failure domains are geographically separated. Remember that zones are a collection of data centers grouped together in a specific geographic area. Spreading your frontend virtual machines across different regions provides better resilience and also better coverage, because region B might be physically closer to your users, and therefore might reduce latency.



Google Cloud services are globally distributed across North America, South America, Europe, Asia, and Australia. These locations are divided into regions and zones. You can choose where to locate your applications to meet your latency, availability, and durability requirements. Since you only pay for what you consume, you have access to a globally distributed infrastructure without paying for the upfront investment. Google Cloud resources are designed to leverage their geographical distribution to create resilient and scalable solutions, such as our global network..

Google has built a large, specialized data network to link all of its data centers together so that content can be replicated or travel across multiple sites, and services can be delivered closest to the end user.

It is designed from the ground up to give customers high speed throughput and reliably low latency for their applications.



The network infrastructure is composed of edge points of presence, which are where Google's network connects to the rest of the internet. Google Cloud can bring its traffic closer to its users because it operates an extensive global network of interconnection points. This reduces costs and provides users with a better experience.

In this illustration, the blue lines represent the private, submarine fiber optic cables that connect all the resources across the globe.

Agenda

Regions and Zones

Virtual Private Cloud (VPC) Network

Lab

Compute Engine Virtual Machines

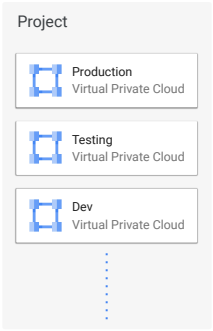
Persistent Disks and NICs

Lab

Interconnecting Networks

In this video, you will learn how to leverage Google's physical infrastructure by creating and configuring your own virtual private cloud network.

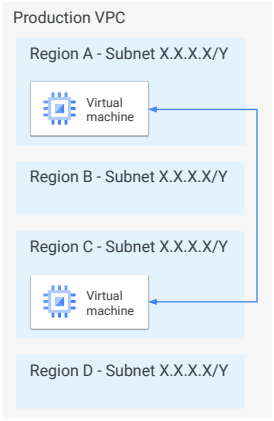
Projects and Virtual Private Cloud (VPC) network



Google Cloud Projects are a global compartment that encompases services and resources under a single administrative unit. It is also where you associate billing, control your expenditure with quotas, and enable APIs.

Each project comes with a default Virtual Private Cloud, or VPC, which is a global network. If you are looking for a way to segregate your resources under the same administrative unit, you can use more than one VPC in each project.

Projects and Virtual Private Cloud (VPC) network

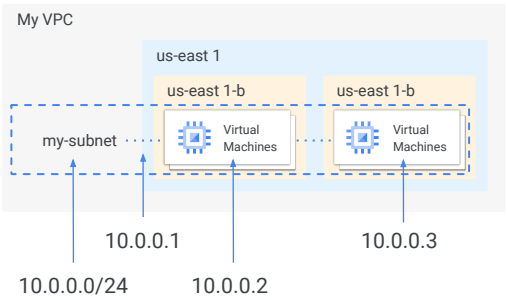


The default VPC is a global network, spanning all available regions across the world that we showed earlier. That provides you with one cloud-based interconnected network that literally exists anywhere in the world—Asia, Europe, Americas—all simultaneously. All resources inside a VPC can communicate over RFC 1918 private IP ranges out of the box and discover one another using a global internal DNS service.

Inside a network, you can segregate your resources with regional subnetworks, which have IP ranges associated with them.

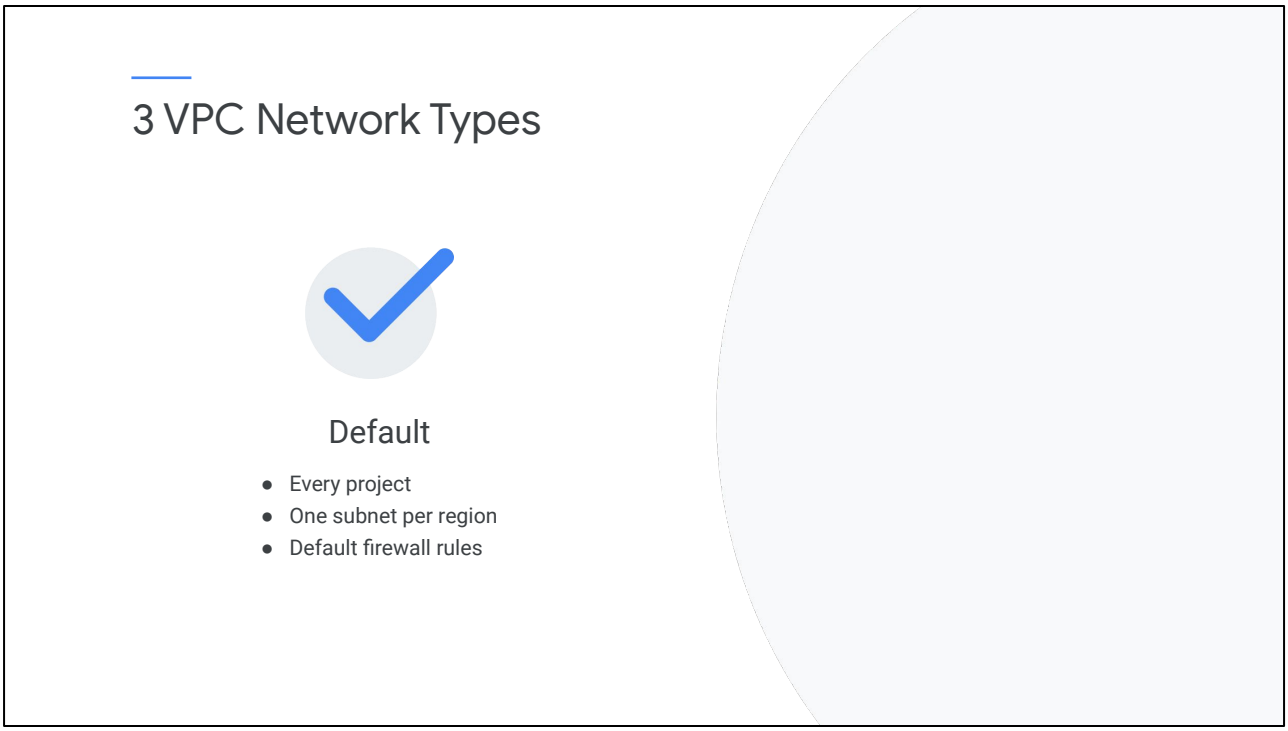
Subnets span the zones that make up a region. That means you can have resources in different zones on the same subnet, which makes management and fault tolerance a lot easier. In this example, the VPC has a subnet in the us-east1 region. The IP range spans across all zones in the region, so both virtual machines you see on the screen are part of the same subnet, despite the fact that they run in different zones.

VPC regional subnets



Notice that the first and second addresses in the range, 10.0.0.0 and 10.0.0.1, are reserved for the network and the subnet's gateway, respectively. This makes the first and second available addresses .2 and .3, which are assigned to the VM instances. The other reserved addresses in every subnet are the second-to-last address in the range and the last address, which is reserved as the "broadcast" address.

To summarize, every subnet has four reserved IP addresses in its primary IP range.



So far you have learned about the default network, which is one of the 3 VPC network types in Google Cloud.

Every project is provisioned with a default VPC network that comes with preset subnets and firewall rules. Specifically, a subnet is allocated for each region with non-overlapping IP address range and firewall rules that allow ingress traffic for ICMP, RDP, and SSH traffic from anywhere, as well as ingress traffic from within the default network for all protocols and ports.

We recommend using the default VPC for prototyping and testing purposes rather than production workloads.

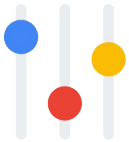
3 VPC Network Types

Auto Mode

- Default network
- One subnet per region
- Regional IP allocation
- Fixed /20 subnetwork per region
- Expandable up to /16

In an auto mode network, one subnet from each region is automatically created. The default network is actually an auto mode network that you can manually add and have greater freedom to modify. These automatically created subnets use a set of predefined IP ranges with a /20 mask that can be expanded to /16. All of these subnets fit within the 10.128.0.0/9 CIDR block. Therefore, as new Google Cloud regions become available, new subnets in those regions are automatically added to auto mode networks using an IP range from that block.

3 VPC Network Types



Custom Mode

- No default subnets created
- Full control of IP ranges
- Regional IP allocation
- Expandable to any RFC 1918 size

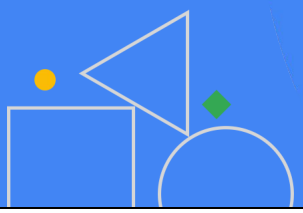
A custom mode network does not automatically create subnets. This type of network provides you with complete control over its subnets and IP ranges. You decide which subnets to create, in regions you choose, and using IP ranges you specify within the RFC 1918 address space. These IP ranges cannot overlap between subnets of the same network. This network is recommended for production because it assumes no implicit trust and gives you maximum control over its layout. It is also the recommended network if you want to interconnect your VPC network with other networks, because you have control over the IP address layout.

You can convert an auto mode network to a custom mode network to take advantage of the control that custom mode networks provide. However, this conversion is one way, meaning that custom mode networks cannot be changed to auto mode networks. So, carefully review the considerations for auto mode networks to help you decide which type of network meets your needs.

Demo

Expand a Subnet

Philipp Maier

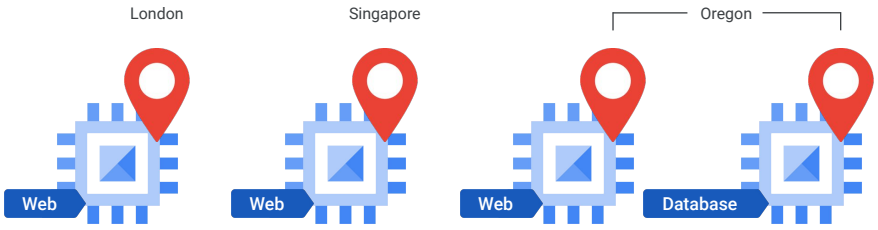


[Presenter]
In this demo you will learn how to expand a subnet within Google Cloud.
We have already created a custom subnet with a /29 mask. A /29 mask provides you with 8 addresses, but of those, 4 are reserved by Google Cloud, which leaves another 4 for your VM instances. Let's try to create another VM instance in this subnet.

[Demo]

[Presenter]
That's how it easy it is to expand a subnet in Google Cloud without any workload shutdown or downtime.

Firewall rules



- VPC network functions as a distributed firewall.
- Can use tags to globally apply rules.

VPCs give you a globally distributed firewall you can utilize to control access to instances, both incoming and outgoing traffic. You can define firewall rules in terms of network tags on virtual machine instances, which makes administration really convenient.

For example, you can tag all your web servers with, say, “WEB,” and write a firewall rule saying that traffic on ports 80 or 443 is allowed into all VMs with the “WEB” tag, no matter what their IP address is or where they are located.

Firewall rules

- Firewall rules are stateful.
- Implied *deny all* ingress and *allow all* egress.

Google Cloud firewall rules are stateful. That means that firewall rules allow bidirectional communication once a session is established.

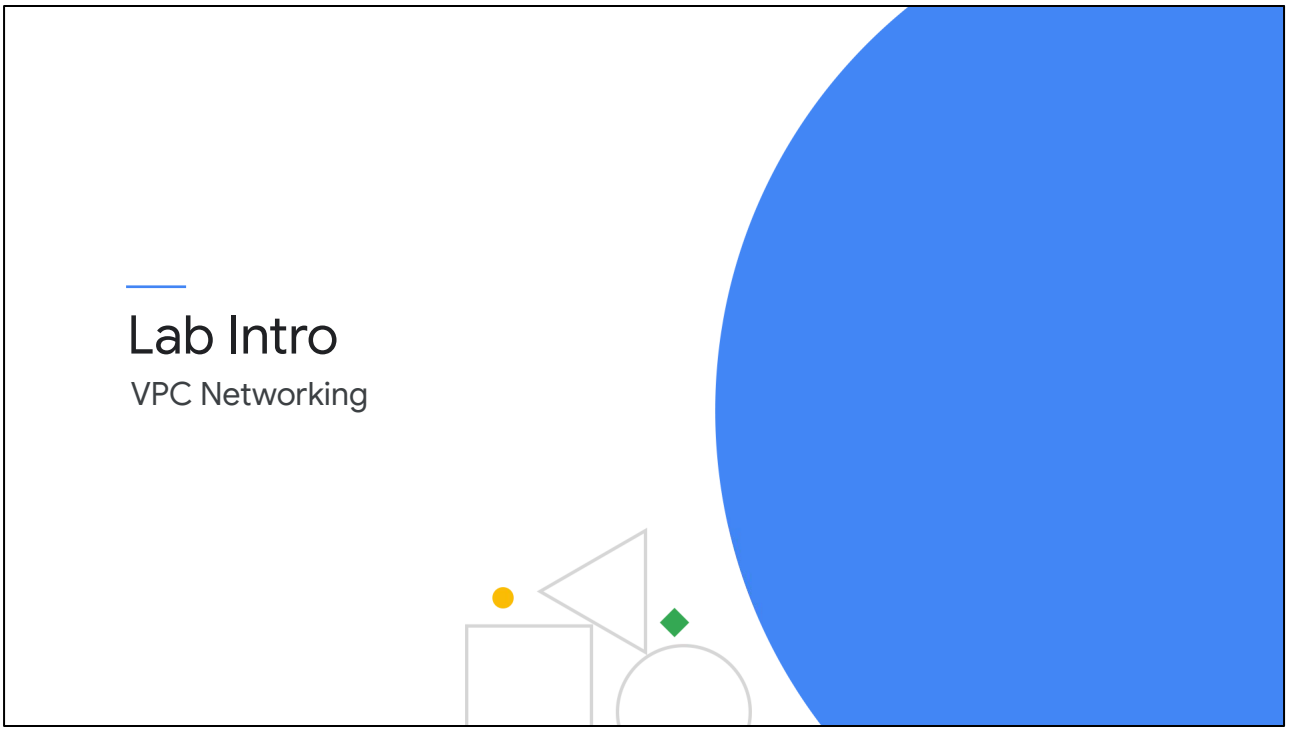
Lastly, it’s important to mention that all networks come with implicit rules. In the absence of firewall rules in a network, there is still an implied "Deny all" ingress rule and an implied "Allow all" egress rule for the network.

A firewall rule is composed of...

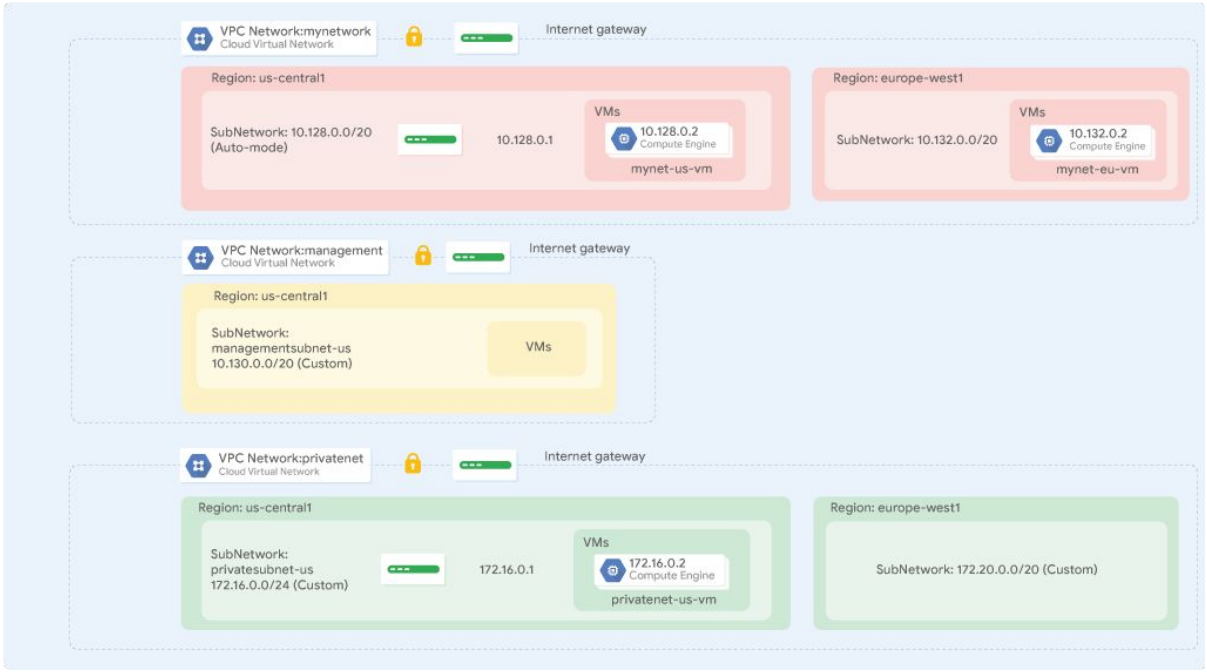
Parameter	Details
Direction	Inbound connections are matched against *ingress* rules only.
	Outbound connections are matched against *egress* rules only.
Source or destination	For the *ingress* direction, *sources* can be specified as part of the rule with IP addresses, source tags, or a source service account.
	For the *egress* direction, *destinations* can be specified as part of the rule with one or more ranges of IP addresses.
Protocol and port	Any rule can be restricted to apply to specific protocols only or specific combinations of protocols and ports only.
Action	To allow or deny packets that match the direction, protocol, port, and source or destination of the rule.
Priority	Governs the order in which rules are evaluated; the first matching rule is applied.

Here are the parameters firewall rules have:

- The **direction** of the rule. Inbound connections are matched against ingress rules only, and outbound connections are matched against egress rules only.
- The **protocol** and **port** of the connection, which can be a single port like 80 or multiple ones like 80 and 443.
- The **priority** of the rule, which governs the order in which rules are evaluated. The lower the number, the more priority the rule has over others.



Let’s apply some of the network features we just discussed in a lab.



In this lab, you create an auto mode VPC network with firewall rules and two VM instances. Then, you convert the auto mode network to a custom mode network and create other custom mode networks, as shown in the network diagram. You also explore the connectivity across networks.

In this lab, you explored the default network and determined that you cannot create VM instances without a VPC network. So you created a new auto mode VPC network with subnets, routes, firewall rules, and two VM instances and tested the connectivity for the VM instances. Because auto mode networks aren't recommended for production, you converted the auto mode network to a custom mode network.

Next, you created two more custom VPC networks with firewall rules and VM instances using the Cloud Console and the gcloud command line. Then you tested the connectivity across VPC networks, which worked when you pinged external IP addresses, but not when you pinged internal IP addresses.

VPC networks are by default isolated private networking domains. Therefore, no internal IP address communication is allowed between networks unless you set up mechanisms such as VPC peering or a VPN connection.

Agenda

Regions and Zones

Virtual Private Cloud (VPC) Network

Lab

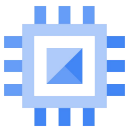
Compute Engine Virtual Machines

Persistent Disks and NICs

Lab

Interconnecting Networks

Compute Engine Features



Compute Engine

- Infrastructure as a Service (IaaS)
- No upfront cost
- Customizable and flexible
- Fast startup time

Google's Compute Engine provides Infrastructure as a Service. That means that Google Cloud manages the physical servers and services for you, providing you with familiar virtual machines without the need to maintain the infrastructure that runs them.

Compute Engine provides you with a lot of configuration options, and you only pay for what you provision with no upfront cost. Essentially, you choose how much memory and how much CPU you want. You choose the type of disk you want, whether you want to just use standard hard drives, SSDs, local SSDs, or a mix. You can even configure the networking interfaces and choose the operating system to run on your virtual machine.

VMs can have internal and external IP addresses

Internal IP

- Allocated from subnet range to VMs by DHCP
- DHCP lease is renewed every 24 hours
- VM name + IP is registered with network-scoped DNS

In Google Cloud, each virtual machine can have two IP addresses assigned. One of them is an internal IP address, which is assigned via the internal DHCP.

When you create a VM in Google Cloud, its symbolic name is registered with an internal DNS service that translates the name to the internal IP address across the global network.

DNS is scoped to the network, so it can translate web URLs and VM names of hosts in the same network, but it can't translate host names from VMs in a different network.

VMs can have internal and external IP addresses

External IP

- Assigned from pool (ephemeral)
- Reserved (static)
- Billed when not attached to a running VM
- VM doesn't know external IP; it is mapped to the internal IP

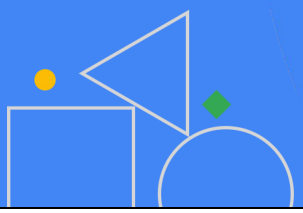
The other IP address is the external IP address, which is optional. You can assign an external IP address if your device or your machine is externally facing. That external IP address can be assigned from a pool, making it ephemeral, or it can be assigned a reserved external IP address, making it static. Remember that you are billed for reserving external IP addresses when they are not attached to a running VM.

For more information about this, see the links section of this video. [https://cloud.google.com/compute/docs/ip-addresses/]

Demo

Internal and External IP

Philipp Maier

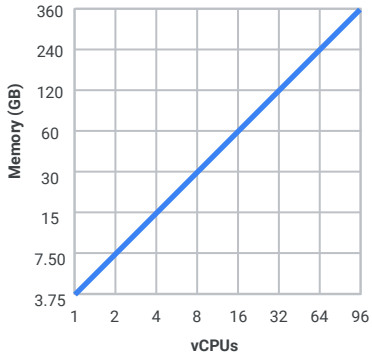


[Presenter]
I just mentioned that VMs can have internal and external IP addresses. Let's explore this in the Cloud Console.

[Demo]

[Presenter]
This demonstrates that every VM needs an internal IP address, but external IP addresses are optional, and by default they are ephemeral.

Standard machine types



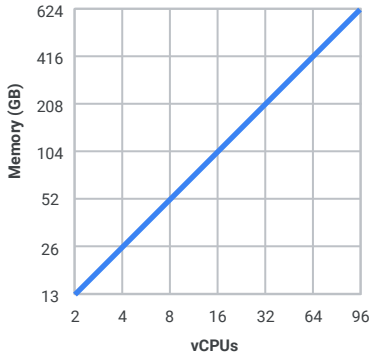
3.75 GB of memory

1 vCPU

Standard machine types are suitable for tasks that have a balance of CPU and memory needs. Standard machine types have 3.75 GB of memory per vCPU. For instance, a 2 vCPU machine will have 7.5 GB of RAM, and a 4 vCPU machine will have 15 GB of RAM.

Each of these machines supports a maximum of more than 100 persistent disks with a total persistent disk size of 64 TB, which is also the case for rest of the machine types.

High-memory machine types

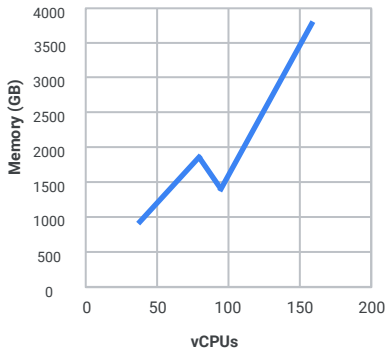


6.5 GB of memory

1 vCPU

High-memory machine types are ideal for tasks that require more memory relative to vCPUs. For example, if your application saves a lot of data in memory for faster access, we recommend choosing this machine family. High-memory machine types have 6.50 GB of system memory per vCPU.

Memory-optimized machine types

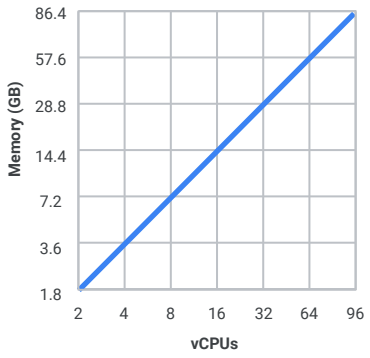


>14 GB of memory

1 vCPU

Memory-optimized machine types are ideal for tasks that require intensive use of memory, with higher memory to vCPU ratios than high-memory machine types. These machine types are perfectly suited for in-memory databases and in-memory analytics, such as SAP HANA and business warehousing workloads, genomics analysis, and SQL analysis services. Memory-optimized machine types have more than 14 GB of memory per vCPU.

High-CPU machine types

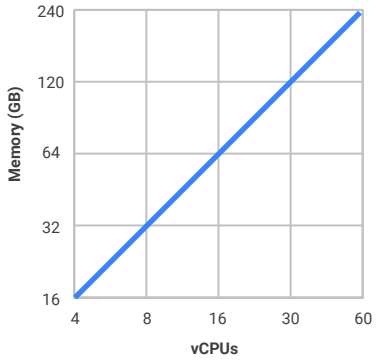


0.9 GB of memory

1 vCPU

For virtual machines that rely more on computational power, choose the High-CPU machine family which comes with more vCPUs relative to memory. High-CPU machine types have 0.90 GB of memory per vCPU. It is important to note that these machines do not have a higher clock rate, but simply have more vCPUs per RAM. If you are looking for more performance per vCPU, the next family, compute-optimized, will be a better choice.

Compute-optimized machine types



Highest performance per vCPU
(up to 3.8Ghz sustained all-core turbo)

Compute-optimized machine types offer the highest performance per core on Compute Engine. Built on the latest-generation Intel Scalable Processors, Cascade Lake, C2 machine types offer up to 3.8 Ghz sustained all-core turbo and provide full transparency into the architecture of the underlying server platforms, enabling advanced performance tuning. C2 machine types offer much more computing power, run on a newer platform, and are generally more robust for compute-intensive workloads than the High-CPU machine types.

Shared-core machine types

	vCPUs	Memory (GB)
f1-micro	0.2	0.60
g1-small	0.5	1.70

Shared-core machine types provide one vCPU that is allowed to run for a portion of the time on a single hardware hyperthread on the host CPU running your instance. Shared-core instances can be more cost-effective for running small, non-resource-intensive applications than other machine types. There are only two shared-core machine types to choose from:

- f1-micro
- g1-small

These machine types offer bursting capabilities that allow instances to use additional physical CPU for short periods of time. Bursting happens automatically when your instance requires more physical CPU than originally allocated. During these spikes, your instance will opportunistically take advantage of available physical CPU in bursts. Note that bursts are not permanent and are only possible periodically.

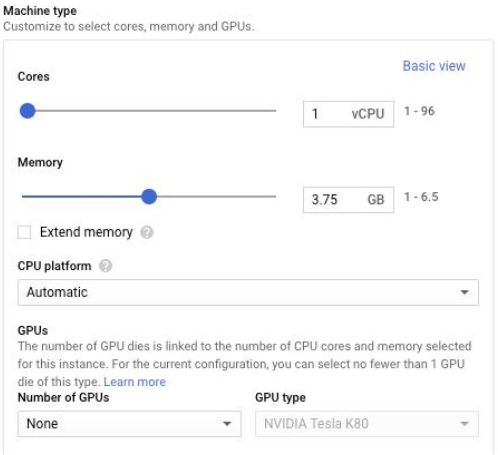
Creating custom machine types

When to select custom:

- Requirements fit between the predefined types
- Need more memory or more CPU

Customize the amount of memory and vCPU for your machine:

- Either 1 vCPU or even number of vCPU
- 0.9 GB per vCPU, up to 6.5 GB per vCPU (default)
- Total memory must be multiple of 256 MB



If none of the predefined machine types match your needs, Google Cloud provides a unique feature called Custom Machine Types where you can independently specify the number of vCPUs and the amount of memory for your instance. Custom Machine Types are ideal for the following scenarios:

- When you have workloads that are not a good fit for the predefined machine types that are available to you.
- When you have workloads that require more processing power or more memory, but don't need all of the upgrades that are provided by the next larger predefined machine type.

Compute Engine Pricing Features

Billing in second increments

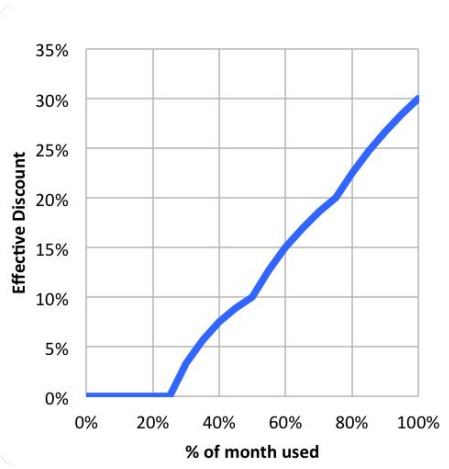
For compute, data processing
and other services

Google Cloud charges the use of your virtual machine per second after the first minute, which means that you only pay for the time your machine was running.

Compute Engine Pricing Features

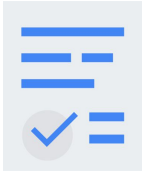
Discounts for sustained use

Automatically applied to virtual machine use over 25% of a month



Sustained use discounts are automatic discounts that you get for running specific Compute Engine resources (vCPUs, memory, GPU devices) for a significant portion of the billing month. For example, when you run one of these resources for more than 25% of a month, Compute Engine automatically gives you a discount for every incremental minute you use for that instance. The discount increases with usage, and you can get up to a 30% net discount for instances that run the entire month.

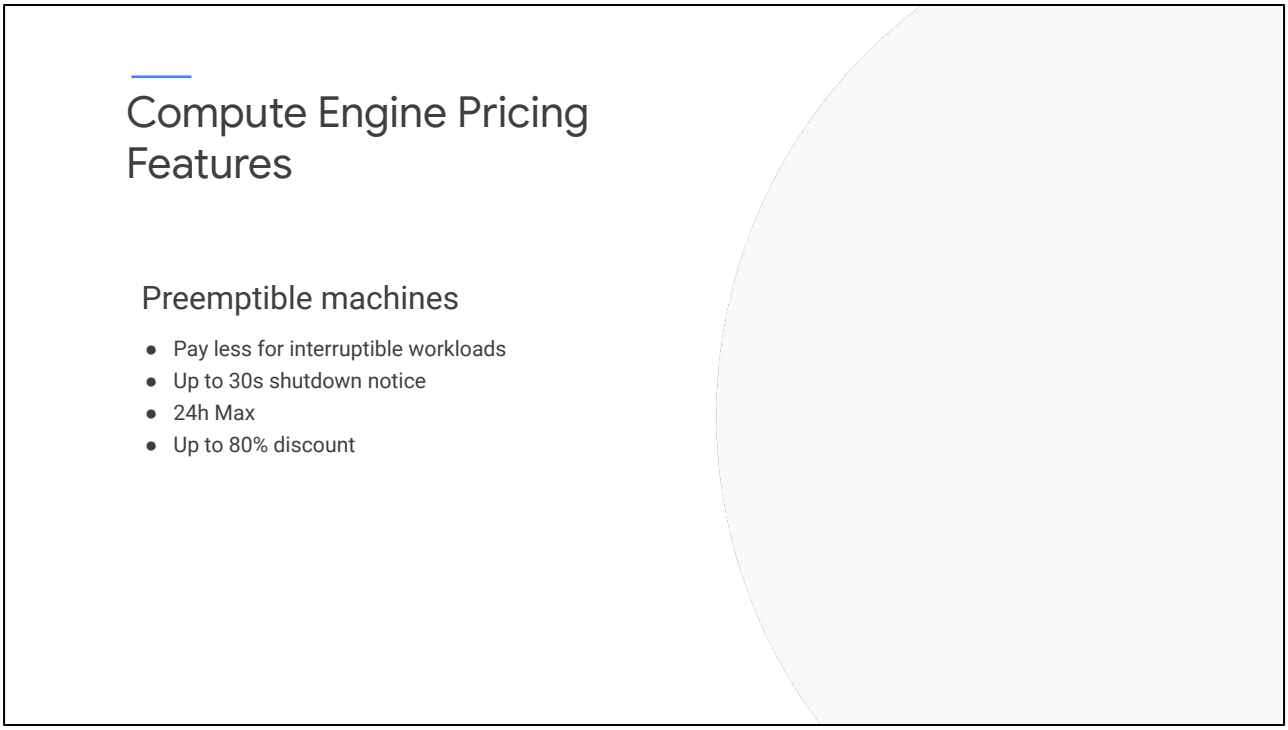
Compute Engine Pricing Features



Discounts for committed use

Pay less for long-term workloads

Compute Engine also offers the ability to purchase committed use contracts in return for higher discounted prices for virtual machine usage. This option resembles the on-premises cost model where most of the compute investment is paid up front. These discounts are known as committed use discounts. If your workload is stable and predictable, you can purchase a specific amount of vCPUs and memory for up to a 57% discount off normal prices in return for committing to a usage term of 1 year or 3 years.



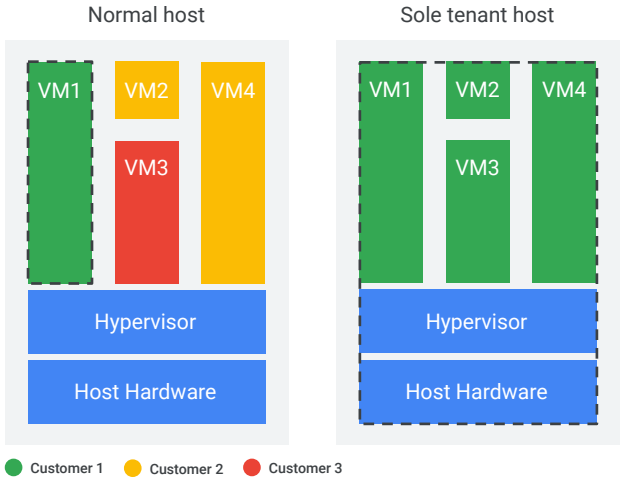
A preemptible virtual machine is an instance that you can create and run at a much lower price than normal instances. The preemptible virtual machine uses excess Compute Engine resources, and therefore its availability varies. Uninterrupted, the virtual machine will operate up to 24 hours and will turn itself off. However, upon increase in demand for the excess resources, the preemptible virtual machine will be terminated within up to 30s.

Preemptible machines are great for non-operational workloads, like batch processes or best effort services.

If you have workloads that require physical isolation from other workloads or virtual machines in order to meet compliance or licensing requirements, consider sole-tenant nodes.

A sole-tenant node is a physical Compute Engine server that is dedicated to hosting VM instances only for your specific project.

Sole-tenant nodes physically isolate workloads



You can bring your own license!

Use sole-tenant nodes to keep your instances physically separated from instances in other projects, or to group your instances together on the same host hardware, for example if you have a payment processing workload that needs to be isolated to meet compliance requirements.

The diagram on the left shows a normal host with multiple VM instances from multiple customers. A sole tenant node as shown on the right also has multiple VM instances, but they all belong to the same project. You can also fill the node with multiple smaller VM instances of various sizes, including custom machine types.

Agenda

Regions and Zones

Virtual Private Cloud (VPC) Network

Lab

Compute Engine Virtual Machines

Persistent Disks and NICs

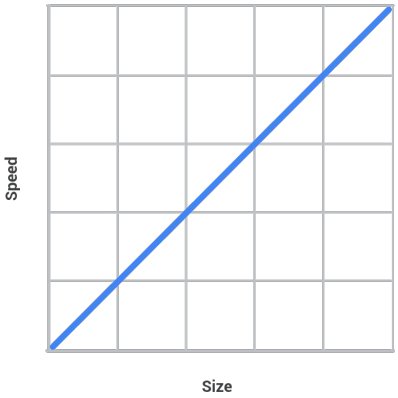
Lab

Interconnecting Networks

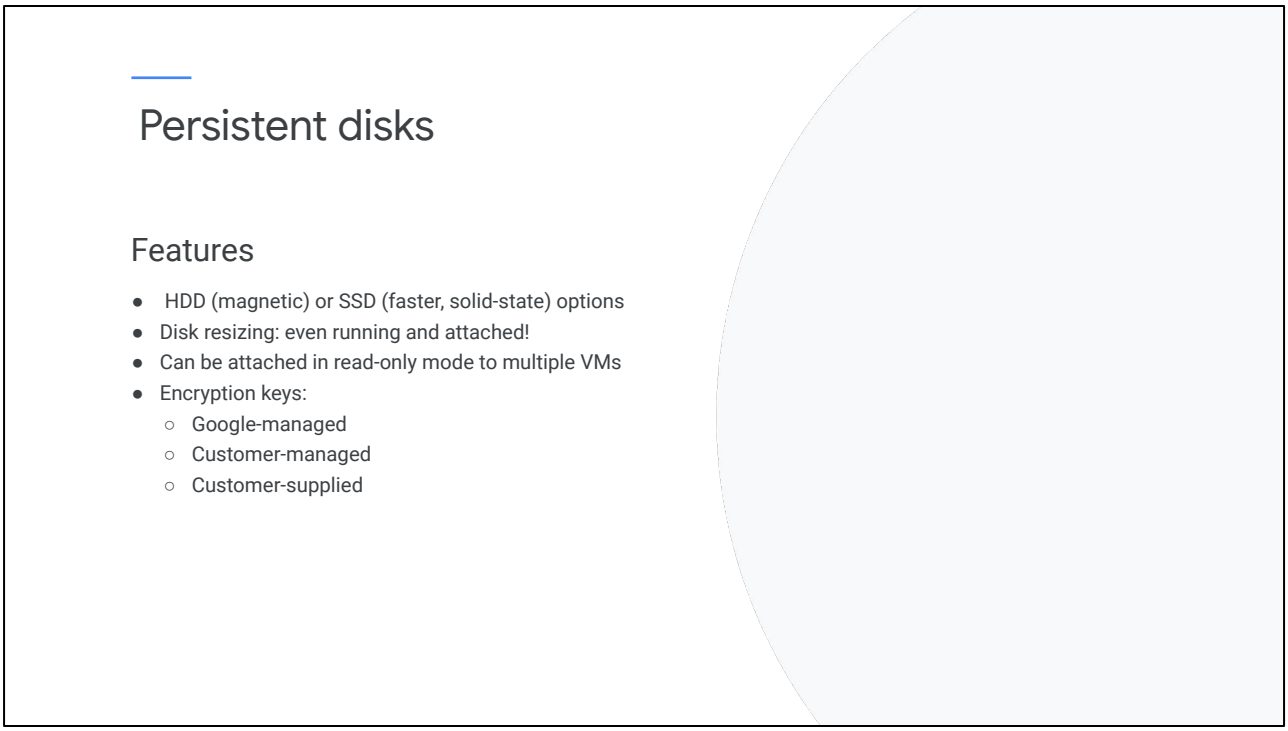
Persistent disks

Network storage appearing as a block device

- Zonal or regional resource
- Durable storage: can survive VM terminate
- Bootable: you can attach to a VM and boot from it
- Performance: Scales with size



Persistent disks resemble iSCSI disks on-premises. Persistent disks are network-attached block storage. As the name implies, your data is persistent, meaning that your data is durable and outlives the machine lifecycle. Persistent disks are a zonal resource, and you can even have a dual zone disk for redundancy. The first persistent disk that a virtual machine attaches to is the boot disk, which is where the operating system exists. You can have more than one persistent disk, and there is a direct correlation between the disk size and its performance. That means that if you need more speed from your persistent disk, you can scale its size to match precisely the speed you need.

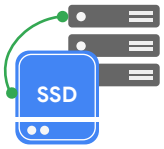


Persistent Disk comes in 2 flavors: HDD and SSD. The choice comes down to cost and performance. HDD is great for longtail files or general bulk data that does not need fast performance, and it is the more economic option per GB. SSD is designed for random reads and writes, and provides better performance for databases.

Another feature of persistent disks is that you can dynamically resize them, even while they are running and attached to a VM, and therefore also benefit from the increase in performance as we saw in the last slide.
You can also attach a disk in read-only mode to multiple VMs. This allows you to share static data between multiple instances, which is cheaper than replicating your data to unique disks for individual instances.

By default, Compute Engine encrypts all data at rest. Google Cloud handles and manages this encryption for you without any additional actions on your part. However, if you want to control and manage this encryption yourself, you can either use Cloud Key Management Service to create and manage key encryption keys (which is known as customer-managed encryption keys) or create and manage your own key encryption keys (known as customer-supplied encryption keys).

Local SSD disks are physically attached to a VM



- More IOPS, lower latency, and higher throughput than persistent disk
- 375-GB disk up to eight, total of 3 TB
- Data survives a reset, but not a VM stop or terminate
- VM-specific: cannot be reattached to a different VM

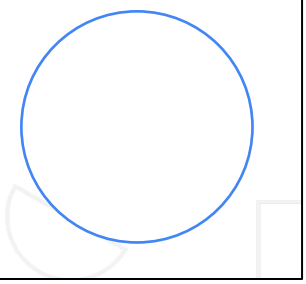
Compute Engine also provides physically attached SSDs, called local SSDs. Because they are locally attached, these disks are considered ephemeral but provide very high IOPS.

Data on these disks will survive a reset but not a VM stop or terminate, because these disks can’t be reattached to a different VM.

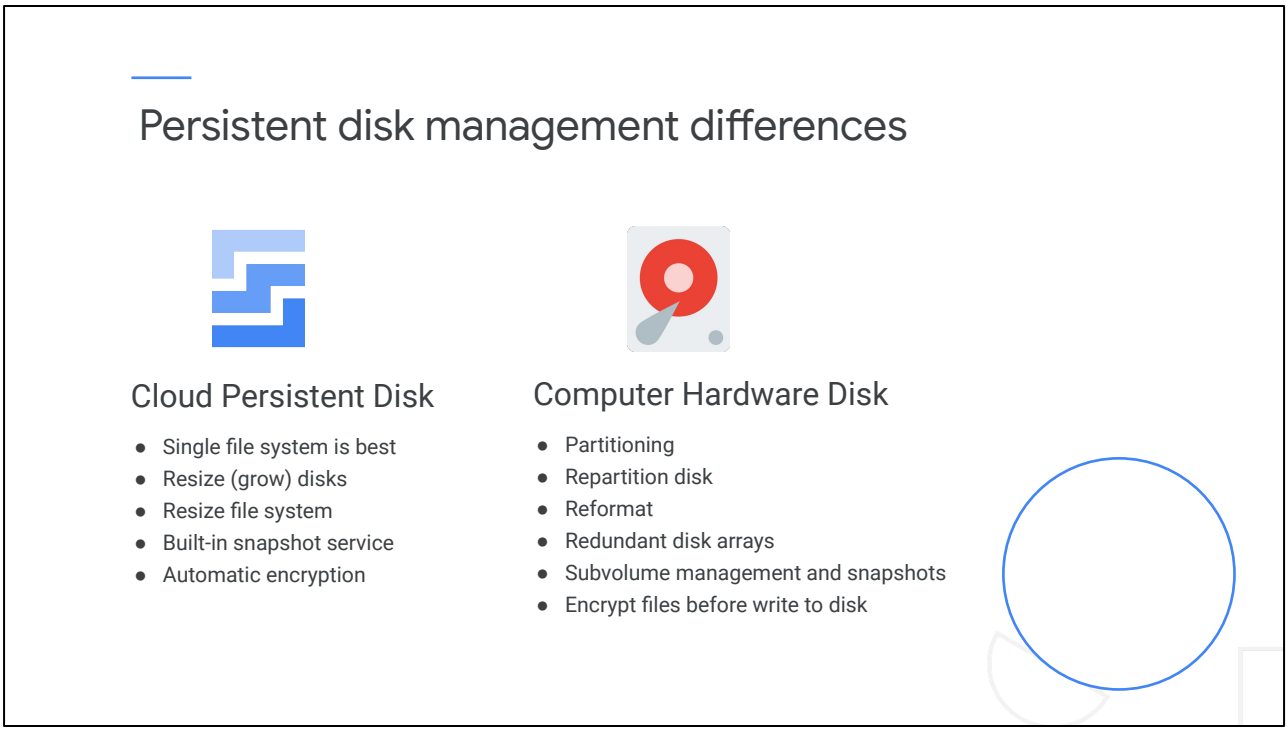
Currently, you can attach up to 8 local SSD disks with 375 GB each, resulting in a total of 3 TB.

Summary of disk options

	Persistent disk HDD	Persistent disk SSD	Local SSD disk
Data redundancy	Yes	Yes	No
Encryption at rest	Yes	Yes	Yes
Bootable	Yes	Yes	No
Use case	General, bulk file storage	Random IOPS	High IOPS and low latency



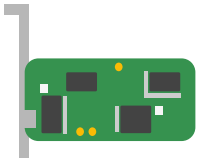
The persistent disks offer data redundancy because the data on each persistent disk is distributed across several physical disks.
We recommend choosing a persistent HDD disk when you need an economic storage solution and performance requirements are relatively low. If you have high performance requirements or your workload relies more heavily on random reads and writes like databases, we recommend the SSD options. For non persistent storage, Local SSDs provide the highest throughput and lowest latency, because they are physically attached to your virtual machine.



There are many differences between a physical hard disk from your on-premises environment and a Compute Engine persistent disk, which is essentially a virtual network attached device.
First of all, if you remember with normal computer hardware disks, you have to partition them. Essentially, you have a drive and you are carving up a section for the operating system to get its own capacity. If you want to grow it, you have to repartition, and if you want to make changes you might even have to reformat. If you want redundancy, you might create a redundant disk array, and if you want encryption, you need to encrypt files before writing them to the disk.

With cloud persistent disks, things are very different because all that management is handled for you on the backend. You can simply grow disks and resize the file system because disks are virtual networked devices. Redundancy and snapshot services are built in and disks are automatically encrypted.

Networking



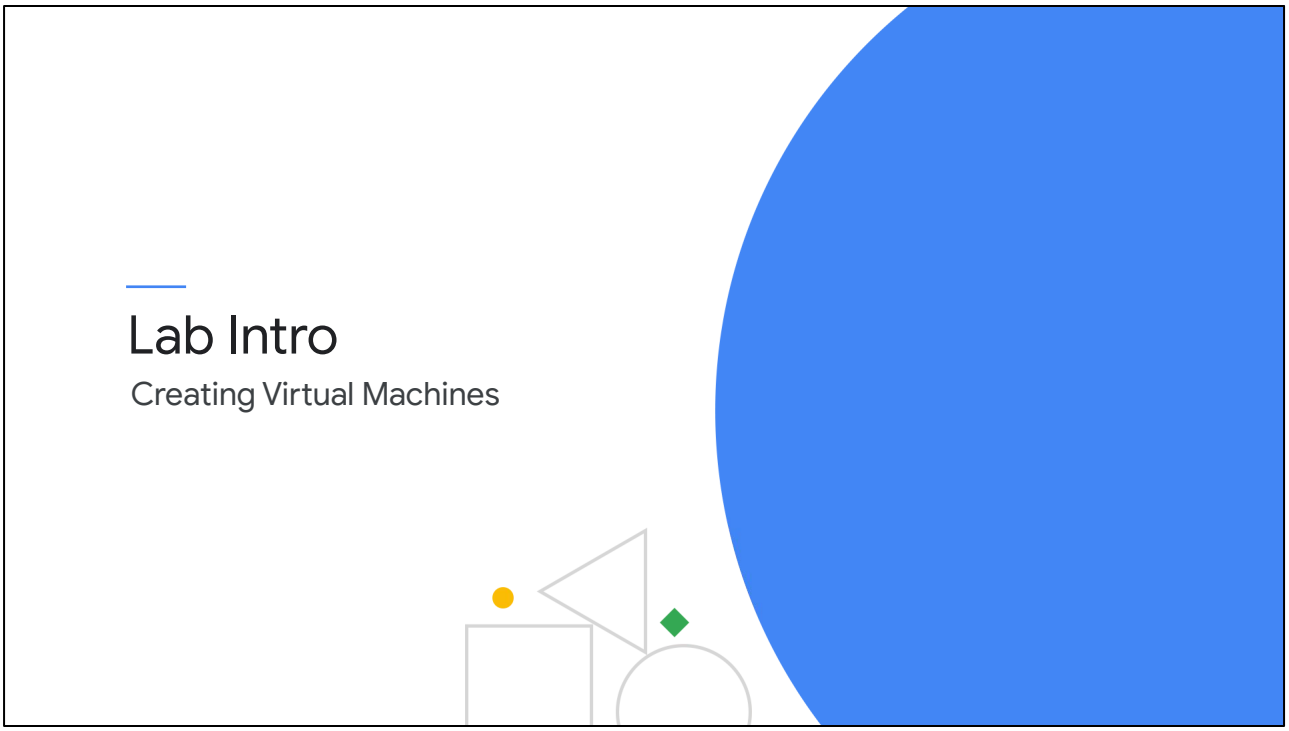
- Network throughput scales at 2 Gbps per vCPU
  - Max throughput of 32 Gbps with 16 vCPU
  - Including persistent disk IOPS throughput
- Up to 8 NIC, connected to different networks
- Can not modify after creation

Each Compute Engine virtual machine comes with a virtual Network Interface Controller, or vNIC. The overall network throughput of a virtual machine scales at 2 Gigabits per second per vCPU, up to 32 Gigabits per second with 16 vCPU cores.

Because persistent disks are accessed over the network instead of physically attached to the virtual machine, they also use the allocated network bandwidth a machine has.

You can have up to 8 NICs, each attached to a different VPC network. For example, you may want to have a network appliance that has one NIC in a DMZ VPC and one in your internal VPC.

One important aspect to remember is that once a virtual machine is created, you cannot make any modifications to the network interfaces. That means that if you want to change the number of NICs, connect them to different networks, or add a NIC, you will have to recreate the VM.



Let’s take some of the Compute Engine concepts that we just discussed and apply them in a lab.

In this lab, you explore virtual machine instance options by creating several standard VMs and a custom VM. You also connect to those VMs using both SSH for Linux machines and RDP for Windows machines.

In this lab, you created several virtual machine instances of different types with different characteristics. Specifically, you created a small utility VM for administration purposes, a Windows VM, and a custom Linux VM. You also accessed both the Windows and Linux VMs and deleted all your created VMs.

In general, start with smaller VMs when you're prototyping solutions to keep the costs down. When you are ready for production, trade up to larger VMs based on capacity. If you're building in redundancy for availability, remember to allocate excess capacity to meet performance requirements. Finally, consider using custom VMs when your application's requirements fit between the features of the standard types.

Agenda

Regions and Zones

Virtual Private Cloud (VPC) Network

Lab

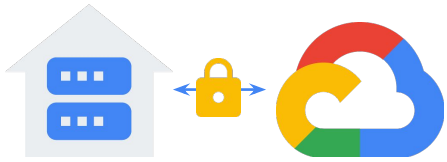
Compute Engine Virtual Machines

Persistent Disks and NICs

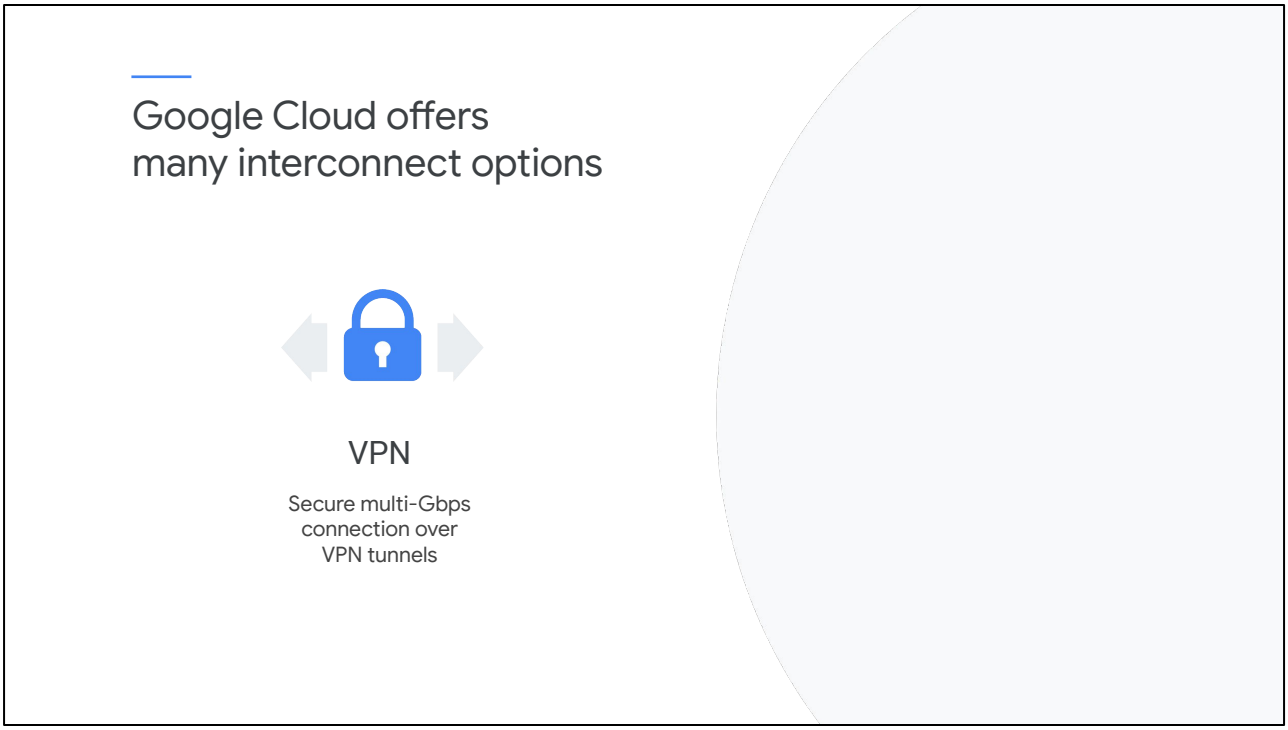
Lab

Interconnecting Networks

Google Cloud offers many interconnect options

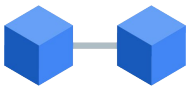


Before starting a migration to Google Cloud, you need to create a secure connection between your on-premises and your VPC. This connection will support your migration process and also enhance the interoperability between your existing on-premises workloads and your cloud environment.



Many customers start with a Virtual Private Network connection over the Internet, using the IPsec protocol. The VPN connection is relatively easy to setup and doesn't require physical connection between your on-premises and a Google Cloud data center.

Google Cloud offers many interconnect options

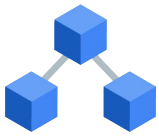


Dedicated Interconnect

Connect up to 8 x 10 Gbps or 2 x 100 Gbps transport circuits for private cloud traffic to Google Cloud at Google POPs

On the other hard, you might not want to use the Internet, either because of security concerns or because you need more reliable bandwidth and lower latency. Google Cloud Interconnect is a Layer 2, private connection to your VPC. If your on premises topology allows it, you can use Dedicated Interconnect, which connects your on premises to one of our data centers directly.

Google Cloud offers many interconnect options

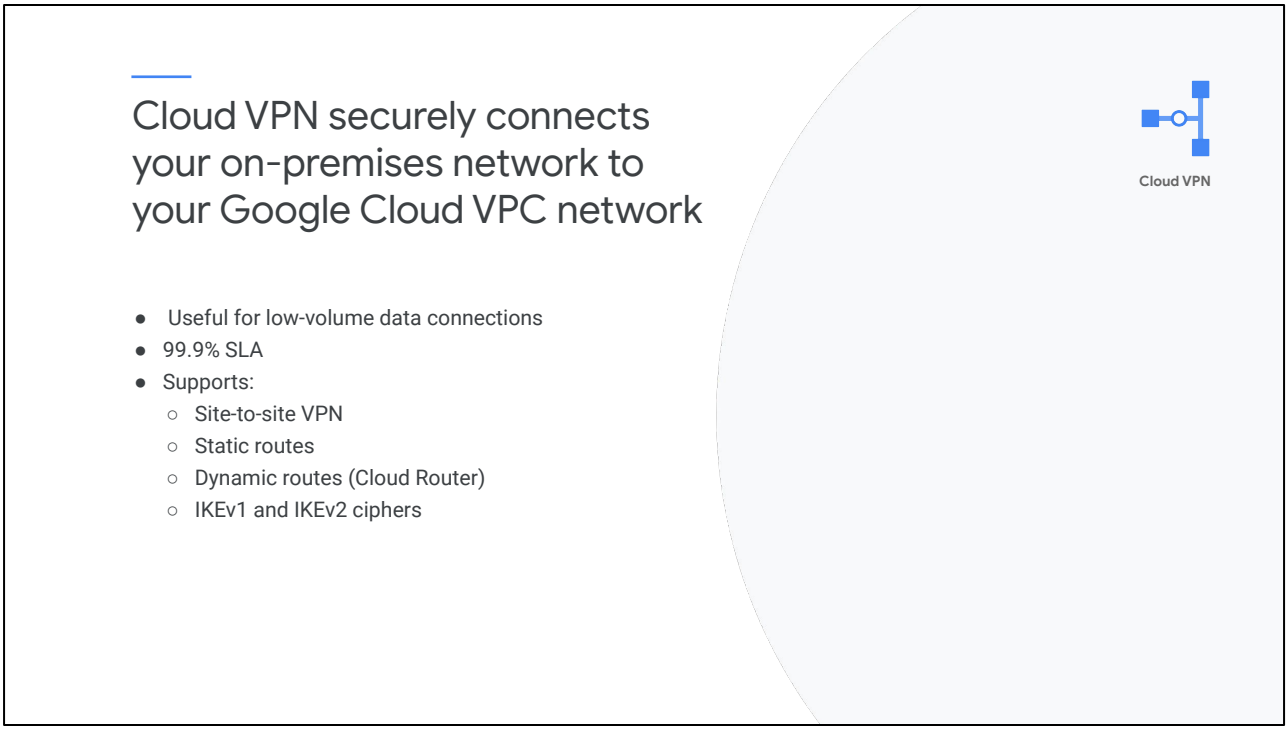


Partner Interconnect

Connectivity between your on-premises network and your VPC network through a supported service provider

Partner Interconnect provides connectivity between your on-premises network and your VPC network through a supported service provider. A Partner Interconnect connection is useful if your data center is in a physical location that can't reach a Dedicated Interconnect colocation facility or if your data needs don't warrant an entire 10 Gbps connection.

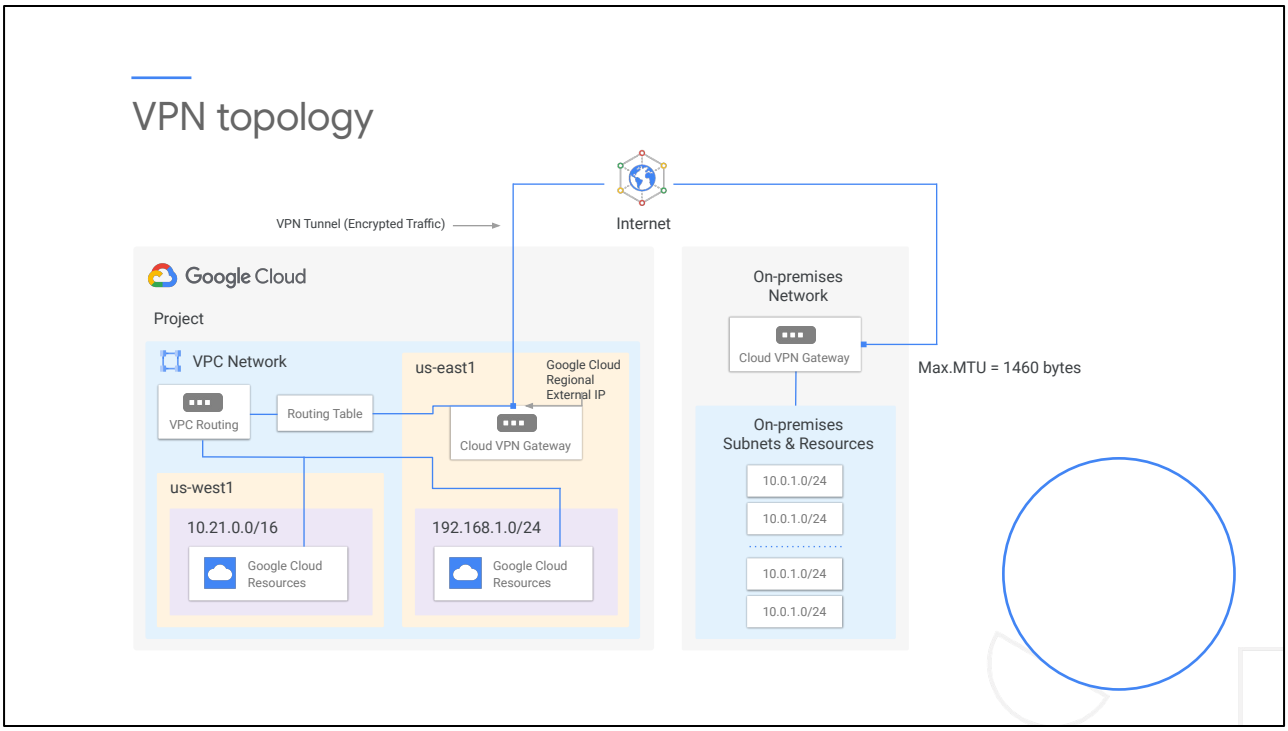
Let’s explore all these options in detail.



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 and 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% service availability and supports the following:

- Site-to-site VPN. It does not support client-to-gateway scenarios. In other words, Cloud VPN doesn't support use cases where client computers need to "dial in" to a VPN using client VPN software.
- Both static routes and dynamic routes to manage traffic between your VM instances and your existing infrastructure. Dynamic routes are configured with Cloud Router, which we will cover briefly.
- Both IKEv1 and IKEv2 ciphers.



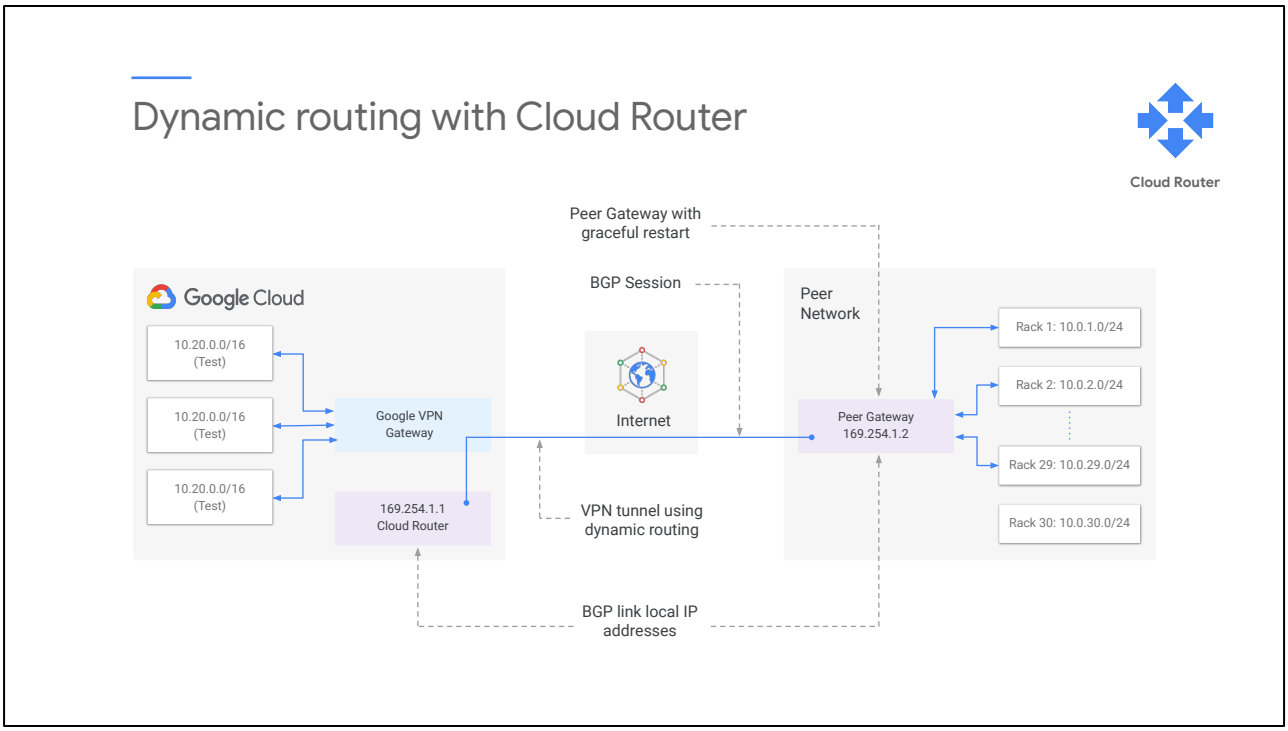
Let’s walk through an example of Cloud VPN. This diagram shows a simple VPN connection between your VPC and on-premises network. Your VPC network has subnets in us-east1 and us-west1, with Google Cloud resources in each of those regions. These resources are able to communicate using their internal IP addresses because routing within a network is automatically configured (assuming that firewall rules allow the communication).

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.

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, see the documentation at https://cloud.google.com/vpn/docs/concepts/mtu-considerations.



We mentioned earlier that Cloud VPN supports both static and dynamic routes. In order to use dynamic routes, you need to configure Cloud Router. 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.

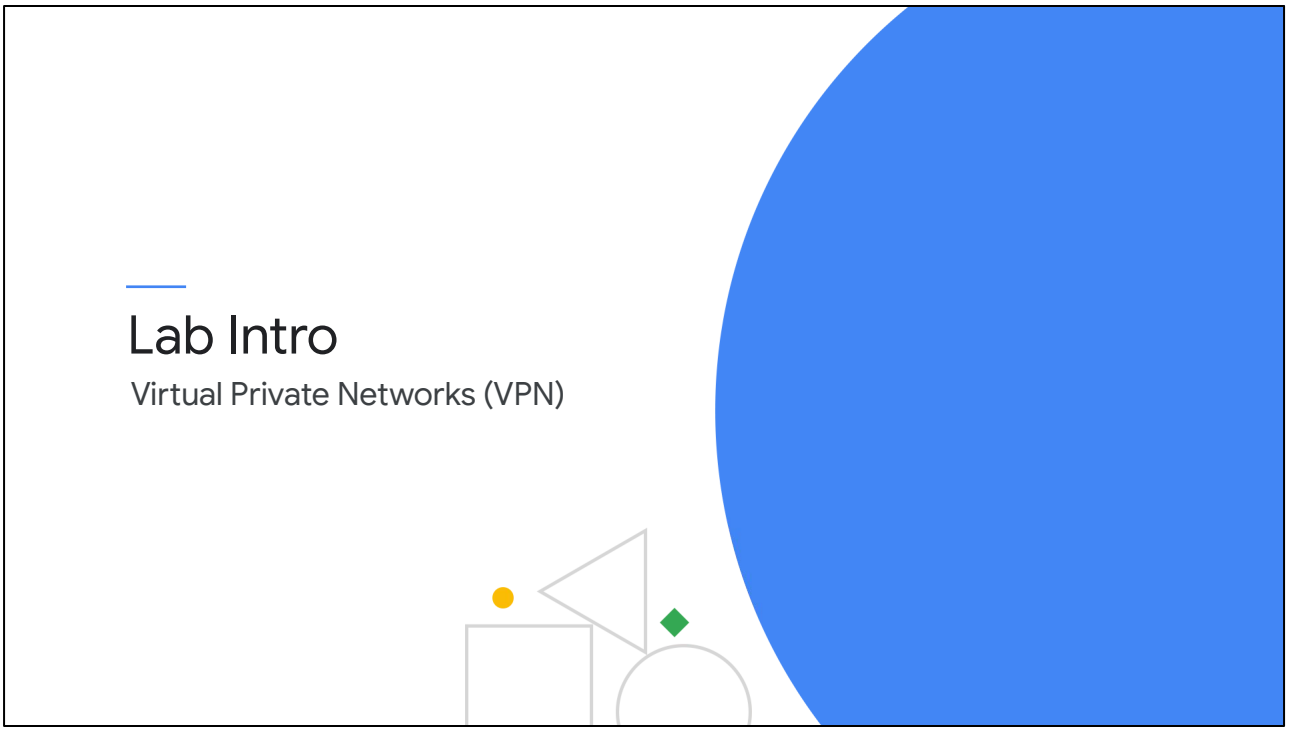
For example, this diagram shows two different regional subnets in a VPC network (Test and Prod) and 29 subnets in the on-premises network. The two networks are connected through Cloud VPN tunnels. How would you handle adding new subnets? For example:

1. A new “Staging” subnet in the Google Cloud network
2. A new on-premises 10.0.30.0/24 subnet to handle growing traffic in your data center

To automatically propagate network configuration changes, the VPN tunnel uses Cloud Router to establish a BGP session between the on-premises VPN gateway, which must support BGP. The new subnets are then seamlessly advertised between networks. This means that instances in the new subnets can start sending and receiving traffic immediately.

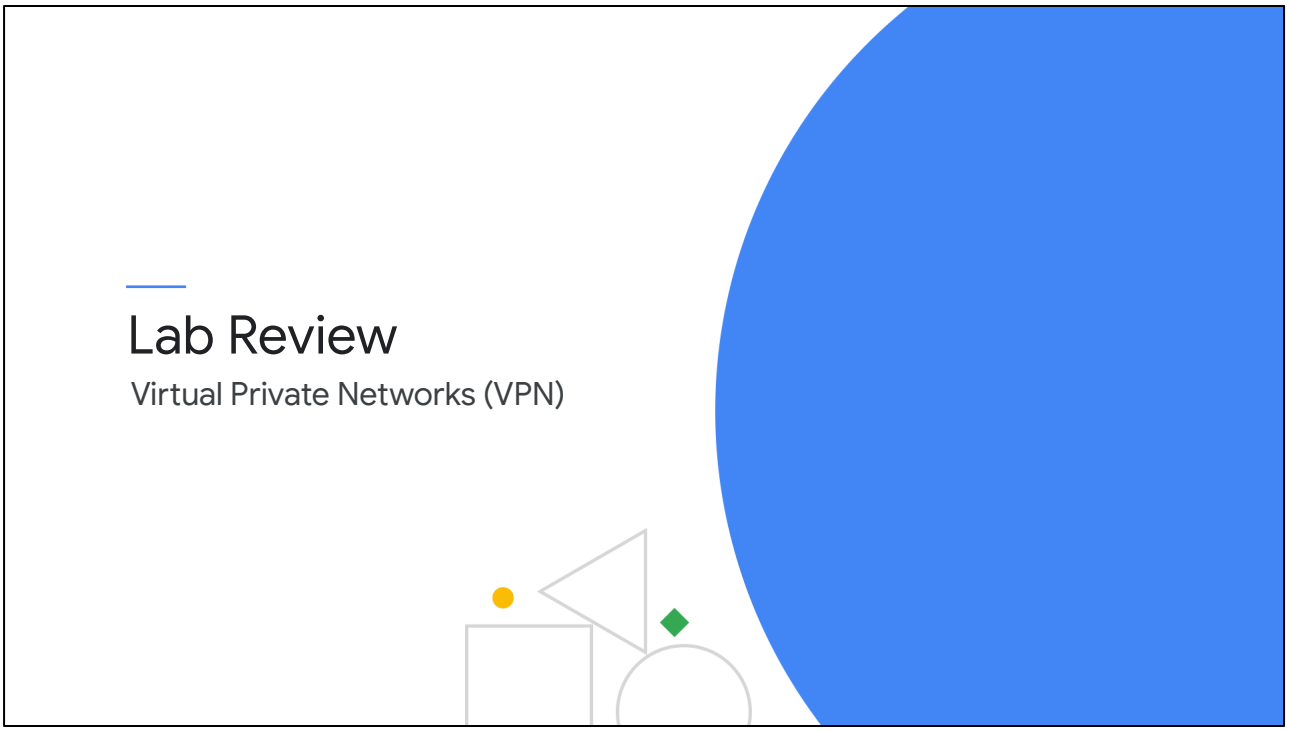
To set up BGP, an additional IP address has to be assigned to each end of the VPN

tunnel. These two IP addresses must be link-local IP addresses, belonging to the IP address range 169.254.0.0/16. These addresses are not part of IP address space of either network and are used exclusively for establishing a BGP session.



Let’s apply what we just covered.

In this lab, you’ll establish VPN tunnels between two networks in separate regions such that a VM in one network can ping a VM in the other network over its internal IP address.

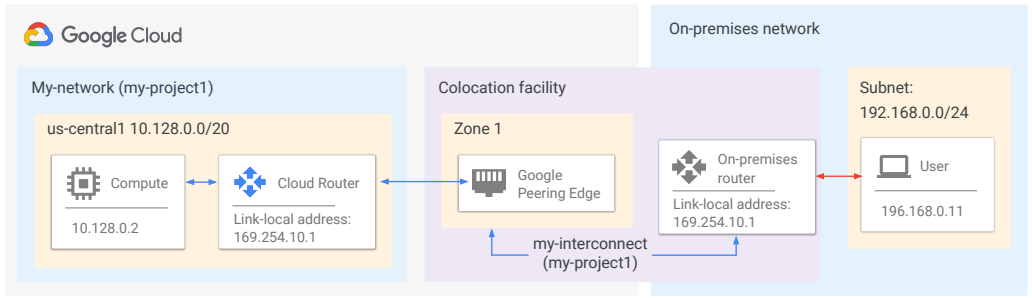


In this lab, you configured a VPN connection between two networks with subnets in different regions. Then you verified the VPN connection by pinging VMs in different networks using their internal IP addresses.

You configured the VPN gateways and tunnels using the Cloud Console. However, this approach obfuscated the creation of forwarding rules, which you explored with the command line button in the Cloud Console. This can help in troubleshooting a configuration.

You can stay for a lab walkthrough, but remember that Google Cloud's user interface can change, so your environment might look slightly different.

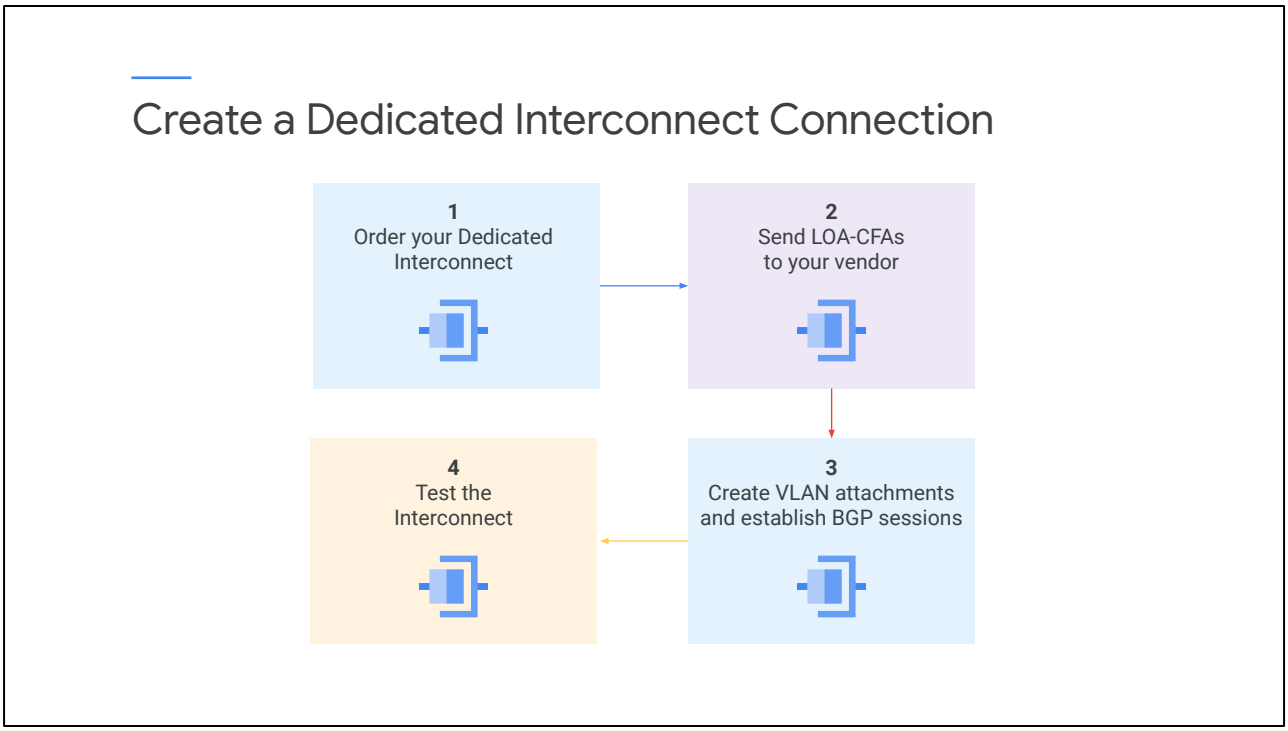
Dedicated Interconnect provides direct physical connections



Dedicated Interconnect provides direct physical connections between your on-premises network and Google's network. This enables you to transfer large amounts of data between networks, which can be more cost-effective than purchasing additional bandwidth over the public internet.

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.

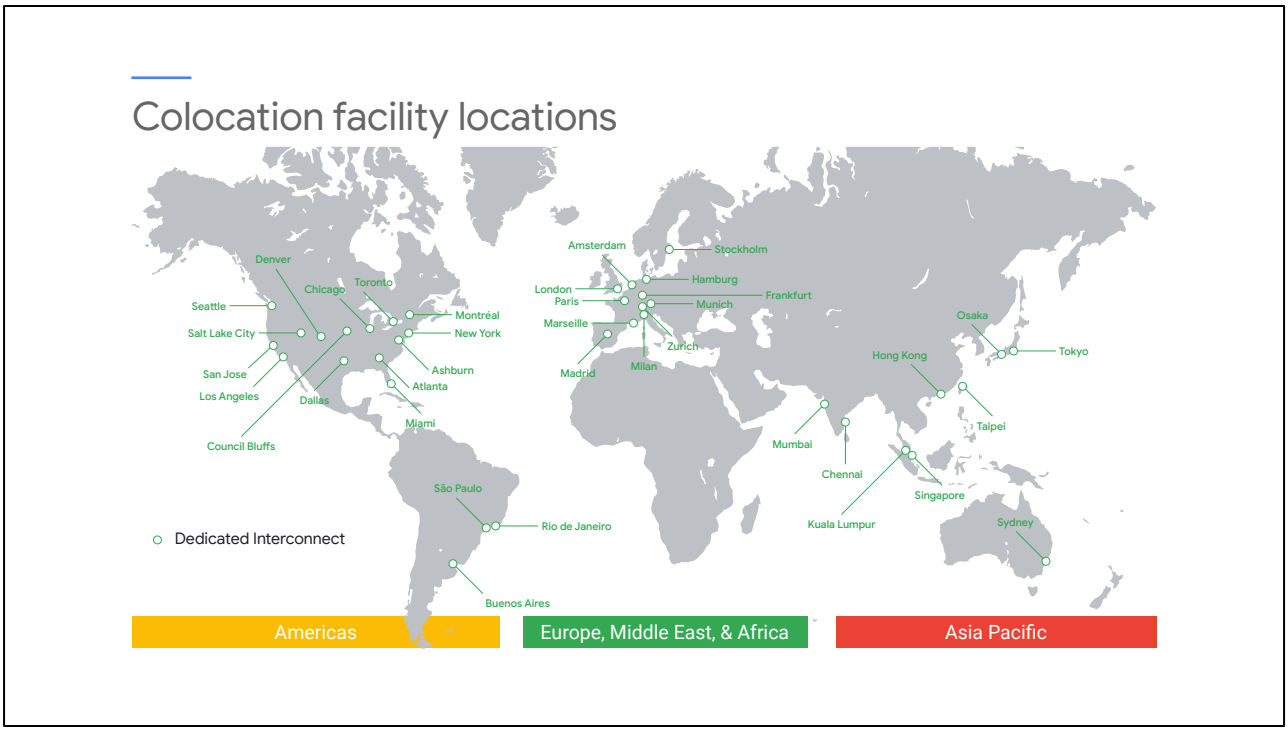
Dedicated Interconnect can be configured to offer a 99.9% or a 99.99% uptime SLA.



Creating a Dedicated Interconnect Connection is as simple as these 4 steps:

1. Order your Dedicated Interconnect.
2. Send LOA-CFAs to your vendor.
3. Create VLAN attachments and establish BGP sessions.
4. Test the Interconnect.

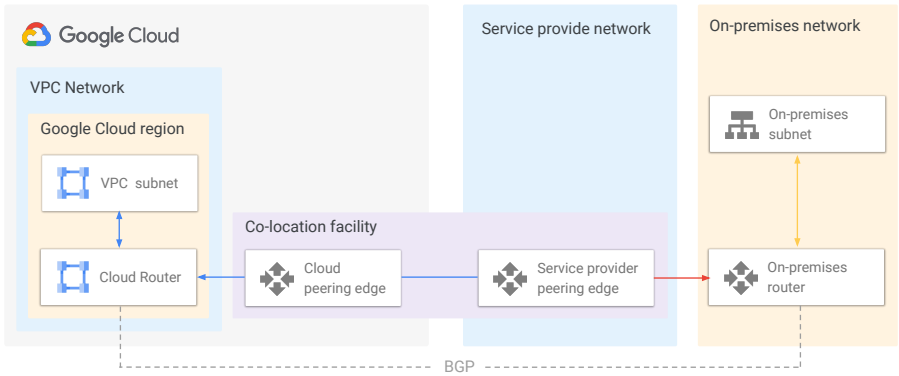
For a demo on how to create a Dedicated Interconnect, refer to the link in the slides: https://storage.googleapis.com/cloud-training/gcpnet/student/M5%20-%20Create%20a%20Dedicated%20Interconnect%20connection.mp4.



In order to use Dedicated Interconnect, your network must physically meet Google's network in a supported colocation facility. This map shows the locations where you can create dedicated connections. The link in the slides of this module has the full list of these locations [https://cloud.google.com/interconnect/docs/concepts/colocation-facilities].

Now, you might look at this map and say, "well I am nowhere near one of those locations." That's when you want to consider Partner Interconnect.

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 can't reach a Dedicated Interconnect colocation facility or if your data needs don't warrant a Dedicated Interconnect.

In order to use Partner Interconnect, you work with a supported service provider to connect your VPC and on-premises networks. The link in the slides has the full list of providers [https://cloud.google.com/interconnect/docs/concepts/service-providers]

These service providers have existing physical connections to Google's network that they make available for their customers to use. After you establish connectivity with a service provider, you can request a Partner Interconnect connection from your service provider. Then, you establish a BGP session between your Cloud Router and on-premises router to start passing traffic between your networks via the service provider's network.

Partner Interconnect can be configured to offer a 99.9% or a 99.99% uptime SLA between Google and the service provider. Please refer to the Partner Interconnect documentation for details on how to achieve these SLAs [https://cloud.google.com/interconnect/docs/concepts/partner-overview#redundancy]

Comparison of Interconnect options

Connection	Provides	Capacity	Requirements	Access Type
IPsec VPN tunnel	Encrypted tunnel to VPC networks through the public internet	1.5-3 Gbps per tunnel	On-premises VPN gateway	Internal IP addresses
Dedicated Interconnect	Dedicated, direct connection to VPC networks	10 Gbps or 100 Gbps per link	Connection in colocation facility	
Partner Interconnect	Dedicated bandwidth, connection to VPC network through a service provider	50 Mbps – 10 Gbps per connection	Service provider	

Let me compare the interconnect options that we just discussed. All of these options provide internal IP address access between resources in your on-premises network and in your VPC network. The main differences are the connection capacity and the requirements for using a service.

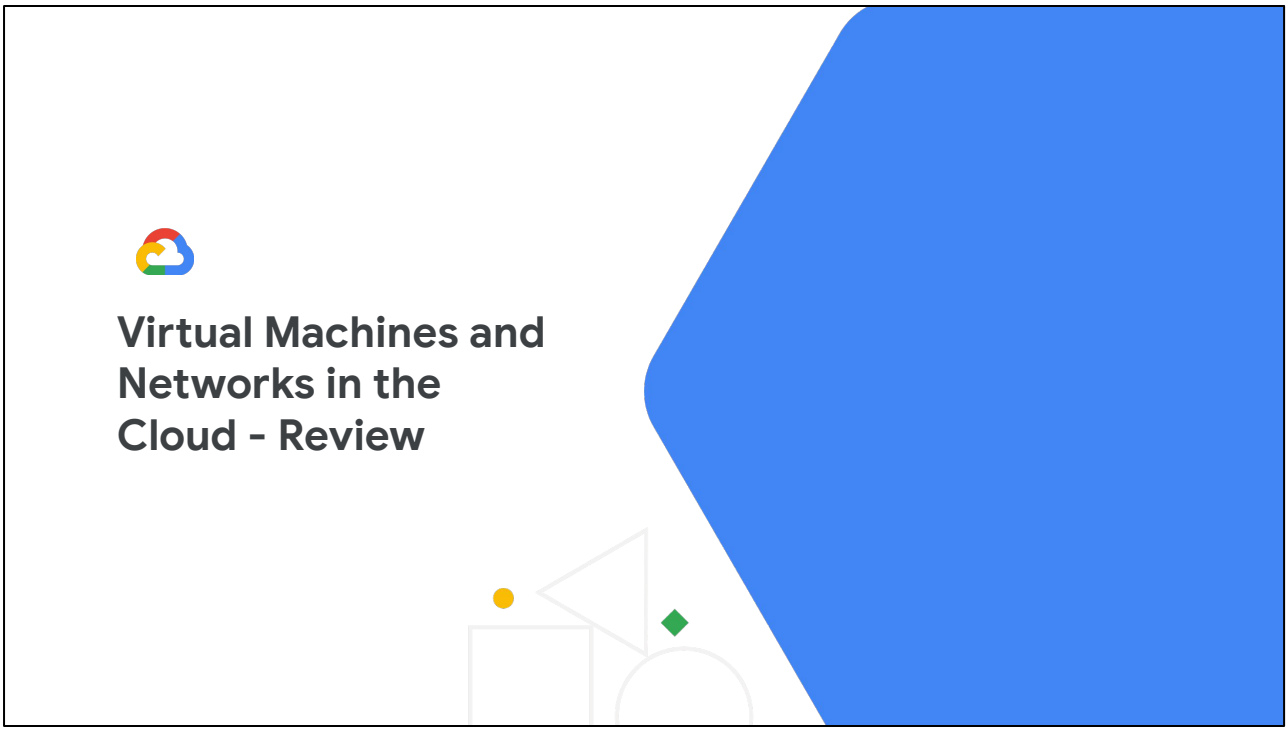
The IPsec VPN tunnels that Cloud VPN offers have a capacity of 1.5 to 3 Gbps per tunnel and require a VPN device on your on-premises network. The 1.5-Gbps capacity applies to traffic that traverses the public internet, and the 3-Gbps capacity applies to traffic that is traversing a direct peering link. You can configure multiple tunnels if you want to scale this capacity.

Dedicated Interconnect has a capacity of 10 Gbps or 100 Gbps per link and requires you to have a connection in a Google-supported colocation facility. You can have up to 8 links to achieve multiples of 10 Gbps, or up to 2 links to achieve multiples of 100 Gbps, but 10 Gbps is the minimum capacity.

Partner Interconnect on the other hand has a capacity of 50 Mbps to 10 Gbps per connection, and requirements depend on the service provider.

My recommendation is to start with VPN tunnels. When you need enterprise-grade connections to GCP, switch to Dedicated Interconnect or Partner Interconnect,

depending on your proximity to a colocation facility and your capacity requirements.



In this module, you learned about the Virtual Private Cloud, including how to control access to your network using firewall rules and how to create subnets. You then learned how to create and manage Virtual Machines in Compute Engine, choose the right configuration, and reduce costs by only paying for the time your virtual machine is running. Finally, you learned how to create a connection between your source environment and your Virtual Private Cloud.

In the next module, we will introduce you to Migrate for Compute Engine, Google Cloud's virtual machine migration tool. This will enable you to get your enterprise applications running in Google Cloud within minutes, while your data migrates transparently in the background. We will explain how to install it in your source environment, the migration process, and how to use special features like running a test clone. If you have a large number of VMs to migrate, don't worry. We will show you how to use migration waves to migrate a large number of machines at once.

Move on to the next module to learn more.