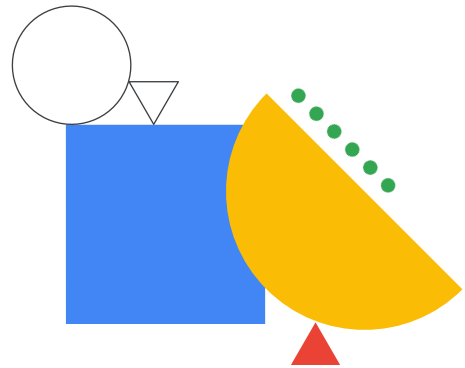
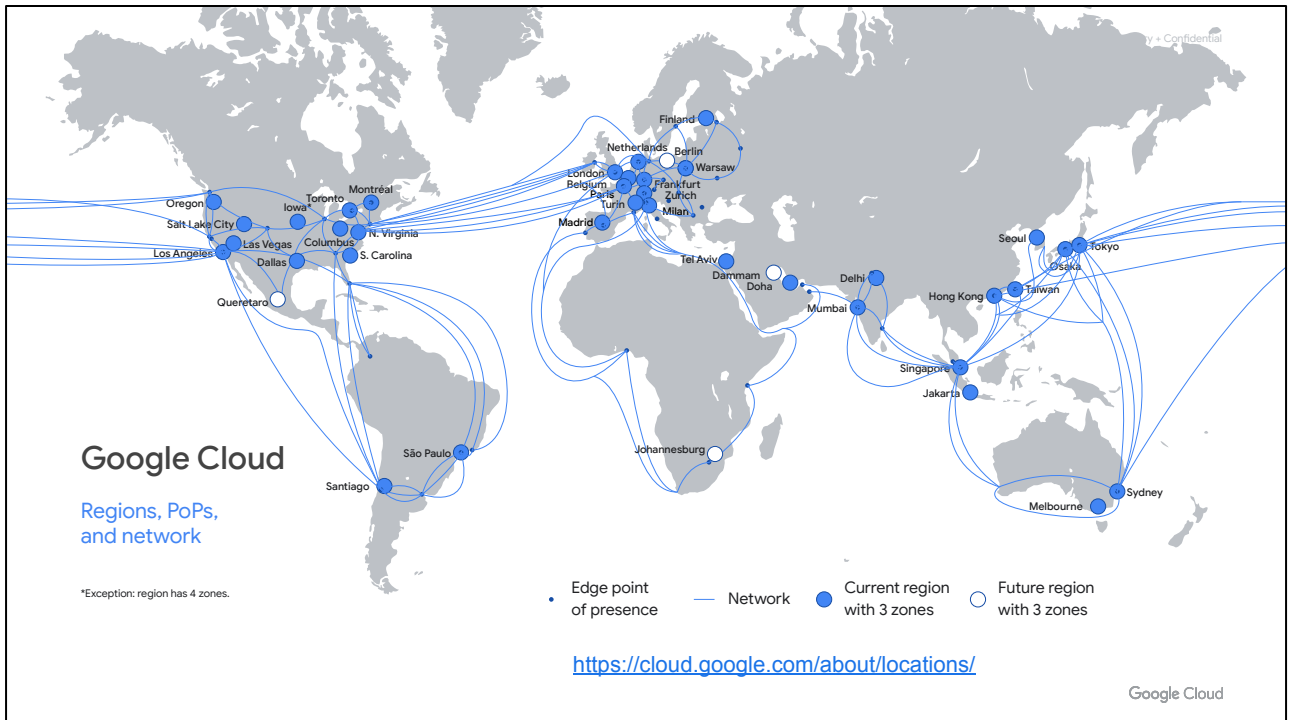


Virtual Networks



In this module, we will be covering virtual networks.

Google Cloud uses a software-defined network that is built on a global fiber infrastructure. This infrastructure makes Google Cloud one of the world's largest and fastest networks. Thinking about resources as services instead of as hardware will help you understand the options that are available, and their behavior.



This map represents Google Cloud. 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 blue; 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. Regions indicated with blue icons have three zones. Iowa is an exception, where the region called us-central1 has four zones: us-central1-a, us-central1-b, us-central1-c, and us-central1-f. For up-to-date information on regions and zones, please refer to the [documentation page](#).

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.

For more information about Google's networking infrastructure, refer to this [site](#).



Virtual Private Cloud (VPC)

Let's start by talking about Google Cloud's network, and specifically Virtual Private Cloud, or VPC.

VPC objects



Virtual Private Cloud

- Projects
- Networks
 - Default, auto mode, custom mode
- Subnetworks
- Regions
- Zones
- IP addresses
 - Internal, external, range
- Virtual machines (VMs)
- Routes
- Firewall rules

With Google Cloud, you can provision your Google Cloud resources, connect them to each other, and isolate them from each other in a Virtual Private Cloud. You can also define fine-grained networking policies within Google Cloud, and between Google Cloud and on-premises or other public clouds. Essentially, VPC is a comprehensive set of Google-managed networking objects, which we will explore in detail throughout this module.

Let me give you a high-level overview of these objects:

- Projects are going to encompass every single service that you use, including networks.
- Networks come in three different flavors: Default, auto mode, and custom mode.
- Subnetworks allow you to divide or segregate your environment.
- Regions and zones represent Google's data centers, and they provide continuous data protection and high availability.
- VPC provides IP addresses for internal and external use, along with granular IP address range selections.
As for virtual machines, in this module we will focus on configuring VM instances from a networking perspective.
- We'll also go over routes and firewall rules.

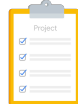


Projects, Networks, and Subnetworks

Let's start exploring the VPC objects by looking at projects, networks, and subnetworks.

Projects and networks

A project:



- Associates objects and services with billing.
- Contains networks (up to 15) that can be shared/peered.

A network:



- Has no IP address range.
- Is global and spans all available regions.
- Contains subnetworks.
- Is available as default, auto, or custom.

Projects are the key organizer of infrastructure resources in Google Cloud. A project associates objects and services with billing. Now, it's unique that projects actually contain entire networks. The default quota for each project is 15 networks, but you can simply request additional quota using the Google Cloud console. These networks can be shared with other projects, or they can be peered with networks in other projects, both of which we will cover later in the Architecting with Google Compute Engine course.

These networks do not have IP ranges but are simply a construct of all of the individual IP addresses and services within that network. Google Cloud's networks are global, spanning all available regions across the world that we showed earlier. So, you can have one network that literally exists anywhere in the world—Asia, Europe, Americas—all simultaneously.

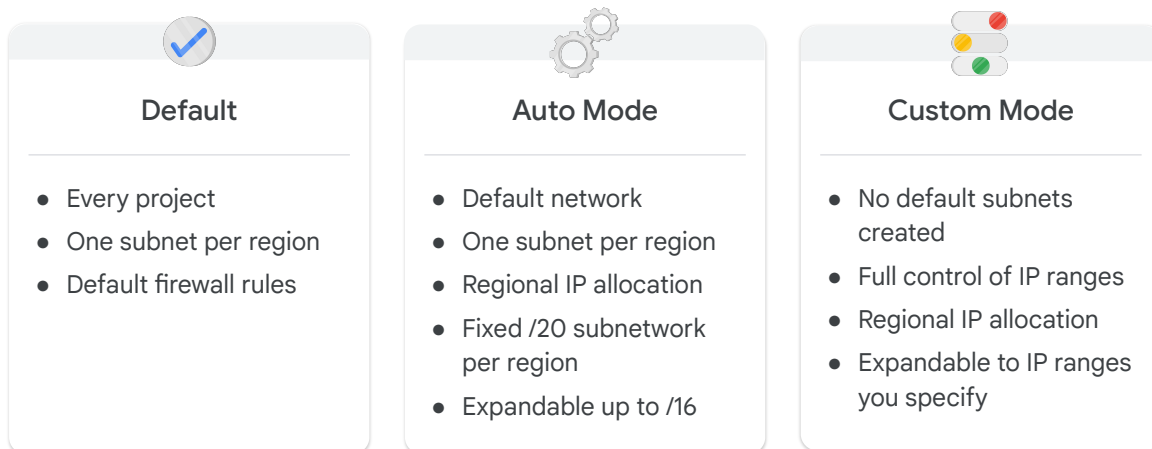
Inside a network, you can segregate your resources with regional subnetworks.

We just mentioned that there are different types of networks: default, auto, and custom. Let's explore these types of networks in more detail.

[More information on setting up a VPC:

<https://www.youtube.com/watch?v=cNb7xKyya5c>]

3 VPC network types



Google Cloud

Every project is provided with a default VPC network with preset subnets and firewall rules. Specifically, a subnet is allocated for each region with non-overlapping CIDR blocks 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.

In an auto mode network, one subnet from each region is automatically created within it. The default network is actually an auto mode network. 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.

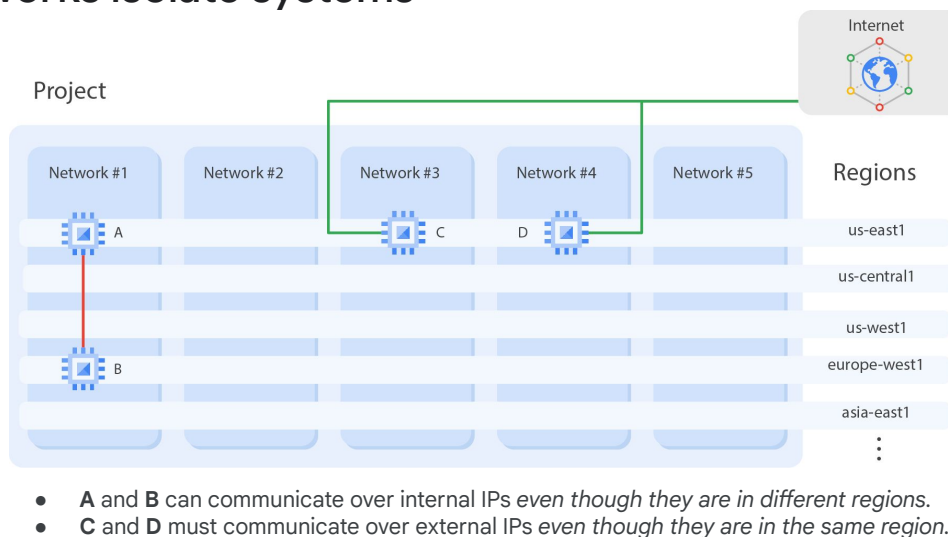
A custom mode network does not automatically create subnets. This type of network provides you with complete control over its subnets and IP ranges. You decide which subnets to create, in regions you choose, and using IP ranges you specify. These IP ranges cannot overlap between subnets of the same network.

Now, 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.

Google Cloud now supports IPv6 in a custom VPC network mode, for example you can configure IPv6 addressing on 'dual-stack' VM instances running both IPv4 and IPv6. You can learn a lot more about IPv6 in the "Networking in Google Cloud" course.

Networks isolate systems



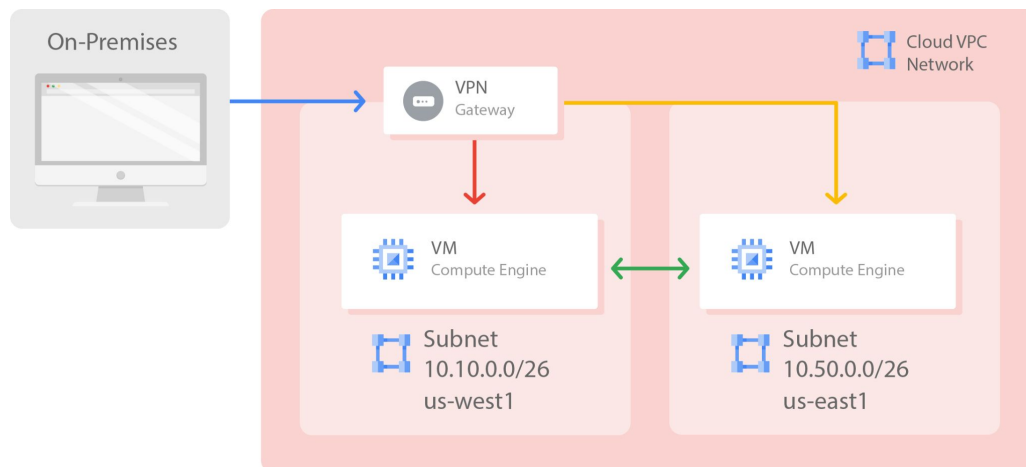
Google Cloud

On this slide, we have an example of a project that contains 5 networks. All of these networks span multiple regions across the world, as you can see on the right.

Each network contains separate virtual machines: A, B, C, and D. Because VMs A and B are in the same network, network 1, they can communicate using their internal IP addresses, even though they are in different regions. Essentially, your virtual machines, even if they exist in different locations across the world, take advantage of Google's global fiber network. Those virtual machines appear as though they're sitting in the same rack when it comes to a network configuration protocol.

VMs C and D, however, are not in the same network. Therefore, by default, these VMs must communicate using their external IP addresses, even though they are in the same region. The traffic between VMs C and D isn't actually touching the public internet, but is going through the Google Edge routers. This has different billing and security ramifications that we will explore later.

Google's VPC is global

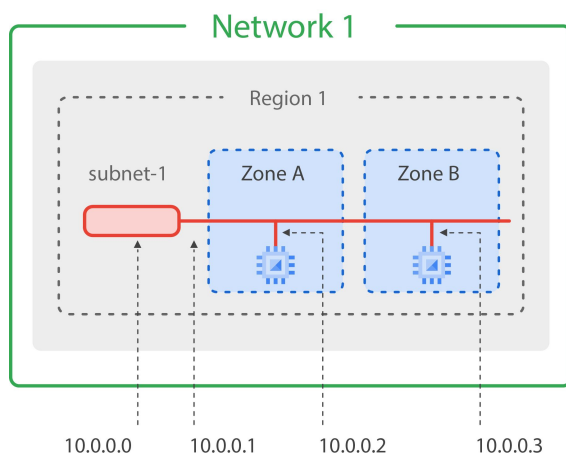


Google Cloud

Because VM instances within a VPC network can communicate privately on a global scale, a single VPN can securely connect your on-premises network to your Google Cloud network, as shown in this diagram. Even though the two VM instances are in separate regions (us-west1 and us-east1), they leverage Google's private network to communicate between each other and to an on-premises network through a VPN gateway.

This reduces cost and network management complexity.

Subnetworks cross zones



- VMs can be on the same subnet but in different zones.
- A single firewall rule can apply to both VMs.

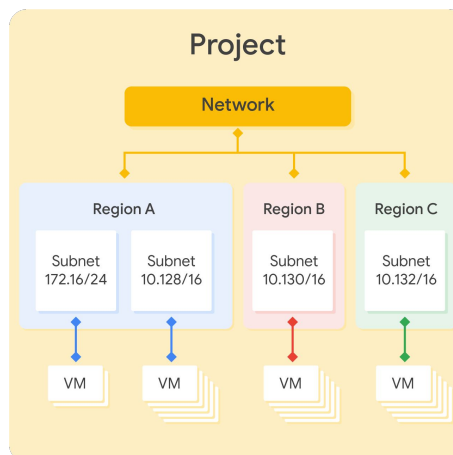
We mentioned that subnetworks work on a regional scale. Because a region contains several zones, subnetworks can cross zones.

This slide has a region, region 1, with two zones, zones A and B. Subnetworks can extend across these zones within the same region, such as, subnet-1. The subnet is simply an IP address range, and you can use IP addresses within that range. Notice that the first and second addresses in the range, .0 and .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.

Now, even though the two virtual machines in this example are in different zones, they still communicate with each other using the same subnet IP address. This means that a single firewall rule can be applied to both VMs, even though they are in different zones.

Expand subnets without re-creating instances

- Cannot overlap with other subnets
- IP range must be a unique valid CIDR block
- New subnet IP ranges have to fall within valid IP ranges
- Can expand but not shrink
- Auto mode can be expanded from /20 to /16
- Avoid large subnets



Google Cloud

Speaking of IP addresses of a subnet, Google Cloud VPCs let you increase the IP address space of any subnets without any workload shutdown or downtime. This diagram illustrates a network with subnets that have different subnet masks, allowing for more instances in some subnets than others. This gives you flexibility and growth options to meet your needs, but there are some things to remember:

- The new subnet must not overlap with other subnets in the same VPC network in any region.
- Each IP range for all subnets in a VPC network must be a unique valid CIDR block.
- Also, the new subnet IP address ranges are regional internal IP addresses and have to fall within [valid IP ranges](#).
 - Subnet ranges cannot match, be narrower, or be broader than a restricted range.
 - Subnet ranges cannot span a valid RFC range and a privately used public IP address range.
 - Subnet ranges cannot span multiple RFC ranges.
- The new network range must be larger than the original, which means the prefix length value must be a smaller number. In other words, you cannot undo an expansion.
- Now, auto mode subnets start with a /20 IP range. They can be expanded to a

- /16 IP range, but no larger. Alternatively, you can convert the auto mode subnetwork to a custom mode subnetwork to increase the IP range further.
- Also, avoid creating large subnets. Overly large subnets are more likely to cause CIDR range collisions when using Multiple Network Interfaces and VPC Network Peering, or when configuring a VPN or other connections to an on-premises network. Therefore, do not scale your subnet beyond what you actually need.

For a demo on how to expand a custom subnet in Google Cloud, refer to this [video](#).



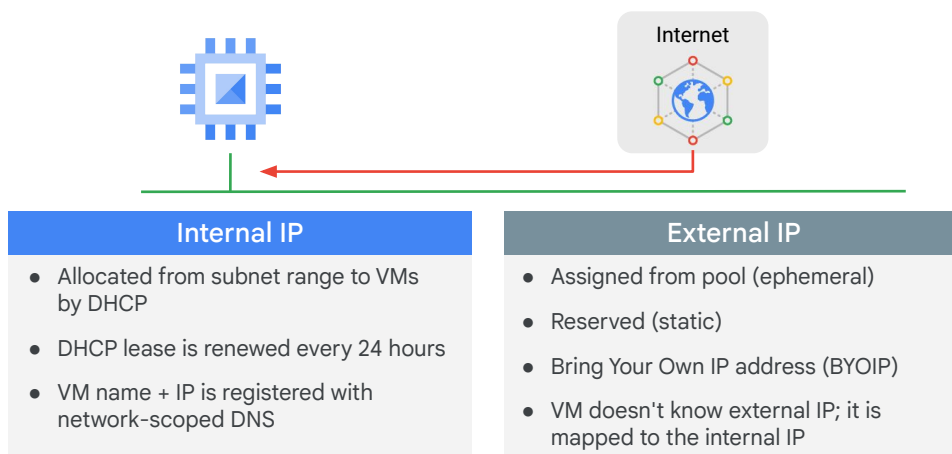
IP Addresses

Now that we covered Google Cloud networks at a high level, let's go deeper by exploring IP addresses.

VMs can have internal and external IP addresses



Cloud External
IP Addresses



Google Cloud

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

Every VM that starts up and any service that depends on virtual machines gets an internal IP address. Examples of such services are App Engine and Google Kubernetes Engine, which are explored in other courses.

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. 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.

The other IP address is the external IP address but this one 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. If you reserve a static external IP address and do not assign it to a resource such as a VM instance or a forwarding rule, you are charged at a higher rate than for static and ephemeral external IP addresses that are in use.

For more information about this, refer to the [documentation page](#). You can use your

own publicly routable IP address prefixes as Google Cloud external IP addresses and advertise them on the internet. In order to be eligible, you must own and bring a /24 block or larger.

For a quick walk through of internal and external IP addresses in Google Cloud, refer to this [demo](#).

External IPs are mapped to internal IPs

Name ^	Zone	Machine type	Recommendation	In use by	Internal IP	External IP	Connect
<input type="checkbox"/> instance-1	us-east1-d	1vCPU, 3.75 GB			10.142.0.2	104.196.149.82	SSH ▾ ⋮

```
$ sudo /sbin/ifconfig
eth0
  Link encap:Ethernet  HWaddr 42:01:0a:8e:00:02
  inet addr:10.142.0.2  Bcast:10.142.0.2  Mask:255.255.255.255
  UP BROADCAST RUNNING MULTICAST  MTU:1460  Metric:1
  RX packets:397 errors:0 dropped:0 overruns:0 frame:0
  TX packets:279 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:66429 (64.8 KiB)  TX bytes:41662 (40.6 KiB)

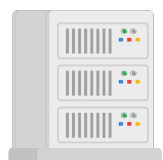
lo
  Link encap:Local Loopback
  inet addr:127.0.0.1  Mask:255.0.0.0
  inet6 addr: ::1/128 Scope:Host
  UP LOOPBACK RUNNING  MTU:65536  Metric:1
  RX packets:0 errors:0 dropped:0 overruns:0 frame:0
  TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:0
  RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Google Cloud

Regardless of whether you use an ephemeral or static IP address, the external address is unknown to the OS of the VM. The external IP address is mapped to the VM's internal address transparently by VPC. We are illustrating this here by running `ifconfig` within a VM in Google Cloud, which only returns the internal IP address.

Let's explore this further by looking at DNS resolution for both internal and external addresses.

DNS resolution for internal addresses



Google Cloud has two types of internal DNS:

- Zonal
- Global

Each instance has a hostname that can be resolved to an internal IP address:

- The hostname is the same as the instance name.
- FQDN is [hostname].[zone].c.[project-id].internal

Example: my-server.us-central1-a.c.guestbook-151617.internal

Name resolution is handled by internal DNS resolver:

- Provided as part of Compute Engine (169.254.169.254).
- Configured for use on instance via DHCP.
- Provides answer for internal and external addresses.

Let's start with internal addresses.

Google Cloud has two types of internal DNS names, Zonal and Global (project wide) DNS. In general, Google strongly recommends using zonal DNS because it offers higher reliability guarantees by isolating failures in the DNS registration to individual zones.

Each instance has a hostname that can be resolved to an internal IP address. This hostname is the same as the instance name. There is also an internal fully qualified domain name, or FQDN, for an instance that uses the format shown on the slide.

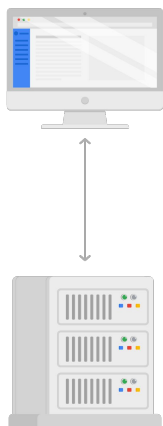
If you delete and recreate an instance, the internal IP address can change. This change can disrupt connections from other Compute Engine resources, which must obtain the new IP address before they can connect again. However, the DNS name always points to a specific instance, no matter what the internal IP address is.

Each instance has a metadata server that also acts as a DNS resolver for that instance. The metadata server handles all DNS queries for local network resources and routes all other queries to Google's public DNS servers for public name resolution. We previously mentioned that an instance is not aware of any external IP address assigned to it. Instead, the network stores a lookup table that matches

external IP addresses with the internal IP addresses of the relevant instances.

For more information, including how to set up your own resolver on instances, refer to the [documentation page](#).

DNS resolution for external addresses



- Instances with external IP addresses can allow connections from hosts outside the project.
 - Users connect directly using external IP address.
 - Admins can also publish public DNS records pointing to the instance.
 - Public DNS records are not published automatically.
- DNS records for external addresses can be published using existing DNS servers (outside of Google Cloud).
- DNS zones can be hosted using Cloud DNS.

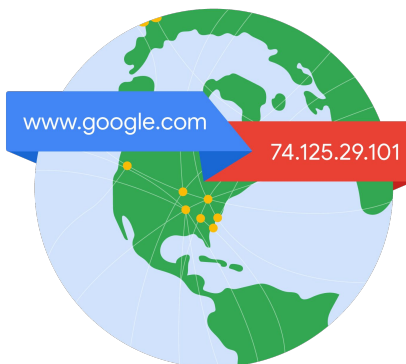
Now, let's look at external addresses.

Instances with external IP addresses can allow connections from hosts outside of the project. Users can do so directly using the external IP address. Public DNS records pointing to instances are not published automatically; however, admins can publish these using existing DNS servers.

Domain name servers can be hosted on Google Cloud, using Cloud DNS. This is a managed service that is definitely worth considering, so let's explore it in more detail.

Host DNS zones using Cloud DNS

- Google's DNS service
- Translate domain names into IP address
- Low latency
- High availability (100% uptime SLA)
- Create and update millions of DNS records
- UI, command line, or API



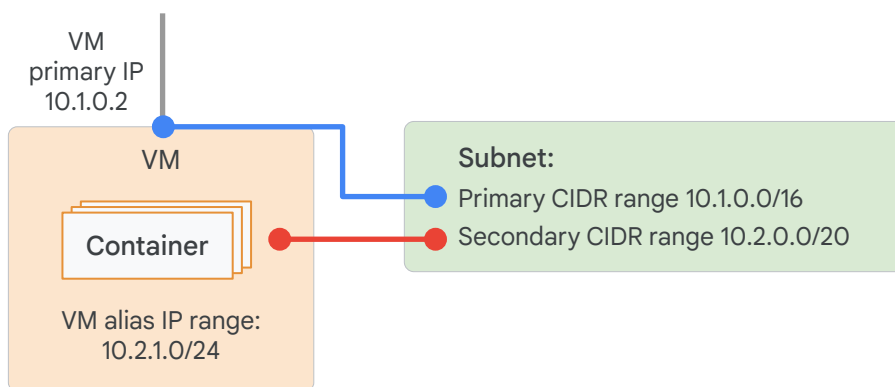
Cloud DNS

Cloud DNS is a scalable, reliable, and managed authoritative Domain Name System, or DNS, service running on the same infrastructure as Google. Cloud DNS translates requests for domain names like google.com into IP addresses.

Cloud DNS uses Google's global network of Anycast name servers to serve your DNS zones from redundant locations around the world, providing lower latency and high availability for your users. High availability is very important because if you can't look up a domain name, the internet might as well be down. That's why Google Cloud offers a 100% uptime Service Level Agreement, or SLA, for domains configured in Cloud DNS. For more information about this SLA, refer to the [documentation page](#).

Cloud DNS lets you create and update millions of DNS records without the burden of managing your own DNS servers and software. Instead, you use a simple user interface, command-line interface, or API. For more information about Cloud DNS, refer to the [documentation page](#).

Assign a range of IP addresses as aliases to a VM's network interface using alias IP ranges



Another networking feature of Google Cloud is Alias IP Ranges.

Alias IP Ranges let you assign a range of internal IP addresses as an alias to a virtual machine's network interface. This is useful if you have multiple services running on a VM, and you want to assign a different IP address to each service.

In essence, you can configure multiple IP addresses, representing containers or applications hosted in a VM, without having to define a separate network interface. You just draw the alias IP range from the local subnet's primary or secondary CIDR ranges. This diagram provides a basic illustration of primary and secondary CIDR ranges and VM alias IP ranges.

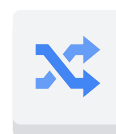
For more information about Alias IP Ranges, refer to the [documentation page](#).



Routes and Firewall Rules

So far you've learned about projects, networks, subnetworks, and IP addresses. Let's use what you learned to understand how Google Cloud routes traffic.

A route is a mapping of an IP range to a destination



Cloud Routes

Every network has:

- Routes that let instances in a network send traffic directly to each other.
- A default route that directs packets to destinations that are outside the network.

Firewall rules must also allow the packet.

By default, every network has routes that let instances in a network send traffic directly to each other, even across subnets. In addition, every network has a default route that directs packets to destinations that are outside the network. Although these routes cover most of your normal routing needs, you can also create special routes that override these routes.

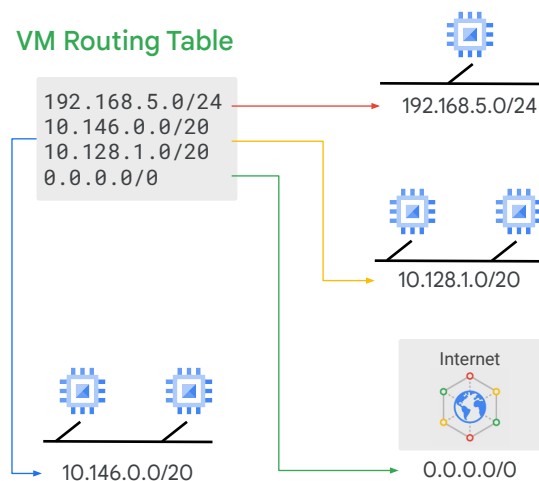
Just creating a route does not ensure that your packets will be received by the specified next hop. Firewall rules must also allow the packet.

The default network has pre-configured firewall rules that allow all instances in the network to talk with each other. Manually created networks do not have such rules, so you must create them, as you will experience in the first lab.

Routes map traffic to destination networks

- Apply to traffic egressing a VM.
- Forward traffic to most specific route.
- Are created when a subnet is created.
- Enable VMs on same network to communicate.
- Destination is in CIDR notation.
- Traffic is delivered only if it also matches a firewall rule.

VM Routing Table



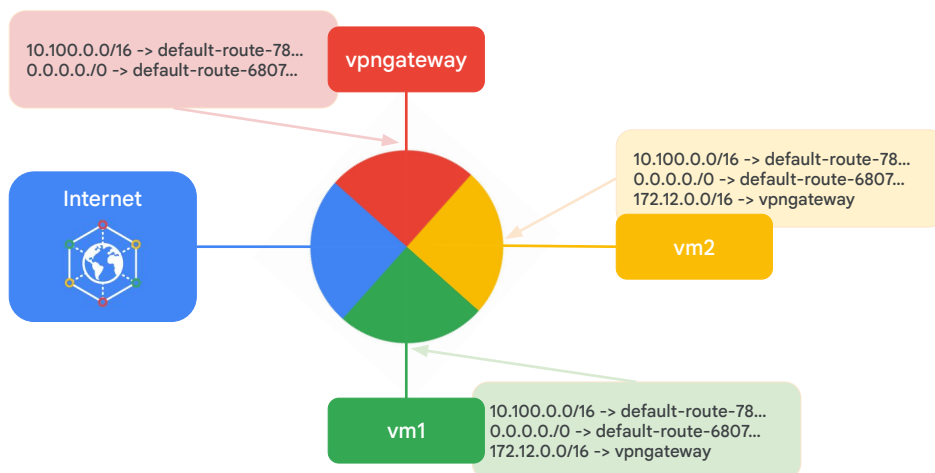
Google Cloud

Routes match packets by destination IP address. However, no traffic will flow without also matching a firewall rule.

A route is created when a network is created, enabling traffic delivery from “anywhere.” Also, a route is created when a subnet is created. This is what enables VMs on the same network to communicate.

This slide shows a simplified routing table, but let's look at this in more detail.

Instance routing tables



Each route in the Routes collection may apply to one or more instances. A route applies to an instance if the network and instance tags match. If the network matches and there are no instance tags specified, the route applies to all instances in that network. Compute Engine then uses the Routes collection to create individual read-only routing tables for each instance.

This diagram shows a massively scalable virtual router at the core of each network. Every virtual machine instance in the network is directly connected to this router, and all packets leaving a virtual machine instance are first handled at this layer before they are forwarded to their next hop. The virtual network router selects the next hop for a packet by consulting the routing table for that instance.

Firewall rules protect your VM instances from unapproved connections



Cloud Firewall Rules

- VPC network functions as a distributed firewall.
- Firewall rules are applied to the network as a whole.
- Connections are allowed or denied at the instance level.
- Firewall rules are stateful.
- Implied deny all ingress and allow all egress.

Google Cloud firewall rules protect your virtual machine instances from unapproved connections, both inbound and outbound, known as ingress and egress, respectively. Essentially, every VPC network functions as a distributed firewall.

Although firewall rules are applied to the network as a whole, connections are allowed or denied at the instance level. You can think of the firewall as existing not only between your instances and other networks, but between individual instances within the same network.

Google Cloud firewall rules are stateful. This means that if a connection is allowed between a source and a target or a target and a destination, all subsequent traffic in either direction will be allowed. In other words, firewall rules allow bidirectional communication once a session is established.

Also, if for some reason, all firewall rules in a network are deleted, 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 <code>ingress</code> rules only.
	Outbound connections are matched against <code>egress</code> rules only.
source or destination	For the <code>ingress</code> direction, sources can be specified as part of the rule with IP addresses, source tags or a source service account.
	For the <code>egress</code> 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.
Rule assignment	All rules are assigned to all instances, but you can assign certain rules to certain instances only.

Google Cloud

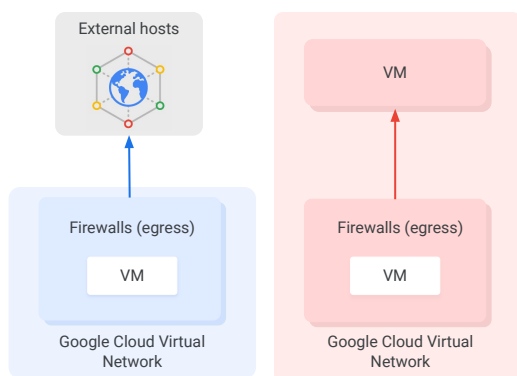
You can express your desired firewall configuration as a set of firewall rules. Conceptually, a firewall rule is composed of the following parameters:

- The **direction** of the rule. Inbound connections are matched against `ingress` rules only, and outbound connections are matched against `egress` rules only.
- The **source** of the connection for `ingress` packets, or the **destination** of the connection for `egress` packets.
- The **protocol** and **port** of the connection, where any rule can be restricted to apply to specific protocols only or specific combinations of protocols and ports only.
- The **action** of the rule, which is to allow or deny packets that match the direction, protocol, port, and source or destination of the rule.
- The **priority** of the rule, which governs the order in which rules are evaluated. The first matching rule is applied.
- The **rule assignment**. By default, all rules are assigned to all instances, but you can assign certain rules to certain instances only.

For more information on firewall rule components, refer to the [documentation page](#).

Let's look at some Google Cloud firewall use cases for both `egress` and `ingress`.

Google Cloud firewall use case: Egress



Conditions:

- Destination CIDR ranges
- Protocols
- Ports

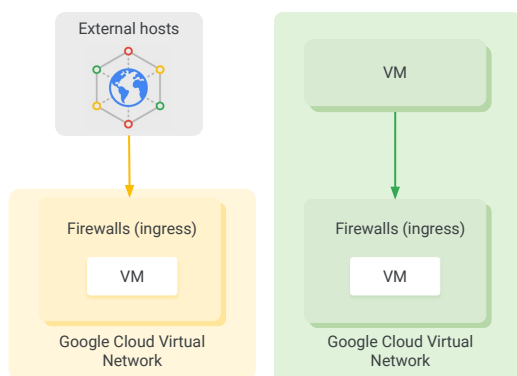
Action:

- Allow: permit the matching egress connection
- Deny: block the matching egress connection

Egress firewall rules control outgoing connections originated inside your Google Cloud network. Egress **allow** rules allow outbound connections that match specific protocol, ports, and IP addresses. Egress **deny** rules prevent instances from initiating connections that match non-permitted port, protocol, and IP range combinations.

For egress firewall rules, destinations to which a rule applies may be specified using IP CIDR ranges. Specifically, you can use destination ranges to protect from undesired connections initiated by a VM instance toward an external host, as shown on the left. You can also use destination ranges to prevent undesired connections from internal VM instances to a specific Google Cloud CIDR range. This is illustrated in the middle, where a VM in a specific subnet is shown attempting to connect inappropriately to another VM within the same network.

Google Cloud firewall use case: Ingress



Conditions:

- Source CIDR ranges
- Protocols
- Ports

Action:

- Allow: permit the matching ingress connection
- Deny: block the matching ingress connection

Ingress firewall rules protect against incoming connections to the instance from any source. Ingress **allow** rules allow specific protocol, ports, and IP addresses to connect in. The firewall prevents instances from receiving connections on non-permitted ports or protocols. Rules can be restricted to only affect particular sources. Source CIDR ranges can be used to protect an instance from undesired connections coming either from external networks or from Google Cloud IP ranges.

This diagram illustrates a VM receiving a connection from an external address, and another VM receiving a connection from a VM within the same network. You can control ingress connections from a VM instance by constructing inbound connection conditions using source CIDR ranges, protocols, or ports.