Google Cloud Platform VPC

Architecting with GCP Fundamentals: Infrastructure

VPC, CLOUD EXTERNAL IP ADDRESSES, CLOUD FIREWALL RULES, CLOUD ROUTES

VIRTUAL NETWORKING, BASTION HOST

Last modified 2017-11-27

Agenda

- **Google Cloud Platform (GCP) VPC**
- Projects, networks, and subnetworks
- IP addresses
- Routes and rules
- Billing
- Lab
- Common network designs
- Lab
- Quiz

Google Cloud Platform Virtual Private Cloud (VPC) Objects

- Projects
- Networks
  - Default, auto mode, custom mode
- Subnetworks
- Regions
- Zones

- IP addresses
  - Internal, external, range
- Virtual machines (VMs)
- Routes
- Firewall rules

Google Cloud

These 9 objects are all you need to know to understand the fundamentals of Google Cloud Patform VPC.
IP Forwarding, Protocol Forwarding, Load Balancing, Cloud DNS, and VPN Tunnels are built on top of this framework and are covered separately, not in this module.

Agenda

- Google Cloud Platform VPC
- **Projects, networks, and subnetworks**
- IP addresses
- Routes and rules
- Billing
- Lab
- Common network designs
- Lab
- Quiz

Projects and networks

A project:

- Associates objects and services with billing.
- Contains networks (quota max 5).

A network:

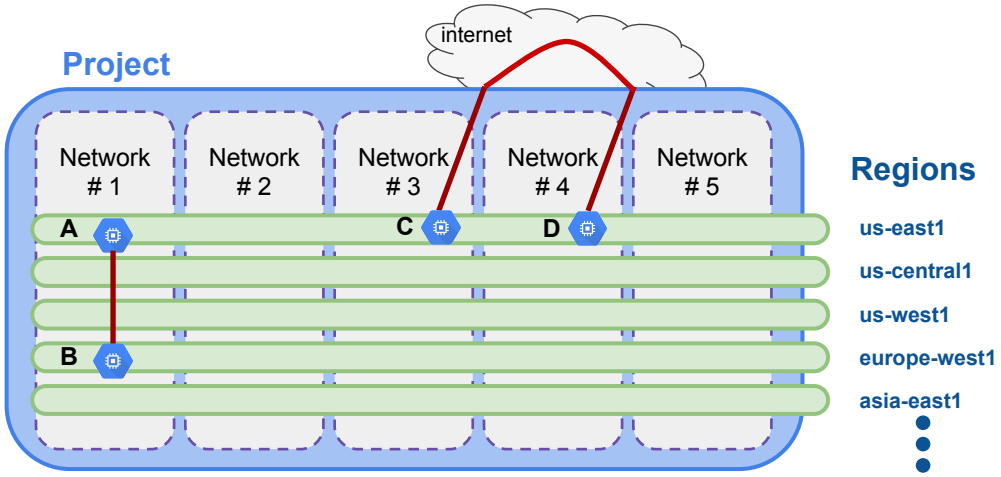
- Has no IP address range.
- Is global and spans all available regions.
- Contains subnetworks.
- Can be of type default, auto mode, or custom mode*.

*An auto mode network can be converted to custom mode network, but *"once custom, always custom."*

Google Cloud

A fourth type of network called "Legacy Google Compute Engine" exists but is not covered here and is not recommended.

Networks isolate systems



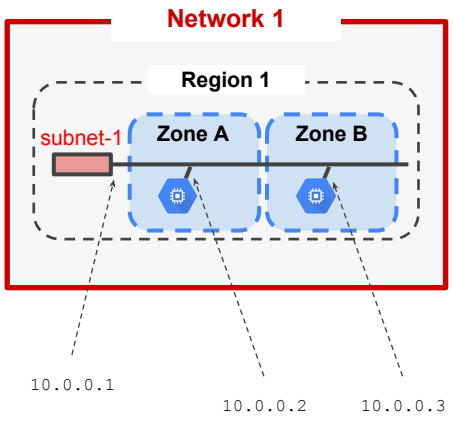
- A and B can communicate over internal IPs *even though they are in different regions.*
- C and D must communicate over external IPs *even though they are in the same region.*

Google Cloud

You can use networks to isolate systems. If you want to make sure that System C can never communicate with System D except over the internet, place them in separate networks. In this scenario, for C and D to communicate, they would communicate by egressing outside of the project. The traffic isn't actually touching the internet, but is going through the Google Edge routers, which has different billing and security ramifications.

Instances in the same network, such as A and B, can communicate with each other over internal IPs even though they are in different regions.

Subnetworks cross zones

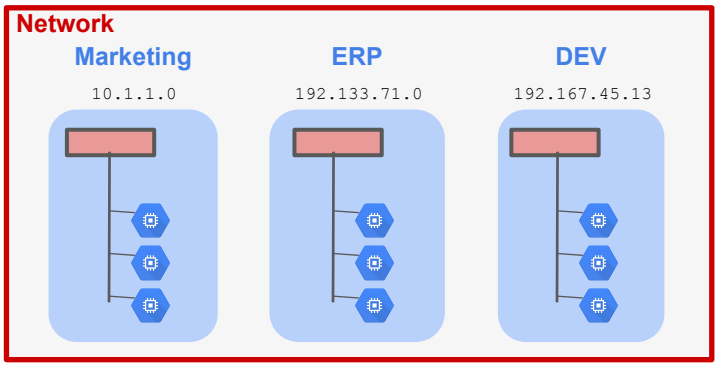


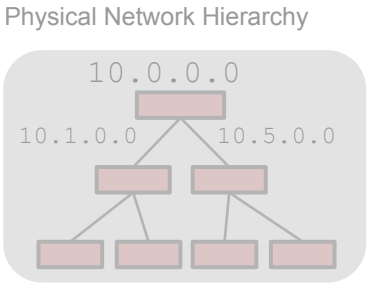
- Subnetworks can extend across zones in the same region.
- One VM and an alternate VM can be on the same subnet but in different zones.
- A single firewall rule can apply to both VMs even though they are in different zones.

Google Cloud

- Defined by internal IP address prefix range
- Specified in CIDR notation
  - IP ranges cannot overlap between subnets
  - IP range can be expanded but can never shrink
  - Specific to one region
- Can cross zones within the region
- Notice that the first address in the range, 10.0.0.1, is reserved for the "router" address. And the last address in the range, 10.0.0.255, is reserved for the "broadcast" address. VPC networks only support IPv4 unicast traffic. IPv4 broadcast and IPv4 multicast are not supported.

Subnetworks are for managing resources





Networks have no IP range, so subnetworks don't need to fit into an address hierarchy. Instead, subnetworks can be used to group and manage resources. They can represent departments, business functions, or systems.

Google Cloud

Each VPC network is subdivided into subnets, and each subnet is contained within a single region. You can have more than one subnet in a region for a given VPC network. Each subnet has a contiguous private RFC1918 IP space. You create instances, containers, and the like in these subnets. When you create an instance, you must create it in a subnet, and the instance draws its primary internal IP address from that subnet.

In physical networks, subnetworks must fit into a tree-shaped hierarchy composed of progressively more specific IP prefix ranges.
In the example, the top-level network IP range is 10.0.0.0, with narrow subnets 10.1.0.0 and 10.5.0.0, and progressively narrower ranges. Data arrives at the network level and is routed to more specific subnets until it reaches the destination subnet. A subnet has to fit into the larger hierarchy to receive traffic. Machines that are components of a system are likely to be distributed within the network based on where in the network bandwidth is available or based on separation of a primary machine from the associated backup for availability. They are not likely to be located on one or a few subnets.

In the GCP VPC, that physical network structure doesn't exist. The network has no top-level IP range. Subnetworks receive traffic without having to fit into a tree-shaped topology. And bandwidth does not depend on location in the hierarchy. Unbounded by traffic-routing requirements, subnetworks take on a different role in a GCP VPC; they become resource management tools.

Subnets are ways to group similar or related resources. In the example, one subnet represents a company organization, "Marketing," another subnet represents a business function, Electronic Resource Planning (ERP), and another subnet represents a technology system, Continuous Integration Development Environment (CIDE). Resource monitoring and management and policy implementation become simplified when related VMs are located in a few subnets defined for that purpose.

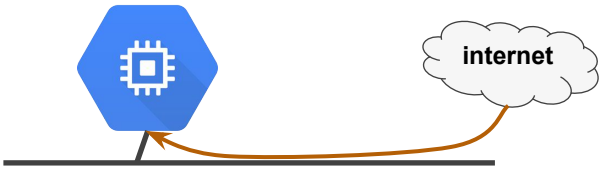
For more information, see
https://cloud.google.com/compute/docs/vpc/#vpc_networks_and_subnets

Agenda

- Google Cloud Platform VPC
- Projects, networks, and subnetworks
- **IP addresses**
- Routes and rules
- Billing
- Lab
- Common network designs
- Lab
- Quiz

Google Cloud

IP addresses



Internal IP	External IP
Allocated from subnet range to VMs by DHCP	Assigned from pool (ephemeral)
DHCP lease is renewed every 24 hours	Reserved (static) Billed when not attached to a running VM
VM name + IP is registered with network-scoped DNS	VM doesn't know external IP; it is mapped to the internal IP

Google Cloud

VMs are allocated internal IP addresses from the prefix range of a subnetwork by DHCP, and the lease is renewed every 24 hours. VMs are assigned external IP addresses from a pool (ephemeral), or can be assigned a reserved external IP (static). In either case, the external address is unknown to the VM and is mapped to the VM's internal address transparently by the GCP VPC.

Static IPs that are assigned to a VM or a load balancer are not charged at all. https://cloud.google.com/compute/pricing

When you create VMs in GCP, their symbolic name is registered with an internal DNS service that translates the name to the internal IP address. DNS is scoped to the network and connected to internet DNS. 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.

For more information, see https://cloud.google.com/compute/docs/ip-addresses/

External IPs are mapped to internal IPs

Name	Zone	Machine type	Recommendation	In use by	Internal IP	External IP	Connect
instance-1	us-east1-d	1 vCPU, 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

When an external IP address is assigned, the VM has no knowledge of the address; instead, an internal IP is mapped to the external IP.

DNS resolution for internal addresses

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].c.[project-id].internal.
  - Example: guestbook-test.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.

Google Cloud

You can view instance names, which are used in hostnames, via:

- Console: https://console.cloud.google.com/compute/instances?project=*[project_id]*
- gcloud: **compute instances list**
- API: **GET /project/zones/zone/instances**

When resolving hostnames for hosts outside of GCP, the resolver forwards queries to the standard Google DNS servers.

Each instance has a metadata server that also acts as a DNS resolver for that instance. DNS lookups are performed for instance names. The metadata server itself stores all DNS information for the local network and queries Google's public DNS servers for any addresses outside of the local network.

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, see: https://cloud.google.com/compute/docs/vpc/internal-dns

DNS resolution for external addresses

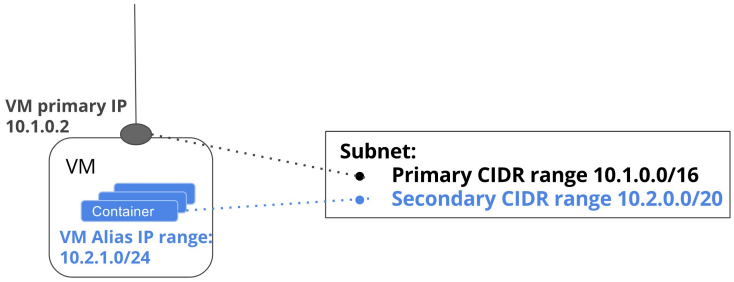
- Instances with external IP addresses can allow connections from hosts outside of the project.
  - Users connect directly using external IP address.
  - Admins can also publish public DNS records pointing to the instance.
  - Public DNS records are not published automatically.
- DNS records for external addresses can be published using existing DNS servers (outside of GCP).
- DNS zones can be hosted using Cloud DNS.
  - Create zone and configure domain DNS to use.
  - Create, update, and remove records manually or via API.

Google Cloud

For more details on using Cloud DNS, see https://cloud.google.com/dns/docs/

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

- The primary CIDR range, 10.1.0.0/16, is configured as part of a subnet.
- The secondary CIDR range, 10.2.0.0/20, is configured as part of a subnet.
- The VM primary IP, 10.1.0.2, is allocated from the primary CIDR range, 10.1.0.0/16.
- An alias IP range, 10.2.1.0/24, is allocated in the VM from the secondary CIDR range, 10.2.0.0/20.
- The addresses in the alias IP range are used as the IP addresses of the containers hosted in the VM.



Google Cloud

Alias IP ranges lets you assign a range of IP addresses as aliases to a VM's primary network interface.

If you have only one service running on a VM, you can reference it using the VM's primary interface. If you have multiple services running on a VM, you may want to assign each one a different internal IP address. You can do this with alias IP ranges. Routing to these alias IP ranges happens automatically. You do not have to configure any routes manually.

Using IP aliasing, you can configure multiple IP addresses, representing containers or applications hosted in a VM, without having to define a separate network interface. Draw the alias IP range from the local subnet's primary or secondary CIDR ranges. Configuring alias IP ranges describes commands for setting up a subnet with secondary ranges and for assigning alias IP addresses to VMs.

The diagram provides a basic illustration of primary and secondary CIDR ranges and VM alias IP ranges.

For more information, see:
Alias IP Ranges: https://cloud.google.com/compute/docs/alias-ip/
Configuring Alias IP Ranges:
https://cloud.google.com/compute/docs/configure-alias-ip-ranges

Agenda

- Google Cloud Platform VPC
- Projects, networks, and subnetworks
- IP addresses
- **Routes and rules**
- Billing
- Lab
- Common network designs
- Lab
- Quiz

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

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.

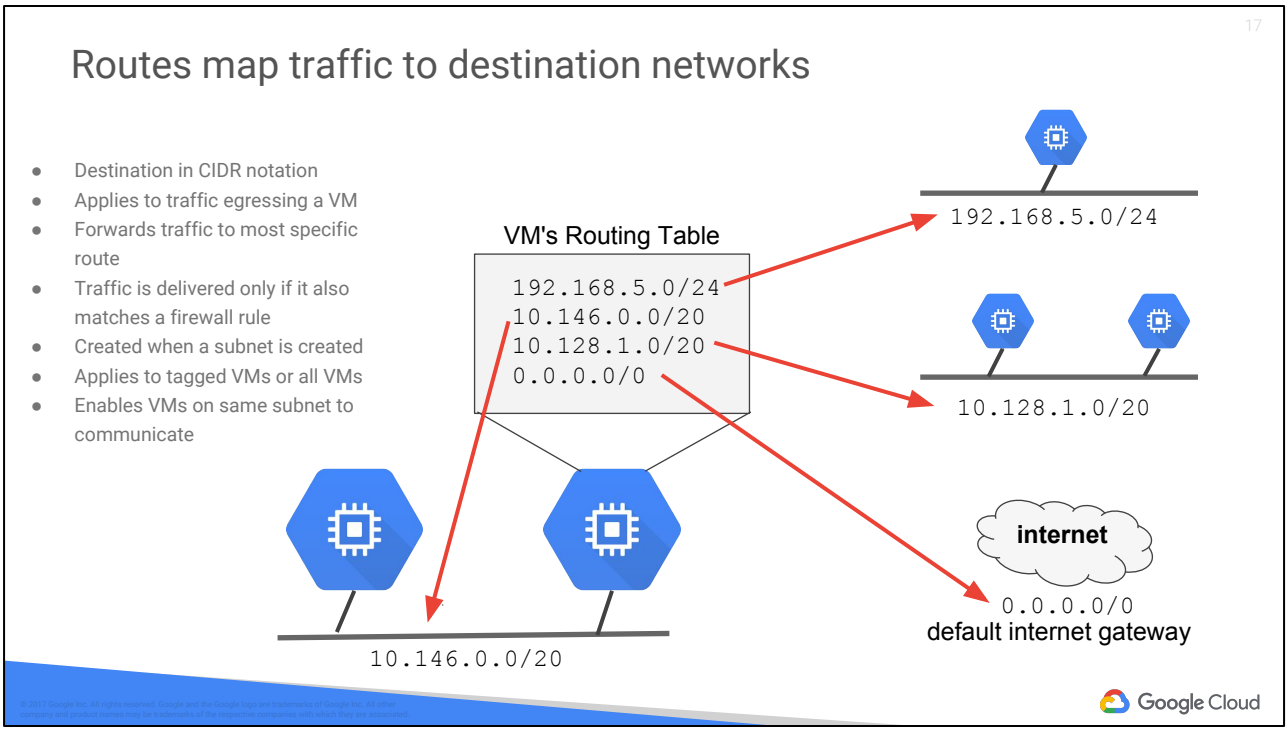
The fact that a packet has a route to a destination doesn't mean it can get there; firewall rules must also allow the packet.

Google Cloud

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 because a packet has a route to a destination does not mean that it can get there. Firewall rules must also allow the packet. The default network has preconfigured 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.

Routes allow you to implement more advanced networking functions in your virtual machines, such as setting up many-to-one NAT and transparent proxies. If you do not need any advanced routing solutions, the default routes should be sufficient for handling most outgoing traffic.

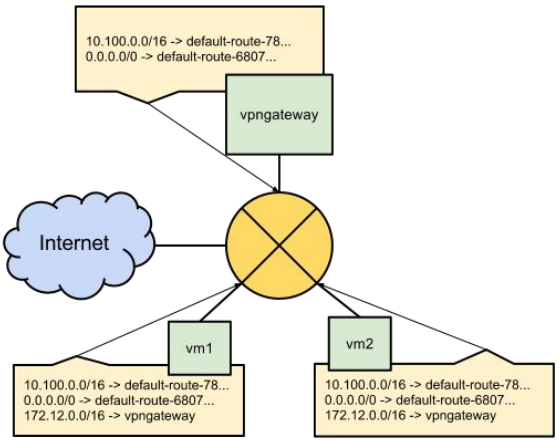


- Specified by CIDR notation
- A route is created when a network is created
  - Enables delivery of traffic from "anywhere"
- A route is created when a subnet is created
  - Enables VMs on the same subnet to communicate
- Traffic is forwarded to the most specific route

However, no traffic will flow without also matching a firewall rule. Routes match packets by destination IP address. More specific IP ranges are preferred over larger ranges. If there is a tie, the route with the smallest priority value is chosen. If there is still a tie, layer 3 and 4 headers are used to select one route.

For more information, see:
https://cloud.google.com/compute/docs/networking#network_routes

Instance routing tables



Google Cloud

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.

The 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 on to their next hop. The virtual network router selects the next hop for a packet by consulting the routing table for that instance. In the diagram, the green boxes are virtual machine instances, the router is in yellow at the center, and the individual routing tables are indicated by the tan boxes.

For more information, see:
Routes Overview: https://cloud.google.com/compute/docs/vpc/routes
Using Routes: https://cloud.google.com/compute/docs/vpc/using-routes

Firewall rules protect your VM instances from unapproved connections

- Every VPC network also functions as a distributed firewall.
- Firewall rules are applied to the network as a whole.
- Connections are *allowed* or *denied* at the instance level.

Google Cloud

Firewall rules protect your VM instances from unapproved connections. Every VPC network also functions as a distributed firewall. Firewall rules are applied to the network as a whole, and connections are allowed or denied at the instance level.

When you specify a rule, GCP assigns it to every instance in the network unless you restrict that assignment. You can restrict a rule so that it only applies to certain instances by using target tags or target service accounts. For example, if you want instances owned by a particular service account only to be reachable from the internet, you can create a rule that allows ingress from the internet and restrict the rule to only that service account. GCP then updates the firewall with that rule for those instances only.

If 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.

For more information, see:
https://cloud.google.com/compute/docs/vpc/firewalls#firewall_rules_in_gcp

A firewall rule is composed of the following parameters

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
Rule assignment	All rules are assigned to all instances, but you can assign certain rules to certain instances only

Google Cloud

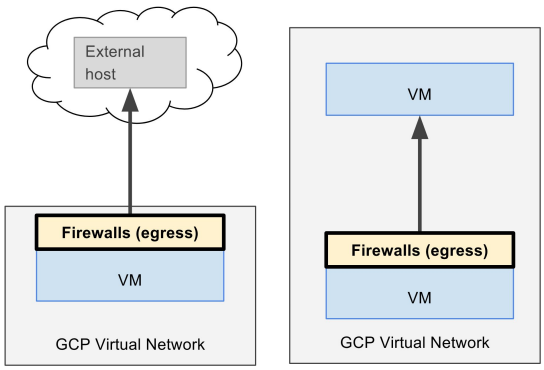
You can create your desired firewall configuration by creating a set of firewall rules. Each firewall rule is composed of the following parameters:

- A rule applies to one connection direction only, either inbound to instances (ingress rules) or outbound from instances (egress rules).
- An ingress rule may be configured to affect connections coming from particular sources only. You can specify the source by IP addresses, source tags, or a source service account. An egress rule may be configured to affect connections headed for particular destinations only. You can specify destinations with one or more ranges of IP addresses. If sources or destinations are unspecified, the rule applies to all sources or destinations.
- Any rule can be restricted to apply to “specific protocols only” or “specific combinations of protocols and ports only.” The protocol can be specified as a well-known protocol string (tcp, udp, icmp, esp, ah, sctp) or as the IP protocol number. If no protocols are specified, the rule applies to all protocols.
- A rule can either allow or deny connections that match the rest of the rule.
- The priority of the rule governs the order in which rules are evaluated. The first matching rule is applied.

- By default, all rules are assigned to all instances, but you can assign certain rules to certain instances only.

For more information, see:
https://cloud.google.com/compute/docs/vpc/firewalls#firewall_rule_components

GCP firewall use case: Egress



Conditions: matching outbound connections are constructed using:

- Destination CIDR ranges
- Protocols
- Ports

Action:

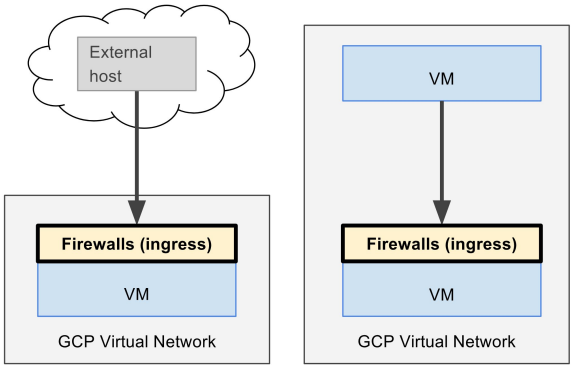
- Allow: permit the matching egress connection
- Deny: block the matching egress connection

Google Cloud

Egress *firewall* rules control outgoing connections originated inside your GCP 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. You can use destination ranges to protect from undesired connections initiated by a VM instance towards an external destination (e.g., certain IP addresses in the internet) or towards specific GCP CIDR ranges (e.g., towards a specific subnet).

The diagram illustrates a VM connecting to an external address and a VM connecting to another VM in the same network. You can control egress connections from a VM instance by constructing outbound connection conditions using destination CIDR ranges, protocols, and ports. You can then either allow the matching egress connections or deny them.

GCP firewall use case: Ingress



Conditions: matching inbound connections are constructed using:

- Source CIDR ranges
- Protocols
- Ports
- SourceTags on instances

Action:

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

Google Cloud

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 by either of the following:

- Source CIDR ranges
- SourceTags on VM instances (resolve to instance primary internal IP address)

Source CIDR ranges can be used to protect from undesired connections coming to an instance either from external networks or from GCP IP CIDR ranges. Source tags can be used to protect from undesired connections coming from specific VM instances that are tagged with a matching tag.

The diagram illustrates a VM receiving a connection from an external address and another VM receiving a connection from a VM in the same network. You can control ingress connections from a VM instance by constructing inbound connection conditions using source CIDR ranges, protocols, ports, and SourceTags on instances. SourceTags can only be used for VM-VM connections and the tag resolved to the primary IP of the VM. You can then either allow the matching ingress connections or deny them.

Agenda

- Google Cloud Platform VPC
- Projects, networks, and subnetworks
- IP addresses
- Routes and rules
- **Billing**
- Lab
- Common network designs
- Lab
- Quiz

Google Cloud

Network billing

Traffic type	Price
Ingress	No charge
Egress to the same zone	No charge
Egress to a different GCP service within the same region	No charge
Egress to Google products (YouTube, Maps, Drive) from a VM in GCP with public or private IP addresses	No charge
Egress between zones in the same region (per GB)	$0.01
Egress between regions within the US (per GB)	$0.01
Egress between regions, not including traffic between US regions	At internet egress rates

Google Cloud

https://cloud.google.com/compute/pricing#network

Note regarding "between zones in the same region": if you have a high-availability solution with components in different zones, and part of the design requires making frequent RPC calls from one zone to the other, zone egress costs could accumulate. In this circumstance, consider using Cloud Pub/Sub instead of the RPC calls for a more efficient solution.

For more information on internet egress rates, please see: https://cloud.google.com/compute/pricing#internet_egress

Network considerations

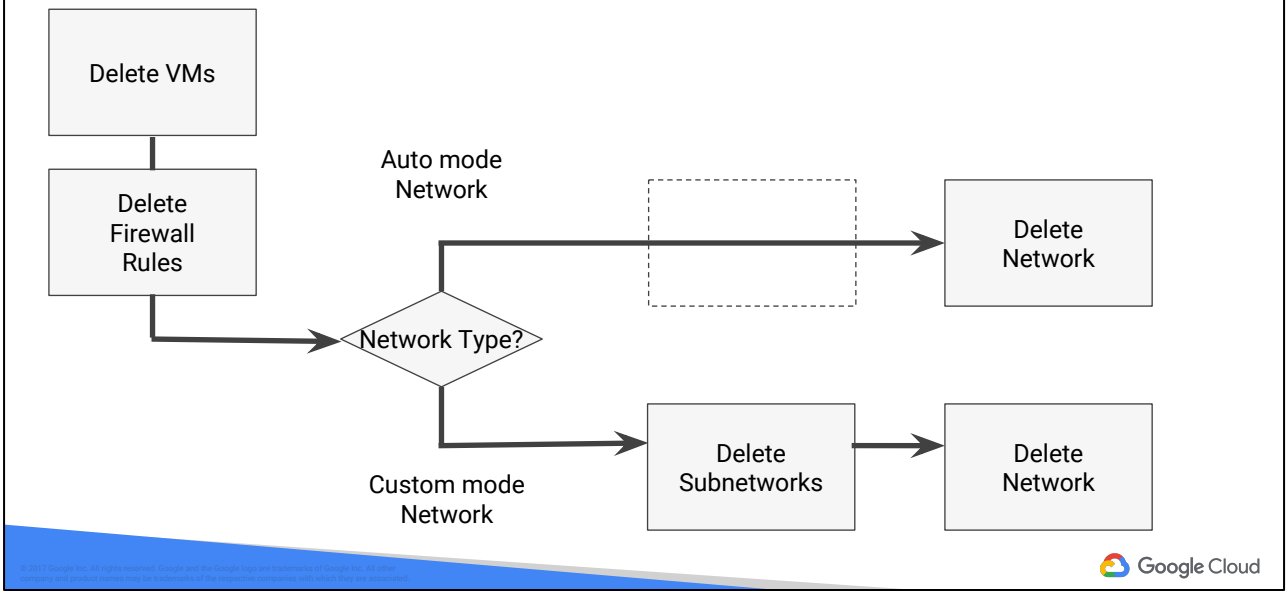
- VPC throughput and round-trip latency between VMs
  - Varies with location
  - Consider requirements
  - See documentation for current specifics
- VPC is constantly evolving
  - Any feature marked BETA has no Service Level Agreement (SLA)
  - For more information, see SLA for specific feature

Google Cloud

Throughput and latency are critical factors in conventional networks.
A VPC's high-throughput and low round-trip latency may influence these factors in your design.
Check your application's requirements against current specifics for VPC.

Note:
VM-to-VM within a single zone has much better performance and much more consistent performance than VM-to-VM between regions in a single continent or communications that span continents.

How to delete networks and subnetworks



- Subnetworks
  - all VMs must be deleted first
  - all firewall rules must be deleted first
  - can be deleted on custom-mode networks
  - cannot be deleted on auto-mode networks
- Networks
  - All VMs and firewall rules on all subnetworks must be deleted first
  - On an auto-mode network, only the entire network can be deleted, not the subnetworks.

Agenda

- Google Cloud Platform VPC
- Projects, networks, and subnetworks
- IP addresses
- Routes and rules
- Billing
- **Lab**
- Common network designs
- Lab
- Quiz

Google Cloud

Lab: Virtual Networking

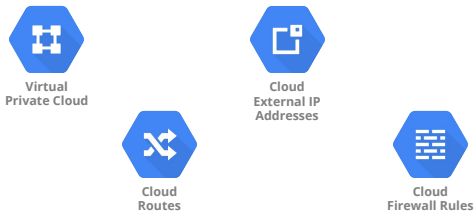
Objectives

In this lab, you learn how to perform the following tasks:

- Create an auto-mode network, a custom-mode network, and associated subnetworks
- Compare connectivity in the various types of networks
- Create routes and firewall rules using IP address and tags to enable connectivity
- Convert an auto-mode network to a custom-mode network
- Create, expand, and delete subnetworks

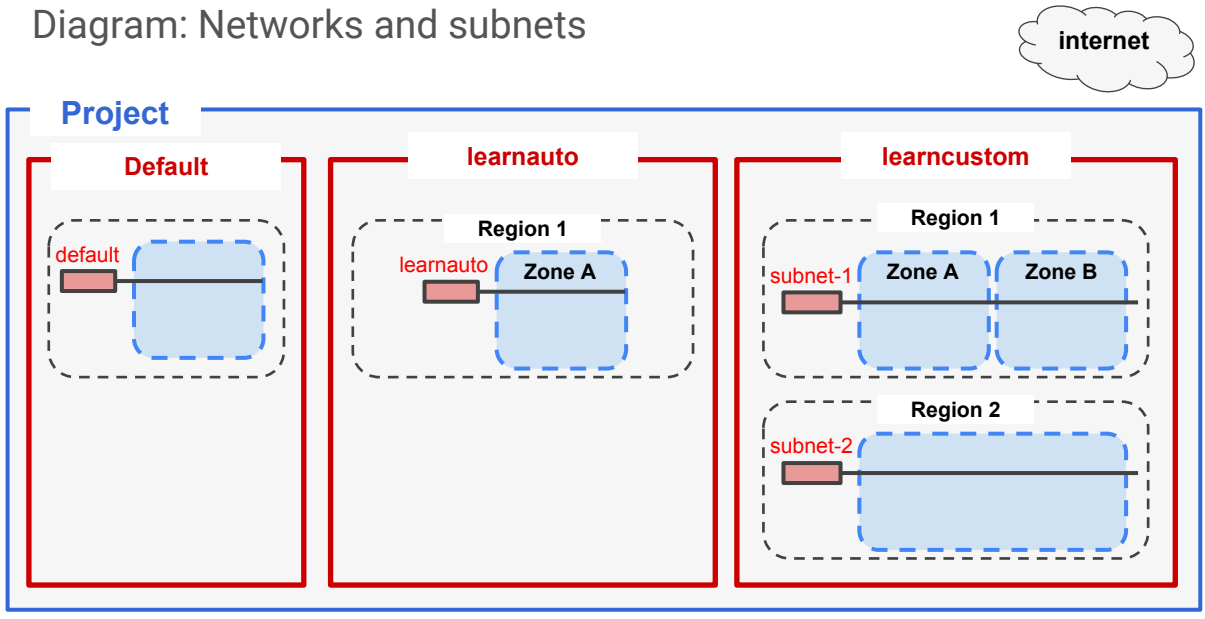
Completion: 45 minutes

Access: 90 minutes



Google Cloud

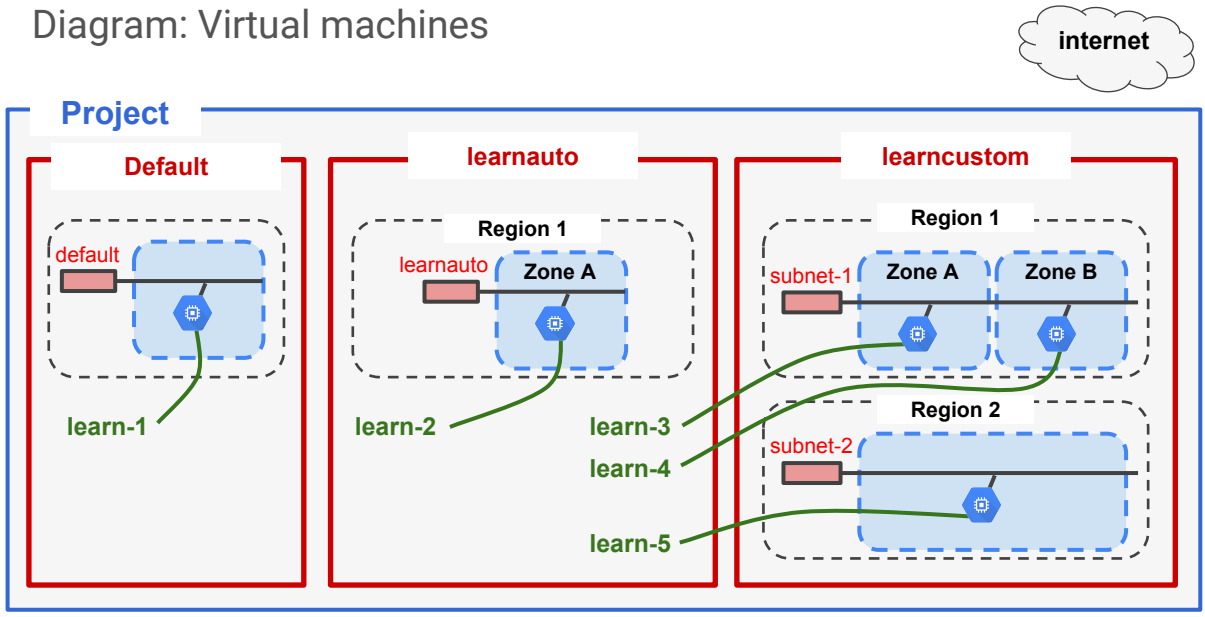
Diagram: Networks and subnets



Google Cloud

In the first part of the lab you will build this complex multiple-network topology.

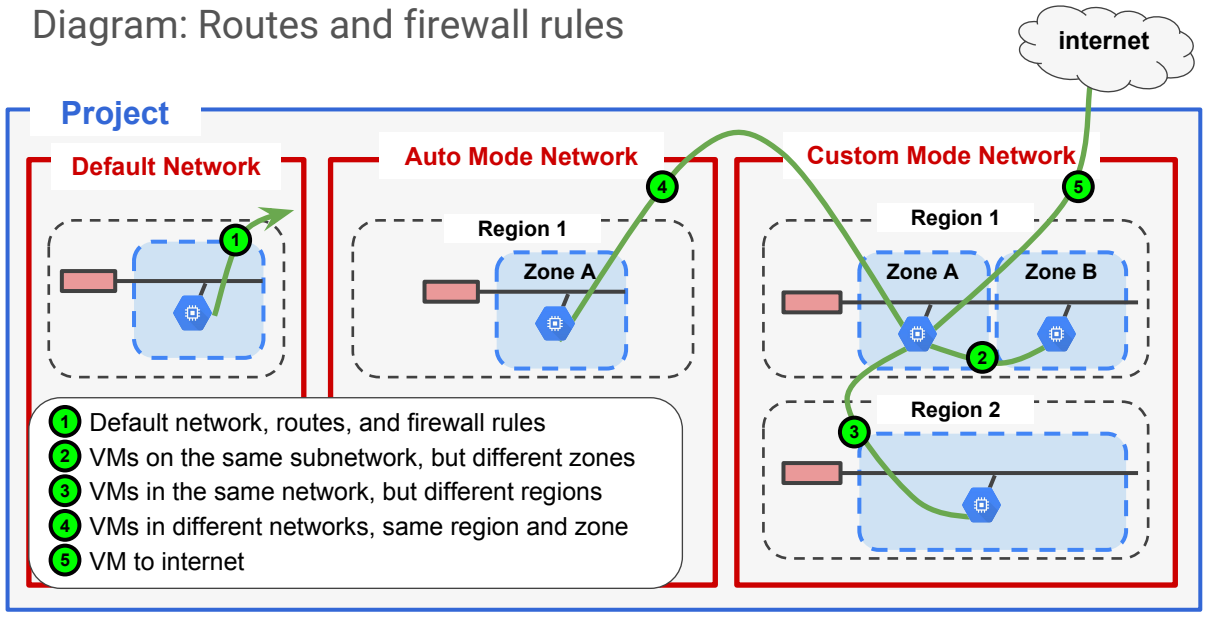
Diagram: Virtual machines



Google Cloud

In the second part of the lab, you will launch VMs in the various regions and subnets. Having the VMs in a variety of locations will enable you to explore connectivity across and within the multiple networks.

Diagram: Routes and firewall rules



Google Cloud

In the third part of the lab, you will use ping, traceroute, and ssh to test connectivity. You will modify the firewall rules to meet policy requirements, and test to verify that the changes worked.

Lab Review

In this lab, you:

- Created networks and subnetworks of many different varieties.
- Started VMs in each location.
- Explored the network relationship between them.

Agenda

- Google Cloud Platform VPC
- Projects, networks, and subnetworks
- IP addresses
- Routes and rules
- Billing
- Lab
- **Common network designs**
- Lab
- Quiz

Google Cloud

Common network designs

How do all these elements work together?

- Projects
- Networks
- Subnetworks
- Regions
- Zones

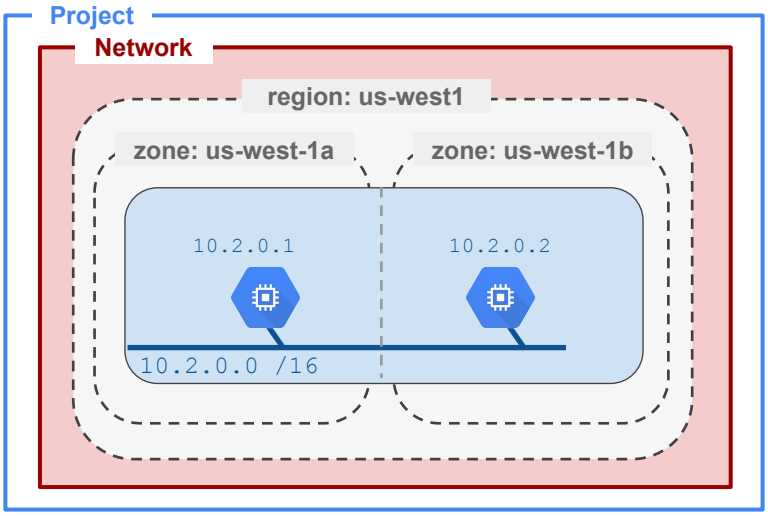
They provide a rich set of alternatives for managing groups of resources with varying availability and access control requirements.

Availability

One project
One network
One region
One subnetwork
Multiple zones

Increased availability due to multiple zones

Simplified security due to a single subnetwork

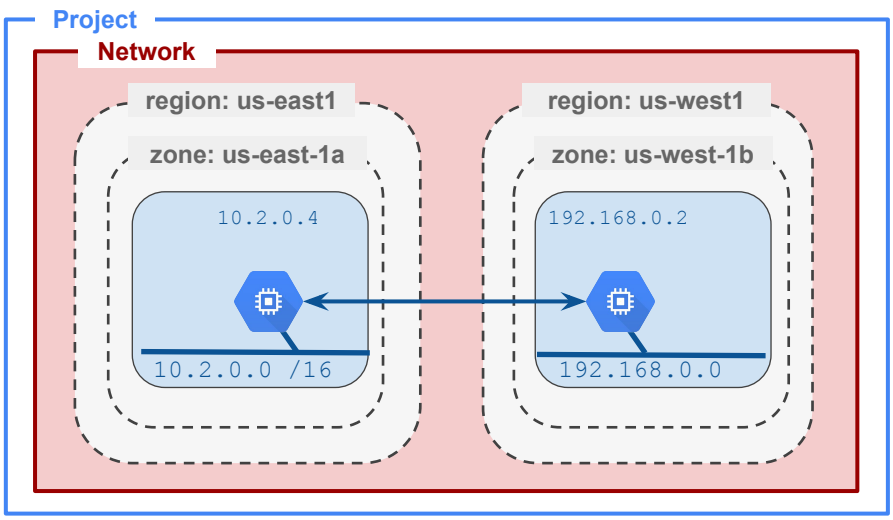


Google Cloud

The point here is that you can write firewall rules that apply to all VMs on a single subnet. By allocating VMs on a single subnet to separate zones, you get improved availability without additional security complexity.

10.2.0.0 crosses zones within a region. Allocating the VMs in different zones provides fault isolation and increases availability.
Additionally, a firewall rule can be written against subnetwork `10.2.0.0`, and it will apply to both servers even though they are in different zones.
Routes and firewalls are global resources in GCP VPC.

Globalization



One project
One network
Multiple regions
Multiple zones
Two subnetworks

Increased availability through globalization

Subnetworks cannot span regions

Google Cloud

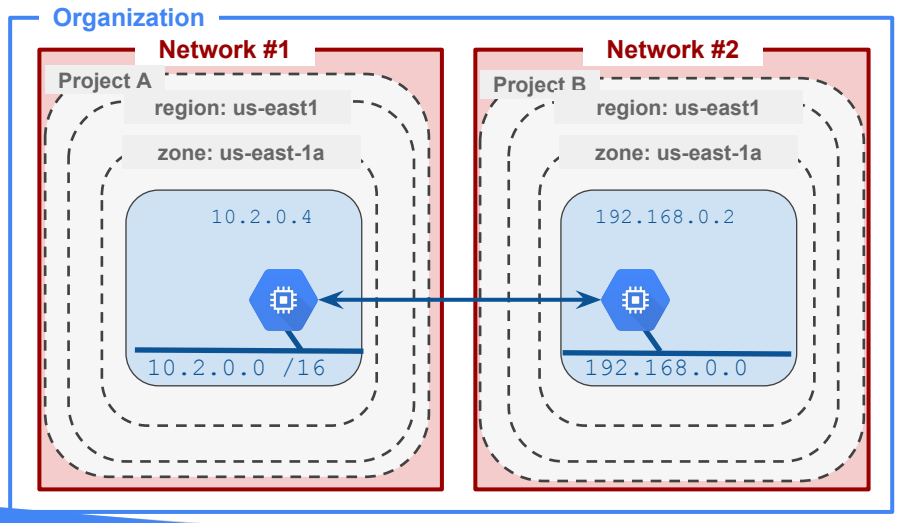
Putting resources in different zones in a region provides isolation from many types of infrastructure, hardware, and software failures. Putting resources in different regions provides an even higher degree of failure independence. This allows you to design robust systems with resources spread across different failure domains.

In this example, the VMs are now in two different regions. Subnetworks are limited to a single region. So you can't get the benefits of a single subnetwork across regions, as you could with the single subnetwork encompassing zones.

If you want global availability—alternatives and failover VMs that are in a different geographic region—there is a bit more complexity involved. Notice that because these VMs are in a single network, even though they are in different regions, they can still communicate through GCP's internal global network.

For more information, see:
https://cloud.google.com/compute/docs/regions-zones/global-regional-zonal-resources

Cross-project VPC network peering



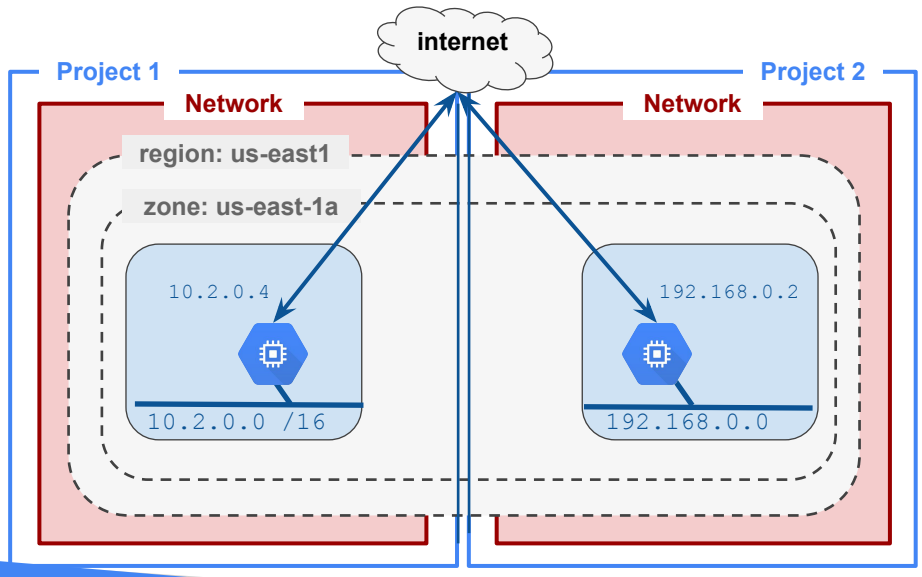
Multiple projects
Multiple networks
Multiple regions
Multiple zones
Multiple Subnetworks

Sharing enables collaboration between parts, while isolation prevents the compromise of one part from spreading to other parts.

Google Cloud

VPCs communicate over Private RFC1918 Address Space. Projects are isolated in separate VPCs. Using network peering, they can communicate over a private address space.

Management separation



Multiple projects
Multiple networks
Single region
Single zone
Multiple subnetworks

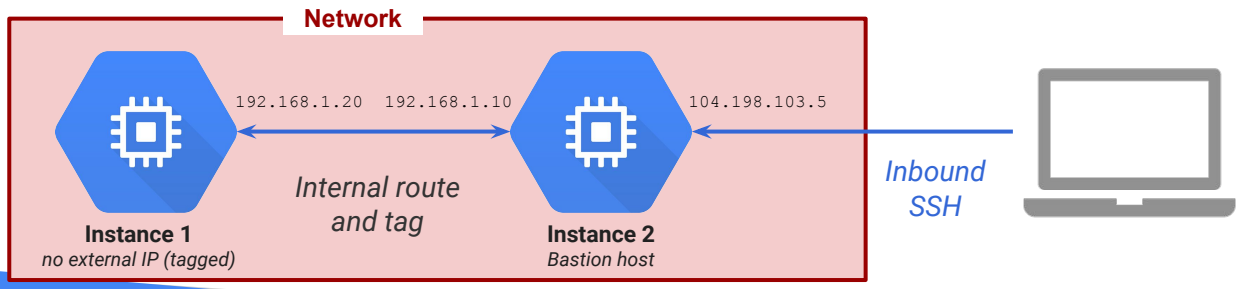
Separate projects means more fine-grained access control.

Google Cloud

Finally, in this case, the VMs are isolated in separate projects. This can be useful for Identity and Access Management.
For example, if Software Development is Project 1 and Test Engineering is Project 2, you can assign different people to different roles in the projects.
Consider dividing a system up into multiple projects for better access control. But remember that a network cannot span projects, so using separate projects implies that the VMs must communicate via the internet.

Bastion host isolation

- Instance used as “jump host”
  - External connections via SSH, used to connect to internal instances
- Be sure to harden bastion host
  - Limit CIDR range of source IPs connecting to bastion
  - Firewall rules allow SSH traffic to private instances only from bastion



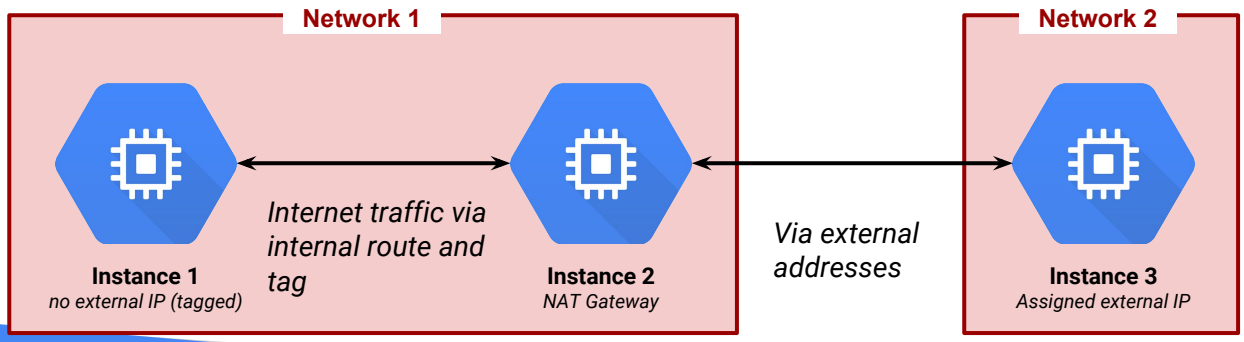
Google Cloud

For more information about configuring a bastion host, see: https://cloud.google.com/solutions/connecting-securely#bastion.

Best Practice
Use VPN or some other more secure form of connection for ordinary activities.
Use SSH with the bastion host as the maintenance avenue of last resort.

NAT gateway host isolation

- Instance 1 has no external IP address and is tagged to match the route
- Instance 2 is configured as a NAT gateway with IP forwarding
- Instance 1 communicates with instance 3 via gateway
- Note: Networks 1 and 2 could be in the same or separate projects



1. Create a Compute Engine instance to act as the gateway
   - Enable IP forwarding with the --can-ip-forward flag
2. Create additional Compute Engine instances with no external IP address
   - Disable external IP addresses with the --no-address flag
   - Also add a tag to identify instances that will use the gateway routing
3. Create a route to send traffic destined to the internet through your gateway instance
4. Log in to the gateway instance and configure iptables to NAT internal traffic to the internet

In a legacy network, you might have the requirement to set up an instance as a NAT gateway.

For more information on configuring a NAT gateway and other special configurations, see: https://cloud.google.com/compute/docs/vpc/special-configurations#natgateway

Agenda

- Google Cloud Platform VPC
- Projects, networks, and subnetworks
- IP addresses
- Routes and rules
- Billing
- Lab
- Common network designs
- **Lab**
- Quiz

Google Cloud

Lab: Bastion Host

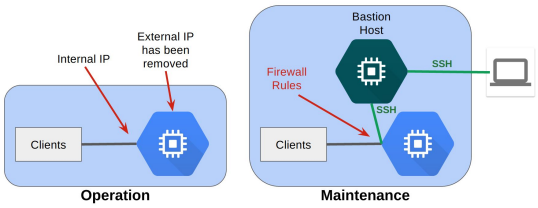
Objectives

In this lab, you learn how to perform the following tasks:

- Create an application web server to represent a service provided to an internal corporate audience
- Prevent the web server from access to or from the internet
- Create a maintenance server, called a *bastion host*, to gain access to and verify internal connectivity to the application server

Completion: 20 minutes

Access: 40 minutes



Lab Review

In this lab, you:

- Restricted access to the webserver VM by removing the external IP address.
- Created a bastion host named *bastion* to gain access to the webserver VM over its internal IP.

Normally, you would harden the bastion host by restricting the source IPs that can access the bastion host, by editing the firewall rules just as you did earlier in this lab. When you're not using the bastion host, you can shut it down.

Agenda

- Google Cloud Platform VPC
- Projects, networks, and subnetworks
- IP addresses
- Routes and rules
- Billing
- Lab
- Common network designs
- Lab
- **Quiz**

Google Cloud

Quiz

What is a key distinguishing feature of networking in the Google Cloud Platform?

1. Unlike other cloud networks, access lists and firewall rules are available.
2. Network topology is not dependent on address layout.
3. Data can be tagged for "late delivery," and it will be delivered on the time and day you set.
4. Supports IPv4 addresses that conform to RFC 1918.

Quiz

What is a key distinguishing feature of networking in the Google Cloud Platform?

1. Unlike other cloud networks, access lists and firewall rules are available.
2. Network topology is not dependent on address layout. *
3. Data can be tagged for "late delivery," and it will be delivered on the time and day you set.
4. Supports IPv4 addresses that conform to RFC 1918.

Explanation:
Networks have no IP address range in Google Cloud, so subnetworks DO NOT have to "fit under" the network range in a hierarchical format defined by the extension of the network mask bits. This breaks with traditional networking and also with the way that networking has been implemented in other clouds.

Quiz

What are the three types of networks offered in the Google Cloud Platform?

1. Zonal, regional, and global
2. Gigabit network, 10-gigabit network, and 100-gigabit network
3. Default network, auto-mode network, and custom-mode network
4. IPv4 unicast network, IPv4 multicast network, IPv6 network

Quiz

What are the three types of networks offered in the Google Cloud Platform?

1. Zonal, regional, and global
2. Gigabit network, 10 gigabit network, and 100 gigabit network
3. Default network, auto-mode network, and custom-mode network *
4. IPv4 unicast network, IPv4 multicast network, IPv6 network

Explanation:
The default network established fixed standard subnetworks with predefined IP ranges and it is fast to set up. The auto-mode network uses the same subnet IP ranges as the default-type, with a network name other than default. And custom-mode allows you to specify the IP ranges of subnets.

Quiz

What is one benefit of applying firewall rules by tag rather than by address?

1. Tags help organizations track firewall billing.
2. Tags in network traffic help with network sniffing.
3. Tags on firewall rules control which ephemeral IP addresses VMs will receive.
4. When a VM is created with a matching tag, the firewall rules apply irrespective of the IP address it is assigned.

Quiz

What is one benefit of applying firewall rules by tag rather than by address?

1. Tags help organizations track firewall billing.
2. Tags in network traffic help with network sniffing.
3. Tags on firewall rules control which ephemeral IP addresses VMs will receive.
4. When a VM is created with a matching tag, the firewall rules apply irrespective of the IP address it is assigned. *

Explanation:
When a VM is created, the ephemeral external IP address is assigned from a pool. There is no way to predict which address will be assigned, so there is no way to write a rule that will match that VM's IP address before it is assigned. Tags allow a symbolic assignment that does not depend on order in the IP addresses. It makes for simpler, more general, and easier-to-maintain firewall rules.

More resources

Using networks and firewalls
https://cloud.google.com/compute/docs/networking

Using subnetworks
https://cloud.google.com/compute/docs/subnetworks

Google Cloud