GCP Certification Series: 3.5 Deploying and implementing networking resources.

Prashanta Paudel

Nov 5, 2018 · 28 min read

If we look closely the most important part of Cloud platform is its network infrastructure as all of the things from instances to API's work because there is a network between them.

Cloud basically is a remote network of computing devices or instances provided to the user via the network that connects them to the user.

We already have some basic concept of how vast is Google's network around the world and how are they connected but let's review the main points for the network.

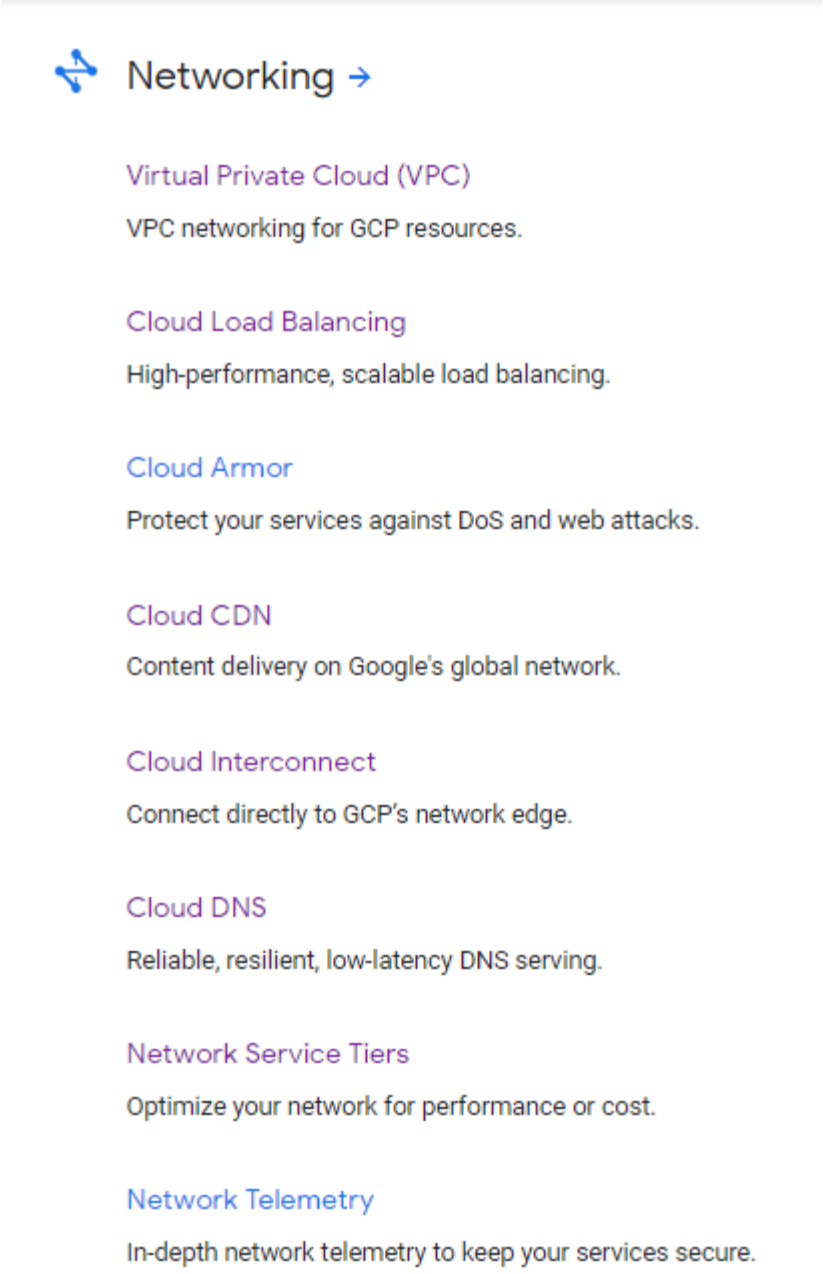
Google is probably tier 1 or tier 2 network provider but as they only transfer traffic within Google's network they are not regarded as ISP. Google has a private global network connecting many data centers and POP(point of presence). They have

---

reference: https://cloud.google.com/about/locations/#network-tab

Google's private network connects their regional locations to more than 100 points of presence(POP). Google Cloud Platform uses software-defined networking and distributed systems technologies to host and deliver services around the world. Since Google has a global private network, this will help make your product global linking all the regions by the high-speed network.

Google's Network Elements in the cloud are:



Reference: https://cloud.google.com/products/

Regions and Zones

When developing your application in GCP it is very important to understand regions and zones,

Resources are also regional and zonal so you must also have an idea about which resource is what before going in detail.

A region is a geographical location that is sub-divided into zones.

While few of the resources in GCP are global, others may be restricted by region or zone.

Regional resources can be used anywhere within the same region, while zonal resources can be used anywhere within the same zone. Some examples of this are:

Global Resources:

- Images
- Snapshots
- VPC Network
- Firewalls
- Routes

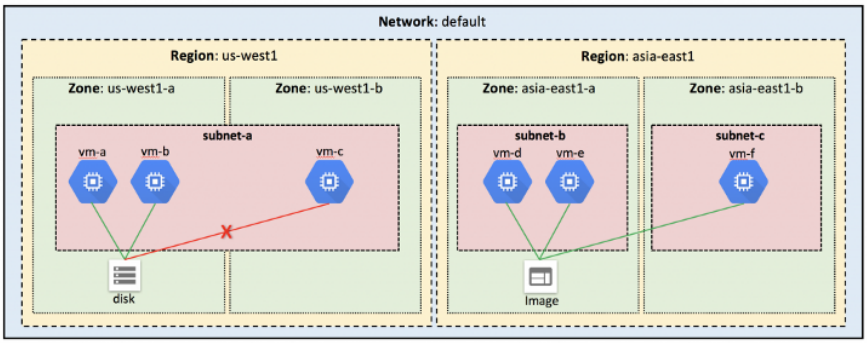
Regional Resources:

- Static external IP addresses
- Subnets

Zonal Resources:

- Instances (VMs)
- Persistent Disks

For example, I can attach a disk from one instance to another within the same zone, but I cannot do this across zones. However, since images and snapshots are Global Resources, I can use these across zones in the same region.



reference: https://www.networkmanagementsoftware.com/google-cloud-platform-gcp-networking-fundamentals/

Virtual Private Cloud

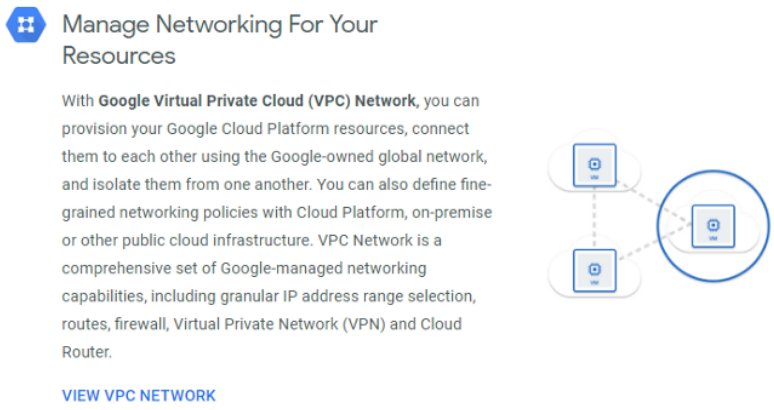
With the help of Google Virtual private cloud (VPC), you can

- provision GCP resources
- connect resources

- isolate resources
- make fine-grained policies for accessing resources and network

VPC consists of

- IP Address
- firewall
- VPN
- cloud router



Reference: https://cloud.google.com/products/networking/

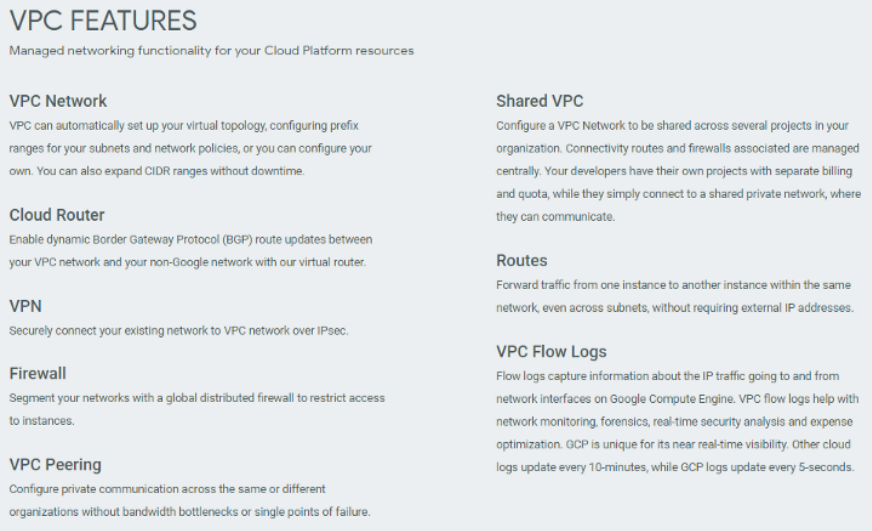
A Private Space within Google Cloud Platform

Virtual Private Cloud (VPC) gives you the flexibility to scale and control how workloads connect regionally and globally. When you connect your on-premises or remote resources to GCP, you'll have global access to your VPCs without needing to replicate connectivity or administrative policies in each region.

VPC is

- **Global**
- **Shareable**
- **Expandable**
- **Private**

- **Transparent**

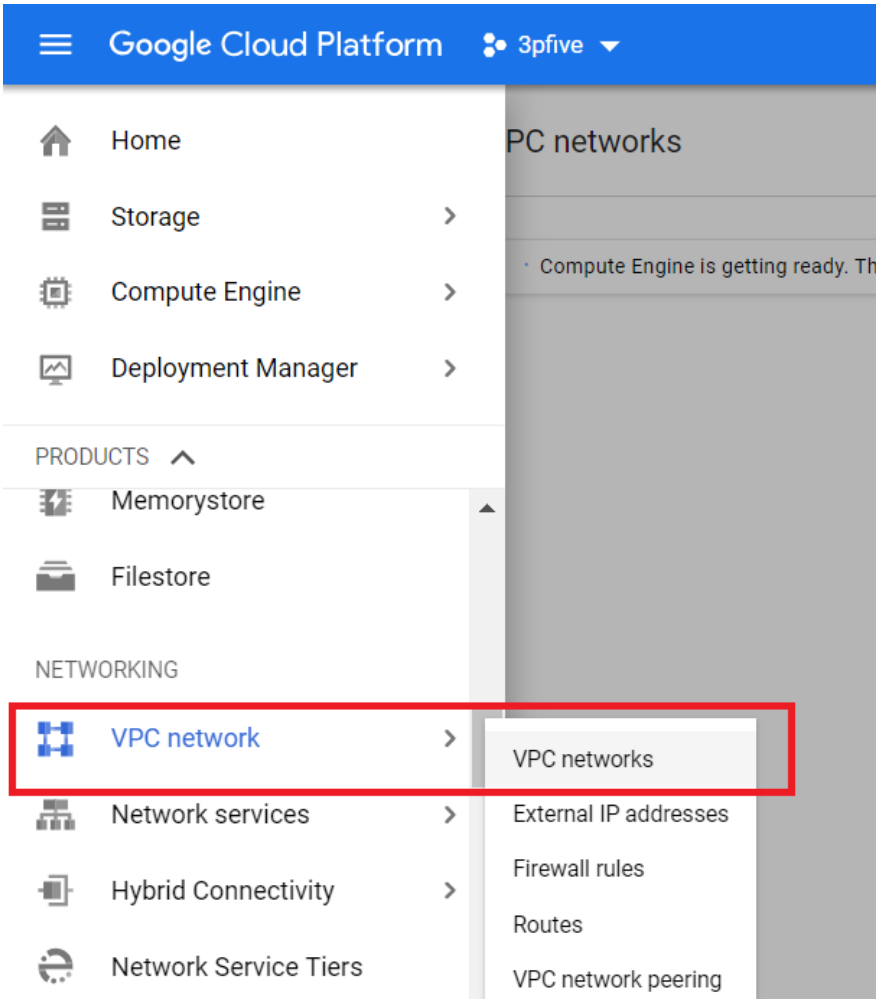


Reference: https://cloud.google.com/vpc/

Creating a VPC with subnets. (e.g., Custom-mode VPC, Shared VPC)

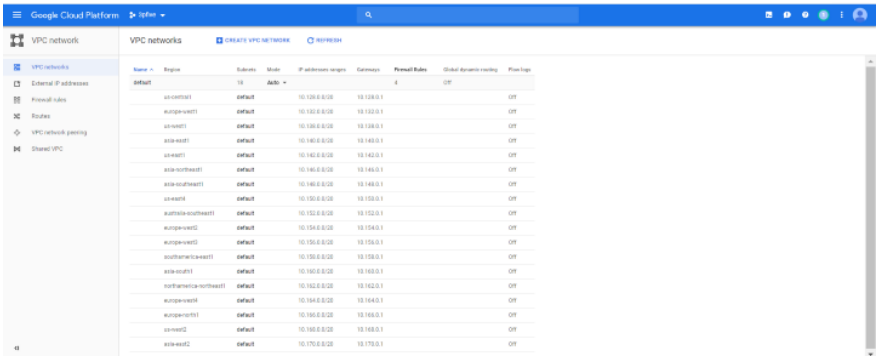
For creating a Virtual private cloud first you must have a Project setup done. Please follow my earlier blogs to set up a project in GCP.

After creating a project, click on the VPC network under NETWORKING and go to the Dashboard



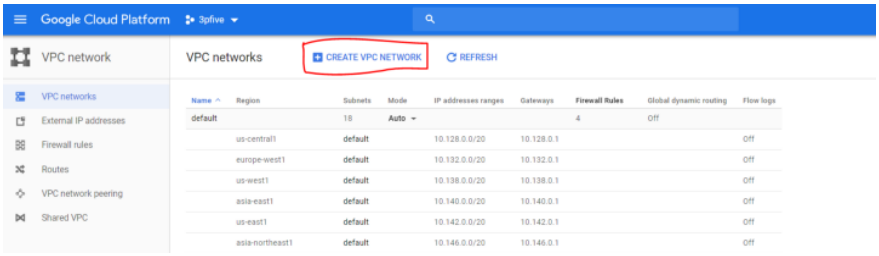
VPC networks

Then you will see a default VPC network ranges.

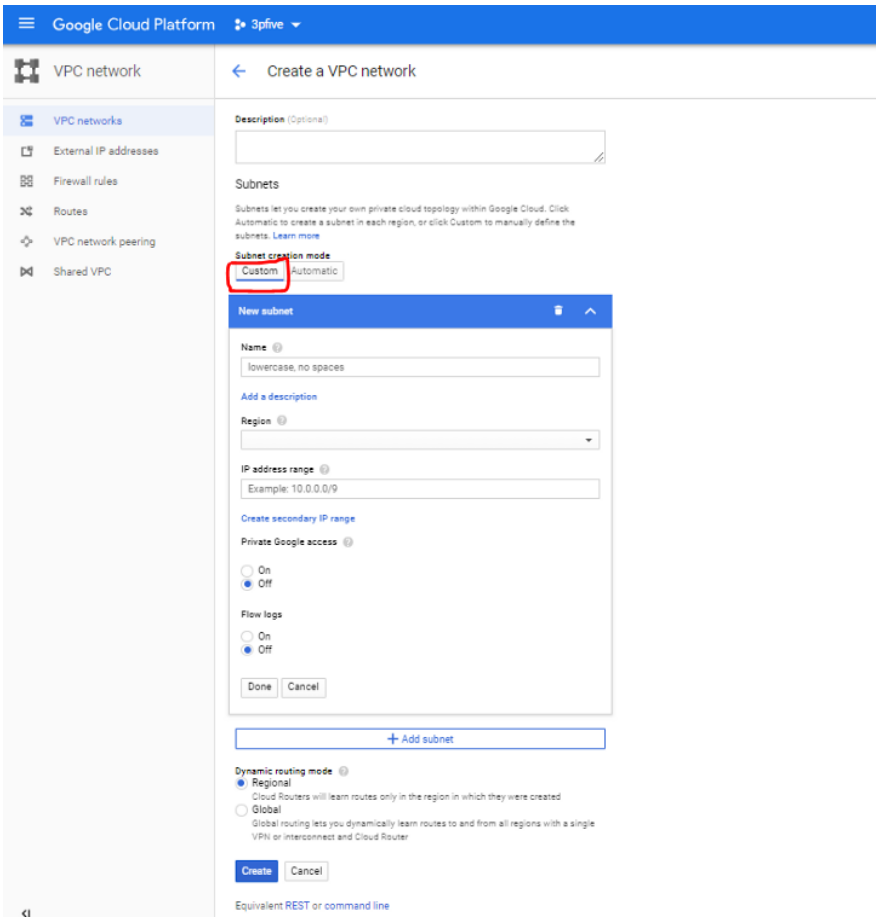


Default VPC networks

Now click on CREATE VPC NETWORK on top of the page



Then you will be presented with a VPC setup page



VPC configuration

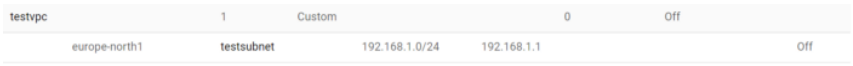
On description, you can name anything you can remember easily.

Now you have two option,

- **Automatic Subnet**
- **Custom Subnet**

Automatic subnet will create one subnet in each region whereas custom will create only on those regions and with IP you declare explicitly.

Let's go for **Custom** at the moment and define 192.168.0.1/24 network



custom subnet

Virtual Private Cloud (VPC) Network Overview

A VPC network, sometimes just called a "network," is a virtual version of a physical network, like a data center network. It provides connectivity for your Compute Engine virtual machine (VM) instances, Kubernetes Engine clusters, App Engine Flex instances, and other resources in your project.

Projects can contain multiple VPC networks. **New projects start with a `default` network that has one subnet in each region (an auto mode network).**

Specifications

VPC networks have the following properties:

- VPC networks, including their associated routes and firewall rules, are global resources. They are *not* associated with any particular region or zone.
- *Subnets* are regional resources. Each subnet defines a range of IP addresses. For more information about networks and subnets, see networks and subnets.
- Traffic to and from instances can be controlled with network firewall rules.
- Resources within a VPC network can communicate with one another using internal (private) IPv4 addresses, subject to applicable network firewall rules. For more information, see communication within the network.

- Instances with internal IP addresses can communicate with Google APIs and services. For more information, see Private Access Options.
- Network administration can be secured using Identity and Access Management (IAM) roles.
- An organization can use Shared VPC to keep a VPC network in a common host project. Authorized IAM members from other projects in the same organization can create resources that use subnets of the Shared VPC network.
- VPC networks can be connected to other VPC networks in different projects or organizations by using VPC Network Peering.
- VPC networks can be securely connected in hybrid environments using Cloud VPN or Cloud Interconnect.
- VPC networks only support IPv4 unicast traffic. They do **not** support broadcast, multicast, or IPv6 traffic *within* the network. However, IPv6 can be used to *reach* resources in the network. For example, IPv6 addresses can be assigned to a global load balancer, and the App Engine standard environment supports IPv6.

Networks and subnets

Each VPC network consists of one or more useful IP range partitions called *subnetworks* or *subnets*. Each subnet is associated with a region. Networks can contain one or more subnets in any given region. Auto mode networks create subnets in each region automatically. Custom mode networks start with no subnets, giving you full control over subnet creation. For information about the differences between auto and custom mode networks, see types of VPC networks.

By themselves, VPC networks do not have any IP address ranges associated with them. When you create a subnet, you must define a primary IP address range. You can optionally define one or more *secondary ranges*:

- **Primary range**: You can choose any private RFC 1918 CIDR block for the primary IP address range of the subnet, subject to these rules. Your subnets don't need to form a predefined contiguous CIDR block, but you can do that if desired. For example, auto

mode networks do create subnets that fit within a predefined auto mode IP range.

- **Secondary ranges**: You can define up to five secondary IP address ranges for use with IP aliasing.

When you create a resource in GCP, you choose a network and subnet. For resources other than instance templates, you also select a zone or a region. Selecting a zone implicitly selects its parent region. Because subnets are regional objects, the region you select for a resource determines the subnets it can use:

- The process of creating an instance involves selecting a zone, a network, and a subnet. The subnets available for selection are restricted to those in the selected region. GCP assigns the instance an IP address from the range of available addresses in the subnet.
- The process of creating a managed instance group involves selecting a zone or region, depending on the group type, and an instance template. The instance templates available for selection are restricted to those whose defined subnets are in the same region selected for the managed instance group.
- Instance templates are global resources. The process of creating an instance template involves selecting a network and a subnet. If you select an auto mode network, you can choose “auto subnet” to defer subnet selection to one that is available in the selected region of any managed instance group that would use the template, because auto mode networks have a subnet in every region by definition.
- The process of creating a Kubernetes container cluster involves selecting a zone or region (depending on the cluster type), a network, and a subnet. The subnets available for selection are restricted to those in the selected region.

Network and subnet terminology

The terms “subnet” and “subnetwork” are synonymous. They are used interchangeably in the GCP Console, `gcloud` commands, and API documentation.

Note: A subnet or subnetwork is **not** the same thing as a (VPC) network. **Networks and subnets (subnetworks) are *different types* of objects in GCP.**

Types of VPC networks

There are two types of VPC networks:

- When an **auto mode** network is created, one subnet from each region is automatically created within it. These automatically created subnets use a set of predefined IP ranges which fit within the `10.128.0.0/9` CIDR block. As new GCP regions become available, new subnets in those regions are automatically added to auto mode networks using an IP range from that block. In addition to the automatically created subnets, you can add more subnets manually to auto mode networks, in regions you choose, using IP ranges outside of `10.128.0.0/9`.
- When a **custom mode** network is created, no subnets are automatically created. 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.

Each project starts with a `default` auto mode network.

You can switch a network from auto mode to custom mode. This conversion is one-way; custom mode networks cannot be changed to auto mode networks. Carefully review the considerations for auto mode networks to help you decide which type of network meets your needs.

Considerations for auto mode networks

Auto mode networks are easy to set up and use, and they are well suited for use cases with these attributes:

- Having subnets automatically created in each region is useful.
- The predefined IP ranges of the subnets do not overlap with IP ranges you would use for different purposes (for example, Cloud VPN connections to on-premises resources).

However, custom mode networks are more flexible and are better suited to production. The following attributes highlight use cases where custom mode networks are recommended or required:

- Having one subnet automatically created in each region isn't necessary.
- Having new subnets automatically created as new regions become available could overlap with IP addresses used by manually created subnets or static routes, or could interfere with your overall network planning.
- You need complete control over the subnets created in your VPC network, including regions and IP address ranges used.
- You plan to connect VPC networks using VPC Network Peering or Cloud VPN. Because the subnets of every auto mode network use the same predefined range of IP addresses, you cannot connect auto mode networks to one another.

Important: Production networks should be planned in advance. It's recommended that you use custom mode networks in production.

Subnets and IP ranges

IP ranges can be assigned to subnets you create according to these rules:

- Each subnet must have a primary address range, which is a valid RFC 1918 CIDR block.
- Subnets in the same network must use unique IP ranges. Subnets in *different* networks, even in the same project, can re-use the same IP address ranges.
- When you create a subnet manually, you can use any RFC 1918 CIDR range subject to these restrictions:
- Subnets in the same GCP network must have unique IP ranges.
- IP ranges for all subnets must be unique among VPC networks that are connected to one another by VPC Network Peering or Cloud VPN.

- IP ranges for on-premises networks connected via Cloud VPN or Cloud Interconnect should not conflict with IP ranges of any subnet. Subnet routes must have the most specific destination.
- IP ranges used by subnets cannot otherwise conflict with ones referenced by a static route.
- When creating additional subnets in an auto mode network, your manually-created subnets **must** use an IP range *outside* of the `10.128.0.0/9` CIDR block. That block is reserved for the primary IP ranges of automatically created subnets.
- You can assign one or more secondary IP ranges to a subnet. These secondary ranges are reserved for VM instances in the subnet that are configured with IP aliases. Secondary ranges can be any RFC 1918 CIDR block subject to the same restrictions discussed in the previous point.
- IP ranges do **not** need to be contiguous from subnet to subnet in the same network.
- IP ranges for subnets in the same network do **not** have to be a member of a larger contiguous CIDR block. For example, one subnet can use `10.0.0.0/8` while another subnet in the same network can use `192.168.0.0/16`.
- The minimum CIDR size for a subnet is `/29`.

Reserved IPs

Every subnet has four reserved IP addresses in its primary IP range:

Reserved AddressDescriptionExampleNetworkFirst address in the primary IP range for the subnet `10.1.2.0` in `10.1.2.0/24` Default GatewaySecond address in the primary IP range for the subnet `10.1.2.1` in `10.1.2.0/24` Second-to-last Reservation Second-to-last address in the primary IP range for the subnet `10.1.2.254` in `10.1.2.0/24` BroadcastLast address in the primary IP range for the subnet `10.1.2.255` in `10.1.2.0/24`

Note: There are no reserved IP addresses in the secondary IP ranges.

Auto mode IP ranges

This table lists the IP ranges for the automatically created subnets in an auto mode network. IP ranges for these subnets fit inside the `10.128.0.0/9` CIDR block. Auto mode networks are built with one subnet per region at creation time, and will automatically receive new subnets in new regions. Hence, unused portions of `10.128.0.0/9` are reserved for future GCP use.

RegionIP Range (CIDR)Default GatewayUsable Addresses (Inclusive)asia-east110.140.0.0/2010.140.0.110.140.0.2 to 10.140.15.253asia-east210.170.0.0/2010.170.0.110.170.0.2 to 10.170.15.253asia-northeast110.146.0.0/2010.146.0.110.146.0.2 to 10.146.15.253asia-south110.160.0.0/2010.160.0.110.160.0.2 to 10.160.15.253asia-southeast110.148.0.0/2010.148.0.110.148.0.2 to 10.148.15.253australia-southeast110.152.0.0/2010.152.0.110.152.0.2 to 10.152.15.253europe-north110.166.0.0/2010.166.0.110.166.0.2 to 10.166.15.253europe-west110.132.0.0/2010.132.0.110.132.0.2 to 10.132.15.253europe-west210.154.0.0/2010.154.0.110.154.0.2 to 10.154.15.253europe-west310.156.0.0/2010.156.0.110.156.0.2 to 10.156.15.253europe-west410.164.0.0/2010.164.0.110.164.0.2 to 10.164.15.253northamerica-northeast110.162.0.0/2010.162.0.110.162.0.2 to 10.162.15.253southamerica-east110.158.0.0/2010.158.0.110.158.0.2 to 10.158.15.253us-central110.128.0.0/2010.128.0.110.128.0.2 to 10.128.15.253us-east110.142.0.0/2010.142.0.110.142.0.2 to 10.142.15.253us-east410.150.0.0/2010.150.0.110.150.0.2 to 10.150.15.253us-west110.138.0.0/2010.138.0.110.138.0.2 to 10.138.15.253us-west210.168.0.0/2010.168.0.110.168.0.2 to 10.168.15.253

Routes and firewall rules

Routes

Routes define paths for packets leaving instances (egress traffic). Routes in GCP are divided into two categories: system-generated and custom. This section briefly describes the two types of system generated routes. You can create custom routes in your network as well. See the routes overview for complete details about routing in GCP.

Every new network starts with two types of system-generated routes:

- The default route defines a path for traffic to leave the VPC network. It provides general Internet access to VMs that meet the Internet access requirements. It also provides the typical path for Private Google Access.
- A subnet route is created for each of the IP ranges associated with a subnet. Every subnet has at least one subnet route for its primary IP range, and additional subnet routes are created for a subnet if you add secondary IP ranges to it. Subnet routes define paths for traffic to reach VMs that use the subnets.

Important: You cannot remove subnet routes manually. Refer to the subnet routes section of the routes overview for details about how subnet routes are created or deleted.

Dynamic routing mode

Each VPC network has an associated *dynamic routing mode* that controls the behavior of all if its Cloud Routers. Cloud Routers share routes to your VPC network and learn custom dynamic routes from connected networks when you connect your VPC network to another network with a Cloud VPN tunnel using dynamic routing, Dedicated Interconnect, or Partner Interconnect.

- **Regional dynamic routing** is the default. In this mode, routes to on-premises resources learned by a given Cloud Router in the VPC network only apply to the subnets in the same region as the Cloud Router. Unless modified by custom advertisements, each Cloud Router only shares the routes to subnets in its region with its on-premises counterpart.
- **Global dynamic routing** changes the behavior of all Cloud Routers in the network such that the routes to on-premises resources that they learn are available in all of subnets in the VPC network, regardless of region. Unless modified by custom advertisements, each Cloud Router shares routes to all subnets in the VPC network with its on-premises counterpart.

See custom advertisements for information about how the set of routes shared by a Cloud Router can be customized.

The dynamic routing mode can be set when you create a VPC network or modify it. You can change the dynamic routing mode from regional to global and vice-versa without restriction. Refer to using VPC networks for instructions.

Caution: Changing the dynamic routing mode has the potential to interrupt traffic within the network, or enable or disable routes in unexpected ways. Carefully review the role of each Cloud Router before changing the dynamic routing mode.

Firewall rules

Firewall rules apply to both outgoing (egress) and incoming (ingress) traffic in the network. Firewall rules control traffic even if it is entirely within the network, such as instance-to-instance communication.

Every VPC network has two implied firewall rules. One implied rule allows most egress traffic, and the other denies all ingress traffic. You cannot delete the implied rules, but you can override them with your own. GCP always blocks some traffic, regardless of firewall rules. For more information, see blocked traffic.

See the firewall rules overview for more information.

You can monitor which firewall rule allowed or denied a particular connection. See Firewall Rules Logging for more information.

Communication within the network

The system-generated subnet routes define the paths for sending traffic among instances within the network using internal (private) IP addresses. For one instance to be able to communicate with another, appropriate firewall rules must also be configured because every network has an implied deny firewall rule for ingress traffic.

Except for the `default` network, you must explicitly create higher priority ingress firewall rules to allow instances to communicate with one another. The `default` network includes a number of firewall rules in addition to the implied ones, including the `default-allow-internal` rule, which permits instance-to-instance communication within the network. The `default` network also comes with ingress rules allowing protocols like RDP and SSH.

Rules that come with the `default` network are also presented as options for you to apply to new auto mode networks that you create using the GCP Console.

Internet access requirements

The following criteria must be satisfied for an instance to have outgoing Internet access:

- The network must have a valid *default Internet gateway* route or custom route whose destination IP range is the most general (`0.0.0.0/0`). This route simply defines the path to the Internet. See Routes for more information about routes.
- Firewall rules must allow egress traffic from the instance. Unless overridden by a higher priority rule, the implied allow rule for egress traffic permits outbound traffic from all instances.
- One of the following must be true:
- The instance must have an external IP address. An external IP can be assigned to an instance when it is created or after it has been created.
- Another instance in the network must serve as a NAT gateway.

VPC network example

The following example illustrates a custom mode network with three subnets in two regions:

- **Subnet1** is defined as `10.240.0.0/24` in the us-west1 region.
- Two VM instances in the us-west1-a zone are in this subnet. Their IP addresses both come from the available range of addresses in *subnet1*.
- **Subnet2** is defined as `192.168.1.0/24` in the us-east1 region.
- Two VM instances in the us-east1-a zone are in this subnet. Their IP addresses both come from the available range of addresses in *subnet2*.
- **Subnet3** is defined as `10.2.0.0/16`, also in the us-east1 region.

- One VM instance in the us-east1-a zone and a second instance in the us-east1-b zone are in *subnet3*, each receiving IP addresses from its available range. Because subnets are regional resources, instances can have their network interfaces associated with any subnet in the same region that contains their zones.

Shared VPC Overview

Shared VPC allows an organization to connect resources from multiple projects to a common VPC network, so that they can communicate with each other securely and efficiently using internal IPs from that network. When you use Shared VPC, you designate a project as a *host project* and attach one or more other *service projects* to it. The VPC networks in the host project are called *Shared VPC networks*. Eligible resources from service projects can use subnets in the Shared VPC network.

Shared VPC lets organization administrators delegate administrative responsibilities, such as creating and managing instances, to Service Project Admins while maintaining centralized control over network resources like subnets, routes, and firewalls. This model allows organizations to:

- Implement a security best practice of least privilege for network administration, auditing, and access control. Shared VPC Admins can delegate network administration tasks to Network and Security Admins in the Shared VPC network without allowing Service Project Admins to make network-impacting changes. Service Project Admins are only given the ability to create and manage instances that make use of the Shared VPC network. Refer to the administrators and IAM section for more details.

- Apply and enforce consistent access control policies at the network level for multiple service projects in the organization while delegating administrative responsibilities. For example, Service Project Admins can be Compute Instance Admins in their project, creating and deleting instances that use approved subnets in the Shared VPC host project.

- Use service projects to separate budgeting or internal cost centers. Refer to the billing section for more details.

Concepts

Organizations and projects

Shared VPC connects projects within the same organization. Participating host and service projects cannot belong to different organizations. Refer to the GCP resource hierarchy for more information about organizations and projects.

A project that participates in Shared VPC is either a *host project* or a *service project*:

- A **host project** contains one or more Shared VPC networks. A Shared VPC Admin must first enable a project as a host project. After that, a Shared VPC Admin can attach one or more *service projects* to it.
- A **service project** is any project that has been attached to a host project by a Shared VPC Admin. This attachment allows it to participate in Shared VPC. It's a common practice to have multiple service projects operated and administered by different departments or teams in your organization.
- A project cannot be both a host and a service project simultaneously. Thus, a service project cannot be a host project to further service projects.
- You can create and use multiple host projects; however, each service project can only be attached to a single host project. See the Multiple host projects example for an illustration.

For clarity, a project that does not participate in Shared VPC is called a **standalone project**. This emphasizes that it is neither a host project nor a service project.

Networks

A **Shared VPC network** is a VPC network defined in a host project and made available as a centrally shared network for eligible resources in service projects. Shared VPC networks can be either auto or custom mode, but legacy networks are not supported.

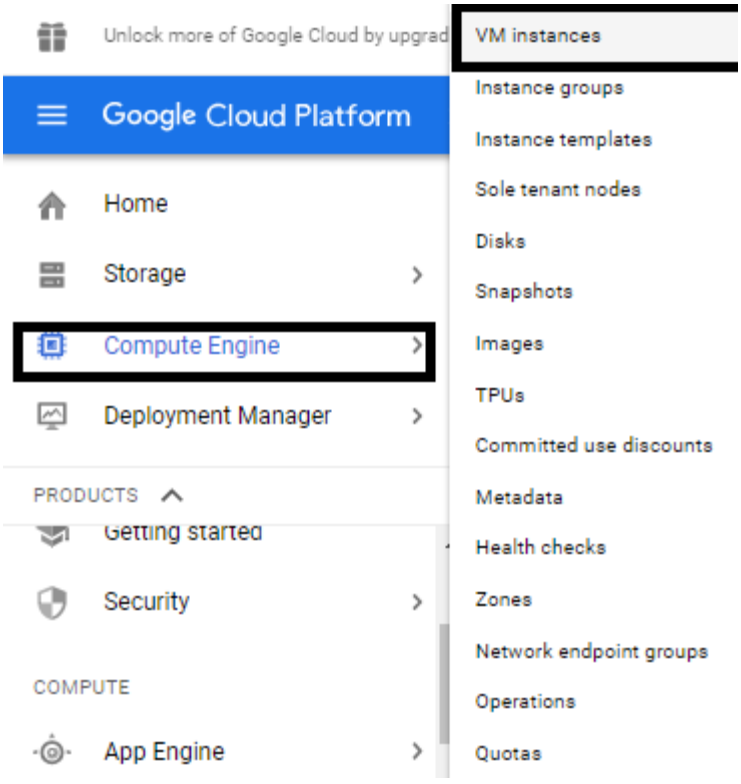
When a host project is enabled, **all of its existing VPC networks become Shared VPC networks, and any new network created in it will automatically be a Shared VPC network as well**. Thus, a single host project can have more than one Shared VPC network.

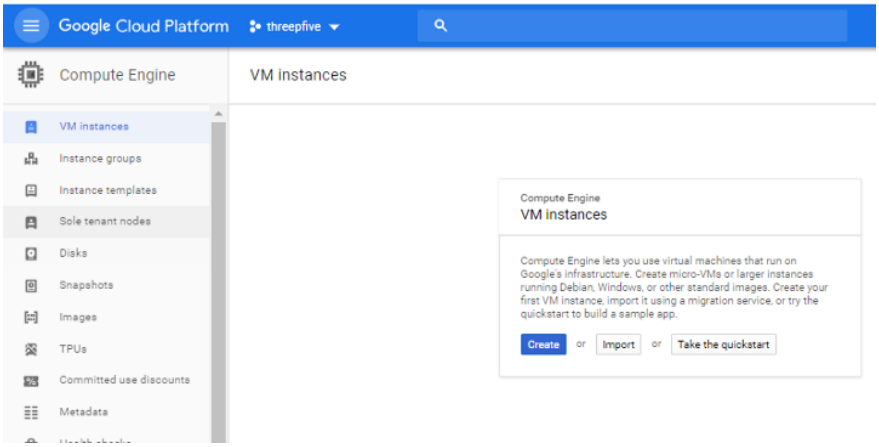
Host and service projects are connected by attachments **at the project level**. Subnets of the Shared VPC networks in the host project are accessible by Service Project Admins as described in the next section, administrators and IAM.

———————————————————————————————————-

Launching a Compute Engine instance with custom network configuration (e.g., Internal-only IP address, Google private access, Static external and private IP address, network tags)

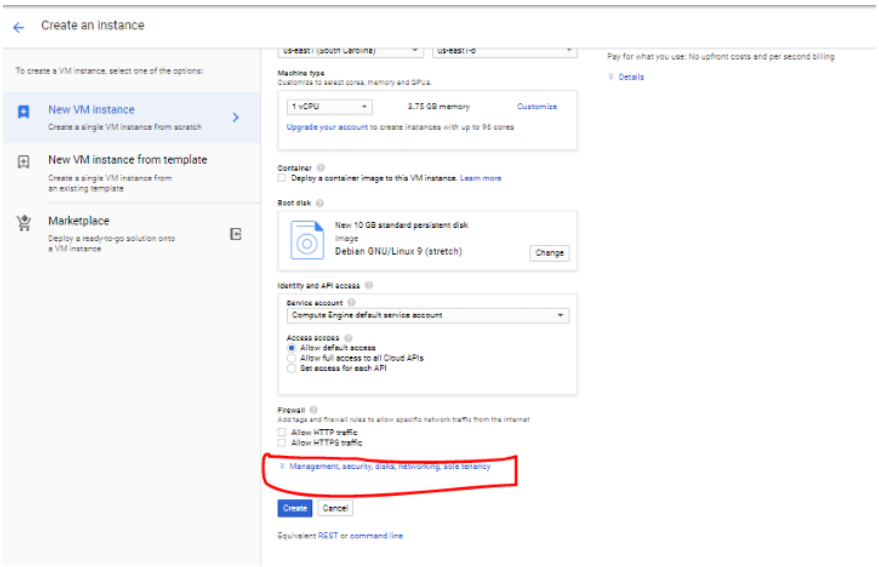
First, create a VM by going to main page>compute Engine>





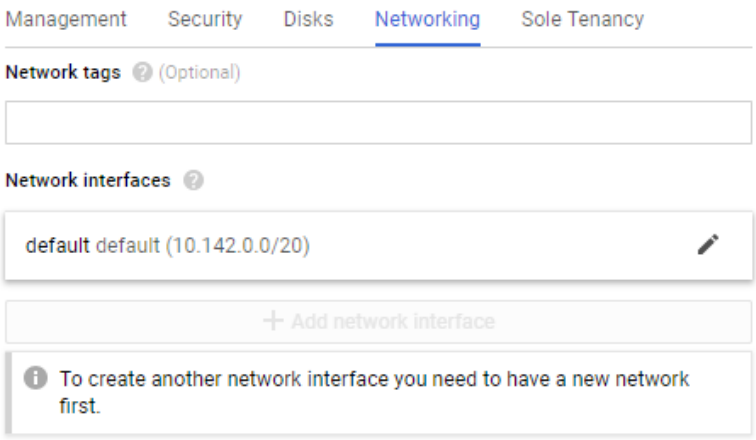
create VM

Now after giving OS, memory and HD space, at the bottom of the page, you will see networking. Click on the down arrow

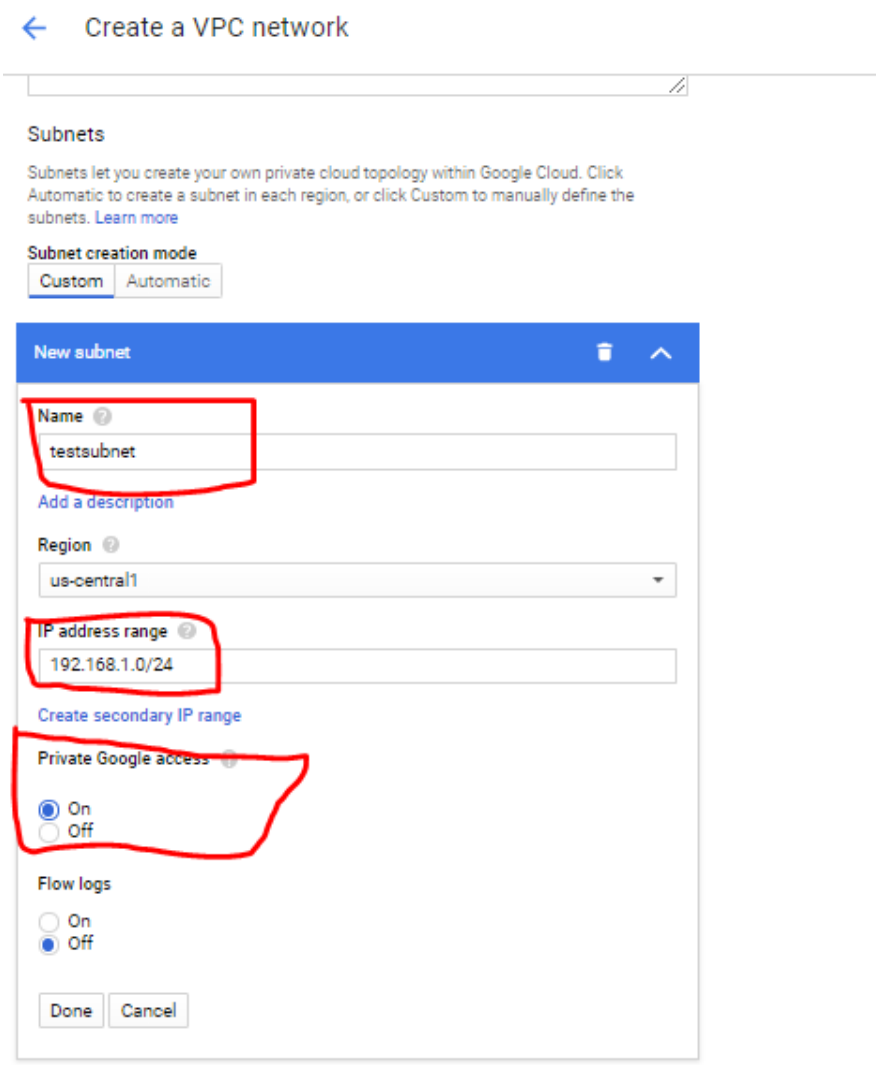


custom options

In Network tag, write something that defines this network and that you can remember easily. Then select the network interface and click on custom



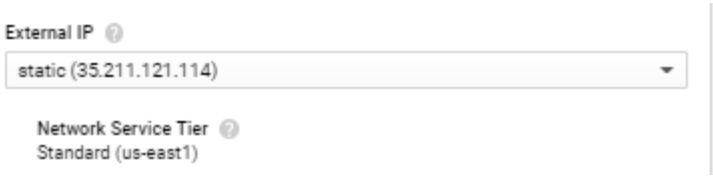
For making the custom network available you have to add a custom network in VPC networks that we did earlier in this post.



custom VPC with subnet info

Regions should be same for both VM and custom network to appear in the list.

You can also specify static external IP during setup



static IP

———————————————————————————————————————

Firewall Rules Overview

Google Cloud Platform (GCP) firewall rules let you allow or deny traffic to and from your virtual machine (VM) instances based on a configuration you specify. GCP firewall rules are applied at the virtual networking level, so they provide effective protection and traffic control regardless of the operating system your instances use.

Every VPC network functions as a distributed firewall. While firewall rules are defined at the network level, connections are allowed or denied on a per-instance basis. You can think of the GCP firewall rules as existing not only between your instances and other networks but between individual instances within the same network.

Firewall rules in GCP

GCP firewall rules are specific to a VPC network. Each rule either allows or denies traffic when its conditions are met. Its conditions allow you to specify the type of traffic, such as ports and protocols, and the source or destination of the traffic, including IP addresses, subnets, and instances. Refer to firewall rule components for descriptions of the components that define a firewall rule.

Every network has two permanent implied firewall rules which permit outgoing connections and block incoming connections. Refer to the default and implied firewall rules section for more information about their applicability and how they interact with rules you define. Additionally, the `default` network is pre-populated with some additional editable rules.

You create or modify GCP firewall rules through the Google Cloud Platform Console, `gcloud` command line tool, and REST API. When you create or modify a firewall rule, you can specify the instances to which it is intended to apply by using the target component of the rule.

Specifications

Firewall rules have the following characteristics:

- Firewall rules are defined at the VPC network level and are specific to the network in which they are defined. The rules themselves cannot be shared among networks.

- Firewall rules only support IPv4 traffic. When specifying a source for an ingress rule or a destination for an egress rule by address, you can only use an IPv4 address or IPv4 block in CIDR notation.
- The action taken by a firewall rule is either `allow` or `deny`. The rule cannot simply log as an action. Refer to the action on match component of a firewall rule for more information.
- Each firewall rule is defined to apply to either incoming (`ingress`) or outgoing (`egress`) traffic, not both. Refer to the direction of the traffic component of a firewall rule for more information.
- GCP firewall rules are stateful. 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 as long as the connection is active. In other words, firewall rules allow bidirectional communication once a session is established. The connection is considered active if at least one packet is sent every 10 minutes. *Firewall rules cannot allow traffic in one direction while denying the associated return traffic.*
- GCP firewall rules do not reassemble fragmented TCP packets. Consequently, *a firewall rule applies to the TCP protocol can only apply to the first fragment* because it contains the TCP header. Firewall rules applicable to the TCP protocol do not apply to the subsequent TCP fragments.
- The maximum number of tracked connections in the firewall rule table depends on the number of stateful connections supported by the machine type of the instance:

Direction of traffic

The direction of a firewall rule can be either `ingress` or `egress`. The direction is always defined from the perspective of the target.

- The `ingress` direction describes traffic sent from a source to a target. Ingress rules apply to packets for new sessions where the destination of the packet is the target.
- The `egress` direction describes traffic sent from a target to a destination. Egress rules apply to packets for new sessions where

the source of the packet is the target.

- If you don't specify a direction, GCP uses `ingress`.

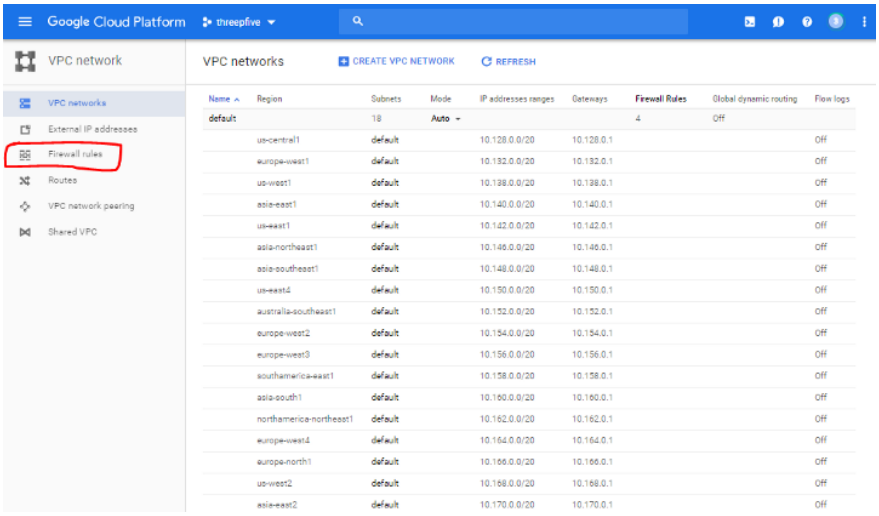
Consider an example connection between two VMs in the same network. Traffic from VM1 to VM2 can be controlled using either of these firewall rules:

- An `ingress` rule with a target of VM2 and a source of VM1.
- An `egress` rule with a target of VM1 and a destination of VM2.

—— -

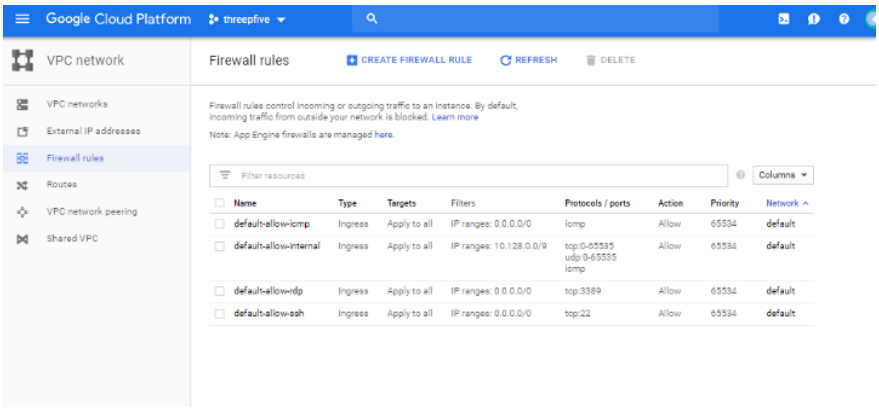
Creating ingress and egress firewall rules for a VPC (e.g., IP subnets, Tags, Service accounts)

From VPC Network from last post, goto Firewall rules.



Firewall rules

By default following rules are allowed in VPC while creating project.



Default rules.

———————————————————————————————————————

—

Cloud VPN Overview

Introduction

Cloud VPN securely **connects your on-premises network to your Google Cloud Platform (GCP) Virtual Private Cloud (VPC) network** through an IPsec VPN connection. Traffic traveling between the two networks is encrypted by one VPN gateway, then decrypted by the other VPN gateway. This protects your data as it travels over the Internet.

Features

Cloud VPN includes the following features:

- **Provides an SLA of 99.9% service availability**.
- **Supports site-to-site VPN** as a simple topology or with redundancy.
- **Supports both dynamic routes** that use Cloud Router, and static routes, to manage traffic between your Compute Engine Virtual Machine (VM) instances and your existing infrastructure.
- **Supports both IKEv1 and IKEv2** using a shared secret (IKE pre-shared key). Supports these IKE ciphers.

- **Uses ESP in Tunnel mode with authentication.** Cloud VPN does not support AH or ESP in Transport mode. Note that Cloud VPN does not perform policy-related filtering on incoming authentication packets. Outgoing packets are filtered based on the IP range configured on the Cloud VPN gateway.

VPN Topology

This diagram shows a simple VPN connection between your Cloud VPN gateway and your on-premises VPN gateway.

With Cloud VPN, your on-premises hosts communicate through one or more IPsec VPN tunnels to Compute Engine Virtual Machine (VM) instances in your project's VPC networks.

Choosing VPN for hybrid networking

See How to choose an Interconnect type to determine whether to use Cloud VPN, Cloud Interconnect—Dedicated or Cloud Interconnect—Partner as your hybrid networking connection to GCP. This page also covers what type of VPN scenarios Cloud VPN supports.

Terminology

The following terms are used throughout the VPN documentation:

Project IDThe ID of your GCP project. This is not the project name, which is the user-created friendly name of your project. To find the ID, see the **Project ID** column in the GCP Console. For more information, see Identifying Projects.Internet Key Exchange (IKE)IKE is the protocol used for authentication and to negotiate a session key for encrypting traffic. **Note:** Cloud VPN always initiates IKE. If two Cloud VPN gateways are involved, either can act as the IKE initiator. Cloud VPN gateway virtual VPN gateway running in GCP managed by Google, using a configuration you specify in your project. Each Cloud VPN gateway is a regional resource using a regional external IP address. A Cloud VPN gateway can connect to an on-premises VPN gateway or another Cloud VPN gateway.On-premises VPN gatewayThe VPN gateway not in GCP, connected to a Cloud 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. Cloud VPN instructions

are written from the point of view of your VPC network, so the “on-premises gateway” is the gateway connecting *to* Cloud VPN.VPN tunnel connects two VPN gateways and serves as a virtual medium through which encrypted traffic is passed. Two VPN tunnels must be established to create a connection between two VPN gateways: Each tunnel defines the connection from the perspective of its gateway, and traffic can only pass once the pair of tunnels is established.

Tunnel routing options

Cloud VPN offers three different routing methods for VPN tunnels:

Dynamic (BGP) routingA Cloud Router can manage routes for a Cloud VPN tunnel using Border Gateway Protocol (BGP) if the corresponding or on-premises VPN gateway supports it. This routing method allows for routes to be updated and exchanged without changing the tunnel configuration. Routes to GCP subnets are exported to the on-premises VPN gateway, and routes to on-premises subnets learned from the on-premises VPN gateway are applied to your VPC network, both according to the dynamic routing option of the network. Dynamic routing is recommended because it does not require that tunnels be re-created when routes change. Policy-based routingWith this routing option, you specify *remote network IP ranges* and *local subnets* when creating the Cloud VPN tunnel. From the perspective of Cloud VPN, the remote network IP ranges are the “right side,” and the local subnets are the “left side” of the VPN tunnel. GCP automatically creates static routes for each of the remote network ranges when the tunnel is created. When creating the corresponding tunnel at the on-premises VPN gateway, the right and left side ranges are reversed. Route based VPN with this routing option, you only specify the *remote network IP ranges* (right side). All incoming traffic is accepted through the tunnel, subject to routes you create manually. **Note:** Some literature refers to the left and right side subnet ranges as *encryption domains*.

For more details about network types and routing options, see the Choosing a VPC Network Type and Routing Options page.

Specifications

Cloud VPN has the following specifications:

- Cloud VPN can be used with VPC networks and legacy networks. For VPC, a custom mode is recommended so you have full control over the ranges of IP addresses used by the subnets in the network.
- If IP address ranges for on-premise subnets overlap with IP addresses used by subnets in your VPC network, refer to Order of routes to determine how routing conflicts are resolved.
- Each Cloud VPN gateway must be connected to another Cloud VPN gateway or an on-premisesVPN gateway.
- The on-premises VPN gateway must have a static external IP address. You'll need to know its IP address in order to configure Cloud VPN.
- If your on-premises VPN gateway is behind a firewall, you must configure the firewall to pass ESP (IPSec) protocol and IKE (UDP 500 and UDP 4500) traffic to it. If the firewall provides Network Address Translation (NAT), refer to UDP encapsulation and NAT-T.
- Cloud VPN only supports a pre-shared key (shared secret) for authentication. You must specify a shared secret when you create the Cloud VPN tunnel. This same secret must be specified when creating the tunnel at the on-premises gateway. Refer to these guidelines for creating a strong shared secret.
- Cloud VPN uses a Maximum Transmission Unit (MTU) of 1460 bytes. On-premises VPN gateways must be configured to use an MTU of no greater than 1460 bytes.
- To account for ESP overhead, you may need to set the MTU values for **systems** sending traffic through the tunnel to lower values. Refer to MTU Considerations for a detailed discussion and recommendations.
- Cloud VPN requires that the on-premises VPN gateway is configured to support fragmentation. Packets must be fragmented *before* being encapsulated.
- Cloud VPN uses replay detection with a window of 4096 packets. You cannot turn this off.
- Refer to Supported IKE Ciphers for ciphers and configuration parameters supported by Cloud VPN.

Maintenance and availability

Cloud VPN undergoes periodic maintenance. During maintenance, Cloud VPN tunnels are taken offline, resulting in brief drops in network traffic. When maintenance completes, Cloud VPN tunnels are automatically re-established.

Maintenance for Cloud VPN is a normal, operational task that may happen at any time without prior notice. Maintenance periods are designed to be short enough so that the Cloud VPN SLA is not impacted.

You can design highly available VPN configurations by using multiple tunnels. Some strategies for doing this are discussed on the Redundant and High-throughput VPNs page.

UDP encapsulation and NAT-T

Cloud VPN **only** supports one-to-one NAT via UDP encapsulation for NAT-Traversal (NAT-T). One-to-many NAT and port-based address translation are **not** supported. In other words, Cloud VPN **can not connect** to multiple on-premises or peer VPN gateways that share a single public IP address.

When using one-to-one NAT, an on-premises VPN gateway **must** be configured to identify itself using a public IP address, not its internal (private) address. When you configure a Cloud VPN tunnel to connect to an on-premises VPN gateway, you specify an external IP address. Cloud VPN expects an on-premises VPN gateway to use its external IP address for its identity.

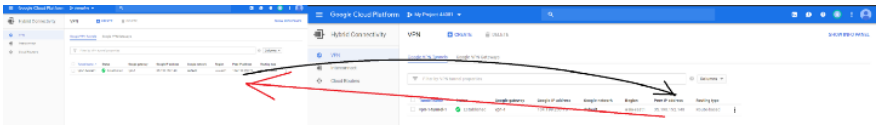
For more details about VPN gateways behind one-to-one NAT, refer to the troubleshooting page.

Best practices

Use these best practices to build your Cloud VPN in the most effective way.

———————————————————————————————————

Creating a VPN between a Google VPC and an external network using Cloud VPN

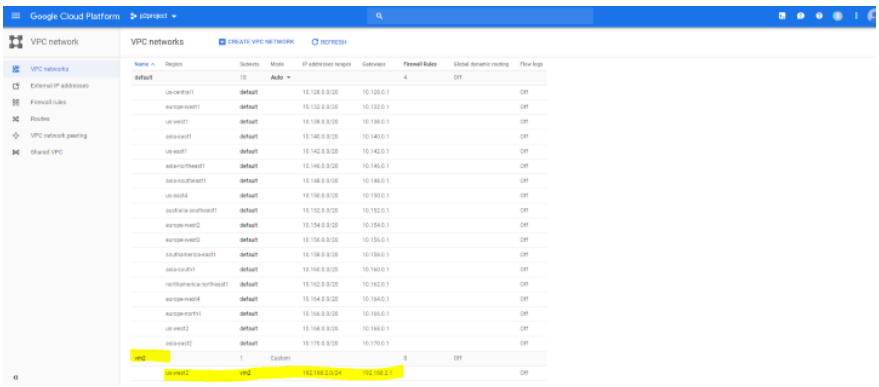


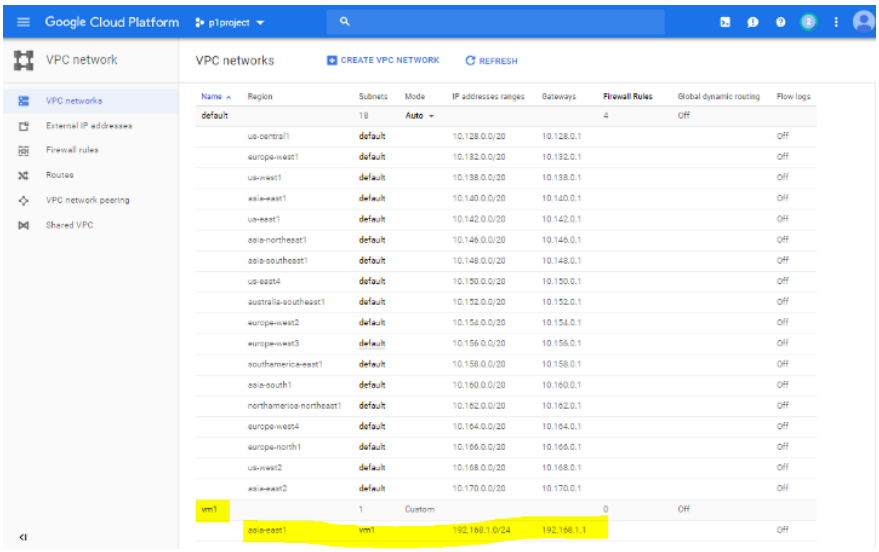
VPN connection

Creating a VPN connection between two networks is a basic purpose of a VPN connection. We will try to connect two networks by using cloud VPN.

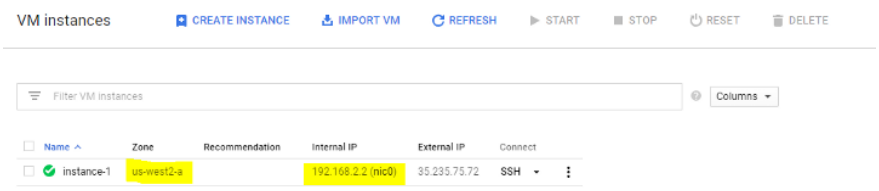
1. Default VPC network with custom subnet having one VM
2. Custom VPC with custom subnet having one VM

First of all, create a subnet within VPC

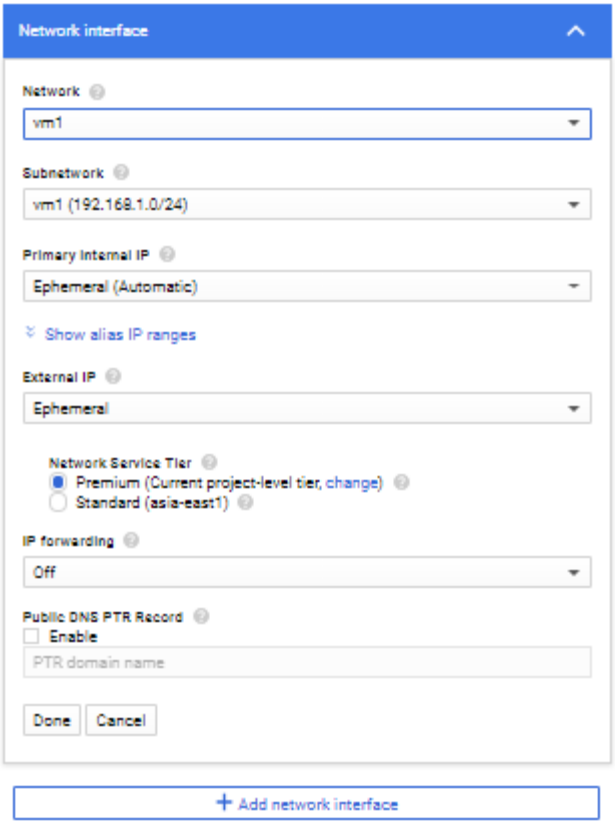




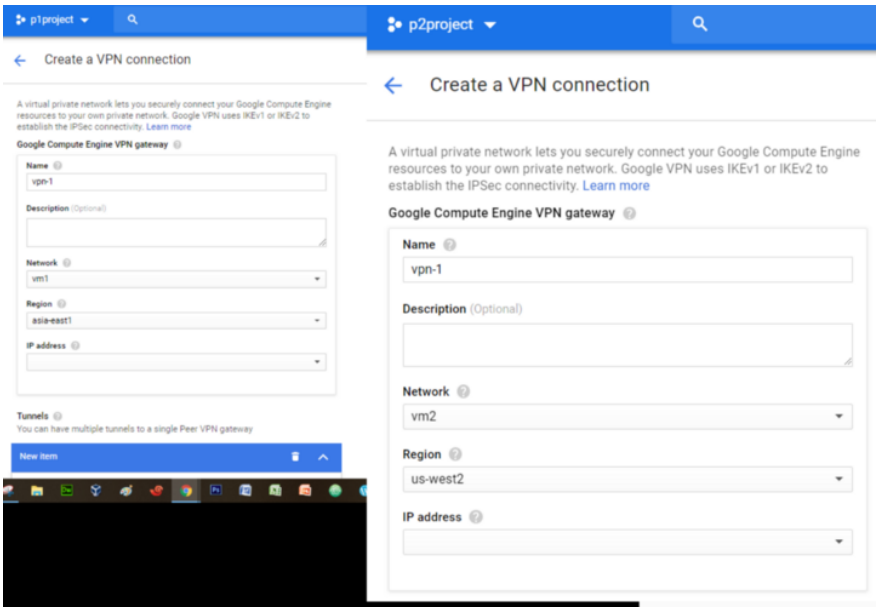
Now create a VM inside the VPC

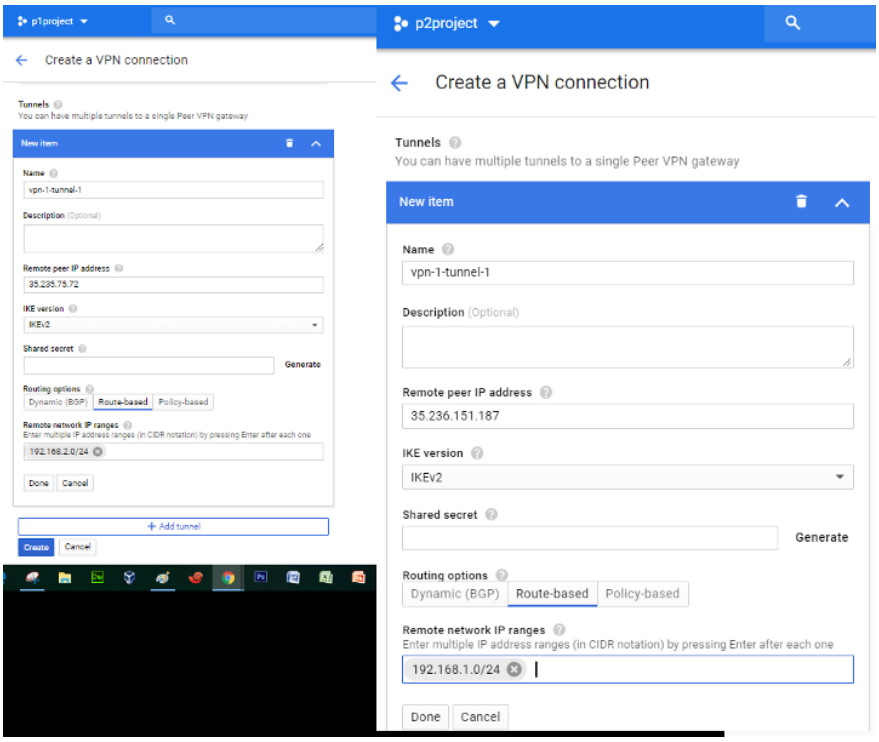


While creating VM select network and IP during setup.

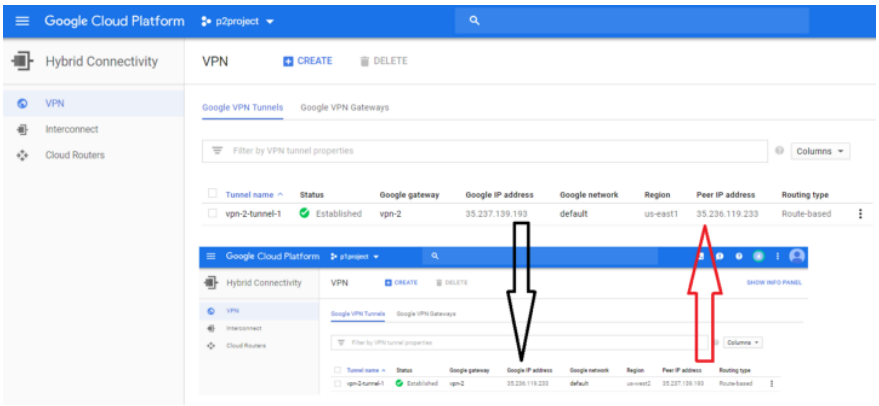


Use the VM and VPC IP detail to connect using cloud VPN.





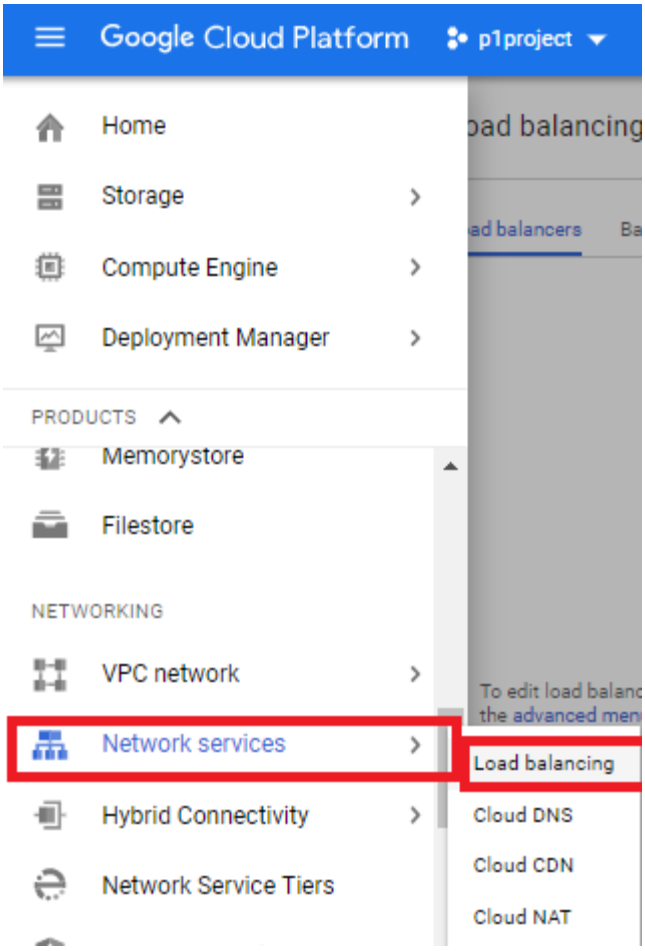
After the VPN connected successfully, a green check mark will be shown and we can ping between two VM's.



———————————————————————————————————————

Creating a load balancer to distribute application network traffic to an application (e.g., Global HTTP(S) load balancer, Global SSL Proxy load balancer, Global TCP Proxy load balancer, Regional Network load balancer, Regional Internal load balancer)

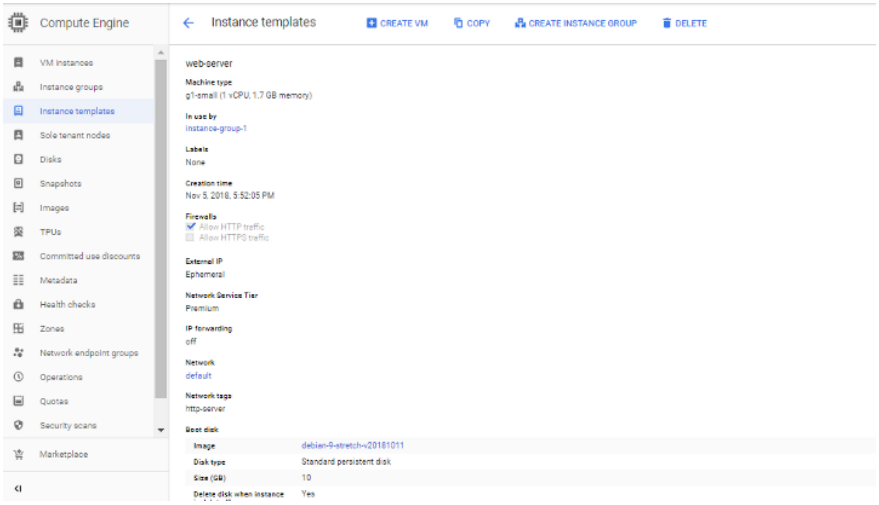
To go to the load balancer setup, click on Network services and then to load balancing



load balancing

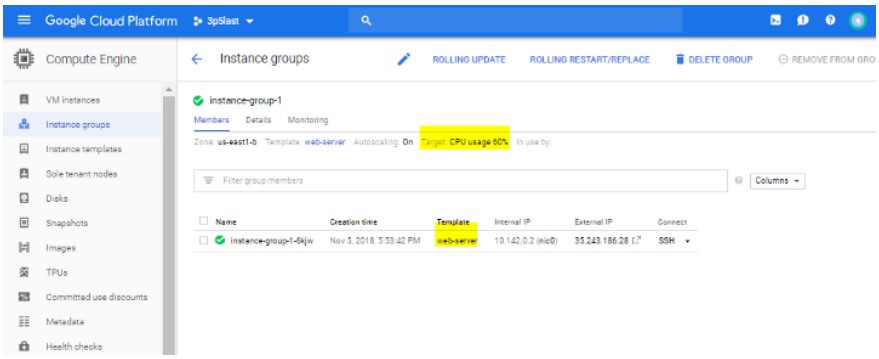
This will lead to another page with various options.

Before going there, I will create an instance template and instance group.



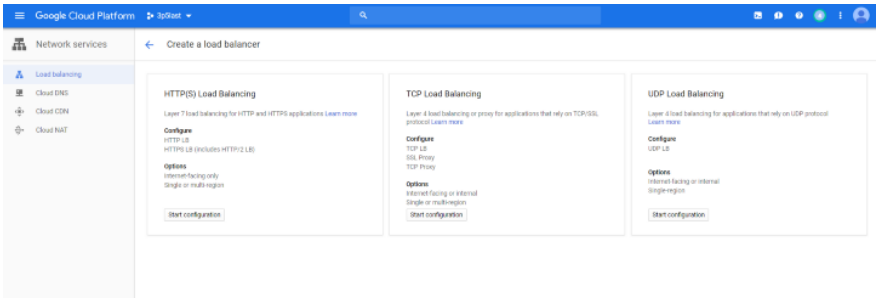
instance template

Based on instance template I created instance group



instance group

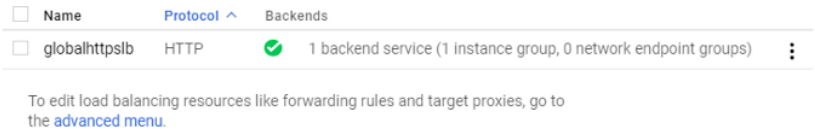
Now when you go to load balancing page, you will see these options.



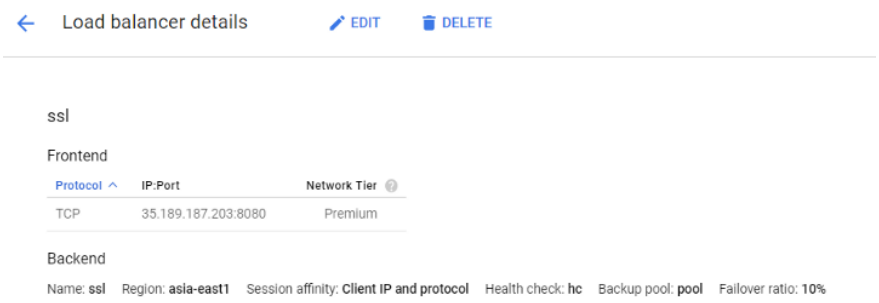
load balancing options

Global HTTPS load balancer

After a while, it will show enabled.

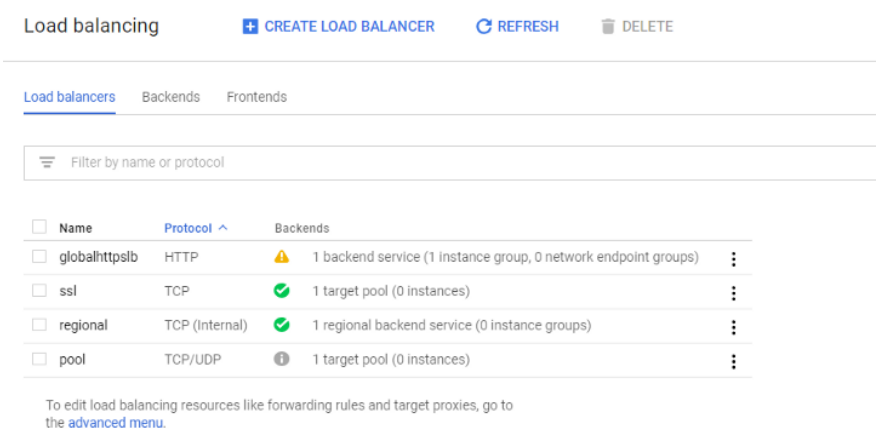


Global SSL Proxy Load Balancer



SSL LB

In this way, various load balancers can be set up easily from the console.


Load balancing
CREATE LOAD BALANCER
REFRESH
DELETE
Load balancers
Backends
Frontends
Filter by name or protocol
Name
Protocol
Backends
globalhttpslb
HTTP
1 backend service (1 instance group, 0 network endpoint groups)
ssl
TCP
1 target pool (0 instances)
regional
TCP (Internal)
1 regional backend service (0 instance groups)
pool
TCP/UDP
1 target pool (0 instances)
To edit load balancing resources like forwarding rules and target proxies, go to the advanced menu.

various LBs