VPC Networking Console

21 June 2021 21:51

Product URL: https://cloud.google.com/vpc

Pending
Best Practices

5 Rule for better network performance

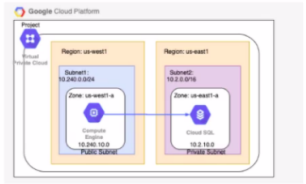
https://cloud.google.com/blog/products/gcp/5-steps-to-better-gcp-network-performance?hl=ml

Configure private google access hybrid

Configure a VPN tunnel between the on-premises data centre and the GCP VPC. Create a custom route in the VPC for Google Restricted APIs IP range (199.36.153.4/30) and propagate the route over VPN. Resolve *.googleapis.com as a CNAME record to restricted.googleapis.com in your on-premises DNS server.

Create VPC

<table>
<tr>
<td>Subnet Introduction

Best Practices (Important): https://cloud.google.com/architecture/best-practices-vpc-design#shared-service</td>
<td>
VPC Subnets
• (Solution) Create different subnets for public and private resources
▪ Resources in a public subnet CAN be accessed from internet
▪ Resources in a private subnet CANNOT be accessed from internet
▪ BUT resources in public subnet can talk to resources in private subnet
• Each Subnet is created in a region
• Example : VPC - demo-vpc => Subnets - region us-central1, europe-west1 or us-west1 or ..

</td>
</tr>
<tr>
<td>Advance VPC Concept

1. VPC network uses linux VIRTIO module to model instance ethernet card and router. But higeht level of stack such as ARP lookup are handled using standard networking software.
2. MAC lookup table, IP lookup table, active connection table: These tables are hosted on the underlying VPC network and cannot be inspected or configured.
3. DNS server: Each instance's metadata server acts as a DNS server. You can add your own search domain or nameservers to the instance's /etc/resolv.conf by modifying the DHCP policy
4. Packet handling between the VPC network and the outside: Packets coming into or out of the VPC network are handled by network code that examines the packet against firewall rules,</td>
<td></td>
</tr>
</table>

against the external IP lookup table, and against the active connections table

External IP to internal IP

External IP -> Instance external IP (owened by VPC network) -> VPC network check active connection table and valid firewall rule -> if success replace the external with internal ip -> instance receive and return packet -> vpc network check active connection and replace the source ip with external ip -> allow the request through

Internal IP to external IP

Instance send packet to subnet gateway mac address, ARP request resolution might be required -> VPC network rewrite ip header with instance external ip -> if no external ip external connection not allowed -> If success VPC network create an active connection -> destinaltion get packets and respond -> VPC network check the active connection and replace external target ip with internal ip and allow the connection -> instance receive packet.

Creating VPC and subnet

Creating VPCs and Subnets

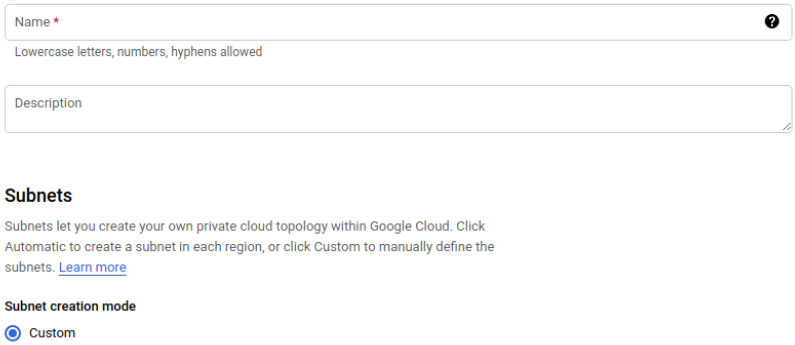
- By default, every project has a default VPC
- You can create YOUR own VPCs:
  - **OPTION 1**: Auto mode VPC network:
    - Subnets are automatically created in each region
    - Default VPC created automatically in the project uses auto mode!
  - **OPTION 2**: Custom mode VPC network:
    - No subnets are automatically created
    - You have complete control over subnets and their IP ranges
    - Recommended for Production
- Options when you create a subnet:
  - Enable **Private Google Access** - Allows VM's to connect to Google API's using private IP's
  - Enable **FlowLogs** - To troubleshoot any VPC related network issues

Dynamic Routing: Global dynamic routing allows all subnetworks regardless of region to be advertised to your on-premise router and region when using cloud router. With global routing you just need a single VPN with cloud router to dynamically learn routes to and from all GCP regions on a network.

In networking, maximum transmission unit (MTU) is a measurement representing the largest data packet that a network-connected device will accept

gcloud compute networks create staging-host-vpc --project=ic-int-sandbox-saraswata --description=Staging\ Host\ VPC --subnet-mode=custom --mtu=1460 --bgp-routing-mode=regional

gcloud compute networks subnets create staging-host-subnet --project=ic-int-sandbox-saraswata --description=Host\ Subnet\ For\ Networking --range=10.0.0.0/15 --network=staging-host-vpc --region=australia-southeast1 --enable-private-ip-google-access

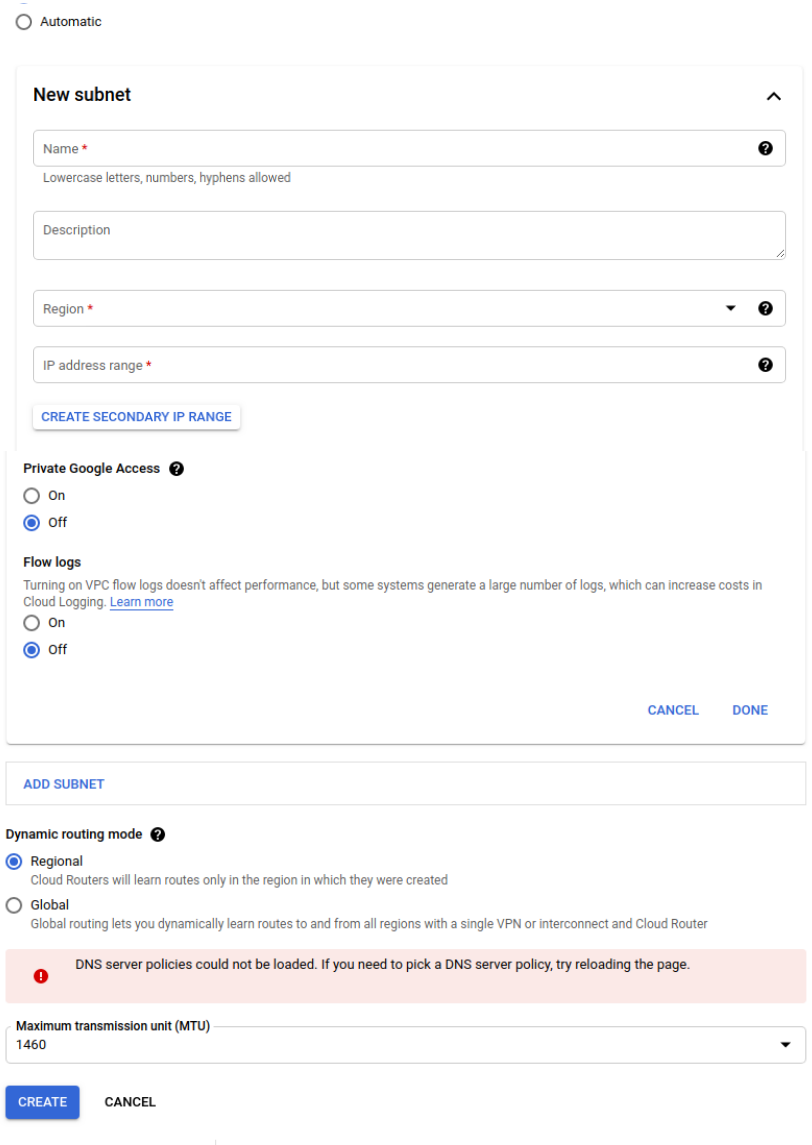


gcloud compute networks subnets create staging-service-subnet-web --project=ic-int-sandbox-saraswata --description=Service\ Web\ Subnet\ For\ Networking --range=10.128.0.0/15 --network=staging-host-vpc --region=australia-southeast1 --enable-private-ip-google-access

gcloud compute networks subnets create staging-service-subnet-db --project=ic-int-sandbox-saraswata --description=Service\ DB\ Subnet\ For\ Networking --range=10.192.0.0/15 --network=staging-host-vpc --region=australia-southeast1 --enable-private-ip-google-access

Web VPC

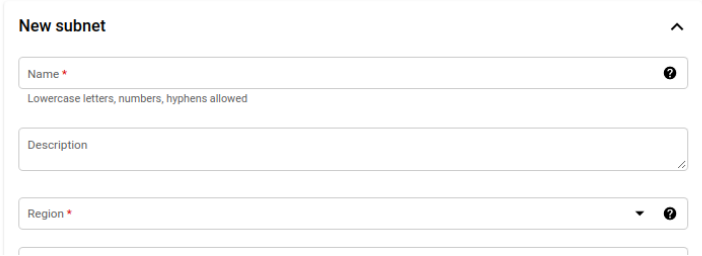
gcloud compute networks create staging-web-vpc --project=ic-int-sandbox-saraswata --description=Staging\ Web\ VPC --subnet-mode=custom --mtu=1460 --bgp-routing-mode=regional



Add Subnet

Private google access: Google Cloud provides several private access options that let virtual machine (VM) instances reach supported APIs and services without requiring an external IP address. Choose an option that supports the APIs and services that you need to access.

Flow Log: VPC **Flow Logs** works with VPC networks, not legacy networks. You enable or disable VPC **Flow Logs** per **subnet**. If enabled for a **subnet**, VPC **Flow Logs** collects data from all VM instances in that **subnet**. VPC **Flow Logs** samples k
VM's TCP, UDP, ICMP, and GRE **flows**.



Advantage of secondary IP ranges

There are advantages to allocating alias IP ranges from a secondary CIDR range. By allocating from a range separate from the range used for primary IP addresses, you can separate infrastructure (VMs) from services (containers). When you configure separate address spaces for infrastructure and services, you can set up firewall controls for VM alias IP addresses separately from the firewall controls for a VM's primary IP addresses. For example, you can allow certain traffic for container pods and deny similar traffic for the VM's primary IP address.

Alias IP Ranges

If you have only one service running on a VM, you can reference it using the interface's primary IP address. 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.

Alias IP is allocated from secondary IP range
Using IP aliasing, you can configure multiple internal IP addresses, representing containers or applications hosted in a VM, without having to define a separate network interface.

Alias IP addresses can be announced by Cloud Router to an on-premises network connected via VPN or Interconnect.

We you can set up firewall controls for VM alias IP addresses separately from the firewall controls for a VM's primary IP addresses. For example in a VPN connection we can allow only secondary IP of a container but deny primary ip connection from on-promises.

Auto mode VPC does not have alias IP ranges created but we can create it.

Only primary ip is associated with a host name in vm network interface (only accessible inside network), we can create own dns to associate ip with a host name

Both primary and secondary IP ranges of a subnet are reachable by VM instances in a peered network.

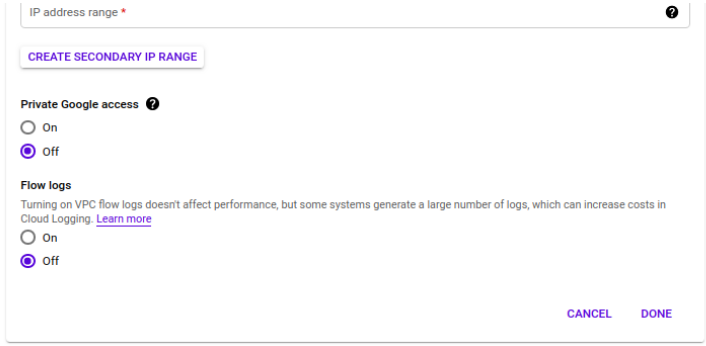
Create two firewall rules.

- One rule that denies traffic traveling across the VPN from on-premises from reaching the subnet primary CIDR range.
- One rule that allows traffic traveling across the VPN from on-premises to reach the subnet secondary CIDR range.

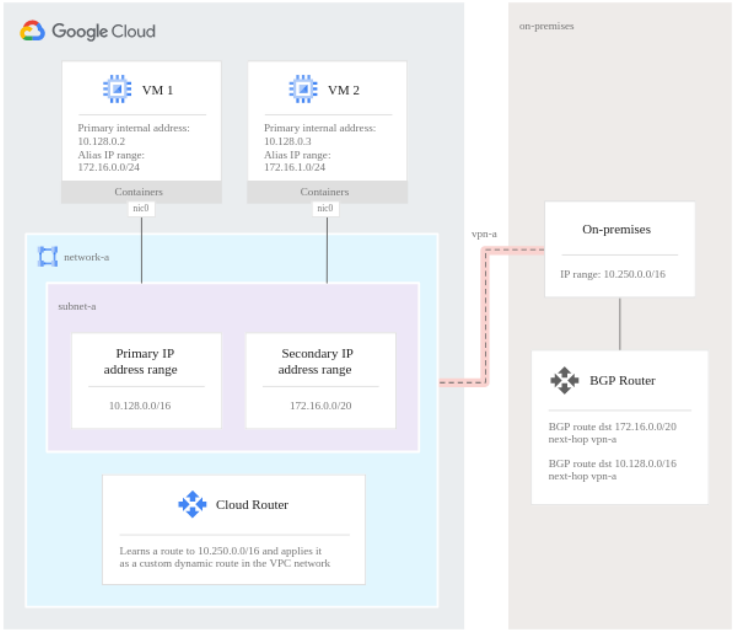
Multiple Network Interface

Every instance in a VPC network has a default network interface. You can create additional network interfaces attached to your VMs, but each interface must attach to a different VPC network. Multiple network interfaces enable you to create configurations in which an instance connects directly to several VPC networks. Each of the interfaces must have an internal IP address, and each interface can also have an external IP address. Each instance can have up to 8 interfaces, depending on the instance's type. For more information, see Maximum number of interfaces.

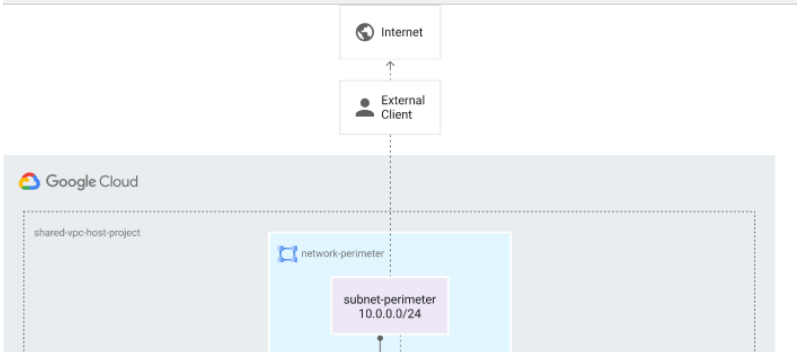
Use case
Use multiple network interfaces when an individual instance needs access to more than one VPC network, but you don't want to connect both networks directly.



Alias IP



Multiple Network Interface



When we want to have one network exposed to public and one network private the we can route and filter request using firewall using multiple interface in vm instance in public network. We can use in shared network also . (Perimeter Isolation)

Multiple interface can work as NAT, Load Balancer and Proxy Server

DHCP and ARP behavior for each interface is same every interface get route for subnet.

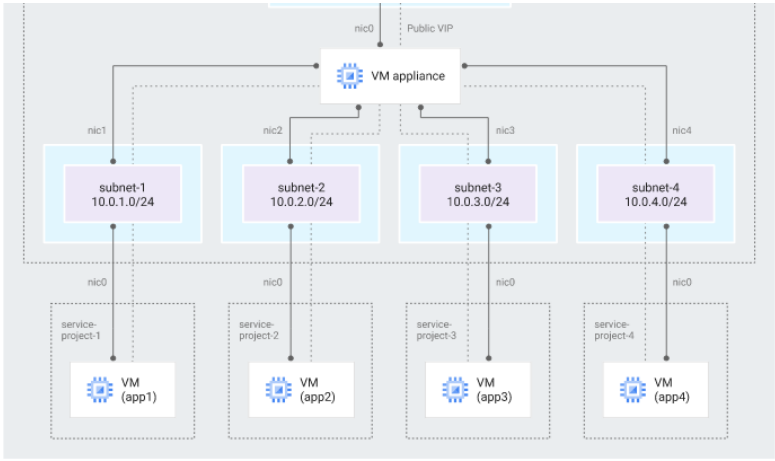
Eth0 is the default route

Each interface of instance is in a specific VPC network and firewall rule of that VPC applies to that interface

Network Service Tier

Premium Tier	Standard Tier
• Highest performance: Traffic between the internet and VM instances in your VPC network is routed by keeping it within Google's network as much as possible. • For services that need global availability. • Unique to Google Cloud. • Premium Tier is the default unless you make configuration changes.	• Cost optimized: Traffic between the internet and VM instances in your VPC network is routed over the internet in general. • For services hosted entirely within a region. • Performance is comparable to other cloud providers.

Traffic	Premium Tier	Standard Tier
Ingress to Google Cloud	Traffic from your users enters Google's network at a location nearest to them.	Traffic from your users enters Google's network through peering, ISP, or transit networks in the region where you have deployed your Google Cloud resources.
Egress from Google Cloud	cold potato routing Egress traffic is sent through Google's network backbone, leaving at a global edge point of presence (PoP) closest to your users.	hot potato routing Egress traffic is sent to the internet through a peering or transit network local to the Google Cloud region from which it originates.



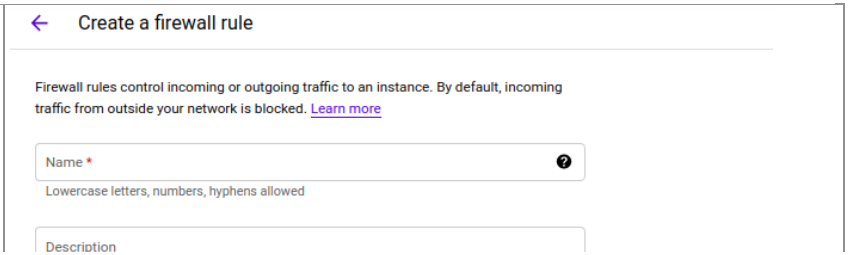
This creates an instance with five network interfaces:

- nic0 is attached to subnet-perimeter, which is part of network-perimeter, with a static address 'reserved-address'
- nic1 is attached to subnet-1, which is part of network-1, with no external IP
- nic2 is attached to subnet-2, which is part of network-2, with no external IP
- nic3 is attached to subnet-3, which is part of network-3, with no external IP
- nic4 is attached to subnet-4, which is part of network-4, with no external IP

Create Firewall

Logs:

1. Helps in analyzing traffic.
2. Let us know about how many connection were affected by a rule.
3. Only supported in VPC network not in legacy.
4. Log TCP and UDP traffic for other protocol check packet mirrioring
5. Logging cannot be enabled for implied ingress and egress rule.
6. The no of connections that can be logged depend upon vm machine type.
7. Changes to firewall rule can be viewed in VPC audit log.



Priority: You can specify the order that a rule will be applied within a network. Rules with lower numbers get prioritized first. Default is 1000.

Targets:
Firewall rule applies only to these instances within the virtual network

1. All instances in the network.
2. Specified target tags.
3. Specified service account

Source IP Ranges

Traffic is only allowed from sources within these IP address ranges. Use CIDR notation when entering ranges.

Second Source Filter

Set additional filters to apply your rule to specific sources of traffic. The filter logic is "Source filter" OR "Second source filter"

1. Source tags
2. Service account

Hirarchical Firewall policy

Hierarchical firewall policies let you create and enforce a consistent firewall policy across your organization. You can assign hierarchical firewall policies to the organization as a whole or to individual folders. These policies contain rules that can explicitly deny or allow connections

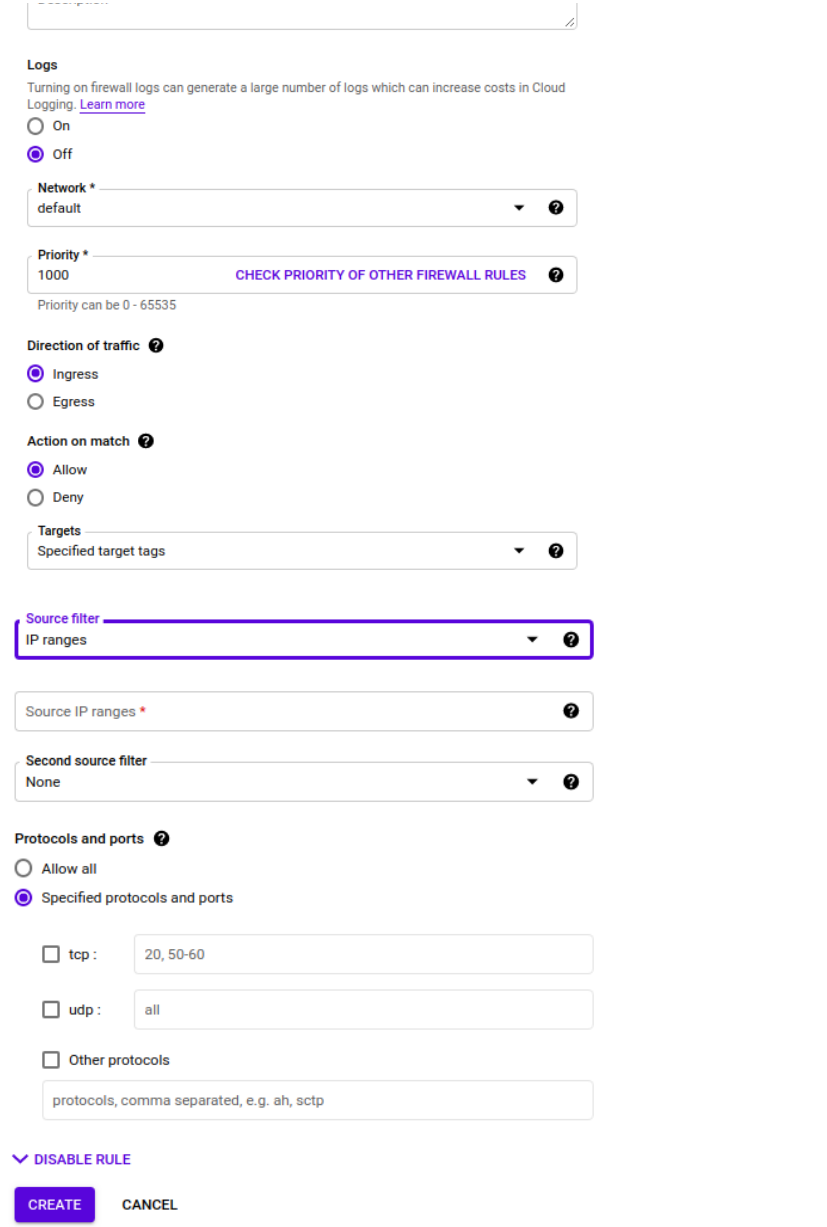
Questions

Q. You need to allow traffic from specific virtual machines in 'subnet-a' network access to machines in 'subnet-b' without giving the entirety of subnet-a access. How can you accomplish this?

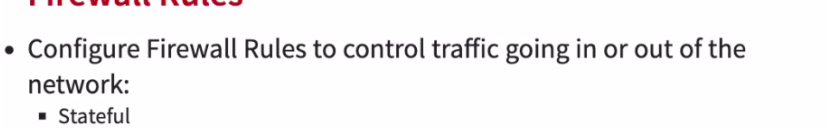
A. Every VPC by default has a firewall rule that allows network traffic between subnet. In summary, traffic between subnets inside a VPC is authorized, this is a default VPC behavior. Therefore, to limit the access from specific VMs in subnet-a, you have first to create a firewall rule that denies all the access from subnet-a, then create a second rule with a higher priority, to allow the traffic from VMs with a specific tag.

Firewall Insight Feature

[Get Started with Firewall Insights in Network Intelligence Center]



Firewall Concepts



- Each firewall rule has priority (0-65535) assigned to it
- 0 has highest priority. 65535 has least priority
- Default implied rule with lowest priority (65535)
  - Allow all egress
  - Deny all ingress
  - Default rules can't be deleted
  - You can override default rules by defining new rules with priority 0-65534
- Default VPC has 4 additional rules with priority 65534
  - Allow incoming traffic from VM instances in same network (**default-allow-internal**)
  - Allow Incoming TCP traffic on port 22 (SSH) **default-allow-ssh**
  - Allow Incoming TCP traffic on port 3389 (RDP) **default-allow-rdp**
  - Allow Incoming ICMP from any source on the network **default-allow-icmp**

Firewall Rules - Ingress and Egress Rules

- **Ingress Rules**: Incoming traffic from outside to GCP targets
  - **Target (defines the destination)**: All instances or instances with TAG/SA
  - **Source (defines where the traffic is coming from)**: CIDR · instances with TAG/SA
- **Egress Rules**: Outgoing traffic to destination from GCP targets
  - **Target (defines the source)**: All instances or instances with TAG/SA
  - **Destination**: CIDR Block
- **Along with each rule**, you can also define:
  - **Priority** - Lower the number, higher the priority
  - **Action on match** - Allow or Deny traffic
  - **Protocol** - ex. TCP or UDP or ICMP
  - **Port** - Which port?
  - **Enforcement status** - Enable or Disable the rule

Firewall Best Practices

Firewall Rules - Best Practices

- Use **network tags** and control allowed traffic into a VM using firewall rules
- Ensure that firewall rule allow the right kind of traffic:
  - Only allow traffic from load balancing into VM instances
  - Remove `0.0.0.0/0` from Source IP ranges
    - Add `130.211.0.0/22` and `35.191.0.0/16`
    - Allows health checks from load balancing to VM instances
- (REMEMBER) All egress from an VM instance is allowed by default:
  - To allow Specific EGRESS ONLY
    - 1: Create an egress rule with low priority to deny all traffic
    - 2: Create egress rule with high priority to allow traffic on specific port

Some more points

1. Some firewall rule are always allowed like , instance metadata, DHCP
2. Firewall rule is for a VPC network it cannot be shared.
3. Default firewall rules in default network can be deleted or modified as necessary.
4. GKE create firewall rule automatically for cluster, service and ingress

5. IPv6 is allowed in firewall rule if IPv6 is enabled for the network
6. When possible use service account instead of tag as only service account user has service account access. And a vm can have only one service account.
7. Firewall rule action, direction cannot be changed once created
8. Service account should be a proper email address, we cannot combine tag and service account in same rule.

Create Routes

Gcloud: gcloud beta compute routes create NAME --project=shinehub-vpp-oauthserver --network=default --priority=1000 --destination-range=DESTINATION_RANGE --next-hop-gateway=default-internet-gateway

Destination IP ranges: The destination IP range that this route applies to. If the destination IP of a packet falls in this range, it matches this route.

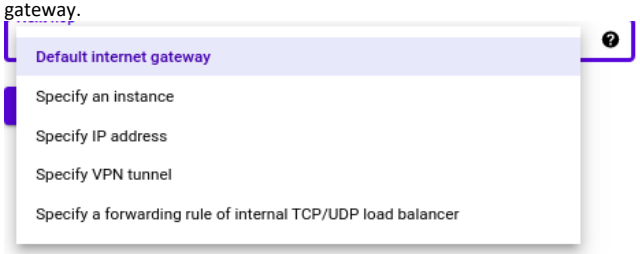
Priority:
Priority is used to break ties when there is more than one matching route of maximum length

Instance tags

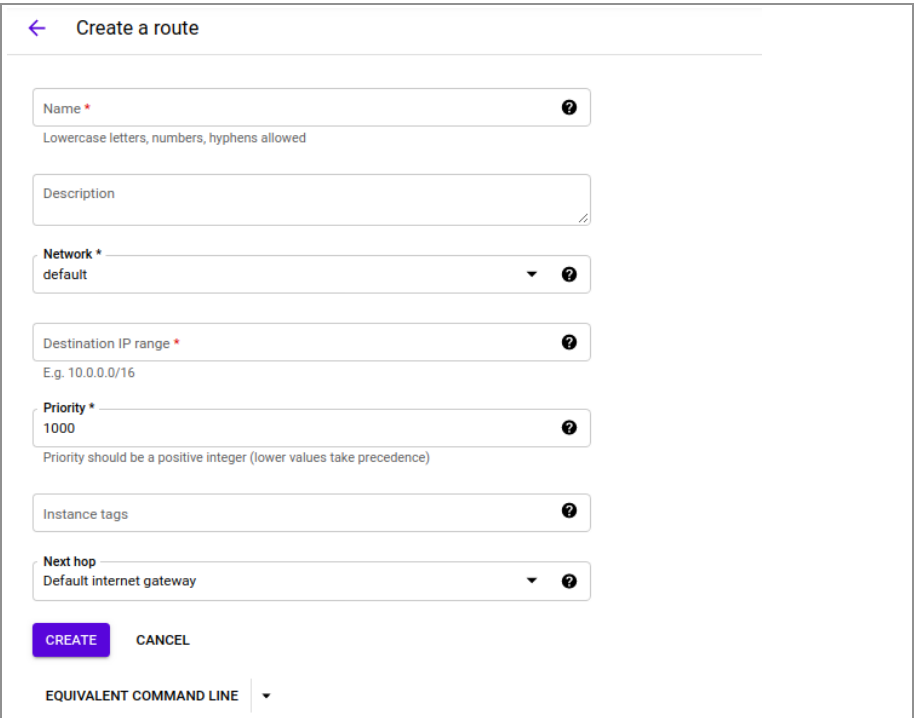
The route applies to all instances with any of these tags, or to all instances in the network if no tags are specified

Next Hop

The next hop handles the matching packets for this route. It can be an instance, an IP address or the default internet gateway.



Name	Description	Destination IP range	Priority	Instance tags	Next hop	Network	
	default		0.0.0.0/0	1000	None	Default internet gateway	vpc-alpha
	default-route-0329d33aac61bb64	Default local route to the subnetwork 10.1.0.0/16.	10.1.0.0/16	0	None	Virtual network shinehub-cluster-network	shinehub-cluster-network



Some more points:

1. Route table for a VPC network is defined at VPC network level. VM instance has a controller which is kept informed about all applicable route from network routing table.
2. On creation of VPC network, network has a default rout which allow connection to internet. This route can be deleted to deny all internet connection. Or we can change the next hop of default route to a proxy VM.
3. For each subnet there is a subnet route with IP range(primary or secondary) of subnet

4. We can create static route manually.
5. Dynamic route is created by a cloud router for VPN (Classic, HA), interconnect.
6. Peer network export there subnet route for all primary and secondary ranges connected using VPC peering.
7. For peering we can import/export custom route (which include static and dynamic route)
8. In VPC peering some route like route with next hop internet gateway or route with network tag scope are not imported and exported.
9. Some routes are present outside of VPC network and cannot be deleted or overridden. Like route from helth check system to VM or from GFE to backend VM.
10. We can set next hop as internet gateway, vm instance, internal tcp/udp load balancer, vpn tunnel and ip address.

Create VPC Peering Connection

Your VPC Network:
Select a VPC network in your current project. The subnets must not overlap between the two peering VPC networks.

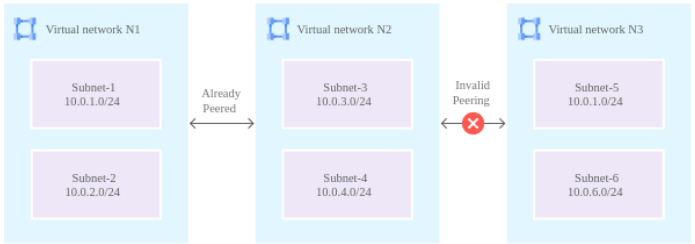
Exchange custom route

By default, a peering connection exchanges subnet routes only. You can also import or export custom routes. Routes that use instance tags or internet gateway routes won't be imported or exported.

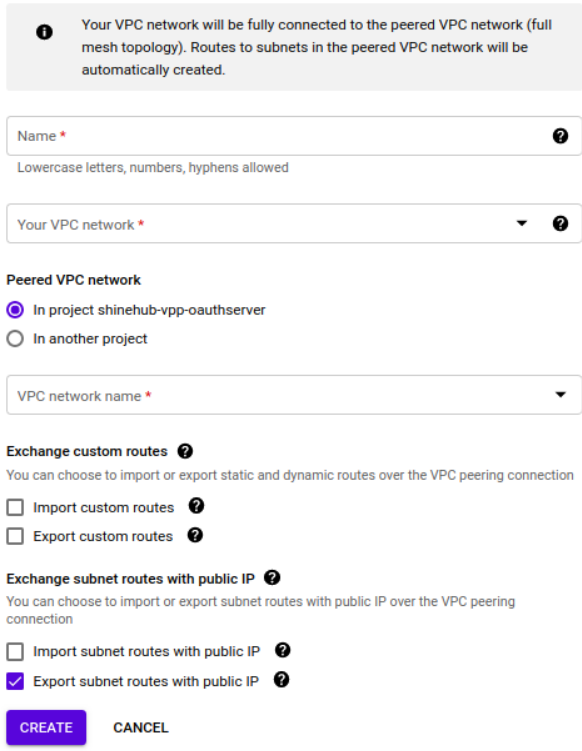
Export subnet route with public ip

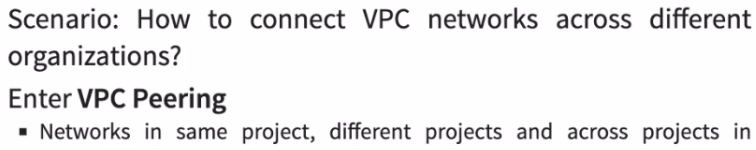
VPC peers always exchange subnet routes that don't use privately used public IP addresses. Peers must explicitly import subnet routes that use privately used public IP addresses to receive routes from peers that export them.

Some useful points

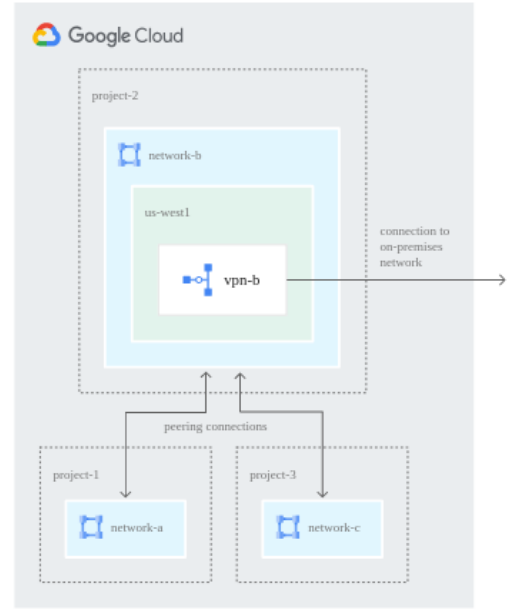
1. VPC network peering peer two vpc regardless of if they present in same project or organization.
2. We cannot limit or restrict sharing of subnet route between VPC peer network.
3. Dynamic route can conflict with subnet route in peering. In this case subnet route is dropped.
4. VPC peering is non-transitive.
5. We cannot use tag or service account across vpc peered network in firewall rule.
6. In VPC peer network benefit of importing and exporting custom route is that if route change in vpc network , vpc networking which is importing the route automatically detect it.
7. Custom route exported from one vpc network cannot be exported to another vpc network transitively.
8. When importing custom route from peer network the destination address remain as is but the next hop change to peer network.
9. If more than one VPC network are peered they cannot have overlapping ip range for subnet in any of network
10. 

11. During the time of expanding of ip range of subnet also overlapping is checked in vpc peering
12. We can use either Cloud VPN or Cloud Interconnect to securely connect your on-premises network to your VPC network. If you export custom routes, peered VPC networks can also connect to your on-premises network. On the on-premises side, you must create routes so that traffic going to VPC networks is directed to the VPN tunnel.





13. Transit Network (All networks are exporting and importing custom routes. network-b acts as the transit network, where the VPN tunnel is located.)



14.

different organizations can be peered

- All communication happens using internal IP addresses
  - Highly efficient because all communication happens **inside Google network**
  - Highly secure because **not accessible from Internet**
  - **No data transfer charges** for data transfer between services
- (REMEMBER) Network administration is NOT changed:
  - Admin of one VPC do not get the role automatically in a peered network

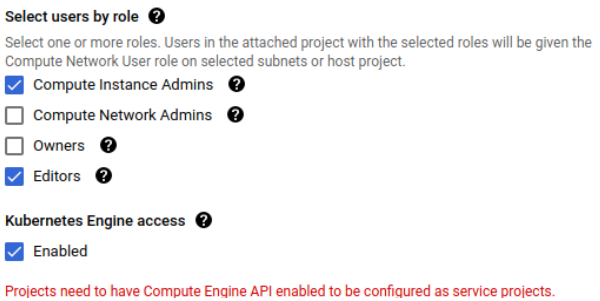
Shared VPC

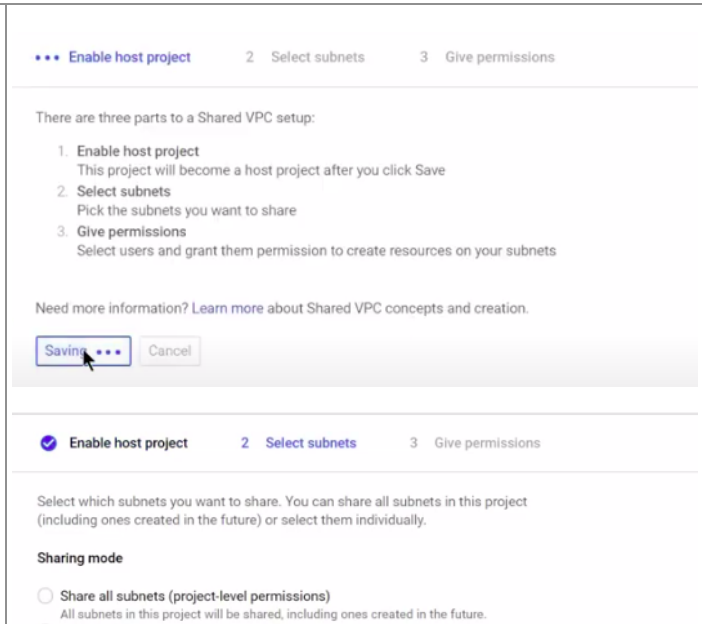
Shared VPC lets you share subnets with other projects. You can then create resources (like VM instances) on those subnets. Learn more

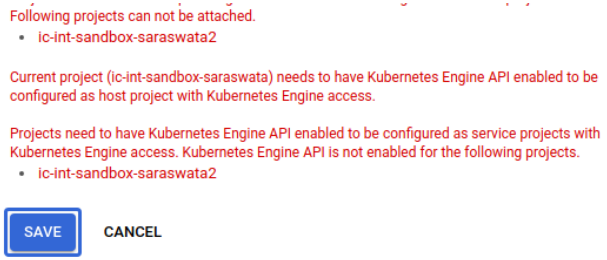
Shared VPC operate in same organization whereas VPC peering work across organization.

Shared VPC allows an organization to connect resources from multiple projects to a common Virtual Private Cloud (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.

Select role require related API to be enabled

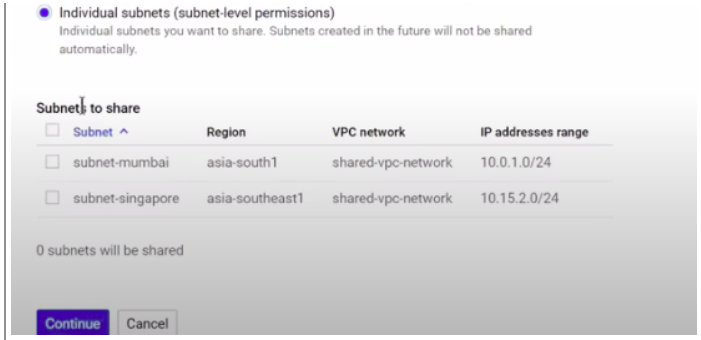


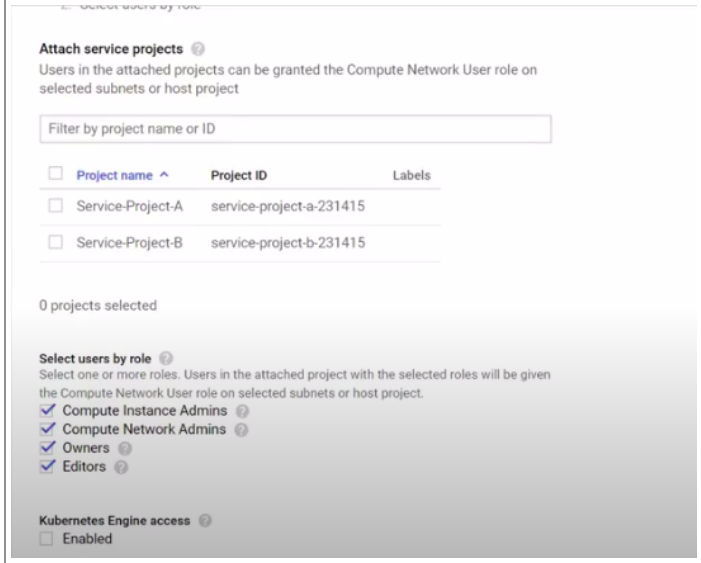




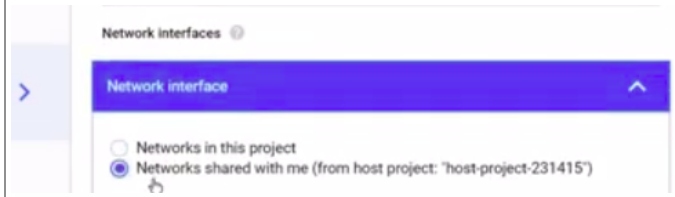
Some useful points

1. Projects in shared network must be in single organization, only exception is while migration.
2. To setup shared VPC use required Compute Shared VPC Admin role.
3. Organization can have multiple host project but a service project can only belong to one host project. (This type of architecture is suitable for Testing and Production environment)
4. Internal DNS name of vm use the project id of service project even though they point to ip of host project.
5. We can create private zone in host project and authorize the zone for shared vpc network.
6. Organization Admin create Shared VPC Admin (compute.xpnAdmin, resourceManager.projectIAMAdmin role)
7. Shared VPC Admin create Service Project Admin (compute.networkuser, compute.instanceAdmin role) in service project. (Access to all networking resource except firewall rule and networking resources)
8. Shared VPC can optionally create Network Admin (compute.networkAdmin) and Security Admin (compute.securityAdmin) in host project (Manage firewall rule and ssl certificate)
9. Each service project is billed separately





While Creating VM we can select shared VPC

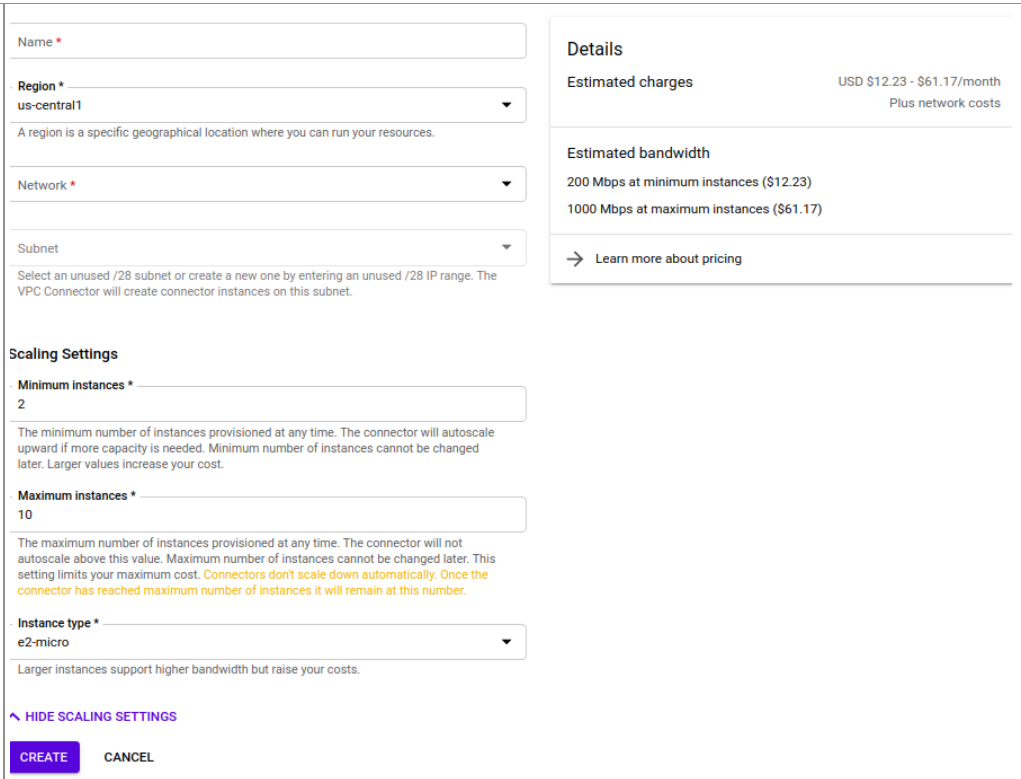
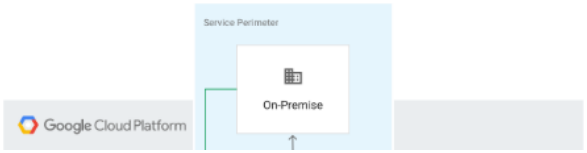


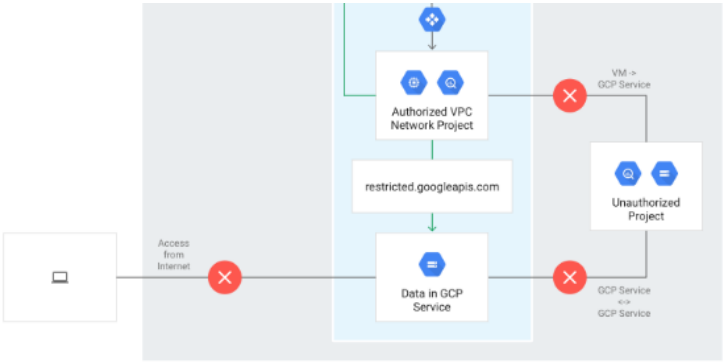
Shared VPC

- Scenario: Your organization has multiple projects. You want resources in different projects to talk to each other?
  - How to allow resources in different projects to talk with internal IPs securely and efficiently?
- Enter **Shared VPC**
  - Created at organization or shared folder level (Access Needed: Shared VPC Admin)

	▪ Allows VPC network to be shared between projects in same organization ▪ Shared VPC contains one host project and multiple service projects: ◦ **Host Project** - Contains shared VPC network ◦ **Service Projects** - Attached to host projects • Helps you achieve **separation of concerns**: ▪ Network administrators responsible for Host projects and Resource users use Service Project

Others

Serverless VPC Access	
VPC Service Controls VPC Service Controls improves your ability to mitigate the risk of data exfiltration from Google Cloud services such as Cloud Storage and BigQuery. You can use VPC Service Controls to create perimeters that protect the resources and data of services that you explicitly specify. **Extend perimeters to authorized VPN or Cloud Interconnect** You can configure private communication to Google Cloud resources from VPC networks that span hybrid environments with Private Google Access on-premises extensions. A VPC network	Securing GCP Projects with VPC Service Controls 

must be part of a service perimeter for VMs on that network to privately access managed Google Cloud resources within that service perimeter.	
	Securing GCP Projects with VPC ...
Private Service Connect	gcp private service connect #gcp #gcpcloud

Using Private Service Connect from on-premises hosts

If your on-premises network is connected to a VPC network, you can use Private Service Connect to access Google APIs and services from on-premises hosts using the internal IP address of the Private Service Connect endpoint.

- Your on-premises network must be connected to a VPC network using either Cloud VPN tunnels or Cloud Interconnect attachments (VLANs).
- The Private Service Connect endpoint is in the VPC network that is connected to your on-premises network.
- The on-premises network must have appropriate routes for the Private Service Connect endpoint. Configure a Cloud Router custom route advertisement to announce routes for the Private Service Connect endpoint on the BGP session that manages routes for the Cloud VPN tunnel or Cloud Interconnect attachment (VLAN).
- You must configure on-premises systems so that they can make queries to your private DNS zones.

  If you've implemented the private DNS zones using Cloud DNS, complete the following steps:

  - Create an inbound server policy in the VPC network to which your on-premises network connects.
  - Identify the inbound forwarder entry points, in the regions where your Cloud VPN tunnels and Cloud Interconnect attachments (VLANs) are located, in the VPC network to which your on-premises network connects.
  - Configure on-premises systems and on-premises DNS name servers to forward the DNS names for the Private Service Connect endpoints to an inbound forwarder entry point in the same region as the Cloud VPN tunnel or Cloud Interconnect attachment (VLAN) that connects to the VPC network.

gcp private service connect #gcp...

Adding internal load balancer as attachment to private service connect

Permissions required for this task

Console gcloud API

1. In the Google Cloud Console, go to the **Private Service Connect** page.

   Go to Private Service Connect
2. Click the **Published services** tab.
3. Click **Publish service**.
4. Select the **Internal load balancer** associated with the service you want to publish.

   The network and region fields are populated with the details for the selected internal load balancer.
5. If prompted, select the **Forwarding rule** associated with the service that you want to publish.
6. For **Service name**, enter a name for the service attachment.
7. Select one or more **Subnets** for the service. If you want to add a new subnet, you can create one:
   a. Click **Reserve new subnet**
   b. Enter a **Name** and optional **Description** for the subnet.
   c. Select a **Region** for the subnet.
   d. Enter the **IP range** to use for the subnet and click **Add**.
8. If you want to view consumer connection information, select **Use Proxy Protocol**. For more information, see Viewing consumer connection information.
9. Select **Automatically accept connections for all projects**.
10. Click **Add service**.

We can explicitly allow and deny access from a project.

Cloud IAP

Header name	Description	Example value
X-Goog-Authenticated-User-Email	The user's email address	accounts.google.com:example@gmail.com
X-Goog-Authenticated-User-Id	A persistent, unique identifier for the user.	accounts.google.com:userIDvalue