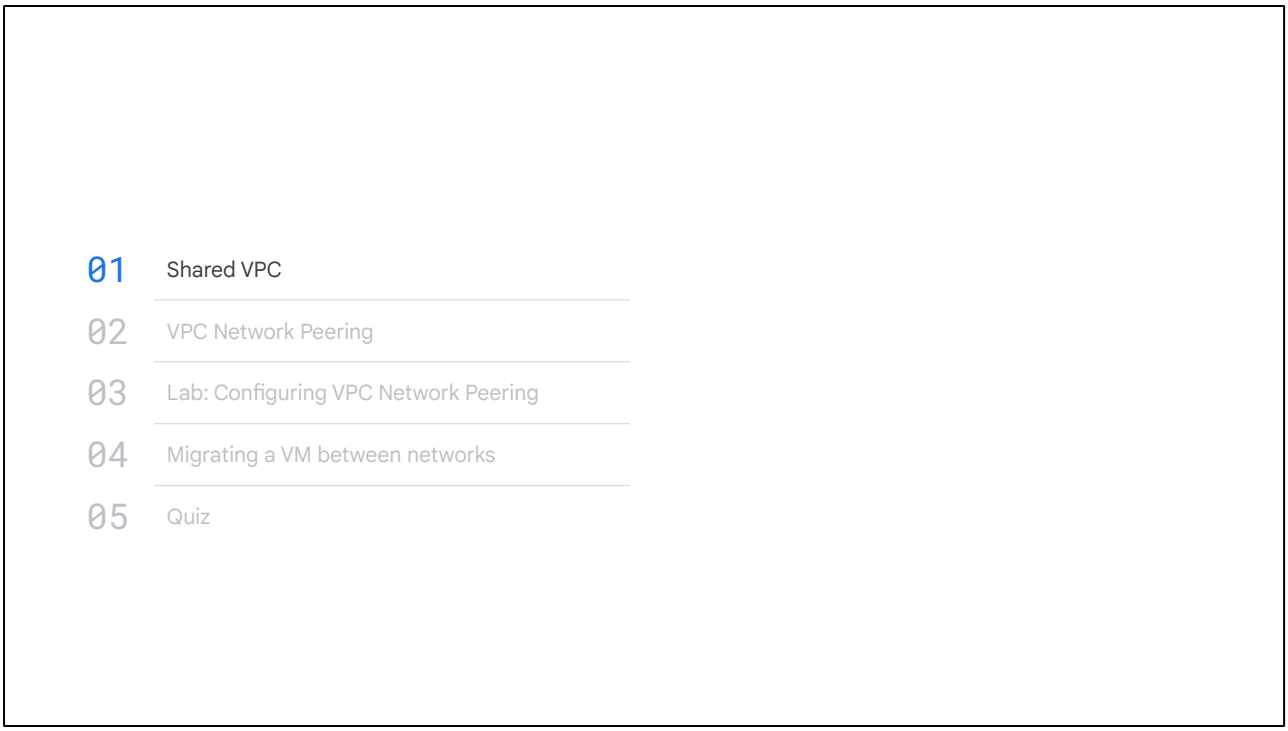
Networking in Google Cloud

Module 3: Sharing Networks across Projects

Welcome to the Sharing Networks across Projects module. This is the third module of the Networking in Google Cloud: Defining and Implementing Networks course.



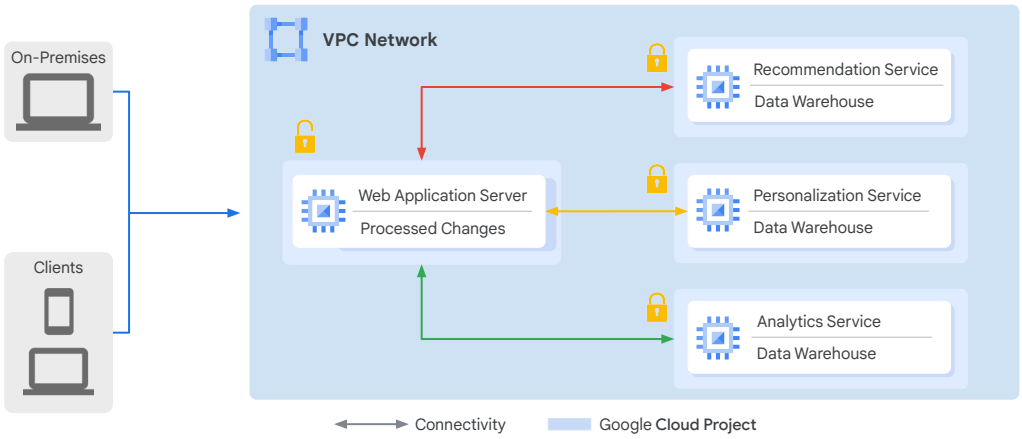
In this module, we are going to cover two configurations for sharing VPC networks across Google Cloud projects.

First, we will go over shared VPC, which allows you to share a network across several projects in your Google Cloud organization.

Then, we will go over VPC Network Peering, which allows you to configure private communication across projects in the same or different organizations.

Let’s start by talking about shared VPC.

Shared VPC

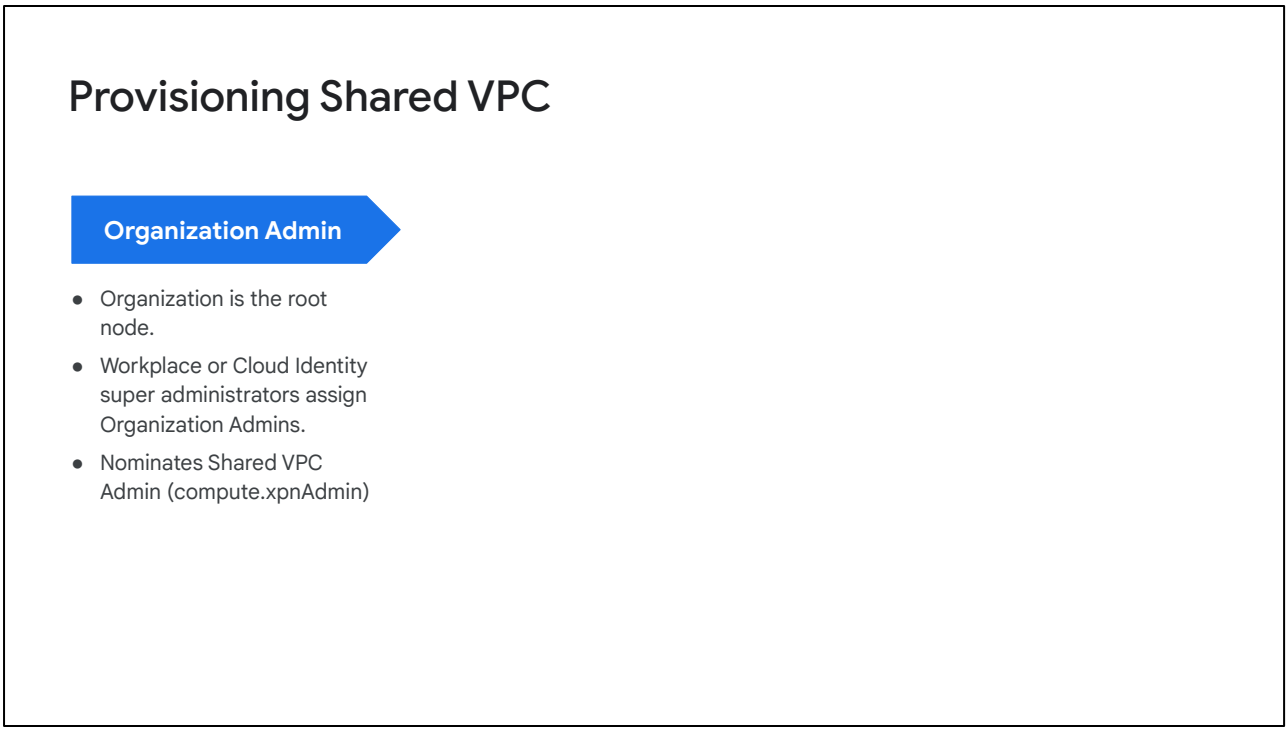


Shared VPC networks allow an organization to connect resources from multiple projects to a common VPC network. The resources can communicate with each other securely and efficiently using internal IP addresses from that network. Shared VPC allows centralized network administration, which provides the capability to implement a security best practice of least privilege for network administration, auditing, and access control. Shared VPC enables resources across multiple projects and networks to communicate with each other without using Cloud VPN or VPC Network Peering.

For example, in this diagram there is one network that belongs to the Web Application Server's project. This network is shared with three other projects, namely the Recommendation Service, the Personalization Service, and the Analytics Service. Each of those service projects has instances that are in the same network as the Web Application Server, allowing for private communication to that server, using internal IP addresses. The Web Application Server communicates with clients and on-premises compute resources using the server's external IP address. The backend services, on the other hand, can't be reached externally, because they only communicate using internal IP addresses.

When you use shared VPC, you designate a project as a host project and attach one or more other service projects to it. In this case, the Web Application Server's project is the host project, and the three other projects are the service projects.

The overall VPC network is called the shared VPC network.



Shared VPC makes use of Cloud IAM roles for delegated administration.

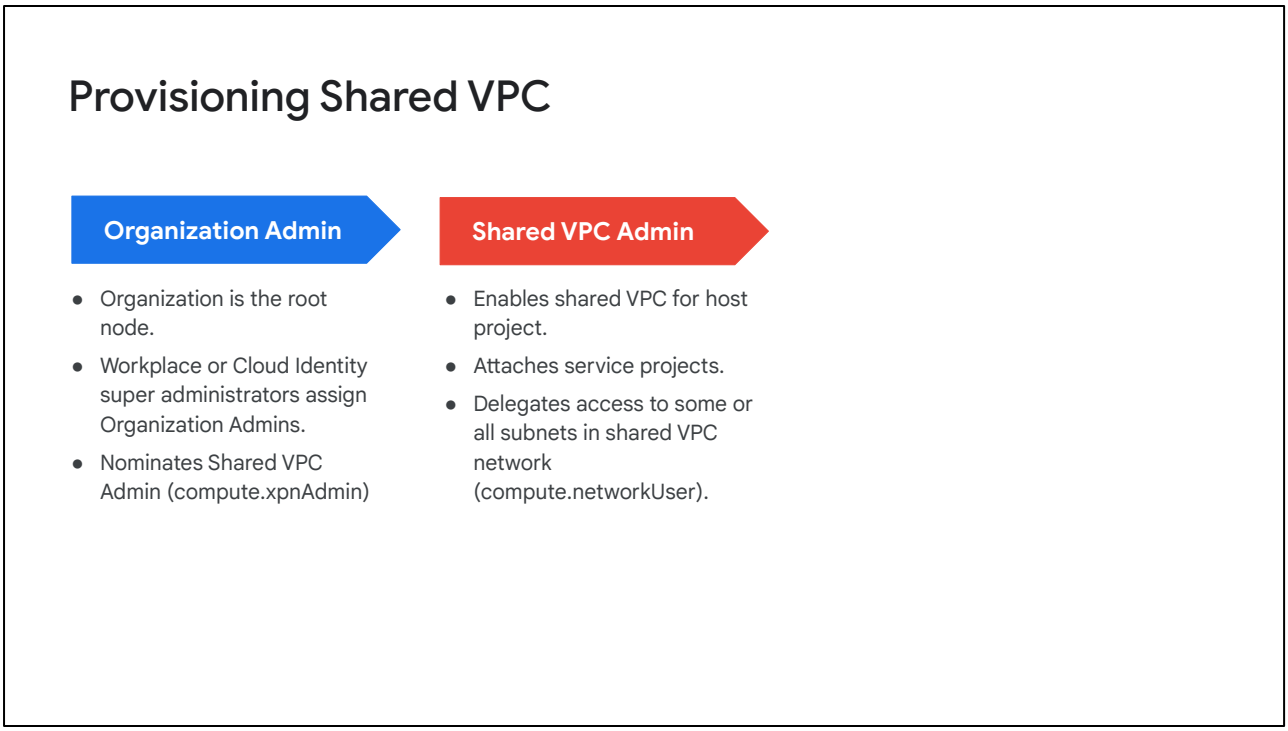
Let's review how to provision shared VPC by focusing on the required administrative roles.

The first required role is the organization admin. The Organization resource represents an organization, for example, a company, and is the root node in the Google Cloud resource hierarchy.

The Google Workplace or Cloud Identity super administrators are the first users who can access the organization, and they assign the organization admin role to users.

The organization admin's role in provisioning shared VPC is to nominate Shared VPC Admins by granting them appropriate project creation and deletion roles, and the compute.xpnAdmin role for the organization.

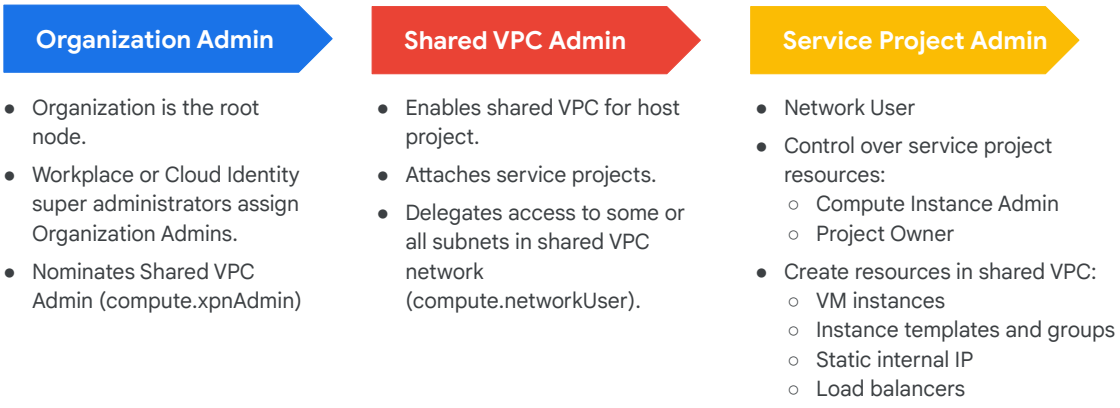
Note that shared VPC is also referred to as "XPN" in the API and command-line interface.



Next, the Shared VPC Admin performs various tasks necessary to set up shared VPC. These tasks include enabling shared VPC on the host project, attaching service projects, and delegating access to subnets to Service Project Admins by granting the compute.networkUser role.

Typically, a Shared VPC Admin is also the project owner for a given host project.

Provisioning Shared VPC



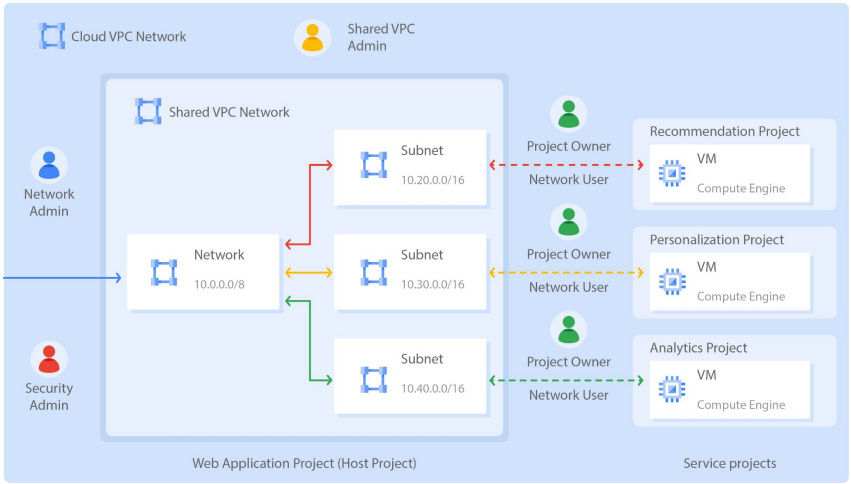
In addition to being a Network User, Service Project Admins also maintain ownership and control over resources defined in their service projects.

The Service Project Admins must at least have the compute.instanceAdmin role to the corresponding service project. However, typically, the Service Project Admins are project owners of their service projects. This allows them to create and manage resources in the shared VPC.

These resources could be:

- VM instances
- Instance templates and groups
- static internal IP addresses
- and load balancers.

Shared VPC

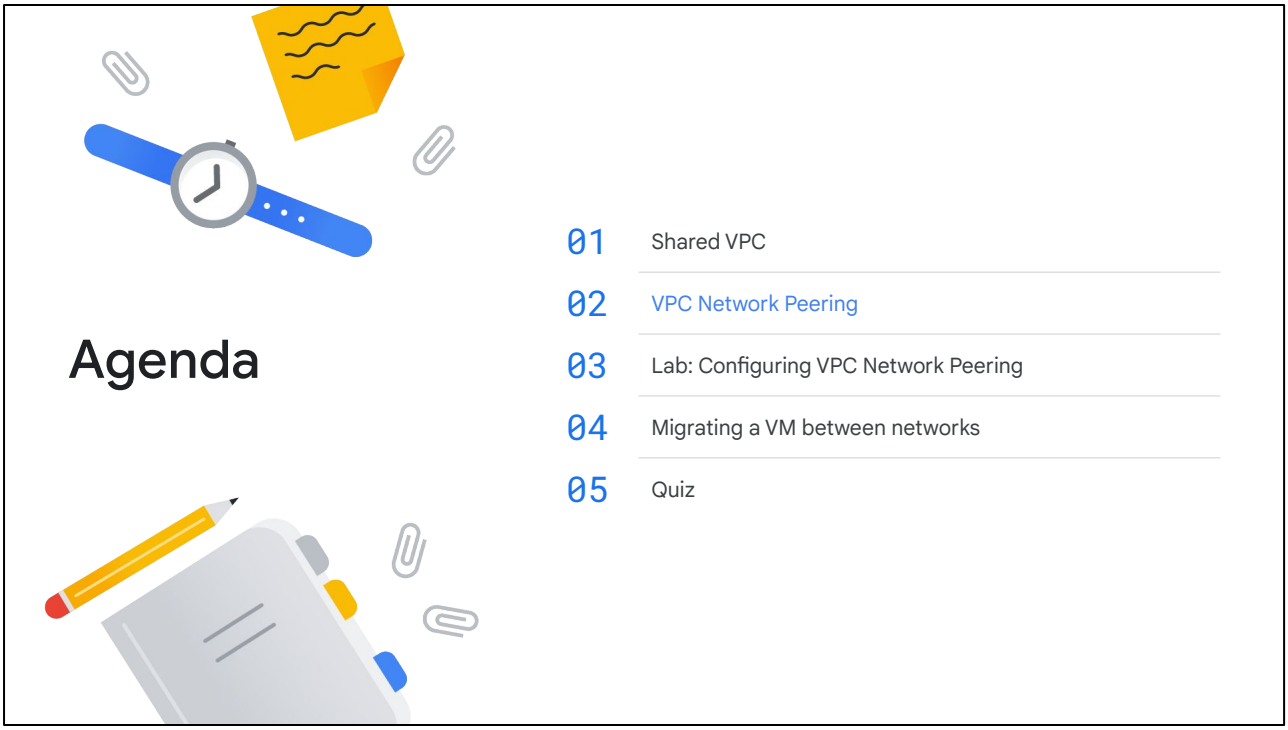


Let's return to our original example that had one host project and 3 service projects:

In this diagram, the Shared VPC Admin, which was nominated by an organization admin, configured the Web Application Project to be a host project with subnet-level permissions. Doing so allowed the Shared VPC Admin to selectively share subnets from the VPC network.

Next, the Shared VPC Admin attached the three service projects to the host project and gave each project owner the Network User role for the corresponding subnets. Each project owner then created VM instances from their service projects in the shared subnets. By the way, billing for those VM instances is attributed to the project where the resources are created, which are the service projects.

Shared VPC Admins have full control over the resources in the host project, including administration of the shared VPC network. They can optionally delegate the Network Admin and Security Admin roles for the host project. Overall, shared VPC is a centralized approach to multi-project networking because security and network policy occurs in a single designated VPC network.



Next, let's talk about VPC Network Peering and how it's different from Shared VPC. We will also consider talk about the pros and cons of VPC Network Peering versus Shared VPC.

VPC Network Peering

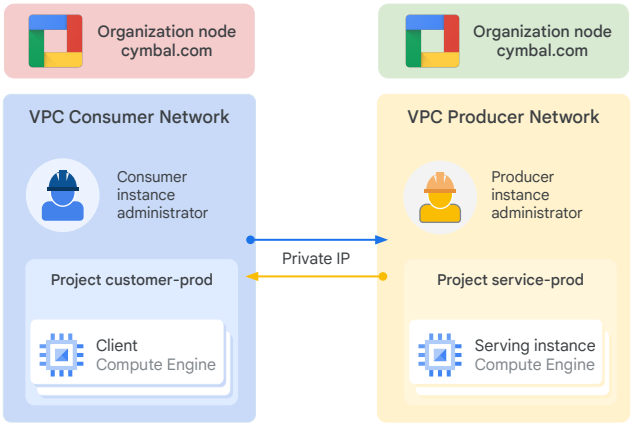
- VPC Network Peering allows private RFC 1918 connectivity across two VPC networks.
- You can peer VPCs:
  - Within the same project
  - That are in different organizations.

VPC Network Peering allows private RFC 1918 connectivity across two VPC networks. Even if both VPC networks belong to the same project or the same organization, you can still peer them.

Remember that each VPC network will have firewall rules that define what traffic is allowed or denied between the peered networks.

VPC Network Peering

- VPC Network Peering allows private RFC 1918 connectivity across two VPC networks.
- You can peer VPCs:
  - Within the same project
  - That are in different organizations.

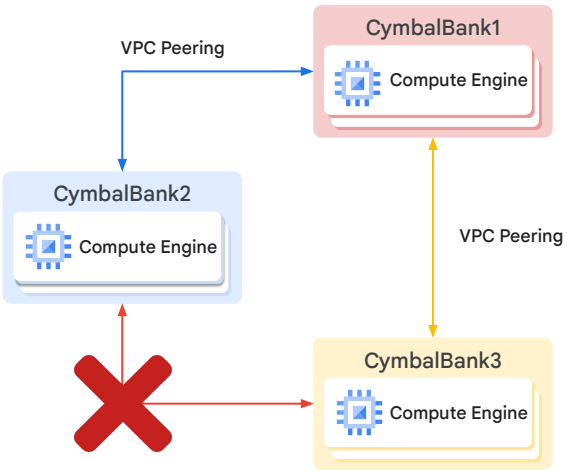


Let's look at an example, shown on the right. Two organizations represent a consumer and a producer, respectively. Each organization has its own organization node, VPC network, VM instances, Network Admin, and Instance Admin. To successfully establish VPC Network Peering, two peering relationships must be created. The Producer Network Administrator must peer the producer network with the consumer network. The Consumer Network Admin must peer the consumer network with the producer network. When both peering connections are created, the VPC Network Peering session becomes active and routes are exchanged. The peering relationships let the VM instance use their internal IP addresses to communicate privately.

VPC Network Peering is a decentralized or distributed approach to multi-project networking. Each VPC network may remain in the control of separate administrator groups and maintains its own global firewall and routing tables. Historically, such projects would consider external IP addresses or VPNs to facilitate private communication between VPC networks. However, VPC Network Peering does not incur the network latency, security, and cost drawbacks that are present when you use external IP addresses or VPNs.

Using VPC Network Peering

- You can peer Compute Engine, Kubernetes Engine, and App Engine flexible environments.
- Peered VPC networks remain administratively separate.
- Each side of a peering association is set up independently.
- No subnet IP ranges overlap across peered VPC networks.
- Transitive peering is not supported.



Let's talk about a few points to remember when using VPC Network Peering. First of all, VPC Network Peering works with Compute Engine, Google Kubernetes Engine, and App Engine flexible environments.

Peered VPC networks remain administratively separate. In other words, routes, firewalls, VPNs, and other traffic management tools are administered and applied separately in each of the VPC networks. These tools are not managed centrally for all network peers.

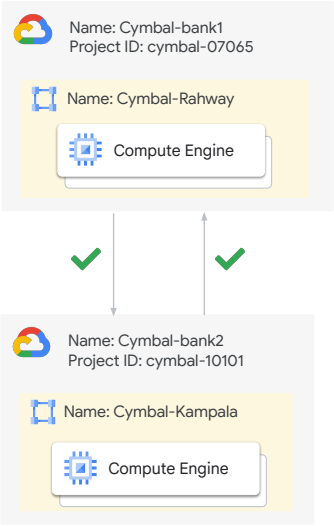
Each side of a VPC Network Peering association is set up independently. Peering will be active only when the configuration from both sides matches. This arrangement allows either side to delete the peering association at any time.

A subnet CIDR prefix in one peered VPC network cannot overlap with a subnet CIDR prefix in another peered network. For example, two auto mode VPC networks that only have the default subnets cannot peer.

Only directly peered networks can communicate, which means that transitive peering is not supported. For example, suppose the VPC network CymbalBank1 is peered with CymbalBank2 and CymbalBank3. In that scenario, CymbalBank2 and CymbalBank3 are not directly connected. VPC network CymbalBank2 cannot communicate with VPC network CymbalBank3 over a peered connection. If CymbalBank1 offered SaaS services to CymbalBank2 and CymbalBank3, this situation could be critical.

Initiating VPC Network Peering

- To initiate VPC Network Peering with another VPC network, you need the name of the other VPC network.
- If the VPC network is located in another project, you also need the project ID.
- The peering connection is not active until it's initiated from both VPC networks.



Before you begin, you must have the name of the VPC network to which you will peer with. If that VPC network is located in another project, you must also have the project ID.

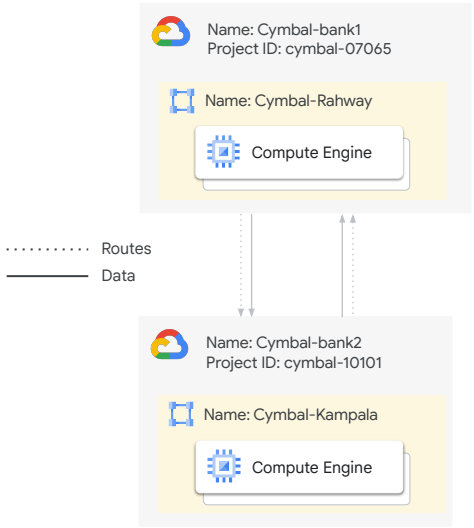
A peering configuration establishes the intent to connect to another VPC network. The VPC networks are not connected until each one has a peering configuration for the other. After the other network has a corresponding configuration to peer with your network, the peering state changes to active in both networks. At that point, the VPC networks are connected through VPC Network Peering.

Until both VPC networks create peering configurations for each other, no peering connection exists between them. The peering state remains inactive. An inactive peering state indicates that there is not yet a full VPC Network Peering connection.

Suppose you want to peer the two VPC networks shown on the slide. Let’s begin with the Cymbal-Rahway VPC network. You must know the name of the other VPC network, which is Cymbal-Kampala. Because Cymbal-Kampala is not in the same project as Cymbal-Rahway, you must also have the project ID. Cymbal-Kampala is in the Cymbal-bank2 project, whose project ID is cymbal-10101.

Sharing custom routes

- Sharing custom routes with peered VPC networks lets networks learn routes directly from their peered networks.
- If a custom route in a peered network is updated, your VPC network automatically learns and uses the updated custom route.
- You can import and export routes.



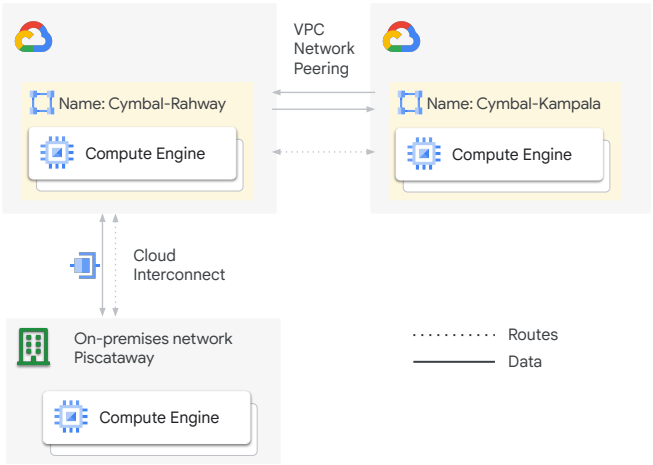
Sharing custom routes with peered VPC networks let networks learn routes directly from their peered networks.

For example, if a custom route in a peered network is updated, your VPC network automatically learns and uses the updated custom route. You are not required to configure any additional settings in your VPC network.

When you create or modify a peering configuration, you can choose to import routes, export routes, or both. The peer network administrator must similarly configure their peering configuration before routes are exchanged. This process ensures that both network administrators explicitly agree to exchange custom routes before they are exchanged.

A sample scenario

Share custom routes so that peered networks can reach your on-premises network.

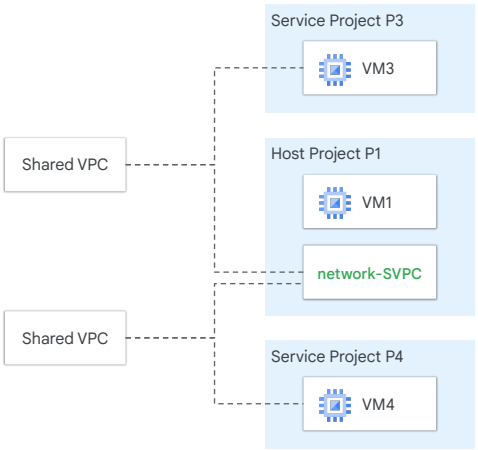


Let's consider a sample scenario.

If you have a VPN tunnel or a Cloud Interconnect connection, you can share custom routes. Sharing these routes enables peered networks to reach your on-premises network, as shown in the graphic. For dynamic routes, you must add Cloud Router custom route advertisements in your VPC network. These advertisements announce peered network subnets to your on-premises network.

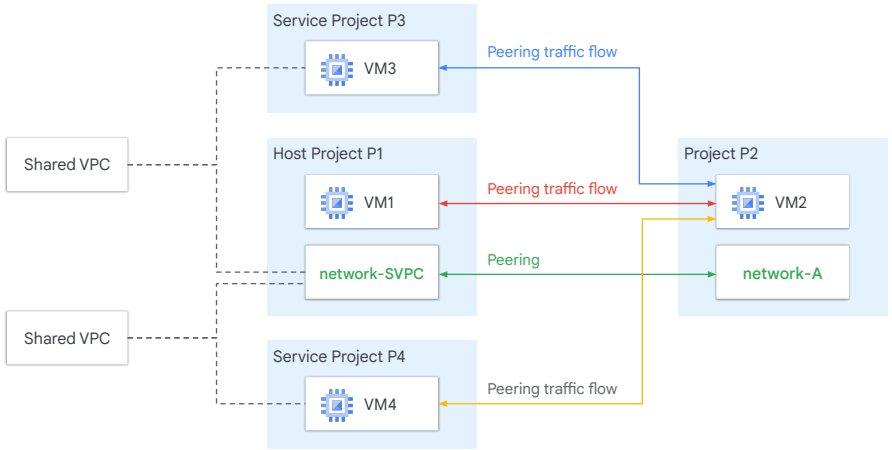
Local routes are always preferred over dynamic routes that are learned by using VPC Network Peering.

Peering with a Shared VPC network



In Google Cloud, you can peer with a Shared VPC network. Here you can see an example of a Shared VPC network called network-SVPC. network-SVPC is in host project P1. Service projects P3 and P4 can attach VM instances to network-SVPC, which enables private communication between VMs 1, 2, and 4.

Peering with a Shared VPC network



If we establish a peering session between network-A and network-SVPC, all VM instances will have private, internal IP connectivity. Each VPC network has firewall rules that define which traffic is allowed or denied between the networks.

You can also set up VPC Network Peering between two Shared VPC networks.

Shared VPC versus VPC Network Peering

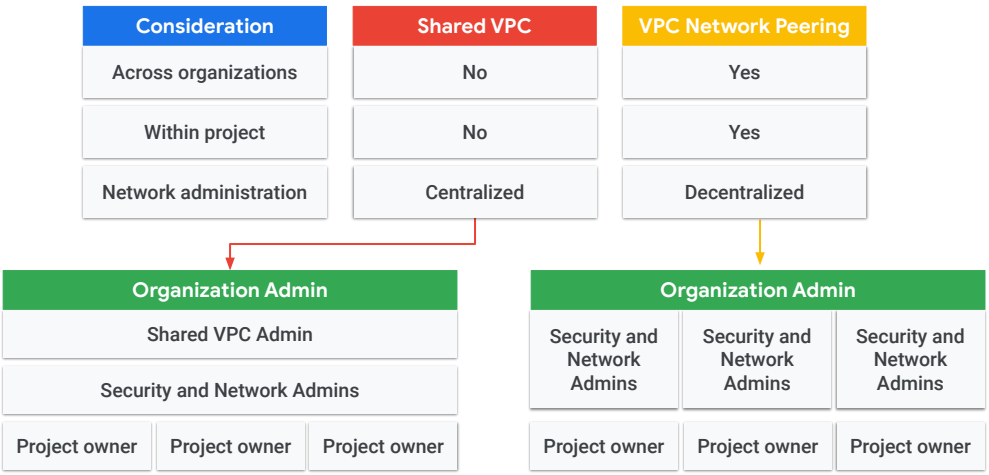
Consideration	Shared VPC	VPC Network Peering
Across organizations	No	Yes
Within project	No	Yes
Network administration	Centralized	Decentralized

Let's compare both of these configurations to help you decide which is appropriate for a given situation.

If you want to configure private communication between VPC networks in different organizations, you must use VPC Network Peering. Shared VPC only works within the same organization.

If you want to configure private communication between VPC networks in the same project, you must use VPC Network Peering. The VPC networks can be in the same project, but it's not required. Shared VPC only works across projects.

Shared VPC versus VPC Network Peering



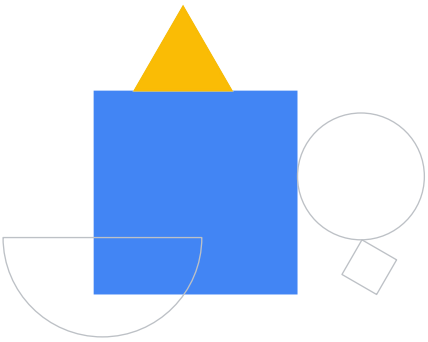
The biggest difference between the two configurations is the network administration models. Shared VPC is a centralized approach to multi-project networking, because security and network policy occurs in a single designated VPC network.

In contrast, VPC Network Peering is a decentralized approach. Each VPC network is controlled by administrator groups in that VPC network's organization. Each VPC network can maintain its own global firewall and routing tables.

For more information about the limits of VM instances per VPC network, see Per Network on the Quotas and Limits page of the Google Cloud documentation.

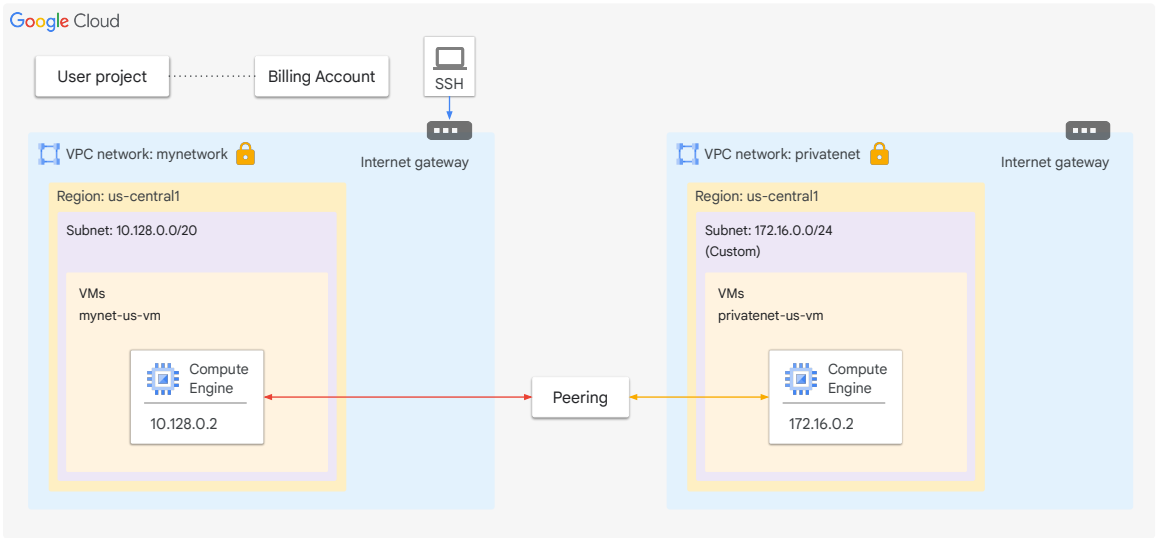
Lab intro

Configuring VPC Network Peering

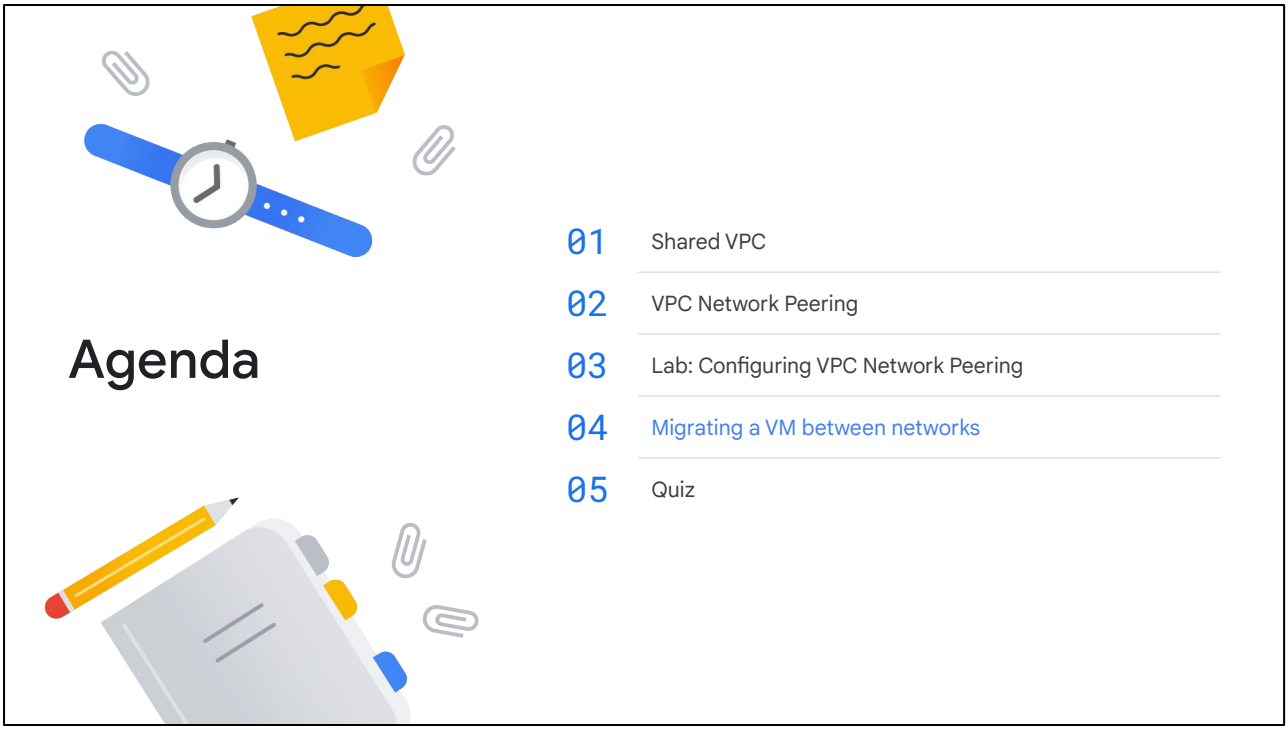


In this lab, you will learn how to perform the following tasks:

- Explore connectivity between non-peered VPC networks.
- Configure VPC network peering.
- Verify private communication between peered VPC networks.
- Delete VPC network peering.

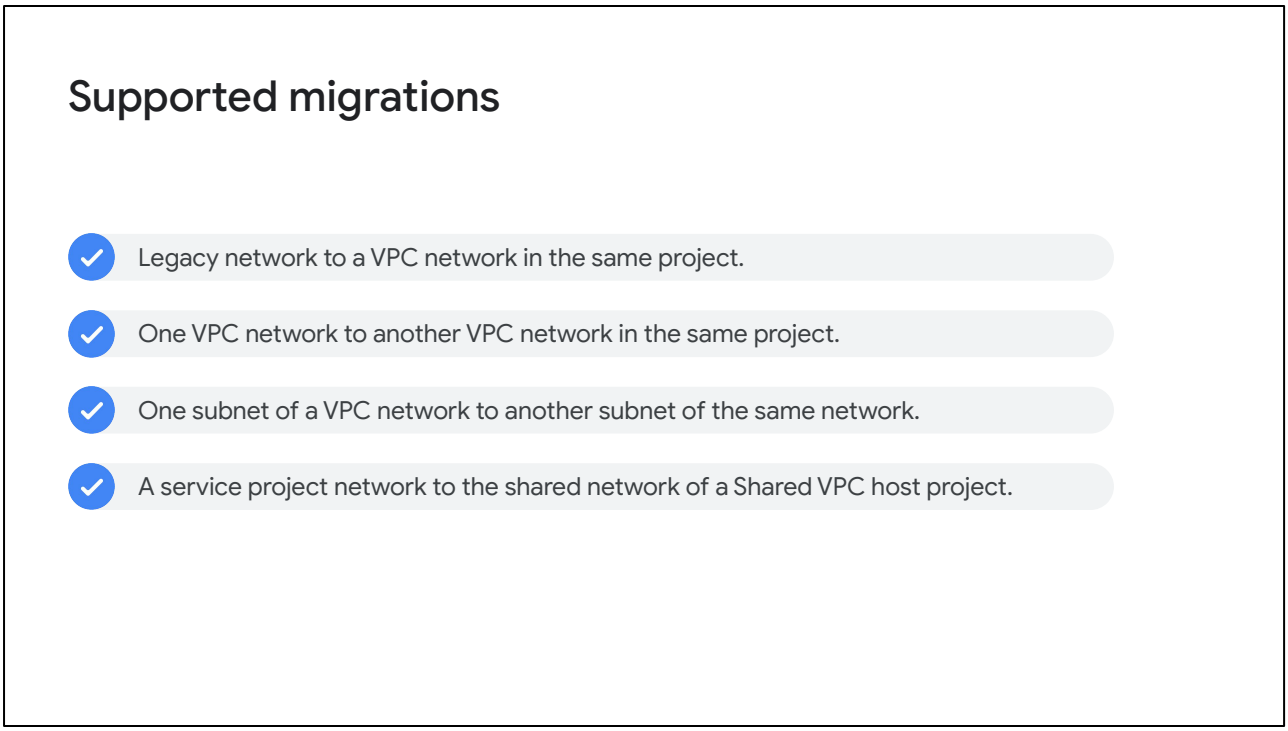


On the screen, you can see a general topology of mynetwork and privatenet. Once VPC Network Peering is implemented, VMs in each VPC network will be able to communicate using internal IP addresses.



Finally, let's discuss how to migrate a VM instance from one network to another.

When a VM is connected to more than one network using multiple interfaces, the migration process updates one of the interfaces and leaves the rest in place.



The migrations supported are:

- From legacy network to a VPC network in the same project.
- From one VPC network to another VPC network in the same project.
- From one subnet of a VPC network to another subnet of the same network.
- And from a service project network to the shared network of a Shared VPC host project.

In all cases, the VM stays in the region and zone where it was before. Only the attached network changes.

Migration requirements

- The VM must be stopped before it can be migrated.
- The VM must not be in an instance group or network endpoint group (NEG).
  - If the VM is in an unmanaged instance group or NEG, you must take it out of the group before migrating it.
  - VMs in managed instance groups cannot be migrated.
  - You can move instances in target pools without removing them first.

Before you migrate a VM, you must meet the following requirements:

- The migration is a "cold" migration. The VM must be stopped before it can be migrated.
- The VM must not be in an instance group or network endpoint group (NEG).
  - If the VM is in an unmanaged instance group or NEG, you must take it out of the group before migrating it.
  - VMs in managed instance groups cannot be migrated. Instead, you must copy your instance template to the new network and use it to rebuild the managed instance group.
  - A target pool is a group of Google Compute Engine instances that receive incoming traffic from a load balancer. You can move instances in target pools without removing them first. The target pool expands to cover both networks.

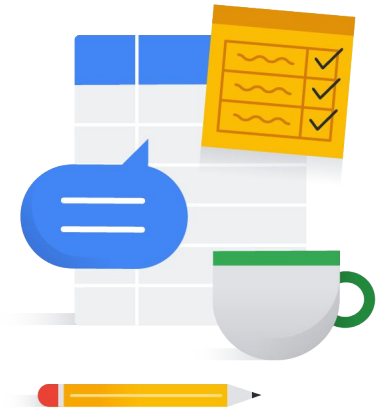
Migration limitations

- A VM interface cannot be migrated to a legacy network.
- The MAC address allocated to the network interface changes during the migration.
- If migrating the VM to a network or subnet with a different IP range, the internal IP address of your instance must change.
- If migrating to a subnet with the same IP range, the old IP address can be kept (if not in use at destination).
- If the target subnet does not have the same IP range as the source, the IP address of the interface changes to match the new subnet range.
- An existing external IP address can be kept in the new location subject to permissions.

There are limitations to migrating and you should consider the following:

- You cannot migrate a VM interface to a legacy network.
- The MAC address allocated to the network interface will change during the migration. This could have an impact on services tightly coupled with MAC addresses such as third-party license agreement.
- If you're migrating the VM to a network or subnet with a different IP range, the internal IP address of your instance must change.
- If you're migrating to a subnet with the same IP range, you can keep the old IP address, as long as it is not already in use at the destination, by specifying it during the migration.
- If the target subnet does not have the same IP range as the source, then the IP address of the interface changes to match the new subnet range.
- Lastly, you can keep the VM's existing external IP address in the new location. However, to do this you must have the compute.subnetworks.useExternalIp permission on the target network, and the target network cannot have external IP addresses disabled by the constraints/compute.vmExternalIpAccess constraint.

Debrief



In this module, we looked at shared VPC and VPC Network Peering, which are two configurations for sharing VPC networks across Google Cloud projects.

You got to explore VPC Network Peering in a lab and we compared both configurations and their network administration models to help you decide when to choose which.

Google Cloud's flexibility to support multiple approaches to network administration allows organizations like yours to more carefully map resource policies, administrative controls, and related accounting to existing structures.

In addition, administrators can carefully control the manner in which environments interact with each other, on-premises networks, and the public internet.