Shared VPC allows an organization (/resource-manager/docs/cloud-platform-resource-hierarchy) to connect resources from multiple projects to a common Virtual Private Cloud (VPC) network (/vpc/docs/vpc), 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 (#resources_that_can_be_attached_to_shared_vpc_networks_from_a_service_project) from service projects can use subnets in the Shared VPC network.

Shared VPC lets organization administrators (/resource-manager/docs/cloud-platform-resource-hierarchy#organizations) delegate administrative responsibilities, such as creating and managing instances, to Service Project Admins (#iam_in_shared_vpc) while maintaining centralized control over network resources like subnets, routes, and firewalls. This model allows organizations to do the following:

- 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 (#iam_in_shared_vpc) 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 (#billing) section for more details.

Shared VPC is also referred to as "XPN" in the API.

Shared VPC connects projects within the same organization (/resource-manager/docs/creating-managing-organization). Participating host and service projects cannot belong to different organizations. Linked projects can be in the same or different folders (/resource-manager/docs/creating-managing-folders), but if they are in different folders the admin must have Shared VPC Admin (#iam_in_shared_vpc) rights to both folders. Refer to the Google Cloud resource hierarchy (/resource-manager/docs/cloud-platform-resource-hierarchy) for more information about organizations, folders, 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 (#shared_vpc_networks). A Shared VPC Admin (#iam_in_shared_vpc) must first enable (/vpc/docs/provisioning-shared-vpc#enable-shared-vpc-host) 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 (/vpc/docs/provisioning-shared-vpc#create-shared) 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 (#example_multiple_host_projects) 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.

A **Shared VPC network** is a VPC network (/vpc/docs/vpc) defined in a host project and made available as a centrally shared network for eligible resources (#resources_that_can_be_attached_to_shared_vpc_networks_from_a_service_project) in service projects. Shared VPC networks can be either auto or custom mode (/vpc/docs/vpc#subnet-ranges), but legacy networks (/compute/docs/vpc/legacy) 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 (#iam_in_shared_vpc).

Organization policies and IAM permissions work together (/resource-manager/docs/organization-policy/overview#differences_from_iam) to provide different levels of access control. Organization policies enable you to set controls at the organization, folder or project level.

If you are an organization policy administrator (/resource-manager/docs/organization-policy/creating-managing-policies), you can specify the following Shared VPC constraints in an organization policy:

- You can limit the set of host projects to which a non-host project or non-host projects in a folder or organization can be attached. The constraint applies when a Shared VPC Admin attaches a service project with a host project. The constraint doesn't affect existing attachments (/resource-manager/docs/organization-policy/overview#violations). Existing attachments remain intact even if a policy denies new ones. For more information, refer to the `constraints/compute.restrictSharedVpcHostProject` (/resource-manager/docs/organization-policy/org-policy-constraints) constraint.
- You can constrain which Shared VPC subnets that service projects can access at the project, folder, or organization level. The constraint applies when you create new resources in the specified subnets and doesn't affect existing resources (/resource-manager/docs/organization-policy/overview#violations). Existing resources continue to operate normally in their subnets even if a policy prevents new resources from being added. For more information, refer to the `constraints/compute.restrictSharedVpcSubnetworks` (/resource-manager/docs/organization-policy/org-policy-constraints) constraint.

oduct or feature is in a pre-release state and might change or have limited support. For more information, see the pr stages (/products/#product-launch-stages).

Shared VPC makes use of Identity and Access Management (IAM) (/iam/docs/overview) roles for delegated administration. The following roles can be granted to IAM members (/iam/docs/overview#concepts_related_identity), such as users, Google groups, Google domains, or Google Cloud service accounts. If you need to contact any of these admins, you can look them up in your organization's or project's IAM policy (/iam/docs/overview#iam_policy). If you don't have the required permissions, you must contact a network or project administrator in your organization.

Administrator (IAM role)	Purpose
Organization Admin (/resource-manager/docs/cloud-platform-resource-hierarchy#organizations) (**`resourcemanager.organizationAdmin`**) • IAM member in the organization	Organization Admins have the **`resourcemanager.organizationAdmin`** role for your organization. They nominate Shared VPC Admins by granting them appropriate project creation and deletion roles (/iam/docs/understanding-roles#resource-manager-roles), and the *Shared VPC Admin* role for the organization. These admins can define organization-level policies, but specific folder and project actions require additional folder and project roles.
Shared VPC Admin (**`compute.xpnAdmin`** and **`resourcemanager.projectIamAdmin`**) • IAM member in the organization, or • IAM member in a folder (beta)	Shared VPC Admins have the *Compute Shared VPC Admin* (**`compute.xpnAdmin`**) (/iam/docs/understanding-roles#compute.xpnAdmin) and *Project IAM Admin* (**`resourcemanager.projectIamAdmin`**) (/iam/docs/understanding-roles#resourcemanager.projectIamAdmin) roles for the organization or one or more folders. They perform various tasks necessary to set up Shared VPC (/vpc/docs/provisioning-shared-vpc#setting_up_shared_vpc), such as enabling host projects, attaching service projects to host projects, and delegating access to some or all of the subnets in Shared VPC networks to Service Project Admins. A Shared VPC Admin for a given host project is typically its project owner as well. A user assigned the Compute Shared VPC Admin role for the organization has that role for all folders in the organization. A user assigned the role for a folder has that role for the given folder and any folders nested underneath it. A Shared VPC Admin can link projects in two different folders only if the admin has the role for both folders.
Service Project Admin (**`compute.networkUser`**) • IAM member in the organization, or • IAM member in a host project, or • IAM member in some subnets in the host project	A Shared VPC Admin defines a Service Project Admin by granting an IAM member the *Network User* (**`compute.networkUser`**) role (/compute/docs/access/iam#compute.networkUser) to either the whole host project or select subnets of its Shared VPC networks (#svc_proj_admins). Service Project Admins also maintain ownership and control over resources defined in the service projects, so they should have the *Instance Admin* (**`compute.instanceAdmin`**) role (/compute/docs/access/iam#compute.instanceAdmin.v1) to the corresponding service projects. They may have additional IAM roles to the service projects, such as project owner.

When defining each Service Project Admin, a Shared VPC Admin can grant permission to use the whole host project or just some subnets:

- **Project-level permissions**: A Service Project Admin can be defined to have permission to use all subnets (/vpc/docs/provisioning-shared-vpc#networkuseratproject) in the host project if the Shared VPC Admin grants the role of `compute.networkUser` for the whole host project to the Service Project Admin. The result is that the Service Project Admin has permission to use all subnets in all VPC networks of the host project, including subnets and VPC networks added to the host project in the future.
- **Subnet-level permissions**: Alternatively, a Service Project Admin can be granted a more restrictive set of permissions to use only some subnets (/vpc/docs/provisioning-shared-vpc#networkuseratsubnet) if the Shared VPC Admin grants the role of `compute.networkUser` for those selected subnets to the Service Project Admin. A Service Project Admin who only has subnet-level permissions is restricted to using only those subnets. After new Shared VPC networks or new subnets are added to the host project, a Shared VPC Admin should review the permission bindings for the `compute.networkUser` role to ensure that the subnet-level permissions for all Service Project Admins match the intended configuration.

Shared VPC Admins have full control over the resources in the host project, including administration of the Shared VPC network. They can optionally delegate certain network administrative tasks to other IAM members:

Administrator	Purpose
Network Admin • IAM member in the host project, or • IAM member in the organization	Shared VPC Admin define a Network Admin by granting an IAM member the *Network Admin* (`compute.networkAdmin`) role (/compute/docs/access/iam#compute.networkAdmin) to the host project. Network Admins have full control over all network resources except for firewall rules and SSL certificates.
Security Admin • IAM member in the host project, or • IAM member in the organization	A Shared VPC Admin can define a Security Admin by granting an IAM member the *Security Admin* (`compute.securityAdmin`) role (/compute/docs/access/iam#compute.securityAdmin) to the host project. Security Admins manage firewall rules and SSL certificates.

:ant: The Network Admin role does **not** include all of the permissions in the Network User role. IAM members having c
twork Admin role do not have permission to use the host project or subnets in its Shared VPC networks.

Shared VPC host projects are subject to standard per-project VPC quotas (/vpc/docs/quota#per_project). Shared VPC networks are subject to the per-network limits (/vpc/docs/quota#per_network) and per-instance limits (/vpc/docs/quota#per_instance) for VPC networks. Additionally, the relationship between host and service projects are governed by limits specific to Shared VPC (/vpc/docs/quota#shared-vpc).

Billing for resources that participate in a Shared VPC network is attributed to the service project where the resource is located, even though the resource uses the Shared VPC network in the host project.

- The rates and rules (/compute/pricing) used to calculate billing amounts for resources in service projects using a Shared VPC network is the same as if the resources were located in the host project itself.
- Billing for egress traffic generated by a resource is attributed to the project where the resource is defined:
    - Egress traffic from an instance is attributed to the project containing the instance. For example, if an instance is created in a service project but uses a Shared VPC network, any billing for egress traffic that it generates is attributed to its service project. In this way, you can use Shared VPC to organize resources into cost centers for your organization.
    - Costs associated with a load balancer (/compute/network-pricing#lb) are charged to the project containing the load balancer components. For more details about load balancing and Shared VPC, refer to the load balancing section (#shared_vpc_with_load_balancing).
    - Egress traffic to VPNs is attributed to the project containing the VPN Gateway resource. For example, if a VPN Gateway is created in the Shared VPC network, it is contained in the host project. Egress traffic through the VPN Gateway — regardless of which service project initiates the egress — is attributed to the host project.

The following service project resources **can** participate in Shared VPC subject to certain practical limitations (#practical_limitations):

- Compute Engine Instances (/compute/docs/instances/): Service Project Admins can create instances that use subnets to which they have been granted permission. Refer to creating an instance (/vpc/docs/provisioning-shared-vpc#creating_an_instance_in_a_shared_subnet) for more information.
- Compute Engine Instance Templates (/compute/docs/instance-templates/): Service Project Admins can create instance templates for use in a Shared VPC scenario. Refer to creating an instance template (/vpc/docs/provisioning-shared-vpc#creating_an_instance_template_in_the_shared_vpc_network) for more information.
- Compute Engine Instance Groups (/compute/docs/instance-groups/): Service Project Admins can create managed or unmanaged instance groups for use in a Shared VPC scenario. Refer to creating a managed instance group (/vpc/docs/provisioning-shared-vpc#creating_mig) for more information.
- Google Kubernetes Engine clusters (/kubernetes-engine/docs/): Refer to Setting up Clusters with Shared VPC (/kubernetes-engine/docs/how-to/cluster-shared-vpc) for more information.
- App Engine flexible environment instances (/appengine/docs/flexible/): Refer to Using the App Engine flexible environment on a Shared VPC network (/appengine/docs/flexible/python/using-shared-vpc) for more information.
- Internal IP Addresses (/compute/docs/ip-addresses/#networkaddresses): Refer to IP addresses (#ip_addresses) and reserving a static internal IP (/vpc/docs/provisioning-shared-vpc#reserve_internal_ip) for more information.
- Internal DNS (#shared_vpc_and_internal_dns): VMs and internal TCP/UDP load balancers can be referenced by the internal DNS names that Google Cloud creates automatically.
- Cloud DNS Private Zones (#shared_vpc_and_cloud_dns_private_zones): Private zones enable you to create custom DNS names for your Google Cloud resources.
- Load Balancing (/load-balancing/docs/load-balancing-overview): Refer to load balancing details (#shared_vpc_with_load_balancing) and creating an internal load balancer (/vpc/docs/provisioning-shared-vpc#creating_an_internal_load_balancer_forwarding_rule) for more information.

- Memorystore for Redis: Refer to Memorystore for Redis Networking (/memorystore/docs/redis/networking) for more information.

For other eligible resources, check the related product's documentation.

The following limitations apply to resources that are eligible for participation in a Shared VPC scenario:

- *Use of a Shared VPC network is not mandatory.* For example, instance admins can create instances in the service project that use a VPC network in that project. Networks defined in service projects are not shared.
- *Some resources must be re-created in order to use a Shared VPC network.* When a Shared VPC Admin attaches an existing project to a host project (/vpc/docs/provisioning-shared-vpc#create-shared), that project becomes a service project, but its existing resources do not automatically use shared network resources. To use a Shared VPC network, a Service Project Admin has to create an eligible resource and configure it to use a subnet of a Shared VPC network. For example, an existing instance in a service project cannot be reconfigured to use a Shared VPC network, but a new instance can be created to use available subnets in a Shared VPC network. This limitation applies to private zones.

Deployment Manager can only manage resources within a single project, so using it in a Shared VPC scenario requires a specific setup. See Creating a Shared VPC with Deployment Manager (/solutions/shared-vpc-with-deployment-manager) for more information.

Instances in service projects attached to a host project using the same Shared VPC network can communicate with one another using either *ephemeral* or reserved *static* internal IP addresses, subject to applicable firewall rules (/vpc/docs/firewalls).

An ephemeral internal IP address can be automatically assigned to an instance in a service project. For example, when Service Project Admins create instances (/vpc/docs/provisioning-shared-vpc#creating_an_instance_in_a_shared_subnet), they select a zone, the Shared VPC network, and an available subnet (/vpc/docs/provisioning-shared-vpc#discovering_subnets_that_can_be_used). The IP address comes from the range of available IP addresses in the selected shared subnet.

Alternatively, a Service Project Admin can choose to reserve a static internal IP address. The IP address object itself must be created in the same service project as the instance that will use it, even though the value of the IP address will come from the range of available IPs in a selected shared subnet of the Shared VPC network. Refer to reserving a static internal IP (/vpc/docs/provisioning-shared-vpc#reserve_internal_ip) on the *Provisioning Shared VPC* page for more information.

External IP addresses defined in the host project are only usable by resources in that project. They are not available for use in service projects. Service projects can maintain their own set of external IP addresses. For example, an instance in a service project must use an external IP address defined in that same service project. This is the case even if the instance uses an internal IP address from a Shared VPC network in the host project.

VMs in the same service project can reach each other using the internal DNS names that Google Cloud creates automatically. These DNS names use the project ID of the *service project* where the VMs are created, even though the names point to internal IP addresses in the host project. For a complete explanation, see Internal DNS names and Shared VPC (/compute/docs/internal-dns#shared-vpc) in the *Internal DNS* documentation.

You can use Cloud DNS private zones in a Shared VPC network. You must create your private zone in the host project and authorize access to the zone for the Shared VPC network.

Shared VPC can be used in conjunction with load balancing (/load-balancing/docs/load-balancing-overview). All load balancing components must exist *in the same project*, either all in a host project or all in a service project. Creating some load balancer components in a host project and others in an attached service project is **not** supported. However, when you create the internal forwarding rule for an internal TCP/UDP load balancer in a service project, you reference a shared subnet in the host project to which the service project is attached. Refer to creating an internal TCP/UDP load balancer (/vpc/docs/provisioning-shared-vpc#creating_an_internal_load_balancer_forwarding_rule) on the *Provisioning Shared VPC* page for more information.

The following table summarizes components for HTTP(S), SSL Proxy, and TCP Proxy Load Balancing. In most cases, you create the backend instances in a service project. In this case, all load balancer components are created in that project. It's also possible to create the backend instances in the host project; in that case, all load balancer components would need to be in the host project.

Load balancer	IP Address	Forwarding Rule	Other Frontend Components
HTTP(S) Load Balancing (/load-balancing/docs/https/)	An external IP address (/compute/docs/ip-addresses/reserve-static-external-ip-address#reserve_new_static) must be defined in the same project as the instances being load balanced (the service project).	The external forwarding rule (/load-balancing/docs/forwarding-rule-concepts/#external_fr) must be defined in the same project as the backend instances(the service project).	The target HTTP proxy or target H proxy (/load-balancing/docs/target-pro and associated URL map (/load-balancing/docs/url-map) r defined in the same project as the backend instances.
SSL Proxy Load Balancing (/load-balancing/docs/ssl/)			The target SSL proxy (/sdk/gcloud/reference/compute ssl-proxies/) must be defined in the same proje the backend instances.
TCP Proxy Load Balancing (/load-balancing/docs/tcp/)			The target TCP proxy (/sdk/gcloud/reference/compute tcp-proxies/) must be defined in the same proje the backend instances.

The following table summaries components for Network Load Balancing (/load-balancing/docs/network/):

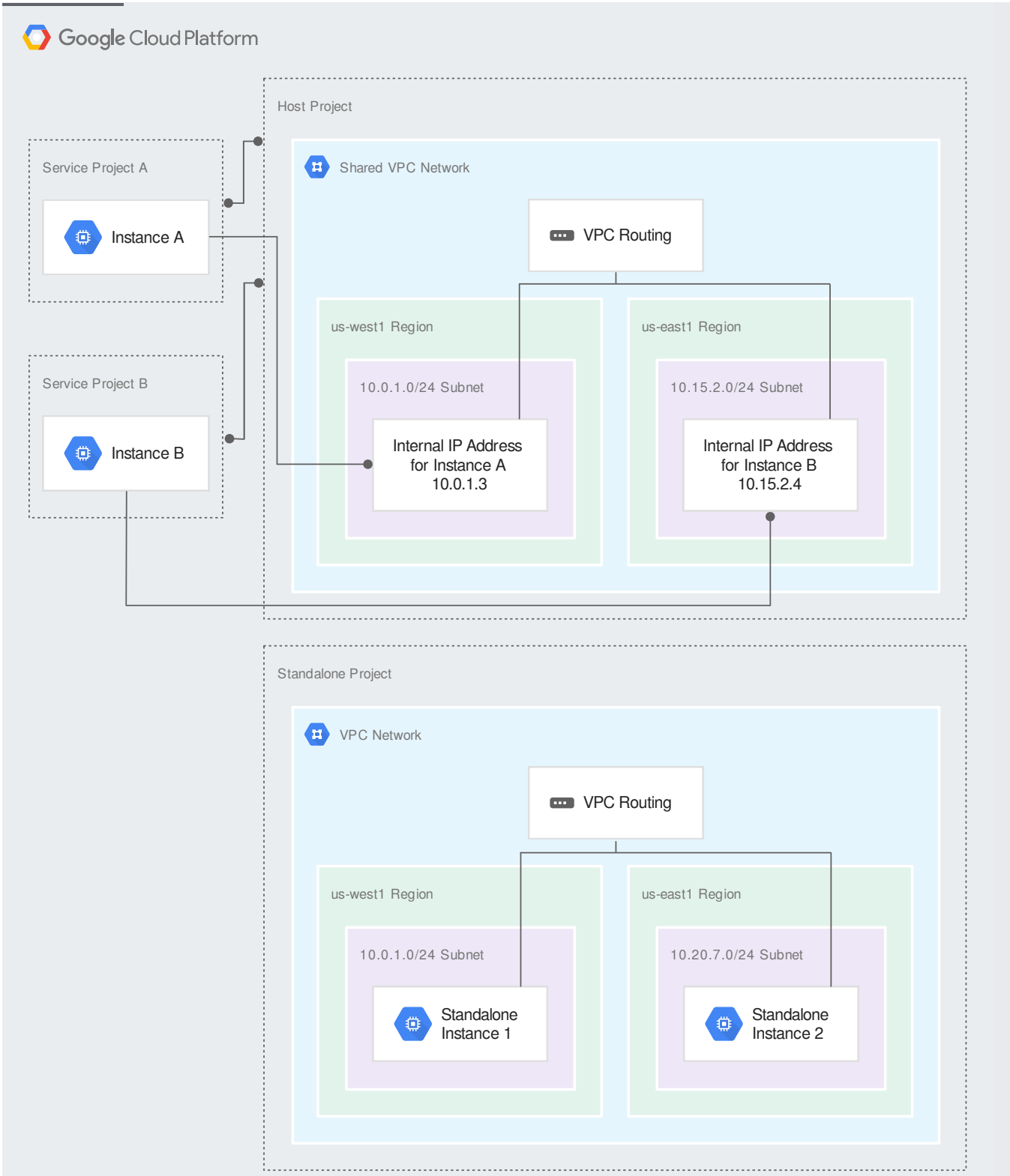
IP Address	Forwarding Rule	Backend Components
A regional external IP address (/compute/docs/ip-addresses/reserve-static-external-ip-address#reserve_new_static) must be defined in the same project as the instances being load balanced.	A regional external forwarding rule (/sdk/gcloud/reference/compute/forwarding-rules/) must be defined in the same project as the instances in the target pool (the service project).	The target pool (/load-balancing/docs/network/#target_pools must be defined in the same project and same region where the instances in the target pool exist. Health checks associated with the target pool must be defined in the same project as well.

The following table summaries components for Internal TCP/UDP Load Balancing (/load-balancing/docs/internal/). See creating an internal TCP/UDP load balancer

(/vpc/docs/provisioning-shared-vpc#creating_an_internal_load_balancer_forwarding_rule) on the *Provisioning Shared VPC* page for an example.

IP Address	Forwarding Rule	Backend Compor
An internal IP address (/load-balancing/docs/internal/#load_balancing_ip_address) **object** must be defined in the same project as the VMs being load balanced. For the load balancer to be available in a Shared VPC network, the Google Cloud internal IP address object **must** be defined in the same service project where the backend VMs are located, **and** it must reference a subnet in the desired Shared VPC network in the host project. The address itself comes from the primary IP range of the referenced subnet. If you create an internal IP address object in a service project that references a subnet in a VPC network in the service project itself, your internal TCP/UDP load balancer will be local to **that** network, **not** any Shared VPC network.	An internal forwarding rule (/load-balancing/docs/internal/#forwarding_rule) must be defined in the same project as the VMs being load balanced. For the load balancer to be available in a Shared VPC network, the internal forwarding rule **must** be defined in the same service project where the backend VMs are located, **and** it must reference the same subnet (in the Shared VPC network) that the associated internal IP address references. If you create an internal forwarding rule in a service project that references a subnet in a VPC network in the service project itself, your internal TCP/UDP load balancer will be local to **that** network, **not** any Shared VPC network.	In a Shared VPC s VMs are located i A regional interna (/load-balancing/docs/i service) and health check that service proje

The following example illustrates a simple Shared VPC scenario:



(/vpc/images/shared-vpc/shared-vpc-example-concepts.svg)

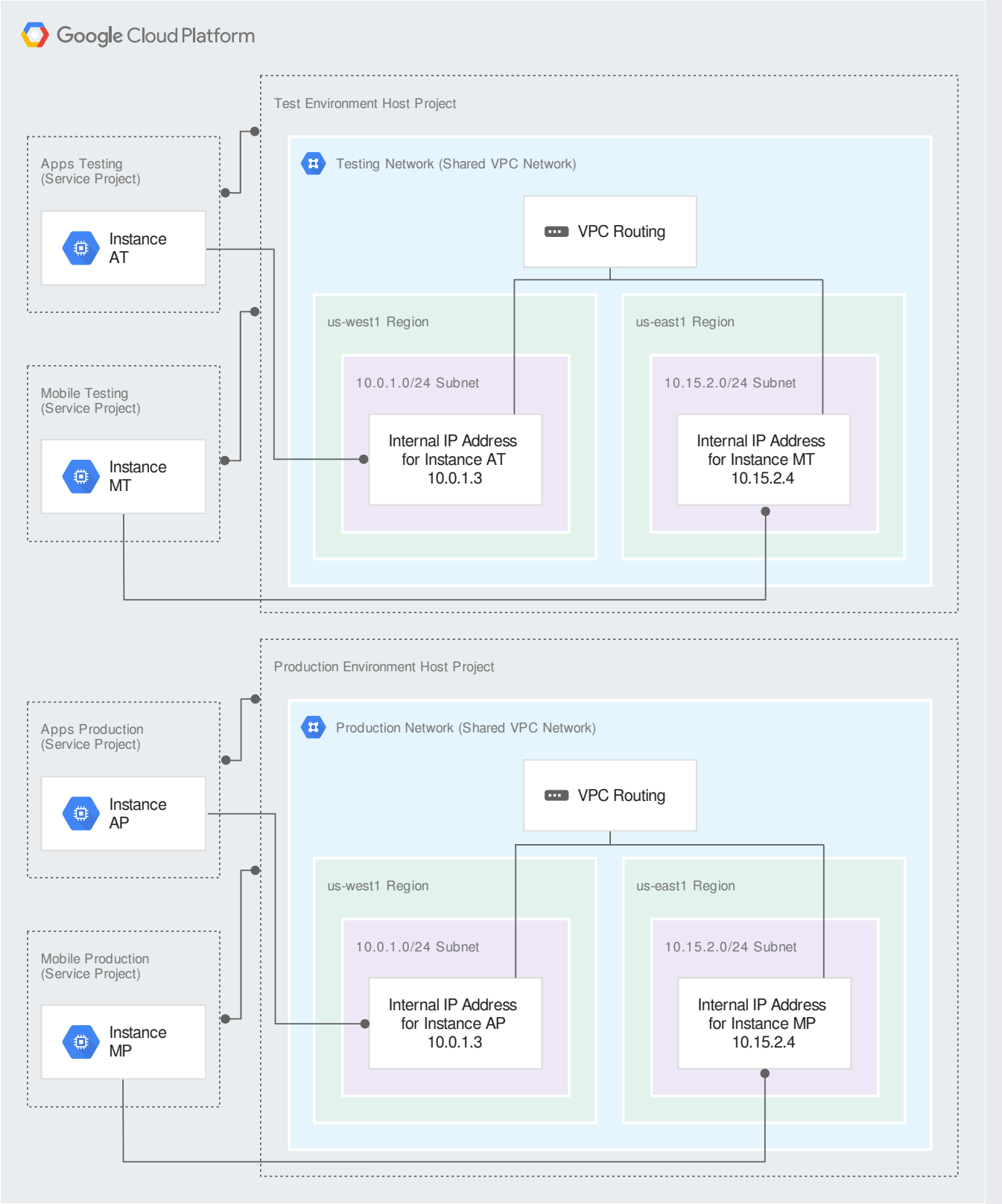
Basic concepts (click to enlarge)

- A Shared VPC Admin for the organization has created a host project and attached two service projects to it:

- Service Project Admins in `Service Project A` can be configured to access all or some of the subnets in the Shared VPC network. A Service Project Admin with at least subnet-level permissions to the `10.0.1.0/24 Subnet` has created `Instance A` in a zone of the `us-west1` region. This instance receives its internal IP address, `10.0.1.3`, from the `10.0.1.0/24` CIDR block.
- Service Project Admins in `Service Project B` can be configured to access all or some of the subnets in the Shared VPC network. A Service Project Admin with at least subnet-level permissions to the `10.15.2.0/24 Subnet` has created `Instance B` in a zone of the `us-east1` region. This instance receives its internal IP address, `10.15.2.4`, from the `10.15.2.0/24` CIDR block.

- The *Standalone Project* does not participate in the Shared VPC at all; it is neither a host nor a service project. Standalone instances are created by IAM members who have at least the `compute.InstanceAdmin` role for the project.

The following example demonstrates how Shared VPC can be used to build separate testing and production environments. For this case, an organization has decided to use two separate host projects, a *Test Environment* and a *Production Environment*.

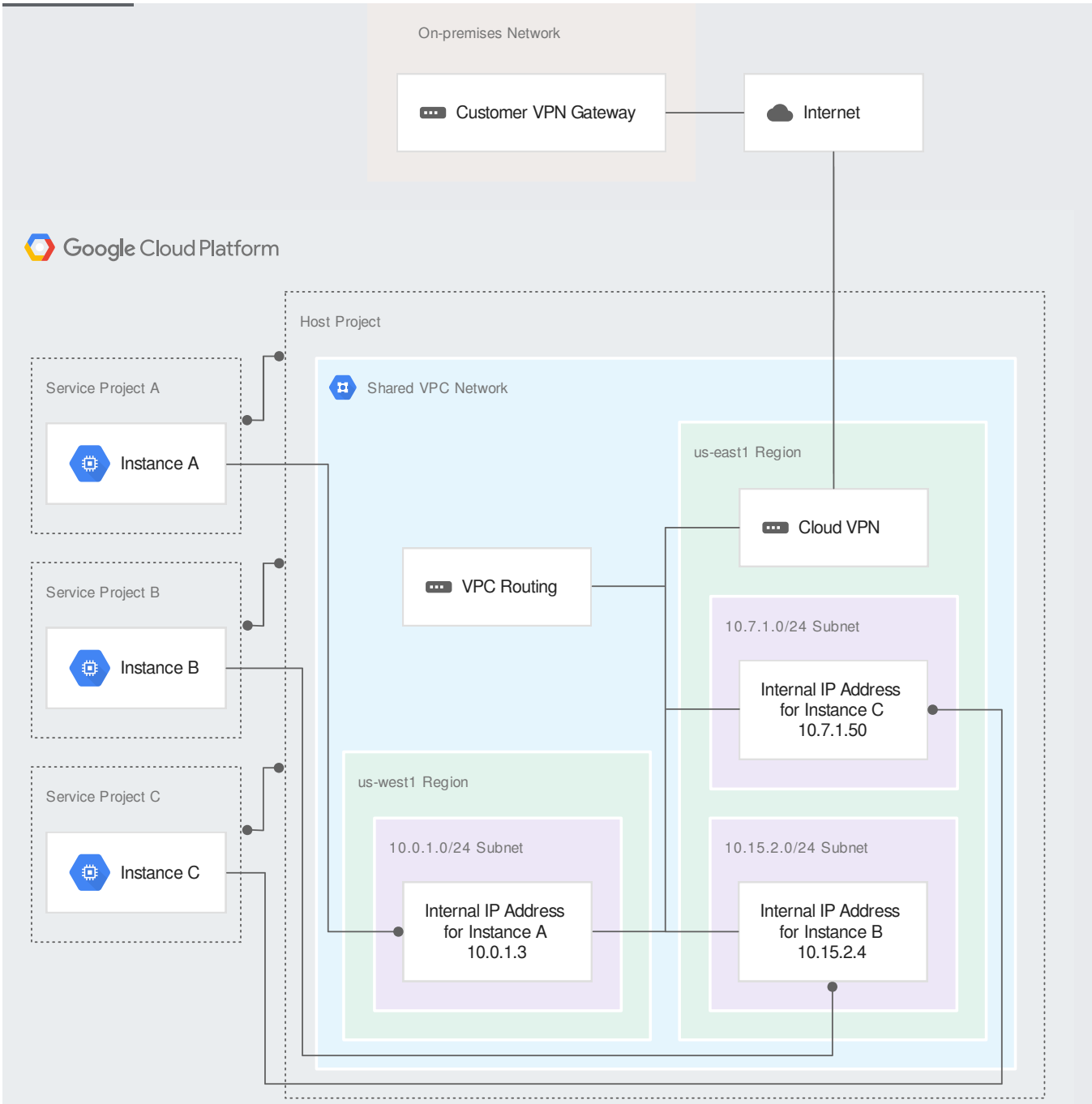


(/vpc/images/shared-vpc/shared-vpc-example-multiple-host-projects.svg)

Multiple host projects (click to enlarge)

- A Shared VPC Admin for the organization has created two host projects and attached two service projects to them as follows:
  - The `Apps Testing` and `Mobile Testing` service projects are attached to the `Test Environment` host project. Service Project Admins in each may be configured to access all or some of the subnets in the `Testing Network`.
  - The `Apps Production` and `Mobile Production` service projects are attached to the `Production Environment` host project. Service Project Admins in each may be configured to access all or some of the subnets in the `Production Network`.
- Both host projects have one Shared VPC network with subnets configured to use the same CIDR ranges. In both the `Testing Network` and `Production Network`, the two subnets are:
  - `10.0.1.0/24 Subnet` in the `us-west1` region
  - `10.15.2.0/24 Subnet` in the `us-east1` region
- Consider `Instance AT` in the `Apps Testing` service project and `Instance AP` in the `Apps Production` service project:
  - Service Project Admins can create instances like them provided they have at least subnet-level permissions to the `10.0.1.0/24 Subnet`.
  - Notice that both instances use the IP address `10.0.1.3`. This is acceptable because each instance exists in a service project attached to a unique host project containing its own Shared VPC network. Both the testing and production networks have been purposefully configured in the same way.
  - Instances using the `10.0.1.0/24 Subnet` must be located in a zone in the same region as the subnet, even though the subnet and instances are defined in separate projects. Because the `10.0.1.0/24 Subnet` is located in the `us-west1` region, Service Project Admins who create instances using that subnet must choose a zone in the same region, such as `us-west1-a`.

The following example demonstrates how Shared VPC can be used in a hybrid environment.



(/vpc/images/shared-vpc/shared-vpc-example-hybrid-cloud.svg)
Hybrid cloud (click to enlarge)

For this example, an organization has a created a single host project with a single Shared VPC network. The Shared VPC network is connected via Cloud VPN (/vpn/docs/concepts/overview) to an on-premises network. Some services and applications are hosted in Google Cloud while others are kept on-premises:

- A Shared VPC Admin enabled the host project and connected three service projects to it: `Service Project A`, `Service Project B`, and `Service Project C`.

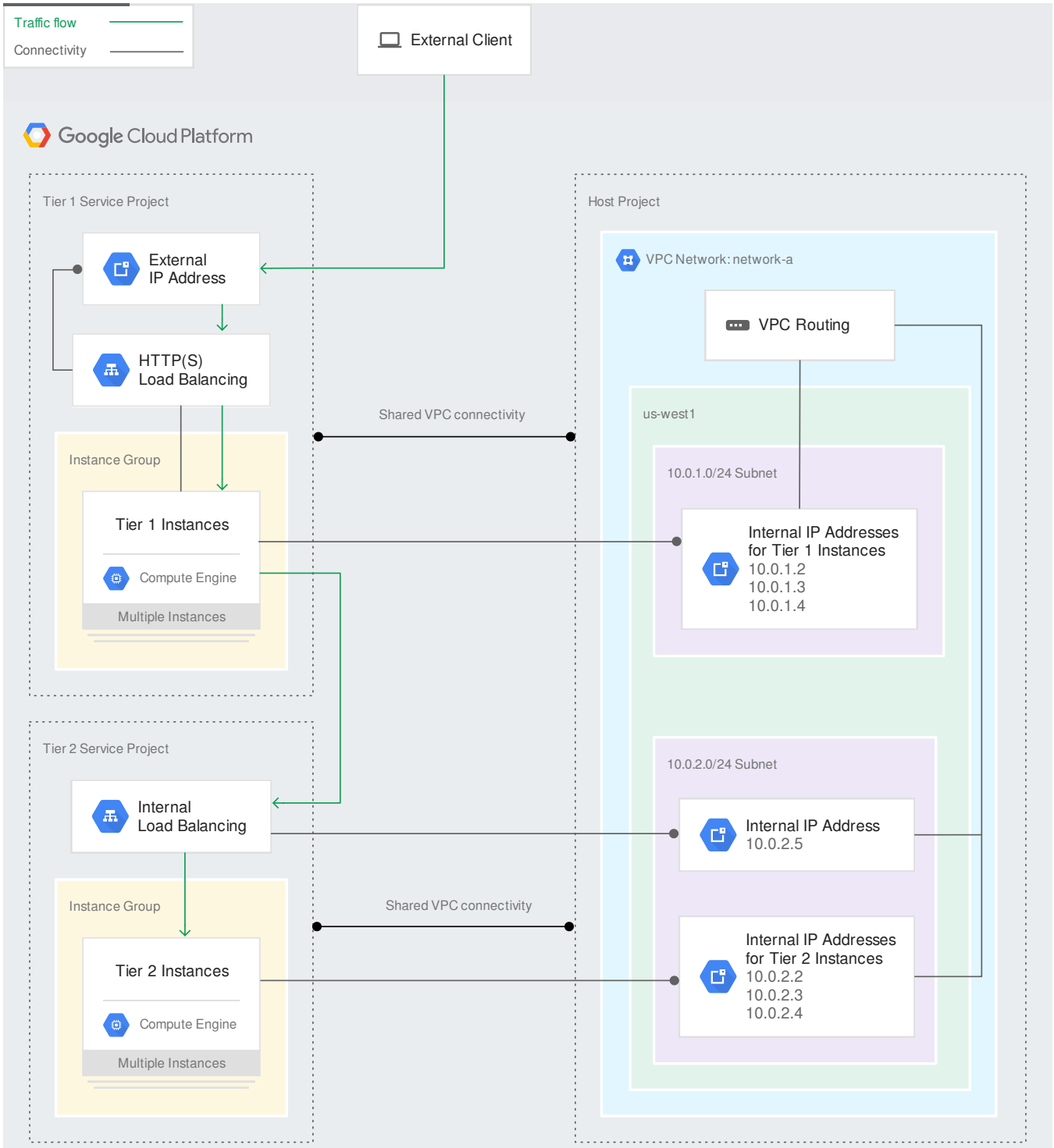
- Separate teams can manage each of the service projects. IAM permissions have been configured such that a Service Project Admin for one project has no permissions to the other service projects.
- The Shared VPC Admin has granted subnet-level or project-level permissions to the necessary Service Project Admins so they can create instances (/vpc/docs/provisioning-shared-vpc#creating_an_instance_in_a_shared_subnet) that use the Shared VPC network:
  - A Service Project Admin for `Service Project A` who has subnet-level permissions to the `10.0.1.0/24 Subnet` can create `Instance A` in it. The Service Project Admin has to choose a zone in the `us-west1` region for the instance, because that is the region that contains the `10.0.1.0/24 Subnet`. `Instance A` receives its IP address, `10.0.1.3`, from the range of free IP addresses in `10.0.1.0/24 Subnet`.
  - A Service Project Admin for `Service Project B` who has subnet-level permissions to the `10.15.2.0/24 Subnet` can create `Instance B` in it. The Service Project Admin has to choose a zone in the `us-east1` region for the instance, because that is the region that contains the `10.15.2.0/24 Subnet`. `Instance B` receives its IP address, `10.15.2.4`, from the range of free IP addresses in `10.15.2.0/24 Subnet`.
  - A Service Project Admin for `Service Project C` who has project-level permissions to the whole host project can create instances in any of the subnets in any of the VPC networks in the host project. For example, the Service Project Admin can create `Instance C` in the `10.7.1.0/24 Subnet`, choosing a zone in the `us-east1` region to match the region for the subnet. `Instance C` receives its IP address, `10.7.1.50`, from the range of free IP addresses in `10.7.1.0/24 Subnet`.
- Each service project is billed separately.
- The Service Project Admins in each project are responsible for creating and managing resources (/vpc/docs/provisioning-shared-vpc#create-resources).

- A Shared VPC Admin has delegated network administration tasks to other IAM members who are Network and Security Admins (#net_and_security_admins) for the Shared VPC network.
  - A Network Admin has created a Cloud VPN gateway and configured a VPN tunnel through the Internet to an on-premises gateway. The Cloud VPN exchanges and receives routes with its on-premises counterpart because a corresponding Cloud Router in the same `us-east1` region has been configured (/vpn/docs/how-to/creating-vpn-dynamic-routes).
  - If the dynamic routing mode (/vpc/docs/vpc#routing_for_hybrid_networks) of the VPC is global, the Cloud Router applies the learned routes to the on-premises network in all

subnets in the VPC network, and it shares routes to all of the VPC subnets with its on-premises counterpart.

- Security Admins create and manage firewall rules in the Shared VPC network to control traffic among instances in Google Cloud and the on-premises network.
- Subject to applicable firewall rules, instances in the service projects can be configured to communicate with internal services, such as database or directory servers located on-premises.

The following example demonstrates how Shared VPC can be employed to delegate administrative responsibilities and maintain the principle of least privilege. For this case, an organization has a web service that is separated into two tiers, and different teams manage each tier. Tier 1 represents the externally-facing component, behind an HTTP(S) load balancer (/load-balancing/docs/https/). Tier 2 represents an internal service upon which Tier 1 relies, and it is balanced using an internal TCP/UDP load balancer (/load-balancing/docs/internal/).



(/vpc/images/shared-vpc/shared-vpc-example-two-tier.svg)

Two-tier web service (click to enlarge)

Shared VPC allows you to map each tier of the web service to different projects so that they can be managed by different teams while sharing a common VPC network:

- Resources, such as instances and load balancer components, for each tier are placed in individual service projects managed by different teams.
- Each tier service project has been attached to the host project by a Shared VPC Admin. The Shared VPC Admin also enabled the host project.
    - Separate teams can manage each tier of the web service by virtue of being a Service Project Admin in the appropriate service project.
    - Each service project is billed separately.
    - The Service Project Admins in each project are responsible for creating and managing resources (/vpc/docs/provisioning-shared-vpc#create-resources).
- Network access control is delineated as follows:
    - IAM members who only work on Tier 1 are Service Project Admins for the `Tier 1 Service Project` and are granted subnet-level permissions for just the `10.0.1.0/24 Subnet`. In this example, one such Service Project Admin has created three `Tier 1 Instances` in that subnet.
    - IAM members who only work on Tier 2 are Service Project Admins for the `Tier 2 Service Project` and are granted subnet-level permissions for just the `10.0.2.0/24 Subnet`. In this example, another Service Project Admin has created three `Tier 2 Instances` in that subnet along with an internal load balancer whose forwarding rule uses an IP address from the range available in that subnet.
    - IAM members who oversee the whole web service are Service Project Admins in both service projects, and they have project-level permissions to the host project so they can use any subnet defined in it.
    - Optionally, a Shared VPC Admin can delegate network administration tasks to Network and Security Admins (#net_and_security_admins).

- To set up Shared VPC, see Provisioning Shared VPC (/vpc/docs/provisioning-shared-vpc).
- To set up Kubernetes Engine clusters with Shared VPC, see Setting up clusters with Shared VPC (/kubernetes-engine/docs/how-to/cluster-shared-vpc).
- To delete a Shared VPC setup, see Deprovisioning Shared VPC (/vpc/docs/deprovisioning-shared-vpc).