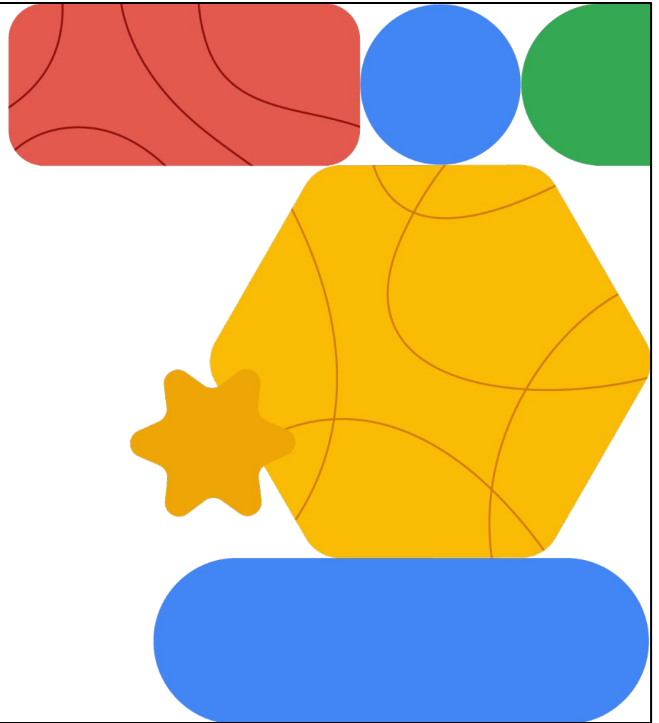


Networking in Google Cloud

Private Connection Options



Private connection options.



Today's agenda

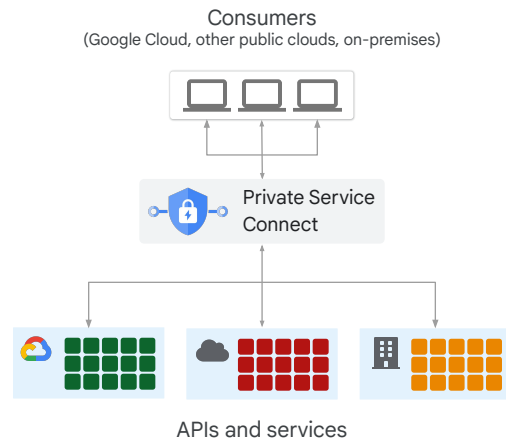


- 01 [Private access overview](#)
- 02 Private Google Access
- 03 Private Service Connect
- 04 Private services access
- 05 Cloud NAT
- 06 Lab: Implement Private Google Access with Cloud NAT
- 07 Quiz

In this module, we'll discuss some general methods of accessing Google and other services privately by using internal IP addresses. Then, we will cover each of the methods: Private Google Access, Private Service Connect, private services access, and Cloud NAT (Network Address Translation). After that, you will try what you learned in a Cloud NAT lab exercise.

Private access for Google APIs and services

- Private access uses internal IP addresses.
- *Private access* refers to the ability to connect to APIs and services locally.
- Access is quicker and more secure.
- Choose a private access option based on your needs.
- All Google Cloud APIs and services support private access.



Private access uses internal IP addresses.

Therefore, consumers connect to supported APIs and services with an internal connection. Unless a consumer connects to Google Cloud by using an external connection, private access communication does not go through the public internet.

Access is quicker and more secure.

Choose a private access option based on your needs.

All Google Cloud APIs and services support private access.

You can also set up private access to APIs and services that you publish. You can access these API and services from Google Cloud, other public clouds, or on-premises.

Google APIs and services have public URLs and are accessible on the public internet.

Private access options

| Option | Connection | Usage |
|-------------------------|---|---|
| Private Google Access | Connect to the public IP addresses of Google APIs and services through the default internet gateway of the VPC network. | Lets you use Google APIs and services without giving your Google Cloud resources external IP addresses. |
| Private Service Connect | Connect to Google, third-party, or your own services by using internal IP addresses. | Lets you use internal IP addresses to consume, produce, and make services available. |
| Serverless VPC Access | Connect serverless products to your VPC network to access Google, third-party, or your own services with internal IP addresses. | Lets Cloud Run, App Engine standard, and Cloud Functions connect to the internal IPv4 addresses in a VPC network. |
| Private services access | Connect Google and third-party services privately and directly to your VPC network with VPC Network Peering. | Lets you use internal IP addresses to connect to specific Google and third-party services by using VPC Network Peering. |

Google provides several private access options. Each option allows VM instances with internal IP addresses to reach certain APIs and services. You can configure one or all of these options, because they operate independently of each other.


Private Google Access for on-premises hosts lets your on-premises hosts connect Google APIs and services through the default internet gateway of the VPC network. Your on-premises hosts don't need external IP addresses; instead, they use internal IP addresses.

Private Service Connect lets you connect to a Google or third-party managed VPC network through a service attachment. As with Private Google Access, the connection is internal.


Serverless VPC Access connects serverless products to your VPC network to access Google, third-party, or your own services with internal IP addresses. For example, Cloud Run, App Engine standard, and Cloud Functions environments send packets to the internal IPv4 address of the resource. Serverless VPC Access is not covered in this module. For more information, refer to [Connect from serverless Google services to VPC networks](#) in the Google Cloud documentation.

Private services access is a private connection between your VPC network and a service producer VPC network. This connection is implemented as a VPC Network Peering connection. The service producer network is created exclusively for you, and it's not shared with other customers.

For more information, see [Private access options for services](#) in the Google Cloud documentation.



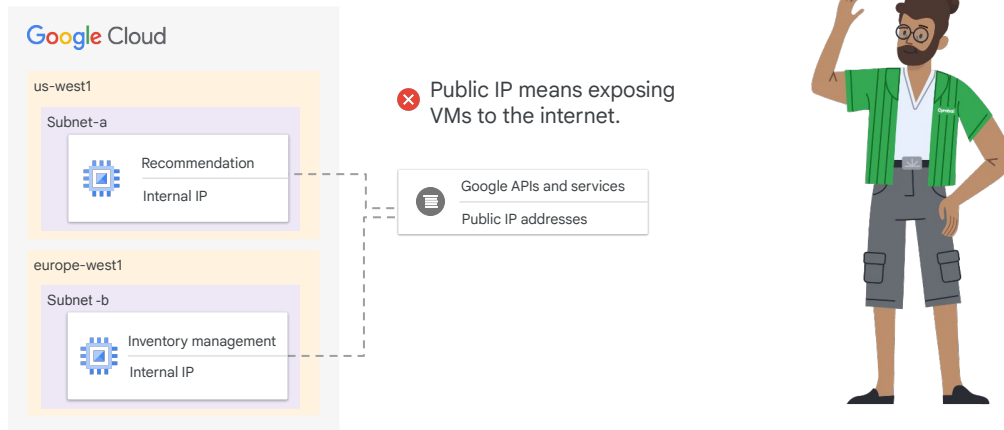
Today's agenda



- | | |
|----|---|
| 01 | Private access overview |
| 02 | Private Google Access |
| 03 | Private Service Connect |
| 04 | Private services access |
| 05 | Cloud NAT |
| 06 | Lab: Implement Private Google Access with Cloud NAT |
| 07 | Quiz |

Next, let's discuss how to use Private Google Access to connect to Google APIs and services over an internal connection.

Use case: Securely connect to Google Cloud API



Joe, Cymbal Corporation's network engineer, faces a challenge. Cymbal runs a recommendation engine on Google Cloud VMs that analyzes customer data to provide tailored product suggestions. This data is highly sensitive, and they need to protect it from potential internet-based attacks and comply with strict data privacy regulations. They also run a large fleet of internal systems (e.g., inventory management, point-of-sale systems). These need regular updates but don't require direct internet access for their core functionality.

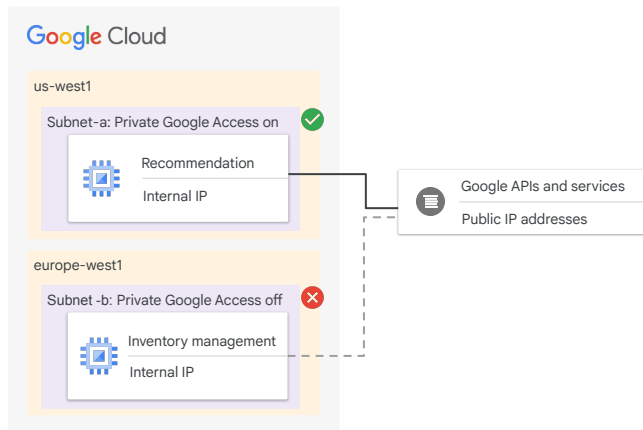
Challenge

The challenge is providing the recommendation engine access to Google Cloud API such as BigQuery for data analysis and Cloud Storage for model storage would traditionally require Public IPs or a NAT set up.

- Using public IPs means exposing the VMs to the internet, increasing the attack surface.
- Using complex NAT setups involves adding management overhead and potential bottlenecks.

Joe wonders if there is a way for Google Cloud VMs and fleet of VMs to securely connect to Google Cloud API without exposing IPs or creating complex NAT.

Private Google Access is enabled at a subnet level



- Securely connecting VMs without external IP addresses to essential Google APIs and services.
- Private Google Access is enabled on a subnet-by-subnet basis.
- If you disable Private Google Access for a subnet, VMs with internal IP addresses can only send traffic within the VPC network.
- Private Google Access has no effect on VMs with external IP addresses.

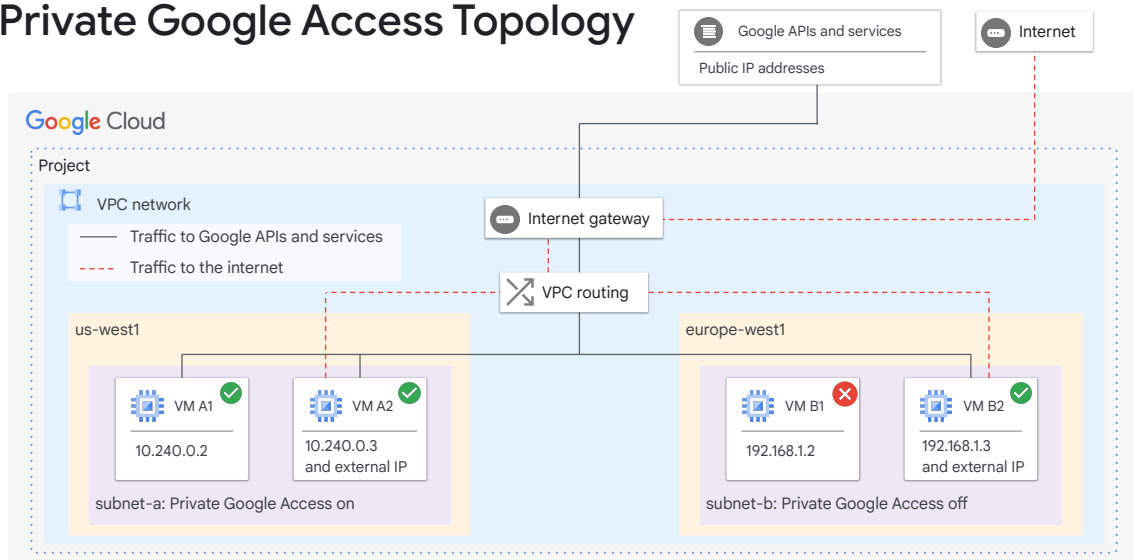
Private Google Access enables securely connecting VMs without external IP addresses to essential Google APIs and services. You enable the Private Google Access feature on a subnet-by-subnet basis by editing the subnet in Google Cloud console or the Google Cloud CLI.

If you disable Private Google Access for a subnet, VMs with internal IP addresses can only send traffic within the VPC network. Later in this module, you will learn about Cloud NAT, which can allow these VMs to send traffic outside of the VPC network.

Private Google Access has no effect on instances that have external IP addresses.

For a list of the services that are supported by Private Google Access, see [Private access options for services](#) in the Google Cloud documentation.

Private Google Access Topology



In the sample topology, the VPC network has two subnets: subnet-a and subnet-b. The network has been configured to meet the Domain Name System (DNS), routing, and firewall network requirements for Google APIs and services.

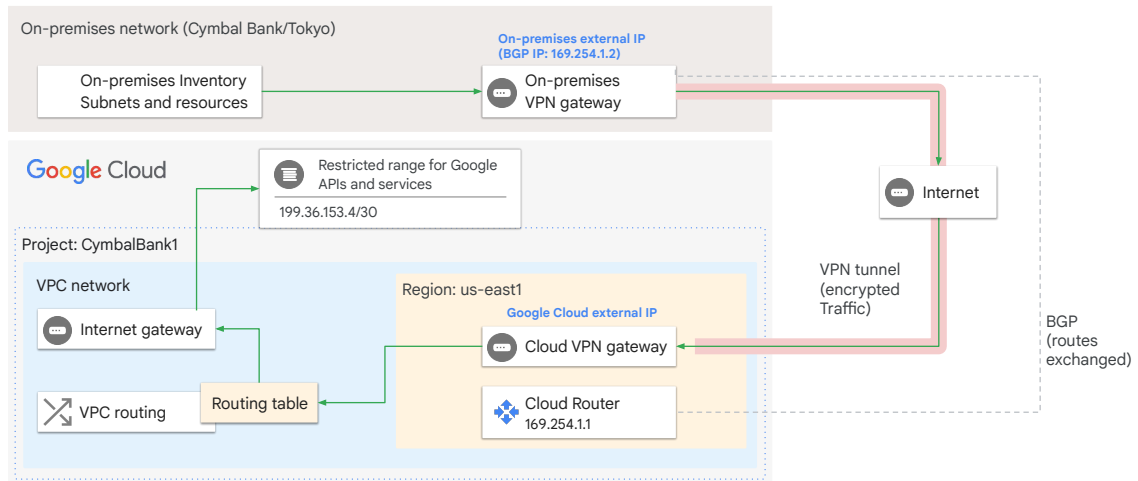
Private Google Access has been enabled on subnet-a, but not on subnet-b.

VM A1 can access Google APIs and services, including Cloud Storage, because its network interface is located in subnet-a, which has Private Google Access enabled. Private Google Access applies to the instance because it only has an internal IP address.

VM B1 can't access Google APIs and services because it only has an internal IP address and Private Google Access is disabled for subnet-b.

VM A2 and VM B2 can both access Google APIs and services, including Cloud Storage, because each of them has its own IP address. Private Google Access has no effect on whether these instances can access Google APIs and services because both have external IP addresses.

Private Google Access for on-premises hosts



Cymbal Bank is expanding and wants to use internal IP addresses to access Cloud SQL and Cloud TPU from their Tokyo on-premises network. This example shows how this access can be achieved. In the example, the on-premises network is connected to a VPC network through a Cloud VPN tunnel. Traffic from on-premises hosts to Google APIs travels through the tunnel to the VPC network. After traffic reaches the VPC network, it's sent through a route that uses the default internet gateway as its next hop. This next hop allows traffic to leave the VPC network and be delivered to `restricted.googleapis.com` (199.36.153.4/30).

The on-premises DNS configuration maps `*.googleapis.com` requests to `restricted.googleapis.com`, which resolves to the 199.36.153.4/30 address range.

In this example, Cloud Router uses a custom route advertisement for this IP address range. This route sends traffic through the Cloud VPN tunnel. The traffic that goes to Google APIs is routed through the tunnel to the VPC network.

A custom static route was added to the VPC network. This route directs traffic with the destination 199.36.153.4/30 to the default internet gateway as the next hop. Google then directs traffic to the appropriate API or service.

In this example, network administrators at Cymbal Bank created a Cloud DNS managed private zone for `*.googleapis.com` that maps to the 199.36.153.4/30 address range. The network administrators authorized the VPC network to use that zone. Requests to the `googleapis.com` domain are sent to the IP addresses that are

used by `restricted.googleapis.com`. Only the supported APIs are accessible with this configuration, which might cause other services to be unreachable. Cloud DNS doesn't support partial overrides. If you require partial overrides, use BIND (a software that interacts with DNS).

Caveats: Private Google Access

- ! Legacy networks are not supported because they don't support subnets.
- ! You must enable the Google APIs to use them
- ! Your VPC network must have appropriate routes and egress firewalls defined.
- ! If you use the `private.googleapis.com` or the `restricted.googleapis.com` domain names, you must create DNS records for them.



Private Google Access has a few caveats.

Because Private Google Access is enabled on a per-subnet basis, you must use a VPC network. Legacy networks are not supported, because they don't support subnets.

Enable the Google APIs that you want to use. You enable these desired APIs on the APIs & services page in the Google Cloud console.

Your VPC network must have appropriate routes and egress firewalls defined. This network must also have appropriate routes for the destination IP ranges that are used by Google APIs and services.

If you use the `private.googleapis.com` or the `restricted.googleapis.com` domain names, you must create DNS records to direct traffic to the IP addresses that are associated with those domains. For more information, see [Network configuration](#) on the Configure Private Google Access page of the Google Cloud documentation. These domain names only offer IPv4 connectivity.

Caveats: Private Google Access that uses IPv6

If you want to use IPv6 to connect to Google APIs and services:


- ! Your VM must be configured with a /96 IPv6 address range.
- ! The software running on the VM must send packets whose sources match one of those IPv6 addresses from that range.
- ! You must send the packets to the IPv6 addresses for the default domains.




If you want to use IPv6 to connect to Google APIs and services:

Your VM must be configured with a /96 IPv6 address range. The software running on the VM must send packets whose sources match one of those IPv6 addresses from that range.

You must send the packets to the IPv6 addresses for the default domains.



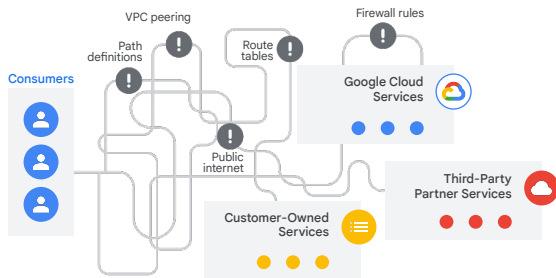
Today's agenda



- 01 Private access overview
- 02 Private Google Access
- 03 [Private Service Connect](#)
- 04 Private services access
- 05 Cloud NAT
- 06 Lab: Implement Private Google Access with Cloud NAT
- 07 Quiz

Next, let's discuss Private Service Connect, which lets you use internal IP addresses to consume, produce, and make services available.

Use case: Access managed services privately



Challenges without PSC

Security

Control

Flexibility

Without Private Service Connect

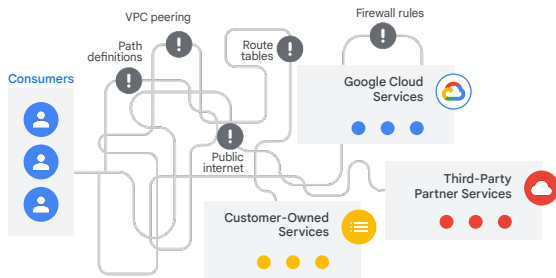
Nur is a network engineer at Cymbal. A large financial institution offers a real-time transaction processing API for its partners. This API handles sensitive financial data and needs to be:

- Highly secure with restricted access.
- Easily accessible to authorized partners over the internet.
- Scalable to handle fluctuating traffic loads.

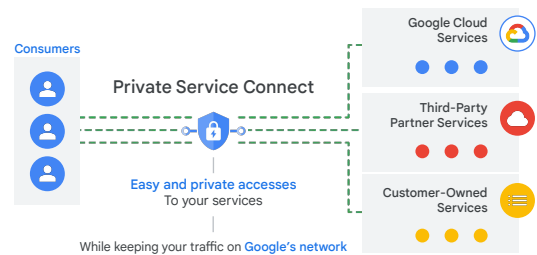
Challenges:

- Security: exposing the API directly to the public internet poses security risks.
- Control: traditional methods (IP whitelisting or VPNs) can be cumbersome for managing access and don't always scale well.
- Flexibility: the API should be consumable by partners who might be in different cloud environments or have their own on-premises infrastructure.

Use case: Access managed services privately



Without Private Service Connect



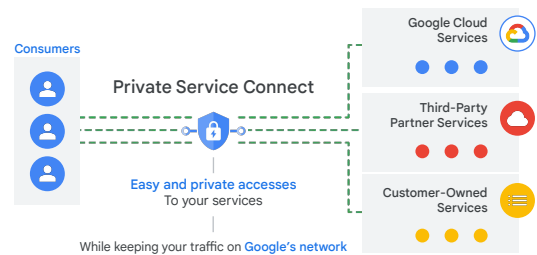
With Private Service Connect

Private Service Connect provides a secure, scalable, and flexible way to expose services to specific partners or networks. It's ideal for scenarios involving sensitive APIs or the need for custom IP addressing.

- **Private endpoint:** The financial institution creates a Private Service Connect endpoint attached to their API within their VPC. This endpoint gets a private IP address within their network.
- **Service publishing:** The service (transaction processing API) is "published", making it discoverable by authorized consumers.
- **Controlled access:** Partners create PSC consumer endpoints in their respective VPCs. These endpoints are assigned private IP addresses from a range the financial institution specifies, facilitating fine-grained control.
- **Secure, scalable consumption:** Authorized partners can now consume the API using the private IP address of the published service. Traffic flows through Google's network. Load balancing is handled on the service producer's side.

Private Service Connect

- ✓ With Private Google Access, Google APIs and services can be accessed with internal IP addresses.
- ✓ With Private Service Connect, third-party resources and intra-organization published services can be also accessed with internal IP addresses.
- ✓ You can access resources through a Private Service Connect endpoint or a backend.
- ✓ Private Service Connect is fast and scalable.



As with Private Google Access, you can use Private Service Connect to access Google APIs and services with a global internal IP address.

Private Service Connect also lets you access third-party services and services provided within the organization with an internal IP address.

To access resources with Private Service Connect, use a Private Service Connect endpoint or a backend. Organizations can choose the internal IP address to associate with each endpoint.

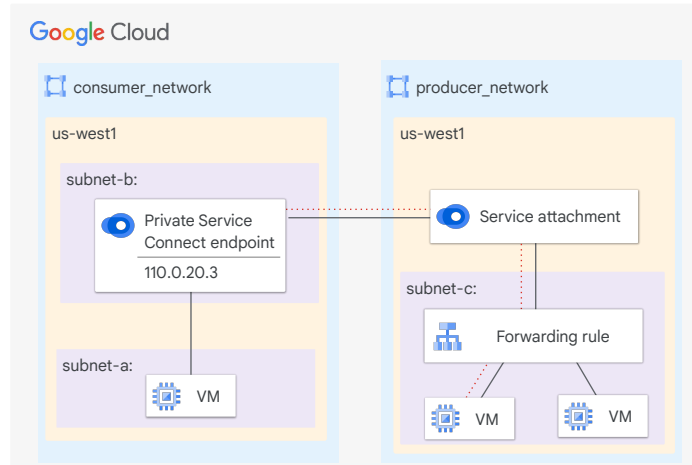
Private Service Connect has line rate performance and scales to enterprise-size networks. In other words, Private Service Connect is fast and grows with your organization.

Look at the example shown on the right. VM1 in the consumer_network uses a Private Service Connect endpoint to connect to services that run in the producer_network on VM2. The Private Service Connect endpoint has an internal IP address, 10.0.20.10. VM1 uses the internal address to access services on VM2.

Next, let's talk about other things you can do with Private Service Connect.

Using a forwarding rule

- ✓ The service attachment:
 - Receives requests redirected from the Private Service Connect endpoint.
 - Sends them to a forwarding rule.
- ✓ The forwarding rule sends the traffic to the correct VM or service.



The consumer contacts a producer service or VM by using the Private Service Connect endpoint in their VPC network. This endpoint has an internal IP address and maps to the service attachment in the producer VPC network.

A service attachment refers to services from a producer.

In this example, the service attachment receives requests redirected from the Private Service Connect endpoint and

sends it to a forwarding rule.

The forwarding rule sends the request to the appropriate VM or service. You can see this flow in the example, with the red, dotted line.

Refer to the [documentation](#) for more details.

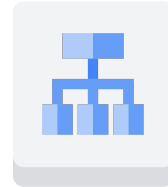
Use a load balancer to access PSC service

Private Service Connect backends let Google Cloud load balancers send traffic through Private Service Connect to reach published services or Google APIs.

Rename services and map them to URLs of your choice.

You can configure the load balancer to log all requests to Cloud Logging.

spanner.example.com
spanner.cymbal.com



Load balancer

Using a load balancer provides some additional features.

Private Service Connect backends let Google Cloud load balancers send traffic through Private Service Connect to reach published services or Google APIs.

You can assign DNS names to these internal IP addresses—or even Google APIs and services—with meaningful names for your organization. For example, if you have a service with the name `spanner.example.com`, you can map `spanner.cymbal.com` or some other name that makes sense for your organization. These names and IP addresses are internal to your VPC network. On-premises networks use Cloud VPN tunnels or VLAN attachments to connect to it.

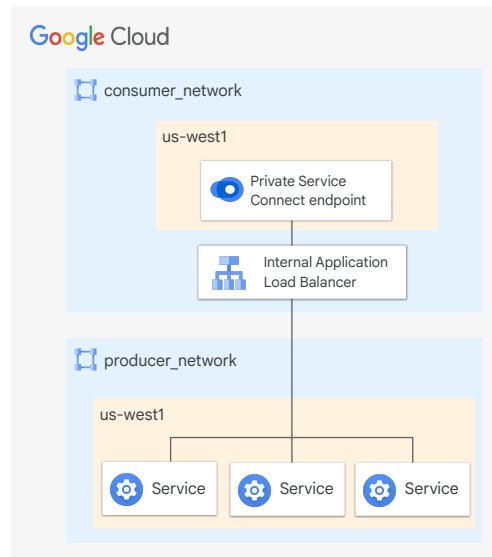
Also, you can control which traffic goes to which endpoint and demonstrate that the traffic stays within Google Cloud.

You can configure the load balancer to log all requests to Cloud Logging.

Using an internal Application Load Balancer

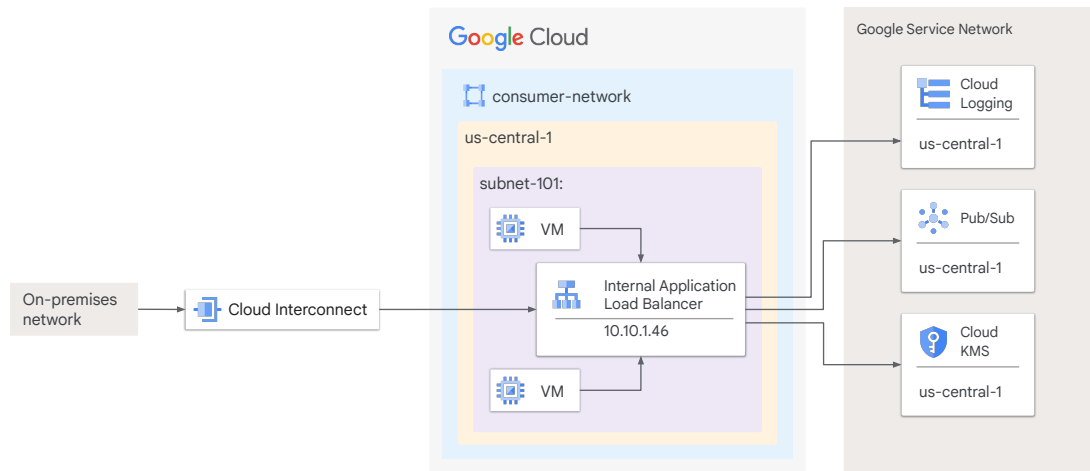
With Private Service Connect and an Application Load Balancer, you can:

- Use a URL map to evaluate requests and route them to the correct VM or service.
- Use customer-managed TLS certificates.
- Enable data residency in-transit by connecting to regional endpoints for Google APIs from workloads in that same region.



You can also enable data residency in transit by connecting to regional endpoints for Google APIs from workloads in that same region. In other words, you can be certain that data at rest is stored in the region you configure.

Sample topology

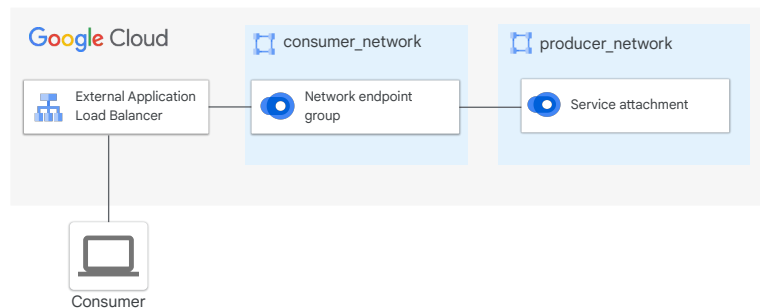


This example shows an on-premises network that is connected to a subnet of a Google Cloud VPC network in the us-central-1 region. The requests that a VM or service in the on-premises network make to Cloud Logging, Pub/Sub, or Cloud KMS are not routed on the public internet. They remain within the Google Cloud backbone network. Requests to other Google Cloud services are routed over the public internet.

The internal Application Load Balancer in the consumer-network and the desired Google services are both located in the us-central-1 zone. You can see that, after the request from the on-premises network is sent through the Cloud Interconnect connection to the load balancer, it remains in the us-central-1 zone. If desired, access through the load balancer can be sent to Cloud Logging.

Using a global external Application Load Balancer

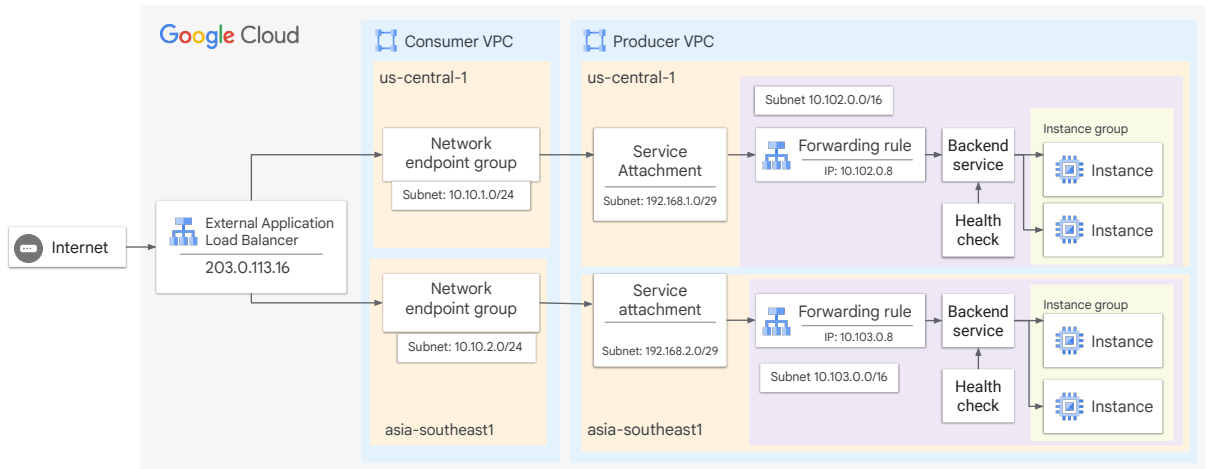
- ✓ Consumers connect to an external IP address.
- ✓ Private Service Connect uses a network endpoint group to route the request to the service producer.



With Private Service Connect and consumer HTTP(S) service controls that use a global external Application Load Balancer, consumers connect to an external IP address. Private Service Connect uses a network endpoint group to route the request to the service producer.

Let's look at a more detailed example of Private Service Connect that uses a global external Application Load Balancer.

Sample topology



This topology shows a Private Service Connect endpoint based on a global external Application Load Balancer. This topology lets service consumers with internet access connect to the load balancer. The load balancer then directs the requests to the appropriate network endpoint group in a consumer network. Each of these endpoints is associated with a service attachment. A forwarding rule then routes the request to the appropriate VM instance or service.

General benefits



Except for the global external Application Load Balancer use case, connections use internal IP addresses.



Traffic stays on the Google backbone network.



Configuration is simple.



Private Service Connect

Private Service Connect lets you configure access to specific Google and third-party services using internal IP addresses.

Except for the global external Application Load Balancer use case, connections use internal IP addresses. Traffic stays on the Google backbone network. Connections are thus more secure and much faster than over the public internet. Services that use Private Service Connect interact like services on a private network.

Configuration is simple. Private Service Connect works with the internal IP address range that you provide and sets up the routing tables.

Benefits for consumers

Consumers:

- ✓ Can control the internal IP address that is used to connect to a managed service.
- ✓ Do not need to reserve internal IP address ranges for backend services that are consumed in their VPC network.
- ✓ Must initiate traffic to the service provider, which improves security.



Private Service Connect is highly scalable and supports thousands of consumers. Consumers use consumer VPC networks to access VMs and services from producer VPC networks.

Consumers can control the internal IP address that is used to connect to a managed service.

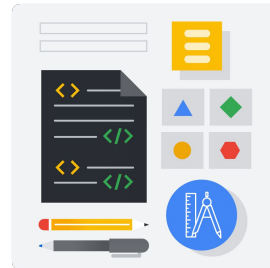
They don't need to reserve internal IP address ranges for backend services that are consumed in their VPC network. Instead, consumers choose an IP address from their own subnet to connect to the producer services.

For security purposes, all communications between the consumer VPC network and service producer VPC network must be initiated by the consumer. Service producers can't initiate this communication. This unidirectional connectivity drastically simplifies firewall configuration, but also reduces risk from rogue traffic originating from the service producer.

Benefits for producers

Producers:

- ✓ Can choose to deploy a multi-tenant model, serving multiple consumer VPC networks.
- ✓ Can scale services to as many VM instances as required without asking consumers for more IP addresses.
- ✓ Don't need to change firewall rules based on the subnet ranges in the consumer VPC networks.



Service producers make VMs and services available to consumers.

Producers can choose to deploy a multi-tenant model, where your VPC network contains services that are used by multiple consumer VPCs. The consumer networks can have overlapping subnet ranges.

Service producers can scale services to as many VM instances as required, without asking consumers for more IP addresses.

Service producers don't need to change firewall rules based on the subnet ranges in the consumer VPC networks. You can simply create firewall rules for the network address translation (NAT) IP address range configured for your service.

Private Service Connect interfaces

A Private Service Connect interface is a special type of network interface that refers to a network attachment.

A Private Service Connect Interface enables services in a producer VPC network to securely reach resources and destinations within a consumer VPC network.

Producer and consumer networks can be in different projects and organizations.



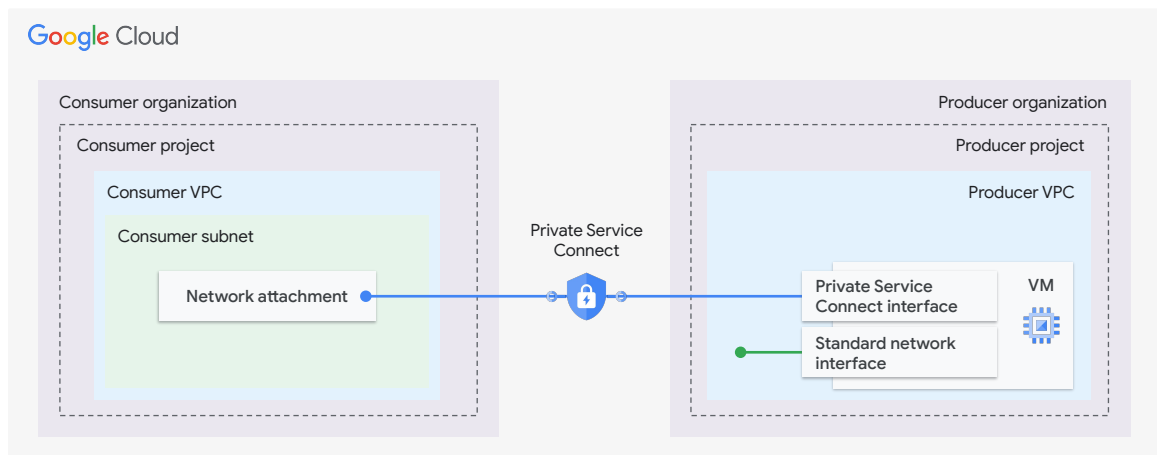
Private Service
Connect

A Private Service Connect interface is a special type of network interface that refers to a network attachment.

A Private Service Connect Interface enables services in a producer VPC network to securely reach resources and destinations within a consumer VPC network.

Producer and consumer networks can be in different projects and organizations.

Private Service Connect interfaces



If the service consumer accepts the connection, Google Cloud allocates the interface an IP address from a subnet in the consumer VPC network that's specified by the network attachment. The VM of the Private Service Connect interface has a second standard network interface that connects to the producer's VPC network.

A connection between a Private Service Connect interface and a network attachment is similar to the connection between a Private Service Connect endpoint and a service attachment, but it has two key differences:

- A Private Service Connect interface lets a producer network initiate connections to a consumer network (managed service egress), while an endpoint lets a consumer network initiate connections to a producer network (managed service ingress).
- A Private Service Connect interface connection is transitive. This means that a producer network can communicate with other networks that are connected to the consumer network.

A common use case is when a managed service needs to securely access data within a customer's VPC network. Private Service Connect interface lets the service securely access the data, whether that data resides in the cloud, on-premises (via VPN or Cloud Interconnect), or with a third-party service. This maintains privacy and isolation for sensitive information.

There are many different types of Private Service Connect interface, so producers will provide documentation on how to use it with their services. For example, some

producers will use an API, where others might develop a user interface (UI).

Making Private Service Connect easier to use

A consumer network administrator and a consumer service administrator are working together to get:

- An easier way to configure Private Service Connect.
- The producer network to be able to initiate a connection to the consumer network.

Satisfy both of these needs using service connection policies.



A consumer network administrator and a consumer service administrator are working together to get:

- An easier way to configure Private Service Connect *or*
- The producer network to be able to initiate a connection to the consumer network.

Satisfy both of these needs using service connection policies.

Using service connection policies

- ✓ A regional Google Cloud resource
- ✓ Network admins specify producer services
- ✓ Consumer service admins can deploy services

Service connection policies have the following fields:

- Service class
- VPC network
- Subnets
- Connection limit

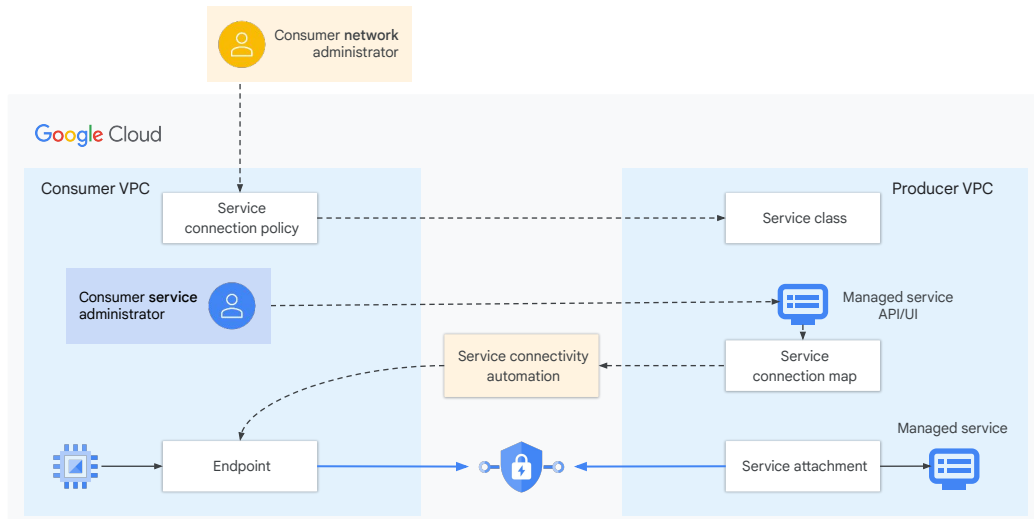


A service connection policy is a regional Google Cloud resource. It lets a network administrator specify which producer services can be deployed and connected through service connectivity automation. If a service connection policy exists for a managed service, a consumer service administrator can deploy that service.

Service connection policies have the following fields:

- **Service class:** specifies the type of managed service that the policy is for. Each producer that supports service connection policies has its own globally unique service class.
- **VPC network:** specifies the VPC network that the policy is scoped for.
- **Subnets:** specifies the subnets that IP addresses for Private Service Connect endpoints are allocated from.
- **Connection limit:** specifies the maximum number of Private Service Connect connections that a producer can create in the policy's VPC network and region.

Service instance deployment



Deploying an instance of a managed service by using service connection policies involves the following steps.

- A consumer network administrator creates a service connection policy for their VPC network. This policy lets Google automatically deploy Private Service Connect endpoints on behalf of a consumer service administrator. Consumer network administrators have more control over who can create and use private service connect endpoints.
- The service connection policy references a service class—a globally unique resource that identifies a specific producer service. A single service connection policy is scoped to a single service class and a single consumer VPC network, which delegates the ability to configure connectivity within that scope.
- A consumer service administrator deploys a managed service using the service's administrative API or UI. Google producer Service Attachments can be found using the UI or a describe command. Self-hosted and third-party service attachments URI's may be shared programmatically or through email depending on the implementation.
- The producer receives the consumer's connectivity configuration and passes this information to a service connection map.
- Private Service Connect service connectivity automation creates an endpoint in the consumer VPC network. This endpoint connects to a service attachment in the producer VPC network.

Caveats: Private Service Connect

01

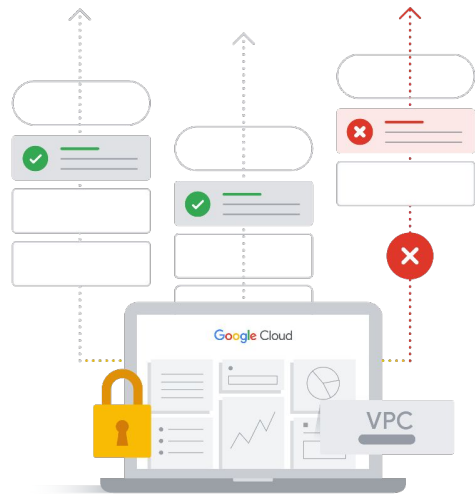
You can't create a Private Service Connect endpoint in the same VPC network as the published service that you are accessing.

02

The IP address that you use for the Private Service Connect endpoint counts toward the project quota for global internal IP addresses.

03

Private Service Connect endpoints are not accessible from peered VPC networks.



Private Service Connect has a few caveats.


You can't create a Private Service Connect endpoint in the same VPC network as the published service that you are accessing. The endpoint can only be used to access a published service in another VPC network.

The address counts toward the project quota for global internal IP addresses.


Private Service Connect endpoints are not accessible from peered VPC networks. Instead, create a Private Service Connect endpoint in the peered VPC network. You can then configure workloads to refer to that endpoint.

Connections from on-premises environments to non-Google services must use Cloud VPN tunnels. These on-premises environments must be in the same region as the Private Service Connect endpoint.

For information about accessing Private Service Connect endpoints from on-premises environments that are connected using Cloud VPN, see [Access the endpoint from on-premises hosts](#) in the Google Cloud documentation.



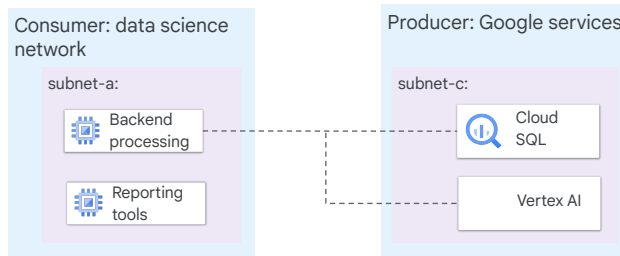
Today's agenda



- 01 Private access overview
- 02 Private Google Access
- 03 Private Service Connect
- 04 **Private services access**
- 05 Cloud NAT
- 06 Lab: Implement Private Google Access with Cloud NAT
- 07 Quiz

Next, let's discuss how to use private services access to provide access to producer services.

Use case: Connect to specific Google services without an external IP address



VMs on consumer network have no external IP addresses



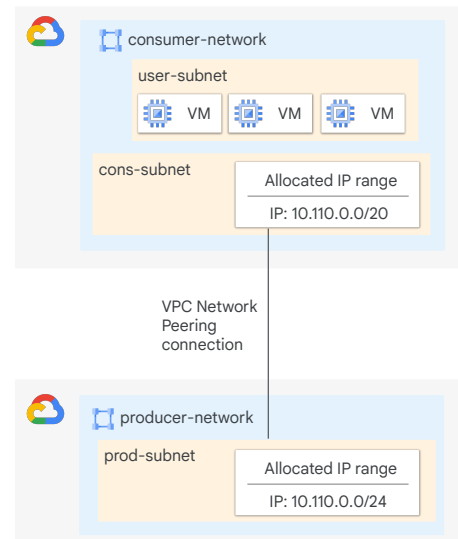
Consider a scenario. Mark is Cymbal Corporation's network architect. Within Cymbal, a data science team has developed a proprietary machine learning API deployed within a private VPC subnet on Google Cloud. These systems need to access essential Google Cloud services like Cloud SQL or Vertex AI.

Challenge: Ensuring connectivity to essential Google services while keeping the API inaccessible from the public internet.

PSA is the solution. Let us see how.

Private services access

- Uses internal IPv4 addresses.
- Uses VPC Network Peering to connect consumer and producer VPC networks.
- Automates much of the VPC Network Peering configuration.
- Doesn't require explicitly importing and exporting routes.
- Is only available for some producer services, like Apigee, Cloud SQL, and Cloud TPU.



Like with Private Service Connect, private services access lets consumers use internal IPv4 addresses to consume producer services. You allocate an internal IP range within your own VPC for PSA.

The connection between consumer and producer uses VPC Network Peering.

Private services access automates much of the VPC Network Peering configuration.

With Private Service Connect, you had to import and export routes between the consumer and producer VPC networks. Because the connection between the consumer and the producer is made using VPC Network Peering, you don't need to import and export routes. Subnet routes that don't use privately used public IP addresses are always exchanged between peered VPC networks.

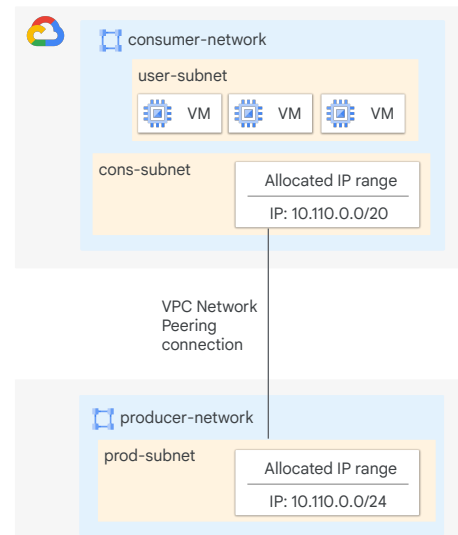
Private services access is available only for supported producer services, like Apigee, Cloud SQL, and Cloud TPU. For a complete list of supported producer services, see Private services access [Supported services](#) in the Google Cloud documentation.

To offer private connectivity, the service producer must complete a one-time onboarding process. To complete the onboarding process, contact your Google representative. For more information, see [Onboarding process](#) on the Enabling private services access page of the Google Cloud documentation.

After the onboarding process is complete, you can configure private services access.

Configuring private services access

- ✓ The service producer and consumer must activate the Service Networking API in their projects.
- ✓ Service producers must allocate an IPv4 address range in the VPC network that contains the service.
- ✓ Consumers must:
 - Allocate an IPv4 address range in their VPC network.
 - Create a private connection to a service producer.



To use private services access, both service consumers and producers must activate the Service Networking API in their projects. The consumer and producer VPC networks require some configuration as well.

Service producers must allocate an IPv4 address range in the VPC network that contains the service. This address range is used for each connection from a service consumer.

Service consumers must also allocate an IPv4 address range in their VPC network for each service producer. For example, to use services from three different producer VPC networks, the consumer must allocate three IPv4 address ranges, one for each producer VPC network.

After the service producer has completed the initial configuration, consumers can create a private services access connection to the producer VPC network. You can use the Google Cloud console or the Google Cloud CLI to create this connection. Consumers and producers can use the Google Cloud console to edit their VPC network to configure private services access.

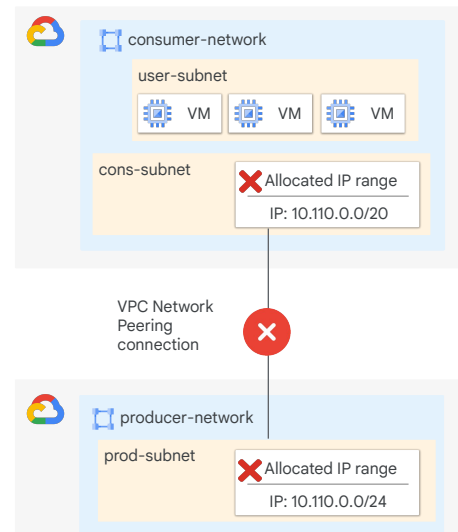
If a service producer offers multiple services, you only need one private connection. For example, if a consumer uses Cloud SQL and Cloud TPU, only one private connection is created.

Google Cloud uses VPC Network Peering to implement the connection between the

consumer and producer VPC networks.

Deleting the connection

- ✓ Consumers can disable the private services access connection between their VPC network and the producer VPC network.
- ✓ Disabling the connection does not:
 - Delete the VPC Network Peering connection.
 - Release the IPv4 address range.



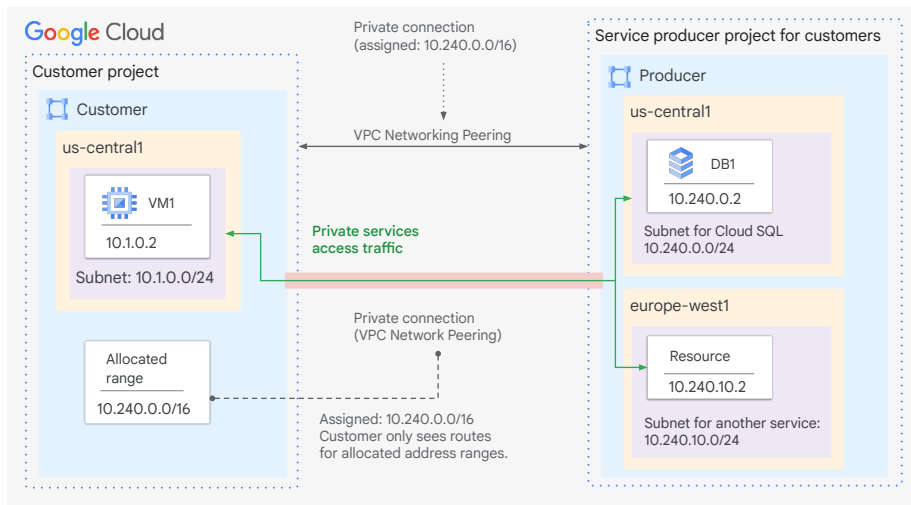
Consumers can disable the private services access connection between their VPC network and the producer VPC network. Consumers can also edit their VPC network settings to disable access.

Disabling the private services access connection does not

delete the VPC Network Peering to the producer VPC network. You can delete the VPC Network Peering connection by editing the VPC network.

Likewise, disabling the private services access connection does not release the IPv4 address range. Consumers must edit the VPC network to release the IPv4 address range.

Sample topology



This example shows a sample private services access topology. The customer VPC network allocated the 10.240.0.0/16 address range for Google services and established a private connection that uses the allocated range. Google then creates a project for the customer. With that project, each Google service creates a subnet from the allocated block to provision new resources in a given region. In the example, you can see a Cloud SQL instance. The Cloud SQL instance is assigned an IP of 10.240.0.2, which is within the 10.240.0.0/16 range.

In the customer VPC network, requests with a destination of 10.240.0.2 are routed over the private connection to the producer VPC network. The request is then sent to the correct resource in the producer VPC network.

If the service supports cross-region communication, VM instances in the customer network can access service resources in any region. Some services might not support cross-region communication. For more information, see the documentation of the relevant service.

Caveats: Private services access

01

If you connect on-premises networks, you must export the routes to the VPC producer network.

02

Not all Google services are supported.

03

The same quota and limits that apply to VPC Network Peering also apply to private services access.



Private services access has a few caveats.

For private services access to an on-premises network to work, you must export custom routes from the on-premises network to the producer VPC network.


Not all Google services are supported. For a complete list of supported Google services, see Private services access [supported services](#) in the Google Cloud documentation.

The same quota and limits that apply to VPC Network Peering also apply to private services access. Private services access uses VPC Network Peering to implement connections and thus has the same restrictions as VPC Network Peering.


A quick summary

| Private Google Access | Private Service Connect | Private service access |
|--|--|--------------------------------------|
| Helps VMs reach Google services with an internal IP address. | Helps expose <i>your</i> / Google-produced / third-party services to others. | Reaches producer services privately. |

This table outlines three Google Cloud networking features: Private Google Access (PGA) simplifies how VMs without public IPs connect to essential Google services; Private Service Connect (PSC) allows you to expose your own services or Google produced or 3rd party IP services securely to external consumers (even outside of Google Cloud); and Private Services Access (PSA) enables private access to services provided by Google or third-parties within your VPC.



Today's agenda



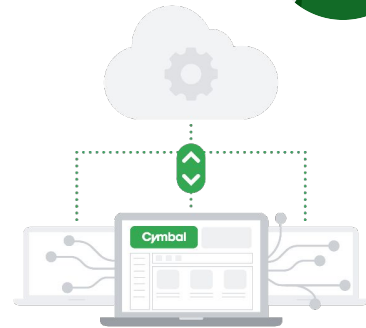
- | | |
|----|---|
| 01 | Private access overview |
| 02 | Private Google Access |
| 03 | Private Service Connect |
| 04 | Private services access |
| 05 | Cloud NAT |
| 06 | Lab: Implement Private Google Access with Cloud NAT |
| 07 | Quiz |

Next, you will learn how to use Cloud NAT to provide access to the public internet for resources without an external IP address.

Use case: Allow access to internet without a public IP address



- ✓ Cymbal has several non-production environments (development, testing, staging) on Google Cloud.
- ✓ These environments host various VMs that need occasional outbound internet access for tasks such as:
 - Downloading software updates and dependencies.
 - Accessing external testing tools or resources.



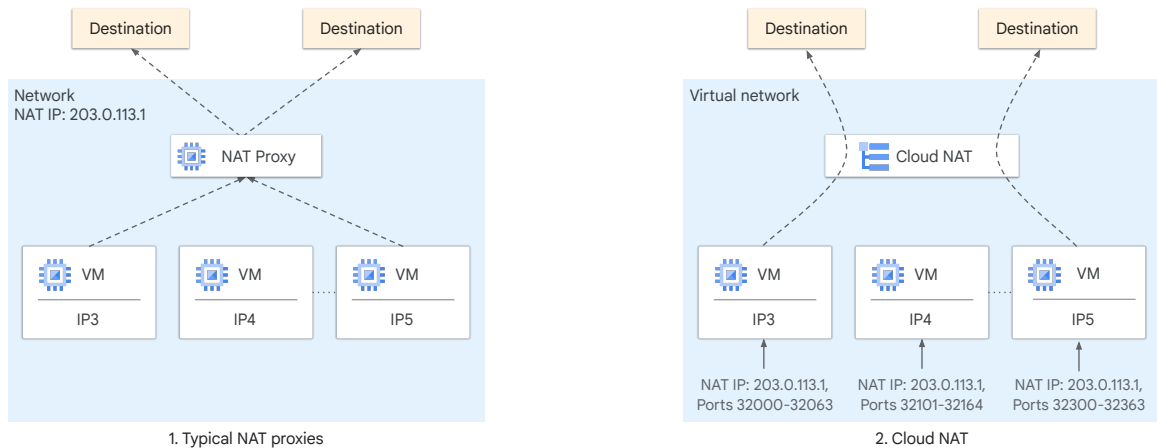
Let us start with a simple use case. Janet is a network engineer at Cymbal Corporation. Cymbal has several non-production environments (development, testing, staging) on Google Cloud. These environments host various VMs that need occasional outbound internet access for tasks such as:

- Downloading software updates and dependencies.
- Accessing external testing tools or resources.

Cymbal needs a streamlined, cost-effective way to handle outbound traffic without burning through their limited pool of public IPs.

Solution: Janet decides to use Cloud NAT. This managed service provides a pool of highly available IP addresses that Janet can dynamically assign to their servers on demand. Cloud NAT translates the private IP addresses of Amal's servers to a public IP address for outbound communication, seamlessly masking the internal network from the outside world.

Cloud NAT is a fully managed, software-defined service



Cloud NAT is the Google-managed network address translation service. It lets you provision your application instances without public IP addresses, and it also lets them access the internet in a controlled and efficient manner. With Cloud NAT, your private instances can access the internet for updates, patching, configuration management, and more.

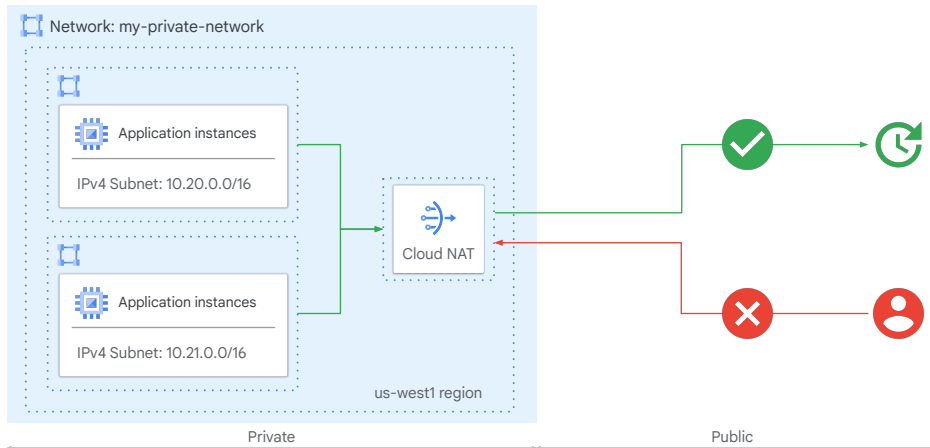
Cloud NAT, as shown on the right, offers several advantages when compared to other NAT offerings, as shown on the left.

As a fully managed, software-defined service, Cloud NAT differs from traditional NAT proxy solutions. There are no NAT middle proxies in the path from the instance to the destination. Instead, each instance is allocated a NAT IP address along with a slice of the associated port range. This allocated IP address and port range are used by the instance to perform NAT. This design is free of chokepoints and is highly reliable, performant, and scalable.

Cloud NAT lets you configure multiple NAT IP addresses per NAT gateway. You can scale based on the size of your network without having to add or manage another NAT gateway. NAT IP allocation has two modes: manual and auto. The manual mode provides full control when specifying IP addresses. If you want to allow NAT addresses on the receiving side, use the manual mode. The auto mode enables the NAT IP addresses to be allocated and scaled automatically based on the number of instances.

For a full overview of Cloud NAT features, see [Cloud NAT overview](#) in the Google Cloud documentation.

Cloud NAT provides internet access to private instances



In this diagram, Cloud NAT enables two private instances to access an update server on the internet, which is referred to as outbound NAT. However, Cloud NAT does not implement inbound NAT.

In other words, hosts outside your VPC network can't directly access any of the private instances behind the Cloud NAT gateway. Your VPC networks remain isolated and secure.

Benefits of Cloud NAT



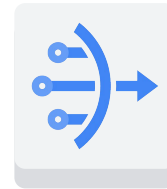
Reduces the need for individual VMs to each have external IP addresses.



Automatically scales the number of NAT IP addresses that it uses.



Is not dependent on a single physical gateway device.



Cloud NAT

With Cloud NAT, VMs without external IP addresses can access destinations on the internet. For example, you might have VMs that only need internet access to download updates or complete provisioning. Cloud NAT allows you to configure these VMs with an internal IP address. Thus, your organization needs fewer external IP addresses.

Cloud NAT can be configured to automatically scale the number of NAT IP addresses that it uses. Cloud NAT supports VMs that belong to managed instance groups, including those with autoscaling enabled.

Cloud NAT is not dependent on a single, physical gateway device. Cloud NAT is a distributed, software-defined managed service. You configure a NAT gateway on a Cloud Router, which provides the control plane for NAT. Cloud Router contains the NAT configuration parameters. Google Cloud runs and maintains processes on the physical machines that run your Google Cloud VMs.

How Cloud NAT works with Private Google Access



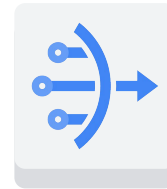
Cloud NAT never performs NAT for traffic that is sent to the select external IP addresses for Google APIs and services.



When you configure a Cloud NAT gateway to apply to a subnet range, Google Cloud automatically enables Private Google Access for that range.



If the gateway provides NAT for a subnet range, Private Google Access can't be disabled manually.



Cloud NAT

With Private Google Access, Cloud NAT never performs NAT for traffic sent to the selected external IP addresses of Google APIs and services. Google Cloud routes this traffic internally.

When you configure a Cloud NAT gateway to apply to a subnet range, Google Cloud automatically enables Private Google Access for that range. Thus, any VMs in that subnet range use Private Google Access to connect to Google APIs and services.

If the gateway provides NAT for a subnet range, Private Google Access is in effect for that range and can't be disabled manually.

Types of Cloud NAT

Public NAT

- Enables your private resources (without public IPs) to access the internet.
- Shares a pool of public IP addresses, assigning them as needed for secure outbound connections.

Private NAT

- Facilitates private communication between different networks.
- Inter-VPC NAT: Translates addresses between VPCs connected to a central hub.

There are two types of Cloud NAT available in Google Cloud.

First up, we have Public NAT. Think of this as your gateway to the internet. It enables the resources inside your Virtual Private Cloud – those without public IP addresses – to securely access the online world.

Here's how it works: when a resource needs to connect to the internet, Public NAT dynamically assigns it a public IP address from a shared pool. This way, your resource can browse, download updates, and do all the things it needs to online, while its internal IP address remains safely tucked away."

Now, let's explore Private NAT. This is the type of NAT you'll use for secure communication between different networks.

One important subtype of Private NAT is Inter-VPC NAT. This comes in handy when you have multiple Virtual Private Clouds that need to talk to each other. It acts like a translator, making sure communication flows smoothly between these different networks – all through a central hub, like the Network Connectivity Center.

With both Public and Private NAT options at your disposal, you have the flexibility to securely connect your resources in Google Cloud, whether they need to access the internet or communicate with other private networks.

Let's ask Gemini



How can I use **private service access** to connect my VMs to **third-party services**?



Sure, here are the steps on how you can use private service access to connect your VMs to third-party services:

...

1.Enable the Private Service Access API.

2.Create a service attachment.


- **gcloud services attachments create my-service-attachment **
- **--project=my-project **
- **--region=us-central1 **
- **--network=my-network **
- **--service=serviceproducer.googleapis.com**

Before we wrap up this module, let us take a look at a sample Gemini prompt related to private service access.


Here is an example of an operational prompt . “How can I use private service access to connect my VMs to third-party services?”.

You will be provided with a well structured response that outlines the pre-requisites, steps, and detailed commands that can help you configure private service access.

The output's structure makes it easy to follow and minimizes the risk of errors during implementation.



Today's agenda

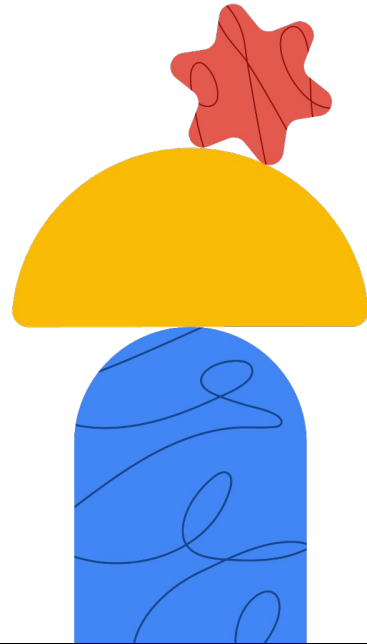


- 01 Private access overview
- 02 Private Google Access
- 03 Private Service Connect
- 04 Private services access
- 05 Cloud NAT
- 06 [Lab: Implement Private Google Access with Cloud NAT](#)
- 07 Quiz

Next, you will apply what you learned by completing a lab exercise.


Lab intro

Implement Private Google
Access and Cloud NAT




In this lab, you will complete the following tasks:

- Configure a VM instance that doesn't have an external IP address.
- Connect to a VM instance using an Identity-Aware Proxy (IAP) tunnel.
- Enable Private Google Access on a subnet.
- Configure a Cloud NAT gateway.
- Verify access to public IP addresses of Google APIs and services and other connections to the internet.



Today's agenda



- 01 Private access to Google APIs and services
- 02 Private Google Access
- 03 Private Service Connect
- 04 Cloud NAT
- 05 Lab: Implement Private Google Access and Cloud NAT
- 06 [Quiz](#)

Next, you will test your knowledge of private connection options with a brief quiz.

Quiz | Question 1

Question

You want to provide access to services that you created in a VPC network. The services should be available to other specified VPC networks through endpoints that have internal IP addresses. Some of these VPC networks have subnets with overlapping internal IP addresses. Which product can you use?

- A. Private Google Access
- B. Private services access
- C. Private Service Connect
- D. Cloud NAT

Quiz | Question 2

Question

To enable Private Google Access for a VPC network:

- A. Enable it on the VPC network.
- B. Enable it on all desired subnets in the VPC network.
- C. Enable it on all desired subnets and on Cloud Router.
- D. Enable it on the VPC network, on the desired subnets, and on Cloud Router.

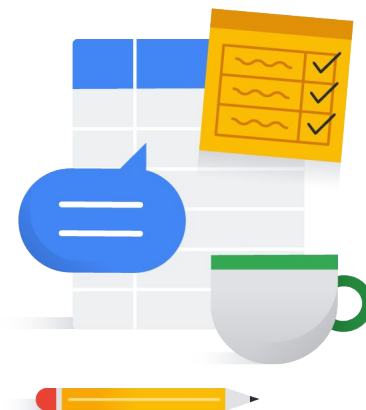
Quiz | Question 3

Question

Private services access automatically configures which Google Cloud product to implement communication between the producer and consumer VPC networks?

- A. Shared VPC
- B. VPC Network Peering
- C. Private Google Access
- D. Cloud NAT

Debrief



In this module, you learned about several ways to connect privately from internal IP addresses to Google Cloud APIs, Google services, and other resources. You can use Private Google Access to connect to Google APIs and services. In addition to Google APIs and services, Private Service Connect lets you connect to other configured resources. Private services access simplifies creating a VPC Network Peering connection between consumer and producer VPC networks. However, private services access only works for some Google products and services. We then discussed using Cloud NAT to provide public internet access for resources without public IP addresses. We finished the module with a lab exercise to implement Private Google Access with Cloud NAT, followed by a short quiz.



THANK YOU