Google Cloud provides several private access options. Each option allows virtual machine (VM) instances with internal (RFC 1918 (https://tools.ietf.org/html/rfc1918)) IP addresses to reach certain APIs and services. Choose an option that supports the APIs and services that you need to access.

The following table summarizes each option. You can configure one or all of these options. They operate independently of each other.

Option	Clients	Connection	Supported services	Usage
Private Google Access (#pga)				
	Google Cloud VM instances *without* external IP addresses.	Connect to the standard public IP addresses (/vpc/docs/configure-private-google-access#dns_resolution) or Private Google Access domains and VIPs (/vpc/docs/configure-private-google-access#private-domains) for Google APIs and services through the Virtual Private Cloud (VPC) network's default internet gateway.	Supports most Google APIs and services (#pga-supported). Also supports access to App Engine applications.	Use this option to connect to Google APIs and services without giving your Google Cloud resources external IP addresses.
Private Google Access for on-premises hosts (#pga-onprem)				
	On-premises hosts with or without external IP addresses.	Connect to Google APIs and services, from your on-premises network, through a Cloud VPN tunnel or Cloud Interconnect by using one of the Private Google Access-specific domains and VIPs (#private-vips).	The Google services (#supported-services-onprem) that you can access depend on which Private Google Access-specific domain you use.	Use this option to connect to Google APIs and services *through* a VPC network. This method doesn't require your on-premises hosts to have external IP addresses.
Private services access (#service-networking)				
	Google Cloud VM instances with or without external IP addresses.	Connect to a Google or third-party managed VPC network through a VPC Network Peering connection.	Supports some Google (#private-services-supported-services) or third-party services	Use this option to connect to specific Google and third-party services without assigning external IP addresses to your Google Cloud and Google or third-party resources.
Serverless VPC Access (#serverless-vpc-access)				

Option	Clients	Connection	Supported services	Usage
	Google Cloud VM instances with or without external IP addresses.	Connect directly from serverless Google services through an internal VPC connection.	App Engine standard environment and Cloud Functions (#serverless-vpc-access-supported-services)	Use this option to connect from the App Engine standard environment and Cloud Functions directly to resources in a VPC network using internal IP addresses.

VM instances that only have internal IP addresses (no external IP addresses) can use Private Google Access. They can reach the external IP addresses of Google APIs and services. The source IP address of the packet can be the primary internal IP address of the network interface or an address in an alias IP range that is assigned to the interface. If you disable Private Google Access, the VM instances can no longer reach Google APIs and services; they can only send traffic within the VPC network.

Private Google Access has no effect on instances that have external IP addresses. Instances with external IP addresses can access the internet, according to the internet access requirements (/vpc/docs/vpc#internet_access_reqs). They don't need any special configuration to send requests to the external IP addresses of Google APIs and services.

You enable Private Google Access on a subnet by subnet basis; it's a setting for subnets in a VPC network. To enable a subnet for Private Google Access and to view the requirements, see Configuring Private Google Access (/vpc/docs/configure-private-google-access).

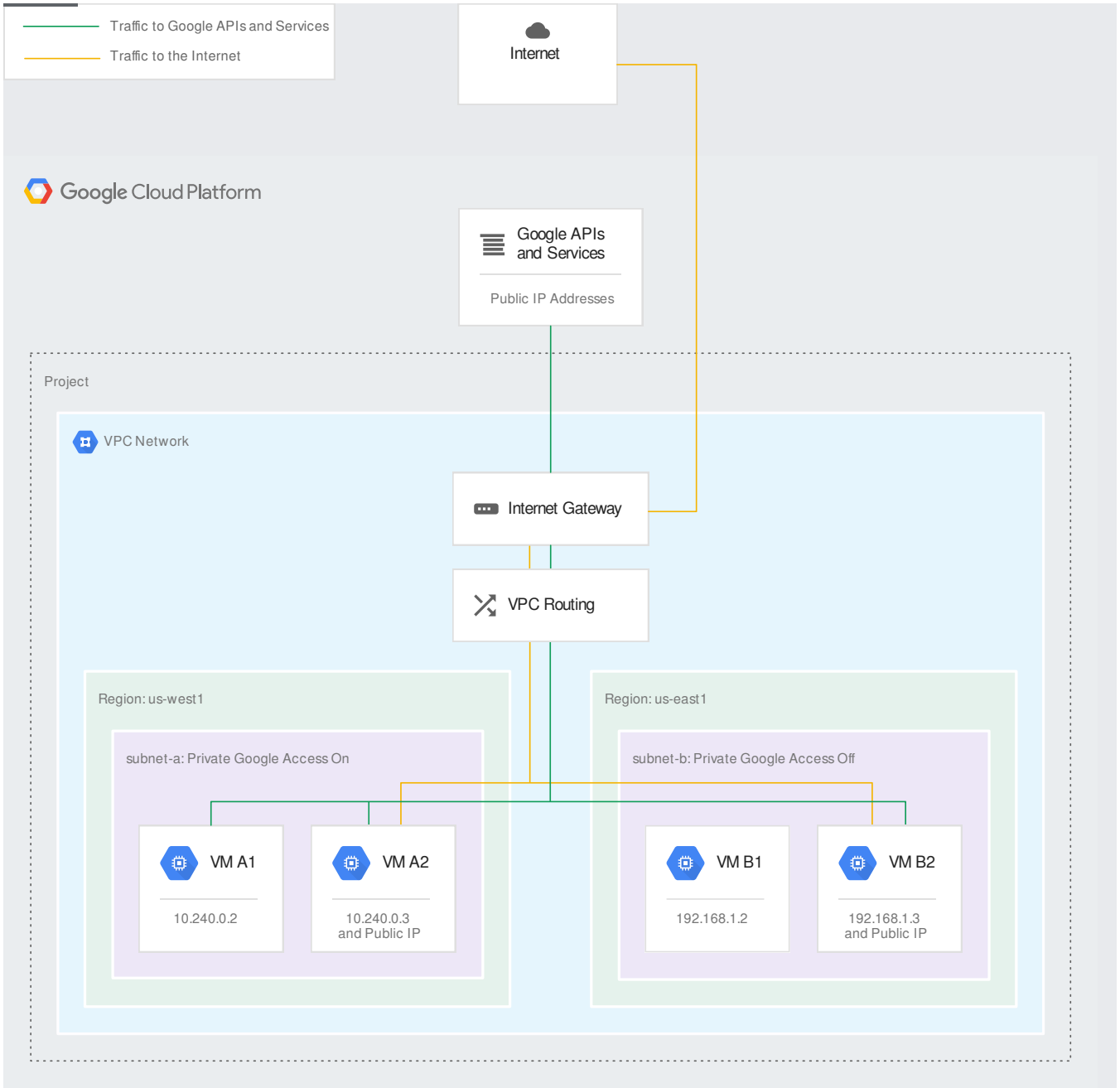
Private Google Access permits access to Cloud and Developer APIs (https://developers.google.com/apis-explorer/#p/) and most Google Cloud services, **except** for the following services:

- App Engine Memcache
- Filestore
- Memorystore
- Cloud SQL

Instead, private services access might support (#private-services-supported-services) one or more of them.

The VPC network in the following example meets the routing requirement for Private Google Access (/vpc/docs/configure-private-google-access#ensuring_that_routing_is_properly_configured) because it has routes to the public IP addresses for Google APIs and services whose next hops are the default internet gateway. Private Google Access is enabled (/vpc/docs/configure-private-google-access#enabling-pga) for `subnet-a` but not for `subnet-b`.

Even though the next hop for the required routes is called "default internet gateway" and the IP addresses for Google ervices are public, requests to Google APIs and services from VMs that have only internal IP addresses in **`subnet-a`** (e Google Access is enabled) are *not* sent through the public internet. Those requests stay within Google's network. onally, VMs that only have internal IP addresses do not meet the internet access requirements 'docs/vpc#internet_access_reqs) for access to other public IP addresses beyond those for Google APIs and services



(/vpc/images/private-google-access.svg)

Implementation of Private Google Access (click to enlarge)

The following list provides details about the above diagram:

- Firewall rules (/vpc/docs/configure-private-google-access#firewall_rules) in the VPC network have been configured to allow egress to `0.0.0.0/0` (or at least to the server IPs for Google APIs and services).
- `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` **cannot** 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 they each have external IP addresses. Private Google Access has no effect on whether or not these instances can access Google APIs and services because both have external IP addresses.

On-premises hosts can reach Google APIs and services by using Cloud VPN (/vpn/docs) or Cloud Interconnect (/interconnect/docs) from your on-premises network to Google Cloud. On-premises hosts can send traffic from the following types of source IP addresses:

- a private IP address, such as an RFC 1918 (https://tools.ietf.org/html/rfc1918) address
- a privately used public IP address, except for a Google-owned public IP address. (Private Google Access for on-premises hosts does not support re-using Google public IP addresses as sources in your on-premises network.)

Sending packets from non-RFC 1918 source IP addresses is in Beta.

To enable Private Google Access for on-premises hosts, you must configure DNS, firewall rules, and routes in your on-premises and VPC networks. You don't need to enable Private Google Access for any subnets in your VPC network as you would for Private Google Access for Google Cloud VM instances.

On-premises hosts must connect to Google APIs and services by using the virtual IP addresses (VIPs) for either the `restricted.googleapis.com` or `private.googleapis.com` domains. Refer to Private Google Access-specific domains and VIPs (#private-vips) for more details.

Google publicly publishes DNS A records that resolve the domains to a VIP range. Even though the ranges are from a public IP range, Google does not publish routes for them. Therefore, you must add a custom route advertisement on a Cloud Router and have an appropriate custom static route in your VPC network for the VIP's destination.

The route must have a destination matching one of the VIP ranges and a next hop being the default internet gateway. Traffic sent to the VIP range stays within Google's network instead of traversing the public internet because Google does not publish routes to them externally.

For configuration information, refer to Configuring Private Google Access for On-premises Hosts (/vpc/docs/configure-private-google-access-hybrid).

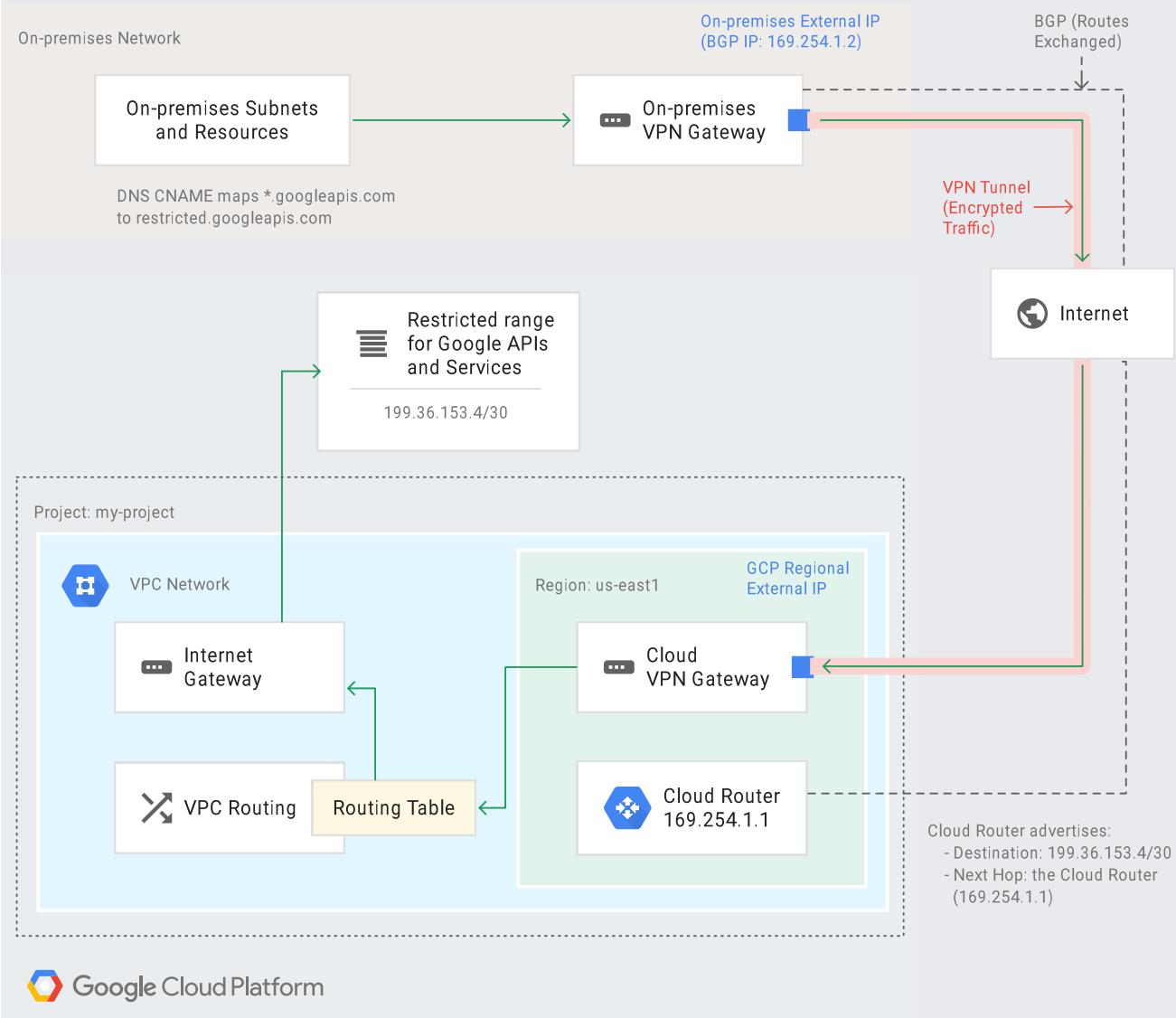
The following table describes the domain names and their VIP range. You must use one of these VIPs for Private Google Access for on-premises hosts.

Domain and VIPs	Supported services	Example usage
`restricted.googleapis.com` **`199.36.153.4/30`**	Enables API access to Google APIs and services that are supported by VPC Service Controls (/vpc-service-controls/docs/supported-products). Blocks access to Google APIs and services that do not support VPC Service Controls (/vpc-service-controls/docs/overview). Does not support G Suite web applications or G Suite APIs.	Use **`restricted.googleapis.com`** to make only the VPC Service Controls restricted services available to hosts in your VPC network or on-premises network.
`private.googleapis.com` **`199.36.153.8/30`**	Enables API access to most Google APIs and services regardless of whether they are supported by VPC Service Controls. Includes API access to Google Maps, Google Ads, Google Cloud platform, and most other Google APIs whose names end in **`googleapis.com`**. Does not support G Suite web applications.	Use **`private.googleapis.com`** when you must access Google APIs and services under the following circumstances: • You are not using VPC Service Controls. • You are using VPC Service Controls but also need to access services that are not supported by VPC Service Controls.

f you use or plan to use VPC Service Controls, Google recommends that you direct DNS requests for **`*.googleapis`** **`restricted.googleapis.com`** VIP unless you must also access Google APIs and services that are not supported b ervice Controls. Although you can access services that are supported by VPC Service Controls on both VIPs, only the **`restricted.googleapis.com`** domain provide additional mitigation for data exfiltration risks because they deny s to Google APIs and services that are not supported by VPC Service Controls.

Services available to on-premises hosts are limited to those supported by the domain name and VIP used to access them. Refer to Private Google Access-specific domains and VIPs (#private-vips) for details.

In the following 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 is 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`).



(/vpc/images/pga-onprem.svg)

Private Google Access for hybrid cloud use case (click to enlarge)

- The on-premises DNS configuration maps `*.googleapis.com` requests to `restricted.googleapis.com`, which resolves to the `199.36.153.4/30`.
- Cloud Router has been configured to advertise the `199.36.153.4/30` IP address range through the Cloud VPN tunnel by using a custom route advertisement. Traffic going to Google APIs is routed through the tunnel to the VPC network.
- A custom static routes was added to the VPC network that directs traffic with the destination `199.36.153.4/30` to the default internet gateway (as the next hop). Google then routes traffic to the appropriate API or service.
- If you created a Cloud DNS managed private zone for `*.googleapis.com` that maps to `199.36.153.4/30` and have authorized that zone for use by your VPC network, requests to anything in the `googleapis.com` domain are sent to the IP addresses that are used by `restricted.googleapis.com`. Only the supported APIs (#supported-services-onprem) 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 (https://www.wikipedia.org/wiki/BIND).

Google and third parties (together known as *service producers*) can offer services with internal IP addresses that are hosted in a VPC network. Private services access enables you to reach those internal IP addresses. This is useful if you want your VM instances in your VPC network to use internal IP addresses instead of external IP addresses. For details about using private services access, see Configuring Private Services Access (/vpc/docs/configure-private-services-access/).

Private services access requires you to first allocate an internal IP address range and then create a private connection. An allocated range is a reserved CIDR block that can't be used in your local VPC network. It's set aside for service producers only and prevents overlap between your VPC network and the service producer's VPC network. When you create a private connection, you must specify an allocation.

The private connection links your VPC network with the service producer's VPC network. This connection allows VM instances in your VPC network to use internal IP addresses to reach the service resources that have internal IP addresses. Your instances can have external IP addresses, but external IP addresses are not required for, and not used by, private services access.

If a service producer offers multiple services, you only need one private connection. When you create a private connection, you use the Service Networking API to create it. However, Google Cloud implements this connection as a VPC Network Peering (/vpc/docs/vpc-peering) connection between your VPC network and the service producer's VPC network. For example, your VPC network shows it as a peering connection, and to delete the private connection, you must delete the peering connection.

Because a private connection is implemented as a VPC Network Peering connection, the behaviors and constraints o
g connections also apply to private connections, such as VPC Network Peering limits (/vpc/docs/quota#vpc-peering

You can use private services access only with services that support (#private-services-supported-services) it. Check with the service producer before creating a private connection.

On the service producer's side of the private connection is a VPC network, where your service resources are provisioned. The service producer's network is created exclusively for you and contains only your resources.

A resource in the service producer network is similar to other resource in your VPC network. For example, it's reachable through internal IP addresses by other resources in your VPC network. You can also create firewall rules in your VPC network to control access to the service producer's network.

For details about the service producer side, see Enabling private services access (/service-infrastructure/docs/enabling-private-services-access) in the Service Infrastructure documentation. This documentation is for your information only and is not required for you to enable or use private services access.

In hybrid networking scenarios, an on-premises network is connected to a VPC network either through a Cloud VPN (/vpn/docs) or Cloud Interconnect (/interconnect/docs) connection. By default, on-premises hosts can't reach the service producer's network by using private services access.

In the VPC network, you might have custom static or dynamic routes to correctly direct traffic to your on-premises network. However, the service producer's network doesn't contain those same routes. When you create a private connection, the VPC network and service producer network exchange subnet routes only.

You must export the VPC network's custom routes so that the service provider's network can import them and correctly route traffic to your on-premises network.

The service producer's network contains a default route (`0.0.0.0/0`) that goes to the internet. If you export a defau o the service producer's network, it is ignored because the service producer network's default route takes precedence d, define and export a custom route with a more specific destination. For more information, see Routing order 'docs/routes#routeselection).

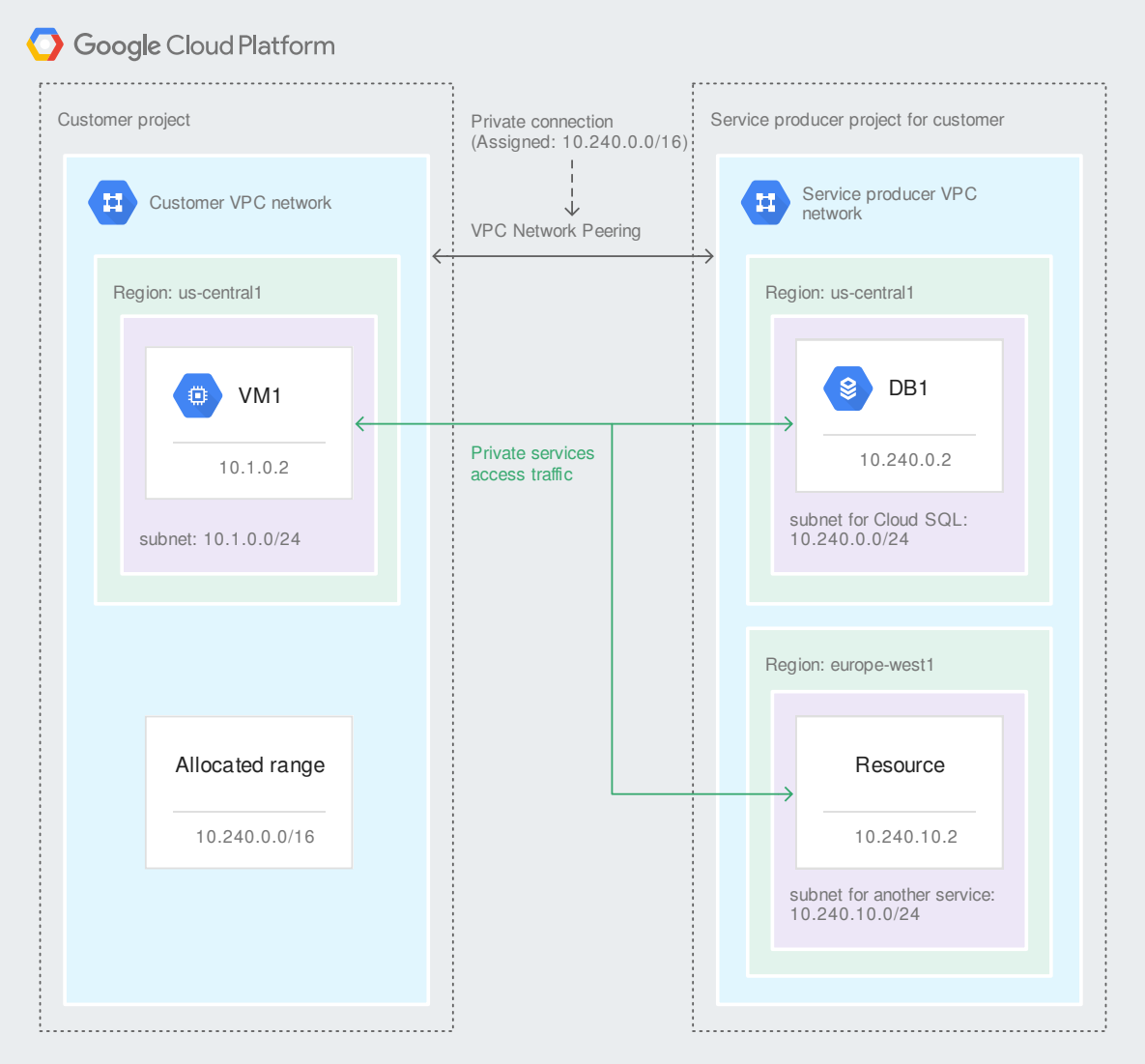
To export custom routes, you must create a private connection and then modify the underlying VPC peering configuration to export custom routes. For information about creating a private connection, see Configuring Private Services Access (/vpc/docs/configure-private-services-access/). For information about exporting custom routes, see Updating a peering connection (/vpc/docs/using-vpc-peering#update-peer-connection).

The following Google services support private services access:

- Cloud SQL (/sql/docs/mysql/private-ip)

When you use private services access as a service consumer, you are solely responsible for securing your VPC netwo resources and data available on them. Google is not responsible for how your data and resources may be accessed y the third party that you are connecting with.

In the following example, 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. Each Google service creates a subnet from the allocated block to provision new resources in a given region, such as Cloud SQL instances.



(/vpc/images/private-services-access-sql.svg)

Private services access (click to enlarge)

- The private connection is assigned the `10.240.0.0/16` allocated range. From this allocation, Google services can create subnets where new resources are provisioned.
- On the Google services side of the private connection, Google creates a project for the customer. The project is isolated, meaning no other customers share it and the customer is billed for only the resources the customer provisions.
- Each Google service creates a subnet in which to provision resources. The subnet's IP address range is typically a `/24` CIDR block that is chosen by the service and comes from the allocated IP address range. You cannot modify the service producer's subnet. A service provisions new resources in existing regional subnets that were previously created by that service. If a subnet is full, the service creates a new subnet in the same region.

- VM instances in the customer's network can access service resources in any region if the service supports it. Some services might not support cross-region communication. For example, VM instances can only communicate with Cloud SQL instances that are in the same region. View the relevant service's documentation for more information.
- Egress costs (/compute/network-pricing#general) for cross-regional traffic, where a VM instance communicates with resources in a different region, still apply.
- The Cloud SQL instance is assigned the IP address `10.240.0.2`. In the Customer VPC network, requests with a destination of `10.240.0.2` are routed to the private connection over to the service producer's network. After reaching the service network, the service network contains routes that direct the request to the correct resource.
- Traffic between VPC networks travels internally within Google's network, not through the public internet.

Serverless VPC Access enables you to connect from the App Engine standard environment and Cloud Functions directly to your VPC network. This connection makes it possible for your App Engine standard environment apps and Cloud Functions to access resources in your VPC network via internal IP addresses.

With Serverless VPC Access, you create a *connector* in your Google Cloud project and attach it to a VPC network. You then configure your serverless services (such as App Engine apps or Cloud Functions) to use this connector for internal network traffic.

Serverless VPC Access only allows your app or function to send requests to resources in your VPC network and receive responses to those requests. Communication in the opposite direction, where a VM initiates a request to an app or function, requires you to use the public address of the app or function—see Private Google Access (#pga) for more information.

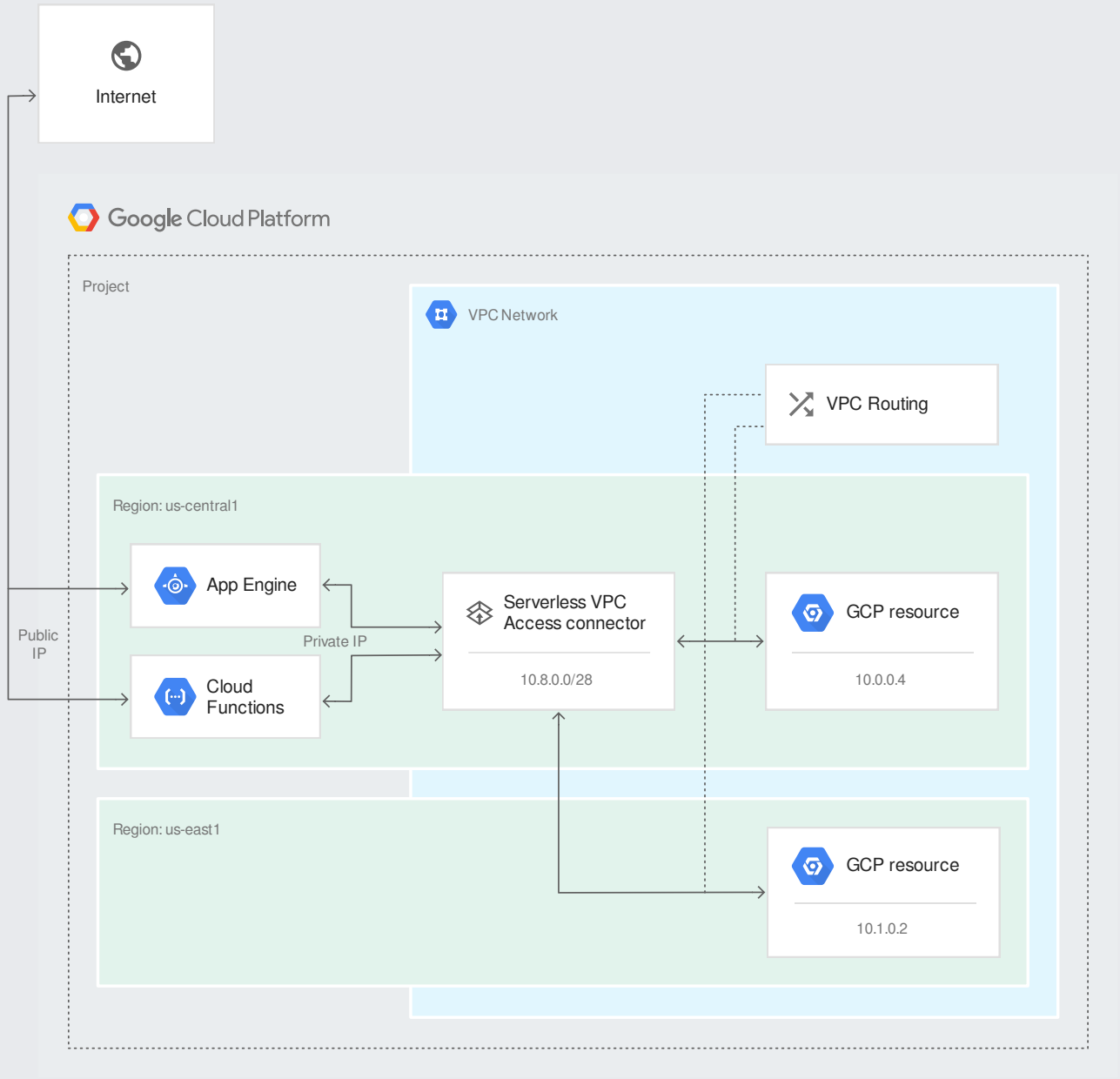
Serverless VPC Access does not support legacy networks (/vpc/docs/legacy) or vpc_shared_vpc networks (/vpc/docs/provisioning-shared-vpc). For more information, see Configuring Serverless VPC Access (/vpc/docs/configure-serverless-vpc-access).

The following Google services support Serverless VPC Access connectors:

- App Engine standard environment

- All runtimes except PHP 5.5 and Go 1.9
- Cloud Functions

In the following example, an App Engine standard environment app and Cloud Functions use a Serverless VPC Access connector to send requests to internal resources in the VPC network.



(/vpc/images/serverless-vpc-access.svg)

Serverless VPC Access example (click to enlarge)

- The Serverless VPC Access connector is in the same project and region as the App Engine app and Cloud Functions.
- The connector is attached to the VPC network that contains the destination resources. The connector can access resources in other VPC networks and Google Cloud projects if you use VPC Network Peering (/vpc/docs/using-vpc-peering).
- The connector is assigned the IP range `10.8.0.0/28`. Requests sent from the connector to the destination have a source IP address in this range.
- The App Engine app and Cloud Functions reach the destination resources by sending requests to their internal IP addresses, `10.0.0.4` and `10.1.0.2`. The destination resources can be in any region. Egress costs (/compute/network-pricing#general) apply to traffic sent from the connector to a resource in a different region.
- Requests sent from App Engine and Cloud Functions to internal (private) IP addresses travel internally through the Serverless VPC Access connector to the destination resource. Requests sent to external (public) IP addresses travel through the public internet.

- To configure Private Google Access see Configuring Private Google Access (/vpc/docs/configure-private-google-access).
- To configure Private Google Access for on-premises hosts, Configuring Private Google Access for on-premises hosts (/vpc/docs/configure-private-google-access-hybrid).
- To configure private services access, see Configuring Private Services Access (/vpc/docs/configure-private-services-access).
- To configure Serverless VPC Access, see Configuring Serverless VPC Access (/vpc/docs/configure-serverless-vpc-access).