Published in Google Cloud - Community

Jasbirs Follow

Jan 2 · 9 min read · Listen

Save

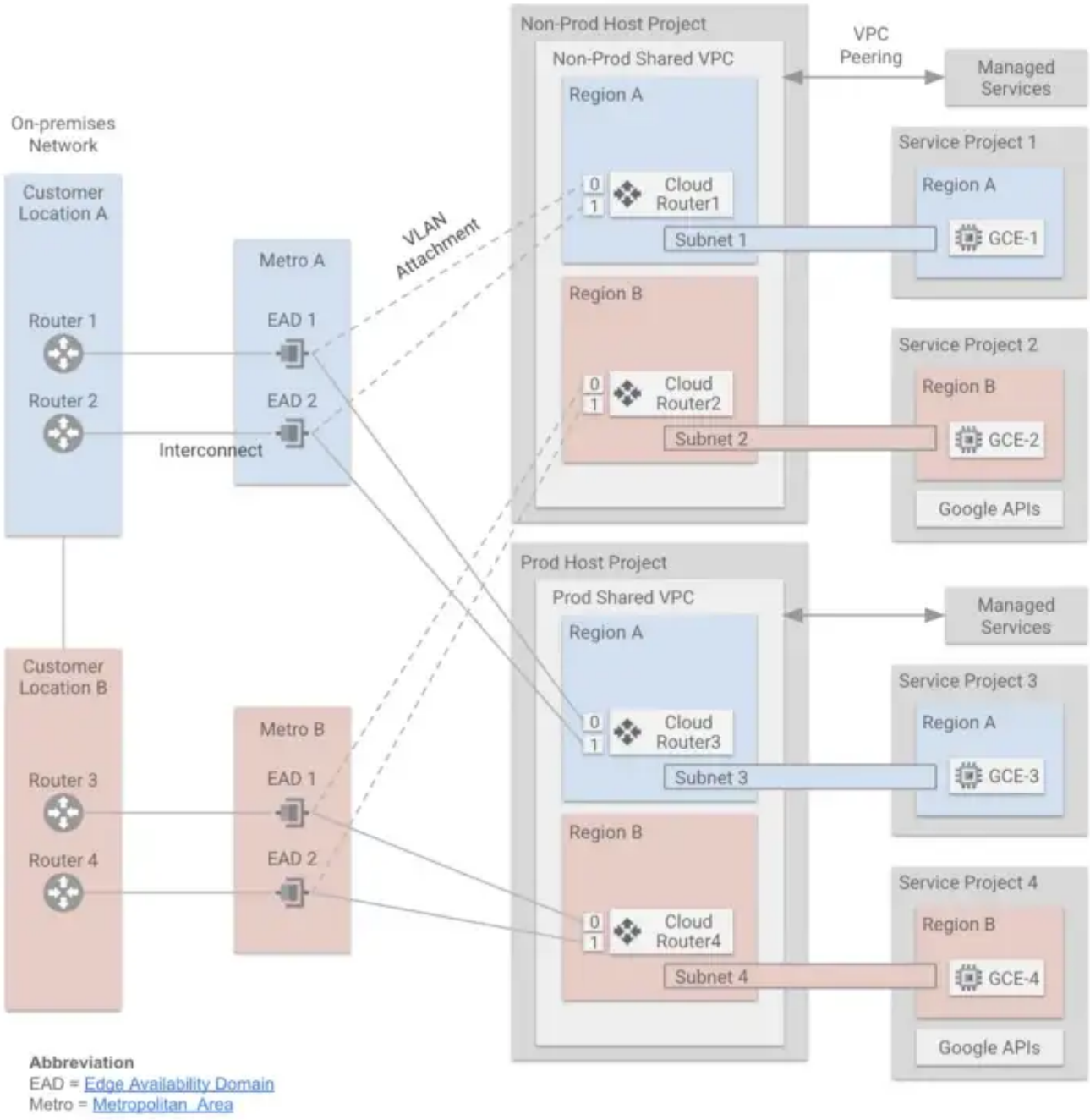
GCP Hybrid Networking Patterns — Part 2

This blog is in continuation to my previous blog(GCP Hybrid Networking Patterns — Part 1) on GCP Hybrid Networking Patterns. In this blog, I would cover about the Hybrid Connectivity to Multiple VPC networks(or Shared VPC networks) on Google Cloud Platform.

Hybrid Connectivity to Multiple VPC networks(or Shared VPC networks)

1. Interconnect and HA-VPN to On-premises

The same principles apply to Interconnect & HA-VPN to On Premises to Hybrid connectivity to Multiple VPC Networks(or Shared VPC networks), that applies to Hybrid Connectivity to Single VPC(or Shared VPC). I have covered that in detail in my previous Blog(GCP Hybrid Networking Patterns — Part 1).



2. Hybrid DNS Option 1(Cloud DNS Forwarding and Peering)

Overview

Let's consider a hybrid-DNS use case where on-premises DNS servers are authoritative for on-premises DNS zones, and Cloud DNS is authoritative for GCP zones.

DNS queries from GCP to on-premises are sourced from the IP range 35.199.192.0/19 — which is the same for all VPC networks. In an architecture with multiple Shared VPC networks, this means only one Shared VPC can forward and receive DNS queries to and from on-premises. In this example, ***Prod Shared VPC*** is

designated as the network used to send and receive DNS queries between GCP and on-premises.

1) On-premises DNS

Configure your on-premises DNS servers to be authoritative for on-premises DNS zones. Configure DNS forwarding (for GCP DNS names) targeting the Cloud DNS inbound forwarding IP address, which is created via the Inbound Server Policy in the ***Prod Shared VPC***. This allows on-premises network to resolve GCP DNS names via the ***Prod Shared VPC***.

2) Prod Shared VPC — DNS Egress Proxy

Advertise the Google DNS Egress Proxy range 35.199.192.0/19 to the on-premises network via the cloud routers. Outbound DNS requests from Google to on-premises are sourced from this IP address range.

3) Prod Host Project — Cloud DNS

a. Configure an Inbound Server Policy for inbound DNS requests from on-premises.

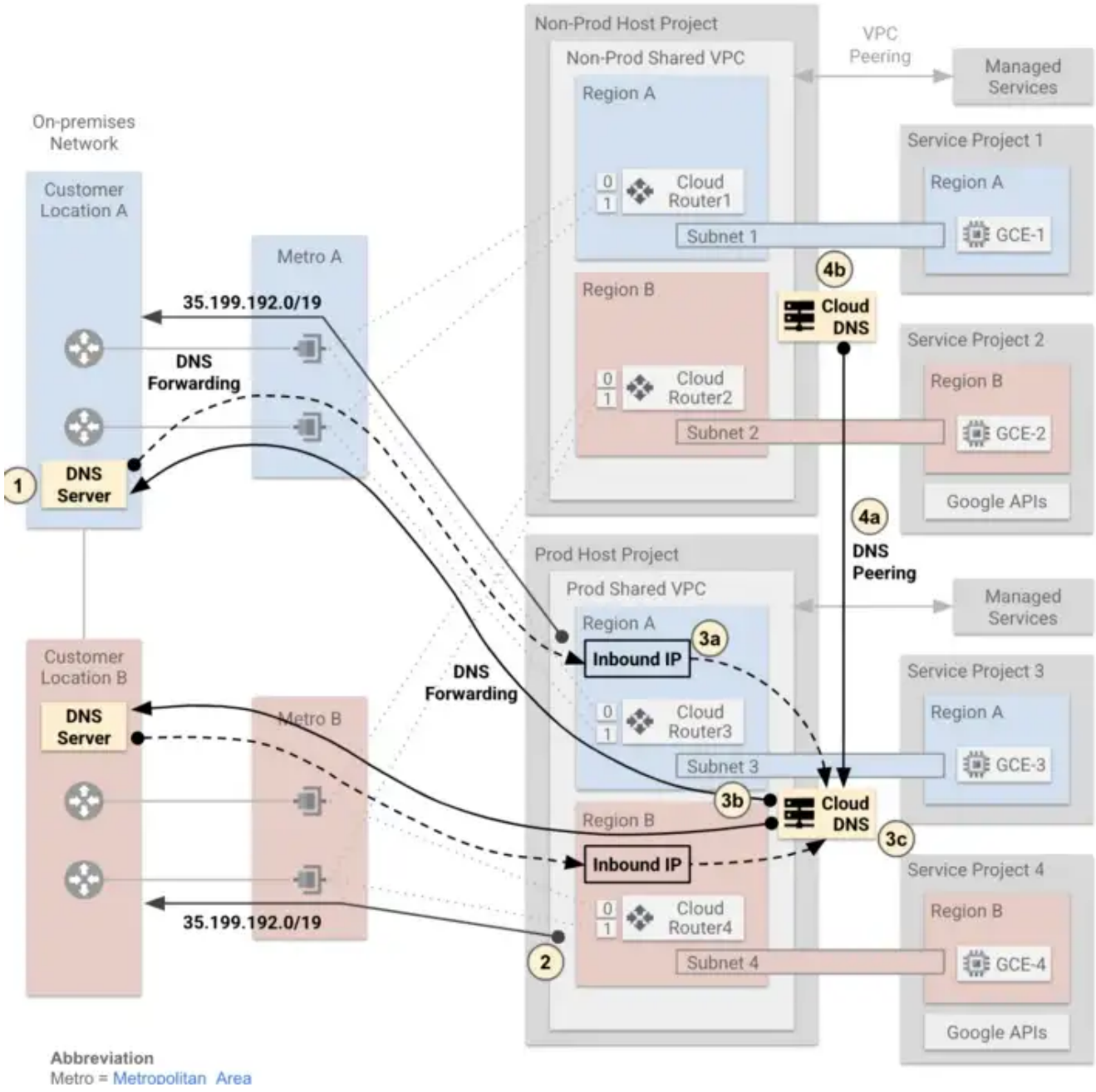
b. Configure Cloud DNS Forwarding Zone (for on-premises DNS names) targeting on-premises DNS resolvers.

c. Configure Prod DNS Private Zones in the **Prod Host Project** and attach **Prod Shared VPC** and **Non-Prod Shared VPC** to the zone. This allows hosts (on-premises and all service projects) to resolve the Prod DNS names.

4) Non-Prod Host Project — Cloud DNS

a. Configure DNS Peering Zone for on-premises DNS names targeting the ***Prod Shared VPC*** as the peer network. This allows Non-Prod resources to resolve on-premises DNS names.

b. Configure Non-Prod DNS private zones in the ***Non-Prod Host Project*** and attach ***Non-Prod Shared VPC*** and ***Prod Shared VPC*** to the zone. This allows hosts (on-premises and all service projects) to resolve the Non-Prod DNS names.



3. Hybrid DNS Option 2 (Custom DNS Forwarding)

Overview

In a hybrid environment, DNS resolution can be done in GCP or on-premises. Let's consider a use case where on-premises DNS servers are authoritative for on-premises DNS zones, and Cloud DNS is authoritative for GCP zones. This scenario assumes that an organization has requirements to deploy custom DNS servers in GCP, which are synchronized with their on-premises DNS servers. In GCP, the custom DNS servers can be deployed on VM instances behind an internal load balancer for high availability.

DNS queries from GCP to on-premises are sourced from the IP addresses of the custom DNS servers in the Shared VPC networks.

1) On-premises DNS

Configure your on-premises DNS servers to be authoritative for on-premises DNS zones. Configure DNS forwarding (for GCP DNS names) by targeting the custom DNS servers in the Shared VPC. This allows on-premises network to resolve GCP DNS names.

2) Host Project — Cloud DNS

Configure Cloud DNS Forwarding Zone (for on-premises DNS names) targeting the custom DNS server in the ***Prod Shared VPC***. Configure ingress firewall rule to allow DNS traffic from Cloud DNS egress proxies (35.199.192.0/19) into the custom DNS server instances. This allows DNS resolution for on-premises DNS names to be forwarded from Cloud DNS (via the egress proxies) to the custom DNS servers; and then forwarded by the custom DNS servers onwards to the on-premises DNS servers.

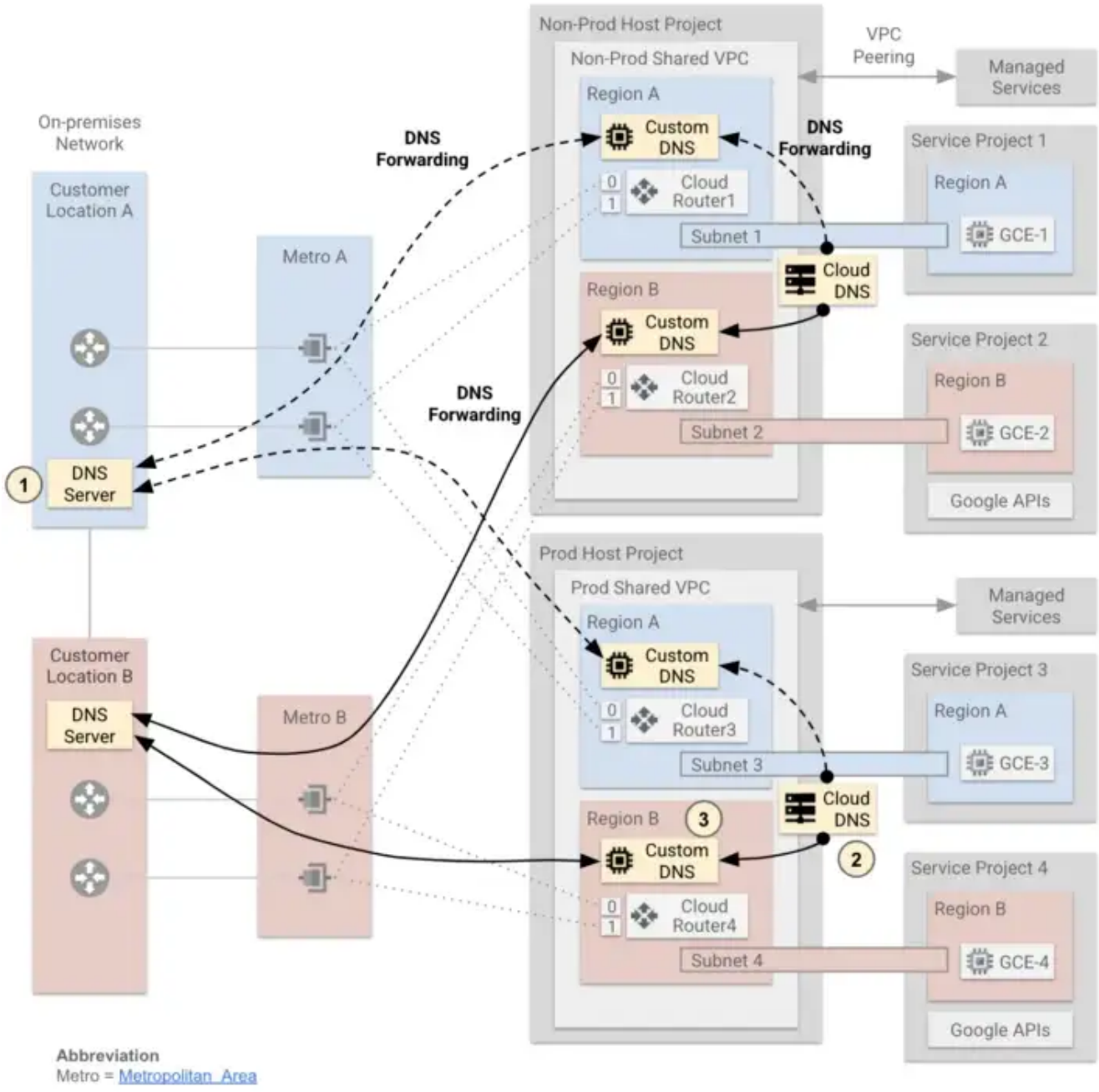
Configure the Cloud DNS Private Zones for the host projects.

Repeat the above configuration for the ***Non-Prod Shared VPC***.

3) Custom DNS Servers — DNS Forwarding

Inside the custom DNS server, configure DNS forwarding (for on-premises DNS names) targeting the on-premises DNS resolvers. This allows DNS queries for on-premises DNS names (received from the Cloud DNS egress proxies) to be forwarded to the on-premises DNS servers.

Inside the custom DNS server, configure a root forwarding zone (“.”) targeting the instance metadata server (169.254.169.254). This allows DNS queries for GCP DNS names (received from on-premises) to be resolved by the customer DNS servers via their instance metadata server.



4. Private Service Connect (PSC) for Google APIs(Access to all Supported APIs and Services)

Overview

You can use Private Service Connect (PSC) to access all supported Google APIs and services from Google Compute Engine (GCE) hosts and on-premises hosts; using the internal IP address of a PSC endpoint in a Shared VPC. In a hybrid connectivity scenario with multiple VPC (or Shared VPC) networks, on-premises hosts can access Google APIs and services through PSC endpoints via any of the VPC (or Shared VPC) network.

The following steps focus on API access via PSC endpoint in *Prod Shared VPC* (10.2.2.2). The same applies to API access via the PSC endpoint in the *Non-Prod*

Shared VPC (10.1.1.1).

Create a PSC Endpoint

1. Choose a PSC endpoint address (e.g. 10.2.2.2) and create a PSC endpoint in ***Prod Shared VPC*** with a target of **"all-apis"**- which gives access to all supported Google APIs and services. Service Directory automatically creates a DNS record (with DNS name of **p.googleapis.com**) linked to the PSC endpoint IP address.

Access from Google Compute Engine (GCE) Hosts

GCE hosts can access all supported Google APIs and services via the PSC endpoint in ***Prod Shared VPC.***

2. Enable Private Google Access on all subnets with compute instances that require access to Google APIs via PSC.

3. If your GCE clients can use custom DNS names (e.g. **storage-xyz.p.**googlepapis.com), you can use the auto-created **p.googleapis.com** DNS name.

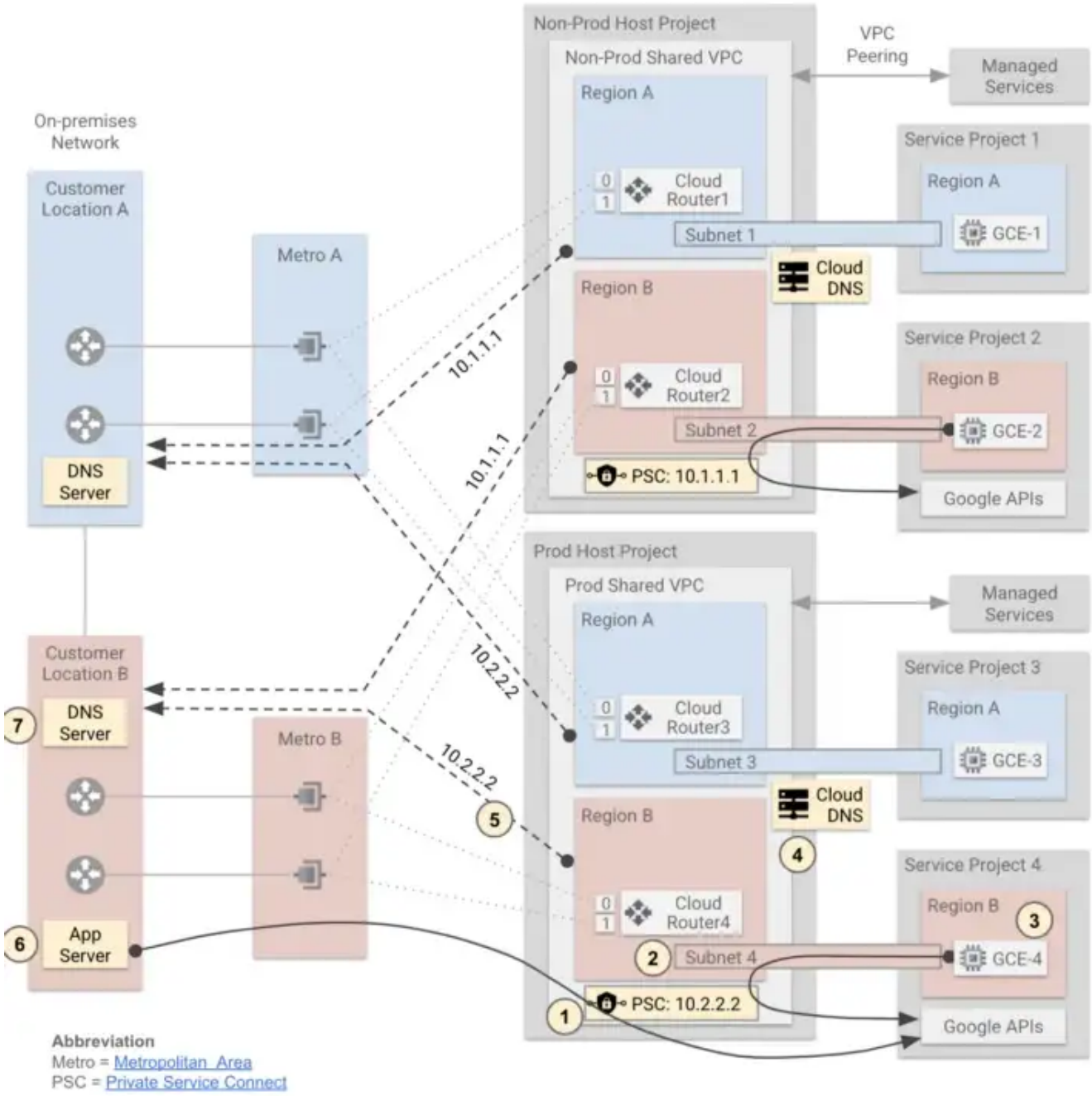
4. If your GCE clients cannot use custom DNS names, you can create Cloud DNS records using the default DNS names (e.g **storage**.googleapis.com).

Access from On-premises Hosts

On-premises hosts can access all supported Google APIs and services via the PSC endpoint in the ***Prod Shared VPC***. 5. On the cloud router, advertise the PSC endpoint address to the on-premises network.

6. If your on-premises clients can use custom DNS names (e.g. **storage-xyz.p.**googlepapis.com), you can create A records mapping the custom DNS names to the PSC endpoint address.

7. If your on-premises clients cannot use custom DNS names, you can create A records mapping the default DNS names (e.g **storage.**googleapis.com) to the PSC endpoint address.



5. Private Service Connect (PSC) for Google APIs (Access to APIs and Services supported under VPC Service Control)

Overview

You can use Private Service Connect (PSC) to access all supported secure Google APIs and services from Google Compute Engine (GCE) hosts and on-premises hosts; using the internal IP address of a PSC endpoint in a Shared VPC. In a hybrid connectivity scenario with multiple VPC (or Shared VPC) networks, on-premises hosts can access Google APIs and services through PSC endpoints via any of the VPC (or Shared VPC) network.

The following steps focus on API access via PSC endpoint in ***Prod Shared VPC*** (10.2.2.2). The same applies to API access via the PSC endpoint in the ***Non-Prod Shared VPC*** (10.1.1.1).

Create a PSC Endpoint

1. Choose a PSC endpoint address (e.g. 10.2.2.2) and create a PSC endpoint in the ***Prod Shared VPC*** with a target of **"vpc-sc"**- which gives access to Google APIs and services that are supported under VPC Service Control. Service Directory automatically creates a DNS record (with DNS name of **p.googleapis.com**) linked to the PSC endpoint IP address.

Access from Google Compute Engine (GCE) Hosts

GCE hosts can access (VPC service control) supported Google APIs and services via the PSC endpoint in the ***Prod Shared VPC***.

2. Enable Private Google Access on all subnets with compute instances that require access to Google APIs via PSC.

3. If your GCE clients can use custom DNS names (e.g. **storage-xyz.p.**googlepapis.com), you can use the auto-created **p.googleapis.com** DNS name.

4. If your GCE clients cannot use custom DNS names, you can create Cloud DNS records using the default DNS names (e.g **storage.**googleapis.com).

Access from On-premises Hosts

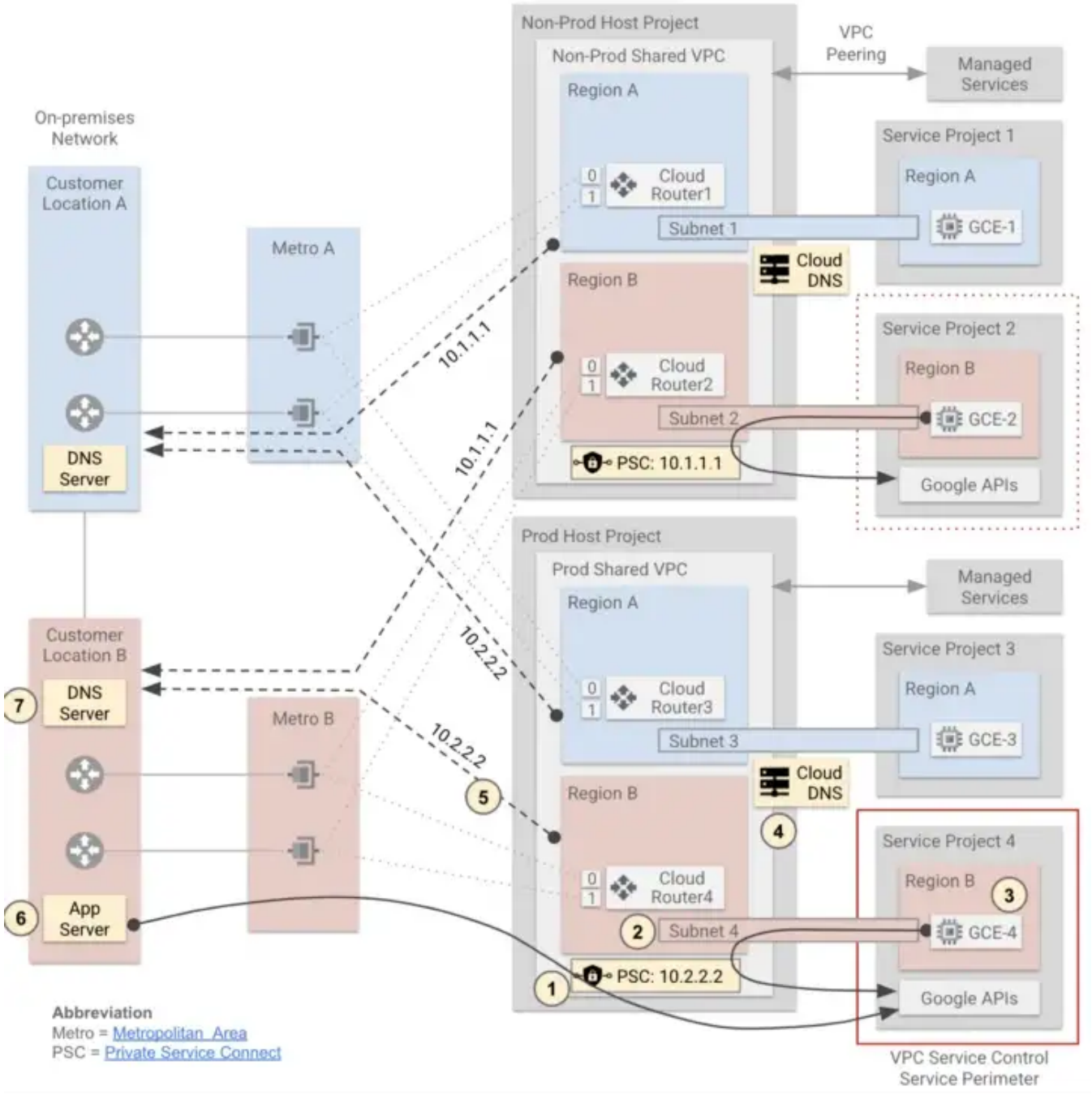
On-premises hosts can access all secure Google APIs and services via the PSC endpoint in the ***Prod Shared VPC***.

5. On the cloud router, advertise the PSC endpoint address to the on-premises network.

6. If your on-premises clients can use custom DNS names (e.g. **storage-xyz.p.**googlepapis.com), you can create A records mapping the custom DNS names to the PSC endpoint address.

7. If your on-premises clients cannot use custom DNS names, you can create A records mapping the default DNS names (e.g **storage.**googleapis.com) to the PSC

endpoint address.



6. VPC Service Control

Overview

VPC Service Control uses Ingress and egress rules to control access to and from a perimeter. The rules specify the direction of allowed access to and from different identities and resources.

Let's consider a specific use case where we require access to a protected service in *Service Project 4* via the *Prod Shared VPC*.

Below is an description of VPC service control action for our specific scenario.

1) Perimeter around Service Projects

In this example, we have a perimeter around ***Service Project 4*** and the Google APIs and services to be protected. Similarly, there's a perimeter around ***Service Project 4***.

2) API access from GCE Hosts

A GCE client can access secured APIs through a PSC endpoint in the Shared VPC. Let's consider the perimeter around ***Service Project 4***. The network interface of the compute instance ***GCE-4*** is in the ***Prod Shared VPC*** of ***Prod Host Project***. API calls from ***GCE-4*** instance to a service (e.g. storage.googleapis.com) in ***Service Project 4***, appear to originate from ***Prod Host Project*** — where the instance interface and PSC endpoint are located.

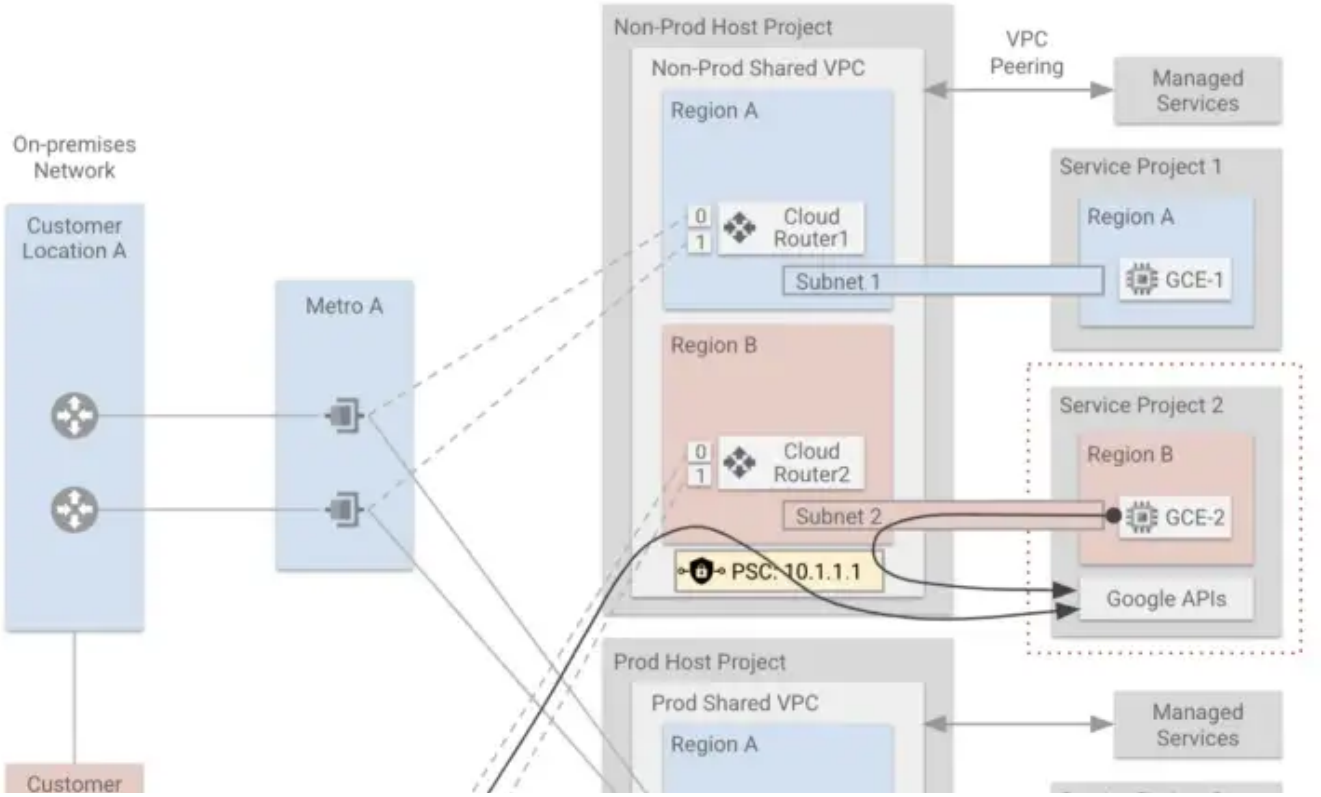
3) Ingress Rule — Prod Host Project into Perimeter

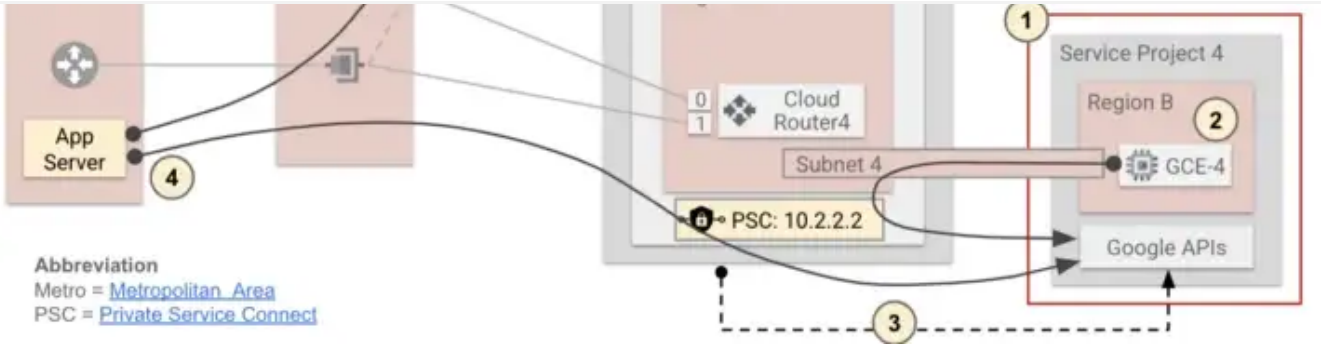
Configure an ingress rule that allows Google API calls from ***Prod Host Project*** to the protected services in ***Service Project 4*** perimeter. This rule allows API calls from the GCE instances (e.g. ***GCE-4***) into the perimeter.

4) API access for on-premises hosts

On-premises hosts can access secured APIs in ***Service Project 4*** via the PSC endpoint in the ***Prod Shared VPC***. API calls from on-premises to services in ***Service Project 4*** appear to originate from ***Prod Host Project*** — where the Interconnect (or VPN) and the PSC endpoint are located.

The ingress rule (in step 3) allows API calls from on-premises to ***Service Project 4*** perimeter via ***Prod Host Project***.





Hybrid Connectivity Shared Vpc Vpc Service Control Google Cloud Platform Infrastructure