Networking in GCP

Memi Lavi
www.memilavi.com

Networking

- All aspects of networking in GCP
- Deals with resources’ network connections, firewalls, etc.
- Might sound boring and not very important, but…

Networking is the foundation of cloud security

ReadIt!

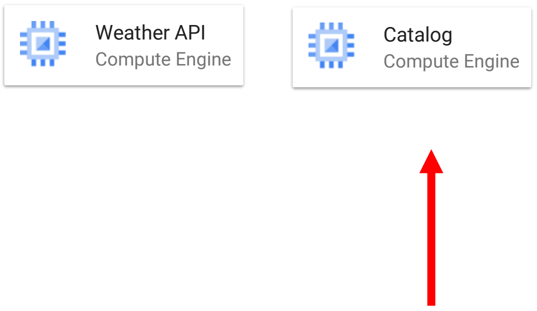
Cloud Architecture

A Word of Caution:

NEVER

leave a VM open to the internet this way

We will learn later on what should be done



Weather API
Compute Engine

Catalog
Compute Engine

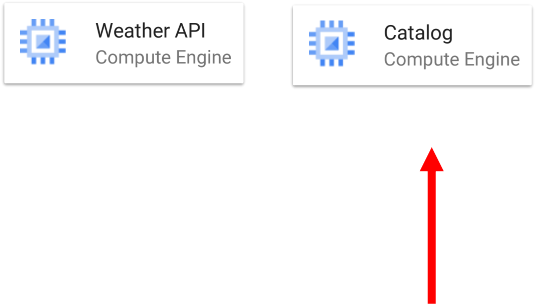
- Directly accessible from the internet
- Can be SSHed from anywhere

ReadIt!

Cloud Architecture

Two main threats:

- Brute force attacks on port 22 (SSH)
- No line of defense in front of the VM web server



- Directly accessible from the internet
- Can be SSHed from anywhere

Networking knowledge is what makes a good cloud architect – an amazing cloud architect

Networking

- We’ll talk about 4 networking-related cloud services:

VPC

Subnets

Firewall

Load Balancers

VPC

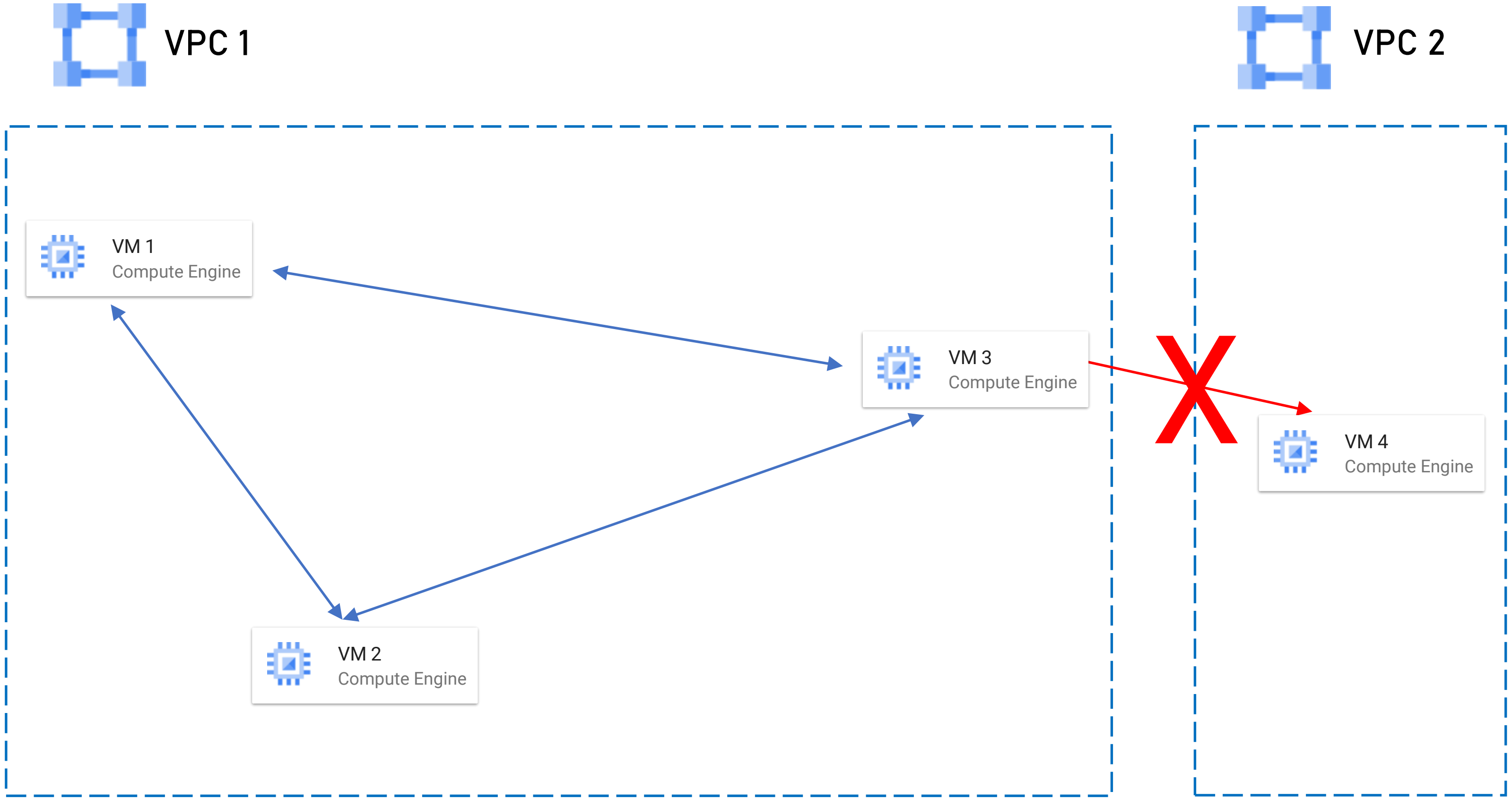
- Virtual Private Cloud
- A network in which you can deploy cloud resources
- Many cloud resources are deployed within VPC
    - VMs, Load balancers
- Many cloud services can be deployed in VPC but are not by default
    - App Engine, Cloud SQL and more

VPC

- “Virtual” as in “based on physical network and logically separated from other virtual networks”

VPC

- Resources in VPC can communicate with each other by default



… but not with

resources in other

VPCs

(Unless with peering)

VPC

- Think of it as your organization’s private network
- Virtual Private Cloud
- Other organizations’ networks cannot communicate with

  your network

VPC Pricing

- VPCs are free

- Limit of 15 VPCs per project

Characteristics of VPC

- Global
  - Resources in different regions can communicate between them
- Automatically created per project
- Can be connected via Peering
- Segmented using Subnets
- Protected using firewall rules

Security and VPC

- The most important thing to think about when designing networking:

How to limit access to the resources in the VPC so that risk is minimized

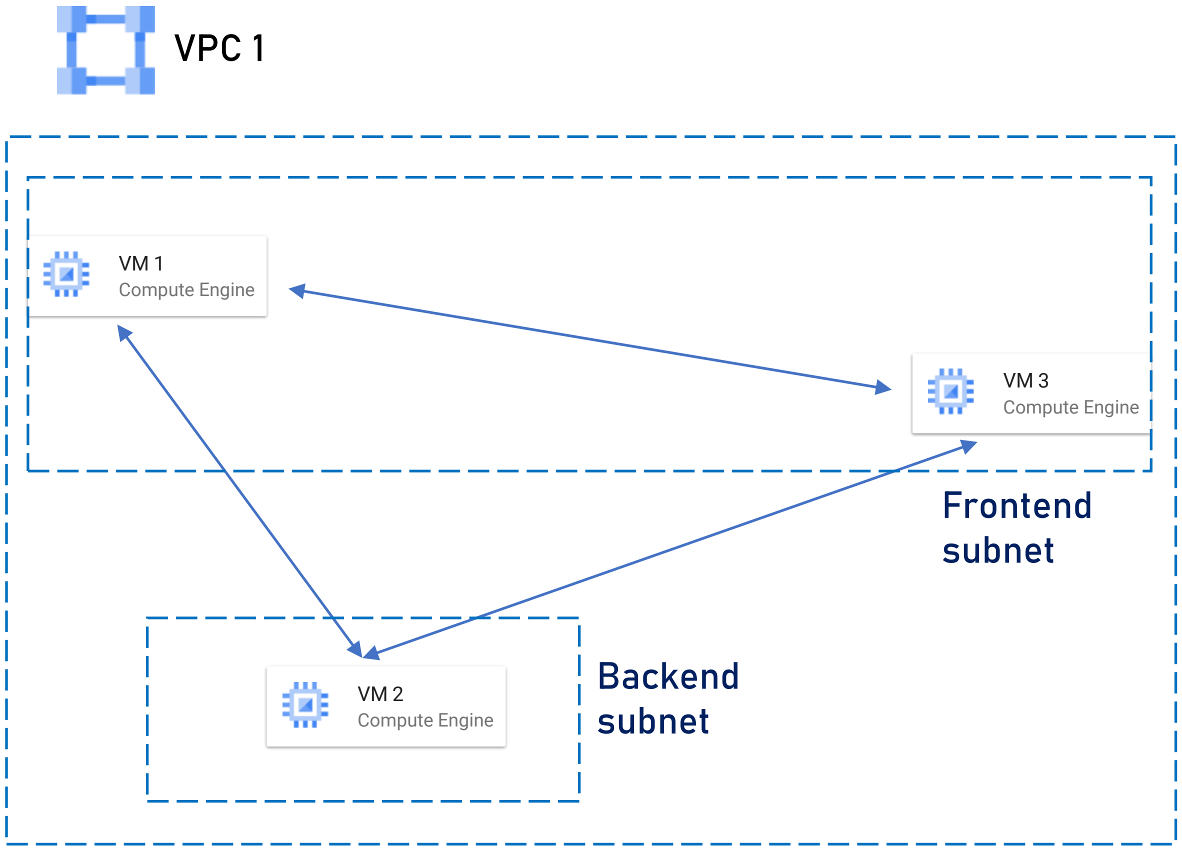
Subnet

- A logical segment in the VPC
- Has its own IP range
- Used as a logical group of resources in the VPC
- Is a must. Resources must be placed in a Subnet, cannot be placed directly in a VPC

Subnet

- Resources in a subnet can talk to resources in other subnets in the same VPC*

*By default,
can be customized



Subnet Pricing

- Subnets are free

Subnet Creation Mode

- New VPCs are created using one of two modes:

Auto mode

- One subnet from each region is automatically created
- Subnets are automatically assigned IP range
- Subnets' ranges do not overlap
- Ranges fit within the 10.128.0.0/9 CIDR block *
- When new regions are added, new subnets are automatically added
- More subnets can be added manually
- Default for new projects (can be disabled)

Custom mode

- No subnets are created automatically
- Full control on subnets and IP ranges

* We'll discuss CIDR blocks in the next lecture

Subnet Creation Mode

- How to select between the creation modes?

Auto mode

- Useful if you want to have subnet in every region
- IP ranges do not overlap with other services (ie. VPN)

Preferred for production scenarios

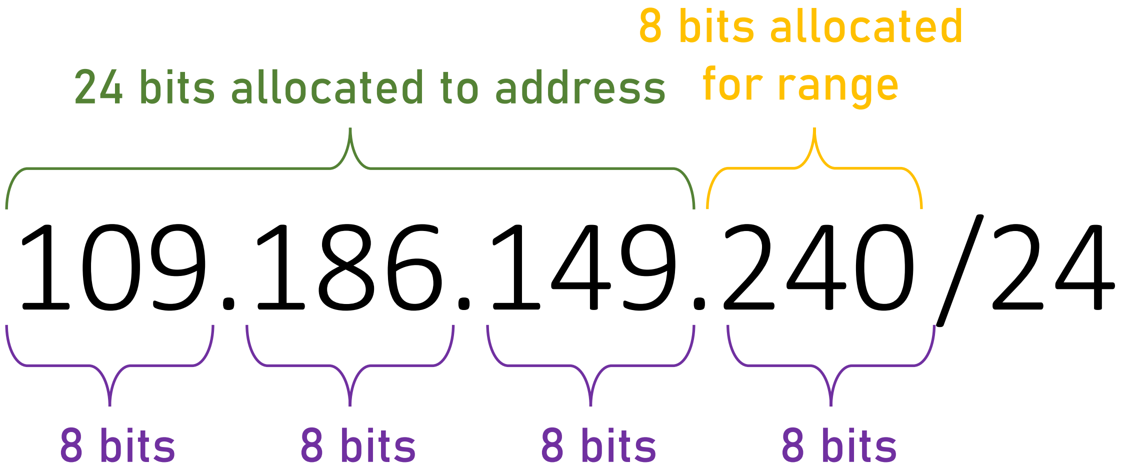
Custom mode

- No need for subnets in every region
- IP ranges overlap with other services
- You want complete control over subnets and IP ranges
- You plan to connect the VPC to other VPC and IP ranges will overlap

CIDR Notation

- Classless Inter-Domain Routing
- A method for representing an IP Range
- Composed of an address in the range and a number between 0 and 32
- The number indicates the number of bits that are allocated to the address. The smaller the number – the larger the range

CIDR Notation Example #1



8 bits allocated for range

24 bits allocated to address

109.186.149.240/24

8 bits

8 bits

8 bits

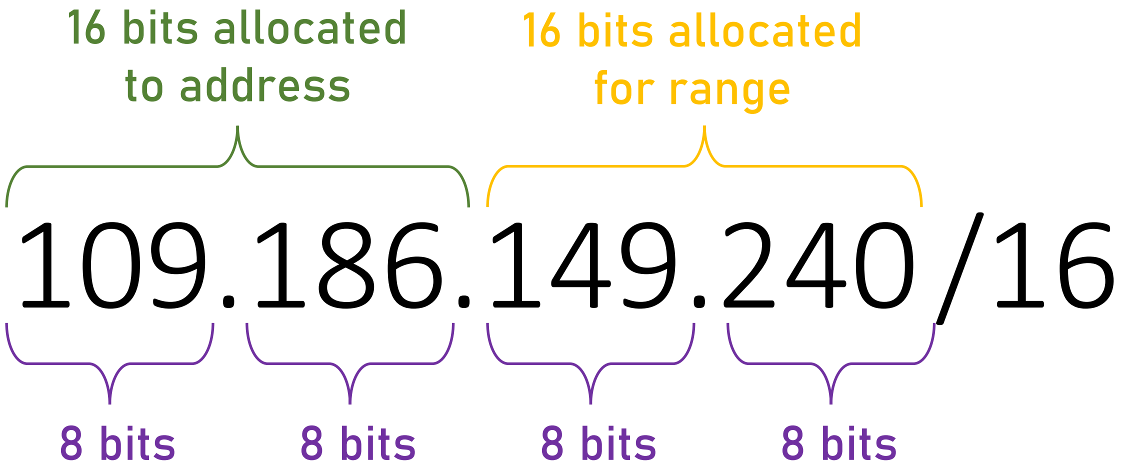
8 bits

109.186.149.000 – 109.186.149.255

256 Addresses

Bits refresher:
00000000 = 0
11111111=255

CIDR Notation Example #2



16 bits allocated to address

16 bits allocated for range

109.186.149.240/16

8 bits

8 bits

8 bits

8 bits

109.186.000.000 – 109.186.255.255

65,536 Addresses

Probably way too big…

Bits refresher:
00000000 = 0
11111111=255

CIDR Notation Example #3

20 bits allocated to address

12 bits allocated for range

149 Dec = 1001 ~~0101~~ Bin

109.186.149.240/20

1001 0000 Bin = 144 Dec

8 bits 8 bits 8 bits 8 bits

109.186.144.000 – 109.186.159.255

4,096 Addresses

Bits refresher:
00000000 = 0
11111111=255

CIDR Notation

- The good news:

You don’t have to remember!

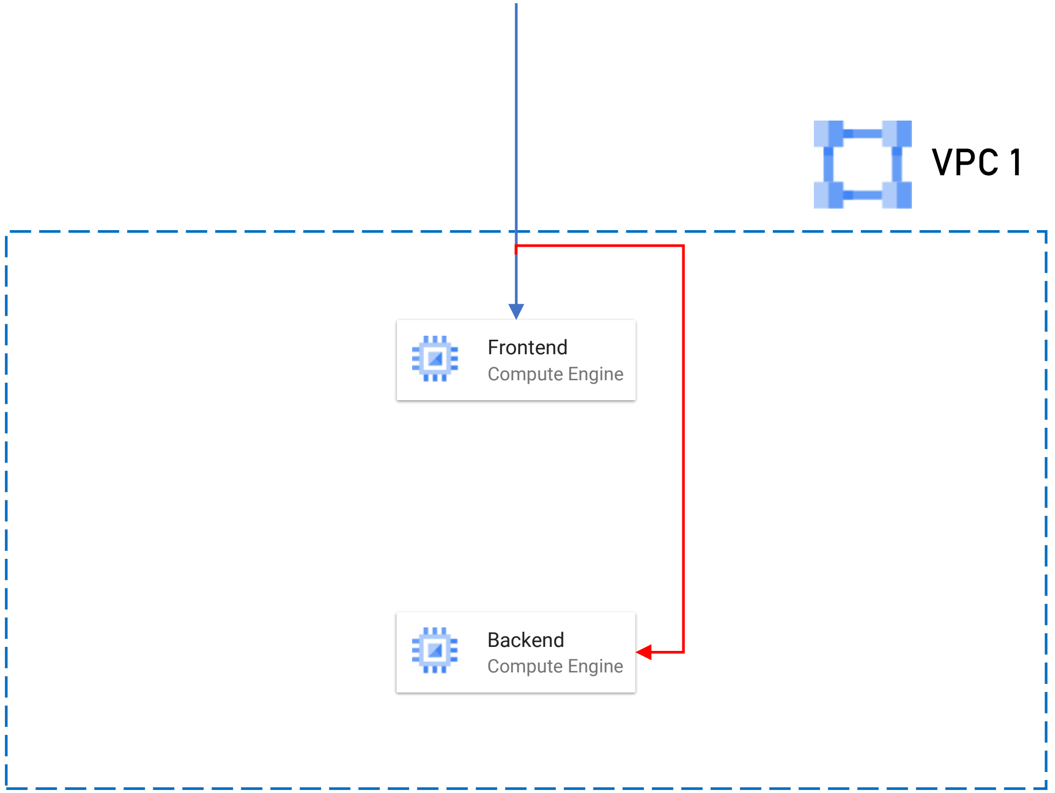
- A lot of CIDR calculators
  - ie. https://www.ipaddressguide.com/cidr

VPC Network Peering

- Sometimes, to increase security, we want to place some resources in a completely different VPC
  - Not just Subnet!
- Examples:
  - Separate systems
  - System layers
  - Sensitive databases

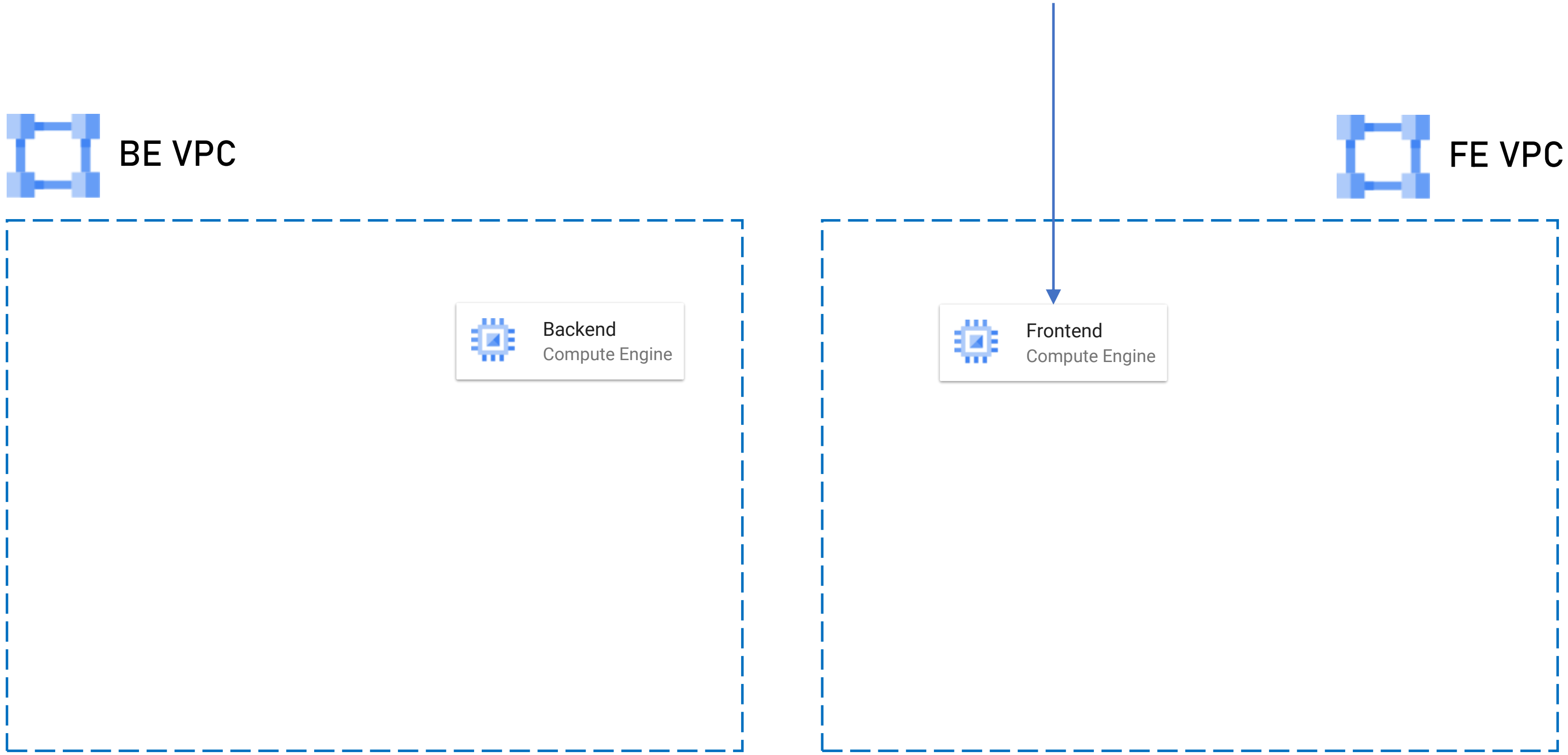
VPC Network Peering

- Main reasoning:
  - Not to place non-public resources in a VPC that has public access



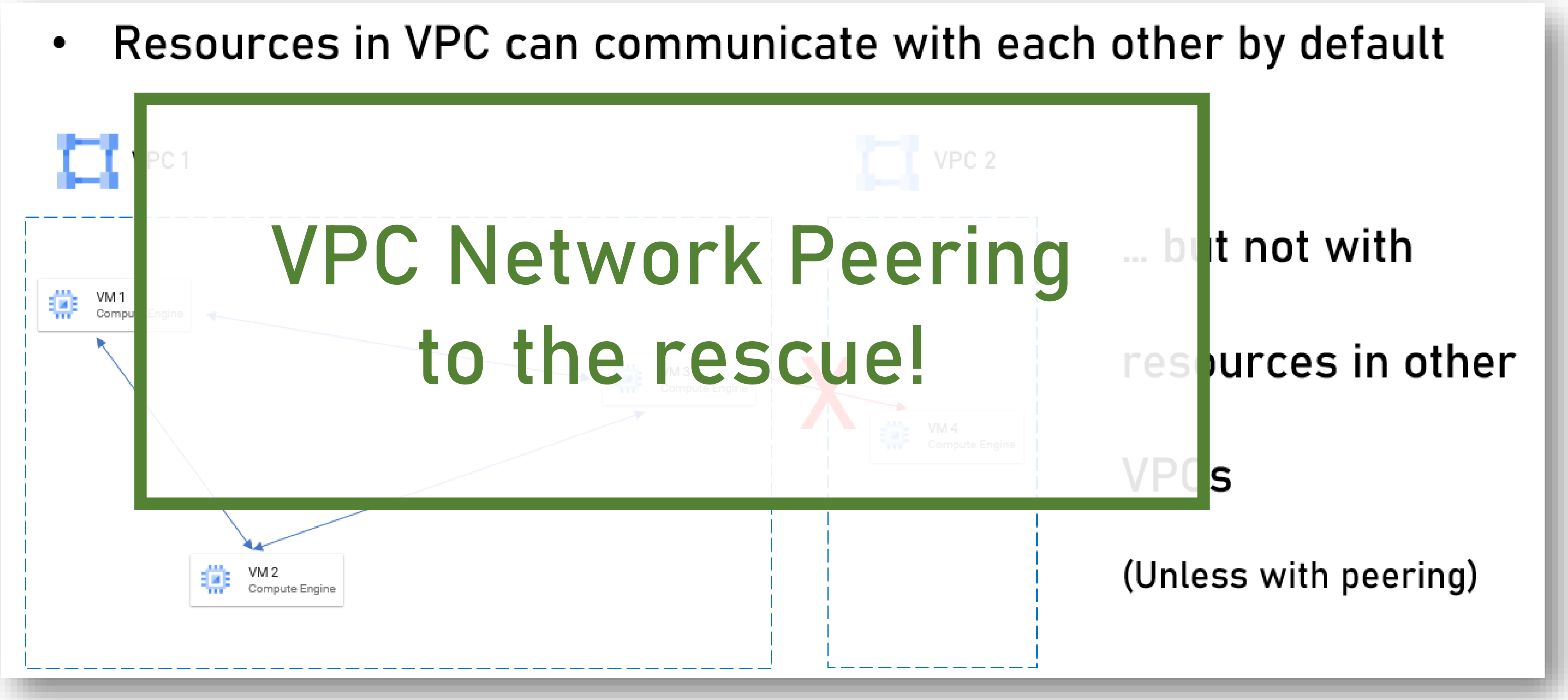
VPC Network Peering

- So…



VPC Network Peering

- But…



- Resources in VPC can communicate with each other by default

… but not with resources in other VPCs

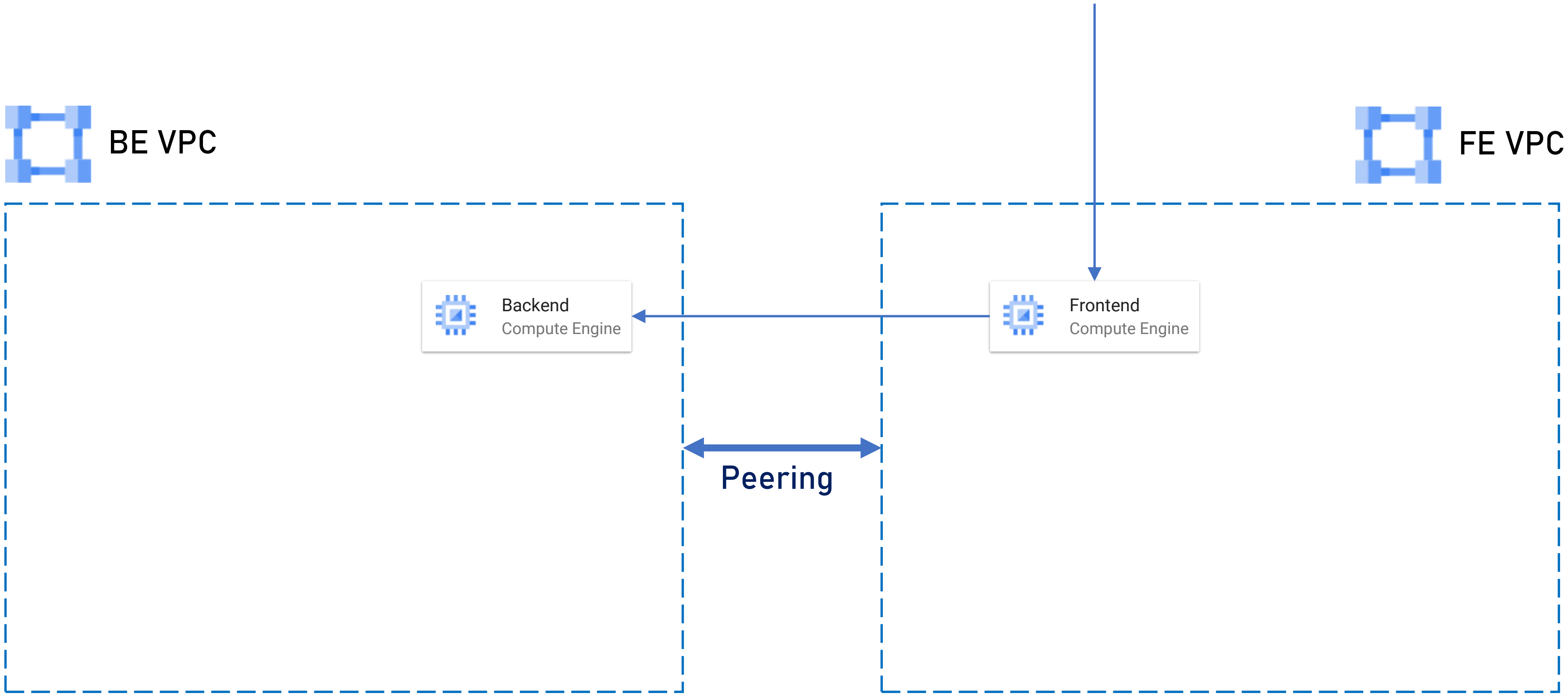
(Unless with peering)

VPC Network Peering to the rescue!

VPC Network Peering

- Allows two VPCs to connect to each other
- From the user's point of view it's a single VPC
- Make sure address spaces are not overlapped!
- Use Firewall Rules for protection
- Works across projects and organizations

VPC Network Peering



Firewall Rules

- By default, every new VPC blocks incoming traffic
  - Allows outgoing traffic
- Protects the instances in it
- In order to allow traffic use Firewall Rules

Firewall Rules

- Used to allow or deny traffic based on various parameters
- Traffic should match 5 tuples in order to be allowed:
  - Source IP
  - Destination IP
  - Source port
  - Destination port
  - Protocol

Firewall Rules

- Target and destinations can be specified also using tags and service accounts
  - We'll discuss both later in this course
- Default network allows access to ports 22, 3389 and the icmp protocol
- New VPCs block everything

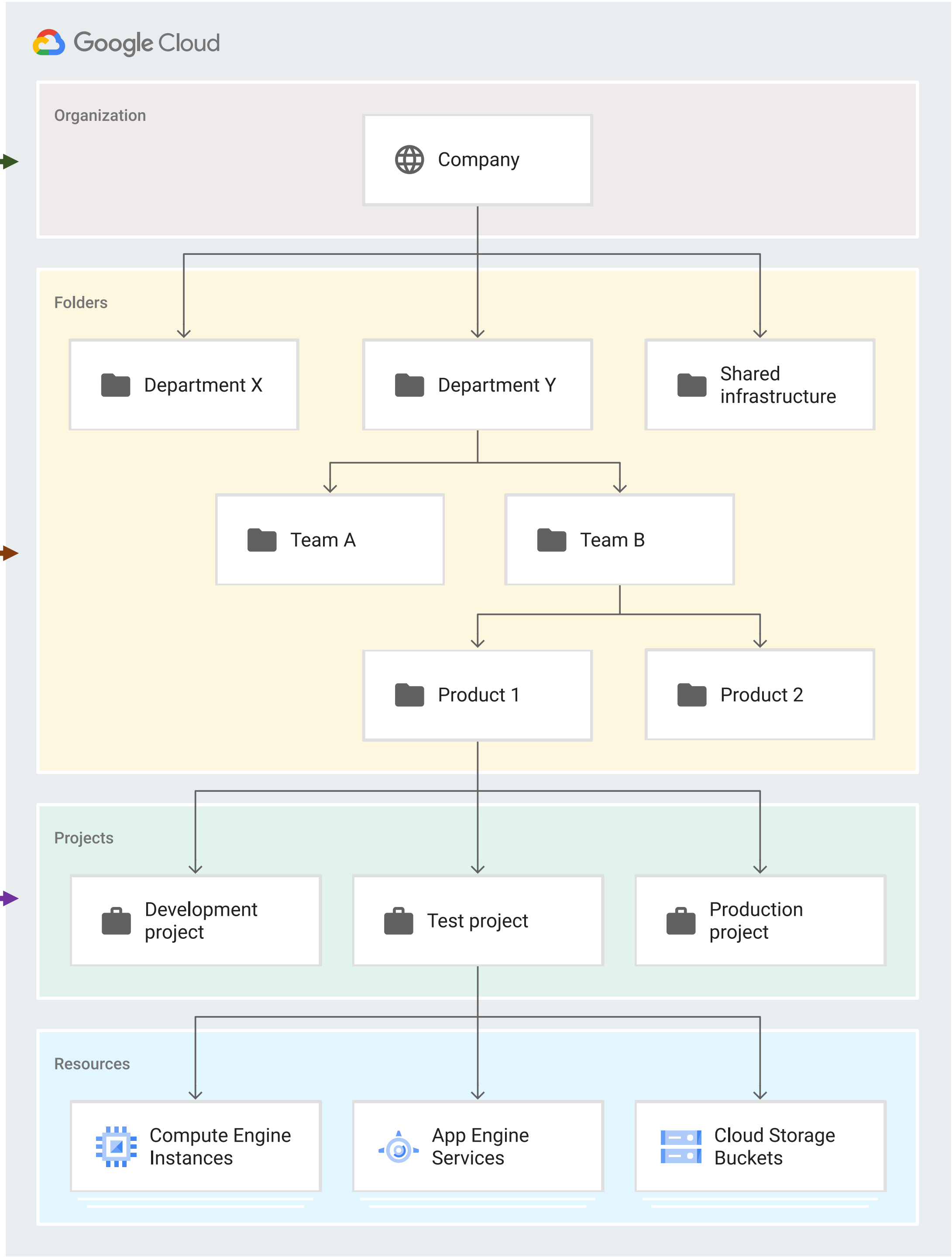
Shared VPC

- The VPCs we used so far are project VPC
- They are part of a single project
  - Called *Standalone VPC* in a *Standalone project*
- Project can be part of Organization

- Top level of the hierarchy
- Usually represents the company using the cloud
- Have access to all underlying resources

- Additional grouping mechanism
- Usually model legal entities, departments, teams etc.
- Can contain other folders
- Optional

- Contain the actual resources
- The most important level in the hierarchy
- Resources must be created in a project



Shared VPC

- Shared VPC allows connecting resources in multiple projects in the same organization
- Two types of projects in Shared VPC:

Host Project

Service Project

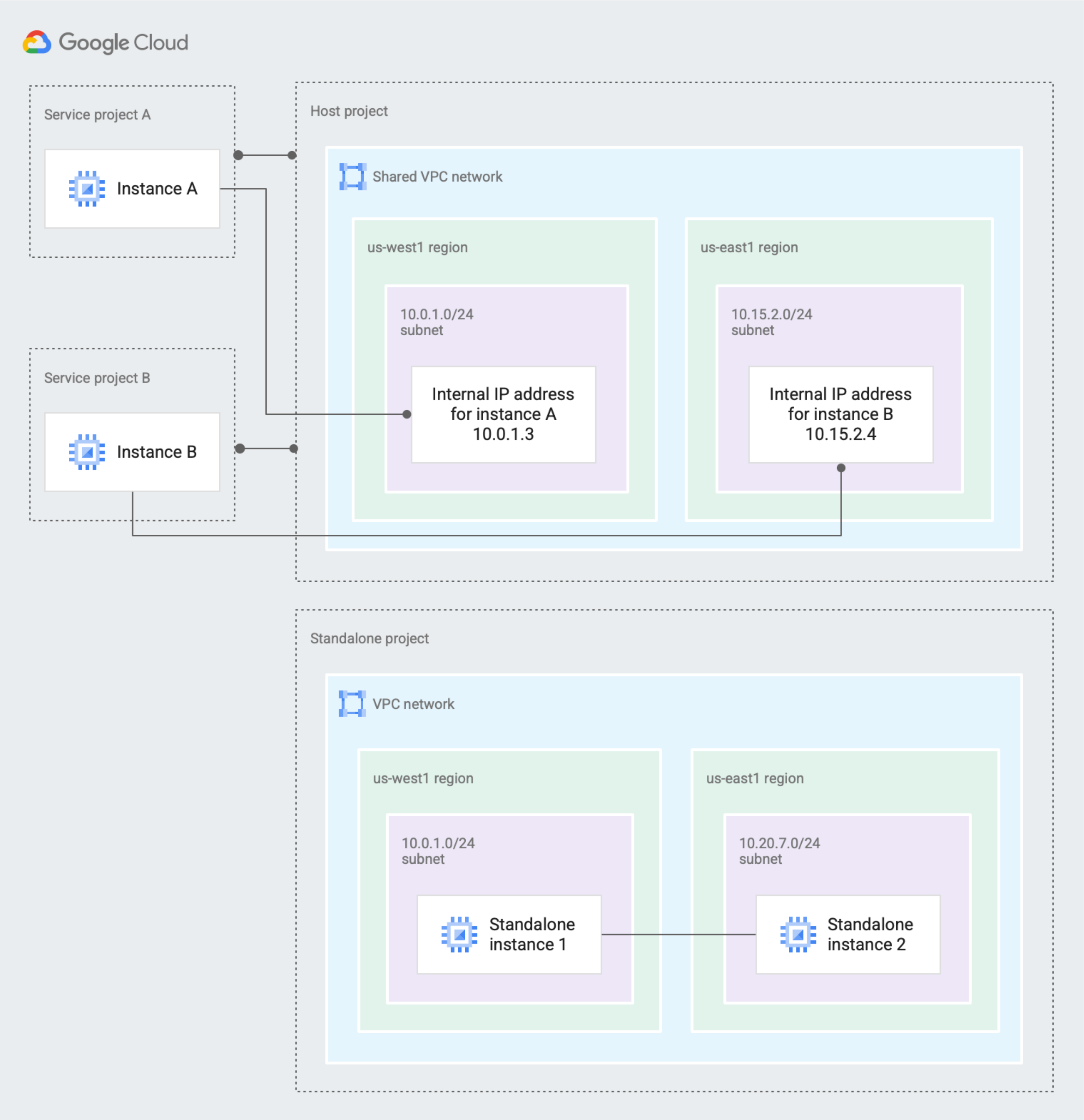
Shared VPC

Host Project

- The project where the VPC is created
- Requires special permissions

Service Project

- Attached to the host project
- Becomes connected to it through the Shared VPC
- Can be connected to a single host project only

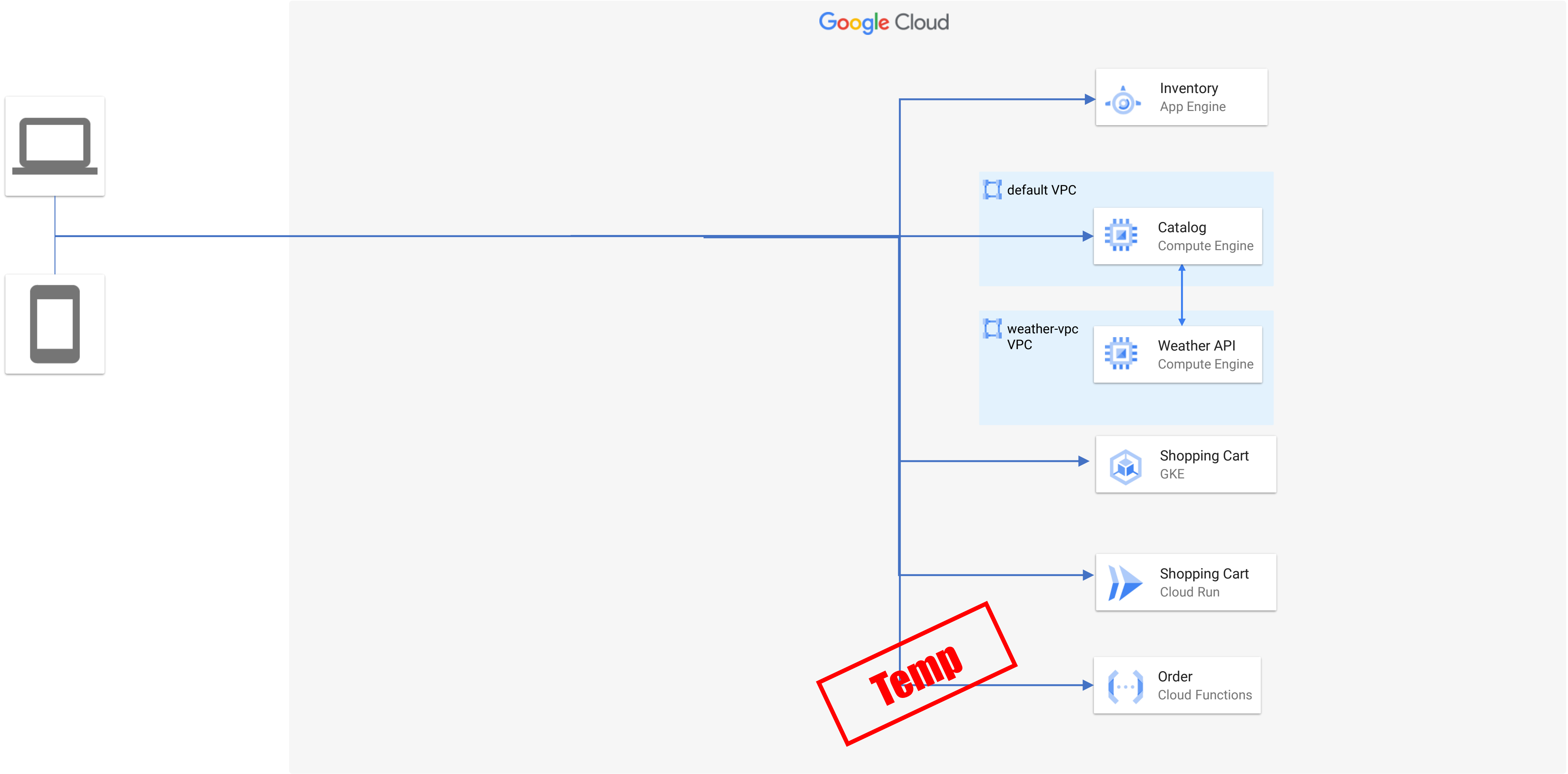

Google Cloud
Service project A
Instance A
Service project B
Instance B
Host project
Shared VPC network
us-west1 region
10.0.1.0/24
subnet
Internal IP address
for instance A
10.0.1.3
us-east1 region
10.15.2.0/24
subnet
Internal IP address
for instance B
10.15.2.4
Standalone project
VPC network
us-west1 region
10.0.1.0/24
subnet
Standalone
instance 1
us-east1 region
10.20.7.0/24
subnet
Standalone
instance 2

Shared VPC

- Can be created only when there's organization
- Free accounts do not have organization
- We cannot demonstrate it…

ReadIt!

Architecture: ReadIt Cloud System



Secure VM Access

- If you're using VMs – you need to be aware of its security
- The larger the attack surface – the greater the risk
- We want to minimize it as much as possible
- Leaving public IPs open is always a risk we want to avoid
- Not directly related to the app design but important nonetheless

Secure VM Access

- What can be done?

Firewall Rules

VPN

Jump Box

Firewall Rules

- Limit the SSH / RDP access only to source IPs that really need it
  - The default rule opens these ports to all IPs – change it!
- If VM does not need SSH / RDP – remove these rules
- In general – do not have unneeded firewall rules

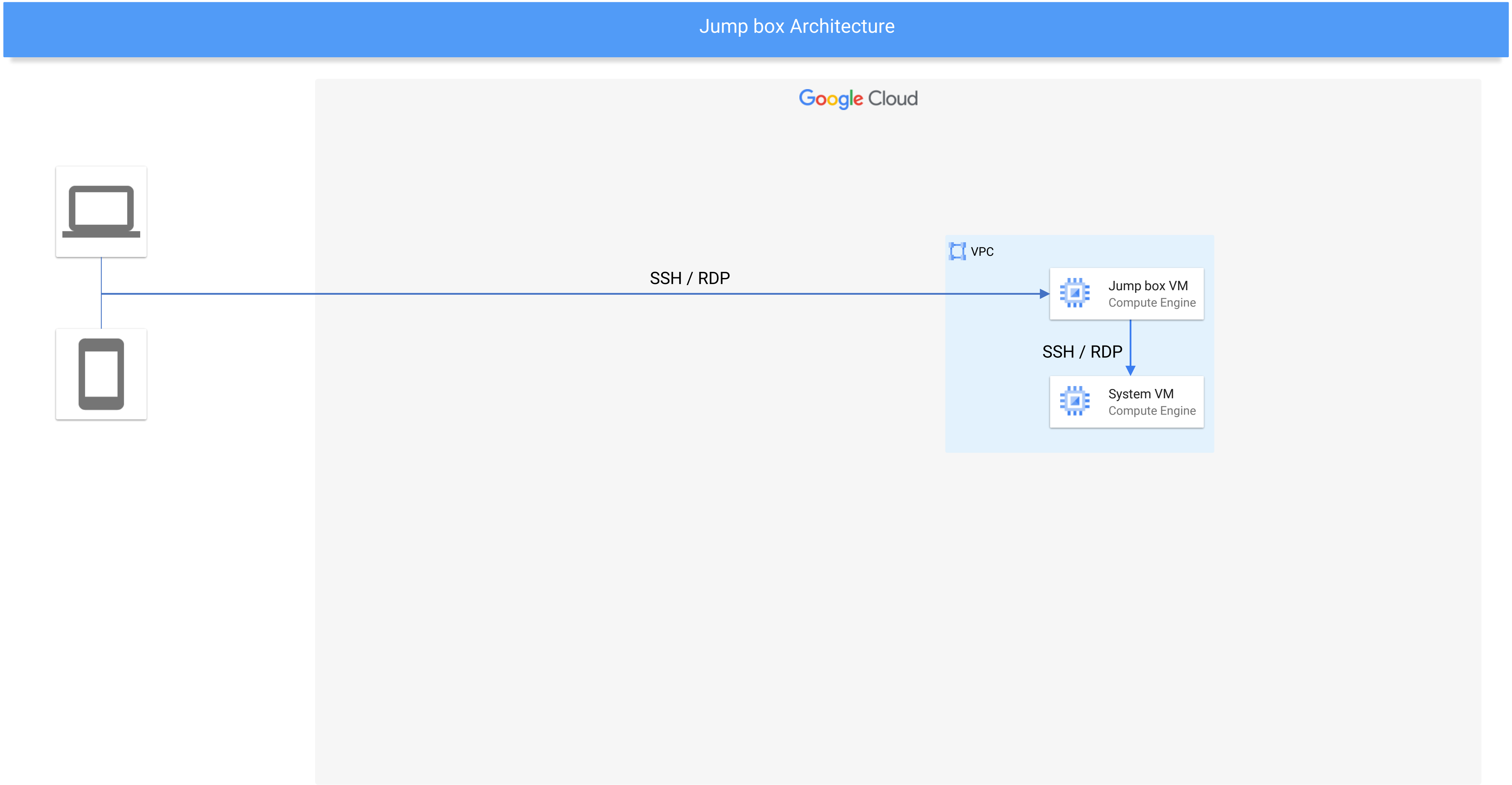
VPN

- A secure tunnel to the VPC
- Can be configured so that no one else can connect to the VPC
- Implemented using the Cloud VPN service in GCP
- Requires VPN software and license (not part of GCP)

Jump Box

- Create another VM in the VPC
- Allow external access (RDP/SSH) ONLY to this VM
- When need to access one of the other VMs – connect to this one and connect from it to the relevant VM
- Only one port and IP is open (still kind of a problem…)
- Cost: The additional VM (the Jump Box)

Jump box Architecture



Secure Access from On-Premises

- Sometimes there’s a need to connect the on-prem organizational network to the cloud
- Should be done securely
- Without exposing public IPs

Secure Access from On-Premises

- Two ways of doing that:

VPN

Interconnect

VPN

- A secure tunnel to the VPC
- Uses the public internet infrastructure
- Implemented using the Cloud VPN service in GCP
- Requires VPN software and license (not part of GCP)

Cloud Interconnect

- Direct physical connection between the organization network and Google's network
- Extremely performant
- Up to 100Gbit/s

Cloud Interconnect Types

Direct

- Dedicated physical connection
- Up to 100Gbig/s
- Costly
  - Can reach tens of thousands $ / month

Partner

- Uses partner infrastructure
- Cheaper
- Better if no need for the speed and security of dedicated connection

Private Access

- Some resources in the cloud are placed in VPC and some not

In VPC

- VM Instances
- GKE
- App Engine Flexible

Not in VPC

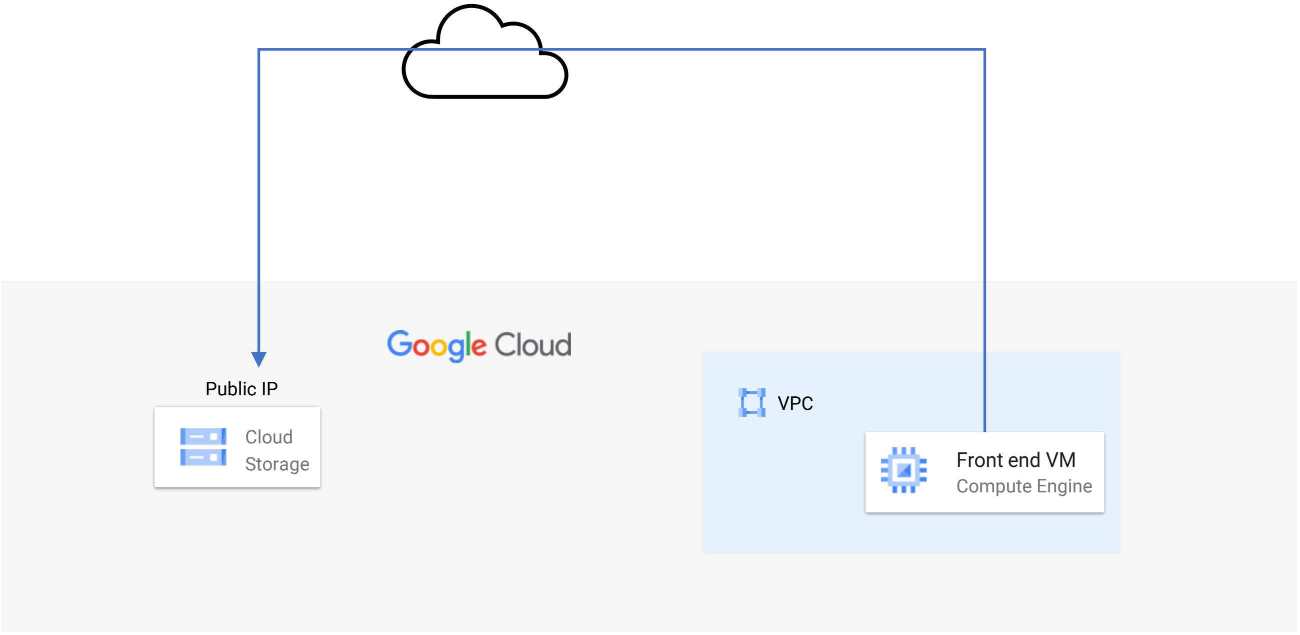
- App Engine Standard
- Cloud Functions
- Cloud Run
- Cloud Storage
- Cloud SQL
- BigTable
- Lots more…

Private Access

- By default, connecting from VPC resources to non-VPC resources is done using public address on the internet

Not secure:

1. Traffic goes through the internet
2. The Cloud Storage is open to the internet

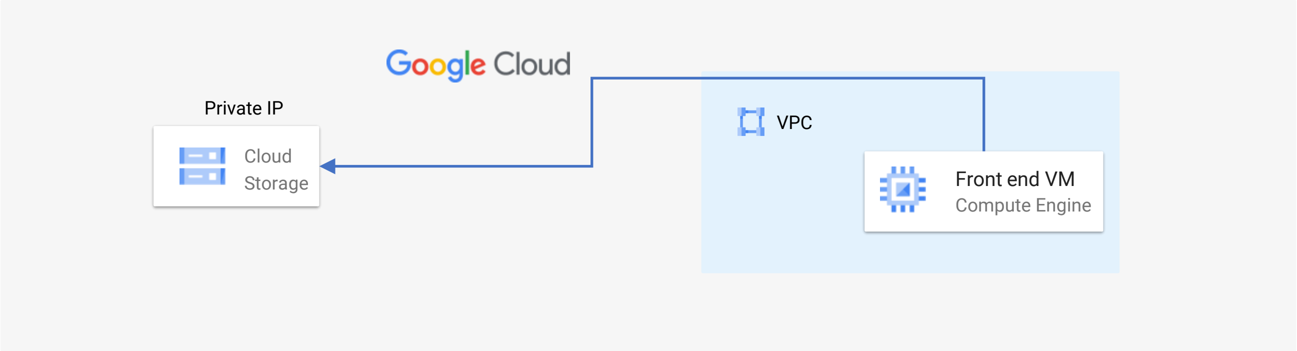


Private Access

- Private Access allows connecting from VPC to cloud resources using private IP without going through the internet

Secure:

1. Traffic stays in the cloud
2. The Cloud Storage is not open to the internet



Implementing Private Access

- Three types of Private Access implementations in GCP:

Private Google Access

Private Service Connect

Private Service Access

Private Google Access

- Easy to set-up
- Client VM should NOT have public IP address
    - Or Private Google Access has no effect
- Uses shared pool of private IPs determined by Google

Private Google Access

- Use when:
  - No public IP on the client VM
  - No fine-grained control is necessary (ie. you don’t care about the IP of the connected service)

Private Google Access

- Supported services:
  - All non-VPC services in GCP
  - Examples: Cloud Storage, App Engine, BigTable, Spanner etc.

Private Service Connect

- Complex to set-up
- Client VM can have public IP address
- Full control on IP, routing, etc.

Private Service Connect

- Use when:
  - Client VM has public IP
  - Fine-grained control is necessary (ie. setting the IP and DNS of connected service)

Private Service Connect

- Supported services:
  - All non-VPC services in GCP
  - Examples: Cloud Storage, App Engine, BigTable, Spanner etc.

Private Service Access

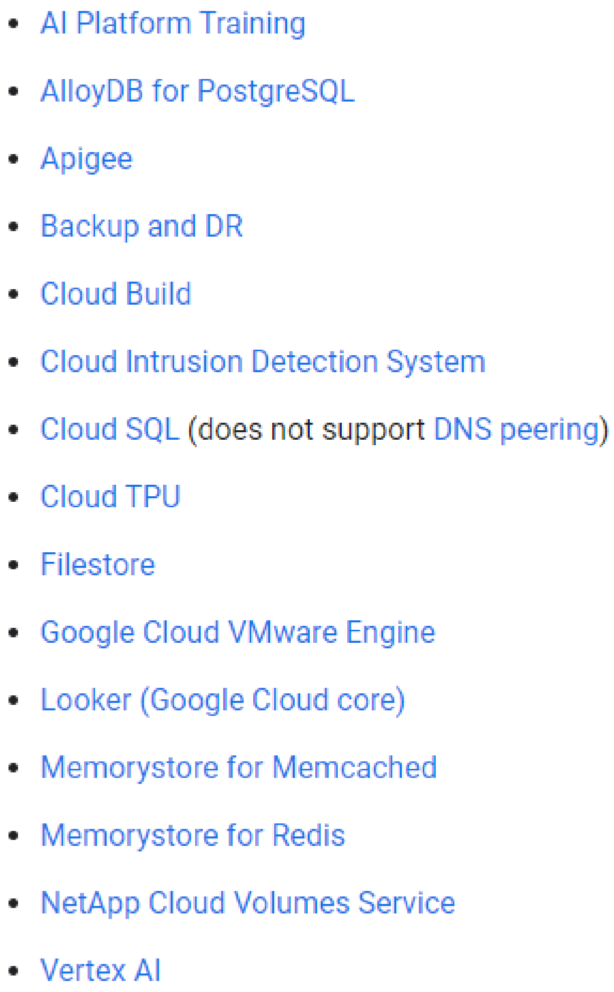
- Used to connect to services that are hosted in VPCs managed by Google or other services providers
- Requires pre-allocating IP range for connecting to the other VPC
- Similar to VPC Peering, but you don’t have control on the other VPC

Private Service Access

- Use when:
  - Need to connect to VPC-hosted services

Private Service Access

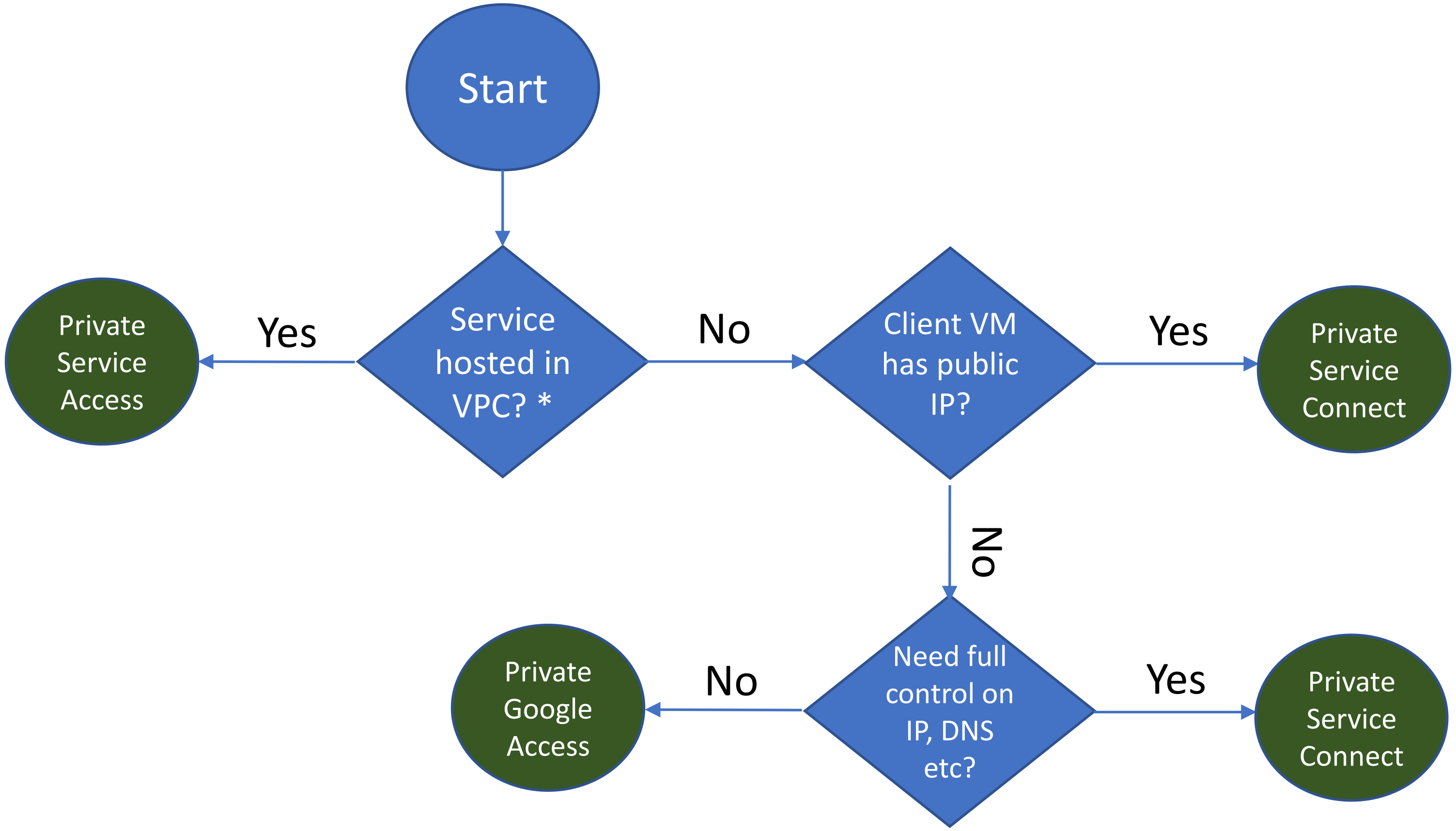
- Supported services:



- AI Platform Training
- AlloyDB for PostgreSQL
- Apigee
- Backup and DR
- Cloud Build
- Cloud Intrusion Detection System
- Cloud SQL (does not support DNS peering)
- Cloud TPU
- Filestore
- Google Cloud VMware Engine
- Looker (Google Cloud core)
- Memorystore for Memcached
- Memorystore for Redis
- NetApp Cloud Volumes Service
- Vertex AI

See updated list here: https://cloud.google.com/vpc/docs/private-services-access#private-services-supported-services

Choosing Private Access Implementation



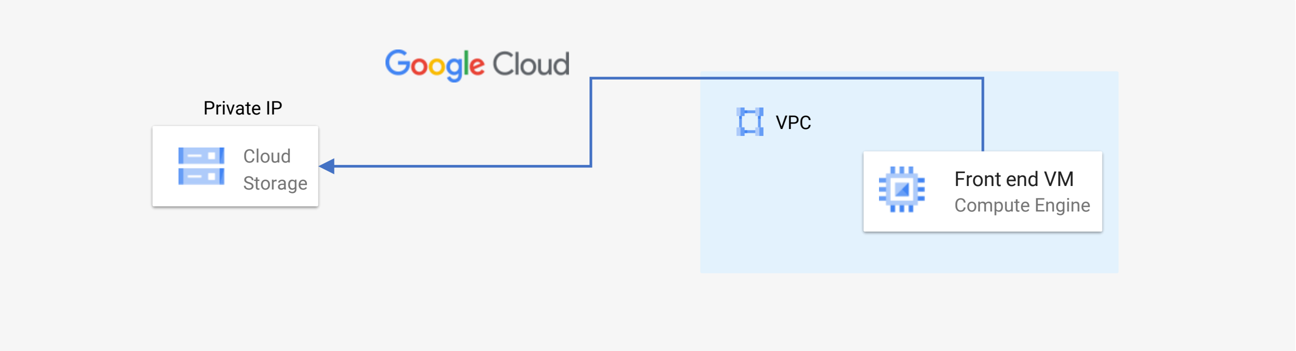
* See updated list here: https://cloud.google.com/vpc/docs/private-services-access#private-services-supported-services

Using Private Access

- We'll use all three types of Private Access later in the course when connecting the app to the data stores

Serverless VPC Access

- Private Access took care of connecting from VPC to non-VPC services
- What about the other way around?

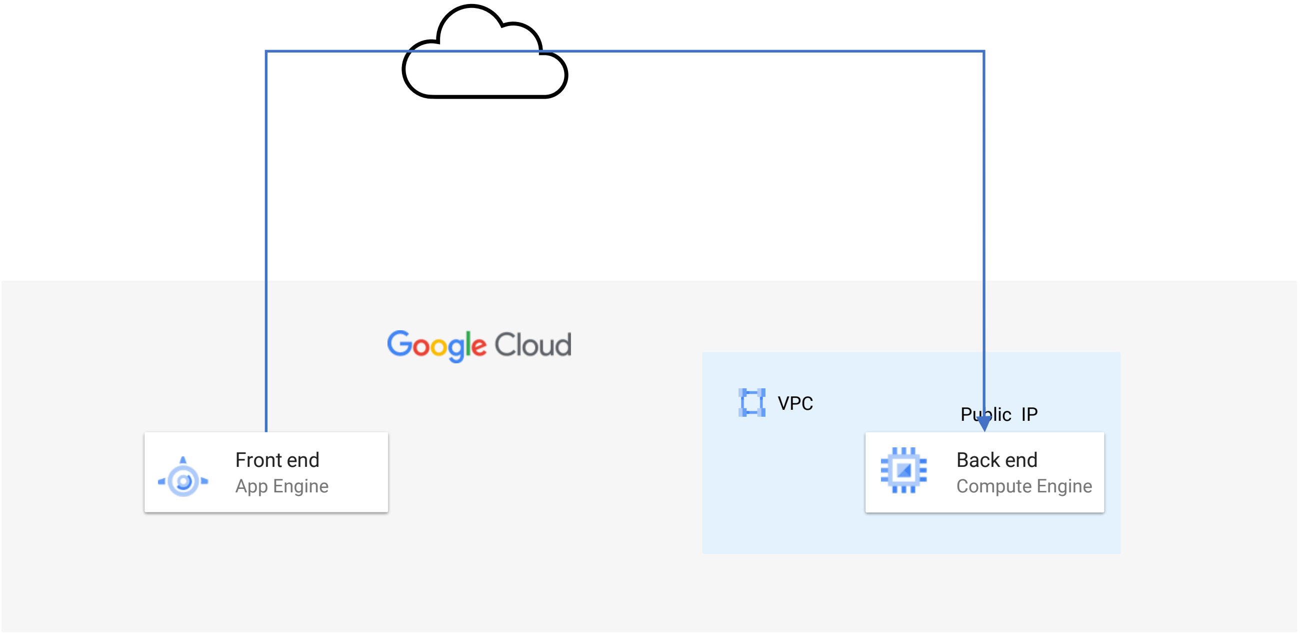


Serverless VPC Access

- Non-VPC services sometimes need access to VPC resources
- Example:
    - App Engine needs access to backend VM Instance in a VPC

Serverless VPC Access

- By default: using the VM public IP, through the internet



Not secure:

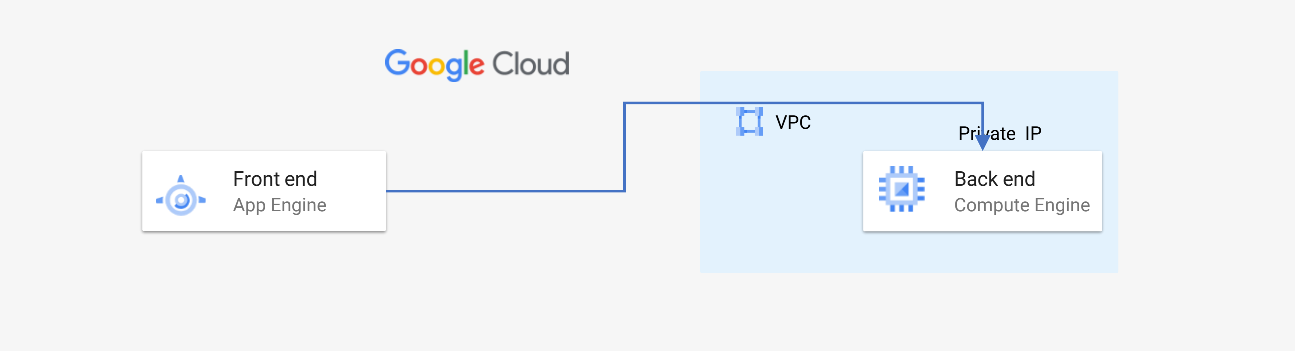
1. Traffic goes through the internet
2. The VM instance is open to the internet

Serverless VPC Access

- Serverless VPC Access allows secure access from serverless services to services in VPC
  - ie. App Engine to VM

Secure:

1. Traffic stays in the cloud
2. The VM instance is not open to the internet



Serverless VPC Access

- Supported services:
  - Cloud Run
  - Cloud Functions
  - App Engine standard environment
    - Except PHP 5

Serverless VPC Access

- How it works:
  - A Connector is created between the serverless and the VPC
    - Basically a VM instance
    - One of:
      - `f1-micro`
      - `e2-micro`
      - `e2-standard-4`

Serverless VPC Access

- Autoscaling as needed
- Min 2 instances, max 10 instances
    - Can be configured
- Pricing: Standard instances pricing
- The instances receive requests from the serverless and relay them to the VPC

Using Serverless VPC Access

- We'll use serverless VPC access later in the course when connecting the Cloud Run securely to the data store

App Engine Firewall Rules

- Firewall Rules can also be defined for App Engine
- Different from Firewall Rules of VPC
- Simpler to set up
- Block or deny requests from specific IP ranges
- Default rule allows access from every IP