Google Cloud (Advance)

Techlanders

Introduction

Your Name

Total experience

Background – Development / Infrastructure / Database / Network

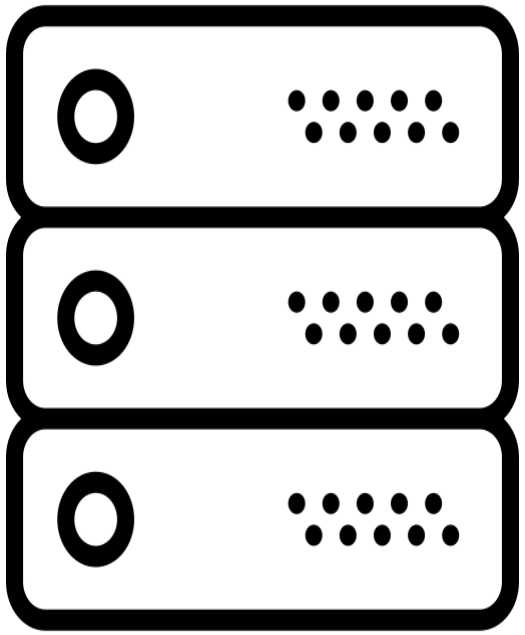
Experience on cloud/AWS

Your expectations from this training

Cloud

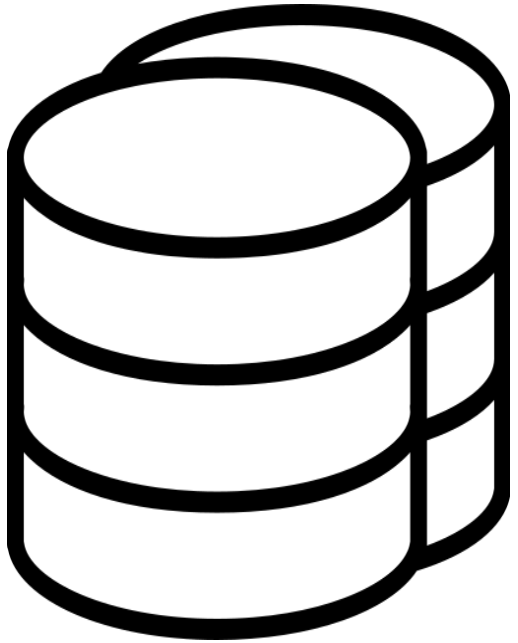
What is Cloud?

Traditional Datacenter



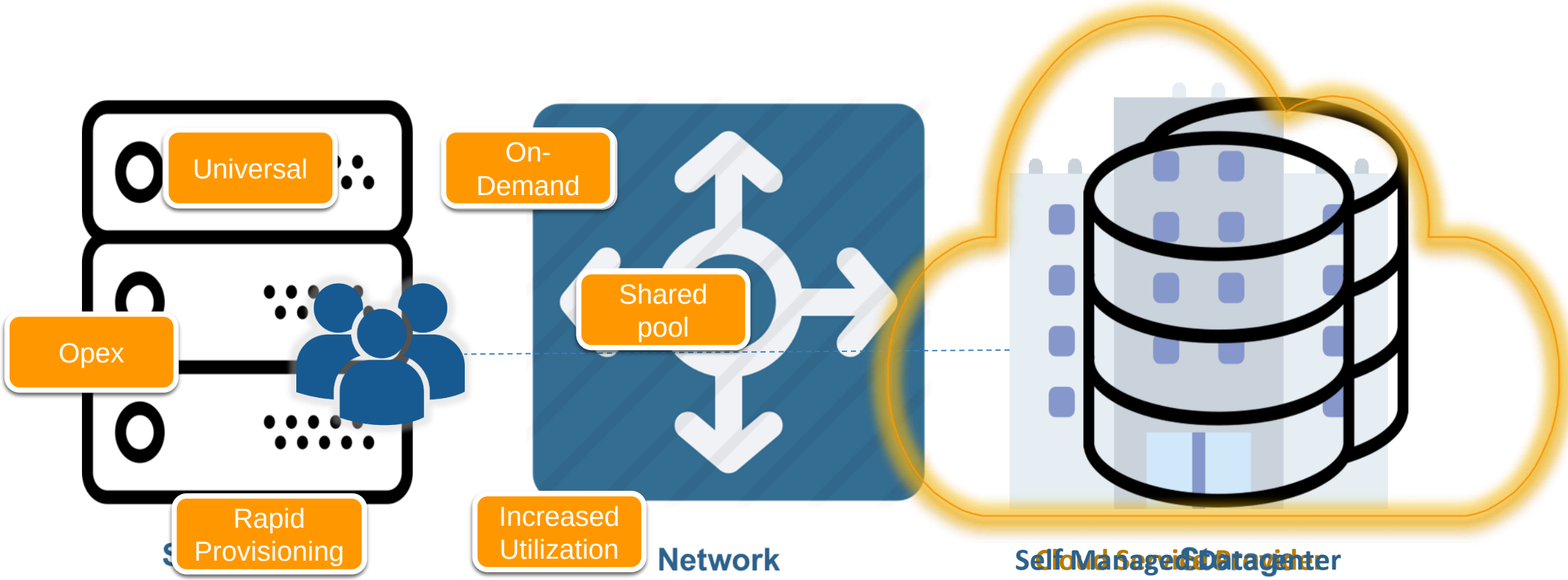
Servers

Network

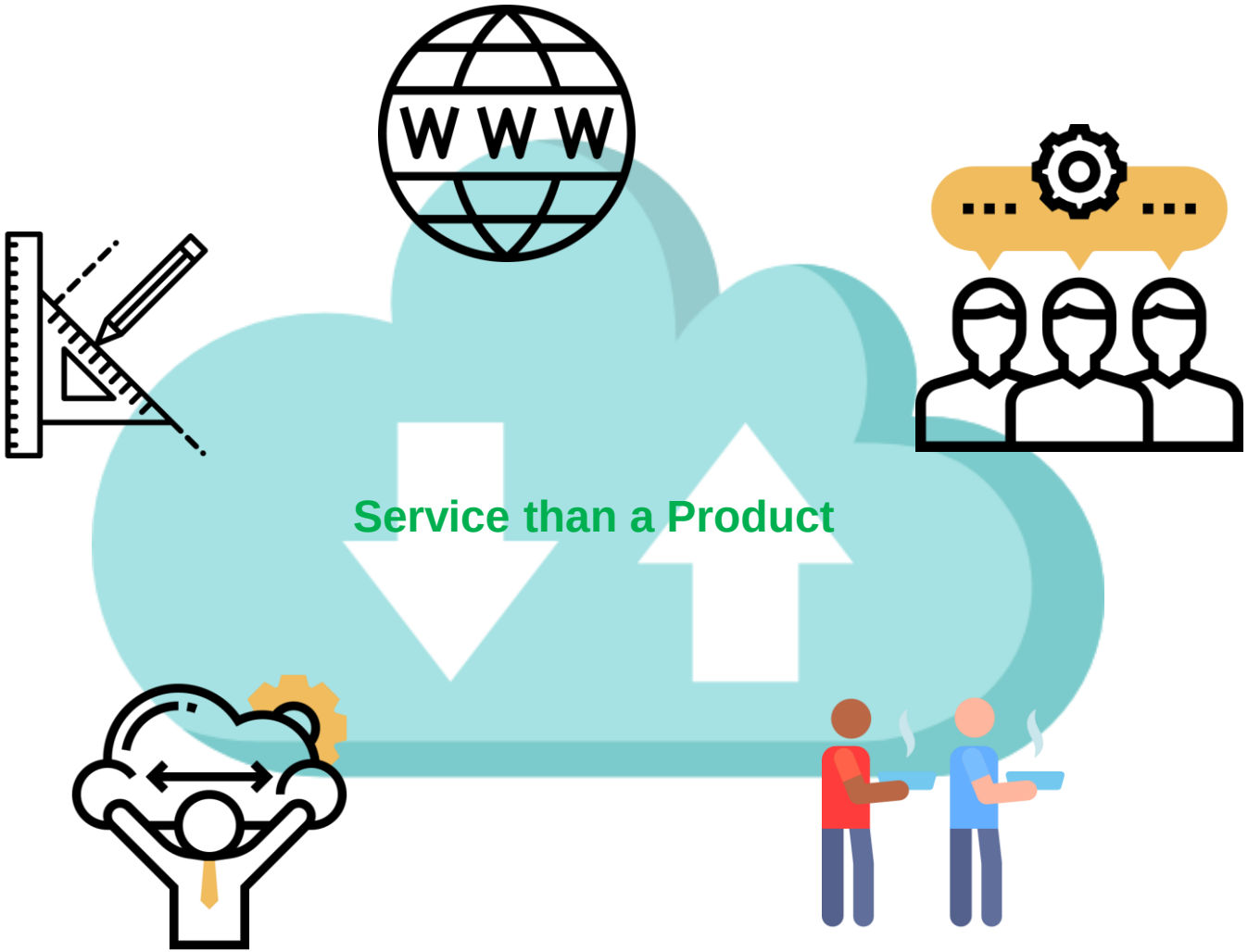


Storage

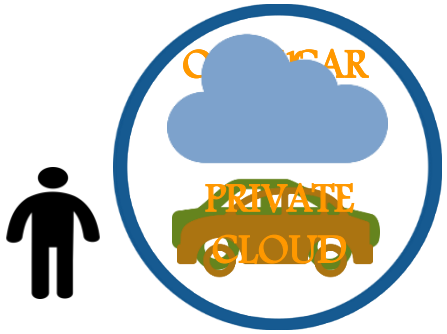
Traditional vs Cloud Datacenter



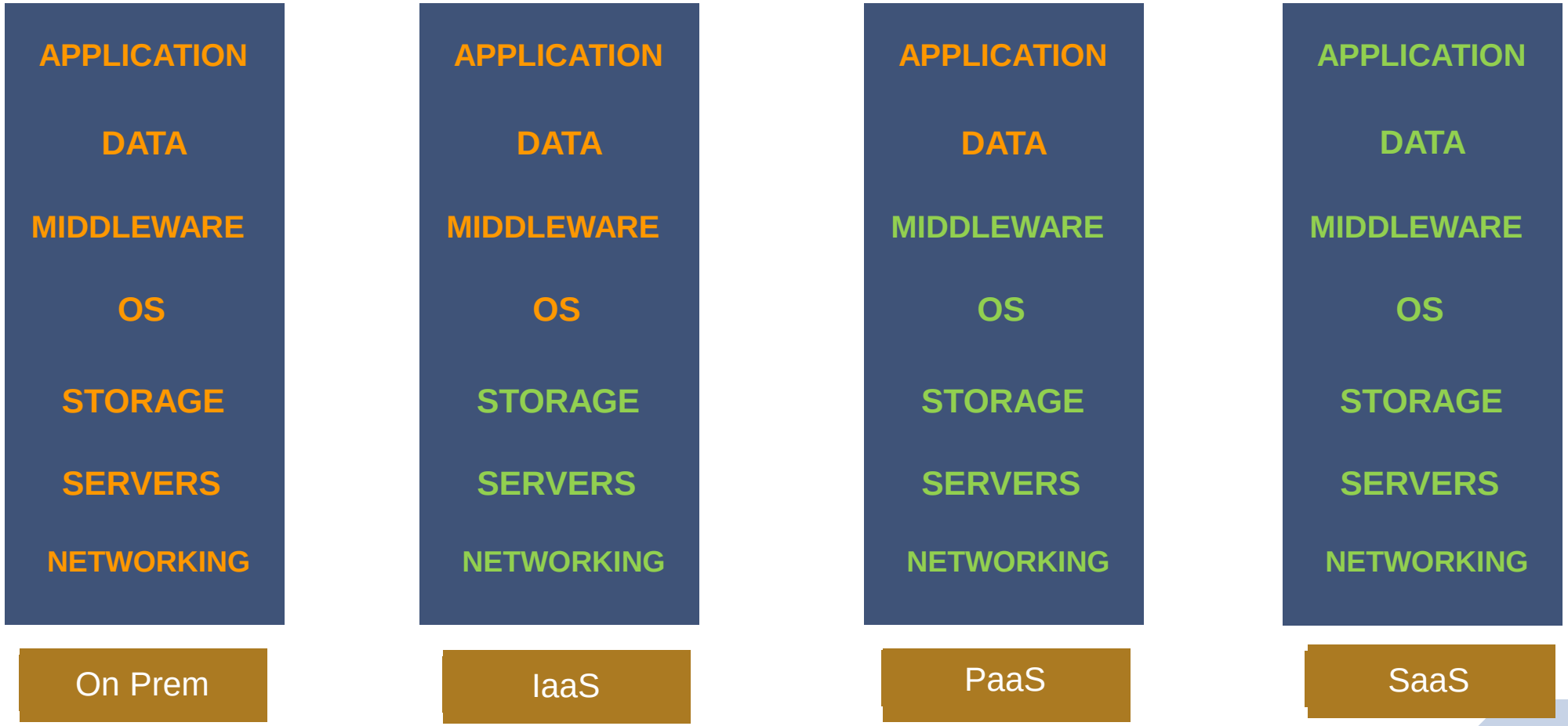
CHARCTERSTICS OF CLOUD



CLOUD DEPLOYMENT TYPE

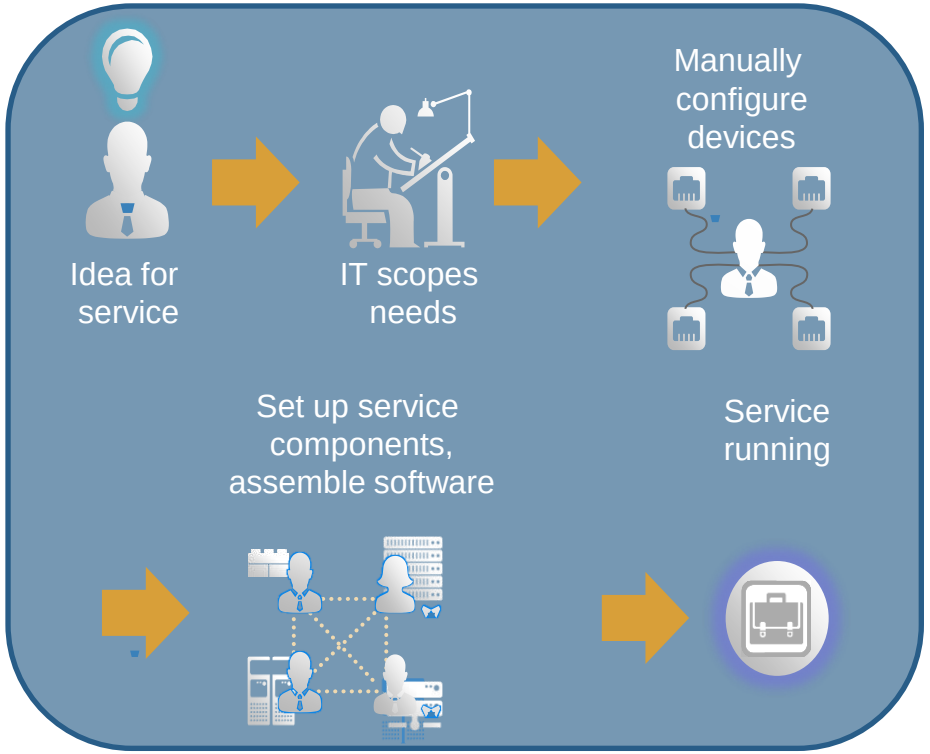


CLOUD SERVICE MODELS



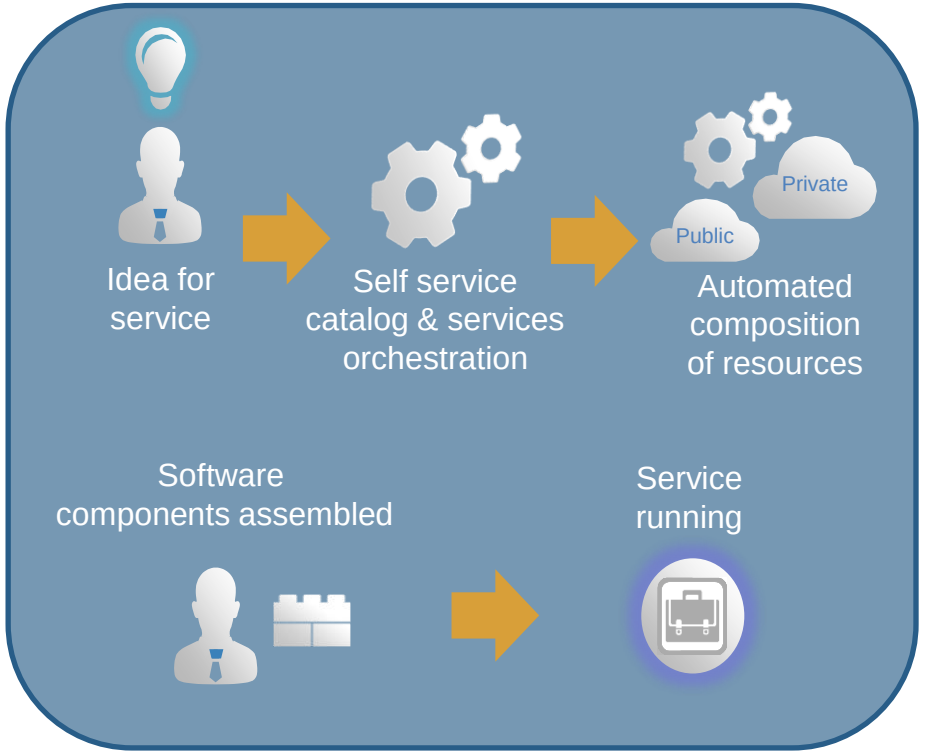
BUSINESS IMPACT

Traditional Datacenter



Time to Provision New Service: Months

Cloud Infrastructure



Time to Provision New Service: Minutes

CLOUD BENEFITS

CLOUD CHARACTERSTICS

Cloud Major Use Cases

On and Off

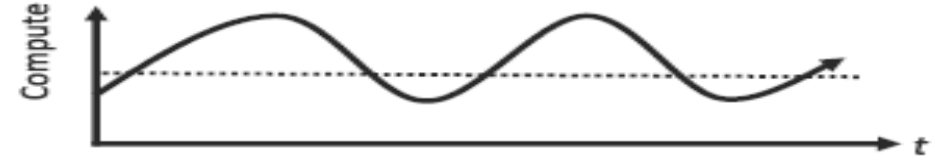
On and off workloads (e.g. batch job)
Over provisioned capacity is wasted
Time to market can be cumbersome

Growing Fast

Successful services needs to grow/scale
Keeping up with growth is a big IT challenge
Cannot provision hardware fast enough

Unpredictable Bursting

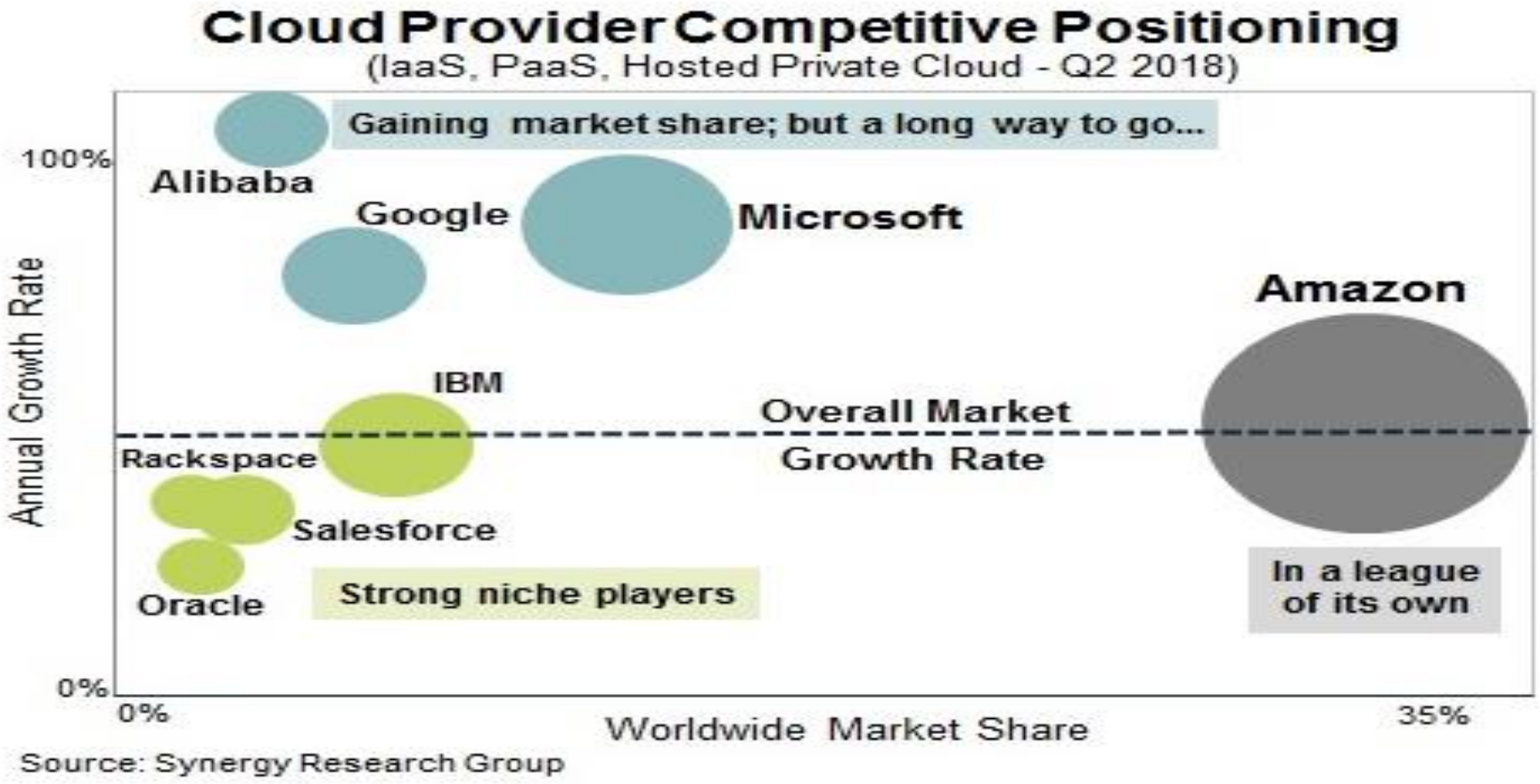
Unexpected/unplanned peak in demand
Sudden spike impacts performance
Cannot over provision for extreme cases



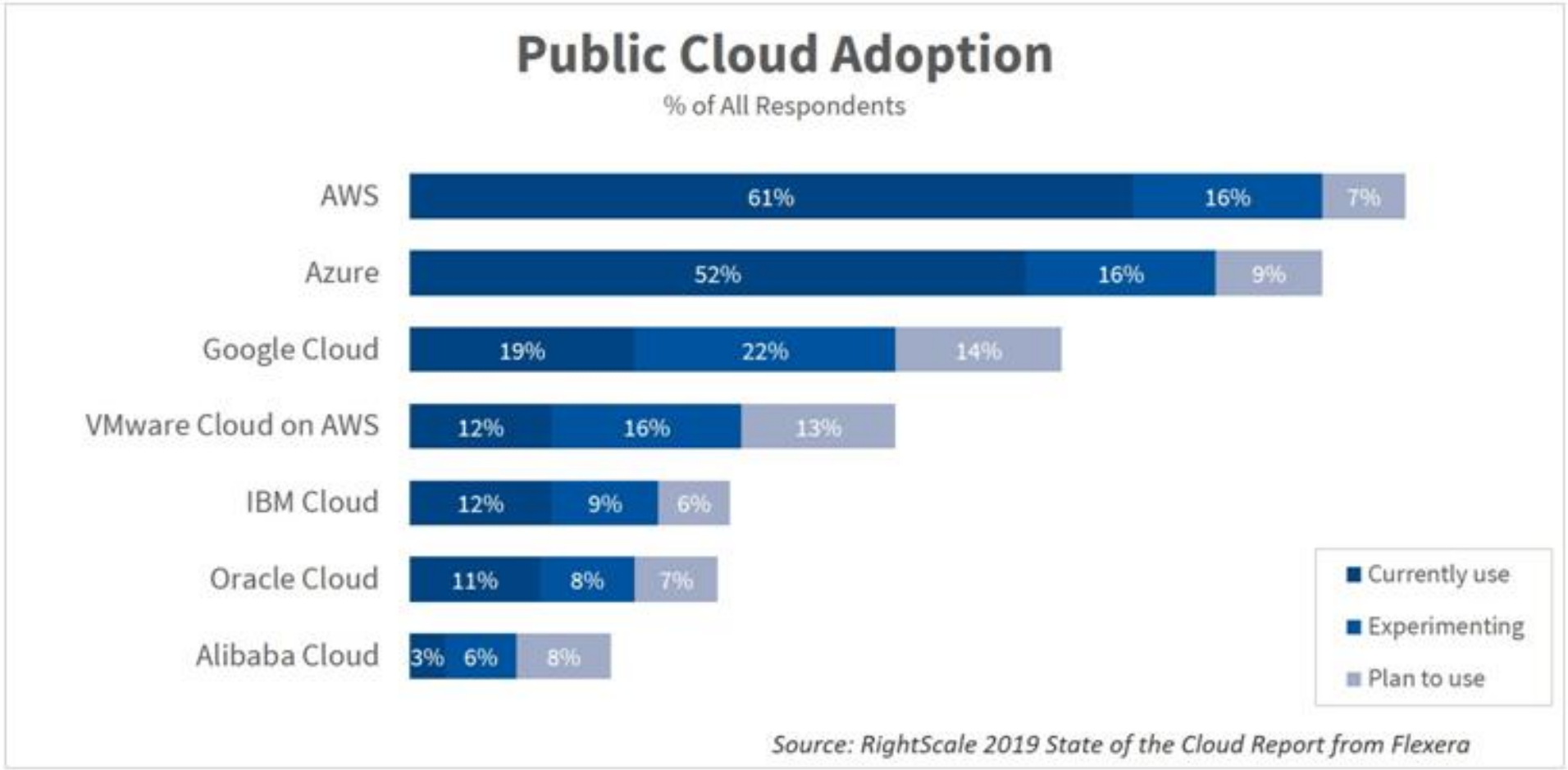
Predictable Bursting

Services with micro seasonality trends
Peaks due to periodic increased demand
IT complexity and wasted capacity

CLOUD PLAYERS



CLOUD PLAYERS



GCP
(Google Cloud Platform)

Google Cloud Platform

GCP (Google Cloud Platform) is a group of web services (also known as cloud services) provided by Google.

GCP provide IT infrastructure like CPU, Storage as a service, which means there is no need for any hardware procurement.

100’s of instances can be build and use in few minutes as and when required, which saves ample amount of hardware cost for any organizations and make them efficient to focus on their core business areas.

Currently GCP is present and providing cloud services many countries and is expanding in IaaS, CaaS and Machine Learning services.

GCP Core benefits

Low Cost: GCP offers, pay as you go pricing. GCP models are usually cheapest among other service providers in the market.

Instant Elasticity: You need 1 server or 1000's of severs, GCP has a massive infrastructure at backend to server almost any kind of infrastructure demands with pay for what you use policy.

Scalability: Facing some resource issues, no problem within in seconds you can scale up the resources and improve your application performance. This cannot be compared with traditional IT datacenters.

Multiple OS's: Choice and use any supported Operating systems.

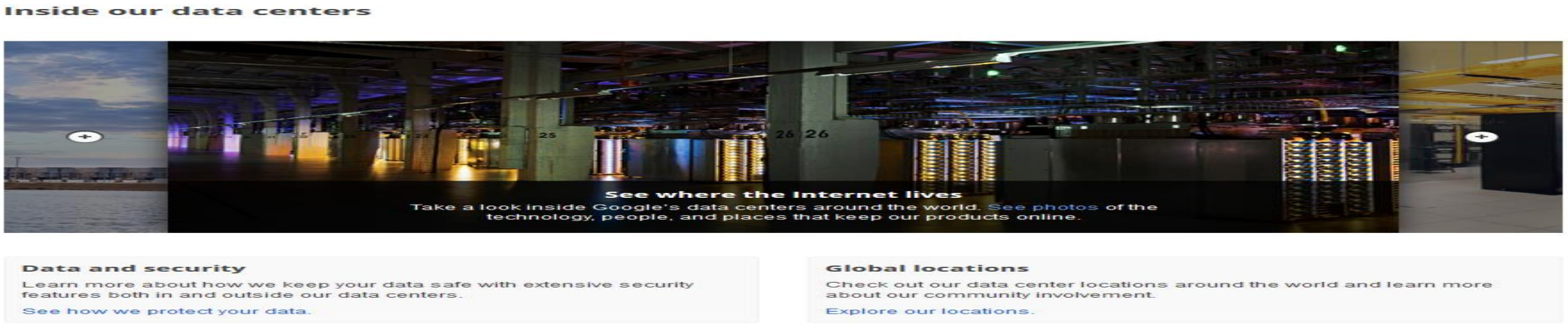
Multiple Storage Options: Choice of high I/O storage, low cost storage. All is available in AWS, use and pay what you want to use with almost any scalability.

Secure: GCP is ISO 27001, SOC 2/3, and PCI DSS 3.0 passed. Infact systems based on GCP are usually more secure than inhouse IT infrastructure systems.

GCP Global Infrastructure

- **Smart Datacenters:**

- Lets have a look and see why Google DC's are different and unique:



- Google is the first company in North America to obtain multi-site ISO 50001 Energy Management System certification, which includes 12 data centers across the US, Europe, and Asia.

- Source: https://www.google.com/about/datacenters/inside/index.html

GCP Global Infrastructure

GCP Regions:
-Geographic Locations
-Consists of at least three GCP Zones(AZs)
-All of the regions are independent of each other with separate Power Sources, Colling and Internet connectivity.
-*Regions* actually consists of *zones*

GCP Zones
-Zones in GCP is a distinct location within a region
-Each zone is insulated (with low-latency links) from other to support single point of failures
-Each Region has minimum three or more AZ's

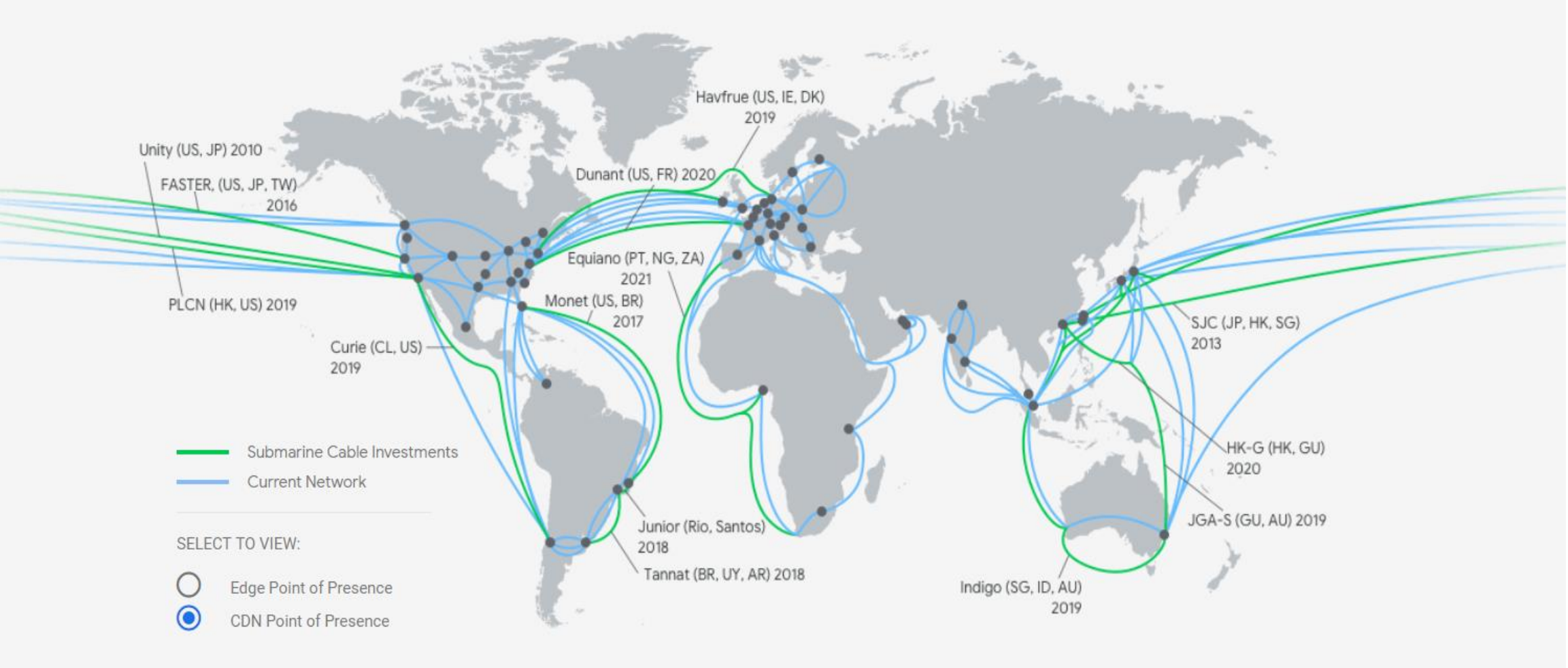
Feb-2020, GCP has 21 Regions, 64 zones, over 130+ points of presence across 200+ countries, and a well-provisioned global network with 100,000s of miles of fiber optic cable.

Source: https://cloud.google.com/about/locations/

Service Locations

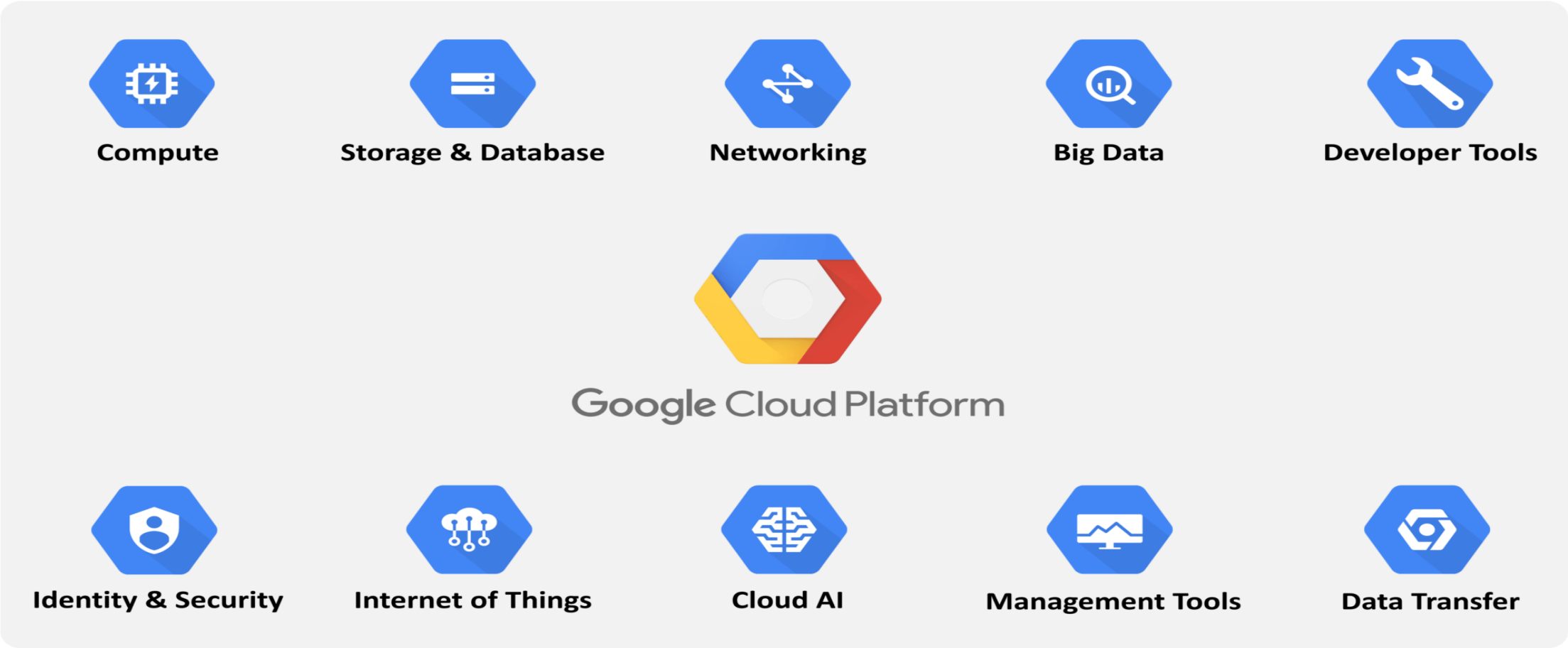
Source: https://cloud.google.com/about/locations/

Service Locations



Source: https://cloud.google.com/about/locations/

GCP Services



Source: https://cloud.google.com/products/

Accessing GCP Platform

GCP Management Console

GCP Command line interface (gcloud utility)

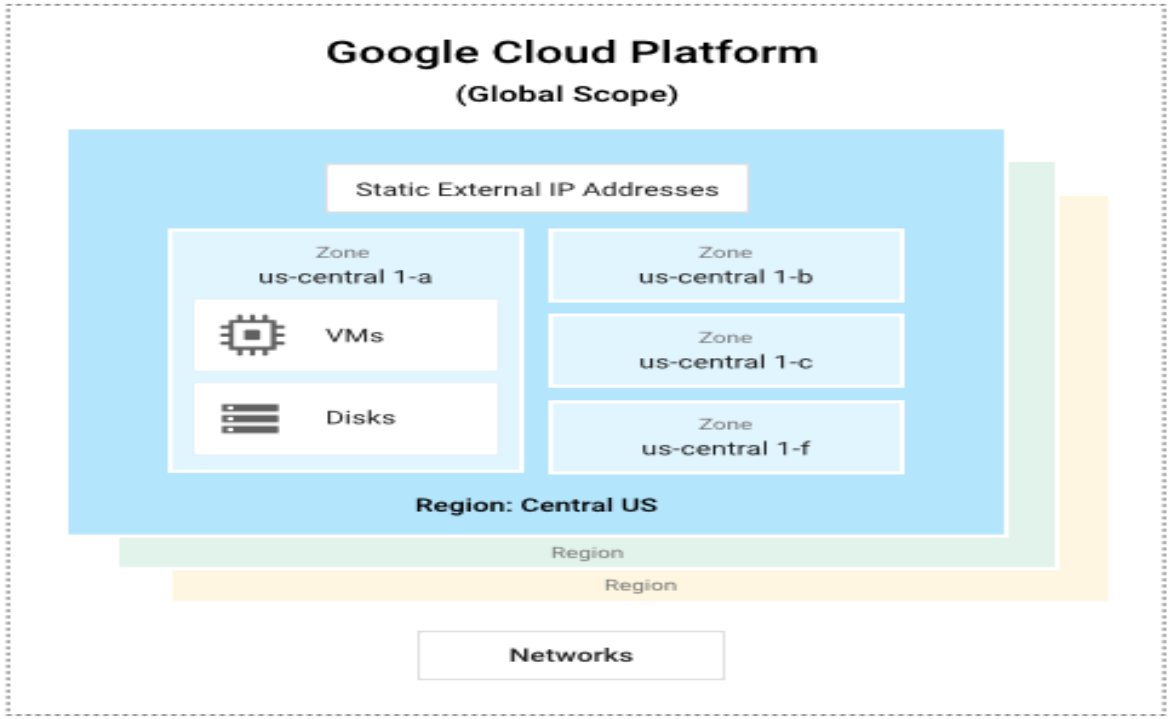
GCP Software Development Kit’s

GCP Resources

Everything in GCP is a resources, started from Physical hardware to Regions to Zones.

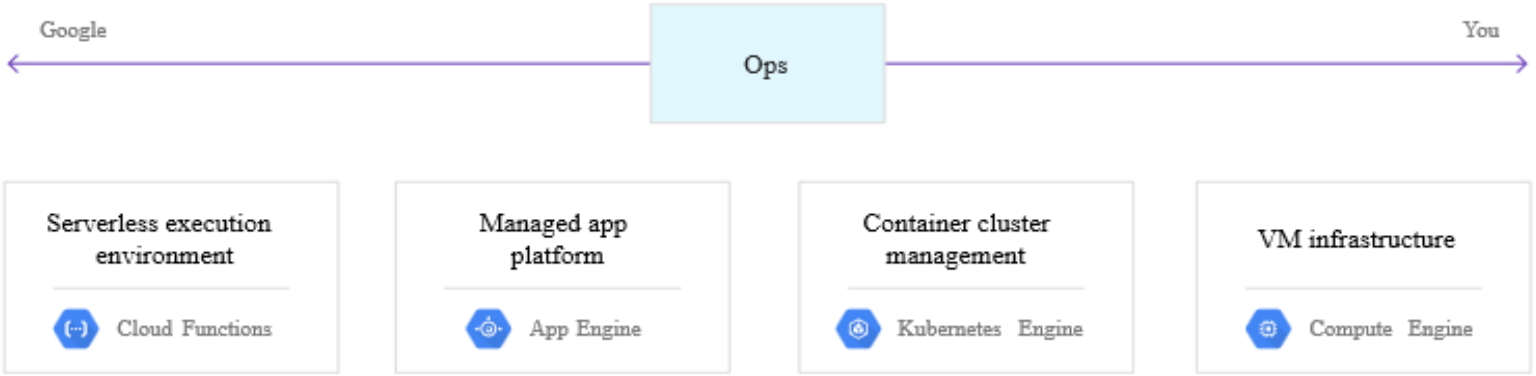
Broadly GCP resources are categorized into three parts:

-Global Resource
-Regional Resource
-Zonal Resource



GCP common services

- **Computing and Hosting Services**



- Compute Engine: Exactly like your VM machines
- Kubernetes Engine: Next layer of virtualization in IT
- App Engine: Google App Engine is GCP's *platform as a service* (PaaS). With App Engine, Google handles most of the management of the resources for you
- Serverless Computing: GCP's *functions as a service* (FaaS) offering, provides a serverless execution environment for building and connecting cloud services

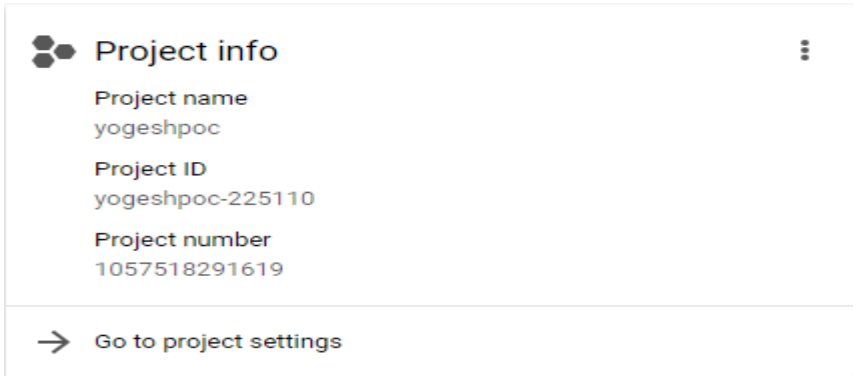
GCP Projects

Project is one of the critical Object in GCP, you can think Projects as Tenants in any other cloud environment.

Any GCP resources that you allocate, and use must belong to a project.

Each GCP project has:

- A project name, which you provide.
- A project ID, which you can provide or GCP can provide for you. - Each project ID is unique across GCP
- A project number, which GCP provides.



GCP Compute Services

GCP VM

- What all are the bare minimum infrastructure requirements for any application?

- Hardware (RAM, Processor, Processor type, HBA's, etc.)
- OS Images
- Storage (Internal Drives, External Drives)
- Network
- Security (Firewall, Networking, Access Mechanism)

GCP VM

- GCP VM stands for Google Cloud Platform Virtual Machines and is the Primary GCP web service.
- Provides Resizable compute capacity
- **Reduces the time required** to obtain and boot new server instances to minutes
- There are two key concepts to Launch instances in GCP:
  - **Images**
  - **Machine Type**

GCP VM's Facts:

- Automatic Live Migration in case of Zonal issues
- Scale capacity as your computing requirements change
- Pay only for capacity that you actually use
- Choose Linux or Windows OS as per need
- Deploy across GCP Regions and Availability Zones for reliability

VM Families

- A machine type is a collection of "virtualized hardware resources available to a virtual machine (VM) instance, including the system memory size, virtual CPU (vCPU) count, and maximum persistent disk capability."

- Google Offers two categories of Machine types:

- Predefined machine types
  - Standard machine types
  - High-memory machine types
  - High-CPU machine types
  - Shared-core machine types (A special type with Bursting CPU capabilities)
  - Memory-optimized machine types

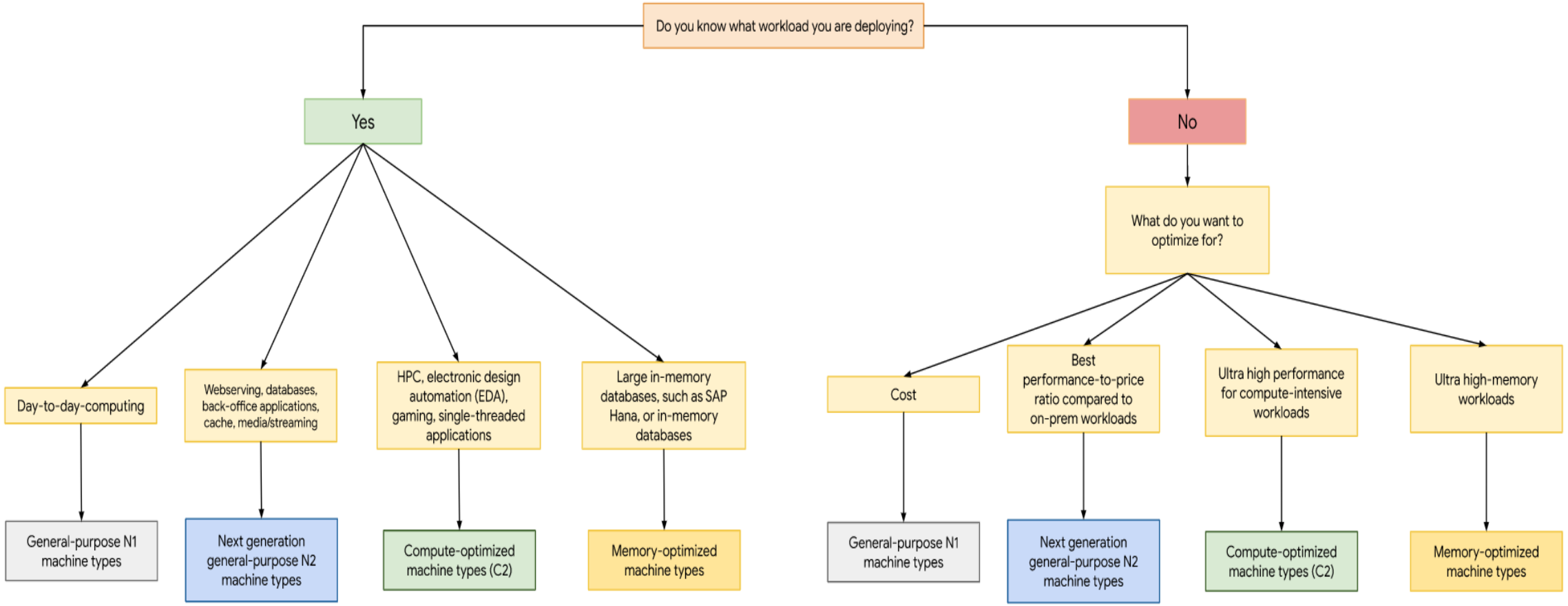
- Custom machine types: Customize as per need

CPU on GCP Platform

CPU Platform	Supported Machine Types	Base Frequency (GHz)	All-Core Turbo Frequency (GHz)	Single-Core Max Turbo Frequency (GHz)
Intel Xeon Scalable Processor (Cascade Lake)	• N2 predefined machine types • N2 custom machine types	2.8	3.4	3.9
	• C2 machine types	3.1	3.8	3.9
	• Memory-optimized machine types (`m2-ultramem` only)	2.5	3.4	4.0
Intel Xeon Scalable Processor (Skylake)	• N1 predefined machine types • Memory-optimized machine types (`n1-megamem` only) • N1 custom machine types	2.0	2.7	3.5
Intel Xeon E7 (Broadwell E7)	• Memory-optimized machine types (`n1-ultramem` only)	2.2	2.6	3.3
Intel Xeon E5 v4 (Broadwell E5)	• N1 predefined machine types • Memory-optimized machine types (`n1-megamem` only) • N1 custom machine types with up to 64 vCPUs and 416 GB of memory or 624 GB when using extended memory	2.2	2.8	3.7
Intel Xeon E5 v3 (Haswell)	• N1 predefined machine types • N1 custom machine types with up to 64 vCPUs and 416 GB of memory or 624 GB when using extended memory	2.3	2.8	3.8
Intel Xeon E5 v2 (Ivy Bridge)	• N1 predefined machine types • N1 custom machine types with up to 64 vCPUs and 416 GB of memory or 624 GB when using extended memory	2.5	3.1	3.5
Intel Xeon E5 (Sandy Bridge)	• N1 predefined machine types • N1 custom machine types with up to 64 vCPUs and 416 GB of memory or 624 GB when using extended memory	2.6	3.2	3.6

Source: https://cloud.google.com/compute/docs/cpu-platforms

CPU on GCP Platform



Source: https://cloud.google.com/compute/docs/cpu-platforms

LAB 1

Create a Windows virtual machines through Portal

Create a Linux virtual machine and login using putty

Accessing a Virtual Machine

Stop, Start, Restart etc. to be performed

Understand Activity Logs, Extensions

Understand monitoring Metrics, Alerts

Check Boot diagnostics, Resource health, reset password, redeploy options

IP Addresses

- Private IP
  - Ephemeral
  - Persistent

- Public IP
  - Ephemeral
  - Persistent

LAB 2

- GCP Console Walkthrough
- First GCP instance
- Compute Engine
- VM Instances
- Name of a machine
- Select Zone
- Select Machine type (shared one in this case)
- Select Ubuntu latest image (18.04)
- And let's select default options here and create your machine

Console Access: https://console.cloud.google.com

LAB 3

Console Access: https://console.cloud.google.com

Exercise: Create a Linux Operating System (lets say Ubuntu 18.04), and login via cloud shell:

Case-1
Reboot and observer the Public IP (there will be no change)

Case-2
Stop the Instance, refresh your screen and Start it back, observe the Public IP

Que is how to make Public IP's Persistent and the Solution is Persistent External IP.

Exercise: Create an EIP, Allocate the EIP to an Instance and perform both cases again.

Understanding Disks

HDD (Magnetic disk) vs SSD (Solid State Disks)

Standard Disk vs Premium Disk

OS Disk vs Data Disk

Data Disk vs Cache Disk

Persistent Disk vs Instance Store disk

Persistent Disk Facts

- Durable: Persistent Disk is designed for high durability. It stores data redundantly to ensure data integrity.
- Independent Volumes: Your storage is located independently from your virtual machine instances, so you can detach or move your disks to keep your data even after you delete your instances.
- Volume Size: Each persistent disk can be up to 64 TB in size, so there is no need to manage arrays of disks to create large logical volumes.
- **Online Resize: Online growth allows volumes to grow on-demand without the need to restart virtual machines or reattach volumes.**

Snapshots

- What are snapshots.
- Snapshots are point in time copies of your Instance Volumes.
- You can create snapshots from your existing volumes assigned on dedicated instance and use them on any other instance with same configuration/data.
- Typically a manual replication of a storage.
- You can create volumes (persistent disks) from GCP snapshots and assign them to other instances.
- Even you can create full Image copy from GCP snapshots using root volumes.

Lab 5: Snapshots

- Let's create a snapshots from one volume assigned on one host and replicate/use the same on other host with a new volume.

- Go to GCP console
- Select Persistent disk for which you want the snapshot
- Create snapshot
- Once snapshot is create, create a Persistent disk out of it
- Assign that Persistent disk to any other instance
- Mount it and test the data availability

- Note: The instance where to store the Snapshots should be in same zone.

GCP VM Images

GCP VM Images

- Images are used to create boot disks for your instances.
- Image is a template for the root volume for the instance (example: an OS image, a webserver, an application server etc.)
- Its contains a block device mapping that specifies the volume to attach to the instance whenever it's launched.
- Two types – "Public" and "Custom"
- All Images are based on x86 OSs, either Linux or Windows.
- Source: https://cloud.google.com/compute/docs/images

LAB 6 - Images

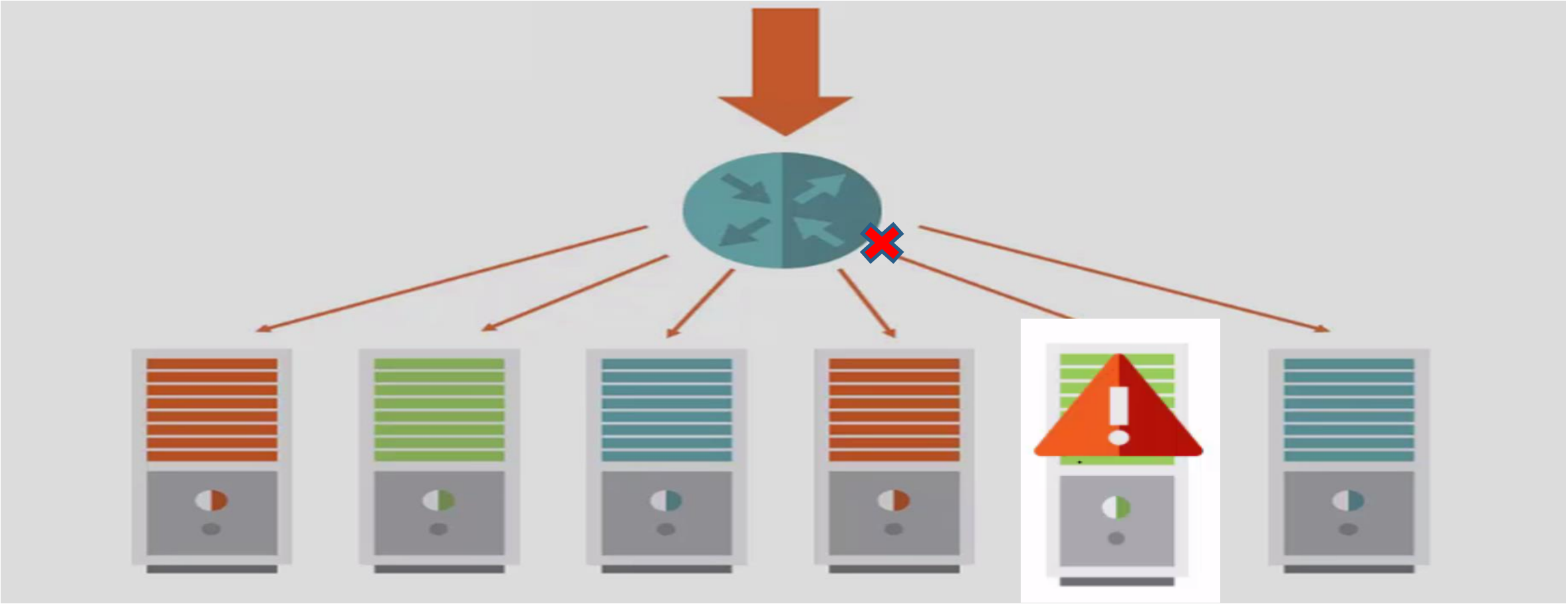
- Let us create a custom image for a webserver:
- Create an Instance "named webserver" from GCP Compute Engine (example ubuntu 18.04)
- Make sure to allow http and https traffic from firewall
- Click create and login into the machine
- Configure the machine as per you add some files and install some package
- Create Image for the machine
- Create machine from the Image

GCP VM Images

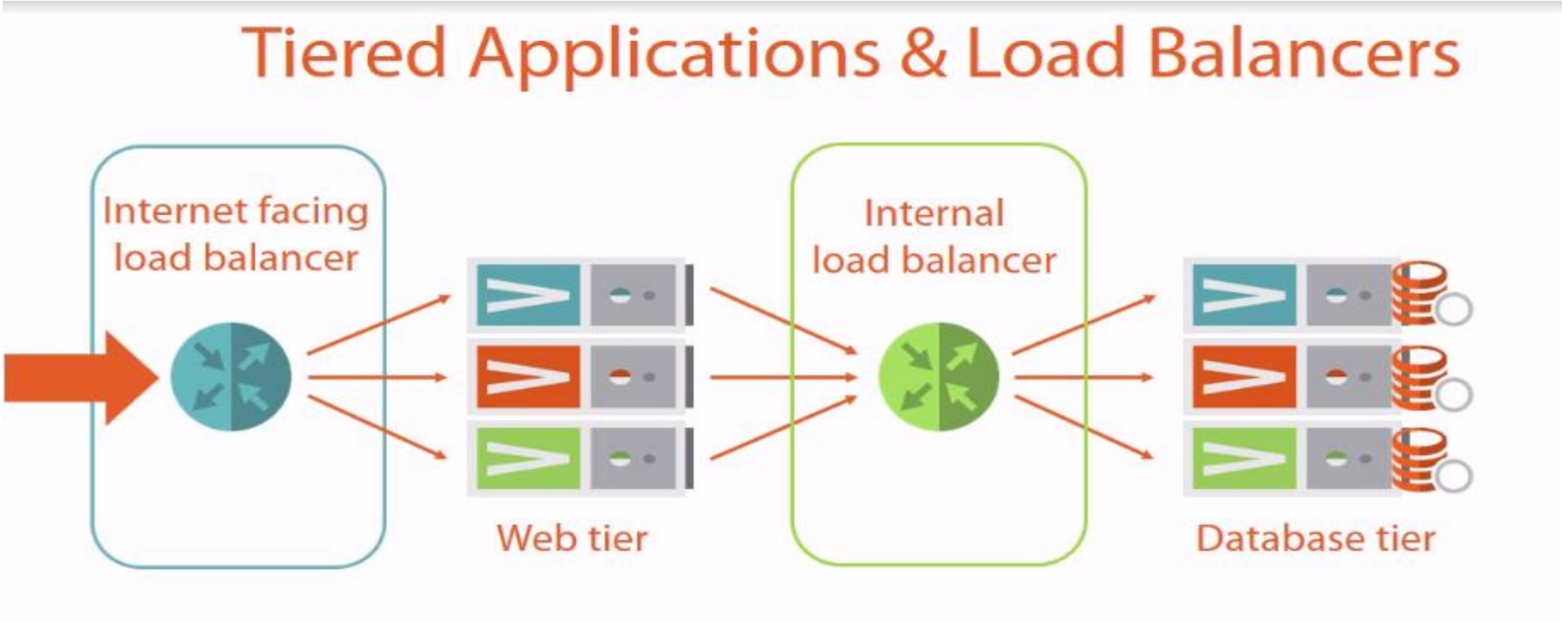
Scenarios	Machine image	Persistent disk snapshot	Custom image	Instance template
Single disk backup	Yes	Yes	Yes	No
Multiple disk backup	Yes	No	No	No
Differential backup	Yes	Yes	No	No
Instance cloning and replication	Yes	No	Yes	Yes
VM instance configuration	Yes	No	No	Yes

GCP
Load Balancer as a Service - NLB

Load Balancers



Load Balancers



GCP Load Balancers

- **Distributes** traffic across multiple instances
- Supports **health checks** to detect unhealthy Virtual instances
- Supports the **routing and load balancing** of HTTP, HTTPS, and TCP traffic to GCP Virtual instances
- Works within Region, Across-region and across Availability zones.

LAB: GCP Load Balancers

L4 Network Load Balancer:

Create two webservers and configure them to have two different web pages

Go to load balancers and create L4 TCP LB

Click start configure and select internet to VM traffic

Now configure frontend (forwarding rules) and backend configuration and put your instances there

You also need to create a health check from backend configuration section

Go to load-balancer and check the status

check the status ip and send the trafic there:

Go to browser and check the Webpage

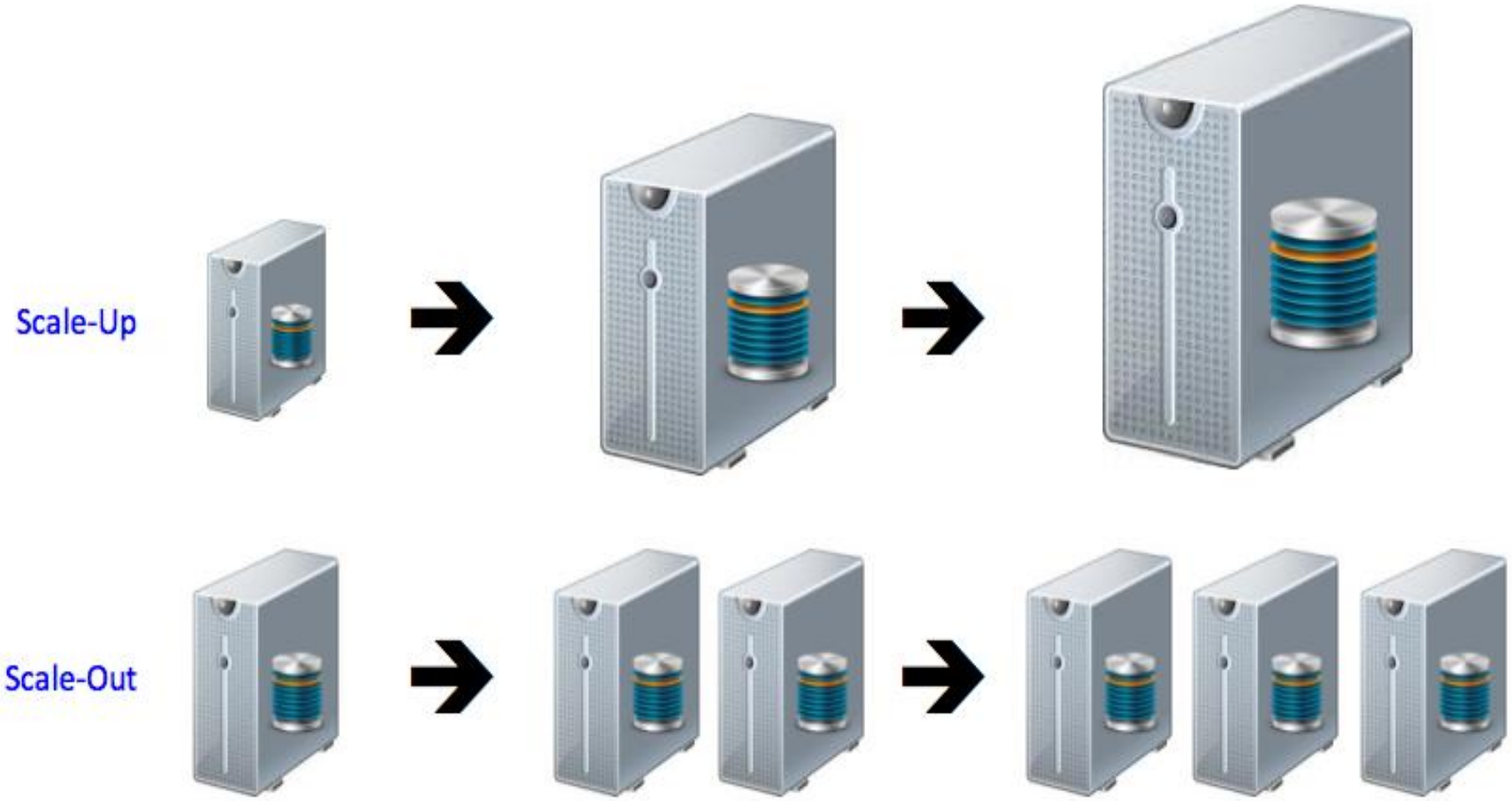
LAB: GCP Load Balancers

Remove one old server from the Load balancer

Now wait from 60 secs and confirm the traffic is getting distributed between two existing servers only

Scaling Infra

Scaling



GCP Instance Groups Auto-Scaling

GCP AutoScaling

- **Scale** your GCP Instances capacity **automatically**
- Well-suited for applications that experience **variability in usage**
- Available at no additional charge
- Instance Template: An instance template is a configuration template that lets you describe a VM instance. You can then create groups of identical instances based on the template. . When you create a template configuration, you can specify: An instance template.

  - Template name
  - Machine Type
  - Custom Disk
  - Firewall rules etc.

GCP AutoScaling

- Instance Group: Contain a collection of GCP instances that share similar characteristics.

- Instances in an Auto Scaling group are treated as a logical grouping for the purpose of instance scaling and management.

- Autoscaling can be based on:
- CPU usage
- HTTP LB
- Stackdriver
- Multiple Matrics

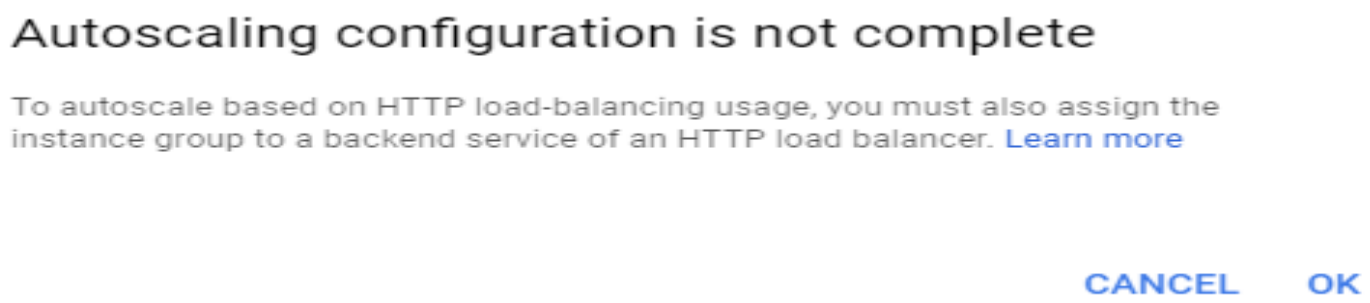
GCP AutoScaling

Now set minimum and maximum number of instances required for autoscaling.

Create and enable health checks on port 80.

Let us reduce the delay time from 300 sec to 60 sec.

Ignore below error as we will be creating load balancer in the next step.



GCP AutoScaling

Go to network services and click on Load Balancing

Select HTTP(S) load balancing

Create Frontend configuration first – this will provide the fronted IP via which traffic will come in

Next create backend services - A backend service directs incoming traffic to an instance group

Select instance group and health check here and click create

Finally give a name to your load balancer and click create, it will take some time and your LB and instance will get reflected here

GCP AutoScaling

Load balancers Backends Frontends

Filter by name or protocol

Name	Protocol	Backends
mylb	HTTP	1 backend service (1 instance group, 0 network endpoint groups)

To edit load balancing resources like forwarding rules and target proxies, go to the advanced menu.

Backend

Backend services

1. newbackend

Endpoint protocol: **HTTP** Named port: **http** Timeout: **30 seconds** Cloud CDN: **disabled** Health check: health-checks

Advanced configurations

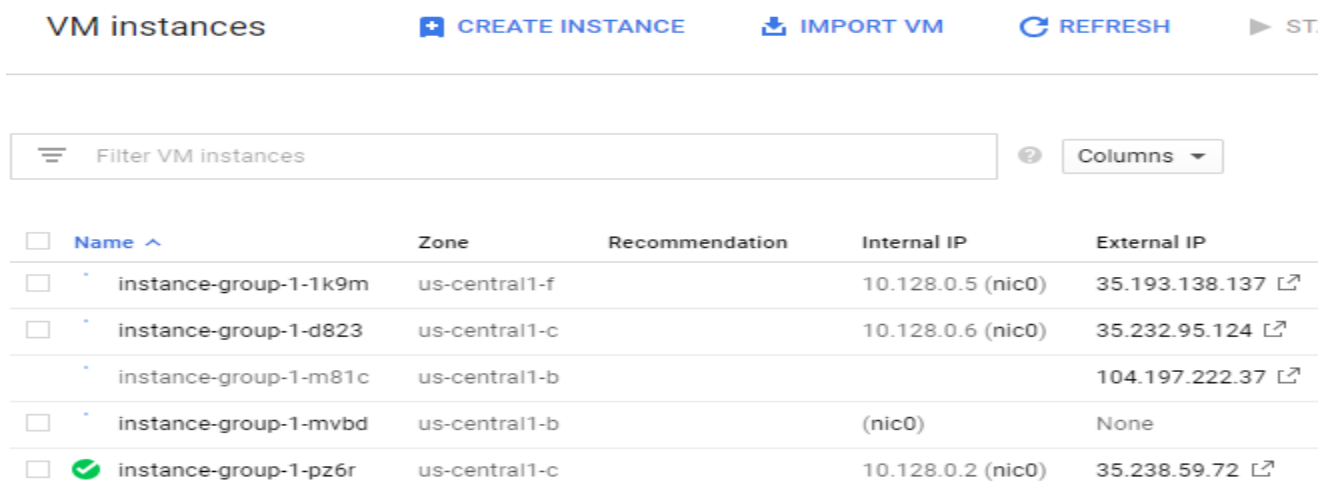
Instance group	Zone	Healthy	Autoscaling	Balancing mode	Capacity
instance-group-1	us-central1	2 / 2	Target LB capacity fraction 80%	Max CPU: 80%	100%

Name	Creation time	Template	Zone	Internal IP	External IP	Connect
instance-group-1-m81c	Dec 13, 2018, 11:23:54 PM	instance-template-1	us-central1-b	10.128.0.3 (nic0)	104.197.222.37	SSH
instance-group-1-pz6r	Dec 13, 2018, 11:23:54 PM	instance-template-1	us-central1-c	10.128.0.2 (nic0)	35.238.59.72	SSH

GCP AutoScaling

Its time to test Autoscaling:
Let us create a new instance and perform apache benchmarking to send lot of traffic to force autoscaler to spin up new instances

Let us use the same machine image here to create an instance and send traffic using:
-ab -n 2000000 -c 1000 http://<LOAD BALANCER IP:80/



LAB - Autoscaling

Create an instance template, managed instance group and finally link them to a load balancer and perform load testing to ensure autoscaling is working as expected.

Case1: Send load and check the AS is creating the new instances and distributing the traffic

Case2: Stop sending the load and observe the instances are getting terminate once the load is over

Case3: Try to delete one node from the “desired number” and you will see the AS group will automatically create an instance back for you

Cloud Armor

Attack Impact

Financial loss

Production loss

Security threat

Cloud Armor

What is Cloud Armor?

Google Cloud Armor is a security service that helps protect applications and services from various types of threats, including DDoS attacks, web application vulnerabilities, and other malicious traffic. It's part of Google Cloud's overall security strategy and integrates with other services like Google Cloud Load Balancing:

- Distributing Denial-of-service (DDoS)
- Cross-site-scripting (XSS)
- SQL injection

Cloud Armor: Features

Key Features of Google Cloud Armor

DDoS Protection:

- Provides built-in defense against large-scale distributed denial-of-service (DDoS) attacks.
- Automatically mitigates threats by filtering malicious traffic before it reaches your application.

Web Application Firewall (WAF):

- Offers configurable rules to help protect against common web application vulnerabilities, including SQL injection, cross-site scripting (XSS), and other OWASP Top 10 threats.
- Allows you to create custom rules to tailor security to your specific application needs.

Traffic Management:

- Integrates with Google Cloud Load Balancing to distribute traffic across multiple instances, improving application availability and performance.
- Can help you define security policies based on various parameters, such as geographic location, IP address, and request type.

Identity-Aware Proxy (IAP) Integration:

- Works seamlessly with Google Cloud's Identity-Aware Proxy to enforce security policies based on user identity, ensuring only authorized users can access specific resources.

Cloud Armor: Features

Key Features of Google Cloud Armor

Logging and Monitoring:

- Provides detailed logging and monitoring capabilities, enabling you to track traffic patterns and identify potential security issues.
- Integrates with tools like Google Cloud Logging and Google Cloud Monitoring for enhanced visibility.

Global Protection:

- Leverages Google's global network infrastructure to protect applications from attacks originating anywhere in the world.
- Ensures low-latency access to applications while maintaining high levels of security.

Security Policies:

- Allows you to define and enforce security policies at both the application and network levels.
- Policies can include rules for rate limiting, IP allowlists, and other security measures.

Real-Time Updates:

- Google Cloud Armor benefits from Google's real-time threat intelligence, which helps in quickly identifying and mitigating emerging threats.

Cloud Armor: Rate Limiting

Google Cloud Armor provides a robust way to implement rate limiting to protect your applications from excessive requests and to ensure fair usage of resources. Here's how rate limiting works in Google Cloud Armor, along with its features, benefits, and setup process.

Features of Rate Limiting in Cloud Armor

Granular Control:
- You can define rate limits at various levels, such as per IP address, geographic region, or even based on specific paths in your application.

Request Quotas:
- Set specific thresholds for the number of requests allowed within a given time frame (e.g., requests per minute).

Action on Exceeding Limits:
- Specify actions to take when a rate limit is exceeded, such as returning an HTTP 429 (Too Many Requests) response or redirecting users to a different page.

Integration with Security Policies:
- Rate limiting can be part of a broader security policy, alongside other features like WAF rules and IP blacklisting.

Logging and Monitoring:
- Monitor the effectiveness of your rate limiting policies through Cloud Armor logs and metrics, allowing you to adjust thresholds based on real usage patterns.

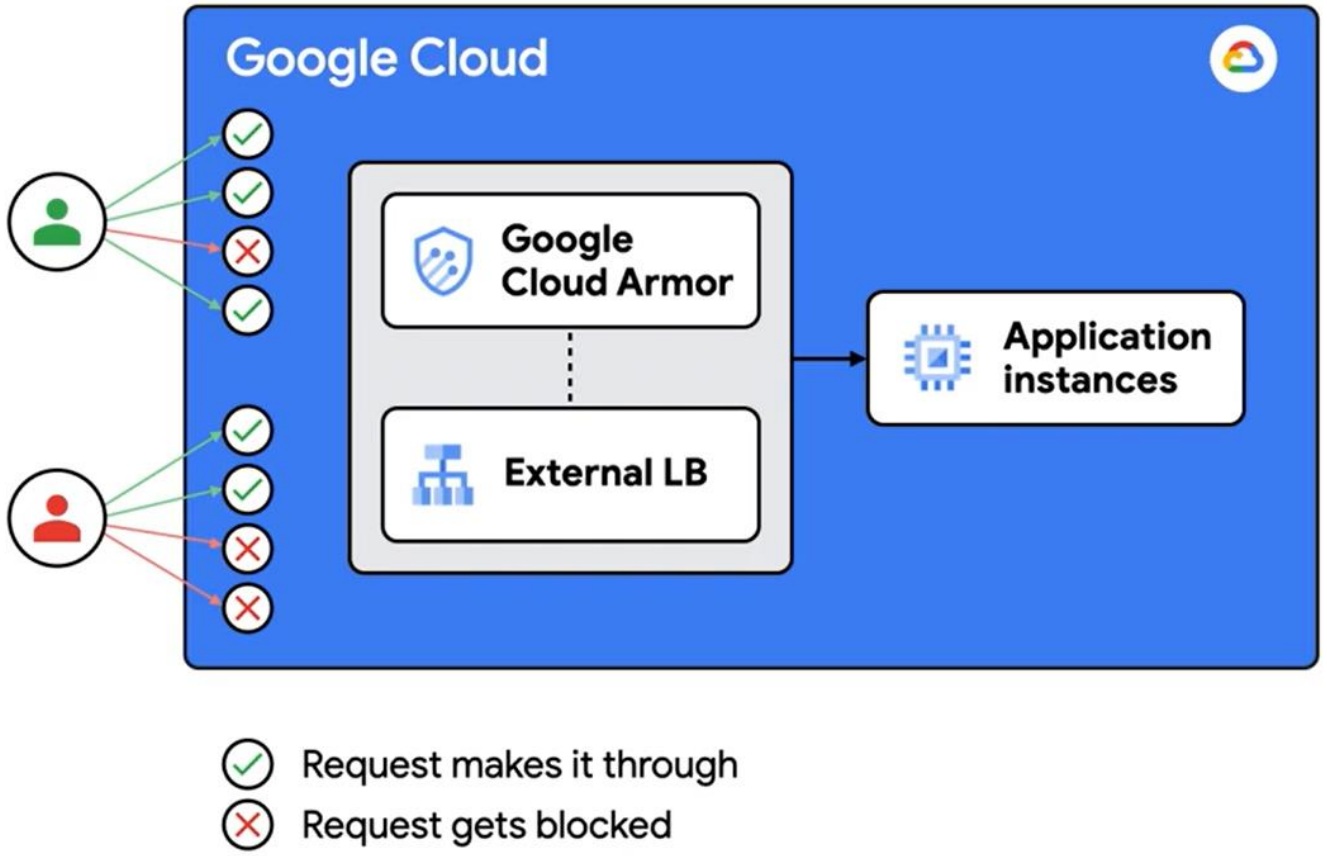
Cloud Armor: Rate Limiting

Throttle

Enforce maximum request limit per client or across clients by throttling individual clients to a user-configured threshold.

Rate-Based-Ban

Rate limit requests that match a rule on per-client basis and temporarily ban clients for configured period of time if threshold is exceeded.



Cloud Armor: Use Cases

Use Cases:

Web Applications:

- Protect web applications from vulnerabilities and DDoS attacks while ensuring high availability and performance.

APIs:

- Secure APIs from unauthorized access and common web vulnerabilities.

E-Commerce:

- Safeguard e-commerce platforms from attacks that could impact availability or compromise sensitive customer data.

Gaming:

- Protect gaming servers and applications from DDoS attacks and ensure a smooth user experience.

Cloud Armor: Adaptive Protection

ML Based L7 DDoS detection and protection

Google Cloud Armor's Adaptive Protection is designed to enhance the security of your applications, websites, and services hosted on Google Cloud against Layer 7 Distributed Denial of Service (DDoS) attacks.

It provides advanced threat detection and mitigation capabilities to protect against various malicious activities, particularly HTTP floods and other high-frequency layer 7 attacks.

Cloud Armor: Adaptive Protection

Features:

Automatic Detection:

- **Behavioral Analysis:** Monitors incoming traffic patterns and automatically detects anomalies that indicate potential DDoS attacks.
- **Machine Learning:** Utilizes machine learning algorithms to identify malicious traffic based on historical data.

Real-Time Mitigation:

- **Dynamic Policies:** Automatically creates security policies to block or rate-limit suspicious traffic in real-time without manual intervention.
- **Custom Rules:** Allows users to define custom rules tailored to specific applications and use cases.

HTTP Flood Protection:

- **Traffic Filtering**: Identifies and mitigates HTTP floods, which are a common form of layer 7 attacks aimed at overwhelming web applications.
- **Threshold-based Mitigation**: Automatically adjusts thresholds for requests based on normal traffic patterns, ensuring legitimate traffic is not disrupted**.**

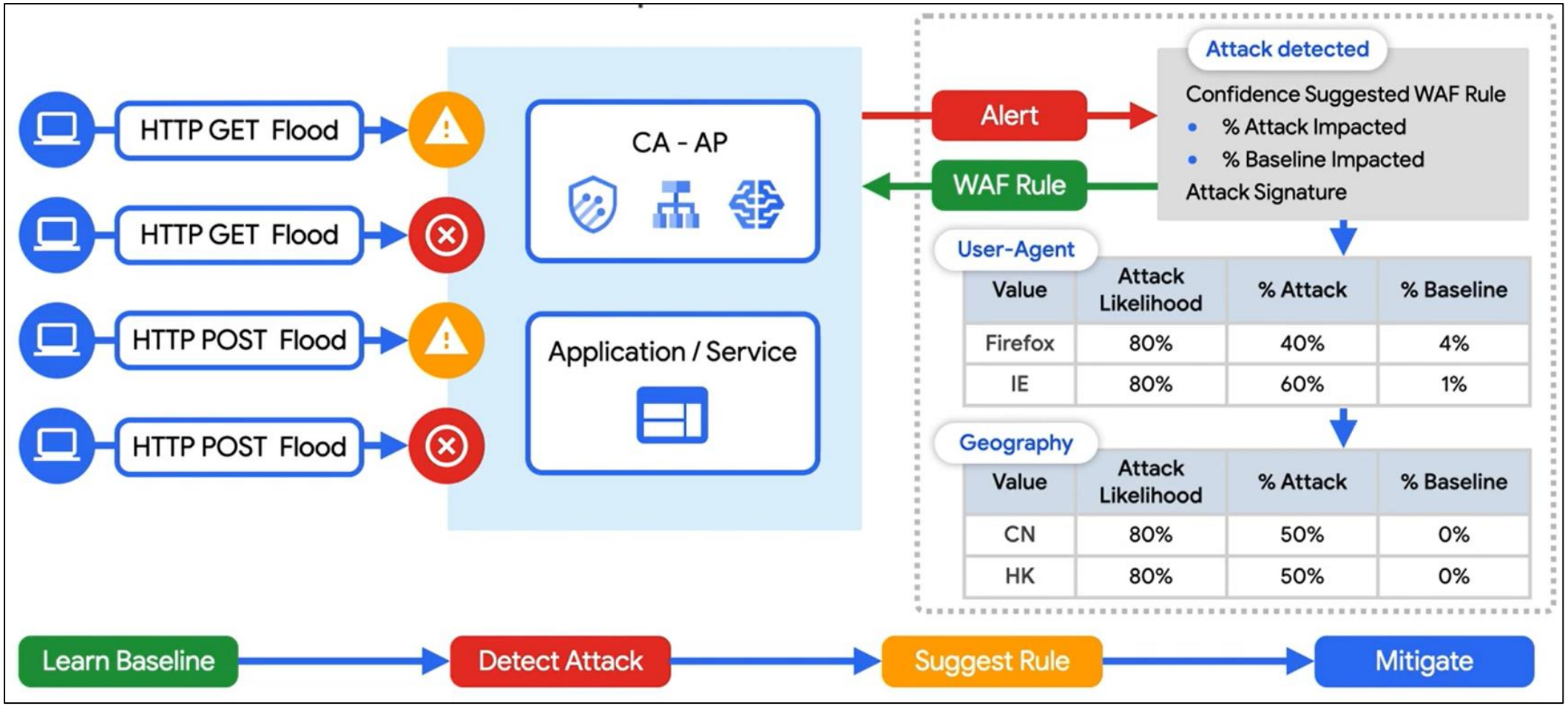
Insights and Reporting:

- **Detailed Reporting**: Provides insights into traffic patterns, attack vectors, and mitigation actions taken.
- **Alerts and Notifications**: Sends alerts based on detected threats and mitigation actions, keeping security teams informed.

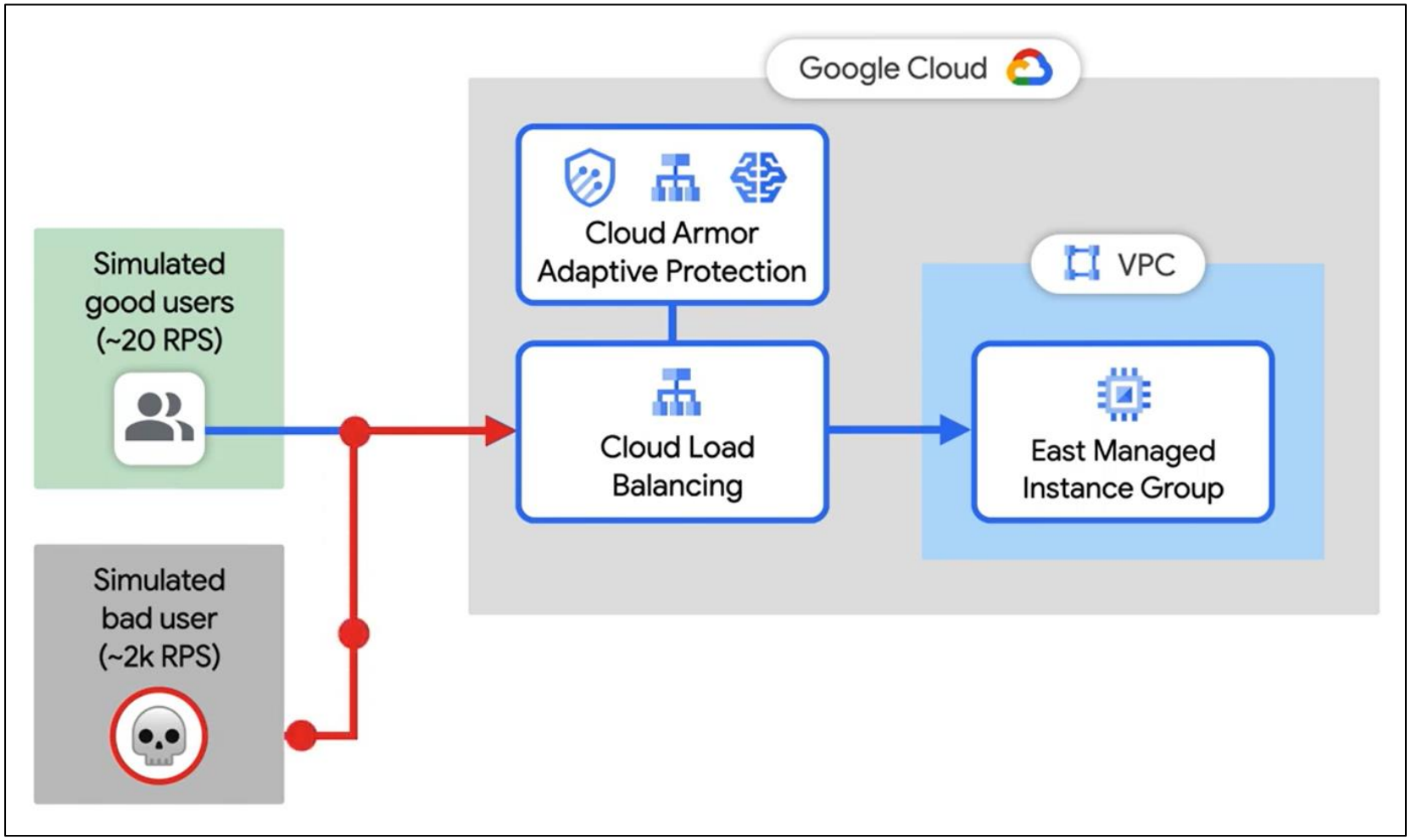
Integration with Google Cloud Services:

- **Seamless Integration**: Works seamlessly with other Google Cloud services such as Load Balancing and Identity-Aware Proxy to provide a comprehensive security solution.
- **Global Load Balancing Support**: Protects applications across multiple regions and environments.

Cloud Armor: Adaptive Protection



Cloud Armor: Adaptive Protection



Storage

GCP Cloud Storage

- Google Cloud Storage is easy-to-use *object storage* with a simple web service interface that you can use to store and retrieve any amount of data from anywhere on the web.
- Natively always online, access via HTTP, It's a Storage services for the internet.
- **Highly scalable**, reliable, fast and durable.
- Google Cloud Storage also allows you to pay only for the storage you actually use, which eliminates the capacity planning and capacity constraints associated with traditional storage.

GCP Cloud Storage

- Able to store an **unlimited number of objects** in a bucket.
- Designed for **99.999999999%** durability and **99.99%** availability of objects over a given year.
- **HTTP/S** endpoint to store and retrieve any amount of data, at any time, from anywhere on the web.
- Very highly available across all storage classes.
- Highly scalable (scalable to exabytes of data), reliable, fast, and inexpensive.
- Optional server-side **encryption** using GCP or customer- managed provided client-side encryption.
- GCP Cloud storage objects are automatically replicated on multiple devices in multiple facilities within a region.

GCP Cloud Storage

- Backup and archive for on-premises or cloud data
- Content, media, and software storage and distribution
- Big data analytics
- Static website hosting
- Software Delivery
- Store Custom images and Snapshots
- Cloud-native mobile and Internet application hosting
- Disaster Recovery

Note: Pay only for what you use, No minimum fee, Estimate monthly bill using the **Google Calculator**

GCP Cloud Storage

High-performance object storage

Backup and archival storage

HIGH FREQUENCY ACCESS

Multi-Regional

Most projects start with Multi-Regional Storage, which is optimized for **geo redundancy** and **end-user latency.**

Regional

Use Regional Storage when your project requires **higher performance local access** to computing resources — for example, when you need to support **high-frequency analytics workloads.**

LOW FREQUENCY ACCESS

Nearline

Nearline Storage is fast, highly durable storage for data accessed less than **once a month** .

LOWEST FREQUENCY ACCESS

Coldline

Coldline Storage is fast, highly durable storage for data accessed less than **once a year** .

GCP Cloud Storage

Available storage classes

The following table summarizes the primary storage classes offered by Cloud Storage. See class descriptions for a complete discussion.

Storage Class	Name for APIs and gsutil	Minimum storage duration	Typical monthly availability[1]
Standard Storage	`standard`	None	• >99.99% in multi-regions and dual-regions • 99.99% in regions
Nearline Storage	`nearline`	30 days	• 99.95% in multi-regions and dual-regions • 99.9% in regions
Coldline Storage	`coldline`	90 days	• 99.95% in multi-regions and dual-regions • 99.9% in regions

GCP Cloud Storage

Buckets: A *bucket* is a container (web folder) for objects (files) stored in Google Cloud Storage. Every object is contained in a bucket. Buckets form the top-level namespace for Google Cloud Storage and bucket names are global.

Bucket names can contain up to 63 lowercase letters, numbers, hyphens, and periods.

Multi Region or Regions: Google Cloud Storage buckets can be created in Multi-region or in specific region. You control the location of your data; data in any Google Cloud Storage bucket is stored in that region unless you explicitly copy it to another bucket located in a different region.

Objects: *Objects* are the entities or files stored in Google Cloud Storage buckets. A single bucket can store an unlimited number of objects.

GCP Cloud Storage

Object URL: Example:

https://storage.googleapis.com/Techlanders/gagan.doc

Techlanders is our bucket name, and gagan.doc is the key or filename.

If another object is created, for instance:

http://storage.googleapis.com/Techlanders/Solutions/gagan.doc then the bucket name is still Techlanders, but now the key or filename is the string Techlanders/Solutions/gagan.doc.

GCP Object Lifecycle

Object Lifecycle Management is roughly equivalent to automated *storage tiering* in traditional IT storage infrastructures. In many cases, data has a natural lifecycle, starting out as “hot” (frequently accessed) data, moving to “warm” (less frequently accessed) data as it ages, and ending its life as “cold” (long-term backup or archive) data before eventual deletion.

For example, the lifecycle rules for backup data might be:

Store backup data initially in Google cloud Standard storage.
After 30 days, transition to Nearline.
After 90 days, transition to Coldline.
After 3 years, delete.

GCP Coldline Storage

Long term low-cost archiving service.

Optimal for infrequently accessed data.

Designed for 99.999999999% durability.

3-5 hours retrieval time.

Less than $0.01 per GB / month (depending on region).

Coldline can store an unlimited amount of virtually any kind of data, in any format.

Cloud Filestore

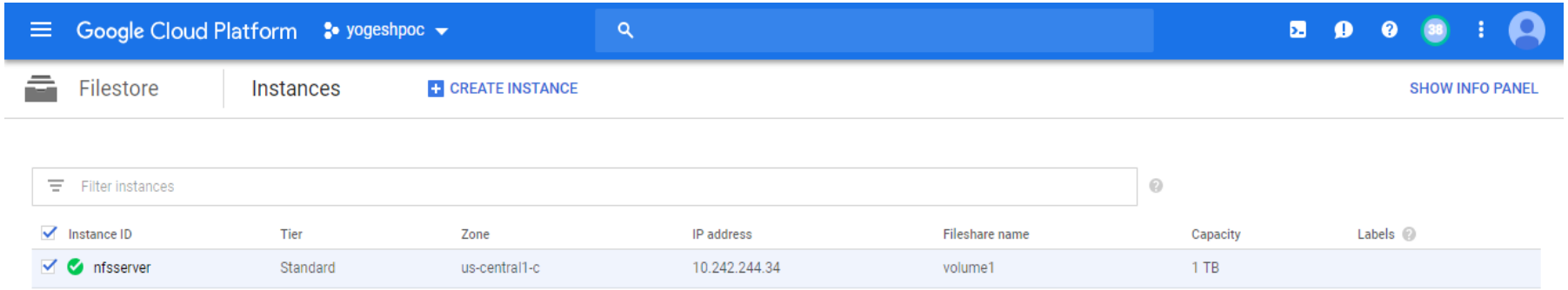
Cloud Filestore is a fully managed NFS file servers on Google Cloud Platform (GCP)

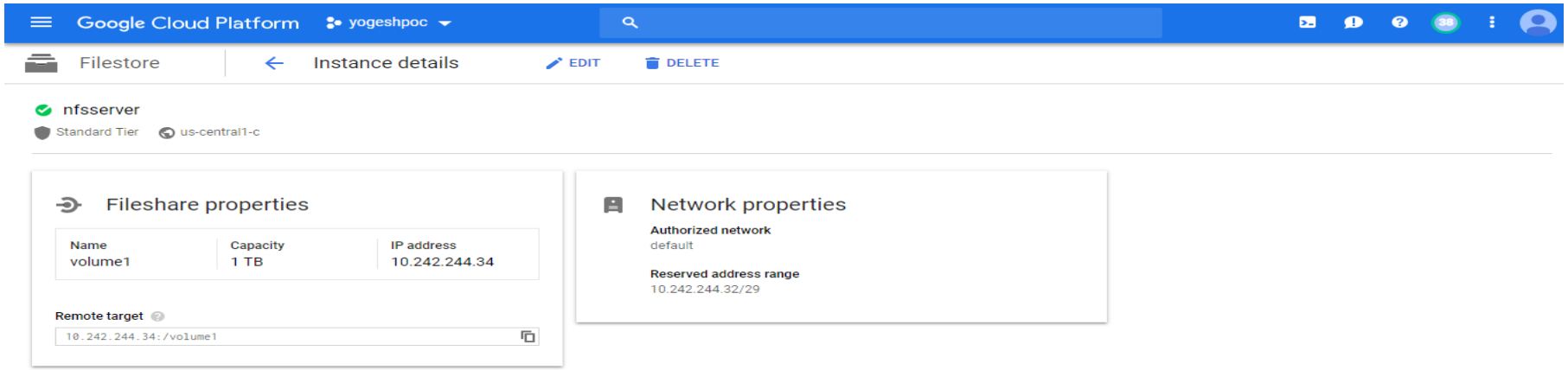
Follow below process to create NFS server and share it with the GCE compute instance for shared storage

Create a filestore server
Create one or two instance in the same zone where you have created your fileserver.

Note: don’t forget to allow http traffic
Install “nfs-common” package on newly created GCE
Mount your filestore on the instance using “mount <IP>:</sharename> <Mount-Point>
Put some data and test the shared data on both servers

Cloud Filestore





Cloud Filestore

Create a filestore server using GCP console (nfs-server)

Create one instance in the same zone (nfs-client)

Mount the shared storage and check the shared mount point

Gcloud CLI

Gcloud CLI

- Command line Interface for Automation.
- The gcloud command-line interface is a tool that provides the primary CLI to Google Cloud Platform.
- For example, you can use the gcloud CLI to create and manage:
  - Google Compute Engine virtual machine instances and other resources
  - Google Cloud SQL instances
  - Google Kubernetes Engine clusters

Command group	Description
`gcloud compute`	Commands related to Compute Engine in general availability
`gcloud compute instances`	Commands related to Compute Engine instances in general availability
`gcloud beta compute`	Commands related to Compute Engine in Beta
`gcloud alpha app`	Commands related to managing App Engine deployments in Alpha

Gcloud Cloud Shell

- Browser based shell with CLI
- Full Power Access From Anywhere
- Secure and Fully Authenticated by default
- Developers will enjoy access to all their favorite development tools pre-configured. You'll find Java, Go, Python, Node.js, PHP and Ruby development and deployment tools.
- Cloud Shell provisions 5GB of persistent disk storage mounted as your $HOME directory on the Cloud Shell instance. All files you store in your home directory, including scripts and user configuration files like .bashrc and .vimrc, persist between sessions.

Gcloud CLI

- gcloud compute instances list
- gcloud compute instances create gaganvm --zone="us-west1-b" --boot-disk-size=15
- gcloud compute images list
- gcloud compute instances create gaganvm1 --zone="us-west1-b" --boot-disk-size=15 --image=centos-7-v20190813 --image-project=centos-cloud
- gcloud config set compute/zone NAME

Gcloud CLI

Managing Buckets:

- Create bucket: gsutil mb -b on -l us-east1 gs://my-awesome-bucket/
- Copy to bucket: gsutil cp Desktop/kitten.png gs://my-awesome-bucket
- Download from bucket: gsutil cp gs://my-awesome-bucket/kitten.png Desktop/kitten2.png
- List content of bucket: gsutil ls gs://my-awesome-bucket
- List object details: gsutil ls -l gs://my-awesome-bucket/kitten.png
- Making object publicly accessible: gsutil iam ch allUsers:objectViewer gs://my-awesome-bucket
- Removing bucket: gsutil rm -r gs://my-awesome-bucket

https://cloud.google.com/storage/docs/quickstart-gsutil#create

Identity & Access Management

IAM

GCP Identity and Access Management (IAM) is a web service that helps you securely control access to GCP resources for your users. You use IAM to control who can use your GCP resources (*authentication*) and what resources they can use and in what ways (*authorization*).

Google Cloud Platform (GCP) offers **Cloud IAM, which lets you manage access control by defining who (identity) has what access (role) for which resource.**

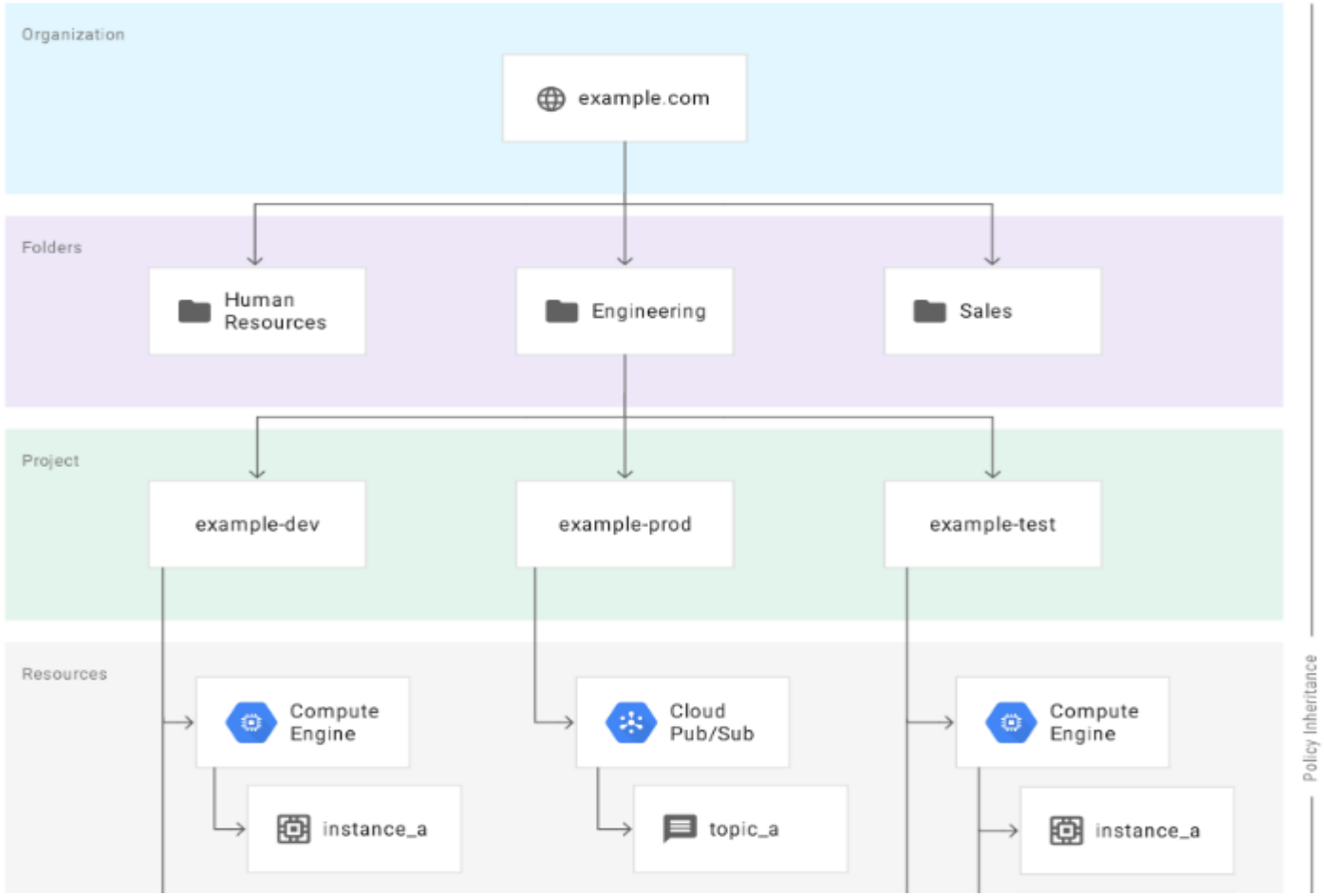
Granular permissions

Secure access to GCP resources for applications that run on GCP VM's

Integrated with many GCP services

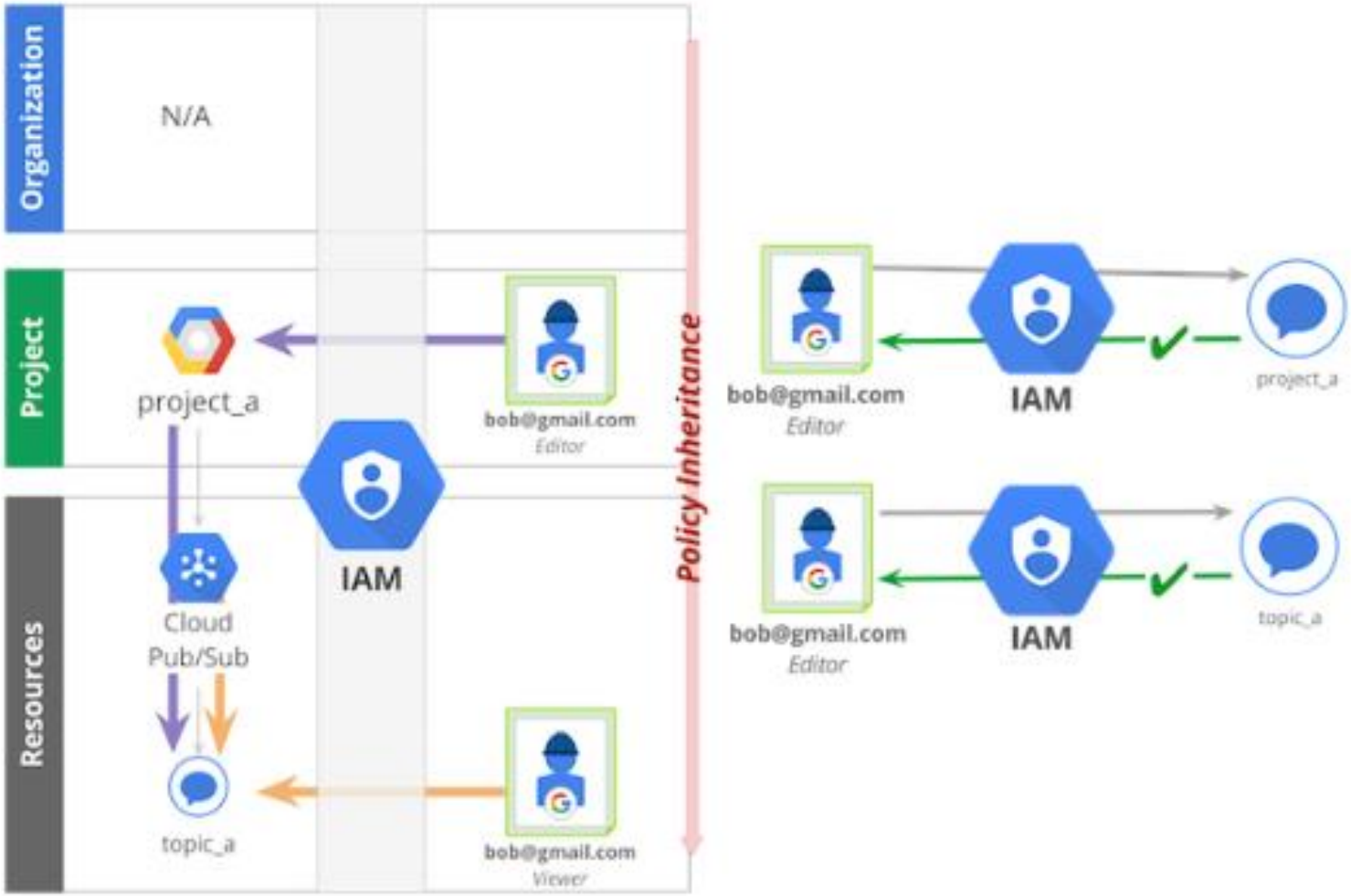
Free to use

IAM Hierarchy



https://cloud.google.com/iam/docs/resource-hierarchy-access-control

IAM Hierarchy



Organisation

- Organisation contains all billing accounts. (For personal account we will not be getting an organisation and folders).
- Project is associated with one billing account.
- A resource belongs to one project only.

Benefit of the Organisation Resource:

With an organisation resource, projects belong to your organisation instead of the employee who created the project. This means that the projects are no longer deleted when an employee leaves the company.

You can grant roles at the organisation level which are inherited by all projects and folders under the organisation resource.

For Example: you can grant the network admin role to your networking team at organization level, allowing them to manage all the networks in all projects in your company, instead of granting them the role for all individual projects.

Folders

Folders:

Folders resources provides an additional grouping mechanism and isolation boundaries between projects.

Folders can contain another folder but it is optional. You can create folders or opt out of making any folders based on the architecture.

Projects

Projects:

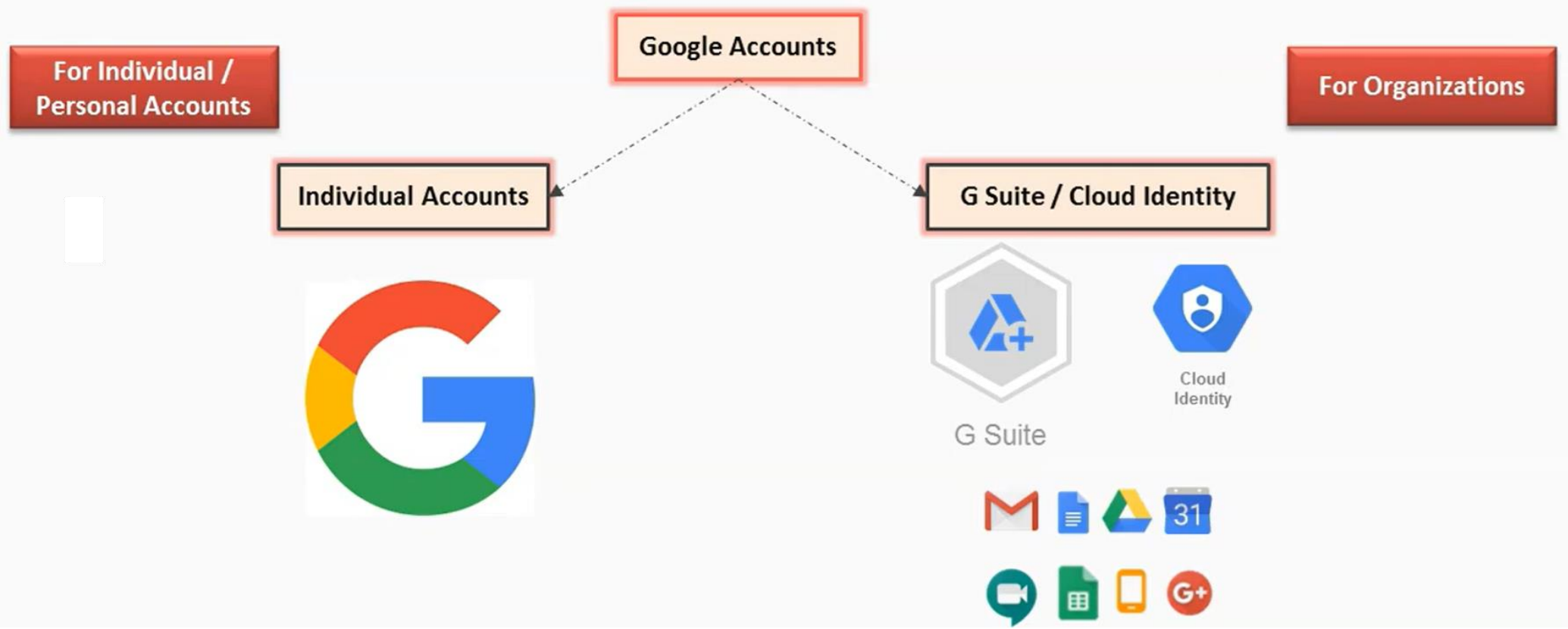
- All GCP resources, we are associated to one specific project.
- You can track resources and Quota usage
- Enable billing and set budget
- Manager permissions and credentials
- Project is a global entity
- Enable services and API
- Equivalent to account in AWS and Subscription in Microsoft Azure.

Projects

Project has three components:

1. **Project ID** (it might be the same as project name if the id is not taken. If you make a project name and you delete it then you will not get the same project ID again as it has been used already even if it is deleted. GCP will add any number automatically to your project name to give an ID to you or you can make your own ID while creating the project. Once it has been created, you can't change the ID.)
2. **Project Name** (small letters only and can be changed after creating it as well unlike Project ID)
3. **Project Number** (12-digit number which has been assigned by GCP automatically)

Access Google Account



IAM

To create organization in GCP, you need to have :

IAM

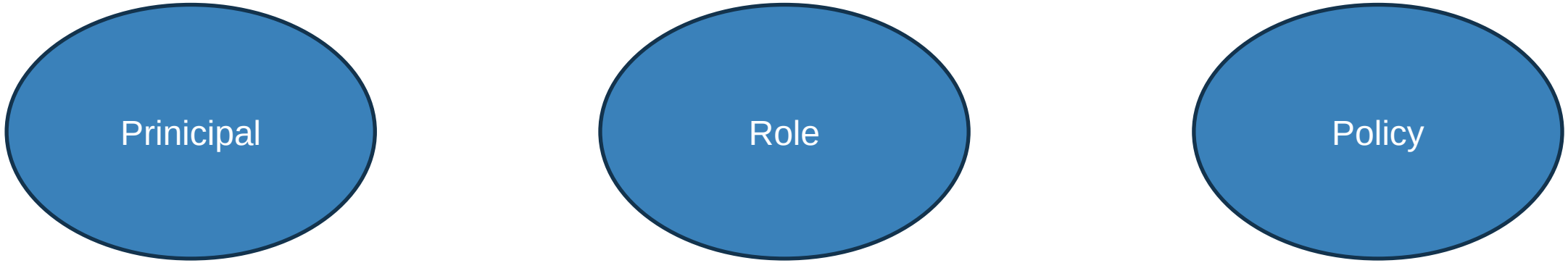
1. IAM lets you grant granular access to specific Google Cloud resources and helps prevent access to other resources.
2. IAM lets you adopt the securely principle of least privilege, which states that nobody should have more permissions than they actually need.
3. With IAM, you manage access control by defining '**Who**' (Identity) has what access (**role**) for which resources.
4. IAM, permission to access a resource is not granted directly to the end user. Instead, permissions are grouped into roles and roles are granted to authenticated Principals (Members).
5. Policy defines and enforces what roles are granted to which principals. Each allow policy is attached to the resource.
6. When an authenticated principal attempts to access a resource, IAM checks the resource's allow policy to determine whether the action is permitted.
7. Person should have valid Gmail address which is checked by Google for Authentication, Authorization and Auditing.

Features of IAM

- Let you authorize who can take specific actions on resources to give you full control and visibility on your google cloud services centrally.
- Permissions are represented in the form of service resource verb.
- Can map job functions into groups and roles
- With IAM, users only get access to what they need to get the job
- Cloud IAM enables you to grant access to cloud resources at fine-grained levels, well beyond project level access

IAM

IAM has three important parts



IAM: Principal

1. Principal/Members- A principal can be a

- Google account
- Service account
- Google group (a collection of Google accounts and Service accounts)
- Google workspace (virtual account associated with your organisation)
- Cloud identity domain (like a google workspace but it does not have access to Google Workspace applications and features)
- Authenticated users (the value allAuthenticatedUsers is a special identifier that represents all service accounts and all Users on the internet who are authenticated with Google account. It does not include Google workspace or Cloud identity domain)
- All users

IAM: Role

1. A role is a collection of Permissions. Permissions determine what operations are allowed on a Resource when you grant a role to a principal, you grant all the permission that the role contains.

We have 3 roles which are basic, predefined and custom roles.

1. A role contains a set of permissions that allows you to perform specific actions on Google cloud resources.
2. You do not directly grant users permissions in IAM, instead you grant them roles which bundle one or more permissions.
3. To make permissions available to members including users, group and service accounts, you grant roles to the members

IAM: Role

Basic/Primitive roles:

IAM basic roles offer fixed, coarse grained levels of access

- **Owner**: super user at project level, add or remove members, delete and create projects. Setup billing for project.
- **Editor**: modify codes, deploy apps, configure service, stop and start service
- **Viewer**: read only access
- **Billing Admin**: manage billing, add or remove administrators
- **Browser**: it cannot view the details like how many projects or organizations or roles. It does not have cloud storage access.

IAM: Role

Predefined Roles:

- Provides granular access for a specific service and is managed and defined by google cloud.
- Prevents unwanted access to other resources.
- Google is responsible for updating and adding permissions as necessary
- You can grant multiple roles to the same users

IAM: Role

Custom Roles:

- Provides granular access according to a user-defined list of permissions.
- You can create a custom IAM role with one or more permissions and then grant that custom role to users and groups
- Custom roles are not maintained by Google
- You can grant multiple roles to a user or a group.
- **For example: Allow user to view, stop, start the VM but do not allow to delete it.**
  - **Role:** compute viewer
  - **Role with Permissions:** compute.instances.stop, compute.instance.start

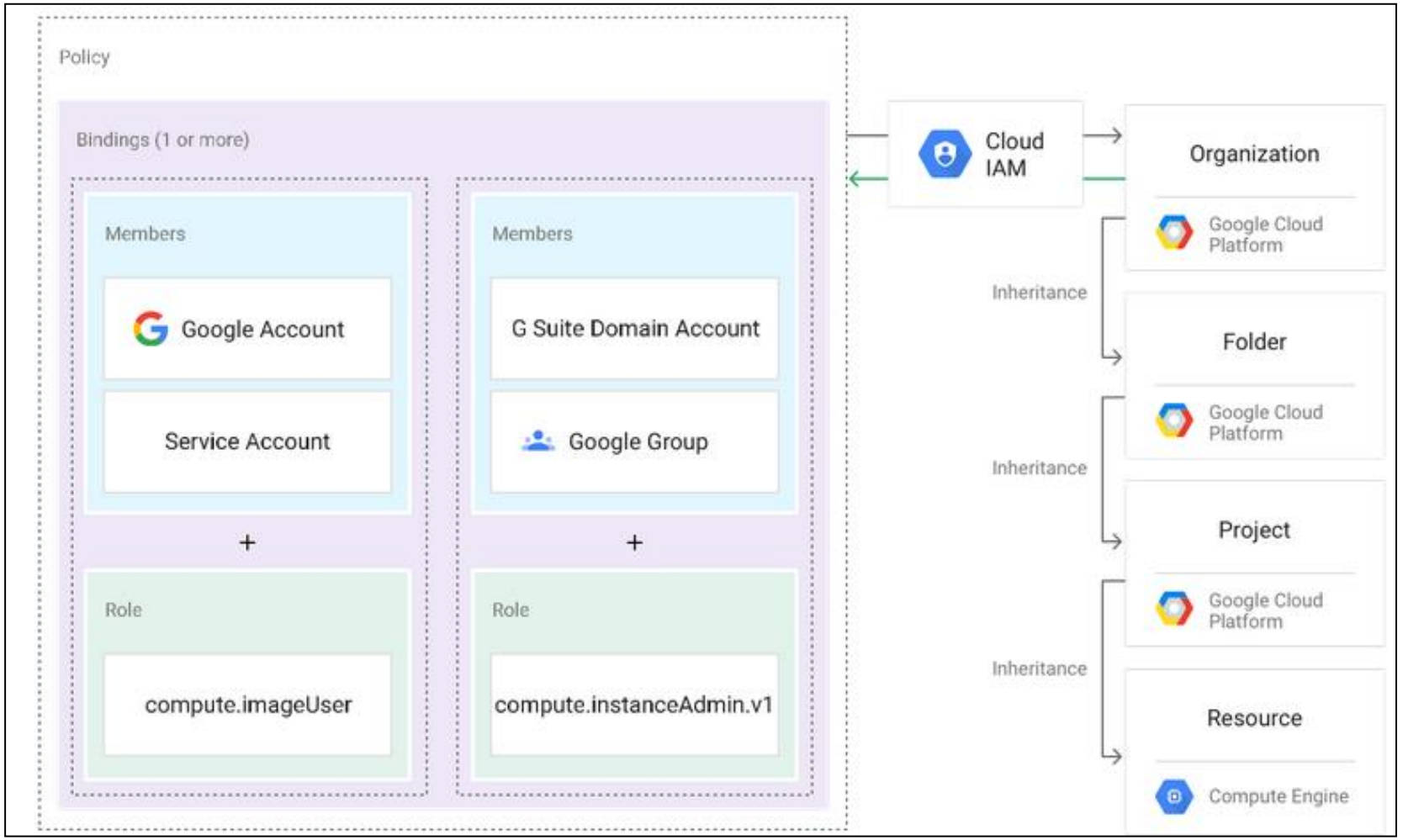
IAM: Policy

Policy is a collection of role bindings that bind one or more principals to individual roles when you want to define who has what type of access on a resource, you create allow policy and attach a resource.

Policy = Role + Permissions

- A policy is a collection of bindings, audit configuration and metadata.
- A binding associates one or more members with a single role and any context-specific conditions that change how and when the role is granted.
- Each binding includes the following fields
    1. A member, known as an Identity or principal, can be a User account, service account, Google group, Domain
    2. A role which is named collection of permissions that grant access to perform actions on Google Cloud Resources.
    3. A condition which is a logical expression that further constraints the role binding based on attributes about the request, such as its origin the target resource etc.

IAM: Policy


Policy
Bindings (1 or more)
Members
Google Account
Service Account
+
Role
compute.imageUser
Members
G Suite Domain Account
Google Group
+
Role
compute.instanceAdmin.v1
Cloud IAM
Organization
Google Cloud Platform
Inheritance
Folder
Google Cloud Platform
Inheritance
Project
Google Cloud Platform
Inheritance
Resource
Compute Engine

IAM: Service Account

- A service account is a special kind of account used by an application or a virtual machine not a person.
- Applications use service accounts to make authorized API calls, authorized as either:
  - The service account itself
  - As Google workspace
  - As cloud identity users though domain-wide delegation
- A service account is identified by its email address which is unique to the account.
  - **saname@project.iam.gserviceacc.com**
- Each service account is associated with two sets of **public/private RSA key pairs** used to authenticate to google.

IAM: Service Account

Difference between Service account and User account:

1. Service account does not have passwords, and can not log in via browsers or cookies.
2. Service accounts are associated with public/private RSA key pairs that are used for authentication to Google and for signing data.
3. You can let other users or service account impersonate a service account
4. Service account does not belong to your Google Workspace domain unlike user accounts.

IAM: Types of Service Account

1. Default Service account:

When you enable or use some google cloud services, they create a default service accounts that enable the service to deploy jobs that access other Google cloud resources. These accounts are known as Default Service account.

These two services create service account App engine, compute engine and any google cloud service that uses App Engine or Compute engine

2. User Managed Service Account:

you can create user-defined Service account in your project using IAM API, console and CLI.

By default, you can create 100 user managed SA in a project.

When you create a user managed SA in your project, you choose a name for the service account which uses the following format.

SAname@project-id,iam.gserviceaccount.com

3. Google Managed Service Account:

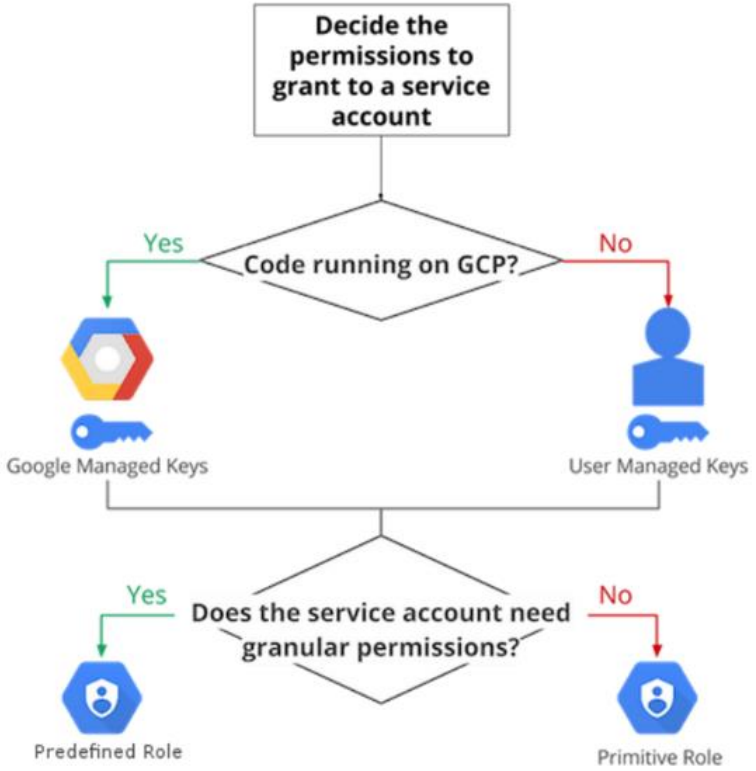
Some google cloud services needs access to your resources so that they can act on your behalf, for eg, when you use Cloud Run to run a container, the service needs access to any Pub/Sub topics that can triggers the container.

To meet these needs google create Google managed SA, it is not listed in the SA page.

IAM: Service Account keys

Type of keys for Service Account:

1. **Google Managed Key Pair**: this is used by service account credentials API, and by Google cloud services such as App Engine & Compute engine.
2. **User Managed Key pairs**: you can create user-managed key pairs for a service account, then use the private key from each key pair to authenticate with Google APIs. This private key is known as Service account key. Each service account can have upto 10 SA keys. User is responsible for the security of private key.



IAM: Service Account keys

Lab:

- Create a service account with a role attached like cloud storage admin.
- Create a machine
- Unset the account if there is any
  - gcloud config list
  - gcloud config configurations list
  - gcloud config unset account
- Install google cloud storage python client
  - pip install google-cloud-storage
- Create the key for the service account and download in a Json Format
- Set it as environment variable in the machine
  - export GOOGLE_APPLICATION_CREDENTIALS="techlanders-internal-49c06afab916.json“
- Run the python program

IAM: Service Account keys

Lab:

Demo.py

```
from google.cloud import storage

def list_buckets():
    """Lists all buckets in the Google Cloud project."""

    # Initialize the Cloud Storage Client
    client = storage.Client()

    # Get a list of all buckets in the project
    buckets = client.list_buckets()

    print("Buckets in your project:")
    for bucket in buckets:
        print(bucket.name)

if __name__ == "__main__":
    # Call the function to list the buckets
    list_buckets()
```

IAM: Access Scope

Access scopes are an older mechanism used to limit the API access a VM instance has when it interacts with Google Cloud services. These scopes act as a secondary layer of security that restricts the service account's access to certain Google Cloud APIs.

Key Points about Access Scopes:

Permissions Limitation: Access scopes define what level of API access a VM instance's service account has. Even if the service account has broad roles assigned, access scopes can further restrict access.

Common Scopes:

- Full Cloud Platform Access: Grants access to all Google Cloud APIs.
  - https://www.googleapis.com/auth/cloud-platform
- Compute Engine Access: Allows the instance to manage Compute Engine resources.
  - https://www.googleapis.com/auth/compute
- Storage Access: Allows full control over Google Cloud Storage.
  - https://www.googleapis.com/auth/devstorage.full_control
- Specific vs. Full Access: Instead of granting full access to Google Cloud APIs, you can choose to provide access only to the specific APIs required for your VM, such as Compute Engine or Cloud Storage.

IAM: Access Scope

Set During VM Creation or Modification: Access scopes are typically set when creating a VM. They can be updated later by stopping the VM and modifying the instance through the Google Cloud Console or via the command line.

Example Use Case:

A VM instance with a service account attached might have the "Storage Admin" role, but if the access scope is restricted to "read-only" for Cloud Storage, the instance won't be able to write to Cloud Storage even though the role permits it.

IAM: Access Scope

Service Account vs. Access Scopes:

Service Account: Dictates which Google Cloud services the VM instance or application is allowed to interact with, based on the roles assigned to it.

Access Scopes: Further limits the actions that the service account can take when interacting with specific APIs.

In modern Google Cloud practice, using Identity and Access Management (IAM) roles is preferred over relying solely on access scopes. However, scopes are still relevant when working with VM instances.

Example Scenario:

- You create a VM instance to run an application that needs to access Cloud Storage.
- You assign a service account to the instance, giving it the "Storage Admin" role.
- You configure access scopes to limit the VM’s access to Cloud Storage APIs (e.g., full access or read-only access).
- By configuring both correctly, you ensure that the VM instance has the exact permissions it needs without being overly permissive.

IAM: Lab

Create a Virtual Machine (VM)

Objective: Create a VM instance and attach the service account to it.

Steps:

- Go to the Compute Engine > VM Instances page.
- Click Create Instance.
- Configure the instance details (choose machine type, region, etc.).
- Under the Identity and API access section:
  - **Service Account**: Select the service account created in step 1 (vm-service-account).
  - **Access Scopes**: Choose Set access for each API and limit the access scopes to:
    - **Compute Engine API**: Set to "Read Write" (https://www.googleapis.com/auth/compute).
    - **Cloud Storage API**: Set to "Full Access" (https://www.googleapis.com/auth/devstorage.full_control).
    - **Cloud Platform API**: Enable
- Click Create to launch the VM instance.

IAM: Lab

Commands:

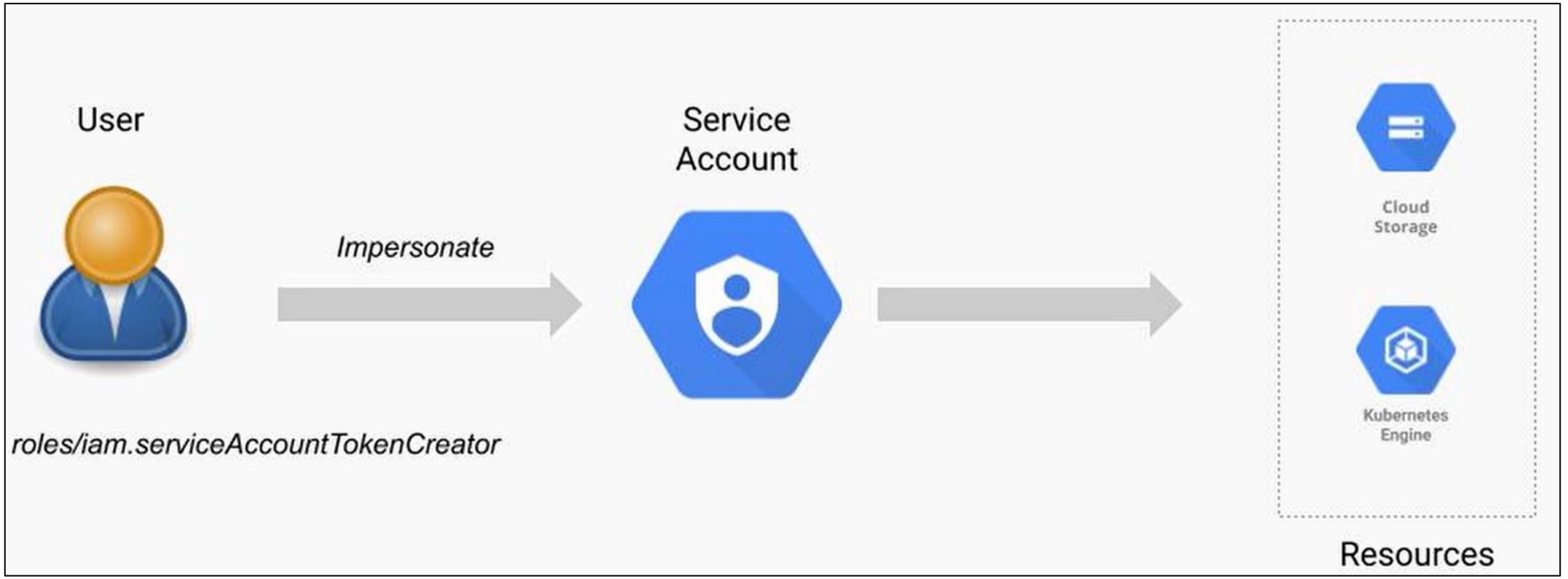
Operation	Command
List Storage Buckets	gsutil ls
Create a Storage Bucket	gsutil mb gs://<bucket-name>
List Objects Inside a Storage Bucket	gsutil ls gs://<bucket-name>
List Compute Engine Instances	gcloud compute instances list
Stop a Compute Engine Instance	gcloud compute instances stop <instance-name> --zone=<zone>
Start a Compute Engine Instance	gcloud compute instances start <instance-name> --zone=<zone>
Create a Compute Engine Instance	gcloud compute instances create <instance-name> --zone=<zone> --machine-type=<machine-type> --image-project=<image-project> --image-family=<image-family> --service-account=<service-account-email>

IAM: Service Account Impersonation

Impersonation:

- When an authenticated principal, such as a user or another service account authenticates as a service account to gain the service account's permissions, it is called impersonating the service account
- Impersonating a service account lets an authenticated principal access whatever the service account can access
- Impersonation is useful when you want to change a user's permissions without changing your identity and access management policies
- Limit user account permissions
- Reduces the risk of service account keys
- Easy maintenance by removing user from service account resource.

IAM: Service Account Impersonation



IAM: Service Account Impersonation

Lab:

1. Create or Select a Service Account
    - Go to the IAM & Admin section in the GCP Console.
    - Click on Service accounts.
    - Either select an existing service account or create a new one.
2. Grant Permissions to Impersonate the Service Account
    - Select the Service Account: Click on the service account you want others to impersonate.
    - Add IAM Policy Binding:
        - Click on the Permissions tab.
        - Click on the Add Principal button.
        - In the New Principals field, enter the email address of the user or service account that will be allowed to impersonate this service account.
        - In the Role dropdown, select Service Account > **Service Account Token Creator**.
    - Click Save.

Note: The user or the service account doing the impersonation should have **Service Account Token Creator** role.

IAM: Service Account Impersonation

Lab:

- Create a machine and authenticate with your user
  - gcloud auth login
- Now use the impersonation
  - gsutil -i sa1-188@techlanders-internal.iam.gserviceaccount.com ls
  - gcloud projects list --impersonate-service-account=sa1-188@techlanders-internal.iam.gserviceaccount.com
  - gcloud config set auth/impersonate_service_account sa1-188@techlanders-internal.iam.gserviceaccount.com
  - gcloud auth print-access-token --impersonate-service-account=sa1-188@techlanders-internal.iam.gserviceaccount.com

IAM: Service Account Impersonation

Use Case:

In a Continuous Integration/Continuous Deployment (CI/CD) pipeline, you have a service account specifically created for deployment tasks. This service account has the necessary permissions to deploy applications to Google Cloud (e.g., deploying to Google Kubernetes Engine or Google App Engine) but does not have full project access for security reasons.

You want your CI/CD pipeline to run with the permissions of this deployment service account, while also allowing developers (who have broader permissions) to trigger deployments without needing full access to production resources.

Set up Permission:

- Developers have the Editor role on the project, which allows them to manage most resources but not the production deployments.

Enable Service Account Impersonation:

- Grant the developers the **roles/iam.serviceAccountTokenCreator** role on the ci-cd-deployment-sa service account. This allows them to impersonate this service account.

Organization Policy

Google Cloud Platform (GCP) Organization Policies allow organizations to define and enforce governance across their resources by setting constraints that control behavior in GCP environments.

Organization policies provide a way to centrally manage settings at different levels of the GCP resource hierarchy (Organization, Folders, Projects) and ensure compliance with security, operational, and regulatory requirements.

Benefits of GCP Organization Policies:

- **Centralized Governance**: Easily manage and enforce policies across the entire organization or specific projects, ensuring consistency and compliance.
- **Fine-grained Control**: Apply policies to specific resources and services, allowing detailed governance of how the GCP environment is managed.
- **Security and Compliance**: Ensure that security best practices and regulatory requirements are adhered to across the organization.
- **Cost Management**: Prevent excessive resource usage by restricting the types of services and resources available.

Organization Policy Concepts

Resource Hierarchy:

- GCP resources are organized into a hierarchy: Organization > Folders > Projects > Resources (e.g., Compute Engine instances, Cloud Storage buckets).
- Organization policies can be applied at any level (Organization, Folder, or Project) in the hierarchy. Policies set at higher levels can be inherited by lower levels, providing centralized governance.

Policy Constraints:

- Constraints are predefined rules that govern specific behaviors, such as which services can be used, what types of resources can be created, and which regions resources can be deployed to.
- Constraints are applied by enabling or disabling them at various levels in the hierarchy.

Policy Inheritance:

- Policies applied at the Organization level are inherited by Folders and Projects under it.
- Override: You can override the inherited policy at a lower level, but this depends on the enforcement of the policy (some constraints cannot be overridden).

Enforcement and Monitoring:

- Organization policies enforce governance without needing to change existing code or infrastructure.
- Violations of policies (such as creating resources in unauthorized regions) will result in errors or denied requests, ensuring compliance with the set policies.

Organization Policy Use Cases

Security Control:

- Restrict which types of virtual machines (VMs) can be deployed (e.g., only allow Shielded VMs).
- Block certain APIs from being enabled across projects (e.g., disabling the use of unsafe APIs).

Location Control:

- Limit the geographic locations (regions) where resources can be deployed (e.g., restrict resources to specific regions to comply with data sovereignty laws).
- Ensure that storage, databases, or other services are only created in approved locations.

Operational Governance:

- Enforce the use of required settings for services (e.g., require the use of customer-managed encryption keys for certain services).
- Prevent the use of public IPs for certain types of resources (e.g., restrict public IPs on Compute Engine instances).

Cost Control:

- Limit the services or resource types that can be used (e.g., prevent the creation of high-cost resources such as certain VM types).
- Enforce quotas and limit resource consumption to control costs at the project level.

Compliance Requirements:

- Enforce the use of specific security configurations (e.g., require multi-factor authentication for certain projects).
- Ensure compliance with industry regulations such as HIPAA, GDPR, or PCI DSS by enforcing governance on data storage and access.

Organization Policy Concepts

Restrict Cloud APIs:

- Constraint: constraints/serviceuser.services
- Use case: Prevent certain APIs from being enabled (e.g., disable APIs that should not be used in production).

Restrict VM Machine Types:

- Constraint: constraints/compute.allowedMachineTypes
- Use case: Limit the types of virtual machine instances that can be created, such as restricting to only cost-effective instance types.

Restrict Resource Locations:

- Constraint: constraints/gcp.resourceLocations
- Use case: Control the geographic regions where resources can be deployed (e.g., only allow resource creation in "us-central1" and "us-east1").

Organization Policy Concepts

Require Secure SSL Policies:

- Constraint: constraints/compute.requireOsLogin
- Use case: Enforce the requirement of SSL/TLS connections for network services.

Disable External IP Addresses:

- Constraint: constraints/compute.vmExternalIpAccess
- Use case: Prevent VMs from obtaining external IP addresses to reduce the attack surface.

Enforce Use of Customer-Managed Encryption Keys (CMEK):

- Constraint: constraints/gcp.restrictCmekCryptoKeyProjects
- Use case: Ensure that all cloud storage and databases use encryption with customer-managed keys.

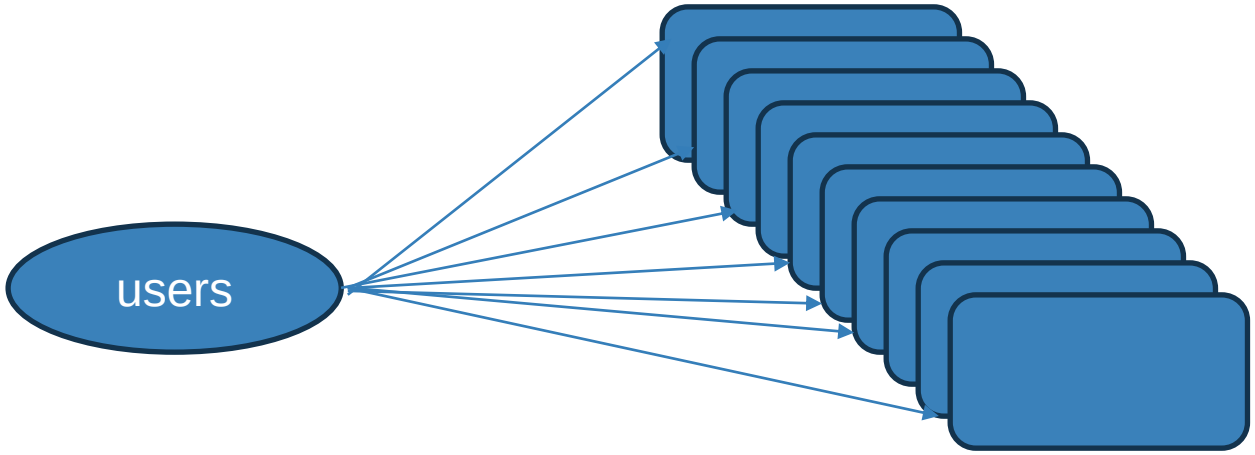
Cloud Identity

Lets assume we have many applications and for a user would be accessing the applications using the separate credentials. We might face issues in below activities:

- User lifecycle management (Provisioning)
- Authentication
- Administrator
- User lifecycle management (De-Provisioning)

Further end up with:

- Time Consuming
- Not Secure
- Poor User Experience



Cloud Identity & G Suite Editions

Google offers several editions of Cloud Identity and Google Workspace (formerly G Suite) tailored to different needs for managing users, applications, and devices. Here's a breakdown of the editions:

1. Cloud Identity Editions

Cloud Identity is a standalone Identity-as-a-Service (IDaaS) solution for managing users, groups, and devices.

a. Cloud Identity Free Edition

- Core Identity Features: Manage users, groups, and devices.
- SSO (Single Sign-On): Use SSO to access third-party applications.
- Multi-Factor Authentication (MFA): Add a layer of security to logins.
- Device Management: Manage and monitor devices accessing your services.
- Basic Reports: Track activity and manage security alerts.

b. Cloud Identity Premium Edition

- All Free Features plus:
- Advanced Device Management: Better control over mobile devices (e.g., device wipe, managed configurations).
- Advanced Security Features: Access to Security Center, and more detailed reports, and insights on risks.
- Context-Aware Access: Control access to apps based on user identity and device.
- App Management: Advanced control over third-party apps.
- Enhanced SSO and MFA Options: Includes additional security controls and policies for authentication.

Cloud Identity & G Suite Editions

2. Google Workspace (formerly G Suite) Editions

Google Workspace includes Cloud Identity features along with a suite of collaboration and productivity tools such as Gmail, Google Docs, Sheets, and Google Drive.

a. Google Workspace Business Starter

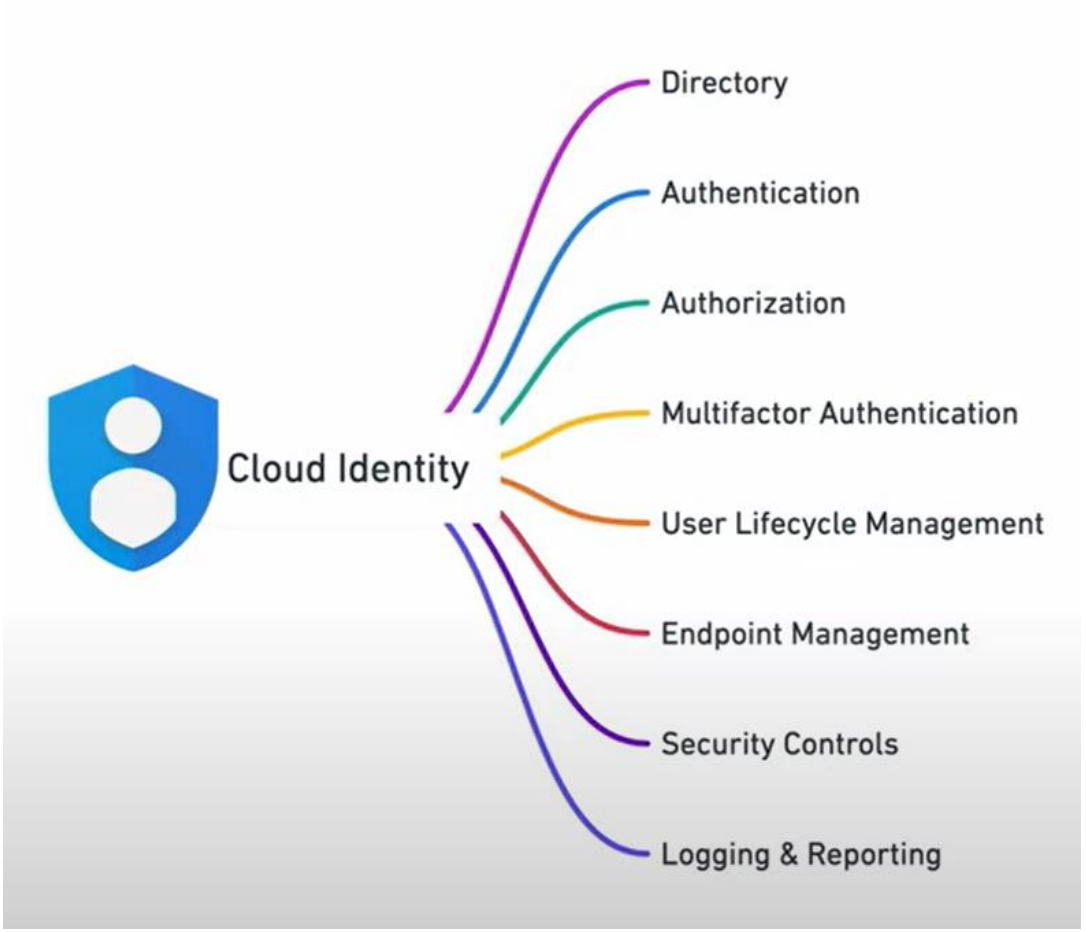
- Professional Email: Business email through Gmail (30 GB storage per user).
- Collaboration Tools: Access to Docs, Sheets, Slides, Meet, Chat, and Calendar.
- Cloud Storage: 30 GB storage per user (Gmail and Drive combined).
- Basic Security and Admin Controls: Basic device management and security features.

b. Google Workspace Business Standard

- Everything in Business Starter, plus:
- Increased Cloud Storage: 2 TB storage per user.
- Enhanced Collaboration: More participants allowed in Google Meet (150 participants).
- Shared Drives: Team collaboration with shared drives.
- Advanced Reporting and Admin Tools: More detailed reporting and admin features.

Cloud Identity

Google Cloud Identity is an identity as a Service (IDaaS) or cloud-based identity management service provided by Google Cloud Platform (GCP). It enables organizations to manage users, devices, and apps in a unified way. Cloud Identity helps you control access to GCP services and resources by providing a central system for authentication and authorization.



Apps

Cloud Identity

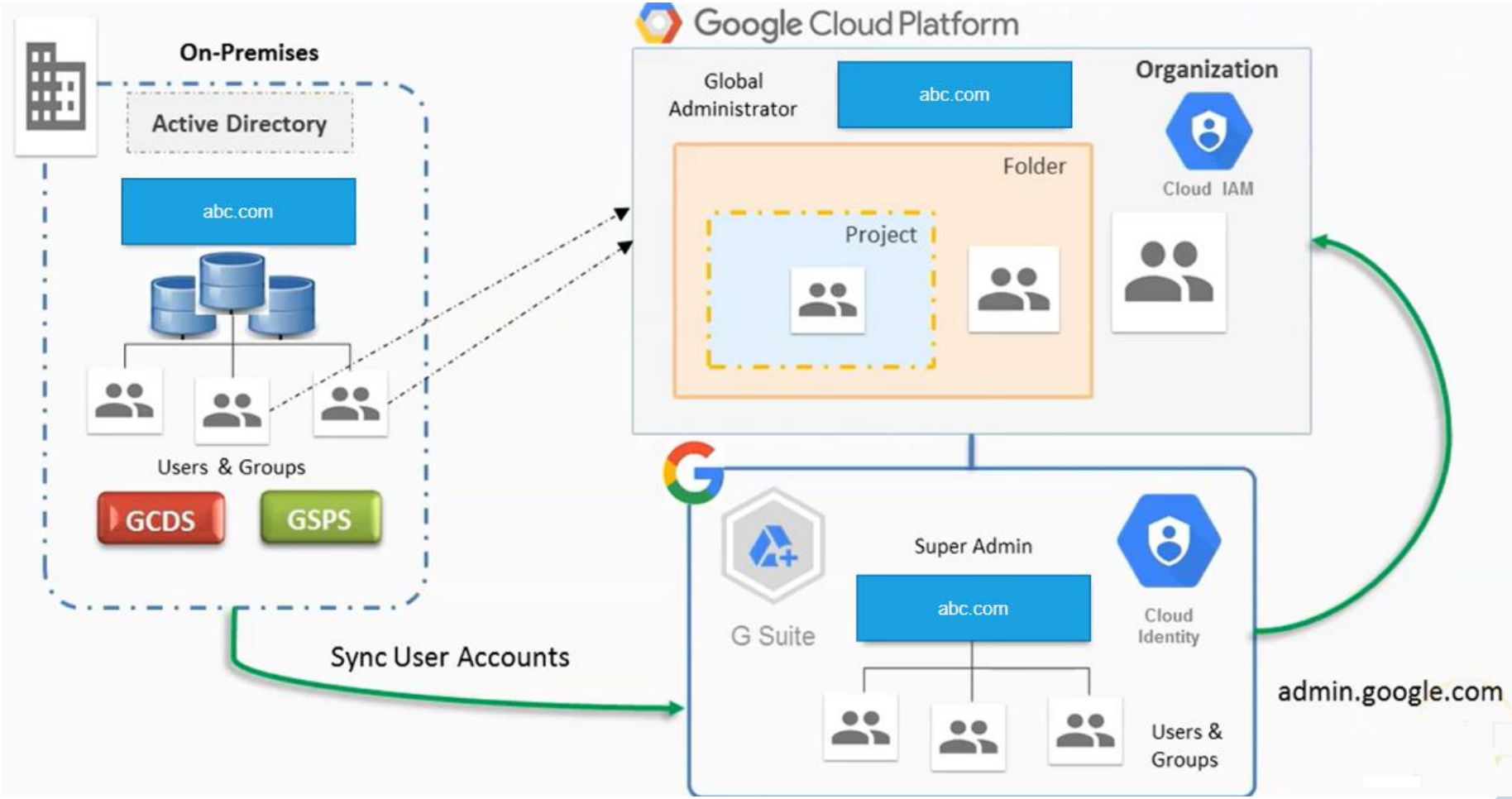
Key Features of GCP Cloud Identity:

- **Identity and Access Management (IAM):** It integrates with GCP's IAM to manage permissions and control access to cloud resources.
- **Single Sign-On (SSO):** Provides SSO capabilities, allowing users to access GCP services and third-party applications with a single login.
- **Multi-Factor Authentication (MFA):** Adds an extra layer of security by requiring a second authentication factor beyond passwords.
- **Device Management:** Cloud Identity allows organizations to manage mobile devices, enforcing security policies like encryption and screen locks.
- **Directory Services:** It can serve as a cloud-based directory for managing users and groups, syncing with on-premise systems like Active Directory or LDAP.
- **App Management:** Supports controlling access to apps, both GCP apps and other business apps, enhancing security and user management.

Why to use Cloud Identity

- **Centralized Management:** It offers a single platform to manage users, groups, and devices across Google services (GCP, Google Workspace) and third-party applications.
- **Improved Security:** By enforcing MFA, controlling access, and monitoring user behavior, Cloud Identity helps protect sensitive data and cloud infrastructure.
- **Access Control:** You can define and enforce granular access permissions through IAM, ensuring users have the least privilege needed to perform their tasks.
- **Seamless Integration:** Cloud Identity works smoothly with GCP and Google Workspace, simplifying access management for Google’s ecosystem of tools.
- **Cost Savings:** It eliminates the need for on-premise identity solutions by offering a scalable, cloud-based identity service.
- **Compliance:** Helps meet regulatory requirements for identity and access control in cloud environments.

Google Cloud Directory Sync



AD Integration to Google

Pre-requisites:

- Must have a Public DNS Domain
- You should be subscribed for Cloud Identity or G Suite Organization Account
- You should have access to the On-Premises Active Directory Machine
- You should be a Global Administrator or Super Admin in Cloud Identity or G Suite

GCP Networking (VPC)

GCP Networking (VPC)

- VPC stands for Virtual Private Cloud. Provision a **private, isolated virtual network** on the GCP cloud.
- Google VPC is the networking layer for GCP compute services, and it allows you to build your own virtual network within GCP.
- Within a region, you can create multiple GCP VPCs, and each GCP VPC is logically isolated.
- The power of GCP platform even allow you to create a “GLOBAL VPC” which spans across multiple Regions.
- When you create an Google VPC, you must specify the IPv4 address range by choosing a *Classless Inter-Domain Routing (CIDR)* block, such as 10.0.0.0/16.
- Google VPC address range may be as large as /8 (16,777,216 available addresses) or as small as /28 (16 available addresses) and should not overlap any other network with which they are to be connected.
- Have complete control over your virtual networking environment.

GCP Networking (VPC)

- VPC network is just like your physical network except that it is virtualised within google cloud.
- A VPC network is a global resource, this consists of a list of regional virtual subnetworks in Data centres, all connected by a global wide area network.
- **A VPC network provides: -**
- Provides connectivity for your VM, including GKE & app engine instances
- Offers native internal TCP/UDP load balancing
- Connects to on-premises network using Cloud VPN Tunnels and Cloud interconnect attachments
- Distributes traffic from Google cloud external Load balancers to backends
- VPC always lies inside a project. One project can contain multiple VPC networks.
- New project starts with a default VPC that has one subnet in each region.

GCP Networking (VPC)

- Go to GCP console and select VPC Network from Network tab.
- A Google VPC consists of following components:

- VPC networks – which can be Auto or Custom

Auto: Which google sets for your by default having subnets in all regions

Custom: Which you create as per your requirements

- VPC consists of Subnets (which can not span across regions)
- Route Tables (To allow packets to go from source to destination)
- Internet Gateway (To allow your resources to reach to Internet)
- External IP's (These are static IP's to be used for applications)
- Firewall Rules (Allow you to allow or deny traffic via specific ports)
- VPC network Peering (This allows two or more GCP VPC to talk to each other via internal IP's)
- Shared VPC (This is a Global VPC which can be setup between multiple regions)

GCP Networking (VPC)

- A **subnet** defines a range of IP addresses in your VPC. A *subnet* is a segment of a GCP VPC's IP address range where you can launch GCP instances.
- Each subnet must reside entirely within a Region and cannot span Regions. You can add one or more subnets in each Region.
- A **private subnet** should be used for resources that won't be accessible over the Internet.
- A **public subnet** should be used for resources that will be accessed over the Internet.
- PS: Generally it's a good practice to have Public and Private Subnets but in GCP this is not something which you get by default.

GCP Networking (VPC)

- **Live Demonstration with VPC!**

- Create one VPC name “gaganvpc”
- Create two subnets “gagansubnet1 and gagansubnet2”

- Case1: Create one server in “gagansubnet1” and try to ssh into the other server

- Result is “unsuccessful”

Routing Tables

- A *route table* is a logical construct within an GCP VPC that contains a set of rules (called routes) that are applied to the subnet and used to determine where network traffic is directed.
- Each route table contains a default route called the default-route, which enables communication within the Google VPC, and this route cannot be modified or removed. Additional routes can be added to direct traffic to exit the GCP VPC via the IGW's etc.
- Your VPC has an implicit router
- Your VPC automatically comes with a main route table that you can modify.
- You can create additional custom route tables for your VPC.

Internet Gateway

- An *Internet Gateway (IGW)* is a horizontally scaled, redundant, and highly available GCP VPC component that allows communication between instances in your GCP VPC and the Internet.

- GCP instances within an GCP VPC are only aware of their private IP addresses. When traffic is sent from the instance to the Internet, the IGW translates the reply address to the instance's public IP address (or External IP address) and maintains the one-to-one map of the instance private IP address and public IP address.

- When an instance receives traffic from the Internet, the IGW translates the destination address (public IP address) to the instance's private IP address and forwards the traffic to the GCP VPC.

EIP's

- GCP maintains a pool of public IP addresses in each region and makes them available for you to associate to resources within your GCP VPCs. An *External IP Addresses* is a static, public IP address in the pool for the region that you can allocate to your account (pull from the pool) and release (return to the pool).
- External IPs are specific to a region or global (in case of load balancers).
- There is a one-to-one relationship between network interfaces and External IPs.
- You can move External IPs from one instance to another, either in the same GCP VPC or a different GCP VPC within the same region.
- External IPs remain associated with your GCP account until you explicitly release them.
- There are charges for External IPs allocated to your account, when they are not associated with a resource.

Firewall Rules

- A *"Firewall rule" in GCP* is a virtual stateful firewall that controls inbound and outbound network traffic to GCP resources and instances.
- All GCP instances must be launched into a Firewall Rules. If a Firewall Rules are not specified at launch, then the instance will be launched into the default rules generated during the default VPC creation time.
- You may change the rules for the default Firewall rules.
- Below are the default rules which comes with default GCP network when you create your GCP account.

default-allow-icmp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	icmp	Allow	65534	default
default-allow-internal	Ingress	Apply to all	IP ranges: 10.128.0.0/9	tcp:0-65535 udp:0-65535 icmp	Allow	65534	default
default-allow-rdp	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:3389	Allow	65534	default
default-allow-ssh	Ingress	Apply to all	IP ranges: 0.0.0.0/0	tcp:22	Allow	65534	default

Firewalls in GCP

Live Demonstration with VPC!

Create one VPC name “gaganvpc”
Create two subnets “gagansubnet1 and gagansubnet2”

Case1: Create one server in “gagansubnet1” and try to ssh into the server

Result is “unsuccessful”

Solution is “Create a firewall rule to allow tcp port 22” and then perform the same test again.

This time the result is a “Success”

Firewalls in GCP

Live Demonstration with VPC!

Create one VPC name “gaganvpc”
Create two subnets “gagansubnet1 and gagansubnet2”

Case1: Create two servers on gagansubnet1 and gagansubnet2 (one server each subnet) and perform ping test

Not able to ping, no worries, create a firewall rule to allow “icmp protocol” and perform the test again

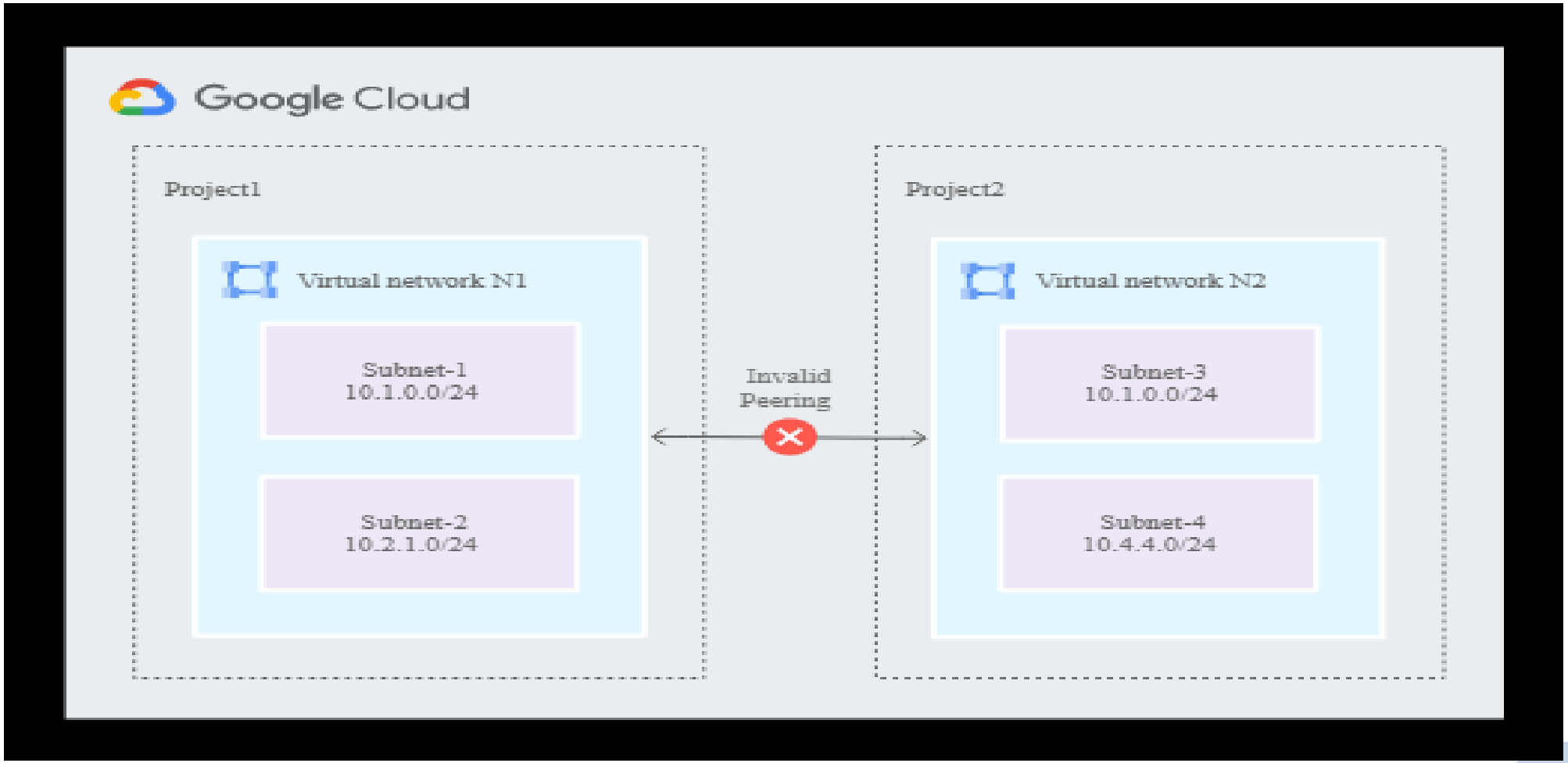
Important Facts

- You can have 5 Networks quota by default present in GCP. You can get this increased any time by raising a request.
- You can have a limit of 7000 IP per network.
- IP's both Public and Private one can be "Persistent or Ephemeral"
- Yes you read it right, even Private IP's in GCP can be Ephemeral.
- In a single VPC, zones across multiple regions can talk to each other via Private IP
- But when you want two VPC's to talk to each other or External on-prem network to talk to the GCP VPC you need to configure connectivity via "VPN, Direct interconnects, Cloud routers or VPC peering"

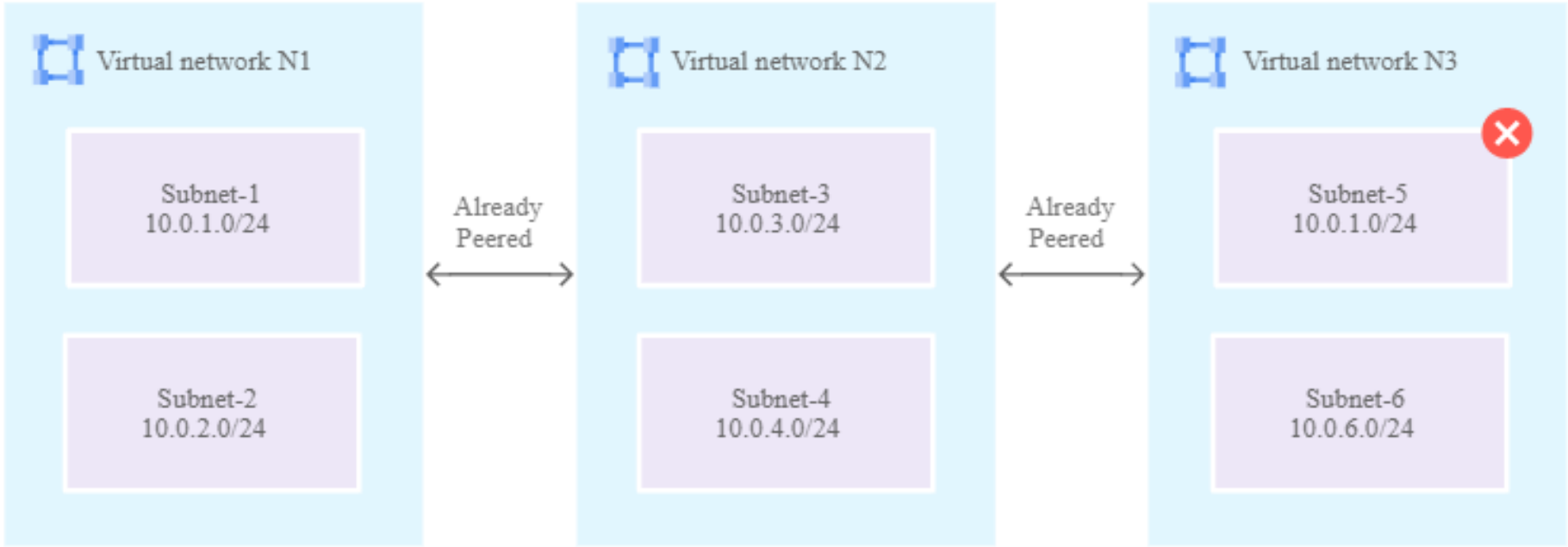
VPC Peering

- Google VPC *peering* connection is a networking connection between two GCP VPCs that enables instances in either GCP VPC to communicate with each other as if they are within the same network.
- Peering connections are created through a request/accept protocol.
- The connection should be setup from both project in order to work correctly.

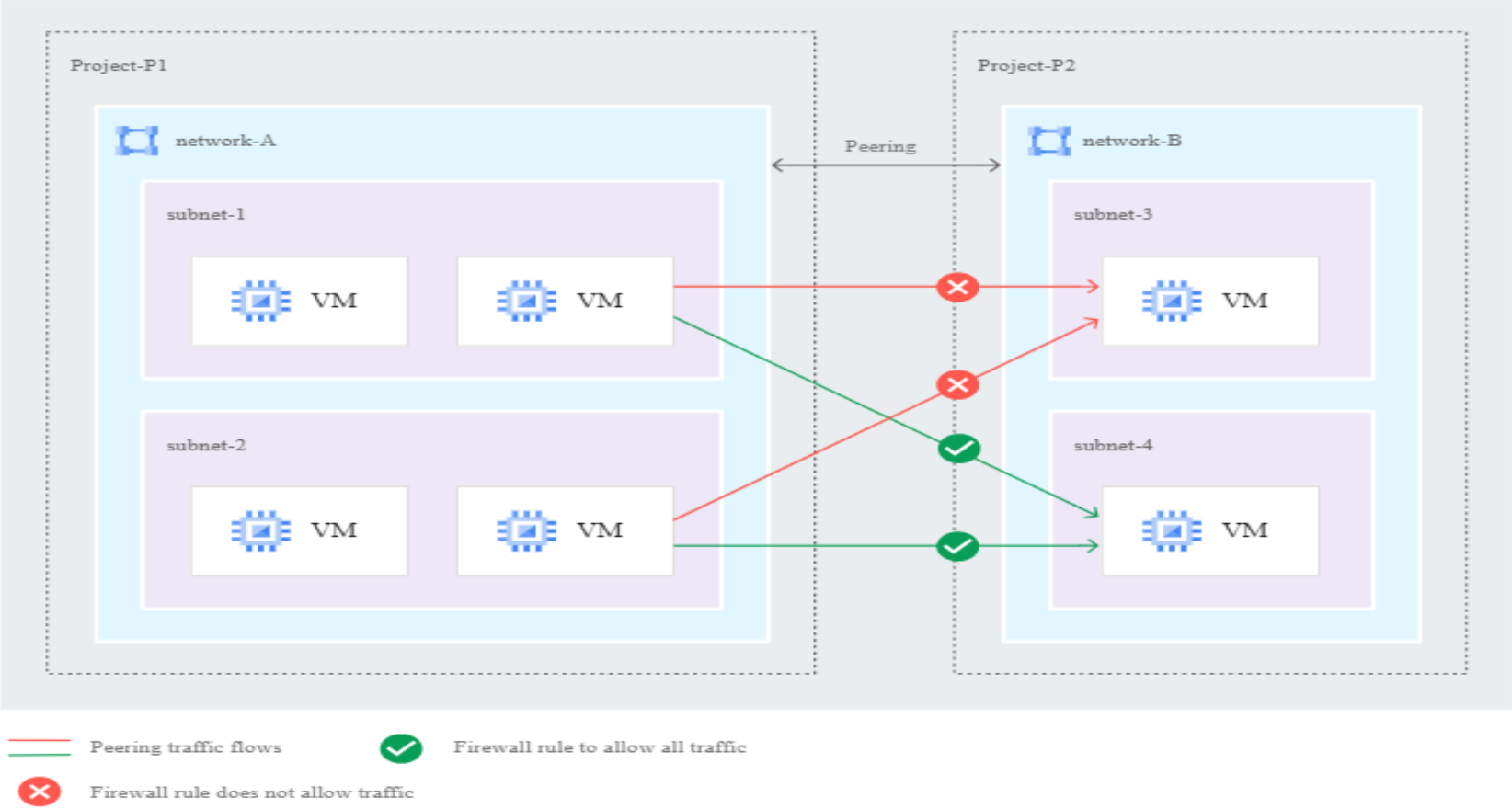
VPC Peering


Google Cloud
Project1
Virtual network N1
Subnet-1
10.1.0.0/24
Subnet-2
10.2.1.0/24
Invalid
Peering
Project2
Virtual network N2
Subnet-3
10.1.0.0/24
Subnet-4
10.4.4.0/24

VPC Peering


Virtual network N1
Subnet-1
10.0.1.0/24
Subnet-2
10.0.2.0/24
Already
Peered
Virtual network N2
Subnet-3
10.0.3.0/24
Subnet-4
10.0.4.0/24
Already
Peered
Virtual network N3
Subnet-5
10.0.1.0/24
Subnet-6
10.0.6.0/24

VPC Peering


Project-P1
Project-P2
network-A
Peering
network-B
subnet-1
subnet-3
VM
VM
VM
subnet-2
subnet-4
VM
VM
VM
Peering traffic flows
Firewall rule to allow all traffic
Firewall rule does not allow traffic

Regional VPC

case1: Regional VPC

Create one VPC with one subnet in us-central region
Create one more VPC with one subnet in asia-east region

Check the route table: you will see two routes for each VPC you created.
- One for internet gateway and other for internal IP communication (i,.e for each subnet)

- Note you cannot delete these default routes (internet gateway can be deleted to avoid internet reach)

Now let us create a firewall rule for these network to allow all traffic in this case

NOTE: To be very specific you can allow traffic on port 22 and for icmp protocol

Now go to compute and create two machine (each in other network)

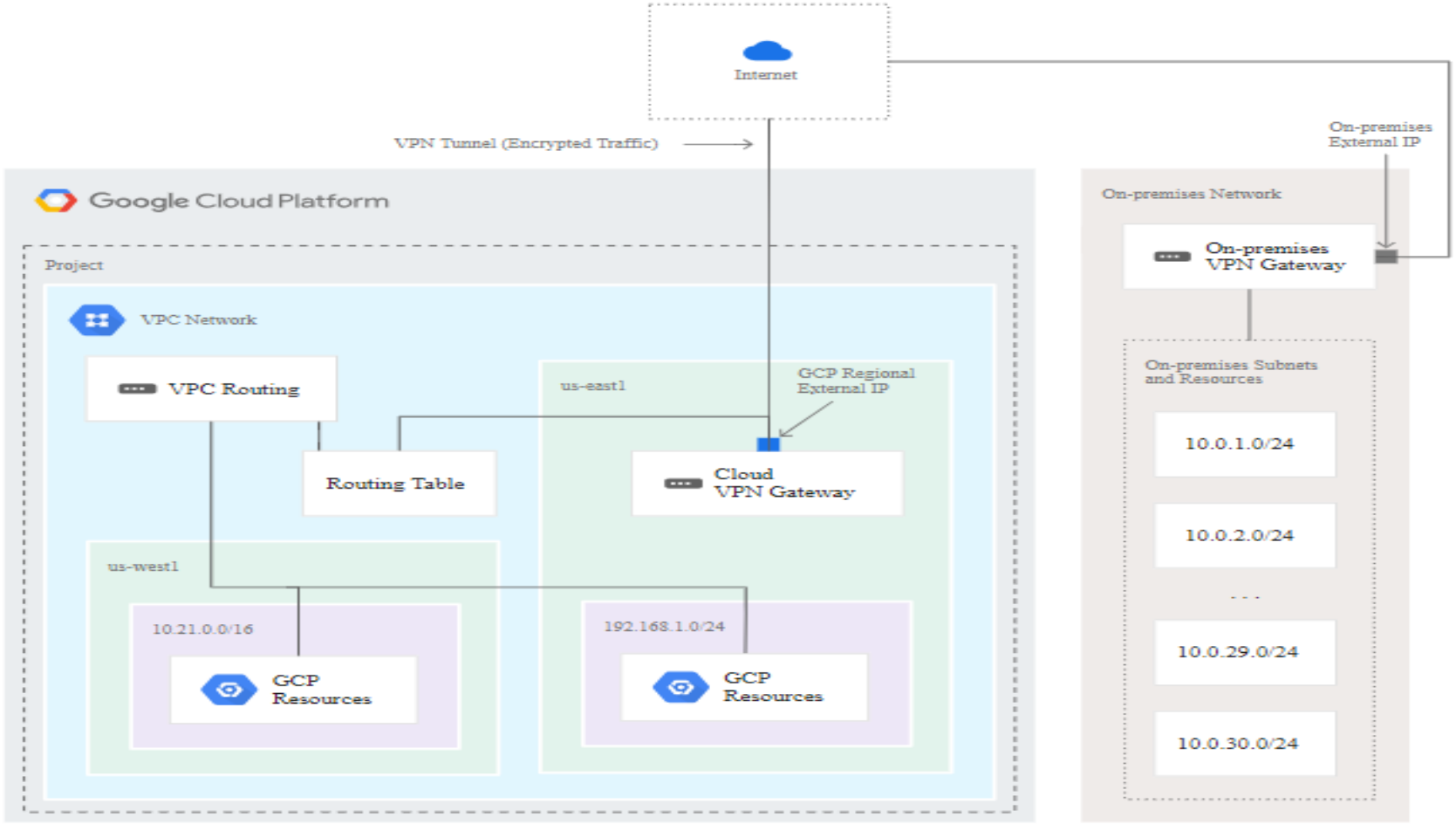
Perform private IP ping test and you will see its fail.

As these VPC's are different so it does not allow you to communicate directly over the Private IP.

VPN's, Direct Interconnect and External Peering

- A virtual private network lets you **securely connect** your Google Compute Engine resources to your own private network. Google VPN uses IKEv1 or IKEv2 to establish the IPSec connectivity.
- You can connect an existing data center to Google VPC using either hardware or software VPN connections, which will make Google VPC an extension of the data center.
- GCP VPC offers two ways to connect a corporate network to a VPC: **VPN** and **Direct Interconnects**.
- A *virtual private gateway (VPG)* is the *virtual private network (VPN)* concentrator on the GCP side of the VPN connection between the two networks.
- A *customer gateway (CGW)* represents a physical device or a software application on the customer's side of the VPN connection.
- After these two elements of an GCP VPC have been created, the last step is to create a VPN tunnel. The VPN tunnel is established after traffic is generated from the customer's side of the VPN connection.

VPN's, Direct Interconnect and External Peering



Direct Interconnect

- Interconnect lets you establish high bandwidth, low latency connections between your Google Cloud VPC networks and your on-premises infrastructure.
- Come in two forms: Direct from Google or From Partners network
- Choose a 10 Gbps pipe directly to a Google location with Dedicated Interconnect, or flexible bandwidth options (50 Mbps - 10 Gbps) with Partner Interconnect.

External Peering

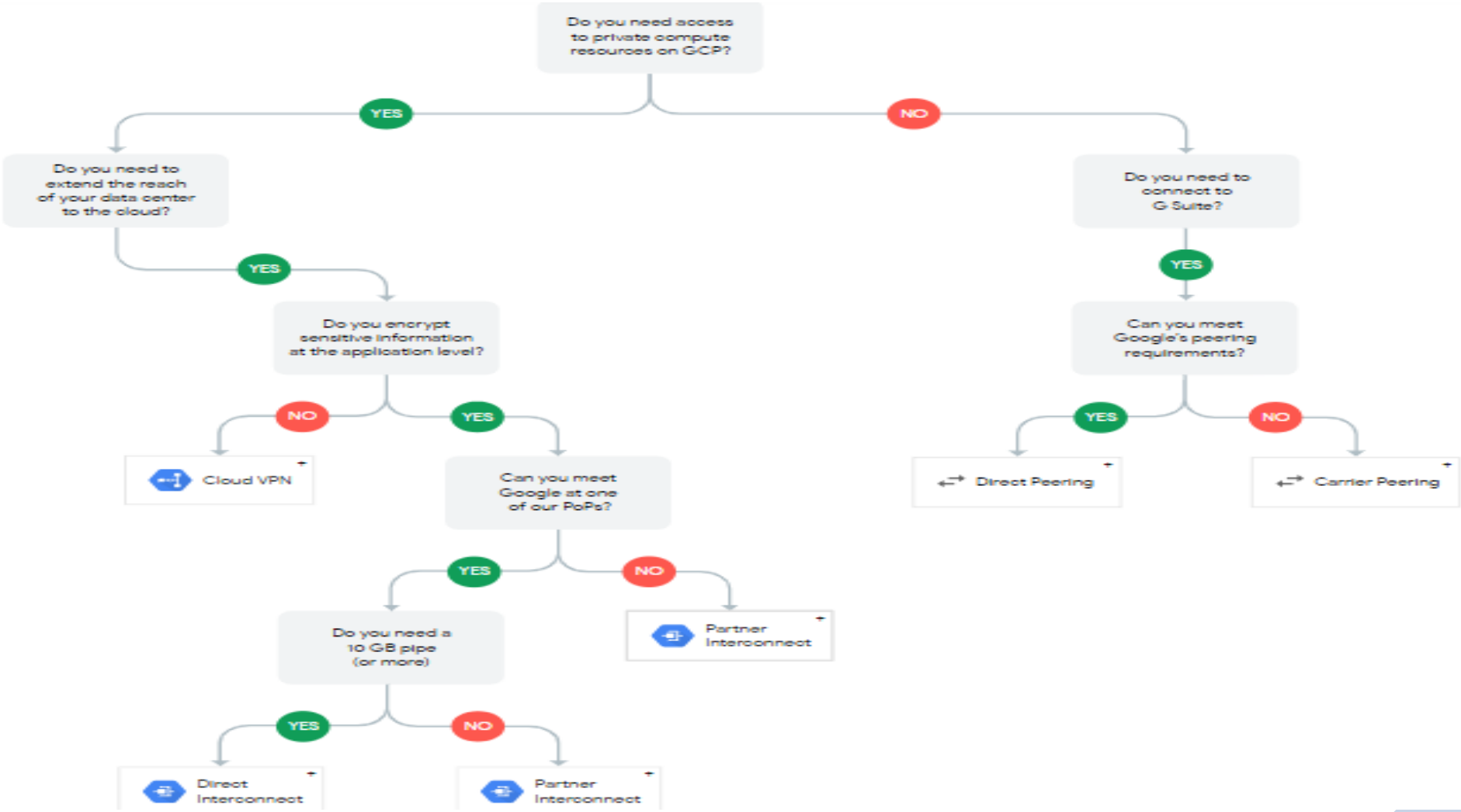
- External Peering comes in two forms:

- Direct
- Via Carrier Provider

- **Direct Peering**: Connect your business network directly to Google at any of over 100 locations in 33 countries around the world and exchange high throughput traffic.

- **Carrier Peering**: For access to Google applications, such as G Suite, you can obtain enterprise-grade network services that connect your infrastructure to Google by using a service provider.

Your choice for connection



Lab VPC

Create a Custom GCP VPC

1. Sign in to the GCP Management Console.

2. Select the GCP VPC Network tab and launch the GCP VPC Dashboard.

3. Create an GCP VPC with two subnets with a CIDR block equal to 192.168.0.0/24 and 192.168.1.0/24.

You have created your first custom VPC.

Lab VPC

- Create a firewall rule to allow traffic on port 22, 3389, icmp and port 80
- Create two instances on newly created subnets (one server each subnet)
- Test they are able to talk to each other via ping test

Lab VPC

- Perform complete clean up of your network VPC

- Delete VM’s
- Delete Subnet and VPC’s

Shared VPC

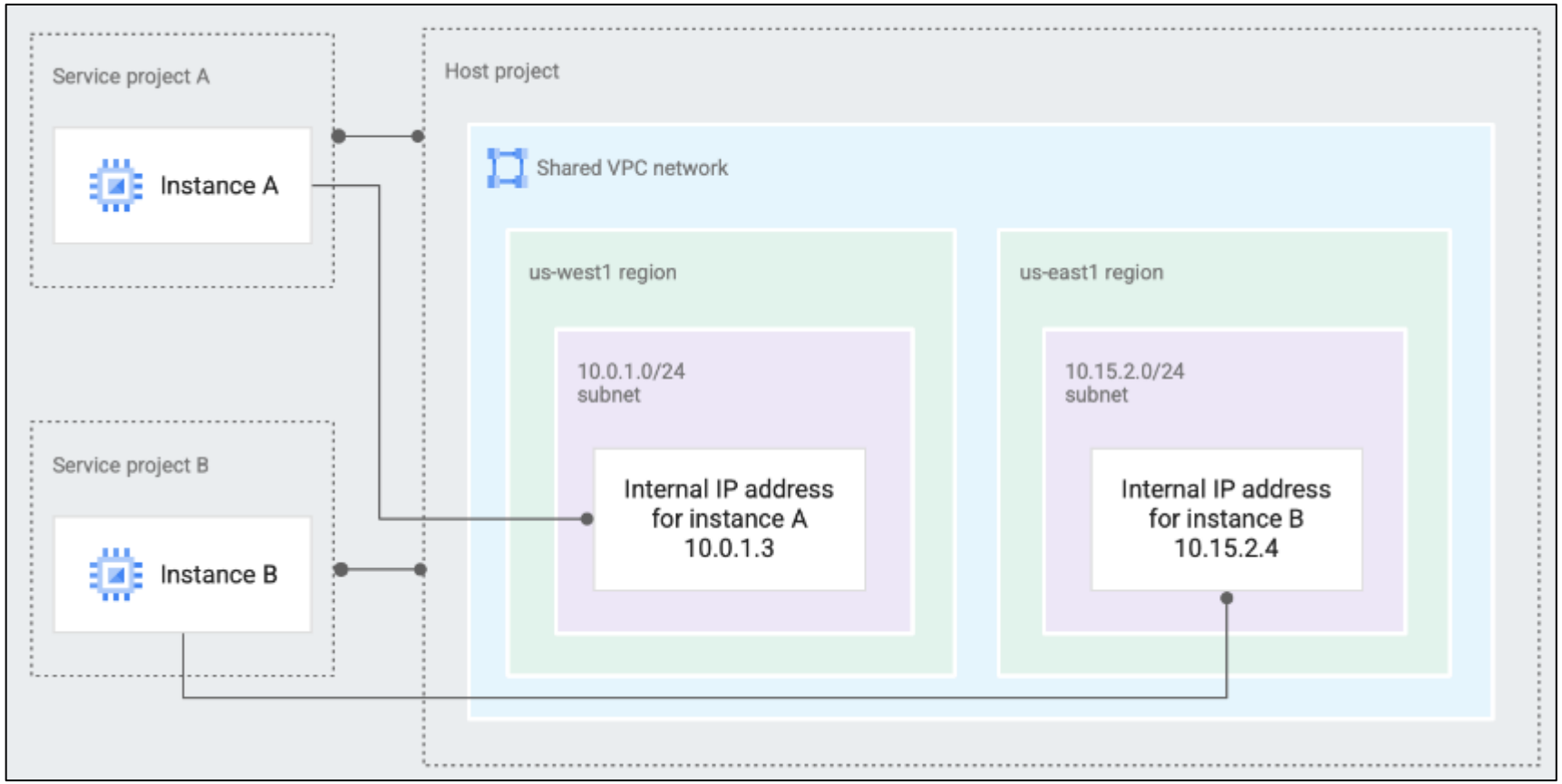
When you use Shared VPC, you designate a project as a host project and attach one or more other service projects to it. The VPC networks in the host project are called Shared VPC networks. Eligible resources from service projects can use subnets in the Shared VPC network.

A Shared VPC network is a VPC network defined in a host project and made available as a centrally shared network for eligible resources in service projects. Shared VPC networks can be either auto or custom mode, but legacy networks are not supported.

When a host project is enabled, you have two options for sharing networks:

- You can share all host project subnets. If you select this option, then any new subnets created in the host project, including subnets in new networks, will also be shared.
- You can specify individual subnets to share. If you share subnets individually, then only those subnets are shared unless you manually change the list.

Shared VPC


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

Private Google Access

- Use to connect Google APIs in Google Services network
- VM instances that only have internal IP addresses (no external IP addresses) can reach the external IP addresses of Google API and services
- Private Google Access has no effect on instances that have IP addresses
- For consumer resources in GCP, private Google Access is enabled on a subnet. For OnPrem, it is configuration DNS, firewall rules and routes

Private Google Access

Lab:

- Create an instance without external IP
- Try to connect to buckets
- Enable Private Google Access on the Subnet
  - Go to the VPC Network Configuration:
  - VPC Network > VPC networks.
  - Edit the Subnet:
  - Under the Subnets tab, find and select **private-subnet**.
  - Click Edit.
  - Scroll down to the **Private Google Access** section and turn it **On**.
  - Click Save.
- Now try to connect with GCS buckets

Cloud DNS

Cloud DNS

- DNS is a hierarchical distributed database that lets you store IP addresses and other data and look them up by name.
- Translates requests for domain names into IP addresses
- Google Cloud DNS lets you publish your zones and records in the Global DNS without the burden of managing your own DNS servers and software

Cloud DNS

Features:

- Fast Anycast Name Servers
- Scalability and Availability
- Zone and Project Management
- Manage through API and Web UI
- Private Zones
- DNS Forwarding
- Stackdriver Logging
- DNS Peering

Cloud DNS

Managed Zones:

- Managed zones hold DNS records for the same DNS suffix
- A project can have multiple managed zones, but they must have a unique name
- The managed zone is the resource that models a DNS zone and the records are hosted on the same Google operated name servers
- These name servers respond to DNS queries against the managed zone depending on the configuration.

Cloud DNS

Public Zones:

- A public zone is visible to the internet
- Cloud DNS has public authoritative name servers that respond to queries about public zones regardless of where the queries originate.
- You can create DNS records in a public zone to publish your service on the internet.
- Cloud DNS assigns a set of name servers when a public zone is created.
- The name servers have to be specified in your domain registrars settings.

Cloud DNS

Private Zones:

- Private zones let you manage custom domain names, load balancers and other GCP resources without exposing them to the internet.
- A private zone is a container of DNS records that can only be queries by authorized VPCs
- Resources and the VPC network need to be in the same project as the private zone, unless using DNS peering.
- Private zones do not support DNS security extensions (DNSSEC)
- Requests for DNS records in private zones must be submitted through the metadata server- 169.254.169.254

Cloud DNS

Forwarding and Peering zones

- A forwarding zone is a type of Cloud DNS managed private zone that sends requests for that zone to the IP addresses of its forwarding targets.
- There are no records defined here and the targets are DNS servers
- DNS peering allows you to send requests for records tht come from one zone's namespace to another VPC network.
- The VPC network where the DNS peering zone performs lookups is called the DNS producer network
- The VPC network authorized to use the peering zone is called the DNS consumer network
- The updates you make on manages zones get logged to an operations viewed from the command line/API

Cloud DNS

Internal DNS

- Virtual Private Cloud networks on GCP have an internal DNS service that allows instances in the same network to access each other by using internal DNS names
- The internal DNS name of a VM instance only resolves to its private internal IP address
- Only resources in the same project/VPC can make use of this
- Internal DNS names are automatically created by GCP and different from Cloud DNS records.

Cloud DNS

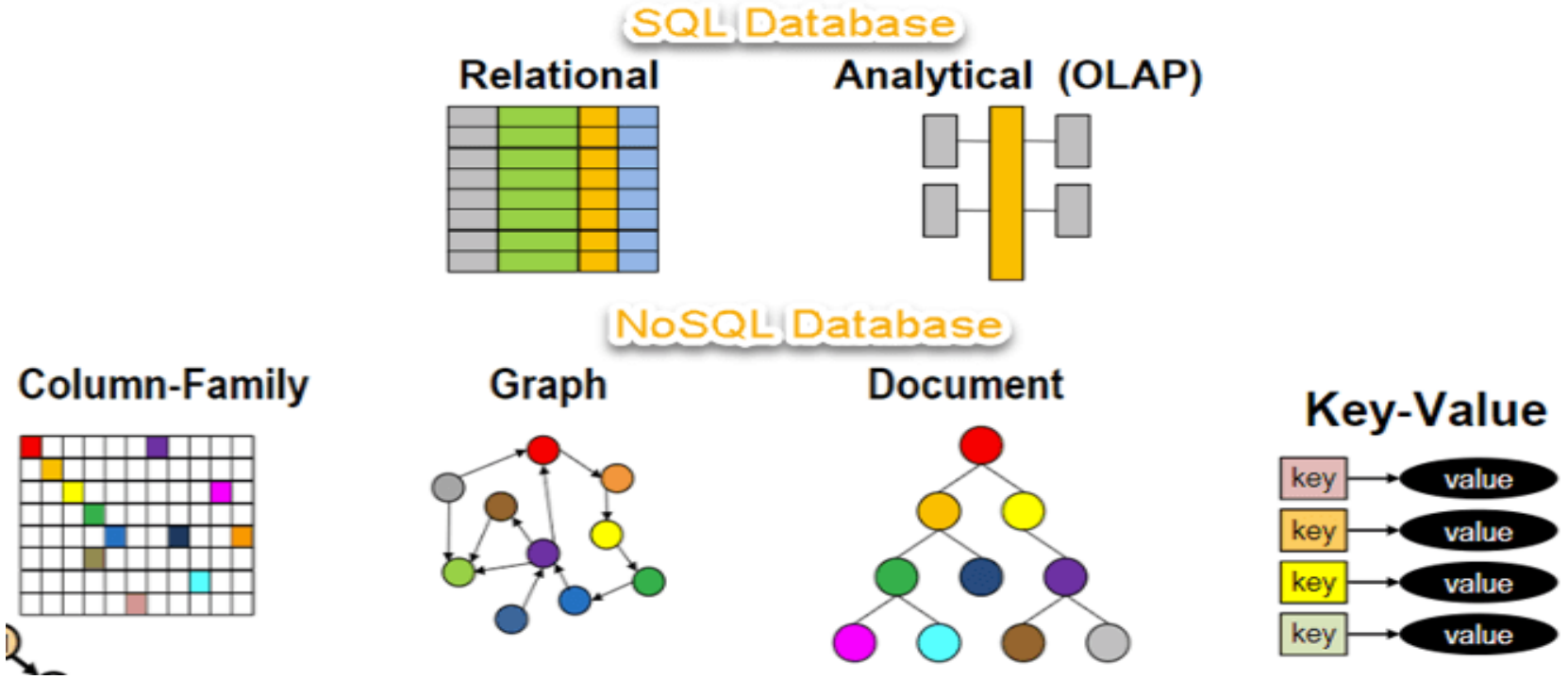
DNSSEC

- DNSSEC authenticates responses to domain name lookups
- Helps prevent attackers from manipulating, spoofing or poisoning the responses to DNS requests
- In order to use this, DNSSES needs to be enabled for the specific zone
- The domain registry for the TLD must have a DS record
- The clients also need to have the capability to validate the DNSSEC signatures
- This can be done by public services like Google Public DNS and Verizon Public DNS

Database

Database Types

In the world of database technology, there are two main types of databases: **SQL** and **NoSQL**— or, **relational** databases and **non-relational** databases. The difference speaks to how they're built, the type of information they store, and how they store it.



Relational DBs

- Relational Databases are the oldest and widely use databases
- A Relational database consists of one or more **tables**, and a table consists of **columns** and **rows** similar to a spreadsheet. A database column contains a specific **attribute** of the record, such as a person's name, address, and telephone number. Each attribute is assigned a **data type** such as text, number, or date, and the database engine will reject invalid inputs.
- A Relational database can be categorized as either an Online Transaction Processing (OLTP)or Online Analytical Processing (OLAP) database system.
- Support ACID (Atomicity, Consistency, Isolation, Durability) transactions
- **Example:** SQL database, Microsoft SQL Server, Oracle Database, IBM DB2, MY SQL, Maria DB, Sybase, PostgreSQL

Non-Relational DBs

- Recently came in limelight due to its unmatched benefits
- Good to use, If your data is unstructured and unpredicted. For example, requirements aren't clear at the outset or you don't have a clearly defined schema .
- Overcome the limitations of SQL databases e.g. horizontal scaling, parallel query performance, replication etc.
- Multiple benefits over SQL, including High concurrency, high volume random reads and writes, massive DB sizes, supporting unstructured data, HA at low cost etc.
- Support eventual consistency instead of ACID transactions.
- **Example:** MongoDB, Cassandra, neo4j, Redis, Couchbase, Elasticsearch

SQL vs NoSQL

	SQL	NoSQL
Performance	Low	High
Transactions	Atomic	Eventual Consistency
Consistency	Good	Poor
Reliability	Good	Poor
Supported DB Size	Mid-to-large size	Supports huge Database
Scalability	Medium (expensive)	High
Supported Datatype	Structured	Both Structured and Unstructured
Scaling method	Scale Up	Scale Out

GCP Cloud SQL Instances

Cloud SQL

- Cloud SQL instances are fully managed, relational MySQL, PostgreSQL, and SQL Server databases.
- It's a PaaS Service. So, Google handles replication, patch management, and database management to ensure availability and performance
- Set flags (for auto_increment_increment, default_time_zone, long_query_time, max_connections, wait_timeout, general_log, slow_query_log etc.) to modify DB Engine behavior (as you'll not get OS login).
- SQL Database handles management of system tables and file groups automatically
- User only needs to manage logical aspects of the database, including logins, tuning indexes, and query optimization
- Three Database types available – MySQL, Postgress, SQL Server.

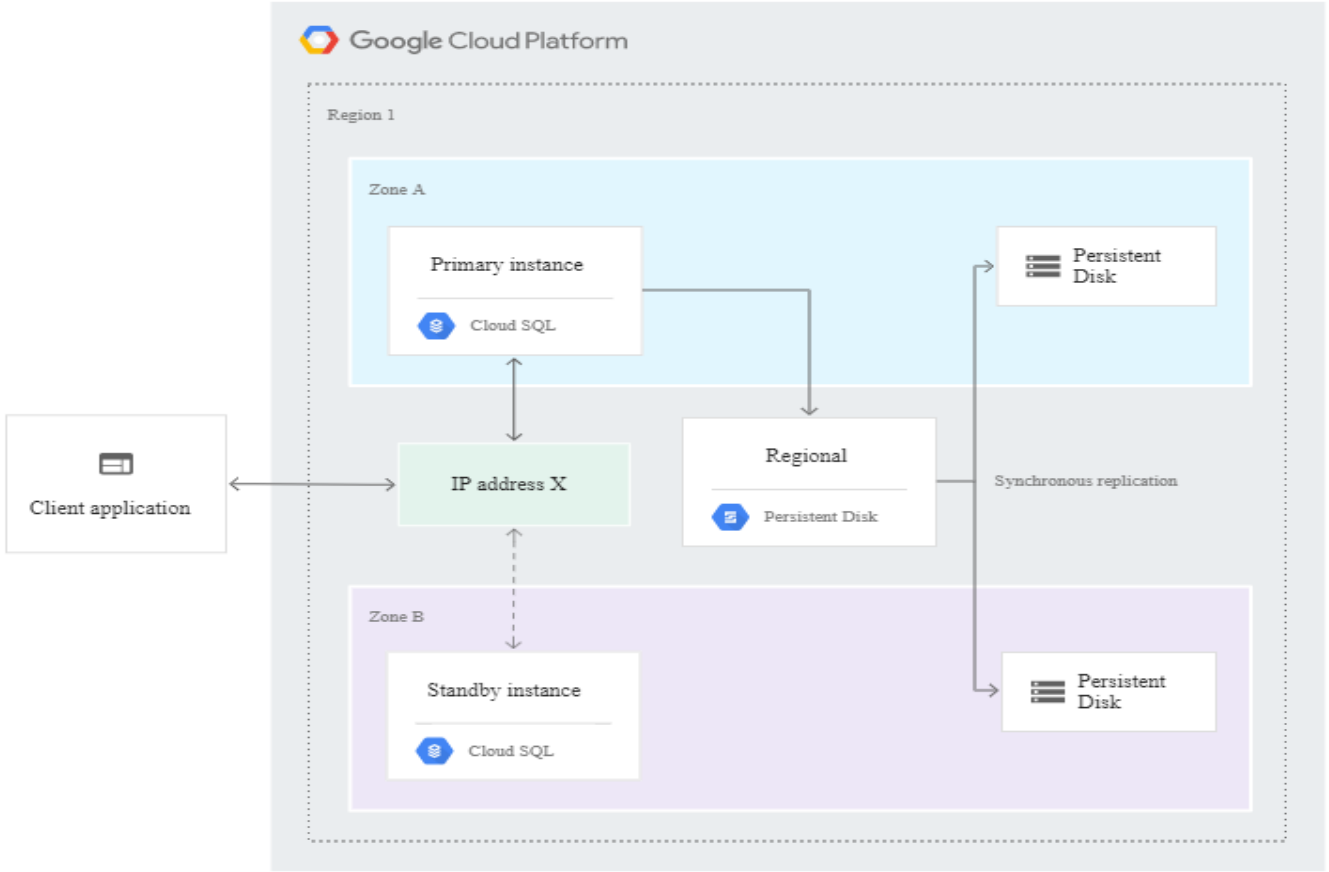
Cloud SQL - MySQL

- Cloud SQL supports MySQL 5.6 or 5.7, and provides up to 416 GB of RAM and 30 TB of data storage, with the option to **automatically increase the storage size as needed**.
- Support for secure external connections with the Cloud SQL Proxy or with the SSL/TLS protocol.
- Option for HA with Data replication between multiple zones with automatic failover.
- Import and export databases using mysqldump, or import and export CSV files.
- Automated and on-demand backups, and point-in-time recovery.
- High availability across Zones (Synchronous Replication)
- Read replica in another regions (ASynchronous Replication), which can be promoted lateron.

MySQL HA

If an HA-configured instance becomes unresponsive, Cloud SQL automatically switches to serving data from the standby instance. This is called a failover. To see if failover has occurred, check your operation log's failover history. When failover occurs, any existing connections to the primary instance and read replicas are closed, and it will take approximately 2-3 minutes for connections to be reestablished

When maintenance occurs on an instance, it does not fail over to the standby instance. Maintenance updates are applied to the standby instance at the same time as the primary instance.



MySQL HA

Failover Process:

1) The primary instance or zone fails.

2) Each second, the primary instance writes to a system database as a heartbeat signal. If multiple heartbeats aren't detected, failover is initiated. This occurs if the primary instance is unresponsive for approximately 60 seconds or the zone containing the primary instance experiences an outage.

3) The standby instance now serves data upon reconnection.

4) Through a shared static IP address with the primary instance, the standby instance now serves data from the secondary zone.

Cloud SQL - MySQL

Unsupported features

- InnoDB memcached plugin
- Federated Engine
- Memory Storage Engine
- The SUPER privilege

Unsupported Statements

Sending any of the following types of SQL statements will generate an error with the message "Error 1290: The MySQL server is running with the google option so it cannot execute this statement":

- LOAD DATA INFILE
- Note that LOAD DATA LOCALINFILE is supported.
- SELECT ... INTO OUTFILE
- SELECT ... INTO DUMPFILE
- INSTALL PLUGIN ...
- UNINSTALL PLUGIN
- CREATE FUNCTION ... SONAME ...

Note: Because Cloud SQL is a managed service, it restricts access to certain system procedures and tables that require advanced privileges.

Cloud SQL - MySQL

Unsupported Statements

The following statements are not supported because MySQL instances use GTID replication:

CREATE TABLE ... SELECT statements
CREATE TEMPORARY TABLE statements inside transactions
Transactions or statements that update both transactional and nontransactional tables

Unsupported functions

LOAD_FILE()

- InnoDB is the only supported storage engine.

- Replication with GTIDs

Lab

Create a SQL Instance and work with gcloud CLI:

gcloud sql connect mysql1 --user=root

CREATE DATABASE guestbook;

USE guestbook;
CREATE TABLE entries (guestName VARCHAR(255), content VARCHAR(255),
entryID INT NOT NULL AUTO_INCREMENT, PRIMARY KEY(entryID));
INSERT INTO entries (guestName, content) values ("first guest", "I got here!");
INSERT INTO entries (guestName, content) values ("second guest", "Me too!");

SELECT * FROM entries;

Drop table entries;

GCP Cloud Spanner

Cloud Spanner

- Cloud Spanner is a fully managed, mission-critical, **Horizontal scaled** relational database service that offers transactional consistency at global scale, schemas, SQL (ANSI 2011 with extensions), and automatic, synchronous replication for high availability.

- Industry-leading high-availability with 99.99% availability SLA for Single Region and 99.999% availability SLA for Multi-Region

- Combine the benefits of relational database structure with non-relational horizontal scale.

- Google-grade security as defaults, with **encryption** by default in **transit** and at **rest**, **granular identity & access management,** comprehensive **audit logging**, **custom-manufactured hardware**, hardware tracking and disposal, and the **Google-owned and controlled global network**.

- Global fully managed replications.

Cloud Spanner

- Compatible with industry-standard ANSI 2011 SQL.
- Handles schema changes as an online operation.
- Multi-Language Support - Client libraries in C#, Go, Java, Node.js, PHP, Python, and Ruby. JDBC driver for connectivity with popular third-party tools.
- Transactional Consistency - Purpose-built for external, strong, global transactional consistency.
- Battle tested by Google's own mission-critical applications and services.
- Google owns the entire software-hardware stack for Spanner. All Spanner machines employ battery-backed RAM, so that writes can be synced to disk even in the (unlikely) event of datacenter power loss. This also allows the filesystem to acknowledge writes without waiting for the latency of writing to disk.

Cloud Spanner

	Cloud Spanner	Traditional Relational	Traditional Non-Relational
Schema	✓ **Yes**	✓ Yes	✗ No
SQL	✓ **Yes**	✓ Yes	✗ No
Consistency	✓ **Strong**	✓ Strong	✗ Eventual
Availability	✓ **High**	✗ Failover	✓ High
Scalability	✓ **Horizontal**	✗ Vertical	✓ Horizontal
Replication	✓ **Automatic**	↻ Configurable	↻ Configurable

Cloud Spanner

- It partitions database tables into contiguous key ranges called splits. A single machine can serve multiple splits, and there is a fast lookup service for determining the machine(s) that serve a given key range. The details of how data is split and what machine(s) it resides on are transparent to Spanner users. The result is a system that can provide low latencies for both reads and writes, even under heavy workloads, at very large scale.

- Consistent replication to the different copies of the split is managed by the **Paxos algorithm**. In Paxos, as long as a majority of the voting replicas for the split are up, one of those replicas can be elected leader to process writes and allow other replicas to serve reads.

- Spanner is able to provide strongly consistent snapshots across the entire database using a Google-developed technology called TrueTime(exact global time with a high degree of accuracy).

- File storage is decoupled from the machines that create, read, update, and delete them. This enables the creation of more robust systems; a single bad disk cannot result in the loss or corruption of data. The combination of Spanner's Paxos-based replication, and the robustness of the underlying distributed file system, provides extremely good data reliability.

Cloud Spanner

CREATE TABLE ExampleTable (Id INT64 NOT NULL, Value STRING(MAX),) PRIMARY KEY(Id);

Split	KeyRange
0	[-∞,3)
1	[3,224)
2	[224,712)
3	[712,717)
4	[717,1265)
5	[1265,1724)
6	[1724,1997)
7	[1997,2456)
8	[2456,∞)

5 Nodes replicated across 3 zones

Cloud Spanner

- Your choice of node count determines the amount of serving and storage resources that are available to the databases in that instance.

- In Single region, each node provides up to 2 TB of storage, 10,000 queries per second (QPS) of reads or 2,000 QPS of writes (writing single rows at 1 KB of data per row). The peak read and write throughput values that nodes can provide depend on the instance configuration, as well as on schema design and dataset characteristics.

- Multi-region performance:

Multi-Region Configuration	Approximate Peak Read (QPS per region)	Approximate Peak Writes (QPS total)
nam3	7,000	1,800
nam-eur-asia1	7,000	1,000

Lab

Create a Cloud Spanner Instance and Work with same:

CREATE TABLE entries (guestName STRING(1024), content STRING(1024), entryID INT64 NOT NULL) PRIMARY KEY(entryID)

INSERT INTO entries (guestName, content, entryID) values ("first guest", "I got here!”,12);

INSERT INTO entries (guestName, content, entryID) values ("secondt guest", "Me too", 2);

Select * from entries;

Drop table entries;

Cloud Spanner

- After you create an instance, you cannot change the configuration of that instance later.

- Cloud Spanner does not have a suspend mode. Cloud Spanner nodes are dedicated resources, and even when you are not running a workload, Cloud Spanner nodes frequently perform background work to optimize and protect your data.

- For optimal performance, follow these best practices:
  - Design a schema that prevents hotspots and other performance issues.
  - Place critical compute resources within the same region as your Cloud Spanner instance.
  - Provision enough Cloud Spanner nodes to keep high priority total CPU utilization under 65%.

Configuration	Availability	Latency	Cost	Data Locality
Regional	99.99%	Lower write latencies within region.	Lower cost; see pricing.	Enables geographic data governance.
Multi-region	99.999%	Lower read latencies from multiple geographic regions.	Higher cost; see pricing.	Distributes data across multiple regions within the configuration.

GCP Bigtable

Cloud Bigtable

- A petabyte-scale, fully managed NoSQL database (**based on Hbase**) service for large analytical and operational workloads.
- Fast and performant - large-scale, low-latency applications as well as throughput-intensive data processing and analytics.
- Provision and scale to hundreds of petabytes and smoothly handle millions of operations per second.
- Supports the open-source industry standard **HBase API** and can be integrated easily with popular big data tools like Hadoop, Cloud Dataflow, and Cloud Dataproc.
- Fully Managed – PaaS Service.

Cloud Bigtable

- **Incredible scalability**
- **Simple administration** – with transparent upgrades, restarts and replications.
- **Cluster resizing without downtime** - You can increase the size of a Cloud Bigtable cluster for a few hours to handle a large load, then reduce the cluster's size again—all without any downtime.

- With Single node you can achieve:
  Reads: up to 10,000 rows/s **or**
  Writes: up to 10,000 rows/s **or**
  Scans: up to 220 MB/s
  Storage: 2.5 TB

- Replication uses a primary-primary configuration with eventual consistency.

- Same database that powers many core Google services, including Search, Analytics, Maps, and Gmail.

Cloud Bigtable

- Cloud Bigtable is a sparsely populated table that can scale to billions of rows and thousands of columns, enabling you to store terabytes or even petabytes of data.

- A single value in each row is indexed; this value is known as the row key. Cloud Bigtable is ideal for storing very large amounts of single-keyed data with very low latency.

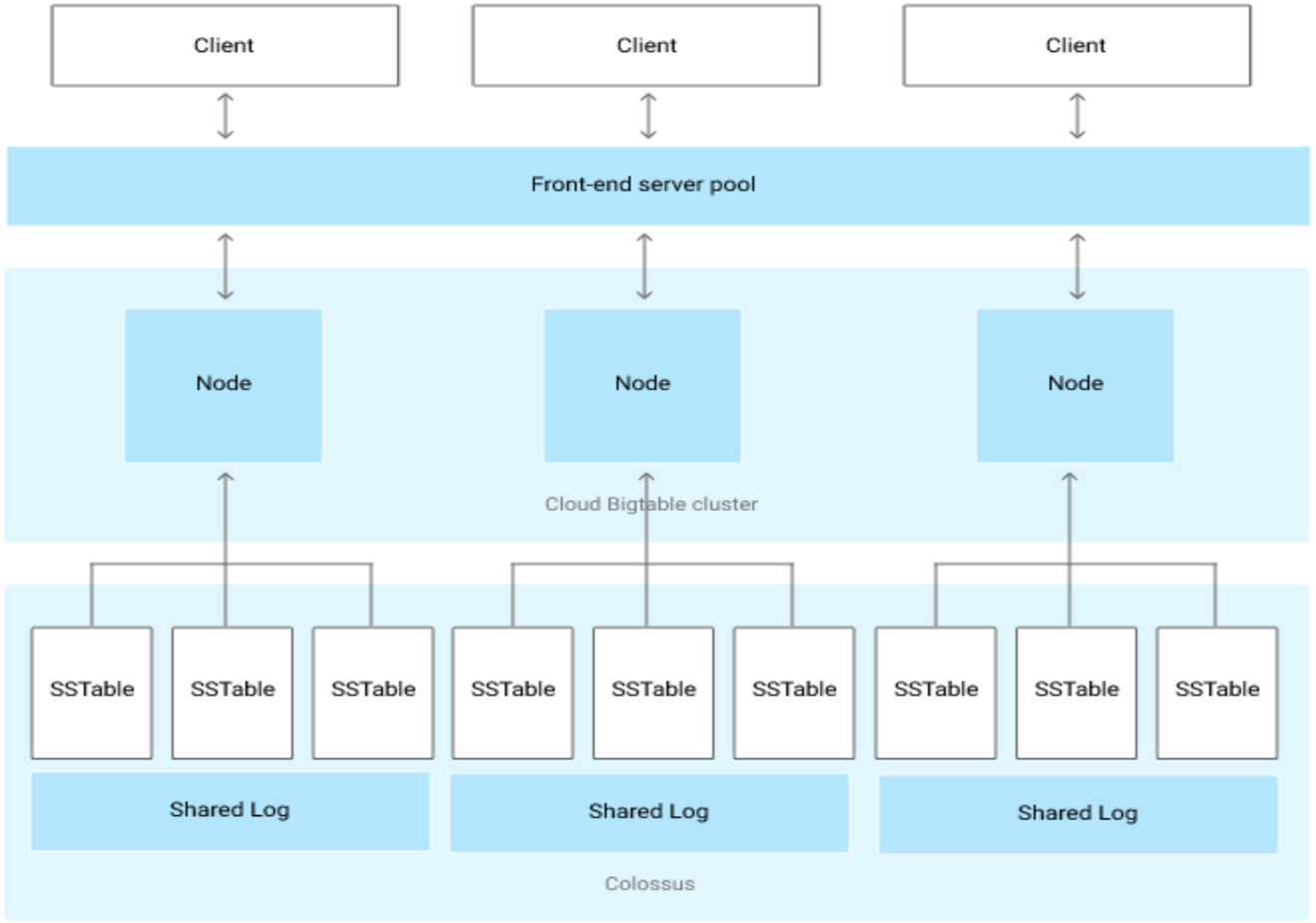
- Cloud Bigtable is ideal for applications that need very high throughput and scalability for **key/value** data, where each value is typically no larger than **10 MB.**

- To use Cloud Bigtable, you create **instances**, which contain up to 4 clusters that your applications can connect to. Each cluster contains **nodes**, the compute units that manage your data and perform maintenance tasks.

Cloud Bigtable

- A table belongs to an **instance**, not to a cluster or node. If you have an instance with more than one cluster, you are using replication. An instance can have up to 4 clusters.
- A **cluster** represents the Cloud Bigtable service in a specific location –Zone. Used for replication.
- Each cluster in an instance has 1 or more **nodes**, which are compute resources that Cloud Bigtable uses to manage your data.
- Cloud Bigtable splits all of the data in a table into separate **tablets**. Tablets are stored on disk, separate from the nodes but in the same zone as the nodes. A tablet is associated with a single node.
- Each node is responsible for:
  - Keeping track of specific tablets on disk.
  - Handling incoming reads and writes for its tablets.
  - Performing maintenance tasks on its tablets, such as periodic compactions.

Cloud Bigtable - Architecture



Cloud Bigtable Use case

You can use Cloud Bigtable to store and query all of the following types of data:

- Time-series data, such as CPU and memory usage over time for multiple servers.
- Marketing data, such as purchase histories and customer preferences.
- Financial data, such as transaction histories, stock prices, and currency exchange rates.
- Internet of Things data, such as usage reports from energy meters and home appliances.
- Graph data, such as information about how users are connected to one another.

Cloud Bigtable

- Create a Bigtable Cluster and perform exercise:

https://cloud.google.com/bigtable/docs/quickstart-cbt

DB on Cloud

Best Practices & Constraints:

Use DB Paas (DBaaS)

- When you want to save License & Support cost (OS + DB License).
- For new deployments and to deal with complex installation & management of DB HA.
- To avoid overhead of management issues alike, DB patching, upgrade, security fixes, failures, scaling, Backup, Mirroring, logs etc.
- When you want to go complaint instantly for your newly deployed DBs
- When your application is on cloud.
- You want to focus on your business and applications, and have CSP take care of DB management
- You want to pay for the DB license as part of the instance cost on an hourly basis instead of making a large upfront investment.
- You would rather focus on high-level tasks, such as performance tuning and schema optimization, rather than the daily administration of the database.
- You want to scale the instance type up or down based on your workload patterns without being concerned about licensing and the complexity involved.

DB on Cloud

Best Practices & Constraints:

Use DB on IaaS

- To utilize your existing licenses, as PaaS don’t always offers you BYOL for DB & OS.
- IaaS is the only choice for legacy and un-supported DB versions.
- You need full control over the database, including SYS/SYSTEM user access
- You need access at the operating system level
- Your database size exceeds the 80% of current maximum database size in Amazon RDS
- You need features or options that are not currently supported by Amazon RDS
- Your database IOPS needs are higher than the current IOPS limit
- When you need to run a small DB and APP together on single system

KMS

Encryption in flight

Data is encrypted before sending and decrypted after receiving

TLS certificates help with encryption (HTTPS)

Encryption in flight ensures no **man in the middle attack** can happen

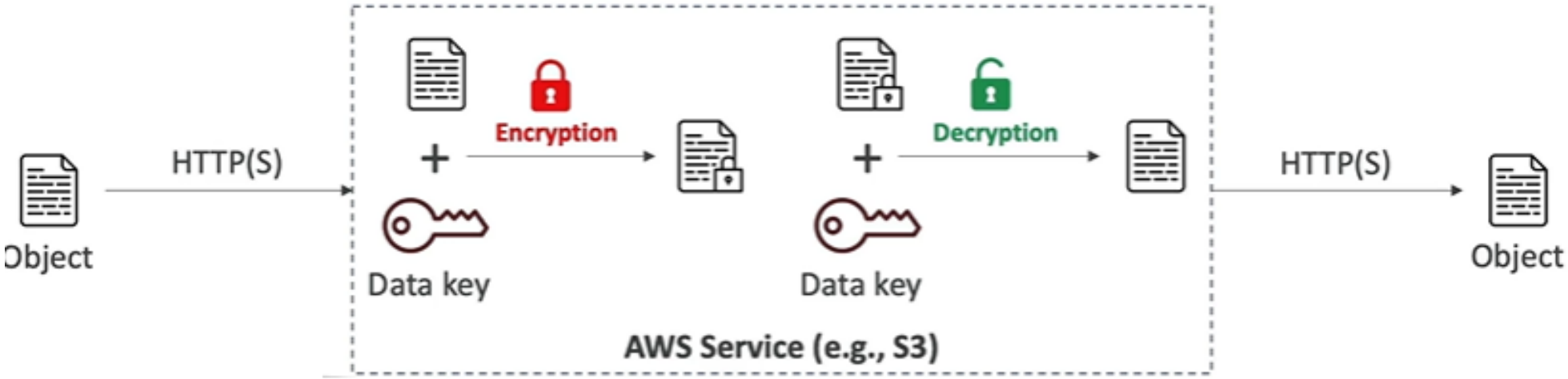
Server-side encryption at rest

Data is encrypted after being received by the server

Data is decrypted before being sent

It is stored in an encrypted form (using data keys)

The encryption/decryption keys must be managed somewhere and server must have access to it



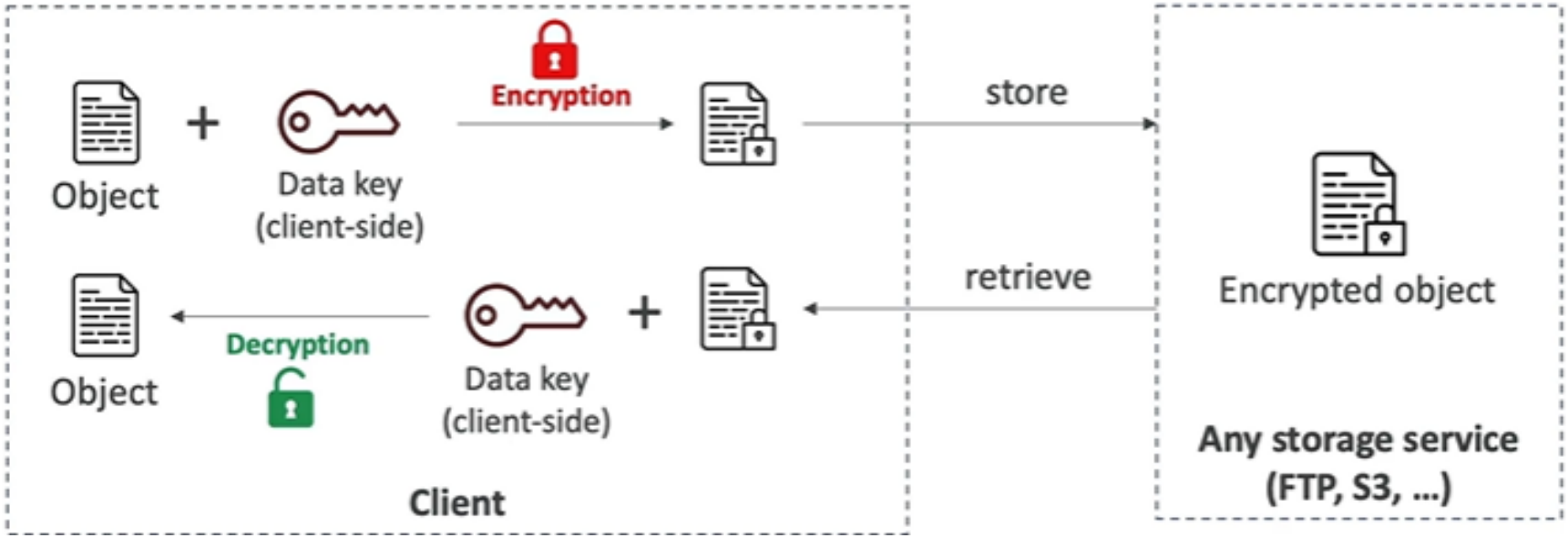
Client-side encryption

Data is encrypted by the client and never decrypted by the server

Data will be decrypted by a receiving client

The server should not be able to decrypt the data

Could leverage Envelope Encryption



KMS

Google Cloud Key Management Service (KMS) is a managed service that allows users to create, manage, and securely use cryptographic keys in Google Cloud.

It helps safeguard sensitive data and enables secure encryption and decryption operations for applications.

Key Features:

- **Create and Manage Keys**: Create, rotate, and destroy cryptographic keys used for data encryption.
- **Encryption and Decryption**: Perform encryption and decryption operations using the keys managed in KMS.
- **Key Rings**: Organize keys into key rings for easier management and access control.
- **Key Rotation**: Automatically rotate keys on a set schedule to enhance security.
- **Key Versions**: Manage multiple versions of a key, which helps during key rotation or key updates.

KMS

Key Features:

- **Audit Logging:** Track access to KMS keys via Google Cloud's Cloud Audit Logs, helping with compliance and security monitoring.
- **IAM Control:** Integrate with Google Cloud Identity and Access Management (IAM) to manage who can access and use keys.
- **Multiple Key Types:**
  - **Symmetric Keys**: Used for encrypting and decrypting data with a single key.
  - **Asymmetric Keys**: Supports signing, encryption, and decryption using public/private key pairs.
  - **Hardware Security Module (HSM):** Option to use Cloud HSM for higher levels of security, where keys are stored in a hardware security module.
  - **Customer-Managed Encryption Keys (CMEK):** Allow integration of KMS keys with Google Cloud storage services like Cloud Storage, BigQuery, Compute Engine, and more to encrypt data with customer-supplied keys.

KMS: Symmetric Key

Lab: Encrypt and Decrypt using the Symmetric Key

Description	Commands
Encrypt the file	gcloud kms encrypt \ --key key-enc \ --keyring demo-key \ --location us-central1 \ --plaintext-file data.txt \ --ciphertext-file data.txt.enc
Decrypt the file	gcloud kms decrypt \ --key key-enc \ --keyring demo-key \ --location us-central1 \ --ciphertext-file data.txt.enc \ --plaintext-file data1.txt

KMS: Asymmetric Key

Lab: Encrypt and Decrypt using the Asymmetric Key

Description	Commands
Encrypt the file using Public Key	openssl pkeyutl -in hello.txt \ -encrypt \ -pubin \ -inkey demo-key-key-dec-1.pub \ -pkeyopt rsa_padding_mode:oaep \ -pkeyopt rsa_oaep_md:sha256 \ -pkeyopt rsa_mgf1_md:sha256 \ > hello-enc
Decrypt the using Private Key	gcloud kms asymmetric-decrypt \ --version 1 \ --key key-dec \ --keyring demo-key \ --location us-central1 \ --ciphertext-file hello-enc \ --plaintext-file hello-dec

KMS: Asymmetric Key

Lab:

- Navigate to Cloud KMS:
  - In the Navigation Menu on the left, scroll down and select Security.
  - Under Security, click on Key Management (KMS).
- Create a Key Ring:
  - If you don't already have a key ring, click the Create Key Ring button.
  - In the dialog box, fill in the required information:
  - Name: Choose a name for the key ring (e.g., my-key-ring).
  - Location: Choose the region where you want to store your keys (e.g., us-central1).
  - Click Create.

KMS: Asymmetric Key

Lab:

- Create an Asymmetric Key:
  - After the key ring is created, select your key ring from the list.
  - Click Create Key to create a new key inside the key ring.
  - In the dialog box:
  - Name: Give the key a name (e.g., my-signing-key).
  - Purpose: Choose Asymmetric Sign/Verify. This ensures the key pair will be used for signing data.
  - Key Protection Level: You can select Software or HSM (Hardware Security Module) for added security if you require hardware-level protection.
  - Key Algorithm: Choose an algorithm for your signing key:
    - RSA Sign-PSS 2048 SHA256, RSA Sign-PSS 3072 SHA256, RSA Sign-PSS 4096 SHA256, RSA Sign-PKCS1 2048 SHA256, RSA Sign-PKCS1 3072 SHA256, RSA Sign-PKCS1 4096 SHA256, EC Sign P-256 SHA256 (Recommended for ECDSA signatures)
  - Click Create.
- **Permissions and IAM:**
  - Ensure that appropriate IAM roles are granted to users or service accounts that need access to the key. Roles like **Cloud KMS CryptoKey Encrypter/Decrypter** or **Cloud KMS CryptoKey Signer/Verifier** are typically assigned based on the action required.

KMS: Asymmetric Key

Lab: Sign Verify using the Asymmetric Key

Description	Commands
Signed the document using a key	gcloud kms asymmetric-sign \ --version 1 \ --key key1 \ --keyring demo-key \ --location us-central1 \ --digest-algorithm sha256 \ --input-file input.txt \ --signature-file input-signed
Check the key if tempered or not	openssl dgst \ -sha256 \ -verify demo-key-key1-1.pub \ -signature input-signed \ input.txt

Note: Download the public key (demo-key-key1-1.pub)

KMS: Mac Key

Lab: Sign Verify using the Mac Key

Description	Commands
To Create MAC Signature	gcloud kms mac-sign \ --version 1 \ --key key \ --keyring demo-key \ --location us-central1 \ --input-file mac-file \ --signature-file mac-signed
To Verify MAC Signature	gcloud kms mac-verify \ --version 1 \ --key key \ --keyring demo-key \ --location us-central1 \ --input-file mac-file \ --signature-file mac-signed

Data Loss Prevention

DLP

Data Loss Prevention (DLP) in Google Cloud Platform (GCP) is a fully managed service designed to help organizations detect, protect, and manage sensitive data. GCP's Cloud DLP allows you to identify and redact sensitive information like personally identifiable information (PII), payment card information (PCI), and other types of confidential data from your datasets.

Common Use Cases for GCP DLP:

Data Compliance:

- Cloud DLP helps organizations comply with data privacy regulations like GDPR, HIPAA, and PCI-DSS by identifying and protecting sensitive information in their datasets.

Data Redaction and Masking:

- Protecting sensitive information by masking or redacting it before storing, sharing, or analyzing data.

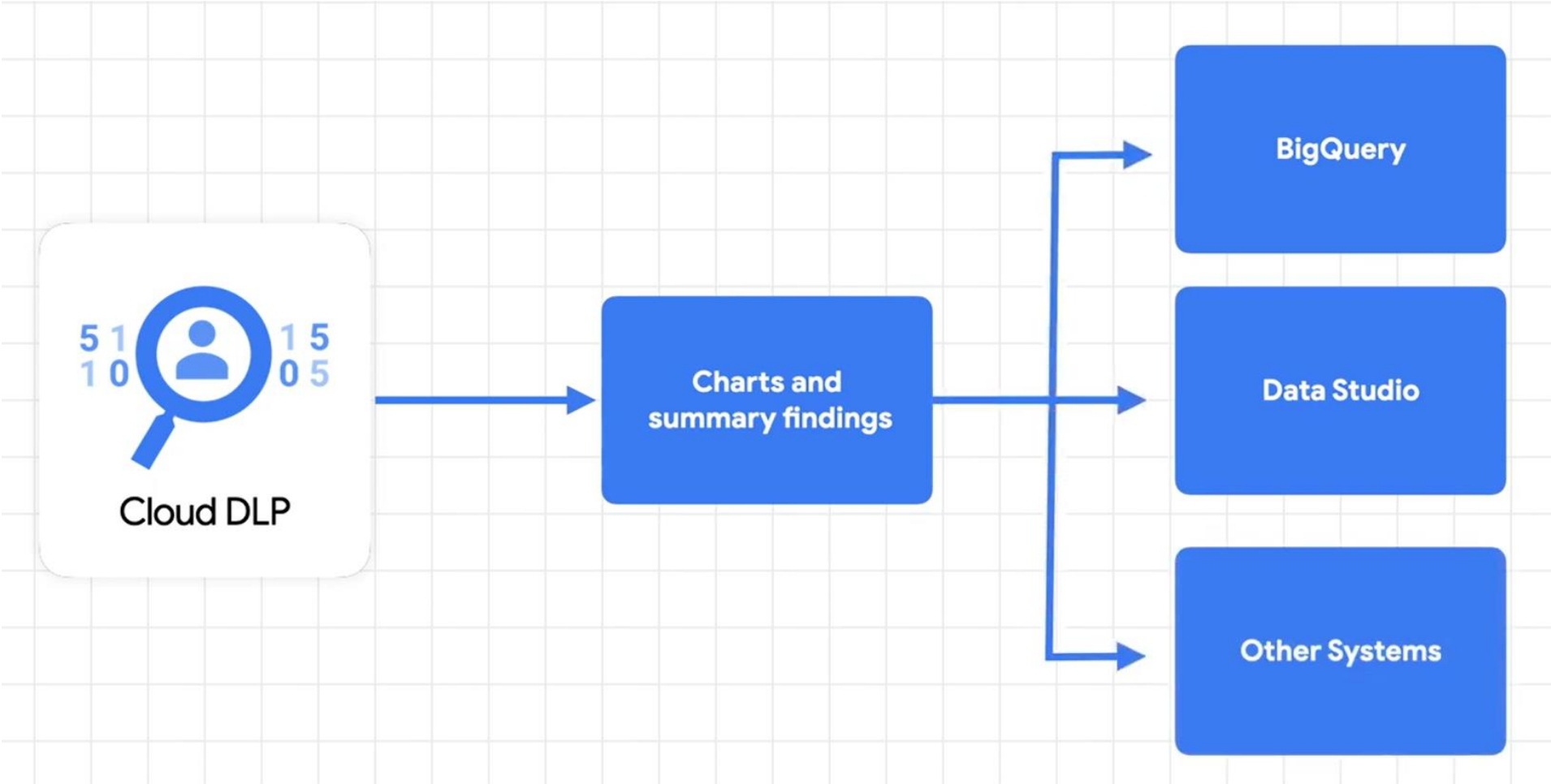
Data Auditing:

- Scanning existing datasets for sensitive information to ensure that they are in compliance with organizational security policies.

Risk Management:

- Identify high-risk data storage locations and classify data to prioritize and manage the security of sensitive data.

DLP


Cloud DLP
Charts and summary findings
BigQuery
Data Studio
Other Systems

Key Features of Cloud DLP

Data Discovery and Classification:

- Cloud DLP can scan data across various GCP services, such as Cloud Storage, BigQuery, and Datastore, to automatically discover sensitive information.
- It identifies sensitive data types like social security numbers, credit card numbers, names, addresses, etc.

Data Masking and Redaction:

- DLP can be used to redact or mask sensitive data to protect it. For example, you can replace parts of sensitive information with random characters, or redact the information entirely.

Tokenization and Pseudonymization:

- Cloud DLP supports tokenization, which replaces sensitive data with reversible tokens, as well as pseudonymization, where original data is replaced with artificial data to ensure privacy.

Data De-Identification:

- Cloud DLP enables de-identification techniques like:
- Data masking (e.g., masking all but the last four digits of a credit card number).
- Format-preserving tokenization (where sensitive data is replaced but in a format that can still be useful).
- Bucketing (grouping data into ranges, like replacing exact ages with age ranges).

Key Features of Cloud DLP

Built-in and Custom Detectors:

- Cloud DLP has a wide range of built-in detectors for common sensitive information like PII, financial data, and healthcare data.
- You can also create custom detectors to identify organization-specific data patterns.

Support for Structured and Unstructured Data:

- It can handle both structured data (databases, tables) and unstructured data (documents, logs) in various formats such as JSON, CSV, text, and more.

Policy-Based Control:

- You can create policies that define what types of sensitive information you want to inspect, the level of confidence required for detection, and the actions that should be taken (like reporting, redacting, or masking data).

Integration with Other GCP Services:

- Cloud DLP integrates with other Google Cloud services like BigQuery, Cloud Storage, Datastore, and Pub/Sub for easier inspection and remediation workflows.

Key Features of Cloud DLP

Cloud Storage:

- Scan and protect sensitive data stored in object storage (e.g., redacting files).

BigQuery:

- Discover and protect sensitive data in tables and datasets.

Cloud Datastore:

- Identify sensitive data in NoSQL databases.

Key Concepts of Cloud DLP

InfoTypes: These are pre-defined types of sensitive information that Cloud DLP can detect (e.g., credit card numbers, phone numbers, names).

Likelihood: This measures the confidence that detected info is sensitive data (e.g., possible, likely, very likely).

Inspection Jobs: Jobs that scan your datasets and report findings of sensitive data.

De-identification: Replaces sensitive data with non-sensitive alternatives.

What is Data de-identification?

Data de-identification is the process of removing or altering personally identifiable information (PII) from datasets to protect the privacy of individuals.

In the context of data protection and privacy, it ensures that sensitive data cannot be easily traced back to the original individual, while still preserving the utility of the data for analytics, research, or other purposes.

Types of Data de-identification

- **Anonymization**: Irreversibly removes personally identifiable information to prevent re-identification.
- **Pseudonymization**: Replaces PII with artificial identifiers while allowing re-identification via a secure reference.
- **Masking**: Hides portions of sensitive data while keeping some parts visible for analysis.
- **Tokenization**: Substitutes sensitive data with non-sensitive tokens, reversible via a secure key.
- **Data Suppression**: Removes sensitive data fields entirely from the dataset.
- **Generalization (Bucketing)**: Reduces data granularity by grouping values into broader categories.

De-identification Techniques

- Replacement
- Masking
- Redaction
- Crypto-based tokenization
- Bucketing
- Date shifting
- Time extraction

De-identification Techniques

- Redaction is the process of removing or obscuring sensitive or confidential information from documents, text, or data to prevent unauthorized access or disclosure. In the context of data protection and privacy, redaction ensures that specific parts of the content, such as personally identifiable information (PII), financial details, or classified data, are not visible or accessible to unintended parties.

Common Examples of Redaction:

- **Text Documents:**
  - Removing names, addresses, or sensitive personal details in legal or official documents before sharing with others.
  - Example: Hiding part of a social security number: 123-**-****.
- **Images:**
  - Blurring or blacking out sections of images, such as license plates, faces, or confidential text within scanned documents or photographs.
- **Audio/Video:**
  - Muting or editing out sensitive spoken information from an audio recording or video clip.
- **Digital Data:**
  - Masking or replacing sensitive information in databases or logs (e.g., credit card numbers, email addresses).

Crypto based tokenization

Crypto-based tokenization (also referred to as "pseudonymization") transformations are de-identification methods that replace the original sensitive data values with encrypted values. Sensitive Data Protection supports the following types of tokenization, including transformations that can be reversed and allow for re-identification:

Types of Tokenization:

- Cryptographic Hashing
- Format Preserving Encryption (FPE)
- Deterministic Encryption

Crypto based tokenization

1. Cryptographic Hashing

Transformation: Converts sensitive data into a fixed-length string using a hash function. This process is non-reversible.

Example:

Input: "1234-5678-9101-1121"

Output: HMAC-SHA-256 Hash: "aW5QkUu9M0Ys/7P5Wl8K9Dw2uP3O5iJ29Er3xKc1TeE="

Description: The original credit card number is transformed into a hash using a secure hash algorithm (SHA-256). This hash cannot be reversed back to the original number.

Crypto based tokenization

2. Format Preserving Encryption (FPE)

Transformation: Encrypts the input data while preserving its format (length and character set).

Example:

Input: "1234-5678-9101-1121"

Output: "XyZ1-2345-6789-1234"

Description: The original credit card number is replaced with an encrypted token that has the same format (e.g., 4-4-4-4 digit grouping) as the input, allowing for re-identification using the original encryption key.

Crypto based tokenization

3. Deterministic Encryption

Transformation: Uses a symmetric encryption algorithm (e.g., AES-SIV) that generates the same token for identical input values.

Example:

Input: "1234-5678-9101-1121"

Output: "Token1234567891011121"

Description: When the same input is provided, it generates the same output token, enabling re-identification. The token can be used in transactions while the original value is securely stored.

DLP Templates

In Google Cloud's Data Loss Prevention (DLP), DLP templates allow you to define reusable configurations for inspection (detection of sensitive data) and de-identification (masking, redacting, or transforming sensitive data). Templates help standardize how sensitive data is handled across different workflows without having to reconfigure settings each time.

There are two main types of DLP templates:

1. Inspection Templates
2. De-identification Templates

DLP Templates

1. **Inspection Templates:**

- Define how to inspect data for sensitive information such as PII (Personally Identifiable Information), financial data, health data, etc.
- These templates specify what type of data to look for (e.g., credit card numbers, names, emails) and where to search (e.g., in databases, files, or text).
- **Use Case**: A company may create an inspection template to scan all documents uploaded by users to detect PII before storing them in a database.

DLP Templates

1. **Inspection Templates:**

```
{
  "displayName": "PII Inspection Template",
  "description": "Inspect documents for PII like SSNs, credit card
numbers, and emails",
  "infoTypes": [
    {"name": "EMAIL_ADDRESS"},
    {"name": "CREDIT_CARD_NUMBER"},
    {"name": "SSN"}
  ],
  "minLikelihood": "LIKELY",
  "limits": {
    "maxFindingsPerRequest": 100
  }
}
```

DLP Templates

Lab: Create an inspection request

Create a file inspect-request.json

```
{
  "item":{
    "value":"My phone number is (800) 555-0123."
  },
  "inspectConfig":{
    "infoTypes":[
      {
        "name":"PHONE_NUMBER"
      },
      {
        "name":"US_TOLLFREE_PHONE_NUMBER"
      }
    ],
    "minLikelihood":"POSSIBLE",
    "limits":{
      "maxFindingsPerItem":0
    },
    "includeQuote":true
  }
}
```

DLP Templates

Lab: Create an inspection request

Run the command to inspect it

```
cd ~ && curl -s \
    -H "Authorization: Bearer $(gcloud auth print-access-token)" \
    -H "X-Goog-User-Project: techlanders-internal" \
    -H "Content-Type: application/json" \
    https://dlp.googleapis.com/v2/projects/techlanders-internal/content:inspect \
    -d @inspect-request.json
```

DLP Templates

2. **De-identification Templates:**

- Define how to de-identify sensitive data once it has been detected. This includes redaction, masking, tokenization, or transformation methods.
- You can specify rules on how to handle sensitive fields like anonymizing data, encrypting fields, or replacing sensitive parts of the data with placeholders.
- **Use Case**: A healthcare company might use a de-identification template to redact patient names and replace them with tokens before sharing data for research.

DLP Templates

2. **De-identification Templates:**

```
{
  "displayName": "PII Redaction Template",
  "description": "Redact sensitive PII fields",
  "deidentifyConfig": {
    "infoTypeTransformations": {
      "transformations": [
        {
          "infoTypes": [
            {"name": "EMAIL_ADDRESS"},
            {"name": "CREDIT_CARD_NUMBER"}
          ],
          "primitiveTransformation": {
            "replaceWithInfoTypeConfig": {}
          }
        }
      ]
    }
  }
}
```

Identity Aware Proxy (IAP)

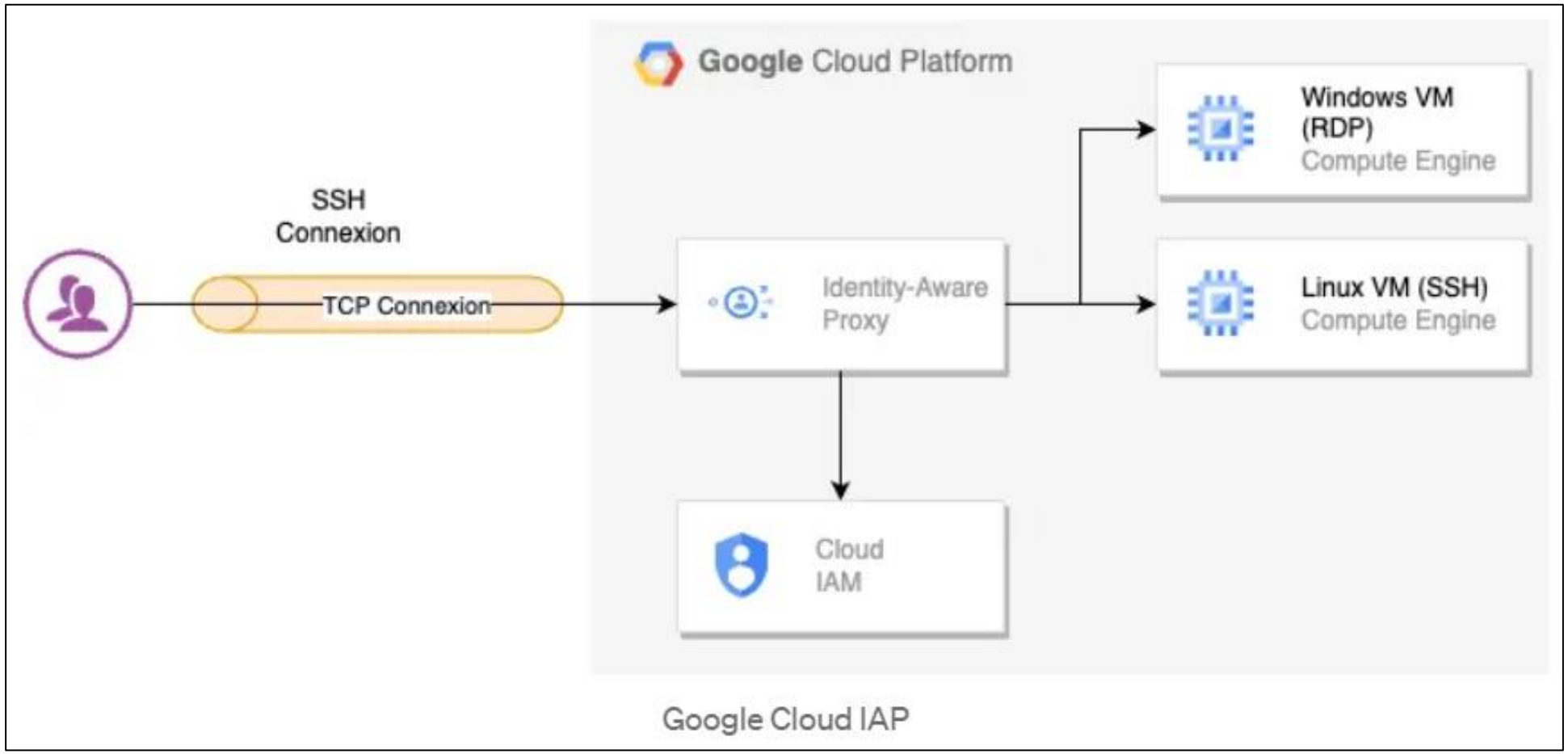
IAP

- Identity-Aware Proxy (IAP) in Google Cloud Platform (GCP) is a security feature that allows you to control access to your web applications and cloud resources.
- It helps enforce identity-based access control for applications running on GCP without requiring any changes to the app itself. IAP ensures that only authenticated and authorized users can access the apps or resources behind it.

Benefits of Using IAP:

- **No need for VPNs or Bastion Hosts**: With IAP, users can access internal applications securely over the internet without needing to set up complex VPNs or bastion hosts.
- **Simplified Identity-based Access**: IAP simplifies the process of granting access to internal applications based on user identity and roles.
- **Integration with Google's Security Stack**: IAP is tightly integrated with other Google Cloud security tools, allowing for comprehensive identity and access management.

IAP


Google Cloud Platform
SSH
Connexion
TCP Connexion
Identity-Aware
Proxy
Windows VM
(RDP)
Compute Engine
Linux VM (SSH)
Compute Engine
Cloud
IAM
Google Cloud IAP

IAP

Lab: SSH connection to a VM

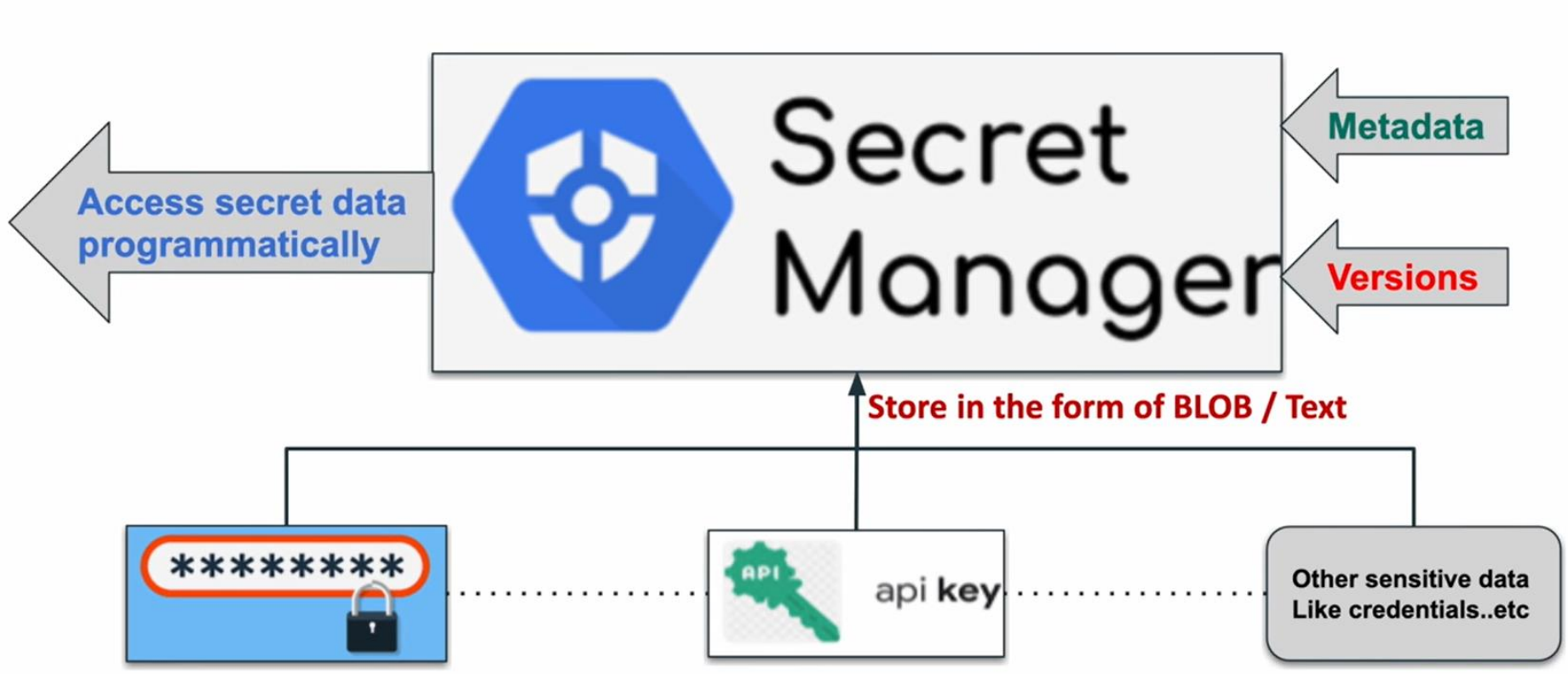
1. Create an instance in a Subnet
2. For IAP, first we must authorize the user logging via IAP by granting them at least the 3 roles below. Most of the time, the user that need to login like this is one of the Admin users, so they should already have them.
   - OS Login role.
   - IAP-secured Tunnel User
   - Compute Viewer
3. Create a firewall rule
   - To allow IAP to connect to your VM instances, create a firewall rule that:
   - applies to all VM instances that you want to be accessible by using IAP.
   - allows ingress traffic from the IP range **35.235.240.0/20**. This range contains all IP addresses that IAP uses for TCP forwarding.
   - allows connections to all ports that you want to be accessible by using IAP TCP forwarding, for example, port **22** for SSH and port **3389** for RDP.
4. Now try to access the VM using browser or below gcloud command
   - gcloud compute ssh INSTANCE_NAME --zone ZONE

Secret Manager

Secret Manager

- Secret manager allows you to store., manage and access secrets as binary blobs or text strings
- User need appropriate permissions to see secret contents
- Secret is a project level global object, contains metadata and its respective versions
- Secret versions stores actual secret data
- Metadata includes Replication locations, labels, annotations and permissions
- User cases: to store and access below details
  - Passwords
  - API Keys
  - TLS certificates

Secret Manager



Secret Manager

Demo:

Python Program

```
pip install google-cloud-secret-manager
```

```
def access_secret_version(project_id, secret_id, version_id):
    """
    Access the payload for the given secret version if one exists. The version
    can be a version number as a string (e.g. "5") or an alias (e.g. "latest").
    """

    # Import the Secret Manager client library.
    from google.cloud import secretmanager

    # Create the Secret Manager client.
    client = secretmanager.SecretManagerServiceClient()

    # Build the resource name of the secret version.
    name = f"projects/{project_id}/secrets/{secret_id}/versions/{version_id}"

    # Access the secret version.
    response = client.access_secret_version(request={"name": name})
    # Print the secret payload.
    # snippet is showing how to access the secret material.
    payload = response.payload.data.decode("UTF-8")
    print("Plaintext: {}".format(payload))

# Function call to show output
access_secret_version('techlanders-internal', 'demo_secret','1')
```

Disaster Recovery

RPO vs RTO

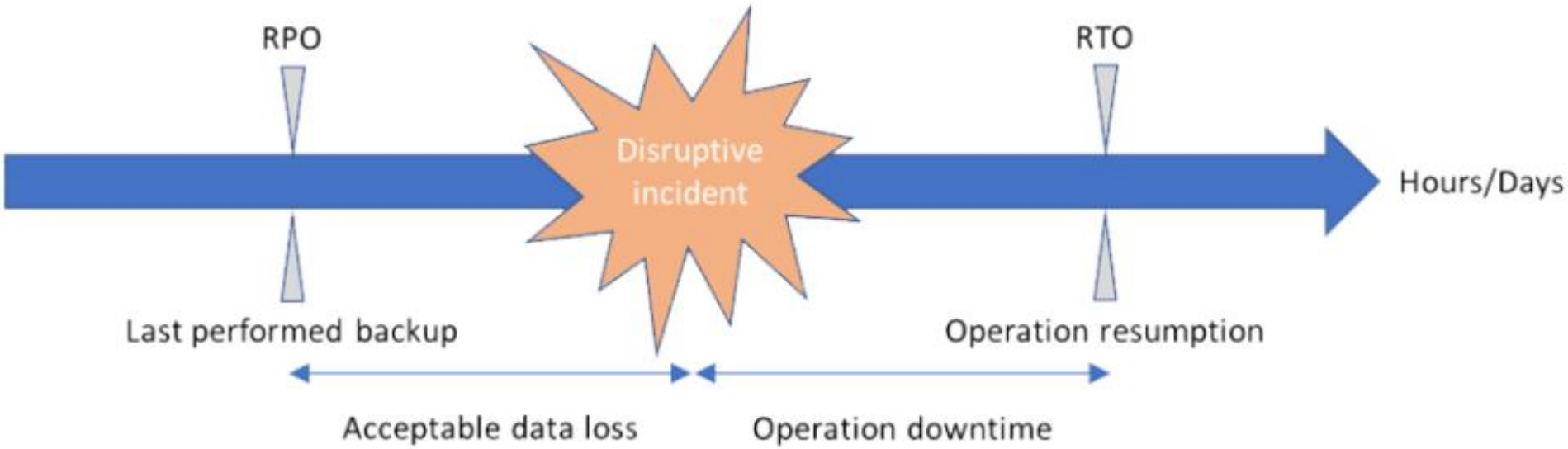
RPO:

- It is the total amount of data that a business can afford to loose.
- We can calculate the RPO by subtracting the point in time at which the backup or replication is scheduled.

RTO:

- Recovery Time Objective, is the amount of time after a disaster in which business operations need to be resumed and resources are again available for use.
- For example, if the RTO is two hours, then this means you must resume delivery of products or services, or execution of activities, in two hours – therefore, your business continuity and disaster recovery plans need to consider the RTO during their development.

RPO vs RTO


RPO
RTO
Disruptive incident
Hours/Days
Last performed backup
Operation resumption
Acceptable data loss
Operation downtime

Data Backup

Organizations face multiple threats securing data is a big concern

- Malware attack
- Compromising account
- Unauthorized access and data exfilteration

Business agility demands a developer-centric, self-service model. Delegating backups has been a challenge.

- Central backup admin needs governance, enforcement and oversight
- Platform admin/app developer are responsible for backing up their applications

Backup and DR Service

- Centralized management
- Comprehensive monitoring and reporting
- Broad Google Cloud Platform workload support
- Backup vault for immutable, indelible backups
- Empower app developers to take backups

Backup Vault

Immutable, indelible backups

Granular IAM-based access control

Cross-project backup and recovery

Security Command Center integration

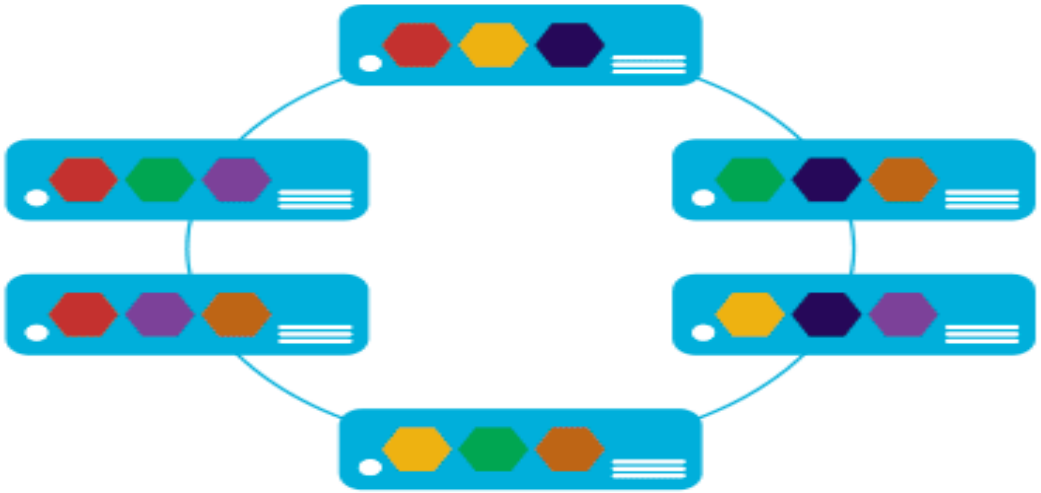
Containers on GCP

Microservice vs Monolithic

Microservices Approach

A microservice approach segregates functionality into small autonomous services.

And scales out by **deploying independently** and replicating these services across servers/VMs/containers.

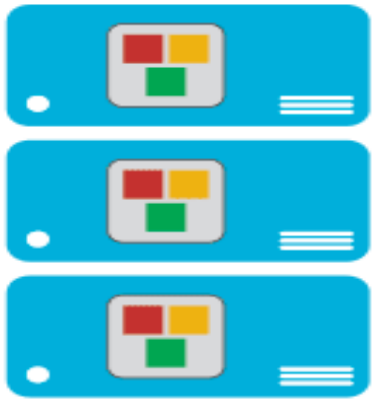


VS.

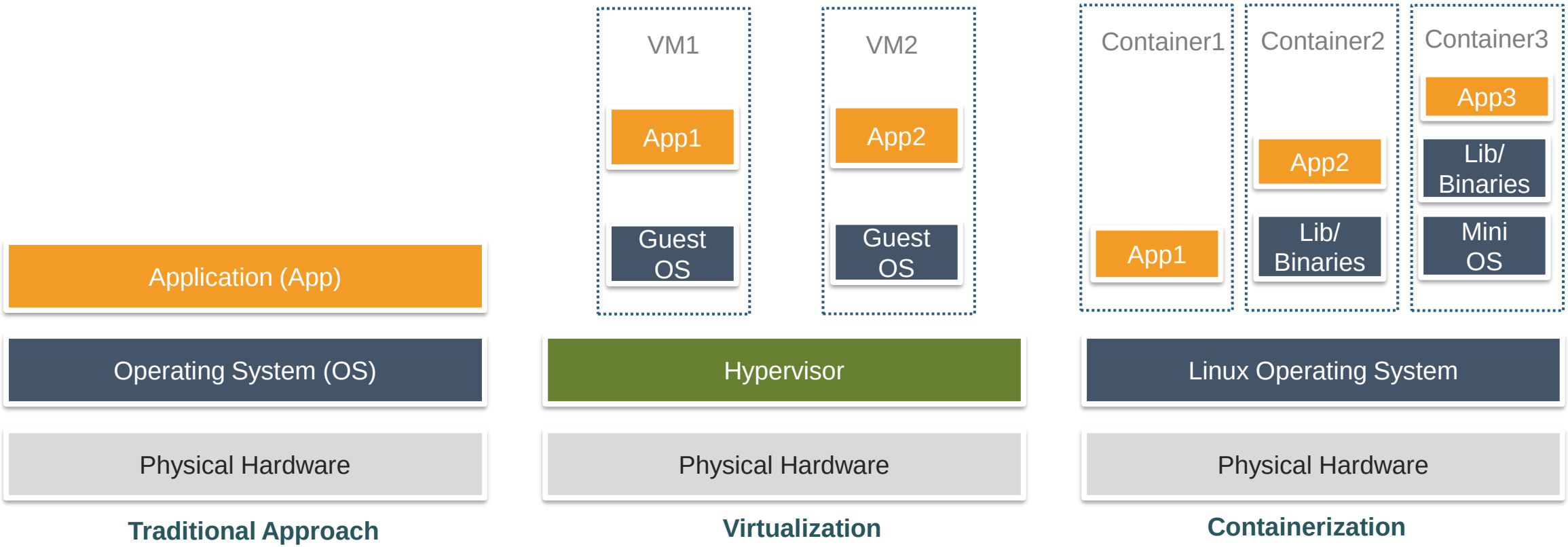
Traditional Approach

A traditional application (Web app or large service) usually has most of its functionality within a single process (usually internally layered, though).

And scales by cloning the whole app on multiple servers/VMs/containers.



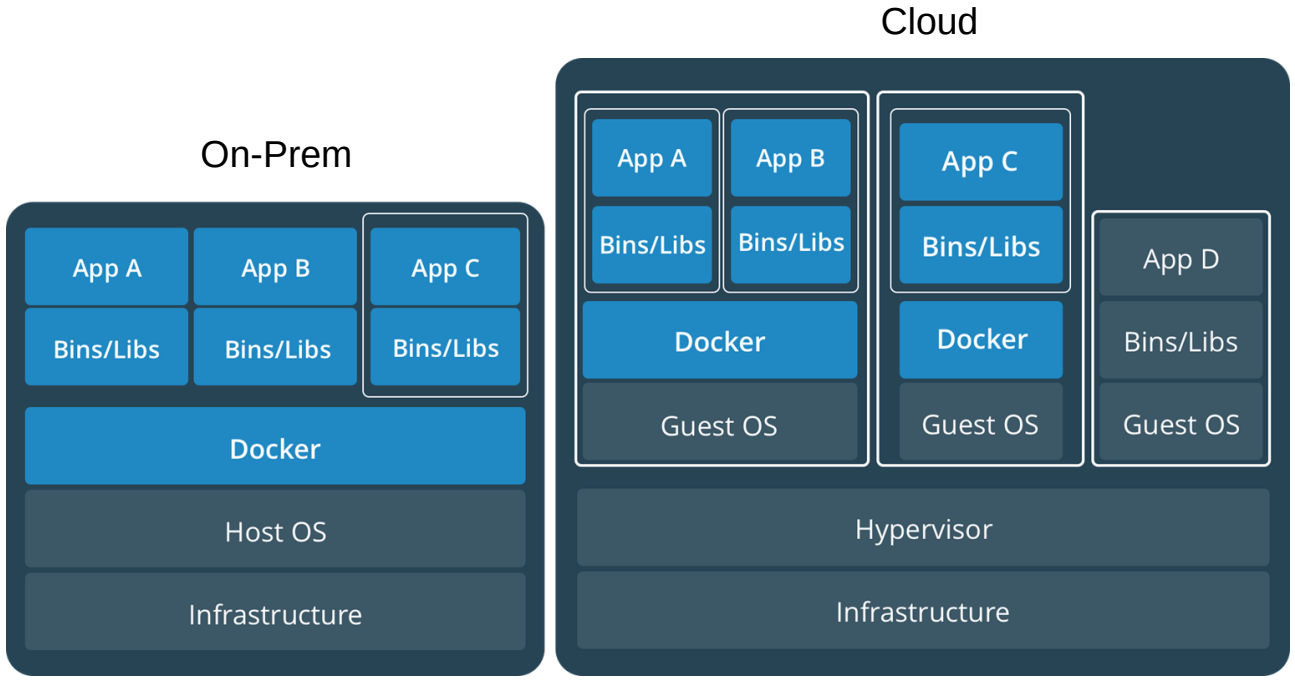
Compute Virtualization



Containers vs VMs

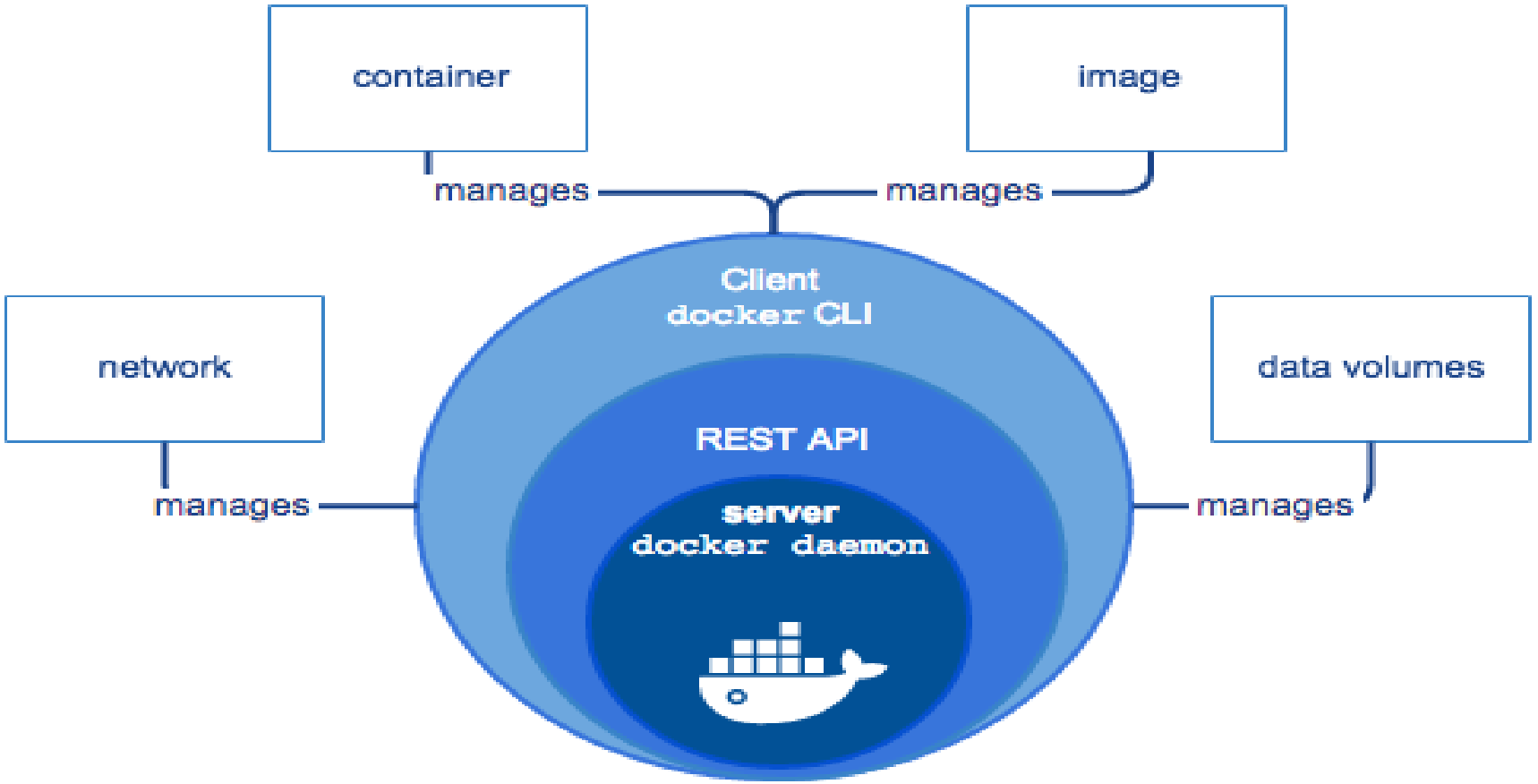
- Container are more light weight
- No need to install dedicated guest OS, no virtualization like VM is required
- Stop/Start time is very fast
- Less CPU, RAM, Storage Space required
- More containers per machine than VM's
- Great Portability

Containers and VM's together

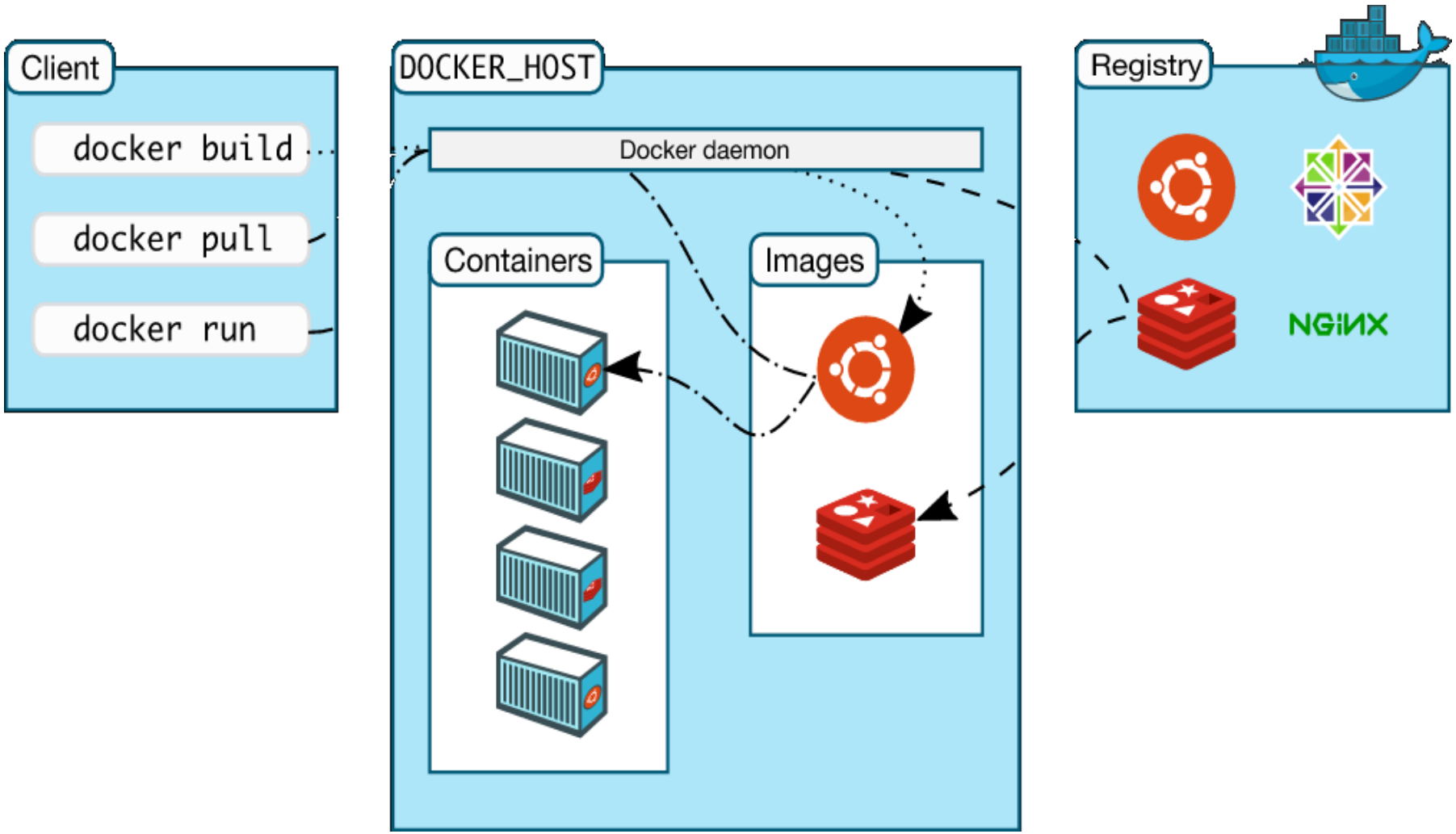


Containers and VMs together provide a tremendous amount of flexibility for IT to optimally deploy and manage apps.

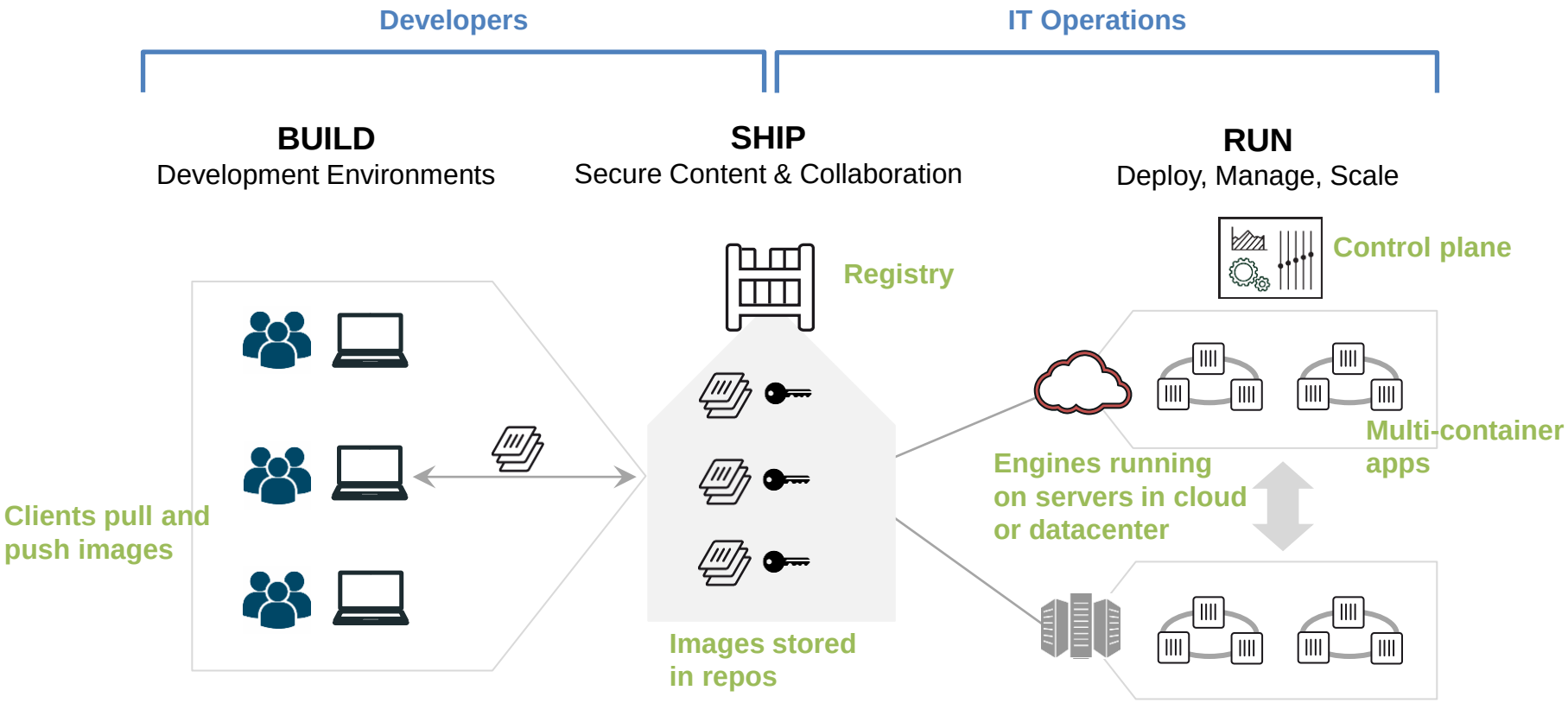
Docker Architecture



Docker Architecture

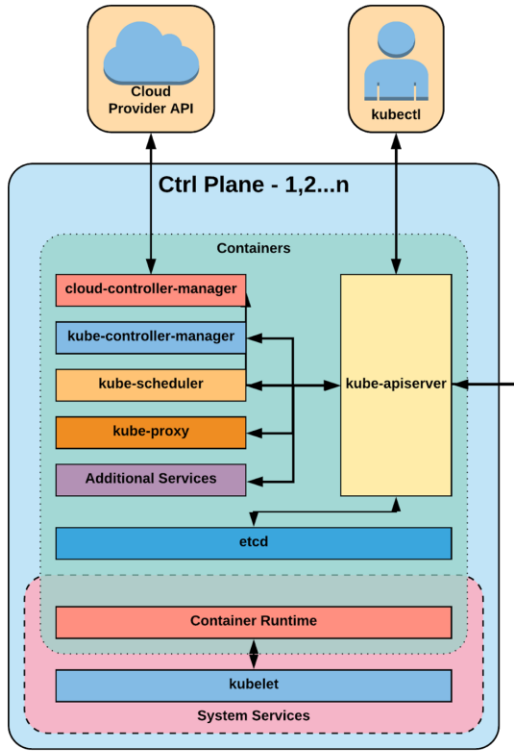


Container Environment



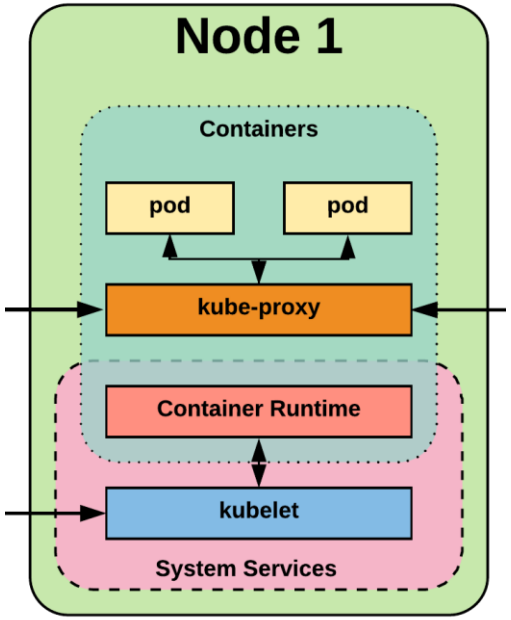
K8S Master Components

- kube-apiserver
- etcd
- kube-controller-manager
- kube-scheduler



K8S Node Components

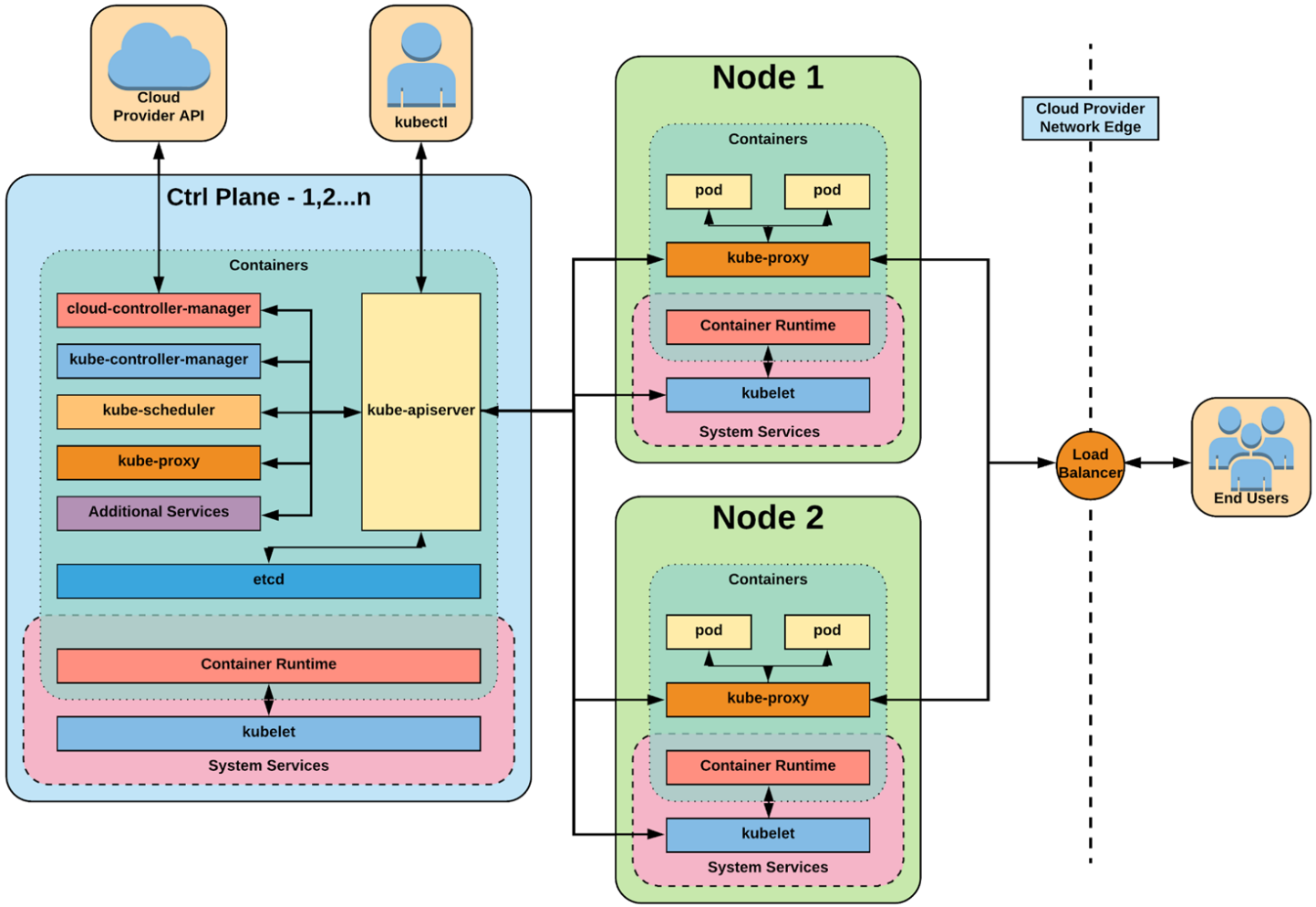
- kubelet
- kube-proxy
- Container Runtime Engine



K8S Terminologies

- Pod
- ReplicaSet/Replication Controller
- Deployments
- Services
- Node Selector
- Secrets
- Persistent Volumes
- And so on….

K8S Architecture



GKE

- Google Kubernetes Engine (GKE) provides a managed environment for deploying, managing, and scaling your containerized applications using Google infrastructure.
- Launched in 2015.
- Kubernetes Engine builds on Google's experience of running services like Gmail and YouTube in containers for over 12 years.
- Kubernetes Engine allows you to get up and running with Kubernetes in no time.
- Eliminates the need to install, manage, and operate your own Kubernetes clusters.

GKE

- Start quickly with single-click clusters
- Leverage a high-availability control plane including multi-zonal and regional clusters
- Eliminate operational overhead with auto-repair, auto-upgrade, and release channels
- Secure by default, including vulnerability scanning of container images and data encryption
- Integrated Cloud Monitoring with infrastructure, application, and Kubernetes-specific views

GKE

- Start quickly with single-click clusters
- Leverage a high-availability control plane including multi-zonal and regional clusters
- Eliminate operational overhead with auto-repair, auto-upgrade, and release channels
- Secure by default, including vulnerability scanning of container images and data encryption
- Integrated Cloud Monitoring with infrastructure, application, and Kubernetes-specific views

GKE features

- Identity and access management
- GKE is backed by a Google security team of over 750 experts and is both HIPAA and PCI DSS compliant.
- Integrated logging and monitoring
- Auto scale, Auto-upgrade and Auto-Repair
- Fully Managed
- Private container registry
- Preemptible VMs and persistent disk support
- Software supply chain security
- Per-second billing

LAB: GKE

- Create a cluster using GCP Console
- Get the credentials for your created cluster to generate kubconfig configuration by running:
- gcloud container clusters get-credentials <CLUSTER-NAME> --zone <ZONENAME> --project <PROJECTNAME>
- Now let perform an webapplication deployment by creating kubernetes deployments
- kubectl run gaganwebapp --image=httpd
- Check the workload now (pods, replicasets and deployments)
- Its time to expose our application outside, lets us create a service now.
- kubectl expose deployment gaganwebapp --type=LoadBalancer --port 80
- Confirm our webserver is running on container.

GKE

- A test:
- Let us delete our container and see what will happen (delete pod and see the magic)
- Let us scale our container by creating more replicas
- kubectl scale deployment gaganwebapp --replicas=3
- Note: I will be demonstrating the whole setup using GCP console.

GKE

- GKE cluster masters are automatically upgraded to run new versions of Kubernetes as those versions become stable, so you can take advantage of newer features from the open source Kubernetes project.

Container Registry

Container Registry is a single place for your team to manage Docker images, perform vulnerability analysis, and decide who can access what with fine-grained access control.

- Secure, private Docker registry
- Build and deploy automatically
- In-depth vulnerability scanning
- In-depth vulnerability scanning
- Native Docker support
- Fast, high-availability access

Registry Authentication methods:
https://cloud.google.com/container-registry/docs/advanced-authentication

LAB: Container Registry

Create a Docker image and push it to Google registry:

Follow steps on:
https://cloud.google.com/container-registry/docs/quickstart

GCP Functions

Cloud Functions

Google Cloud Functions is a **serverless** execution environment for building and connecting cloud services.

With Cloud Functions you write simple, **single-purpose functions** that are **attached to events emitted from your cloud infrastructure and services**.

Your Cloud Function is **triggered when an event being watched is fired**.

Your **code executes in a fully managed environment**.

There is **no need to provision any infrastructure** or worry about managing any servers.

GCP Security

Audit and Logs

Audit Logs, Audit trail logs and System logs can be seen at Logs section.

- Securely store, search, analyze, and alert on all of your log data and events
- Ingest custom log data from any source
- An exabyte-scale, fully managed service for your application and infrastructure logs
- Analyze log data in real time

https://forms.gle/ZVj2NeZWnmVA791J8

THANK YOU