

Google Cloud Digital Leader

Getting Started



Compute
Engine



Cloud
Functions



Cloud
Datastore



Cloud SQL



App
Engine

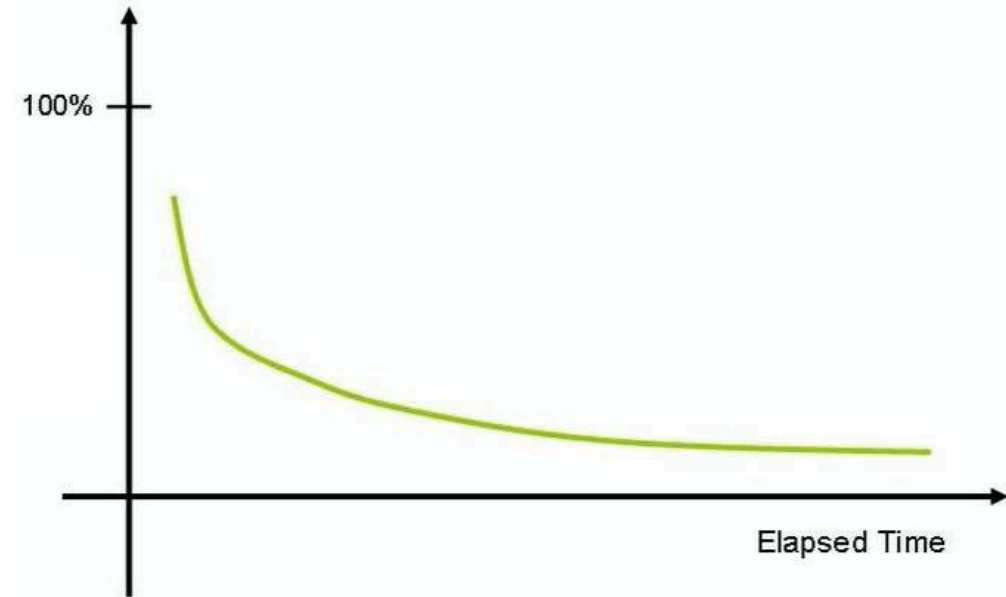


Container
Engine

- GCP has *200+ services* :
 - This exam expects knowledge of *40+ Services*
- Exam *tests* your **decision making abilities**:
 - Which service do you choose in which situation?
- This course is **designed** to help you *make these choices*
- **Our Goal** : Help you start your cloud journey AND get certified

How do you put your best foot forward?

- **Challenging certification** - Expects you to understand and **REMEMBER** a number of services
- As time passes, humans forget things.
- How do you improve your chances of remembering things?
 - **Active learning** - think and take notes
 - **Review** the presentation every once in a while



Our Approach

- Three-pronged approach to reinforce concepts:
 - Presentations (Video)
 - Demos (Video)
 - **Two kinds of quizzes:**
 - Text quizzes
 - Video quizzes
- (Recommended) Take your time. Do not hesitate to replay videos!
- (Recommended) Have Fun!



FASTEST ROADMAPS

in28minutes.com



In28
Minutes



Google Cloud
Certifications



Azure
Certifications



AWS
Certifications



DevOps



Java Full Stack

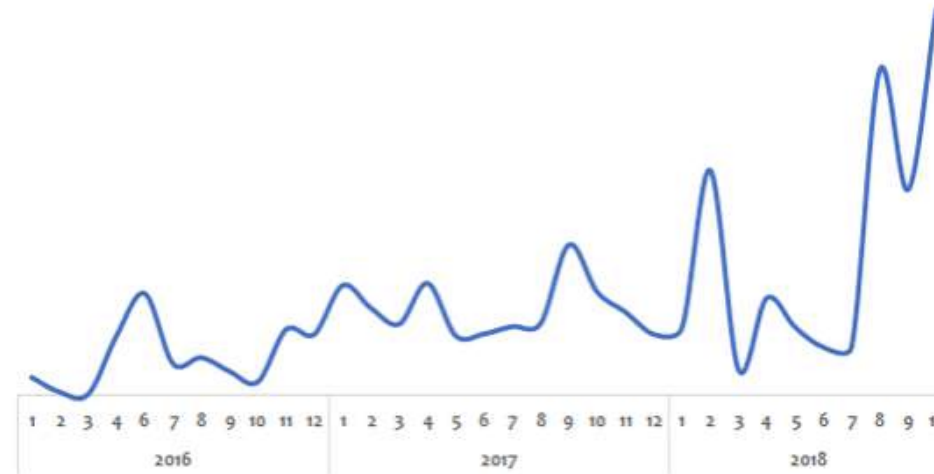


Java Microservices



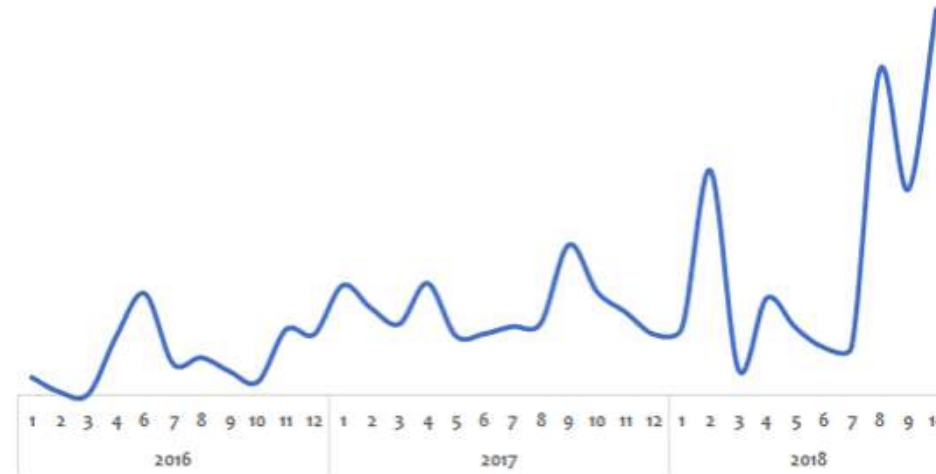
GCP - Getting started

Before the Cloud - Example 1 - Online Shopping App



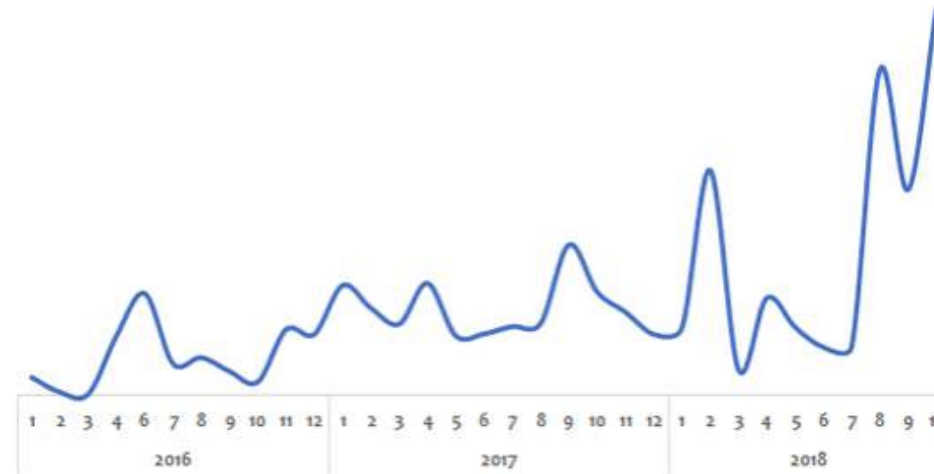
- Challenge:
 - Peak usage during holidays and weekends
 - Less load during rest of the time
- Solution (before the Cloud):
 - **PEAK LOAD provisioning : Procure (Buy) infrastructure for peak load**
 - What would the infrastructure be doing during periods of low loads?

Before the Cloud - Example 2 - Startup



- Challenge:
 - Startup suddenly becomes popular
 - How to handle the **sudden increase** in load?
- Solution (before the Cloud):
 - **Procure** (Buy) infrastructure assuming they would be successful
 - What if they are not successful?

Before the Cloud - Challenges



- High cost of procuring infrastructure
- Needs ahead of time planning (**Can you guess the future?**)
- Low infrastructure utilization (**PEAK LOAD** provisioning)
- Dedicated infrastructure maintenance team (**Can a startup afford it?**)

Silver Lining in the Cloud

- How about **provisioning** (renting) **resources** when you want them and releasing them back when you do not need them?
 - On-demand resource provisioning
 - Also called **Elasticity**



Cloud - Advantages

- Trade "capital expense" for "variable expense"
- Benefit from massive economies of scale
- Stop guessing capacity
- Stop spending money running and maintaining data centers
- "Go global" in minutes



Google Cloud Platform (GCP)

- One of the Top 3 cloud service providers
- Provides a number of services (200+)
- Reliable, secure and highly-performant:
 - Infrastructure that powers 8 services with over 1 Billion Users: Gmail, Google Search, YouTube etc
- One thing I love : "**cleanest cloud**"
 - Net carbon-neutral cloud (electricity used matched 100% with renewable energy)
- The entire course is all about GCP. You will learn it as we go further.



Google Cloud

Best path to learn GCP!



Compute
Engine



Cloud
Functions



Cloud
Datastore



Cloud SQL



App
Engine



Container
Engine

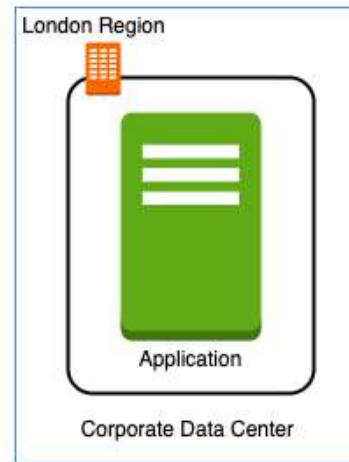
- Cloud applications make use of multiple GCP services
- There is **no single path** to learn these services independently
- HOWEVER, we've worked out a simple path!

Setting up GCP Account

- Create GCP Account

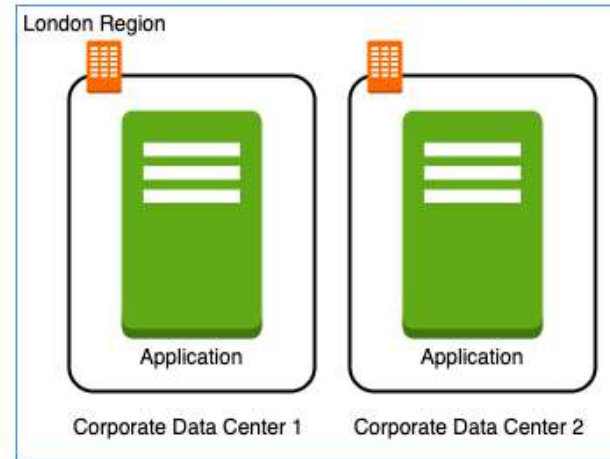
Regions and Zones

Regions and Zones



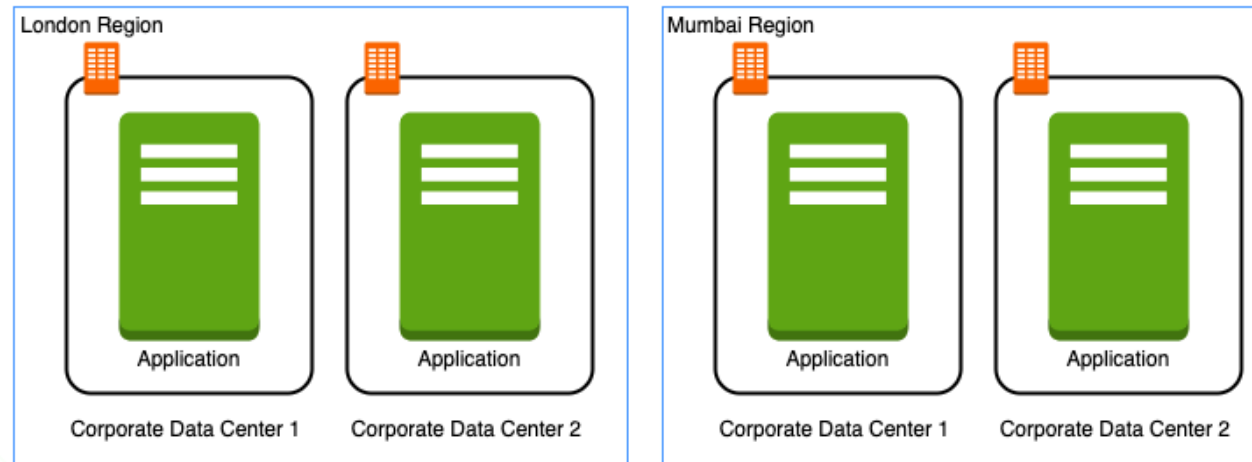
- Imagine that your application is deployed in a data center in London
- What would be the challenges?
 - Challenge 1 : Slow access for users from other parts of the world (**high latency**)
 - Challenge 2 : What if the data center crashes?
 - Your application goes down (**low availability**)

Multiple data centers



- Let's add in one more data center in London
- What would be the challenges?
 - Challenge 1 : Slow access for users from other parts of the world
 - Challenge 2 (**SOLVED**) : What if one data center crashes?
 - Your application is **still available** from the other data center
 - Challenge 3 : What if **entire region** of London is unavailable?
 - Your application goes down

Multiple regions



- Let's add a new region : Mumbai
- What would be the challenges?
 - Challenge 1 (**PARTLY SOLVED**) : Slow access for users from other parts of the world
 - You can solve this by adding deployments for your applications in other regions
 - Challenge 2 (**SOLVED**) : What if one data center crashes?
 - Your application is still live from the other data centers
 - Challenge 3 (**SOLVED**) : What if entire region of London is unavailable?
 - Your application is served from Mumbai

Regions

- Imagine setting up data centers in different regions around the world
 - Would that be easy?
- (Solution) Google provides **20+ regions** around the world
 - Expanding every year
- **Region** : Specific geographical location to host your resources
- **Advantages:**
 - High Availability
 - Low Latency
 - Global Footprint
 - Adhere to government **regulations**



Zones

- How to achieve high availability in the same region (or geographic location)?
 - Enter **Zones**
- Each Region has three or more **zones**
- (Advantage) **Increased availability and fault tolerance** within same region
- (Remember) Each Zone has **one or more discrete clusters**
 - **Cluster** : distinct physical infrastructure that is housed in a data center
- (Remember) Zones in a region are connected through **low-latency** links



Regions and Zones examples

New Regions and Zones are constantly added

| Region Code | Region | Zones | Zones List |
|-------------|-----------------------------------|-------|---|
| us-west1 | The Dalles, Oregon, North America | 3 | us-west1-a us-west1-b us-west1-c |
| eu-north1 | Helsinki, Finland, Europe | 3 | eu-north1-a, eu-north1-b eu-north1-c |
| asia-south1 | Mumbai, India APAC | 3 | asia-south1-a, asia-south1-b asia-south1-c |

Compute

Google Compute Engine (GCE)

- In corporate data centers, applications are deployed to physical servers
- Where do you deploy applications in the cloud?
 - Rent virtual servers
 - **Virtual Machines** - Virtual servers in GCP
 - **Google Compute Engine (GCE)** - Provision & Manage Virtual Machines



Compute Engine - Features



Compute
Engine



Persistent
Disk



Cloud Load
Balancing

- Create and manage lifecycle of Virtual Machine (VM) instances
- **Load balancing** and **auto scaling** for multiple VM instances
- **Attach storage** (& network storage) to your VM instances
- Manage **network connectivity and configuration** for your VM instances
- **Our Goal:**
 - Setup VM instances as HTTP (Web) Server
 - Distribute load with Load Balancers

Compute Engine Hands-on

- Let's create a few VM instances and play with them
- Let's check out the lifecycle of VM instances
- Let's use SSH to connect to VM instances



Compute Engine Hands-on : Setting up a HTTP server

```
#!/bin/bash
sudo su
apt update
apt -y install apache2
sudo service apache2 start
sudo update-rc.d apache2 enable
echo "Hello World" > /var/www/html/index.html
echo "Hello world from $(hostname) $(hostname -I)" > /var/www/html/index.html
```

- Commands:
 - `sudo su` - execute commands as a root user
 - `apt update` - Update package index - pull the latest changes from the APT repositories
 - `apt -y install apache2` - Install apache 2 web server
 - `sudo service apache2 start` - Start apache 2 web server
 - `echo "Hello World" > /var/www/html/index.html` - Write to index.html
 - `$(hostname)` - Get host name
 - `$(hostname -I)` - Get host internal IP address

IP Addresses - Virtual Machines

| IP Address | Description |
|----------------------------------|---|
| Internal IP Address | Permanent Internal IP Address that does not change during the lifetime of an instance |
| External or Ephemeral IP Address | Ephemeral External IP Address that changes when an instance is stopped |
| Static IP Address | Permanent External IP Address that can be attached to a VM |

Simplify VM HTTP server setup

- How do we **reduce the number of steps** in creating an VM instance and setting up a HTTP Server?
- Let's explore a few options:
 - Startup script
 - Instance Template
 - Custom Image



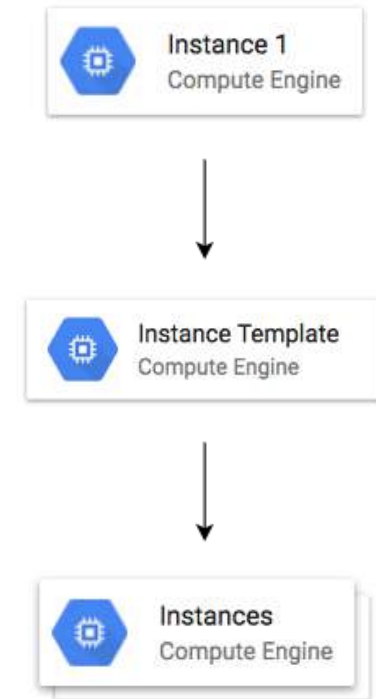
Bootstrapping with Startup script

```
#!/bin/bash  
apt update  
apt -y install apache2  
echo "Hello world from $(hostname) $(hostname -I)" > /var/www/html
```

- **Bootstrapping:** Install OS patches or software when an VM instance is launched.
- In VM, you can configure **Startup script** to bootstrap
- **DEMO** - Using Startup script

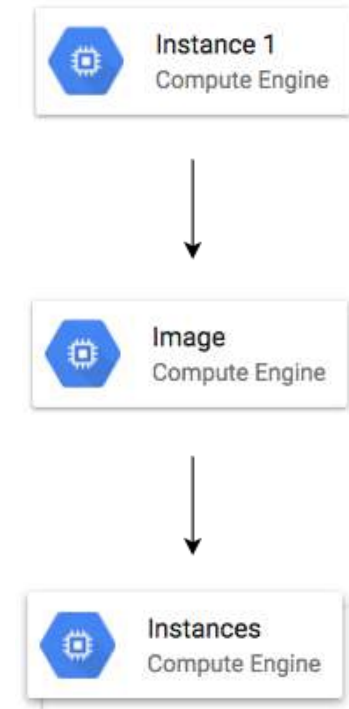
Instance templates

- Why do you need to specify all the VM instance details (Image, instance type etc) **every time** you launch an instance?
 - How about creating a **Instance template**?
 - Define **machine type**, **image**, **labels**, **startup script** and other properties
- Used to create **VM instances** and **managed instance groups**
 - Provides a **convenient way** to create similar instances
- **CANNOT** be updated
 - To make a change, copy an existing template and modify it
- (Optional) Image family can be specified (example - debian-9):
 - Latest non-deprecated version of the family is used
- **DEMO** - Launch VM instances using Instance templates



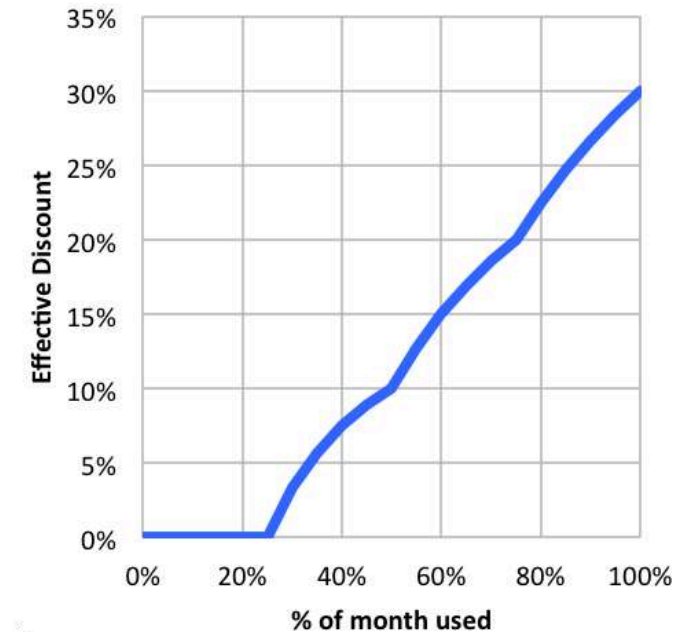
Reducing Launch Time with Custom Image

- Installing OS patches and software at launch of VM instances **increases boot up time**
- How about creating a custom image with OS patches and software **pre-installed**?
 - Can be created from an instance, a persistent disk, a snapshot, another image, or a file in Cloud Storage
 - Can be shared across projects
 - (Recommendation) Deprecate old images (& specify replacement image)
 - (Recommendation) **Hardening an Image** - Customize images to your corporate security standards
- **Prefer using Custom Image to Startup script**
- **DEMO** : Create a Custom Image and using it in an Instance Template



Sustained use discounts

- **Automatic discounts** for running VM instances for significant portion of the billing month
 - Example: If you use N1, N2 machine types for more than 25% of a month, you get a 20% to 50% discount on every incremental minute.
 - Discount increases with usage (graph)
 - No action required on your part!
- **Applicable** for instances created by **Google Kubernetes Engine** and **Compute Engine**
- **RESTRICTION:** Does NOT apply on certain machine types (example: E2 and A2)
- **RESTRICTION:** Does NOT apply to VMs created by App Engine flexible and Dataflow



Source: <https://cloud.google.com>

Committed use discounts

- For workloads with **predictable resource** needs
- **Commit** for 1 year or 3 years
- **Up to 70% discount** based on machine type and GPUs
- **Applicable** for instances created by **Google Kubernetes Engine** and **Compute Engine**
- (Remember) You **CANNOT cancel** commitments
 - Reach out to Cloud Billing Support if you made a mistake while purchasing commitments



Preemptible VM

- **Short-lived cheaper** (upto 80%) compute instances
 - Can be stopped by GCP any time (preempted) within 24 hours
 - Instances get 30 second warning (to save anything they want to save)
- **Use Preempt VM's if:**
 - Your applications are **fault tolerant**
 - You are very **cost sensitive**
 - Your workload is **NOT immediate**
 - Example: Non immediate batch processing jobs
- **RESTRICTIONS:**
 - NOT always available
 - NO SLA and CANNOT be migrated to regular VMs
 - NO Automatic Restarts
 - Free Tier credits not applicable



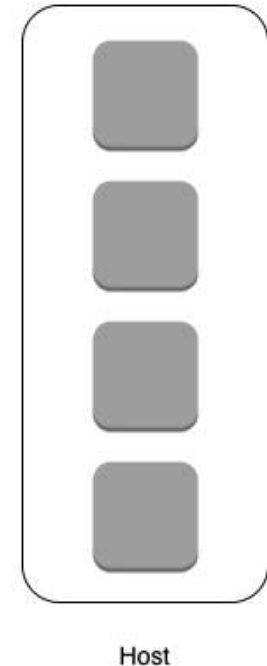
Spot VMs

- **Spot VMs:** Latest version of preemptible VMs
- **Key Difference:** Does not have a maximum runtime
 - Compared to traditional preemptible VMs which have a maximum runtime of 24 hours
- **Other features similar to traditional preemptible VMs**
 - May be reclaimed at any time with 30-second notice
 - NOT always available
 - Dynamic Pricing: 60 - 91% discount compared to on-demand VMs
 - Free Tier credits not applicable



Compute Engine - Sole-tenant Nodes

- **Shared Tenancy (Default)**
 - Single host machine can have instances from multiple customers
- **Sole-tenant Nodes:** Virtualized instances on hardware dedicated to one customer
- **Use cases:**
 - **Security and compliance** requirements: You want your VMs to be physically separated from those in other projects
 - **High performance** requirements: Group your VMs together
 - **Licensing** requirements: Using per-core or per-processor "Bring your own licenses"



Compute Engine Features: Custom Machine Types



- What do you do when predefined VM options are NOT appropriate for your workload?
 - Create a machine type customized to your needs (a **Custom Machine Type**)
- **Custom Machine Type: Adjust vCPUs, memory and GPUs**
 - Choose between E2, N2, or N1 machine types
 - Supports a wide variety of Operating Systems: CentOS, CoreOS, Debian, Red Hat, Ubuntu, Windows etc
 - **Billed per vCPUs, memory provisioned** to each instance
 - Example Hourly Price: $\$0.033174 / \text{vCPU} + \$0.004446 / \text{GB}$

Google Compute Engine - VM Costs



- **2 primary costs** in running VMs using GCE:
 - **1: Infrastructure cost** to run your VMs
 - **2: Licensing cost** for your OS (ONLY for **Premium Images**)
 - **Premium Image Examples:** Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server (SLES), Ubuntu Pro, Windows Server, ..
 - **Options For Licensing:**
 - **1:** You can use Pay-as-you-go model (PAYG) OR
 - **2:** (WITHIN A LOT OF CONSTRAINTS) You can use your existing license/subscription (Bring your own subscription/license - BYOS/BYOL)
- **(RECOMMENDED)** If you have existing license for a premium image, use it while your license is valid
 - After that you can shift to Pay-as-you-go model (PAYG)

Quick Review

Image

- What **operating system** and what **software** do you want on the VM instance?
- Reduce boot time and improve security by creating custom **hardened Images**.
- You can share an Image with other projects

Machine Types

- Optimized combination of compute(CPU, GPU), memory, disk (storage) and networking for specific workloads.
- You can **create your own Custom Machine Types** when existing ones don't fit your needs

Quick Review

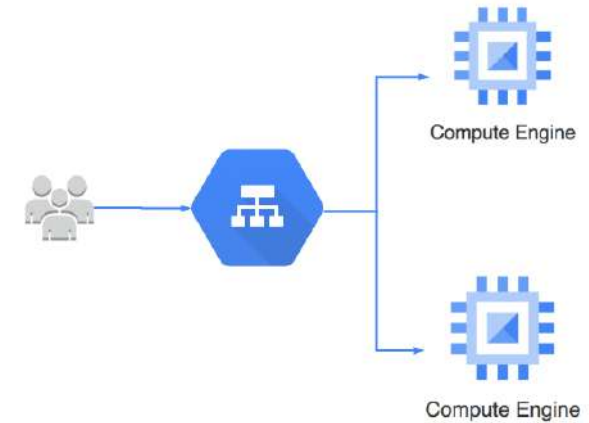
- **Static IP Addresses:** Get a constant IP addresses for VM instances
- **Instance Templates:** Pre-configured templates simplifying the creation of VM instances
- **Sustained use discounts:** Automatic discounts for running VM instances for significant portion of the billing month
- **Committed use discounts:** 1 year or 3 year **reservations** for workloads with **predictable resource needs**
- **Preemptible VM:** Short-lived cheaper (upto 80%) compute instances for non-time-critical fault-tolerant workloads

Google Compute Engine - Scenarios

| Scenario | Service |
|---|-------------------------|
| I want to ensure my VM runs a specific operating system and software stack for my application | Custom Image |
| I need to optimize my VM for a specialized workload requiring a unique mix of CPU, memory, and storage | Custom Machine Types |
| My application requires a fixed IP address that doesn't change between reboots or reassignments | Static IP Addresses |
| I have predictable resource needs and want to commit to a 1 or 3-year plan to enjoy deeper discounts | Committed Use Discounts |
| I need to run short-lived, fault-tolerant workloads that can tolerate interruptions in exchange for lower costs | Preemptible VMs |

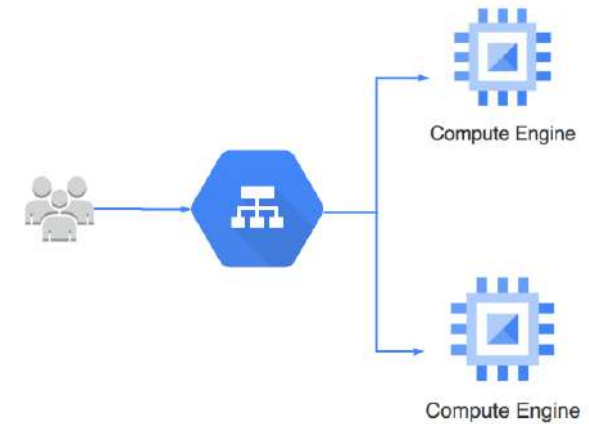
Instance Groups

- How do you create a group of VM instances?
 - **Instance Group** - Group of VM instances managed as a single entity
 - **Manage group** of similar VMs having similar lifecycle as **ONE UNIT**
- **Two Types of Instance Groups:**
 - **Managed** : Identical VMs created using a template:
 - Features: Auto scaling, auto healing and managed releases
 - **Unmanaged** : Different configuration for VMs in same group:
 - Does NOT offer auto scaling, auto healing & other services
 - NOT Recommended unless you need different kinds of VMs
- **Location** can be Zonal or Regional
 - Regional gives you higher availability (RECOMMENDED)



Managed Instance Groups (MIG)

- **Managed Instance Group** - Identical VMs created using an instance template
- **Important Features:**
 - **Maintain** certain number of instances
 - If an instance crashes, MIG launches another instance
 - **Detect application failures** using health checks (Self Healing)
 - Increase and decrease instances based on load (**Auto Scaling**)
 - Add **Load Balancer** to distribute load
 - Create instances in multiple zones (regional MIGs)
 - Regional MIGs provide higher availability compared to zonal MIGs
 - **Release** new application versions without downtime
 - **Rolling updates:** Release new version step by step (gradually). Update a percentage of instances to the new version at a time.
 - **Canary Deployment:** Test new version with a group of instances before releasing it across all instances.



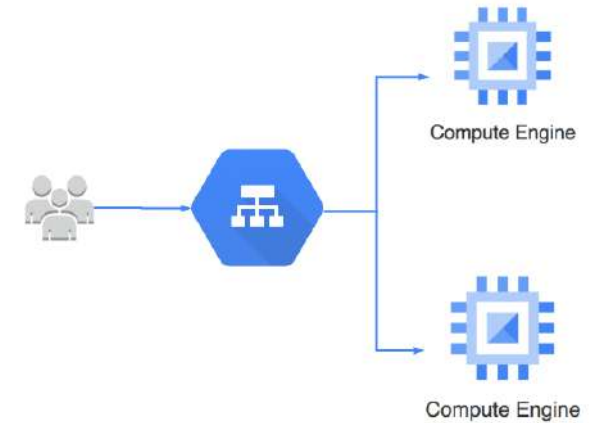
Creating Managed Instance Group (MIG)

- Instance template is mandatory
- Configure **auto-scaling** to automatically adjust number of instances based on load:
 - **Minimum** number of instances
 - **Maximum** number of instances
 - **Autoscaling metrics:** CPU Utilization target or Load Balancer Utilization target or Any other metric from Stack Driver
 - **Cool-down period:** How long to wait before looking at auto scaling metrics again?
 - **Scale In Controls:** Prevent a sudden drop in no of VM instances
 - **Example:** Don't scale in by more than 10% or 3 instances in 5 minutes
 - **Autohealing:** Configure a Health check with Initial delay (How long should you wait for your app to initialize before running a health check?)
- Time for a **Demo**



GCP - Cloud Load Balancing

- Distribute traffic across VM instances in one or more regions
- **Managed service:**
 - Google Cloud ensures that it is highly available
 - Auto scales to handle huge loads
 - Load Balancers can be **public or private**
- **Types:**
 - External HTTP(S)
 - Internal HTTP(S)
 - SSL Proxy
 - TCP Proxy
 - External Network TCP/UDP
 - Internal TCP/UDP



Managed Services

Managed Services

- Do you want to continue **running applications in the cloud**, the **same way you run them in your data center**?
- OR are there **OTHER** approaches?
- You should **understand some terminology** used with cloud services:
 - IaaS (Infrastructure as a Service)
 - PaaS (Platform as a Service)
 - FaaS (Function as a Service)
 - CaaS (Container as a Service)
 - Serverless
- Let's get on a quick **journey** to understand these!



IAAS (Infrastructure as a Service)

- Use **only infrastructure** from cloud provider
- **Example:** Using VM to deploy your applications or databases
- You are responsible for:
 - Application Code and Runtime
 - Configuring load balancing
 - Auto scaling
 - OS upgrades and patches
 - Availability
 - etc.. (and a lot of things!)

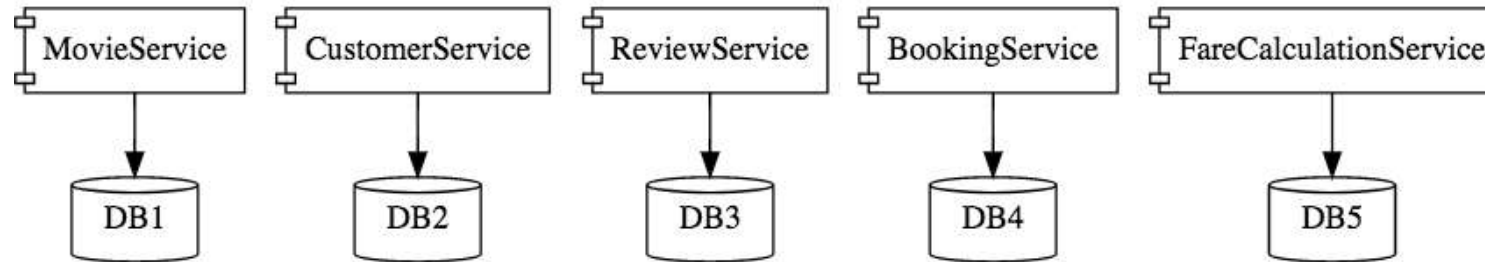


PAAS (Platform as a Service)

- Use a platform provided by cloud
- **Cloud provider** is responsible for:
 - OS (incl. upgrades and patches)
 - Application Runtime
 - Auto scaling, Availability & Load balancing etc..
- **You** are responsible for:
 - Configuration (of Application and Services)
 - Application code (if needed)
- Varieties:
 - **CAAS (Container as a Service)**: Containers instead of Apps
 - **FAAS (Function as a Service)**: Functions instead of Apps
 - Databases - Relational & NoSQL (Amazon RDS, Google Cloud SQL, Azure SQL Database etc), Queues, AI, ML, Operations etc!



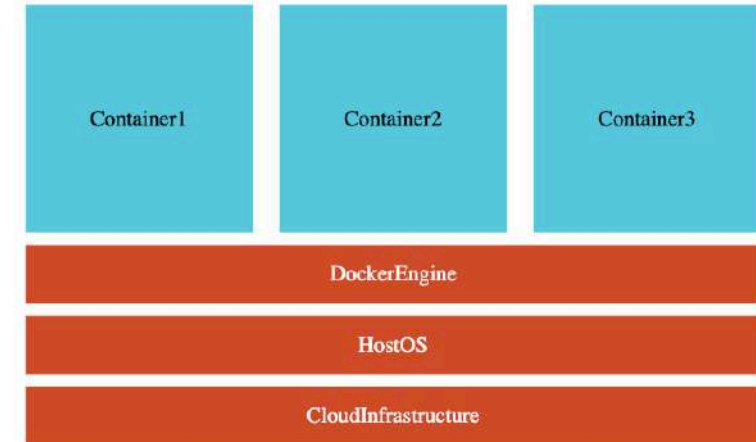
Microservices



- Enterprises are heading towards microservices architectures
 - Build small focused microservices
 - **Flexibility to innovate** and build applications in different programming languages (Go, Java, Python, JavaScript, etc)
- **BUT deployments become complex!**
- How can we have **one way of deploying** Go, Java, Python or JavaScript .. microservices?
 - Enter **containers!**

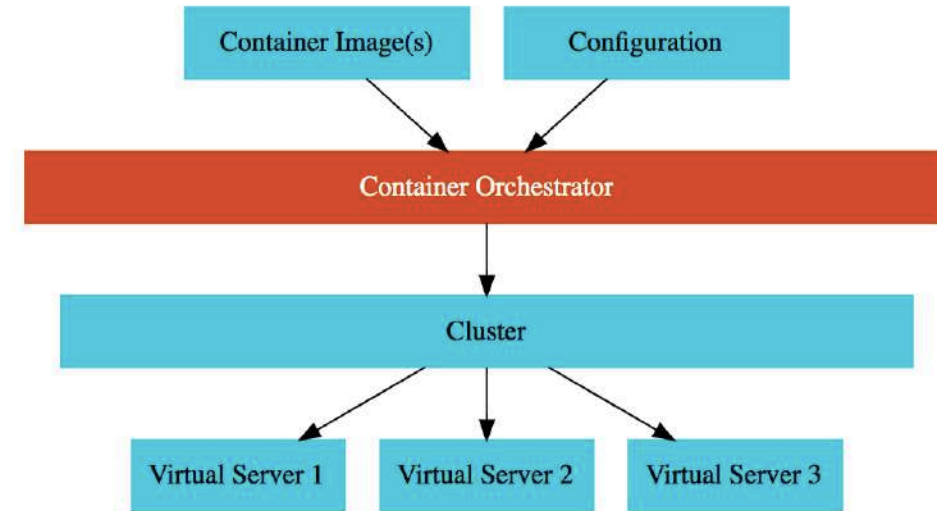
Containers - Docker

- Create **Docker images** for each microservice
- Docker image **has all needs of a microservice**:
 - Application Runtime (JDK or Python or NodeJS)
 - Application code and Dependencies
- Runs **the same way** on any infrastructure:
 - Your local machine
 - Corporate data center
 - Cloud
- Advantages
 - Docker containers are **light weight**
 - Compared to Virtual Machines as they do not have a Guest OS
 - Docker provides **isolation** for containers
 - Docker is **cloud neutral**



Container Orchestration

- **Requirement** : I want 10 instances of Microservice A container, 15 instances of Microservice B container and
- **Typical Features**:
 - **Auto Scaling** - Scale containers based on demand
 - **Service Discovery** - Help microservices find one another
 - **Load Balancer** - Distribute load among multiple instances of a microservice
 - **Self Healing** - Do health checks and replace failing instances
 - **Zero Downtime Deployments** - Release new versions without downtime



Serverless

- What do we think about when we develop an application?
 - Where to deploy? What kind of server? What OS?
 - How do we take care of scaling and availability of the application?
- **What if you don't need to worry about servers and focus on your code?**
 - Enter **Serverless**
 - Remember: **Serverless does NOT mean "No Servers"**
- **Serverless for me:**
 - You **don't worry** about infrastructure (ZERO visibility into infrastructure)
 - Flexible scaling and automated high availability
 - Most Important: **Pay for use**
 - Ideally ZERO REQUESTS => ZERO COST
- **You focus on code** and the cloud managed service takes care of all that is needed to scale your code to serve millions of requests!
 - And you pay for requests and NOT servers!

SaaS (Software as a Service)

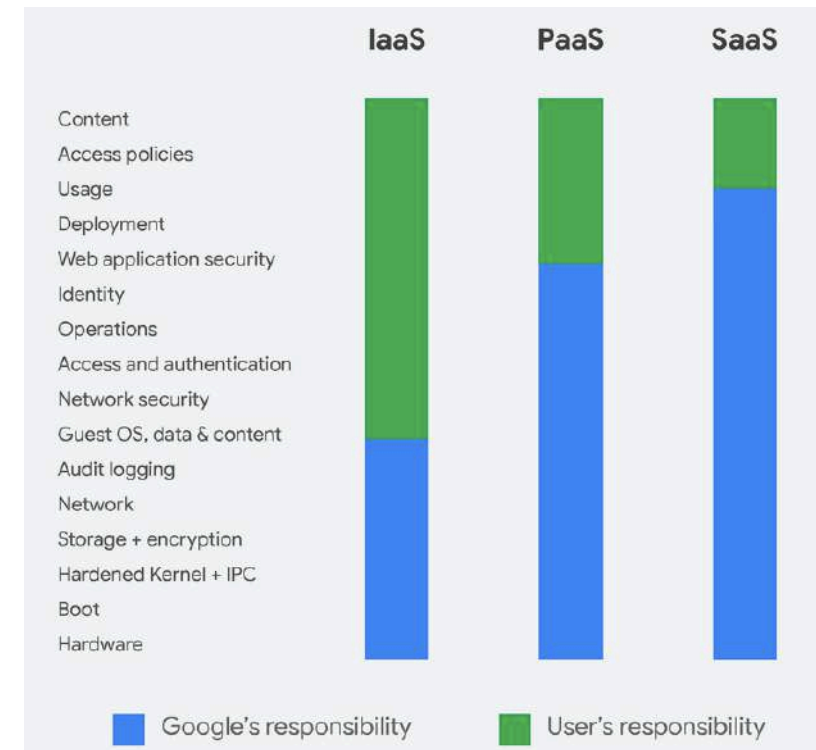
- **Centrally hosted software** (mostly on the cloud)
 - Offered on a subscription basis (pay-as-you-go)
 - Examples:
 - Email, calendaring & office tools (such as Outlook 365, Microsoft Office 365, Gmail, Google Docs)
- **Cloud provider** is responsible for:
 - OS (incl. upgrades and patches)
 - Application Runtime
 - Auto scaling, Availability & Load balancing etc..
 - Application code and/or
 - Application Configuration (How much memory? How many instances? ..)
- **Customer** is responsible for:
 - Configuring the software!
 - And the content (example: docs, sheets etc)



Google Cloud





Shared Responsibility Model

- Security in cloud is a **Shared Responsibility**:
 - Between GCP and the Customer
- GCP provides features to make security easy:
 - Encryption at rest by default
 - IAM
 - KMS etc
- Customer responsibilities vary with the model:
 - **SaaS**: Content + Access Policies + Usage
 - **PaaS**: SaaS + Deployment + Web Application Security
 - **IaaS**: PaaS + Operations + Network Security + Guest OS
- Google Cloud is always responsible for Hardware, Network, Audit Logging etc.



<https://cloud.google.com>

GCP Managed Services for Compute

| Service | Details | Category | |
|--------------------------|--|-------------------------|---|
| Compute Engine | High-performance and general purpose VMs that scale globally | IaaS |  Compute Engine |
| Google Kubernetes Engine | Orchestrate containerized microservices on Kubernetes Needs advanced cluster configuration and monitoring | CaaS |  Container Engine |
| App Engine | Build highly scalable applications on a fully managed platform using open and familiar languages and tools | PaaS (CaaS, Serverless) |  App Engine |
| Cloud Functions | Build event driven applications using simple, single-purpose functions | FaaS, Serverless |  Cloud Functions |
| Cloud Run | Develop and deploy highly scalable containerized applications. Does NOT need a cluster! | CaaS (Serverless) | |

Managed Compute Service in GCP

App Engine

- **Simplest way** to deploy and scale your applications in GCP
 - Provides end-to-end application management
- **Supports:**
 - Go, Java, .NET, Node.js, PHP, Python, Ruby using pre-configured runtimes
 - Use custom run-time and write code in any language
 - Connect to variety of Google Cloud storage products (Cloud SQL etc)
- **No usage charges** - Pay for resources provisioned
- **Features:**
 - Automatic load balancing & Auto scaling
 - Managed platform updates & Application health monitoring
 - Application versioning
 - Traffic splitting



Compute Engine vs App Engine

- **Compute Engine**

- IAAS
- MORE Flexibility
- MORE Responsibility
 - Choosing Image
 - Installing Software
 - Choosing Hardware
 - Fine grained Access/Permissions (Certificates/Firewalls)
 - Availability etc

- **App Engine**

- PaaS
- Serverless
- LESSER Responsibility
- LOWER Flexibility



App
Engine



Compute
Engine

App Engine environments

- **Standard:** Applications run in language specific sandboxes
 - **V1:** Java, Python, PHP, Go (OLD Versions)
 - **V2:** Java, Python, PHP, Node.js, Ruby, Go (NEWER Versions)
 - Complete isolation from OS/Disk
 - Supports scale down to Zero instances
- **Flexible** - Application instances run within Docker containers
 - Makes use of Compute Engine virtual machines
 - Support ANY runtime (with built-in support for Python, Java, Node.js, Go, Ruby, PHP, or .NET)
 - CANNOT scale down to Zero instances

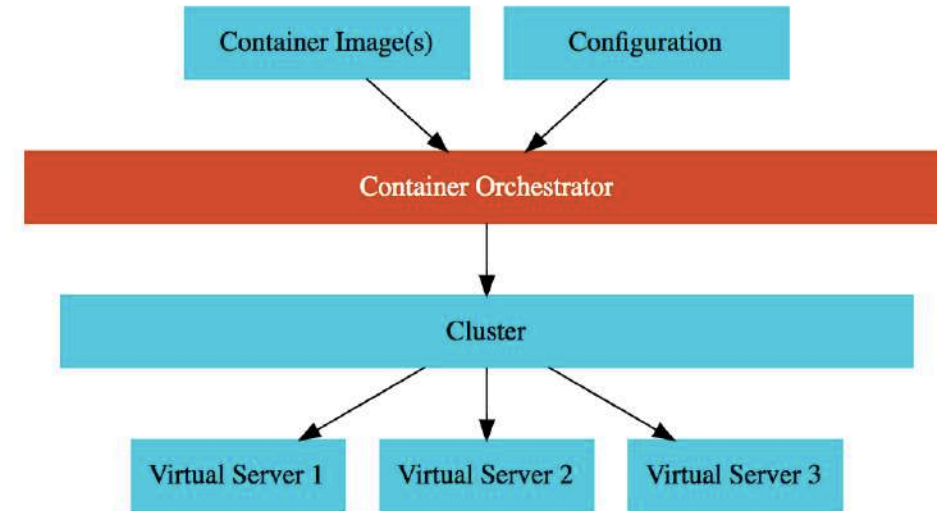


Service Categories - Scenarios

| Scenario | Solution |
|--|----------|
| IaaS or PaaS or SaaS: Deploy Custom Application in Virtual Machines | IaaS |
| IaaS or PaaS or SaaS: Using Gmail | SaaS |
| IaaS or PaaS or SaaS: Using App Engine to deploy your app | PaaS |
| True or False: Customer is responsible for OS updates when using PaaS | False |
| True or False: In PaaS, customer can configure auto scaling needs | True |
| True or False: Customer is completely responsible for Availability when using PaaS | False |
| True or False: In PaaS, customer has access to VM instances | False |
| True or False: In PaaS, customer can install custom software | False |
| True or False: PaaS services only offer Compute services | False |

Kubernetes

- Most popular open source container orchestration solution
- Provides Cluster Management (including upgrades)
 - Each cluster can have different types of virtual machines
- Provides all important container orchestration features:
 - Auto Scaling
 - Service Discovery
 - Load Balancer
 - Self Healing
 - Zero Downtime Deployments



Google Kubernetes Engine (GKE)

- **Managed** Kubernetes service
- Minimize operations with **auto-repair** (repair failed nodes) and **auto-upgrade** (use latest version of K8S always) features
- Provides **Pod and Cluster Autoscaling**
- Enable **Cloud Logging** and **Cloud Monitoring** with simple configuration
- Uses **Container-Optimized OS**, a hardened OS built by Google
- Provides support for **Persistent disks** and **Local SSD**



Kubernetes Engine

Kubernetes - A Microservice Journey - Getting Started



Kubernetes Engine

- **Let's Have Some Fun:** Let's get on a journey with Kubernetes:
 - Let's create a cluster, deploy a microservice and play with it in **13 steps!**
- **1:** Create a Kubernetes cluster with the default node pool
 - `gcloud container clusters create` or use cloud console
- **2:** Login to Cloud Shell
- **3:** Connect to the Kubernetes Cluster
 - `gcloud container clusters get-credentials my-cluster --zone us-central1-a --project solid-course-258105`

Kubernetes - A Microservice Journey - Deploy Microservice

In **28**
Minutes

- 4: Deploy Microservice to Kubernetes:
 - Create deployment & service using kubectl commands
 - `kubectl create deployment hello-world-rest-api --image=in28min/hello-world-rest-api:0.0.1.RELEASE`
 - `kubectl expose deployment hello-world-rest-api --type=LoadBalancer --port=8080`
- 5: Increase number of instances of your microservice:
 - `kubectl scale deployment hello-world-rest-api --replicas=2`
- 6: Increase number of nodes in your Kubernetes cluster:
 - `gcloud container clusters resize my-cluster --node-pool my-node-pool --num-nodes 5`
 - You are NOT happy about manually increasing number of instances and nodes!



Kubernetes Engine

Kubernetes - A Microservice Journey - Auto Scaling and ..

In 28
Minutes



Kubernetes Engine

- 7: Setup auto scaling for your microservice:
 - `kubectl autoscale deployment hello-world-rest-api --max=10 --cpu-percent=70`
 - Also called horizontal pod autoscaling - HPA - `kubectl get hpa`
- 8: Setup auto scaling for your Kubernetes Cluster
 - `gcloud container clusters update cluster-name --enable-autoscaling --min-nodes=1 --max-nodes=10`
- 9: Delete the Microservice
 - Delete service - `kubectl delete service`
 - Delete deployment - `kubectl delete deployment`
- 10: Delete the Cluster
 - `gcloud container clusters delete`

Cloud Functions



- Imagine you want to **execute some code when an event happens?**
 - A file is uploaded in Cloud Storage
 - An error log is written to Cloud Logging
 - A message arrives to Cloud Pub/Sub
- Enter **Cloud Functions**
 - **Run code in response to events**
 - Write your business logic in Node.js, Python, Go, Java, .NET, and Ruby
 - **Don't worry** about servers or scaling or availability (only worry about your code)
 - **Pay only for what you use**
 - Number of invocations
 - Compute Time of the invocations
 - Amount of memory and CPU provisioned
 - **Time Bound** - Default 1 min and MAX 60 minutes(3600 seconds)
 - **Each execution runs in a separate instance**
 - No direct sharing between invocations

Cloud Run & Cloud Run for Anthos



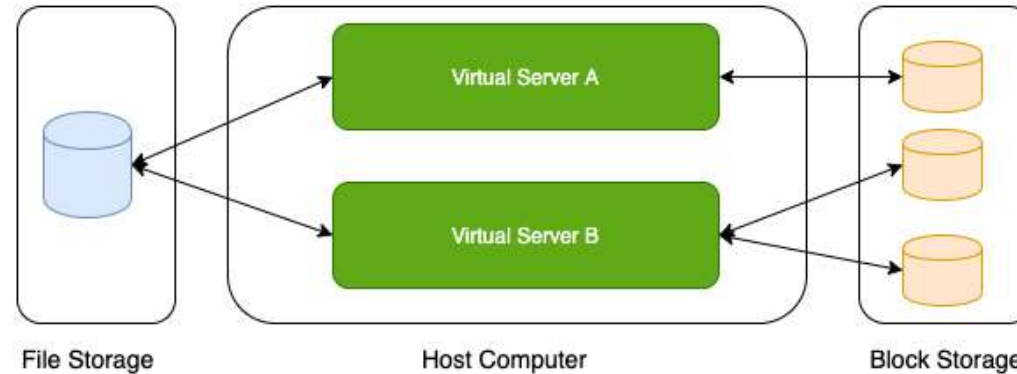
- **Cloud Run - "Container to Production in Seconds"**
 - Built on top of an open standard - **Knative**
 - **Fully managed** serverless platform for containerized applications
 - ZERO infrastructure management
 - Pay-per-use (For used CPU, Memory, Requests and Networking)
- Fully integrated **end-to-end developer experience**:
 - **No limitations** in languages, binaries and dependencies
 - Easily portable because of **container** based architecture
 - Cloud Code, Cloud Build, Cloud Monitoring & Cloud Logging Integrations
- **Anthos** - Run Kubernetes clusters anywhere
 - Cloud, Multi Cloud and On-Premise
- **Cloud Run for Anthos**: Deploy your workloads to Anthos clusters running on-premises or on Google Cloud
 - Leverage your existing Kubernetes investment to quickly run serverless workloads

Scenarios - GCP Compute Services

| Scenario | GCP |
|---|--------------------------------|
| How do you create Virtual Machines? | Compute Engine |
| How do you create a group of similar VMs? | MIG |
| How do distribute load among VMs? | Cloud Load Balancing |
| How do you simplify setting up your web applications? | App Engine |
| What is the easiest way to run one container? | Google Cloud Run |
| How do you orchestrate containers? | Google Kubernetes Engine (GKE) |
| How do you build serverless event driven functions? | Cloud Functions |
| How can you centrally manage multi-cloud and on-premise Kubernetes clusters ? | Anthos |

Storage

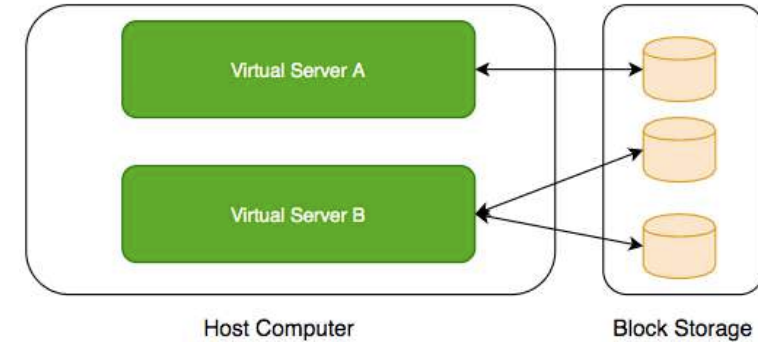
Storage Types - Block Storage and File Storage



- What is the type of storage of your hard disk?
 - Block Storage
- You've created a file share to share a set of files with your colleagues in a enterprise. What type of storage are you using?
 - File Storage

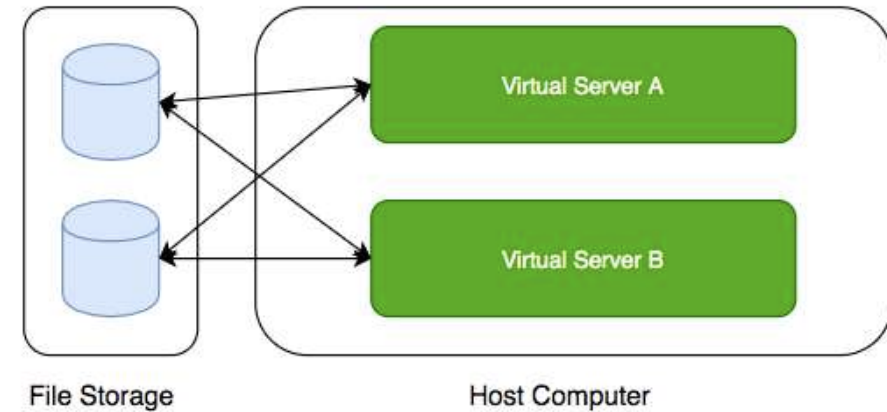
Block Storage

- Use case: Harddisks attached to your computers
- Typically, ONE Block Storage device can be connected to ONE virtual server
 - (EXCEPTIONS) You can attach read only block devices with multiple virtual servers and certain cloud providers are exploring multi-writer disks as well!
- HOWEVER, you can connect multiple different block storage devices to one virtual server
- Used as:
 - **Direct-attached storage (DAS)** - Similar to a hard disk
 - **Storage Area Network (SAN)** - High-speed network connecting a pool of storage devices
 - Used by Databases, Oracle and Microsoft SQL Server



File Storage

- Media workflows need huge shared storage for supporting processes like video editing
- Enterprise users need a quick way to share files in a secure and organized way
- These file shares are shared by several virtual servers



GCP - Block Storage and File Storage



Persistent
Disk

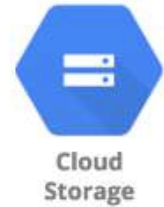


Filestore

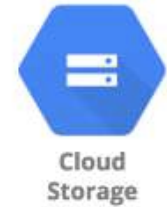
- **Block Storage:**
 - **Persistent Disks:** Network Block Storage
 - Zonal: Data replicated in one zone
 - Regional: Data replicated in multiple zone
 - **Local SSDs:** Local Block Storage
- **File Storage:**
 - **Filestore:** High performance file storage

Cloud Storage

- **Most popular, very flexible & inexpensive** storage service
 - Serverless: Autoscaling and infinite scale
- Store large objects using a **key-value** approach:
 - Treats entire object as a unit (Partial updates not allowed)
 - Recommended when you operate on entire object most of the time
 - Access Control at Object level
 - Also called **Object Storage**
- Provides REST API to access and modify objects
 - Also provides CLI (gsutil) & Client Libraries (C++, C#, Java, Node.js, PHP, Python & Ruby)
- **Store all file types** - text, binary, backup & archives:
 - Media files and archives, Application packages and logs
 - Backups of your databases or storage devices
 - Staging data during on-premise to cloud database migration

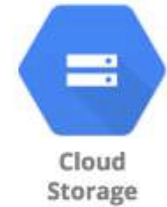


Cloud Storage - Objects and Buckets



- Objects are stored in buckets
 - Bucket names are **globally unique**
 - Bucket names are used as part of object URLs => Can contain ONLY lower case letters, numbers, hyphens, underscores and periods.
 - 3-63 characters max. Can't start with **goog prefix** or should not contain **google (even misspelled)**
 - Unlimited objects in a bucket
 - Each bucket is associated with a project
- Each object is identified by a **unique key**
 - **Key is unique** in a bucket
- Max object size is **5 TB**
 - BUT you can store unlimited number of such objects

Cloud Storage - Storage Classes - Introduction



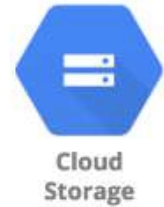
- **Different kinds of data** can be stored in Cloud Storage
 - Media files and archives
 - Application packages and logs
 - Backups of your databases or storage devices
 - Long term archives
- Huge variations in **access patterns**
- Can I pay a cheaper price for objects I access less frequently?
- **Storage classes** help to optimize your costs based on your access needs
 - Designed for durability of 99.999999999%(11 9's)

Cloud Storage - Storage Classes - Comparison

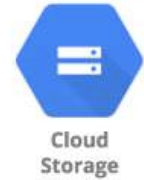
| Storage Class | Name | Minimum Storage duration | Typical Monthly availability | Use case |
|------------------|----------|--------------------------|---|---|
| Standard | STANDARD | None | > 99.99% in multi region and dual region, 99.99% in regions | Frequently used data/Short period of time |
| Nearline storage | NEARLINE | 30 days | 99.95% in multi region and dual region, 99.9% in regions | Read or modify once a month on average |
| Coldline storage | COLDLINE | 90 days | 99.95% in multi region and dual region, 99.9% in regions | Read or modify at most once a quarter |
| Archive storage | ARCHIVE | 365 days | 99.95% in multi region and dual region, 99.9% in regions | Less than once a year |

Features across Storage Classes

- High durability (99.999999999% annual durability)
- **Low** latency (first byte typically in tens of milliseconds)
- **Unlimited** storage
 - Autoscaling (No configuration needed)
 - **NO** minimum object size
- Same APIs across storage classes
- **Committed SLA** is 99.95% for multi region and 99.9% for single region for Standard, Nearline and Coldline storage classes
 - No committed SLA for Archive storage



Object Lifecycle Management



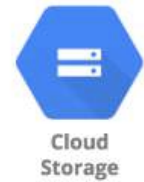
- Files are frequently accessed when they are created
 - Generally **usage reduces with time**
 - How do you save costs by **moving files automatically between storage classes**?
 - Solution: Object Lifecycle Management
- Identify objects using conditions based on:
 - Age, CreatedBefore, IsLive, MatchesStorageClass, NumberOfNewerVersions etc
 - Set multiple conditions: all conditions must be satisfied for action to happen
- Two kinds of actions:
 - **SetStorageClass** actions (change from one storage class to another)
 - **Deletion** actions (delete objects)
- Allowed Transitions:
 - (Standard or Multi-Regional or Regional) to (Nearline or Coldline or Archive)
 - Nearline to (Coldline or Archive)
 - Coldline to Archive

Object Lifecycle Management - Example Rule

```
{
  "lifecycle": {
    "rule": [
      {
        "action": {"type": "Delete"},
        "condition": {
          "age": 30,
          "isLive": true
        }
      },
      {
        "action": {
          "type": "SetStorageClass",
          "storageClass": "NEARLINE"
        },
        "condition": {
          "age": 365,
          "matchesStorageClass": ["STANDARD"]
        }
      }
    ]
  }
}
```

Transferring data from on premises to cloud

- Most popular data destination is **Google Cloud Storage**
- Options:
 - **Online Transfer:** Use gsutil or API to transfer data to Google Cloud Storage
 - Good for one time transfers
 - **Storage Transfer Service:** Recommended for large-scale (petabytes) online data transfers from your private data centers, AWS, Azure, and Google Cloud
 - You can set up a repeating schedule
 - Supports incremental transfer (only transfer changed objects)
 - Reliable and fault tolerant - continues from where it left off in case of errors
 - **Storage Transfer Service vs gsutil:**
 - gsutil is recommended only when you are transferring less than 1 TB from on-premises or another GCS bucket
 - Storage Transfer Service is recommended if either of the conditions is met:
 - Transferring more than 1 TB from anywhere
 - Transferring from another cloud
 - **Transfer Appliance:** Physical transfer using an appliance



Migrating Data with Transfer Appliance

- **Transfer Appliance:** Copy, ship and upload data to GCS
 - **Recommended** if your data size is **greater than 20TB**
 - OR online transfer takes > 1 week
 - **Process:**
 - Request an appliance
 - Upload your data
 - Ship the appliance back
 - Google uploads the data
 - **Fast copy** (upto 40Gbps)
 - **AES 256 encryption** - Customer-managed encryption keys
 - Order **multiple devices** (TA40, TA300) if need

| | Physical Transfer | | | Physical / Online Transfer | | Online Transfer |
|--------|-------------------|-------------|------------|----------------------------|------------|-----------------|
| | 1 Mbps | 10 Mbps | 100 Mbps | 1 Gbps | 10 Gbps | 100 Gbps |
| 1 GB | 3 hours | 18 minutes | 2 minutes | 11 seconds | 1 second | 0.1 seconds |
| 10 GB | 30 hours | 3 hours | 18 minutes | 2 minutes | 11 seconds | 1 second |
| 100 GB | 12 days | 30 hours | 3 hours | 18 minutes | 2 minutes | 11 seconds |
| 1 TB | 124 days | 12 days | 30 hours | 3 hours | 18 minutes | 2 minutes |
| 10 TB | 3 years | 124 days | 12 days | 30 hours | 3 hours | 18 minutes |
| 100 TB | 34 years | 3 years | 124 days | 12 days | 30 hours | 3 hours |
| 1 PB | 340 years | 34 years | 3 years | 124 days | 12 days | 30 hours |
| 10 PB | 3,404 years | 340 years | 34 years | 3 years | 124 days | 12 days |
| 100 PB | 34,048 years | 3,404 years | 340 years | 34 years | 3 years | 124 days |

<https://cloud.google.com>

Storage in Google Cloud - Scenarios

| Scenario | Service |
|---|---|
| My team requires a shared space for collaborating on media projects that involve large files | Filestore (File Storage) |
| I'm looking for a cost-effective solution to store and serve a large amount of unstructured data (Videos, Music, Files) globally | Cloud Storage (Object Storage) |
| I want to ensure that my data is automatically managed and transitioned between storage classes to reduce costs without manual intervention | Object Lifecycle Management in Cloud Storage |
| For a massive, one-time migration of data to the cloud, where online transfer is not feasible due to size and time constraints | Using Transfer Appliance for large-scale, physical data migration |

Database Fundamentals

Database Categories

- There are **several categories** of databases:
 - Relational (OLTP and OLAP), Document, Key Value, Graph, In Memory among others
- **Choosing type of database** for your use case is not easy. A few factors:
 - Do you want a **fixed schema**?
 - Do you want flexibility in defining and changing your schema? (schemaless)
 - What level of **transaction properties** do you need? (atomicity and consistency)
 - What kind of **latency** do you want? (seconds, milliseconds or microseconds)
 - **How many transactions** do you expect? (hundreds or thousands or millions of transactions per second)
 - **How much data** will be stored? (MBs or GBs or TBs or PBs)
 - and a lot more...



Cloud SQL



Cloud Spanner



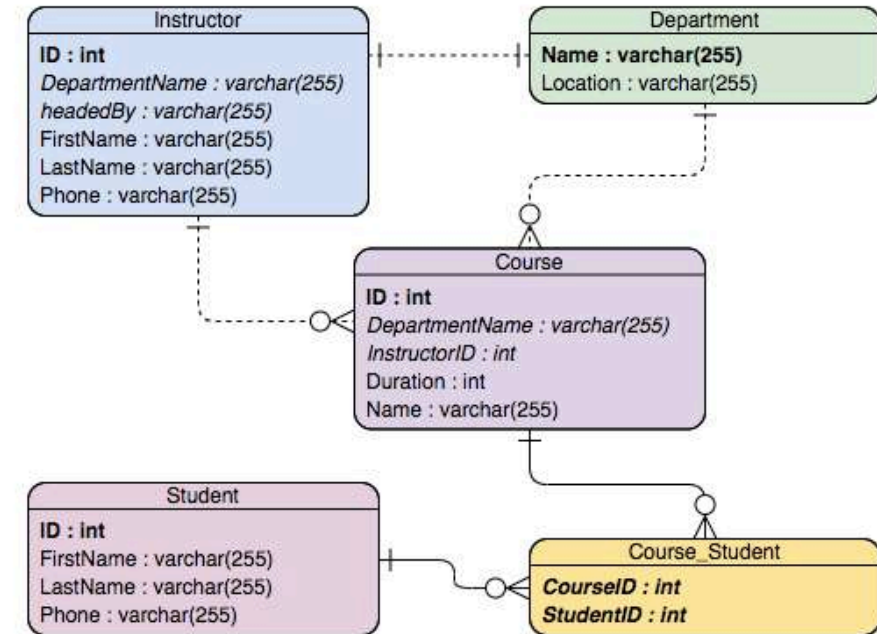
Cloud Datastore



BigQuery

Relational Databases

- This was the **only option** until a decade back!
- Most **popular (or unpopular)** type of databases
- **Predefined schema** with tables and relationships
- **Very strong transactional capabilities**
- Used for
 - OLTP (Online Transaction Processing) use cases and
 - OLAP (Online Analytics Processing) use cases



Relational Database - OLTP (Online Transaction Processing)

In 28
Minutes

- Applications where large number of users make large number of small transactions
 - small data reads, updates and deletes
- **Use cases:**
 - Most traditional applications, ERP, CRM, e-commerce, banking applications
- **Popular databases:**
 - MySQL, Oracle, SQL Server etc
- **Recommended Google Managed Services:**
 - **Cloud SQL** : Supports PostgreSQL, MySQL, and SQL Server for regional relational databases (upto a few TBs)
 - **Cloud Spanner**: Unlimited scale (multiple PBs) and 99.999% availability for global applications with horizontal scaling



Cloud SQL



Cloud Spanner

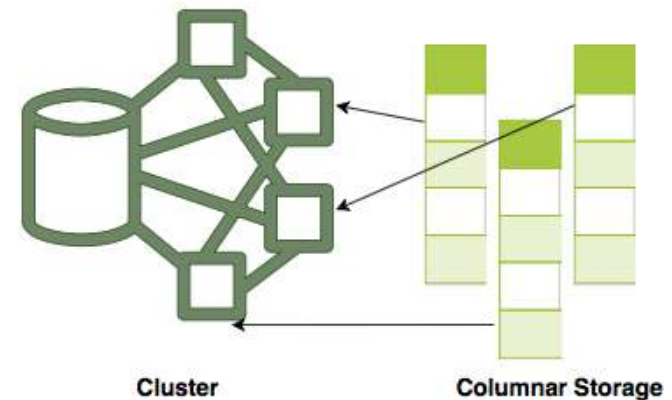
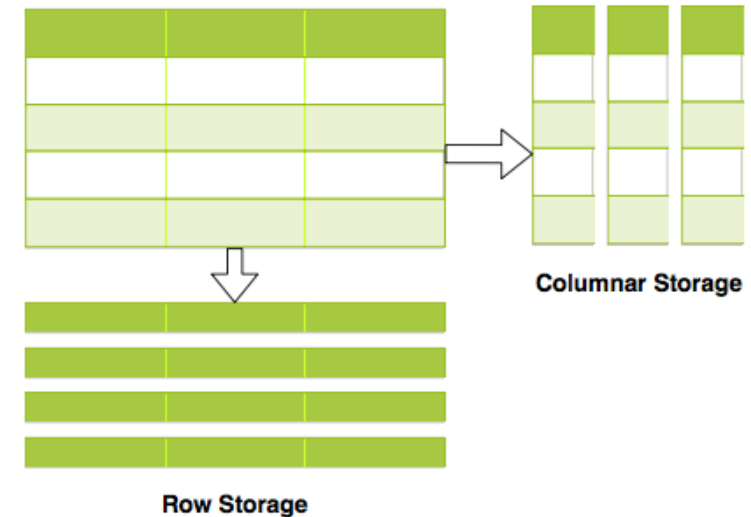
Relational Database - OLAP (Online Analytics Processing)

- Applications allowing users to **analyze petabytes of data**
 - **Examples** : Reporting applications, Data ware houses, Business intelligence applications, Analytics systems
 - **Sample application** : Decide insurance premiums analyzing data from last hundred years
 - Data is consolidated from multiple (transactional) databases
- Recommended GCP Managed Service
 - **BigQuery**: Petabyte-scale distributed data ware house



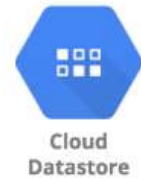
Relational Databases - OLAP vs OLTP

- OLAP and OLTP use similar data structures
- BUT very different approach in how data is stored
- **OLTP databases** use row storage
 - Each table row is stored together
 - Efficient for processing small transactions
- **OLAP databases** use columnar storage
 - Each table column is stored together
 - **High compression** - store petabytes of data efficiently
 - **Distribute data** - one table in multiple cluster nodes
 - **Execute single query across multiple nodes** - Complex queries can be executed efficiently

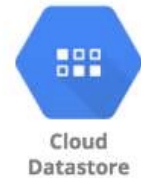


NoSQL Databases

- **New approach** (actually NOT so new!) to building your databases
 - NoSQL = not only SQL
 - Flexible schema
 - Structure data **the way your application needs it**
 - Let the schema evolve with time
 - Horizontally scale to petabytes of data with millions of TPS
- **NOT a 100% accurate generalization** but a great starting point:
 - Typical NoSQL databases trade-off "Strong consistency and SQL features" to achieve "scalability and high-performance"
- **Google Managed Services:**
 - Cloud Firestore (Datastore)
 - Cloud BigTable



Cloud Firestore (Datastore) vs Cloud BigTable



- **Cloud Datastore** - Managed serverless NoSQL document database
 - Provides ACID transactions, SQL-like queries, indexes
 - Designed for transactional mobile and web applications
 - Firestore (next version of Datastore) adds:
 - Strong consistency
 - Mobile and Web client libraries
 - Recommended for small to medium databases (0 to a few Terabytes)
- **Cloud BigTable** - Managed, scalable NoSQL wide column database
 - NOT serverless (You need to create instances)
 - Recommend for data size > 10 Terabytes to several Petabytes
 - Recommended for large analytical and operational workloads:
 - NOT recommended for transactional workloads (Does NOT support multi row transactions - supports ONLY Single-row transactions)

In-memory Databases

- Retrieving data from memory is much faster than retrieving data from disk
- In-memory databases like Redis deliver microsecond latency by storing **persistent data in memory**
- Recommended GCP Managed Service
 - Memory Store
- **Use cases** : Caching, session management, gaming leader boards, geospatial applications



Memorystore

Databases - Summary

| Database Type | GCP Services | Description |
|---------------------------|--|--|
| Relational OLTP databases | Cloud SQL, Cloud Spanner | Transactional usecases needing predefined schema and very strong transactional capabilities (Row storage) Cloud SQL : MySQL, PostgreSQL, SQL server DBs Cloud Spanner : Unlimited scale and 99.999% availability for global applications with horizontal scaling |
| Relational OLAP databases | BigQuery | Columnar storage with predefined schema. Datawarehousing & BigData workloads |
| NoSQL Databases | Cloud Firestore (Datastore) , Cloud BigTable | Apps needing quickly evolving structure (schema-less) Cloud Firestore - Serverless transactional document DB supporting mobile & web apps. Small to medium DBs (0 - few TBs) Cloud BigTable - Large databases(10 TB - PBs). Streaming (IOT), analytical & operational workloads. NOT serverless. |
| In memory | Cloud Memorystore | Applications needing microsecond responses |

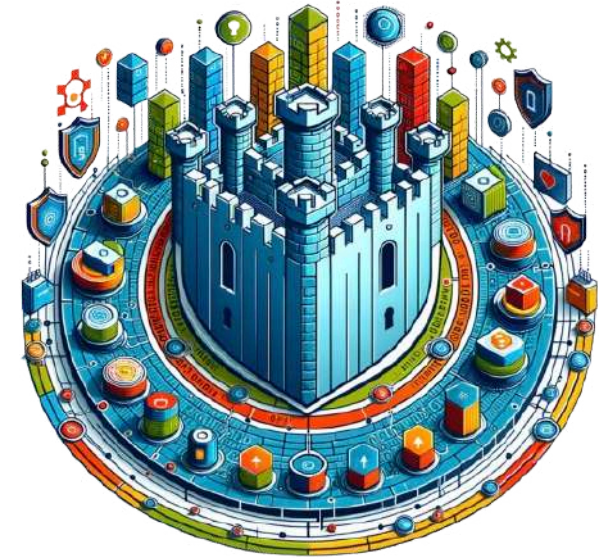
Databases - Scenarios

| Scenario | Solution |
|--|---------------------------|
| A start up with quickly evolving schema (table structure) | Cloud Datastore/Firestore |
| Non relational db with less storage (10 GB) | Cloud Datastore |
| Transactional global database with predefined schema needing to process million of transactions per second | Cloud Spanner |
| Transactional local database processing thousands of transactions per second | Cloud SQL |
| Cache data (from database) for a web application | MemoryStore |
| Database for analytics processing of petabytes of data | BigQuery |
| Database for storing huge volumes stream data from IOT devices | BigTable |
| Database for storing huge streams of time series data | BigTable |

IAM

Security Today is Complex

- **Various Threats:** Security is important but complex:
 - **Ransomware:** Lock up your company's files and demand money to unlock them
 - **Phishing Scams:** Trick emails that look real but aim to steal your information
 - **Insider Threats:** Disgruntled employees might take or leak confidential info, hurting your business from the inside.
 - **Malware Attacks:** Harmful software that sneaks into your systems to steal data or cause damage
 - **DDoS Attacks:** Attack your website with traffic until it crashes
 - **Data Breaches:** When someone unauthorized gets into your systems and steals sensitive information
 - **Cloud Vulnerabilities:** Weaknesses in cloud services configuration can expose your data



Security Today - Scenarios

| Scenario | Threat |
|--|--------------------|
| An employee opens an invoice attached to an email that seems to come from a known vendor. This action installs software that encrypts all the data on their computer, and a message appears demanding payment to unlock the files. | Ransomware |
| You receive an email that looks like it's from your bank, asking you to update your login details via a link. The link leads to a fake website that collects your username and password when you try to log in. | Phishing Scams |
| A former employee, still holding grudges, uses their still-active login credentials to access and download customer data, which they then leak online. | Insider Threats |
| While browsing the internet, an employee clicks on a seemingly harmless link, which downloads a program onto their computer without their knowledge. This program starts sending sensitive information to a cybercriminal. | Malware Attacks |
| Your company's website suddenly becomes unreachable. Investigation shows that it's receiving millions of requests per minute from a coordinated network of compromised computers, overwhelming the server. | DDoS Attacks |

Typical identity management in the cloud

- You have **resources** in the cloud (examples - a virtual server, a database etc)
- You have **identities (human and non-human)** that need to access those resources and perform actions
 - For example: launch (stop, start or terminate) a virtual server
- How do you **identify users** in the cloud?
 - How do you configure resources they can access?
 - How can you configure what actions to allow?
- In GCP: *Identity and Access Management (Cloud IAM)* provides this service



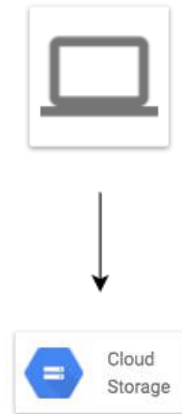
Cloud Identity and Access Management (IAM)



- **Authentication** (is it the right user?) and
- **Authorization** (do they have the right access?)
- **Identities** can be
 - A GCP User (Google Account or Externally Authenticated User)
 - A Group of GCP Users
 - An Application running in GCP
 - An Application running in your data center
 - Unauthenticated users
- Provides very **granular** control
 - Limit a single user:
 - to perform single action
 - on a specific cloud resource
 - from a specific IP address
 - during a specific time window

Cloud IAM Example

- I want to provide access to manage a specific cloud storage bucket to a colleague of mine:
 - Important Generic Concepts:
 - **Member:** My colleague
 - **Resource:** Specific cloud storage bucket
 - **Action:** Upload/Delete Objects
 - In Google Cloud IAM:
 - **Roles:** A set of permissions (to perform specific actions on specific resources)
 - Roles do NOT know about members. It is all about permissions!
 - How do you assign permissions to a member?
 - **Policy:** You assign (or **bind**) a role to a member
- 1: **Choose a Role** with right permissions (Ex: Storage Object Admin)
- 2: **Create Policy** binding member (your friend) with role (permissions)
- IAM in AWS is very different from GCP (Forget AWS IAM & Start FRESH!)
 - Example: Role in AWS is NOT the same as Role in GCP



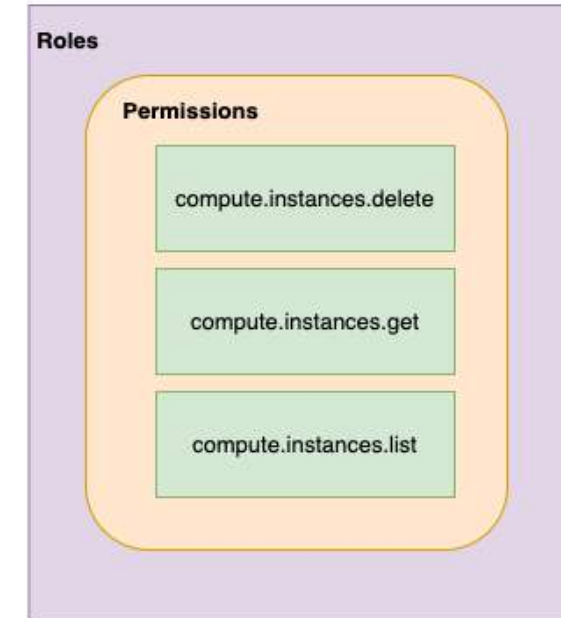
IAM - Roles



- **Roles are Permissions:**
 - Perform some set of actions on some set of resources
- **Three Types:**
 - **Basic Roles (or Primitive roles)** - Owner/Editor/Viewer
 - **Viewer(roles.viewer)** - Read-only actions
 - **Editor(roles.editor)** - Viewer + Edit actions
 - **Owner(roles.owner)** - Editor + Manage Roles and Permissions + Billing
 - EARLIEST VERSION: Created before IAM
 - NOT RECOMMENDED: **Don't use in production**
 - **Predefined Roles** - Fine grained roles predefined and managed by Google
 - Different roles for different purposes
 - **Examples:** Storage Admin, Storage Object Admin, Storage Object Viewer, Storage Object Creator
 - **Custom Roles** - When predefined roles are NOT sufficient, you can create your own custom roles

IAM - Most Important Concepts - A Review

- **Member** : Who?
- **Roles** : Permissions (What Actions? What Resources?)
- **Policy** : Assign Permissions to Members
 - Map Roles (What?) , Members (Who?) and Conditions (Which Resources?, When?, From Where?)
 - Remember: Permissions are NOT directly assigned to Member
 - Permissions are represented by a Role
 - Member gets permissions through Role!
- A Role can have multiple permissions
- You can assign multiple roles to a Member



IAM policy

- Roles are assigned to users through **IAM Policy** documents
- Represented by a **policy object**
 - Policy object has list of bindings
 - A binding, binds a role to list of members
- Member type is identified by **prefix**:
 - Example: user, serviceaccount, group or domain



IAM policy - Example

```
{
  "bindings": [
    {
      "role": "roles/storage.objectAdmin",
      "members": [
        "user:you@in28minutes.com",
        "serviceAccount:myAppName@appspot.gserviceaccount.com",
        "group:administrators@in28minutes.com",
        "domain:google.com"
      ]
    },
    {
      "role": "roles/storage.objectViewer",
      "members": [
        "user:you@in28minutes.com"
      ],
      "condition": {
        "title": "Limited time access",
        "description": "Only upto Feb 2022",
        "expression": "request.time < timestamp('2022-02-01T00:00:00.000Z')",
      }
    }
  ]
}
```

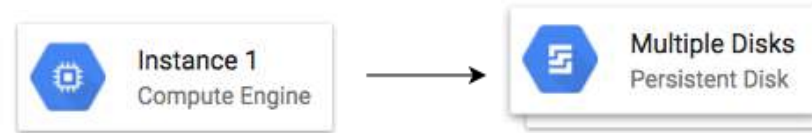
Service Accounts

- Scenario: An Application on a VM needs access to cloud storage
 - You DONT want to use personal credentials to allow access
- (RECOMMENDED) Use **Service Accounts**
 - Identified by an email address (Ex: id-compute@developer.gserviceaccount.com)
 - Does NOT have password
 - Has a **private/public RSA key-pairs**
 - Can't login via browsers or cookies
- Service account types:
 - **Default service account** - Automatically created when some services are used
 - (NOT RECOMMENDED) Has **Editor role** by default
 - **User Managed** - User created
 - (RECOMMENDED) Provides fine grained access control
 - **Google-managed service accounts** - Created and managed by Google
 - Used by GCP to perform operations on user's behalf
 - In general, we DO NOT need to worry about them



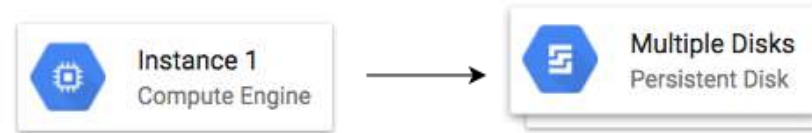
Encryption

Data States



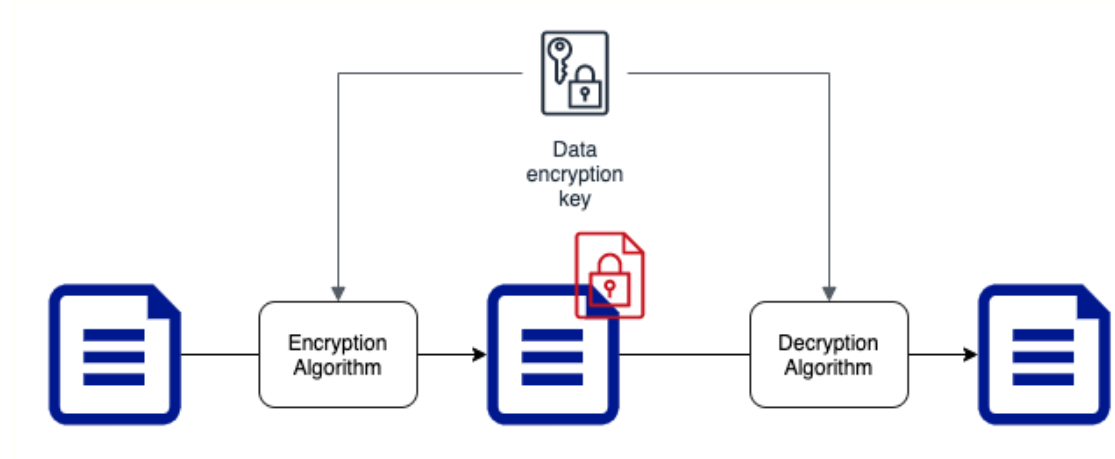
- **Data at rest:** Stored on a device or a backup
 - Examples : data on a hard disk, in a database, backups and archives
- **Data in motion:** Being transferred across a network
 - Also called **Data in transit**
 - **Examples :**
 - Data copied from on-premise to cloud storage
 - An application talking to a database
 - **Two Types:**
 - In and out of cloud (from internet)
 - Within cloud
- **Data in use:** Active data processed in a non-persistent state
 - Example: Data in your RAM

Encryption



- If you store data as is, what would happen if an **unauthorized entity** gets **access** to it?
 - Imagine losing an unencrypted hard disk
- **First law of security** : Defense in Depth
- Typically, enterprises encrypt all data
 - Data on your hard disks
 - Data in your databases
 - Data on your file servers
- Is it sufficient if you encrypt data at rest?
 - **No. Encrypt data in transit** - between application to database as well.

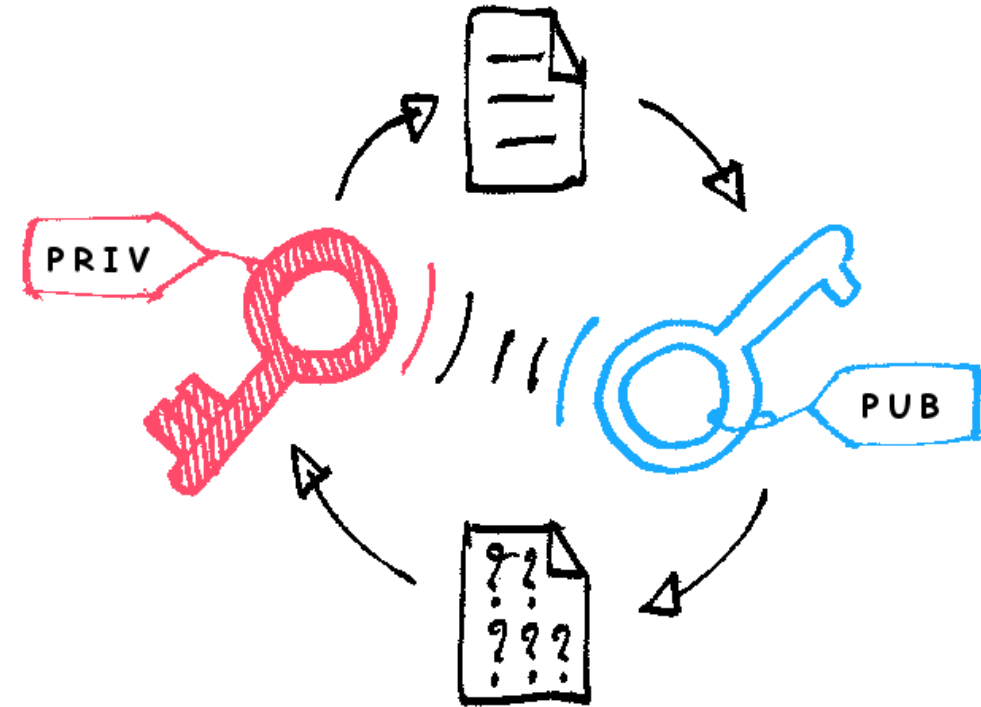
Symmetric Key Encryption



- Symmetric encryption algorithms use the **same key for encryption and decryption**
- Key Factor 1: Choose the **right encryption algorithm**
- Key Factor 2: How do we **secure the encryption key**?
- Key Factor 3: How do we **share the encryption key**?

Asymmetric Key Encryption

- **Two Keys** : Public Key and Private Key
- Also called **Public Key Cryptography**
- Encrypt data with Public Key and decrypt with Private Key
- Share Public Key with everybody and keep the Private Key with you(YEAH, ITS PRIVATE!)
- No crazy questions:
 - Will somebody not figure out private key using the public key?
- How do you create Asymmetric Keys?



[https://commons.wikimedia.org/wiki/File:Asymmetric_encryption_\(colored\).p](https://commons.wikimedia.org/wiki/File:Asymmetric_encryption_(colored).p)

Cloud KMS

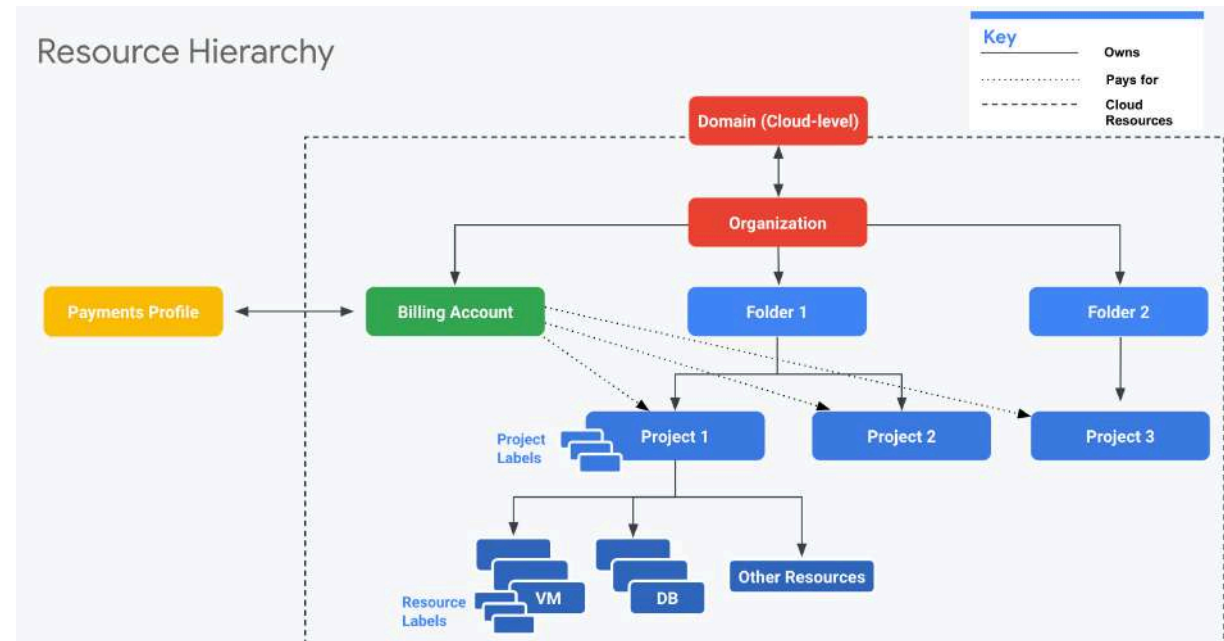
- Create and manage **cryptographic keys** (symmetric and asymmetric)
- **Control their use** in your applications and GCP Services
- Provides an API to encrypt, decrypt, or sign data
- Use existing cryptographic keys created on premises
- **Integrates with almost all GCP services** that need data encryption:
 - Google-managed key: No configuration required
 - Customer-managed key: Use key from KMS
 - Customer-supplied key: Provide your own key



Organizing GCP Resources

Resource Hierarchy in GCP

- **Well defined hierarchy:**
 - Organization > Folder > Project > Resources
- **Resources** are created in projects
- A **Folder** can contain multiple projects
- **Organization** can contain multiple Folders



source: (<https://cloud.google.com>)

Resource Hierarchy - Recommendations for Enterprises

- Create **separate projects for different environments**:
 - Complete isolation between test and production environments
- Create **separate folders for each department**:
 - Isolate production applications of one department from another
 - We can create a shared folder for shared resources
- **One project per application per environment**:
 - Let's consider two apps: "A1" and "A2"
 - Let's assume we need two environments: "DEV" and "PROD"
 - In the ideal world you will create four projects: A1-DEV, A1-PROD, A2-DEV, A2-PROD:
 - Isolates environments from each other
 - DEV changes will NOT break PROD
 - Grant all developers complete access (create, delete, deploy) to DEV Projects
 - Provide production access to operations teams only!

Billing Accounts

- **Billing Account** is mandatory for creating resources in a project:
 - Billing Account contains the payment details
 - Every Project with active resources should be associated with a Billing Account
- Billing Account can be associated with one or more projects
- You can have multiple billing accounts in an Organization
- (RECOMMENDATION) Create Billing Accounts representing your organization structure:
 - A startup can have just one Billing account
 - A large enterprise can have a separate billing account for each department
- Two Types:
 - **Self Serve** : Billed directly to Credit Card or Bank Account
 - **Invoiced** : Generate invoices (Used by large enterprises)

Managing Billing - Budget, Alerts and Exports

- Setup a **Cloud Billing Budget** to avoid surprises:
 - (RECOMMENDED) **Configure Alerts**
 - Default alert thresholds set at 50%, 90% & 100%
 - Send alerts to Pub Sub (Optional)
 - Billing admins and Billing Account users are alerted by e-mail
- Billing data can be **exported (on a schedule)** to:
 - **Big Query** (if you want to query information or visualize it)
 - **Cloud Storage** (for history/archiving)

IAM Best Practices

- **Principle of Least Privilege** - Give least possible privilege needed for a role!
 - Basic Roles are NOT recommended
 - Prefer predefined roles when possible
 - Use Service Accounts with minimum privileges
 - Use different Service Accounts for different apps/purposes
- **Separation of Duties** - Involve at least 2 people in sensitive tasks:
 - Example: Have separate deployer and traffic migrator roles
 - AppEngine provides App Engine Deployer and App Engine Service Admin roles
 - App Engine Deployer can deploy new version but cannot shift traffic
 - App Engine Service Admin can shift traffic but cannot deploy new version!
- **Constant Monitoring:** Review Cloud Audit Logs to audit changes to IAM policies and access to Service Account keys
 - Archive Cloud Audit Logs in Cloud Storage buckets for long term retention
- Use Groups when possible
 - Makes it easy to manage users and permissions

User Identity Management in Google Cloud

- Email used to create free trial account => **"Super Admin"**
 - Access to everything in your GCP organization, folders and projects
 - Manage access to other users **using their Gmail accounts**
- However, this is **NOT recommended** for enterprises
- **Option 1:** Your Enterprise is using **Google Workspace**
 - Use Google Workspace to manage users (groups etc)
 - Link Google Cloud Organization with Google Workspace
- **Option 2:** Your Enterprise uses an Identity Provider of its own
 - **Federate** Google Cloud with your Identity Provider



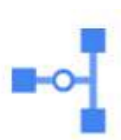
Corporate Directory Federation

- **Federate** Cloud Identity or Google Workspace **with your external identity provider (IdP)** such as Active Directory or Azure Active Directory.
- **Enable Single Sign On:**
 - 1: Users are redirected to an external IdP to authenticate
 - 2: When users are authenticated, SAML assertion is sent to Google Sign-In
- **Examples:**
 - Federate Active Directory with Cloud Identity by using Google Cloud Directory Sync (GCDS) and Active Directory Federation Services (AD FS)
 - Federating Azure AD with Cloud Identity



Cloud VPN

- Cloud VPN - Connect on-premise network to the GCP network
 - Implemented using **IPSec VPN Tunnel**
 - Traffic through internet (public)
 - Traffic encrypted using **Internet Key Exchange** protocol
- Two types of Cloud VPN solutions:
 - HA VPN (SLA of 99.99% service availability with two external IP addresses)
 - Only dynamic routing (BGP) supported
 - Classic VPN (SLA of 99.9% service availability, a single external IP address)
 - Supports Static routing (policy-based, route-based) and dynamic routing using BGP



Cloud VPN

Cloud Interconnect



- High speed physical connection between on-premise and VPC networks:
 - Highly available and high throughput
 - Two types of connections possible
 - Dedicated Interconnect - 10 Gbps or 100 Gbps configurations
 - Partner Interconnect - 50 Mbps to 10 Gbps configurations
- Data exchange happens through a private network:
 - Communicate using VPC network's internal IP addresses from on-premise network
 - Reduces egress costs
 - As public internet is NOT used
- (Feature) Supported Google API's and services can be privately accessed from on-premise
- Use only for high bandwidth needs:
 - For low bandwidth. Cloud VPN is recommended

Direct Peering

- Connect customer network to google network using network peering
 - Direct path from on-premises network to Google services
- **Not a GCP Service**
 - Lower level network connection outside of GCP
- **NOT RECOMMENDED:**
 - Use Cloud Interconnect and Cloud VPN

User Identity Management in Google Cloud

- Email used to create free trial account => **"Super Admin"**
 - Access to everything in your GCP organization, folders and projects
 - Manage access to other users **using their Gmail accounts**
- However, this is **NOT recommended** for enterprises
- **Option 1:** Your Enterprise is using **Google Workspace**
 - Use Google Workspace to manage users (groups etc)
 - Link Google Cloud Organization with Google Workspace
- **Option 2:** Your Enterprise uses an Identity Provider of its own
 - **Federate** Google Cloud with your Identity Provider



IAM Members/Identities



- **Google Account** - Represents a person (an email address)
- **Service account** - Represents an application account (Not person)
- **Google group** - Collection - Google & Service Accounts
 - Has an unique email address
 - Helps to apply access policy to a group
- **Google Workspace domain:** Google Workspace (formerly G Suite) provides collaboration services for enterprises:
 - Tools like Gmail, Calendar, Meet, Chat, Drive, Docs etc are included
 - If your enterprise is using Google Workspace, you can manage permissions using your Google Workspace domain
- **Cloud Identity domain** - Cloud Identity is an Identity as a Service (IDaaS) solution that centrally manages users and groups.
 - You can use IAM to manage access to resources for each Cloud Identity account

IAM Members/Identities - Use Cases

| Scenario | Solution |
|--|--|
| All members in your team have G Suite accounts. You are creating a new production project and would want to provide access to your operations team | Create a Group with all your operations team. Provide access to production project to the Group. |
| All members in your team have G Suite accounts. You are setting up a new project. You want to provide a one time quick access to a team member. | Assign the necessary role directly to G Suite email address of your team member If it is not a one time quick access, the recommended approach would be to create a Group |
| You want to provide an external auditor access to view all resources in your project BUT he should NOT be able to make any changes | Give them roles/viewer role (Generally basic roles are NOT recommended BUT it is the simplest way to provide view only access to all resources!) |
| Your application deployed on a GCE VM (Project A) needs to access cloud storage bucket from a different project (Project B) | In Project B, assign the right role to GCE VM service account from Project A |

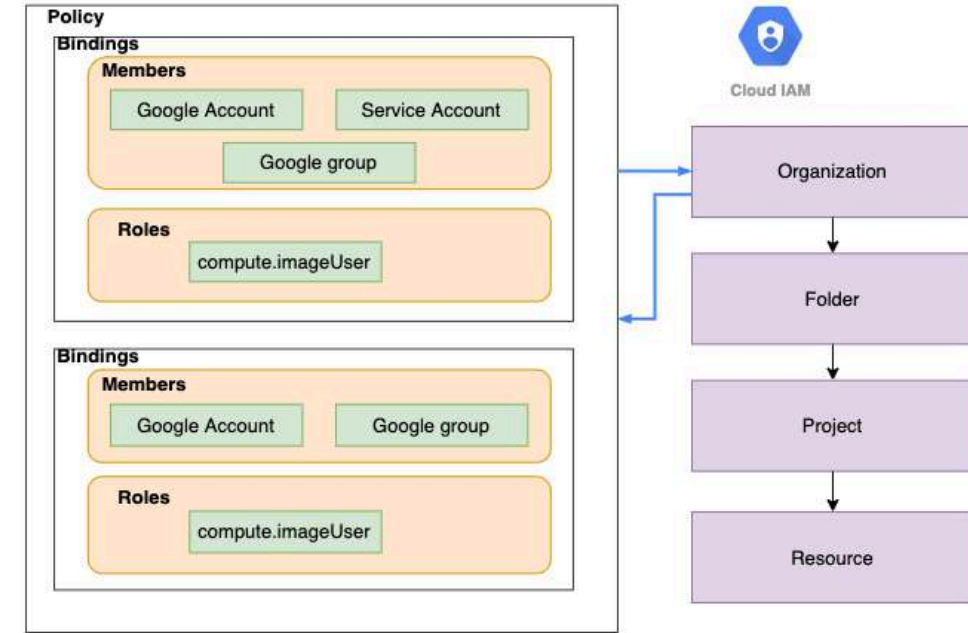
Organization Policy Service

- How to enable **centralized constraints** on all resources created in an Organization?
 - Configure **Organization Policy**
 - Example: Disable creation of Service Accounts
 - Example: Allow/Deny creation of resources in specific regions
- Needs a Role - Organization Policy Administrator
- (Remember) **IAM** focuses on **Who**
 - Who can take specific actions on resources?
- (Remember) Organization Policy focuses on **What**
 - What can be done on specific resources?



Resource Hierarchy & IAM Policy

- IAM Policy can be set at any level of the hierarchy
- Resources inherit the policies of **All parents**
- The effective policy for a resource is the union of the policy on that resource and its parents
- Policy inheritance is transitive:
 - For example: Organization policies are applied at resource level
- You can't restrict policy at lower level if permission is given at an higher level



Getting Started with Identity Platform

- **Identity Platform:** Customer identity and access management
- **What's the difference:** Cloud IAM vs Identity Platform
 - **Cloud IAM:** Employees and Partners Authorization
 - Control access to Google Cloud Resources
 - Member, Roles, Policy, Service Accounts
 - **Identity Platform:** Customer identity and access management (CIAM)
 - Authentication and Authorization for your applications and services
- **Identity Platform: Key Features**
 - Authentication & authorization for web & mobile apps (iOS, Android, ..)
 - Multiple authentication methods
 - SAML, OIDC, email/password, phone, social - Google/Facebook/Twitter/..
 - Features: User sign-up and sign-in, MFA etc.
 - An upgrade from Firebase Authentication Legacy
 - Integrates well with Identity-Aware Proxy

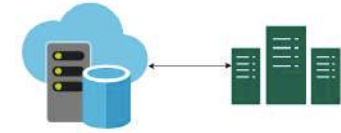


Cloud IAM vs Identity Platform - Scenarios

| Scenario | Solution |
|--|-----------------------------|
| An Application on a GCE VM needs access to cloud storage | Cloud IAM - Service Account |
| An enterprise user need access to upload objects to a Cloud Storage bucket | Cloud IAM |
| I want to manage end users for my application | Identity Platform |
| I want to enable "Login using facebook/twitter" for my application | Identity Platform |
| I want to create user sign-up and sign-in workflows for my application | Identity Platform |

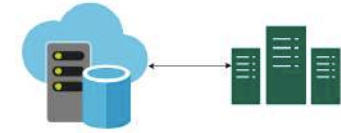
Cloud Computing: Public vs Private vs Hybrid Cloud - 1

- **Public Cloud:** You host everything in the cloud
 - You DO NOT need a data center anymore
 - NO Capital Expenditure needed
 - UNLIMITED scale at your disposal
 - Hardware resources are owned by Google Cloud
 - Capacity management, hardware failures and security of the data center are Google Cloud's responsibility
 - **Summary:** Hardware owned by Google Cloud and shared between multiple customers
- **Private Cloud:** You host everything in your own data center
 - Needs Capital Expenditure
 - Incur staffing and maintenance expenses for infrastructure
 - Adding infrastructure needs planning (time consuming and expensive)
 - For example: You might NOT be able to quickly handle a sudden increase in user load



Cloud Computing: Public vs Private vs Hybrid Cloud - 2

- **Hybrid Cloud:** Combination of both (Public & Private)
 - Use Public Cloud for some workloads and Private cloud for others
 - **Examples:**
 - Using Google Cloud Dataflow to process a data stream from your on-premise applications
 - Connect an on-premise application to Google Cloud SQL database
 - **Advantage:** Provides you with **flexibility**
 - Go on-premises or cloud based on specific requirement
 - **Disadvantage:** Increases complexity
- **Multi Cloud:** Using Multiple Cloud Platforms with/without on-premise infrastructure
 - Even MORE flexibility
 - BUT increased complexity



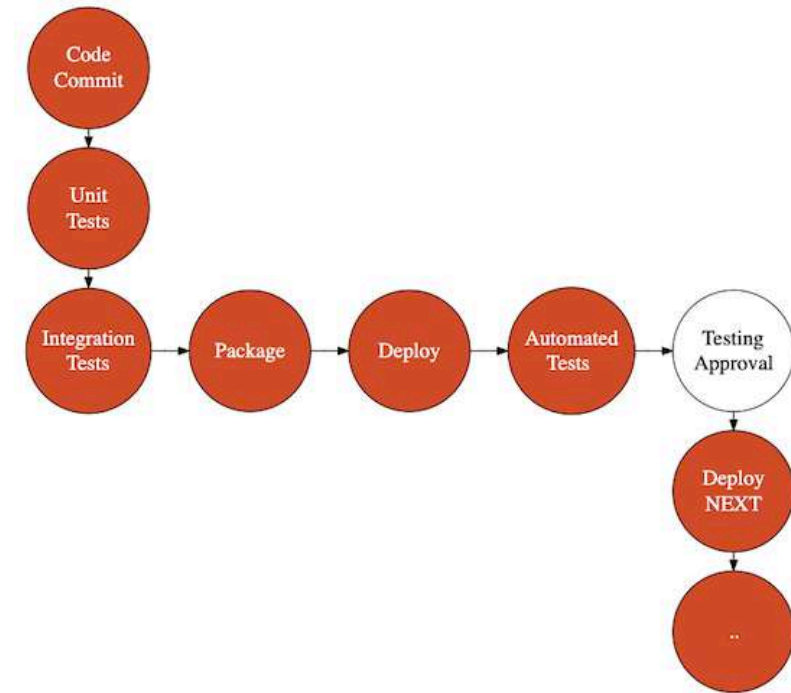
DevOps



- Getting Better at "**Three Elements of Great Software Teams**"
 - **Communication** - Get teams together
 - **Feedback** - Earlier you find a problem, easier it is to fix
 - **Automation** - Automate testing, infrastructure provisioning, deployment, and monitoring

DevOps - CI, CD

- **Continuous Integration**
 - Continuously run your tests and packaging
- **Continuous Deployment**
 - Continuously deploy to test environments
- **Continuous Delivery**
 - Continuously deploy to production



DevOps - CI CD - Recommended Things to Do

- **Static Code Analysis**

- Lint, Sonar
- Including Static Security Checks (Source Code Security Analyzer software like Veracode or Static Code Analyzer)

- **Runtime Checks**

- Run Vulnerability Scanners (automated tools that scan web applications for security vulnerabilities)

- **Tests**

- Unit Tests (JUnit, pytest, Jasmine etc)
- Integration Tests (Selenium, Robot Framework, Cucumber etc)
- System Tests (Selenium, Robot Framework, Cucumber etc)
- Sanity and Regression Tests

DevOps - CI, CD Tools

- **Cloud Source Repositories:** Fully-featured, private Git repository
 - Similar to Github
- **Container Registry:** Store your Docker images
- **Jenkins:** Continuous Integration
- **Cloud Build:** Build deployable artifacts (jars or docker images) from your source code and configuration
- **Spinnaker:** Multi-cloud continuous delivery platform
 - Release software changes with high velocity and confidence
 - Supports deployments to Google Compute Engine, Google Kubernetes Engine, Google App Engine and other cloud platforms
 - Supports Multiple Deployment Strategies

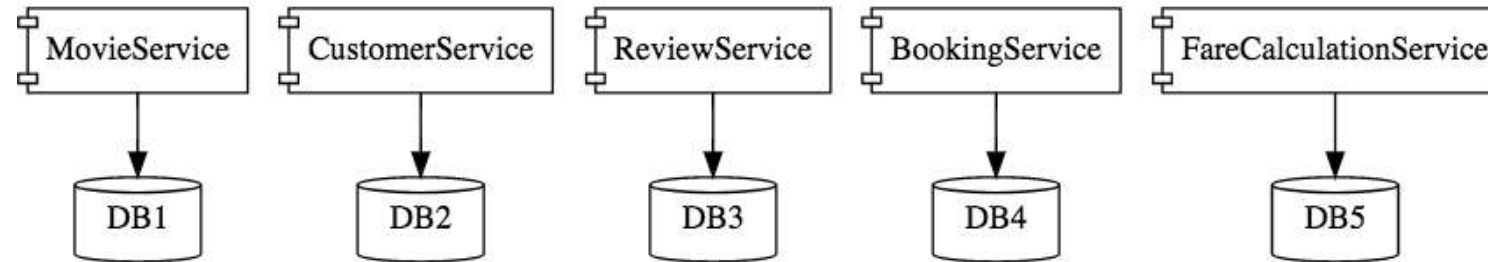


Container
Registry



Cloud Source
Repositories

Exploring Container Registry and Artifact Registry



- You've created docker images for your microservices:
 - Where do you store them?
 - Two Options: Container Registry and Artifact Registry
 - **Container Registry:** Uses GCS bucket to store images
 - Supports Container images only
 - (Alternative) Docker Hub
 - Example: `us.gcr.io/PROJECT-ID/...`
 - Permissions are managed by managing access to GCS buckets
 - **Artifact Registry:** Evolution of **Container Registry**
 - Builds upon the capabilities of Container Registry
 - **Manage BOTH container images and non-container artifacts**
 - Example: `us-central1-docker.pkg.dev/PROJECT-ID/...`

Exploring Artifact Registry

- Supports **multiple artifact formats**:
 - Container images, language packages, and OS packages are supported
- You need to create **separate repository**
 - Does NOT use GCS buckets
 - Repository can be regional or multi-regional
- Example: *us-central1-docker.pkg.dev/PROJECT-ID/...*
- (RECOMMENDED) Control access by using Artifact Registry Roles
 - Artifact Registry Reader
 - Artifact Registry Writer
 - Artifact Registry Administrator etc..
- You can also configure repository specific permissions



DevOps Example: Cloud Run with Cloud Build

In 28
Minutes

- **STEP 1:** Checkout Source Code from Cloud Source Repositories
- **STEP 2:** Build a Docker Image
- **STEP 3:** Store Docker Image in the Container Registry
- **STEP 4:** Deploy Docker Image to Cloud Run



DevOps - Infrastructure as Code



- Treat infrastructure the same way as application code
- Track your infrastructure **changes over time** (version control)
- Bring **repeatability** into your infrastructure
- Two Key Parts
 - **Infrastructure Provisioning**
 - Provisioning compute, database, storage and networking
 - Open source cloud neutral - Terraform
 - GCP Service - Google Cloud Deployment Manager
 - **Configuration Management**
 - Install right software and tools on the provisioned resources
 - Open Source Tools - Chef, Puppet, Ansible and SaltStack

Operations

| Operation | GCP |
|------------------------------------|------------------|
| Monitoring - Metrics and Alerts | Cloud Monitoring |
| Centralized Logging | Cloud Logging |
| Audit logging | Cloud Audit Logs |
| Real-time exception monitoring | Error Reporting |
| Live Debugging | Cloud Debugger |
| Distributed tracing | Cloud Trace |
| Statistical, low-overhead profiler | Cloud Profiler |



Monitoring



Logging







Trace



Debugger

Cloud Operations Scenarios - Microservices

| Scenario | Service | |
|---|------------------|---|
| I want to get metrics related to a specific microservice instance | Cloud Monitoring |  Monitoring |
| I want to look at logs for a specific microservice | Cloud Logging |  Logging |
| I want to track exceptions happening in a specific microservice | Error Reporting | |
| I want to trace request across microservices | Cloud Trace |  Trace |
| I want to solve a performance issue in a specific microservice | Cloud Profiler |  Debugger |

Site Reliability Engineering (SRE)

- DevOps++ at Google
- SRE teams **focus on every aspect of an application**
 - availability, latency, performance, efficiency, change management, monitoring, emergency response, and capacity planning
- **Key Principles:**
 - Manage by Service Level Objectives (SLOs)
 - Minimize Toil
 - Move Fast by Reducing Cost of Failure
 - Share Ownership with Developers



Google Cloud

Site Reliability Engineering (SRE) - Key Metrics

- **Service Level Indicator(SLI):** Quantitative measure of an aspect of a service
 - Categories: availability, latency, throughput, durability, correctness (error rate)
 - Typically aggregated - "Over 1 minute"
- **Service Level Objective (SLO) - SLI + target**
 - 99.99% Availability, 99.999999999% Durability
 - Response time: 99th percentile - 1 second
 - Choosing an appropriate SLO is complex
- **Service Level Agreement (SLA):** SLO + consequences (contract)
 - What is the consequence of NOT meeting an SLO? (Defined in a contract)
 - Have stricter internal SLOs than external SLAs
- **Error budgets: (100% – SLO)**
 - How well is a team meeting their reliability objectives?
 - Used to manage development velocity

Site Reliability Engineering (SRE) - Best Practices



Google Cloud

- **Handling Excess Loads**

- **Load Shedding**

- API Limits
 - Different SLAs for different customers
 - Streaming Data
 - If you are aggregating time series stream data, in some scenarios, you can drop a part of data

- **Reduced Quality of Service**

- Instead of talking to a recommendations API, return a hardcoded set of products!
 - Not always possible:
 - Example: if you are making a payment

- **Avoiding Cascading Failures**

- **Plan to avoid thrashing**

- Circuit Breaker
 - Reduced Quality of Service

Site Reliability Engineering (SRE) - Best Practices - 2

- **Penetration Testing (Ethical Hacking)**
 - Simulate an attack with the objective of finding security vulnerabilities
 - Should be authorized by project owners
 - No need to inform Google
 - Ensure you are only testing your projects and are in compliance with terms of service!
 - Can be white box (Hacker is provided with information about infrastructure and/or applications) or black box (No information is provided)
- **Load Testing** (JMeter, LoadRunner, Locust, Gatling etc)
 - Simulate real world traffic as closely as possible
 - Test for spiky traffic - suddenly increases in traffic



Google Cloud

Site Reliability Engineering (SRE) - Best Practices - 3

- **Resilience Testing** - "How does an application behaves under stress?"
- **Resilience** - "Ability of system to provide acceptable behavior even when one or more parts of the system fail"
- **Approaches:**
 - **Chaos Testing (Simian Army)** - cause one or more layers to fail
 - "unleashing a wild monkey with a weapon in your data center to randomly shoot down instances and chew through cables"
 - Add huge stress on one of the layers
 - **Include network in your testing** (VPN, Cloud Interconnect etc..)
 - Do we fall back to VPN if direct interconnect fails?
 - What happens when internet is down?
 - **Best Practice: DiRT** - disaster recovery testing at Google
 - Plan and execute outages for a defined period of time
 - Example: Disconnecting complete data center



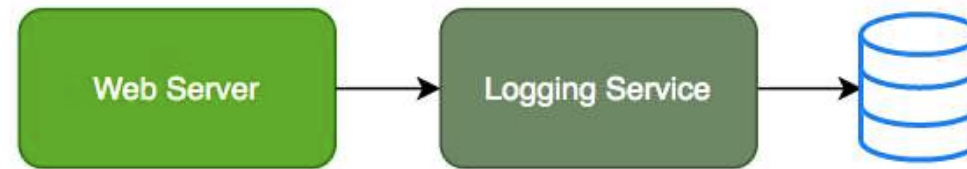
Google Cloud

Decoupling Applications with Pub/Sub

Need for Asynchronous Communication

- Why do we need asynchronous communication?

Synchronous Communication



- Applications on your web server make synchronous calls to the logging service
- What if your logging service goes down?
 - Will your applications go down too?
- What if all of sudden, there is high load and there are a lot of logs coming in?
 - Log Service is not able to handle the load and goes down very often

Asynchronous Communication - Decoupled



- Create a topic and have your applications put log messages on the topic
- Logging service picks them up for processing when ready
- Advantages:
 - Decoupling: Publisher (Apps) don't care about who is listening
 - Availability: Publisher (Apps) up even if a subscriber (Logging Service) is down
 - Scalability: Scale consumer instances (Logging Service) under high load
 - Durability: Message is not lost even if subscriber (Logging Service) is down

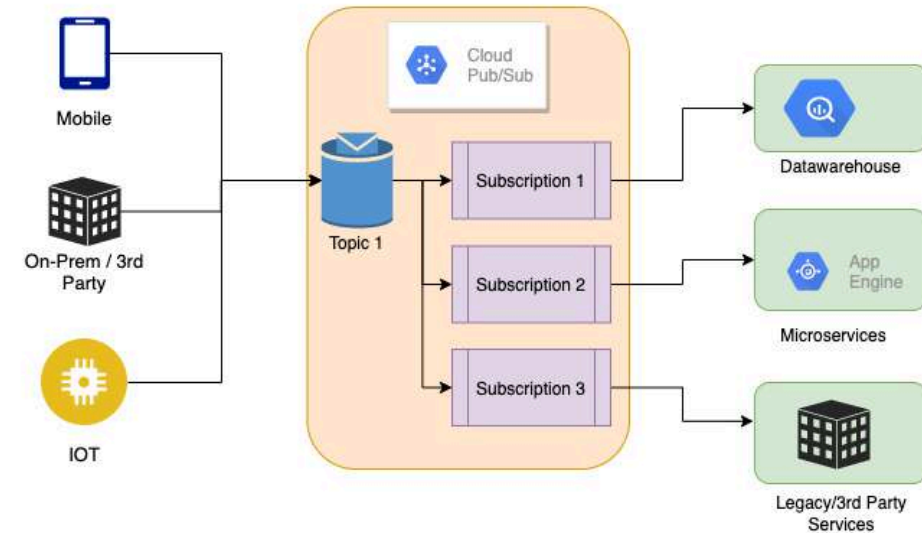
Pub/Sub

- Reliable, scalable, fully-managed asynchronous messaging service
- Backbone for **Highly Available** and **Highly Scalable** Solutions
 - Auto scale to process billions of messages per day
 - Low cost (Pay for use)
- Usecases: Event ingestion and delivery for streaming analytics pipelines
- Supports push and pull message deliveries



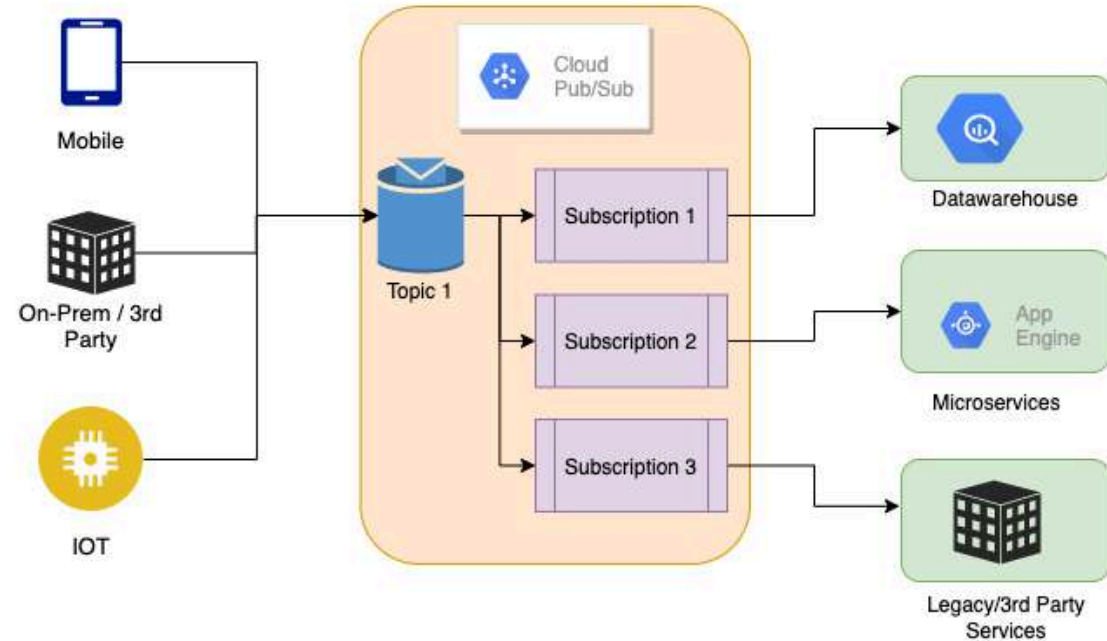
Pub/Sub - How does it work?

- **Publisher** - Sender of a message
 - Publishers send messages by making HTTPS requests to pubsub.googleapis.com
- **Subscriber** - Receiver of the message
 - **Pull** - Subscriber pulls messages when ready
 - Subscriber makes HTTPS requests to pubsub.googleapis.com
 - **Push** - Messages are sent to subscribers
 - Subscribers provide a web hook endpoint at the time of registration
 - When a message is received on the topic, A HTTPS POST request is sent to the web hook endpoints
- **Very Flexible** Publisher(s) and Subscriber(s) Relationships: One to Many, Many to One, Many to Many



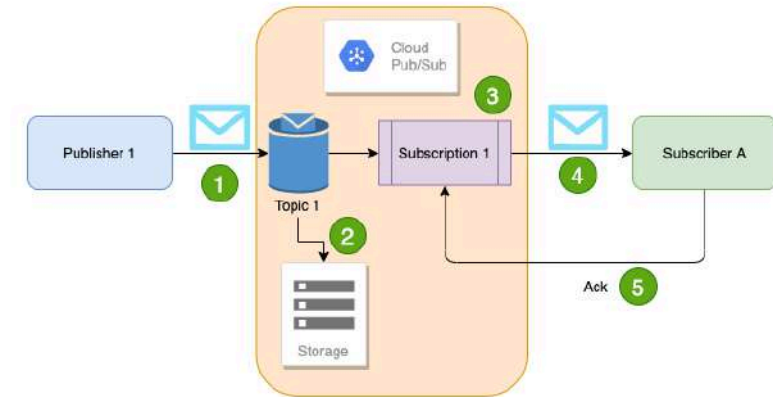
Pub/Sub - Getting Ready with Topic and Subscriptions

- Step 1 : Topic is created
- Step 2 : Subscription(s) are created
 - Subscribers register to the topic
 - Each Subscription represents discrete pull of messages from a topic:
 - Multiple clients pull same subscription => messages split between clients
 - Multiple clients create a subscription each => each client will get every message

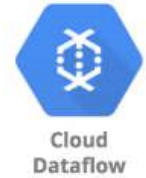


Pub/Sub - Sending and Receiving a Message

- Publisher sends a message to Topic
- Message **individually** delivered to each and every subscription
 - Subscribers can receive message either by:
 - Push: Pub/Sub sends the message to Subscriber
 - Pull: Subscribers poll for messages
- Subscribers send acknowledgement(s)
- Message(s) are removed from subscriptions message queue
 - Pub/Sub ensures the message is retained **per subscription** until it is acknowledged



Cloud Dataflow

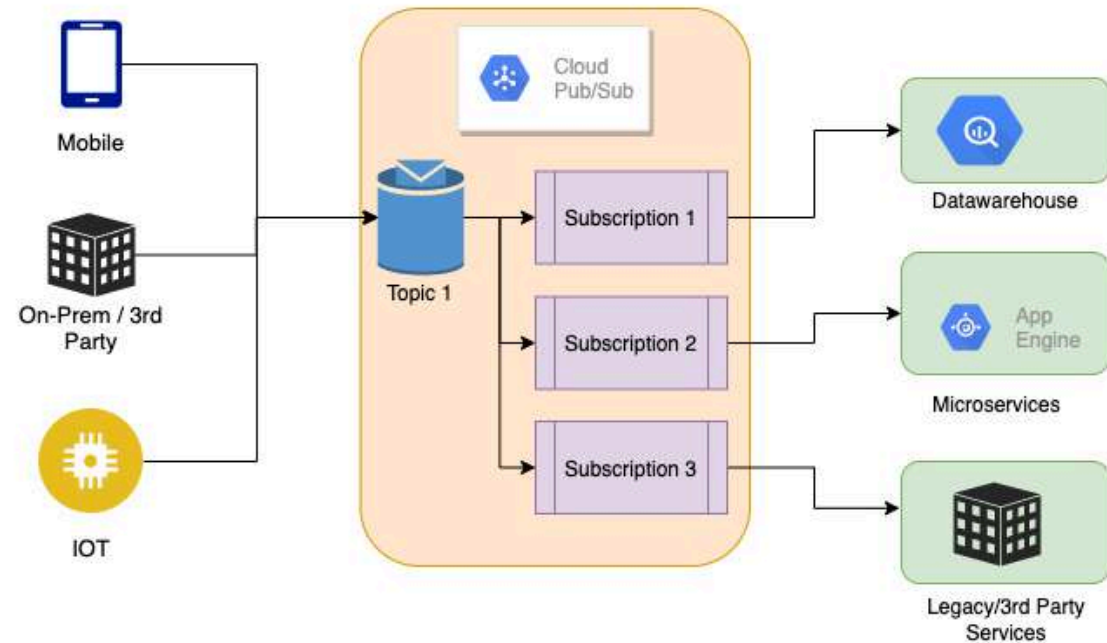


- **Cloud Dataflow** is a difficult service to describe:
 - Let's look at a **few example pipelines** you can build:
 - Pub/Sub > Dataflow > BigQuery (Streaming)
 - Pub/Sub > Dataflow > Cloud Storage (Streaming - files)
 - Cloud Storage > Dataflow > Bigtable/CloudSpanner/Datastore/BigQuery (Batch - Load data into databases)
 - Bulk compress files in Cloud Storage (Batch)
 - Convert file formats between Avro, Parquet & csv (Batch)
- **Streaming and Batch Usecases**
 - Realtime Fraud Detection, Sensor Data Processing, Log Data Processing, Batch Processing (Load data, convert formats etc)
- Use **pre-built** templates
- Based on **Apache Beam** (supports Java, Python, Go ...)
- Serverless (and Autoscaling)

Data Architectures in Google Cloud

Architecture - Loose Coupling with Pub/Sub

- Whenever you want to **decouple** a publisher from a subscriber, consider Pub/Sub
- Pub/Sub is used in:
 - Microservices Architectures
 - IOT Architectures
 - Streaming Architectures



Data Formats

- **Three Data Formats**

- **Structured: Tables, Rows and Columns (Relational)**

- Example: Order Information, Product Inventory, etc
 - Google Cloud Services:
 - Cloud SQL (Regional Transactional)
 - Cloud Spanner (Global Unlimited Scale Transactional)
 - BigQuery (Data warehousing and ML using SQL)

- **Semi Structured: Flexible Schema**

- Key-Value, Document (JSON) - Social Media Profile Information
 - Google Cloud Services: Cloud Firestore/Datastore

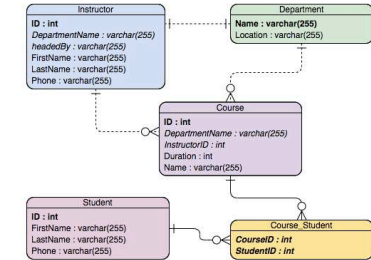
- **Unstructured: Video, Audio, Image, Text, Binary files**

- Example: Product images, Product videos
 - Google Cloud Services: Cloud Storage

- **(NEW) BigQuery can also store Semi Structured data**

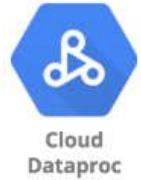
- BigQuery ML can be used to do ML using Unstructured data (images, videos) stored in Cloud Storage

| ID | DepartmentName | Name | Duration | InstructorID |
|----|------------------|-----------------------------|----------|--------------|
| 1 | Computer Science | Algorithms | 8 | 2 |
| 2 | Computer Science | Data Structures | 6 | 4 |
| 3 | Computer Science | Operating Systems | 5 | 4 |
| 4 | Computer Science | Database Management Systems | 20 | 2 |



```
{
  "customerId": "99999999",
  "firstName": "Ranga",
  "lastName": "Ranga",
  "address": {
    "number": "505",
    "street": "Main Street",
    "city": "Hyderabad",
    "state": "Telangana"
  },
  "socialProfiles": [
    {
      "name": "twitter",
      "username": "@in28minutes"
    },
    {
      "name": "linkedin",
      "username": "rangaraokaranam"
    }
  ]
}
```

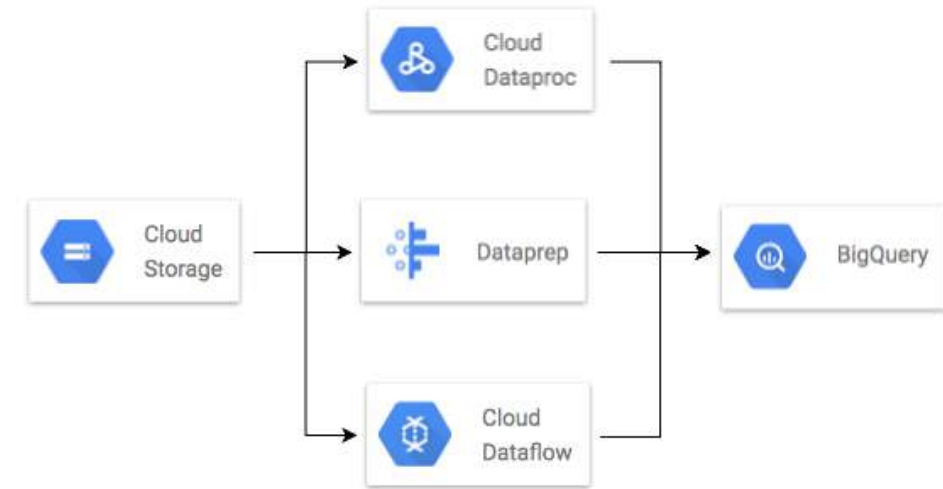

Cloud Dataproc



- Managed **Spark and Hadoop** service:
 - Variety of jobs are supported:
 - Spark, PySpark, SparkR, Hive, SparkSQL, Pig, Hadoop
 - Perform complex batch processing
- **Multiple Cluster Modes:**
 - Single Node / Standard/ High Availability (3 masters)
 - Use regular/preemptible VMs
- Use case: Move your Hadoop and Spark clusters to the cloud
 - Perform your machine learning and AI development using open source frameworks
- (ALTERNATIVE) **BigQuery** - When you run SQL queries on Petabytes
 - Go for Cloud Dataproc when you need more than queries (Example: Complex batch processing Machine Learning and AI workloads)
- (ALTERNATIVE) **Dataflow** - Simple pipelines without managing clusters

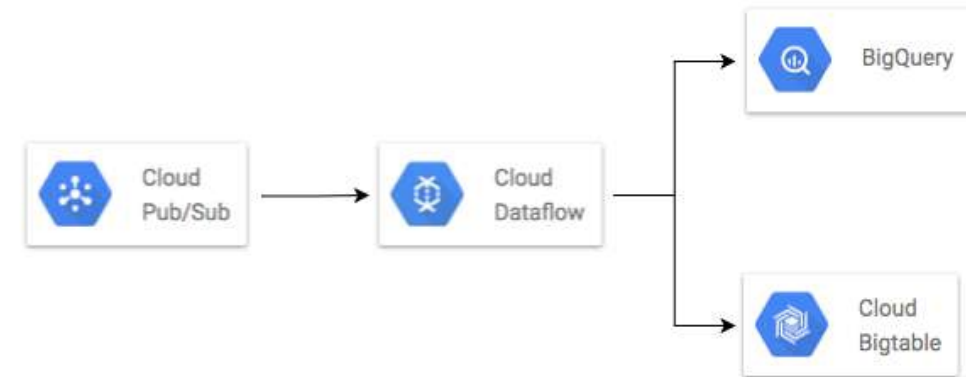
Architecture 1 - Big Data Flow - Batch Ingest

- Use extract, transform, and load (ETL) to load data into BigQuery
 - **Dataprep:** Clean and prepare data
 - **Dataflow:** Create data pipelines (and ETL)
 - **Dataproc:** Complex processing using Spark and Hadoop
- **Data Studio:** Visualize data in BigQuery
- **Looker:** Multi-cloud Enterprise Business Intelligence



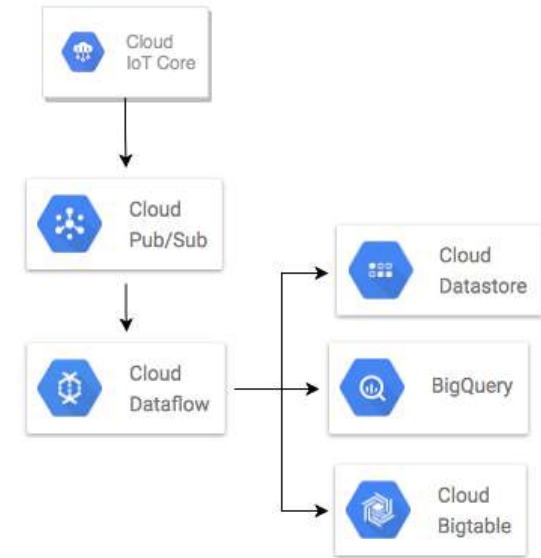
Architecture 2 - Streaming Data

- **Pub/Sub:** Receive messages
- **Dataflow:** Analyze, aggregate and filter data
- For **pre-defined time series** analytics, storing data in **Bigtable** gives you the ability to perform rapid analysis
- For **ad hoc complex analysis**, prefer **BigQuery**



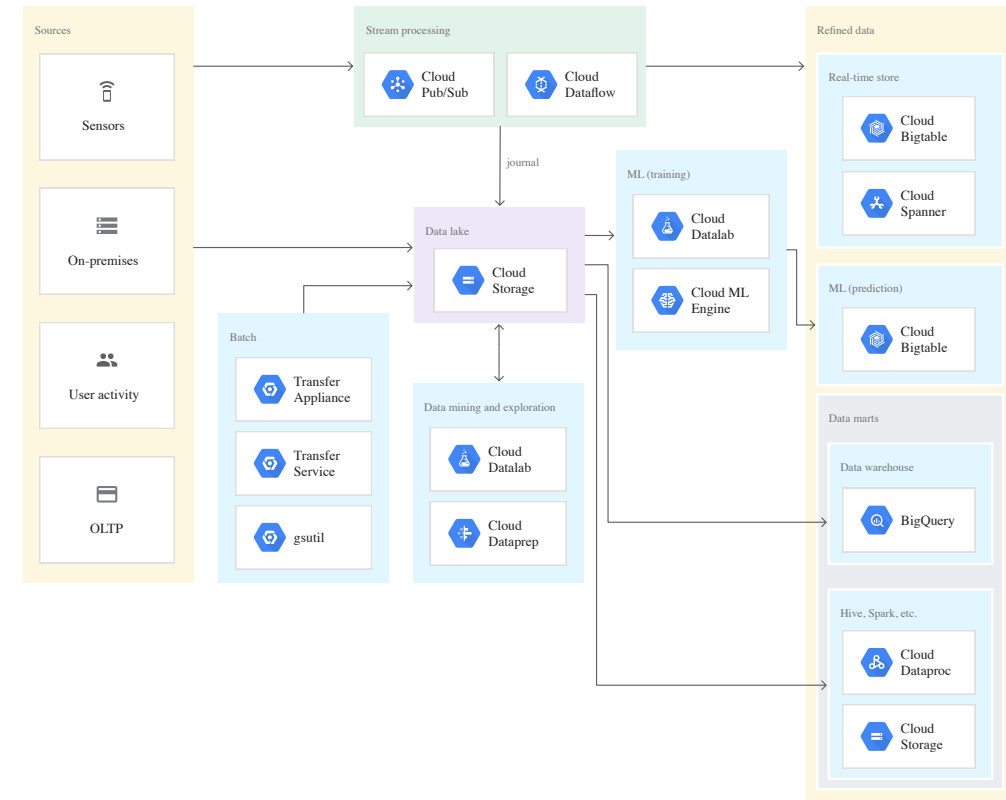
Architecture 3 - IOT

- **IoT Core:** Manage IoT (registration, authentication, and authorization) devices
 - Send/receive messages/real-time telemetry from/to IoT devices
- **Pub/Sub:** Durable message ingestion service (allows buffering)
- **Dataflow:** Processing data (ETL & more..)
 - Alternative: Use Cloud Functions to trigger alerts
- **Data Storage and Analytics:**
 - Make IOT data available to mobile or web apps => **Datastore**
 - Execute pre-defined time series queries => **Bigtable**
 - More complex or ad hoc analytics/analysis => **BigQuery**



Data Lake - Simplified Big Data Solutions

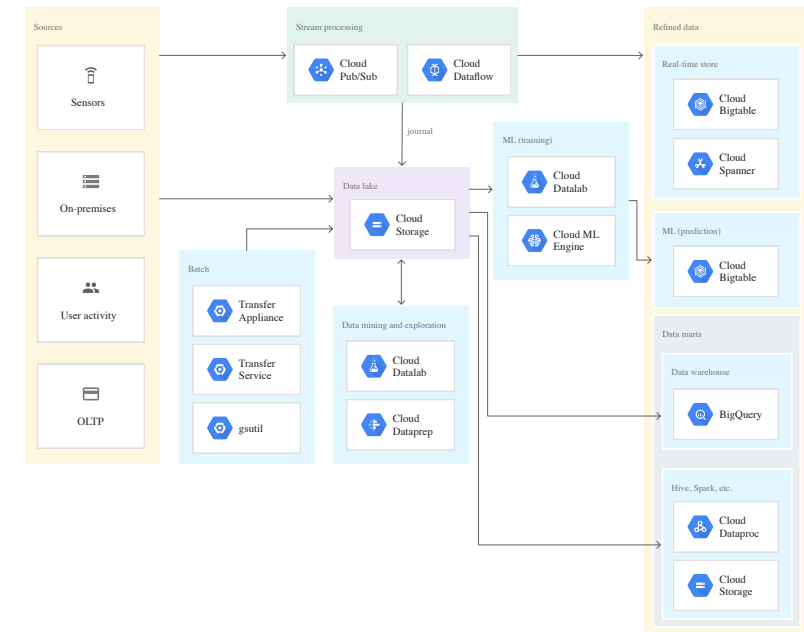
- Usual big data solutions are complex
- How can we make collecting, analyzing (reporting, analytics, machine learning) and visualizing huge data sets easy?
- How to design solutions that scale?
- How to build flexibility while saving cost?
- **Data Lake**
 - Single platform with combination of solutions for data storage, data management and data analytics



<https://cloud.google.com/solutions/build-a-data-lake-on-gcp>

GCP Data Lakes - Storage and Ingestion

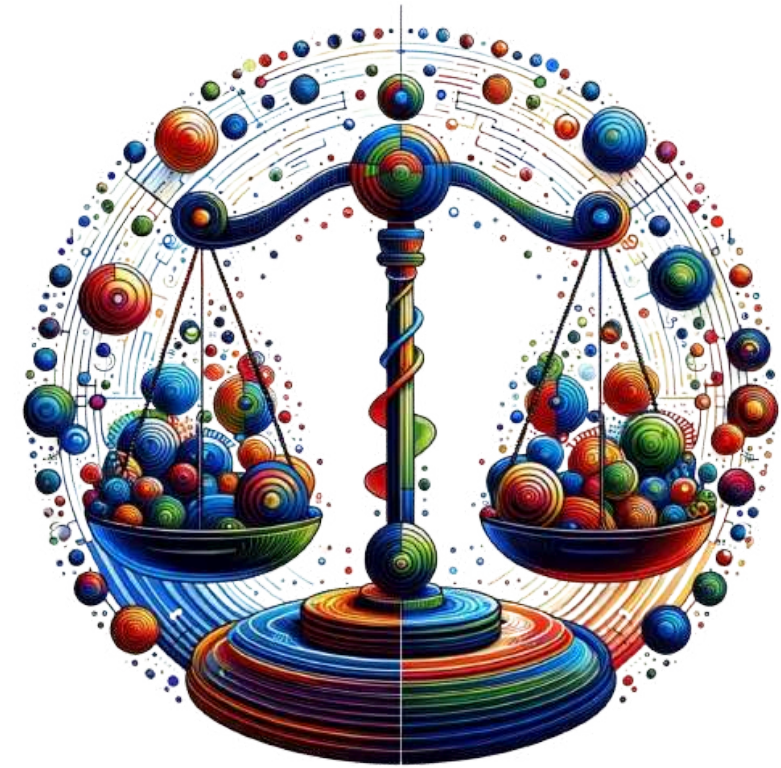
- **Storage:** Cloud Storage (low cost + durability + performance + flexible processing)
- **Data Ingestion:**
 - Streaming data - Cloud Pub/Sub + Cloud Dataflow
 - Batch - Transfer Service + Transfer Appliance + gsutil
- **Processing and analytics:**
 - Run in-place querying using SQL queries using BigQuery or (Hive on Dataproc)
- **Data Mining and Exploration:**
 - Clean and transform raw data with Dataprep
 - Use Cloud Datalab (data science libraries such as TensorFlow and NumPy) for exploring



<https://cloud.google.com/solutions/build-a-data-lake-on-gcp>

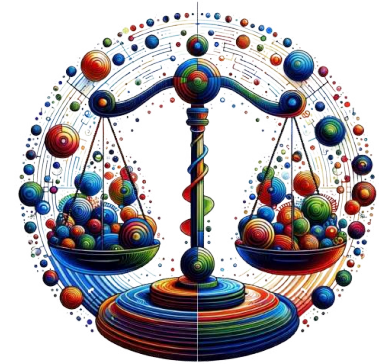
Why do we need Data Governance?

- **Bad data:** Bad data leads to poor business decisions
- **Data leaks:** Data leaks can lead to a reputation loss
- **How to avoid these?**
 - **Data Governance**



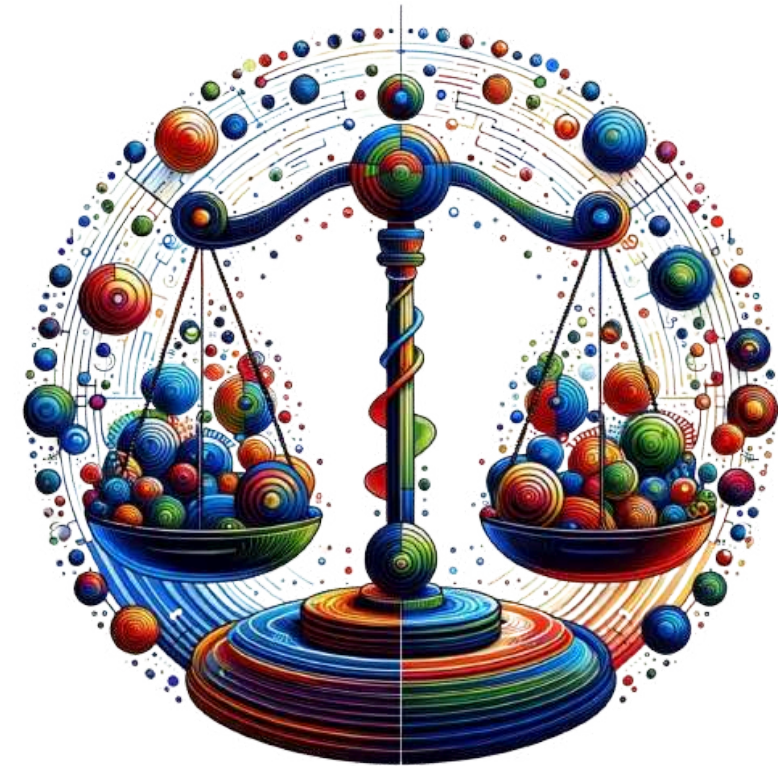
What is Data Governance about?

- **Key Decisions:** Here are the important things to think about:
 - **Data Management:** What's our method for keeping our data organized and easy to find?
 - **Life Cycle Management:** How do we handle our data from when we first get it until we don't need it anymore?
 - **Ownership and Accountability:** Who looks after our data to make sure it's correct, safe, and private?
 - **Data Quality Assurance:** How do we check that our data is right and trustworthy for making decisions?
 - **Data Security:** What do we do to keep our data safe from hackers?
 - **Risk Management:** How are we planning to reduce the chances of our data being lost or exposed?
 - **Data Transparency:** Are we open about the data we collect and how we use it, both inside and outside the company?



Dataplex - Data Mesh in Google Cloud

- **Dataplex is a Data Mesh:** Unified dashboard with visibility into all data assets (data lakes, data warehouses, ..)
 - **Single pane of glass:** Data management across silos
 - **Centralized security and governance:** Comprehensive security & governance policies (IAM, Access Controls,..) that are uniformly enforced across all data assets
 - **Unified search and data discovery:** Integrates seamlessly with Google Cloud's Data Catalog, offering a rich, searchable repository of metadata
 - **Built-in data intelligence:** Advanced data intelligence capabilities by integrating with BigQuery for analytics and Cloud Storage for data lake solutions



Making Best Use of Data - Scenarios

| Scenario | Solution |
|--|--|
| Sales representatives are struggling to identify buying patterns and personalize customer interactions due to the inability to access and analyze sales data scattered across various sources | Combine BigQuery's data integration with Looker's intuitive dashboards |
| A fast-growing online gaming platform needs to analyze player data in real-time to improve user experience and retention rates | Implement a streaming pipeline - capture live gaming event data with Pub/Sub. Use Cloud Dataflow for real-time data processing. Load the processed data into Google BigQuery. Visualize using Looker (if needed) |
| A multinational corporation collects vast amounts of diverse data - sales figures, customer feedback, and sensor data from manufacturing plants. CHALLENGE: Store this data cost-effectively while keeping it accessible for future analytics and machine learning applications | Implement a data lake using Google Cloud Storage |

API Management in Google Cloud

REST API Challenges

- Most applications today are built around **REST API**:
 - **Resources** (/todos, /todos/{id}, etc.)
 - **Actions** - HTTP Methods - GET, PUT, POST, DELETE etc.
- Management of REST API is not easy:
 - You've to take care of **authentication and authorization**
 - You've to be able to set limits (rate limiting, quotas) for your API consumers
 - You've to take care of implementing **multiple versions** of your API
 - You would want to implement monitoring, caching and a lot of other features..



Client



Apigee API Platform



App
Engine

Exploring API management in Google Cloud

- **Apigee API Management:** Comprehensive API management platform
 - Deployment options: Cloud, on-premises or hybrid
 - Manage **Complete API life cycle**
 - Design, Secure, Publish, Analyze, Monitor and Monetize APIs
 - Powerful Features
 - On-boarding partners and developers
 - Supports complex integrations (REST, gRPC, Non-gRPC-REST, integrate with GCP, on-premises or hybrid apps)
- **Cloud Endpoints:** Basic API Management for Google Cloud backends
 - Little complicated to setup: You need to build a container and deploy to Cloud Run
 - Supports REST API and gRPC
- **API gateway:** Newer, Simpler API Management for Google Cloud backends
 - Simpler to setup
 - Supports REST API and gRPC

Trust and Security with Google Cloud

Cloud vs On-Premises Security

| Factor | On-Premises Security | Cloud Security |
|-----------------------|--|--|
| Responsibility | The business is solely responsible for its security | Security is a shared responsibility between the provider and the business |
| Costs | Requires significant upfront investment in hardware and software | Typically involves lower upfront costs but ongoing operational expenses |
| Maintenance | Businesses need to manage their own updates and maintenance | Depending on the service, the provider helps with security updates and system maintenance |
| Expertise | Requires in-house security expertise or external consultants | Option to get access to top-tier security expertise through the provider |
| Data Control | Full control over data storage and security protocols | Less direct control over where and how data is stored |
| Compliance | Businesses must individually ensure and maintain compliance | Cloud providers often have certifications making compliance easier (Shared responsibility) |

Key Characteristics for Cloud Security

- **Control:** Decide who gets access
 - **Example:** Only few employees can view sensitive company data
- **Compliance:** Follows legal rules
 - **Example:** Protect customer data as the law requires
- **Confidentiality:** Keeps information secret
 - **Example:** Encrypt messages so that only sender and recipient can read them
- **Integrity:** Ensures data stays accurate
 - **Example:** A bank system checks that no one changes your balance without permission
- **Availability:** Ensure apps & data are available always
 - **Example:** A banking website remains accessible even during high traffic or an attack



Key Characteristics for Cloud Security - Scenarios

| Scenario | Concept |
|--|-----------------|
| Only the HR department has the ability to access employee records, while all other departments are restricted from viewing this sensitive information. | Control |
| A healthcare provider implements robust data protection measures to ensure patient records are handled in accordance with HIPAA regulations, safeguarding personal health information. | Compliance |
| A company uses end-to-end encryption for all internal communications, ensuring that only the sender and the intended recipient can read the contents of a message. | Confidentiality |
| An online banking application regularly verifies transactions and account updates to ensure that no unauthorized changes have been made to user accounts. | Integrity |
| Despite experiencing a significant spike in web traffic during a promotional event, an e-commerce platform remains fully operational, thanks to scalable cloud resources and DDoS protection measures. | Availability |

Trusted Infrastructure From Google

- **World's most popular websites:** Google runs and manages high traffic websites like Google Search and Youtube
- **Own Infrastructure:** Google has build a world class infrastructure for its use
- **Used by Google Cloud:** The same infrastructure is made available to us by Google Cloud



Trusted Infrastructure From Google - Advantages

- **Tailored Security:** Google custom-makes its security, making it super tough for hackers
- **Advanced Protection:** Google's own security tech is always a step ahead of hackers
- **Innovative Security Features:** Google keeps adding new security tricks to keep data safe
- **Reduced Vulnerabilities:** Google's unique systems have fewer weak spots for attacks
- **Rapid Response:** Google fixes security problems super fast because it controls everything



Enhanced Security Using 2 Step Verification (2SV)

- **2 Step Verification (2SV):** Add a 2nd step to verify user
 - **MFA:** Also called Multifactor authentication
 - **Online Banking Example:** Use a password and a code from your phone to access your bank account safely
 - **Google Cloud Example:** 2SV ensures that even if someone obtains your administrator's password, they wouldn't be able to access the account without the additional verification code
- **Make 2SV Mandatory:** For Google Cloud accounts
 - **Security keys:** A physical key inserted into a USB port
 - **Google Authenticator app:** Generates single-use 2SV codes
 - **Backup codes:** Generate backup verification codes and print them ahead of time
 - **Text message or phone call:** Receive 2SV codes via a text message or voice call



Exploring SecOps

- **DevOps:** Combines development and operations to speed up project delivery while enhancing teamwork and process efficiency
 - Communication
 - Automation
 - Quick Feedback
- **SecOps:** Layers security into this fast-paced setup
 - **Be Secure:** Ensures the project remains safe from threats without slowing down progress



Exploring DevOps and SecOps in Depth

| Aspect | DevOps | Example (DevOps) | SecOps | Example (SecOps) |
|----------|---|---|--|--|
| Focus | Speed up delivery | Automating code deployment for faster release | Ensures security throughout | Regularly checking code for security vulnerabilities |
| Teamwork | Collaboration between developers and operations | Teams meet daily to solve problems together | Includes security in the team | Security experts join planning meetings to integrate security from the start |
| Tools | Uses tools for efficiency in development and deployment | Continuous integration tools to test and merge code quickly | Employs tools for security monitoring and threat detection | Intrusion detection systems to alert on real-time threats |

Exploring DevOps and SecOps in Depth - Contd

| Aspect | DevOps | Example (DevOps) | SecOps | Example (SecOps) |
|----------------|---|---|--|--|
| Routine | Continuous integration and delivery for regular updates | Automatically updating software nightly | Constant security assessments to prevent threats | Regular security scans to identify and fix vulnerabilities |
| Culture | Promotes quick innovation and learning from failures | Encouraging rapid prototyping of new features | Prioritizes security awareness and practices | Conducting regular security training sessions for all team members |

Exploring Google Cloud's Trust Principles

- **Trust is key:** Would we share any confidential information to some one if we don't trust them?
- Why would customers trust Google to protect their data?
- A cloud provider should earn trust by:
 - **Clear Principles:** Clearly stating what are their principles
 - **Sharing information:** About how they protect data
 - **Sharing more information:** How do they handle requests from Governments for data?



7 Google Cloud's Trust Principles

- You own your data, not Google
- Google does not sell customer data to third parties
- Google Cloud does not use customer data for advertising
- All customer data is encrypted by default
- We guard against insider access to your data
- We never give any government entity "backdoor" access
- Our privacy practices are audited against international standards



Exploring Google Transparency Report

- Have you ever wondered how Google handles:
 - Governments request for data
 - Request to remove content from Google sites
 - Complaints against copyright violations
- **Google publishes** all these information in the transparency reports
 - End users know how it affects their privacy, security and access to information
- **Transparency** Reports:
<https://transparencyreport.google.com/?hl=en>



Third-party Audits for Google Cloud

- **Why do we do Certifications?**
 - **Prove our skills reliably:** We can prove to outside world, that our skills are verified by reputable 3rd party
- **Cloud Providers are the same**
 - **Auditing:** Go through the audit process to prove that everything is managed properly and securely



Third-party Audits for Google Cloud - Examples

- **Audit Examples:**

- **SOC 2:** Checks if cloud provider safely handles and protects customer data according to five key principles
- **ISO 27001:** Verifies cloud provider's system for securing & managing info. meets standards
- **PCI DSS:** Ensures cloud providers securely process and store credit card information to protect against fraud
- **HIPAA:** Confirms cloud services protect patient health information, keeping it confidential and secure

- **Compliance Resources:**

- *Compliance Reports Manager:* Audit reports, ..
- *Compliance resource center:* Guidance for understanding and navigating the compliance landscape (best .. , .. , ..)



Data Privacy and Data Residency Management

- **Strict Data Laws:** Different countries have strict laws related to handling of customer data
 - GDPR for Europe
 - California Consumer Privacy Act for US
- **Example GDPR:** Strict requirements like:
 - Data of the Europe's customer should stay within European union
 - Customer's data should be secured
 - Customer can correct data
- How can businesses adhere to these local laws while doing business internationally?
 - Google cloud can help you get started!



Data Privacy and Data Residency Management

- **Google Cloud makes it easy to get started:**
 - **Default Data Encryption:** Keeps data secure, both stored and during transfer
 - **Data Residency:** Choose data storage locations
 - **IAM Access Controls:** Limit data access based on roles
 - **Cloud Audit Logs:** Provides detailed records of data access
 - **Expertise and Resources:** Offers guidance on GDPR compliance through resources and expert advice:
 - <https://cloud.google.com/privacy/gdpr>
 - **Compliance Resources:**
 - *Compliance Reports Manager:* Audit reports, ..
 - *Compliance resource center:* Guidance for understanding and navigating the compliance landscape (best practices, white papers)



Exploring Google Cloud Security Offerings

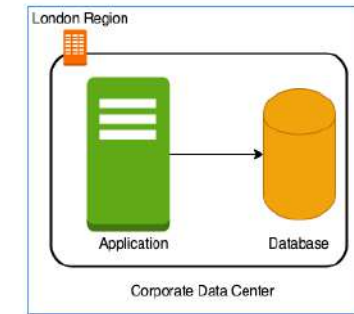
| Service | Description |
|----------------------------|---|
| KMS | Create and manage cryptographic keys (symmetric and asymmetric). Control their use in your applications and GCP Services |
| Secret Manager | Manage your database passwords, your API keys securely |
| Cloud Data Loss Prevention | Discover, classify, & mask sensitive data (like Credit Card numbers, SSNs, clear text passwords & Google Cloud credentials) Integrates with Cloud Storage, BigQuery, and Datastore Provides APIs that can be invoked from your applications |
| Cloud Armor | Protect your production apps (at run time) from denial of service and common web attacks (OWASP Top 10) like XSS (cross-site scripting) and SQL injection |

Exploring Google Cloud Security Offerings - 2

| Service | Description |
|----------------------------|--|
| Web Security Scanner | Identify vulnerabilities by running security tests. Examples: Cross-site scripting (XSS) MIXED_CONTENT,OUTDATED_LIBRARY, XSS |
| Binary Authorization | Ensure that only trusted container images are deployed to Google Cloud |
| Container Threat Detection | Detects container runtime attacks. Examples: Added binary executed |
| Security Command Center | Get a consolidated picture of security in Google Cloud (Security posture management) Discover misconfiguration and vulnerabilities (Built-in threat detection) Compliance monitoring (Review and export compliance reports. Check compliance of your resources with PCI-DSS 3.2.1, OWASP Top Ten, ..etc) |

Understanding Zero Trust Security Model

- **Traditional IT security model:** Security implemented at the network perimeter
 - Assumes everything inside can be trusted
- **Zero Trust** - "No person or device should be trusted by default, even if they are already inside an organization's network"
 - Strict identity authentication and authorization throughout the network
 - Resources might be secure even if attackers gain access to a network
 - **Simple Concept:** Every user, device, and component is considered untrusted at all times, regardless of whether they are inside or outside of an organization's network
- **Three Key Principles**
 - 1: Assume all network traffic is a threat, at all times
 - 2: Enforce least-privileged access
 - 3: Always monitor



Machine Learning in Google Cloud

Machine Learning - 10,000 Feet Overview

- **Traditional Programming:** Based on Rules
 - IF this DO that
 - Example: Predict price of a home
 - Design an algorithm taking all factors into consideration:
 - Location, Home size, Age, Condition, Market, Economy etc
- **Machine Learning:** Learning from Examples (NOT Rules)
 - Give millions of examples
 - Create a Model
 - Use the model to make predictions!
- **Challenges:**
 - No of examples needed
 - Availability of skilled personnel
 - Complexity in implementing MLOps

| Home size (Square Yds) | Age | Condition (1-10) | Price \$\$\$ |
|---------------------------|-----|---------------------|-----------------|
| 300 | 10 | 5 | XYZ |
| 200 | 15 | 9 | ABC |
| 250 | 1 | 10 | DEF |
| 150 | 2 | 34 | GHI |

ML in Google Cloud - Pre-Trained Models

- Use Pre-Built Models - Provided as APIs
- **Speech-to-Text API:** convert speech into text
- **Text-to-Speech API:** convert text into speech
- **Translation API:** Translate texts into more than one hundred languages
- **Natural Language API:** Derive insights from unstructured text
- **Cloud Vision API:** Recommended for generic usecases
 - Example: Identify if there is a cloud in the picture
 - Classify images into predefined categories
 - Detect objects and faces
 - Read printed words



Google Cloud

ML in Google Cloud - Custom Models

- **1: Simplify Building of Custom Models**
 - **AutoML:** Build custom models with minimum ML expertise and effort
 - **AutoML Vision:** Build custom models based on Images
 - Example: Identify the specific type of cloud
 - Provide examples - Example images and categorization
 - AutoML creates the model for you!
 - **AutoML Video Intelligence:** Add labels to Video
 - Streaming video analysis, Object detection and tracking
 - **AutoML Tables:** Automatically build models on structured data
- **2: Have Data Scientists build complex models**
 - **Frameworks:** TensorFlow, PyTorch, and scikit-learn
- **3: BigQuery ML: Build ML models using Queries**
 - Use data directly from BigQuery datasets (NO exports needed)
- **Vertex AI: Build & deploy ML models faster**
 - Custom tooling within a unified AI platform
 - Makes MLOps easy

| Home size (Square Yds) | Age | Condition (1-10) | Price \$\$\$ |
|---------------------------|-----|---------------------|-----------------|
| 300 | 10 | 5 | XYZ |
| 200 | 15 | 9 | ABC |
| 250 | 1 | 10 | DEF |
| 150 | 2 | 34 | GHI |

Faster ML in Google Cloud - TPUs

- Do you have models that train for weeks or months?
 - Go for Tensor Processing Units (TPUs)
- Fine-tuned for **running ML workloads**
- **20-30X faster** than traditional approaches
- Helps you quickly iterate on your ML solutions
- Supported in Google Compute Engine, Google Kubernetes Engine and AI platform
- **Custom AI Platform Deep Learning VM Image** is available
- **Preemptible Cloud TPUs** are also available



Google Cloud

Machine Learning - Data is the Key

- **Machine Learning:** Learning from Examples (NOT Rules)
- (IMHO) Most important factor in a successful ML implementation is examples (or data, as it is often called)
 - You need millions of examples (or data points)
 - The data has to be accurate
 - Should NOT have bias
 - Should NOT have errors
- A number of enterprises face challenges in getting clean data

| Home size (Square Yds) | Age | Condition (1-10) | Price \$\$\$ |
|---------------------------|-----|---------------------|-----------------|
| 300 | 10 | 5 | XYZ |
| 200 | 15 | 9 | ABC |
| 250 | 1 | 10 | DEF |
| 150 | 2 | 34 | GHI |

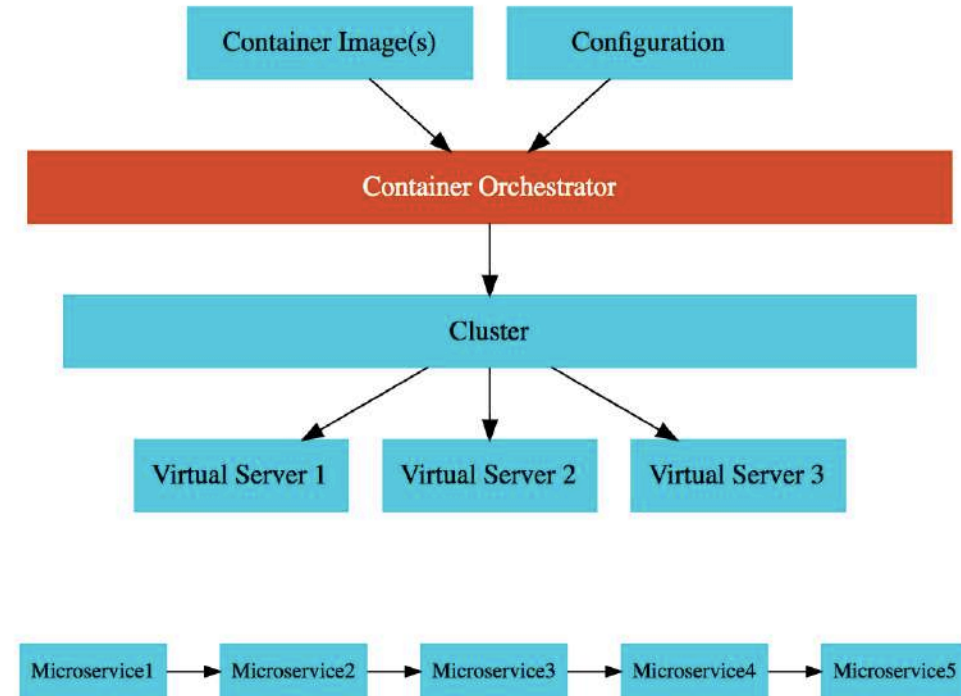
Machine Learning Scenarios

| Scenario | Solution |
|--|-------------------------------------|
| Translate from one spoken language to another | Prebuilt Model -Translation API |
| Convert speech to text | Prebuilt Model - Speech-to-Text API |
| Generic identification of objects in an image | Prebuilt Model - Cloud Vision API |
| Identify the type of cloud or a machine part based on an Image | AutoML Vision |
| Simplify implementation of MLOps | Vertex AI |

Cloud Native

What is Cloud Native?

- I would love to say that there is ONE definition for Cloud Native
 - HOWEVER there isn't one
- **(MY DEFINITION):** Cloud Native Architectures help you get the best value from the Cloud
 - **GOOGLE CLOUD DEFINITION:** Designed from the ground up to take advantage of the elasticity and distributed nature of the cloud
- **Goal:** Increase **software delivery velocity** and increase **service reliability** while increasing **collaboration among stakeholders**



Exploring Cloud Native Pillars

- **Four Cloud Native Pillars**

- **1: Microservices**

- Fix issues and deliver new features quickly
 - Without impacting other services

- **2: Containers**

- Portable - build once, run anywhere
 - Simplified consistent deployments
 - Lightweight (Faster deployments than VMs)

- **3: Container Orchestration**

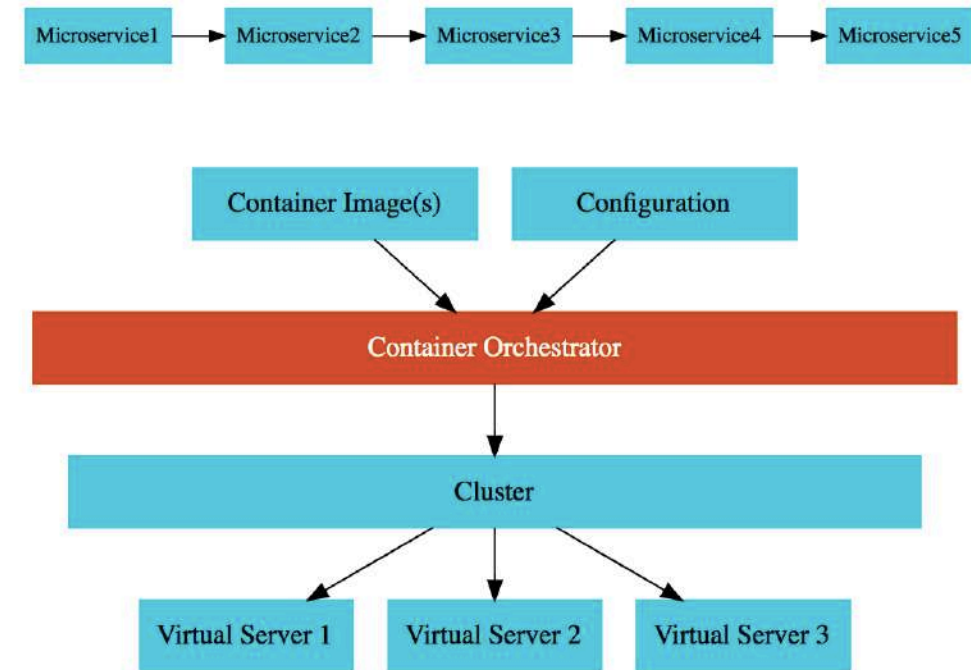
- Kubernetes (GKE) - Auto Scaling, Load Balancing, Self Healing, Zero Downtime Deployment etc

- **4: DevOps (Dev + Ops, CI/CD, IaC)**

- Increased automation of processes

- **Examples of NOT Cloud Native**

- Using VMs, Manual deployments, Creating infrastructure manually



Modern Architectures - 3 Container Compute Examples

| Service | Description | Type |
|--------------------------|--|----------------------|
| Cloud Run | Develop and deploy highly scalable containerized applications. Does NOT need a cluster! | CaaS (Serverless) |
| Google Kubernetes Engine | Orchestrate containerized microservices on Kubernetes Needs advanced cluster configuration and monitoring | CaaS |
| Anthos | Manage Kubernetes Clusters in Multi-cloud and On-premises | Hybrid Cloud |

Modern Architectures - Serverless Examples

| Service | Description |
|-----------------------------|---|
| Cloud Functions | Serverless compute for event-driven apps. Execute functions (or code) in response to events. |
| Cloud Run | Run isolated containers, without orchestration (Serverless) You DO NOT need to provision and manage VMs. Start containers in seconds. |
| Cloud Firestore (Datastore) | Apps needing quickly evolving structure (schema-less). Serverless transactional document DB supporting mobile & web apps. Small to medium DBs (0 - few TBs) |
| Cloud Dataflow | Serverless Stream and Batch processing using Apache Beam (open-source) |
| Cloud Pub/Sub | Realtime Messaging in the cloud. Pay for number of messages. |
| BigQuery | Relational OLAP, Data warehousing & BigData workloads. Pay for data stored and queries executed. |

Choosing Region(s) and Zone(s)

- **1: Compliance** - Adhere to regulations & standards
 - Store data in right region(s) based on the regulations
 - Some countries don't allow their citizens data to be stored in other countries
 - Evaluate compliance for each region where you are storing data
- **2: Latency and Performance** - Be near to users or on-premises (based on your use case)
 - Use **Premium Tier** for optimum network performance
 - To keep costs low, use Standard Tier (traffic over internet)
 - Example: HPC workloads need low latency between VMs
 - Greater distance between VMs => Greater network latency
- **3: Fault Tolerance:** Distribute apps across Region(s)
 - Even if a zone or region is not available, apps are NOT impacted
- **4: Pricing:** Pricing varies from region to region as well
- And a lot of other factors



Digital Transformation

What has changed in last decade or so?

- How consumers make purchase decisions? (**Social**)
- How we do things? (**Mobile**)
- How much data we have? (**Big Data**)
 - How much intelligence we can get? (**AI/ML**)
- How much access startups have to technology at scale? (**Cloud**)



Enterprises have to adapt (or get disrupted)



- **Enterprises can ADAPT by:**
 - Providing awesome (omni-channel - social, mobile) customer experiences
 - Getting intelligence from data (Big Data, AI/ML)
 - Example: Personalize consumer offerings
 - Enabling themselves to make changes faster
 - Cultural change from "traditional Datacenter, SDLC, manual IT Ops" to "Cloud, Containers, DevOps/SRE, Automation"
- **Digital Transformation:** Using modern technologies to create (or modify) business processes & customer experiences by innovating with technology and team culture
 - Focus on WHY (NOT HOW)
 - Increase pace of change
 - Revenue Growth
 - Cost Savings
 - Higher customer engagement/retention

Cloud - Enabler for Digital Transformation

- Cloud can **ENABLE** Digital Transformations
 - Lower cost
 - Reduced responsibilities
 - Higher capabilities
 - Increased speed to market
- **BUT needs a change** in skills, mindset and culture
 - Modern Architectures (Microservices, Serverless, Containers, Kubernetes)
 - More Agile Processes (DevOps, SRE)
 - Right Talent
 - Right Culture (of data driven experimentation and innovation)



Cloud Mindset

| Factor | Data Center | Cloud |
|-------------------------|--------------------|---|
| Infrastructure | Buy | Rent |
| Planning | Ahead of time | Provision when you need it |
| Deployment | VMs | PaaS or Containers or Serverless |
| Team | Specialized skills | T-shaped skills |
| Releases | Manual | CI/CD with flexible release options (Canary, A/B Testing,) |
| Infrastructure Creation | Manual | Infrastructure as Code |
| Attitude | Avoid Failures | Move Fast by Reducing Cost of Failure (Automation of testing, releases, infrastructure creation and monitoring) |

Google Cloud Adoption Framework

- Streamlined framework for adopting the cloud
 - **Four themes**
 - **Learn:** How do you build the right skills?
 - **Lead:** How do you structure teams so that they are cross-functional, collaborative, and self-motivated?
 - **Scale:** How do you reduce operational overhead and automate manual processes? (provisioning and scaling infrastructure, application releases, monitoring)
 - **Secure:** How to protect from unauthorized and inappropriate access? (controls, strategies and technology)
 - **Three phases:**
 - **Tactical:** Move to cloud with minimum changes (to people, process and technology)
 - Use IaaS - Mainly for cost savings
 - **Strategic:** Make some degree of change (to people, process and technology) in isolated part of an enterprise (early success stories)
 - Harness additional value of cloud
 - **Transformational:** Fully invested in Cloud
 - Cloud-first, fully-automated, cross-functional feature-teams
 - Driven by data and intelligence, Adopting DevOps and SRE



1: Infrastructure Modernization

- **Lift and shift** - Move AS-IS to Google Cloud Infrastructure
 - **Examples:**
 - **Virtual desktop solutions:** Make use of virtual desktop solutions on Google Cloud
 - Backup and disaster recovery (Simple starting step to cloud)
 - **VMware as a service:**
 - **Google Cloud VMware Engine:** Lift and Shift VMware infrastructure to Google Cloud
 - **Bare Metal Solution:** Move specialized workloads (SAP HANA, Oracle databases, ..) that need really high performance
 - **Migrate for Compute Engine:** Migrate VMs and VM storage to GCE
- **Benefits:**
 - Lower costs
 - Reduced focus on infrastructure
- BUT you are not yet making use of all the benefits of being in the cloud!



2: Application Modernization

- Migrate to PaaS or Serverless offerings:
 - **Containerization**
 - **Container Orchestration** (GKE, Anthos)
 - **Migrate for Anthos and GKE:** Modernize apps by moving from VMs to containers
 - Make use of cloud databases and data warehouses
- Use DevOps and SRE practices (Cloud Build, Cloud Monitoring, ..)
 - Move Fast by Reducing Cost of Failure
- Benefits:
 - Managed services simplify application maintenance and lifecycle
 - Managed Services have good integration with Cloud Build, Cloud Monitoring and Cloud Logging
 - App Engine, GKE, Cloud Run support multiple release approaches
 - Additional innovation provided by managed services
 - BigQuery ML: Create and execute ML models directly in BigQuery using standard SQL queries



Compute Engine



Cloud Functions



Kubernetes Engine



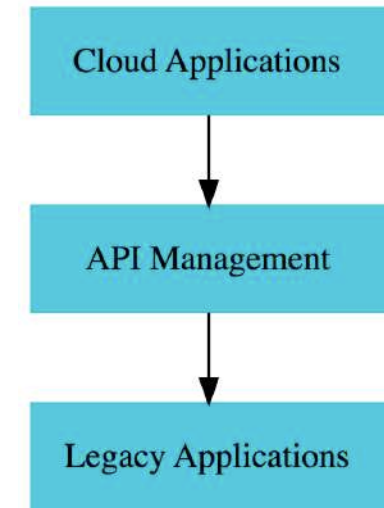
Cloud Build



Cloud SQL

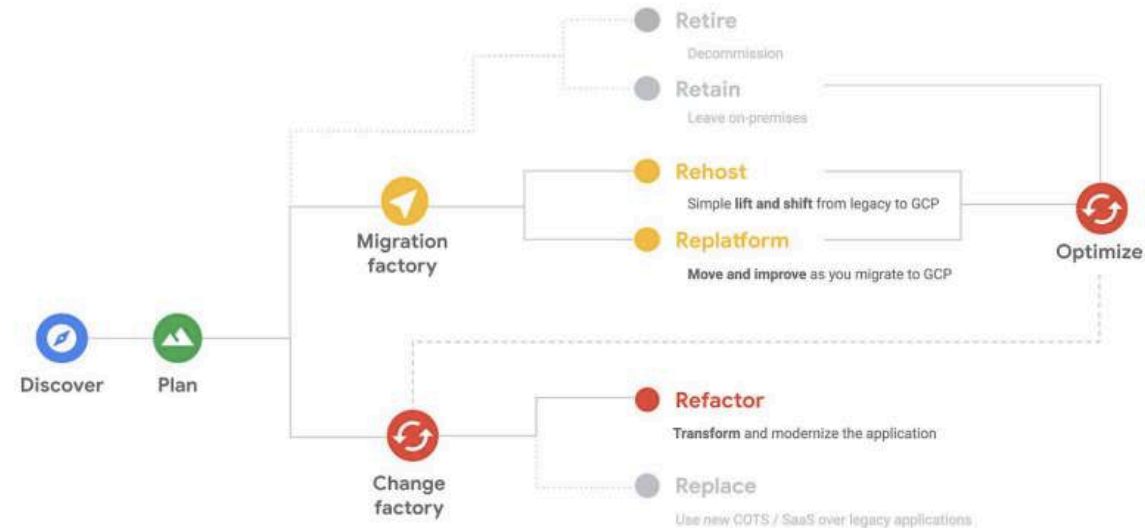
3: Business Platform Modernization

- What if you **DONT** want to move legacy system to Cloud?
- What if you want to enable external developers and partners to build apps for you?
- **Build APIs** around legacy code to simplify integration
- **Managed Services:** Apigee API Management, Cloud Endpoints
 - Design, Secure, Publish, Analyze, Monitor and Monetize APIs
- **Advantages:**
 - Integrate with legacy applications
 - Open new business channels
 - Create an ecosystem of developers and partners



Moving an Application to the Cloud

- Choice varies from app to app
 - Retire, Retain, Rehost, Replatform, Refactor, Replace
 - Rehost - Lift and Shift
 - Replatform - Improve
 - Refactor - Transform



Source - CIO Guide to Application Modernization

Cloud Migration - Few more examples

| Problem | Solution |
|--|--|
| Quickly retire a data center | Start with Infrastructure Modernization |
| Very slow release processes | Automate as much testing as possible. Make use of CI/CD (Cloud Build). |
| Bugs due to differences between environments | Migrate to Containers. Use Infrastructure as Code. |
| Cannot move a legacy app to the cloud but other apps need it | Build an API around legacy app |
| Huge volumes of analytical data in being stored in a relational database | Move it to BigQuery or archive it with Cloud Storage |
| Difficult to maintain and scale transactional relational database | Migrate to a managed relational database |

Cloud Migration Scenarios

| Scenario | Strategy |
|---|------------|
| Our company has an old application that nobody uses anymore. It's costing us money to keep it running, so we want to shut it down. | Retire |
| We have a legacy system that's critical for our day-to-day operations, but it's not yet suitable for cloud due to compliance issues. We'll keep it on-premises for now. | Retain |
| We want to move our existing web application to the cloud as quickly and with as little change as possible to benefit from the cloud's scalability and cost savings. | Rehost |
| Our application is a good candidate for the cloud, but we want to update its underlying database to a managed cloud service to reduce administrative overhead. | Replatform |
| We see the potential for significant improvements in our application by adopting cloud-native features, so we're going to redesign it to fully leverage the cloud environment. | Refactor |
| Our current customer relationship management (CRM) system is outdated. We've decided to switch to a cloud-based CRM solution rather than moving our existing system to the cloud. | Replace |

Scaling Operations with Google Cloud

Exploring Google Cloud Customer Care

- How can you get help from Google about your Google Cloud implementations?
 - **Google Cloud Customer Care**
- Things You Can Get Help With:
 - **Google Cloud Skills Boost:** Training credits for your team
 - **Event Management Service:** Support planned peak events, such as a product launch or major sales event
 - **Technical Account Manager (TAM):** Advisors that focus on your operational rigor, platform health, & architectural stability
 - **Customer Aware Support:** Support experts who understand your implementations so that you can get quick support
 - **Operational Health Reviews:** Reviews to measure your progress and address blockers



Exploring Google Cloud Customer Care Options

| Feature | Standard Support | Enhanced Support | Premium Support |
|--|------------------------------------|--|--|
| Recommended for | Development workloads/environments | Production workloads | Enterprises with critical workloads |
| Pricing | \$ | \$\$ | \$\$\$\$ |
| Service times | 8/5 - high impact issues | 24/7 - high and critical impact issues | 24/7 - high and critical impact issues |
| Technical account management, Event management service, Customer aware support | No | No | Yes |
| Google cloud skills boost | No | No | Yes |

Exploring Value Added Services

- **Value-Add Services:** Additional purchase for Enhanced and Premium Support customers:
 - **Technical Account Advisor Service:** Enhanced oversight of your cloud experience, combining proactive guidance with regular service reviews
 - **Planned Event Support:** Complete cycle for planned events: Architecture Essentials Review > Accelerated response time (15 mins) for P1 issues > Performance summary report (review for improvement opportunities)
 - **Assured Support:** Reach your compliance objectives including FedRAMP High, ..
 - **Mission Critical Services:** Fastest possible impact mitigation response



Customer Care Support Case Lifecycle

| Status | Description |
|---------------------------------|--|
| New | The case is not assigned yet. |
| Assigned | The case is assigned to a specialist |
| In progress Cloud Customer Care | Customer Care specialists are working on the case |
| In progress Google engineering | Google product engineers are investigating the case |
| In progress Google other | Another Google team is investigating the case |
| Waiting on customer response | We need more information from you |
| Waiting on customer action | We need you to do something |
| Solution offered | Solution is offered. The customer can reopen the case if the offered solution is insufficient. |
| Closed | The case is resolved. Reopen within 15 days if needed. |

Working with Customer Care - Best practices

- **Single Point Tracking:** Create one support case per issue
- **Right Priority:** Set clear priority
 - P1 - Critical Impact — Service Unusable in Production
 - P2 - High Impact — Service Use Severely Impaired
 - P3 - Medium Impact — Service Use Partially Impaired
 - P4 - Low Impact — Service Fully Usable
- **Clear Description:** Include as many details as possible
 - Time, Product, Location, Identifiers, Description..
 - **Route cases to the required time zone:** Include something like "Please route this to Pacific time zone (GMT-8)" in description
- **Escalate when needed:** When business impact increases or breakdown of the resolution process
 - Example: You haven't received an update in the agreed upon time period



What is Cloud Sustainability?

- Data centers have significant environment implications:
 - Massive Energy consumption
 - Lot of hardware wastes
 - CO2 emission etc
- **Sustainability:** "meeting the needs of the present without compromising the ability of future generations to meet their own needs." (United Nations Definition)
- **Cloud sustainability:** Sustainable operation and delivery of cloud services
 - Enabling sustainable consumption and use of cloud services by organizations



How is Google Cloud doing?

- **Google's Mission:** Google is on a mission to help customers to reduce their carbon foot print
- **More Efficient:** Google owned and operated data centers are more than 1.5 times energy efficient compared to enterprise data centers (Estimates)
- **And Improving:** Google delivers 3x more compute power for same amount of electrical power compared to 5 years ago
- **Certified:** ISO certification for energy management
- **Transparent:** <https://sustainability.google/reports/>



Google Cloud Sustainability Tools

- **Google Cloud Carbon Footprint:** Measure and understand your cloud emissions:
 - Granular breakdown of each customer's cloud emissions by usage (project, service, GCP region)
 - Can be exported to BigQuery and build custom dashboards to better visualize and track emissions
- **Choose LOW CO2 cloud region:** Publishes carbon data for each cloud region
 - Low CO2 indicators in location selectors
- **Google Cloud Region Picker:** Pick a Google Cloud region considering approximated carbon footprint, price and latency
- **Unattended Project Recommender:** Discover, reclaim, and shut down unattended projects



Cost Management

Total Cost of Ownership (TCO)

- How do you estimate the cost savings of moving to cloud?
 - Take **Total Cost of Ownership** into account
- **Total Cost of Ownership:**
 - Infrastructure Costs
 - Procuring Servers, Databases, Storage, Networking ..
 - Infrastructure maintenance costs
 - IT personnel costs
 - Software costs
 - Electricity costs
 - ...
- Compare Apples to Apples!



Google Cloud

Consumption-based vs Fixed-price Pricing Models

In 28
Minutes

- **Consumption-based** - You are billed for only what you use
 - Example: Cloud Functions - You pay for no of invocations!
- **Fixed-price** - You are billed for instances irrespective of whether they are used or not
 - **Example:** You provision a VM instance
 - You pay for its lifetime irrespective of whether you use it or NOT
 - **Example:** You provision a GKE cluster
 - You are billed irrespective of whether you use it or not



Google Cloud

Expenditure Models: CapEx vs OpEx

- **Capital Expenditure (CapEx):** Money spent to buy infrastructure
 - Additional cost to maintain infrastructure with time
 - You might need a team to manage the infrastructure
 - **Example:** Deploying your own data center with physical servers
 - **Example:** Purchasing Committed use discounts
 - **Example:** Leasing Software
- **Operational Expenditure (OpEx):** Money spent to use a service or a product
 - **Zero upfront costs**
 - You Pay for services as you use them (Pay-as-you-go model)
 - **Example:** Provisioning VMs as you need them
 - **Example:** Using Cloud Functions and paying for invocations



Google Cloud

How is Cost Decided?

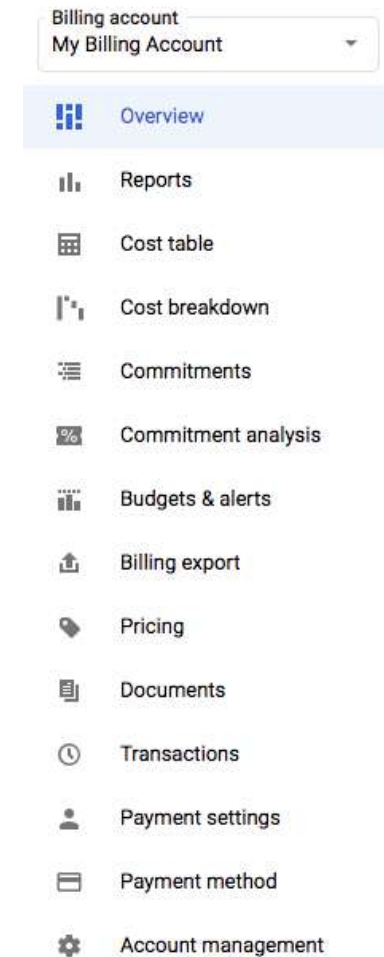
| Factor | Details |
|---------------------------------|---|
| Resource type and configuration | How much memory? How much CPU? Which access tier? |
| Usage meters | How long was your VM running for? How much ingress and How much egress? How many invocations of an Cloud function? |
| Which Region? | Price varies from Region to Region |
| Data transfer | Ingress and Egress Inbound data from on-premises to GCP is free Outbound data from GCP to On-Premises is NOT free Egress to the same Google Cloud zone when using the internal IP addresses of the resources is free |
| Reserved or Not | Some services offer reservations ahead of time |

Pricing Calculator

- **Estimating** the cost of a Google Cloud solution is **NOT** easy
- You would need to take a **number of factors** into account
- How do you estimate the cost of your GCP solution?
 - Use **Google Cloud Pricing Calculator**
- Estimates for **40+ Services**:
 - Compute Engine
 - Google Kubernetes Engine
 - Cloud Run
 - App Engine
 - Cloud Storage
 - etc
- **(REMEMBER) These are Estimates! (NOT binding on GCP)**

GCP Cost Management

- **Cost Management:** Tools for monitoring, controlling, and optimizing your costs
 - **Cost Billing Reports:** 10,000 feet overview of usage costs
 - Analyze Trends by Project, Service, Location, Labels etc..
 - **Cost Table report:** Detailed view
 - Dynamically filter, sort and group various line items
 - **Cost breakdown:** Base usage cost, credits, adjustments and taxes
 - **Budgets and alerts:** Set budgets and get alerted by email or Pub/Sub
 - **Commitments:** Manage and analyze committed use discounts
 - **Enable committed use discount sharing** to share discounts across projects
 - **BigQuery Export:** Sends billing data to a BigQuery data set:
 - **Do your own analysis with custom dashboards** - Export to BigQuery and analyze using Data Studio
 - **Account management:** Manage projects linked to this billing account
 - **Other features:** Transactions & Payment method



Understanding Google Cloud Quotas

- How to prevent unforeseen spikes in usage and overloaded services?
- How to avoid unexpected bills from using expensive resources?
- **Cloud Quotas:** Restrict how much of a particular shared Google Cloud resource that you can use
 - Most limits are applied per project



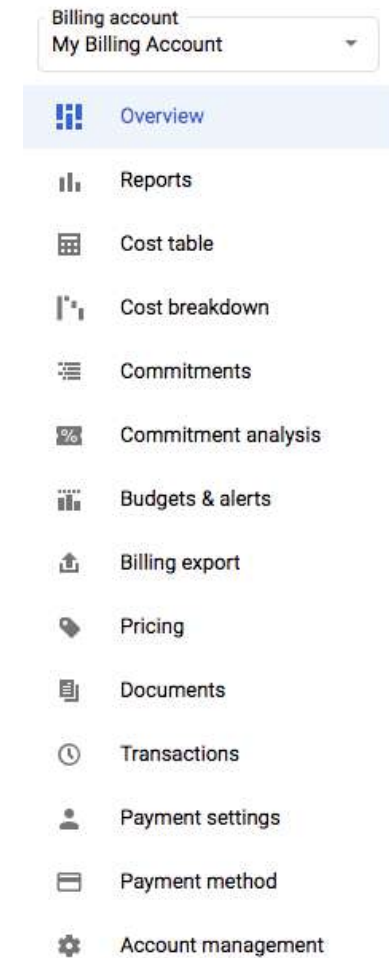
Understanding Google Cloud Quota Types

- **Rate quotas:** Limit the number of requests to an API or service per time interval
 - **Example:** Maximum number of requests to Compute Engine API per minute (Ex: 1500)
- **Allocation quotas:** Restrict the use of resources that don't have a rate of usage
 - **Example:** No of VMs used by your project at a given time
- **Concurrent quotas:** Restrict the total number of concurrent operations in flight at any given time
 - **Example:** Concurrent global operations per project - Limits the total number of concurrent global operations for a project



Managing Costs - Best Practices

- **Group resources** based on cost ownership
 - Folders, Projects, Labels etc.
- **Regular cost reviews** (at least weekly)
 - CapEx (Ahead of time planning) -> OpEx (regular reviews)
- **Estimate costs** before you deploy (Pricing Calculator)
- **Use Cost Management features**
 - Cost Table reports, Budgets and Cost alerts etc.
- **Others:**
 - Stop Resources when you don't need them
 - Use Managed Services (PaaS >>> IaaS)
 - Reserve VMs for 1 or 3 years (Committed use discounts)
 - Use Preemptible VMs for fault tolerant non-critical workloads
 - Involve all teams - executive, management, business, technology & finance



Cost Management - Scenarios

| Scenario | Practice |
|---|---|
| I want to understand how much I could save by moving our on-premises infrastructure to the cloud. | Compare Total Cost of Ownership (TCO) |
| I need to budget for a new project and want to estimate cloud costs accurately. | Use Pricing Calculator |
| I want to monitor and control our cloud spending to avoid unexpected charges. | Cost Management Tools: Use GCP's Cost Management tools - Cost Billing Reports, Cost Table report, Budgets and alerts.. |
| Our team is working on several projects, and I need to allocate cloud costs accurately to each. | Organize resources using projects. Use labels for easy tracking. Give ownership at project level. Review Regularly. |

Quick Review

Basic Compute Services - Google Cloud

| GCP Service Name | Description |
|--------------------------------|--|
| GCE or Compute Engine | Windows or Linux VMs (IaaS) Use VMs when you need control over OS OR you want to run custom software |
| Preemptible VMs | Short lived VMs for non time-critical workloads |
| Sole-tenant Nodes | Dedicated physical servers |
| VMware Engine | Run VMware workloads in Google Cloud |
| Managed Instance Groups | Create multiple Compute Engine VMs |
| Cloud Load Balancing | Balance load to multiple instances of an application or a service Usually considered as networking solution |

Managed Compute Services

| GCP Service Name | Description |
|--------------------------|---|
| App Engine | PaaS. Deploy web apps and RESTful APIs quickly. |
| Cloud Run | Run isolated containers, without orchestration (Serverless) You DO NOT need to provision and manage VMs. Start containers in seconds. Knative compatible. |
| GKE or Kubernetes Engine | Managed Kubernetes Service. Provides container orchestration. |
| Cloud Functions | Serverless compute for event-driven apps |
| Anthos | Manage Kubernetes Clusters in Multi-cloud and On-premises |
| Firebase | Google's mobile platform . Build Apps for iOS, Android, the web, C++, and Unity. |

Storage

| GCP Service Name | Description |
|------------------|--|
| Persistent Disk | Block Storage for your VMs |
| Local SSD | Local ephemeral block storage for your VMs |
| Cloud Filestore | File shares in the cloud |
| Cloud Storage | Object storage in the cloud |

Databases - Managed Services

| GCP Service Name | Description |
|-----------------------------|---|
| Cloud SQL | Regional Relational OLTP database (MySQL, PostgreSQL, SQL server) |
| Cloud Spanner | Global Relational OLTP database. Unlimited scale and 99.999% availability for global applications with horizontal scaling. |
| Cloud Firestore (Datastore) | Apps needing quickly evolving structure (schema-less). Serverless transactional document DB supporting mobile & web apps. Small to medium DBs (0 - few TBs) |
| Cloud BigTable | Large databases(10 TB - PBs). Streaming (IOT), analytical & operational workloads. NOT serverless. |
| Cloud Memorystore | In memory databases/cache. Applications needing microsecond responses |

Streams, Analytics, Big Data & .. - Managed Services

| GCP Service Name | Description |
|-------------------|--|
| Cloud Pub/Sub | Realtime Messaging in the cloud |
| BigQuery | Relational OLAP databases. Datawarehousing & BigData workloads. |
| BigQuery ML | Simplified Machine Learning using data in BigQuery |
| Cloud Dataflow | Serverless Stream and Batch processing using Apache Beam (open-source) |
| Cloud Dataproc | Managed Service for Spark and Hadoop. Not serverless (needs cluster management). |
| Cloud Data Fusion | Visually manage your data pipelines |
| Data Studio | Visualize data |
| Looker | Enterprise Business Intelligence |

Migration - Managed Services

| GCP Service Name | Description |
|--------------------------------|---|
| Database Migration Service | Migrate to Cloud SQL |
| Storage Transfer Service | Online Transfer to Cloud Storage |
| Transfer Appliance | Physical transfer using an appliance |
| Migrate for Compute Engine | Migrate VMs and VM storage to GCE From VMware, Microsoft Azure, Amazon EC2 ... |
| Migrate for Anthos | Migrate VMs to GKE containers |
| BigQuery Data Transfer Service | Migrate your analytics data |

Get Ready

Cloud Digital Leader - Certification Resources

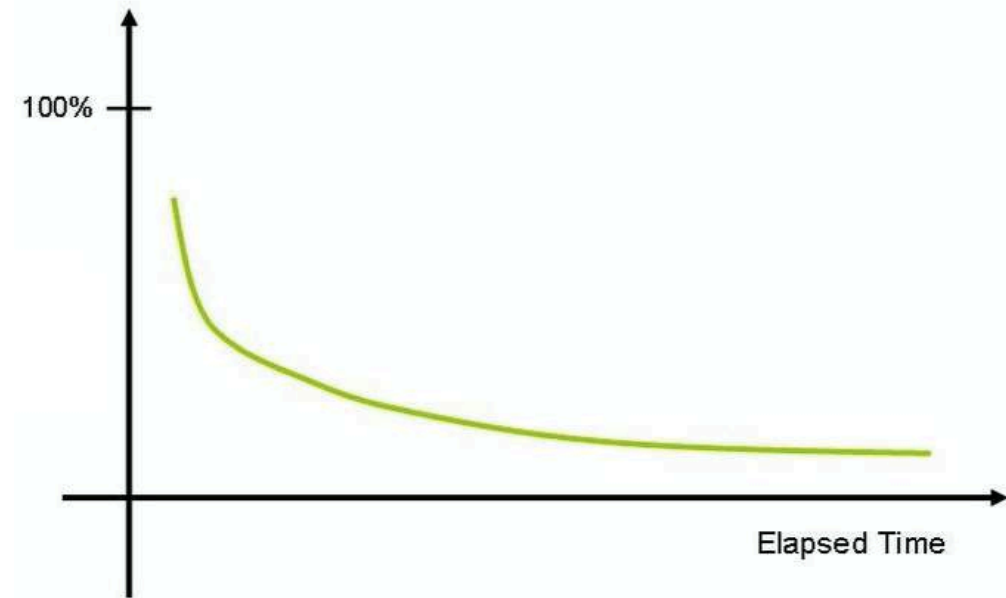
| Title | Link |
|----------------------|---|
| Home Page | https://cloud.google.com/certification/cloud-digital-leader |
| Exam Guide | https://cloud.google.com/certification/guides/cloud-digital-leader |
| Sample Questions | https://cloud.google.com/certification/sample-questions/cloud-digital-leader OR NEW LINK |
| Registering For Exam | https://support.google.com/cloud-certification/#topic=9433215 |

Cloud Digital Leader - Certification Exam

- **50 questions and 90 Minutes**
 - **No penalty** for wrong answers
 - **Questions:**
 - Type 1 : Multiple Choice - 4 options and 1 right answer
 - Type 2 : Multiple Select - 5 options and 2 right answers
 - **Result immediately shown** after exam completion
 - Email (a couple of days later)
- **My Recommendations:**
 - Read the **entire question**
 - Identify and write down the **key parts of the question**
 - More than sufficient time
 - **Flag questions** for future consideration (Review before final submission)
 - **TIP: Answer by Elimination!**

Get Ready For Your Exam

- How do you improve your chances of remembering things for the exam?
 - **1:** Review the presentation
 - **2:** Watch videos again at 2X speed
 - **3:** Use the Flash Cards (NEW!)
 - Link in the next lecture
 - Would love to get your feedback



You are all set!

Let's clap for you!

- You have a lot of patience! **Congratulations**
- You have put your best foot forward to be an Google Cloud Digital Leader
- Make sure you prepare well
- Good Luck!

Do Not Forget!

- Recommend the course to your friends!
 - Do not forget to review!
- **Your Success = My Success**
 - Share your success story with us on LinkedIn (Tag - in28minutes)
 - Share your success story and lessons learnt in Q&A with other learners!

What Next?

FASTEST ROADMAPS

in28minutes.com



In28
Minutes



Google Cloud
Certifications



Azure
Certifications



AWS
Certifications



DevOps



Java Full Stack



Java Microservices

