In28 Minutes

Google Cloud Digital Leader

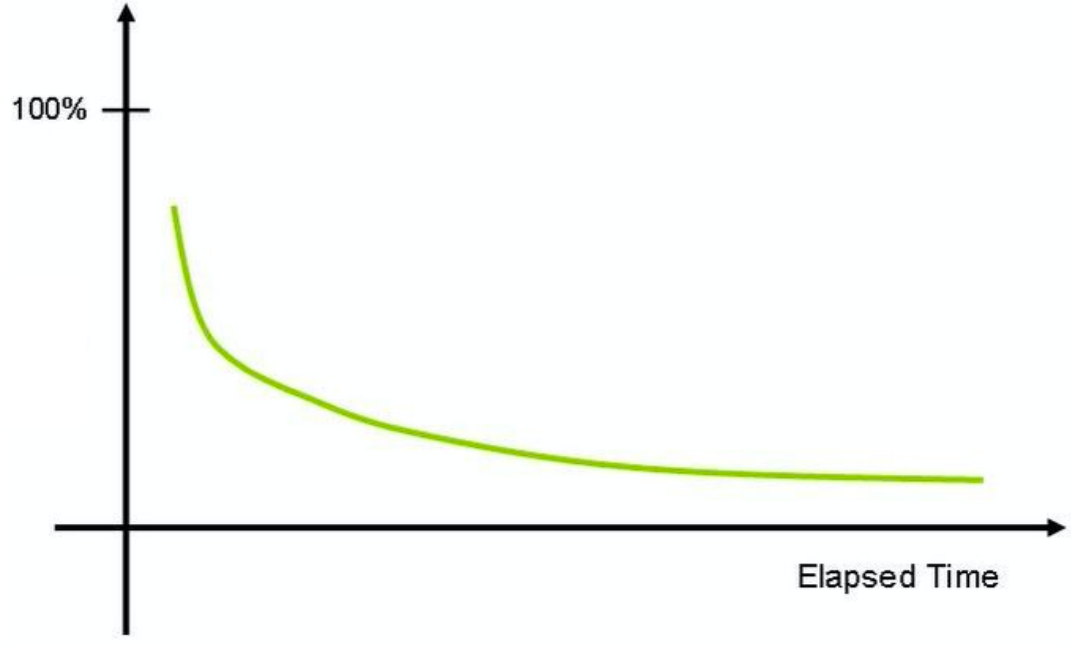
Getting Started

- 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?

In28 Minutes

- **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

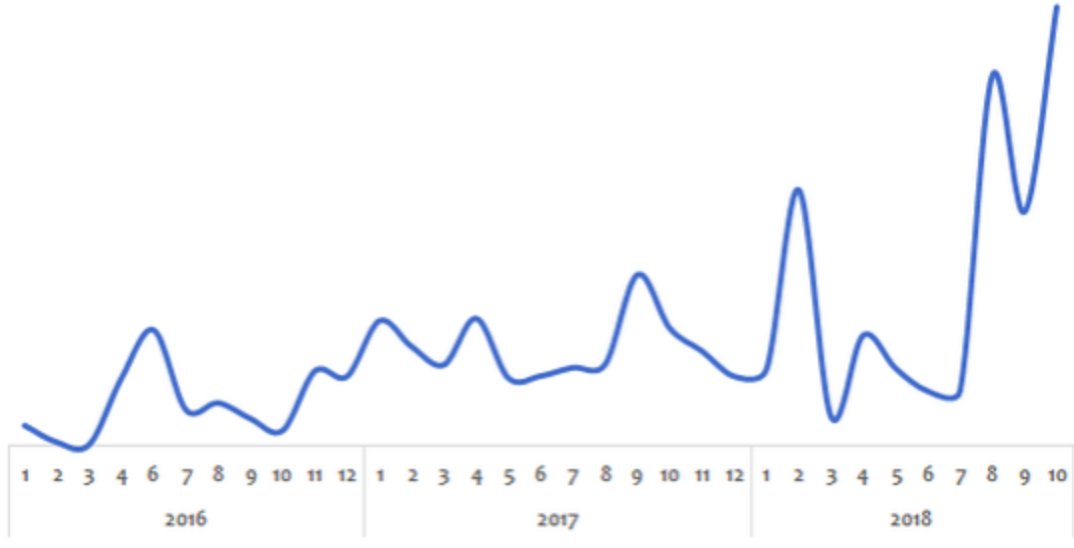
In28 Minutes

- 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!

GCP - Getting started

Before the Cloud - Example 1 - Online Shopping App

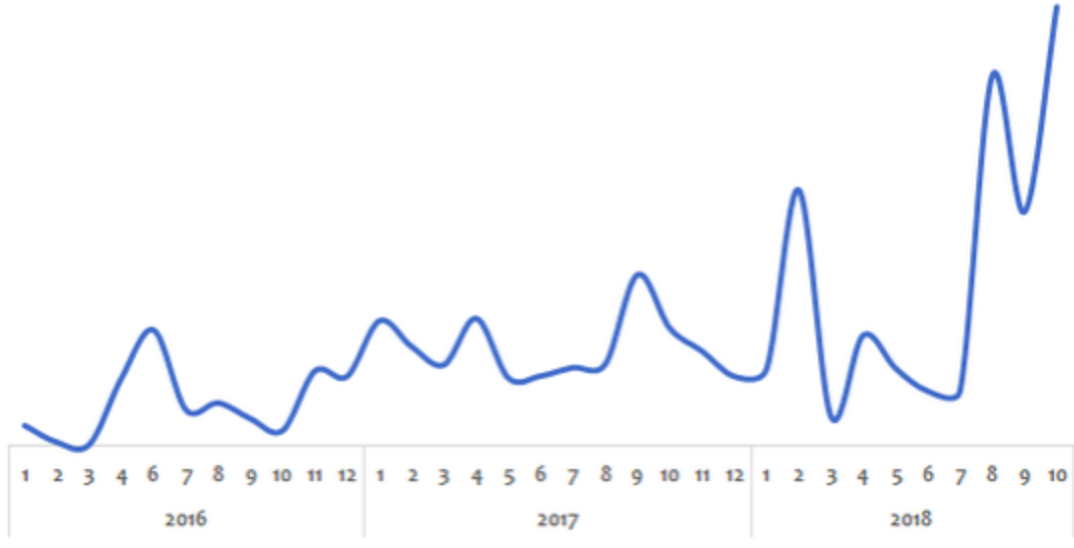
In 28 Minutes



- 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

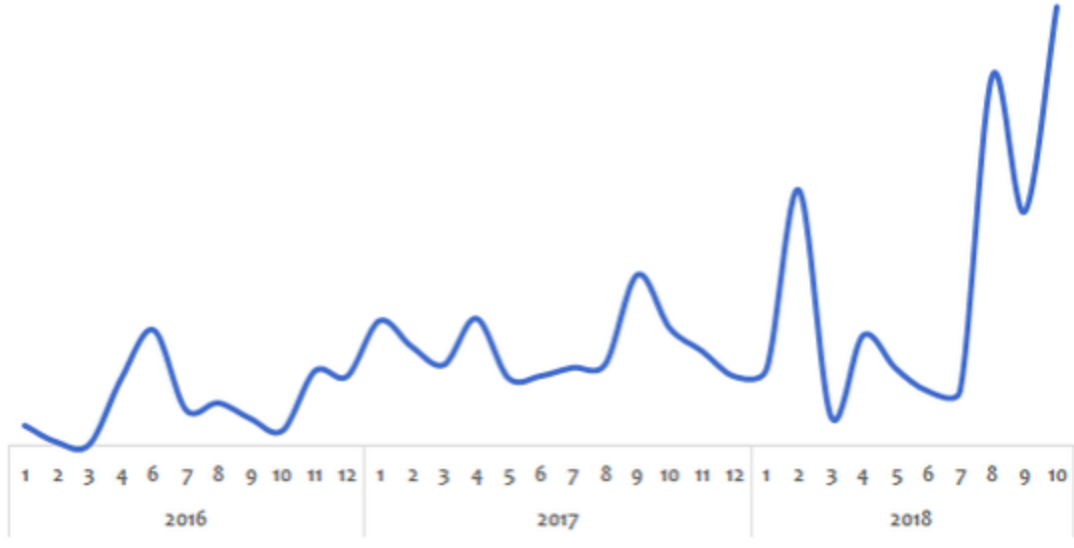
In28 Minutes



- 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

In28 Minutes



- 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

In28 Minutes

- 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

In28 Minutes

- 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)

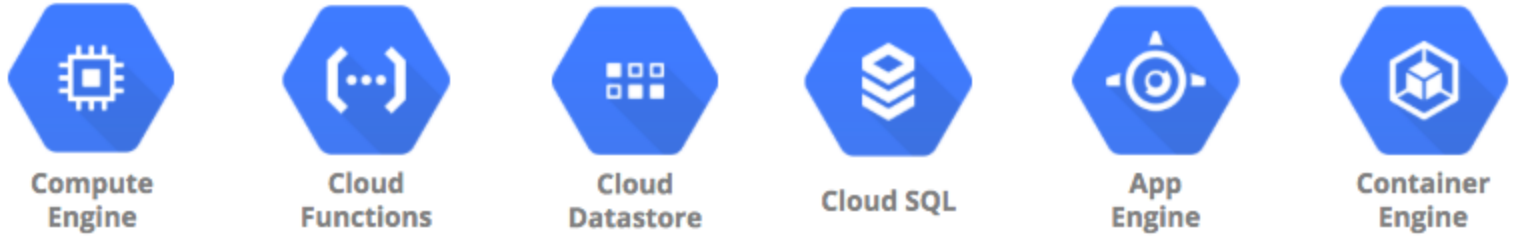
In28 Minutes

- **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!

In28 Minutes



- 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

In28 Minutes

- Create GCP Account

Regions and Zones

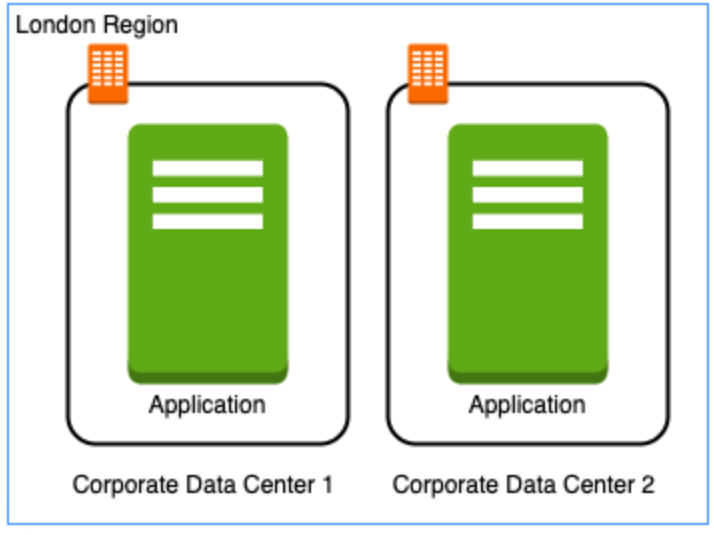
Regions and Zones

In28 Minutes

- 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**)

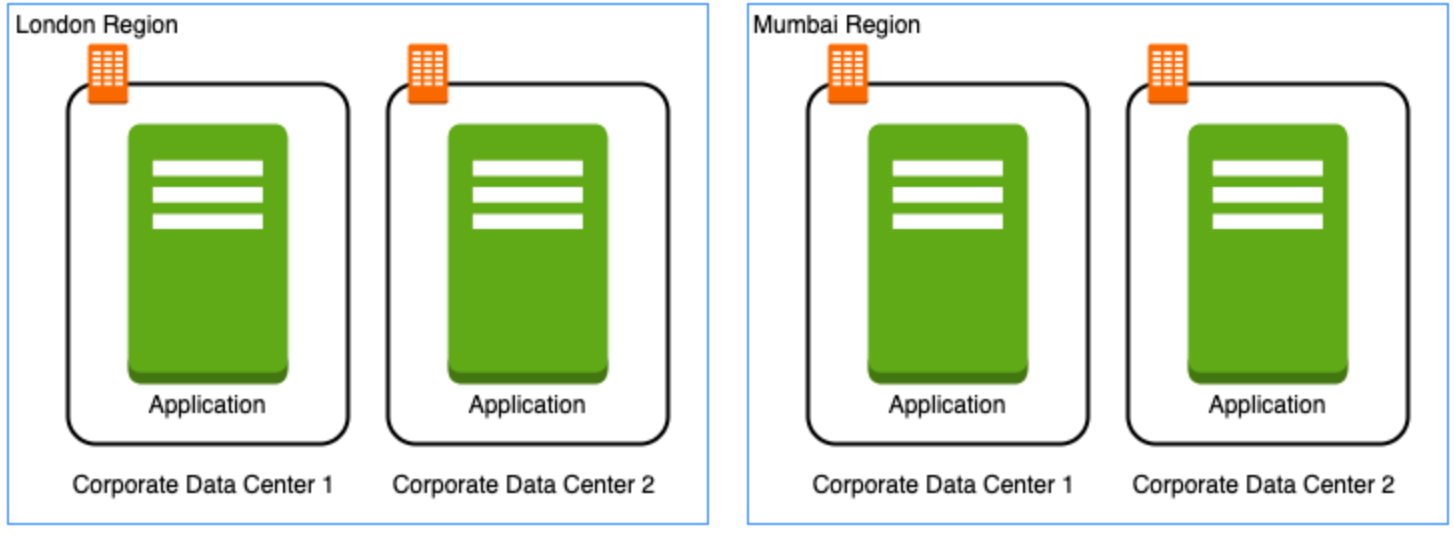
Multiple data centers

In28 Minutes



- 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

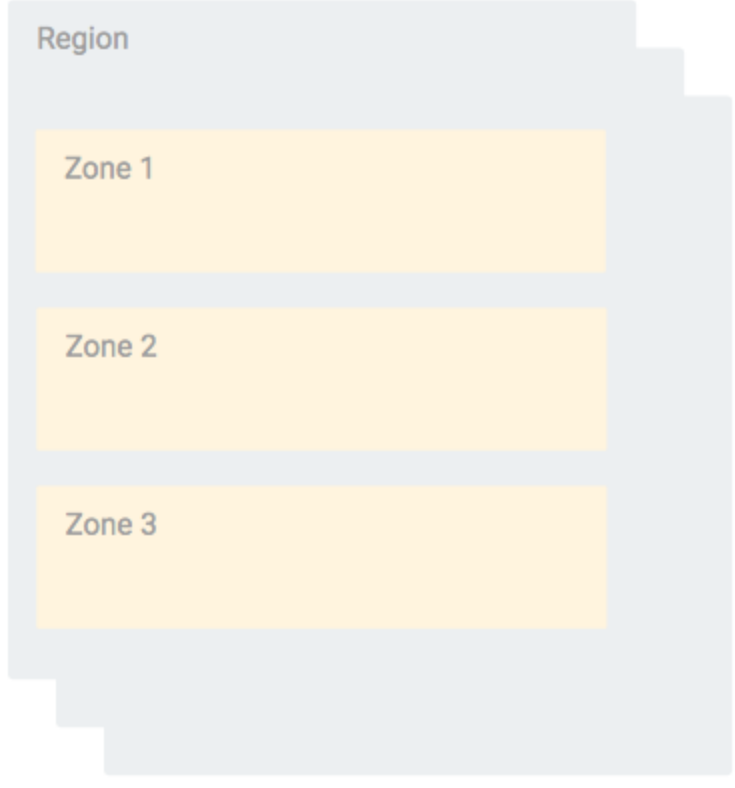
In28 Minutes

- 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

In28 Minutes

- 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

In28 Minutes

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
europe-north1	Hamina, Finland, Europe	3	europe-north1-a, europe-north1-b europe-north1-c
asia-south1	Mumbai, India APAC	3	asia-south1-a, asia-south1-b asia-south1-c

Compute

Google Compute Engine (GCE)

In 28 Minutes

- 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

In28 Minutes

- 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

In28 Minutes

- 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

In 28 Minutes

```
#! /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

In28 Minutes

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

In28 Minutes

- 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

In28 Minutes

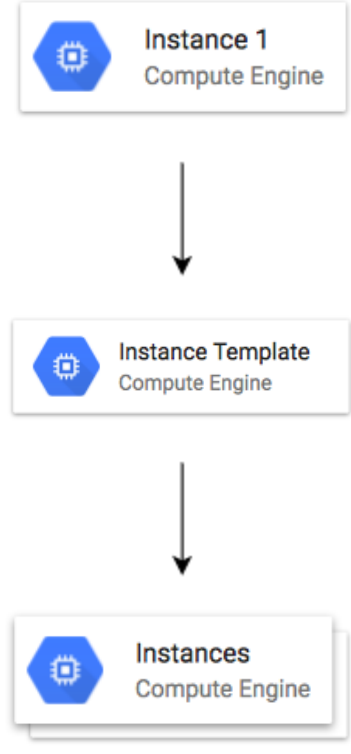
```
#!/bin/bash
apt update
apt -y install apache2
echo "Hello world from $(hostname) $(hostname -I)" > /var/www/html/index.html
```

- **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

In28 Minutes

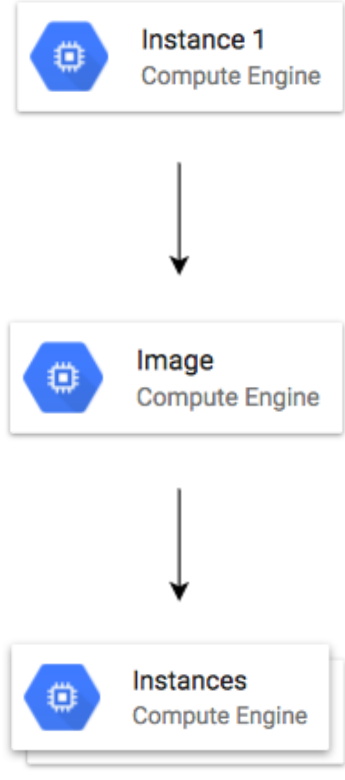
- Why do you need to specify all the VM instance details (Image, instance type etc) **every time** you launch an instance?
  - How about creating a **Instance template**?
  - Define **machine type, image, labels, startup script** and other properties
- Used to create **VM instances** and **managed instance 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

In28 Minutes

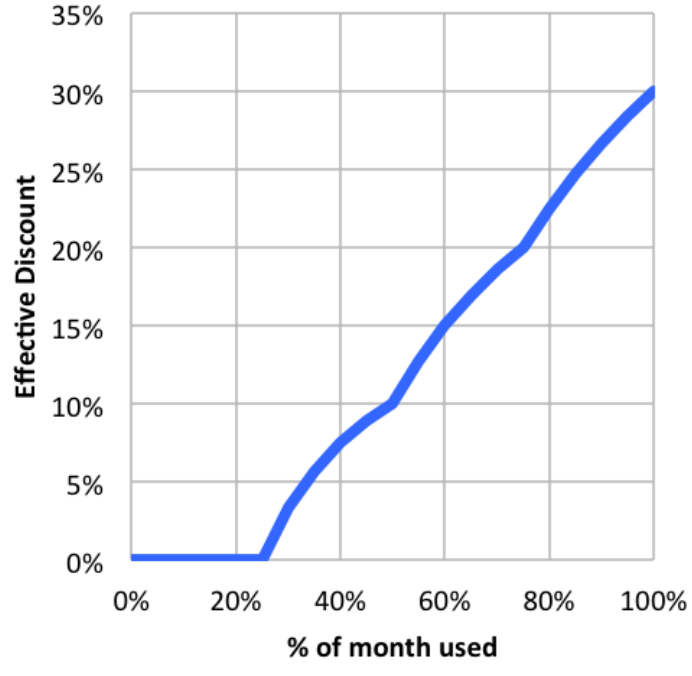
- Installing OS patches and software at launch of VM instances **increases boot up** time
- How about creating a custom image with OS patches and software **pre-installed**?
  - Can be created from an instance, a persistent disk, a snapshot, another image, or a file in Cloud Storage
  - Can be shared 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

In28 Minutes

- **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

In28 Minutes

- 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

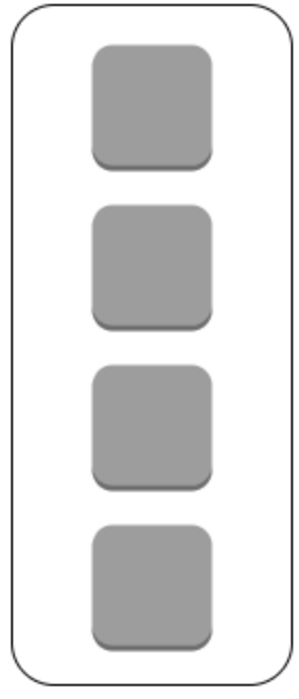
In28 Minutes

- **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

Compute Engine - Sole-tenant Nodes

In28 Minutes

- **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

In28 Minutes

- 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 / vCPU + $0.004446 / GB

Quick Review

In28 Minutes

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

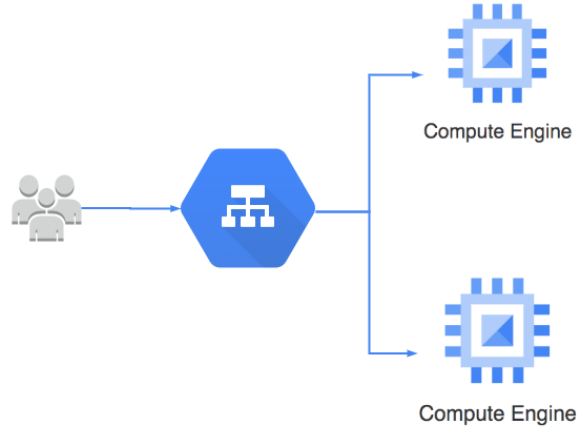
Quick Review

In28 Minutes

- **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

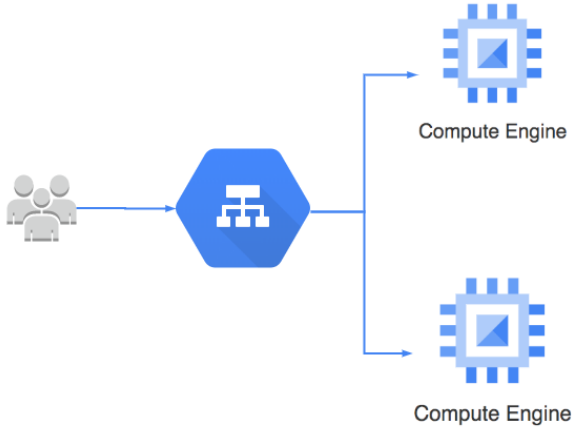
Instance Groups

- 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.



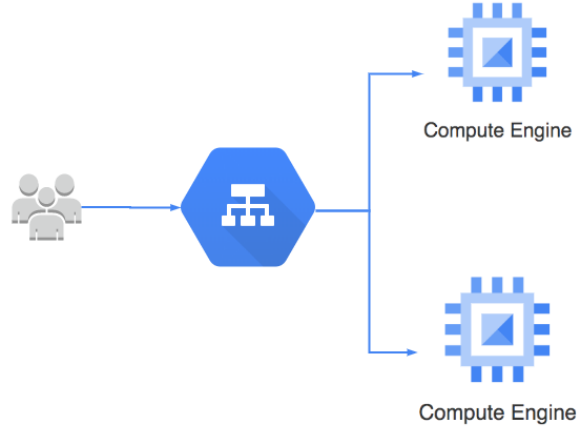
Creating Managed Instance Group (MIG)

In28 Minutes

- **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

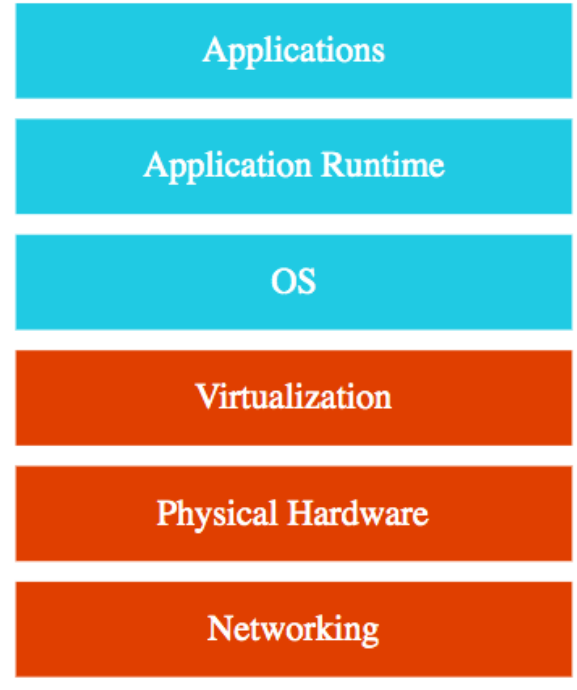
In28 Minutes

- 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)

In28 Minutes

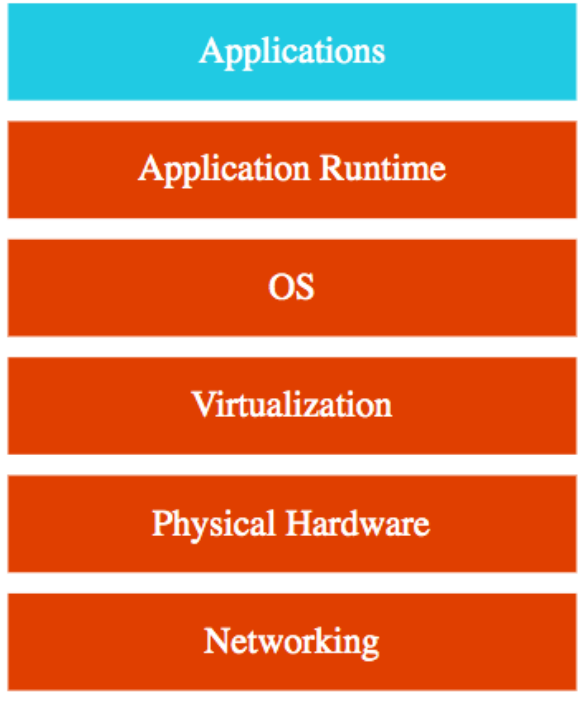
- Use **only infrastructure** from cloud provider
- **Example**: Using VM to deploy your applications or databases
- You are responsible for:
  - Application Code and Runtime
  - Configuring load balancing
  - Auto scaling
  - OS upgrades and patches
  - Availability
  - etc.. (and a lot of things!)



PAAS (Platform as a Service)

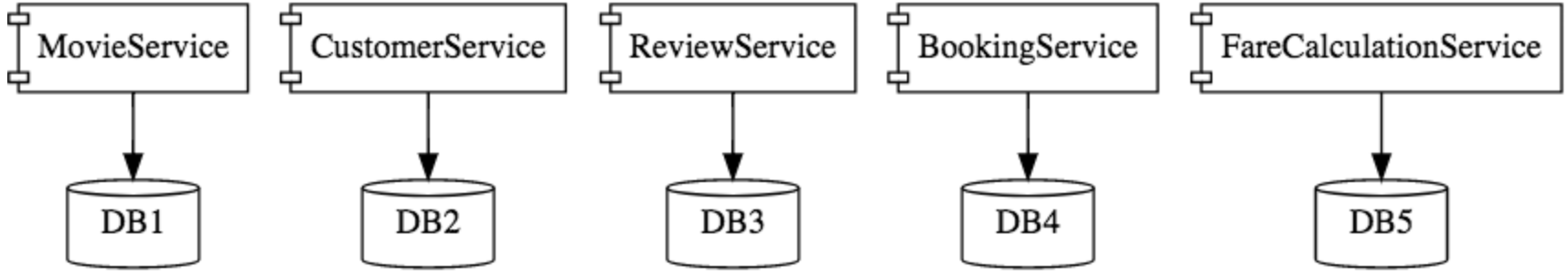
In28 Minutes

- 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

In28 Minutes

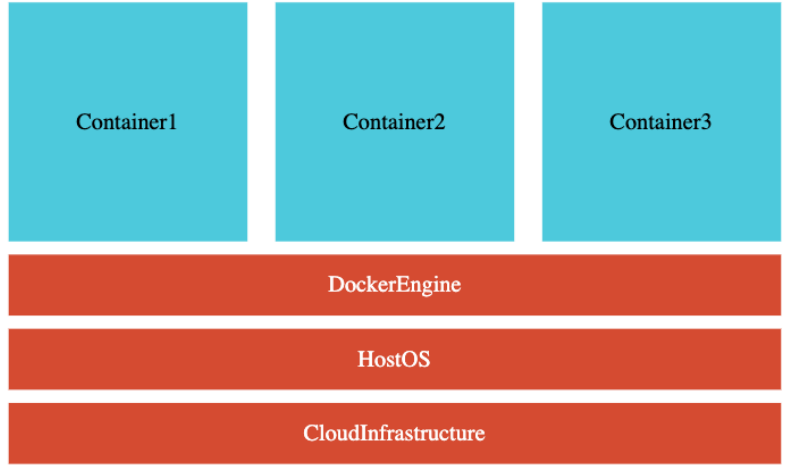


- Enterprises are heading towards microservices architectures
  - Build small focused microservices
  - **Flexibility to innovate** and build applications in different programming languages (Go, Java, Python, JavaScript, etc)
- BUT **deployments become complex**!
- How can we have **one way of deploying**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

Containers - Docker

In28 Minutes

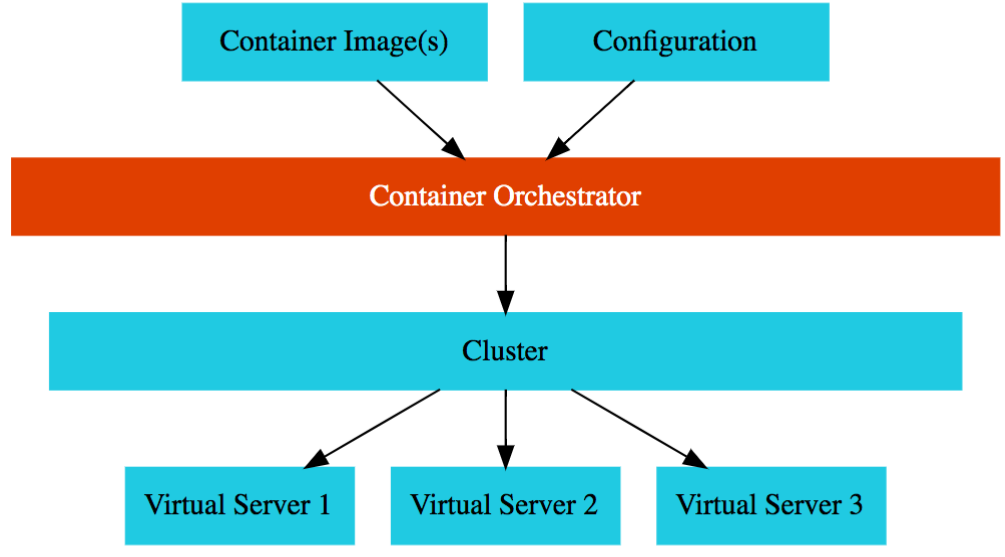
- Create **Docker images** for each microservice
- Docker image **has all needs of a microservice**:
  - Application Runtime (JDK or Python or NodeJS)
  - Application code and Dependencies
- Runs **the same way** on any infrastructure:
  - Your local machine
  - Corporate data center
  - Cloud
- Advantages
  - 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

In28 Minutes

- **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

In28 Minutes

- 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)

In28 Minutes

- **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)

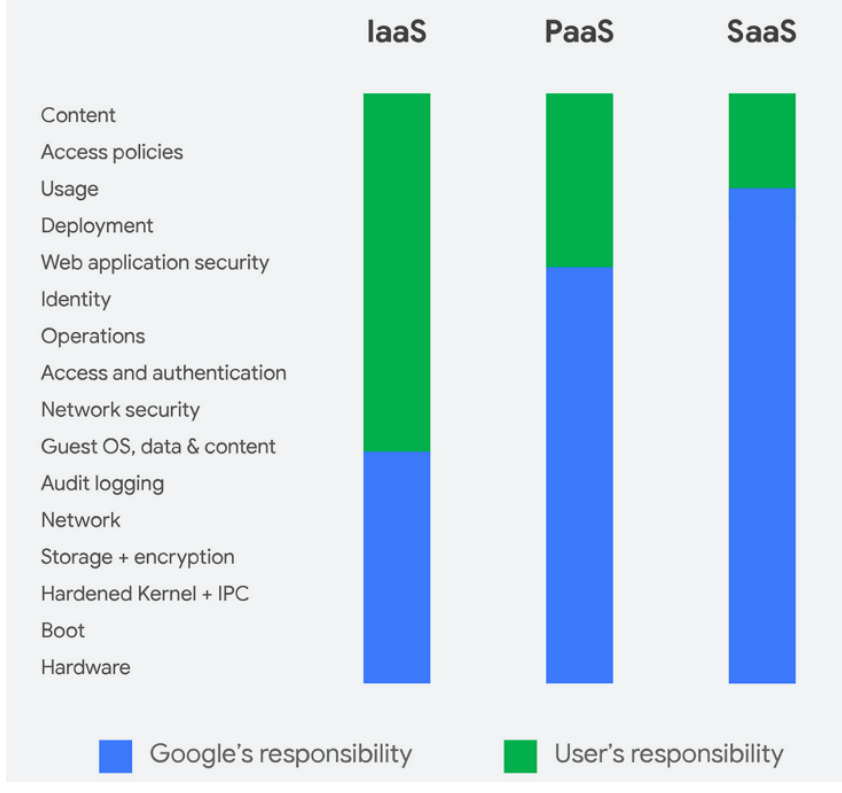
Google Cloud

- **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)

Shared Responsibility Model

In28 Minutes

- 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

In28 Minutes

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

Managed Compute Service in GCP

App Engine

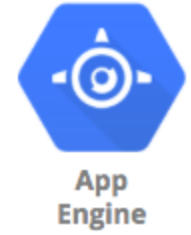
In28 Minutes

- **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

In28 Minutes

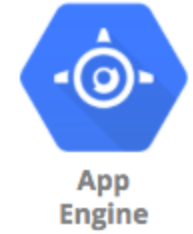
- **Compute Engine**
  - IAAS
  - MORE Flexibility
  - MORE Responsibility
    - Choosing Image
    - Installing Software
    - Choosing Hardware
    - Fine grained Access/Permissions (Certificates/Firewalls)
    - Availability etc
- **App Engine**
  - PaaS
  - Serverless
  - LESSER Responsibility
  - LOWER Flexibility



App Engine environments

In28 Minutes

- **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

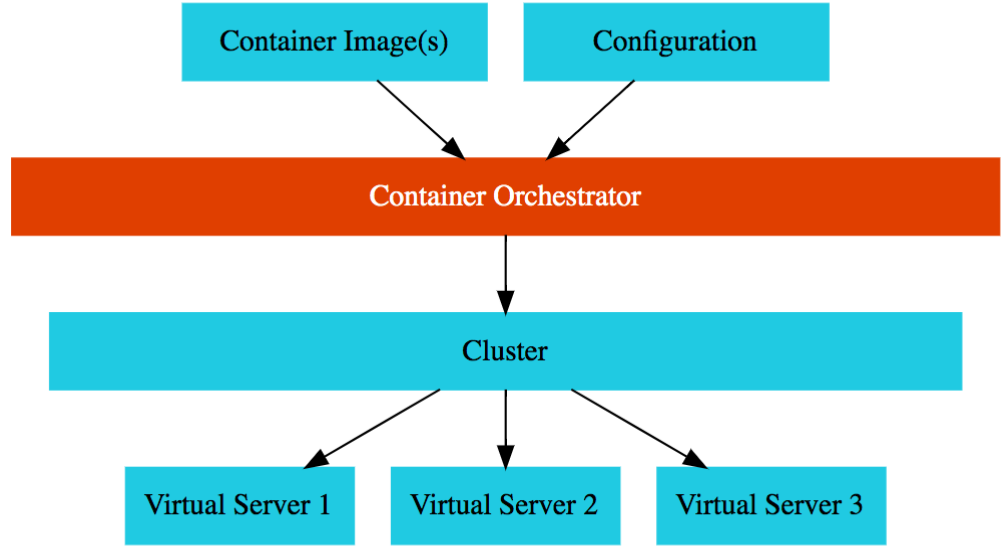
In28 Minutes

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

In28 Minutes

- 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)

In28 Minutes

- **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

In28 Minutes

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

- **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 - A Microservice Journey - Auto Scaling and ..

In28 Minutes

- **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***

Kubernetes Engine

Cloud Functions

In28 Minutes

- 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 9 minutes(540 seconds)
  - **Each execution runs in a separate instance**
    - No direct sharing between invocations

Cloud Run & Cloud Run for Anthos

In28 Minutes

- **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

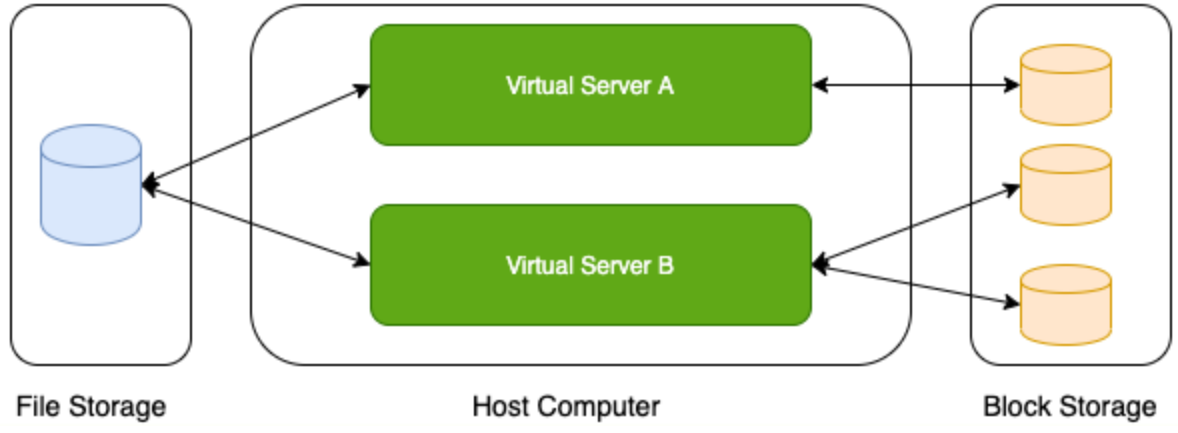
Scenarios - GCP Compute Services

In28 Minutes

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

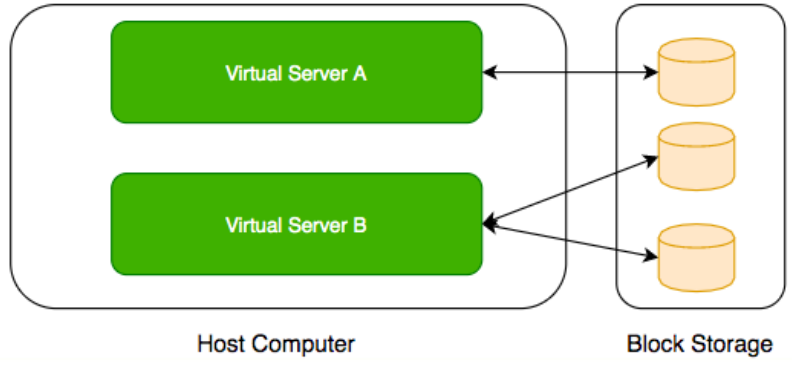
Storage Types - Block Storage and File Storage



- 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**

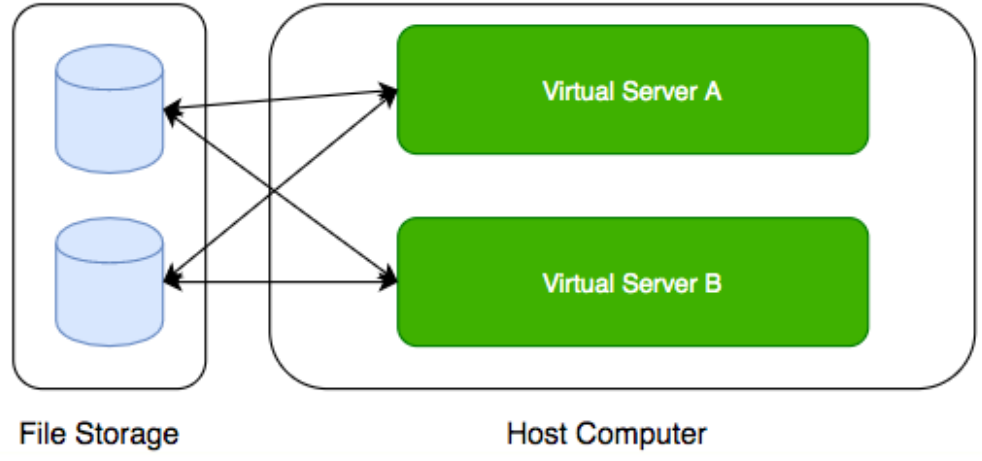
Block Storage

- 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

In28 Minutes

- **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

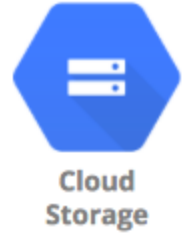
Cloud Storage

In28 Minutes

- **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

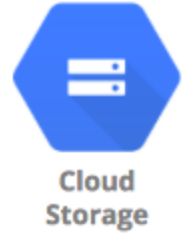
In28 Minutes



- 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

In28 Minutes



- **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)**

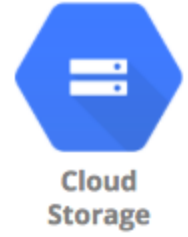
Cloud Storage - Storage Classes - Comparison

In28 Minutes

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

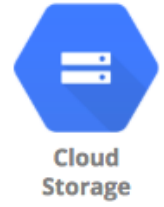
In28 Minutes



- 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

In28 Minutes



- 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

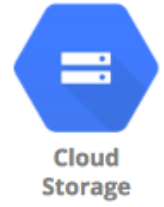
Object Lifecycle Management - Example Rule

In28 Minutes

```
{
    "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

In28 Minutes



- **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

In 28 Minutes

- **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

Database Fundamentals

Database Categories

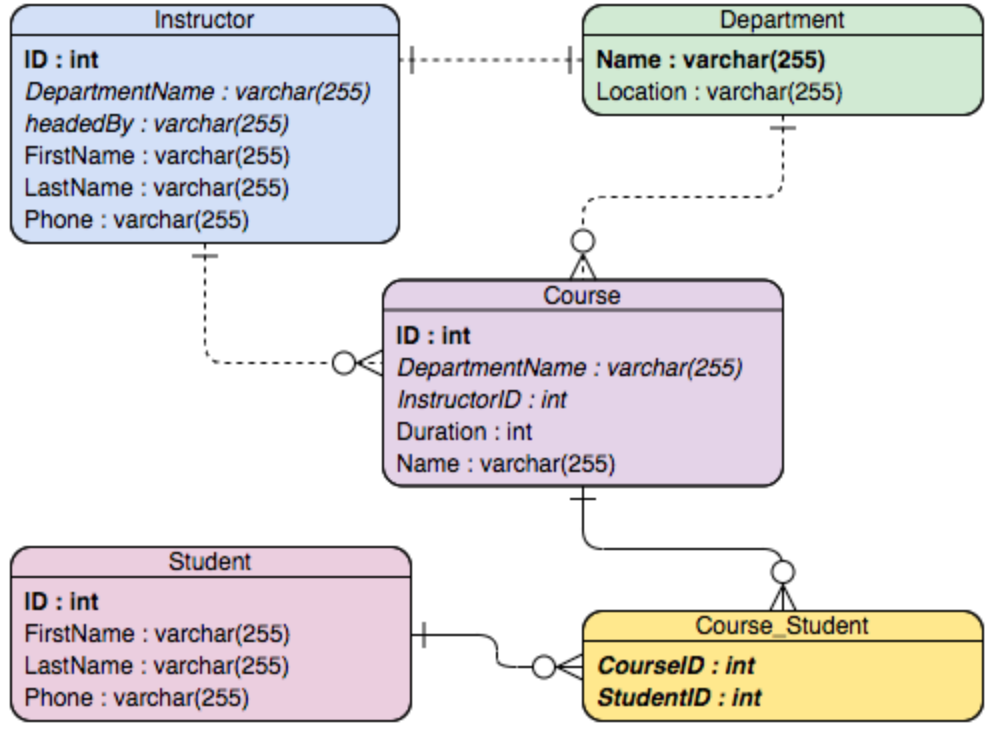
In28 Minutes

- 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...

Relational Databases

In28 Minutes

- 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)

- 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)

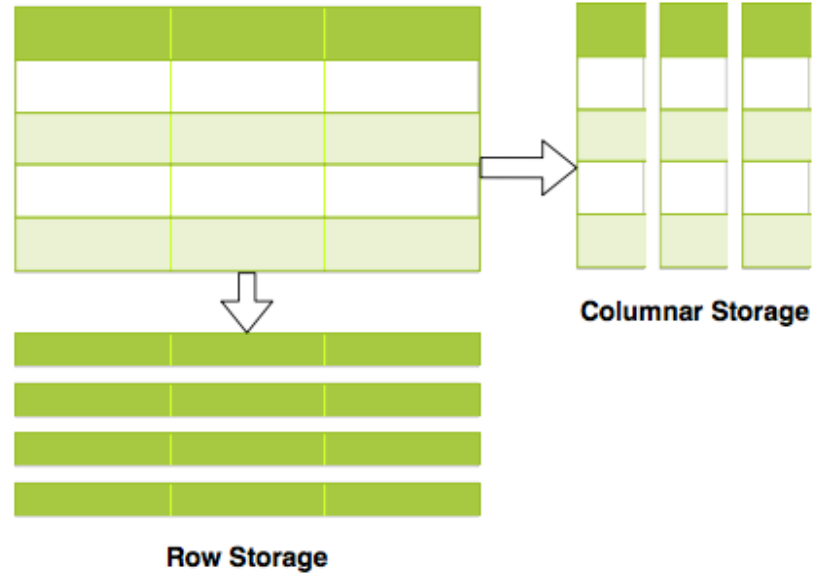
In28 Minutes

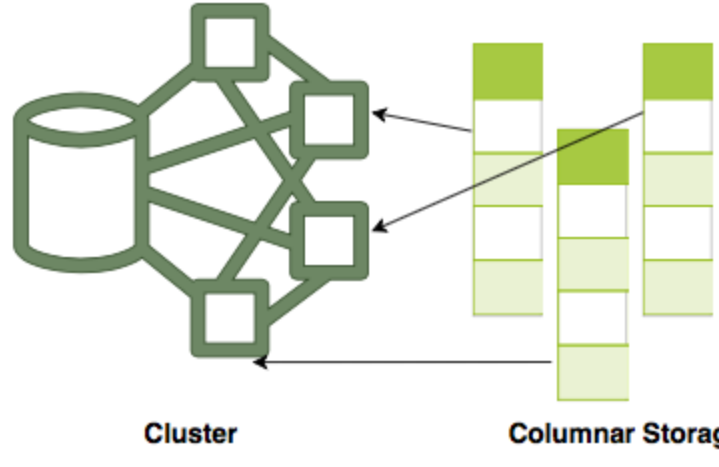
- Applications allowing users to **analyze petabytes of data**
  - **Examples** : Reporting applications, Data ware houses, Business intelligence applications, Analytics systems
  - **Sample application** : Decide insurance premiums analyzing data from last hundred years
  - Data is consolidated from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

Relational Databases - OLAP vs OLTP

In28 Minutes

- 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

In28 Minutes

- **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

In28 Minutes

- **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

In28 Minutes

- **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

In28 Minutes

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 databases/caches	Cloud Memorystore	Applications needing **microsecond** responses

Databases - Scenarios

In28 Minutes

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	CloudSpanner
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

Typical identity management in the cloud

In28 Minutes

- 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)

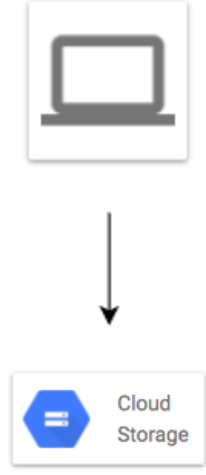
In28 Minutes

- **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

In28 Minutes

- 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

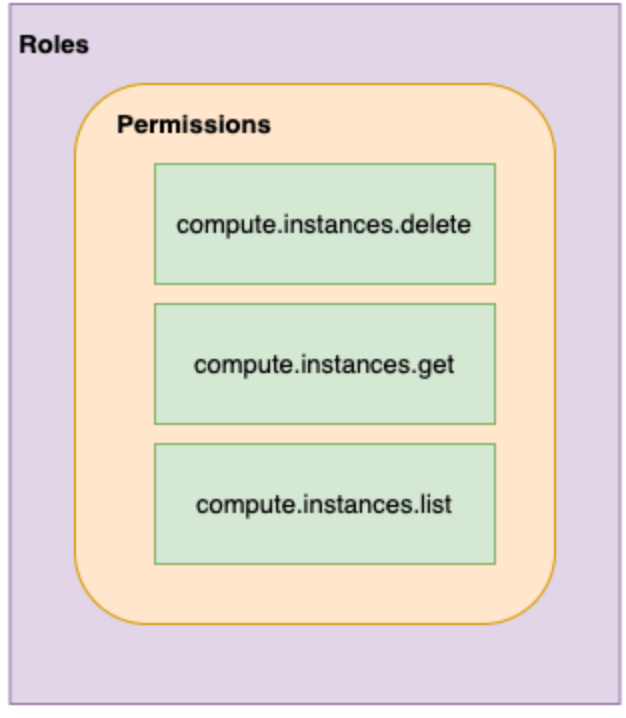


- 1: **Choose a Role** with right permissions (Ex: Storage Object Admin)
- 2: **Create Policy** binding member (your friend) with role (permissions)
- IAM in AWS is very different from GCP (Forget AWS IAM & Start FRESH!)
  - Example: Role in AWS is NOT the same as Role in GCP

IAM - Most Important Concepts - A Review

In28 Minutes

- **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

In28 Minutes

- 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

Cloud IAM

IAM policy - Example

In28 Minutes

```
{
  "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

In28 Minutes

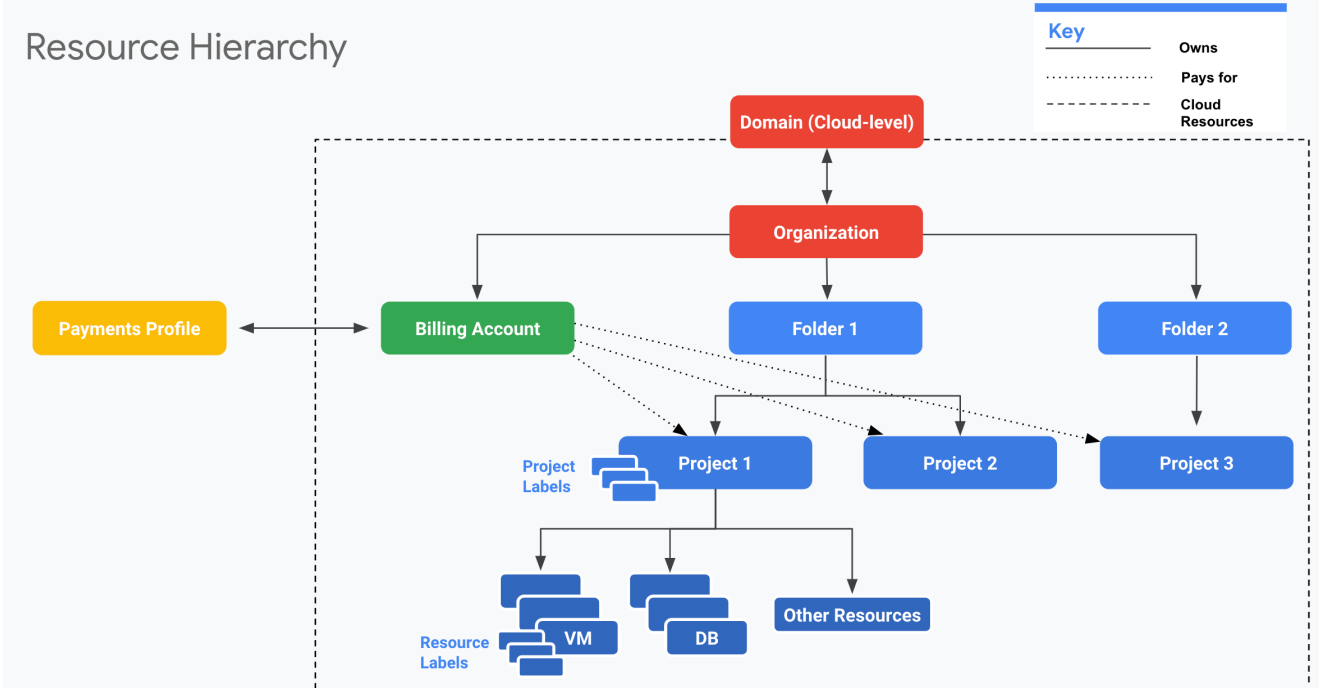
- Scenario: An Application on a VM needs access to cloud storage
  - You DONT want to use personal credentials to allow access
- (RECOMMENDED) Use **Service Accounts**
  - Identified by an email address (Ex: id-compute@developer.gserviceaccount.com)
  - Does NOT have password
    - Has a **private/public 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

Organizing GCP Resources

Resource Hierarchy in GCP

In28 Minutes

- **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

In28 Minutes

- 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

In28 Minutes

- **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

In28 Minutes

- 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

In28 Minutes

- **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 atleast 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

In28 Minutes

- 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

In28 Minutes

- **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

In28 Minutes

- 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

In28 Minutes

- 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 Gpbs 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

In28 Minutes

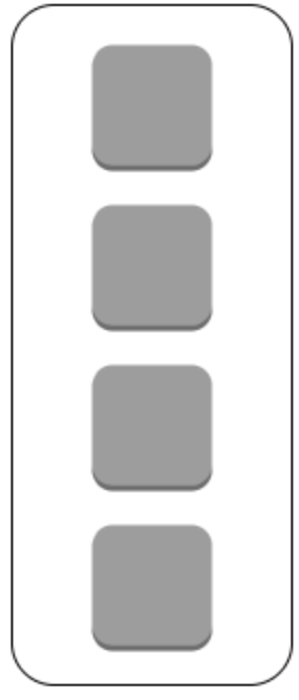
- Connect customer network to google network using network peering
  - Direct path from on-premises network to Google services
- **Not a GCP Service**
  - Lower level network connection outside of GCP
- NOT RECOMMENDED:
  - Use Cloud Interconnect and Cloud VPN

Other Virtual Machines Options in GCP

Going Bare Metal in the Cloud

In28 Minutes

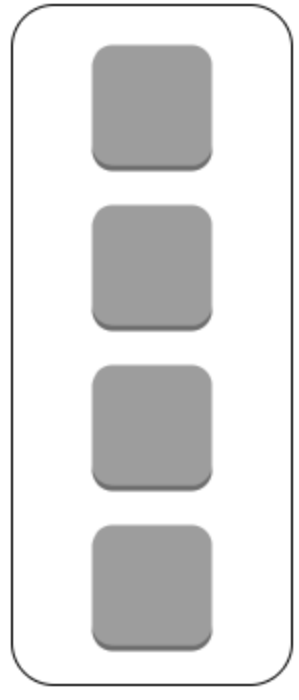
- **Google Compute Engine**: Create Virtual Machines (VMs)
  - Multiple VMs are created on a Single Host
    - **Virtualization**: One host computer supporting multiple guest VMs
    - **Hypervisor**: Software that creates and runs VMs
    - Hypervisor introduces "Hypervisor Tax"
      - A tax on performance - 5 – 10% overhead usually
- What if you need to run specialized workloads (SAP HANA, Oracle databases, ..) that need really high performance?
  - **Go Bare metal**: Hardware without any software pre-installed
    - Customizable and Dedicated Hardware
    - No Hypervisor (better performance than VMs)
    - Choose Your OS
    - Customer is responsible for licensing, installation, and maintenance of all software
    - **Usecases**:
      - Third-party virtualization software
      - Applications that require direct, low-level access to the server
        - Examples: SAP HANA, Oracle databases



Google Cloud VMware Engine

In28 Minutes

- Google Compute Engine provides virtualization
- **VMware provides virtualization solutions** as well
  - Very popular with Enterprises (virtual desktops (VDI) ..)
  - How do you shift you VMware workloads to cloud?
- **Google Cloud VMware Engine: Expand and Migrate (Lift and Shift)** VMware infrastructure to Google Cloud
  - Runs natively on Google Cloud bare metal infrastructure in a **dedicated, private cloud**
  - You **do NOT need to change** apps, tools, or processes
    - Continue using your existing VMware tools
  - Do NOT worry about Hardware and VMware licenses
  - Make use of the **high performance Google network**
  - Simplified connection to other Google Cloud services (Cloud SQL, BigQuery, ..)



Migrate for Compute Engine

In28 Minutes

- **Migrate for Compute Engine**: Migrate VMs and VM storage to GCE
  - From VMware, Microsoft Azure, Amazon EC2 ...
  - Migrate one or 1000s of VMs across data centres and clouds
- **Important Features**:
  - Test migration before making the move
    - Test-clone capability: Allows **non-disruptive testing of production workloads** in Google Cloud
      - Uses an isolated environment (ZERO impact on live prod apps)
  - Migrate VMs in batches (or waves) using VM groups
- **Phases**:
  - **1:** Planning migration waves
  - (OPTIONAL) **2:** Pre-migration testing using Test-clones
  - **3:** Cutover to cloud
    - Move VM to GCE
    - Migrate the application storage to Google Cloud
  - **4:** Detach (from Migrate for Compute Engine)

Migrate for Anthos and GKE

In28 Minutes

Kubernetes Engine

- How to modernize apps by moving from VMs to containers?
  - Use **Migrate for Anthos and GKE**
  - (Advantage of Containers) NO VM layers (like Guest OS)
  - (Advantage of Containers) More efficiently and cost effective
- **Source** GCE VM operating systems: Linux or Windows
- **Supported Destination** Platforms:
  - Google Kubernetes Engine (GKE)
  - Anthos
  - Anthos clusters on VMware
  - Anthos clusters on AWS
- **If you have VMs outside GCP**, this can be done using two steps:
  - **1:** Migrate VMs to GCE with Migrate for Compute Engine
  - **2:** Migrate GCE VMs to containers with Migrate for Anthos and GKE

DevOps

DevOps

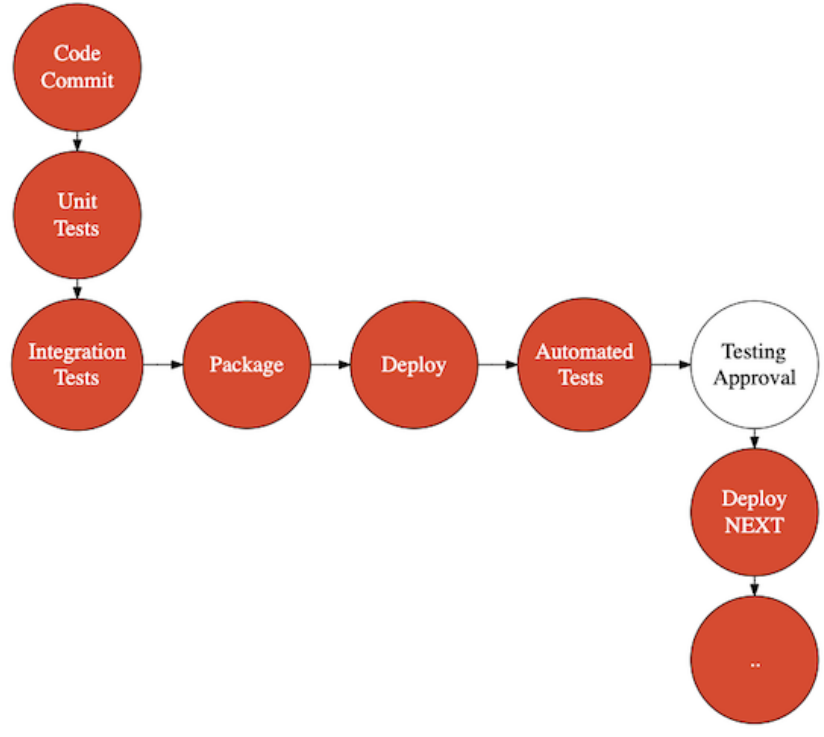
In28 Minutes

- 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

In28 Minutes

- **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

In28 Minutes

- **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

In28 Minutes

- **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

DevOps - Infrastructure as Code

In28 Minutes

- 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

In28 Minutes

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

Site Reliability Engineering (SRE)

In28 Minutes

- **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

In28 Minutes

- **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

In28 Minutes

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

In28 Minutes

Google Cloud

- **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

Site Reliability Engineering (SRE) - Best Practices - 3

In28 Minutes

Google Cloud

- **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

Decoupling Applications with Pub/Sub

Need for Asynchronous Communication

In 28 Minutes

- Why do we need asynchronous communication?

Synchronous Communication

In28 Minutes

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

Asynchronous Communication - Decoupled

In28 Minutes

- 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

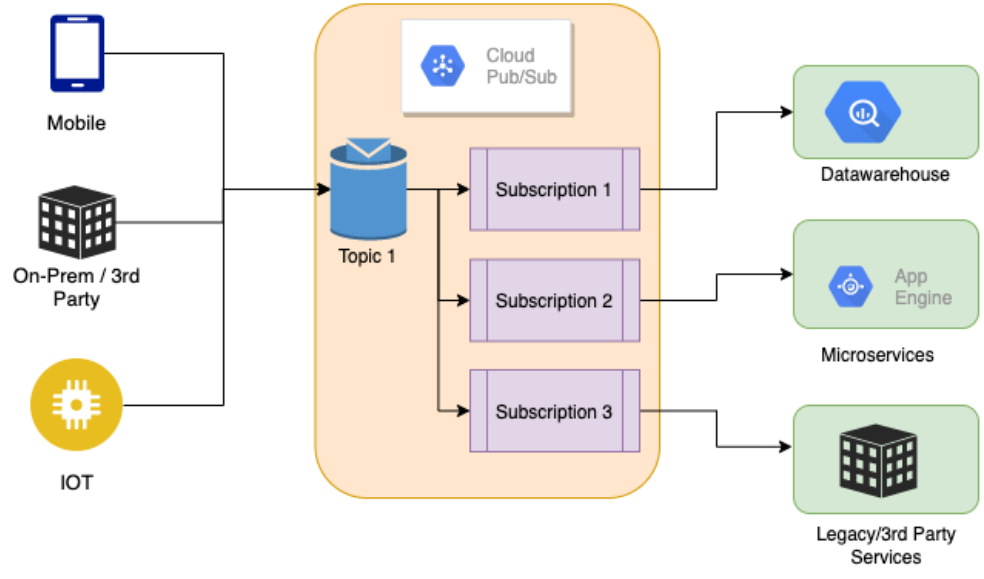
In28 Minutes

- 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?

In28 Minutes

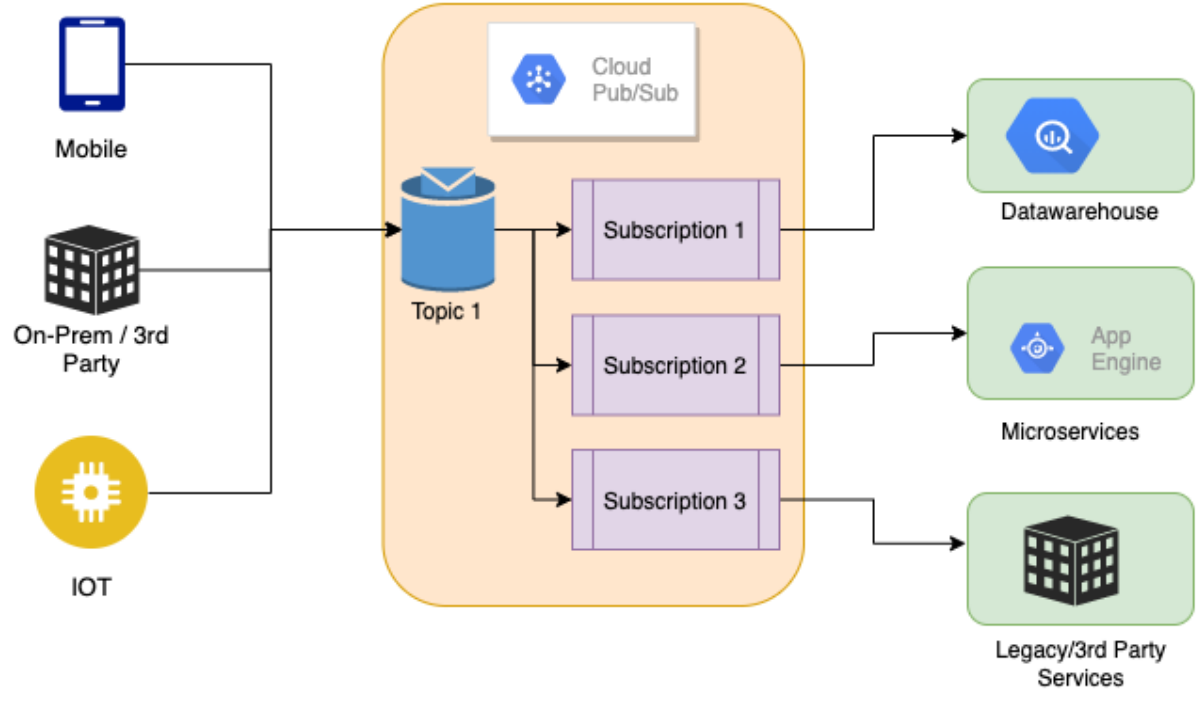
- **Publisher** - Sender of a message
  - Publishers send messages by making HTTPS requests to pubsub.googleapis.com
- **Subscriber** - Receiver of the message
  - **Pull** - Subscriber pulls messages when ready
    - Subscriber makes HTTPS requests to pubsub.googleapis.com
  - **Push** - Messages are 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

In28 Minutes

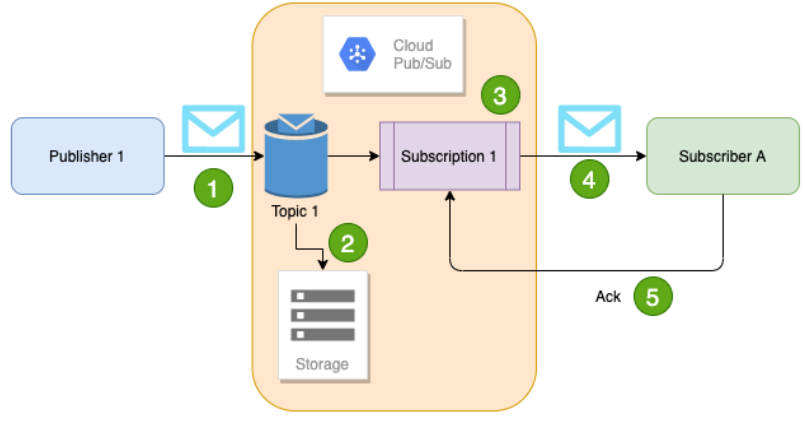
- Step 1 : Topic is created
- Step 2 : Subscription(s) are created
  - Subscribers register to the topic
  - Each Subscription represents discrete pull of messages from a topic:
    - Multiple clients pull same subscription => messages split between clients
    - Multiple clients create a subscription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



Pub/Sub - Sending and Receiving a Message

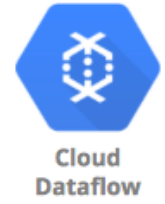
In28 Minutes

- 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

In28 Minutes



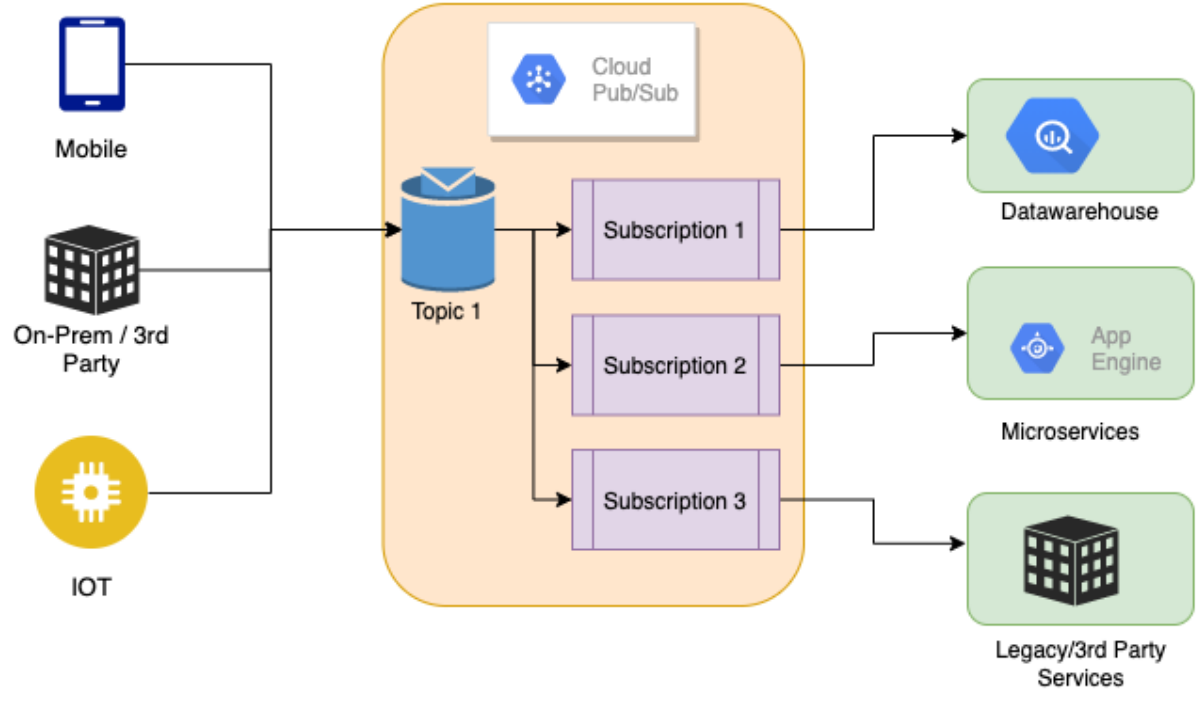
- **Cloud Dataflow** is a difficult service to describe:
  - Let's look at a **few example pipelines** you can build:
    - Pub/Sub > Dataflow > BigQuery (Streaming)
    - Pub/Sub > Dataflow > Cloud Storage (Streaming - files)
    - Cloud Storage > Dataflow > Bigtable/CloudSpanner/Datastore/BigQuery (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)

Architecture in Google Cloud: 10,000 Feet Overview

Architecture - Loose Coupling with Pub/Sub

In28 Minutes

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



Cloud Dataproc

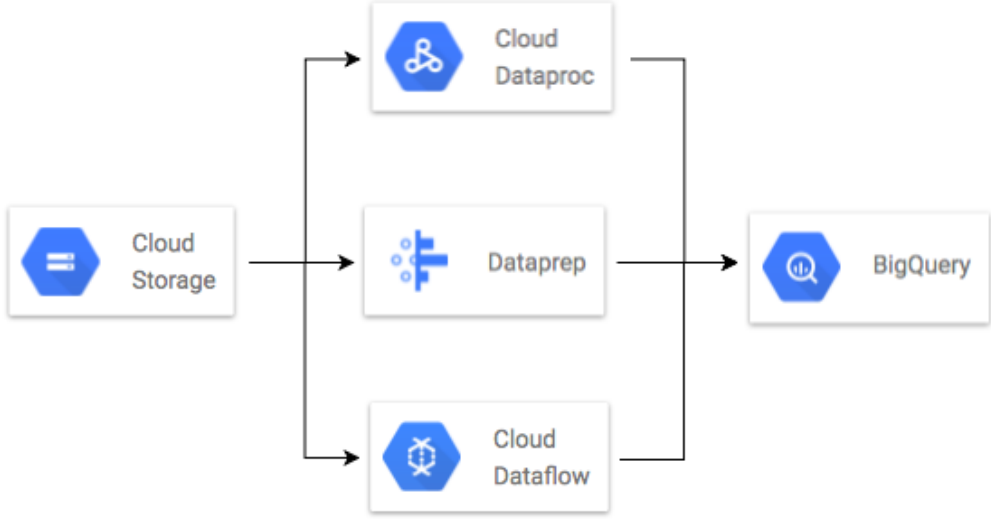
In28 Minutes

- 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

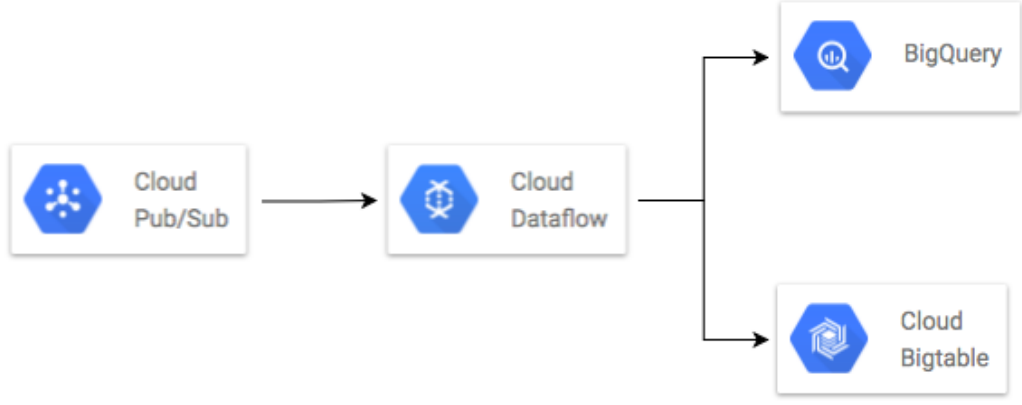
In28 Minutes

- 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



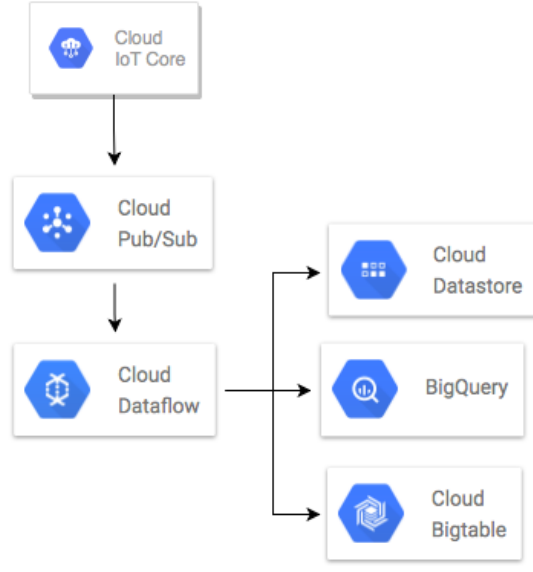
Architecture 2 - Streaming Data

- **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

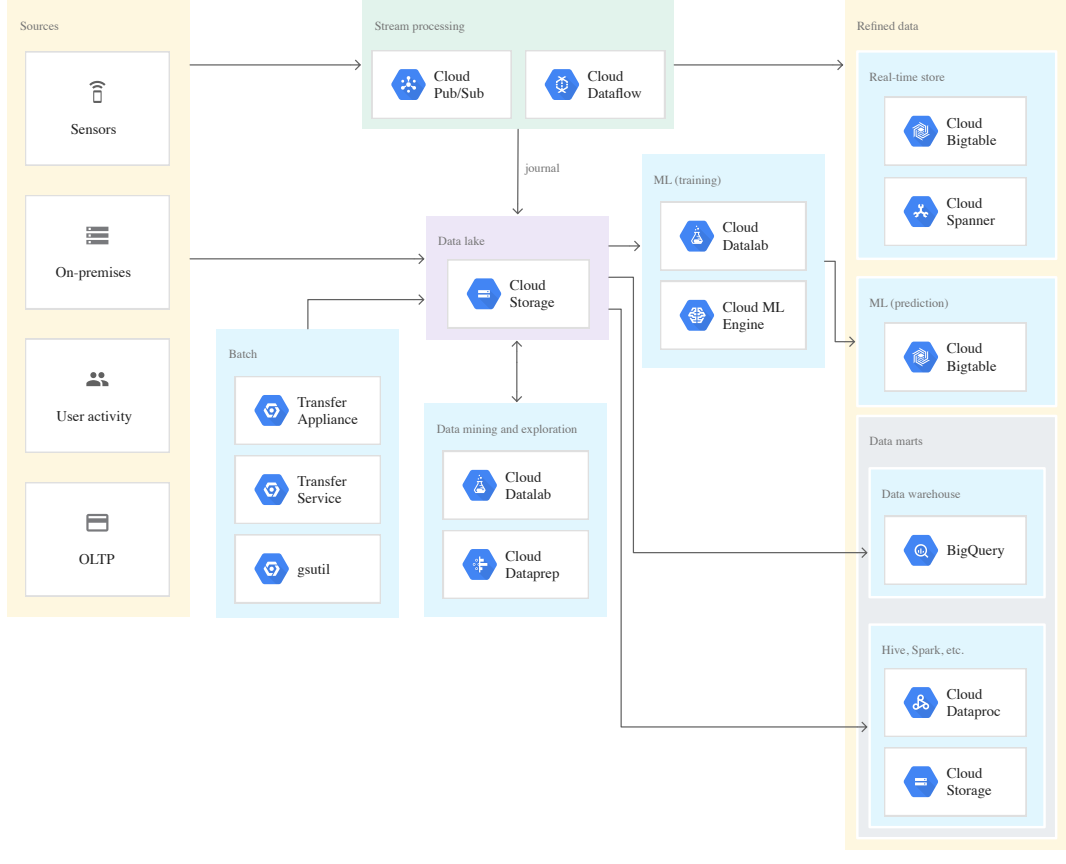
- **IoT Core**: Manage IoT (registration, authentication, and authorization) devices
  - Send/receive messages/real-time telemetry from/to IoT devices
- **Pub/Sub**: Durable message ingestion service (allows buffering)
- **Dataflow**: Processing data (ETL & more..)
  - Alternative: Use Cloud Functions 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

In28 Minutes

- 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

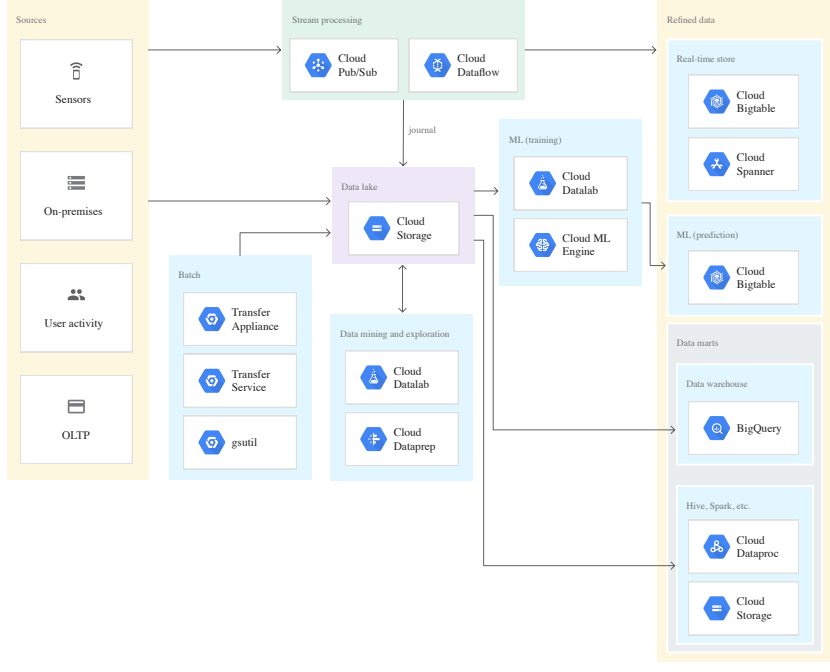


https://cloud.google.com/solutions/build-a-data-lake-on-gcp

GCP Data Lakes - Storage and Ingestion

In28 Minutes

- **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

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..

Exploring API management in Google Cloud

In28 Minutes

- **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

Machine Learning - 10,000 Feet Overview

In28 Minutes

- **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

In28 Minutes

Google Cloud

- 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

ML in Google Cloud - Custom Models

In28 Minutes

- **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

In28 Minutes

- 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 Scenarios

In28 Minutes

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

Firebase

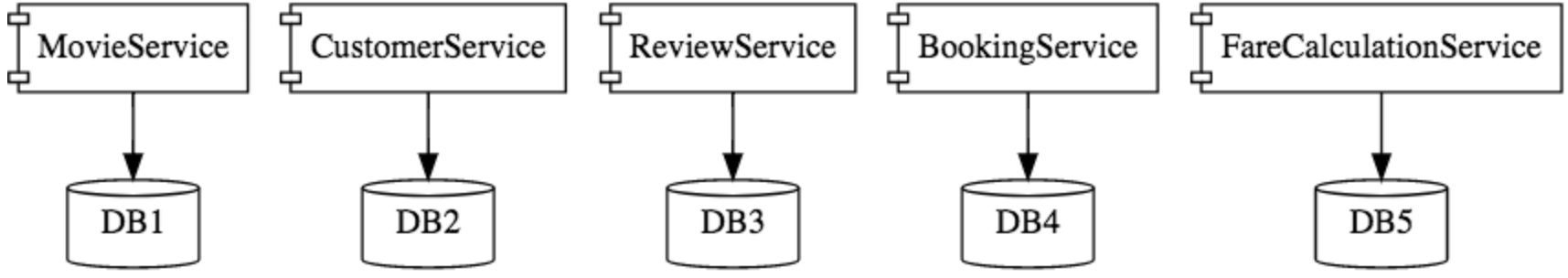
In28 Minutes

- Google's **mobile platform**: Build Apps for iOS, Android, the web, C++, and Unity
- **Features**:
  - Serverless **App Logic**: Cloud Functions
  - **Fully managed backend**: Firestore
    - Offline Support
  - Simplified Authentication and Sign-In flows
  - Simplified Monitoring and Analytics
  - Send and receive push messages: Cloud Messaging
  - Store user generated content: Cloud Storage
  - Simplifies Project Management: Integration with Slack, JIRA etc

Google Cloud

Exploring Container Registry and Artifact Registry

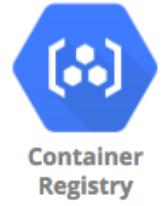
In28 Minutes



- 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

In28 Minutes



- 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

Monolith to Microservices: Application Modernization

In28 Minutes

- Prefer Managed Services
- Prefer Containers and Container Orchestration
- Prefer Stateless, Horizontally Scalable Microservices
- Use Automation (DevOps, SRE - CI/CD)
- Take a step by step approach:
  - Experiment and design proofs of concept
  - Replace application features with appropriate microservices in phases

Exploring Google Cloud Security Offerings

In28 Minutes

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

In 28 Minutes

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. Provides an intelligent risk dashboard and analytics system

Digital Transformation

What has changed in last decade or so?

In28 Minutes

- 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)

In28 Minutes

- **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

In 28 Minutes

- 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

In28 Minutes

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

In28 Minutes

- 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

In28 Minutes

- **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!

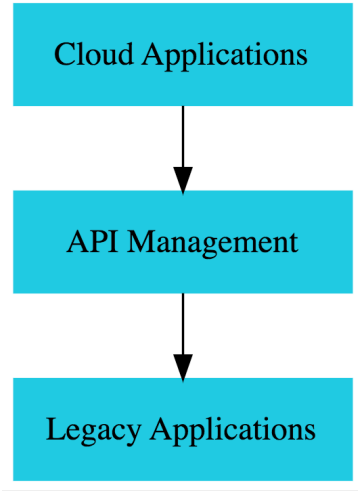
2: Application Modernization

- 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

3: Business Platform Modernization

In28 Minutes

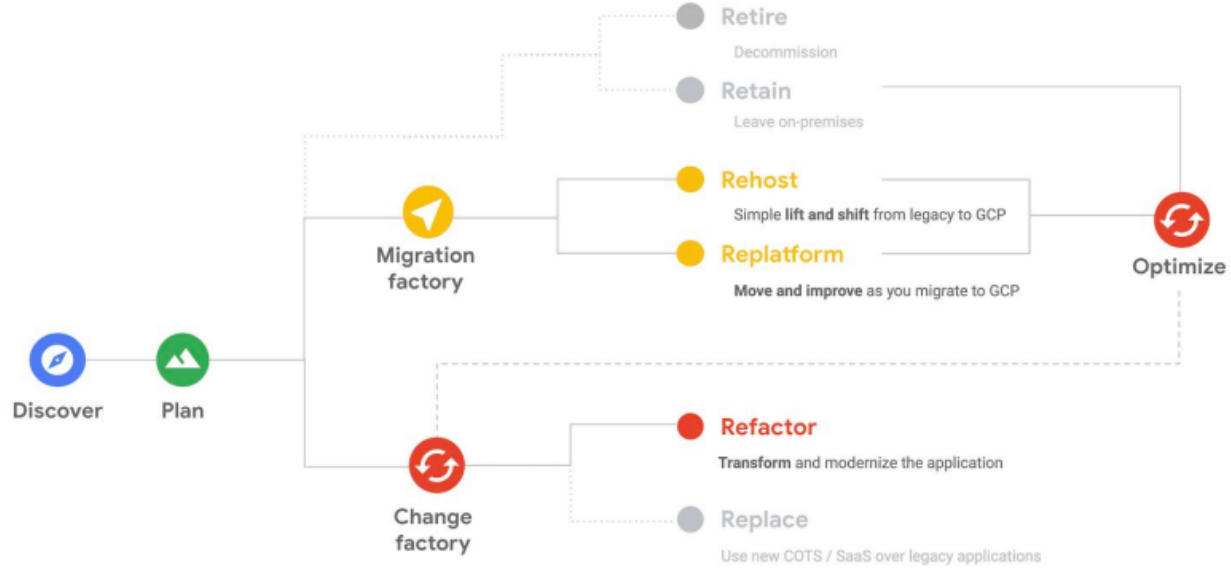
- What if you **DONT want to move** legacy system to Cloud?
- What if you want to enable external developers and partners to build apps for you?
- **Build APIs** around legacy code to simplify integration
- **Managed Services**: Apigee API Management, Cloud Endpoints
  - Design, Secure, Publish, Analyze, 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

In28 Minutes

- 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

In28 Minutes

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

Cost Management

Total Cost of Ownership (TCO)

In28 Minutes

- 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

In28 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

In28 Minutes

- **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?

In28 Minutes

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

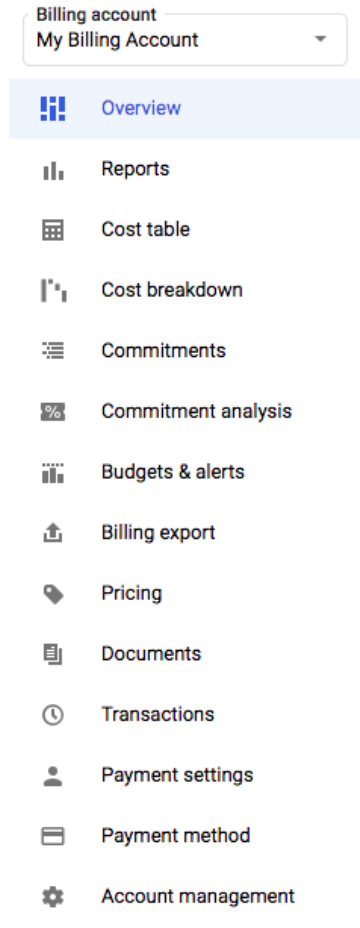
In28 Minutes

- **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

In28 Minutes

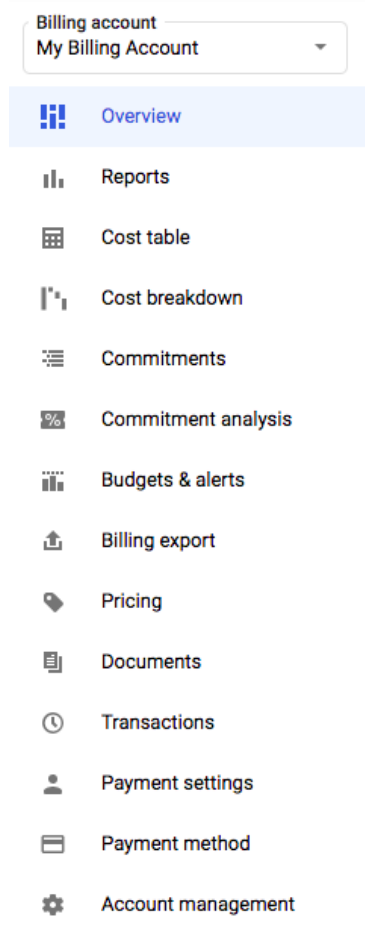
- **Cost Management**: Tools for monitoring, controlling, and optimizing your costs
  - **Cost Billing Reports**: 10,000 feet overview of usage costs
    - Analyze Trends by Project, Service, Location, Labels etc..
  - **Cost Table report**: Detailed view
    - Dynamically filter, sort and group various 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



Managing Costs - Best Practices

In28 Minutes

- **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



Quick Review

Basic Compute Services - Google Cloud

In28 Minutes

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

In28 Minutes

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

In28 Minutes

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

In28 Minutes

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

In28 Minutes

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

In28 Minutes

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

In28 Minutes

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

In28 Minutes

- **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!**

You are all set!

Let's clap for you!

In28 Minutes

- 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!

In28 Minutes

- 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?

In28 Minutes

- Go Deeper into AWS!
  - Three things I would recommend
    - Serverless (Lambda, API Gateway DynamoDB)
    - Elastic Beanstalk
    - ECS
- Learn other Cloud Platforms:
  - Gartner predicts a multi cloud world soon
  - Get certified on AWS, Azure and Google Cloud
- Learn DevOps (Containers and Container Orchestration)
- Learn Full Stack Development