In28 Minutes

Google Cloud Platform for AWS Experts

Getting Started

In28 Minutes

- GCP and AWS have *200+ services* each.
- You can learn GCP step by step as you would have learnt AWS
- OR you can take a shortcut
  - Learn GCP by comparing with AWS (and its services)
- **Our Goal** : Give you a quick start for GCP!

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

Setting up GCP Account

In28 Minutes

- Create GCP Account

10,000 Feet - GCP vs AWS - Compute and Networking

In 28 Minutes

Question	AWS	GCP
How do you create Virtual Machines?	Amazon EC2	Compute Engine
How do you attach permanent storage (block storage) with Virtual Machines?	Amazon EBS	Persistent Disk
How do distribute load among VMs?	Elastic Load Balancer	Cloud Load Balancing
How do you simplify setting up web applications?	AWS Elastic Beanstalk	App Engine
How do you orchestrate containers?	Amazon EKS, Amazon ECS	Google Kubernetes Engine (GKE)
How do you build serverless applications?	AWS Lambda	Cloud Functions
How do you build private networks?	Amazon VPC	Cloud VPC
How do you connect on premise with cloud?	AWS VPN(shared), AWS Direct Connect (dedicated)	Cloud VPN(shared), Cloud Interconnect (dedicated)

10,000 Feet - GCP vs AWS - Storage

In28 Minutes

Question	AWS	GCP
What is the Object storage solution?	Amazon S3	Cloud Storage
What is the Block storage solution?	Amazon EBS	Persistent Disk
What is the File storage solution?	Amazon EFS	Filestore

10,000 Feet - GCP vs AWS - Database

In28 Minutes

Question	AWS	GCP
How do you create relational OLTP databases?	Amazon RDS (Amazon Aurora)	Cloud SQL, Cloud Spanner
What is the relational data warehouse solution?	Amazon Redshift	BigQuery
What are the NoSQL database options?	Amazon DynamoDB, Amazon DocumentDB	Datastore/Firestore, Cloud Bigtable
How do you cache data from a database?	Amazon ElastiCache	Memorystore

10,000 Feet - GCP vs AWS - Others

In28 Minutes

Question	AWS	GCP
What are the messaging services?	Amazon SNS, Amazon SQS	Cloud Pub/Sub
How do you manage authentication and authorization to Cloud?	Amazon IAM	Cloud IAM
How do you manage keys used for encrypting data?	AWS KMS	Cloud KMS
How do you automate deployment?	AWS CloudFormation	Cloud Deployment Manager
How do you monitor metrics around your applications	Amazon CloudWatch	Cloud Monitoring
How do you manage application and service logs?	Amazon CloudWatch Logs	Cloud Logging
How do you trace requests across applications and services?	AWS X-Ray	Cloud Trace

10,000 Feet - GCP vs AWS - Remember

In28 Minutes

- Question (or Context or Problem)
  - AWS Solution
  - GCP Solution
- You will NOT be able to remember all the services on Day1
- BUT make an effort!
- (Recommended) Revise previous slides often. Do not hesitate to replay videos!
- (Recommended) Have Fun!

AWS - Regions and Availability Zones

In28 Minutes

Region Code	Region	Availability Zones	Availability Zones List
us-east-1	US East (N. Virginia)	6	us-east-1a us-east-1b us-east-1c us-east-1d us-east-1e us-east-1f
eu-west-2	Europe (London)	3	eu-west-2a eu-west-2b eu-west-2c

- AWS provides **20+ regions** around the world
  - **Advantages** - High Availability, Low Latency and Adhere to government **regulations**
  - Choose region(s) based on - Users Location, Data Location, Regulatory and compliance needs
- Each AWS Region has at least two Availability Zones
  - **Isolated locations** in a Region
  - **Increase availability** of applications in the same region

GCP - Regions and Zones

In28 Minutes

Region Code	Region	Zones	Zones List
us-west1	The Dalles, Oregon, North America	3	us-west1-a us-west1-b us-west1-c
asia-south1	Mumbai, India APAC	3	asia-south1-a, asia-south1-b asia-south1-c

- Regions in AWS = Regions in GCP
- Availability Zones in AWS = Zones in GCP
  - Three or more Zones in a Region (GCP)
- Additional Geographical Groups in GCP:
  - **Multi-region Locations**: Two or more regions. Ex: United States, Europe, and Asia.
  - **Dual Regions**: Specific pair of regions. Ex: nam4 (Iowa and South Carolina), eur4 (Netherlands and Finland), asia1 (Tokyo and Osaka).

Compute

Google Compute Engine (GCE)

In28 Minutes

- In AWS, we use EC2 service to provision Virtual Instances
- In GCP, the corresponding service is GCE or Google Compute Engine
  - **Google Compute Engine (GCE)** - Provision & Manage Virtual Instances
  - **Virtual Machines** - Virtual Instances in GCP

Compute Engine

10,000 Feet - Virtual Machines in GCP and AWS

In28 Minutes

Feature	AWS	GCP
Create Virtual Machines	Amazon EC2	Google Compute Engine (GCE)
Choose Operating System and Software	AMI (Amazon Machine Image)	Image
Choose the right family of hardware (Generic or high memory or high compute)	Instance Family	Machine Family
Choose the right quantity of hardware (2 vCPUs, 4GB of memory)	Instance Type	Machine Type
Restrict inbound and outbound traffic	Security Groups	Firewall Rules
Attach Permanent Hard Disks (Block Storage)	Amazon EBS	Persistent Disks

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

Feature	AWS	GCP
Permanent Internal IP Address that does not change during the lifetime of an instance	Private IP Address	Internal IP Address
Ephemeral External IP Address that changes when an instance is stopped	Public IP Address	External or Ephemeral IP Address
Permanent External IP Address that can be attached to a VM	Elastic IP Address	Static IP Address

Managing Virtual Machines

In28 Minutes

Feature	AWS	GCP
Templates to simplify creation of Virtual Machines	Launch Templates/ Configuration	Instance templates
Simplify creation of multiple Virtual Machines	Auto Scaling Group	Instance Groups
Physical hosts dedicated to one customer	EC2 Dedicated Hosts	Sole-tenant nodes
Simplify management (software, OS patches etc) of 1000's of Virtual Machines	Systems Manager	VM Manager

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**

Compute Engine

Bootstrapping with Startup script

In28 Minutes

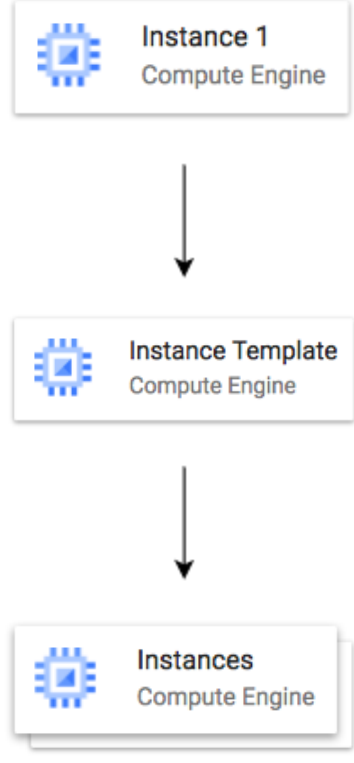
```
#!/bin/bash
apt update
apt -y install apache2
echo "Hello world from $(hostname) $(hostname -I)" > /var/www/html/index.html
```

- **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

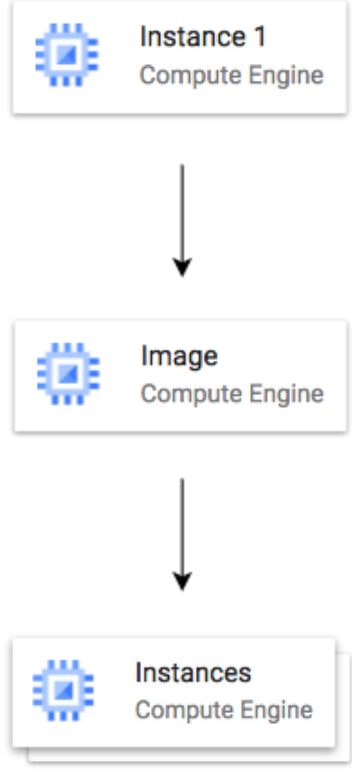
- Why do you need to specify all the VM instance details (Image, instance type etc) **every time** you launch an instance?
  - How about creating a **Instance template**?
  - Define **machine type, image, labels, startup script** and other properties
- Used to create **VM instances** and **managed instance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



Manage Costs for Virtual Machines

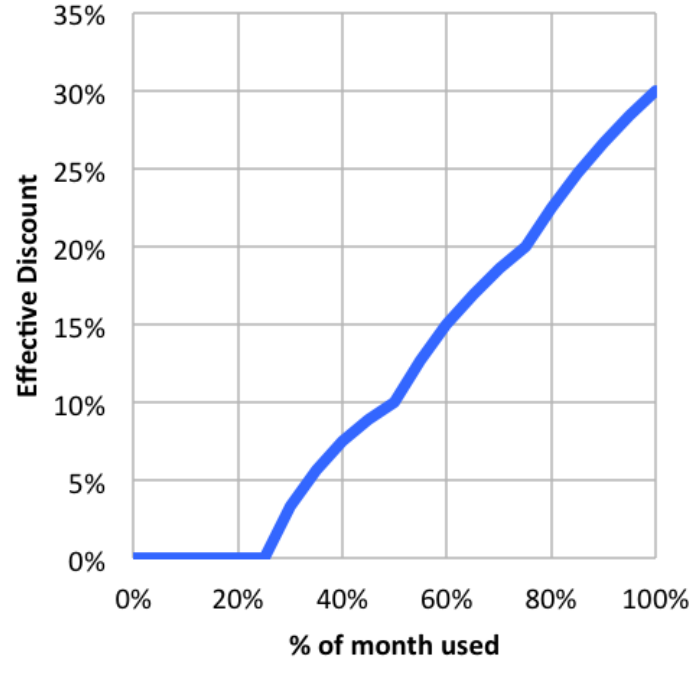
In28 Minutes

Feature	Amazon EC2	GCE
Billing	Billed by second	Billed by second (after 1st minute)
Create cheaper, temporary instances for non critical workloads	Spot instances	Preemptible VMs (Fixed pricing, Max 24 hrs)
Reserve compute instances ahead of time	Reserved instances	Committed use discounts
Get discounts for using resources for long periods of time	None	Sustained use discounts
Budget Management	Budget alerts	Budget alerts

GCE - Sustained use discounts

In28 Minutes

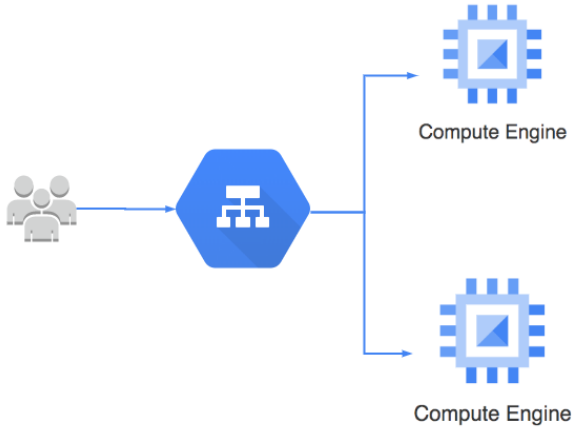
- **Automatic discounts** for running VM instances for significant portion of the billing month
  - Example: If you use N1, N2 machine types for more than 25% of a month, you get a 20% to 50% discount on every incremental minute.
  - Discount increases with usage (graph)
  - No action required on your 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*

Instance Groups

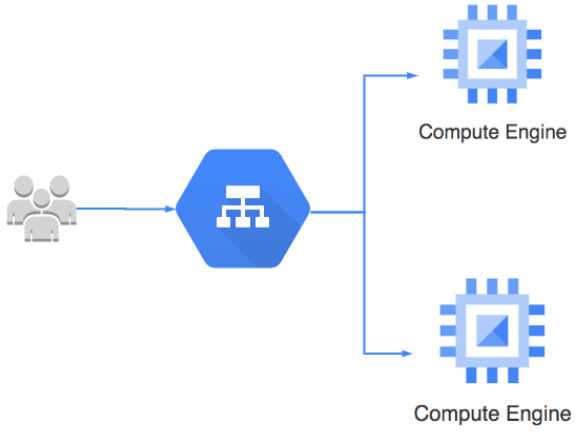
- How do you create a group of VM instances?
  - **Instance Group** - Group of VM instances managed as a single entity
    - **Manage group** of similar VMs having similar lifecycle as **ONE UNIT**
- **Two Types of Instance Groups:**
  - **Managed** : Identical VMs created using a template:
    - Features: 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)

In28 Minutes

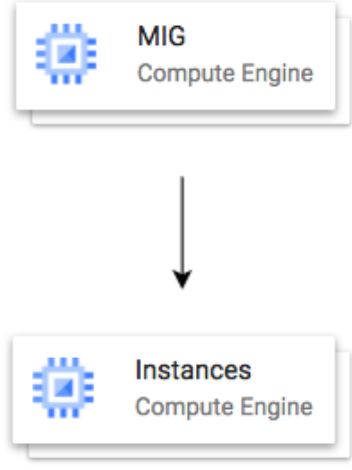
- **Managed Instance Group** - Identical VMs created using an **instance template**
- Important Features:
  - **Maintain** certain number of instances
    - If an instance crashes, MIG launches another instance
  - **Detect application failures** using health checks (Self Healing)
  - Increase and decrease 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

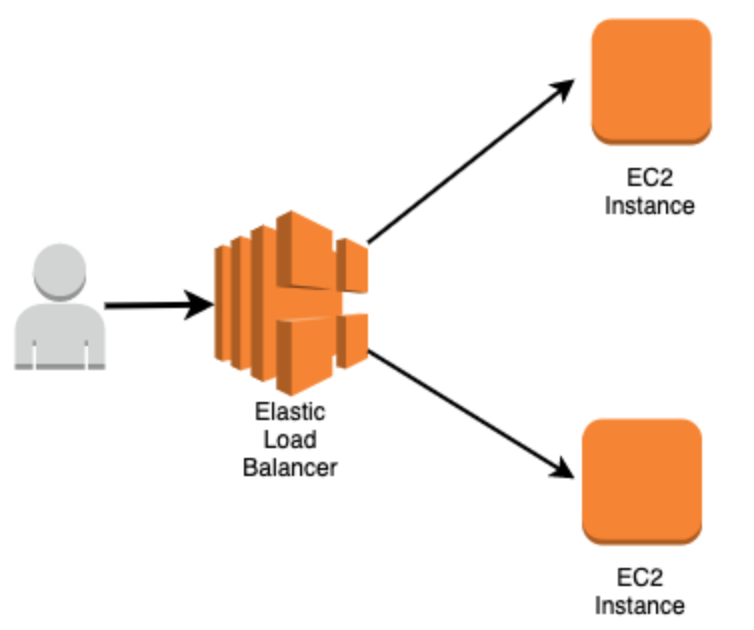
- **Instance template** is mandatory
- Configure **auto-scaling** to automatically adjust number of instances based on load:
  - **Minimum** number of instances
  - **Maximum** number of instances
  - **Autoscaling metrics**: CPU Utilization target or Load Balancer Utilization target or Any other metric 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**



AWS - Elastic Load Balancer

In28 Minutes

- Distribute traffic across EC2 instances in one or more AZs in a single region
- **Managed service**:
  - AWS ensures that it is highly available
  - Auto scales to handle huge loads
  - Load Balancers can be **public or private**
- Types:
  - **Classic** Load Balancer (OLD - Layer 4 and Layer 7)
  - **Application** Load Balancer (Layer 7 HTTP/HTTPS)
  - **Network** Load Balancer (High Performance - Layer 4 TCP/TLS and UDP)



GCP - Cloud Load Balancing

In28 Minutes

- Distribute traffic across VM instances in one or more regions
- Single anycast IP
- **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

Compute

In28 Minutes

Category	AWS	GCP
IAAS	Amazon EC2	Google Compute Engine
PAAS	AWS Elastic Beanstalk	App Engine
CAAS - Kubernetes	Amazon EKS	Google Kubernetes Engine
CAAS - Custom	Amazon ECS	
CAAS - Serverless	AWS Fargate	Cloud Run
FAAS - Serverless	AWS Lambda	Cloud Functions

Compute Engine

Cloud Functions

Cloud Run

Cloud SQL

App Engine

Kubernetes Engine

App Engine

AWS Elastic BeanStalk

In28 Minutes

- **Simplest way** to deploy and scale your web application in AWS
  - Provides end-to-end web application management
    - Programming languages (Go, Java, Node.js, PHP, Python, Ruby)
    - Application servers (Tomcat, Passenger, Puma)
    - Docker containers (Single and Multi Container Options)
    - **No usage charges** - Pay only for AWS resources you provision
    - **Features**: Load Balancing, Auto scaling and Managed Platform updates
    - Multiple Release Options
      - **All at once** – Deploy V2 to all existing instances in a SINGLE batch.
      - **Rolling** – Deploy V2 to existing instances in multiple batches.
      - **Rolling with additional batch** – Launches a new batch with V2 first. Each batch with V2 will replace existing instances with V1 deployed.
      - **Immutable** – Second ASG created with V2.
      - **Traffic splitting** – Canary testing approach. Deploy V2 to few new instances. Send a portion of traffic to V2 (While serving majority of users from V1).
      - **(ADDITIONAL OPTION) BLUE GREEN with SWAP URL** - Create New Environment with V2 instances. Test them. SWAP URL of V1 environment with V2 environment. One time switch!

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

App Engine

App Engine environments

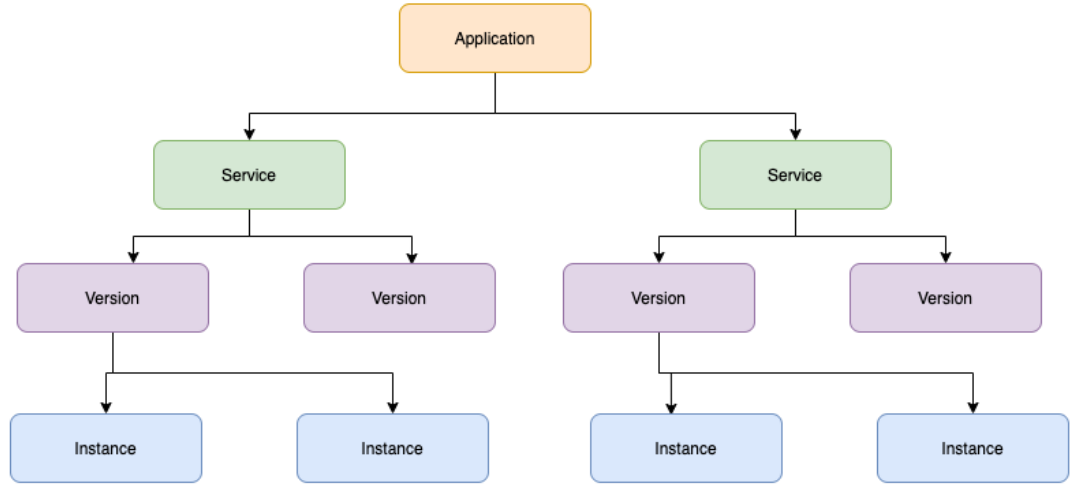
In28 Minutes

- **Standard**: Applications run in language specific sandboxes
  - Complete isolation from OS/Disk/Other Apps
  - **V1**: Java, Python, PHP, Go (OLD Versions)
    - ONLY for Python and PHP runtimes:
      - Restricted network Access
      - Only white-listed extensions and libraries are allowed
    - No Restrictions for Java and Go runtimes
  - **V2**: Java, Python, PHP, Node.js, Ruby, Go (NEWER Versions)
    - Full Network Access and No restrictions on Language Extensions
- **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)
  - Provides access to background processes and local disks

App Engine - Application Component Hierarchy

In28 Minutes

- **Application**: One App per Project
- **Service(s)**: Multiple Microservices or App components
  - You can have multiple services in a single application
  - Each **Service** can have different settings
  - Earlier called Modules
- **Version(s)**: Each version associated with code and configuration
  - Each **Version** can run in one or more instances
  - Multiple versions can co-exist
  - Options to rollback and split traffic



App Engine vs AWS Elastic Beanstalk

In28 Minutes

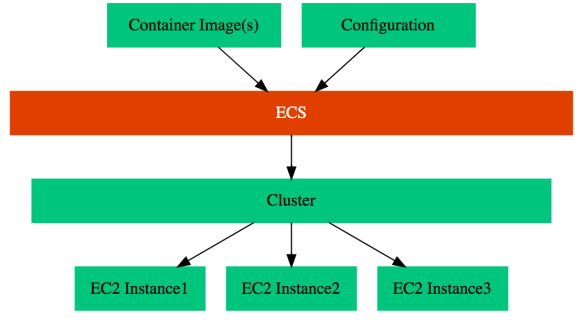
Feature	AWS Elastic Beanstalk	App Engine
Recommended for	Simple Web Apps and Batch Apps	Simple Web Apps and Batch Apps (For simple microservices)
Database Integrations	Amazon RDS, Amazon DynamoDB	Firestore, Cloud SQL
Batch Programs	Worker Tier with SQS integration	Asynchronous task queues - Pub Sub
Hierarchy	Application > Application version > Environment	Application > Service > Version
Run Containers	Yes	Yes (App Engine flexible)
New Releases	Rolling updates, blue/green deployment (using Swap URL)	Rolling updates, blue/green deployment

Google Kubernetes Engine (GKE)

Amazon Elastic Container Service (Amazon ECS)

In28 Minutes

- Microservices are built in multiple languages (Go, Java, Python, JavaScript, etc)
- Containers simplify deployment of microservices:
  - Step I : Create a self contained Docker image
    - Application Runtime (JDK or Python), Application code and Dependencies
  - Step II : Run it as a container any where
    - Local machine OR Corporate data center OR Cloud
- How do you manage 1000s of containers?
  - **Elastic Container Service (ECS)** - Fully managed service for container orchestration
    - **Step I** : Create a Cluster (Group of one or more EC2 instances)
    - **Step II**: Deploy your microservice containers
  - **AWS Fargate**: Serverless ECS. DON'T worry about EC2 instances.
  - **Cloud Neutral**: Kubernetes
    - AWS - AWS Elastic Kubernetes Service (EKS)



Google Kubernetes Engine (GKE)

In28 Minutes

- **Managed** Kubernetes service
  - Provides all important container orchestration features:
    - **Auto Scaling**
    - **Service Discovery**
    - **Load Balancer**
    - **Self Healing**
    - **Zero Downtime Deployments**
- Provides **Pod and Cluster Autoscaling**
- Enable **Cloud Logging** and **Cloud Monitoring** with simple configuration
- Uses **Container-Optimized OS**, a hardened OS built by Google

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:** Add some application configuration for your microservice
  - Config Map - *kubectl **create configmap** todo-web-application-config --from-literal=RDS_DB_NAME=todos*
- **10:** Add password configuration for your microservice
  - Kubernetes Secrets - *kubectl **create secret** generic todo-web-application-secrets-1 --from-literal=RDS_PASSWORD=dummytodos*

Kubernetes Engine

Kubernetes Deployment YAML - Deployment

In28 Minutes

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: hello-world-rest-api
  name: hello-world-rest-api
  namespace: default
spec:
  replicas: 3
  selector:
    matchLabels:
      app: hello-world-rest-api
  template:
    metadata:
      labels:
        app: hello-world-rest-api
    spec:
      containers:
      - image: in28min/hello-world-rest-api:0.0.3.RELEASE
        name: hello-world-rest-api
```

Kubernetes Deployment YAML - Service

In28 Minutes

```yaml
apiVersion: v1
kind: Service
metadata:
  labels:
    app: hello-world-rest-api
  name: hello-world-rest-api
  namespace: default
spec:
  ports:
  - port: 8080
    protocol: TCP
    targetPort: 8080
  selector:
    app: hello-world-rest-api
  sessionAffinity: None
  type: LoadBalancer
```

Kubernetes - A Microservice Journey - The End!

In28 Minutes

Kubernetes Engine

- **11:** Deploy a new microservice which needs nodes with a GPU attached
  - Attach a new node pool with GPU instances to your cluster
    - gcloud **container node-pools create** POOL_NAME --cluster CLUSTER_NAME
    - gcloud container node-pools list --cluster CLUSTER_NAME
  - Deploy the new microservice to the new pool by setting up `nodeSelector` in the `deployment.yaml`
    - nodeSelector: cloud.google.com/gke-nodepool: POOL_NAME
- **12:** Delete the Microservices
  - Delete service - *kubectl **delete service***
  - Delete deployment - *kubectl **delete deployment***
- **13:** Delete the Cluster
  - *gcloud container **clusters delete***

Google Cloud Functions

Going Serverless with AWS Lambda

In28 Minutes

- **Serverless - Don't worry about servers. Focus on building your app**
  - Remember: **Serverless does NOT mean "No Servers"**
  - **Serverless for me:**
    - You don't worry about infrastructure
    - Flexible scaling and automated high availability
    - Pay for use NOT FOR SERVERS
  - **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!
- **AWS Lambda** - Write and Scale Your Business Logic
  - Supports Node.js (JavaScript), Java, Python, Go, C# and more..
  - Don't worry about servers or scaling or availability
  - Pay for Use: Number of requests, Duration of requests and Memory Configured
    - Free tier - 1M free requests per month
  - Integrates with AWS X-Ray(tracing), AWS CloudWatch (monitoring and logs)

AWS Lambda

API Gateway

DynamoDB

Cloud Functions

In28 Minutes

- **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 Functions

AWS Lambda vs Cloud Functions

In28 Minutes

Feature	AWS Lambda	Cloud Functions
Memory	128 MB to 10 GB	128MB to 4 GB
Max Runtime	15 minutes	9 minutes
Pricing	1ms increments Based on memory provisioned	100ms increments Based on memory provisioned
Trigger	Very wide variety	HTTP, Cloud Storage, Pub/Sub, Firebase, Cloud Logging
Exposing REST API	API Gateway or something else	HTTP Trigger
Logging	Amazon CloudWatch	Cloud Logging

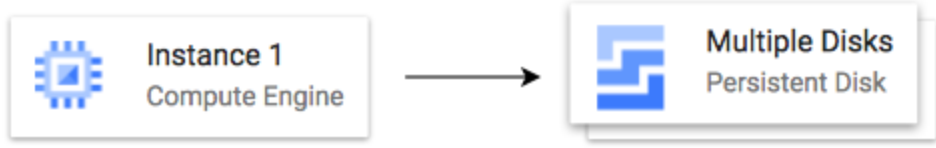
Cloud Run & Cloud Run for Anthos

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

Encryption

Encryption



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

AWS - KMS and Cloud HSM

In28 Minutes

- Generate, store, use and replace your keys(symmetric & asymmetric)
- **KMS**: Multi-tenant Key Management Service
  - **KMS** integrates with all storage and database services in AWS
  - Define key usage permissions (including **cross account** access)
  - **Automatically rotate master keys** once a year
- **CloudHSM**: **Dedicated single-tenant** HSM for regulatory compliance
  - (Remember) AWS KMS is a Multi-tenant service

AWS KMS

Cloud HSM

Cloud KMS vs AWS KMS

In28 Minutes

- **Cloud KMS** is very similar to **AWS KMS**
  - Create and manage **cryptographic keys** (symmetric and asymmetric)
  - **Control their use** in your applications and GCP Services
  - Provides an API to encrypt, decrypt, or sign data
  - Use existing cryptographic keys created on premises
  - **Integrates with almost all GCP services** that need data encryption:
    - Google-managed key: No configuration required
    - Customer-managed key: Use key from KMS
    - Customer-supplied key: Provide your own key
- **Cloud HSM**: FIPS 140-2 Level 3 certified HSMs (Similar to AWS CloudHSM)

Storage

Storage

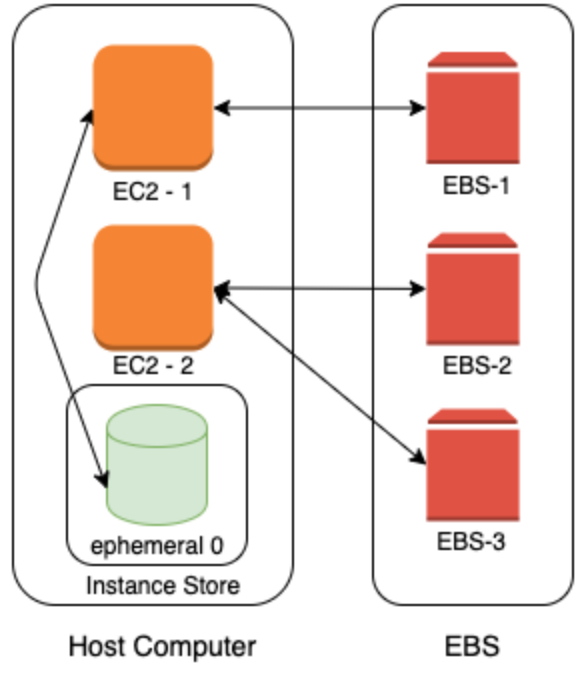
In28 Minutes

Type	AWS	GCP
Persistent Block storage	Amazon Elastic Block Store	Persistent Disk
Ephemeral Block storage	Instance Store	Local SSDs
Object storage	Amazon S3 (Simple Storage Service)	Cloud Storage
Infrequent Access Object Storage	Amazon S3 - Standard-IA, One Zone-IA	Cloud Storage - Nearline and Coldline classes
Archival Object Storage	Amazon Glacier	Cloud Storage - Archive class
File storage	Amazon Elastic File System	Filestore

Block Storage - AWS

In28 Minutes

- Two popular types of Block Storage:
  - **Instance Store**: Physically attached to EC2 instance
    - Ephemeral storage - Temporary data (Data lost - hardware fails or instance termination)
    - Lifecycle tied to VM instance
    - **Use case**: cache or scratch files
  - **Elastic Block Store (EBS)**: Network Storage
    - More Durable. Very flexible **Provisioned capacity**
    - **Increase size when you need it** - when attached to EC2 instance
    - Replicated within the same AZ
    - **Use case** : Run your custom database
    - Support Differential Snapshots (Regional)



Block Storage - GCP

In28 Minutes

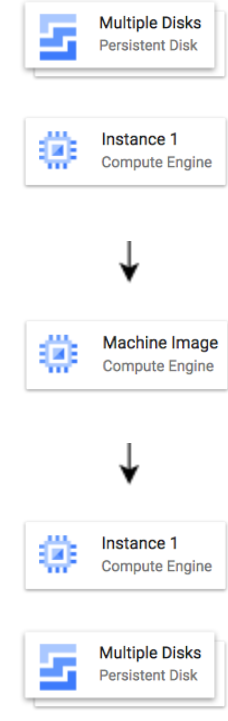
- Two popular types of Block Storage:
  - **Local SSDs**: Physically attached to VM instance
    - Ephemeral storage - Temporary data (Data lost - hardware fails or instance termination)
    - Lifecycle tied to VM instance
    - **Use case**: cache or scratch files
  - **Persistent Disks**: Network Storage
    - Flexible **Provisioned capacity**
    - **Increase size when you need it** - when attached to a VM instance
    - Zonal and Regional options
    - **Use case** : Run your custom database
    - Support Differential Snapshots (Global)

Persistent Disk

Playing with Machine Images

In28 Minutes

- (Remember) **Machine Image** is different from Image
- **Multiple disks can be attached** with a VM:
  - One Boot Disk (Your OS runs from Boot Disk)
  - Multiple Data Disks
- An image is created from the boot Persistent Disk
- HOWEVER, a Machine Image is created from a VM instance:
  - Machine Image **contains everything you need** to create a VM instance:
    - Configuration
    - Metadata
    - Permissions
    - Data from one or more disks
- **Recommended for** disk backups, instance cloning and replication



Object Storage - Cloud Storage

Object Storage in AWS - Amazon S3

In28 Minutes

- **Most popular, very flexible & inexpensive** storage service
- Store large objects using a **key-value** approach
  - Objects are stored in buckets:
    - Bucket names are **globally unique** and used as part of object URLs
      - Can contain ONLY lower case letters, numbers, hyphens and periods
    - Unlimited objects in a bucket
- Provides REST API to access and modify objects
- Provides **unlimited storage**:
  - Objects are **replicated in a single region (across multiple AZs)**
- **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

Amazon S3 - Important Features

In28 Minutes

- Amazon S3 Versioning(**Optional** - Enabled at bucket level):
  - Protects against **accidental deletion**
- How do you save costs and **move files between storage classes**?
  - Solution: S3 Lifecycle configuration
  - Two kinds of actions:
    - **transition** actions (one storage class to another)
    - **expiration** actions (delete objects)
- **Different kinds of data** can be stored in Amazon S3
  - Huge variations in **access patterns**
  - **Trade-off** between access time and cost
  - **S3 storage classes** help to optimize your costs while meeting access time needs
    - Designed for durability of **99.999999999%(11 9's)**

Amazon S3 Storage Classes

In28 Minutes

Storage Class	Scenario
Standard	Frequently accessed data
Standard-IA	Long-lived, infrequently accessed data (backups for disaster recovery)
One Zone-IA	Long-lived, infrequently accessed, non-critical data (Easily re-creatable data - thumbnails for images)
Intelligent-Tiering	Long-lived data with changing or unknown access patterns
Glacier	Archive data with retrieval times ranging from minutes to hours
Glacier Deep Archive	Archive data that rarely, if ever, needs to be accessed with retrieval times in hours
Reduced Redundancy (Not recommended)	Frequently accessed, non-critical data

Cloud Storage

In28 Minutes

- Very similar to Amazon S3
- Serverless: Autoscaling and infinite scale
- 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:
- Objects are stored in buckets
  - Bucket names are **globally unique**
  - Unlimited objects in a bucket
  - Each bucket is associated with a project
  - Max object size is **5 TB**
    - BUT you can store unlimited number of such objects
- Buckets can be in a Region, Multi Region or Dual Region

Cloud Storage

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

- Important to Remember: Unlike AWS, most features are the same across storage classes
- **Low** latency (first byte typically in tens of milliseconds)
- **Unlimited** storage
  - Autoscaling (No configuration needed)
  - **NO minimum** object size
- Same APIs across storage classes
- High durability (99.999999999% annual durability)
- **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

Cloud 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

Cloud Storage

Object Lifecycle Management - Example Rule

In 28 Minutes

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

IAM

Typical identity management in the cloud

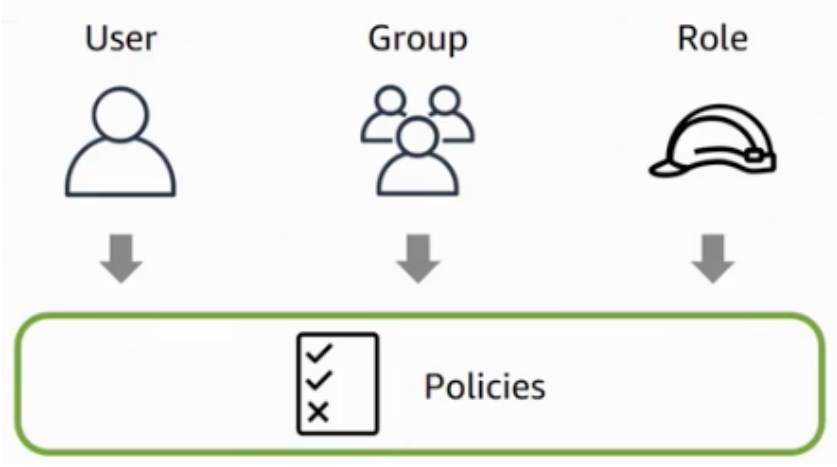
In28 Minutes

- 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?
- Important IAM Concepts:
  - **Authentication** (is it the right user?) and
  - **Authorization** (do they have the right access?)
- In both AWS and GCP, the service to handle this is called IAM (Identity and Access Management)

AWS : Important IAM Concepts

In28 Minutes

- **IAM users**: Users created in an AWS account
  - Have credentials (name/password or access keys)
- **IAM groups**: Collection of IAM users
- **Roles**: Temporary identities
  - Does NOT have credentials attached
  - (Advantage) Expire after a set period of time
  - Use case 1 : EC2 talking with S3
  - Use case 2: Cross Account Access
- **Policies**: Define permissions
  - **AWS managed** - Standalone policy predefined by AWS
  - **Customer managed** - Standalone policy created by you
  - **Inline** - Directly embedded into a user, group or role



Cloud Identity and Access Management (IAM) - GCP

In28 Minutes

- 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
- 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 - GCP vs AWS

In28 Minutes

Question	AWS	GCP
Can IAM manage User Identity?	Yes	No
How can a VM talk with a Cloud Service (Programmatic Identities)?	IAM role	IAM service account
How can you track usage of IAM Identities?	AWS CloudTrail	Audit logging
What is a Policy?	Policy is a set of permissions	Policy is a list of bindings - Each binding binds roles with members

AWS IAM Policy Example

In28 Minutes

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "*", //["s3:Get*","s3:List*"],
            "Resource": "*" //"arn:aws:s3:::mybucket/somefolder/*"
        }
    ]
}
```

- Policy is a JSON document with one or more permissions
  - Effect - Allow or Deny
  - Resource - Which resource are you providing access to?
  - Action - What actions are allowed on the resource?
  - Condition - Are there any restrictions on IP address ranges or time intervals?
  - Example above: AWS Managed Policy : AdministratorAccess
  - Give Read Only Access to S3 buckets - `"Action": ["s3:Get*","s3:List*"]`

GCP 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')",
      }
    }
  ]
}
```

IAM - Roles

In28 Minutes

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

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

Networking

Amazon VPC (Virtual Private Cloud) and Subnets

In28 Minutes

- **VPC(Virtual Private Cloud)** - Your **own isolated network** in AWS cloud
  - Network traffic within a VPC is isolated (not visible) from all other Amazon VPCs
  - You **control all the traffic** coming in and going outside a VPC
  - **(Best Practice)** Create AWS resources **in a VPC**
    - Secure resources from unauthorized access AND
    - Enable secure communication between your cloud resources
    - Each VPC is created in a Region
- **Subnet - Separate public resources from private resources** in a VPC
  - **Create different subnets** for public and private resources
    - Resources in a public subnet **CAN** be accessed from internet
    - Resources in a private subnet **CANNOT** be accessed from internet
    - BUT resources in public subnet can talk to resources in private subnet
    - Each Subnet is created in an Availability Zone
    - VPC - us-east-1 => Subnets - AZs us-east-1a or us-east-1b or ..

Google Cloud VPC (Virtual Private Cloud) and Subnets

In28 Minutes

- VPC and Subnets in GCP are very similar to AWS:
  - Default VPCs are provided with default subnets
  - You can create your own custom VPCs and subnets
  - You can create different types of resources in different subnets
  - You can use VPC flow logs to debug problems
  - You can setup peering between VPCs to connect different VPCs
  - You can create Shared VPCs to share resources across multiple GCP projects or AWS accounts
- However, it is important to note the differences:
  - In AWS, VPCs are regional and Subnets belong to an Availability Zone
    - However, in GCP VPCs are global and Subnets belong to specific region
  - There are significant differences in default VPCs and subnets configuration
    - In GCP, default VPCs are created per project
      - Each default VPC has multiple subnets - one in each region

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

- 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

Cloud Operations

Operations

In28 Minutes

Operation	AWS	GCP
Monitoring	Amazon CloudWatch	Cloud Monitoring
Logging	Amazon CloudWatch Logs	Cloud Logging
Audit logging	AWS CloudTrail	Cloud Audit Logs
Debugging	AWS X-Ray	Cloud Debugger
Performance tracing	AWS X-Ray	Cloud Trace

Cloud Monitoring

Cloud Logging

Trace

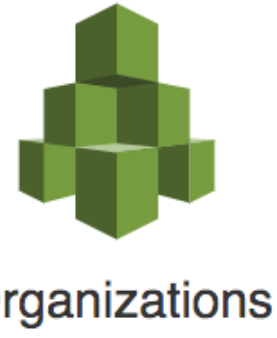
Debugger

Organizing GCP Resources

Organizing Resources in AWS

In28 Minutes

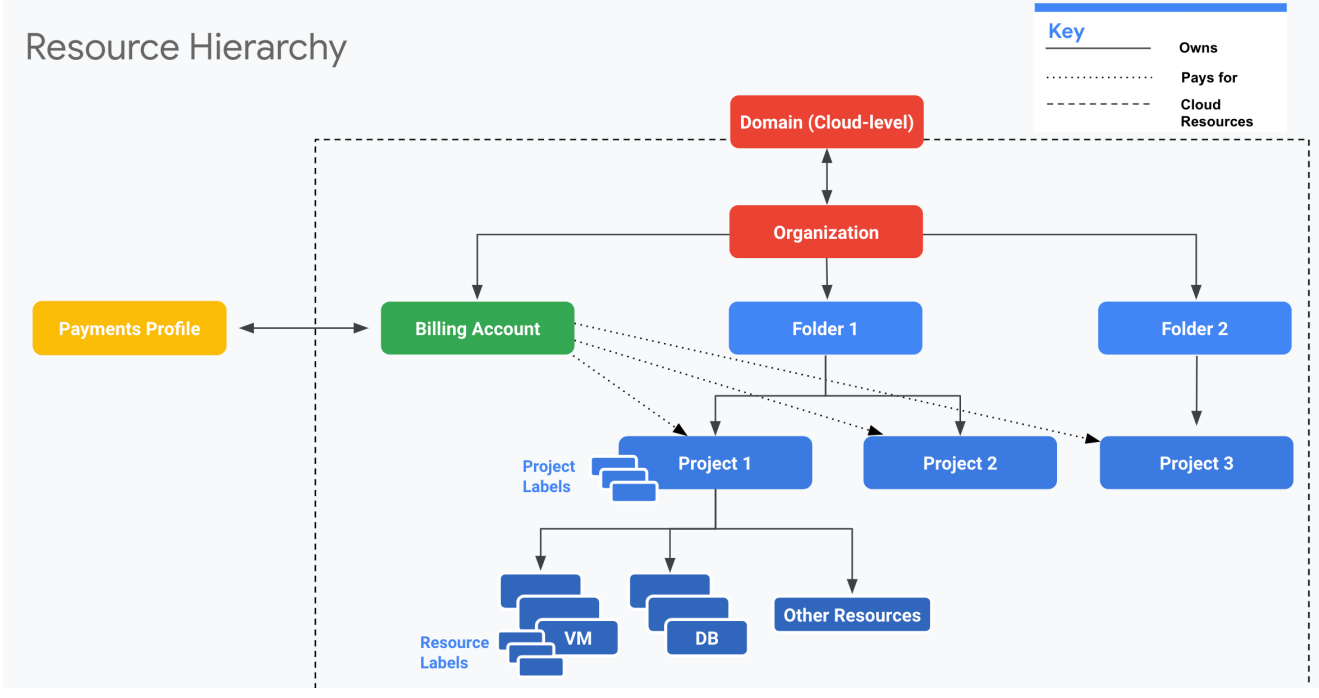
- Where do we create resources in AWS?
  - In an AWS Account!
  - By default, you will be billed per AWS Account!
- What if you want to create resources for multiple environments?
  - One of the recommended approaches is to create separate AWS accounts
    - Each AWS account provides natural security, access and billing boundaries
    - Create **AWS Organization** to organize accounts into Organizational Units (OU)
      - Consolidated bill for AWS accounts
    - Use **AWS Resource Access Manager** to share AWS resources:
      - Share AWS Transit Gateways, Subnets, AWS License Manager configurations etc



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

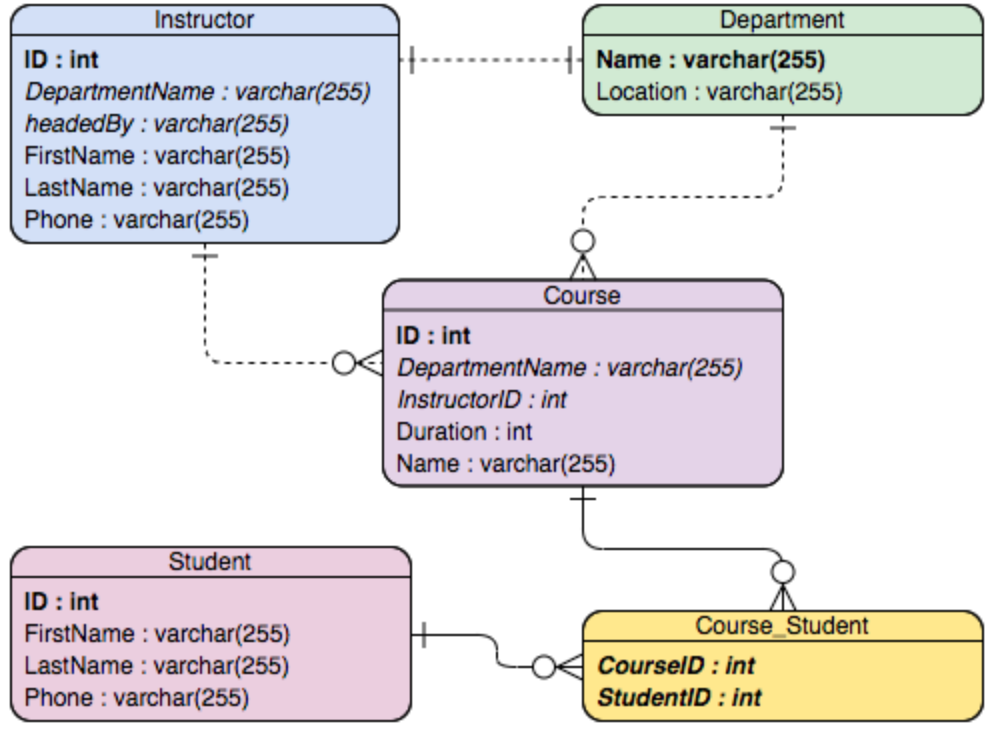
In28 Minutes

- 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!

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 reads & updates)
  - **Use cases**: Most traditional applications, ERP, CRM, e-commerce, banking applications
- **Popular databases**: MySQL, Oracle, SQL Server etc
- Recommended AWS Services
  - **Amazon RDS** (Relational Database Service) - (Amazon Aurora, PostgreSQL, MySQL, MariaDB (Enhanced MySQL), Oracle Database, and SQL Server)
    - Amazon Aurora Provides "Global Database" option
- Recommended GCP 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)

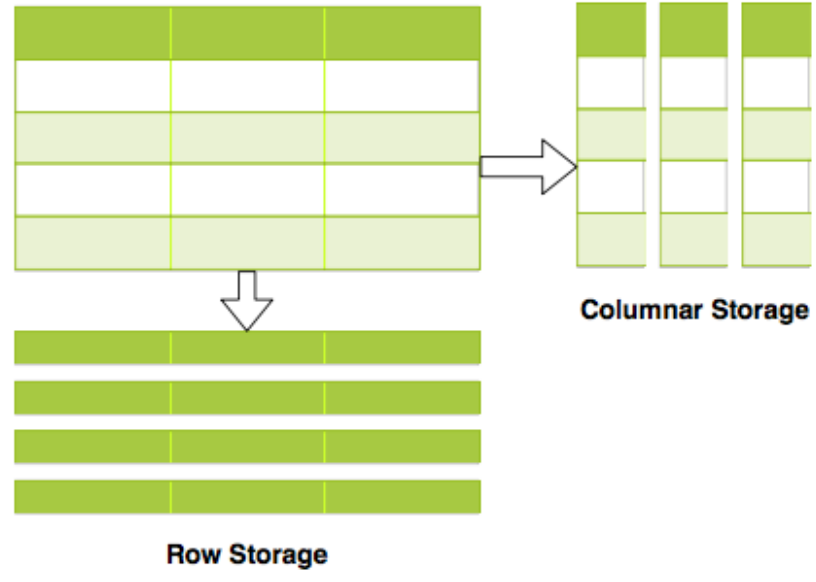
In28 Minutes

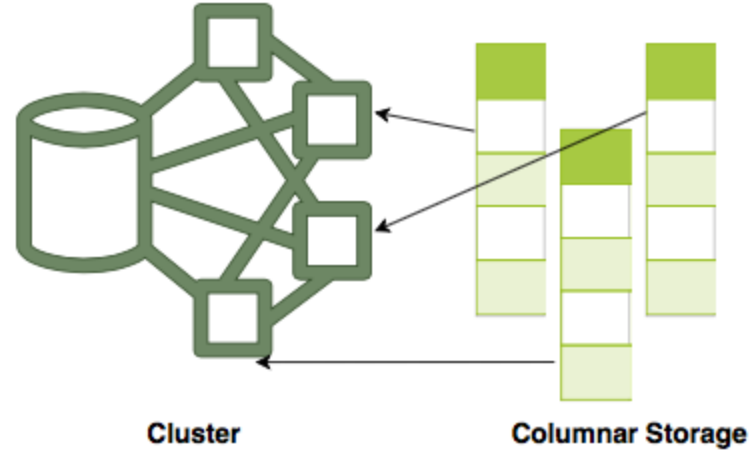
- Applications allowing users to **analyze petabytes of data**
  - **Examples** : Reporting applications, Data ware houses, Business intelligence applications, Analytics systems
  - **Sample application** : Decide insurance premiums analyzing data from last hundred years
  - Data is consolidated from multiple (transactional) databases
- Recommended AWS Managed Service
  - **Amazon Redshift**: **Petabyte-scale** distributed data ware house
- 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"
- **AWS Managed Services**:
  - Amazon DynamoDB & Amazon DocumentDB
- **Google Managed Services**:
  - Cloud Firestore (Datastore) & Cloud BigTable

Datastore

BigQuery

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)

Datastore

BigQuery

In-memory Databases

In28 Minutes

Memorystore

- **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 AWS Managed Service
  - **Amazon ElastiCache**
- Recommended GCP Managed Service
  - **Memorystore**
- Both **Amazon ElastiCache** and **Memorystore** support Redis and Memcached
- **Use cases** : Caching, session management, gaming leader boards, geospatial applications

Databases

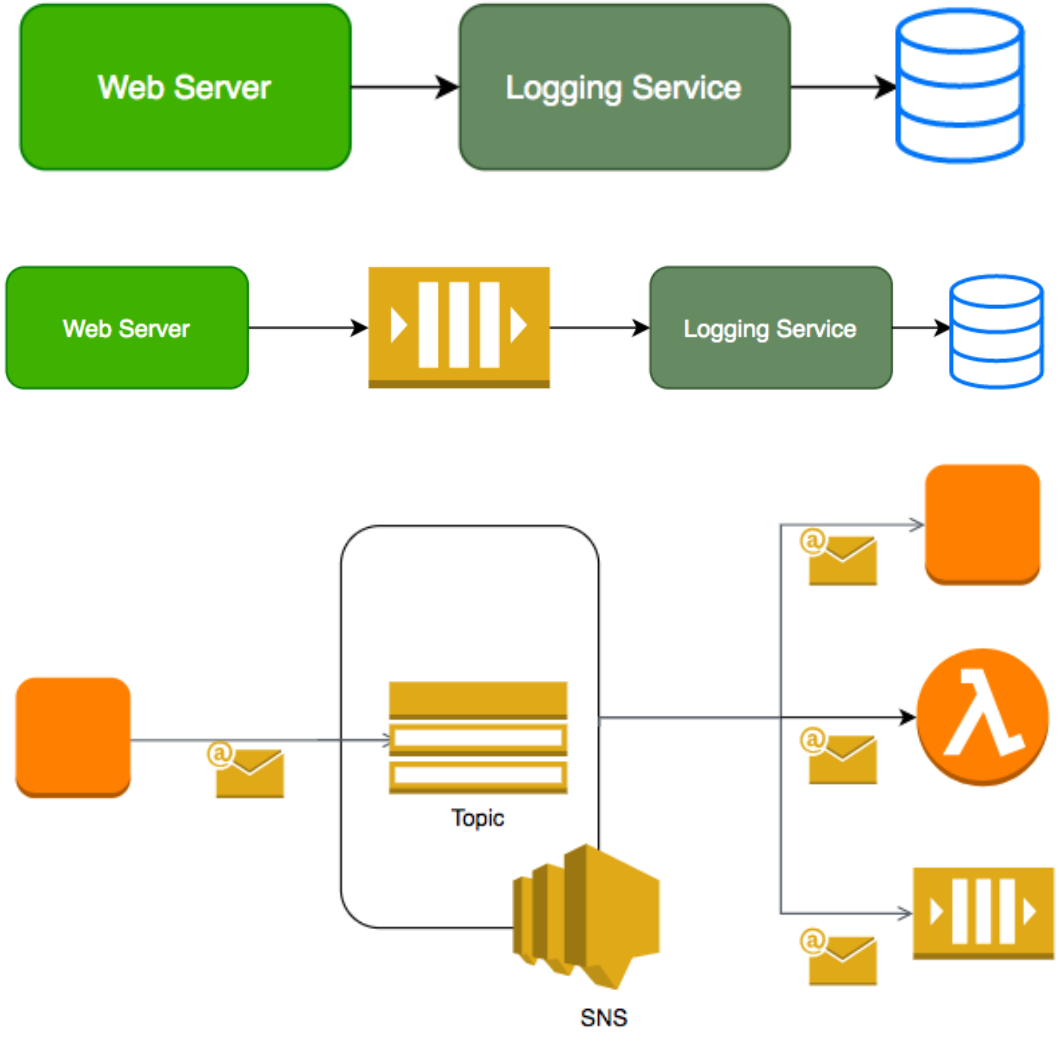
In28 Minutes

Type	AWS	GCP
RDBMS	Amazon Relational Database Service, Amazon Aurora	Cloud SQL, Cloud Spanner
NoSQL	Amazon DynamoDB, Amazon DocumentDB	Datastore/Firestore, Cloud Bigtable
In-memory	Amazon ElastiCache	Memorystore
Data warehouse	Amazon Redshift	BigQuery

Synchronous vs Asynchronous Communication

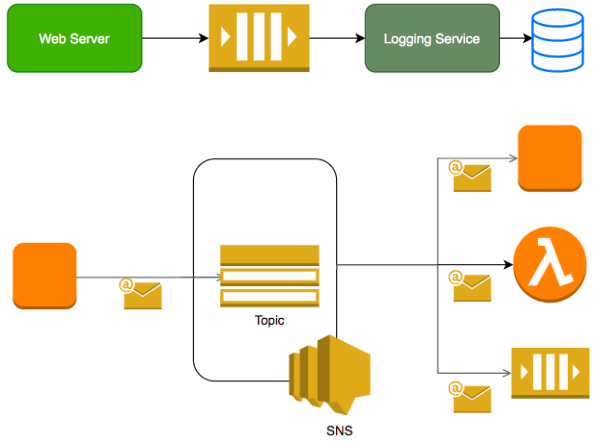
In28 Minutes

- Synchronous Communication:
  - What if your logging service goes down?
    - Will you applications go down too?
  - What if there is high load?
    - Log Service unable to handle and goes down
- Asynchronous Communication:
  - Create a queue or a topic
    - Your applications put the logs on the queue
    - Picked up when the logging service is ready
  - Good example of decoupling!
  - (Possible) Multiple logging service instances reading from the queue!
- Two Types:
  - Pull based
  - Push based



AWS - Amazon SQS and Amazon SNS

- Pull Based: **Amazon SQS**
  - Producers put messages. Consumers poll on queue.
    - Only one of the consumers will successfully process a message
  - Standard Queue
    - Unlimited throughput
    - BUT NO guarantee of ordering (Best-Effort Ordering)
    - and NO guarantee of exactly-once processing
  - FIFO (first-in-first-out) Queue
    - First-In-First-out Delivery, Exactly-Once Processing
    - BUT throughput is lower
- Push Based: **Amazon SNS**
  - 1: Create an SNS Topic
  - 2: Subscribers can register for a Topic
  - 3: When an SNS Topic receives an event notification (from publisher), it is broadcast to all Subscribers



Messaging in GCP - Pub/Sub

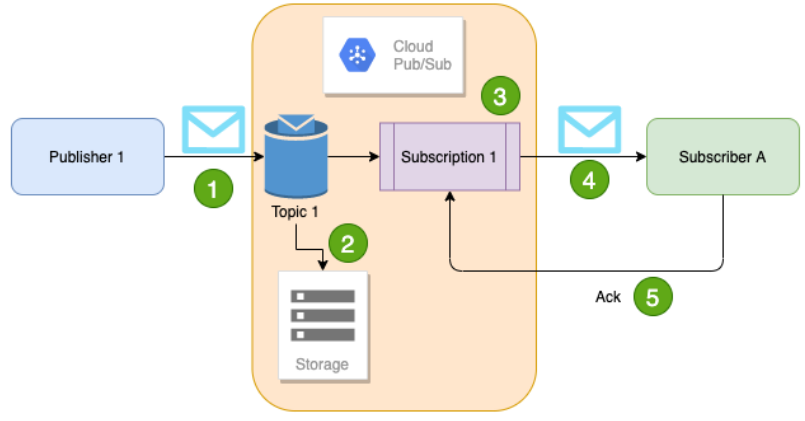
In28 Minutes

- 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)
- Supports push and pull message deliveries
- Pub/Sub does NOT guarantee message ordering
  - BUT there are mitigation options --enable-message-ordering
- Pub/Sub does NOT guarantee exactly-once delivery
  - Applications can handle duplicate messages

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



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 learn GCP
  - Goal of the course is to provide you with a quick start to GCP
- 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 me on LinkedIn (Ranga Karanam)
  - Share your success story and lessons learnt in Q&A with other learners!

What next?

- Go Deeper into GCP!
  - Get Certified
    - Associate Cloud Engineer
    - Professional Cloud Architect
- 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

Cloud SQL

Cloud Spanner

BigQuery