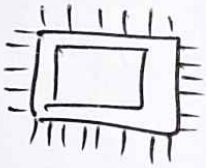
"AWS - Amazon Web Services"

Cloud Practitioner CLF-C01
(CCP)

Server :

Composed of

- CPU : Compute
- RAM : Memory



- Storage data

- Database : Stored data in a Structured way

- Network : Routers, Switch, DNS Server

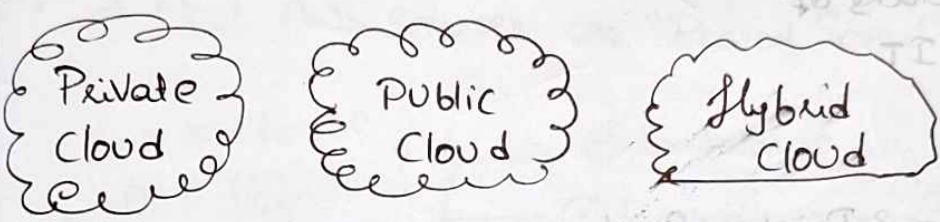
DATA CENTER : Servers + Servers + Servers

All the Traditional IT Approach can be externalize with "CLOUD"

Cloud Computing:

"on-demand delivery" of compute power, database storage, applications and other IT resources.

- Pay as you go Pricing.
- Can access as many resources as you need "INSTANTLY"



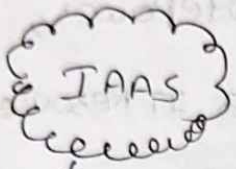
TOP 5 Characteristics of Cloud Computing

→ On-demand Self Service
→ Broad Network Access
→ Multi-tenancy and Resource Pooling
→ Rapid Elasticity & Scalability.
→ Measured Services.

Advantages of Cloud Computing

→ Trade (CapEx) for (OpEx)
→ Economies of Scale.
→ Stop guessing the capacity
→ Increase Speed and Agility.
→ Stop Costs on Running & Maintaining Data Centers.

AWS Global Infrastructure.

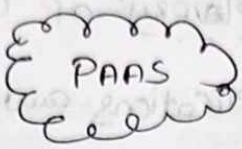


↓
EC2

* N/w, Computers, Data Storage Space.

↓

* Building Blocks of Cloud IT



↓ Elastic Beanstalk

* FOCUS on deployment & Management of APPlications

↓ Rekognition.

* complete Product that is run and Managed by Service Provider

→ AWS - 3 Pricing fundementals
(PAY - AS - you - GO)

* Compute - Pay for Compute Time
* Storage - Pay for data Stored in cloud
* Networking - Pay for only DATA Transfer OUT of the Cloud

AWS Global Infrastructure.

- AWS Regions
- AWS Availability zones
- AWS Data Centers
- AWS Edge Locations / Points of Presence

AWS Regions: → Regions are all around the world.

→ Names can be Asia Pacific (Mumbai) ap-south-1
US East (N. Virginia) us-east-1
US East (Ohio) us-east-2

→ Region is a cluster of "DATA CENTERS"

→ Most AWS services are "REGION Scoped"

How to Choose AWS Region ?

- Compilance with data governance & Legal requirents
- Proximity to customers: Reduced latency
- Available services within a Region
- Pricing

AWS Availability Zones

→ Each region has many Availability zones (usually '3' Min-2 Max-6)

AWS Region
Sydney: ap-southeast-2

AZ | ap-southeast-2a

AZ | ap-southeast-2b

AZ | ap-southeast-2c

→ Each (AZ) is one (a) more discrete data centers with Redundat Power, Networkig and Connectivity.

AZ Connected with High Bandwidth, uHra low latency Networking

AWS Edge locations:

↳ 216 Points of Presence (205 edge locations & 11 Regional Caches) in 84 cities across 42 countries

→ Content is delivered to end users with lower latency.

AWS has

Global Services

- → IAM
- → Route 53 (DNS)
- → Cloud Front (Content Delivery N/w)
- → WAF (Web Application Firewall)

Region Scoped

- → Amazon EC2 (IAAS)
- → Elastic Beanstalk (PAAS)
- → Lambda (Function AAS)
- → Rekognition (SAAS)

"REGION TABLE"

→ **Shared Responsibility:**

As Customer - Responsible for Security IN the Cloud

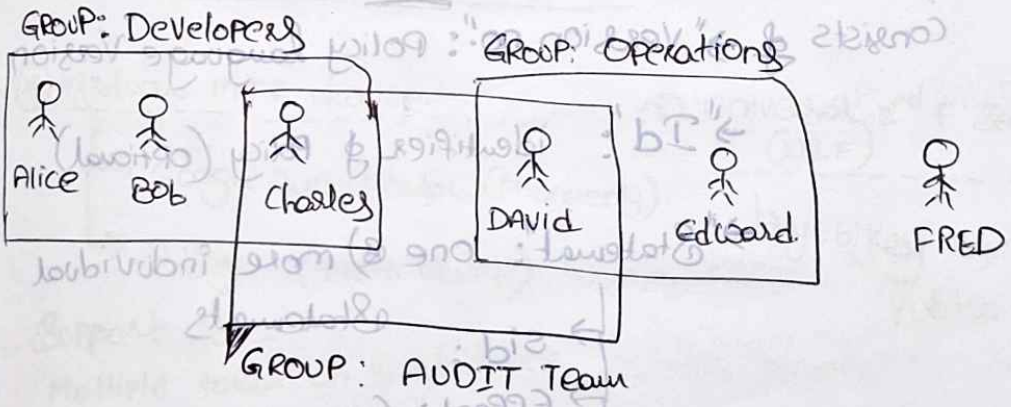
As AWS - Responsible for Security OF the Cloud

→ IAM - Identity and Access Management

↓

- Users & Groups → GLOBAL SERVICE

* ROOT ACCOUNT - Created by default, Shouldn't be used / Shared
* USERS - PPl within Organisation and Can be grouped
* GROUPS - only Contain USERS - Not other groups
* USERS doesn't have to belong to a GROUP & USER Can belong to Multiple GROUPS

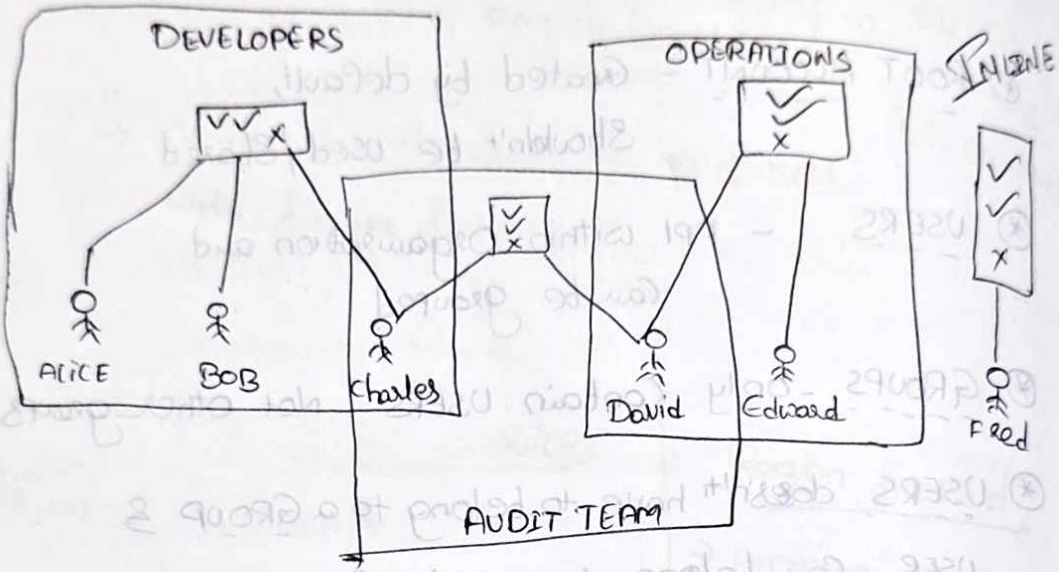


IAM: Permissions

* USERS & GROUPS Can be assigned "POLICIES"
  ↓
  'JSON DOCS'
* POLICIES → define Permissions of "USERS"

→ LEAST PRIVILEGE PRINCIPLE

→ IAM POLICIES INHERITANCE



IAM Policies Structure

↓ JSON DOC

Consists of → "Version no": Policy language Version

→ "Id": identifier of Policy (optional)

→ "Statement": one or more individual Statements

- → Sid:
- → Effect: (Allow / Deny)
- → Principal: Account / user / Role to which policy applied to
- → Action:

→ "Resource":

→ "Condition":

IAM - MFA (MULTI FACTOR Authentication)

IAM - Password Policy.

↳ U can setup a Password Policy
- Min length
- Require specific char types
- Allow all IAM users to change thier own passwords
- Password Expiration
- Prevent Password Reuse

IAM - MFA - Password U know + Security device U own + Sometimes which U are

MFA device options

* Virtual MFA device.
  - ↳ Google Authenticator (Phone only)
  - ↳ Authy (Multi device)
  - Support for Multiple tokens on single device

* Universal 2nd F Security Key (U2F)
  - ↳ yubikey by Yubico (3rd party)

* Hardware Key Fob MFA device by Gemalto

* Hardware key Fob MFA for AWS GovCloud (US) by Sure PassID

→ AWS - USER ACCESS

3 options

- → AWS Management Console - (Password + MFA)
- → AWS Command Line Interface - (CLI) (Protected by Access keys)
- → AWS Software Developer kit - for Code APIs (SDK) Protected by Access keys

* Access keys are generated through - AWS Console
  - → Secret like Password

Access Key ID ≃ UserName

Secret Access Key ≃ Password

Azure Cloud Shell: Available in some regions.

IAM Role for Services:

→ Some AWS service will need to perform actions on our behalf
- To do so, we will "Assign Permissions" to AWS services with "IAM Roles"

Common Roles:
- EC2 Instance Roles
- Lambda Function Roles
- Roles for cloud Formation

→ IAM Security Tools

- IAM Credentials Report (account-level) → All account's users and status of various credentials.
- IAM Access Advisor (user-level)
  - Shows the service permissions granted to a user and when those services were last accessed
  - Use this information to revise your policies.

→ IAM Guidelines & Best Practices:

- Don't use the root account except for AWS Account Setup
- One Physical User = One AWS User
- Assign users to group and assign permission to group
- Create a strong Password Policy
- Use & Enforce MFA
- Create & use Roles for giving permissions to AWS Services.
- Use Access Keys for Programmatic Access (CLI/SDK)
- Use IAM Credentials Report often
- Never share IAM users / Access keys / Passwords.

→ Shared Responsibility for IAM

AWS	YOU
• Infrastructure (global N/w Security)	- Users / Groups / Roles / Policies Management & Monitoring
• Configuration & Vulnerability Analysis	- Enable MFA on all Accounts
• Compilance Validation	- Rotate all your keys often
	- Use IAM tools to apply appropriate Permissions
	- Analyse Access Patterns and Review Permissions

→ AWS Budget Setup:

↓

" Billing and Cost Management Dashboard "

⊛ Root user need to "ACTIVATE IAM ACCESS" in IAM user and Role Access to Billing information in My Billing Dashboard.

⊛ we Can view usage of AWS Free Tier "AWS Billing"

Budget Alert
- Cost Savings Budget
- Usage Budget
- Service Planning Budget
- Reservation Budget

→ EC2 - Elastic Compute Cloud

AMAZON EC2 - (IAAS)

Mainly consists of

↳ Renting Virtual Machines (EC2)

↳ Storing data on Virtual drives (EBS)

↳ Distributing load across Machines (ELB)

↳ Scaling of services using an Auto-Scaling Group (ASG)

EC2 ↳ Sizing & Configuration options

— OS (Linux, Windows, MacOS)

— CPU (Compute Power & Cores)

— RAM

— Storage Space ← N/w attached (EBS & EFS)
Hardware (EC2 instance Store)

— Network Card (Speed of Card, Public IP address)

— Firewall Rules (Security group)

— Bootstrap Script: Configure at first launch (EC2 User data)

→ EC2 User data: Runs with "ROOT USER"

↳ Bootstrap instances → launching Comands when Machine starts

→ Script Runs only once at instance "First start"

AMI - Amazon Machine Image

→ EC2 Instance Types ⟶ 7 different Types
  ↳ General Purpose
  → Compute Optimized
  → Memory optimized
  → Accelerated Computing
  → Storage optimized
  → Instance features
  → Measuring Instance Performance

VPC - Virtual Private Cloud.

EC2 Instance Type Naming Convention.

" m 5. 2x large "

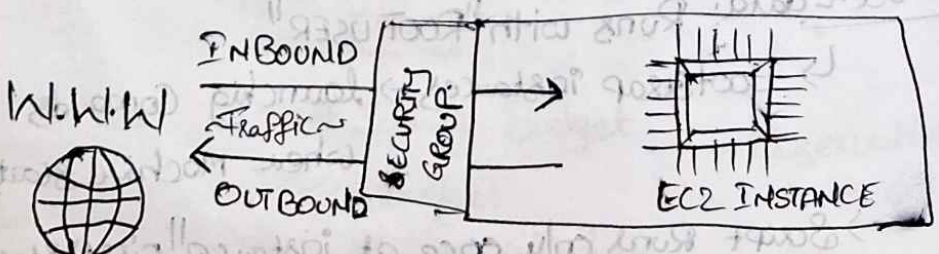
→ m : instance class/type
→ 5 : generation (AWS improves them over time)
→ 2x large : size with in the instance class

t2.micro → free tier → General Purpose

→ Security Groups: Fundemental of N/w Security

  ↳ ⊛ Control how Traffic is allowed into & Out of our EC2 instances

  ⊛ Security Groups only Contain "ALLOW" Rules



Security Group as → FIREWALL on EC2 Instances

↳ Regulate - Access to Ports
"PROTOCOL" - Authorised IP Ranges - IPV4 & IPV6
- Control of IB & OB Network

Security Groups

Important Points to Know

→ Can be attached to multiple Instances & Instance can have multiple Security groups too
→ Locked down to a region/VPC Combination
→ It's good to maintain one seperate SG for SSH access
→ ALL IB traffic is "Blocked" by default
→ ALL OB traffic is "Authorised" by default
→ Referenced from the other Security groups

PORTS to Know

→ 22 = SSH (Secure Shell)
→ 21 = FTP (File transfer Protocol)
→ 22 = SFTP
→ 80 = HTTP
→ 443 = HTTPS
→ 3389 = RDP

→ EC2 - Instance Roles:

* Use IAM Roles to attach the Permissions for EC2-Instances

EC2 - Instance Purchasing options

* Ondemand instances – Short workload, Predictable, Pay by second
* Reserved (1 & 3 yrs) – long workloads
  - Reserved Instances
  - Convertible RI – ↓ with flexible Instances
* Savings Plan (1 & 3 yrs)
  - Commitment to amt of usage, long workload
* Spot Instance – Short workloads / less Reliable
* Dedicated Hosts – Book an Entire Physical Server, Control Instance Placement
* Dedicated Instances – No other customers will share ur Hardware
* Capacity Reservations – Reserve Capacity in specific AZ for any duration.

Shared Responsibility for EC2

AWS	You
- Infrastructure (Global N/w Security)	- Security Group Rules
- Isolation on physical hosts	- OS Patches & updates
- Replacing faulty H/W	- S/W & utilities installed
- Compliance Validation	- IAM Roles Management
	- Data Security

SUMMARY EC2

EC2 instance -

→ AMI (OS) + Instance Size (CPU + RAM)
+ Storage + Security Groups + EC2 User data

→ EC2 - Instance Storage:

→ EBS VOLUME:

Elastic Block Store Volume

↓

- Network drive - Attached to instances while they RUN.
- Bound to a specific Availability Zone.
- Can be mounted to One Instance at a Time
- Uses Network to Communicate the instance So little bit of latency.
- Can be detached from an EC2 instance and attached to another one quickly.
- Snapshots Can be Created.
- Have a Provisioned Capacity.
- ~~Multiple Ins~~ Multiple EBS Can be mapped to Single Instance.
- Delete on Termination - Attribute EBS

→ EBS Snapshots: ⊛ Make a backup (Snapshot) of EBS Volume at any point in time

⊛ Not necessary to detach volume to do Snapshot But Recommended

⊛ Can Copy Snapshots across AZ (&) Region

Some Features

→ EBS Snapshot Archive → "Archive Tier" 75% Cheaper
↳ Takes 24 - 72 hrs for restoring

→ Recycle Bin for EBS Snapshots
- Setup rules to retain deleted Snapshots (Accidental deletion)
- Specify retention (from 1 day to 1 yr)

→ AMI - Amazon Machine Image:

↳ Customisation of an EC2 instance
- Add ur own s/w config, O.S, Monitoring
- Faster Boot / Config time

↳ AMI are built for "SPECIFIC REGION"

⊛ Public AMI
⊛ Your own AMI
⊛ AWS Market Place AMI

→ EC2 Image Builder.

↳ used to Automate Creation of VM's & Container Images

→ Automate the Creation, Maintain, Validate & Test EC2 AMI's

→ Can be Run on a Schedule.

→ Free Resource (only Pay for underlying resources)

→ EC2 INStance Store

→ EBS volumes are Nw drives with good but "limited" Performance

EC2 Instance Store → High Performance Hardware disk.

→ Better I/O performance

→ Ephemeral Storage

→ Good for Buffer/Cache/Scratch data/Temporary content

→ Risk of data loss if H/w fails

UDM370731

→ EFS - Elastic File System

→ Manage NFS Can be mounted on 100s of EC2 (Shared)

→ works with Linux EC2 instances / Multiple AZ

→ high Available, Scalable, Expensive (PAY PER USE)

/ EFS - IA / → Cost Optimised. - Life cycle Policy.

→ AMAZON FSX - EVE

↳ To launch 3rd Party High Performance File Systems on AWS.

→ Fully Managed Service

FSX for Lustre	FSx for windows File Server	FSX for NETAPPONTAP
(HPC High Performance Computing) (Linux cluster)	(NFS - Shared File System) SMB, NTFS	

→ ELASTIC LOAD BALANCING & AUTO SCALING GROUPS

Scalability & High Availability

↓

@ APP/System can handle greater loads by Adapting

- Horizontal Scalability (= Elasticity)
- Vertical Scalability (increasing size of instance)

Vertical Scalability:

- increasing the size of the Instance
- common for non-distributed systems such as database
- usually a limit to how much you can Vertical Scale (Hardware limit).

Horizontal Scalability

- increasing the no of instances/systems for your Applications
- implies Distributed Systems.
- Very common for web Applications / modern Applications

→ High Availability:

- Running Application / System in Atleast 2 AZs
- Goal of high availability is to Survive Disaster / Loss.

→ Scalability
→ Elasticity
→ Agility.

→ ELB - Elastic Load Balancing

- Load Balancing are Servers that forward internet traffic to multiple servers (EC2 Instances) downstream

Managed
LOAD BALANCER

- Spread load across multiple downstreams
- Single Point of Access (DNS) to our App
- Seamlessly handle failures of downstream instances
- Do regular health checks to our instances
- Provide SSL Termination (HTTPS) for our websites
- High Availability across zones

3 Kinds of Load Balancers by AWS

① Application load Balancers - layer 7
(HTTP/HTTPS only)

② Network load Balancer - layer 4
(Ultra-High Performance, Allows for TCP)

③ Classic load Balancer - layer 7 & 4
(Slowly retiring)

"GATEWAY LOAD BALANCER"

AUTO SCALING GROUP – ASG

↳ Real life - load on APPS/sites can change.

→ In cloud, we can create/delete servers easily.

Goal of ASG

- Scale out (add EC2 instances) to match high load
- Scale in (remove EC2 instances) to match decreased load
- Ensure we have min/max no. of machines running
- Automatically register new instances to a load balancer
- Replace unhealthy instances.

- Cost Savings: only running at optimal capacity.

ASG → Manual Scaling

Scaling Strategies → Dynamic Scaling

↳ Simple/Step Scaling

↳ Target tracking scaling

↳ Schedule Scaling

→ Predective Scaling.

→ AMAZON · S3 – Cloud Native Storage

* S3 is one of main building blocks of AWS
* "Infinitely Scaling Storage"

S3 use cases

* Backup & Storage → NASDAQ stores 7yrs data into S3 Glacier
* Disaster Recovery
* Archive
* Hybrid Cloud Storage → SYSCO Runs Analytics and gain Business insights
* Application Hosting
* Media Hosting
* Data lakes & Big Data Analytics
* Software Delivery
* Static website

S3: → Allow PPL to store **Buckets** "Objects" (files) in "Buckets" (directories)

→ Buckets must have "GLOBALLY UNIQUE NAME" (Across All regions, All accounts)

→ Buckets are defined at regional level.

→ Naming Convention
- No uppercase
- No underscore
- 3-63 chars.
- Not an IP
- Must start with lowercase letter/number

S3 - objects : ① Objects (files) have a Key

② "Key" is the full Path:

S3://my-bucket/my_file.txt

"Prefix + Object Name"

③ No concept of "directories"

- object Values are Content of body
  - Max object size is 5TB (5000GB)
  - if uploading > 5GB :- must use "Multi-Part upload"
- Metadata (list of text Key/Value Pairs
  - System (or) user_metadata)
- Tags (Unicode key/Value Pair - upto 10)
- Version ID

S3 - Security:

- user Based - IAM Policies
- Resource Based - Bucket Policies
  (Bucket wide rules from S3 console. allows Across Account)
  - Object ACL - finer_grain
  - Bucket ACL - less common
- Encryption

S3 Bucket policies - JSON based Policies.

S3 Websites:

- S3 can host static websites and have them accessible on www
- S3 website URL will be
  - <bucket-name>.s3-website-<AWS-region>.amazonaws.com

AMAZON S3 - Versioning:

- U can version ur files in Amazon S3
- Enabled at Bucket level - Version: 1, 2, 3...
- Protect against unintended deletes (Ability to restore version)
- Easy roll back to previous version

S3 Access Logs:

- For Audit, U may want to log all Access to S3 Buckets
- Any request made to S3 from Any account Authorised / denied will be logged into another S3 Bucket
- Data can be analysed using Data Analysis tools

S3 Replication:

↳ Cross Region Replication (CRR)
↳ Same Region Replication (SRR)

- Must enable Versioning in Source & Destination
- Buckets can be different accounts
- Copying is Asynchronous.
- Must give Proper IAM Permissions to S3
- CRR - use cases: Compilance, lower latency access, Replication across accounts
- SRR - use cases: log Aggregation, live replication between Prod & Test acounts

S3 Storage Classes:

Amazon S3 Standard - General Purpose
S3 Standard - Infrequent Access
S3 Onezone - Infrequent Access
S3 Glacier - Instant Retrieval
S3 Glacier - Flexible Retrieval
S3 Glacier - Deep Archive
S3 Intelligent Tiering

Can move b/w classes Manually (&) using S3 Lifecycle configurations.

S3 Durability & Availability

- **Durability:** High Durable (99.999999999%)
  - Same for all Storage classes
- **Availability:**
  - Measures how readily available a service is
  - Varies depending on Storage class

99.99% - S3 ~~was~~ mins available

S3 Object Lock:

- WORM (write once Read Many)
- Block an Object Version deletion for a specified amt of time.

Glacier Vault Lock

- WORM
- Lock the Policy for future edits
- Compilance & data Retention requirements.

S3 Encryption:

No Encryption | Server Side Encryption | Client Side Encryption

S3 - Shared Responsibility

AWS	User
• Infrastructure (Global Security, Durability, Availability, sustain concurrent loss of data in 2 facilities)	• S3 Versioning
• Configuration & Vulnerability Analysis	• S3 Bucket policies
• Compilance Validation.	• S3 Replication Setup
	• Logging & Monitoring
	• S3 Storage classes
	• Data Encryption at Rest & In Transit

AWS SNOW FAMILY:

↳ Highly Secure, Portable devices to collect and Process data at the edge and migrate the DATA INTO AND OUT OF AWS

* Data Migration : SNOW CONE / SNOWBall Edge / SNOWMobile
* Edge Computing : SNOW CONE / SNOWBall Edge.

AWS SNOW FAMILY: OFFline devices to Perform data Migrations.

SNOWBALL Edge:

- Physical data Transport Solution
  : move TBs, PBs of data in & out
- Pay Per data transfer JOB
- Provide Block Storage & S3-object storage

- Snowball Edge Storage Optimised
  80TB HDD
- Snowball Edge Compute optimised
  42TB HDD

USE CASES:- Large Data Migrations
- DC decommisions
- Disaster Recovery (BACKUP in AWS)

AWS SNOWCONE:

- Small, Portable Computing, Anywhere, Rugged & Secure withstands harsh ENVs
- Device used for Edge Computing, Storage & data transfer
- 8TB of usable storage.

"AWS DATA SYNC"

AWS SNOWMobile (Truck)

- Transfer Exabytes of data
  (1EB = 1000 PB = 1000,000 TBs)
- Each Snowmobile has 100PB of Capacity.
  (use Mutiplein Parallel)
- High Security : temp Controlled, GPS, 24/7 Video Cameras

Edge Computing:

↳ Process data while it's being created on an 'edge-locations'

→ A truck on Road, A ship on a sea, Mining station under ground

↓

" limitd / no internet Access."

⊗ No access to Computing Power

For Edge Computing

- we can setup a Snowball Edge / SnowCone to do Edge Computing.

AWS OPS HUB:

- To use a Snow Family devices you needed a 'CLI Tool'

Now you can use 'AWS OPS HUB' (Software you install on Computer/Laptop) to Manage your Snow family device.
(GRaphical Interface)

Hybrid Cloud for Storage:

S3 → AWS Storage Gateway.

⊛ Bridge b/w on premise data & cloud data in S3

DATABASES

- for More Structured data
- Indexes for Efficiatly query/search data
- Define Relationships b/w or datasets

→ Relational Databases: Can use 'SQL' language to perform queries/lookup

→ NOSQL Databases: (Non-relational databases)

↳ Are built for specific data models and have flexible schemas for building modern applications.

→ Flexible, Scalable (Scale-out by using distributed clusters)
High Performance, High functional

* KEY VALUE, DOCUMENT, GRAPH, IN-MEMORY - SEARCH DATABASES

* NO SQL data Example: "JSON" (JavaScript Object Notation)

- Data can be Nested
- Fields can change over time
- Support for New Types: Arrays, etc--

AWS RDS →

AWS RDS - Relational Database Service

↓

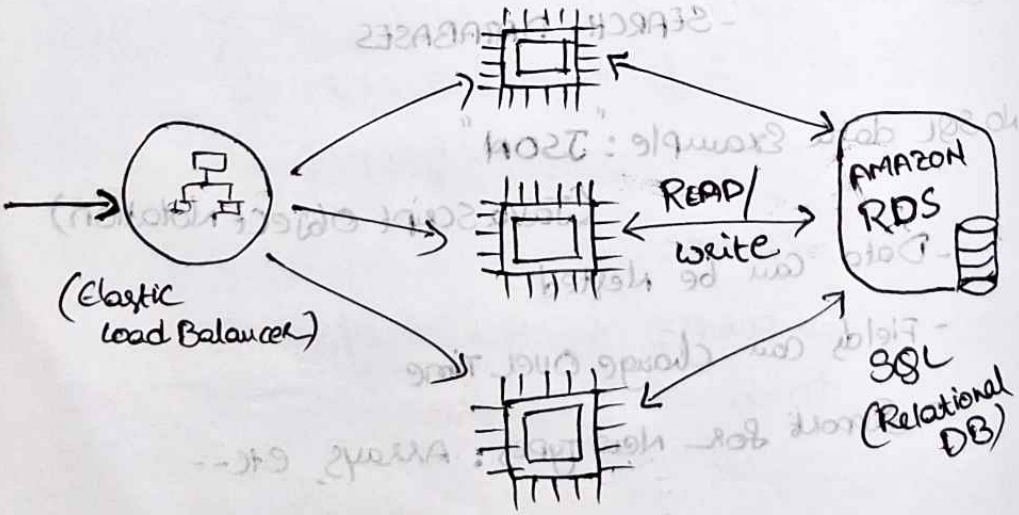
- Managed DB Service - DB use 'SQL'

RDS

- Allows you to create DBs in cloud that are Managed by AWS

  - Postgres
  - MySQL
  - MariaDB
  - Oracle
  - Microsoft SQL Server
  - Aurora (AWS Proprietary database)

RDS Solution Architecture:



AMAZON AURORA :

↳ Proprietary technology from AWS
(NOT OPEN SOURCED)

* PostgreSQL & MySQL are both supported as AURORA DB
↓
AWS Cloud optimised
5X, 3X ↑ Performance.

- Not in free tier
- More Cloud Native.
- AURORA costs more than RDS (20% more)
  - But it's more efficient.

RDS deployments

> Read Replicas:
- Scale the read workload of your DB
- Can create upto 5 Read Replicas
- Data is written only to main DB

> MultiAZ:
- Failover in case of AZ outage (High Availability)
- "Failover DB"
- Can only have '1' other AZ as failover
- Data is only read/written to main DB.

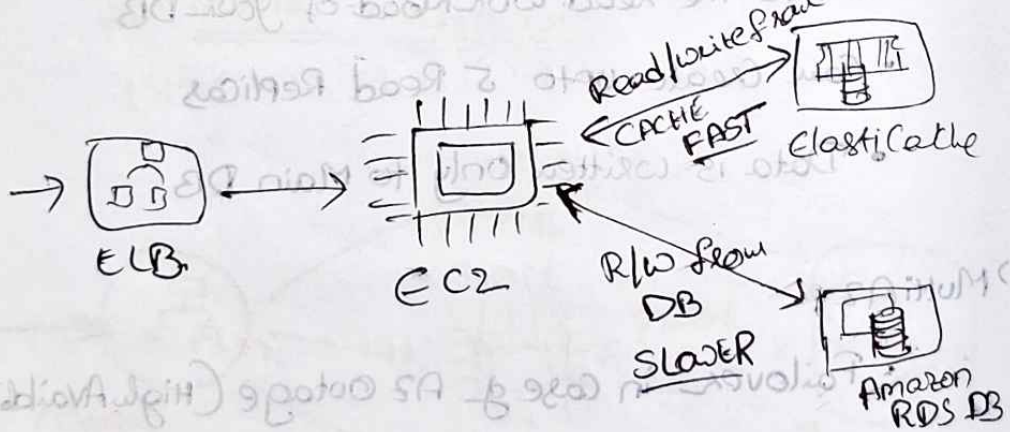
> MultiRegion: → Read Replica.
- ~~Failover~~ across Region.
DR

→ AMAZON ELASTICACHE

↓

is to get Redis / Memcached DB

* Caches are in-memory DB with high performance, low latency
* Helps reduce load off DBs for read intensive workloads.
* AWS take care of
  - OS Maintainance
  - Patching
  - Optimisations
  - Setup, Configuration, Monitoring, failure recovery
  - Backups.



DYNAMO DB: * Fully Managed - High Available with replication across AZ

* NoSQL DB - Not a relational DB.
* Scale to Massive workload - Distributed "Serverless" DB
* Fast / consistent / Very Low Latency retrival / Low cost Autoscaling

DynamoDB: Standard & Infrequent (IA) Tableclass

↓

Typed data → Key-Value DB

NOSQL

DynamoDB Accelerator - DAX

- Fully Managed in-memory cache

"Just for DynamoDB."

- Using DAX — 10x performance improvement
- Single digit millisecond latency to Microseconds latency when accessing DynamoDB
- Secure, highly Scalable & highly available

DynamoDB - Global Tables:

↓

- Make a DynamoDB accessble with low LAtency in Multiple regions.
- Active-Active replication (Read/write to any AWS Region)

Red Shift:

→ based on Postgre SQL (Not used for OLTP)

→ It's OLAP - Online Analytical Processing (Analytics & DWH)

Redshift Contd...

- Load data once every hr. Not every second
- 10x Better performance than other DWH
  Scale to PBs of data
- Columnar Storage of data (instead of row based)
- MPP - Massively Parallel Query Execution.
  - highly available
- Pay as you go - based on instances provided
- Has a SQL interface for performing queries
- BI Tools such as - AWS QuickSight, Tableau } Integrate with it

→ EMR - Elastic Map Reduce

↓

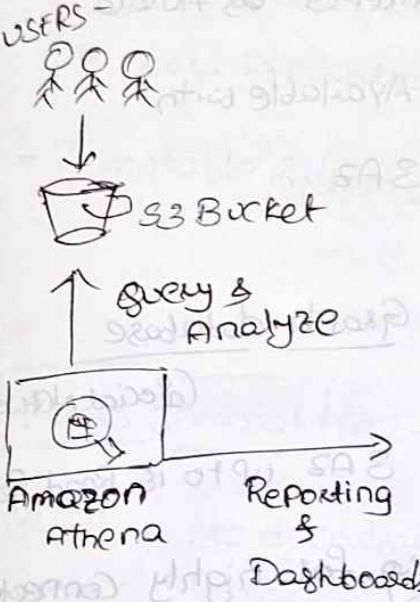
- helps Creating Hadoop Clusters (Big Data) to analyze and Process Vast Amount of data
- The clusters can be made of "hundreds of EC2 instances"

  Also Supports - Apache Spark, HBase, Presto, Flink...
- EMR takes care of Provisiong & Configuration.
- Data Processing, Machine Learning, web indexing, Big data

→ ATHENA:

↳ Serverless query service - To Perform Analytics S3 Objects.

* uses Standard SQL queries to query the files
* Supports CSV, JSON, ORC, AVRO, and Parquet (Presto)



* $5.00 Per TB of data Scanned
* use Compressed / columnar data for Cost-Savings (Less Scan.

Use Cases: Business Intelligence, Analytics Reporting, Analyze & Query VPC flowlogs ELB logs, Cloudtrails etc..

→ QUICK SIGHT:

↳ Serverless Machine learning-Powered "BI" Service to Create interactive Dashboards

- Fast, Automatically Scalable, Embeddable

UseCases: Business Analytics Business Visualization, Business Insights

Document DB: like Aurora for Postgres/MySQL

↓

- is for MongoDB (which is a NOSQL DB)

↓

- To store, query, and index JSON data

- Similar "deployments concepts" as Aurora

- Fully Managed, highly Available with Replication across 3 AZ

AMAZON Neptune: fully Managed Graph database

(Social Network)

- Highly available across 3 AZ upto 15 Read Replica

- Build & Run APPS working for highly connected datasets - optimised for these complex & hard queries.

- Milliseconds Latency query.

- Great for Knowledge graphs, Fraud detection, Recomandation Engines, Social Networking.

→ AMAZON QLDB → Quantum ledger Database.

↓

Book recording financial Transactions.

- Fully Managed, Serverless, High available Replication across 3 AZ
- used to Review history of all changes made to your Application data over Time
- Immutable System: - No Entry can be removed.
  @ Modified.
  Cryptographically Veriable

QLDB Journal @2-3x better performance than common ledger Blockchain frameworks

QLDB - No decentralised Component,
In accordance with financial regulations.

→ Amazon Managed Blockchain:

Blockchain → Makes possible to Build Apps where Multiple Parties can Execute Transactions - without need for a trusted, central Authority.

(Decentralised)

Amazon Managed Blockchain → Managed Service to
1) Join Public Blockchain N/ws
2) Create your own Private Scalable N/w

Compatible with
1) Hyperledger Fabric 2) Ethereum.

→ DMS - Database Migration Service

- Quickly & Securely Migrate DBs to AWS
  Resilient, Self healing
- The Source DB remains Available during Migration

Supports:

1) Homogenous Migration: ex Oracle to Oracle
2) Hetrogenous Migration: ex MySQL to Aurora

→ AWS GLUE:

↳ Managed ETL (Extract-Transform-Load) Service.

* Useful to Prepare & Transform for Analytics
* Fully Serverless Service.

Glue Data Catalog: Catalog of Datasets

→ DATABASES & ANALYTICS

SUMMARY

- Relational Databases - OLTP : RDS & Aurora (SQL)
- Diff. b/w Multi-AZ, Read Replicas, Multi-Region
- In-Memory DB : Elasticache
- Key-Value DB : DynamoDB (Serverless)
  DAX (cache for DynamoDB)
- warehouse - OLAP : Redshift (SQL)
- Hadoop Cluster : EMR
- Athena : Query data on Amazon S3 (serverless SQL)
- QuickSight : Dashboards on your Data (Serverless)
- DocumentDB : "Aurora for MongoDB" (JSON - NOSQL DB)
- AMazon QLDB : Financial Transactions ledger
- Amazon Managed Blockchain : Decentralised. ethereum
- Glue : Manged ETL & Data Catalog Service
- Neptune : Graph DB
- DMS : Database Migration Service.

ECS, FARGATE, ECR

→ DOCKER: → Software development platform to deploy APPS

* APPS are Packaged in "Containers" that can be run on any OS
* APPS run the same, regardless of where they're run
  - Any Machine
  - No Comatibility issues
  - Predictable behaviour
  - less work
  - Easier to maintain & deploy
  - Works with Any language, Any OS, Any Technology with Docker
* Scale Containers UP & down very quickly

where Docker images are Stored?

- Docker images are Stored in Docker Repositories.

PUBLIC: DOCKER HUB (we can find many base images for many technologies los)

- Ubuntu, MySQL, NodeJS, Java.

PRIVATE: AMAZON ECR (Elastic Container Registry)

DOCKER Vs VMS.

→ ECS = Elastic Container Service

↳ To launch Docker Containers on AWS

↳ you must Provision & Maintain the infrastructure (the EC2 instances)

↳ AWS takes care of Starting/Stopping Containers

↳ Has integrations with Application Load Balancer

→ Fargate: → Launch Docker Containers on AWS but you donot Provision the infrastructure (NO EC2 instances to manage) - Simpler

↳ Serverless Offering.

↳ AWS Just runs Containers for you based on CPU/RAM as you need.

→ ECR = Elastic Container Registry

↳ Private docker Registry on AWS.

↳ To Store docker images so they can be run by ECS/Fargate.

→ Serverless: New Paradigm - developers don't Manage Servers anymore.

- They Just deploy code, functions.

Initially Serverless = FAAS (Function-As-A-Service)

'AWS LAMBDA'

→ AWS LAMBDA:

↓

- Virtual functions - no Servers to Manage
- Limited by Time - Short Executions
- Run On-demand
- Scaling is Automated

Benifits:

- EASY PRICING:
  - Pay Per request & Compute time
  - Free tier of 1M AWS Lambda requests
    400,000 GBs of Compute time
- Integrated with Whole AWS Suite of Services
- Event-Driven: functions get invoked by AWS when Needed. (Reactive)
- Easy Monitoring through AWS Cloud watch
- Easy to get more resources Per functions (UP to 10GB of RAM)
  Increasing RAM will also improve CPU / Network

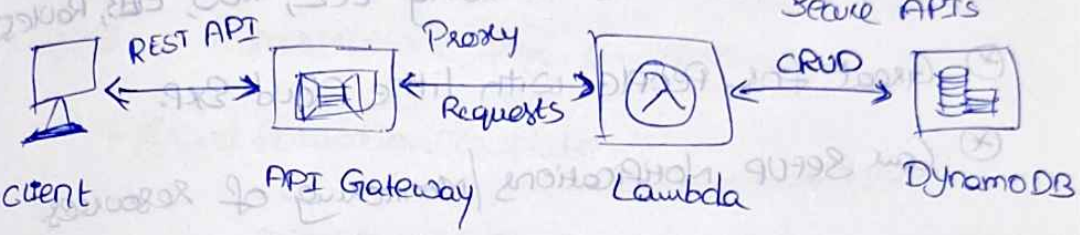
Language Support: NodeJs, Python, Java8, C#, Golang, C#/Powershell, Ruby, Custom Runtime API

- Lambda Container Image. (Must implement Lambda Runtime API)

→ AMAZON API GATEWAY:

↓

- Building Serverless HTTP API
- Fully Managed Service for developers to Easily Maintain, Created Publish, Monitor & Secure APIs



- Serverless & Scalable
- Supports RESTful APIs & WebSocket APIs
- Support for Security, User Authentication, API Throttling, API Keys, Monitoring...

→ AWS BATCH: Fully Managed Batch Processing at any Scale

* Batch Job is a Job that has Start & an End.
* Batch will dynamically launch EC2 / Spot instances.
  ↓
  will Provision the right amt of Compute / Memory.

- Batch Jobs are defined as Docker images and run on ECS.

→ AMAZON LightSail:

(Staud Alone)

- Virtual Servers, Storage, Databases & N/w
- Low and Predictable Pricing
- Simple Alternative to using EC2, RDS, EBS, Routes
- Great for People with little Cloud Exp. →
- Can SetUP Notifications / Monitoring of Resources

DEPLOYMENTS & MANAGING INFRASTRUCTURE AT SCALE

→ Cloud FORMATION : (IAAS)

- declarative way of outlining your AWS infrastructure, for any resource

- Cloud formation Template
  ↓
  AWS CF will create those for you in the right order with exact config. you specify.

∴ "Infrastructure as CODE"

- COST - Optimised, Saving Strategy
- Productivity
- Supports almost all AWS Resources

⊛ CLOUD FORMATION Stack Designer

→ AWS - CDK : Cloud Development kit

↓

⊛ define ur cloud infrastructure using a familiar language (Java, JS, Python ---)

- The code is "compiled" into Cloud formation Template.

→ BEANSTALK:

AWS Elastic Beanstalk - Managed Service

↓

* developer centric view of deploying an application on AWS
* It uses all the components we've seen before: EC2, ASG, ELB, etc--

Beanstalk = PAAS

* using Beanstalk is free, But you pay for underlying instances.

Three Architecture Models

1) Single Instance deployment: good for dev
2) LB + ASG : Great for Pre-Prod/Prod web APPS
3) ASG Only : Great for non-web APPS

* Support for Many Platforms

- GO, Java SE, Java with Tomcat,
- Net on win., NodeJS, PHP, Python, Ruby
- Packer Builder - - -

Elastic Beanstalk - Health Monitoring.

→ AWS Code Deploy: deploy your Application Automatically.

Hybrid Service (on Premise, EC2 instances)

→ AWS Code Commit: 'Fully Managed Code Repository'

* Before Pushing the application code to Servers, it needs to be stored somewhere

Store Code in Repository using Git Tech.

Public Offering — AWS
"Git Hub" Code Commit

→ AWS Code Build: Code building Service in Cloud

↓

* compiles source code, run tests and Produces Package to be deployed.

- Fully Managed, Serverless
- Continously Scalable & Highly Available
- Secure.
- Pay as you go Pricing.

→ AWS Code Pipeline

Basis for "Orchestration" (CICD)

Code ⇒ Build ⇒ Test ⇒ Provision ⇒ Deploy.

- Fully Manged Service.

→ AWS CodeArtifact:

↓

* Code dependencies - Artifact Management
* Secure, Scalable & Cost-effective for S/w development
- Works with common dependency Management tools such as Maven, Gradle, npm, Yarn, twine, Pip, NuGet.

→ AWS CodeStar:

* 'Unified UI' to Easily Manage S/w dev activities in one place
* Can Edit the code in cloud using "Cloud 9"

→ AWS Cloud 9: → Cloud IDE
- for writing, running, & debugging Code
'Web Browser'
* Allows for Code-Collaboration / Pair Programming

AWS Systems Manager: (SSM)

↓

Ⓐ Helps to manage your EC2 / on-premise systems at scale

Ⓧ Hybrid AWS service.

Ⓐ Get operational insights abt infrastructure.

Imp features are:

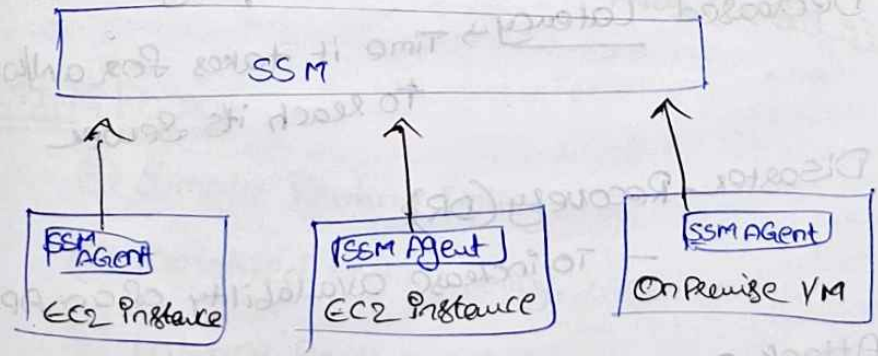
- Patching Automation for enhanced compilance
- Run cmds across entire fleet of servers
- Store Parameter Config with SSM Parameter store.

works for Both windows, linux servers.

How SSM works:

1) we need to install SSM agent on to systems we control

2) installed by default on Amazon Linux AMI / some Ubuntu AMI



SSM - Session Manager — Allows to start a secure shell on your EC2 / on prem servers

- No SSH, bastion hosts / SSH keys are needed.
- Supports linux, macOS, windows.

→ AWS OPS WORKS:

* Chef & Puppet helps you Perform Server Config automatically @ Repetitive Actions
* They work great with EC2 & On-Prem VM
* AWS OPS WORK = Managed chef & PUPPET
  Alternative to AWS SSM
* Only Provision Standard AWS Resources.

LEVERAGING THE AWS GLOBAL INFRASTRUCTURE

→ why making a Global Application
↓
is an Application deployed in
"Multiple Geographies"
On AWS → could be "Regions" / "Edge Locations"

* Decreased Latency → Time it takes for a N/w PKT to reach its Server
* Disaster Recovery (DR):
  — To increase availability of an Application
* Attack Protection:
  - Distributed global infrastructure is harder to attack.

→ Global AWS Infrastructure:

Regions, Availability Zones, Edge Locations (Points of Presence)

Global Applications in AWS:

→ AWS ROUTE 53:

↓

Managed DNS (Domain Name System)

↓

⊛ Collection of rules & records which helps client understand how to reach a server through URLs

In AWS, the most common records are:

- www.google.com ⇒ 12.34.56.78 == A record (IPV4)
- www.google.com ⇒ 2001:0db8:8563:000:000:7334 == AAAA (IPV6)
- search.google.com ⇒ www.google.com == CNAME: hostname to hostname
- example.com ⇒ AWS Resource == Alias (ex: ELB, S3, RDS etc.)

ROUTING POLICY:

⊛ Simple Routing Policy.

⊛ Weighted Routing Policy.

⊛ Latency Routing Policy.

⊛ Failover Routing Policy. (DR)

→ AWS CLOUD FRONT: → CDN (Content Delivery Network)

- ⊛ improves read performance, content is cached at edge
- ⊛ improves users experience.
- ~ 216 edge locations globally.
- ⊛ DDOS Protection, integration with Shied AWS WAF & Shield.

→ S3 Transfer Accelerator:

- ⊛ increase transfer speed by transfering file to an AWS edge location which will fwd the data to S3 bucket in target location

→ AWS Global Accelerator:

- ⊛ Improve global application Availability & Performance
  - → 2 Anycast IP are created for your application.
  - → Good for HTTP use cases

→ AWS OUTPOSTS: (Fully Managed)

Hybrid Cloud — Businesses that keep on premises infrastructure alongside a cloud infrastructure.

"Server Racks" that offers the same AWS infrastructure Services

⊛ AWS will setup and Manage "outpost Racks"

→ AWS wave lengths : 5G N/w

* wave length zones are infrastructure deployments embeded within the telecommunications Providers Datacenters at the Edge of 5G N/w.

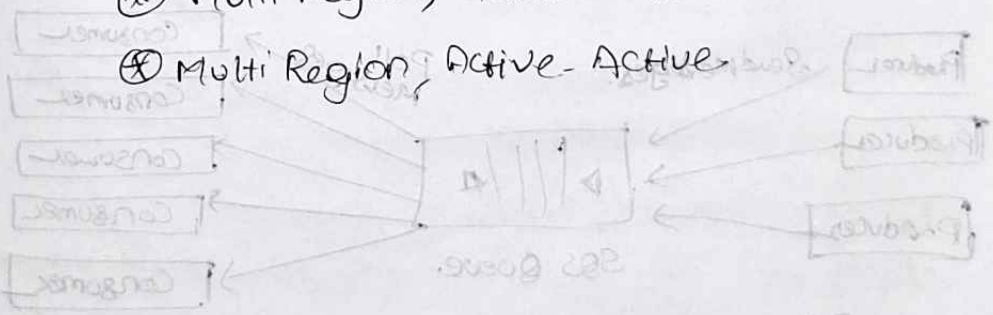
→ AWS Local Zones:

↓ Place AWS Compute, Storage, DB & other selected AWS Services closer to end users to Run latency-sensitive APPs.

↓ Availability Zones!
VPC - Sub Nets.

→ Global Applications Architecture.

* Single Region, Single AZ
* Single Region, Multi AZ
* Multi Region, Active-Passive.
* Multi Region, Active-Active.



CLOUD INTEGRATIONS

* Applications need to communicate with one another

↓

2 Patterns of application communication.

1) Synchronous communications
2) Asynch / event based. Communications.

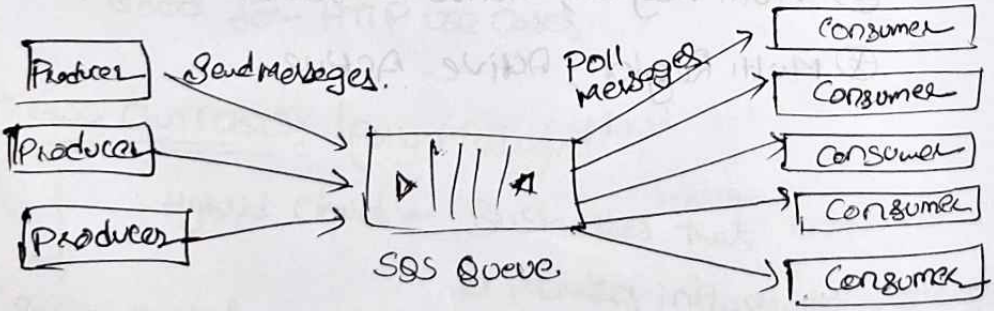
* When there is sudden spike in traffic.

Synchronous Communications can be problematic

In that case you better decouple your Apps.

- using : SQS : Queue Model
- using : SNS : Pub/Sub Model.
- using : Kinesis

→ Amazon SQS: Simple Queue Service.

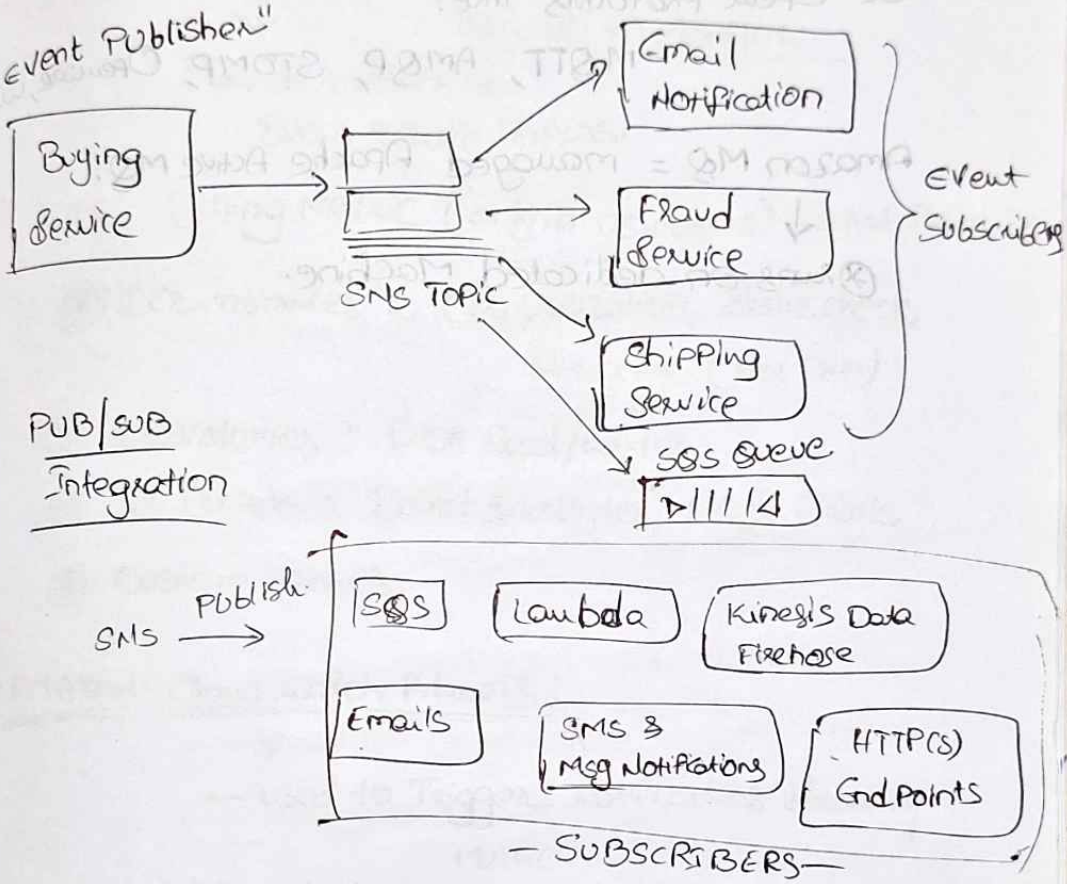


- Fully Managed (Serverless) use to decouple Applicat
- Scales from 1 Msg/sec to 10,000s/second.
- No limit on msg in queue

→ AMAZON SNS: Simple Notification Service.

↓

Send one Message to Many Recievers



→ KINESIS:

Real Time Big-data Streaming.

Managed Service → TO collect Process & Analyze Real time Streaming data At any Scale.

① Kinesis Data Streams:

↓

low latency Streaming to ingest Data.

② Kinesis Data Firehouse: → load Streams into S3, Red shift

③ Kinesis Data Analytics. ④ Kinesis Video Streams.

→ Amazon MQ:

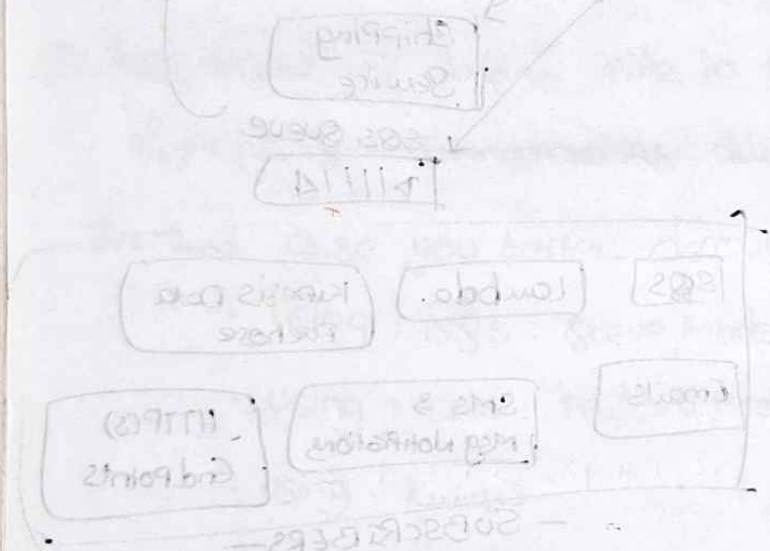
- Traditional Applications running On Prem may use Open Protocols like:

  MQTT, AMQP, STOMP, Openwire, WSS

Amazon MQ = managed Apache Active MQ.

↓

- runs on dedicated Machine.



CLOUD MONITORING

→ AMAZON CLOUD WATCH METRICS:

Variable to Monitor

Provide Metrics to Every Service in AWS.

ex:- Billing Metric (only in us-east-1): Total Estimate

* EC2-instances : CPU Utilization, Status check, Network. (Not RAM)
* EBS volumes : Disk Read/writes
* S3 Buckets : Bucket size Bytes, No of Objects
* Custom Metrics.

→ AMAZON Cloud watch Alarms :

- used to Trigger Notifications for any Metric

- Alarms Actions
  * ASG: Increase / Decrease EC2 instances "desired" count
  * EC2 Actions: Stop / terminate / Reboot / Recover an EC2 instance
  * SNS Notifications: Send a notification into an SNS TOPIC.
  * Billig Alarm

→ Alarm States: OK, INSUFFICIENT_DATA, ALARM

→ AMAZON CLOUD WATCH LOGS:

↓

Can collect logs from:

- ✳ Elastic Beanstalk
- ✳ ECS
- ✳ AWS Lambda
- ✳ Cloud Trails
- ✳ Cloud watch log agents on EC2 machines / on-Prem Servers.
- ✳ Route 53: log DNS queries.

— Enables Real-time Monitoring logs.

→ AMAZON CLOUD WATCH EVENTS:

✳ Schedule : CRON JOBS

✳ EVENT PATTERN: Event Rules to react to a Service doing Something

✳

↳ AMAZON EVENT BRIDGE:

1. Default Event Bus
2. Partner Event Bus
3. Custom Event Busses.

→ Schema Registry. (Model event schema)

→ AWS Cloud Trail → Provides Governance, Compliance & Audit for AWS Account

↓

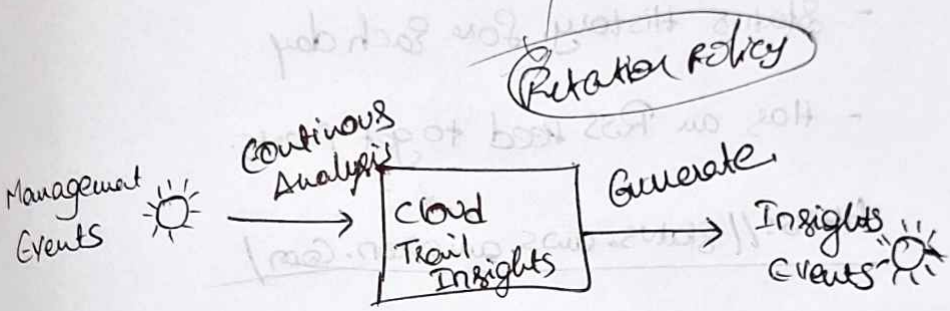
Enabled by default

✳ Used to Inspect and Audit

cloud trail Events → ① Management Events - READ - WRITE

② DATA Events

③ Cloud Trail Insights Events



→ AWS X-RAY:

- Debugging in Production
- Visual Analysis of our Applications
- Troubleshooting Performance
- Understand dependencies in Microservice Architecture.
- Pinpoint Service issues : Tracing.
- Find Errors & Exceptions.
- where i am throttled
- Are we meeting SLA?
- Identify users that are impacted.

→ AMAZON

CODEGURU → ML powered Service

↳ Automated Code Reviews

↳ Application Performance recomendations.

① Codeguru Reviewer

② Codeguru Profiler

→ AWS Status: Service Health Dashboard

↓

- Shows all regions, all services health.
- Status History for each day
- Has an RSS feed to get Alerts.

https://status.aws.amazon.com/

→ Personal Health Dashboard

↓ Personalised Alerts / remediation guidance

Alerts / Remediation / Proactive / Scheduled Alerts.

VPC & NETWORKING

→ VPC: Virtual Private Cloud

- VPC, Subnets, Internet Gateways & NAT Gateways
- Security Groups, Network ACL, VPC Flow logs
- VPC Peering, VPC Endpoints
- Site to Site VPN & Direct Connect
- Transit Gateway

→ VPC: Virtual Private Cloud

↘ Private Network to deploy your Resources
(Regional Resources)