

## AWS Cloud Certified Practitioners (CCP) Note

### IAM Identity and Access Management

- IAM = Identity and Access Management, it is a Global Service
  - ✓ Root account created by default, root account should not be shared and used
  - ✓ Users are people within your organization, and can be grouped
  - ✓ Groups only contain users, not other groups
  - ✓ Users don't have to belong to a group, and user can belong to multiple groups
  - ✓ In AWS a user can belong from without from groups so those users we can assign the inline policy
- **IAM: Permissions:** -
  - ❖ **Users or Groups** – Can be assigned JSON document called policies
  - ❖ These policies define the permission of the users
  - ❖ In AWS you apply the least privilege principle: don't give more permissions than a user needs

#### ➤ IAM Policies Structure: -

- Consists of
  - **Version:** policy language version, always include "2012-10-17"
  - **Id:** an identifier for the policy (optional)
  - **Statement:** one or more individual statements (required)
- Statements consists of
  - **Sid:** an identifier for the statement (optional)
  - **Effect:** whether the statement allows or denies access (Allow, Deny)
  - **Principal:** account/user/role to which this policy applied to
  - **Action:** list of actions this policy allows or denies
  - **Resource:** list of resources to which the actions applied to
  - **Condition:** conditions for when this policy is in effect (optional)

```
{
  "Version": "2012-10-17",
  "Id": "S3-Account-Permissions",
  "Statement": [
    {
      "Sid": "1",
      "Effect": "Allow",
      "Principal": {
        "AWS": ["arn:aws:iam::123456789012:root"]
      },
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": ["arn:aws:s3:::mybucket/*"]
    }
  ]
}
```

- **IAM – Password Policy:** - Password policy really helpful against brute force attacks on your accounts.
  - ✓ Strong password = higher security for your account
  - ✓ In AWS, you can setup a password policy:
    - Set a minimum password length
    - Require specific character types:
      - Including upper letters
      - Lowercase letters
      - Numbers
      - No-alphanumeric characters
    - Allow all IAM users to change their own passwords
    - Require users to change their password after some time (password expiration)
    - Prevent password re-use
- **MFA (Multi Factor Authentication):** - MFA is using the combination of a password that you know and a security device that you own and these two things together have a much greater security than just a password.
  - ❖ **MFA Device options in AWS:** -
    - ✓ **Virtual MFA Device:** - Like Google/Microsoft/Authy Authenticator apps for authenticate your account
    - ✓ **Universal 2<sup>nd</sup> Factor (U2F) security key:** - That is a physical device such as Yubikey by Yubico is a 3<sup>rd</sup> party to AWS this is not a AWS provided and we used a physical device to authenticate your account
    - ✓ **Hardware Key Fob MFA Device:** - Such as "Gemalto" is provided the hardware key for authenticate your account which is also 3<sup>rd</sup> party to AWS
    - ✓ **Hardware key Fob MFA Device for AWS GovCloud (US):** - In the US if you used AWS Gov cloud then you have a special key Fob this is provided by SurePassID which is also a 3<sup>rd</sup> party

- **AWS Access:** - There are 3 different way to access AWS.
  - 1) AWS Management console (Protected by password + MFA)
  - 2) AWS Command Line interface (CLI): protected by access keys
  - 3) AWS Software Developer kit (SDK) – for code: protected by access keys
  - ❖ Access keys are generated through the AWS Console
  - ❖ Users Manage their own access keys
  - ❖ Access keys are secret, just like a password, Don't share them because they can generate their own access keys as well so really make sure that you treat your access key ID just like user name and secret access key just like your password you do not share them with other people
- **AWS CLI:** - A tool that enable you to interact with AWS services using command in your command-line shell. Direct access to the public APIs of AWS services. You can develop scripts to manage your resources. It's open-source <https://github.com/aws/aws-cli> Also alternative to using AWS management console
- **SDK (Software development kit):** - Support multiple languages [JavaScript, Python, PHP, .net, Ruby, Java, Go, Node.js, C++] specific APIs call (set of libraries). Enable you to access and manage AWS services programmatically. Embedded within your application. Mobile SDKs (Android & iOS,...). IoT device SDKs (Embedded C, Arduino,...). Example – AWS CLI is build on AWS SDK for python.
- **Currently, AWS CloudShell is available in the following AWS Regions:** -
  - ❖ US East (Ohio)
  - ❖ US East (N. Virginia)
  - ❖ US West (Oregon)
  - ❖ Asia Pacific (Mumbai)
  - ❖ Asia Pacific (Sydney)
  - ❖ Asia Pacific (Tokyo)
  - ❖ Europe (Frankfurt)
  - ❖ Europe (Ireland)
- **IAM Roles for services:** - Some AWS service will need to perform actions on your behalf, on our account.
  - ❖ To do so, we will assign permission to AWS services with IAM Roles we are going to create what's called an IAM role. So these IAM role just like a user, but they are intended to be used not by physical people but instead they will be used by AWS services.
  - ❖ **Common Roles:** -
    - EC2 Instances Roles
    - Lambda Function Roles
    - Roles of CloudFormation
- **IAM Security Tools:** - There are 2 types security tools we can implement
  - 1) **IAM Credentials Report (account level):** - A report that lists all your account's users and the status of their various credentials
  - 2) **IAM Access Advisor (user-level):** - Access advisor shows the service permission granted to a user and when those service were last created. You can 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 groups and assign permissions to groups
  - ❖ Create a strong password policy
  - ❖ Use and enforce the use of Multi factor Authentication (MFA)
  - ❖ Create and use Roles for giving permissions to AWS services
  - ❖ Use access keys for programming access (CLI /CDK)
  - ❖ Audit permissions of your account with the IAM Credentials Report
  - ❖ **Never ever share IAM user and access keys**
- **Shared Responsibility Model for IAM:** -
  - ❖ **AWS:** -
    - ✓ Infrastructure (global network security)
    - ✓ Configuration and vulnerabilities analysis

- ✓ Compliance validation
- ❖ **User: -**
  - ✓ Users, Group, Roles, Policies management and monitoring
  - ✓ Enable MFA on all accounts
  - ✓ Rotate all your keys often
  - ✓ Use IAM tools to apply appropriate permissions
  - ✓ Analyze access patterns & review permissions
- ❖ Question: - Which permission you should apply regarding IAM Permission? Ans: - Grant least privilege

### EC2 (Elastic Compute Cloud)

- **EC2: -** EC2 is one of most popular of AWS offering. It is definitely used everywhere. It is IaaS services.
  - ❖ It mainly consists in the capability of: -
    - ✓ Renting Virtual Machines (EC2)
    - ✓ Storing Data on virtual drive (EBS) you can store data on virtual drives or EBS volumes
    - ✓ Distribute load across machines (ELB)
    - ✓ Can scale services using an auto-scaling group or ASG
- **EC2 sizing & configuration: -** We can choose Linux, Windows or Mac OS while configuring VMs.
  - ✓ How much compute power, RAM
  - ✓ Storage space: - Do want storage that is going to be attached through the network and we'll see about it with (EBS & EFS), and do want to be hardware attached in this case it will be an EC2 instance store
  - ✓ Network card: - Speed of the card, Public IP address (Want to attach to your EC2 instance do want a network card that is going to be fast)
  - ✓ Firewall rules: Security Group
  - ✓ Bootstrap script (configure at first launch): - EC2 User Data
- **EC2 User Data: -** It is possible to bootstrap our instances using an **EC2 User data** script
  - ✓ **bootstrapping** means launching commands when a machine starts
  - ✓ That script is **only run once** at the instances **first start** and then will never be run again
  - ✓ EC2 user data is used to automate boot tasks such as: (hence the name bootstrapping)
    - **Installing Updates**
    - **Installing Software**
    - **Downloading common files from the internet**
    - **Anything you can think of**
  - ✓ Remember if you add more user data script the more your instance has to do at boot time. The EC2 user data script runs with the root user.
- **EC2 instances types: - example**

Instance	vCPU	Mem (GiB)	Storage	Network Performance	EBS Bandwidth (Mbps)
t2.micro	1	1	EBS-Only	Low to Moderate	
t2.xlarge	4	16	EBS-Only	Moderate	
c5d.4xlarge	16	32	1 x 400 NVMe SSD	Up to 10 Gbps	4,750
r5.16xlarge	64	512	EBS Only	20 Gbps	13,600
m5.8xlarge	32	128	EBS Only	10 Gbps	6,800

- **EC2 Instances types: -** AWS has 7 types of EC2 instances such as: - <https://aws.amazon.com/ec2/instance-types/>
  - 1) General Purpose
  - 2) Compute Optimized
  - 3) Memory Optimized
  - 4) Accelerated Computing
  - 5) Storage Optimized

6) Instance features

7) Measuring Instance performance

❖ AWS has the following naming convention: - m5.2xlarge

M → instances class

5 → generation (AWS improves them over time)

2xlarge → size within the instance class

❖ **General Purpose:** - Great for a diversity of workloads such as web servers or code repository.

✓ Balance Between: - Compute, Memory, Networking

❖ **Compute Optimized:** - Great for compute-intensive tasks that require high performance processors

✓ Batch Processing Workloads

✓ Media Transcoding

✓ High Performance web servers

✓ High performance computing (HPC)

✓ Scientific modeling & machine learning

✓ Dedicated gaming servers

❖ **Memory Optimized:** - Fast performance for workloads that process large data sets in memory

**Use Cases:** -

✓ High performance, relational/non-relational databases

✓ Distributed web scale cache stores

✓ In-memory databases optimized for BI (business intelligence)

✓ Application performing real-time processing of big unstructured data

❖ **Storage Optimized:** - Great for storage-intensive tasks that require high, sequential read and write access to large data sets in local storage

**Use Cases:** -

✓ High frequency online transaction processing (OLTP) systems

✓ Relational & NoSQL database

✓ Cache for in-memory database (for example Redis)

✓ Data Warehousing applications

✓ Distributed file systems

➤ **Security Groups:** - Security groups are the fundamental of network security in AWS. They control how traffic is allowed into or out of our EC2 instances. Security groups only contains allow rules so we can say what is allowed to go in and go out.

✓ Security groups rules can references by IP or by security group

❖ **Security Groups Deeper Dive:** - Security groups are acting as a “firewall” on EC2 instances.

**They regulate:** -

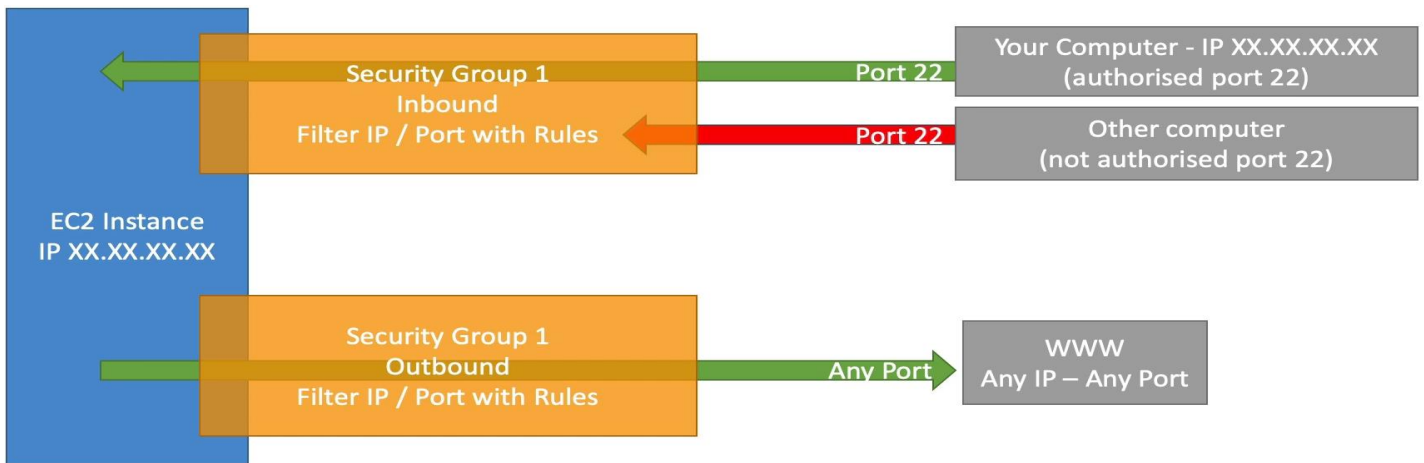
○ Access to ports

○ Authorised IP ranges – Ipv4 and Ipv6

○ Control of inbound network (from other to the instances)

○ Control of outbound network (from the instances to other)

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ
HTTP	TCP	80	0.0.0.0/0	test http page
SSH	TCP	22	122.149.196.85/32	
Custom TCP Rule	TCP	4567	0.0.0.0/0	java app



- **Security group good to know:** - Can be attached to multiple instances and security group can be assigned to multiple EC2 instances or 1 EC2 instances have can multiple security group too.
  - ✓ Security group is locked down to a region / VPC combination so if you switch in other region you have to create a new security group
  - ✓ does live “outside” the EC2 - if traffic is blocked to the EC2 instances won't to see it
  - ✓ It's good to maintain one separate security group for SSH access
  - ✓ If your application is not accessible (time out), then it's a security group issue
  - ✓ If your application gives a “connection refused” error, then it's an application error or it's not launched
  - ✓ By default all traffic inbound is blocked
  - ✓ All outbound traffic is authorised by default
- **SSH:** - SSH is one of more important function. It allows you to control a remote machine, all using the command line interface.

### SSH Troubleshooting: - 1) There's a connection timeout

This is a security group issue. Any timeout (not just for SSH) is related to security groups or a firewall. Ensure your security group looks like this and correctly assigned to your EC2 instance.

Security Group: sg-0781d3c194f33d751

Type	Protocol	Port Range	Source
SSH	TCP	22	0.0.0.0/0
SSH	TCP	22	::/0

### 2) There's still a connection timeout issue

If your security group is properly configured as above, and you still have connection timeout issues, then that means a corporate firewall or a personal firewall is blocking the connection.

### 3) SSH does not work on Windows

- If it says: `ssh command not found`, that means you have to use Putty

### 4) There's a connection refused

This means the instance is reachable, but no SSH utility is running on the instance

- Try to restart the instance
- If it doesn't work, terminate the instance and create a new one. Make sure you're using **Amazon Linux 2**

## 5) **Permission denied (publickey,gssapi-keyex,gssapi-with-mic)**

This means either two things:

- You are using the wrong security key or not using a security key. Please look at your EC2 instance configuration to make sure you have assigned the correct key to it.
- You are using the wrong user. Make sure you have started an **Amazon Linux 2 EC2 instance**, and make sure you're using the user **ec2-user**. This is something you specify when doing **ec2-user@<public-ip>** (ex: **ec2-user@35.180.242.162**) in your SSH command or your Putty configuration

## 7) **I was able to connect yesterday, but today I can't**

This is probably because you have stopped your EC2 instance and then started it again today. **When you do so, the public IP of your EC2 instance will change.** Therefore, in your command, or Putty configuration, please make sure to edit and save the new public IP.

- **EC2 Instances Connect:** - This is allow us connect to browser based instances. Without using any SSH, putty and pem key.
- **EC2 Instances Purchasing options:** -
  - ❖ **On-Demand Instances:** - Short workload, predictable pricing, pay by second.
  - ❖ **Reserved (1 or 3 years)**
    - ✓ **Reserved Instances** – Long workloads
    - ✓ **Convertible Reserved Instances** – Long workloads with flexible instances
  - ❖ **Saving plans (1 & 3 years)** - commitment to an amount of usages, long workload
  - ❖ **Spot instances** - short workload, cheap, can lose instances (less reliable)
  - ❖ **Dedicated hosts** - book an inter physical server, control instances placement
  - ❖ **Dedicated instances** - no other customer will share your hardware
  - ❖ **Capacity Reservation** - Reserve capacity in a specific availability zone for any duration
- **EC2 on Demand:** -
  - ❖ **Pay for what you use:**
    - Linux or windows: billing per second after first minute
    - All other operating system - billing per hour
  - ❖ Has the highest cost but no upfront payment
  - ❖ No long term commitment
  - ❖ Recommender for short-term and un-interrupted workloads, where you cannot predict how the application will behave
- **EC2 reserve instances:** - Up to 72% discount compared to on-demand
  - ❖ You reserve a specific instant attribute (**Instance Type, Region, Tenancy, OS**)
  - ❖ **Reservation Period – 1 year (+ discount) or 3 years (+++ discount)**
  - ❖ **Pyment options – No Upfront (+), Partial Upfront (++), All Upfront (+++)**
  - ❖ **Reserved Instances Scope – Regional or Zonal** (reserve capacity in an AZ)
  - ❖ Recommended for steady-state usage applications (think database)
  - ❖ You can buy and sell in the Reserved instances in marketplace
  - ❖ **Convertible Reserved Instances**
    - Can change the EC2 instance type, instance famaliy, OS, Scope and tenancy
    - Up to 66% discount
- **EC2 savings plans:** - Get a discount based on long-term usage (up to 72% - same as RIs)
  - ❖ Commit to certain type of usages (\$10/hour for 1 or 3 years)
  - ❖ Usage beyod EC2 savings plan is billed at the On-Demand price



- ❖ Locked to a specific instances family & AWS region (e.g., M5 in us-east-1)
- ❖ Flexible Across:
  - Instance Size (e.g., m5.xlarge, m5.2xlarge)
  - OS (e.g., Linux, Windows)
  - Tenancy (Host, Dedicated, Default)
- **EC2 spot instances:** - can a discount of up to 90% compared to on demand
  - ❖ Instances is that you can “lose” at any point of time if your Max price is less then the current is spot price
  - ❖ The most cost effective instances in AWS
  - ❖ Useful for workload that are resilient to failure
    - Batch Jobs
    - Data Analysis
    - Image processing
    - Any distributed workloads
    - Workloads with a flexible start and end time
  - ❖ Not Suitable for critical jobs or databases
- **EC2 Dedicated Hosts:** - A physical server with EC2 instances capacity fully dedicated to your use
  - ❖ Allows you address compliance requirements and use your existing server bound software licences (pre -socket, pre ---VM software licences)
  - ❖ **Purchasing Option:** -
    - **On-Demand** – pay per second for active dedicated host
    - **Reserved** – 1 or 3 years (No Upfront, Parial Upfront, All Upfront)
  - ❖ The most expensive option
  - ❖ Useful for software that have complicated licensing model (BYOD)
  - ❖ Or for companies that have strong regulatory or compliance needs
- **EC2 Dedicated instances:** - Instances run on hardware that’s dedicated to you.
  - ❖ May share hardware with other instances in same account
  - ❖ No control over instance placement (can move hardware after stop / start)
- **EC2 capacity reservations:** - Reserve On-Demand instances capacity in a specific AZ for any duration
  - ❖ You always have access to EC2 capacity when you need it
  - ❖ No Time commitment (created/cancel anytime), /no billing discounts
  - ❖ Combine with Regional instances and saving plans to benefit from billing discounts
  - ❖ You’re charged at On-Demand rate whether you run instances or not
  - ❖ Suitable for short-term, uninterrupted workloads that needs to be in a specific AZ

Below are the example for purchasing/billing for AWS instances.

## Which purchasing option is right for me?



- **On demand:** coming and staying in resort whenever we like, we pay the full price
- **Reserved:** like planning ahead and if we plan to stay for a long time, we may get a good discount.
- **Savings Plans:** pay a certain amount per hour for certain period and stay in any room type (e.g., King, Suite, Sea View, ...)
- **Spot instances:** the hotel allows people to bid for the empty rooms and the highest bidder keeps the rooms. You can get kicked out at any time
- **Dedicated Hosts:** We book an entire building of the resort
- **Capacity Reservations:** you book a room for a period with full price even you don't stay in it

➤ **Shared Responsibility model for EC2: - AWS: -**

- ✓ Infrastructure (global network security)
- ✓ Isolation on physical hosts
- ✓ Replacing faulty hardware
- ✓ Compliance validation

**User: -**

- ✓ Security group rules
- ✓ Operating system patches and update
- ✓ Software and utility installed on the EC2 instances
- ✓ IAM roles assigned to EC2 & IAM user and access management
- ✓ data security on your instance

**EC2 Instances Storage**

- **EBS Volume:** - It is a network drive you can attach to your instances while they run. It allows your instances persist data even after their termination so that can help purpose we can recreate instances and mount to the same EBS volume for before and will get back same data.
- ❖ One EBS can be only mounted to one EC2., When create EBS volume you can bound to a specific availability zone.
  - ❖ Free tier: 30GB of free EBS storage of type General Purpose (SSD) or Magnetic per month.
  - ❖ It's locked to an Availability Zone
    - An EBS volume in us-east-1a cannot be attached to us-east-1b
    - To move a volume across different availability zone, you first need to snapshot it
  - ❖ **EBS – Delete on Termination attribute**
    - ✓ Controls the EBS behaviour when an EC2 instances terminates
      - By Default, the root EBS volume is deleted (attribute enabled)
      - By Default, any other attached EBS volume is not deleted (attribute disabled)
- **EC2 Instance Store:** - EC2 Instance Store has a better I/O performance, but data is lost if: the EC2 instance is stopped or terminated, or when the underlying disk drive fails.
- **EBS Snapshots:** - You can take your EBS volumes and make a snapshot which is also called backup at any point of time you wanted to. Even when you terminated the EBS volume you can restore from snapshot. Not necessary to detach volume to do snapshot but recommended to do first detach the volume from instances.
- ❖ Also can copy the snapshot across availability zones and regions and the idea is that you would be able to transfer some of your data in a different region on AWS to leverage the global infrastructure.
  - ❖ **EBS Snapshots features:** -
    - EBS snapshot archive – so it allow you to move your snapshot to another storage tier called an archive tier that is 75% cheaper.
    - If you have in the archive it takes you between 24 to 72 hours to restore from the archive.
  - ❖ **Recycle Bin for EBS Snapshots:** -
    - So by default, when you delete snapshots, they're gone. But you can setup recycle bin and the recycle bin will have all the snapshots that are deleted then after a while, maybe you can specify from one day to one year the snapshots are gone from the bin (like retention).
- **AMI Overview:** - (AMI – Amazon Machine Image) and they represent a customization of an EC2 instances you can customize your own and what is in an IAM – well we have our own software configuration, we can define and set up the operating system, monitoring. If we create our own AMI we're going to get a faster boot time and configuration time because all the software that we want to install onto EC2 instance is going to pre-packaged through the AMI.
- ❖ AMI are build for a specific region and can be copied across regions if we wanted to use it and leverage the AWS global infrastructure.
  - ❖ You can launch EC2 instances from different kind of AMIs.
    - **A public AMI:** these are provided by AWS so Amazon Linux 2 AMI is a very popular AMI for AWS and it was provided by AWS themselves
    - **Your own AMI:** - therefore you have to make and maintain them yourself there are tools obviously to automate this but this is a task you have to do as a cloud user



- **An AWS Marketplace AMI:** - Which is AMI that has been made by someone else and potentially sold by someone else so it quite common to have vendors on AWS to create their own AMIs or their own software with nice configuration and so on and they will sell it through the marketplace AMI for you buy it and to save some time. (And even you as a user you could create a business of selling AMIs on the AWS Marketplace [this also called 3<sup>rd</sup> parties AMIs])
- ❖ **AMI Process (from an EC2 instances):** - Start an EC2 instances and customize it
  - Stop the instances (for data integrity)
  - Build an AMI – this will also create EBS snapshots
  - Launch instances from other AMIs
- **EC2 Image Builder:** - It is used to automate the creation of virtual machines or containers images. With EC2 image builder to automate the creation, maintain, validate and test AMIs for EC2 instances.
  - ❖ Can be run on a schedule (weekly, whenever packages are updated, etc..)
  - ❖ Free services (only you have to pay for underlying resources)
- **EC2 instance store:** - EBS volume are network drives with good but “limited” performance., If you need a high-performance hardware disk, use EC2 instances store.
  - Better I/O performance
  - EC2 instance store lose their storages if they’re stopped the is called ephemeral storage.
  - Good for buffer / cache / scratch data / temporary content – this would be a great place to do these things but not for long term storage. For long term storage, EBS for example is a great use case. Finally, in case the on the line server of the EC2 instances does fail, then you will risk to have a data loss because the hardware to the EC2 instance will fail as well. So if you do decide to use an EC2 instance store
  - Backup and replication are your responsibility make sure that you back it up and that you replicate it correctly based on your needs.
- **EFS (Elastic File System):** - Managed NFS Network file system that can be mounted on 100s of EC2 at a time. EFS only will work with your Linux EC2 instances and on top it, it works across multiple availability zones. So, it is possible for an instance in one AZ to be attaching the same EFS volume as the instance in another AZ.
  - ❖ Now, EFS is highly available, scalable, pretty expensive. But you pay for use and don’t plan for capacity. So that means that if you store 20GB of data onto your EFS drive then you’re only going to pay for these 20GB.
  - ❖ **EFS Infrequent Access (EFS-IA):** -
    - **Storage class** – That is cost-optimized for files not accessed every day
    - This storage class give you up to 90% lower cost of storing data compared to EFS Standard
    - If you enable EFS-IA, then EFS will automatically move your files to EFS-IA based on the last time they were accessed and something called a lifecycle policy
    - Example: move files that are not accessed for 60 days to EFS-IA
    - Transparent to the application accessing EFS
- **Shared Responsibility Model of EC2 storage:** -
  - ❖ **AWS:** - Infrastructure
    - ✓ Replicated for data for EBS volumes & EFS drives (data is replicated across many hardware to perform that replication let say if someday is hardware is not working you as a customer is not impacted)
    - ✓ Replacing faulty hardware
    - ✓ Ensuring their employees cannot access your data
  - ❖ **User:** - Setting up backup / snapshot procedures
    - ✓ Setting up data encryption
    - ✓ Responsibility of any data on the drives
    - ✓ Understanding the risk of using EC2 instance store
- **Amazon FSx:** - Launch 3<sup>rd</sup> party high-performance file system on AWS. Fully managed service. You have 3 kinds of offering FSx file system. **1) FSx for Lustre 2) FSx for windows file server 3) FSx for NetApp ONTAP**
  - They can add file system over time to the FSx service.
- 1) FSx for Windows File Server:** - A fully managed, highly reliable, and scalable Windows native shared file system

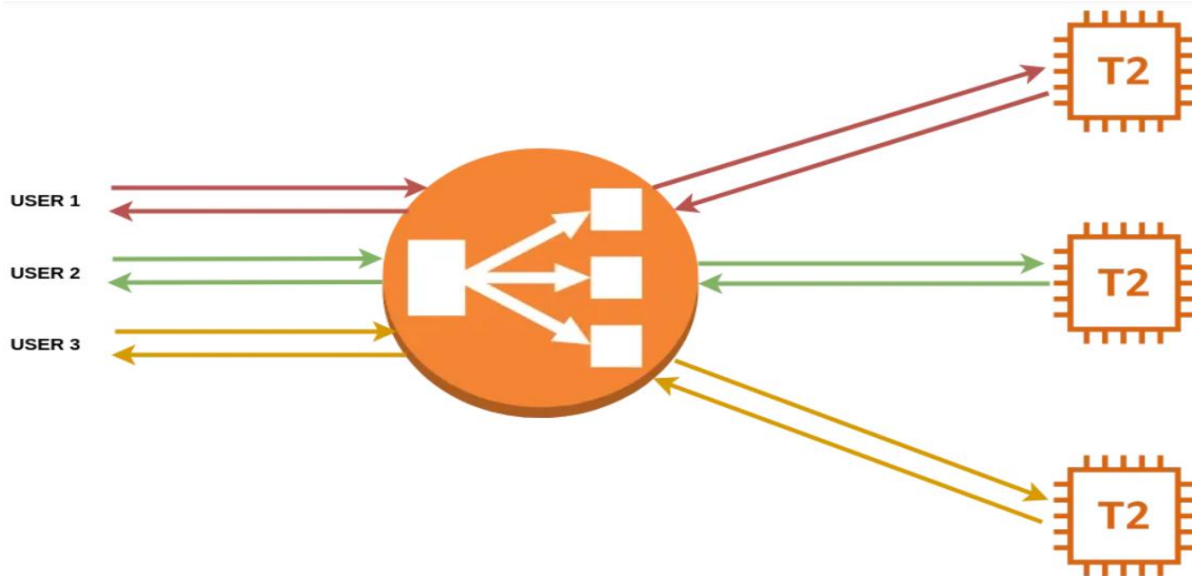
- ✓ Build on windows file server so this is meant for windows instances. So you way do it is that deploy the FSx usually across two availability zones, and then there is support for all the windows native protocols.
- ✓ Supports SMB protocol & Windows NTFS which allows you to mount this file system onto your windows machine.
- ✓ Integrated with Microsoft Active Directory
- ✓ Can be access from AWS or your on-premise infrastructure

2) **FSx for Lustre:** - A fully managed, high-performance, scalable file storage for **High Performance Computing (HPC)**, whenever you see storage for HPC, thinks FSx for Lustre.

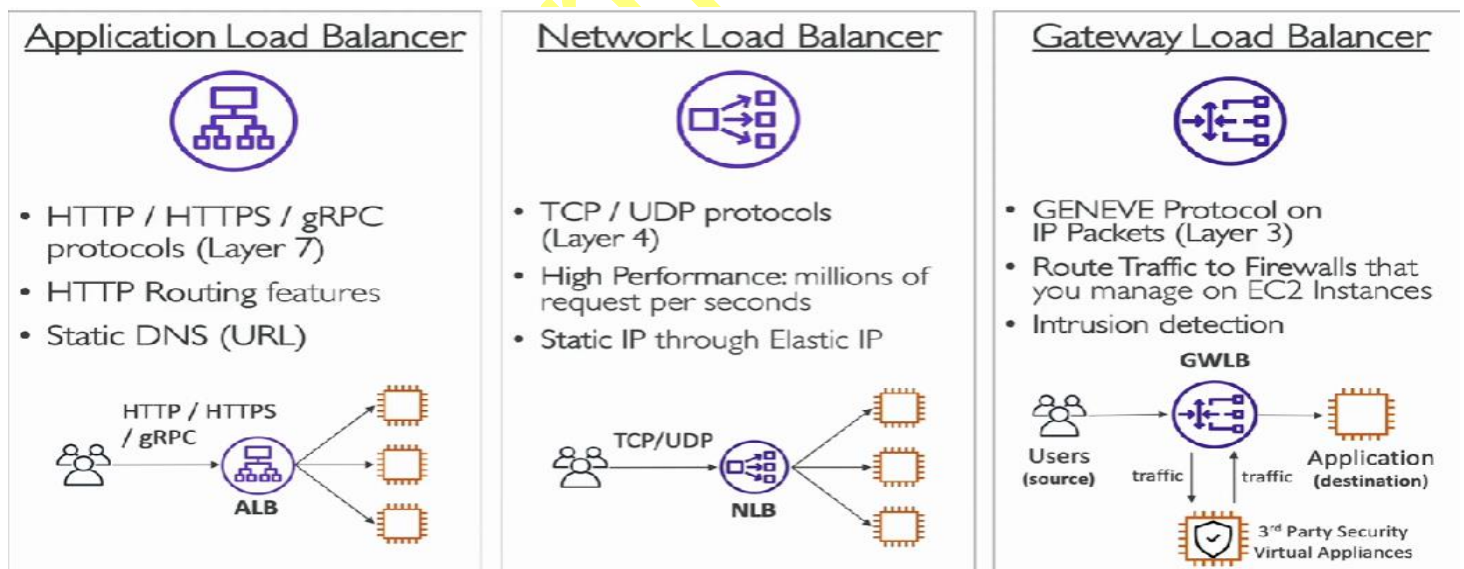
- ✓ The name Lustre is derived from "Linux" and "cluster" so put together it's Lustre
- ✓ This allow you to run lot of use cases for high performance comuting, such as machine learning, Analytics, Video Processing, Financial Modeling,..
- ✓ Scales to extremely high traffics up to 100s GB/s data exchanged, Millions of IOPS, sub-ms latency its really high performance file system

### ELB & ASG – Elastic Load Balancing & Auto Scaling

- **Scalability & High Availability:** - Scalability menas that an application / system can handel greater loads by adapting. There are two kinds of scalability:
  - ✓ Vertical Scalability
  - ✓ Horizontal Scalability also called elasticity
- ❖ **Vertical Scalability:** - Verical scalability means increasing the size of the instances. If instances is running on t2.micro and any how your load get increase and automatically get created a new instances like t2.large eq:- junior oprator and senior operator. Verical Scalability is very common for non distributed system, such as a database because on daily will increasse the data so you would just increase the size of your database. There's usually a limit to how much you can vertically scale (hardware limit).
- ❖ **Horizontal Scalability:** - Horizontal scalability means increasing the number of instances / system for your application. Eq:- one oprater is working there if get more work and not handeld by one person then add one more member for devided the task. This is very common for web application / modern applications.
- ❖ **High Availability:** - High availability usually goes hand in hand horizontal scaling. Hight availability means running your application / system in at least 2 Availability Zones. **(Run instances for the same application across multi AZ)**
- ❖ **Hight availability & Scalability for EC2:** -
  - ✓ **Vertical Scaling:** - Increase instances size (= scale up / down)  
**From:** t2.nano – 0.5G of RAM, 1 vCPU **to** u-12tb 1.metal – 12.3 TB of RAM, 448 vCPUs
  - ✓ **Horizontal Scaling:** Increase number of instances (= scale out / in) bu **Autoscaling Group, Load Balancer**
- **Scalability:** - Ability to accommodate a larger load by making the hardware stronger (scale up), or by adding nodes (scale out)
- **Elasticity:** - Elasticity is bit more cloud native, once system is scalabal either system scale up/out elasticity means that there will be some short of auto-scaling in it so that system can scale based on the load that it's receiving in this case, wh're going to pay per use, match demand, optimize costs.
- **Agility:** - (Agility is absolutely not related to scalability or elasticities, it's a distractor.) new IT resources are only a click away, which means that you can redices the tome to make those resources is available to your developers from weeks to just minutes. So your organization is more agile, it can iterate more quickly and you are going to faster.
- **Load Balacing:** - Load balancer are servers that forward internet traffic to multiple server (EC2 instances) downstrems. ELB something is manage by AWS. We can expose single point of access (DNS) to your application. You can seamlessly handel failures of downstream instances, we do regular health check of them to your instances if one of them is failing then the load balancer will not direct traffic to that instance so we can hide the failure of an instances using a load balancer.



- ❖ Can also provide SSL termination (HTTPS) for your websites and you are able to use a load balancer across multiple availability zones which is making your application highly available.
- ❖ ELB is a managed load balancer you don't need to be provisioning servers AWS will do for you.
- ❖ AWS guarantees that it will be working
- ❖ AWS takes care of upgrades, maintenance, high availability of that elastic load balancer.
- ❖ AWS provides only a few configuration knobs
- ❖ It cost less to setup your own load balancer but it will be a lot more effort on your end (maintenance, integrations) like on EC2 instances
- ❖ **4 Kinds of load balancers offered by AWS:**
  - ✓ **Application Load Balancer (HTTP / HTTPS only) – Layer 7**
  - ✓ **Network Load Balancer (ultra-high performance, allows for TCP) – Layer 4**
  - ✓ **Gateway Load Balancer – Layer 3**
  - ✓ **Classic Load Balancer (retired in 2023) – Layer 4 & 7**



- **Auto Scaling Group:** - If any instances get unhealthy due to application bugs then auto scaling group will identify and deregister and register new healthy instances automatically.
  - ❖ **Scaling Strategies:** - **Manual Scaling:** - Update the size of an ASG manually
  - ❖ **Dynamic Scaling:** - Respond to changing demand
    - **Simple / Step Scaling**
      - When a CloudWatch alarm is triggered (example CPU > 70%), then add 2 units
      - When a CloudWatch alarm is triggered (example CPU < 30%), then remove 1

- **Target Tracking Scaling**
  - Example: I want the average ASG CPU to stay at around 40%
- **Scheduled Scaling**
  - Anticipate a scaling based on known usage patterns
  - Example: Increase the min. capacity to 10 at 5 pm on Fridays
- **Predictive Scaling:** - Uses Machine Learning to predict future traffic ahead of time., Automatically provisions the right number of EC2 instances in advance., Useful when your load has predictable time-based patterns.

### S3 Simple Storage Service

- **S3:** - S3 is one of the main building blocks of AWS. Many website use amazon S3 as a backbone. “Infinity scaling” storage.
  - ❖ **S3 Use case:** -
    - ✓ **Backup and storage** – It could be for your files could be discs and so on
    - ✓ **Disaster Recovery** – You will move your data to another region in case region going down then your data backud up somewhere else
    - ✓ **Archive** – You can archive files in Amazon S3 and retrieve it at a later stage for much, much cheaper
    - ✓ **Hybrid Cloud Storage** – In case you have storage on premises, but you won’t expended into the cloud you can use amazon S3 for this
    - ✓ **Application hosting** – Host application, media such as video files, images, and so on
    - ✓ **Data lakes & big data analysis** – Have to data lake so to store a lot of data and to perform big data analytics
    - ✓ **Software delivery** – Delivering software updates
    - ✓ **Static website** – Hosting static websites
- **S3 – Buckets:** - Amazon S3 allow to store object (files) in “buckets” (directories/folder). Bucket must have a globally unique name (across all regions all accounts). Buckets are defined at the region level so S3 look like a global service but buckets are created in a region. **Lifecycle Rules** – Allow you to move S3 objects between different storage classes
  - ❖ **Naming Convention:** -
    - ✓ **No Uppercase, No underscore**
    - ✓ **3-63 character long**
    - ✓ **Not an IP**
    - ✓ **Must start with lowercase latter or number**
    - ✓ **Must NOT start with the prefix xn-**
    - ✓ **Must NOT end with the suffix -s3alias**
- **S3 Objects:** - Object files have a key. The key is full path like – s3://my-bucket/my\_file.txt / s3://my-bucket/my\_folder1/another\_folder/my\_file.txt.
  - ✓ **The key is composed of prefix + object name** (into the prefix, which is my folder one and another folder, and the object name, which is my file dot TXT)
  - ✓ So the max object size is 5(TB) terabytes. (5000GB)
  - ✓ If you upload a file that is very big and if that file is grater then 5GB okey then you must use the multi-part upload to upload that file into several parts. So if you have file in 5TB then you must upload at least 1000 parts of 5GB.
  - ✓ Now, object also can have metadata their list of key and value pairs, and that could be set by the system or set by the user to indicate some elements about the file, some metadata.
  - ✓ Tags (Unicode key / value pairs – up to 10) – useful for security / lifecycle
  - ✓ Version ID (if versioning is enabled)
- **S3 -Security:** -
  - ❖ **User Based: - IAM Policy** - As a user you have IAM policies that you and this IAM policy is going to authorize which API calls should be allowed for a specific IAM user.

- ❖ **Resource Based: - Bucket Policy** – This policy that are bucket wide rules that you can assign directly from the S3 console. And this will allow for example, a specific user to come in or allow a user from another account this is called cross account to access your S3 buckets. This is also how will make our S3 bucket public as you will so in minute.
- ❖ **Object access control list (ACL):** - They are fine grain security and they can be disabled
- ❖ **Bucket access control list (ACL):** - less common can be disabled
- ❖ **Note: an IAM principal can access an S3 object if**
  - ✓ The user IAM permissions ALLOW it OR the resources policy ALLOW it
  - ✓ There is no explicit DENY in the action then the IAM principal can access the S3 object on the specified API call.
- ❖ **Encryption:** - encrypt objects in Amazon S3 using encryption keys.
- **S3 – Static Website Hosting:** - S3 can host static websites and have them accessible on the internet. The website URL will be (depending on the region) on the AWS region where you create this, either this or that
  - ✓ <http://bucket-name.s3-website-aws-region.amazonaws.com> or
  - ✓ <http://bucket-name.s3-website.aws-region.amazonaws.com>
- **S3- Versioning:** - You can version your files in Amazon S3. This is setting you have to enable at bucket level. So whenever user uploads a file, it's going to create a version of that file at the selected key. Some key overwrite will change the version 1,2,3....
  - ✓ It is best practice to version your buckets. Protected against unintended deletes (ability to restore a version)
  - ✓ Can also easily roll back to a previous version if you want to go back to what happened two days ago you can take a file and roll it back.
  - ✓ **Note:** - Any file that is not versioned prior to enabling versioning will have version “null”, if Suspending versioning does not delete the previous versions.
- **S3 – Replication (CRR & SRR):** - We want to setup asynchronous replication between two buckets. So to do so we must first enable Versioning in source and destination buckets
  - ✓ If we do (CRR) Cross-Region Replication – the two region must be different.
  - ✓ If do (SRR) Same-Region Replication – the two region are the same
  - ✓ Now it's possible for you to have these buckets in different AWS accounts and copying happens asynchronously. So the replication mechanism happens behind the scenes in the background.
  - ✓ And make replication work, you must give proper IAM permissions to the S3 service so that it has the permissions to read and write from specified buckets.
- ❖ **Use Cases:** -
  - ✓ **CRR** – Compliance, lower latency access, replication across accounts
  - ✓ **SRR** – Log aggregation, live replication between production and test accounts
- **S3 Storage Classes:** -
  - ✓ **S3 Standard – General Purposra**, 99.99% Availability, Used for frequently accessed data, Low latency and high throughput, Sustain 2 concurrent facility failures, **Use Case:** - Big Data analytics, mobile & gaming applications, content distribution.
  - ✓ **S3 Storage classes – Infrequent Access (IA):** - For data that is less frequently accessed, but require rapid access when needed., Lower cost than S3 standard., but you have cost on retrieval., Availability 99.9 %, Use Case for Disaster Recovery, Backup
  - ✓ **S3 One Zone – Infrequent Access:** - Durability 99.999999999% in single AZ, data is going loss when AZ is destroyed, Availability – 99.5%, Use Case: - Strong secondary backup copies of on-premise data, or data you can recreate.
  - ✓ **S3 Glacier Storage classes:** - **Intant Retrieval:** - Low-cost object storage meant for archiving / backup., **Pricing:** Price for storage + object retrival cost., **There is 3 classes of storage within glacier:** - **Millisecond retrieval** - great for data accessed once a quarter., Minimum Storage duration of 90 days.,
  - ✓ **S3 Glacier Flexible Retrieval:** - **Expedited** – you get data back between 1 to 5 minutes., **Standard** get back data between 3 to 5 hours., **Bulk** get back data between 5 to 12 hours. - free, Minimum storage duration of 90 days.
  - ✓ **S3 Glacier Deep Archive:** - for long term storage – **Standard (12 hours)**, **Bulk (48 hours)**, Minimum Storage duration of 180 days., You maybe ready to wait a lot of time to retrieve data., Its give to lowest cost. **Amazon**



Glacier Deep Archive is the most cost-effective option if you want to archive data and do not have a retrieval time requirement. You can retrieve data in 12 or 48 hours.

- ✓ **S3 Intelligent Tiering:** - Small monthly monitoring and auto-tiering fee., Its going to allow you to move objects between excess tiers based on usage patterns., There is no retrival charges in S3 Intelligent tiering.
  - ✚ Frequent Access tier (automatic): default tier.
  - ✚ Infrequent Access tier (automatic): objects not accessed for 30 days
  - ✚ Archive Instant Access tier (automate): object not accessed for 90 days
  - ✚ Archive Access trie (optional): configurable from 90 days to 700+ days
  - ✚ Deep archive tier (optional): that you can configured for objects that have not been accessed between 180 days to 700+ days.
- ❖ When you create an object an object in Amazon S3 three, you can shoose it class, you can also modify its stoarge class manually, or can use Amazon S3 Lifecycle configurations to move the objects automatically between all these storage classes.
- ❖ **Durability:** -
  - ✓ High durability (**99.999999999%, 11 9's**) of object across multiple AZ
  - ✓ If you store 10,000,000 objects with Amazon S3, you can on average expect to incur a loss of a single object once every 10,000 years
  - ✓ Durabiity is same for all stoarge classes
- ❖ **Availability:** - Measures how readily available a service is
  - ✓ Varies depending on stoarge class
  - ✓ Example: - S3 standard has 99.99% availability = not available 53 minutes a year the service is not going to be available. That means you will get some errors when you deal with the services
- **S3 Encryption:** -
  - ❖ **Server-Side Encryption (Default):** - It is by default whenever you create a bucket or whenever you upload an object, it will be encrypted. (Whenever user upload the object in S3 bukket and then object when it arrives in the bucket is going to be encrypted by Amazon S3 for security purposes) The idea is server doing the encryption and therefor we called this is server side encryption. (**by default server-side encryption is always on**)
  - ❖ **Client-Side Encryption:** - When user will actually take the file will encrypted it before uploading it so lock is doen by user, and then put it in the bucket and that is called client-side encryption.
- **S3 Shared Responsibility Model:** -
  - ❖ **AWS –**
    - ✓ AWS is responsible for **all** infrastructure level (Global Security, Durability, availability, sustain concurrent loss of data in two facilities)
    - ✓ Internal configuration and vulnerability analysis.
    - ✓ And their own compliance validation internally within their infrastructure.
  - ❖ **User:** -
    - ✓ **S3 Versioning** - You are supposed to set up correctly S3 versioning
    - ✓ **S3 Bucket Policy** - to make sure you set up the right S3 Bucket Policy so that the data is protected within your buckets
    - ✓ **S3 Replication Setup** - You need to make sure that if you want verification you set it up yourself
    - ✓ **Logging & Monitoring** – You have enable yourself
    - ✓ **S3 storage classes** – Making sure you are using the most optimal cost storage cloud that is going to be most cost friendly is alos your responsibility
    - ✓ **Data encryption at rest and in transit** – If you wanted to encrypt your data onto your amazon S3 bucket that is up to you as well.
- **AWS Snow family:** - Highly-secure, portable device to collect and process data at edge, and migrate data into and out of AWS. There is have 2 use case for that first is either it's used to collect and process data at the edge, or to migrate data in and out of AWS.
  - ❖ **Data Migration:** - Have 3 different type devices within snow family migration. 1) Snowcone , 2) Snowball Edge, 3) Snowmobile
  - ❖ **Edge computing:** - 1) snowcone , 2) Snowball Edge

- ❖ Why do we want to do data migration with the AWS snow family let say we have huge amount of data like 10PB, 100TB at the time it take transfer lot of data over the network, it can take ot of time,

## Data Migrations with AWS Snow Family

	Time to Transfer		
	100 Mbps	1Gbps	10Gbps
<b>10 TB</b>	12 days	30 hours	3 hours
<b>100 TB</b>	124 days	12 days	30 hours
<b>1 PB</b>	3 years	124 days	12 days

### Challenges:

- Limited connectivity
- Limited bandwidth
- High network cost
- Shared bandwidth (can't maximize the line)
- Connection stability

**AWS Snow Family: offline devices to perform data migrations**

If it takes more than a week to transfer over the network, use Snowball devices!

- **Snowball Edge (for data transfer):** - Physical data transport solution: move TBs or PBs of data in or out of AWS with computing capabilities. Snowball Edge Storage Optimized devices are well suited for large-scale data migrations and recurring transfer workflows, as well as local computing with higher capacity needs.
  - ❖ Alternative to moving data over the network (and paying network fees)
  - ❖ Pay for data transfer job
  - ❖ Provide block storage and Amazon S3-compatible object storage
  - ❖ **Snowball edge storage optimized**
    - ✓ 80 TB of HDD capacity for block volume and S3 compatible object storage
  - ❖ **Snowball Edge compute optimized**
    - ✓ 42 TB of HDD capacity for block volume and S3 compatible object storage
  - ❖ **Use cases:** Large data cloud migration, DC decommission, disaster recovery
- **AWS Snowcone:** - Small, Portable computing, anywhere, rugged & secure, withstands harsh environments
  - ❖ Light (4.5 pounds, 2.1 kg)
  - ❖ Device used for edge computing, storage, and data transfer
  - ❖ 8 TBs of usable storage
  - ❖ Use snowcone where snowball does not fit (space-constrained environment)
  - ❖ Must provide your own battery / cables
  - ❖ Can be sent back to AWS offline, or connect it to internet and use AWS DataSync to send data
- **Snowmobile:** - We used this one exabytes of data moving service in or out of AWS (1EB = 1000 PB = 1,000,000 TBs)., Each snowmobile will have 100PB of capacity (use multiple in parallel)
  - ❖ High security: temperature controlled, GPS, 24/7 video surveillance it's quite way to transfer secure data.
  - ❖ **Better than snowball if you transfer more than 10PB**

# AWS Snow Family for Data Migrations



	Snowcone	Snowball Edge Storage Optimized	Snowmobile
Storage Capacity	8 TB usable	80 TB usable	< 100 PB
Migration Size	Up to 24 TB, online and offline	Up to petabytes, offline	Up to exabytes, offline
DataSync agent	Pre-installed		
Storage Clustering		Up to 15 nodes	

## ➤ Snow Family – Usage Process: -

- ✓ Request snowball devices from the AWS console for delivery
- ✓ Install the snowball client / AWS OpsHub on your servers
- ✓ Connect the snowball to your servers and copy files using the client
- ✓ Ship back the device when you're done (goes to the right AWS facility)
- ✓ Data will be loaded into an S3 bucket
- ✓ Snowball is completely wiped

## ➤ What is Edge computing: - Process data while it's being created on an edge location.

- ❖ A truck on the road, a ship on the sea, a mining station underground
- ❖ These location may have
  - ✚ Limited / no internet access
  - ✚ Limited / no easy access to computing power

## ➤ We setup a Snowball Edge / Snowcone device to do edge computing

## ➤ Use cases of Edge computing:

- ✚ Preprocess data
- ✚ Machine learning at the edge
- ✚ Transcoding media streams

## ➤ Eventually (if need be) we can ship back the device to AWS (for transferring data for example)

## ➤ Snowcone (smaller)

- ❖ 2 CPUs, 4 GB of memory, wired or wireless access
- ❖ USB-C power using a cord or the optional battery

## ➤ Snowball Edge – Compute Optimized

- ❖ 52 vCPUs, 208 GiB of RAM
- ❖ Optional GPU (usefull for video processing or machine learning)
- ❖ 42 TB usable storage

## ➤ Snowball Edge – Storage Optimized

- ❖ Up to 40 vCPUs, 80 GiB of RAM
- ❖ Object storage clustering available

## ➤ All: Can run EC2 instances & AWS Lambda function (using AWS IoT Greengrass)

## ➤ Long-Term deployment option: 1 and 3 years discounted pricing

## ➤ AWS OpsHub: - Historically, use snow family devices, you needed a CLI.

- ❖ Today you can use AWS OpsHub (a software you install on your computer / laptop) to manage your snow family device
- ❖ Unlocking and configuring single or clustered device
- ❖ Transferring files

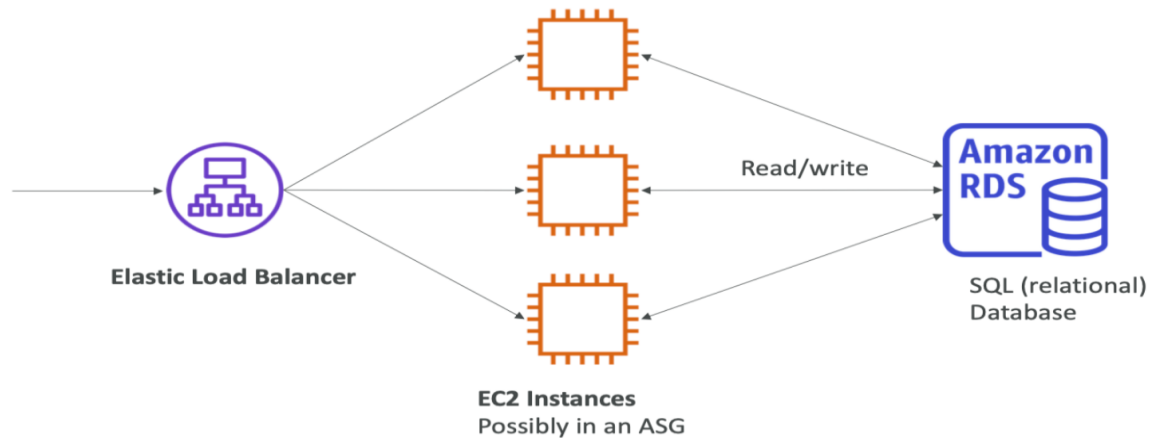
- ❖ Launching and managing instances running on snow family devices
  - ❖ Monitor device metrics (storage capacity, active instances on your device)
  - ❖ Launch compatible AWS services on your devices (ex: Amazon EC2 instances, AWS DataSync, Network File system NFS)
- **Hybrid Cloud for Storage:** - Also possible to use hybrid cloud in AWS so AWS want to bridge between your on-premises environment to AWS that called hybrid cloud. Below the point why we used cloud technology.
- ✓ Long cloud migrations
  - ✓ Security requirements
  - ✓ Compliance requirements
  - ✓ IT strategy
  - ❖ S3 is a proprietary storage technology so only we can exposing the S3 data on-premises so by the Storage Gateway can make establish connection between both (unlike EFS / NFS )
  - ❖ **Block Storage:** - EBS, EC2
  - ❖ **File Storage:** - EFS
  - ❖ **Object Storage:** - S3, Glacier
  - ❖ **Storage Gateway:** - Bridge between on-premises data and cloud data in S3. **Use case for that:** - Disaster recovery, backup & restore, tiered storage. **Type of storage gateway:** - File Gateway, Volume Gateway, Tape Gateway.

### Databases and Analytics

- ✚ **What is database:** - If you want to store the data so you have disk like EBS, EFS, EC2 and S3 like that to store unstructured way but can have limit. But for storing data in structure way need some kind of database to build the index to efficiency query / search through the data and define the relationship with datasets.
- ✚ **Relational Databases(SQL[RDBMS]):** - Assuming that your excel spreadsheet, with links between them with unique identifier it called SQL Database.,
- ✚ **No Relational Database NoSQL Database[DBMS]:** - We can store the data kind of without relational database(are purpose to build for specific data models and have flexible schemas for building modern application). Benefit of NoSQL database: - **Flexibility:** easy to evolve data model., **Scalability:** - designed to scale-out by using distributed clusters. In NoSQL data can be nested.
- ✚ **Benefit of using the AWS managed database(AWS database shared responsibility):** - **Benefits of include.**
  - 1) Quick Provisioning, High Availability, Vertical and horizontal Scaling
  - 2) Automated Backup and Restore, Operations, Upgrades
  - 3) Operating System Patching is handled by AWS
  - 4) Monitoring, alerting

**Note:** - Many databases technologies could be run on EC2, but you must handle yourself the resiliency, backup, patching, high availability, fault tolerance, scaling.....

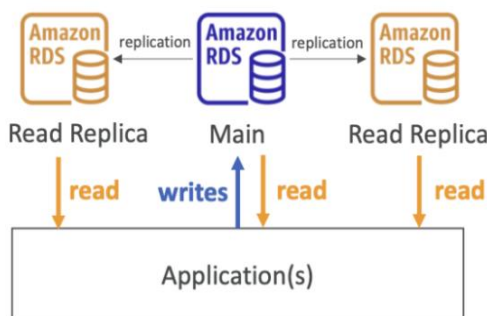
- **RDS Database:** - RDS database use SQL as a query language. It allow to create database in the cloud that are managed by AWS these are database it kinds a different types. (Postgres, MySQL, MariaDB, Oracle, Microsoft SQL Server, Aurora (AWS Proprietary database))
- ❖ **Advantage of using RDS instead deploying DB on EC2:** - RDS is managed services, Automated provisioning, OS patching take care by AWS, continuous backups and restore to specific timestamp (Point in Time Restore), Monitoring Dashboards for seen database doing good, Read replicas for improved read performance, Will have to set **Multi AZ setup** for DR (Disaster Recovery), Maintenance windows for upgrade, Scaling capability (vertical and horizontal), storage backup by EBS.



- ❖ With RDS and ElastiCache need to provision any instance types but with DynamoDB no need to.
- ❖ RDS Database is good for OLTP (Online Transaction Protocols)

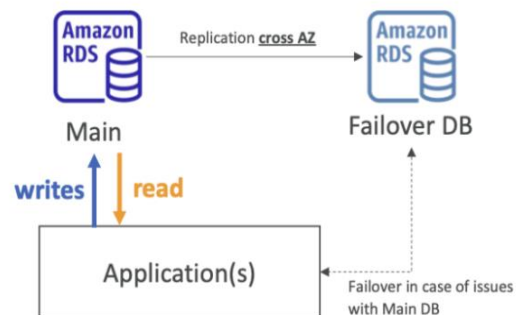
#### • Read Replicas:

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

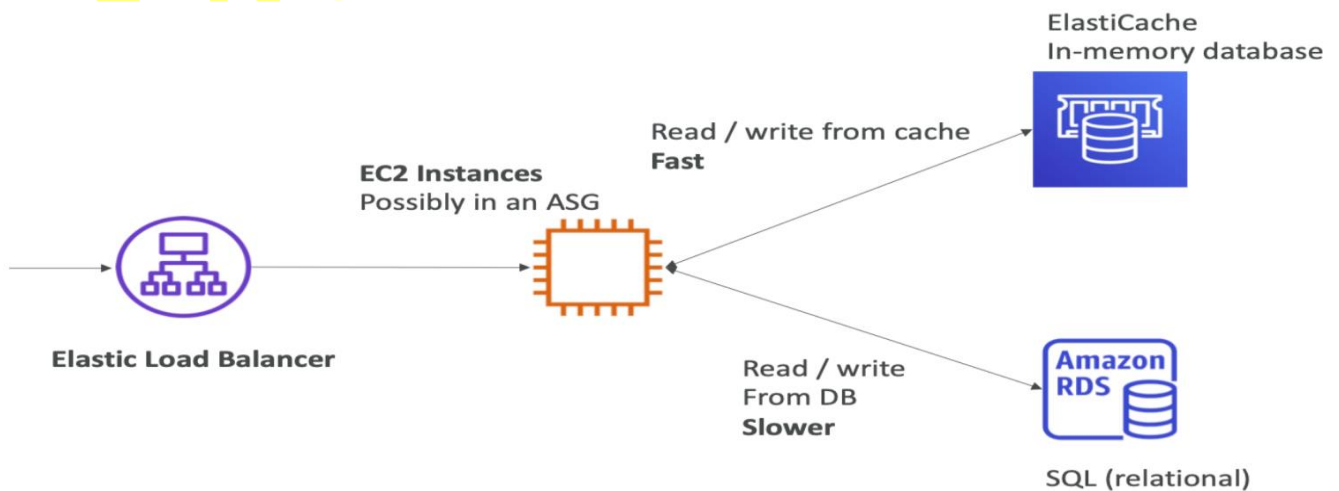


#### • Multi-AZ:

- Failover in case of AZ outage (high availability)
- Data is only read/written to the main database
- Can only have 1 other AZ as failover



- **Amazon Aurora:** - Aurora is a AWS database technology and not a open sourced. Aurora is only support 2 types technology, PostgreSQL and MySQL DB. Aurora is 5x performance improvement over MySQL on RDS, over 3x performance of Postgres on RDS. Storage automatically grows increments of 10GB, up to 64 TB. **(Aurora and RDS both are relational database)**. Aurora is not free tier.
- **Amazon ElastiCache:** - Store the data are in-memory database with high performance, low latency., AWS take care of OS maintenance / patching, optimizations, setup, configuration, monitoring, failure recovery and backup.

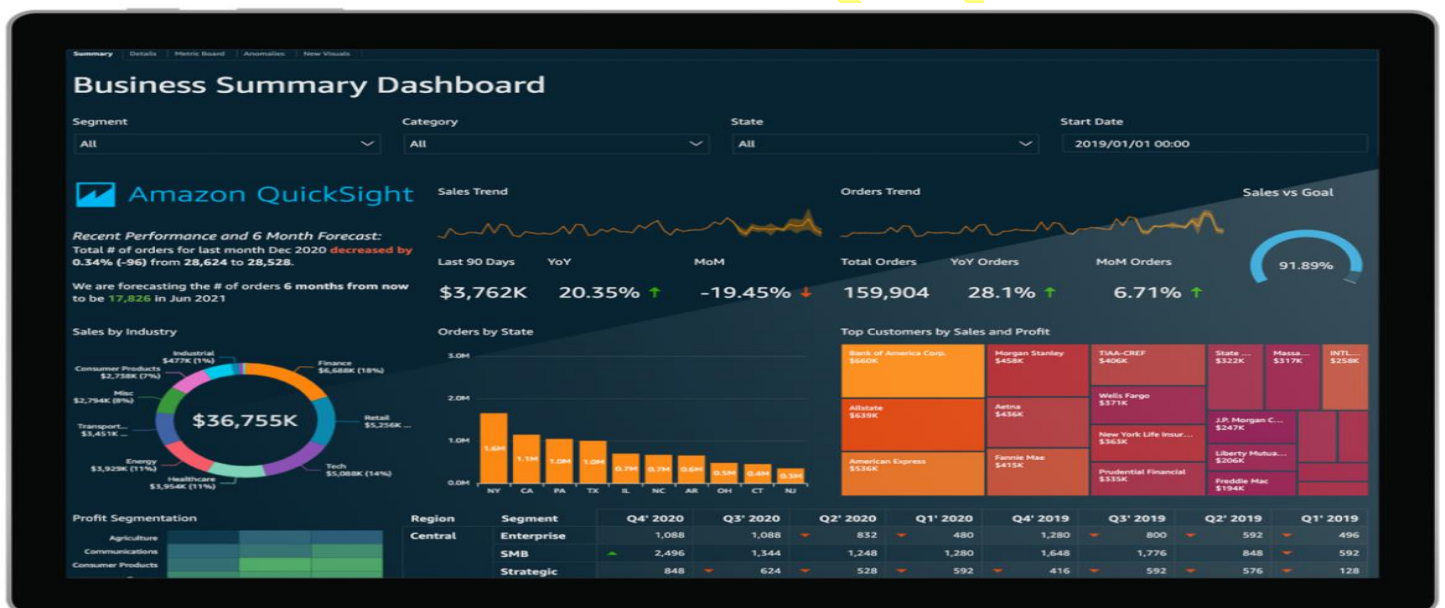


- **DynamoDB:** - Fully managed highly available with replication across 3-AZ, it is part of NoSQL Database – not a relational database. No need to provision any server it's serverless database. In DynamoDB store the data in key/value



database, Standard and Infrequent Access table class. DynamoDB Accelerator – DAX – Fully managed in-memory cache for DynamoDB and while accessing your database it provides millisecond/ microsecond latency.

- ❖ **Different Between DynamoDB Accelerator DAX and ElastiCache:** - **DAX** is only used and integrated with DynamoDB for cache database in-memory. **ElastiCache** can use with RDS managed databases for in-memory the data retrieve with low latency.
- ❖ **DynamoDB – Global Table:** - Accessible database with read/write low latency in **multiple regions**. Should to configure in Active-Active replication (read/write to any AWS region, 2 way replication).
- **Redshift Database:** - Redshift is based on PostgreSQL, but it's not used for OLTP, Load data once every hour, not every second, Pay-as-you-go based on the instances provisioned, it's used for data warehouses, scale to PBs of data and you can also create dashboard like BI tools with help of **QuickSight** or Tableau integrated tool.
- **EMR (Elastic MapReduce):** - It's not a database but it helps to create Hadoop clusters (Big Data) to analyse and process vast amount of data, Hadoop is an open source technology it's allow to work multiple server (EC2 instances) to analyse and collaborate the data work together, Also support Apache Spark, HBase, Presto, Flink to in Hadoop Cluster, it is also do Auto-scaling and integrated with spot instances.
- **Amazon Athena:** - Athena is serverless query service to perform analytics against the object which is store in S3. Analyze data in S3 using serverless SQL it's done by Athena, Business intelligence / analytics / reporting, analyze and query VPC flow logs, ELB logs, CloudTrail trails etc.
- **Amazon QuickSight:** - It's a tool just like BI in aws. Serverless machine learning-powered business intelligence service to create interactive dashboards and present the data visually, that is integrate with RDS, Aurora, Athena, Redshift and S3.



- **DocumentDB:** - DocumentDB is a NoSQL database (DocumentDB is same for MongoDB (Which is a NoSQL database), if anything is there NoSQL database it will be DocumentDB/MongoDB, it has store the data query and index JSON data, Fully Managed, highly available with replicate across **3-AZ**, DocumentDB storage automatically grows in increment of 10GB up to 64 TB. Deployment is having like Aurora DB.
- **Amazon Neptune:** - It's a fully managed graph database, it like social networking that is integrated like – chat, like, share, comment for everyone. Highly available accros 3-AZ, with up to 15 read replicas. Can store up to billions of relations and query the graph with milliseconds latency.
- **Amazon QLDB(Quantum Ledger Database):** - It is fully managed and serverless database, which is reading the financial transaction, highly available and replicated across 3 AZ. It is used to review history all the changes made to your application data over time. This is immutable system no entry can removed and modified.
- **Amazon Blockchain:** - This is a decentralise blockchain, with blockchain it possible to build the application where multiple parties can execute the transection without the need for a trusted, central authority. It is a managed blockchain is a managed service to join public blockchain and create your own scalable private network.

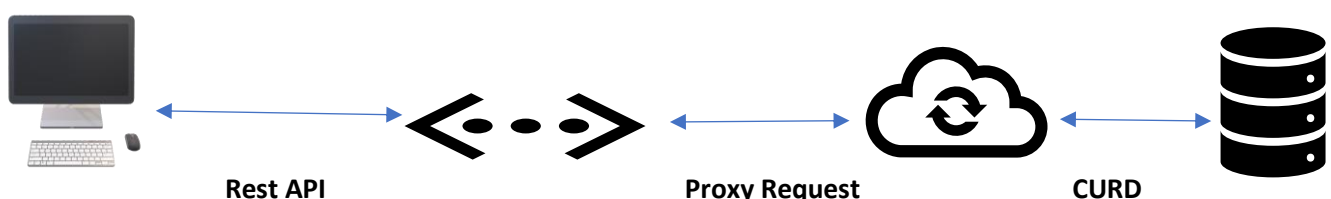
- **AWS Glue:** - Glue is managed extract, transform and load (ETL) service. Eq (if have data in unformat way so with help of glue you can prepare and transection the data for analytics in structure way) Its is fully serverless service. Data we can extract from S3 and RDS database and transform to in amazon redshift for analysis for structure.
- **DMS (Database migration service):** - With the help of DMS can extract the data from database to DMS (DMS will be run on EC2 instances) the send data to target database. With DMS quickly and secure migrate the database to AWS.

**Question:** - What is the name of a central repository to store structural and operational metadata for data assets in AWS Glue?

**Answer:** - Glue Data Catalog

### Compute Services: ECS, Lambda, Batch, Lightsail

- **Docker:** - Docker is software development platform to deploy apps. Before docket we have been install the apps by creating the virtual enviroment. Apps are packaged in container that can be run on any os. Apps run the same regardless where they'er run.
  - ❖ Any machine
  - ❖ No compatibility issues
  - ❖ Predictable behavior
  - ❖ Less work
  - ❖ Easier to maintain and deploy
  - ❖ Works with any laguages, any OS, any technology
  - ❖ Docker images are can store in Docker repositories. => Public Docker Hub <https://hub.docker.com/> here can find base image for many technologies or OS: Ubuntu, MySQL, NodeJS, Java etc. => Private: Amazon ECR (Elastic Container Registry)
- **Note:** - Scale containers up and down very quickly.(Seconds)
- **ECS(Elastic Container Service):** - It's used to launch docker containers on AWS. ECS must need to provision and maintnain infrastructure byself, AWS only take care starting and stoping the containers for you. Has integration with the Application Load Balancer if you host the web services. It is not a serverless.
- **Fargate:** - Fargate also used to launch the AWS containers on AWS. Here no need to provision EC2 instance to manage the container. This is serverless services offeing in AWS for lunch the container services. In ECR(Elastic Container Registry) those images you have saved you can run in ECS and Forgate both containers
- **Serverless services in AWS:** - S3, MangoDB, Fargate, Lambda, **Note:** - ECS/Fargate is preferred for runningarbitrary docker images, CodeBuild.
- **Lambda AWS:** - In lambda have vertual function to work on – no server to manage, this will run on-demand(so that means that whenever we run a function, there will be run and chnarged for pay) but whenever don't need a function it will be not run and be build for it. If need to scaling it will be automated scaling as be part of scaling. **Event-Drive** is a functions get invoked by AWS when needed and when an event happens or when needed. So that makes Lambda the reactivate the service. It is fully integrated many programming languages and easy to monitoring through CloudWatch. Also we can set CRON job in lambda function.
- **Pricing Model in lambda:** - We can pay for per call, first 1,000,000/1M request are free after that will be pay for \$0.20 per 1 million requests thereafter.
- **By the rest API gateway we can provide the lambda service to client**



**Support RESTful APIs and websocket APIs. Support for security, User Authentication, API throttling, API Keys, monitoring...**

- **AWS Batch:** - Batch is fully managed service in AWS that allow to batch processing at any scale sets. Batch job means you have started any task about at start 01 AM to finish 3 PM that is called batch job. Batch services will dynamically launch EC2 instances and spot instances to accommodate the load that you have to run these batch jobs. Batch will provision right amount of compute and memory for you to deal with your batch queue.
- **Different Between Batch and Lambda:** - **Batch:** - 1) No Limit Time, 2) Any runtime as long as it's packaged as a Docker image, 3) Rely on EBS / instance store for Disk Space, 4) Relies on EC2 (Can be managed by AWS)  
**Lambda:** - 1) Time Limit, 2) Limited Runtime, 3) Limited temporary disk space, 4) Serverless.
- **Amazon Lightsail:** - It's a stand-alone service, it gives you multiple services in one place like Virtual Server, storage, Database, and networking. Give the Low and predictable pricing
- **In exam perspective if seen that has no cloud experiences and need to get started quickly with low and predictable pricing without configuring much that will be answer is Lightsail.**

### Deploying and Managing Infrastructure at Scale Section

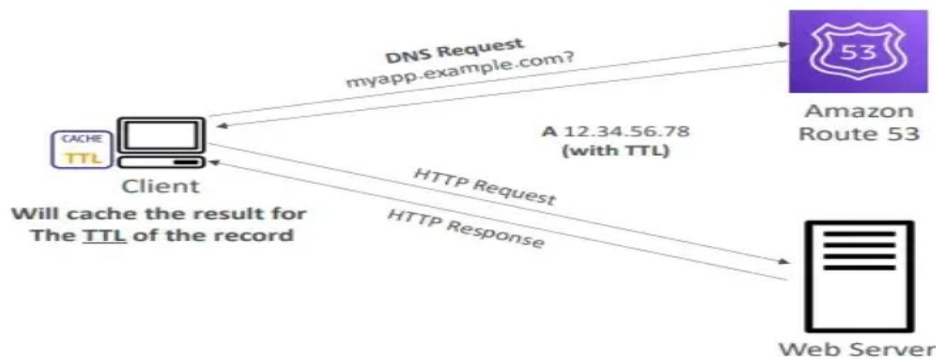
- **CloudFormation:** - CloudFormation is declarative way of outlining your AWS, infrastructure, for any resources.
  - ❖ For example, Within CloudFormation template you say: -
    - ✚ I want security group
    - ✚ I want two EC2 instances using the security group
    - ✚ I want an S3 bucket
    - ✚ I want a load balancer (ELB) in front of these machines
  - ❖ Then CloudFormation creates those for you, in the right order, with the exact configuration that you specify., CloudFormation allow you to deploy any AWS infrastructure as a code.
  - ❖ With the help of CloudFormation no need to create resources manually we can create by infrastructure as a code if in any resources are required to changes so need to modify and do in code.
  - ❖ Cost: - Each resource the stack is tagged with an identifier so you can easily see how much a stack cost you.
  - ❖ You can estimate your cost of your resources using the CloudFormation Template
  - ❖ Saving Strategy: In Dev you could set automation delete of template at 5 PM and recreated at 8 AM safely.
- **CDK (Cloud Development Kit):** - In CDK we can define cloud infrastructure using familiar coding languages.
  - ❖ JavaScript/TypeScript/Python/Java/.net
  - ❖ The code is compiled in CloudFormation Template (JSON/YAML)
  - ❖ You can deploy infrastructure and application run together. Grate for Lambda function and Grate for Docker containers in ECS / EKS
- **Beanstalk:** - It's a PaaS Services. developer centric view of deploying an application on AWS with full control over the configuration. It's a free service but you need to pay only underlying infrastructure, its managed service, Beanstalk is used for Application Health Monitoring – Health agent pushes metrics to CloudWatch, check for app health, publishes health events.
  - ✚ **There is 3 architecture model:** -
    - ❖ Single instance deployment: good for dev
    - ❖ LB + ASG: grate for production or pre-production web application
    - ❖ ASG: Only great for non-web apps in production (workers,etc).
- **AWS CodeDeploy:** - CodeDeploy work with our application automatically. This is work with EC2, On-Premises Server and Hybrid also.
- **AWS CodeCommit:** - In AWS provide codecommit for store code as like GitHub-Base repository for collaborate your code and it also provide the versioning code changes automatically. It is fully managed, Scalable & highly available, Private, Secured, Integrated with AWS.

- **AWS CodeBuild:** - This is provided you to build the code and deploy, Compiles source code, run tests, and produces packages that are ready to deploy. **Benefits:** - Fully managed serverless, continuously scalable and highly available, secure, Pay-as-you-go pricing model for build time.
- **AWS CodePipeline:** - Orchestrate the different steps to have the code automatically pushed to production like CI/CD. Code → Build → Test → Provision → Deploy
  - ❖ **Fully managed, compatible with CodeCommit, CodeBuild, CodeDeploy, Elastic Beanstalk, CloudFormation, GitHub, 3<sup>rd</sup>-Party services (GitHub) and common plugins., Fast Delivery and rapid updates.**
- **AWS CodeStar:** - **Unified UI** to easily manage software development activities in one place. Can edit the code “in-the-cloud” using AWS **Cloud9**. Cloud9 is use for writing, running, and debugging the code on cloud. At a time, many people can do work on same code in cloud9.
- **AWS System Manager (SSM):** - SSM is provide the system patching from AWS-to-AWS instances and On-Premises as well, it has worked as Hybrid model, we can install patches directly or by command line also. Works on both Windows and Linux. SSM is agent need to install on server but by-default install in some server like Amazon Linux AMI and some Ubuntu Linux.
- **AWS OpsWorks:** - OpsWorks Chef & Puppet help you perform server configuration automatically, or repetitive actions. OpsWorks is made by AWS to work as a Chef and Puppet and it is alternative of SSM, Chef and Puppet we can only provision EC2 instances, Databases, Load Balancer, EBS Volume.
- **Question:** - CodeStar can orchestrate the different steps to have code automatically pushed to production, while CodePipeline is a unified UI to easily manage software development activities in one place.  
**Answer:** - False/No
- **Question:** - CloudFormation and Elastic Beanstalk are free of use.  
**Answer:** - Yes/True
- **Question:** -Which of the following services can a developer use to store code dependencies?  
**Answer:** - **CloudArtifact**

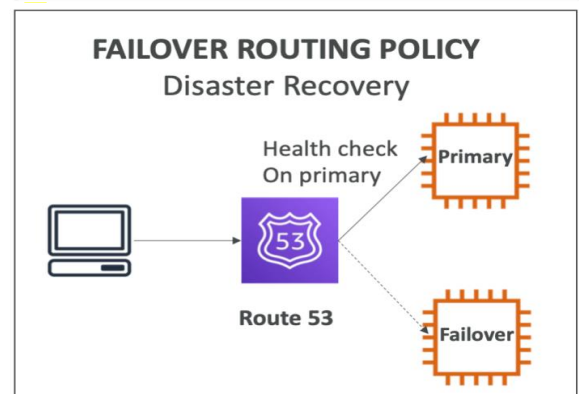
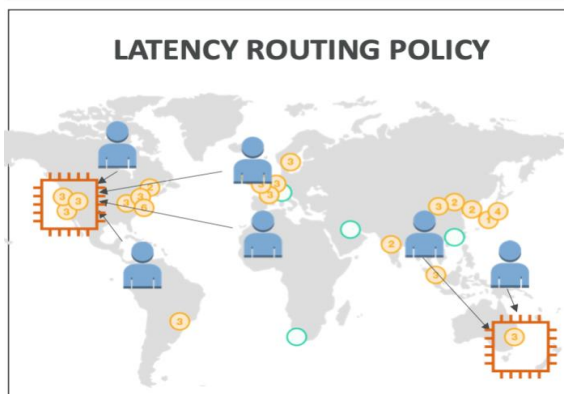
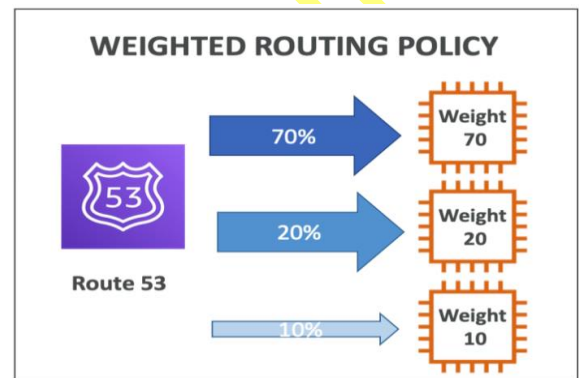
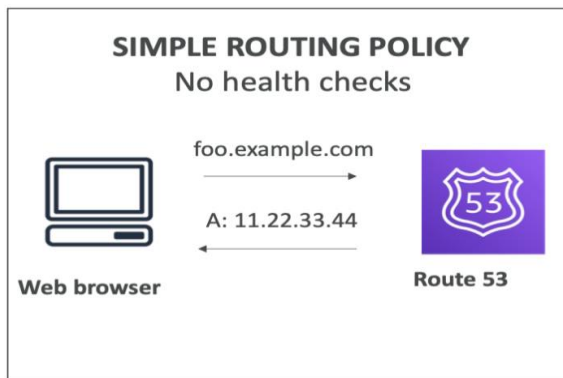
### AWS Global Infrastructure

- **Global Application:** - Global application we can deploy in multiple geographies this could be region or edge location. Benefit of the Global Application → Decreased Latency, Disaster Recovery (DR), Attack Protection.
  - ❖ Can't deploy the resources in CloudFront can use only as an CDN for low latency.
  - ❖ **S3 Transfer Acceleration:** - Accelerate global uploads and downloads into Amazon S3. Increase the transfer speed by transferring file to an AWS edge location which will forward the data to S3 bucket in target region.
  - ❖ **AWS Global Accelerator:** - Improve global application availability and performance using the AWS global network. Data routed between from host to edge location on AWS private network so that why provide the data with very low latency.
  - ❖ **Difference between Global Accelerator to CloudFront:** - They both use the AWS global network and its edge location around the world., Both services integrate with AWS shield for DDoS Protection.
  - ✚ **CloudFront:** - Improve performance for your cacheable content (Such as images and videos), Content is served at the edge.
  - ✚ **Global Accelerator:** - No caching, proxying packets at the edge to application running in one or more AWS region., Improve performance for a wide range of application over TCP or UDP., Good for HTTP use cases that require static IP address., Good for HTTP use cases that required deterministic, fast regional failover.
- **Region:** - For deploying application and infrastructure
- **Availability Zone:** - Made of multiple data centers
- **Edge Location (Points of Presence):** - For content delivery as close as possible to users
- **Route53:** - Route53 is a Managed DNS (Domain Name System)
  - ❖ DNS is a collection of rules and records which help clients understand how to reach a server through URLs.
  - ❖ In AWS most common records are: - [www.google.com](http://www.google.com) → 12.10.56.78 → A record (IPv4)

- ❖ [www.google.com](http://www.google.com) → 2001:0db8:85a3:0000:0000:8a2a:0370:5543 → AAAA IPv6
- ❖ Search.google.com → [www.google.com](http://www.google.com) → CNAME: hostname to hostname
- ❖ Example.com → AWS resources → Alias (ex: ELB, CloudFront, S3, RDS, etc ...)



- ❖
- ❖ Route53 Routing Policy: -



- **CloudFront:** - Is a Content Delivery Network (CDN) it is provide and improve the latency for read performance, content is cached at the edge location.

- ❖ **CloudFront – Origins**

- ✚ **S3 bucket:** -

- ✓ For distributing files and caching them at the edge
- ✓ Enhanced security with CloudFront Origin Access Control (OAC)
- ✓ OAC is replacing Origin Access Identity (OAI)
- ✓ CloudFront can be used as an ingress (to upload files to S3)

- ✚ **Custom Origin (HTTP)**

- ✓ Application Load Balancer
- ✓ EC2 Instance
- ✓ S3 website (must first enable the bucket as a static S3 website)
- ✓ Any HTTP backend you want

- ❖ **Different Between CloudFront and S3 Cross Region Replication:** -

- ✚ **CloudFront:** -

- ✓ Global Edge Network
- ✓ Files are cached for a TTL (maybe a day)



- ✓ Great for static content that must be available everywhere

#### **S3 Cross Region Replication:** -

- ✓ Must be setup for each region you want replication to happen
- ✓ Files are updated in near real-time
- ✓ Read only
- ✓ Great for dynamic content that needs to be available at low latency in few regions

- **AWS Outposts:** - Outpost give us Hybrid cloud infrastructure. Therefore, two ways dealing with IT system, one for AWS Cloud (Using AWS console, AWS CLI, AWS APIs, and tools to build your own applications on-premises just as in the cloud. AWS will setup and manage outpost racks within your on-premises infrastructure and you can start leveraging AWS services on-premises. But you are responsible for Outpost rack physical security.


❖ **Some Service that works on Outpost:** - Amazon EC2, Amazon EBS, Amazon S3, Amazon AKS, Amazon ECS, Amazon RDS, Amazon EME.


- **AWS WaveLength:** - Are infrastructure embedded within the telecommunication providers datacentres at the edge of 5G networks. Ultra-low latency application through 5G networks. High bandwidth and secure connection to the internet AWS region. No additional charges and service agreement. Use cases: - Smart Cities, ML-assisted diagnostics, Connected Vehicles, Interactive Live Video Streams, AR/VR, Real time gaming.


#### **Cloud Integration**

- Those application is dependent on one to another that is called **synchronous communications 1 → (application to application) 2 → Asynchronous / Event based (application to queue to application)**

❖ Here we used to decouple model your application for transmit data.

 Using SQS: - queue model

 Using SNS: - pub/sub model

 Using Kinesis: - real-time data streaming model

- **Amazon SQS:** - Fully managed service (serverless), used to decouple applications by queue. Scale from 1 message per second to 10,000s per second, default messages retention policy is 4 to 14 days., No limit to how many messages can be in the queue., Message are deleted after they're read by consumers. Consumers share the work to read & scale horizontally.

- **Amazon Kinesis:** - Kinesis is help to real time big data streaming. Managed service to collect, process and analyze real-time streaming data at any scale.

❖ **Kinesis Data Streams:** - Low latency streaming to ingest data at scale from hundreds of thousands of sources

❖ **Kinesis Data Firehose:** - Load streams into S3, Redshift, Elasticsearch, etc....

❖ **Kinesis Data Analysis:** - Perform real-time analysis on streams using SQL.

❖ **Kinesis Video Streams:** - Monitor real time videos streams for analytic and ML

- **Amazon SNS:** - When any time seen that publisher, email notification, any directly notification this will done by SNS. As many event subscribers as we want to listen to the SNS notification. SNS also used HTTP and HTTPS for send notification.

- **Amazon MQ:** - SNS, SQS it's a cloud native protocol Amazon MQ is On-premises protocol for send notification. Amazon MQ is managed message broker service for Rabbit MQ and Active MQ. Amazon MQ doesn't much scale as compared to SNS, SQS, Amazon MQ run on server and multiple AZ for failover and availability. Amazon MQ comes both features like SNS, SQS.

#### **Cloud Monitoring**

- **CloudWatch:** - CloudWatch provide metrics for every server monitor in AWS, CPU utilization, Network, Billing etc.... Can create CloudWatch dashboard of matrix. (RAM is not available to monitor in EC2) default matrix for monitor every 5 minutes but every 1 minute's monitoring is available, but it has most cost expensive. **EBS** volumes can monitor read/write. **S3 bucket** BucketSizeByte, NoOfObject, All request. **Billing** Total Estimated Charge (only in us-east-1). **Service Limits:** How much you've been using a service API. **Custom metrics:** push your own metrics.

- **Amazon CloudWatch Alarms:** - Alarms are used to trigger notifications for any metrics like **Auto Scaling:** increase and decrease EC2 instances "desired count" **EC2 Action:** stop, terminate, reboot, or recover EC2 instance. **SNS Notification:** send a notification into an SNS topic. **Various Alarms option:** (Sampling, %, max, min, etc). Also, can choose period on which to evaluate an alarm. **Example:** Create a billing alarm on the CloudWatch Billing matrix.

- **Amazon CloudWatch Logs:** - CloudWatch collect the log from those application run in your aws environment like Elastic Beanstalk collect log from application, ECS collection log from container, AWS Lambda collect from function log, if you install CloudWatch agent on EC2 instance or on-premises servers so that you can collect log also (By default, no log from EC2 instances will go CloudWatch Logs), Route53 DNS queries log can collect. This will allow to your real-time monitoring, and you can react what happening on you cloud infra.
- **Amazon EventBridge (Formerly CloudWatch Events):** - In EventBridge we can Schedule as per requirement (like every hour) Cron jobs (scheduled scripts). Also, can set event rule like whenever someone signing like Root user/other AWS services the notification will sent to respective person with help of SNS.
- **AWS CloudTrail:** - CloudTrail is provide governance, compliance, and audit for your AWS account. Whenever you use your account the CloudTrail is enable by default. CloudTrail give you an all the history event / API calls made with your AWS account. Example if someone login on Console, SDK, CLI, AWS Services the CloudTrail will login. And can put the log from CloudTrail to CloudWatch Logs or S3 in two locations for audit purpose. CloudTrail can be applied to all region (default) or a single region. If someone aws resources is deleted so with the help of CloudTrail we can investigate first.
- **AWS X-Ray:** - With the X-Ray we can tracing the application and go for visual analysis the application. We can debug the issue with the help of X-Ray.
  - ✓ **Troubleshoot Performance (bottlenecks)**
  - ✓ **Understand dependencies in a microservice architecture**
  - ✓ **Pinpoint service issues**
  - ✓ **Review question behaviour**
  - ✓ **Find error and exceptions**
  - ✓ **Are we meeting time SLA?**
  - ✓ **Where i am throttled?**
  - ✓ **Identify users that are impacted**

**Amazon CodeGuru:** - CodeGuru is ML powered service for automated code reviews and application performance recommendations. When developer push the code, another developer can review and write the code and able to monitor the performance of your code. CodeGuru provide two functionalities (1) **CodeGuru Reviewer:** - Automated code review for static code analysis (development) (2) **CodeGuru Profiler:** - Visibility/recommendation about application performing during runtime (production).

- **AWS Service Health Dashboard:** - What service are we using in AWS can monitor health related issue in all regions. Show the historical information for each day has been working historically. There is have RSS feed protocols you can subscribed for receive the notification about the services status.
- **AWS Personal Health Dashboard:** - Provide the health alert, proactive, scheduled activity, and remediation guidance when AWS is experiencing events that may impact you what you have deployed.

### **Security and Compliance**

- **AWS Shared model and responsibilities:** - Security of the cloud. Protecting infrastructure (hardware, software, facilities and networking) that run all AWS service. Also protect managed service like S3, DynamoDB, RDS etc. ➔ **Software, Compute, Storage, Database, Networking, Hardware & Global Infrastructure, Region, Availability Zone, Edge Location.**
- **Customer Responsibilities:** - Once the service provides you as a customer responsibility how you use that service. As a customer you are responsible for the security in the cloud. Like for EC2 instance, customer is responsible for management of guest OS including security patches and update, firewall and network configuration, and customer is responsible for EC2 instances has the correct IAM information through the use of IAM instance role. Then also need to ensure that the data is encrypted according to our compliance requirement. ➔ **Customer DATA, Platform, Application, Identity and access Management, Operating System, Network & Firewall configuration, Client-Side Data encryption & Data Integrity Authentication, Server-Side Encryption (File System and/or Data), Networking Traffic Protection (Encryption, Identity, Integrity)**
- **Shared Control:** - Patch Management, Configuration Management, Awareness & Training. Example for that let say if use RDS, the RDS patch management is done by AWS but for EC2 customer responsible for patch management. For Awareness & Training, AWS has train their employees to use their facilities correctly, and make sure they adhere

their security and guidance and customer have make sure to train your employees correctly, to use the cloud and doing this training is one of these ways.

- ❖ **RDS Responsibility → AWS Responsible:** -
  - ✓ Manage the underlying EC2 instances, Disable the SSH access
  - ✓ Automated DB Patching
  - ✓ Automated OS Patching
  - ✓ Audit the underlying instances and disk & guarantee it function
- ❖ **Customer Responsibility:** -
  - ✓ Check the port, IP, Security Group, inbound rules in DB's SG
  - ✓ In-Database user creation and permission
  - ✓ Creating a database with or without public access
  - ✓ Ensure parameter groups or DB is configuring to only allow SSL connections
  - ✓ Database encryption setting
- ❖ **S3 AWS responsible:** -
  - ✓ Guarantee you get unlimited storage
  - ✓ Guarantee you get encryption
  - ✓ Ensure separation of the data between different customers
  - ✓ Ensure AWS employees can't access your data
- ❖ **S3 Customer responsibility:** -
  - ✓ Bucket Configuration
  - ✓ Bucket Policy / public setting
  - ✓ IAM user and roles
  - ✓ Enabling encryption
- **DDoS, WAF, Shield:** - It's protected from malware, attacker from web attack by enabling DDoS, DDoS it's kind of firewall so in AWS cloud have multiple flavours for DDoS protection.
  - ❖ **AWS Shield Protection:** - Protect against DDoS attack for your website and application, for all customer at no additional cost.
  - ❖ **AWS Advance Shield:** - 24/7 premium DDoS protection
  - ❖ **WAF:** - Can filter specific request and rules, this is web application firewall
  - ❖ **CloudFront and Route53:** - These two service give us protection by using the global edge network so it's combined with shield it will provide attack mitigation at the edge locations.
- **AWS Shield:** - In the shield have two types of protection. Shield ONLY role is to safeguard running applications from DDoS attacks.
  - ❖ **Shield Standard:** - Free services that is activated for every AWS customer. Provides protection from attacks such as SYS/UDP Floods, Reflection attacks and other layer 3/layer 4 attacks.
  - ❖ **AWS Shield Advanced:** - Optional DDoS mitigation service, it's chargeable base very costly. It provides protection against most sophisticated attacks on EC2, ELB, CloudFront, AWS Global Accelerator, Route53. You can also access DDoS response team (DRP) when you needed to help you protect your self during these DDoS attacks.
- **WAF (Web Application Firewall):** - Protect your web applications from common web exploits (Layer 7). We can deploy on Application Load Balancer, API Gateway, CloudFront. We can define the rules from WACL (Web Access Control List) can filter → Based on IP Address, HTTP headers, HTTP body, or URI Strings, Block Countries, Rate base access (means user request more then 4-5 times per seconds).
- **Penetration Testing on AWS Cloud:** - Penetration testing is when you are trying to attack your own infrastructure to test your security. **Penetration Testing is allowed without prior approval on 8 services. DDoS, port flooding and protocol flooding are examples of prohibited activities.**
- **Encryption with KMS & CloudHSM:** - In AWS have two types of encryption happening. (1) Encryption At Rest (2) Encryption In Transit.
  - ❖ **At Rest:** - Data stored or archived on device on a physical device it could be Hard-Disk, RDS, S3 Glacier Depp Archive, etc. It's at rest because not moving, it's written somewhere.

- ❖ **In Transit:** - Is this means while data are moved from one place to another place. From on-premises to AWS, EC2 to DynamoDB, etc. Means data transfer to the on Network.
- **KMS (Key Management Service):** - Anytime you have "encryption" for an AWS it's most likely KMS. With KMS we don't have access of keys AWS will manage the keys for us and we just define who can access the keys. So there is option for encryption. With KMS AWS who manages the software for encryption.
  - ❖ **Encryption opt-In:** -
    - ✓ **EBS volumes: encrypt volumes**
    - ✓ **S3 buckets: Server-side encryption of objects**
    - ✓ **Redshift database: encryption of data**
    - ✓ **RDS Database: encryption of data**
    - ✓ **EFS drives encryption of data**
  - ❖ **Encryption Automatically enabled:**
    - ✓ **CloudTrail Logs**
    - ✓ **S3 Glacier**
    - ✓ **Storage Gateway**
- **CloudHSM:** - CloudHSM (HSM → Hardware Security Module) is second service in AWS to perform encryption it's called Cloud HSM. With Cloud HSM AWS will just provision to us the encryption Hardware but we are managing the keys ourselves.
- **CMK (Customer Manage Keys):** - With the CMK we can self-manage the encryption keys like create them, enable them, disable them. We can define the rotation policy for key generated every year, old key preserved. Also we can bring our own key.
- **AWS managed key:** - Create and managed and used on customers key behalf by AWS. They are used specifically and only by AWS services. Used by aws/S3, aws/ebs, aws/redshift.
- **AWS owned CMK:** - Collection of CMKs that an AWS services owns and manages to use in multiple accounts. AWS can use those to protect resources in your account (but you can't view the keys)
- **CloudHSM Keys (custom keystore):** - Key generated from your own CloudHSM hardware device. Cryptographic operations are performed within the CloudHSM cluster.
- **AWS ACM (Amazon Certificate Manager):** - Let you easily provision, manage, and deploy SSL/TLS Certificates. Use to provide in-flight encryption for website (HTTPS). Support both public and private TLS certificate. Free of charge for public TLS certificates. Automatic TLS certificate renewal. Integrate with load TLS certificate on → ELB, CloudFront Distributions, APIs on API Gateway. This is provide in-flight encryption and generate the certificates by ACM.
- **Secret Manager:** - By the secret manager we can set the policy of password for expire, regenerate and managing the secret in RDS, also can integrate with Amazon RDS, (MySQL, PostgreSQL, Aurora). Secret are encrypted using KMS. It is a paid service.
- **Artifact:** - Artifact portal provides customers with on-demand access to AWS compliance and documentation and AWS agreements. Can be used to support internal audit and compliance. It's a global service.
  - ❖ **Artifact Reports:** - Allows you to download AWS security and compliance documents from third-party auditors, like AWS ISO certificates, payment card industry (PCI), and System and Organization Control (SOC) reports.
  - ❖ **Artifact Agreements:** - Allow you to review, accept and track the status of AWS agreements such as the Business Associate Addendum (BAA) or the Health Insurance Portability and Accountabilities Act (HIPAA) for an individual account or in your organization.
- **Amazon GuardDuty:** - GuardDuty that will perform intelligent threat discovery in order to protect your AWS account. It is used to machine learning algorithm in the backend and do anomaly detection and used to 3<sup>rd</sup> party data to detect if your account is under attack. Once click to enable (30 days trail) no need to install software it works in the backend as it.
  - ❖ **Input Data includes:** -
    - 🚩 **CloudTrail Event Log:** - unusual API calls, unauthorised deployments
      - ✓ **CloudTrail S3 data event** – Get object, list object, delete object....
      - ✓ **CloudTrail Management event** – create VPC subnet, create trail...

- ✓ **VPC Flow Logs:** - unusual internal traffic, unusual IP address
- ✓ **DNS Logs:** - Compromised EC2 instances sending encoded data within DNS queries
- ✓ **Kubernetes Audit Logs:** - suspicious activities and potential EKS cluster compromises
- ✚ Can setup CloudWatch event rules to be notified in case of finding
- ✚ CloudWatch event rule can target AWS Lambda or SNS
- ✚ Can protect against Cryptocurrency attacks (has a dedicated “finding” for it)



- **Amazon Inspector:** - Inspector allows you to run automated security assessments on a couple of things. Analyze the unintended network, analyze the known vulnerabilities of OS. Also analyses the against known vulnerabilities ECR (Elastic container registry) images while pushed. Also reporting & integration with AWS security Hub, Send finding to Amazon Event Bridge, Check the network reachability to EC2, Continues scanning of the infrastructure, only when needed, (only evaluate EC2 instances, Container image & Lambda Functions)
- **AWS Config:** - Config helps to auditing and recording the compliance of your AWS resources. Helps record configuration and changes over time (whenever there have been manual changes are done of the configuration in AWS, we did not have list of all the changes that happened, but we can have the using config., then this configuration data can be stored in S3 to be later analyzed by Athena, or to be recovered. Config is not free service once you enable you have to pay.
- **Amazon Macie:** - Macie is fully managed data security and data privacy service that uses Machine Learning and pattern matching to discover and protect your sensitive data in AWS. Macie helps identify and alert you to security data, such as personally identifiable information (PII).
- **Security Hub:** - Security Hub is a central security tool to manage security several AWS accounts and automate security check. Integrate dashboards showing current security and compliance status to quickly take actions. Automatically aggregates alerts in predefined or personal finding formats from various AWS services & AWS partner tools. (For security Hub to work in the first place, you need to first enable the AWS config services.) It is not free service, but it has to give 30 day's trial.
  - ❖ **GuardDuty**
  - ❖ **Inspector**
  - ❖ **Macie**
  - ❖ **IAM Access Analyzer**
  - ❖ **AWS system manager**
  - ❖ **AWS firewall manager**
  - ❖ **AWS partner network solutions**
- **Amazon Detective:** - GuardDuty, Macie, and Security Hub are used to identify potential security issue, or findings. Sometimes security findings require deeper analysis to isolate the root cause and take action – it's a complex process. Amazon Detective **analyses, investigates,** and quickly identifies **the root cause of security issue** or suspicious or suspicious activities (using ML and graphs). Automatically collects and processes events from VPC Flow Logs, CloudTrail, GuardDuty and create a unified view. Produces visualizations with details and context to get to the root cause.
- **AWS Abuse:** - Report suspected AWS resources used for abusive or illegal purposes. Abuse prohibited behaviour are.
  - ✓ **Spam**
  - ✓ **Port Scanning**
  - ✓ **DoS and DDoS attacks**
  - ✓ **Intrusion attempts**
  - ✓ **Hosting objectionable or copyrighted content**
  - ✓ **Distributing Malware**



- ✓ **Contact AWS Team for abusing report. AWS Abuse Form, [abuse@amazonaws.com](mailto:abuse@amazonaws.com)**

➤ **Root User:** - This is account owner (created when the account is created). Has complete access to all of AWS account root user and access key. **Lock away your AWS account root user access keys!** Do Not use root account for everyday task, even administrative tasks.

✚ **Action that can be performed by AWS: -**

- ✓ **Change account settings** (account name, mail address, root user password, root user access keys)
- ✓ **View certain tax invoices**
- ✓ **Close your account**
- ✓ **Restore IAM permission**
- ✓ **Change and Cancel your AWS support plan**
- ✓ **Register as a seller in the Reserved Instance Marketplace**
- ✓ **Configure an Amazon S3 bucket to enable MFA**
- ✓ **Edit or Delete an Amazon S3 bucket policy that includes an invalid VPC ID or VPC endpoint ID**
- ✓ **Sign up for GovCloud**

### Machine Learning

➤ **Amazon Rekognition:** - It is provided to recognize objects, people, text, scenes in image and videos using ML so you can do facial recognition/ analysis and facial search to do verification and count people and with this, we can create a database of familiar faces or compare against celebrities.

- ❖ **Use Case: - (1) Labelling (2) Content Moderation (3) Text Detection (4) Face Detection and Analysis (Gender, age, range, emotion...) (5) Face search and Verification (6) Celebrity Recognition (7) Pathing (ex: for sport game analysis)**

➤ **Amazon Transcribe:** - It's allow to you automatically convert speech into text. Uses a **deep learning process** called **automatic speech recognition (ASR)** to covert speech to text quickly and accurately. Automatically remove Personally Identifiable Information (PII) using rekognition (age, name or social security number) this can be automatically removed. Support automatic language identification for multilingual audio so if you have some French and some English, Spanish, Transcribe is smart enough to recognize all of those.

- ❖ **Use cases: - (1) Transcribe customer service calls (2) Automate closed captioning and subtitling (3) Generate Metadata for media assets to create a fully searchable archive.**

➤ **Amazon Polly:** - You turn text into speech using deep learning. This allow you to create application that will talk.

➤ **Amazon Translate:** - As the name indicate, is a natural and accurate languages translation. Translate allow you to localize content, for example your website and your application for your international users and easily translate large volume of text efficiently.

➤ **Amazon Lex and connect:** - AWS Lex is the same technology that powers the Alexa devices by Amazon. Automatic Speech Recognition (ASR) to convert speech to text. Natural Language understanding to recognise to intent of text, caller. Helps build chat-boats, call centre bots.

➤ **Amazon Connect:** - It's a visual contact center that allow you to receive calls, create contact flows and it's all cloud based and it can integrate with other CRM system or AWS. No upfront payment, payments 80% cheaper than traditional contact center solutions.

➤ **Amazon Comprehend:** - It is for Natural Language Processing, or NLP. Fully managed and serverless service. Uses machine learning to find insights and relationships in text.

- ❖ **Language of the text**
- ❖ **Extracts key phrases, places, people, brands or events**
- ❖ **Understand how positive or negative the text is**
- ❖ **Analyzes text using tokenization and parts of speech**
- ❖ **Automatically organizes a collection of text files by topic**

✚ **Simple and Use case: -**

- ✓ **Analyze customer interaction (email) to find what leads to a positive and negative experience**
- ✓ **Create and groups articles by topics that Comprehend will uncover**

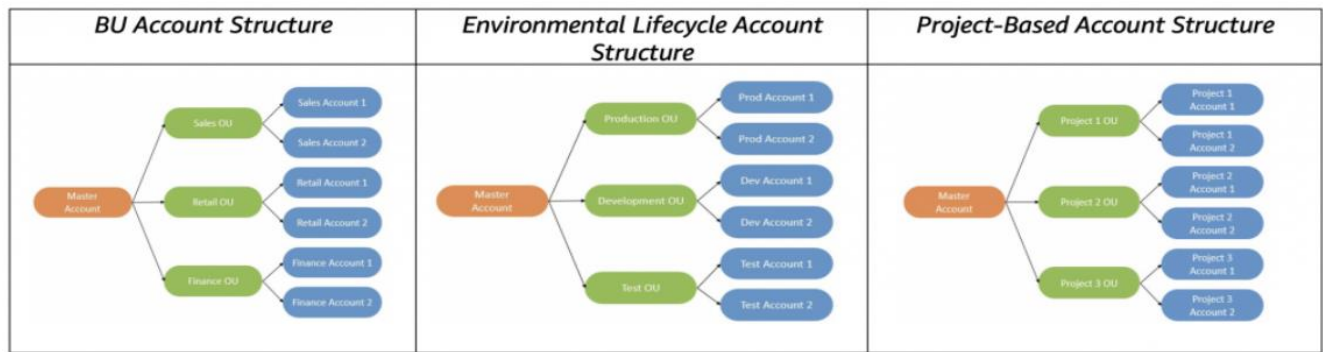
➤ **Amazon SageMaker:** - SageMaker fully managed service for developers / da/a scientists to build ML models. Typically difficult to do all the processes in one + provision servers. Machine Leaning process (simplified): predicted

your exam score. So basically SageMaker and all of this will help you with labelling, the building, training but not only we have machine learning model and it is created, it is fully working.

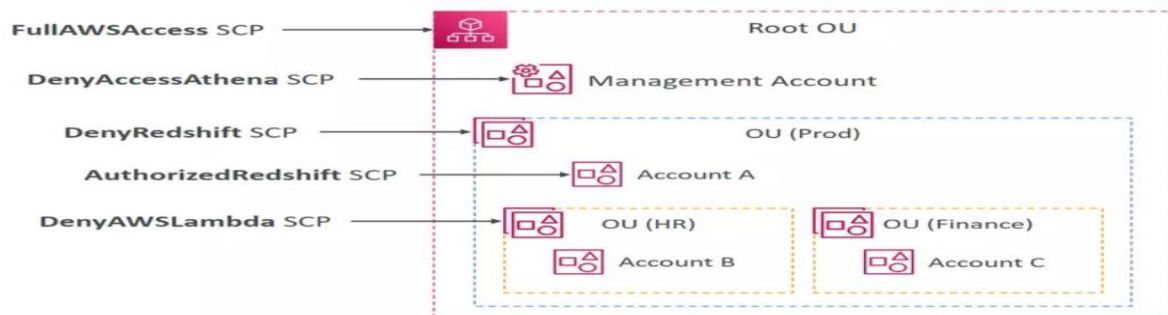
- **Amazon Forecast:** - This is also fully managed service that uses ML to deliver highly accurate forecast.
  - ❖ Example: Predict the future sales of a raincoat.
  - ❖ 50% more accurate than looking at the data itself
  - ❖ Reduce forecasting time from months to hours
  - ❖ Use cases: Production Demand Planning, Financial Planning, Resources Planning...
- **Amazon Kendra:** - Fully managed document search service powered by ML. It allow you to extract answers from within a document (text, pdf, HTML, PowerPoint, MS Word, FAQs.) Natural Language search capabilities.
- **Amazon Personalize:** - Fully managed ML-service to build apps with real-time personalized recommendations.  
Example: Personalized product recommendations/re-ranking, customize direct marketing.
  - ❖ **Example:** User bought gardening tools, provide recommendations on the next one to buy.
  - ❖ Same technology used by Amazon.com
  - ❖ Integrate into existing website, application, SMS, email marketing systems.
  - ❖ Implement in days, not months (you don't need to build, train and deploy ML Solutions)
  - ❖ Use cases: retail stores, media and entertainment.
- **Amazon Textract:** - Automatically extracts text, handwriting, and data from any scanned documents using AI and ML.
  - ❖ Extract Data from tables
  - ❖ Reads and process any types of documents (PDFs, images etc)
  - ❖ User cases: -
    - ✓ Financial Service (e.g., medical record, insurance claims)
    - ✓ Public Sector (e.g., tax form, ID documents, passports)

#### Account Management, Billing & Support Section

- **AWS Organization:** - It is Global Service, Allow to manage multiple AWS accounts, the main account is called master account and all the other ones will be called child accounts.
  - ❖ **Cost Benefits:** -
    - ✓ Consolidate Billing across all accounts – single payment methods (All the accounts just paid by master account so you will have one longer bill at the end so don't need to setup payment method for all the other accounts.
    - ✓ Pricing Benefit – From aggregated usage when you use lot EC2, S3 you get discount because you have used that at lots. Within an organization billing is consolidated the aggregated usage is as well consolidated that means you get more discounts.
    - ✓ Pooling for Reserved EC2 instances – if you are using reserved instances they are shared across all the accounts make sure that if one account does not use a reserved instance another one can and again, maximise the cost saving.
  - ❖ API is available to automate the AWS accounts creation to do so automatically, which is very helpful. For example if you have some processes to create an accounts programmatically for someone.
  - ❖ Restrict account privilege using Service Control Policy (SCP)
  - ❖ **Multi Account Strategies:** - Create account per department, per cost center, per dev / test / prod, based on regulatory restrictions (using SCP), for better resources isolated (ex:VPC), you have could different VPC for different account it a good for account separate per account service limit and also can isolate account for logging.
  - ❖ Use tagging standards for billing purposes. Should enable CloudTrail on all accounts , send the log to central S3 accounts. Send CloudWatch logs to central logging accounts.



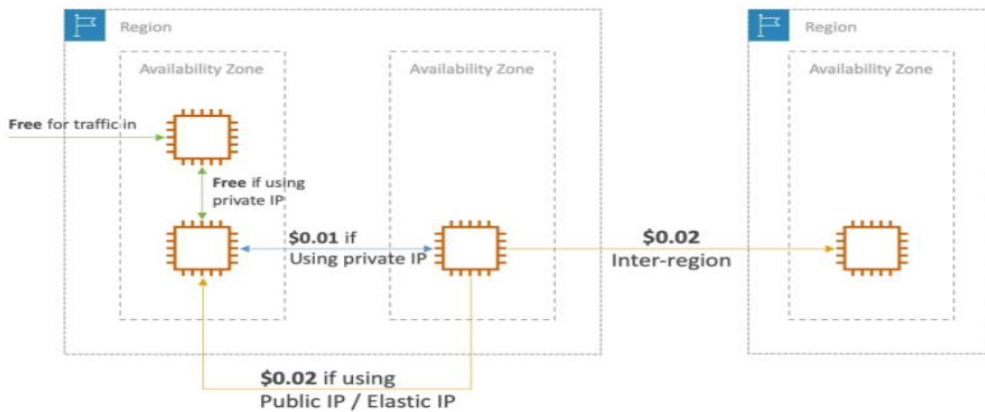
- ❖ **Service Control Policy (SCP):** - Whitelist and blacklist IAM actions, Applied at the OU or Account level, Does not to apply to the Master Account, SCP is applied to all the User and Roles of the Accounts, Including Root, The SCP does not affect service linked roles.
  - ✓ Service-linked roles enable other AWS services to integrate with AWS Organizations and can't restricted by SCPs.
- ❖ SCP must have explicit allow (does not allow anything by default)
- ❖ **Use Case:** -
  - ✓ Restrict access to certain services (for example can't use EMR)
  - ✓ Enforce PCI compliance by explicitly disabling services



- **AWS Organization – Consolidated Billing:** - When you enable provide two thing.
  - ❖ **Combined Usage:** - Combine the usage across all AWS accounts in the AWS accounts in the AWS organization to the volume pricing, Reserved instances and saving plans discounts.
  - ❖ **One Bill:** - Get one bill for all AWS accounts in the AWS organization.
  - ❖ The management account can turn off Reserved instances discount sharing for any account in AWS organization, including itself.
- **AWS Control Tower:** - Easy way to set up and govern and secure and compliant multi-account AWS environment based on best practices.
  - ❖ **Benefits:**
    - ✓ Automate the set-up of your environment in a few clicks
    - ✓ Automate ongoing policy management using guardrails
    - ✓ Detect Policy violations and remediate them
    - ✓ Monitor compliance through an interactive dashboard
  - ❖ **AWS Control Tower runs on top of AWS organizations:**
    - ✓ It automatically sets up AWS organization to organize accounts and implement SCPs (Service Control Policies)
- **AWS Pricing Model:** - AWS has 4 pricing model.
  - ❖ **Pay-As-You-Go** – Pay for what you use, remain agile, responsive, meet scale demand.
  - ❖ **Save when you reserve** – Minimize Risks, predictably manage budgets, comply with long-terms requirements. ➔ Reservation are available for EC2 Reserved Instances, DynamoDB Reserved Capacity, ElastiCache Reserved Nodes, RDS Reserved Instances, Redshift Reserved Nodes.
  - ❖ **Pay less by using more:** - Volume-based discounts for example S3.

- ❖ **Pay less as AWS grows:** - For example a instances are using but after some time the number of more users are add for used same server and you increase the instance size/type/add new instances in that scenario AWS grow to get you discount.
- ❖ **Upfront:** - Means you pay your amount in starting get more discount.
- ❖ **Partial Upfront:** - Means you pay your bill amount in meddle of purchasing type.
- ❖ **No Upfront:** - Means you pay your bill in end of the month/year.
- **Compute Pricing – EC2: - On-Demand instances**
  - ✓ Minimum of 60s
  - ✓ Pay for second (Linux/Windows) or per hour (other)
    - ❖ **Reserved Instances:** -
      - ✓ Up to 75% discount compared to On-Demand on hourly rate
      - ✓ 1- or 3-Years commitment
      - ✓ All upfront, partial upfron, no upfront
    - ❖ **Spot Instances:** -
      - ✓ Up to 90% discount compared to On-Demand on hourly rate
      - ✓ Bid for unused capacity
    - ❖ **Dedicated Host:** -
      - ✓ On-Demand
      - ✓ Reservation for 1 year or 3 years commitment
    - ❖ **Saving Plans:** -
      - ✓ As an alternative to save on sustained usage
- **Compute Price – Lambda & ECS: -**
  - ❖ **Lambda:** - (1) Pay per call, (2) Pay per duration
  - ❖ **ECS:** - EC2 Launch Type Model: No additional fees, you pay for, AWS resources stored and created in your application
  - ❖ **Fargate:** - Fargate Lunch type Model: Pay for vCPU and memory resources allocated to your application in your containers
- **S3 price:** - Sending data in S3 is free but retrieving data from S3, you will have to pay something. If you used S3 Transfer Acceleration it is also a cost that. Lifecycle transitions between the storage classes then you have to go for pay.
- **Similar Service: EFS** (Pay per use, has infrequent access & lifecycle rules)
- **EBS Pricing:** - In EBS have pricing base on volume types that we provision. Storage volume in GB per month provision.
  - ❖ **IOPS:** -
    - ✓ General Purpose SSD: Included
    - ✓ Provisioned IOPS SSD: Provisioned amount in IOPS (you're going to pay for the IOPS provision).
    - ✓ Magnetic: Numbers of request
  - ❖ **Snapshots:** -
    - ✓ Added data cost per GB per month (You are go to get a cost per GB of snapshot per month.
    - ✓ Any data transfer out from EBS is going to be paid and is going to be tiered for volume discounts.
    - ✓ Anything you write into EBS inbound is going to be free.
- **Database Pricing – RDS:** - Per hour billing. Base on database you have use it all has different pricing. (For RDS you have to pay for the number of input and output request per month.
  - ❖ **Database characteristics:** - Engine, Size, Memory Class,
  - ❖ **Purchase type:** - On Demand, Reserved instances (1 or 3 year) with require up-front
  - ❖ **Backup Storage:** - There is no additional charge for backup storage up to 100% of total database storage for a region.
- **CloudFront (CDN) Pricing:** - Pricing is different across different Geographic/country regions. Aggregated for each edge location, then applied to your bill. You are going to pay any data transfer out from CloudFront but not for in. Also you going to pay for HTTP/HTTPS based on the number of requests that are made into CloudFront.

# Networking Costs in AWS per GB - Simplified



- Use Private IP instead of Public IP for good savings and better network performance
- Use same AZ for maximum savings (at the cost of high availability)

- For the data goes into EC2, is Free
- Transfer data inside same AZ is free
- Transfer data from one AZ to another through Private IP: cost \$0.01
- Transfer data from one AZ to another through Public Ip / Elastic IP: cost double
- Transfer data from one Region to another Region double \$0.02

## Summary:

- Prefer private IP over public IP for saving costs and better network performance
- Using same AZ for maximum saving (at the cost of high availability)

- **Saving Pane:** - Saving plan has come to Cost Explorer for exploring your instances saving plan.
- **Compute Optimizer:** - Reduce the costs and improve performance by recommendation optimal AWS resources for your workloads. Helps you choose optimal configurations and right – size your workloads (over/under provisioned). Use machine learning to analyze your resources configuration and their utilization CloudWatch metrics.
- **Billing and Costing tools:** - There are multiple tools have for analysis the cloud services.
  - ❖ **Estimate cost in the cloud:** - Pricing Calculator
  - ❖ **Question:** - A company is not sure whether or not it is cost-effective to migrate to the AWS Cloud. Which service can help the executive board make a decision? Answer is: - **Pricing Calculator**
  - ❖ **Tracking cost in the cloud:** - Billing Dashboard, Cost Allocation Tag, Cost and Usage report, Cost Explorer
  - ❖ **Monitoring against costs plans:** - Billing Alarms, Budgets
- **Billing Dashboard:** - Billing Dashboard is powerful tools which will show you all the cost actually for the month, the forecast, and the month-to-date. On this page also will show you free tier which usages for each three-tier based on what you have doing so far for the month.
- **Cost Allocation tag and usage report:** - We can tag with aws resources and also with particular service for analysis the cost and generate the report and track accordingly to services and child account. Also can be integrated with Athena, Redshift and QuickSight for made visually dashboard.
- **Cost Explorer:** - Visualize, understand and manage your AWS costs and usage over time. Create custom reports that analyze cost and usage data. Analyze your data at a high level: total costs and usage across all accounts. Or monthly, hourly, resources level granularity. Choose an optional Saving Plan (to lower prices on your bill). **Forecast usage up to 12 months based on previous usage.**
- **Billing Alarms in CloudWatch:** - Billing data metrics is stored in CloudWatch us-east-1, Billing data are for overall worldwide AWS costs. It's for actual cost, not for projected costs. Intended a simple alarm (not as powerful as AWS Budgets).
- **AWS Budgets:** - Create budgets and send alarm when costs exceeded to budget. 3 types of budgets: Usage, Cost, Reservation.
  - ❖ **For Reserved Instances:** - Track utilization, Supports EC2, ElastiCache, RDS, Redshift



- ❖ Up to 5 SNS notification per budget
- ❖ Can filter by: Service, Linked Account, Tag, Purchase option, Instance type, Region, Availability Zone, API Operation, etc, Same options as AWS Cost Explorer!, 2 Budget are free, then \$0,02/day/budget.
- **Trusted Advisor:** - No need to install anything – high level AWS account management. Analyze your AWS accounts and provides recommendations on 5 categories. **(1) Cost optimization (2) Performance (3) Security (4) Fault tolerance (5) Service Limits.**
- **Trusted Advisers – Support Plan:** - There are few types of support plans.
  - ❖ **7 Core Checks:** - Basic & Developers Support Plan
    - ✓ S3 Bucket Permissions
    - ✓ Security Groups – Specific Ports Unrestricted
    - ✓ IAM use (one IAM user minimum)
    - ✓ MFA on Root account
    - ✓ EBS Public Snapshot
  - ✓ Service Limits
  - ❖ **Full checks:** - **Business & Enterprise support plan**
    - ✓ Full checks available on the 5 categories
    - ✓ Ability to set CloudWatch alarms when reaching limits
    - ✓ Programming Access using (AWS support API)
- **Support Plan Pricing:** -
  - ❖ **Basic Support:** (Free) **Customer Service Communities:** - 24\*7 access to customer service, documentation, whitepapers, and support forums.
  - ❖ **AWS Trusted Advisor:** - Access to the 7 core Trusted Advisor checks and guidance to provision your resources following best practices to increase performance and improve security.
  - ❖ **AWS Personal Health Dashboards:** - A personalized view of the health of AWS services, and alerts when your resources are impacted.
- **Developer Support Plan:** - Give all Basic support plan +
  - ❖ **Business hours email access** to cloud support associates.
  - ❖ Unlimited cases / 1 Primary contact
  - ❖ Case severity / response times: (1) **General guidance:** < 24 Business hours (2) **System impaired:** < 12 business hours
- **Business Support Plan (24/7):** - Intended to be used if you have production workloads
  - ❖ **Trusted Advisor:** - Full set of checks + API access
  - ❖ **24\*7 phone, email, and chat access** to cloud support engineers
  - ❖ Unlimited cases / unlimited contacts
  - ❖ Access to infrastructure Event Management for **additional fee.**
  - ❖ Case Severity / response times:
    - ✓ **General guidance:** < 24 business hours
    - ✓ **System impaired:** < 12 business hours
    - ✓ **Production system impaired:** < 4 hours
    - ✓ **Production system down:** < 1 hour
- **Enterprise On-Ramp Support Plan (24/7):** - Intended to be used if you have **production or business critical workload**
  - ❖ All of business support plan +
  - ❖ Access to a poll Technical Account Manager (TAM)
  - ❖ Concierge Support Team (for billing and account best practices)
  - ❖ Infrastructure Event Management, Well-Architected & Operation Reviews
  - ❖ Case severity / Response times:
    - ✓ **Production system impaired < 4 hours**
    - ✓ **Production system down < 1 hour**
    - ✓ **Business critical system down < 30 minutes**
- **Enterprise Support Plan (24/7):** - Intended to be used if you have mission **critical workload.**
  - ❖ All of business support plan +

- ❖ Access to a **designated** Technical Account Manager (TAM)
- ❖ Concierge Support Team (for billing and account best practices)
- ❖ Infrastructure Event Management, Well-Architected & Operation Reviews
- ❖ Case severity / Response times:
  - ✓ **Production system impaired < 4 hours**
  - ✓ **Production system down < 1 hour**
  - ✓ **Business critical system down < 15 minutes**

### Advanced Identity

- **Service Token Service (STS):** - Enable to you create **temporary, Limited-privileges credentials** to access your AWS resources. Short-term credentials you can configured expiration period.
  - ❖ **Use Case:** -
    - ✓ **Identity Federation:** Manage the user identities in external systems and provide them with STS tokens to access AWS resources.
    - ✓ **IAM** roles for cross/same account access.
    - ✓ **IAM** roles for Amazon EC2 provide temporary credentials for EC2 instances to access AWS resources.
- **Amazon Cognito (Simplified):** - Its provide the Identity for Web and Mobile applications users (potentially millions). Instead of creating them an IAM user, you create a user in Cognito.
- **Microsoft Active Directory:** - On windows server we have installed the AD domain services. Database of object: - User, Accounts, Computers, Printers, File Share, Security Groups. Centralized security management, create account, assign permissions.
- **AWS Directory Services:** - In AWS have 3 types of directory services.
  - ❖ **AWS Managed Microsoft AD:** - Create your own AD in AWS, manage users locally, support MFA, Establish "trust" connections with your on-premises AD, supports MFA.
  - ❖ **AD Connector:** - Directory Gateway (proxy) to redirect on-premises AD, support MFA, Users are managed on the on-premises AD
  - ❖ **Simple AD:** - AD-compatible managed directory on AWS, Cannot be joined with on-premises AD.
- **IAM Identity Center (Successor to AWS Single Sign On):** - One login (Single Sign-On) for all you.
  - ✓ AWS account in AWS organization.
  - ✓ Business cloud application (e.g., Salesforce, Box, Microsoft 365).
  - ✓ SAML2.0-enabled applications.
  - ✓ EC2 Windows Instances.
- **Identity Providers:** -
  - ✓ Built-in identity store in IAM identity center
  - ✓ 3<sup>rd</sup> party: Active Directory (AD), OneLogin, Okta.
- **AppStream 2.0:** - It's desktop application steaming service, so it's from very different from WorkSpace. Deliver to any computer, without acquiring, provisioning infrastructure. **The application is delivered from within a web browser.**
  - ❖ **Workspaces:** - Fully managed VDI and desktop available, The users connect to the VDI and open native or WAM application. Workspaces are on-demand or always on.
  - ❖ **AppStream 2.0:** - Stream a desktop application to web browsers (no need to connect to a VDI)., Works with any device (that has a web browser)., Allow to configure an instances type per application type (CPU, RAM, GPU)
- **Amazon Sumerian:** - Create and run virtual reality (VR), augmented reality (AR), and 3D applications. Can be used to quickly **Create 3D models with animations**. Ready-to-use templates and assets – no programming or 3D expertise required. Accessible via a web-browser URLs or on popular hardware for AR/VR.
- **IoT Core:** - This is network of internet connected devices that are able to collect and transfer data. IoT core allow you to easily connect IoT devices to the AWS Cloud. Serverless , secure & scalable to billions of devices and trillions of messages. Integrated with lots of AWS services (Lambda, S3, SageMaker, etc). Build IoT applications that gather, process, analyze, and act on data.

- **Elastic Transcoder:** - Elastic transcoder is used to convert media files stored in S3 into media files in the format required by consumer playback device (phone etc).
  - ❖ **Benefits:** -
    - ✓ **Easy to use**
    - ✓ **Highly available – Can handle large volumes of media files and large file sizes**
    - ✓ **Cost effective – duration – based pricing model**
    - ✓ **Fully managed & secure, pay for what you use**
- **AWS AppSync:** - Store and sync data across and web apps in real-time.
  - ✓ **(Makes use for GraphQL (Mobile technology from Facebook)).**
  - ✓ Client Code can be generated automatically.
  - ✓ Integration with DynamoDB / Lambda.
  - ✓ Real-time subscriptions.
  - ✓ Offline data synchronization (Replace Cognito sync).
  - ✓ File generated security.
  - ✓ AWS amplify can leverage AWS AppSync in the background.
- **AWS Amplify:** - A set of tools and services that helps you to develop and deploy scalable full stack web and mobile applications. Authentication, storage API (REST, GraphQL), CI/CD from AWS, GitHub, etc.
- **Device Farm:** - Fully-managed service that tests your web and mobile apps against desktop browsers, real mobile device, and tables.
  - ✓ Run tests concurrently on multiple devices (speed up execution)
  - ✓ Ability to configure device settings (GPS, language, Wi-Fi, Bluetooth)
- **Disaster Recovery Strategies:** - (1) Backup and Restore – This charge minimum cost.
  - ✓ **(2) Pilot Light** – This is little-bit costly compared to Backup and Restore.
  - ✓ **(3) Warm Standby** – This is bit higher – compared to Pilot Light
  - ✓ **(4) Multi-Site / Hot-Site** – This is most expensive
- **AWS Elastic Disaster Recovery (DRS):** - Before this name is called “CloudEndure Disaster Recovery” after acquire by AWS its rename it by “Elastic Disaster Recovery”. Quick and easily recover your physical, virtual and cloud-based server into AWS. Example: Protect your most critical database (including Oracle, MySQL and SQL Server), enterprise apps (SAP), protect your data from ransomware attacks. Continuous block-level replication for your servers.
- **AWS DataSync:** - Move large amount of data from on-premises to AWS.
  - ✓ Can synchronize to: Amazon S3 (any storage classes – including Glacier), Amazon EFS, Amazon FSx for windows.
  - ✓ Replication tasks can be scheduled hourly, daily, weekly.
  - ✓ The replication tasks are **incremental** after the first full load.
- **AWS Application Discovery Service:** - Plan migration projects by gathering information about on-premises data centers.
  - ✓ Server utilization data and dependency mapping are important for migrations
  - ❖ **Agentless Discovery (AWS Agentless Discovery Connector):** - VM inventory, configuration, and performance history such as CPU, memory , and disk usage.
  - ❖ **Agent-based Discovery (AWS Application Discovery Agent):** - System configuration, system performance, running processes, and details of the network connection between systems.
  - ❖ Resulting data can be viewed within AWS Migration Hub.
  - ❖ **AWS Application Migration Service (MGN):** - The “AWS evolution” of CloudEndure Migration, replacing AWS Server Migration Service (SMS).
    - ✓ Lift-and-shift (rehost) solution which simplify migration application to AWS
    - ✓ Converts your physical, virtual, and cloud-based servers to run natively on AWS
    - ✓ Support wide range of platform, Operating System, and databases
    - ✓ Minimal downtime, reduced costs
- **AWS Fault Injection Simulator (FIS):** - A fully managed service for running fault injection experiment on AWS workloads. We can basically monitor the was there any performance issue, was there observability issues or resiliency

issues. And you can improve your application to see where the bottlenecks are okay. So basically FIS is advance type monitoring and debugging service.

- ✓ **Based on Chaos Engineering** – Stressing an application by creating disruptive events (e.g., sudden increase in CPU or memory) and want to see how your entire application stack reacts to these disasters. This is why called chaos engineering because you are creating chaos engineering within your infrastructure. Well to make sure our application is really solid, observing how the system responds and implementing improvements.
  - ✓ Helps you uncover hidden bugs and performance bottlenecks
  - ✓ Supports the following AWS services: EC2, ECS, EKS, RDS...
  - ✓ Use pre-build template that generate the desired disruptions
- **Step Functions:** - Step function is the way to build a serverless visual workflow to perform orchestrate, and it's usually of your Lambda functions. So you define the Graph, in case of success or failure, what goes on next. So step function have some internal features, such as sequencing, parallel, conditions, timeouts, error handling so much. And it doesn't just do Lambda functions. It is actually integrate with EC2 instances, ECS tasks, On-premises servers, API Gateway, SQS queues, etc. It also possible to implementing human approval feature.
- ❖ **Use Case:** - Order fulfilment, data processing, web application, any workflow that is complex and describe and that would require some sort of graph for you to visualize.
- **AWS Ground Station:** - Fully managed service that let you control satellite communications, process data, and scale your satellite operations, but it is not for everyone but you have satellites running around the Earth, a lot of them, and you may want to get access to their data for whatever reason. So Ground station provide you a global network of satellite ground stations near AWS regions. So that it's easier for you to get data from your satellites onto your AWS cloud. Allow you to download satellite data to your AWS VPC within seconds, download the data to amazon S3 buckets, or your EC2 instances, and from that you can process it the way you want.
- ❖ **Use Cases:** - Weather forecasting, surface images, communications, video broadcasts.

### AWS Architecting & Ecosystem Pillars

- **Well Architected Framework 6 Pillars:** -
- 1) Operational Excellence
  - 2) Security
  - 3) Reliability
  - 4) Performance Efficiency
  - 5) Cost Optimization
  - 6) Sustainability
- **Operational Excellence:** - Includes the ability to run and monitor systems to deliver business value and to continually improve supporting processes and procedures it's about having operation are great.
- ❖ **Design Principles:** -
    - ✓ **Perform operations as code:** - Infrastructure as code. Infrastructure as a code is going to be the cornerstone of operational excellence and remember infrastructure as code is CloudFormation
    - ✓ **Annotate documentation** – Automate the creation of annotated documentation after every build and generated after every build of your application
    - ✓ **Make frequent, small, reversible changes:** - So that encase of any failure you can reverse it, if you make huge changes every three months this is not going to go well
    - ✓ **Refine operations procedures frequently:** - So as the team gets more familiar with the operations, refine it, automate more, and make improvements
    - ✓ **Anticipate failure:** - That will always happen, need to learn from all these failures. Failure do happens and they are great, failures is when you actually learn, when I fail, I learn, so you should do the same and this is define principal excellence

🔧 **Prepare** ➔ AWS CloudFormation, AWS Config

🔧 **Operate** ➔ AWS CloudFormation, AWS Config, AWS CloudTrail, Amazon CloudWatch, AWS X-Ray

🔧 **Evolve** ➔ AWS CloudFormation, AWS CodeBuild, AWS CodeCommit, AWS CodeDeploy, AWS CodePipeline

- **Security:** - Security we all of know that what is. But it includes the ability to protect information, system, and assets while delivering business value through risk assessments and mitigation strategies. You are really minimizing a risk over time and you save cost from disasters and you really don't want to have a risk, or a security issue in your company.
  - ❖ **Design Principles:** -
    - ✓ **Implement a strong identity foundation:** - Centralize privilege management and reduce (or even eliminate) reliance on long-term credentials – Principal of least privilege – IAM. So going to be one of these services to help us do that.
    - ✓ **Enable traceability:** - Integrate logs and metrics with systems to automatically respond and take action.
    - ✓ **Apply security at all layers:** - You need to secure every single layer, such as if one fail maybe the next one will take over. So edge network, VPC, subnet, load balancer, every instance you have, the OS, patching it, the application, making sure it's up to date, all these things.
    - ✓ **Automate security best practices:** - Security is not do something manually, it's mostly done well, when it's automated.
    - ✓ **Protect data in transit and at rest:** - Always enable encryption, always do SSL, always use tokenization and do access control.
    - ✓ **Keep people away from data:** - Reduce or eliminate the need to direct access or manual processing of data.
    - ✓ **Prepare for security events:** - So security events must happen some day in every company, I think, and so run response simulations, use tools to automate the speed of detection, investigation and recovery.
- ✚ **Identity and access management** ➔ IAM, AWS-STS, MFA Token, AWS Organization.
- ✚ **Detective controls** ➔ AWS Config, AWS CloudTrail, Amazon CloudWatch
- ✚ **Infrastructure protection** ➔ Amazon CloudFront, Amazon VPC, AWS Shield, AWS AFW, Amazon Inspector
- ✚ **Data protection** ➔ KMS, S3, ELB, Amazon EBS, Amazon RDS
- ✚ **Incident response** ➔ IAM, AWS CloudFormation, Amazon CloudWatch Events
- **Reliability:** - Reliability is ability of a system to recover from infrastructure or service disruptions, dynamically acquire computing resources to meet demand, and mitigate disruption such as misconfiguration and transient network issue so about that make sure your application runs on matter.
  - ❖ **Design Principles:** -
    - ✓ **Test recovery procedure:** - Use automation to simulate different failures or to recreate scenarios that lead to failures before
    - ✓ **Automatically recover from failure:** - Anticipate and remediate failures before they occur
    - ✓ **Scale horizontal to increase aggregate system availability:** - Distribute request across multiple, smaller resources to ensure that they don't share a common point of failure
    - ✓ **Stop guessing capacity:** - Basically if you think oh need four streams in this for my application that probably isn't going to work in long term. Use autoscaling wherever you can to make sure you have the right capacity at any time
    - ✓ **Manage change in automation:** - In terms of automation you need to basically change everything through automation and this is ensure that your application will be reliable or you can roll back, or whatever
- ✚ **Foundation** ➔ IAM, Amazon VPC, Service Limits, AWS Trusted Advisor
- ✚ **Change Management** ➔ AWS Auto Scaling, Amazon CloudWatch, AWS CloudTrail, AWS Config
- ✚ **Failure Management** ➔ Backups, AWS CloudFormation, Amazon S3, Amazon S3 Glacier, Amazon Route 53
- **Performance Efficiency:** - Includes the ability to use computing resources efficiently to meet system requirements, and to maintain that efficiency as demand changes and technologies develop.
  - ❖ **Design Principles:** -



- ✓ **Democratize advanced technologies:** - As basically advanced technologies become services and hence you can focus more on product development
- ✓ **Go global in minutes:** - Easy deployment in multiple regions
- ✓ **Use Serverless architectures:** - Avoid burden of managing servers that means you don't managing servers and everything scales for you, which is really awesome
- ✓ **Experiment more often:** - Easy to carry out comparative testing
- ✓ **Mechanical sympathy:** - Be aware of all AWS services, and that's really, really hard.

✚ **Selection → AWS Auto Scaling, AWS Lambda, EBS, S3, RDS**

✚ **Review → AWS CloudFormation**

✚ **Monitoring → Amazon CloudWatch, AWS Lambda**

✚ **Tradeoffs → RDS, Amazon ElastiCache, Snowball**

➤ **Cost Optimization:** - Include the ability to run system to deliver business value at the lowest price point.

❖ **Define Principles:** -

- ✓ **Adopt a consumption mode:** - Pay only for what you use
- ✓ **Measure overall efficiency:** - Use CloudWatch
- ✓ **Stop spending money on data center operations:** - AWS does the infrastructure part and enable customer to focus on organization project.
- ✓ **Analyze and attribute expenditure:** - Accurate identification of system usage and cost, help measure return on investment (ROI) – Make sure to use tags (means if you don't use tags on your AWS resources so you are going have a lot of trouble figuring out which application is costing you a lot of money.
- ✓ **Use managed and application-level service to reduce cost of ownership:** - So means manage service operate at cloud scale, they can offer a lower cost per transaction services operate at cloud scale, they can offer a lower cost per transaction or service.

✚ **Expenditure Awareness → AWS Budgets, AWS Cloud and Usage Report, AWS Cost Explorer, Reserved Instance Reporting.**

✚ **Cost-Effective Resources → Spot Instance, Reserved Instance, Amazon S3 Glacier**

✚ **Matching Supply and demand → AWS Auto Scaling, AWS Lambda**

✚ **Optimizing Over Time → AWS Trusted Advisor, AWS Cost and Usage Report, AWS News Blog**

➤ **Sustainability:** - The sustainability pillar focuses on minimizing impacts of running cloud workloads.

❖ **Design Principal:** -

- ✓ **Understand your impact** – Establish performance indicators, evaluate improvements
- ✓ **Establish sustainability goals** – Set long-term goals for each workloads, model return on investment (ROI)
- ✓ **Maximize utilization** – Right size each workload to maximize the energy efficiency of the underlying hardware and minimize idle resources.
- ✓ **Anticipate and adopt new, more efficient hardware and software offerings** – (Design for flexibility to adopt new technologies over time) AWS does some hardware optimizations to their infrastructure and if you use their newer stuff.
- ✓ **Use managed services** – Shared service reduce the amount of infrastructure; Managed service help automate sustainability best practice as moving infrequent access data to cold storage and adjust compute capacity.
- ✓ **Reduce the downstream impact of your cloud workloads** – Reduce the amount of energy or resources required to use your services and reduce the need for your customers to upgrade their devices.

✚ **EC2 Auto Scaling, Serverless offering (Lambda, Fargate)**

✚ **Cost Explorer, AWS Graviton 2, EC2 T Instances, @Spot Instances**

✚ **EFS-IA, Amazon S3 Glacier, EBS Cold HDD volumes**

✚ **S3 Lifecycle Configuration, S3 Intelligent Tiering**

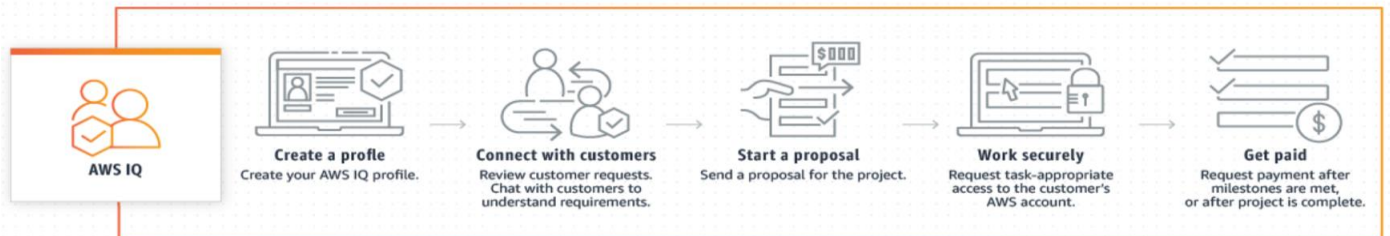
- 🚩 **Amazon Data Lifecycle Manager**
- 🚩 **Read Local, Write Global: RDS Read Replicas, Aurora Global DB, DynamoDB Global Table, CloudFront**

- **Well-Architected Tool:** - Free tool to review your architected against 6 pillars Well-Architected Framework and adopt architectural best practices.
  - ❖ **How does work:** - Select your workload and answer question., Review your answers against the 6 pillars., Obtain advice: Get videos and documentations, generate a report, see the results in a dashboard.
- **Right Sizing:** - EC2 has many instances type but choosing the most powerful instances type isn't the best choice, because the cloud is elastic.
  - ✓ Right sizing is the process of matching instance types and sizes to your workload performance and capacity requirements at the lowest possible cost.
  - ✓ **Scaling up is easy so always start small**
  - ✓ It's also the process of looking at deployed instances and identifying opportunities to eliminate or downsize without compromising capacity or other requirement, which result in lower costs
  - ✓ It's important to Right Size...
    - Before a Cloud Migration
    - Continuously after the cloud onboarding process (requirements changes over time)
  - ✓ **CloudWatch, Cost Explorer, Trusted Advisor, 3<sup>rd</sup> party tools can help.**
- **Whitepapers (Guides):** - Whitepaper it's well architected framework is one of the white paper. They are usually very long guides that it was did approved or did right and they help you understand how to do something correctly. For example – security, around architecture, networking.
- **Quick Start:** - This is automated gold-standard deployments in the AWS cloud., Allow you to build your production environment quickly with templates for example have **WordPress** on AWS you will get the option to get a confirmation template deployed in a specific region. Then you will be able to get your entire WordPress deployments.
- **AWS Marketplace:** - Digital Catalog with thousands of software listings from **independent software vendors (3<sup>rd</sup> party)**
  - ❖ **Example:** -
    - ✓ Custom AMI (custom OS, firewalls, technical solutions...)
    - ✓ CloudFormation templates
    - ✓ Software as a Service
    - ✓ Containers
  - ❖ If you buy through the **AWS Marketplace**, it goes into your AWS bill
  - ❖ You can **Sell Your own solutions on the AWS Marketplace**
- **AWS Training:** - AWS Digital (online) and classroom training (in-person or virtual)., AWS Private training (for your organization)., Training and certification for the U.S Government., Training and Certificate for the Enterprise., AWS Academy: Helps universities teach AWS., And your favourite online teacher like Udemy, YouTube etc.
- **AWS Professional Service & Partner Network:** - It is a global team of experts. The work alongside your team and a chosen of the APN :- AWS Partner Network.
  - ✓ **APN Technology Partners:** - Providing Hardware, connectivity, and software
  - ✓ **APN Consulting Partners:** - Professional Services firm to help build on AWS
  - ✓ **APN Training Partners:** - Find who can help learn AWS
  - ✓ **AWS Competency Program:** - AWS Competencies are granted to APN Partners who have demonstrated technical proficiency and proven customer success in specialized solution areas.
  - ✓ **AWS Navigate Program:** - Help Partners become better partners.
- **AWS IQ:** - Help you to quickly find help for your AWS projects. AWS IQ is kind of freelance work base. Engage and pay AWS Certified 3<sup>rd</sup> party experts for on-demand project work.

## For Customers



## For Experts



- **AWS re:Post:** - It's a community forum in which you find answers, you answer question, you will find best practices or you can join groups. So it is AWS managed Q&A Service that is offering crowd-sourced expert reviewed answers to your technical questions about AWS and replaces what used to be the original AWS forums. (For example on Stack Overflow someone ask the question and then you get answers, someone of them can be up voted of them can be accepted and they are reviewed by experts all the time.) This is part of AWS free tier. **If you are a premium customer on AWS and you do not receive a response from the community, then automatically your question is passed to AWS support engineers and they will answer your question.**

<https://aws.amazon.com/certification/certified-cloud-practitioner/>

- In this exam will ask 65 question in 90 minutes. **Once you submit your exam you get popup for passed or fail your exam. You will know the overall score a few days later (email notification). To pass you need a score if at least 700 out of 1000. If you fail, you can retake the exam again 14 days later.**

.pem format key we can use for linux, mac and windows

.ppk format key we can use for putty on windows 10,11,7

Recycle-Bin protect your Amazon EBS snapshots and Amazon Machine Images (AMIs) from accidental deletions.

**Server-Less:** - AWS Athena, AWS Lambda, Amazon S3, Forget (its like container), AWS DynamoDB, Amazon QuickSight, AWS Glue,

**PaaS Services:** - Elastic Beanstalk is PaaS services,

### Global Application in AWS

**Global DNS: Route-53:** - (1) Route users to the closest deployment with least latency., (2) Route for disaster recovery strategies.

**Global Content Delivery Network (CDN): CloudFront:** - (1) Replicate part of your application to AWS Edge Locations – decrease latency., (2) Cache common request – improved user experience and decrease latency.

**S3 Transfer Acceleration:** - Accelerate global uploads & downloads into Amazon S3.

**AWS Global Accelerator:** - Improve global application availability and performance using the AWS global network.

### Free Services and Free tier in AWS

IAM, VPC, Consolidated Billing, Elastic Beanstalk, CloudFormation, Auto Scaling Groups, EC2 t2.micro instances for a year, S3, EBS, ELB, AWS Data transfer.

**Trusted Advisor Category:** - There are 5 categories= 1) Cost optimization, 2) Performance, 3) Security, 4) Fault Tolerance, 5) Service Limits