

AWS I.

Cloud Practitioner Essentials

Aws offers compute, storage and network security services, leveraging blockchain ML and AI.

In aws a server side application is called ^{instance} ↑
amazon elastic compute cloud (Amazon EC2)

Instances of services are dynamic, can add/
remove them depending on the needs.

Client is a web browser or an application, while
server is a service such as (virtual server)

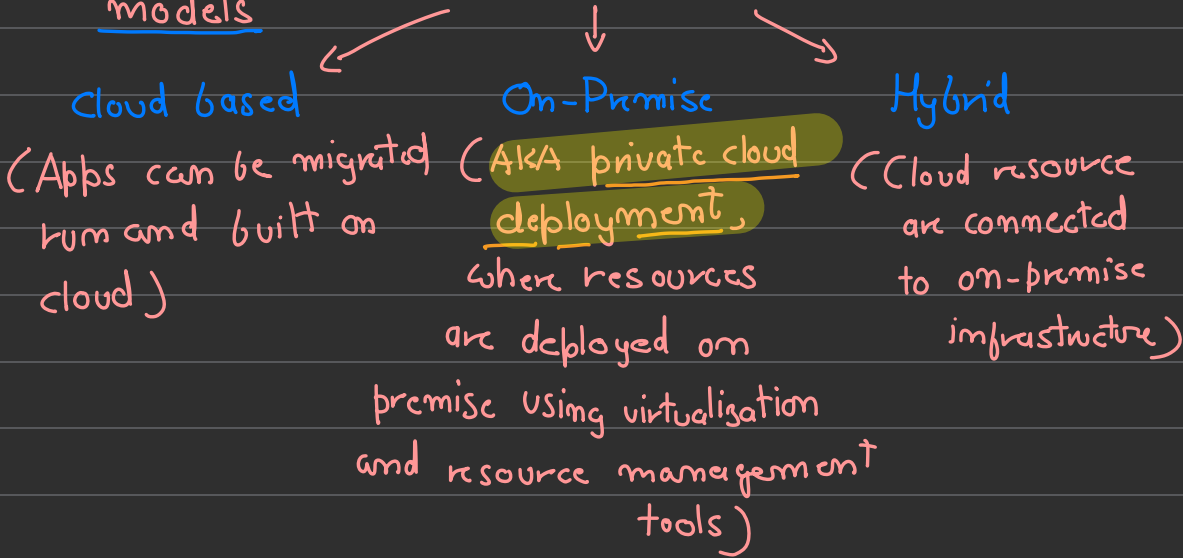
IT service

Cloud Computing is on demand ^{IT service} delivery over the internet.

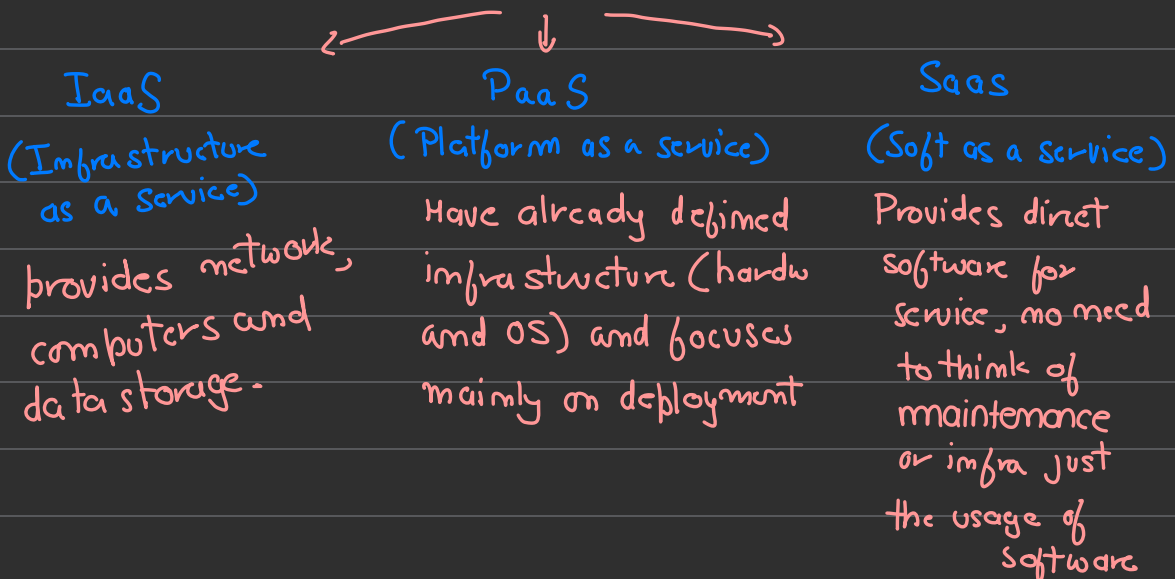
Aws works on Compute as a service (CAAS) model.

IT

There are 3 main cloud computing deployment models



We also have 3 cloud computing models





Compute

Amazon EC2

AKA amazon elastic compute cloud. or 'EC2'.

In aws the servers are virtual - servers used to gain access to those virtual servers is EC2.

Aws already built and secured datacenters, purchased servers, installed em and are online - ready to use.

Multitenancy - sharing underlying hardware between virtual machines. (done by hypervisor)

(ie EC2

resource is

Secure because

one VM doesn't know about each other while still working together)



(helps isolate virtual machines while they share resource from host server.)

Vertical Scaling - Instance can be bigger or smaller based on the needs.

Horizontal Scaling - when we multiply instances.

While configuring an EC2 instance one can configure all of the below

→ OS : like linux or windows

→ Software : Web apps, Database, 3rd party
Software
etc.

EC2 are resizable, vertically as well as horizontally.

↙
giving
more memory
and CPU

Amazon Machine Images (AMIs) is a pre configured virtual machine image that contains all OS, software and other required components needed to launch an instance.

↓
can be used to create instances in same/
diff regions and helps in disaster management.
(pre configured VM template for EC2)

There are diff types of EC2 instances, and each type of instance belongs to a instance family

```
graph TD; A[instance family] --> B[gen purpose]; A --> C[compute optimized]; A --> D[memory optimized]; A --> E[Accelerated computing]; A --> F[Storage optimized]
```

- Gen purpose have balance between all aspects
- Compute optimized is for apps needing high performance processors eg. gaming, scientific modelling.
- Memory Optimized for memory intensive tasks
- Acc Computing for decimal calculation, graphic processing or data-pattern matching - all apps which require hardware accelerators.
- Storage Opt. is for locally stored data.

EC2 also have diff billing/pricing.

```
graph TD; A[EC2 also have diff billing/pricing.] --> B[on-demand]; A --> C[savings plan]; A --> D[Reserved instance]; A --> E[Spot instance]; A --> F[Dedicated]; C --> C1["(commitment on usage for 1-3 years)"]; D --> D1["(for predictable usage)"]; D --> D2["75% discount"]; E --> E1["(90% off price, but instance can be reclaimed)"]; F --> F1["(no sharing, only for you)"]
```

- on-demand**
- savings plan**
(commitment on usage for 1-3 years)
- Reserved instance**
(for predictable usage)
75% discount
- Spot instance**
(90% off price, but instance can be reclaimed)
- Dedicated**
(no sharing, only for you)

→ Scaling and Elasticity :

For automatic scaling of EC2 instance, we use Amazon EC2 auto scaling. (does automatic horizontal scaling)
↓

It automatically adds/removes instance based on demand. In EC2 auto scaling we have 2 options

Dynamic

(responds to change in demand)

Predictive

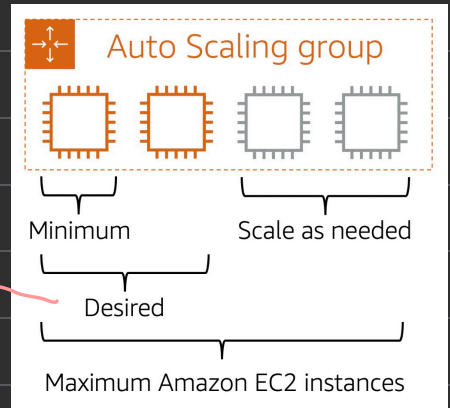
(Schedules instance based on prediction)

(can be used together)

Min capacity is no. of instance which run on the auto scaling group.

Max capacity is also supposed to be configured.

have to specify it otherwise aws makes it equal to minimum.



Even distribution and routing of requests (load - balancing) is done using elastic load balancing (ELB)

Elastic Load Balancer

→ (Even dist. of routing of requests)

It could be either set in the org or can be set using AWS's Load Balancer. (its a managed service and a regional construct)

↳ ELB is auto scalable (without more cost).

helps the decoupled architecture and can be used for

external EC2 requests and internal as well (ie between the application's frontend and backend).

(Distributes incoming request over diff instance in a region)

Buffer messaging and queuing is used by applicatⁿ when loosely coupled to communicate.

For this we have 2 aws service

across multiple AZs

Amazon simple Queue service
(amazon SQS)

amazon simple notification service
(amazon SNS)

(ELB is a regional construct and does not work on an instance level)

→ SQS allows to send store and receive messages between **software component** without losing messages.

→ SNS is used to send messages but it can also publish message to end users too.

SNS and SQS allows applications to be more independent.

-----> **can be**
web servers,
lambda functⁿs
etc.

In **monolithic** applications, all the components are housed in the same application bubble → which leads to **tight coupling** - in this if single component fails, the entire application suffers/fails.

While in a **micro service** architecture application's components are **loosely coupled** - they are independent, and if one fails, others (components) are not affected.

AWS Lambda

If the above AWS resources are to be not configured manually, AWS provides **infrastructure** which takes care of all these **internally**, and we just have to worry about the **application**.

The Amazon service which takes care of this is the **AWS Lambda**. It provides a **lambda function** - which is fed the application's code/source, and all we need to do is configure a **trigger**. - more like jobs

↙ **

Lambda runs in < 15 mins and is suitable for quick processing and **not long time taking application like deep learning.**

→ (server management is taken care by underlying infra)

It's a type of serverless architecture which lets you deploy code without provisioning and managing servers

Compute Services

EC2 instances host simple backend projects, but maintaining them is the role of the owner, if any new packages are rolled out etc.

From a management perspective, AWS provides server less computing which manages everything from setting up instance, ELB etc.

↳ it means though the app is running on the server the owner doesn't need to worry about updating, provisioning or managing these instances manually. - AWS Lambda

If we still are looking for AWS products but want to handle provisioning ourselves, AWS got service like Amazon elastic container service (Amazon ECS)

Amazon ECS :

AKA **elastic container service** is a container management system enabling to run and scale containerized application in AWS.

Its called **container orchestration tool**. These containers **run on EC2 instances** but run in isolation (with each other) — similar to how virtual machine works.

Number of EC2 instances working together are called **a cluster**. Now if multiple EC2 instances are running containers, these containers are supposed to be start, stop, restart and be monitored upon together over the **cluster**. And to orchestrate these together, is a big process, so to manage it we use **ECS**.

Amazon EKS :

AKA elastic kubernetes service , is just like amazon ECS but uses diff tooling and features.

Both ECS and EKS run on top of EC2 but the need of configuring the instances makes path for what service is to be chosen . These define rule for how is the ECS or EKS are managed .

1. EC2 launch type → when we need full control of customizing the usage and features of EC2 instance

2. Fargate launch type → when we do not need to think of manual configuration and we want to let AWS handle it for us - like configuring ELB, SQS, SNS etc .
Cfor this - internally AWS uses the infrastructure layer to manage instance - just like AWS lambda .

though Fargate and Lambda have similar application, their usecase is totally different — while Fargate is designed for containerized application

↓
Lambda is used for event-driven serverless compute.

(takes care of all the orchestration of containers.)

While Lambda controls the number of instances and controls scaling, scaling in case of Fargate is done according to the EC2 instance configuration.

↓
can use EKS or ECS. So,

what EC2 is to Lambda
similarly, EKS/ECS is to Fargate

→ kind of



Global infra.

Global Infrastructure :

Helps with fault tolerance and continuous -
- availability. ^{and high}
↗

Amazon have several **data centers**, not just one because if they had only 1, any mishap can lead to obstruction in using the server and related applications.

In the past organizations used to have their own data centers but it would require lots of config and costs.

AWS have divided their datacenters into groups by location - called 'regions' → location like

All regions are connected

via high speed fibre network

maintained by AWS.

paris

tokyo

dublin

ohio

sao paulo

As a customer, it's a choice to which region we want our application to be deployed on. While on the other hand each region is isolated from all the other region.

(unless explicitly someone asks for data sharing)

→ like govt. compliance etc.

To choose a region there are some main points to take into consideration,

1. Compliance - requirement based on region, working under some country's borders.
2. Proximity - How close one is to their customer base. (because over regions, latency matters)
3. Feature Availability - Some services might not be available in certain regions

4. Pricing - Some locations are comparatively more expensive to maintain and run.

These would be the 4 key features before selecting a region for AWS.

AWS also have multiple data center groups within a certain region, and is called an **availability zone**.

An **availability zone (AZ)** is one or more set of data centers with redundant power, networking and connectivity.

Each center in an AZ is isolate and physically separate from the others to prevent certain geological mishaps to ruin the whole **AZ** altogether.

*
Am **availability zone (AZ)** is a fully isolated portion of the AWS global infrastructure.

*
* Built on - Design of Failure - Cloud Architecture ^{Design} concept

Best part is to use multiple instances (between different availability zones (in the same region)) and its recommended to atleast use 2 instance over 2 different AZ in the same region.

Regional Construct is the AWS services which work synchronously with all the AZs in a particular region.

eg. ELB - also called as a regionally scoped service.

Regions are mostly spread across the globe and are easy to access but what if a certain place on the globe is quite far from the nearest region.

That certain place is added to a network of similar places called edge locations.

Availability Zones are best for fault tolerance, but for

Disaster management → AWS Regions.

Edge locations are sites where the reach to the nearest region is pretty far and what an edge location have in unique is called the amazon cloudfront. ↗ on the globe

Amazon Cloudfront is a **shared network** AKA as **a global CDN (content delivery network)** that delivers data, videos, application to users with low latency and transfer speeds

How?

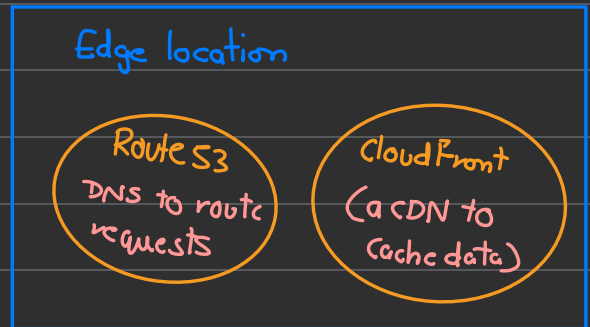
It **cache's / copies** content over certain **edge locations** over the globe.

It also aids by lowering Bandwidth cost and load on the actual servers.

(Its not physical like datacenter, it works more like a network and all the data is shared via that CDN)

And using cloudfront instead of making an instance is way cheaper (but limits applications)

Edge locations doesn't just run CDN network but it also runs DNS (domain name service) of amazon called as Amazon Route 53.



AWS Outposts is a managed AWS service which allows business to run AWS services in their own datacenters. More like installing and manage AWS hardware in once own on-premise environment — eg. a corporate private (outpost is an edge device) data center, factory, hospital.

* AWS outpost is when an org requires AWS to setup their physical servers onto their own premises.

How to provision AWS Services :

AWS services like creating a EC2 instance or making a lambda function, everything is done over API calls in AWS.

For services one can use AWS management console, or AWS's CLI (command line interface) or AWS SDKs (Software dev kits) or other tools to create request for APIs.

Console - good for initial setup, test env., to view AWS bills, monitor services or to work with non-technical resources.

CLI - better for automation, scripting which makes it less susceptible to human error.

VIP (more production type environment)

SDK - Allows to use AWS resources over diff applications built on diff languages.

AWS Elastic Beanstalk

(scaling, network, ELB config)

It's a manage tool for amazon EC2 based env.

This service takes the application code and required configuration for the EC2 instance and creates the services for us and helps saving configuration so that future deployments are easy.

* → databases too!

It automatically handles deployment → capacity provisioning, load balancing, auto scaling and health monitoring.

AWS CloudFormation :

uses cloud dev kit (AWS CDK)

It's an infrastructure as a code tool used to define wide variety of AWS resources using cloud formation templates → JSON or YAML.

→ (not just EC2)

CloudFormation takes care of calling API by parsing the template and creates apps for storage, data, analytics etc and provisions all of them in parallel.



Networking

only used to configure EC2

config. so that we don't have to do that again.

- Elastic Beanstalk is also helpful because it saves emv
- For a whole emv cloudFormation templates could be used in diff region to setup whole infra (unlike 'beanstalk')

↓
like storage, server, network

Virtual Private Cloud

Abbreviated as **amazon VPC** is a service which lets user provision logically isolated section of the cloud when we can use other AWS resources/services **on the virtual network that the user defines.**

→ ones with access to internet
These can be public or private. facing.

For backend services like database, application the public and private grouping is also known as **Subnets.**

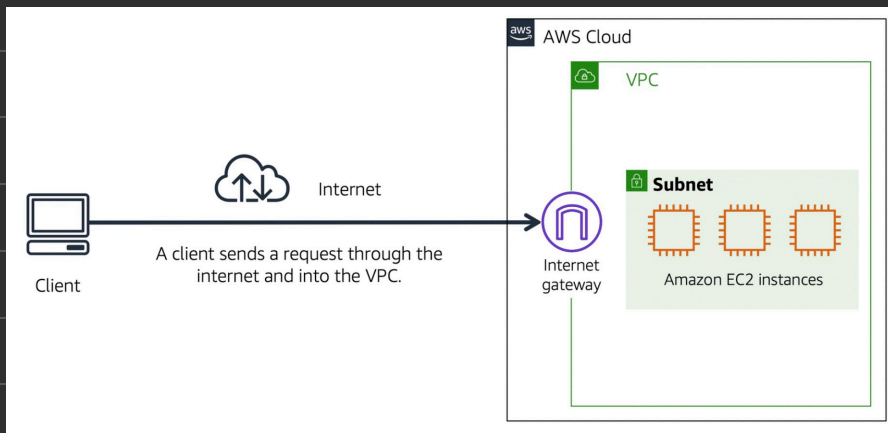
There are diff set of IP address available in a **VPC**. - can be configured for applications inside the infrastructure to use.

VPC is like a private network in AWS. Allows to define private IP ranges for the resources.

And the services like EC2 or ELB are placed inside of VPC.

→ (or IGW)

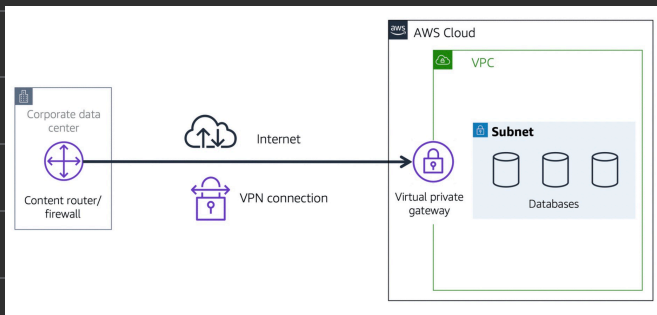
VPC uses **internet gateway** to let the public traffic into the public facing resources.



VPC can span over all availability zones within a AWS Region.

For a VPC with private resources, have a **virtual private gateway** which allows to create a VPN connection between private network (like office's private network) to our VPC.

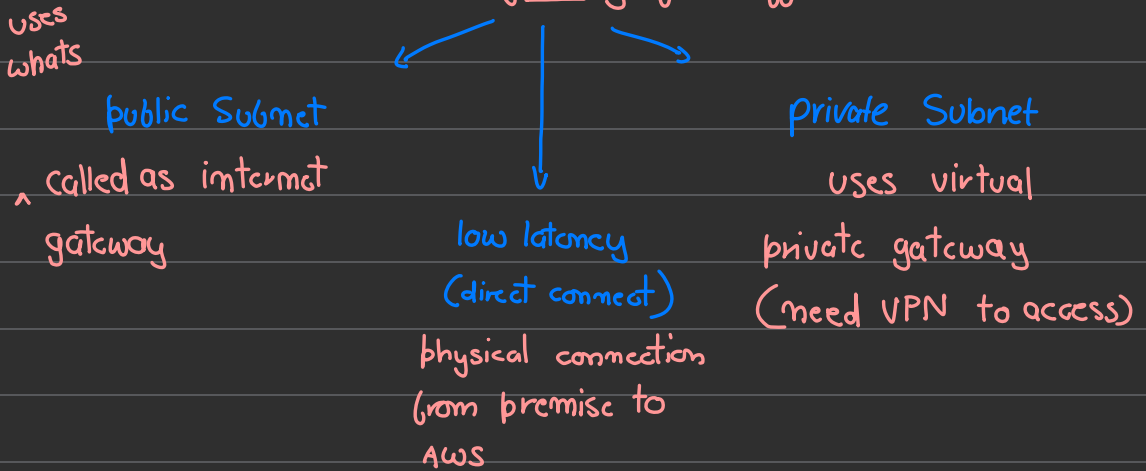
If we wanna establish encrypted VPN connectⁿ to private internal AWS resource we are supposed to attach **VPG** to our VPC.



VPN is private and encrypted but still network traffic could be a problem.

Just like **VPG**, **customer gateway** is the on-premise side of **VPN connection**. Its responsible to establish connection to **VPC** from on-premise network.

Virtual Private Cloud (VPC)



- 1 aws account = multiple VPC
 - 1 VPC = multiple gateway (public, private)
 - 1 VPC = multiple subnets
 - 1 VPC = multiple resource (EC2, S3 etc)
 - 1 subnet = multiple resource (" , " etc)
 - 1 subnet = 1 gateway (attached)
- (highly Scalable cloud router.)

AWS Transit Gateway connects VPCs to on-premise networks through **central hub**.

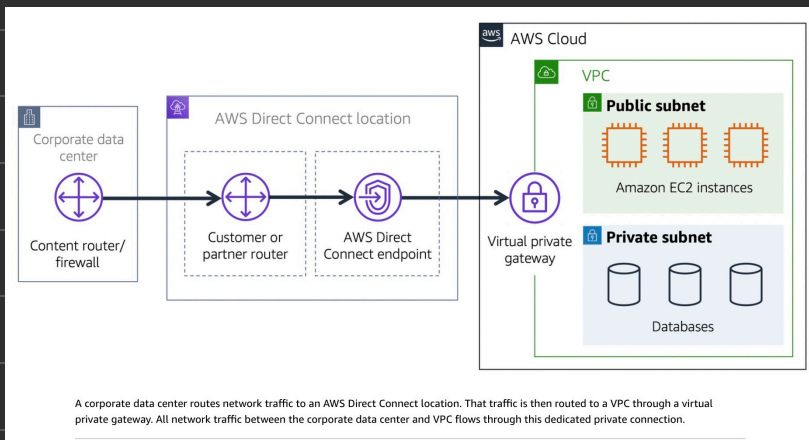
↓

* *

If we still want a private connection and don't want to share it with anyone and lowest latency with highest security can be achieved using AWS direct connect.

Direct Connect allows direct connection from data center to AWS. We use third party direct connect company (because it needs physical connection).

One VPC can have diff gateways attached for multiple resources all in same VPC but diff subnets.



Network Hardening

Each subnet (be it private or public) have a security layer called network access control list. ←

Each packet going inside a Subnet has to pass the network ACL. (checks traffic in and out of the subnet) → Processes rules in order.

Say we have 2 diff EC2 instance in a Subnet with diff config running on each, for this we configure instance level security.

To solve this security group are defined, initially doesn't allow any traffic to come in. We have to explicitly define what traffic is in and by default all packets out.

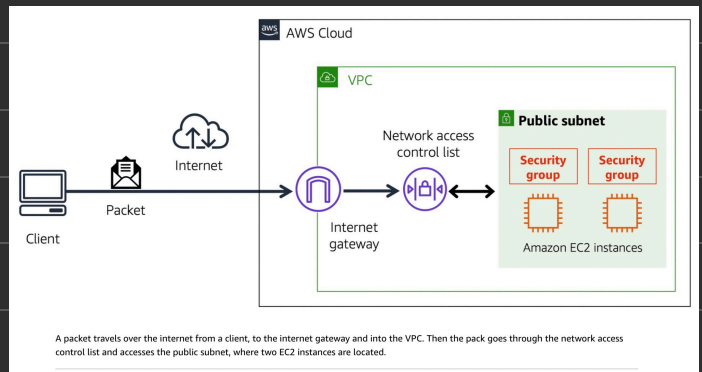
→ have memory (rules)

While security group is stateful, network ACL is stateless

→ checks every package I/Os the subnet.

For a request - response cycle between 2 instances in the same VPC (but diff subnet) the req/response goes through security layer 8 times in a full cycle.

(Both public and private subnets have network ACL)



security group.

On the way back of a packet (a response) the Security doesn't check it again because it remembers the same request due to its stateful nature.

These exchanges happen very quickly and there isn't any delay due to any security configuration

By default network ACL are stateless and allows all inbound and outbound traffics.

request/response
check.

for a security group = Some in, all out

for a network ACL

pre defined (default) ACL

= all in, all out

Custom ACL

= no in, no out

(until list of allowed
traffic is defined)

Network ACL

Handles traffic for subnet

Stateless - remember nothing

Default - all in, all out

Security Group

Handles traffic on particular AWS ^{resource}

Stateful - remember decision for
incoming packets

Default - no in, all out

Global Networking

From a client side's perspective, the access to a website is possible using 2 .

(DNS)

1. Route 53 - its AWS's domain name service highly available & scalable.

It can route/direct traffic to diff endpoints using several routing policies like latency-based, geolocation DNS, geoproximity and weighted round robin. - consider it more like a phone book.
Can also use route 53 to register domain name.

2. CloudFront - uses CDN and edge locations due to poor latency. (refer page 18)

(Its ELB and cloudFront which decides what AZ takes what request)

3. VPC Peering : Its network connection between 2 VPCs . They can be in diff regions or diff accounts too.



Storage

Storage :

Block level storage can also be considered like a physical hard drive.

-----> better for scratch/
temp data.

Instance store volumes are storage physically attached to the host on top of which our resource is running (like EC2). But it's a temp storage - kind of like local storage in any backend application.

Once the instance stops/terminated, the volume is deleted.

Elastic Block Store

(It's just like instance store volumes but are permanent)

Aka amazon's EBS, one can create virtual harddrive also called as EBS volume.

One can configure EBS based on Size, type and configs needed.

↳ can be attached to EC2 instance.

only data modified or newly introduced
→ is going to be backed up.

Snapshots are incremental way of backing up data provided with EBS. If drive corrupts, data can be backed up after.

Simple Storage Service (S3)

AKA amazon S3 is a simple data storage solution which enables to store **unlimited amount** of data.

Data is stored as objects, and are stored in **buckets**. instead of file directory. (unlike EBS)

Max object size — 5TB (single files)

- We can also version data objects if needed.
- permissions can also be set according to bucket. (Write once / read many)

EBS → Snapshots (upto 16 TiB storage, solid state)

S3 → Versioning (unlimited storage, 1 object upto 5TBs)

In an object type storage, an object consists of the Data, MetaData and key → combines to form an object.

Diff datatypes can be divided into diff stage like some logs and historic data can be stored in a kind of tier while other more frequently accessed data can be moved to diff tier.

→ Amazon S3 Standard - comes with 11-9s of durability i.e. it has probability 99.99...% durability (will stay intact after 1 year) → AZs
(data is stored in atleast 3 diff locations)

→ S3 Static website hosting - Collection of HTML file for an actual site. pages
(a whole bucket can be Deployed)
there's an option to deploy website statically like

→ S3 Standard Infrequent Access - AKA S3 Standard IA
(Standard IA)

Used for data which is not frequently needed but require rapid access when needed eg backup, disaster recovery files etc. (infrequent but more regular access)

→ S3 one-zone IA - Stores data in single AZ and has lower price than Standard IA.

Better storage class over standard IA when one has to save cost and data can be reproduced in case of an AZ failure

→ S3 Glacier Instant Retrieval - Best for **archive data** for immediate access and can be retrieved within milliseconds. (long term archiving)

Amazon's S3 lifecycle policies are rules which can be defined by the user to automatically move certain data from one tier to another.

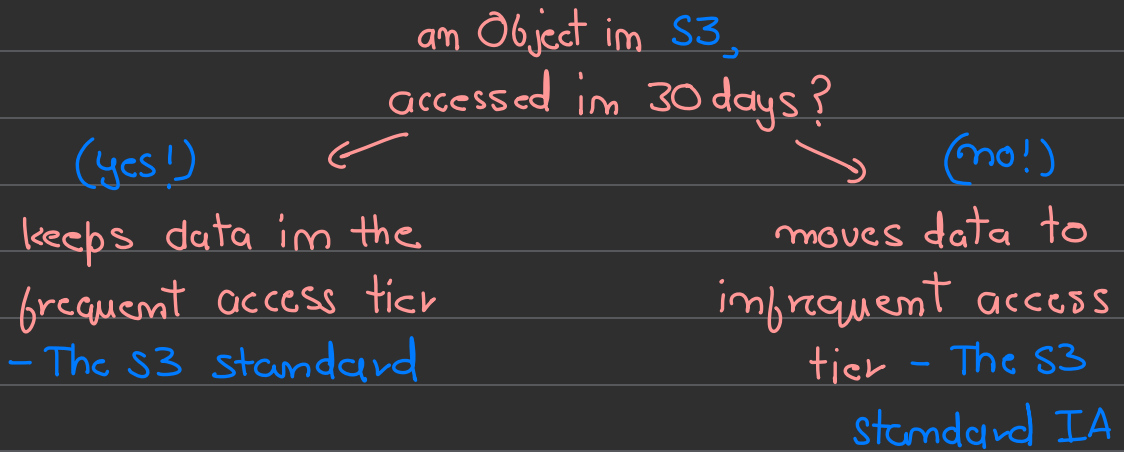
Amazon S3's data can be individually managed and each data object can be accessed by ^{web} their own specific URL. (S3 is serverless) enabled

→ S3 Glacier Flexible Retrieval - Data could be simply moved to this tier or we can create vaults and populate them with archives, and for a time-bound/limit data one can deploy glacier lock policy to lock the vault. (can have time limit on retrieval, certain rules on what is allowed to write and what's not) - Takes minutes to hours to retrieve data.

→ S3 Glacier Deep Archive - lowest cost storage class
→ to 48 ideal for archiving and requires ~ 12 hours time to retrieve objects. Supports long term retention and preservation for data accessed once or twice a year.

→ Amazon Athena is a serverless interactive Query Service that analyses data in S3 using SQL. No need to move data, it reads data from S3 and parses it in CSV/JSON and then queries it.

S3 intelligent Tiering storage class is a type of S3 storage where data/object's access pattern is monitored.



Amazon S3 standard and S3 standard IA keeps data in min 3 availability zones. S3 standard IA provides SAME availability as standard but its cheaper.

Amazon Macie is a data security and data privacy service that uses ML and pattern matching to discover and protect sensitive Data in S3.

While on the other hand **EBS** is a block storage unlike **S3** which is object-storage, to edit one storage entity from both of these, in **S3** you will have to upload the whole file again (usually **S3's data looks like certain image, video, text files etc.**) while on the other hand in **EBS** we don't need to upload the whole file again, we could just do the necessary updation.

EBS is perfect for micro changes kind of data.

Elastic File System

AKA amazon's **EFS** is ^{→ managed/shared} file system, when need a shared file system across applications one can go for **EFS**.

One can have **multiple instances** accessing data from **EFS** at the same time.

Elastic Block Storage

EBS scales up and down automatically. The diff from EBS is that EBS volume are usually attached to instances and are **availability zone level resource** and to attach an EC2 to an EBS both are supposed to be in the **same availability zone**. And EBS cannot scale itself.

Elastic File System

While EFS can have multiple instances reading and writing on it at the same time, it's an **actual Linux based file system** and it's a **regional resource** unlike EBS which is AZ based.

✓ And finally EFS automatically scales.

*

EBS → 1 AZ

EFS → multiple AZ

Relational Database Service

Aka amazon's RDS can have data related to each other. AWS supports MySQL, PostgreSQL, Oracle, Microsoft etc. (also provides online transaction processing)

Amazon's Lift and Shift migration helps migrate relational database to aws for it to work with EC2.

Amazon RDS have automated patching, backup, redundancy, failover and disaster management. which allows to focus on application more.

Amazon Aurora is most managed DB option from amazon, comes in either MySQL or PostgreSQL its cheaper, allows data replication, backs up to S3, can deploy 15 read replicas. → does 6 replicas (use this when require high availability) across 3 AZs by default.

Dynamo DB (non-relational)

→ Serverless

Its a **server less** (no need to configure) kind of DB where we create tables (data can be stored and queried from the table).

It is auto scaled and stored redundantly across multiple availability zones and multiple drives.

It has a millisecond response time - required for a big user base application.

Its a normal DB but does not use SQL, because its not rigid, its a non relational DB. But uses tables tho.

And queries are written based on some small subset of attributes (column names) that are designated as 'keys'.

Non relational, No-SQL, Fully managed and highly scalable. Also data at rest is by default encrypted.

RDS

Automatic high availability and recovery.

Customer ownership of data and schema

Customer control of network

has to be configured & managed.

Dynamo DB

key-value (global DB)

Massive throughput and PB size data

Granular API access

(is serverless)

→ Fully Managed AWS Databases :

Amazon RDS

Amazon DocumentDB

Amazon KeySpaces

Amazon ElastiCache

Amazon Redshift :

For a DB with real time read and write functionality and the volume and type of data for the DB would be too much. (can be used for Big Data Analytics)

Once data becomes too complex to handle with traditional database, is when we use data warehousing. We perform historical analytics instead of operational analysis.

If a Business uses historic data for operation - ^{time series} best practice would be to use data warehousing.

Amazon Redshift is a data warehousing as a service, massively scalable - nodes go upto certain petabyte size is common. (unstructured data running in data lakes)

Single SQL query can work on exabytes of data by cooperating with Amazon Redshift Spectrum.

Redshift offers 10x higher performance than traditional databases.

When we need big data BI Solution, Redshift allows to get started with single API call.

A cluster in Redshift is a set of nodes that work together to manage (store and process) data and execute queries.

A node in Redshift is a individual compute resource within a cluster that stores data and process queries



leader node

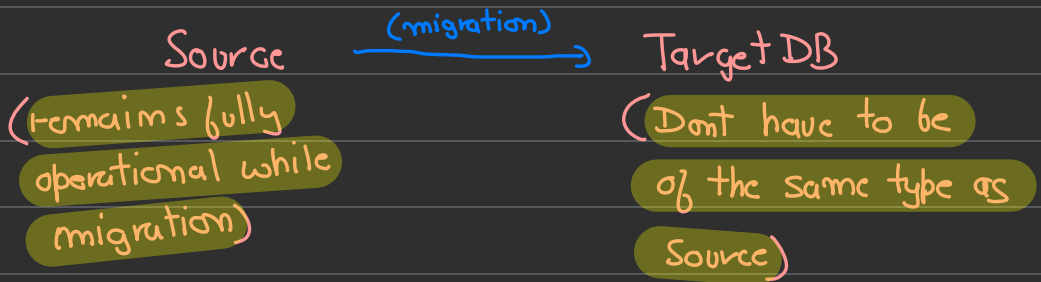
Coordinate Query execution and manage communication

Compute Node

Handles data storage and performs any computation assigned by.

Database Migration Service

AKA amazon's DMS, helps users migrate DBs in secure and easy way.



Homogeneous Migration is the one which takes place between 2 similar type of database eg MySQL to RDS MySQL
→ and target

Source can be anything, on-Premise, EC2 DB or an RDS DB.

Heterogeneous Migration is when source and target is diff. 1st Step is to convert the schema using aws Schema conv. tool. And then 2nd step is to use DMS to migrate data.

There could be other not-so-common use cases of migration like

→ Development and test DB migration

when we wanna test against production data but without affecting production users

(uses DMS to copy DB to dev or test env)

→ Database consolidation - when multiple DBs are to be consolidated into 1.

→ Continuous DB replication - Used to perform continuous data replication.

(due to geographic separation or data recovery.)

Choosing the right DB is complex but one can also choose one of these complex DBs

Amazon's Document DB (with mongoDB compatibility) great for content management, catalogues etc

Amazon Neptune is a graph-based DB for social network application DB and recommendation system, fraud detection needs.

Amazon managed Blockchain for supply chain or Banking type DB needs.

Amazon Quantum ledger Database AKA amazon QLDB is an immutable system of record where any entry can never be removed from audits.

=> DB Accelerators :

Amazon ElastiCache provides caching layers on top of any DB and comes with memecash and redis, helps read time of common request from 1 milisecond to 1 microseconds.

Amazon DynamoDB Accelerator AKA DAX is a native caching layer design to increase read time on non relational Data.



Security

Aws Security :

(Shared responsibility model)

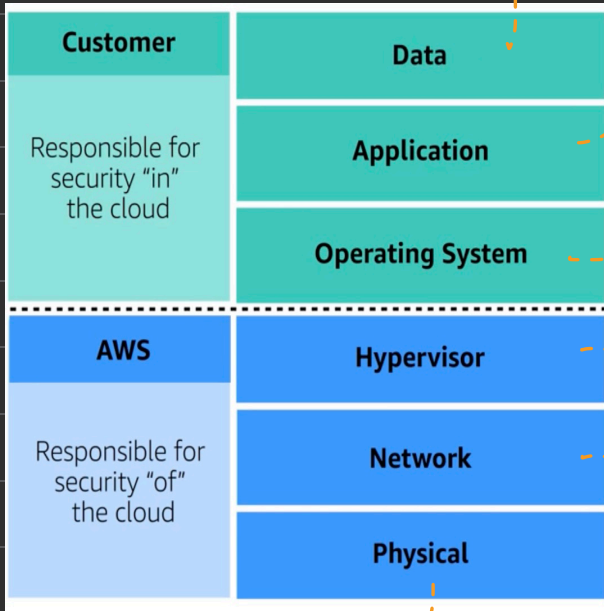
Customer	Customer Data		
	Platform, applications, identity and access management		
	Operating system, network and firewall configuration		
	Client-side data encryption	Server-side data encryption	Network traffic protection
AWS	AWS Foundation Services		
Responsible for security "of" the cloud	Compute	Storage	Database
	Networking		
	AWS global infrastructure		Regions Edge locations Availability Zones

Both the users and amazon takes care of AWS security.

Aws is responsible for the security of some objects (aws looks at our env. as collection of parts build on each other) and for the rest of the objects the end user is responsible 100%.

To normally put it, for infrastructure level security AWS is responsible and for application level, its the end user.

Data is always user's domain to secure and control.



Application run is also 100% owned by the user.

User get to choose the OS and AWS won't access OS ever.

(Magic line separate AWS and user responsibility)

Tamper proof network with high level of abstraction and security.

Physical data center
Security guards, concrete,
Fences etc.

Guest OS **

Customer → Selection, configuration and patching OS to run on an EC2 instance. Configuring Security group, managing user accounts etc.

↑ **

Amazon AWS → responsible for host OS, global infra →
ie regions, AZs. Also covers network
and virtualization infrastructure.

User/Customer Responsibility

Guest OS operations
Rotation of security keys
Manage IAM user's access

AWS Responsibility

host OS operations
Physical changes

User Permission and Access :

AWS gives a root account user who is the owner of the account (have all permissions) and cannot be restricted.

* *

↳ A root user can view tax invoice, close account, view/restore IAM permissions and modify support plans.

Because root user is so important MFA and tokenization login is suggested for the root user

AWS Identity and Access Management AKA amazon's AWS IAM, one can create IAM USER and initially an IAM user have no permissions and cannot even login.

This is called least Privilege Principal - how a user is granted permission/access only to what they need.

IAM policy is JSON document which includes what API an IAM user can/cannot make.

Effect can be either 'allow' or 'deny'

IAM Policy ↴

Action can be any type of API call

```
{  
  "Version": "2012-10-17",  
  "Statement": {  
    "Effect": "Allow",  
    "Action": "s3:ListBucket",  
    "Resource": "arn:aws:s3:::coffee_shop_reports"  
  }  
}
```

Resource lists the AWS resource for which the API call is for

(can either be attached to an IAM group or IAM user)

As a user these policies are not to be written individually, but they're used all over the AWS account.

An **IAM GROUP** is a collection of **IAM users**.

To manage permissions easily we can divide these **IAM users** into **IAM groups**.

can attach a policy to a group and can add multiple users to let them have that permission.

IAM role is predefined type with associated permissions and can be assumed for temp

amount of time, same as user but doesn't have Id and password. → unlike an IAM user have.

IAM role could be used to grant permission of any AWS resource temporarily to any user body → could be a user, external identity, application or any other AWS resource.

When an identity takes up a role, it abandons all the previous permissions it had and assumes the permission of that role.

One can avoid creating IAM user for everyone in an Org, instead one can federate users into the account → which allows users to use regular corporate credentials to login to AWS by mapping the corporate identities to IAM roles.

AWS IAM → Authentication and Authorization as a service.

(According to the shared responsibility model, the CUSTOMER is supposed to manage user access and rotation of secret keys) - **

AWS Organizations :

It helps consolidate and manage multiple AWS account within a central location given that an org/user/company has multiple AWS accounts.

It centralizes management to all accounts, and consolidate payments/billing. Also bulk discounts.

Hierarchical Grouping of accounts can be done to meet security, compliance or budgetary needs and Permissions (per account) can be centrally controlled using Service Control Policies (SCPs)

Amazon SCPs helps the root user to specify max permission for the member accounts in the AWS Organization.

In aws organizations one can group accounts into organization units (OUs) to manage accounts with similar business or security requirements.

SCPs not IAM

Similar to IAM we can also attach policies ↑ to OUs which controls the access of that particular account.

→ Service control Policies

↙ ***

A SCP can be applied to any organization root, an individual member account or an OU. A SCP affects all IAM users, groups and roles of that account including the root user!

Compliance :

Depending on types of resources one use on AWS we need to make sure the application is in compliance with the regulations.

Data protection is a config setting on a lot of AWS resources. reports

AWS also provides multiple white papers and documents to download and use for compliance ↗

AWS Artifact is a service provided by AWS which gives access to AWS security and compliance reports (on-demand) and some select online agreements.

Artifact agreement ←

Used to accept, review and manage agreements for accounts or organizations.

→ **Artifact Reports**

These are up to date compliance reports of AWS and its services tested and verified by auditors.

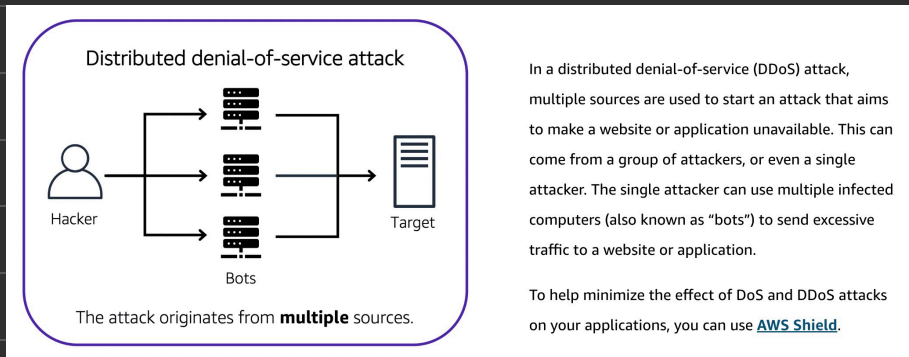
Customer Compliance Center contains resources to help user learn more about AWS compliance.

It also contains compliance stories of companies that have solved various compliance, governance and audit challenges.

Distributed Denial of Services :

Distributed denial of service/ **DDoS** attack is objected to shut down the application's ability to function by overwhelming the system to the point where it can no longer operate.

The attacker leverages other machines on the internet to unknowingly attack the infrastructure eg. **UDP Flood**, **HTTP** level attacks, **Slowloris** attack



These attacks are already taken in consideration and are prevented using the AWS architecture. ↴

eg. low level attack like UDP Flood : Security Groups in aws only allows in proper request traffic, things like weather report uses totally diff protocol than the one customers use, And Security group work at AWS network security level and not EC2 instance level like an OS firewall might.

(ELB)

eg Slowloris attack : AWS elastic load Balancer handles the HTTP traffic request first, so it waits for the entire request to complete before sending it into the application and ELB is Scalable (takes requests in parallel) and works at regional level.

AWS shield

WAF → web application firewall is used to filter incoming traffic of requests, it has extensive ML capabilities and can detect new threats as they evolve.

→ containing unique signatures

Encryption is done on the **data at rest** and the **data in transit** for limited secure access

→ It's a process of securing message or data in a way that only authorized parties can access it.

→ Encryption at rest is applied on data which is stored in the DB, ex DynamoDB (uses AWS Key management Service - AWS KMS to store and secure encryption keys)

For data in transit across applications/internet Secure Socket layer (SSL) is used to encrypt data and server certificates are used to validate and authorize a client.

AWS Shield helps protect application against DDOS attacks. NOT AWS WAF !!

AWS WAF is a web application Firewall that lets a user monitor network requests.

WAF works with Amazon CloudFront and application Load Balancer.

WAF is similar to Network ACL, WAF uses web access control list - web(ACL).

WAF

works at application layer

works with CDN and ELB

Uses managed rule set

Has ML capabilities

Network ACL

works at subnet level


Controls traffic entering and leaving entire subnet.

Uses numbered rules, works in ascending order

Doesn't have it.

AWS Shield covers more resources than WAF. Shield monitors traffic patterns while WAF uses defined rules, while Shield is serverless AWS WAF isn't. Shield being serverless works on its own to protect application from DDOS.

Amazon's Inspector:

It helps to improve security and compliance of the AWS deployed app by running **automated security assessment** against the infrastructure. Particularly points out deviations of best practices, exposure of EC2 instances, vulnerability etc.  Security

Provides network configuration reachability piece, amazon's agent and Security assessment service.

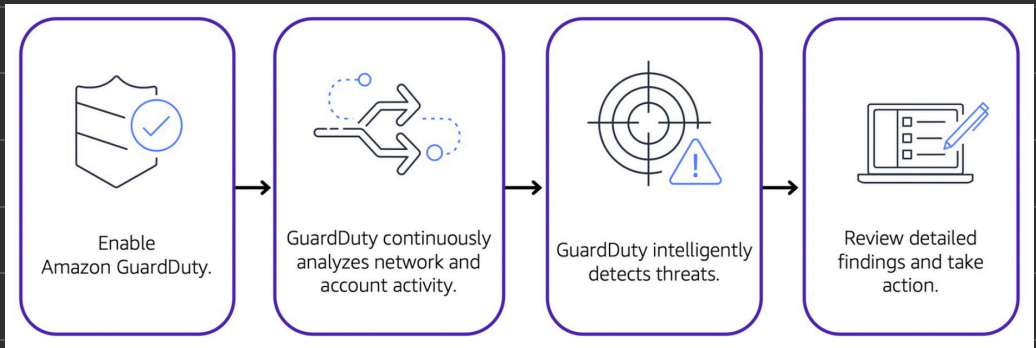
Amazon Guard duty :

↳ runs independently, does not affect performance of other resources

Analyzes continuous streams of metadata generated from an account and network activity found in AWS DNS logs, VPC flowlogs.

Uses integrated threat intelligence like IP addresses, anomaly detection and ML to identify threats more accurately.

* One can configure AWS Lambda Function to take remediation steps automatically in response to GuardDuty's findings.



Amazon GuardDuty is a service that provides intelligent threat detection for AWS infra. and resources.

Use AWS inspector for automated vulnerability scanning of resources OR

Use AWS Trusted Advisor for account-wide best practices, security and cost optimization.

→ **



Monitoring

AWS CloudWatch :

It allows to monitor AWS Infrastructure and applications in **real time**. It works by tracking and monitoring **metrics**.

↳ variables tied to resources
eg. CPU utilization

One can create an AWS cloudwatch alarm, set a threshold for a metric and cloudwatch generates an alert when that happens. (can be a custom metric)

On cloudwatch we also have SNS integration which allows us to send message when any trigger is alerted.

↗ can be configured

Cloudwatch have a dashboard to access real time metrics. Also help reduce MTR and improve TCO.

↓
total cost
of ownership.

↓
mean time to
resolution

Cloudwatch Alarms can be used to trigger custom actions too, such as terminating an EC2 instance after certain amount of requests etc.

AWS CloudTrail :

Its a comprehensive API auditing tool. Any AWS request be it an external request to an AWS EC2 instance or internal like adding another instance etc is logged in the cloudTrail engine.

includes time, response, IP address etc.

Cloudtrail can also save these logs in S3 Buckets.

CloudTrail Insights is another feature which could be enabled to automatically detect unusual API activities.

AWS Trusted Advisor :

provides
guidance.

It's an automated advisor, a service could be used in the AWS account which evaluates resources against 5 Pillars

It compiles categorized items to view in AWS console.

Cost optimization

Performance

Security

Fault tolerance

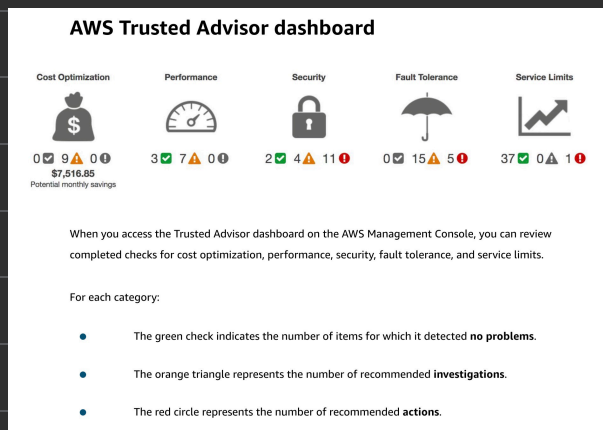
Service limits

Some checks are FREE

while others are available

based on the **Support Plans.**

It could be used to assist the user at all stages of deployment like to create new workflow and to develop new applications.



One can setup email alerts that go out to billing, operation and security contacts as checks runs in an account. (Trusted advisor just have to be turned on)

Cost Optimization checks idle or unused resources that could be eliminated to save cost.

Fault Tolerance checks to help improve an application's availability and redundancy.

Performance checks for high utilization of resources like EC2 instances.

AWS Config is a service that enables the user to access audit and evaluate configuration of AWS resources.



(Utilized to audit change management of AWS Resources)

Just the configs



Pricing

Pricing :

For free tier, depending on the service/product a free tier can be one of these 3



AWS Lambda is always free if the usage is under 1 million invocation per month. (never expires)

AWS S3 is free for 12 months for upto 5gb of storage.

AWS lightsail (used to deploy ready made applications) offers 1mo. free trial of 750hrs usage.

most common services which comes under free tier are Sagemaker, comprehend medical, DynamoDB, SNS, cognito etc.

AWS pricing is more of pay-as-you-go type.

Pay for what

you use, without long term contract and complex licensing.

Pay less when

you reserve, for some AWS services for when you stick to a limited resource (~75% savings)

Pay less with vol.

based discount

when you use

more, so

per-unit cost decrease with increase of units used.

AWS Pricing Calculator could be used to create an estimate for the cost of use case. Estimates can be organized by groups defined. ↷

Once estimate generated, link is created to be shared with others.

AWS billing and cost management dashboard monitors the usage, analyse and control the costs. One can

- Compare current month-to-date Balance with prev month, and get next month estimate based on current month.
- View month-to-date spend by service
- " free tier usage by service.
- Access cost explorer and create budgets.
- Purchase and manage savings plans.
- Publish cost & usage reports.

AWS Consolidated Billing is billing feature on top of AWS's **organisation service**. It lets you bill all the accounts under an org. together to build 1 consolidated bill.

- It makes it easier to track combined cost
- It lets the organisation share bulk discount pricing, saving plans and reserved instance across all accounts.

AWS Budgets allows to set custom budgets (cost - thresholds) for variety of scenarios for aws services.

User gets message when the usage exceeds or is forecasted to exceed the budget.

Budget can be customized and one can set an alert threshold to get notified.

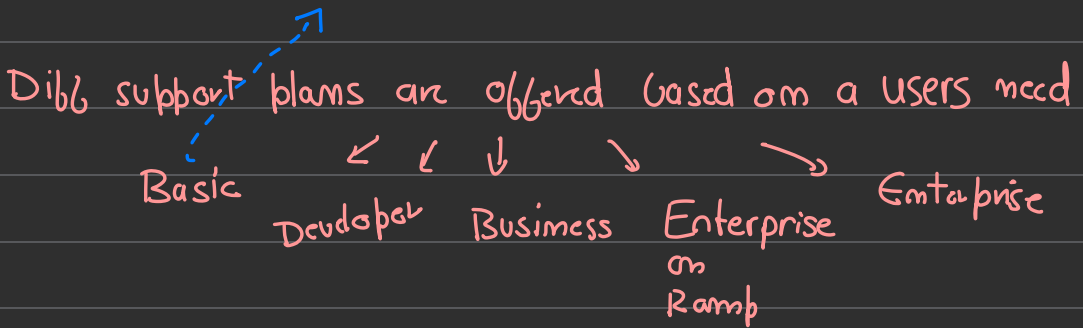
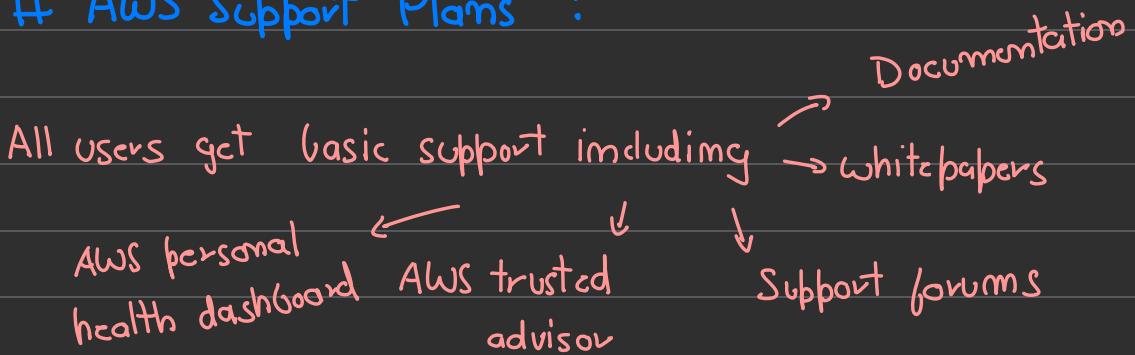
AWS Cost Explorer is a tool to lets you visualize, understand and manage cost and usage over time.

It contains usage data upto past 12 months.

The cost explorer -by default generate report of cost and usage of top 5 cost accruing service. ↴

custom filters and groups can be applied.

AWS Support Plans :



Other plans than the Basic ones have pay-by-month pricing. (no long-term contract)

Developer Support

Customers in the **Developer Support** plan have access to features such as:

- Best practice guidance
- Client-side diagnostic tools
- Building-block architecture support, which consists of guidance for how to use AWS offerings, features, and services together

For example, suppose that your company is exploring AWS services. You've heard about a few different AWS services. However, you're unsure of how to potentially use them together to build applications that can address your company's needs. In this scenario, the building-block architecture support that is included with the Developer Support plan could help you to identify opportunities for combining specific services and features.

(Basic support + Email access to customer support)

Business Support

Customers with a **Business Support** plan have access to additional features, including:

- Use-case guidance to identify AWS offerings, features, and services that can best support your specific needs
- All AWS Trusted Advisor checks
- Limited support for third-party software, such as common operating systems and application stack components

Suppose that your company has the Business Support plan and wants to install a common third-party operating system onto your Amazon EC2 instances. You could contact AWS Support for assistance with installing, configuring, and troubleshooting the operating system. For advanced topics such as optimizing performance, using custom scripts, or resolving security issues, you may need to contact the third-party software provider directly.

[Basic and Dev
Support
+
Trusted advisor
+
Direct phone access
to customer support]

Enterprise On-Ramp Support

In November 2021, AWS opened enrollment into AWS Enterprise On-Ramp Support plan. In addition to all the features included in the Basic, Developer, and Business Support plans, customers with an Enterprise On-Ramp Support plan have access to:

- A pool of Technical Account Managers to provide proactive guidance and coordinate access to programs and AWS experts
- A Cost Optimization workshop (one per year)
- A Concierge support team for billing and account assistance
- Tools to monitor costs and performance through Trusted Advisor and Health API/Dashboard

Enterprise On-Ramp Support plan also provides access to a specific set of proactive support services, which are provided by a pool of Technical Account Managers.

- Consultative review and architecture guidance (one per year)
- Infrastructure Event Management support (one per year)
- Support automation workflows
- 30 minutes or less response time for business-critical issues

(Business
support
+
30 min response
for business critical
workload
+
Access to pool of
technical account
managers
(TAMs)

Enterprise Support

In addition to all features included in the Basic, Developer, Business, and Enterprise On-Ramp support plans, customers with Enterprise Support have access to:

- A designated Technical Account Manager to provide proactive guidance and coordinate access to programs and AWS experts
- A Concierge support team for billing and account assistance
- Operations Reviews and tools to monitor health
- Training and Game Days to drive innovation
- Tools to monitor costs and performance through Trusted Advisor and Health API/Dashboard

The Enterprise plan also provides full access to proactive services, which are provided by a designated Technical Account Manager:

- Consultative review and architecture guidance
- Infrastructure Event Management support
- Cost Optimization Workshop and tools
- Support automation workflows
- 15 minutes or less response time for business-critical issues

(Enterprise on
ramp support
+
15 min response
time
+
Designated
TAM)

AWS Technical Account Manager (TAMs) is the primary contact person at AWS (if gone for any of the top 2 support plans)

TAM educates, provides engineering guidance, help design solution, helps with cost effective and resilient architectures and provides direct access to AWS programs and broad community of experts.

AWS marketplace is a digital catalogue of 3rd party independent software vendors. Its used to find, test and buy software that runs on AWS.

AWS Service Quotas can be used to view and manage service quotas when AWS workloads grow for a user/organization/account.

AWS basic support and AWS Developer Support can access CORE SECURITY CHECKS.





Migration

AWS Migration:

AWS Cloud Adoption Framework AKA AWS CAF helps manage migration through guidance. CAF provides advice to migrate to AWS.

It's divided into 6 diff parts based on what kind of people/workforce is needed for migration.

Business	People	Governance	Platform	Security	Operations
					
Business capabilities.			Technical capabilities		

Each perspective is used to bridge gap between skills and processes which are recorded as inputs which are then used to create AWS CAF Action Plan.

Action Plan helps guide the organization for cloud migration.

Business Perspective

The **Business Perspective** ensures that IT aligns with business needs and that IT investments link to key business results.

Use the Business Perspective to create a strong business case for cloud adoption and prioritize cloud adoption initiatives. Ensure that your business strategies and goals align with your IT strategies and goals.

Common roles in the Business Perspective include:

- Business managers
- Finance managers
- Budget owners
- Strategy stakeholders

People Perspective

The **People Perspective** supports development of an organization-wide change management strategy for successful cloud adoption.

Use the People Perspective to evaluate organizational structures and roles, new skill and process requirements, and identify gaps. This helps prioritize training, staffing, and organizational changes.

Common roles in the People Perspective include:

- Human resources
- Staffing
- People managers

Governance Perspective

The **Governance Perspective** focuses on the skills and processes to align IT strategy with business strategy. This ensures that you maximize the business value and minimize risks.

Use the Governance Perspective to understand how to update the staff skills and processes necessary to ensure business governance in the cloud. Manage and measure cloud investments to evaluate business outcomes.

Common roles in the Governance Perspective include:

- Chief Information Officer (CIO)
- Program managers
- Enterprise architects
- Business analysts
- Portfolio managers

Platform Perspective

The **Platform Perspective** includes principles and patterns for implementing new solutions on the cloud, and migrating on-premises workloads to the cloud.

Use a variety of architectural models to understand and communicate the structure of IT systems and their relationships. Describe the architecture of the target state environment in detail.

Common roles in the Platform Perspective include:

- Chief Technology Officer (CTO)
- IT managers
- Solutions architects

Security Perspective

The **Security Perspective** ensures that the organization meets security objectives for visibility, auditability, control, and agility.

Use the AWS CAF to structure the selection and implementation of security controls that meet the organization's needs.

Common roles in the Security Perspective include:

- Chief Information Security Officer (CISO)
- IT security managers
- IT security analysts

Operations Perspective

The **Operations Perspective** helps you to enable, run, use, operate, and recover IT workloads to the level agreed upon with your business stakeholders.

Define how day-to-day, quarter-to-quarter, and year-to-year business is conducted. Align with and support the operations of the business. The AWS CAF helps these stakeholders define current operating procedures and identify the process changes and training needed to implement successful cloud adoption.

Common roles in the Operations Perspective include:

- IT operations managers
- IT support managers

The 6 R's :

→ option

These include the 6 strategies used for migration.

1. **Rehosting** : AKA 'lift and shift' involves moving apps without change. No modification is done to the infra or logic just the cloud is changed to AWS.
2. **Replatforming** : AKA 'lift-tinker-shift' involves making small cloud optimization to realize tangible benefit and is done without changing core architecture of application.
3. **Refactoring / re-architecting** : It involves changing the application to use most out of cloud-native features which would otherwise be difficult to achieve in the application's existing environment.
(highest initial cost for planning and human effort)

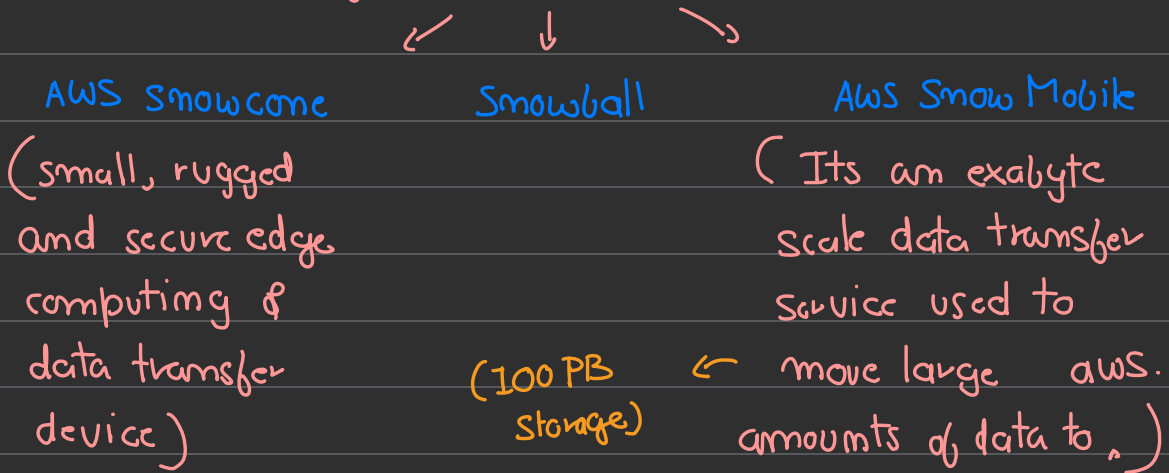
4. Repurchasing : involves traditional license and SaaS model. eg migration of existing CRM to Salesforce.

5. Retaining : It consist of keeping the apps that are crutial/critical in the source env. where the major application's migration can be delayed for later time.

6. Retiring : Its the process of removing apps which are no longer needed. eg older versions of application etc.

AWS Snow Family :

Snow is a collection of physical devices that helps to physically transport upto exabytes of data to and from AWS.



Snowball offers 2 types of devices, edge storage optimized and compute optimized.

Snowball edge storage optimized devices are well suited for large scale data migration and recurring transfer workflows.

It provides storage of 80 TB of HDD and 1 TB of SATA SSD.

For compute service it provides 40 vCPUs and 80 GiB of memory to support EC2.

Snowball edge compute Optimized provides powerful computing resource for use case such as ML, video analytics.

It provides 80 TB HDD and 28 TB of usable NVMe SSD. And for compute it provides 104 vCPUs, 416 GiB memory and optional NVIDIA Tesla V100 GPUs.

AWS Misc Services :

- AWS Sagemaker : Helps quickly build, deploy and train ML models at scale, or to build custom models with support of all possible frameworks.
- Augmented AI(A2I) : provides ML platform (requires human review for ML predictions) that anyone can use to build application without needing extensive knowledge.
- Amazon Lex : Heart of Alexa helps build chatbots.
- Amazon textract : Extract text and data from documents to use them
- AWS Deepbracer : Helps to use/integrate reinforcement learning.
- AWS Ground Station : Satellite usage.

AWS Well Architected Framework :

It helps architects and developers to build secure, high performance, resilient and efficient infrastructure for the application.

It consists of 6 pillars to ensure a consistent approach to viewing and designing architectures.

1. **Operational Excellence** is the ability to run and monitor systems to deliver business value and to continually improve supporting process and procedures. **working effectively,**
eg. Deployment Pipelines etc.

2. **Security** : Ability to **protect information,** systems, and assets while delivering business value through risk assessment and mitigation strategies.
eg. data encryption and authentication.

* (also ability to do work consistently and correctly)
↓

3. Reliability : Ability of system to recover from any infra or service disruptions, Dynamically acquire computing resource to meet requirement.

eg. includes testing recovery solutions, scaling horizontally etc.

4. Performance Efficiency : Ability to use computation resources efficiently and maintain the efficiency when the demand changes.

eg. using correct EC2 type based on workload

5. Cost optimization : Ability to run systems to deliver business value at lowest price

6. Sustainability : Ability to continually improve sustainability by reducing energy consumption and increasing efficiency.

All of these factors are calculated and taken care by a solution's architect BUT the AWS Framework the AWS Well-Architected Tool could be used to generate reports for areas which need improvement in the application on the Basis of the 6 pillars defined above.

Looks like traffic signal Green, Orange and Red.

←
You're good, keep
up the good work

↙
Probably look
into the specific
area which can use
some help/optimization

↓
Priority change
required,
something
at risk.

Also provides plans to use best practices. (its customisable and if questions don't apply one can override it).

Advantages of AWS Cloud :

- Trade upfront expense to variable expense.
- Benefit from massive economies of scale.
(achieve comparatively lower cost to running a data center on-premise.)
- Stop guessing capacity. (can use scaling)
- Increase speed and agility.
- Stop spending money by running and maintaining data centers.
- Go global in minutes.