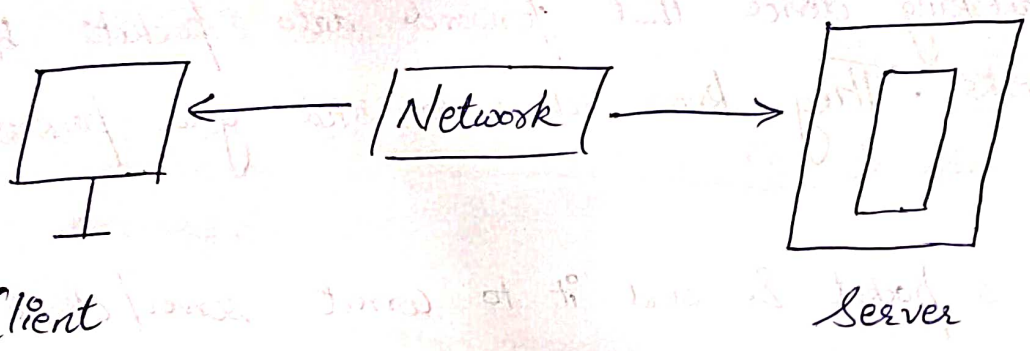
CLOUD COMPUTING

How Website Work?



We have a server hosted somewhere, and we as a web browser want to get access to that server to visualize a website. (Use a network)

The browser/client/us will find the network & will use network to route the packets, the data to the server & then the server will respond to us and we can view a website

⊙ Client and servers both have IP addresses to locate each other

What is a server Composed of?

→ Compute : CPU (Computations / Calculations)

→ Memory : RAM (very fast, helps store info & quickly set)

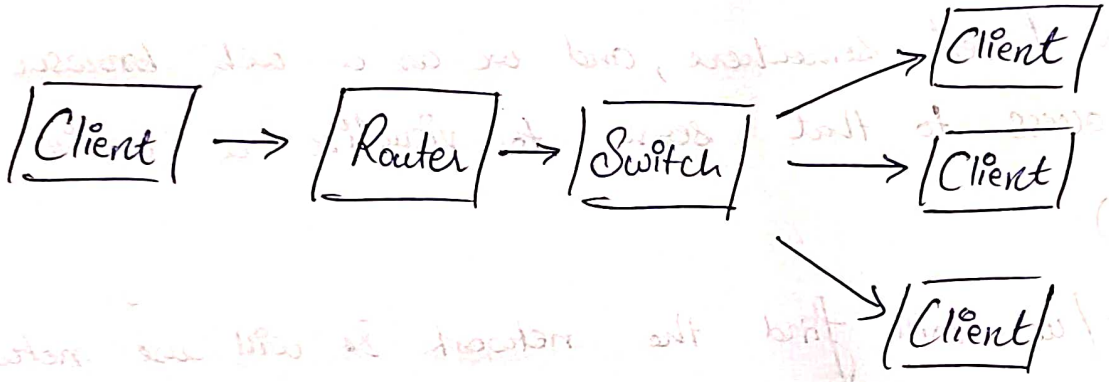
→ Storage : Data (Storing data somewhere)

→ Database : Store data in a structured way.

→ Network : Routers, switch, DNS Server

Important Terms

- Network : cables, routers & servers connected with each other
- Router : A networking device that forwards data packets b/w computer networks. They know where to send your packets on the internet
- Switch : Takes a packet & send it to correct server/client on your network



Problems With Traditional IT Approach

1. Pay for rent for data center
2. Pay for power supply, cooling, maintainence
3. Adding & replacing hardware takes time
4. Limited Scaling
5. Hire 24/7 team to monitor infrastructure
6. Disasters (earthquake, fire, --)

What is Cloud Computing?

- Cloud Computing is the on-demand delievery of computer power, database storage, applications & other IT Resources
- Through a service (cloud) platform with pay-as-you-go pricing.
- You can provision exactly the right type & size of computing resources you need.
- You can access as many resources as you need, almost instantly
- Simple way to access servers, storage, databases & a set of application services

→ Amazon Web Service (AWS) owns & maintains the network connected hardwares required for these applications - services, while you provision & use what you need via a web application

Cloud Services

- GMAIL : Email Cloud Service
- DROPBOX : Cloud Storage Service
- NETFLIX : Built on Aws (Videos on demand)

THE Deployment Models of the Cloud

Private Cloud

- Cloud services used by a single organization, not exposed to the public
- Complete Control
- Security for sensitive applications
- Meet specific business needs.

Public Cloud

- Cloud Services owned & operated by a third party cloud service provider delievered over the Internet
  Eg → Microsoft Azure, Google Cloud, AWS etc

Hybrid Cloud

- Keep some servers on premises & extend some capabilities to the Cloud
- Hybrid of our own infrastructure & the cloud (AWS)
- Control over sensitive assets in your private infrastructure
- Flexibility & cost-effectiveness of the public cloud

Characteristics of Cloud Computing

① On-demand self Service → Users can provision resources & use them without human interaction from the service provider

② Broad Network Access → Resources available over the network & can be accessed by diverse client platforms

③ Multi-tenancy & resource pooling → ⊙ Multiple customers can share the same infrastructure & applications with security & privacy
⊙ Multiple customers are serviced from the same physical resources

④ Rapid elasticity & Scalability → ⊙ Automatically & quickly acquire and dispose resources when needed
⊙ Quickly and easily scale based on demand

⑤ Measured Service → Usage is measured, users pay correctly for what they have used.

Advantages of Cloud Computing

① Trade Capital Expense (CAPEX) for operational expense (OPEX)
⊙ Pay on demand → Dont own hardware
⊙ Reduced total cost of ownership (TCO) & Operational Expense (OPEX)

② Benefit from massive economies of scale
⊙ Prices are reduced as AWS is more efficient due to large scale

③ Stop guessing capacity
- Scale based on actual measured usage.

④ Increase speed & agility

⑤ Stop spending money running & maintaining data centers

Problems Solved By Cloud

① Flexibility
② Cost - effectiveness
③ Scalability
④ Elasticity
⑤ High availability & fault tolerance
⑥ Agility

Types Of Cloud Computing

① Infrastructure as a Service (IAAS)
- Provides Building Blocks for Cloud IT
- Provides networking, computers, data storage space
- Highest level of flexibility
- Easy parallel with traditional on premises IT

Eg: EC2, GCP, Azure, Linode

② Platform as a Service (PAAS)

- Removes the need for your organization to manage underlying infrastructure
- Focus on deployment & management of your applications.

Eg → Elastic Beanstalk, Heroku, Google App Engine, Azure

③ Software as a Service (SAAS)

- Completed product that is run and managed by the service provider

Eg → Google Apps (Gmail), Dropbox, Zoom

AWS Regions

- AWS has regions all around the world
- Names can be us-east-1, eu-west-3, ---
- A region is a cluster of data centers
- Most AWS services are region-scoped

How to choose an aws region?

- Compliance with govt.
- Proximity to customers
- Available service within a region
- Pricing

AWS Availability Zones

- Each region has many availability zones (usually 3, min is 3, max is 6). Eg → ap-south-1a, ap-south-1b, ap-south-1c
- Each availability zone (AZ) is one or more discrete data centers with redundant power, networking & connectivity
- They are separated from each other, so that they are isolated from disasters
- They are connected with high bandwidth, ultra low latency networking.

AWS Points of Presence

- Amazon has 400+ Point of Presence (400+ Edge locations and 10+ Regional Caches) in 90+ cities across 40+ countries.
- Content is delivered to end user with low latency.

Tour of AWS Console

- AWS has global services
  1. Identity and Access Management (IAM)
  2. Route 53 (DNS)
  3. CloudFront (Content Delivery Network)
  4. WAF (Web Application Firewall)
- Most AWS services are Region-scoped.
  1. Amazon EC2 (IAAS)
  2. Elastic Beanstalk (PAAS)
  3. Lambda (FAAS)
     ↳ function as a service
  4. Rekognition (SAAS)

IAM : Users & Groups

- IAM → Identity and Access management, Global Service
- Root account created by default, shouldn't be used or shared
- Users are people within your organization, and can be grouped
- Groups (only) contain users, not other groups
- Users dont have to belong to a group, and users can belong to multiple groups.

IAM : Permissions

- Users or groups can be assigned JSON documents called policies
- These policies define the permissions of the users
- In AWS, you apply the least privilege principle : Dont give more permissions than a user need

IAM : Password Policy

IAM > Account Settings > Edit Password Policy

- 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 uppercase letters
    - Lowercase letters
    - numbers
    - non-alphanumeric characters (?_-)

→ Allow all IAM users to change their own password
→ Require users to change their passwords after sometime (password expiration)
→ Prevent Password re-use (dont use previously used pass.)

MULTI-FACTOR AUTHENTICATION (MFA)

- Login > Security Credentials > Assign MFA > Install app on your other device (MFA can also be removed)

⊙ Users have access to your account & can possibly change configuration or delete resources in your AWS account.

⊙ We want to protect our root accounts & IAM user

⊙

MFA = Password you know + Security device you own

⊙ Main benefit of MFA :
if a password is stolen or hacked, the account is not compromised

MFA device options in AWS

⊙ Virtual MFA device → Support for multiple token on a single device. Eg→ Google Authenticator (phone only)
Authy (multi-device)

⊙ Universal 2nd factor (U2F) Security Key → Support for multiple root & IAM users using a single security key
Eg→ Yubikey by Yubico (3rd party)
(device)

→ Allow all IAM users to change their own password
→ Require users to change their passwords after sometime (password expiration)
→ Prevent Password re-use (dont use previously used pass.)

MULTI-FACTOR AUTHENTICATION (MFA)

- Login > Security Credentials > Assign MFA > Install app on your other device (MFA can also be removed)

⊙ Users have access to your account & can possibly change configuration or delete resources in your AWS account.

⊙ We want to protect our root accounts & IAM user

⊙

MFA = Password you know + Security device you own

⊙ Main benefit of MFA :
if a password is stolen or hacked, the account is not compromised

MFA device options in AWS

⊙ Virtual MFA device → Support for multiple token on a single device. Eg→ Google Authenticator (phone only)
Authy (multi-device)

⊙ Universal 2nd factor (U2F) Security Key → Support for multiple root & IAM users using a single security key
Eg. yubikey by yubico (3rd party)
(device)

- Hardware key fob MFA Device → Provided by Gemalto (3rd party)
- Hardware key fob MFA Device for AWS GovCloud → Provided by SurePass ID (3rd party)

How can users access AWS ?

- To access AWS, you have 3 options
  - → AWS Management Console
    Protected by password + MFA (optional)
  - → AWS Command Line Interface (CLI)
    Protected by access keys
  - → AWS Software Developer kit (SDK)
    for code: protected by access keys

- Access keys are generated through AWS console
- Users manage their own access keys
- Access keys are secret, just like a password
- Access key ID ~= username] (just like this!)
  Access key ~= password
  ↳ secret

AWS CLI

- A tool that enables you to interact with AWS service using commands in your command line shell
- Direct access to the public APIs of AWS services
- Alternative for using AWS management console

AWS SDK

- AWS Software Development Kits
- Language Specific APIs (set of libraries)
- Enables you to access & manage AWS services programatically.
- Embedded within your application
- Supports
  - → SDKs (Javascript, Python, PHP, .Net, Ruby, Java, GO, Node.js, C++)
  - → Mobile SDKs (iOS, Android, ---)
  - → IoT Device SDKs (Embedded C, Arduino --)
- Example :- AWS CLI is build on AWS SDK for python

⇒ Instead of using the AWS management console, we can also use the AWS CloudShell (>_)

Commands (CloudShell & CLI)

- aws
- aws --version
- aws iam lPst-user
- ls
- echo "test" > demo.txt (add "text" as content in demo.txt)
- cat demo.txt
- pwd (present working directory)

(Download / Upload files using Actions button)

IAM Roles for Services

- Some aws service will need to perform actions on your behalf
- To do so, we will assign permissions to AWS services with IAM Roles
- Common Roles:
  - → EC2 instance Roles
  - → Lambda function Roles
  - → Roles for Cloud formation

IAM Security Tools

IAM > Users > 'User' > Access Advisor

- IAM Credentials Report (account-level)
  A report that lists all your account's users and the status of their various credentials
- IAM Access Advisor (user-level)
  - → Access advisor shows the service permissions granted to a user and when those services were last accessed
  - → You can use this information to revise your policies.

SHARED RESPONSIBILITIES MODEL FOR IAM

AWS

- Infrastructure (global network security)
- Configuration & vulnerability analysis
- Compliance validation

YOU

- Users, groups, Roles, Policies management & monitoring
- Enable MFA on all device (accounts)
- Rotate all your keys often
- Use IAM tools to apply appropriate permission
- Analyze access patterns & review permissions

IAM Section (Summary)

- Users :- mapped to a physical user, has a password for aws console
- Groups & contains users only, groups cannot contain another groups
- Policies & JSON document that outlines permission for users or groups
- Roles & For EC2 Instances or AWS Services
- Security & MFA + Password Policy
- AWS CLI & manage your aws service using command line
- AWS SDK & " " " " " programming language
- Access Keys & access aws using CLI or SDK
- Audit & IAM Credentials Reports & IAM Access advisor

IAM Credentials Report lists all your Account's users and the status of their various credentials. The other IAM Security tool is IAM Access Advisor. It shows the service permission granted to a user & when those services were last accessed.

EC2 - Elastic Compute Cloud (IAAS)

AWS Budget Setup → AWS budgets lets you set custom cost & usage budgets that alert you when your budget thresholds are exceeded

Amazon EC2

- EC2 is one of the most popular of AWS' offering
- EC2 = Elastic Compute Cloud = IAAS
- It mainly consists in capability of
  - → Renting virtual machines (EC2)
  - → Storing data on virtual drives (EBS)
  - → Distributing load across machines (ELB)
  - → Scaling the services using an auto scaling group (ASG)
- Knowing EC2 is fundamental to understand how the Cloud works

EC2 sizing & configuration options

- Operating System (OS): Linux, Windows or Mac OS
- How much compute power & cores (CPU)
- How much RAM
- How much storage space
  - → Network attached (EBS & EFS)
  - → Hardware (EC2 Instance store)
- Network speed: speed of the card, Public IP addresses
- Firewalls & Security group
- Bootstrap script (Config, at first launch): EC2 user data

EC2 - Elastic Compute Cloud (IAAS)

AWS Budget Setup → AWS budgets lets you set custom cost & usage budgets that alert you when your budget thresholds are exceeded

Amazon EC2

- EC2 is one of the most popular of AWS' offering
- EC2 = Elastic Compute Cloud = IAAS
- It mainly consists in capability of
  - → Renting virtual machines (EC2)
  - → Storing data on virtual drives (EBS)
  - → Distributing load across machines (ELB)
  - → Scaling the services using an auto scaling group (ASG)
- Knowing EC2 is fundamental to understand how the Cloud works

EC2 sizing & configuration options

- Operating System (OS) : Linux, Windows or Mac OS
- How much compute power & cores (CPU)
- How much RAM
- How much storage space
  - → Network attached (EBS & EFS)
  - → Hardware (EC2 Instance store)
- Network speed : speed of the card, Public IP addresses
- Firewalls & Security group
- Bootstrap script (Config, 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
- The script is only run once at the instance first start
- EC2 user data is used to automate boot tasks such as
  - → Installing updates
  - → " softwares
  - → Downloading common files from the Internet
  - → Anything you can think of
- The EC2 user data scrip runs with root user

Eg → t2.micro
t2.xlarge

EC2 Instance Types

AWS has the following naming convention

m5.2xlarge

m → instance class
5 → generation
2xlarge → size within instance class

EC2 instance types are →

→ → →

① General Purpose → ⊙ Great for a diversity of workloads such as web servers or code repositories

⊙ Balance between
- → Compute
- → Memory
- → Networking

⊙ In the course, we use t2.micro, which is a general purpose EC2 instance

② 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

③ 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)
- → Applications performing real time processing of big unstructured data

④ Storage Optimized → great for storage-intensive tasks that require high, sequential read & write access to large data sets on local storage

⊙ Use cases →

- High frequency online transactions processing (OLTP) systems
- Relational & NoSQL data bases
- Cache for in-memory databases (Eg- Redis)
- Data warehousing applications
- Distributed file system

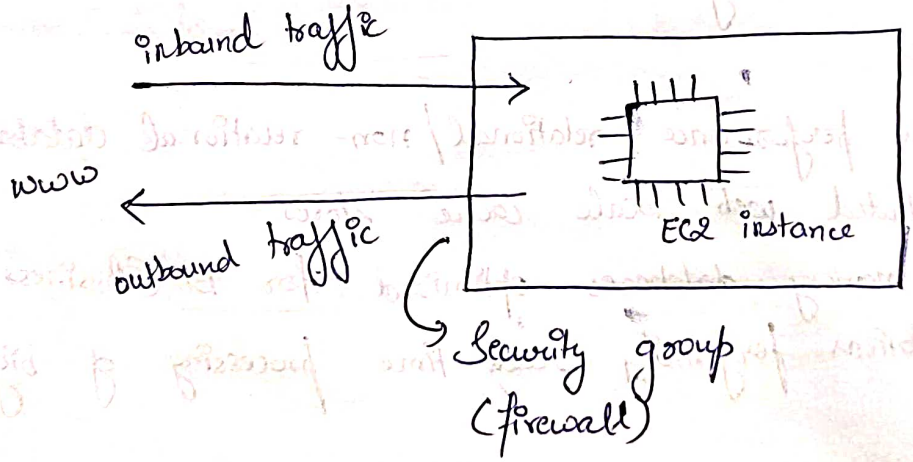
SECURITY GROUPS

⊙ Fundamental of network security in AWS

⊙ They control how traffic is allowed into or out of our EC2 instances

⊙ Security groups only contain allow rules

⊙ Security groups rules can reference by IP or by security group



⊙ Security group acts as a firewall on EC2 instances.

⊙ They regulate :

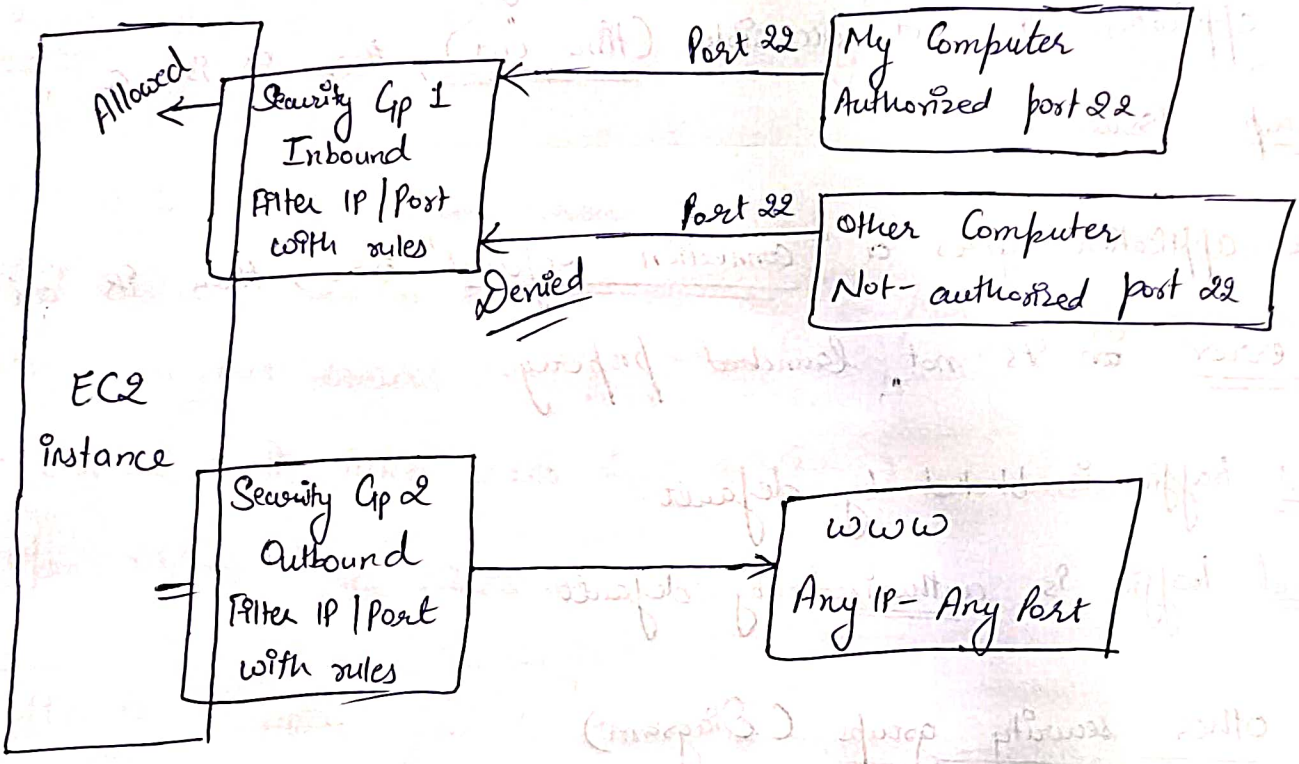
→ access to ports

→ Authorized IP ranges - IPv4 or IPv6

→ Control of inbound network

→ " " outbound network

Diagram



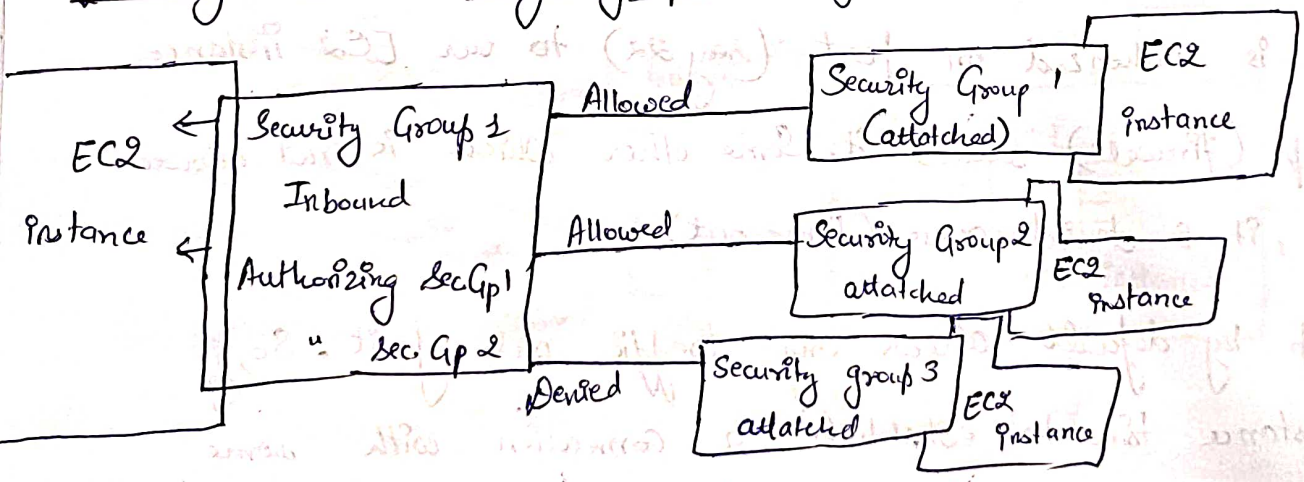
Our computer is authorized on port (say 22) to our EC2 instance Security group (Firewall) allows it. Some other which is not allowed at port 22, it is denied access (time-out)

Security group by default allows any traffic out of it. So, if our EC2 instance tries to establish a connection with some website, our security groups allows it.

Security Groups - Good to Know

- Can be attatched to multiple instances
- Locked down to a region / VPC combination (present in one VPC, not in other)
- Does 'live' outside the EC2 - if the traffic is blocked the EC2 instance won't see it
- Its good to maintain one separate security group for SSH access
- If your application is not accessible (time-out), then it is a security group issue
- If your application gives a "connection refused" error, then its an application error or its not launched properly.
- All inbound traffic is blocked by default.
- All outbound traffic is authorized by default

Referencing other security groups (Diagram)



Ports

- 22 = SSH (Secure Shell) - log into a Linux Instance
- 21 = FTP (File Transfer Protocol) - upload files into a file share
- 22 = SFTP (Secure FTP) - upload files using SSH
- 80 = HTTP - access unsecured websites
- 443 = HTTPS - access secured websites
- 3389 = RDP (Remote Desktop Protocol) - log into a window Instance

⇒ We cant see the text when we click on open address under Public IPv4 address (Timeout), but if we copy & paste that respective IP address in our browser, it works fine.

Ctrl + C → it takes uses port 80 (HTTP) not secured

Open address → uses HTTPS (secured) It doesn't work

SSH → It allows you to control a remote machine, all using the command line.

Shared Responsibility Model for EC2

AWS	YOU
Infrastructure (Global network security)	Security group rules
Isolation on physical hosts	Operating system patches & updates
Replacing faulty hardware	Softwar & utilities installed on EC2 Instance.
Compliance Validation	IAM Roles assigned to EC2 & IAM user access management
	Data Security on your Instance

EC2 Instance Pricing Options

- On-Demand Instances → Short workload, predictable pricing, pay by second
- Reserved (1 & 3 years)
  - → Reserved Instances : long workload
  - → Convertible Reserved Instance : Long workload with flexible instances
- Savings Plan (1 & 3 years) → Commitment to an amount of usage, long workload
- Spot instances → Short workloads, cheap, can loose instance (less reliable)
- Dedicated hosts → book an entire physical server, control instance placement
- Dedicated instance → No other customer will share your hardware
- Capacity Reservation → Reserve capacity in a specific Availability zone for any duration

On-Demand

- Pay for what you use
  - → Linux or Windows : Billing per second, after first min
  - → All other OS : Billing per hour
- Has the highest cost but no upfront payment.
- No long term commitment
- Recommended for short workload, when you cant predict how the application will behave.

Reserved

- Upto 72% discount compared to On-Demand
- You reserve a specific instance attribute
  ↳ (Instance type, Region, Tenancy, OS)
- Reservation Period : 1 year (+dis) or 3 yr (+++dis)
- Recommended for steady-state usage applications
- You can buy & sell in reserved instance marketplace

Convertible Reserved Instance

- Upto 66% discount
- Can change the EC2 Instance type, instance family, OS, scope & tenancy

Saving Plan

- Get a discount based on long term usage (upto 72% - same as Reserved)
- Commit to a certain type of usage
- Usage beyond EC2 saving plan is billed at the on demand price
- Flexible across
  - → Instance size (eg m5.2xlarge)
  - → OS (Linux/Windows)
  - → Tenancy (Host, dedicated, Default)

Spot Instances

- Get a discount of 90% compared to on-Demand
- Instances that you can "lose" at any point of time if your max price is less than the current spot price
- MOST cost-efficient instances in AWS

- Useful for workloads that are resilient to failure
  - → Batch Jobs
  - → Data Analysis
  - → Image processing
  - → Any distributed workloads
  - → Workloads with a flexible start & end time
- Not suitable for critical jobs or databases.

Dedicated Hosts

- A physical server with EC2 instance capacity fully reserved (dedicated) to your use
- Allows you to address compliance requirements & use your existing server-based software licenses (per-socket, per-core, per-VM software licenses)
- Purchasing options
  - → On-demand - pay per sec. for active dedicated Hosts
  - → Reserved (1 or 3 yr)
- Most expensive option
- Useful for software that have complicated licensing model (BYOL - Bring your own license)
- Or for companies that have strong or compliance needs.

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

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 commitments (create / cancel anytime), no billing discounts
- Combine with Regional Reserved Instances & saving plan to benefit from billing discounts.
- You are 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.

EC2 - Summary

- EC2 instance : AMI (OS) + Instance size (CPU + RAM) + Storage + Security groups + EC2 user data
- Security groups : Firewall attached to the EC2 instance
- EC2 user data : Script launched at first start of an instance
- SSH : start a terminal into our EC2 instances (port 22)
- EC2 Instance Role : Link to IAM Roles.
- Purchasing Options : On demand, Spot, Reserved (Standard + Convertible), Dedicated host, Capacity Reservations, Dedicated Instances

Types of Storage you can attatch to EC2 instance

1. EBS (Elastic Block Store)
2. Instance Store
3. EFS (Elastic File System)

stop &
- When you start instance, it changes its Public IP, to have fix public IP, you need elastic IP
- You can have 5 elastic IP in your acc.

EBS Volume

- An EBS (Elastic Block Store) volume is a network drive you can attatch to your instances while they run
- It allows your instances to persist data, even after their termination
- They can only be mounted to one instance at a time (at the CCP level)

Except Io1/Io2 family → Certified Cloud Practitioner

- They are bounded to specific AZ
- 30GB of free EBS storage of type General Purpose (SSD)
- Its a network drive (ie not a physical drive)
  - → It uses the network to communicate the instance, which means there might be a bit of delay
  - → It can be detatched from an EC2 instance & attatched to another one quickly
- We can have one EBS attatched to one EC2 instance
- We can have multiple EBS attatched to one EC2 instance
- No single EBS can be attatched to 2 instances
- An EBS can be left idle (un-attatched)

EBS - Delete On Termination Attribute

- Controls the EBS Behaviour when an EC2 instance terminates
  - → By default, the root EBS volume is deleted (attribute enabled)
  - → By default, any other attatched EBS Volume is not deleted (attribute disabled)
- This can be controlled by AWS Console / AWS CLI

⊙ Use Case :

Preserve root volume when instance is terminated

⇒ EC2 Instance > Storage

EBS > Volume > Create volume

2GiB

gp2

az → has to be same as of Instance

az → us-east-1a

actions > attatch volume > select instance (EBS)

&

EBS is attatched to our Instance

[Create multiple EBS and attatch them to one Instance]

On default EBS, delete on terminates → Yes

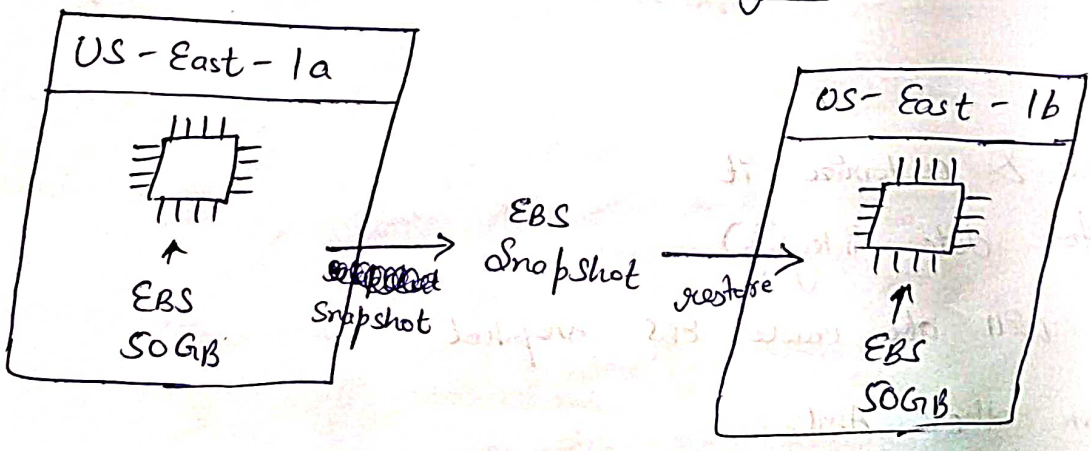
Terminate Instance → that EBS will be gone

EBS Snapshot

⊙ Make a backup (snapshot) of your EBS volume at a point in time

⊙ Not necessary to detach volume to do snapshot but recommended

⊙ Can copy snapshots across AZ or regions



Features →

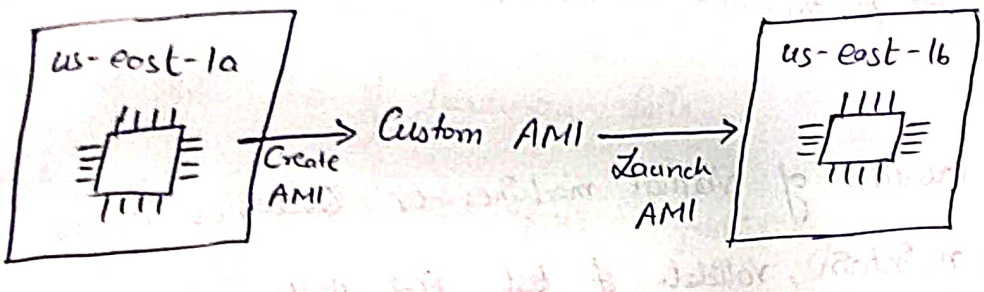
- EBS Snapshot Archive
  - → Move a snapshot to an "archive tier" that is 75% cheaper
  - → Takes within 24 to 72 hours for restoring the archive
- Recycle bin for EBS Snapshots
  - → Setup rules to retain deleted snapshots so you can recover them after an accidental deletion
  - → Specify retention (from 1 day to 1 yr)

AMI Overview

- AMI - Amazon Machine Image
- AMI are a customization of EC2 instance
  - → you add your own software, configuration, operating system, monitoring —
  - → Faster boot / configuration time because all your software is pre-packaged
- AMI are built for a specific region (and can be copied across regions)
- You can launch EC2 instance from
  - → A Public AMI : AWS provided
  - → Your own AMI : You make & maintain them yourself
  - → An AWS Marketplace AMI : an AMI someone else made (& potentially sells)

AMI Process

- Start an EC2 instance & customize it
- Stop the instance (for data integrity)
- Build an AMI → this will also create EBS snapshot
- Launch instances from other AMIs.



Create AMI Yourself

Create an instance, place the below code in user data (Advanced Settings)

```
#!/bin/bash
#Use this for your user data (script top to bottom)
#Install httpd (Linux & version)
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
```

(Installs Apache Server)

Instance will run through EC2 script. Wait for some time to copy paste Public IPv4 because we need to give it sometime even if it says running for EC2 user data script to run for first time.

AMI - Save the state for our EC2 instance

Right Click the instance > Image & Template > Create Image

name: Demo Image

Create Image.

AMIs > Demo Image is being created

Launch Instances from AMI

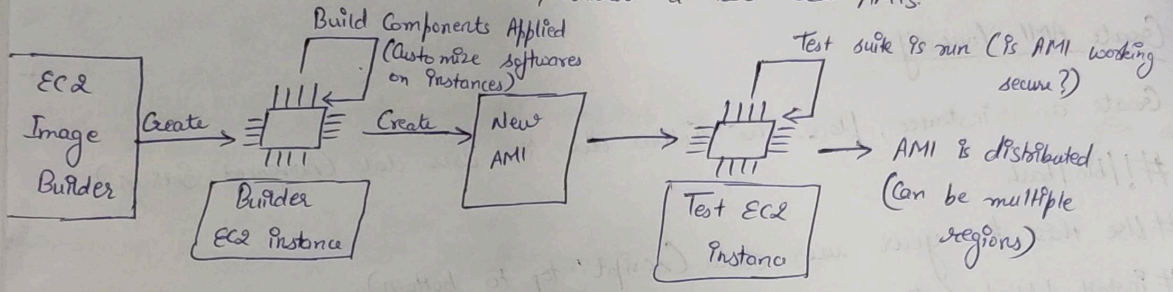
name: From AMI

AMI > My AMI's (Owned by me)

Search the Public IPv4 & its loaded very soon. (not installing httpd again)

EC2 Image Builder → Automation framework

- Used to automate the creation of virtual machines or container images
  ⇒ Automate the creation, maintain, validate & test EC2 AMIs.



- Free Service (only pay for the underlying resources)
  (like if we create an instance using this image builder, we are only going to pay for our instances)

EC2 Instance Store

- "EBS Volumes" are network drives with "good" but limited performance
  Sometimes we need even higher performance & that is going to be a hardware disk attached onto your EC2 instance
- If you need a high performance hardware disk use EC2 instance store
  ⇒ EC2 instance is a virtual machine but it is obviously attached to a real hardware server
- Better I/O performance
- If you stop or terminate EC2 instance, that has an Instance store, then the storage will be lost
  (∴ it is called ephemeral storage)
- Good for buffer / cache / scratch data / temporary content.

- For long term
- Risk of data
- Backups & Repli

If you see very
EC2 instances, co

EFS (Elastic

- Managed NFS
- EFS
- A

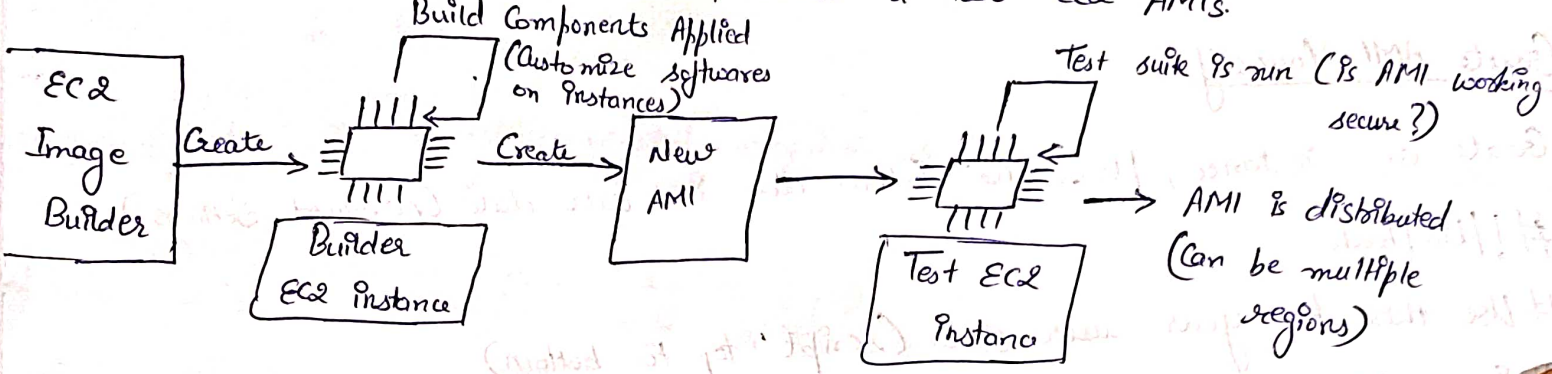
Instance

EBS vs EF

With EBS,
EBS volume
But if we
we could
availability
But this

EC2 Image Builder → Automation framework

⊙ Used to automate the creation of virtual machines or container images
⇒ Automate the creation, maintain, validate & test EC2 AMIs.



⊙ Free Service (only pay for the underlying resources) (like if we create an instance using this image builder, we are only going to pay for our instances)

EC2 Instance Store

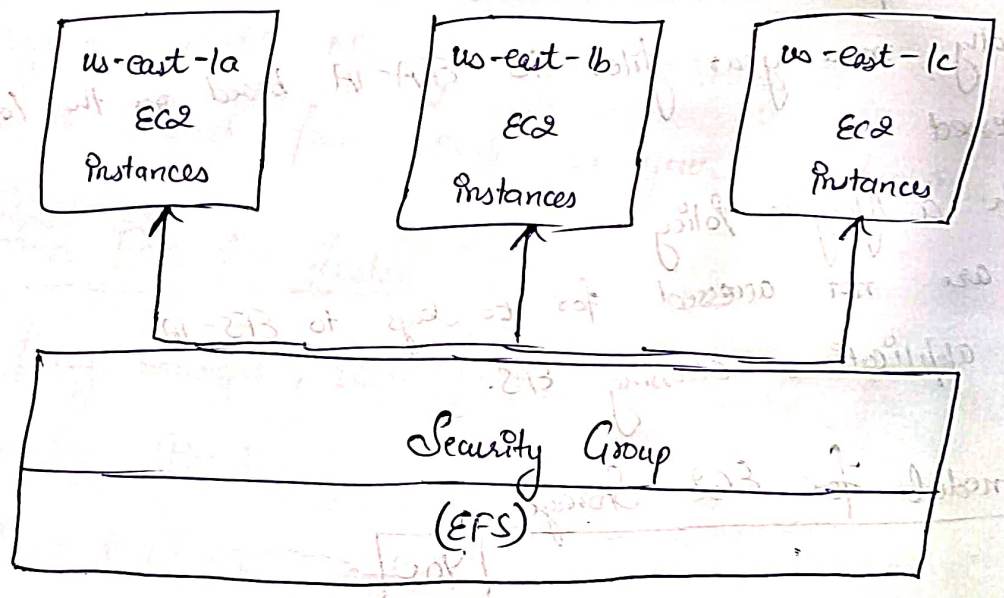
⊙ "EBS Volumes" are network drives with "good" but limited performance
Sometimes we need even higher performance & that is going to be a hardware disk attached onto your EC2 instance
⊙ If you need a high performance hardware disk use EC2 instance store
⇒ EC2 instance is a virtual machine but it is obviously attached to a real hardware server
⊙ Better I/O performance
⊙ If you stop or terminate EC2 instance, that has an Instance store, then the storage will be lost
(∴ it is called ephemeral storage)
⊙ Good for buffer / cache / scratch data / temporary content.

- For long term memory storage, EBS is a better choice
- Risk of data loss if hardware fails
- Backups & Replication are your responsibility

If you see very high performance hardware attached volume for your EC2 Instances, consider that it's Local EC2 instance store

EFS (Elastic File System)

- Managed NFS (Network File System) that can be mounted on 100s of EC2
- EFS works with Linux EC2 Instances in multi-AZ
- Highly available, scalable, expensive (3x gp2), pay per use, no capacity planning



EBS vs EFS

With EBS,

EBS volume can be attached to one specific Instance in one specific AZ

But if we wanted to move over EBS volume from one AZ to other, we could create a snapshot & then restore that snapshot into a new availability zone.

But this is a copy, this is not an in-sync replica.

EFS is a network file system

This means that whatever is on the EFS drive is shared by everything that is mounted to it. Say, we have many instances in AZ 1 on one or many instances in AZ 2.

At the same time, all these instances can mount the same EFS drive, using a mount target & they will all see the same files. So, this makes it a shared file system

EFS Infrequent Access (EFA-IA)

- Storage class that is cost optimized for files that are not accessed everyday
- Upto 98% lower cost compared to EFS standard
- EFS will automatically move your files to EFA-IA based on the last time they were accessed
- Enable EFS-IA with a lifecycle policy
- Eg → move files that are not accessed for 60 days to EFS-IA
- Transparent to the applications accessing EFS.

Shared Responsibility model for EC2 Storage

AWS	YOU
• Infrastructure	• Setting up backup / snapshot procedures
• Replication for data for EBS volumes & EFS drives	• Setting up data encryption
• Replacing faulty hardware	• Responsibility of any data on the drives
• Ensuring (data privacy) i.e employee: can't access your data	• Understanding the risk of using EC2 Instance store

- Launch 3rd party high performance file systems on AWS
- Fully managed service
- FSx for Lustre
- FSx for windows file server
- FSx for NetApp ONTAP

FSx for Windows File Server

- A fully managed, highly reliable & scalable windows native shared file system
- Built on Windows File Server
- Supports SMB protocol & Windows NTFS.
- Integrated with Microsoft Active Directory
- Can be accessed from AWS or your on-premises infrastructure

Amazon FSx for Lustre

- A fully managed, high performance, scalable file storage for high performance computing (HPC).
- Linux + Cluster → Lustre
- Machine learning, Analytics, Video processing, Financial Modeling
- Scales upto 100 Gb/s, millions of IOPS, sub-ms latencies

EC2 Instance Storage - Summary

- EBS Volume
  - → network drives attatched to one EC2 instance at a time
  - → mapped to an availability zone
  - → Can use EBS snapshots for backups / transfering EBS volumes across AZ

- AMI : Create ready to use EC2 instances with customizations
- EC2 Image Builder : Automatically build, test & distribute AMI's
- EC2 Instance Store
  - → High performance hardware disk attached to our EC2 instance
  - → Lost if our instance is stopped / terminated
- EFS : network file system, can be attached to 100s of instances in a region
- EFS - IA : cost optimized storage class for infrequent accessed files
- Fsx for windows → File system for windows server
- Fsx for Lustre → High performance computing Linux file system

Scalability & High Availability

- Scalability means that an application / system can handle greater loads by adapting
- 2 Kinds of scalability
  - Vertical Scalability
  - Horizontal "
- Scalability is linked but diff to High Availability

⇒ Vertical Scalability (Scale up / down)

- It means ↑ing the size of instance
- Eg: Your applications runs on t2.micro, scaling that application vertically means running it on a t2.large
- It is very common for non distributed systems such as database
- There is usually a limit to how much you can vertically scale

⇒ Horizontal Scalability (Scale out / in)

- It means ↑ing the no. of instances / systems for your applications
- Horizontal scaling implies distributed systems
- This is very common for web applications / modern applications
- It's easy to horizontally scale.

High Availability

- It usually goes hand in hand with horizontal scaling
- It means running your application in atleast 2 availability zones
- Its goal is to survive a data center loss (disaster)

High Availability & Scalability

- Vertical Scaling → ↑ Instance size (= scale up/down)
  - From: t2.nano - 0.5GB of RAM, 1 vCPU
  - To: u-12tb.1.metal - 12.3TB of RAM, 448 vCPUs
- Horizontal Scaling → ↑ no. of instances (= scale out/in)
  - → Auto Scaling group
  - → Load Balancer
- High Availability → Run instances for the same application across multiple AZ.

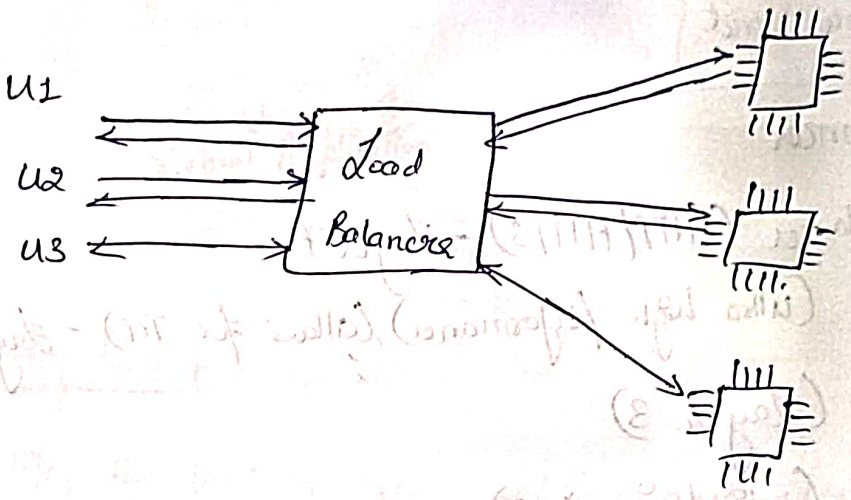
Scalability vs Elasticity

- Scalability → Ability to accomodate a larger load by making the hardware stronger (scale up) or by adding nodes (scale out)
- Elasticity → Once a system is scale, elasticity means that there will be some "autoscaling" so that system can scale based on load. This is "cloud-friendly": pay per use, match demand, optimize cost

- Agility - New IT resources are a click away, which means that you reduce the time to make those resources available to your developers from weeks to just minutes

What is Load Balancing ?

- Load Balancers are servers that forward Internet traffic to multiple servers (EC2 Instances) downstream



Why we use a Load Balancer ?

- Spread load across multiple downstream instances
- Expose a single point of access (DNS) to your application
- Seamlessly handle failures of downstream instances
- Do regular health checks to your instances
- Provide SSL termination (HTTPS) to your websites
- High availability across zones.

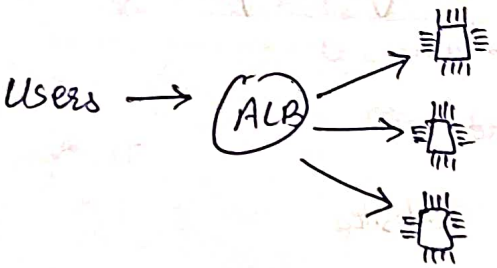
Why Use an Elastic Load Balancer?

⊙ An ELB is a manged load Balancer
- → AWS guarantee that it will be working
- → AWS take care of upgrades, maintainence, high availability
- → AWS provides only a few configuration knobs

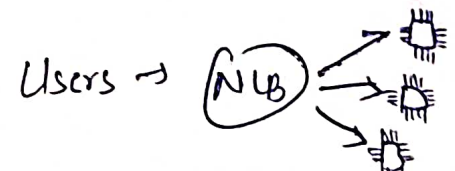
⊙ It costs less to setup your own load balancer but it will be a lot more effort on your end

⊙ 4 kinds of Load Balancer
- → Application load Balancer (HTTP/HTTPS) - Layer 7
- → Network " " (ultra high performance), (allows for TCP) - Layer 4
- → Gateway " " (Layer 3)
- → Classic " " (retired in 2023) - Layer 4 & 7

Application Load Balancer → ⊙ HTTP/HTTPS/gRPC protocols (Layer 7)
⊙ HTTP Routing features
⊙ Static DNS (URL)



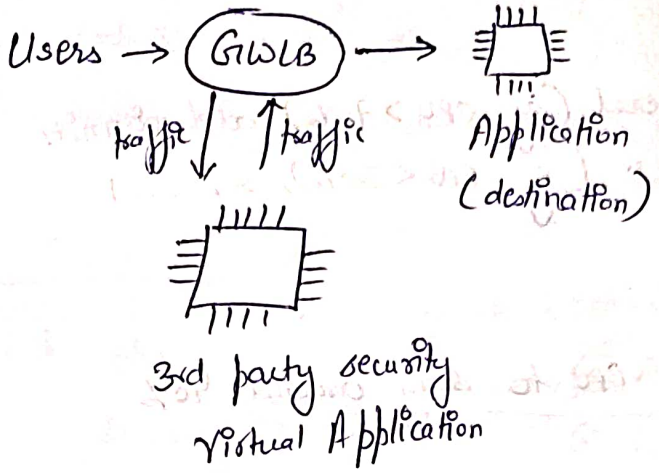
Network Load Balancer → ⊙ TCP/UDP protocols (Layer 4)
⊙ High performance : millions of request per second
⊙ Static IP through Elastic IP.



Gateway Load Balancer → ⊙ GENEVE protocol on IP packets (Layer 3)

⊙ Route traffic to firewalls that you manage on EC2 instances

⊙ Intrusion detection



Health Checks

They enable load balancer to know if instances are available to accept requests or not (2/3)

Auto-Scaling Group?

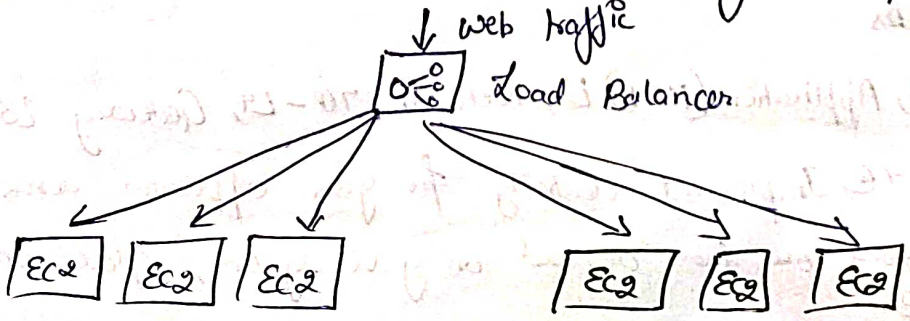
⊙ In real life, the load on your websites and application can change

⊙ In the cloud, you can create and get rid of servers very quickly

⊙ The goal of an ASG is to :-

- → Scale out (add EC2 instances) to match ↑ing load
- → Scale in (remove EC2 ") to match ↓ed load
- → Ensure we have minimum & max. no. of machines running
- → Automatically register new instances to load balancer
- → Replace unhealthy instances

⊙ Cost Savings : only run at optimal capacity. (principle of cloud)



Auto Scaling Groups – Scaling Strategies

- Manual Scaling → Update size of ASG manually
- Dynamic " → Respond to changing demand
  - → Simple/Step Scaling
    - When a CloudWatch is triggered (eg CPU > 70%), add 2 units
    - " " " " (eg CPU < 30%), remove 1
  - → Target Tracking Scaling
    - Eg → I want the avg. ASG CPU to stay around 40%
  - → Schedule Scaling
    - Anticipate a scaling based on known patterns
    - Eg → Increase the min capacity to 10 at 5pm on Fridays
- Predictive Scaling
  - → Use Machine Learning to predict future traffic ahead of time
  - → Automatically provisions to the right no. of EC2 instance in advance
  - → Useful when your load has predictable time based patterns

ELB and ASG – Summary

- High Availability vs Scalability (vertical & horizontal) vs Elasticity vs Agility in the Cloud
- Elastic Load Balancer
  - → Distributed traffic across backend EC2 instances, can be Multi-AZ
  - → Supports health checks
  - → 4 types : Classic (old), Application (HTTP) L7, Network TCP – L4, Gateway L3
- ASG (Auto Scaling groups) → Implement elasticity for your application across multiple AZ
  - Scale EC2 instances based on the demand on your system, replace unhealthy
  - Integrated with ELB

AMAZON S3

- Amazon S3 is one of the main building blocks of AWS
- Its advertised as "infinite scaling storage"
- Many websites use Amazon S3 as backbone
- Many AWS services use Amazon S3 as an integration as well

Amazon S3 Use Cases

- Backup & storage
- Disaster Recovery
- Archive
- Hybrid Cloud Storage
- Application Hosting
- Media Hosting
- Data lakes & big data analytics
- Software delivery
- Static Website

Amazon S3 - Buckets

- Amazon S3 allows people to store objects (files) in "buckets" (directories)
- Buckets must have a global unique name (across all regions all accounts)
- Buckets are defined at the region level
- S3 looks like a global service but buckets are created in a region
- Naming Convention
  - No upper case, No underscore
  - 3-63 characters long
  - Not an IP
  - Must start with lowercase letter or number
  - Must NOT start with the prefix xn--
  - Must NOT end with the suffix -s3alias

Amazon S3 - Objects

- Objects (files) have a key
- The key is the full path
  - s3://my-bucket/my-file.txt
  - s3://my-bucket/my-folder1/another-folder/my-file.txt
- The key is composed of prefix + object name
  - s3://my-bucket/my-folder1/another-folder/my-file.txt
- There is no concept of "directories" within buckets
- Just keys with very long names that contain slashes (/)
- Object values are the content of the body
  - Max Object size is 5TB (5000 GB)
  - If uploading more than 5GB, must use "multi part upload"
- Metadata (list of text key/value pairs - system or user data)
- Tags (Unique key/value pair - upto 10) - useful for security/lifecycle
- Version ID (if versioning is enabled)

Amazon S3 - Security

- User-Based
  - IAM-policies - which API calls should be allowed for a specific user from IAM
- Resource Based
  - Bucket policies - Bucket wide rules from S3 console - allows cross account
  - Object Access Control List (ACL) - finer grain (can be disabled)

→ Bucket Access Control List (ACL) - less common (can be disabled)

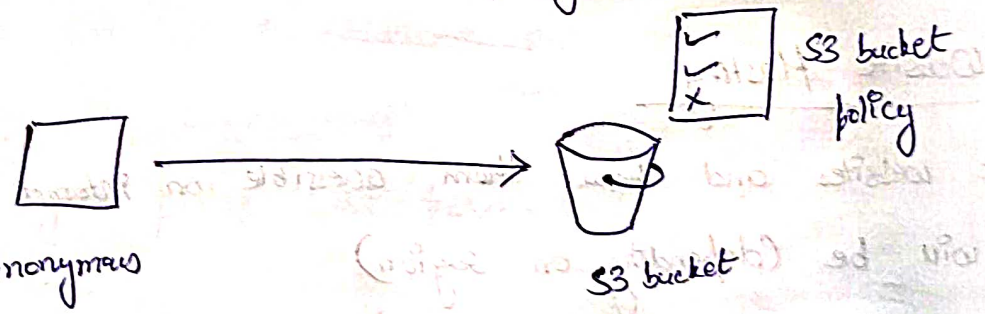
⊙ NOTE: an IAM principal can access an S3 object if
- → The user IAM permission ALLOW it OR the resource policy ALLOWS it
- → AND there is no explicit deny

⊙ Encryption: encrypt objects in S3 using encryption keys

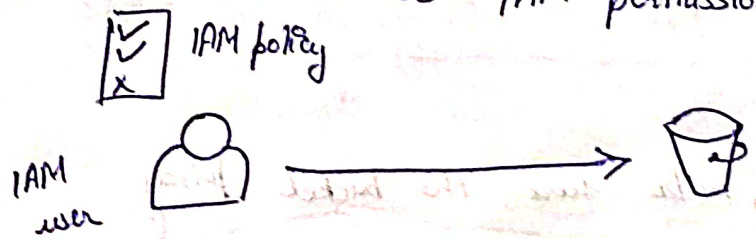
⊙ JSON - Based policy
- → Resources: buckets and objects
- → Effect: Allow / Deny
- → Actions: Set of API to Allow or deny
- → Principal: The account or user to apply the policy to

⊙ Use S3 bucket for policy to
- → Grant public access to the bucket
- → Force objects to be encrypted at upload
- → Grant access to another account (Cross account)

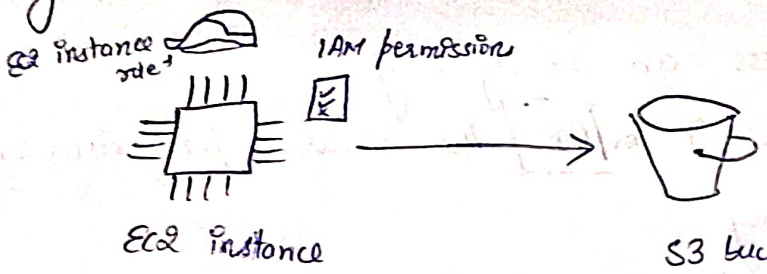
Eg: Public Access - Use Bucket policy



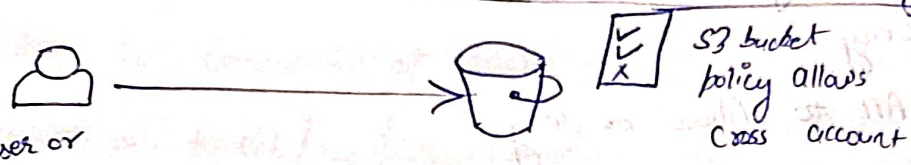
Eg → User access to S3 - IAM permissions



Eg → EC2 Instance Access - Use IAM Roles



Advanced : Cross-Account Access - Use Bucket Policy



Bucket settings for Block Public access

- These settings were created to prevent company data leaks
- If you know your bucket should never be public, leave these on
- Can be set at account level.

Amazon S3 - Static Website Hosting

- S3 can host static websites and have them accessible on Internet
- The website URL will be (depending on region)
  - http://bucket-name.s3-website-aws-region.amazonaws.com
    OR
  - _____.aws-_____
- If you get a 403 forbidden error, make sure the bucket policy allows public reads

Amazon S3 - Versioning

- You can version your files in Amazon S3
- It is enabled at bucket level
- Same key overwrite will change the "version": 1, 2, 3, ...
- It is best practice to version your buckets
  - → Protect against unintended delete (ability to restore a version)
  - → Easy roll back to previous version
- Notes:
  - → Any file that is not versioned prior to enabling versioning will have version "null".
  - → Suspending versioning does not delete previous versions

Amazon S3 - Replication (CRR & SRR)

CRR → Cross Region

SRR → Same region replication

- Must enable versioning in source & destination
- Buckets can be in different aws accounts
- Copying is asynchronous
- Must give proper IAM permissions to S3
- Use cases:
  - → CRR → compliance, lower latency access, replication across accounts
  - → SRR → log aggregation, live replication between production & test accounts

Creating Bucket in S3

- Create Bucket
- Upload an image
- When we click open, our image is displayed on the Internet
- But when we use 'Object URL', it does not work (access denied)

(S3 - pre-signed URL)

↳ A signature that verifies I am the one making the request & has my credentials in it

Versioning
- Deleting a version → Permanently delete (cant be undone)
- " an object → delete (delete marker) (checkpoint to roll back to in future)

S3 Storage Classes

- Amazon S3 Standard - General Purpose
- " " " - Infrequent Access (IA)
- " " one zone - Infrequent Access
- " " Instant retrieval
- " " Glacier flexible retrieval
- " " Glacier Deep Archive
- " " Intelligent Tiering.

- Can move b/w classes manually or using S3 lifecycle configurations.

S3 Durability & Availability

Durability → High durability (99.9999--- %) of object across multiple AZ

- If you store 10 million objects with Amazon S3, you can on average expect to incure a loss of a single object once every 10,000 years
- Durability represents the concept of how many objects will be lost by S3
- Same for all storage classes.

- Measures how readily a servia is
- Varies depending on storage class.
- Eg → S3 standard has 99.99% availability = not available for 53 minutes a year.

- 99.99 % availability
- Used for frequently accessed data
- Low latency & high throughput
- Sustain 2 concurrent facility failures
-

Big data, Analytics, mobile & gaming applications, content distribution

S3 Standard - Infrequent Access (IA)

- For data that is less frequently accessed, but requires rapid access when needed.
- Lower cost than S3 Standard
- 99.99% availability
- Use Cases → Disaster recovery, backups

S3 One-Zone - Infrequent Access

- High durability (99.999999%) in a single AZ, data lost when AZ is destroyed
- 99.5% availability
- Use Cases → Storing secondary backup copies of on-premises data or data you can re-create

S3 Glacier Storage Classes

- Low-cost object storage meant for archiving / backup
- Pricing: price for storage + object retrieval cost
- Amazon S3 Glacier Instant Retrieval
  - ↳ (I want data instantly)
  - → Millisecond retrieval, great for data accessed once a quarter
  - → Minimum storage duration of 90 days.

(I want data within some time)

- Amazon S3 Glacier Flexible Retrieval (formerly S3 glacier)
  - → Expedited (1 to 5 mins), Standard (3 to 5 hrs), Bulk (5 to 12 hrs) — free
  - → Minimum storage duration is 90 days

⊙ Amazon S3 Glacier Deep Archive - for long term storage
(I can wait to recieve data for 12hrs)

→ Standard (12hrs), Bulk (48 hrs) 24hrs
→ Minimum storage duration of 180 days
→ Least expensive

S3 Intelligent - Tiering

⊙ Small monthly monitoring & auto-tiering fee
⊙ Moves objects automatically b/w Access Tiers based on usage
⊙ There are no retrieval 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 (automatic) : objects not accessed for 90 days
⊙ Archive Access Tier (optional) : configurable from 90 days to 700 days
⊙ Deep Archive to Access Tier (optional) : configurable from 180 days to 700 days

Management > Lifecycle Rules > Set automated storage classes (acc to certain rules we set)

S3 Encryption

⊙ Server Side Encryption (default)

By default, applied, whenever we create a bucket, it will be encrypted
A user uploads an object in S3, that object when it arrives in the bucket is encrypted by S3 for securing purposes
Server encrypts the file after recieving it.

- Client Side Encryption

  When a user takes a file & encrypts it before uploading to the server

IAM Access Analyzer for S3

- Ensures that only intended people have access to your S3 buckets
- Eg → Publicly accessible buckets, bucket shared with other AWS account ...
- Evaluates S3 Bucket policies, S3 ACLs, S3 Access Point Policies
- Powered by IAM Access Analyzer

Shared Responsibility Model for S3

AWS	YOU / USER
• Infrastructure (global security, durability, availability, sustainability, sustain concurrent loss of data in two facilities)	• S3 versioning • S3 bucket policies • S3 Replication setup • Logging & monitoring • S3 storage classes • Data encryption at rest & in transit.
• Configuration & vulnerability analysis	
• Compliance Validation	

AWS Snow family

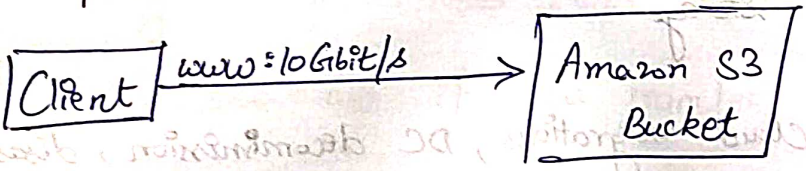
- Highly secure, portable devices to collect & process data at the edge, and migrate data into & out of AWS
- Data migration : Snow cone, Snowball edge, Snow mobile
- Edge computing : Snow cone, Snowball Edge

Data Migration with AWS Snow Family

Challenges &
- Limited Connectivity
- Limited Bandwidth
- High network cost
- Shared Bandwidth (cant maximise 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

- Direct upload to S3



- With Snow family

Client request Snowball which is delievered by AWS. We load the data directly onto the devices locally, and then we ship back the device to AWS and then they will plug that device into their infrastructure to import / export file from / to Amazon S3 bucket

Snow Ball Edge (for data transfer)

- Physical data transfer solution : move TBs or PBs of data in or out of AWS
- Alternating to moving data over the network (& paying network fees)
- Pay per data transfer job
- Provide block storage and Amazon-S3-compatible object storage
- Snowball Edge Storage Optimized
  - → 80 TB of HDD capacity for block volume & S3-compatible object storage
- Snowball Edge Compute Optimized
  - → 42 TB of HDD or 28 TB NVMe capacity for block volume & S3 compatible object storage
- Use Cases → Large scale cloud migrations, DC decommission, disaster management

AWS Snowcone & Snowcone SSD

- Small, portable computing, anywhere, rugged & secure, withstands harsh environments
- Light (4.5 pounds, 2.1 Kgs)
- Device used for edge computing, storage & data transfer
- Snowcone – 8TB of HDD storage
- Snowcone SSD – 14 TB of SSD 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 & use AWS DataSync to send data

AWS SnowMobile

- Transfer exabytes of data (1 EB = 1000 PB = 1,000,000 TBs)
- Each snowmobile has 100 PB of capacity (use multiple in parallel)
- High Security : Temp. Controlled, Gps, 24/7 video surveillance
- Better than snowball if you transfer more than 10PB.

	Snowcone & Snowcone SSD	Snowball Edge Storage Optimized	Snow mobile
Storage Capacity	8TB of HDD 14TB of SSD	80 TB usable	> 100 PB
Migration size	upto 24 TB, online & offline	upto Petabytes, offline	upto exabytes, offline
Data Sync agent	Pre-Installed		

Snowball Family : Usage Process

1. Request Snowball devices from the AWS console for delivery
2. Install the Snowball Client / AWS OpsHub on your servers
3. Connect the Snowball to your servers & copy files using the client
4. Ship back the device when you are done (goes to right AWS facility).
5. Data will be loaded into an S3 Bucket
6. Snowball is completely wiped

What is Edge Computing?

- Process data while its being created on an edge location.
  Eg → A truck on the road, a ship on the sea etc
- These locations 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 :
  - → Pre process 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)

Snow Family - Edge Computing

- Snowcone & Snowcone SSD (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
  - 104 vCPUs, 416 GiB of RAM
  - Optional GPU (useful for video processing or machine learning)
  - 28 TB NVMe or 42 TB HDD usable storage
  - Storage clustering available (upto 16 nodes)
- Snowball Edge - Storage Optimized
  - Upto 40 vCPUs, 80 GiB of RAM, 80 TB storage
- All: Can run EC2 instances & AWS Lambda functions (using AWS IoT Greengrass)
- Long term deployment options → 1 & 3 years discounted pricing

AWS OpsHub

- Historically, to use snow family devices, you need CLI
- Today, you can use AWS OpsHub (a software you install on your device) to manage your snowfamily device
  - → Unlocking & configuring single or clustered devices
  - → Transferring files
  - → Launching & managing instances running on snowfamily devices
  - → Monitor devices metrics (storage capacity, active instances on your device)
  - → Launch compatible AWS services on your devices (Eg → Amazon EC2 instances, AWS DataSync, Network File System (NFS))

Snowball Edge Pricing

- You pay for device usage & data transfer out of AWS
- Data transfer IN to Amazon S3 is $0.00 per GB
- On-Demand → Includes a one time service fee per job, which includes
  - → 10 days of usage for snowball Edge storage optimized 80TB
  - → 15 days of usage for snowball Edge storage optimized 210TB
  - → Shipping days are not counted towards the included 10 or 15 days
- Committed Upfront → Pay in advance for monthly, 1-year & 3 years of usage (Edge-computing)
  - → Upto 62% discount pricing

Hybrid Cloud for Storage

- AWS is pushing for "hybrid Cloud"
  - → Part of your infrastructure is on-premises
  - → " " " " on the cloud
- This can be due to
  - → Long Cloud migrations
  - → Security Requirements
  - → Compliance "
  - → IT Strategy
- S3 is a proprietary storage technology (unlike EFS/NFS). So, how do you expose S3 data on premises?

(AWS Storage Gateway)

AWS Storage Cloud Native Options

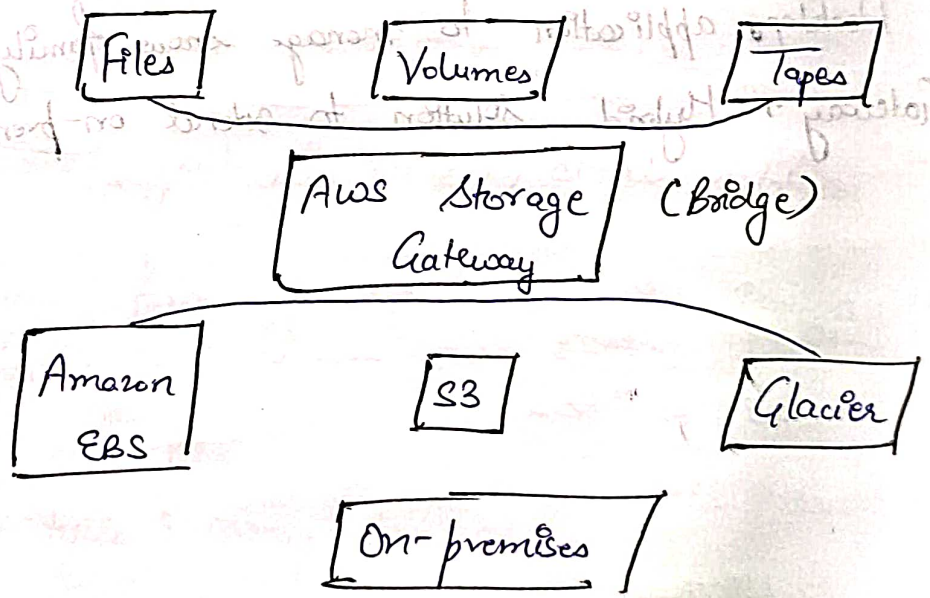
Block → Amazon EBS, EC2 Instance Store

File → Amazon EFS

Object → Amazon S3, Glacier

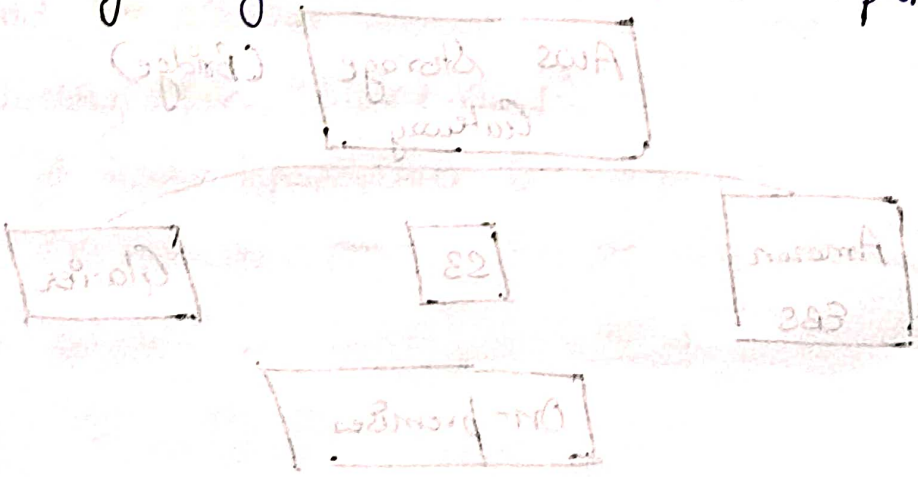
AWS Storage Gateway

- Bridge btw on-premises data & cloud data in S3
- Hybrid storage service to allow on premises to seamlessly use AWS clou
- Use Cases → Disaster recovery, backup & restore, tiered storage
- Types of Storage Gateways
  - File Gateway
  - Volume Gateway
  - Tape Gateway



S3 - Summary

- Buckets vs Objects → Buckets have a globally unique name, tied to a region
  Object lives within buckets
- S3 Security → IAM policy, S3 Bucket policy (Public access), S3 Encryption
- S3 Websites → Host a static website on Amazon S3. (ensuring our bucket is public)
- S3 versioning → Multiple versions for files, prevent accidental deletes
- S3 Replication → Same region or cross region, must enable versioning
- S3 Storage classes → Standard, IA, IZ-IA, Intelligent, Glacier (Instant, Flexible, Deep)
- Snow family → Import data on S3 through a physical device, edge computing
- Ops-Hub → Desktop application to manage snow family devices
- Storage Gateway → Hybrid solution to extend on-premises storage to S3



Databases

- Storing data on disk (EFS, EBS, Instance Store, S3) can have limits

Relational Database

- Excel spreadsheets, with links between them
- Can use SQL language to perform queries

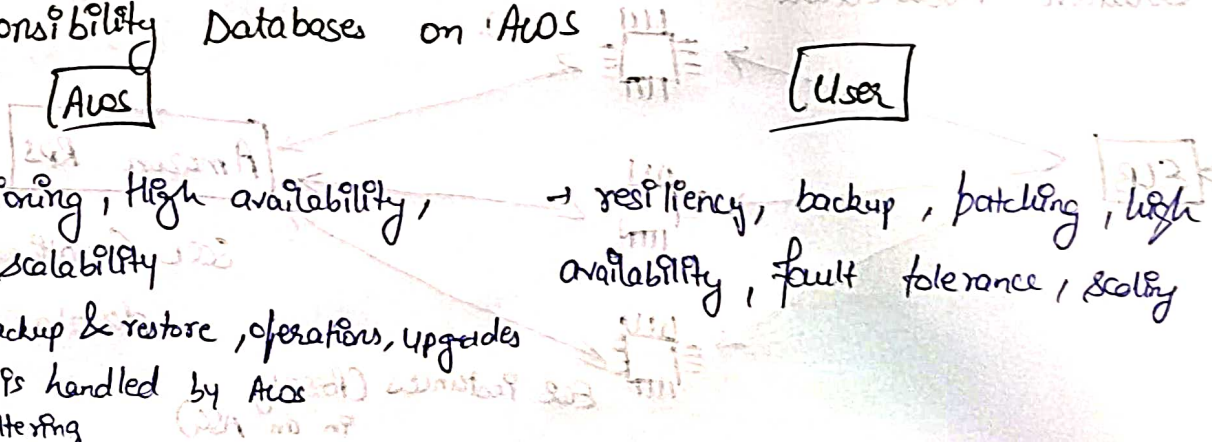
NoSQL Databases

- Non-relational DB's
- These are purpose built for specific data models and have flexible schemas for building
- Benefits
  - → Flexibility
  - → Scalability
  - → High performance
  - → Highly functional

Eg → key-value, document, graph, in-memory, search databases

- JSON is a common form of data that fits into a NoSQL model
- Data can be nested
- Fields can change over time
- Support for new types : arrays etc

Shared Responsibility Databases on AWS



→ Quick provisioning, High availability, V/Horizontal scalability
→ Automated backup & restore, operations, upgrades
→ OS patching is handled by AWS
→ Monitoring, altering

→ resiliency, backup, patching, high availability, fault tolerance, scaling

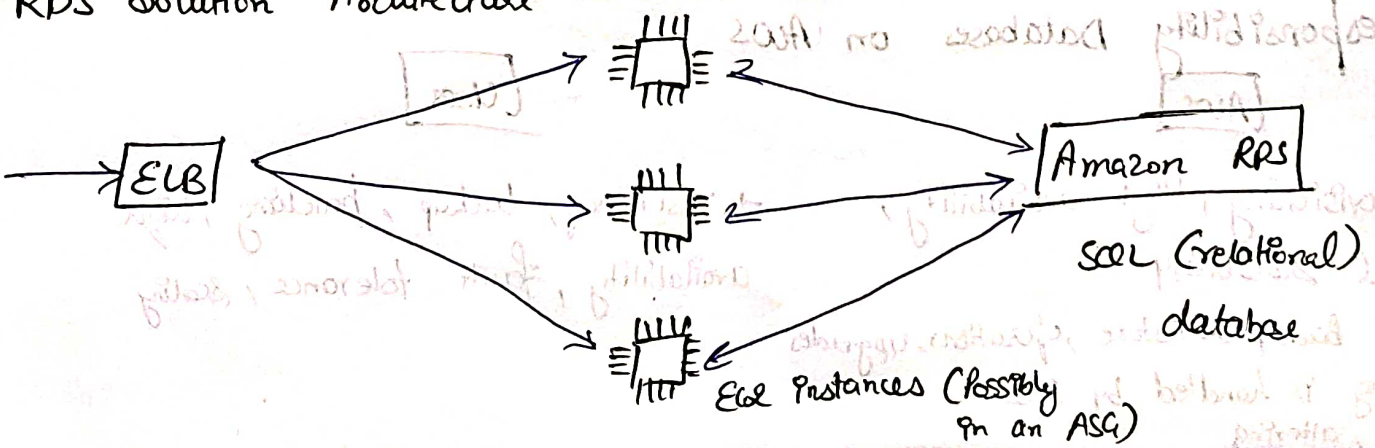
Amazon RDS Overview

- Relational Database Service (RDS)
- DB that use SQL as a query language
- Allows us to create databases in cloud that are managed by AWS
  - Postgres
  - MySQL
  - MariaDB
  - Oracle
  - Microsoft SQL Server
  - IBM DB2
  - Aurora (AWS Proprietary Database)

Why RDS Over Deploying DB on EC2?

- Managed Service
  - → Automated provisioning, OS patching
  - → Continuous backups & restore to specific timestamp
  - → Monitoring dashboards
  - → Read replicas for improved read performance
  - → Multi AZ setup for disaster recovery (DR)
  - → Maintainance windows for upgrades
  - → Vertical / Horizontal Scaling
  - → Storage backed by EBS
- But you can't SSH into your instance

RDS Solution Architecture



Amazon Aurora

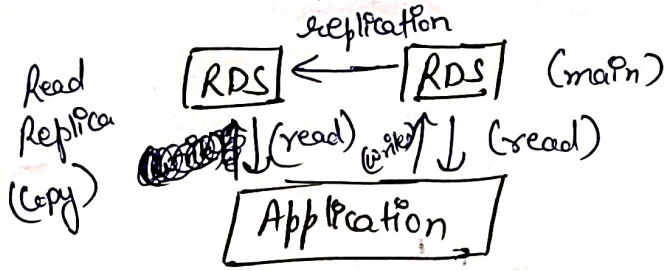
- Aurora is a proprietary technology from AWS (not open-sourced)
- Postgre SQL and MySQL are both supported as Aurora DB
- Aurora is "AWS Cloud Optimized" & claims 5x performance improvement over MySQL on RDS, over 3x performance of Postgres on RDS
- Aurora storage automatically grows in increments of 10GB, upto 128TB
- Costs more than RDS (20% more) - more efficient
- Not in free-tier

Amazon Aurora Serverless

- Automated DB instantiation and auto scaling based on actual usage
- Postgre & MySQL both supported
- No capacity planning needed
- Least management overhead
- Pay per second, can be more cost-effective
- Use Cases → good for infrequent, intermittent or unpredictable workloads

RDS Deployments : Read Replicas, Multi AZ

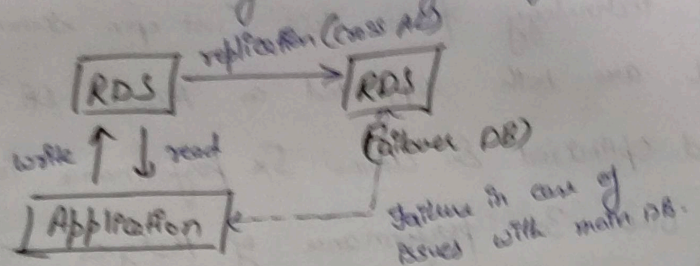
- Read Replicas
  - Scale the read workload of your DB
  - Can create upto 15 Read replicas



(Applications can read from any RDS)
(Application only have to write to main RDS)

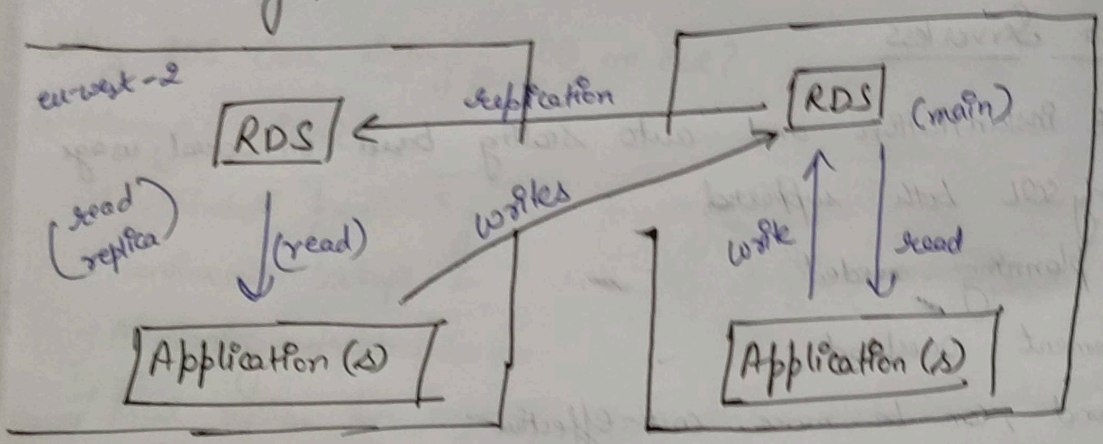
Multi-AZ

- Failure in case of AZ outage (high availability)



- Data is only read/written to the main DB
- Can only have 1 AZ as failover

Multi-Region



Multi-Region (Read Replica)

→ Disaster recovery in case of region issues
→ Local performance for global reads
→ Replication cost

Amazon

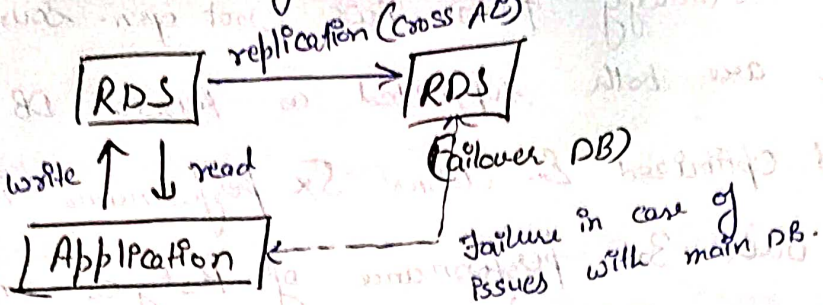
- Ela
- Caches
- Helps
- AWS tak
  Configurati

Solution

ELB

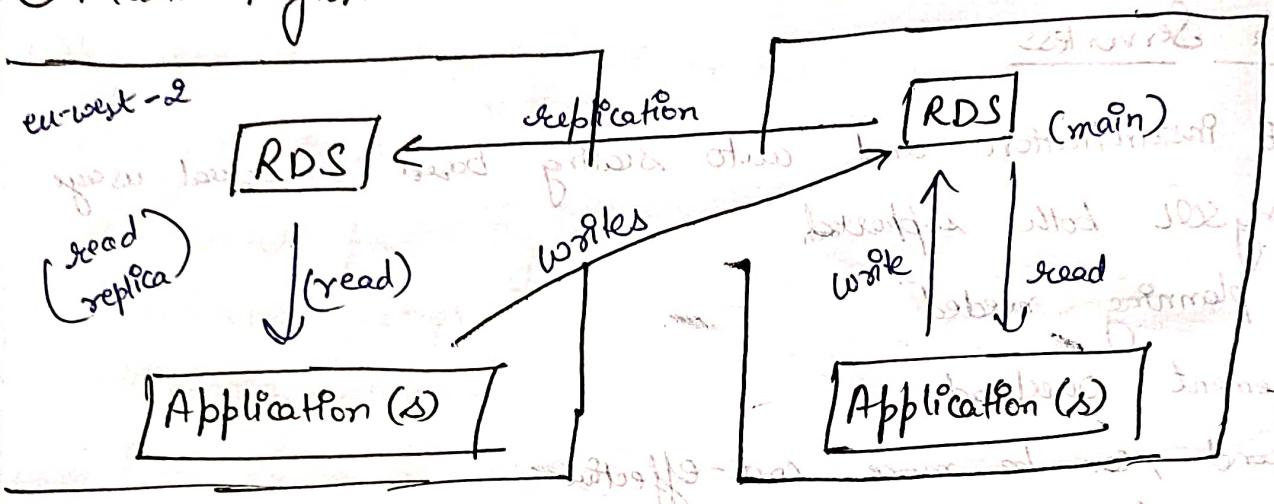
Multi - AZ

- Failure in case of AZ outage (High availability)



- Data is only read/written to the main DB
- Can only have 1 AZ as failover

Multi - Region



Multi-Region (Read Replica)

→ Disaster Recovery in case of region issues
→ Local performance for global reads
→ Replication cost

Amazon ElastiCache Overview

- Elasticache is to get managed Redis or Memcached
- Caches are in-memory databases with high performance, low latency
- Helps reduce load off databases for read-intensive workloads
- AWS takes care of OS maintainence/patching, optimizations, setup, configuration, monitoring, failure recovery & backups

Solution Architecture - Cache