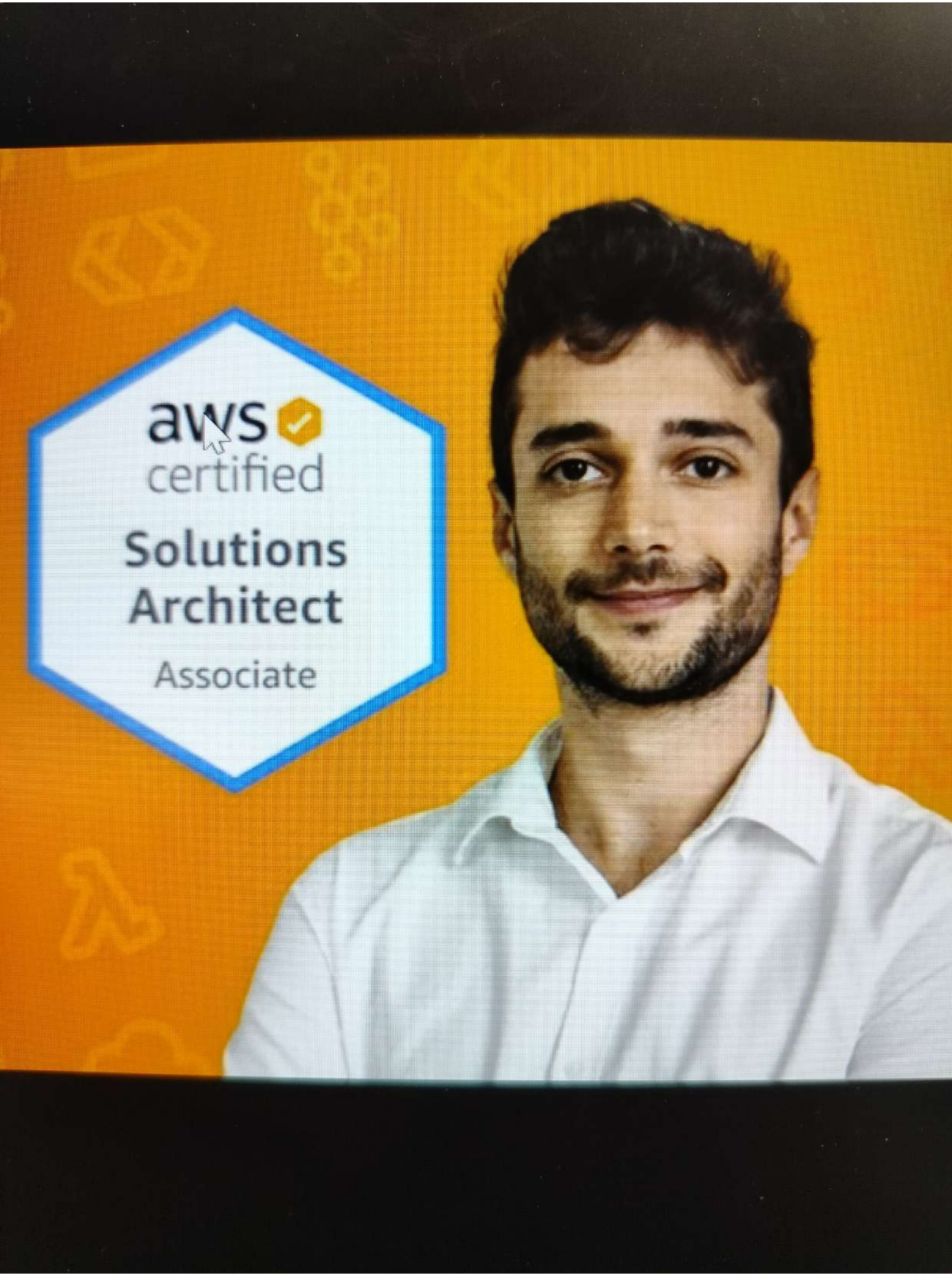

aws
certified
Solutions
Architect
Associate

EC2 - Elastic Compute Cloud

Types of IP

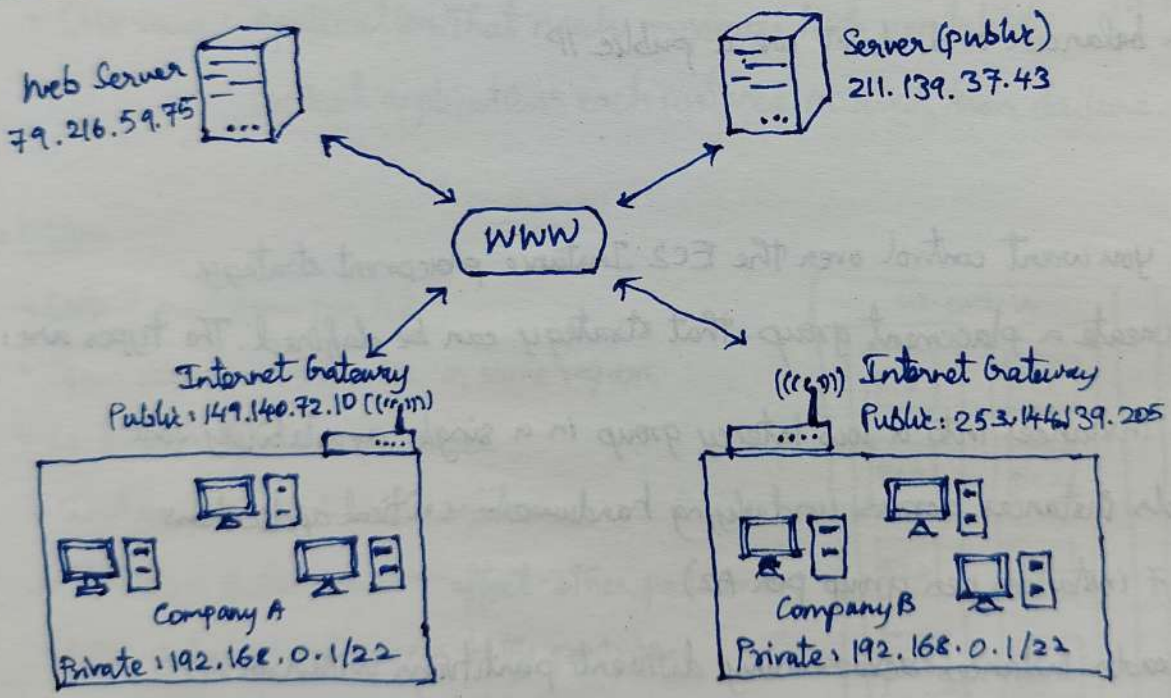
a) IPv4

- Most commonly used format.
- Eg: 1.160.10.240
- Allows 3.7 billion different addresses in the public space.
- Range : [0-255]-[0-255].[0-255].[0-255]

b) IPv6

- Newest and solves problems for the Internet of Things (IOT)
- Eg: 3ffe:1900:4545:3:200:f8ff:fe21:67cf

Private vs Public IP



Public IP

- Can be identified on the internet
- Must be unique across whole web
- Two machines cannot have same IP
- Can be geo-located easily

Private IP

- Can be identified on private network only
- Must be unique across private network
- 2 different private networks (2 companies) can have same IPs
- Machines connect to www using Internet gateway
- Only specified range can be used as private IP

Elastic IPs

- When you stop and then start an EC2 instance, it can change its public IP
- If you need to have a fixed public IP for your instance, you need an Elastic IP
- It is a public IPv4 IP. you own as long as you don't delete it.
- You can attach to one instance at a time only
- You can mask the failure of an instance or software by rapidly remapping the address to another instance in your account.
- You can have 5 Elastic IP in your account (you can ask AWS to increase)

Avoid using Elastic IP:

- They often reflect poor architectural decisions.
- Instead, use a random public IP and register a DNS name to it.
- Use a load balancer and don't use a public IP

Placement Groups

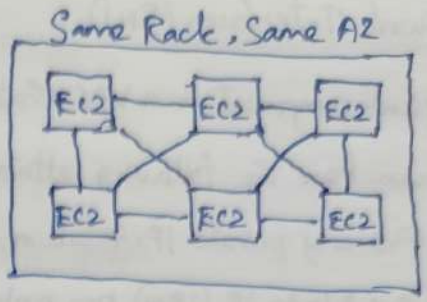
- Sometimes you want control over the EC2 Instance placement strategy.
- When you create a placement group, that strategy can be defined. The types are:

a) Cluster → clusters instances into a low latency group in a single availability zone.

b) Spread → spreads instances across underlying hardware - critical applications
(max 7 instances per group per AZ)

c) Partition → spreads instances across many different partitions within an AZ
Scales to 100s of EC2 instances per group (Hadoop, Cassandra, Kafka)

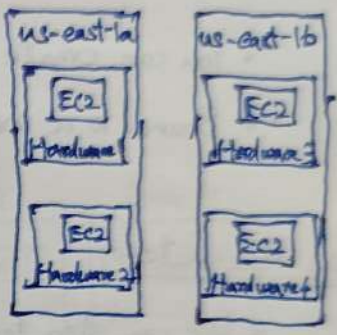
a) Cluster



- Pros: Great network (10 Gbps bandwidth)
- Cons: If rack fails, all instances fail
- Use case:
  - Big data job that needs to complete fast
  - Application that needs extremely low latency and high network throughput

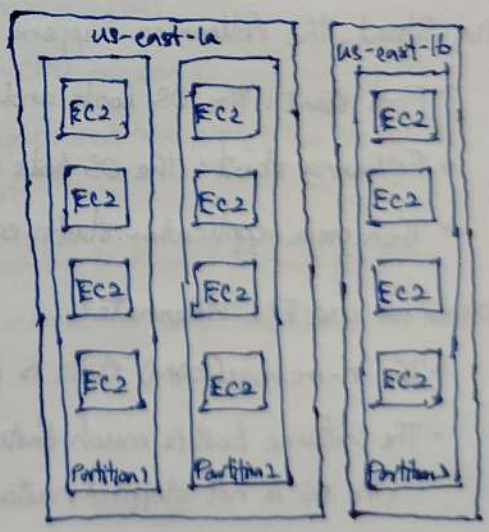
b) Spread

- Pros: Span across multiple AZ
  Reduced risk in simultaneous failure
  EC2 instances are on different physical hardware
- Cons: Limited to 7 instances per AZ per placement group
- Use case: Application that needs maximize high availability
  Critical applications each instance isolated from failure.



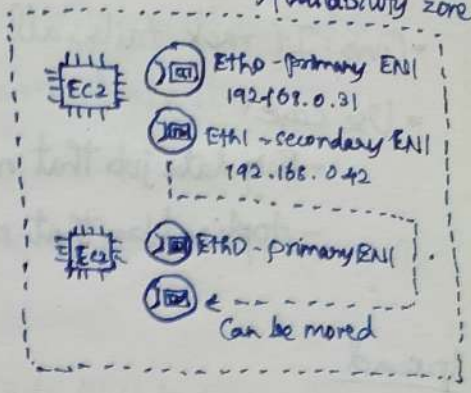
c) Partition

- Upto 7 partitions per AZ
- Span across multiple AZ in same region
- Upto 100s of EC2 instances
- Instances do not share racks
- Partition failure won't affect other partitions
- EC2 instances get access to the partition information as metadata
- Use cases: HDFS, HBase, Cassandra, Kafka



Elastic Network Interfaces (ENI)

- Logical component in a VPC that represents a virtual network card
- It can have the following attributes:
  - Primary private IPv4, one or more secondary IPv4
  - One Elastic IP (IPv4) per private IPv4
  - One Public IPv4
  - One or more security groups
  - A MAC address



- You can create ENI independently and attach them on the fly (move them) on EC2 instances (ba
- Bound to a specific availability zone (AZ)

EC2 Hibernate

- We can stop, terminate instances
  - Stop: the data on disk (EBS) is kept intact in the next start
  - Terminate: any EBS volumes (root) also set-up to be destroyed is lost
- On Start, the following happens
  - First start: the OS boots and the EC2 User Data script is run
  - Following start: the OS boots up
  - Then your application starts, caches get warmed up and that can take time
- When we use EC2 Hibernate:
  - The in-memory (RAM) state is preserved
  - The instance boot is much faster
    (the OS is not stopped/restarted)
  - RAM state is written to a file in root EBS volume
  - The root EBS volume must be encrypted
- Use Cases:
  - long-running processing
  - saving the RAM state
  - services that take time to initialize

Note:

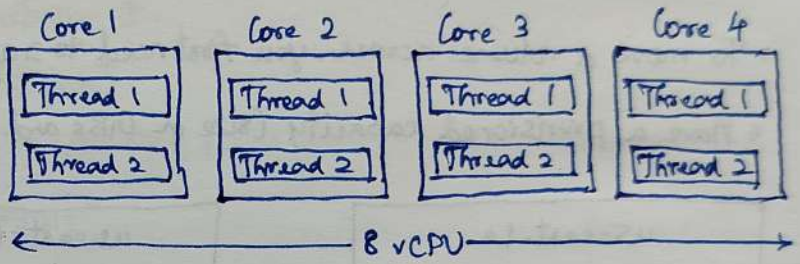
- Instance RAM size < 150 GB
- Available for on-demand & Reserved instances
- Instance cannot be hibernated > 60 days

EC2 Nitro

- Underlying platform for the next generation of EC2 instances.
- New virtualization technology.
- Allows for better performance
  - Better networking options (enhanced networking, HPC, IPv6)
  - Higher Speed EBS (Nitro is necessary for 64000 EBS IOPS - max 32000 on non-Nitro)
- Better underlying security.

EC2 vCPU

- Multiple threads can run on one CPU (multithreading)
- Each thread is represented as a virtual CPU (vCPU)
- Eg: m5.2xlarge
  - 4 CPU
  - 2 threads per CPU
  - 8 vCPU in total



Optimizing CPU

- EC2 instances come with a combination of RAM and vCPU
- But in some cases, you may want to change the vCPU options:
  - # of CPU cores: you can decrease it (if you need high RAM & low No. of CPU)
    - to decrease licensing costs
  - # of threads per core: disable multithreading to have 1 thread per CPU
    - helpful for high performance computing (HPC) workloads
- Only specified during instance launch.

EC2 Capacity Reservations

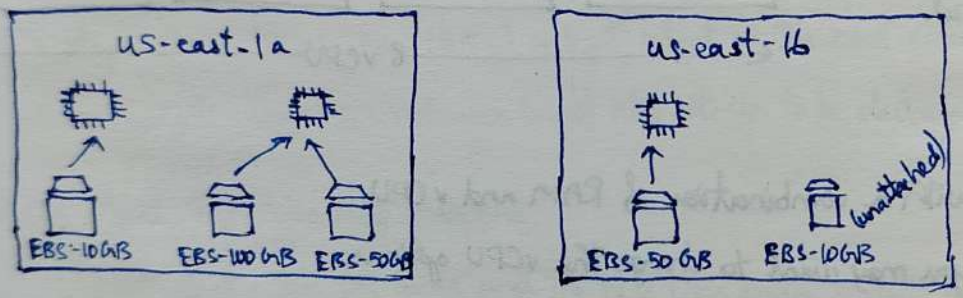
- Ensure you have EC2 capacity when needed
- Manual or planned end-date for reservation
- No need for 1 or 3-year commitment
- Capacity access is immediate
- You get billed as soon as it starts
- Combine with Reserved Instances and Savings plans to do cost saving.

Specify:

- Availability zone to reserve capacity.
- No. of instances to reserve capacity.
- Instance attributes, instance type, tenancy and platform/OS

EBS Volume

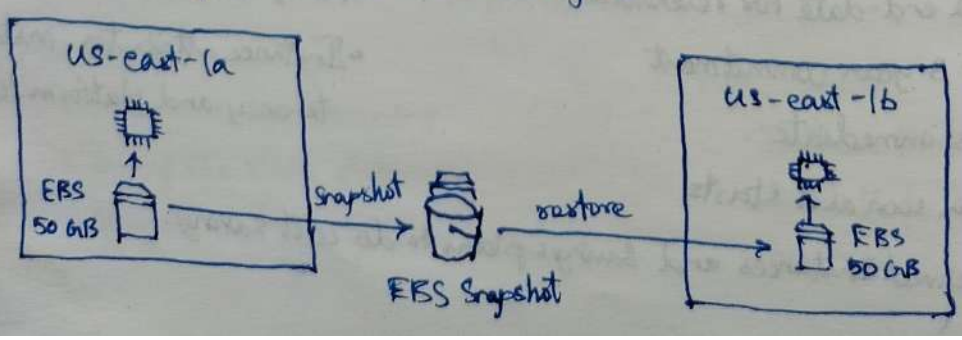
- An Elastic Block Store volume is a network drive you can attach to your instances.
- It allows your instances to persist data, even after their termination.
- They can only be mounted to one instance at a time. Multi-attach for some EBS
- They are bound to a specific availability zone.
- Analogy: Think of them as a 'Network USB stick'
- Free Tier: 30 GB of free EBS storage of General Purpose (SSD) or Magnetic per month
- It is a network drive (i.e. not a physical drive). It uses the network to communicate the instance, which means there might be a bit of latency
- It can be detached from an EC2 instance and attached to another one quickly.
- To move a volume across, you first need to snapshot it.
- Have a provisioned capacity (size in GBs and IOPS)



- Delete on termination attribute controls the EBS behaviour when an EC2 instance terminat
- By default root volume is deleted. Other EBS volumes attribute is not marked
- Use case: preserve root/EBS volume when instance is terminated.

EBS Snapshots

- Make a backup (snapshot) of EBS volume at a point in time.
- Not necessary to detach volume to do snapshot but recommended.
- Can copy snapshots across AZ or region

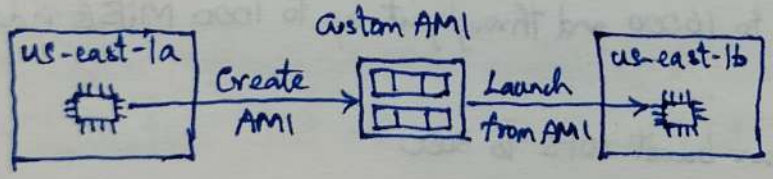


AMI Overview

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

AMI Process [from EC2 instance]

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



EC2 Instance Store

- EBS volumes are network drives with good but "limited" performance.
- If you need a high-performance hardware disk, use EC2 Instance Store.
- Better I/O performance
- EC2 instance store lose their storage if they're stopped (ephemeral)
- Good for buffer/cache/scratch data/temporary content
- Risk of data loss if hardware fails
- Backups and Replication are your responsibility

EBS Volume Types

i) gp2/gp3 (SSD) → General purpose SSD. balances price & performance for wide variety.
ii) io1/io2 (SSD) → Highest performance SSD, mission critical low-latency or high throughput
iii) st1 (HDD) → Low cost HDD volume designed for frequently accessed, intensive workload
iv) sc1 (HDD) → Lowest cost HDD volume designed for less frequently accessed workloads

- EBS volumes are characterized in Size | Throughput | IOPS (I/O per sec)
- Only gp2/gp3 and io1/io2 can be used as boot volumes.

i) General Purpose SSD

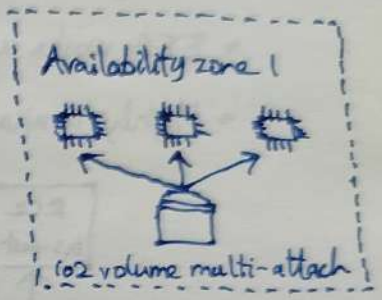
- Cost effective storage, low latency.
- System boot volumes, Virtual desktops, Development and test environments
- 1 GiB - 16 TiB
- gp3
  - Baseline of 3000 IOPS and throughput of 125 MiB/s
  - Can increase IOPS up to 16000 and throughput up to 1000 MiB/s independently
- gp2
  - Small gp2 volumes can burst IOPS to 3000
  - Size of the volume and IOPS are linked, max IOPS is 16,000
  - 3 IOPS per GiB, means at 5334 GiB we are at max IOPS

ii) Provisioned IOPS SSD

- Critical business applications with sustained IOPS performance or more than 16000 IOPS
- Great for databases workloads (sensitive to storage perf and consistency)
- io1/io2 (4 GiB - 16 TiB)
  - Max PIOPS: 64000 for Nitro EC2 instances & 32000 for other
  - Can increase PIOPS independently from storage size
  - io2 have more durability and more IOPS per GiB (at the same price as io1)
- io2 Block Express (4 GiB - 64 TiB)
  - Sub-millisecond latency
  - Max PIOPS: 256000 with an IOPS:GiB ratio of 1000:1
- Supports EBS Multi-attach

ii) Hard Disk Drives (HDD)

- Cannot be a boot volume.
- 125 MiB to 16TiB
- Throughput Optimized HDD (st1)
  - Big data, Data warehouses, Log processing
  - Max throughput 500 MiB/s - max IOPS 500
- Cold HDD (sc1)
  - For data that is infrequently accessed
  - Scenarios where lowest cost is important
  - Max Throughput 250 MiB/s - max IOPS 250



EBS Multi-Attach

- Exclusively for io1/io2 family
- Attach the same EBS volume to multiple EC2 instances in the same AZ
- Each instance has full read & write permission to the volume
- Use case:
  - Achieve higher application availability in clustered Linux application (Eg: Teradata)
  - Application must manage concurrent write operations
- Must use a file system that's cluster aware (not XFS, EXT4, etc...)

EBS Encryption

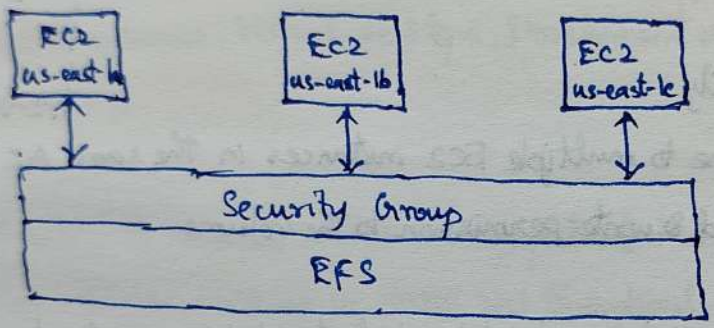
- Data at rest is encrypted inside the volume.
- All the data in flight moving between the instance and the volume is encrypted
- All snapshots are encrypted. All volumes created from the snapshot
- Encryption and decryption are handled transparently (you have nothing to do)
- Encryption has a minimal impact on latency.
- EBS Encryption leverages keys from KMS (AES-256)
- Copying an unencrypted snapshot allows encryption
- Snapshots of encrypted volumes are encrypted

Encrypt an unencrypted EBS volume

- Create an EBS snapshot of the volume.
- Encrypt the EBS snapshot (using copy)
- Create new EBS volume from the snapshot (the volume will also be encrypted)
- Now you can attach the encrypted volume to the original instance

EFS - Elastic File System

- Managed NFS (network file system) that can be mounted on many EC2
- EFS works with EC2 instances in multi-AZ
- Highly available, scalable, expensive (3 x gp2) - pay per use



- Use cases: content management, web serving, data sharing, Wordpress
- Uses NFSv4.1 protocol
- Uses security group to control access to EFS
- Compatible with Linux based AMI (not Windows)
- Encryption at rest using KMS
- POSIX file system (~Linux) that has a standard file API
- File system scales automatically, pay per use, no capacity planning.

EFS - Performance & Storage Classes

a) EFS Scale

- 1000s of concurrent NFS clients, 10 GB+/s throughput
- Grow to Petabyte-scale network file system, automatically.

b) Performance mode (set at EFS creation time)

- General purpose (default): latency-sensitive use cases (web server, CMS, etc...)
- Max I/O - higher latency, throughput, highly parallel (big data, media processing)

c) Throughput mode

- Bursting (1TB = 50 MiB/s + burst of up to 100 MiB/s)
- Provisioned: set your throughput regardless of storage size, ex: 1 GiB/s for 1TB storage

d) Storage Tiers (Lifecycle management feature - move file after N days)

- Standard: for frequently accessed files
- Infrequent access (EFS-IA): cost to retrieve files, lower price to store

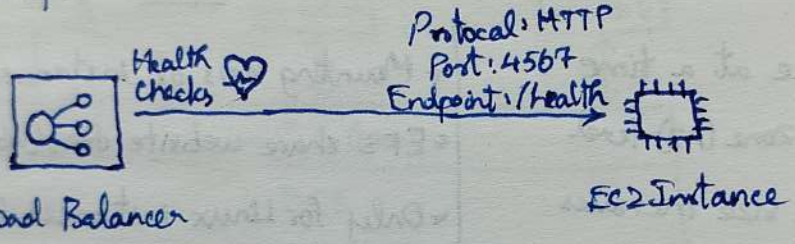
EBS	vs	EFS
• Can be attached to one instance at a time		• Mounting 100s of instances across AZ
• Locked at the Availability Zone (AZ) level		• EFS share website files (wordpress)
• gp2: IO increases if the disk size increases		• Only for Linux instances (POSIX)
• io1: can increase IO independently		• EFS has a higher price point than EBS
To migrate EBS volume across AZ		• Can leverage EFS-IA for cost savings
• Take a snapshot		
• Restore the snapshot to another AZ		
• EBS backups use IO and you shouldn't run them while application is handling lot of traffic		
• Root EBS volumes of instances get terminated by default if the EC2 instance gets terminated (you can disable that)		

ELB - Elastic Load Balancing

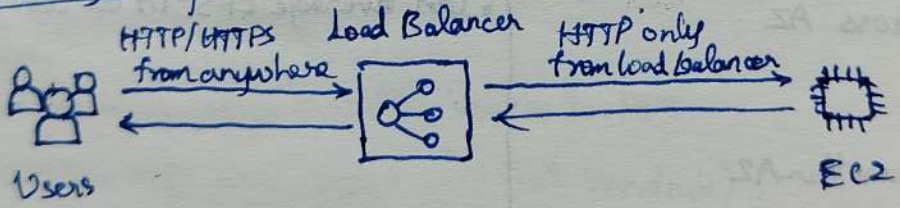
- An Elastic Load Balancer is a managed load balancer
- AWS takes care of upgrades, maintenance, high availability
- It costs less to setup your own load balancer but it will be a lot more effort on your end
- It is integrated with many AWS offerings/services

Health Checks

- Health checks are crucial for Load Balancers
- They enable the load balancer to know if instances it forwards traffic to are available to reply to requests.
- The health check is done on a port and a route (/health is common)
- If the response is not 200 (OK), then the instance is unhealthy

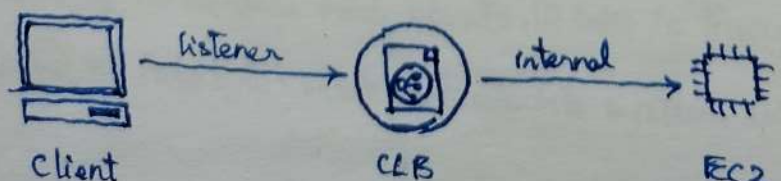


Load Balancer Security Groups



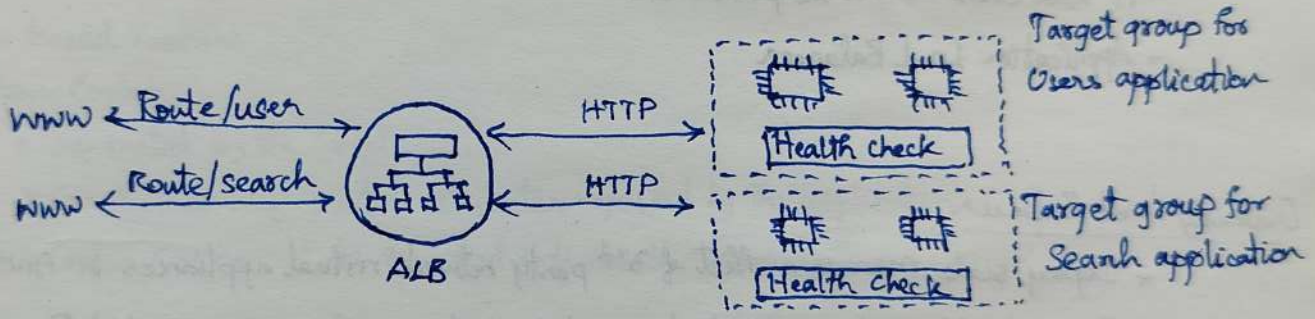
Classic Load Balancers

- Supports TCP (Layer 4) - HTTP & HTTPS (Layer 7)
- Health checks are TCP or HTTP based
- Fixed hostname XXX.region.elb.amazonaws.com



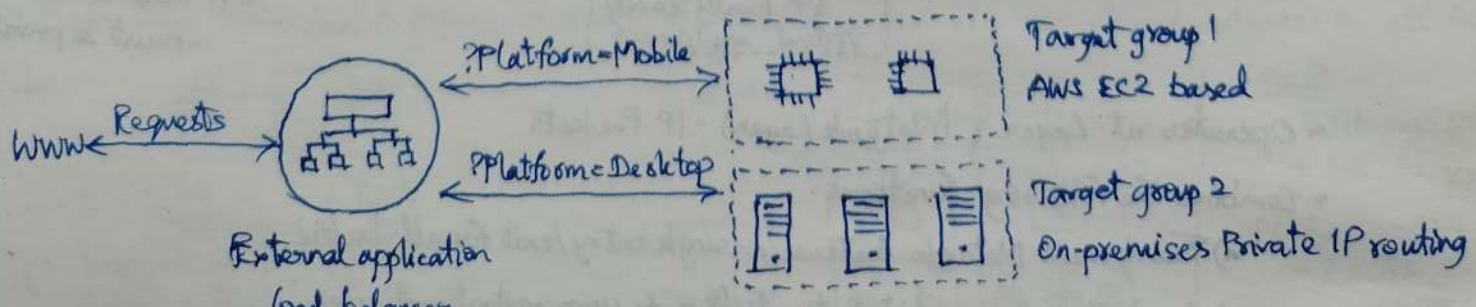
Application Load Balancer

- Application Load Balancer is Layer 7 (HTTP)
- Load balancing to multiple HTTP applications across machines (target groups)
- Load balancing to multiple applications on the same machine (Eg: containers)
- Support for HTTP/2 and WebSocket
- Support redirects (from HTTP to HTTPS for example)
- Routing tables to different target groups:
  - Routing based on path in URL (example.com/users & example.com/posts)
  - Routing based on hostname in URL (one.example.com & other.example.com)
  - Routing based on Query String, Headers (example.com/users?id=123&order=false)
- ALB are a great fit for micro services & container-based application (Eg: Docker, ECS)
- Has a port mapping feature to redirect to a dynamic port in ECS



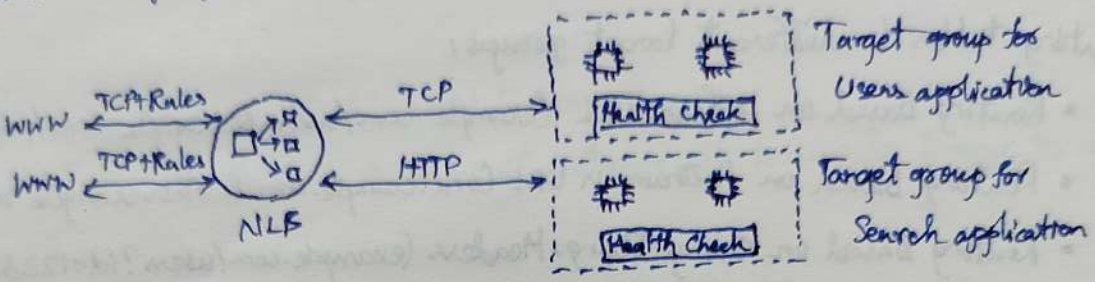
Target Groups - ALB

- EC2 instances (can be managed by an Auto Scaling Group) - HTTP
- ECS tasks (managed by ECS itself) - HTTP
- Lambda functions - HTTP request is translated into a JSON event
- IP addresses - must be private IPs
- ALB can route to multiple target groups.
- Health checks are at the target group level



Network Load Balancer

- It is Layer 4 that handle millions of requests per second
- Forward TCP & UDP traffic to your instances.
- Less latency ~ 100 ms (vs 400 ms for ALB)
- It has one static IP per AZ and supports assigning Elastic IP (helpful for whitelisting specific IP)
- Used for extreme performance TCP or UDP traffic

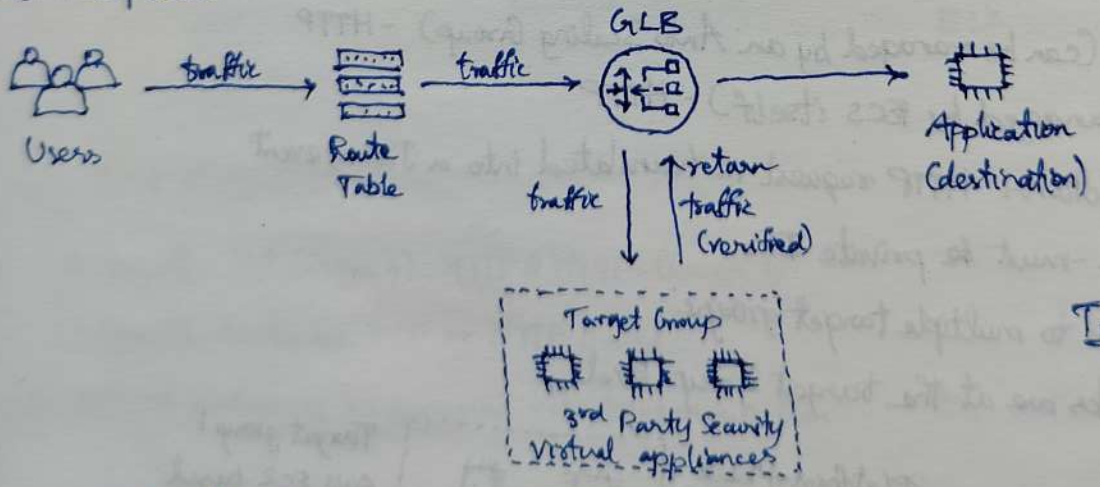


Target Groups - NLB

- EC2 instances
- IP addresses - must be private IPs
- Application Load Balancer

Gateway Load Balancer

- Deploy, scale, manage a fleet of 3rd party network virtual appliances in AWS
- Eg: Firewalls, Intrusion Detection and Prevention Systems, Deep Packet Inspection Systems, payload manipulation.



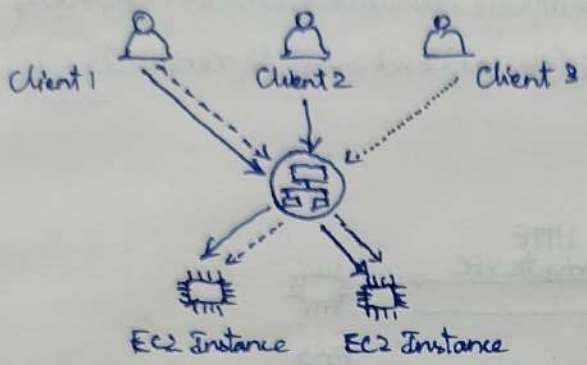
Target Groups - GLB

- EC2 instances
- IP addresses - must be private IPs

- Operates at Layer 3 (Network Layer) - IP Packets
- Combines the following functions:
  a) Transparent Network Gateway - single entry/exit for all traffic
  b) Load Balancer - distributes traffic to your virtual appliances
- Uses the GENEVE protocol on port 6081

Sticky Sessions (Session Affinity)

- Implement stickiness so that the same client is always redirected to the same instance behind a load balancer. It works for Classic Load Balancer & Application Load Balancer
- The 'cookie' used for stickiness has an expiration date you control.
- Use case: make sure the user doesn't lose his session data
- Enabling stickiness may bring imbalance to the load over the backend EC2 instances



Cookie Names

i) Application based Cookies

a) Custom Cookie

- Generated by the target
- Can include any custom attributes required by the application
- Cookie name must be specified individually for each target group

b) Application Cookie

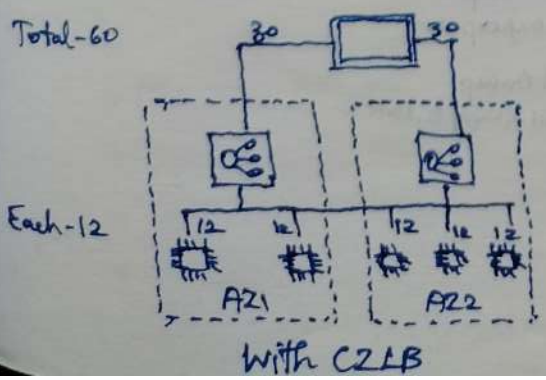
- Generated by the load balancer
- Cookie name is AWSALBAPP

ii) Duration based Cookies

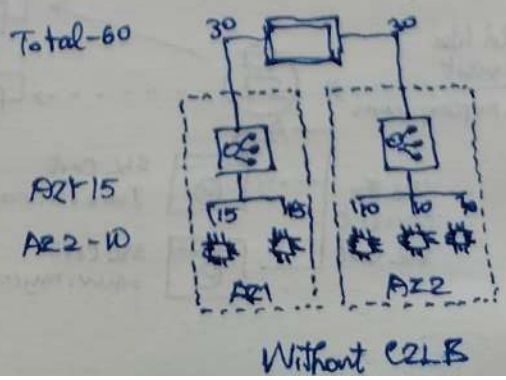
- Cookie generated by the load balancer
- Cookie name is AWSALB for ALB, AWSELB for CLB

Cross-Zone Load Balancing

- Each load balancer instance distributes evenly across all registered instances in all AZ



With C2LB



Without C2LB

ALB
- Always ON
- No charges

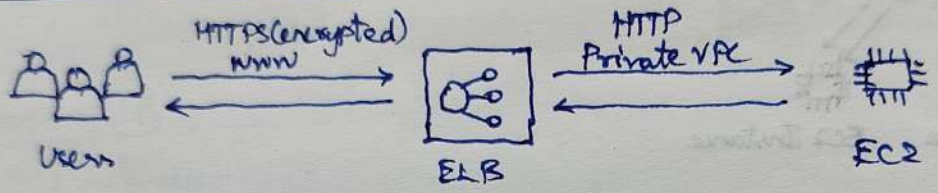
NLB
- Disabled
- Charges if enable

CLB
- Disabled
- No charges

SSL/TLS

- SSL certificate allows traffic between your clients and your load balancer to be encrypted in transit (in-flight encryption)
- SSL → Secure Sockets Layer used to encrypt connections
- TLS → Transport Layer Security newer version
- Nowadays TLS certificates are mainly used but people still refer as SSL
- Public SSL certificates are issued by Certificate Authorities (CA) Eg: GoDaddy
- SSL certificates have an expiration date (you set) and must be renewed.

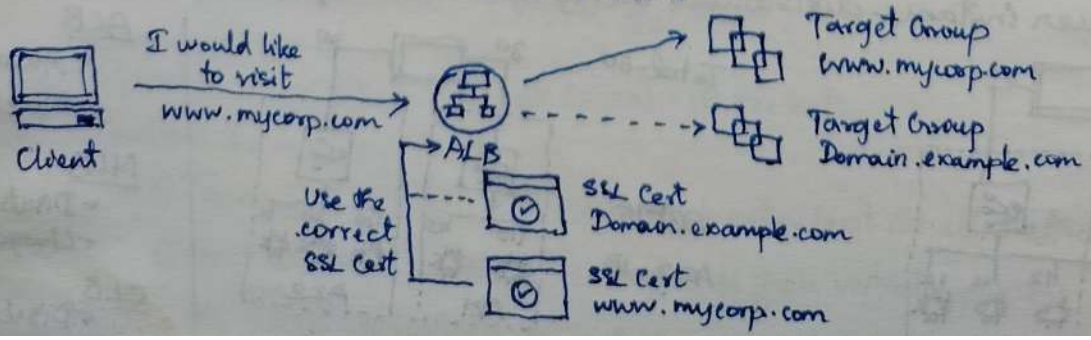
Load Balancer - SSL Certificates



- Load Balancer uses an X.509 certificate (SSL/TLS server certificate)
- You can manage certificates using ACM (AWS Certificate Manager)
- You can create upload your own certificates alternatively
- HTTPS listener
  - You must specify a default certificate
  - You can add an optional list of certs to support multiple domains.
  - Clients can use SNI (Server Name Indication) to specify the hostname they reach
  - Ability to specify a security policy to support older versions of SSL/TLS (legacy clien

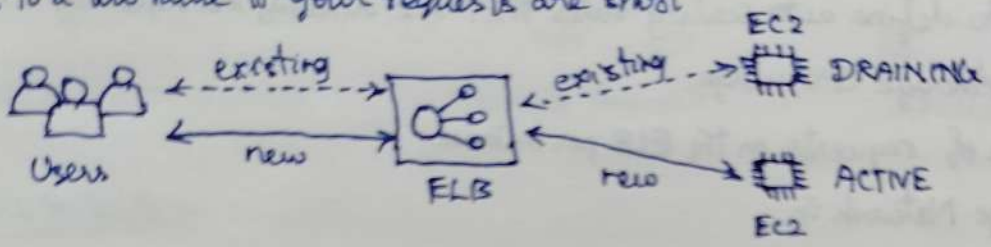
Server Name Indication

- Solves the problem of loading multiple SSL certificates onto one web server (to serve multiple websites)
- It's a 'newer' protocol and requires the client to indicate the hostname of the target server in the initial SSL handshake. Server will find correct certificate or return the default one.
- Only works for ALB & NLB, CloudFront



Connection Draining

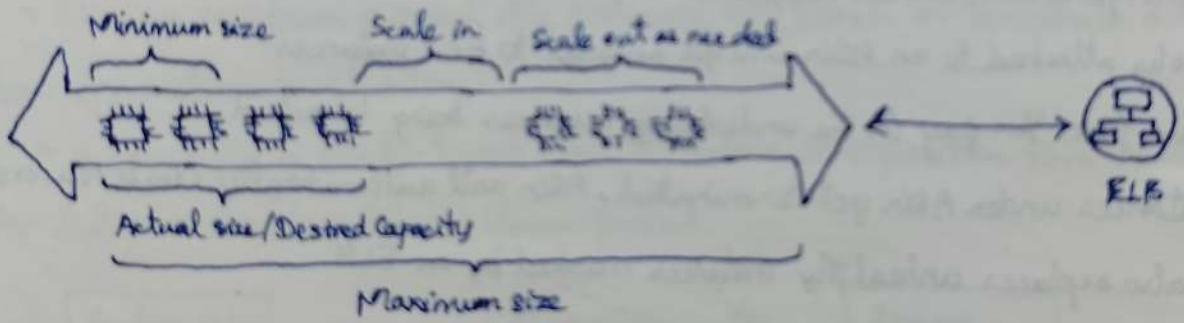
- Also known as 'Deregistration Delay'
- Time to complete 'in-flight' requests while the instance is de-registering or unhealthy
- Stops sending new requests to the EC2 instance which is de-registering.
- Between 1 to 3600 seconds (default: 300 seconds). It can be disabled (set value to 0)
- Set to a low value if your requests are short



Auto-Scaling Group

Why?

- Scale out (add EC2 instances) to match an increased load
- Scale in (remove EC2 instances) to match a decreased load
- Ensure we have a minimum and maximum number of machines running.
- Automatically register new instances to a load balancer.



ASG Attributes

- Launch configuration
  - AMI + Instance Type
  - EC2 User Data
  - EBS Volumes
  - Security Groups
  - SSH Key Pair
- Min Size / Max Size / Initial Capacity
- Network + Subnets information
- Load Balancer Information
- Scaling policies

Auto Scaling Alarms

- It is possible to scale an ASGr based on CloudWatch alarms.
- An alarm monitors a metric (such as Average CPU)
- Metrics are computed for the overall ASGr instances
- Based on the alarm we can create scale-out or scale-in policies
- It is possible to define auto scaling rules that are directly managed by EC2
    - Target Average CPU usage
    - Number of requests on the ELB per instance
    - Average Network in
    - Average Network out
- We can auto scale based on a custom metric (ex: no. of connected users)
    - Send custom metric from application on EC2 to CloudWatch (PutMetric API)
    - Create CloudWatch alarm to react to low/high values.
    - Use the CloudWatch alarm as the scaling policy for ASGr
- ASGrs use Launch Configuration or Launch Templates. To update an ASGr, you must provi a new launch configuration / launch template.
- IAM roles attached to an ASGr will get assigned to EC2 instances.
- ASGr are free. You pay for the underlying resources being launched
- If instances under ASGr get terminated, ASGr will automatically create new ones as replacement. It also replaces unhealthy instances marked by an ELB

Dynamic Scaling Policies

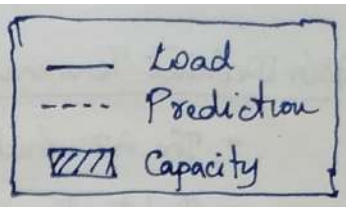
a) Target Tracking Scaling
- Most simple and easy to set-up
- Eg: Average ASGr CPU to stay at around 40%

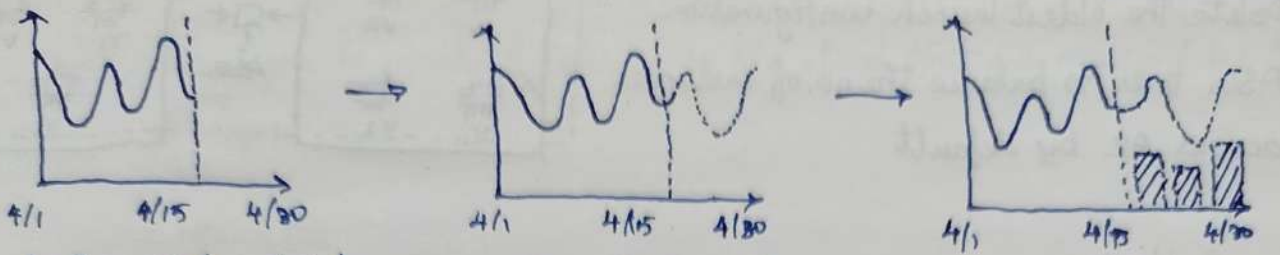
b) Simple / Step Scaling
- CloudWatch alarm (CPU > 70%) then add 2 units
- CloudWatch alarm (CPU < 30%) then remove 1 unit

c) Scheduled Actions
- Anticipate a scaling based on known usage patterns
- Eg: Increase the min capacity to 10 at 5 pm on Fridays

Predictive Scaling

- Continuously forecast load and schedule scaling ahead.





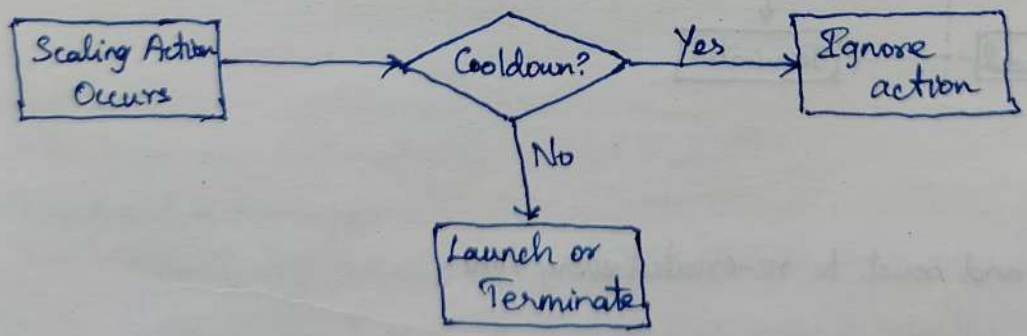
Analyze historical data → Generate forecast → Schedule scaling actions

Good Metrics to Scale on

- CPU Utilization : Average CPU utilization across your instances
- RequestCountPerTarget : Make sure no. of requests per EC2 instance is stable
- Average Network In/Out : If your application is network bound
- Any custom metric : Push using CloudWatch

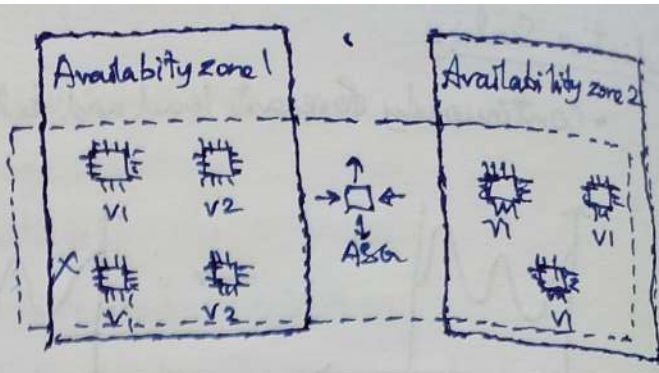
Scaling Cooldowns

- After a scaling activity happens, it will be in cooldown period (default 300 seconds)
- During cooldown period, the ASG will not launch or terminate additional instances (to allow for metrics to stabilize)
- Advice : Use a ready-to-use AMI to reduce configuration time in order to serve requests faster and reduce the cooldown period.



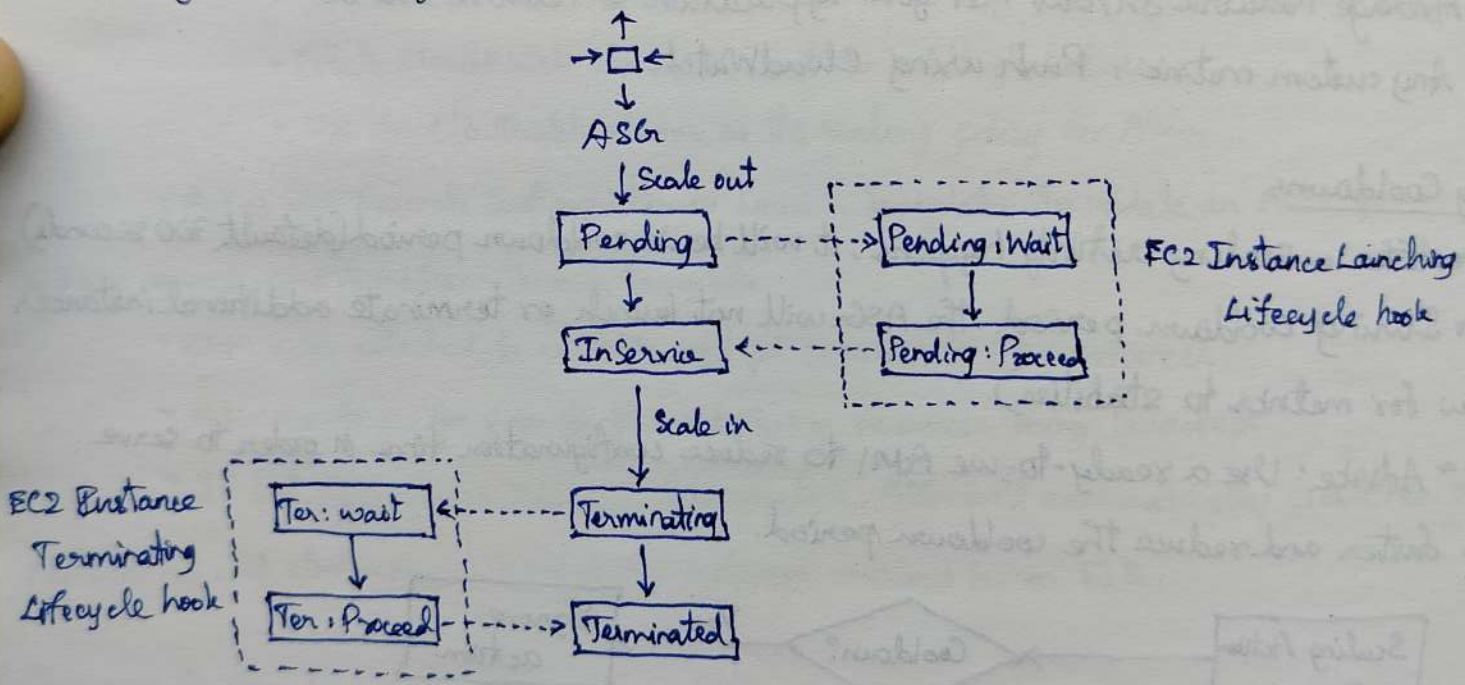
ASG Default Termination Policy

- The AZ which has most no. of instances
- Delete the oldest launch configuration
- ASG tries to balance the no. of instances across AZ by default



ASG Lifecycle Hooks

- By default as soon as an instance is launched in an ASG it's in service.
- You have the ability to perform extra steps before the instance goes in service (Pending or before the instance is terminated (Terminating state)
- Eg: Extract logs, data or any other info before the instance is terminated.



Launch Configuration

- It is legacy and must be re-created every time.

Launch Template

- Can have multiple versions
- Create parameters subsets (partial configuration for re-use and inheritance)
- Provision using both On-demand and Spot instances (or a mix)
- Can use T2 unlimited burst feature

AWS RDS Overview

- RDS stands for Relational Database Service
- It's a managed DB service for DB use SQL as a query language
- It allows you to create databases in the cloud that are managed by AWS
  - Postgres
  - MySQL
  - MariaDB
  - Oracle
  - Microsoft SQL Server
  - Aurora

Advantages over DB on EC2

- It is a managed service.
- Automated provisioning, OS patching, Monitoring dashboards
- Continuous backups and restore to specific timestamp (Point in Time Restore)
- Read replicas for improved read performance
- Multi AZ setup for DR (Disaster Recovery)
- Maintenance windows for upgrades
- Scaling capability (vertical and horizontal)
- Storage backed by EBS
- But you can't SSH into the RDS instances.

RDS Backups

- Backups are automatically enabled in RDS

a) Automated backups:
  - Daily full backup of the database (during the maintenance window)
  - Transaction logs are backed by RDS every 5 minutes
  - Ability to restore to any point in time (from oldest backup to 5 minutes ago)
  - 7 days retention (can be increased to 35 days)

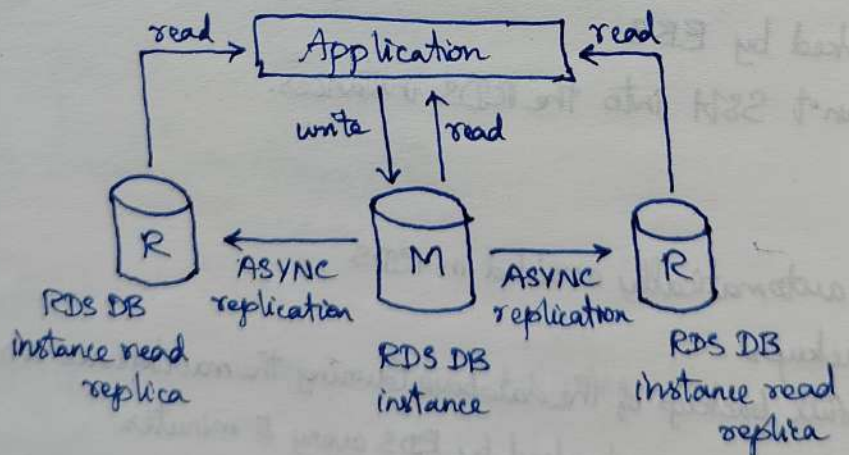
b) DB Snapshots:
  - Manually triggered by the user.
  - Retention of backup for as long as you want.

RDS - Storage Auto Scaling

- Helps you increase storage on your RDS DB instance dynamically.
- When RDS detects you are running out of free database storage, it scales automatically.
- Avoid manually scaling your database storage.
- You have to set Maximum Storage Threshold (maximum limit for DB storage)
- Automatically modify storage if:
  - Free storage is less than 10% of allocated storage
  - Low storage lasts at least 5 minutes
  - 6 hours have passed since last modification.
- Useful for applications with unpredictable workloads

RDS Read Replicas

- Upto 5 Read Replica
- Within AZ, Cross AZ or Cross Region
- Replication is ASYNC, so reads are eventually consistent.
- Replicas can be promoted to their own DB
- Applications must update the connection string to leverage read replicas.



Use Case:

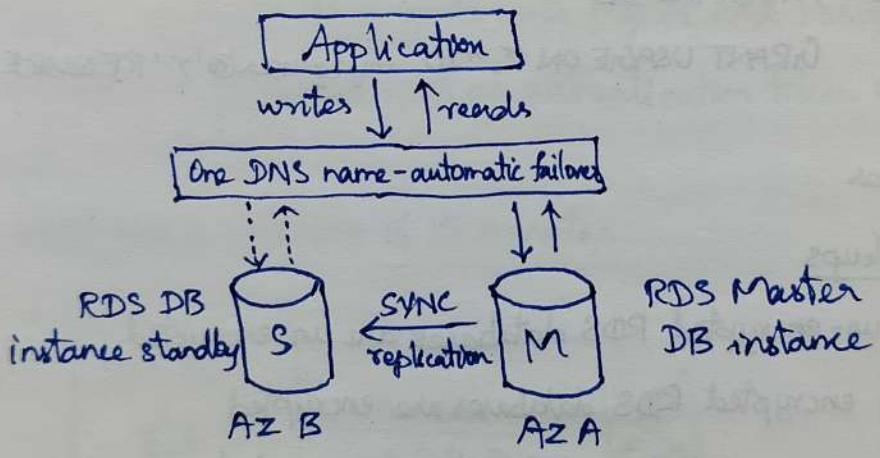
- You have a production database that is taking on normal load.
- You want to run a reporting application to run some analytics.
- You create a Read Replica to run the new workload so production application is unaffected.
- Read replicas are used for SELECT statement only
- Do not use INSERT, UPDATE, DELETE as it will modify the DB

RDS-Network Cost

- In AWS, there is a network cost when data goes from one AZ to another
- For RDS Read replicas within the same region, you don't pay that fee.

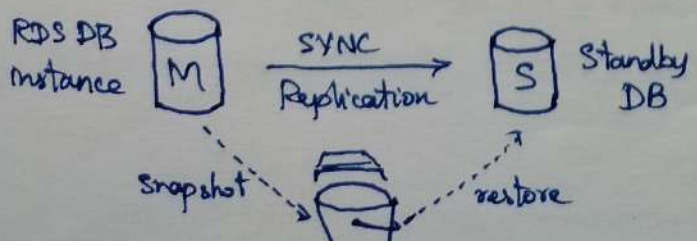
RDS Multi AZ

- SYNC replication
- One DNS name - automatic app failover to standby.
- Increase availability.
- Failover in case of loss of AZ, loss of network, instance or storage failure.
- No manual intervention in apps.
- Not used for scaling.
- Note: Read Replicas can be setup as Multi AZ for Disaster Recovery.



RDS Single AZ to Multi AZ

- Zero downtime operation (no need to stop the DB)
- Just click on "modify" for the database and enable multi AZ
- The following happens behind the scenes:
  - A snapshot is taken
  - New DB is restored from the snapshot in a new AZ
  - Synchronization is established between the two databases



RDS Security - Encryption

a) At rest encryption

- Possibility to encrypt the master and read replicas with AWS KMS - AES 256 en
- Encryption has to be defined at launch time
- If the master is not encrypted, the read replicas cannot be encrypted
- Transparent Data Encryption (TDE) available for Oracle and SQL Server

b) In-flight encryption

- SSL certificate to encrypt data to RDS in flight
- Provide SSL options with trust certificate when connecting to database
- To enforce SSL:
  - PostgreSQL: rds.force_ssl=1 in the AWS RDS Console (Parameter Groups)
  - MySQL: Within the DB

```
GRANT USAGE ON *.* TO 'mysqluser'@'%' REQUIRE SSL;
```

RDS Encryption Operations

a) Encrypting RDS backups

- Snapshots of un-encrypted RDS databases are un-encrypted
- Snapshots of encrypted RDS databases are encrypted
- Can copy an unencrypted snapshot into an encrypted one

b) Encrypting RDS database:

- Create a snapshot of the unencrypted database
- Copy the snapshot and enable encryption for the snapshot
- Restore the database from the encrypted snapshot
- Migrate applications to the new database and delete the old database

RDS Security

a) Network security

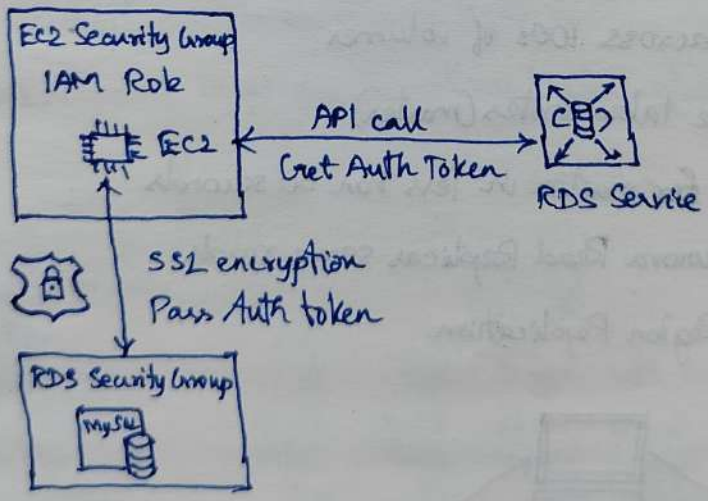
- RDS databases are usually deployed within a private subnet, not in a public one
- RDS security works by leveraging security groups - it controls which IP or security group can communicate with RDS

b) Access Management

- IAM policies help control who can manage AWS RDS (through the RDS API)
- Traditional username and password can be used to login into the database.
- IAM based authentication can be used to login into RDS MySQL & PostgreSQL

RDS IAM Authentication

- IAM database authentication works with MySQL and PostgreSQL
- You don't need a password, just an authentication token obtained through IAM & RDS API calls
- Auth token has a lifetime of 15 minutes



Benefits

- Network in/out must be encrypted using SSL
- IAM to centrally manage users instead of DB
- Can leverage IAM roles and EC2 instance profiles for easy integration.

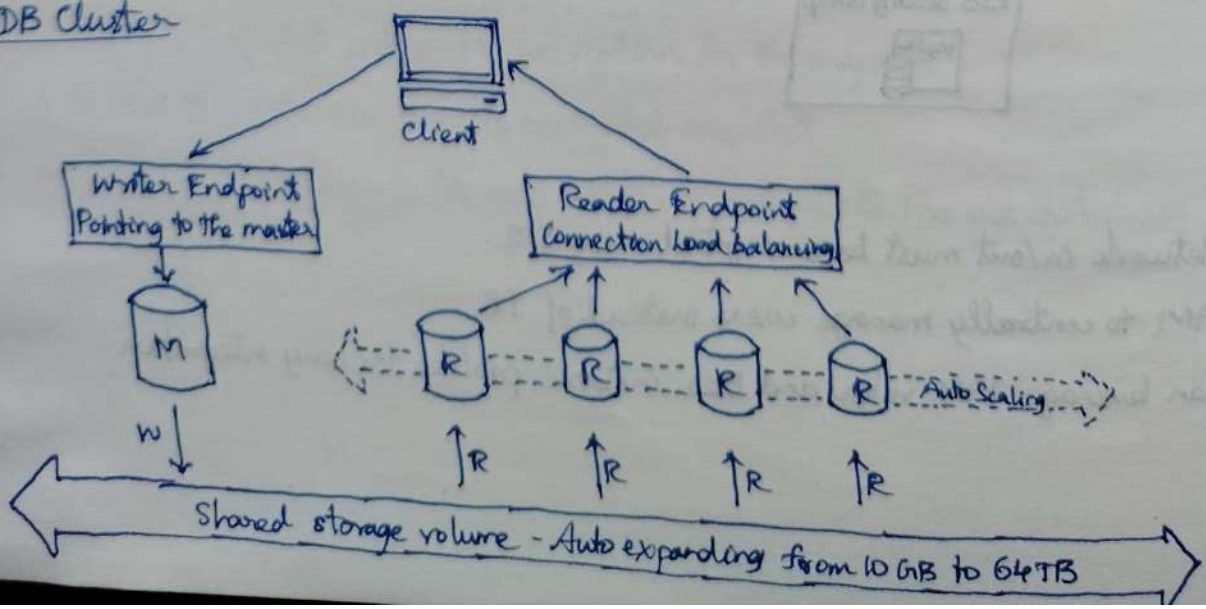
Amazon Aurora

- Proprietary technology from AWS (not open sourced)
- Postgres and MySQL are both supported as AuroraDB
- It is cloud-optimized. 5x performance improvement over MySQL on RDS, over 3x the performance of Postgres on RDS
- Aurora storage automatically grows in increments of 10 GB, up to 128 TB
- Aurora can have 15 replicas (MySQL has 5) and the replication process is faster (sub 10 ms replica lag)
- Failover in Aurora is instantaneous. It's High availability native
- Aurora costs more than RDS (20% more) - but it is more efficient.

Aurora - High Availability

- 6 copies of your data across 3 AZ
  - 4 copies out of 6 needed for writes
  - 3 copies out of 6 needed for reads
- Self-healing with peer-to-peer replication
- Storage is striped across 100s of volumes
- One Aurora Instance takes writes (master)
- Automated failover for master in less than 30 seconds
- Master + up to 15 Aurora Read Replicas serve reads
- Support for Cross Region Replication

Aurora DB Cluster



Aurora - Features

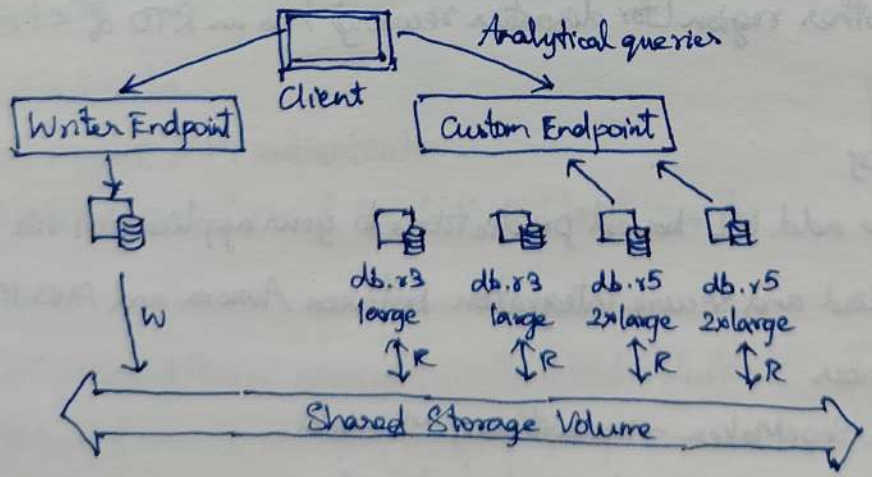
- Automatic fail-over
- Backup and Recovery
- Isolation and security
- Industry compliance
- Push-button scaling
- Automated patching with zero downtime
- Advanced Monitoring
- Routine Maintenance
- Backtrack: restore data at any point of time without using backups

Aurora - Security

- Similar to RDS because uses the same engines.
- Encryption at-rest using KMS
- Automated backups, snapshots and replicas are also encrypted
- Encryption in flight using SSL
- Possibility to authenticate using IAM token
- You are responsible for protecting the instance with security groups. You can't SSH

Aurora - Custom Endpoints

- Define a subset of Aurora instances as a Custom Endpoint
- Eg: Run analytical queries on specific replicas
- The Reader Endpoint is generally not used after defining Custom Endpoints

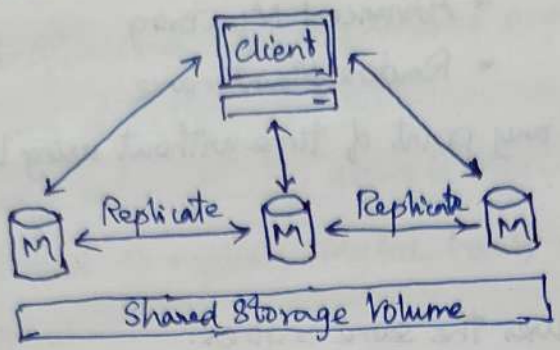


Aurora - Serverless

- Automated database instantiation and auto scaling based on actual usage.
- Good for infrequent, intermittent or unpredictable workloads
- No capacity planning needed. Pay per second can be more cost-effective
- Client is going to talk to 'Proxy fleet' many Aurora instances will be created

Aurora - Multi Master

- For immediate failover of write node to achieve High availability
- Every node does R/W - vs promoting a RR as the new master



Global Aurora

a) Aurora Cross Region Read Replicas

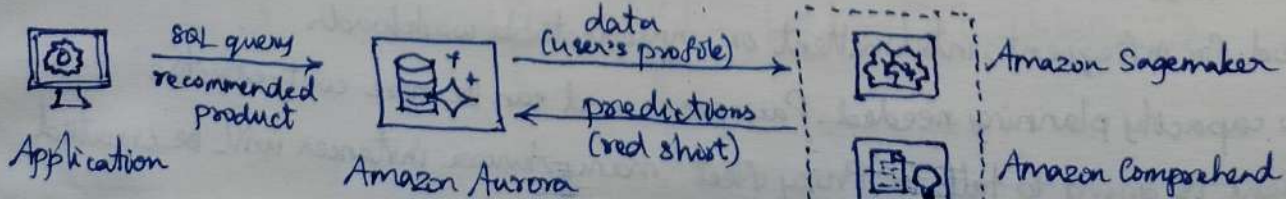
- Useful for disaster recovery
- Simple to put in place

b) Aurora Global Database

- One primary region (read/write)
- Upto 5 secondary (read-only) regions, replication lag is less than 1 second
- Upto 16 Read Replicas per secondary region
- Helps for decreasing latency
- Promoting another region (for disaster recovery) has an RTO of <1 minute

Aurora - Machine Learning

- Enables you to add ML-based predictions to your application via SQL
- Simple, optimized and secure integration between Aurora and AWS ML services
- Supported services
  - Amazon SageMaker - use with any ML model
  - Amazon Comprehend - for sentiment analysis
- Use Cases: Fraud detection, ads targeting, product recommendations



Amazon ElastiCache

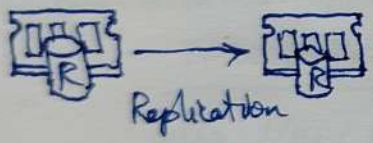
- To get managed Redis or Memcached
- Caches are in-memory databases with really high performance, low latency
- Helps reduce load off of databases for read intensive workloads
- Helps make your application stateless
- AWS takes care of OS maintenance / patching, optimization, setup, configuration monitoring, failure recovery and backups.
- Using ElastiCache involves heavy application code changes.

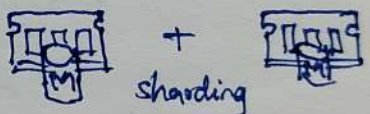
REDIS

- Multi AZ with Auto-Failover
- Read Replicas to scale reads and have high availability
- Data durability using AOF persistence
- Backup and restore features

MEMCACHED

- Multi-node for partitioning of data (sharding)
- No high availability (replication)
- Non persistent
- Multi-Threaded architecture
- No backup and restore





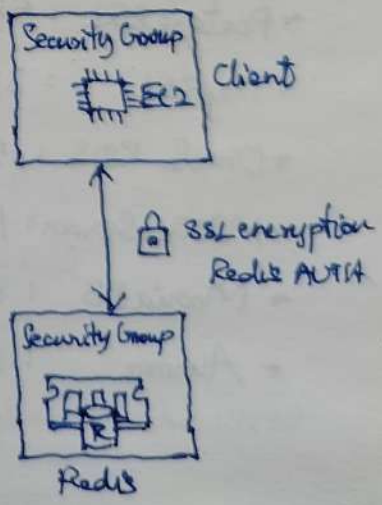
ElastiCache - Security

a) All Cache in ElastiCache

- Do not support IAM authentication
- IAM policies are only used for AWS-API level security

b) Redis AUTH

- Set a 'password/token' when you create a Redis cluster
- Extra level of security on top of security groups
- Support SSL in flight encryption



c) Memcached

- Supports SASL-based authentication (advanced)

ElastiCache - Patterns

- Lazy loading : All the read data is cached. data can become stale in cache
- Write Through : Adds or update data in the cache when written to a DB (no stale data
- Session Store : Store temporary session data in a cache (using TTL features)

ElastiCache - Redis Use Case

- Gaming leaderboards are computationally complex
- Redis sorted sets guarantee both uniqueness and element ordering (1, 2, 3..)
- Each time a new element is added, it's ranked in real time, then added in correct

Important ports

- FTP : 21
- SSH : 22
- SFTP : 22
- HTTP : 80
- HTTPS : 443

RDS Database ports

- PostgreSQL : 5432
- MySQL : 3306
- Oracle RDS : 1521
- MSSQL Server : 1433
- MariaDB : 3306
- Aurora : 5432 (if PostgreSQL)
  3306 (if MySQL)

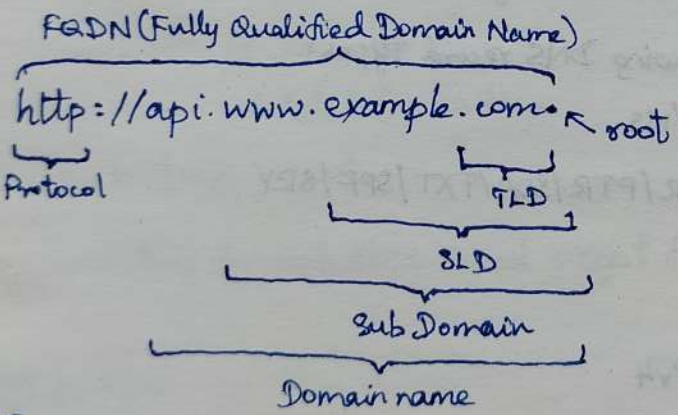
DNS

- Domain Name System which translates the human friendly hostnames into the machine IP addresses. Eg: www.google.com → 172.217.18.36
- DNS is the backbone of the Internet. It uses hierarchical naming structure

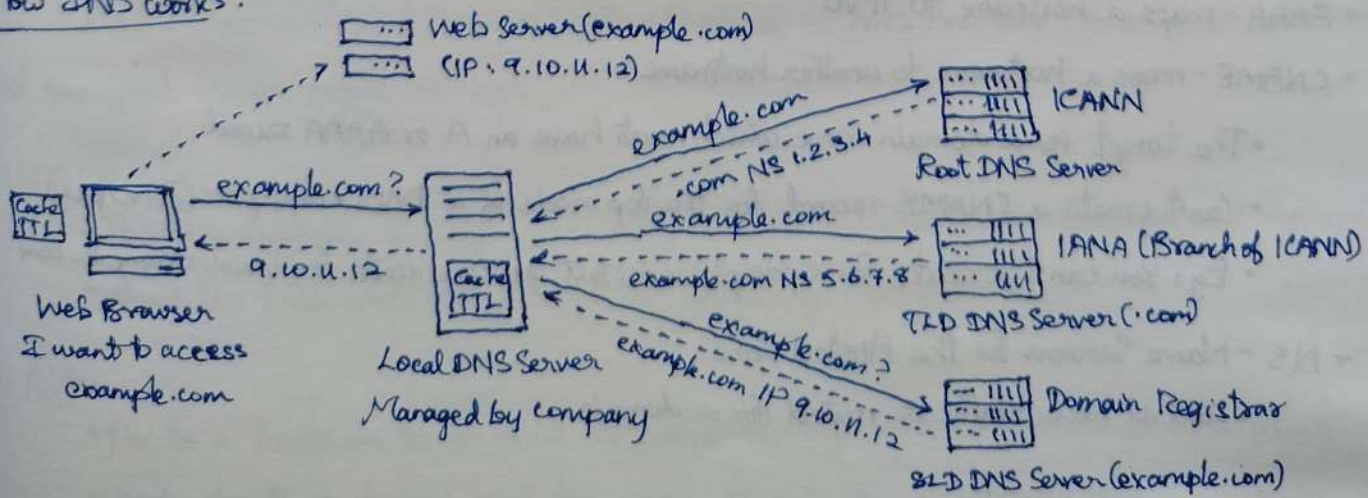
.com
example.com
www.example.com
api.example.com

DNS - Terminologies

- Domain Registrar : Amazon Route 53, GoDaddy
- DNS Records : A, AAAA, CNAME, NS
- Zone File : Contains DNS records
- Name server : Resolves DNS queries (Authorative or Non-Authorative)
- Top Level Domain (TLD) : .com, .us, .in, .gov, .org
- Second Level Domain (SLD) : amazon.com, google.com



How DNS works?



Amazon Route 53

- Highly available, scalable, fully managed and Authoritative DNS
  - Authoritative → Customer (you) can update the DNS records
- Route 53 is also a Domain Registrar
- Ability to check the health of your resources
- Only AWS service which provides 100% availability SLA
- Why number 53? 53 is a reference to the traditional DNS port

Route 53 - Records

- Define how you want to route traffic for a domain
- Each record contains:
  - Domain/Subdomain name - Eg: example.com
  - Record Type. Eg: A or AAAA
  - Value. Eg: 12.34.56.78
  - Routing Policy - how Route 53 responds to queries
  - TTL - Time to live, the amount of time the record cached at DNS Resolvers
- Route 53 supports the following DNS record types:
  - A/AAAA/CNAME/NS
  - CAA/DS/MX/NAPTR/PTR/SOA/TXT/SPF/SRV

Route 53 - Record Types

- A - maps a hostname to IPv4
- AAAA - maps a hostname to IPv6
- CNAME - maps a hostname to another hostname
  - The target is a domain name which must have an A or AAAA record
  - Can't create a CNAME record for the top node of a DNS namespace (Zone Apex)
  - Eg: You can't create for example.com but you can create for www.example.com
- NS - Name Servers for the Hosted Zone
  - Control how traffic is routed for a domain

Route 53 - Hosted Zones

- A container for records that define how to route traffic to a domain and its subdomains

a) Public Hosted Zones

- Contains records that specify how to route traffic on the internet (public domain names)
  application.mypublicdomain.com

b) Private Hosted Zones

- Contains records that specify how you route traffic within one or more VPCs

(private domain names) application.company.internal

- You pay $0.50 per month per hosted zone. i.e minimum of $12 per year

Route 53 - Records TTL (Time to Live)

a) High TTL - eg: 24 hours

- Less traffic on Route 53
- Possibly outdated records

b) Low TTL - eg: 60 seconds

- More traffic on Route 53 ($)
- Records are outdated for less time
- Easy to change records

→ TTL is mandatory for each DNS record except for Alias records.

Route 53 - CNAME vs Alias

AWS Resources (Load Balancer, CloudFront) expose an AWS hostname : lb-1234.us-east2.elb.aws.com and you want myapp.mydomain.com

a) CNAME

- Points a hostname to any other hostname (app.mydomain-com ⇒ blabla.anything.com)
- Only for Non-root domain (Eg: something.mydomain.com)

b) Alias

- Points a hostname to an AWS Resource (app.mydomain.com ⇒ blabla.amazonaws.com)
- Works for Root domain and Non-root domain (Eg: mydomain.com)
- Free of charge and Native Health check capability

Route 53 - Alias Records

- Maps a hostname to an AWS resource
- An extension to DNS functionality
- Automatically recognizes changes in the resource's IP addresses
- Unlike CNAME it can be used for the top node of a DNS namespace (Zone Apex)

Eg: example.com

- Alias Record is always of type A/AAAA for AWS resources (IPv4/IPv6)
- You can't set the TTL, it is set automatically by Route 53

Route 53 - Alias Records Targets

- Elastic Load Balancers
- CloudFront Distributions
- API Gateway
- Elastic Beanstalk environments
- S3 Websites
- VPC Interface endpoints
- Global Accelerator
- Route 53 record in the same hosted zone

> You cannot set an ALIAS record for an EC2 DNS name

Route 53 - Routing Policy

- Define how Route 53 responds to DNS queries
- DNS does not route any traffic, it only responds to the DNS queries.
- Route 53 supports the following Routing Policies:

a) Simple
b) Weighted
c) Failover
d) Latency based
e) Geolocation
f) Multi-value answer
g) Geoproximity (using Route 53 traffic Flow feature)

Simple Routing Policy

- Typically route traffic to a single resource
- Can specify multiple values in the same record.
- If multiple values are returned, a random one is chosen by the client
- When Alias is enabled, specify only one AWS resource
- Can't be associated with Health Checks

Weighted Routing Policy

- Control the % of the requests that go to each specific resource.
- Assign each record a relative weight

$$\text{traffic \%} = \frac{\text{Weight for a specific record}}{\text{Sum of all weights for all records}}$$

Note: Weights don't need to sum up to 100

- DNS records must have same name and type
- Can be associated with Health checks
- Use cases: Load balancing between regions, testing new application versions
- Assign a weight of 0 to a record to stop sending traffic to a resource.

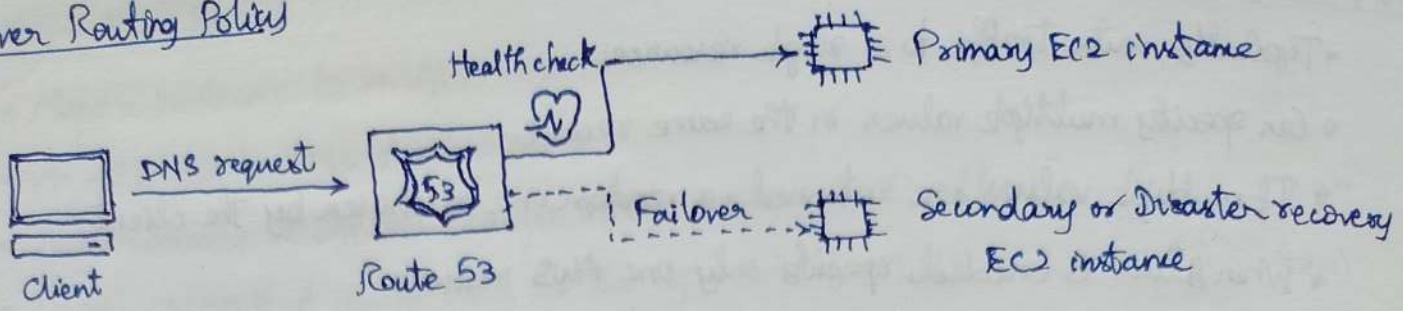
Latency-based Routing Policy

- Redirect to the resource that has the least latency close to us
- Super helpful when latency for users is a priority
- Latency is based on traffic between users and AWS Regions.
- Can be associated with Health checks (has a failover capability)

Geolocation Routing Policy

- Different from Latency-based. Routing is based on user location
- Specify location by continent, Country or by US State
- Should create a "Default" record (incase there is no match on location)
- Can be associated with Health checks
- Use cases: website localization, restrict content distribution, load balancing

Failover Routing Policy



Multi-value Routing Policy

- Use when routing traffic to multiple resources.
- Route 53 return multiple values/resources
- Can be associated with Health checks (return only values for health resources)
- Up to 8 healthy records are returned for each Multi-value query
- Multi-value is not a substitute for having an ELB

Geoproximity Routing Policy

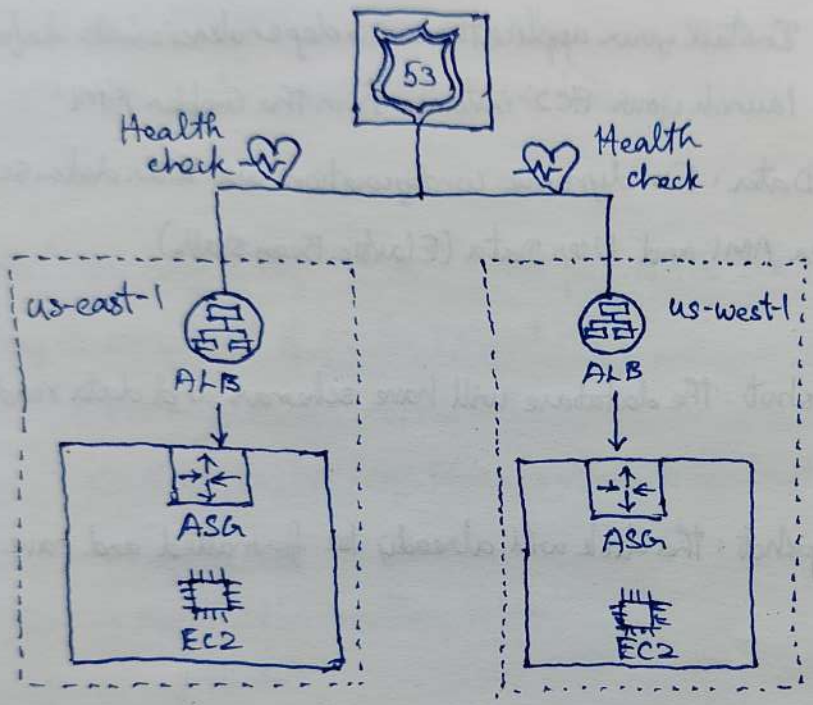
- Route traffic to your resources based on the geographic location of users and resources
- Ability to shift more traffic to resources based on the defined bias
- To change the size of the geographic region, specify bias values:
  - To expand (1 to 99) - more traffic to the resource
  - To shrink (-1 to -99) - less traffic to the resource
- Resources can be:
  - AWS resources (specify AWS region)
  - Non-AWS resources (specify Latitude and Longitude)
- You must use Route 53 Traffic flow (Advanced) to use this feature

Route 53 - Traffic Flow

- Simplify the process of creating and maintaining records in large and complex configurations
- Visual editor to manage complex routing decision trees.
- Configurations can be saved as Traffic flow policy
  - Can be applied to different Route 53 Hosted Zones (different domain names)
  - Supports versioning

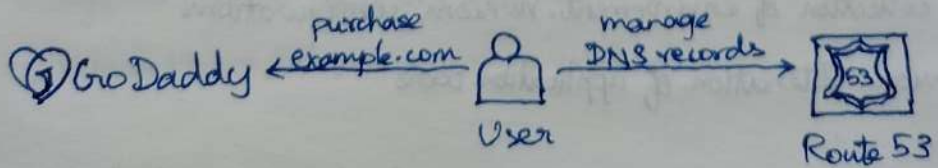
Route 53 - Health Check

- HTTP Health Check are only for public resources
- Automated DNS failover:
  - i) Health checks that monitor an endpoint - Application server, other AWS resources
  - ii) Health checks that monitor other health checks - Calculated Health Checks
  - iii) Health checks that monitor CloudWatch Alarms - DynamoDB throttle, alarms on RDS
- Health Checkers are integrated with CloudWatch metrics



Domain Registrar vs DNS Service

- You buy or register your domain name with a Domain Registrar typically by paying annual charges (Eg: GoDaddy, Amazon Registrar)
- The domain registrar usually provides you with a DNS service to manage your DNS records
- But you can use another DNS service to manage your DNS records
- Eg: Purchase domain from GoDaddy and use Route 53 to manage your DNS records



Domain Registrar != DNS Service

Instantiating Applications quickly

- When launching a full stack (EC2, EBS, RDS) it can take time to:
  - Install applications
  - Insert initial (or recovery) data
  - Configure everything
  - Launch the application

a) EC2 Instances

- Use a Golden AMI : Install your applications, OS dependencies etc beforehand and launch your EC2 instance from the Golden AMI
- Bootstrap using User Data : For dynamic configuration use User data scripts
- Hybrid : mix Golden AMI and User Data (Elastic Beanstalk)

b) RDS Databases

- Restore from a snapshot : the database will have schemas and data ready

c) EBS Volumes

- Restore from a snapshot : the disk will already be formatted and have data

Elastic Beanstalk

- Developer centric view of deploying an application on AWS
- It is a managed service that handles EC2, ASG, ELB, RDS ...
- It automatically handles capacity provisioning, load balancing, scaling, monitoring, etc.
- Just the application code is responsibility of the developer
- We still have full control over the configuration
- Beanstalk is free but you pay for the underlying instances

Components

- Application : collection of environment, versions, configurations
- Application Version : Iteration of application code
- Environment :
  - Collection of AWS resources running an application version (only one version at a time)
  - Tiers : Web Server Environment Tier & Worker Environment Tier
  - You can create multiple environments (dev, test, prod)

Amazon S3

- It allows to store objects (files) in buckets (directories)
- Buckets must have a globally unique name.
- Buckets are defined at the region level.
- Naming convention
    - No uppercase
    - No underscore
    - 3-63 characters long
    - Not an IP
    - Must start with lowercase letter or number

S3 Objects

- Objects (files) have a key. It is the full path of the file
- The key is composed of prefix + object name

  s3://my-bucket/myfile.txt

  s3://my-bucket/my-folder/another-folder/myfile.txt

- Bucket → my-bucket

  Prefix → my-folder/another-folder

  Object → myfile.txt

- There is no concept of 'directories' within buckets. The UI folder structure is for UX only

Just keys with very long ~~no~~ names that contain slashes (/)

- Object values are the content of the body
    - Max Object Size is 5TB (5000 GiB)
    - For uploading more than 5 GiB, use 'multi'-part' upload
- Each object can have:
    - Metadata → list of key/value pairs - system or user metadata
    - Tags → Unicode key/value pair - up to 10 useful for security/lifecycle
    - VersionID → If versioning is enabled

S3 Versioning

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

S3 Encryption

a) SSE-S3 : encrypts S3 objects using keys handled & managed by AWS

b) SSE-KMS : leverage AWS Key Management Service to manage encryption keys

c) SSE-C : when you want to manage your own encryption keys

d) Client Side Encryption

a) SSE-S3

- Object is encrypted server side
- AES-256 encryption type
- Must set header: 'X-amz-server-side-encryption': 'AES256'

b) SSE-KMS

- Object is encrypted server side
- KMS advantages: user control + audit trail
- Must set header: 'X-amz-server-side-encryption': 'aws:kms'

c) SSE-C

- Object is encrypted server side
- Amazon S3 does not store the encryption key you provide
- HTTPS must be used for uploading
- Encryption key must be provided in HTTP headers for every HTTP request made

d) Client Side Encryption

- Client library such as the Amazon S3 Encryption Client
- Clients must encrypt data themselves before sending to S3
- Clients must decrypt data themselves when retrieving from S3
- Customer fully manages the keys and encryption cycle

S3 Security

- User based
  - IAM policies - which API calls should be allowed for a specific user from IAM console
- Resource based
  - Bucket Policies - bucket wide rules from the S3 console - allows cross account
  - Object Access Control list (ACL) - finer grain
  - Bucket Access Control List (ACL) - less common
- An IAM principal can access an S3 object if
  - The user IAM permissions allow it or the resource policy allows it
  - And there is no explicit Deny

S3 Bucket Policies

JSON based policy

- Resources : buckets and objects
- Actions : Set of API to Allow or Deny
- Effect : Allow / Deny
- Principal : 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)

```
{
  "Version": "2022-01-20",
  "Statement": [
    {
      "Sid": "PublicRead",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3::: bucket/*"
      ]
    }
  ]
}
```

Block Public Access

- Block public access to buckets and objects granted through
  - new access control lists (ACLs)
  - any access control lists (ACLs)
  - new public bucket or access point policies
- Block public and cross-account access to buckets and objects through any public bucket or access point policies.
- These settings were created to prevent company data leaks
- If you know your bucket should never be public, leave these on
- It can be set at the account level.

S3 - Security Other

a) Networking
- Supports VPC endpoints (for instances in VPC without www internet)

b) Logging and Audit
- S3 Access Logs can be stored in other S3 bucket
- API calls can be logged in AWS CloudTrail

c) User Security
- MFA Delete: MFA (multi factor authentication) can be required in versioned buckets to delete objects
- Pre-Signed URLs: URLs that are valid only for a limited time
  Eg: Premium video service for logged in users

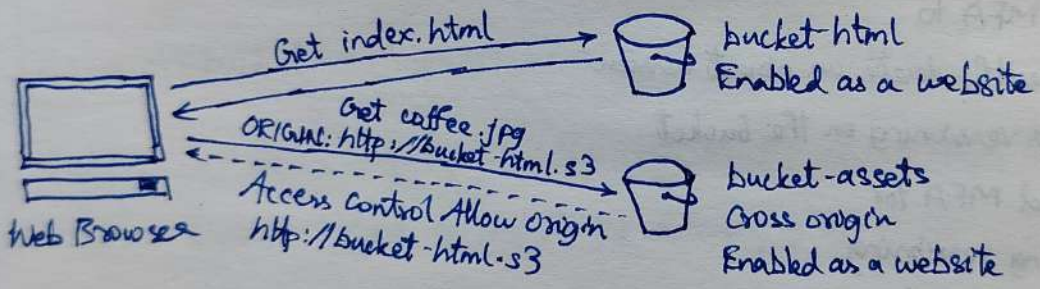
S3 Websites

- S3 can host static websites and have them accessible on the www
- The website URL will be:
  - <bucket-name>.s3-website-<AWS-region>.amazonaws.com
  - <bucket-name>.s3-website.<AWS-region>.amazonaws.com
- If you get a 403 (Forbidden) error, make sure the bucket policy allows public reads

S3 → Properties → Static Website Hosting

S3 CORS

- An origin is a scheme (protocol), host (domain) and port
  - Eg: https://www.example.com (port is 443 for HTTPS)
- CORS → Cross-Origin Resource Sharing
- Web Browser based mechanism to allow requests to other origins while visiting the main origin
- Same origin: https://example.com/app & https://example.com/app2
- Different origins: https://www.example.com & https://other-example.com
- The requests won't be fulfilled unless the other origin allows for the requests, using CORS Headers (ex: Access-Control-Allow-Origin)
- If the client does a cross-origin request on our S3 bucket, we need to enable the correct CORS headers.
- You can allow for a specific origin or for * (all origins)



AWS EC2 Instance Metadata

- It is powerful but one of the least known features to developers.
- It allows AWS EC2 instances to "learn about themselves" without using an IAM Role for that purpose.
- The URL is http://169.254.169.254/latest/meta-data
- You can retrieve the IAM Role name from the metadata but you cannot retrieve the IAM policy
- Metadata → Info about the EC2 instance
- Userdata → Launch script of the EC2 instance

AWS SDK

- To perform actions on AWS directly from your applications code (without using CLI)
- Official SDKs are: Java, .NET, Node.js, PHP, Python, Go, Ruby, C++
- We have to use AWS Software Development Kit when coding against AWS services such as DynamoDB
- The AWS CLI uses the Python SDK (boto 3)
- If you don't specify or configure a default region, then us-east-1 will be chosen by default

S3 - MFA

- Multi-factor Authentication forces user to generate a code on a device (usually a mobile phone or hardware) before doing important operations on S3
- To use MFA - Delete, enable versioning on the S3 bucket
- You will need MFA to
  - Permanently delete an object version
  - Suspend versioning on the bucket
- You won't need MFA for
  - Enabling versioning
  - Listing deleted versions
- Only the bucket owner (root account) can enable/disable MFA - Delete
- MFA - Delete currently can only be enabled using the CLI

S3 - Default Encryption

- One way to "force encryption" is to use a bucket policy and refuse any API call to PUT an S3 object without encryption headers

  "s3:x-amz-server-side-encryption": "AE256"

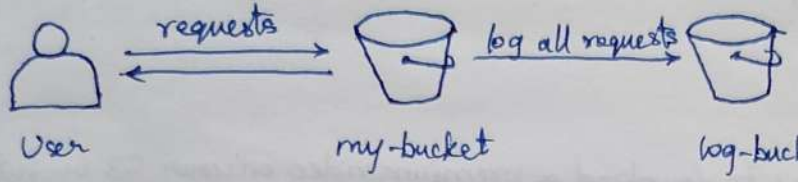
- Another way is to use the "default encryption" option in S3. If you already uploaded a encrypted object it will not be re-encrypted
- Bucket Policies are evaluated before 'default encryption'

S3 → Bucket → Properties → Default encryption

Automatically encrypt new objects stored in this bucket

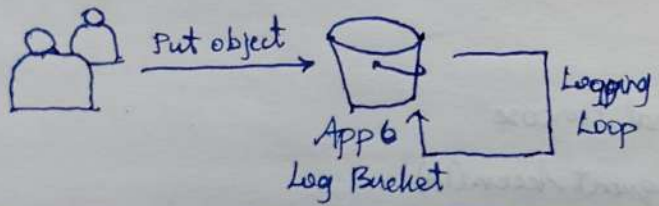
S3 Access Logs

- For audit purpose, you may want to log all access to S3 buckets
- Any request made to S3, from any account, authorized or denied, will be logged into another S3 bucket
- The log data can be analyzed using data analysis tools or Amazon Athena



Warning!

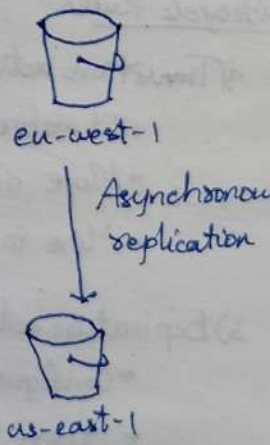
- Do not set your logging bucket to be monitored bucket as well. It will create a logging loop and your bucket will grow in size exponentially.



S3 → Appbucket → Properties → Server Access Logging

Note: By enabling S3 Access Logging, it will automatically update your bucket access control list (ACL) to include access to the S3 log delivery group

S3 Replication



- Must enable versioning in source and destination
- Cross Region Replication (CRR) & Same Region Replication (SRR)
- Buckets can be in different accounts. Copying is asynchronous
- Must give proper IAM permissions to S3
- CRR Use cases: compliance, lower latency access
- SRR Use cases: log aggregation, live replication
- After activating, only new objects are replicated (not retroactive)
- For DELETE operations:
  - Can replicate delete markers from source to target (optional setting)
  - Deletions with a version ID are not replicated (to avoid malicious deletes)
- There is no "chaining" of replication - if bucket 1 has replication into bucket 2, which has replication into bucket 3, then objects created in bucket 1 are not replicated to bucket 3

S3 Pre-signed URL

- Can generate pre-signed URLs using SDK or CLI
- For downloads (easy-use CLI) and for uploads (hard - use SDK)
- Valid for a default of 3600 seconds, can change timeout with --expires-in [TimeSeconds]
- Users given a pre-signed URL inherit the permissions of the person who generated the URL for GET/PUT

Examples

- Allow only logged in users to download a premium video on your S3 bucket.
- Allow an ever changing list of users to download files by generating URLs dynamically
- Allow temporarily a user to upload a file to a precise location in our bucket.

S3 - Storage classes

- Amazon S3 Standard - General Purpose
- Amazon S3 Standard - Infrequent Access (IA)
- Amazon S3 One Zone - Infrequent Access
- Amazon S3 Intelligent Tiering
- Amazon Glacier
- Amazon Glacier - Deep Archive

S3 Lifecycle Rules

a) Transition actions

- It defines when objects are transitioned to another storage class
- Move objects to Standard IA class 60 days after creation
- Move to Glacier for archiving after 6 months

b) Expiration actions

- Configure objects to expire (delete) after some time
- Access log files can be set to delete after a 365 days
- Can be used to delete old versions of files (if versioning is enabled)
- Can be used to delete incomplete multi-part uploads

- Rules can be created for a certain prefix (Eg: s3://mybucket/mp3/*)
- Rules can be created for certain objects tags (Eg: Department: Finance)

S3 Analytics

- You can setup S3 Analytics to help determine when to transition objects from Standard to Standard_IA
  - Does not work for ONEZONE IA or GLACIER
  - Report is updated daily - Takes about 24 h to 48 h for first start
  - Good first step to put together Lifecycle Rules or to improve them

S3 - Baseline Performance

- Amazon S3 automatically scales to high request rates - latency 100-200 ms
- Your application can achieve atleast 3500 PUT/COPY/POST/DELETE and 5500 GET/HEAD requests per second per prefix in a bucket
- If you use SSE-KMS your S3 performance may be impacted by the KMS limits
- When you upload it calls the GenerateDataKey KMS API, when you download it calls the Decrypt KMS API
- Count towards the KMS quota per second (5500, 10000, 30000 req/s based on region)
- You can request a quota increase using Service Quotas Console

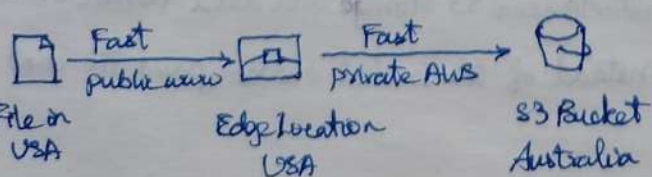
S3 - Enhanced Performance

a) Multi-part upload

- Recommended for files > 100 MB, must use for files > 5 GB
- Can help parallelize uploads (speed up transfers)

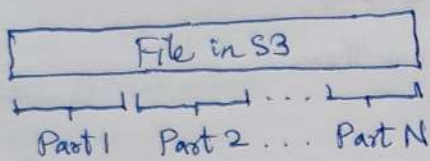
b) S3 Transfer Acceleration

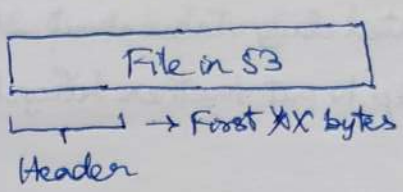
- Increase transfer speed by transferring file to an AWS edge location which will forward the data to the S3 bucket in the target region
- Compatible with multi-part upload



S3 Byte-Range Fetches

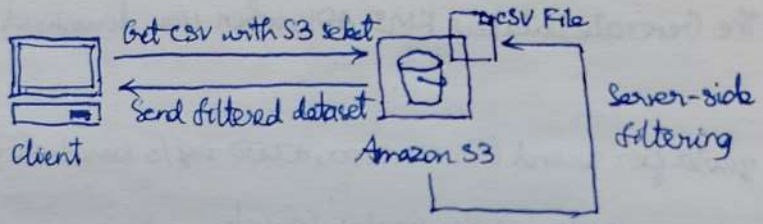
- Parallelize GETs by requesting specific byte ranges.
- Better resilience in case of failures
- Can be used to speed up downloads (or) retrieve only partial data (head of a file)





S3 Select & Glacier Select

- Retrieve less data using SQL by performing server side filtering
- Can filter by rows & columns (simple SQL statements)
- Less network transfer, less CPU cost client-side



S3 Event Notifications

- S3:ObjectCreated, S3:ObjectRemoved, S3:ObjectRestore, S3:Replication
- Object name filtering is possible (*.jpg)
- Can create as many S3 events as desired, typically delivers events in seconds but can sometimes take a minute or longer
- If two writes are made to a single non-versioned object at the same time, it is possible that only a single event notification will be sent. To ensure for all write, enable versioning.
- Targets are i) SNS ii) SQS iii) Lambda

S3 Requester Pays

- In general, bucket owners pay for all Amazon S3 storage and data transfer costs associated
- With Requestor Pays, the requester instead of the bucket owner pays the cost of the request and the data download from the bucket.
- Helpful when you want to share large datasets with other accounts.
- The requestor must be authenticated on AWS (cannot be anonymous)

Amazon Athena

- Serverless query service to perform analytics against S3 objects.
- Uses standard SQL language to query the files
- Supports CSV, JSON, ORC, Avro and Parquet (built on Presto)
- Pricing: $5.00 per TB of scanned data
- Use compressed or columnar data for cost savings (less scan)
- Use cases: Business Intelligence/analytics/reporting, analyze & query VPC Flow logs, ELB logs, CloudTrail, etc...

AWS CloudFront

- Content Delivery Network (CDN)
- Improves read performance, content is cached at the edge
- 216+ Points of presence globally (edge locations)
- DDoS protection, integration with Shield, AWS Web Application Firewall
- Can expose external HTTPS and can talk to internal HTTPS backends

CloudFront Geo Restriction

- Whitelist: Allow your users to access your content only if they're in one of the countries on a list of approved countries.
- Blacklist: Prevent your users from accessing your content if they're in one of the countries on a list of banned countries.
- The 'country' is determined using a 3rd party Geo-IP database
- Use case: Copyright laws to control access to content

CloudFront Signed URL / Signed Cookies

- You want to distribute paid shared content to premium users over the world
- We can use CloudFront Signed URL/Cookie. We attach a policy with:
  - Includes URL expiration
  - Includes IP ranges to access the data from
  - Trusted signers (which AWS accounts can create signed URLs)
- Signed URL → Access to individual files (one signed URL per file)
- Signed Cookies → Access to multiple files (one signed cookie for many files)

CloudFront Signed URL vs S3 Pre-signed URL

CloudFront Signed URL

- Allow access to a path, no matter the origin
- Account wide key-pair, only the root can manage it
- Can filter by IP, path, date, expiration
- Can leverage caching features

S3 Pre-Signed URL

- Issue a request as the person who pre-signed the URL
- Uses the IAM key of the signing IAM principal
- Limited lifetime

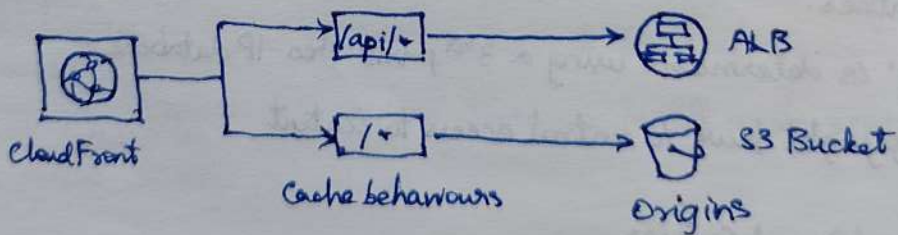
CloudFront - Pricing

- CloudFront edge locations are all around the world. The cost of data out per edge location varie
- You can reduce the no. of edge locations for cost reduction.
- Three Price classes:
  - i) Price class All : all regions - best performance
  - ii) Price class 200 : most regions but excludes the most expensive regions
  - iii) Price class 100 : only the least expensive regions

CloudFront - Multiple Origin

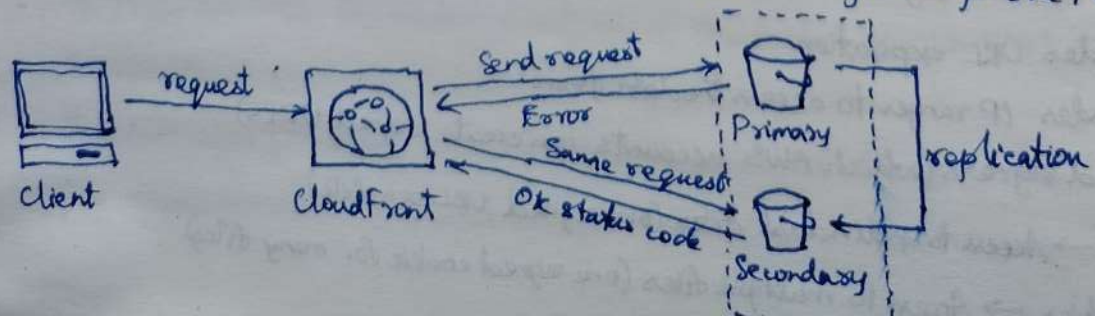
- To route different kind of origins based on the content type. Based on path pattern:

- /images/*
- /api/*
- /*



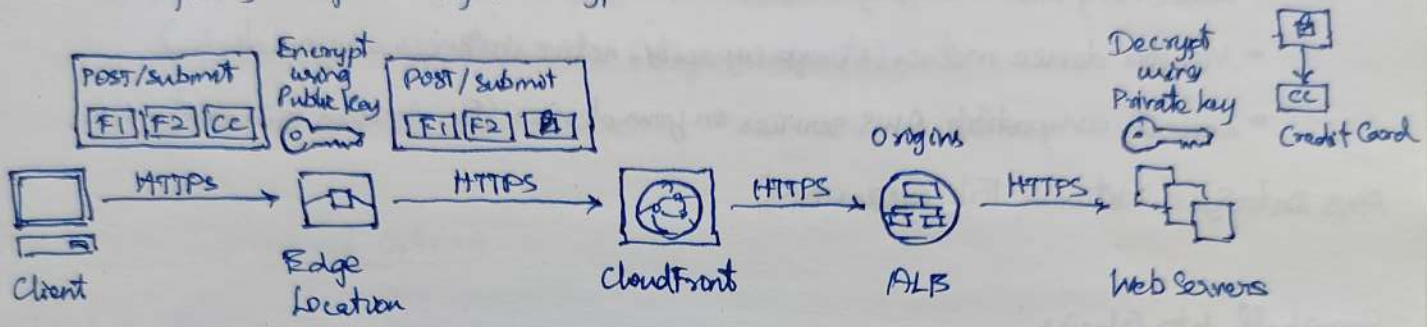
CloudFront - Origin Groups

- To increase high availability and do failover. Origin Group: One Primary & One Secondary



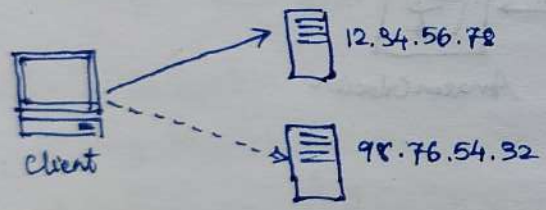
CloudFront - Field Level Encryption

- Protect user sensitive information through application stack
- Adds an additional layer of security along with HTTPS
- Sensitive information encrypted at the edge closer to user. Uses asymmetric encryption.
- Usage:
  - Specify set of fields in POST requests that you want to be encrypted (upto 10 fields)
  - Specify the public key to encrypt them

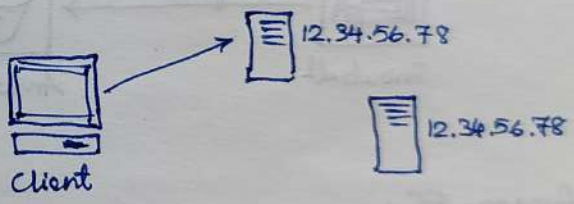


Unicast IP vs Anycast IP

- One server holds one IP address
- Routing based on different address



- All servers hold same IP address
- Routed to the nearest one to client



AWS Global Accelerator

- Leverage AWS internal network to route to your application. 2 anycast IP are created
- The Anycast IP send traffic directly to Edge locations and Edge locations send the traffic to App
- Works with Elastic IP, EC2 instances, ALB, NLB, public or private

a) **Consistent performance**
- Intelligent routing to lowest latency and fast regional failover
- Internal AWS Network. No issue with client cache (because the IP doesn't change)

b) **Health checks**
- Performs health check of applications - Create for disaster recovery
- Helps make your application global (failover less than 1 minute for unhealthy)

c) **Security**
- Only 2 external IP need to be whitelisted. DDoS protection from AWS Shield

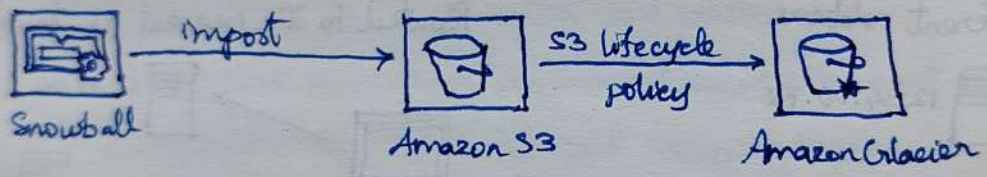
AWS OPSHUB

- Historically to use Snow family devices, you needed a CLI
- Now, you can use AWS Ops Hub - a software you install on your computer / laptop to manage your Snow Family Device
  - Unlocking and configuring single or clustered devices
  - Transferring files
  - Launching and managing instances running on Snow family devices
  - Monitor device metrics (storage capacity, active instances on your device)
  - Launch compatible AWS services on your devices (Eg: Amazon EC2 instances.

AWS DataSync, Network File system (NFS)

Snowball into Glacier

- Snowball cannot import to Glacier directly.
- You must use Amazon S3 first, in combination with an S3 Lifecycle policy.



Amazon FSx

- Launch 3rd party high-performance file systems on AWS
- Fully managed service.
  a) FSx for Lustre
  b) FSx for Windows File Server
  c) FSx for NetApp ONTAP

FSx for Lustre

- It is a type of parallel distributed file system for large scale computing. Linux + Cluster
- Machine learning, High performance computing, Video processing, Finance Modeling
- Scale up to 100s GB/s, millions of IOPS, sub-ms latencies
- Seamless integration with S3. Can read S3 as a file system and write back to S3
- Can be used from on-premise servers

FSx for Windows

- EFS is a shared POSIX system for Linux systems.
- FSx for Windows is a fully managed Windows file system share drive
- Supports SMB protocol & Windows NTFS
- Microsoft Active Directory integration, ACLs, user quotas
- Built on SSD, scale up to 10s of GB/s, millions of IOPS, 100s PB of data
- Can be accessed from your on-premise infrastructure
- Can be configured to be Multi-AZ (High availability)
- Data is backed-up daily to S3

FSx File System Deployment Options

a) Scratch File System

- Temporary storage
- Data is not replicated (doesn't persist if file server fails)
- High burst (6x faster, 200 MBps per TiB)
- Usage: short-term processing, optimize costs

b) Persistent File System

- Long-term storage
- Data is replicated within same AZ
- Replace failed files within minutes
- Usage: long-term processing, sensitive data

Amazon FSx File Gateway

- Native access to Amazon FSx for Windows File Server
- Local cache for frequently accessed data for lower latency
- Windows native compatability (SMB, NTFS, Active Directory)
- Useful for group file shares and home directories

File Gateway

- Configured S3 buckets are accessible using the NFS and SMB protocol
- Supports S3 standard, S3 IA, S3 One Zone IA
- Bucket access using IAM roles for each File Gateway
- Most recently used data is cached in the file gateway
- Can be mounted on many servers.
- Integrated with Active Directory (AD) for user authentication

Volume Gateway

- Block storage using iSCSI protocol backed by S3
- Backed by EBS snapshots which can help restore on-premises volumes
- Cached volumes: low latency access to most recent data
- Stored volumes: entire dataset is on premise, scheduled backups to S3

Tape Gateway

- Some companies have backup processes using physical tapes
- With Tape Gateway, companies use the same processes but in the cloud
- Virtual Tape Library (VTL) backed by Amazon S3 and Glacier
- Backup data using existing tape-based processes (and iSCSI interface)
- Works with leading backup software vendors.

Hardware Appliance

- Storage Gateway needs on-premises virtualization. You can use SG Hardware Appliance
- Works with File, Volume and Tape Gateway. It has the required CPU, memory, network, SSD cache resources. It is helpful for daily NFS backups in small data centers.

Storage Gateway Summary

- File Gateway → File access / NFS - user auth with Active Directory (backed by S3)
- Volume Gateway → Volumes / Block storage / iSCSI (backed by S3 with EBS Snapshots)
- Tape Gateway → VTL Tape solution / Backup with iSCSI (backed up by S3 and Glacier)

AWS Transfer Family

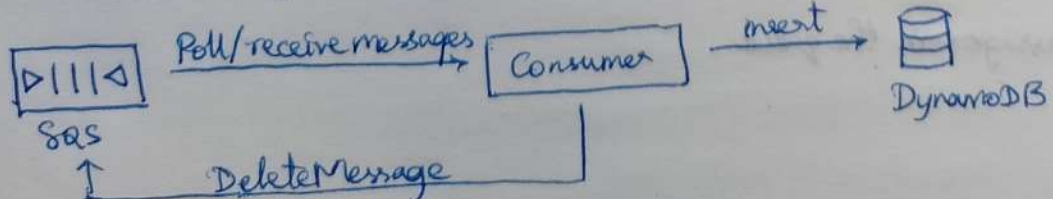
- A fully managed service for file transfers into and out of Amazon S3 or Amazon EFS using the FTP protocol
  - Supported Protocols
    - AWS Transfer for FTP (File Transfer Protocol)
    - AWS Transfer for FTPS (File Transfer Protocol over SSL)
    - AWS Transfer for SFTP (Secure File Transfer Protocol)
  - Managed infrastructure, Scalable, Reliable, Highly available (multi-AZ)
  - Pay per provisioned endpoint per hour + data transfers in GB
  - Store and manage users' credentials within the service
  - Integrate with existing authentication systems (Microsoft Active Directory, LDAP, Okta, Amazon Cognito, custom)
  - Usage: sharing files, public datasets, CRM, ERP...

SQS - Producing Messages

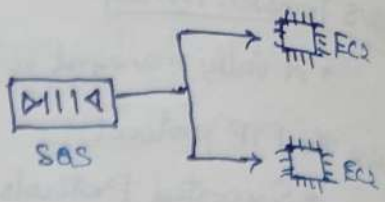
- Produced to SQS using the SDK (SendMessage API)
- The message is persisted in SQS until a consume reads it and it gets deleted from queue.
- Message retention: default 4 days up to 14 days
- Eg: Send an order to be processed
  - Order id, custome id, any attribute you want

SQS - Consuming Messages

- Consumers can be running on EC2 instance, servers or AWS Lambda
- Poll SQS for messages (receive upto 10 messages at a time)
- Process the messages (Eg: insert the message into DynamoDB)
- After processing, delete the messages using the DeleteMessage API

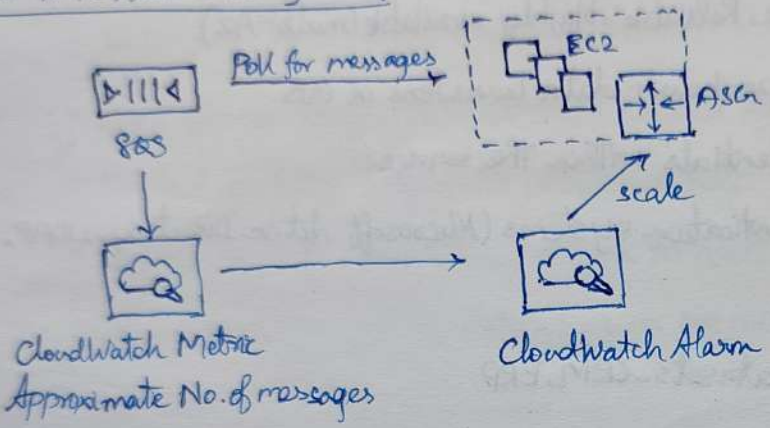


SQS - Multiple EC2 Instances Consumers



- Consumers receive and process messages in parallel.
- Atleast once delivery
- Best-effort message ordering
- Consumers delete message after processing them
- We can scale consumers horizontally to improve throughput of processing

SQS - With Auto Scaling Group



Amazon SQS - Security (Same as SNS)

a) Encryption
- In-flight encryption using HTTPS API
- At-rest encryption using KMS keys
- Client-side encryption if the client wants to do encryption/decryption himself

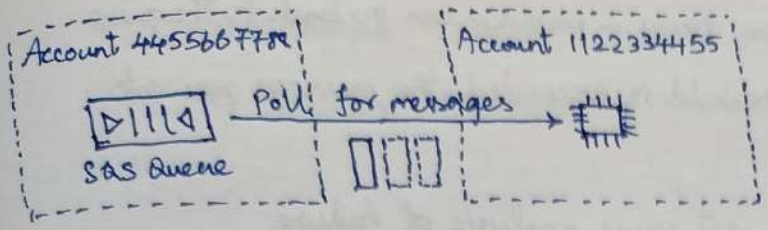
b) Access Controls
- IAM policies to regulate access to the SQS API

c) SQS Access Policies
- Useful for cross-account access to SQS queues
- Useful for allowing other services (SNS, S3...) to write to an SQS queue

Purge
- Delete all messages in the queue

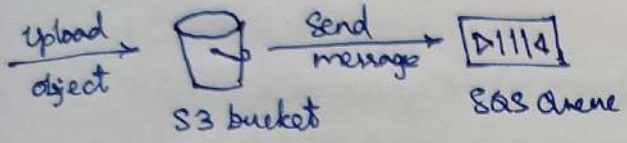
SQS Queue Access Policy

Cross Account Access




```
{
  "Version": "2022-01-21"
  "Statement": [{
      "Effect": "Allow",
      "Principal": {"AWS": ["1122334455"]},
      "Action": ["sqs:ReceiveMessage"],
      "Resource": "arn:aws:sqs:us-east-1:
                   445566778:queue"
    }]
}
```

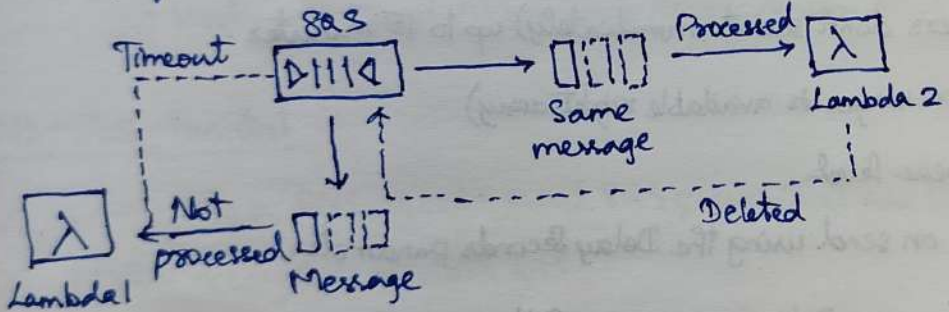
Publish S3 Event to SQS Queue




```
"Condition": {
    "ArnLike": {"aws:SourceArn": "arn:aws:s3:
    "StringEquals": {"aws:SourceAccount": "
```

SQS - Message Visibility Timeout

- After a message is polled by a consumer, it becomes invisible to other consumers.
- By default, the 'message visibility timeout' is 30 seconds, to be processed
- After the timeout is over, the message is "visible" in SQS



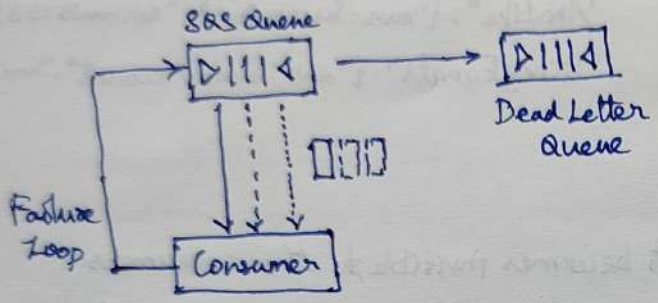
- If a message is not processed within the visibility timeout it will be processed twice
- A consumer could call the ChangeMessageVisibility API to get more time
- If visibility timeout is high (hours) and consumers crashes, re-processing will take a lot of time
- If visibility timeout is too low (seconds), we may get duplicates
- So the visibility timeout should be set something reasonable for your application.

Amazon SQS - Dead Letter Queue

- If the consumer fails to process a message multiple times, it will create Failure Loop
- We can set a threshold of how many times a message can go back to the queue
- After the Maximum-Receives threshold is exceeded, the message goes into

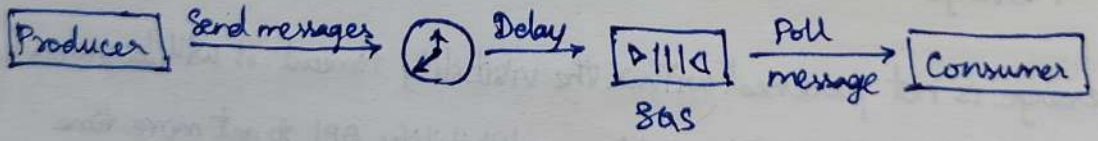
Dead Letter Queue (DLQ)

- It is useful for debugging and root cause analysis of failure
- Make sure to process the messages in the DLQ before they expire:
  - Good to set retention of 14 days in the DLQ



SQS - Delay Queue

- Delay a message (consumers don't see it immediately) up to 15 minutes.
- Default is 0 seconds (messages is available right away)
- Can set a default at queue level
- Can override the default on send using the Delay Seconds parameter



SQS - Long Polling

- When a consumer requests messages from the queue, it can optionally wait for messages to arrive if there are none in the queue
- Long Polling decreases the no. of API calls made to SQS while increasing the efficiency and latency of your application.
- The wait time can be between 1 sec to 20 sec (20 sec preferable)
- Long Polling is preferable to Short Polling
- Long Polling can be enabled at the queue level or at the API level using WaitTimeSeconds

SQS - FIFO queue

- FIFO → First In First out (Ordering of messages in the queue)
- Limited throughput: 300 msg/s without batching, 3000 msg/s with batching
- Exactly-once send capability (by removing duplicates)
- Messages are processed in order by the consumer

AWS SNS - How to publish

a) Topic Publish (using the SDK)

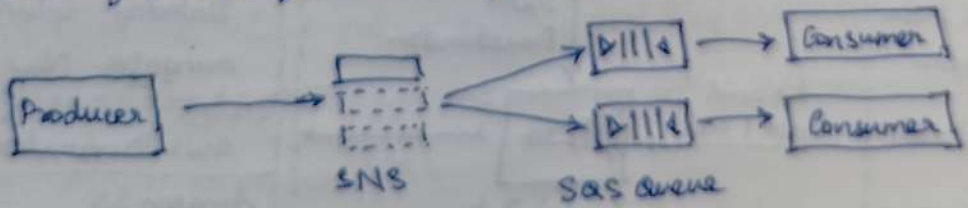
- Create a topic
- Create a subscription (or many)
- Publish to the topic

b) Direct Publish (for mobile apps SDK)

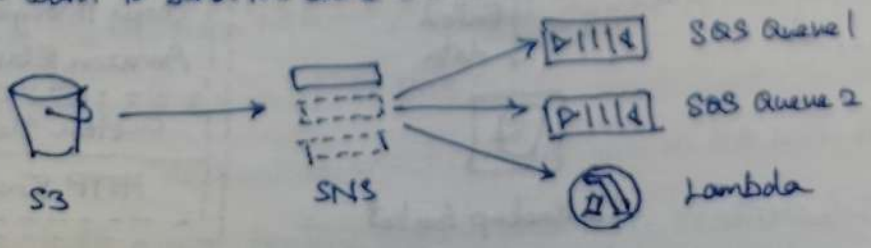
- Create a platform application
- Create a platform endpoint
- Publish to the platform endpoint
- Works with Google GCM, Apple APNS, Amazon ADM...

SNS + SQS: Fan Out

- Push once in SNS, receive in all SQS queues that are subscribers
- Fully decoupled, no data loss
- SQS allows for: data persistence, delayed processing and retries of work
- Ability to add more SQS subscribers over time
- Make sure your SQS queue access policy allows for SNS to write

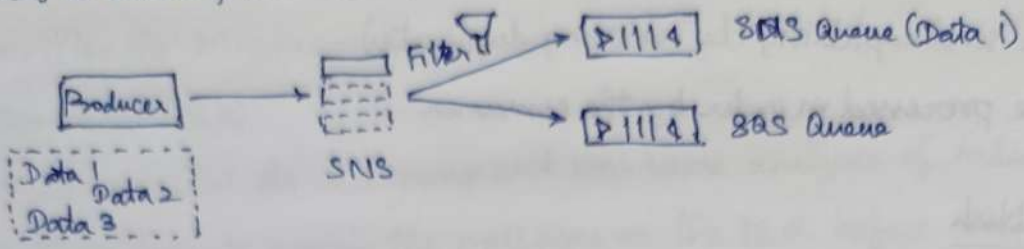


- If you want to send the same S3 event to many SQS queues, use fan-out

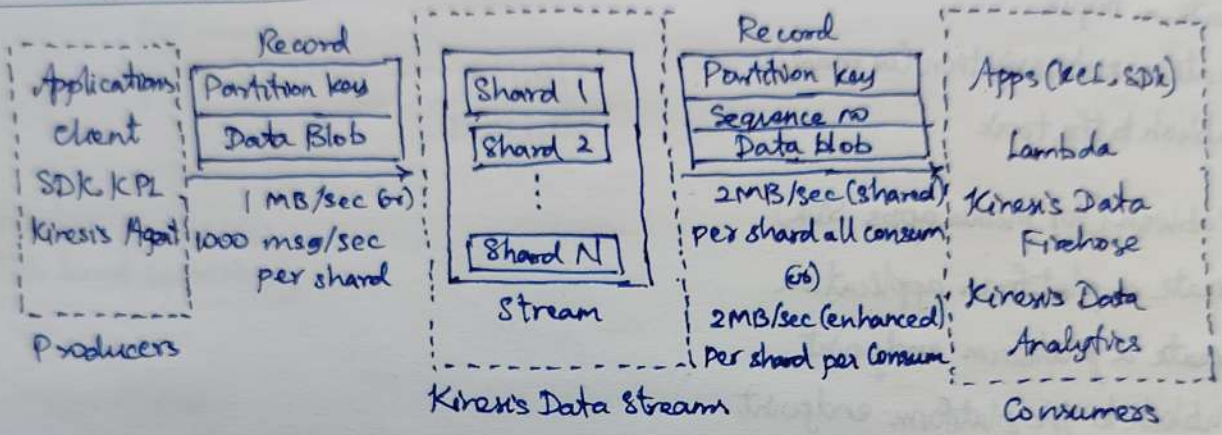


SNS - Message Filtering

- JSON policy used to filter messages sent to SNS topic's subscriptions
- If a subscription doesn't have a filter policy, it receives every message

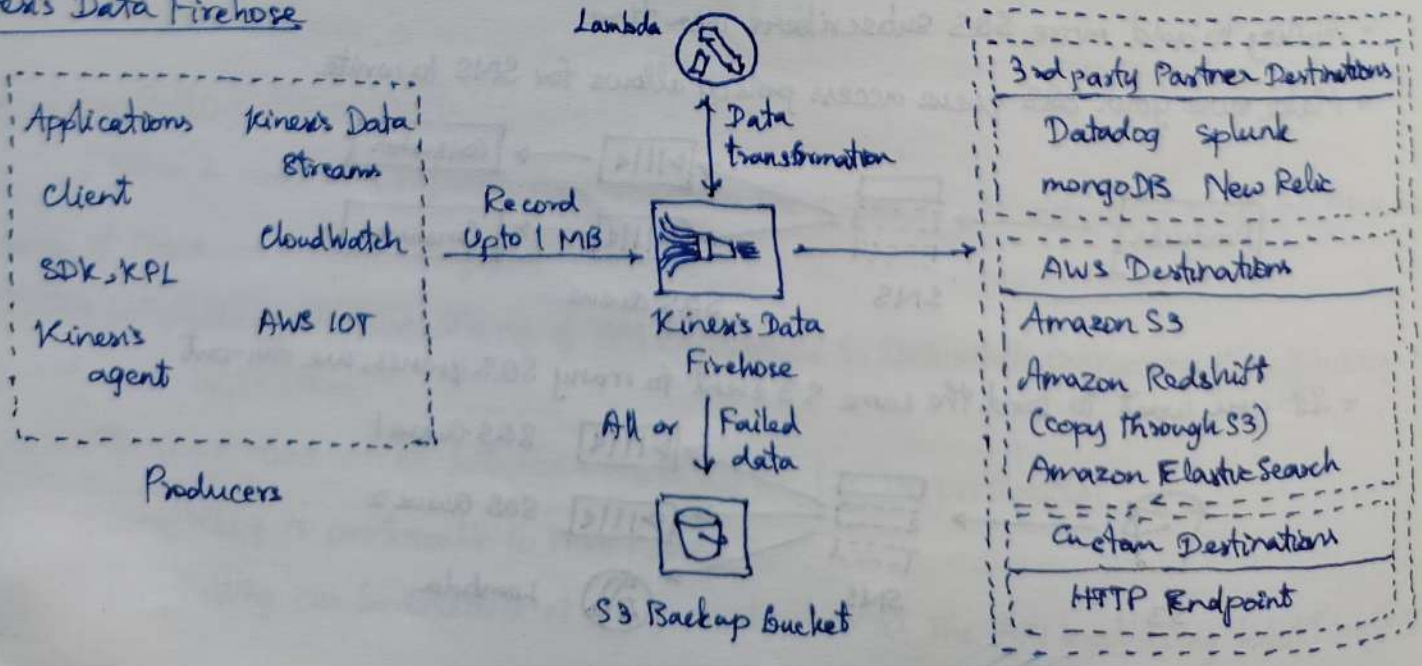


Kinesis Data Streams



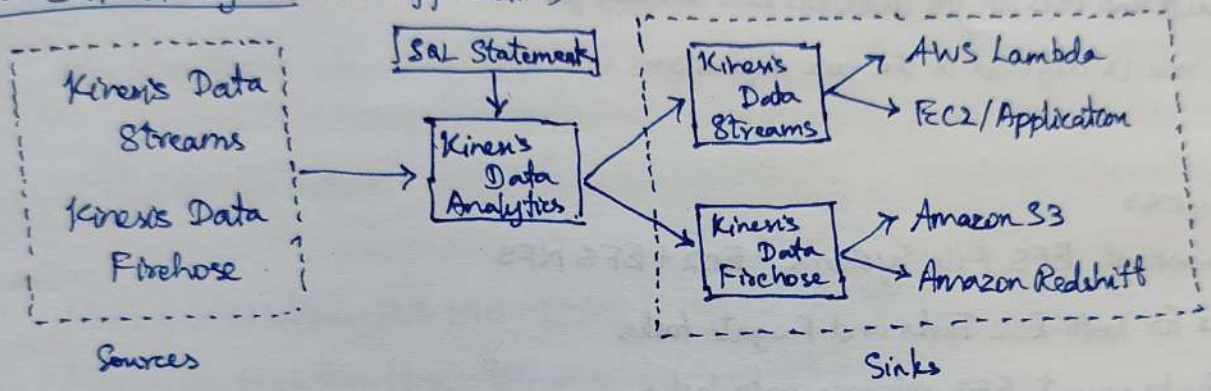
- Billing is per shard provisioned, can have as many shards as you want
- Retention between 1 day (default) to 365 days
- Ability to reprocess (replay) data
- Once data is inserted in Kinesis, it can't be deleted (immutability)
- Data that shares the same partition goes to the same shard (ordering)

Kinesis Data Firehose



- Kinesis Data Firehose is a fully managed service, no administration, automatic scaling
- Pay for data going through Firehose
- Near Real Time
  - 60 seconds latency minimum for non full batches
  - Or minimum 32 MB of data at a time
- Supports many data formats, conversions, transformations, compression
- Supports custom data transformations using AWS Lambda
- Can send failed or all data to a backup S3 bucket

Kinesis Data Analytics (SQL Application)



- Perform real time analytics on kinesis streams using SQL
- Fully managed, no servers to provision. Automatic scaling.
- Real-time analytics, pay for actual consumption rate
- Can create streams out of the real-time queries
- Use cases: Time series analytics, Real-time dashboards, Real-time metrics

Amazon MQ

- SQS & SNS are 'cloud-native' services and they are using proprietary protocols from AWS
- Traditional applications running from on-premise may use open protocols such as MQTT, AMQP, STOMP, Openwire, WSS
- When migrating from onprem to cloud, instead of re-engineering the application to use SQS & SNS, we can use Amazon MQ → Managed Apache ActiveMQ
  - It doesn't scale as much as SQS/SNS
  - It runs on a dedicated machine, can run in HA with failover
  - It has both queue feature (~SQS) and topic features (~SNS)

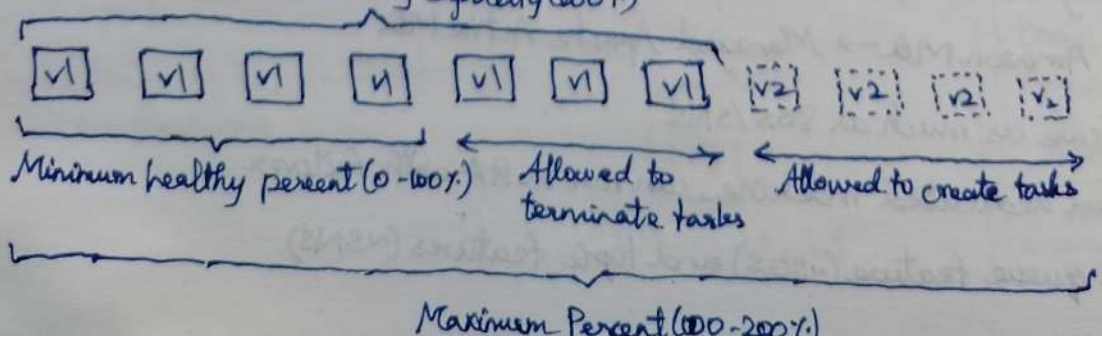
IAM Roles for ECS Tasks

a) EC2 Instance Profile
- Used by the ECS agent
- Makes API calls to ECS service
- Send container logs to cloudWatch logs
- Pull Docker image from ECR
- Reference sensitive data in Secrets Manager or SSM Parameter Store

b) ECS Task Role
- Allow each task to have a specific role
- Use different roles for the different ECS services you run
- Task role is defined in the task definition

ECS Data Volumes
- Make use of EFS File systems - EC2 + EFS NFS
- Works for both EC2 Tasks and Fargate tasks
- Ability to mount EFS volumes onto tasks
- Tasks launced in any AZ will be able to share the same data on the EFS volume.
- Fargate + EFS = serverless + data storage without managing servers.
- Use Case: persistent multi-AZ shared storage for your containers

ECS Rolling Updates
- When updating from v1 to v2, we can control how many tasks can be started and stopped and in which order.
- When you have an ECS updates, you will have 2 settings:
  i) Minimum healthy percent = 100
  ii) Maximum percent = 200



Amazon EKS

- Amazon Elastic Kubernetes Service
- It is a way to launch managed Kubernetes clusters on AWS
- Kubernetes is an open-source system for automatic deployment, scaling and management of containerized (usually Docker) application
- It's an alternative to ECS. similar goal but different API
- EKS supports EC2 if you want to deploy worker nodes or Fargate to deploy serverless containers
- Use Case: If you company is already using Kubernetes on-premises or in another cloud, and wants to migrate to AWS using Kubernetes
- Kubernetes is cloud-agnostic (can be used in any cloud - Azure, GCP)

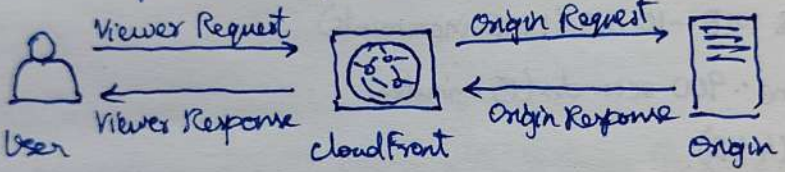
AWS Lambda Limits - per region

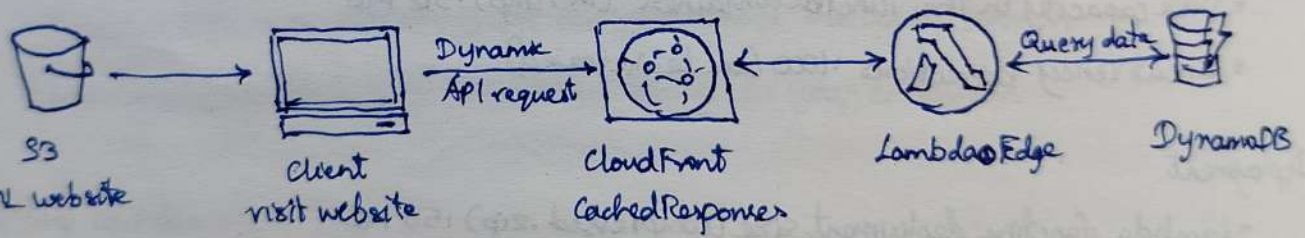
a) Execution
- Memory allocation: 128 MB - 10 GB (1 MB increments)
- Maximum execution time: 900 seconds (15 minutes)
- Environment variables (4 KB)
- Disk capacity in the "function container" (in /tmp): 512 MB
- Concurrency executions: 1000 (can be increased)

b) Deployment
- Lambda function deployment size (compressed .zip): 50 MB
- Size of uncompressed deployment (code + dependencies): 250 MB
- Can use the /tmp directory to load other files at startup
- Size of environment variables: 4 KB

Lambda@Edge

- You have deployed a CDN using CloudFront. What if you wanted to run a global AWS Lambda alongside? Or how to implement request filtering before reaching your application?
- Use Lambda@Edge to deploy Lambda functions alongside your CloudFront CDN
  - Build more responsive applications
  - You don't manage servers. Lambda is deployed globally
  - Customize the CDN content
- You can change CloudFront requests and responses
  - i) Viewer request → After CloudFront receives a request from a viewer
  - ii) Origin request → Before CloudFront forwards the request to the origin
  - iii) Origin response → After CloudFront receives the response from the origin
  - iv) Viewer response → Before CloudFront forwards the response to the viewer
- You can also generate responses to viewers without ever sending the request to the origin





Use Cases:

- Website security and Privacy
- Dynamic web applications at the Edge
- Search Engine Optimization (SEO)
- Intelligently Route across origins and data centers
- Bot mitigation at the Edge
- Real-time Image Transformation
- A/B Testing
- User Authentication and Authorization
- User Prioritization
- User tracking and Analytics

DynamoDB

- The database is already created. We have to just create tables.
- Each table has a Primary Key (must be decided at creation time)
- Each table can have infinite number of items (rows)
- Each item has attributes (can be added over time - can be null)
- Maximum size of an item is 400 KB
- Data types supported are:
  - Scalar Types - String, Number, Binary, Boolean, Null
  - Document Types - List, Map
  - Set Types - String Set, Number Set, Binary Set

DynamoDB - Read/Write Capacity Modes

- Control how you manage your table's capacity (read/write throughput)

a) Provisioned Mode (default)

- You specify the number of reads/writes per second
- You need to plan capacity beforehand
- Pay for provisioned Read Capacity Units (RCU) & Write Capacity Units (WCU)
- Possibility to add auto-scaling mode for RCU & WCU

b) On-Demand Mode

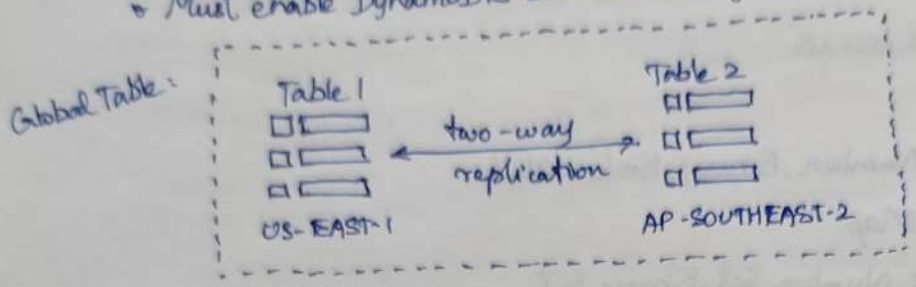
- Read/writes automatically scale up/down with your workloads
- No capacity planning needed
- Pay for what you use, more expensive ($$$)
- Great for unpredictable workloads

DynamoDB Streams

- Ordered stream of item-level modifications (create/update/delete) in a table.
- Stream records can be sent to Kinesis Data Streams, read by AWS Lambda & KCL App
- Data retention for up to 24 hours
- Use Cases: react to changes in real-time (welcome email to users)
  - Analytics
  - Insert into derivative tables or into ElasticSearch
  - Implement Cross-region replication

DynamoDB Global Table

- Make a DynamoDB table accessible with low latency in multiple-regions
- Active-Active replication
- Applications can Read and Write to the table in any region
- Must enable DynamoDB Streams as a pre-requisite



DynamoDB - Time to Live (TTL)

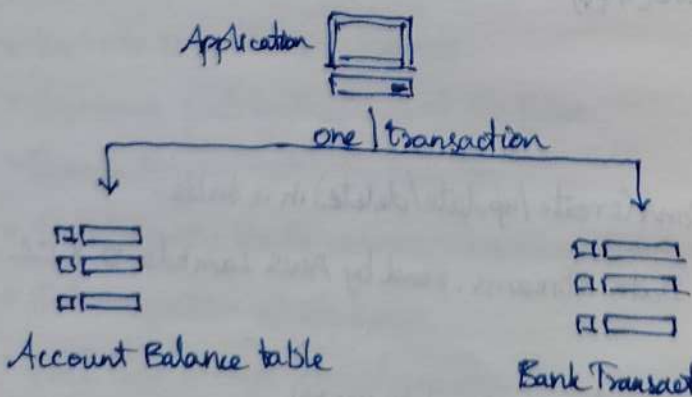
- Automatically delete items after an expiry timestamp
- Use cases: reduce stored data by keeping only current items
  adhere to regulatory obligations

DynamoDB - Indexes

- Global Secondary Index (GSI) or Local Secondary Index (LSI)
- Allow to query on attributes other than the Primary key
- With Indexes, we can query by other attributes of the items

DynamoDB - Transactions

- Transaction is written to both tables or none



Read Modes:
- Eventual Consistency
- Strong Consistency
- Transactional

Write Modes
- Standard
- Transactional

- Provides Atomicity, Consistency, Isolation and Durability (ACID)
- Consumes 2x WCUs & RCUs
- Use cases: financial transactions, managing orders

API Gateway - Endpoint Types

a) Edge-Optimized (default): For global clients
- Requests are routed through the CloudFront Edge Locations (improves latency)
- The API Gateway still lives in only one region

b) Regional
- For clients within the same region
- Could manually combine with CloudFront (more control over the caching strategies and the distribution

c) Private
- Can only be accessed from your VPC using an interface VPC endpoint (ENI)
- Use a resource policy to define access

API Gateway - Security

IAM Permissions - For users within AWS
- Create an IAM policy authorization and attach to User/Role
- API Gateway verifies IAM permissions passed by the calling application
- Good to approve access within your own infrastructure
- Leverages 'Sig v4' capabilities where IAM credential are in headers

Lambda Authorizer (Custom Authorizer)
- Uses AWS Lambda to validate the token in header being passed. OAuth/SAML/3rd party
- Option to cache result of authentication.
- Lambda must return an IAM policy for the user

Cognito User Pools
- Cognito fully manages user lifecycle
- API Gateway verifies identity automatically from AWS Cognito
- No custom implementation required
- Cognito only helps with Authentication, not Authorization
- Authentication - Cognito User Pools
  Authorization - API Gateway Methods

AWS Cognito

We want to give our users an identity so that they can interact with our application.

a) Cognito User Pools:
- Signin functionality for app users
- Integrate with API Gateway

b) Cognito Identity Pools (Federated Identity)
- Provide AWS Credentials to users so they can access AWS resources directly
- Integrate with Cognito User Pools as an identity provider

c) Cognito Sync
- Synchronize data from device to Cognito. Store preferences, configuration, state of app
- Might be deprecated and replaced by AppSync. Offline capability, requires FIP not CUP

AWS Cognito User Pools (CUP)

- Create a serverless database of user for your mobile apps
- Simple login: Username (or email) / password combination
- Possibility to verify emails/phone numbers and add MFA
- Can enable Federated Identities (Facebook, Google, SAML).
- Sends back a JSON Web Token (JWT)
- Can be integrated with API Gateway for authentication and Application Load Balancer

Federated Identity Pools

- Provide direct access to AWS Resources from Client side
- Log in to federated identity provider or remain anonymous
- Get temporary AWS credentials back from the Federated Identity Pool
- These credentials come with a pre-defined IAM policy stating their permissions
- Eg. Provide (temporary) access to write to S3 bucket using Facebook login

AWS SAM

- Serverless Application Model
- Framework for developing and deploying serverless applications
- All the configuration is YAML code
  - Lambda Functions
  - DynamoDB tables
  - API Gateway
  - Cognito User Pools
- SAM can help you to run Lambda, API Gateway, DynamoDB locally
- SAM can use CodeDeploy to deploy Lambda functions

Redshift

- Data is loaded from S3, DynamoDB, DMS, other DBs
- From 1 node to 128 nodes, up to 128 TB of space per node
- Leader node: for query planning, results aggregation
- Compute node: for performing the queries, send results to leader
- Redshift Spectrum: perform queries directly into S3 (no need to load)
- Backup & Restore, Security VPC/IAM/KMS, Monitoring
- Redshift Enhanced VPC Routing: Copy/unload goes through VPC

Snapshots & DR

- Redshift has no 'Multi-AZ' mode
- Snapshots are point-in-time backups of a cluster, stored internally in S3
- Snapshots are incremental (only what has changed is saved)
- You can restore a snapshot into a new cluster
- Automated: every 8 hours, every 5 GB, or on a schedule. Set retention
- Manual: snapshot is retained until you delete it.
- You can configure Amazon Redshift to automatically copy snapshots (automated or manual) of a cluster to another AWS Region

ElasticSearch / OpenSearch

- You can search any field, even partially matches
- It's common to use ElasticSearch as a complement to another database
- It also has some usage for Big Data applications
- You can provision a cluster of instances
- Built-in integrations: Amazon Kinesis Data Firehose, AWS IoT and Amazon CloudWatch Logs
- Security through Cognito & IAM, KMS encryption, SSL & VPC
- Comes with Kibana (visualization) & Logstash (log ingestion) - ELK stack
- Eg: In DynamoDB, you can find only by primary key or indexes. With ElasticSearch you can search any field
- ElasticSearch = Search / Indexing

CloudWatch Custom Metrics

- Possibility to define and send your own custom metrics to CloudWatch
- Eg: memory (RAM) usage, disk space, number of logged in users
- Use API call PutMetricData
- Ability to use dimensions (attributes) to segment metrics
  - Instance.id
  - Environment.name
- Metric resolution (StorageResolution API parameter - two possible value)
  - Standard: 1 minute (60 seconds)
  - High Resolution: 1/5/10/30 second(s) - Highest cost
- Important: Accepts metric data points two weeks in the past and two hours in the future (make sure to configure your EC2 instance time correctly)

Cloudwatch Dashboards

- Custom dashboards for quick access to key metrics and alarms. Dashboards are global
- It can include graphs from different AWS accounts and regions. You can change the time zone & time range of the dashboards. You can setup automatic refresh (10s, 1m, 2m, 5m, 15m)
- It can be shared with people who don't have an AWS account.
  (public, email address, 3rd party SSO provider through Amazon Cognito)

CloudWatch Logs

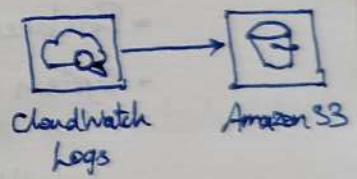
- Log groups : arbitrary name, usually representing an application
- Log stream : instances within application / log files / containers
- Can define log expiration policies (never expire, 30 days, etc...)
- It can send logs to:
  - Amazon S3 (exports)
  - Kinesis Data Streams
  - Kinesis Data Firehose
  - AWS Lambda
  - ElasticSearch

CloudWatch - Metric Filter & Insights

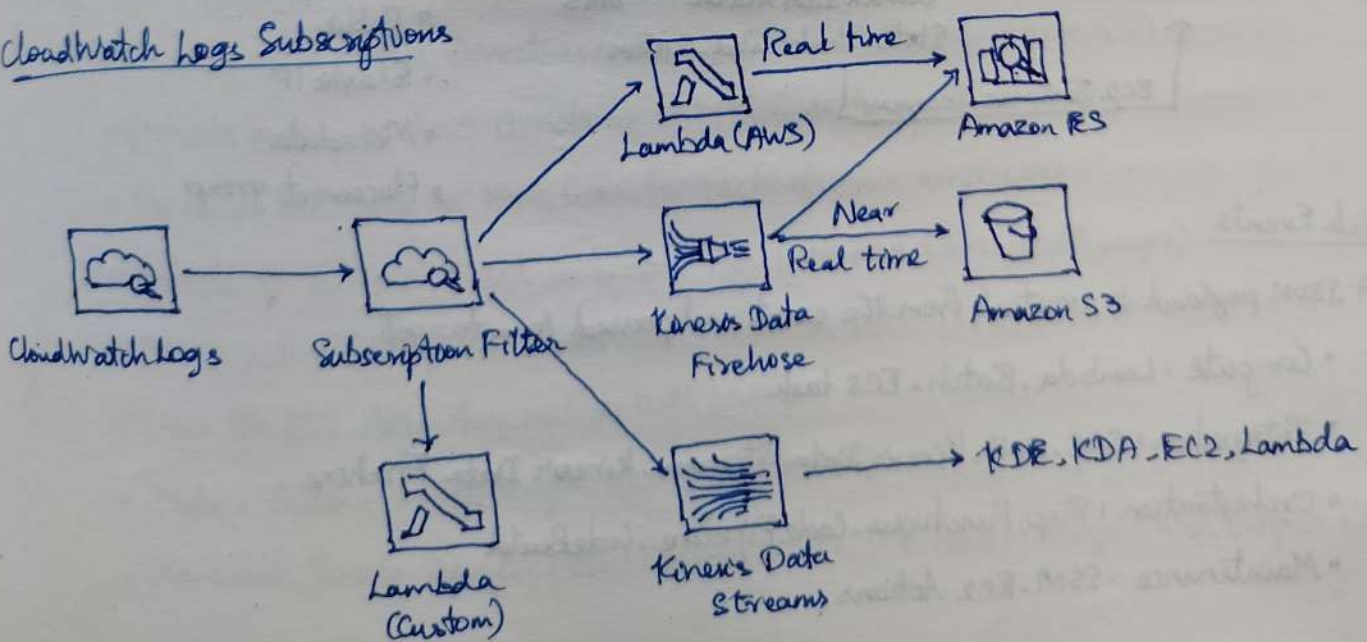
- CloudWatch Logs can use filter expressions
  - For eg: find a specific IP inside of a log
  - Or count occurences of 'Error' in your logs
- Metric filters can be used to trigger CloudWatch alarms
- Logs Insights can be used to query logs and add queries to CloudWatch Dashboards

CloudWatch - S3 Export

- Log data can take up to 12 hours to become available for export
- The API call is CreateExportTask
- Not near-real time or real time, use logs subscriptions instead

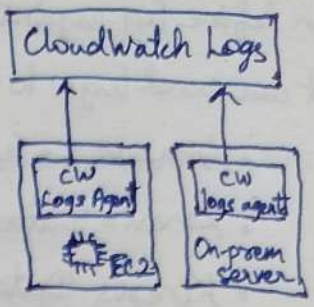


CloudWatch Logs Subscriptions



CloudWatch Logs for EC2

- By default, no logs from your EC2 machine will go to CloudWatch.
- You need to run a CloudWatch agent on EC2 to push the log files you want.
- Make sure IAM permissions are correct
- CloudWatch log agent can be setup on-premises too

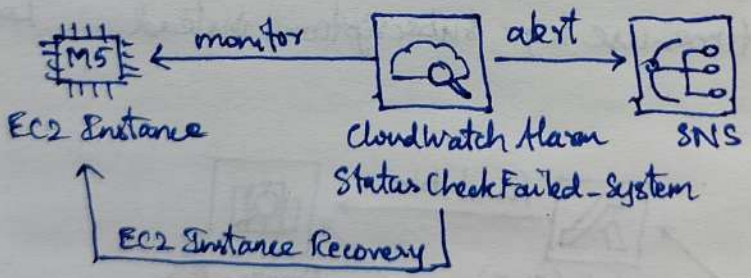


a) CloudWatch Logs Agent
- Old version of the agent
- Can only send to CloudWatch Logs

b) CloudWatch Unified Agent
- Collect additional system level metrics such as RAM, processes, etc...
- Collect logs to send to CloudWatch logs
- Centralized configurations using SSM Parameter Store

EC2 Instance Recovery

- Status Check:
  - Instance status → check the EC2 VM
  - System status → check the underlying hardware



Recovery
- Same Private
- Public
- Elastic IP
- Metadata
- Placement group

CloudWatch Events

- A JSON payload is created from the event and passed to a target...
  - Compute: Lambda, Batch, ECS task
  - Integration: SQS, SNS, Kinesis Data Streams, Kinesis Data Firehose
  - Orchestration: Step Functions, CodePipeline, CodeBuild
  - Maintenance: SSM, EC2 Actions

Amazon EventBridge Schema Registry

- EventBridge can analyze the events in your bus and infer the schema
- The schema registry allows you to generate code for your application, that will know in advance how data is structured in the event bus.
- Schema can be versioned.
- EventBridge allows extension to add event buses for your custom applications and your third-party SaaS apps.

Identity Federation in AWS

- Federation lets users outside of AWS to assume temporary role for accessing AWS resources.
- These users assume identity provided access role.
- Federation can have many flavors:
  - SAML 2.0
  - Custom Identity Broker
  - Web Identity Federation w/wo Amazon Cognito
  - Single Sign-on
  - Non SAML with AWS Microsoft AD
- Using federation, you don't need to create IAM users (user management is outside of AWS)

SAML 2.0 Federation

- To integrate Active Directory / ADFS with AWS (or any SAML 2.0)
- Provide access to AWS Console or CLI (through temporary creds)
- No need to create an IAM user for each of your employees
- Needs to setup a trust between AWS IAM and SAML (both ways)
- SAML 2.0 enables web-based, cross domain SSO
- Uses the STS API: AssumeRole with SAML
- Note: federation through SAML is the 'old way' of doing things
- Amazon Single Sign On (SSO) Federation is the new managed and simpler way.

Custom Identity Broker Application

- Use only if identity provider is not compatible with SAML 2.0
- The identity broker must determine the appropriate IAM policy
- Uses the STS API: AssumeRole or GetFederationToken

IAM Conditions

- aws:SourceIP: restrict the client IP from which the API calls are being made

```
"aws:SourceIp": [
    "192.0.2.0/24",
    "203.0.113.0/24"
]
```

- aws:RequestedRegion: restrict the region the API calls are made to

```
"aws:RequestedRegion": [
    "eu-central-1",
    "eu-east-1"
]
```

- Restrict based on tags

```
"ec2:ResourceTag/Project": "~~",
"aws:PrincipalTag/Department": "~~"
```

- Force MFA

```
"BoolIfExists": {"aws:MultiFactorAuthPresent": false}
```

IAM Roles vs Resource Based Policies

- When you assume a role (user, application or service) you give up your original permissions and take the permissions assigned to the role.
- When using a resource based policy, the principal doesn't have to give up his permissions
- Eg: User on Account A needs to scan a DynamoDB table in Account A and dump it in an S3 bucket in Account B
- Supported by: Amazon S3 buckets, SNS topics, SQS queues.

IAM Permission Boundary

- IAM Permission Boundaries are supported by users and roles (not groups)
- Advanced feature to use a managed policy to set the maximum permissions an IAM entity can get.

Example:

```
"Action": [
    "S3:*",
    "cloudwatch:*",
    "ec2:*"
]
```

IAM Permission Boundary

+

```
"Action": "iam:CreateUser"
```

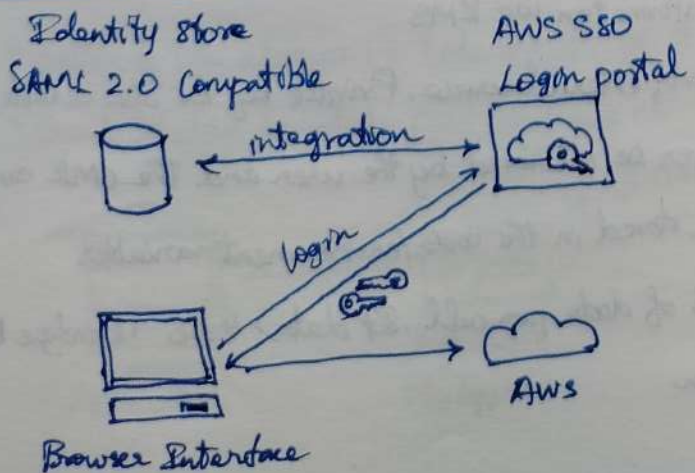
IAM Permission

= No permission

AWS Resource Access Manager (RAM)

- Share AWS resources that you own with other AWS accounts
- Share with any account or within your Organization
- Avoid resource duplication
- VPC Subnets
    - Allow to have all the resources launched in the same subnets
    - Must be from the same AWS Organizations
    - Cannot share security groups and default VPC
    - Participants can share their own resources in there
    - Participants can't view, modify, delete resources that belong to others or owner
- AWS Transit Gateway, Route53 Resolver Rules, License Manager Configurations

AWS Single-Sign On (SSO)



KMS - Customer Master Key (CMK) Types

a) Symmetric (AES-256 keys)

- First offering of KMS - single encryption key that is used to Encrypt and Decrypt
- AWS services that are integrated with KMS uses Symmetric CMKs
- Necessary for envelope encryption
- You never get access to the key unencrypted (must call KMS API to use)

b) Asymmetric (RSA & ECC keypairs)

- Public (Encrypt) and Private key (Decrypt) pair
- Used for Encrypt/Decrypt or Sign/Verify operations
- The public key is downloadable but you can't access the Private key unencrypted
- Use Case: encryption outside of AWS by users who can't call the KMS API

AWS KMS

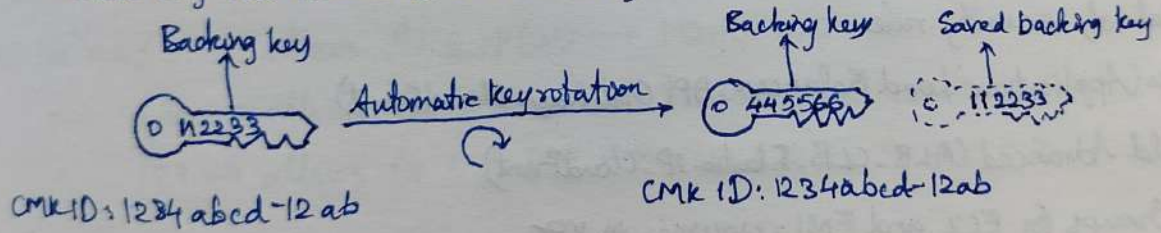
- Able to fully manage the keys & policies
  - Create
  - Rotation policies
  - Disable / Enable
- Able to audit key usage (using CloudTrail)
- Three types of Customer Master Keys (CMK):
  - AWS Managed Service Default CMK: free
  - User Keys created in KMS: $1/month
  - User Keys imported (must be 256-bit Symmetric key): $1/month
- + pay for API call to KMS ($0.03/10000 calls)
- Anytime you need to share sensitive information use KMS
  - Database passwords, Credentials of external service, Private key for SSL certificates
- The CMK used to encrypt data can never be retrieved by the user and the CMK can be rotated for extra security. Encrypted keys can be stored in the code/environment variables
- KMS can help only in encrypting 4 KB of data per call. If data > 4 KB → Envelope Encryption
- KMS keys are bound to a specific region

KMS Key Policies

- Control access to KMS keys. 'similar' to S3 bucket policies
- Difference: you cannot control access without them
- Default KMS key Policy:
  - Created if you don't provide a specific KMS Key Policy
  - Complete access to the IAM policies to the KMS key
  - Complete access to the key to the root user → entire AWS account.
- Custom KMS key Policy:
  - Define users, roles that can access the KMS key
  - Define who can administer the key
  - Useful for cross-account access of your KMS key

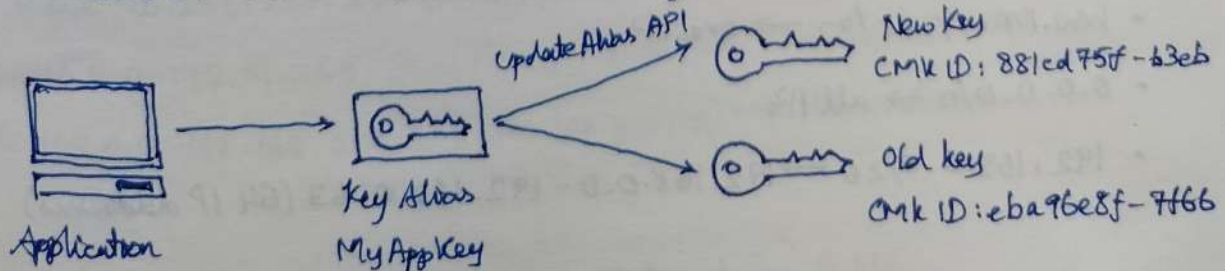
KMS Key Rotation

- For Customer-managed CMK (not AWS managed CMK)
- If enabled: automatic key rotation happens every 1 year
- Previous key is kept active so you can decrypt old data
- New key has the same CMK ID (only the backing key is changed)



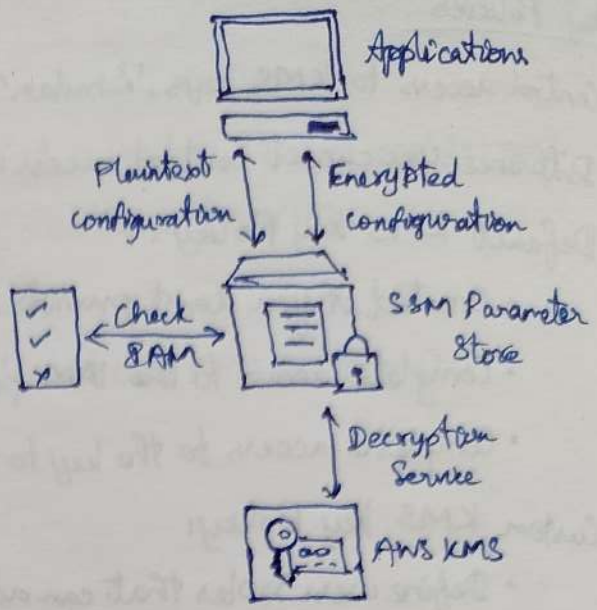
KMS Manual Key Rotation

- When you want to rotate key every 90 days, 180 days, etc...
- New key has a different CMK ID. Keep the previous key active so you can decrypt old data
- Better to use aliases (to hide the change of key for the application)
- Good solution to rotate CMK that are not eligible for automatic rotation (asymmetric CMK)



SSM Parameter Store

- Secure storage for configuration and secrets
- Optional Seamless Encryption using KMS
- Serverless, Scalable, durable, easy SDK
- Version tracking of configuration / secrets
- Configuration management using Path & IAM
- Notifications with CloudWatch events
- Integration with cloudFormation



Parameter Policies

- SPt is for advanced parameters, not for standard parameters
- Allow to assign a TTL to a parameter (expiration date) to force updating or deleting sensitive data such as passwords
- Can assign multiple policies at a time

AWS Firewall Manager

- Manage rules in all accounts of an AWS Organization
- Common set of security rules
- WAF rules (Application Load Balancer, API Gateway, CloudFront)
- AWS Shield Advanced (ALB, CLB, Elastic IP, CloudFront)
- Security Groups for EC2 and ENI resources in VPC

CIDR - IPv4

- Classless Inter-Domain Routing - a method for allocating IP addresses
- Used in Security Groups rules and AWS networking in general
- They help to define an IP address range:
  - WW.XX.YY.ZZ/32 → one IP
  - 0.0.0.0/0 → all IPs
  - 192.168.0.0/26 → 192.168.0.0 - 192.168.0.63 (64 IP addresses)

Understanding CIDR - IPv4

- A CIDR consists of two components
- Base IP
  - Represents an IP contained in the range (xx.xx.xx.xx)
  - Eg: 10.0.0.0, 192.168.0.0
- Subnet Mask
  - Defines how many bits can change in the IP
  - Eg: /0, /24, /32
  - Can take two forms:
    - /8 → 255.0.0.0
    - /16 → 255.255.0.0
    - /24 → 255.255.255.0
    - /32 → 255.255.255.255

Octets

IP: ___ . ___ . ___ . ___
(1st, 2nd, 3rd, 4th)

- /32 → no octet can change
- /24 → last octet can change
- /16 → last 2 octets can change
- /8 → last 3 octets can change
- /0 → all octets can change

Subnet Mask

- Basically allows part of the underlying IP to get additional next values from the base IP

192.168.0.0 /32 → allows for 1 IP (2^0) → 192.168.0.0

192.168.0.0/31 → allows for 2 IP (2^1) → 192.168.0.0 - 192.168.0.1

- /30 → allows for 4 IP (2^2) → 192.168.0.0 - 192.168.0.3
- /29 → allows for 8 IP (2^3) → 192.168.0.0 - 192.168.0.7

Public vs Private IP (IPv4)

- The Internet Assigned Numbers Authority (IANA) established certain blocks of IPv4 addresses for the use of private (LAN) and public (Internet) addresses.

Private IP can only allow certain values:

- 10.0.0.0 - 10.255.255.255 (10.0.0.0/8) ⟹ In Big Networks
- 172.16.0.0 - 172.31.255.255 (172.16.0.0/12) ⟹ AWS default VPC
- 192.168.0.0 - 192.168.255.255 (192.168.0.0/16) ⟹ Home networks

Public IP

- All the rest of the IP addresses on the Internet are Public

Default VPC

- All new AWS accounts have a default VPC
- New EC2 instances are launched into the default VPC if no subnet is specified
- Default VPC has Internet connectivity and all EC2 instances inside it have public IPv4 addresses
- We also get a public and private IPv4 DNS names

VPC in AWS - IPv4

- VPC → Virtual Private Cloud
- You can have multiple VPCs in an AWS region (max 5 per region - soft limit)
- Max. CIDR per VPC is 5, for each CIDR:
  - Min size is /28 (16 IP addresses)
  - Max size is /16 (65536 IP addresses)
- Because VPC is private, only the Private IPv4 ranges are allowed
- Your VPC CIDR should not overlap with your other networks (Eg. Corporate)

VPC - Subnet IPv4

- AWS reserves 5 IP addresses (first 4 & last 1) in each subnet
- These 5 IP addresses are not available for use and can't be assigned to an EC2 instance
- Example: If CIDR block 10.0.0.0/24, then reserved IP addresses are
  - 10.0.0.0 → Network Address
  - 10.0.0.1 → reserved by AWS for the VPC router
  - 10.0.0.2 → reserved by AWS for mapping to Amazon provided DNS
  - 10.0.0.3 → reserved by AWS for future use
  - 10.0.0.255 → Network Broadcast Address. AWS does not support broadcast in a VPC Therefore the address is reserved.

- Note: If you need 29 IP addresses for EC2 instances:

  ✗ You can't choose a subnet of size /27 (32 IP addresses, 32 - 5 = 27)

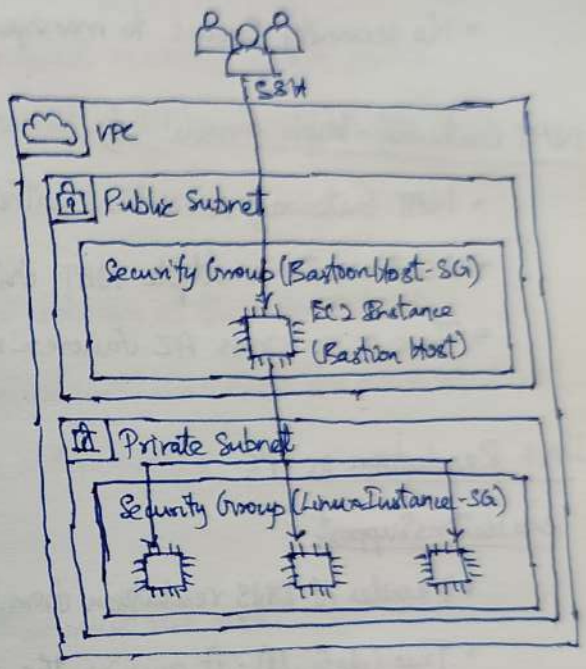
  ✓ You need to choose a subnet of size /26 (64 IP addresses, 64 - 5 = 59)

Internet Gateway (IGW)

- Allows resources (e.g. EC2 instances) in a VPC connect to the Internet
- It scales horizontally and is highly available and redundant
- Must be created separately from a VPC
- One VPC can only be attached to one IGW and vice versa
- Internet Gateways on their own do not allow Internet access. Route table must also be edited.

Bastion Hosts

- We can use a Bastion Host to SSH into our private EC2 instances.
- The bastion is in the public subnet which is then connected to all other private subnets.
- Bastion Host security group must be tightened.
- Note: Make sure the bastion host only has port 22 traffic from the IP address you need, not from the security groups of your other EC2 instances



NAT Instance

- NAT → Network Address Translation
- Allows EC2 instances in private subnets to connect to the Internet
- Must be launched in a public subnet
- Must disable EC2 setting: Source / Destination Check
- Must have Elastic IP attached to it
- Route tables must be configured to route traffic from private subnets to the NAT Instance

Comments

- Pre-configured Amazon Linux AMI is available - Reached end of support on Dec 31, 2020
- Not highly available / resilient group setup out of the box
- Internet traffic bandwidth depends on EC2 instance type
- You must manage Security Groups & rules

NAT Gateway

- AWS-managed NAT, higher bandwidth, high availability, no administration
- Pay per hour for usage and bandwidth
- NATGW is created in a specific Availability zone, uses an Elastic IP
- Can't be used by EC2 instance in the same subnet (only from other subnets)
- Requires an IGW (Private Subnet → NATGW → IGW)
- 5 Gbps of bandwidth with automatic scaling up to 45 Gbps
- No security Groups to manage/required.

NAT Gateway - High Availability

- NAT Gateway is resilient within a single Availability Zone
- Must create multiple NAT Gateways in multiple AZs for fault-tolerance
- There is no cross AZ failover needed because if an AZ goes down it doesn't need NAT

DNS Resolution in VPC

enableDnsSupport

- Decides if DNS resolution from Route 53 Resolver server is supported for the VPC
- True (default) - it queries the Amazon Provider DNS Server at 169.254.169.253 or the reserved IP address at the base of the VPC IPv4 network range plus two (.2)

enableDnsHostnames

- By default,
  - True → default VPC
  - False → newly created VPCs
- Won't do anything unless enableDNSSupport = true
- If True, assigns public hostname to EC2 instance if it has a public IPv4

EC2 Instance

Private DNS : ip-private-ipv4-address.ec2.internal

Public DNS : ec2-public-ipv4-address.compute-1.amazonaws.com

NACL

- Network Access Control List are like a firewall which control traffic from and to subnets
- One NACL per subnet, new subnets are assigned the Default NACL
- NACL Rules:
  - Rules have a number (1-32766), higher precedence with a lower number
  - First rule match will drive the decision
  - Eg: If you define #100 Allow 10.0.0.10/32 & #200 Deny 10.0.0.10/32
    the IP address will be allowed because 100 has a higher precedence over 200
  - The last rule is an asterisk (*) and denies a request in case of no rule match
  - AWS recommends adding rules by increment of 100, just so that new rules can be added inbetween.
- Newly created NACLs will deny everything
- NACL are a great way of blocking a specific IP address at the subnet level
- Default NACL accepts everything inbound/outbound with the subnets it's associated with
- Do not modify the default NACL, instead create custom NACLs

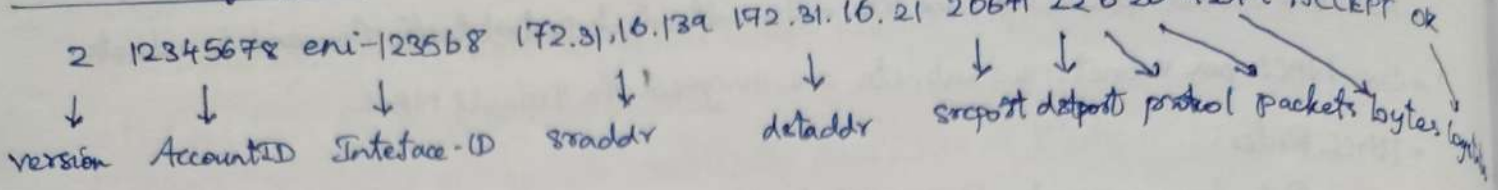
Ephemeral Ports

- For any two endpoints to establish a connection, they must use ports.
- Client connects to a defined port, and expects a response on an ephemeral port
- Different Operating Systems use different port ranges, examples:
  - IANA & MS Windows 10 → 49152 - 65535
  - Many Linux Kernals → 32768 - 60999

VPC - Reachability Analyzer

- A network diagnostics tool that troubleshoots network connectivity between two endpoints in VPC
- It builds a model of the network configuration, then checks the reachability based on these configurations (it doesn't send packets)
- When the destination is
  - Reachable - it produces hop-by-hop details of the virtual network path
  - Not reachable - it identifies the blocking component(s). Eg: Configuration issues.
- Use cases: Troubleshoot connectivity issues, ensure network configuration is as intended

VPC Flow Logs Syntax



- srcaddr & dstaddr → help identify problematic IP
- srcport & dstport → help identify problematic ports
- Action - success or failure of the request due to Security Group/NACL
- Can be used for analytics on usage patters or malicious behaviour
- Query VPC flow logs using Athena on S3 or CloudWatch Logs insights

AWS Site-to-Site VPN

a) Virtual Private Gateway (VGW)

- VPN concentrator on the AWS side of the VPN connection
- VGW is created and attached to the VPC from which you want to create the Site-to-Site VPN connection
- Possibility to customize the ASN (Autonomous System Number)

b) Customer Gateway (CGW)

- Software application or physical device on customer side of the VPN connection

Customer Gateway Device (on-premises)

What IP address to use?

- Public Internet-routable IP address for your Customer Gateway Device
- If it's behind a NAT device that's enabled for NAT traversal (NAT-T) use the public IP address of the NAT device

Important: Enable Route Propagation for the virtual Private Gateway in the route table that is associated with your subnets

Note: If you need to ping your EC2 instances from on-premises, make sure you add the ICMP protocol on the inbound of your security groups

AWS VPN CloudHub

- Provide secure connection/communication between multiple sites, if you have multiple VPN connections
- Low cost hub and spoke model for primary or secondary network connectivity between different locations (VPN only)
- It's a VPN connection so it goes over the public Internet
- To set it up, connect multiple VPN connections on the same VGW setup dynamic routing and configure route tables.

Direct Connect (DX)

- Provides a dedicated private connection from a remote network to your VPC.
- Dedicated connection must be setup between your DC and AWS Direct Connect locations.
- You need to setup a Virtual Private Gateway on your VPC
- Access public resources (S3) and private (EC2) on same connection
- Supports both IPv4 and IPv6
- Use cases:
  - Increase bandwidth throughput - working with large data sets - lower cost
  - More consistent network experience - applications using real-time data feeds
  - Hybrid Environments (on prem + cloud)
- If you want to setup a Direct Connect to one or more VPC in many different regions (same account), you must use a Direct Connect Gateway

Connection Types

a) Dedicated Connections: 1 Gbps and 10 Gbps capacity
- By Physical ethernet port dedicated to a customer
- Request made to AWS first, then completed by AWS Direct Connect Partners.

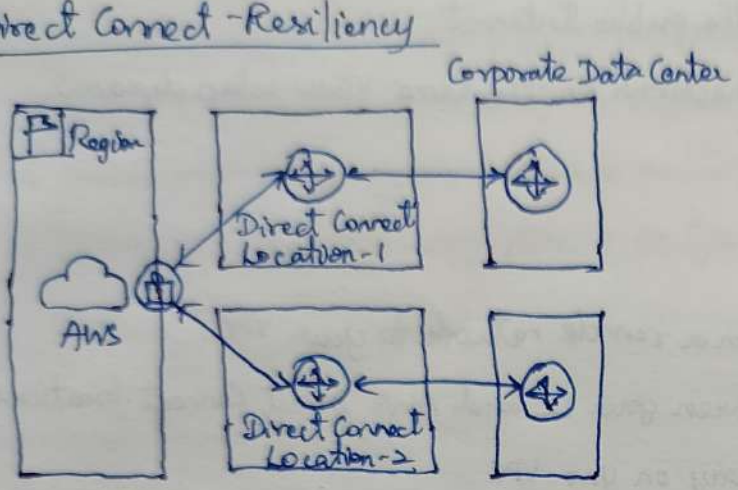
b) Hosted Connections: 500 Mbps, 50 Mbps, to 10 Gbps
- Connection requests are made via AWS Direct Connect Partners
- Capacity can be added or removed on demand
- 1, 2, 5, to 10 Gbps available at select AWS Direct Connect Partners.

Note: For both the connections the lead times are often longer than 1 month to establish a new connection

Direct Connect - Encryption

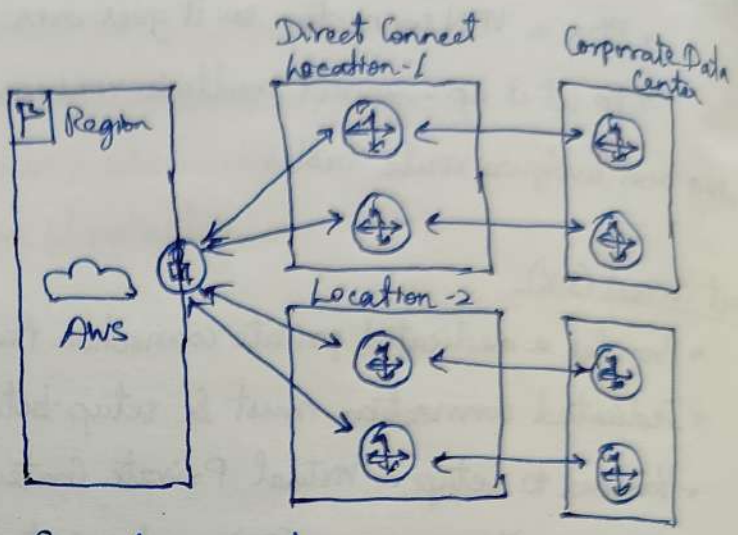
- Data in transit is not encrypted but is private
- AWS Direct Connect + VPN provides an IPsec-encrypted private connection
- Good for an extra level of security but slightly more complex to put in place

Direct Connect - Resiliency



One connection at multiple locations

HIGH RESILIENCY FOR CRITICAL WORKLOADS

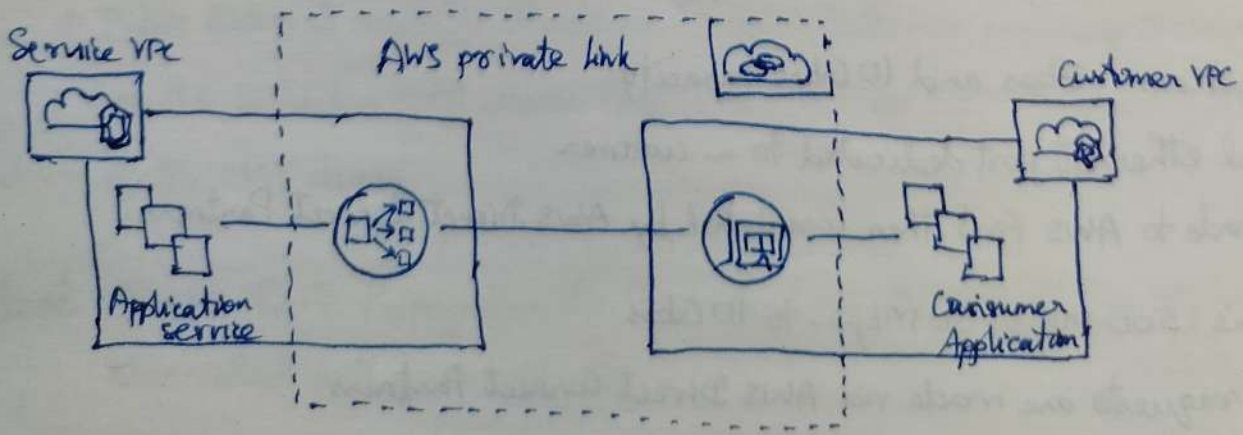


Separate connections in more than one location

MAXIMUM RESILIENCY FOR CRITICAL WORKLOADS

AWS PrivateLink (VPC Endpoint Services)

- Most secure and scalable way to expose a service to 1000s of VPC (own or other accounts)
- Does not require VPC peering, internet gateway, NAT, route tables
- Requires a network load balancer (Service VPC) and ENI (Customer VPC) or GWLB
- If the NLB is in multiple AZ, and the ENIs in multiple AZ, the solution is fault tolerant



EC2-Classic & AWS ClassicLink (deprecated)

- EC2-Classic → instances run in a single network shared with other customers
- ClassicLink → Allows you to link EC2-Classic instances to a VPC in your account.
  - Must associate a security group
  - Enables communication using private IPv4 addresses
  - Removes the need to make use of public IPv4 addresses or Elastic IP addresses.

Transit Gateway: Site to Site VPN ECMP

- ECMP → Equal cost multi path routing
- Routing strategy to allow to forward a packet over multiple best path
- Use case: create multiple Site-to-Site VPN connections to increase the bandwidth of your connection to AWS

VPC-Traffic Mirroring

- Allow you to capture and inspect network traffic in your VPC
- Route the traffic to security appliances that you manage
- Capture the traffic
  - From (Source) - ENIs
  - To (Targets) - an ENI or a Network Load Balancer
- Capture all packets or capture the packets of your interest (optionally truncate packets)
- Source and Target can be in the same VPC or different VPCs (VPC Peering)
- Use cases: content inspection, threat monitoring, troubleshooting...

IPv6

- IPv6 is successor of IPv4. It is designed to provide 3.4×10^{38} unique IP addresses.
- Every IPv6 address is public and Internet-routable (no private range)
- Format → x.x.x.x.x.x.x.x (x is hexadecimal, range can be from 0000 to ffff)
- Examples:
  - 2001:db8:3333:4444:5555:6666:7777:8888
  - 2001:db8:3333:4444:cccc:dddd:eeee:ffff
  - :: → all 8 segments are zero
  - 2001:db8:: → last 6 segments are zero | ::1234:5678 → first 6 segments are zero
  - 2001:db8::1234:5678 → middle 4 segments are zero

IPv6 in VPC

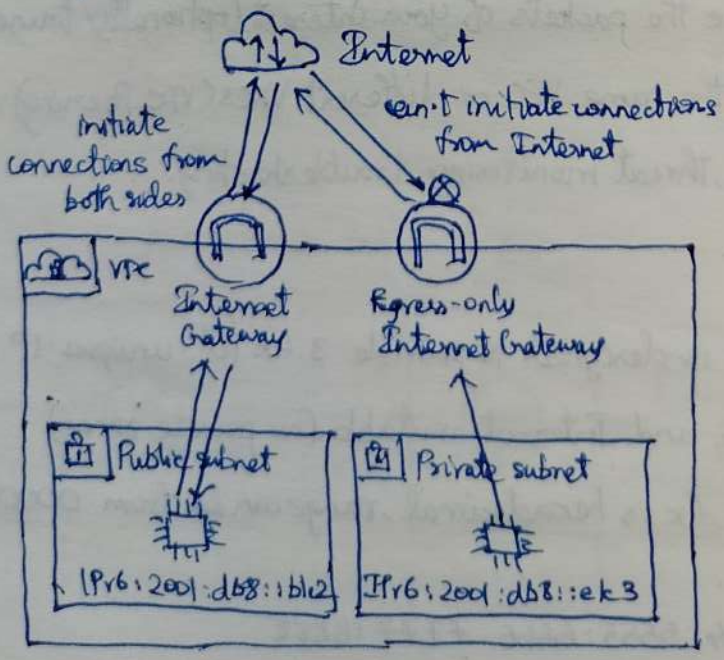
- IPv4 cannot be disabled for your VPC and subnets
- You can enable IPv6 (they're public IP addresses) to operate in dual-stack mode.
- Your EC2 instances will get atleast a private internal IPv4 and a public IPv6
- They can communicate using either IPv4 or IPv6 to the internet through an Internet Gateway

IPv6 Troubleshooting

- If you cannot launch an EC2 instance in your subnet
  - It's not because it cannot acquire an IPv6 (the space is very large)
  - It's because there are no available IPv4 in your subnet
- Solution: create a new IPv4 CIDR in your subnet

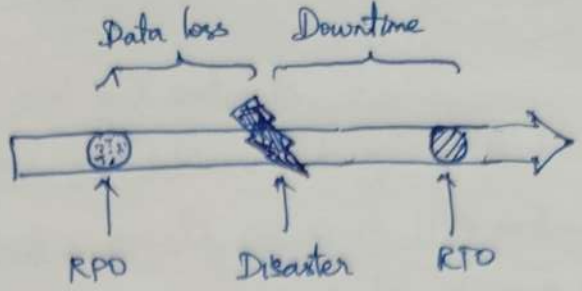
Egress-only Internet Gateway

- Similar to NAT Gateway but for IPv6
- Allow instances in your VPC outbound connections over IPv6 while preventing the internet to initiate an IPv6 connection to your instances.
- You must update the Route Tables.



Disaster Recovery Strategies

a) Backup and Restore
b) Pilot light
c) Warm standby
d) Hot site / Multi-site



RPO: Recovery Point Objective

RTO: Recovery Time Objective

b) Pilot Light

- A small version of the app is always running in the cloud
- Use for critical core.
- Faster than Backup and Restore as critical systems are already up

c) Warm Standby

- Full system is up and running but at minimum size
- Upon disaster, we can scale to production load.

d) Multi Site / Hot Site

- Very low RTO (minutes or seconds) - very expensive
- Full production scale is running AWS and on premise

DMS - Database Migration Service

- Quickly and securely migrate databases to AWS. resilient, self-healing
- The source database remains available during the migration
- Supports:
  - Homogeneous migrations. ex: Oracle to Oracle
  - Heterogeneous migrations. ex: Microsoft SQL Server to Aurora
- Continuous Data replication using CDC
- You must create an EC2 instance to perform the replication tasks.

AWS Schema Conversion Tool (SCT)

- Convert your Database's Schema from one engine to another
- Eg: OLTP: (SQL Server or Oracle) to MySQL, PostgreSQL, Aurora
- Eg: OLAP: (Teradata or Oracle) to Amazon Redshift
- You do not need to use SCT if you are migrating the same DB engine

On-Premise strategy with AWS

- Ability to download Amazon Linux 2 AMI as a VM (.iso format)
  - VMware, KVM, VirtualBox (Oracle VM), Microsoft Hyper-V
- VM Import/Export
  - Migrate existing applications into EC2
  - Create a DR repository strategy for your on-premise VMs
  - Can export back the VMs from EC2 to on-premise
- AWS Application Discovery Service
  - Gather information about your on-premise servers to plan a migration
  - Server utilization and dependency mappings
  - Track with AWS Migration Hub
- AWS Database Migration Service (DMS)
  - Replicate on-premise → AWS, AWS → AWS, AWS → on-premise
- AWS Server Migration Service (SMS)
  - Incremental replication of on-premise live servers to AWS

AWS DataSync

- Move large amount of data from on-premise to AWS
- Can synchronize to: Amazon S3 (any storage classes - including Glacier), EFS, FSx
- Move data from your NAS or file system via NFS or SMB
- Replication tasks can be scheduled hourly, daily, weekly
- Leverage the DataSync agent to connect to your systems
- Can setup a bandwidth limit