Udemy SAA-C03 Notes (Stephane Maarek Version)

- **AWS IAM**
  - Audit
    - IAM Credentials Report (account-level)
    - IAM Access Advisor (user-level)
  - Permission boundary
    - control the maximum permissions employees can grant to the IAM principals that they create and manage
- **Advanced Identity in AWS**
  - **Org**
    - SCP - service control policies
    - do not apply management account
  - **IAM role**
    - assume a role, give up original permission and take the permission assigned to the role
  - **Resource-based policies**
    - principal does not have to give up permission
  - **AWS IAM identity center**
    - one login for all AWS accounts in AWS Org
  - **AD - Microsoft active directory**
    - helps administrators manage network resources and permissions
    - objects are organized in trees, a group of trees is a forest
    - Directory services
      - AWS managed Microsoft AD
        - establish 2 way “truth” connections
      - AD Connector
        - directory gateway(proxy) to redirect(support MFA)
      - Simple AD
        - cannot be joined with on-premises AD
  - **AWS control tower**
    - setup and govern a **secure** and **compliant** multi-account AWS environment based on best practices
    - Guardrails
      - Preventive guardrail - using SCP
      - Detective guardrail - using AWS Config

- **AWS EC2**
  - QUOTAS - request a limit increase
  - Default termination policy:
    - starting point: helps ensure that instances are **distributed evenly** across AZ for HA
    - if AZ has equal num of instance, then checks for the **oldest** launch configuration

- **Capabilities**
  - Renting VM (EC2)
  - Storing data on virtual drives (EBS)
  - Distributing load across machines (ELB)
  - Scaling the services using ASG
- **Instance types**
  - General purpose
    - Great for diversity of workload
    - Balance between Compute, memory, and networking
  - Compute Optimized
    - For **high performance** processors
      - Batch, Media transcoding, dedicated gaming servers, ML
  - Memory Optimized
    - Fast performance for **workloads** that process large data sets in memory
      - High performance, SQL/NoSQL DB
  - Storage optimized
    - For high, sequential read and write access to **large data sets** on local storage
      - High frequency online transaction processing systems
      - Data warehousing app
- **Security group**
  - **Default** blocked all inbound traffic and authorize all outbound traffic
  - **Stateful**, only contain allow rules, can reference by IP or by SG
  - Act as a “firewall” on EC2 instances
  - Lock down to a region/VPC combination
- **Instances purchasing options**
  - On-demand - short workload, predictable pricing, pay by second
  - Reserved (1 & 3 years)
    - reserved instances - long workloads
    - convertible reserved instances - long workloads with flexible instances
  - Saving plans - (1 & 3 years) –commitment to an amount of usage, long workload
  - Spot instances - short workloads, cheap, can lose instances (less reliable)
  - dedicated hosts - book an entire physical server, control instance placement, support business **license**
  - dedicated instances - no other customers will share your hardware
  - capacity reservations - reserve capacity in a specific AZ for any duration, **No time commitment, no billing discounts**
- Launch configuration = EC2
- Launch template = AMI
  - defining a **launch template** instead of a **launch configuration** allows you to have **multiple versions of a template.** (i.e: use different instance type)

- **AWS EC2 - Associate**
  - **Public Ip**
    - Unique across whole web
  - **Private IP**
    - Unique across the private network
  - **Elastic IP**
    - Remap the address to another instance
  - **Placement groups**
    - Cluster
      - Great network, but all instances fails if the AZ fails.
    - Spread
      - Span across AZ and EC2 instances are on different physical hardware
      - **Limited to 7 instances/AZ**
    - Partition
      - Up to **7 partitions/AZ**
      - Up to 100s of EC2 instances
  - **Elastic Network Interfaces - ENI**
    - Logical component in a VPC represents a virtual network card
  - **EC2 hibernate**
    - RAM state is preserved, boot is faster
    - RAM state is written into **root EBS volume**, which **must be encrypted**
    - not able to hibernate more than **60 days**

- **AWS EC2 - instance storage**
  - **EBS**
    - a network drive can be attached to instances
    - Only be mounted to one instance at at time (CCP level)
    - Lock to an AZ
    - By default, the root volume for an AMI backed by Amazon EBS is **deleted when the instance terminates.**
  - **EBS snapshot**
    - Can copy snapshot across AZ or region
    - Features:
      - EBS snapshot archive, 75% cheaper, 24-72 hrs for restoring the archive
      - Recycle Bin for EBS snapshots
      - Fast snapshot restore.
    - **Fast snapshot restore** (FSR) enables you to create a volume from a snapshot that is fully initialized at creation.
  - **EBS Volume types:**
    - **gp2/gp3(SSD)**: General purpose SSD volume that balances price and performance for a wide variety of

workloads, **can be used as boot volumes. 3000IOPS**

- **io1/io2 Block Express(SSD)**: Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads, **can be used as boot volumes**
  - Great for databases workloads
  - up to **16,000 IOPS**
  - support EBS **multi-attach,** up to **16 instances**
- st1 (HDD): Low cost HDD volume designed for frequently accessed, throughput- intensive workloads
  - max IOPS 500
- sc1(HDD**)**: Lowest cost HDD volume designed for less frequently accessed workloads

- EBS encryption
- **EFS - Elastic File System**
  - Managed NFS(network file system) that can be mounted on many EC2
  - EFS works with instances in multi-AZ
  - **HIghly available, scalable**
  - Compatible with **Linux based AMI (not windows)**
  - SG to control access to EFS
- **EFS mode**
  - Performance mode - general purpose, higher latency and parallel
  - Throughput mode - can instantly provision the throughput of your file system (in MiB/s), independent of the amount of data stored.
  - Bursting Throughput mode - throughput on Amazon EFS **scales as the size of your file system grows.**
- **EFS storage classes**
  - standard: frequently accessed files
  - EFS-IA:
  - Archive: rarely accessed data, 50% cheaper
- **EC2 instance store**
  - High performance hardware disk
- **AMI**
  - customization of EC2 instance, faster boots/configuration time bc pre-packaged
  - Built for specific region
  - Process: Launch EC2 instance, **stop** it, build an AMI, launch instance from AMI

- **High Availability & Scalability**
  - **benefits of LB**
    - **expose single point of DNS to app**
    - Health check
      - done on a port and a route
      - 200 = healthy, !200 = unhealthy
    - Provide SSL termination (HTTPS) for web
    - Separate public traffic from private traffic

- Enforce **stickiness** with cookies

- **Types of LB**
  - Classic load balancer - HTTP, HTTPS(layer 7), TCP(layer 4), SSL
  - **Application load balancer(layer 7)** - **HTTP, HTTPS, WebSocket**
    - By default, ALB waits **300 seconds** before the completion of the deregistration process
    - can provide URL to endpoint
    - great fit for micro services & container-based app
    - Has port mapping to redirect to a dynamic port in ECS
    - **support redirects(http → https)**
    - to multi-HTTP app across machines (target groups)
    - Target groups
      - EC2, ECS
      - Lambda functions - HTTP request is translated into a JSON event
      - Must be private IP address
    - to multi app on the same machine (containers)
  - **Network load balancer** - TCP(layer 4), **TLS, UDP, TCP**
    - can't provide URL to endpoint, **instance/IP only**
    - Forward TCP & UDP traffic to instances
    - **One static IP**/AZ, support elastic IP
    - ! free tier
    - For **extreme performance**
    - Target group
      - EC2, ALB
      - must be private IP
      - Health check support TCP, HTTP, HTTPS
  - **Gateway load balancer** - Operates - layer 3(network layer) - IP protocol
    - Deploy, scale, and manage a fleet **3rd party** network virtual app in AWS
    - transparent network gateway - single entry/exit for all traffic
    - LB - distributes traffic to virtual apps
    - Uses the GENEVE protocol on port 6081
    - Target groups
      - EC2
      - must be private IP

- **Sticky sessions**
  - implement stickiness so same client always redirect to same instance behind a LB
  - Can control expiration date for CLB & ALB
  - **apply to CLB, ALB, NLB**
  - Cookies name
    - Application-based cookies
      - do not use **AWSALB**, **AWSALBAPP**, **AWSALBTG**

        - Duration-based cookies
            - AWSALB for ALB
            - AWSELB for CLB

- **RDS, Aurora & ElasticCache**
    - **RDS**
        - auto provisioning, OS patching
        - **time restore**
        - can't ssh in bc RDS is a managed service
        - Storage auto scaling by setting Max storage threshold for unpredictable workloads
        - **up to 15 read replicas**
        - need to update connection string to leverage read replicas
        - **read replicas**
            - **is free in same region**
            - **ASYNC replication**
    - **RDS multi AZ - DR**
        - **SYNC replication**
        - **One DNS name - auto app failover to standby**
        - not used for scaling
        - **Zero downtime operation**
    - **RDS custom**
        - Managed **Oracle** and **Microsoft SQL** DB with OS and database customization
        - full admin access to the underlying OS and the DB
    - **RDS Proxy**
        - Improving DB efficiency by reducing the stress on DB resources and min open connections
        - Reduced RDS & Aurora **failover** time by up 66%
        - **Enforce IAM authentication** for DB, and securely **store credentials in Secrets Manager**
        - **Must access from VPC** (never publicly accessible)
    - **Amazon Aurora**
        - **Aurora Replicas have their own endpoints**
        - Aurora Serverless **automatically starts up, shuts down**, and **scales capacity** based on an application's needs
        - pay for only the database resources that you consume on a per-second basis.
        - 5x performance improvement over RDS MySQL, 3x RDS Postgres
        - up to **15 read replicas** If the writer instance in a cluster becomes unavailable, Aurora automatically promotes one of the reader instances to take its place as the **new writer.**
        - **failover is instantaneous, less than 30 sec for master**
        - Support cross region replication
        - Reader endpoint connect all the read replicas

        - Custom Endpoints
            - **Run analytical queries on specific replicas**
            - The reader endpoint is not used after defining Custom Endpoints
        - Global Aurora
            - **Cross region read replicas -** DR
            - Aurora global database
                - **1 second RPO, RTO of less than 1 minute.**
                - one primary region (R/W)
                - up to **5 secondary regions**
                - up to **16 read replicas/region**
                - cross-region replication less than 1 sec
        - **Aurora Cloning**
            - Faster than snapshot & restore
            - Uses **copy-on-write protocol**
            - Useful to create a '**staging**' DB from a "production" DB **without impacting** the production DB
        - Aurora ML
            - support Amazon SageMaker, Comprehend
        - **backup comparison**
            - RDS backup
                - **daily full backup** of the DB
                - Transaction logs are backed-up **every 5 mins**
            - Aurora backup
                - Automated backup **1 to 35** days(cannot be disabled)
                - **point-in-time recovery**
    - **RDS & Aurora Restore options**
        - restoring MySQL RDS from S3
            - create backup on-premises DB
            - store backup file on S3
            - Restore the backup file onto a new RDS instance running MySQL
        - Restoring MySQL Aurora cluster from S3
            - create backup on-premises DB using **Percona Xtrabackup**
            - store backup file on S3
            - Restore the backup file onto a new Aurora cluster running MySQL
    - **RDS & Aurora Security**
        - At-rest encryption using **KMS**
        - in-flight encryption: **TLS**-ready by default
        - IAM authentication
        - SG
        - No SSH
        - Audit logs can be enabled
    - **Amazon ElastiCache**

        - Involve heavy code changes
        - in-memory DB with really **high performance**, low latency
    - *** ElastiCache - Redis vs Memcached**
        - ElastiCache - Redis
            - ideal front-end for data stores, providing a high-performance middle tier for applications with **extremely high request rates & low latency requirements.**
            - **Multi AZ with auto failover**
            - Pattern
                - **Lazy loading**: all read data is cached
                - **Write Through**: adds or update data in cache when written to a DB
                - **Session store** (stateless): store temporary session data in cache using TTL
            - security
                - support **IAM authentication** for AWS API-level security
                - support **SSL** in flight encryption
                - set password/token
                - read replicas to scale read and HA
            - backup and restore feature
            - **support Sets and Sorted Set** (real time rank)
        - Memcached
            - security
                - support **SASL**-based authentication
            - No HA
            - No persistent
            - No backup and resotre
            - Multi-threaded architeture

- **Amazon route 53 - DNS**
    - Domain Registrar != DNS service
    - **DNS terminology**
        - DNS records: A, AAAA, CNAME, NS...
        - TLD - Top level domain: com, us, in, gov, org
        - SLD - Second level domain: amazon.com, google.com
    - **Records**
        - Domain/subdomain name
        - Record type: A / AAAA / CNAME / NS(name servers for the hosted zone)
            - CNAME: only for not root romain (aka.sth.mydomain.com)
            - Alias: Works for both root/non-root domain
            - native health check
        - Value: 1.2.34.567

        - Routing policy
            - Simple: **no health check**, select random one if return multi values
            - Weighted: A weight of 0 means stop, all weight of 0 means equal
            - Latency: traffic between users and AWS regions
            - Failover: **Health Check mandatory**
            - Geolocation: based on user location, should create **default** record
            - Geoproximity: ability to shift more traffic to resources based on bias, must use Traffic Flow
            - IP based:
            - Multi-value: not a substitute for having a ELB
        - TTL (except for Alia records, TTL is mandatory for each DNS record)
    - Calculated health checks
        - OR, AND, NOT
        - health check are outside the VPC, can't access private endpoints, we can create CloudWatch and create Health Check checks the alarm itself

---

- **Amazon S3**
    - **Globally unique name across all region all accounts**
        - not start with prefix "xn—"
        - not end with suffix "-s3alias"
    - **The key is composed of prefix + object name**
        - s3://my-bucket/my_folder1/another_folder/my_file.txt
    - **Max object size = 5 TB, use multi-part upload if more than 5 GB**
    - **Security**
        - users-based
            - IAM policy
        - Resource-based
            - Bucket Policy
            - Object ACL 0 finer grain
            - Bucket ACL - less common
        - Storage class - S3 $\xrightarrow{\text{need at least 30 then}}$ other S3 class
            - **S3 standard**: 11-9's%, frequently accessed data, store data in a min of three AZ
            - **S3 standard-IA**:
            - **S3 Intelligent tiering**: Ideal for data with unknown or changing access patterns
            - **S3 One Zone-IA**: in single availability zone
            - **S3 Glacier instant retrieval**: data requires immediate access, retrieve objects within a few milliseconds
            - **S3 Glacier Flexible Retrieval**: low-cost, within few mins to hours
            - **S3 Glacier deep archive**: lowest-cost, retrieve within 12 hours
        - S3 outposts: Creates S3 buckets on Amazon S3 Outposts, Makes it easier to retrieve, store, and access data on AWS Outposts

- **Amazon S3 - advanced**
  - Bucket policy - **for account & user level control**
  - **Lifecycle rules**
    - can be created by certain **prefix**(example: *s3://mybucket/mp3/*)* OR certain **object tags**(example:*Department:Finance*)
    - Transition actions
    - Expiration actions
      - can be used to delete old versions of files
      - can be used to delete incomplete Multi-Part uploads
  - **Storage class analysis**
    - recommendations for **Standard** and **Standard IA**
    - not work for One-Zone IA or Glacier
    - report is updated daily, which takes **24-48** hrs to start seeing data analysis
  - **Requester pays**
    - requester must be **authenticated** in AWS
  - **S3 event notifications**
    - → SNS / SQS / Lambda
    - → EventBridge - multiple destinations
  - **Baseline performance**
    - auto scale, latency 100-200ms
    - at least - [below]**/sec/prefix** in a bucket
      - **3500** PUT/COPY/POST/DELETE
      - **5500** GET/HEAD requests
      - i.*e: If you spread reads across all four prefixes evenly, you can achieve 22,000 requests per second for GET and HEAD*
  - **Performance**
    - Multi-part upload
      - **increase resiliency and avoid upload restarts.**
      - files > 100MB, must use for files > 5GB
      - **parallelize uploads**
    - S3 transfer acceleration
      - transfer file to **edge location** → s3 bucket in the target region
      - **compatible with Multi-part upload**
    - S3 Byte-range fetches - speed up download
      - parallelize GETs by requesting specific byte range
    - S3 select & glacier select
      - server-side filtering by rows & cols
        - retrieve less data using SQL
      - Less network transfer and CPU cost client-side

- S3 batch operations
  - can use S3 Inventory to get object list and use S3 Select to filter your objects
- Storage lens
  - analyze, optimize storage across entire **AWS Org**
  - discover **anomalies**, identify cost efficiencies, apply data protection
  - aggregate data for Org, accounts, regions, buckets, prefixes
  - dashboard/customize dashboard
  - can be configured to export metrics daily to S3 bucket
    - Metrics
      - summary metrics
      - cost-optimization metrics
      - data-protection metrics
      - access-management metrics
      - event metrics
      - performance metrics
      - activity metrics
      - detailed status code metrics
  - Free
    - 28 usage metrics
    - data is available for queries for **14 days**
  - Paid
    - advance metrics
    - cloudwatch publishing
    - prefix aggregation
    - data is available for queries for **15 months**

- **Amazon S3 - security**
  - Object encryption
    - SSE - Server-side encryption
      - SSE-S3 - enabled by default for new buckets & objects
        - Encryption type is AES-256
        - header → "x-amz-server-side-encryption": "AES256"
      - SSE-KMS, KMS to manage encryption keys
        - header → "x-amz-server-side-encryption": "aws:kms"
        - limitation
          - upload → generateDateKey KMS API
          - download → decrypt KMS API
      - SSE-C, customer provided keys
        - customer send key to S3 and then encrypt data in server side

                - **must use HTTPS**
        - Client side encryption
    - Encryption in transit (SSL/TLS)
        - HTTPS is recommended, but mandatory for SSE-C
    - Default encryption vs. Bucket policy
        - bucket policy evaluate first then default encryption
    - **CORS** - cross-origin resource sharing
        - origin = protocol + host + port
    - **MFA delete**
        - reuqire MFA when
            - permanently delete an object version
            - suspend versioning on the bucket
        - must enable versioning
        - Only bucket owner(root) can enable/disable MFA delete
    - S3 Access Logs
        - must be in the same region, target bucket and resource bucket
        - Do not set logging bucket to be the monitored bucket, which will create logging loop
    - **Pre-Signed URLs**
        - URL expiration
            - console: 1 min to 12 hr
            - CLI: 1 sec to 168 hrs
        - Users given a pre-signed URL inherit the permission of the user that generated the URL for GET/PUT
    - Lock
        - Common
            - adopt WORM model - write once read many
        - Glacier vault lock
            - **For compliance and data retention**
            - Create a Vault Lock Policy
        - S3 object lock
            - versioning must be abled
            - retention mode - compiance
                - cant overwritten or deleted by any user(root as well)
                - object retention modes/period cant be changed
            - retention mode - governance
                - most user cant overwrite or delete an object version
            - retention period
                - protect object for fixed period
            - Legal hold
                - protect object **indefinitely**, independent from retention period
                - can be freely placed and removed using S3:PutObjectLegalHold
    - **Access points**

        - **has own DNS name, access point policy**
        - Access point VPC Origin
    - Object Lambda
        - change object before it's retrieved by the caller app
        - only one S3 bucket is needed by creating S3 access point & S3 object lambda access point
        - use case
            - redacting PII
            - convert format
            - resizeing

- **CloudFront & Global Accelerator**
    - **<u>CloudFront</u>**
        - General
            - Improve read performance
            - content is cached at edge
            - DDoS protection
        - Origins
            - S3 bucket
                - OAC - Origin Access Control enhanced security with CloudFront
                - CloudFront can be used as an ingress to upload file to S3
            - Custom Origin (HTTP)
                - ALB
                    - CloudFront → **public** ALB → **private** instance
                - EC2 instance
                    - CloudFront → **public** instance
                - S3 static website
        - Geo Restriction
            - Allowlist and Blocklist
        - price classes
            - Price Class All: all region
            - Price Class 200: most regions, but excludes the most expensive regions
            - Price Class 100: only the least expensive regions
        - Cache Invalidations
            - **cache refresh**
        - Unicast Ip
            - one server holds one IP address
        - Anycast IP
            - All servers hold same IP address and the client is routed to the nearest one
    - **<u>Global Accelerator -</u>** <u>directs traffic to optimal endpoints over the AWS global network</u>

        - General
            - **2 Anycast IP** are created for app
                - Antcast IP → Edge location →(internal network) app
            - improve over **TCP or UDP**
            - Works with Elastic IP, EC2 instance, ALB, NLB, public or private
            - Intelligent routing to lowest latency and fast regional failover
            - Internal AWS network
            - health checks
                - Great for DR
    - **Comparison**
        - Global Accelerator
            - Improves performance for wide range of app over **TCP or UDP**
            - proxying packets at edge to app running in multi-regions
            - Good for UDP, IoT, or Voice over IP
            - Good for HTTP with static IP address and fast regional failover
        - CloudFront
            - Global Edge network, content is served at the edge
            - Files are cached for TTL
            - **static & dynamic** content
        - S3 Cross Region Replication
            - Must setup for each region
            - Files
            - Read only
            - Great for **dynamic** content at low-latency in few regions

---

- **AWS storage extras**
    - **More than a week to transfer over network, use Snowball devices!**
    - **Snowcone**
        - small, portable computing
        - Snowcone - 8 TB pf HDD storage
        - Snowcone SSD - 14 TB of SSD storage
        - can connect it to internet and use AWS DataSync to send data
    - **Snowball Edge**
        - Edge storage optimized
            - 80 TB of HDD or 210TB NVMe capacity for block volume
        - Snowball edge compute optimized
            - 42 TB of HDD or 28 TB NVMe capacity for block volume
    - **Snowmobile**
        - > 10 PB
    - **Edge computing** - Snowball, Snowcone

- **AWS OpsHub**
  - manage snow family device
- **Snowball into Glacier**
  - Snowball cannot import to Glacier directly
  - snowball →S3 → Glacier
- **Amazon FSx**
  - Launch **3rd party** high-performance file systems on AWS
  - Fully managed service
  - **FSx for Lustre**
    - **parallel distributed** file system for large-scale computing
    - ML, **high performance computing**
    - **can be used from on-premises servers (VPN or DX)**
    - **seamless integration with S3**
    - options
      - Scratch File System
        - short term
        - data not replicated
      - Persistent File Sytem
        - long term
        - data is replicated within same AZ
        - replace failed files within mins
  - **FSx for NetAPP ONTAP**
    - **High OS Compatibility** - File system compatible with NFS, SMB, iSCSI protocol
    - **Point-in-time instantaneous cloning (helpful for testing new workloads)**
    - storage auto scale
  - **FSx for Windows File Server**
    - fully managed **Windows** file system share drive
    - Can be mounted on Linux EC2 instance
    - support Microsoft's Distributed File System(DFS) Namespaces
    - Multu-AZ
    - Can be accessed from on-premises
    - Data is backed-up daily to S3
  - **FSx for OpenZFS**
    - **NFS**
    - Point-in-time instantaneous cloning (helpful for testing new workloads)
    - Up to **1,000,000 IOPS** with < 0.5ms
- **AWS storage gateway**
  - bridge between on-premises data and cloud data
  - types of storage gateway
    - **S3 file gateway**
      - extend on premise's storage space

        - Most recently used data is **cached** in the file gateway
        - Transition to S3 Glacier using a Lifecycle Policy
      - FSx file gateway
        - Local **cache** for frequently accessed data
      - Volume gateway
        - Cached volumes: low latency access to most recent data
        - Stored volumes: entire dataset is on premise, scheduled backups to S3
      - Tape gateway
      - storage gateway - hardware appliance
  - **Transfer family** - FTP, FTPS, SFTP interface on top of Amazon S3 or Amazon EFS
  - **AWS DataSync - automates** and **accelerates** moving data between **on premises and AWS Storage services**.
    - **not support EBS**
    - File permissions and metadata are preserved
    - **Move large amount data to and from**
      - on-premises - need agent
      - AWS to AWS - no agent needed

- **AWS Integration & Messaging**
  - **SQS**
    - Dead-letter queue - uses redrive policy on the main SQS queue to send messages to a dead-letter queue after failing a certain times
    -
    - Two patterns
      - SYN: client → server
      - ASYNC: client → SQS → Server
    - SQS attributes
      - unlimited throughput
      - **default retention 4days**, **max 14 days**
      - low latency
      - limitation of **256KB**/message
      - message is persisted in SQS until a consumer deletes it(DeleteMessage API)
      - Poll message - receive up to 10 messages at a time
    - SQS & SNS security
      - Encryption
        - in-flight encryption using HTTPS API
        - at-rest using KMS keys
      - Access controls - IAM policies
      - SQS Access policies
        - useful for cross-account access to SQS

            - for other service to write to SQS
        - SQS Message Visibility Timeout
            - **by default is 30sec**
            - After polling by a consumer, it becomes invisible to other consumers.
        - SQS long polling
            - **optimize num of API call, decrease latency**
        - SQS FIFO - first in first out
            - FIFO queues support up to **3,000 messages/sec** with batching, or up to 300 messages/sec without batching
        - SQS as a buffer to databased writes so make sure not losing data
        -
    - **SNS**
        - Pub/Sub pattern
        - Fan-out: one message → topic → multi-target
        -
        - up to 12,500,000 subscriptions
        - 100,000 topic limit
        - SNS - FIFO
            - queue ends with the .fifo suffix
            - ordering by **group ID**
            - limited throughput (same as SQS)
        - SNS Message Filtering
    - **Kinesis**
        -
        - Collect, process, ad analyze streaming data in real time
        - Kinesis data stream
            - data up to **1MB** with **Partition Key**
            - retention 1 ~ 365 days
            - Once inserted in Kinesis, can't be deleted
            - **Fanout feature: auto scale with the num of shard in a stream** (for performance lag)
            - Provisioned mode
                - choose num of shards, scale manually
                - 1 MB/s per in shard
                - 2 MB/s per out shard
                - pay /shard/hr
            - On demand mode
                - default 4 MB/s in
                - auto scale
                - pay /stream/hr & date in/out per GB
        - Kinesis data firhose
            - supports custom data transformations using AWS Lambda.
            - load stream data into S3/../3rd party
            - Fully managed service, serverless, auto scale

        - **Near real time**
        - pay for data going through Firehose
    - **Amazon MQ**
        - managed message **broker service**
        - run on server, Multi-AZ with failover
        - has both features of SQS and SNS

- **Containers on AWS**
    - **Docker**
        - platform to deploy apps, apps are packaged in containers that can be run on any OS
        - Docker images stored in
            - Docker hub (public)
            - Amazon ECR - elastic container registry (private & public)
        - vs VM
    - **ECS** - elastic container service
        - data volume
            - EFS
        - EC2 launch type
            - **charged based on EC2 instances and EBS volumes used**
            - user provision & maintain the infrastruture
        - Fargate launch type
            - **charged based on vCPU and memory resources** that the containerized application requests
            - **minimal costs when the application is idle**
            - serverless
        - IAM roles
            - obtain temporary security credentials to access cross-account resources
            - EC2 instance profile (EC2 launch type only)
            - ECS task role
                - **define in the task definition**
    - **EKS** - elastic Kubernetes service
        - open source system for auto deployment, scaling, management of containerized app
        - Kubernetes is cloud-agnostic
        - data volumes
            - EBS, EFS, FSx for Lustre, FSx for NetApp ONTAP
    - **AWS APP Runner**
        - fully managed service to deploy web app and APIs at scale

---

- **Serveless Overview**

- Lambda
  - scale extremely quickly(better have monitor, like CloudWatch)
  - by default, lambda function has the full ability to make network requests to any public internet address
  - VPC-enabled, all network traffic from your function is subject to the routing rules of your VPC/Subnet
- **Lambda Snapstart**
  - improve performance up to 10x(by pre-initialized) at no extra cost for Java 11 and above
- **Customization at the edge**
  - attach to CloudFront distributions
  - run close to users to min latency
- **CloudFront function**
  - lightweight function in JS
  - millions of request/sec
- **lambda@Edge in NodeJS/Python**
  - 1000s of requests/sec
- **Lambda with RDS proxy**
  - Lambda must be deployed in VPC bc RDS Proxy is never publicly accessible
- **RDS event notification**
  - near real time
- **DynamoDB**
  - more cost-effective than RDS
  - NoSQL
  - max size /item is **400KB**
  - can rapidly evolve schemas
  - **Provisioned mode(default)**
    - specify the num of reads/writes per sec
  - **On-demand mode**
    - read/write auto scale
  - *** DynamoDB accelerator**
    - DAX improves **read performance for repetitive queries**
    - API compatible with DynamoDB
    - help solve read congestion by caching, DAX is used to natively **cache** Amazon DynamoDB reads.
    - **Microseconds latency** for cached data
    - 5 mins TTL for cache by default
  - Stream processing
    - DynamoDB stream
    - Kinesis data stream
  - Global table
    - active-active replication, can read and write to table in any region
    - low latency
  - Backup for DR
    - Continuous backup using point-in-time recovery

        - On-demand backup
            - for long-term retention

- **Databases in AWS**
    - Database types
        - RDBM - RDS, Aurora - great for joins
        - NoSQL
            - DynamoDB(JSON): **max 400KB upload file**
            - ElastiCache(key/value pairs), DocumentDB(MongoDB), Neptune, Keyspace - no joins
        - Object store - S3
        - Data warehouse - Redshift, Athena, EMR
        - Search - OpenSearch(JSON)
        - Graphs - Amazon Neptune
            - example
            - real-time ordered
            - no duplicates, strict order
        - Ledger - Quantum ledger DB
            - a ledger is a book recording financial transactions
            - used to review history of all the changes made to your application data over time
            - Immutable system
        - Time series - Amazon Timestream

- **Data & Analytics**
    - **<u>Athena</u>**
        - **analyze data in S3 using serverless SQL**
        - If file size < 128MB in S3, the runtime engine might spend additional time to open S3 files, list, get...
        - perform queries periodically, pay only for the queries that users run
        - Performance improvement
            - Use columnar data for cost-savings(less scan)
            - Compress data
            - Partition datasets in S3 for easy querying on virtual columns
            - use larger files(>128MB) to minimize overhead
    - **<u>Redshift</u>**
        - For large inserts (few hundred GB to PT)
        - It's OLAP - online analytical processing
        - Columnar storage of data
        - **vs Athena**: faster queries / joins / aggregations thanks to indexes
        - Snapshot & DR

        - has Multi-AZ mode for some clusters
        - can configure amazon redshift to auto copy snapshots of a cluster to another region
    - **Redshift spectrum**
        - **efficiently query and retrieves structured and semistructured data** from S3 without loading data into Redshift tables
        - use **less cluster's processing capacity** than other queries.
        - must have a redshift cluster available to start the query
- **Glue**
    - ETL (convert data to Parquet format) → S3
    - Job Bookmarks - prevent re-processing old data
- **Amazon EMR -** Elastic MapReduce
    - for data processing, ML, web indexing, big data
    - have long-running cluster, or temporary cluster
- **QuickSight**
    - can share the analysis or the dashboard with Users and Groups
- **Lake formation**
    - central place to have all data for analytics purpose
    -
- **Kinesis data analytics**
    - **real-time** analytics on Kinesis data stream & firehose using SQL
    - Kinesis data stream vs. MSK - Managed streaming for Apache Kafka
        -

- **ML**
    - **Rekognition**
        - Find objects, people, text, scenes in **images** and **videos** using ML
        - Facial analysis/search
    - **Transcribe**
        - speech → text
        - auto remove PII - personally identifiable info using redaction
    - **Polly**
        - text → speech
        - Speech Synthesis Markup Language (SSML) - enable more customization
    - **Lex & Connect**
        - Lex
            - like Alexa
        - Connect
            - receive calls, create contact flows, cloud-based virtual contact center
    - **Comprehend**
        - For NLP - Natural Language Processing
        - use ML to find insights and relationship in text

        - Comprehend Medical
            - detect PHI - Protected Health Information
    - **SageMaker** - to build ML models
    - **Forecast** - use ML to forecast
    - **Kendra** - document search service powered by ML
    - **Personalize** - to build app with real-time personalized recommendations
    - **Textract** - auto extract text, handwriting, and data from any scanned document

---

- **AWS Monitoring**
    - CloudWatch
        - Share a single dashboard and designate as many as **five email addresses** of people(do not have AWS account) who can view the dashboard.
    - **CloudWatch Log - S3 export**
        - CloudWatch alarm actions provide auto stop, terminate, reboot, or recover instance
        - log data can take up to **12 hr** to become available for export
        - API call CreateExportTask
        - not near-real time
    - **Subscription Filter**
        - filter which logs are events delivered to different destination
    - **Amazon EventBridge - schema registry**
        - allow you to generate code for app that will know in advance how data is structured in the event bus
    - **AWS CloudTrail**
        - events are stored for 90 days
        - provides **governance, compliance and audit for AWS account**
        - Get history of event/API calls made within your AWS account
        - apply to all regions or single region
    - **CloudTrail Insight**
        - detect unusual activity
    - **AWS Config**
        - AWS Config conformance packs
            - collections of AWS Config rules and **remediation** actions
        - help record configuration and changes over time
        - Config rules no deny
        - **check for expiring certificates**

- ***AWS security**
    - **KMS**
        - KMS keys types
            - Symmetric (ASE-256 keys) - single key

        - Asymmetric(RSA & ECC pair keys)
            - public + private key
    - Automatic key rotation **every 1 year**
        - must be enabled for Customer-managed KMS Key
    - Key policies
        - Default KMS Key policy → for entire account
        - Custom KMS Key policy → specific users
    - Copy snapshot across account
        - Snapshot with KMS key → **Attach KMS key policy to authorize cross-account access** → share encrypted snapshot → create copy snapshot with another KMS key → release snapshot
    - KMS multi-region
        - For global client-side encryption, Global DynamoDB, Global Aurora
        - Each multi-region key is managed **independently**
        - for protecting specific field even from database admin (SSN col)
- **S3 replication encryption**
    - encrypted & unencrypted objects with SSE-S3 are replicated by default
- **AMI sharing process encrypted via KMS**
    - AMI + KMS from source account → add **launch permission** to target account → share encrypted snapshot → assign permission in target to DescribeKey, ReEncrypted, CreateGrant, Decrypt → launch EC2 from AMI with new KMS key
- **SSM parameter store - in System manager**
    - require **customer** to **rotate** the keys
    - standar
        - 10,000 parameters, max 4KB of parameter value
    - Advanced
        - 100,000, 8KB
        - Allow to assign TTL to a parameter to force
        - allow multi-policies at a time
- **AWS secrets manager**
    - Can force rotation of secrets every X days
    - replicate secrets multi-regions, can keep read replicas in sync with primary secret
- **ACM - AWS Certificate Manager**
    - provision, manage, deploy TLS certificate (HTTPS)
    - can not use ACM with EC2
    - import 3rd party/public certificates
        - No automatic renwal
        - ACM send daily expiration event → EventBridge
        - **AWS Config** (->EventBridge) **can check for expiring certificates**
    - ACM with ALB
        - can redirect HTTP to HTTPS
    - ACM with API gateway
        -

        - Edge-optimized
            - Request are from CLoudFront edge location, thats why the TLS certificate must be in the same region as CloudFront
        - Regional
            - TLS certificate must be in the same region as API gateway
        - Private
    - **AWS WAF**
        - Layer 7(HTTP)
        - for ALB, API gateway, CloudFront, AppSync GraphQL API, Cognito User Pool
        - IP set up to 10,000, can use multi-rules for more IPs
        - security
            - **SQL injection, Cross-Site scripting(XSS)**
            - size contraints, geo-match
            - rate-based rules for DDoS
        - WebACL is regional
    - **AWS shield - for DDoS**
    - **AWS firewall manager**
        - manage rules in all accounts of an AWS Organization
        - rules apply to **new resources** across all and future account in Org
    - **Best practices for DDoS**
        - edges location have DDoS protection (CloudFront, Route53, Global Accelerator)
        - ELB, ASG
    - **Amazon Inspector**
        - automated security assessments
        - **only for EC2, container images & lambda function**
    - **AWS Macie**
        - fully managed data security and data privacy service that uses ML and pattern to protect **sensitive** data(**PII**)

- **Amazon VPC**
    - **CIDR**
        - base ip - xx.xx.xx.xx
        - subnet mask - /0, /24...
    - **Private IP only allow**
        - 10.0.0.0 – 10.255.255.255 (10.0.0.0/8)çin big networks
        - 172.16.0.0 – 172.31.255.255 (172.16.0.0/12) AWS defaultVPC in that range
        - 192.168.0.0 – 192.168.255.255 (192.168.0.0/16) ç e.g., home networks
    - **VPC**
        - max 5 CIDR/VPC
            - min CIDR /28
            - max CIDR /16

- **Subnet**
  - 5 reserves IP address (F4L1)
    - 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 aVPC, therefore the address is reserved
    - example: if need 29 IP address, /27 = 32 IPs, 32 - 5 = 27 < 29, thus subnet mask should /26
  - IGW - allow resources in a VPC connect to internet
    - scale horizontally and highly available and redundant
    - must edit route table
    - one IGW/VPC
  - NAT instance - network address translation
    - must **launch in a public subnet**, can be used as a bastion server
    - must disable EC2 setting: Source/destination Check
    - Must have Elastic IP
    - Route table must to route traffic from private subnet to NAT instance
  - NAT gateway
    - is assigned to **public subnet**
    - pay /hr and bandwidth
    - 5 Gbps and auto scale up to 100 Gbps
    - no SG to manage
    - must create multi NAT Gateway in multi AZs for fault-tolerance
- **NACL - network access control list**
  - one NACL / subnet
  - newly created NACLs deny everything
  - great way of blocking a specific IP address at the subnet level
- **VPC sharing -** share one or more subnet
  - allows multiple AWS accounts to create their application resources into shared and **centrally-managed** Amazon VPC
- **VPC peering**
  - **private connection**
  - can connect VPC in different AWS accounts/regions
  - Not transitive
  - must update route table in each VPC's subnet
- **VPC endpoints**
  - AWS PrivateLink - not require IGW
    - **not require a public** IP address on the instances or public access from the instance subnet.
    - traffic within the region of VPC
  - AWS service is publicly exposed

        - **private network**
        - redundant and scale horizontally
        - remove the need of IGW, NATGW
        - types
            - **Interface endpoints**(power by privateLink) - paid
                - is an **ENI** with a private IP address from the IP address range of your subnet that serves as an entry point for traffic destined to a supported service.
            - **Gateway endpoints - free**
                - S3 & DynamoDB
                - is a gateway that you specify as a target for a route in your route table for traffic destined to a supported AWS service.
                - provisions a gateway and must be used as a target in a route table
                - not using SG
    - **VPC flow logs**
        - monitor & troubleshoot connectivity issues
        - → s3/ cloudwatch logs/ kinesis data firehouse
        - Captures network information from AWS managed interfaces too: ELB, RDS, ElastiCache, Redshift, WorkSpaces, NATGW,Transit Gateway
    - **Site-to-Site VPN** - securely connect your on-premises network to your Amazon VPC
        - **two tunnels**, and each tunnel uses a unique VPC public IP
        - Improve VPN throughput: Use a transit gateway with ECMP routing and **add additional VPN tunnels**
        - VGW - virtual private gateway
            - need to enable route propagation for the VGW in the route table
        - CGW - customer gateway - on-premises
    - **AWS VPN CloudHub**
        - **go over public internet**
        - multi site-to-site VPN connections
    - **DX - private internet**
        - **not encrypted but is private connection** from remote network to your VPC
            - AWS DX + VPN provide IPsec-encrypted private connection
        - Access public resources and private on same connection
        - support both IPv4, IPv6
        - DX gateway
            - for connecting one or more VPC in many regions with same account
        - Dedicated connections - 1, 10, 100 Gpbs
            - physical ethernet port dedicated to a customer
        - Host connection - 50, 500Mbps, to 10Gbps
    - **Transit gateway** to interconnect VPC and on-premises network
        - For having transitive peering between thousands of VPC and on-premises, hub-and spoke connection
        - ECMP - equal cost multi path routing

        - share DX between multi-accounts
        - increasing bandwidth by creating multi Site-to-Site VPN connections
    - **VPC traffic mirroring**
        - Capture from sources(ENIs) to target(ENI)
    - **IPv6**
        - They are public and Internet-routable in AWS
        - IPv4 cannot be disabled for VPC and subnets
        - IPv6 can be disabled
        - EC2 has both, if EC2 can't be launched, maybe IPv4 issue, fix this but creating a new IPv4 CIDR in subnet.
    - **Egress-only internet gateway**
        - similar to NAT gateway but IPv6 only
        - must update route table
    - **Cost**
    - **AWS network firewall**
        - Protect entire VPC from layer 3 - 7
        - centrally managed cross-account to apply VPCs
        - **Traffic filtering - allow, drop, alert for the traffic that matches the rules**
        - active flow inspection to protect against network threats

---

- **Disaster Recovery & Migrations**
    - **DR**
        - RPO - recovery point objective (data loss between disaster)
        - RTP - recovery time objective (downtime between diaster)
        - DR strategies
            - Backup and restore (hours)
                - on-premise - AWS: AWS storage gateway, snowball
                - cloud: snapshot
            - Pilot light (10+ mins) - for **critical core**
                - a small version of app/instance is always running in the cloud
            - warm standby (mins)
                - full system(ELB, ASG, EC2) is up and running, but **min size**
            - hot site / multi site approach (real-time)
                - **full production** scale is running AWS and on-premise
    - **DMS** - Database migration service
        - **continuously replicate with high availability and consolidate databases**
            - i.e: During a database migration to Amazon Redshift, AWS DMS first moves data to an Amazon S3 bucket. When the files reside in an Amazon S3 bucket, AWS DMS then transfers them to the proper tables in the Amazon Redshift data warehouse.
        - must create EC2 instance to perform the replication tasks

        - use **SCT**(schema conversion tool) If schema of target and source is different
        - Multi-AZ deployment, DMS provisions and maintains a syn stand replica in a different AZ
            - provides data redundancy
            - eliminates I/O freezes
            - minimizes latency spikes
    - **AWS backup**
        - centrally manage and automate backups across AWS services
        - support cross-region/account backups
        - Backup Vault lock
            - enforce WORM(write once read many)
            - even root user cannot delete
    - **AWS application discovery service**
        - Agentless discovery connector
        - Agent-based discovery
    - **VMware cloud**
        - extend data center capacity to the AWS
        - manage on-premises data center
    - **MGN** - application migration service
        - Lift-and-shift(rehost) solution

- **Other services**
    - **GuardDuty** - threat detection
        - VPC Flow Logs, DNS logs, AWS CloudTrail events
        - uses integrated threat intelligence such as known **malicious IP addresses**, **anomaly detection**, and machine learning to identify threats more accurately.
    - **Cognito**
        - user pool
            - provide built-in user management or integrate with external identity providers, such as Amazon, facebook...
        - Identity pool
            - provide AWS credentials to grant your users access to other AWS services.
    - **AWS SES** - simple email service
        - fully managed service to send email securely, globally, at scale
    - **AWS Pinpoint**
        - scalable 2-way (in/outbound) marketing communications service
    - **AWS System manager**
        - SSM session manager
            - No open inbound ports and no need to manage bastion hosts or SSH keys
        - patch Manager
            - auto the process of patching managed instances

        - patch on-demand or on a schedule using Maintenance Windows
    - Maintenance windows
        - defines a schedule for when to perform actions on your instaces
    - Automation
        - simplifies common maintenance and deployment tasks of EC2 instances and other AWS resources
        - Automation Runbook - SSM documents to define actions preformed on EC2 instance or AWS resources
- **AWS Cost explorer**
    - Visualize
    - Forecast usage up to 12 months based on previous usage
    - choose an optimal savings plan
- **AWS Cost Anomaly Detection**
    - Continuously monitor cost and usage ML to detect unusual spends
- **AWS Batch**
    - fully managed batch processing at any scale
    - Batch will dynamically launch EC2 instance or Spot instances
    - Batch jobs are defined as Docker images and run on ECS
    - Batch vs. Lambda
        - Lambda
            - limited runtimes,
            - limited temporary disk space
            - serverless
        - Batch
            - no time limit
            - reply on EBS/instance store for disk space
            - relies on EC2
- **AWS AppFlow**
    - Fully managed integration service that enables you to securely transfer data between **SaaS application and AWS**
- **AWS Amplify**
    - develop and deploy scalable full stack web and mobile app