# Cheat Sheet

## Simple Storage Service (S3)

- Object based Storage with **unlimited** amount of data
- Replicates data across at least 3 AZs
- 99.99% Availability and 11 9s of durability
- Objects contain data from **0 Byte to 5TB**
- Buckets contain objects and folders containing objects
- Bucket names are unique globally between all AWS accounts
- Uploading a file gives you HTTP 200
- **Lifecycle Management** Objects can be moved between storage classes or deleted automatically
- **Versioning** Objects are given a Versioning ID, Old Objects are kept, cannot be turned off just suspended
- **MFA Delete** enforce DELETE operations to require MFA token(Versioning required)
- New Buckets are **private by default**
- Logging can be turned on
- **Access control** configured using **Bucket policies** (JSON) and **ACL**(legacy)
- **Security in Transit** over SSL
- Server Side Encryption: **SSE-AES, SSE-KMS, SSE-C**
- Client-Side Encryption: **CSE-KMS, CSE-C**
- **Cross Region Replication** for more durability (requires Versioning)
- **Transfer Acceleration** for faster and secure uploads through EdgeLocations
- Generate **Presigned URLs** via CLI and SDK for temporary access of private objects
- 6 Storage Classes: **Standard** (Fast), **Intelligent Tiering**, **Standard IA** (Cheaper Storage, more retrieval cost), **One Zone IA**, **Glacier**, **Glacier Deep Archive**

## Snowball

- **Snowball (Edge)** = container with storage device(**50-100TB**) for **peta-scale** migration
- **Snowmobile** = Shipping container on Truck(**100PB**) for **Exabyte-scale** migration
- **Low cost** and higher **speed** compared to network transfer
- **Export** or **import** data in **S3** or **Glacier**
- Snowball Edge offers local processing and edge-computing workloads

## Virtual Private Cloud (VPC)

### VPC Endpoint

- Keep traffic **between AWS services** in the AWS Network
- **Interface Endpoints cost money**, **Gateway Endpoints are free** but only support few services
- Interface Endpoints use **Elastic Network Interface** with Private IP
- Gateway Endpoints is target for specific route in route table

### VPC Flow Logs

- Monitor in and out traffic within VPC for VPC, Subnet or Network Interface
- **Cannot be tagged, cannot be changed after creation**
- Logs in **CloudWatch Logs**
- Contains source and destination IP(not hostnames)

### Network Access Control List (NACL)

- Each subnet must be associated with a NACL, only 1 at a time
- Inbound and outbound traffic rules (**allow or deny**), deny all by default
    - Rules are evaluated by order (lowest to highest)
- **Stateless** (allowed inbound is also allowed outbound)

### Security Groups

- Firewall at instance level
- Blocking all inbound and allowing all outbound by default (**Stateful** rules)
- EC2 instances can belong to multiple SG and SGs can contain multiple EC2s
- Rules only for allowing not denying

### Network Address Translation (NAT)

- Exist in **public subnets** with route from private subnets to NAT
- **HA** with **AutoscalingGroups**, multiple **subnets in different AZs**
- NAT Gateways are **redundant** in AZ
    - Only **1 per AZ**(not spanning multiple)
    - **Automatically assigned public IP**
    - **Route tables** must be updated

## Identity Access Management (IAM)

- Manage access to users and resources, applied to all regions
- Root initially created
- New users no permissions on creation
- Access Keys for CLI/SDK
- Setup **MFA** for Root account
- Set Password policies and rotation
- **IAM users, Groups** and **Roles**
- **IAM Policies**: JSON Documents to Allow/Deny
- **Managed Policies**: Read Only Policies from AWS
- **Customer Managed Policies**: created by customer
- **Inline Policies**: attached to a user

## Cognito

- Decentralized managed authentication system
- **User Pools** allows users to authenticate using OAuth IpD to connect to web-apps
  - **JSON Web Token** for authentication
- **Identity Pools** provide **temporary AWS credentials** to access services
- **Cognito sync** syncs **data** and **preferences** across devices with SNS
- **Web Identity Federation** exchange identity and security information between IdP and app
- **Identity Provider (IdP)**: trusted provider of user identity like Facebook, Google, Amazon, etc.
- **OIDC**: type of Identity Provider using OAuth
- **SAML**: type of identity provider using Single Sign-On

## Command Line Interface (CLI) and Software Development Kit (SDK)

- **CLI**: Let's you interact with AWS using a command line
- **SDK**: Set of tools and API libraries to control AWS services using programming languages
  - C++, Go, Java, JS, .NET, NodeJs, Python, PHP, Ruby
- **Programmatic Access** must be enabled via IAM to use CLI and SDK
  - Stored in text files (use roles instead of credentials when possible)

## Domain Name System (DNS)

- **IPv4**: 32 bit, **IPv6**: 128 bit
- **Top Level Domain** last part of domain name
- **Domain Registrar**: 3rd party company to register domains
- **Name Server**: servers which contain DNS records for domain
- **Start of Authority (SOA)**: contains information about DNS zone
- **A Record**: converts domain name into IP
- **CNAME Record**: converts domain name into another domain name
- **Time To Live (TTL)**: time DNS record will be cached

**Route53**

- DNS provider, register and manage domains, create record sets
- Simple Routing: Default Policy, multiple IP addresses with random pick
- Weighted Routing: Split traffic based on assigned weights
- Latency Based Routing: Directs traffic based on region, for lowest possible latency
- Failover Routing: primary site and secondary recovery site ( change on health check)
- Geolocation Routing: Direct traffic based on geographic location
- Geoproximity Routing: Direct traffic based on location and assigned bias to locations
- Multi-Value Answer: Like simple Routing Policies, using additional Health Checks to pick IP
- Traffic Flow: visual editor for chaining routing policies
- AWS Alias Record: smart DNS record, detecting change of IP and automatically adjusting
- Route 53 Resolver: regionally route DNS queries between VPCs and on-premise network
- Health Checks: to monitor availability of endpoints, allows automation

## Elastic Compute Cloud (EC2)

- Configure EC2 by choosing **OS, Memory, Storage, Network Throughput**
- **General Purpose**: balance of compute, memory, network (web servers, code repos)
- **Compute Optimized**: higher CPU (modeling, gaming servers, ad server engines)
- **Memory Optimized**: process large data in memory (databases, big data analysis)
- **Accelerated Optimized**: hardware accelerators (ML, speech recognition)
- **Storage Optimized**: high sequential read/write on local storage (NoSQL, data warehousing)
- Instance sizes **generally double** in prize and key attributes
- **Placement groups** to choose logical placement of instances
- **UserData**: Customer provided Script which runs automatically when launching EC2 instances
- Access **MetaData** via endpoint at http://169.254.169.254/latest/metadata
- **InstanceProfiles**: container for IAM role attached to instances

### Pricing

- **On Demand** (least commitment)
  - Low cost, flexible, pay by hour
  - **Use case**: short term, spiky, unpredictable workloads
- **Reserved Instances** (best long-term value)
  - Reduced Pricing based on Term x Class Offering x Payment Option
  - **Payment Terms**: 1 or 3 years
  - **Payment Options**: All Upfront, Partial Upfront, No Upfront
  - **Class Offering**:
    - **Standard** (75%, can't change attributes)
    - **Convertible** (54%, change attributes)
    - **Scheduled** (reserved for time periods like one day a week)
  - **Use case**: steady state or predictable usage
- **Spot Instances** (biggest savings)
  - Request spare compute capacity
  - Can be **terminated by AWS** at any time
  - **Use case**: non critical background jobs, can handle interruptions
- **Dedicated Instances** (most expensive)
  - Single Tenancy
  - On-demand or reserved
  - **Use case**: Need of isolated hardware (regulations)

## Amazon Machine Image (AMI)

- Contains information required to launch EC2 instance
- Region specific, needs to be copied to other regions, AMI ID varies in regions
- Create AMIs from running or stopped existing instances
- Community AMI: free, maintained by community
- AWS Marketplace: free or paid subscription AMIs maintained by vendors
- Holds following information:
  - Template for root volume, Launch permissions, Block device mapping

## Auto Scaling Groups (ASG)

- Collection of grouped EC2 instances
- Scale Out: Add instances, Scale in: Remove instances, Scale up: increase size of instance
- Size of ASG based on **Min**, **Max**, **Desired Capacity**
- **Target Scaling Policy**: based on target value for metric (e.g. CPU exceeds 75%)
- **Simple scaling Policy**: policy triggers when alarm is breached
- **Scaling Policy with Steps**: when alarm is breached with escalation on alarm values
- Health Checks determine current state on instances, can be run against EC2 or ELB
- **Launch configurations** used to launch instances, cannot be edited

## Elastic Load Balancer (ELB)

- Must have at least two AZs
- Cannot go cross-region
- **Application Load Balancer (ALB):**
    - uses **Listeners**, **Rules** and **Target Groups**
    - based on HTTP(S) → good for web applications
    - WAF can be attached
    - Advanced Request Routing rules possible
- **Network Load Balancer (NLB)**:
    - uses **Listeners** and **Target Groups**
    - based on UDP/TCP → good for high network throughput
- **Classic Load Balancer (CLB)**:
    - uses **Listeners** and instances are **registered directly** as targets
    - not recommended (legacy)
- X-Forward-For(XFF) to get origin IP address
- Amazon Certification Manager SSL can be attached to all ELBs
- Sticky sessions for CLB or ALB and remembered via Cookie

## Elastic File System (EFS)

- Volumes automatically grow and shrink (petabyte scale)
- Stored across multiple AZs in Region
- Mount multiple EC2 instances to same EFS in one VPC
- Read after Write Consistency

## Elastic Block Store (EBS)

- Virtual hard disk with Snapshots(incremental, exist on S3, can be taken when instance is running)
- **Durable block-level storage** that can be attached to single EC2 instance
- You can create AMIs from Volumes or Snapshots
- **Instance Store: temporary** storage type physically attached to host machine

## Cloud Front

- **Content Delivery Network (CDN)** making websites load fast by ser5ving cached content
- Distributes cached copies to **Edge Locations** (read, write objects)
- **Time to Live (TTL)** defines how long until cache expires
- **Refreshing cache costs money**
- **Origin**: address of where original files are stored (S3,EC2,ELB,Route53)
- 2 Types of **Distribution**: Web Distribution and RTMP (streaming)
- **Origin Identity Access (OIA)** is used to access private S3 objects
- **Lambda@Edge** allows to pass request through Lambdas to change response behavior

## Relational Database Service RDS

- Instances managed by AWS , no SSH possible
- AWS Aurora, MySQL, PostgresSQL, MariaDB, Oracle, MSSQL
- **Multi-AZ** to make exact copies in other AZ for synchronizing and automatic failover
- **Read-Replicas** to run multiple read only copies with asynchronous replication
  - Up to 5, also Cross-Region
- Multi-AZ and read Replicas can be combined
- Automated Backups: choose retention period for backup, no cost for backup storage
- Manual Snapshots: manually create snapshots, persistent even when deleting instance
- Encryption at-rest possible with KMS

## Aurora

- **Fully-managed** Postgres/MySQL db with autoscale, auto backups HA and fault tolerance
- 5x faster than MySQL, 3x faster than Postgres with 1/10 of cost
- Replicates **6 copies** over **3 AZs** with up to 15 replicas
- Can span multiple regions with **Aurora Global Database**
- **Aurora Serverless automatically starts and stops Aurora** for low volume apps and keeping costs down

## Redshift

- Fully-Managed Petabyte-size Data Warehouse loaded from S3, EMR, DynamoDB
- Is Columnar Storage Database: Stores data together data in columns instead of rows
- Uses SQL like queries and is Online Analytics Processing System (OLAP)
- Used for Data Warehousing →Business Intelligence
- **Single-AZ** Service(Snapshots can be restored to different AZ)
- Single Node (160GB) or Multi Node (up to 128)
- Bill per hour for node, not billed for leader node
- **Dense Compute Node**: high performance, less storage
- **Dense Storage Node**: clusters which you have lot of data
- Attempts to maintain 3 copies
- Similar data stored sequential on disk, not requiring indexes
- Uses Massively Parallel Processing (MPP)
- Backups enabled by default with 1 day retention (up to 35 days) stored in S3
- Data-in-transit and Data-at-rest Encryption

## DynamoDB

- Fully managed NoSQL key-value and document database
- Scales with whatever read and write capacity you specify
- Eventually consistent read: data returned immediately but data can be inconsistent on updates
- Strongly Consistent Reads: will wait for upgrades with slower possible read times
- Copies will be consistent with guarantee of 1 second
- Stores 3 copies of data on SSD over 3 Regions

## CloudFormation

- Provisioning of AWS resources through templates (YAML, JSON) → **IaC**
- When detecting errors **ROLLBACK_IN_PROGRESS** will be triggered
- **NestedStacks** to break down templates into smaller reusable ones
- Content:
  - **MetaData**
  - **Description**
  - **Parameters** (Values to pass to template at runtime)
  - **Mappings** (Maps Keys to values)
  - **Conditions** (If-Else)
  - **Transform** (Applies Macros)
  - **Resources** (Resource you want to create, requires **at least one**)
  - **Outputs** (Return values)

## CloudWatch

- Collection of monitoring services
- **CloudWatch Logs**: log data from AWS services
- **CloudWatch Metrics**: time ordered set of data points for variable (CPU, Memory, etc.)
- **CloudWatch Events**: trigger event based on condition
- **CloudWatch Alarms**: triggers notifications based on metrics when threshold is breached
- **CloudWatch Dashboards**: create visualization based on metrics
- Logs must belong to **Log Groups**
- **CloudWatch Agent** needed for EC2 Instance details (Memory, CPU, etc.)
- EC2 monitors every 5 mins, every min with Detailed monitoring
- Most other services monitor every min

## CloudTrail

- Logs API calls between AWS services
- Keywords: **Governance compliance, operational auditing, risk auditing**
- Logs by default for past 90 days via **Event History**
- **Trail** to track beyond 90 days, stored on S3, analyze with Athena
- **Log File validation** Option to ensure integrity
- Can be encrypted with **KMS**
- Can be set to log all accounts of Organization
- Can be set to log over all Regions
- **Management Events**: management operations eg. Routing settings, creating users, etc. turned on by default
- **Data Events**: Logs S3 or Lambda, disabled by default

## Lambda

- **Serverless functions** without provision or manage servers
- Good fit for short running tasks, **Fargate** for longer tasks
- 7 runtime language environments**: Ruby, Python, Java, Go, Powershell, NodeJs, C#**
- Pay per invocation (**duration** and amount **memory** used), First 1M requests per month free
- Duration timeout up to **15 mins** and memory up to **3008 MB**
- Triggered from SDK or by other Services
- Can scale to **1000 concurrent functions** (can be increased by AWS team)
- Cold starts: delay if function not recently executed

## Simple Queue Service (SQS)

- Queuing service using messages with queue
- Used for Application Integration
- To read need to **pull** queue using SDK
- Standard (nearly unlimited messages per sec, but messages can be delivered out of order) or First-In-First-Out (limit of 300 per second) queues
- Short polling (default): returns messages immediately even if queue is pulled empty
    - when you need message right away
- Long polling: Waits until message arrives in queue or long poll timeout expires
    - Makes it inexpensive to retrieve messages as soon as they are available
    - Will reduce cost, most use-cases should use it
- **Visibility time-out**: period of time messages are invisible in queue because messages will be deleted after job has processed
- Retention: 60 seconds to 14 days, 4 default

## Simple Notification Service (SNS)

- Fully managed pub/sub messaging service for **Application Integration**
- **Topic**: logical access point and communication channel, able to deliver multiple protocols
- Encrypt Topics with KMS
- **Publishers** use AWS API, CLI or SDK to push messages to topic
- **Subscriptions** subscribe to topics and get messages immediately
- HTTP(s), Email, Email-JSON, Amazon SQS, AWS Lambda, SNS, Platform application endpoints

## ElastiCache

- Managed **in-memory** caching service (**temporary storage area**)
- Resources only **within same VPC** may be connected
- **Memcached**: key/value store preferred for HTML fragments, faster than redis
- **Redis**: richer data types and operations, for leaderboards, unread notifications, etc.

## Elastic Beanstalk

- Handles deployment (load balancing, provisioning, auto-scaling, monitoring)
- Run web-apps without infrastructure knowledge
- Free service, only instances cost money
- Java, .NET, PHP, Node.js, Python, Ruby, Go, Docker

## API Gateway

- Create secure APIs at any scale
- Front door for applications
- Limit 10,000 requests per seconf
- Stages allow to have multiple API versions, each with Invoke URL
- Need to publish API via Deploy API and choose the stage
- Resources are URLs(projects) and can have child Resources
- Define Methods (POST, GET, …)
- CORS can be enabled (disabled by default)
- Caching for improved latency and reduced call amount
- Same Origin Policies to prevent XSS

## Kinesis

- **Collecting**, **processing** and **analyzing** streaming data in **real-time**
- **Data Streams**: pay per shard, data can persist (**24h to 168h**), multiple consumers possible
- **Firehose**: Pay for ingested data, data immediately disappears, choose consumer from predefined list
- **Data Analytics**: perform queries in real-time needs streams as input and output
- **Video Analytics**: ingest and store video/audio data to ML services or Video services
- Kinesis Producer Library KPL to write data to stream

## Storage Gateway

- Connects on-prem storage to cloud storage
- **File Gateway**: S3 acts as local filesystem using NFS or SMB
- **Volume Gateway**: used for backups to AWS
  - **Stored Volume Gateway**: Primary data on-prem, **1GB to 16TB**
  - **Cached Volume Gateway**: Primary data in S3, local data as cache, **1GB to 32GB**
- **Tape Gateway**: backups virtual tapes to S3 Glacier