



S3 CheatSheet

- **Simple Storage Service (S3)** Object-based storage. Store **unlimited** amount of data without worry of underlying storage infrastructure
- S3 replicates data across at least 3 AZs to ensure 99.99% Availability and 11' 9s of durability
- Objects contain your data (they're like files)
- Objects can be size anywhere from **0 Bytes** up to 5 Terabytes
- Buckets contain objects. Buckets can also contain folders which can in turn can contain objects.
- Bucket names are unique across all AWS accounts. Like a domain name.
- When you upload a file to S3 successfully you'll receive a HTTP 200 code
- **Lifecycle Management** Objects can be moved between storage classes or objects can be deleted automatically based on a schedule
- **Versioning** Objects are given a Version ID. When new objects are uploaded the old objects are kept. You can access any object version. When you delete an object the previous object is restored. Once Versioning is turned on it cannot be turned off, only suspended.
- **MFA Delete** enforce DELETE operations to require MFA token in order to delete an object. Must have versioning turned on to use. Can only turn on MFA Delete from the AWS CLI. Root Account is only allowed to delete objects
- All new buckets are **private by default**
- Logging can be turned on a bucket to log to track operations performed on objects
- **Access control** is configured using **Bucket Policies** and **Access Control Lists (ACL)**
- **Bucket Policies** are JSON documents which let you write complex control access
- **ACLs** are the legacy method (not deprecated) where you grant access to objects and buckets with simple actions



S3 CheatSheet

- S3 has **6 different** Storage Classes:
 - **Standard** Fast! 99.99% Availability, 11 9's Durability. Replicated across at least three AZs
 - **Intelligent Tiering** Uses ML to analyze your object usage and determine the appropriate storage class. Data is moved to the most cost-effective access tier, without any performance impact or added overhead.
 - **Standard Infrequently Accessed (IA)** Still Fast! Cheaper if you access files less than once a month. Additional retrieval fee is applied. 50% less than Standard (reduced availability)
 - **One Zone IA** Still Fast! Objects only exist in one AZ. Availability (is 99.5%). but cheaper than Standard IA by 20% less (Reduce durability) Data could get destroyed. A retrieval fee is applied.
 - **Glacier** For long-term cold storage. Retrieval of data can take minutes to hours but the off is very cheap storage
 - **Glacier Deep Archive** The lowest cost storage class. Data retrieval time is 12 hours.



S3 CheatSheet

- **Security in Transit** Uploading files is done over SSL
- **SSE** stands for Server Side Encryption. S3 has **3 options** for SSE.
- **SSE-AES** S3 handles the key, uses AES-256 algorithm
- **SSE-KMS** Envelope encryption via AWS KMS and you manage the keys
- **SSE-C** Customer provided key (you manage the keys)
- **Client-Side Encryption** You must encrypt your own files before uploading them to S3
- **Cross Region Replication (CRR)** allows you to replicate files across regions for greater durability. You must have versioning turned on in the source and destination bucket. You can have CRR replicate to bucket in another AWS Account
- **Transfer Acceleration** provide faster and secure uploads from anywhere in the world. data is uploaded via distinct url to an Edge Location. Data is then transported to your S3 bucket via AWS backbone network.
- **Presigned Urls** is a url generated via the AWS CLI and SDK. It provides temporary access to write or download object data. Presigned Urls are commonly used to access private objects.



VPC Endpoint *CheatSheet*

- VPC Endpoints help keep traffic between AWS services **within the AWS Network**
- There are two kinds of VPC Endpoints. Interface Endpoints and Gateway Endpoints
- Interface Endpoints **cost money**, Gateway Endpoints **are free**
- Interface Endpoints uses an Elastic Network Interface (ENI) with Private IP (powered by AWS PrivateLink)
- Gateway Endpoints is a target for a specific route in your route table
- Interface Endpoints support many AWS services
- Gateway Endpoint only support DynamoDB and S3



VPC Flow Logs *CheatSheet*

- **VPC Flow Logs** monitor the in-and-out traffic of your Network Interfaces within your VPC
- You can turn on Flow Logs at the VPC, Subnet or Network Interface level
- VPC Flow Logs **cannot be tagged** like other AWS resources
- You **cannot change the configuration** of a flow log **after it's created**.
- You **cannot enable** flow logs for VPCs which are peered with your VPC **unless it is in the same account**
- VPC Flow Logs can be delivered to an **S3** or **CloudWatch Logs**
- VPC Flow Logs contains the source and destination **IP addresses** (not hostnames)
- Some instance traffic is **not monitored**:
 - Instance traffic generated by contacting the AWS DNS servers
 - Windows license activation traffic from instances
 - Traffic to and from the instance metadata address (169.254.169.254)
 - DHCP Traffic
 - Any traffic to the reserved IP address of the default VPC router



NACLs *CheatSheet*

- Network Access Control List is commonly known as NACL
- VPCs are automatically given a default NACL which allows **all** outbound and inbound traffic.
- Each subnet within a VPC must be associated with a NACL
- Subnets can only be associated with 1 NACL at a time. Associating a subnet with a new NACL will remove the previous association.
- If a NACL is not explicitly associated with a subnet, the subnet will automatically be associated with the default NACL.
- NACL has inbound and outbound rules (just like Security Groups).
- Rule can either **allow** or **deny** traffic. (unlike Security Groups which can only allow)
- NACLs are STATELESS (any allowed inbound traffic is also allowed outbound)
- When you create a NACLs it will deny all traffic by default
- NACLs contain a numbered list of rules that gets evaluated in order from lowest to highest.
- If you needed to block a single IP address you could via NACLs (Security Groups cannot deny)



Security Groups *CheatSheet*

- Security Groups acts as a firewall at the instance level
- Unless allowed specifically, all **inbound traffic** is **blocked by default**.
- All **Outbound traffic** from the instance is **allowed by default**.
- You can specify for the source to be either an IP range, single IP Address or another security group
- Security Groups are **STATEFUL** (if traffic is allowed inbound it is also allowed outbound)
- Any changes to a Security Group take effect immediately.
- EC2 Instances can belong to multiple security groups
- Security groups can contain multiple EC2 Instances.
- You **cannot block specific IP addresses** with Security Groups, for this you would need a Network Access Control List (NACL)
- You can have up to 10,000 Security Groups per Region (default 2,500)
- You can have 60 inbound and 60 outbound rules per Security Group
- You can have 16 Security Groups associated to an ENI (default is 5)



NAT Instance and NAT Gateway *CheatSheet*

- When creating a NAT instance you **must disable source and destination checks** on the instance
 - NAT instances **must exist in a public subnet**
 - You must have a **route out** of the private subnet to the NAT instance
 - The size of a NAT instance determines **how much traffic can be handled**
 - **High availability** can be achieved using **Autoscaling Groups**, multiple **subnets in different AZs**, and **automate failover between them using a script**.
-
- NAT Gateways are **redundant inside an Availability Zone** (can survive failure of EC2 instance)
 - You can only have **1 NAT Gateway inside 1 Availability Zone** (cannot span AZs)
 - Starts at 5 Gbps and scales all the way up to 45 Gbps
 - NAT Gateways are the **preferred setup for enterprise systems**.
 - There is no **requirement to patch NAT Gateways**, and there is no **need to disable Source/Destination checks** for the NAT Gateway (unlike NAT Instances)
 - NAT Gateways are **automatically assigned a public IP address**
 - **Route Tables** for the NAT Gateway **MUST** be updated
 - Resources in multiple AZs sharing a Gateway will **lose internet access if the Gateway goes down**, unless you create a **Gateway in each AZ** and configure **route tables** accordingly



IAM CheatSheet

- **Identity Access Management** is used to manage **access** to users and resources
- IAM is a universal system. (applied to all regions at the same time). IAM is a free service
- A root account is the account initially created when AWS is set up (full administrator)
- New IAM accounts have no permissions by default until granted
- New users get assigned an Access Key Id and Secret when first created when you give them programmatic access
- Access Keys are only used for CLI and SDK (cannot access console)
- Access keys are only shown once when created. If lost they must be deleted/recreated again.
- Always setup MFA for Root Accounts
- Users must enable MFA on their own, Administrator cannot turn it on for each user
- IAM allows your set password policies to set minimum password requirements or rotate passwords
- **IAM Identities** as Users, Groups, and Roles
- **IAM Users** End users who log into the console or interact with AWS resources programmatically
- **IAM Groups** Group up your Users so they all share permission levels of the group
- eg. Administrators, Developers, Auditors
- **IAM Roles** Associate permissions to a Role and then assign this to an Users or Groups
- **IAM Policies** JSON documents which grant permissions for a specific user, group, or role to access services. Policies are attached to to IAM Identities
- **Managed Policies** are policies provided by AWS and cannot be edited
- **Customer Managed Policies** are policies created by use the customer, which you can edit
- **Inline Policies** are policies which are directly attached to a user



Cognito CheatSheet

- Cognito is decentralized managed authentication system. When you need to easily add authentication to your mobile and desktop app *think* Cognito
- **User Pools** user directory, allows users to authenticate using OAuth to IdP such as Facebook, Google, Amazon to connect to web-applications. Cognito User Pool is in itself a IdP
- User Pools use **JWTs** for to persist authentication
- **Identity Pools** provide **temporary AWS credentials** to access services eg. S3, DynamoDB
- **Cognito Sync** can sync **user data** and **preferences** across devices with one line of code (powered by SNS)
- **Web Identity Federation** exchange identity and security information between an identity provider (IdP) and an application
- **Identity Provider (IdP)** a trusted provider of your user identity that lets you use authenticate to access other services. eg. Facebook, Twitter, Google, Amazon
- **OIDC** is a type of Identity Provider which uses OAuth
- **SAML** is a type of Identity Provider which is used for Single Sign-on



AWS CLI &



SDK *CheatSheet*

- **CLI** stands for Command Line Interface
- **SDK** stands for Software Development Kit
- The **AWS CLI** lets you interact with AWS from anywhere by simply using a command line
- The **AWS SDK** is a set of API libraries that let you integrate AWS services into your applications.
- **Programmatic Access** must be enabled per user via the IAM console to use CLI or SDK
- **aws configure** command used to setup your AWS credentials for the CLI
- The CLI is installed via a Python script
- Credentials get stored in a plain text file (whenever possible use roles instead of AWS credentials)
- The SDK is available for the following programming languages
 - C++
 - Go
 - Java
 - Javascript
 - .NET
 - NodeJs
 - PHP
 - Python
 - Ruby



DNS CheatSheet

- **Domain Name System (DNS)** - Internet service that converts domain names into routable IP addresses
- **IPv4** - Internet Protocol Version 4 - 32 bit address space (**limited** number of addresses)
- IPv4 eg. **52.216.8.34**
- **IPv6** - Internet Protocol Version 6 - 128 bit address space (**unlimited** number of addresses)
- IPv6 eg. **2001:0db8:85a3:0000:0000:8a2e:0370:7334**
- **Top-Level Domain** example.**com** last part of the domain
- **Second-Level Domain** example.**CO**.UK **second last part of the domain**
- **Domain Registrar** 3rd party company who you register domains through
- **Name Server** The server(s) which contain the DNS records for a domain
- **Start of Authority (SOA)** Contains information about the DNS zone and associated DNS records
- **A Record** DNS record which directly converts a domain name into an IP address
- **CNAME Record** DNS record which lets you convert a domain name into another domain name
- **Time to Live (TTL)** The time that a DNS record will be cached for (lower time means changes propagate faster)



Route53 CheatSheet

- Route53 is a DNS provider, register and manage domains, create record sets. Think Godaddy or NameCheap
- Simple Routing - Default routing policy, multiple addresses result in a random endpoint selection
- Weighted Routing - Split up traffic based on different 'weights' assigned (percentages)
- Latency-Based Routing - Directs traffic based on region, for lowest possible latency for users.
- Failover Routing - Primary site in one location, secondary data recovery site in another. (change on health check)
- Geolocation Routing - Route traffic based on the geographic location of a requests origin.
- Geo-proximity Routing - Route traffic based on geographic location using 'Bias' values (needs Route53 Traffic Flow)
- Multi-value Answer Routing - Return multiple values in response to DNS queries. (using health checks)
- Traffic Flow - visual editor, for chaining routing policies, can version policy records for easy rollback
- AWS Alias Record - AWS' smart DNS record, detects changed IPs for AWS resources and adjusts automatically.
- Route53 Resolver - Lets you regionally route DNS queries between your VPCs and your network **Hybrid Environments**
- Health checks can be created to monitor and automatically over endpoints. You can have health checks monitor other health checks



EC2 CheatSheet

- **Elastic Compute Cloud (EC2)** is a Cloud Computing Service
- Configure your EC2 by choosing your **OS, Storage, Memory, Network Throughput**.
- Launch and SSH into your server **within minutes**.
- EC2 comes in variety Instance Types specialized for different roles:
 - **General Purpose** balance of compute, memory and networking resources
 - **Compute Optimized** Ideal for compute bound applications that benefit from high performance processor
 - **Memory Optimized** fast performance for workloads that process large data sets in memory.
 - **Accelerated Optimized** hardware accelerators, or co-processors
 - **Storage Optimized** high, sequential read and write access to very large data sets on local storage
- Instance Sizes **generally double** in price and key attributes
- **Placement Groups** let you to choose the logical placement of your instances to optimize for communication, performance or durability. Placement groups are free.
- **UserData** a script that will be automatically run when launching an EC2 instance.
- **MetaData** meta data about the current instance. You access this meta data via a local endpoint when SSH'd into the EC2 instance. eg. curl <http://169.254.169.254/latest/meta-data> meta data could be the instance type, current ip address etc...
- **Instance Profiles** a container for an IAM role that you can use to pass role information to an EC2 instance when the instance starts.



EC2 Pricing – *CheatSheet*

- EC2 has for 4 pricing models **On-Demand**, **Spot**, **Reserved Instances (RI)** and **Dedicated**
- **On-Demand** (least commitment)
 - low cost and flexible
 - only pay per hour
 - **Use case:** short-term, spiky, unpredictable workloads, first time apps
 - Ideal when your workloads cannot be interrupted
- **Reserved Instances** upto 75% off (Best long-term value)
 - **Use case:** steady state or predictable usage
 - Can resell unused reserved instances (Reserved Instance Marketplace)
 - Reduced Pricing is based on **Term** x **Class Offering** x **Payment Option**
 - **Payment Terms:** 1 year or 3 year
 - **Payment Options:** All Upfront, Partial Upfront, and No Upfront
 - **Class Offerings**
 - **Standard** Up to 75% reduced pricing compared to on-demand. Cannot change RI Attributes.
 - **Convertible** Up to 54% reduced pricing compared to on-demand. Allows you to change RI Attributes if greater or equal in value.
 - **Scheduled** You reserve instances for specific time periods eg. once a week for a few hours. Savings vary



EC2 Pricing – *CheatSheet*

- **Spot Pricing** upto 90% off (Biggest Savings)
 - request spare computing capacity
 - flexible start and end times
 - **Use case:** Can handle interruptions (server randomly stopping and starting)
 - **Use case:** For non-critical background jobs
 - Instances can be terminated by AWS **at anytime**
 - If your instance is **terminated by AWS, you don't get charged** for a partial hour of usage.
 - If **you terminate** an instance **you will still be charged** for any hour that it ran.
- **Dedicated Hosting** (Most Expensive)
 - Dedicated servers
 - Can be on-demand or reserved (upto 70% off)
 - **Use case:** When you need a guarantee of isolate hardware (enterprise requirements)



AMI CheatSheet

- **Amazon Machine Image (AMI)** provides the information required to launch an instance.
- AMIs are region specific, if you need to use an AMI in another region you can copy an AMI into the destination region via **Copy AMI**
- You can **create an AMI** from an existing EC2 instance that's either **running** or **stopped**.
- **Community AMI** are free AMIs maintained by the community
- **AWS Marketplace** free or paid subscription AMIs maintained by vendors
- AMIs have an **AMI ID**. The same AMI eg. (Amazon Linux 2) will vary in both AMI ID and options eg. Architecture options in different regions
- **An AMI holds the following information:**
 - A template for the root volume for the instance (EBS Snapshot or Instance Store template) eg. an operating system, an application server, and applications
 - Launch permissions that control which AWS accounts can use the AMI to launch instances.
 - A block device mapping that specifies the volumes to attach to the instance when it's launched.



EC2 Auto Scaling Groups *CheatSheet*

- An ASG is a collection of EC2 instances grouped for scaling and management
- Scaling Out is when add servers
- Scaling In is when you remove servers
- Scaling Up is when you increase the size of an instance (eg. updating Launch Configuration with larger size)
- Size of an ASG is based on a **Min**, **Max** and **Desired Capacity**
- **Target Scaling policy** scales based on when a target value for a metric is breached eg. Average CPU Utilization exceed 75%
- **Simple Scaling** policy triggers a scaling when an alarm is breached
- **Scaling Policy with Steps** is the new version of Simple Scaling policy and allows you to create steps based on eculation alarm values.
- Desired Capacity is how many EC2 instances you want to ideally run
- An ASG will always launch instances to meet minimum capacity
- Health checks determine the current state of an instance in the ASG
- Health checks can be run against either an ELB or the EC2 instances
- When an Autoscaling launches a new instance it uses a Launch Configuration which holds the configuration values for that new instance eg. **AMI**, **InstanceType**, **Role**
- Launch Configurations cannot be edited and must be cloned or a new one created
- Launch Configurations must be manually updated in by editing the Auto Scaling settings.



ELB CheatSheet

- There are three Elastic Load Balancers: **Network**, **Application** and **Classic** Load Balancer
- A Elastic Load Balancer must have **at least two** Availability Zones.
- Elastic Load Balancers **cannot go cross-region**. You must create one per region.
- ALB has **Listeners**, **Rules** and **Target Groups** to route traffic
- NLB use **Listeners** and **Target Groups** to route traffic
- CLB use **Listeners** and EC2 instances are **directly registered** as targets to CLB
- Application Load Balancer is for HTTP(S) traffic and the name implies it good for Web Applications
- Network Load Balancer is for TCP/UDP is good for high network throughput eg. Video Games
- Classic Load Balancer is legacy and its recommended to use ALB or NLB
- Use X-Forwarded-For (XFF) to get original IP of incoming traffic passing through ELB
- You can attach Web Application Firewall (WAF) to ALB but not to NLB or CLB
- You can attach Amazon Certification Manager SSL to any of the Elastic Load Balancers for SSL
- ALB has advanced Request Routing rules where you can route based on subdomain header, path and other HTTP(S) information
- Sticky Sessions can be enable for CLB or ALB and sessions are remembered via Cookie



EFS CheatSheet

- **Elastic File System** (EFS) supports the Network File System version 4 (NFSv4) protocol.
- You pay GB of storage per month
- Volumes can scale to petabyte size storage
- Volumes will shrink and grow to meet current data stored (elastic)
- Can support thousands of concurrent connections over NFS.
- Your data is stored across multiple AZs within a region.
- Can mount multiple EC2 instance to a single EFS (as long as they are all in the same VPC)
- Creates Mount Points in all your VPC subnets so you can mount from anywhere within your VPC
- Provides Read After Write Consistency.



EBS CheatSheet

- **Elastic Block Store (EBS)** is a virtual hard disk. Snapshots are a point-in-time copy of that disk.
- Volumes exist on EBS. Snapshots exist on S3.
- Snapshots are incremental, only changes made since the last snapshot are moved to S3.
- Initial Snapshots of an EC2 instance will take longer to create than subsequent Snapshots
- If taking Snapshot of a root volume, the EC2 instance should be stopped before Snapshotting
- You can take Snapshots while the instance is still running.
- You can create AMIs from Volumes, or from Snapshots.
- **EBS Volumes** A **durable**, block-level storage device that you can attach to a single EC2 instance
- **EBS Volumes** can be modified on the fly eg. storage type or volume size.
- Volumes always exist in the same AZ as the EC2 instance.
- **Instance Store Volumes** A **temporary** storage type located on disks that are physically attached to a host machine.
- **Instance Store Volumes** (ephemeral) cannot be stopped. If the host fails then you lose your data.
- EBS Backed instances can be stopped and you will not lose any data.
- By default root volumes are deleted on termination.
- **EBS Volumes** can have termination protection (don't delete the volume on termination)
- Snapshots or restored encrypted volumes will also be encrypted.
- You cannot share a snapshot if it has been encrypted.
- Unencrypted snapshots can be shared with other AWS accounts or made public.



CloudFront *CheatSheet*

- CloudFront is a CDN (Content Distribution Network). It makes website load fast by serving cached content that is nearby
- CloudFront distributes cached copy at **Edge Locations**
- Edge Locations aren't just not read-only, you can write to them eg. PUT objects
- **TTL** (Time to live) defines how long until the cache expires (refreshes cache)
- When you invalidate your cache, you are forcing it to immediately expire (refreshes cached data)
- Refreshing the cache **costs money because of transfer costs** to update Edge Locations
- **Origin** is the address of where the original copies of your files reside eg. S3, EC2, ELB, Route53
- **Distribution** defines a collection of Edge Locations and behaviour on how it should handle your cached content
- **Distributions** has 2 Types: **Web Distribution** (static website content) **RTMP** (streaming media)
- **Origin Identity Access (OAI)** is used access private S3 buckets
- Access to cached content can be protected via **Signed Urls** or **Signed Cookies**
- **Lambda@Edge** allows you to pass each request through a Lambda to change the behaviour of the response.



RDS CheatSheet

- Relational Database Service (RDS) is the AWS Solution for relational databases.
- RDS instances are managed by AWS, You cannot SSH into the VM running the database.
- There are 6 relational database options currently available on AWS, Aurora, MySQL, MariaDB, Postgres, Oracle, Microsoft SQL Server
- Multi-AZ is an option you can turn on which makes an exact copy of your database in another AZ that is only standby
- For Multi-AZ AWS automatically synchronizes changes in the database over to the standby copy
- Multi-AZ has Automatic Failover protection if one AZ goes down failover will occur and the standby slave will be promoted to master
- Read-Replicas allow you to run **multiples copies** of your database, these copies only allows **reads** (no writes) and is intended to alleviate the workload of your primary database to improve performance
- Read-Replicas use Asynchronous replication
- You must have automatic backups enabled to use Read Replicas



Aurora CheatSheet

- When you need a **fully-managed** Postgres or MySQL database that needs to scale, automatic backups, high availability and fault tolerance *think* Aurora
- Aurora can run MySQL or Postgres database engines
- Aurora MySQL is 5x faster over regular MySQL
- Aurora Postgres is 3x faster over regular Postgres
- Aurora is 1/10 the cost over its competitors with similar performance and availability options.
- Aurora replicates **6 copies** for your database across **3 availability zones**.
- Aurora is allowed up to **15 Aurora Replicas**
- An Aurora database can span multiple regions via **Aurora Global Database**
- **Aurora Serverless** allows you to stop and start Aurora and scale automatically while keeping costs low
- Aurora Serverless is ideal for new projects or projects with infrequent database usage



Redshift *CheatSheet*

- Data can be loaded from S3, EMR, DynamoDB, or multiple data sources on remote hosts.
- Redshift is Columnar Store database which can SQL-like queries and is an OLAP.
- Redshift can handle petabytes worth of data. Redshift is for Data Warehousing
- Redshift most common use case is Business Intelligence
- Redshift can only run in a 1 availability zone (**Single-AZ**)
- Redshift can run via a single node or multi-node (clusters)
- A single node is 160 GB in size
- A multi-node is comprised of a leader node and multiple compute nodes
- You are bill per hour for each node (excluding leader node in multi-node)
- You are not billed for the leader node
- You can have up to 128 compute nodes
- Redshift has two kinds of Node Type **Dense Compute** and **Dense Storage**
- Redshift attempts to backup 3 copies of your data, the original, on compute node and on S3
- Similar data is stored on disk sequentially for faster reads
- Redshift database can be encrypted via KMS or CloudHSM
- Backup Retention is default to 1 day and can be increase to maximum of 35 days
- Redshift can asynchronously back up your snapshot to Another Region delivered to S3
- Redshift uses Massively Parallel Processing (MPP) to distribute queries and data across all loads
- In the case of empty table, when importing Redshift will sample data to create a schema.



DynamoDB *CheatSheet*

- **DynamoDB** is a fully managed **NoSQL** key/value and document database.
- Applications that contain large amounts of data but require predictable read and write performance while scaling is a good fit for DynamoDB
- DynamoDB scales with whatever **read and write capacity you specific** per second.
- DynamoDB can be set to have **Eventually Consistent Reads (default)** and **Strongly Consistent Reads**
- **Eventually consistent reads** data is returned immediately but data can be inconsistent. Copies of data will be generally consistent in 1 second.
- **Strongly Consistent Reads** will wait until data is consistent. Data will never be inconsistent but latency will be higher. Copies of data will be consistent with a guarantee of 1 second.
- DynamoDB stores 3 copies of data on SSD drives across 3 regions.



CloudFormation *CheatSheet*

- When being asked to **automate** the provisioning of resources *think* CloudFormation
- When Infrastructure as Code (IaC) is mentioned *think* CloudFormation
- CloudFormation can be written in either JSON or YAML
- When CloudFormation encounters an error it will rollback with **ROLLBACK_IN_PROGRESS**
- CloudFormation templates larger than 51,200 bytes (0.05 MB) are too large to upload directly, and must be imported into CloudFormation via an S3 bucket.
- **NestedStacks** helps you break up your CloudFormation template into smaller reusable templates that can be composed into larger templates
- **At least one resource** under resources: must be defined for a CloudFormation template **to be valid**
- **MetaData** extra information about your template
- **Description** a description of what the template is suppose to do
- **Parameters** is how you get user inputs into templates
- **Transforms** Applies marcos (like applying a mod which change the anatomy to be custom)
- **Outputs** are values you can use to import into other stacks
- **Mappings** maps keys to values, just like a lookup table
- **Resources** defines the resources you want to provision, **at least one resource is required**
- **Conditions** are whether resources are created or properties are assigned



CloudWatch *CheatSheet*

- CloudWatch is a collection of monitoring services: **Dashboards, Events, Alarms, Logs** and **Metrics**
- CloudWatch **Logs**: log data from AWS services. eg. CPU Utilization
- CloudWatch **Metrics**: Represents a time-ordered set of data points, A variable to monitor eg. CPU Utilization over time
- CloudWatch **Events**: trigger an event based on a condition eg. ever hour take snapshot of server
- CloudWatch **Alarms**: triggers notifications based on metrics when a defined threshold is breached
- CloudWatch **Dashboards**: create visualizations based on metrics
- EC2 monitors at 5 min intervals and at Detailed Monitoring 1 minute intervals
- Most other service monitor at 1 minute intervals, with intervals of 1 , 3 , 5 minutes.
- Logs must belong to a **Log Group**
- CloudWatch Agent needs to be installed on EC2 host to track **Memory Usage** and **Disk Size**
- You can stream custom log files eg. production.log
- Custom Metrics allow you to track High Resolution Metrics a sub minute intervals all the way down to 1 second.



CloudTrail *CheatSheet*

- CloudTrail logs calls between AWS services
- **governance, compliance, operational auditing**, and **risk auditing** are keywords relating to CloudTrail
- When you need to know **who to blame** *think* CloudTrail
- CloudTrail by default logs event data for the past 90s days via **Event History**
- To track beyond 90 days you need to create **Trail**
- To ensure logs have not been tampered with you need to turn on **Log File Validation** option
- CloudTrail logs can be encrypted using **KMS (Key Management Service)**
- CloudTrail can be set to log across all **AWS** accounts in an Organization and all regions in an account.
- CloudTrail logs can be streamed to CloudWatch logs
- Trails are outputted to an S3 bucket that you specify
- CloudTrail logs two kinds of events: **Management Events** and **Data Events**
- **Management events** log management operations eg. AttachRolePolicy
- **Data Events** log data operations for resources (S3, Lambda) eg. GetObject, DeleteObject, and PutObject
- Data Events are **disabled** by default when creating a Trail.
- Trail logs in S3 can be analyzed using Athena



Lambda CheatSheet

- **Lambda's** are serverless **functions**. You upload your code and it runs without you managing or provisioning any servers.
- Lambda is **serverless**. You don't need to worry about underlying architecture
- Lambda is a good fit for short running tasks where you don't need to customize the os environment. If you need long running tasks (> 15mins) and a custom OS environment than consider using **Fargate**
- There are **7 runtime language environments** officially supported by Lambda: **Ruby, Python, Java, NodeJs, C#, Powershell** and **Go**
- You pay per invocation (The **duration** and the amount of **memory** used) rounded up to the nearest 100 milliseconds and you based on amount of requests. First 1M requests per month are free
- You can adjust the duration timeout for up to **15 mins** and memory up to **3008 MB**
- You can trigger Lambdas from the SDK or multiple AWS services eg. S3, API Gateway, DynamoDB
- Lambdas by default run in No VPC. To interact with some services you need to have your Lambda in the same VPC eg. RDS
- Lambda can scale to **1000 of concurrent functions** in seconds. (1000 is the default, you can increase with AWS Service Limit Increase)
- Lambdas have **Cold Starts**. If a function has not been recently been execute there will be a delay



Simple Queue Service (SQS) *CheatSheet*

- SQS is a queuing service using messages with a queue. Think Sidekiq or RabbitMQ
- SQS is used for Application Integration, it lets decoupled services and apps to talk to each other
- To read SQS use need to **pull** the queue using the AWS SDK. SQS is **not pushed-based**
- SQS supports both Standard and First-In-First-Out (FIFO) queues
- Standard allows nearly unlimited messages per second, does not guarantee order of delivery, always delivers at least once, you must protect against duplicate messages being processed
- FIFO maintain the order of messages with a 300 limit
- There are two kinds of polling Short (Default) and Long Polling
- Short polling returns messages immediately, even if the message queue being polled is empty.
- Long polling waits until message arrives in the queue, or the long poll timeout expires.
- In majority of cases Long polling is preferred over short polling.
- **Visibility time-out** is the period of time that messages are invisible in the SQS queue
- Messages will be deleted from queue after a job has processed. (before visibility timeout expires)
- If Visibility timeout expires then a job will become visible to the queue
- The default Visibility time-out is 30 seconds. Timeout can be **0 seconds** to a maximum of 12 hours.
- SQS can retain messages from 60 seconds to 14 days and by default is 4 days
- Message size between 1 byte to 256 kb, Extended Client Library for Java can increase to 2GB



SNS CheatSheet

- **Simple Notification Service (SNS)** is a fully managed pub/sub messaging service
- SNS is for **Application Integration**. It allows decoupled services and apps to communicate with each other
- **Topic** a logical access point and communication channel.
- A topic is able to deliver to multiple protocols
- You can encrypt topics via KMS
- **Publishers** use the AWS API via AWS CLI or SDK to push messages to a topic. Many AWS services integrate with SNS and act as publishers.
- **Subscriptions** subscribe to topics. When a topic receives a message it automatically and immediately pushes messages to subscribers.
- All messages published to SNS are stored redundantly across multiple Availability Zones (AZ)
- The following protocols:
 - **HTTPs and HTTPs** create webhooks into your web-application
 - **Email** good for internal email notifications (only supports **plain text**)
 - **Email-JSON** sends you json via email
 - **Amazon SQS** place SNS message into SQS queue
 - **AWS Lambda** triggers a lambda function
 - **SMS** send a text message
 - **Platform application endpoints** Mobile Push eg. Apple, Google, Microsoft Baidu notification systems



ElastiCache *CheatSheet*

- **ElastiCache** is a managed **in-memory** caching service
- ElastiCache can launch either **Memcached** or **Redis**
- **Memcached** is a simple key / value store preferred for caching HTML fragments and is arguably faster than Redis
- **Redis** has richer data types and operations. Great for leaderboard, geospatial data or keeping track of unread notifications.
- A cache is a **temporary storage** area.
- Most frequently identical queries are stored in the cache
- Resources only **within the same VPC** may connect to ElastiCache to ensure low latencies.



Elastic Beanstalk *CheatSheet*

- **Elastic Beanstalk** handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring
- When you want to run a web-application but you don't want to have think about the underlying infrastructure.
- It costs nothing to use Elastic Beanstalk (only the resources it provisions eg. RDS, ELB, EC2)
- Recommended for test or development apps. Not recommended for production use
- You can choose from the following preconfigured platforms: Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker
- You can run dockerized environments on Elastic Beanstalk.



API Gateway CheatSheet

- API Gateway is a solution for creating secure APIs in your cloud environment at any scale.
- Create APIs that act as a front door for applications to access data, business logic, or functionality from back-end services.
- API Gateway throttles api endpoints at **10,000** requests per second (can be increase via service request through AWS support)
- **Stages** allow you to have multiple published versions of your API eg. prod, staging, QA
- Each Stage has an **Invoke URL** which is the endpoint you use to interact with your API
- You can use a custom domain for your Invoke URL eg. api.exampro.co
- You need to publish your API via Deploy API. You choose which Stage you want to publish your API
- Resources are your URLs eg. /projects
- Resources can have child resources eg. /projects/**-id/-edit**
- You defined multiple **Methods** on your Resources eg GET, POST, DELETE
- CORS issues are common with API Gateway, CORS can be enabled on all or individual endpoints
- Caching improves latency and reduces the amount of calls made to your endpoint
- Same Origin Policies help to prevent XSS attacks
- Same Origin Policies ignore tools like postman or curl
- CORS is always enforced by the client.
- You can require **Authorization** to your API via AWS Cognito or a custom Lambda.



Kinesis CheatSheet

- **Amazon Kinesis** is the AWS solution for **collecting, processing, and analyzing streaming data** in the cloud. When you need **“real-time”** think Kinesis.
Kinesis Data Streams Per per running shard, data can persist within the stream, data is ordered and every consumer keep its own position. Consumers have to be manually added (coded), Data persists for **24 hours (default) to 168 hours**
- **Kinesis Firehose** - Pay for only the data ingested, data **immediately disappears** once processed. Consumer of choice is from a predefined set of services: S3, Redshift, Elasticsearch or Splunk
- **Kinesis Data Analytics** - allows you to perform **queries in real-time**. Needs a Kinesis Data Streams/Firehose as the input and output.
- **Kinesis Video Analytics** securely ingests and stores video and audio encoded data to consumers such as SageMaker, Rekognition or other services to apply Machine learning and video processing.
- KPL (Kinesis Producer Library) is a Java library to write data to a stream
- You can write data to stream using AWS SDK, but KPL is more efficient



Storage Gateway *CheatSheet*

- **Storage Gateway** connects on-premise storage to cloud storage (hybrid storage solution)
- There are three types of Gateways: File Gateway, Volume Gateway, Tape Gateway
- **File Gateway** lets S3 act a local file system using NFS or SMB, extends your local hard drive to S3
- **Volume Gateway** is used for backups and has two types: **Stored** and **Cached**
- **Stored Volume Gateway** continuously backups local storage to S3 as EBS Snapshots **Primary Data on-Premise**
- Stored Volumes are **1GB to 16TB** in size
- **Cached Volume Gateway** caches the frequently used files on-premise. **Primary Data** is stored on S3
- Cached Volumes are **1GB to 32GB** in size
- **Tape Gateway** backups up virtual tapes to S3 Glacier for long archive storage