CloudWatch CheatSheet

- The Pillars of Observability are **Metrics**, **Logs** and **Traces.** You need to use all three together to obtain **Observability**
- AWS CloudWatch is a collection of monitoring services:
  - **CloudWatch Logs** – centralization log management service. Logs from AWS Services, Application Logs running on your EC2s
  - **CloudWatch Log Insights** — Robust way to filter your logs, it auto-detects log patterns creates dynamic fields, compose queries.
  - **CloudWatch Metrics** — Aggerates datapoints from AWS Service to produce a metric. eg AvgCPUUtilization
  - **CloudWatch Alarms** — Define a threshold up on a metric and reacts by triggering action such an EC2, Auto Scaling or SNS action
  - **CloudWatch Events (EventBridge)** — A serverless account wide event bus. Create event-driven loosely coupled apps with AWS Services
  - **CloudWatch Dashboards** — Create dashboards filled with graphs powered by CloudWatch metrics
  - **CloudWatch Container Insights** — collect, aggregate, and summarize metrics and logs from your containerized apps and microservices
  - **CloudWatch Contributor Insights** — view the top contributors impacting the performance of your systems and applications in real-time
  - **CloudWatch Synthetics** — Test your web-apps to see if they're broken. Uses puppeteer underneath.
  - **CloudWatch Service Lens** — "Observability" for distributed or serverless apps by pulling X-Ray Traces, CloudWatch Logs, and Metrics

CloudWatch Logs CheatSheet

CloudWatch Logs is used to **monitor**, **store**, and **access** your log files

- Export logs to S3 to analyze with Athena
- Stream logs to ElasticSearch Service to run full-text search
- Stream CloudTrail Events to CloudWatch Logs to allow you to react to CloudTrail Events
- Encrypted by default, You can apply your own KMS Key
- Retains logs indefinitely by default you can choose a custom retention period between 1 Day and 10 years
- Support a simple Filtering Syntax
- **Log Groups** — A container for collection of log streams. Uses the forward slash naming convention eg. /my-app/prod/us-east/
- **Log Streams** — Represent a log, usually created by AWS Services, can be manually created
- **Log Events** — Represents a single event (a single line from the log file)
- In order to send custom application logs running on your EC2 instance you need to install the **SSM CloudWatch Agent**
- When the agent is install you update the **/etc/awslogs/awslogs.conf** on your EC2 instance to include the logs you want to send to CloudWatch

CloudWatch Logs Insights CheatSheet

CloudWatch Logs Insights enables you to **interactively search and analyze your CloudWatch log data**

- CloudWatch Logs Insights supports all kind of logs
- CloudWatch Logs Insights is generally used from the AWS Console to create ad-hoc queries
- CloudWatch Logs Insights has its own language called CloudWatch Logs Insights **Query Syntax**
- A single request can query up to **20 log groups**.
- Queries **time out after 15 minutes**, if they have not completed.
- Query results are **available for 7 days.**
- CloudWatch Logs Insights comes with a bunch of sample queries
- CloudWatch Logs Insights analyzes the log events in imported logs to generated out **Dynamic Fields**
- CloudWatch Logs Insights always generates out 5 of the following fields:
  - **@message** — the raw unparsed log event.
  - **@timestamp** — the event timestamp contained in the log event's timestamp field.
  - **@ingestionTime** — the time when the log event was received by CloudWatch Logs.
  - **@logStream** — the name of the log stream that the log event was added to.
  - **@log** — is a log group identifier in the form of account-id:log-group-name.
- CloudWatch auto-discovery fields for the following AWS Services
  - Amazon VPC Flow Logs
  - Amazon Route53
  - AWS Lambda
  - AWS CloudTrail
- JSON logs keys will all become discovered fields
- For other logs that are not automatically discovered you can use the parse command to create ephemeral fields

CloudWatch Metrics CheatSheet

CloudWatch Metrics represents a **time-ordered set of data points** Its a **variable** that is **monitored over time**.

- Metrics are data about the performance of your systems.
- CloudWatch comes with many **predefined** metrics that are generally name spaced by AWS Service.
- Very common predefined metrics at the EC2 Per-Instance Metrics: CPUUtilization, DiskReadOps, DiskWriteOps, DiskReadBytes, DiskWriteBytes,NetworkIn, NetworkOut, NetworkPacketsIn, NetworkPacketsOut
- Metric data is kept for 15 months, enabling you to view both up-to-the-minute data and historical data.
- **Metric math** enables you to **query multiple CloudWatch metrics** and **use math expressions** to create new time series
- You can publish your own metrics using the AWS CLI or AWS sDK eg. aws cloudwatch put-metric-data
- You can define custom metrics in two different resolutions:
  - standard resolution (1 minute)
  - high resolution (> 1 minute to 1 second)
- With High Resolution you can track in intervals of: 1 second, 5 seconds, 10 seconds, 30 seconds, multiple of 60 seconds.
- Data availability (when you can see the data) varies based on service
  - **EC2 Basic Monitoring (5 minutes)**
  - EC2 Detailed Monitoring (1 minute)
  - Other Services generally 1 minutes but could also be 3 or 5 minutes
- You can turn on EC2 Detailed Monitoring for a price
- Not all EC2 metrics are tracked by default you have to install the CloudWatch Agent to collect
  - **Memory utilization (Am I running out of memory?)**
  - Disk Swap utilization
  - **Disk Space utilization (Am I running out of disk space?)**
  - Page file utilization

CloudWatch Alarms CheatSheet

A CloudWatch Alarm monitors a **CloudWatch Metric** based on **a defined threshold.**
When an alarm **breaches** (goes outside the defined threshold) than it changes **state**.

Metric Alarm States

- **OK** The metric or expression is **within** the defined threshold
- **ALARM** The metric or expression is **outside** of the defined threshold
- **INSUFFICIENT_DATA**
  - The alarm has **just started**
  - the metric is **not available**
  - **Not enough data** is available

When it changes state we can define what **action it should trigger:**

- Notification
- Auto Scaling Group
- EC2 Action

- You can define Conditions of either a **Static** or **Anomaly Detection**
  - **Static** set a static value as the threshold eg. 100 USD
  - **Anomaly Detection** sets a band around the data points, helps prevents false-positives, more flexible if you have seasonal data
- **Composite Alarms** — Allows you watch multiple alarms and require both to trigger before resulting in an alarm action
  - The alarms being watch must have no actions sets
  - You can only trigger an SNS as the action (so no EC2 or ASG actions)

EventBridge (CloudWatch Events) CheatSheet

EventBridge is a serverless **account wide** or **organization** event bus. Create event-driven loosely coupled apps with AWS Services

- EventBridge was formally known as CloudWatch Events (both are still accessible within the AWS console)
- **An event bus receives events** from a **source** and **routes events** to a **target** based on **rules**
- EventBridge/CloudWatch **Events** are json objects that are used to pass data around
- Many AWS Services emit Event data, but for AWS services that do not, you can turn on CloudTrail and react to those events.
- AWS API call events that are **larger than 256 KB** in size are not supported.
- To react to event data you need to create an **EventBridge Rule**
  - You can choose up to 5 AWS Service Target eg. Lambda Function, SQS queue, SNS topic, Firehose delivery stream, ESC Task
  - When you select a target you have additional fields to narrow do the AWS Service you want to target
  - You can apply Event Matching to filter what events should be passed to the Target
    - You simply provide a json schema with the fields you want to match on, or you can use complex matchers:
    - **Prefix Matching** match on the prefix of a value in the event source
    - **Anything-but Matching** matches anything except what's provided in the rule
    - **Numeric Matching Matches** against numeric operator for “<“, “>”, “=“, “<=“, “>=“
    - **IP Address Matching** matching against available for both IPv4 and IPv6 addresses
    - **Exists Matching** matching works on the presence or absence of a field in the JSON
    - **Empty Value Matching** For strings you can use “” to match empty For other values you can null
    - **Multiple Matching** combine multiple matching rules into a more complex event pattern
  - You can set **Configure Input** which is used to transform / filter the **event’s data structure** that will be passed:
    - **Match Event** The entire event pattern text is passed to the target when the rule is triggered.
    - **Part of the matched event** only the part of the event text that you specify is passed to the target.
    - **Constant (JSON text)** send static content instead of the matched event data. (Mocked JSON)
    - **Input Transformer** You can transform for the event text a different format of a string or a JSON object

Event Bridge (CloudWatch Events) CheatSheet

- A common use case of EventBridge is to use it as a serverless cronjob. Eg. Trigger database backup everyday
- EventBridge can schedule events using either **Cron Expression** or **Rate Expressions**
  - **Cron Expression** – very fine tune control eg. 15 12 * MON-FRI * = 12pm UTC Monday to Friday
  - **Rate Expression** — Easy to set, not as fine grained eg. Choose every X Hours / Minutes / Days
  - All scheduled events use **UTC time zone**
  - the minimum precision for schedules is **1 minute**.
- The event bus can extend to third-party SaaS Products to accept **Partner Event Sources** eg. React to Datadog event and trigger
- EventBridge Schema allows you to keep track of changes to your event schemas an:
  - It will automatically detect the schema changes and create versions
  - You can download the Schema as Code Bindings
  - You can use the AWS Toolkit VSCode plugin to view generated Schemas or apply Code Bindings

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

OpsWorks *CheatSheet*

OpsWorks is a **configuration management** service for the following:

- **OpsWorks Stacks** – Deploy AWS apps using Chef 11 or 12 and a GUI that simplifies different stack layers
- **OpsWorks Chef Automate** — Fully managed server of Chef Automate
- **OpsWorks Chef Enterprise** — Fully managed server of Chef Enterprise

OpsWorks Stacks

- A managed version of **Chef 11** and **Chef 12**
- Can attach a Load Balancer (only Classic Load Balancer)
- Can defined Container Layers and RDS Layers
- OpsWorks Stacks is composed of the following:
  - **Stack** The container for the entire stack.
  - **Layers** a blueprint for a set of Amazon EC2 instances
  - **Instances** An instance represents a server.
  - **Apps** An app represents code stored in a repository that you want to install on application server instances.
- You can grant IAM users access to as OpsWorks Users, and set their permissions eg. Allowed to SSH into server.
- A layer has 5 lifecycle events that you can attach a chef recipe to:
  1. Setup —after a started instance has finished booting
  2. Configure - on all of the stack's instances when:
     - Instance enters/leaves the online state.
     - Elastic IP address associated/disassociate from an instance.
     - Elastic Load balancer attach/detached to a layer
  3. Deploy — when you run a deploy command
  4. Undeploy — when you delete an app or run an undeploy command to remove an app from a set of application server instances.
  5. Shutdown —you direct OpsWorks Stacks to shut an instance down but before the associated EC2 instance is actually terminated.

OpsWorks CheatSheet

OpsWorks Stacks

- **Run commands** allow you to **perform manual operations** on your Stack and underlying instances.
  - **Update Custom Cookbooks** Updates the instances' custom cookbooks with the current version from the repository.
  - **Execute Recipes** Executes a specified set of recipes on the instances.
  - **Setup** Runs the instances' Setup recipes
  - **Configure** Runs the instances' Configure recipes
  - **Upgrade Operating System** (Amazon Linux 1 Only) Upgrades the instances Amazon Linux operating systems to the latest version.
    - You can set AWS OpsWorks to reboot the instance after installing the upgrade (default to YES).
- **App Settings** represents the web-app settings that will be applied to your running instances:
  - **Set the Application Source** the location of your application code on S3, Git Repo, or an HTTP Archive
  - **Set Environment Variables**
  - **Pass an SSL Certificate** (terminates at the instance, end-to-end encryption)
  - **Map a custom domains** (uses Classic Load Balancer)
- **Three Kinds of Layers:**
  - OpsWorks Layer (EC2 Instances)
  - ECS Layer
    - You have to create the ECS cluster first than create and associate layer
  - RDS Layer
    - You have to create the RDS instance first than create and associate layer
  - Load Balancer Layer
    - You create a CLB and then you edit your OpsWorks layer and associate the Load balancer
- **Stack Settings** you can change the default Operation System and Chef Cookbooks repo
  - If you are using **Chef 11** **you can use Berkshelf** to provide your Custom Chef cookbooks.

OpsWorks CheatSheet

OpsWorks Stacks

- You can create two special types of instances:
  - **Time-based** automatically starts and stops instances based on a specified schedule.
  - **Load-based** automatically starts and stops instances in response to CPU, memory, and application load changes across all the instances in a layer.
- OpsWorks provides you a monitoring dashboard called Monitoring Layers
  - **Monitoring Layers** allows you to averaged stats across all your Instances within that layer.
  - **OpsWorks is tracking Memory Used by default** Normal EC2 instances require to manually install the CloudWatch Agent.
- OpsWorks Stacks **installs on every Linux instance** the **OpsWork Stacks Agent CLI**

Chef is a company and the name of a **configuration management tool** written in Ruby

- A **cookbook** defines a scenario and contains everything that is required to support that scenario
- A **recipe** is the most fundamental configuration element within the organization
- **Knife** is a CLI (command line interface) between a local chef repository and the Chef Infra Server.
- **Berskelf** is a **dependency manager** for Chef cookbooks.
  - Similar to a Gemfile, requirements.txt or package.json but called Berkshelf
- Chef recipes are written in Ruby language DSL

Puppet is a server automation framework and application.

- Puppet **has its only programming language** for describing how servers should be configured.
- **Manifest File** Holds all the resource declarations
- A **resource** describes something about the state of the system, such as a certain user or file should exist, or a package should be installed.

Cloud Networking *CheatSheet*

The **Open Systems Interconnection (OSI) Model** defines standards of communication for Telecom and Computer Systems.

- There are 7 layers to the OSI model:
  1. Physical - responsible for transmitting **raw bits** as a **physical signal** to the the destination network
  2. Data Link - responsible for the packaging data into **Frames** to transfer to **network nodes** on the same layer
  3. Network - is responsible for routing (forwarding) IP addresses.
  4. Transport - responsible for end-to-end connections and reliability.
  5. Session - responsible for **creating**, **maintaining** and **destroying** sessions.
  6. Presentation - **formats and delivers** information to the Application Layer
  7. Application - The **closet to the end-user.** Used by software applications such as Email, Web-Apps, Shell Terminals…
- You will want to remember Layer 3 (Network), 4 (Transport) and 7 (Application) for Cloud Networking
  - AWS DDOS protection occurs on Layer 3,4 and 7
  - AWS WAF protection occurs on Layer 7
  - AWS Application Load Balancer operates on Layer 7 (HTTP/S)
  - AWS Network Load Balancer operates on Layer 4 (TCP/UDP)
- **A Network Interface Controller (NIC)** connects a computer to a computer network (operate on OSI Layer 1 and 2)
  - AWS has virtual NICs called **Amazon VPC Elastic network interfaces (ENIs)**
    - Every EC2 instance has at least one ENI
    - EC2 instance can have multiple ENIs attached

Cloud Networking *CheatSheet*

- An IP address serves two main functions:
  - host or network interface identification (**Who** is this?)
  - location addressing (**Where** do they live in the network?)
- There are two versions of IP currently in use:
  - **IPv4** (invented 1981) — has available **4 billion** addresses
  - **IPv6** (invented 1995) — has available ~**340 undecillion** addresses
- An IPv4 address has a size of **32 bits** and uses Dotted Decimal Notation eg. 192.168.0.1
- An IPv4 address has a size of **128 bit** and uses Hexadecimal Notation eg. 0123:4567:89ab:cdef:0123:4567:89ab:cdef
- **Binary** is a **base-2** numeral system **of 0 or 1**. It called Binary because we only use two numbers.
- A **bit** is a basic unit of information that **represents either a 0 or 1**.
- A **byte** is a basic unit of information that **represents consecutive bits.**
- The most common type of byte is eight consecutive bytes known as an **octet**
- In Binary Math every number doubles when its 1, from right to left: 1,2,4,8,16,32,64,128,256
  - 0000 0001 = 1
  - 0000 0011 = 3
  - 1111 1111 = 256
  - 1000 0000 = 128
- Private Address Space commonly used are 10.0.0.0 (Class A), 172.168.0.0 (Class B), 192.168.0 (Class C)
- **Dynamic Host Configuration Protocol (DHCP)** server **dynamically assigns** an **IP address** to a device on the network
  - With AWS VPC you can change the DHCP settings to another DNS server (maybe for extra security)

Cloud Networking *CheatSheet*

- **A Subnet Mask** is a 32-bit number looks like an IP address **but it is not**. eg. 255.0.0.0
  - A Subnet masks divides the IP address into a the NET ID and HOST ID via masking.
  - A Subnet Mask is also sometimes referred to as a **Netmask** but 99.9% of the time they mean the same thing.
- AWS reserved 5 IP addresses when you defined a CIDR eg using 10.0.0.0/28.
  - 10.0.0.0 **Reserved by AWS for Network Address**
  - 10.0.0.1 **Reserved by AWS VPC Router**
  - 10.0.0.2 **Reserved by AWS**
  - 10.0.0.3 **Reserved by AWS**
  - 10.0.0.16 ***Reserved for Broadcast Address**
  - If you allocate /24 (256 addresses) you'll only have 251 host addresses you can assign.
- **Classless Inter-Domain Routing (CIDR)** is A method of allocating IP addresses for IP routing where you can choose the choose the size of Networks vs Hosts
- **CIDR block range** defines the size of Network vs Host and the range of IP Addresses. eg. 10.0.0.0/24
- **CIDR Notation** defines the subnet mask (how many host addresses will be available) eg. /24 = 256 possible addresses
  - The number in CIDR notation indicates leading bits flipped to 1. /24 = 1111 . 1111 . 0000 . 0000
  - /32 = 1 address
  - /24 = 256 addresses
  - /23 = 128 addresses
  - /16 = 65,536 addreses

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

Service Catalog CheatSheet

- **AWS Service Catalog** enables organizations to **create and manage catalogs of products** that are **approved for use** on AWS to achieve consistent governance and meet compliance requirements.
- There are two types of Service Catalog users:
  1. **Catalog Administrator**: They managed the catalog
  2. **End Users**: They use the catalog
- A **product** is a **CloudFormation template** that defines the resources that will be launched
  - You can create or associate an **AWS Budget** to a product
  - Once a product is created it cannot be edited only **deleted**.
  - Also a product must be removed from the portfolio and not provisioned by a user in order to delete.
  - In order for products to be made visible to users it needs to added to a portfolio.
- **A Portfolio** Is a **collection of products** with the following options:
  - **Constraints** can restrict how products are used.
  - **Permissions** determine who can see and launch products is by associating either **Groups, roles or users**
  - You cannot restrict certain products to certain users in a portfolio, you have to just make additional portfolios
- When a Product is launched its called a **Provisioned Product**
- You can attach **Service Actions** (Which are just SSM Documents) to a product so end-users can preform governed actions on the product.

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 scrip**t.

- 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** fo 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

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)

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.

Systems Manager CheatSheet

AWS Systems Manager is also known as AWS Simple Systems Manager (SSM)

- AWS SSM is an **umbrella of AWS services** used to automate the management of Virtual Machines.
- In order for VMs to use SSM you need two things:
  1. **SSM Agent** installed on the EC2 instances
     - Already Installed on AWS Managed AMIs eg. Amazon Linux 1/2, Ubuntu 16/18 …
  2. **SSM AWS Managed Policy** attached to your EC2 IAM Role (**AmazonSSMManagedInstanceCore**)
- **SSM Explorer** is a customizable **operations dashboard** that reports information about your AWS resources
- **OpsCenter** is a consolidation of your operational work items (**OpsItems)** for you IT team to: View, Investigate, Resolve
  - **OpsItems** represents operational work that needs to performed eg. ”EC2 instance disk full”
- **SSM Run Command** lets you remotely and securely **manage the configuration of your managed instances**.
  - Execute a Bash or PowerShell Command, Run Chef Recipes or Ansible Playbooks …
- **SSM Automation** is a service that allows you to define documents (runbooks) that execute a sequence of commands carrying out the runbook actions.
  - A runbook is document that contains a series of instructional steps to perform an operation.
- **AWS Tags** are words or phrases that act as metadata for organizing your AWS resources
- **Resource Groups** are a collection of AWS resources that share one or more **tags**
- Resource Groups can display details about a group of resource based on Metrics, Alarms and Configuration Settings
- **SSM AppConfig** is used to create, manage, and quickly deploy **application configuration**
  - When you need to rollout changes to your application configuration files, and need to avoid errors such as typos that could break your production environment
- **SSM Change Calendars** allows you to **define when** SSM Automations **are allowed not allowed to be executed** by your team.

Systems Manager CheatSheet

- **SSM Maintenance Windows** let you **define a schedule for when to perform potentially disruptive actions** on your instances such as patching an operating system, updating drivers, installing or patching software.
- **SSM Configuration Compliance** **scans your fleet of managed instances** for **patch compliance and configuration inconsistencies**
- **SSM Inventory** provides visibility into your Amazon EC2 and on-premises computing environment.
- **SSM Activations** allows you to **register external resources to be managed by AWS Systems Managers**
  - Activations issue you a Code and ID that functions like an EC2 Access ID and Secret to your managed external instances.
- **SSM Session Manager** lets you manage your EC2 instances, on-premises instances, and virtual machines (VMs) through an **interactive one-click browser-based shell** or through the AWS CLI.
- **SSM State Manager** is a secure and scalable configuration management service that automates the process of keeping your Amazon EC2 and hybrid infrastructure in a state that you define.
- **SSM Patch Manager** automates the process of **patching managed instances** with both security related and other types of updates.
- **SSM Distributor** lets you **package your own software** or **install AWS-provided agent software packages**, such as **AmazonCloudWatchAgent** to install on AWS Systems Manager managed instances.
- **SSM Documents** are variety of code management documents relating to different SSM Services.
  - SSM Documents are either **YAML or JSON files** with parameters and a series of steps.
  - SSM Documents console **consolidates** all your documents into one place.

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.

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

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.

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.

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

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.

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 acce
- 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

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 giving 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 turn 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 to 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

- **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.

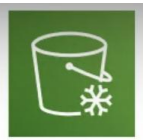
S3 CheatSheet

- 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 Glacier CheatSheet

S3 Glacier is an **extremely low-cost storage service.** **durable** storage with **security** features for data archiving and backup

- Common use case: company needs to hold financial tax records for of 7 years to meet gov and financial regulations
- S3 Glacier is **automatically server-side encrypted** using 256-bit Advanced Encryption Standard (AES-256)
- As an additional safeguard, AWS encrypts the key itself with a master key that is **regularly rotated**.
- **Data-in-transit** between S3 and S3 Glacier via lifecycle policies is **encrypted** using **SSL**
- S3 Glacier Data Model is made up of **Vault**, **Archive** and a **Job**
  - **Vault** A vault is a container for storing archives
  - **Archive**: The Base unit of storage, Can be photo, video or document
  - **Job**: Perform an operation on the Archive
    - perform a select query on an archive
    - retrieve an archive
    - get an inventory of a vault
- Because **jobs** take time to complete, S3 Glacier supports a notification mechanism to **notify you when a job is complete.**
- A **Vault Inventory** refers to the **list of archives in a vault**.
  - The vault inventory automatically **updates once a day**
- To Retrieve an Archive (File) you **InitiateJob** as the type **Archive Retrieval**
- With S3 Select you can write **simple SQL expressions** to pull **only the bytes you need** from those objects
- When initiating a select or an archive retrieval job you can specify how fast you want the data



S3 Glacier **CheatSheet**

When initiating a select or an archive retrieval job you can **specify how fast you want** the data.

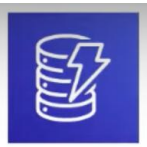
- **Expediated Tier** (1-5 mins, costly) For urgent requests. Limited to 250 MB archive size
- **Standard Tier** (3-5 hours, cheap) No archive size limit. This is **default** option
- **Bulk Tier** (5-12 hours, very cheap) No archive size limit, even petabytes worth of data.
- There are three ways to get data into S3 Glacier (there is no console like S3 to upload files):
  - **S3 Snowball, Snowball Edge or Snowmobile**
  - **S3 Lifecycle Policies**
  - **S3 Multipart Upload API**
- To delete a Vault you must first **delete all files** and there have been **no writes to the vault since last inventory** (inventory every 24hours)
- **Vault Access policy** controls **who** is able to access the vault.
  - Require MFA to delete files
  - Don't let files delete for this period of time
- **Vault Lock policy** how a vault can be modified for a period of time. Locking a vault takes two steps:
  - Initiate the lock by attaching a vault lock policy to your vault. (24 hours window where you can stop the lock)
  - Use the lock ID to complete the lock process.
- S3 Glacier has 3 data **retrieval policies**
  1. **No Retrieval Limit (default)** No retrieval quota is set and all valid data retrieval requests are accepted.
  2. **Free Tier Only** Keep your retrievals within your daily free tier allowance and not incur any data retrieval cost.
  3. **Max Retrieval Rate** Control the peak retrieval rate by **specifying a data retrieval quota** that has a bytes-per-hour maximum.
- S3 Glacier provisioned capacity allows you to pay a fixed up-front fee to save money.
- S3 Stores Metadata along side the archive. Uploading many small archives can result in additional data meaning greater costs
  - Try to store large archives to reduce this "metadata tax"

Snowball & Snowball Edge & Snowmobile *CheatSheet*

- **Snowball** and **Snowball Edge** is a rugged container which contains a storage device
- **Snowmobile** is a a 45-foot long ruggedized shipping container, pulled by a semi-trailer truck.
- Snowball and Snowball Edge is for **peta-scale** migration. Snowmobile is for **exabyte-scale** migration
- **Low Cost** thousands of dollars to transfer 100TB over high speed internet. Snowball is **1/5th**
- **Speed** 100 TB over 100 days to transfer over high speed internet, Snowball takes **less than a week**
- **Snowball come in two sizes:**
  - **50 TB** (42 TB of usable space)
  - **80 TB** (72 TB of usable space)
- **Snowball Edge comes in two sizes:**
  - **100 TB** (83 TB of usable space)
  - **100 TB Clustered** (45 TB per node)
- **Snowmobile comes in one size:** 100PB
- You can both **export** or **import** data using Snowball or Snowmobile
- You can import into **S3** or **Glacier**
- **Snowball Edge** can undertake local processing and edge-computing workloads
- **Snowball Edge** Can use in a cluster in groups of 5 to 10 devices
- **Snowball Edge** provides three options for device configurations
  - storage optimized (24 vCPUs)
  - compute optimized (54 vCPUs)
  - GPU optimized (54 vCPUs)

RDS CheatSheet

- You can have upto 5 read replicas
- You can combine Read Replicas with Multi-AZ
- You can have Read Replicas in another Region (Cross-Region Read Replicas)
- Replicas can be promoted to their own database, but this breaks replication
- You can have Replicas of Read Replicas
- RDS has 2 backup solutions: Automated Backups and Database Snapshots
- Automated Backups, you choose a retention period between 1 and 35 days, There is no additional cost for backup storage, you define your backup window
- Manual Snapshots, you manually create backups, if you delete your primary the manual snapshots will still exist and can be restored
- When you restore an instance it will create a new database. You just need to delete your old database and point traffic to new restored database
- You can turn on encryption at-rest for RDS via KMS



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 in 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 Cheat Sheet

- 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**
- CloudFormation **Template Sections**
  - **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

CloudFormation Cheat Sheet

- Stack Updates can be preformed two different ways:
  - Direct updates
    - You directly update the stack
    - You submit changes and AWS CloudFormation immediately deploys them
  - Executing Change Sets
    - You can preview the changes to CloudFormation will make to your stack (Change Set)
    - Then decide whether you want to apply those changes
  - Stack Updates will change state of your resources based on circumstances:
    - **Update with No Interruption** Updates the resource **without disrupting** operation and **without changing** the resource's physical ID
    - **Updates with Some Interruptions** Updates the resource **with some interruption** and **retains** the physical ID.
    - **Replacement** **recreates** the resource during an update, also **generates new** physical ID.
    - You can use a **StackPolicy** to prevent stack updates on resources to prevent data loss or interruption to services
  - **Drift Detection** feature lets CloudFormation tell you when your expected configuration has changed to due manual overrides. eg. A CFN creates an SG but a Developer deletes it.

CloudFormation Cheat Sheet

- **Rollbacks** occur when a CloudFormation encounters an error when you create, update or destroy a stack
  - When a rollback is in progress you'll see **ROLLBACK_IN_PROGRESS**
  - When a rollback succeeds you'll see **UPDATE_ROLLBACK_COMPLETE**
  - When a rollback fails you'll see **UPDATE_ROLLBACK_FAILED**
- **Pseudo Parameters** are predefined parameters eg. !Ref AWS:Region return us-east-1
- Resource Attributes
  - **CreationPolicy** – prevent its status from reaching create complete until AWS CloudFormation receives a specified number of success signals or the timeout period is exceeded.
  - **DeletionPolicy** – reserve or (in some cases) backup a resource when its stack is deleted **Delete**, **Retain** or **Snapshot**
  - **UpdatePolicy** – How to handle an update for ASG, ElastiCache, Domain or Lambda Alias
  - **UpdateReplacePolicy** – To retain or (in some cases) backup the existing physical instance of a resource when it is replaced during a stack update operation. **Delete**, **Retain** or **Snapshot**
  - **DependsOn** That resource is created only after the creation of the resource specified in the DependsOn attribute
- **Intrinsic Functions** allow you to assign properties that are not available during runtime most important two to know:
  - **Ref** returns the value of the specified parameter or resource
  - **Fn:GetAttr** returns the value of an attribute from a resource in the template
- aws cloudformation **create-stack –** CLI command to create a stack
- aws cloudformation **update-stack –** CLI command to update a stack
- **Serverless Application Model (SAM)** is an **extension** of CloudFormation that lets you define serverless applications