# IAM

## Securing Root Account

* Create users and grant permissions
* Create group and roles
* Control access to AWS resources
* Root account is the email address you used to sign up for AWS
* The root account has full administrative access to AWS
* **YOU SHOULD secure using multi-factor authentication**
* **Create an Admin Group for your administrator and assign the appropriate permissions to this group**
* **Create user accounts for your administrator**
* **Add your users to the admin group**

## IAM Policy Documents

* Written in JSON format
* It is written in key-pair
* Policies can be attached to Groups, Users, Roles

## Permanent IAM Credentials

* **IAM is universal**
* **YOU SHOULD NOT USE ROOT ACCOUNT FOR ANYTHING ELSE ACCEPT ADMINISTRATING ACCESS**
* **New Users by Default have no permissions**
* **Setup password rotation**
* Users should be grouped into Groups by job function (i.e., administrator, developers)
* Roles Internal usage that allows one resource to access another resource
* **Best practice is for users to inherit permissions from groups**
* **Users = Physical Person**
* **Access Key ID and secret keys are not the same as username and password**
* **Principle of Least Privilege: Assign minimum amount of access to do the job**
* **Amazon has prepopulated policy documents for certain job functions**
* **Account Settings can be used to manage password policies**
  + **Number of characters**
  + **Alphanumeric**
  + **Update every 90 days**
* **SAML 2.0 allows you to link AWS Users to Active Directory Federation Services**
* **INLINE POLICIES ARE FOR ON-OFF POLICY ATTACHMENTS**

# S3

## S3 Overview

* Simple Storage Service: Object Storage that allows for secure, durable and highly scalable object storage and simple to use
* Object based storage
* Cannot be used to run an operating system or database
* Unlimited Storage, up to 5TB file size
* Bucket is like a folder
* **Universal namespace**
* **You can assign URL to S3 Objects**
* **Uploading a file to an S3 bucket, remember you get a HTTP 200 code if the upload is successful**
* S3 is a key value store, key is the name, value is the data
* Supports versioning and metadata
* S3 is secure object storage platform that ensures availability and durability
* **Highly available (99.95%-99.99% service availability)**
* **Designed for durability (99.999999999% 9 d.p service durability)**
* Standard S3
  + Highly available (3 AZ’s)
  + Durable
  + Designed for Frequent Access
  + Suitable for most workload
* Offers lifecycle management: TTL for files
* You can set default encryption on a bucket, such that all new objects added to the bucket are encrypted
* Offers Access Control Lists to individual objects
* Can define Bucket Level Policies that can be attached to each bucket
* Offers Strong Read-after-write consistency
  + After a write or overwrite of a file, any request for a **read or list** of the file gets the new file **immediately**
* **S3 BUCKET URL’S CAN BE OF THE FORM:**
  + [**https://yamer.s3.us-west-1.amazonaws.com**](https://yamer.s3.us-west-1.amazonaws.com)
  + **https://s3.us-west-1.** **yamer.amazonaws.com**

## Secure S3 Buckets: ACL vs Bucket Policy

* **ACL: Object level control**
* **Bucket Policy: Bucket wide and ONLY TO FILES OWNED BY THE AWS ACCOUNT THAT CREATED THE FILE**
* Buckets are private by default: when you create an S3 Bucket, it is private by default (including all objects within it**). You must allow public access on BOTH the bucket and its objects in order to make the bucket public**
* Object ACL: You can individual objects public using object ACL’s
* Bucket Policies: You can make entire buckets public using bucket policies

## S3 Static Website Hosting

* You can host static websites using S3 such as .html sites. These require database connections and cannot be hosted on S3
* **S3 scales automatically so many enterprises just load their static websites on S3 when they think there is going to be a large number of requests**
* MUST MAKE BUCKET PUBLIC
* **Must enable Static Website Hosting Option**
  + **Must specify index page and error page**
* **Bucket objects can be made public individually or by Bucket Policy**

## S3 Object Versioning

* Can store multiple versions of files on S3, even if you delete a file
* Primarily for backup
* Can only enable versioning ONCE and cannot be disabled ONLY suspended
* **Lifecycle rules can be integrated into versioning**
* **It supports MFA to ensure that files are securely deleted**
* **Use list-version to see past version**
* **Each version has its own access and is NOT COVERED under the bucket policy. You need to manually make each version PUBLIC**
* **Deleting file does not Delete the previous versions. You can restore files by just deleting the Delete Marker**

## S3 Storage Classes

* S3 Standard
  + Highly Available (99.99%)
  + Highly Durable (11 9’s)
  + Multiple AZ’s >=3
  + Designed for Frequent Access
  + Suitable for most workloads
    - Static websites, content distributions, gaming
* S3 Standard-Infrequent Access
  + Highly Available (99.99%)
  + Highly Durable (11 9’s)
  + Rapid access but accessed less frequent
  + Per gigabyte access charge
  + Suitable for long term storage and backups and as a data store for disaster recovery
* S3 One Zone -Infrequent Access
  + **Highly Available (99.95%)**
  + Highly Durable (11 9’s)
  + Like above but only stored redundantly in one AZ (20% cheaper that above)
  + Suitable for non-critical data that needs to be accessed frequently
* S3 Intelligent Tiering
  + **Highly Available (99.9%)**
  + Highly Durable (11 9’s)
  + When you don’t know if you will access your data frequently or infrequently then use intelligent tiering
  + Moves your data based on usage to the most cost-effective tiers from above solutions
  + Uses AI to figure out what is better
  + $0.00025 monthly additional fee per 1000 objects
* Glacier Options
  + You pay each time you access your fata
  + Used only for archiving data
  + Cheap storage
  + Optimized for data that is VERY infrequently accessed
  + Option 1
    - Glacier Instant Retrieval: Provides long-term data archiving with instant retrieval time for your data
  + Option 2
    - Glacier Flexible Retrieval: Ideal storage class for archive data that does not require immediate access but should allow for extracting large sets of data at any time can be minutes to up to 12 hours
  + Option 3
    - Glacier Deep Archive: Cheapest Storage solution and primarily used to meet customer data retention policy. Can take up to 12 hours for standard retrieval and 48 hours for bulk retrieval

## S3 Lifecycle Management

* Lifecycle management automates moving your objects between different storage tiers, to save you money



* Combining lifecycle management with versioning can move different versions of the objects to different storage tiers.
* **You can only create life cycle policies to move FILES FROM HIGHER FREQUENCY CLASSES TO LOWER FREQUENCY CLASSES**
* Can do this by creating a lifecycle rule
  + Define lifecycle rule action
  + Based on days between access

## S3 Object Lock

* Used to access storage objects based on the WORM model (write once, read many)
* **Can be applied on individual object or on buckets as a whole**
* Help prevent objects from being deleted or modified
* Two Mode
  + **Governance Mode: Special permissions need to delete or modify files or its retention settings**
  + **Compliance Mode: Files or its retention setting cannot be deleted or modified its BY ANYONE**
* Retention Period: protects an object version from being deleted or modified for a fixed amount of time. After the retention period the object can be deleted or modified UNLESS there is also a LEGAL HOLD on the object
* Legal Hold is basically an object lock that does not have an expiring retention period but is a flag that you can add or remove
* **Glacier Vault Lock: a way of applying a WORM model to Glacier using a vault lock policy which can be setup so the policy can NOT be changed by ANYONE**

## S3 Encryption

* Types of Encryptions
  + Encryption in Transit
    - SSL/TLS
    - HTTPS
  + Encryption at Rest: Server-side Encryption
    - SSE-S3: S3-managed keys for encryption AES 256 bit
    - SSE-KMS: AWS Key management service-managed keys
    - SSE-Customer-provided keys
  + Encryption at Rest: Client-side Encryption
    - Have encrypted files on desktop
    - Console
    - Bucket Policy

## S3 Performance Optimization

* S3 prefixes are folders inside our buckets
* S3 has extremely low latency 100-200 ms
* **5500 GET/HEAD request per prefix and 3500 PUT/COPY/POST/DELETE requests per prefix**
* **HIGHER THE NUMBER OF PREFIXES THE BETTER THE PERFORMANCE**
* Limitation for KMS if you are using it to encrypt it
  + Uploading and downloading will count towards KMS quota
  + **Region specific with 5500, 10000, 30000 request per second options**
  + **Current you can’t EVEN increase a quota increase!!!**
  + **BEST TO AVOID USING KMS FOR ENCRYPTION**
* **Multipart Upload**
  + **Recommended for files larger than 100MB**
  + **REQUIRED FOR FILES OVER 5GB**
  + Helps Parallelize upload
* Multipart Download
  + Can download files by byte range: S3 Byte Range Fetches
  + If one part fails only specific part needs to be redownloaded

## S3 Replication

* Replicating object from one bucket to another
* **Version MUST BE enabled on both buckets separately for replication to happen**
* Objects in an existing bucket are not replicated automatically
  + Once replication is turned on, ONLY subsequent objects are updated and will be replicated automatically
* Delete markers are replicated by default
* Setup replication rules – Setup is under Lifecycle Management

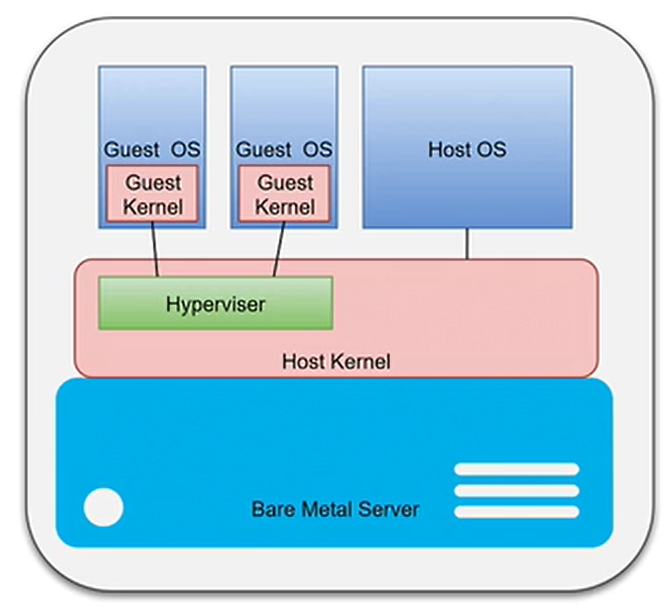
## CORS

* Cross Origin Resource Sharing defines a way for client web applications loaded in one domain to interact with resources in another domain.
* CORS allows for rich client side web applications to access cross-origin source of your Amazon S3 resources

# EC2: Elastic Compute Cloud

## EC2 Overview

* Secure, resizable, compute capacity in the cloud
* Fundamentally a VM that is just hosted on AWS instead of your own data center
* Designed to make web-scale cloud computing easier for developers
* Problems with on-premise infrastructure
  + Long term investments: 3 -5 Years minimum
  + Months to setup
* **Types of EC2 Instances:**
  + **On-Demand: Pay by the hour or the second, depending on the type of instance you run**
  + **Reserved: Reserved for 1 – 3 years. Up to 72% cheaper on the hourly charge**
  + **Spot: Purchase unused capacity at up to 90% discount. Price fluctuates with supply and demand**
  + **Dedicated: A physical EC2 server dedicated to your personal use. Most expensive option**
* On Demand
  + Use Case: Low cost, flexible usage. Building a test website, spiky, short term unpredictable use. Testing the water
* Reserved
  + Use Case: Predictable Usage, Specific Capacity Requirements, Pay up front. Only Single **Region**
    - Standard RI’s: Up to 72% off the on-demand prices. Unused can be sold in Marketplace
    - Convertible RI’s: 54% off the on-demand price. Option to change to other instance types. Cannot be sold
    - Scheduled RI’s: Launch within the time window you define
  + Saving Plans available for all AWS compute usage, regardless of instance type or Region
    - Commit to 1- or 3-years period
    - Super Flexible and include other serverless architecture (Lambda and Fargate)
* Spot
  + Use Case: Applications that have flexible start and finish times. Applications that are only feasible at very low compute prices. Users with an urgent need for large amounts of additional computing capacity
  + Image Rendering. Genomic Sequencing. Algorithmic Trade Engines
* Dedicated Hosts
  + Use Cases: Compliance Regulatory requirements. Licensing conditions
    - Can be purchased for hourly use
    - Can be purchased as a reservation for up to 70% off the on-demand dedicated host price
    - One physical server for the entire instance
  + Instances
    - Hardware is only assigned per customer
    - But several dedicated instances may run on the same bare metal server
* Capacity Reservations
  + Can be purchased at any time without entering into a contract, with capacity available immediately
  + Need to cancel capacity reservation to stop incurring charges
  + Only in One AZ, Instance type, tenancy and platform OS
* **AWS Pricing Calculator can be used to identify the cost of moving to the cloud**



## EC2 Setup

* Starts with defining name for EC2 Instance
* Select an AMI (Amazon Machine Image)
* Then choose Instance Type (i.e., t2.micro)
* Configure instances to add to a certain VPC, subnet group (1 subnet = 1 AZ)
* Configure Storage: EBS volume attached for each EC2 Instance
* Add Tags: To bucket EC2 instances by department or by user
* Configure Security Group: You can define new security group or attach existing security group
  + Define Port Range, Source IP address and Protocol
* Add Security Key Pair: You can define new key-pair or attach existing one
* Connect via SSH or EC2 Instant Connect

## AWS Command Line

* You can access AWS through command line
* To connect into EC2 you need to change permissions for key file
* Need to get IPv4 Public IP to SSH in
* For user configuration on CLI you either need to login using SSO or use access key ID and secret access key for a user added to a group with policy attached that allows access to the resource
* Secret Access Key is only generated ONCE

## IAM Role

* Role is an identity you can create in IAM that has specific permissions. A role is like a user as it is an AWS Identity with permission policies that determine what the identity can and cannot do in AWS
* **IAM role can be assumed by anyone who needs it**
* Roles are temporary and provide temporary security credentials for your role session
* Roles can be assumed by people, AWS Architecture and other system level accounts
* **Roles can provide cross-account access**
* **Roles allow you to avoid Hard-Coding Credentials**
* **A ROLE CAN ONLY DEFINE ACCESS TO WHICH API ACTIONS CAN BE MADE ON RD INSTANCE**
* **When assigning STS Federated User Credentials are dependent on the users access to an AWS service**

## Bootstrap Scripts (User Data)

* Communication Channels
  + Linux: SSH – port 22
  + Windows: RDP -port 3389
  + HTTP: Web Browsing – port 80
  + HTTPS: Encrypted Web Browsing – port 443
* Security Groups
  + Like virtual firewalls
  + Inbound: By default, everything blocked
  + Outbound: By default, everything allowed
  + To let everything in 0.0.0.0/0
  + Can have multiple security groups attached to EC2 groups
* Bootstrap Scripts
  + Scripts that run when the instance first starts to run

## Meta Data

* Curl from within the EC2 instance to the following site
* curl <http://169.254.169.254/latest/meta-data/local-ipv4>

Text

Description automatically generated

* Get a bunch of metadata about our instances
* Can be used with user-data to automate getting meta-data about the instances

## EC2 Networking

* Three types of virtual networking cards:
  + ENI – Elastic Network Interface
  + Enhanced Networking – Uses single root I/O virtualization to provide high performance
  + Elastic Fabric Adapter – Accelerates High Performance Computing and Machine Learning Applications
* **ENI**
  + **A primary private IPv4 address from the IPv4 address range of your VPC**
  + **One or more secondary private IPv4 addresses from the IPv4 address range of your VPC**
  + **One Elastic IP address (IPv4) per private IPv4 address**
  + **One public IPv4 address**
  + **One or more IPv6 addresses**
  + **One or more security groups**
  + **A MAC address**
  + 1 or more Security Groups
  + Use Cases:
    - Management Network
    - Use for network and security appliances in your VPC
    - Create dual-home instances to split workloads on distinct subnets
    - Useful for creating a low budget, high-availability solution
* Enhanced Networking
  + Low CPU Utilization
  + Provides higher bandwidth, higher packer per second performance and consistently lower inter-instance latency
  + Two ways to use these:
    - Elastic Networking Adapter (ENA): Supports network speeds up to 100 Gbps and uses single root I/O virtualization (SR-IOV) to provide high-performance networking capabilities
    - Intel 82599 Virtual Function Interface: Supports network speeds up to 10 Gbps
    - **ALWAYS USE Elastic Networking Adapter**
  + **Only supported by some instances**
* EFA
  + Provides lower and more consistent latency and higher throughput than TCP transport traditionally used in cloud based HPC systems
  + EFA OS by-pass lets you bypass operating system kernel and directly communicate with EFA device. ONLY IN LINUX

## EC2 Placement Groups

* Three types
  + Cluster Placement Groups:
    - **Grouping of instances within a single AZ**
    - For applications that need low network latency and high network throughput or both
    - Only works with certain instance types
  + Spread Placement Groups:
    - **Can be in multiple AZ’s**
    - Instances based on different hardware
    - Each **instance** is placed on its own rack
    - For application where small number of **individual** critical instances that must be kept separate from each other
    - Not supported for dedicated instances or hosts
  + Partition Placement Groups:
    - **Can be in multiple AZ’s**
    - Each **partition** **(multiple instances)** is placed on its own rack
    - Each partition has its own power and network
    - For application where you want **multiple** instances to have different power sources and networking
* AWS Suggests keeping the instances in each placement group homogenous
* You can’t merge placement groups
* You can only move and remove instances from placement group using CLI or SDK

## Timed Workload using Spot Instances

* Spot Instances are great for Stateless, fault-tolerant or flexible applications
* **Application like Big-data, containerized workloads, CI/CD, high-performance computing (HPC), or other test and development workloads**
* To use spot instances
  + Define maximum spot prices
  + Instance will be provisioned so long as the Spot price is BELOW your maximum Spot Price
  + Hourly Spot prices vary
  + You have 2 mins to decide whether to terminate or stop your instance
* Spot blocks allow you to stop your Spot Instances from being terminated even if the Spot price goes over your max Spot Instances. You can set Spot blocks for between 1 to 6 hours currently
* DO NOT USE SPOT INSTANCE FOR:
  + CRITICAL JOBS
  + DATABASES
  + PERSITENT WORKLOADS
* Terminating Spot Instances
  + Spot instances can be setup as
    - One-time: Only starts once and then ends when price is breached
    - Persistent: must define window for when this is valid. Keeps checking Open, Active, Disabled. If you don’t disable your persistent request, AWS will keep provisioning new instances!!!
    - MUST CANCEL THE PERSISTENT REQUEST TO STOP ABOVE FROM HAPPENING
* Spot Fleets is a collection of Spot Instances and (optionally) On-Demand Instances
  + It attempts to launch the target number of Spot Instances and (optionally) On-Demand Instances asked for in your request for the price constraint you have defined

## AWS with VMware Cloud on AWS

* Use for Hybrid Cloud use cases
* Useful for Cloud Migration
* Disaster Recovery environment
* Use AWS services in a hybrid cloud infrastructure
* How is it deployed?
  + It runs a dedicated hardware hosted in AWS using a single AWS account
  + Each host has two sockets with 18 cores per socket, 512 Gb RAM and 15.2 TB raw SSD storage
  + Each host is capable for running multiple VPWare Instances (hundreds)
  + Clusters can start with two hosts up to a maximum or 16 per cluster: Each cluster is a vCenter

## AWS Outposts -Brings Entire Data-Center to You

* Outpost brings the AWS data center directly to you on-premises
* Allows you to have a large number of AWS Service with you at your own datacenter
* 1U -42U servers
* Benefits
  + Hybrid Cloud
  + Fully manager Infrastructure
  + Consistency providing SDK
* Outpost Family Members
  + Outpost Rack
    - Starting with single 42U Rack and scales up to 96 racks
    - Provides AWS compute, storage, database, and other services locally
    - Gives the same AWS infrastructure, services, and APIs in your own data center
  + Outpost Sever
    - Individual servers in 1U or 2U form factor
    - Useful from small space requirements, retails stores, pharmacies
    - Provides local compute and networking services

## AWS Wavelength

* AWS Infrastructure to build low latency and high bandwidth applications over 5G networks
* Connect to AWS Services over 5G access via Wavelength Zones
* Contains most of the AWS Backbone
* A wavelength zone can be deployed within an VPC and can include a VPC subnet
* Access carrier network using the Carrier Gateway (similar to NAT Gateways)
* Primarily for gaming
* Use Cases
  + ML-assisted Diagnostics for Healthcare
  + Connected Vehicles
  + Smart Factories
  + Real-time Gaming
  + AR/VR/XR
  + Interactive Live Video-Streams

# EBS: Elastic Block Storage

## Overview

* EBS stands for elastic block storage
* Storage volumes you can attach to your EC2 instances
* Use them the same way as you would use any system disk
  + Create a file system
  + Run a database
  + Run an operating system
  + Store data
  + Install application
* Built for Mission Critical Data
  + Production Workloads: Designed for mission-critical workloads
  + Highly Available: automatically replicated within a single Availability Zone to protect against hardware failure
  + Scalable: Dynamically increase capacity and change the volume type with no downtime or performance impact to your live systems NO DOWNTIME
* EBS Volume Types:
* SSD Drives
  + General Purpose SSD (gp2)
    - 3 IOPS per GB, up to a maximum of 16,000 IOPS per volume
    - gp2 volumes smaller than 1 TB can burst up to 3,000 IOPS
    - **Good for boot volumes or development and test application that are not latency sensitive**
  + General Purpose SSD (gp3)
    - Predictable 3000 IOP’s baseline performance and 125MB/s regardless of the volume sizes
    - **Ideal for applications that require high performance at a low cost such as** **MySQL, Cassandra, virtual desktops, and Hadoop Analytics**
    - Customers looking for higher performance can scale up to 16,000 IOPS and 1000 MB/s for an additional fee
    - Gp3 is 4 times faster than max throughput of gp2 volumes
  + Provisioned IOPS SSD (io1)
    - Up to 64000 IOP’s per volume 50 IOP’s per GB
    - **If you need more than 16,000 IOP’s**
    - It’s required for IO intensive applications
  + Provisioned IOPS SSD (io2)
    - Latest generation
    - Higher durability and more IOPS
    - **It is useful for I/O intensive apps, large databases and latency-sensitive workloads. Applications that require high levels of durability**
* Hard Disk Drives
  + Throughput Optimized HDD’s (st1)
    - Low-Cost HD volume
    - Baseline throughput of 40 MB/s
    - Ability to burst up to 250 MB/s per TB
    - Maximum throughput of 500 MB/s per TB
    - Frequently accessed, throughput-intensive workloads
    - **Big Data, Data Warehouses, ETL, and log processing**
    - Cost-effective way to store mountains of data
    - **Cannot be a boot volume**
  + Cold HDD (sc1)
    - Baseline throughput of 12 MB/s
    - Ability to burst up to 80 MB/s per TB
    - Maximum throughput of 250 MB/s per TB
    - **Colder data requiring fewer scans per day**
    - **Good for applications that need the lowest cost where performance is not a factor**
* IOP’s vs Throughput
* IOP’s
  + Measures the number of reads and write operations per second
  + Important metric for quick transactions, low latency apps, transactional workloads
  + The ability to action reads and writes very quickly
  + Choose provisioned IOP’s SSD (io1 or io2)
  + High transactions OLTP systems
* Throughput
  + Measures the number of bits read or written per second
  + Important metric for large datasets, large I/O sizes, complex queries
  + Need the ability to deal with large datasets
  + Choose Throughput Optimized HDD (st1)
  + ETL’s, big data

## EBS Volumes and Snapshots

* Volumes exist on EBS Hard Disks i.e., virtual hard disks. You need AT MINIMUM 1 volume per EC2 Instance. This is called the root volume devices
* Snapshots are a photograph of the virtual disk on S3
* Point in time copy
* It is incremental in nature. Only changes are replicated
* First snapshot takes the longest to copy
* 3 Tips
  + **Snapshots only capture data that has been written to your EBS volume, which might exclude any cached data by your application. Best is to stop you instance and then take snapshot**
  + If you take snapshot of encrypted EBS volume the snapshot is also automatically encrypted
  + **You can share snapshots REGIONALLY, if you want to share cross region, you need to copy them to that specific region**
* **EBS volumes are always in the same AZ as the EC2 instance**
* You can resize the EBS volumes on the fly **HOWEVER you will have to extend the filesystem in the OS, so the OS sees the change**
* **You can switch volume types on the fly (remove or edit)**
* **You can create images from snapshot**
* **Snapshots can now be restored using Archive snapshots or Recycle Bin snapshot**
* **You can take snapshots on running instances**
* **Lazy Loading: Volumes created from EBS snapshot can be accessed immediately BUT the copying process happens only for files that are being accessed first, remaining are copied after that on schedule**
* **Support cross region copy**
* **RAID 0:** 
  + **Single copy split over multiple volumes**
  + **I/O is more important than fault tolerance. DO NOT USE for boot volumes**
  + **LOSS OF DATA IF volume fails**
* **RAID 1:** 
  + **Multiple copies over separate volumes**
  + **Fault tolerance is more important than I/O**
  + **Safer from standpoint of data durability**

## EBS Encryption

* Encryption happens using AES-256 algorithm that is provisioned using AWS KMS customer master keys when creating encrypted volumes and snapshot
* **Data at rest, in transit and in snapshot are encrypted. All volumes created from snapshots are also encrypted**
* Encryption happens transparently (you don’t have to do anything)
* Minimal latency impact
* You can copy an unencrypted snapshot and then encrypt it
* Snapshots of encrypted volumes are encrypted as well
* Root device volumes can be encrypted on creation
  + **Create a snapshot**
  + **Create a copy of the snapshot and select the encrypt option**
  + **Create an AMI from the encrypted snapshot**
  + **Use that AMI to launch new instances**

## EC2 Hibernation

* If we **stop (not terminate)** EC2 instances, the data is kept on disk and data will remain on disk until the EC2 instance is restarted
* When we start EC2 instance
  + Operating System boots up
  + User data script runs
  + Application Starts
* When you hibernation,
  + hibernation saves the contents of RAM to disk (EBS Volume)
  + Both EBS root and attached volume data is persisted to disk
  + Hardware may change but instance ID will remain the same
* When you restart
  + EBS volumes are restored to previous state
  + RAM content is reloaded
  + The processes that were previously running on the instance are resumed
  + Previously attached data volumes are reattached, and the instance retains its Instance ID
  + EC2 hibernate systems boot up A LOT FASTER
  + **RAM must be less than 150 GB to hibernate**
  + Hibernation time is limited to 60 days
* Only certain instance types and AMI are supported
* Only GP2 and IOPS SSD are supported
* **MUST ENABLE HIBERNATION AT LAUNCH AND YOU CANNOT DISABLE IT UNLESS YOU TERMINATE THE INSTANCE**
* **MUST ENCRYPT ROOT VOLUME TO TURN ON HIBERNATE**
* Cannot hibernate instances in Autoscaling groups or used by Amazon ECS
* Hibernated instances cannot be updated until started
* IPv4 public IP’s change when EC2 instances are restarted. Keeps private IPv4, IPv6 and public IPv6 intact

# EFS: Elastic File System

* Managed network filesystem that can be mounted on EC2 instances in multiple AZ’s
* Highly available and scalable and EXPENSIVE
* **EFS IS VPC SPECIFIC**
* **Great for content management systems where content is regularly share. i.e., Web Servers**
* **Great for situations where you need highly scalable share storage using NFS**
* **NFSv4 protocol on port 2049**
* Compatible with only Linux based AMI
* Encryption using KMS
* Filesystem is elastically scalable
* **1000’s of concurrent connects**
* **10Gbps transfer rate**
* **Scales to Petabyte level**
* **Read after write consistency**
* Two Performance levels
  + General Purpose: webservers, CMS etc.
  + Max I/O: Big data processing
* Two different storage tiers
  + Standard: For frequently accessed files
  + Infrequently Accessed
  + Can define lifecycle management rule to move file between the two
* EFS using One-Zone Storage Classes
  + Data is stored redundantly in **single AZ**
  + Can connect to EC2 instances from multiple AZ
  + Useful for Big Data and analytics and media processing
* EFS using Regional Storage Classes
  + Data is stored redundantly in **multiple AZ**
  + Can connect to EC2 instances from multiple AZ
  + Useful for Big Data and analytics and media processing
* Setup
  + **Define VPC where it will live**
  + Can chose Regional or only One AZ
  + Can define fixed (provisioned) or scalable (bursting) throughput level
* Encryption during transit can be enabled when mounting the filesystem using the Amazon EFS mount helper. It uses TLS version 1.2

## FSx Overview

* Amazon FSx for Windows File Server
  + Provides a fully managed native Microsoft Windows file system so you can easily move your windows-based Linux applications that require file storage to AWS
  + Runs SMB based file services
  + Is designed for Windows
  + Supports Active Directory users, ACL, group, security policies
* EFS
  + A managed NAS filer based on NFSv4
  + Linux oriented
* Amazon FSx for Lustre
  + Fully managed file system that is optimized for compute-intensive workloads
    - High Performance Computing
    - Machine Learning
    - Media Data Processing Workflows
    - Useful for situations where you need to run a Lustre file system that needs to process massive datasets at up to hundreds of GB per second of throughput with millions of IOPS and sub millisecond latencies
    - Can store data directly on S3

## Amazon Machine Images

* Amazon Machine Images provide the information required to launch an instance
* You must specify and AMI when you launch an instance
* Can launch multiple instances with the same AMI’s
* 5 Things That Define an AMI
  + Region
  + OS
  + Architecture (32bit, 64-bit, ARM)
  + Launch permissions
  + Storage for the root device (root device volume)
* Two Types
  + Amazon EBS: The root device for an instance is launched from an EBS volume created from an EBS snapshot
  + Instance Store: The root device for an instance is launched from an instant store volume created from a template stored in S3
* Instant Store Volume
  + **Ephemeral: EC2 Instances CANNOT BE STOPPED. You will lose that data**
  + Also, if you terminate the instance the volume is also deleted
* EBS backed instances
  + **Persistent: EC2 Instances CAN BE STOPPED. You will not lose your data**
  + You have the **option** to keep the EBS root volume when the EC2 Instance is terminated
* **REBOOT WILL NOT DELETE YOUR DATA. ONLY TERMINATION**
* **You can specify launch permissions for which AWS accounts can launch a specific AMI-backed instance**
* **Sharing AMI’s does not change the ownership of the account, owning account is charged for storage in the Region**
* **Copying a shared AMI will then give ownership of the copied AMI to the shared account**

## AWS Backup - Consolidate all your backups across multiple AWS services – Standby DB NOT REQUIRED FOR RECOVERY – CAN BE USED WITH AWS Organizations – Cross Region Backup

* Consolidate all your backups across multiple AWS services (EC2, EBS, EFS, FSx for Windows and Lustre, and AWS Storage - Volume Gateway, VM’s)
* Can include other database technologies like RDS, Aurora, Neptune and DynamoDB, DocumentDB
* Backup can be used with AWS Organizations to back up multiple AWS accounts in your organizations
* It gives you centralized control across all AWS Services in multiple AWS accounts across the entire AWS organization
* Benefit
  + Central Management: Centralize backup across multiple AWS services across multiple AWS accounts
  + Automation: Create **BACKUP SCHEDULES, RETENTION POLICIES, LIFECYCLE POLICIES**
  + **Improved Compliance: BACKUP POLICIES can be enforced while backups can be encrypted at rest and in transit. Auditing made easy. Regulation compliance made easy**
* Allows you to copy data across multiple region without maintaining multiple scripts (cross region

# Databases

## RDS – Relational Database Services – Basic RDMS

* Database Types:
  + SQL Server, Postgres, MySQL, MariaDB, Aurora
* For OLTP – Small Transactions
* Multi AZ Available – But Optional (Except Aurora) – **For Disaster Recovery Only**
* **Upon failure multi-AZ changes the canonical name of primary DB to secondary DB**
* Automated Backups
* Running 3 EC2 Instances on backend with Load Balancer
* Has unique DNS Endpoint
* Free tier does not have multi-AZ available
* You can create snapshots of RDS instances via EC2

## RDS – Read Replicas

* Used to improve performance of RDS
* Exact read-only copy of data on RDS **– For Performance Only**
* Read-replica can be cross-AZ and cross-region
* Used for Performance
* Has its own DNS endpoint
* Works exactly like RDS
* Can be promoted to primary RDS (giving it read and write capability)
* Promotion can be for running massive queries
* Multiple read replicas are possible (5 for each instance)

## Amazon Aurora – High Performance RDMS

* Proprietary database – MySQL PostgreSQL
* 5 times better than MySQL and 3 times better than PostgreSQL
* Starts with 10 GB, Auto scales 10 GB at a time to 128 TB
* Compute Resource scale up to 96 vCPU and 768 GB Memory
* **2 copies of your data contained in each availability zone with 3 minimum availability zones i.e., 6 copies**
* You can lose 2 copies of data without losing write ability and 3 copies without affecting read ability
* It is self-healing, errors fixed automatically
* Can define failover secondary region
* **Can have read replicas:**
  + **Aurora Replicas: 15 read replicas, automated failover, asynchronous replication type(ms)**
  + **MySQL: 5 read replicas, no automated Failover, asynchronous replication type(s)**
  + **PostgreSQL: 5 read replicas**
* **Automated Backup with no performance impact**
* **Take snapshot with no performance impact and can share the snapshot across multiple accounts**
* **Aurora Serverless:** On demand, autoscaling configuration, shuts down and starts up whenever required. i.e., when you want performance of Aurora but have spiky workloads
  + **For infrequent relatively simple workloads**

## DynamoDB - High Performance NoSQL

* Propriety NoSQL database by amazon
* 0-9 millisecond latency requirements
* Fully managed – Document and key-value data models
* **Has secondary indexing capability**
* **Stored on SSD**
* **Spread across 3 data centers by default**
* **Eventual consistent reads**
* **Can have strongly consistent reads**
* DynamoDB Accelerator:
  + Fully managed, highly available, in memory cache
  + 10x performance improvements
  + Reduces request time from milliseconds to **microseconds** – even under load
  + No development requirements for caching
  + Works with DynamoDB API calls
  + Traditional Cache
    - 1) Queries Cache first
    - 2) if not available queries DB
    - 3) Stores in cache for TTL
  + DynamoDB Accelerator:
    - 1) Queries DAX first
    - 2) if not available DAX queries DB and stores it and passes it back
* Pay per request pricing
* No capacity requirements
* Pay more per request than if you had provisioned capacity
* **Use for new product launches. Uncertain about scale**
* Encryption at rest using KMS
* Site -to-site VPN
* Direct Connect (DX)
* IAM policies and roles
* Fine grained access
* Integrates with Cloud Watch and Cloud Trail
* Lives outside VPC: needs VPC Endpoints

## DynamoDB - Transactions – ACID Compliance

* **ACID compliant within one or more tables within single AWS account**
* Can perform multiple inserts, updates, deletes as part of complicated transactions
* Use cases:
  + Processing financial transactions
  + Fulfilling and managing orders
  + Building multiplayer game engines
  + Coordinating actions across distributed components and services
  + Provides: Eventual, strong, and transactional consistency
  + 2 optional for writes: Standard or transactional
  + Up to 25 items or 4 MB of data

## DynamoDB - Backups

* Can do full back up at any time without any impact
* Consistent within seconds
* **Backup in same region as source table**
* **Point in time recovery – Time Travel feature**
  + **Can be restored to anytime in the last 35 days**
  + **Not enabled by default**
  + **Smallest timeframe is 5 mins**

## DynamoDB - Streams and Global Tables – Global Replication

* Streams Records – Time ordered sequence of item-level changes in a table
* Sequences operations are broken up based on time of arrival into shards

A picture containing diagram

Description automatically generated

* Combine with AWS Lambda to get stored procedure functionality
* **Global Tables: Managed Multi-Master, Multi-Region Replication**
* **Works based on Dynamo DB stream, need to have it switched on to have it work**
* **Multi-region redundancy for disaster recovery or high availability**
* **Replication latency is 1 seconds**

## Amazon Document DB - MongoDB-Compatible Database – Open Source NoSQL

* MongoDB is a document database that allows for scalability and flexibility with your data as well as robust querying and indexing features
* Amazon Document DB allows you to run MongoDB on the AWS cloud.
* Scales with your workload
* Safely stores your database information
* Using AWS, you can get rid of operational overheads like managing backup
* Moving MongoDB on-premises can be moved to AWS using AWS Database Migration and using Document DB as the end source

## Amazon Keyspaces - Apache Cassandra database service – Big Data NoSQL

* Apache Cassandra:A distributed database that uses NoSQL that is primarily used for big data solutions
* Amazon Keyspaces: This is an Amazon managed Apache Cassandra database service where you don’t need to manage servers, software and patching and
* It is serverless

## Amazon Neptune - Graph Database

* Graph databases is a node/edge instead of tables and documents
* Build connections between identities and identity resolution solutions such as social graphs and accelerate updates for ad targeting, personalization, and analytics
* Helps query highly connected datasets
* Build graph queries for near real time identity fraud pattern detection in financial purchases

## Amazon Quantum Ledger Database – Cryptocurrency NoSQL

* Ledger Database: It’s a NoSQL database that is immutable, transparent, and has cryptographically verifiable transaction logs that is owned by one authority
* Records are not updatable
* For cryptocurrency
* Used for shipping tracking and supply chain
* Maintaining claims history

## Amazon Timestream – Time Series Data Analysis

* Data points that are logged over a series of time, allowing you to track your data
  + IoT sensor
  + Analytics
  + Devops Application: applications that change based on users’ updates
  + Handle trillions of events per day up to 1000 times faster and as little as 1/10th the cost of traditional relational

# VPC

## VPC Overview

* Virtual data centers in the cloud
* EC2 is deployed into default VPC
* Logical isolated part of AWS
* 1 VPC subnet can only exist in 1 Availability Zone
* Fully customizable network with multiple layers of security
* Three Tier Architecture
  + Web: Public Facing, Port 80
  + Application: Private subnet, can only speak to database tier
  + Database: Private subnet, can only speak to application tier
* VPN – Can create hardware VPN to extend corporate network
* Steps
  + **Select Region**
  + Select IP Address Range i.e., 10.0.0.0/16
    - **Can extend IP CIDR Range by adding secondary CIDR Range**
  + This creates Network ACL, Route Table, Router
    - ROUTE TABLE AUTOMATICALLY HAVE ROUTES TO ALL SUBNETS IN VPC
  + Create a Public subnet with security groups
  + Attach an internet Gateway
  + Create Private Subnet with security groups
  + For organizations communication create a Virtual Private Gateway
* What can we do inside VPC’s?
  + Launch instances into subnet
  + Create custom IP address for each subnet (10.0.0.0/28-10.0.0.0/16)
  + Configure route tables between subnets
  + Create Internet Gateways
  + Create access control lists
* You can use network ACLs to block specific IP Addresses
* Default VPC vs Custom VPC
  + Default VPC
    - All subnets have route out to the internet
    - Each EC2 instance has both private and public IP address
  + Custom VPC
    - Fully customizable
* When creating subnets AWS reserves a total of 5 IP Address
* You can only have one internet gateway per VPC
* To access the internet, you need to specify a route table
* It is important to create a new route table and not use the default route table because if you give the main route table access to the internet then any new subnets created will by default give access to the internet

## NAT Gateways – Internet Access for Private Subnets

* NAT Gateways Redundant EC2 instances inside AZ (i.e., DON’T NEED IT)
* Starts at 5Gbps that scales up to 45 Gbps
* **No need to patch**
* **Not associated with security groups**
* **Automatically assigned a public IP address**
* You need to put the NAT Gateway in the public subnet
* Change the route table to access the internet
* When facing network issues check in following order
  + Route Table
  + Network ACL
  + Security Group

Graphical user interface

Description automatically generated

## Security Groups – Control EC2 communications within a Subnet

* Security Groups are **STATEFUL** –
  + Allowed Outbound requests are allowed to flow-in in response regardless of inbound rules
  + Allowed Inbound requests are allowed to flow-out in response regardless of outbound rules
  + i.e., if you block internet access by blocking port 80 inbound but allow outbound connection on port 80 **it won’t work**
  + **Use NETWORK ACL: they are STATELESS**
  + **NAT GATEWAYS ARE AZ DEPENDENT. DO NOT USE SAME NAT GATEWAY FOR SUBNETS IN DIFFERENT AVAILABILITY ZONE. IF ONE NAT GATEWAY GOES ALL SUBNETS WILL LOSE NETWORK ACCESS**

## Network ACLs - Controlling Subnet Traffic and Block Specific External IP Addresses

* Network ACLs are the first line of defense
* VPC comes with default ACL that **ALLOWS** all inbound and outbound traffic
* Custom ACL the **DENIES** all inbound and outbound traffic
* Each subnet in your VPC must be associated with a network ACL
* If you don’t associate network ACL with a subnet, the default network ACL is adopted
* **BLOCK IP ADDRESSES USING NETWORK ACL’s and NOT SECURITY GROUPS**
* **USE SECURITY GROUPS TO CONTROL EC2 COMMUNICATION**
* ONE network ACL per SUBNET, MANY SUBNETS per network ACL
* **Rules are evaluated in order of Rule Number**
* Network ACL’s have separate inbound and outbound rules and can either allow or deny traffic
* **NETWORK ACL: they are STATELESS**
  + Allowed Outbound requests are DENIED to flow-in in response UNLESS inbound rules specified
  + Allowed Inbound requests are DENIED to flow-in in response UNLESS outbound rules specified
* Network ACL’s rules are in terms of 100
* **DEFINE BOTH INBOUND AND OUTBOUND RULES**
* Ephemeral Ports are only open for the session

## VPC Endpoints - Private Communication with other AWS Services

* VPC Endpoints enable you to privately connect your VPC to supported AWS Services and VPC endpoint services powered by PrivateLink without requiring an Internet Gateway, NAT devices, VPN connection, or AWS Direct Connect connection
* These are virtual devices that scale horizontally allow internal communication between VPC instances without imposing availability risk or bandwidth issue.
* Use VPC endpoint instead of NAT Gateway for writing files in AWS services
* Two types of endpoints
  + Interface Endpoint: elastic, with private IP Address as entry point
  + Gateway Endpoint: similar to NAT Gateway – virtual device to connect to S3 and DynamoDB via a Route Table
* VPC Endpoint policies control what level of access to the AWS service is possible

## Bastion Host - Securely access to a private subnet from a public network

* A bastion host is used to provide secure access to a private subnet from a public network

## VPC Peering – Inter Regional and Account VPC communication

* May require multiple VPC

Diagram

Description automatically generated

* VPC peering allows one VPC to talk to another using a direct network route using private IP addresses
* Instances behave as if they were on the same private network
* **You can peer VPC’s with other AWS accounts as well as with other VPC’s in the same account**
* You can peer between regions
* Peering in star configuration (You **CANNOT** transitively peer from one VPC to another, you need to create a separate peer for every two VPC)
* **You can’t have overlapping CIDR ranges for your VPC’s.** If they overlap the VPC’s will not be able to communicate

## AWS PrivateLink – Connect Service VPC to Customer VPC Simplified

* To open your VPC to other VPC, you can open your VPC up to the internet OR use VPC Peering
* You will have to create multiple VPC peers for each VPC combinations.
* Your whole VPC network will be accessible
* **Best way to expose a service VPC to thousands of customers VPC’s is using Private Endpoints**
* **Doesn’t require VPC peering, no route table, NAT Gateway, Internet Gateway**
* **Requires** 
  + **Service VPC - VPC Endpoint Service – Requires Network Load Balancer**
  + **Customer VPC – VPC Endpoint - Requires Elastic Network Interface**

## VPN Cloud Hub – Communication between Multiple VPN Site

* If you have multiple sites each with its own VPN connection, you can user AWS VPN CloudHub to connect those sites together
* Hub Spoke Model
* Steps:
  + Create Virtual Private Gateway
  + Create customer gateway for each site
  + Assign each gateway a unique Border Gateway Protocol Autonomous System Number
* Low Cost and Easy to Manage
* It operates over the public internet, but all traffic between customer gateway and AWS VPN is encrypted

## Direct Connect – Dedicated Line Connection for REGIONAL Access to multiple VPC’s

* Direct Connect is a cloud service solution that makes it easy to establish a dedicated network connection from premise to AWS.
* **Can reduce cost, increase bandwidth throughput, and provide a more consistent network experience that internet-based connections**
  + Two Type
    - Dedicated Connection: Physical Ethernet Connection via AWS
    - Hosted Connection: Physical Ethernet Connection via a Partner (Verizon)

Graphical user interface

Description automatically generated

## Transit Gateway – Reduce Complexity of connecting multiple VPC

* Connects VPC’s and on-premises networks through a central hub. Reduce complicated VPC networks. Each connection only needs to be made **ONCE**

Diagram

Description automatically generated

Diagram

Description automatically generated

* Allows you to have transitive peering between thousands of VPC’s and on-premises peering
* Works on spoke and hub model
* Works of a regional basis BUT you can have it across multiple regions
* You can use it across multiple AWS accounts using RAM
* Supports route tables to limit VPC communications
* Works with Direct Connect
* Support IP multicast (multi publish)

## VPC Flow Logs – Capture IP Traffic to and from network interfaces in your VPC

* VPC Flow logs capture IP traffic going to and from network interfaces in your VPC
* VPC Flow logs are stored in Amazon CloudWatch logs

# Route53

* DNS: Convert domain name to IP addresses
* IPV4: addresses are running out. 32 bit
* IPV6: 128 bit. Will be universally moving to this
* Top Level Domain -.com/. us/.edu
* Second Level Domain .co./
* Top Level Domain is assigned by IANA
* Domain registrar – assigns domain names to top level domain. i.e., domainname.com GoDaddy.com
* Logs can be written to AWS Cloud Trail and Cloud Watch
* DNS record type
  + SOA Record store:
  + Name of the server that supplied the data for the zone
    - The administrator of the zone
    - Current version of the data file
  + NS records:
    - Used by top level domains to direct traffic to content DNS servers

A picture containing logo

Description automatically generated

* + A record is fundamental type of DNS record
  + A record is used by a computer to translate a name of the domain to an IP address
  + TTL is time to live: length a DNS record is cached
  + CName: canonical name – mapping one domain name to another
  + **Alias records – Map resource record within AWS to load balancers in hosted zone. ONLY IN AWS. Can select S3, ELB IP, etc.**
  + **You cannot have CNAME for naked domain names. i.e., only for m.acloudguru.com not acloudguru.com**
  + You cannot have Alias Record for naked domain names. i.e., only for m.acloudguru.com not acloudguru.com

## Simple Routing Policy

* With simple routing policy you can have only one record with multiple IP Addresses
* If you specify multiple values in the values in a random order
* You can specify TTL for DNS caching. Default is 5 mins

## Weighted Routing Policy

* Weighted routing policy you can split traffic based on different weights for routing to each IP Address
* You can set health checks on individual record sets
* If a record fails the health check, it will be removed from Route **53 until it passes the health check**
* You can set up an SNS notification alert when the health check fails
* **When creating weighted routing, split the IP addresses to multiple IP addresses**

## Failover Routing Policy

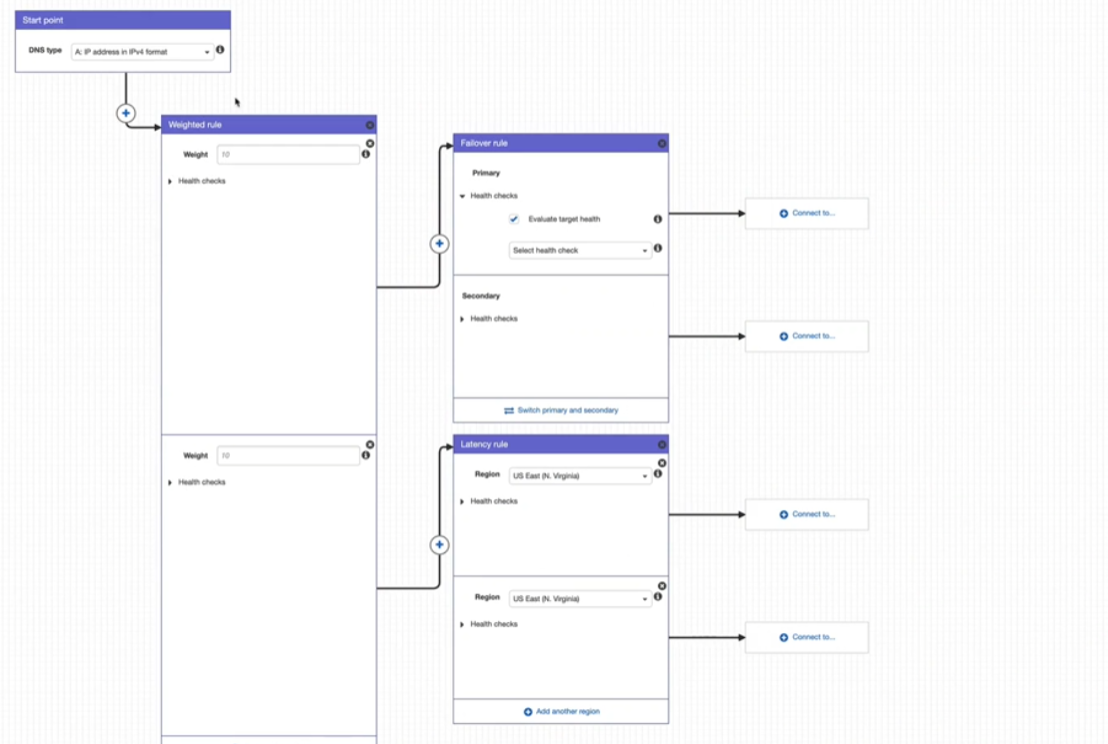
* Failover Routing Policy are used when you want to create an active/passive setup
* **When active site goes down the health check will fail, the failover routing policy will switch to the passive connection**
* Setup same way as weighted routing policies, except you specify as Primary or Secondary

## Geolocation Routing Policy

* **Geolocation routing lets you choose where your traffic will be sent based on geographical location of your users**

## Geo-proximity Routing Policy - ONLY when USING Route 53 Traffic Flow

* Route53 traffic flow uses a combination of geographic location, latency, and availability to route traffic to your cloud or on premises endpoints



* Geo-proximity routing is only available when using traffic flow which splits traffic based on geographical location. You can pick and choose how you want to route traffic by specifying a value known as **bias**

## Latency Based Routing Policy

* Allows you to route traffic based on the routing that provides the least latency response to the user

## Multivalued Answer Routing Policy

* Same as simple routing but with health check on each address

# Elastic Load Balancers

* Elastic load balancer automatically distributes incoming traffic across multiple traffic across multiple AZ’s
* **Cross-zone load balancing allows each load balancer node to distribute traffic across the registered targets in all enabled AZ’s. Otherwise the load is distributed within its OWN AZ**
* Three Load Balancer:
  + Application Load Balancer: Best Suited for loaf balancing HTTP and HTTP’s traffic. They operate at Layer 7 and are application aware. Intelligent Load Balancer
  + Network Load Balancer: Handle millions of requests per second with ultra-low latencies
  + Classic Load Balancer: Also, HTTP and HTTPs application balancing but for DEV and UAT environments
* Health check queries backend registered EC2 instances to verify if they are live
* Load balancer routes requests only to the healthy instances
* Will resume routing requests when it is in healthy state again
* **MUST DEFINE A ROUTE IN THE ROUTE TABLE OF THE LOAD BALANCER’S SUBNET TO THE INTERNET OR ATTACH IT TO THE PUBLIC SUBNET.**
* **THE PRIVATE SUBNET MUST BE IN THE SAME AZ AS THE ELB**

## Application Load Balancers – Layer 7 Load Balancer

* Layer 7 Load balancer: Works on Application Layer. Evaluates the listener rules in priority order to determine which rule to apply, and then selects a target from the target group for rule action
* **Listeners checks for connection requests from clients, using the protocol and port to configure**
* **You define rules that determine how the load balancer routes requests to its registered targets**
* Each rule consists of a priority, one or more actions, and one or more conditions
* When the condition for a rule is met then its action is performed,
* You must define a default rule for each listener, and you can optionally define additional rules.
* Target Groups routes requests to one of your resources in a target group

Timeline

Description automatically generated

* Application Load balancers are aware of layer 7 routing i.e., address routing and can route traffic to different availability zones
* Limitations of Application Load Balancers: Can only use HTTP and HTTPS listeners
* **To use an HTTPS listener, you must deploy at least one SSL/TLS server certificate on your load balancer**
* **The load balancer uses a server certificate to terminate the front-end connections, THEN decrypt requests from clients BEFORE sending them to the target**
* **DECRYPTION is done on the Load Balancer**.
* Certificates can be assigned via AWS itself. AWS is a certified authority
* IP address will change on Application load balancer unless specified

## Network Load Balancers \_Layer 4 Load Balancer

* **You use network load balancers when you expect millions of requests**
* **Works on Layer 4**
* **Can be used for any protocol not supported by Application Load Balancer**
* When it receives connection request, it selects a target from the target group for the default rule
* Attempts to open a TCP connection to the selected target on the port specified in the listener configuration
* **There are no rules on listeners on network load balancers**
* Protocols supports TCP, TLS, UDP, TCP\_UDP, 1-65535 ports
* **If no host is healthy, it will send the request to all instances**
* If protocol is TLS, you must deploy exactly one SSL server certificate

## Classic Load Balancers – Layer 4 Load Balancer w/ X-forwarded Sticky sessions

* These are legacy load balancers, used with HTTP/HTTPs with specific features like X-forwarded Sticky sessions.
* Strict Layer 4 Load Balancing for applications that rely purely on the TCP protocol
* X-forwarded header provide external IP address
* Gateway timeouts 504 error, troubleshoot web layer or database layer

## Sticky Sessions

* Classic Load Balancer routes each request independently to the registered EC2 instance with the smallest load
* Allows to bind user to a specific EC2 instances
* Problem with scaling if we terminate the instance, user will see error. Best to disable this.
* Application load balancers can also have sticky sessions but at the target group level

## Load Balancer with Deregistration Delay

* Deregister allows load balancer to keep existing connections open if the EC2 instances become de-registered or become unhealthy
* Enables load balancer to complete inflight requests
* You can disable this if you want connections to end immediately

## CloudWatch – Single Place That Logs/Monitors/Alarms/Actions Everything for almost all AWS Services (Health checks, CPU util, etc.)

* Cloud watch is a monitoring and observability platform designed to give us insight into our AWS architecture. It allows us to monitor multiple levels of our application and identify potential issues
  + Collects system level metrics like EC2 instances operating system
  + Application-level metrics such as: Is my application running?
  + Create alarms
* Default Metrics: Provided by default
  + CPU Utilization
  + Network Throughput
* Custom Metrics: Need to define CloudWatch Agent
  + EC2 Memory Utilization
  + EBS Storage Capacity
* You can have Cloud Watch perform actions
* There are NO default alarms
* AWS cannot see past the hypervisor level. I.e., if you need custom metrics for usage you need to define a cloud watch agent manually
* The more managed the service by AWS the more checks AWS provides
* Standard metric provision time is 5 mins whereas detailed is 1-min
* A period is the length of time associated with a specific Amazon CloudWatch statistic. Each statistic represents an aggregation of the metrics data collected for a specified period of time. Periods are defined in numbers of seconds, and valid values for period are 1, 5, 10, 30, or any multiple of 60.

## CloudWatch Logs – Logs Everything into Events à Streams àGroups

* Single location to setup all logs from every location
* Log Events: This is the record of what happened
* Log Streams: A collection of log events from the same source create a log stream. One continuous set of logs from a single instance
* Log Groups: A collections of log streams
* **Subscription FilterPatterns: You can create these to look for specific events and send these to specific AWS Services**

# Monitoring

## Amazon Managed Grafana – Container Metric Data Visualization

* **Amazon Managed Grafana: Fully Managed AWS service allowing secure data visualization for instantly querying, correlating, and visualizing operational metrics, logs and traces**
* Grafana: Widely deployed data visualization tool
* Create workspaces for dashboard and querying
* Several built-in data sources
* AWS Managed
* Built in security features
* Pricing: per active user in a workspace
* Data sources: Integrates with Amazon CloudWatch, Amazon Managed service from Prometheus, Amazon OpenSearch Service, Amazon Time Stream
* Used for
  + **Container Metric Visualization (data sources like Prometheus from EKS, ECS, or own Kubernetes Cluster)**
  + **IoT devices**
  + **Troubleshooting**

## Amazon Managed Service for Prometheus - Container Metric Auto-Scaling Monitoring

* **Serverless Prometheus-compatible service that allows to securely monitor container metrics at scale**
* Using same open-source Prometheus data model
* **AWS autoscaling and high availability**
* Can use PromQL
* Data is stored for 150 days
* Used for
  + **Container Metric Visualization (data sources like Prometheus from EKS, ECS, or own Kubernetes Cluster)**

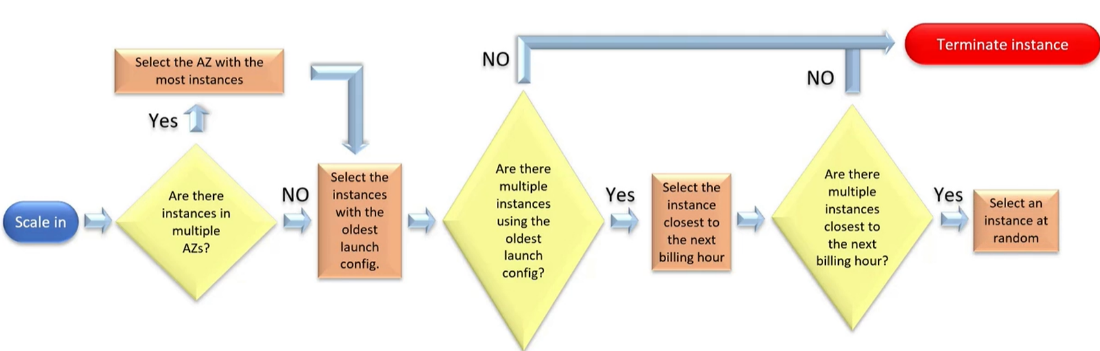
# Workflows

## Launch Templates and Launch Configurations

* A launch of template is all the needed setting to launch an EC2 Instance. It is a collection of settings you can configure so you don’t have to walk through the EC2 wizard
* Templates can be used for autoscaling
* If we include VPC in the template, then we can't use it for autoscaling
* Support versioning
* More granularity

## Auto-Scaling Groups

* Collection of EC2 instances considered to be a single cohesive unit and scaled all at once
* You first define a launch template
* How many AZ’s. At least two.
* You can register Autoscaling groups with Load Balancers to distribute incoming traffic
* Autoscaling can be used to define type of instances: Reserved, Spot, Or On-Demand
* These EC2 instances can be registered behind a load balancer. The Autoscaling group can be set to respect the load balancer health checks
* If the load balancer health check fails, the Autoscaling group can de-register that EC2 instance and restart it
* Define minimum, maximum and desired capacity that needs to be set to ensure you don’t have too many or too few resources
* Minimum of two EC2 instances
* Maximum can be anything
* Desired are how many instances at this second
* You can put instances in standby mode you can bypass health checks while performing upgrades or troubleshooting specific EC2 Instances
* Lifecycle hooks help with performing backups before instances are terminated
* EC2 Instances Termination Logic:



## Auto Scaling Policies

* Warm up period is the waiting period before being placed behind the load balancer, failing the health check and being terminated
* Cool-Down period is the pausing of autoscaling for a set amount of time to help avoid runaway scaling events
* Avoid Thrashing by creating instances quickly but spinning them down slowly
* Types of scaling
  + Reactive Scaling: Reactive based on utilization
  + Schedule Scaling: We know how to scale
  + Predictive Scaling: ML based scaling
  + Simple Scaling:

## Scaling Relational Databases

* 4 Types of scaling:
  + Vertical Scaling: Resizing the database from one size to another can create greater performance
  + Scaling Storage: It can be resized, but it can only go up, not down. Aurora automatically scales database storage
  + Read replicas: Create additional read only copies. Up to 15 read replicas
  + Aurora Serverless: Used for unpredictable workloads
* Switching data base types from relational to non-relational is easy and can be a viable scaling option

## Scaling Non-Relational Databases

* AWS does all the scaling for DynamoDB
* Provisioned Scaling
  + Generally unpredictable workload
  + Need to review past usage to set upper or lower scaling bounds
  + Most cost-effective model
* On-Demand Scaling
  + Pay small amount of money per read and write
  + Less cost-effective
* Can switch back and forth between provisioned and on-demand scaling
* Cannot switch back to on-demand for at least 24 hours

## Decoupling Workflow

* Tightly coupled means one instance is dependent on another
* Solution:
  + Load Balancer
  + SQS
* **NEVER WANT TIGHT COUPLING. ALWAYS WANT LOOSE COUPLING**
* Load balancers aren’t always the answer
* May be want something that can HOLD on to the message until it is ready to be retrieved
* SQS is a fully managed message queueing service that enables you to decouple and scale micro services, distributed systems, or serverless applications
* SNS is a fully managed messaging service for both application-to-application (A2A) and application to person (A2P) communications
* API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale

## SQS

* Poll-Based Messaging: Producer of message and consumer of message can work at different time frames **ASYNCHRONOUS PROCESSING**
* Load Balancer is **SYNCHRONOUS PROCESSING**
* **AWS Lambda can at maximum can retrieve 10 queue messages per batches from each ReceiveMessage Call**
* SQS Settings:
  + Delivery Delay: Delay message by a certain period post arrival
  + Message size: 256 KB is max size of text
  + Encryption is default IN TRANSIT NOT AT REST. You have option to encrypt at rest
  + Message Retention: Default is 4 days. Can be set between 1 min and 14 days
  + Long vs Short Polling: Consumer of message can connect for a certain period and wait for message instead of repeated polling. NOT ENABLED BY DEFAULT
  + Queue Depth: This can be a trigger for autoscaling
  + Visibility Timeout: When a message is being handled by consumer the message is hidden for a certain timeout period called the visibility timeout. MAKE SURE THE TIMEOUT IS SET FOR AS LONG AS IT TAKES TIME TO PROCESS MESSAGE

## Sidelining Messages with a Dead Letter Queue

* Same as SQS
* Same retention period 14 days
* You can create SQS DLQ from SNS topics
* When there is a problem handling a message, you don’t want the message to remain in the queue until the retention period
* Sideline message to dead letter queue instead when the number of retries have been hit
* ALWAYS SETUP CLOUDWATCH ALARM to watch dead letter queue depth

## SQS FIFO - Ordered Messages

* Standard SQS only offers best ordered offering
* May return duplicate records
* **Where ordering and deduplication matter use SQS FIFO**
* **Why use standards SQS? It allows for unlimited messages per second whereas SQS FIFO allows for ONLY 300 messages per second**
* If creating FIFO queue, you need to create a queue with “. FIFO” extension in name
* **Messages are added in a group using group ID ensure EACH group is sorted in order**
* **Message de-dup ID allows to check for the same ID for 5 mins if the same ID is seen it will be received by not processed**
* You can technically order messages manually by ordering in the text itself
* **FIFO costs a little bit more**

## SNS - Delivering Messages

* Pushed based messaging: messaged is pushed to consumer rather than being pulled by them
* SNS proactively deliver messages to endpoint subscription to it. This can be used to alert a system or person
* SNS Settings:
  + **Subscribers: Kinesis Data Firehose, Lambda, SQS, Email (JSON), HTTP, HTTPS, SMS**
  + Message size: 254KB
  + DLQ Support: Messaged to be delivered can be kept in DLQ
  + FIFO vs Standard: FIFO only
  + Encryption is default IN TRANSIT NOT AT REST. You have option to encrypt at rest
  + Access Policy: a resource policy can be added to a topic
* ALERT = SNS
* ALARMS = CLOUDWATCH à ALERT=SNS
* NO RETRIES except HTTP(s) endpoints and messages are automatically sent to DLQ

## API Gateway - Fronting Applications

* Controlled front door access to our application
* Security: API Gateway allows attaching a WAF to our endpoint to block certain country traffic; look for SQL injection attacks
* Stop Abuse: Adding rate limits, protect against DDOS
* Ease of use: Simple to setup that will kick of AWS services in your account
* Prevents baking of credentials when trying to access AWS services
* This how I connect to AWS services from public environment when it is not Web Traffic
* EVERYTHING on website that is some sort of API call goes through API Gateway to AWS Service like AWS Lambda

## AWS Batch - Batch Workload

* Allows you to run batch computing workloads within AWS (EC2/ECS/Fargate)
* Removes any heavy lifting for configuration and management
* Capable of provisioning accurately sized compute resources based on the number of jobs submitted and optimizes the distribution of workloads
* No installation required for batch computing software
* Four main components
  + **Job: unit of work that gets submitted (shell scripts executables, or docker images)**
  + **Job Definition: specify how your jobs are to be run**
  + **Job Queues: submitted to a specific queue**
  + **Compute Environment: Set of managed or unmanaged compute resources)**
* **Fargate is the recommended way of launching most batch jobs**
* **Sometimes EC2 is better compute resource** 
  + **whenever you need a custom AMI**
  + **Need more than four vCPU’s**
  + **Anything needing more than 30 GB memory**
  + **GPU or Graviton CPU (i.e., ARM based CPU)**
  + **LinuxParameters**
  + **For a large number of jobs its best to run on EC2 dispatched at a higher rate**
* **AWS batch vs AWS Lambda:** 
  + **Time limit: Lambda has 15 min limit**
  + **Disk Space: Limited disk space**
  + **Runtime Limitation: Limited types of runtimes vs Batch uses Docker**
* Managed Compute Environments
  + AWS manages capacity and instance types
  + Compute resource specs are defined when the environment is created
  + ECS instances are launched into VPC subnets that you specify and need access to ECS service endpoints
  + Default is the most recent and approved Amazon approved ECS AMI
  + You can use your own AMI
* Unmanaged Compute Environments
  + **You manage your own resources entirely**
  + AMI must meet Amazon ECS AMI spec
  + Good choice when you have extremely complex environment

## Amazon MQ - Brokering Messages

* **Managed message broker service allowing easier migration of existing applications to the AWS Cloud**
* AMOP 0-9-1, AMQP-1.0 OpenWire Protocol support
* Variety of programming application support
* Current support for Apache MQ
* Allows you to easily leverage existing apps without managing and maintaining your own system
* Why use Amazon MQ against SNS with SQS
  + Both offer topics and queues: one-to-one or one-to-many messaging
  + If migrating messaging service for existing application use Amazon MQ
  + For new applications use SNS with SQS
  + **Amazon MQ REQUIRES PRIVATE NETWORKING via VPC, Direct Connect or VPN, whereas SNS and SQS is publicly accessible**
  + **Amazon MQ HAS NO default AWS integration**
* Single instance broker:
  + Each broker lives within one availability zone. Perfect for development environment. RabbitMQ has a network load balancer in front
  + Highly available architecture to minimize downtime during maintenance, BUT depends on broker engine type
  + Amazon MQ for Active MQ: With active/standby deployments one instance will remain active at all times and one on standby
  + Amazon MQ for RabbitMQ: Cluster deployment are logical grouping of three broker nodes across multiple AZ’s sitting behind network load balancer

## AWS Step Functions - Coordinating Distributed Apps

* **This is a serverless orchestration service combining different AWS services for business applications**
* Comes with graphical console for workflow visualization
* Each component can either be a task or a state machine
  + State Machines: A particular workflow with different event driven steps
  + Task: specific state within a workflow representing a single unit of work
  + States: Every step within a workflow is considered a state
* Workflows have executions
* Executions are instances of a workflow run
  + **Standard Workflow**
    - Have an exactly once execution
    - Can run for up to ONE YEAR
    - Useful for long running workflows that require auditable history
    - 2000 execution per second
    - Pricing based on per state transition
  + **Express Workflow**
    - At least once execution workflow
    - Can run for up to five minutes
    - Useful for high event rate executions
    - Example use is IoT data streaming and ingestion
* States
  + Flexible: leverage states to either make decisions based on input, perform certain actions or pass output
  + **Language: Amazon State Language**
  + States: Elements within your state machine
* Type of State
  + Pass: Passes input directly to its output
  + Task: Single unit of work performed (Lambda Batch SNS)
  + Choice: Branching State
  + Wait: Creates a specific time delay within State Machine
  + Succeed: Stops execution
  + Fail: Stops execution and maps them as failure
  + Parallel: Runs parallel branches of executions within state machines
  + Map: Runs a set of steps based on element of an input array

## Amazon AppFlow - Ingesting Data from SaaS Applications to AWS

* **Integration with several SaaS apps and AWS Service**
* Data Ingestion from third party SaaS vendors
* **Bidirectional data transfer**
* Flow: Transfer data between source and destination
* Data Mapping: Determines how source data is stored within your destination
* Filters: Control which data is transferred from a source to a destination
* Trigger: Run on Demand, Run on Event
* Can be stored in Redshift, S3 or any compatible storage space

# Big Data

## Amazon RedShift

* Cloud Datawarehouse: Can hold up 16 Petabytes of data
* Relational Database
* Redshift is for Business Intelligence
* RDS is for App backend server
* Can have from 2 nodes to 125 nodes
* Can only be in one AZ in one VPC
* Available endpoints
* Can use Amazon Query Accelerator to boost querying performance by 10x and requires no additional charges and works on
  + RA3.16XL
  + RA3.4XL

## Elastic MapReduce

* This is for ETL process
* EMR is a managed ETL platform that allows you to process vast amounts of data using open-source tools, such as Spark, Hive, HBase, Flink, Hudi, and Presto
* EMR lives inside our VPC and uses internet gateway to store data in the S3 bucket

## Kinesis

* Kinesis is a real-time streaming data
* **SQS doesn’t offer real-time messaging service**
* Data Stream:
  + **Real-time streaming for ingesting data**
  + You’re responsible for creating the consumer and scaling the stream
  + You need to **define number of shards**
  + Upside is its real time; downside need to put in a lot of time
* Data Firehose:
  + **Near Real-time streaming**
  + Data transfer tool to get information to S3, Redshift, Elastic Search or Splunk
  + Easier to setup, **handles scaling**
  + Managed version of Data Stream
* Kenesis Data Analytics:
  + Very simple to tie Data Analytics into your Kenesis pipeline
  + No Servers
  + You only pay for amount of data you are passing
  + **ETL process**
* **Kenesis Video Stream**
  + **Like Datastream but build for video streaming applications**

## Athena & Glue

* Query data in a Data Lake
* Serverless Data Integration Service
* Replaces EMR

Graphical user interface, diagram, application, Teams

Description automatically generated

## QuickSight

* Quicksight is a fully managed business intelligence data visualization service
* It allows to create dashboards and share them within your company
* Used as a front to the architecture
* Usually used for interpreting the data or anything related to business intelligence

## AWS Data Pipeline

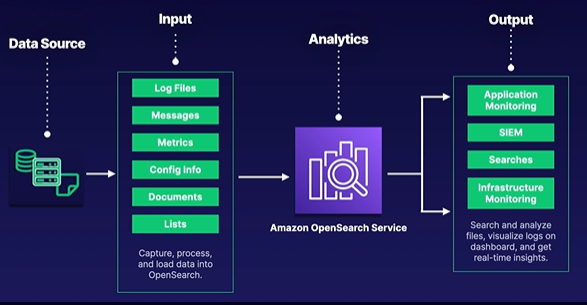
* **Data Pipeline is a Managed ETL tool for movement and transformation of data**
* **Data driven workflows** and orchestration tool that uses **task dependencies**
* AWS Data Pipeline enforces your chosen logic
* It is highly available, provides distributed infrastructure and is also fault tolerant
* It automatically tries failed activities. Configure via Amazon SNS for failure or even successful tasks
* **It syncs with Amazon DynamoDB, Amazon RDS, Amazon Redshift, and Amazon S3**
* **Syncs with AWS compute service, EC2 and EMR**
* Components:
  + Pipeline definition: Specify the business logic of your data management needs
  + Managed compute: Service will create EC2 instances to preform your activities
  + Task Runners: polls for different tasks and performs them when they are found
  + Data Nodes: Defines the locations and types of data that will be input and output
  + Activities: These are pipeline components used to perform work
* Used Cases
  + Processing data in EMR using Hadoop Streaming
  + Importing and importing DynamoDB Data
  + Copy csv files or data between S3 buckets
  + Exporting RDS data to S3
  + Copying data to Redshift
* Can be run daily or weekly

## Amazon MSK

* **Fully managed service for Apache Kafka** for streaming data applications that leverage Apache Kafka
* **Provides control-plane operations. Creates, updates and deletes clusters as required**
* **Allows you to work on the data-plane operations for producing and consuming streaming data**
* Good for applications that need to use open source of Apache Kafka
* Components
  + Broker Nodes
  + Zoo-keeper Nodes
  + Producers, Consumers, and Topics
  + Flexible Cluster Operations
* Why use MSK over self-setup Apache Kafka
  + **Automatic Recovery: automatic detection and recovery from common failure scenarios**
  + Detection: Detects broker failures result in mitigation or replacement of unhealthy nodes
  + Reduce Data: Tries to reuse storage from older brokers for data replication
  + Reduction in Time Required: It is limited to however long it takes Amazon MSK to complete detection and recovery
  + After Recovery: Communication happens with the same broker IP as before
  + Good to Know
    - MSK Serverless: Auto provisioning and scaling
    - Fully compatible with Apache Kafka. You can use same client apps for producing and consuming data
    - MSK Connect: Allows developers to easily stream data to and from Apache Kafka Clusters
* Securing and Logging
* **TLS 1.2 for encryption in transit can be enabled and encrypted by default at rest**
* **Broker logs can be collected in Amazon CloudWatch, Amazon S3 and Amazon Kinesis Data Firehose**
* **Metrics can be gathered and sent to CloudWatch**
* **Logs all API calls to AWS Cloud Trails**

## OpenSearch

* **OpenSearch is a managed service allowing you to run search and analytics for many different use cases**
* Allows you to quickly ingest search and analyze data in your cluster and is commonly part of an ETL process
* Easily scalable infrastructure running the open-source OpenSearch Service
* Leverage IAM for access control VPC security groups, encryption at rest and in transit, and field level security
* Multi-AZ-capable service with master nodes and automated snapshots
* Use SQL for BI Apps
* Easily integrate it with Amazon CloudWatch, AWS CloudTrail, Amazon S3, and Kenesis
* Carrying out ETL jobs on logging work



# Serverless

* Layers
  + Physical: Data Center – All computing needs
  + Virtualization – We started to visualize computers. Computers running more computers
  + Running physical hardware is holding us back so we now manage virtual compute
  + Serverless – We focus on coding while leaving management of compute to the cloud
* Benefits of Serverless
  + Ease of use: Compute is AWS Managed
  + Event Based: Serverless compute resources can be bought online response to an event happening
  + Billing Model: Pay as you go

## Lambda

* Lambda is a serverless compute service that lets you run code without provisioning
* Choose specific runtimes I.e., Java, Python etc.
* Permissions If your Lambda functions need to make an API call, you’ll need to attach a role
* **Networking: You can run it in VPC and subnet or run it without any network settings**
* Resources: You get to define the amount of available memory and will allocate how much CPU and RAM your code gets
* Trigger: Defining a trigger will kick of a Lambda function
* Built in logging using CloudWatch.
* **Limit is 15 mins**
* **Memory max is 10 GB and CPU scales accordingly**
* **Lambda is user to add features to AWS**
* **If you need to automatically remove entries from a security group, start and stop instances, or do anything else that isn’t built in**
* **When you update the code for the Lambda function there is a small time window for which the request may be handle with old or new code**
* **Create an Alias for ARN’s of you Lambda function instead of specifying the version qualified ARN with your applications. This is to prevent the need to change ARN’s with each new version release**

## AWS Serverless Application Repository

* Allows users to find, deploy, or even publish their own serverless applications
* Share privately within the organization or publicly for the world
* **Manifest File: SAM template which is integrated with Lambda**
* **Publishing Apps makes them available for others to find and deploy**
  + Define AWS SAM Template
  + Private by default
* **Deploy: Find and deploy published applications**
  + Browse public apps without needing an AWS Lambda console
  + Be careful of trusting all applications

## Container

* **Packages can be deployed in an enclosed environment, flexible**
* **Easy way to standardize package deployments**
* Container is stripped form of a virtual machine with only core dependencies required to run an application
* Dockerfile – Set of instructions to define a container
* Image – immutable template for config files
* Registry – Place to deploy your images
* Container – A running copy of your image

## Running Containers in ECS and EKS

* ECS can manage 1, 10, hundreds or thousands of containers. It will appropriately place the containers and keep them online
* **Customer’s have operating system access for ECS**
* Has integration with elastic load balancers as they come online and go offline
* Role Integration: Containers can have individual roles attached to them making security a breeze
* Ease of use
* Problem: Can only work on AWS
* EKS is the open-source container management and orchestration alternative to ECS
* Can be used on-premises and in the cloud
* EKS – AWS-managed version of Kubernetes
* **ECS is best for use for container deployments completely in the cloud**
* **EKS is a managed version of the open-source Kubernetes container**
* ECS:
  + Define task – Task Memory, Task CPU, Name, Image file location, Startup command
    - **A task is a JSON format that describe one or more containers that define your applications**
  + Define Cluster - cluster size and type, VPC, security group
  + Assign Task to Cluster
  + You’ll find the ECS instance on EC2
* ECS service definition: defines which task definition to use with your service, how many instantiations of that task to run and which load balancers associated with your tasks
  + Define Cluster
  + Service Name
  + Task Definition
  + Load Balancers
  + Client Token
  + Desired count

## Fargate – Severless Container Deployments (No Access to OS system i.e. EC2 Instances)

* Serverless architecture for running containers without managing the instances ourselves
* Fargate loses that operating system access
* When do you use Lambda vs Faregate vs EC2
  + **EC2**
    - **Better pricing model**
    - **Used for long running container**
    - **Allows multiple containers to share the same hosts if we want to do so.**
  + **Fargate**
    - **FARGATE DOES NOT WORK BY ITSELF. NEEDS EKS and ECS**
    - **More consistent workloads**
    - **Allow docker use across the organization and greater level of control by developers**
    - **Primarily for containers that do not need to run all the time**
    - **Pay based on resources allocated and time ran**
  + **Lambda**
    - **Lightweight functions, processing files from S3**
    - **Great for predictable or inconsistent workloads**
    - **Perfect for applications that can be expressed via single functions**
* Code (Lambda) vs Container (Fargate)
* Setup:
  + Define Task – Task Memory, Task CPU, Name Image File Location
  + Define Cluster – No need to define EC2 instance
  + Assign Task to Cluster
  + Deploy Task

## EventBridge

* **EventBridge is CloudWatch Events and is a serverless event bus. It allows you to pass events from a source to an endpoint. Used to bind serverless applications together**
* **ANY API CALL CAN TRIGGER AN EVENTBRIDGE**
* **Fastest way to respond to things happening in your environment**
* Used primarily for triggering serverless resources based on state changes of any other resources
* Creating a rule:
  + Define pattern: Event Based or Scheduled
  + Custom Events or AWS Events: Choose AWS Provider
  + Select Event Type
  + Select your Target Resource
  + **Tag everything to make sure what each event does**

## ECR

* **AWS managed container image registry that offers secure, scalable, and reliable infrastructure**
* Private container image repositories with resource-based permissions via IAM
* **Support Open Container Initiative (OCI) Images, Docker Images, and OCI Artifacts**
* Private Registry provided to each AWS Account: Create one or more for image storage
* Authentication token is required to push and pull images to and from registries
* Repositories manage the docker image, OCI Image, OCI Artifact
* Repository policies control all access to repos and images
* Container Images that get pushed and pulled from the repository
* **ECR Public is a public repository**
* Features
  + **ECR offers lifecycle policies which helps to manage images and define rule by which images should be cleaned. You can test your rules before applying them**
  + **It offers image scanning when you push an image to the registry and provides reports for each scan which identify vulnerabilities**
  + **Cross-region and cross-account support for share repository**
  + **Each registry is regional for each account**
  + **Pull through cache rules allow for caching public repos privately and define periodic update checks**
  + **Tag mutability: this prevents image tags from being overwritten. This is configured per repository**
* Integrations
  + Integrates with you own container infrastructure
  + ECR, ECS, Amazon Linux Container

## Amazon EKS Distro - Open-Source Kubernetes Distribution

* **This is a Kubernetes distribution based on and used by Amazon EKS**
* **Same version and dependencies deployed by Amazon EKS**
* **EKS-D is managed by you unlike Amazon EKS which is managed by AWS**
* **Can run it anywhere and managed by YOU**
* **Use this when you need to run versioned deployments of clusters outside of AWS managed services**

## Amazon ECS Anywhere and Amazon EKS Anywhere - Orchestrating Containers Outside AWS

* EKS Anywhere
  + Allows you to run Kubernetes clusters on-premises in the same managed way as Amazon EKS
  + **Based on EKS Distro**
  + Offers full lifecycle management of multiple Kubernetes clusters
  + **Control Plane is managed by the customer**
  + **Location of the control plane is at the CUSTOMERS datacenter or operational datacenter**
  + **Updates are performed using manual CLI or Flux**
  + Offers curated packages that provide extended core functionalities of K8’s cluster
  + Is part of the Enterprise Subscription packaging only
* ECS Anywhere
  + **Feature of Amazon ECS allowing you to manage container-based apps on premise**
  + No orchestration needed: No need to install and operate local container orchestration tools
  + Completely managed solution enabling standardization of container management across environment
  + NO ELB Support
  + New launch type called EXTERNAL for creating services or running tasks on-premises
  + Requirements
    - **SSM Agent, ECS Agent and Docker Installed**
    - Register external instance as SSM Managed instances
    - **Easily create an installation script within the ECS console and then run it in the external instance**
    - The scripts will contain activation keys and commands for required software
    - **Execute scripts on you on-premises VM’s or bare metal servers**
    - Deploy containers using External Launch Type

## Amazon Aurora Serverless

* Aurora Provisioned is different from Aurora Serverless
* **On-Demand and Auto-scaling configuration for the Amazon Aurora database service**
* Automation of monitoring workloads and adjusting capacity for databases
* **It is based on Demand and per second billing and is therefore good billing**
* **Based on Aurora Capacity Units**
* **Can set minimum or maximum ACU for scaling requirements – can be shut off and be zero**
* Allocated very quickly by AWS managed warm pools
* One ACU = 2GB memory, matching CPU and network capability
* **Same data resiliency as Aurora provisioned: six copies of data across three AZ’s**
* **Multi – AZ offered for high availability**
* **Key Use Cases:**
  + **Variable Workloads – Unpredictable sudden activity**
  + **Multi-Tenant Apps – Let the service manage database capacity for each app**
  + **New Apps – Load could be unpredictable**
  + **Dev Testing Apps – Testing Dev Loads**
  + **Mixed Use Apps – Where you have spotty requirements for database OLTP work**
  + **Capacity planning – Can switch back and forth between provisioned and serverless**

## Amazon X-Ray – Application Insights and viewing downstream calls

* **Collects application data for viewing filtering and gain insight about requests and responses (response time)**
* Collects data for any downstream resource calls as well I.e., microservices, API’s or databases
* Receives traces from your applications for allowing insights
* Multiple options: Can run X-Ray Daemen or resources can send tracing headers
* Components
  + Segments: Data containing resource names, request details, and other information
  + Sub-Segments: Segments providing more granular timing information and details
  + Service Graph: Graphical representation of interacting services in requests
  + **Traces ID: Track path to a requests**
  + Tracing Header: HTTP headers that contains sampling decision and trace-ID
* **AWS X-Ray Daemon runs on port 2000 UDP**
  + **Installed on EC2, ECS, AWS Lambda (as toggle)**
  + **AWS Elastic Bean Stalk**
  + **API Gateway**
  + **AWS SNS and SQS**

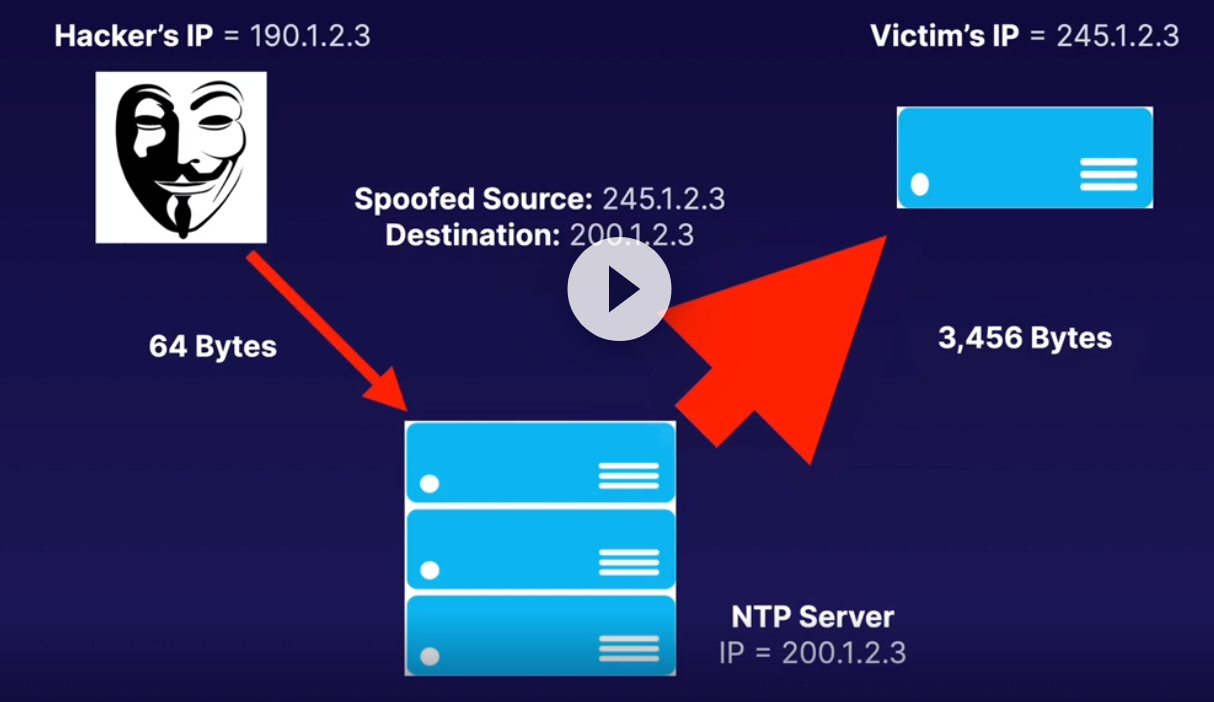
## AWS AppSync

* **Robust, scalable GraphQL Interface for application developers**
* Combines data from multiple sources
* Enables data interaction for developers via GraphQL
* GraphQL: Data language that enables apps to fetch data from servers
* Seamless integration with React, ReactNative, IOS and Android
* **Declarative coding, front end app data fetching**

# Security

## Distributed Denial of Service

* **Layer 4 DDoS attack refers to SYN flooding**
* Works at the transport layer
* TCP connection works based on the 3-way handshake
  + The client sends a SYN packet to a server, the server replies with a SYN-ACK and the client then responds to that with an ACK
  + After 3-way handshake, the TCP connection is established. After this the application starts sending data using Layer 7 such as HTTP
* Syn Flooding uses the built-in patience of the TCP stack to overwhelm the server by sending large amount of SYN packets and the ignoring the SYN-ACK responses. This throws the server into an indefinite loop waiting for several ACK.
* Amplification Attack: sending attacks using spoofed IP address – NTP. The server is sent the spoofed NTP request to which it responds with a greater payload than the initial request (28 to 54 times larger) and hence overload the server



* **Layer 7 Attacks – Tons of GET or POST request sent to a web server**

## Cloud Trails – Logs ANY API CALLS or Console Calls in AWS

* Records AWS console actions and API Calls to increase visibility of user and resource activity
* You can use this to identify which users and accounts called the AWS, the source IP address from which the calls were made and when the calls occurred
* **Like CCTV for API calls or Console**
* **DOES NOT LOG RDP or SSH traffic**
* **What is logged?**
  + **Metadata around API Calls**
  + **The identity of the API caller**
  + **The time of the API call**
  + **The source IP Address of the API caller**
  + **The request parameters**
  + **The response parameters**
* **Key Uses**
  + **After the fact incident investigation**
  + **Near real time intrusion detection**
  + **Industry and regulatory compliance**

## AWS Shield

* Free DDoS Protection for AWS customers on ELB, CloudFront, and Route53
* Protects against SYN/UDP floods, reflection attacks, and other layer 3 and 4 attacks
* AWS Shield Advanced provides enhanced protections for your applications running on ELB, Amazon CloudFront and Route 53 against larger more sophisticated attacks
  + **Offers always on, flow-based monitoring of network traffic and active application monitoring to provide near real-time notification on DDoS Attacks**
  + **Protects your AWS bills against higher fees**
  + **Gives you 24/7 access to DDoS Response Team**

## AWS WAF – Firewall to block any Layer 7 based traffic based on Rules for ELB and CloudFront

* It is a web application firewall that lets you monitor the HTTP and HTTPS requests that are forwarded to Amazon CloudFront or an Application Load Balancer
* It also lets you control access to you content
* Can configure to allow specific IP addresses to make this request or what control what query string parameters need to be passed for the request to be allowed
* WAF rules define how to inspect HTTP’s Requests and the action to be taken on a request when it matches the inspection criteria. Rules are defined based on Rule groups or web ACL;s
* Will either allow the content to be received or give an HTTP 403 error
* **Operates at Layer 7**
* **Block Layer 7 DDoS Attacks**
* You can define conditions based on:
  + Originating IP Address
  + Originating Country
  + Values in request header
  + **Presence of SQL code**
  + **Presence of a script**
  + Strings that appear in requests

## Amazon GuardDuty

* It is a threat detection service that uses machine learning to continuously monitor for malicious behavior
  + Unusual IP calls, calls from a known malicious IP
  + Attempts to disable Cloud Trail logging
  + Unauthorized deployments
  + Compromised instances
  + Reconnaissance by would be attackers
  + Port scanning and failed logins
* Alerts appear in the GuardDuty console and CloudWatch Events
* **Receives feeds from third parties like Proofpoint and CrowdStrike, as well as AWS Security, about known malicious domains and IP Addresses, etc.**
* **Monitor CloudTrail Logs, VPC Flow Logs, and DNS logs**
* Centralized threat detection across multiple AWS accounts
* **Automated response using CloudWatch Events and Lambda**
* Machine learning and anomaly detection
* **Threat detection with AI**
* 7-14 days to set a baseline for what is normal flow of log traffic
* **Once active, you will see finding on GuardDuty console and in Cloud Watch Events only if** GuardDuty detects behavior it considers a threat
* 30 days free
  + Then based on CloudTrail events
  + Volume of DNS and VPV flow logs

## AWS Firewall Manager

* AWS Firewall Manager is a security management service in a single pane of glass. **Allows you to centrally set up and manage firewall rules across multiple AWS accounts and applications in AWS Organizations**
* **Allows you to setup WAF rules for your Application Load Balancers, API gateways, and Amazon CloudFront distributions**
* **Ensures compliance for existing and new applications across multiple accounts**

## Macie – ML based Scanning of Personal Identity/Health Information

* Personally, Identity Information: Home Address, SIN Number, etc.
* **Automated analysis of data using AI to identify PII PHI (health) information**
* Alerts you to unencrypted buckets
* Alerts you to public buckets
* Can also alert you about buckets shared with AWS accounts outside of those defined in you AWS organizations
* **HIPAA and GDPR compliance**
* Can filter and search Macie alerts in AWS console
* **Alerts sent to Amazon EventBridge can be integrated with your security incident and event management systems**
* Can be integrated with AWS Security Hub for a broader analysis for your organization security posture
* **Can be integrated with AWS Step Functions to automatically take remediation actions**

## Amazon Inspector

* Automated security assessment service
* Helps improve security compliance for applications deployed on AWS
* **Automatically assesses applications for vulnerabilities and deviations from best practices on EC2 Instances or VPC’s**
* Provides detailed list of security findings order by level of severity
* Can review them directly of as part of assessment report on Amazon Inspector Console or API
* Two types:
  + **Network Assessments: checks for reachable ports – Inspector Agent is not required**
  + **Host Assessments: checks for vulnerable software, host hardening and security best practices – Inspector Agent is required**
* How does it work:
  + Create assessment target
  + **Install agent on EC2 instance**
  + Create assessment template
  + Perform assessment run
  + Review findings against rules
  + Provides report and provides severity of concerns

## Amazon KMS and Cloud HSM

* **Key management service that makes it easy for you to create and control the encryption keys to encrypt your data**
* Integrates with EBS, S3, RDS and other services
* Provides centralized control over lifecycle and permissions of your keys allowing you to create new ones and control who can manage them separately from who can use them
* CMK: Customer Master Key: Logical representation of the master key, the CMK includes meta data such as key ID, creation date, description and key state
  + It contains the key material to encrypt and decrypt your data
* **Steps for KMS:**
  + **Start by requesting a CMK**
  + **Define its lifecycle and permissions**
* HSM is a hardware security module that is a physical computing device that safeguards and manages digital key as well as performs encryption and decryption secure crypto functions
* Contains one or more crypto processor chips
* **3 ways to generate CMK**
  + **AWS creates CMK for you: the key material for a CMK is generated within HSM managed by AWS KMS**
  + **Import key material from on-premHSM and associate with CMK**
  + **Generate key material and use in an AWS CloudHSM cluster as part of the custom key store in AWS KMS**
* Key Rotation:
  + Can be rotated every year PROVIDED they were generated by AWS KMS HSM
  + Not supported for keys generated in CloudHSM clusters using the custom key store feature
* Policies
  + Primary way to manage access AWS KMS CMK’s
  + Key policies must be resource policies attached to customer master keys
* **3 Ways to Control Permissions**
  + **Use Key Policies only: define access to CMK in a single document**
  + **Use IAM Policies in combination with the key policy**
  + **Use grants in combination with the key policies: this allows users to delegate access to others**
* CloudHSM
  + Cloud HSM: hardware security module allows you to generate your own encryption key on the cloud
  + Physical device, entirely dedicated to you that can be deployed in a highly available fashion
* **KMS vs CloudHSM**
  + **KMS** 
    - **Shared tenancy**
    - **Automatic Key Rotation**
    - **Automatic Key Generation**
  + **CloudHSM**
    - **Dedicated HSM for you**

**Full control of underlying hardware**

* + - **Full control of users, groups, keys etc.**
    - **No automatic Key Rotation**

## AWS Secret Manager

* **Service that securely stores, encrypts and rotates your database credentials and other secrets**
* Encryption in transit and at rest using KMS
* **Applications can make API calls to secret manager to retrieve secrets automatically**
* Reduces the risk of credentials being compromised
* You can store
  + RDS credentials
  + Credentials for non-RDS DB’s
  + Other types of key-value pairs (SSH keys)
* **If you enable rotation, Secret Manager immediately rotates the secret once to test the configuration**
* **Make sure all applications that use these credentials are updated to retrieve credentials from this secret using Secret Manager**
* **If your applications are using embedded credentials do not enable rotation because the embedded credentials will no longer work, and this will break your application**
* Enable Rotation is recommended

## Parameter Store

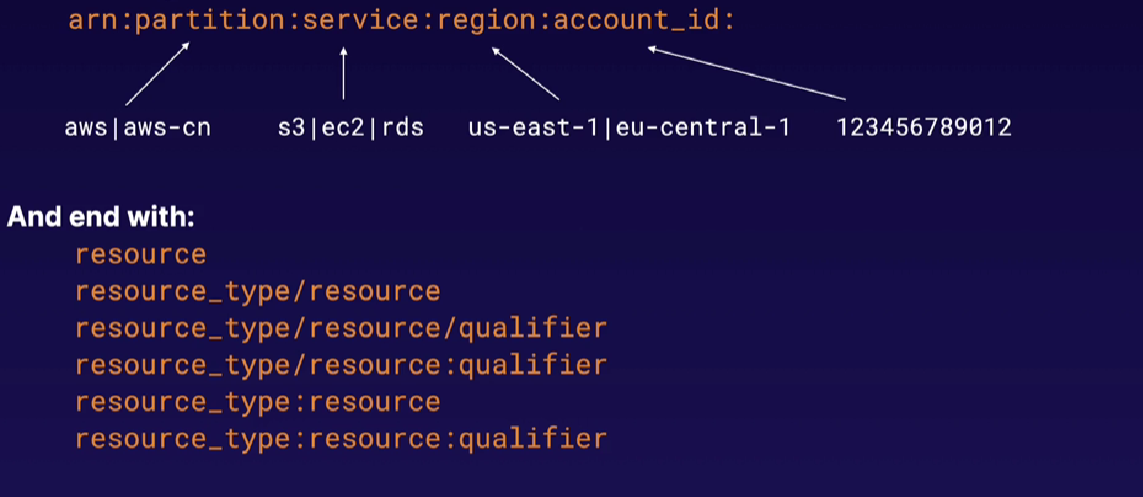
* Capability of AWS System Manager that provides secure, hierarchical storage for configuration data management and secrets management
* You can store data such as passwords, databases strings, Amazon Machine Image ID’s and license codes as parameter values as plain text or encrypted
* It is free
* Limits
  + Only 10,000 parameters allowed
  + No key rotation

## Temporarily Sharing S3 Objects Using Pre-assigned URLs or Cookies

* All objects in S3 are private by default
* Only the object owner has permissions to access the objects
* Owners can optionally share objects by creating a Pre-assigned URL using their own credentials to grant time limited access to download objects
* Pre-assigned URL: You must provide security credentials specify a bucket name and an object key, and indicate the HTTP method as well as expiration date and time
* Anyone with the Pre-assigned URL can access the file
* Pre-assigned Cookies allow you to access multiple files: Cookie is stored on the user’s computer. I.e., paywall access

## Advance IAM Policy Documents

* ARN: Amazon Resource Name:



* IAM, S3 has omitted value
* IAM policies
  + JSON documents that define permissions
  + Identity policy
  + Resource policy
  + No effect until attached
  + List of statements
* Policy document is a list of statements
* Each statement matches and AWS API request
* {
  + ‘Version:’ ”2012-10-17",
  + ‘Statement’: [
    - {
      * “Sid”:” Specific Table”
      * “Effect”: “Allow”
      * “Action”: [
        + “DynamoDB: BatchGet”
        + “DynamoDB: DescribeStream”
      * ],
      * “Resources”: “\*”
    - }
  + ]
* }
* Permission Boundaries used to delegate administration to other users
* Prevent escalation or unnecessarily broad permissions
* Control maximum permissions and IAM policy can grant
* Implicitly access is denied unless granted
* Explicit deny trumps everything else
* Only attached policies have an affect
* Multiple policies can be joined to a resource
* AWS Managed vs custom managed policies

## AWS Certificate Manager

* **Allows you to create public and private SSL certificates to use with other AWS services**
* **Integrates with Cloudfront Distribution, Elastic Load Balancer and allows you to deploy SSL certificates in AWS Environments**
* **Cost: Free to deploy Pay: for each use**
* **Automated renewals and deployments: rotate certificates**
* Easy to setup

## AWS Audit Manager - Continuously audit your AWS usage to make sure you stay compliant with industry standards and regulations

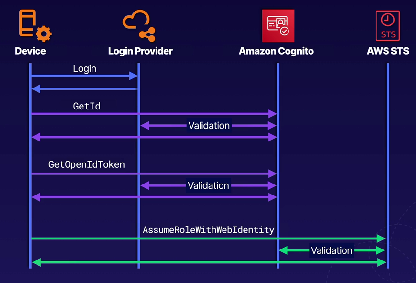
* Continuously audit your AWS usage to make sure you stay compliant with industry standards and regulations
* Automated service and produces reports services for GDPR compliance
* Automated Evidence Collection
* Continuous Auditing and compliance

## AWS Artifact - Download Compliance Documents

* Single source where you can get compliance information about compliance reports or select online agreements
* Huge number of compliance reports like GDPR reports, SOC reports or PCI reports, ISO reports
* DISTRACTOR QUESTION

## Amazon Cognito - Provides authentication, authorization and user management engine for web and mobile apps

* Provides authentication, authorization and user management engine for web and mobile apps without the need for custom code. Can login using password they create or use third party
* Sign up and sign in for your apps
* Access for guest users
* Acts as identity broker between app and user
* Synchronize user data across multiple devices
* Recommended for all AWS dependent services
* Use Cases
  + Authentication
  + Third Party Authentication
  + Access Server-Side Resources
  + Access AWS AppSync Resources
* User Pools and Identity Pools
* **User Pools are directories of users that provide sign-up and sign in options for you application**
* **Identity Pools allow you to give your users access to other AWS services**



* **Steps:**
  + **Authentication happens first by using the User Pool with provides an authentication token**
  + **The token is then used by the identity pool to get AWS credentials**
  + **AWS Services can then be accessed using the credentials**

## Amazon Detective Best for Root Cause Analysis of suspicious activities

* **Used to analyze, investigate and quickly identify potential security issues or suspicious activities**
* **Works with MULTIPLE AWS Services. USEFUL FOR FINDING ROOT CAUSE**
* **Detective pulls data from AWs resources and uses machine learning, graph theory to build linked set of data that enable you to quickly figure out the root cause of security issues**
* Sources:
  + VPC flow logs
  + Amazon Guard Duty
  + Amazon Kubernetes services
* Use Cases
  + Triage your security findings
  + Threat Hunting

## AWS Network Firewall – Firewall for VPC to VPC communication

* Physical firewall protection across VPC.
* Managed Service
* Block SMP requests
* Provides intrusion Prevention System
* Filter Internet Traffic
* Filter Outbound Traffic
* Inspect VPC-to-VPC traffic
* **Internal filtering**

## AWS Security Hub

* Single place to view all security alerts from services like Amazon Guard Duty, Inspector, Macie, and Firewall Manager

# Automation

* Manual processes can have costly consequences
* Automation conserves time
* Easier to prevent security incidents
* Consistency: providing the same results every single day
* 3 Upcoming Services
  + CloudFormation: allows you to provision resources quickly and consistent and manage them throughout the lifecycles, by treating infrastructure as code
  + Elastic Beanstalk: Easy to use all in one service for deploying and scaling web application and services developed with a variety of supported languages
  + System Manager: gives you the ability to easily patch, update, manage and configure EC2 Instances along with on-premises architecture

## CloudFormation

* CloudFormation is a declarative programming language. Supports JSON or YAML formatting

Text

Description automatically generated

* Deploy YAML or JSON file as template
* Allows parameterization
* Maps can be created to match key value pairs
* Perfect for creating immutable architecture
* HARD CODED values for Resource ID’s can cause templates to fail. Use mappings
* If it finds errors, CloudFormation rolls back
* CloudFormation makes same API Calls you make manually
* Immutable: can easily deploy or destroy architecture all together

## Elastic Beanstalk - No-code solution for deployment of resources

* Elastic Beanstalk builds your infrastructure for you. No-code solution for deployment of resources; you just bring your code
* Will also build out and manage the inside of the EC2 instances for you
* Steps
  + Select Application Name
  + Identify Platform: Python, Java, Node.js, Docker
  + Upload your code
  + More option include:
    - Identify Software
    - Instance setup
    - VPC setup
    - Network setup
* Supports both Windows and Linux applications

**Systems Manager**

* Collection of tools to automate BOTH AWS resources and on-prem resources
* It is FREE
* Features
  + Automation Documents (Run Books): Can be used to control your instances or AWS resources. Sort of like template Bash Scripts
  + Run Command: Execute commands on your host
  + Patch Manager: Manage application updates
  + **Parameter Store: Securely store your secret values**
  + **Hybrid Architecture: Allows you to control on-premises architecture using System** Manager
  + Session Manager: Remotely connect and interact with your architecture

# Caching

* External: Cache Data that’s going to be returned to our users (e.g., images, videos, and static content)
* Internal: Cache within the database
* **ALWAYS PICK SOULTIONS THAT PREFER CACHES**
* PUT CACHES EVERYWHERE
* Helps with Speed
* Need to know which is External vs Internal caches and when to use which

## CloudFront: Fast paced cache enabled content delivery network

* How does it use Edge Locations
* Cloud Front is a fast content delivery network for delivering videos, applications and APIs to customers globally
* Helps reduce high latency and provide higher transfer speeds using AWS edge locations
* Important Setting
  + **Defaults to HTTPS connection with ability to add SSL certificates**
  + Global Distribution: You CAN’T PICK SPECIFIC COUNTRIES OR EDGE LOCATIONS
  + Supports AWS Endpoints or Non-AWS Endpoints
  + Expiring Content: Can put TTL on content cache or force expire cache
  + **Can limit access by assigning pre-signed URL’s or pre-signed cookies**
  + **Use Presigned Cookies when you don’t want to change URL’s multiple times**
  + Can setup custom SSL certificates
  + CAN BE USED FOR AWS AND ON-PREM ARCHITECTURE
  + WAF Better tool for denying or providing access to specific edge locations
* **AWS Lambda function can be triggered by the following CloudFront events:**
  + **Viewer Request**
  + **Viewer Response**
  + **Origin Request**
  + **Origin Response**

## ElasticCache and DAX

* ElasticCache is managed version of Memcached and Redis
* MemCache:
  + Simple database caching solution
  + Not a database by itself
  + No failover or multi-AZ support
  + No backup
* Redis:
  + Support as a caching solution
  + **Function as the standalone database**
  + Failover and Multi-AZ support
  + Supports Backup
* **DynamoDB Accelerator (DAX)**
  + In memory cache and allows you to reduce DynamoDB response time from milliseconds to microseconds
  + You have control over DAX: I.e., TTL for the data, and maintenance window for changes and updates
  + What type of cache should you be using
    - DAX: It is only for DynamoDB
    - ElasticCache provides a bit more flexibility
* In-memory database: DAX is better, but options are between Redis and DAX

## Global Accelarator

* This is a networking service that sits in front of your user traffic through AWS global network infrastructure. It can increase performance and help deal with IP caching
* If ELB IP address is cached by user and then the ELB goes offline, USER CANNOT ACCESS WEBSITE. **Put Global Accelerator in front of ELB**
* **Allows to mask complex infrastructure. All IP addresses can be mapped a few in Global Accelerator**
* **It speeds up by routing traffic through AWS backbone**
* **Weighted pools: you can create weighted groups behind the IPs to test new features and failure in environment**
* Steps:
  + Give it a name
  + Setup custom IP addresses for accelerator
  + Define listener port i.e., Port 80
  + Chose endpoint: ELB or non-AWS endpoint
  + Chose different weights if multiple endpoints exit

# Governance

## AWS Organizations - Free governance tools that allow you to create and manage multiple AWS accounts

* Free governance tools that allow you to create and manage multiple AWS accounts. With it you can control your accounts from a single location rather than jumping from account to account
* Key features and best practices in Organization
  + **Logging Account: dedicated to only storing logs. Cloud Trail integration supports log aggregation and then using SCP to limit making changes on that account**
  + Programmatic creation of AWS accounts
  + **Reserved Instances can be shared cross accounts: Can be turned off**
  + Consolidated Billing
  + **Service Control Policies: set limits to what users can do within an account. Can apply it as global policy. Trumps everything. Even ROOT ACCOUNT. THEY NEVER GIVE YOU PERMISSION, ONLY TAKE THEM AWAY**

## AWS RAM (Resource Access Manager) - Cross Account Role Access with Organization

* Free service that allows you to share AWS resources with other accounts and within your organization
* You can share your resources easily rather than copy them between accounts
* What can we share:
  + Transit Gateway
  + VPC subnets
  + License Manager
  + Dedicated Hosts
  + Route53 Resolver
  + Etc.
* VPC Peering vs RAM
  + If using in the same region: Use RAM
  + Else use VPC Peering
  + But can also use VPC Peering
* This is to reduce duplicate costs

**Cross Account Role Access**

* As the number of AWS accounts you manage increases, you’ll need to setup cross account access. Duplicating IAM accounts creates a security vulnerability. Cross-account role access gives you the ability to set up temporary access you can easily control.
* Step
  + Create IAM role
  + Give it some permissions
  + Give user or group access to that role

## AWS Config – Resource comparison against standards. Set rules and define actions. Multi account, multi region data aggregation service, allows to check

* It is an inventory management and control tool. It allows you to show the history of your infrastructure along with creating rules to make sure it conforms to the best practices you’ve laid out
* Allows us to query our resources and allows us to identify all the resources we have in our account
* **Rules can be created to flag when something is going wrong. Enforce when a rule is violated**
* Can tell us the history of the environment
* **Resource Timeline allows us to see when a specific resource failed.**
* Linked with Cloud Trail to see the logs
* You can define remediation actions for rules that check for compliance
  + Kick of automation document (run books)
* **Config = Standards**
* **Can create your own custom rules**
* **You can roll up rules to a single region**

## AWS Directory Service - Fully managed version of Active Directory

* Fully managed version of Active Directory
* **Managed Microsoft AD – Entire AD suite built from scratch**
* **AD Connector – Tunnel between AWS environment and on-prem AD**
* Simple AD – Standalone AD powered by Linux

## Cost Explorer - Visualization tool for cost of account.

* Visualization tool for cost of account. Can see them by tags
* Breakdown costs on a service-by-service basis
* See bill by month
* Filter and breakdown data by tag, categories, region, etc.
* **Need to enable resource cost allocation tag to use the tag here**
* **Can get estimated costs**

## AWS Budget – Manage Cost against budgets – Actions – SNS, EventBridge, Lambda

* AWS Budgets lets you set custom cost and usage budgets that alert you when your budget thresholds are exceeded (or forecasted to exceed).
* 4 Types
  + Cost Budget: How much are we spending
  + Usage Budget: How much are we using
  + Reservation Budget: Are we being efficient with RI’s
  + Saving Plans Budget: Is what we’re doing covered by our saving plan
* Can define fixed budget
* **Can define alerts to email and kick off actions**
* **Can setup 2 budgets for free per month**
* **Use Cost Explorer to create budgets**

## AWS Cost and Usage Reports (CUR): Comprehensive service set of cost and usage data available for spending

* **Comprehensive service set of cost and usage data available for spending**
* **Publish billing reports to S3**
* **Breakdown by time span by resource or tags**
* **Daily CSV updates via AWS CUR**
* **Can be integrated with Amazon Athena, Amazon RedShift or Amazon QuickSight**
* **Use Cases**
  + **Use within AWS organizations for entire OU groups or individual member accounts**
  + Track savings plans utilizations, charges and current allocations
  + Monitor On-Demand capacity reservations
  + Break down your AWS data transfer charges (external or inter-region)
  + Deep dive by cost allocation tag

## AWS Compute Optimizer - Checks resource utilization and provides recommendations. Provides Savings Plans, Disabled By Default

* **Analyzes configuration and utilization metrics of your AWS resources and reports current usage optimization and potential recommendations**
* Provides graphs of historical data
* **Works with**
  + **EC2**
  + **Autoscaling Group**
  + **EBS**
  + **Lambda**
* Supported Account Types
  + AWS account without AWS Org setup
  + Single member account within AWS Org
  + AWS Org management account withing recommendations based on the entire org
* DISABLED BY DEFAULT. NEED TO OPTIONALLY OPT IN
* After opting in, turn on activation recommendation preferences
* **Savings Plans:**
  + **Offer flexible pricing for certain resources. Up to 72% savings**
  + **Lower Prices for EC2 Instances regardless of instance family size, OS, tenancy or Regions**
  + **Savings can also apply to Lambda and Fargate**
  + **Sagemaker pricing can be lowered**
* **Terms:**
  + **Commitments required long term 1-year to 3-year contracts**
  + **Pay all-upfront, partial upfront or no upfront**
* Types:
  + Compute Saving: applies to any EC2 compute. 66%
  + EC2 Instance Saving: applies to Ec2 instance of a specific family in specific region. 72%
  + Sagemaker Saving: Applies to Sagemaker instances of any family or size, any region and any component 64%
* How to Use
  + View recommendations with Billing Console
  + Recommendations are automatically calculated
  + Add to cart and purchase directly
  + Applied after existing reserved instances are used and exhausted
  + Applied to account owner first but can be distributed to others

## AWS Trusted Advisor - Auditing Tool, checks Cost, Performance, Fault Tolerance, Service Limits, Security – Actions – SNS, EventBridge and Lambda

* Free Full managed best practice Auditing tool. Scans 5 different parts of your account and look for places where you
* 5 Ways Trusted Advisor Helps:
  + **Cost Optimization: Are you spending money on resources that aren’t needed**
  + **Performance: Are your services configured properly**
  + **Security: Is your AWS architecture full of vulnerabilities**
  + **Fault Tolerance: Are you protected when something fails**
  + **Service Limits: Do you have room to scale**
* Get additional support if you pay for Business Support plan
* Send SNS to send Alert to users
* Trusted Advisor WILL NOT FIX THE PROBLEM
* USE EVENTBRIDGE TO kick off Lambda job to solve problem for you

## AWS Control Tower - Setup Multi-Account environment using best practices, Provides Guardrails

* **Easy way to setup and govern an AWS multi-account environment**
* **Automate account creation and security controls via AWS Services**
* Extends AWS Organizations to prevent governance drift and leverages different guardrails
* Users can provision new AWS accounts quickly, using central admin established compliance policies
* Quickest way to setup a multi account environment using best practices
* Features
  + Landing Zone: Well-architected, multi-account environment based on compliance and security best practices
  + Guardrails: High Level rules providing continuous governance for AWS Environment
  + Account Factory: Configurable account template for standardizing pre-approved configs of new accounts
  + CloudFormation StackSet: Automated Deployment of Templates Deploying repeated resources for governance
  + **Shared Accounts: Three accounts used by Control Tower created during landing zone creation**
* **Types Of GuardRails**
  + **Preventative**
    - **Disallow any violating actions**
    - **Leverages Service Control Policies**
    - **Statuses of enforced on not enabled**
    - **Supported in all regions**
  + **Detective**
    - **Detects and alerts for non-compliant resources within all accounts**
    - **Leverages AWS config rules**
    - **Statuses of clear, in violations, not enabled**
    - **Only apply to certain regions**

Graphical user interface, diagram

Description automatically generated

## AWS License Manager - Simplify managing software licenses

* Simplifies managing software licenses
* Centralized management of licenses across AWS accounts and on-premises environments
* Control Visibility into usage of licenses and enabling license usage limit
* Helps reduce overages and penalties via rule-based controls for consumption
* It is versatile and supports software based on vCPU, physical cores, sockets and number of machines

## AWS Personal Health Dashboard - Visibility of resource performance and availability of AWS Services or accounts – Actions – EventBridge and Lambda

* **Visibility of resource performance and availability of AWS Services or accounts**
* How the health events affect you
* AWS attempts to maintain timeliness and relevant information with the events
* Provides capability to view upcoming maintenance tasks that may affect your accounts and resources
* **Alerts near-instant delivery of notifications and alerts to speed up trouble shooting or prevention**
* **Can automate actions based with EventBridge and Lambda**
* Concepts
  + Health Events: Notifications sent on behalf of AWS services or AWS
  + Account Specific Events: Events specific to your AWS Accounts or AWS organization
  + Public Events: Events reported on services that are public, not specific to any account
  + AWS Health Dashboard: Dashboard showing account and public events
  + Event Type Code: Include the affected services and the specific type of event
  + Event Type Category: Associated category – Will be attached to every event
  + Event Status: Reports if the event is open, closed, or upcoming
  + Affected Entities: Resources affected by the event

## AWS Service Catalog

* Service Catalog: allows organizations to create and manage catalogs of approved IT Services for deployment in AWS
* Can be list of things like AMI’s, servers, software’s, databases, or other preconfigured components
* Centralized: AWS Organization centrally manages IT services and maintain compliances
* End users can be allowed to easily deploy preapproved catalog items within an organization
* All the end users are doing are deploying CloudFormation Templates
* Benefits
  + Standardized Deployments
  + Self Service Product Deployments
  + Access control using AWS IAM for products
  + Versioning: update products automatically

## AWS Proton

* is a service that creates and manages infrastructure and deployment tooling for users as well as **serverless and container-based applications**
* Automate infrastructure as code provisioning and deployments
* Define standardized infrastructure for your serverless and container-based apps
* Use templates to define and manage app stack that contain ALL components
* AWS Proton automatically provisions resource, configures CI/CD pipelines and deploys the code
* Support AWS CloudFormation and Terraform

## AWS Well Architected Tool

* Six Pillars
  + Operational Excellence
  + Reliability
  + Security
  + Performance Efficiency
  + Cost Optimization
  + Sustainability
* **AWS Well Architected Tool measures your architecture against best practices**
* Enables assisting with documenting workloads and architectures
* Guides for making workload follow the Six Pillars
* Intended for specific audiences, tech teams, CTO’s, architecture, and operations teams
* It aids in documentation

# Migration

## Snow Family

* How can we move data to AWS:
  + Internet: Can be potentially slow
  + Direct Connect: Using a direct line can be more secure but not always practical
  + Physical: Physically moving data to the cloud
* Snow family are a set of secure appliances to move petabyte scale data onto the cloud. Provide built in compute capability to run operations in remote locations
* SnowCone:
  + 8TB storage, 4GB memory and 2 vCPU
  + Easily migrate data to AWS after you’ve processed it
* IoT Sensor Integration
  + Perfect for edge computing where space and power are constrained
* Snowball Edge:
  + 48TB to 81 TB in storage
  + Storage, Compute, GPU flavors
  + Perfect for off-the-grid computing
* SnowMobile:
  + 100 PB of storage
  + Designed for exabyte-scale data center migration
  + Data is encrypted at rest and in transit
* General turn around for migration is a week

## Storage Gateway - VM Mount Cloud Filesystem Locally (File Gateway, Storage Gateway, Tape Gateway)

* Is a VM that runs locally
* Is a hybrid cloud storage service that helps merge on-premises resources with the cloud
* Useful for one-time migration or a long-term pairing of the architecture
* **File Gateway:**
  + **Mount Filesystem locally using NFS and SMB**
  + **Allows you to keep most recently used files on prem while moving the remaining onto the cloud**
  + **This is primarily useful for file sharing**

Diagram

Description automatically generated

* + **Primary method to extend on-premises storage**
  + **with AWS Migration**
* **Volume Gateway**
  + **This is primarily for sharing entire drives which may be running VM’s**
  + **iSCSI Mount backs up entire mount**
  + **Cache or store mode**
  + **Allows creation of EBS snapshots with these drives**
  + **Perfect for backup of migration**

Graphical user interface, application

Description automatically generated

* **Tape Gateway**
  + **Replace physical tapes**
  + **Doesn’t change current workflow**
  + **Encrypted communication**

Diagram

Description automatically generated

## AWS DataSync – Copy on-prem files from on prem to S3 EFS or FSx for Windows (ONE TIME MIGRATIONS)

* Is an agent-based solution for migrating on-premises storage to AWS. It allows you to easily move data between NFS and SMB shares and AWS storage solutions
* **Agent based solution that is a secure way to transfer data to S3, EFS and FSx for Windows File Server**
* **Best for one-time migration**
* **Use Storage Gateway for hybrid setup**

## AWS Transfer Family – FTPS, SFTP or FTP (in VPC) setup to transfer files to S3

* Transfer Family allow you to easily move files in and out of EFS using Secure File Transfer Protocol, File Transfer Protocol over SSL, or FTP
* Legacy Solutions
* Can swap out endpoints in legacy application to use Transfer Family
* AWS Transfer for FTP is only supported inside VPC not applicable (FTPS and SFTP)

## AWS Migration Hub - Single Pane to track progress of your migration to AWS

* It is a single pane to track progress of your migration to AWS.
* Integrates with Application Migration Service (AMS), Server Migration Service (SMS) and Database Migration Service (DMS)
* Migration Hub: singular place to discover existing servers, plan your migration efforts and track your migration statuses
* Visualize connection or server and database statuses
* Option to start migration immediately or group servers into application groups first
* **Only discovers and plans your migrations. Doesn’t do the migrations**
* **Steps**
  + **Discover: Find servers and databases to migrate**
  + **Migrate: Connect tools to Migration Hub and begin migrating**
  + **Track: Follow migration statuses**

## AWS Application Discovery Service - Agent Based or Agentless (VMware) Discovery of servers

* **Helps you plan migrations to AWS via collection of usage and configuration data from on-premises servers**
* Integrates with AWS Migration Hub, simplifying migrations and tracking migration statuses
* Helps you easily view and discovered servers, groups them by application and track each application migration
* Discover Types
  + **Agentless:**
    - **OVA file within VMWare vCenter**
    - **Identifies hosts and VM’s in vCenters**
    - **IP and MAC Addresses, resource allocation and hostnames**
    - **Utilization metrics**
  + Agent Based
    - Deploy Application Discovery Agent
    - Each VM and physical server needs this
    - Windows and Linux Support
    - Config data time-series performance info, network connection and OS processes

## AWS Application Migration Service - Lift and Shift Whole Servers (continuous, block-level replication) - Avoiding Cut Over Windows

AWS MGN Service is for migrating applications

* + **Lift and shift entire application**
  + Useful when you need to avoid cutover windows
  + **Replicates source servers (VM’s physical servers) on AWS and automatically launches on AWS to migrate quickly**
  + Features
    - **RTO: Recovery Time Objective: just minutes**
    - **RPO: Recovery Point Objective: sub second range can recover to any point in time**

## AWS Server Migration Service – Lift and Shift Whole Servers (incremental, snapshot-based replication) Deprecated after March 2023 – Hourly Cut Over Windows

* Server Migration Services
  + Automates migrating on-premises servers to AWS Cloud
  + Supports several VM types
  + Incremental replication of VM’s over to AWS AMI’s that can be deployed on Amazon EC2
  + Simplifies complex volume replication
  + Allows you to schedule replication intervals and allows to track progress for chosen server group
  + Wide support for majority of common operating systems
  + Supports incremental testing for small iterative changes
  + Minimizes downtime

## AWS Database Migration Service (AWS DMS) – Lift and Shift DB by DB w/CDC compatibility

* **Migration tool to migrate relational databases, data warehouses, NoSQL Databases and other data stores**
* Migrate data to and from AWS and on-premises. At least one of the target or sources must be on AWS
* Can be for one time migration or ongoing
* Migrate different database engines using Schema Conversion Tool which translates these schemas onto the new platform
* Cloud service, so it provides cost optimization and efficiency
* Steps
  + **DMS is a server running replication software**
  + **Create source and target connections for loading from and to**
  + **Schedule tasks to run on the DMS Server to move data**
  + AWS creates the table and primary keys (if they do not exist on the target)
  + Optionally create your target tables beforehand if desired
  + Leverage the SCT for creating some or all of your tables, indexes, and more
* Source and target data stores are referred to as endpoints
* Important Concept
  + Same Engine Migration
  + Different Engine Migration: DMS and SCT
  + **ONE ENDPOINT MUST BE IN AWS**
* **Leverage the SCT to convert existing database schemas from one engine to another**
* **Converts many types of relational databases including both OLTP and OLAP. Even supports data warehouses**
* **Supports many targets engine types: Redshift, Aurora, RDS**
* **Can convert schemas and databases running on EC2 instances or data stored in S3**
* **3 Types of migration**
  + **Full Load Migration: All existing data moved from sources to targets in parallel**
  + **Full Load and CDC: Full load + CDC to capture changes to source tables**
  + **CDC only: Only replicate changes in source database**
  + **ONLY CDC GUARANTEES TRANSACTIONAL INTEGRITY**
* Issues during migrations
  + **Some data migrations can be Terabytes of information**
  + **Can lead to bandwidth throttling**
  + **In this case make use of Snowball Edge**
  + **CAN USE SCT WITH SNOWBALL EDGE AND THEN INTO S3**
  + **Can then move data from S3 to chosen destination**
  + **CDC compatible**

# Front-End Web and Mobile

## AWS Amplify

* **It is a set of tools for mobile developers to quickly build out full-stack applications on AWS**
* Two services
  + Amplify Hosting: Single page frameworks, React, Angular and Vue
    - Supports Gatsby and Hugo static site generators
    - Allows for separate production and staging environments for front end and back end
    - **Supports Server-Side rendering apps like Next.js**
  + Amplify Studio:
    - Provides easy authentication and authorization
    - Simplified development: very visual
    - Has ready to use components to make backend easier

## Device Farm

* **Application testing service that works on phone and tablets hosted by AWS**
* Two test methods
  + Automated: Upload scripts or use built in tests for automated parallel tests on mobile devices
  + Remote access testing: Swipe, gesture and interact with devices in real time via web browsers

## Amazon Pinpoint

* Customer messaging service to engage with customers: emails, SMS
* Primarily for marketing and business users
* Features
  + Projects: Collections of information, segments, campaigns, and journeys
  + Channels: The platform where you intend to engage your audience
  + Segments: Dynamic or imported; designates which users receive specific messages
  + Campaigns: Engaging specific audience segments using tailored messages
  + Journeys: Customized, multistep engagements
  + Message Template: Content and settings for easily reusing repeated messages
  + Machine Learning: Leverage machine learning models to predict user patterns
* Where is it used?
  + Marketing
  + Transactions
  + Bulk

# Media

## Amazon Elastic Transcoder

* Allows businesses and developers to convert media files from their original source formats into versions that are optimized for various devices, such as smartphones, tablets, and PC’s
* Features
  + Easy to Use: API and SDK compatible or just use AWS Console
  + Scales on demand: transcode small file to very large video files

## Amazon Kinesis Video Streams

* Streaming media content from large number of devices to AWS and then run analytics, machine learning, playback, and other processing consider using Kinesis Video Streams. i.e., Amazon Ring devices
* Feature
  + Elastically scalable to millions of devices, stores, encrypts and indexes files
  + Smart Homes
  + Smart Cities
  + Industrial Automation: LIDAR

# Machine Learning

## Amazon Comprehend, Amazon Kendra, and Amazon Textract

* Comprehend uses natural language processing to help you understand the meaning and sentiment in your text.
* Use Cases
  + Call Center Analytics
  + Index and Search Product Review
  + Legal Brief Management
  + Process Financial Documents
* Kendra allows you to create intelligent search service powered by machine learning. Cal help you index searches between different silos of information (such as S3, fileserver, websites)
* Allowing your enterprise to have all the data intelligently in one place
* Benefits
  + Accelerate Research and Development
  + Improve Customer Interaction
  + Minimize Regulatory Compliance and Risk
  + Increase Employee Productivity
* Amazon Textract uses machine learning to automatically extract text, handwriting, and data from scanned documents
  + Beyond OCR: Can process texts, handwriting, tables and more with no mutual intervention
* User Cases
  + Financial Services: loan applications
  + Health Care and Life Sciences: patient and doctor notes
  + Public Sector: Handwritten tax returns

## Amazon Forecast

* It’s a time series forecasting service that uses machine learning and is built to give your important business insight
* Uses:
  + IoT
  + Analytics
  + DevOps Application

## Amazon Fraud Detector

* **Creates a fraud detection machine learning model**
* Use cases
  + Suspicious online payments
  + Detect new account fraud
  + Prevent trial loyalty program abuse
  + Improve account takeover detection

## Amazon Polly, Amazon Transcribe, and Amazon Lex

* Amazon Transcribe: Converts speech to text
  + Convert audio files to text
* Amazon Lex: Allows you to build conversational interfaces in your application using natural language models
* Chances are when you are talking to an automated bot online it is using Lex
* Amazon Polly: Turns texts into life-like speech
  + Content Creation

## Amazon Rekognition

* **Is a computer vision product that automates the recognition of pictures and videos using deep learning and neural networks**
  + **Content Moderation**
  + Face Detection and Analysis
  + Celebrity Recognition
  + Streaming video event detection

**Amazon SageMaker**

* Running machine learning model on AWS
* Options
  + Ground Truth: Setup and manage labelling jobs for training datasets using active learning and human labeling
  + Notebooks: Jupyter Notebook environment
  + Training: Train and tune models
  + Inference: Package and deploy your ML models at scale
* Two Types of Deployments
  + Two types of deployments
    - Immediate response: Online Usage
    - Late response: Offline usage
* Steps
  + Create a model
  + Create an endpoint configuration
  + Create an endpoint
* Training

Diagram

Description automatically generated

* Modelling

Diagram

Description automatically generated

* Sagemaker Neo customizes machine learning model for specific CPU hardware, such as ARM, Intel, and NVIDIA processors
* Elastic Inference Speeds up throughput and decreases latency of real-time inferences deployed on SageMaker hosted services. Uses CPU inferences instead of GPU inferences to reduce cost
* Provides auto scaling and is highly available

## Amazon Translate

* Automatically translates from one language to another using deep learning
  + Highly accurate and continuously improving
  + Easily Integratable with application