AWS Certified Solutions Architect – Associate (SAA-C02) Exam Guide

AWS SAA Cert Page: <https://aws.amazon.com/certification/certified-solutions-architect-associate/?ch=sec&sec=rmg&d=1>

KEY:

Need All

Need Training Video

Need TD CheatSheet

# Domain 1: Design Resilient Architectures

## 1.1 Design a multi-tier architecture solution

* Determine a solution design based on access patterns.
* Determine a scaling strategy for components used in a design.
* Select an appropriate database based on requirements.
* Select an appropriate compute and storage service based on requirements.
  + See S3 Appendix
  + See EFS Appendix
  + See EBS Appendix

## Design highly available and/or fault-tolerant architectures

* Determine the amount of resources needed to provide a fault-tolerant architecture across Availability Zones.
  + See ELB Appendix
  + See EC2 Appendix
  + See EFS/EBS Appendix
* Select a highly available configuration to mitigate single points of failure.
* Apply AWS services to improve the reliability of legacy applications when application changes are not possible.
* Select an appropriate disaster recovery strategy to meet business requirements.
* Identify key performance indicators to ensure the high availability of the solution.

## 1.3 Design decoupling mechanisms using AWS services

* Determine which AWS services can be leveraged to achieve loose coupling of components.
  + See SQS Appendix
  + See SNS Appendix
  + See Kinesis Appendix
  + Active MQ
    - <https://tutorialsdojo.com/amazon-mq/>
    - Use well established protocols for message queuing
    - Managed Apache ActiveMQ
    - Doesn’t scale as much as SQS/SNS
    - Runs on dedicated machine a
    - Can run in HA with failover
    - Has both SQS and SNS features
* Determine when to leverage serverless technologies to enable decoupling.
  + SNS+SQS Fan Out
    - Push Once to SNS and receive in all SQLS queues that are subscribers
    - Fully decoupled
    - No data loss
    - SQS allows for
      * Data persistence
      * Delayed processing
      * Retries
    - Can add more queues over time
    - Access policy must allow for SNS to write
    - SNS FIFO to SQS FIFO: Fan out
      * Can be used with SQS FIFO
      * Fan out + ordering + deduplication
    - Message Filtering
      * JSON
      * Filter msg sent to SNS topic subscibers
    - Example
      * S3 Events to multiple queues
        + Only one event rule per event type / prefix
        + Can use fan out to send event to many SQS Queues
  + Data Ordering for Kinesis vs SQS FIFO
    - Kinesis uses partition key
      * Same partition key will go to same shard
      * So each partition will be ordered at the shard level
    - SQL FIFO
      * Order in is order out

## 1.4 Choose appropriate resilient storage

* Define a strategy to ensure the durability of data.
  + See S3 Appendix
  + See EFS Appendix
  + See EBS Appendix
* Identify how data service consistency will affect the operation of the application.
* Select data services that will meet the access requirements of the application.
  + See S3 Appendix
  + See EFS Appendix
  + See EBS Appendix
* Identify storage services that can be used with hybrid or non-cloud-native applications.
  + See S3 Appendix
  + See EFS Appendix
  + See EBS Appendix
  + See FSx Appendix

# Domain 2: Design High-Performing Architectures

## 2.1 Identify elastic and scalable compute solutions for a workload

* Select the appropriate instance(s) based on compute, storage, and networking requirements.
  + See EC2 Appendix
  + See Auto Scaling Appendix
* Choose the appropriate architecture and services that scale to meet performance requirements.
  + See EC2 Appendix
  + See Auto Scaling Appendix
* Identify metrics to monitor the performance of the solution.
  + See EC2 Appendix
  + See Auto Scaling Appendix
  + See CloudWatch Appendix

## 2.2 Select high-performing and scalable storage solutions for a workload

* Select a storage service and configuration that meets performance demands.
  + See S3 Appendix
  + See EFS Appendix
  + See EBS Appendix
* Determine storage services that can scale to accommodate future needs.

## 2.3 Select high-performing networking solutions for a workload

* Select appropriate AWS connectivity options to meet performance demands.
* Select appropriate features to optimize connectivity to AWS public services.
* Determine an edge caching strategy to provide performance benefits.
* Select appropriate data transfer service for migration and/or ingestion.

## 2.4 Choose high-performing database solutions for a workload

* Select an appropriate database scaling strategy.
* Determine when database caching is required for performance improvement.
* Choose a suitable database service to meet performance needs.

# Domain 3: Design Secure Applications and Architectures

## 3.1 Design secure access to AWS resources

* Determine when to choose between users, groups, and roles.
  + See IAM Appendix
* Interpret the net effect of a given access policy.
  + See IAM Appendix
* Select appropriate techniques to secure a root account.
  + See IAM Appendix
* Determine ways to secure credentials using features of AWS IAM.
  + See IAM Appendix
* Determine the secure method for an application to access AWS APIs.
  + See IAM Appendix
* Select appropriate services to create traceability for access to AWS resources.
  + See CloudTrail Appendix

## Design secure application tiers

* Given traffic control requirements, determine when and how to use security groups and network ACLs.
  + See VPC Appendix
* Determine a network segmentation strategy using public and private subnets.
  + See VPC Appendix
* Select the appropriate routing mechanism to securely access AWS service endpoints or internet-based resources from Amazon VPC.
  + See VPC Appendix
* Select appropriate AWS services to protect applications from external threats.
  + See WAF Appendix
  + See Shield Appendix

## 3.3 Select appropriate data security options

* Determine the policies that need to be applied to objects based on access patterns.
  + See IAM Appendix
* Select appropriate encryption options for data at rest and in transit for AWS services.
  + See Amazon KMS appendix
* Select appropriate key management options based on requirements.
  + See Amazon KMS appendix

# Domain 4: Design Cost-Optimized Architectures

## 4.1 Identify cost-effective storage solutions

* Determine the most cost-effective data storage options based on requirements.
  + See S3 Appendix
  + See EFS Appendix
  + See EBS Appendix
* Apply automated processes to ensure that data over time is stored on storage tiers that
* minimize costs.
  + See S3 Appendix
  + See EFS Appendix
  + See EBS Appendix

# 4.2 Identify cost-effective compute and database services

* Determine the most cost-effective Amazon EC2 billing options for each aspect of the workload.
  + See EC2 Appendix
* Determine the most cost-effective database options based on requirements.
  + See the DB Appendices
* Select appropriate scaling strategies from a cost perspective.
  + See Auto Scaling
* Select and size compute resources that are optimally suited for the workload.
  + See EC2 Appendix
* Determine options to minimize total cost of ownership (TCO) through managed services and serverless architectures.

## 4.3 Design cost-optimized network architectures

* Identify when content delivery can be used to reduce costs.
  + See CloudFront Appendix
  + See S3 Appendix
* Determine strategies to reduce data transfer costs within AWS.
  + See S3 Transfer Acceleration
  + See Global Accelerator
* Determine the most cost-effective connectivity options between AWS and on-premises environments.
  + See Site VPN
  + See Direct Connect
  + See DataSync
  + See DB Migration

# Appendix

Which key tools, technologies, and concepts might be covered on the exam?

The following is a non-exhaustive list of the tools and technologies that could appear on the exam. This list is subject to change and is provided to help you understand the general scope of services, features, or technologies on the exam. The general tools and technologies in this list appear in no particular order. AWS services are grouped according to their primary functions. While some of these technologies will likely be covered more than others on the exam, the order and placement of them in this list is no indication of relative weight or importance:

* Compute
* Cost management
* Database
* Disaster recovery
* High availability
* Management and governance
* Microservices and component decoupling
* Migration and data transfer
* Networking, connectivity, and content delivery
* Security
* Serverless design principles
* Storage

# AWS services and features

## Analytics:

### Amazon Athena

* Serverless query service perform analytics against S3 objects
* Use standard SQL
* Supports output
  + CSV
  + JSON
  + ORC
  + AVRO
  + Parquet
* Billing
  + 5.00 per TB data scanned
  + Compressed or columnar data = less data to scan = less cost
* Use Cases (note these must all reside in S3)
  + BI
  + Analytics
  + Reporting
  + Analyze and query
    - VPC Flow logs
    - ELB Logs
    - CloutTrail trails

### Amazon Elasticsearch Service (Amazon ES)

* Now called OpenSearch
* Allows you to search a DB for any data including partial fields
* Can be a cluster
* Kibana
  + Visualization
* Logstash
  + Log ingestion

### Amazon EMR

* Elastic Map Reduce
* Helps create Hadoop Clusters to analyze huge amounts of info
* Big Data
* Hundreds of EC2 instances
* Provisions and configs cluster
* Auto scaling and integrated with spot instances

### AWS Glue

* Managed ETL service
  + Extract, transform, load
* Prepare and transform data for analytivcs
* Fully serverless
* Data catalog
  + Catalog of datasets
  + Metadata

### Amazon Kinesis

* <https://tutorialsdojo.com/amazon-kinesis/>
* Real-time Streaming
  + Collect, process and analyze streaming data in real time
* Data such as
  + App logs
  + Metrics
  + Website click streas
  + IOT telemetry
  + Etc
* Kinesis Data Streams
  + Capture, Process, And store data streams
  + Steams contain shards
    - More shards mean more throughput
    - Used to scale
    - 1000 msg/sec
  + The following can use data streams (Producers)
    - Applications
    - Client
    - SDK, KPL (kinesis Producer Library)
    - Kinesis agent
  + Consumers
    - Apps ( KCL (Kinesis Consumer Library), SDK )
    - Lambda
    - Kinesis Data Firehose
    - Kinesis Data Analytics
    - Throughput
      * Shared
        + 2 MB/Sec per shard across all consumers
      * Enhanced
        + 2 MB/Sec per shard per consumer
  + Billing
    - Per shard
  + Retention
    - 1 day default up to 365 days
  + Ability to reprocess ( replay ) data
  + DynamoDB
    - Used to store Shard Info
    - Shard info will expire if no GetRecord call within 5 minutes
    - If expire immediately, the DynamoDB write capacity needs to be increased
  + Data cannot be deleted once in Kinesis (immutability)
  + Data that shares same partition goes to same shard (ordering)
  + Requires manual administration of shards
* Kinesis Data Firehose
  + Load data streams into AWS data stores and analytics tools
  + Store data to target locations
  + Producers
    - App
    - Client
    - SDK, KPL
    - Kinesis Agent
    - Kinesis Data Streams
      * Note: when data streams is the source, PutRecord and PutREcoredBatch are disabled and Kinesis Agent cannot write to firehose delivery stream directly
    - Amazon CloudWatch ( logs and events)
    - AWS IOT
  + Lambda functions can be used to transform data in firehose
  + Batch writes out
  + Storage locations
    - S3
    - Amazon Redshift
      * Copy through S3
    - Amazon ElasticSearch
  + Fully managed
  + No admin
  + Auto scaling
  + Serverless
  + Billing
    - Data through firehose
  + Near real time
  + Can send all or just failed to S3 bucket
* Kinesis Data Analytics
  + Analyze data streams with SQL or Apache Flink
  + Sources
    - Kinesis Data Streams
    - Kinesis Data Firehose
  + Querys during data analyze
  + Destination
    - Kinesis data streams
    - Kinesis Firehose
  + Fully managed
  + Serverless
  + Auto scaling
  + Realtime
  + Pay for throughput
* Kinesis Video Streams
  + Capture, Process , and store video streams

### Amazon QuickSight

## **AWS Billing and Cost Management:**

### **AWS Budgets**

### **Cost Explorer**

* Manage AWS Cost
* Custom reports
* Allows you to choose a savings plan to lower your bill
* Forecast usage up to 12 months

## **Application Integration:**

### **Amazon Simple Notification Service (Amazon SNS)**

* Publish / Subscription model
* OverView
  + Send message to SNS topic (publish)
  + Subscription gets message from topic
  + Each subscriber will get message
  + 10000000 subscriptions per topic
  + 100000 topics limit
  + Subscibers
    - SQS
    - HTTP/HTTPS (with delivery retries – how many times)
    - Lambda
    - Email
    - SMS messages
    - Mobile notification
  + Publisher
    - Cloudwatch ( alarms )
    - ASG notifications
    - S3 ( bucket events )
    - CloudFormation (Upon state changes – failed to build, etc )
    - Etc
  + Publishing
    - Topic Publish
      * Create topic
      * Assign subscibers
      * Pub to topic
    - Direct Publish
      * Create platform app
      * Create platform endpoint
      * Publish to endpoint
      * Works with Google GCM, Apple APNS, Amazon ADM, etc
  + Encryption
    - In-flight using HTTPS
    - At rest using KMS keys
    - Client sid encryption if the client wants to perform themselves
  + Access Controls
    - IAM policies to regulate access to the SNS API
  + SNS Access Policies
    - Similar to S3 buckets
    - Useful for cross-account access to SNS topics
    - Useful for allowing other services to write to an SNS topic
* SNS FIFO
  + With SQS FIFO strict message ordering and deduplication
  + Data consistency in near real time

### **Amazon Simple Queue Service (Amazon SQS)**

* Queue model
* Buffer between
  + producer (send messages) and
  + consumer (poll messages)
    - will also delete messages from queue after reading
* Types of Queues
  + Standard queue
    - Decouple applications
    - Unlimited throughput
      * Send as many messages per second as you want
    - Unlimited num of messages in queue
    - Message Visibility timeout
      * When polled by consumer message becomes invisible to other comsumers
      * Invisible for 30 seconds to be processed
      * If not deleted then message back in queue
      * ChangeMessageVisibility API
        + Consumer can set visibility if it needs more time to process
    - Message Expiration
      * will stay in queue for 1 minuteto 14 days ( default 4 days )
    - Low latency
    - Less than 256 kb per message (configurable)
    - Can have duplicate messages
      * At least once delivery
    - Out of order messages
      * Best effort
    - Types of standard queues
      * Dead Letter Queue
        + Can set threshold of times a message is sent back to queue

MaximumRecieves threshold

If exceeded message sent to dead letter queue

* + - * + If not process letter goes to this queue
        + Useful for debugging
        + Should set high expiration date
      * Delay Queue
        + Delay messages

Up to 15 minutes

* + - * + DelaySeconds Parameter

Will override delay

* + - Log Polling
      * If no messages in queue, consumer can wait
      * This reduces number of calls to queue
      * Reduces latency
      * Increases efficiency
      * Can set 1 sec to 20 sec ( 20 preferable )
      * Can Configure
        + At queue level
        + Or via API

WaitTimeSeconds

* + - Request REsponse
      * Producer ( requester ) sends item to queue, Consumer (responder) processes and sends response back via another queue with correlation ID
      * SQSTemporyQueueClient
        + Sets up this pattern
        + Uses virtual queues
  + FIFO Queue
    - First in first out
    - Ordered messages
    - Limited throughput
      * 300 msg/s
      * Batchin 3000 msg/s
    - Exactly once send
    - Process in order
    - Name must end in .FIFO
* Cloudwatch metrics
  + Queue Length
    - How many messages in queue
    - Can be used to trigger alarm to increase instances in ASG
* Security
  + Encryption
    - Inflight using HTTPS API
    - At rest using KMS keys
    - Client-side encryption if client wants to encrypt/decrypt itself
  + Access Controls
    - IAM policies to regulate access to SQS API
  + SQS Access Polices
    - Similar to S3 bucket policies
    - Useful for cross-account access to queues
    - Useful for allowing other services to write to queues
* Auto Scaling Groups
  + Custom metric from EC2 instances pushed to cloudwatch
    - Queue length / Number of instances
  + Alarm on threshold of metric will triger scaling policy on ASG

## **Compute:**

### Amazon EC2

* AMI
  + Amazon Machine Image
  + Customized EC2 Instance
    - Sofware
    - Config
    - OS
    - Monitoring
    - Etc
  + Allows faster boot / config time because all is prepackaged
  + Built in one region and can be copied to other region
  + Types
    - Public AMI
      * AWS provided
    - Build your own
      * Used tools to create AMI
      * Process
        + Start EC2 instance and customize
        + Stop instance (optional)
        + Build AMI (this will also create EBS snapshots)
        + Luanch new instance from AMI
    - AWS Marketpalce
      * Use someone elses AMI
      * Potential cost
* EC2 Instance Store
  + High permormance
  + Not persistent (Ephemeral)
    - Gets deleted when Instance is stopped or terminated
* EC2 User Data
  + Used to run scripts at first start
  + Runs as root user
* EC2 Instance Types
  + Naming conventions
    - M5.2xlarge
      * M = instance class
      * 5 = generation
      * 2xlarge = size within class
  + Instance Class
    - General purpose
      * Diversity of workloads such as web severs or code repo
      * Balance between computer, memory, networking
      * T class
    - Computer Optimized
      * Compute intensive workloads
      * High performance
      * Use case
        + Batch processing workloads
        + Media transcoding
        + High performance web servers
        + High performance computing (HPC)
        + Scientific modeling and machine learning
        + Dedicated gaming servers
      * C class
    - Memory Optimized
      * Large sets in memory
      * Use cases
        + High performance relational/non-relational DB  
          distributed web scal cache stores
        + In-memory DB optimized for BI
        + Applications performing real-time processing or big unstructured data
      * R Class (ram)
      * X1
      * Z1
    - Storage Optimized
      * Storage intensive tasks require high, sequential Read / Write access to large data sets on local storage
      * Use Cases
        + High frequency online transaction processing (OLTP)
        + Relational and NoSQL DB
        + Cache for in-memory DB
        + Data warehousing apps
        + Distributed File Systems
      * I class
  + S
* Security Groups
  + Control how traffic is allowed into and out of EC2 instances
  + Only contain Allow Rules
  + Rules can reference Ips or Other Security Groups
  + Firewall on instance
  + Regulate
    - Access to ports
    - Authorized IP ranges
    - Control of inbound network
    - Control of outbound network
  + Can be attached to multiple instances
  + Locked down to Region/VPC
    - Must create new SG in another region or VPC
  + Lives outside of instance
  + If app not accessible ( timeout ) check SG
  + By default
    - All inbound is blocked
    - All outbound is allowed
* SSH
  + Secure shell to Linux instance
  + Requires key downloaded from your AWS Account
  + Troubleshooting
    - Key security
      * Owner = you
      * Remove inheritance and remove all except you
    - Connection timeout
      * Check SG to allow SSH
    - Still Connection timeout
      * SG is correct, then corp firewall does not allow SSH
    - SSH not work on Windows
      * Use Putty
    - Connection Refused
      * Restart instance
      * Make sure using Amazon Linux 2
* EC2 Instance Connect
* IAM Roles for EC2
  + Can attach roles to instance to give it aws credentials to access AWS resources
* EC2 Launch types
  + On-Demand
    - Short workload
    - Predictable pricing
    - Pay for what you use
    - High cost but no upfront payment
    - No long term
  + Reserved
    - Up to 75% discount
    - Purchase options
      * No up front
      * Partial upfront
      * All upfront
    - Minimum 1 Year or 3 years
    - Useful for steady state apps ( DB )
    - Types
      * Reserved Instance
        + Long workloads
      * Convertible Reserved Instances
        + Long workloads
        + Flexible instance types
      * Scheduled Reserved Instances
        + Schedule when you will need the instance
        + Deprecated
  + Spot Instances
    - Up to 90%
    - Short workloads
    - Cheap
    - Can lose instance
      * If someone will pay more for instance than you
      * 2 minute grace period warning to stop or terminate instance
    - Use when you can restart workload and recreate data
      * Batch jobs
      * Data analysis
      * Image processing
      * Any distributed workload
      * Workloads with a flexible start / end time
    - DO NOT USE FOR CRITICAL WORKLOADS or DB
    - Spot Block
      * Block spot instance during a specified time from ( 1-6 hours )
    - Terminate spot instance
      * Must cancel persistent mode spot request and then terminate instance
    - Persistent mode Spot instanc e
      * If stop Spot instance, it will be restarted
    - Spot Fleet
      * Set of spot instances and optionally on-demand instances
      * Will tryp to meet target capacity with price constraints
        + Launch from multiple launch pools
        + Stop launching when capacity or max cost reached
      * Strategy
        + Lowest price

Launch from pool with lowest price

Cost optimization

Short workload

* + - * + Diversified

Distributed across all pools

Great for availability

Long workloads

* + - * + Capacity Optimized

Pool with optimal capacity for number of instances

* + Dedicated Host
    - Book an entire physical server
    - Control instance placement
    - Address compliance requirements
    - Use server-bound software licenses
    - 3 year reservation
  + Dedicated Instances
    - Instances running on HW dedicated to you
    - May share HW with other instances in same account
    - No control over instance placement
      * Can move after start/stop
* Elastic IP
  + Instances get new IP when they stop and start
  + Elastic IP = static IP
  + Attach to instance so it keeps same IP
  + Can only have 5 within account
  + Try to avoid Elastic IP
  + Instead assign DNS name to instance
  + Billed when not attached to instance
* Placement Group
  + Instance placement strategy
  + States how place instance on hardware
  + Strategies
    - Cluster
      * Low latency group
      * All in single AZ
      * All on same rack / hardware
      * If HW failure then all instances fail
      * Use case
        + Big data job
        + Low latency jobs
    - Spread
      * Spreads instances across hardware
      * Max 7 instances per group per AZ
      * Critical applications
      * Minimize failure risk
    - Partition
      * Spreads instances across many different partitions
      * Each partition is a different racks with in AZ
      * Up to 7 partitions per AZ
      * Scales to 100s of instances per group
      * Metadata contains which partition instance is in
      * Use case
        + Hadoop
        + Cassandra
        + Kafka
* Elastic Network Interface (ENI)
  + Virtual network card
  + Logical component in VPC
  + ENI Attributes
    - Primary Private IPv4
    - One or more secondary IPv4
    - One Elastic IP per private IPv4
    - One Elastic IP public IPv4
    - One or more security Groups
    - MAC Address
  + Can have more than one per instance
  + Can be created idependently of instance
  + Can be moved between instances
    - Failover
  + Bound to AZ
* EC2 Instance Hibernate
  + In memory state is reserved
    - RAM written to EBS encrypted root volume
  + Restart is much faster
  + Use cases
    - Long running processing
    - Saving ram state
    - Service that take time to initialize
  + Supported instances
    - C3,C4,C5,M3,M4,M5,R3,R4,R5
  + Ram size less than 150 MB
  + Linux and windows
  + Root volume must be EBS and encrypted and enough free space for RAM
  + Only on-demand and reserved instances
  + Can only Hibernate for 60 days
* EC2 Nitro
  + New underlying virtualization tech for next gen EC2 instances
  + Better performance
  + Better network
  + Higher speed IOPS
  + Beter security
  + Types
    - Gen 5 and above
* vCPU
  + to find out the number of vCPU your instance has, look at the number of cores the instance has and multiply by 2 (2 hyperthreads per core)
  + can reduce the number of cores and threads per core on an instance
  + specify during launch
* capacity reservation
  + ensure you have enough capacity
  + Manual or planned end date for reservation
  + No need for 1 or 3 year commitment

### AWS Elastic Beanstalk

* Developer centric view of deploying and application
* Managed service that uses Other AWS services to deploy app
  + Auto handles
    - LB
    - Capacity
    - Scaling
    - Monitoring
    - Etc
* Components
  + Application
    - Collection of beanstalk components
      * Environments
      * Versions
      * Configurations
  + Application version
    - Iteration of code
  + Environment
    - Collection of AWS resources running an application version
    - Only one running version per environment
    - Tiers
      * Web server environment Tier
        + ELB to ASG to Instances
      * Worker Environment Tier
        + SQS to ASG to Instances
        + Scales based on SQS messages
        + Can push messages to SQS queue from another web server tier
    - Create multiple environments (dev,test, prod)
* Supports multiple platform languages
  + Can create custom platform

### Amazon Elastic Container Service (Amazon ECS)

* Launch docker containers on AWS
* You provision and maintain the infrastructure (EC2 instances)
* AWS takes care of starting and stopping containers
* Has integration with load balancer
* Launch types for ECS
  + Amazon EC2 Launch type for ECS
    - Multi az
    - Auto scaling group
    - ECS agent
      * Register to ECS cluster
      * Runs ECS Tasks
  + Fargate
    - See Fargate Appendix
* ECS Tasks
  + Run docker container
  + Requires IAM Role
* IAM Roles for Containers in AWS
  + For EC2 Instance
    - Used by ECS Agent
    - Makes API calls to ECS Service
    - Send container logs to cloudwatch logs
    - Pull docker image from ECR
    - Reference sensitive data in Secrets manager or SSM Parameter Store
  + ECS Task Role
    - Allow each task to have a specific role
    - Use different roles for the different ECS services you run
    - Task role defined in task definition
* ECS Data Volumes
  + EFS File Systems mounted to containers (Tasks)
* ESC Services
  + Multiple ECS tasks
  + Load balancer in front
    - EC2 launch type
      * Run task with dynamic port assignment
      * ALB uses dynamic port mapping to find which ports to forward requests to
      * EC2 SG must allow any port from ALB security group
    - Fargate
      * Each task has an ENI
      * ALB will talk to each ENI over the port
      * ENI SG must allow ALB SG access via task port
* Eventbridge can invoke ECS Task
  + Event trigger running container tasks
* ECS Scaling
  + ECS Service CPU Usage (avg)
    - Both EC2 and Fargate
    - Cloudwatch alarm on metric triggers scaling event
    - Might need to scale the ASG for the EC2 Instances as well
  + SQS queue length
    - Same setup
* ECS rolling updates
  + Can control how many tasks can be started and stopped and in which order
* AWS ECR
  + Elastic Container Registry
  + Repo for container images

### Amazon Elastic Kubernetes Service (Amazon EKS)

* Launch and manage Kubernetes cluster on AWS
* Management of Containers
* Launch mode
  + EC2
  + Fargate
* Use if you already know kubernetics
* Cloud agnostic
* pods

### Elastic Load Balancing

* OverView
  + Forwards traffic downstream to multiple servers
  + Expose single DNS access point
  + Handle downstream failures
    - Healthchecks
      * Verify if instance is working and not send traffic if failed
      * Uses ports and routes
      * 200 ok means healthy
  + SSL
  + Sticky Session ( Session Affinity )
    - Same client is always redirected to same instance behind LB
    - Works for ALB and CLB
    - Uses cookies
      * Has TTL
      * Application based cookie
        + Custom cookie

Generated by the target

Can include any custom attributes required by the application

Name must be specified individually for each target group

Do not use

AWSALB

AWSALBAPP

AWSSALBTG

* + - * + Application cookie

Genterated by LB

Cookie name

AWSALBAPP

* + - * Duration Based Cookies
        + Generated by LB
        + Cookie name

AWSALB for ALB

AWSELB for CLB

* + - Session data
  + Cross Zone Load Balancing
    - HA Across Zones
    - Each LB distributes evenly across all registered instances in AZ
    - For ALB
      * Always on can’t disable
      * Not charge for inter AZ Data
    - For NLB
      * Disable by default
      * Pay for inter AZ Data
    - For CLB
      * Disabled by default
      * No charge for inter AZ data if enabled
  + Separate public traffic from private traffic
  + Manage
* Load Balancer Types
  + Classic
    - Deprecated
    - TCP Layer 4
    - HTTP and HTTPS ( Layer 7 )
    - Health checks TCP or HTTP bases
    - Fixed hostname
      * Xxxx.reagion.elb.amazonaws.com
  + Application Load Balancer
    - HTTP ( Layer 7 ) Only
    - Route to multiple applications across machines
      * Target Groups
        + Routes based on path in url

Example.com/users

* + - * + Routes based on hostname in url

Users.exampl.com

* + - * + Route based on query string and headers

Example.com/users?id=123

This is how you can direct based on type of device

* + - * + Can be

EC2 instances (can be managed by ASG) HTTP

ECS tasks ( managed by ECS ) HTTP

Lambda functions HTTP converted to JSON

IP Address – must be private

* + - Multiple applications on same machine
      * Containers
    - Support HTTP/2 and Websockets
    - Supports redirection
    - Great for microservices and container based apps
    - Port mapping
      * Redirect to a dynamic port in ECS
    - X-Forwarded-For
      * Header with client IP
      * Application cant see the ip except for header
    - X-Forwarded-Port
      * Client port
    - X-Forwarded-Proto
      * Client Protocol
  + Network Load Balancer
    - Layer 4 TCP/UDP
    - Millions of request per second
    - Low latency
    - One static IP per AZ
    - Supports assigning Elastic IP (static)
    - Helpful for whitelisting
    - When to use Extreme performance TCP/UDP
    - Can send to (target Groups)
      * EC2 instances
      * IP Address – must be private
      * Application load balancer
  + Gateway Load Balancer
    - Deploy, scal and manage fleet of 3rd party network virtual appliances in AWS
      * Firewall
      * Intrusion detection
      * Etc
    - Layer 3 ( network layer) IP Packets
    - Combines
      * Transparent Network Gateway
        + Single entry/exit
      * Load balancer
        + Distributes load
    - Uses GENEVE protocol port 6081

### AWS Fargate

* No instances to provision
* Serverless
* Each ECS Task gets ENI (with IP)
* Otherwise same as ECS

### AWS Lambda

* Serverless Functions
* Limits
  + Execution
    - Memory Allocation
      * 128 MB – 10 Gb
    - Limited by how long they can run ( 15 minutes )
    - Environment Variables
      * 4KB
    - Disk Capacity
      * Function Container
      * 512 MB
    - 1000 concurrent executions
  + Deployment
    - Size Compressed = 50 MB
    - Size Uncompressed (code + dependencies) = 250 MB
      * Can use /tmp directory after startup to load files
    - Size of environment var = 4 KB
* Run on Demand
* Automatically scaling
* Billing
  + Pay for requests and compute time
* Integrated with AWS Services
  + API Gateway – Rest API
  + Kinesis
  + DynamoDB – Trigger
  + S3 - Trigger
  + CloudFront – Lambda Edge
  + EventBridge – Trigger
  + Cloudwatch Logs
  + SNS – react to messages in topic
  + SQS – consumer to process messages
  + Cognito – react to user login
* Programing laguages
  + Node.js
  + Python
  + Java
  + C# (.NET Core)
  + Golang
  + C# / Powershell
  + Ruby
  + Custom Runtime API
  + Lambda Container Image
    - Container image must implement Lambda runtime API
    - ECS / Fargate is preferre3d for running arbitrary Docker Images
* CLoudwatch monitoring
* Can get more resources per function
  + Upto 10GB RAM
  + More RAM increased CPU and Network
* Lambda Edge
  + Deploy lambda functions alongside Cloudfront CDN
  + Build more responsive applications
  + Don’t manage server lambda is deployed globally
  + Customize CD Content
  + Pay for what you use
  + Use lambda to change CloudFront Requests and responses
    - Viewer Request
      * Function modifies before cloudfront processes
    - Origin Request
      * Function modifies before cloudfront sends request to origin
    - Origin Respons
      * Modify when Cloudfront recieves response from origin
    - Viewer Response
      * Modify before cloudfront sends to user
  + Can generate responses to viewers without ever sending the request to the origin
  + Use cases
    - Website security and privacy
    - Dynamic web application at the edge
    - Search engine optimization
    - Intelligently route across origins and data centers
    - BOT mitigation at edge
    - A/B Testing
    - User authentication and authorization
    - User prioritization
    - User Tracking and analytics

## **Database:**

### Amazon Aurora

* Proprietary AWS DB
* Compatible with Postgress and MySQL
* Cloud Optimized
  + 5x performance over MySQL
  + 3x over Postgres
* Storage grows from 10GB to 128 TB
* Upto 15 replicas
  + Fast replication
* Instantaneous Failover
  + 1 master and upto 15 replicas
  + Cross region replication
    - Useful for DR
    - Simple to enable
* HA Native
  + 6 copies of data across 3 AZ
  + Only needs 4 of 6 for writes
  + Only needs 3 of 6 for reads
  + Self healing with peer to peer replication
  + Storage striped across 100s volumes
* Costs 20% more but more efficient
* DB Cluster
  + Master
    - Only one
    - Write to master
    - Writer Endpoint points to master.
      * App points to writer endpoint
  + Read Replica
    - Upto 15
    - Auto scales
    - Reader Endpoint
      * Connects to read replicas
      * App points to reader endpoint
      * Load balances among read replicas
* Features
  + Automatic Failover
  + Backup and recovery
  + Isolation and security
    - Similar to RDS
  + Industry compliance
  + Push button scaling
  + Automated patching with zero downtime
  + Advanced monitoring
  + Routine maintenance
  + Backtrack
    - Restore data at any point of time without using backups
  + Custom Endpoints
    - Read replicas use different instance size
    - Custom endpoint can point to larger instance to run analytics for example
* Aurora Serverless
  + Automated DB instantiation and auto scaling
  + Good for infrequent or unpredicatbale work loads
  + No capacity planning
  + Pay per second
* MultiMaster
  + Immediate failover for write node
  + Every node does RW vs only one
* Global Aurora
  + 1 primary Region ( read/write )
  + Up to 5 secondary (Read only) regions
    - Replication lag less than 1 second
  + Up to 16 read replicas per secondary region
  + Decrease read latency across globe
  + Promoting another region RTO < 1 minute
* Integrates with Machine Learning
  + Add machine learning preditions to apps via SQL
  + Supported Services
    - Sagemaker
    - comprehend

### Amazon DynamoDB

* Fully managed
* HA
* Replication across multi-AZ
  + Auto replicated across multi AZ in a region
  + Provides built in HA and Durability
* NoSQL DB
  + Not relational
  + This means the schema can be updated easily without downtime unlike RDS
* Scals to massive workloads
* Fast and consistent
* Integrates with IAM
* DynamoDB Streams
  + optional
  + Event driven programming
  + Ordered stream of item level modifications (create/update/delete)
  + Stream records can be
    - Sent to Kinesis data streams
    - Read by AWS Lambda
    - Read byKinesis Client Library Applications
  + 24 hour data retention
  + Use cases
    - React to changes in real-time
    - Analytics
    - Insert into derivative tables
    - Insert into Elastic Search
    - Implement Coss-region replication
* Auto scaling
* Basics
  + Made of Tables
    - Collection of data
    - Schemaless
    - 256 tables per region limit
  + Each table has a primary key
    - Determined at creation time
    - Simple Primary key
      * Partition key
      * Only one attribute
    - Partition ID + sort ID
      * Composite primary key
      * Two attributes
    - All items with the same partition key are stored together in sorted order by sort key value
  + Secondary key
    - Provides more querying flexibility
    - Can have multiple secondary indexes
    - Global secondary index
      * Partition key and sort key can be different from those on table
    - Local Secondary Index
      * Same partion key as the table but different sort key
  + Items
    - Collection of attributes
    - Attributes for each item
      * Can be added over time
      * Schema update
      * Can nest attributes
  + Infinite number of rows (items)
    - Max Item size 400 KB
  + Data types
    - Scalar Types = String, Number, Binary , Boolean, Null
    - Document Types = List, Map
    - Set Types = String Set, Number Set, Bianary Set
* Read/Write Capacity Modes – how to manage tables capacity ( throughput)
  + Provisioned Mode ( default )
    - Specify read/write per second
    - Know capacity before hand
    - Pay for provisioned Read Capacity Units (RCU) and Write Capacity Unites (WCU)
    - Possible to auto scale RCU and WCU
  + On-Demadn Mode
    - Read/writes automatically scale up/down with workload
    - No capacity planning
    - Pay for what you use.
    - More expensive
    - Great for unpredictable workloads
* DynamoDB Accelerator (DAX)
  + Fully manged
  + HA
  + In memory cache
    - Read thru / write thru
    - Read / write cache
  + Fixes read congestion
  + Microsecond latency
  + Doesn’t require application logic modification
  + Can be clustered
  + 5 minute TTL
* DynamoDB Global Tables
  + Two way replication across multiple reagions
  + Read/write to any region
  + Require DB Streams
  + Tables can be auto replicated across tow or more reagions
  + Multi master
* DynamoDB TTL
  + Auto delete items after an expiry timestamp
* Indexes
  + Allow to query attributes other than primary key
  + Global secondary Indexes
    - GSI
  + Local Secondary Indexes
    - LSI
* Transactions
  + Transaction is written to multiple tables or no tables
* Encryption
  + At rest
* Backups
  + On demand backups
  + Point in time recovery for DynamoDB tables
    - Upto 35 days ago

### Amazon ElastiCache

* <https://tutorialsdojo.com/amazon-elasticache/>
* in-memory databases with really high performance, low latency
* reduce read instensive workload off DB
* helps make application stateless
* Security
  + Do not support IAM Auth
  + Iam Policies on elasticache are only for API
* Patterns
  + Lazy Loading
    - All read data is cached
    - Data can become stale
  + Write Through
    - Adds or update data in the cache when written to DB
    - No stale data
  + Session Store
    - Store temp session data in cache
    - Uses TTL
* Redis
  + Multi-AZ with auto failover
  + Read Replicas to scale reads
  + Data durability using AOF Persistence
  + Backup and restore
  + Redis Auth
    - Set password/token when you create redis cluster
    - Support SSL
  + Use cases
    - Gaming leader boards
      * Redis sorted sets
        + Guarantee both uniqueness and element ordering
        + Each time new element is added its ranked in real time then added in correct order
* MemCached
  + Multi-Node partitioning of data (sharding)
  + No HA or replication
  + Non Persistent
  + No backup and restore
  + Multi threaded
  + Auth
    - SASL-based auth (advanced)

### Amazon RDS

* https://tutorialsdojo.com/amazon-relational-database-service-amazon-rds/
* Relational Database Service
* Managed DB Service
  + Auto provisioning
  + OS Patching
  + Continuous backups
    - Automated
    - Daily full backup
    - Transaction log backup every 5 minutes
      * Point in time Restore
    - 7 day retention (max 35)
  + Monitoring Dashboard
  + Read replicas for improved read performance
    - Up to 5 replicas
      * Same AZ, Cross AZ, Cross Region
    - Async replication
      * Eventually consistent
    - Can be promoted to own DB
    - Application must update the connection string to leverage read replicas
    - Billing
      * Within same region replicas traffic does not cost
      * Cross region cost
    - Read replica can have different storage type than master
  + Multi-AZ
    - Disaster Recovery
    - Sync replication
    - One DNS name so auto failover for app connections
    - Not used for scaling
    - Can setup read replica as Multi AZ
    - Setup Multi AZ
      * Zero downtime
      * Modify DB
    - Can modify single AZ to be MultiAZ
    - Primary switches to Secondary if
      * AZ outage
      * Primary DB instance fails
      * DB Instances ser5ver type is changed
      * OS of the DB instance is undergoing software patching
      * Manual failover
    - Route 53 CNAME is updated during failover
  + Maint windows for upgrades
  + Scaling – vertical and horizontal
    - Auto scaling
      * Size of DB
      * Must set Maximum Storage Threshold to limit how big
      * Scales if
        + Free storage < 10%
        + Low storage last at least 5 minutes
        + 6 hours have passed since last scale
      * Useful for unpredictable workloads
  + Backed by EBS
  + Cannot SSH to OS
* Uses SQL as query language
* Types
  + Postgres
  + MySQL
  + MariaDB
  + Oracle
  + Microsoft SQL Server
  + Aurora
* DB Snapshots
  + Type of backup
  + Manually triggered by user
  + Unlimited Retention period
* Encryption
  + Encrypt Master and read replicas with AWS KMS
    - Defined at launch time
    - If master is not encrypted, the read replicas cannot be encrypted
  + TDE available for Oracle and SQL Server
  + Inflite
    - SSL Certificates
    - To enforce SSL
      * Postgress
        + Rds.force\_ssl = 1 in AWS RDS Console
      * MySQL
        + Within DB: GRANT USAGE ON \*.\* TO ‘mysqluser’@%’ REQUIRE SSL;
  + Encryption Operations
    - Encrypting RDS Backup
      * Snapshots of un-encrypted RDS Databases are un-encrypted and vice versa
      * Copy snapshot to an encrypted one (enable during copy)
  + Network Security
    - RDS datbases are usually deployed within a private subnet
    - Security groups control what can communicate with RDS
  + Access Management
    - IAM Policies control who can manage RDS
    - Traditional username / password to login to database
      * MYSQL and PostGreSQL
        + Can use IAM based auth
        + Uses auth token ( which has TTL )
* Security
  + IAM Database Authentication
    - <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.IAMDBAuth.html>
    - For
      * MariaDB
      * MySQL
      * PostreSQL
    - No username password
    - Uses auth token
      * TTL 15 minutes
    - Auth managed by IAM
    - Benefits
      * To/from traffic encrypted with SSL
      * IAM central managed users
      * EC2 uses Profile cred specific to EC2 instance to access DB

### Amazon Redshift

* <https://tutorialsdojo.com/amazon-redshift/>
* PostgresSQL
* OLAP
  + Online Analytical Processing
  + Analytics and data warehousing
* Data warehouse
* Columnar not rows
* Uses zone maps and data compression
* Massively Parallel Query Execution
* Pay as you go
* SQL interface
* Data is loaded from
  + S3
  + DynamoDB
  + DMS
  + Other DBs
  + Kinesis Data Firehose
    - Through S3 Copy to Redshift
  + EC2 instance
    - JDBC driver
    - Large amounts at a time
* Cluster
  + 1 node – 128 nodes with 128 TB per node
  + Set of nodes
    - Leader Node
      * Query planning
      * Result aggregation
    - Compute Node
      * Performing the queried
      * Send results to leader
* Redshift spectrum
  + Run queries directly against S3 no need to load data into redshift
  + Leverages other redshift cluster generic
* Backup and restore
  + Automatically backed up to S3
* Snapshots and DR
  + No Multi AZ for redshift
  + Snapshots are point in time backups
    - Stored in S3
    - Incremental
    - Can be restored to new Cluster
    - Can do auto snapshots
    - Manual snapshots are retained until delete
  + Can auto copy snapshots to another cluster in a different region
* Enhanced VPC Routing
  + Allows using VPC features to manage flow of data between cluster and other resources
  + Can use VPC flow log to monitor COPY and UNLOAD

## **Management and Governance:**

### AWS Auto Scaling

* Scale Out
  + Add instances to match increased load
* Scale In
  + Remove instances to match decreased load
* Registers / deregisters instances with LB
* ASG Attributes
  + Launch Template
    - What is needed to launch a new instance
    - Used to be called Launch Configuration
    - AMI+instance type
    - EC2 User Data
    - EBS Volumes
    - Security Groups
    - SSH Key Pair
  + Min
    - Minimum instances in ASG
  + Max
    - Max limit instances in ASG
  + Initial Capacity
    - How many instances during normal
  + Network and Subnet info
  + Load Balancer Info
    - Can link ELB Target groups to ASG
  + Scaling Policy
    - Trigger for scale out / in
    - Can scale based on CloudWatch Alarms
      * Almost any metrics
        + Including Custom metrics
        + Good to scale on

CPUUtilization – Average CPU

RequestCountPerTarget – number of requests per Instance is stable

Average Network In/Out

* + - * Metrics are computed for the overall ASG Instances
    - Can be based on a schedule
    - New Scaling Rules
      * Can define rules directly managed by EC2
        + Target Average CPU Usage
        + Number of requests on ELB per instance
        + Avg network in
        + Avg network out
      * Easier to set up and make more sense (more intuitive)
    - Types of Scaling Policies
      * Dynamic Scaling
        + Target Tracking Scaling

Easiest

Example

Track average ASG CPU to stay at around 40%

* + - * + Simple / Step Scaling

Cloudwatch alarm is triggered then scale

* + - * + Scheduled actions

Scale at scheduled time due to known patterns

* + - * Predictive Scaling
        + Continuously forecast load and schedule scaling ahead
    - Scaling Cool Down
      * Default 300 sec (5min)
      * Happens when scale in or out
      * ASG will ot launch or terminate additional instances
        + Allow metrics to stabilize
      * Recommend using AMI that is ready to work as soon as launch. This way you can reduce the cooldown thus increasing efficiency
* Default Termination Policy (tries to balance the instances across AZ)
  + Find AZ with most instances
  + Delete instance with oldest launch template
* Lifecycle Hooks
  + Pending
    - Can be used to install addition software, run tests, etc
  + Terminating
    - Used to extract info Logs etc form instance before terminating
* Billing
  + Free, only pay for instances / volumes
* Security
  + IAM roles attached to ASG get assigned to Instances
* ASG will replace instance that get terminated
* Health Checks
  + [Health checks for Auto Scaling instances - Amazon EC2 Auto Scaling](https://docs.aws.amazon.com/autoscaling/ec2/userguide/healthcheck.html)
  + ASG will terminate (and replace) instances marked as unhealthy
  + EC2
    - Checks for software or hardware failures
    - Default must be enabled
    - Uses results of EC2 status Checks
      * System
        + Underlying hardware insfrastructure
      * Status
        + EC2 OS and Apps
  + ELB
    - Can enable if ASG has ELB
    - Healthy if instance is running and reported as healthy to ELB
* Troubleshooting
  + If instances always change to unhealthy when created
    - Misconfigured EC2 instance
      * Security Group
      * User Data

### AWS Backup

* Fully managed
* Centrally managed and automate backups
* Supports many services
* Supports cross-region backup
* Supports cross account backups
* Point in Time backups
* On demand and scheduled
* Tag based backup policies
* Backup plans
  + Backup policies
  + Backup frequency
  + Backup window
  + If / when to transition to cold storage
  + Retention Period

### AWS CloudFormation

* Infrastructure as Code
* Declarative
* Stack Sets
  + Cloudformation stack
  + Across accounts and regions

### AWS CloudTrail

* Provides
  + Governance
  + Compliance
  + Audit
* Enabled by default
* Gets history of events / api calls made from
  + Console
  + SDK
  + CLI
  + AWS Services
* Retention
  + Cloudtrail keeps logs for 90 days
  + If you want to keep them longer Can store logs in Cloudwatch logs or S3
    - Can then use athena to analyze
* Trail can apply to all regions(default) or a single region
* CloudTrail Events
  + Management Events
    - Operations that are performed on resources in your AWS Account
    - Enabled by default
    - Can separate into Read / Write events
  + Data Events
    - No logged by default because high volume
    - Amazon S3 object-level activiety
    - AWS Lambda function execution activity
  + CloudTrail insights Events
    - Must be enabled and will be billed for this service
    - Detect unusual activity
      * Inaccurate resource provisioning
      * Hitting service limits
      * Bursts of AWS IAM actions
      * Gaps in periodic maintenance activity
    - Analyzes normal management events to create baseline

### Amazon CloudWatch

* Provides metrics for every service in AWS
* Set alarms etc
* dashboard
* Metrics
  + Data from service
  + Namespace
    - Group of similar metrics
  + Dimension
    - Attribute of metric
    - 10 attributes per metic
  + Basic
    - 5 minutes
  + Detailed Monitoring
    - Every 1 minute
  + EC2 Memeory is not pushed by default
    - Must use custom metric
* Custom Metics
  + Define own metrics
    - RAM
    - Usage
    - Disk space
    - Logged in users
    - Etc
  + Use API
    - PutMetricData
  + Resulution (how often info sent
    - Standard = 1 minute
    - High resolution = 1/5/10/30 seconds
  + With custom metrics can push data points two weeks in the past or 2 hours in the future
* Dashboard
  + Global
  + Can include graphs from diff accounts and regions
  + Auto refresh
  + Can be shared with people who don’t have an AWS Account
* Logs
  + Log groups
    - Arbitraty
    - Usually represent an application
  + Log Stream
    - Instances within application / log files/ containers
  + Logs can expire
    - Will get deleted
  + Cloudwatch logs can send to
    - S3
    - Kinesis Data Streams
    - Kinesis Data Firehose
    - Lambda
    - ElasticSearch
  + Sources
    - SDK
    - Cloudwatch Logs Agent
    - Cloudwatch Unified Agent
    - Elastik Beanstalk application logs
    - ECS container logs
    - Lambda function logs
    - API gateway logs
    - Cloudtrail based on filter
    - Route 53 DNS logs
  + Cloudwatch Log Filter
    - Filter data by error or metic etc
    - Metric filters can trigger cloudwatch alarms
  + Cloudwatch logs insight
    - Query logs and add queries to dashboards
  + S3 export
    - Takes up to 12 hours to become available
    - AP CreateExportTask
    - Not real time
  + Cloudwatch Logs Subscriptions
    - Real time
  + Log Aggregation
    - Across Accounts
    - Across Regions
* Cloudwatch Logs Agent
  + Replaced by Unified Agent
* Cloudwatch Unified Agent
  + Allows internal log in EC2 to go to Cloudwatch
  + EC2 instance needs the correct IAM permissions to write to cloudwatch
  + Can also be used on prem
  + Can also send system level metrics
  + Centralized config with SSM parameter Store
* Cloudwatch Alarms
  + Trigger notifications for any metric
  + Targets
    - EC2 Instance
      * Stop
      * Start
      * Terminate
      * Reboot
      * Recover
      * EC2 Instance Recovery
        + Instance Status = VM check
        + System Status = underlying hardware
        + Can recover if alarm triggered

Same Private, public, elastic IP, metadata, placement group

* + - Auto scaling
      * Trigger action
    - SNS
      * Send notification

### AWS Config

* Audit and compliance recording
* Record configs and changes overtime
* Alerts for any change
* Per region service
* Can be aggregated across regions and accounts
* Possibility of storing config data in S3
* Rules
  + AWS Managed Rules
  + Custom Rules
  + Rules are evaluated / triggered
    - For each config change
    - At scheduled times
  + Do not prevent actions from happening
* Can view
  + Compliance of resource over time
  + Configuration of resource over time
  + Cloudtrail api calls of resource over time
* Rules Remediation
  + Automate remediation of non-compliant resources using SSM Automation documents
  + Can have retries
* Notification
  + Eventbridge to notify when out of compliance
  + SNS

### Amazon EventBridge (Amazon CloudWatch Events)

* Used to be called Cloudwatch Events
* Buses
  + Default event bus
    - Cloudwatch events
    - Intercept events from AWS services
    - Schedule or cron
  + Partner event bus
    - Receive events from SaaS service or application
    - Non AWS
  + Custom Event bus
    - Your own applications
* Can have cross account access to buses
* Rules
  + How to process the vents
* Scheme Registry
  + Used by your app to let eventbridge know how the data is configured on the bus

### AWS Organizations

* Global Service
* Allows you to manage multiple AWS Accounts
* Main acct is called master
  + Cannot be changed
* Member accounts can only be part of one organization
* Benefits
  + Consolidated Billing across all accounts
  + Pricing benefits – reduced pricing
  + Set tagging standards for billing purposes
  + Enable cloudtrail on all accounts to sent logs to central location
  + Cloudwatch logs sent to central location
* API to automate account creationg
* Uses
  + Separate per-account service limists
  + Isolate for logging
  + Security
  + Regulation
  + Etc
* Can set cross account roles for admin purposes
* Organizational Unti (OU)
  + Groups accounts
  + Can be nested
  + Inherit rules
* Service Control Policies
  + Whitelist or blacklist IAM Actions
  + Applied at the OU or Account Level
  + Does not apply to master account
  + Applies to all users and roles of account including root
  + Does not affect service linked roles
    - Enable other AWS services to integrate with AWS Organizations and can’t be restricted
  + SCP must have explicit Allow
  + Use cases
    - Restrict access to certain services
    - Enforce PCI compliance by explicitly disable services
  + Precedence
    - Deny takes precedence
* Moving Accounts
  + Remove account from old org
  + Invite account to new org
  + Accept invite

### AWS Resource Access Manager

* Allows sharing of resources to other accounts
* Share with any account or within org
* Avoid resource duplication
* Shared Services VPC
* Resources that you can share
  + VPC Subnets
    - Allow to have all resources in same subnets
    - Accounts must be from same Org
    - Cannot share security groups
    - Cannot share default VPC
    - Participants can manage their own resources
    - Participants can’t view, modify, delete resources that belong to other participants or owner
    - Network is shared so resources can talk across account to other resources owned by other accounts (over Private IP)
  + AWS Transit Gateway
  + Route 53 Resolver Rules
  + License Manager Configurations

### AWS Systems Manager

* Parameter Store
  + Secure storage for configuration and secrets
  + Optional encryption
  + Serverless
  + Scalable
  + Durable
  + SDK
  + Version tracking of config/secrets
  + Notification with Eventbridge
  + Integration with CloudFormation
  + Hierachial
  + Can retrieve parameters directly from AWS
    - Like the latest AMI version
  + Can retrieve from the Sectres Manager as well
  + API
    - GetParameter
    - GetParameterbyPath
  + Tiers
    - Standard
      * Free
      * 10000 parameters per account and region
      * 4KB max size
    - Advanced
      * .05 per advance parameter per month
      * 100000 parameters per account and region
      * 8KB max size
      * Path for throughput
      * Parameter Policies
        + Only for advanced parameters
        + Allow TTL to parameter to force updating or deleting sensitive data
        + Can assign multiple polices

### AWS Trusted Advisor

## **Migration and Transfer:**

### AWS Database Migration Service (AWS DMS)

* Migrate DB to AWS
* Resilient
* Self healing
* Source DB remains up during migration
* Homogeneous migration same to same
* Heterogenous ex MSSQL to Aurora
* Contiuous data transfer
* Sources
  + On Prem and EC2 instances
  + Azure
  + AWS RDS
  + S3
* Targets
  + On prem and EC2 instances
  + AWS RDS
  + Redshift
  + DynamoDB  
    S3
  + ElasticSearch
  + Kinesis Data Streams
  + DocumentDB
* AWS Schema Conversion Tool (SCT)
  + Convert DB schema from one engine to another

### AWS DataSync

* <https://tutorialsdojo.com/aws-datasync/>
* Move **large amount** of data from onprem to AWS
* Can sync to
  + S3
    - Can be
      * S3 Standard
      * S3 intelligent tiering
      * S3 Standard IA
      * S3 one zone IA
      * S3 Glacier
      * S3 glacier Deep Archive
  + EFS
  + FSx for Windows
* Move data from on prem via NFS or SMB
* Replication scheduled
  + Hourly
  + Daily
  + Weekly
* Requires Data Sync agent
* Can limit bandwidth used
* Can be used to sync AWS EFS to AWS EFS
* Use cases
  + Data migration
  + Data process for hybrid workloads
    - Can schedule and accelerate data moving into and out of AWS
  + Large amounts of cold data you can move to AWS Glacier or Glacier Deep Archive.
  + Replicate NAS to AWS
* Use Data Sync to migrate large amounts of data and then use File Gateway to retain access and for ongoing updates
* Billing
  + Pay for amount of data copied

### AWS Migration Hub

### AWS Server Migration Service (AWS SMS)

* Incremental replication of On-Premis live servers to AWS

### AWS Snowball

* https://tutorialsdojo.com/aws-snowball/
* Highly secure
* Portable devices
* Use for
  + Collect process data at the edge
    - Any location with limited or no network connectivity
    - Use case
      * Preprocess data
      * Machinelearning at edge
      * Transcoding media streams
    - Can run EC2 instances and AWS Lambda Functions ( using AWS IOT GreenGrass)
    - Long term deployment options
    - SnowCone
      * 2 CPU
      * 4GB RAM
      * Wired or wireless
      * USB-C power using cord or optional battery
    - Snowball Edge
      * Compute Optimized
        + 52VCPU
        + 208 GB RAM
        + 42 TB usable storage
      * Storage Optimized
        + Up to 40 vCPU
        + 80 GB RAM
  + Data Migration
    - Offline devices to perform data migrations
      * If it take for than a week over the network. Use snowball devices
    - migrate in and out of AWS
    - Three devices
      * Snowcone
        + Small portable computing anywhere rugged and secure
        + Light weight
        + Used for edge computing, storage, data transfer
        + 8TB
        + Can transfer data

by shipping to AMZ

AWS DataSYnc

* + - * Snowball edge
        + TB or PB of data
        + Pay per data transfer job
        + Provide block storage and S3 compatable object storage
        + Storage optimized

80TB Block volue S3 compatible

* + - * + Compute optimized

42 TB Block volue S3 compatible

* + - * Snowmobile
        + Truck Container
        + Exabytes
        + Each snowmobile has 100 PB of capacity. Use multiple in parallel
        + High security
        + Temp controlled
        + GPS monitored
        + Video surveillance
        + Better if you are transferring more than 10 PB
* OpsHub
  + Software on computer to manage snow devices
  + Unlock and configure single or clustered devise
  + Transfer files
  + Launching and manging instances on device
  + Monitor
  + Launch compatible AWS services on device
* SnowBall into Glacier
  + Cannot do it directly
  + Upload to S3
  + Use lifecycle policy to then move to glacier

### AWS Transfer Family

* [AWS Transfer Family features | Amazon Web Services](https://aws.amazon.com/aws-transfer-family/features/)
* Fully manages file transfers over
  + SFTP
  + FTPS
  + FTP
* In and out of
  + S3
  + EFS
* Store and manage user creds
* Integrate with existing auth systems

## **Networking and Content Delivery:**

### Amazon API Gateway

* Create rest API
* Websocket protocols
* API Versioning
* Security
  + IAM Permissions
    - Create Policy to attach to user/role so they can access API Gateway
    - API Gateway verifies IAM Permissions passed by the calling app
    - Goot to provide access within own infrastructure
    - Leverages SIG v4 headers
  + Lambda Authorizer
    - (formerly Custom Authorizers)
    - Uses AWS Lambda to validate the token in the header
    - Can cache results of auth
    - Used with OAuth/ SAML/ 3rd party
    - Lambda must return an IAM policy for the user
  + Cognito User Pools
    - Fully manages user lifecycle
    - API Gateway verifies identity auto from Cognito
    - No custom implementation required
    - Authentication not authorization
      * Backend must verify authorization
* API Keys
* Throttling
  + [Throttle API requests for better throughput - Amazon API Gateway](https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-request-throttling.html)
  + Protects API from being overwhelmed by requests
  + Best effort
  + Uses Token Bucket Algorithm
* Swagger / Open API
* Transform and validate request responses
* Generate SDK and API specification
* Cache API Responses
* Integrates with
  + Lambda Function
  + HTTP
    - Including ON prem
  + AWS Service
* Endpoint Types
  + Edge-Optimized
    - Default
    - Global clients
    - Requests are routed throught cloudfron edge locations
    - API is still only in one region
  + Regional
    - Clients within same region
    - Could manually combine with Cloudfront
      * More control over the caching strategies and distribution
  + Private
    - Only access via VPC using interface VPC endpoint (ENI)

### Amazon CloudFront

* CDN
* Improves read performance as content is cached at edge locations
* DDOS protection
* Integration with Shield
* AWS WAF integration
* Expose external and internal HTTP
* Great for static content
* Origins
  + S3 Bucket
    - Distributing files and caching at edge
    - Enhanced security with Origin Access Identity (OAI)
      * Uses OAI and S3 Bucket Policy to allow access
    - Ingress to upload files to S3
  + Custom Origin
    - HTTP
    - ALB,EC2 Instance
      * Security group must let CloudFront request through
    - S3 website
    - Any HTTP backend (including on prem)
* Geo Restriction
  + Restrict who can access distribution
  + Whitelist
    - Allow users to access content from approved countries
  + Blacklist
    - Countries not allowed
  + Uses 3rd party to determine GEO-IP of country
  + Uses
    - Copyright laws
* Signed Url/signed cookie
  + Access to paid premium users
  + Policy
    - Includes URL expiration
    - IP Ranges to access the data
    - Trusted signers ( which AWS accounts can ceate signed urls)
  + Signed Url Gives access to individual files
    - One URL per file
  + Signed Cookies gives access to multiple files
  + Types of signed URL
    - Cloudfront signed URL
      * Allow access to a path no matter the origin
      * Account wide, only root can manage
      * Filter by IP, path, date , expiration
      * Leverage caching
    - S3 Pre-signed URL
      * Issue a request as the person who pre signed url
      * Uses IAM key of the signing IAM principal
      * Limited lifetime
* Cloudfront advanced
  + Billing
    - Cost of data out depending on location
    - Price classes
      * All
        + All regions
        + Best performance
      * 200
        + Most regions
        + Excludes most expensive
      * 100
        + Only least expensive regions
  + Multiple Origin
    - Route to different kind of origins based on content type
    - Based on path pattern
  + Origin Group
    - Increase high availability
    - Failover
    - One primary and one secondary origin
  + Field Level Encryption
    - Protect info through stack at field level
    - Additional layer with HTTPS
    - Sensitive info encrypted at edge close to user
    - Asymmetric encryption

### AWS Direct Connect

* Dedicated private connection to VPC
  + Setup between your data center and AWS Direct Connect Location
* Needs VPN Gateway on VPC
* Use cases
  + Increased bandwitdth throughput
    - Large data sets
  + More consistent network experience
    - Real time data feeds
* Direct Connect Gateway
  + Allows you to connect to multiple VPC in multiple AZ / Regions with one Direct connect
* Types
  + Dedicated Connection
    - 1 GBps and 10 Gbps capacity
    - Physical port dedicated to customer
    - Request to AWS then they sub to AWS direct connect partner
  + Hosted Connection
    - 50MBps, 500 MBps, 10 GBps
    - Request made to AWS Direct connect partners
    - Capacity can be added or removed on demand
* Lead times are often over 1 month
* Direct Connect + VPN
  + Direct connect is private but not encrypted
  + Encrypt with VPN
* Resiliency
  + High Resiliency for critical workloads
    - 2 data centers each with its own Direct Connect
  + Maximum Resiliency for critical workloads
    - 2 data centers each with Two direct connect

### AWS Global Accelerator

* Leverage AWS internal network
* Client talks to edge and then request goes through AWS
* Uses anycast
  + All edge locations have the same IP for your app
  + Client connects to closest
* Works with
  + Elastic IP
  + EC2 instances
  + ALB
  + NLB
  + Public or private
* Consistend performance
  + Internal route to lowest latency with fast failover
  + No client cache
* Health Checks
  + Performs health checks on your app
* Packets are proxy at edge to interal services
* Use Case
  + Non-HTTP
    - Gaming
    - IOT
    - VOIP
  + HTTP require static IP
  + HTTP require deterministic fast regional failover
  + Assiste with blue/green deployments using Endpoint weights to slowly dial how much traffic to new deployment
    - Endpoint weights
      * Determines the proportion of traffic that is directed to endpoints in an endpoint group (aws region where endpoints are deployed

### Amazon Route 53

* <https://tutorialsdojo.com/amazon-route-53/>
* HA
* Scalable
* Fully Managed
* Authoratative DNS
  + Means you as the customer can update the DNS records
* Domain Registrar
  + 12 bucks per year
  + Registers domain name for your use
  + Can transfer from non AWS to route 53
  + Can configure DNSSEC
* Health check resources
  + HTTP Only for public resources
  + Automated DNS failover
    - Health Checks monitor Endpoint
      * Application, server, other AWS resource
      * 15 global health checks
      * Healthy/unhealthy threshold (default 3)
      * Interval = 30 SEC (10 sec smalles)
      * Supports
        + HTTP, HTTPS, TCP
      * If > 18% report healthy, then Route 53 considers it healthy otherwise unhealthy
      * Can choose which locations to use for healthchecks
      * Pass = 2xx and 3xx status codes
      * Can pass fail based on text in response
    - Health check that monitor other health checks
      * Calculated health check
      * Combine results of multiple health checks into one check
      * Can monitor up to 256
      * Specify how many need to pass to make parent pass
      * Usage
        + Perform maintenance on your website without causing all health checks to fail
    - Health check that monitor Cloudwatch Alarms
      * Helpful for private resources
  + Integrated with Cloudwatch metrics
* Record types
  + A
    - Maps hostname to IPv4
    - Can be both AWS IP or on-prem
  + AAAA
    - Maps hostname to IPv6
  + CNAME
    - Maps hostname to another hostname
    - Target must have A or AAAA record
    - Cant create a cname record for top node of DNS namespace
  + ALIAS
    - Point host name to AWS resource
    - Works for root domain and non root domain
    - Free
    - Health check built in
    - Always of type A/AAAA
    - No TTL
    - Cannot set Alias for EC2 DNS name
  + NS
    - Name services for hosted zone
    - Control how traffic is routed to domain
* Hosted Zone
  + Container for records that define how a route traffic to a domain and subdomains
  + Public Hosted Zone
    - Contains records that specify how to route traffic on the internet (public domain names)
  + Private Hosted Zone
    - Records for internal VPC ( private domain names)
    - Must set to true on each VPC
      * enableDNSHostNames
      * EnableDNSSupport
* Billing
  + .50 per month per hosted zone
* TTL
  + Time To Live
  + How long to cache DNS record locally
* Routing Policies
  + Define how Route 53 responds to DNS Queiries
  + Types
    - Simple
      * Route to single resource
      * Can specify multiple values
        + Returned value will be randomly chosen by client
      * When Alias enabled specify only one AWS resource
      * Can’t be associated with Health checks
    - Weighted
      * Control % of the requests that go to each specific resource
      * DNS Records must be of same name and type
      * Can use Health Checks
      * Weight of 0, will stop sending traffic to resource
      * If all have weight of 0, then all records will be returned equally
    - Failover
      * Mandatory health check
      * Fails over to secondary if primary unhealthy
    - Latency based
      * Route to resource that has the least latency close to us
      * Based on traffic between users and AWS regions
      * Can use health checks
    - Geolocation
      * Based on where user is located
      * If overlapping then most precise location wins
      * Should create default record
      * Use case
        + Website localization
        + Restring content distribution
        + Load balanching
      * Can use healthchecks
    - Multi-Value Answer
      * Route to multiple resources
      * Returns multiple values (up to 8)
      * Can use health checks and Route 53 will return only healthy resources
      * Client will randomly choose one
    - Geoproximity
      * Route traffic based on location of user and resources
      * Bias
        + Shift more traffic to resources
        + More traffic bias positive
        + Les traffic bias negative
      * Requires Route 53 Traffic Flow
      * Resources can be
        + AWS Resource ( specify AWS Region)
        + Non-AWS resource ( Specify latitude and longitude)
* 3rd party domains
  + can register domain with other registrar and use Route 53 as DNS
    - Create Public Hosted Zone in Route 53
    - Then on restrar site enter the Route 53 name servers as custom servers
* Routing traffic to website hosted in S3 Bucket
  + S3 bucket configured to host static website
    - Bucket must have the same name as your domain or subdomain
  + Registered domain name

### AWS Transit Gateway

* <https://tutorialsdojo.com/aws-transit-gateway/>
* Transitive peering between thousans of VPC and on prem
* Hub and spoke
* Cross region and cross account
* Route tables limit which VPC can talk with another
* Works with Direct Connect and VPN
* Supports multicast (unlike any other AWS service)
* ECMP
  + Equal cost multi path routing
  + Create multiple site to site VPN connections to increase bandwidth of connections
  + Routing strategy allow multiple best path
  + Creates more VPN tunnels which allow more throughput
* Can share Transit Gateway between multiple accounts

### Amazon VPC (and associated features)

* OverView
  + Virtual Private Cloud
  + Max 5 per regions (soft limit)
  + Max CIDR per VPC is 5
    - Min Size /28 (16 Ips)
    - Max Size /16 (65536 Ips)
  + Subnets
    - Each has 5 reserved IP
      * .0 – Network Address
      * .1 – VPC Router (default gateway)
      * .2 – mapping Amazon-provied DNS
      * .3 – Not used yet
      * .255 – broadcast ( not supported )
  + Only Private IPv4 ranges allowed
  + VPC CIDR should not overlap with other networks – on prem or in AWS
* Internet Gateway
  + Allows VPC resources to connect to internet
  + Scales horizontally
  + HA
  + Only one per VPC and only one VPC per Internet gateway
  + Require route table to direct traffic to internet
* Bastion Host
  + EC2 Instance in Public Subnet of VPC allowing admin to remote in and then connect to Instances in Private subnet
* Nat Instance
  + Outdated
  + Allows private instances to access internet
  + Requires Elastic IP
  + Route table must be configured to route traffic from private subnets to Nat instance
  + Not HA or resilience
  + You manage everything
* NAT Gateway
  + Fully managed
  + Pay per hour
  + Uses Elastic IP
  + Created in specific AZ
    - Resilient within AZ
    - Must create multiple NAT Gateways in Multiple AZ for HA
  + Can’t be used by instances in same subnet
  + Requires Internet gateway
* DNS Resolution
  + Setting that affect how VPC instances handles DNS
    - enableDNSSupport (DNS Resolution)
      * Enabled by default
        + Queries are sent to provider DNS Server at 169.254.169.253 or the reserved DNS IP (.2) +2
      * If disabled
        + You must build your own DNS Server
      * Decides if DNS resolution from Route53 resolver is supported
    - EnableDNSHostnames (DNS hostnames)
      * Default
        + True for default VPC
        + False for new VPC
      * Needs to have enableDNSSupport enabled for this to work
      * True
        + Assigns public hostname to EC2 Instance if it has public IP
  + Good for if you use Custom DNS domain names in Private Hosted Zone in Route 53
* NACL
* Security Group
* Reachability Analyzer
  + Network diagnostic
  + Troubleshoot network connectivity between two endpoint in VPC
  + Checks connectivity without sending packets
* VPC Peering
  + Privately connect two VPC
  + Make behave as if on same network
  + CIDR must not overlap
  + Not Transitive
  + Route tables must be updated to point to the peering connection name
  + Can be cross account
  + Can be cross region
  + Security groups can reference security group in different VPC in same region
* VPC Endpoint
  + Allows access to AWS services outside of the VPC without going through the internet. All traffic stays private inside of AWS
  + Can connect to service/vpc in another accont as welll
  + Also called privatelink
  + Connects to the private URL instead of public
  + Redundant and scale horizontally
  + Removes the need for IGW, NATGW
  + To connect to service behind ALB, you need to add an NLB in front.
  + Troubleshooting
    - Check DNS Setting Resolution on your VPC
    - Check Route Tables
  + Types of Endpoints
    - Interface
      * Provisions ENI
      * Must attach security group to allow access to other resources
    - Gateway
      * Support only
        + S3
        + DynamoDB
      * Provisions Gateway must be used as a target in route table
* VPC Flowlog
  + Captures traffic going throught your VPC interfaces
    - VPC Flow logs
    - Subnet Flow logs
    - Elastic Network Interface (ENI) Flow Logs
  + Helps monitor and troubleshoot connectivity issues
  + Captures network information from
    - AWS managed insterfaces
    - ELB
    - RDS
    - ElastiCache
    - Redshift
    - Workspaces
    - NATGW
    - Transit Gateway
  + Can be sent to S3 / Cloudwatch logs
  + Can be used for analytics on usage pattererns or malicious behaviour
  + Can be queried with Athena on S3 or Cloudwatch logs insights
* Site to Site VPN
  + Customer gateway
    - VPN software or device on customer site
    - Use your public routable IP for Customer Gateway or if behind a NAT then use the public NAT IP
  + VPN Gateway
    - Service in VPC to pair with Customer Gateway
  + Must set up Route Propagation for the VPN Gateway in route tabled
  + VPN CloudHub
    - Hub and spoke model using VPN
    - Between VPN Gateway and multiple on-prem sites
    - Setup VPN Gateway with dynamic routing and configure route tables
* Traffic Mirroring
  + Allows to capture and inspect traffic
  + Can filter traffic that you think is interesting
  + Mirror sends traffic to security appliance you manage
  + Does not interrupt traffic
  + Source and target of mirror must be in same VPC (or VPC peered)
* IPv6
  + All public and internet routable
  + Can be enabled in VPC
  + Troubleshooting
    - EC2 instance cannot launch in subnet
      * IPv4 (which cannot be disabled) has run out of IP
* Egress Only Internet Gateway
  + Used only for IPv6
  + Like NAT Gateway
  + Allow instance with IPv6 to access internet but nothing can originate from internet to access instance
  + Must update route table

## **Security, Identity, and Compliance:**

### AWS Certificate Manager (ACM)

### AWS Cognito

* Used to give users an identity so they can interact with our app
* Cognito User Pools
  + Sign in functionality for app users
  + Directory service
    - Can use built-in user management
  + Or Integrate with external identity Providers
    - Can enable Federated Identities ( Facebook, Google, SAML…)
  + Integrated with API Gateway
  + Creates serverless DB for mobile apps
  + Simple login
    - Username / password
  + Possibility to verify emails / phone numbers
  + Can use MFA
  + Sends back JSON web Token
  + Customized workflows with AWS Lambda Triggers
* Cognito Identity Pools
  + Federated Identity
  + Provide AWS Credentials to users so they can access AWS resources directly
  + Not a authentication mechanism
* Cognito Sync
  + Synchronize data from devices to Cognito
  + May be deprecated and replaced by App Sync
  + Requires CUP

### AWS Directory Service

* Non-SAML with AWS Microsoft AD Federation
* types
  + AWS Directory Services
    - AWS Managed Microsoft AD
      * Create own AD in AWS
      * Manage users locally either on prem or in AWS
      * Supports MFA
      * Trust connection with your own AD on prem
      * User shared between on prem and AWS
    - AD Connector
      * Directory Gateway (proxy) to redirect to on prem AD
      * Users managed on prem only
    - Simple AD
      * AD Compatible managed directory on AWS
      * Cannot be joined with on-prem AD

### Amazon GuardDuty

* Intelligent Threat discovery
* Machine learning algorithms to detect anomaly using 3rd party data
* Input data
  + CloudTrail logs
    - Unusual API calls
    - Unauth deployments
  + VPC Flow Logs
    - Unusual internal traffic
    - Unusual IP Address
  + DNS Logs
    - Compromised EC2 instances sending encoded data within DNS Queries
* Eventbridge rules can be set up to take action if anomalies found
  + Lambda
  + SNS
* Protect against CryptoCurrency attacks

### AWS Identity and Access Management (IAM)

* <https://tutorialsdojo.com/aws-identity-and-access-management-iam/>
* Global service
* Root account has full access. Do not use except to create users
* Users are people with access rights
* Groups
  + Contain users not other groups
  + Users can belong to more than one group
* IAM policy
  + JSON
  + Describes what user/group can do
  + Permissions
  + Inheritance
    - Users inherit from group
  + Inline policy
    - Policy only assigned to user
  + Syntax
    - Version
      * Policy language version
      * 2012-10-17 (usually)
    - ID
      * Identifier or the policy
      * Optional
    - Statement
      * One or more statements
      * SID
        + ID of the statement
        + Optional
      * Effect
        + Allow or Deny
      * Principal
        + Account/user/role to which policy is applied
      * Action
        + List of API calls that are allowed or denied
      * Resource
        + List of resources to apply policy to
      * Conditional
        + Conditions
        + Optional
* Password Policy
  + Set minimum password length
  + Require specific character types
  + Allow IAM users to change own passwords
  + Require users to change password after some time
  + Prevent password reuse
* MFA
  + Multi Factor Authentication
  + Best for privledge accounts
  + If password is stolen user cannot be used because they need the MFA token
  + MFA Device
    - Google Authenticator
      * Phone only
    - Authy
      * Multi-device
    - Universal 2nd factor (u2f) key
      * Yubikey
    - Hardwarye keyfob
  + Support for multiple tokens on one device
* Assume Role
  + User gives up account permissions to assume another set of permissions
* IAM Policies
  + Conditions
    - Make policies more granular
    - Aws:sourceIP
      * Apply policy to this IP
    - AWS:requesteRegion
      * Applies policy to this region where call is made to
    - Resourcetags
      * Applies policy only if resource tag matches
    - Force MFA
      * Apply if MFA
* IAM Permission Boundaries
  + Supported for user and roles
  + Sets the maximum permissions an IAM entity can get
  + Need IAM Policies also
  + Used in combination with to establish effective permissions
    - SCP
    - IAM Polices
* Policy Evaluation Logic
  + Explicit deny most powerful
* Access Keys
  + Generated through AWS Console
  + Users manage own access keys
  + Just like User/pw
    - Access key iD = username
    - Secret Access Key = Password
  + \
* CLI
  + Access API in AWS
  + Begin with AWS
  + Alternative to console
* SDK
  + Language specific APIs
* Cloudshell
  + Terminal on AWS
  + Alternative to CLI
  + Only available in some regions
* IAM Roles for Services
  + Allow services to do work on your behalf
  + Permissions for service
* IAM Credentials Report
  + Account level
  + List of all users and the status of various credentials
* IAM Access Advisor
  + User level
  + Shows service permission granted to a user and when those services were last accessed
  + Can use to revise (least privilege)
* Federated Users
  + Use on prem or other authentication to get IAM token
* STS
  + Provide trusted user with temp security token
  + Short term
  + Getting token
    - GetFederationToken
      * Returs temp creds for federated user
      * Typical proxy app
        + Gets temp creds on behalf of distributed app inside corp network
    - GetSessionToken
      * Returns temp creds for aws account or IAM User
      * Typical use
        + MFA to protec programmatic calls to AWS API

### Amazon Inspector

* Auto security assessments for EC2 instances
* Analyze OS against
  + known vulnerabilities
* analyze unintended network accessibility
* requires agent on instance

### AWS Key Management Service (AWS KMS)

* Easy to control access to your data
* Manages keys
* Keys can be shared with other AWS accounts
* Fully integrated with IAM for auth
* Integrated with lots of AWS Services
* Customer Master Key Types
  + Symetric
    - AES-256
    - Single key for encrypt / decrypt
    - Required for envelope encryption
    - Never get access to the key unencrypted (must use KMS API)
  + Asymetric Key
    - RSA and ECC
    - Public ( Encrypt) Private (Decrypt)
    - Use
      * Encryption outside of AWS by users who can’t call KMS API
* Able to audit key usage with CloudTrail
* Types of CMK
  + AWS Managed Service Default CMK : Free
  + Create own key : 1$ per month
  + User keys imported: 1$ per month
* Pay for API call to KMS: .03 cents per 10000 calls
* KMS can only encrypt up to 4KB of data per call
  + > 4KB requires envelope encryption
* Give access to KMS
  + Key policy allows user
  + IAM Policy allows API calls
* KMS Keys are bound to region
  + To copy resource across region
    - Create snapshot
    - Copy snapshot and create new key to reencrypt in that region
* Key Polices
  + Cannot control access without Key policies
  + Default KMS Key Policy
    - Created if you don’t provide a specific KMS Policy
    - Complete access to key to root user = entire AWS Account
    - Give access to IAM policies to KMS key
  + Custom KMS Key Policy
    - Define users, roles, that can access key
    - Define who can admin key
    - Useful for cross-account access to the key
* KMS Key Rotation
  + Automatic Key Rotation
    - Only for Customer-Managed CMK
    - Happens every 1 year
    - After rotation previous key is active so you can decrypt old data
    - New key has the same CMK ID ( only backing key is changes )
  + Manual Key Rotation
    - Use if you want to rotate different than 1 year
    - New key will have different CMK ID
    - Keep the previous key active to decrypt existing
    - Alias
      * Helps hid change of key ID for the application
    - Solution to rotate keys not eligible for auto key rotation
* Cloud HSM
  + Hardware Security Module
  + AWS provisiond Hardware
  + Dedicated hardware
  + You manage own encryption keys
  + Tamper resistant, FIPS 140-2 Level 3 compliant
  + Sysmtric and Asymentric keys
  + No free tier
  + Redshift supports
  + Good option for SSE-C
  + Must use Cloud HSM Software
  + Highly Available

### Amazon Macie

* Manages service
* Machine learning to discover and protect sensitive data
* Alert to sensitive data such as PII

### AWS Secrets Manager

* Store secrets
* Can force rotation of secrets x days
* Automate generation of secrets on rotation
  + Uses Lambda
* Integration with RDS
* Secrets are encrypted using KMS

### AWS Shield

* DDOS Protection
* Standard
  + Free service activated for all customers
  + Protection from
    - SYN/UDP Floods
    - Reflection attachs
    - Layer 3 and 4 attacks
* Advanced
  + 3000 per month per org
  + Protect against more sophisticated attacks
  + DDOS mitigation service
  + 24/7 access to DDOS response team
  + Protects against higher fees during usage spikes due to DDOS

### AWS Single Sign-On

* Security Token Service
  + Grant limited and temporary access to AWS Resources
  + Token is valid for up to one hour (must be refreshed)
  + API
    - AssumeRole
      * Within your own account for enhanced security
      * Cross account access to perform actions there
    - AssumeRolewithSAML
      * Return credentials for users logged with SAML
    - AssumeRoleWithWebIdenity
      * Return creds for users logged with an Id Provider (Facebook, google, ODIC compatible, etc
      * AWS recommends against using this and to use Cognito instead
    - GetSessionToken
      * For MFA, From a user or AWS account root user
  + Using STS to assume role
    - Define IAM role within your account or cross account
    - Define which principals can access this IAM role
    - Use AWS STS to retrieve credentials and impersonate the IAM role you have access to (AssumeRole API)
* Identity Federation
  + Allows users outside of AWS to assume temp role for accessing AWS resources
  + Do not need IAM users
  + Types
    - SAML
      * Integrates with AD / ADFS
      * Provides access to AWS Console or CLI
      * Need to setup a trust between AWS IAM and SAML (both ways)
      * Enables web-based, cross domain SSO
      * Uses AssumeRolewithSAML
      * Older way
    - Custom Identity Broker
      * Custom code
      * ID Broker must determine the appropriate IAM Policy
      * Uses STS API
        + AssumeRole or GetFederationToken
    - Web Identity Federation with Amazon Cognito
      * Provide direct access to AWS resources from client side
    - Web Identity Federation without Cognito
      * AssumeRolewithWebIdentity
    - Single Sign on
    - Non-SAML with AWS Microsoft AD
      * See AWS Directory Services
* Centrally manage multiple accounts and 3rd party business apps
* Integrated with AWS Org
* Supports SAML 2
* Integrates with On-prem AD
* Centralized permission management
* Centralized auditing with cloudtrail

### AWS WAF

* Web Application Firewall
* Layer 7 exploits
* Deployed on
  + Application Load Balancer
  + API Gateway
  + CloudFront
* Define Web ACL
  + Rules can include
    - IP Addresses
    - HTTP Headers
    - HTTP Body
    - URI Strings
  + Size Constraints
  + Geo-match ( Block countries)
  + Protects common attack
    - SQL injection
    - Cross-site Sripting (XSS)
  + Rate based rules (counts per second)
    - Protect DDOS
* AWS Firewall Manager
  + Manage rules in all accounts of an AWS Org
  + Define Common set of rules
    - WAF rules
    - AWS Shield Advanced
    - Security Groups for EC2 and ENI

## **Storage:**

### Amazon Elastic Block Store (Amazon EBS)

* <https://tutorialsdojo.com/amazon-ebs/>
* Data persists even after EC2 termination
* Can only be mounted to one instance at a time unless Multi-Attach
* Bound to specific AZ
  + Cannot have EBS created in AZ 1 attached to EC2 in zone AZ2
  + To move to another AZ, you need to snapshot Volume first
* Network USB stick
* Has provisioned Capacity ( Size and IOPS)
  + Can increase over time
* Billing
  + Billed for provisioned capacity
* Delete on Termination attribute
  + By default
    - Root volume – deleted when instance deleted
    - Attached Volume – not deleted
* Snapshot
  + Point in time backup
  + Not necessary to detach
  + Can copy across AZ or region
    - Snapshot and restore
  + Incremental
* Volume Types
  + Gp2/gp3
    - SSD
    - General purpose
    - Balance price and performance
    - Can be used as boot volume
    - 1GB-16TB
    - GP3:
      * 3000 IOPs – 16000 IOPS
      * Throughput of 125 MBps – 1000 MBps
    - GP2:
      * Burst to 3000 IOPS
      * IOPS linked to volume size mad IOPS 16000
  + Io1/io2
    - Provisioned IOPs
    - High performance SSD
    - Mission critical low latency
    - High throughput
    - Can be used as boot volume
    - 4 GB – 16 TB
    - Max IOPS 64000 for Nitro and 32000 for others
    - IOPS provisioned separately from size
    - IO2 have more durability and IOPs per GB
    - IO2 Block Express
      * 4 GB – 16 TB
      * Sub millisecond latency
      * Max IOPS 256000 with IOPS:GB ration 1000:1
    - Supports Multi-Attach
  + St1
    - HDD
    - Throughput Optimized
    - Requently accessed, throughput intensive workloads
  + Sc1
    - Lowest cost HDD
    - Cold HDD
    - Less frequently accessed workloads
* EBS Volumes are characterized
  + Size
  + Throughput
  + IOPS
* Multi-Attach
  + IO1/IO2 only
  + Attach same EBS to multiple instances in same AZ
  + Use
    - Higher application availability
    - Applications must manage concurrent write operations
  + Cluster aware File system
* Encryption
  + Leverages keys from AWS KMS (AES-256)
  + When encrypted EBS volume
    - Data at rest encrypted
    - Data in fline between instance and volume encrypted
    - Snapshots encrypted
    - All volumes created from snapshot are encrypted
  + How encrypt unencrypted volume
    - Create BS Snapshot
    - Encrypte snapshot using copy
    - Create new volume from snapshot
    - Attach encrypted volume to the original instance
* Backup
  + AWS Backup can automate
* Amazon Data Lifecycle Manager
  + <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/snapshot-lifecycle.html>
  + Automate EBS Snapshots
  + Does the following
    - Enforce backup schedule
    - Create standardized AMIs that can be refreshed at regular intervals
    - Retain backups as required by auditors or internal compliance
    - Reduce storage costs by deleting outdated backups
    - Create disaster recovery backup policies that back up data to isolated accounts

### Amazon Elastic File System (Amazon EFS)

* Manage NFS v4.1
* Can be mounted to more than one EC2 instance
* Can be mounted across AZ
* HA
* Scalable automatically
* Billing
  + Pay per Use
* Security groups can be applied to control connections
* Only works with Linux
* Encryption at rest using KMS
* Modes
  + Performance Mode
    - GP
      * Default
      * Latency sensitive
    - Max I/O
      * Higher latency
      * Throughput
      * Highly parallel
  + Throughput Mode
    - Bursting
    - Provisioned
      * Set your throughput regardless of storage size
* Storage Tiers
  + Lifecycle management
    - Can move data after N days\
  + Standard
    - Frequently access files
  + Infreaquent Access (EFS-IA)
    - Cost to retrieve files, lower price to store

### Amazon FSx

* <https://tutorialsdojo.com/amazon-fsx/>
* Launch 3rd party hig performance file systems on AWS
* Fully managed
* FSx for Windows
  + Windows file system share drive
  + SMB and windows NTFS
  + MS AD integrated, ACL, Quotas
  + SSD scales big
  + Can be accessed from your on-prem infrastructure
  + Can be configured to be Multi-AZ
  + Daily backup to S3
* FSx for Lustre
  + Lustre = Linux and Cluster
  + Parallel distributed file system for large scale computing
  + Used for
    - Machine learning
    - High Performance Computing
    - Video Processing
    - Financial Modeling
    - Electronci Design automation
  + Seamless integration with S3
    - Can read/write S3 as a file system through FSx Lustre
  + Can be used on prem
* FSx Deployment Options
  + Scratch file system
    - Temp storage
    - Data is not replicated (not persist if file server fails)
    - High burst
    - Usage
      * Short term processing optimizing cost (no replication)
  + Persistent File System
    - Long term storage
    - Replicated in same AZ
    - Replace failed files within minutes

### Amazon S3

* <https://tutorialsdojo.com/amazon-s3/>
* Infinitely scaling
* Bucket
  + Each must have globally unique name
  + Defined at regional level
* Objects
  + Files in bucket
  + Key / value pair
    - Key
      * full path to bucket
    - Value
      * Content of file
  + Prefix
    - Subfolders
  + Max size 5 TB
  + Cannot upload more than 5GB
    - If larger use multi part upload
  + Metadata
    - Key/value pairs describing object
  + Tags
* Performance Obtimize
  + Multipart upload
    - Recommended for files > 100MB
    - Must use for files > 5GB
    - Helps parallelize uploads to speed up transfer
  + S3 Transfer Acceleration
    - Increase transfer spped by transferring file to AWS edge location which then forwards to bucket via high speed internal network
  + S3 Byte-Range Fetches
    - Parallelize GETS by requesting specific byte ranges
    - Better resilience in case of failures
* S3 Event Notification
  + Rules can be filtered
  + Events can be sent to sns,sqs, lambda
  + Delivered in seconds but sometimes longer
  + Enable versioning to make sure that multip writes at same time trigger event
* Versioning
  + Must enable at bucket level
  + Keeps each version
  + Prevents unintended deletes
  + Can rollbak
  + Any file not versioned prior to versioning will have a version of NULL
  + Suspending versioning
    - does not delete the previous versions
    - any new object with have a version of NULL
  + Deleting
    - Creates a delete marker
      * Pretends the version is deleted but older versions still exist
  + Restoring deleted object
    - Delete the delete marker
  + Deleting version or delete marker is a permanent delete.
* Encryption
  + Methods to encrypt
    - SSE-S3
      * Encrypts S3 objects using keys handled and managed by S3
      * Aes-256
      * Must set Header
        + X-amz-server-side-encryption:AES256
    - SSE-KMS
      * Leverages AWS KMS to manage encryption keys
      * User control of keys
      * Audit trail
      * Set header
        + X-amz-server-side-encryption:aws:kms
    - SSE-C
      * Customer managed keys
      * Must use HTTPS
      * Key sent with object, S3 encrypts and then discards key
      * Data key in header
      * Must be done via CLI
    - Client Side Encryption
      * Encrypted / Decrypted on the client device
      * Client libraries handle such as Amazon S3 Encryption Client
  + Encryption in transit ( TLS/SSL )
    - S3 exposes HTTP Endpoint and HTTPS endpoint
    - HTTPS is encrypted in flight
  + S3 Default Encryption
    - Bucket policy can be used to enforce encryption. Or set Default encryption
    - Bucket policy if evaluated before default encryption
    - When set will not encrypt existing object. Only newly uploaded objects
    - Does not reencrypt uploaded objects that are already encrypted
* Security
  + User Based
    - IAM Policies
      * Which API calls should be allowed for a specific user from IAM console
  + Resource Based
    - Bucket Polices
      * [Using bucket policies - Amazon Simple Storage Service](https://docs.aws.amazon.com/AmazonS3/latest/userguide/bucket-policies.html)
      * Bucket wide rules that apply to bucket or objects in the bucket
        + Can apply conditions to apply to only selected objects
      * Allows cross account
      * JSON Based
      * Can be applied to bucket and objects
      * Actions
        + Set of API to allow or Deny
      * Effect
        + Allow or Deny
      * Principal
        + Accont or user to apply policy
      * Resource
        + ARN of bucket/ prefix/object
      * Uses
        + Grant public access
        + Force encryption at upload
        + Grand access to another account (cross account)
      * Block settings for block public access
        + Block public access to buckets and object granted through

New ACL

Any ACL

New public bucket or access point policies

* + - * + Block public and cross-account access to buckets and objects throu any public bucket or access point policies
        + Can be set at bucket and account level
      * Only bucket owner can apply bucket policies
      * Do not apply to objects owned by other accounts
    - Object Access Control List (ACL) ( legacy. Aws recommends bucket policy)
      * Finer grain
      * At object level
    - Bucket Access Control List ( ACL )
      * At bucket level
      * Less common
  + IAM Principal can access S3 object if
    - User IAM permissions allow OR the resource policy ALLOWS
    - AND there is not explicit DENY
    - By default object is owned by S3 account that created it. If the account is different than S3 Bucket owner, then the bucket account will not have access to object
  + Networking
    - Can access bucket via VPC endpoint
  + Logging and auditing
    - Access logs
      * can be stored in an S3 bucket (use different bucket)
      * can be analyzed with Athena
    - API calls can be logged in Cloudtrail
  + User Security
    - MFA delete
      * MFA required to delete in versioned buckets
      * Forces user to generate code to before doing important operations
        + Permanently delete object version
        + Suspend versioning on the bucket
      * Requires versioning be enabled
      * Only bucket owner (root account) can enable/disable MFA-Delete
      * MFA delete only works from CLI
    - Pre signed URL
      * URL that are valid only for limited time
      * Generate using SDK or CLI
      * Valid for a default of 3600 seconds
        + Can change with –expires-in
      * Presigned url inherit permsissions of person who generated URL for Get/put
  + S3 Object Lock
    - Requires Versioning
    - WORM
    - Block object version deletion for a specified amount of time
    - Object Retention
      * Retention Period
        + Object will be blocked from deletion for specific time
      * Legal Hold
        + Blocked from deletion with no expire date
    - Modes
      * Governance Mode
        + Users can’t overwrite or delete version or alter its lock uless they have special permissions
      * Compliance mode
        + Can’t be over written or deleted by any user
        + Includes root user
        + Retention mode can’t be changed
        + Retention period can’t be shortend
* Websites
  + S3 can host websites accessible www
  + Website URL
    - <bucketname>.s3-website.<AWS-region>.amazonaws.com
    - <bucketname>.s3-website-<AWS-region>.amazonaws.com
  + 403 error indicates no public access
* CORS
  + Cross Origin Resource Sharing
  + Origin
    - Scheme (protocol), host(domain),port
  + Get data from a different origin
  + Web browser based allow requests from other origins while visiting main origin
  + CORS Headers
    - Access-Control-Allow-Origin
    - Allows cross origin request
  + You can allow for a specific origin or for all (\*)
* Replication
  + Cross Region Replication (CRR) and Same Region Replication (SRR)
  + Asynch replication
  + Requires versioning on source and destination
  + Can be in different accounts
  + Requires correct IAM role
  + Only new objects are replicated
    - Existing objects will need to be manually copied
  + Delete
    - Optional to replicate delete markers
    - Deletions with version ID are not replicated to prevent malicious deletes
  + Cannot chain replication from bucket a to b to c
* S3 Analytics - Storage Class Analysis
  + Can set to help determine when to transition objects from standard to IA
  + Does not work for onezone-IA or glacier
  + Updated daily
  + Taje 24-48 hours to gain data
* S3 Requester Pays
  + Normally bucket owner pays for storage and transfer cost
  + But with requester pays, then who downloads pays for transfer costs
  + Requires downloader to have AWS Account
* S3 Select and S3 Glacier Select
  + Retrieve less data using SQL by performing server side filtering
  + Can filter by rows and columns only
  + Less network transfer
  + Less CPU
  + Less cost client side
* Athena
  + <https://tutorialsdojo.com/amazon-athena/>
  + SQL query service directly from S3
  + Supports formats
    - CSV
    - JSON
    - ORC
    - Avro
    - Parquet

### Amazon S3 Glacier

* Storage classes
  + S3 Standard
    - General Purpose
    - 99.99% Availability
    - Frequently accessd data
    - Low latency
    - High throughput
    - Sustain 2 concurrent facility failures
    - Use case
      * Big Data analytics
      * Mobile and gaming application
      * Content distribution
  + S3 Standard-IA
    - Les frequently accessed data
    - Lower cost than standard
    - 99.9% availability
    - Use case
      * DR
      * Backups
    - Minimum 30 days
  + S3 One Zone-IA
    - 99.5% availability
    - Single AZ so if AZ dies then data is lost
    - Use cases
      * Storing secondary backup copies of on-prem data you can recreate
    - Minimum 30 days
  + S3 Glacier instant Retreval
    - Millisecond retrieval
    - Great for data accessed once per quarter
    - Minimum storage duration 90 days
  + S3 Glacier Flexible Retreval
    - Formerly S3 Glacier
    - Retrieval time
      * Expedited
        + 1 – 5 minutes
      * Standard
        + 3 – 5 days
      * Bulk
        + 5 – 12 hours
        + Free
    - Minimum storage duration 90 days
  + S3 Glacier Deep Archive
    - Retrieval
      * Standard
        + 12 hour
      * Bulk
        + 48 hours
    - Minimum storage duration 180 days
  + S3 Intelligent Tiering
    - Small monthly monitoring and auto tiering fee
    - Move objects automatically between access tiers based on usage
    - There are no retrieval charges in S3 intelligent-tiering
    - Tiers
      * Frequent Access Tier (automatic)
        + Default
      * Infrequent access tier ( automatic
        + Objects not accessed for 30 days
      * Archive Instant Access Tier ( automatic )
        + Objects not accessed for 90 days
      * Arhive Access Tier (optional)
        + Configurable from 90 days to 700+ days
      * Deep Archive Access tier ( Optional)
        + Config from 180 to 700+ days
* S3 LifeCycle Configurations
  + Rules to move objects between classes
  + Transition actions
    - Defines when objects are transitioned to another storage class
  + Expiration Actions
    - Configure objects to expire (delete) after some time
    - Can be used to delete old versions
    - Can be used to delete incomplete multipart uploads
  + Rules can be created for certain prefix
  + Rules can be created for object tags
* Vault Lock
  + Adopt WORM ( write once read many) model
  + Lock the policy so no one can make changes
  + Useful for compliance and data retention

### AWS Storage Gateway

* <https://docs.aws.amazon.com/storagegateway/latest/userguide/StorageGatewayConcepts.html>
* <https://tutorialsdojo.com/aws-storage-gateway/>
* Bridge between on prem and cloud storage
* VM on prem usually
* Or Hard ware appliance from AWS
* Use cases
  + DR
  + Backup / restore
  + Tiered storage
* Types
  + File Gateway
    - S3 buckets accessible using NFS and SMB
      * S3 standard,
      * S3 IA
      * S3 one Zone IA
    - Access using IAM Roles for each File gateway
    - Recently used data cached in file gateway
    - Can be mounted on many servers
    - Integrated with AD for user auth
    - Support S3 Oject lock
      * WORM
  + Volume Gateway
    - Block storage ISCSI
    - Backed by EBS snapshots
    - Types
      * Cached
        + Most recent files in cache on gateway
      * Stored Volume
        + All data local but backed up to S3
  + Tape Gateway
    - Virtual tape library on S3
    - Existing tape base process using iSCSI
* FSx file Gateway
  + Native access to FSx for windows
  + Local cache
  + Useful for home directories and file shares