Professor Monfort’s AWS Study Guide

Security Edition

This document is a review study guide for the AWS Security exam. This is NOT a substitute for taking the prep courses, practice exams and independent studying, but instead a complement to all of that.

What is Cloud Computing?

On-demand self-service – You decide what you want and allocate it yourself
Resource pooling – Sharing resources with other companies
Rapid elasticity – Your resources can grow to the size you need when you need it
Metered services – You only pay for what you use
Network access – You can access it from anywhere

Why is the Cloud the Way to go?

Secure & reliable infrastructure (data centers) – Specialized & expensive skills
Low capital expenditures (capex) – No need to buy a bunch of HW upfront
No long-term commitments – You can experiment as needed
Fast time to market – No need to wait for HW to come in and get configured

General Design Principles

- **Stop guessing your capacity needs**: Make sure you can scale up & down as needed.
- **Test systems at production scale**: Temporarily spin up the test environment capacity you need to fully test at prod scale.
- **Automate to make architectural experimentation easier**: Replicate infrastructure and track changes by automating infrastructure builds.
- **Allow for evolutionary architectures**: Decouple components to allow for evolutionary changes to each as needed.
- **Drive architectures using data**: Collect data (metrics) and let this drive improvements to your architecture.
- **Improve through game days**: Test failures to identify resiliency improvements and develop organizational experience.
- **Reference the Well Architected Framework**: This is a documented framework for how to “best” implement applications within AWS (or more generically any cloud provider). The framework used to consist of 5 pillars: operational excellence, security, reliability, performance efficiency & cost optimization, but recently a sixth pillar (sustainability) was added.

High Level Concepts

Availability Zone (AZ) – One or more data centers physically close to each other.
Region – Multiple availability zones within the same geo-political area. Reason to choose a reason is due to 1) Proximity to users, 2) Compliance requirement (e.g., data needs to stay in country), 3) Costs/Availability of services.
Virtual Private Cloud (VPC) – Private networking area within a Region. Foundational networking concept in AWS
Edge Locations – Smaller data centers geographically dispersed all over the world – used by CloudFront
Local Zone - Provides you the ability to place resources, such as compute and storage, in multiple locations closer to your end users
AWS Outposts - Brings native AWS services, infrastructure, and operating models to virtually any data center, co-location space, or on-premises facility. An Outpost relies on connectivity to the parent AWS Region. Outposts rack is not designed for disconnected operations.
Tags – Available to be used for just about all AWS services. Allows for naming, grouping of resources, cost allocation, administration of resources, etc.
General guidelines – Understand IaaS, PaaS, and SaaS. If technically feasible, choose PaaS offerings over IaaS, serverless over regular PaaS, automation over manual, IAM Roles over groups and groups over users, and least privilege over broad privilege.
Amazon Resource Names (ARN) – Unique name associated with every resource created in AWS including users, groups, servers, databases. S3 buckets, etc. These names are colon delimited format; for example -> "arn:aws:dynamodb:us-west-1:123456789012:table/myDynamoTable"

Functionality - Networking

NOTES:

1) Most services support IPv6 or both IPv4 & IPv6, but this document focuses on IPv4 as that's still by far the dominant use in AWS.
2) You are charged for data egressing (leaving) AWS, but not data ingressing (entering) AWS.
3) You are not usually charged for data transfers *within* a region (VPC peering, transit gateways, etc. - there are a few exceptions), but you are charged for data transferred *between* regions.

The following is a summary of the AWS networking options/capabilities.

- **VPC** – An AWS VPC supports five (5) private IP address ranges. Each of these ranges can be between /28 (CIDR notation) and /16 in size. The IP address ranges of your VPC should not overlap with the IP address ranges of your existing network to allow for direct communications. FREE SERVICE.

- **Route tables** – One main route table, but up to 200 allowed in a VPC. Route tables are associated to subnets. Allow for automatic routing within VPC. Can include routes to various gateways (Internet, NAT, Customer, Storage, Transit, etc.) or other VPCs (VPC Peering). FREE SERVICE.
- **VPC Peering** – One to one peering between VPCs. Cannot have overlapping IP addresses. Transitive peering (peering through one VPC to another VPC) is not allowed – only one-to-one direct peering is allowed. Can't scale once you have more than 4 or 5 VPCs because too many peering connections are needed. FREE within AZ, minimal costs within region.
- **Transit Gateway** – A hub for many types of connections within, in or out of a region like to VPCs, Direct Connect, VPN, other gateways, etc. Allows for scaling number of VPCs that need to connect to each other. Monthly charge by number of connections and usage.
- **Virtual Private Gateway (VPG)** – Allows connectivity into VPC from an outside network (e.g., on-prem network). You attach a VPG to a VPC and can then route CIDR blocks to the VPG. There are two primary use cases -Direct Connect or Site-to-site VPN connections described below. There is no cost for a VPG, but you are charged for active connections to it (VPN or Direct Connect).
- **Direct Connect** – Private circuits (usually through 3rd party) from on-prem into AWS to a VPG or Direct Connect gateway which in turn can connect to VPGs. Fairly expensive with multi-year commit.
- **Site to site VPN** – Encrypted connections from your on-prem data center to multiple AZs over the Internet. Connects "Customer Gateway" (router on customer on-prem) to VPG or Transit Gateway (AWS side of connection). Fairly inexpensive monthly and usage charges per connection.
- **PrivateLink** – Allows you to create a private endpoint (IP) for AWS public services (e.g., S3, Lambda, SQS, SNS, etc.) as well as on-prem services coming in through a gateway. You place the private endpoint in one or more VPC subnets, protect with security groups and can even apply IAM policies on them. Not all AWS services support PrivateLink, but most do now. Charged by usage (GB transferred).
- **Internet Gateway (IGW)** – Allows access to Internet, attaches to VPC and is routed to (0.0.0.0/0) by a route table. However, for an object (e.g., EC2 instance, ELB, etc.) within a subnet that routes to an IGW to access or be accessed by the Internet, it needs a public IP attached to it. FREE SERVICE.
- **Elastic IP** – This is a public IP that is dedicated to you; it will not change. Limit of 5 Elastic IPs per account, per region. EC2 instances can get FREE public IPs instead of Elastic IPs, but they change whenever the instance is stopped and started

(not when rebooted). ELBs do not need Elastic IPs because a DNS name is provided and a CNAME alias to that name should be used instead. FREE when attached to a running EC2 instance; if the EC2 instance is down or the Elastic IP is not attached to anything, then there is an hourly cost.

- **NAT Gateway** – A device that allows instances in a "private subnet" (a subnet with a route table that does not route to an Internet Gateway and thus where instances only have private IP addresses) to access the Internet by having the route table route 0.0.0.0/0 to the NAT gateway. They do not allow the Internet to initiate a conversation back to the private instances. The NAT Gateway itself must reside in a "public subnet". For resiliency, you can have NAT gateways in multiple AZs. Charged on per-hour basis at all times.
- **Subnet** – Allocated to an AZ with a subset of one of the VPCs CIDR network blocks. A mechanism for segmenting a VPC, especially into AZs or tiers (e.g., ELB, Applications Server, Database Server). 5 IPs within each subnet are reserved by AWS for router, DNS, broadcasts and future use. CIDR ranges can be /16 to /28. FREE SERVICE.
- **Network ACL** – Subnet-based firewall rules associated to one or more subnets. FREE SERVICE.
- **Route 53** – Multiple DNS services as follows:
    - Can define private hosted zone (full authoritative DNS capabilities)
    - Can obtain DNS domain
    - Can act as GSLB with multiple methods of load-balancing:
        - Simple (return all records)
        - Primary/failover with health checks
        - Latency-based (Best for performance)
        - Weighted Routing
        - Geo-proximity
        - Geo-location (Best for compliance)
        - Combination (especially with failover health checks)
    - Can create DNS inbound/outbound resolver endpoints (only necessary if integrating with on-prem DNS servers as each subnet gets a FREE DNS resolver)

  Charged ala cart for each service.
- **CloudFront** – Content Delivery Network for any defined "origin" location (URL but can also be S3 URL). Allows for faster download times of static content. Costs are usage based.

Functionality - Compute

NOTES:

1) Serverless offerings are typically charged by usage (e.g., number of requests), while server-based offerings are typically charged by time (secs).
2) Serverless offerings are preferred when feasible.
3) Auto scaling and multiple AZ deployments are preferred when resiliency is a key goal.
4) Decouple layers so that each can scale independently (e.g., place load balancer or queuing service in between layers of the application).
5) Security best practice is to create separate subnets for “public” load balancer, “private” web/application servers and “private” database servers such that each subnet has security groups limiting access to only the ports and security groups/subnets that need access (e.g., only ports 80/443 for load balancer security group, only port 80 from load balancer to web/application servers and only the DB port from web/application servers to database servers)

The following is a summary of the AWS compute options/capabilities.

- **Amazon Machine Image (AMI)** – A snapshot of an operating system. There are many Amazon base images as well as 3rd party and community images to choose from. No charge for using Amazon or community AMI, but 3rd party AMI’s may come with a charge. If you create your own AMI, you only pay per-GB hourly fee for disk space used by disk storage snapshot holding the AMI.
- **EC2 instance** – Core compute offering; analogous to VM server. Launched from AMI (Windows or Linux base with other SW installed) in a subnet within an AZ. Associated to a security group. Can be associated to an IAM role through an “instance profile” to obtain permissions to access services. Can be provided “userdata”, which is a script that will run when the instance is first launched (can make it run with every reboot) – used to install products, patching, deploy apps, etc. From within the OS of a running EC2 instance, you can read the AWS metadata about the instance (e.g., instance ID, AMI ID, security groups, userdata, public IP, etc.) by querying the EC2 metadata URL (http://169.254.169.254/latest/meta-data/). EC2 instances are typically charged on a per-second basis, but only when instance is running; when stopped, you are only charged for the EBS volume(s) attached. Per-second charges vary depending on instance type (t2, t3, m3), size (micro, small, medium, etc.), region and how the instance was purchased:
  - On-demand – Most expensive, but only billed by the second, if used. Thus, also most flexible purchase option.
  - Spot instances – Cheapest (about 90% savings), but least reliable (can be terminated at any time with only a few minutes of notice)

  - Reserved instances – Major savings (about 70% savings), but need to commit to paying 24x7 costs for 1-3 years
  - Scheduled Reserve instances – Major savings with less time commitment over one year, but you can only run your instance during pre-decided times
  - Dedicated hosts – Isolated physical hardware. Should only be used if necessary due to SW licensing requirements. Can be reserved or on-demand, but more expensive.
  - Dedicated instances – Run on hosts that are isolated to your AWS account. There is an additional hourly charge for these instances, but they can be on-demand, reserved, or spot instances
- **Security Groups** – Host-based firewall rules that can be associated to most compute services (EC2, ELB, Lambda, etc.), database services (RDS, DynamoDB, etc.), storage services (EFS, FSx, etc.), services exposed over PrivateLink, and more within your VPCs. Contain both "Inbound" and "Outbound" rules. Only "Allow" rules can be provided, NOT "deny" rules (there is an implicit DENY ALL included, so if no rules are provided, no access is allowed). Security groups are stateful (they follow conversations, so that reply packets are automatically allowed). Allow rules can provide source as CIDR/security group/ALL, destination protocol and destination port/port range. FREE SERVICE.
- **ELB** – Three types: 1) Application Load Balancer (ALB) – only supports HTTP/HTTPS but is able to split traffic to different targets based on URL; best choice for web traffic; 2) Network Load Balancer (NLB) – Supports ALL protocols, but limited functionality; best choice for non-web traffic; 3) Classic ELB – Supports all traffic, but limited HTTP functionality; not recommended. Charged on hourly basis plus usage.
- **Target group** – Creates group of instances, IP addresses, Lambda functions, etc. that can then be balanced by an ALB or NLB. Include configurable health checks to avoid sending traffic to unhealthy instances. FREE, but only useful with ALB or NLB.
- **EC2 Auto Scaling group / AWS Auto Scaling plan** – EC2 Auto Scaling groups allow the adding and removing of EC2 instances (typically from a target group) based on a min/max capacity adjusted automatically by either Cloudwatch metrics (e.g., CPU exceeds 60% add an instance; CPU goes below 40% remove an instance), fixed schedule and/or predictive scheduling. Launch Templates are needed. AWS Auto Scaling plans extend this capability to ECS services, RDS Replicas, EMR clusters and DynamoDB Tables (read/write capacity units). FREE SERVICE.
- **Launch Template / Launch Configuration** (launch configurations no longer recommended) – You specify AMI ID, instance type & size, VPC, subnet, security group, userdata (commands to execute when instance is launched), purchase option

(on-demand, spot, etc.), disk allocations, etc. This template or launch configuration is then associated with an auto scaling group. FREE SERVICE.

- **ECR –** Allows for the creation and maintenance of a Docker repository of Docker container images. Can be integrated with CodeCommit, CodeBuild, CodePipeline and/or CodeDeploy as part of a CI/CD pipeline. Cost per GB of storage and data transfer out of AWS.
- **ECS** – ECS allows for the creation of clusters, which can be of type EC2 (just EC2 instances with a container agent service to communicate with ECS.), Fargate or Outpost (EC2 instances on your premises with a similar agent). One or more services can be placed in a cluster. A service can be associated with a load balancer and AWS Auto Scaling Plan – this simply launches or stops task instances as needed. A Fargate cluster can be assigned to the AWS Auto Scaling plan such that it would scale vCPUs & storage according to the provided scaling policy. FREE, but you need to pay for the EC2 instances/associated storage or Fargate vCPUs/storage.
- **EKS** – Create a Kubernetes cluster on EC2 instances (node groups) or Fargate. EC2 cluster can use EC2 Auto Scaling groups. The choice between ECS vs. EKS comes down to a few factors – simplicity (ECS), previous Kubernetes knowledge/automation already built (EKS), and vendor lock-in (ECS) vs. portability (EKS). Per-hour costs per EKS cluster in addition to EC2 or Fargate compute and storage costs.
- **Fargate** – Not a standalone service. It provides serverless compute capacity for ECS or EKS containers. Charged by vCPUs & storage allocated.
- **Elastic Beanstalk** – Automates creation of Web/DB stack w/Auto Scaling and ELB (any kind). Highly flexible with options for private VPC, instance types, application platforms (.NET, PHP, Python, Java, Go, Docker, Ruby & Node.js), security groups, monitoring, RDS choices, etc. Supports rolling application updates. FREE service, but you pay for supporting deployed infrastructure (EC2 instances, ELB, S3 buckets, RDS database, DynamoDB databases, etc.)
- **Lambda** – Allows you to create serverless functions written in either .NET, Java, Python, Go, Ruby, or Node.js OR provide a container image with any code. Lambda functions are associated to IAM roles to give them access to other services. Can be associated to VPC, multiple subnets/AZ’s, & security groups. Can be triggered by many AWS services like Step Functions, S3, SQS, SNS, EventBridge, MQ, Kinesis, DynamoDB, ALB, API Gateway, CloudWatch, etc. Can subsequently trigger SQS, SNS, another Lambda function, EventBridge event bus. Logs are automatically written to CloudWatch Logs. Lambda Functions have quota limits that you need to be wary of, especially that each request can only run for 15 minutes (hard limit). Pricing is based on the amount of RAM taken per

second and the number of requests, but it's generally an inexpensive compute function.
- **Batch** – Allows you to run batch jobs in a serverless environment, but only supports Linux (not Windows). FREE SERVICE, but you pay for resources used (Fargate, EC2 instances, storage, etc.).
- **Step Functions** – Allows you to create sophisticated workflow (state machine). You are charged per "State transition" in addition to any other service costs.

Functionality - Storage

NOTES:

1) Intra-region data transfer is usually free, but inter-region data transfer is charged. Similarly, data coming into AWS is usually free, but data leaving AWS is not.
2) Other than S3, all other offerings provide for "block storage" (S3 offers "object storage").
3) All forms of storage support server-side encryption.

The following is a summary of the AWS storage options/capabilities.

- **Instance store** – Useful for temporary, non-redundant, super low-latency, super high throughput SSD block storage for EC2 instances. This is local, direct-attached storage. Instance store data is lost if the instance is stopped, hibernated, or terminated, but not if just rebooted. There is no extra cost for instance stores as they are baked into the per-second cost of the instance type.
- **EBS volume** – Highly durable, network attached block storage within a specific AZ that can be attached to one EC2 instance at a time within that same AZ. Most EC2 instances use EBS for their boot volume. Can persist beyond the life of the EC2 instance. Can create snapshots and then archive them. Can create data lifecycle policy to auto create/archive/delete snapshots. Can restore from a snapshot. Four basic types (General purpose SSD (gp2 or gp3 – best for most use cases), Provisioned IOPS (io1 & io2 – if more than 16K IOPS needed), Cold HDD (sc1 – cheapest & slowest storage) and Throughput optimized HDD (st1 – low-cost throughput-intensive workload). The volume type, size and IOPS (if applicable) can be changed on the fly. File system types supported based on the OS (e.g., NTFS, EXT4, XFS, etc.). Pricing based on volume type and size. Snapshot pricing based on size & whether archived or not.
- **S3** – Highly scalable and durable object storage that is generally multiple times less expensive than EBS, EFS or FSx storage. You create "buckets" which reside within a region. Bucket names must be "globally unique". Folders and objects can be created within a bucket with a key (directory path and/or filename). An object (file) can be up to 5TB in size and there is no limitation to the size of a bucket.

Objects can be access via HTTP, CLI or S3 APIs. Objects are stored in any of a few storage classes as follows:

- Standard – Data is replicated across 3 AZs; most expensive and flexible option
- Infrequent Access (IA) – 45% savings, but 30-day minimum and higher data retrieval costs
- One-zone IA – 20% additional savings over IA with same 30-day minimum and data retrieval costs
- Intelligent Tiering – Automatically moves objects (>128KB) from standard to IA to Glacier to save on costs

Some key S3 features:

- Versioning – Keeps as many versions you define of an object; can be turned on/off at any point, but versions persist unless explicitly deleted
- Encryption – Server-side (objects encrypted/decrypted automatically after they arrive) or client-side (objects encrypted before sent to S3).
- Object lock – Disallows deletion of files until certain amount of time has passed; Must be enabled when bucket is created
- Static website hosting – Ability to use S3 as a static website (no server-side code)
- Lifecycle rules – You set rules as to when objects or object versions should be moved to another object class, Glacier or deleted based on file age and number of versions
- Replication rules – You can set rules for replicating objects to other buckets in other regions
- ACLs – Control which principals can do what with objects
- Block public access – Unauthenticated users cannot access objects
- Create pre-signed URL for an object – Random link for which authentication is not required, but that’s only valid for a short time
- Multipart Upload - You can upload parts in parallel to improve throughput; smaller part size minimizes the impact of restarting a failed upload due to a network error. Recommended if you're uploading large objects or over a spotty network.
- Can create event notifications when objects are created, removed, etc.
- Can enable transfer acceleration that uses Cloudfront to find fastest path to client uploading data to S3

- **S3 Glacier** - Highly scalable and durable object storage that is 60% to 90% less expensive than S3 IA. While storage is much less expensive, data retrieval costs are much greater when the data is needed. Tightly integrated with the S3 Lifecycle rules. There are three storage classes:

  - S3 Glacier Instant Retrieval – Highest cost, but data is immediately available when needed
  - S3 Glacier Flexible Retrieval (Formerly S3 Glacier) – 10% savings over Instant Retrieval, but with 1-minute to 12 hours retrieval times
  - S3 Glacier Deep Archive – Over 70% savings from Flexible Retrieval, but 12-hour retrieval times
- **EFS** – Network shared file system for Linux (Windows not supported) – designed for multiple system access and can be available through multiple AZs. Need to install "amazon-efs-utils" on Linux before being able to mount the file system. Pricing is based on GB/month and data transfer requests as well as whether filesystem is placed in one-AZ (1/2 the cost) or multiple AZ's.
- **FSx** – 4 Additional network shared file system offerings (NetApp OnTap, Open ZFS, Windows File Server, & Lustre). NetApp OnTap supports all protocols (NFS, SMB, etc.), Windows & Linux. ZFS allows for high throughput & IOPs supporting Linux & Windows, but only supports NFS and single AZ. Windows File Server is limited to 64TB, has limited throughput & IOPs, only supports SMB, and requires Active Directory. Lustre supports super-high throughput & IOPs and is fairly inexpensive, but only supports Linux with customized protocol. All offerings have large size minimums raising the starting price as opposed to EFS. Pricing based on offering chosen, type of storage (SSD, HDD, Backup), GB/month, throughput capacity selected, and cross-region data transferred per month (FREE data transfer within region).
- **Backup** – Allows for the creation of backup plans, including schedules for RDS, Aurora cluster, DynamoDB, DocumentDB, EBS, EFS, FSx, S3, Neptune, Storage Gateways and On-prem VMWare VMs. FREE service, but you pay for the storage taken up by the backups.
- **Storage Gateways** – Four options for creating on-prem virtual appliance to interact with AWS storage. For each you download a virtual server template and start it up on your on-prem virtual infrastructure - supports VMWare, Hyper-V, Linux KVM, Amazon EC2 (for outposts), or you can even order a hardware appliance from AWS. 1) S3 File Gateway allows you to store and access objects in Amazon S3 from NFS or SMB file data with local caching. 2) FSx File Gateway – Allows you to access fully managed file shares in Amazon FSx for Windows File Server using SMB. 3) Volume Gateway – Allows you to store and access iSCSI block storage volumes in Amazon S3 – these are EBS snapshots backed by S3. 4) Tape Gateway – Allows you to store virtual tapes in Amazon S3 using iSCSI-VTL, and store archived tapes in Amazon S3 Glacier Flexible Retrieval or Amazon S3 Glacier Deep Archive. Costs are mostly related to the backend storage used and written, but for the FSx File Gateway, there is also a per-hour gateway cost.

- **Snowball Family** – These are physical devices that you order from AWS to copy large amounts of storage on to them so that you can then ship them back to AWS to load into the cloud OR vice versa. All products are encrypted. Three products: 1) Snowcone (only up to 8TB or storage, but very sturdy – meant for extremely remote conditions with poor network connectivity), 2) Snowball (Suitcase sized and stores up to 100TB – comes in storage optimized and compute optimized options, 3) Snowmobile (Truck that stores up to 100PB – best option if you need to transfer more than 10PB of data). Pricing is usually based on days of usage and the amount of data transferred out of AWS (if any).

Functionality - Database

NOTES:

1) When data has stable schema and is highly relational (lots of joins between tables), then a relational database (RDS, Aurora) is needed.
2) When you need a dynamic schema and eventually consistent data, then a NoSQL database is best (DynamoDB, DocumentDB). Unless MongoDB compatibility is needed, DynamoDB is better choice.
3) If you need a relational data warehouse, then Redshift is the answer.
4) If you need a graph database, then Neptune is the answer.

The following is a summary of the AWS database options/capabilities.

- **RDS** – Service that allows you to create & manage relational databases more easily. There are several options – Oracle, SQL Server, MySQL, Postgres, MariaDB, Aurora (AWS proprietary implementation of MySQL or Postgres) or Aurora Serverless (also MySQL or Postgres). RDS has many options (thousands of pages of documentation), and they vary by the database type being deployed. Some key features include:
    - AWS managed backups, software patching, automatic failure detection, and recovery, but you can manually create your own backup snapshots. You can also choose a maintenance window.
    - Database instances can be deployed to private or public subnets within your VPC protected by security groups.
    - Except for Aurora MySQL (can have multi-master setup), there is only one read/write (primary) instance for each database, but you can have multiple read replicas in multiple AZs (read replicas are good for offloading report processing). One of the read replicas can also be setup for automatic failover.
    - You can control who can access your RDS databases by using native database users/passwords (all but Aurora), AWS IAM (all but Oracle,

MySQL or SQL Server), Windows authentication (SQL Server only) or Kerberos (Aurora & Postgres only). Encryption at rest supported.

Pricing dependent on many factors like size of server (or ACUs used in the case of Aurora Serverless), number of replicas, amount of storage (primary & backups), type of DB (Oracle & SQL Server require licensing).

- **RDS Proxies** – Provide connection pooling for RDS to protect the backend database from oversubscription. Only supported for MySQL and Postgres RDS databases.
- **DynamoDB Tables** – Service that allows you to create NoSQL tables with a flexible schema. You need to define a primary key that uniquely identifies an item. Capacity can be “on-demand” where you are charged per write and/or read OR capacity can be set to a min/max number of RCUs (read capacity units) and WCUs (write capacity units), if you feel that the workload is predictable. Here are some key features:
    - Auto scaling of compute and storage
    - Automatically replicated across multiple AZs
    - Automated backups can be enabled
    - Up to 20 secondary global indexes (additional partition key and optionally sort key) and up to 5 local indexes (additional sort key against primary partition key) can be added. Local indexes must be added when table is created.
    - Access to tables can be granted via IAM policies. A private VPC endpoint (Private link) can be created for DynamoDB tables.
- **DAX Clusters** – DynamoDB Accelerator (DAX) is meant to be a “cache” for DynamoDB tables. You choose the size of the cluster and the number of nodes spread out across multiple AZs.
- **DocumentDB** – MongoDB compatible NoSQL database offering.
- **Redshift** – Relational database data warehouse service – enormous nodes/very expensive.
- **Glue** – A serverless ETL (Extract, Transform & Load) tool that can be used to take data from source systems and place it in something like Redshift.
- **Neptune** – Graph database for highly connected data.
- **Redis Cache –** Simple key/value pair database good for session state and other simple structures. *Can* be made resilient across multiple availability zones.
- **Memcached -** Simple key/value pair database good for session state and other simple structures. *Cannot* be made resilient across multiple availability zones.
- **EMR** – Elastic Map Reduce offering for advanced data analytics. EMR supports several families of applications including variations of Hadoop, HBASE, Apache Spark, Hive, Presto, etc.

Functionality - Security

NOTES:

1) Shared responsibility security model – AWS is responsible for the security of the cloud (facilities, hardware, virtualization, PaaS services, etc.), you are responsible for the security of what you put in the cloud (firewall settings, access rights, patching IaaS operating systems, etc.).
2) Remember that least privilege is always preferred.
3) If technically possible, IAM roles are preferred.
4) If IAM roles do not apply, then groups are preferred over individual users.
5) When determining if an action is allowed, all relevant policies are considered. If there is an applicable "Deny" then access is denied. If there is no "Deny" and an applicable "Allow" then access is allowed. If there is no applicable permission statement, then access is denied implicitly.

The following is a summary of the AWS database options/capabilities.

- **Root account** – Email address you used to create your account. You should enable MFA on this account immediately and only use it when absolutely necessary. You cannot restrict this account. You should create an IAM admin user and begin to use that account for regular administrative duties.
- **IAM user –** Individual user account that is associated with a password as well as potentially MFA devices and/or signing certificates. Users can also have Access Key with associated secret key for programmatic access through AWS CLI or SDK. The limit is 5000 users per account. This is a free service.
- **IAM group –** A IAM group is made up of IAM users. Groups cannot be nested (only IAM users can be a part of a group). The max number of users in a group is 5000. The max number of groups is 300 but can be increased to 500. It is best practice to assign permissions to groups instead of users. This is a free service.
- **IAM role** – Account that can be used on a temporary basis by either an AWS service (e.g., EC2 instance, Lambda function, etc. – many services can use an IAM role for temporary permissions), users from another AWS account, or federated web identities. Roles can be granted permissions in the same way that IAM users and groups can be granted permissions. The default maximum number of roles is 1000 but can be increased to 5000. For terminology understanding - when an IAM role is associated to an EC2 instance, it's associated to the "instance profile" for that EC2 instance; it's semantics because an instance profile can only have one IAM role. This is a free service.
- **IAM Policies** – A JSON formatted document that is attached to users/groups/roles to grant them permissions to specified resources. IAM policies have 4 sections – "Effect" (Allow or Deny), "Action" (the action in affect for this permission – for example dynamodb:DeleteItem), "Resource" (ARN of the resource being

addressed – can include wildcard *), “Condition” (only apply this permission if certain conditions apply). The “Principal” is NOT included in IAM policies because the policies are attached to a principal (IAM user/group/role). There are many out-of-the-box AWS policies to choose from or you can create your own. The default limit for number of custom policies is 1500 but it can be increased to 5000. By default, only 10 policies can be attached to a user, group or role but limit can be increased to 20. This is a free service.

- **Resource Policies** – A JSON formatted document that is attached to a resource to determine which principals can have what permissions. The format is the same as an IAM policy except that there is an additional section called “Principal” to specify the ARN of the users, groups or roles for whom this document is applicable – wildcards are allowed. This is a free service.
- **IAM Identity Center (formerly AWS SSO)** – Allows you to assign your existing users accounts in Microsoft AD, Azure AD, Okta, PingIdentity, and other such user repositories to IAM accounts that can then be granted access to AWS resources like any other IAM user or role. By default, AWS Identity Center is limited to 50K users and 10K groups, but these values can be increased. This is a free service.
- **Cloudtrail** – Creates a log of management, data or insights. For management events (all AWS CLI/API calls on any resource), AWS automatically creates and keeps these free of charge for 90 days and they are available through the console or Cloudtrail API – you can keep these longer but must send them to an S3 bucket that you would then pay for in terms of storage. Custom Cloudtrail logs must be associated to an S3 bucket. Custom Cloudtrail logs can also be sent to Cloudwatch logs to alert on metrics (e.g., certain events). Custom Cloudtrail logs can also be enabled for “log file validation” which ensures that files are not tampered with. All Cloudtrail logs are encrypted at rest. For data events, you specify the resource on which you want to perform logging and whether on reads, writes or both. You pay for any custom Cloudtrail log by the number and type of events logged.
- **AWS Certificate Manager (ACM)** – Request a public certificate signed by AWS for Internet usage OR a request a private certificate created by AWS Certificate Manager Private Certificate Authority for internal use OR import a certificate generated by another certificate authority. Regardless of the method, once the certificate exists in ACM, it can be used by AWS services like ELB, API Gateway, Cloudfront, Cognito and many more. Imported certificates must be manually renewed and imported into each region it's needed, which is a challenge with services like Cloudfront. Public and imported certificates are free, but you are charged for OCSP (check for revocation) calls, if enabled. Private certificates are charged on a per certificate basis.

- **Cognito** – Service that allows for the management of large numbers of users through User Pools (authentication) and Identity pools (authorization). With user pools, you can define how users can choose a username, rules for their password, how to recover their password, how long temporary passwords are good for, whether to require MFA, customize emails for password recovery, etc. By default, the limit is 40M users per user pool, but this can be increased. Identity pools can be associated to unauthenticated users or users authenticated through Cognito, Google, Facebook, Apple, OpenID and others and then associate them to an IAM role, which can then be provided resource permissions. Cognito User Pools are charged by Monthly Active Users (MAUs) while Identity Pools are free of charge.
- **Inspector** – Vulnerability management tool (like Qualys). There is a "modern" and "classic" version. Scans EC2 instances and containers (not in "classic" offering) for vulnerabilities (installed software versions known to have vulnerabilities). Multi-account support (not in "classic"). Uses Systems Manager Agent ("classic" version has its own agent). Supports continual scanning (not in "classic"). Windows only supported in "classic" version at this time. Integrates with Security Hub. Priced by number of EC2 instances and containers scanned.
- **Macie** – Sensitive data detection (PII, SSN, PCI, etc.). Pricing by S3 buckets used and GB analyzed.
- **GuardDuty** – Network IDS offering. Basic threat detection by analyzing VPC flow logs, CloudTrail logs, DNS logs, S3 events, EKS audit logs, etc. Does not perform deep packet inspection. Sends findings to AWS Detective and AWS Security Hub. Priced by number of events analyzed or GBs of logs analyzed.
- **Detective** – Collects log data from your AWS resources and uses machine learning, statistical analysis, and graph theory to help you visualize and conduct faster and more efficient security investigations. Analogous to Exabeam. Priced by GBs analyzed.
- **Security Hub** – SIEM (Security information and event management) like offering. Provides a consolidated view of your security status in AWS. Automate security checks, manage security findings, and identify the highest priority security issues across your AWS environment. Integrates with AWS Config, GuardDuty, Detective, Inspector, Macie, Systems Manager (patching), Trusted Advisor, and more, including 3rd party SW. Priced by number of security checks executed and findings ingested.
- **Simple AD** – Supports up to 500 (small version)/5,000 (large version) users and 2,000/20,000 objects. Does NOT support MFA, trust relationships with other AD domains, AD Admin Center or PowerShell. Supports LDAP. Priced by size (small or large), but fairly inexpensive.

- **Microsoft AD** – Full Active Directory support. Comes in two sizes (Standard & Enterprise) which vary in terms of storage (1GB/17GB), objects(30K/500K) & pricing.
- **AD Connector** – Allows connectivity to on-premises AD. Requests are simply proxied back to on-prem AD without caching any data in the cloud. Two sizes (small & large) that vary in terms of requests/second allowed and pricing.
- **Cloud Directory** – Allows for the creation of a proprietary directory. Uses proprietary HTTP API interface, not LDAP. Priced by reads/writes – inexpensive.
- **WAF –** Web Application Firewall offering. Protects web applications from well-known attacks like SQL-injection, cross-site scripting, bot control, Captcha challenges, etc. Can be place in front of any web service, like EC2 instance, ALB, CloudFront, API Gateway. You can subscribe to marketplace rulesets to enhance protections. Includes AWS Shield standard (DDoS protection). Pricing by number of rules, number of Web ACLs & requests protected.
- **Shield** – DDoS protection offering. Standard version protects from layer 3 & 4 attacks. Advanced version adds SRT (Shield Response Team – security experts engaged in case of attack) and more sophisticated defenses. Pricing by data transferred out per month.
- **AWS Secrets Manager** – Password manager where you provide usernames/passwords that can be used to log into databases or APIs (key/value pairs). Allows you to specify a Lambda function to automate the rotation of the secret. Secrets can be replicated to other regions. Charged per secret and the number of times the secret is used.
- **Network Firewall** – A stateful firewall used to protect a subnet within a VPC. This is analogous to Network ACLs, but stateful like security groups instead and not free. Pricing is based on how many hours each subnet is protected in addition to the amount of data that traverses the firewall. Generally not worth the expense – security groups are preferred.

Functionality – Automation, Monitoring, Administration & Other

NOTES:

The following is a summary of the AWS database options/capabilities.

- **The Well Architected Framework (tool)** – This is a documented framework for how to “best” implement applications within AWS (or more generically any cloud provider). The framework used to consist of 5 pillars (operational excellence, security, reliability, performance efficiency & cost optimization), but recently a sixth pillar (sustainability) was added. For each pillar, the document provides

Design Principles, Best Practices and then relevant questions to ask to determine for each best practice. AWS also offers an online questionnaire called the "Well Architected Framework Tool". This is a FREE service.

- **Billing** – Provides costs & usage reports, "Cost Explorer" a can deep dive into the costs (what services in what regions are driving costs). Savings plans (pre-payments to obtain savings on services), Budget alarms, Cost allocation tags, etc. also available. This is a FREE service.
- **AWS Organizations** – Allows you to consolidate multiple AWS accounts within one hierarchical structure. You can then limit the services each account can use with SCPs (Service Control Policies). SCPs cannot grant user permissions – they can only limit the services allowed by an entire account. This is a FREE service.
- **AWS Artifact** – Provides static compliance-related reports for customers to share with 3rd parties as needed. This is a FREE service.
- **Trusted Advisor** – Warns of service limits being approached, bad security practices, idle resources, fault tolerance & performance shortcomings, provides cost optimization guidance real-time. Functionality is limited depending on purchased support plan.
- **AWS Support plans** – *Basic* (Free – can only open cases about billing, accounts, and service quotas over web; only security and service quota Trusted Advisor checks), *Developer* (about $30/month – same + best practice guidance, single user unlimited web support cases during business hours), *Business* (% of AWS usage – unlimited 24x7 phone, web & chat support – all Trusted Advisor checks), *Enterprise* (% of AWS usage – same + faster responsiveness on issues & Technical Account Manager).
- **Systems Manager** – Collection of tools to help manage your EC2 instances and even on-prem VMs. There's a lot you can do with Systems Manager. Here are some features:
    - Operations Management – Create and track tickets for staff to work on (OpsItems), and automatically generate problem tickets (incidents) from Cloudwatch alerts.
    - Change Management – Register, approve & track changes with change calendar. Create automation scripts that can be executed as changes. Create maintenance windows to limit the timeframe when certain changes can be performed.
    - Applications Management – Use "Parameter store" to create parameters you can then reference in your applications (e.g., database login credentials). You can use "AppConfig" to deploy application configurations, but OpsWork & CodeDeploy do the same.

  - Node Management – Automate patching and verify compliance of patching. Create one-off commands to run on many systems. Create inventory reports. Log into any of your managed EC2 instances with Session Manager.
- **AWS Config** – Provides pre-defined conformance packs to validate infrastructure. Only checks AWS configurations, not inside applications/servers.
- **Cloudwatch** – Monitoring, Alerting and Logging service. Can create alarms based on any metric. Triggered alarms can take all sorts of actions.
- **EventBridge** – Formerly Cloudwatch Rules. Supports event-driven applications. Events can be triggered by any of hundreds service events OR by a scheduled time of day/week/month. Triggered events can then call many services.
- **X-Ray** – Instrumentation tool useful for troubleshooting application issues.
- **MQ** – Apache ActiveMQ & Rabbit MQ compatible service with support for standard queuing protocols.
- **SQS** – Provides proprietary queuing service to decouple applications - each message is intended to only go to one receiver/consumer. The consumer of a message must "pull/poll" from the queue.
- **SNS** – Provides "Push" service. As soon as messages are received, they are sent to multiple consumers (no consumer polling). SNS topics can be standard or FIFO (throughput limited, but guaranteed order). Priced by usage (number of notifications) and data transferred.
- **SES** – AWS Email service.
- **CloudFormation** – Infrastructure as Code service (Elastic Beanstalk and a few other services use CloudFormation behind the scenes). You define your infrastructure as JSON or YAML (called stacks) and CloudFormation will automatically create the infrastructure within AWS when stack is created. You can define scripts to execute on EC2 instances through the AWS::CloudFormation::Init type which is referenced by cfn-init helper scripts. FREE service: you pay for the cloud services you provision.
- **OpsWorks** – Puppet & Chef implementation support. Infrastructure and Code Deploy. Create stacks like CloudFormation but limited to EC2 instances and installation packages installed on them, not other AWS resources.
- **CodeCommit** – Source code repository
- **CodeArtifact** – Share packages for others to use
- **CodeStar** – Hub for developers to work on AWS projects; integrates with other Code* developer tools
- **CodeBuild** – Allows developers to automate the build of their applications
- **CodeDeploy** – Allows developers to automate the deployment of their applications
- **CodePipeline** – Allows developers to automate entire CI/CD pipelines
- **Kinesis Data Streams** – Stream large amounts of data for real-time analysis (click streams, logs, IoT telemetry, etc.).

- **Kinesis Data Firehose** – Firehose is a "delivery stream" in that it sends data to any of a bunch of endpoints. You can directly write to a Firehose stream from similar producers as those supported by Kinesis Data Streams or it can ingest a Kinesis Data Streams as it's input. Firehose can then send this stream to any of a bunch of endpoints.
- **Kinesis Data Analytics** – A service that allows you to run Apache Flink to process data streams for real-time analytics (like a real-time dashboard).