AWS Certified Cloud Practioner Exam Notes

Matt Waismann

July 29, 2021

1 Introduction

1.1 Exam Blueprint

The exam validates ability to:

- Understand the value of AWS Cloud
- Understand and expalin the AWS shared responsibility model
- Understand AWS Cloud security best practices
- Understand AWS Cloud costs, economics, and billing practices
- Identify most core services
- Identify common AWS use cases

Exam conent:

- Multiple choice (out of 4 questions)
- Multiple response (you will be given how many answers of 5 or more are correct)

Domain Areas:

- Cloud Concepts (26%): AWS cloud value propostion, cloud economics, and cloud architecture design principles
- Technology (33%): Methods of deployments, global infrastructure, AWS services, and technology support
- Security and Compliance (25%): shared responsibility model, security and compliance concepts, and access management capabilities
- Billing and Pricing (16%): Comparing pricing models, account structures, billing, pricing, and billing support resources

Minimum passing score: 700 (70%)

2 Foundations of Cloud Computing

2.1 Understanding Cloud Computing

AWS has thousands of servers grouped together in places called **Data Centers**. Cloud computing is the delivery of computing services over the internet through these servers. Common categories are:

- Compute: EC2 and Lambda
- Networking: VPC and Direct Connect
- Storage: S3 and EBS
- Analytics: Athena and Redshift
- Development: Cloud9 and CodeCommit
- Security: IAM and Macie
- Databases: RDS and DynamoDB

There exists a whitepaper called Overview of Amazon Web Services that is 72 pages long and gives details on all the services. To maximize the use of a server, AWS allows you to share a AWS server with other customers through a process called **Virtualization**. Virtualization lets you divide hardware resources on a single physical server into smaller units called **Virtual Machines**, each with its own storage, OS, and network connection. The usage of these services is billed **On Demand** (no long-term contracts or upfront payments) and **Pay as you Go** (billed by the hour or second of usage).

2.2 Advantages of Cloud Computing

There are 6 advantages to cloud computing:

1. Go global in minutes - AWS allows applications to be deployed to multiple regions at the click of a button
2. Stop spending money running and maintaining data centers
3. Benefit from massive economies of scale
4. Increase speed and agility - This gives faster time to market.
5. Stop guessing about capacity - Your capacity is matched exactly to your demand
6. Trade capital expense for variable expense - Instead of the upfront costs of data centers you pay for what you use when you use it

Here are the benefits in technical terms:

1. High Availability - A system which operates continuously without failure for a long time
2. Elasticity - You don't need to plan ahead of time with how much capacity you need. You can provision only what you need, and then grow and shrink based on demand
3. Agility - AWS services help you innovate faster and give a faster speed to market
4. Durability - Durability is all about long term data protection. This means your data will remain intact without corruption

2.3 Cloud Computing and Deployment Models

There are 3 common cloud computing models:

1. Infrastrucutre - Fundamental building blocks that can be rented e.g. EC2. Analogy: web hosting
2. Software as a Service (SaaS) - Complete Applications e.g. SageMaker which does Machine Learning for you. Analogy: email provider
3. Platform as a Service (PaaS) - Used by developers. Analogy: A service gives you tools to build a storefront website

There are 3 common cloud deployment models:

1. Private Cloud (aka "on-premises") - Exists in your internal data center. Doesn't offer the advantages of cloud computing
2. Public Cloud (e.g. AWS) - Offered by AWS. You get all the benefits listed earlier
3. Hybrid Cloud - A mix of private and public cloud. Highly sensitive data is locally stored but the app that runs on that server is run on AWS services. AWS **Direct Connect** links internal data centers with AWS services

2.4 Leveraging the AWS Global Infrastructure

Regions are physical locations. AWS logically groups its Regions into **geographic locations** (e.g. US West, US East, Europe, South America, Asia Pacific). It is best practice to use regions close to where the users of the services will be. Regions have several characteristics. Each region is fully independent and isolated (i.e. if one region is impacted, the others will not be) and regions are resource and service specific (i.e. your services live in a region and cannot necessarliy be replicated across other regions)

Availability Zones consist of one or more physcially separated data centers, each with redundant power, networking, and connectivity, housed in separate facilites. An example:

- Geographic Location: US East
- Region: Ohio
- Availability Zone: 2B

N. Virgina Region has 6 Availability Zones. In Availability Zone US-EAST-1B there are 4 data centers. Each Availability Zone has multiple data centers.

characteristics of AZs:

- Physcially separated (different power grids)
- All AZs in the same region are connected through low-latency links
- Fault tolerant. If one AZ fails the others won't
- Allows for high availability. If one AZ fails your application can still run on another one

Edge Locations - There are way more Edge Locations than AZs and Regions. These Edge Locations are like mini data centers that cache content instead of launching resources like EC2. They servce to reduce latency and speed up delivery of your applications. **Latency** is the time that passes between a user request and the resulting response.

2.5 Exploring Your AWS Account

The **AWS Management Console** allows you to access your AWS account and manage applications running in your account from a web browser. You will see a Region at the top right and the services listed in the management console will be the services available in that region. The **Root User** is created when you intially sign up your account. This user has access to everything, therefore it is best practice to almost never use the root user and instead create a seperate user for your day-to-day activities. The Root User will need to be used to delete a AWS account. Protect the root user with **Multi-Factor Authentication (MFA)**.

Within the **Identity and Acess Managment (IAM)** service you can see all your users (including the root user) and all security settings.

Virtual Private Cloud is a service that you should set up right away. This is a way to create a secure private network (your own slice of the cloud) and

within the network you deploy the resources you want to protect. If you deploy a resource and don't select a VPC there's a VPC already set up for you and AWS will automatically place the service within that VPC.

AWS Command Line Interface (CLI) allows you to access your AWS account through a **terminal** or **command** window. Developers use the CLI more than the console.

Programmatic Access provides access to your AWS resources through an application or a tool like the CLI. Example of programmatic ways to access your account:

1. CLI - a terminal session
2. Application Code - AWS services can be accessed from application code using SDKs and programmatic calls
3. **Software Development Kits (SDKs)** allow you to access AWS services from popular langauges from popular languages like Python, Java, C#, and more

3 Technology

3.1 Exploring Compute Services (EC2)

EC2 allows you to rend and manage virtual servers in the cloud. You have elastic compute power (adjusts based on need) and it is compute/processing power in the cloud. **Servers** are the physcial compute hardware running in a data center. EC2 **instances** are the virtual servers running on these physical servers. Insatances are not considered serverless. More on EC2:

1. You can provision an EC2 instance at the click of a button
2. You can use a preconfigured template called an **Amazon Machine Image (AMI)** to launch your instance.
3. You can deploy your applications directly to your EC2 instances
4. You recieve 750 compute hours per month on the Free Tier plan

Use cases for EC2:

- Deploy a database to EC2 which gives you full control over the database
- Deploy a web application to multiple AZs

Methods to Access an EC2 Instance:

- AWS Management Console - configure and manage instances through web browser
- Secure Shell (SSH) - establish a secure connection to your instance from your local laptio
- EC2 Insance Connect (EIC) - EIC allos you to use IAM policies to control SSH access to your instances, removing the need to manage SSH keys.
- AWS Systems Manager - Systems Manager allows you to maange your EC2 instances via a web browswer or CLI

The most common way to connect to Linux EC2 instances is via Secure Shell (SSH):

1. Generate a key pair - private key and public key which proves your identity when connecting to the EC2 instances
2. SSH client on laptop uses the private key and the EC2 instance uses the public key

EC2 Pricing Options:

1. **On-Demand** - the typical model where you use a fixed price which is billed down to the second. Pay for what you use. Use an on demand instance when:
   (a) You care about low cost without any upfront payment
   (b) YOur applications have unpredictable workloads that can't be interrupted
   (c) Your applications are under Development
   (d) Your workloads will not run longer than a year
2. **Spot instances** - let you take advantage of unused EC2 capacity for a very nice discount (90%). Your request is only fulfilled if capcity is available. Use Spot Instances when:
   (a) When you are not concerned about the start or stop time of your application
   (b) Your workloads can be interrupted
   (c) Your application is only feasible at very low compute prices
3. **Reserved Instances** - RIs allow you to commit to a specific instance type in particular Region for 1 or 3 years. Use a RI when:
   (a) Your applicatoin has steady state usage and you can commit to 1 or 3 years
   (b) Upfront payment for a discount (All Upfront, Partial Upfront, No Upfront)

(c) Your applicaton requires a capacity reservation

4. **Dedicated Hosts** allow you to pay for a physical server that is fully dedicated to running your instances. Use a Dedicated Host when:

   (a) You want to bring your own server-bound software liscense from vendors like Microsoft or Oracle

   (b) You have regulatory or corporate compliance requirements around tenancy model (**Multi-tenancy** is when the server is not shared with other customers.)

5. Remember a Dedicated Host is different from a Dedicated Instance (One is on the full server the other is on a virtual machine or instance that runs on the host)

6. Savings plan - allows you to commit to compute usage (measured per hour) for 1 or 3 years. This is not a RI because your only commiting to a certain amount of usage not a specific instance. A savings plan is not a commitment to a dedicated host, just to compute services like EC2, Fargate, and Lambda. Use Savings Plans when:

   (a) You want to lower your bill across multiple compute services

   (b) You want the flexibility to change compute services, instance types, operating systems, or Regions

Features of EC2:

1. **Elastic Load Balances** - Balances the load (requests/traffic) across servers. The types of load balancers are **Classic, Application, Gateway, and Network**

2. **EC2 Auto Scaling** - adds or replaces EC2 instances automatically across AZs, based on need and changing demand. This is the concept of **horizontal scaling** - adds or replaces EC2 instances automatically across AZs, based on need and changing demand. This is not **Vertical Scaling** which is upgrading an EC2 instance by adding more power (CPU, RAM) to an existing server.

3.2 AWS Lambda

Lambda is a serverless compute service that lets you run code wihout managing servers. You other application code, called **functions**, using many popular languages. Lambda is serverless (don't have to manage serveres like with EC2) and it scales automatically. Developers love Lambda because they don't don't need to worry about patching, provisioning, and scaling servers. **Serverless** simply means AWS manages the servers for you and you cannot access them.

Use cases:

1. Real-time file processing - A CSV is uploaded to S3 bucket then the upload triggers a Lambda function to read the file and store that data in a DynamoDB Table.
2. Sending email notifications - An action can trigger a lambda function which triggers SNS which sends an email
3. Backend business logic

Features:

1. Supports languages like Java, Go, PowerShell, Node.js, C#, Python, and Ruby
2. You author code using your favorite IDE or via the console
3. Lambda can execute your code in response to events
4. Lambda functions have a 15-minute timeout

Pricing Model:

1. Compute time
2. Request count
3. Free tier for 1,000,000 requests

3.3 Additional Compute Services

Fargate is a serverless compute engine for containers. Fargate allows you to manage containers like Docker. Fargate scales automatically and is serverless.

Lightsail allows you to quickly launch all the resources you need for small projects. Deploy configured applications, like WordPress websites, at the click of a button. Simple to use for people with no cloud experience. Includes a virtual machine, SSD-based storage, data transfer, DNS management, and a static IP. It's designed for small applications.

AWS Outposts allows you to run cloud services in your internal data center. AWS delivers and installs servers in your internal data center. Supports workloads that need to be on premises. Supports the hybrid deployment model.

AWS Batch allows you to process large workloads in smaller chunks (or batches). Runs hundreds and thousands of smaller batch processing jobs. Dynamically provisions instances based on volume.

3.4 Leveraging Storage Services: S3

S3 is an object sotrage service for the cloud that is highly available. **Objects** (or files) are stored in **buckets** (or directories). Millions of objects can be held per bucket. Objects can be public or private. Objects can be uploaded through code, the CLI, or the console. You can set security at the bucket level or object level using **access control lists (ACLs)**, **bucket policies**, or **access point policies**. You can enable **versioning** to create multiple versions of your file in order to protect against deletion or access a previous version. You can also use **S3 access logs** to track the access to your bucket. S3 is regional service but each bucket name must be globally unique. You can also setup S3 to have your data replicated across regions.

S3 Storage Classes:

1. S3 Standard - General-purpose storage, recommended for frequently accessed data
2. S3 Intelligent-Tiering - Automatically moves your data to the most cost-effective storage class, recommended for data with unkown or changing access patterns
3. S3 Standard Infrequent Access - Data accessed less frequently but requires rapid access, a bit cheaper than S3 Standard, recommended for long-live data that is infrequently accessed but millisecond access when needed.
4. S3 One Zone-Infrequent Acess - Like the above but all the data is in one availability zone so if that AZ gets destroyed the data will be gone, recommended for data that is easy to recreate, infrequently accessed
5. S3 Glacier - Long-term data storage and archival for lower costs. Data retrieval takes longer, can take several hours, reccomended for long term backups, very cheap.
6. S3 Glacier Deep Archive - Like S3 Glacier but longer access times but retrieval is 12 hours or 48 hours, cheapest option of all, reccomended for data on regulatory compliance.
7. S3 Outposts - provides object storage on-premises, reccomended for data that needs to be kept local, demanding performance needs.

Use cases:

1. Static websites - deploy static websites to S3 and use CloudFront for global distribution.
2. Data archive - Archive data using Amazon Glacier as a storage option for S3

3. Analytics systems - Store data in Amazon S3 for use with analytics services like Redshift (Data Warehousing) and Athena (Query your data in S3 using standard SQL)
4. Mobile applications - Mobile application users can upload files to an Amazon S3 bucket

3.5 Additional Storage Services

EC2 supports several storage options for your instances:

1. **Elastic Block Store (EBS)**
    - EBS is a storage deice (caslled a **volume**) that EBS be attached to (or removed from) your instance (like a flash drive).
    - Data persists when the instance is not running
    - Tied to one AZ
    - Can only be attached to one instance in the same AZ
    - Use Cases: Quickly accessible data, running a database on an instance, long-term data storage
2. **Instance Store**
    - An instance store is a local storage that is physically attached to the host computer and cannot be removed
    - Storage on disks physically attached to an instance
    - Storage is temporary since data loss occurs when EC2 instance is stopped
    - recommended for: temporary storage needs, data replicated across multiple instances
3. **Elastic File System**
    - EFS is a serverless network file system for sharing files
    - Only supports the Linux file system
    - More expensive than EBS
    - Accessible across different AZs in the same Region
    - Use case: Main directories for business-critical apps, Life-and-shift existing enterprise applications

Storage Gateway is a hybrid storage service.

- Connect on-premises and cloud data
- Supports a hybrid deployment model
- Use cases: Moving backups to the cloud, reducing costs for hybrid cloud storage, low-latency data storage

3.6 Understanding Content Delivery Systems

A **content delivery network (CDN)** is a service that ensures fast content download. More specifically, a mechanism to deliver content quicly and efficiently based on geographic location. A CDN provides low latency.

Amazon CloudFront is a CDN that delivers data and applications globally with low latency

- Makes content available globally or restricts it based on location
- Speeds up delivery of static (a website were all the files needed to run the website run on the client (i.e. web browser) rather than the server) and dynamic web content
- Uses edge locations to cache content (caches are just copies of files)

If the content being delivered is not already in the edge location, CloudFront How this plays out:

First a user submits a request to view content and that request goes to a CloudFront distribution (which is really just a collection of edge locations called a **Distribution Cache**) and if the content is there it goes immediately back to the user. If the content is NOT in an Edge Location then there's a request to the **Origin**. The Origin is the orignal location of the content e.g.(S3 Bucket, EC2 Instance, or Elastic Load Balancer) and sends it back to the user.
Other benefits of CloudFront:

- CloudFront is often used with S3 to deploy content globally
- CloudFront can stop certain web attacks like DDoS
- Geo-restriction prevents users in certain countries from accessing your content with an IP Address blocker.

Amazon Global Accelerator is a service that improves the availability of your apps

- Improves latency and availability of single-Region applications
- Sends traffic through the AWS global network infrastructure
- 60% performance boost
- Automatically re-routes traffic to healthy available regional endpoints

S3 Transfer Acceleration Imporves content uploads and downloads to and from S3 buckets.

- Fast transfer of files over long distances
- Uses CloudFront's globally distributed edge locations
- Customers around the world can upload to a central bucket

3.7 Understanding Networking Services: VPC and its Subcomponents

Networking connects comuters together and allows for the sharing of data and applications, around the globe, in a secure manner using virtual routers, firewalls, and network management services.

Amazon Virtual Private Cloud (VPC) is a foundational service that allows you to create a secure private network in the AWS cloud hwere you launch your resources.

- Private virtual network
- Launch resources like EC2 instances inside the VPC
- Isolate and protect resources
- Spans multiple AZs in a single region

A VPC is like a fence around your resources, protecting them. In this analogy the VPC is the fence surrounding your resources deployed across several Availablity Zones in a region. A **subnet** allows you to split the network inside a VPC and you launch the resources within the subnet. For example, you can have private resources like a database inside a private subnet. You can also have a public subnet where you put resources that you want to be public (accessible from the internet). There are several components which help make a subnet public:

- **Network Access Control List** - ensures the proper traffic is allowed into the subnet
- **Router and Route Table** - defined where network traffic is directed
- **Internet Gateway** - Allows public traffic from the internet to the VPC

VPC peering allows you to connect 2 VPCs together. **Peering** facilitates the transfer of data in a secure manner.

3.8 Additional Networking Services

What is DNS? Every computer on the internet has an address. The address of a computer is called an IP (Internet Protocol) address. A website/application on the web is on a computer with an IP address. Instead of accessing the website by the IP address, we access them through IP address. A **DNS** stands for Domain Name System and directs internet traffic by connecting domain names with web servers.

Amazon Route 53 is a DNS service that routes users to applications.

- Allows for domain name registration

- Performs health checks on AWS resources
- Supports hybrid cloud architectures

AWS Direct Connect is a dedicated physical network connection from your on-premises data center to AWS.

- Dedicated physical network connection
- Connects your on-premises data to AWS
- Data travels over a private network
- Supports a hybrid deployment model

A use case for Direct Connect is linking private data in the private cloud on site and the application runs on the public cloud (AWS services). Other use cases:

- Large datasets
- Business-critical data
- Hybrid model

AWS Site-to-Site VPN a Site-to-Site VPN creates a secure connection between your internal networks and your AWS VPCs (different fro VPC because data travels over public internet not a private network)

- Similar to Direct Connect but data travels over the public internet
- Data is automatically encrypted
- Connects your on-premises data center to AWS
- Support a hybrid environment
- Cheaper than Direct Connect

Site-to-Site VPN in the Real World is used to make moving applications to the cloud easier. A **Virtual Private Gateway** is a VPN connector on the AWS side that supports the connection ("tunnel") from on-premises to cloud. The customer also has their own connection called the **Customer Gateway**. Between the Virtual Private Gateway and Customer Gateway is where the Site-to-Site VPN facilitates the connection.

3.9 Databases

In the AWS ecosystem, there are many different types of databases that support different use cases. Databases allow us to collect, store, retrieve, and manipulate data. A database is an organized collection of various forms of data. Databases are used by many applications. Databases are typically controlled by a **Database Management Systems (DBMS)**. AWS databases for different use cases:

- Relational - **RDS, Aurora**
- NoSQL - **DynamoDB**
- Graph Databases - **Neptune**
- In-memory - **ElastiCache**
- Document Data Stores - **DocumentDB**

Amazon Relational Database Service (RDS) is a service that makes it easy to launch and manage relational databases.

- Supports popular database engines like Aurora, PostgreSQL, MySQL, Oracle, and SQL Server
- Offers high availability and fault tolerance using Multi-AZ deployment option
- AWS manages the patching, backups, OS maintence
- Launch read replicas across Regions in order to provide enhanced performance and availablility

Aurora is a relational database compatible with MySQL and PostgreSQL that was created by AWS

- 5x faster than normal MySQL and 3x faster than PostgreSQL
- Scales automatically
- Managed by RDS

DynamoDB is a fully managed NoSQL key-value document database.

- Fully managed and serverless
- Non-relational
- Scales automatically to large workloads with fast performance

DocumentDB is a fully managed document database that supports MongoDB

- Fully managed and serverless
- Non-relational

ElastiCache is a fully managed in-memory datastore compatible with Redis or Memcached.

- Data can be lost because its stored in memory

Neptune is a fully managed graph database that supports highly connected datasets (like social media networks)

- Fast and serverless

Case studies:

1. If you want to migrate Oracle to the cloud? RDS
2. If you want to migrate on-premises PostgreSQL database to the cloud? RDS or Aurora
3. Alleviate database load for data that accessed often? ElastiCache
4. Social Media network user data site? Neptune
5. NoSQL database fast enough to handle millions of requests per second? DynamoDB
6. Operate MongoDB workloads at scale? DocumentDB

3.10 Migration and Transfer Services

Comapnies are moving to the cloud and they need services to aid the transfer of on-premises to AWS.

Database Migration Service (DMS) DMS helps you migrate databses to or within AWS.

- Migrate on-premises to AWS
- continuous data replication
- Supports homogenous and heterogeneous migrations
- Virtually no downtime

DMS use cases:

- Oracle to Aurora MySQL
- Oracle on premises to Oracle on EC2
- RDS Oracle to Aurora MySQL

Server Migration Service (SMS) SMS allows you to migrate on-premises servers to AWS

- Server saved as a new Amazon Machine Image (AMI)
- Use AMI to deploy to EC2 instances

The **AWS Snow Family** allow you transfer on premises to the cloud using a physical device. The Snow Family consists of:

- Snowcone - Smallest member of the data transport devices. Holds 8 terabytes of usable storage. Offline shipping. Online with **DataSync**

- Snowball and Snowball Edge - Pentabyte-scale data transport solution. Cheaper than internet transfer. Edge supports EC2 and Lambda
- Snowmobile - Multi-petabyte or exabyte scale data data. Moves your data in a 45 foot long shipping container. Securely transported.

DataSync allows for online data transfer from on-premise to AWS storage serices like S3 or EFS.

- Migrates data from on-premises to AWS
- Copy data over Direct Connect or the internet
- Copy data between AWS storag services
- Replicate data across a Region

3.11 Analytics Services

A **Data Warehouse** is a data storage solution that aggregates massive amounts of historical data from disparate sources. Data Warehouses suppot querying, reporting, analytics, and business intelligence. They are not used for transcation processing.

Redshift is a scalable data warehouse solution. It handles exabyte-scale data. Use cases:

- Data consolidation - when you need to consolidate multiple data sources for reporting
- Relational - When you want to run a database that doesn't require real-time transaction processing (insert, update, delete)

Analytics is the act of querying or processing your data. There are several services that allow you to make insights with these data.

Athena is a query service for Amazon S3 that allows you to use standard SQL.

Glue prepares your data for analytics. It is an **ETL (Extract, Tranform, and Load)** service. Helps you better understand your data. **Kinesis** allows you to analyze data and video streams in real time. Supports video, audio, application logs, website clickstreams, and IoT data. Useful for real time fraud detection. **Elastic MapReduce (EMR)** helps you process large amounts of data. Analyze data using Hadoop which leverages distributed computing. Also works with Apache Spark. **Data Pipeline** helps you move data between compute and storage services running either on AWS or on-premises. You move data based on conditions or at specific intervals. It can send notifications of sucess or failure.

3.12 Machine Learning

Arificial Intelligence (AI) teaches computers to do things that normally require human intelligence.

Rekognition allows you to automate your image and video analysis. Face and text detection in images and video.

Comprehend is a natural language processing (NLP) service that finds relationships in text. It finds insights and relationships within text.

Polly turns text into speech. It mimic natural sounding human speech. Several voices across many languages.

SageMaker helps you build, train, and deploy machine learning models quickly. It allows you to use Deep Learning AMIs. **Translate** provides real-tiem and batch language translation, supporting many languages. **Lex** helps you build conversational interfaces like chatboxes. It can recognize speech and understands language. Lex actually powers Amazon Alexa.

3.13 Developer Tools

Software developers use tools to accelearte the software development and release cycle.

Cloud9 is an integrated development environment (IDE) that supports many programming languages. Cloud9 preconfigures the development environt with the proconfigured SDKs and libraries.

CodeCommit is a source control system for private Git repositories. This can be used to manage source code and the different versions of these files. CodeCommit is like Github.

CodeBuild allows you build and test your application source code. Complies source code and runs tests. It enables continuous integration and delivery. Produces build artificats ready to be deployed.

CodeDeploy manages the deployment of code to compute services in the cloud or on-premise. Deploy code to EC2, Fargate, Lambda, and on-premises. It helps maintain application up-time. Useful for rolling out a new version of your application because it has something called rolling deployments.

CodePipeline automates the software release process. Quickly deliver new features and updates to production. Integrates with CodeBuild to run builds and unit tests. Integrates with CodeCommit to retrieve source code. Integrates with CodeDeploy to deploy your changes. CodePipeline is useful for adding

DevOps automation to the building, testing, and deploying of your application.

X-Ray helps you debug and analyze production applications. View requests end to end. X-Ray is useful to help map requests made to your RDS database from within your application. You can track information about SQL queries and more.

3.14 Deployment and Infrastructure Management Services

These services help you quickly stand up new applications, automate the maangemetn of infrastructure, and provide real-time visiability into the health of your systems.

Infrastructure as Code (IaC) allows you to write a script to provision AWS resources. The benefit is that you provision resources in a reproducible manner that saves time.

CloudFormation allows you to provision AWS resources using Infrastrucutre as Code (IaC). You can create templates with the resources you want to provision. This can be used to automate the creation of EC2 instances in your AWS account. The stack of resources created will depend on the template. **Elastic Beanstalk** is technically a compute service when you're ready to deploy your applications. Its an orchestration service that provisions resources. It automatically handles deployment from capacity provisioning, load balancing, and autoscaling. Monitors application health via a health dashboard. It is useful to quickly deploy a web application to AWS.

OpsWorks allows you to deploy code and manage applications. Manage on-premises servers or EC2 instances in AWS Cloud. Works with automation platforms like Chef and Puppet. This service is useful for automating software configurations and infrastrucutre management for your application.

3.15 Messaging and Integration Services: SQS

Coupling defines the interdependencies or connections between components of a system. Loose compuling helps reduce the risk of cascading failures between componenets. **Monolithic Applications** are large applications with lots of interdependencies, these are described by **Tight coupling**. A monolithic application broken down into microservices that are connected but not depenent on each other is an example of **loose coupling**. **Queues** are used to implement loosely coupled systems. A Queue is a data structure that holds requests or messages (like waiting in line) (FIFO - First in, First out).

Simple Queue Service (SQS) is a message queuing service that allows you to build loosely coupled systems. Allows component-to-component communication using messages. Multiple components (or producers) can add messages to

the queue. Messages are processed in an asynchronous manner. An example: SQS lets you build a money transfer app the performs well under heavy load. The messaging queue will help you send a money transfer request. A request gets added to the queue once it is submitted. Requests are processed in FIFO order.

3.16 Messaging and Integration Services: SNS and SES

Sometimes users of applications need to know when certain events happen. Notifications such as text messages or emails can be sent through services in the cloud. **Simple Notification Service (SNS)** allows you to send emails and text messages from your applications. These messages are formatted as text. SES (next) allows for richly formatted html emails. Example: SNS works with CloudWatch when an EC2 instance's utilization is over 80%.

Simple Email Service (SES) SES is an email service that allows you to send richly formatted HTML emails from your applications. Ideal for professional email or marketing campaigns.

3.17 Auditing, Monitoring, and Logging Services

These services give you insight and visibility into how well your systems are performing and help you proactively find and resolve errors. This can help answer questions like:

- Who signed in and made changes via the AWS management console?
- What is the current load on the EC2 instance?
- What is the root cause of this application error?
- Which execution path resulted in this error?

CloudWatch is a collection of services that helps you monitor and observe your cloud resources. Collects metrics, logs, and events. Detect anomilies in your enviornment. You can also set alarms.

CloudWatch Alarms is a service that helps set high resolution alarms (e.g. a billing alarm to help estimate AWS charges)

CloudWatch Logs allow you to monitor application logs for performance data

CloudWatch Metrics helps you visualize time series data (e.g. CPU usage of an EC2 instance over time)

CloudWatch Events Trigger an event on a condition

CloudTrail tracks user activity and API calls within your account. You can

log and retain account activity. Track activity through the console, SDKs, and CLI. Identify which user made changes. Detect unusal activty your account. You can track the time a particular event occured in your account for any event in the last 90 days. You can track the username, event time and name, IP address, access key used, Region, and Error code.

4 Security and Compliance

4.1 The Shared Responsibility Model

In the public cloud, there is shared security responsibility between you and AWS. AWS is responsible for security OF the cloud. You are responsible for security IN the cloud.

AWS is responsible for the global infrastructure like Regions, edge locations, and Availability Zones. In addition, AWS is responsible for the building security which house the data centers. AWS maintains networking components: generators, air conditioning units, fire suppression, and more. AWS is responsible for any managed service like RDS, S3, ECS, or Lambda, patching of host operating systems and data access endpoints.

You are responsible for how the services are implement and managing your application data. In addition, you are responsible for security configurations, patching (updates and security patches) on guest EC2 instances. You are responsible for application security and identity and access management. You are responsible for network traffic protection and group firewall configuration. Lastly, you are responsible for your application code and installed software.

The EC2 shared responsibility model. You are responsible for: installed applications, patching the guest operating system, security controls. AWS is responsible for: EC2 service, patching the host operating system, and security of the host server.

The Lambda shared responsibility model. You are responsible for: Security of the code, storage of sensitive data, and IAM for permissions. AWS is responsible for: Lambda Service, Lambda Lanugage, Lambda endpoints, Operating system, underlying infrastructre, and software dependencies.

Which security responsibilities are shared?

- Patch Management - AWS patches infrastructure, You patch guest OS and applications
- Configuration Management - AWS configures infrastructure devices, You configure databases and applications

How do I report abuse of AWS resources? Contact the AWS Trust & Safety team using the Report Amazon AWS abuse form

4.2 The Well-Architected Framework

The 5 pillars of the Well-Architected Framework describe design principles and best practices for running workloads in the cloud.

1. Operational Excellence. Creating applications that effectively support production workloads. Plan for and anticape failure, Deploy smaller, reversible changes. Script operations as code. Learn from failure and refine.
2. Security. Putting mechanisms in place that support the systems and data. Automate security tasks. Encrypt data in transit and at rest. Assign only the least priviledges required. Track who did what and when. Ensure secuirty at all application layers.
3. Reliability. Designing systems that work consistently and recover quickly. Recover from failure automatically. Scale horizontally for resilience. Reduce idle resources. Manage change through automation. Test recovery procedures.
4. Performance Efficiency. The effective use of computing resources to meet system and business requirements while removing bottlenecks. Use serverless architectures first. Use multi-region deployments. Delegate tasks to a cloud a vendor. Experiment with virtual resources.
5. Cost Optimization. Delivering optimum and resilient solutions at the least cost to the user. Utilize consumption-based pricing. Implement Cloud Financial Management. Meaure overall efficiency. Pay only for resources your applications require.

Corresponding services to support the 5 pillars

- Operational Excellence. You can use AWS CodeCommit for version controlling of source code and CloudFormation templates
- Security. You can configure central logging of all actions performed in your account using CloudTrail
- Reliability. You can use Multi-AZ deployments for enhanced availability and reliability of your RDS Databases
- Performance Efficiency. Lambda can run your code with zero administration

- Cost Optimization. You can use S3 Intelligent-Tiering to move our data between access tiers based on your usage patterns

4.3 IAM Users

Identity and Access Management (IAM) allows you to control access to your AWS services and resources. Helps you secure your cloud resources. You define who has access and what they can do. Free global service.

Identities define who can access your resources. Examples are Root User, individual users, groups, and roles.

Access defines what resources the users can access. This is controlled through policies, AWS managed policies, customer managed policies, and permission boundaries.

Authentication (Who) is where you present your identity and provide verification (username and passwork)

Authorization (What) determines which services and resources the authenticated identity has access to.

Users are the entities you create in IAM to represent the person or application needing to access your AWS resources. There are some things only the Root User can do: close account, change email address, and modify your support plan. Individual users are created in IAM and are used for everday tasks like starting up an EC2 instances. Pro tip: applications can be users.

Principle of least privilege invovles giving a user the minimum access required to get their job done.

In the real world, you need to create access keys for an IAM user that needs access to the AWS CLI. Those keys can be generated using IAM. Private key accesses the SSH client and the public key accesses the instance.

A **group** is a collection of IAM users that helps you apply common access controls to all group members (e.g. Administrators, Developers, and Analysts). Side Note: Don't confuse security groups in EC2 (which are firewalls) with IAM groups (collections of users). Access to the groups is assigned using policies and roles. In the real world, groups are used to apply the same access controls for a large set of users.

4.4 IAM Permissions

Roles define access permissions and are temporarily assumed by an IAM user or service (e.g. a DevOps-Engineer Role or Lambda-Execution Role). Think of

a role as wearing a hat where you assume the permissions given to that role.

- You assuem arole to perform a task in a single session
- Assumed by any user or service (e.g. you can let Lambda access a RDS database or S3) that needs it
- Access is assigned using policies
- You grant users in one AWS account access to resources in another AWS account

In the real world you can attach a role to an EC2 instance for access to S3. Roles allow you to avoid long term credential sharing (access keys) and prevents your instances from unauthorized access.

You manage permissions for IAM users, groups, and roles by creating a **policy** document in JSON format and attaching it.

A common use case is attaching the AWS managed (created) policy which gives access to a specific group (typically of developers) for Cloud9 and EC2.

Another use case is attaching a S3FullAccess Managed Policy to a role, that way whenever a user takes on that role they have full access to S3 until that role expires.

You can limit access to an Amazon S3 bucket to specific users by using a bucket access policy which can be directly attached to the S3 bucket.

IAM best practices:

- Enable MFA for priviledged users
- Implement strong password policies
- Create individual users instead of using root
- Use roles for Amazon EC2 instances

The **credential report** gives you a list of all users and the status of their various credentials.

4.5 Application Security Services

AWS has several software-based security tools available to help you monitor and protect your resources.

Firewalls prevent authorized access to your networks by inspecting incoming

and outgoing traffic against security rules you've defined. **Web Application Firewall (WAF)** helps protect your web applications against common web attacks. Protects against SQL injection, cross-site scripting.

Distributed Denial of Service (DDoS) A DDoS attacka causes a traffic jam on a website or application to crash it. A hacker can send many Bots which make many requests to your web application. **Shield** is a managed DDoS protection service. Shield Standard provides free protection against common and frequently occuring attacks. Shield Advanced provides access to AWS experts 24/7 for a fee. Shield works with CloudFront, Route53, Elastic Load Balancing, and AWS Global Accelerator.

Macie helps you discover and protect sensitive data. It uses machine learning and evaluates your S3 environment to uncover personally identifiable information.

4.6 Additional Security Services

Config allows you access, audit, and evaluate the configurations of your resources. It delivers the configuration history file to S3 and notifies via SNS of every configuration change. This can be used to identfy system-level configuration changes made to your EC2 changes.

GuardDuty is an intelligent threat detection system that uncovers unathorized behavior. It uses machine learning and has built in detection features for EC2, S3, and IAM. It reviews ClouTrail, VPC flow logs, and DNS logs. This can be used to detect unsual API calls in your account. This is anomaly detection for common techniques used by hackers. **Inspector** works with EC2 instances to uncover and report vulnerabilites. The agent is installed on an EC2 instance, and reports vulnerabilities found. This can be used to identitfy unintened network access to an EC2 instance.

Artifact offers on-demand access to AWS security and compliance reports. It is a central repository for compliance reports from third-party auditors. This can be used to access AWS' certification for ISO compliance because Artifact is a central repo for all compliance reports.

4.7 Data Encryption and Secrets Management Services

Data encryption is important because it is a way to secure your data. It encodes data so it cannot be read by unauthorized users.

Data in flight are data that are moving from one location to another. **Data at rest** are data that is inactive or stored for later use.

Key Management Service (KMS) allows you to generate and store encryption keys. It can generate keys and it can store and control keys. These keys are to encrypt and decrypt data. You can create an encrypted EBS volume by specifying a KBS customer master key.

CloudHSM is a hardware security module HSM) used to generate encryption keys. It is dedicated hardware for security but AWS won't have access to the keys. This is useful for meeting compliance requirements for data security by using dedicated hardware in the cloud.

Secrets Manager allows you to manage and retrieve secrets (passwords or keys). You can encrpyt secrets at rest or rotate, manage, and retireve secrets. It integrates with RDS, Redshift and DocumentDB. If you need to retireve database credentials needed for you application code, that way credentials aren't hard coded into application code.

5 Pricing, Billing, and Governance

5.1 Pricing

There are 3 fundamental drivers of cost: compute, storage, outbound data transfer (inbound data transfer generally is free). There's a detailed whitepaper on how pricing works.

There are 3 different types of free offers available: 12 months free, always free, trials

There's 5 ways to pay for EC2 instances. On-Demand, Savings Plan, Reserved Instances, Spot Instances, and Dedicated Hosts

There's 3 factors that play into how Lambda is charged: number of requests, code execution time, first 1,000,000 requets are always free

With S3 you way for the storage you use: Storage class, Storage, Data transfer (Data transferred out of S3 Region (within region is free)), and Request and data retrieval

There's several features that drive pricing: running clock hours, type of database (size, engine, memory class, etc.), storage, purchase type (on-demand or reserved), database count, API requests, deployment type (single AZ or multiple AZ), and data transfer.

Total Cost of Ownership is a financial estimate that helps you understand both the direct and indirect costs of AWS.

Application Discovery Service helps you plan migration projects to the AWS Cloud. Used to estimate TCO and works with other services to migrate services.

A few ways to reduce TCO: Minimize capital expenditures, utilize reserved instances, and right sized resources

Pricing Calculator provides an estimate of AWS fees and charges.

5.2 Billing Services

AWS provides tools to track your spend.

Budgets allow you set custom budgets that alert you when your costs or usage excced defined limits. There's 3 budget types: cost budgets, usage budgets, and reservation budgets. AWS budgets are useful for monitoring free tier usage so you don't generate usage beyond them.

Cost and Usage Report contains the most comprehensive set of cost and usage data up to the instance ID. This is useful for viewing granular data about your AWS bill **Cost Explorer** Allows you to analyze your past costs and forecast future costs.

5.3 Governance Services

Governance and management services help you maintain control over cost, compliance, and security across all your AWS accounts.

Organizations allows you to centrally manage multiple AWS accounts under one umbrella. Group multiple accounts, with a single payer for all accounts. An Organazation is an entity that allows you to consolidate multiple AWS accounts with a single master payer through 'consolidated billing'. Organizaton units are similar AWS accounts that are grouped together.

Benefits of using organizations: consolidated billing, cost savings (volume discounts), and account governance (a quick and automated way to create and invite AWS accounts to your organization).

Control Tower helps you ensure your accounts conform to company-wide policies. This helps set up new accounts using multi-account strategy, works with AWS orgnizations by automatically setting up organizations as the underlying AWS service for all your accounts. It also provides a dashboard to manage your accounts. This service is useful when you want to disallow public write access to all your S3 buckets across all your accounts.

Systems Manager gives you visibility and control over your AWS resources like automating operational tasks on your resources and take actions on group resources. You can also patch and run commands on multiple EC2 instances or manage several RDS instances. This is useful if you want to deploy operating system patches to all your instances.

Trusted Advisor provides real-time guidance to help you provision your resources following AWS best practices. It checks your account and makes reccomendations and it helps you see service limits and understand best practice. Often time Trusted Advisor checks for unrestricted access on EC2 ports. It checks for public access into S3 buckets. It checks for IAM password policy. It checks for service usage greater than 80% of service limit. It checks for exposed access keys. It checks for CloudFront delivery optimization. This service can be useful for checking read and write capacity for DynamoDB.

5.4 Management Services

There are several services that help you migrate to and build faster in the cloud.

Managed Services helps you efficeintly operate your AWS infrastructure. Augments your internal staff, provides ongoing management of your infrastructure, and reduces overhead. This service can be useful if you want to develop application specific health monitoring using CloudWatch, then Managed Services will help you do this.

Professional Services helps enterprise customers move to a cloud-based operating model. They propose solutions, architect solutions, and help implement solutions. If you need help evaluating an application for migration to a cloud then you can levearge Professional Services.

AWS Partner Network (APN) is a global community of approved partners that offer software solutions and consulting solutions for AWS. Offers technology partner that provide software solutions and provides consultin partners that offer professional services. An APN could help you get a new application up and running quickly.

Marketplace is a digital catalog of prebuilt solutions you can purchase or license. You may also sell your own solutions to others via Markeplace. This is useful for testing software before you buy software you need in Marketplace because many sellers allow free trials.

5.5 Support Plans

There are 4 support plans on AWS:

1. Basic support is included for free for all AWS accounts but gives access to discussion forums and 24/7 support.

2. Developer starts at $29 a month and is reccomended for testing and development. You have access to techincal support and you are allowed a cloud support associate during business hours only.

3. Business starts at $100 a month and is recommened for production workloads. You have access to technical support and you have unlimited contacts and can access a full set of Trusted Advisor checks. 24/7 phone, email, or chat support with Cloud Support Engineers.

4. Enterprise starts at $15,000 a month is reccomended for mission-critical production workloads. You have access to technical support. Unlimited contacts. A TAM (Technical Account Manager), Concierge Support Team (for billing and account questions), Infrastructure Events, 24/7 contact with Cloud Support Engineers. Less than 15 minute response times for business-critical system down.

There are 3 types of support cases you can open with AWS Support (AWS Support doesn't allow cases for code development or system administration tasks):

1. Account and Billing related cases

2. Service limit increases (some services have quotas/limits and you need to request to go beyond them)

3. Techincal Support cases can only be opened by customers on developer, business, or enterpise plans.