AWS Certified Cloud Practioner Exam Notes

Matt Waismann

July 29, 2021

1 Introduction

1.1 Exam Blueprint

The exam validates ability to:

- Value of AWS Cloud
- Understand and expalin the AWS shared responsility 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 exist 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 opn a single physical server into smaller untis 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 the 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 resource 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 AWS (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