

AWS Developer Associate Notes

Matt Waismann

August 24th, 2021

1 Introduction

There's five domains for the exam:

1. Deployment - 22%
2. Security - 26%
3. Development with AWS Services - 30%
4. Refactoring - 10%
5. Monitoring and Troubleshooting - 12%

A minimum passing score is 720/1000. The exam has 65 questions and is 130 minutes in length. The questions will be multiple choice.

2 IAM Identity Access Management

- Manage users and their access to the AWS console
- Allows for Multi-Factor Authentication
- Can provide temporary access to aws services
- Supports PCI DSS requirement (payments industry)
- Allows for users, groups (which you can add users too), and roles (you create roles and assign them to aws services or users)
- IAM policy is a document that defines one or more permissions for a user, group, or role
- IAM Policy Simulator allows you to test the effects of IAM policies before committing them to production or for troubleshooting existing policies.
- IAM is universal, not regional
- New users have no permissions when they are first created

- New users are assigned an access key ID and secret access key when their accounts are created. You can use access keys to access AWS via the APIs and command line.
- You only get to view the access key ID and secret access key once. If you lose them, you must regenerate them.
- You can create and customize your own password rotation policies for your users.
- AWS does not recommend that EC2 instances have credential stored on them

3 Elastic Compute Cloud (EC2)

- Resizable compute capacity in the Cloud. Like a VM hosted on AWS. Designed to make web-scale cloud computing easier for developers.
- AWS was the first company to introduce a public cloud compute service like EC2
- Pay only for what you use
- No wasted capacity, you can grow or shrink as you need

There are different pricing options for EC2

- On Demand - by the hours or the second, depending on the instance you run
- Reserved Instance - for one or three years, up to 72% discount. But the reservation only applies to a region so other instances in other regions won't be covered. Three types:
 - Standard RIs: Up to 72% off. But cannot change instance size (e.g. t3 large to t3 extra large)
 - Convertible RIs: Up to 54% off on-demand price. Has the option to change to a different RI type of equal or greater value (allows for increasing instance size)
 - Scheduled RIs: Launch within the time window you define (e.g. A process happens once a month on a given date)
- Spot - Purchase unused capacity at a discount of 90%. Prices fluctuate according to supply and demand
- Dedicated - A physical EC2 server dedicated for your use. The most expensive option. Useful for regulatory requirements that may not support multi-tenant virtualization.

Savings Plans

- Save up to 72% by committing a specific amount of compute power measured in \$/per hour for a one year or three year period
- Not just for EC2 but also other serverless technologies like Lambda and Fargate.

The pricing calculator is a way see how the different EC2 pricing options will affect pricing.

3.1 EC2 Instance Types

- Hardware - The instance types determines the hardware of the host computer used for your instance.
- Capabilities - Each instance type offers different compute, memory, and storage capabilities and are grouped in instance families.
- Application Requirements - Select an instance type based on the requirements of your application
- We do not need to memorize the different instance types for this exam. There's several instance type families
 - Micro - low cost, often free tier eligible
 - General Purpose
 - Compute Optimized - a higher ratio of vCPU to memory. This is useful for batch processing, analytics, and high performance science and engineering purposes
 - FPGA - Field programmable data arrays
 - GPU instances - Great for graphics processing or parallel processing
 - Machine learning ASIC instances - uses the inferentia chip
 - Memory Optimized - Memory optimized with the lowest cost per GB of random
 - Storage Optimized - Provides direct storage options attached. Helpful for databases

The instance type names read like: 't2.nano, t2.xlarge, p2.8xlarge, etc.'

3.2 Elastic Block Store

EBS volumes are storage volumes that you can attach to your EC2 instances. Think of these EBS volumes as the disks attached to your PC. Every EC2 instance has an EBS volume attached to it for the operating system, often Amazon Linux 2. You can add more volumes depending on the needs of your application.

- Designed for mission critical workloads
- These can dynamically scale in real time without any downtime for your service

EBS Types

- General Purpose SSD (gp2) - a balance of price and performance. 3 IOPS per GiB, up to max of 16,000 IOPS per volume. Good for development and testing or applications that aren't latency sensitive.
- Provisioned IOPS SSD (io1) - The high performance option. Up to 64,000 IOPS per volume. 50 IOPS per GiB. Use if you need more than 16,000 IOPS.
- Provisioned IOPS SSD (io2) - Latest generation. Higher durability and more IOPS. This is the same price as io1.
- Throughput Optimized HDD (st1) - Low-cost HDD volume. Baseline throughput of 40 MB/s per TB. Frequently-accessed, throughput-intensive workloads. Great for Big Data, data warehouses, ETL, and log processing. A cost effective way to store mountains of data. Max throughput is 500 MB/s per volume.
- Cold HDD (sc1) - Lowest cost option. Baseline throughput of 12 MB/s per TB. Ability to burst up to 80 MB/s per TB. Max throughput of 250 MB/s per volume. A good option if performance is not a factor. Max throughput of 250 MB/s per volume

IOPS vs. Throughput

- IOPS - measures the number of read and write operations per second. Important metric for low latency apps. Choose Provisioned IOPS SSD (io1 or io2)
- Throughput - measures the number of bits read or written per second (MB/s). Important metric for large datasets, large I/O sizes, complex queries.

Few extra tips

- If you create a EBS volume from an encrypted snapshot, then you will get an encrypted volume
- If you create an EBS volume from an unencrypted snapshot, then you will get an unencrypted volume

4 Serverless Computing

4.1 Introduction

Serverless allows you to run applications in the cloud without having to worry about managing servers. This typical server management tasks are capacity provisioning, patching, auto scaling, and high availability. Also, serverless applications scale easily. Serverless applications are also low cost because they are event driven, meaning you are only charged when your code is executed. Some examples of Serverless AWS services are Lambda, SQS (Simple Queue Service), SNS (Simple Notification Service), API Gateway, DynamoDB, and S3.

4.2 Lambda

Lambda is a serverless compute service. Lambda takes care of everything to run your code, including the runtime environment. It supports common languages like Java, C#, Python, and Ruby. You are charged based on the number of requests, their duration, and the amount of memory used by your Lambda function. Lambda is event-driven, meaning Lambda functions can be automatically triggered by other AWS services or called directly from any web or mobile app. These events could be changes made to data in S3 or DynamoDB table. Other triggers include DynamoDB, Kinesis, SQS, Application Load Balancer, API Gateway, Alexa, CloudFront, S3, SNS, SES, CloudFormation, CloudWatch, CodeCommit, CodePipeline, and many more. You can use API Gateway to configure an HTTP endpoint, allowing you to trigger your function at any time using an HTTP request. Lambda functions are independent, meaning each event will trigger a single function.

4.3 API Gateway

API stands for Application Programming Interface. We use APIs to interact with web applications, and applications use APIs to communicate with each other. Generally, an API sends a response in JSON format. API Gateway is a service that allows you to publish, maintain, and monitor APIs. API Gateway supports two types of APIs

- RESTful APIs - Stateless, serverless workloads
- Websocket APIs - real-time, two-way, stateful communication (e.g. chat apps)

API Gateway provides a single endpoint for all client traffic interacting with the backend of your application. So if a user makes a request to our AWS environment then API Gateway directs that request to the appropriate services. More specifically what is API Gateway?

1. It allows you to connect to applications running on Lambda, EC2, or Elastic Beanstalk and services like DynamoDB and Kinesis

2. Supports multiple endpoints and targets
3. Support multiple versions of your API. This allows for different versions for development, testing, and production environments.

In addition it is serverless, integrates with cloud watch to log API calls, latencies and error rates, and it supports throttling to manage traffic spikes and DDoS attacks.

4.4 Lambda Versions

When you create a new Lambda function there's only one version: `$LATEST`. When you upload a new version of the code to Lambda, this version becomes `$LATEST`. You can create multiple versions of your function code and use aliases to reference the version you want to use as part of the ARN. In a development environment you might want to maintain a few versions of the same function as you develop and test your code. An alias is like a pointer to a specific version of the function code. Use Lambda versioning and aliases to point your applications to a specific version if you don't want to use `$LATEST`.

4.5 Lambda Concurrent Executions Limit

There is a limit to the number of concurrent executions in Lambda. AWS Support can help increase that. Reserved concurrency guarantees a certain number of concurrent requests are allowed for a given Lambda function.

4.6 Lambda and VPCs

If a Lambda needs to access/communicate a service within a subnet in a private VPC. To enable Lambda to Access VPC Resources:

- You need to allow the function to connect to the private subnet
- Lambda needs the following VPC Configuration information so it can access the VPC: private subnet ID and security group ID. Lambda uses this information to set up ENIs (Elastic Network Interface) using an available IP address from your private subnet. All of this can be added inside the CLI. In the console, a network can be added when creating the Lambda function very easily
- Doing this will allow your function to access resources in VPC.