AWS

Overview

- What is AWS?
- AWS Console
- IAM
- AWS CLI
- EC2
- S3
- Lambda

Learning Objectives

- Define the role AWS plays in modern software development
- Identify the different use cases for the console and CLI
- Implement services such as IAM, EC2, S3 and Lambda

What is AWS?

Amazon **W**eb **S**ervices is a cloud computing platform.

Offerings encompass computing power, database storage, content delivery, logging and monitoring - if you need to do a thing, there's an AWS product for it.

At last count, there were over 175 AWS products to choose from...

Regions

- A physical location somewhere in the world where AWS data centers are clustered
- Each group of logical data centres is called an **Availability Zone**
- Multiple geographic Regions, including North America, South America, Europe, China, Asia Pacific, South Africa, and the Middle East
- Regions have a code name, such as `eu-west-1` which represents the Irish region

---

Availability Zones

- One (or more) discrete data center(s) in an AWS region
- AZs in a region are physically-separate, but within 100km of each other - high-bandwidth, low-latency networking
- Gives customers the ability to operate production applications and databases that are more highly available, fault tolerant, and scalable than would be possible from a single data center
- If an application is partitioned across AZ's, companies are better isolated and protected from issues such as power outages, lightning strikes, tornadoes, earthquakes, and more

Data centres are just enormous buildings that operate a vast amount of computer machinery, with its own cooling and power setup.

Currently 24 AWS regions, containing 77 AZs.

This is a useful tool to visualise it: https://aws.amazon.com/about-aws/global-infrastructure

---

Exercise

Follow the *AWS Account Setup* steps in the `aws-exercise.pdf` handout.

IAM

IAM

- **I**dentity and **A**ccess **M**anagement
- Manage users and their level of access to the CLI or console
- Manage roles permissions for the roles
- Manage authentication for users accessing AWS
- Free to use - you can create as many roles as you wish

IAM Features

- Granular permission - user can access service X but not service Y
- Identity Federation (login with Facebook, Google, Active Directory etc.)
- MFA
- Password rotation policy
- Integrates with many different AWS services

IAM Key Terms

- Users
- Groups
- Roles
- Policies

We will dive into what each means.

IAM - Users

- End users such as people, employees etc.

- Accounts with username and password
- Can define level of access to AWS services
- Manage the permissions of what the user can perform
- Manage their security credentials (MFA etc.)
- You are either the account owner (root) or an IAM user.

Example: Each learner is an IAM user.

IAM - Groups

- A collection of users, where you can define permissions for all of them in an easier way
- A group can contain many users, and a user can belong to multiple groups
- Groups can't be nested; they can contain only users, not other groups
- There's no default group that automatically includes all users in the AWS account

Example: All learners will be in a group called Learners.

IAM - Roles

- Similar to an IAM user, except a role is intended to be assumed by anyone or any service that needs it
- Provides temporary security credentials for the length of the session, as opposed to a username and password
- Specific permissions on AWS services and resources
- Policies are attached to roles to grant them access/privilege

Example: Each learner assumes a role with their IAM user that gives them wider access to AWS services as opposed to the user itself.

IAM - Policies

- You manage access in AWS by creating policies and attaching them to IAM identities (users, groups, roles) or AWS resources
- A policy is an object that, when associated with an identity/resource, defines their permissions
- These permissions determine if a request is allowed or denied
- Most policies are stored as JSON

Example: The role/group the learners are in have policies associated to give them a certain amount of access to services.

IAM - Best Practices

- Create **individual** users
- Manage permissions with groups
- Use IAM roles for as many actions
- Grant **least privilege** with permissions
- Configure a **strong** password policy
- Enable MFA for privileged users

IAM - Best Practices

- Setup audits with AWS CloudTrail.
- CloudTrail logs for exactly who did what, when, and from where
- Use IAM roles to allow users and services to share access to another service
- Rotate security credentials **regularly**
- Restrict privileged access further with conditions (for instance, only allowing a range of IPs that a request must come from)
- Reduce use of root (mostly used for billing and locking down account securely)

An example of a 'condition' you could impose would be, for example, allowing a user to use a certain service but only Mon - Fri.

Demo the IAM dashboard.

AWS CLI

AWS CLI

- The way we interact with AWS services through the CLI
- Ease of use over logging in to the console
- If you can do it on the Console, you can do it in the CLI - YAY!
- Simple use-cases: searching logs, quick S3 upload/download

CLI Installation

Follow the *AWS CLI Setup* steps in the `aws-exercise.pdf` handout.

```
$ aws < command > < subcommand > [options and parameters]
```

Quiz Time! 🤓

What is an AWS Region?

1. An AWS Infrastructure offering that's optimised for mobile edge computing applications.
2. A physical location somewhere in the world where AWS data centers are clustered.
3. A type of AWS infrastructure deployment that places AWS compute, storage, database, and other select services close to large population, industry, and IT centers.
4. One (or more) discrete data center(s) in an AWS region.

Answer: 2

What are the four main areas of AWS IAM?

1. Groups, Permissions, Roles, Users
2. Groups, Policies. Roles, People
3. Pools, Policies, Roles, Users
4. Groups, Policies, Roles, Users
5. Groups, Policies, Requirements, Users

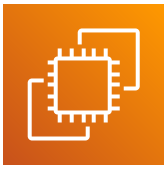
Answer: 4

What are policies used for in AWS IAM?

1. An object that, when associated with an identity/resource, defines their permissions.
2. An object that provides temporary security credentials for the length of the session, as opposed to a username and password.
3. A document that is intended to be assumed by anyone or any service that needs it.
4. A document that defines a user permissions for one specific AWS service.

Answer: 1

EC2



EC2 (Elastic Compute Cloud)

- Service that allows you to rent virtual computers on which you can run your own applications
- 'Elastic' because you pay by the second for what you use!
- You get control over the geographical location of your virtual computers

Before cloud computing, you'd need to put in a request for physical hardware which could take weeks to provision, now it takes seconds, with a few clicks.

EC2 Pricing Types

On Demand:

Allows you to pay a fixed rate by the hour/minute/second with no commitment.

Reserved:

Provides you with a capacity reservation and a significant discount on the hourly charge of an instance. Locked into contract terms of 1 or 3 years.

EC2 Pricing Types

Spot:

Enables you to bid whatever price you want for instance capacity, making better savings if your applications have flexible start/end times.

Dedicated Hosts:

Physical EC2 server dedicated for your own use.

EC2 - Concepts

Image: what is being used to build an instance (similar to Docker)

Instance: the machine you're creating

Security: security groups, key management, network interfaces

Image - essentially a sort of template that contains the software configuration required to launch your instance.

Security Group - a virtual firewall for your EC2 instances to control incoming & outgoing traffic.

Key management: You use key pairs to connect to your EC2 instances (public key is stored in .ssh directory of instance).

Network interface: Configuring stuff like port numbers and network access.

EC2 & EBS

- Elastic Block store - high performance, highly available storage for EC2
- Block-level (organised/identified in blocks) storage that can be attached to EC2 instances
- 2 options available: SSD or HDD

Block storage breaks up data into blocks and then stores those blocks as separate pieces, each with a unique identifier.

The SAN places those blocks of data wherever it is most efficient. That means it can store those blocks across different systems and each block can be configured (or partitioned) to work with different operating systems.

SSD - Solid State Drive
HDD - Hard Disk Drive

Exercise

Follow the *AWS EC2* steps in the `aws-exercise.pdf` handout.

Quiz Time! 🤓

You have created an instance in EC2, and you want to connect to it. What should you do to log in to the system for the first time?

1. Use the username/password combination you created within the EC2 setup.
2. Use the key-pair combination you created within the EC2 setup.
3. Generate a secure login from your AWS Secret Access Key.
4. Log in with your AWS username/password/MFA.

Answer: `2`

True or False: You can use the AWS Console to add a role to an EC2 instance after that instance has been created and powered up.

Answer: `True`

True or False: When creating a new security group, all inbound traffic is allowed by default.

Answer: `False`

S3

S3 - Simple Storage Service

- Secure, durable, highly scalable object store
- Safe place to store files
- **Object**-based storage
- Files can be 0 bytes to 5TB
- **Unlimited** storage
- Files are stored in **buckets** (basically a folder)
- Globally distributed

S3 - Objects

S3 is object-based. Think of objects just like files. They consist of the following:

Key: The name of the object

Value: The sequence of bytes containing the data

Version ID: For versioning

Metadata: Data about data you're storing

S3 Guarantee Model

- Up to 99.99% availability
- Up to 99.999999999% durability (11x 9s)

99.99% availability equates to 52.60 minutes of downtime per year.

99.999999999% durability means that if you store 10 million objects then you expect to lose an object of your data every 10,000 years.

S3 - Advanced Features

- Object versioning
- Storage class: trade durability/availability for cost
- Lifecycle policies: manage the lifetime of your files automatically
- Encryption at-rest
- MFA Delete
- Bucket policies to control who can access them

Exercise

Follow the *AWS S3* steps in the `aws-exercise.pdf` handout.

Lambda

Lambda

- 100% code, 0% infrastructure
- Run code without worrying about OS, patching, scaling, any physical hardware
- Never worry about capacity again
- Lambdas run in response to events such as data changes in S3, DB record being inserted
- You can even call them from through HTTP requests, SDK, or the AWS CLI

Lambda Triggers

A lambda function is automatically invoked when one of its triggers is activated.

For example:

- When a record has been inserted into a DB table
- When a file has been uploaded to S3
- When a commit is pushed onto a repo hosted in CodeCommit (Git for AWS)
- When a monitoring alarm goes off

Lambda Pricing Model

Number of requests: First 1 million requests are free, $0.20 per 1 million after (cheap!)

Duration: Calculated from the time your code begins until it terminates, up to the millisecond. The price depends on how much memory you allocate. Roughly $0.00001667 for every GB-second used. The first 400,000 are free.

It used to be rounded to the nearest ms but is now at a per ms basis.

Limitations

- Cold starts: Time it takes to kick off an instance (it's a container under the hood)
- Difficult to scale without understanding the concurrency execution model
- Tightly integrated to work with other AWS services so may have potential 'lock-in'
- Can be difficult to develop locally
- Unsuitable for tasks that take 15+ minutes

Use Cases

- Tasks that take less than 15 minutes to complete
- Asynchronous, event-driven workloads
- Consistent level of traffic

Exercise

Follow the *AWS Lambda* steps in the `aws-exercise.pdf` handout.

Quiz Time! 🤓

Amazon S3 is which type of storage service?

1. Object
2. Block
3. Network
4. SAN (Storage Area Network)

Answer: 1

Amazon S3 offers developers which combination?

1. Low scalability and high latency data storage infrastructure at high costs.
2. High scalability and low latency data storage infrastructure at high costs.
3. Low scalability and high latency data storage infrastructure at low costs
4. High scalability and low latency data storage infrastructure at low costs.

Answer: 4

How can you trigger a Lambda?

1. Manually in the console.
2. From a trigger, such as S3.
3. A HTTP request.
4. All of the above.

Answer: 4

How many minutes can a Lambda run for?

1. 5
2. 10
3. 15
4. 20

Answer: 3

Clean Up

Make sure to delete the following once you are done:

- Lambdas
- EC2 instances
- S3 buckets

Learning Objectives Revisited

- Define the role AWS plays in modern software development
- Identify the different use cases for the console and CLI
- Implement services such as IAM, EC2, S3 and Lambda

Terms and Definitions Recap

Cloud Computing: The on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user.

Data Centre: A building, dedicated space within a building, or a group of buildings used to house computer systems and associated components, such as telecommunications and storage systems.

Region: A physical location somewhere in the world where data centers are clustered.

Availability Zone: One (or more) discrete data center(s) in a region.

Terms and Definitions Recap

IAM: Defining and managing the roles and access privileges of individual users and the circumstances in which users are granted (or denied) those privileges.

EC2: A web service that provides secure, resizable compute capacity in the cloud.

EBS: An easy to use, high-performance, block-storage service designed for use with Amazon EC2 for both throughput and transaction intensive workloads at any scale.

S3: An object storage service that offers industry-leading scalability, data availability, security, and performance.

Lamba: A serverless compute service that lets you run code without provisioning or managing servers.

Further Reading

- AWS IAM Introduction
- AWS Docs