AWS Certified Developer

Agenda

- AWS Developer Associate Certification
  - Why
  - What to expect
  - How
  - Important AWS services that we will cover
- AWS History
- AWS Global Infrastructure
- Tour of the AWS Console
- AWS Identity and Access Management (AWS IAM)

Who am I?

- Mobeen Ahmed
  - CTO at Codup | VP Engineering at Dastgyr | Architect at Folio3
  - Over all 20 years of experience in the field of software development
  - Taught in different universities as visiting faculty
  - AWS community builder for the last 2 years
  - Speaker
  - Father, Son, and Husband
  - Mentor for young entrepreneurs

Why should you do this course?

- Role versatility
- Makes you a better developer
- Role shift
- AWS is the market leader

What will you learn?

Domain 1: Development with AWS Services

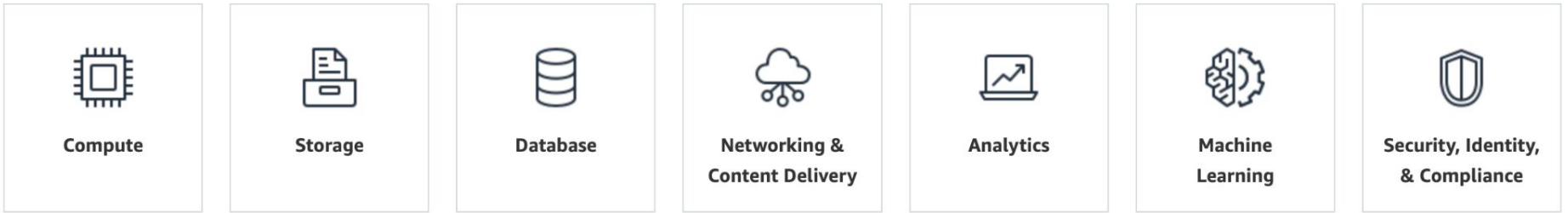
Domain 2: Security

Domain 3: Deployment

Domain 4: Troubleshooting and Optimization

AWS Services

- Over 200+ services
- We will not cover them all
- Focus will be on key services



AWS services on which we'll focus - I

Compute:

Amazon EC2

AWS Elastic Beanstalk

AWS Lambda

AWS Serverless Application Model (AWS SAM)

Containers:

AWS Copilot

Amazon Elastic Container Registry (Amazon ECR)

Amazon Elastic Container Service (Amazon ECS)

Amazon Elastic Kubernetes Service (Amazon EKS)

AWS services on which we'll focus - II

Database:

Amazon Aurora

Amazon DynamoDB

Amazon ElastiCache

Amazon MemoryDB for Redis

Amazon RDS

AWS services on which we'll focus - III

Developer Tools:

AWS Amplify

AWS CloudShell

AWS CodeArtifact

AWS CodeBuild

AWS CodeCommit

AWS CodeDeploy

Amazon CodeGuru

AWS CodePipeline

AWS CodeStar

AWS X-Ray

AWS services on which we'll focus - IV

Management and Governance:

AWS AppConfig

AWS CLI

AWS Cloud Development Kit (AWS CDK)

AWS CloudFormation

AWS CloudTrail

Amazon CloudWatch

Amazon CloudWatch Logs

AWS Systems Manager

AWS services on which we'll focus - V

Networking and Content Delivery:

Amazon API Gateway

Amazon CloudFront

Elastic Load Balancing (ELB)

Amazon Route 53

Amazon VPC

AWS services on which we'll focus - VI

Security, Identity, and Compliance:

AWS Certificate Manager (ACM)

Amazon Cognito

AWS Identity and Access Management (IAM)

AWS Key Management Service (AWS KMS)

AWS Private Certificate Authority

AWS Secrets Manager

AWS Security Token Service (AWS STS)

AWS WAF

AWS services on which we'll focus - VII

Storage:

Amazon Elastic Block Store (Amazon EBS)

Amazon Elastic File System (Amazon EFS)

Amazon S3

AWS services on which we'll focus - IX

Application Integration:

AWS AppSync

Amazon EventBridge

Amazon Simple Notification Service (Amazon SNS)

Amazon Simple Queue Service (Amazon SQS)

AWS Step Functions

AWS services on which we'll focus - X

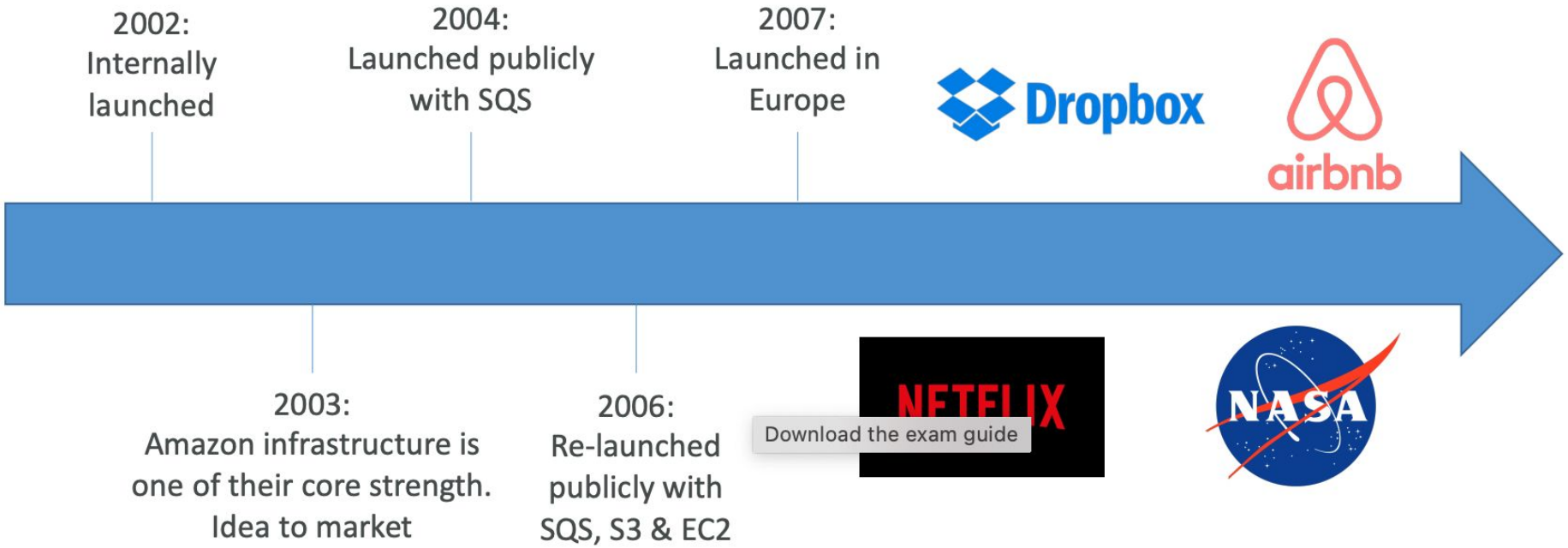
Analytics:

Amazon Athena

Amazon Kinesis

Amazon OpenSearch Service

AWS History



AWS Cloud Number Facts

In 2019, AWS had $35.02 billion in annual revenue

AWS accounts for 47% of the market in 2019 (Microsoft is 2nd with 22%)

Pioneer and Leader of the AWS Cloud Market for the 9th consecutive year

Over 1,000,000 active users

AWS Global Infrastructure

AWS Regions

AWS Availability Zones

AWS Data Centers

AWS Edge Locations / Points of Presence

AWS Regions

- Region is a physical location around the world where we cluster data centers
- AWS has regions all around the world
- Names can be us-east-1, eu-west-3…
- Most AWS services are region-scoped
- AWS 33 launched regions


Download the exam guide
Regions
Coming soon

How to choose an AWS Region?

Compliance with data governance and legal requirements: data never leaves a region without

your explicit permission

Proximity to customers: reduced latency

Available services within a Region: new services and new features aren’t available in every Region

Pricing: pricing varies region to region and is transparent in the service pricing page

AWS Availability Zones

Each region has many availability zones(usually 3, min is 3, max is 6). Example:

ap-southeast-2a

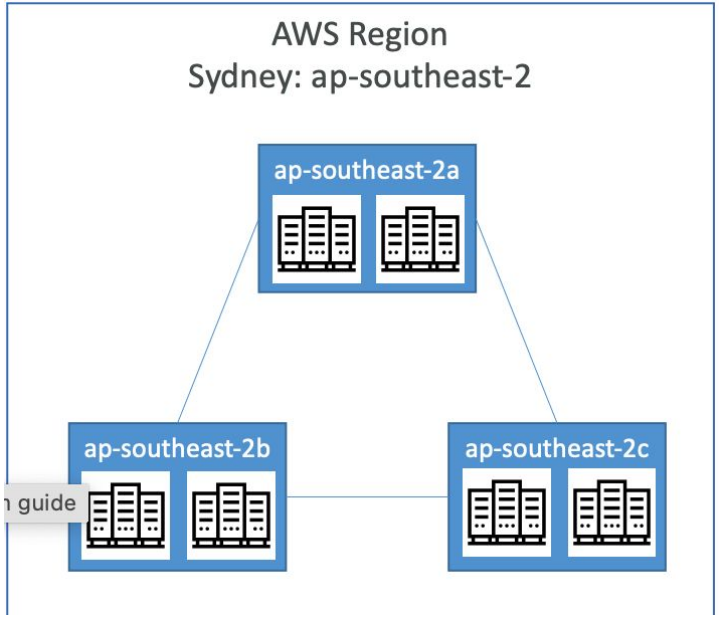
ap-southeast-2b

ap-southeast-2c

Each availability zone (AZ) is one or more discrete data centers with redundant power, networking, and connectivity

They’re separate from each other, so that they’re isolated from disasters

They’re connected with high bandwidth,ultra-low latency networking



AWS Points of Presence (Edge Locations)

Amazon has 400+ Points of Presence (400+ Edge Locations & 10+ Regional Caches) in 90+ cities across 40+ countries

Content is delivered to end users with lower latency

Lecture II

IAM: Users & Groups

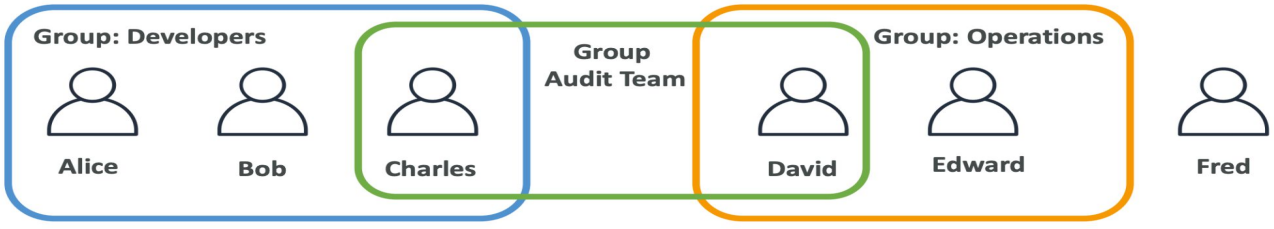
IAM = Identity and Access Management, Global service

Root account created by default, shouldn’t be used or shared

Users are people within your organization, and can be grouped

Groups only contain users, not other groups

Users don’t have to belong to a group, and user can belong to multiple groups



IAM: Permissions

Users or Groups can be assigned JSON documents called policies

These policies define the permissions of the users

In AWS you apply the least privilege principle: don’t give more permissions than a user needs

A policy is a statement that specifies a combination of

- Who
- What action
- Which AWS resource
- When
- Where
- How

IAM: Permissions example

- Who
- What action
- Which AWS resource
- When
- Where
- How

- Ali
- Cab GET/PUT object in S3
- Bucket=”*”
- Until Dec 31,2025
- From IP range 123.456.789.012
- If using MFA

IAM: Permissions example

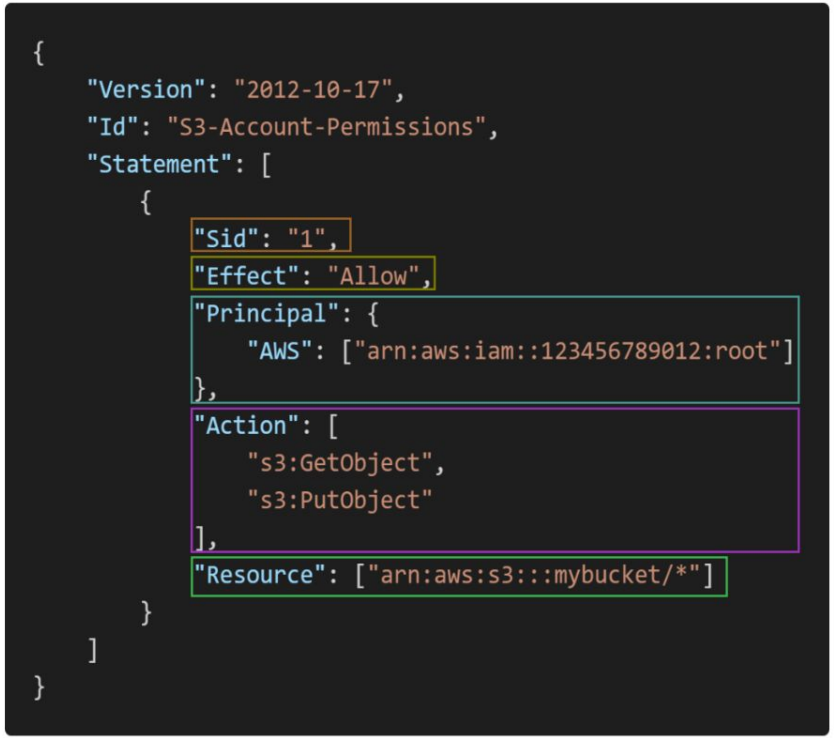
Example of an Amazon S3 Read-Only Access Template

```
{
 "Statement": [
   {
     "Effect": "Allow",
     "Action": ["s3:Get*", "s3:List*"],
     "Resource": "*"
   }
 ]
}
```

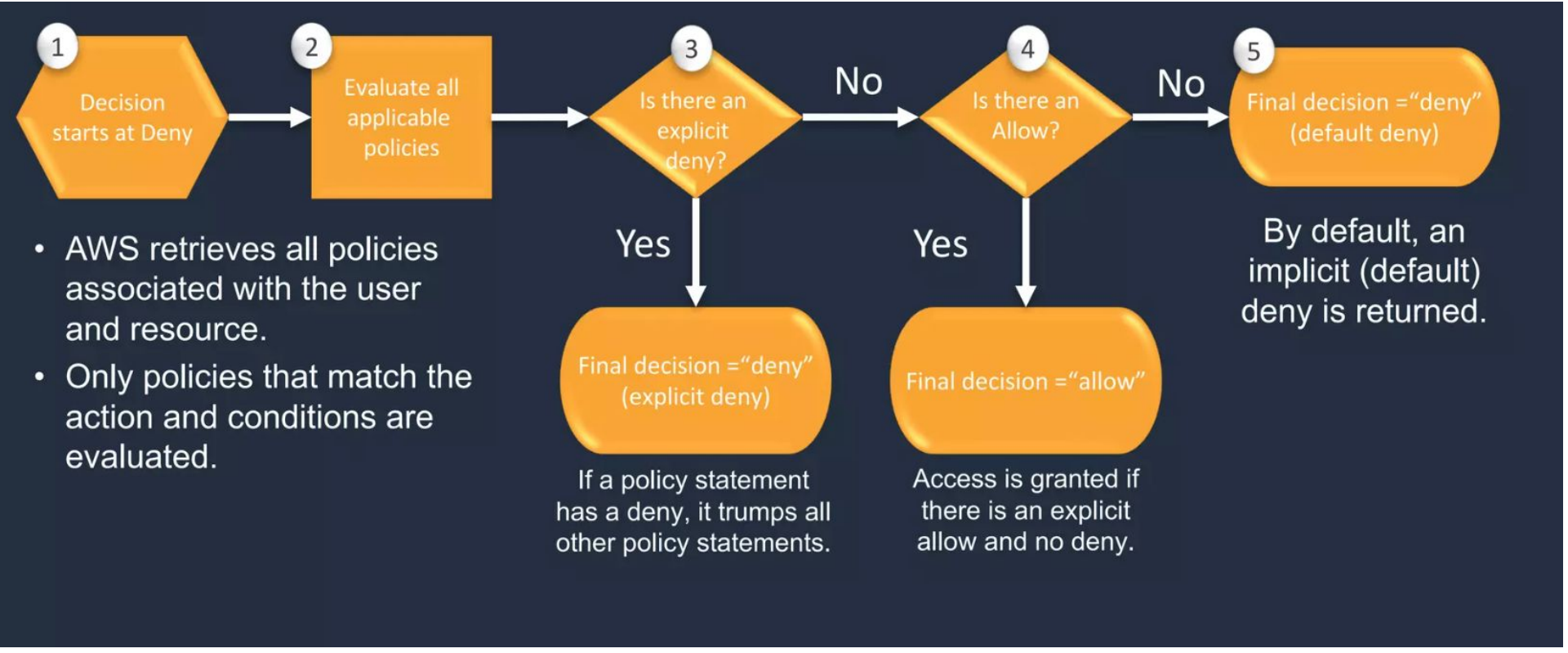
```
 "Statement": {
  "Effect": "Allow",
  "Action": "s3:ListBucket",
  "Resource": "arn:aws:s3:::example_bucket”
}
```

IAM Policies Structure

- Consists of
  - Version: policy language version, always include "2012-10-17"
  - Id: an identifier for the policy (optional)
  - Statement: one or more individual statements (required)
- Statements consists of
  - Sid: an identifier for the statement (optional)
  - Effect: whether the statement allows or denies access (Allow, Deny)
  - Principal: account/user/role to which this policy applied to
  - Action: list of actions this policy allows or denies
  - Resource: list of resources to which the actions applied to
  - Condition: conditions for when this policy is in effect (optional)



IAM: Policy Evaluation Logic



IAM – Password Policy

- Strong passwords = higher security for your account
- In AWS, you can setup a password policy:
- Set a minimum password length
- Require specific character types:
  - including uppercase letters
  - lowercase letters
  - numbers
  - non-alphanumeric characters
- Allow all IAM users to change their own passwords
- Require users to change their password after some time (password expiration)
- Prevent password re-use

Multi Factor Authentication - MFA

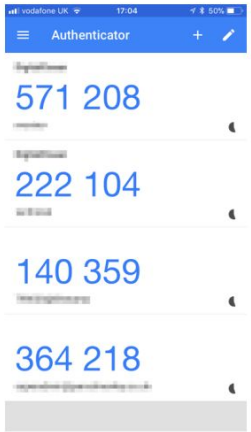
Users have access to your account and can possibly change configurations or delete resources in your AWS account

You want to protect your Root Accounts and IAM users

MFA = password you know + security device you own

MFA devices options in AWS

Virtual MFA device



Google Authenticator
(phone only)

Authy
(phone only)

Hardware Key Fob MFA Device

Provided by Gemalto (3rd party)

Universal 2nd Factor (U2F) Security Key

YubiKey by Yubico (3rd party)

How can users access AWS ?

- To access AWS, you have three options
  - AWS Management Console (protected by password + MFA)
  - AWS Command Line Interface (CLI): protected by access keys
  - AWS Software Developer Kit (SDK) - for code: protected by access keys
- Access Keys are generated through the AWS Console
- Users manage their own access keys
- Access Keys are secret, just like a password. Don’t share them
- Access Key ID ~= username
- Secret Access Key ~= password

What’s the AWS CLI?

A tool that enables you to interact with AWS services using commands in your command-line shell

Direct access to the public APIs of AWS services

You can develop scripts to manage your resources

It’s open-source https://github.com/aws/aws-cli

Alternative to using AWS Management Console

What’s the AWS SDK?

- AWS Software Development Kit (AWS SDK)
- Language-specific APIs (set of libraries)
- Enables you to access and manage AWS services programmatically
- Embedded within your application
- Supports
  - SDKs (JavaScript, Python, PHP, .NET, Ruby, Java, Go, Node.js, C++)
  - Mobile SDKs (Android, iOS, …)
  - IoT Device SDKs (Embedded C, Arduino, …)
- Example: AWS CLI is built on AWS SDK for Python

IAM Roles for Services

- Some AWS service will need to perform actions on your behalf
- To do so, we will assign permissions to AWS services with IAM Roles
- Common roles:
  - EC2 Instance Roles
  - Lambda Function Roles
  - Roles for CloudFormation

IAM Security Tools

- IAM Credentials Report (account-level)
  - a report that lists all your account's users and the status of their various credentials
- IAM Access Advisor (user-level)
  - Access advisor shows the service permissions granted to a user and when those services were last accessed.
  - You can use this information to revise your policies.

IAM Guidelines & Best Practices

Don’t use the root account except for AWS account setup

One physical user = One AWS user

Assign users to groups and assign permissions to groups

Create a strong password policy

Use and enforce the use of Multi Factor Authentication (MFA)

Create and use Roles for giving permissions to AWS services

Use Access Keys for Programmatic Access (CLI / SDK)

Audit permissions of your account using IAM Credentials Report & IAM

Access Advisor

Never share IAM users & Access Keys

Amazon EC2

Amazon EC2

- EC2 is one of the most popular of AWS' offering
- EC2 = Elastic Compute Cloud = Infrastructure as a Service
- It mainly consists in the capability of :
  - Renting virtual machines (EC2)
  - Storing data on virtual drives (EBS)
  - Distributing load across machines (ELB)
  - Scaling the services using an auto-scaling group (ASG)

- Knowing EC2 is fundamental to understand how the Cloud works

EC2 sizing & configuration options

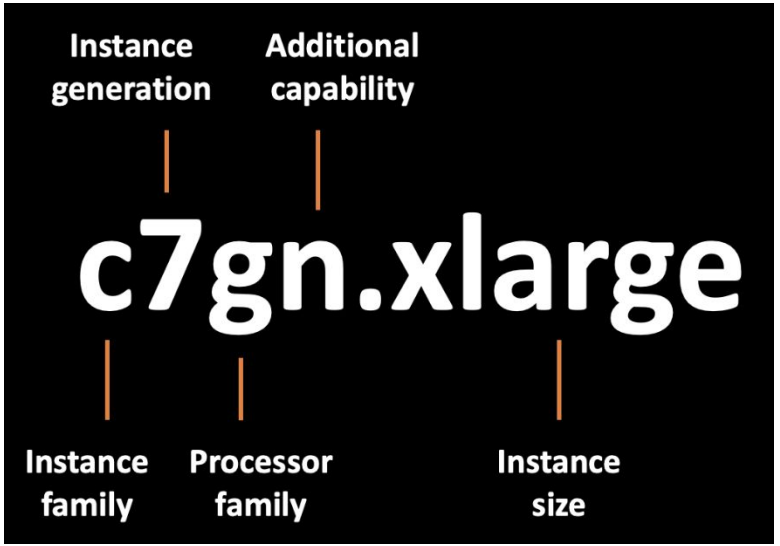
- Operating System (OS): Linux, Windows or Mac OS
- How much compute power & cores (CPU)
- How much random-access memory (RAM)
- How much storage space:
  - Network-attached (EBS & EFS)
  - hardware (EC2 Instance Store)
- Network card: speed of the card, Public IP address
- Firewall rules: security group
- Bootstrap script (configure at first launch): EC2 User Data

EC2 User Data

- It is possible to bootstrap our instances using an EC2 User data script.
- bootstrapping means launching commands when a machine starts
- That script is only run once at the instance first start
- EC2 user data is used to automate boot tasks such as:
  - Installing updates
  - Installing software
  - Downloading common files from the internet
- Anything you can think of

The EC2 User Data Script runs with the root user

EC2 Instance Types - Overview



EC2 Instance Types – General Purpose

- Great for a diversity of workloads such as web servers or code repositories
- Balance between:
  - Compute
  - Memory
  - Networking
- In the course, we will be using the t2.micro which is a General Purpose EC2 instance

EC2 Instance Types – Compute Optimized

Great for compute-intensive tasks that require high performance processors:

Batch processing workloads

Media transcoding

High performance web servers

High performance computing (HPC)

Scientific modeling & machine learning

Dedicated gaming servers

EC2 Instance Types – Memory Optimized

Fast performance for workloads that process large data sets in memory

Use cases:

High performance, relational/non-relational databases

Distributed web scale cache stores

In-memory databases optimized for BI (business intelligence)

Applications performing real-time processing of big unstructured data

EC2 Instance Types – Storage Optimized

Great for storage-intensive tasks that require high, sequential read and write access to large data sets on local storage

Use cases:

High frequency online transaction processing (OLTP) systems

Relational & NoSQL databases

Cache for in-memory databases (for example, Redis)

Data warehousing applications

Distributed file systems

EC2 Instance Types: example

Instance	vCPU	Mem (GiB)	Storage	Network Performance	EBS Bandwidth (Mbps)
t2.micro	1	1	EBS-Only	Low to Moderate	
t2.xlarge	4	16	EBS-Only	Moderate	
c5d.4xlarge	16	32	1 x 400 NVMe SSD	Up to 10 Gbps	4,750
r5.16xlarge	64	512	EBS Only	20 Gbps	13,600
m5.8xlarge	32	128	EBS Only	10 Gbps	6,800

t2.micro is part of the AWS free tier (up to 750 hours per month)

END