**Cloud Computing**

* Cloud computing is the on-demand delivery of compute power, database storage, applications, and other IT resources
* Through a cloud services platform with pay-as-you-go pricing
* You can provision exactly the right type and size of computing resources you need
* You can access as many resources as you need, almost instantly
* Simple way to access servers, storage, databases and a set of application services
* Amazon Web Services owns and maintains the network-connected hardware required for these application services, while you provision and use what you need via a web application.

### The Deployment Models of the Cloud

| **Private Cloud:** | **Public Cloud:** | **Hybrid Cloud:** |
| --- | --- | --- |

### Six Advantages of Cloud Computing

* **Trade capital expense (CAPEX) for operational expense (OPEX)**
  + Pay On-Demand: don’t own hardware
  + Reduced Total Cost of Ownership (TCO) & Operational Expense (OPEX)
* **Benefit from massive economies of scale**
  + Prices are reduced as AWS is more efficient due to large scale
* **Stop guessing capacity**
  + Scale based on actual measured usage
* **Increase speed and agility**
* **Stop spending money running and maintaining data centers**
* **Go global in minutes:** leverage the AWS global infrastructure

### ypes of Cloud Computing

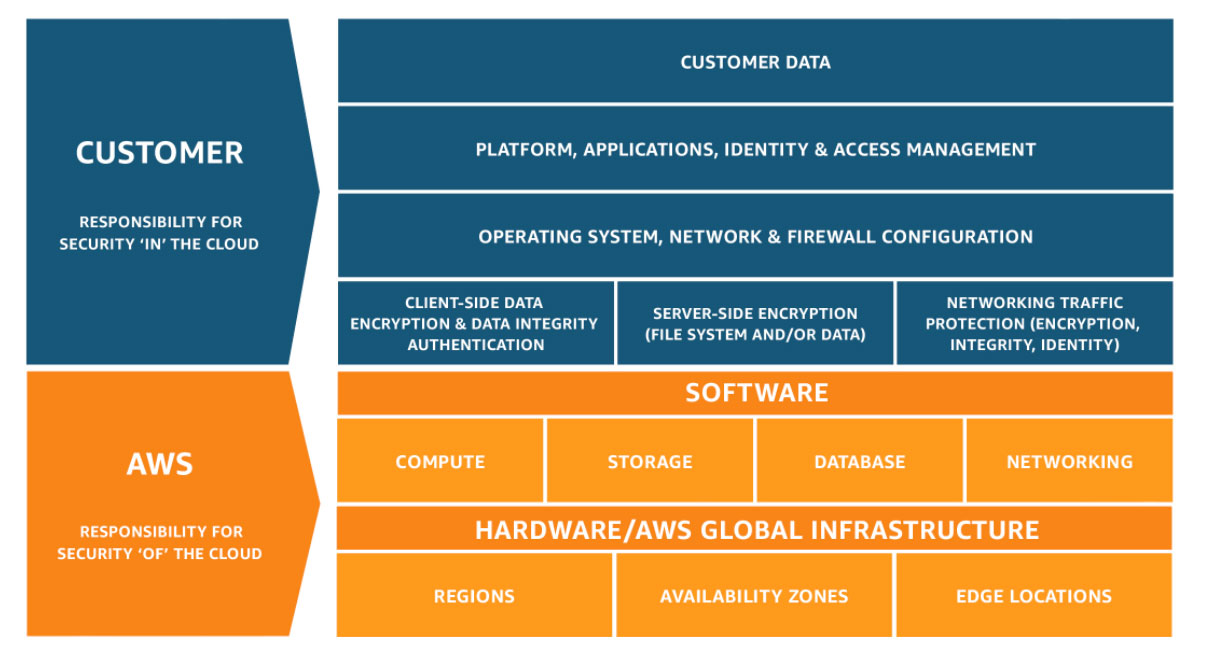
* **Infrastructure as a Service (IaaS)**
  + Provide building blocks for cloud IT
  + Provides networking, computers, data storage space
  + Highest level of flexibility
  + Easy parallel with traditional on-premises IT
* **Platform as a Service (PaaS)**
  + Removes the need for your organization to manage the underlying infrastructure
  + Focus on the deployment and management of your applications
* **Software as a Service (SaaS)**
  + Completed product that is run and managed by the service provider

## AWS Global Infrastructure

* AWS Regions
* AWS Availability Zones
* AWS Data Centers
* AWS Edge Locations / Points of Presence

## Shared Responsibility Model

* CUSTOMER = RESPONSIBILITY FOR THE SECURITY **IN** THE CLOUD
* AWS = RESPONSIBILITY FOR THE SECURITY **OF** THE CLOUD

[](https://github.com/kananinirav/AWS-Certified-Cloud-Practitioner-Notes/blob/master/images/Shared_Responsibility_Model.jpg)

**IAM: Identity Access & Management**

## What Is IAM?

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

### 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

### 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)

Example:

```json

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": "ec2:Describe\*",

"Resource": "\*"

},

{

"Effect": "Allow",

"Action": "elasticloadbalancing:Describe\*",

"Resource": "\*"

},

{

"Effect": "Allow",

"Action": [

"cloudwatch:ListMetrics",

"cloudwatch:GetMetricStatistics",

"cloudwatch:Describe\*"

],

"Resource": "\*"

}

]

}

```

### 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

### 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 with the IAM Credentials Report

- \*\*Never share IAM users & Access Keys\*\*

### Shared Responsibility Model for IAM

| AWS | YOU |

| ---------------------------------------- | ------------------------------------------------------------------------------------------------------------------------ |

| Infrastructure (global network security) | Users, Groups, Roles, Policies management and monitoring |

| Configuration and vulnerability analysis | Enable MFA on all accounts |

| Compliance validation | Rotate all your keys often, Use IAM tools to apply appropriate permissions, Analyze access patterns & review permissions |

## 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

- Main benefit of MFA: if a password is stolen or hacked, the account is not compromised

## MFA devices options in AWS

- Virtual MFA device (Support for multiple tokens on a single device.)

- Google Authenticator (phone only)

- Authy (multi-device)

- Universal 2nd Factor (U2F) Security Key (Support for multiple root and IAM users using a single security key)

- YubiKey by Yubico (3rd party)

- Hardware Key Fob MFA Device

- Hardware Key Fob MFA Device for AWS GovCloud (US)

## 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 Section – Summary

- \*\*Users:\*\* mapped to a physical user, has a password for AWS Console

- \*\*Groups:\*\* contains users only

- \*\*Policies:\*\* JSON document that outlines permissions for users or groups

- \*\*Roles:\*\* for EC2 instances or AWS services

- \*\*Security:\*\* MFA + Password Policy

- \*\*AWS CLI:\*\* manage your AWS services using the command-line

- \*\*AWS SDK:\*\* manage your AWS services using a programming language

- \*\*Access Keys:\*\* access AWS using the CLI or SDK

- \*\*Audit:\*\* IAM Credential Reports & IAM Access Advisor

\* \* \*

**https://github.com/kananinirav/AWS-Certified-Cloud-Practitioner-Notes**

# EC2: Virtual Machines

- [EC2: Virtual Machines](#ec2-virtual-machines)

- [What is Amazon EC2?](#what-is-amazon-ec2)

- [EC2 sizing & configuration options](#ec2-sizing--configuration-options)

- [EC2 User Data](#ec2-user-data)

- [EC2 Instance Types - Overview](#ec2-instance-types---overview)

- [General Purpose](#general-purpose)

- [Compute Optimized](#compute-optimized)

- [Memory Optimized](#memory-optimized)

- [Storage Optimized](#storage-optimized)

- [EC2 Instance Types: example](#ec2-instance-types-example)

- [Introduction to Security Groups](#introduction-to-security-groups)

- [Deeper Dive](#deeper-dive)

- [Security Groups Diagram](#security-groups-diagram)

- [Good to know](#good-to-know)

- [Classic Ports to know](#classic-ports-to-know)

- [EC2 Instance Launch Types](#ec2-instance-launch-types)

- [On Demand Instance](#on-demand-instance)

- [Reserved Instances](#reserved-instances)

- [Savings Plans](#savings-plans)

- [Spot Instances](#spot-instances)

- [Dedicated Hosts](#dedicated-hosts)

- [Dedicated Instances](#dedicated-instances)

- [Capacity Reservations](#capacity-reservations)

- [Which purchasing option is right for me?](#which-purchasing-option-is-right-for-me)

- [Price Comparison Example – m4.large – us-east-1](#price-comparison-example--m4large--us-east-1)

- [Shared Responsibility Model for EC2](#shared-responsibility-model-for-ec2)

- [EC2 Section – Summary](#ec2-section--summary)

## What is Amazon EC2?

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) Cloud.

- 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

- You can use different types of EC2 instances that are optimised for different use cases (<https://aws.amazon.com/ec2/instance-types/>)

- [General Purpose](#general-purpose)

- [Compute Optimized](#compute-optimized)

- [Memory Optimized](#memory-optimized)

- [Storage Optimized](#storage-optimized)

- Accelerated Computing

- AWS has the following naming convention: m5.2xlarge

- m: instance class

- 5: generation (AWS improves them over time)

- 2xlarge: size within the instance class

#### General Purpose

- Great for a diversity of workloads such as web servers or code repositories

- Balance between:

- Compute

- Memory

- Networking

#### 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

#### 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

#### 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)

## Introduction to Security Groups

- Security Groups are the fundamental of network security in AWS

- They control how traffic is allowed into or out of our EC2 Instances.

- Security groups only contain allow rules

- Security groups rules can reference by IP or by security group

### Deeper Dive

- Security groups are acting as a “firewall” on EC2 instances

- They regulate:

- Access to Ports

- Authorised IP ranges – IPv4 and IPv6

- Control of inbound network (from other to the instance)

- Control of outbound network (from the instance to other)

### Security Groups Diagram

![ Security Groups Diagram](../images/Security\_Groups\_Diagram.png)

### Good to know

- Can be attached to multiple instances

- Locked down to a region / VPC combination

- Does live “outside” the EC2 – if traffic is blocked the EC2 instance won’t see it

- It’s good to maintain one separate security group for SSH access

- If your application is not accessible (time out), then it’s a security group issue

- If your application gives a “connection refused“ error, then it’s an application error or it’s not launched

- All inbound traffic is \*\*blocked\*\* by default

- All outbound traffic is \*\*authorized\*\* by default

## Classic Ports to know

- 22 = SSH (Secure Shell) - log into a Linux instance

- 21 = FTP (File Transfer Protocol) – upload files into a file share

- 22 = SFTP (Secure File Transfer Protocol) – upload files using SSH

- 80 = HTTP – access unsecured websites

- 443 = HTTPS – access secured websites

- 3389 = RDP (Remote Desktop Protocol) – log into a Windows instance

## EC2 Instance Launch Types

- [\*\*On Demand Instances\*\*](#on-demand-instance): short workload, predictable pricing

- [\*\*Reserved\*\*](#reserved-instances): (1 & 3 years)

- \*\*Reserved Instances\*\*: long workloads

- \*\*Convertible Reserved Instances\*\*: long workloads with flexible instances

- [\*\*Savings Plans\*\*](#savings-plans) (1 & 3 years): commitment to an amount of usage, long workload

- [\*\*Spot Instances\*\*](#spot-instances): short workloads, for cheap, can lose instances

- [\*\*Dedicated Instances\*\*](#dedicated-instances): no other customers will share your hardware

- [\*\*Dedicated Hosts\*\*](#dedicated-hosts): book an entire physical server, control instance placement

- [\*\*Capacity Reservations\*\*](#capacity-reservations): reserve capacity in a specific AZ for any duration

### On Demand Instance

- Pay for what you use:

- Linux or Windows - billing per second, after the first minute

- All other operating systems - billing per hour

- Has the highest cost but no upfront payment

- No long-term commitment

- Recommended for \*\*short-term\*\* and \*\*un-interrupted workloads\*\*, where you can't predict how the application will behave

### Reserved Instances

- Up to 72% discount compared to On-demand

- You reserve a specific instance attributes (Instance Type, Region, Tenancy, OS)

- Reservation Period – 1 year (+discount) or 3 years (+++discount)

- Payment Options – No Upfront (+), Partial Upfront (++), All Upfront (+++)

- Reserved Instance’s Scope – Regional or Zonal (reserve capacity in an AZ)

- Recommended for steady-state usage applications (think database)

- You can buy and sell in the Reserved Instance Marketplace

- Convertible Reserved Instance

- Can change the EC2 instance type, instance family, OS, scope and tenancy

- Up to 66% discount

### Savings Plans

- Get a discount based on long-term usage (up to 72% - same as RIs)

- Commit to a certain type of usage ($10/hour for 1 or 3 years)

- Usage beyond EC2 Savings Plans is billed at the On-Demand price

- Locked to a specific instance family & AWS region (e.g., M5 in us-east-1)

- Flexible across:

- Instance Size (e.g., m5.xlarge, m5.2xlarge)

- OS (e.g., Linux, Windows)

- Tenancy (Host, Dedicated, Default)

### Spot Instances

- Can get a discount of up to 90% compared to On-demand

- Instances that you can “lose” at any point of time if your max price is less than the current spot price

- The MOST cost-efficient instances in AWS

- Useful for workloads that are resilient to failure

- Batch jobs

- Data analysis

- Image processing

- Any distributed workloads

- Workloads with a flexible start and end time

- Not suitable for critical jobs or databases

### Dedicated Hosts

- A physical server with EC2 instance capacity fully dedicated to your use

- Allows you address compliance requirements and use your existing server- bound software licenses (per-socket, per-core, pe—VM software licenses)

- Purchasing Options:

- On-demand – pay per second for active Dedicated Host

- Reserved - 1 or 3 years (No Upfront, Partial Upfront, All Upfront)

- The most expensive option

- Useful for software that have complicated licensing model (BYOL – Bring Your Own License)

- Or for companies that have strong regulatory or compliance needs

### Dedicated Instances

- Instances run on hardware that’s dedicated to you

- May share hardware with other instances in same account

- No control over instance placement (can move hardware after Stop / Start)

### Capacity Reservations

- Reserve On-Demand instances capacity in a specific AZ for any duration

- You always have access to EC2 capacity when you need it

- No time commitment (create/cancel anytime), no billing discounts

- Combine with Regional Reserved Instances and Savings Plans to benefit from billing discounts

- You’re charged at On-Demand rate whether you run instances or not

- Suitable for short-term, uninterrupted workloads that needs to be in a specific AZ

## Which purchasing option is right for me?

- On demand: coming and staying in resort whenever we like, we pay the full price

- Reserved: like planning ahead and if we plan to stay for a long time, we may get a good discount.

- Savings Plans: pay a certain amount per hour for certain period and stay in any room type (e.g., King, Suite, Sea View, …)

- Spot instances: the hotel allows people to bid for the empty rooms and the highest bidder keeps the rooms. You can get kicked out at any time

- Dedicated Hosts: We book an entire building of the resort

- Capacity Reservations: you book a room for a period with full price even you don’t stay in it

## Price Comparison Example – m4.large – us-east-1

| Price Type | Price (per hour) |

| -------------------------------------- | ------------------------------------------ |

| On-Demand | $0.10 |

| Spot Instance (Spot Price) | $0.038 - $0.039 (up to 61% off) |

| Reserved Instance (1 year) | $0.062 (No Upfront) - $0.058 (All Upfront) |

| Reserved Instance (3 years) | $0.043 (No Upfront) - $0.037 (All Upfront) |

| EC2 Savings Plan (1 year) | $0.062 (No Upfront) - $0.058 (All Upfront) |

| Reserved Convertible Instance (1 year) | $0.071 (No Upfront) - $0.066 (All Upfront) |

| Dedicated Host | On-Demand Price |

| Dedicated Host Reservation | Up to 70% off |

| Capacity Reservations | On-Demand Price |

## Shared Responsibility Model for EC2

| AWS | USER |

| ---------------------------------------- | -------------------------------------------------------------------------------------- |

| Infrastructure (global network security) | Security Groups rules |

| Isolation on physical hosts | Operating-system patches and updates |

| Replacing faulty hardware | Software and utilities installed on the EC2 instance |

| Compliance validation | IAM Roles assigned to EC2 & IAM user access management, Data security on your instance |

## EC2 Section – Summary

- EC2 Instance: AMI (OS) + Instance Size (CPU + RAM) + Storage + security groups + EC2 User Data

- Security Groups: Firewall attached to the EC2 instance

- EC2 User Data: Script launched at the first start of an instance

- SSH: start a terminal into our EC2 Instances (port 22)

- EC2 Instance Role: link to IAM roles

- Purchasing Options: On-Demand, Spot, Reserved (Standard + Convertible + Scheduled), Dedicated Host, Dedicated Instance

\* \* \*