Day 1

EC2

What is EC2?

EC2 (Elastic Compute Cloud) is a web service interface that provides resizable compute capacity in the AWS cloud. It is designed for developers to have complete control over resources.EC2 instances can be resized and the number of instances scaled up or down as per our requirement. These instances can be launched in different locations or regions.

EC2 Components

In AWS EC2, the users must be aware about the EC2 components, their operating systems support, security measures, pricing structures, etc.

Operating System Support

Amazon EC2 supports multiple OS in which we need to pay additional licensing fees like: Red Hat Enterprise,Oracle Enterprise Linux, UNIX, Windows Server, etc.

Security

Users have complete control over the visibility of their AWS account. In AWS EC2, the security systems allow create groups and place running instances into it as per the requirement.

Pricing

AWS offers a variety of pricing options, depending on the type of resources, types of applications and database.

Migration

This service allows the users to move existing applications into EC2. It costs $80.00 per storage device and $2.49 per hour for data loading.

Features of EC2

Reliable – Amazon EC2 offers a highly reliable environment where replacement of instances is rapidly possible.

Designed for Amazon Web Services – Amazon EC2 works fine with Amazon services like Amazon S3, Amazon RDS.
.

Secure – Amazon EC2 works in Amazon Virtual Private Cloud to provide a secure and robust network to resources.

Flexible Tools – Amazon EC2 provides the tools for developers .

Inexpensive – Amazon EC2 wants us to pay only for the resources that we use.

How to Use AWS EC2

- Sign in to the AWS Management Console.
- Click on the EC2 service.
- Click on the Launch Instance button to create a new instance.
- Now, we have different Amazon Machine Images.
- Choose an Instance Type, and then click on the Next.
- The main setup page of EC2 Where,
- Number of Instances
- Tenancy
- Subnet**:** It is a range of IP addresses in a virtual cloud. In a specified subnet, you can add new AWS resources.
- Monitoring**:** We can monitor things such as CPU utilization. Right now, I uncheck the Monitoring.
- Delete on termination**:** It is checked means that the termination of an EC2 instance.
- Now, Add the Tags and then click on the Next.

Then we add two tags, i.e., the name of the server and department. Create as many tags as you can as it reduces the overall cost.

- Configure Security Group. The security group allows some specific traffic to access your instance.
- Review an EC2 instance that you have just configured, and then click on the Launch button.

.

What is Docker

Docker is an **open-source and freely available platform that offers tools and utilities** to create and manage containers**.**

It is a software development tool and a virtualization technology that makes it easy to develop, deploy, and manage applications by using containers. A container refers to a lightweight, stand-alone, executable package of a piece of software that contains all the libraries, configuration files, dependencies, and other necessary parts to operate the application

Applications run the same irrespective of where they are and what machine they are running on because the container provides the environment throughout the software development life cycle of the application. Since containers are isolated, they provide security, thus allowing multiple containers to run simultaneously on the given host. Also, containers are lightweight.

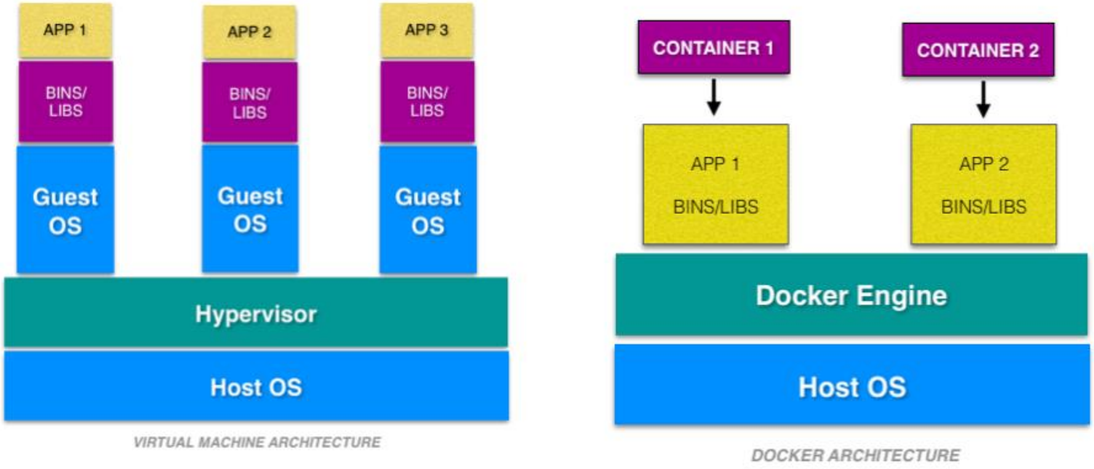
What are Virtual Machines?

A virtual machine is capable of performing tasks such as running applications and programs like a separate computer making them ideal for testing other operating systems like beta releases, creating operating system backups, and running software and applications. A host can have several virtual machines running at a specific time..

Another sector where VMs are of great use is server virtualization. In server virtualization, a physical server is divided into multiple isolated and unique servers, thereby allowing each server to run its operating system independently. Each virtual machine provides its virtual hardware, such as CPUs, memory, network interfaces, hard drives, and other devices.

OS Support and Architecture

The main difference between Docker and VMs lies in their architecture,



VMs have the host OS and guest OS inside each VM. A guest OS can be any OS, like Linux or Windows, irrespective of the host OS. In contrast It containers host on a single physical server with a host OS, which shares among them. Sharing the host OS between containers makes them light and increases the boot time. Docker containers are considered suitable to run multiple applications over a single OS kernel; whereas, virtual machines are needed if the applications or services required to run on different OS

Security

The second difference between VMs and Docker is that Virtual Machines are stand-alone with their kernel and security features. Therefore, applications needing more privileges and security run on virtual machines.

Portability

Another relevant Docker vs Virtual Machine difference is about portability: VMs are isolated from their OS, and so they are not ported across multiple platforms without incurring compatibility issues. Docker containers must be considered. Docker container packages are self-contained and can run applications in any environment, and since they don’t need a guest OS, they can be easily ported across different platforms. Docker containers can be easily deployed in servers since containers being lightweight can be started and stopped in very less time compared to virtual machines.

Performance

The last main Docker vs VM difference refers to performance: Virtual Machines are more resource-intensive than Docker containers as the virtual machines need to load the entire OS to start. The lightweight architecture of Docker containers is less resource-intensive than virtual machines

In the case of a virtual machine, resources like CPU, memory, and I/O may not be allocated permanently to containers — unlike in the case of a Docker container, where the resource usage works with the load or traffic.

Virtual Machine	**Docker Container**
Hardware-level process isolation	OS level process isolation
Each VM has a separate OS	Each container can share OS
Boots in minutes	Boots in seconds
VMs are of few GBs	Containers are lightweight (KBs/MBs)
Ready-made VMs are difficult to find	Pre-built docker containers are easily available
VMs can move to new host easily	Containers are destroyed and re-created rather than moving
Creating VM takes a relatively longer time	Containers can be created in seconds
More resource usage	Less resource usage