What is EC2?

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

Features of Amazon EC2

Amazon EC2 provides the following features:

- Virtual computing environments, known as *instances*
- Preconfigured templates for your instances, known as *Amazon Machine Images (AMIs)*, that package the bits you need for your server (including the operating system and additional software)
- Various configurations of CPU, memory, storage, and networking capacity for your instances, known as *instance types*
- Secure login information for your instances using *key pairs* (AWS stores the public key, and you store the private key in a secure place)
- Storage volumes for temporary data that's deleted when you stop, hibernate, or terminate your instance, known as *instance store volumes*
- Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as *Amazon EBS volumes*
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as *Regions* and *Availability Zones*
- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using *security groups*
- Static IPv4 addresses for dynamic cloud computing, known as *Elastic IP addresses*
- Metadata, known as *tags*, that you can create and assign to your Amazon EC2 resources
- Virtual networks you can create that are logically isolated from the rest of the AWS Cloud, and that you can optionally connect to your own network, known as *virtual private clouds* (VPCs)

2. What is the difference between Virtual Machines and Docker

    a. What is Docker

Docker is popular virtualization software that helps its users in developing, deploying, monitoring, and running applications in a Docker Container with all their dependencies.
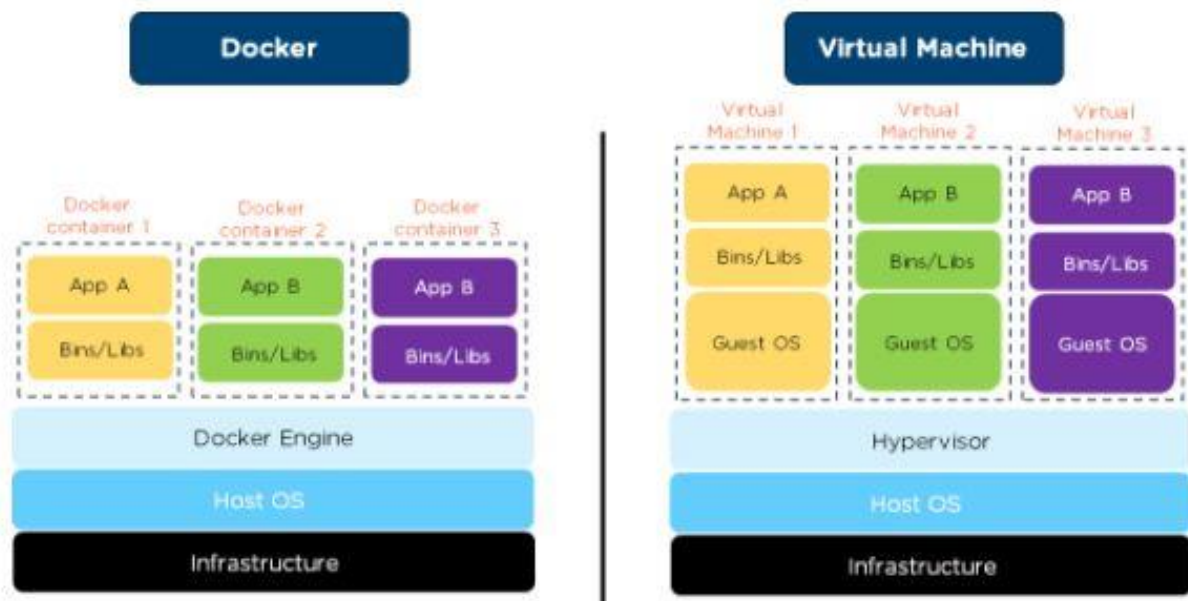
Docker containers include all dependencies (frameworks, libraries, etc.) to run an application in an efficient and bug-free manner.

Docker Containers have the following benefits:

- Light weight
- Applications run in isolation
- Occupies less space
- Easily portable and highly secure
- Short boot-up time

    b. What is Virtual Machine

A virtual machine (VM) is a computing environment or software that aids developers to access an operating system via a physical machine.

Differences	Docker	Virtual Machine
Operating system	Docker is a container-based model where containers are software packages used for executing an application on any operating system In Docker, the containers share the host OS kernel Here, multiple workloads can run on a single OS	It is not a container-based model; they use user space along with the kernel space of an OS It does not share the host kernel Each workload needs a complete OS or hypervisor
Performance	Docker containers result in high-performance as they use the same operating system with no additional software (like hypervisor) Docker containers can start up quickly and result in less boot-up time	Since VM uses a separate OS; it causes more resources to be used Virtual machines don't start quickly and lead to poor performance

Portability	With docker containers, users can create an application and store it into a container image. Then, he/she can run it across any host environment Docker container is smaller than VMs, because of which the process of transferring files on the host's filesystem is easier	It has known portability issues. VMs don't have a central hub and it requires more memory space to store data While transferring files, VMs should have a copy of the OS and its dependencies because of which image size is increased and becomes a tedious process to share data
Speed	The application in Docker containers starts with no delay since the OS is already up and running These containers were basically designed to save time in the deployment process of an application	It takes a much longer time than it takes for a container to run applications To deploy a single application, Virtual Machines need to start the entire OS, which would cause a full boot process