

## Docker Assignment Day 1

Yusuf Ahsan Ansari, Batch 3

# Docker



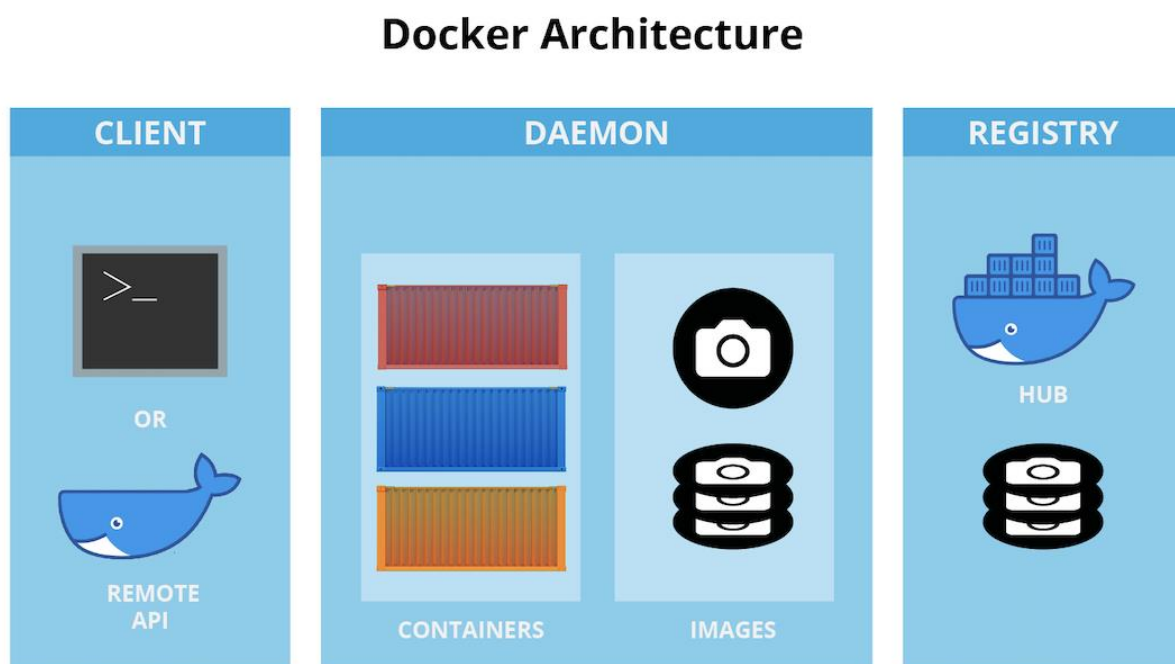
**Docker is an open source platform for building, deploying, and managing containerized applications. Learn about containers, how they compare to VMs, and why Docker is so widely adopted and used.**

## Why use Docker

- **Improved—and seamless—portability:** While LXC containers often reference machine-specific configurations, Docker containers run without modification across any desktop, data center and cloud environment.
- **Even lighter weight and more granular updates:** With LXC, multiple processes can be combined within a single container. With Docker containers, only one process can run in each container. This makes it possible to build an application that can continue running while one of its parts is taken down for an update or repair.
- **Automated container creation:** Docker can automatically build a container based on application source code.
- **Container versioning:** Docker can track versions of a container image, roll back to previous versions, and trace who built a version and how. It can even upload only the deltas between an existing version and a new one.

- **Container reuse:** Existing containers can be used as *base images*—essentially like templates for building new containers.
- **Shared container libraries:** Developers can access an open-source registry containing thousands of user-contributed containers.

## Docker Architecture



## Docker tools and terms

Some of the tools and terminology you'll encounter when using Docker include:

### **DockerFile**

Every Docker container starts with a simple text file containing instructions for how to build the Docker container image. *DockerFile* automates the process of Docker image creation. It's essentially a list of command-line interface (CLI) instructions that Docker Engine will run in order to assemble the image.

### **Docker images**

*Docker images* contain executable application source code as well as all the tools, libraries, and dependencies that the application code needs to run as a container.

When you run the Docker image, it becomes one instance (or multiple instances) of the container.

It's possible to build a Docker image from scratch, but most developers pull them down from common repositories. Multiple Docker images can be created from a single base image, and they'll share the commonalities of their stack.

## **Docker containers**

Docker containers are the live, running instances of Docker images. While Docker images are read-only files, containers are live, ephemeral, executable content. Users can interact with them, and administrators can adjust their settings and conditions using docker commands.

## **Docker Hub**

Docker Hub is the public repository of Docker images that calls itself the “world’s largest library and community for container images.” It holds over 100,000 container images sourced from commercial software vendors, open-source projects, and individual developers. It includes images that have been produced by Docker, Inc., certified images belonging to the Docker Trusted Registry, and many thousands of other images

## **Docker daemon**

Docker daemon is a service running on your operating system, such as Microsoft Windows or Apple MacOS or iOS. This service creates and manages your Docker images for you using the commands from the client, acting as the control center of your Docker implementation.

## **Docker registry**

A Docker registry is a scalable open-source storage and distribution system for docker images. The registry enables you to track image versions in repositories, using tagging for identification. This is accomplished using git, a version control tool.

## **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:

1. Virtual computing environments, known as instances
2. 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)
3. Various configurations of CPU, memory, storage, and networking capacity for your instances, known as instance types
4. Secure login information for your instances using key pairs (AWS stores the public key, and you store the private key in a secure place)
5. Storage volumes for temporary data that's deleted when you stop, hibernate, or terminate your instance, known as instance store volumes
6. Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as Amazon EBS volumes
7. Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as Regions and Availability Zones
8. A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using security groups
9. Static IPv4 addresses for dynamic cloud computing, known as Elastic IP addresses

10. Metadata, known as tags, that you can create and assign to your Amazon EC2 resources

11. 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).

## What Is Virtual Machine?

A Virtual Machine (VM), on the other hand, is created to perform tasks that if otherwise performed directly on the host environment, may prove to be risky. VMs are isolated from the rest of the system; the software inside the virtual machine cannot tamper with the host computer. Therefore, implementing tasks such as accessing virus-infected data and testing of operating systems are done using virtual machines. We can define a virtual machine as: A virtual machine is a computer file or software usually termed as a guest, or an image that is created within a computing environment called the host.

VMs are broadly divided into two categories depending upon their use:

1. System Virtual Machines: A platform that allows multiple VMs, each running with its copy of the operating system to share the physical resources of the host system. Hypervisor, which is also a software layer, provides the virtualization technique. The hypervisor executes at the top of the operating system or the hardware alone.

2. Process Virtual Machine: Provides a platform-independent programming environment. The process virtual machine is designed to hide the information of the underlying hardware and operating system and allows the program to execute in the same manner on every given platform.

Benefits of virtual Machine:

## Benefits of a Virtual Machine



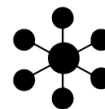
### Operational flexibility

Operate separate instances of multiple OS types



### Reducing overhead

Run multiple virtual machines on the same underlying hardware



### Centralization

Consolidate systems to simplify management



### Scalability

Easily scale your virtual environment as your business grows



### Disaster recovery

Restore data and system states from VM instances

## Difference Between Docker and Virtual Machine

