Why is Docker needed?

- **Isolation:** Containers isolate applications from each other and from the underlying infrastructure, preventing conflicts and ensuring reliability.
- **Efficiency:** Containers share the host OS kernel, making them more lightweight than virtual machines, resulting in faster startup times and better resource utilization.
- **Portability:** Containers encapsulate the application and its dependencies, ensuring consistency across different environments.
- **Scalability:** Containers can be easily scaled up or down to meet the demands of applications.
- **Consistency:** Docker ensures that the development, testing, and production environments are consistent, reducing the "it works on my machine" problem.
- **Ecosystem**: Docker has a rich ecosystem with a wide range of tools and services that complement containerization, making it a versatile platform for application deployment and management.
- **Deployment:** Docker makes it easier and safer to deploy. Instead of managing packages and their versions, we upload our Docker image to a server.

What is an image?

A package or template used to create one or more containers

What is a container?

Instances of an image, isolated from each other, with their own environment

What is Docker?

An open-source project that automates the deployment of software applications inside containers by providing an additional layer of abstraction and automation of OS-level virtualization on Linux.

Docker is an OS virtualized software platform that allows IT organizations to quickly create, deploy, and run applications in Docker containers, which have all the dependencies within them. The **container** itself is a very lightweight package with all the instructions and dependencies—such as frameworks, libraries, and bins—within it.

1. What is Docker?

Definition: Docker is a platform for developing, shipping, and running applications in containers.

2. What is Docker Image?

An executable package that includes application code, libraries, dependencies, and a runtime.

3. What is a container?

Container: A lightweight, standalone, executable package that includes everything needed to run a piece of software, including the code, runtime, libraries, and system tools.

What is Hadoop and its components?

Definition: Hadoop is an open-source framework for distributed storage and processing of large datasets.

Components:

- **Hadoop Distributed File System (HDFS):** Distributed storage system for big data.
- MapReduce: is a programming model for processing and generating large datasets.

What is Mapreduce?

Definition: A programming model and processing engine for distributed data processing on large clusters.

- **Mapper:** Processes input data and produces intermediate key-value pairs.
- **Reducer:** Aggregates and processes the intermediate key-value pairs to produce the final output.

Why Hadoop MapReduce is Important:

- Scalability: Scales horizontally by distributing data and processing across multiple nodes.
- Fault Tolerance: Handles node failures by replicating data and rerunning tasks on other nodes.
- Batch Processing: Well-suited for processing large volumes of data in a batch-oriented manner.