DARSHAN NIKAM DATE: /04/2024

DOCKER

Docker is a platform that makes it easier to develop, deploy, and run applications using containerization technology. Containerization allows to package an application and its dependencies into a standardized unit called a container. These containers are portable, lightweight, and can run consistently across different environments.

Key Concepts:

❖ ***Docker Engine:***

- ***Core Component:*** ***Docker Engine is the heart of Docker, responsible for managing containers and their lifecycle.***
- ***Architecture:*** ***It consists of a server (daemon) and a REST API that exposes functionality for interacting with containers, images, volumes, and networks.***
- ***Containerd:*** ***Docker Engine uses Containerd as its core container runtime, which handles the low-level container operations like starting, stopping, and managing container processes.***
- ***CLI (Command Line Interface):*** ***Docker provides a command-line interface (CLI) tool that allows users to interact with Docker Engine, performing actions such as building images, running containers, and managing Docker resources.***

❖ ***Dockerfile:***

- ***Definition:*** ***A Dockerfile is a plain text file that contains a set of instructions for building a Docker image.***
- ***Instructions:*** ***Dockerfile instructions include commands like FROM, RUN, COPY, ADD, EXPOSE, CMD, and ENTRYPOINT, among others.***
- ***Layered Filesystem:*** ***Each instruction in a Dockerfile creates a layer in the Docker image's filesystem. This layered approach allows for efficient caching and incremental updates during image builds.***
- ***Best Practices:*** ***Writing efficient and maintainable Dockerfiles involves following best practices, such as minimizing the number of layers, using multi-stage builds, and keeping images small and secure.***

❖ ***Docker Image:***

- ***Definition:*** ***A Docker image is a lightweight, standalone, and executable package that contains everything needed to run an application, including code, runtime, libraries, and dependencies.***
- ***Layers******: Docker images are composed of layers, where each layer represents a set of filesystem changes. This layered filesystem enables image sharing and efficient storage.***
- ***Immutable******: Docker images are immutable, meaning once built, their contents cannot be changed. Instead, any modifications result in the creation of new layers.***
- ***Tagging:*** ***Images can be tagged with version identifiers or labels to differentiate between different versions or variants of the same application.***

❖ ***Docker Container:***

- ***Instance of an Image:*** ***A Docker container is a runnable instance of a Docker image. It encapsulates an application along with its runtime environment and dependencies.***
- ***Isolation******: Containers provide process and filesystem isolation, ensuring that applications running in one container do not interfere with those running in other containers or on the host system.***
- ***Lifecycle:*** ***Containers have a lifecycle that includes creation, start, stop, pause, and deletion. Docker Engine manages these lifecycle operations.***
- ***Statelessness:*** ***Containers are designed to be stateless, meaning they do not retain data or changes made during runtime by default. However, data can be persisted using volumes or other external storage mechanisms.***