Title: Docker Summary

Introduction to Docker

Docker is a platform used for developing, shipping, and running applications inside containers. It enables developers to package applications with all dependencies into a standardized unit called a container, which can run on any system with Docker installed.

Introduction to Containers & Docker Architecture

Understood the difference between containers and virtual machines (VMs).

Containers share the host OS kernel, making them lightweight and faster than VMs.

Learned about the key components of Docker:

Docker Engine (includes Docker Daemon and Docker CLI)

Docker Images

Docker Containers

Docker Hub (registry to share/pull images)

Docker Installation and Basic Commands

Installed Docker on local system (Linux/Windows).

Practiced basic Docker commands:

```
docker --version
```

```
docker pull <image_name>
```

```
docker run <image_name>
```

`docker ps`, `docker ps -a`

`docker stop`, `docker rm`, `docker rmi`

Ran a simple `hello-world` container to validate installation.

Working with Docker Images

Understood the concept of Docker images as read-only templates.

Used `Dockerfile` to build custom images.

Explored common instructions in Dockerfile:

`FROM`, `RUN`, `COPY`, `ADD`, `CMD`, `EXPOSE`, `WORKDIR`

Built and tagged images using:

```
docker build -t <image_name>:<tag> .
```

Docker Containers & Volume Management

Created containers using `docker run`.

Learned about container lifecycle and container ID.

Understood the concept of Volumes:

Used for persisting data even if container is removed.

Commands: `docker volume create`, `docker volume ls`, `docker volume inspect`

Mounted volumes into containers using `-v` option.

Docker Networking

Explored Docker's default bridge network.

Created user-defined bridge networks.

Connected multiple containers for communication.

Used commands like:

`docker network create`, `docker network ls`, `docker network inspect`

Docker Compose

Introduction to Docker Compose: a tool to define and manage multi-container Docker applications.

Wrote a `docker-compose.yml` file to run a web app with a database.

Used `docker-compose up`, `docker-compose down`, and other commands.

Real-time Use Case & Best Practices

Deployed a small application using Docker Compose.

Discussed real-time scenarios:

Microservices architecture

CI/CD pipeline integration

Learned best practices:

Keep images small and clean

Use `.dockerignore`

Proper tagging and version control