Docker Notes

What is Docker?

- Docker is a platform that helps us to build, package (bundle), and deploy applications using containers.
- A Docker container is a unit that contains an application and all of its dependencies.
- - Portable and lightweight.
- - Runs in a separate isolated environment.
- To share containers, we use Docker images.

Docker Image vs Container

- A Docker Image is a blueprint or a static file that contains instructions for creating containers.
- A Docker Container is a running instance of a Docker image whereas docker image is a static snapshot how the local development environment should look like
- Analogy: Image is like a Class, and Container is like an Object.
- Docker Daemon is the core of Docker Desktop
- ls – to list directory
- Mkdir – to make directory
- Exit – to exit

Docker Commands

- docker -v — Check Docker version.
- docker — Verify if Docker is installed and see available commands.
- docker pull <image-name> — Downloads the image from Docker Hub if not present locally.
- docker images — Lists all images on the local system.
- docker run <image-name> — Creates and starts a container from the image.
- docker run -it <image-name> — Runs container in interactive mode.
- docker ps — Shows running containers.
- docker ps -a — Shows all containers, including stopped ones.
- docker start <container-name or ID> — Starts a stopped container.
- docker stop <container-name or ID> — Stops a running container.
- docker rm <container-name or ID> — Removes a container.
- docker rmi <image-name> — Removes a Docker image.
- env – to display different Environments variable

Detached Mode and Naming Containers

- docker run -d <image-name> — Runs container in detached mode.
- docker run -d --name <new-name> <image-name> — Runs a container with a custom name.

Docker Layers

- Images are built in layers:
- - Base Layer
- - Intermediate Layers
- - Container Layer (when image is run)

Port Binding

- All Docker container are binded to a default Port
- docker run -p <host-port>:<container-port> <image-name>
- Maps a host port to a container port.

Troubleshooting Commands

- docker logs <container-name or ID> — View logs of a container.
- docker exec -it <container-ID> /bin/bash — Access the terminal of a running container.

Docker vs Virtual Machine

- Docker uses the host OS kernel and is lightweight.
- Virtual Machines virtualize both the application and the OS kernel, making them heavier.
- Docker were not compatible with all OS as it was initially developed for Linux

Docker Network

- If we want to make two docker containers interact with each other we create docker network
- docker network ls — List all networks.
- docker network create <network-name> — Create a network.
- Network Drivers: bridge (default), host, none (isolated).

Docker Compose

- docker-compose -f <filename>.yaml up -d — Starts services defined in the YAML file in detached mode.

- docker-compose -f <filename>.yml down — Stops and removes the containers/services.

Dockerizing an App

- The process of preparing an application to run inside a Docker container.
- Docker push IMAGE-NAME – to push docker image to docker hub

```
FROM <base-image>

WORKDIR /app

COPY . .

RUN pip install -r requirements.txt

EXPOSE 8080

ENV MODE=production

CMD ["python", "app.py"]
```

Docker Volumes

- Used for persistent data storage.
- docker volume ls — List volumes.
- docker volume create <vol-name> — Create volume.
- docker volume rm <vol-name> — Remove volume.
- docker volume prune — Remove unused volumes.
- Mounting Volumes:
- - Named: docker run -v <vol-name>:<container-path> <image>
- - Anonymous: docker run -v <container-path> <image>
- - Host Bind Mount: docker run -v <host-path>:<container-path> <image>