To add a user in docker group and refresh group

A screen shot of a computer

AI-generated content may be incorrect.

Dockerfile----(docker build)🡪 image ------(docker run)🡪 container

Container is process in which application is running.

--- Docker run attached mode

A screenshot of a computer

AI-generated content may be incorrect.

--- Docker runs in detached mode

A screenshot of a computer

AI-generated content may be incorrect.

--- How to enter in Docker container

A screen shot of a computer

AI-generated content may be incorrect.

--- Docker login

A screenshot of a computer screen

AI-generated content may be incorrect.

--- docker rename old new

A computer screen with white text and green line

AI-generated content may be incorrect.

--- Remove Docker container



--- Remove Multiple Docker container in single

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

--- Remove Multiple Docker Images in single

A black screen with white text

AI-generated content may be incorrect.

--- Docker prune

A screenshot of a computer program

AI-generated content may be incorrect.

**--------Dockerfile---------**

FROM: - Specifies the base image to use for the container.

**FROM oprnjdk:17-jdk-alpine**

RUN: - Executes commands in the container, often used to install dependencies or perform setup tasks.

**RUN javac Main.java**

MAINTAINER: - **Author/Owner/Description**

COPY: - Copies files or directories from the local filesystem into the container.

**COPY Main.java /app**

ADD: - Like copy, but it will download files from internet

**ADD https://example.com/file.tar.gz /app/file.tar.gz**

EXPOSE: - Informs Docker that the container listens on the specified network port at runtime.

**EXPOSE 80**

WORKDIR: - Sets the working directory inside the container.

**/app**

CMD: - Specifies the command to run when the container starts.

**CMD ["java", "Main"]**

ENTRYPOINT: - Defines the command that will always run in the container, even if arguments are passed to docker run.

**ENTRYPOINT ["java", "Main"]**

===================== Project-1 Java Based=====================

--- Build a Java based App, clone code from git hub, if Docker file exist then removed and create a new Docker file to build java Application.

A screenshot of a computer program

AI-generated content may be incorrect.

Create new Docker file:- Vim Dockerfile

A screenshot of a computer

AI-generated content may be incorrect.

--- Create image from Dockerfile

A screenshot of a computer program

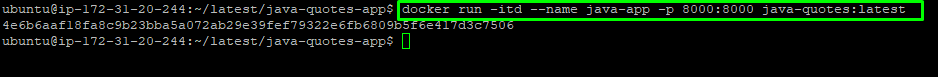
AI-generated content may be incorrect.

Show images

A screenshot of a computer program

AI-generated content may be incorrect.

Create container from image



A screenshot of a computer

AI-generated content may be incorrect.

===================== Project-2 Python Based=====================

--- Build a Python based App, clone code from git hub, if Docker file exist then removed and create a new Docker file to build Python Application.

A screenshot of a computer screen

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer screen

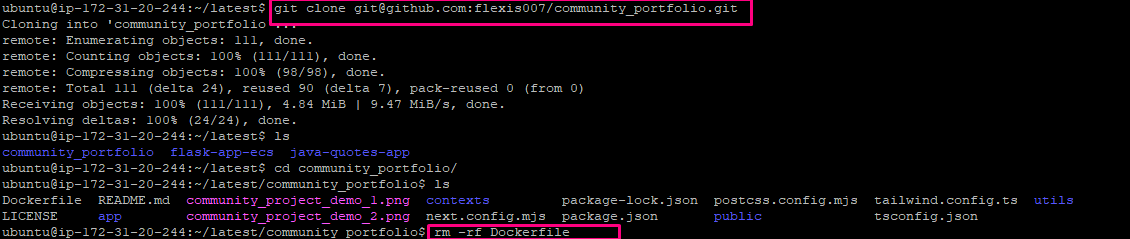
AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

===================== Project-3 Nodejs Based=====================

--- Build a Node.js App(portfolio), clone code from git hub, if Docker file exist then removed and create a new Docker file to build Nodejs Application.



A screen shot of a computer

AI-generated content may be incorrect.

\*\* Create docker image

\*\* Create docker container

**--------Multi-Stage-Dockerfile---------**

Distroless: - Distroless image contain only your application and its runtime dependencies, they do not contain package managers.

vim Docker-multi-stage

A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.



**==============Docker Image Push oh Docker hub: -====================**

A computer screen with text and a red box

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**========================Docker Volume======================**

**Named Volumes: -**

A screen shot of a computer

AI-generated content may be incorrect.

**Without Docker volume, we have created container and make a database and by mistake remove container now data lost.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A black screen with white text

AI-generated content may be incorrect.**

**Now first we create a docker volume, With Docker Volume we create a container and remove container data will persist.**

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Second Example of Volume Host Volumes (Bind Mounts): -**

A computer screen with text on it

AI-generated content may be incorrect.

**====================Docker Network==================**

Type of Tiers: -

Presentation Tier (Front end)

Logical Tier (Back end) Called Three tier App

Data base (DB tier)

A black screen with white text

AI-generated content may be incorrect.



Types of Networks: -

Bridge, None, Host, (User define bridge), {MacVlan, IP vlan and Overlay}

--- Build a two-tier flask App front end and DB.

git clone [git@github.com:flexis007/two-tier-flask-app.git](mailto:git@github.com:flexis007/two-tier-flask-app.git)

A black screen with white text

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

Create a MySQL container

docker run -d --name mysql -v mysql\_data:/var/lib/mysql --network=twotier -e MYSQL\_PASSWORD=admin -e MYSQL\_USER=admin -e MYSQL\_ROOT\_PASSWORD=admin -e MYSQL\_DATABASE=tws\_db -p 3306:3306 mysql:latest

Create a flask app container

docker run -d --name flask -p 5000:5000 --network=twotier -e MYSQL\_HOST=mysql -e MYSQL\_USER=admin -e MYSQL\_PASSWORD=admin -e MYSQL\_DB=tws\_db flask-app:latest

A screenshot of a chat

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

**====================Docker Compose==================**

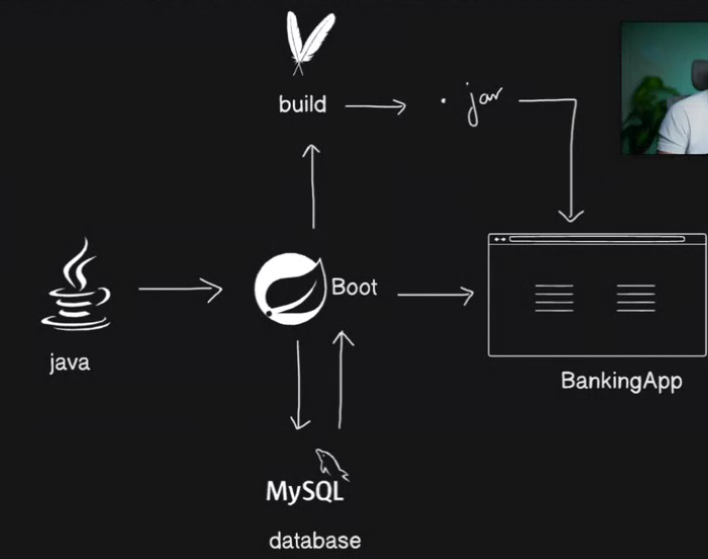
A screen shot of a computer

AI-generated content may be incorrect.

docker compose up -d

docker compose down

**============ End To End Banking App Project Java Based =============**

****