**Deploying the Django Project**

**Day 3**

Deploy.sh

# according to the Wsl Use

#!/bin/bash

<< 'comment'

Deploy a Django app and handle the code for the errors

comment

code\_clone() {

echo "Cloning the code to the computer"

git clone https://github.com/LondheShubham153/django-notes-app.git || {

echo "The code directory already exists"

cd django-notes-app

}

}

wait\_for\_lock\_release() {

echo "Waiting for apt/dpkg lock to be released"

while sudo fuser /var/lib/dpkg/lock-frontend >/dev/null 2>&1; do

echo "Waiting for lock..."

sleep 5

done

}

install\_requirements() {

echo "Installing all the requirements"

sudo apt-get update

# Remove any existing Docker packages to avoid conflicts

sudo apt-get remove -y docker docker-engine docker.io containerd runc

# Install Docker from the official repository

sudo apt-get install -y \

apt-transport-https \

ca-certificates \

curl \

software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository \

"deb [arch=amd64] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) \

stable"

sudo apt-get update

sudo apt-get install -y docker-ce docker-ce-cli containerd.io

# Install Nginx

sudo apt-get install -y nginx

}

required\_restarts() {

echo "Restarting Nginx server and Docker"

sudo chown $USER /var/run/docker.sock

sudo systemctl enable docker

sudo systemctl enable nginx

sudo systemctl restart docker

sudo systemctl restart nginx

}

deploy() {

echo "Deploying the project live"

docker build -t notes-app .

docker run -d -p 8000:8000 notes-app:latest

}

echo "\*\*\*\*\*\*\*\*\*\*\*\* Deployment Started \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

# Error handling

code\_clone

wait\_for\_lock\_release

if ! install\_requirements; then

echo "Installation failed"

exit 1

fi

if ! required\_restarts; then

echo "System failed"

exit 1

fi

deploy

**Dockerfile**

# Use an official Python runtime as a parent image

FROM python:3.9-slim

# Set the working directory in the container

WORKDIR /app/backend

# Copy the requirements file into the container

COPY requirements.txt .

# Install any needed packages specified in requirements.txt

RUN pip install --no-cache-dir -r requirements.txt

# Copy the current directory contents into the container at /app/backend

COPY . .

# Make port 8000 available to the world outside this container

EXPOSE 8000

# Run the application

CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]

Commads to run this

### Ensure Docker is Running

**Install Docker Desktop for Windows**: Docker Desktop provides integration with WSL 2. Download and install Docker Desktop from Docker's website.

**Enable WSL 2 Integration**:

* 1. Open Docker Desktop.
  2. Go to Settings > Resources > WSL Integration.
  3. Enable integration for your WSL distributions.

**Start Docker**: Docker Desktop should start automatically. Ensure it's running by checking the Docker icon in your system tray.

### 2. Open WSL Terminal

Open your WSL terminal (e.g., Ubuntu on WSL) where your deploy.sh script is located.

**To Run the project the Command Use is**

docker run -d -p 8000:8000 notes-app:latest

Then open browser and search the

<http://<WSL_IP>:8000> or open Docker Desktop and then Click on the Link 8000:8000