

1. Commands used for deploying the flask application on the ec2 instance

To deploy a Flask application on an EC2 instance using Docker, you can follow these general steps:

- a. Set up and configure your EC2 instance:
 - a. Launch an EC2 instance with your desired specifications.
 - b. Connect to the instance via SSH.
- b. Install Docker on the EC2 instance:
 - a. Update the package manager: **sudo apt update**
 - b. Install Docker: **sudo apt install docker.io**
 - c. Start and enable Docker service: **sudo systemctl start docker** and **sudo systemctl enable docker**
- c. Build the Docker image:
 - a. Create a Dockerfile in the root directory of your Flask application. An example Dockerfile may look like this:
 - b. Docker file code
 - c. # Use a base image FROM python:3.9
 - d. # Set the working directory WORKDIR /app # Copy the requirements file COPY requirements.txt .
 - e. # Install the dependencies RUN pip install --no-cache-dir -r requirements.txt
 - f. # Copy the application code COPY . .
 - g. # Expose the necessary port EXPOSE 5000
 - h. # Define the command to run the application CMD ["python", "app.py"]
- d. Build the Docker image using the Dockerfile: **sudo docker build -t flask-app .**
- e. Run the Docker container:
 - a. Start a Docker container from the image: **sudo docker run -d -p 80:5000 flask-app**
 - b. Here, we're mapping port 80 of the EC2 instance to port 5000 of the Docker container, assuming your Flask application is listening on port 5000.
- f. Verify the deployment:
 - a. Access your Flask application by entering the public IP address or DNS of your EC2 instance in a web browser, followed by the mapped port (e.g., **http://<public_ip>:80**).
 - b. These steps provide a basic outline of deploying a Flask application on an EC2 instance using Docker. However, the exact commands and setup might vary depending on your specific application and requirements. Make sure to

modify the Dockerfile and Docker run command according to your application's structure, dependencies, and desired port configuration.