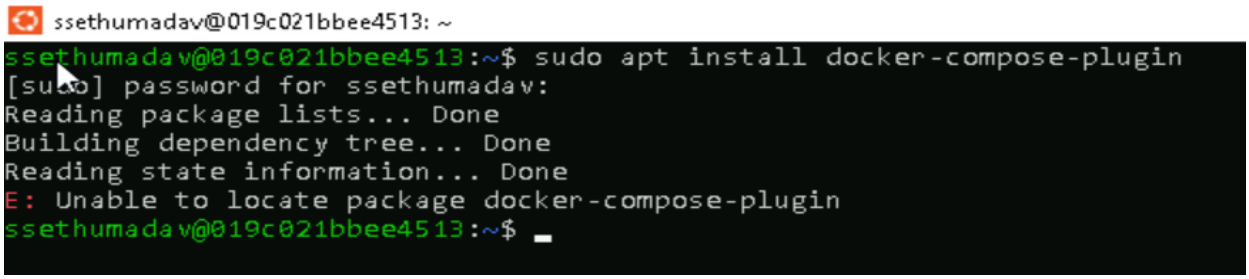
Project 2: Multi-Container Flask Application with PostgreSQL Using Docker Compose

This project demonstrates how to containerize a Flask web application and integrate it with a PostgreSQL database using Docker Compose. It simplifies the setup, ensures scalability, and automates deployment.

sudo apt install docker-compose-plugin

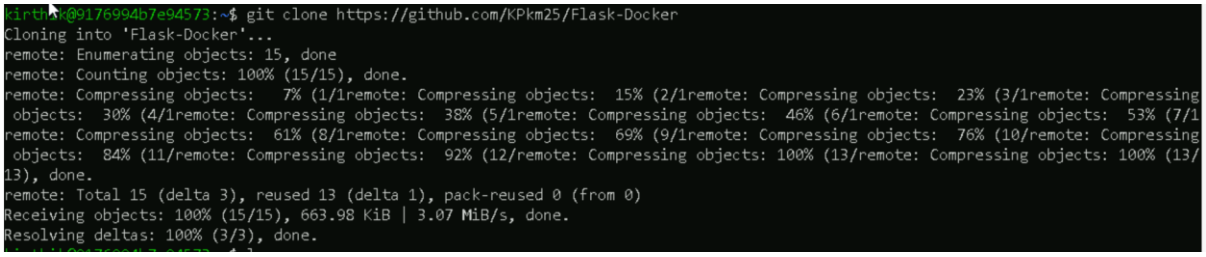


Step 1: Clone the Repository

Run the following command in your terminal:

git clone https://github.com/ssethumadav/Flask-Docker

cd Flask-Docker



Step 2: Build and Start the Containers

docker compose up -d --build

Builds the Flask application image
Starts the PostgreSQL database container

```
kirthik@9176994b7e94573:~/Flask-Docker$ docker-compose up -d --build
WARN[0000] /home/kirthik/Flask-Docker/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid
 potential confusion
[+] Running 14/14
 ⠿ db Pulled                                                    19.4s
   ⠿ bc13f9b1d80d Download complete                              0.6s
   ⠿ 55c54708c8e7 Download complete                              2.3s
   ⠿ 600e770d797e Download complete                              1.6s
   ⠿ 543c6dea2e39 Download complete                              0.4s
   ⠿ 878a40f56a67 Download complete                              2.1s
   ⠿ 42e76ffa3e07 Download complete                              1.5s
   ⠿ 553d1749e29f Download complete                              1.4s
   ⠿ 6424ae1ae883 Download complete                              1.9s
   ⠿ a21a08dbca2c Download complete                              1.7s
   ⠿ 783086ffbe8e Download complete                              8.2s
   ⠿ 420a047e4570 Download complete                              1.5s
   ⠿ fcccafd45a4d Download complete                              1.3s
   ⠿ dc87fb4dbc03 Download complete                              1.8s
[+] Building 84.1s (12/12) FINISHED                      docker:default
 => [web internal] load build definition from Dockerfile       0.1s
 => => transferring dockerfile: 189B                           0.0s
 => [web internal] load metadata for docker.io/library/python:3.9  3.5s
 => [web auth] library/python:pull token for registry-1.docker.io  0.0s
 => [web internal] load .dockerignore                          0.1s
 => => transferring context: 2B                                0.0s
 => [web 1/5] FROM docker.io/library/python:3.9@sha256:5ea663a1c6b  49.2s
 => => resolve docker.io/library/python:3.9@sha256:5ea663a1c6ba266f  0.1s
 => => sha256:521cad6ddc5302ec0b1d426cdf6df64316fd18ddf 250B / 250B  0.3s
 => => sha256:95b7226c62e1a4719940920ae7fffd1ea49 19.85MB / 19.85MB  4.6s
```

Step 3: Verify Running Containers

To check if the containers are running:

docker ps

You should see flask_app (Flask) and postgres_db (PostgreSQL) running.

```
kirthik@9176994b7e94573:~/Flask-Docker$ docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED              STATUS              PORTS                    NAMES
a640e13e37d7   flask-docker-web   "python app.py"          About a minute ago   Up About a minute   0.0.0.0:5000->5000/tcp   flask-do
cker-web-1
a6707d7c7d6d   postgres           "docker-entrypoint.s…"   About a minute ago   Up About a minute   0.0.0.0:5433->5432/tcp   flask-do
cker-db-1
kirthik@9176994b7e94573:~/Flask-Docker$ docker-compose down
```

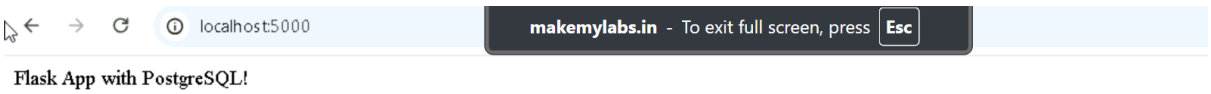
Step 4: Test the Application

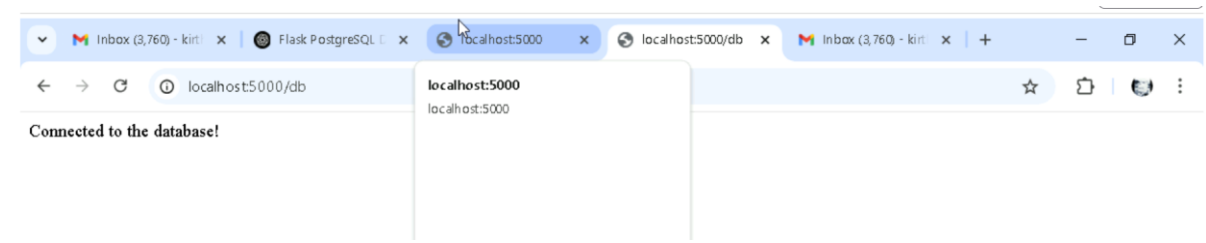
Open in Browser

- Visit http://localhost:5000/ → Expected output:

"Flask App with PostgreSQL!"

- Visit http://localhost:5000/db → Should confirm database connection.





Stopping & Cleaning Up

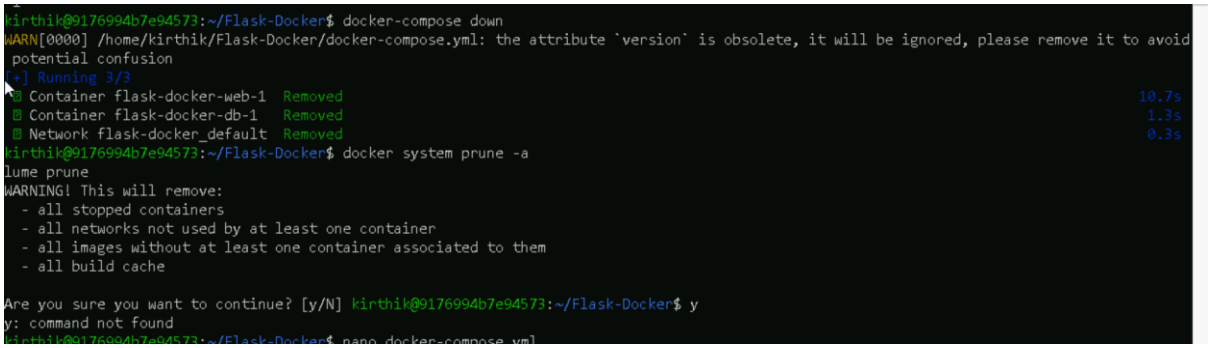
To stop the running containers:

docker compose down

To remove unused images and volumes:

docker system prune -a

docker volume prune



Troubleshooting & Common Issues

Error: "Port 5432 Already in Use"

Change PostgreSQL Port in docker-compose.yml

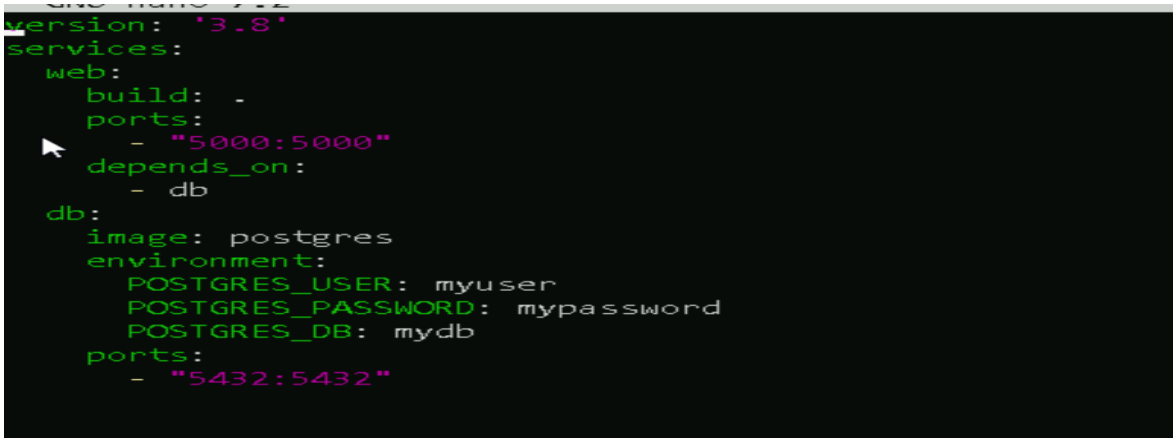
Edit this section:

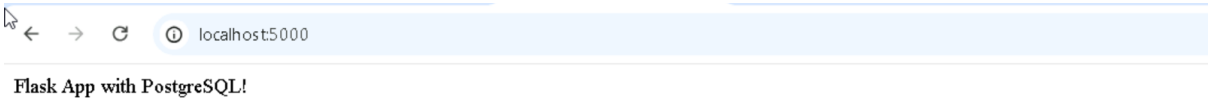
ports:

- "5433:5432"

Then restart the containers:

docker compose up -d –build





Flask App with PostgreSQL!