**📘 Multi-Container Flask Application with PostgreSQL Using Docker Compose**

**📝 Overview**

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.

📌 Prerequisites

Before running this project, ensure you have the following installed:

✅ Docker → Check installation:

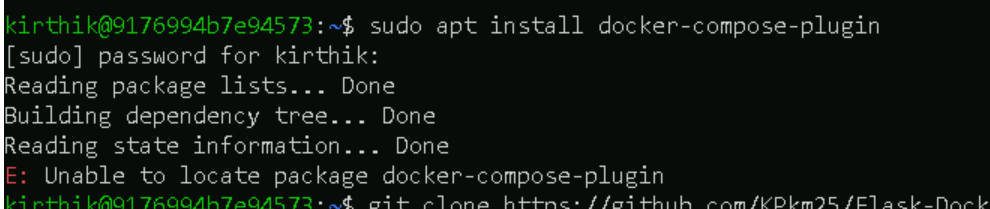
docker --version

✅ Docker Compose → Check installation:

docker-compose version

If not installed, install it manually:

sudo apt install docker-compose-plugin



📂 Project Structure

Flask-Docker/

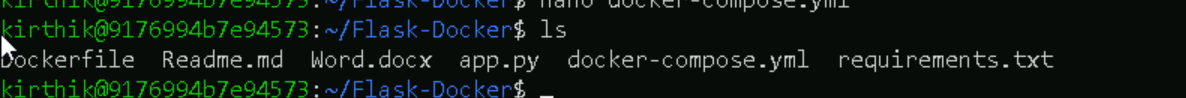
│── app.py # Flask application

│── requirements.txt # Python dependencies

│── Dockerfile # Dockerfile for Flask app

│── docker-compose.yml # Docker Compose configuration

└── README.md # Project documentation



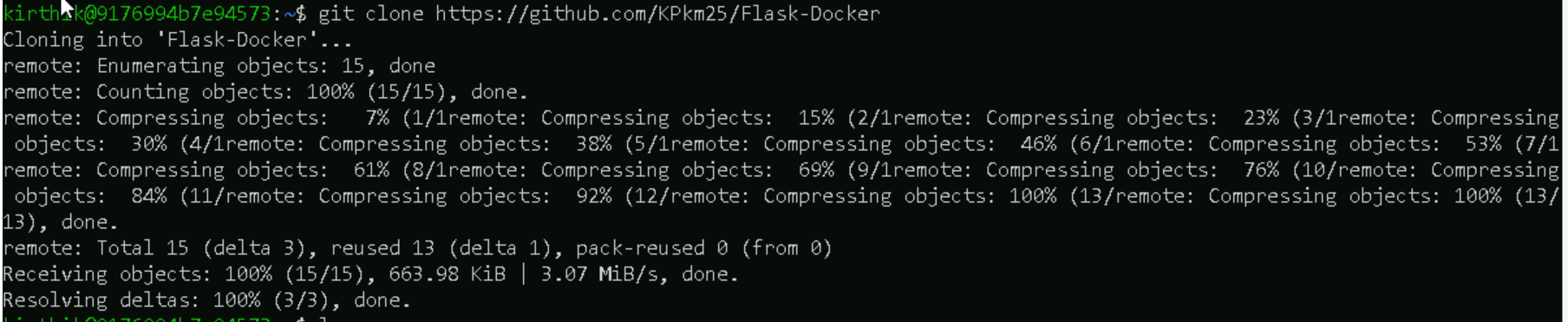
🚀 Step-by-Step Setup & Execution

📌 Step 1: Clone the Repository

Run the following command in your terminal:

git clone [https://github.com/KPkm25/Flask-Docker](https://urldefense.com/v3/__https:/github.com/KPkm25/Flask-Docker__;!!PdM5GIU!Q0_wVgrfjsqo6JTL92-HXzUJiiW8mjnsOXPsrO6pRsDyz50wCPFYSlujix3zAzar7-ZphYr5ZWPL2wW_8h33rdwD$)

cd Flask-Docker

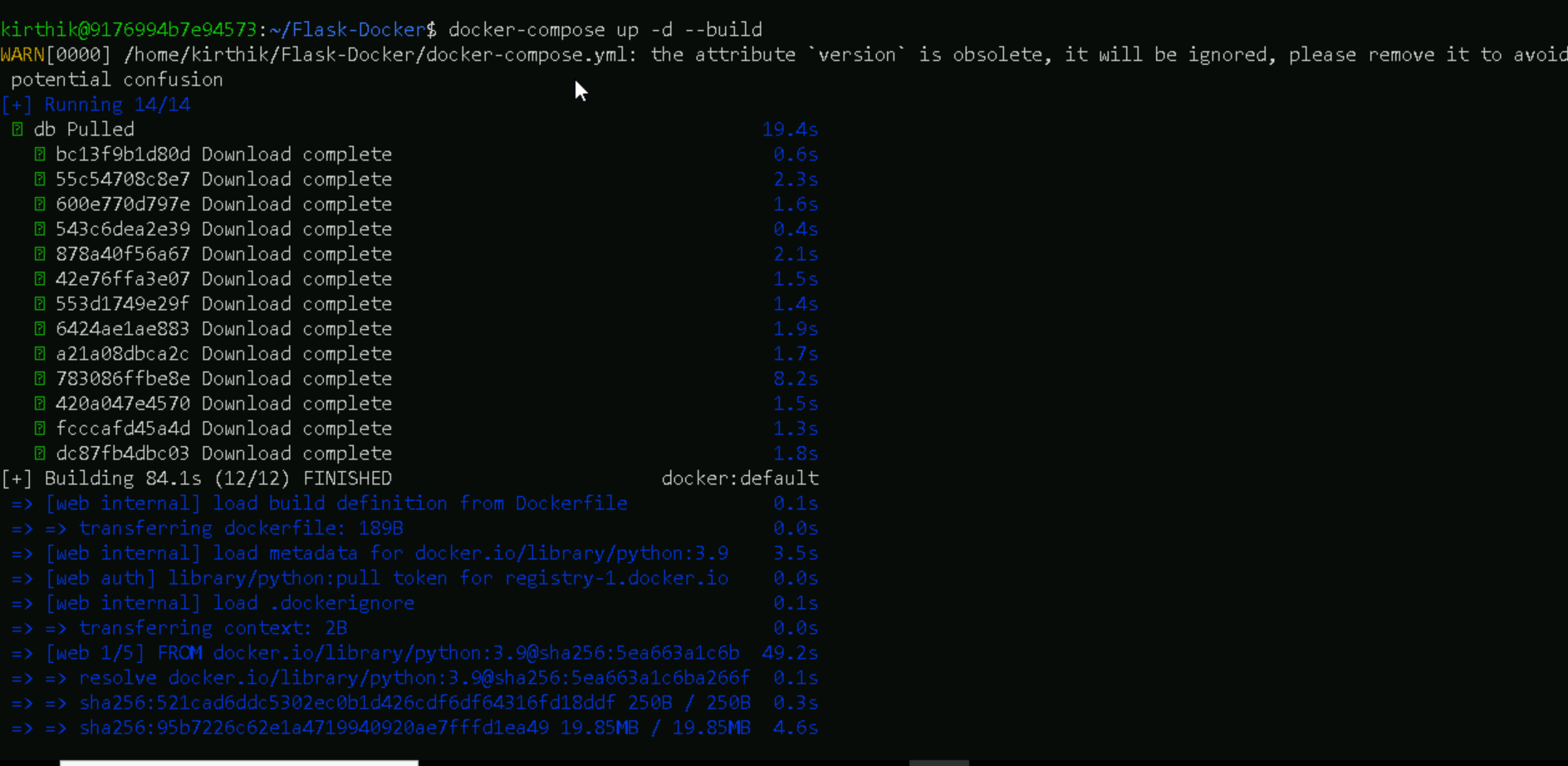


📌 Step 2: Build and Start the Containers

Run the following command:

docker compose up -d --build

✔ Builds the Flask application image  
✔ Starts the PostgreSQL database container

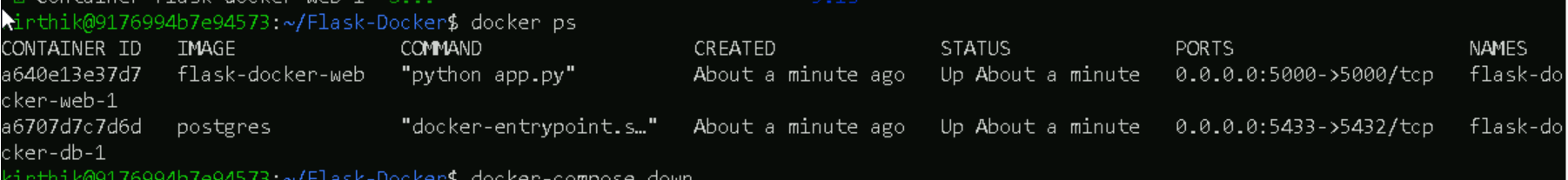


📌 Step 3: Verify Running Containers

To check if the containers are running:

docker ps

You should see flask\_app (Flask) and postgres\_db (PostgreSQL) running.



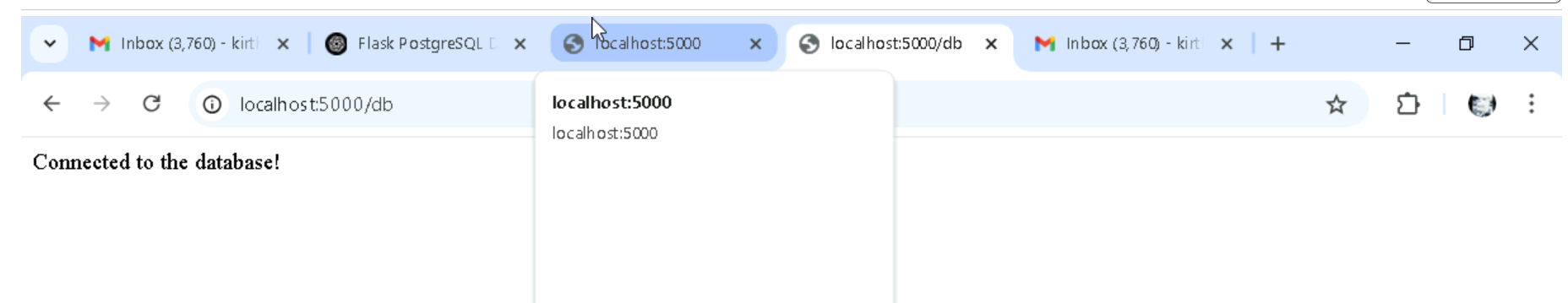
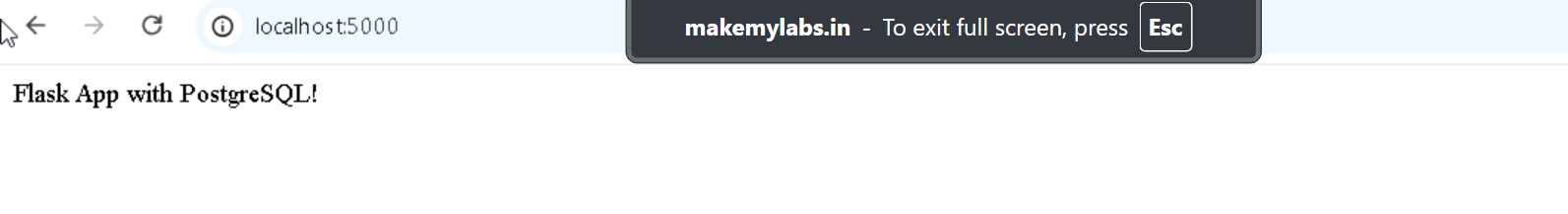
📌 Step 4: Test the Application

🔹  Open in Browser

* Visit [http://localhost:5000/](https://urldefense.com/v3/__http:/localhost:5000/__;!!PdM5GIU!Q0_wVgrfjsqo6JTL92-HXzUJiiW8mjnsOXPsrO6pRsDyz50wCPFYSlujix3zAzar7-ZphYr5ZWPL2wW_8i-GxDFv$) → Expected output:

"Flask App with PostgreSQL!"

* Visit [http://localhost:5000/db](https://urldefense.com/v3/__http:/localhost:5000/db__;!!PdM5GIU!Q0_wVgrfjsqo6JTL92-HXzUJiiW8mjnsOXPsrO6pRsDyz50wCPFYSlujix3zAzar7-ZphYr5ZWPL2wW_8pTVkPRW$) → Should confirm database connection.



🛑 Stopping & Cleaning Up

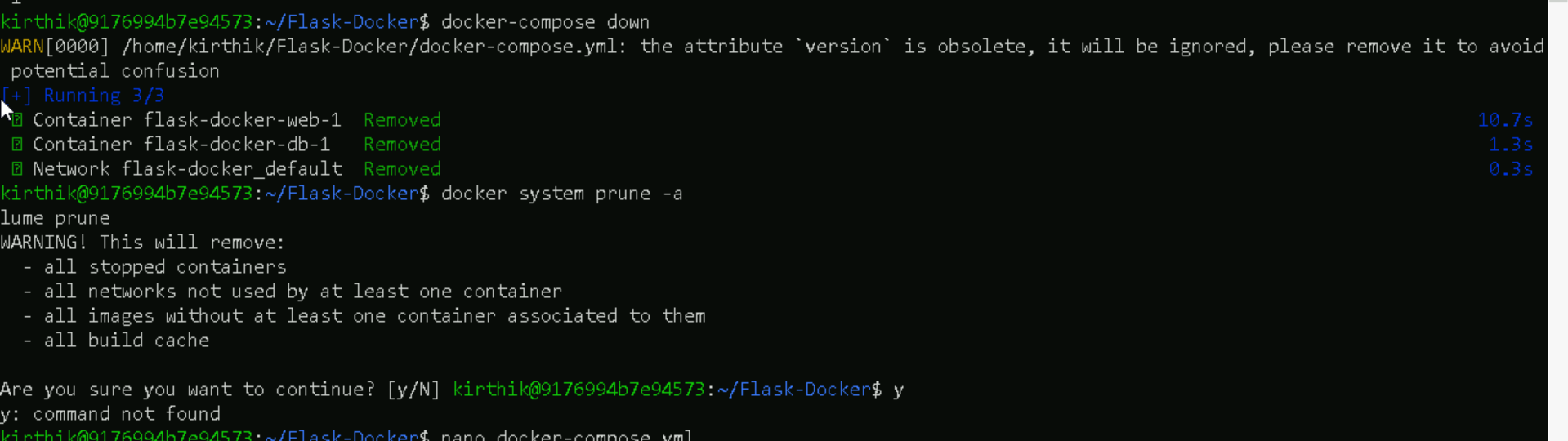
To stop the running containers:

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

