Let's recreate the setup with all the necessary information, including specifying the database name in the Docker Compose YAML file and ensuring the SQLite database is mounted correctly in the volume. Here's a step-by-step guide that includes these details.

Step-by-Step Guide

1. Streamlit App for Visualizing Student Performance

Step 1: Install Streamlit

Ensure you have Python installed and install Streamlit using pip:

```
pip install streamlit
```

Step 2: Create a New Directory for Your Project

Create a new folder for your Streamlit app:

```
mkdir student_performance_app
cd student_performance_app
```

Step 3: Create a Python Script for the Streamlit App

Create a new Python file named `app.py` in the project directory:

```python
# app.py

import streamlit as st
import pandas as pd
import sqlite3

# Function to load data from SQLite database
def load_data():
    conn = sqlite3.connect('students.db')  # Database name is specified here
    query = "SELECT * FROM students"
    df = pd.read_sql(query, conn)
    conn.close()
    return df

# Streamlit app layout
st.title('Student Performance Visualization')

# Load data
data = load_data()

# Sidebar for filtering
st.sidebar.header('Filter Options')
```

```
selected_grade = st.sidebar.selectbox('Select Grade', data['grade'].unique())

# Filter data based on selected grade
filtered_data = data[data['grade'] == selected_grade]

# Display filtered data
st.write(filtered_data)

# Create visualizations
st.subheader('Performance Metrics')
st.bar_chart(filtered_data['performance_score'])
```

Step 4: Run the Streamlit App

You can run the Streamlit app locally for testing:

```
streamlit run app.py
```

2. Create SQLite Database with Data for 1200 Students

Step 1: Install Required Libraries

If you haven't already, you need the `faker` library to generate dummy data:

```
pip install faker
```

Step 2: Create a Python Script to Generate Dummy Data

Create a new Python file named `generate_data.py` in your project directory:

```
# generate_data.py

import sqlite3
from faker import Faker
import random

# Create a SQLite database
conn = sqlite3.connect('students.db')  # Database name is specified here
cursor = conn.cursor()

# Create a table for students
cursor.execute('''
CREATE TABLE IF NOT EXISTS students (
    id INTEGER PRIMARY KEY,
    name TEXT,
    grade TEXT,
    performance_score INTEGER
)
''')

# Generate dummy data
fake = Faker()
grades = ['A', 'B', 'C', 'D', 'F']
for _ in range(1200):
    name = fake.name()
    grade = random.choice(grades)
    performance_score = random.randint(0, 100)
    cursor.execute('''
    INSERT INTO students (name, grade, performance_score) VALUES (?, ?, ?)
    ''', (name, grade, performance_score))

# Commit changes and close the connection
conn.commit()
conn.close()
```

Step 3: Run the Data Generation Script

Execute the script to create the SQLite database and populate it with data:

```
python generate_data.py
```

3. Create a Docker Configuration

Step 1: Create a Dockerfile

Create a file named `Dockerfile` in your project directory:

```
# Dockerfile

# Use the official Python image
FROM python:3.9

# Set the working directory
WORKDIR /app

# Copy the requirements file and install dependencies
COPY requirements.txt ./
RUN pip install --no-cache-dir -r requirements.txt

# Copy the application code
COPY . .

# Command to run the Streamlit app
CMD ["streamlit", "run", "app.py", "--server.port=8501", "--server.address=0.0.0.0"]
```

Step 2: Create a Requirements File

Create a `requirements.txt` file with the necessary dependencies:

```
streamlit
pandas
faker
```

Step 3: Create a Docker Compose File

Create a file named `docker-compose.yml` in your project directory with the following content:

```yaml
version: '3.8'

services:
  streamlit-app:
    build: .
    ports:
      - "8501:8501"
    volumes:
      - .:/app  # Mount the current directory to /app in the container
      - students_data:/app/students.db  # Mount SQLite database to persist data
    environment:
      - PYTHONUNBUFFERED=1

volumes:
  students_data:  # Named volume for SQLite database
```

Running Your Application with Docker

Step 1: Build and Run the Docker Container

Navigate to your project directory and run the following command to build and run your Docker container:

```
docker-compose up --build
```

Step 2: Access Your Streamlit App

Open your web browser and go to `http://localhost:8501` to see your Streamlit app running in a Docker container.

Summary

Now, you have:

- A Streamlit app to visualize student performance data.
- An SQLite database named `students.db` with dummy data for 1200 students.

- A Docker setup that specifies the database name and mounts the SQLite database in the volume for persistence.

This setup allows you to visualize student performance and ensures that your database is properly managed and persistent across container restarts. You can enhance your app further by adding more visualizations or features as needed.