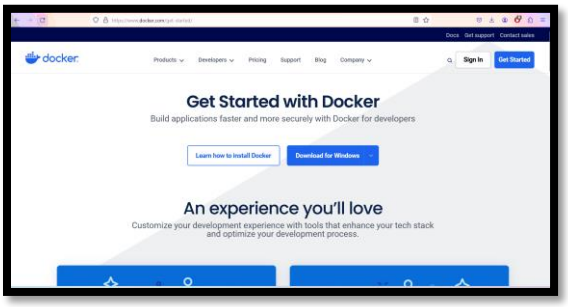
Niyat Habtom Seghid

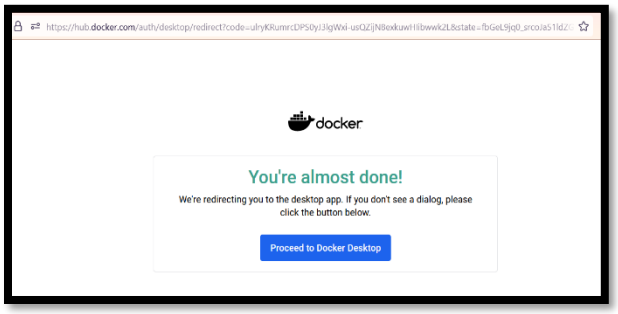
CS571 – Cloud computing Infrastructure

GenAI - Develop your containerized app

Step 1: GenAI - Containerize your app.

1. First, install the latest version of Docker Desktop for windows.





2. Go to terminal and navigate to our working directory.

```
C:\Users\Niyat Habtom Seghid>mkdir GenAIApplication

C:\Users\Niyat Habtom Seghid>cd GenAIApplication
```

3. Clone the sample application. We run the following command to clone the repository:

   git clone https://github.com/craig-osterhout/docker-genai-sample

```
C:\Users\Niyat Habtom Seghid\GenAIApplication>git clone https://github.com/craig-osterhout/docker-genai-sample
Cloning into 'docker-genai-sample'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 11 (delta 0), reused 11 (delta 0), pack-reused 0
Receiving objects: 100% (11/11), 10.17 KiB | 2.54 MiB/s, done.

C:\Users\Niyat Habtom Seghid\GenAIApplication>cd docker-genai-sample
```

- You should now have the following files in your docker-genai-sample directory.

```
PS C:\Users\Niyat Habtom Seghid\GenAIApplication\docker-genai-sample> ls


    Directory: C:\Users\Niyat Habtom Seghid\GenAIApplication\docker-genai-sample


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----          4/4/2024   8:50 AM            625 .dockerignore
-a----          4/4/2024   8:49 AM           3895 app.py
-a----          4/4/2024   8:49 AM           9099 chains.py
-a----          4/4/2024   8:50 AM           1642 compose.yaml
-a----          4/4/2024   8:50 AM           1667 Dockerfile
-a----          4/4/2024   8:49 AM            967 env.example
-a----          4/4/2024   8:49 AM           7169 LICENSE
-a----          4/4/2024   8:50 AM            826 README.Docker.md
-a----          4/4/2024   8:49 AM            179 README.md
-a----          4/4/2024   8:49 AM            106 requirements.txt
-a----          4/4/2024   8:49 AM           1945 utils.py
```

4. Now that we have an application, we can use docker init to create the necessary Docker assets to containerize our application. Inside the docker-genai-sample directory, run the **docker init** command.

```
PS C:\Users\Niyat Habtom Seghid\GenAIApplication\docker-genai-sample>  docker init

Welcome to the Docker Init CLI!

This utility will walk you through creating the following files with sensible defaults for y
our project:
  - .dockerignore
  - Dockerfile
  - compose.yaml
  - README.Docker.md

Let's get started!

? What application platform does your project use? Python
? What version of Python do you want to use? 3.11.7

? What version of Python do you want to use? 3.11.7
? What port do you want your app to listen on? (8000) 8000

? What port do you want your app to listen on? 8000
? What is the command you use to run your app (e.g., gunicorn 'myapp.example:app' --bind=0.0
.0.0:8000)? streamlit run app.py --server.address=0.0.0.0 --server.port=8000
? What is the command you use to run your app (e.g., gunicorn 'myapp.example:app' --bind=0.0
.0.0:8000)? streamlit run app.py --server.address=0.0.0.0 --server.port=8000

CREATED: .dockerignore
CREATED: Dockerfile
CREATED: compose.yaml
CREATED: README.Docker.md

✔Your Docker files are ready!

Take a moment to review them and tailor them to your application.

When you're ready, start your application by running: docker compose up --build

Your application will be available at http://localhost:8000

Consult README.Docker.md for more information about using the generated files.
PS C:\Users\Niyat Habtom Seghid\GenAIApplication\docker-genai-sample>
```

Niyat Habtom Seghid

Step 2: GenAI - Develop your app.

Adding a Local Database

Here we will update the compose.yaml file to define a database service, and we will specify an environment variables file to load the database connection information rather than manually entering the information every time. To run the database service:

1. In the cloned repository's directory, rename **env.example** file to **.env**. This file contains the environment variables that the containers will use.

```
    Directory: C:\Users\Niyat Habtom Seghid\GenAIApplication\docker-genai-sample


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----          4/4/2024   8:50 AM            625 .dockerignore
-a----          4/4/2024   8:49 AM           3895 app.py
-a----          4/4/2024   8:49 AM           9099 chains.py
-a----          4/4/2024   8:50 AM           1642 compose.yaml
-a----          4/4/2024   8:50 AM           1667 Dockerfile
-a----          4/4/2024   8:49 AM            967 env.example
-a----          4/4/2024   8:49 AM           7169 LICENSE
-a----          4/4/2024   8:50 AM            826 README.Docker.md
-a----          4/4/2024   8:49 AM            179 README.md
-a----          4/4/2024   8:49 AM            106 requirements.txt
-a----          4/4/2024   8:49 AM           1945 utils.py


PS C:\Users\Niyat Habtom Seghid\GenAIApplication\docker-genai-sample> Rename-Item -Path "env.example" -NewName ".env"
PS C:\Users\Niyat Habtom Seghid\GenAIApplication\docker-genai-sample> ls


    Directory: C:\Users\Niyat Habtom Seghid\GenAIApplication\docker-genai-sample


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----          4/4/2024   8:50 AM            625 .dockerignore
-a----          4/4/2024   8:49 AM            967 .env
-a----          4/4/2024   8:49 AM           3895 app.py
-a----          4/4/2024   8:49 AM           9099 chains.py
-a----          4/4/2024   8:50 AM           1642 compose.yaml
```

2. Then open the compose.yaml file in an IDE or text editor.

```
PS C:\Users\Niyat Habtom Seghid\GenAIApplication\docker-genai-sample> vim compose.yaml
```

3. In the compose.yaml file, add the following:
   - Add instructions to run a Neo4j database
   - Specify the environment file under the server service in order to pass in the environment variables for the connection.

```
services:
  server:
    build:
      context: .
    ports:
      - 8000:8000
    env_file:
```

```
      - .env
    depends_on:
      database:
        condition: service_healthy
  database:
    image: neo4j:5.11
    ports:
      - "7474:7474"
      - "7687:7687"
    environment:
      - NEO4J_AUTH=${NEO4J_USERNAME}/${NEO4J_PASSWORD}
    healthcheck:
      test: ["CMD-SHELL", "wget --no-verbose --tries=1 --spider
localhost:7474 || exit 1"]
      interval: 5s
      timeout: 3s
      retries: 5
```

```
compose.yaml + (~\GenAIAp|
version: "3.8"

services:
  server:
    build:
      context: .
    ports:
      - "8000:8000"
    env_file:
      - .env
    depends_on:
      database:
        condition: service_healthy

  database:
    image: neo4j:5.11
    ports:
      - "7474:7474"
      - "7687:7687"
    environment:
      NEO4J_AUTH: ${NEO4J_USERNAME}/${NEO4J_PASSWORD}
    healthcheck:
      test: ["CMD-SHELL", "wget --no-verbose --tries=1 --spider localhost:7474 || exit 1"]
      interval: 5s
      timeout: 3s
      retries: 5
~
```

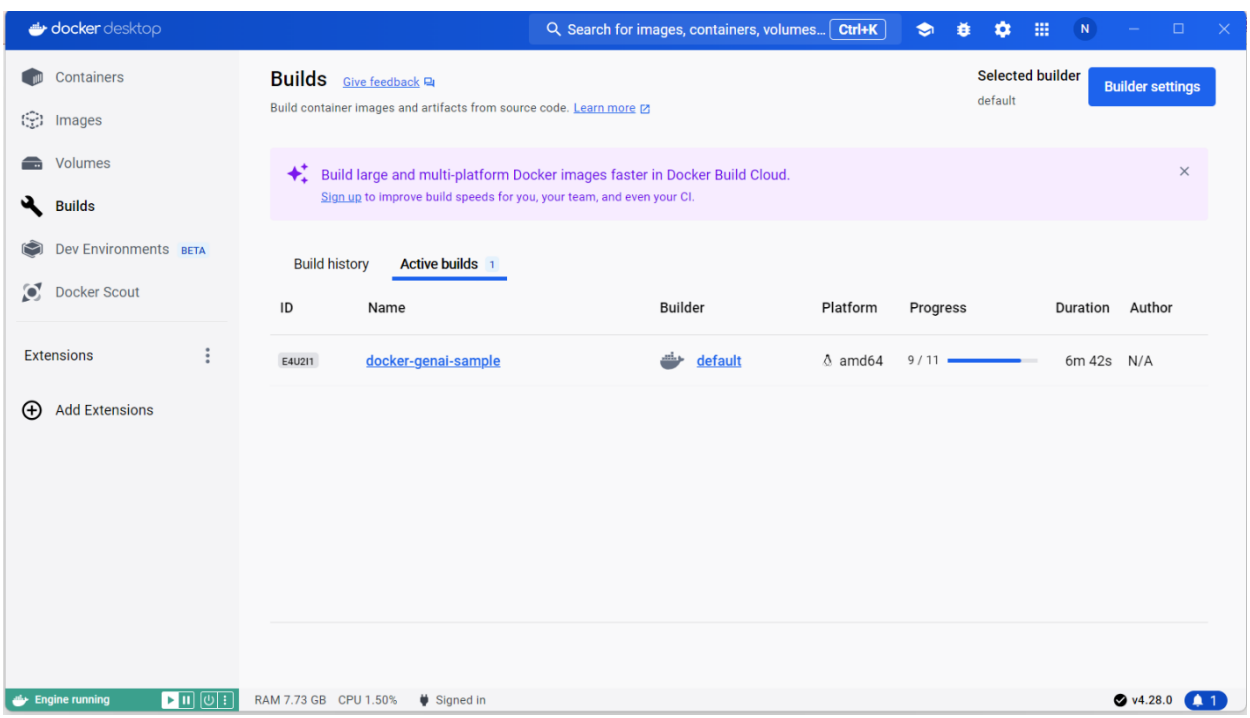
4. Run the application. Inside the docker-genai-sample directory, run the following command in a terminal.

   $ docker compose up --build

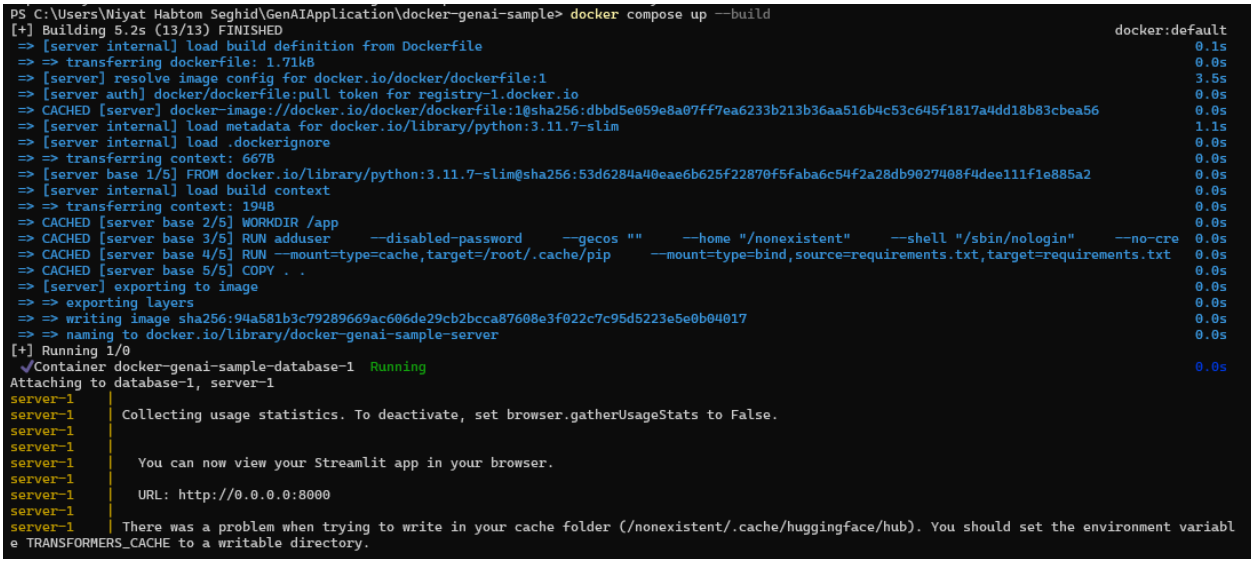
Niyat Habtom Seghid

```
PS C:\Users\Niyat Habtom Seghid\GenAIApplication\docker-genai-sample> docker compose up --build
[+] Running 6/6
 ✔database 5 layers [⣿⣿⣿⣿⣿]      0B/0B      Pulled                                          245.6s
   ✔7d97e254a046 Pull complete                                                               48.6s
   ✔732d09690fed Pull complete                                                              230.3s
   ✔e8cba66f5b65 Pull complete                                                                2.0s
   ✔9e41d761a8cf Pull complete                                                                9.8s
   ✔33a66ada74dc Pull complete                                                              134.2s
[+] Building 40.7s (14/14) FINISHED                                              docker:default
 => [server internal] load build definition from Dockerfile                                   0.1s
 => => transferring dockerfile: 1.71kB                                                        0.0s
 => [server] resolve image config for docker.io/docker/dockerfile:1                          11.3s
 => [server auth] docker/dockerfile:pull token for registry-1.docker.io                       0.0s
 => [server] docker-image://docker.io/docker/dockerfile:1@sha256:dbbd5e059e8a07ff7ea6233b2  9.7s
 => => resolve docker.io/docker/dockerfile:1@sha256:dbbd5e059e8a07ff7ea6233b213b36aa516b4c  0.4s
 => => sha256:dbbd5e059e8a07ff7ea6233b213b36aa516b4c53c645f1817a4dd18b83cb 8.40kB / 8.40kB  0.0s
 => => sha256:4611ea7b7d89ce41ec5c63df83076ccec3fe8daa32a2d9c96e5decb72e9a8d67 482B / 482B  0.0s
 => => sha256:ab56f6885c985024a40925d2fa322df997655db5f361ad9f221861f9c665 1.26kB / 1.26kB  0.0s
 => => sha256:ccef65a67ab38a038c615e74c797b11a43d36505710abe93c87b021401 11.98MB / 11.98MB  8.7s
 => => extracting sha256:ccef65a67ab38a038c615e74c797b11a43d36505710abe93c87b021401bebd81   0.5s
 => [server internal] load metadata for docker.io/library/python:3.11.7-slim                 4.0s
 => [server auth] library/python:pull token for registry-1.docker.io                          0.0s
 => [server internal] load .dockerignore                                                      0.2s
 => => transferring context: 667B                                                             0.1s
 => [server base 1/5] FROM docker.io/library/python:3.11.7-slim@sha256:53d6284a40eae6b625f  0.0s
 => [server internal] load build context                                                      0.3s
 => => transferring context: 115.79kB                                                         0.1s
 => CACHED [server base 2/5] WORKDIR /app                                                     0.0s
 => CACHED [server base 3/5] RUN adduser     --disabled-password     --gecos ""     --home  0.0s
 => CACHED [server base 4/5] RUN --mount=type=cache,target=/root/.cache/pip     --mount=ty  0.0s
 => [server base 5/5] COPY . .                                                                0.2s
 => [server] exporting to image                                                               0.1s
 => => exporting layers                                                                       0.1s
 => => writing image sha256:94a581b3c79289669ac606de29cb2bcca87608e3f022c7c95d5223e5e0b040  0.0s
 => => naming to docker.io/library/docker-genai-sample-server                                 0.0s
[+] Running 2/2
 ✔Container docker-genai-sample-database-1  Created                                           0.4s
 ✔Container docker-genai-sample-server-1    Re...                                             0.6s
Attaching to database-1, server-1
database-1  | Changed password for user 'neo4j'. IMPORTANT: this change will only take effect if performed before the database is star
ime.
database-1  | 2024-04-15 07:10:08.876+0000 INFO  Starting...
database-1  | 2024-04-15 07:10:11.010+0000 INFO  This instance is ServerId{fe6d4df3} (fe6d4df3-6fbf-4c55-99bc-9f804e7f70f6)
```

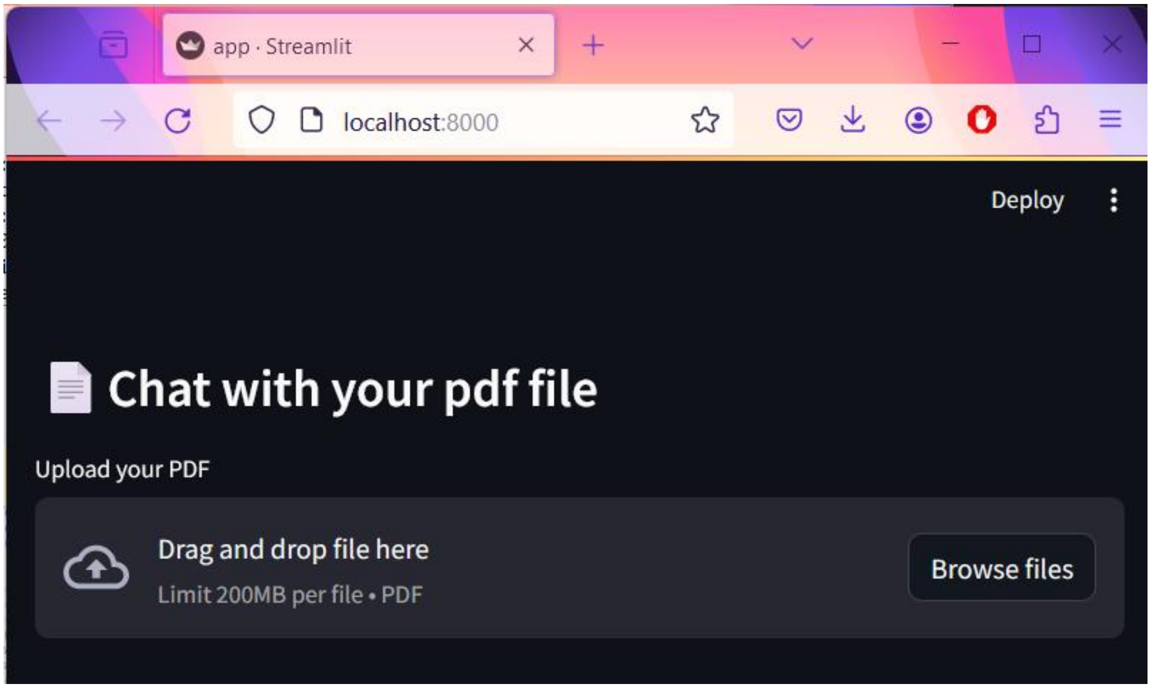
- We can also see the progress from Docker Desktop.



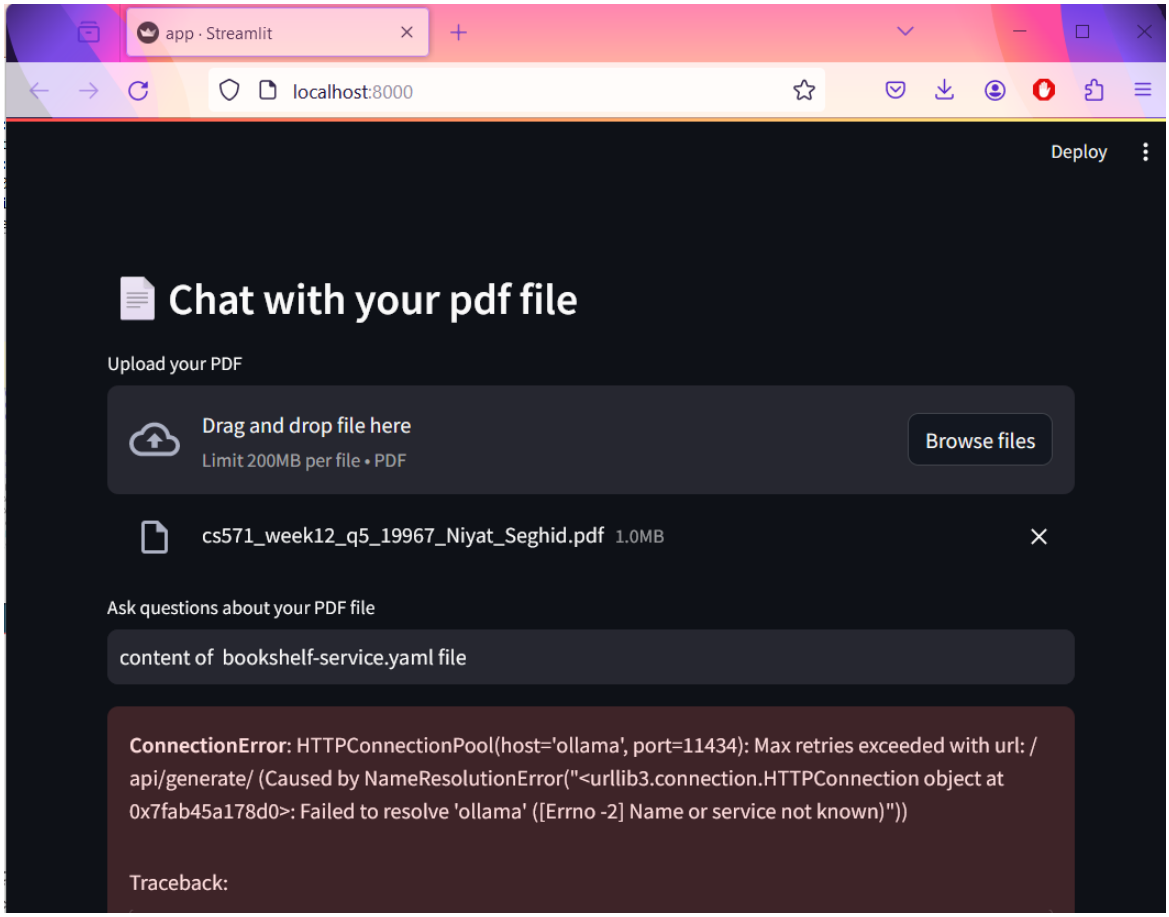
5. Access the application. Open a browser and view the application at http://localhost:8000. You should see a simple Streamlit application.



```
PS C:\Users\Niyat Habtom Seghid\GenAIApplication\docker-genai-sample> docker compose up --build
[+] Building 5.2s (13/13) FINISHED                                                                          docker:default
 => [server internal] load build definition from Dockerfile                                                          0.1s
 => => transferring dockerfile: 1.71kB                                                                               0.0s
 => [server] resolve image config for docker.io/docker/dockerfile:1                                                  3.5s
 => [server auth] docker/dockerfile:pull token for registry-1.docker.io                                              0.0s
 => CACHED [server] docker-image://docker.io/docker/dockerfile:1@sha256:dbbd5e059e8a07ff7ea6233b213b36aa516b4c53c645f1817a4dd18b83cbea56  0.0s
 => [server internal] load metadata for docker.io/library/python:3.11.7-slim                                         1.1s
 => [server internal] load .dockerignore                                                                             0.0s
 => => transferring context: 667B                                                                                    0.0s
 => [server base 1/5] FROM docker.io/library/python:3.11.7-slim@sha256:53d6284a40eae6b625f22870f5faba6c54f2a28db9027408f4dee111f1e885a2  0.0s
 => [server internal] load build context                                                                             0.0s
 => => transferring context: 194B                                                                                    0.0s
 => CACHED [server base 2/5] WORKDIR /app                                                                            0.0s
 => CACHED [server base 3/5] RUN adduser     --disabled-password     --gecos ""     --home "/nonexistent"     --shell "/sbin/nologin"     --no-cre  0.0s
 => CACHED [server base 4/5] RUN --mount=type=cache,target=/root/.cache/pip     --mount=type=bind,source=requirements.txt,target=requirements.txt  0.0s
 => CACHED [server base 5/5] COPY . .                                                                                0.0s
 => [server] exporting to image                                                                                      0.0s
 => => exporting layers                                                                                              0.0s
 => => writing image sha256:94a581b3c79289669ac606de29cb2bcca87608e3f022c7c95d5223e5e0b04017                         0.0s
 => => naming to docker.io/library/docker-genai-sample-server                                                        0.0s
[+] Running 1/0
 ✔Container docker-genai-sample-database-1  Running                                                                  0.0s
Attaching to database-1, server-1
server-1     |
server-1     | Collecting usage statistics. To deactivate, set browser.gatherUsageStats to False.
server-1     |
server-1     |
server-1     |   You can now view your Streamlit app in your browser.
server-1     |
server-1     |   URL: http://0.0.0.0:8000
server-1     |
server-1     | There was a problem when trying to write in your cache folder (/nonexistent/.cache/huggingface/hub). You should set the environment variabl
e TRANSFORMERS_CACHE to a writable directory.
```



- Note that asking questions to a PDF will cause the application to fail because the LLM service specified in the .env file isn't running yet.



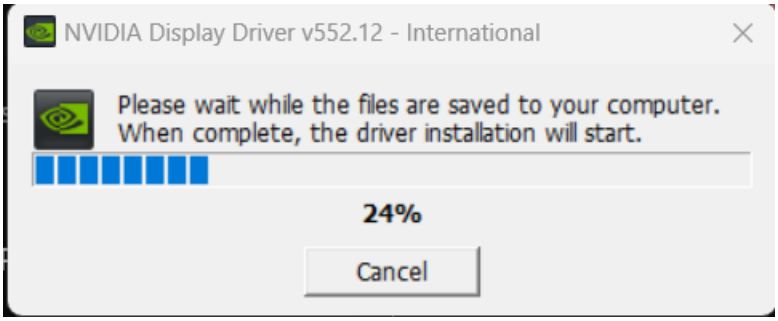
6. Stop the application. In the terminal, press ctrl+c to stop the application.

```
Gracefully stopping... (press Ctrl+C again to force)
[+] Stopping 0/1
 - Container docker-genai-sample-server-1  Stopping                [+] Stopping 0/1
[+] Stopping 2/2ker-genai-sample-server-1  Stopping          6.0s
 ✔Container docker-genai-sample-server-1    Stopped
 ✔Container docker-genai-sample-database-1  Stopped
canceled
PS C:\Users\Niyat Habtom Seghid\GenAIApplication\docker-genai-sample>
```

Adding a Local or Remote LLM Service

1. Install the prerequisites.

   - For Docker Engine on Linux, install the NVIDIA Container Toolkilt.
   - For Docker Desktop on Windows 10/11, install the latest NVIDIA driver and make sure you are using the WSL2 backend



2. Add the Ollama service and a volume in your compose.yaml. The following is the updated compose.yaml:

```
services:
  server:
    build:
      context: .
    ports:
      - 8000:8000
    env_file:
      - .env
    depends_on:
      database:
        condition: service_healthy
  database:
    image: neo4j:5.11
    ports:
      - "7474:7474"
      - "7687:7687"
    environment:
      - NEO4J_AUTH=${NEO4J_USERNAME}/${NEO4J_PASSWORD}
    healthcheck:
      test: ["CMD-SHELL", "wget --no-verbose --tries=1 --spider
localhost:7474 || exit 1"]
      interval: 5s
      timeout: 3s
      retries: 5
  ollama:
    image: ollama/ollama:latest
    ports:
      - "11434:11434"
    volumes:
      - ollama_volume:/root/.ollama
    deploy:
      resources:
        reservations:
          devices:
            - driver: nvidia
              count: all
              capabilities: [gpu]
volumes:
  ollama_volume:
```

compose.yaml + (~\GenAIApp

```yaml
version: "3.8"

services:
  server:
    build:
      context: .
    ports:
      - "8000:8000"
    env_file:
      - .env
    depends_on:
      database:
        condition: service_healthy

  database:
    image: neo4j:5.11
    ports:
      - "7474:7474"
      - "7687:7687"
    environment:
      NEO4J_AUTH: ${NEO4J_USERNAME}/${NEO4J_PASSWORD}
    healthcheck:
      test: ["CMD-SHELL", "wget --no-verbose --tries=1 --spider localhost:7474 || exit 1"]
      interval: 5s
      timeout: 3s
      retries: 5

  ollama:
    image: ollama/ollama:latest
    ports:
      - "11434:11434"
    volumes:
      - ollama_volume:/root/.ollama
    deploy:
      resources:
        reservations:
          devices:
            - driver: nvidia
              count: all
              capabilities: [gpu]
  volumes:
    ollama_volume:
```

3. Add the ollama-pull service to your compose.yaml file. This service uses the docker/genai:ollama-pull image, based on the GenAI Stack's pull_model.Dockerfile and will automatically pull the model for your Ollama container. The following is the updated section of the compose.yaml file:

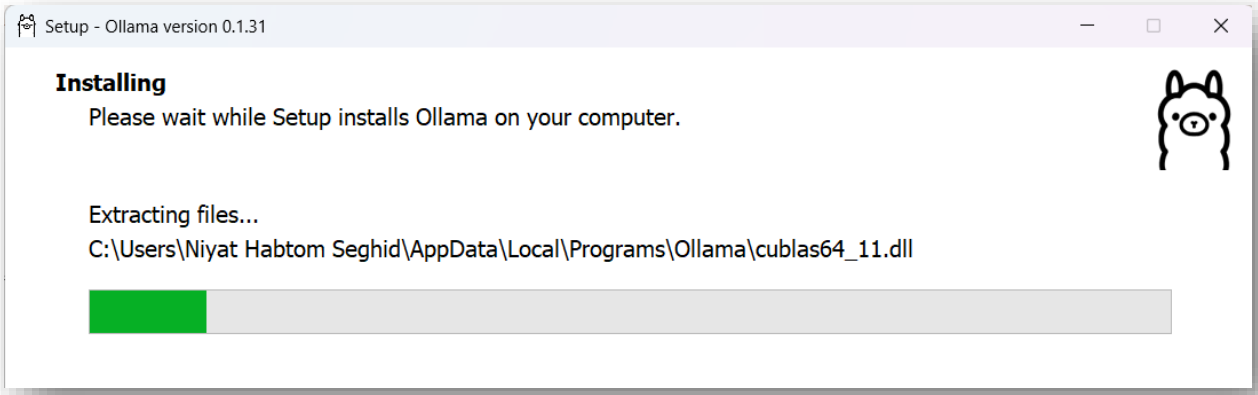
Niyat Habtom Seghid

compose.yaml + (~\GenAIApp

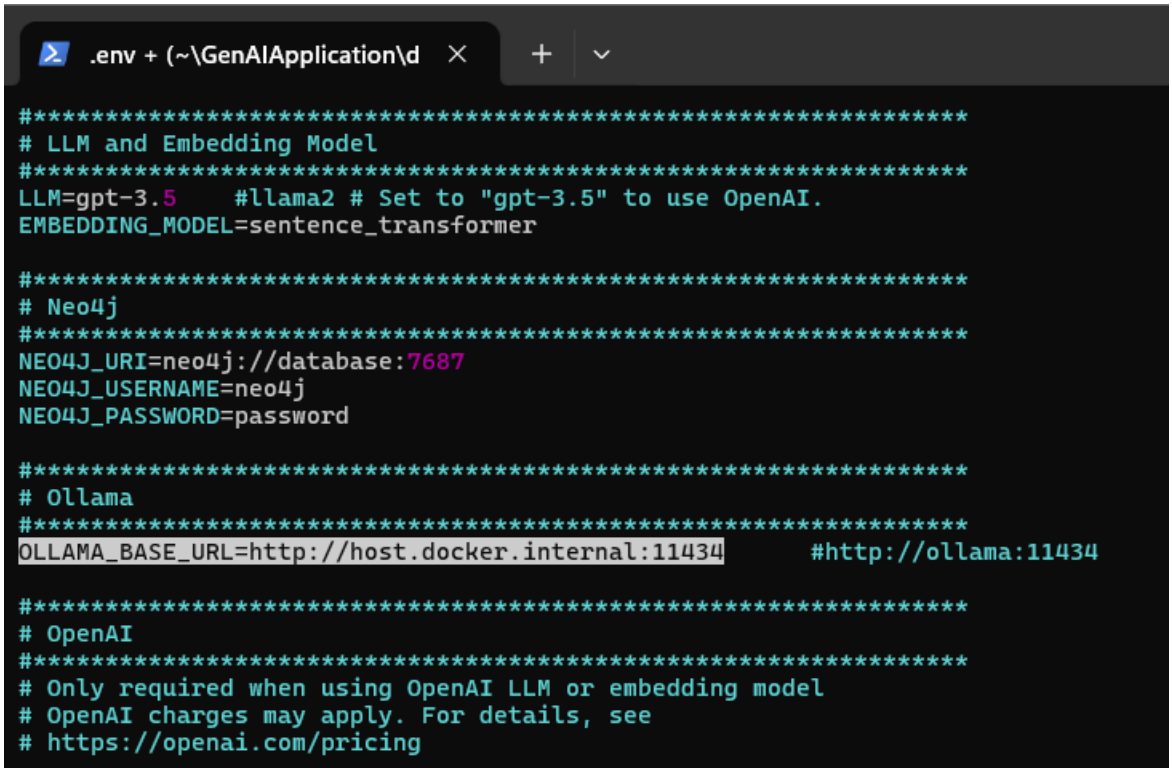
```yaml
version: "3.8"

services:
  server:
    build:
      context: .
    ports:
      - "8000:8000"
    env_file:
      - .env
    depends_on:
      database:
        condition: service_healthy
      ollama-pull:
        condition: service_completed_successfully
  ollama-pull:
    image: docker/genai:ollama-pull
    env_file:
      - .env

  database:
    image: neo4j:5.11
    ports:
      - "7474:7474"
      - "7687:7687"
    environment:
      NEO4J_AUTH: ${NEO4J_USERNAME}/${NEO4J_PASSWORD}
    healthcheck:
      test: ["CMD-SHELL", "wget --no-verbose --tries=1 --spider localhost:7474 || exit 1"]
      interval: 5s
      timeout: 3s
      retries: 5

  ollama:
    image: ollama/ollama:latest
    ports:
      - "11434:11434"
    volumes:
      - ollama_volume:/root/.ollama
    deploy:
      resources:
        reservations:
          devices:
            - driver: nvidia
              count: all
              capabilities: [gpu]
  volumes:
    ollama_volume:
```

Run Ollama in a Container

1. Install and run Ollama on your host machine.

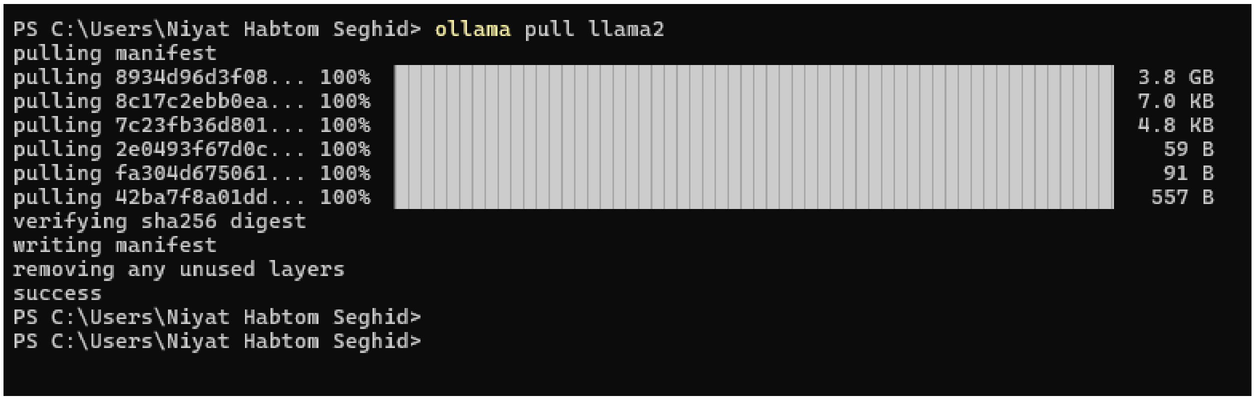


2. Update the OLLAMA_BASE_URL value in your .env file to http://host.docker.internal:11434.



3. Pull the model to Ollama using the following command.

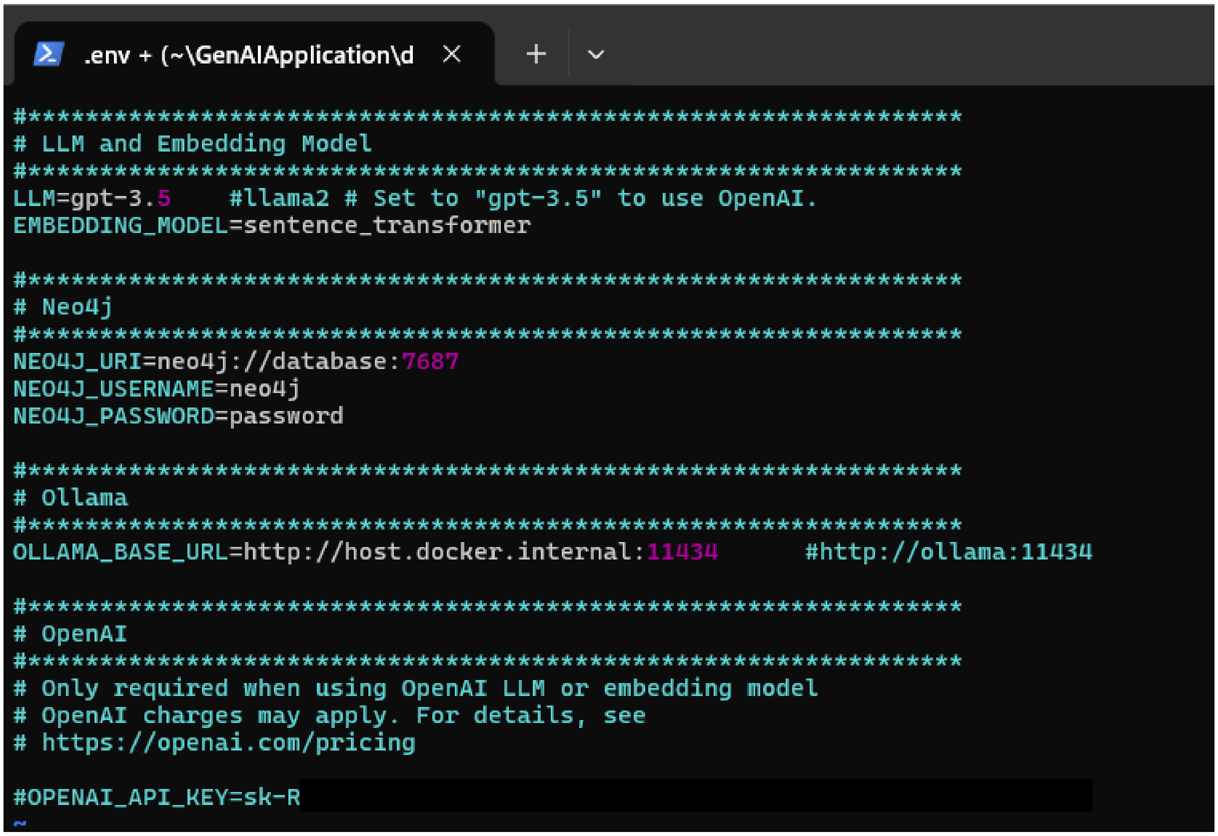
```
$ ollama pull llama2
```



```
PS C:\Users\Niyat Habtom Seghid> ollama pull llama2
pulling manifest
pulling 8934d96d3f08... 100%                                                    3.8 GB
pulling 8c17c2ebb0ea... 100%                                                    7.0 KB
pulling 7c23fb36d801... 100%                                                    4.8 KB
pulling 2e0493f67d0c... 100%                                                      59 B
pulling fa304d675061... 100%                                                      91 B
pulling 42ba7f8a01dd... 100%                                                     557 B
verifying sha256 digest
writing manifest
removing any unused layers
success
PS C:\Users\Niyat Habtom Seghid>
PS C:\Users\Niyat Habtom Seghid>
```

Note: In case you are using OpenAI you can do the following steps instead.

i. Update the LLM value in your .env file to gpt-3.5.
ii. Uncomment and update the OPENAI_API_KEY value in your .env file to your OpenAI API key



```
#*****************************************************************
# LLM and Embedding Model
#*****************************************************************
LLM=gpt-3.5    #llama2 # Set to "gpt-3.5" to use OpenAI.
EMBEDDING_MODEL=sentence_transformer

#*****************************************************************
# Neo4j
#*****************************************************************
NEO4J_URI=neo4j://database:7687
NEO4J_USERNAME=neo4j
NEO4J_PASSWORD=password

#*****************************************************************
# Ollama
#*****************************************************************
OLLAMA_BASE_URL=http://host.docker.internal:11434      #http://ollama:11434

#*****************************************************************
# OpenAI
#*****************************************************************
# Only required when using OpenAI LLM or embedding model
# OpenAI charges may apply. For details, see
# https://openai.com/pricing

#OPENAI_API_KEY=sk-R[redacted]
~
```

Run Your GenAI Application

Niyat Habtom Seghid

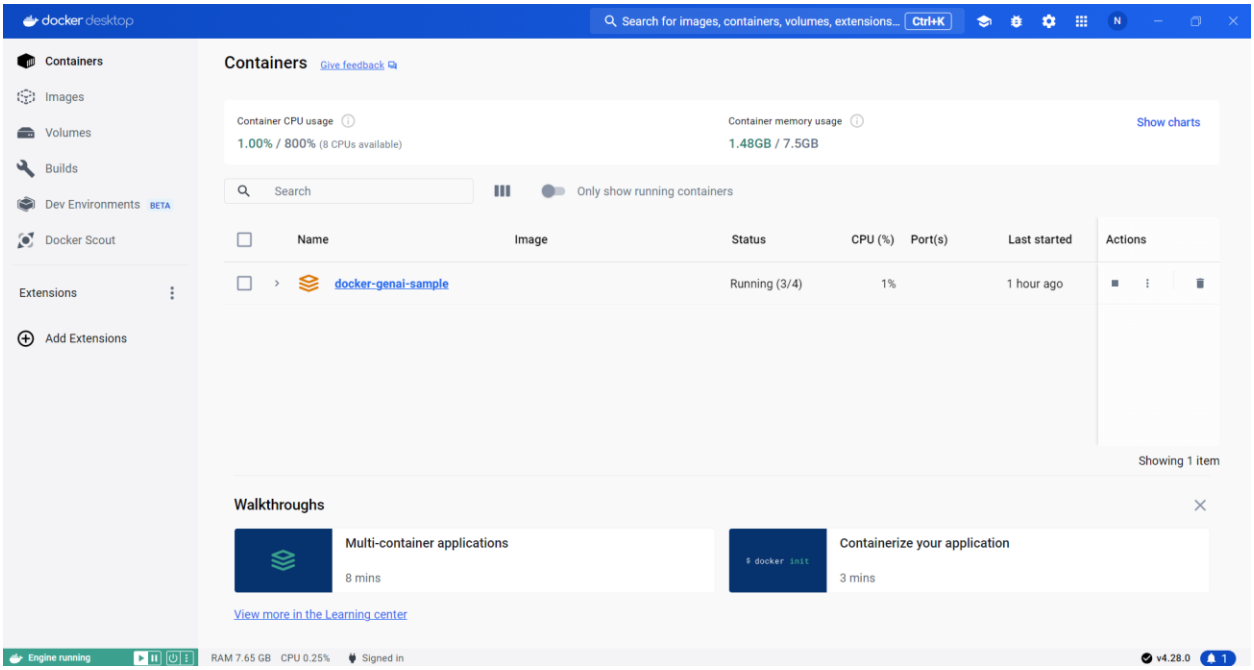
1. To run all the services, run the following command.

   $ docker compose up --build

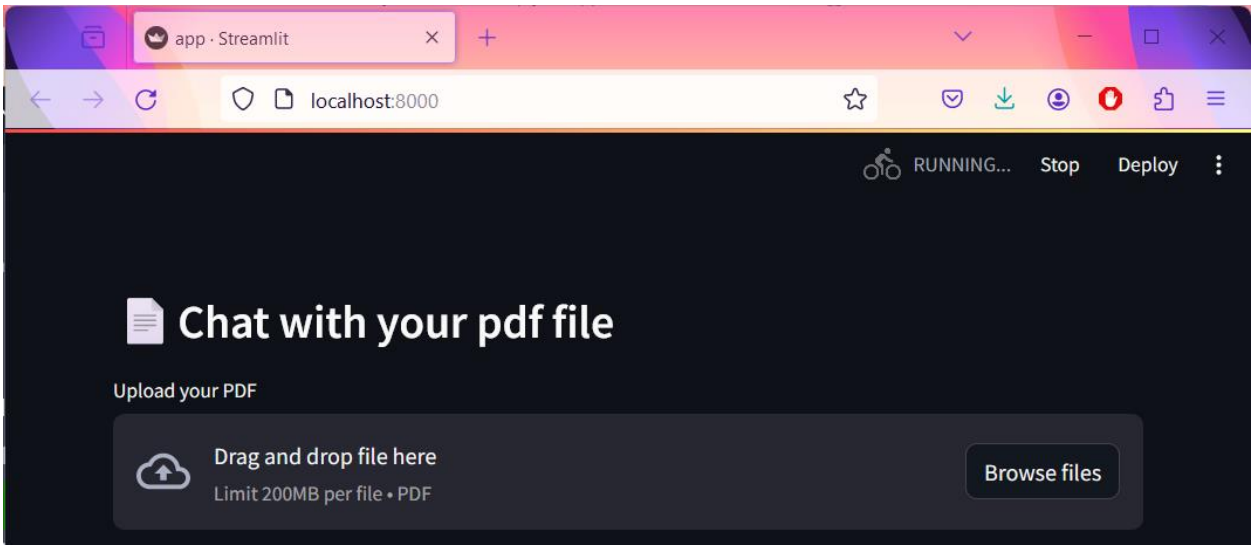
```
PS C:\Users\Niyat Habtom Seghid\GenAIApplication\docker-genai-sample> vim compose.yaml
PS C:\Users\Niyat Habtom Seghid\GenAIApplication\docker-genai-sample> docker compose up --build
[+] Running 8/12
 - ollama 3 layers [⣿⣿⣿⣿] 277.6MB/282.8MB Pulling                                    458.8s
   ✔bccd10f490ab Pull complete                                                         51.2s
   ✔fa3c11b406f0 Pull complete                                                         57.0s
   - 331bdb4bb0dd Downloading     [==================================...              452.1s
 - ollama-pull 7 layers [⣿⣿⣿⣿⣿⣿⣿⣿]   261MB/380.6MB Pulling                          458.8s
   ✔a48641193673 Pull complete                                                         86.1s
   ✔496e8c35aa41 Pull complete                                                         67.9s
   ✔d7f704120c50 Pull complete                                                         70.9s
   ✔25219de7956a Pull complete                                                        128.8s
   ✔4f4fb700ef54 Pull complete                                                         87.5s
   - 0977b56ccc02 Downloading     [==================================...              452.0s
   ✔1eadfce5a711 Download complete                                                    131.8s
```

   - Wait until everything is built and service is started.

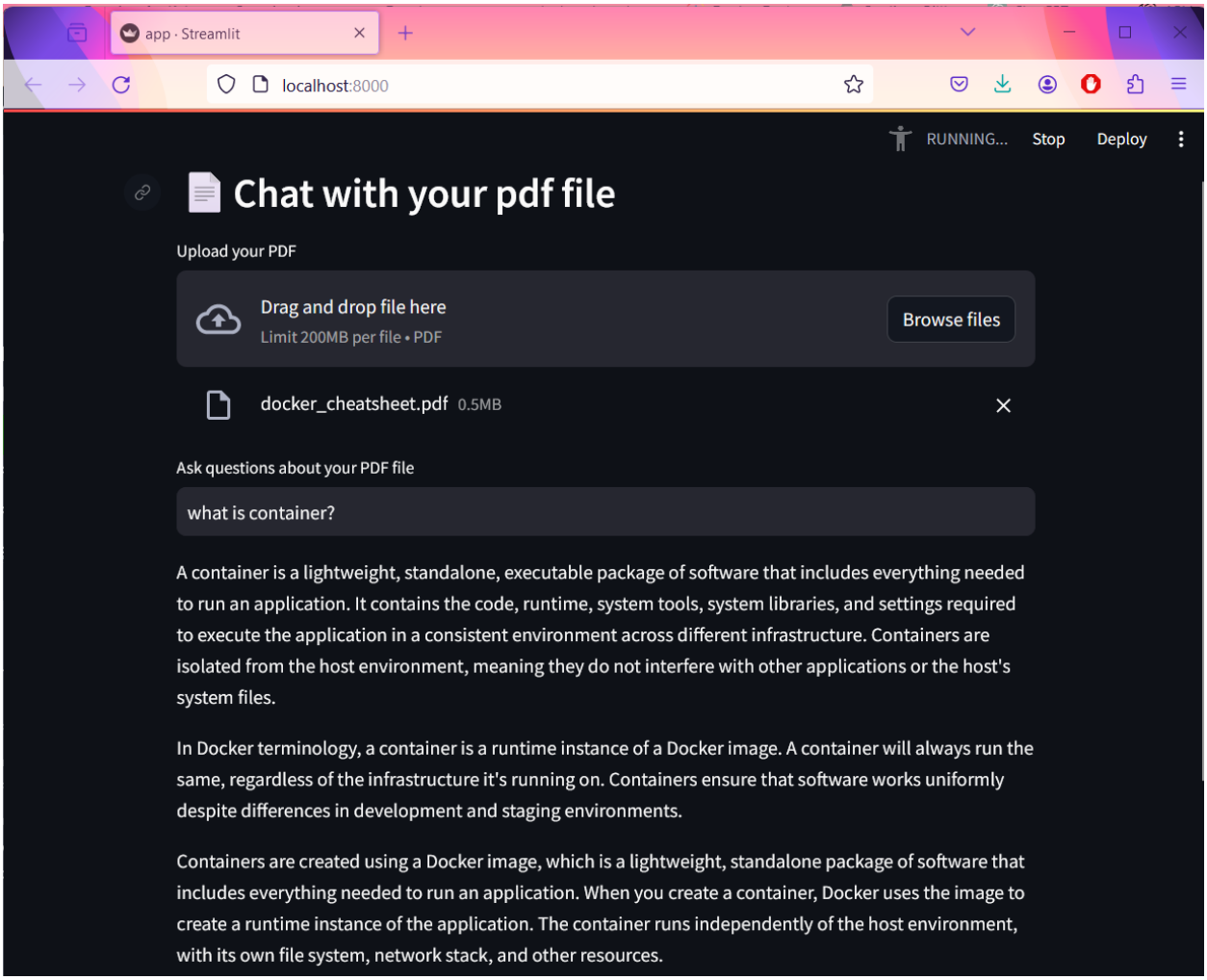
```
PS C:\Users\Niyat Habtom Seghid\GenAIApplication\docker-genai-sample> docker compose up --build
[+] Running 12/12
 ✔ollama 3 layers [⣿⣿⣿⣿]      0B/0B      Pulled                                        475.4s
   ✔bccd10f490ab Pull complete                                                         51.2s
   ✔fa3c11b406f0 Pull complete                                                         57.0s
   ✔331bdb4bb0dd Pull complete                                                        464.8s
 ✔ollama-pull 7 layers [⣿⣿⣿⣿⣿⣿⣿⣿]      0B/0B      Pulled                              562.6s
   ✔a48641193673 Pull complete                                                         86.1s
   ✔496e8c35aa41 Pull complete                                                         67.9s
   ✔d7f704120c50 Pull complete                                                         70.9s
   ✔25219de7956a Pull complete                                                        128.8s
   ✔4f4fb700ef54 Pull complete                                                         87.5s
   ✔0977b56ccc02 Pull complete                                                        541.7s
   ✔1eadfce5a711 Pull complete                                                        131.8s
[+] Building 53.6s (14/14) FINISHED                                          docker:default
 => [server internal] load build definition from Dockerfile                           0.9s
 => => transferring dockerfile: 1.71kB                                                0.5s
 => [server] resolve image config for docker.io/docker/dockerfile:1                  32.9s
 => [server auth] docker/dockerfile:pull token for registry-1.docker.io               0.0s
 => CACHED [server] docker-image://docker.io/docker/dockerfile:1@sha256:dbbd5e059e8a07ff7ea6233  0.0s
 => [server internal] load metadata for docker.io/library/python:3.11.7-slim         17.5s
 => [server auth] library/python:pull token for registry-1.docker.io                  0.0s
 => [server internal] load .dockerignore                                              0.2s
 => => transferring context: 667B                                                     0.0s
 => [server base 1/5] FROM docker.io/library/python:3.11.7-slim@sha256:53d6284a40eae6b625f22870  0.0s
 => [server internal] load build context                                              0.1s
 => => transferring context: 160.03kB                                                 0.1s
 => CACHED [server base 2/5] WORKDIR /app                                             0.0s
 => CACHED [server base 3/5] RUN adduser     --disabled-password     --gecos ""     --home "/no  0.0s
 => CACHED [server base 4/5] RUN --mount=type=cache,target=/root/.cache/pip     --mount=type=bi  0.0s
 => [server base 5/5] COPY . .                                                        0.1s
 => [server] exporting to image                                                       0.1s
 => => exporting layers                                                               0.1s
 => => writing image sha256:5e0941101260057660 22e0a07bd8a707ba49147d40ea7aedda5615251b430c77  0.0s
 => => naming to docker.io/library/docker-genai-sample-server                         0.0s
[+] Running 5/5
 ✔Volume "docker-genai-sample_ollama_volume"   Created                                0.0s
 ✔Container docker-genai-sample-ollama-pull-1  Created                                0.3s
 ✔Container docker-genai-sample-ollama-1       Crea...                                0.3s
 ✔Container docker-genai-sample-database-1     Cr...                                  0.0s
 ✔Container docker-genai-sample-server-1       Recr...                                0.8s
Attaching to database-1, ollama-1, ollama-pull-1, server-1
Error response from daemon: failed to create task for container: failed to create shim task: OCI runti
me create failed: runc create failed: unable to start container process: error during container init:
error running hook #0: error running hook: exit status 1, stdout: , stderr: Auto-detected mode as 'leg
acy'
nvidia-container-cli: initialization error: WSL environment detected but no adapters were found: unkno
wn
PS C:\Users\Niyat Habtom Seghid\GenAIApplication\docker-genai-sample> Start-Process -FilePath "C:\User
```



2. Once the application is running, open a browser and access the application at http://localhost:8000.



3. Then we can upload a PDF file, for example the Docker CLI Cheat Sheet, and ask a question about the PDF.



Through this we have set up a development environment that provides access to all the services that our GenAI application needs.

==========End==========

Step 3: Link to GitHub

https://github.com/niyat33/Cloud-Computing/tree/eb29750a1bb65c923550a73e64e13cf402390e26/Kubernetes/Generative_AI/My%20containerized%20app