Project: GenAI - Develop your containerized app

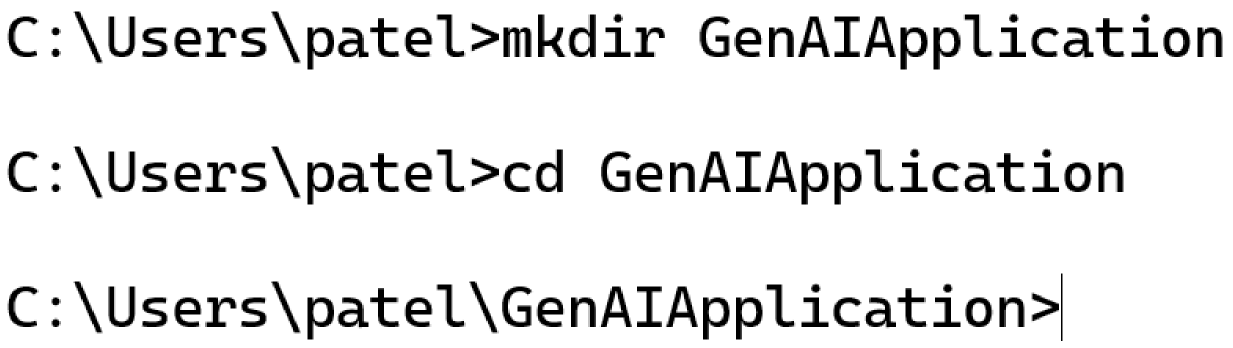
Step 1: GenAI - Containerize your app.

1. First, install the latest version of Docker Desktop for windows.
    - Docker Desktop: The #1 Containerization Tool for Developers | Docker

The #1 containerization software for developers and teams

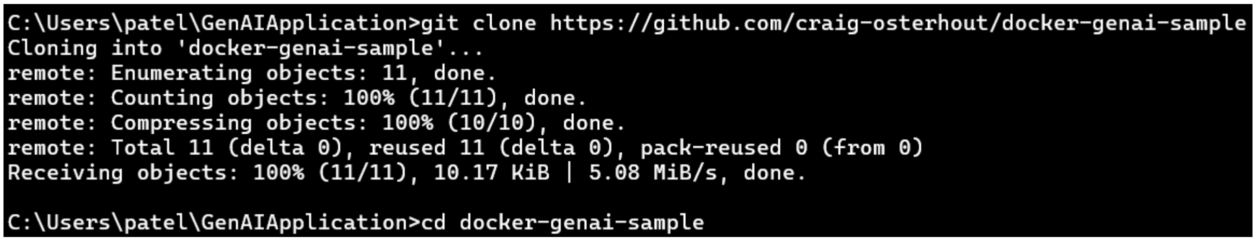
Streamline development with Docker Desktop's powerful container tools.

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



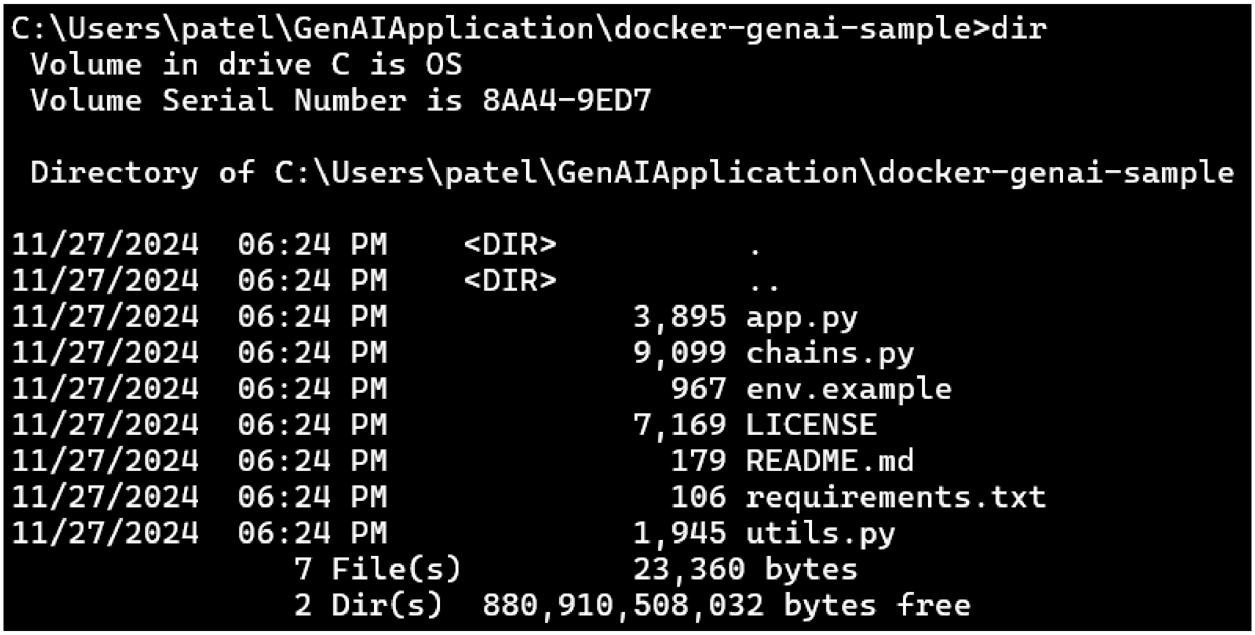
```
C:\Users\patel>mkdir GenAIApplication

C:\Users\patel>cd GenAIApplication

C:\Users\patel\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



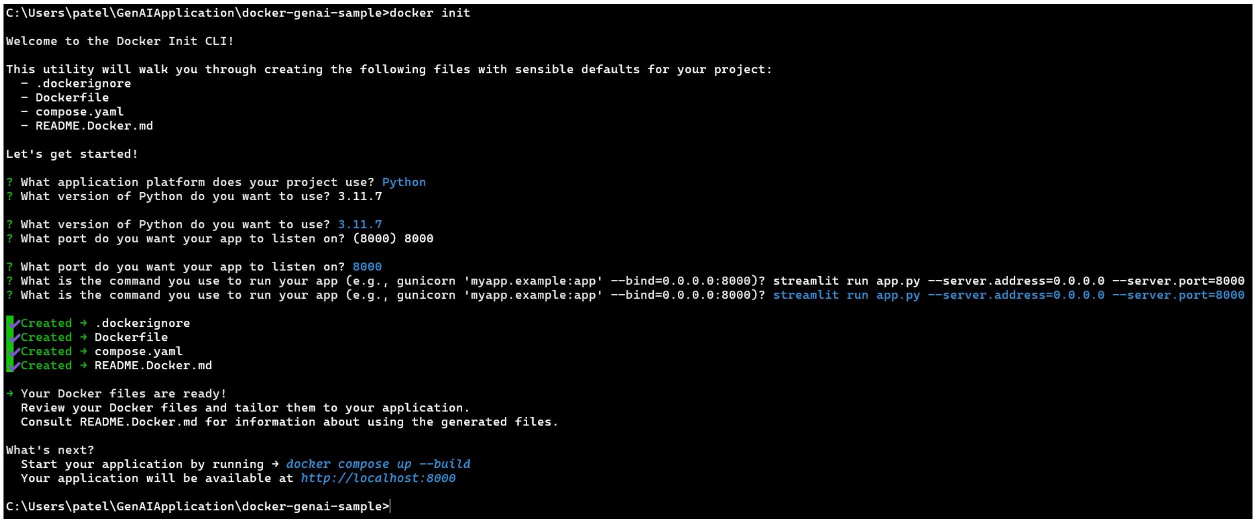
```
C:\Users\patel\GenAIApplication>git clone https://github.com/craig-osterhout/docker-genai-sample
Cloning into 'docker-genai-sample'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 11 (delta 0), reused 11 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (11/11), 10.17 KiB | 5.08 MiB/s, done.

C:\Users\patel\GenAIApplication>cd docker-genai-sample
```

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



```
C:\Users\patel\GenAIApplication\docker-genai-sample>dir
 Volume in drive C is OS
 Volume Serial Number is 8AA4-9ED7

 Directory of C:\Users\patel\GenAIApplication\docker-genai-sample

11/27/2024  06:24 PM    <DIR>          .
11/27/2024  06:24 PM    <DIR>          ..
11/27/2024  06:24 PM             3,895 app.py
11/27/2024  06:24 PM             9,099 chains.py
11/27/2024  06:24 PM               967 env.example
11/27/2024  06:24 PM             7,169 LICENSE
11/27/2024  06:24 PM               179 README.md
11/27/2024  06:24 PM               106 requirements.txt
11/27/2024  06:24 PM             1,945 utils.py
               7 File(s)         23,360 bytes
               2 Dir(s)  880,910,508,032 bytes free
```

5. 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.



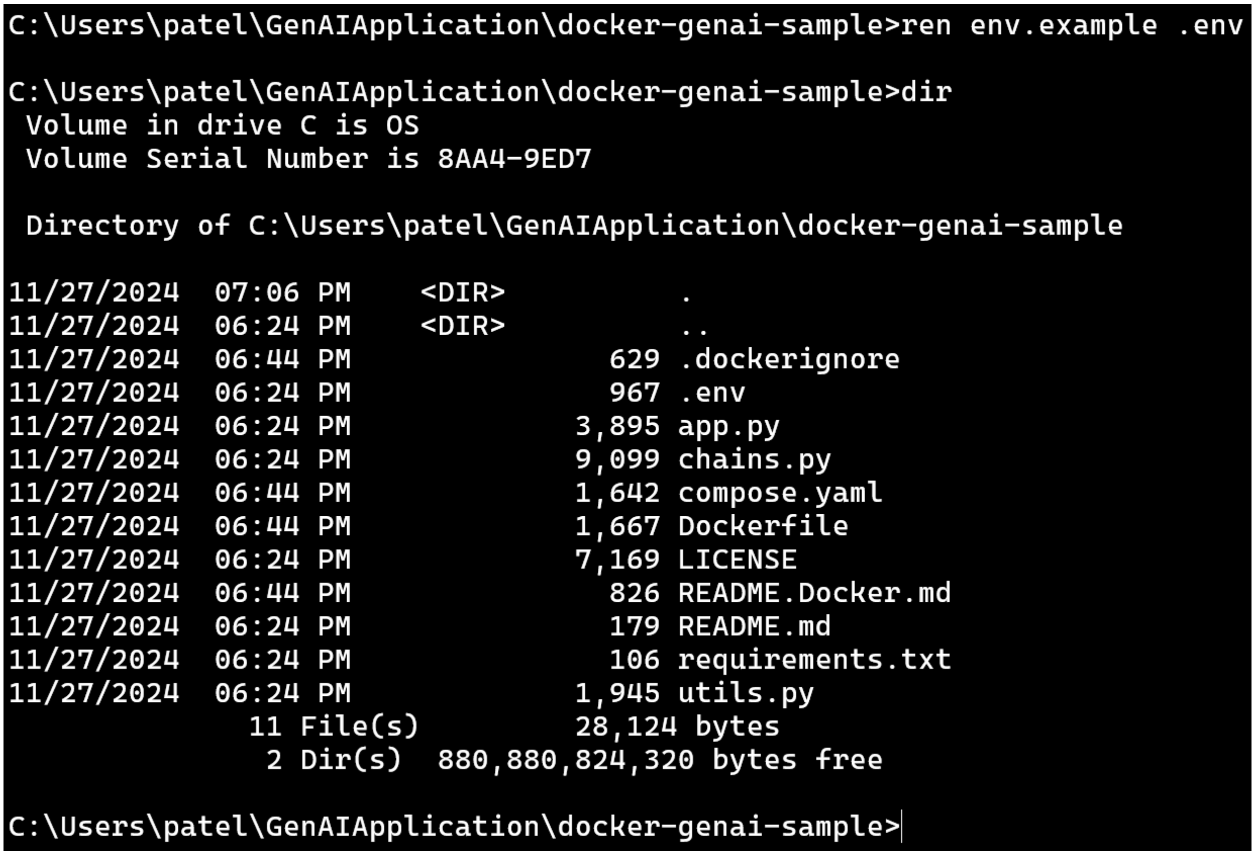
```
C:\Users\patel\GenAIApplication\docker-genai-sample>docker init

Welcome to the Docker Init CLI!

This utility will walk you through creating the following files with sensible defaults for your 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!
  Review your Docker files and tailor them to your application.
  Consult README.Docker.md for information about using the generated files.

What's next?
  Start your application by running → docker compose up --build
  Your application will be available at http://localhost:8000

C:\Users\patel\GenAIApplication\docker-genai-sample>
```

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.



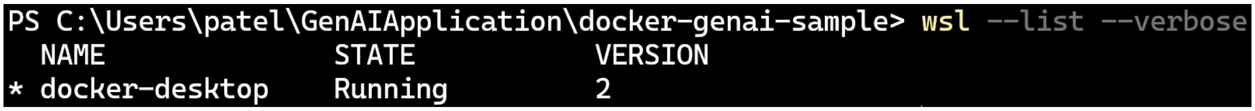
```
C:\Users\patel\GenAIApplication\docker-genai-sample>ren env.example .env

C:\Users\patel\GenAIApplication\docker-genai-sample>dir
 Volume in drive C is OS
 Volume Serial Number is 8AA4-9ED7

 Directory of C:\Users\patel\GenAIApplication\docker-genai-sample

11/27/2024  07:06 PM    <DIR>          .
11/27/2024  06:24 PM    <DIR>          ..
11/27/2024  06:44 PM               629 .dockerignore
11/27/2024  06:24 PM               967 .env
11/27/2024  06:24 PM             3,895 app.py
11/27/2024  06:24 PM             9,099 chains.py
11/27/2024  06:44 PM             1,642 compose.yaml
11/27/2024  06:44 PM             1,667 Dockerfile
11/27/2024  06:24 PM             7,169 LICENSE
11/27/2024  06:44 PM               826 README.Docker.md
11/27/2024  06:24 PM               179 README.md
11/27/2024  06:24 PM               106 requirements.txt
11/27/2024  06:24 PM             1,945 utils.py
              11 File(s)         28,124 bytes
               2 Dir(s)  880,880,824,320 bytes free

C:\Users\patel\GenAIApplication\docker-genai-sample>
```

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

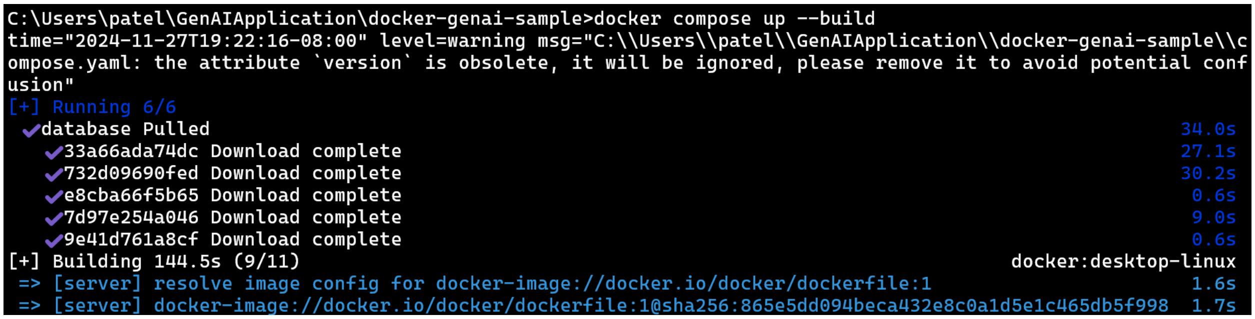
4. Run the application. Inside the docker-genai-sample directory, run the following command in a terminal.
Before running below step, open your Docker Desktop

- wsl --list --verbose



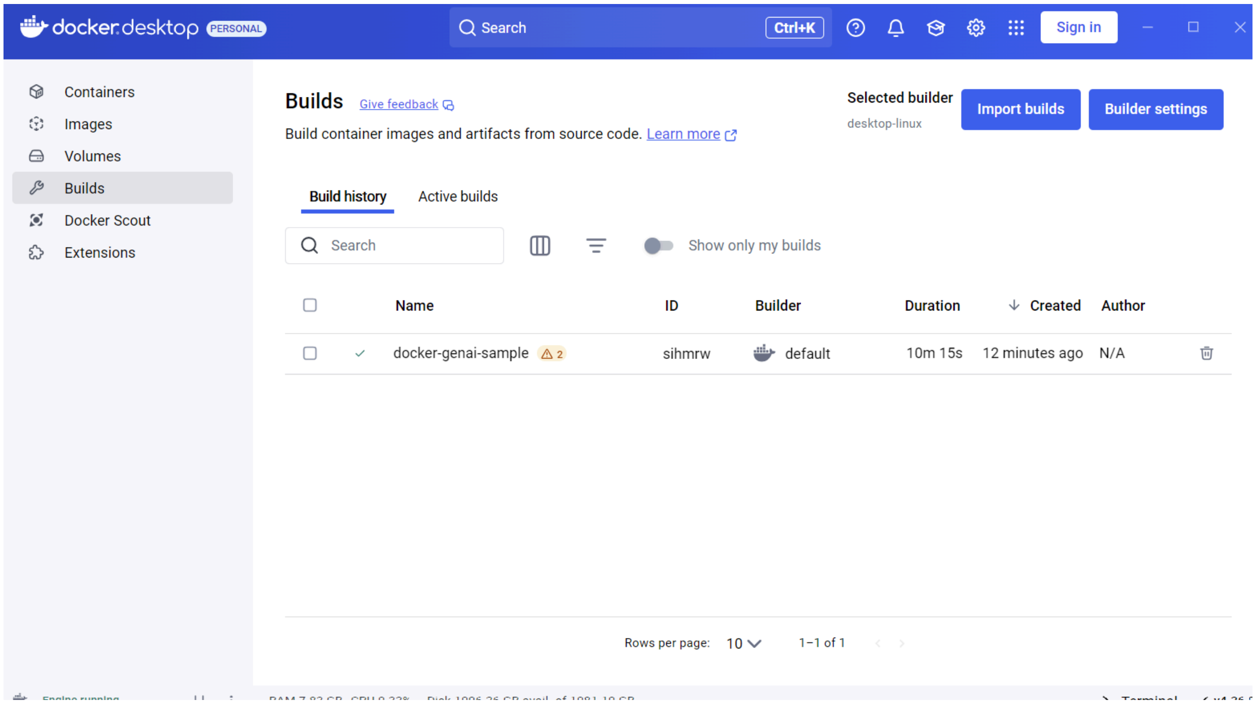
```
PS C:\Users\patel\GenAIApplication\docker-genai-sample> wsl --list --verbose
  NAME                   STATE           VERSION
* docker-desktop         Running         2
```

- docker compose up --build



```
C:\Users\patel\GenAIApplication\docker-genai-sample>docker compose up --build
time="2024-11-27T19:22:16-08:00" level=warning msg="C:\\Users\\patel\\GenAIApplication\\docker-genai-sample\\c
ompose.yaml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential conf
usion"
[+] Running 6/6
 ✔database Pulled                                                                                        34.0s
   ✔33a66ada74dc Download complete                                                                       27.1s
   ✔732d09690fed Download complete                                                                       30.2s
   ✔e8cba66f5b65 Download complete                                                                        0.6s
   ✔7d97e254a046 Download complete                                                                        9.0s
   ✔9e41d761a8cf Download complete                                                                        0.6s
[+] Building 144.5s (9/11)                                                              docker:desktop-linux
 => [server] resolve image config for docker-image://docker.io/docker/dockerfile:1                        1.6s
 => [server] docker-image://docker.io/docker/dockerfile:1@sha256:865e5dd094beca432e8c0a1d5e1c465db5f998  1.7s
```

- We can also see the progress from Docker Desktop.



Builds Give feedback

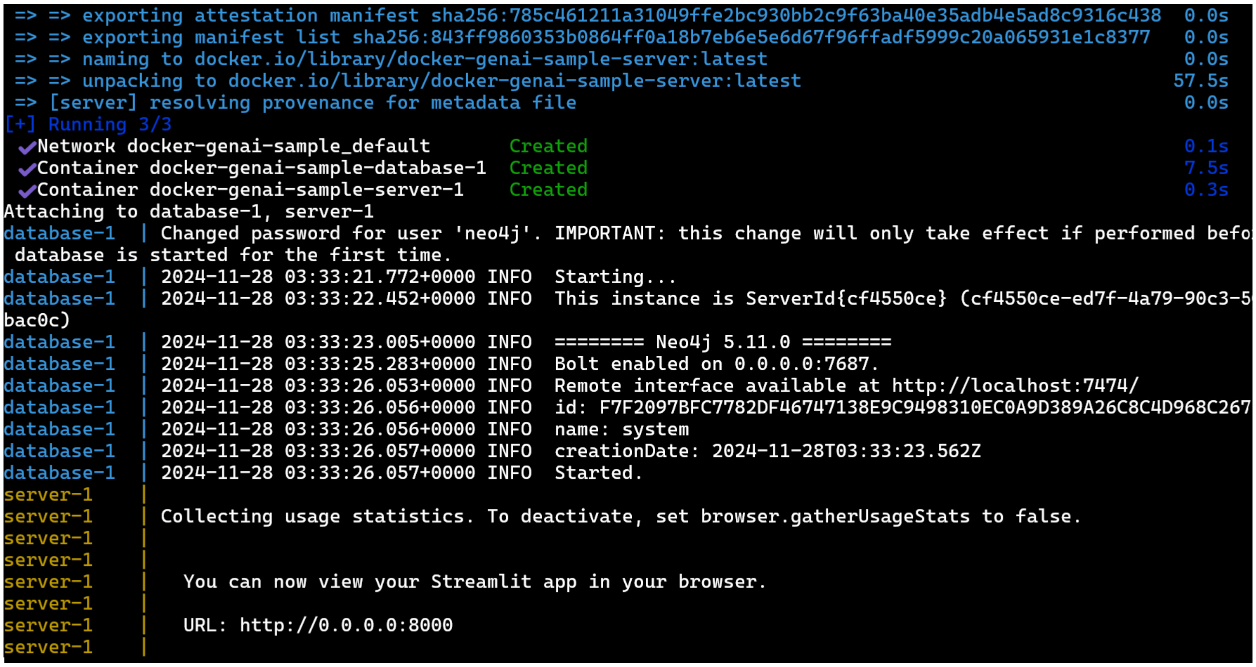
Build container images and artifacts from source code. Learn more

Selected builder
desktop-linux

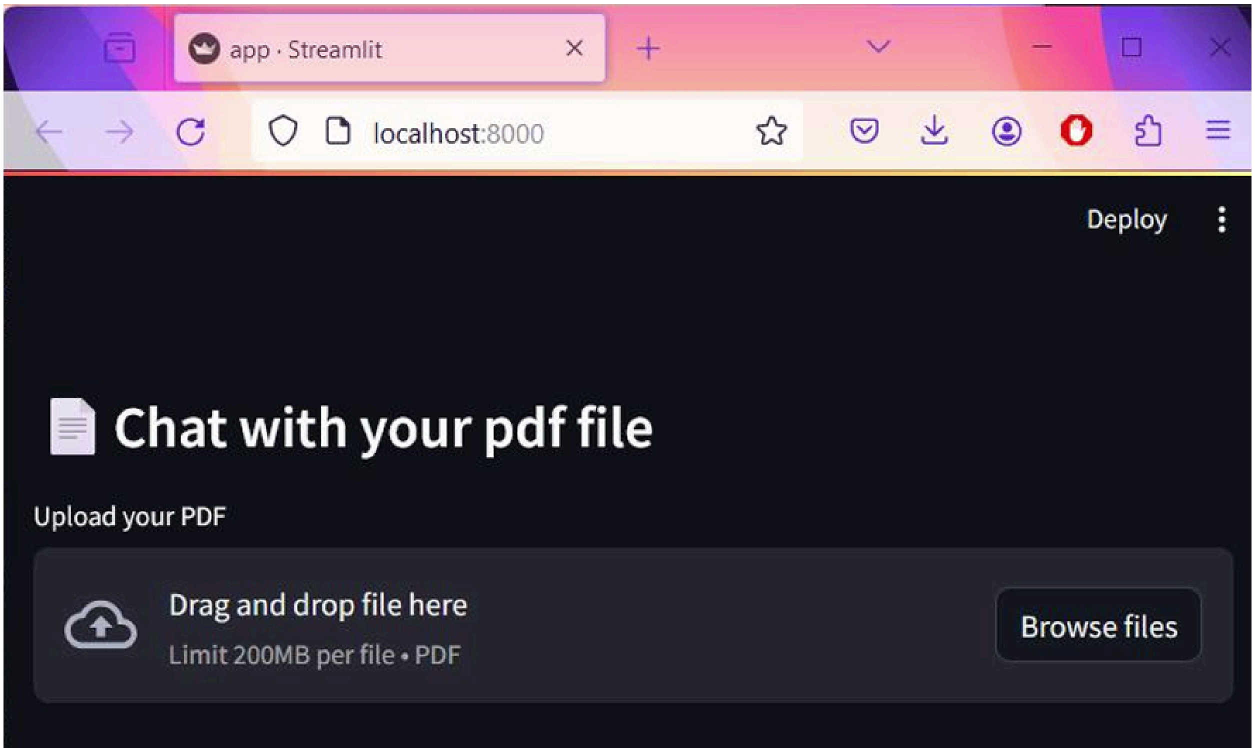
□		Name	ID	Builder	Duration	↓ Created	Author	
□	✓	docker-genai-sample ⚠ 2	sihmrw	default	10m 15s	12 minutes ago	N/A	

Rows per page: 10 1–1 of 1

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



```
 => => exporting attestation manifest sha256:785c461211a31049ffe2bc930bb2c9f63ba40e35adb4e5ad8c9316c438  0.0s
 => => exporting manifest list sha256:843ff9860353b0864ff0a18b7eb6e5e6d67f96ffadf5999c20a065931e1c8377   0.0s
 => => naming to docker.io/library/docker-genai-sample-server:latest                                     0.0s
 => => unpacking to docker.io/library/docker-genai-sample-server:latest                                 57.5s
 => [server] resolving provenance for metadata file                                                      0.0s
[+] Running 3/3
 ✔Network docker-genai-sample_default       Created                                                      0.1s
 ✔Container docker-genai-sample-database-1  Created                                                      7.5s
 ✔Container docker-genai-sample-server-1    Created                                                      0.3s
Attaching to database-1, server-1
database-1  | Changed password for user 'neo4j'. IMPORTANT: this change will only take effect if performed befo
 database is started for the first time.
database-1  | 2024-11-28 03:33:21.772+0000 INFO  Starting...
database-1  | 2024-11-28 03:33:22.452+0000 INFO  This instance is ServerId{cf4550ce} (cf4550ce-ed7f-4a79-90c3-5
bac0c)
database-1  | 2024-11-28 03:33:23.005+0000 INFO  ======== Neo4j 5.11.0 ========
database-1  | 2024-11-28 03:33:25.283+0000 INFO  Bolt enabled on 0.0.0.0:7687.
database-1  | 2024-11-28 03:33:26.053+0000 INFO  Remote interface available at http://localhost:7474/
database-1  | 2024-11-28 03:33:26.056+0000 INFO  id: F7F2097BFC7782DF46747138E9C9498310EC0A9D389A26C8C4D968C267
database-1  | 2024-11-28 03:33:26.056+0000 INFO  name: system
database-1  | 2024-11-28 03:33:26.057+0000 INFO  creationDate: 2024-11-28T03:33:23.562Z
database-1  | 2024-11-28 03:33:26.057+0000 INFO  Started.
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    |
```



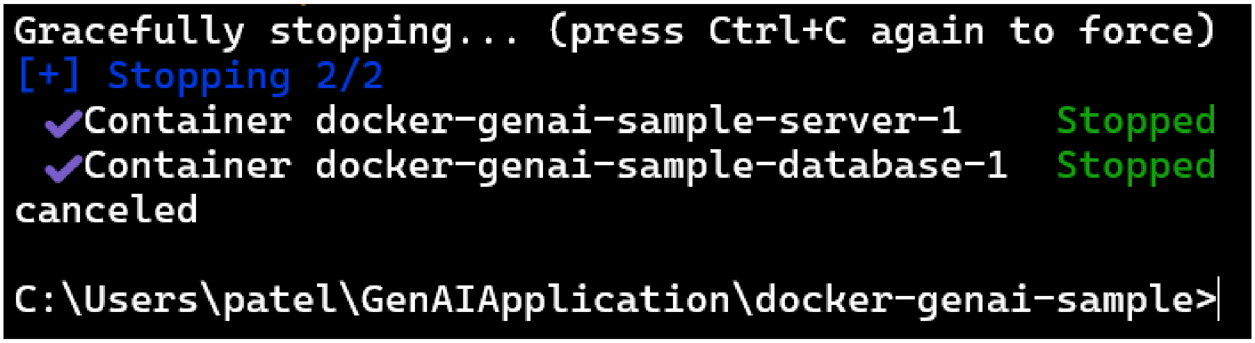
Chat with your pdf file

Upload your PDF

Drag and drop file here

Limit 200MB per file • PDF

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

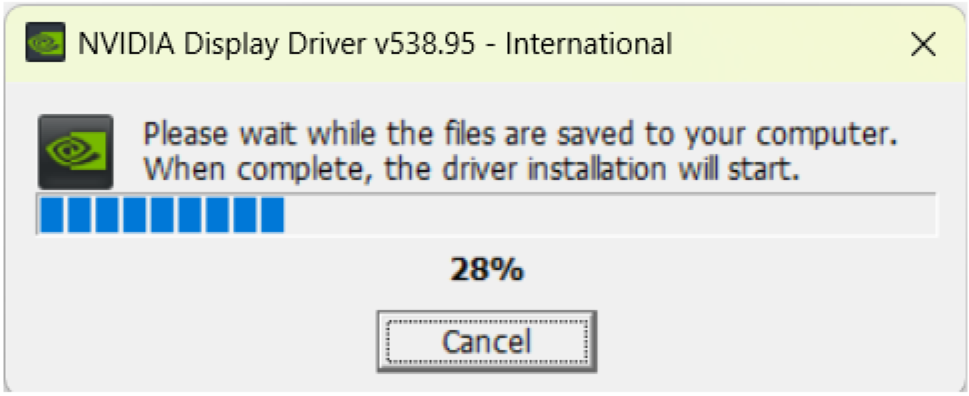


```
Gracefully stopping... (press Ctrl+C again to force)
[+] Stopping 2/2
 ✔Container docker-genai-sample-server-1     Stopped
 ✔Container docker-genai-sample-database-1   Stopped
canceled

C:\Users\patel\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**

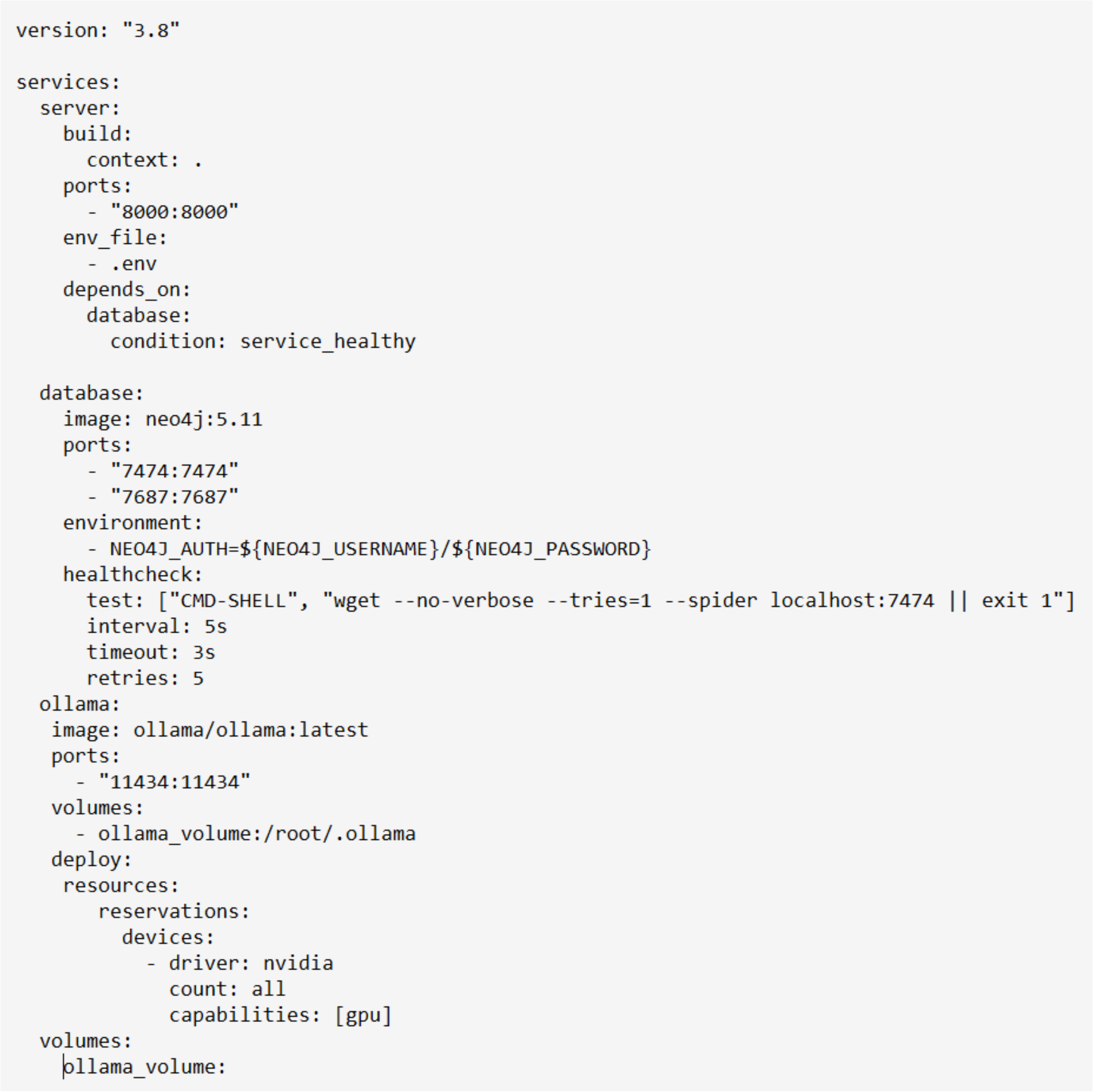


NVIDIA Display Driver v538.95 - International

Please wait while the files are saved to your computer.
When complete, the driver installation will start.

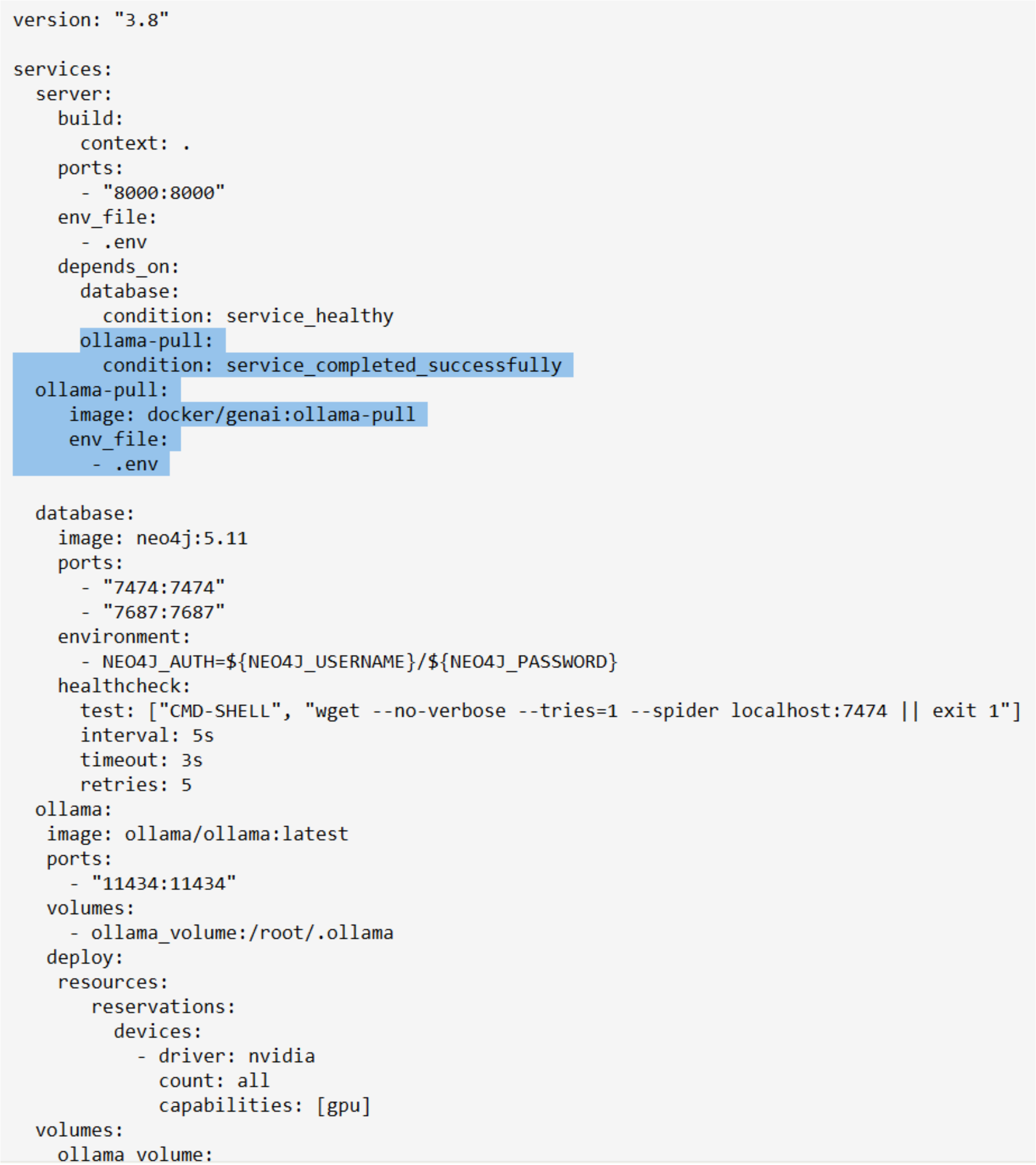
28%

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



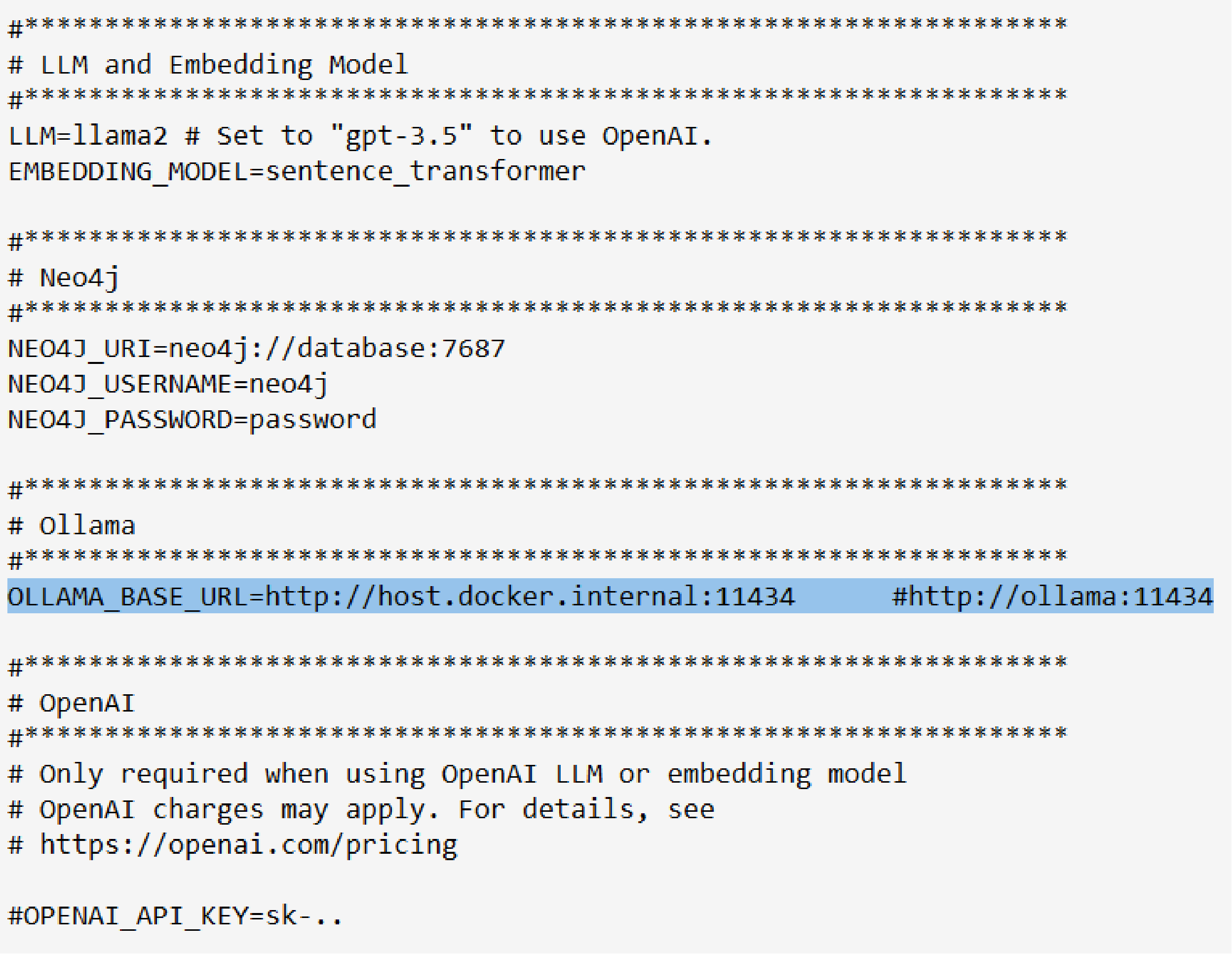
```
version: "3.8"

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:



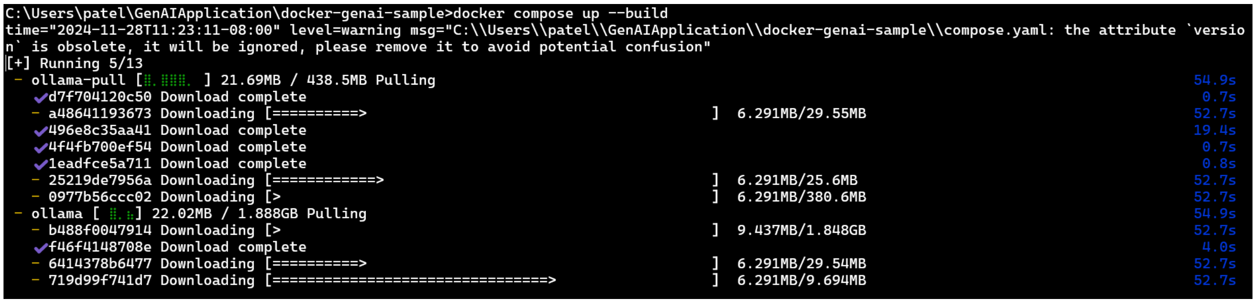
```yaml
version: "3.8"

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:
```

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

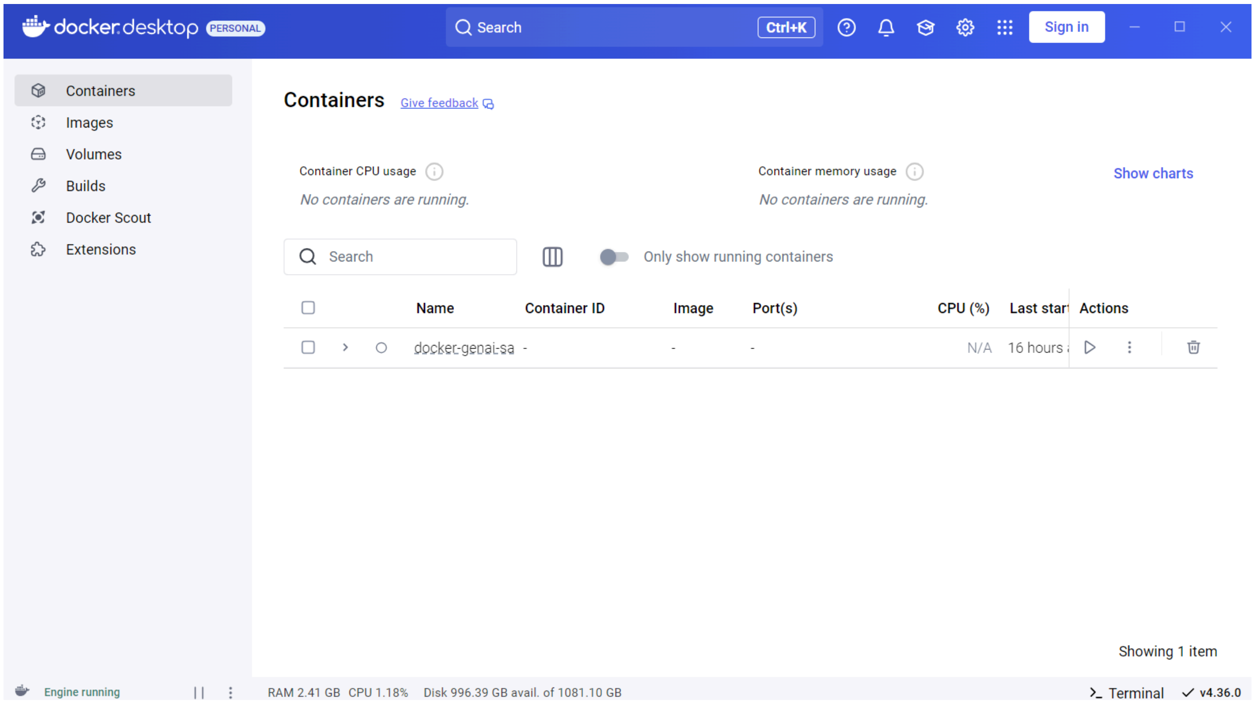


```
#*****************************************************************
# LLM and Embedding Model
#*****************************************************************
LLM=llama2 # Set to "gpt-3.5" to use OpenAI.
EMBEDDING_MODEL=sentence_transformer

#*****************************************************************
# 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-..
```

Add configuration for ollama pull and ollam in docker compose file



```
C:\Users\patel\GenAIApplication\docker-genai-sample>docker compose up --build
time="2024-11-28T11:23:11-08:00" level=warning msg="C:\\Users\\patel\\GenAIApplication\\docker-genai-sample\\compose.yaml: the attribute `versio
n` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 5/13
 - ollama-pull [⣿.⣿⣿⣿⣿. ] 21.69MB / 438.5MB Pulling                                                                                    54.9s
  ✓d7f704120c50 Download complete                                                                                                         0.7s
  - a48641193673 Downloading [==========>                                             ]  6.291MB/29.55MB                                 52.7s
  ✓496e8c35aa41 Download complete                                                                                                        19.4s
  ✓4f4fb700ef54 Download complete                                                                                                         0.7s
  ✓1eadfce5a711 Download complete                                                                                                         0.8s
  - 25219de7956a Downloading [============>                                           ]  6.291MB/25.6MB                                  52.7s
  - 0977b56ccc02 Downloading [>                                                       ]  6.291MB/380.6MB                                 52.7s
 - ollama [ ⣿.⣶] 22.02MB / 1.888GB Pulling                                                                                               54.9s
  - b488f0047914 Downloading [>                                                       ]  9.437MB/1.848GB                                 52.7s
  ✓f46f4148708e Download complete                                                                                                         4.0s
  - 6414378b6477 Downloading [==========>                                             ]  6.291MB/29.54MB                                 52.7s
  - 719d99f741d7 Downloading [==================================>                     ]  6.291MB/9.694MB                                 52.7s
```



Containers

Container CPU usage
No containers are running.

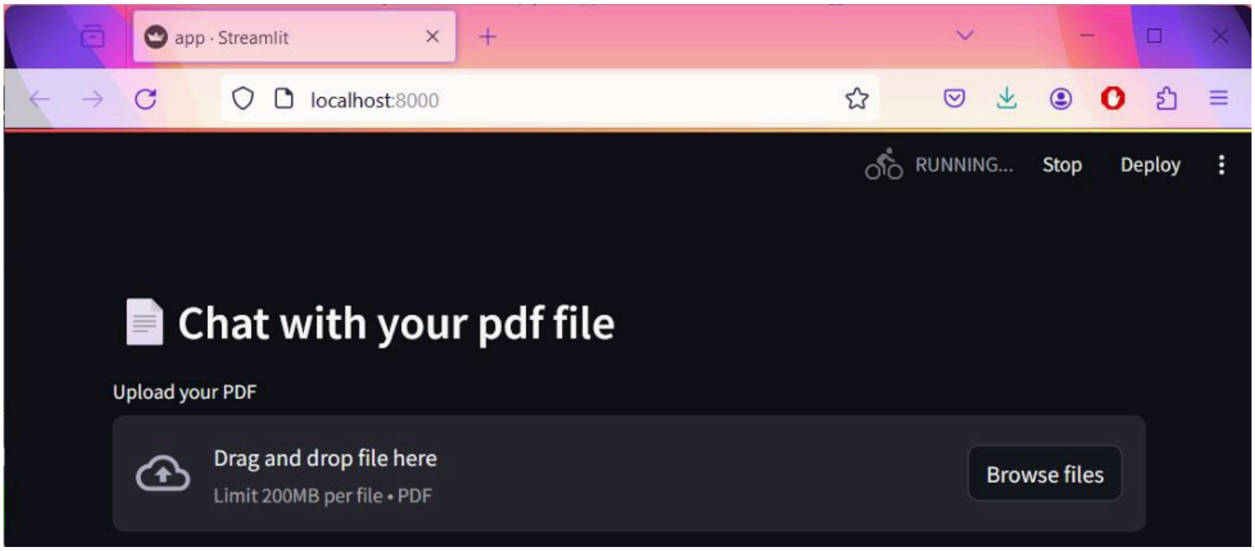
Container memory usage
No containers are running.

□	Name	Container ID	Image	Port(s)	CPU (%)	Last start	Actions
□	docker-genai-sa	-	-	-	N/A	16 hours	

RAM 2.41 GB CPU 1.18% Disk 996.39 GB avail. of 1081.10 GB

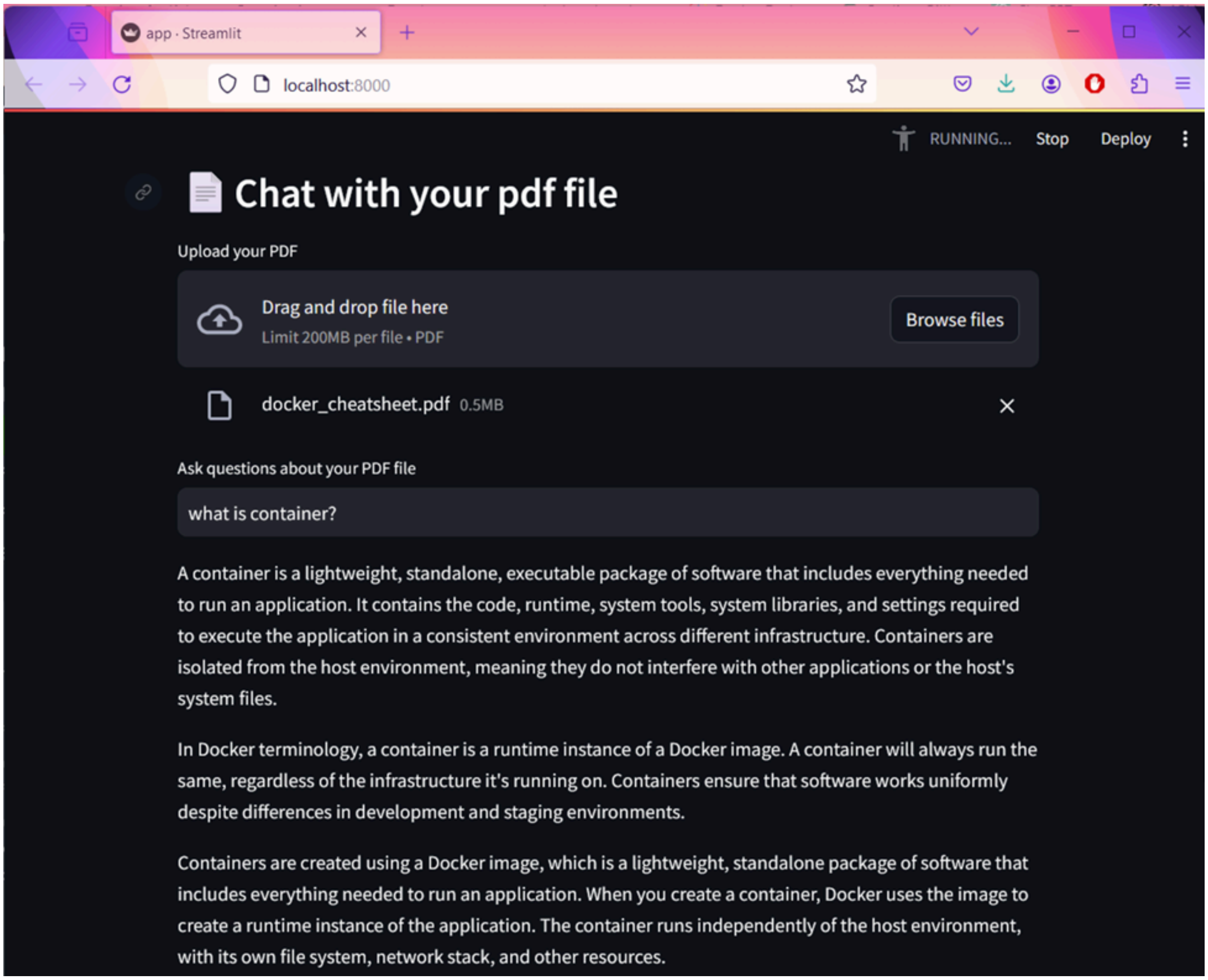
Build new images and containers using docker compose --build

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



Chat with your pdf file

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



Chat with your pdf file

Upload your PDF

Drag and drop file here
Limit 200MB per file • PDF

docker_cheatsheet.pdf 0.5MB

Ask questions about your PDF file

what is container?

A container is a lightweight, standalone, executable package of software that includes everything needed to run an application. It contains the code, runtime, system tools, system libraries, and settings required to execute the application in a consistent environment across different infrastructure. Containers are isolated from the host environment, meaning they do not interfere with other applications or the host's system files.

In Docker terminology, a container is a runtime instance of a Docker image. A container will always run the same, regardless of the infrastructure it's running on. Containers ensure that software works uniformly despite differences in development and staging environments.

Containers are created using a Docker image, which is a lightweight, standalone package of software that includes everything needed to run an application. When you create a container, Docker uses the image to create a runtime instance of the application. The container runs independently of the host environment, with its own file system, network stack, and other resources.

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

Link to GitHub - Cloud-Computing/kubernetes at main · hpatel65373/Cloud-Computing