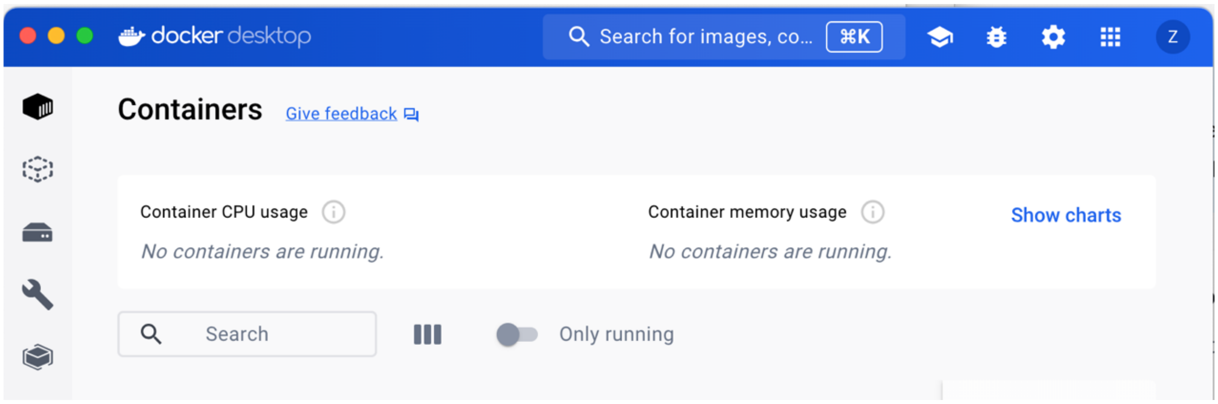
19953 Zhiyu Zhang

CS571: Week 11: Homework 1: GenAI - Containerize your app

1. Prerequisites

Install Docker Desktop here and log into your Docker account:



Container CPU usage

No containers are running.

Container memory usage

No containers are running.

Install a git client here:

In MacOS environment, I used homebrew to install git in terminal with:

```
% brew install git
```

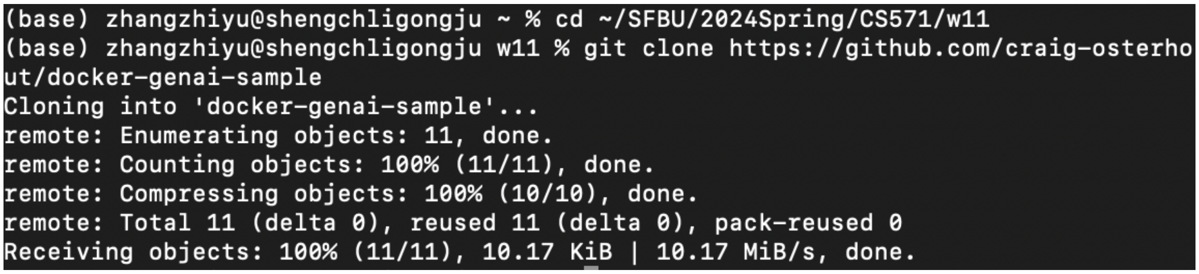
2. Get the sample application

In terminal, go to a desired directory:

```
% cd ~/SFBU/2024Spring/CS571/w11
```

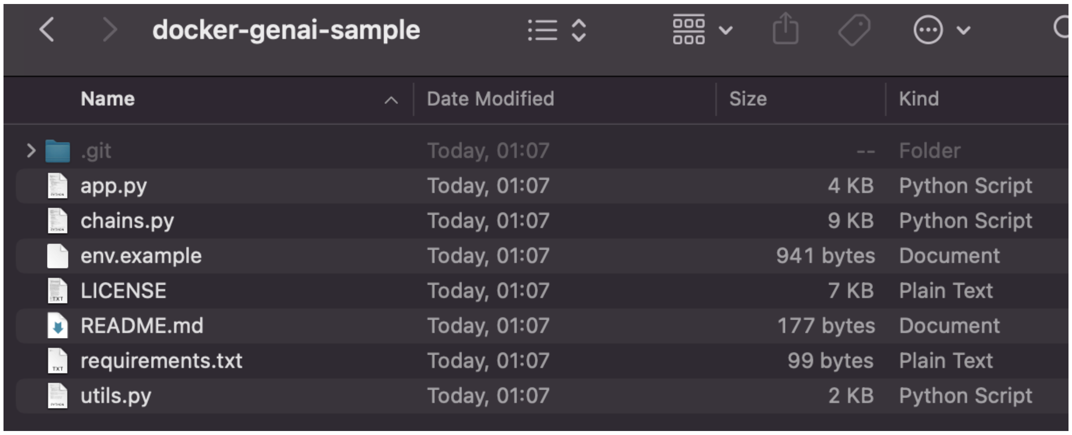
Clone the sample application:

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



```
(base) zhangzhiyu@shengchligongju ~ % cd ~/SFBU/2024Spring/CS571/w11
(base) zhangzhiyu@shengchligongju w11 % git clone https://github.com/craig-osterho
ut/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 | 10.17 MiB/s, done.
```

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



Name	Date Modified	Size	Kind
.git	Today, 01:07	--	Folder
app.py	Today, 01:07	4 KB	Python Script
chains.py	Today, 01:07	9 KB	Python Script
env.example	Today, 01:07	941 bytes	Document
LICENSE	Today, 01:07	7 KB	Plain Text
README.md	Today, 01:07	177 bytes	Document
requirements.txt	Today, 01:07	99 bytes	Plain Text
utils.py	Today, 01:07	2 KB	Python Script

3. Initialize Docker assets

Go to docker-genai-sample directory:

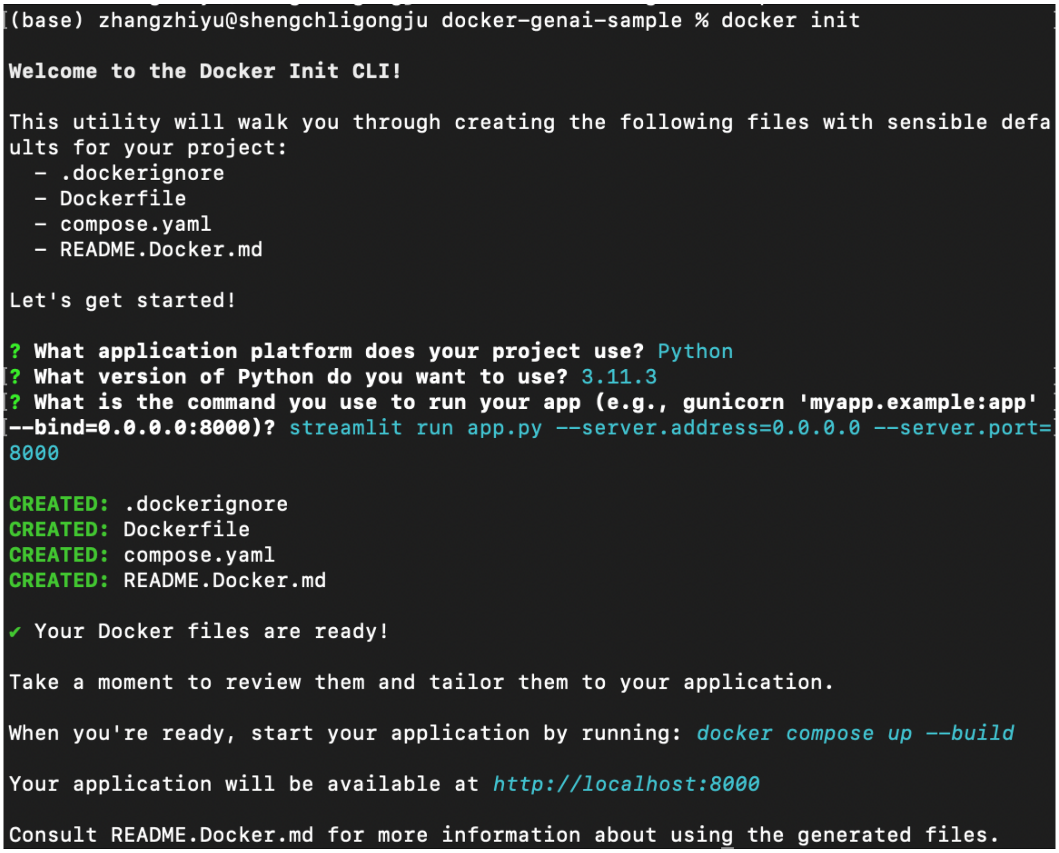
```
    % cd docker-genai-sample
```

Run the following command to containerize your application

```
     % docker init
```

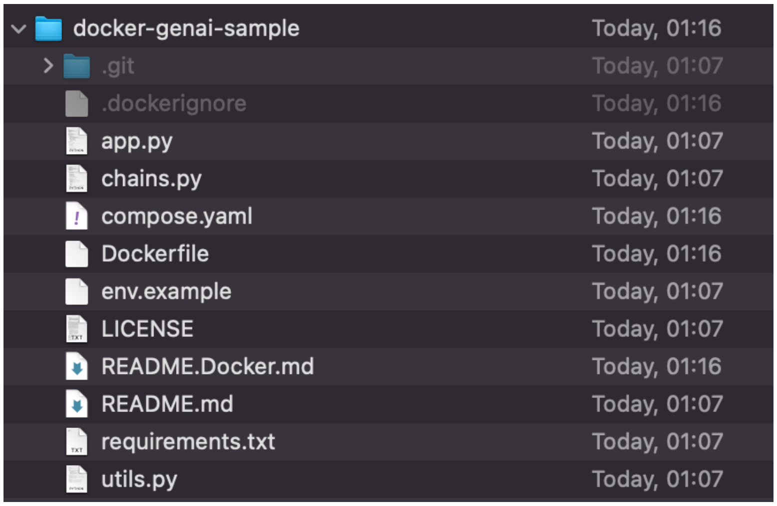
At the prompts following “Let’s get started”, enter the following answers highlighted by red:

```
     ? What application platform does your project use? Python
     ? What version of Python do you want to use? 3.11.3
     ? What port do you want your app to listen on? 8000
     ? What is the command to run your app? streamlit run app.py –
     server.address=0.0.0.0 --server.port=8000
```



```
(base) zhangzhiyu@shengchligongju docker-genai-sample % docker init

Welcome to the Docker Init CLI!

This utility will walk you through creating the following files with sensible defa
ults 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? 3.11.3
? 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.
```

You should now have the following contents in your docker-genai-sample directory:

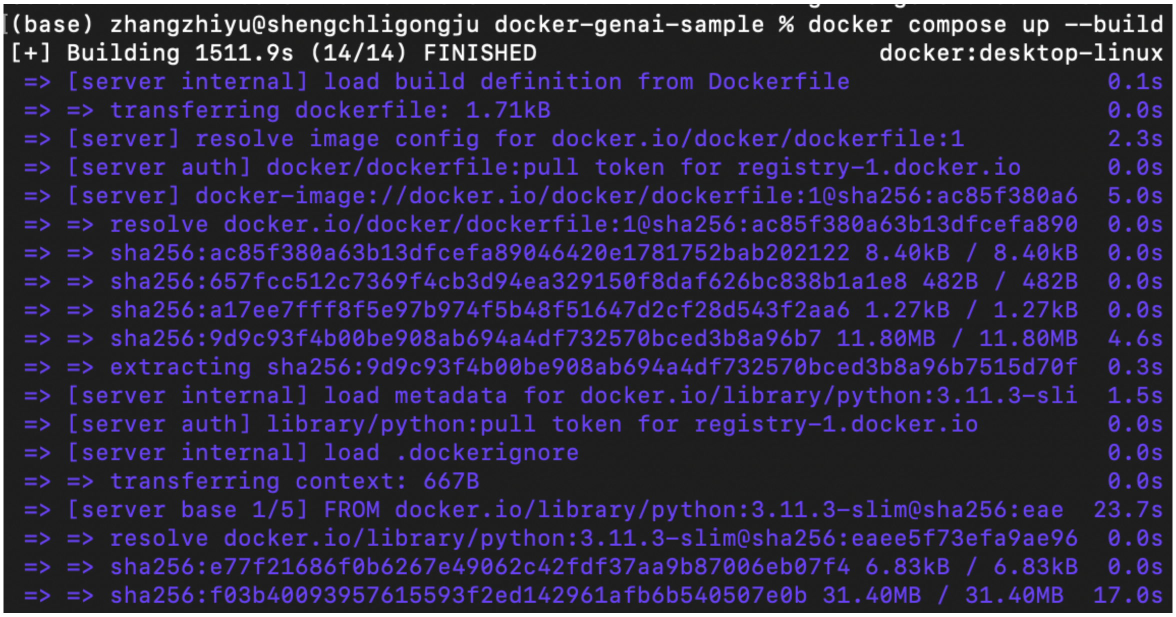


docker-genai-sample	Today, 01:16
.git	Today, 01:07
.dockerignore	Today, 01:16
app.py	Today, 01:07
chains.py	Today, 01:07
compose.yaml	Today, 01:16
Dockerfile	Today, 01:16
env.example	Today, 01:07
LICENSE	Today, 01:07
README.Docker.md	Today, 01:16
README.md	Today, 01:07
requirements.txt	Today, 01:07
utils.py	Today, 01:07

4. Run the application

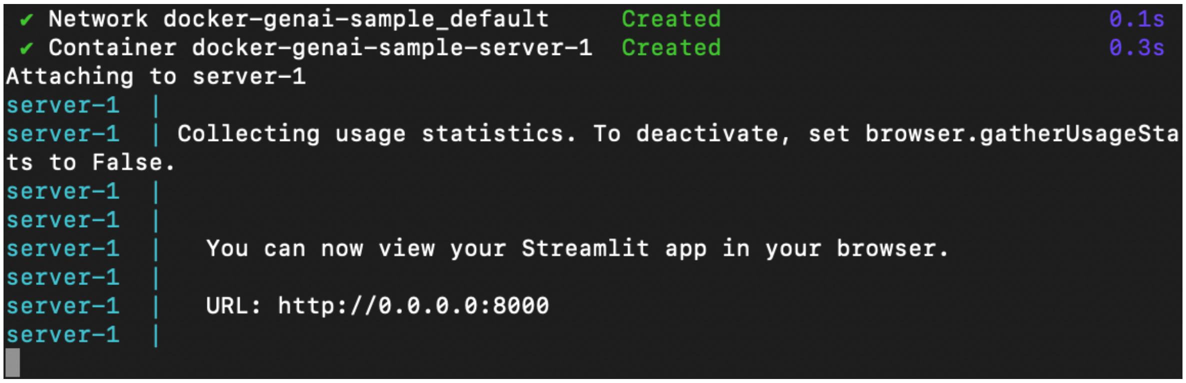
```
% docker compose up –build
```

You will see something like this while Docker is building the application:



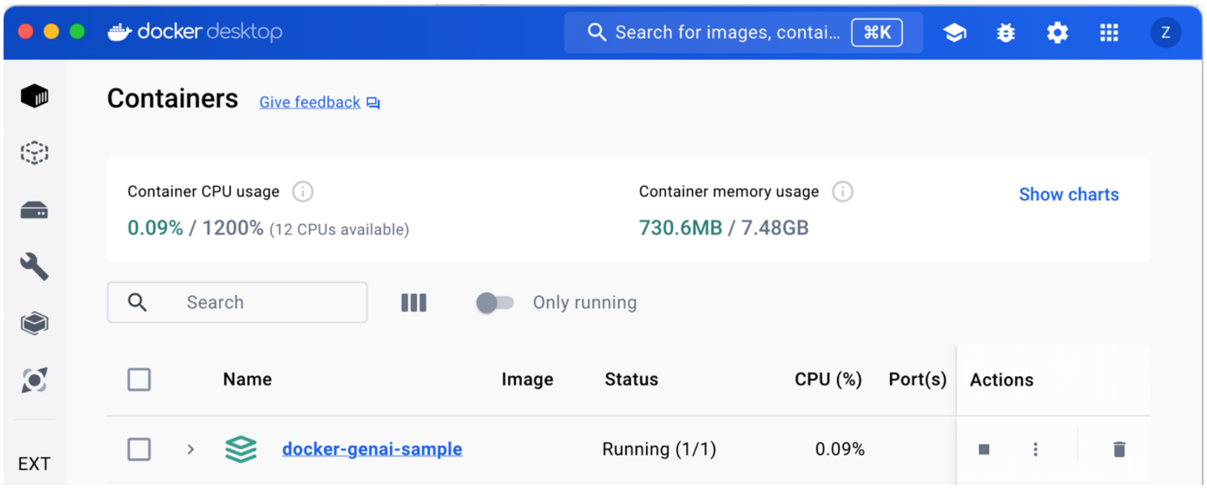
```
(base) zhangzhiyu@shengchligongju docker-genai-sample % docker compose up --build
[+] Building 1511.9s (14/14) FINISHED                          docker:desktop-linux
 => [server internal] load build definition from Dockerfile                   0.1s
 => => transferring dockerfile: 1.71kB                                        0.0s
 => [server] resolve image config for docker.io/docker/dockerfile:1           2.3s
 => [server auth] docker/dockerfile:pull token for registry-1.docker.io       0.0s
 => [server] docker-image://docker.io/docker/dockerfile:1@sha256:ac85f380a6   5.0s
 => => resolve docker.io/docker/dockerfile:1@sha256:ac85f380a63b13dfcefa890   0.0s
 => => sha256:ac85f380a63b13dfcefa89046420e1781752bab202122 8.40kB / 8.40kB   0.0s
 => => sha256:657fcc512c7369f4cb3d94ea329150f8daf626bc838b1a1e8 482B / 482B   0.0s
 => => sha256:a17ee7fff8f5e97b974f5b48f51647d2cf28d543f2aa6 1.27kB / 1.27kB   0.0s
 => => sha256:9d9c93f4b00be908ab694a4df732570bced3b8a96b7 11.80MB / 11.80MB   4.6s
 => => extracting sha256:9d9c93f4b00be908ab694a4df732570bced3b8a96b7515d70f   0.3s
 => [server internal] load metadata for docker.io/library/python:3.11.3-sli   1.5s
 => [server auth] library/python:pull token for registry-1.docker.io          0.0s
 => [server internal] load .dockerignore                                      0.0s
 => => transferring context: 667B                                             0.0s
 => [server base 1/5] FROM docker.io/library/python:3.11.3-slim@sha256:eae   23.7s
 => => resolve docker.io/library/python:3.11.3-slim@sha256:eaee5f73efa9ae96   0.0s
 => => sha256:e77f21686f0b6267e49062c42fdf37aa9b87006eb07f4 6.83kB / 6.83kB   0.0s
 => => sha256:f03b40093957615593f2ed142961afb6b540507e0b 31.40MB / 31.40MB  17.0s
```

Depending on your network connection, it may take several minutes to download all the dependencies. You'll see a message like the following in the terminal when the application is running:



```
 ✔ Network docker-genai-sample_default      Created                           0.1s
 ✔ Container docker-genai-sample-server-1   Created                           0.3s
Attaching to server-1
server-1  |
server-1  | Collecting usage statistics. To deactivate, set browser.gatherUsageSta
ts 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  |
```

You can also see the container running in your Docker Desktop:



Containers

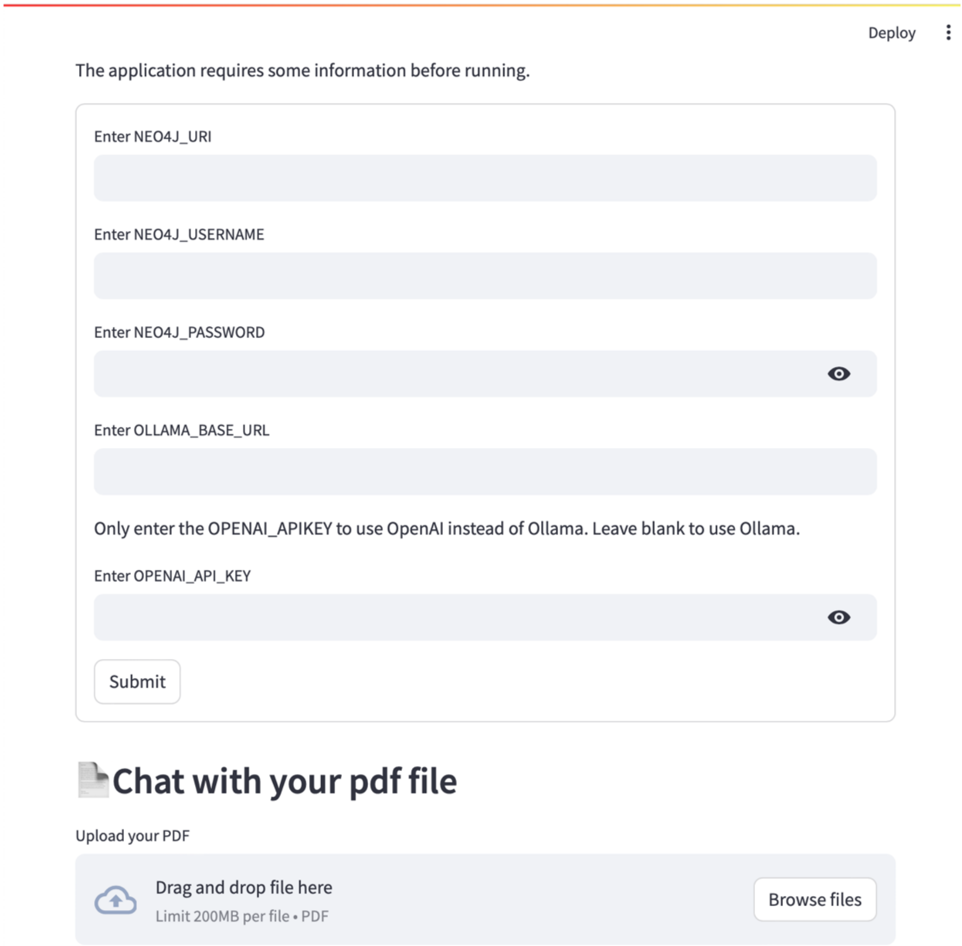
Container CPU usage
0.09% / 1200% (12 CPUs available)

Container memory usage
730.6MB / 7.48GB

		Name	Image	Status	CPU (%)	Port(s)	Actions
□	›	docker-genai-sample		Running (1/1)	0.09%		■ ⋮ 🗑

5. Access the application

Open a browser and view the application at http://localhost:8000:



The application requires some information before running.

Enter NEO4J_URI

Enter NEO4J_USERNAME

Enter NEO4J_PASSWORD

Enter OLLAMA_BASE_URL

Only enter the OPENAI_APIKEY to use OpenAI instead of Ollama. Leave blank to use Ollama.

Enter OPENAI_API_KEY

Chat with your pdf file

Upload your PDF

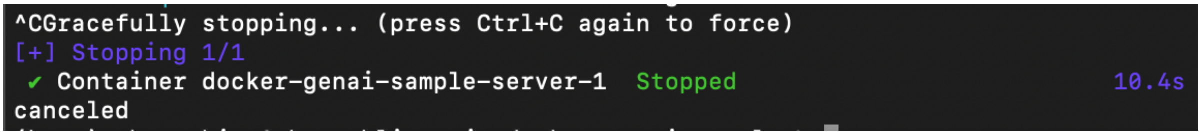
Drag and drop file here

Limit 200MB per file • PDF

To use this application, you'll need a Neo4j database service and an LLM service.

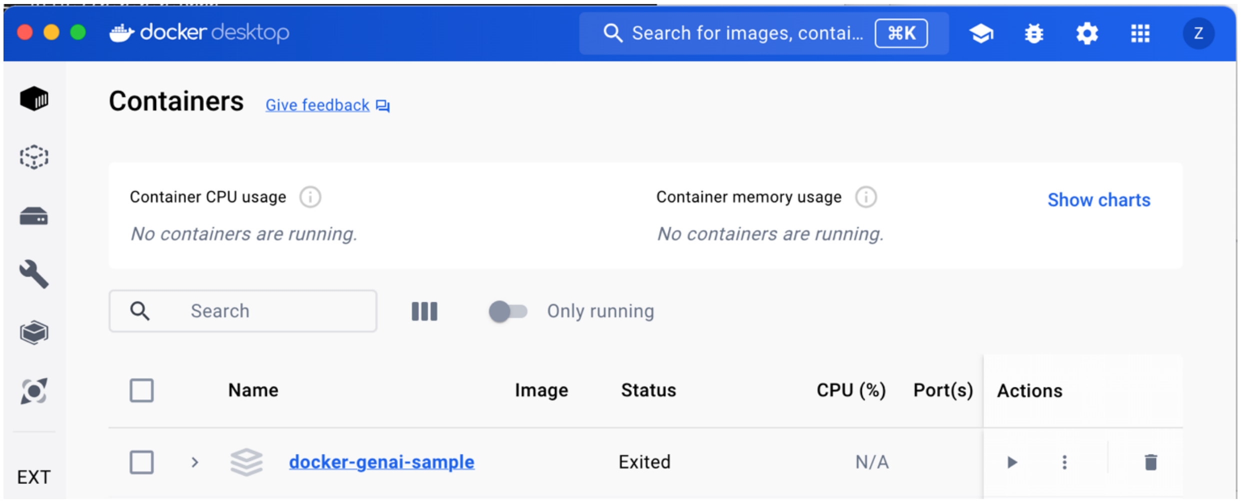
6. Stop the application

In terminal, press ctrl + C to stop the application:



```
^CGracefully stopping... (press Ctrl+C again to force)
[+] Stopping 1/1
 ✔ Container docker-genai-sample-server-1  Stopped                             10.4s
canceled
```

You can also see the container stops running in Docker Desktop:



Containers

Container CPU usage

No containers are running.

Container memory usage

No containers are running.

□	Name	Image	Status	CPU (%)	Port(s)	Actions
□	docker-genai-sample		Exited	N/A		▶ ⋮ 🗑