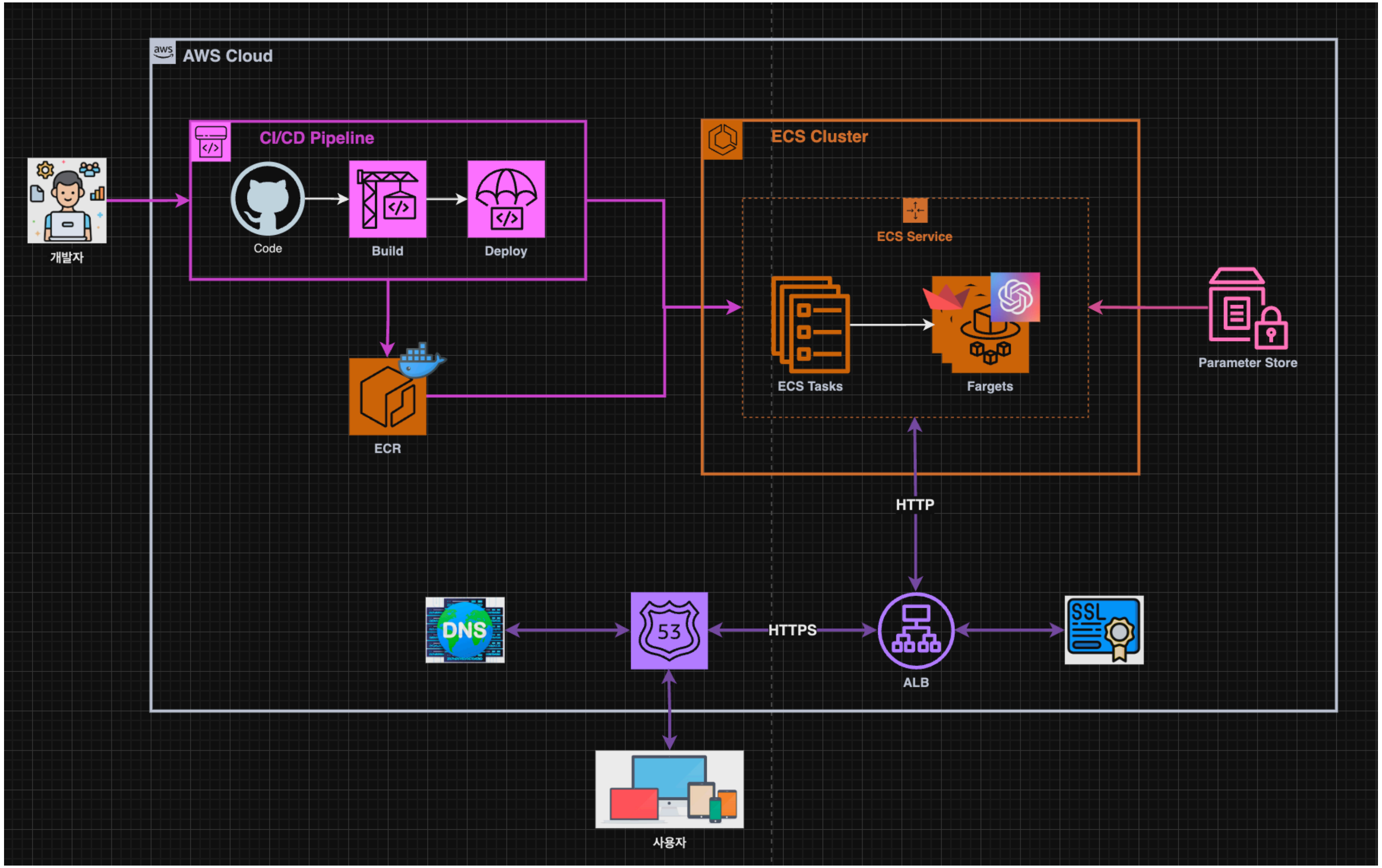
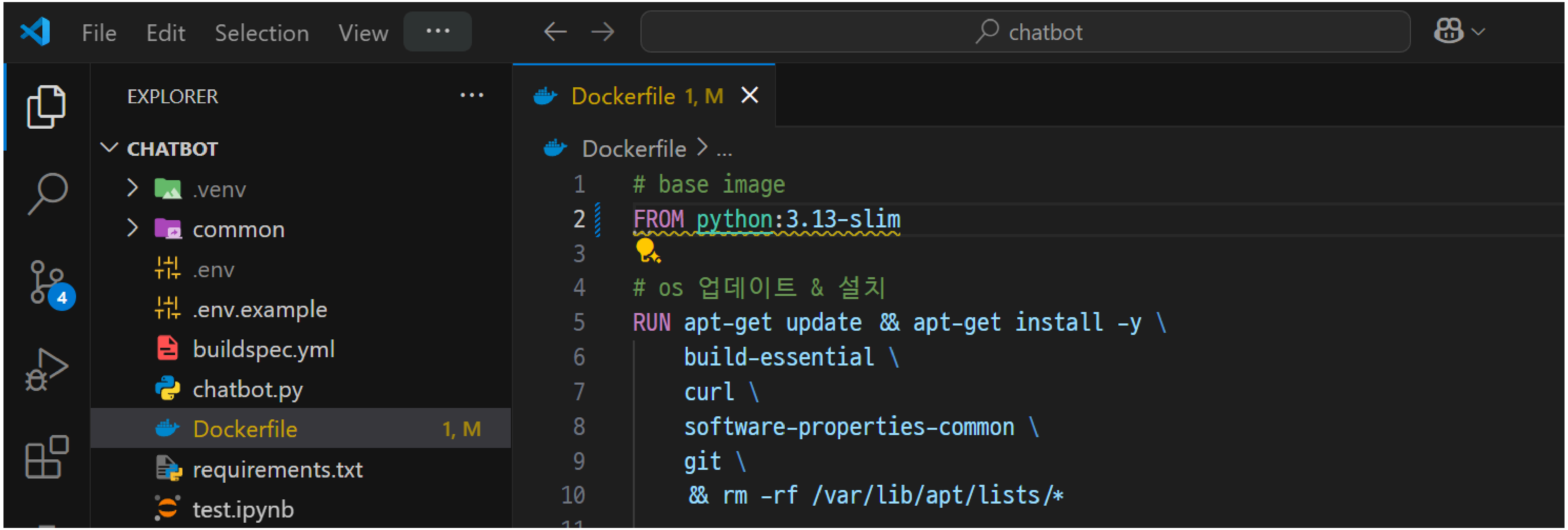
Architecture



ChatBot on Docker

단계1: Dockerfile

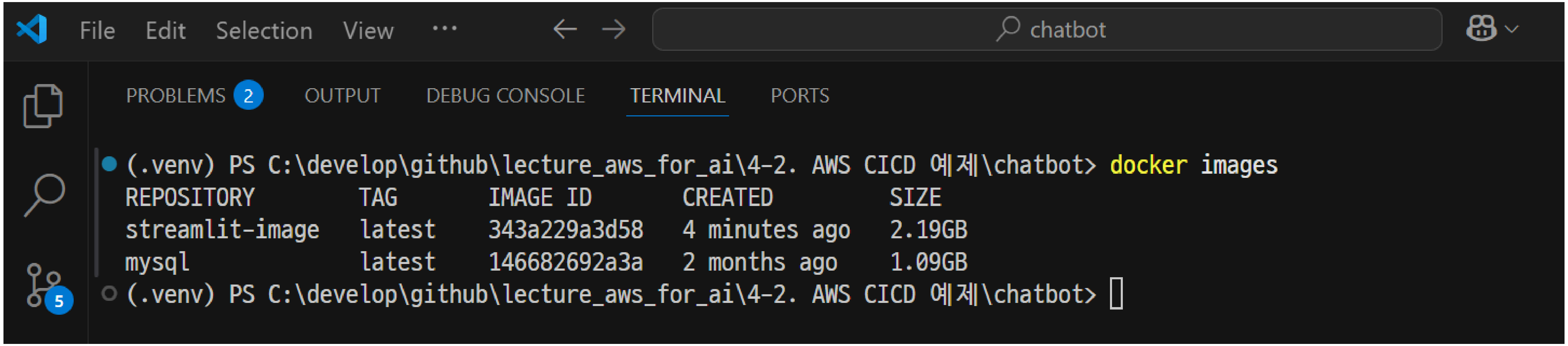


```
# base image
FROM python:3.13-slim

# os 업데이트 & 설치
RUN apt-get update && apt-get install -y \
    build-essential \
    curl \
    software-properties-common \
    git \
    && rm -rf /var/lib/apt/lists/*
```

단계2: Make docker image

- 명령어: `docker build --platform linux/amd64 -t [이미지명] .`

```
# Make docker image
docker build --platform linux/amd64 -t streamlit-image .
docker images # 생성된 이미지 확인
```



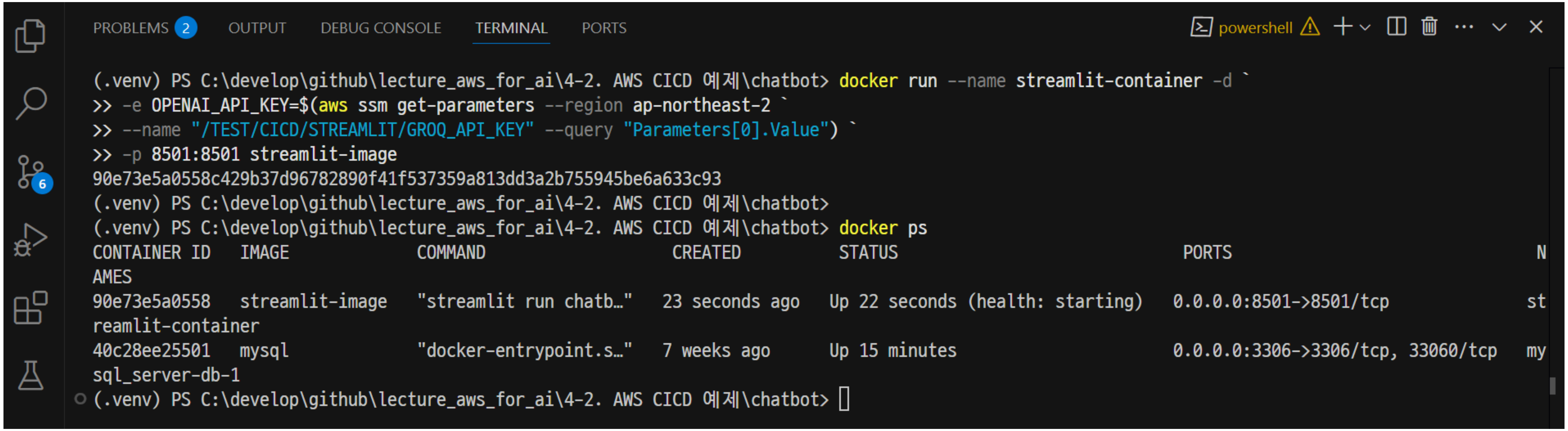
```
(.venv) PS C:\develop\github\lecture_aws_for_ai\4-2. AWS CICD 예제\chatbot> docker images
REPOSITORY        TAG       IMAGE ID       CREATED          SIZE
streamlit-image   latest    343a229a3d58   4 minutes ago    2.19GB
mysql             latest    146682692a3a   2 months ago     1.09GB
(.venv) PS C:\develop\github\lecture_aws_for_ai\4-2. AWS CICD 예제\chatbot>
```

단계3: Run container

- 명령어: `docker run --name [컨테이너명] -d -e [환경변수] -p 8501:8501 [이미지명]`

```
# 리눅스
docker run --name streamlit-container -d \
-e GROQ_API_KEY=$(aws ssm get-parameters --region ap-northeast-2 \
--name "/TEST/CICD/STREAMLIT/GROQ_API_KEY" --query "Parameters[0].Value" | tr -d '"') \
-p 8501:8501 streamlit-image

docker ps # 실행중인 컨테이너 확인
```

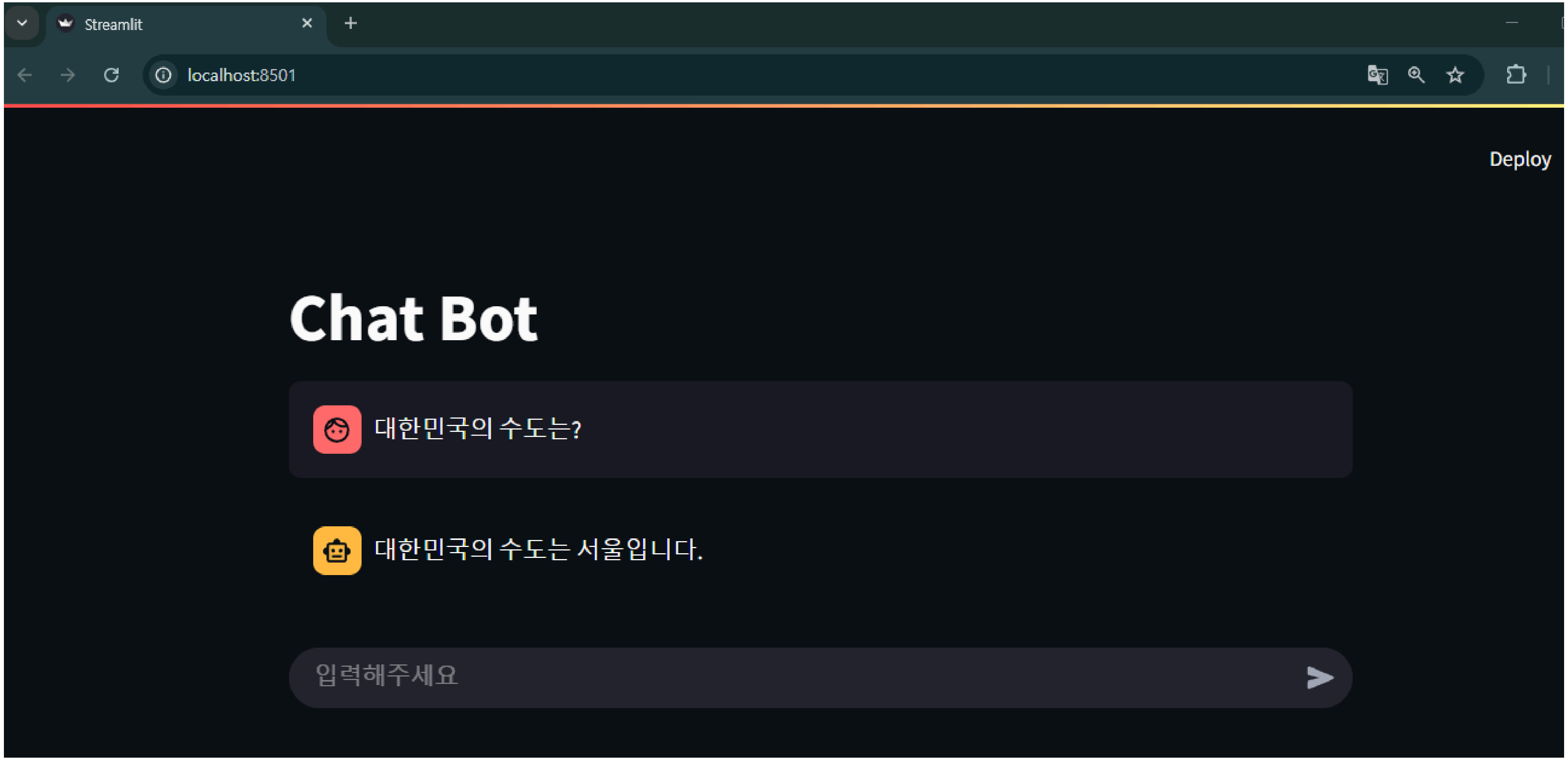
```
# 윈도우
docker run --name streamlit-container -d `
-e GROQ_API_KEY=$(aws ssm get-parameters --region ap-northeast-2 `
--name "/TEST/CICD/STREAMLIT/GROQ_API_KEY" --query "Parameters[0].Value") `
-p 8501:8501 streamlit-image

docker ps # 실행중인 컨테이너 확인
```



```
(.venv) PS C:\develop\github\lecture_aws_for_ai\4-2. AWS CICD 예제\chatbot> docker run --name streamlit-container -d `
>> -e OPENAI_API_KEY=$(aws ssm get-parameters --region ap-northeast-2 `
>> --name "/TEST/CICD/STREAMLIT/GROQ_API_KEY" --query "Parameters[0].Value") `
>> -p 8501:8501 streamlit-image
90e73e5a0558c429b37d96782890f41f537359a813dd3a2b755945be6a633c93
(.venv) PS C:\develop\github\lecture_aws_for_ai\4-2. AWS CICD 예제\chatbot>
(.venv) PS C:\develop\github\lecture_aws_for_ai\4-2. AWS CICD 예제\chatbot> docker ps
CONTAINER ID   IMAGE             COMMAND                  CREATED          STATUS                            PORTS                               N
AMES
90e73e5a0558   streamlit-image   "streamlit run chatb…"   23 seconds ago   Up 22 seconds (health: starting)   0.0.0.0:8501->8501/tcp              st
reamlit-container
40c28ee25501   mysql             "docker-entrypoint.s…"   7 weeks ago      Up 15 minutes                      0.0.0.0:3306->3306/tcp, 33060/tcp   my
sql_server-db-1
(.venv) PS C:\develop\github\lecture_aws_for_ai\4-2. AWS CICD 예제\chatbot>
```

단계4: 접속 및 실행

```
Local URL: http://localhost:8501
```

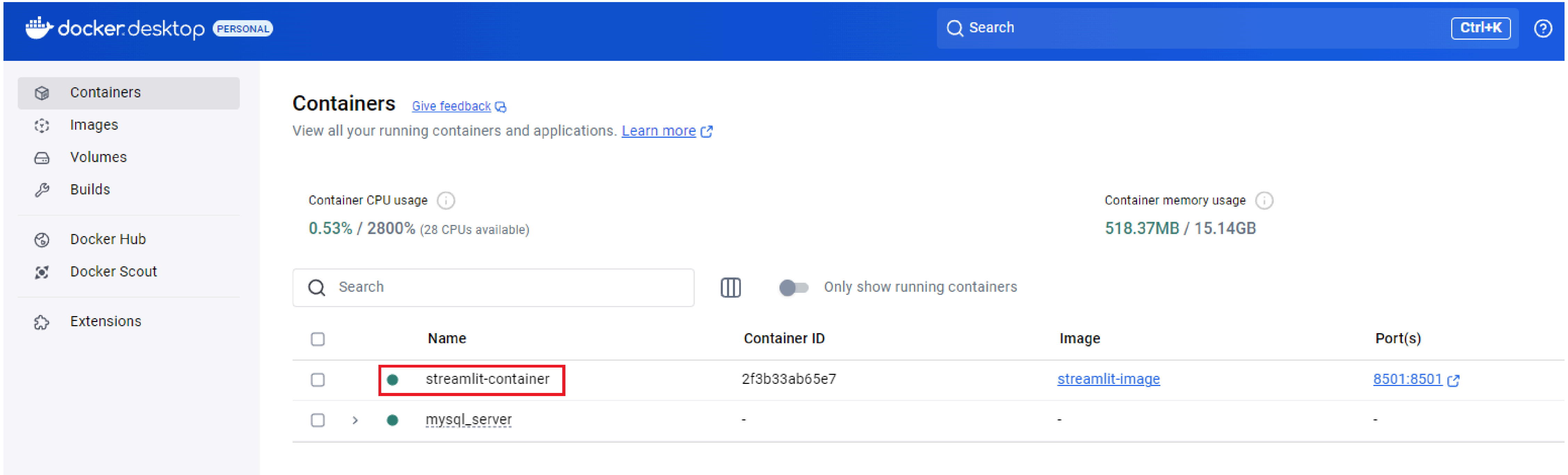


Chat Bot

대한민국의 수도는?

대한민국의 수도는 서울입니다.

단계5: HEALTH CHECK



Containers

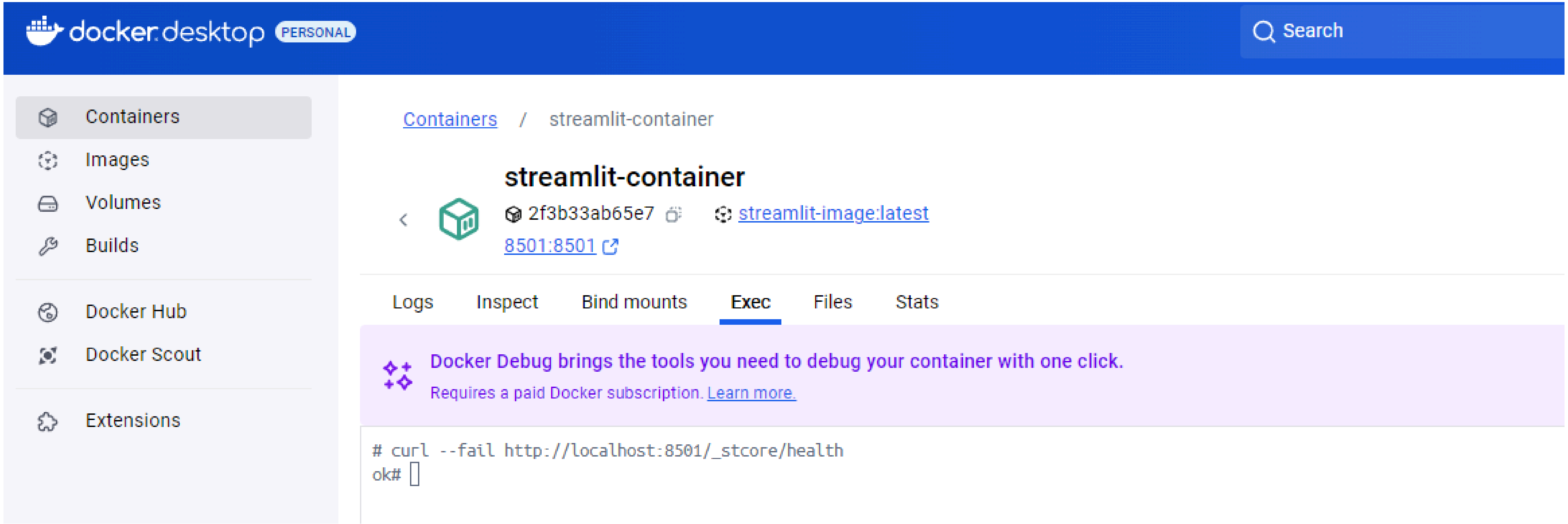
View all your running containers and applications. Learn more

Container CPU usage
0.53% / 2800% (28 CPUs available)

Container memory usage
518.37MB / 15.14GB

□		Name	Container ID	Image	Port(s)
□		● streamlit-container	2f3b33ab65e7	streamlit-image	8501:8501
□	›	● mysql_server	-	-	-

```
curl --fail http://localhost:8501/_stcore/health
```



streamlit-container

2f3b33ab65e7 streamlit-image:latest

8501:8501

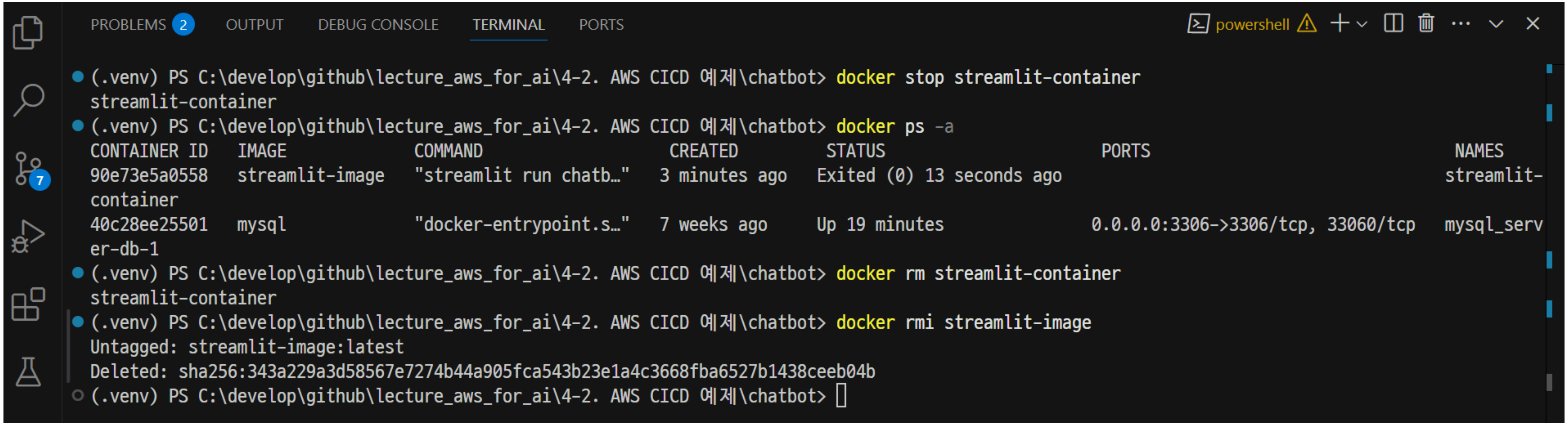
Docker Debug brings the tools you need to debug your container with one click.

Requires a paid Docker subscription. Learn more.

```
# curl --fail http://localhost:8501/_stcore/health
ok# 
```

참고

```
docker stop streamlit-container # 컨테이너 멈춤
docker ps -a # 컨테이너 멈춤 확인
docker rm streamlit-container # 컨테이너 삭제
docker rmi streamlit-image # 이미지 삭제
```



```
● (.venv) PS C:\develop\github\lecture_aws_for_ai\4-2. AWS CICD 예제\chatbot> docker stop streamlit-container
streamlit-container
● (.venv) PS C:\develop\github\lecture_aws_for_ai\4-2. AWS CICD 예제\chatbot> docker ps -a
CONTAINER ID   IMAGE             COMMAND                  CREATED         STATUS                      PORTS                               NAMES
90e73e5a0558   streamlit-image   "streamlit run chatb…"   3 minutes ago   Exited (0) 13 seconds ago                                       streamlit-
container
40c28ee25501   mysql             "docker-entrypoint.s…"   7 weeks ago     Up 19 minutes               0.0.0.0:3306->3306/tcp, 33060/tcp   mysql_serv
er-db-1
● (.venv) PS C:\develop\github\lecture_aws_for_ai\4-2. AWS CICD 예제\chatbot> docker rm streamlit-container
streamlit-container
● (.venv) PS C:\develop\github\lecture_aws_for_ai\4-2. AWS CICD 예제\chatbot> docker rmi streamlit-image
Untagged: streamlit-image:latest
Deleted: sha256:343a229a3d58567e7274b44a905fca543b23e1a4c3668fba6527b1438ceeb04b
○ (.venv) PS C:\develop\github\lecture_aws_for_ai\4-2. AWS CICD 예제\chatbot> 
```