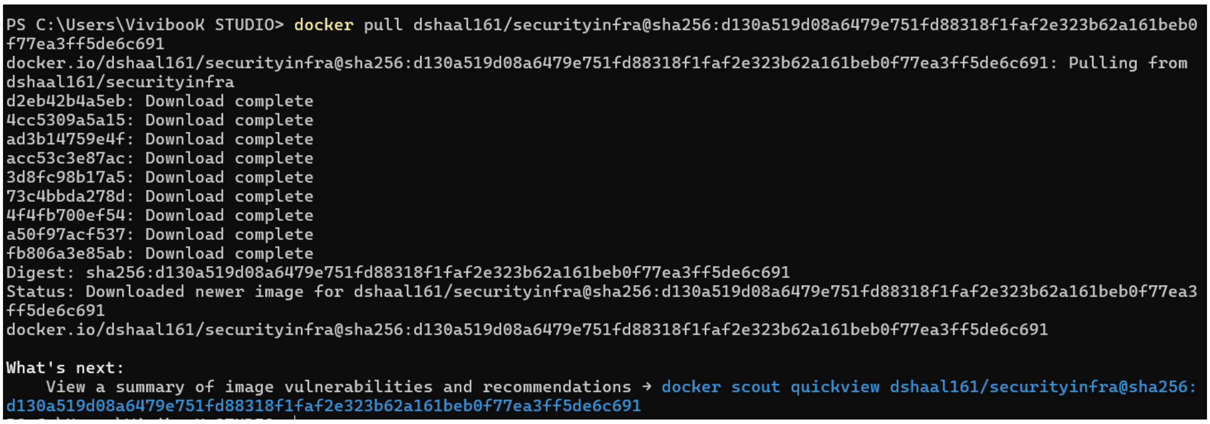
Bug Ticket: Web Application Issues

Reported by: Husney Ali Saleh

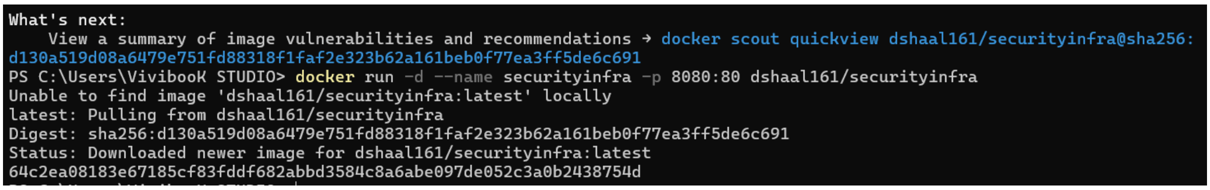
Date: 19/02/2025

Reported to: Daniel Shaal/ Palo Alto Networks Development Team

Pulling the Docker image:



```
PS C:\Users\VivibooK STUDIO> docker pull dshaal161/securityinfra@sha256:d130a519d08a6479e751fd88318f1faf2e323b62a161beb0
f77ea3ff5de6c691
docker.io/dshaal161/securityinfra@sha256:d130a519d08a6479e751fd88318f1faf2e323b62a161beb0f77ea3ff5de6c691: Pulling from
dshaal161/securityinfra
d2eb42b4a5eb: Download complete
4cc5309a5a15: Download complete
ad3b14759e4f: Download complete
acc53c3e87ac: Download complete
3d8fc98b17a5: Download complete
73c4bbda278d: Download complete
4f4fb700ef54: Download complete
a50f97acf537: Download complete
fb806a3e85ab: Download complete
Digest: sha256:d130a519d08a6479e751fd88318f1faf2e323b62a161beb0f77ea3ff5de6c691
Status: Downloaded newer image for dshaal161/securityinfra@sha256:d130a519d08a6479e751fd88318f1faf2e323b62a161beb0f77ea3
ff5de6c691
docker.io/dshaal161/securityinfra@sha256:d130a519d08a6479e751fd88318f1faf2e323b62a161beb0f77ea3ff5de6c691

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview dshaal161/securityinfra@sha256:
d130a519d08a6479e751fd88318f1faf2e323b62a161beb0f77ea3ff5de6c691
```

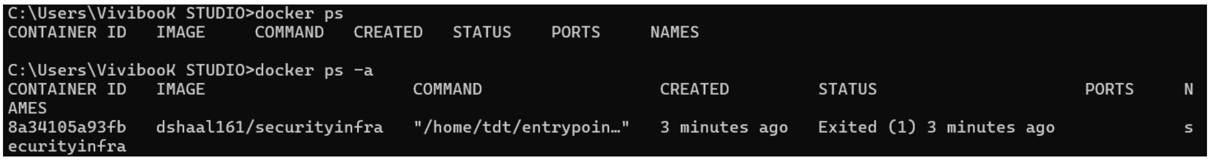


```
What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview dshaal161/securityinfra@sha256:
d130a519d08a6479e751fd88318f1faf2e323b62a161beb0f77ea3ff5de6c691
PS C:\Users\VivibooK STUDIO> docker run -d --name securityinfra -p 8080:80 dshaal161/securityinfra
Unable to find image 'dshaal161/securityinfra:latest' locally
latest: Pulling from dshaal161/securityinfra
Digest: sha256:d130a519d08a6479e751fd88318f1faf2e323b62a161beb0f77ea3ff5de6c691
Status: Downloaded newer image for dshaal161/securityinfra:latest
64c2ea08183e67185cf83fddf682abbd3584c8a6abe097de052c3a0b2438754d
```

Expected: The application is running.

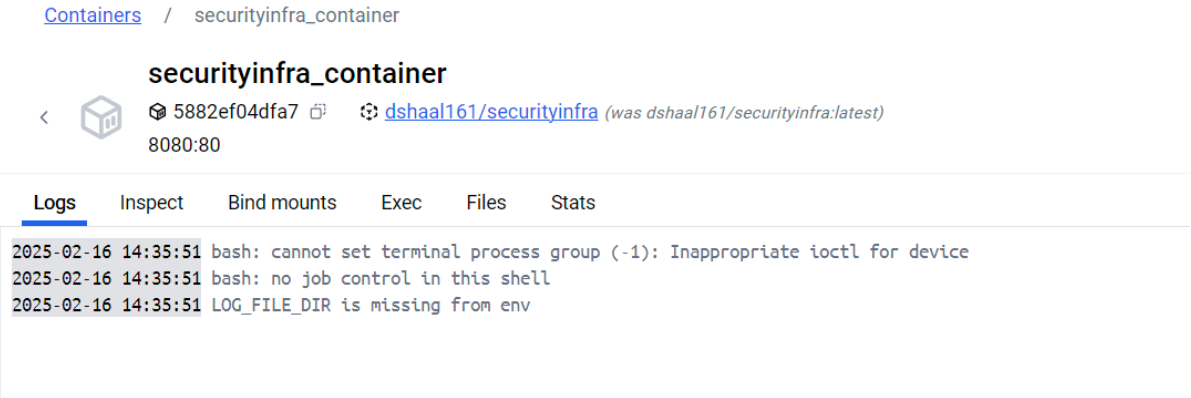
Actual Results:

Didn’t run, so I checked the running containers:



```
C:\Users\VivibooK STUDIO>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

C:\Users\VivibooK STUDIO>docker ps -a
CONTAINER ID   IMAGE                     COMMAND                  CREATED         STATUS                     PORTS     N
AMES
8a34105a93fb   dshaal161/securityinfra   "/home/tdt/entrypoin…"   3 minutes ago   Exited (1) 3 minutes ago             s
ecurityinfra
```

checked the logs:



Containers / securityinfra_container
securityinfra_container
5882ef04dfa7
dshaal161/securityinfra (was dshaal161/securityinfra:latest)
8080:80
Logs
Inspect
Bind mounts
Exec
Files
Stats

```
2025-02-16 14:35:51 bash: cannot set terminal process group (-1): Inappropriate ioctl for device
2025-02-16 14:35:51 bash: no job control in this shell
2025-02-16 14:35:51 LOG_FILE_DIR is missing from env
```

BUG:

“LOG_FILE_DIR is missing from env” means that the Environment variable is missing.

Tried to fix it by changing this:

log_file_dir = os.getenv("LOG_FILE_DIR")

if not log_file_dir:

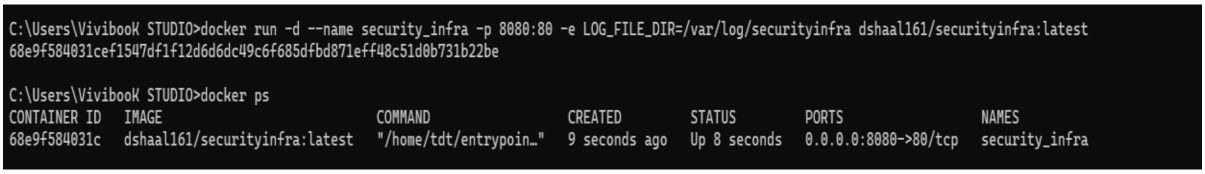
    raise ValueError("Environment variable LOG_FILE_DIR is missing")

To this:
log_file_dir = os.getenv("LOG_FILE_DIR", "/home/tdt/logs")

But it didn't work.

I had to set it manually.

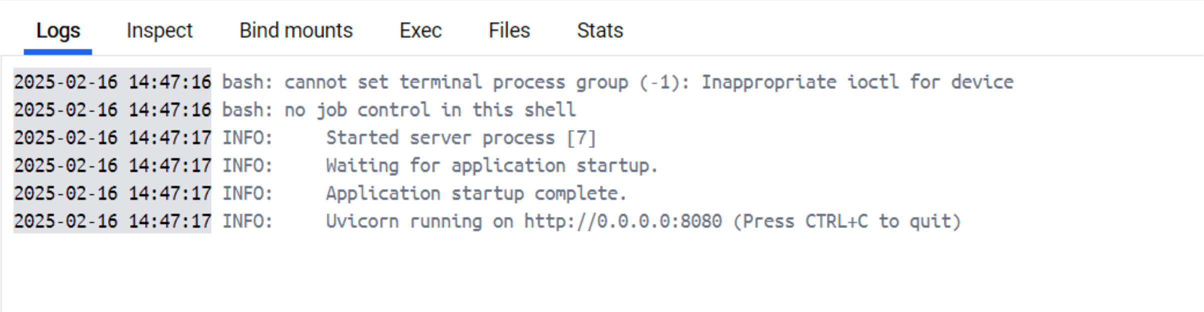
docker run -d -p 8080:8080 --name securityinfra_final -**e LOG_FILE_DIR=/home/tdt/logs**
7usney1802/securityinfra:latest



```
C:\Users\VivibooK STUDIO>docker run -d --name security_infra -p 8080:80 -e LOG_FILE_DIR=/var/log/securityinfra dshaal161/securityinfra:latest
68e9f584031cef1547df1f12d6d6dc49c6f685dfbd871eff48c51d0b731b22be

C:\Users\VivibooK STUDIO>docker ps
CONTAINER ID   IMAGE                            COMMAND                  CREATED         STATUS         PORTS                  NAMES
68e9f584031c   dshaal161/securityinfra:latest   "/home/tdt/entrypoin…"   9 seconds ago   Up 8 seconds   0.0.0.0:8080->80/tcp   security_infra
```

now it's running!

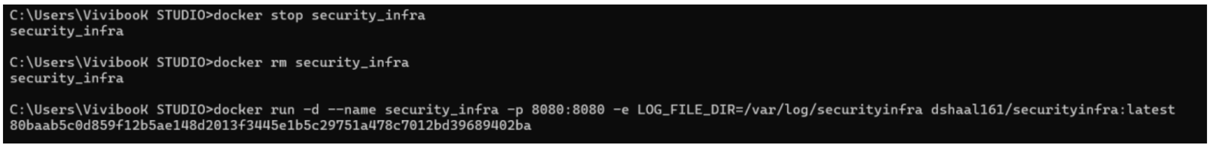
Bug solved!



```
2025-02-16 14:47:16 bash: cannot set terminal process group (-1): Inappropriate ioctl for device
2025-02-16 14:47:16 bash: no job control in this shell
2025-02-16 14:47:17 INFO:     Started server process [7]
2025-02-16 14:47:17 INFO:     Waiting for application startup.
2025-02-16 14:47:17 INFO:     Application startup complete.
2025-02-16 14:47:17 INFO:     Uvicorn running on http://0.0.0.0:8080 (Press CTRL+C to quit)
```

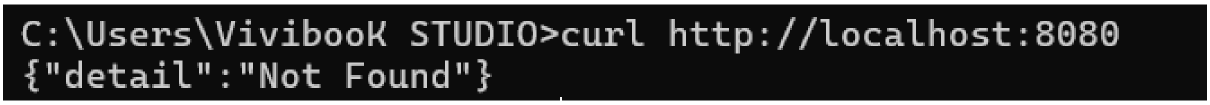
After checking the Logs, we see that it's running, now we should see if it's responding

Had to change the port to 8080:8080.



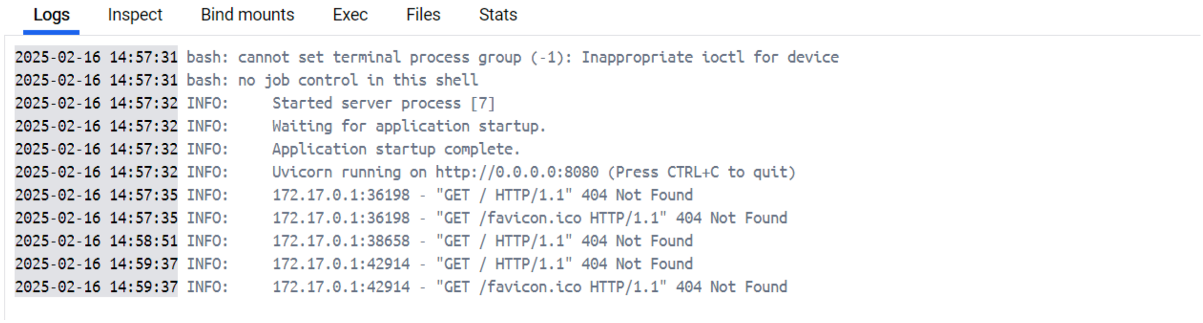
```
C:\Users\VivibooK STUDIO>docker stop security_infra
security_infra

C:\Users\VivibooK STUDIO>docker rm security_infra
security_infra

C:\Users\VivibooK STUDIO>docker run -d --name security_infra -p 8080:8080 -e LOG_FILE_DIR=/var/log/securityinfra dshaal161/securityinfra:latest
80baab5c0d859f12b5ae148d2013f3445e1b5c29751a478c7012bd39689402ba
```

After trying to open the web by the link: localhost:8080
I got this error:



```
C:\Users\VivibooK STUDIO>curl http://localhost:8080
{"detail":"Not Found"}
```

I had to check the logs:



```
2025-02-16 14:57:31 bash: cannot set terminal process group (-1): Inappropriate ioctl for device
2025-02-16 14:57:31 bash: no job control in this shell
2025-02-16 14:57:32 INFO:     Started server process [7]
2025-02-16 14:57:32 INFO:     Waiting for application startup.
2025-02-16 14:57:32 INFO:     Application startup complete.
2025-02-16 14:57:32 INFO:     Uvicorn running on http://0.0.0.0:8080 (Press CTRL+C to quit)
2025-02-16 14:57:35 INFO:     172.17.0.1:36198 - "GET / HTTP/1.1" 404 Not Found
2025-02-16 14:57:35 INFO:     172.17.0.1:36198 - "GET /favicon.ico HTTP/1.1" 404 Not Found
2025-02-16 14:58:51 INFO:     172.17.0.1:38658 - "GET / HTTP/1.1" 404 Not Found
2025-02-16 14:59:37 INFO:     172.17.0.1:42914 - "GET / HTTP/1.1" 404 Not Found
2025-02-16 14:59:37 INFO:     172.17.0.1:42914 - "GET /favicon.ico HTTP/1.1" 404 Not Found
```

BUG:

"GET / HTTP/1.1" 404 Not Found – A request to (/) root route is returning 404, means **Not found**.

"GET /favicon.ico HTTP/1.1" 404 Not Found - A request for `/favicon.ico` also returned **404 Not Found**.

had to check the common API endpoints:
curl http://localhost:8080/api

curl http://localhost:8080/status

curl http://localhost:8080/health

curl http://localhost:8080/check

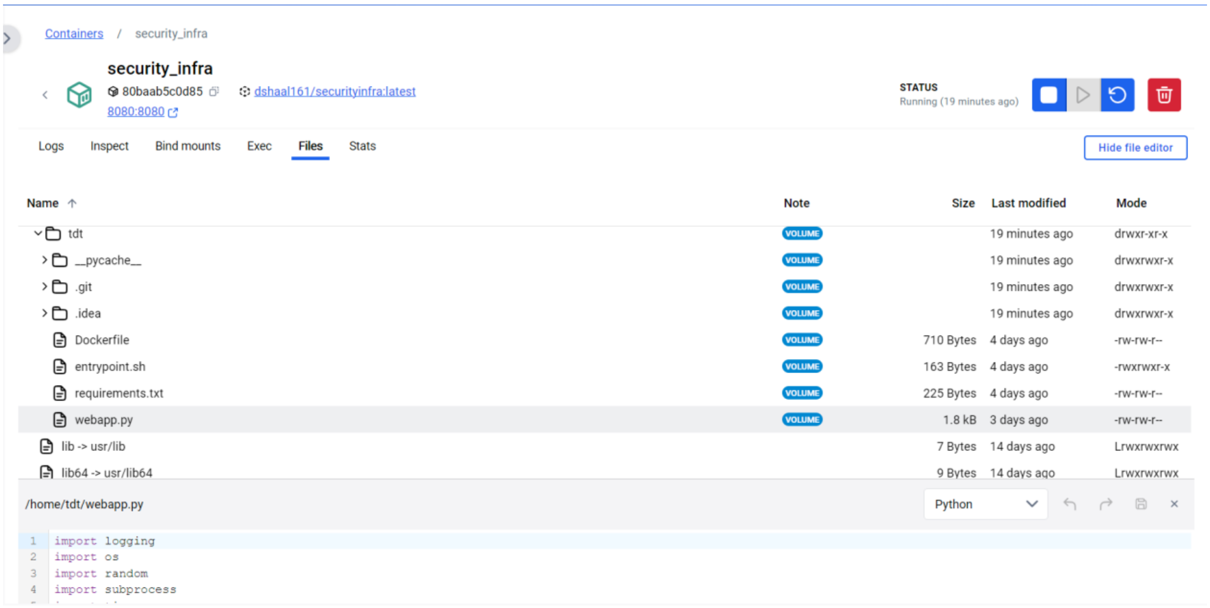
curl http://localhost:8080/v1/status

And got: {"detail":"Not Found"} means that there are no active endpoints, the correct path might be not be loaded properly.

So I entered the Container by **exec** command: docker exec -it security_infra /bin/sh.

Searched for the main file in the docker container:
(I could also searched for it in the cmd(ends with .py): **find / -name "*.py**".)

Found the main python file.
“webapp.py”



Name	Note	Size	Last modified	Mode
tdt	VOLUME		19 minutes ago	drwxr-xr-x
__pycache__	VOLUME		19 minutes ago	drwxrwxr-x
.git	VOLUME		19 minutes ago	drwxrwxr-x
.idea	VOLUME		19 minutes ago	drwxrwxr-x
Dockerfile	VOLUME	710 Bytes	4 days ago	-rw-rw-r--
entrypoint.sh	VOLUME	163 Bytes	4 days ago	-rwxrwxr-x
requirements.txt	VOLUME	225 Bytes	4 days ago	-rw-rw-r--
webapp.py	VOLUME	1.8 kB	3 days ago	-rw-rw-r--
lib -> usr/lib		7 Bytes	14 days ago	Lrwxrwxrwx
lib64 -> usr/lib64		9 Bytes	14 days ago	Lrwxrwxrwx

```python
import logging
import os
import random
import subprocess
```

you can see that it’s path is /home/tdt/webapp.py and it contains the main code.

I had to debug the code to understand what going wrong,

I started debugging the python code:
1)Started checking the execute function(main function), can see that it’s calling for:

result = generate_log().

2)By going to generate_log function we can see that there’s a loop passing on a list of commands.

3)By checking the list of commands we can see 2 things:

Got the error that there’s no such file or directory **file.txt**

A text has been written to file1.txt, but in the end the “cat” command is on file.txt instead of file1.txt

2)Payed attention to the commands(possible error,duplicate commands)

So I changed the command:
"cat /tmp/test-dir/**file**.txt"

To:

"cat /tmp/test-dir/**file1**.txt"

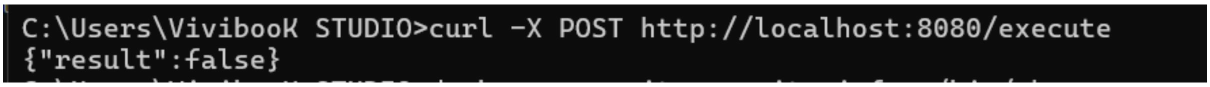
Tried to run the code to check if I have solved the error:

curl –X POST http://localhost:8080/execute
(The curl command sends HTTP requests , -X POST specifies the HTTP method as POST as in the code)

Expected: Bug solved!

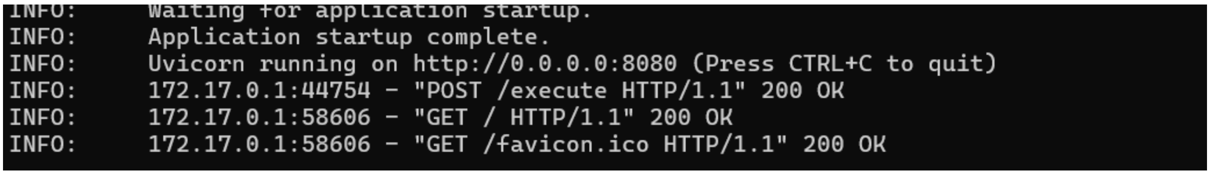
Actual Results:

But still having this:



```
C:\Users\VivibooK STUDIO>curl -X POST http://localhost:8080/execute
{"result":false}
```

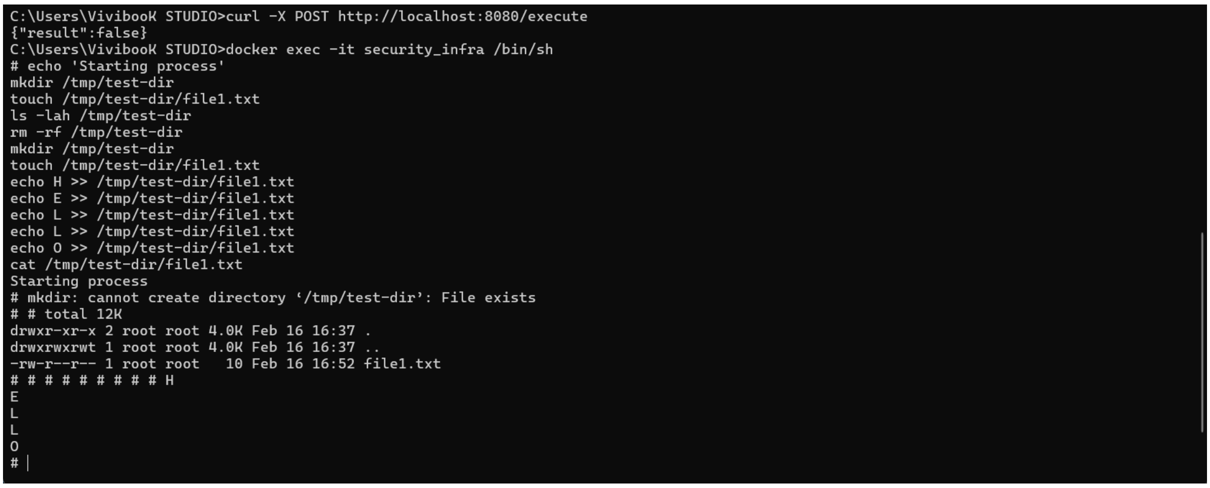
checked the logs by the command ”docker logs security_infra”, got no errors here:



```
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8080 (Press CTRL+C to quit)
INFO:     172.17.0.1:44754 - "POST /execute HTTP/1.1" 200 OK
INFO:     172.17.0.1:58606 - "GET / HTTP/1.1" 200 OK
INFO:     172.17.0.1:58606 - "GET /favicon.ico HTTP/1.1" 200 OK
```

had to run the command:

Docker exec –it security_infra /bin/sh to get inside the container
and runned the command #echo ‘starting process’

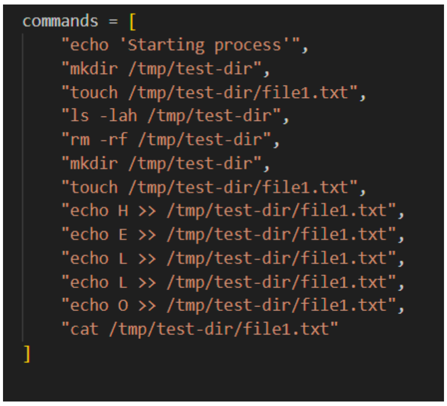


```
C:\Users\VivibooK STUDIO>curl -X POST http://localhost:8080/execute
{"result":false}
C:\Users\VivibooK STUDIO>docker exec -it security_infra /bin/sh
# echo 'Starting process'
mkdir /tmp/test-dir
touch /tmp/test-dir/file1.txt
ls -lah /tmp/test-dir
rm -rf /tmp/test-dir
mkdir /tmp/test-dir
touch /tmp/test-dir/file1.txt
echo H >> /tmp/test-dir/file1.txt
echo E >> /tmp/test-dir/file1.txt
echo L >> /tmp/test-dir/file1.txt
echo L >> /tmp/test-dir/file1.txt
echo O >> /tmp/test-dir/file1.txt
cat /tmp/test-dir/file1.txt
Starting process
# mkdir: cannot create directory ‘/tmp/test-dir’: File exists
# # total 12K
drwxr-xr-x 2 root root 4.0K Feb 16 16:37 .
drwxrwxrwt 1 root root 4.0K Feb 16 16:37 ..
-rw-r--r-- 1 root root   10 Feb 16 16:52 file1.txt
# # # # # # # # # H
E
L
L
O
# |
```

BUG:

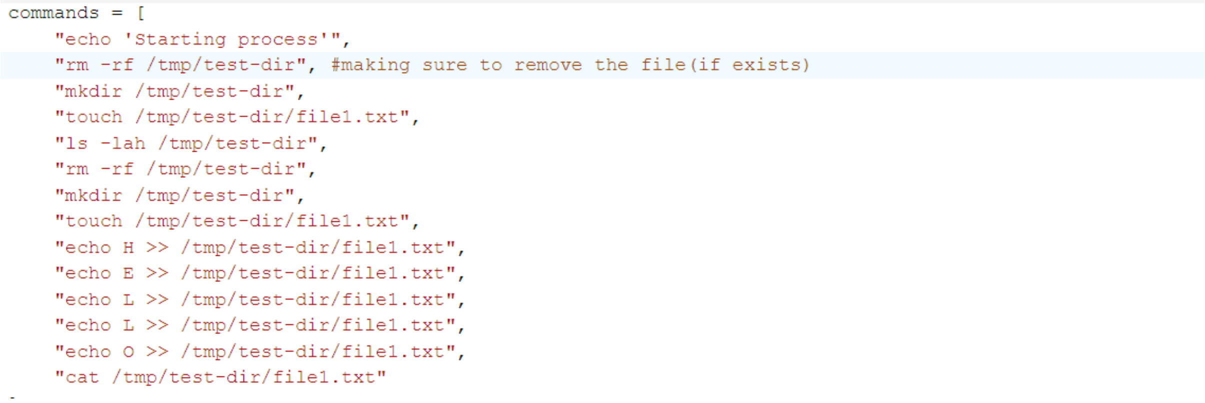
mkdir: cannot create directory ‘/tmp/test-dir’: File exists

had to make sure removing the existing file before the mkdir command so I changed the command from this:



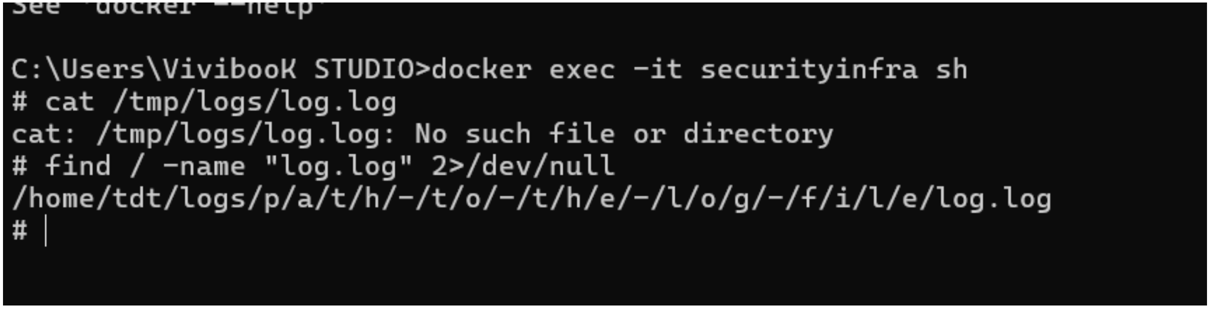
```
commands = [
    "echo 'Starting process'",
    "mkdir /tmp/test-dir",
    "touch /tmp/test-dir/file1.txt",
    "ls -lah /tmp/test-dir",
    "rm -rf /tmp/test-dir",
    "mkdir /tmp/test-dir",
    "touch /tmp/test-dir/file1.txt",
    "echo H >> /tmp/test-dir/file1.txt",
    "echo E >> /tmp/test-dir/file1.txt",
    "echo L >> /tmp/test-dir/file1.txt",
    "echo L >> /tmp/test-dir/file1.txt",
    "echo O >> /tmp/test-dir/file1.txt",
    "cat /tmp/test-dir/file1.txt"
]
```

To this:
(adding “rm –rf /tmp/test-dir”,)



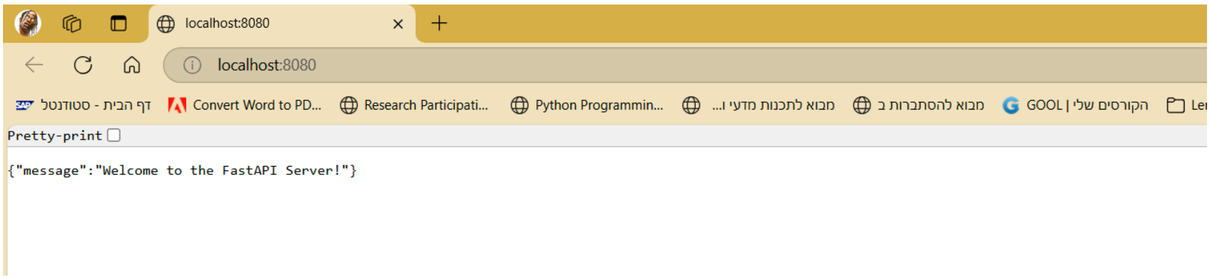
```
commands = [
    "echo 'Starting process'",
    "rm -rf /tmp/test-dir", #making sure to remove the file(if exists)
    "mkdir /tmp/test-dir",
    "touch /tmp/test-dir/file1.txt",
    "ls -lah /tmp/test-dir",
    "rm -rf /tmp/test-dir",
    "mkdir /tmp/test-dir",
    "touch /tmp/test-dir/file1.txt",
    "echo H >> /tmp/test-dir/file1.txt",
    "echo E >> /tmp/test-dir/file1.txt",
    "echo L >> /tmp/test-dir/file1.txt",
    "echo L >> /tmp/test-dir/file1.txt",
    "echo O >> /tmp/test-dir/file1.txt",
    "cat /tmp/test-dir/file1.txt"
```

Note : While working and debugging I found the log file:
All the logs are written inside of this log file.
(I know that there’s another one, but hasn’t been used)

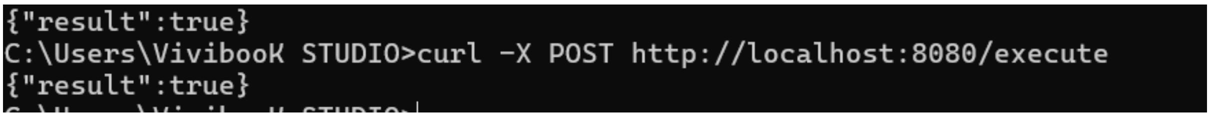


```
See 'docker --help'

C:\Users\VivibooK STUDIO>docker exec -it securityinfra sh
# cat /tmp/logs/log.log
cat: /tmp/logs/log.log: No such file or directory
# find / -name "log.log" 2>/dev/null
/home/tdt/logs/p/a/t/h/-/t/o/-/t/h/e/-/l/o/g/-/f/i/l/e/log.log
# |
```

After testing the corrected code:



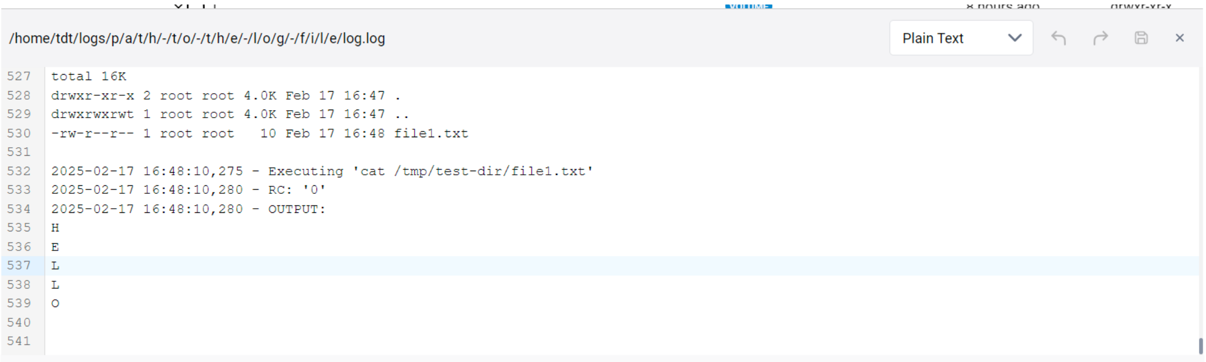
```
{"message":"Welcome to the FastAPI Server!"}
```



```
{"result":true}
C:\Users\VivibooK STUDIO>curl -X POST http://localhost:8080/execute
{"result":true}
```

We can see that the execute function is returning True, means that all the bugs have been solved.

To make sure, I have checked the log file to see if there's any errors:



/home/tdt/logs/p/a/t/h/-/t/o/-/t/h/e/-/l/o/g/-/f/i/l/e/log.log

```
total 16K
drwxr-xr-x 2 root root 4.0K Feb 17 16:47 .
drwxrwxrwt 1 root root 4.0K Feb 17 16:47 ..
-rw-r--r-- 1 root root   10 Feb 17 16:48 file1.txt

2025-02-17 16:48:10,275 - Executing 'cat /tmp/test-dir/file1.txt'
2025-02-17 16:48:10,280 - RC: '0'
2025-02-17 16:48:10,280 - OUTPUT:
H
E
L
L
O

```

Expected:

The app is running and the log file is printing :
H
E
L
L
O

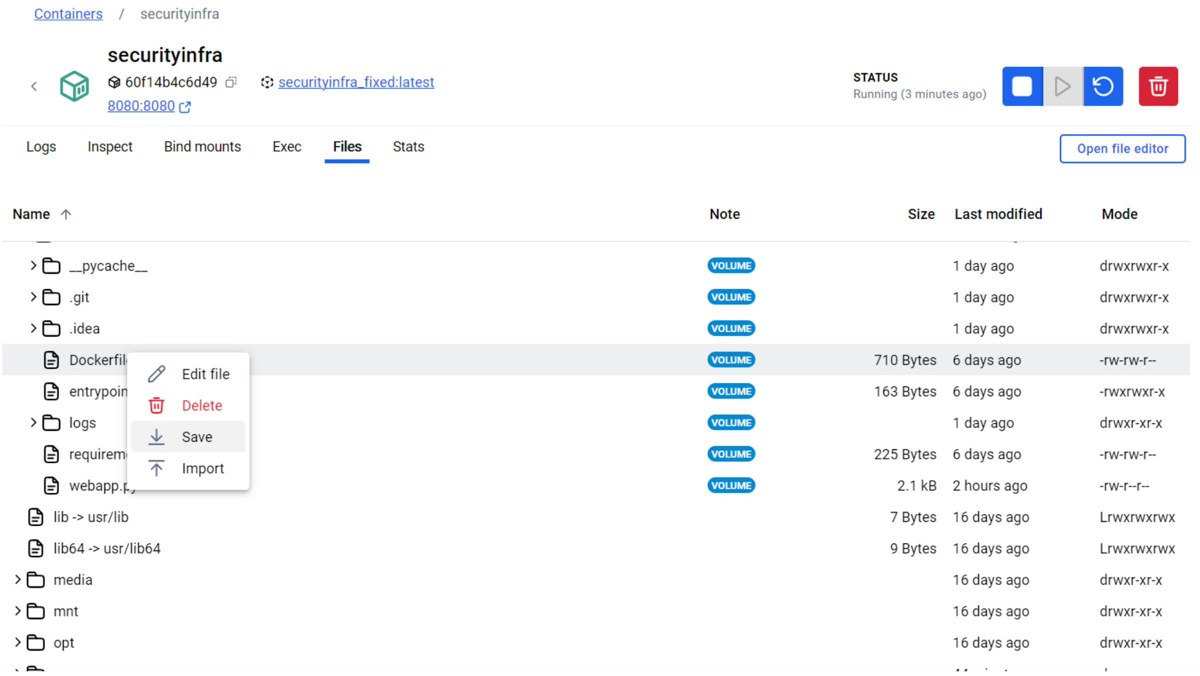
Actual Reality:

Everything looks fine, it's returning and printing as expected.

Rebuild the image:

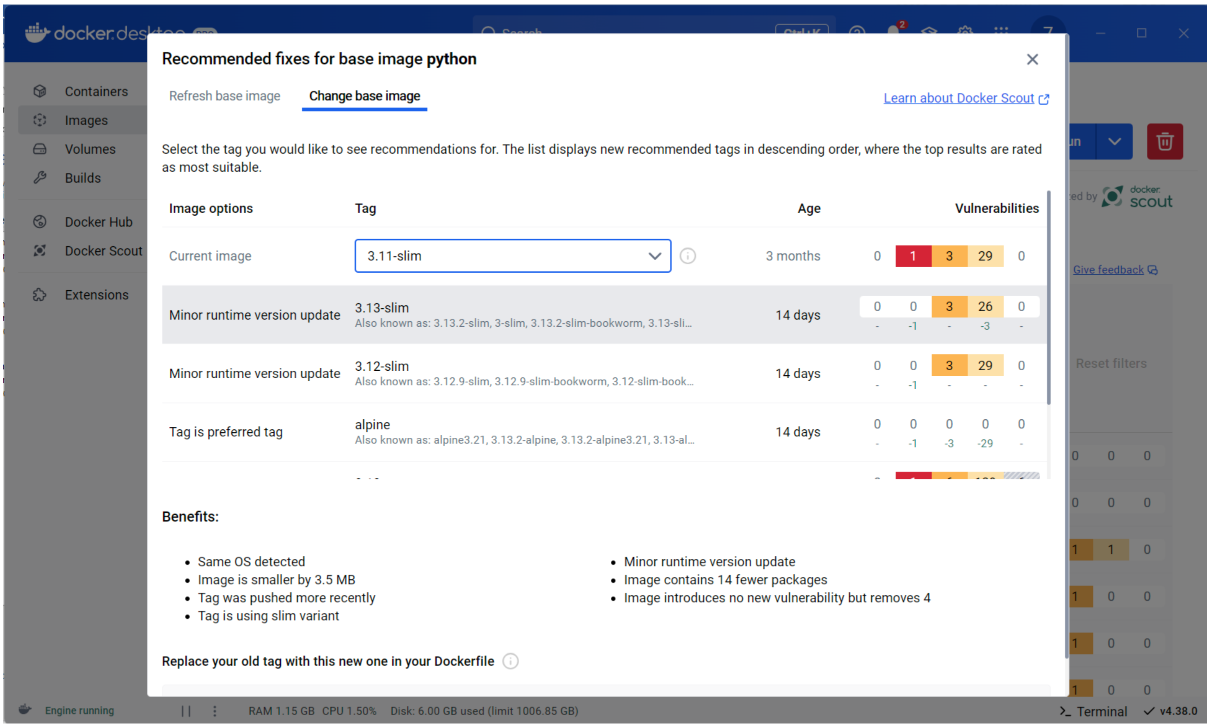
Created a folder called Dockerfile in the desktop

Saved the Dockerfile inside the folder



Name	Note	Size	Last modified	Mode
__pycache__	VOLUME		1 day ago	drwxrwxr-x
.git	VOLUME		1 day ago	drwxrwxr-x
.idea	VOLUME		1 day ago	drwxrwxr-x
Dockerfil	VOLUME	710 Bytes	6 days ago	-rw-rw-r--
entrypoin	VOLUME	163 Bytes	6 days ago	-rwxrwxr-x
logs	VOLUME		1 day ago	drwxr-xr-x
requirem	VOLUME	225 Bytes	6 days ago	-rw-rw-r--
webapp.p	VOLUME	2.1 kB	2 hours ago	-rw-r--r--
lib -> usr/lib		7 Bytes	16 days ago	Lrwxrwxrwx
lib64 -> usr/lib64		9 Bytes	16 days ago	Lrwxrwxrwx
media			16 days ago	drwxr-xr-x
mnt			16 days ago	drwxr-xr-x
opt			16 days ago	drwxr-xr-x

Changed the base image from(3.11) -slim :



Recommended fixes for base image python

Select the tag you would like to see recommendations for. The list displays new recommended tags in descending order, where the top results are rated as most suitable.

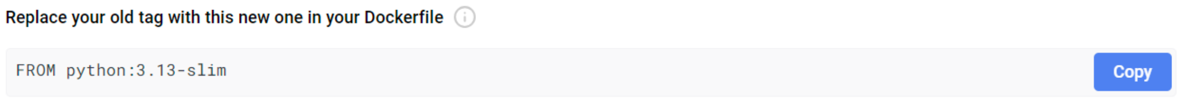
Image options	Tag	Age	Vulnerabilities				
Current image	3.11-slim	3 months	0	1	3	29	0
Minor runtime version update	3.13-slim Also known as: 3.13.2-slim, 3-slim, 3.13.2-slim-bookworm, 3.13-sli...	14 days	0 -	0 -1	3 -	26 -3	0 -
Minor runtime version update	3.12-slim Also known as: 3.12.9-slim, 3.12.9-slim-bookworm, 3.12-slim-book...	14 days	0 -	0 -1	3 -	29 -	0 -
Tag is preferred tag	alpine Also known as: alpine3.21, 3.13.2-alpine, 3.13.2-alpine3.21, 3.13-al...	14 days	0 -	0 -1	0 -3	0 -29	0 -

Benefits:

- Same OS detected
- Image is smaller by 3.5 MB
- Tag was pushed more recently
- Tag is using slim variant
- Minor runtime version update
- Image contains 14 fewer packages
- Image introduces no new vulnerability but removes 4

Replace your old tag with this new one in your Dockerfile

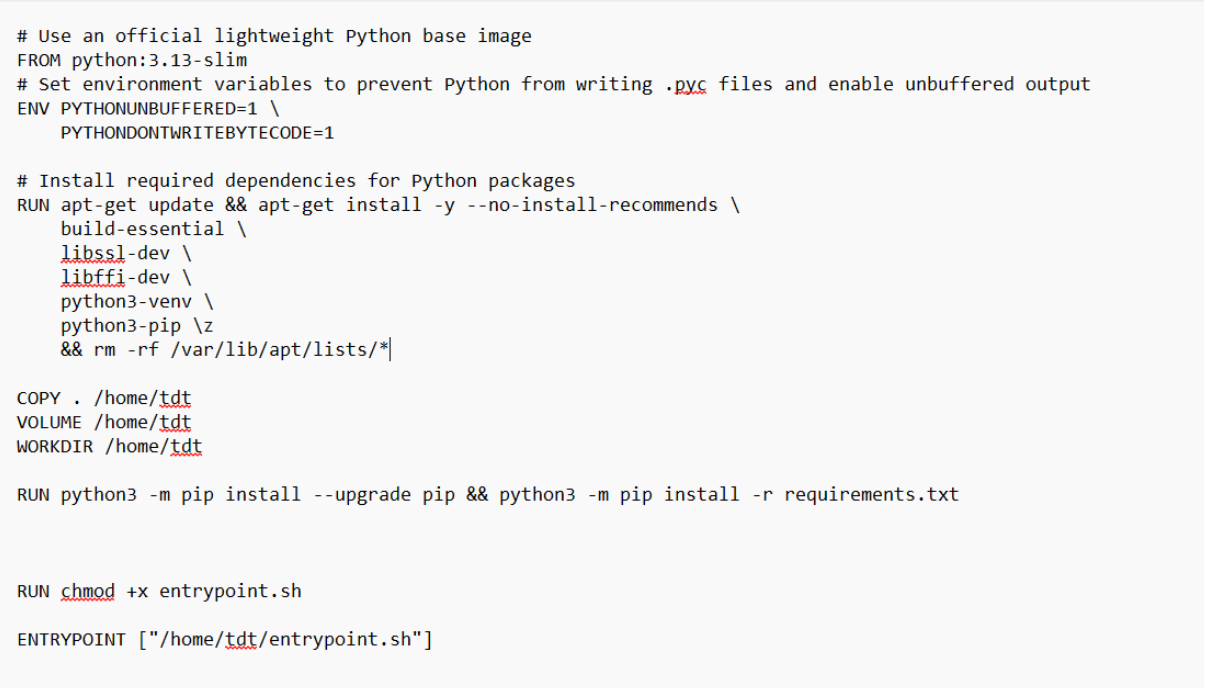
To(as recommended in the dockerfile recommended fixes):



Replace your old tag with this new one in your Dockerfile

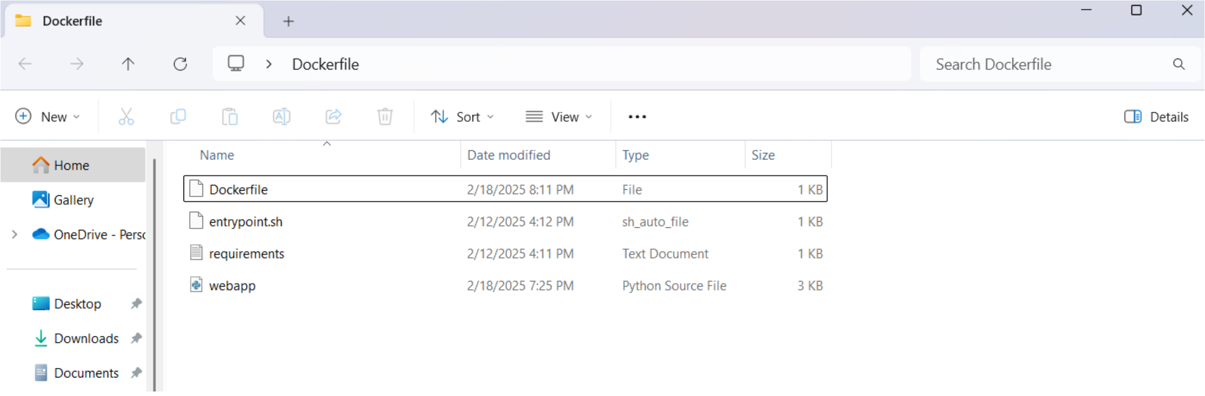
```
FROM python:3.13-slim
```

The Dockerfile after changing the second line:



```
# Use an official lightweight Python base image
FROM python:3.13-slim
# Set environment variables to prevent Python from writing .pyc files and enable unbuffered output
ENV PYTHONUNBUFFERED=1 \
    PYTHONDONTWRITEBYTECODE=1

# Install required dependencies for Python packages
RUN apt-get update && apt-get install -y --no-install-recommends \
    build-essential \
    libssl-dev \
    libffi-dev \
    python3-venv \
    python3-pip \z
    && rm -rf /var/lib/apt/lists/*

COPY . /home/tdt
VOLUME /home/tdt
WORKDIR /home/tdt

RUN python3 -m pip install --upgrade pip && python3 -m pip install -r requirements.txt



RUN chmod +x entrypoint.sh

ENTRYPOINT ["/home/tdt/entrypoint.sh"]
```

Added also to entrypoint.sh, requerements.txt,webapp.py files.(after getting an error that requirements.txt doesn't exist and the container crashed):



Name	Date modified	Type	Size
Dockerfile	2/18/2025 8:11 PM	File	1 KB
entrypoint.sh	2/12/2025 4:12 PM	sh_auto_file	1 KB
requirements	2/12/2025 4:11 PM	Text Document	1 KB
webapp	2/18/2025 7:25 PM	Python Source File	3 KB

And runned these commands:

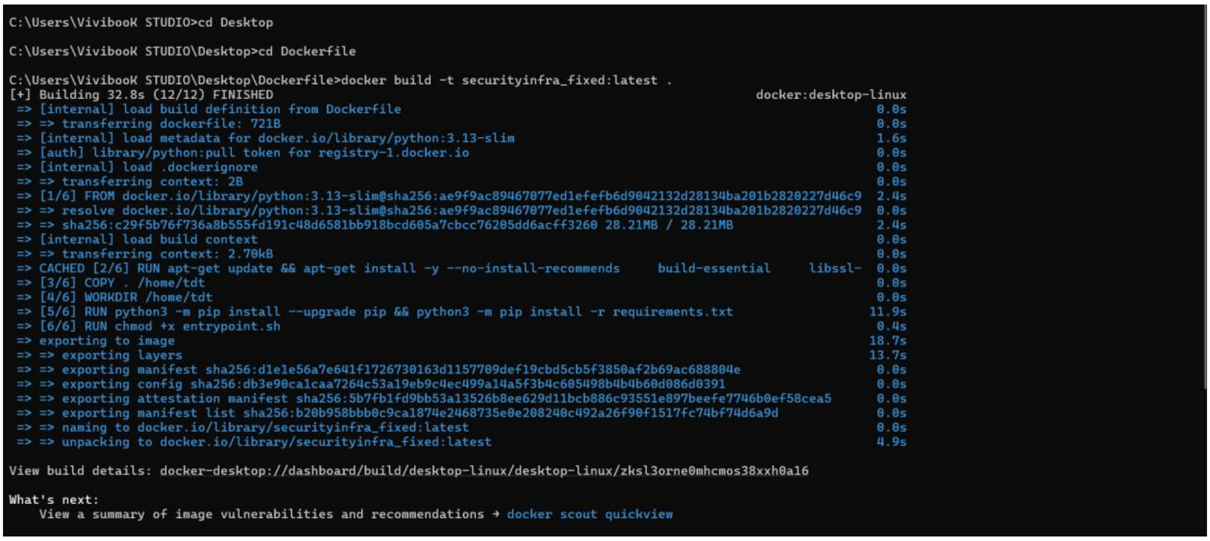
Cd Desktop(to enter the desktop)

Cd Dockerfile(to enter the dockerfile that located in the desktop

Docker build –t securityinfra_fixed:latest . (Building the updated image)

Expected:
Successfully Building the image.

Actual Results:



```
C:\Users\VivibooK STUDIO>cd Desktop

C:\Users\VivibooK STUDIO\Desktop>cd Dockerfile

C:\Users\VivibooK STUDIO\Desktop\Dockerfile>docker build -t securityinfra_fixed:latest .
[+] Building 32.8s (12/12) FINISHED                                                              docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                             0.0s
 => => transferring dockerfile: 721B                                                                             0.0s
 => [internal] load metadata for docker.io/library/python:3.13-slim                                              1.6s
 => [auth] library/python:pull token for registry-1.docker.io                                                    0.0s
 => [internal] load .dockerignore                                                                                0.0s
 => => transferring context: 2B                                                                                  0.0s
 => [1/6] FROM docker.io/library/python:3.13-slim@sha256:ae9f9ac89467077ed1efefb6d9042132d28134ba201b2820227d46c9  2.4s
 => => resolve docker.io/library/python:3.13-slim@sha256:ae9f9ac89467077ed1efefb6d9042132d28134ba201b2820227d46c9  0.0s
 => => sha256:c29f5b76f736a8b555fd191c48d6581bb918bcd605a7cbcc76205dd6acff3260 28.21MB / 28.21MB                  2.4s
 => [internal] load build context                                                                                0.0s
 => => transferring context: 2.70kB                                                                              0.0s
 => CACHED [2/6] RUN apt-get update && apt-get install -y --no-install-recommends     build-essential     libssl-  0.0s
 => [3/6] COPY . /home/tdt                                                                                       0.0s
 => [4/6] WORKDIR /home/tdt                                                                                      0.0s
 => [5/6] RUN python3 -m pip install --upgrade pip && python3 -m pip install -r requirements.txt                 11.9s
 => [6/6] RUN chmod +x entrypoint.sh                                                                             0.4s
 => exporting to image                                                                                          18.7s
 => => exporting layers                                                                                         13.7s
 => => exporting manifest sha256:d1e1e56a7e641f1726730163d1157709def19cbd5cb5f3850af2b69ac688804e                 0.0s
 => => exporting config sha256:db3e90ca1caa7264c53a19eb9c4ec499a14a5f3b4c605498b4b4b60d086d0391                   0.0s
 => => exporting attestation manifest sha256:5b7fb1fd9bb53a13526b8ee629d11bcb886c93551e897beefe7746b0ef58cea5     0.0s
 => => exporting manifest list sha256:b20b958bbb0c9ca1874e2468735e0e208240c492a26f90f1517fc74bf74d6a9d            0.0s
 => => naming to docker.io/library/securityinfra_fixed:latest                                                    0.0s
 => => unpacking to docker.io/library/securityinfra_fixed:latest                                                 4.9s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/zksl3orne0mhcmos38xxh0a16

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview
```

Finished building the image.

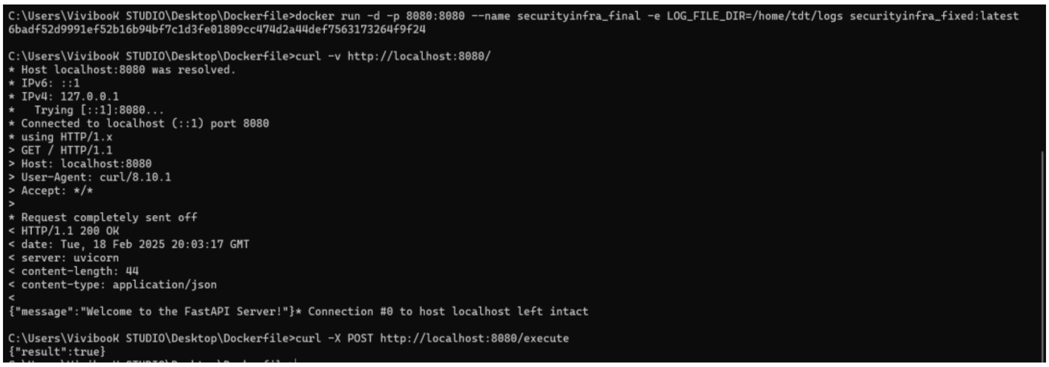
After building the image, had to run the new container using the updated image and tested if it’s working using the commands:

docker run -d -p 8080:8080 --name securityinfra_final -e LOG_FILE_DIR=/home/tdt/logs securityinfra_fixed:latest

curl –X POST http://localhost:8080/execute

Expected:
{“result” :true}

Acutal Result:



```
C:\Users\VivibooK STUDIO\Desktop\Dockerfile>docker run -d -p 8080:8080 --name securityinfra_final -e LOG_FILE_DIR=/home/tdt/logs securityinfra_fixed:latest
6badf52d9991ef52b16b94bf7c1d3fe01809cc474d2a44def7563173264f9f24

C:\Users\VivibooK STUDIO\Desktop\Dockerfile>curl -v http://localhost:8080/
* Host localhost:8080 was resolved.
* IPv6: ::1
* IPv4: 127.0.0.1
*   Trying [::1]:8080...
* Connected to localhost (::1) port 8080
* using HTTP/1.x
> GET / HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/8.10.1
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 200 OK
< date: Tue, 18 Feb 2025 20:03:17 GMT
< server: uvicorn
< content-length: 44
< content-type: application/json
<
{"message":"Welcome to the FastAPI Server!"}* Connection #0 to host localhost left intact

C:\Users\VivibooK STUDIO\Desktop\Dockerfile>curl -X POST http://localhost:8080/execute
{"result":true}
```

{“result” :true}

Bonus Points

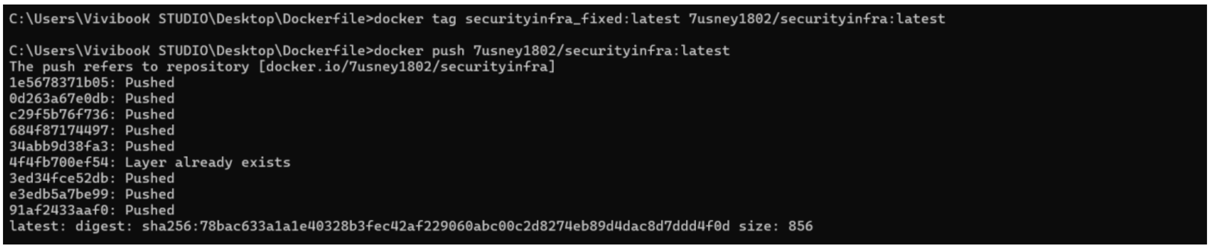
1. **Upload the Fixed Docker Image to Docker Hub**

   Tagged the image using the command:

   docker tag securityinfra_fixed:latest 7usney1802/securityinfra:latest

   Push it to the docker hub using the command

   docker push 7usney1802/securityinfra:latest

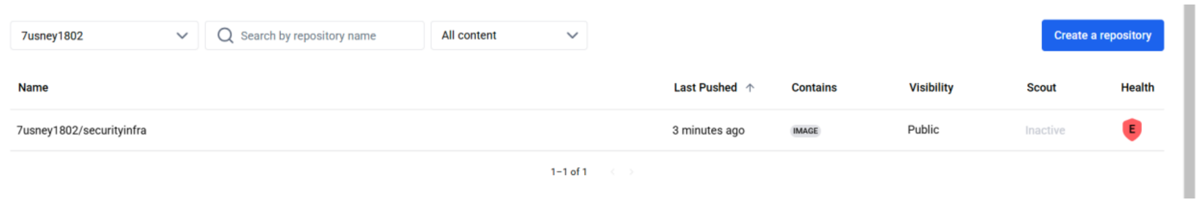


```
C:\Users\VivibooK STUDIO\Desktop\Dockerfile>docker tag securityinfra_fixed:latest 7usney1802/securityinfra:latest

C:\Users\VivibooK STUDIO\Desktop\Dockerfile>docker push 7usney1802/securityinfra:latest
The push refers to repository [docker.io/7usney1802/securityinfra]
1e5678371b05: Pushed
0d263a67e0db: Pushed
c29f5b76f736: Pushed
684f87174497: Pushed
34abb9d38fa3: Pushed
4f4fb700ef54: Layer already exists
3ed34fce52db: Pushed
e3edb5a7be99: Pushed
91af2433aaf0: Pushed
latest: digest: sha256:78bac633a1a1e40328b3fec42af229060abc00c2d8274eb89d4dac8d7ddd4f0d size: 856
```

Expected:
fixed image is in the repository

Actual Result:

As expected.



Name	Last Pushed ↑	Contains	Visibility	Scout	Health
7usney1802/securityinfra	3 minutes ago	IMAGE	Public	Inactive	E

Link:

https://hub.docker.com/repository/docker/7usney1802/securityinfra/tags/latest/sha256-a9dde5a19745aaaeabbe51d002c4c1bfc1c5a962c522225cff0bbeecd65f2351

Testing by pulling the image again:

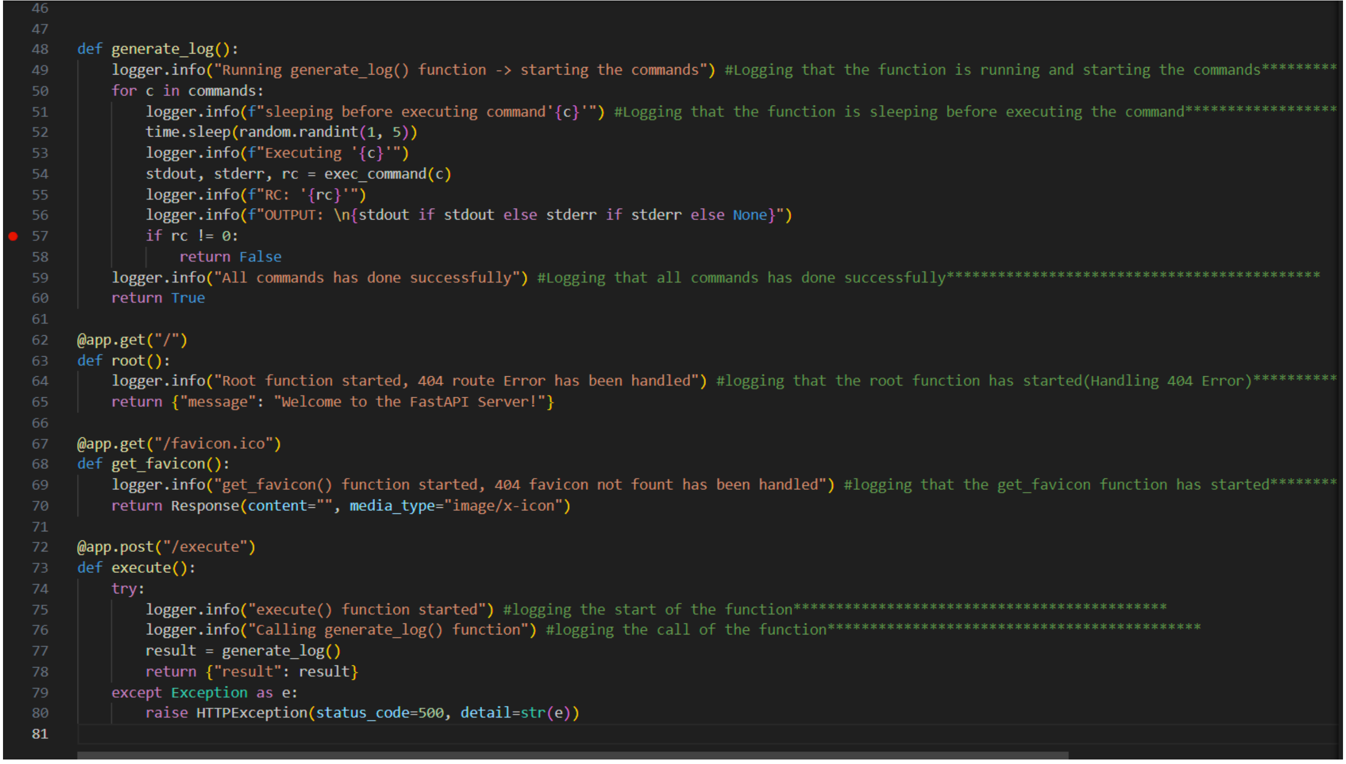
docker pull 7usney1802/securityinfra:latest

then run it:
docker run -d -p 8080:8080 --name securityinfra_Finalpull -e LOG_FILE_DIR=/home/tdt/logs 7usney1802/securityinfra:latest

now you’re having the corrected image!

- **Improve Logging & Monitoring**

  Added logger.info to check that every step in the code is running correctly, and



```python


def generate_log():
    logger.info("Running generate_log() function -> starting the commands") #Logging that the function is running and starting the commands*********
    for c in commands:
        logger.info(f"sleeping before executing command'{c}'") #Logging that the function is sleeping before executing the command*****************
        time.sleep(random.randint(1, 5))
        logger.info(f"Executing '{c}'")
        stdout, stderr, rc = exec_command(c)
        logger.info(f"RC: '{rc}'")
        logger.info(f"OUTPUT: \n{stdout if stdout else stderr if stderr else None}")
        if rc != 0:
            return False
    logger.info("All commands has done successfully") #Logging that all commands has done successfully********************************************
    return True

@app.get("/")
def root():
    logger.info("Root function started, 404 route Error has been handled") #logging that the root function has started(Handling 404 Error)**********
    return {"message": "Welcome to the FastAPI Server!"}

@app.get("/favicon.ico")
def get_favicon():
    logger.info("get_favicon() function started, 404 favicon not fount has been handled") #logging that the get_favicon function has started********
    return Response(content="", media_type="image/x-icon")

@app.post("/execute")
def execute():
    try:
        logger.info("execute() function started") #logging the start of the function********************************************
        logger.info("Calling generate_log() function") #logging the call of the function********************************************
        result = generate_log()
        return {"result": result}
    except Exception as e:
        raise HTTPException(status_code=500, detail=str(e))

```

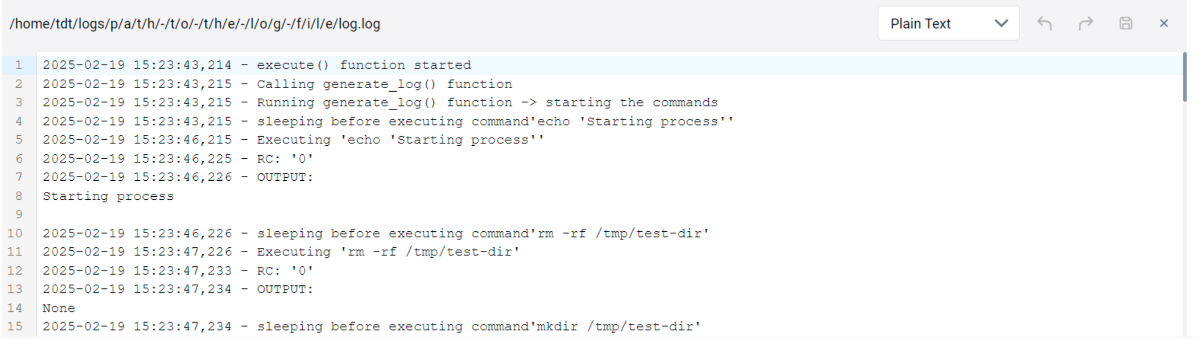
Expected:

Printing the logs into the log file

Actual Result:

Printing the logs as expected.

Part of the added logs in the log file:



/home/tdt/logs/p/a/t/h/-/t/o/-/t/h/e/-/l/o/g/-/f/i/l/e/log.log

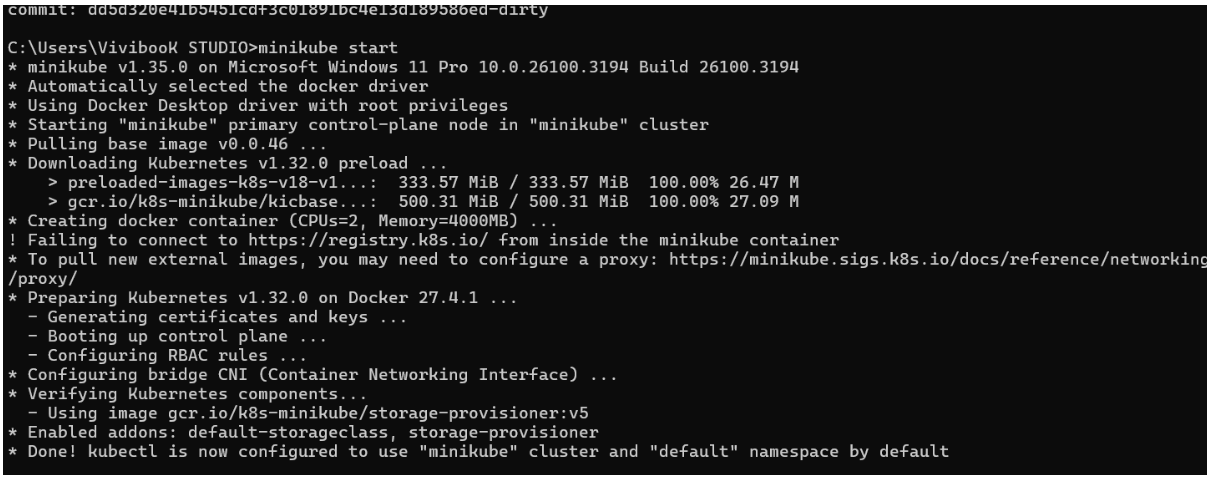
```
2025-02-19 15:23:43,214 - execute() function started
2025-02-19 15:23:43,215 - Calling generate_log() function
2025-02-19 15:23:43,215 - Running generate_log() function -> starting the commands
2025-02-19 15:23:43,215 - sleeping before executing command'echo 'Starting process''
2025-02-19 15:23:46,215 - Executing 'echo 'Starting process''
2025-02-19 15:23:46,225 - RC: '0'
2025-02-19 15:23:46,226 - OUTPUT:
Starting process

2025-02-19 15:23:46,226 - sleeping before executing command'rm -rf /tmp/test-dir'
2025-02-19 15:23:47,226 - Executing 'rm -rf /tmp/test-dir'
2025-02-19 15:23:47,233 - RC: '0'
2025-02-19 15:23:47,234 - OUTPUT:
None
2025-02-19 15:23:47,234 - sleeping before executing command'mkdir /tmp/test-dir'
```

Didn’t Complete:

- **Deploy the Web App to a Kubernetes Cluster (Managed Or Unmanaged)**

Tried to use Minikube(Unmanaged kubernetes)
Installed it:
Adding minikube to the systems Path env variables:

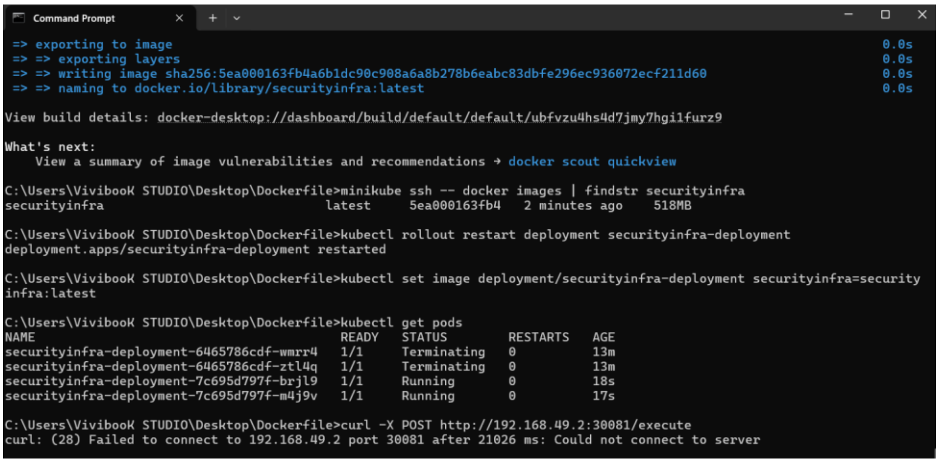
- **"This PC" -> "Properties".**
- **Advanced system settings** -> **Environment Variables**.
- Under **System variables**, find `Path`, click **Edit**.
- Click **New**, and add: `C:\minikube`



```
commit: dd5d320e41b5451cdf3c01891bc4e13d189586ed-dirty

C:\Users\VivibooK STUDIO>minikube start
* minikube v1.35.0 on Microsoft Windows 11 Pro 10.0.26100.3194 Build 26100.3194
* Automatically selected the docker driver
* Using Docker Desktop driver with root privileges
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.46 ...
* Downloading Kubernetes v1.32.0 preload ...
    > preloaded-images-k8s-v18-v1...:  333.57 MiB / 333.57 MiB  100.00% 26.47 M
    > gcr.io/k8s-minikube/kicbase...:  500.31 MiB / 500.31 MiB  100.00% 27.09 M
* Creating docker container (CPUs=2, Memory=4000MB) ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube container
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking
/proxy/
* Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: default-storageclass, storage-provisioner
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

BUG:

Tried really hard to deploy the web app to a kubernetes cluster(minikube) ☹
but getting this error:



```
 => exporting to image                                                                    0.0s
 => => exporting layers                                                                   0.0s
 => => writing image sha256:5ea000163fb4a6b1dc90c908a6a8b278b6eabc83dbfe296ec936072ecf211d60  0.0s
 => => naming to docker.io/library/securityinfra:latest                                   0.0s

View build details: docker-desktop://dashboard/build/default/default/ubfvzu4hs4d7jmy7hgi1furz9

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview

C:\Users\VivibooK STUDIO\Desktop\Dockerfile>minikube ssh -- docker images | findstr securityinfra
securityinfra                             latest     5ea000163fb4   2 minutes ago    518MB

C:\Users\VivibooK STUDIO\Desktop\Dockerfile>kubectl rollout restart deployment securityinfra-deployment
deployment.apps/securityinfra-deployment restarted

C:\Users\VivibooK STUDIO\Desktop\Dockerfile>kubectl set image deployment/securityinfra-deployment securityinfra=security
infra:latest

C:\Users\VivibooK STUDIO\Desktop\Dockerfile>kubectl get pods
NAME                                       READY   STATUS        RESTARTS   AGE
securityinfra-deployment-6465786cdf-wmrr4   1/1     Terminating   0          13m
securityinfra-deployment-6465786cdf-ztl4q   1/1     Terminating   0          13m
securityinfra-deployment-7c695d797f-brjl9   1/1     Running       0          18s
securityinfra-deployment-7c695d797f-m4j9v   1/1     Running       0          17s

C:\Users\VivibooK STUDIO\Desktop\Dockerfile>curl -X POST http://192.168.49.2:30081/execute
curl: (28) Failed to connect to 192.168.49.2 port 30081 after 21026 ms: Could not connect to server
```