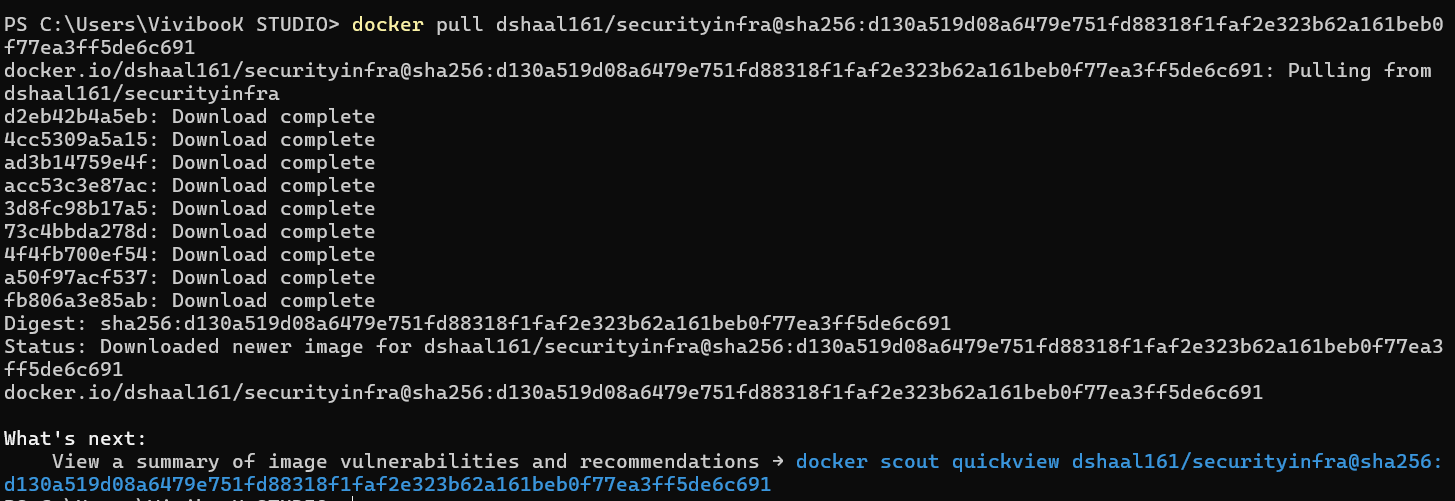
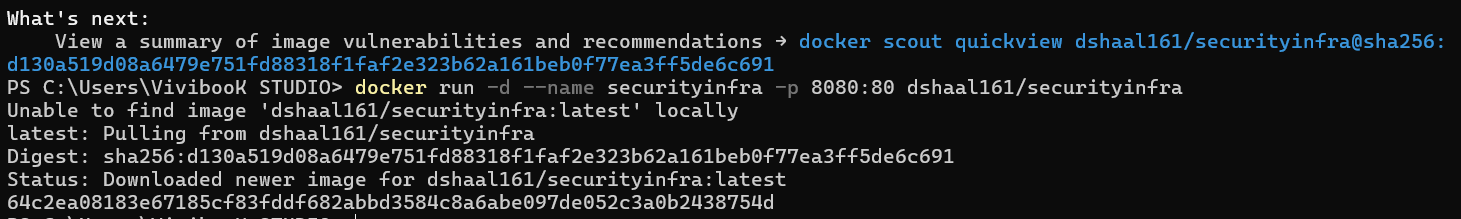
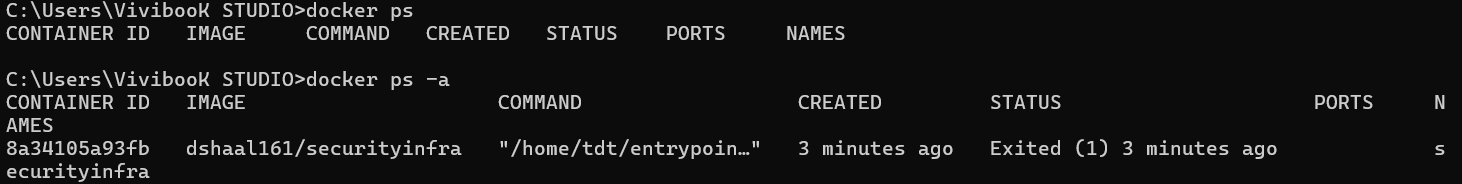
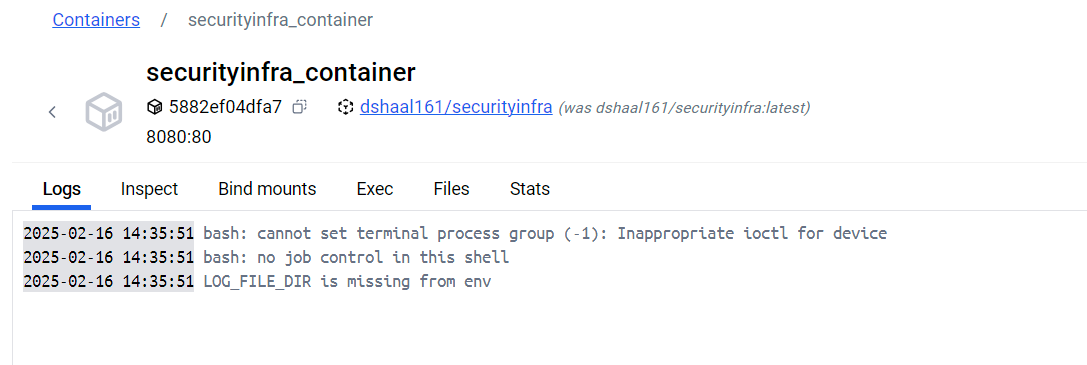
Started with pulling the Docker image:  




Verify if the container is running(Didn’t run), so I checked the running containers:



Had to check the logs:  


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")

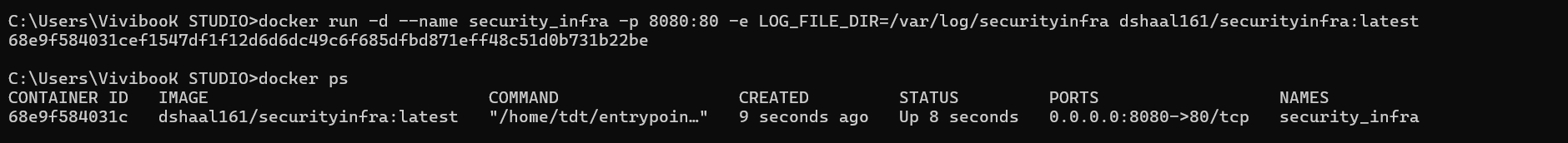
if not log\_file\_dir:

raise ValueError("Environment variable LOG\_FILE\_DIR is missing")

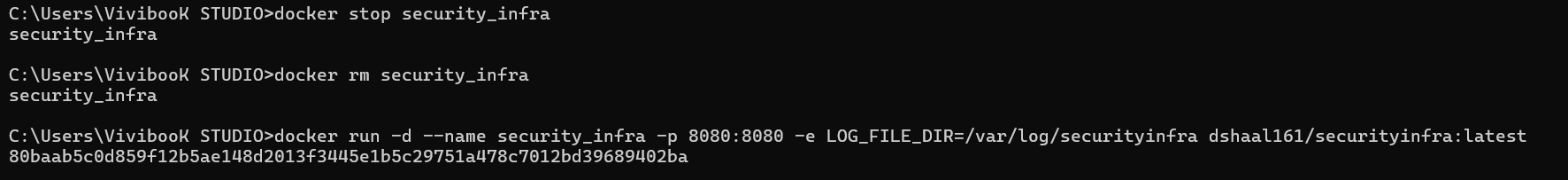
To this:  
log\_file\_dir = os.getenv("LOG\_FILE\_DIR", "/home/tdt/logs")

But it didn’t work.

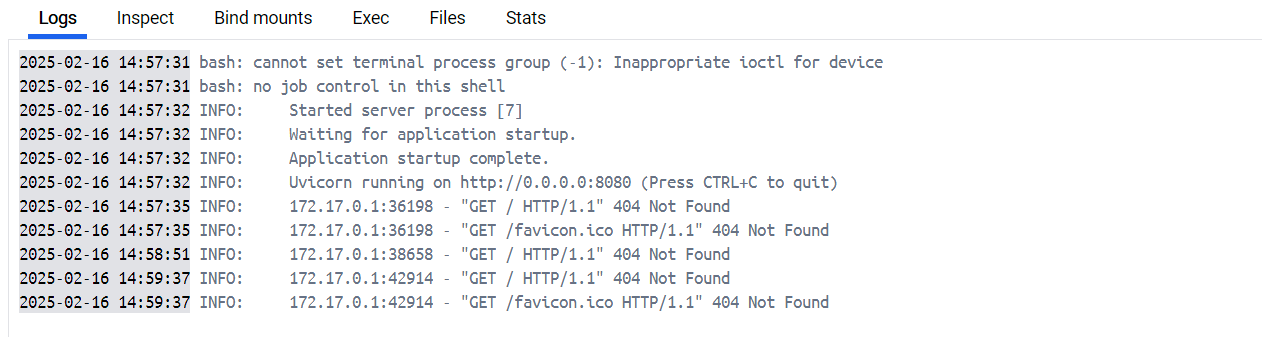
I had to set it manually.

docker run -d -p 8080:8080 --name securityinfra\_final -**e LOG\_FILE\_DIR=/home/tdt/logs** 7usney1802/securityinfra:latest  
now it’s running!

  
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.

After trying to open the web by the link: [localhost:8080](http://localhost:8080/)  
I got this error:  


I had to check the logs: 

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>

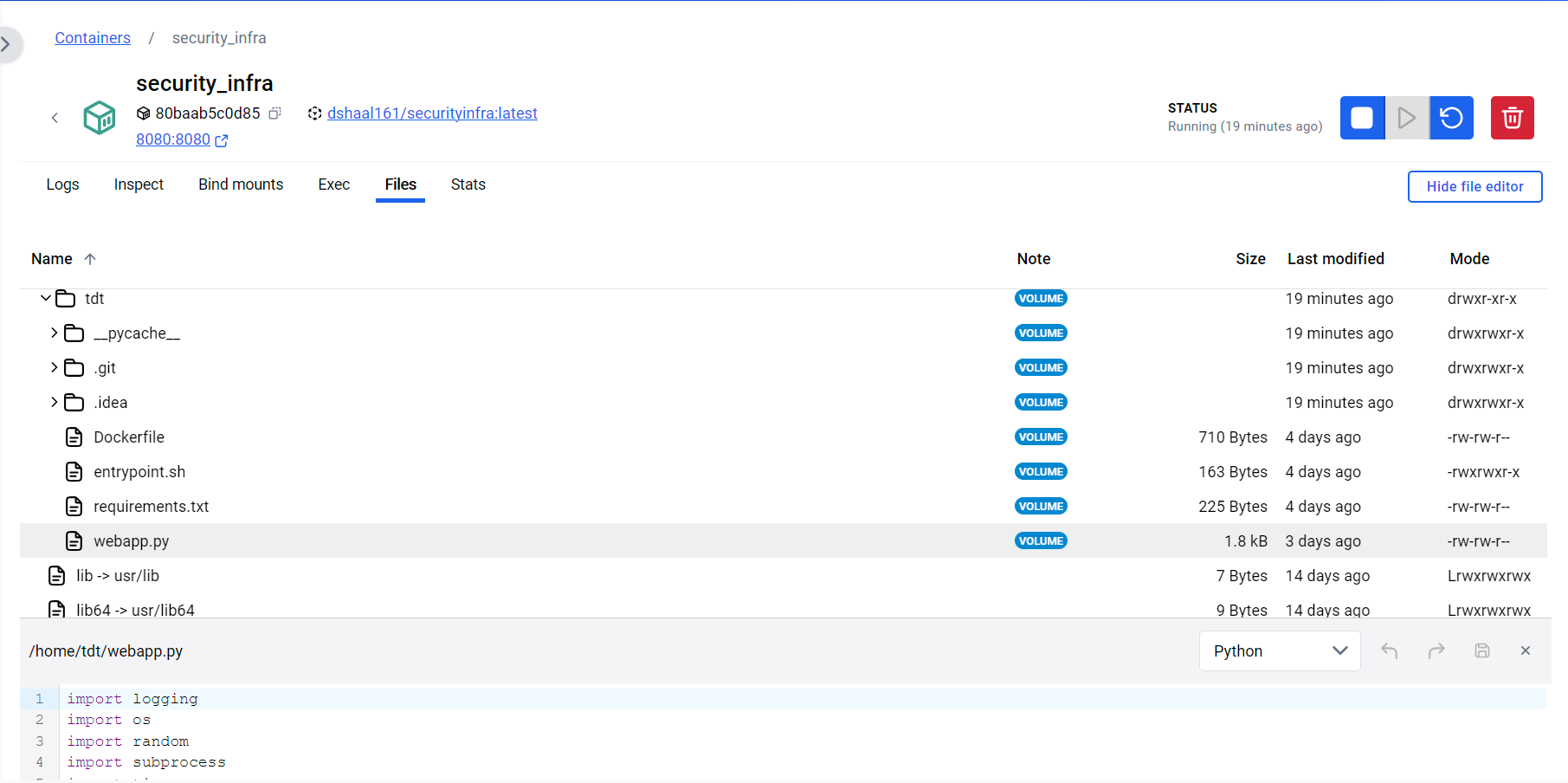
curl <http://localhost:8080/check>

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”  
  
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"

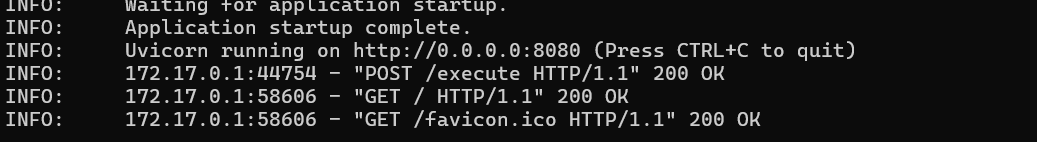
Tried to run the code to check if I have solved the error:

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)

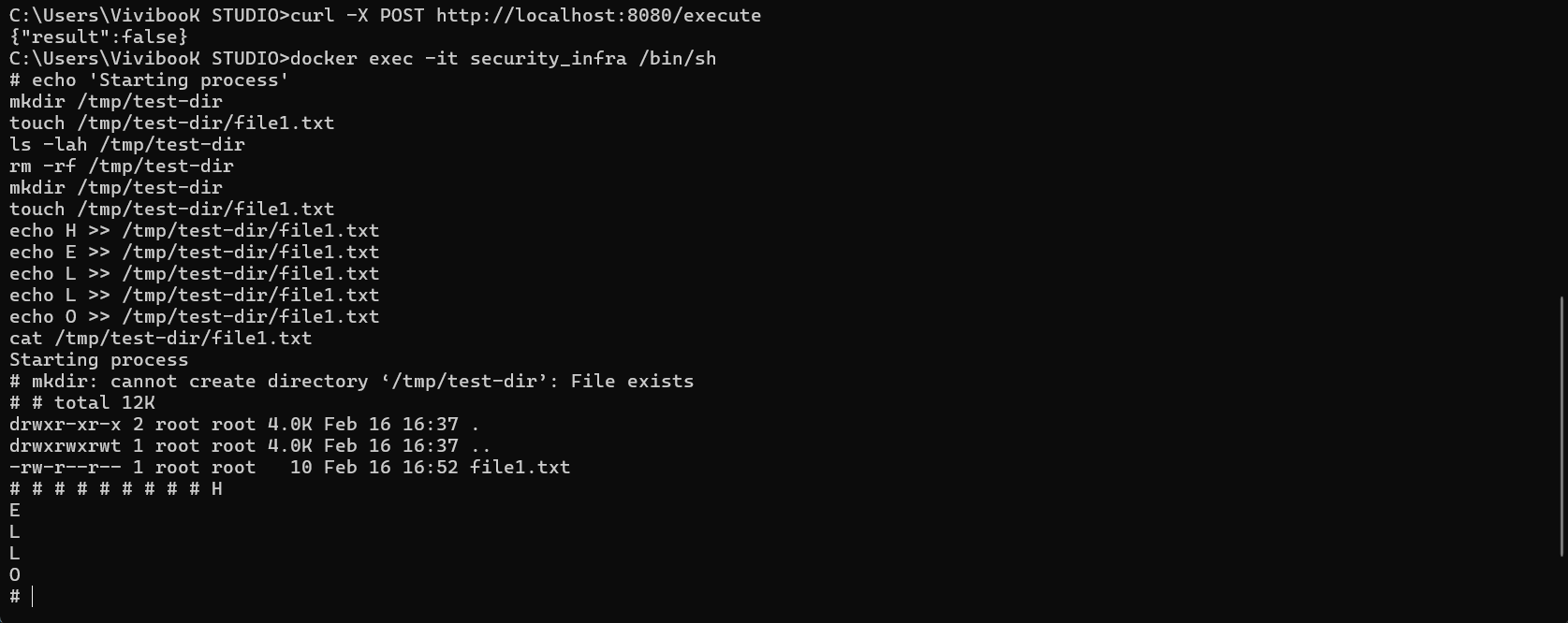
But still having this:



So I checked the logs by the command ”docker logs security\_infra”, but got no errors here:



So I had to run the command:

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


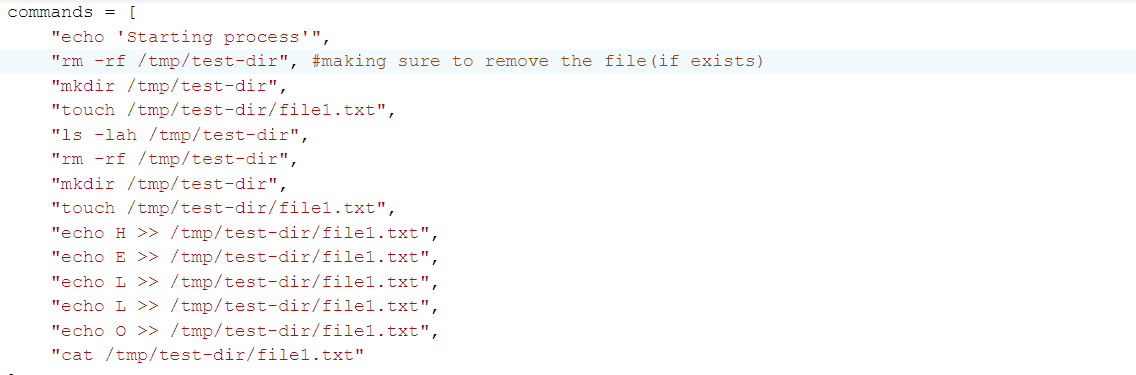
BUG:

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

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

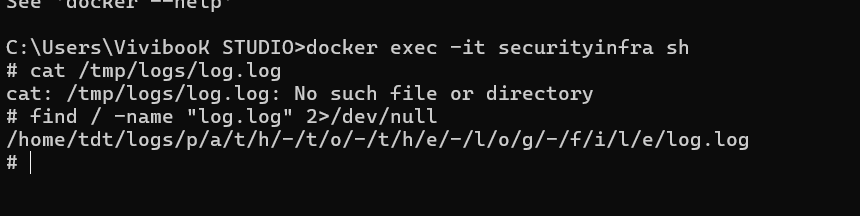
****

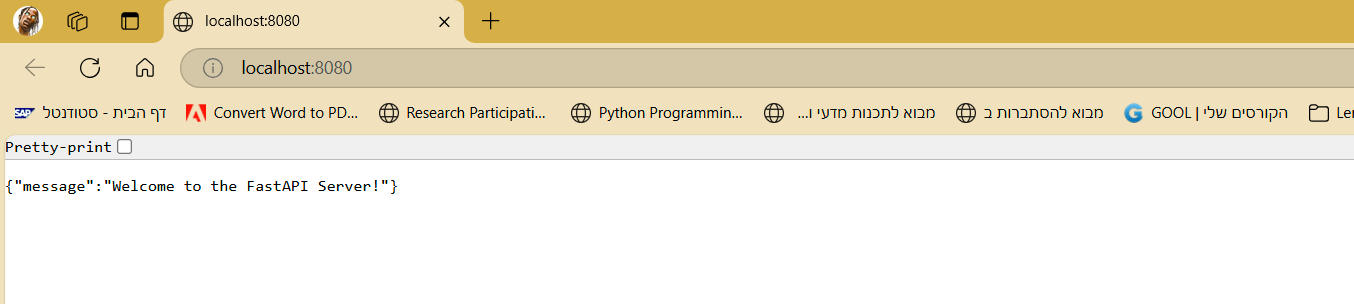
To this:  
(adding “rm –rf /tmp/test-dir”,)



**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)   


After testing the corrected code:



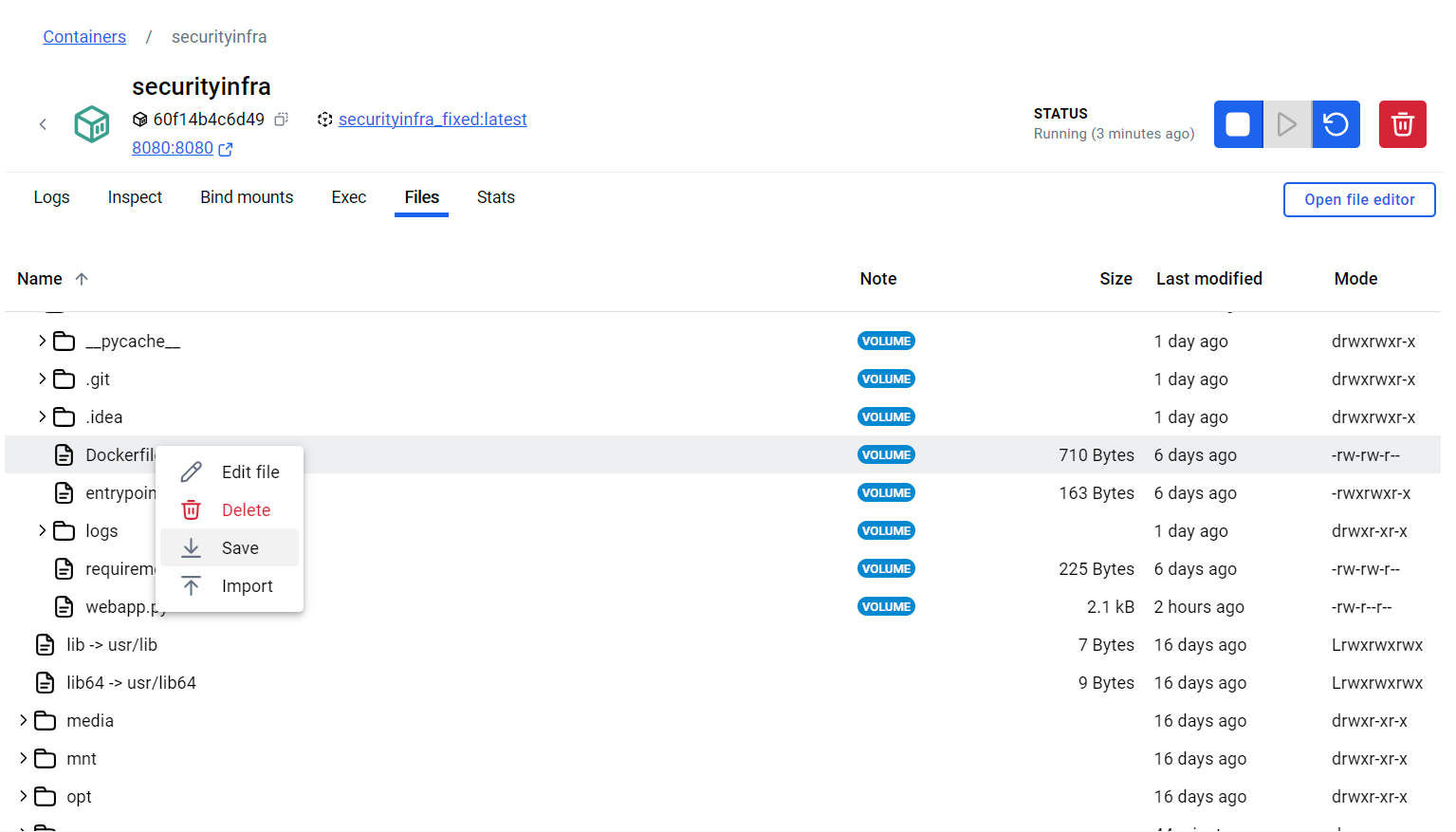
We can see that the execute function is returning True, means that all the bugs have been solved.

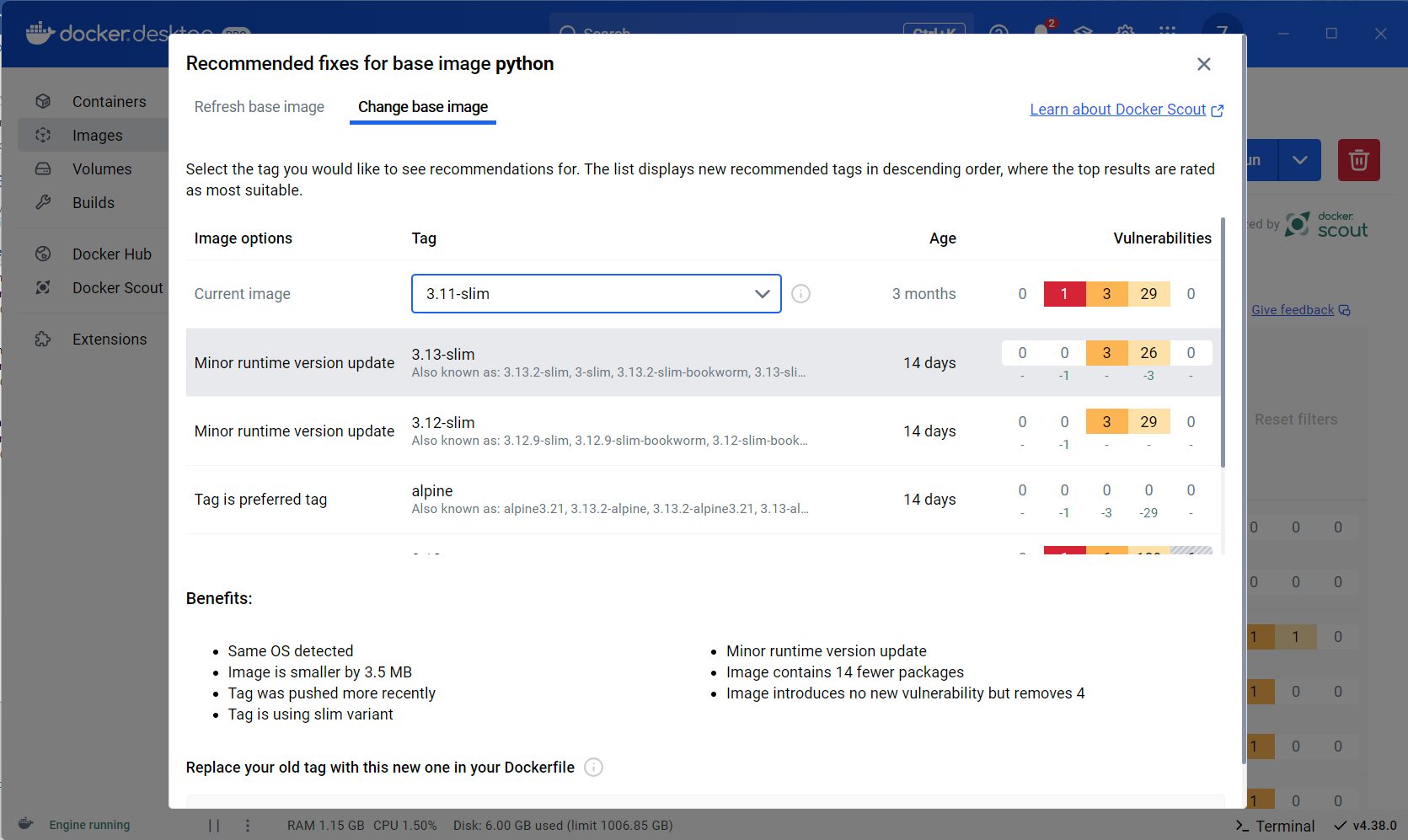
To make sure, I have checked the log file to see if there’s any errors:  

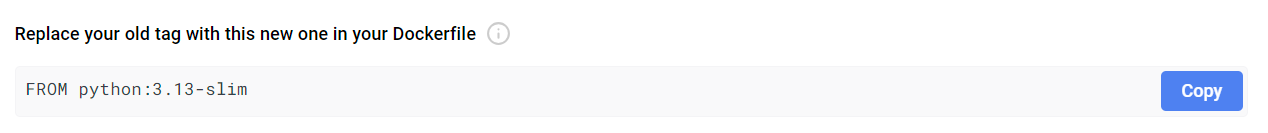

Everything looks fine, it’s returning and printing as expected.

Rebuild the image:

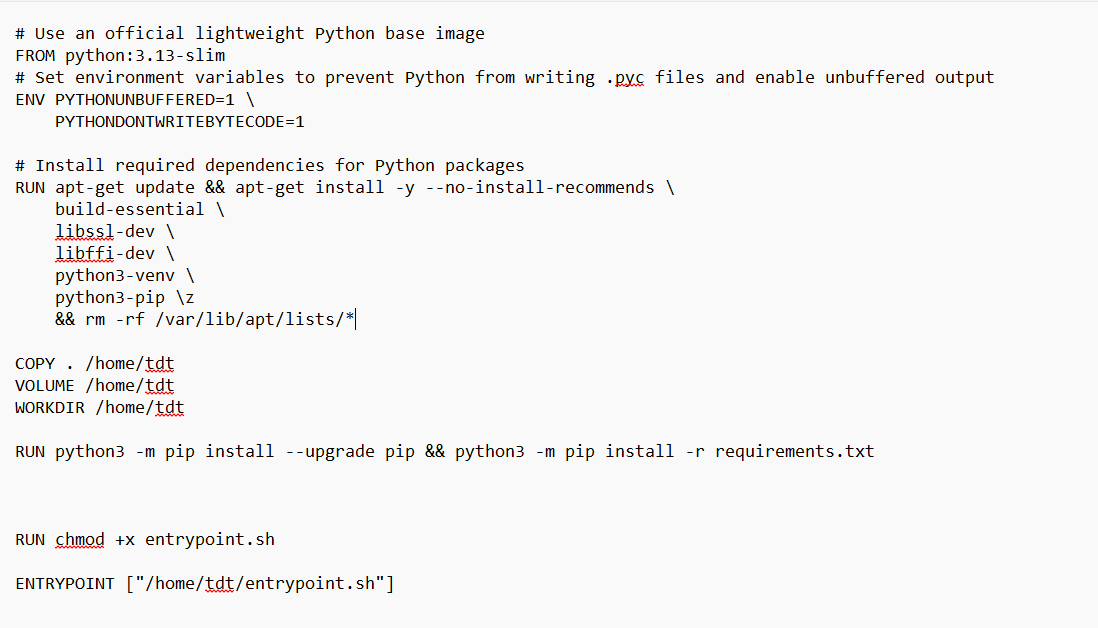
Created a folder called Dockerfile in the desktop

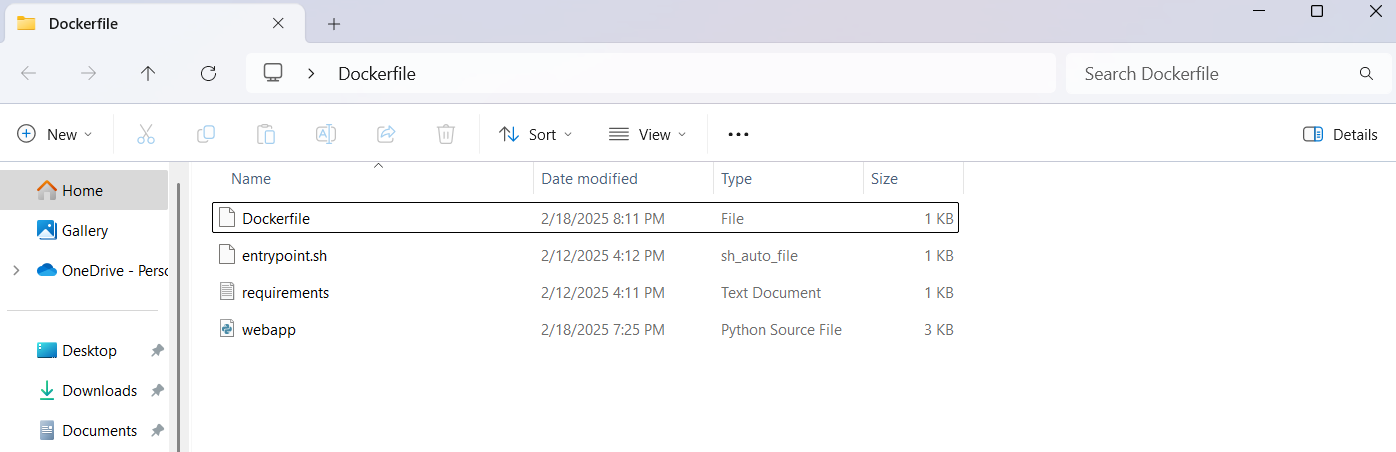
Saved the Dockerfile inside the folder  


Changed the base image from(3.11) -slim : 

To(as recommended in the dockerfile recommended fixes): 

The Dockerfile after changing the second line:



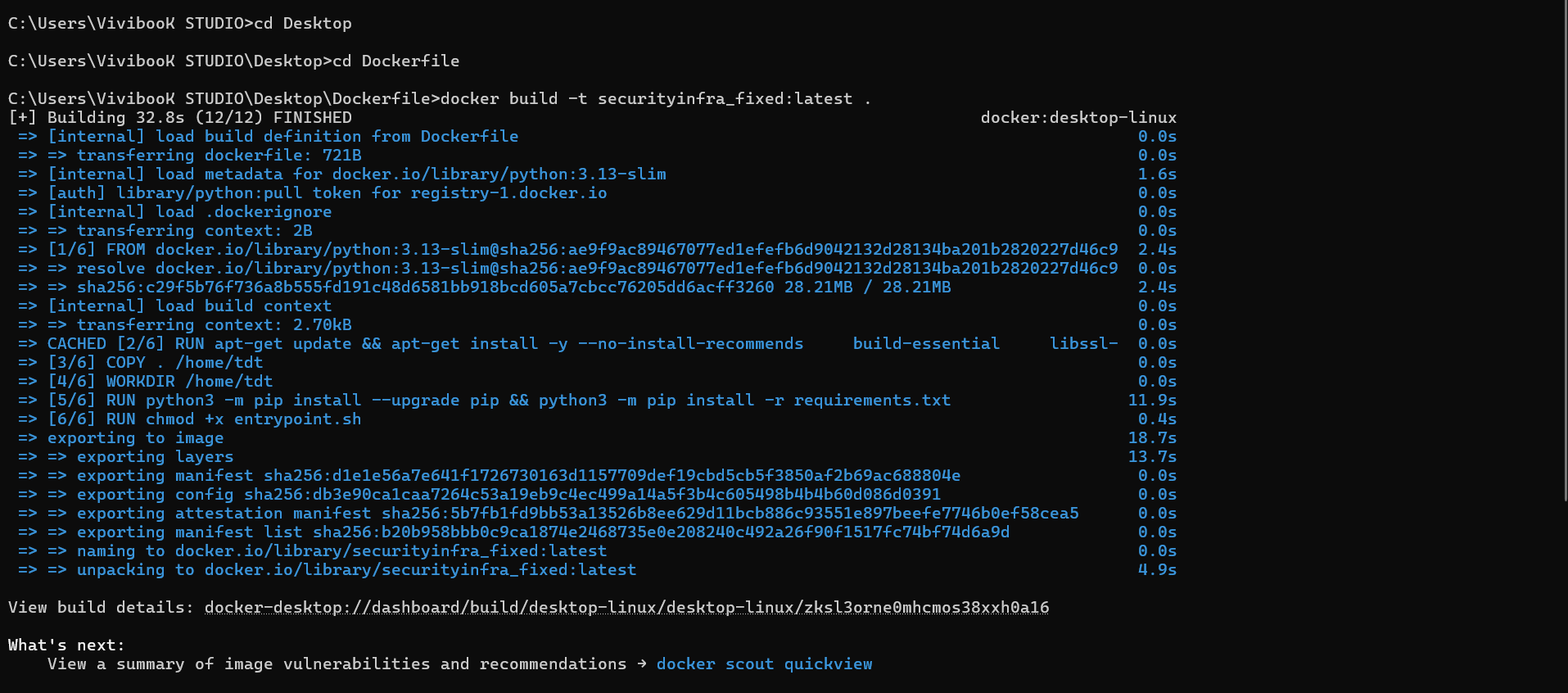
Added also to entrypoint.sh, requerements.txt,webapp.py files.(after getting an error that requirements.txt doesn’t exist and the container crashed):  


And runned these commands:

Cd Desktop(to enter the desktop)

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

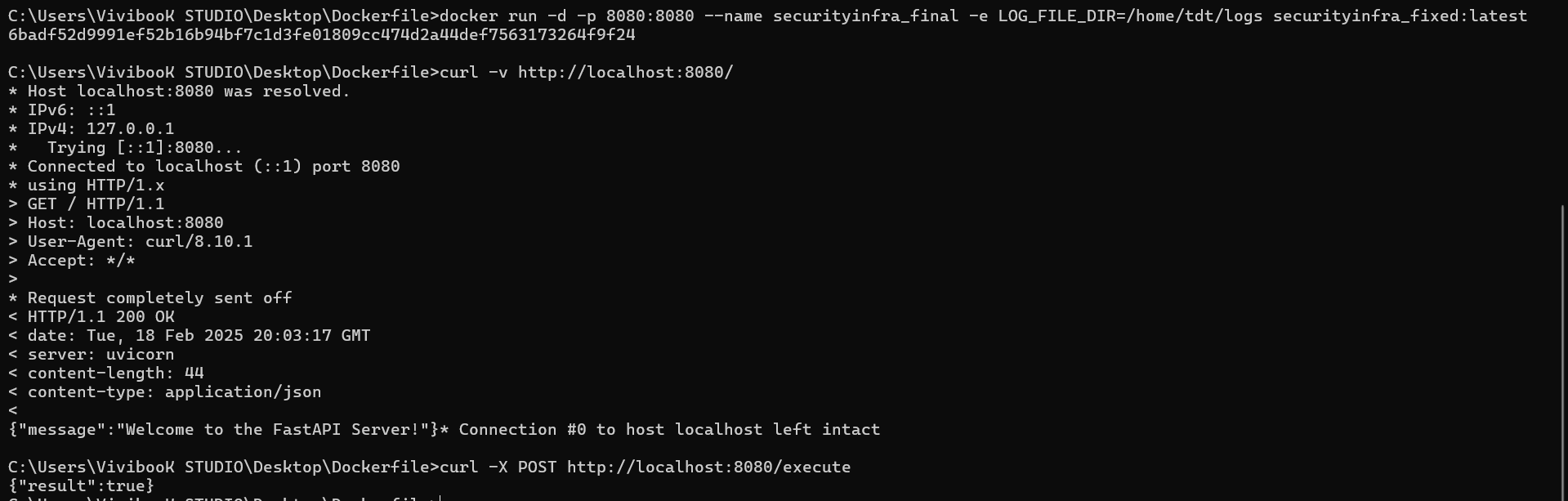
Docker build –t securityinfra\_fixed:latest . (Building the updated image)



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



Everything looks good!

**Bonus Points**

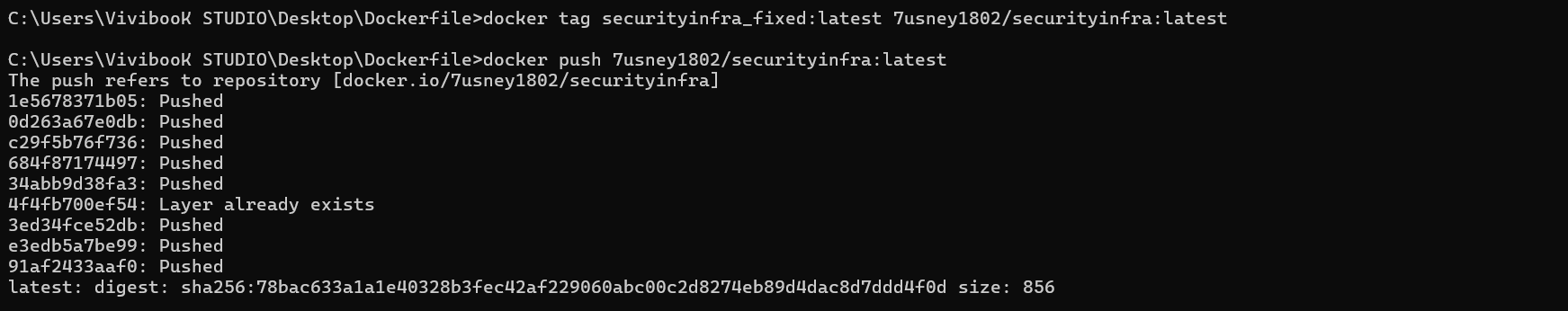
1. **Upload the Fixed Docker Image to Docker Hub**

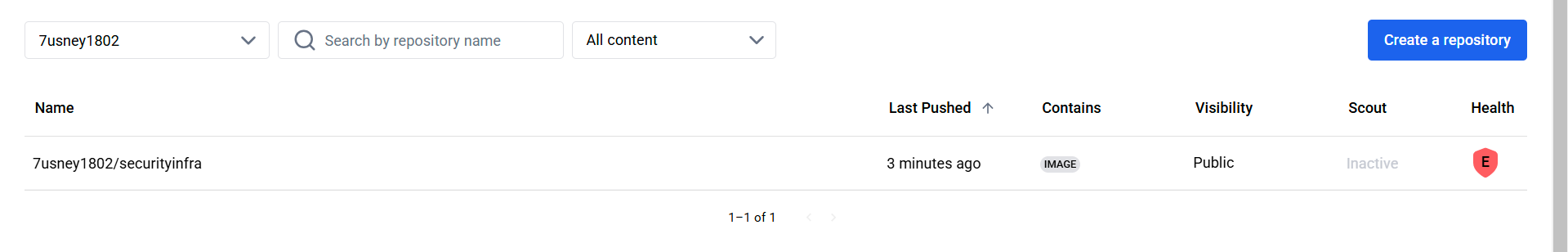
Tagged the image using the command:

docker tag securityinfra\_fixed:latest 7usney1802/securityinfra:latest

Push it to the docker hub using the command

docker push 7usney1802/securityinfra:latest





**Testing by pulling the image again:**

docker pull 7usney1802/securityinfra:latest

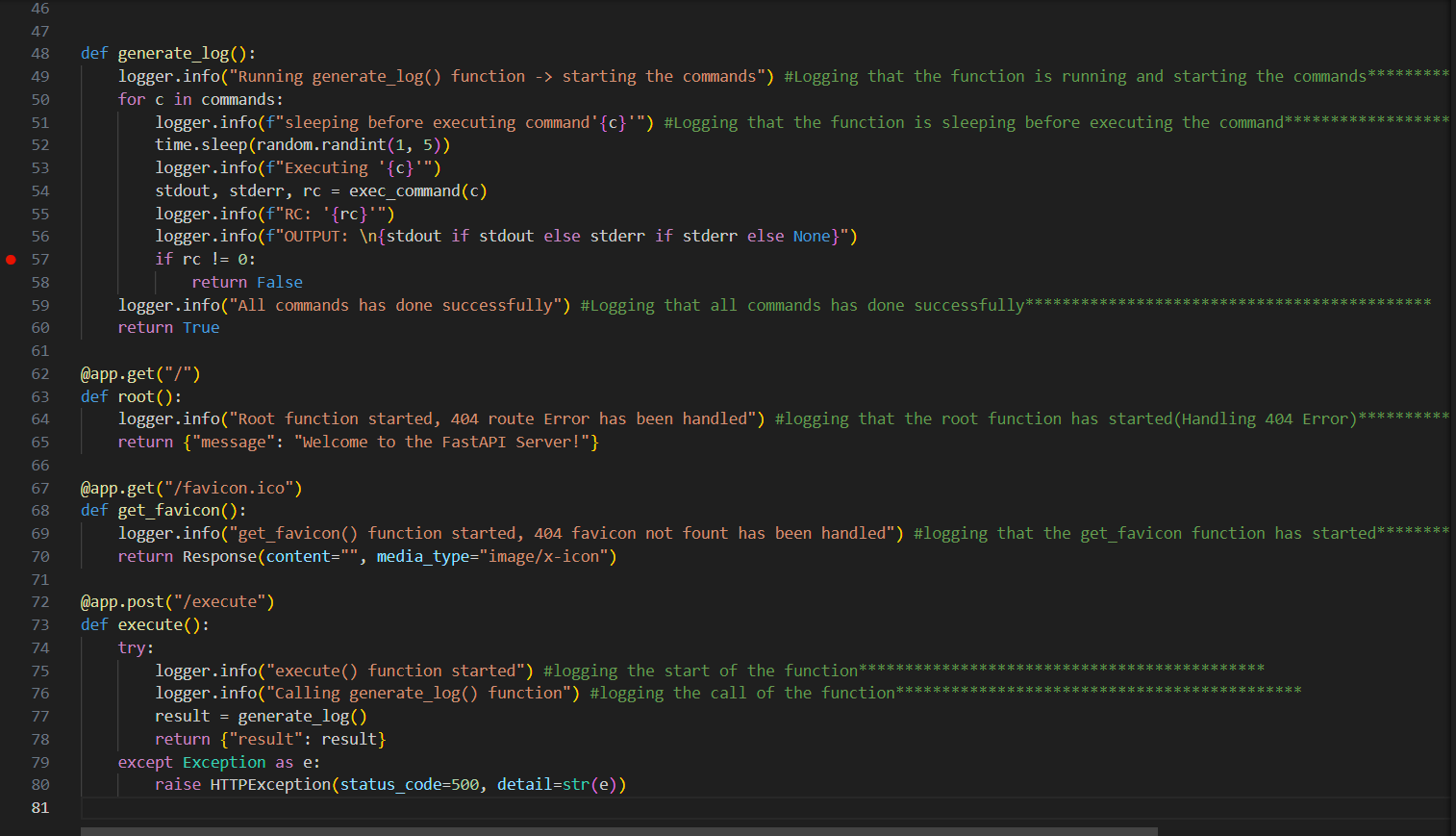
**then run it:**  
docker run -d -p 8080:8080 --name securityinfra\_Finalpull -e LOG\_FILE\_DIR=/home/tdt/logs 7usney1802/securityinfra:latest

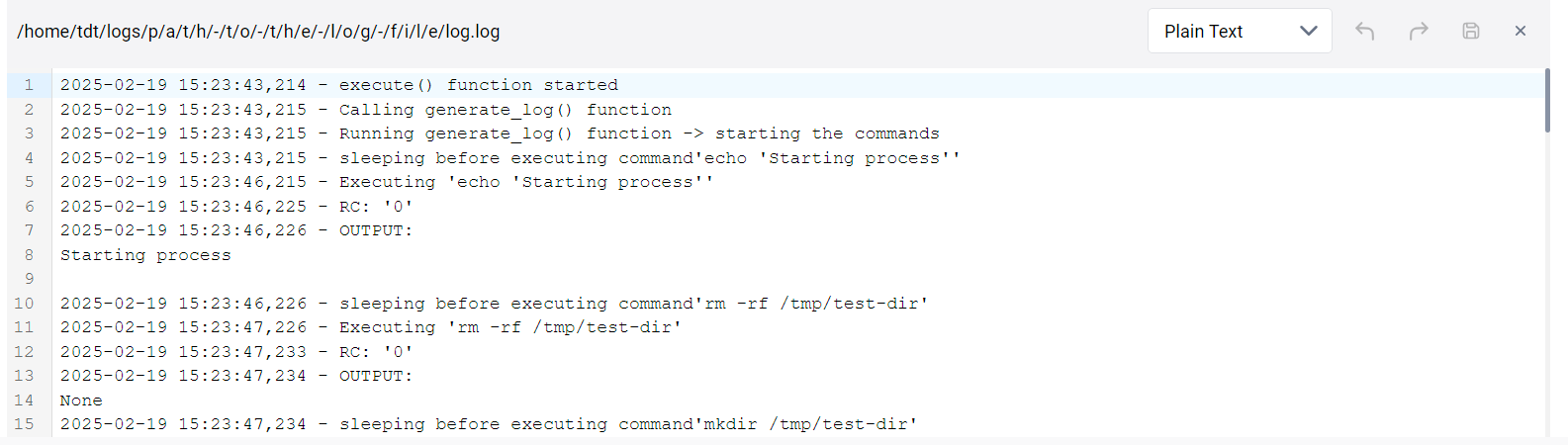
now you’re having the corrected image!

[https://hub.docker.com/repository/docker/7usney1802/securityinfra/](https://hub.docker.com/repository/docker/7usney1802/securityinfra/general)

* **Improve Logging & Monitoring**

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



Part of the added logs in the log file:  


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

Tried to use Minikube(Unmanaged kubernetes)

Installed it:

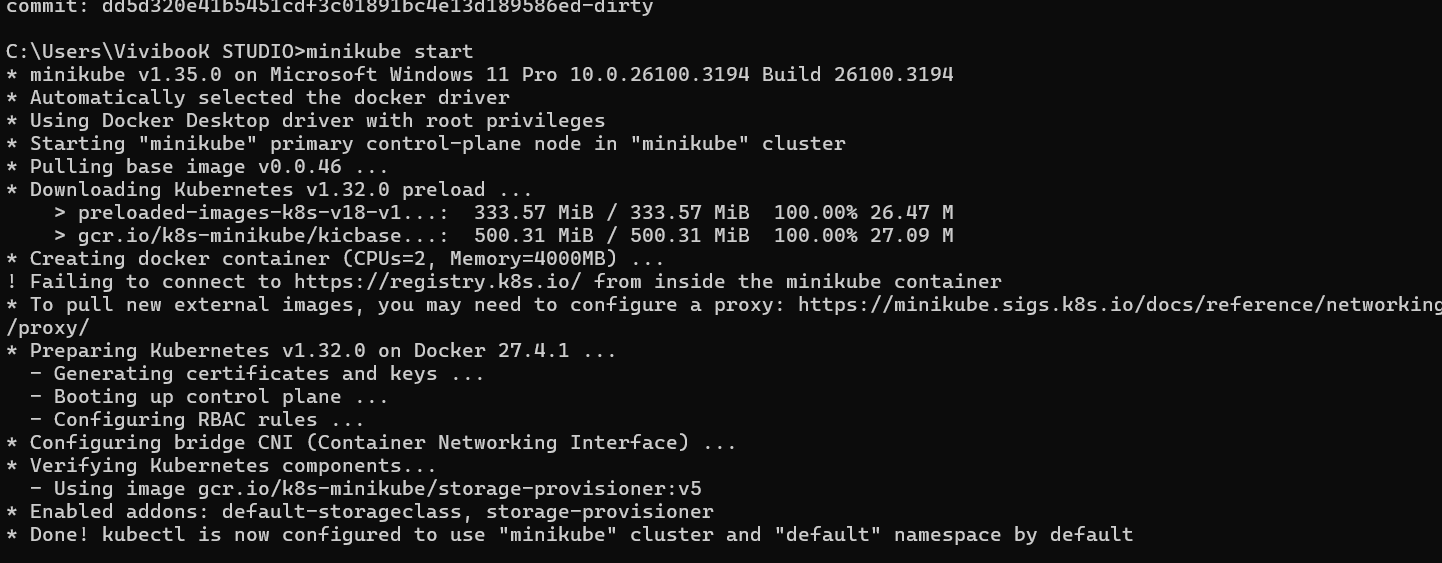
Adding minikube to the systems Path env variables:

 **"This PC" -> "Properties".**

 **Advanced system settings** -> **Environment Variables**.

 Under **System variables**, find Path, click **Edit**.

 Click **New**, and add: C:\minikube



BUG:

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