This PDF is a guide to use the application

step 1: download the repository or clone it .

Step 2: In the directory of the project (Microservices SSI), we see the following subdirectories

```
config
                                                 2/13/2025 3:57 PM
                                                                            File folder
frontend
                                                 2/13/2025 2:02 PM
                                                                            File folder
microservices_project
                                                                            File folder
                                                 2/13/2025 2:02 PM
= zap
                                                                            File folder
                                                 2/13/2025 3:14 PM
OWASP ZAP Testing Guide
                                                 2/15/2025 1:24 PM
                                                                            Markdown Source File
                                                                                                            2 KB
```

First, open poweshell in the frontend and microservices projet

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\achra\Microservices SSI> cd frontend
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\achra> cd "Microservices SSI"

PS C:\Users\achra\Microservices SSI> CD MICROSERVICES_PROJECT

PS C:\Users\achra\Microservices SSI\MICROSERVICES_PROJECT>
```

Don't forget to login to docker with this command: docker login

Then we build the images for each file containing docker-compose.yml with: **docker-compose up --build** 

```
[+] Running 3/3g to docker.io/library/frontend-frontend:latest

Service frontend

Network frontend_default

Created

Container frontend-frontend-1

Created

Attaching to docker.io/tibrary/frontend frontend.tatest

Created

Created
```

```
=> exporting manifest list sha256:c087081989f8ee4c9d6c5d75ec5324b6d48c44ca1c76b1117
 ✓Service product-service
[+] Running 6/7.5s (56/57)
   => unpacking to docker.io/library/microservices_project-notification-service:latest
✓Service order-service
                                Built
✓Service product-service
✓Service monitoring
                                Built

✓Service notification-service Built

  Service employee-service
[+] Building 20.1s (56/57)
  Service order-service
                                Built
[+] Building 20.3s (56/57)
✓Service monitoring
=> => unpacking to docker.io/library/microservices_project-notification-service:latest
✓Service notification-service Built
 ✓Service employee-service
[+] Running 6/7-service
[+] Building 20.4s (56/57)
=> => unpacking to docker.io/library/microservices_project-notification-service:latest
   [employee-service] exporting to image
=> => exporting layers
 ✓Service order-service
                                Built
[+] Building 20.5s (56/57)
                          io/library/microservices project-notification-service:late
```

## We may confront with this problem:

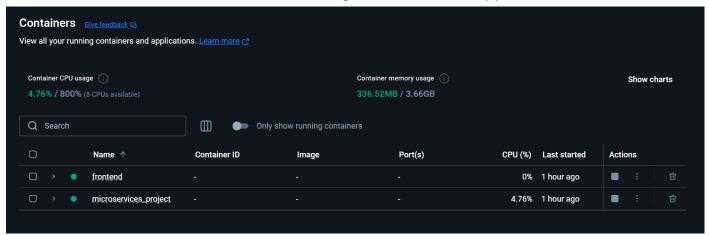
```
user-service-1 | * Serving Flask app 'app'
monitoring-1 | time=2025-02-14T16:08:20.793Z level=INFO source=main.go:764 msg="Leaving GOMAXPROCS=8: CPU quota undefined"
component=automaxprocs
user-service-1 | * Debug mode: on
Error response from daemon: Ports are not available: exposing port TCP 0.0.0.0:8080 -> 0.0.0.0:0: listen tcp 0.0.0.0:8080: bind: Only
one usage of each socket address (protocol/network address/port) is normally permitted.
```

Not a big deal, it says that the port 8080 is already in use by another service

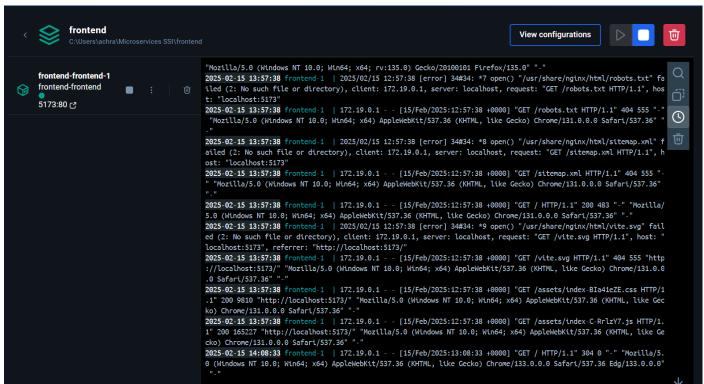
the solution is to search for the service that uses port 8080 (netsat -ano | findstr :8080), and terminate it with the command: taskkill /PID cess id> /F

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\WINDOWS\system32> netstat -ano | findstr :8080
 TCP
         0.0.0.0:8080
                                0.0.0.0:0
                                                        LISTENING
                                                                        6164
  TCP
         [::]:8080
                                [::]:0
                                                        LISTENING
                                                                        6164
PS C:\WINDOWS\system32> taskkill /PID 6164 /F
SUCCESS: The process with PID 6164 has been terminated.
PS C:\WINDOWS\system32>
```

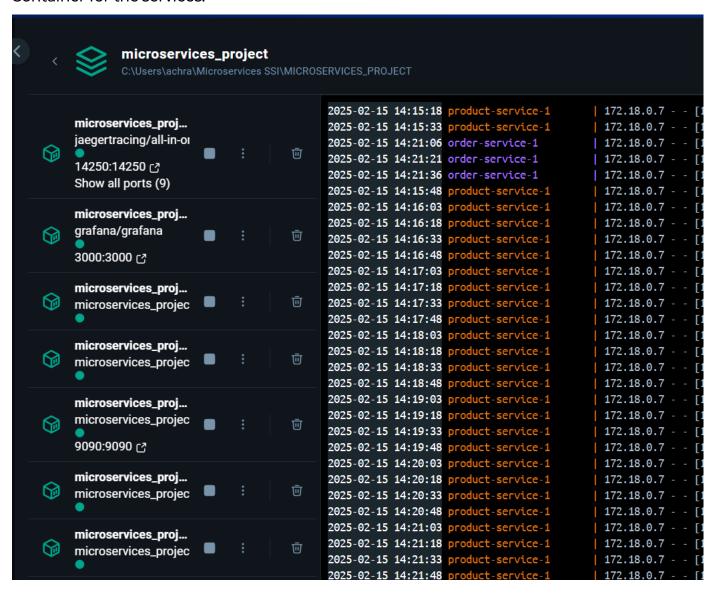
Now we created the containers and the images to initiate the application



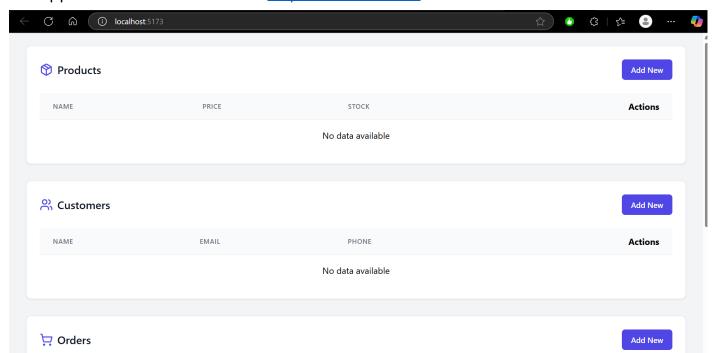
## Container for the frontend:



## Container for the services:



The application is now available in <a href="http://localhost:5173/">http://localhost:5173/</a>



TO CHECK THE TRACES , WE NEED TO VISITE  ${\bf JAEGER}$  IN ITS PORT , AND TO TEST VULNs , OPEN  ${\bf OWASP}$   ${\bf ZAP}$  .