**Containerization**

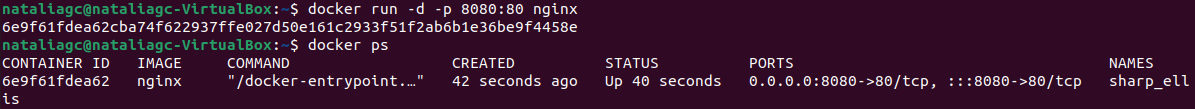
**Challenges**

1. **Run the "Hello, World" Docker container. Make sure you understand the concepts of Container Registry, pulling and pushing an image to a Registry.**



A screenshot of a computer screen

Description automatically generated

1. **Run a program that you want from a Docker container. It can be a web server, a database, or even a programming language.**

**A screenshot of a computer

Description automatically generated**

1. A screenshot of a computer program

   Description automatically generated**Run a container based on this image devopsdockeruh/simple-web-service:ubuntu. The image creates a container that outputs logs into a file. Go inside the container and use tail -f command to follow the logs. What's the secret message it outputs?**

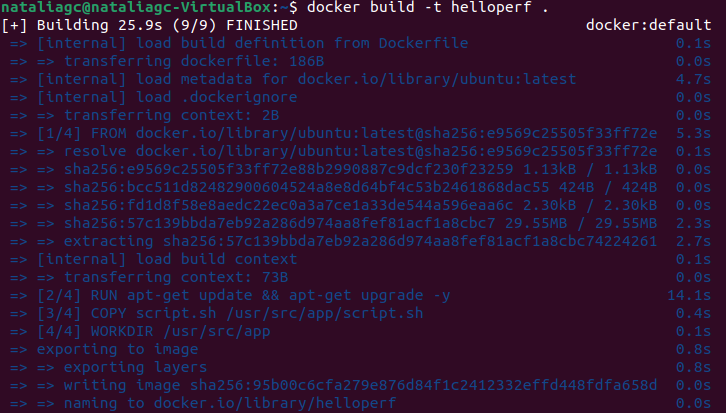
A screenshot of a computer program

Description automatically generated

1. **Given the following script, create an image from it.**

#!/bin/sh

echo "Hello, Perficient!"

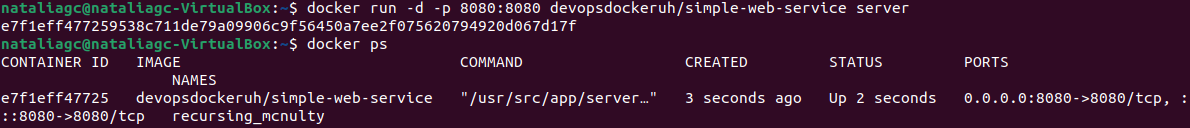


**A screen shot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

1. **When you pass the server command to the devopsdockeruh/simple-web-service image, it will create a container with a web service running on port 8080. Access it from your localhost address. You will get a message like this: "{ message: "You connected to the following path: ..."**

**A blue line on a white background

Description automatically generated**

1. **Make sure you understand the docker-compose command. How to install it and what we need it for. Here you can find information about it.**

Docker Compose is a tool that allows to define and run multi-container Docker applications using a YAML file to configure services.

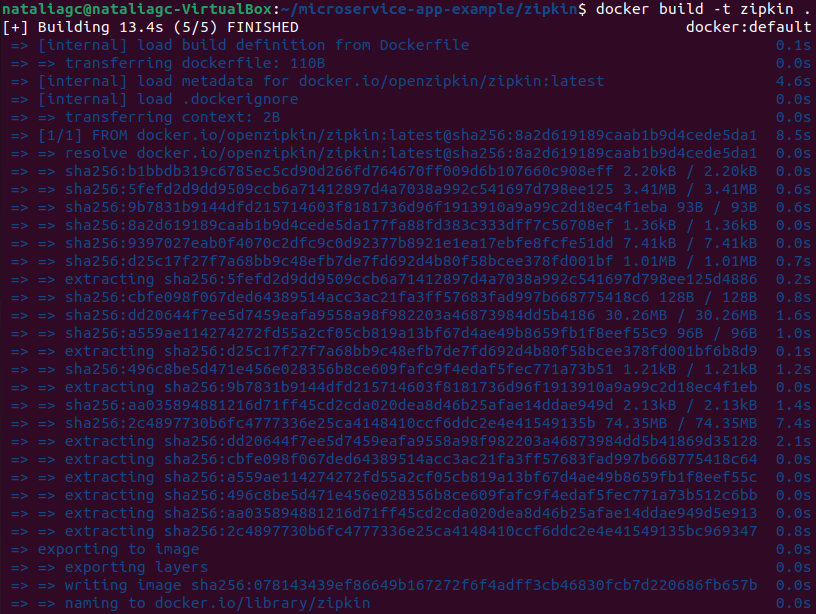
To install it: <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-compose-on-ubuntu-20-04>

To run this application, go to the directory where **docker-compose.yml** file is located and run the following command: ***docker-compose up.***

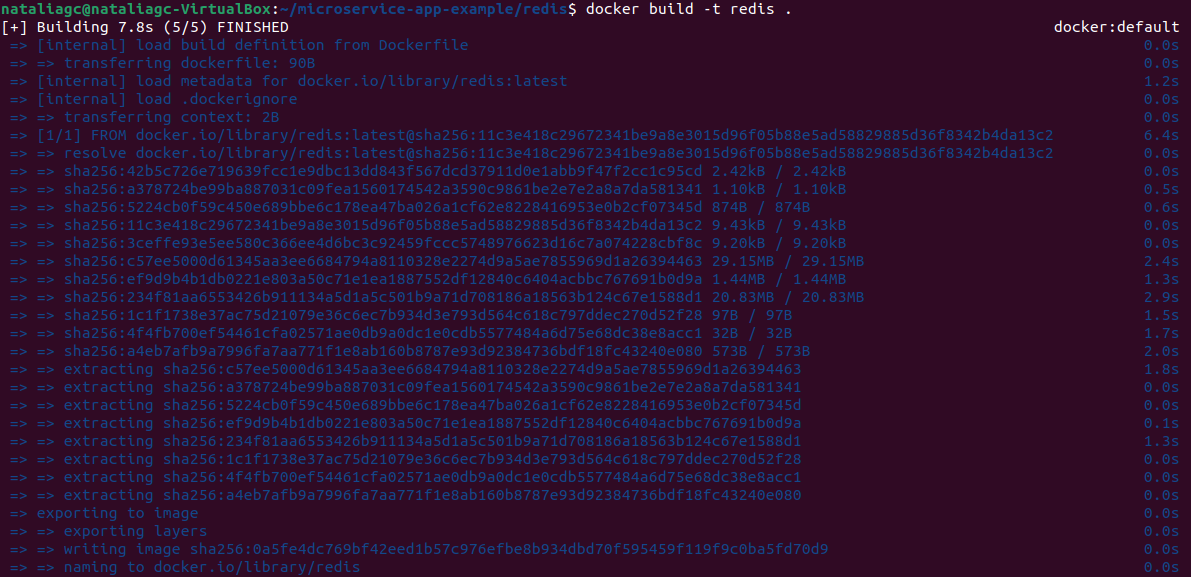
1. **Create a Dockerfile for every microservice in our microservice application. After that, run each microservice separately.**

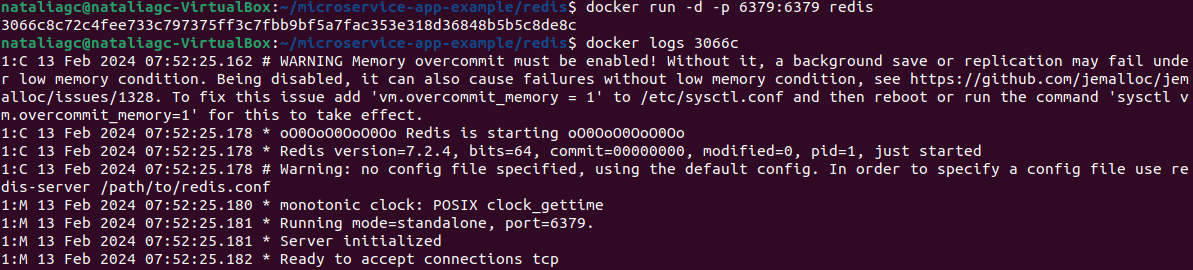
Each Dockerfile of the microservices are in this repo:

<https://github.com/nataliaguerreroc/microservice-app-example>

**Zipkin**

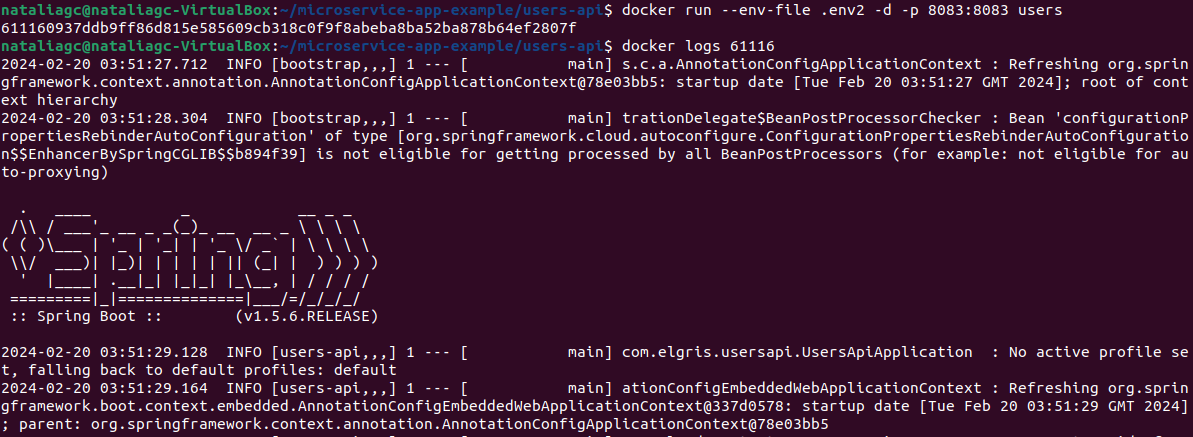
****

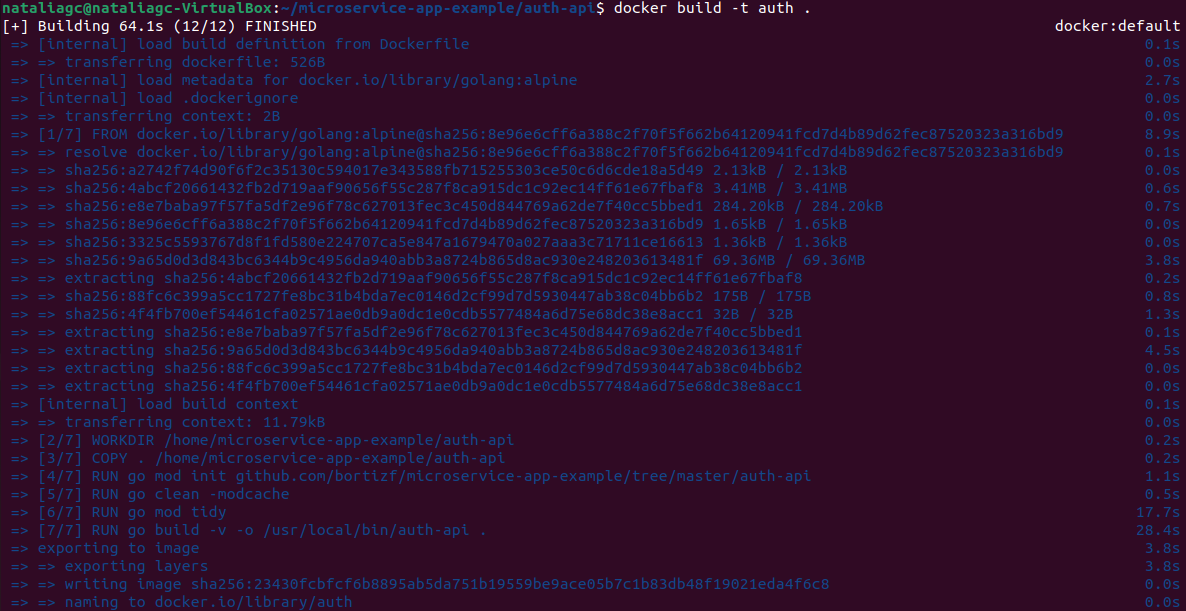
**Redis**

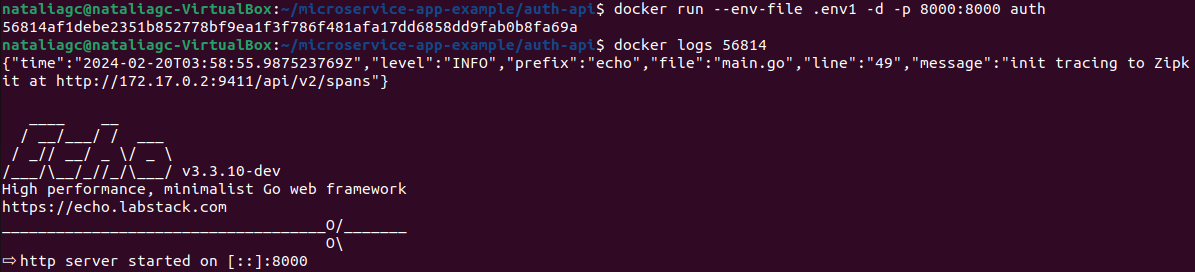
****

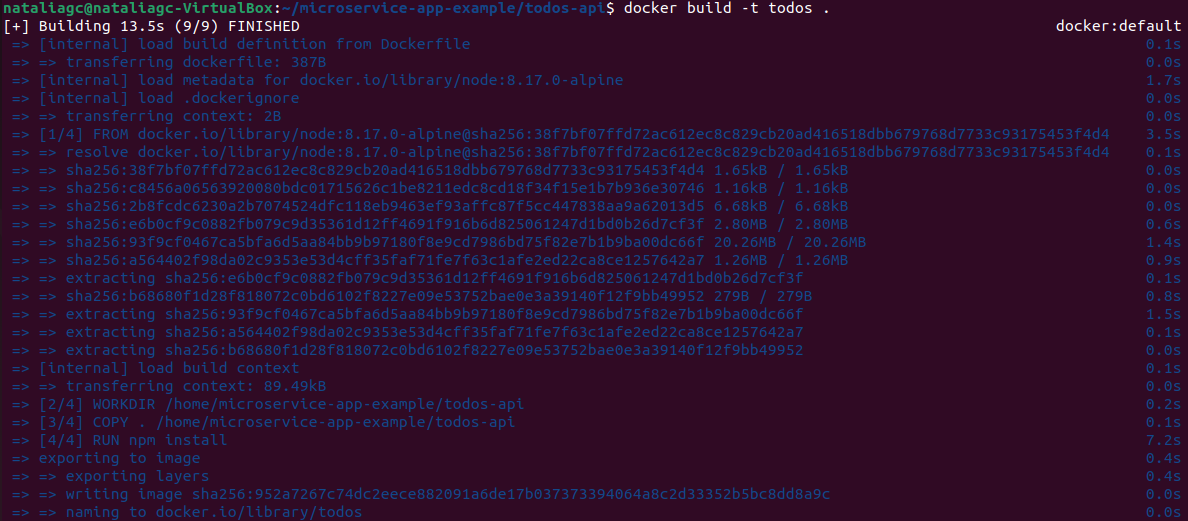
**Users**

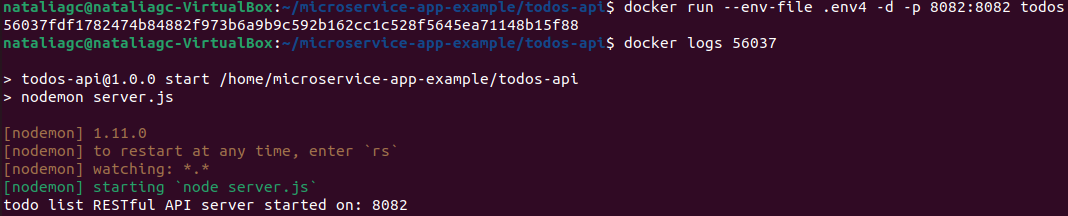
****

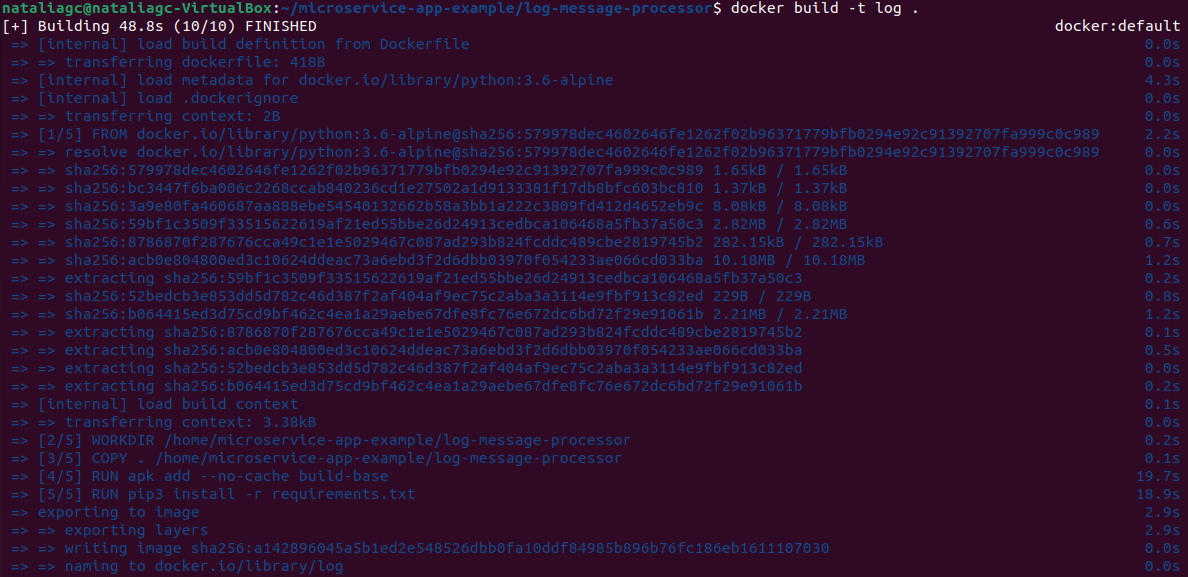
****

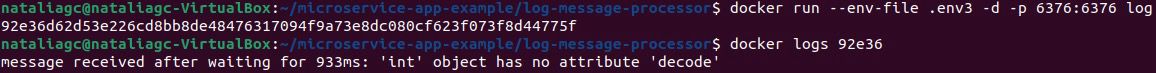
**Auth**

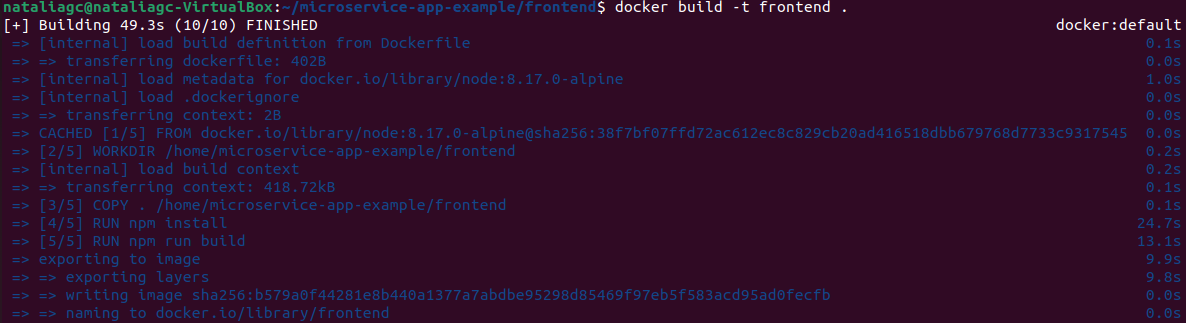
****

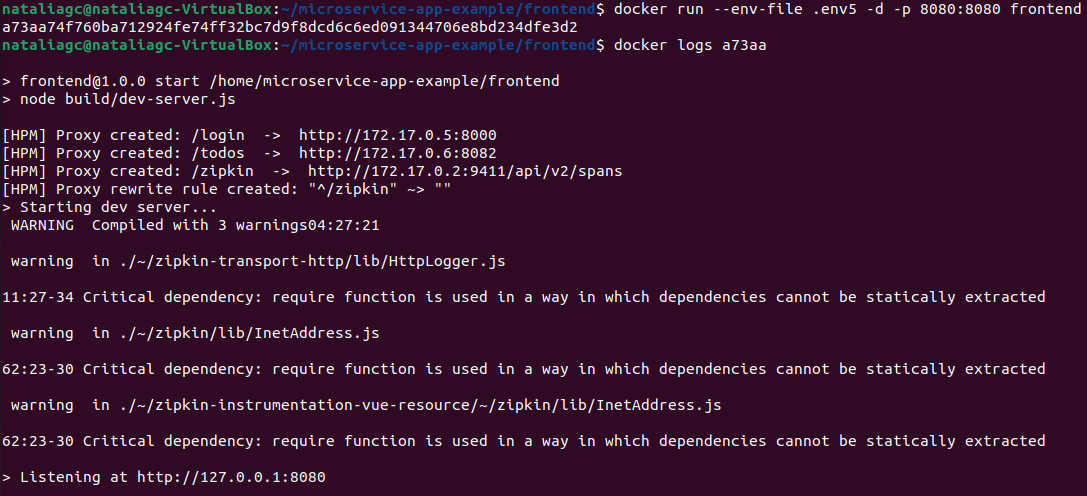
**Todos**

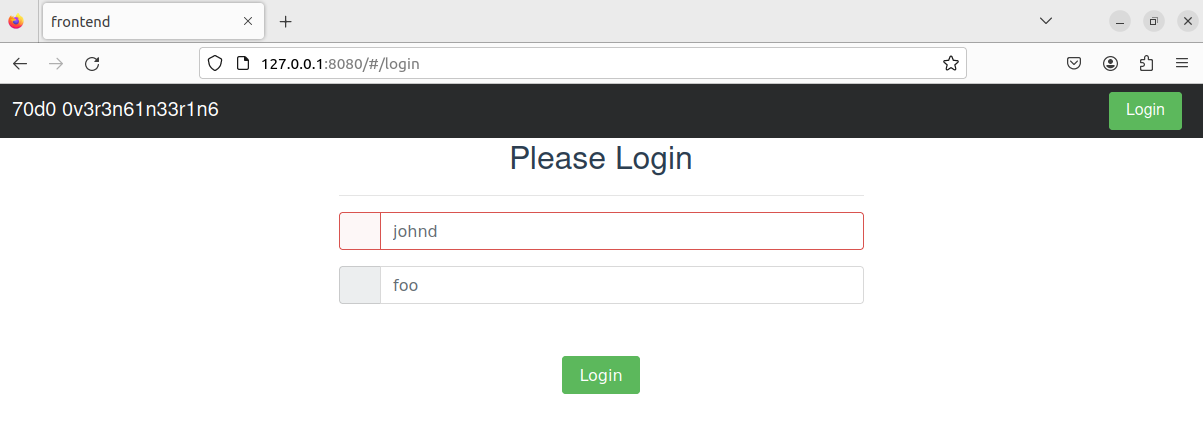
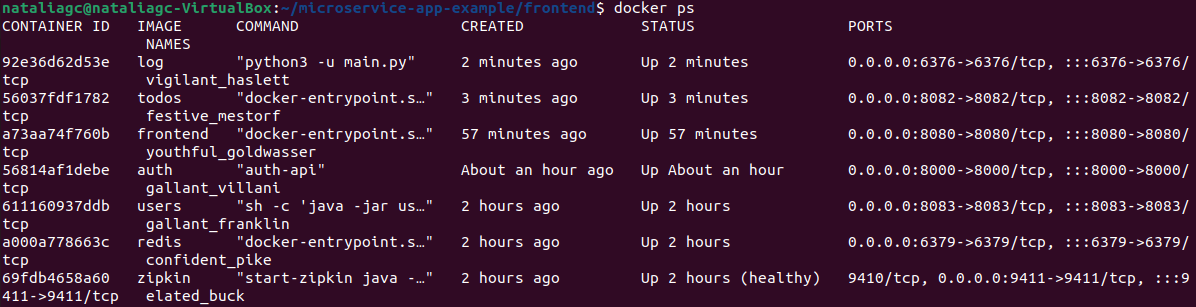
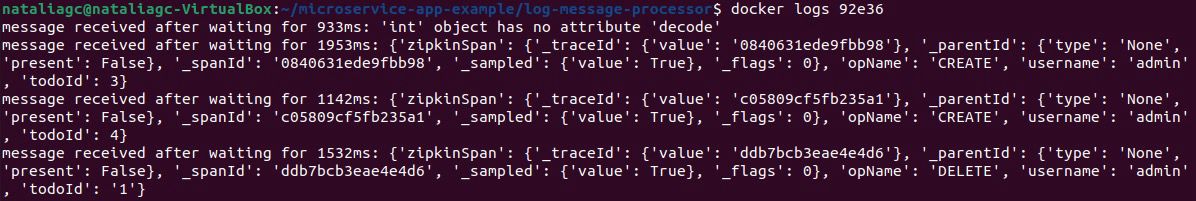
****

**Log**

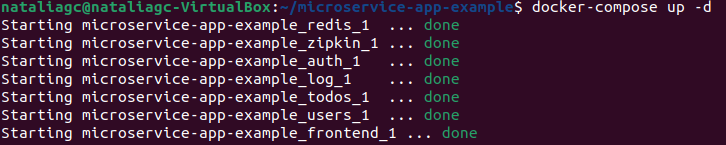
****

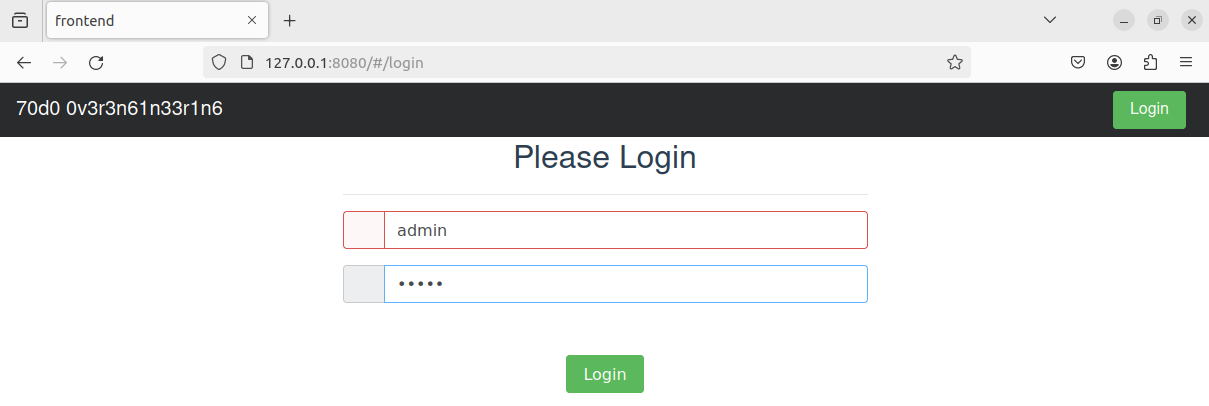
**Frontend**

****

****

1. **Finally, create a docker-compose file to run all the microservices at once.**



****

****