# Task 2

## Introduction

As a part of a cutting-edge technology team of an innovative company, I am responsible for designing, developing and securing a modern ecommerce-application that promises to shape the landscape of online shopping we do. My journey began with setting up the initial infrastructure for the project that includes creating a Kubernetes cluster on GKE (Google Kubernetes Engine) which provides us all the computing resources to host our application.

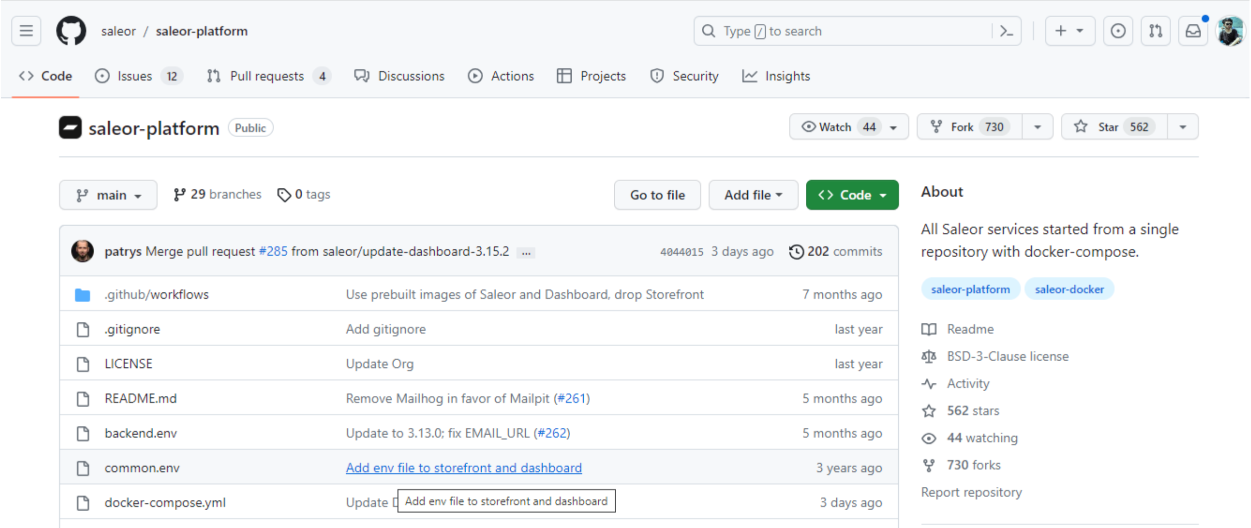
In my journey to create a modern ecommerce application, I now dive into creating the application and deploying it. My objectives in this phase is to be familiarize with the different components of the project such as Saleor API, Saleor Storefront, Saleor Dashboard and the platform. Next, forking the repository, running the application, tailoring the compose file and coming & pushing our modifications are my other objectives.

Following is the step-by-step guide of the process.

## Task 1: Forking the Saleor Platform Repository

Forking the project allows me to work on my own copy of the project. To fork the project:

a) Go to the repository



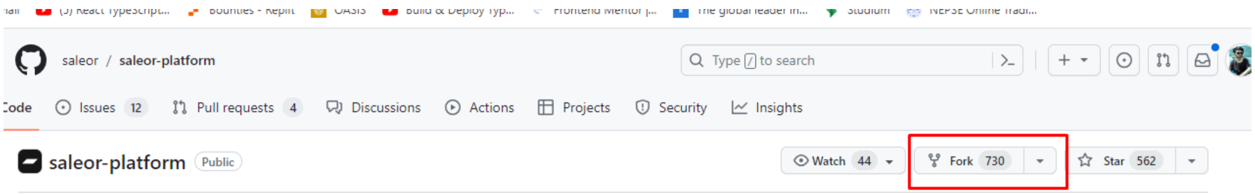
patrys Merge pull request #285 from saleor/update-dashboard-3.15.2 … 4044015 3 days ago 202 commits

| | | |
|---|---|---|
| .github/workflows | Use prebuilt images of Saleor and Dashboard, drop Storefront | 7 months ago |
| .gitignore | Add gitignore | last year |
| LICENSE | Update Org | last year |
| README.md | Remove Mailhog in favor of Mailpit (#261) | 5 months ago |
| backend.env | Update to 3.13.0; fix EMAIL_URL (#262) | 5 months ago |
| common.env | Add env file to storefront and dashboard | 3 years ago |
| docker-compose.yml | Update [ | 3 days ago |

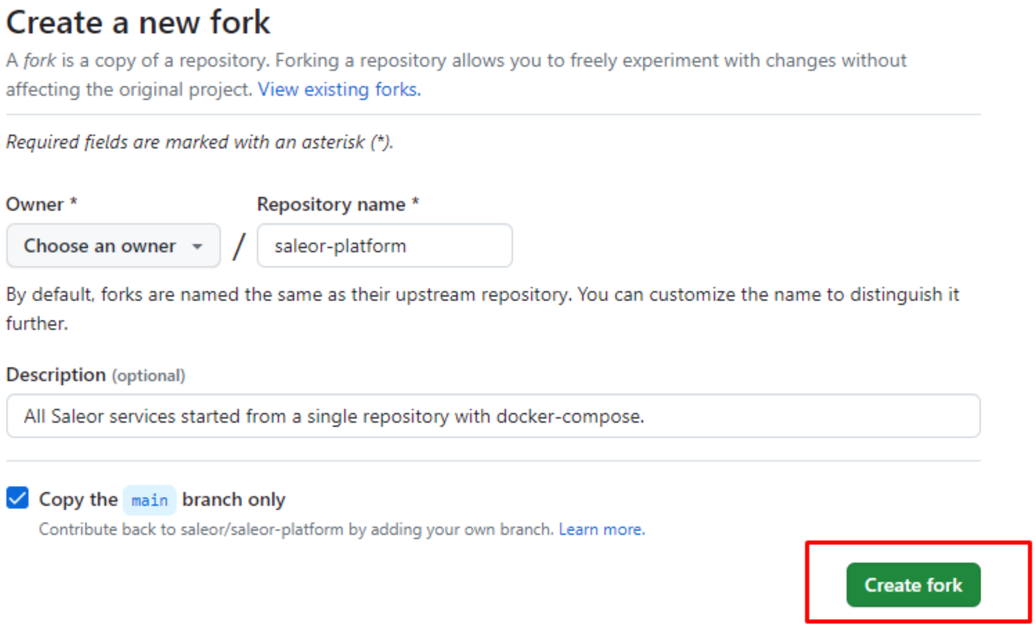
All Saleor services started from a single repository with docker-compose.

This is the GitHub repo of the Saleor Platform.

b) Click on "**Fork**" button.



c) Give the repository name and click **"Create Fork"**



Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. View existing forks.

*Required fields are marked with an asterisk (*).*

Owner *: Choose an owner / Repository name *: saleor-platform

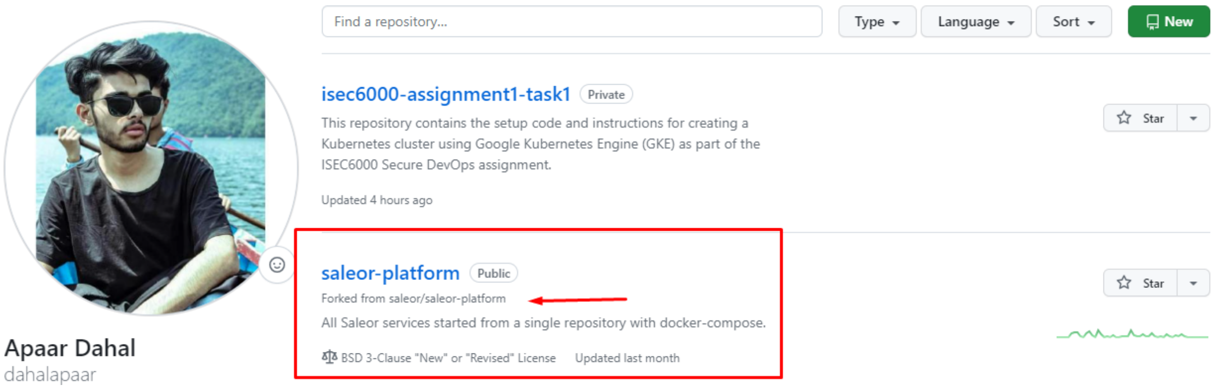
By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional): All Saleor services started from a single repository with docker-compose.

☑ Copy the `main` branch only

Contribute back to saleor/saleor-platform by adding your own branch. Learn more.

You can see your forked repository in your repositories section of your GitHub



isec6000-assignment1-task1 Private

This repository contains the setup code and instructions for creating a Kubernetes cluster using Google Kubernetes Engine (GKE) as part of the ISEC6000 Secure DevOps assignment.

Updated 4 hours ago

saleor-platform Public

Forked from saleor/saleor-platform

All Saleor services started from a single repository with docker-compose.

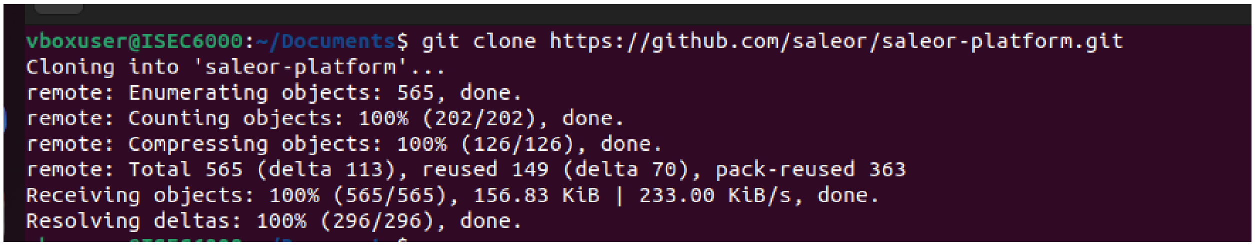
BSD 3-Clause "New" or "Revised" License Updated last month

Apaar Dahal
dahalapaar

### Task 2: Clone the repository

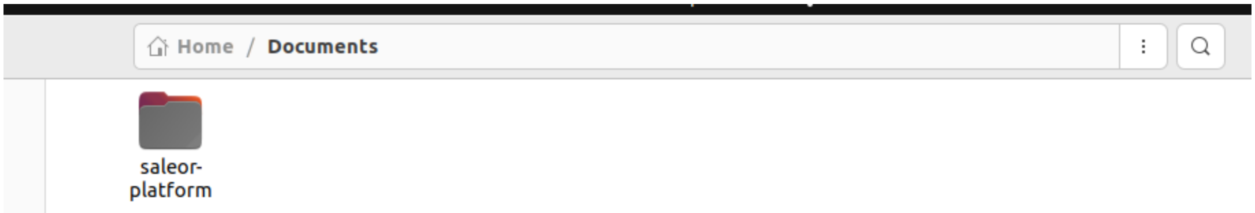
To clone the repository run the following command in your terminal:

**git clone https://github.com/saleor/saleor-platform.git**



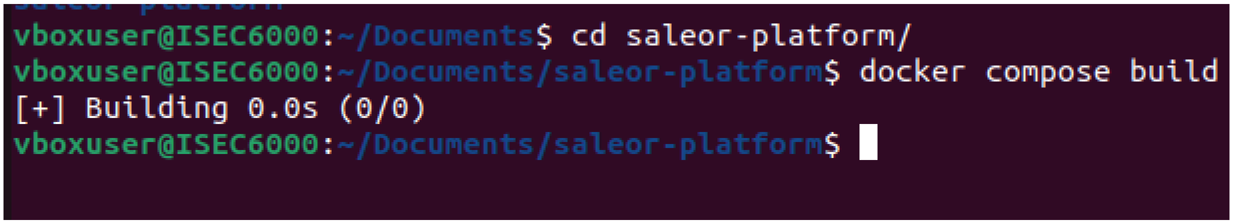
```
vboxuser@ISEC6000:~/Documents$ git clone https://github.com/saleor/saleor-platform.git
Cloning into 'saleor-platform'...
remote: Enumerating objects: 565, done.
remote: Counting objects: 100% (202/202), done.
remote: Compressing objects: 100% (126/126), done.
remote: Total 565 (delta 113), reused 149 (delta 70), pack-reused 363
Receiving objects: 100% (565/565), 156.83 KiB | 233.00 KiB/s, done.
Resolving deltas: 100% (296/296), done.
```

Now, you can see the cloned project on your computer



### Task 3: Build the application

Go to the cloned directory and type “docker compose build” command in your terminal and hit enter. This command will build the Docker images for the services defined in the Docker Compose file.

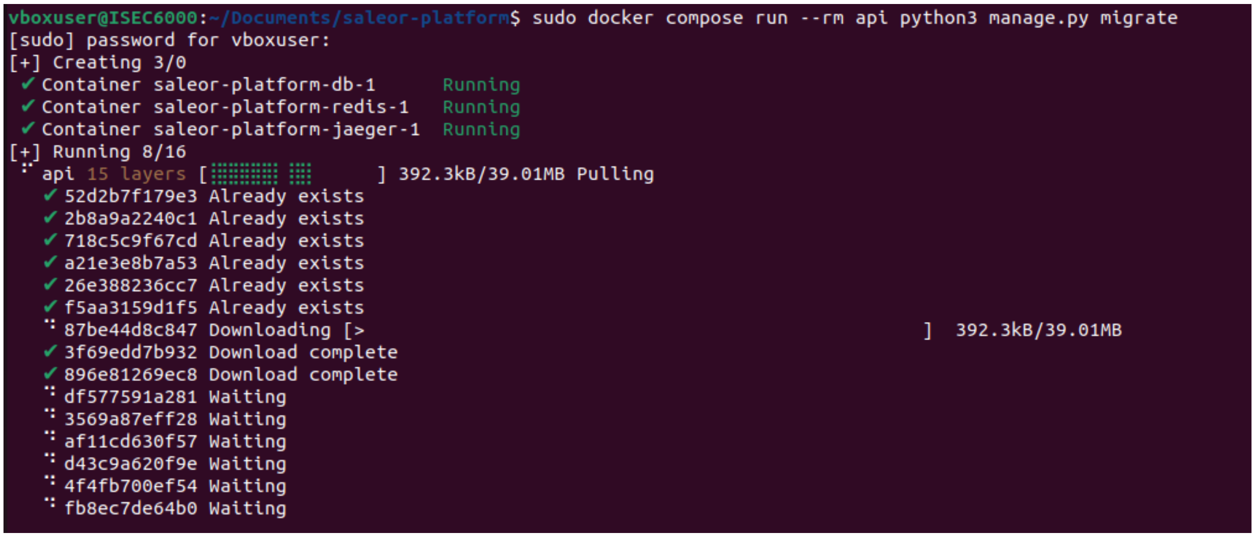


```
vboxuser@ISEC6000:~/Documents$ cd saleor-platform/
vboxuser@ISEC6000:~/Documents/saleor-platform$ docker compose build
[+] Building 0.0s (0/0)
vboxuser@ISEC6000:~/Documents/saleor-platform$ 
```

### Task 4: Apply Django Migration

In this step, we are synchronizing the database structure with the changes we have made to our Django database models. To do so, run the following command in your terminal.

**docker compose run --rm api python3 manage.py migrate**



```
vboxuser@ISEC6000:~/Documents/saleor-platform$ sudo docker compose run --rm api python3 manage.py migrate
[sudo] password for vboxuser:
[+] Creating 3/0
 ✔ Container saleor-platform-db-1      Running
 ✔ Container saleor-platform-redis-1   Running
 ✔ Container saleor-platform-jaeger-1  Running
[+] Running 8/16
 ⠙ api 15 layers [⣿⣿⣿⣿⣿⣿ ⣿⣿      ] 392.3kB/39.01MB Pulling
   ✔ 52d2b7f179e3 Already exists
   ✔ 2b8a9a2240c1 Already exists
   ✔ 718c5c9f67cd Already exists
   ✔ a21e3e8b7a53 Already exists
   ✔ 26e388236cc7 Already exists
   ✔ f5aa3159d1f5 Already exists
   ⠙ 87be44d8c847 Downloading [>                                                  ]  392.3kB/39.01MB
   ✔ 3f69edd7b932 Download complete
   ✔ 896e81269ec8 Download complete
   ⠙ df577591a281 Waiting
   ⠙ 3569a87eff28 Waiting
   ⠙ af11cd630f57 Waiting
   ⠙ d43c9a620f9e Waiting
   ⠙ 4f4fb700ef54 Waiting
   ⠙ fb8ec7de64b0 Waiting
```

### Task 5: Populate the database with example data and create the admin user

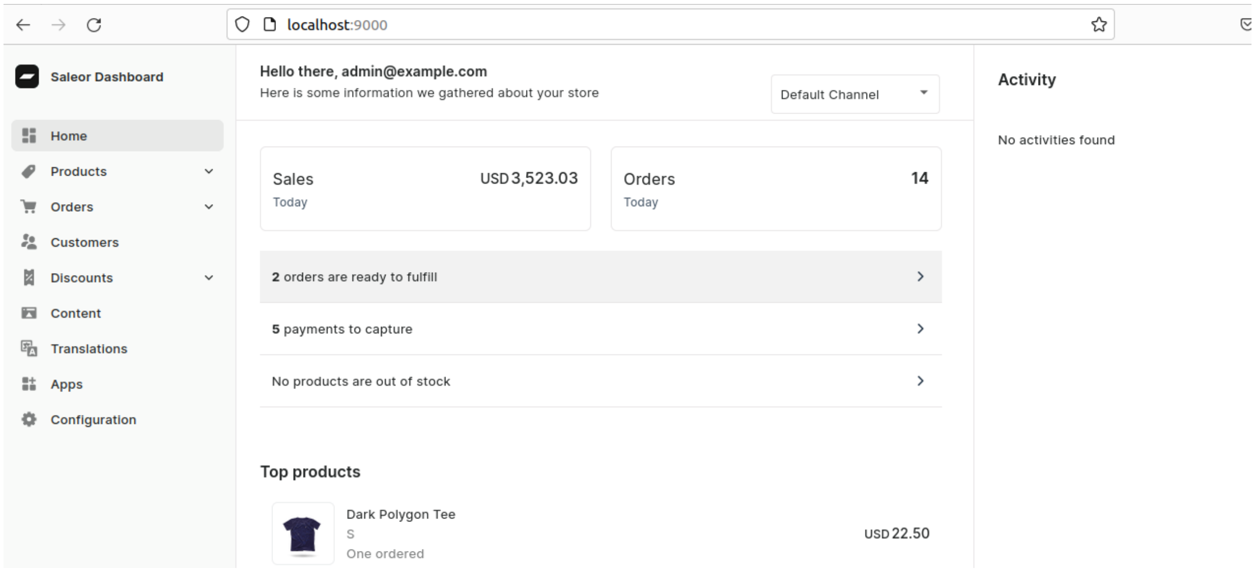
To populate the database, type the following command in your terminal

```
docker compose run --rm api python3 manage.py populatedb –createsuperuser
```

### Task 6: Run the application

To run the services of application, type docker compose up on your terminal.

The application is ready. Dashboard will be running at localhost:9000 port. Similarly, API and UI will run in port 8000 and 16686 respectively.



Hello there, admin@example.com
Here is some information we gathered about your store

Sales USD 3,523.03
Today

Orders 14
Today

2 orders are ready to fulfill

5 payments to capture

No products are out of stock

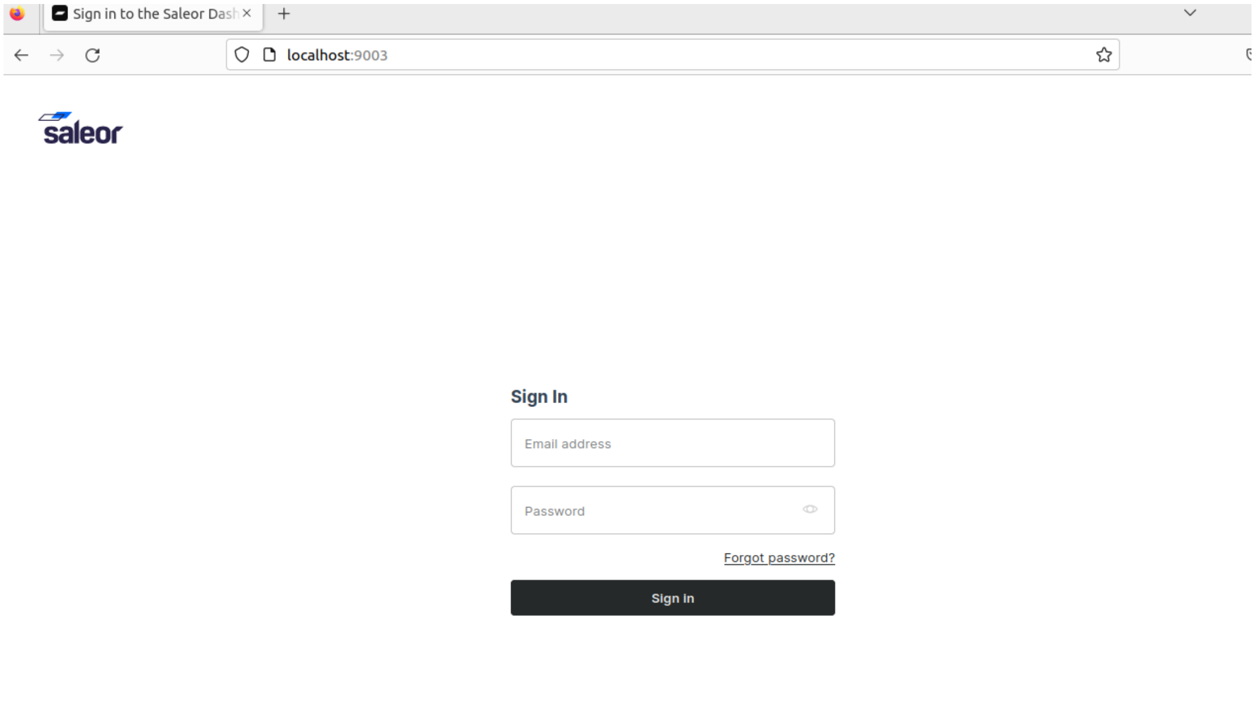
No activities found

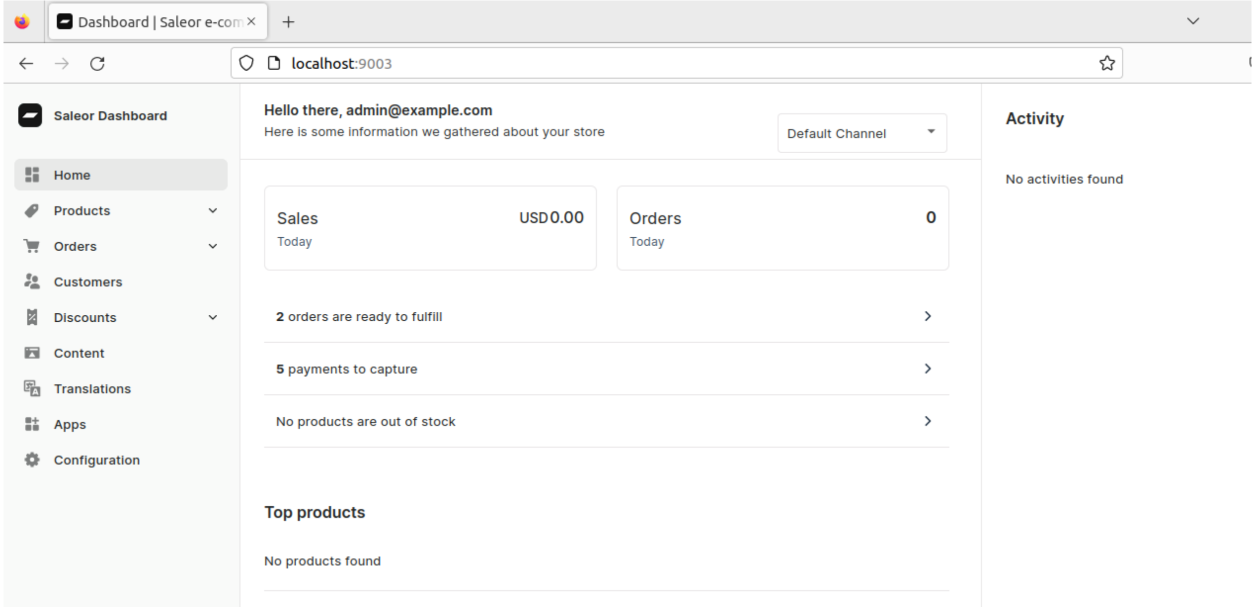
Dark Polygon Tee
S
One ordered
USD 22.50

### Changing ports as our scenario.

As a DevOps team member my responsibility is to run the application in different ports. The dashboard of application should run in port 9003.

To do so we have to make some changes in our yml file of the project.





Hello there, admin@example.com

Here is some information we gathered about your store

| | |
|---|---|
| Sales<br>Today | USD0.00 |
| Orders<br>Today | 0 |

2 orders are ready to fulfill

5 payments to capture

No products are out of stock

Top products

No products found

No activities found

We can see above that our dashboard is running in 9003 port which was required to do as a task.

NOTE: THE LINK TO MY GITHUB REPO IS https://github.com/dahalapaar/isec6000-assignment1-task2