**SPRINT2 PROJECT**

**DOCUMENTATION**

**CAB BOOKING APPLICATION**

**(AWS+DOCKER+KUBERNATES)**

TEAM MEMBERS

ARYAN BAMPAL(TL)

CHANDRIKA YERRAMSETTI (P0)

SOUMYA BODDUPALLY

ANVAY SHINDE

**Software Installations**

**Docker:**

Docker is an open-source containerization platform. It enables developers to package applications into containers—standardized executable components combining application source code with the operating system (OS) libraries and dependencies required to run that code in any environment.

Installation steps for docker:

1.Download Docker:

<https://docs.docker.com/desktop/windows/install/>

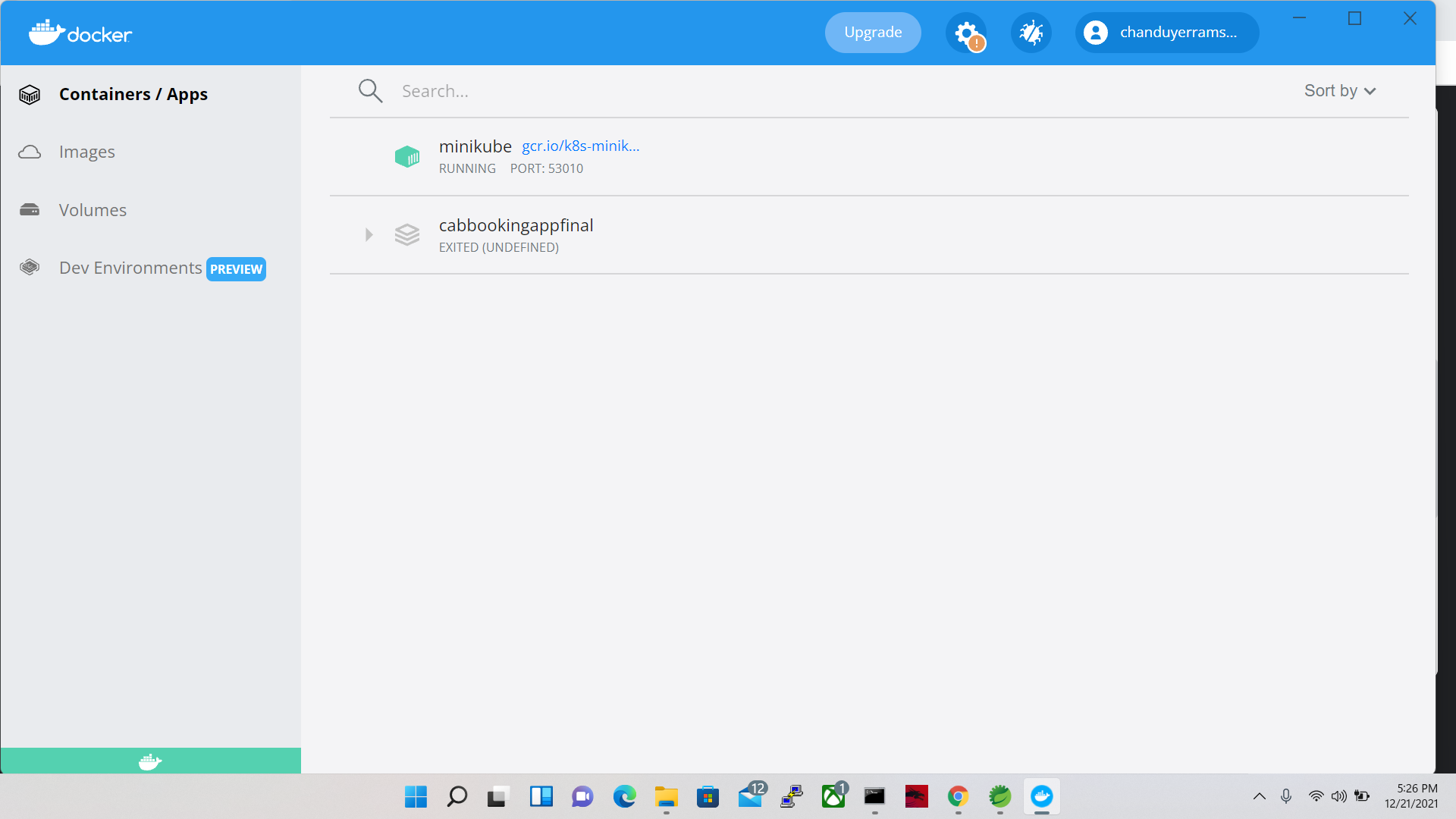
2.Double –click Install Docker.

3.Follow the install wizard: accept the license, authorize the installer, and proceed with the install

4.Click finish to launch Docker.

5.Docker starts automatically.

6.Docker loads a Welcome windows giving you tips and access to the Docker documentation.



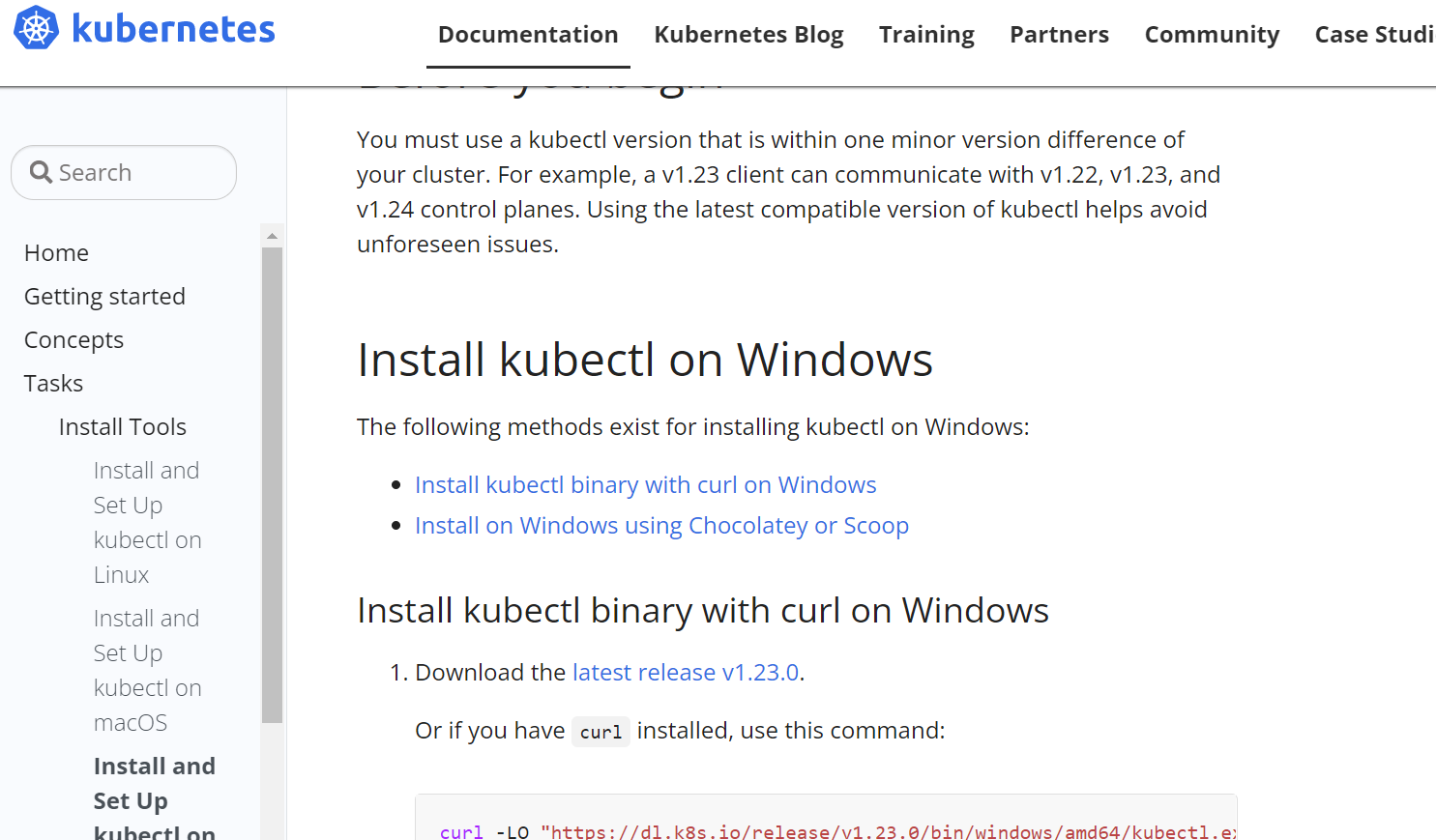
**Kubernetes:**

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation

1.To Install Kubernetes on windows go to this website:

<https://kubernetes.io/docs/tasks/tools/install-kubectl-windows/>

2.Then click on latest version v1.23.0.to download the Kubernetes.



3.Add the path in Environment variables.

Graphical user interface, text, application

Description automatically generated

4. Test to ensure the version of kubectl is the same as downloaded.

$ kubectl version –client

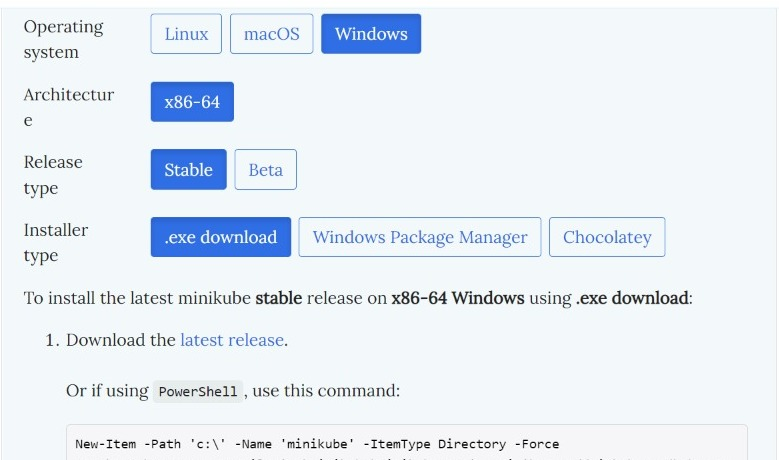
**MINIKUBE:**

Minikube is a tool that lets you run Kubernetes locally. Minikube runs a single-node Kubernetes cluster on your personal computer (including Windows, macOS and Linux PCs) so that you can try out Kubernetes, or for daily development work.

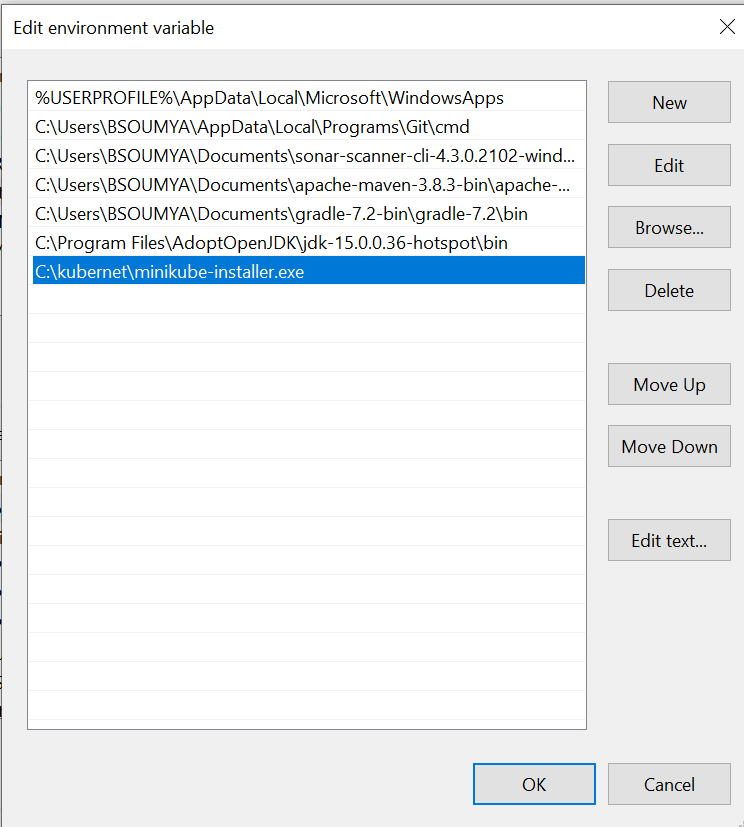
1. To Install minikube on windows go to this website

[**https://minikube.sigs.k8s.io/docs/start/**](https://minikube.sigs.k8s.io/docs/start/)

2.Then click on latest release and download



3. Add the path in Environment variables



4.Start your cluster by using below command

$ minikube start

5.Then it will display as shown below in the picture

**Text

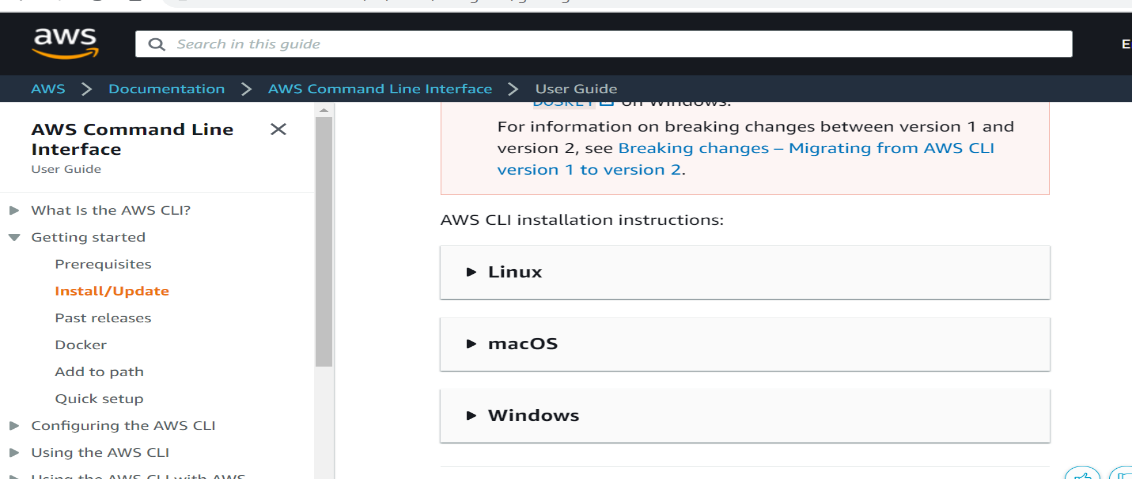
Description automatically generated**

AWSCLI:

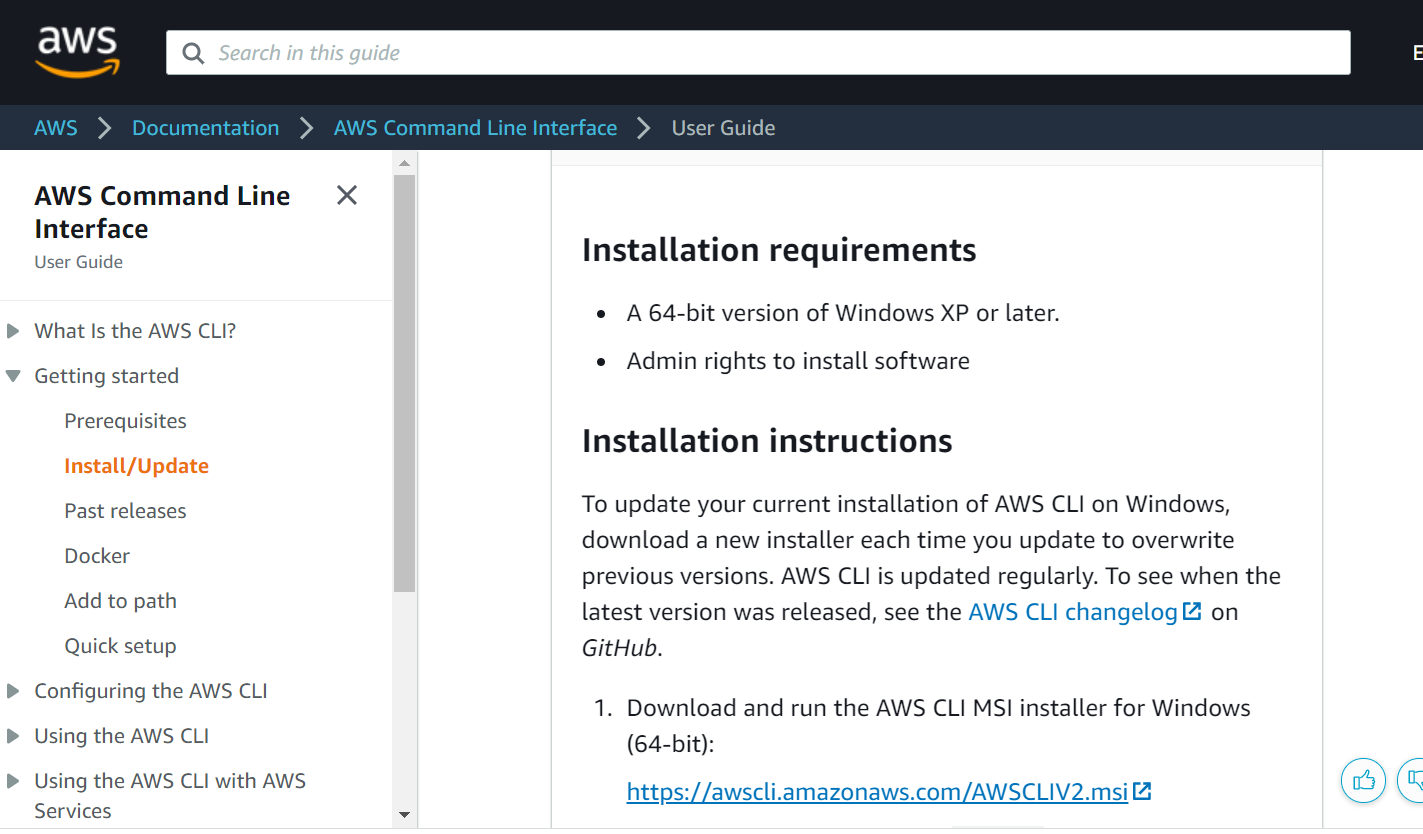
1. To Install AWSCLI on windows go to this website:

<https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-getting->

2.Then go to install and click on windows.



3. click on link to download the awscli



4. Follow the install wizard: accept the license, authorize the installer, and proceed with the install

5. Click finish to launch AWSCLI.

6. To confirm the installation, open the Start menu, search for cmd to open a command prompt window, and at the command prompt use the aws --version command.

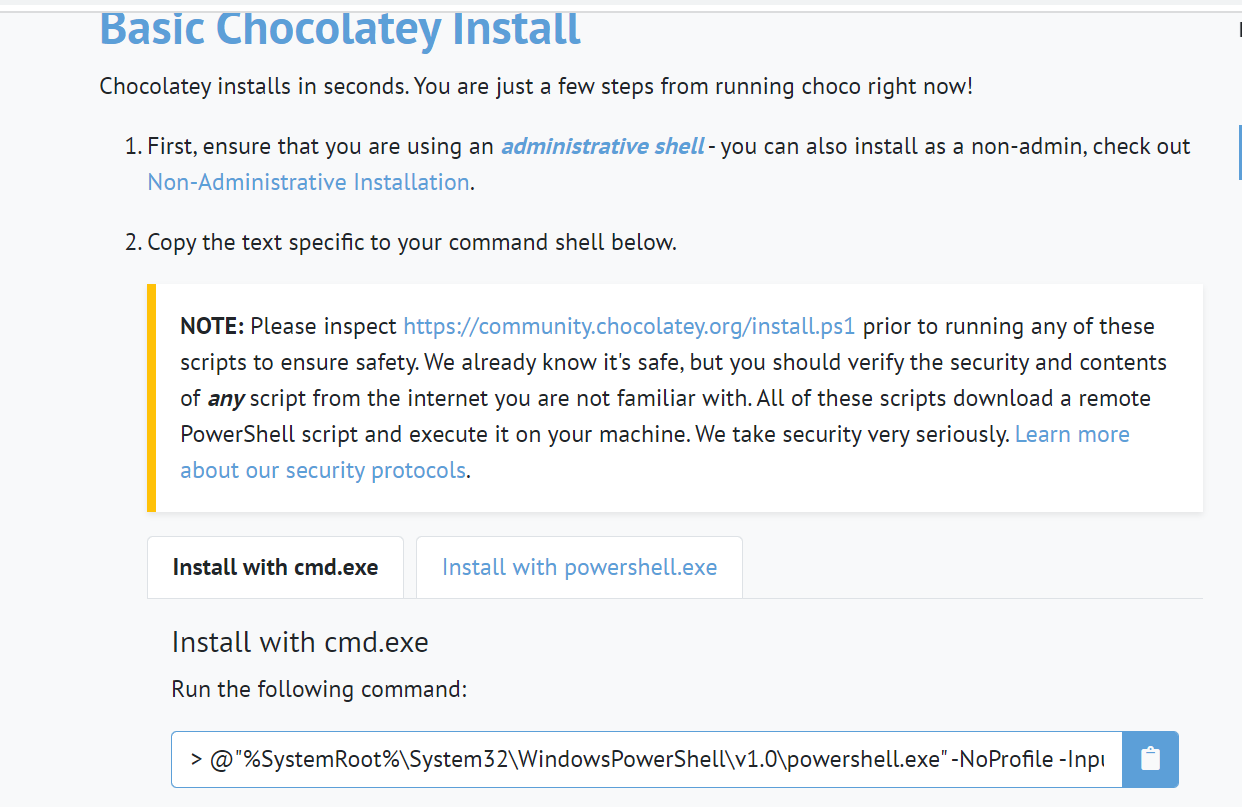
**CHOCOLATEY:**

Chocolatey is a software management solution that gives you the freedom to create a simple software package and then deploy it anywhere you have Windows using any of your familiar configuration or system management tools

1.Goto chocolatey website:

https://chocolatey.org/install

2.click on take the installation course.



3. Paste the copied text into your shell and press Enter.

4.Wait a few seconds for the command to complete.

5.Then upgrade by using the below command:

$ choco upgrade chocolatey

**EKSCTL:**

Eksctl is a tool jointly developed by AWS and Weave works that automates much of the experience of creating EKS clusters. In this module, we will use eksctl to launch and configure our EKS cluster and nodes

1.To install eksctl go to these AWS website and open Amazon EMR on EKS Development Guide.

2. If you do not already have Chocolatey installed on your Windows system, see [Installing Chocolatey](https://chocolatey.org/install)

3. Install or upgrade eksctl

$ choco install -y eksctl

4. If they are already installed, run the following command to upgrade:

$ choco upgrade -y eksctl

5. Test that your installation was successful with the following command.

$ eksctl version

**Cab booking Application**

**Procedure:**

1. **Steps To create Docker file:**

A) Create project.

B) Installing Docker

C) Create a file called Docker File.

D) Build your Docker File using properties.

E) Save the project.

**2. Creating Docker Image**

A) Create a Base Container

B) Inspect Containers

C) Start the Container

D) Create an Image from a Container by using below command.

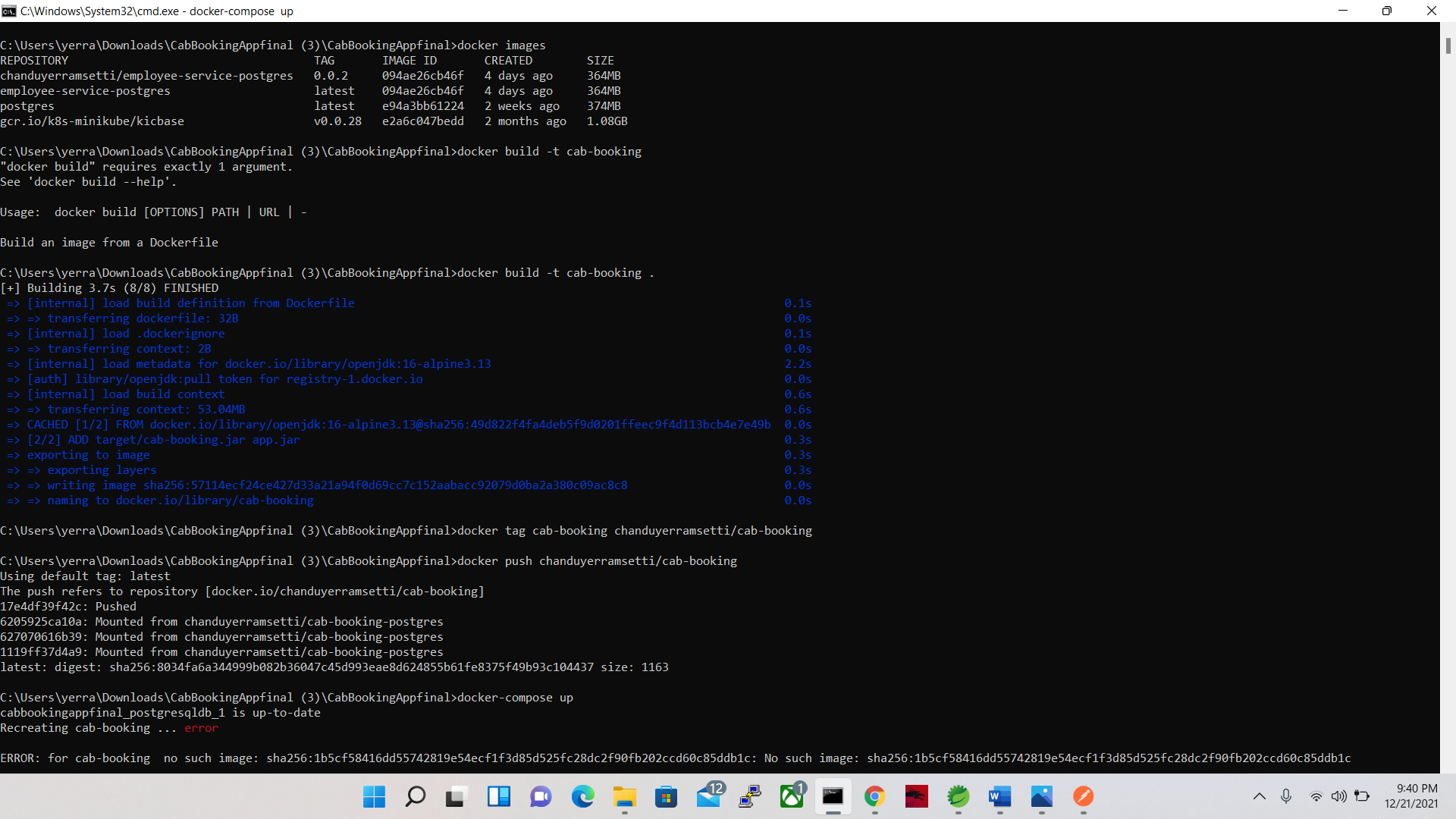
docker build -t cab-booking  **.**

3.Tagging Image

docker tag cab-booking chanduyerramsetti/cab-booking

4.push that image into docker hub using below command.

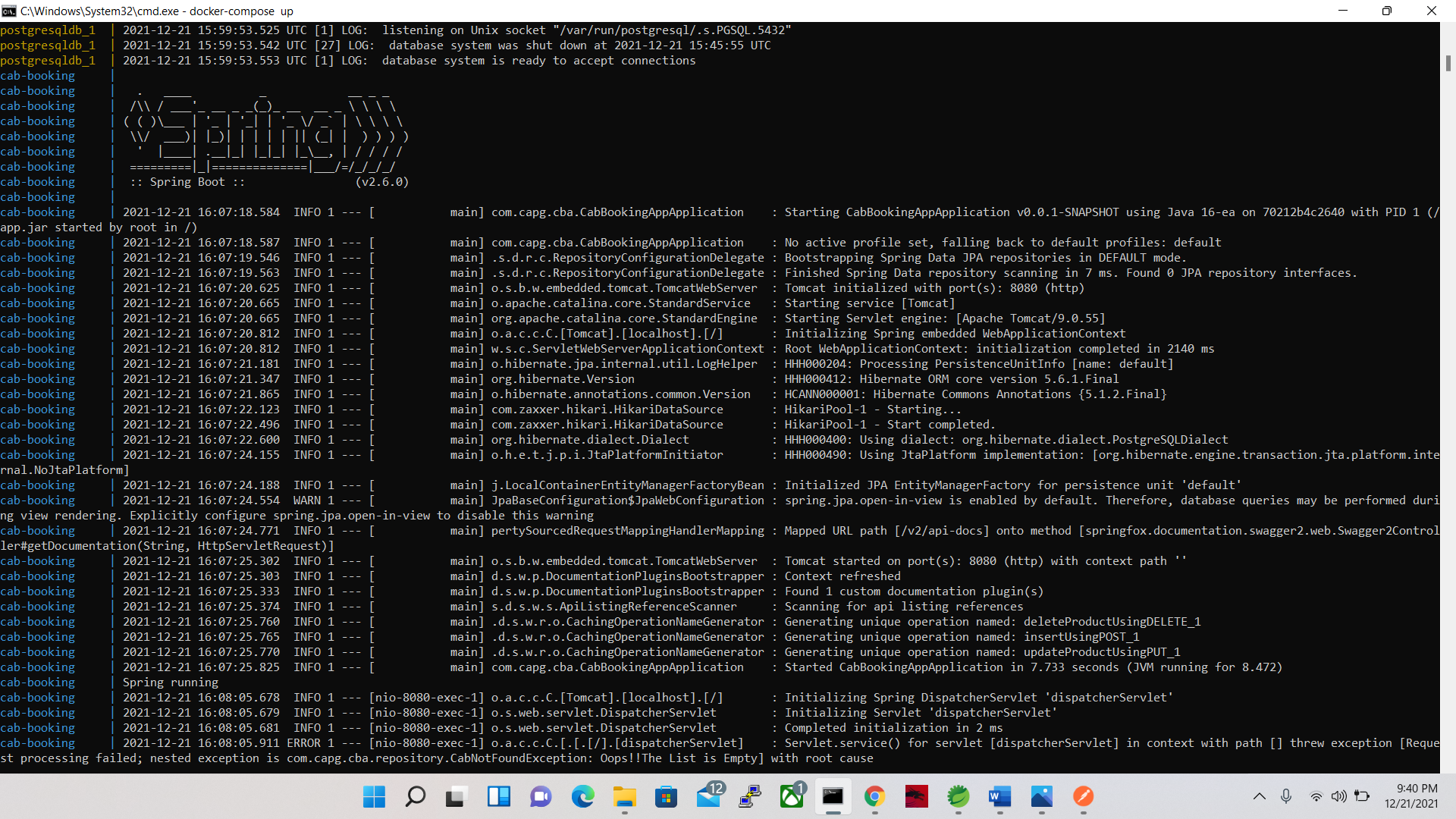
docker push chanduyerramsetti/cab-booking

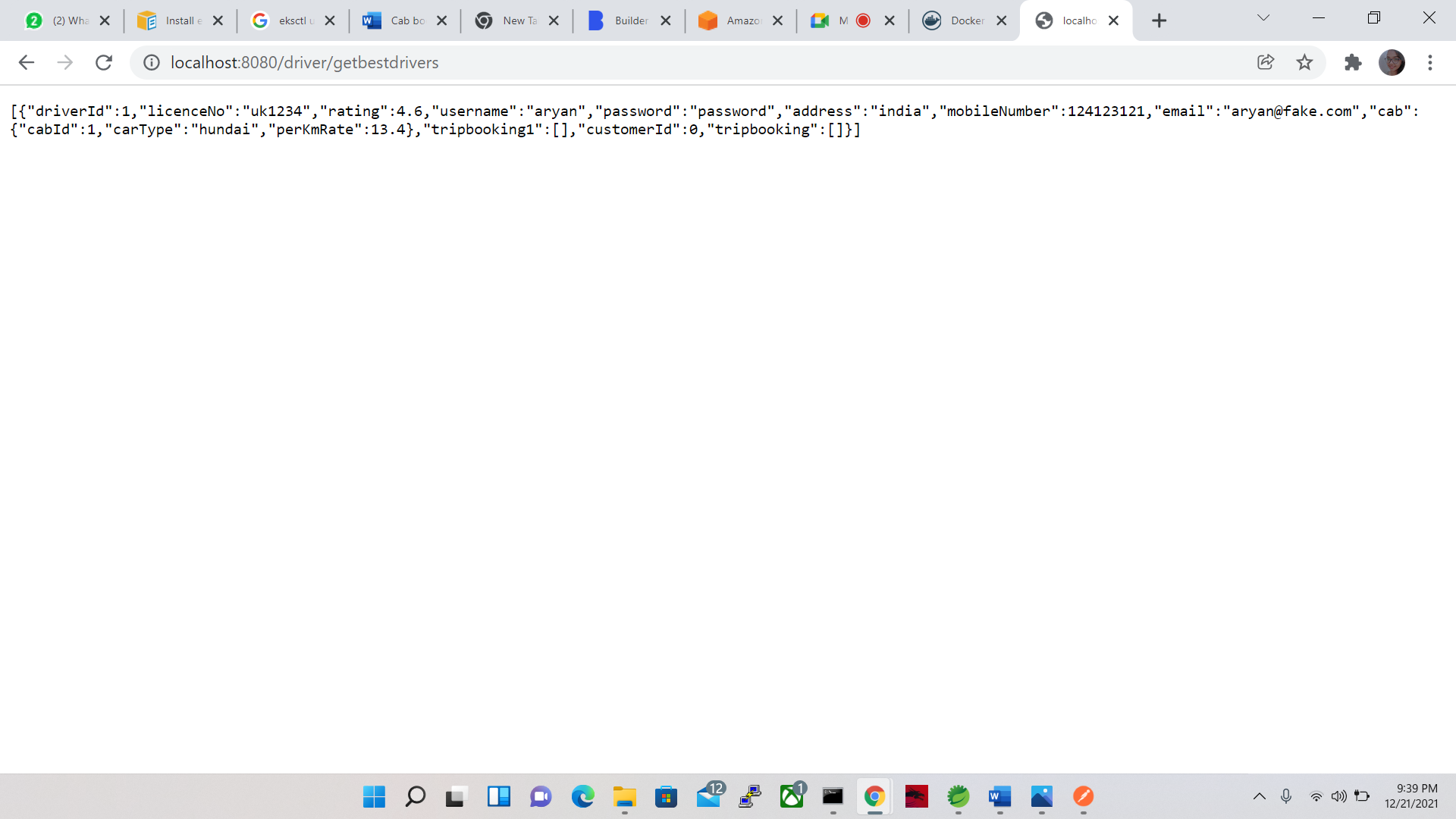


5. Steps to create docker-compose file

1. A) Create the docker-compose.yml
2. B) Define services in a Compose file
3. C) Run the application with compose using the below command

docker-compose up





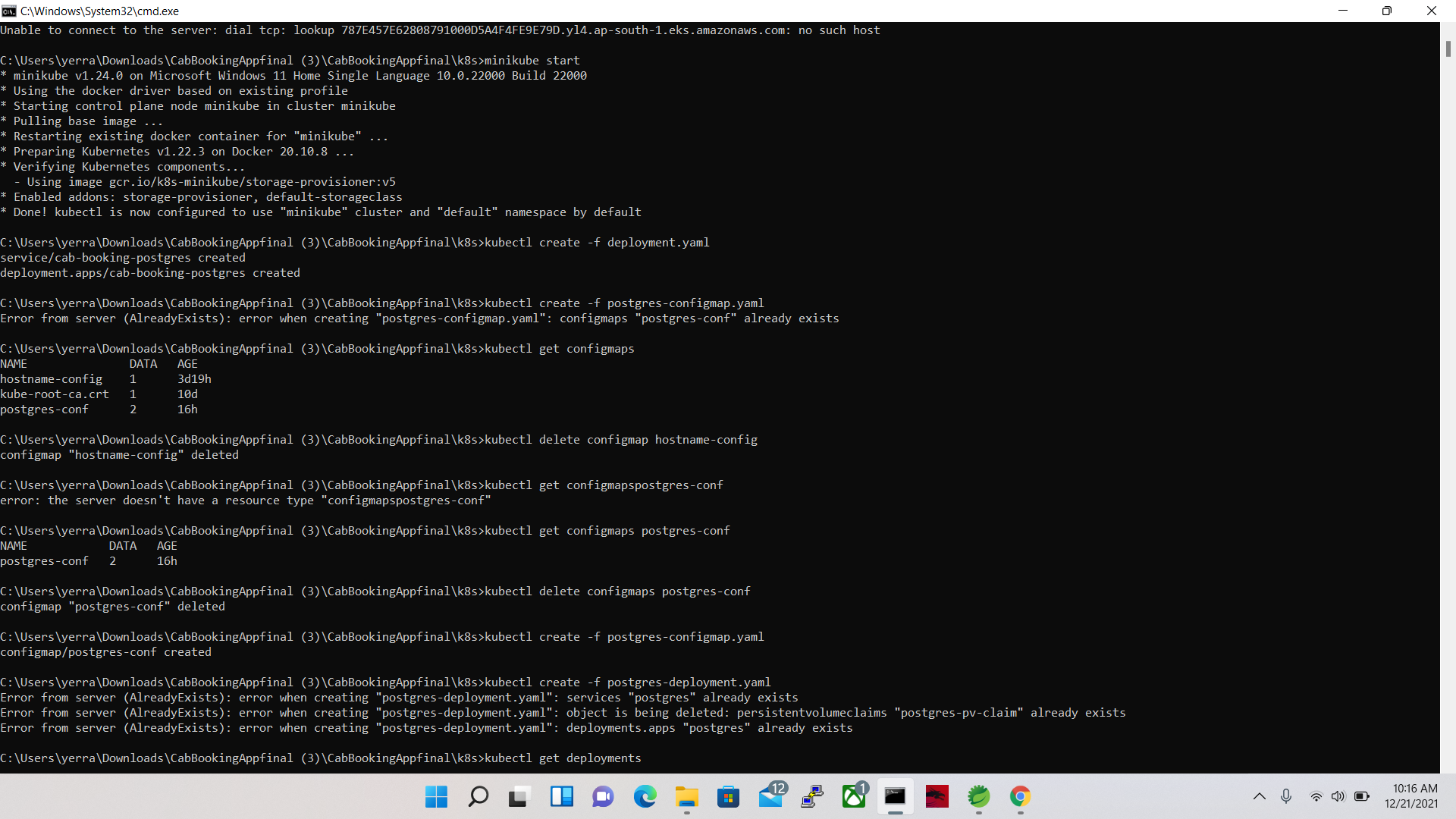
**Deploying the application on Kubernetes environment**

1) In Application file

A) change the application properties

B) write yaml files

* deployment.yaml
* postgres-credentials.yaml
* postgres-configmap.yaml
* postgres-deployment.yaml
* Build the jar
* In command prompt start the minikube

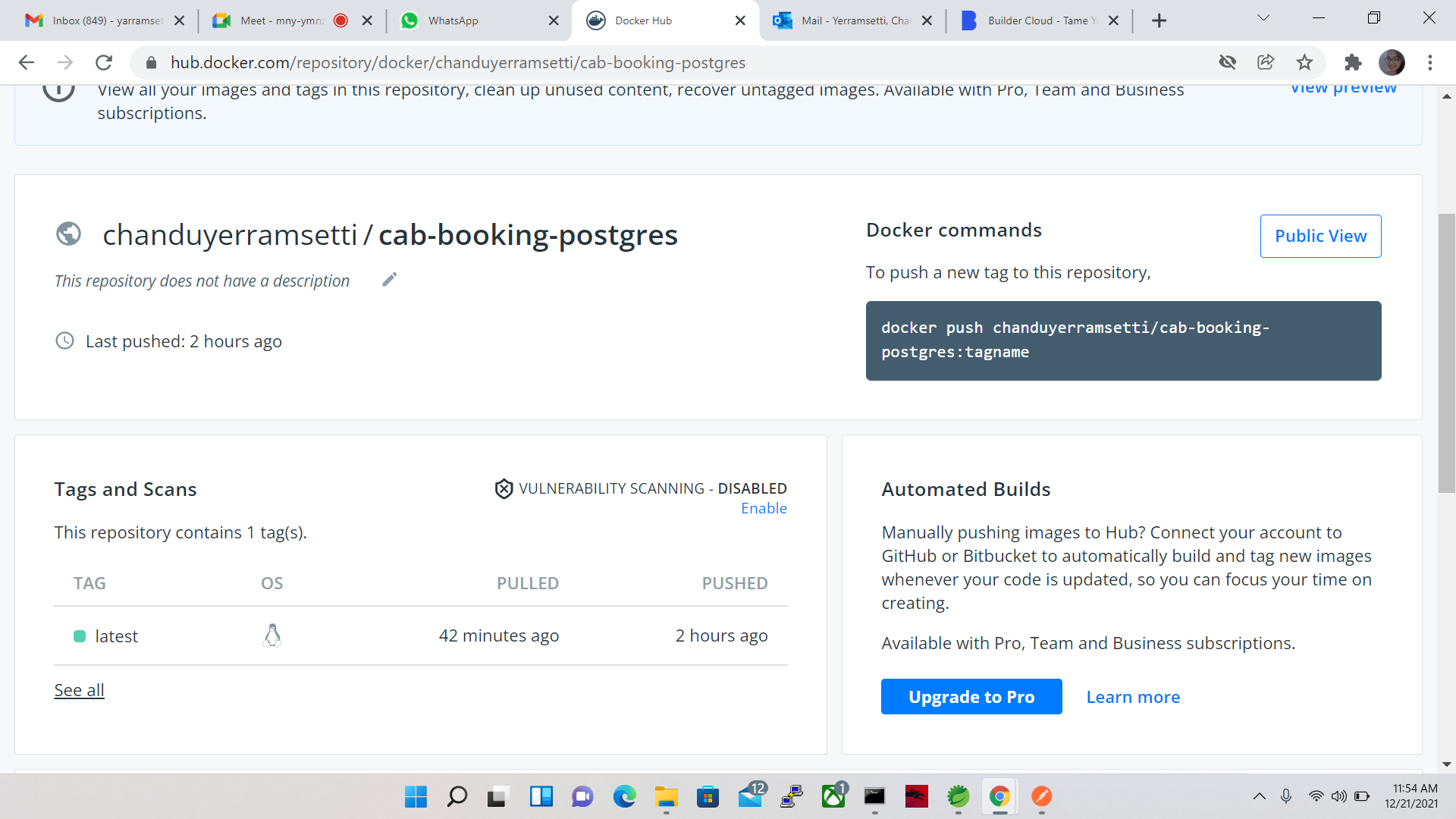


2) Create Docker Image, tag the image and push the image into Docker Hub

Image: docker build -t cab-booking-postgres.

Tag: docker tag cab-booking chanduyerramsetti/cab-booking-postgres

Push: docker push chanduyerramsetti/cab-booking-postgres

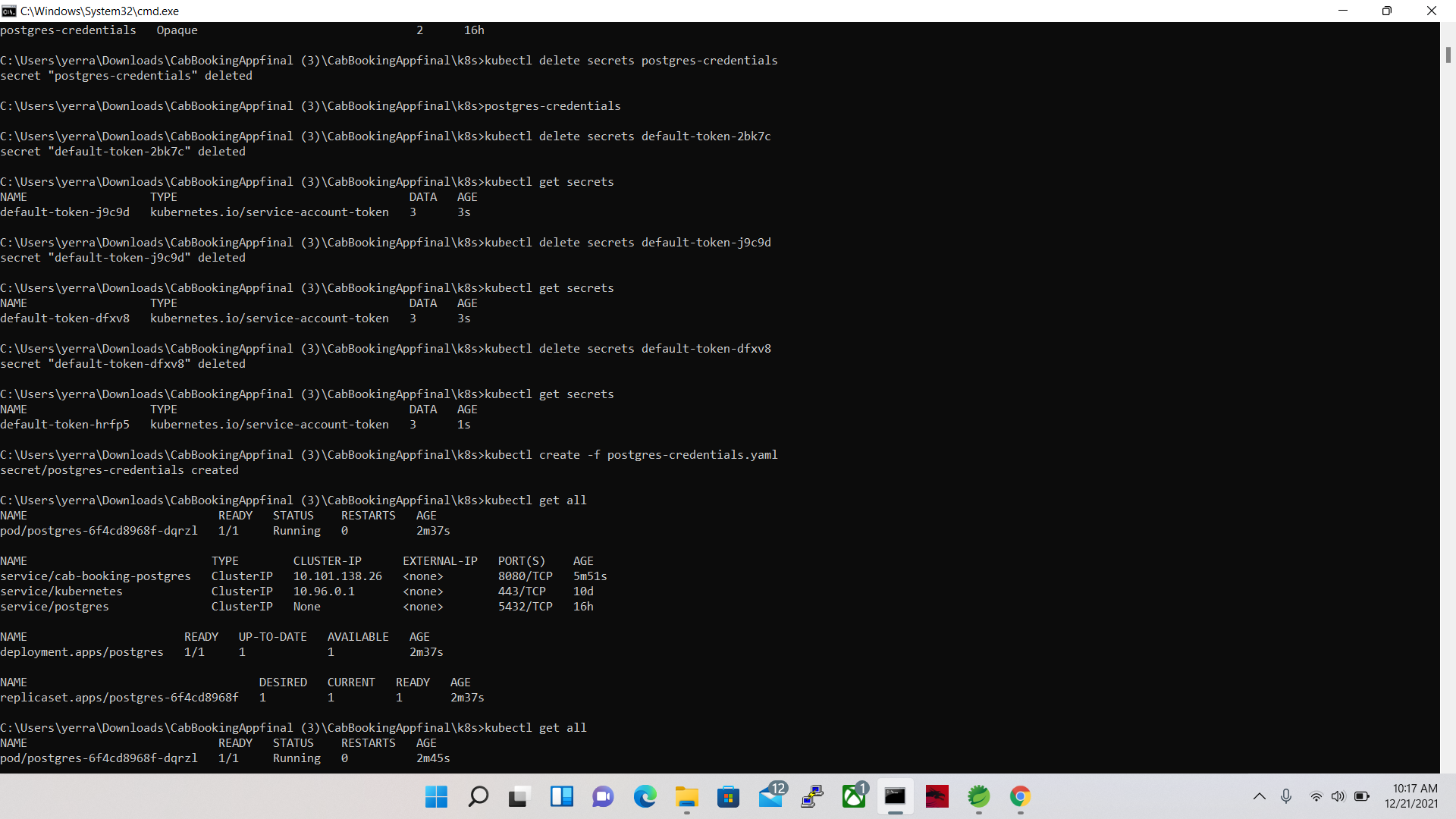


3) Change the directory to k8s by using the below command

Cd k8s

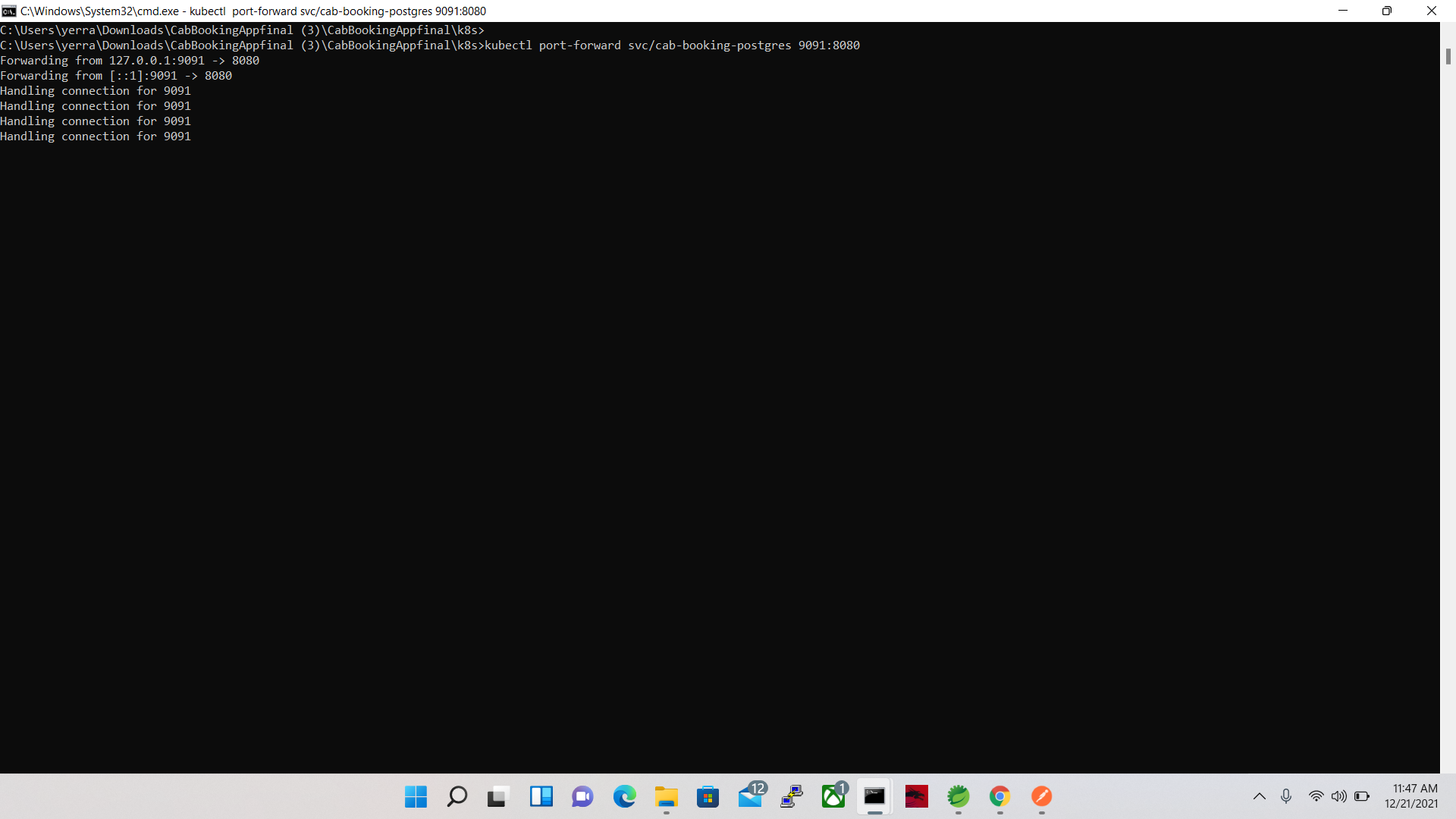
4) Now, Add the below deployment commands

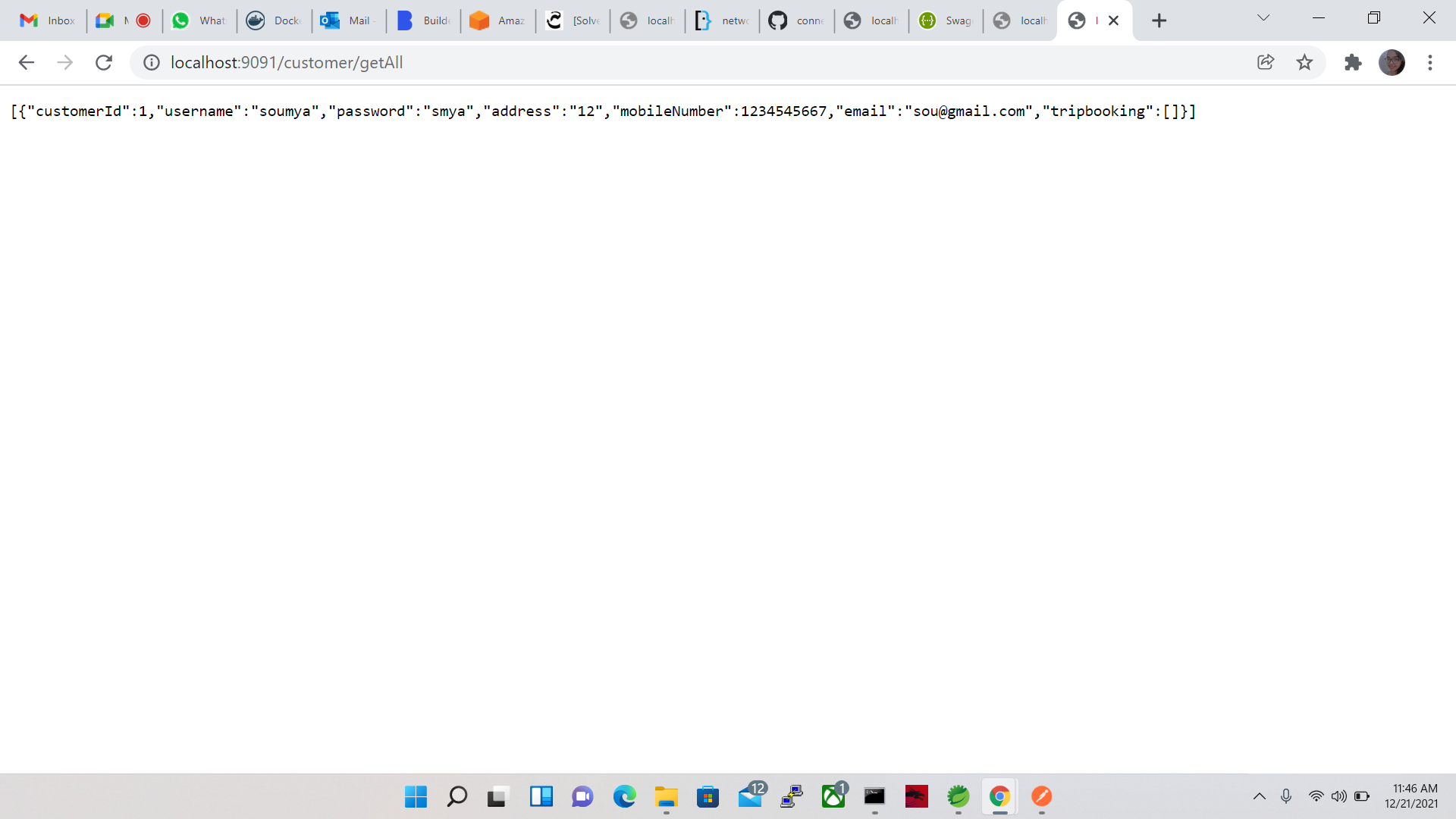
1. kubectl create -f deployment.yaml
2. kubectl create -f postgres-credentials.yaml
3. kubectl create -f postgres-configmap.yaml
4. kubectl create -f postgres-deployment.yaml



5) After successful deployment forward the port

kubectl port-forward svc/cab-booking 9091:8080

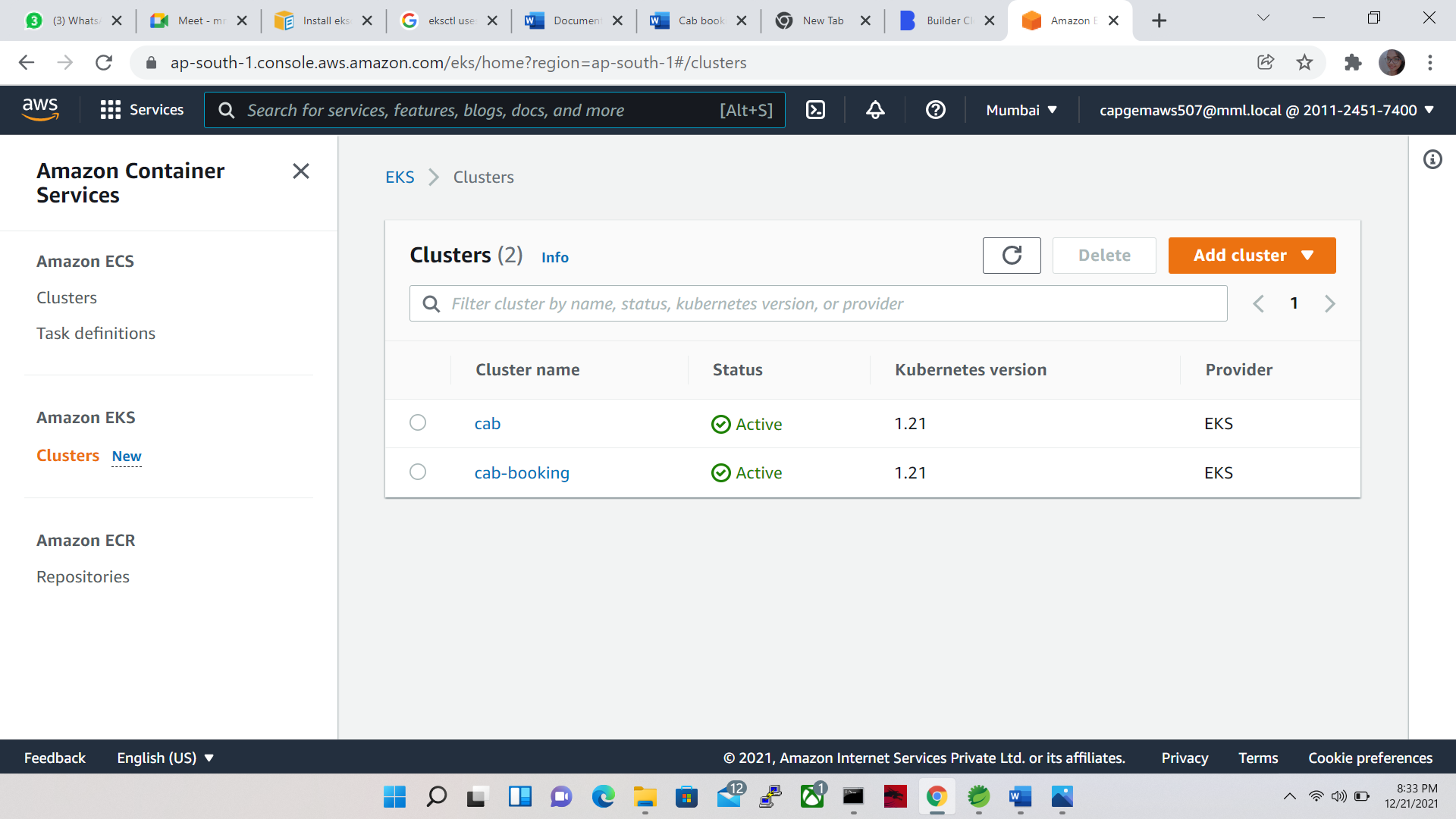


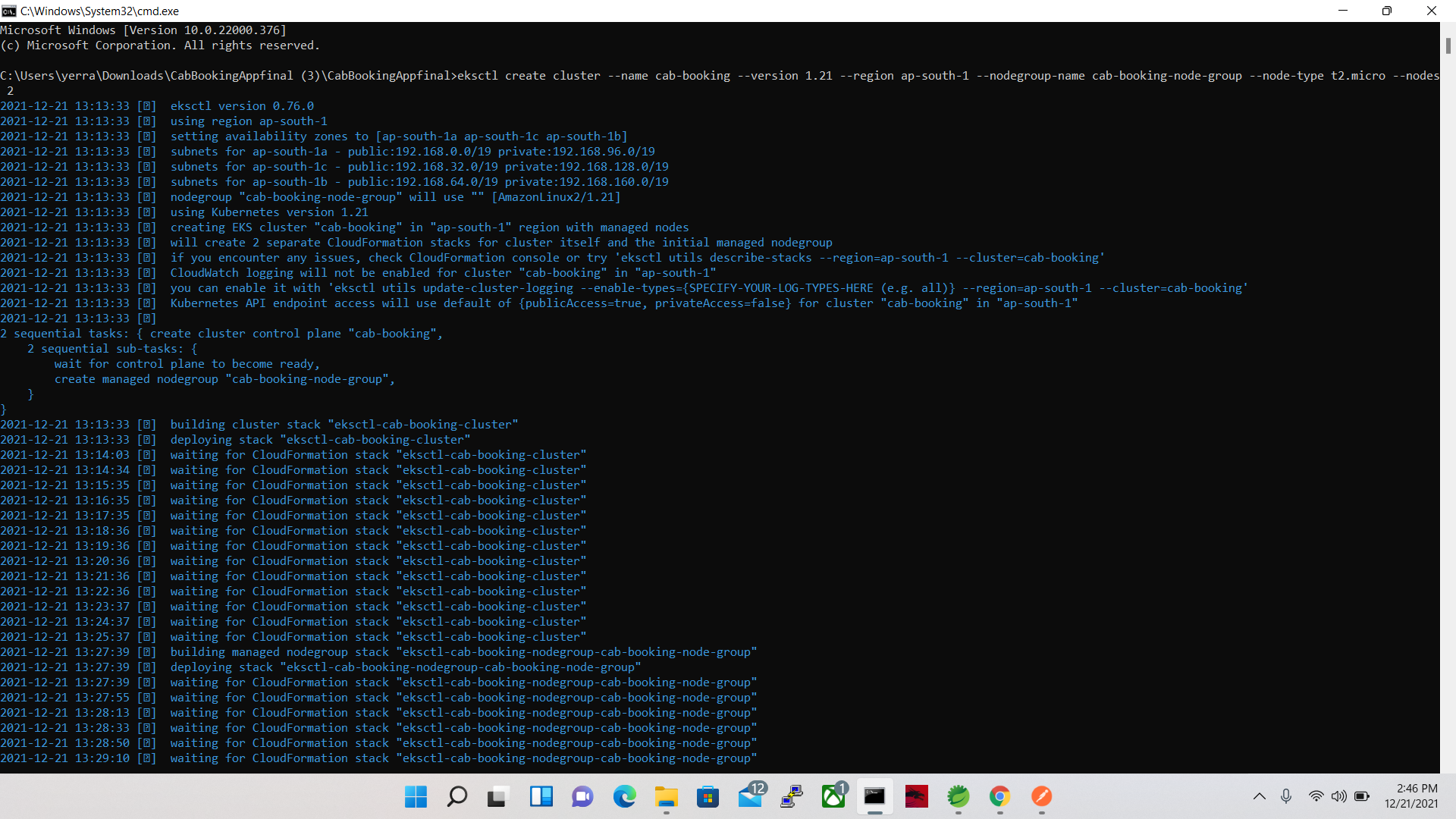


**Deploy the application on EKS cluster:**

1) Create a cluster in EKS with eksctl command

* eksctl create cluster --name cab-booking --version 1.21 --region ap-south-1 --nodegroup-name cab-booking -node-group --node-type t2.micro --nodes 2
* Aws eks --region ap-south-1 update-kubeconfig --name cab-booking



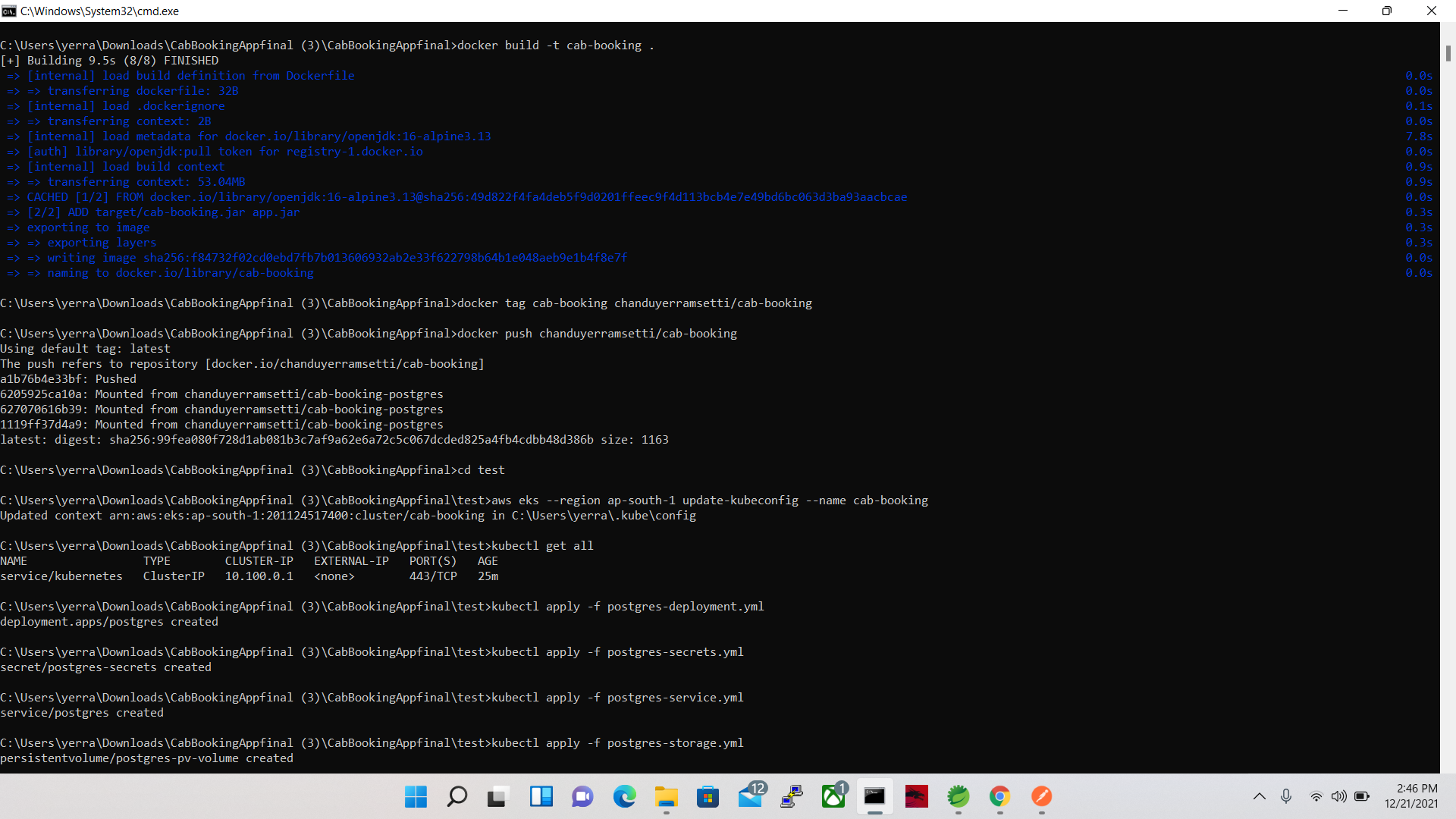


2) Create the docker image and push the image on docker hub

3) Change the directory to test by cd test command

4) Deploying the application on eks cluster using the following commands

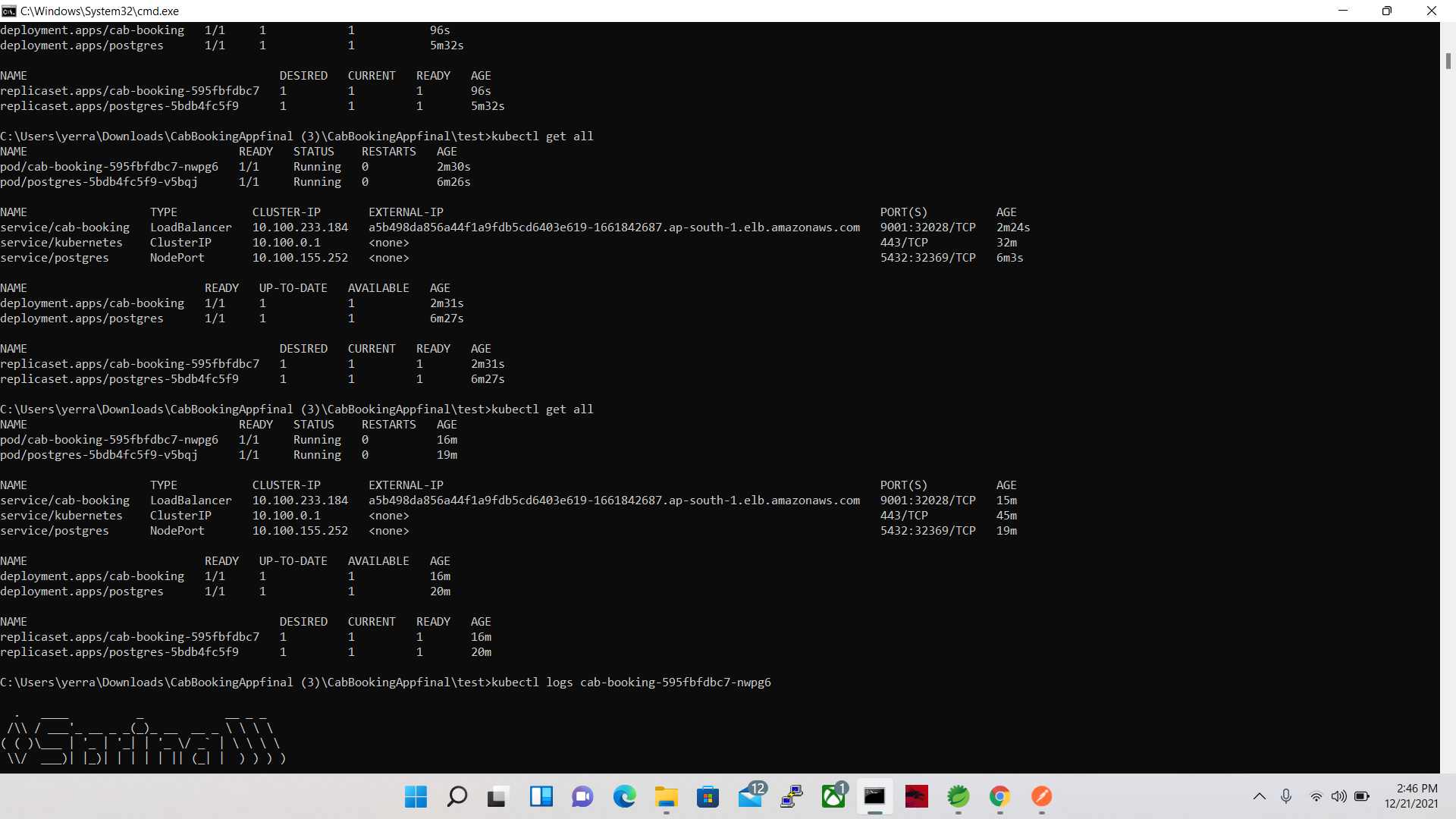
1. kubectl apply -f postgres-storage.yml
2. kubectl apply -f postgres-secrets.yml
3. kubectl apply -f postgres-deployment.yml
4. kubectl apply -f postgres-service.yml



5)kubectl get all

6)Get the Postgres Host IP Address:

* kubectl get svc postgres -o jsonpath="{.spec.clusterIP}"
* get the IP Address and put in the below command
* kubectl create configmap hostname-config –from literal=postgres\_host=10.100.233.184
* kubectl apply -f springboot-deployment.yml
* kubectl apply -f springboot-service.yml

0

**EKS Cluster Link:-**

http://a5b498da856a44f1a9fdb5cd6403e619-1661842687.ap-south-1.elb.amazonaws.com:8080/driver/getbestdrivers

