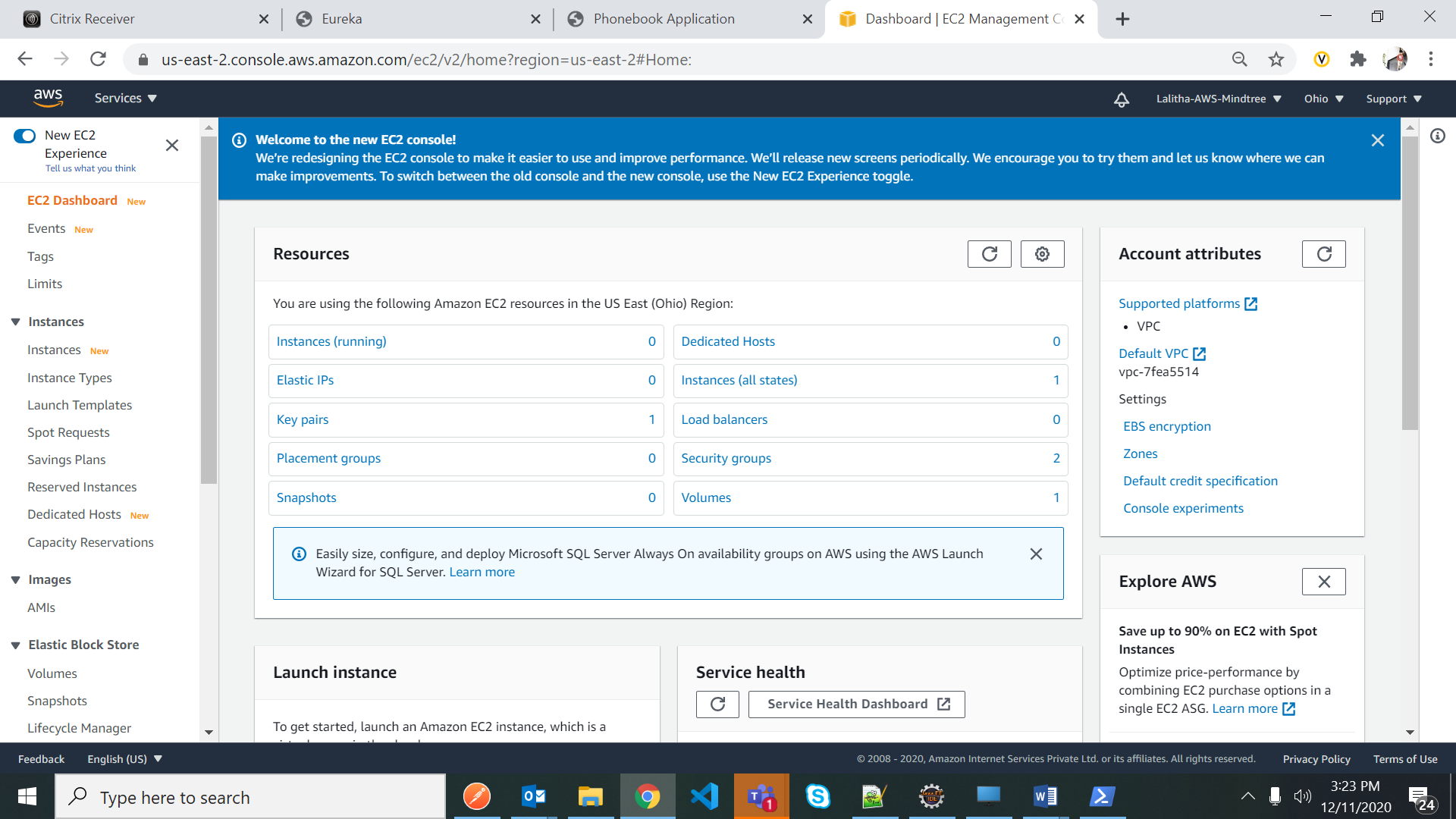
**Microservice Application Deployment in Kubernetes in AWS**

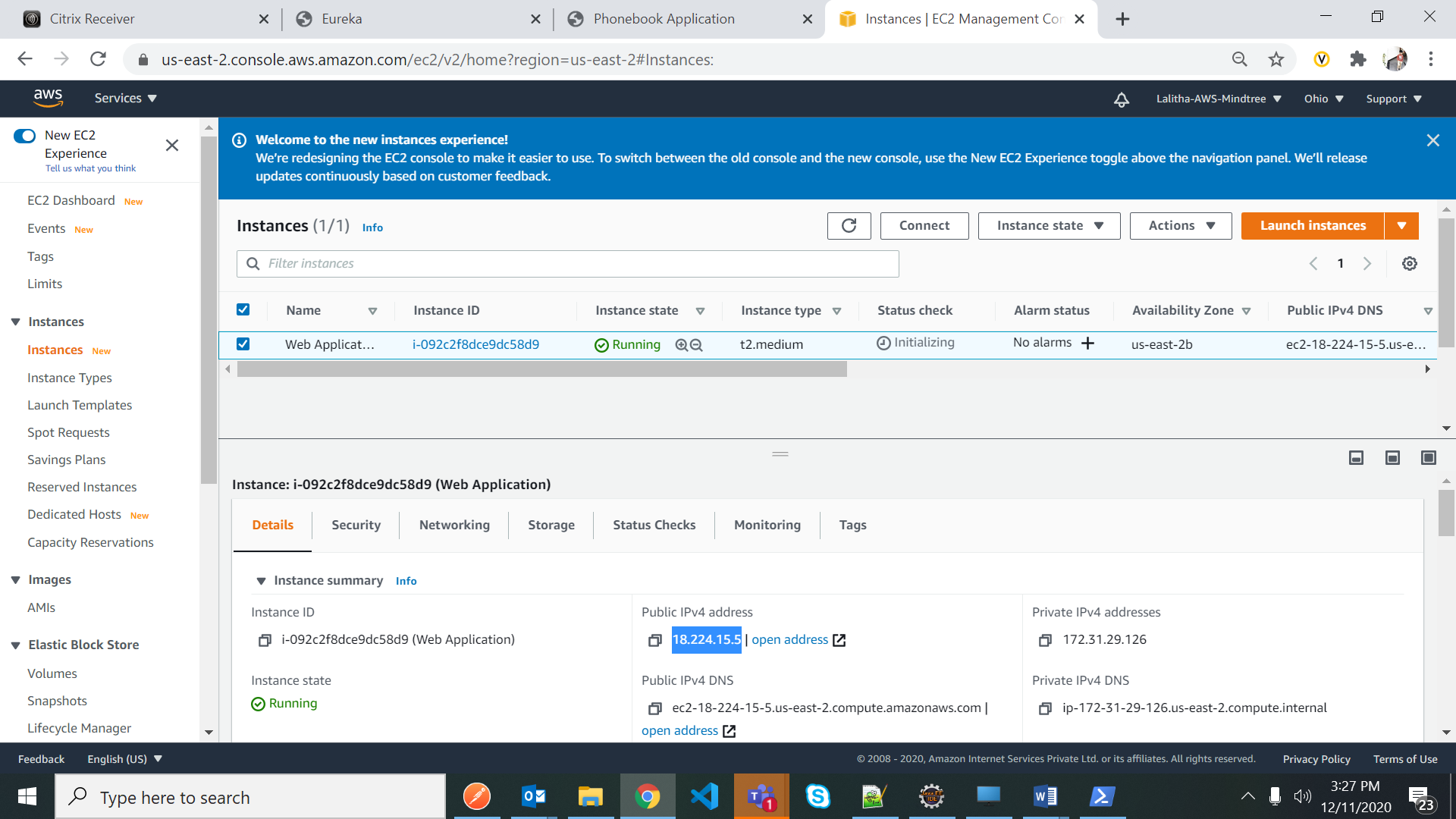
**Eureka-Service,Inventory-Service,Customer-Service,Order-Service**

Login to free AWS account



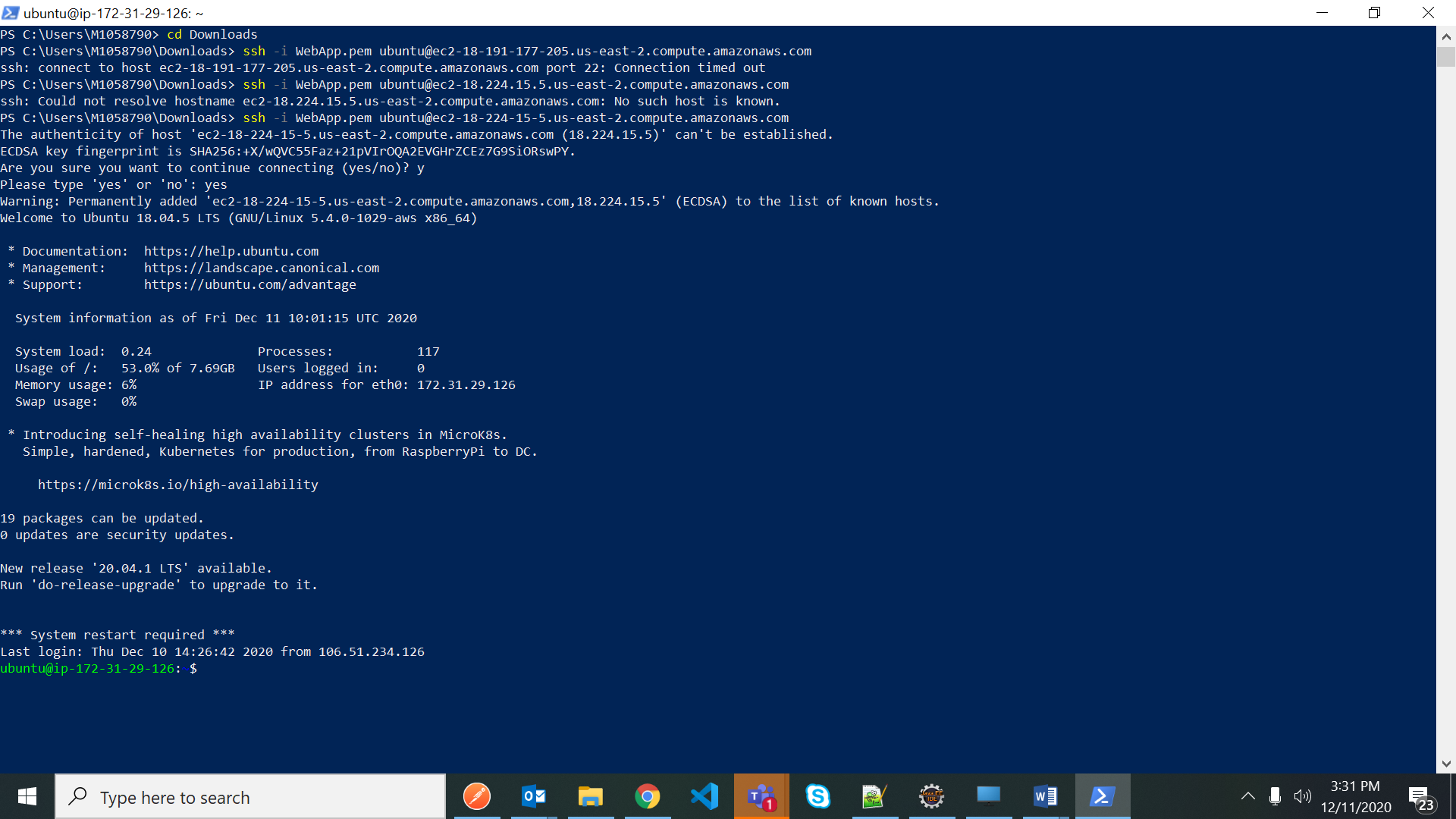
Open created Ubuntu OS EC2 instance





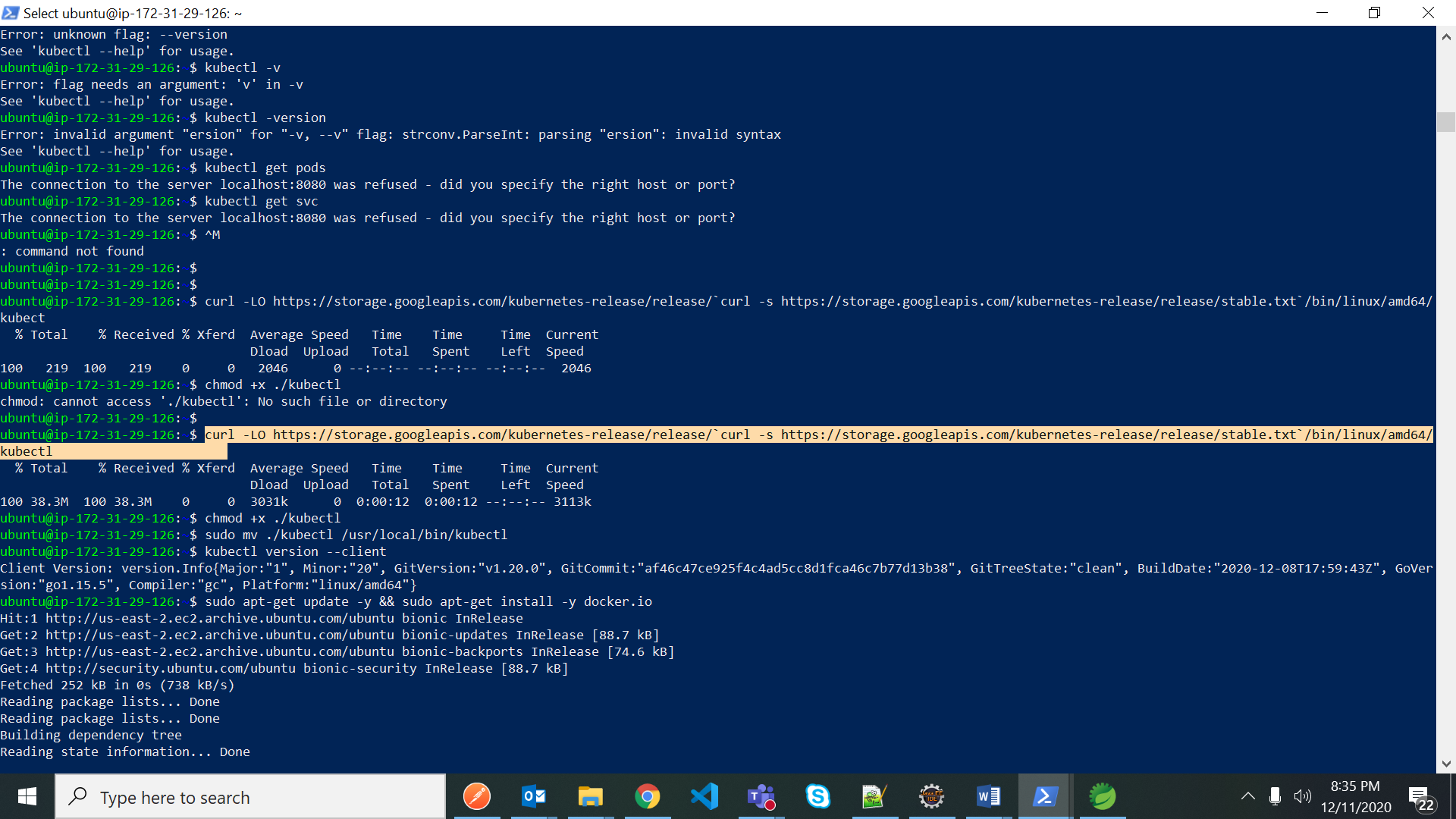
Copy Ip address and paste it in poweshell to connect to Ubuntu OS – goto downloads where this WebApp.pem file is downloaded from there run the command

ssh -i WebApp.pem [ubuntu@ec2-18-224-15-5.us-east-2.compute.amazonaws.com](mailto:ubuntu@ec2-18-224-15-5.us-east-2.compute.amazonaws.com)



Install Kubernetes

curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl

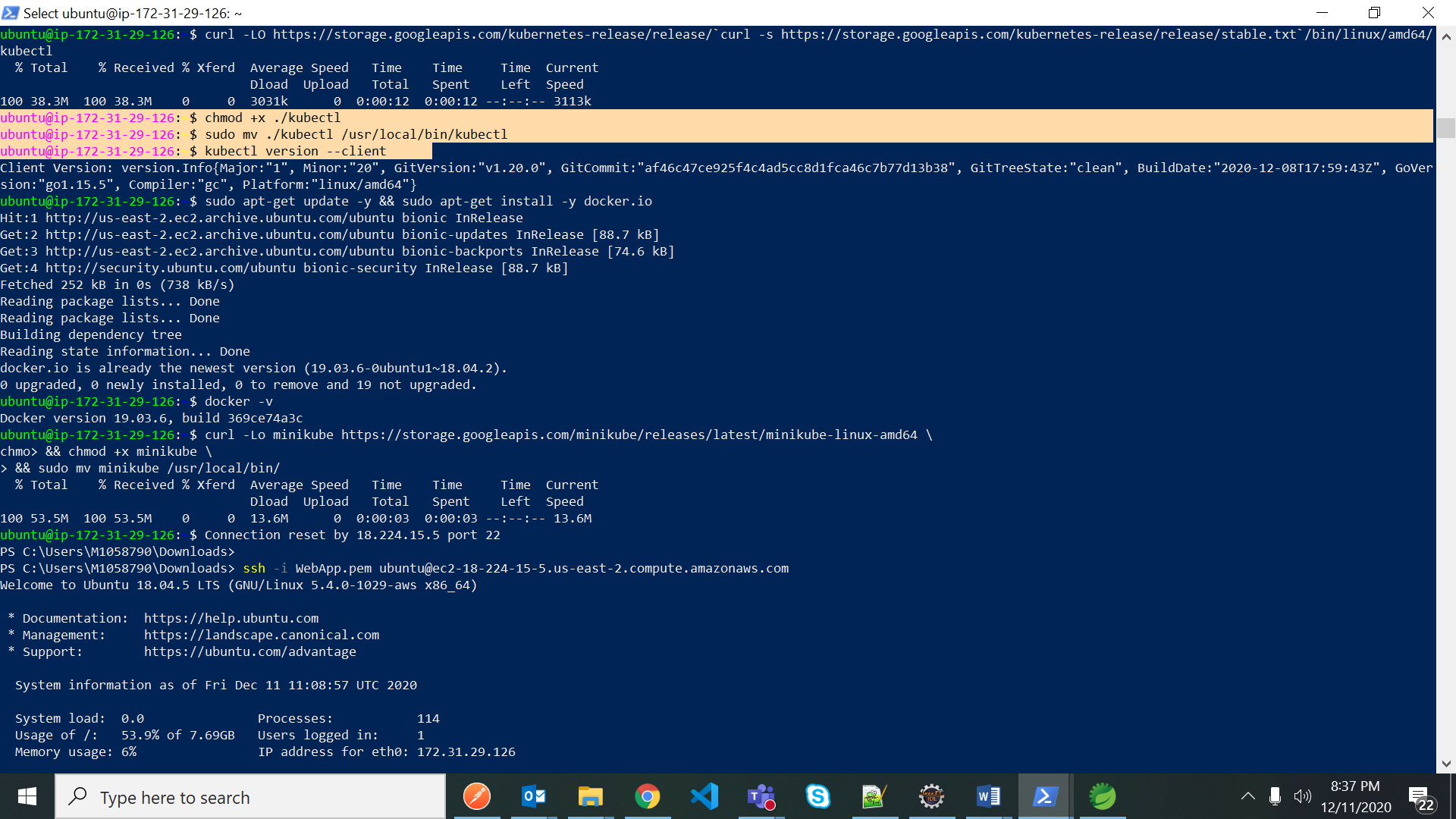


Move kubectl to bin folder – use below commands to move to bin and check kubernetes version

chmod +x ./kubectl

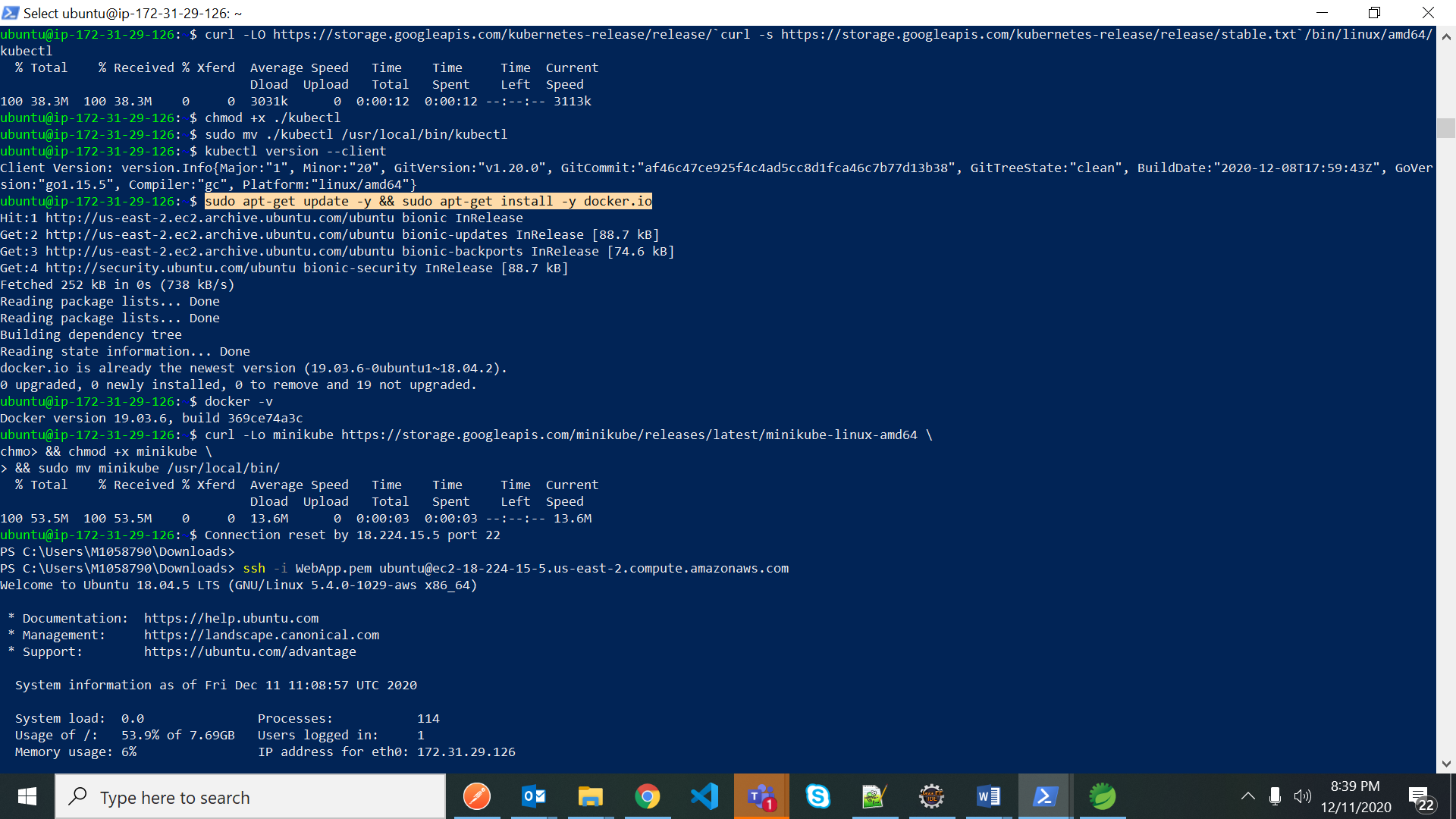
sudo mv ./kubectl /usr/local/bin/kubectl

kubectl version --client



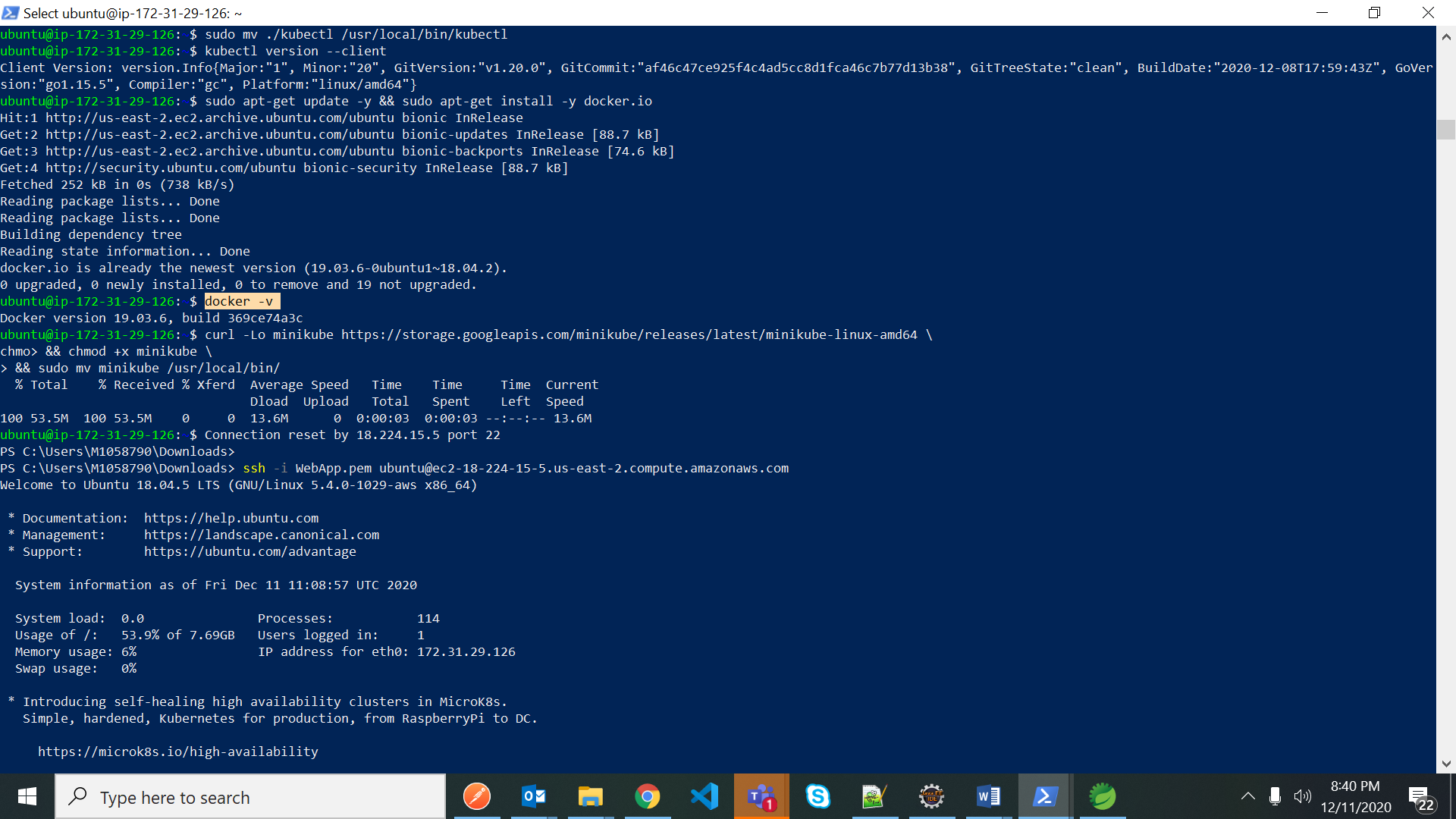
Install Docker

sudo apt-get update -y && sudo apt-get install -y docker.io



Check Docker version

docker –v

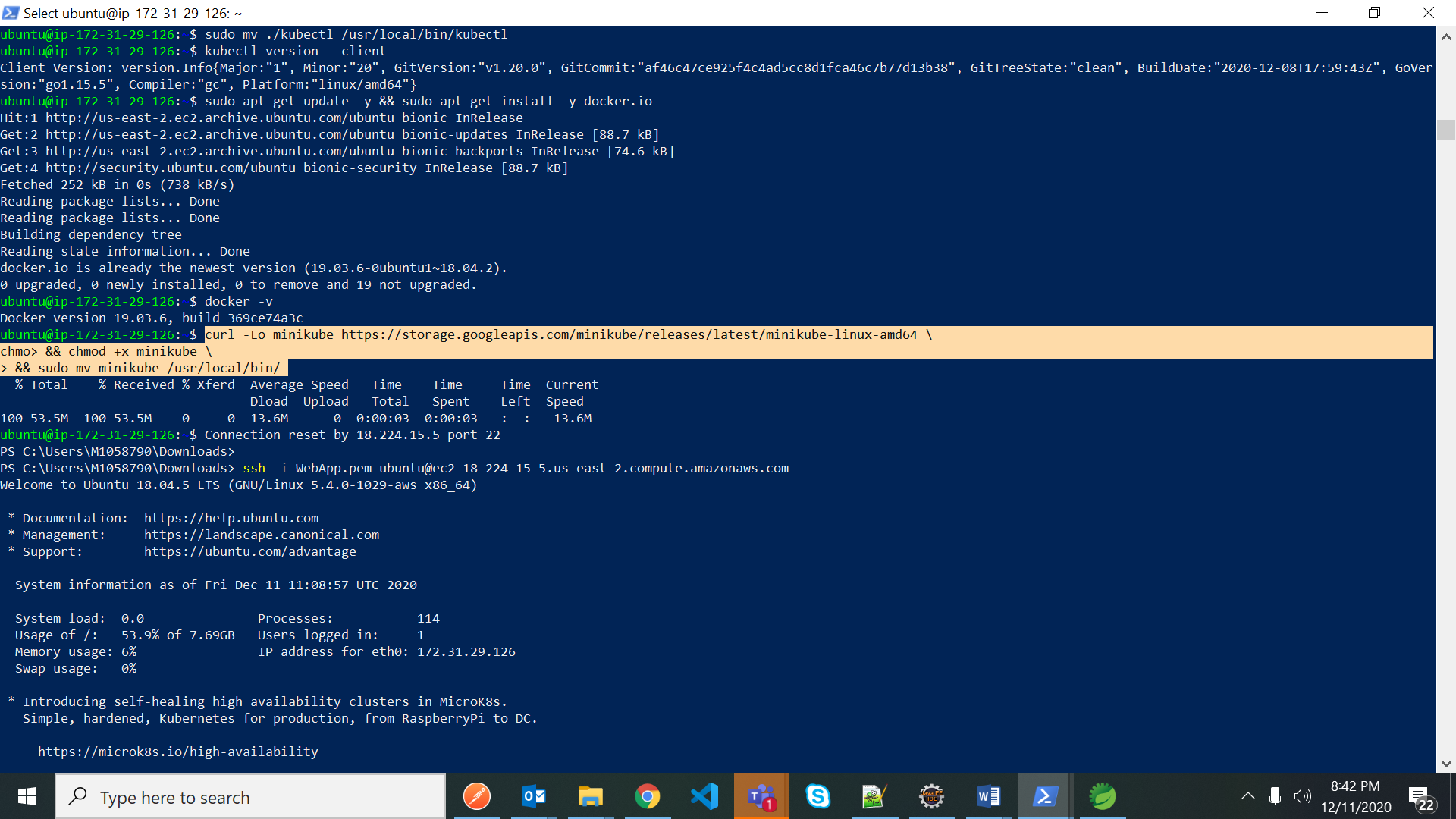


Now install minikube

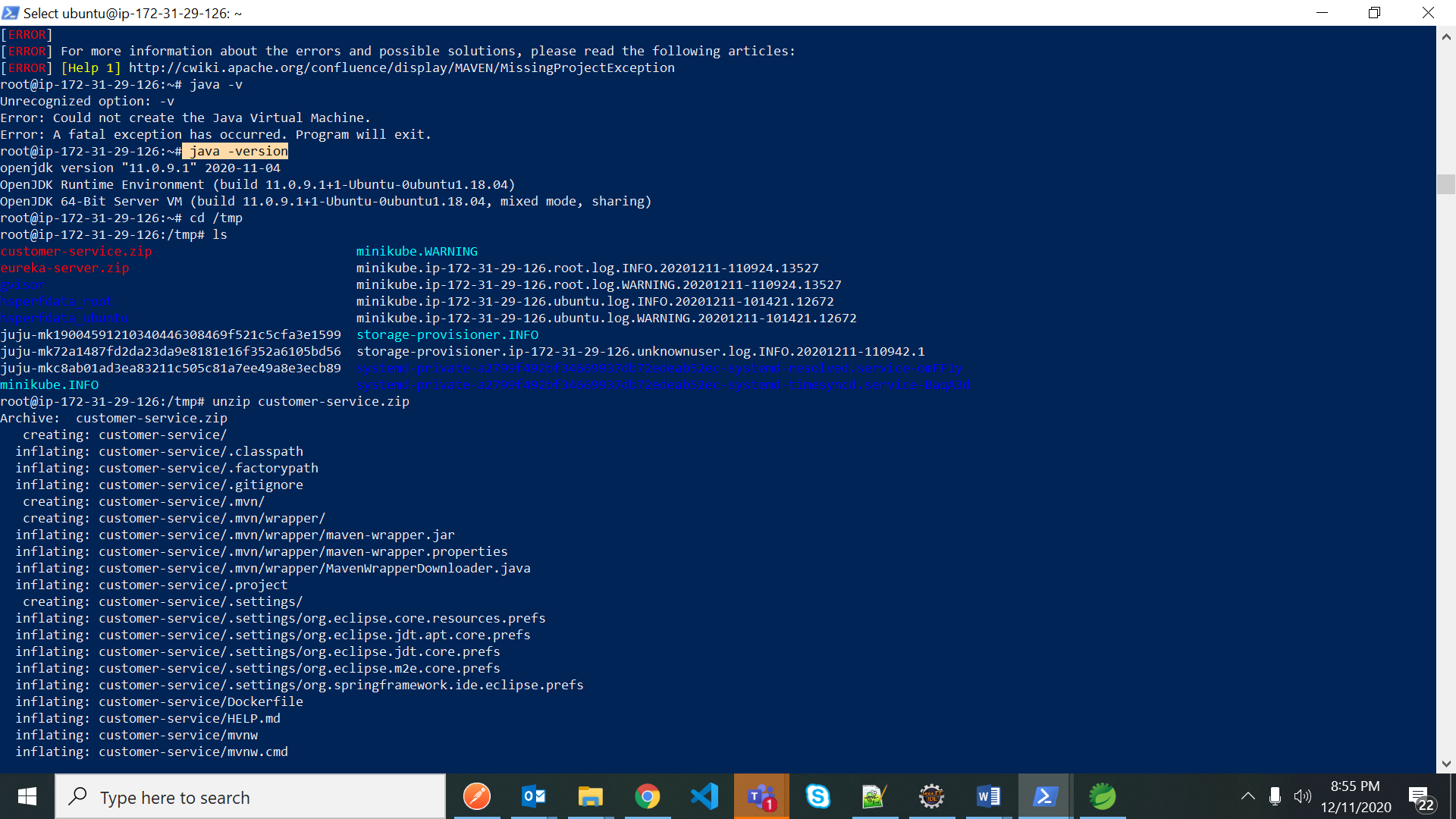
curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 \

chmo> && chmod +x minikube \

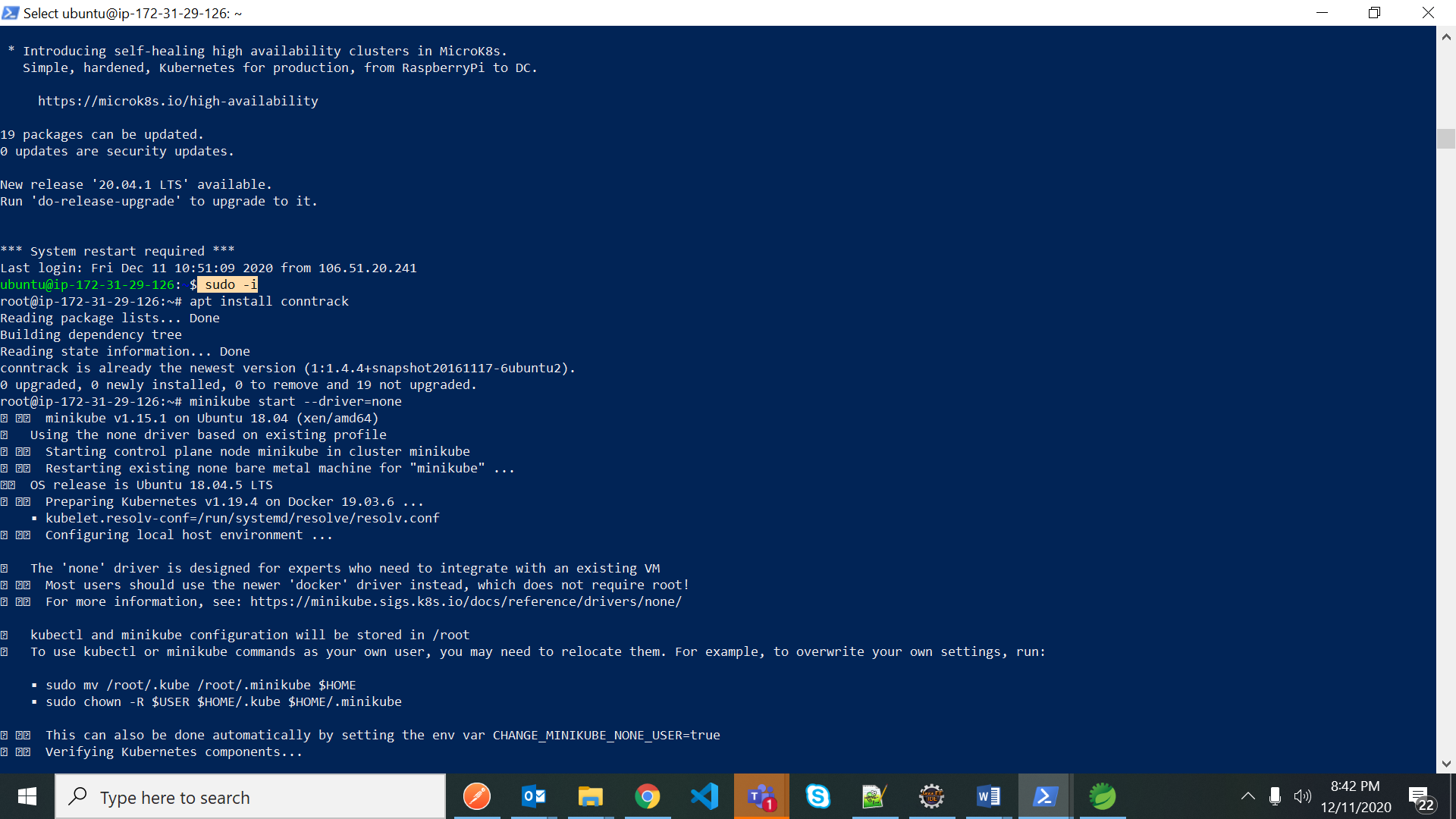
> && sudo mv minikube /usr/local/bin/



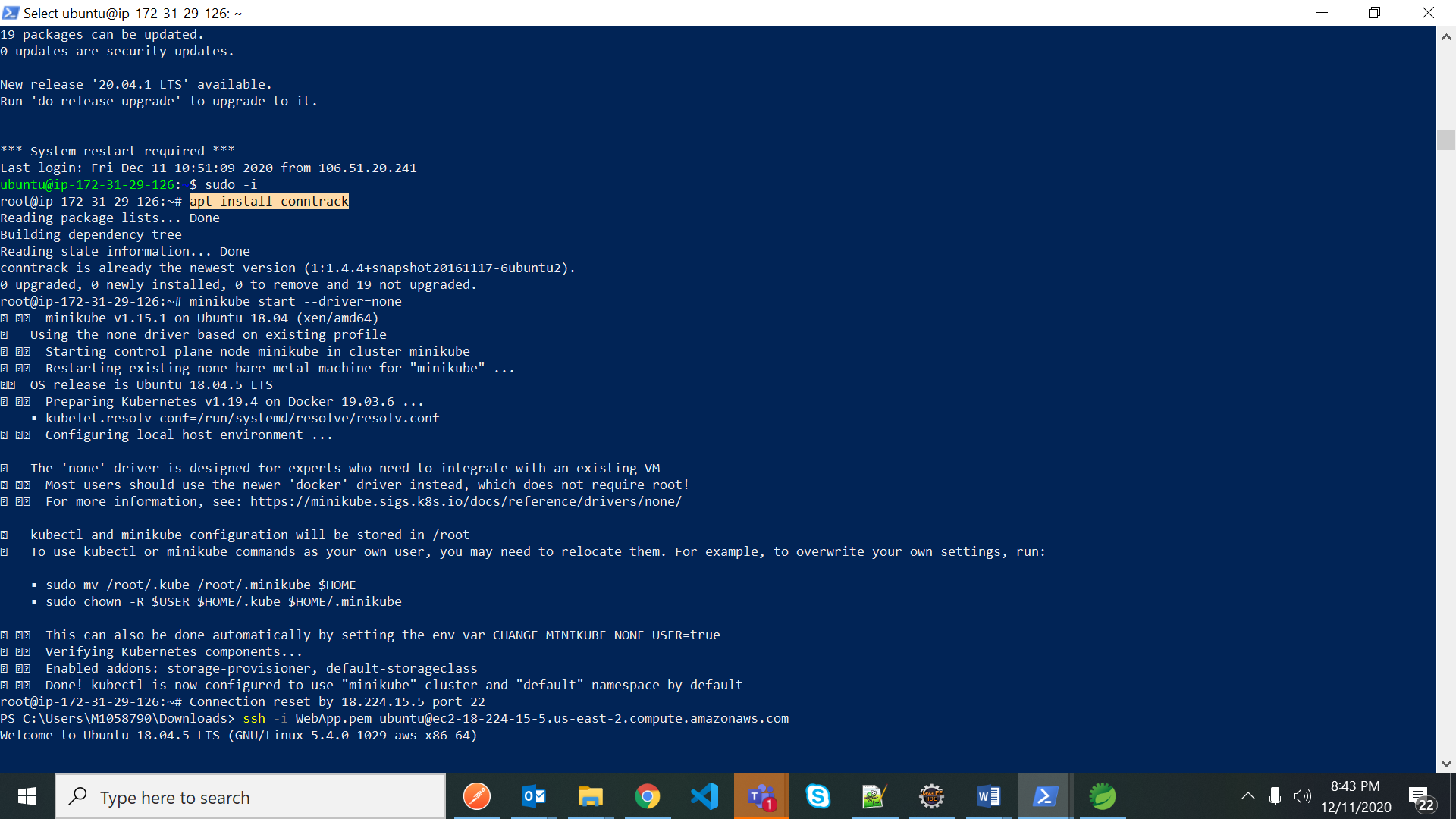
java –version



sudo –i



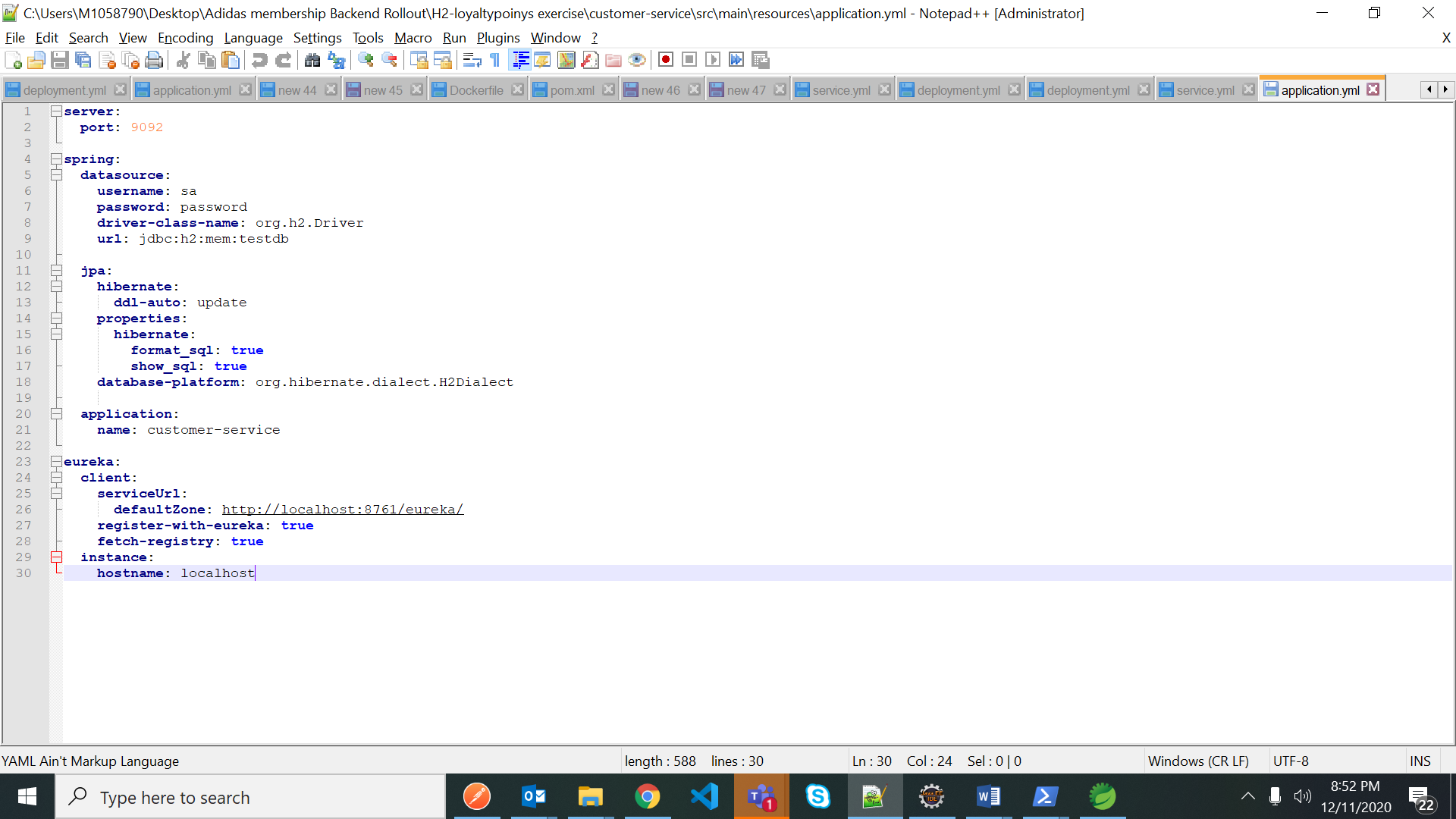
apt install conntrack

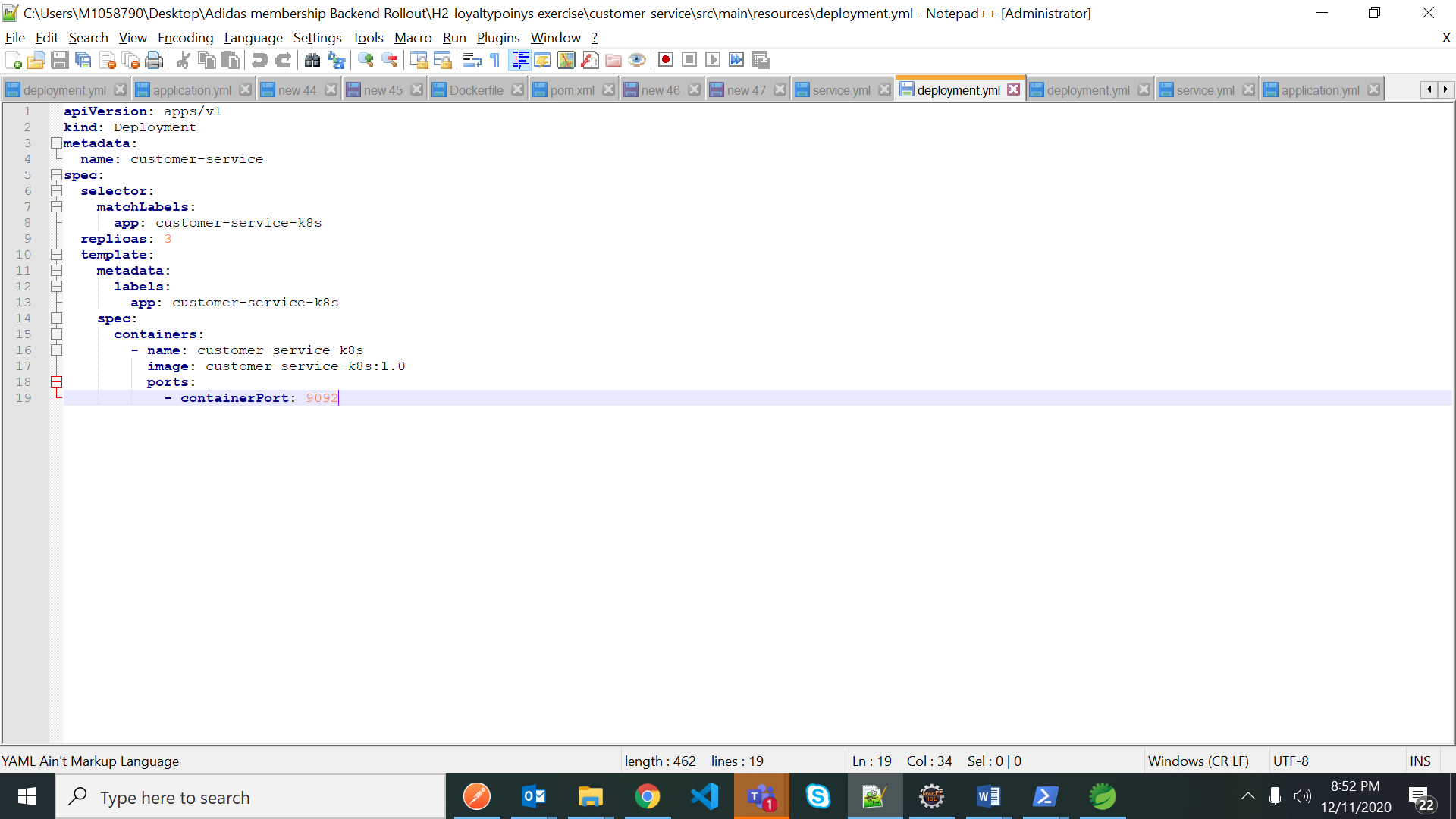


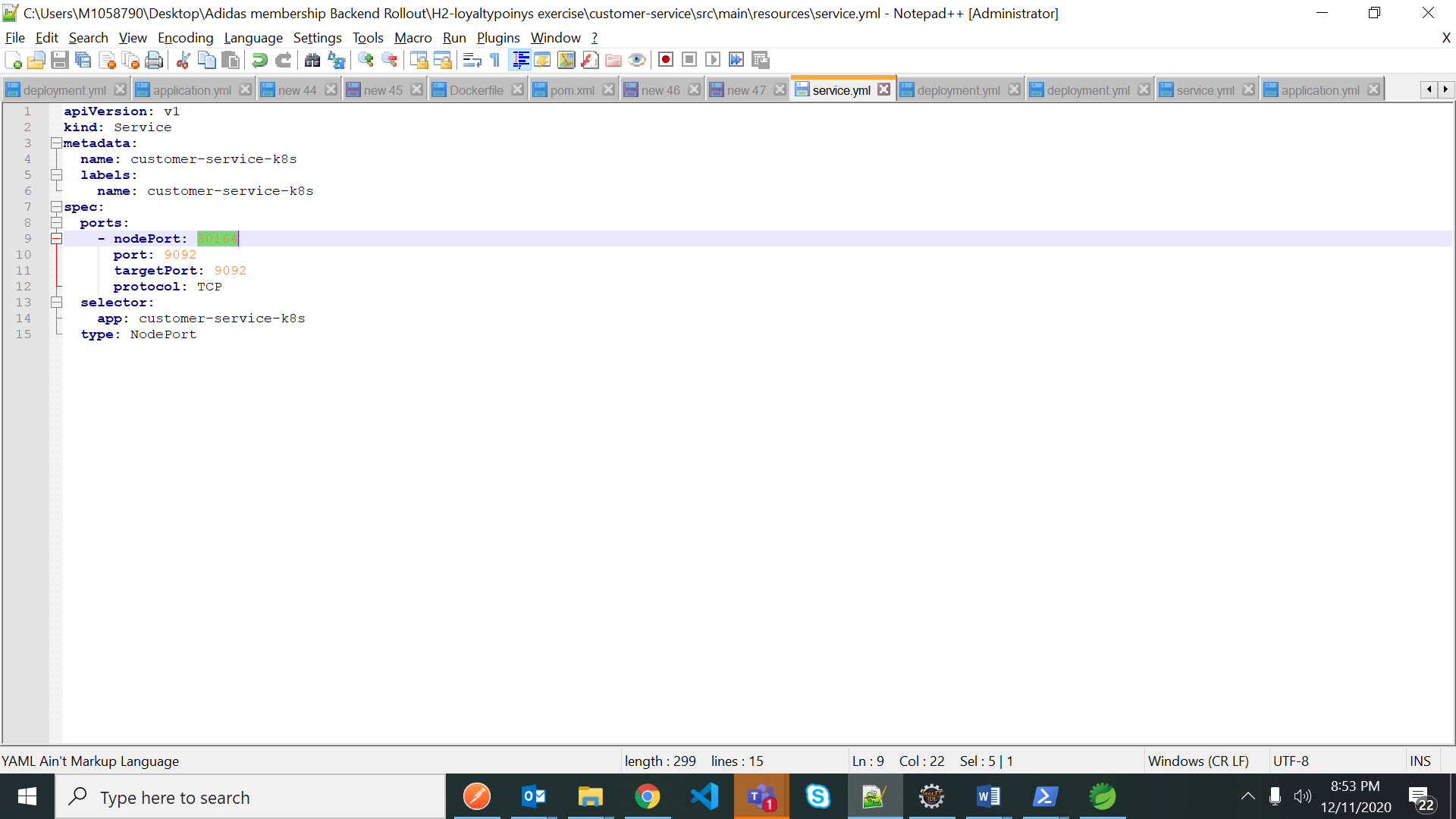
Before adding zip folder to tmp and converting it to docker image we have to change the localhost configurations to AWS cloud configuration like changing IP etc

- change localhost to : 18.224.15.5 which is EC2 Ubuntu Instance IP Address

Change the application.yml,deployment.yml and service.yml as per the requirement

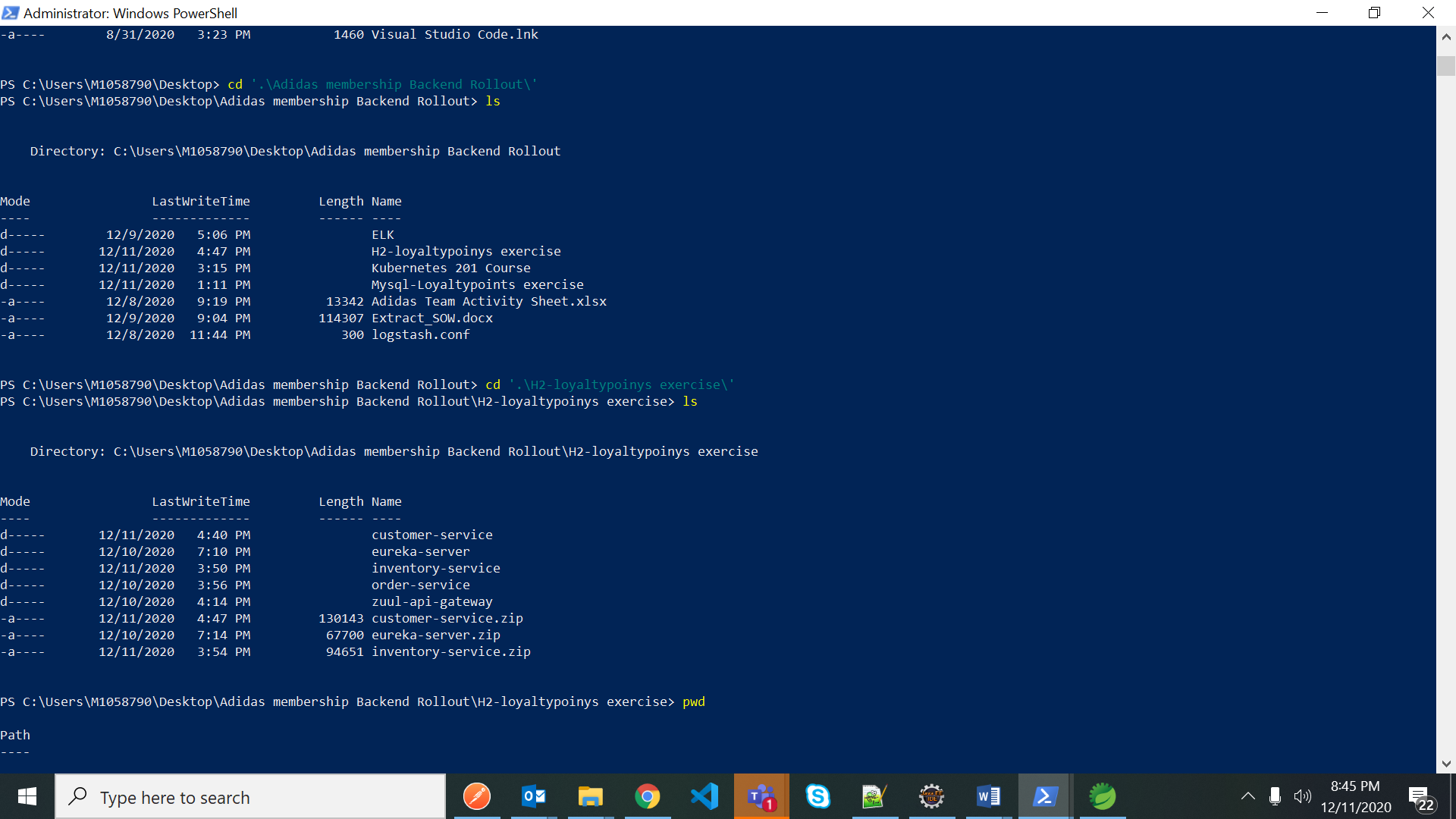






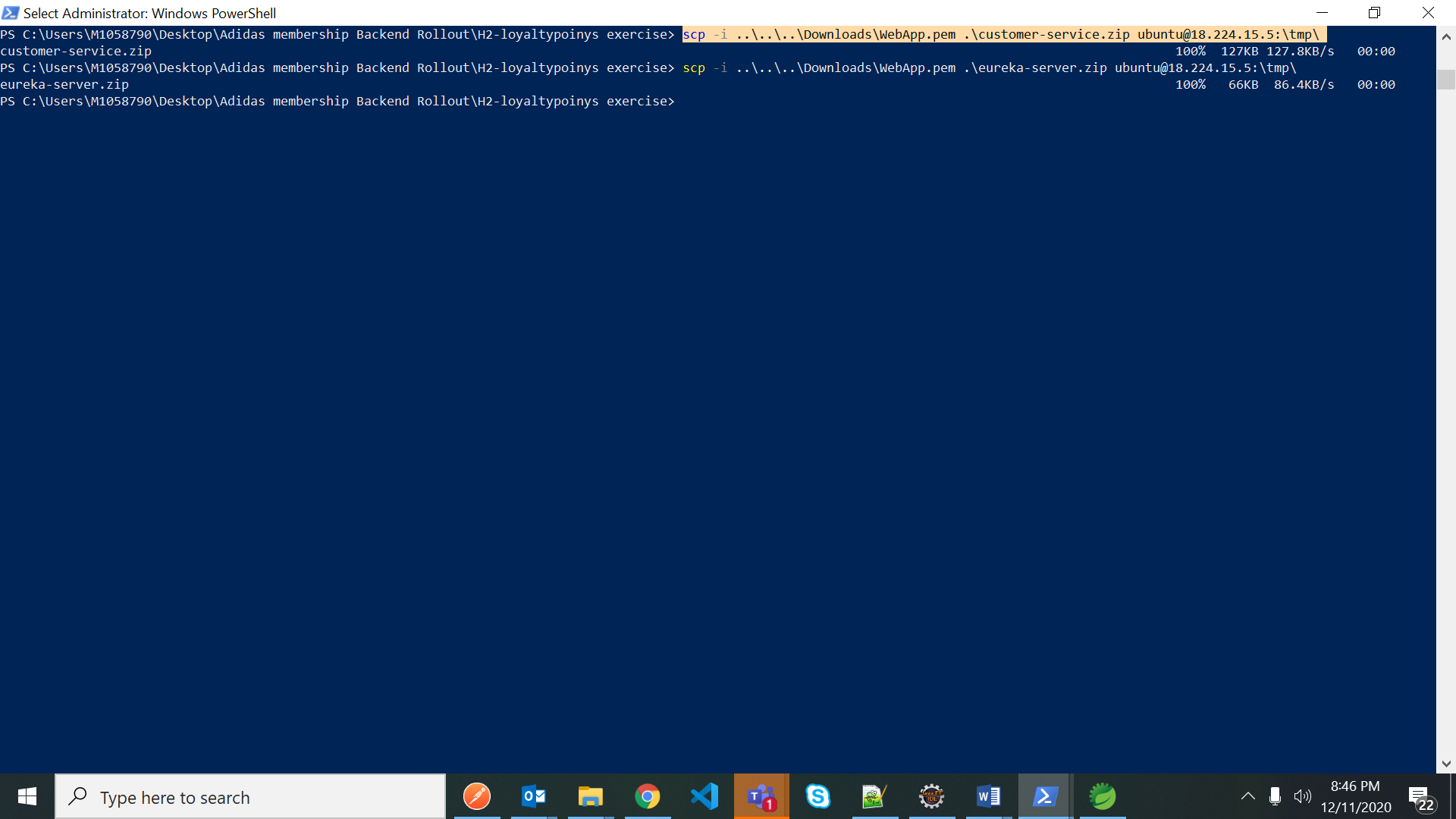
Save all the changes and convert it to ZIP file

Now open another powershell and goto the path where microservice zip file is available



to move the customer service zip folder to tmp folder use below cmd

scp -i ..\..\..\Downloads\WebApp.pem .\customer-service.zip [ubuntu@18.224.15.5:\tmp\](mailto:ubuntu@18.224.15.5:\tmp\)



Goto ubantu powershell

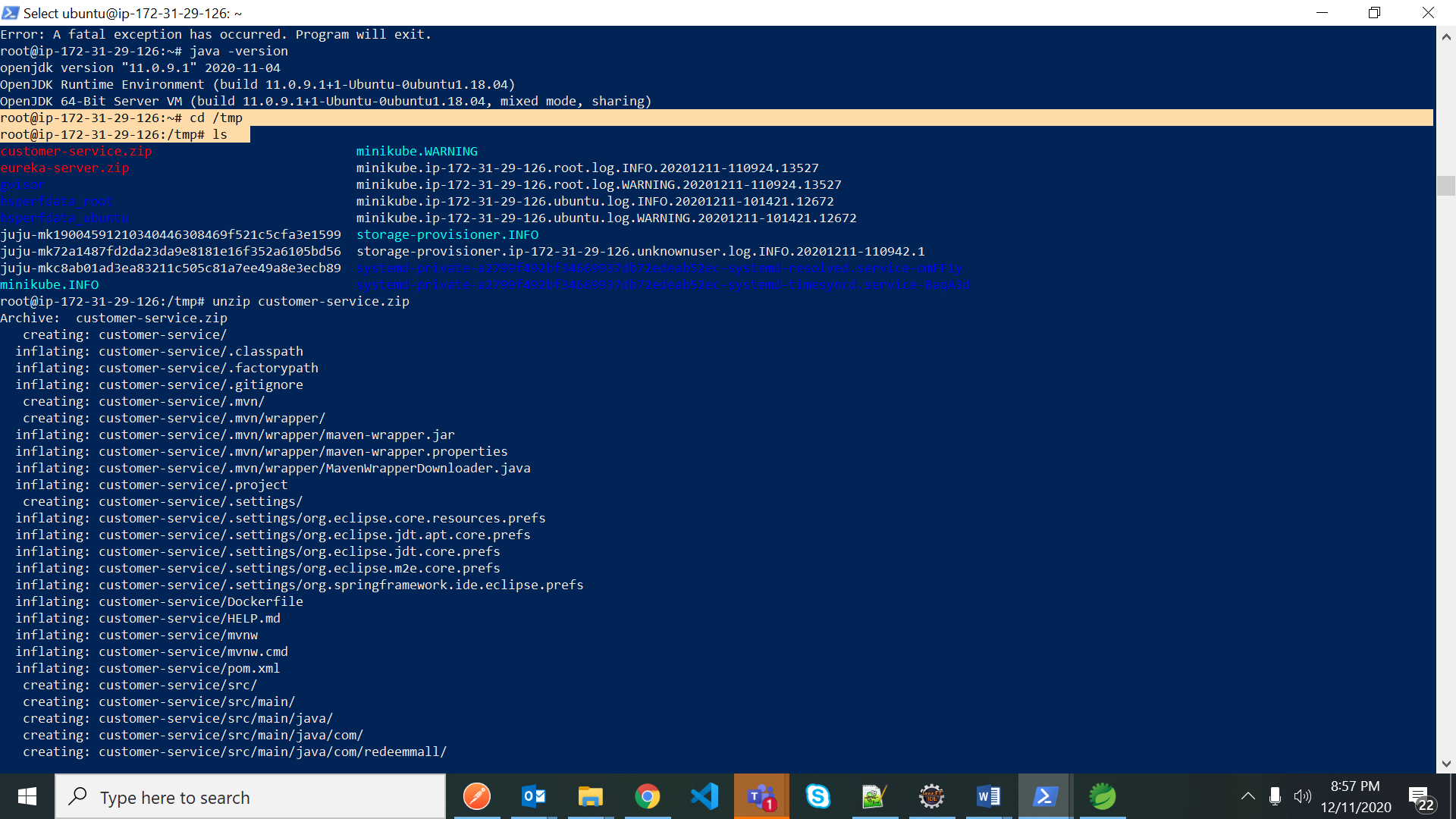
Goto tmp directory where our customer-service.zip file is available

cd /tmp

Note : if any directory to remove use below command

sudo rm -r /tmp/inventory-service

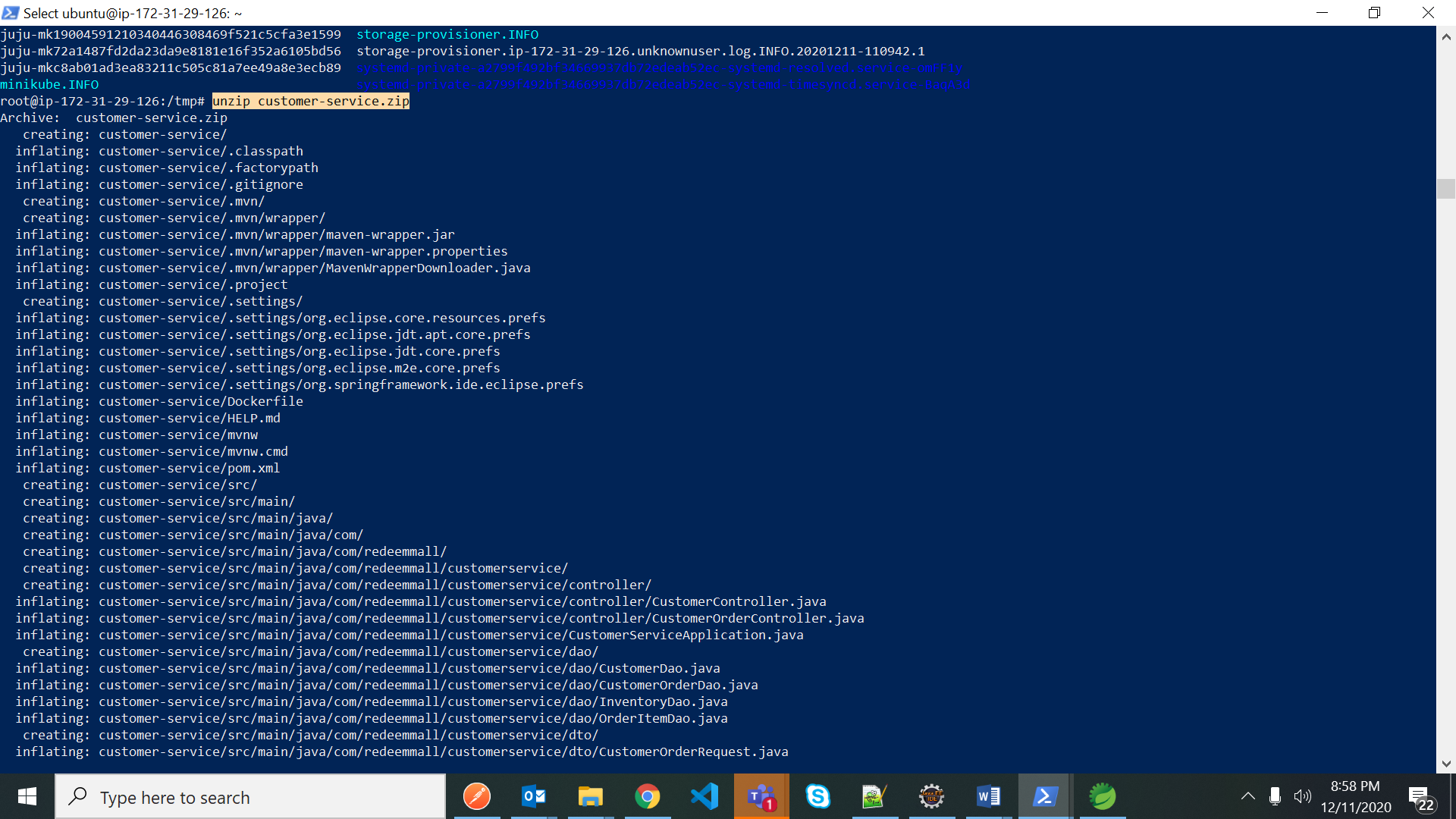
ls



Now customer-service and eureka-service are available in tmp folder

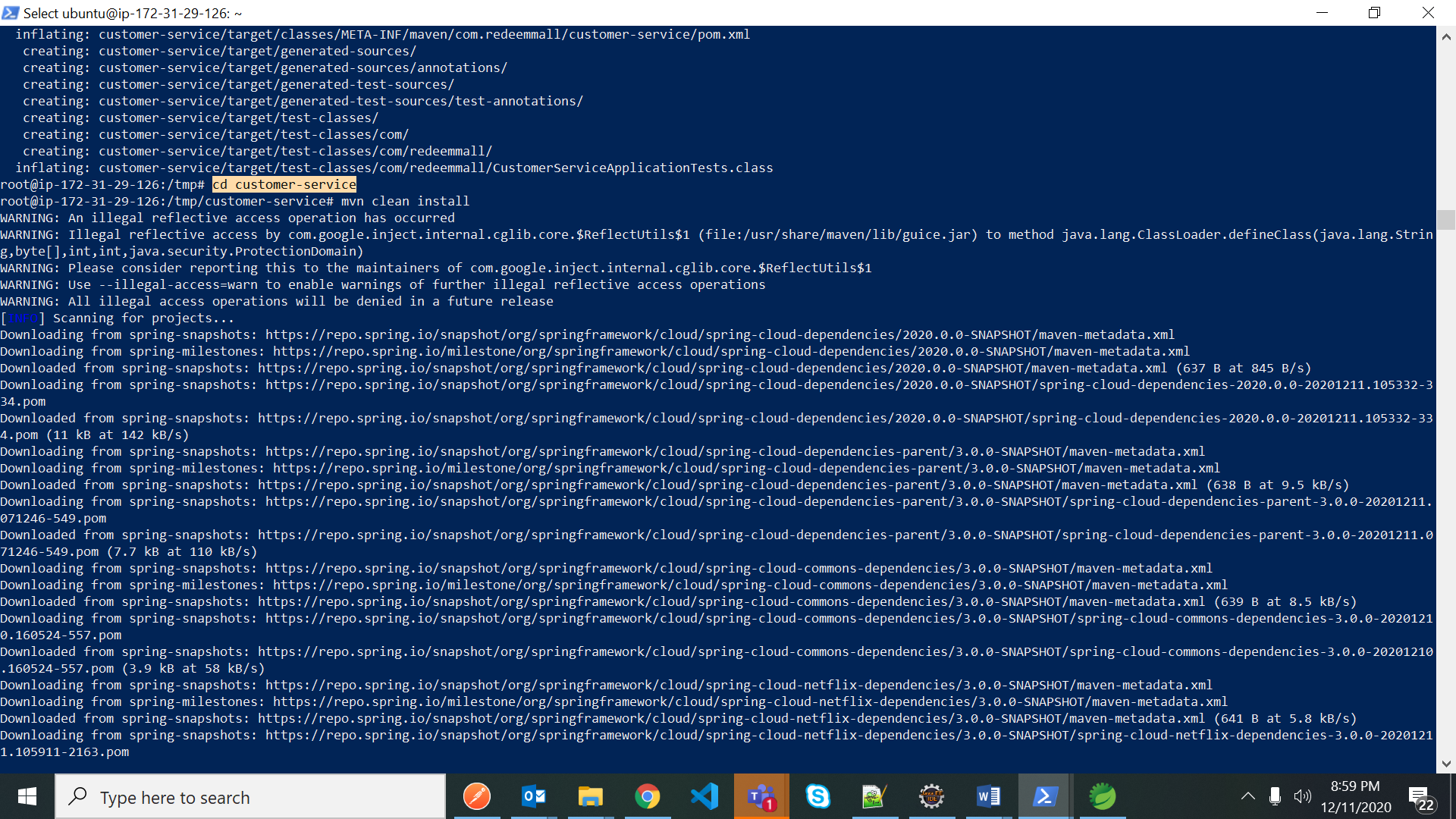
Now unzip the customer-service

unzip customer-service.zip



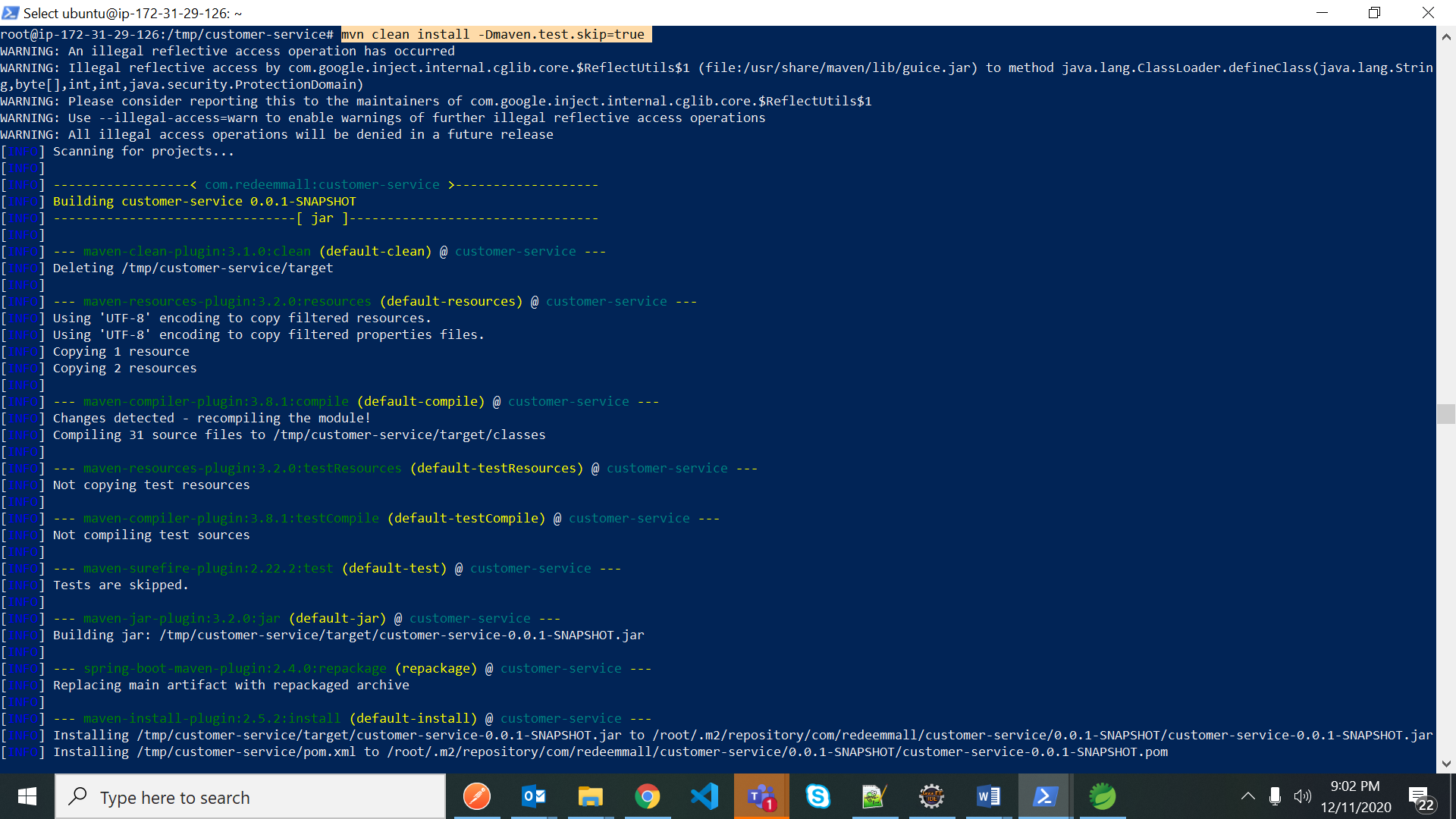
After file is unzipped – goto that customer-service file

> cd customer-service



Do maven clean install – use below cmd to skip the test failures

> mvn clean install -Dmaven.test.skip=true



Follow the above process to unzip the eureka-server.zip file

>unzip eureka-server.zip

> cd eureka-server

>mvn clean install -Dmaven.test.skip=true

After Maven build is successful create/build the docker image for microservice

Now change the directory to customer-service to build the docker image for that microservice

>cd ..

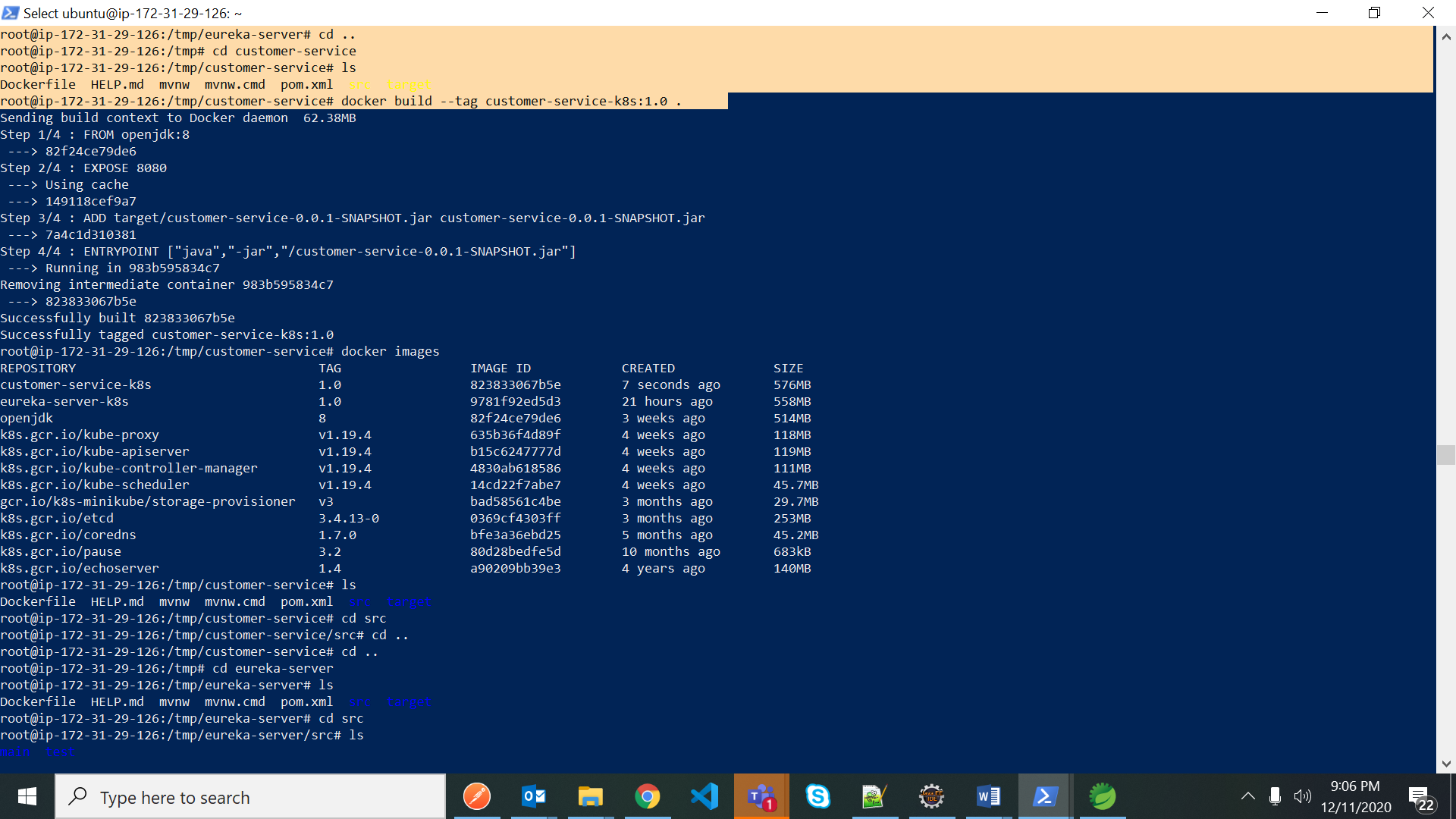
root@ip-172-31-29-126:/tmp/eureka-server# cd ..

root@ip-172-31-29-126:/tmp# cd customer-service

root@ip-172-31-29-126:/tmp/customer-service# ls

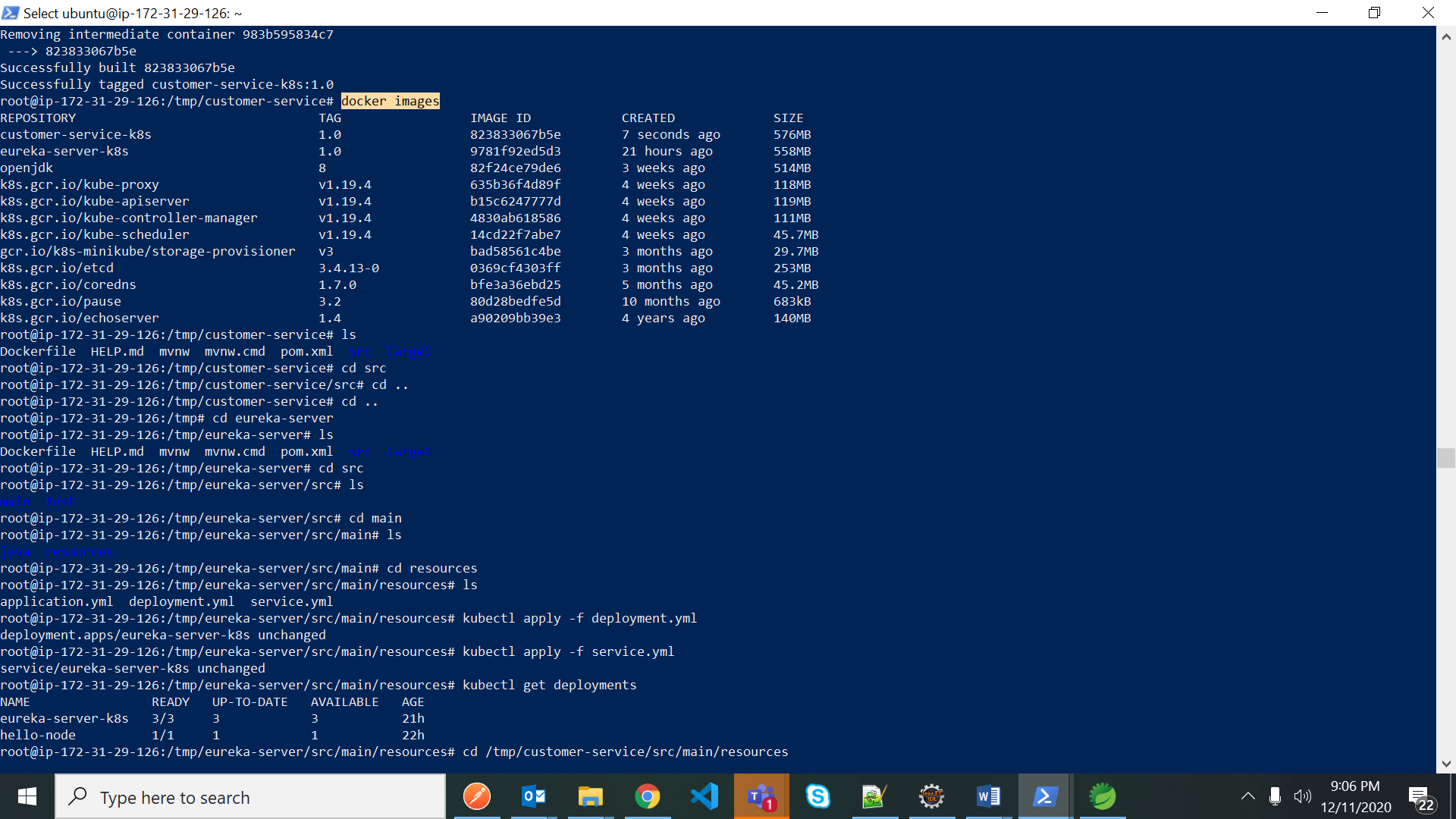
Dockerfile HELP.md mvnw mvnw.cmd pom.xml **src** target

>docker build --tag customer-service-k8s:1.0 .



After docker build image is successful go for below cmd to check the images

>docker images



>ls

Dockerfile HELP.md mvnw mvnw.cmd pom.xml src target

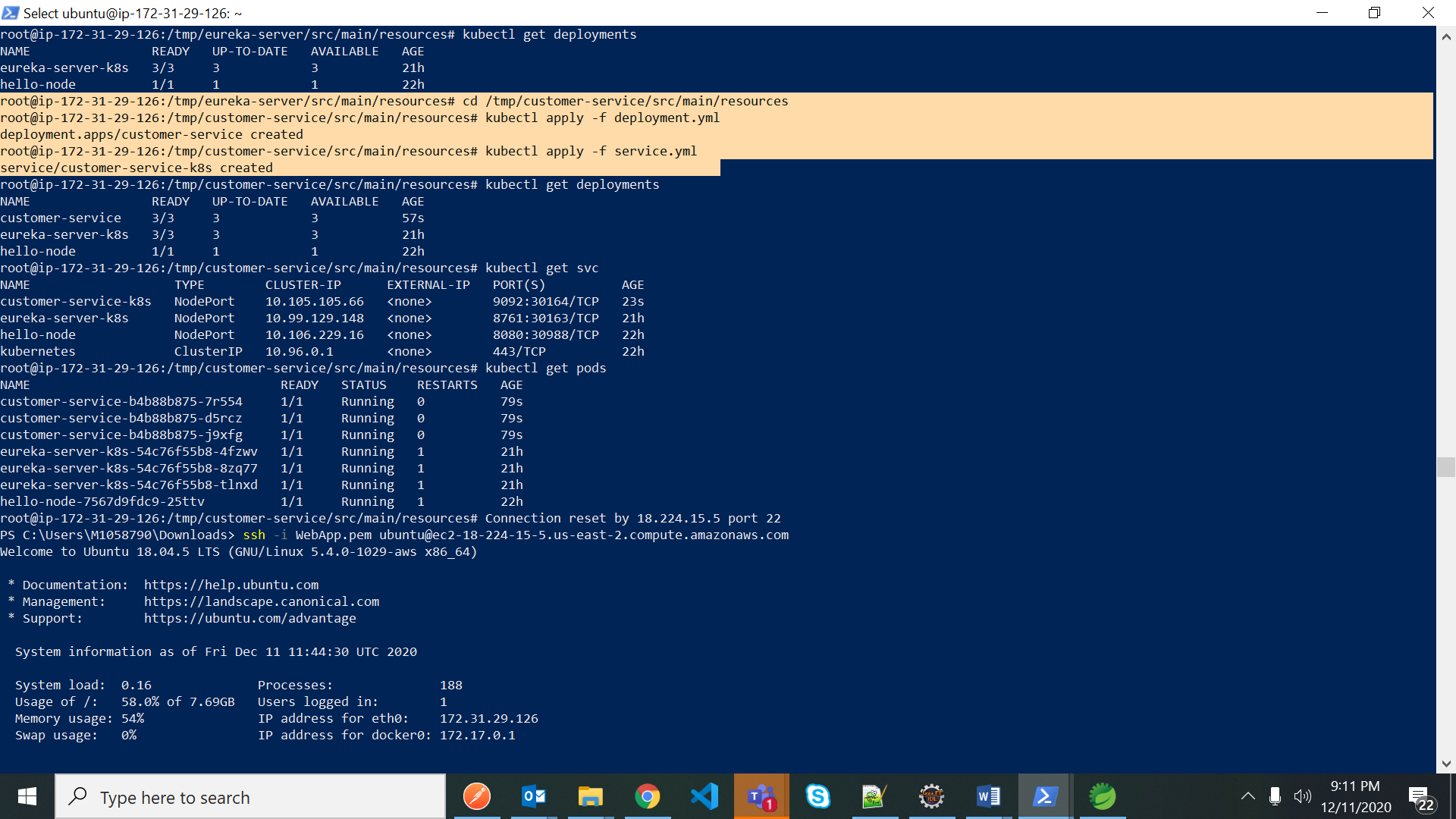
> cd src

>cd src/main/resources

Deploy all 2 yml files deployment.yml,service.yml from resourcse folder

-kubectl apply -f deployment.yml

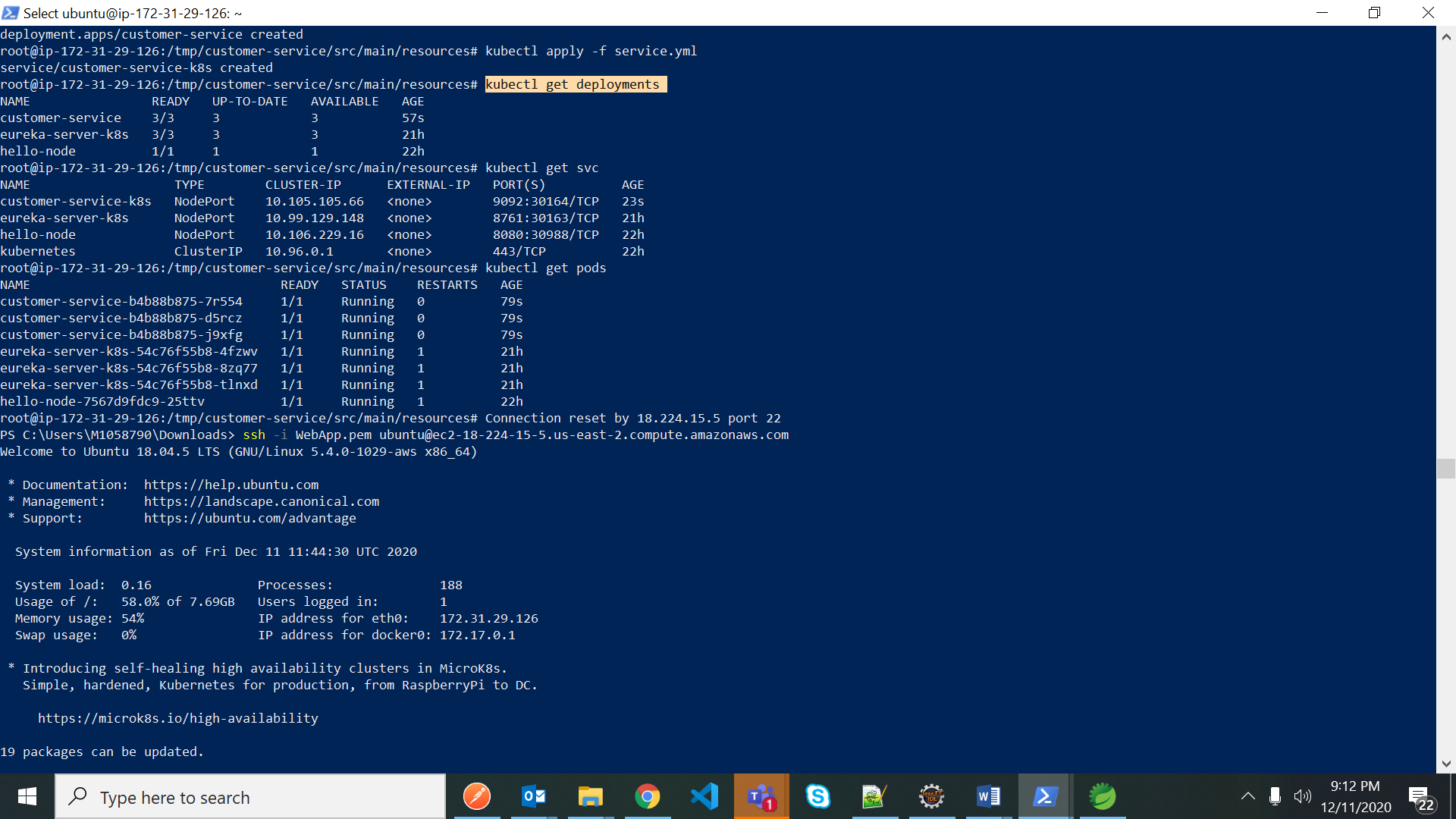
-kubectl apply -f service.yml

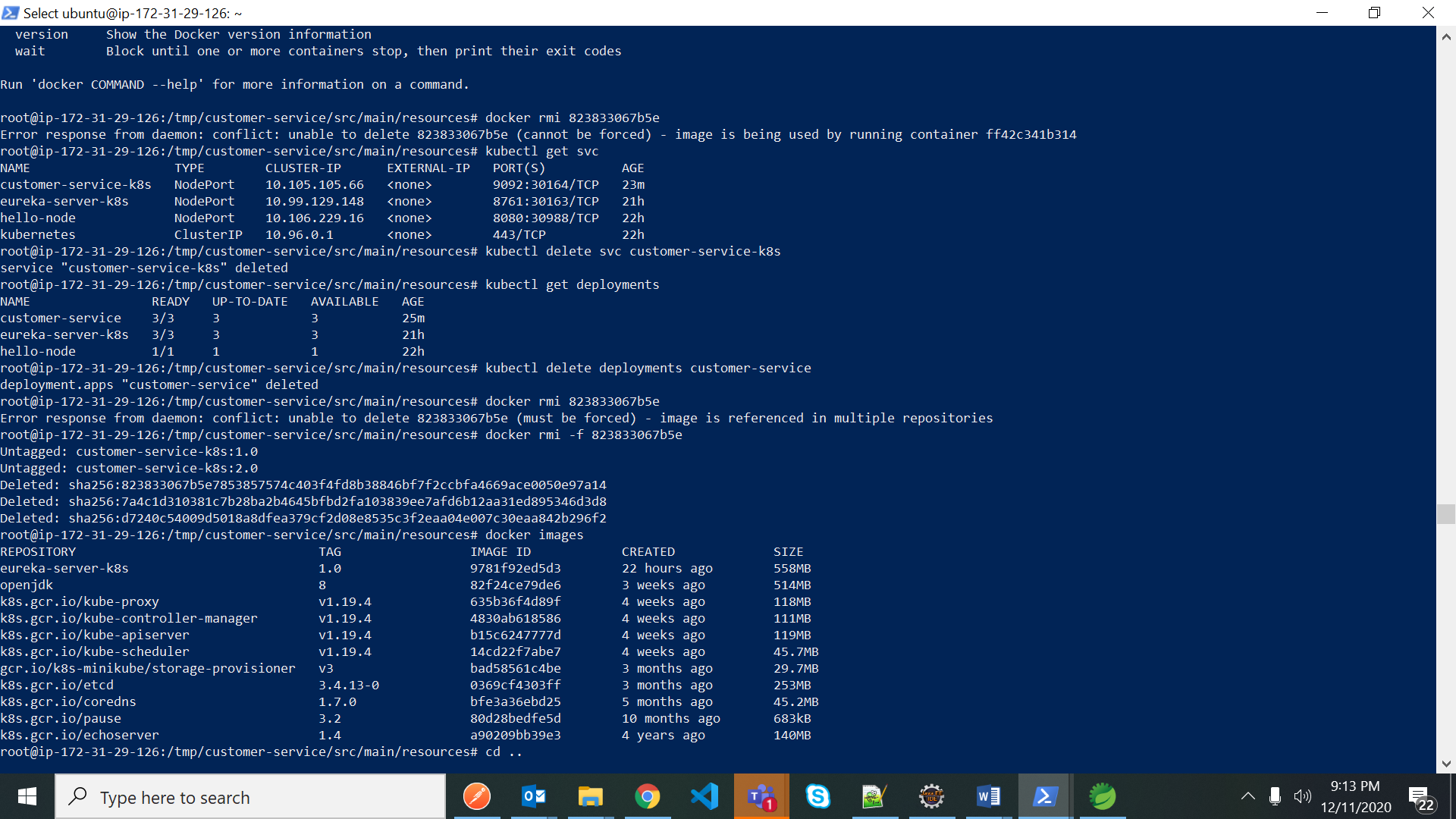


kubectl get deployments

kubectl get svc

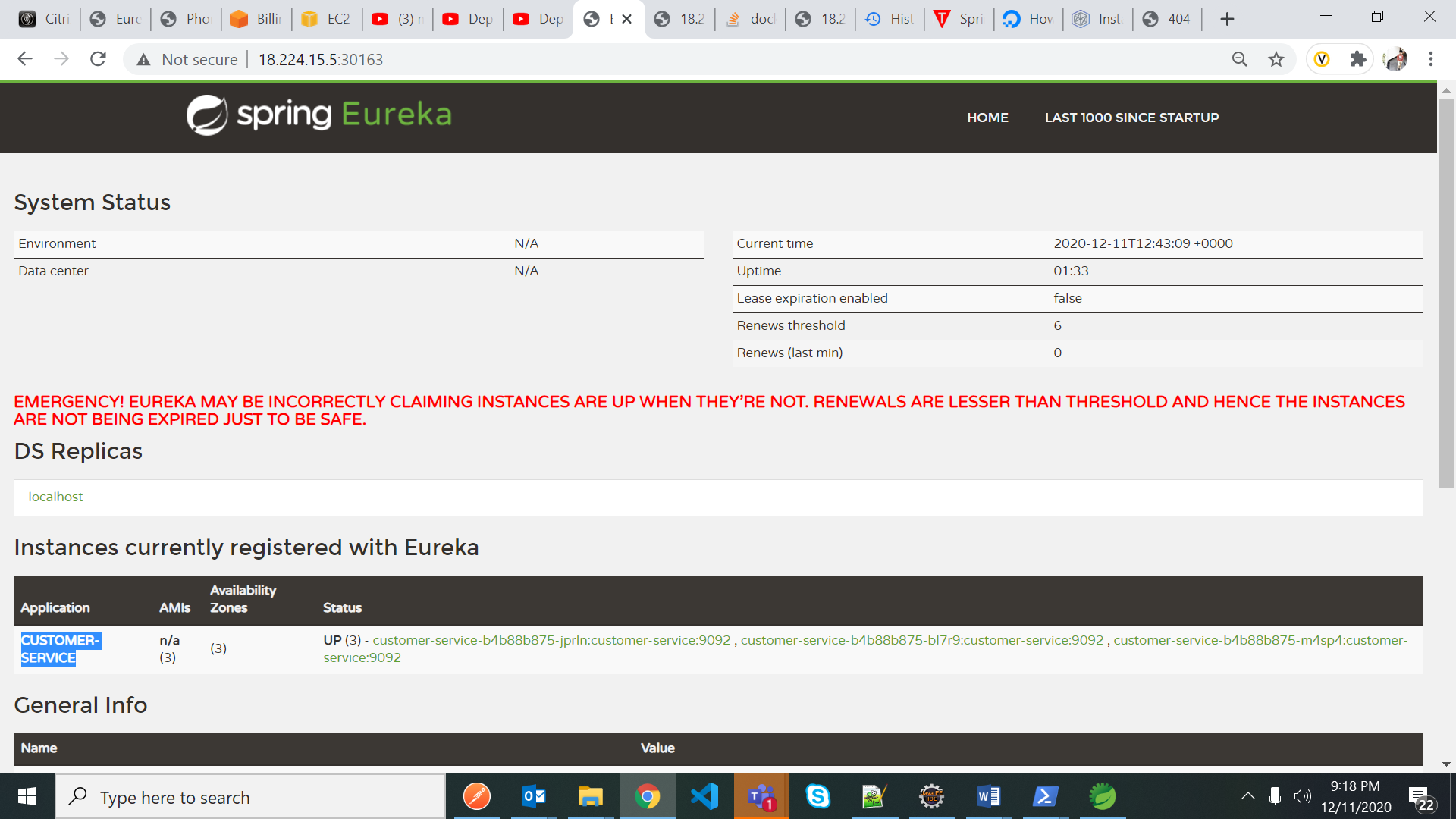
kubectl get pods





After deployment is done test the applications on IP address

Eureka Server : <http://18.224.15.5:30163/>

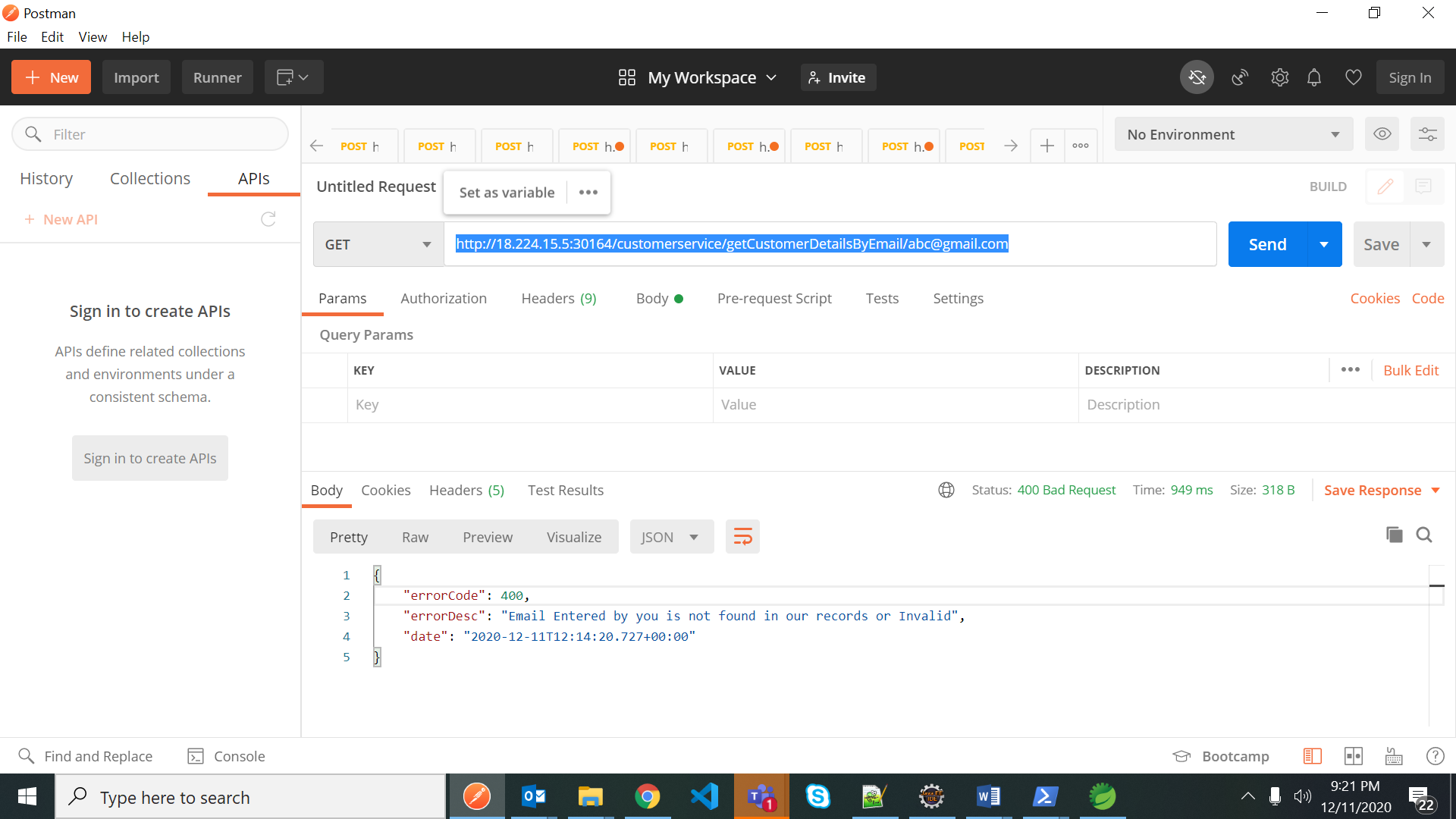


Customer-service is registered with eureka

Customer service: testing one get request endpoint like below:

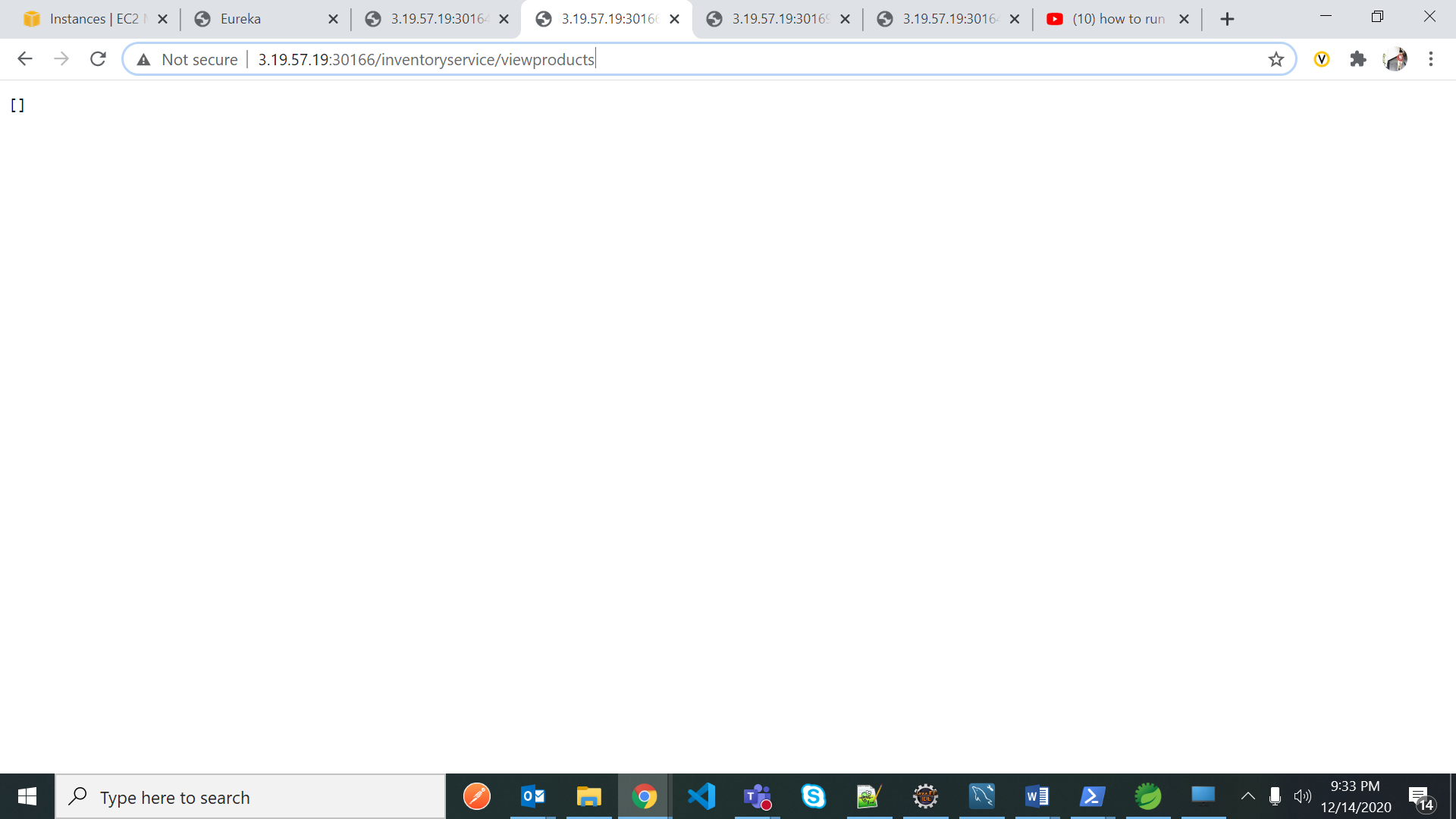
<http://18.224.15.5:30164/customerservice/getCustomerDetailsByEmail/abc@gmail.com>



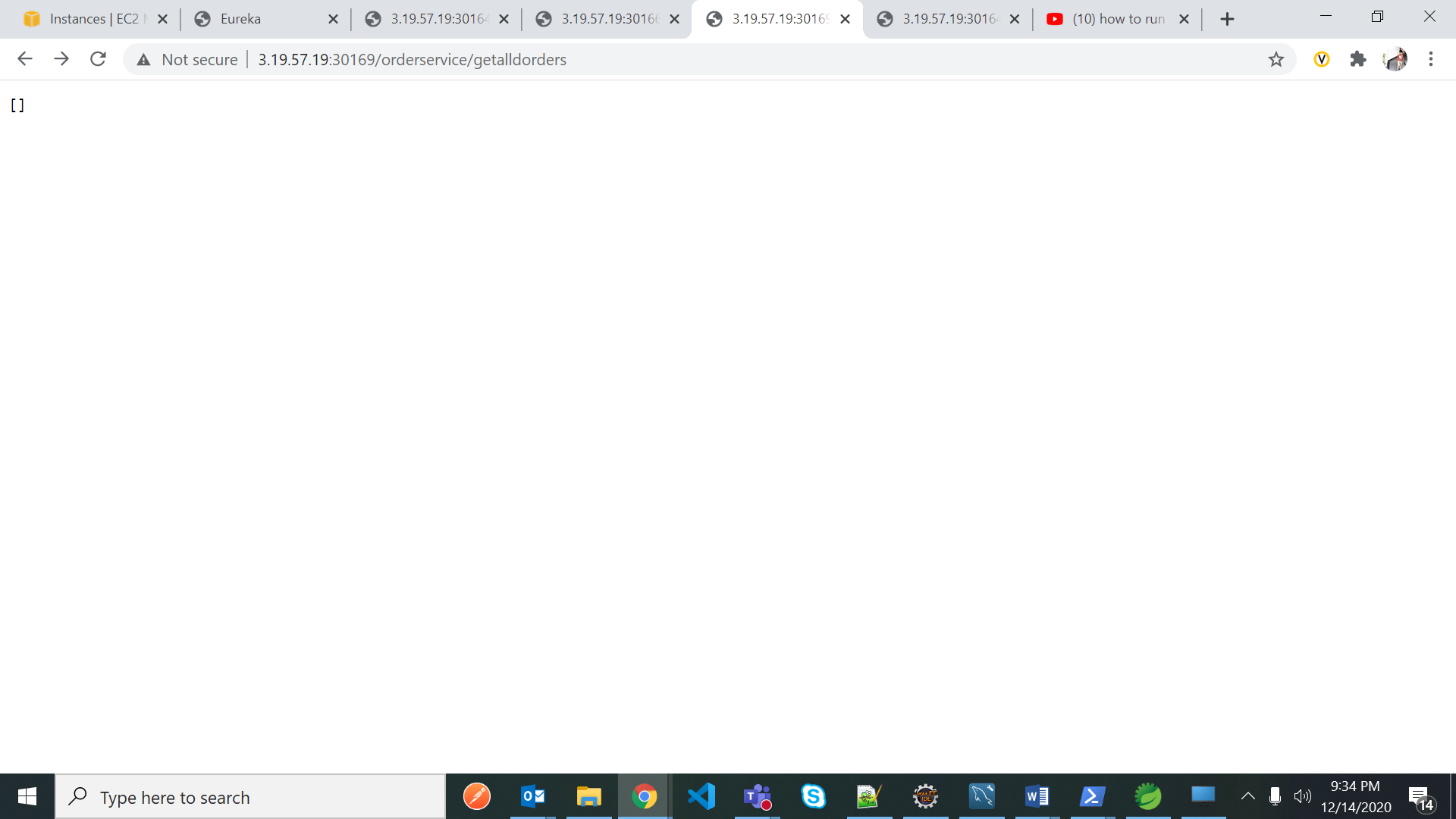


Deploying other 2 microservices following above procedure into Kubernetes in AWS EC2 Ubuntu Instance

Inventory-service : <http://3.19.57.19:30166/inventoryservice/viewproducts>



Order-service: <http://3.19.57.19:30169/orderservice/getalldorders>



commands to redeploy

deleting pods:

root@ip-172-31-29-126:/tmp/order-service/src/main/resources# kubectl delete pods order-service-k8s-f9b475dcb-blzt9

pod "order-service-k8s-f9b475dcb-blzt9" deleted

deleting services

root@ip-172-31-29-126:/tmp/order-service/src/main/resources# kubectl delete svc order-service-k8s

service "order-service-k8s" deleted

deletting deployments

root@ip-172-31-29-126:/tmp/order-service/src/main/resources# kubectl delete deployments order-service-k8s

deployment.apps "order-service-k8s" deleted

deleting docker image

root@ip-172-31-29-126:/tmp/order-service/src/main/resources# docker rmi -f d56d4ad8f347

Untagged: orde-service-k8s:1.0

Deleted: sha256:d56d4ad8f347e16f95aeb66cf2d0e81ac639a0257993bf15def674b0f8def8af

Deleted: sha256:5d0721e7a8572120a5918861103df2407ac24b3aabec45c78d79ed6403ee881b

Deleted: sha256:d31b572e5f7228a385044fbf19ee532d7b812553dafbb48f2c59b8b381b7e29e

to remove the service from tmp folder

root@ip-172-31-29-126:/tmp# sudo rm -r /tmp/order-service

root@ip-172-31-29-126:/tmp# sudo rm -r /tmp/order-service.zip

to move the zip file to tmp folder in AWS EC2 Ubauntu Instance

PS C:\Users\M1058790\Desktop\Adidas membership Backend Rollout\H2-loyaltypoinys exercise> scp -i ..\..\..\Downloads\WebApp.pem .\order-service.zip ubuntu@3.19.57.19:\tmp\

order-service.zip

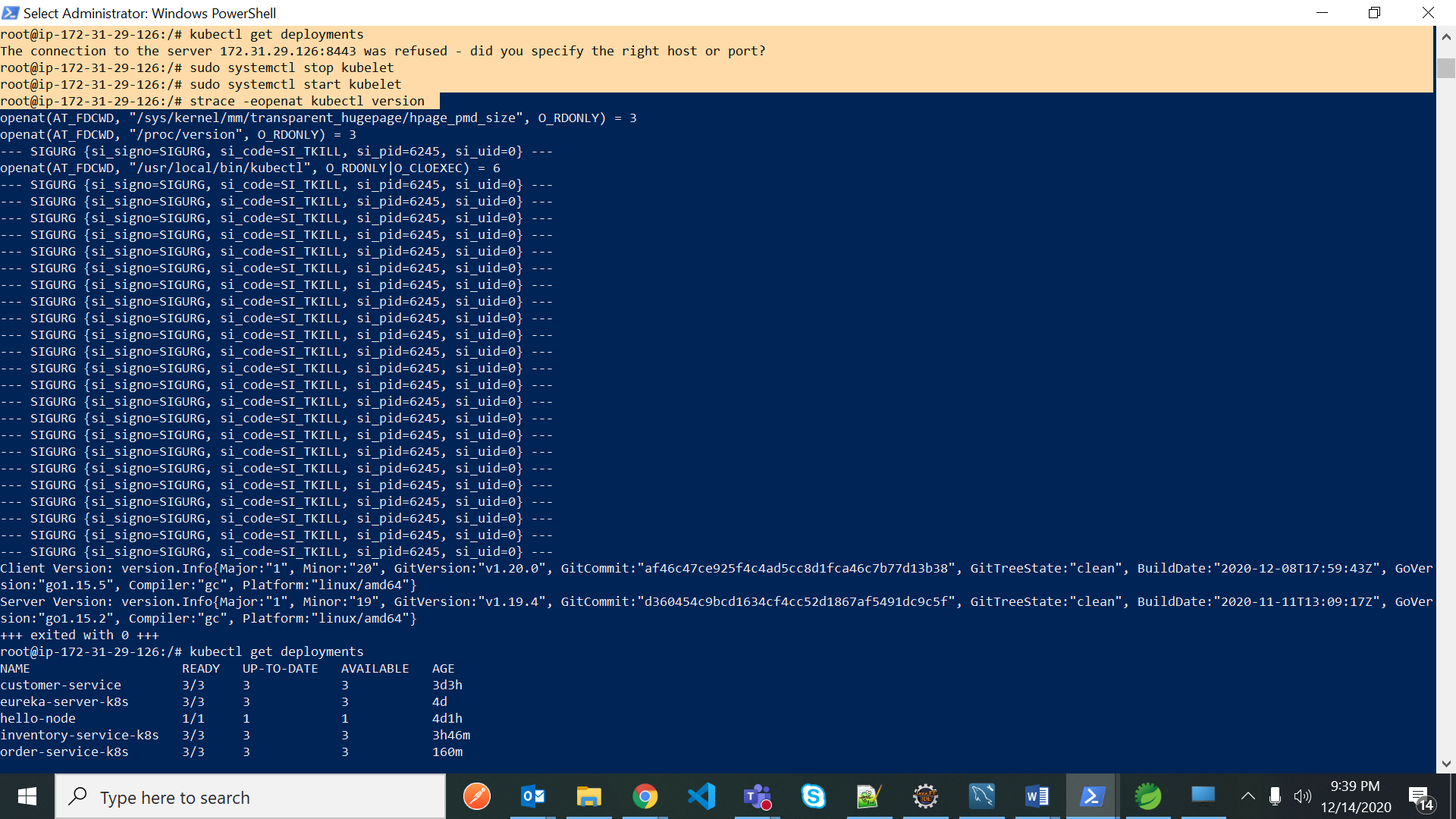
some other imp cmnds

$ sudo vi /etc/mysql/my.cnf

Press I to enter insert mode in the editor

Press esc and then :wq to save and quit the file.

Press esc and then :q to quit the file without saving.



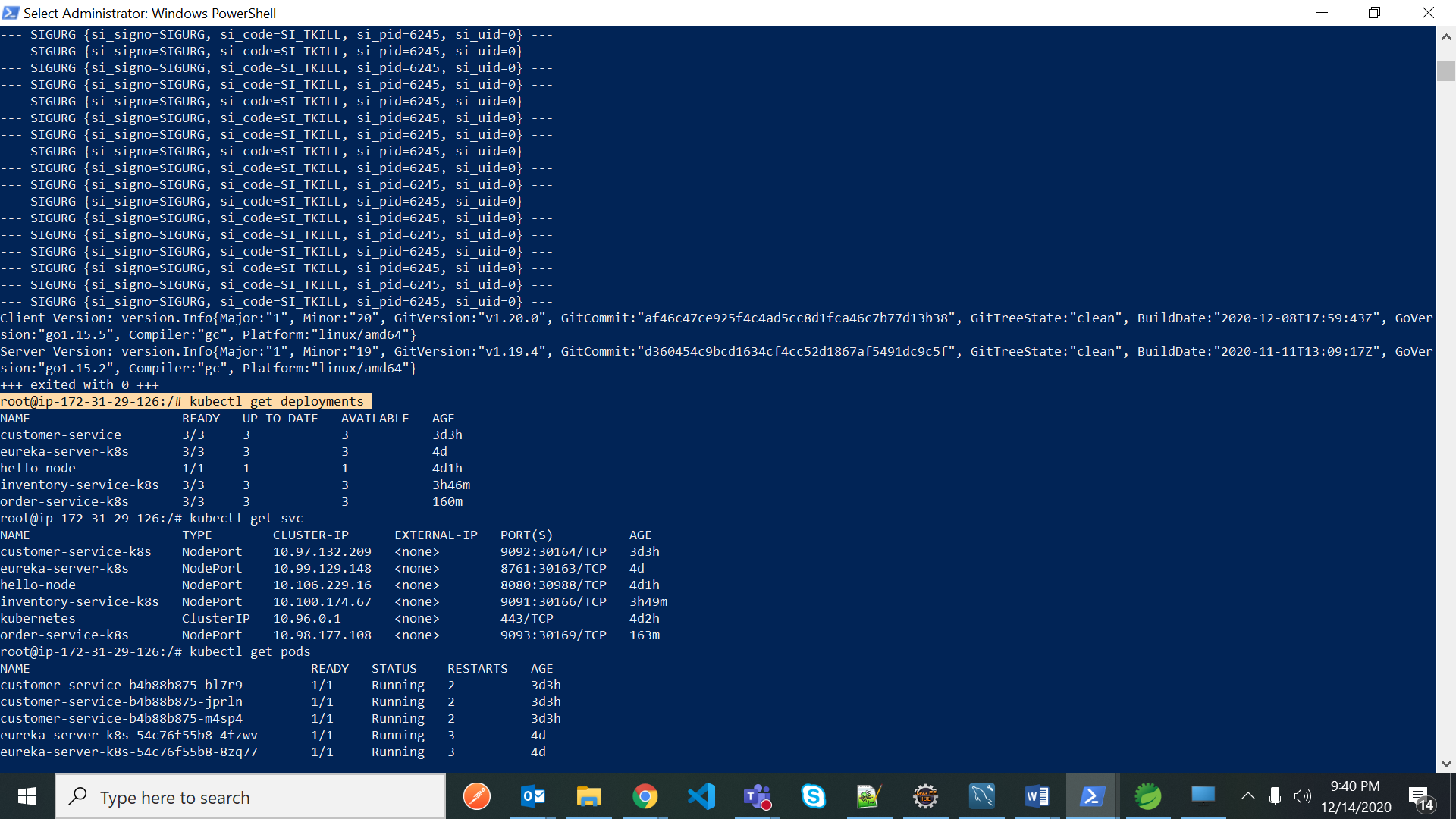
If we get issue like above

Then stop and start kubelet like using below commands

$ sudo systemctl stop kubelet

$ sudo systemctl start kubelet

$ strace -eopenat kubectl version



curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl

chmod +x ./kubctl

sudo mv ./kubectl /usr/local/bin/kubectl

kubectl version --client

sudo apt-get update -y && sudo apt-get install -y docker.io

curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 \

&& chmod +x minikube \

&& sudo mv minikube /usr/local/bin/

apt install conntrack

minikube start --driver=none

kubectl create deployment hello-node --image=k8s.gcr.io/echoserver:1.4

kubectl expose deployment hello-node --type=NodePort --port=8080

kubctl get nodes //no of nodes created

Kubctl get deploymets //deployemt

kubctl get svc //service

kubctl appy -f deployemt.yml

kubctl apply -f service.yml

mvn clean install -Dmaven.test.skip=true

mvn spring-boot:run

FROM adoptopenjdk/openjdk11:alpine-jre

ADD target/springboot-example.jar app.jar

ENTRYPOINT ["java","-jar","app.jar"]

scp -i ..\..\Downloads\WebApp.pem .\eureka-server.zip ubuntu@18.191.177.205:\tmp\

kubectl set image deployment/customer-service customer-service=customer-service:2.0

$ sudo vi /etc/mysql/my.cnf

Press I to enter insert mode in the editor

Press esc and then :wq to save and quit the file.

$ sudo systemctl stop kubelet

$ sudo systemctl start kubelet

$ strace -eopenat kubectl version