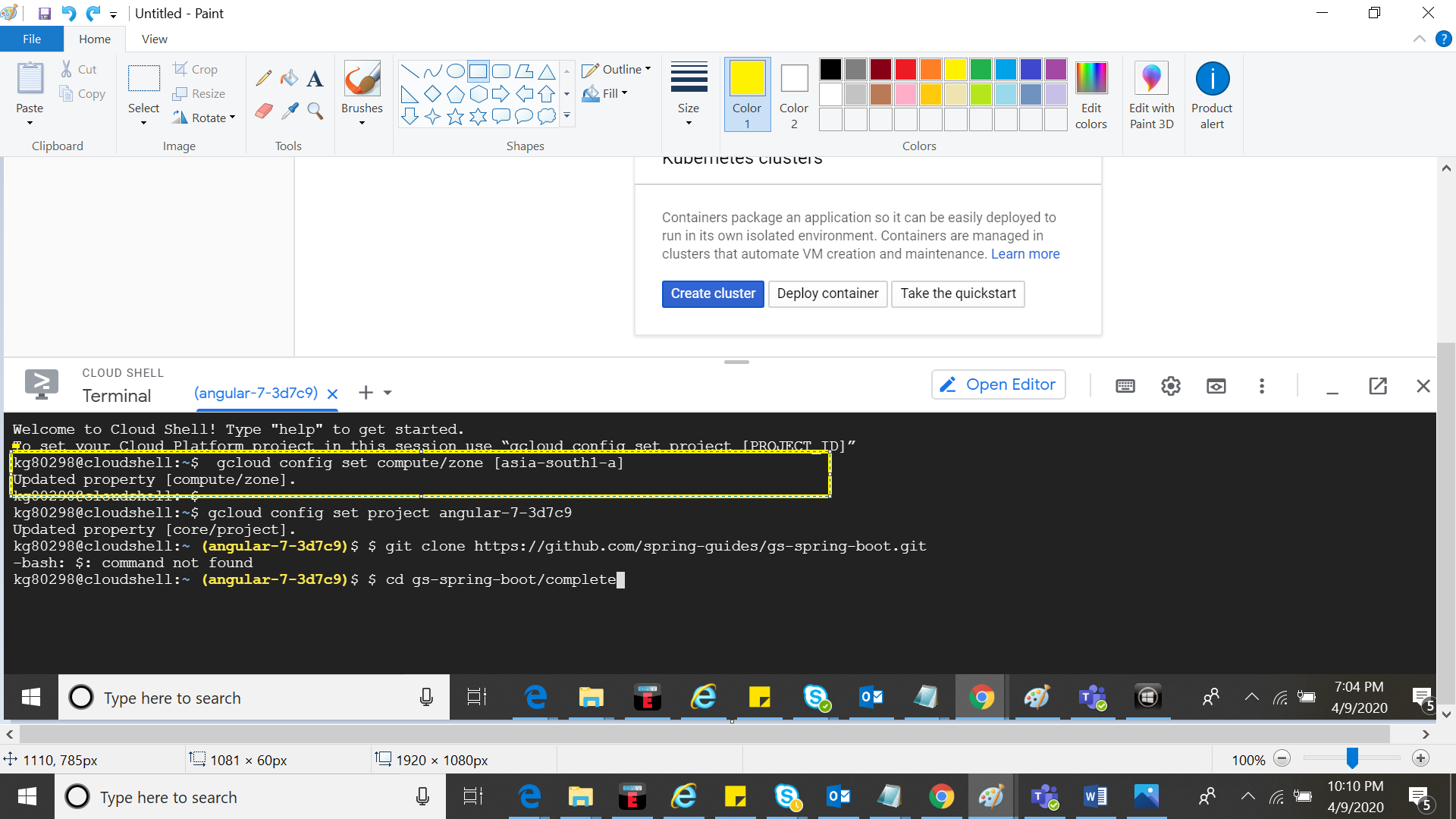
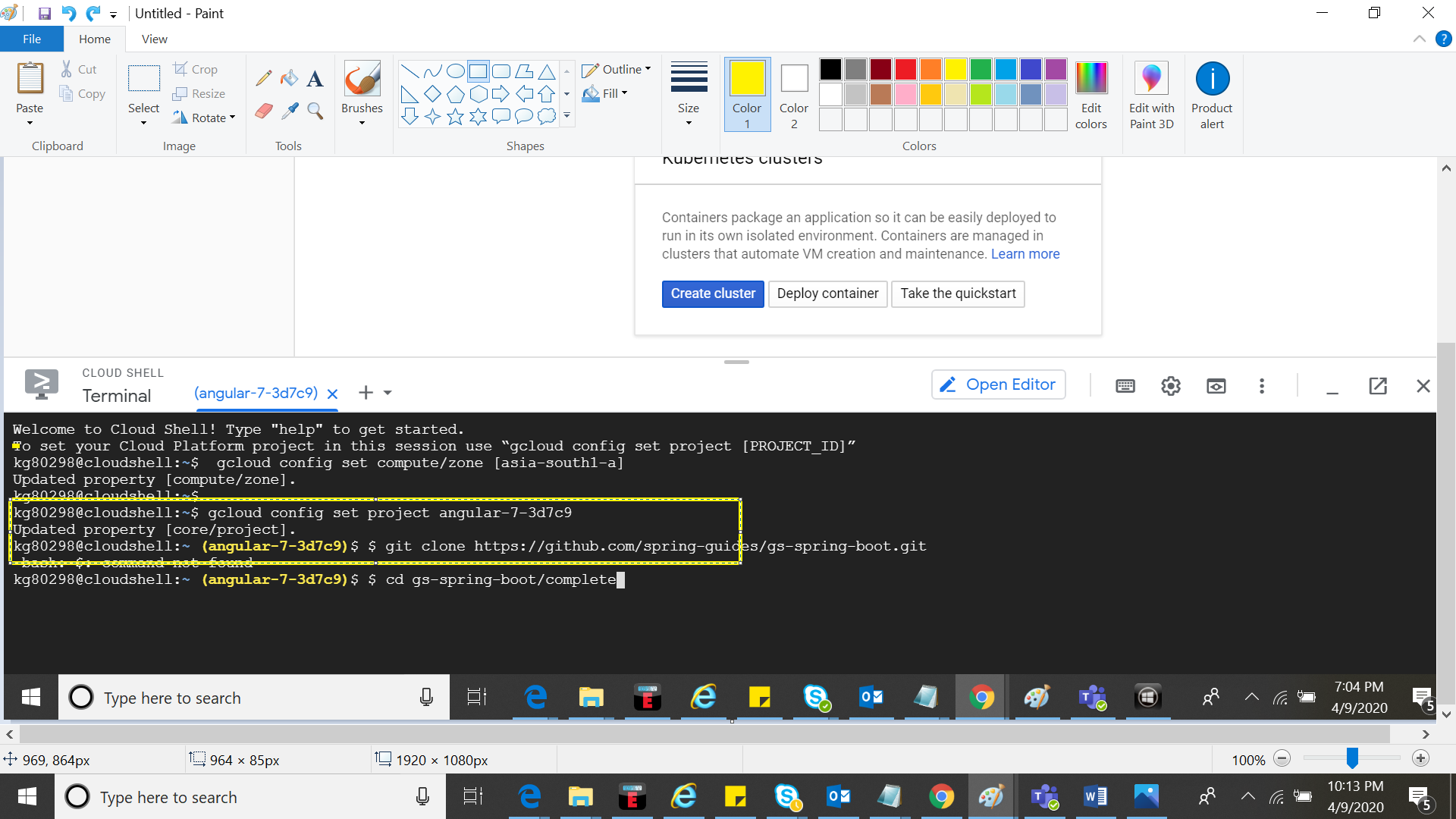
1. Setting up the zone: **gcloud config set compute/zone [asia-south1-a]**

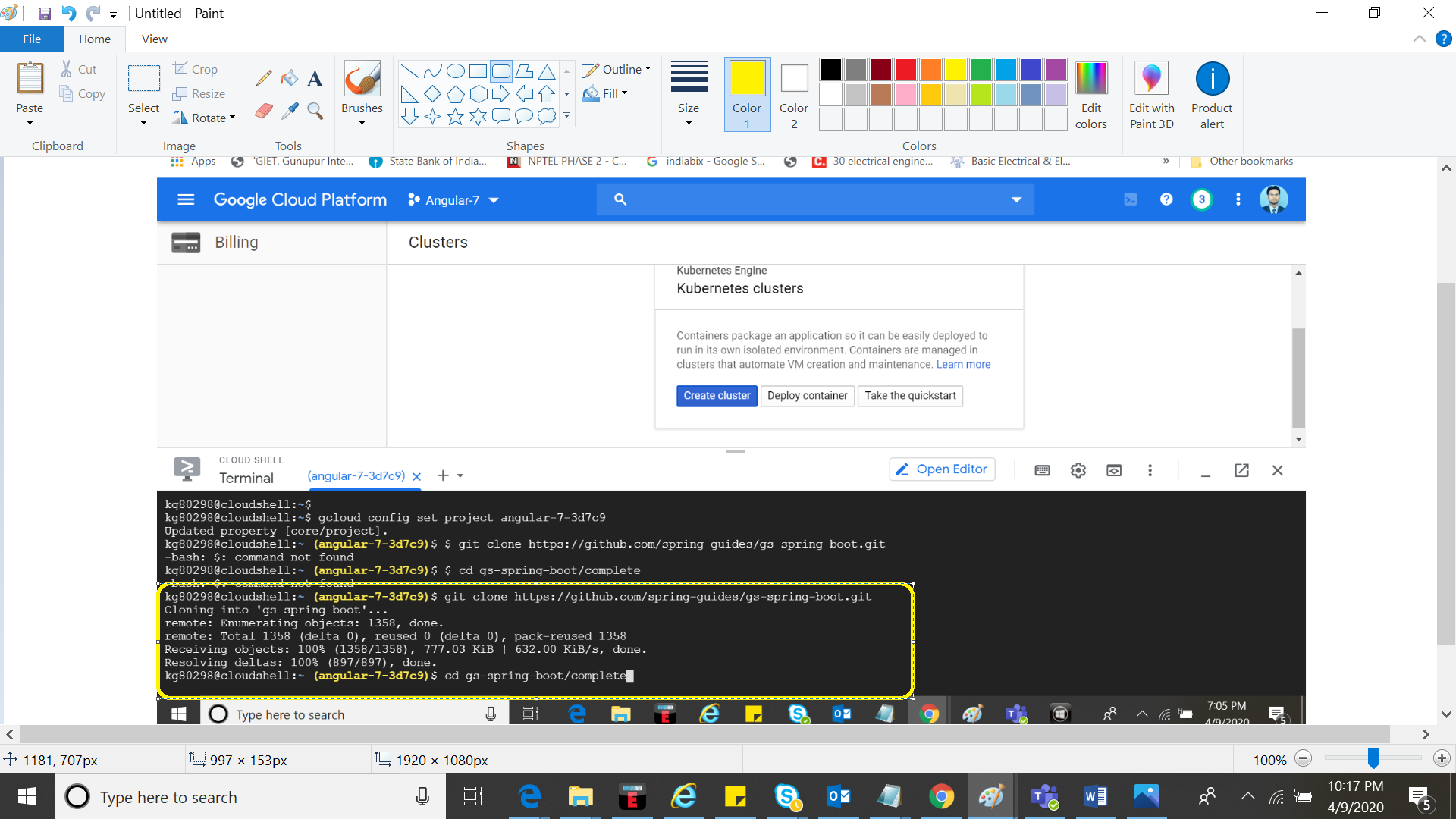


1. Setting project ID: **gcloud config set project angular-7-3d7c9**

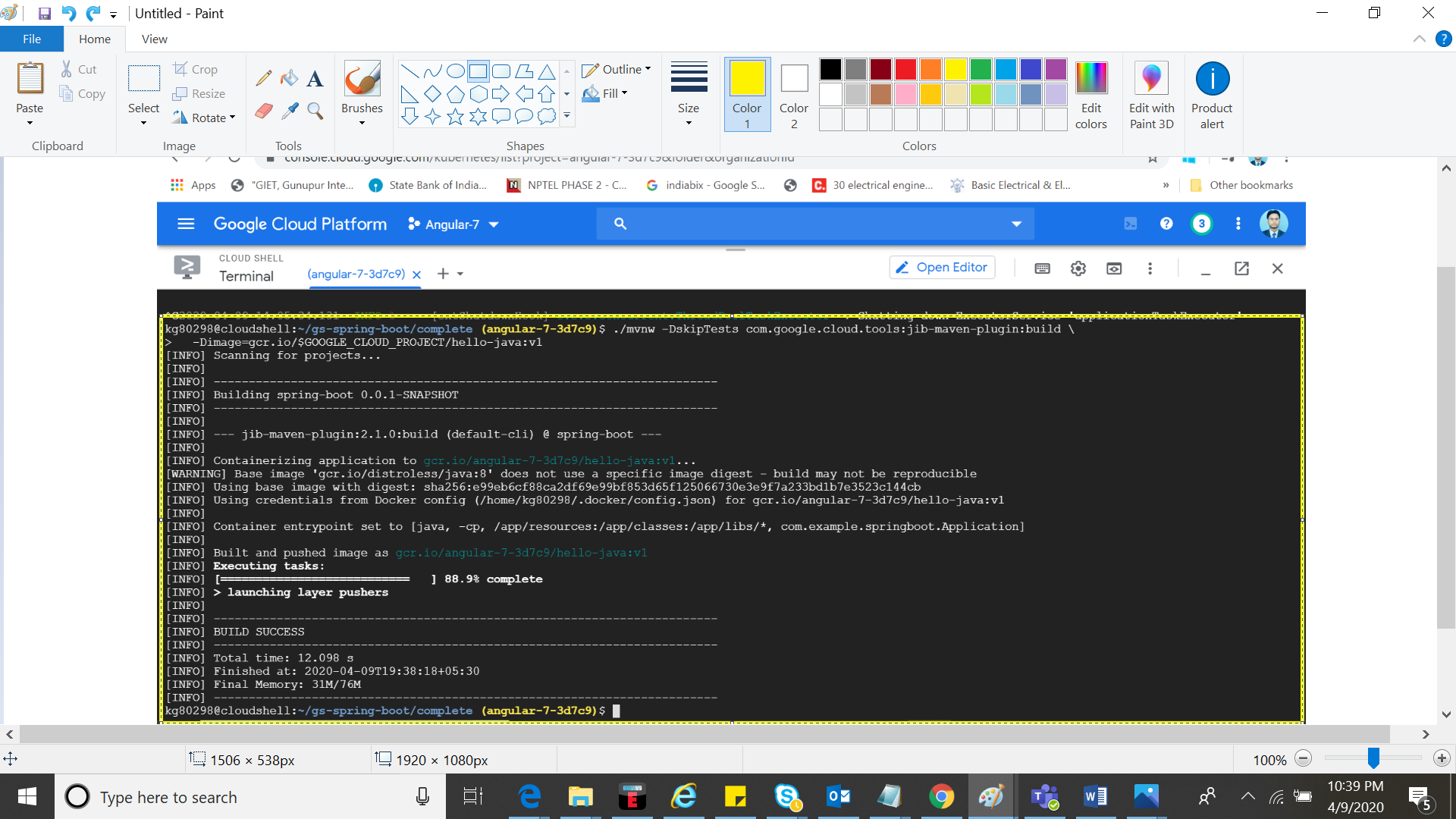


1. Cloning the application:- **git clone https://github.com/spring-guides/gs-spring-boot.git**

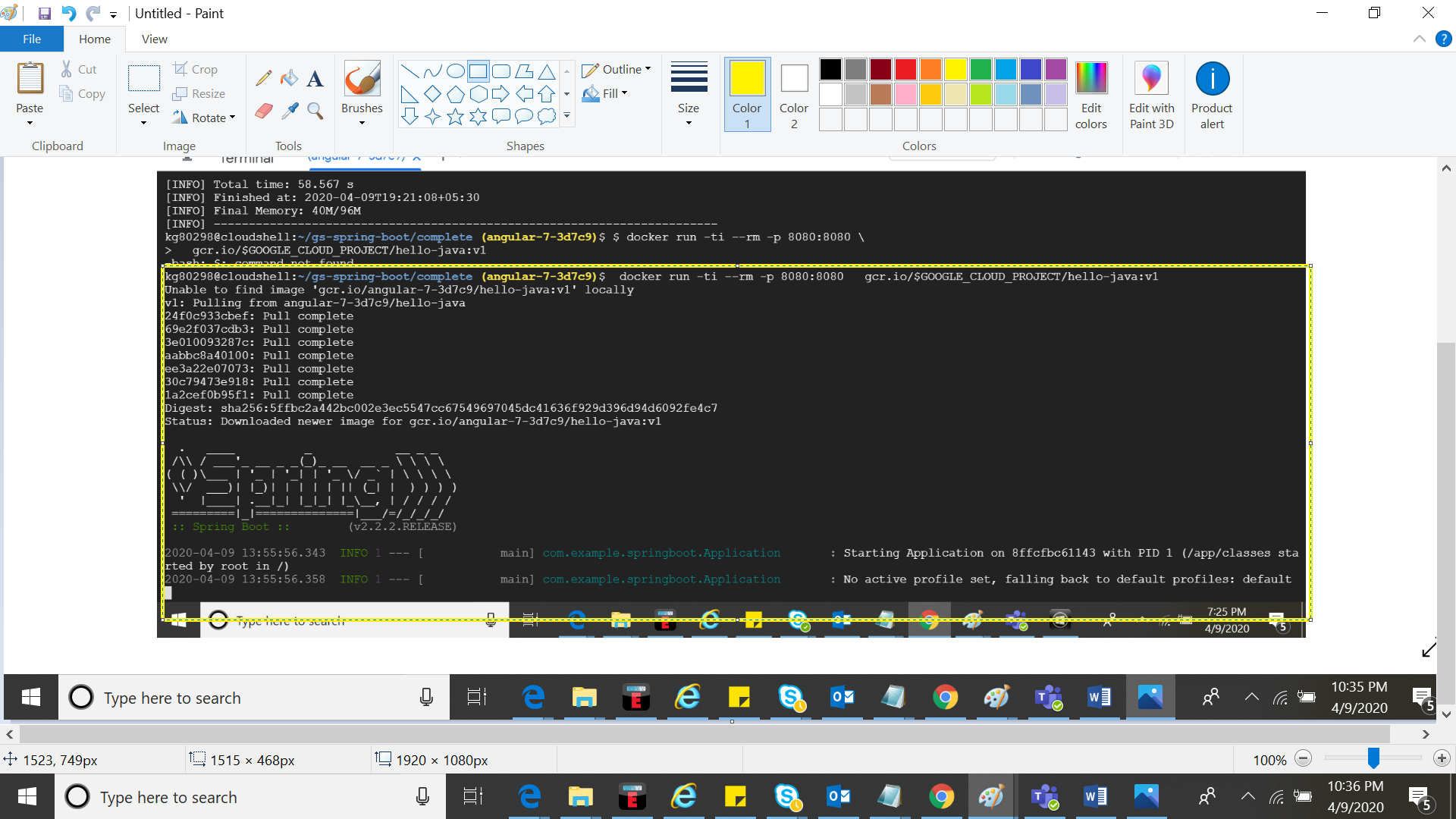
**cd gs-spring-boot/complete**



1. Build the container image using jib: **./mvnw -DskipTests com.google.cloud.tools:jib-maven-plugin:build \ -Dimage=gcr.io/$GOOGLE\_CLOUD\_PROJECT/hello-java:v1**



1. Running docker to Preview the application image:- **docker run -ti --rm -p 8080:8080 gcr.io/$GOOGLE\_CLOUD\_angular-7-3d7c9/hello-java:v1**



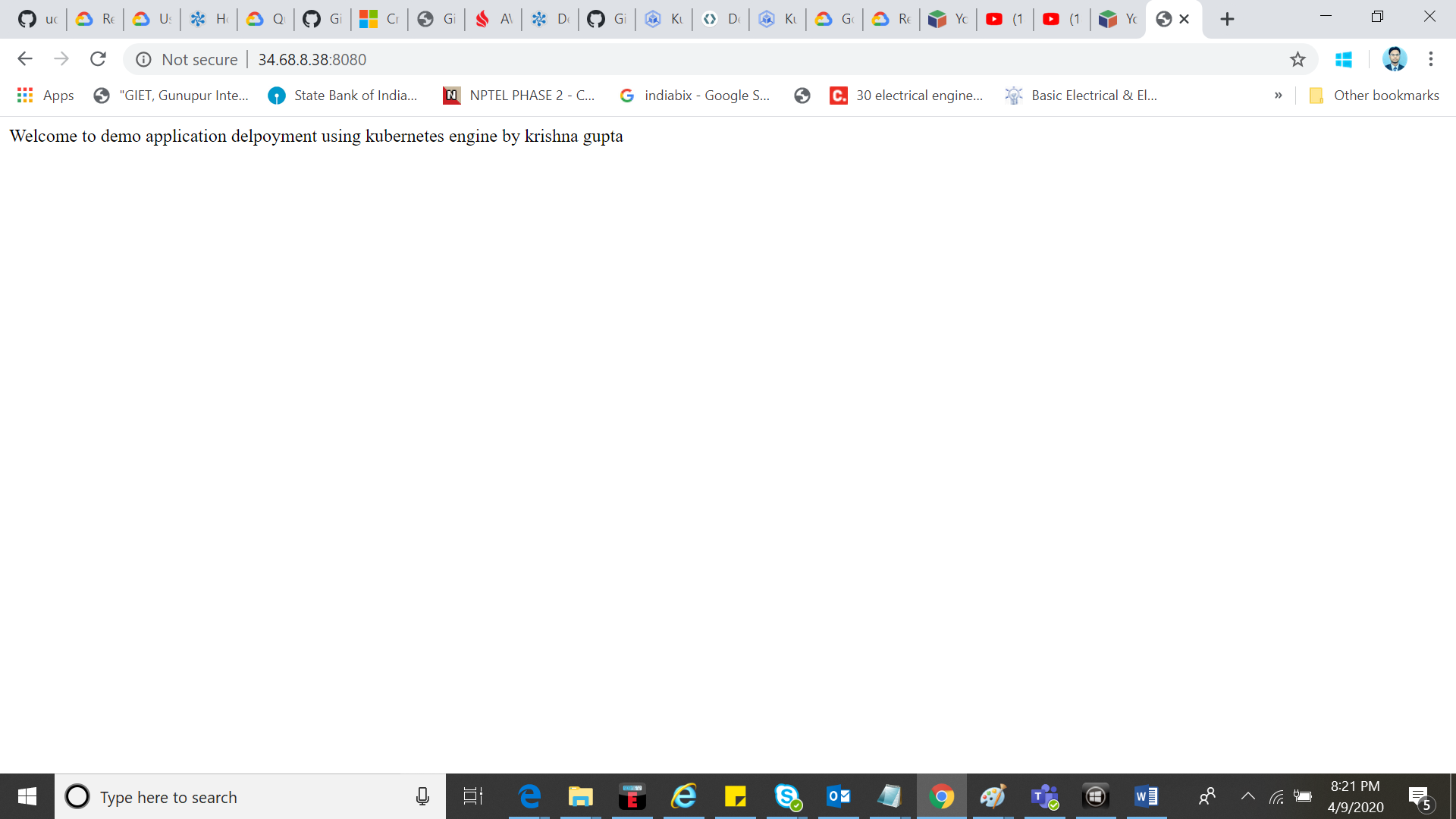
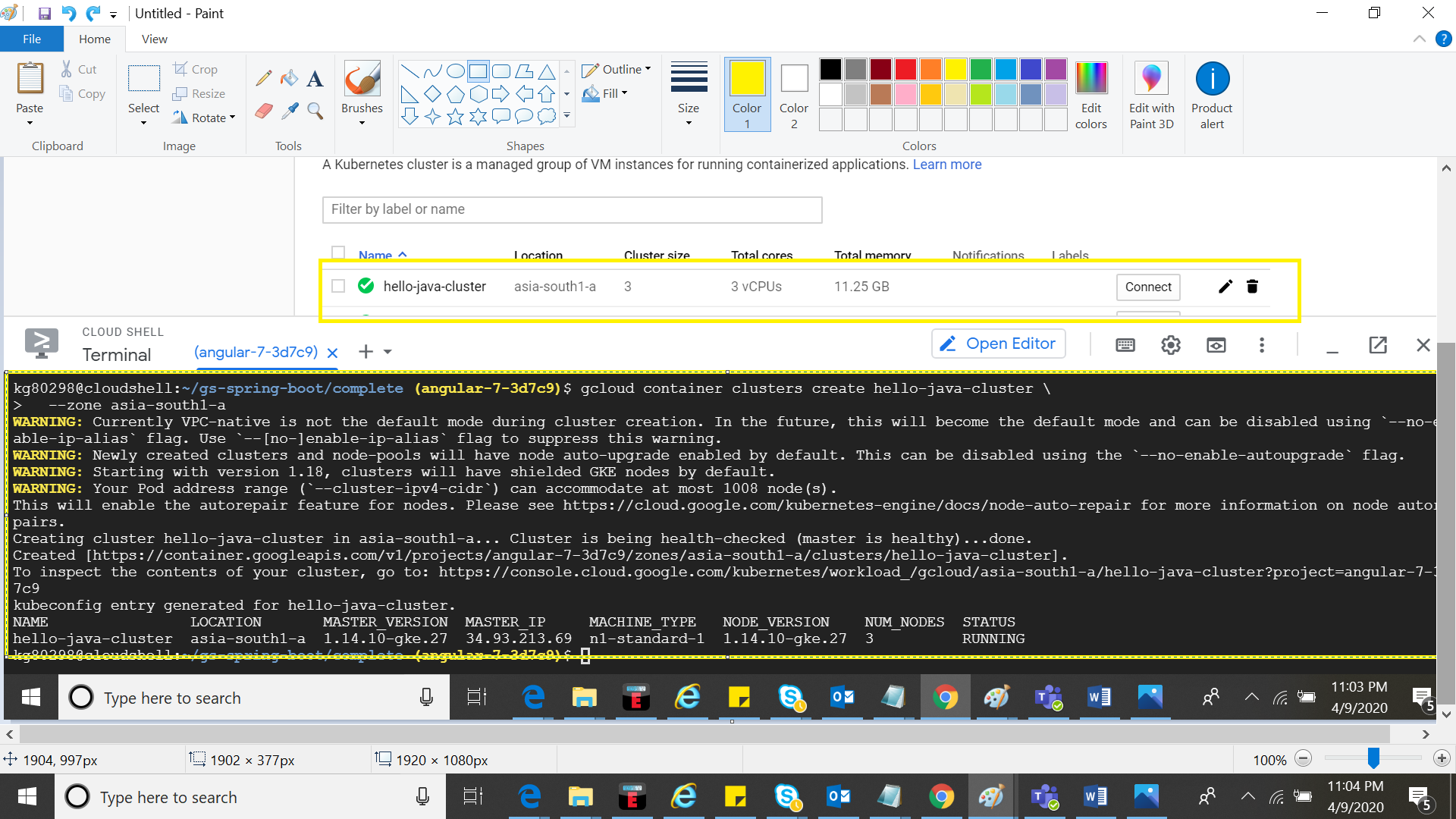
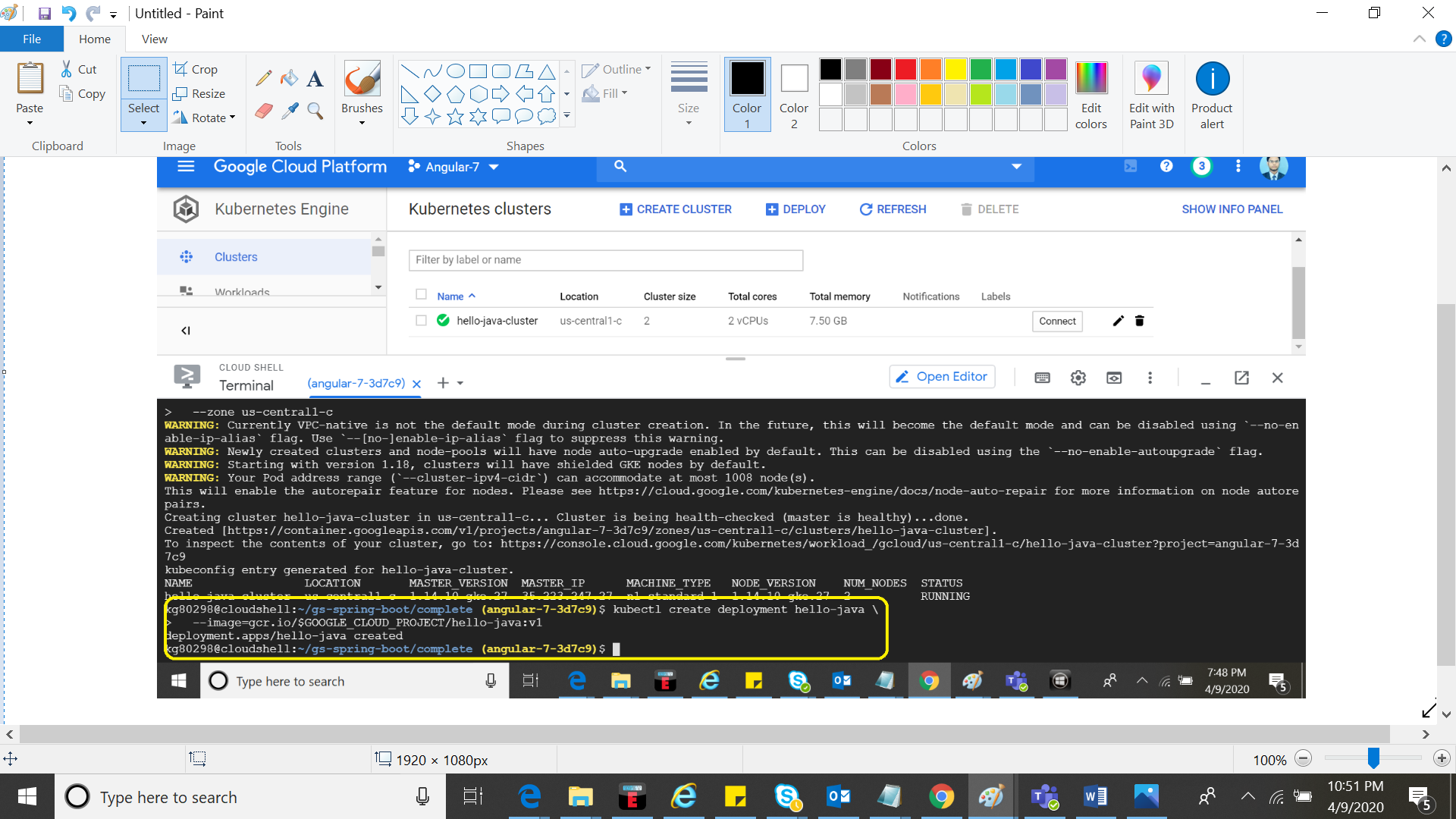


Fig –docker image

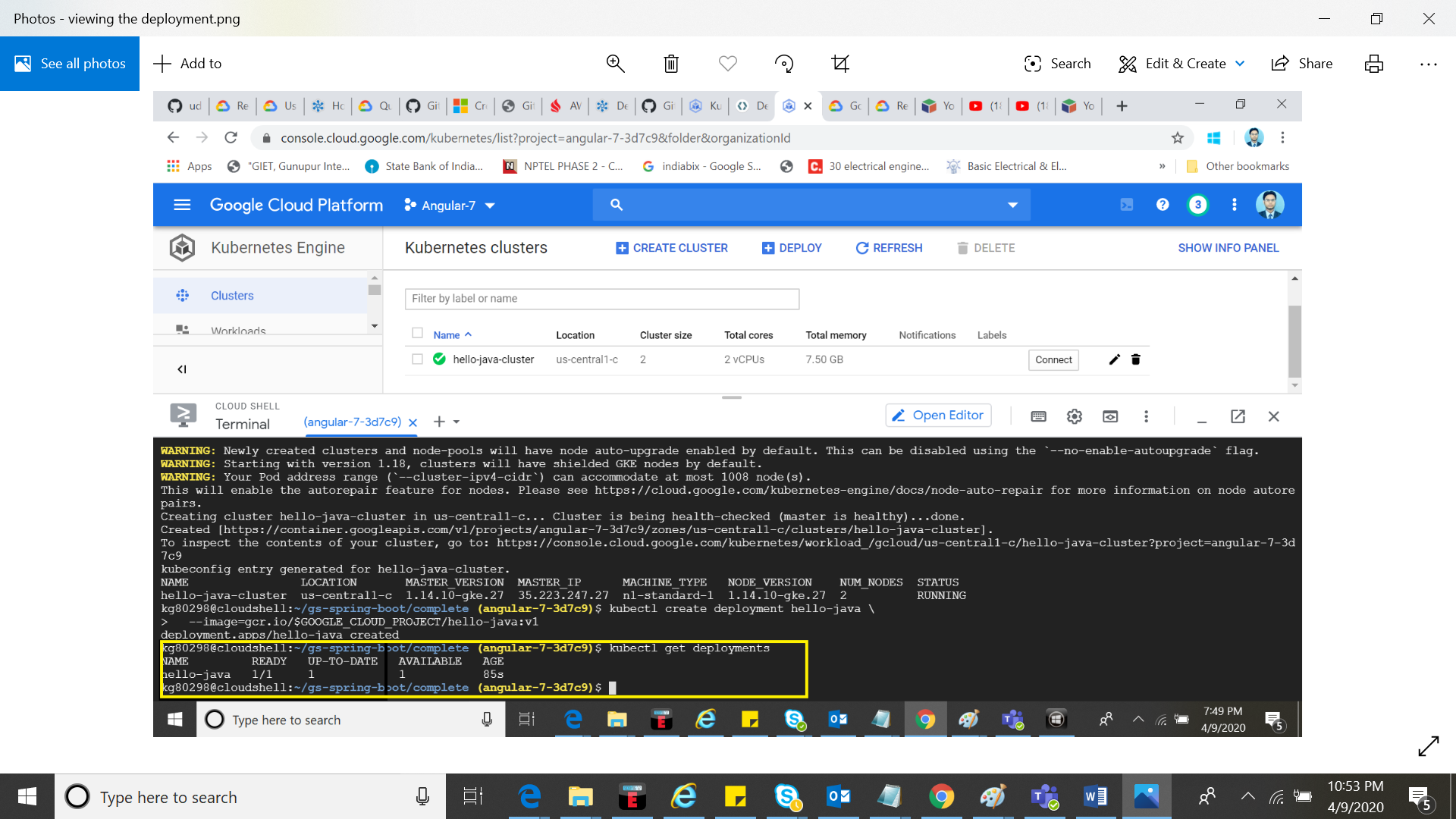
1. Creating Cluster:- **gcloud container clusters create hello-java-cluster \--zone asia-south1-a**



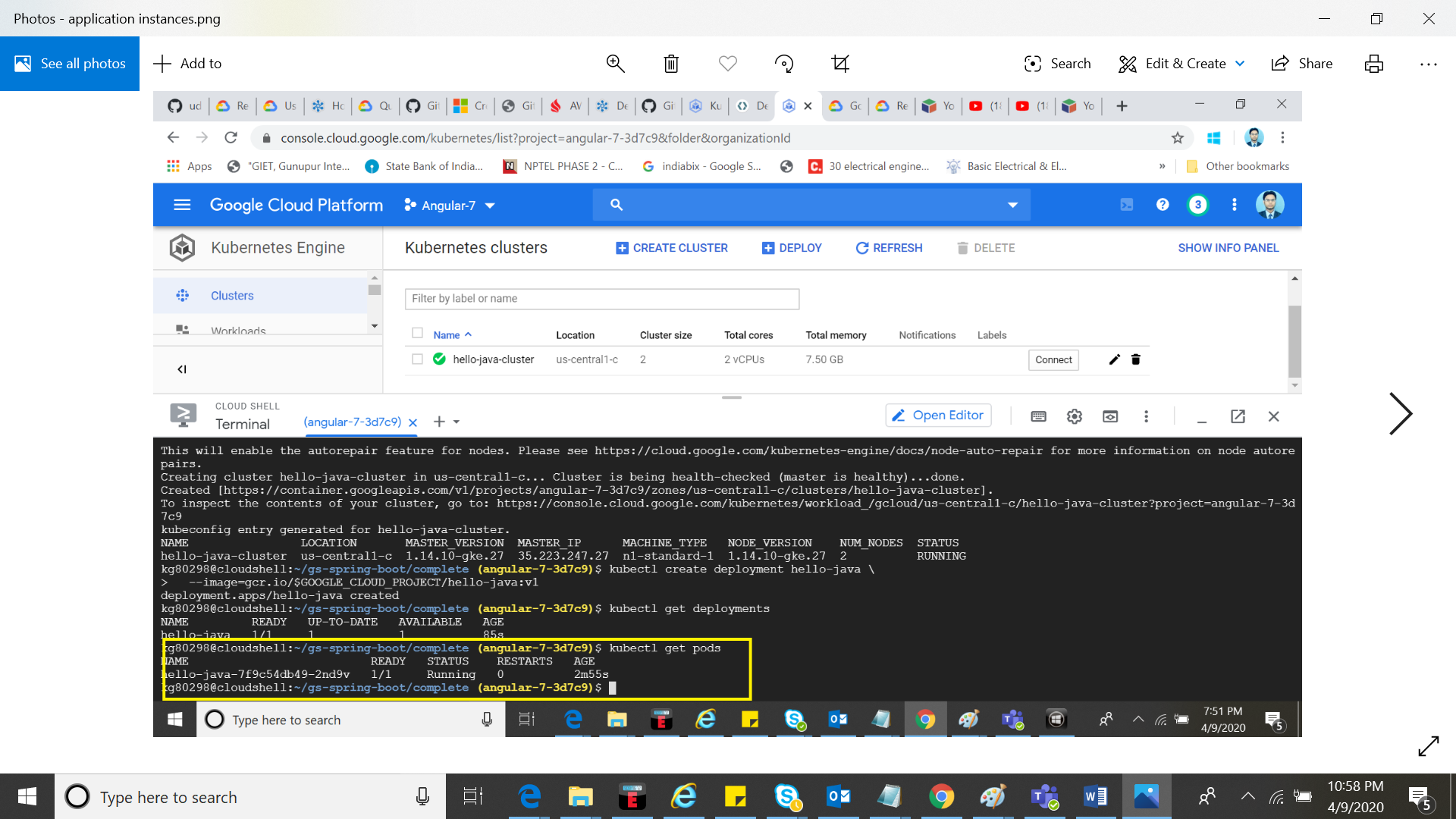
1. Deploying application to kubernetes:- **kubectl create deployment hello-java \ --image=gcr.io/$GOOGLE\_CLOUD\_PROJECT/hello-java:v1**



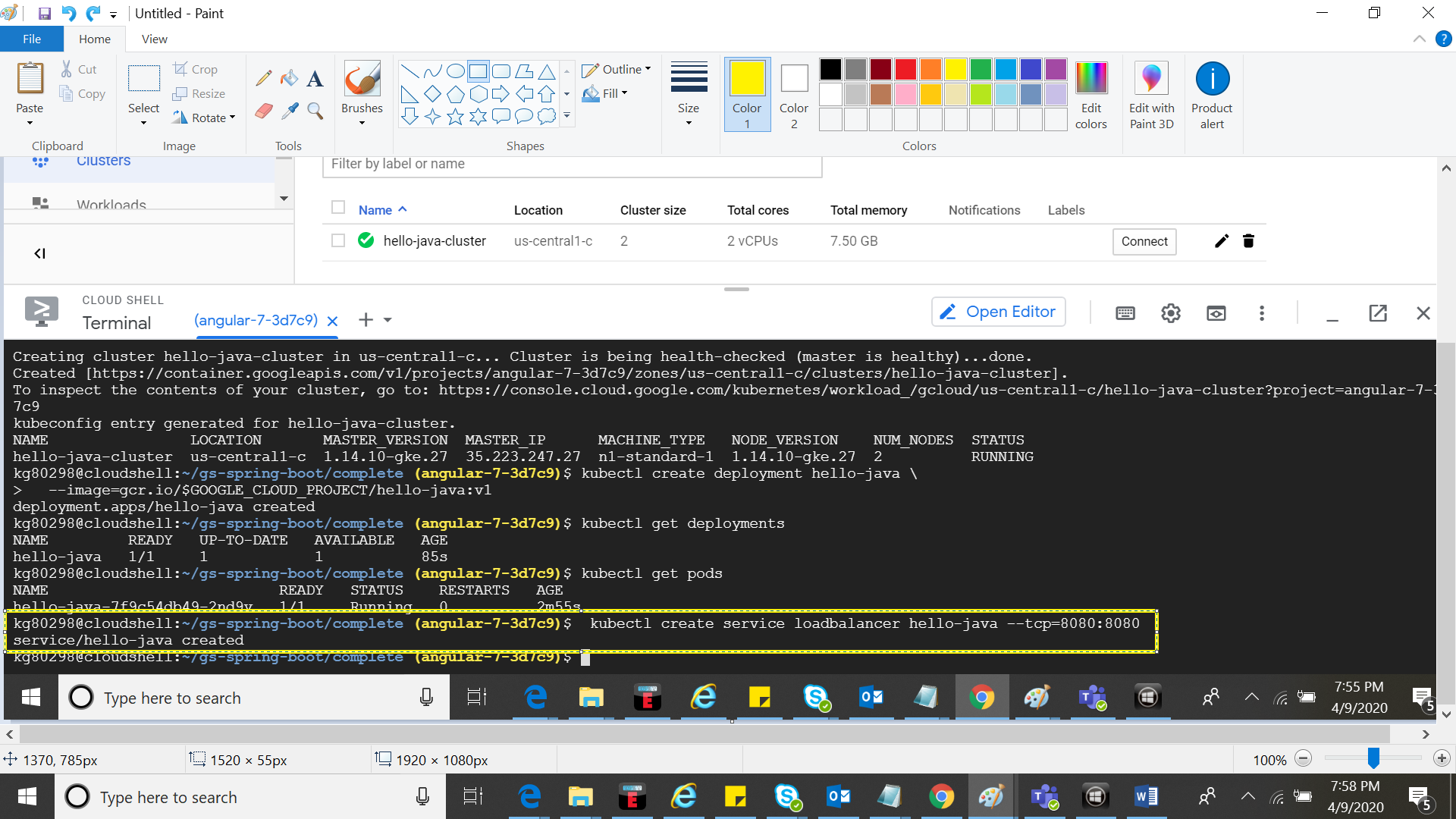
1. Viewing the deployment:- **kubectl get deployments**



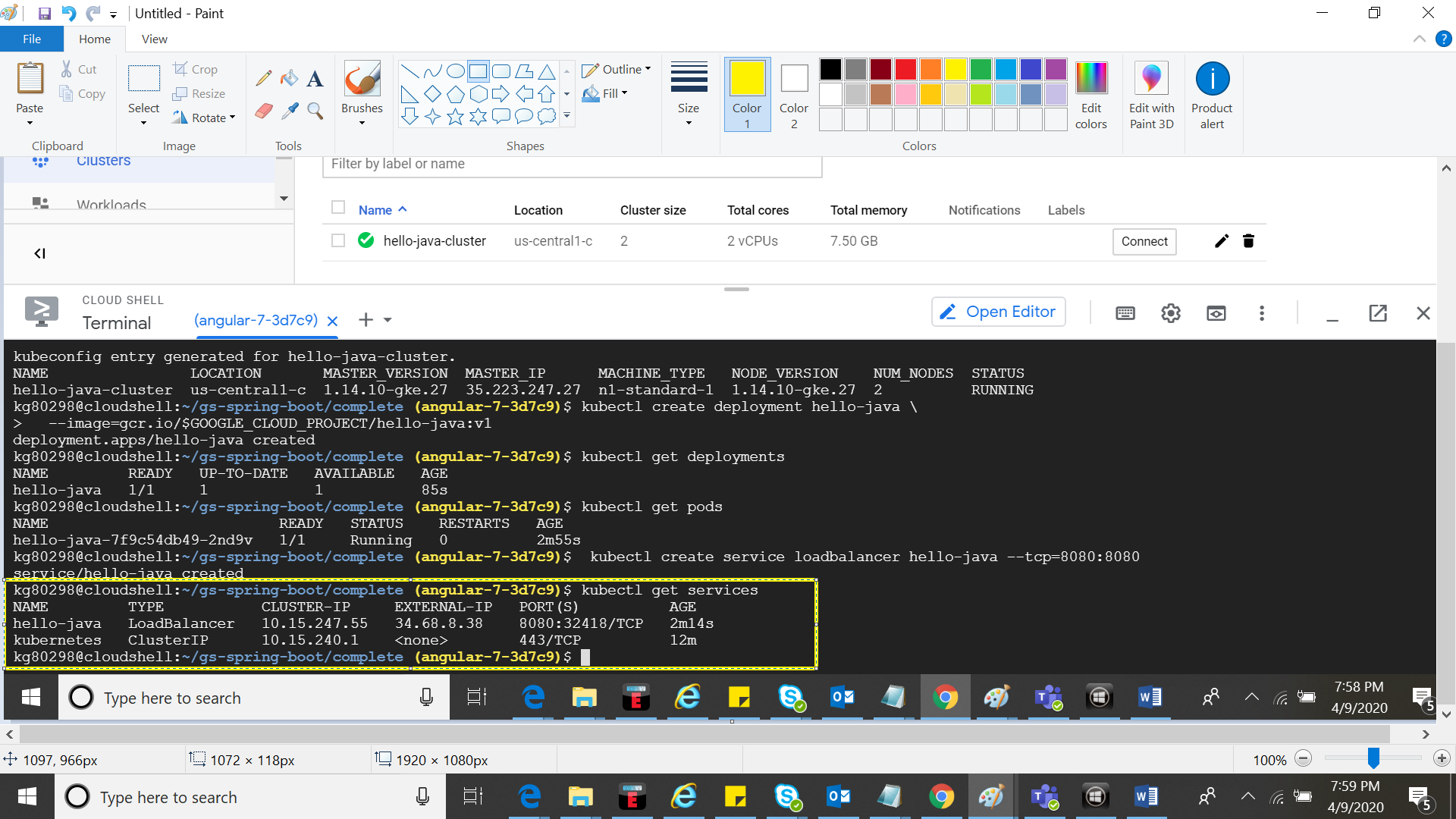
1. viewing the application instances created by the deployment:- **kubectl get pods**



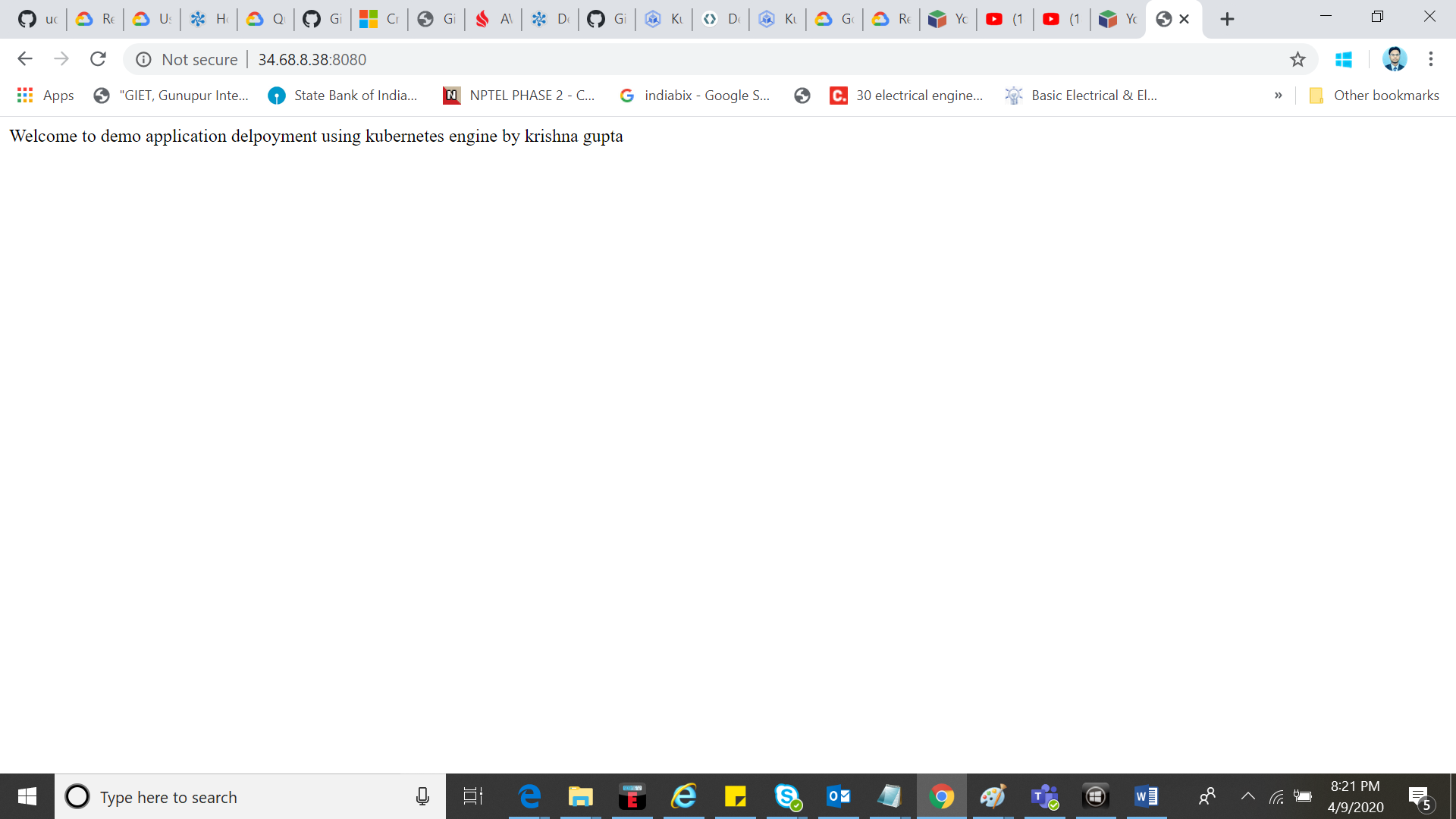
1. Creating LoadBalancer services for the external network to access app: **kubectl create service loadbalancer hello-java --tcp=8080:8080**



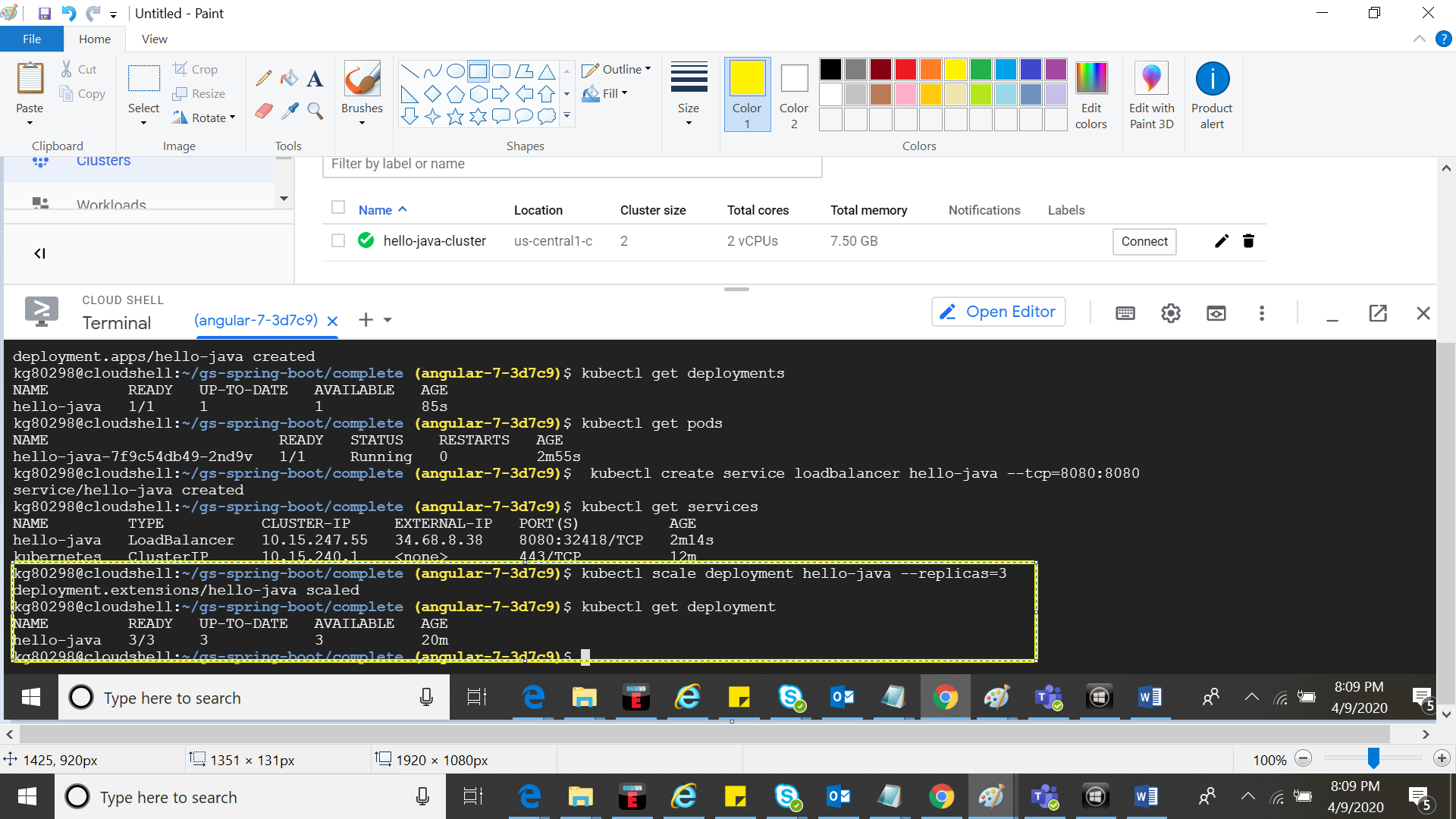
1. To check all the cluster services: **kubectl get services**



External IP: **http://34.68.8.38:8080**

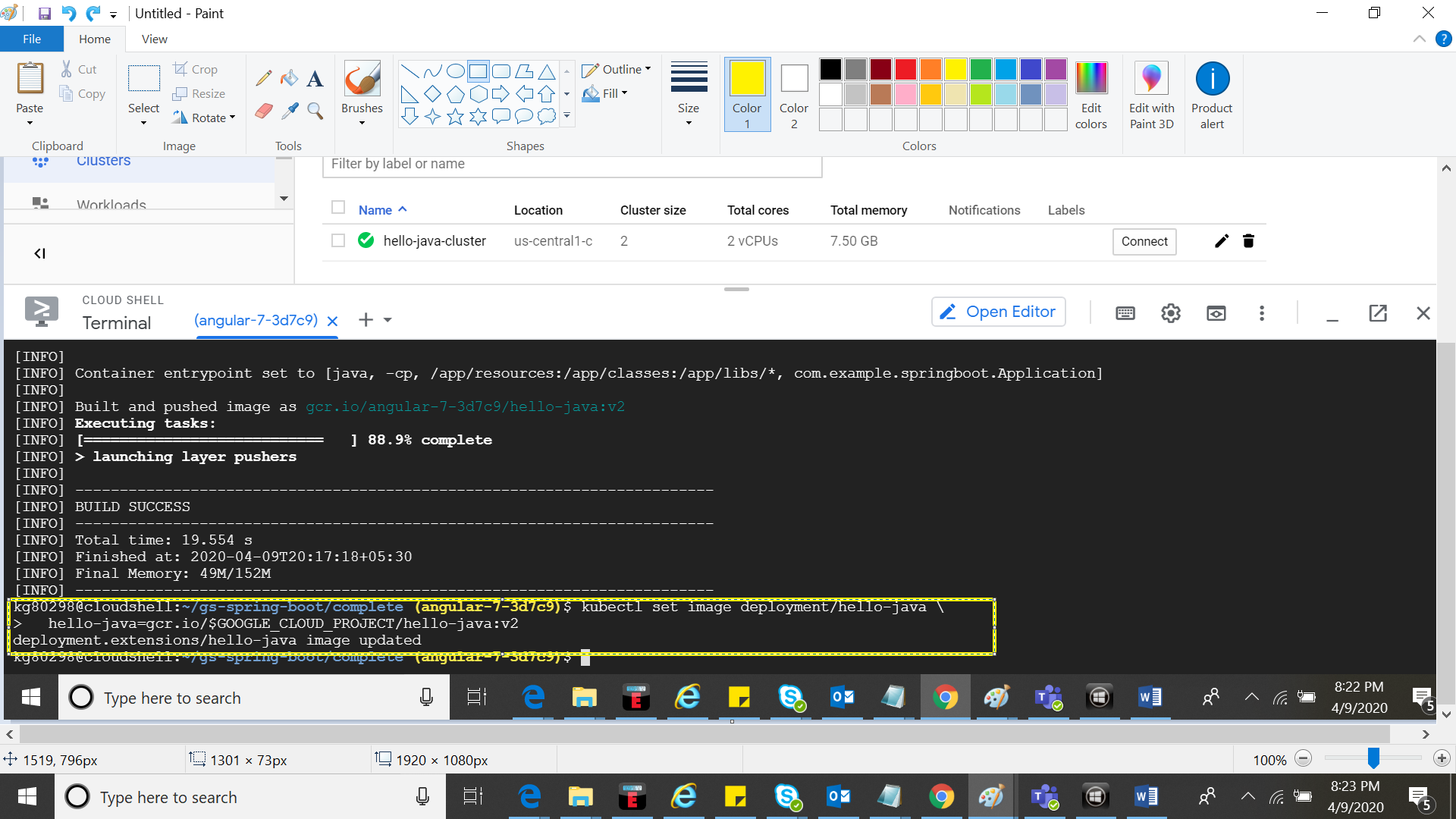


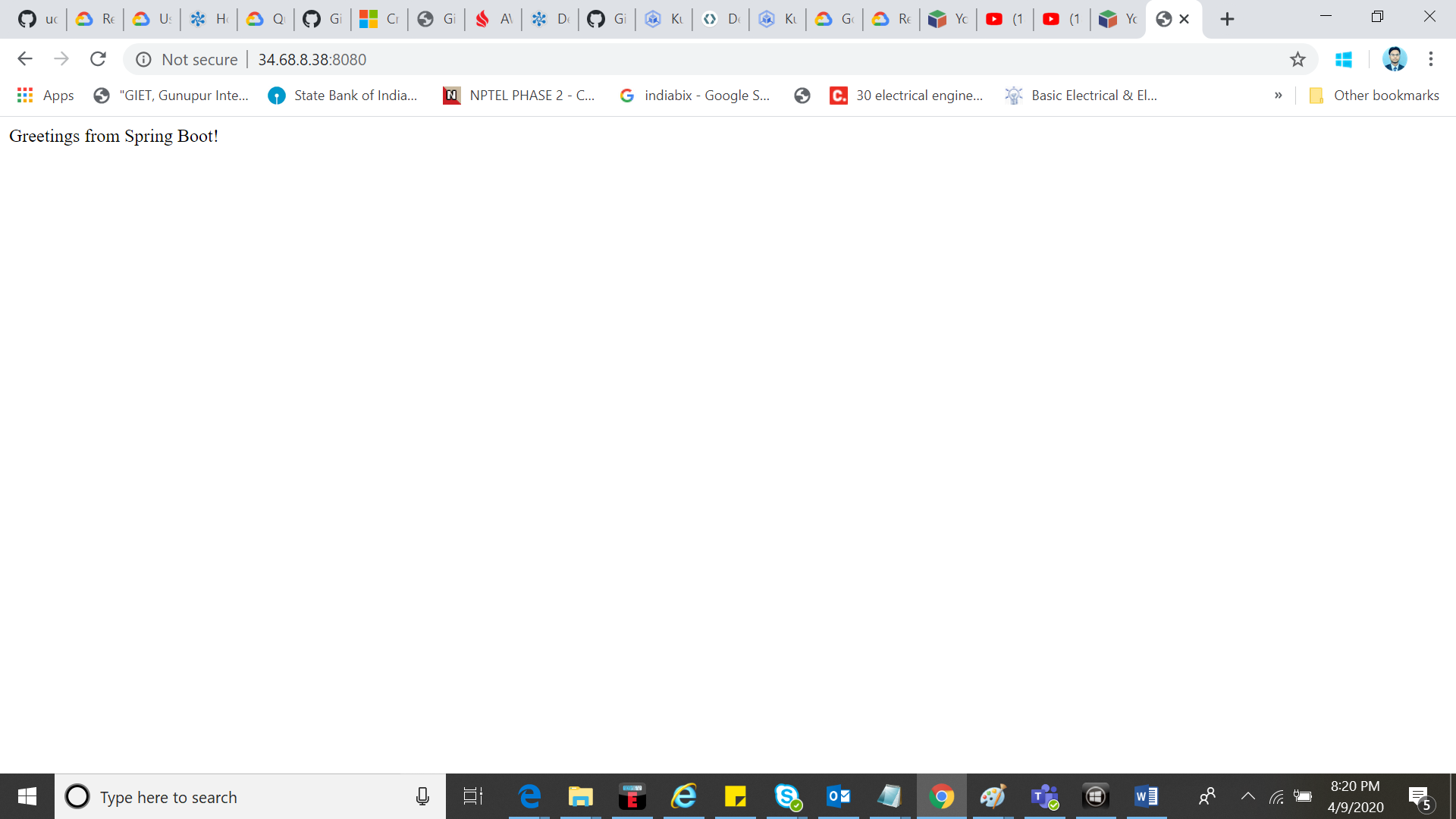
1. Scaling the service:- **kubectl scale deployment hello-java --replicas=3**



13. Editing the application and creating new image.

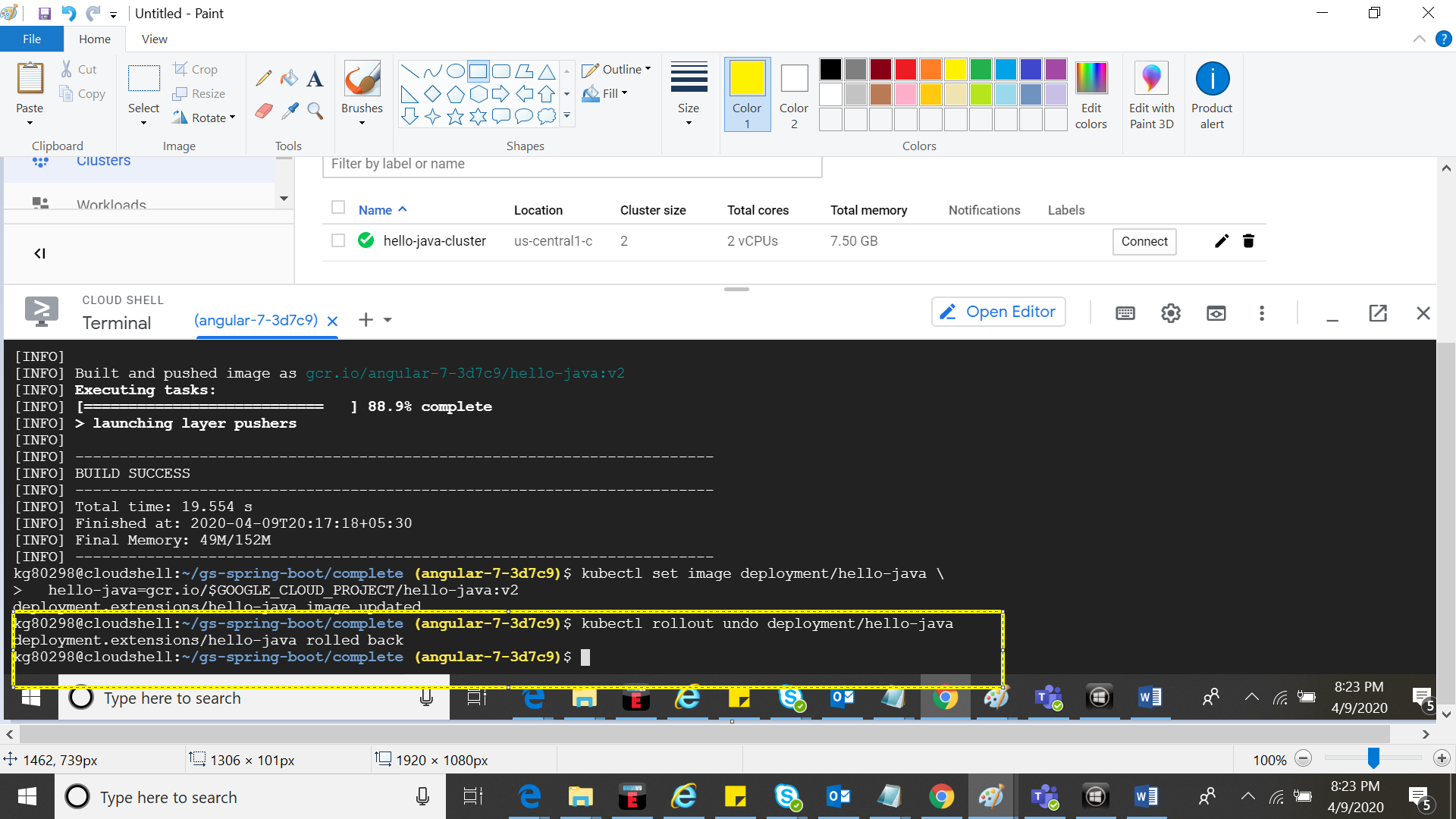
**kubectl set image deployment/hello-java \ hello-java=gcr.io/$GOOGLE\_CLOUD\_PROJECT/hello-java:v2**

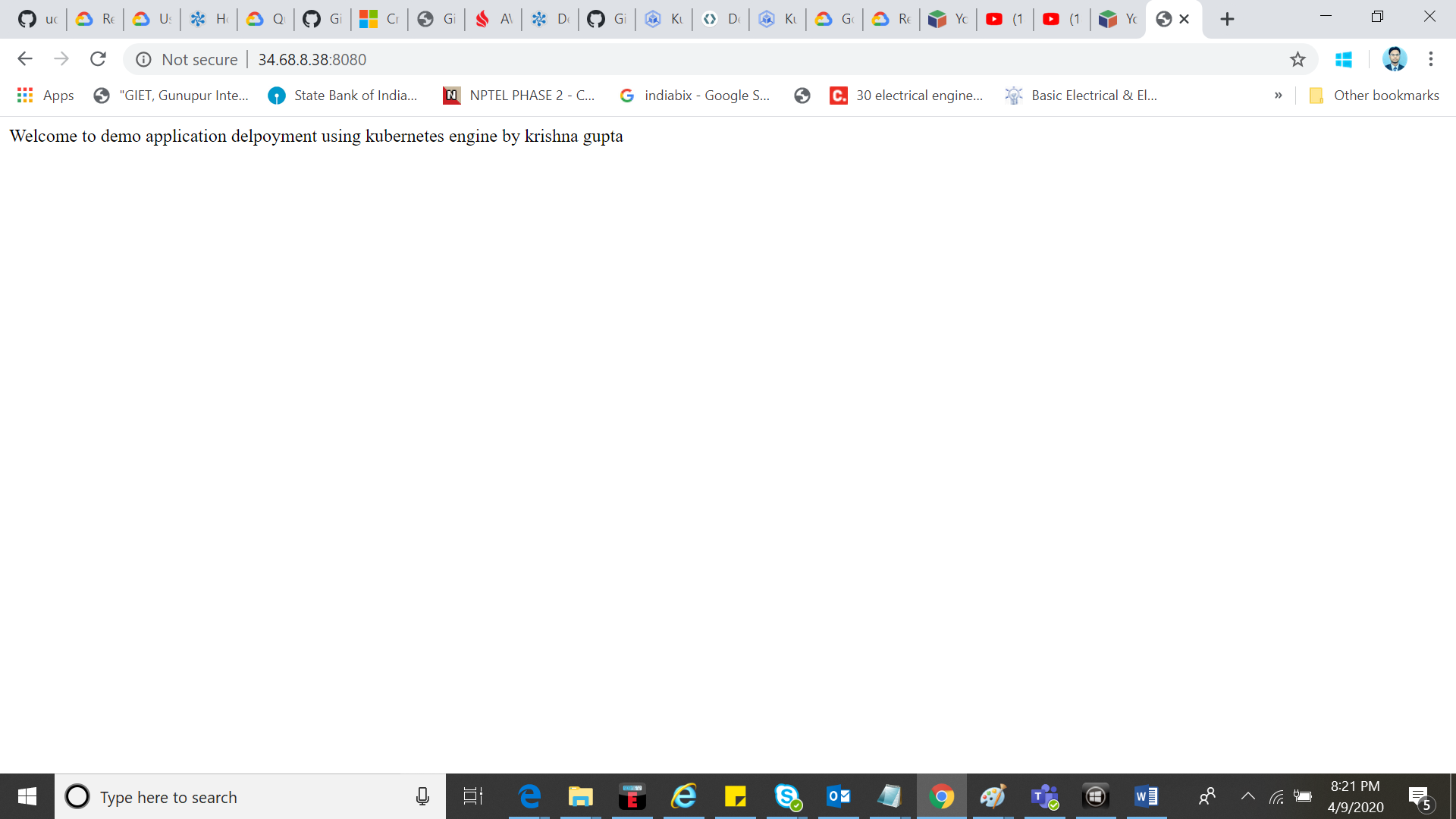




1. Roll Back application**:**

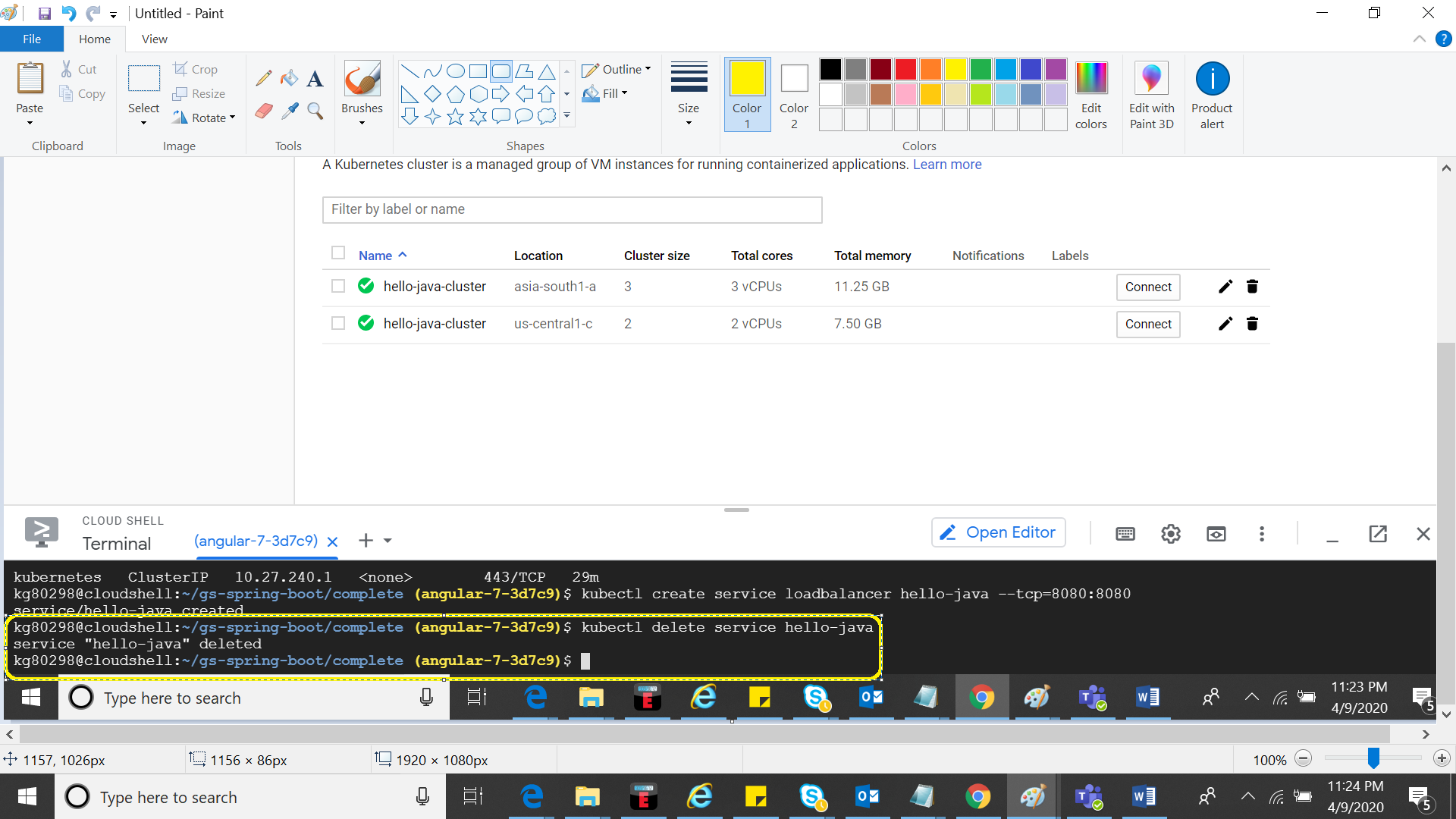
**kubectl rollout undo deployment/hello-java**





1. **Cleanup**

**kubectl delete service hello-java**



**gcloud container clusters delete hello-java**

