To see which version of docker installed

>docker version

To view the docker images

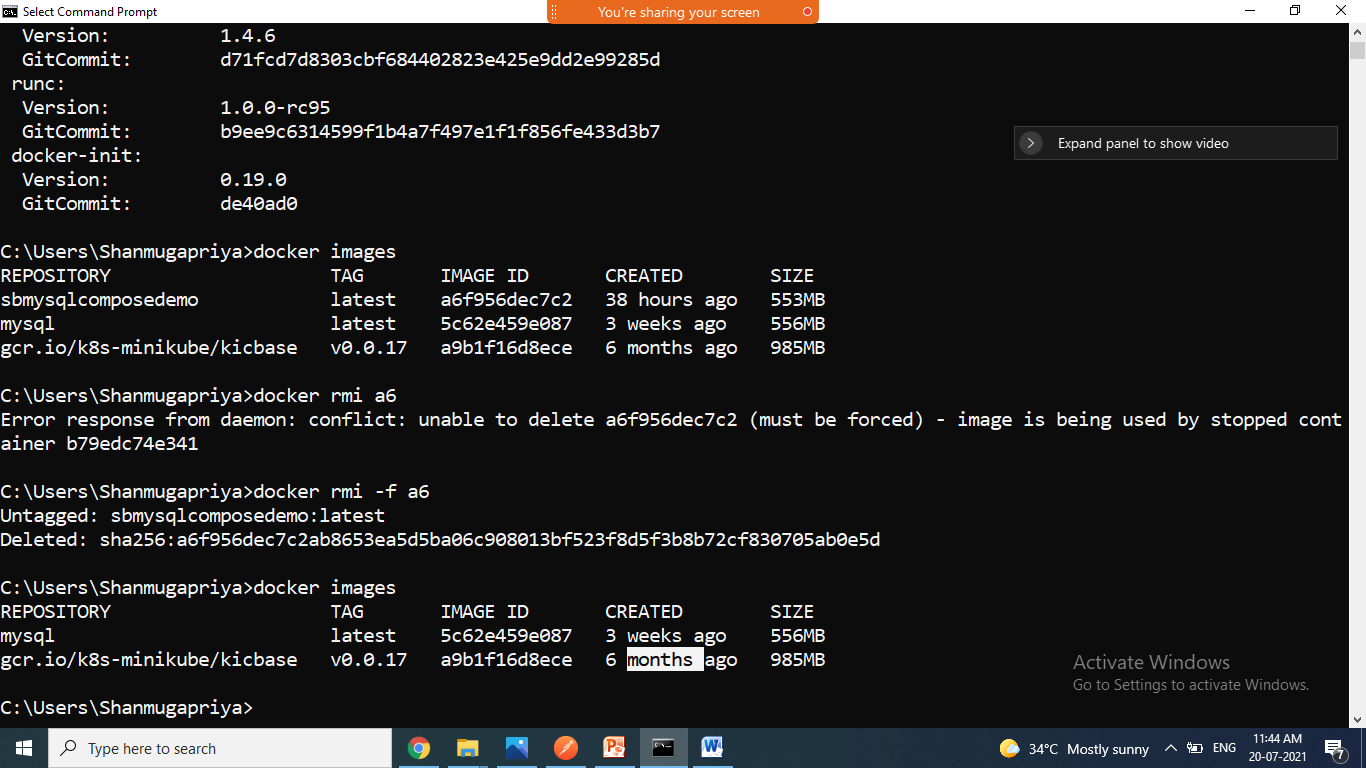
>docker images

To delete docker images

>docker rmi name/ID

To delete by force images

>docker rmi –f name/ID

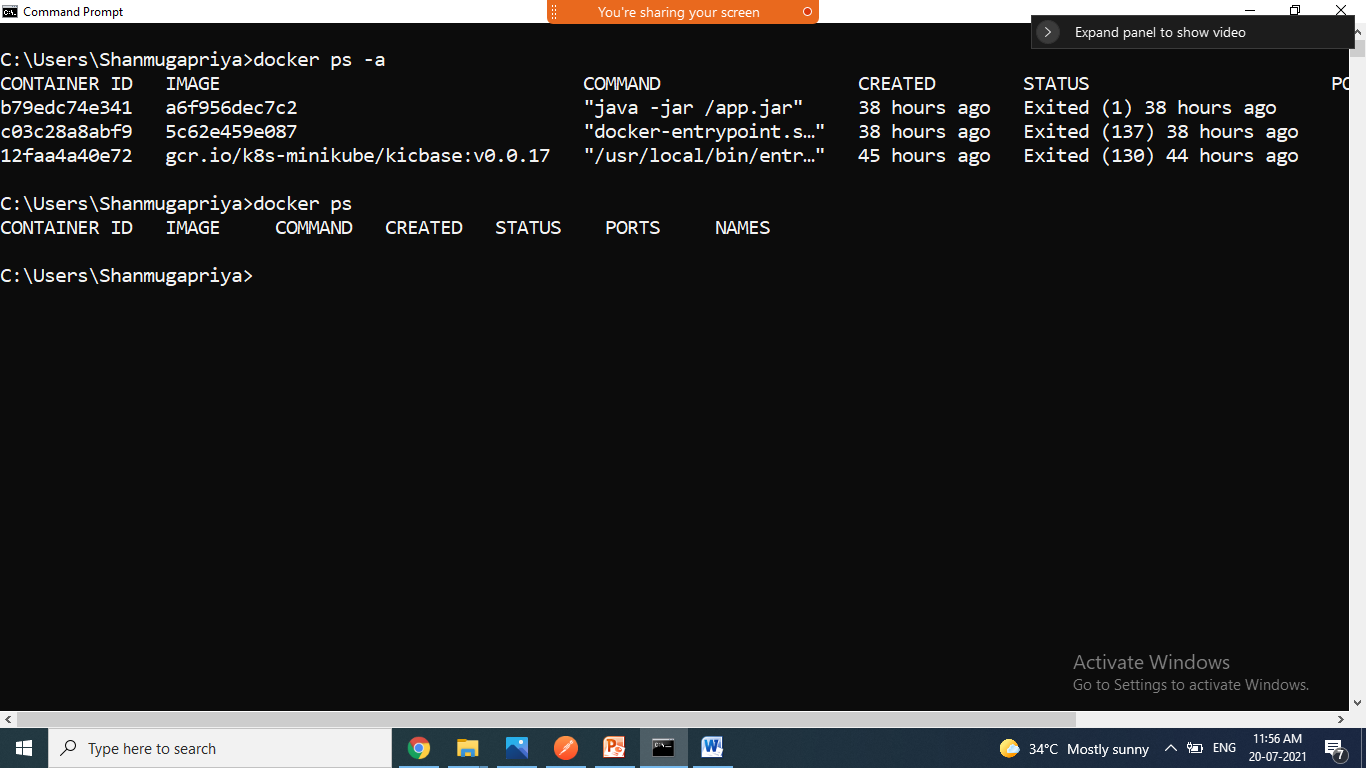


To see running containers(list of running containers)

>docker ps

To view all containers(running/not running)

>docker ps –a

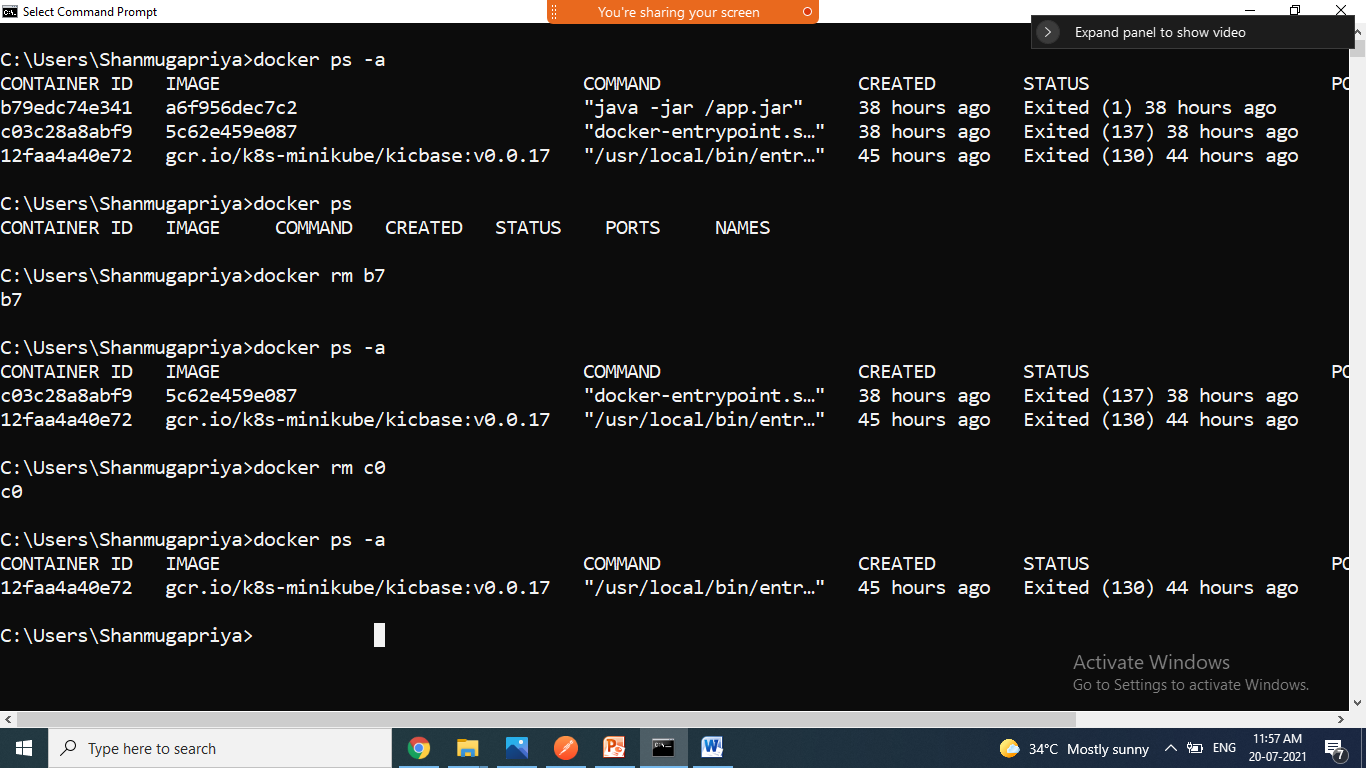


To remove the container

>docker rm name/ID

To remove running conatainer(f ->force)

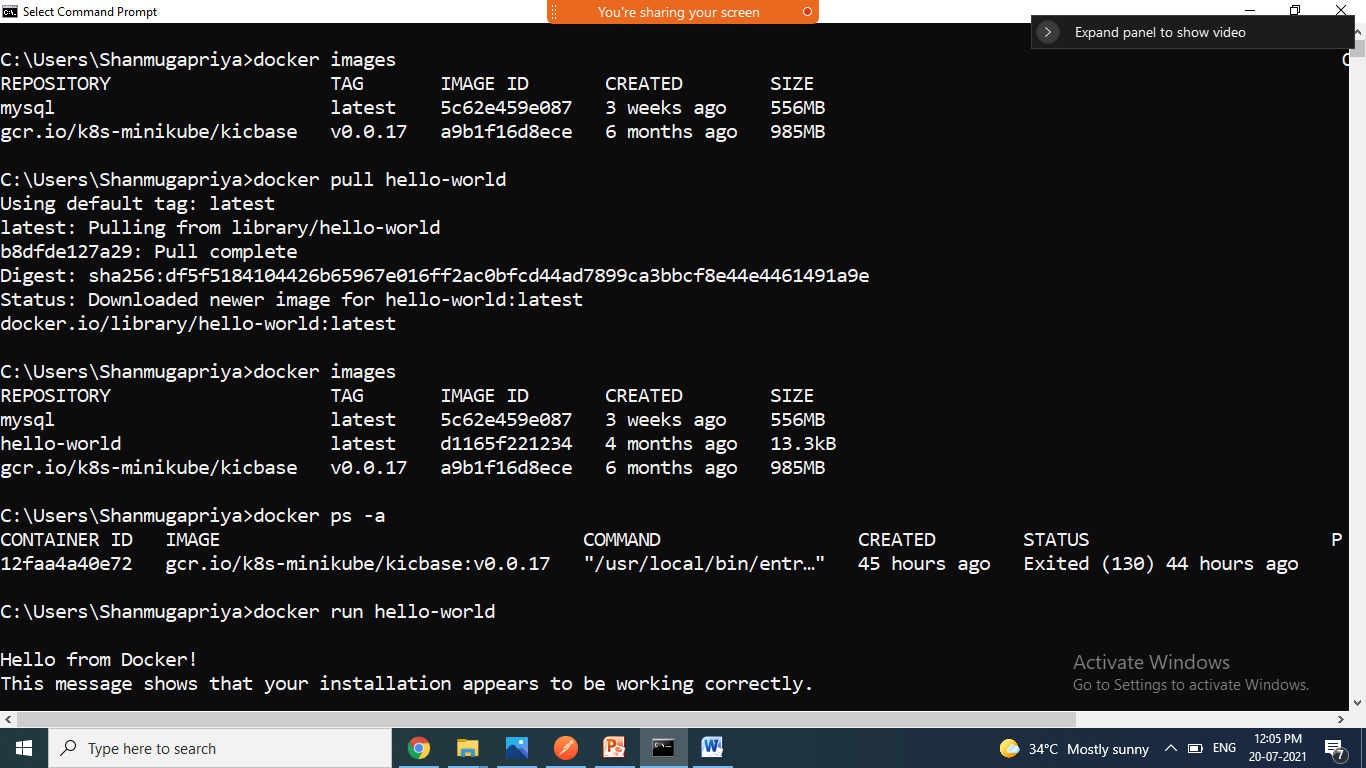
>docker rm –f name/ID

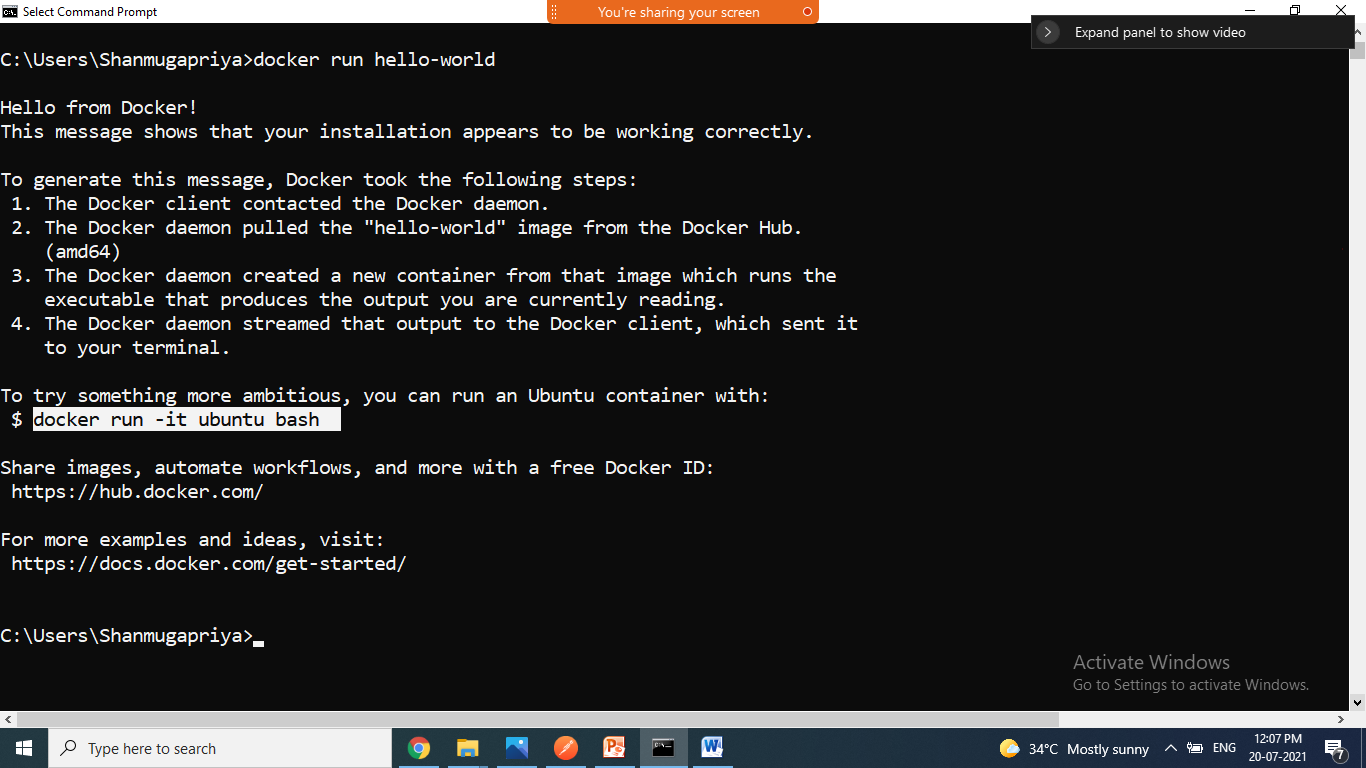


Pull hello-world from docker hub and run it

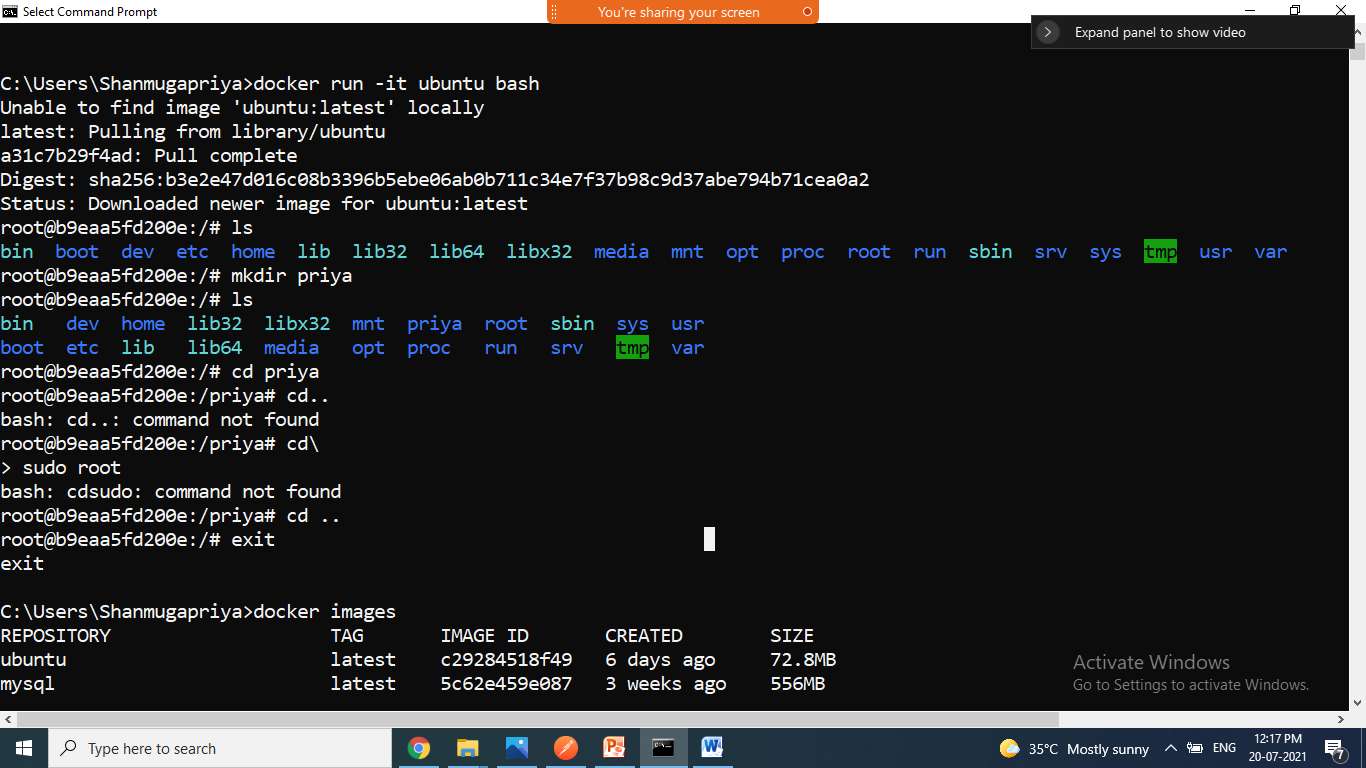
>docker pull hello-world

>docker run hello-world





Pull ubuntu and run



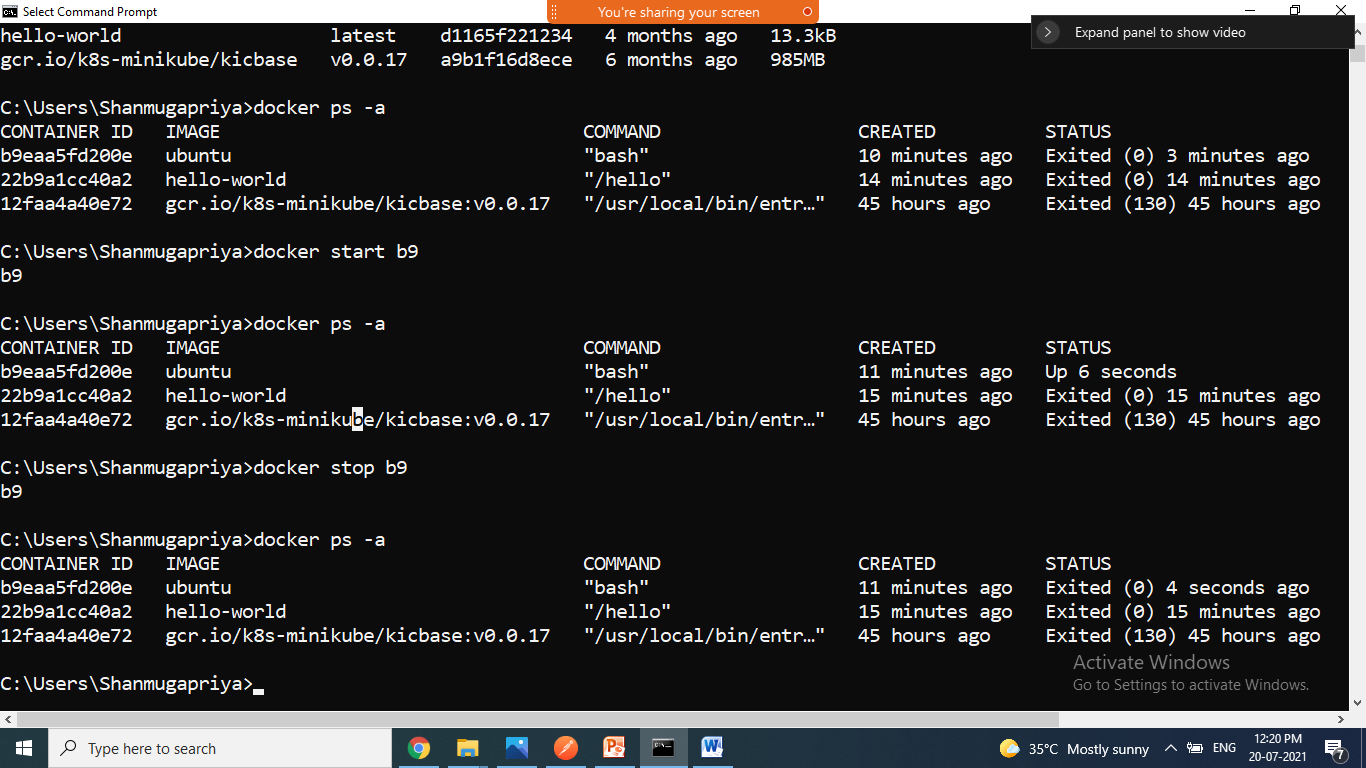
>docker pull ubuntu

>docker run Ubuntu

To start and stop the containers

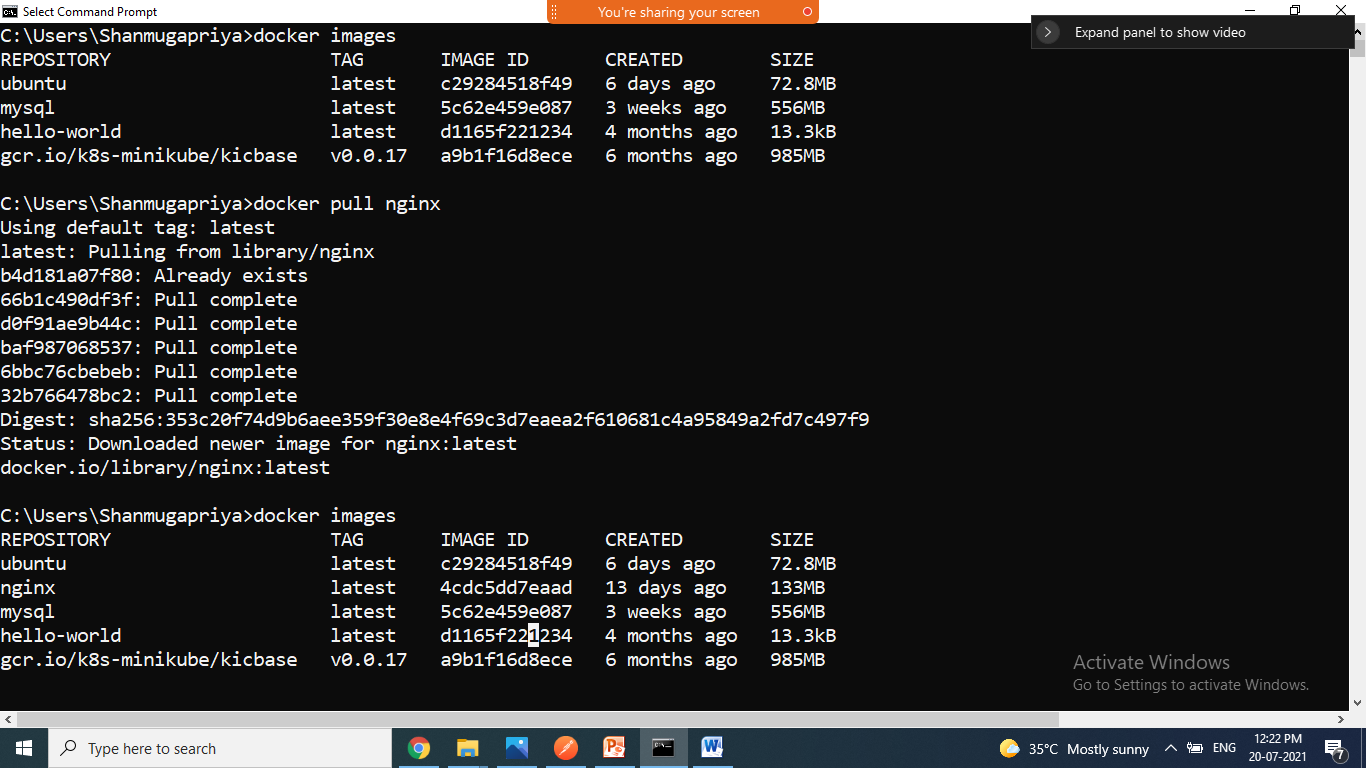
>docker start CID

>docker stop CID



Pulling nginx webserver and run it

>docker pull nginx



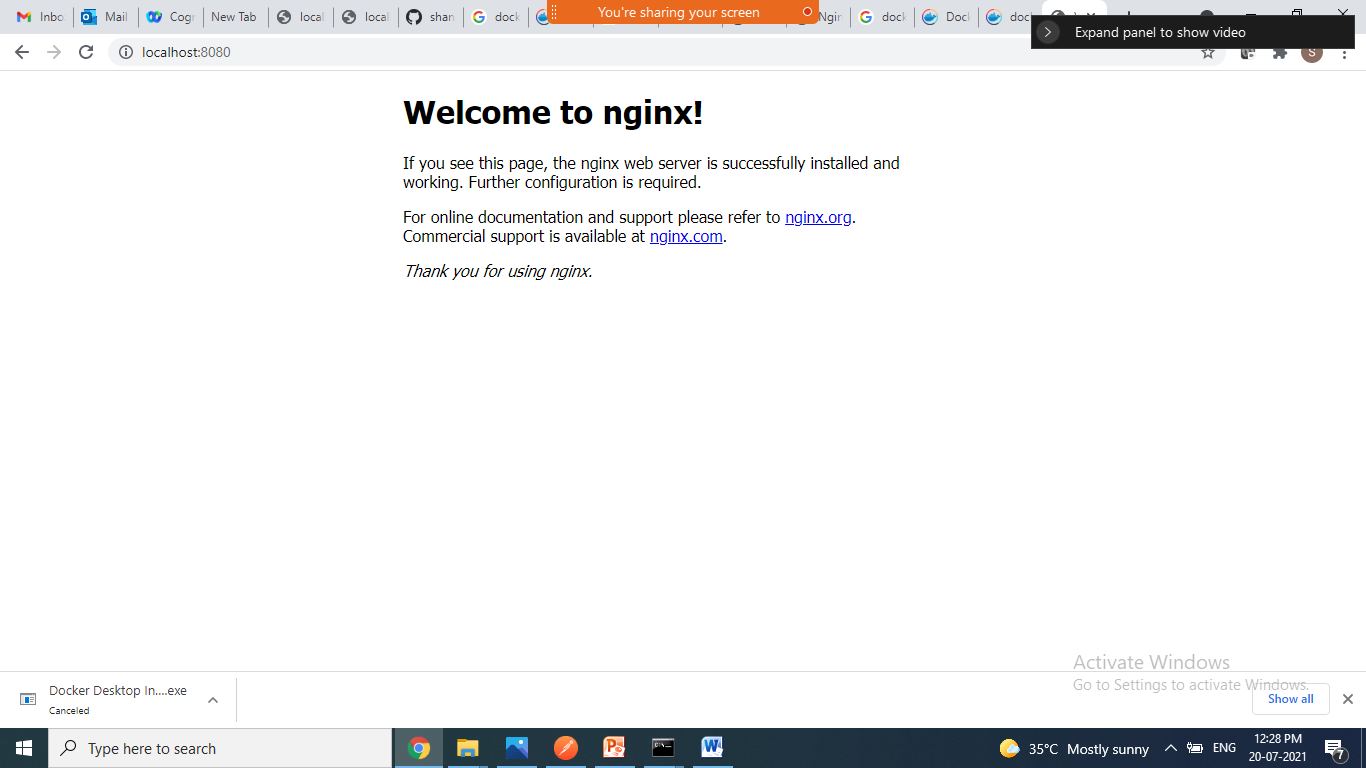
Command to run the nginx

>docker run --name my-nginx -p 8080:80 nginx



Go to bowser check the nginx is working

<http://localhost:8080/>

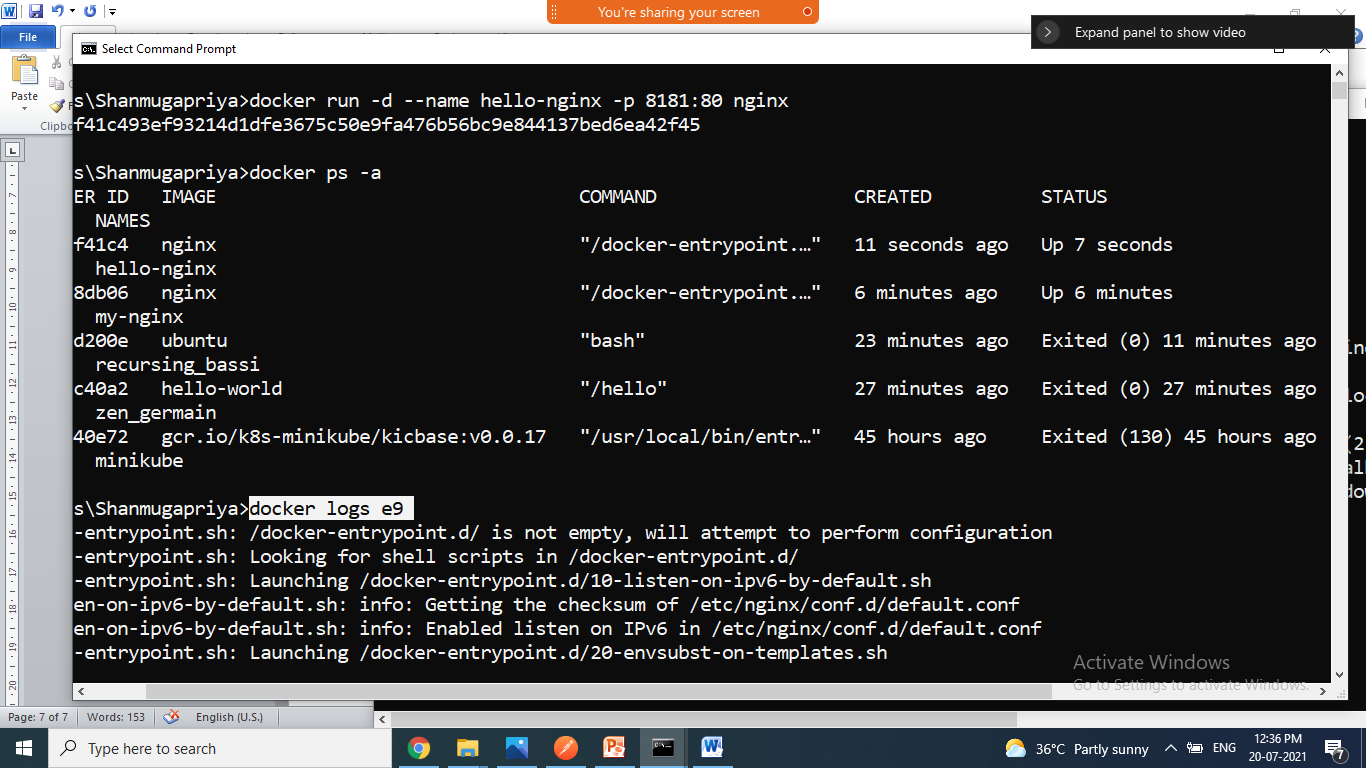


-d for detached mode

>docker run -d --name hello-nginx -p 8181:80 nginx

To view the logs

>docker logs CID/name



----------------------------------------------------------------------------

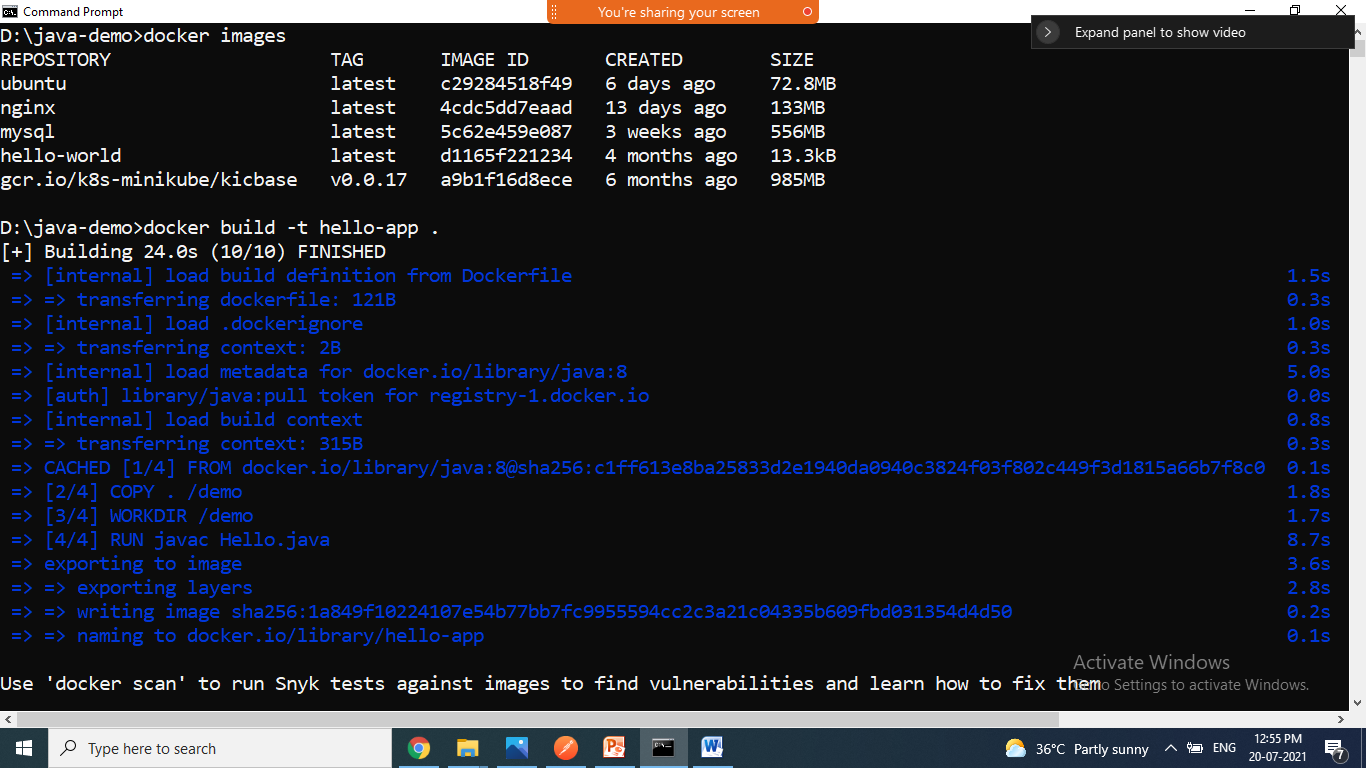
* create java app image and run it
* create one folder , create one Hello.java and Docker file

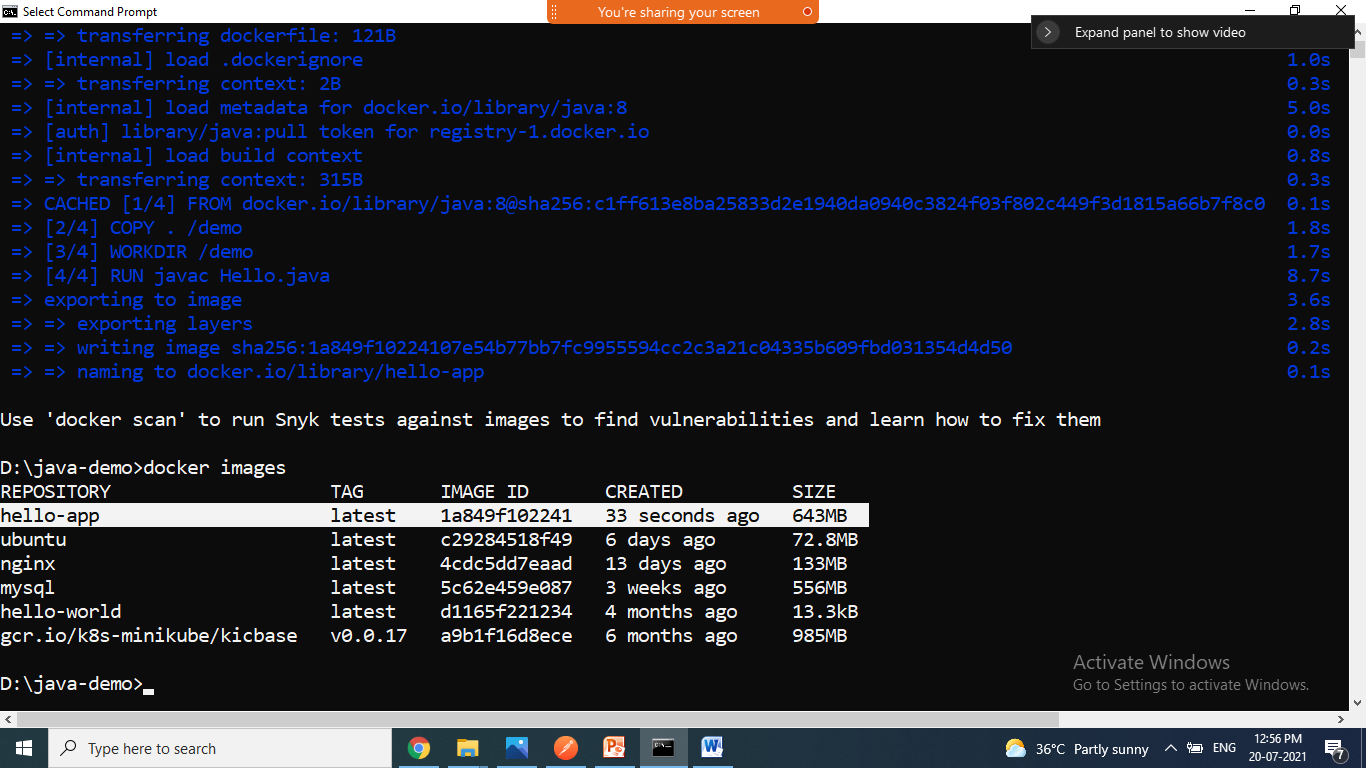
Command to create the docker image

Go to folder dir type below cmd

>docker build –t name dockerfilepath

>docker build –t hello-app .

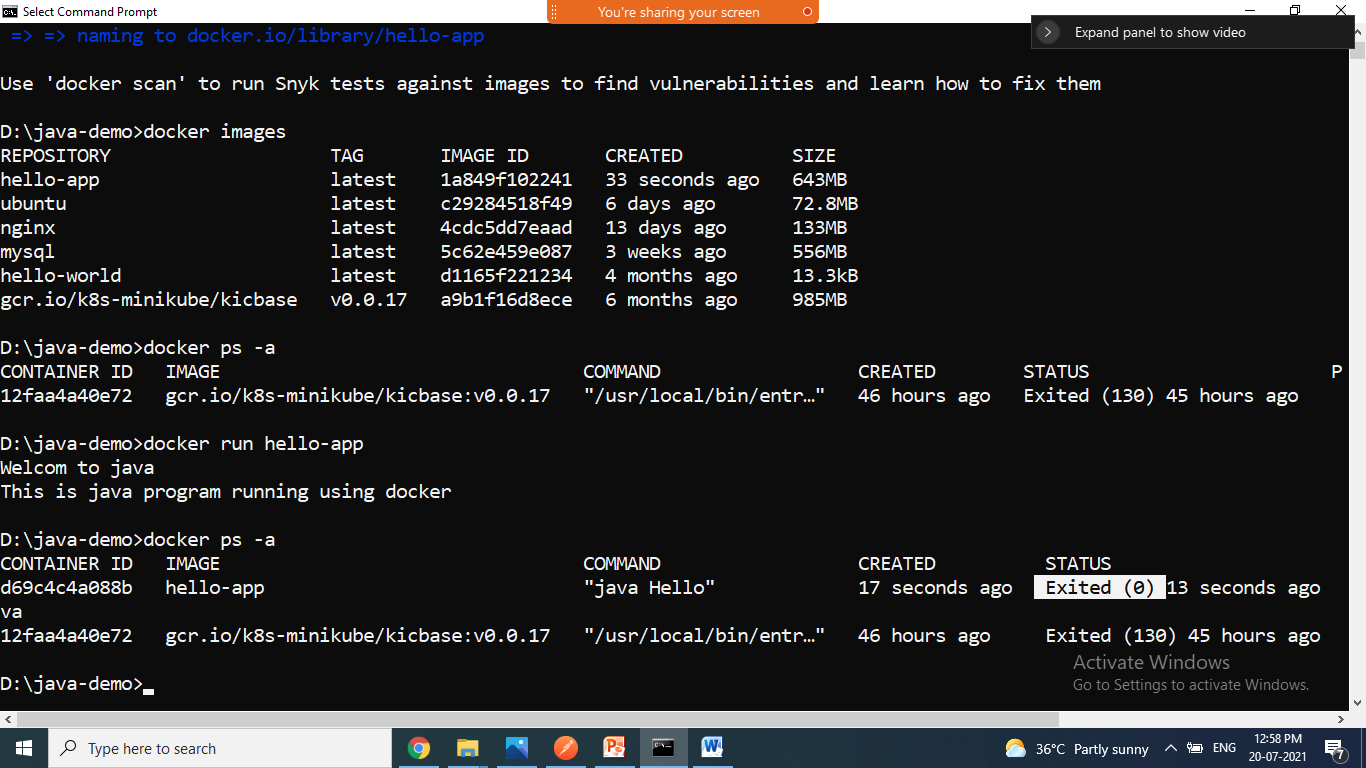




Command to run docker images

>docker run name/ID

>docker run hello-app



=================================================

Spring Application creating image and running it locally and push the to docker hub and pull from docker hub

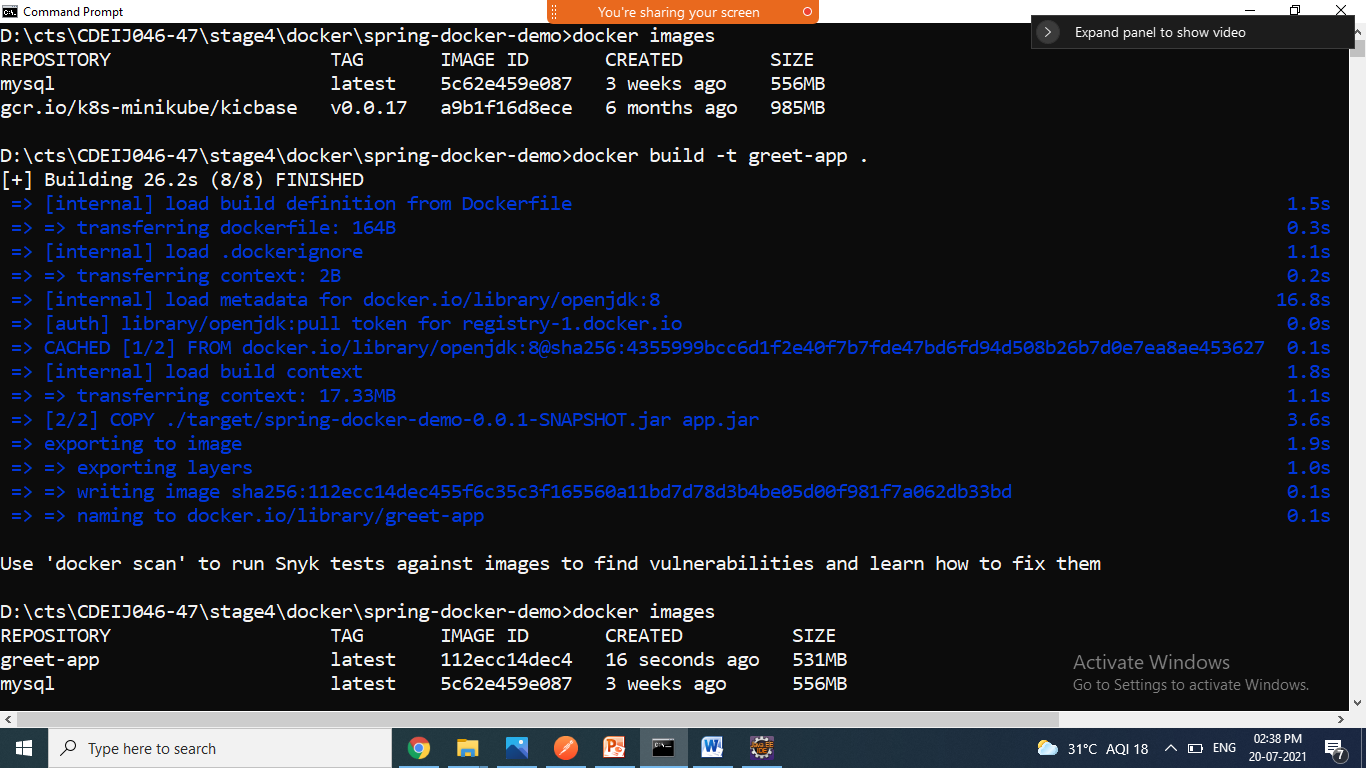
1. create spring starter project with web dependency
2. right click run as => maven build => goals type clean package/install => apply => run => jar file created in target folder
3. create Dockerfile in project folder
4. goto cmd of project folder dir
5. create docker image

>docker build –t name path

>docker build –t greet-app .

-t =>image tag name

Successfully image created.



1. Run the image command is:

-d =>detached

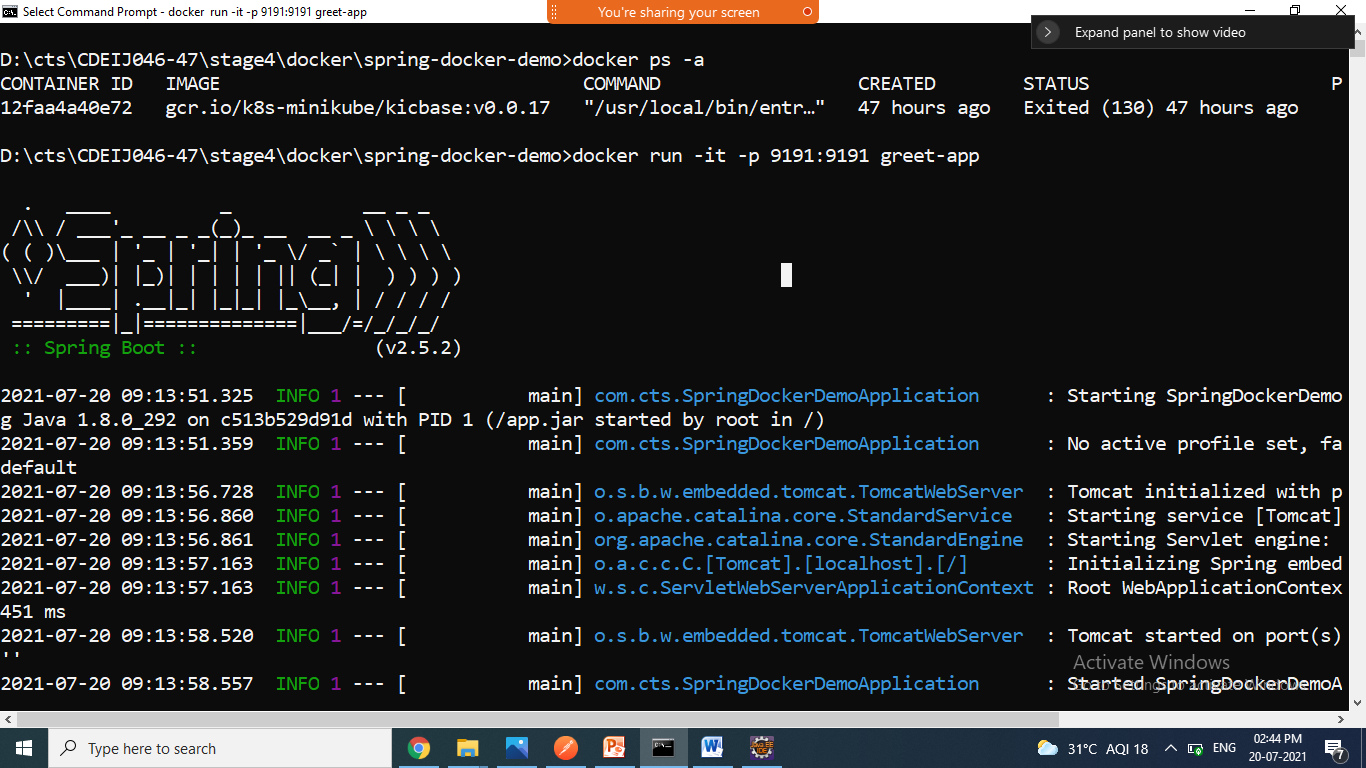
-it => interactive mode

-p =>port number 9090:80

--name => name for container

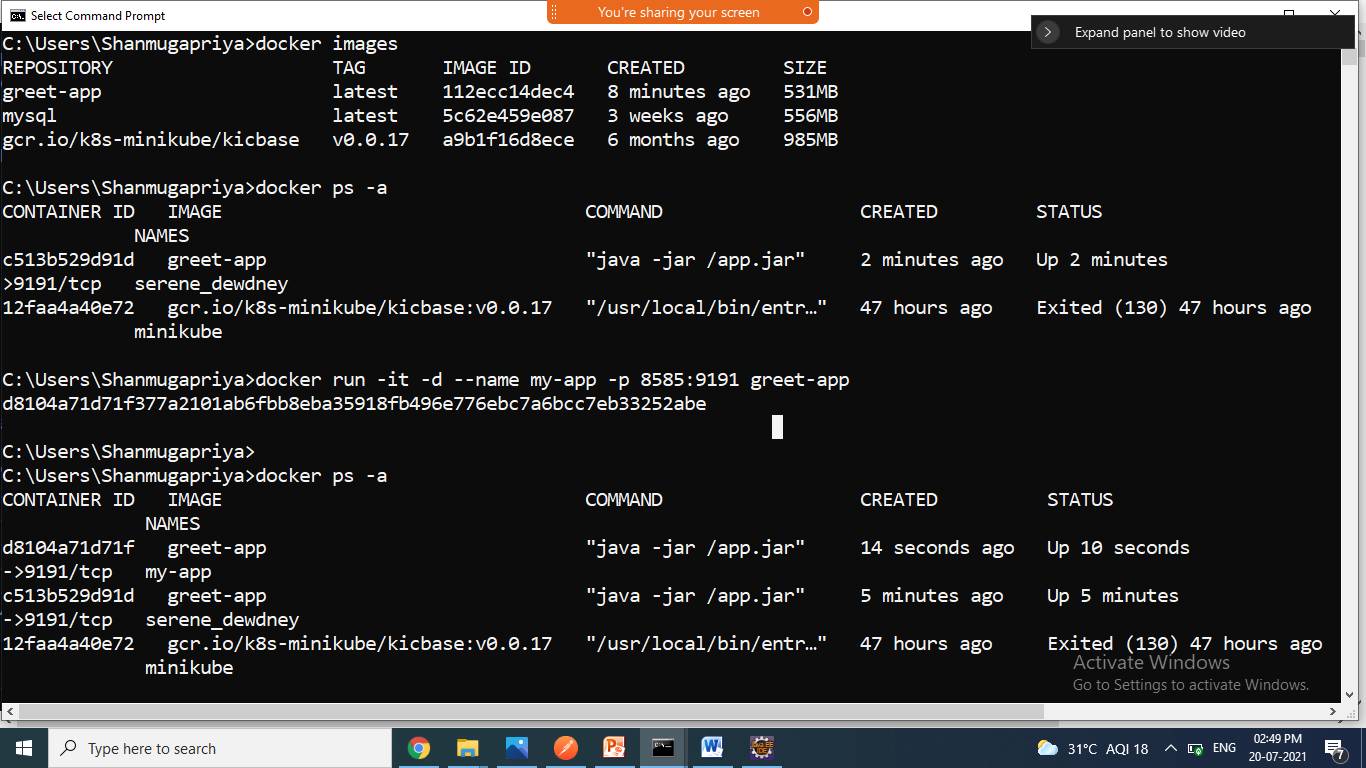
>docker run –it –p 9191:9191 imagename

>docker run –it –p 9191:9191 greet-app



1. Goto browser check it => <http://localhost:9191/>
2. Run the same with different port number

> docker run -it -d --name my-app -p 8585:9191 greet-app



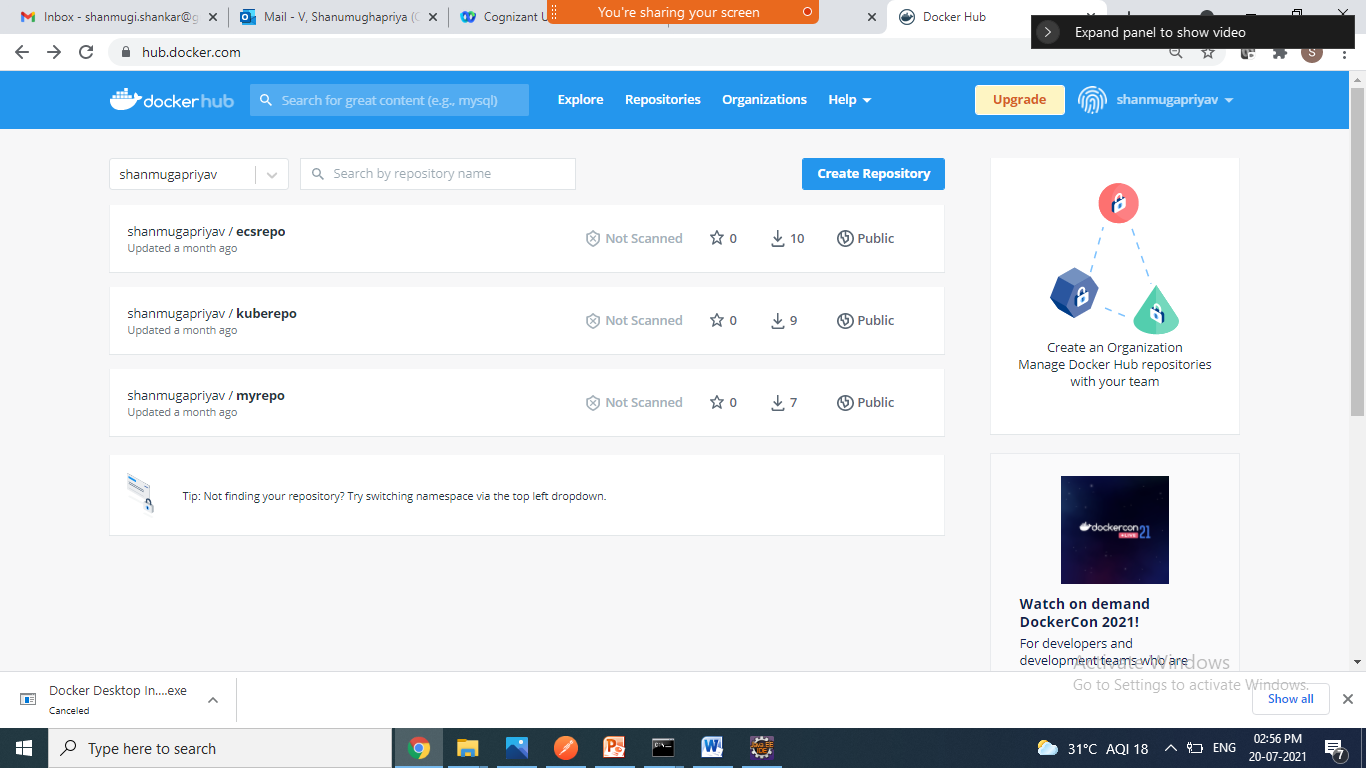
1. Stop/remove running containers
2. >docker login

userid:

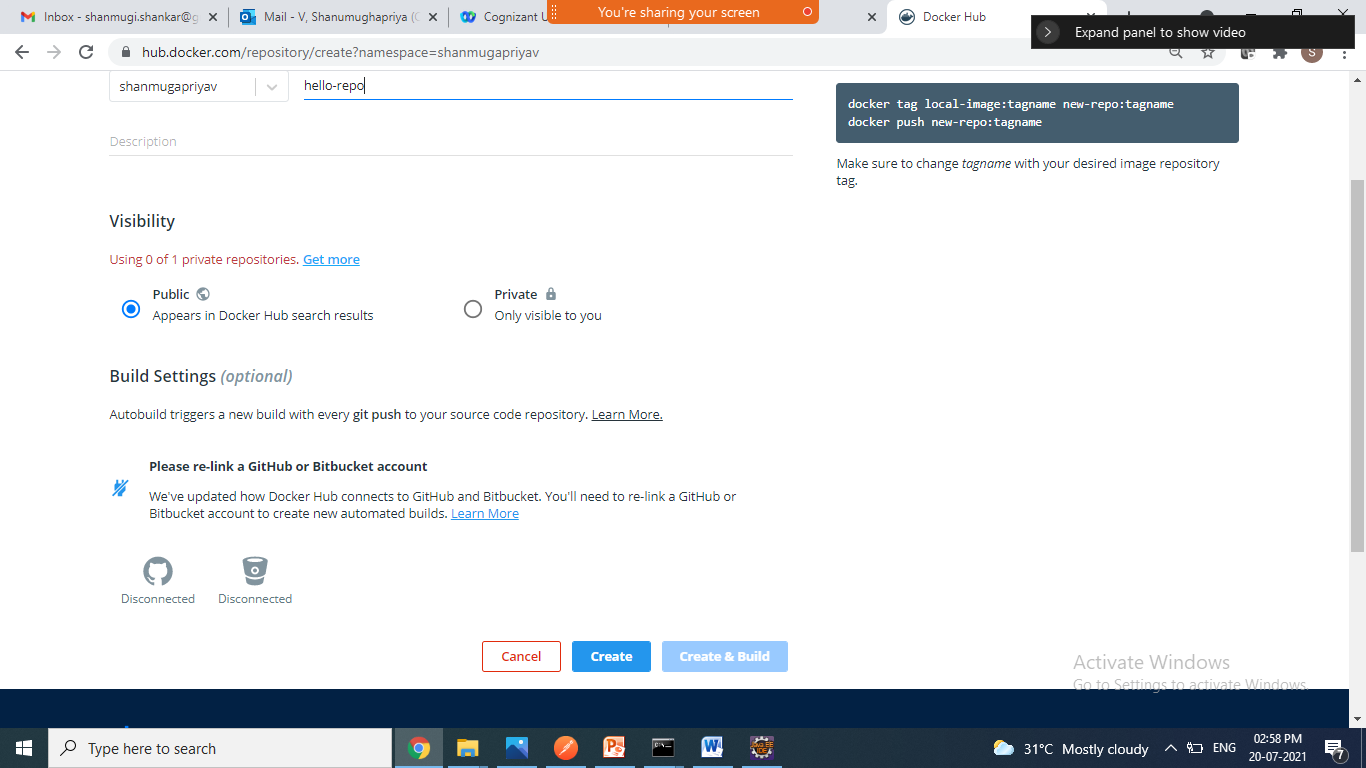
password:

login succeeded

1. Go to browser login docker hub
2. Create repository by clicking create repository button

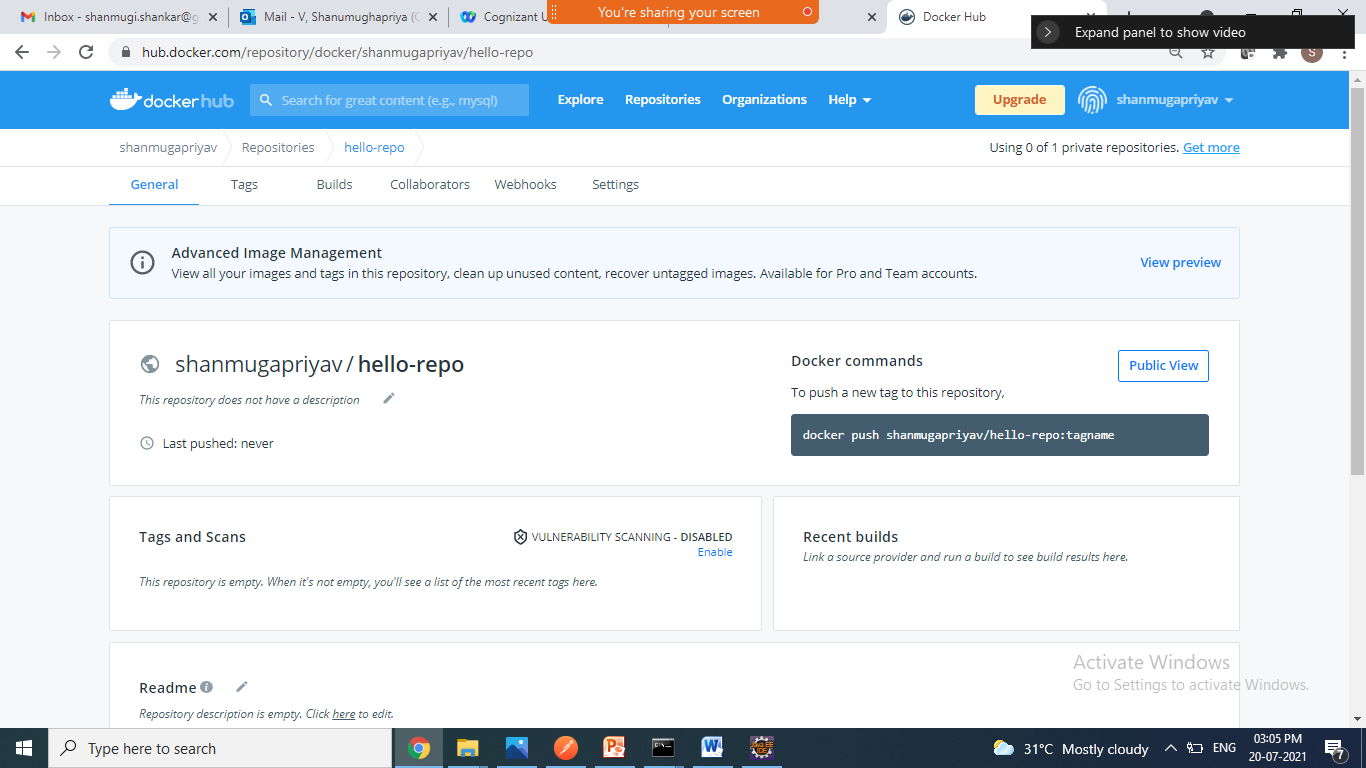


1. Give the name and click create button



docker tag local-image:tagname new-repo:tagname

docker push new-repo:tagname



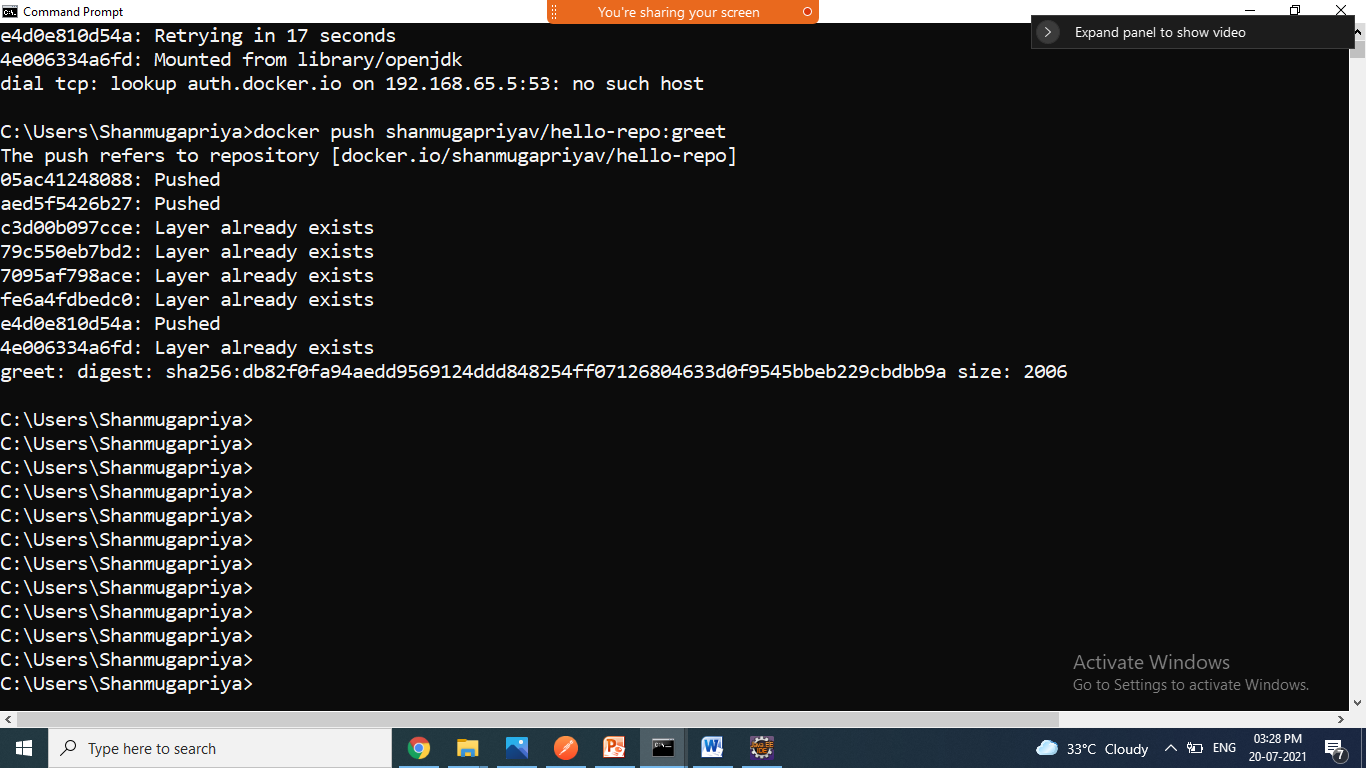
1. Push command is

docker push shanmugapriyav/hello-repo:tagname

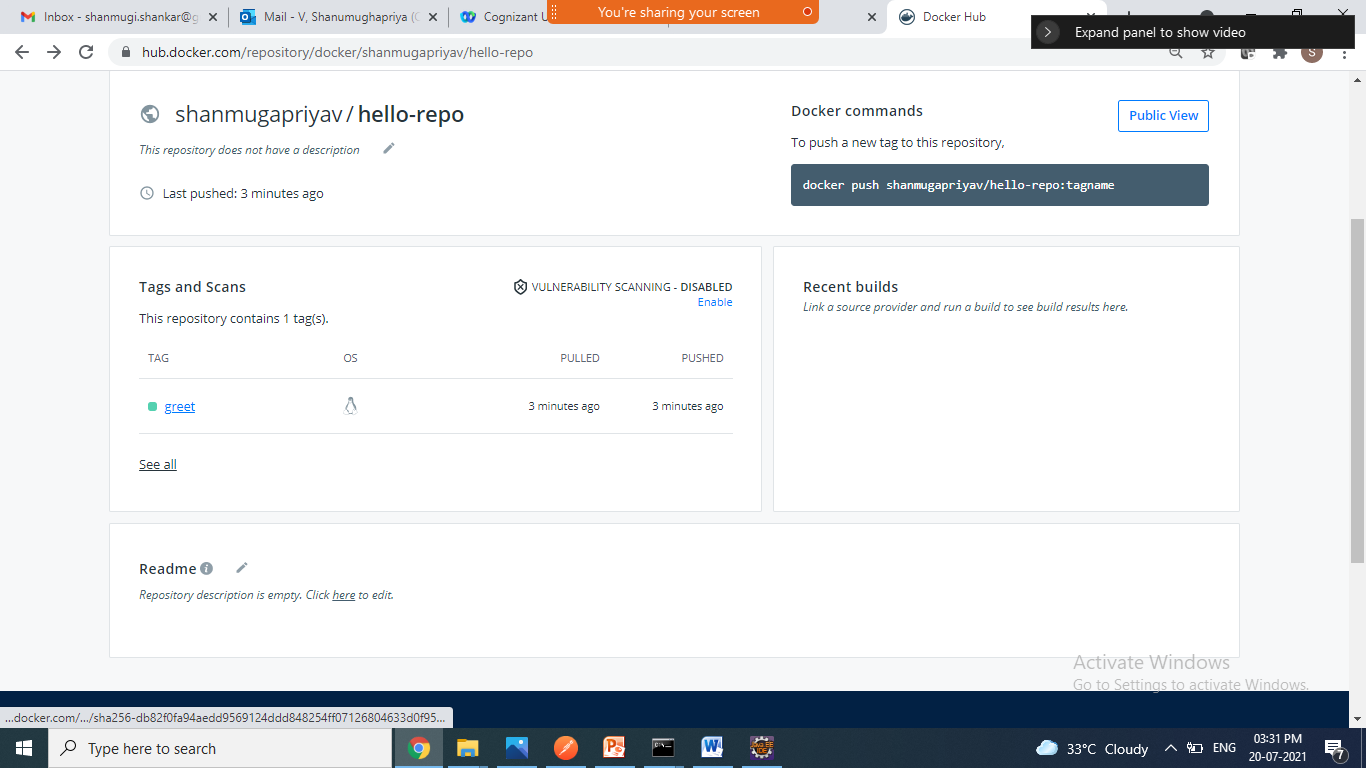
1. Goto command prompt type below commands

> docker tag greet-app shanmugapriyav/hello-repo:greet

>docker push shanmugapriyav/hello-repo:greet

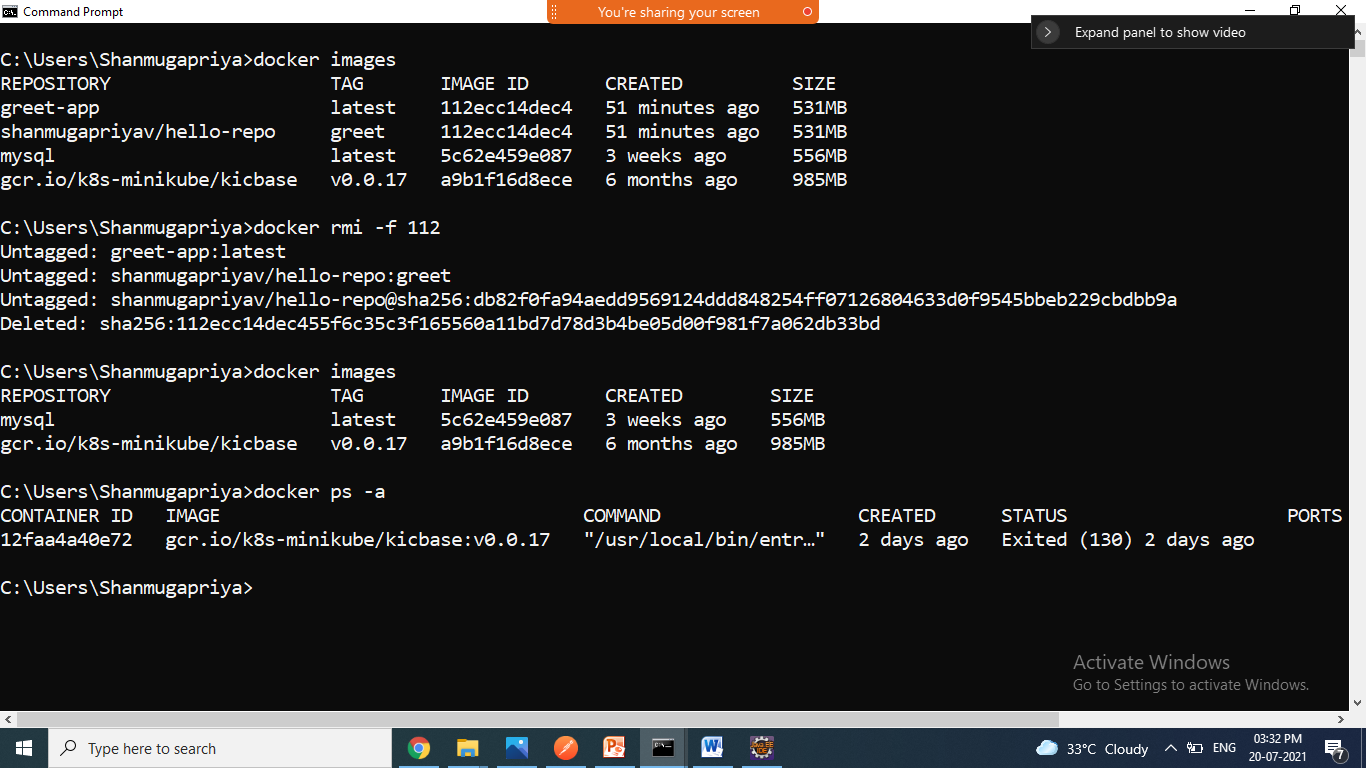


1. Goto docker hub check it, goto hello-repo repository greet image pushed



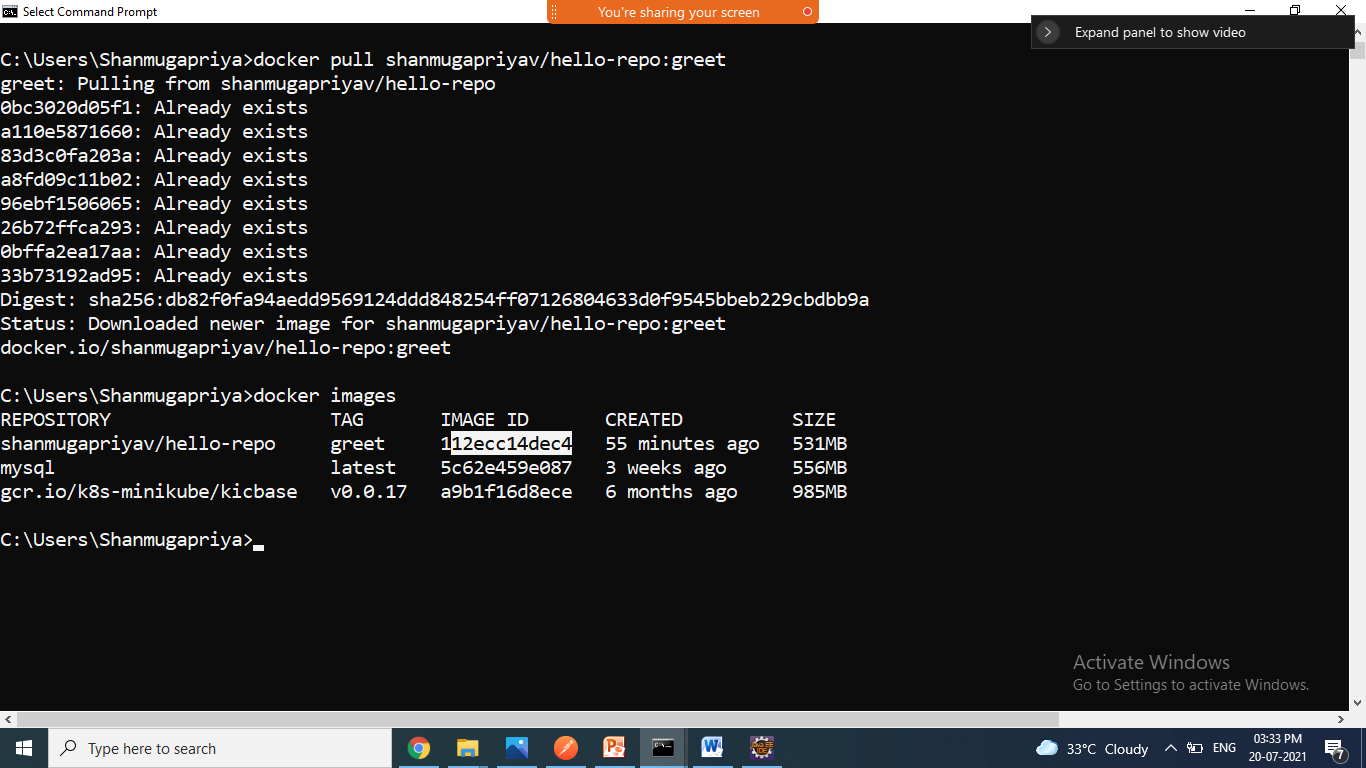
1. Goto cmd, remove the existing greet-app from local

>docker rmi ID



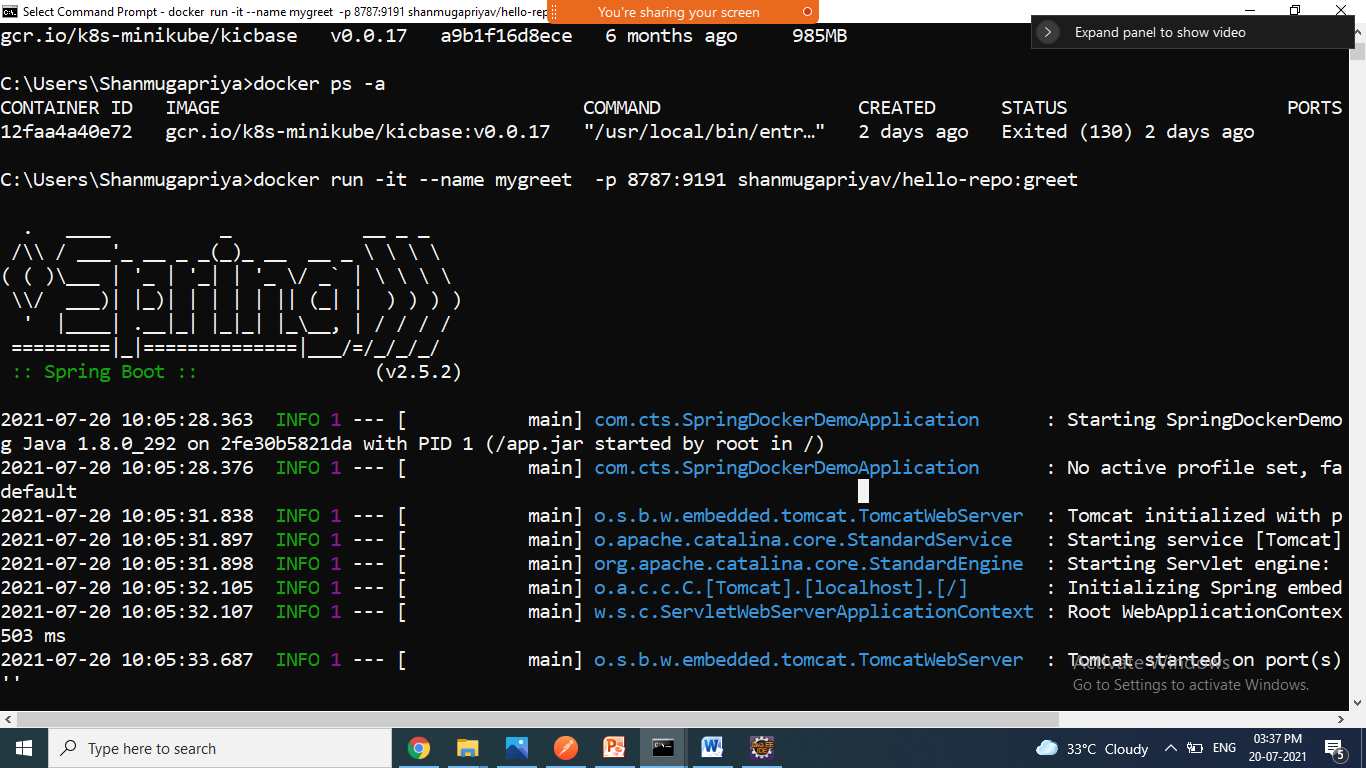
1. Use docker pull command pull the greet image from docker hub and run it here

> docker pull shanmugapriyav/hello-repo:greet



>docker run –it --name mygreet -p 8787:9191 shanmugapriyav/hello-repo:greet

> docker run –it --name mygreet -p 8787:9191 ID

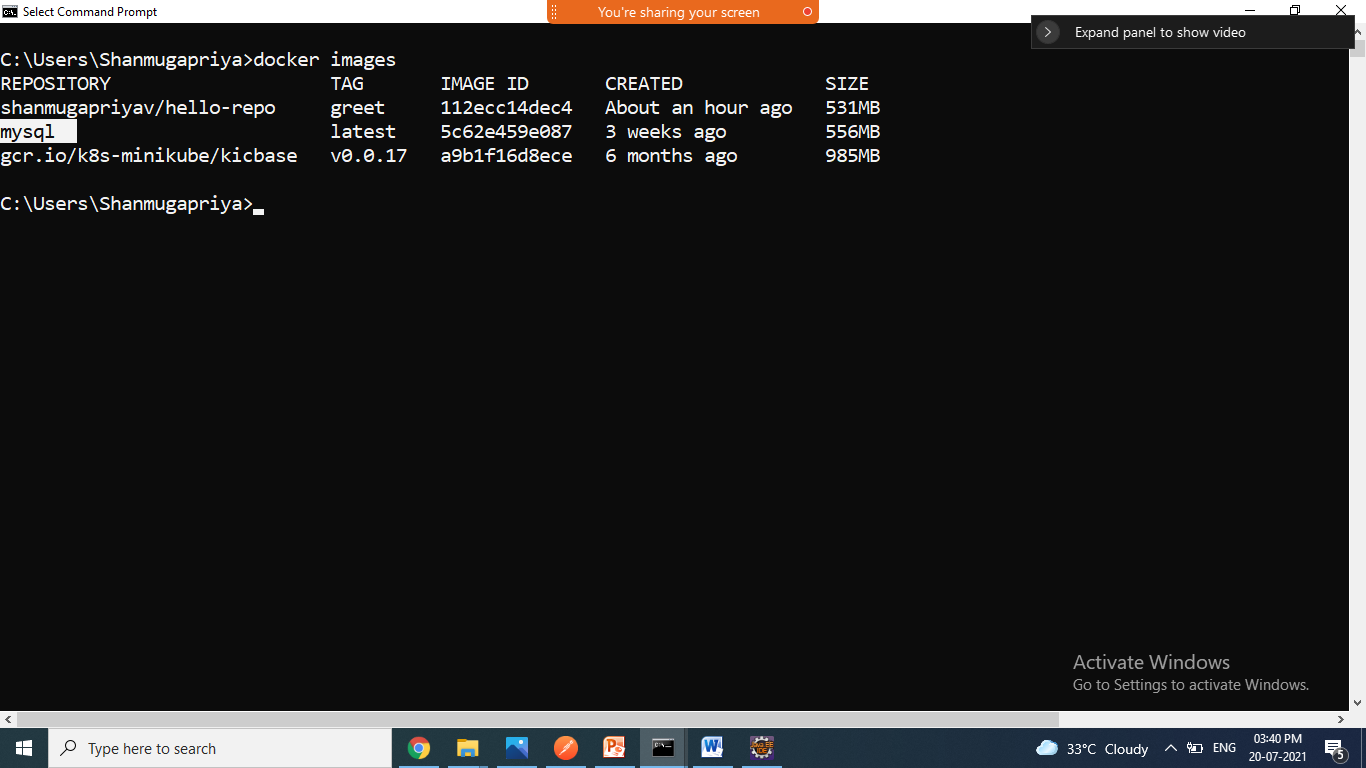


1. Go to browser <http://localhost:8787/>

============================================

Pulling mysql fro docker hub

>docker pull mysql

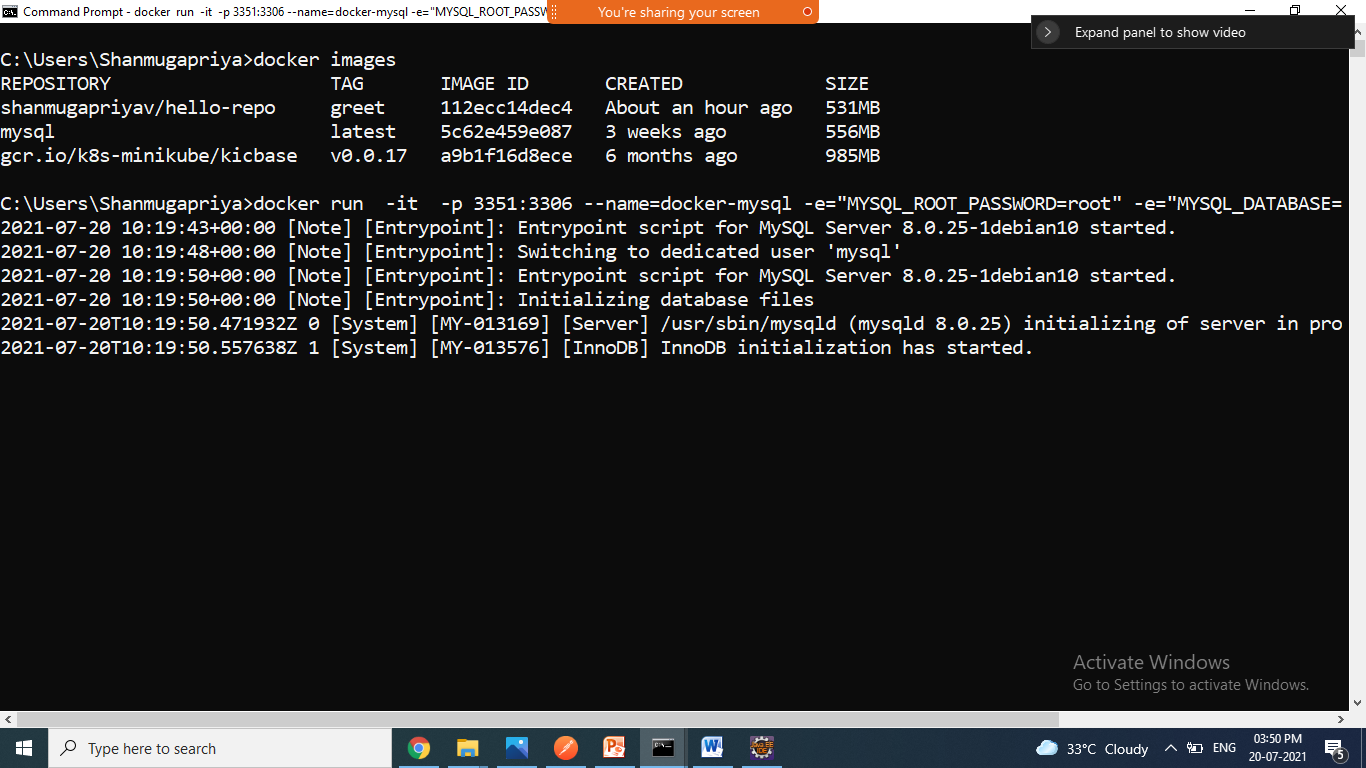


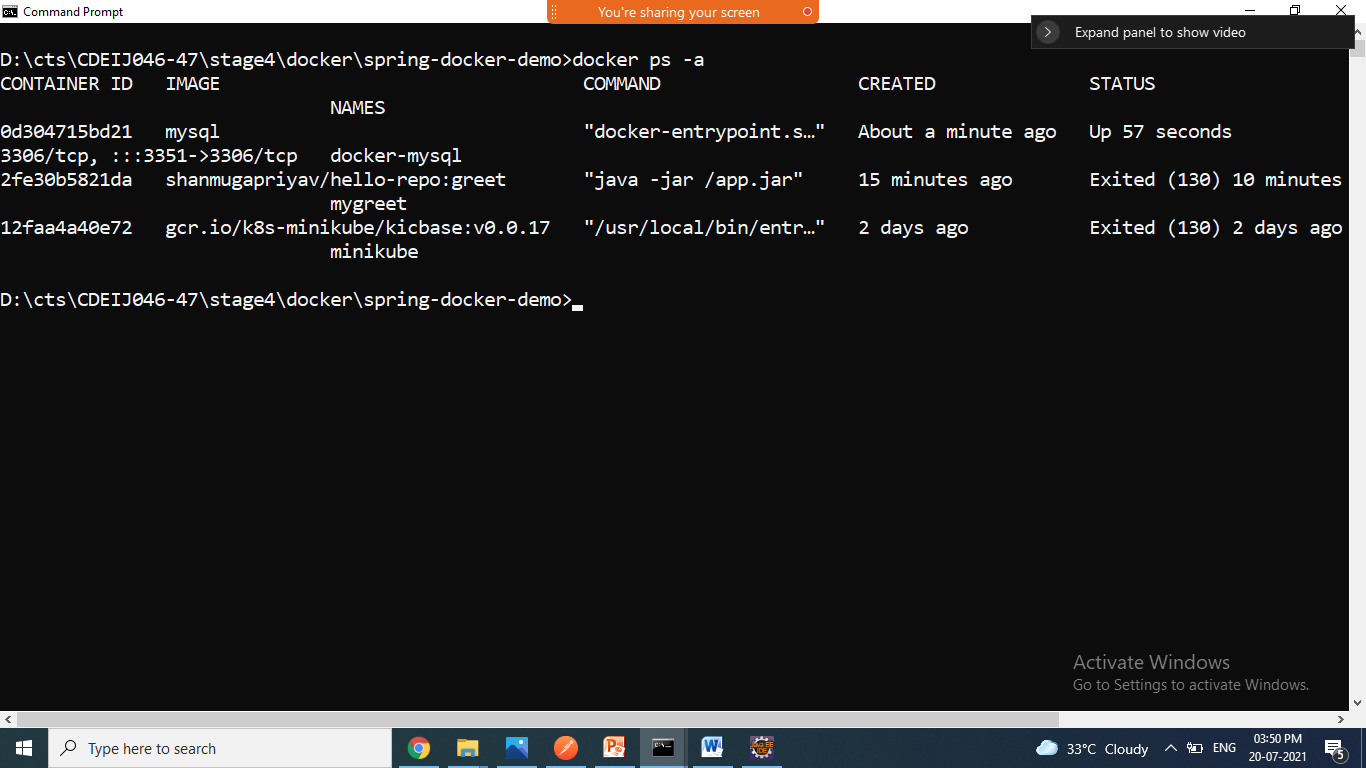
To run mysql in local machine , use below command

The following command starts another mysql container instance and runs the mysql command line client against your original mysql container, allowing you to execute SQL statements against your database instance:

$ docker run -it --network some-network --rm mysql mysql -hsome-mysql -uexample-user -p

>docker run -it -p 3351:3306 --name=docker-mysql-e=”MYSQL\_ROOT\_PASSWORD=root” –e=”MYSQL\_DATABASE=testdb” mysql





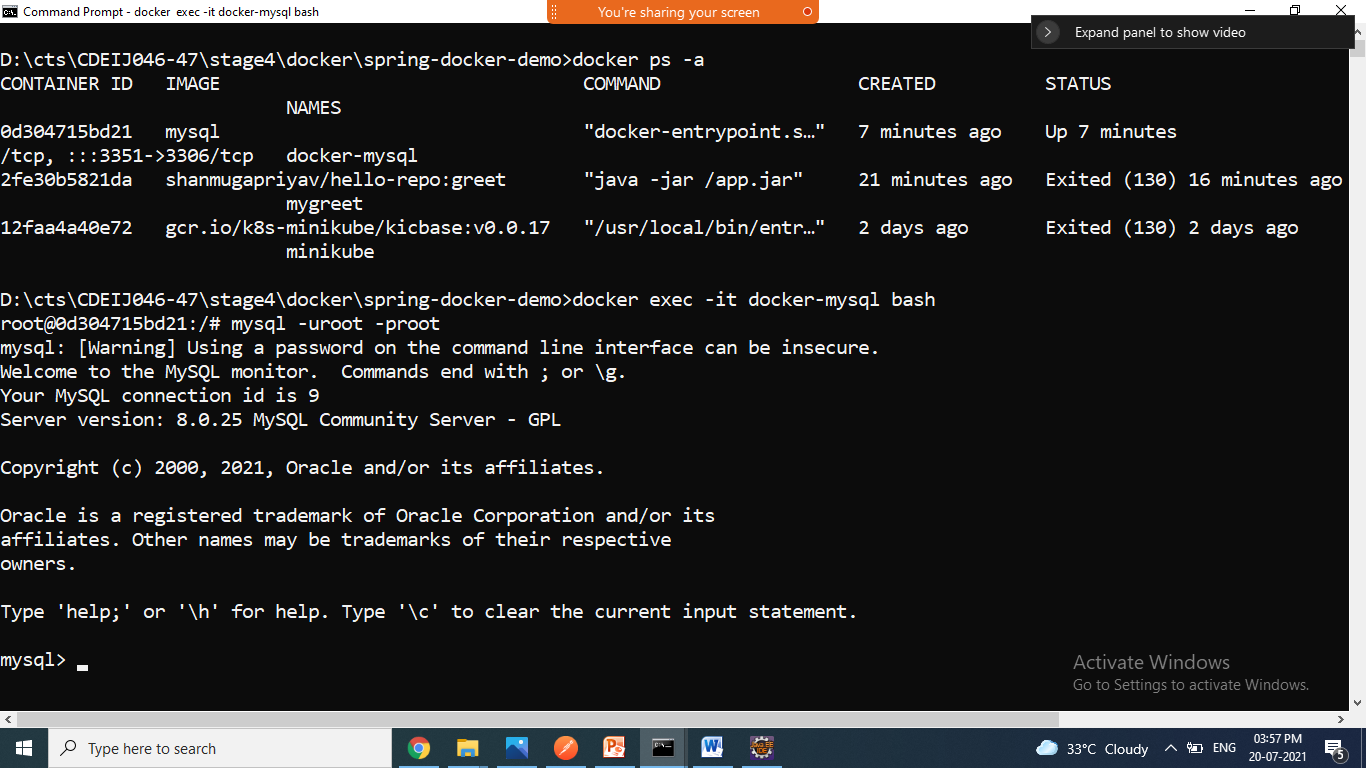
The docker exec command allows you to run commands inside a Docker container. The following command line will give you a bash shell inside your mysql container:

$ docker exec -it some-mysql bash

>docker exec –it docker-mysql bash

#mysql -uroot –proot

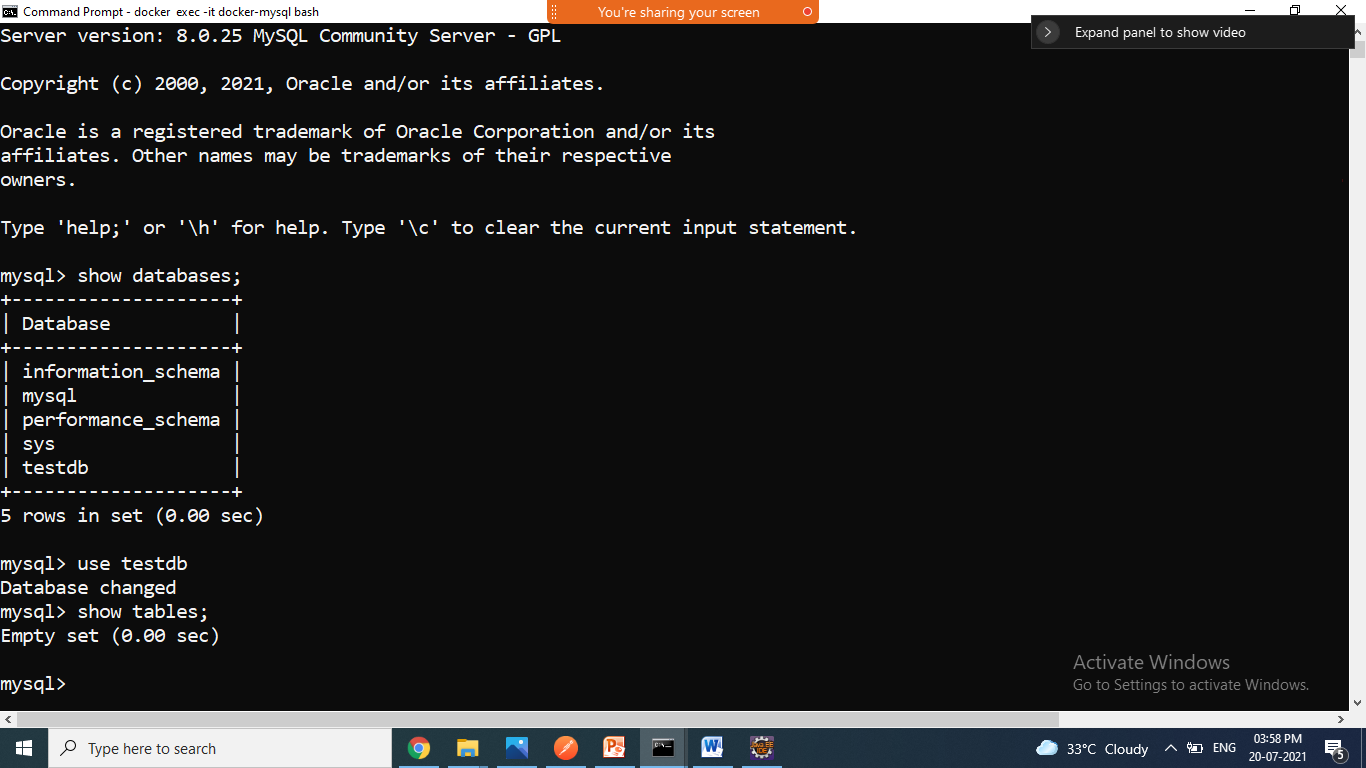
It connected MYSQL prompt



>show databases;

>use testdb

>show tables;



================================================

**SPRING+MYSQL**

To view all available network

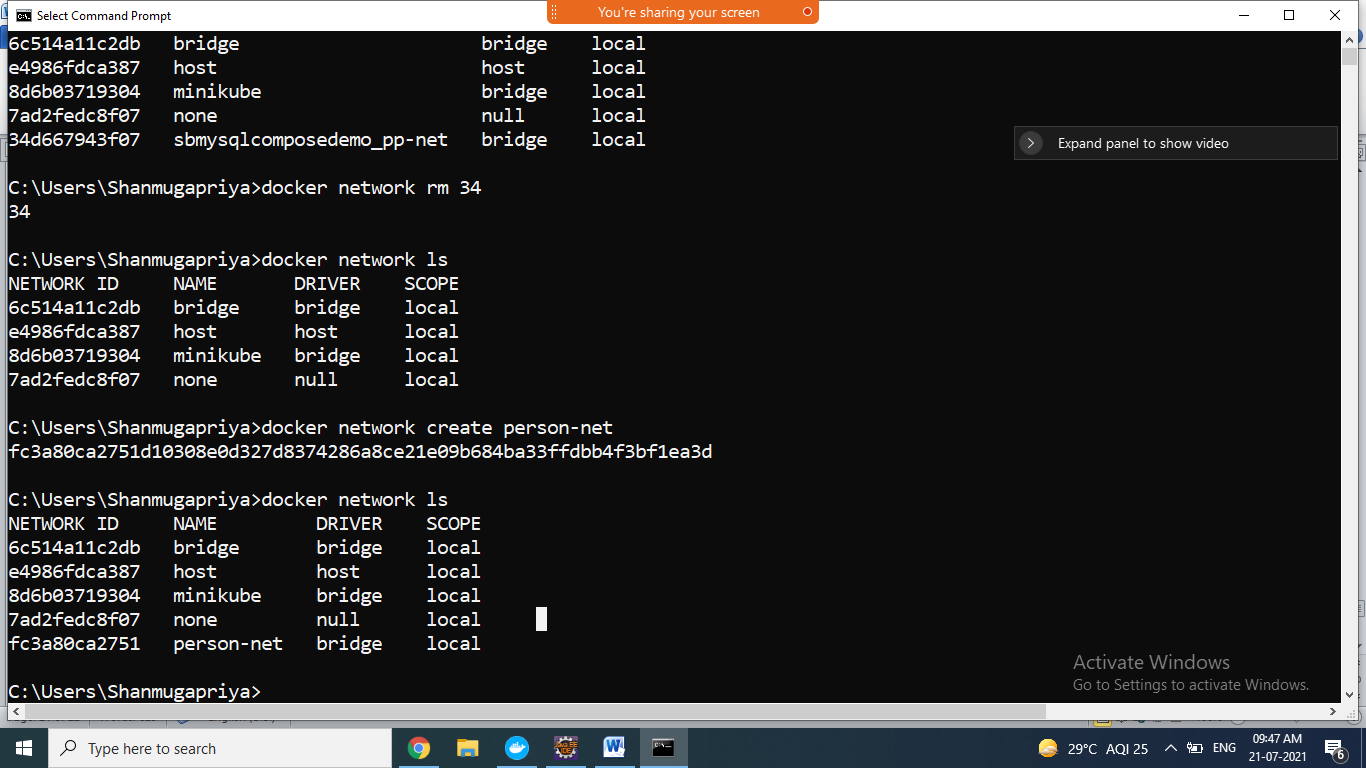
>docker network ls

To remove the network

>docker network rm name/Id

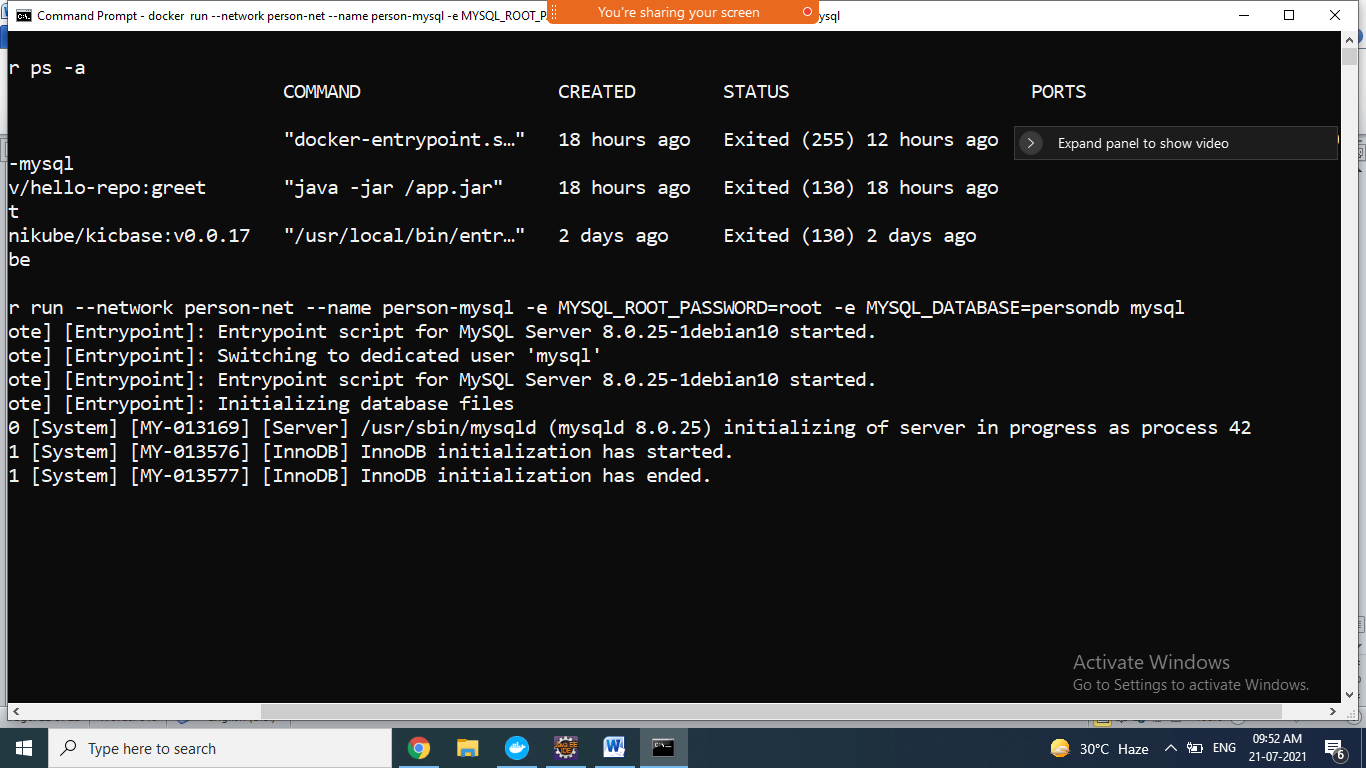
To create network

>docker network create name



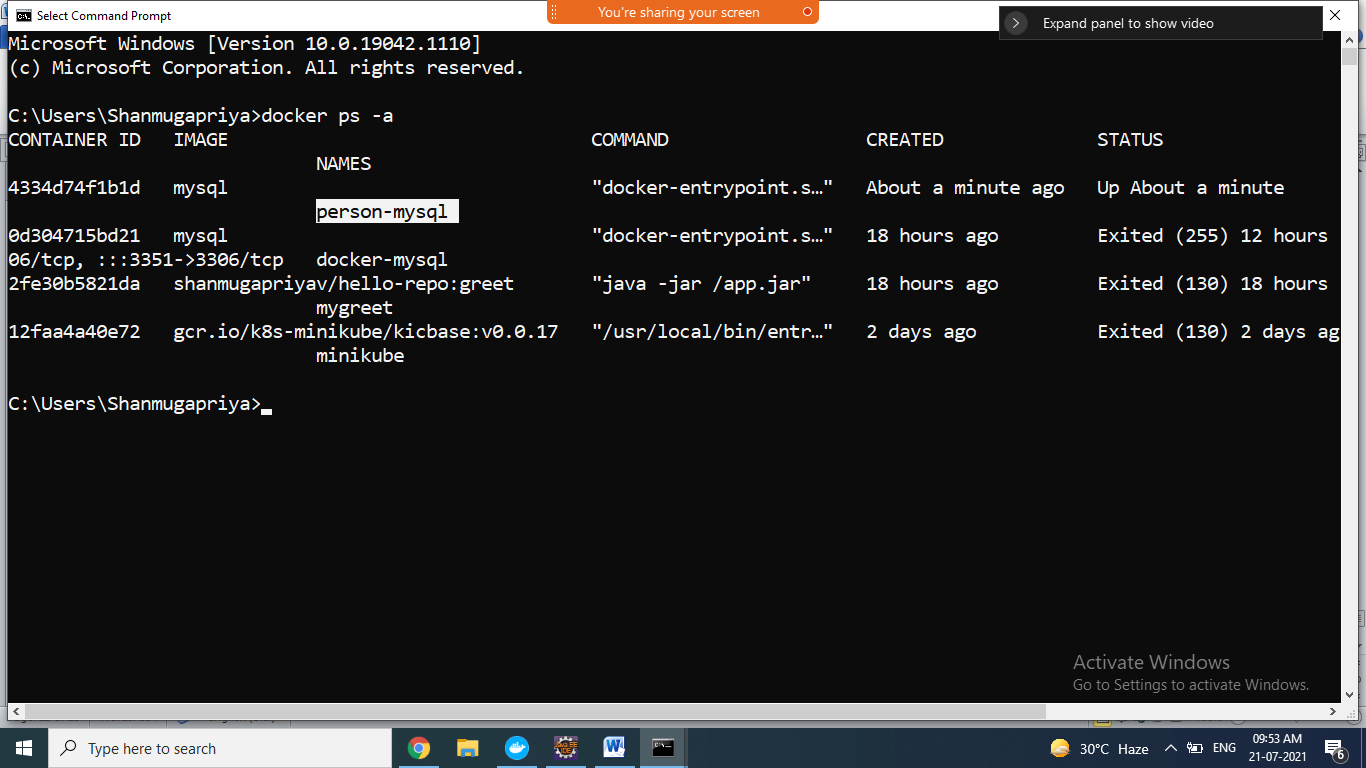
To running mysql container

>docker run --network person-net --name person-mysql -e MYSQL\_ROOT\_PASSWORD=root -e MYSQL\_DATABASE=persondb mysql



Check the mysql running status

>docker ps –a



To execute and connect to the mysql

>docker exec –it person-mysql bash

root@4334d74f1b1d:/# mysql -uroot -proot

mysql: [Warning] Using a password on the command line interface can be insecure.

ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/run/mysqld/mysqld.sock' (2)

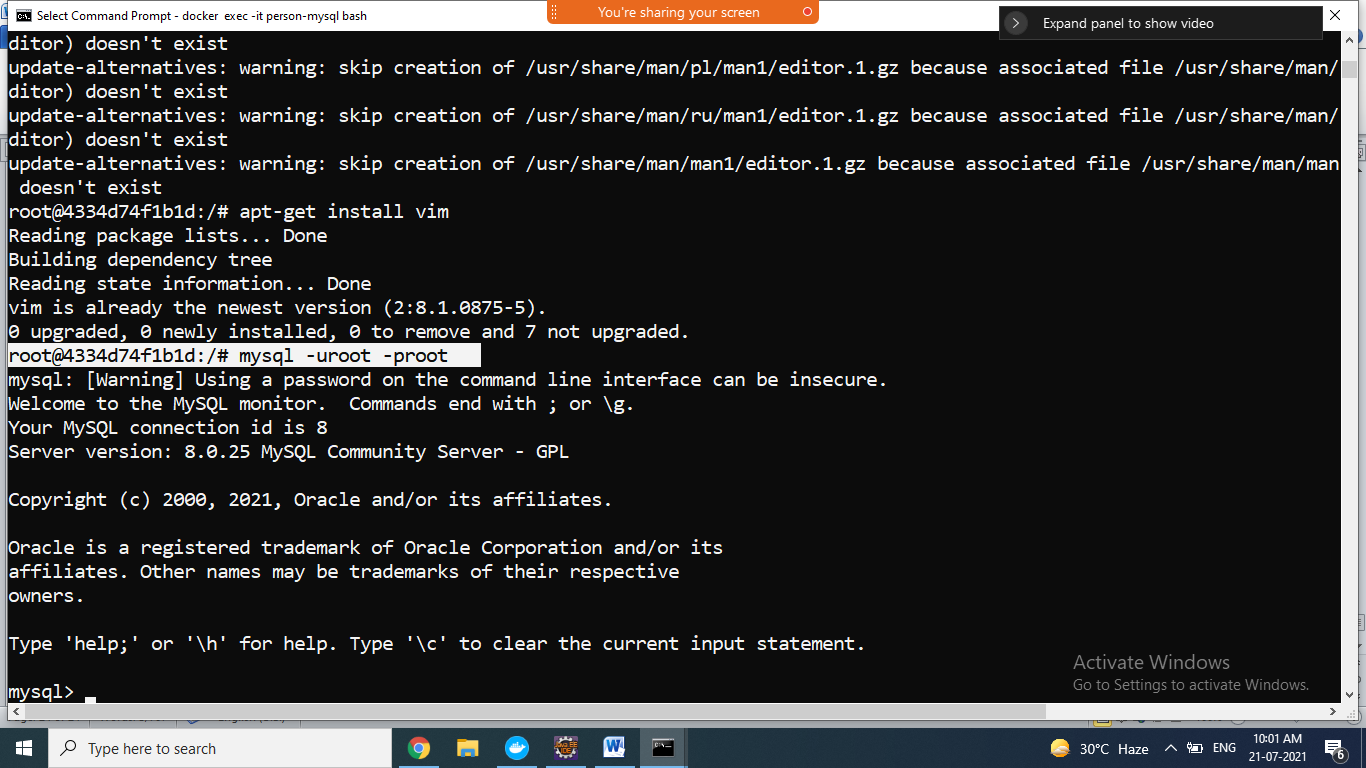
root@4334d74f1b1d:/# apt-get update

#apt-get install apt-file

#apt-file update

#apt-get install vim





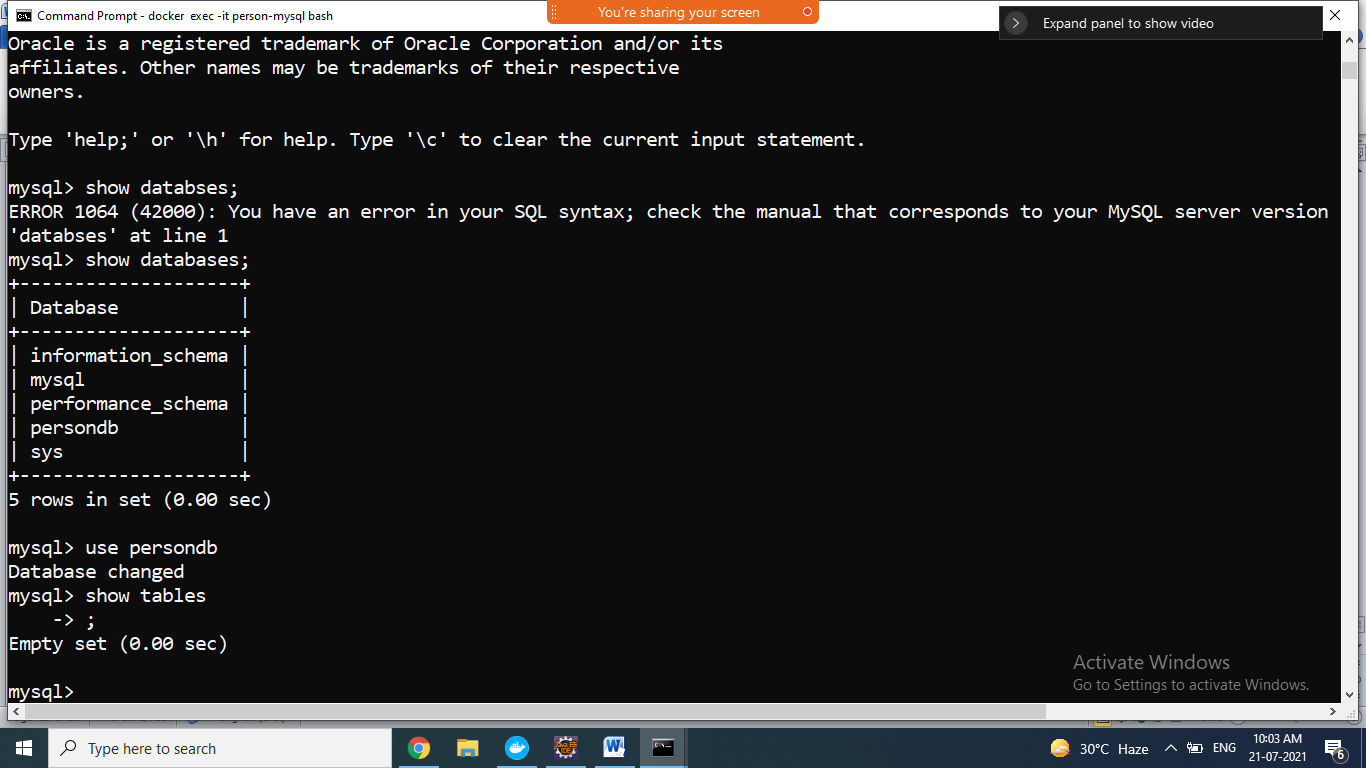
After installation try

#mysql –uroot –proot

Mysql> show databases;

Mysql>use persondb

Mysql>show tables;



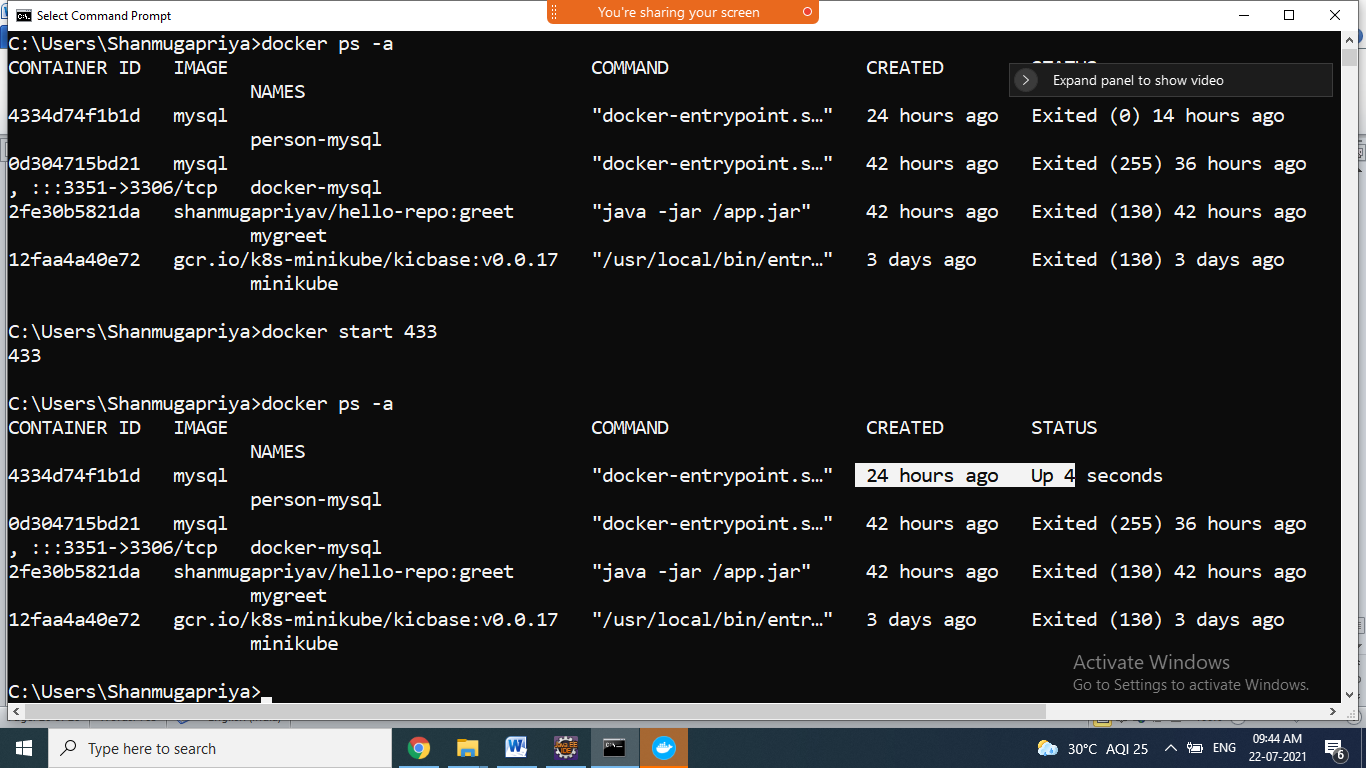
Create spring boot rest application with CRUD operation

And connect to person-mysql

1. Create spring boot CRUD application
2. Application.properties change database url instead of localhost use container name and portno and database name

spring.datasource.url=jdbc:mysql://person-mysql:3306/persondb

1. Create Dockerfile
2. Generate jar file for ur application
3. If mysql container is not running , u start by using docker start command

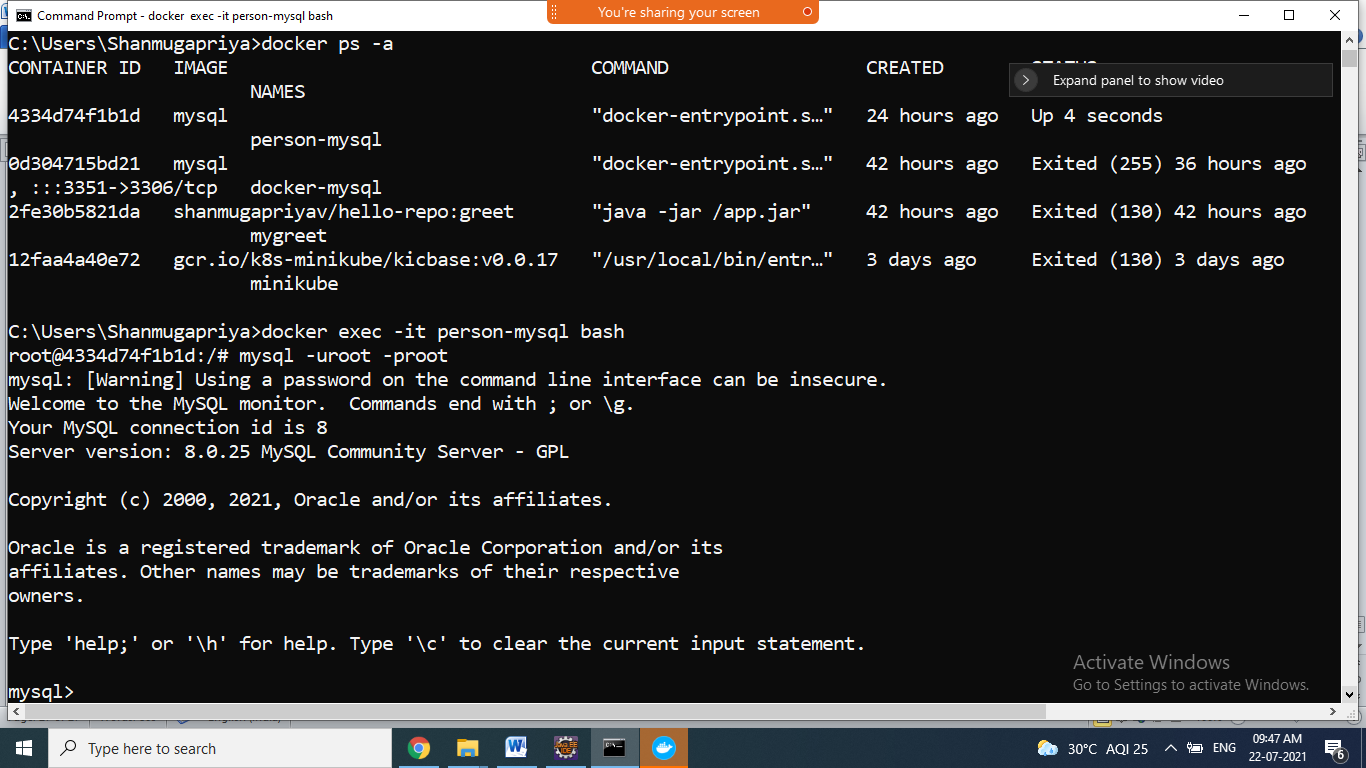


1. After running u connect to mysql command prompt by -using the command

>docker exec –it person-mysql bash

#mysql –uroot –proot

Mysql> u can try here ur sql queries



Mysql>show databases

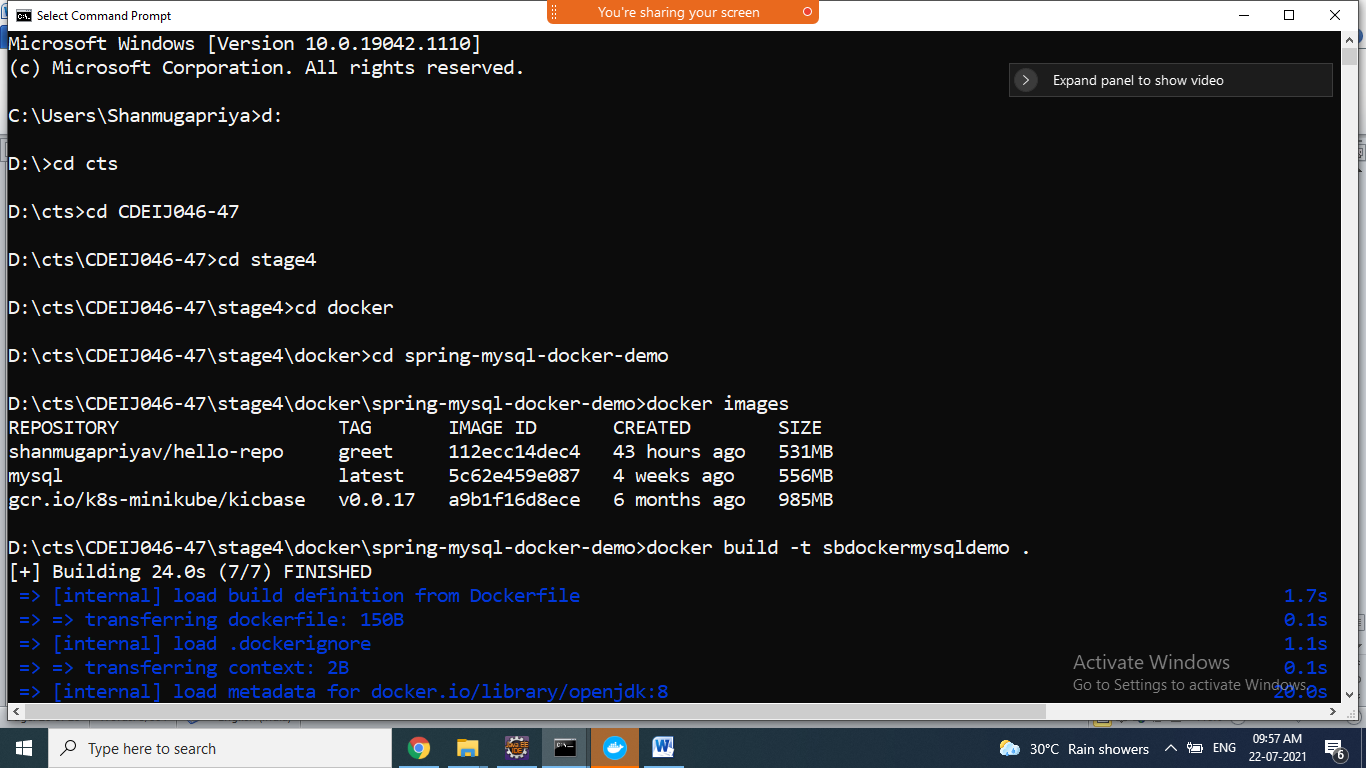
Mysql>use persondb;

Mysql>show tables;

Empty

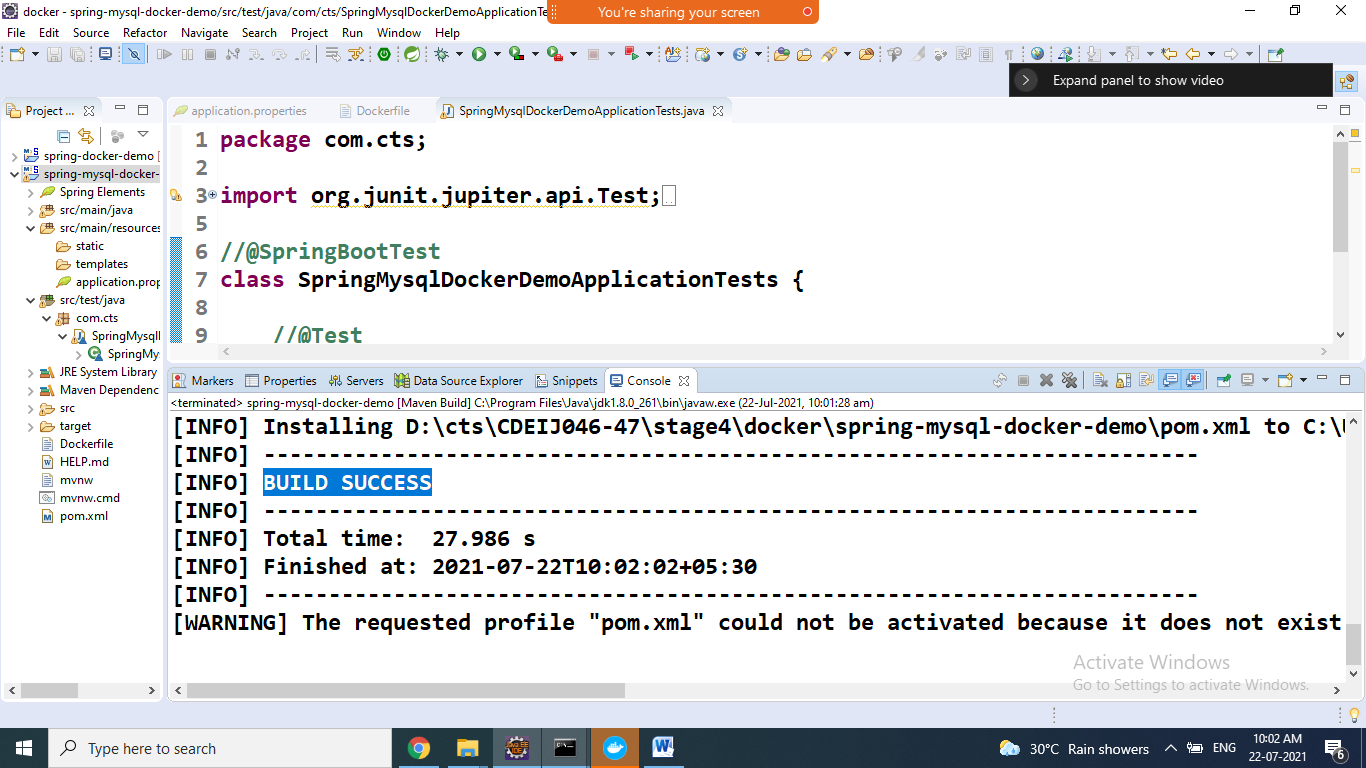
1. Create jar file ur application by using runas=>maven build=>type goals: clean install=> click run button
2. Jar file created successfully
3. Open command prompt of the project folder
4. Create docker image for ur spring boot application by using docker build command

>docker build –t sbdockermysqldemo .



If u get error while creating jar file coomucation link failure , connection refued error

U comment the test annotations in test class then create jar file



Now u create docker image

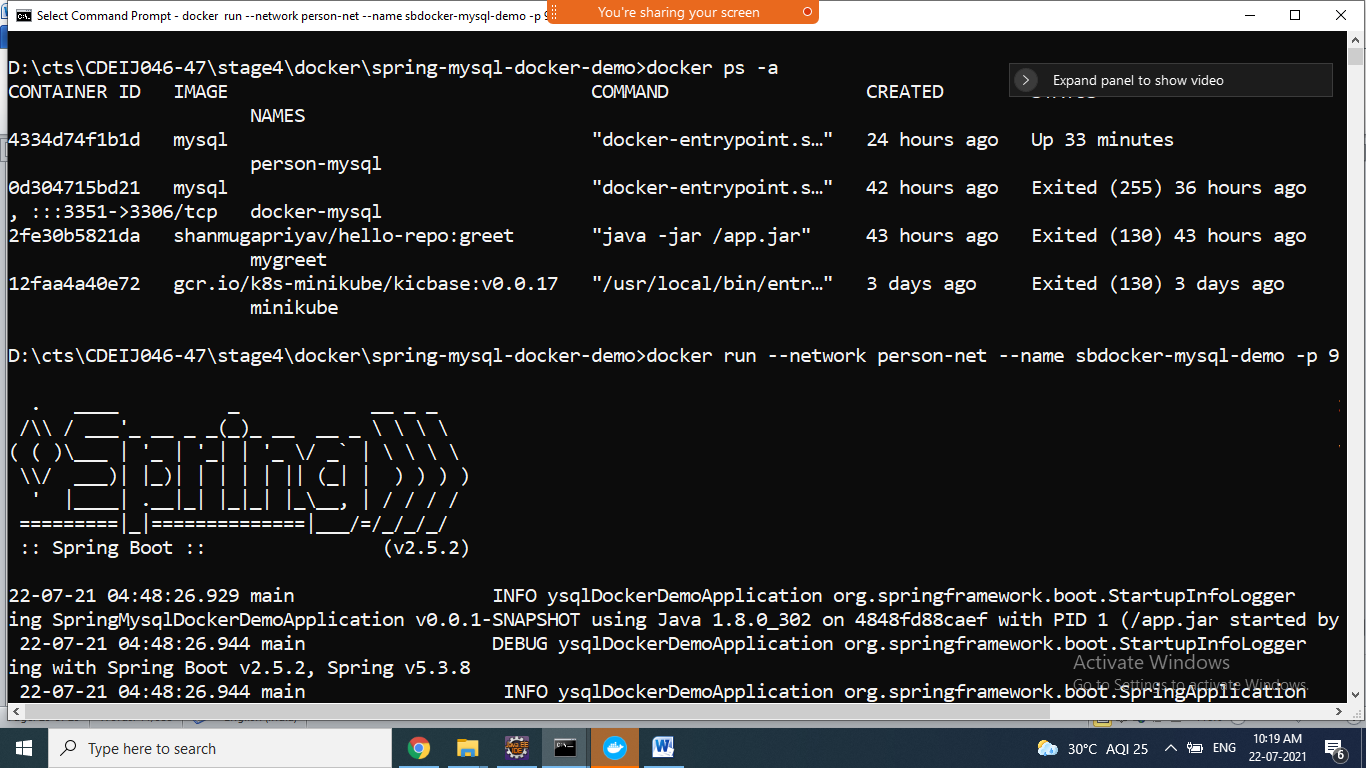
>docker build –t sbdockermysqldemo .

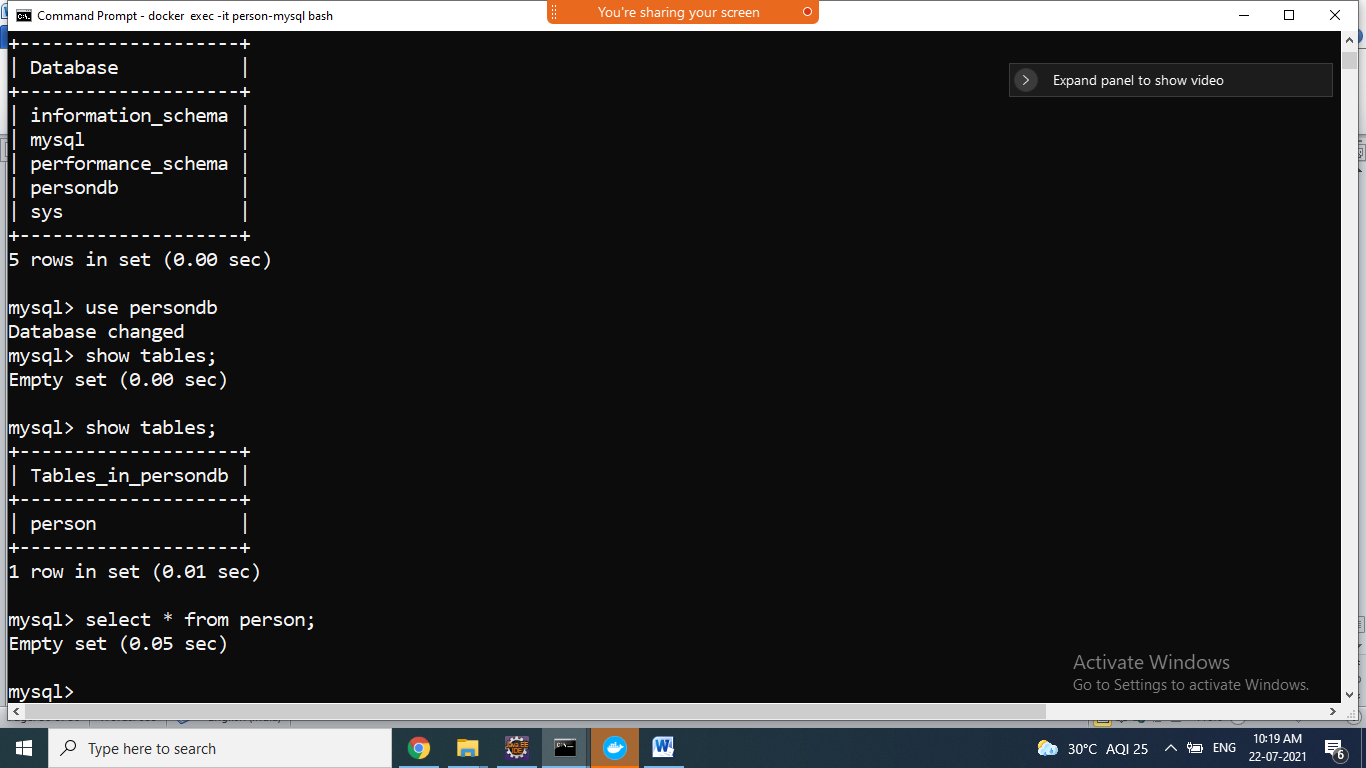
Successfully image created

1. Now u run the spring application image using below command

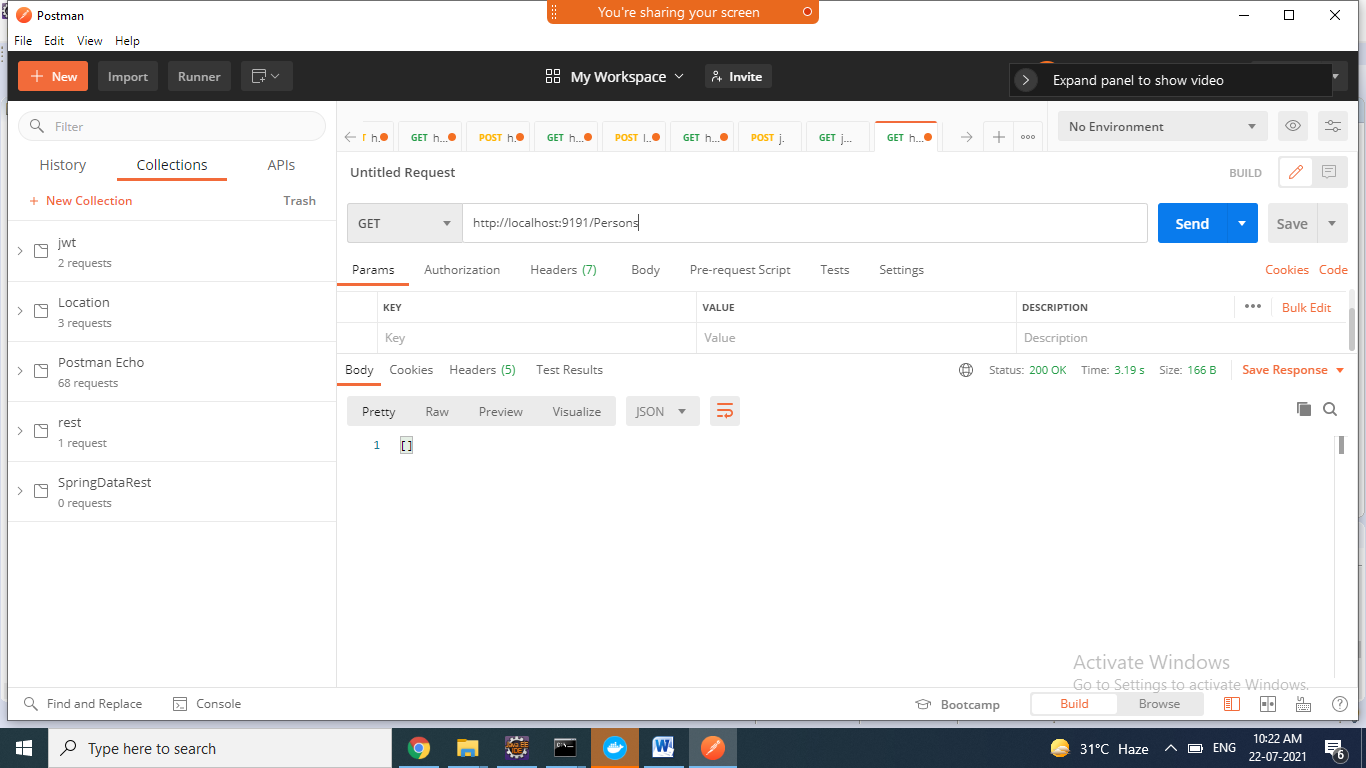
>docker run --network person-net --name sbdocker-mysql-demo –p 9191:9191 sbdockermysqldemo

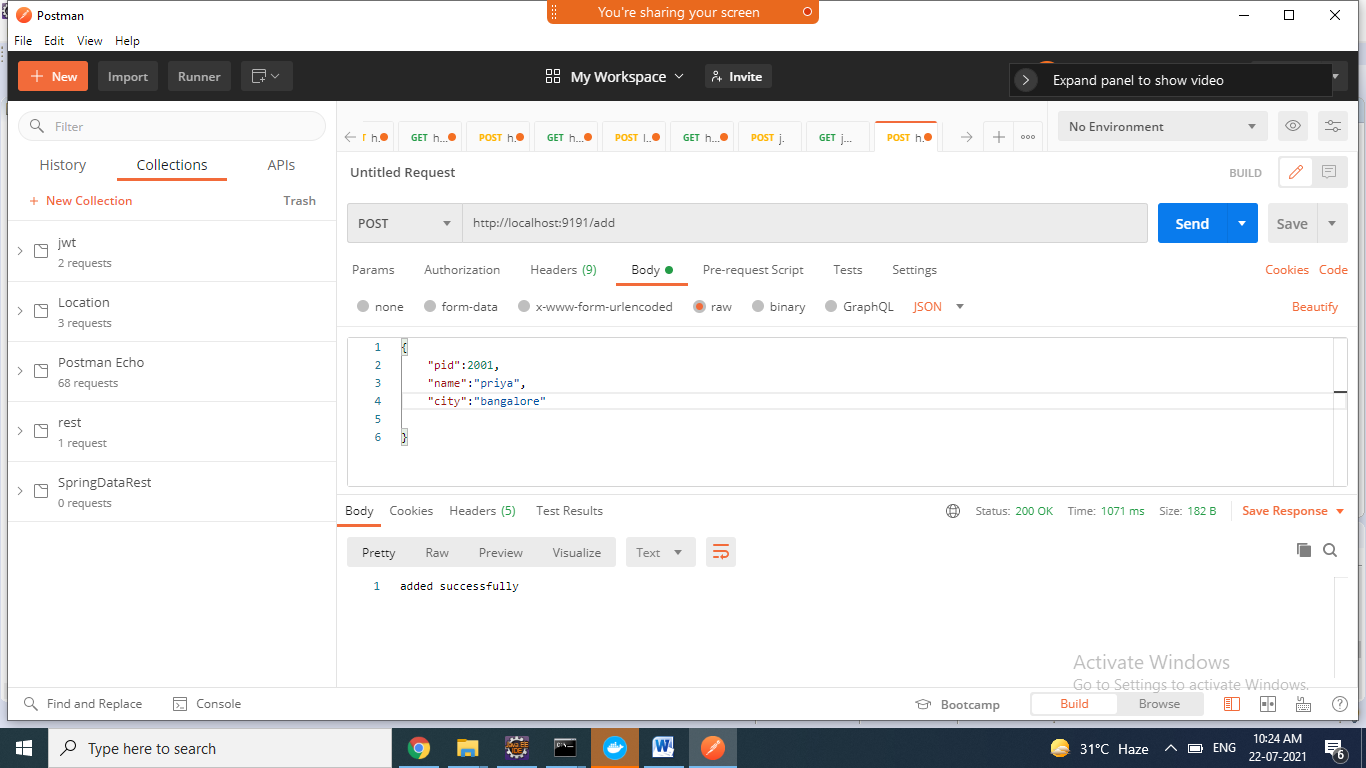
docker run –network name –name containername –p 8080:9191 –d imagename

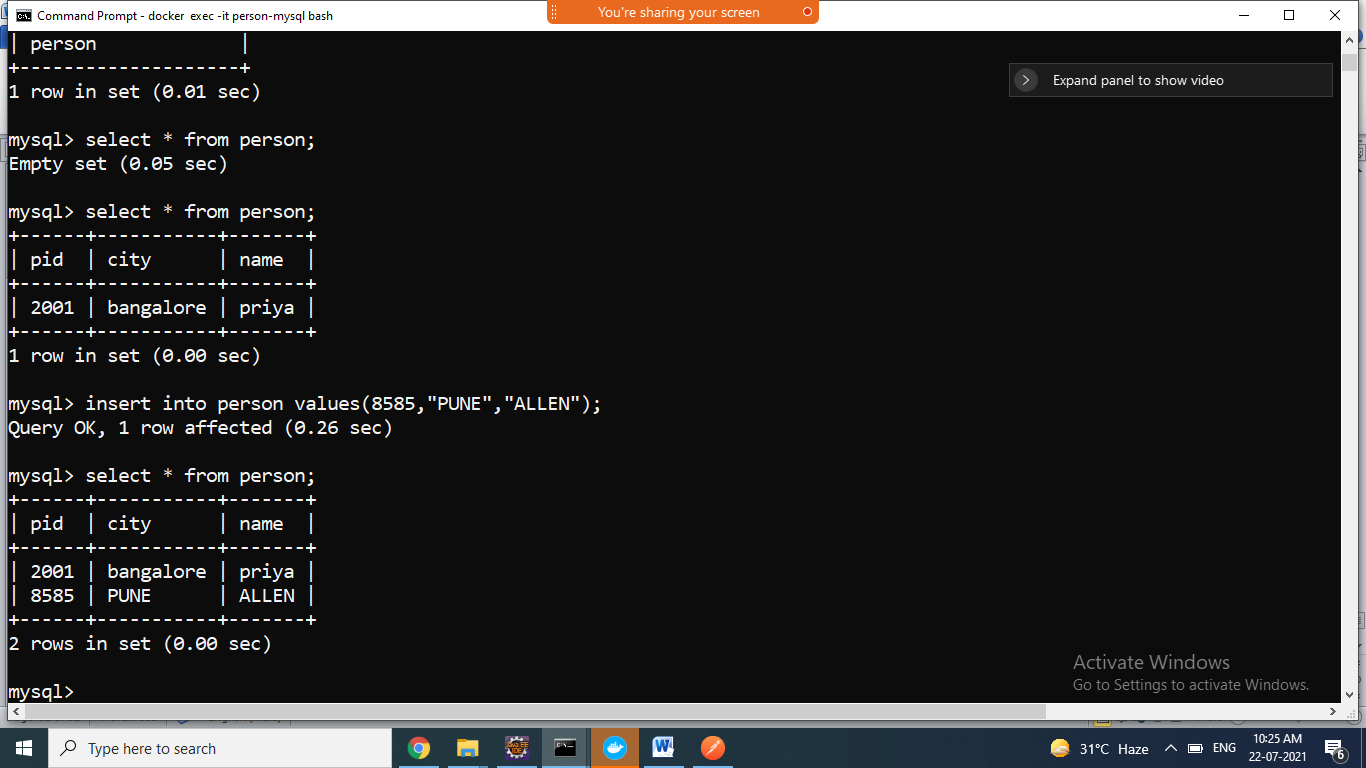


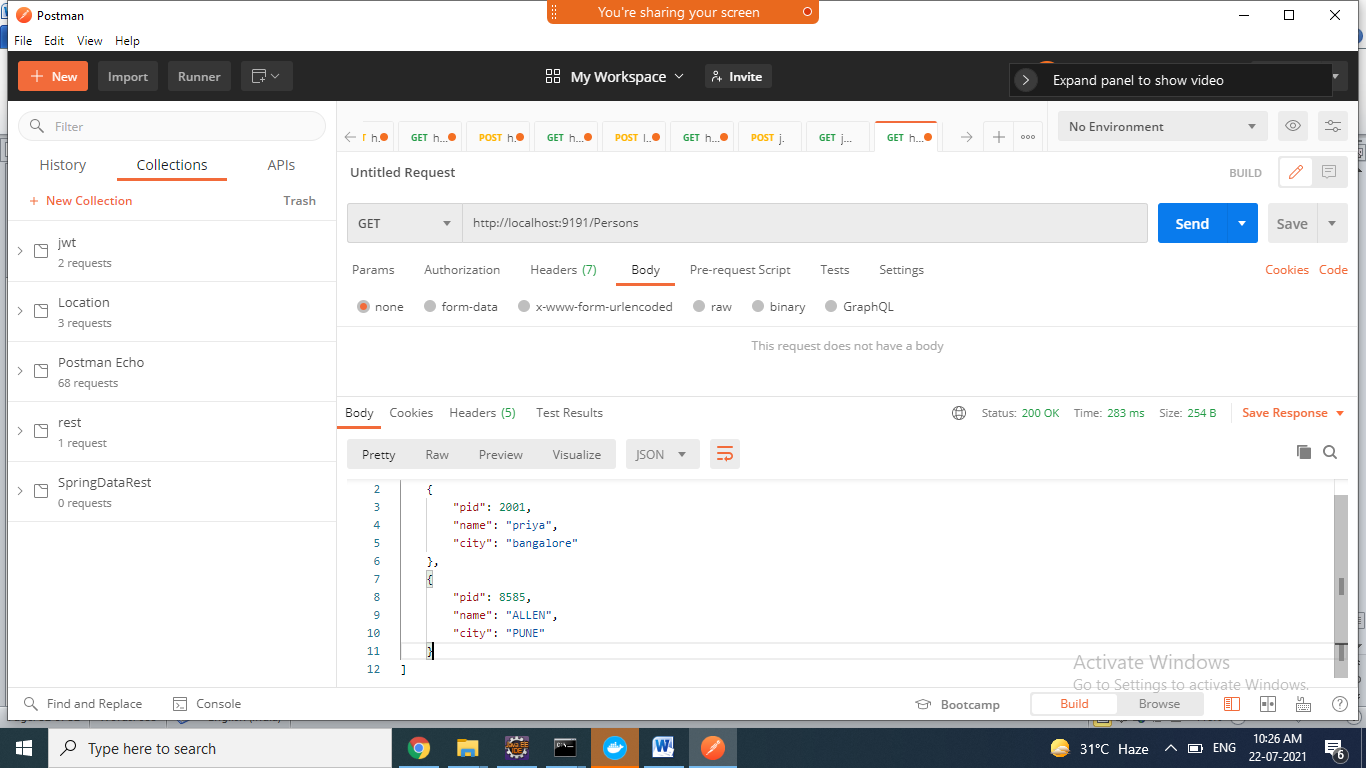


1. Go to browser/postman u can access endpoints and also check docker mysql command prompt updated data using query









===========================================================

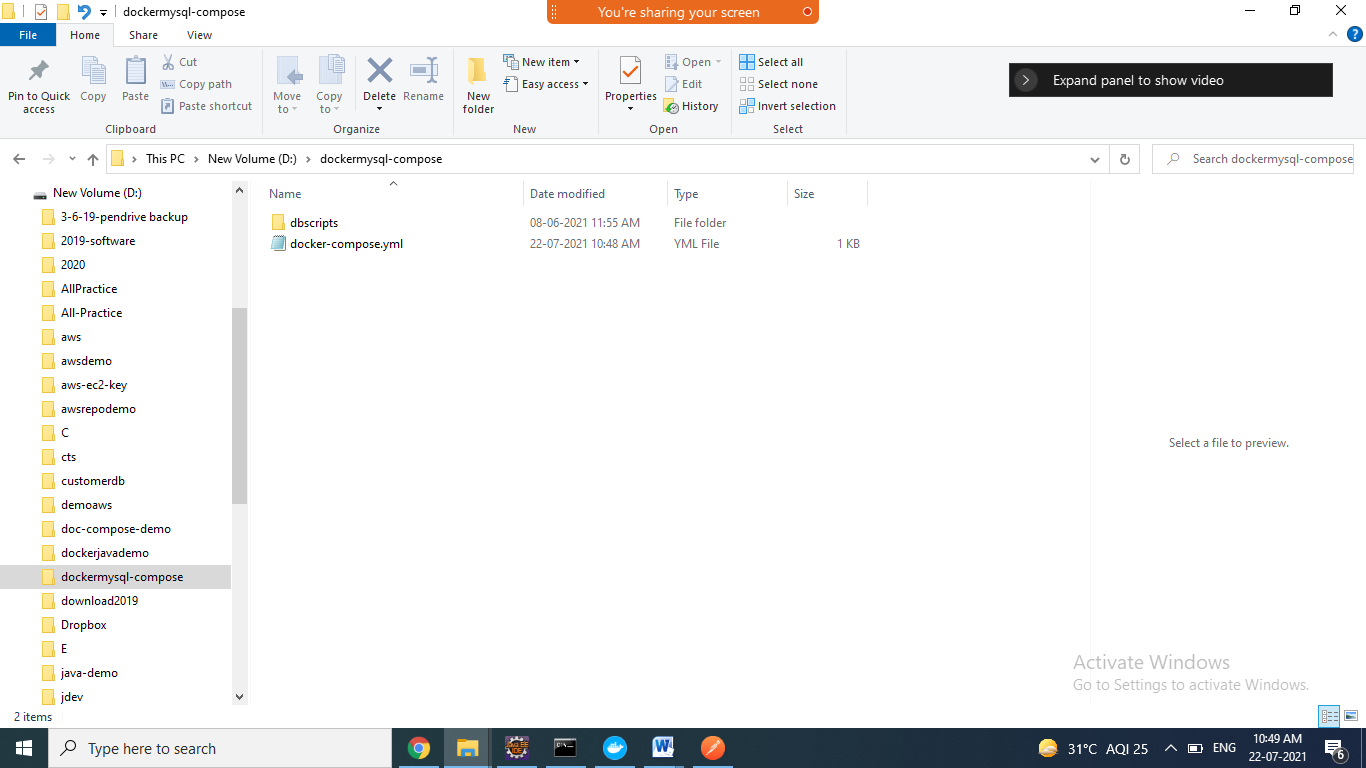
**Docker-compose** => it automate all ur process

MYSQL => DOCKERCOMPOSE

1. Create one folder in D drive ie dockermysql-compose
2. Create dbscripts folder inside dockermysql-compose

Create one data.sql file inside dbscripts

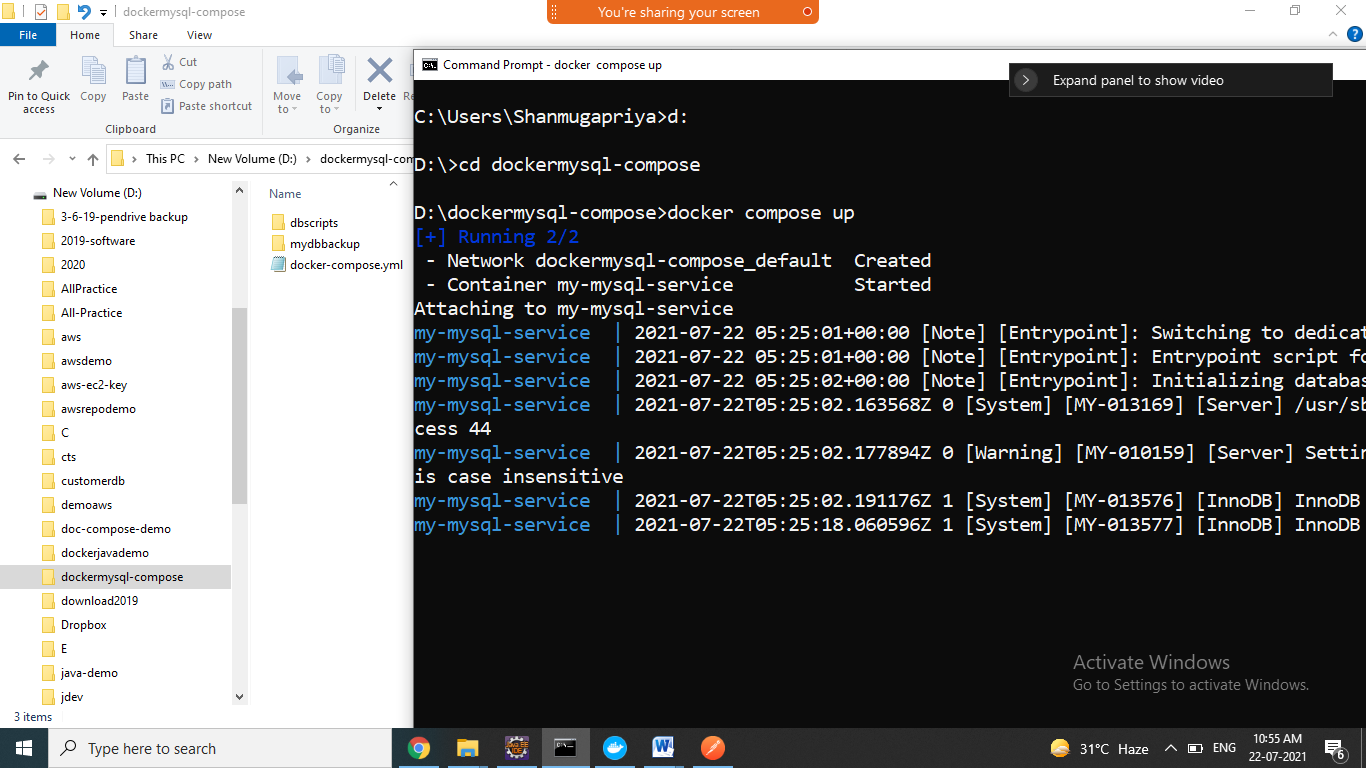
1. docker-compose.yml file in dockerrmyql-compose folder



1. goto the command prompt of the current folder
2. type the below command

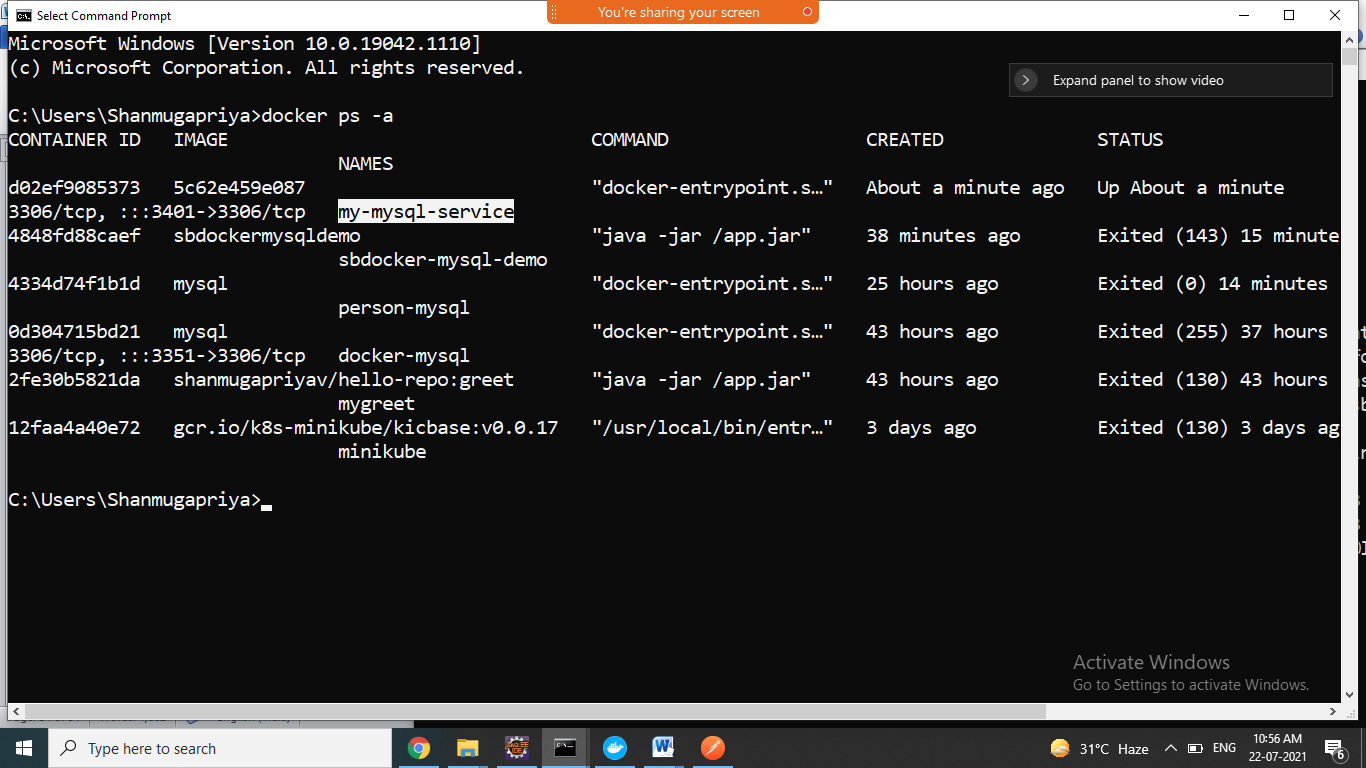
>docker compose up

* it create mysql container and runs the data.sql file file



1. go to another command prompt check container is created

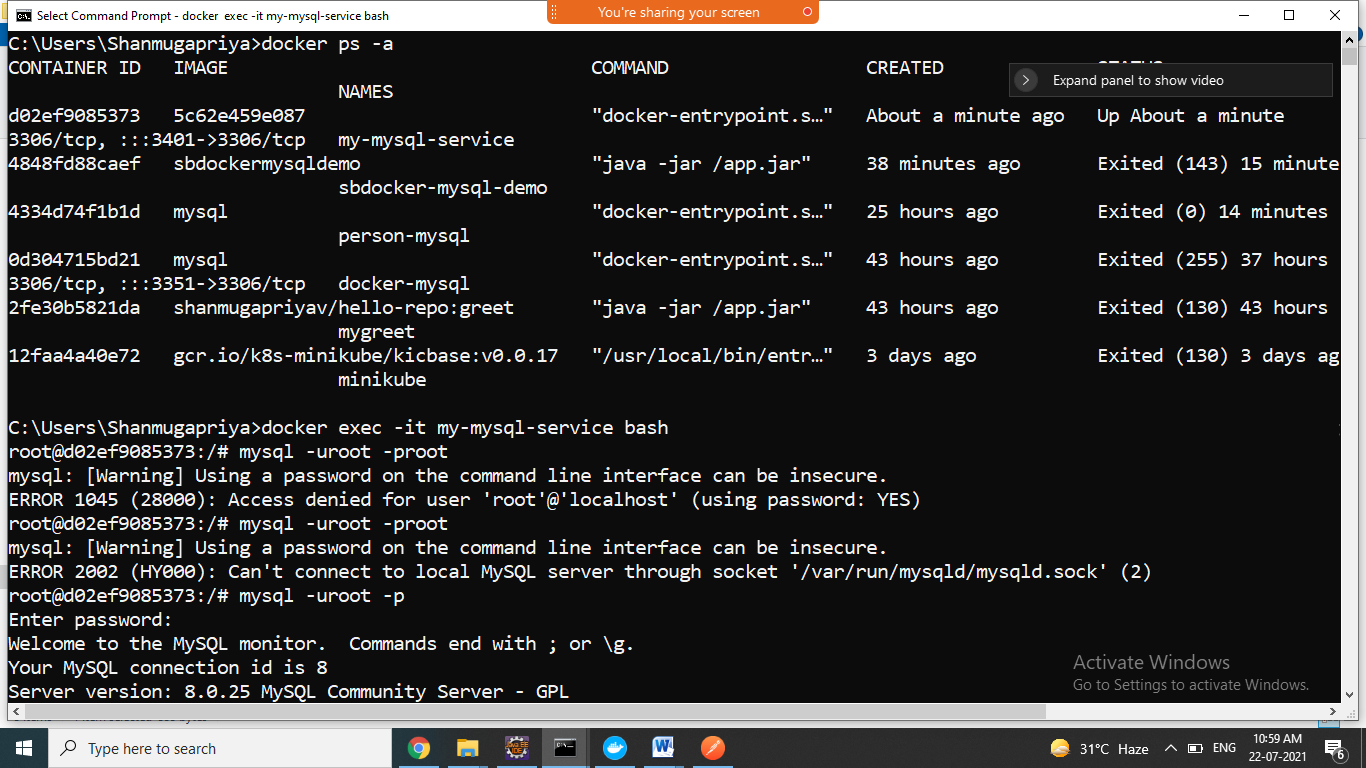
>docker ps –a

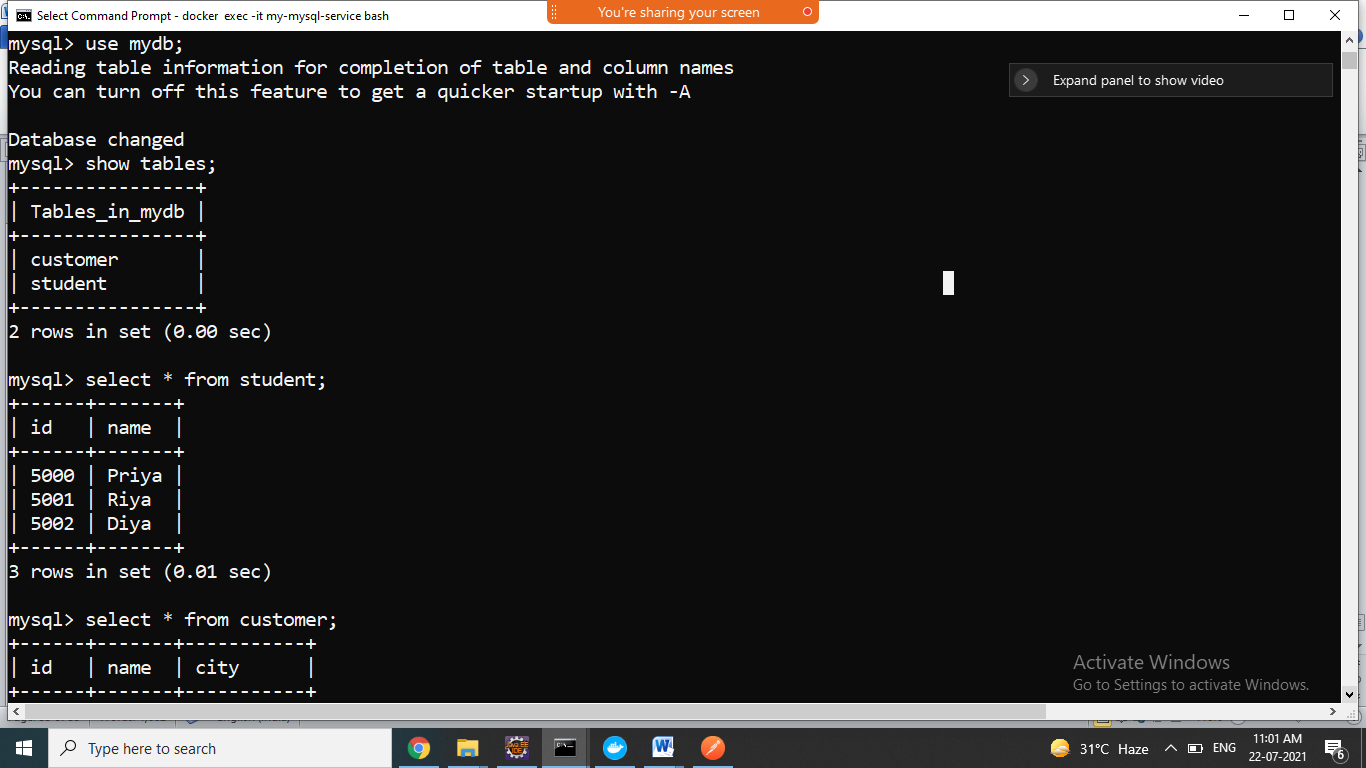


1. now connect to the mysql container check the tables are created , using below command

>docker exec –it my-mysql-service bash

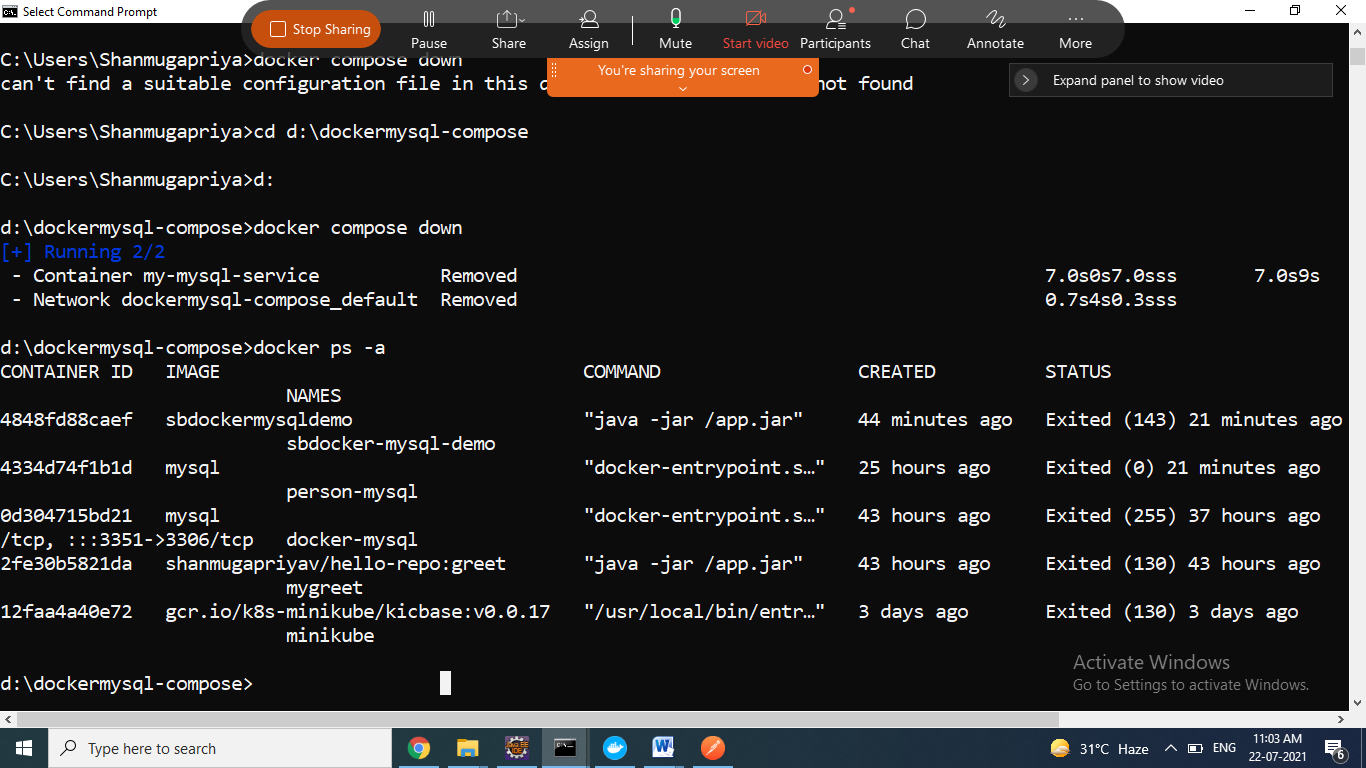
#mysql –uroot –proot





1. if u want remove all container which is create by docker-compose.yml file use below command

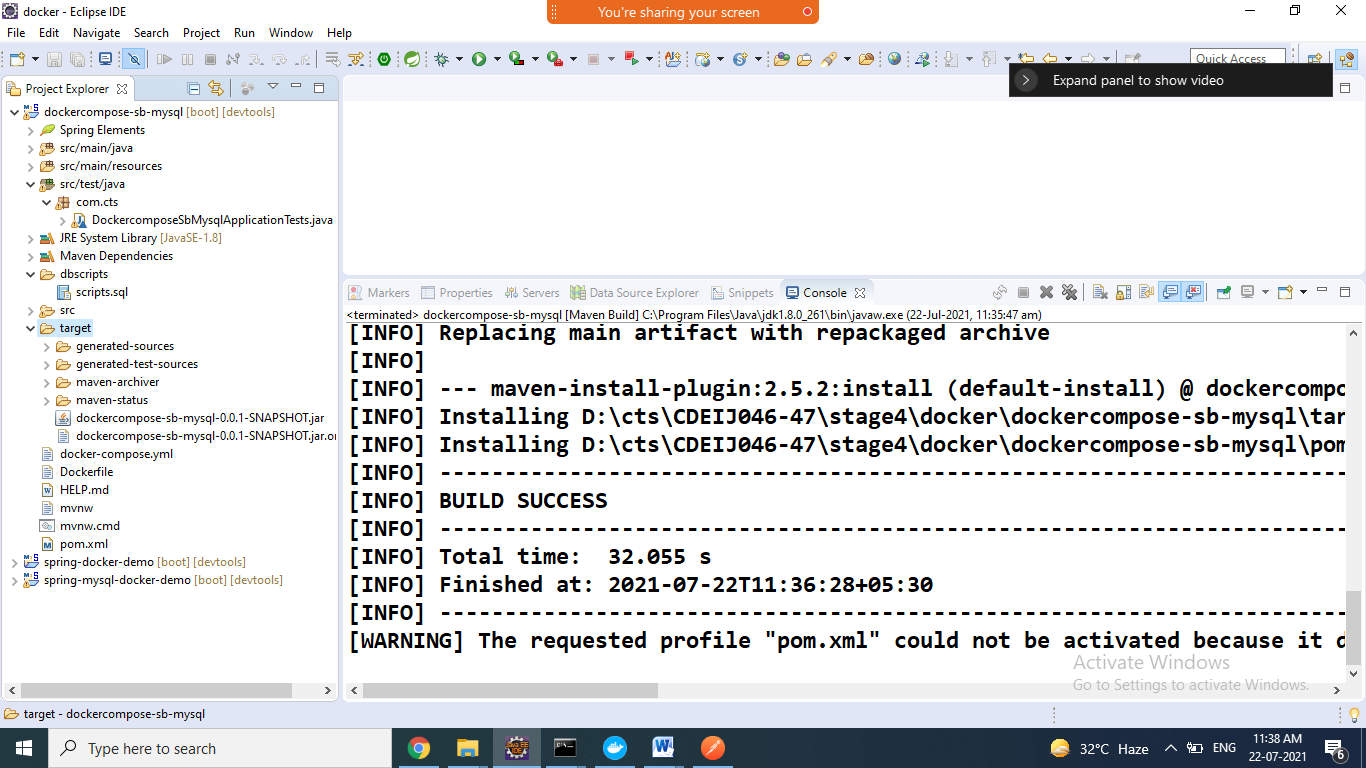
>docker compose down



=========================================================

SPRING APP +MYSQL DOCKER COMPOSE

1. spring boot CRUD application
2. Create one dockerfile and docker-compose.yml file and dbscripts in same project folder
3. Generate jar file for the project



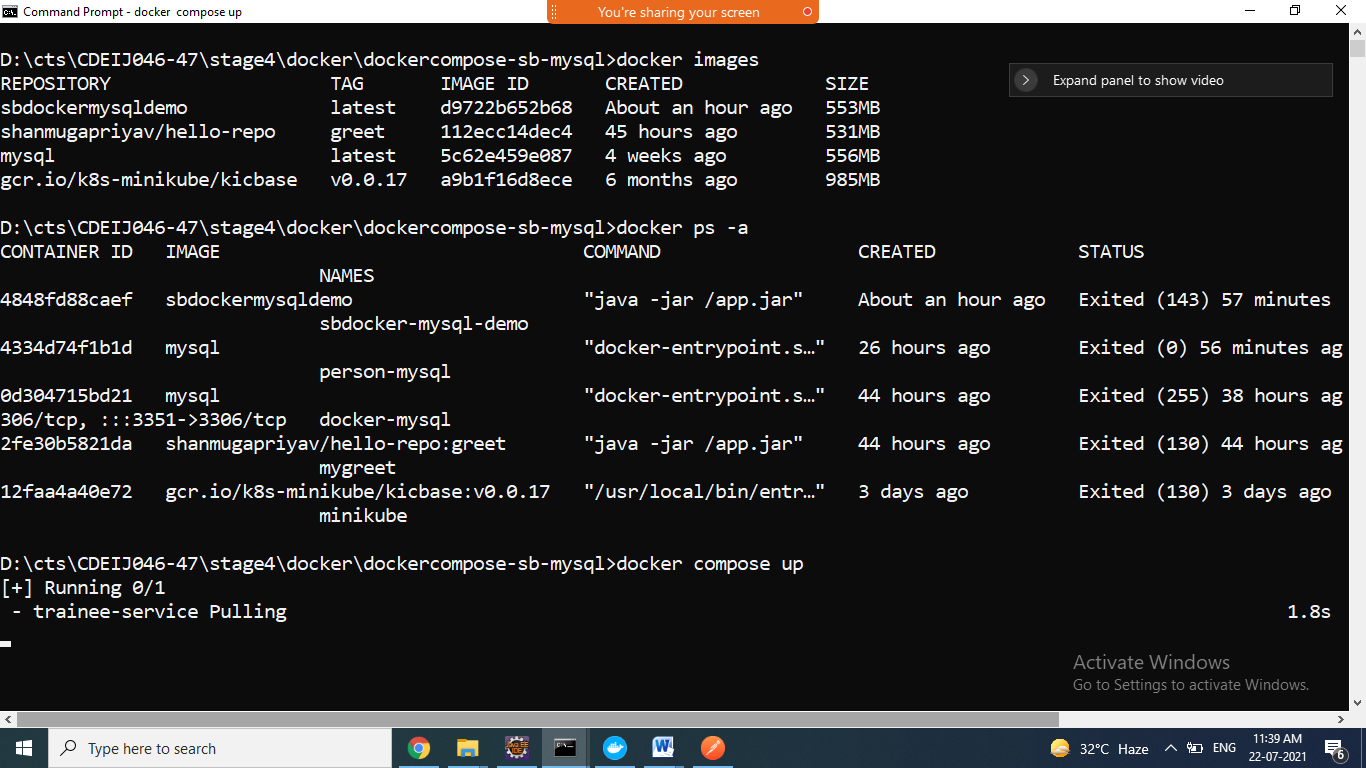
1. Open command prompt project folder directory
2. Type below command

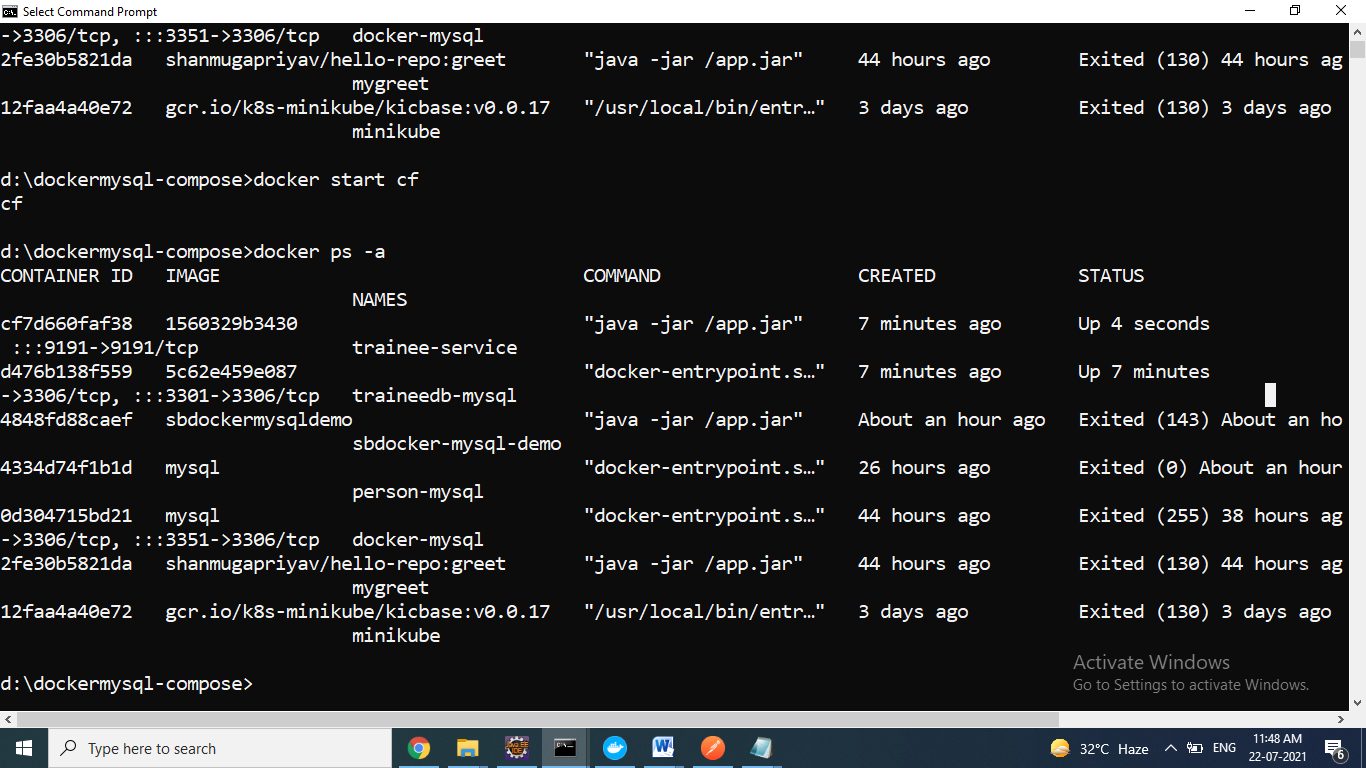
>docker compose up

It create mysql container and run the mysql container

It create sb application image container the run the application container

Finally it runs both(mysql+sb-app) containers





1. Go to browser/postman u check it using endpoints



1. Goto cmd prompt check mysql contains all records

