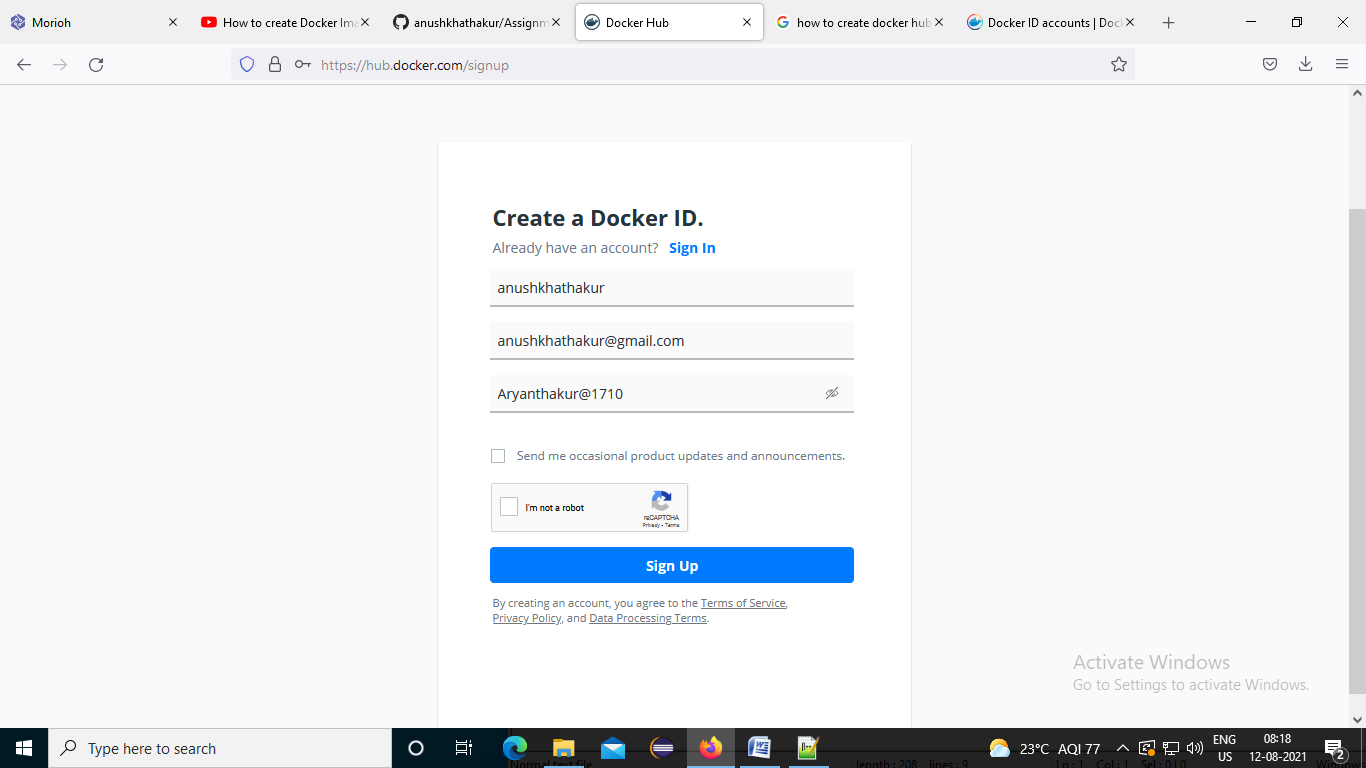
# Docker

Hub.docker.com

https://hub.docker.com/repositories

<https://docs.docker.com/get-started/02_our_app/>

https://morioh.com/p/68a319024ddc



Once you register and verify your Docker ID email address, you can log in to [Docker Hub](https://hub.docker.com).

You can also log in through the CLI using the docker login command.

1. Go to spring boot app and put <finalName>docker-service-registry</finalName> in pom.xml

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

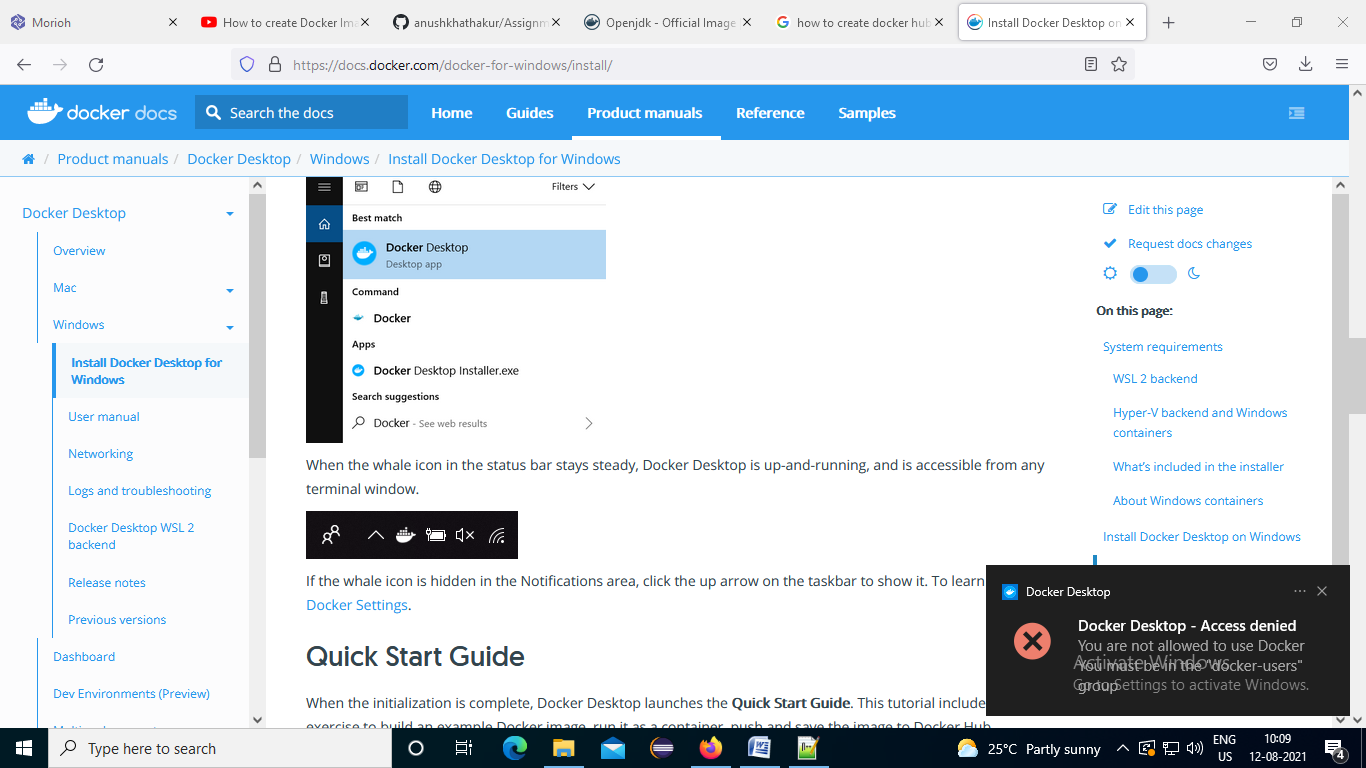
</plugin>

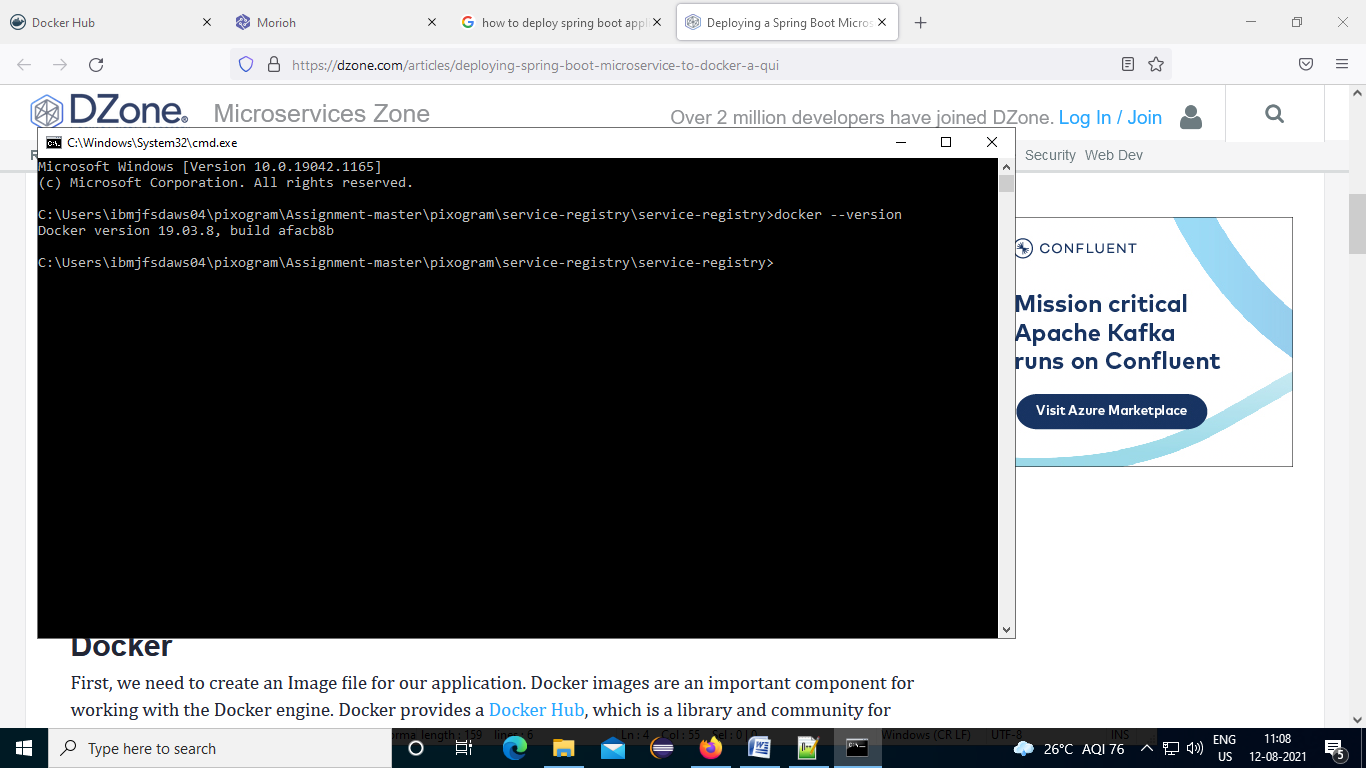
</plugins>

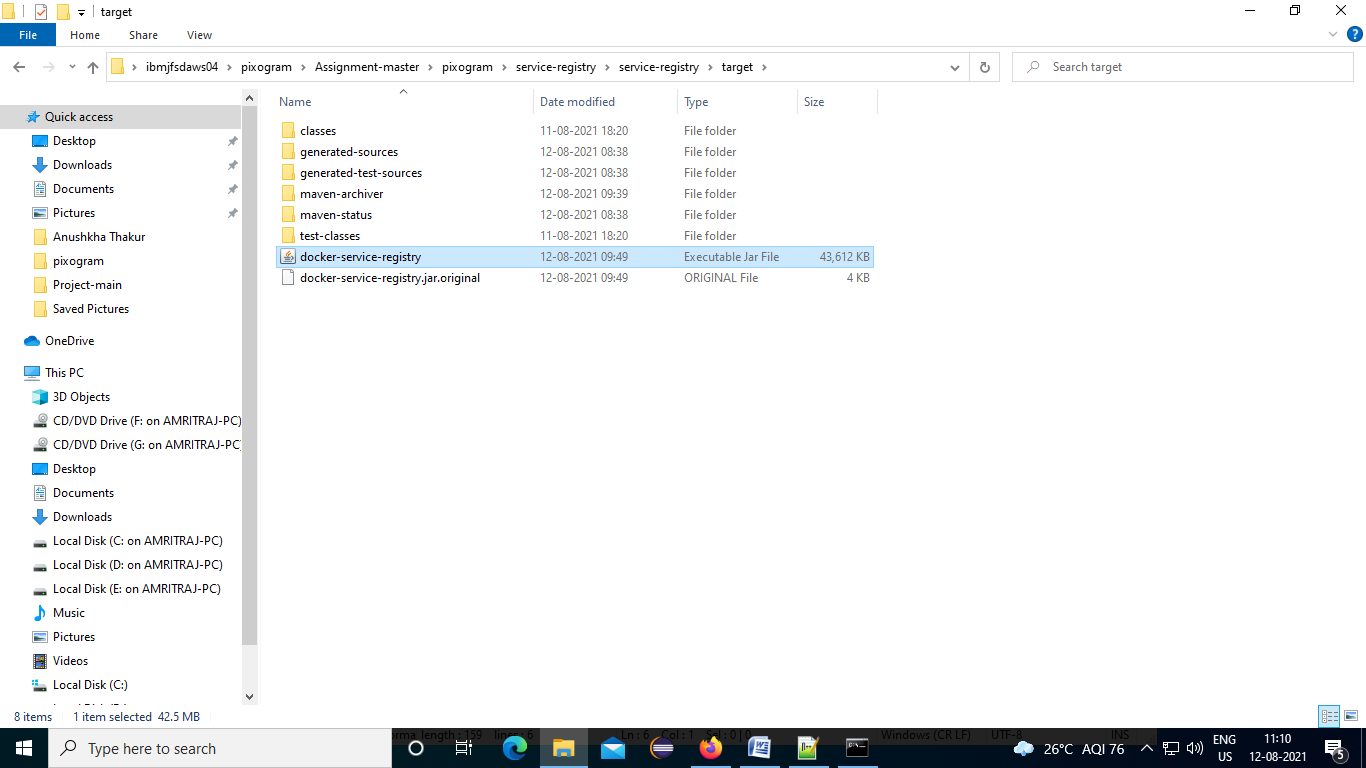
<finalName>docker-service-registry</finalName>

</build>

1. Create docker file



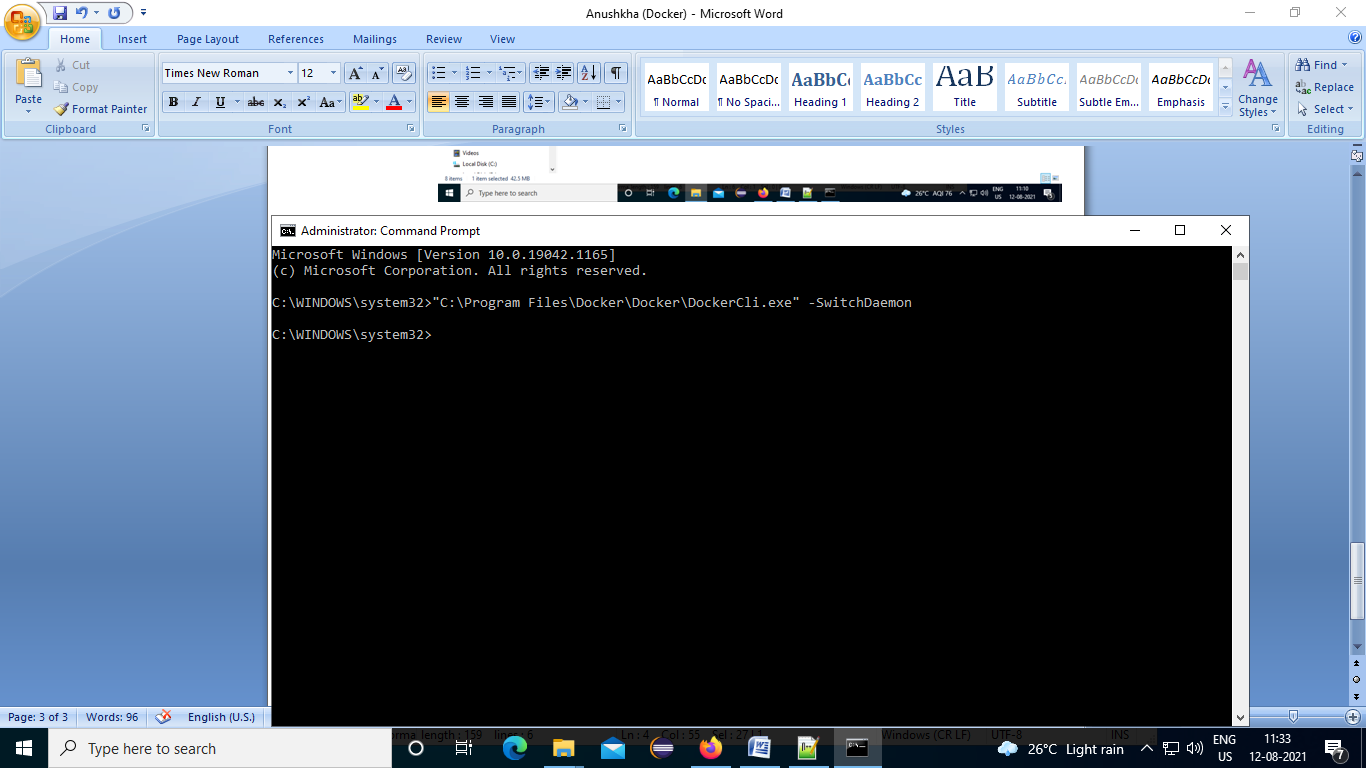


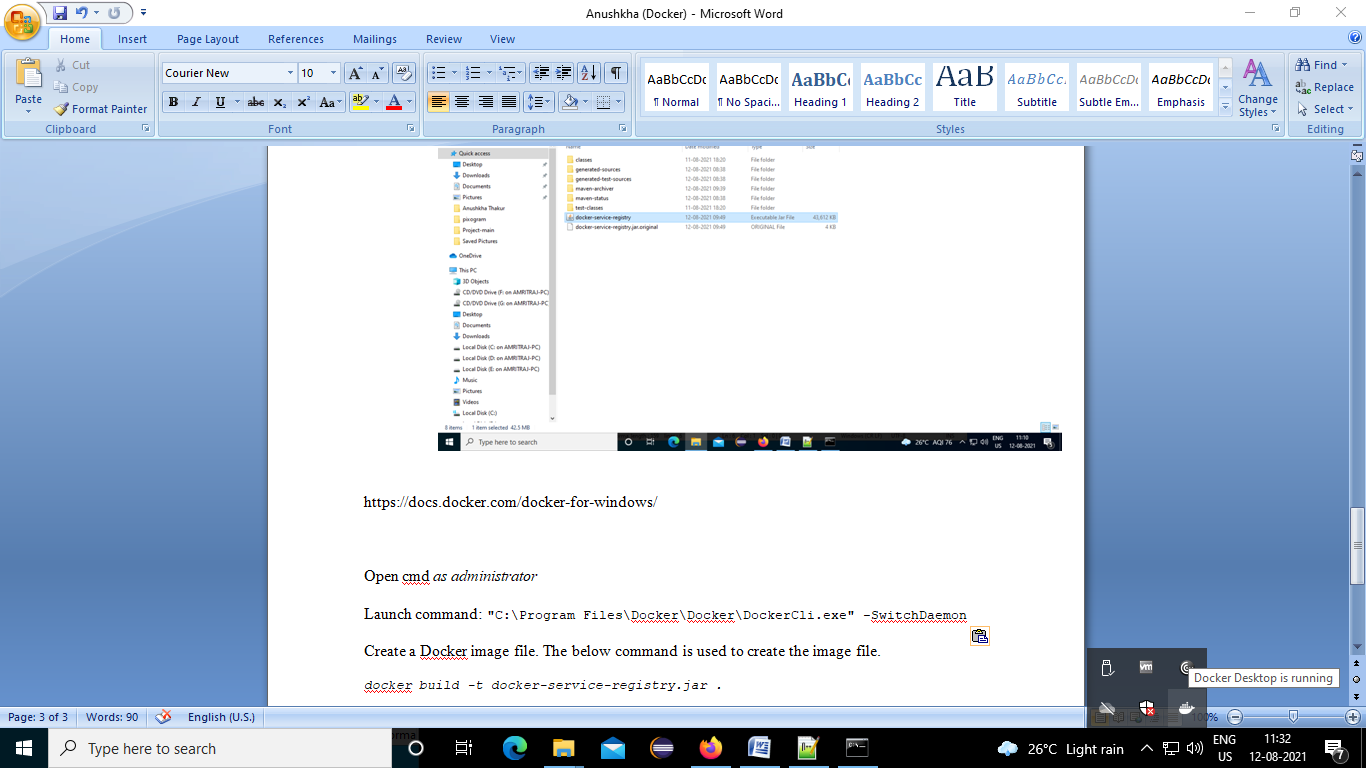


https://docs.docker.com/docker-for-windows/

1. click on docker desktop
2. Open cmd as administrator

Launch command: "C:\Program Files\Docker\Docker\DockerCli.exe" –SwitchDaemon

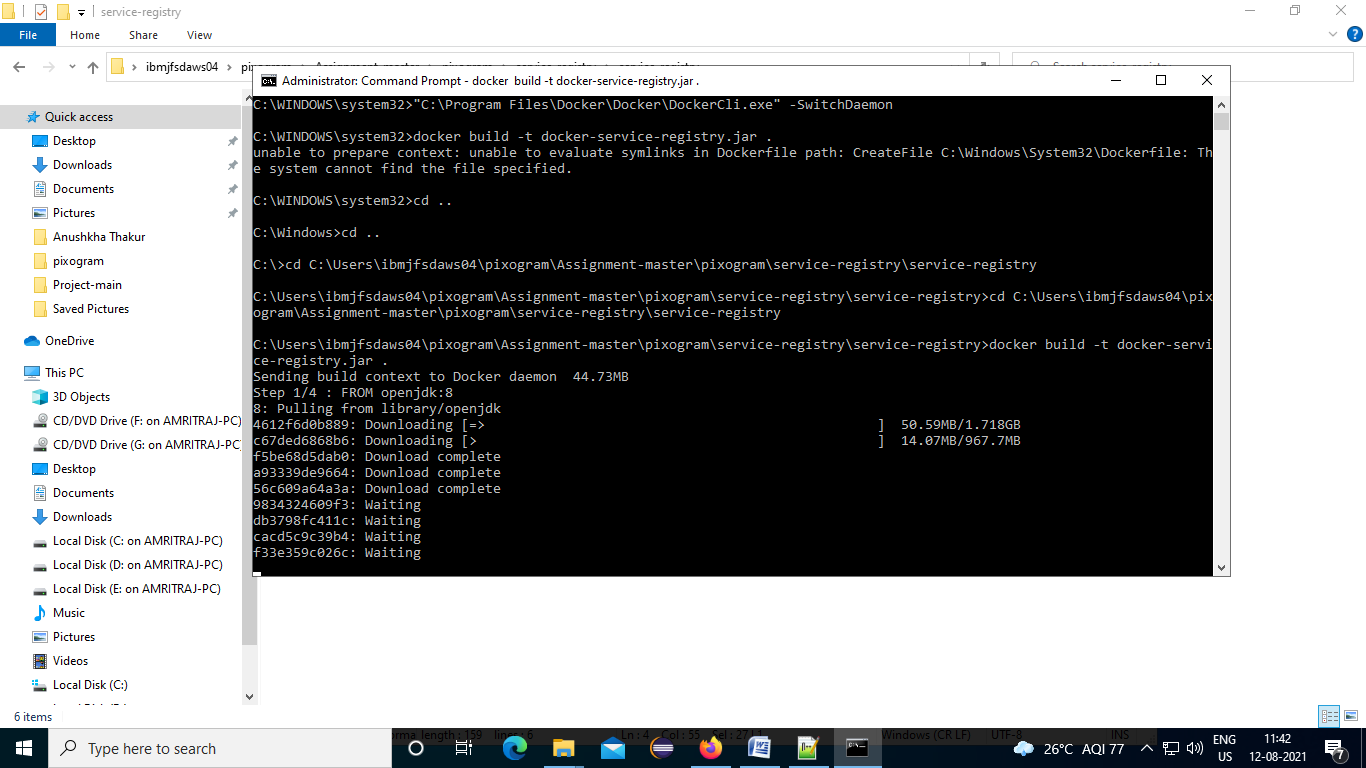


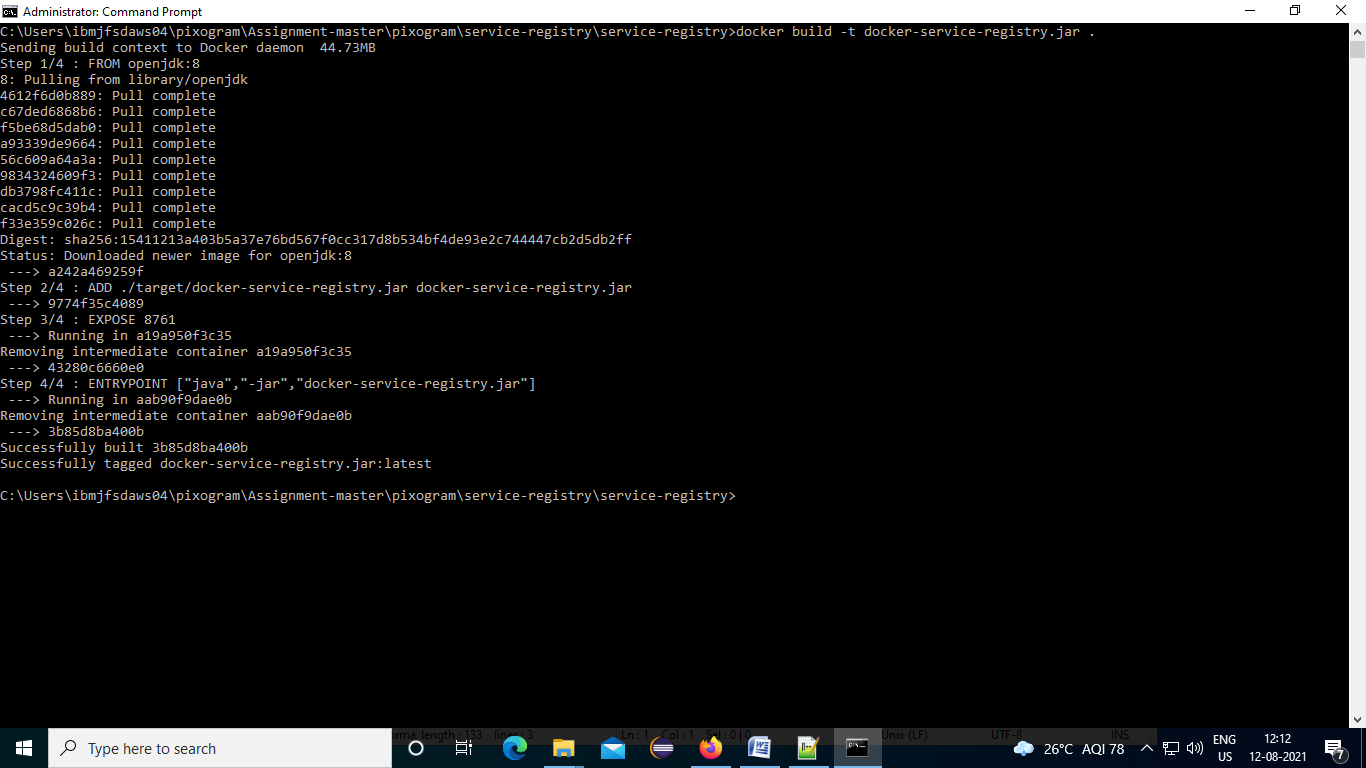


1. Create a Docker image file. The below command is used to create the image file.

*docker build -t docker-service-registry.jar .*

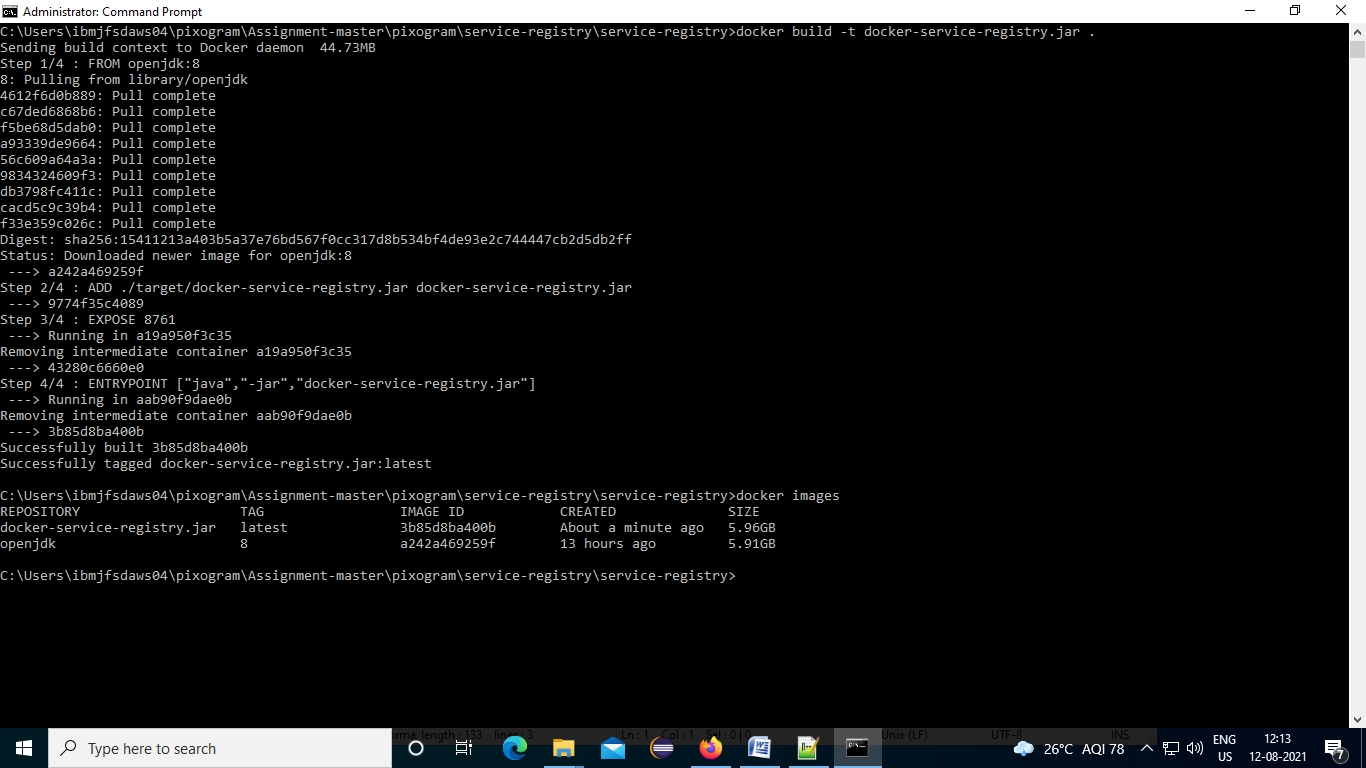
1. Go to the file location where your docker file is placed in spring boot application



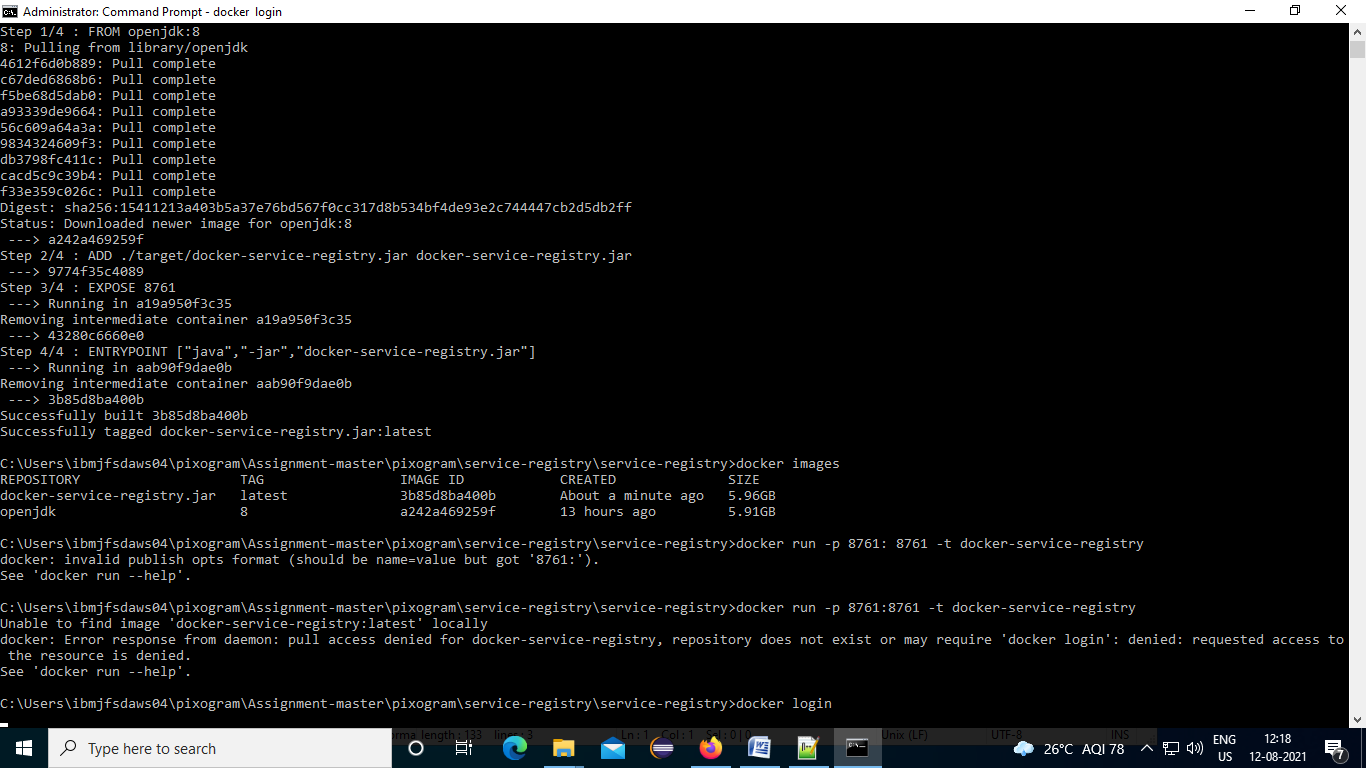


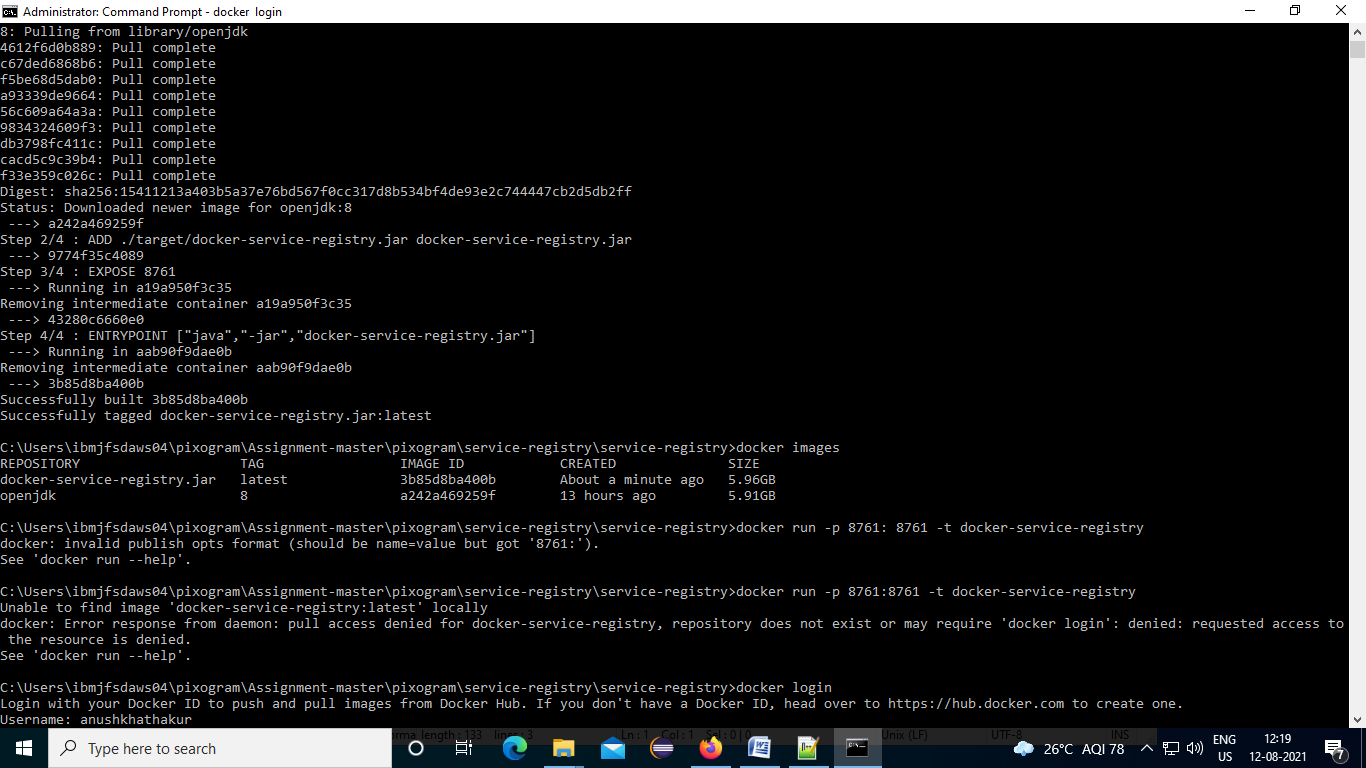
1. You can check the created image by using the "docker images" command.

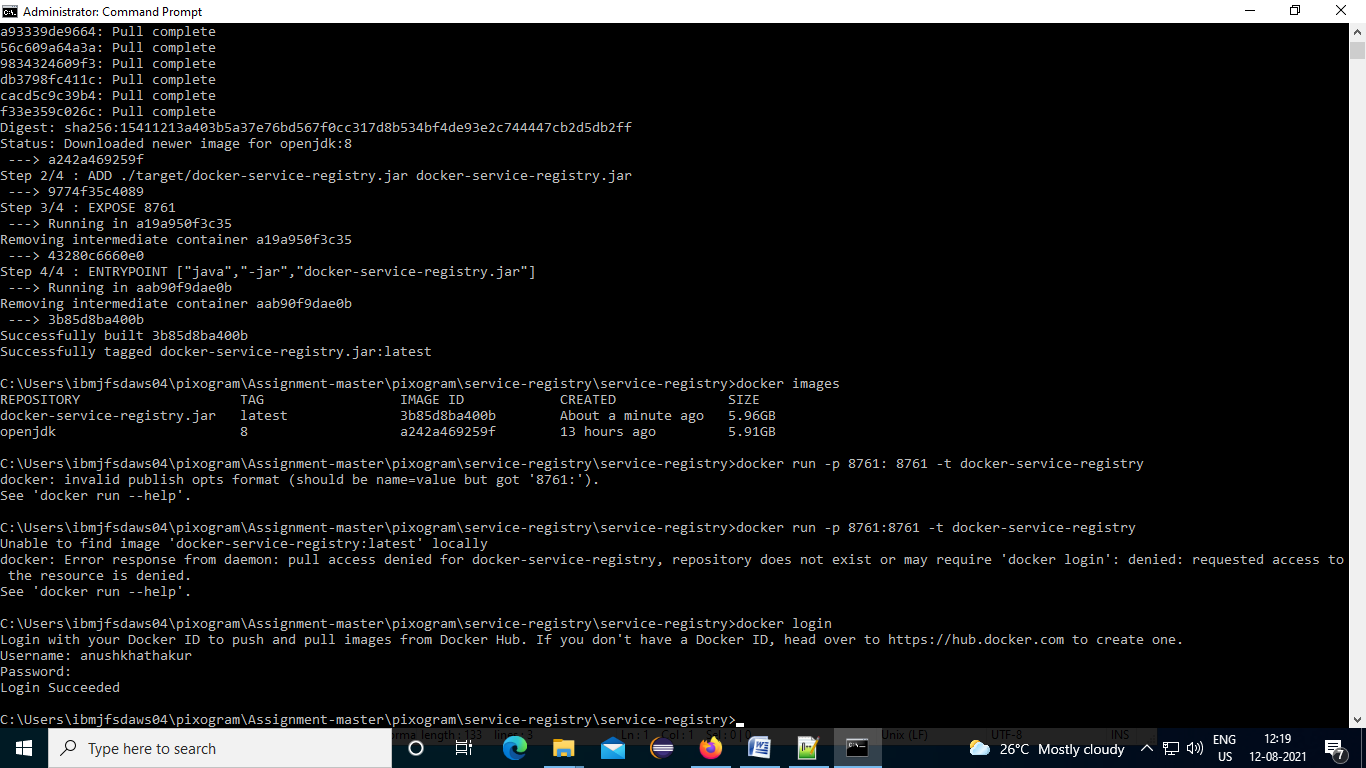
docker images

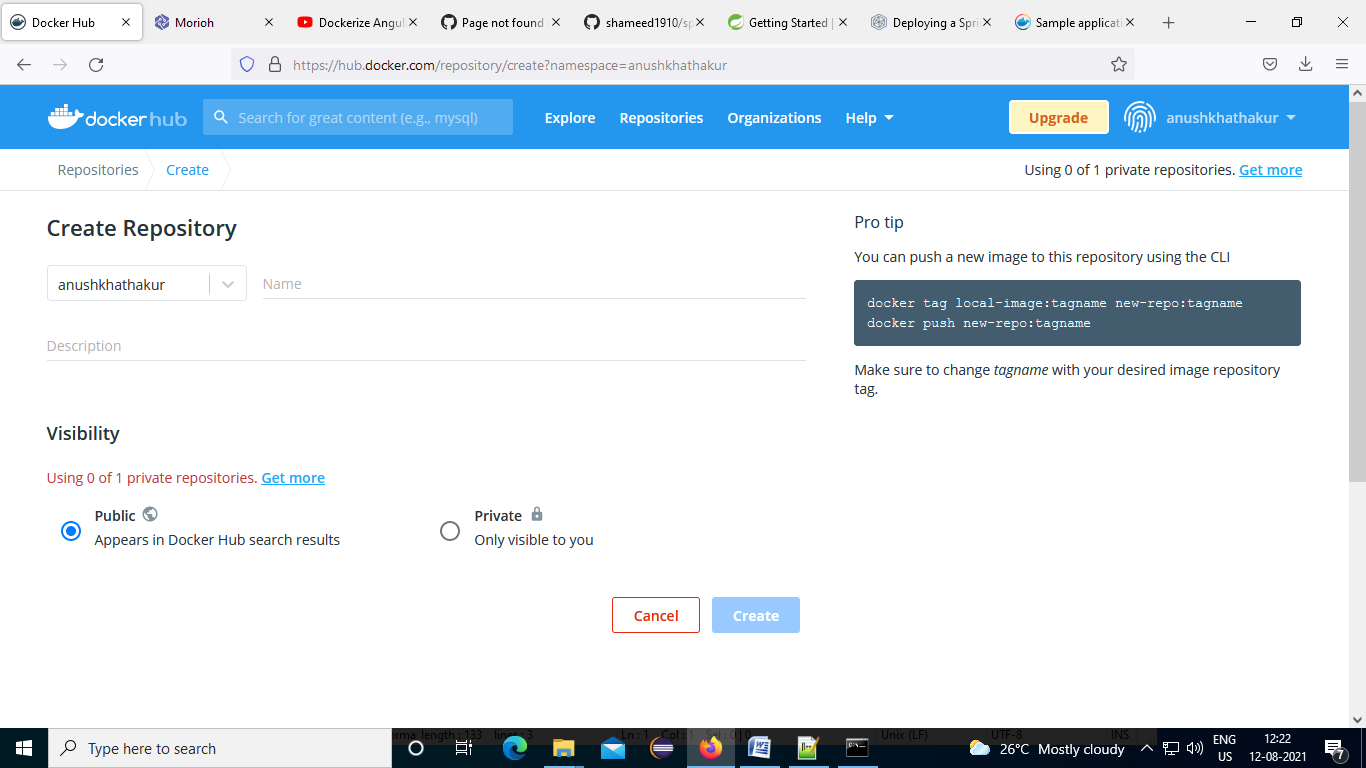


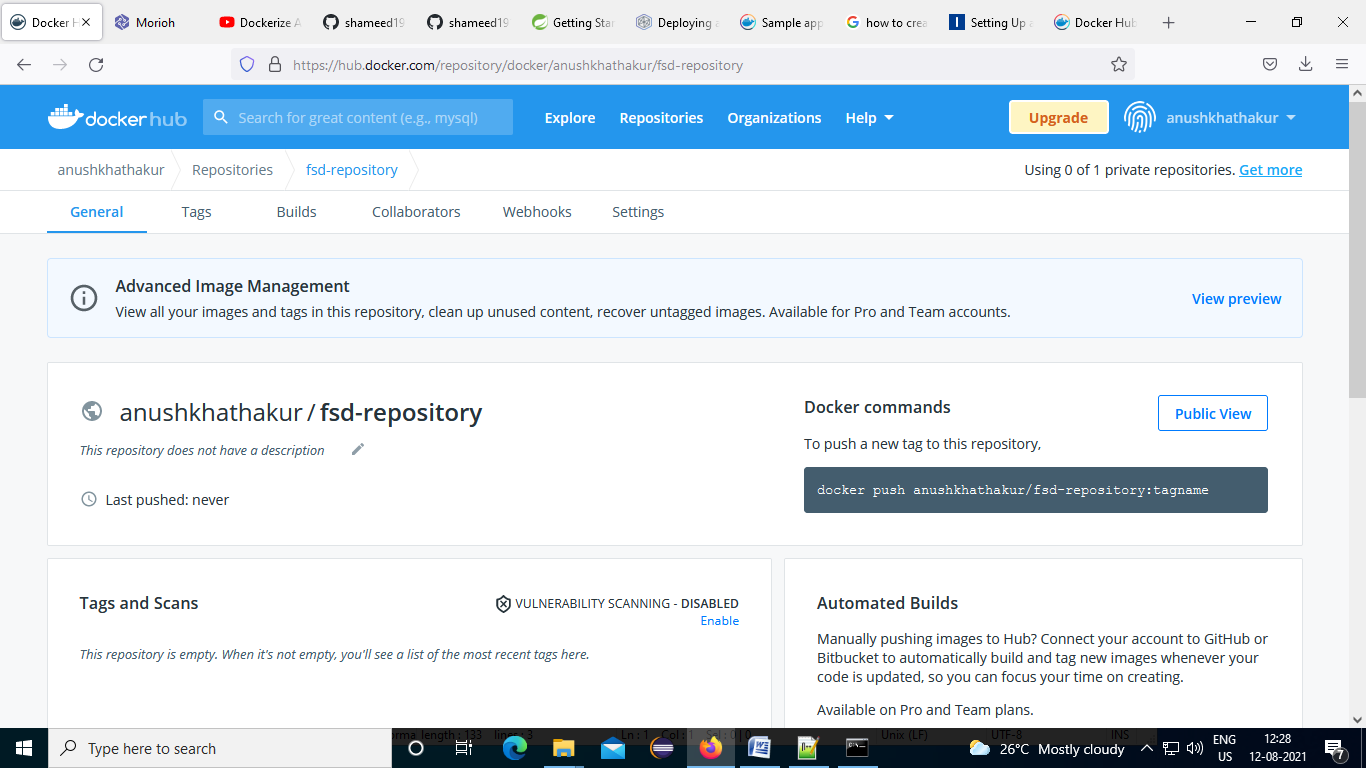
1. To push to a Docker registry, you need to have permission to push, which you do not have by default. Change the image prefix to your own Dockerhub ID and docker login to make sure you are authenticated before you run Docker.

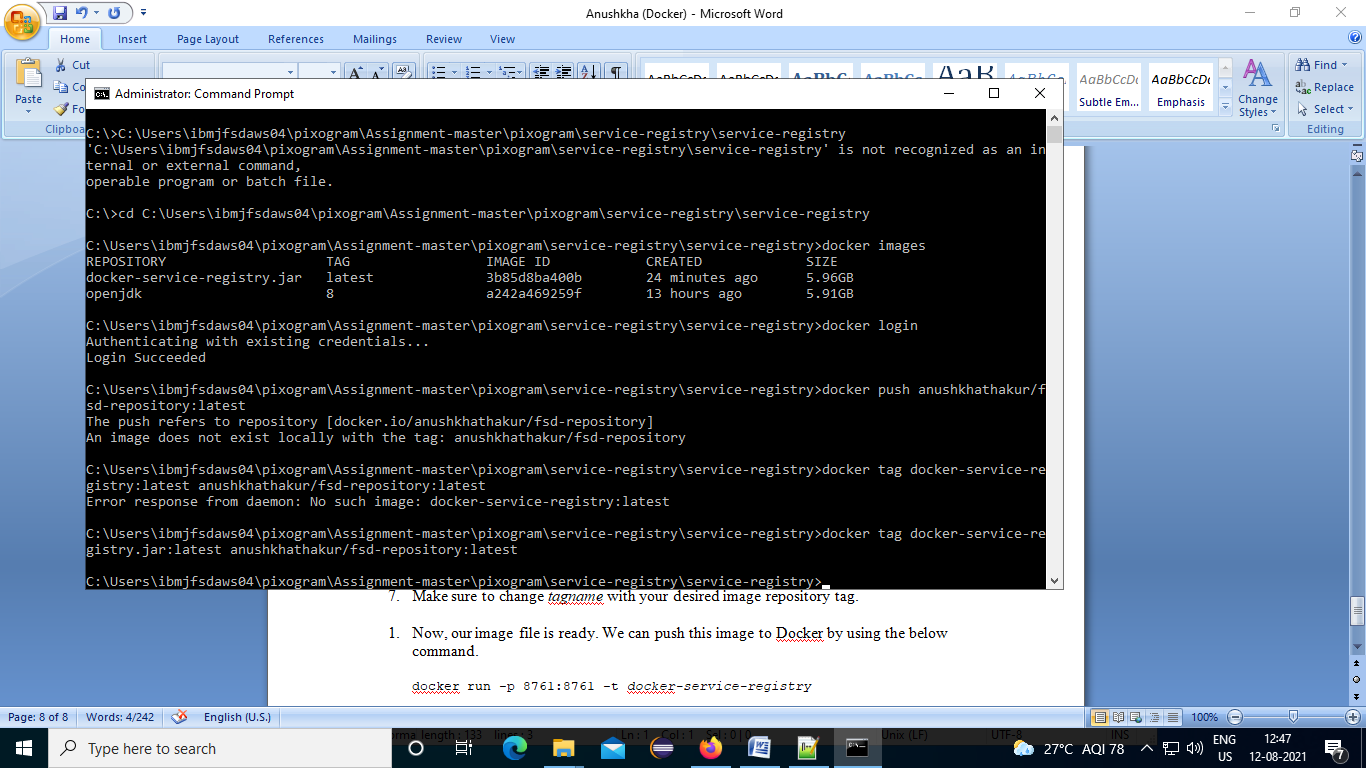










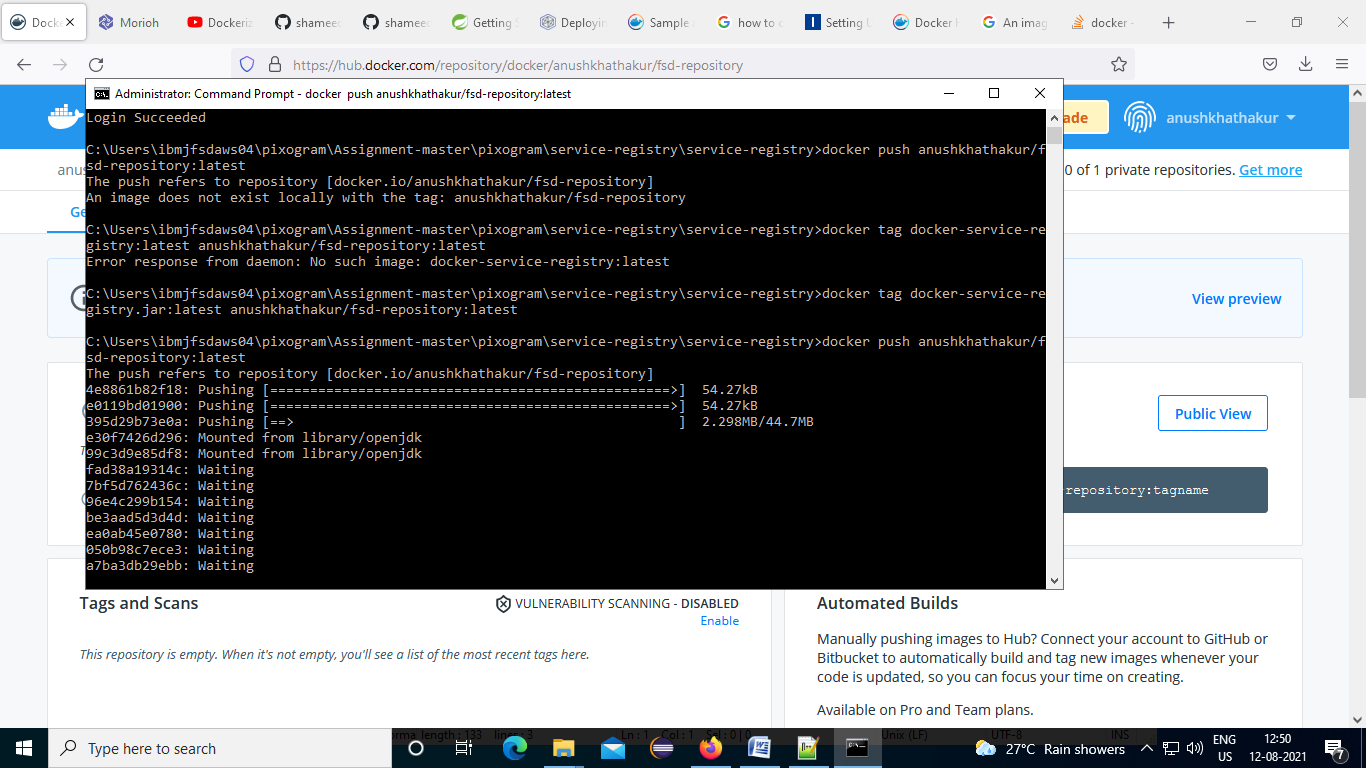


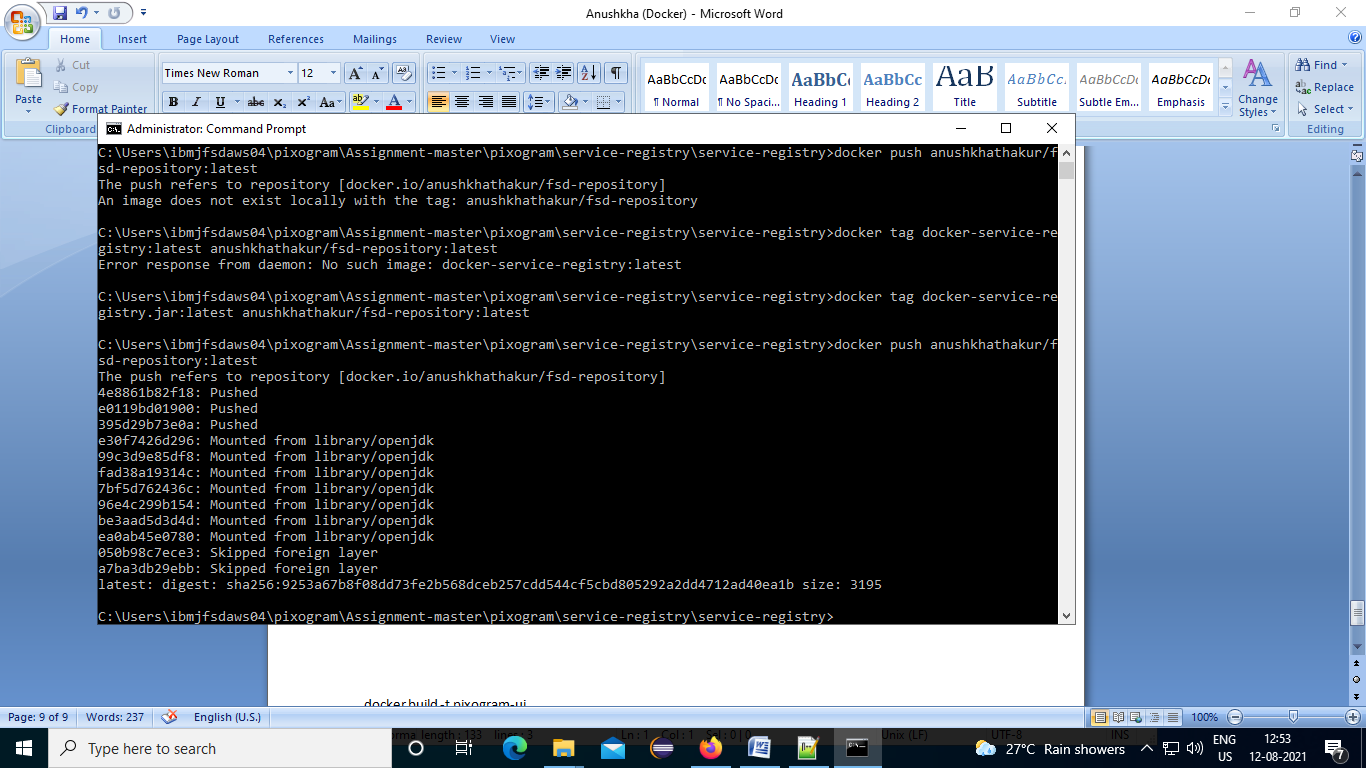
docker tag *docker-service-registry.jar*:latest anushkhathakur/fsd-repository:latest  
docker push anushkhathakur/fsd-repository:latest

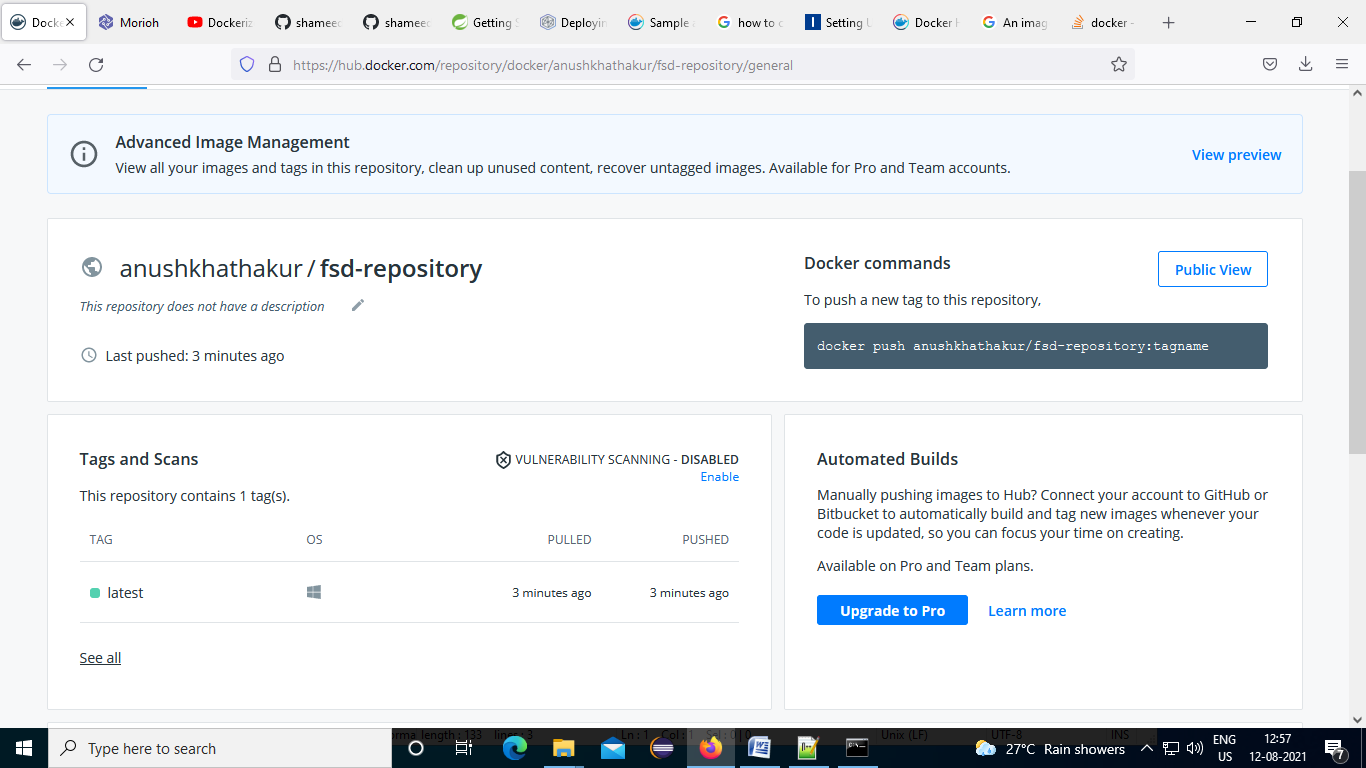
You can push a new image to this repository using the CLI

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

1. Make sure to change tagname with your desired image repository tag.

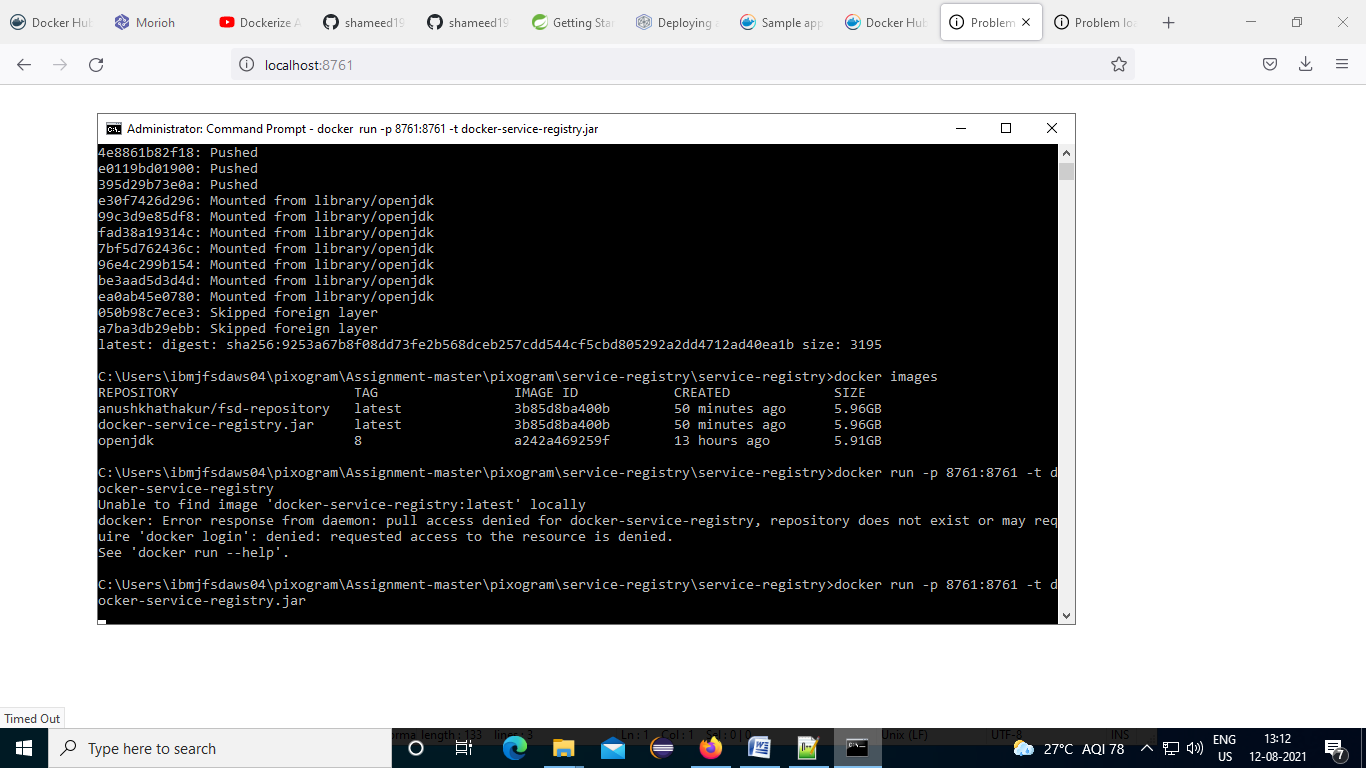


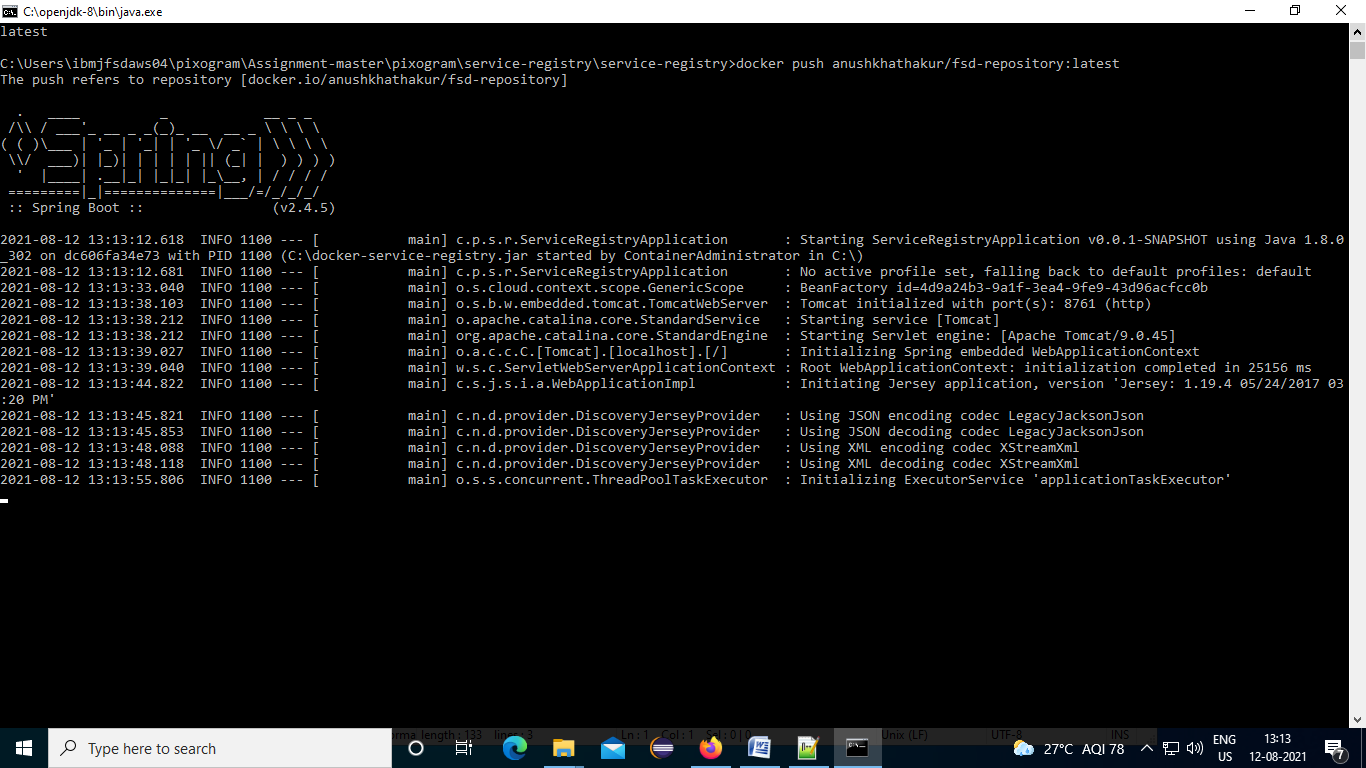


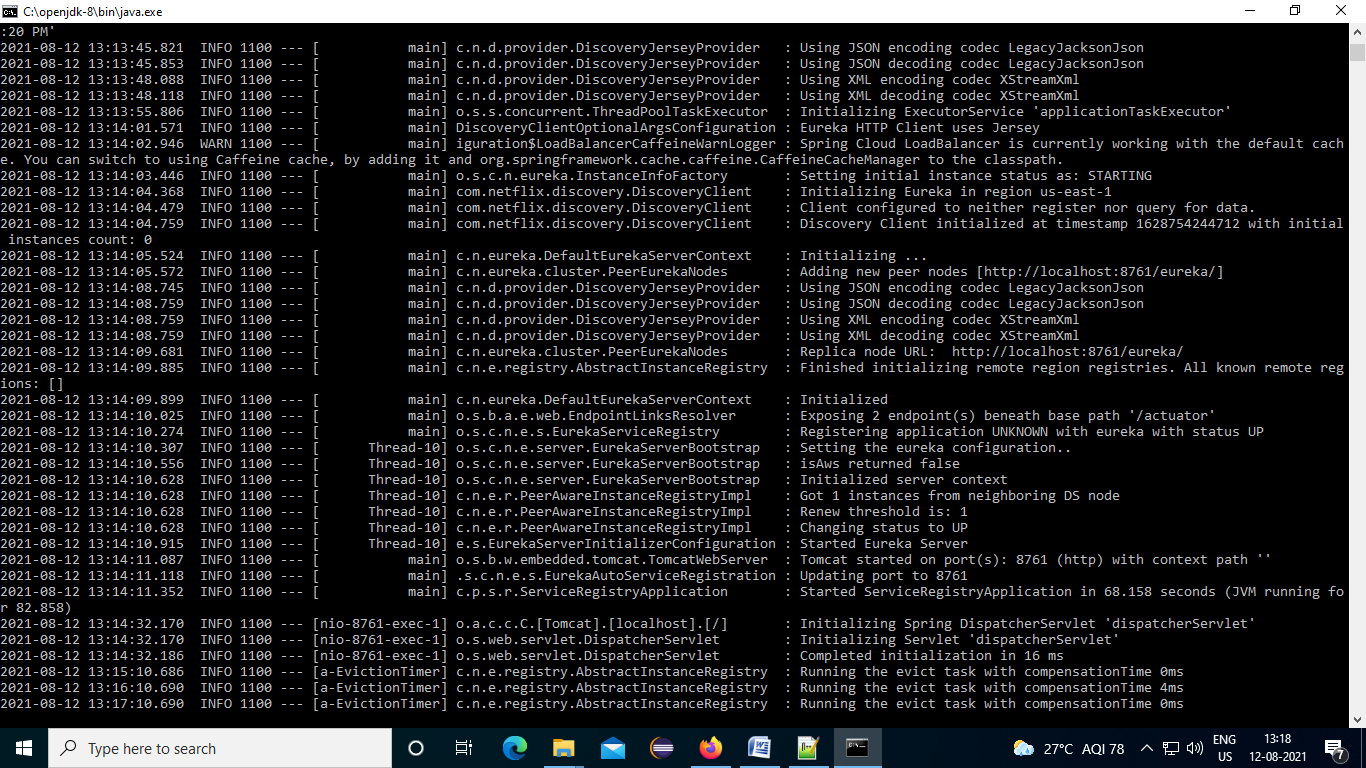


1. Now, our image file is ready. We can push this image to Docker by using the below command.

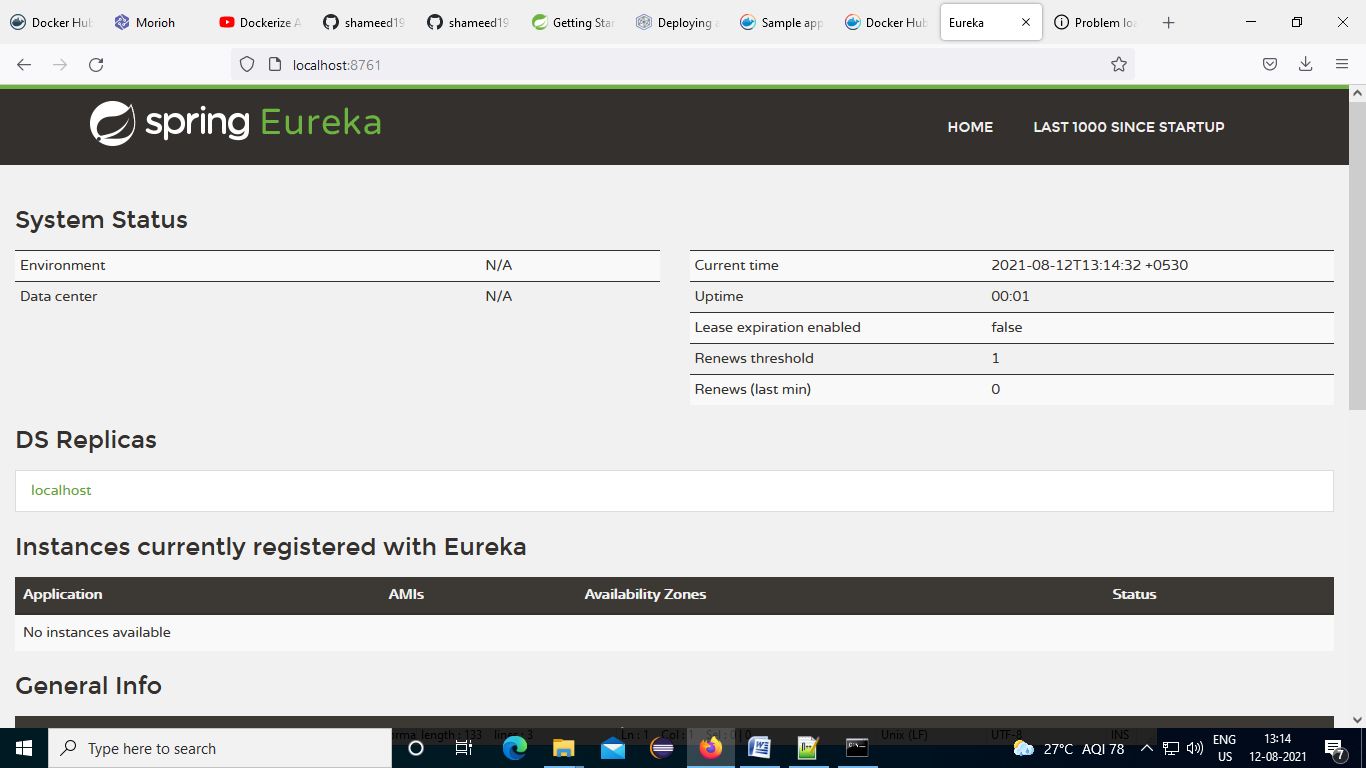
docker run -p 8761:8761 -t *docker-service-registry.jar*

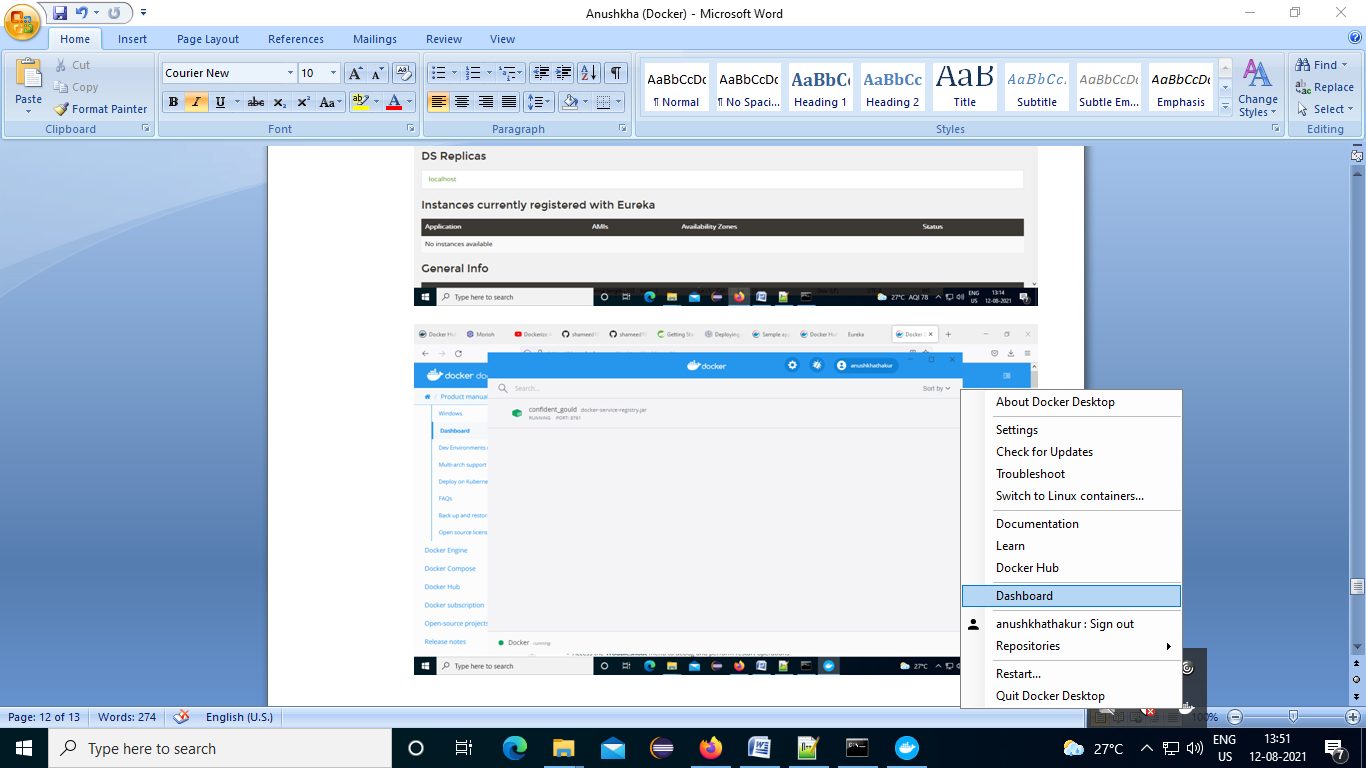
**

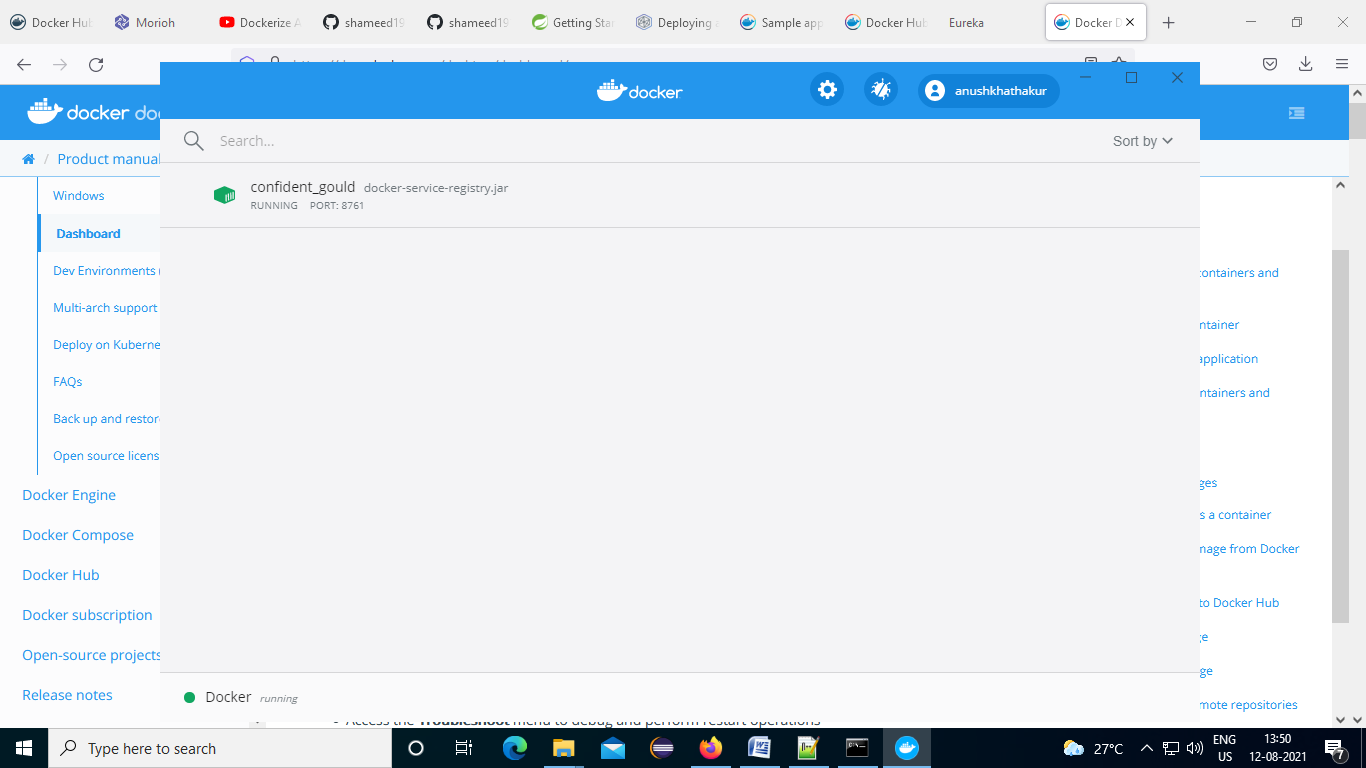
**

**

*http://localhost:8761/*

**

**

**

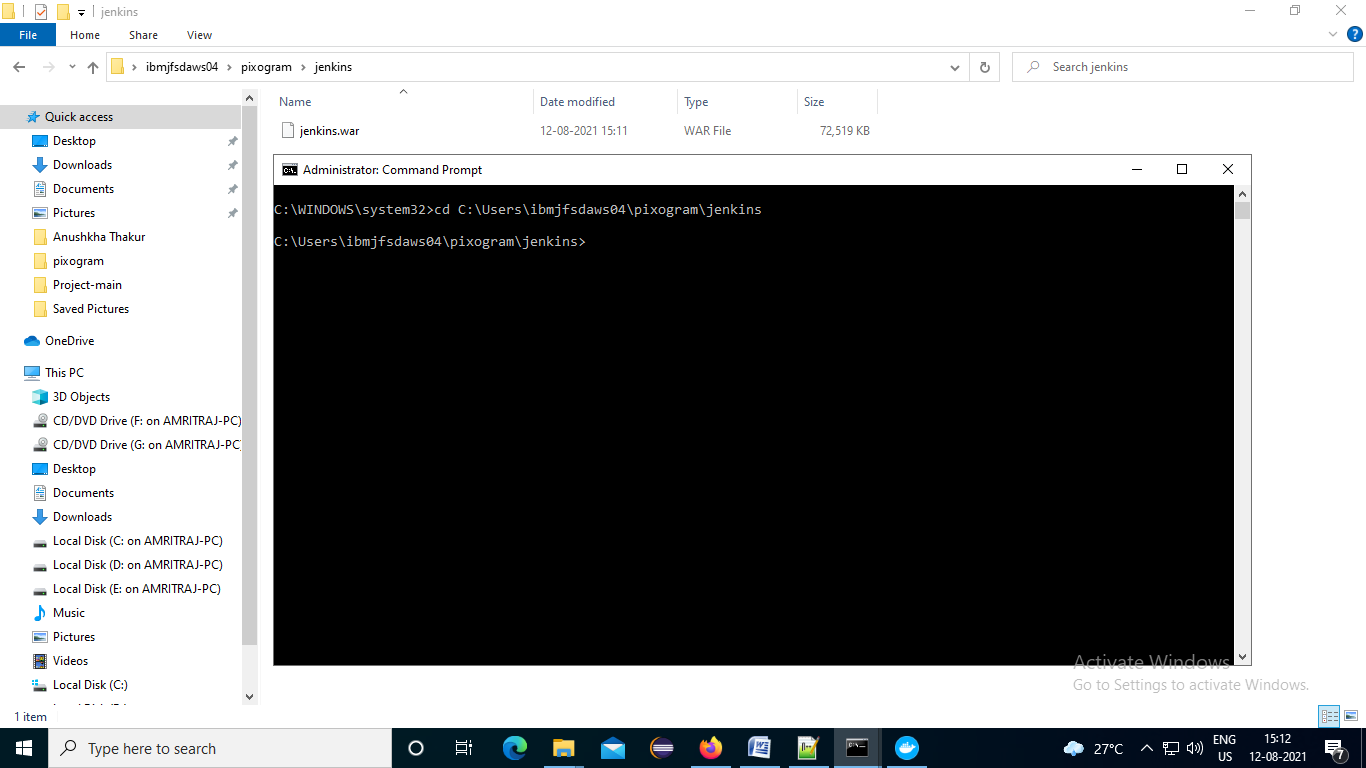
 Run docker build -t <your\_username>/my-private-repo . to build your Docker image.

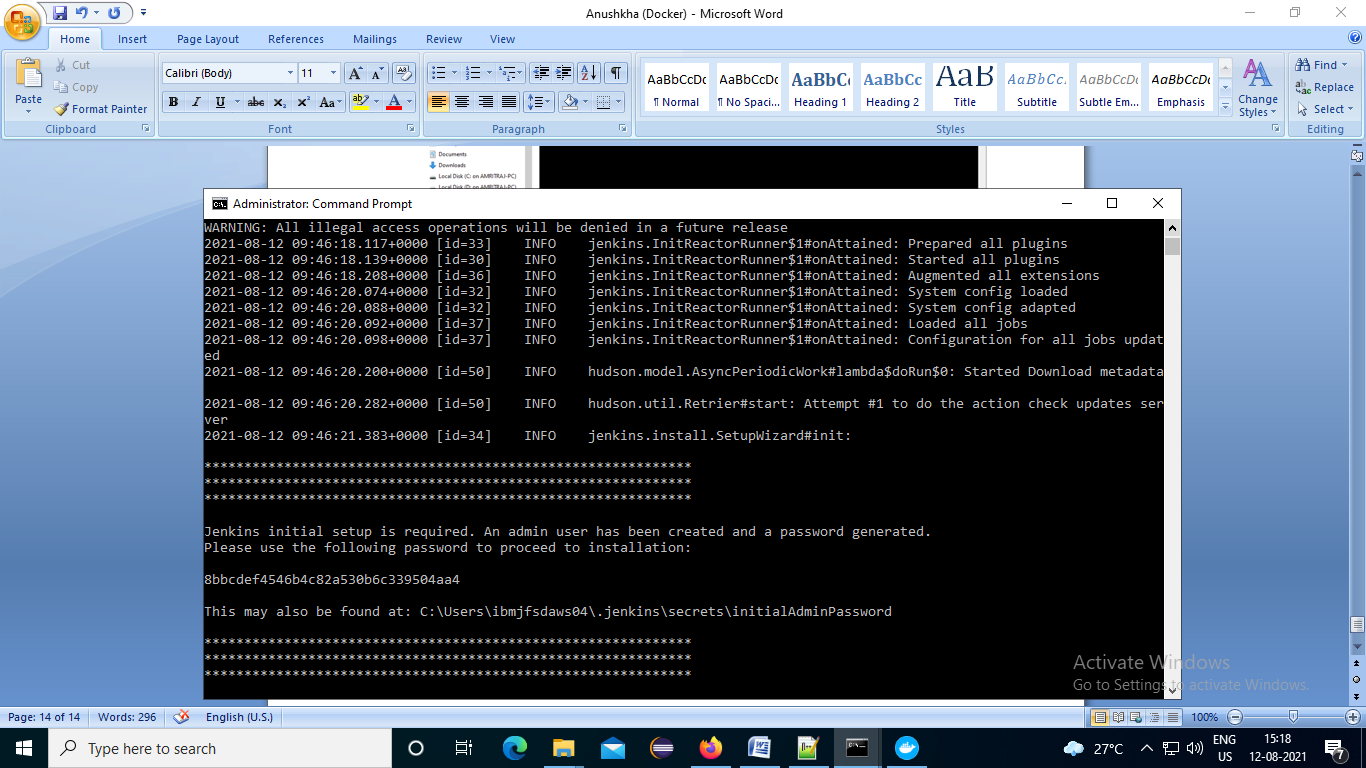
 Run docker run <your\_username>/my-private-repo to test your Docker image locally.

 Run docker push <your\_username>/my-private-repo to push your Docker image to Docker Hub.

# Jenkins

<http://mirrors.jenkins.io/war-stable/latest/jenkins.war>

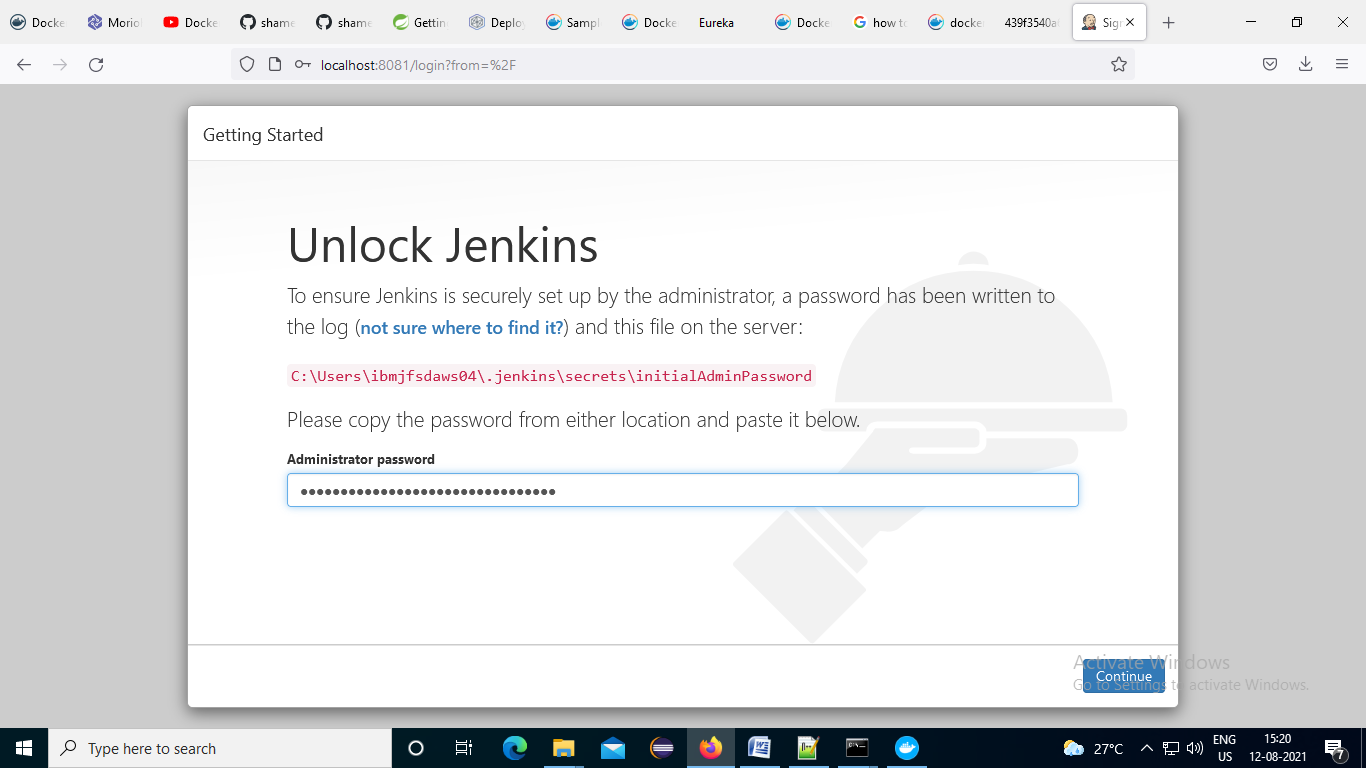
1. Download war file and place in folder
2. Command promt -> run as administrator
3. 
4. java -jar jenkins.war --httpPort=8081



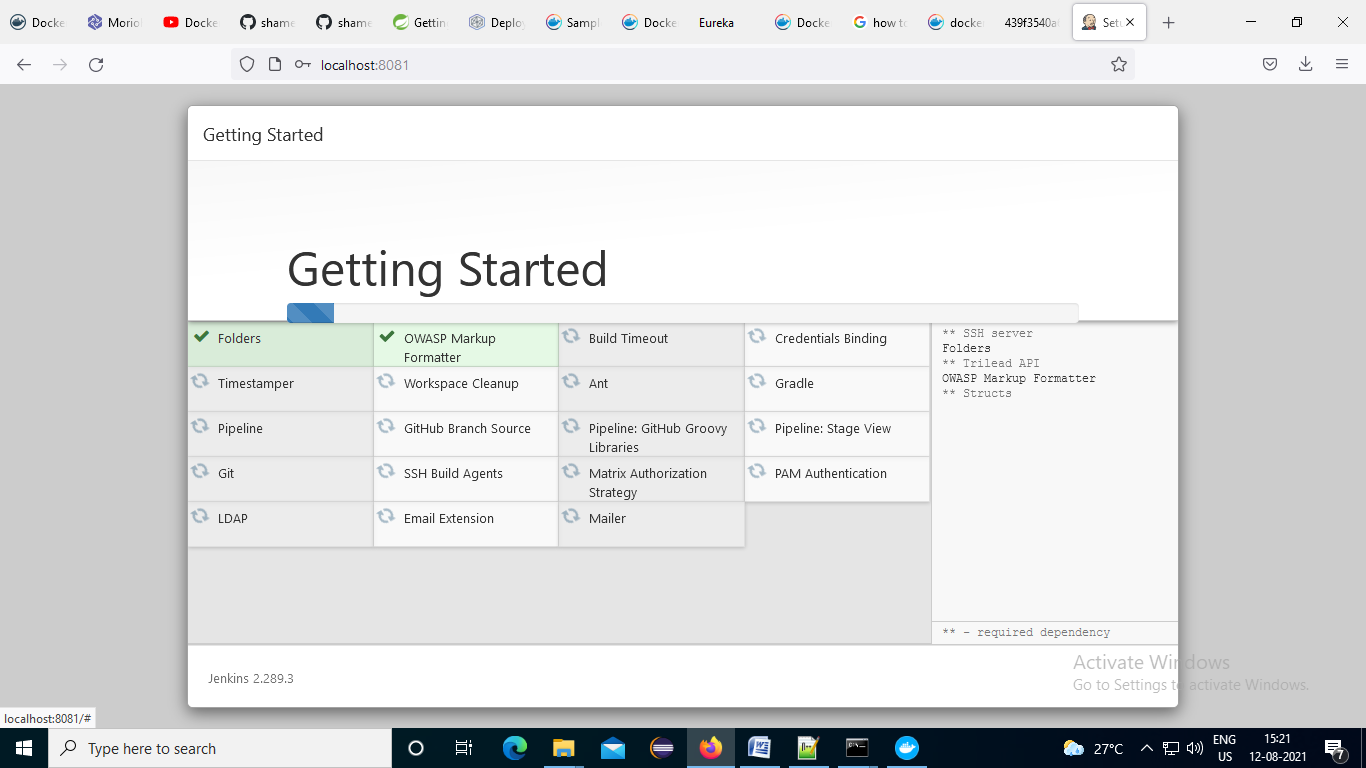
C:\Users\ibmjfsdaws04\.jenkins\secrets

Pass : 8bbcdef4546b4c82a530b6c339504aa4

Jenkin Url : http://localhost:8081



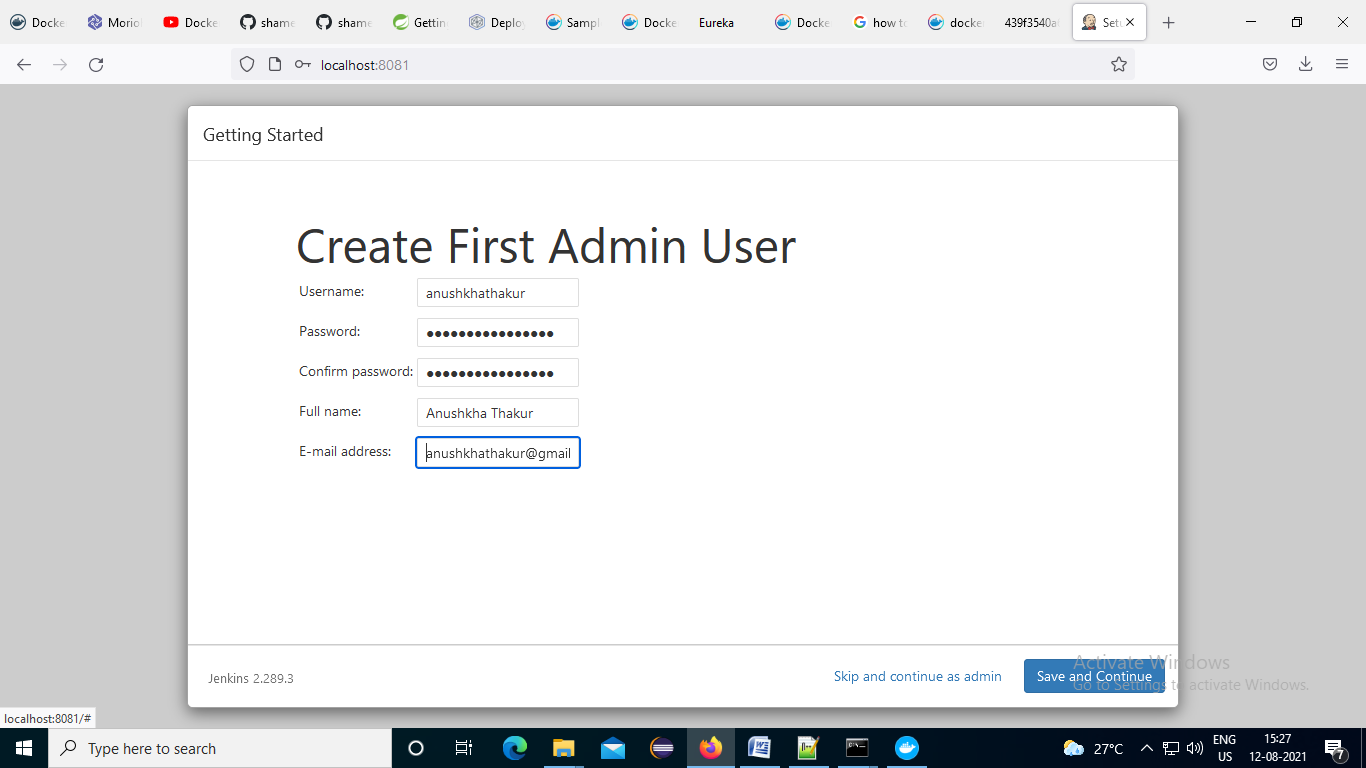
Install suggested plugin

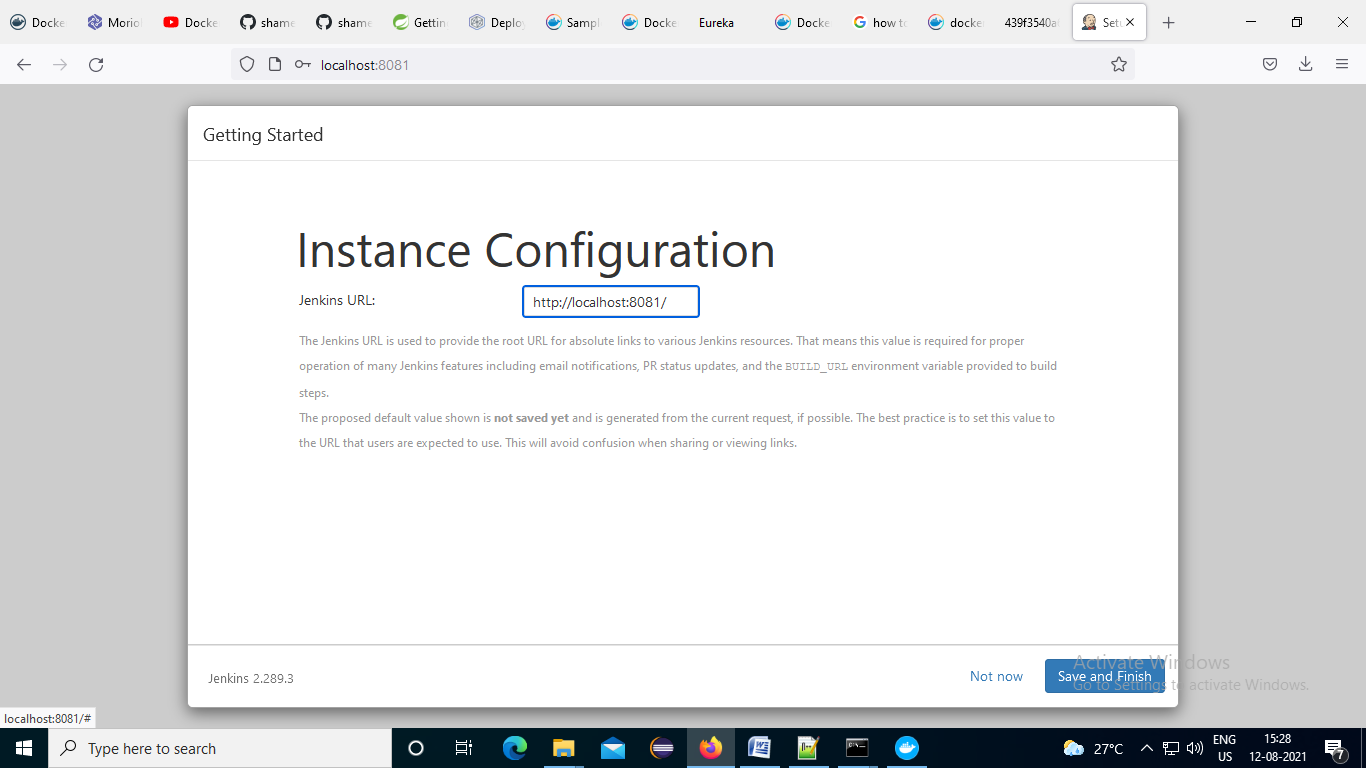


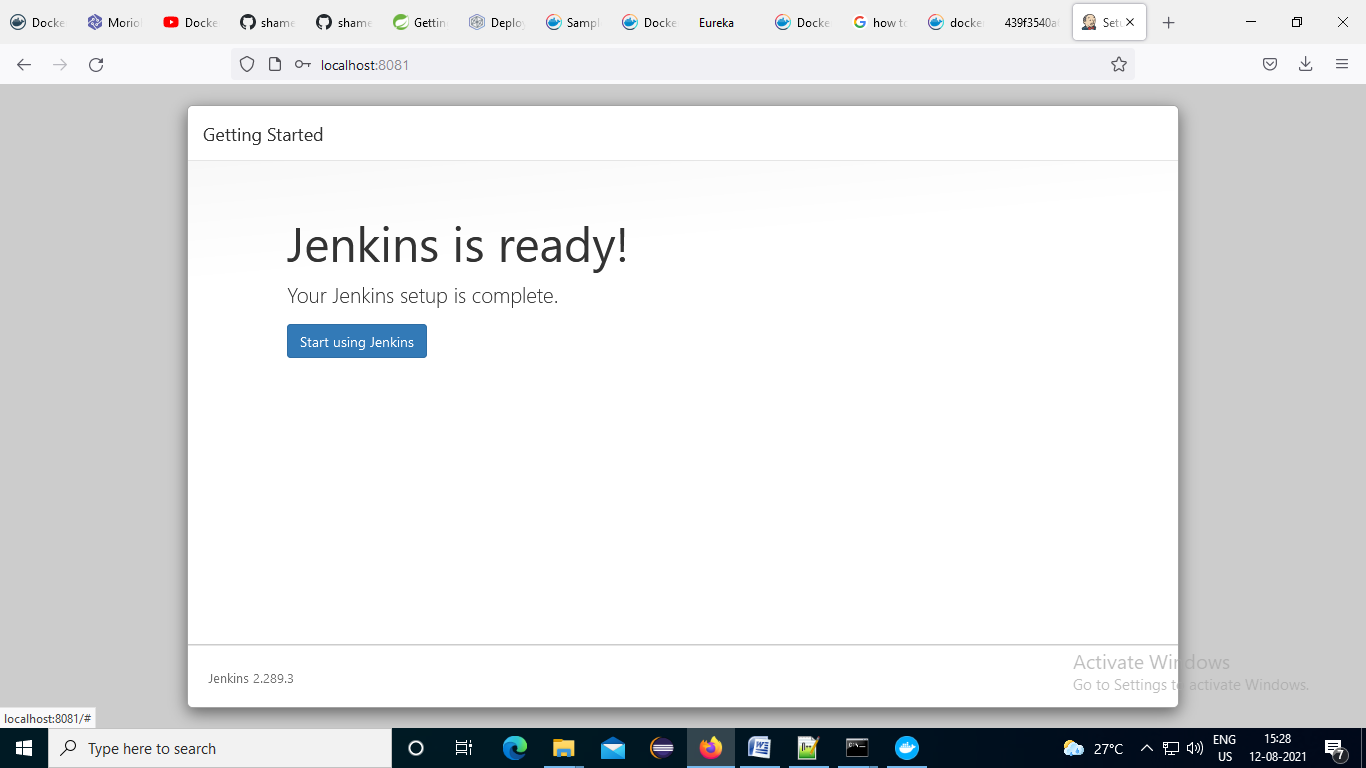
## Jenkins credentials

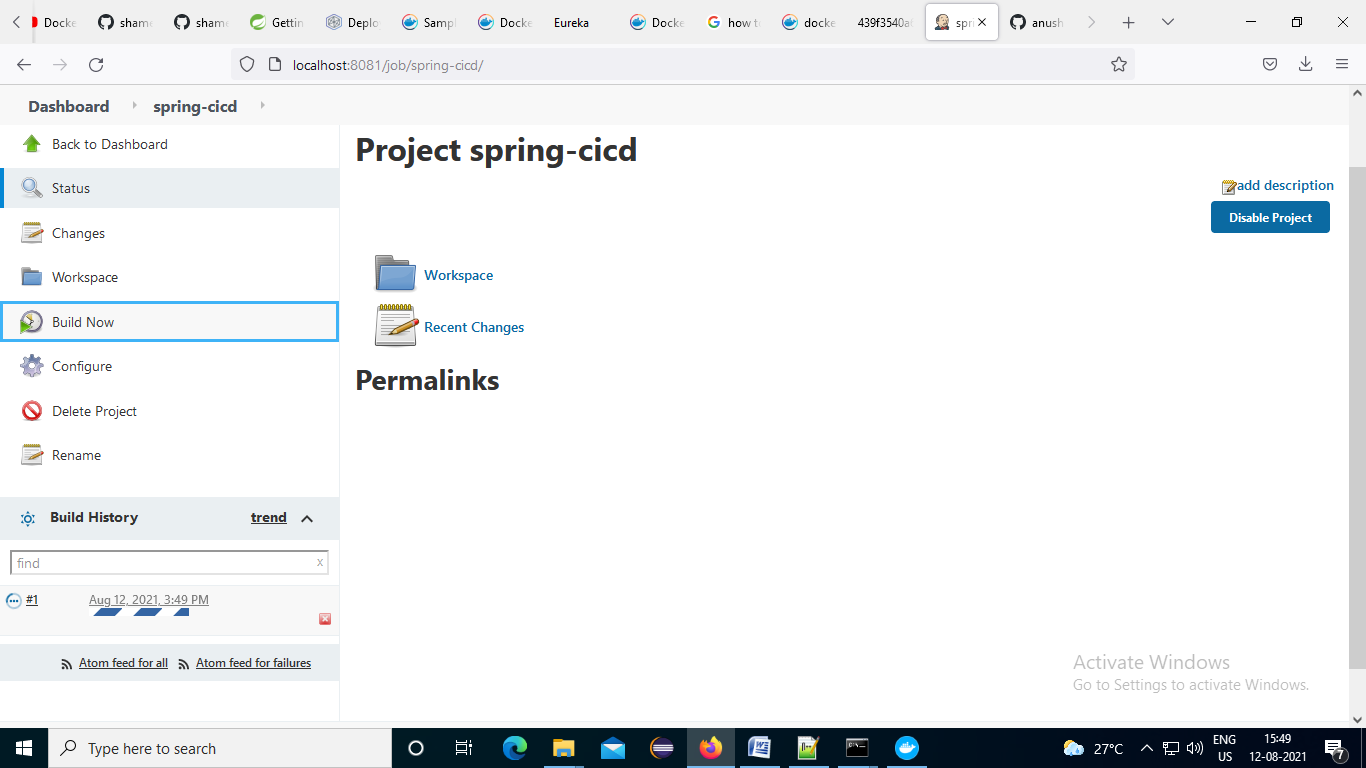
User name :anushkhathakur

Pass: Aryanthakur@1710

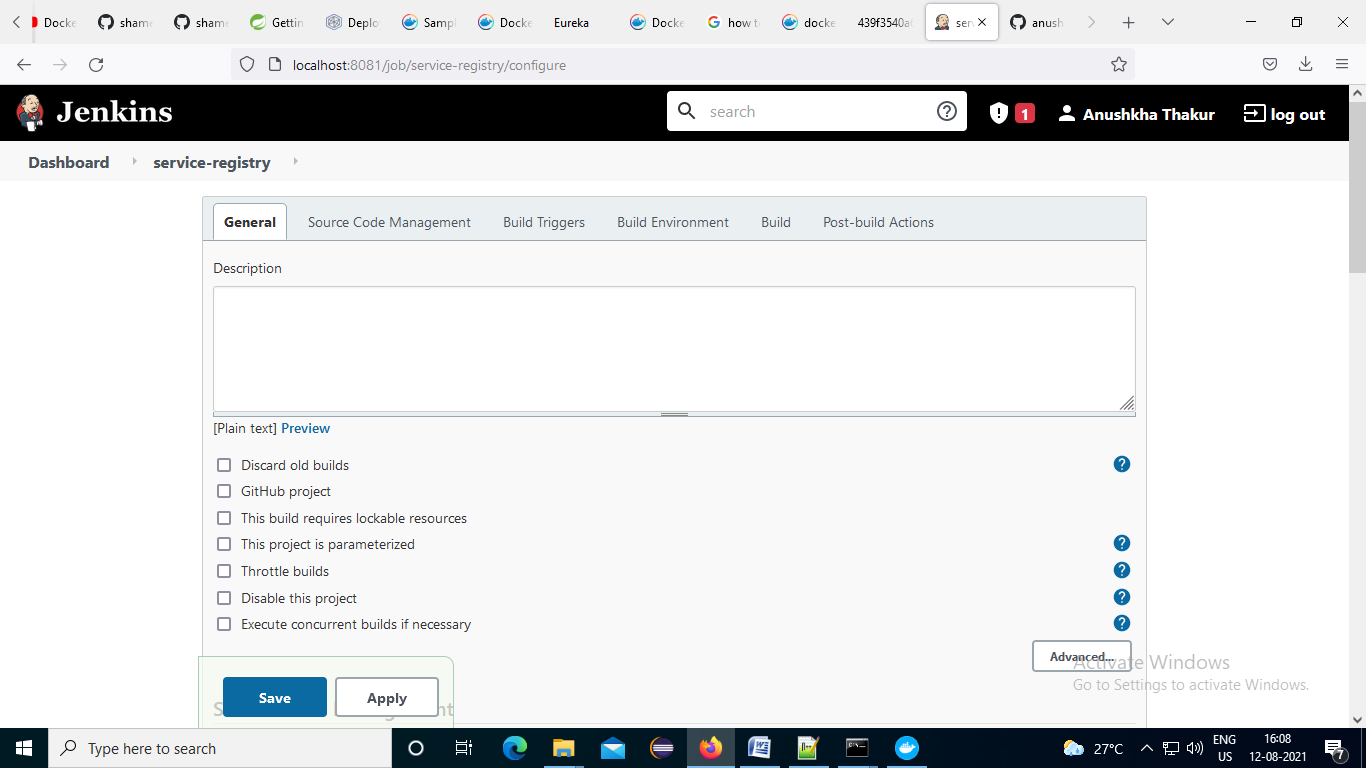


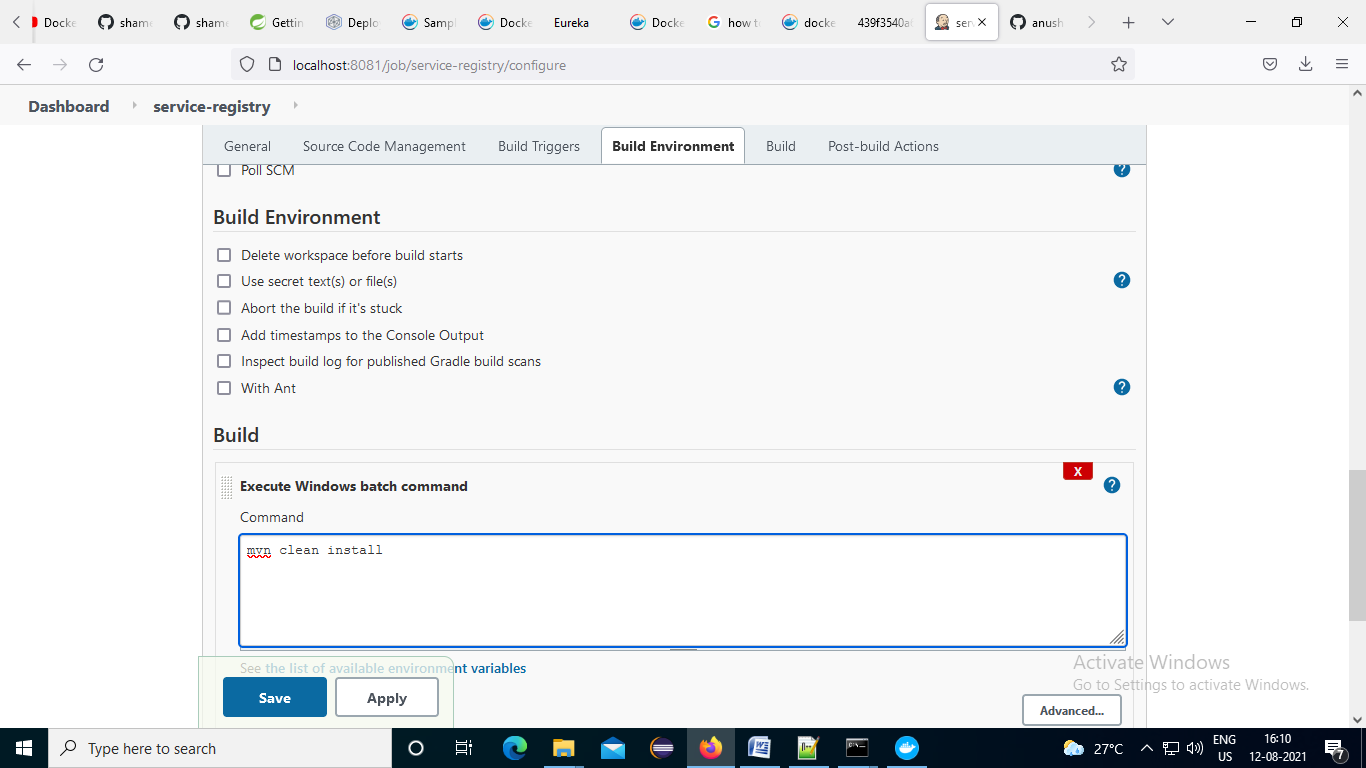


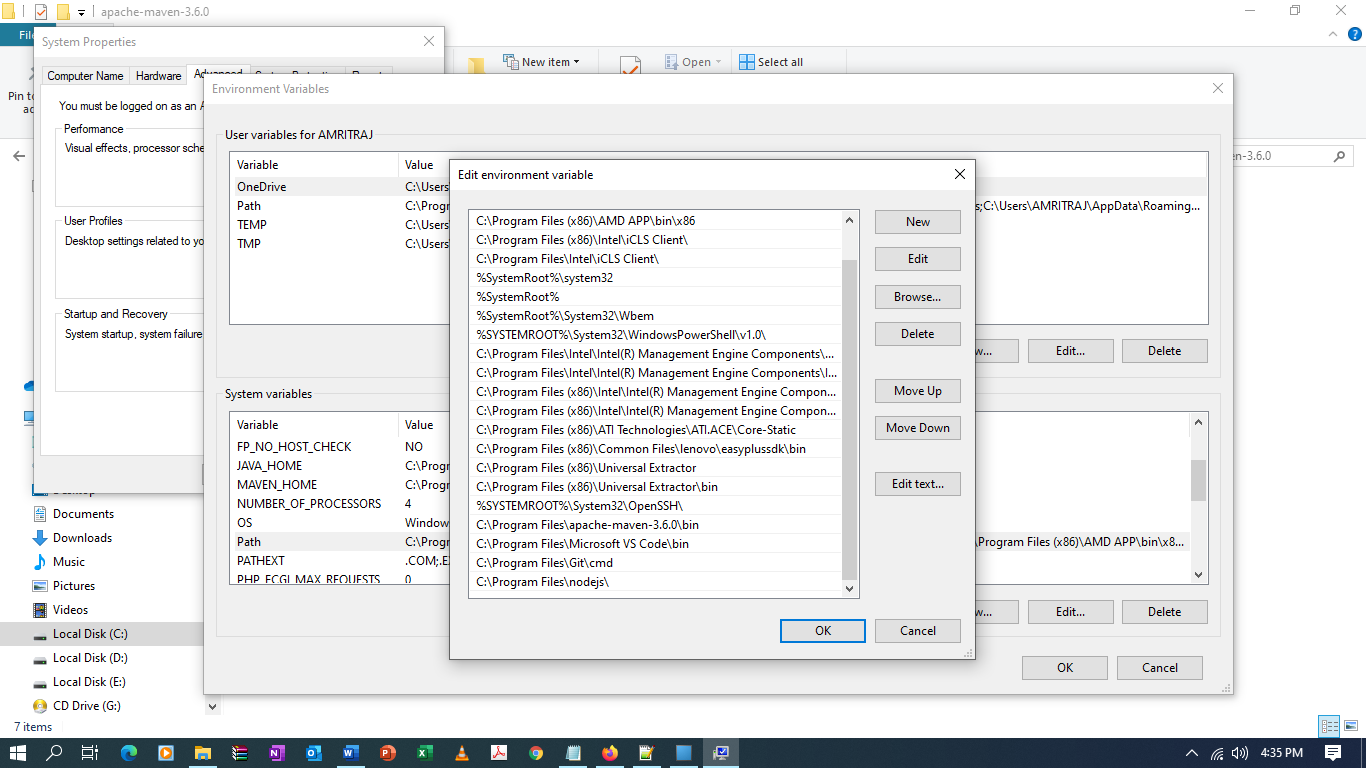


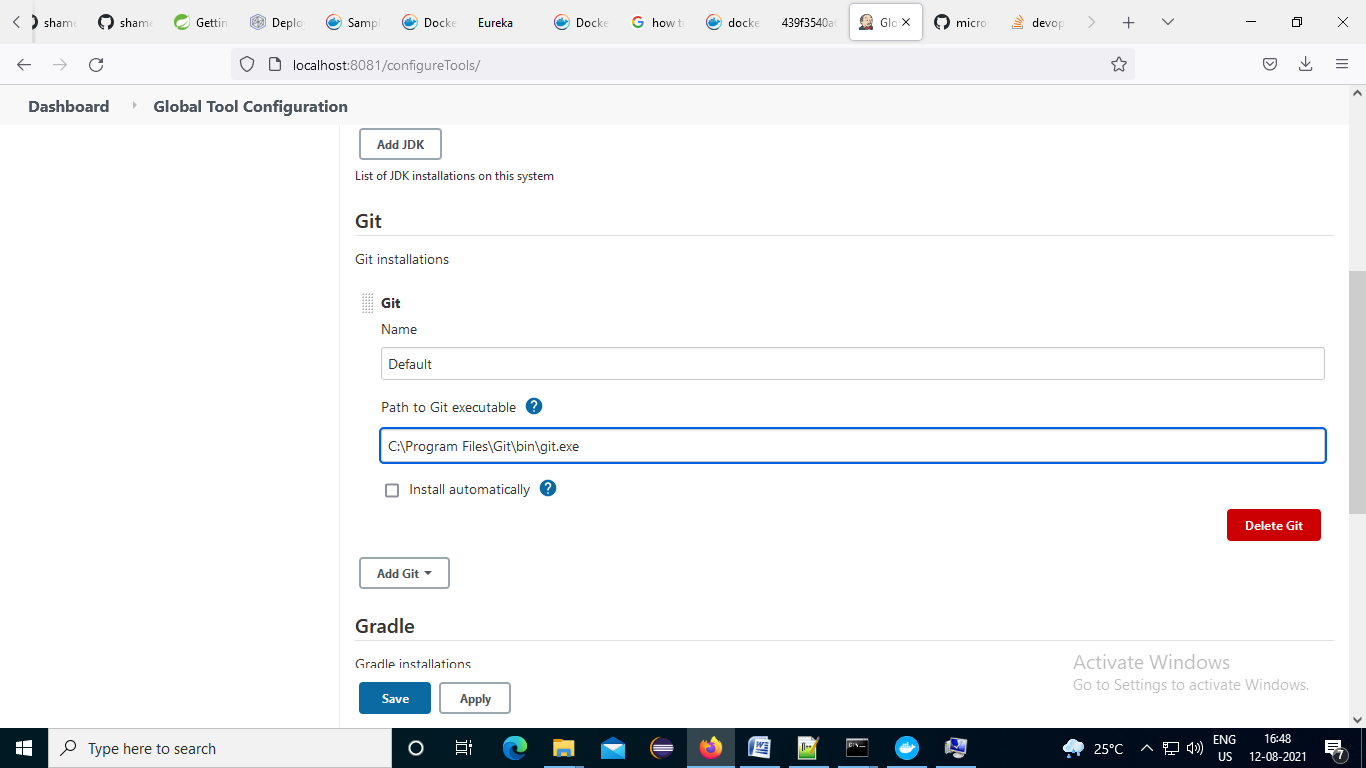


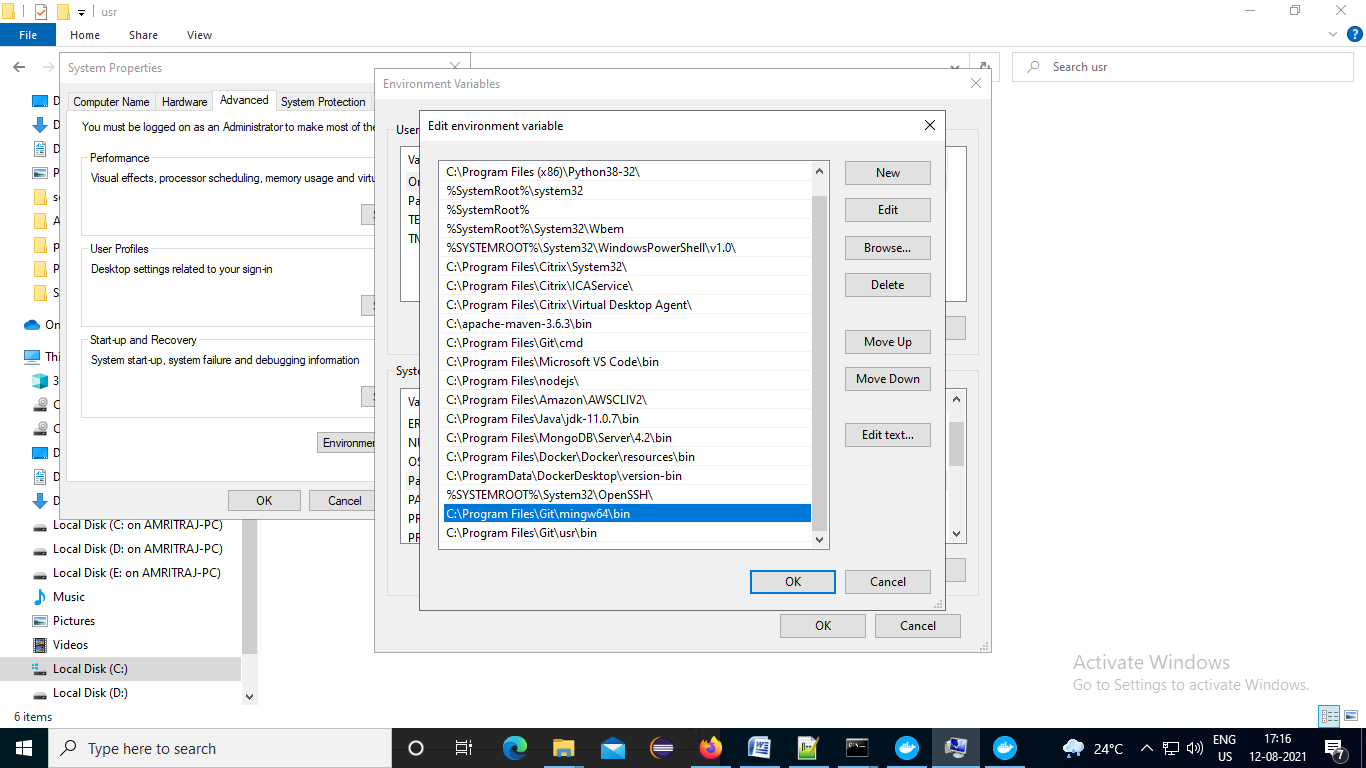
## Service-registery in jenkins job: freestyle

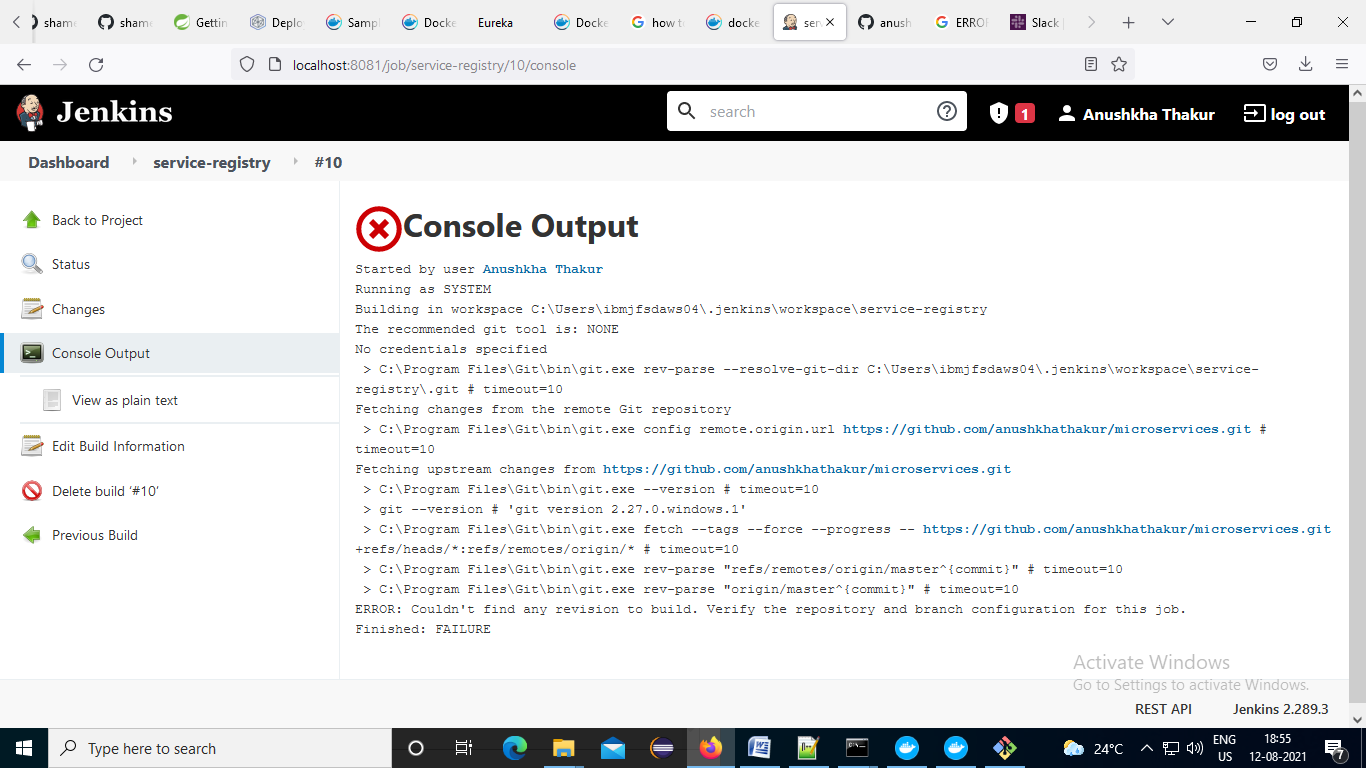


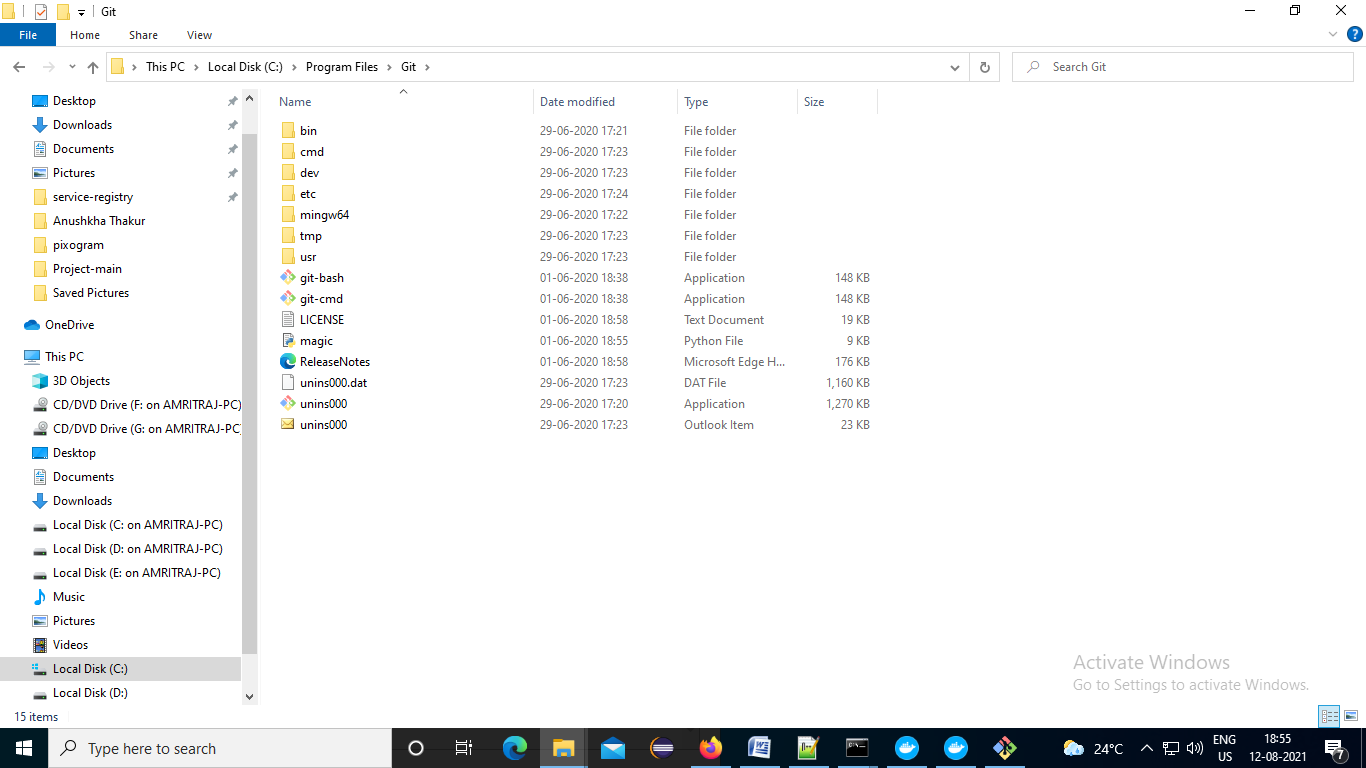




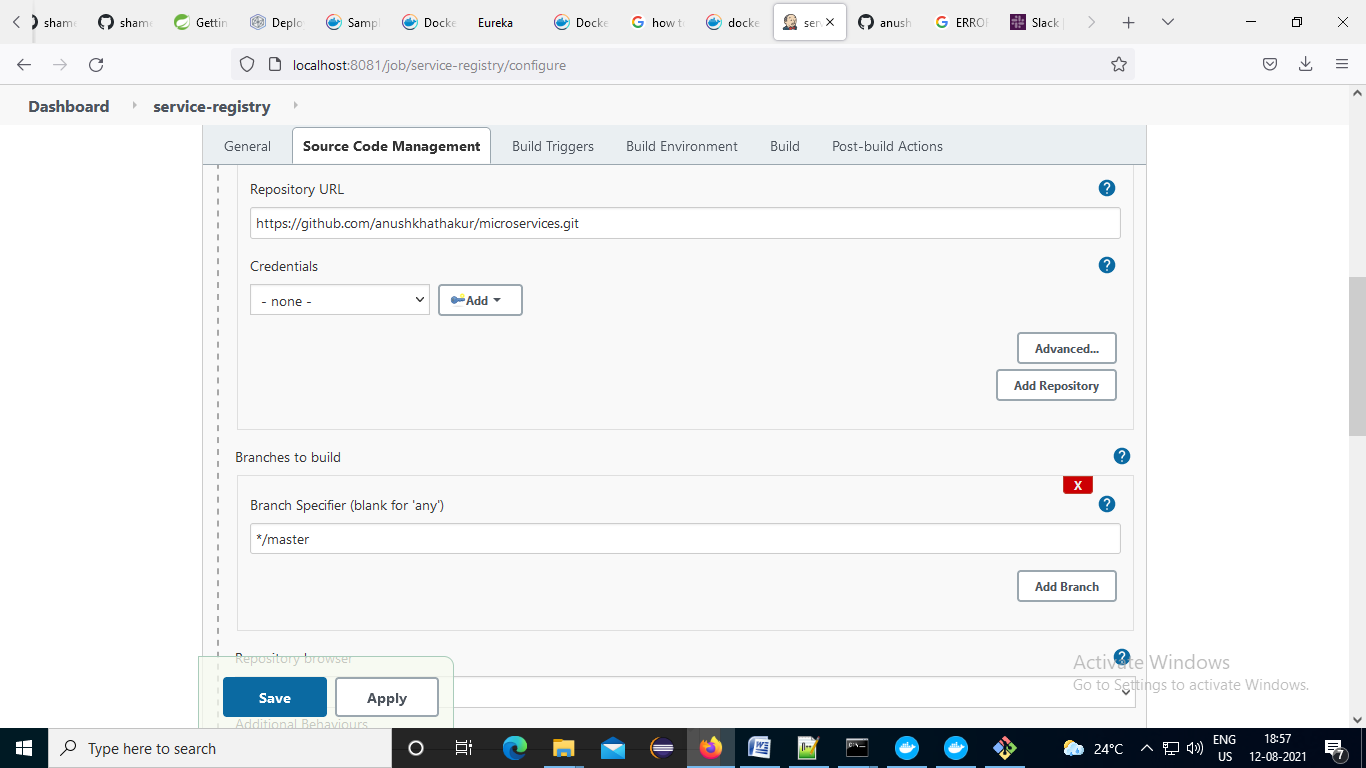


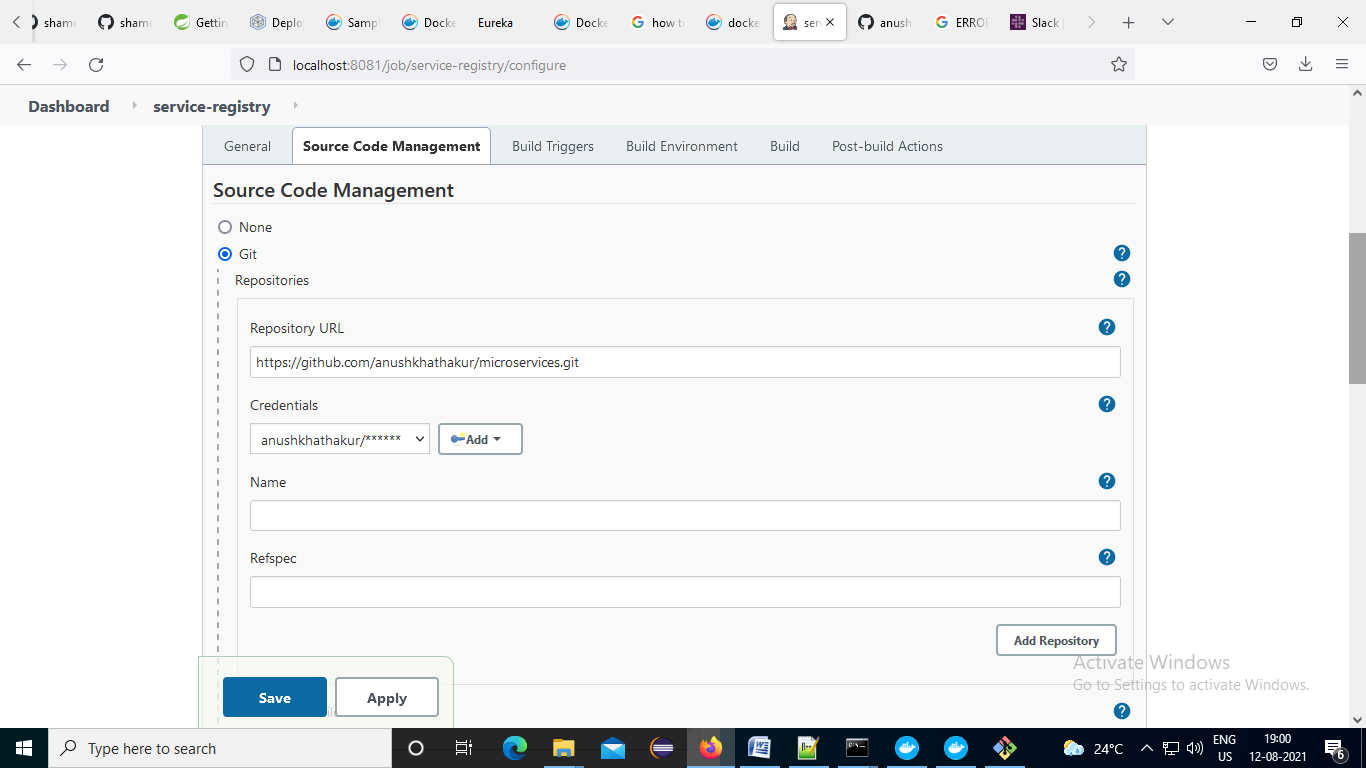


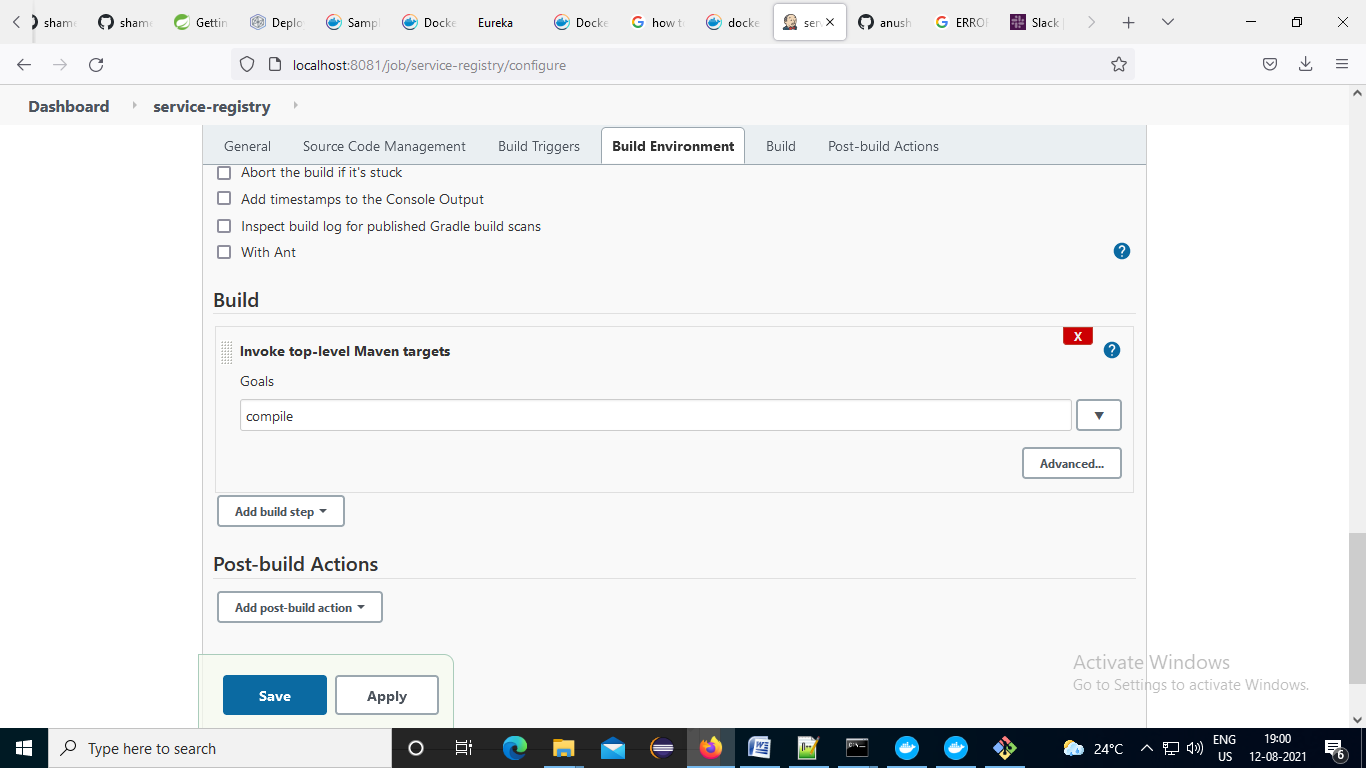


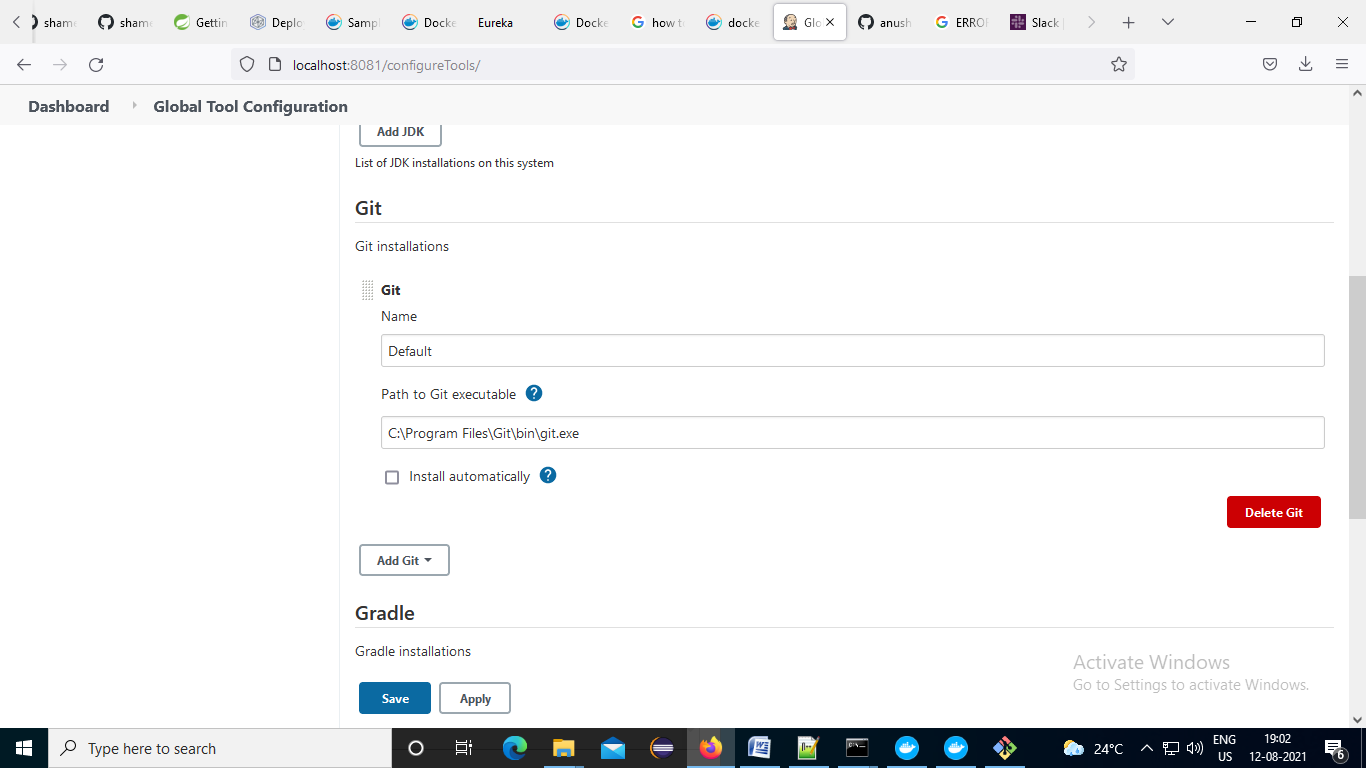


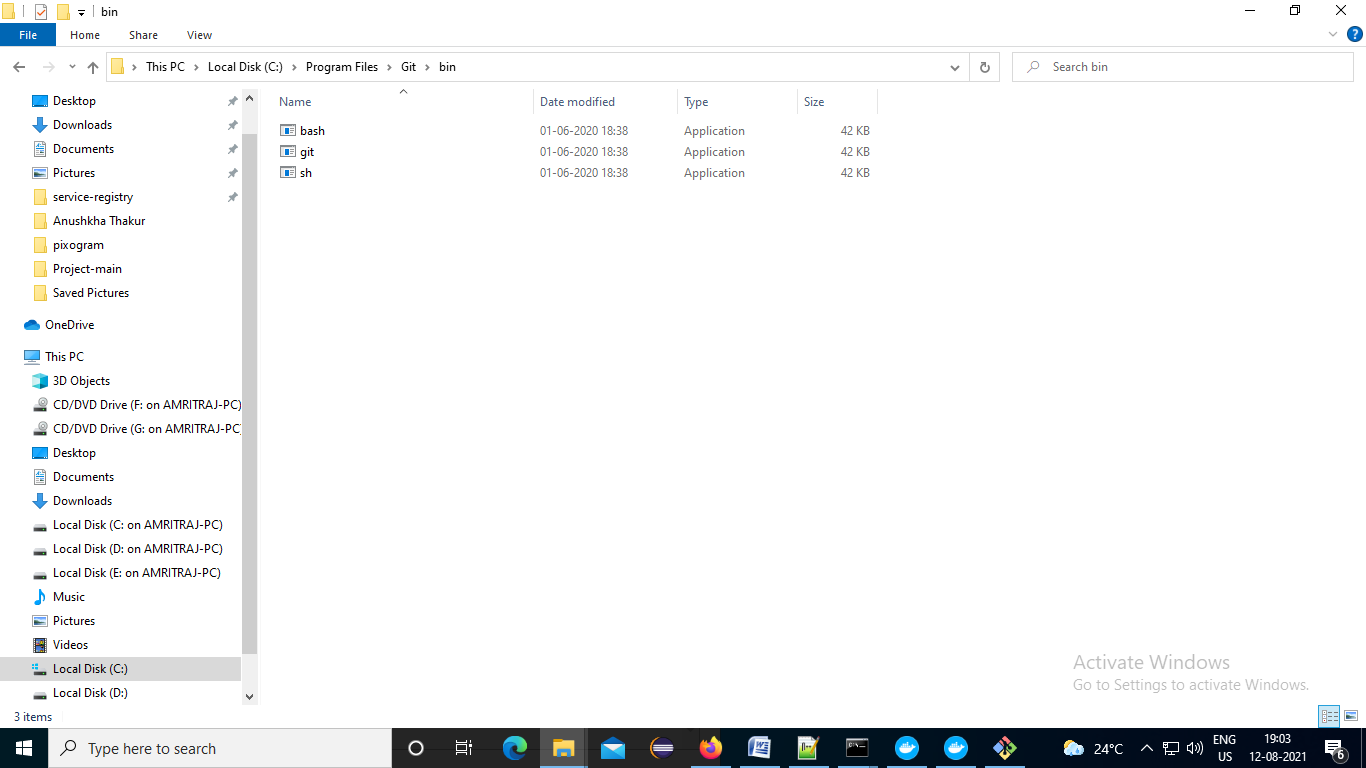
https://github.com/anushkhathakur/microservices.git



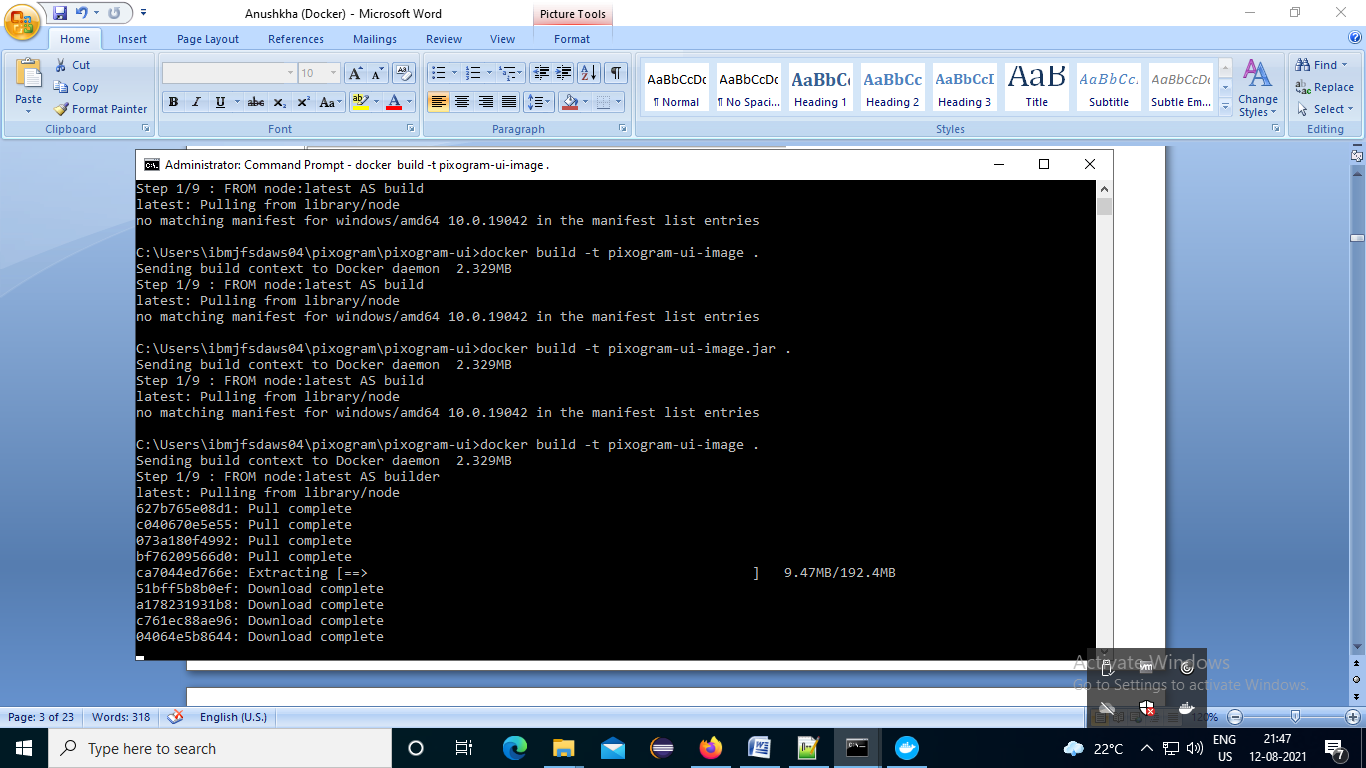








# Angular dockerization



# If you face error : [Docker: “no matching manifest for windows/amd64 in the manifest list entries”](https://stackoverflow.com/questions/48066994/docker-no-matching-manifest-for-windows-amd64-in-the-manifest-list-entries)

In Docker:

1. go to **Settings**
2. go to **Docker Engine**
3. change **experimental** to **true**
4. press **Apply and Restart**

