Project Title: Dockerizing Jenkins Pipeline

**Pushing Code From eclipse to GitHub:**

When any code changes are happened, then these changes are committed to local repository. From the GitHub Repository we have to take the latest code before pushing our code.once this is done we can commit our code changes to the github repository. In this scenario the code changes are pushed

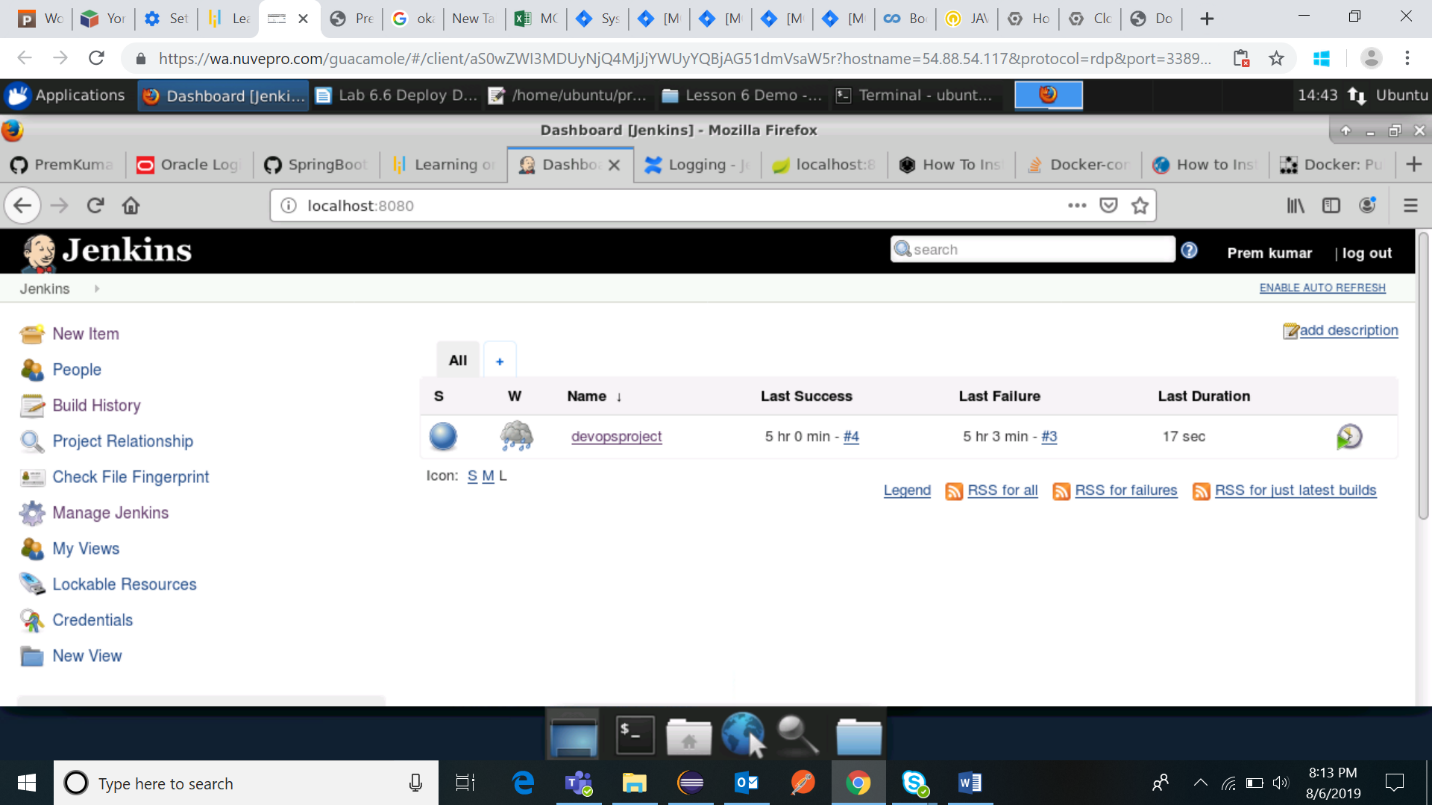
to <https://github.com/PremKumar1997/DevopsProject.git>

**Create first Jenkins job:**

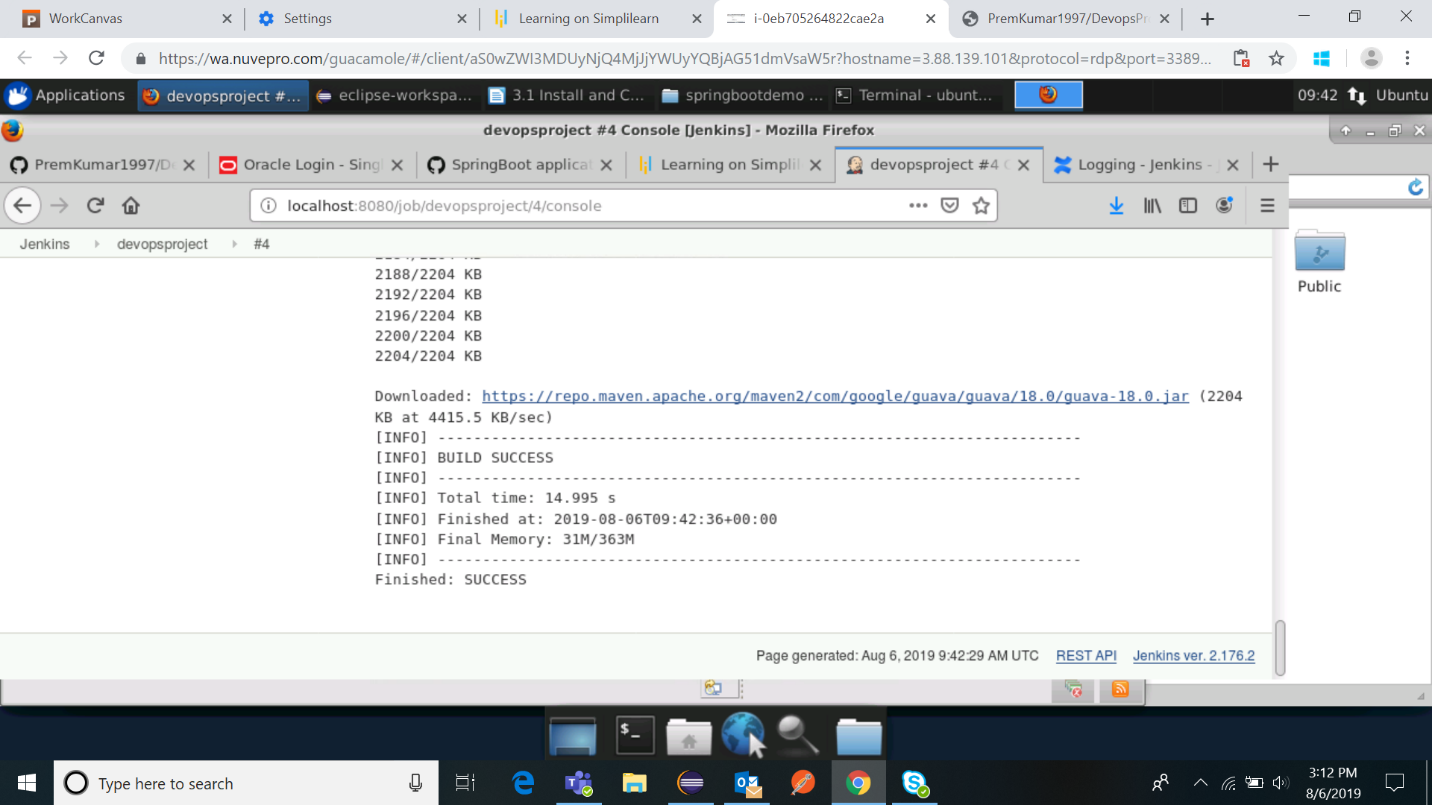
Login to Jenkins on <http://x.x.x.x:8080> using credentials: ***admin/password,***

and click on “**create new jobs**”.

Create new jobs will take us to create a new project under “**free style project**” category.



Now, set up the required configuration for github access and credentials for it. Set up the **JDK path** in it to execute the mvn **clean package** command. After that go to project and click on “**build now**”. It will build the project, upon build success we can proceed for further steps.



**Dockerfile:**

**Dockerfile** contains the command and instructions for building the **docker image** from the project.  The contents of the **Dockerfile** related to this project, can be given as follows.

|  |
| --- |
| FROM java:8  LABEL maintainer=“Prem.Kumar4@mindtree.com”  WORKDIR /app  COPY target/springbootsample-0.0.1-SNAPSHOT.jar /app/springbootsample-0.0.1-SNAPSHOT.jar  ENTRYPOINT ["java","-jar","springbootsample-0.0.1-SNAPSHOT.jar"] |

**YML File (docker-compose.yml):**

Create a docker-compose.yml file.

**touch docker-compose.yml**

**vi docker-compose.yml**

Add the content mentioned below in the docker-compose.yml file.

|  |
| --- |
| version: '3'  services:  spring-boot-rest-api-app:  image: springboot-docker-compose-app  build:  context: ./  dockerfile: Dockerfile  ports:  - 8080:8080  volumes:  - /data/springboot-docker-compose-app |

Look at the Docker Compose YAML file.

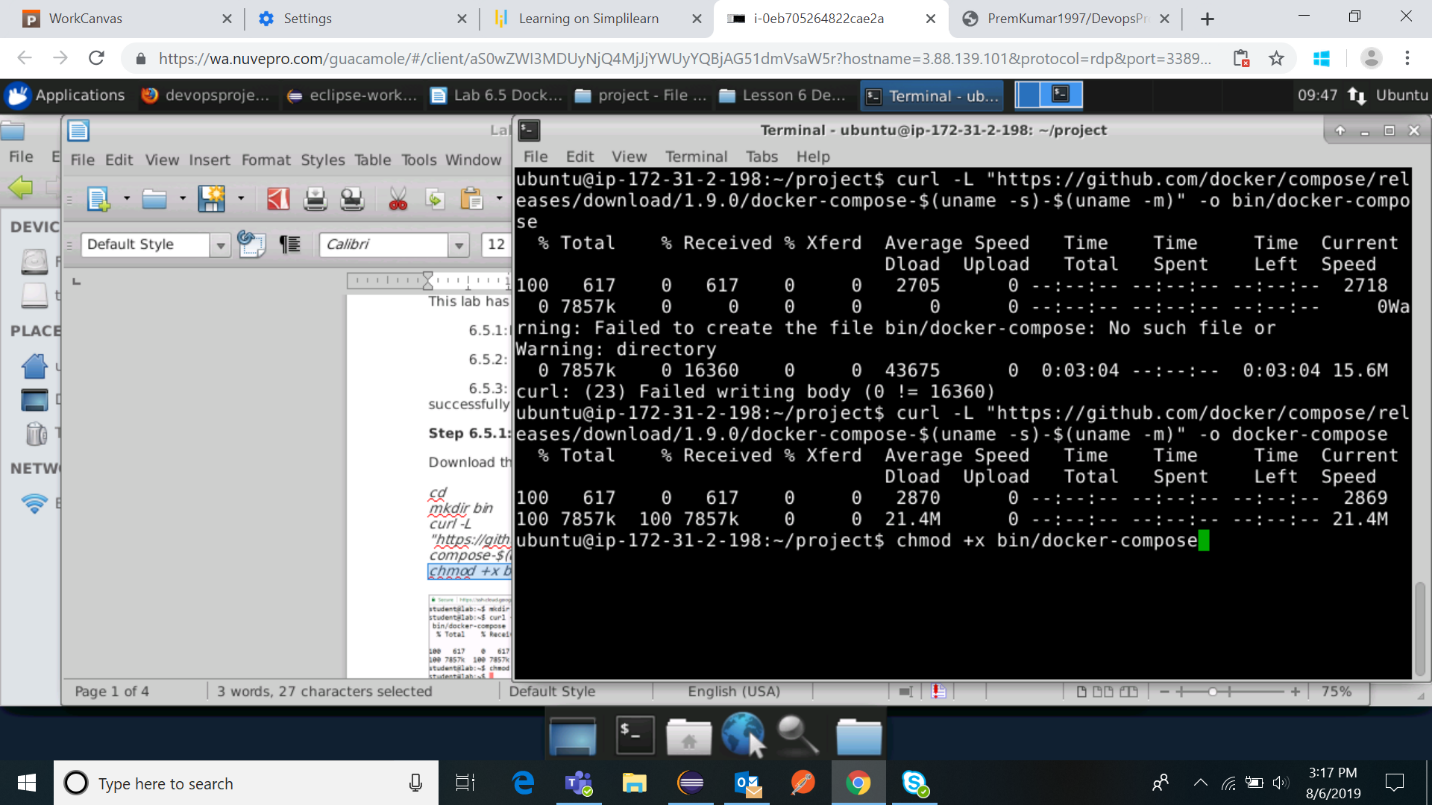
*cat docker-compose.yml*

**Docker Compose:** In order to download docker-compose type the below command.

***curl -L "https://github.com/docker/compose/releases/download/1.9.0/docker-compose-$(uname -s)-$(uname -m)" -o docker-compose***

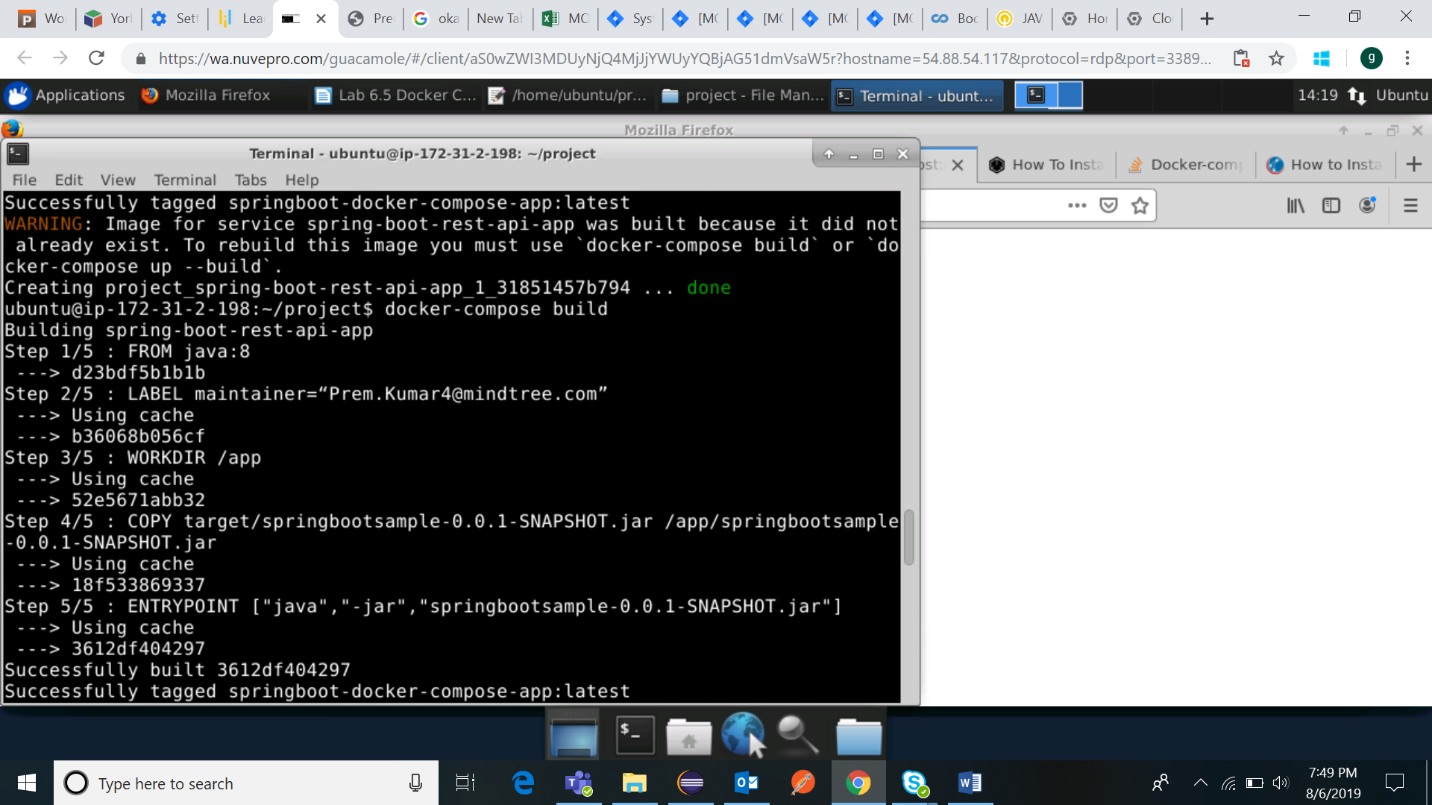
For clearing the permissions execute the below command

***chmod +x docker-compose***



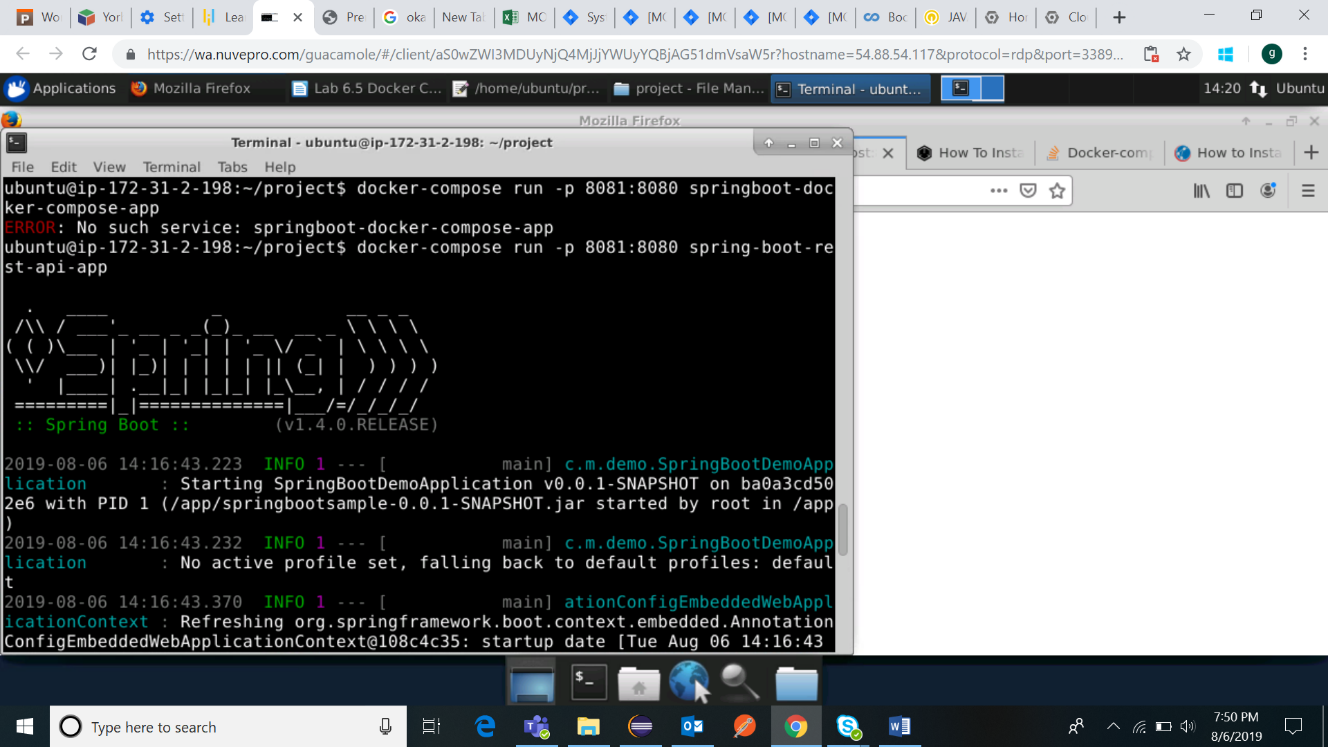
**Docker Build:** Now Build the Docker by building the docker-compose with the command

***docker-compose build***



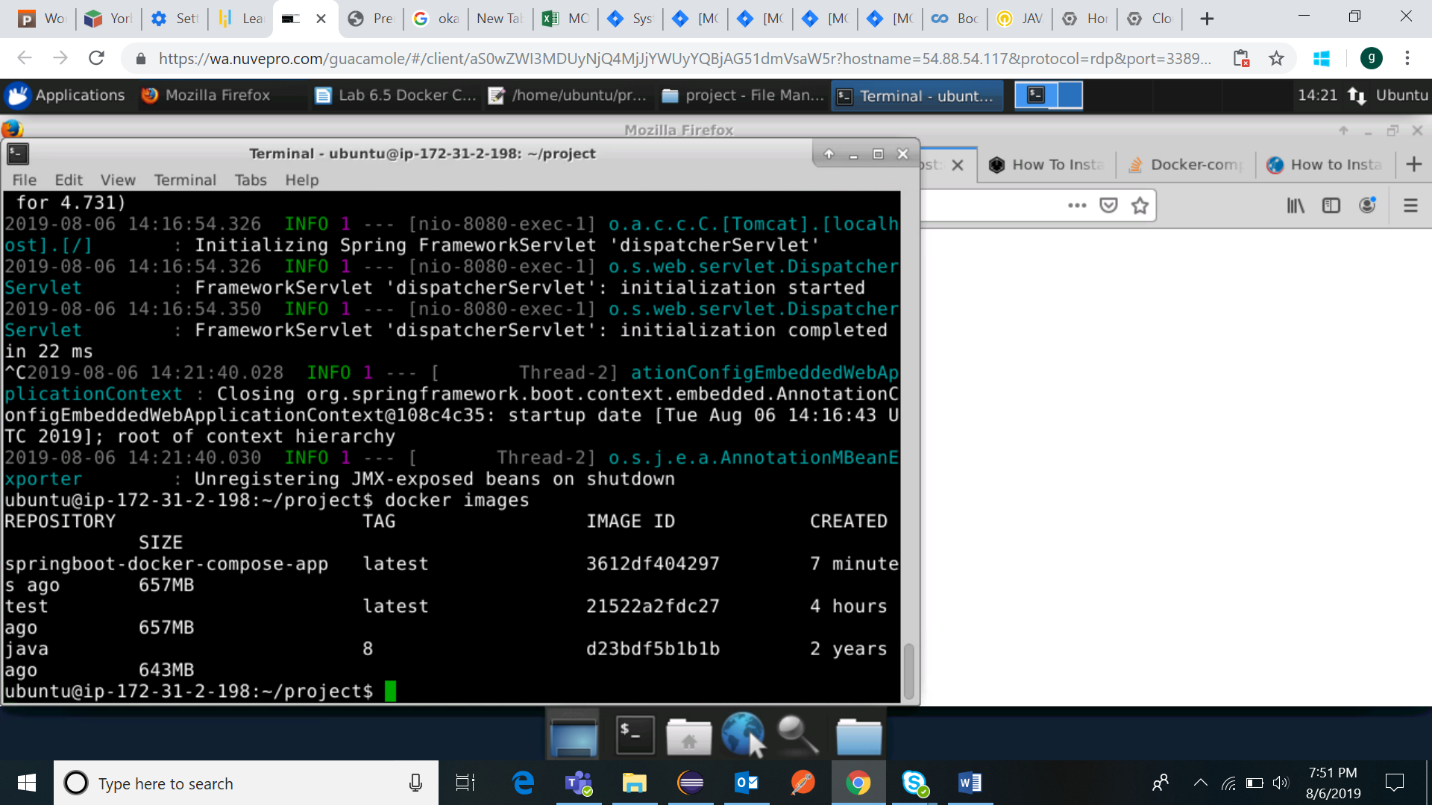
**Docker Run**: Now Run the image after successful Docker build. Run the application by typing the command

**docker-compose run –p 8081:8080 spring-boot-rest-api-app**

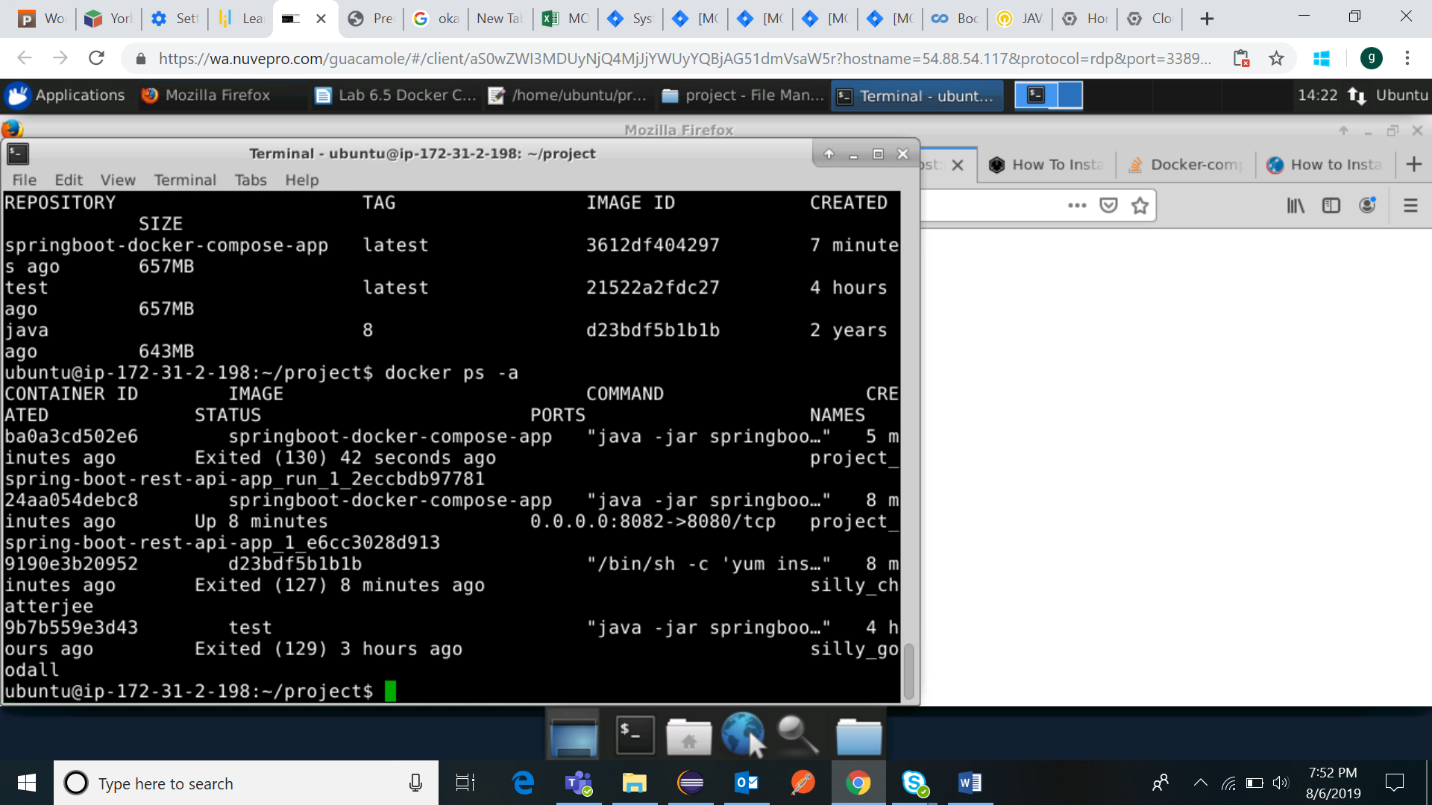


**Docker Images:** For checking the docker images, run the below command

**docker images**



docker ps –a



**Pushing to Docker Hub:**

Log in at DockerHub.

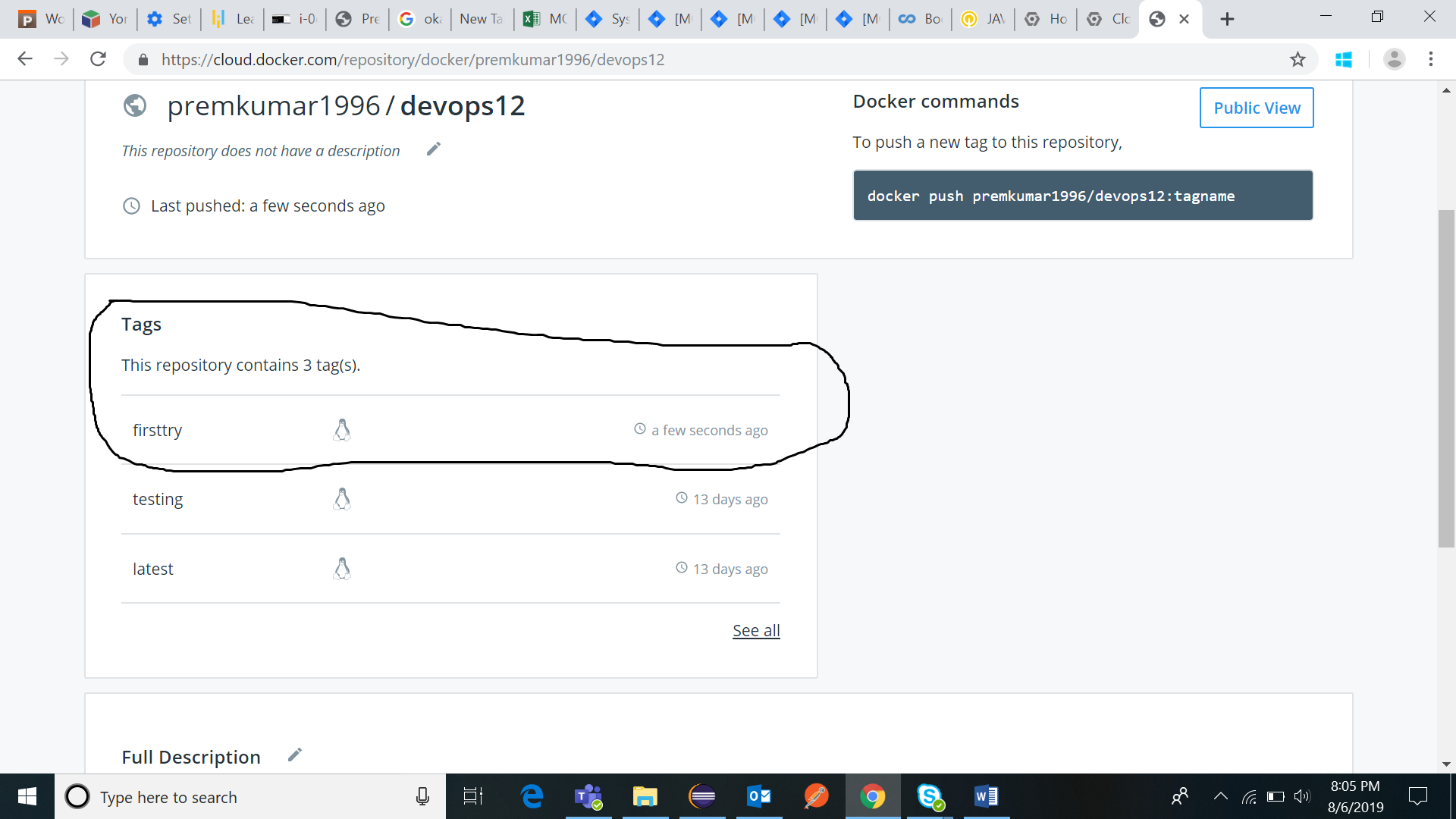
**docker login –username=<Your\_username> --password=<Your\_password>**

Upon successful login, tag the image with command

**docker tag <Image Id> <usename>/<RepositoryName>:tagName**

Now, Push the image into docker hub with the command and refresh the dockerHub page a new image will be added in it.

**docker push <Repository/Image>**

****

**Docker Pull:**

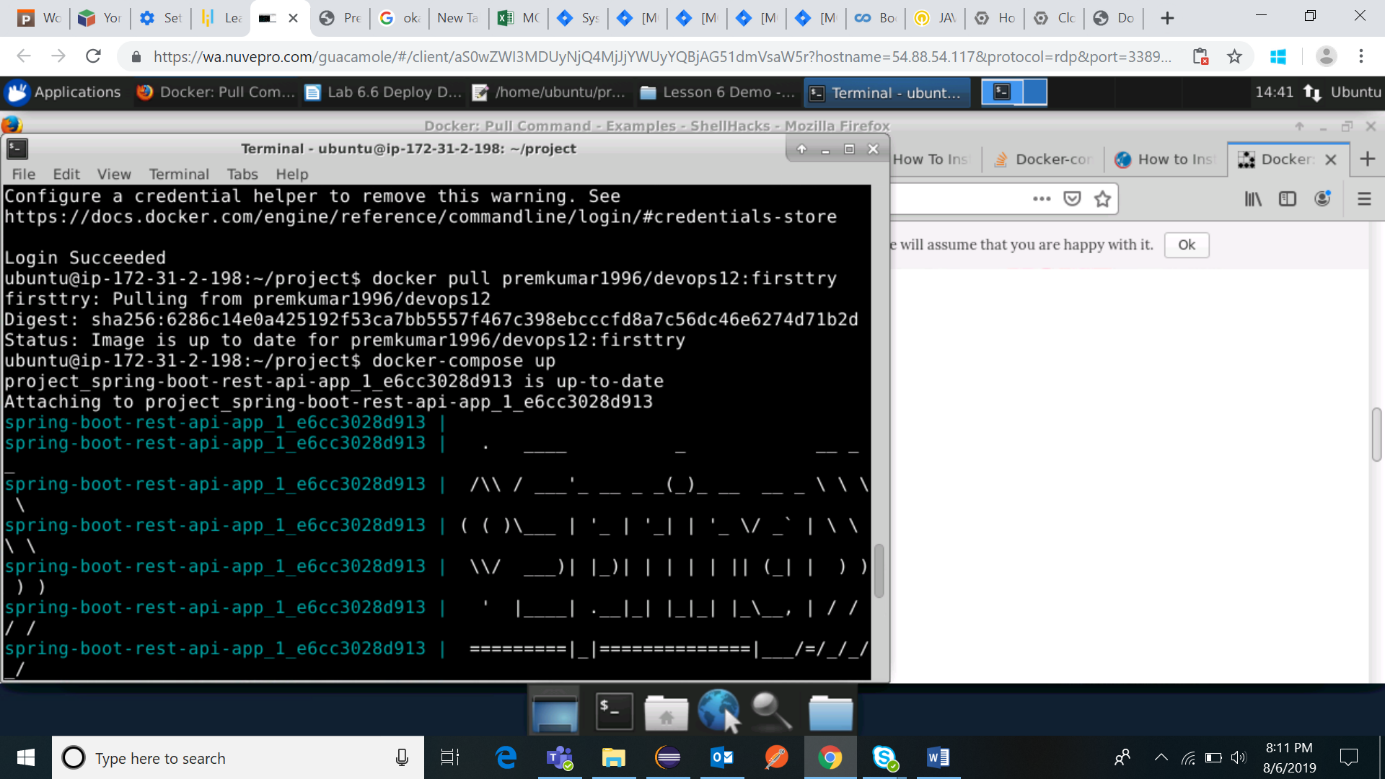
Now pull the docker image from docker hub with the command

**docker pull <username>/<repository name>: tagName**

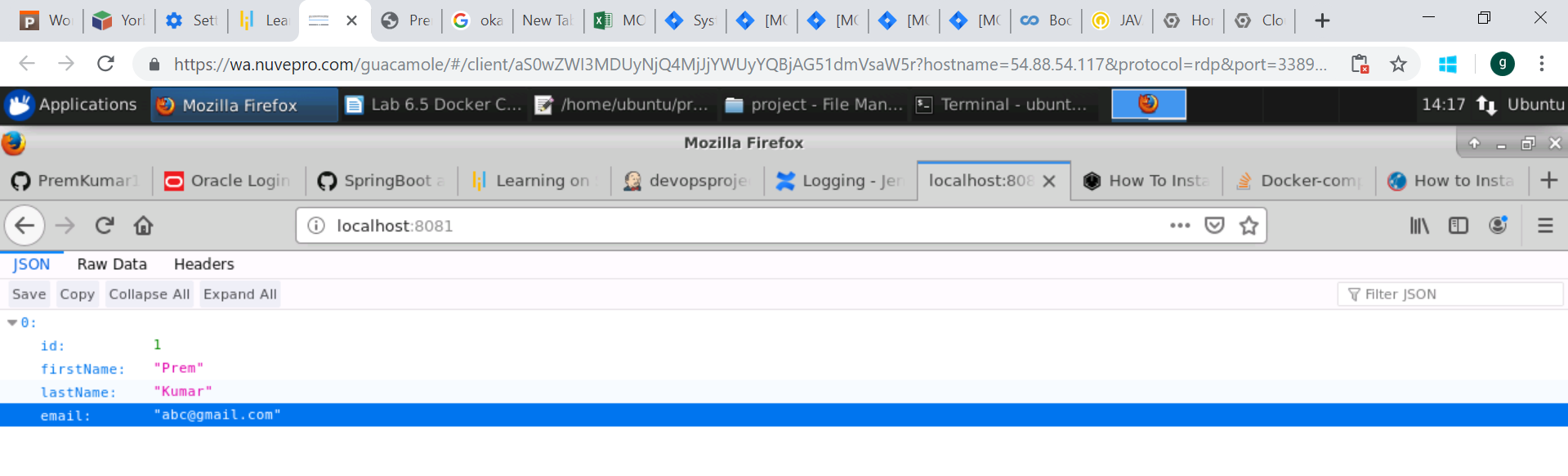
****

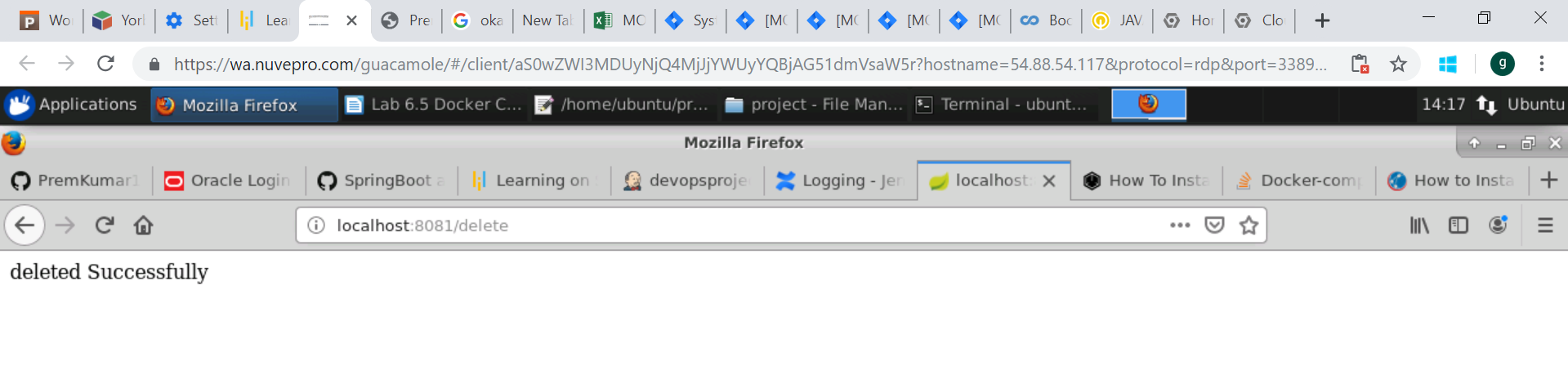
After taking pull from docker hub run the app by typing the command

**docker-compose up**

****

Go to web browser and open localhost:8081 where our application is running.

****

****

For stoping the running application execute the command with

**docker-compose down**