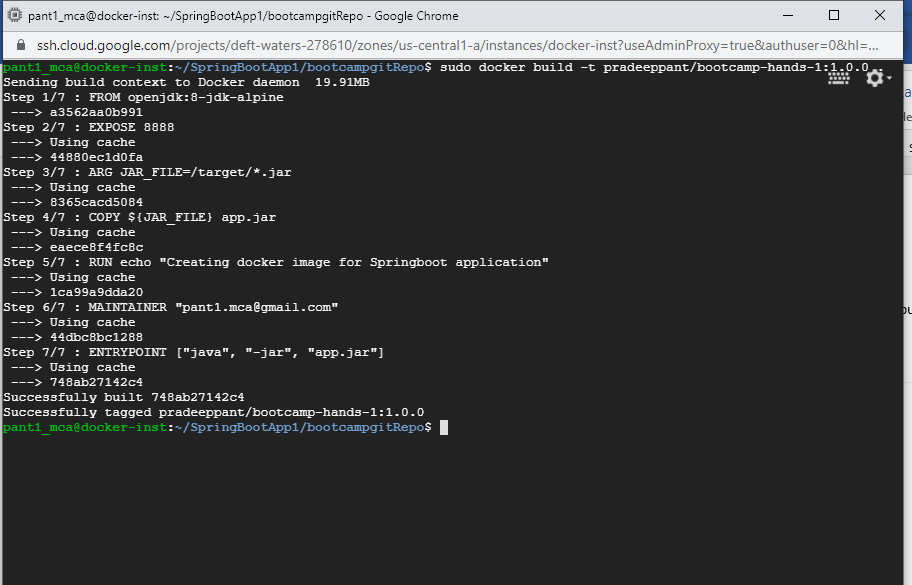
Day3

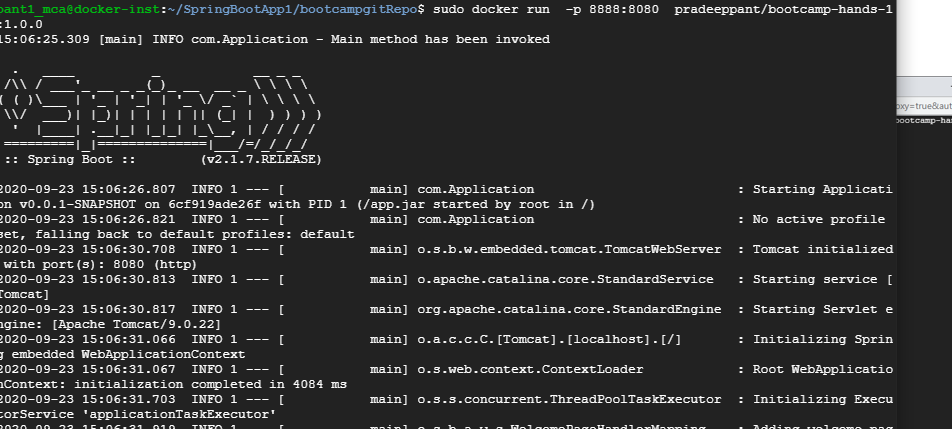
1. Write a Dockerfile to create a docker image for your Spring boot Application and push that image to docker hub, try running it different environment by pulling it.
2. Create a docker file in existing springboot application



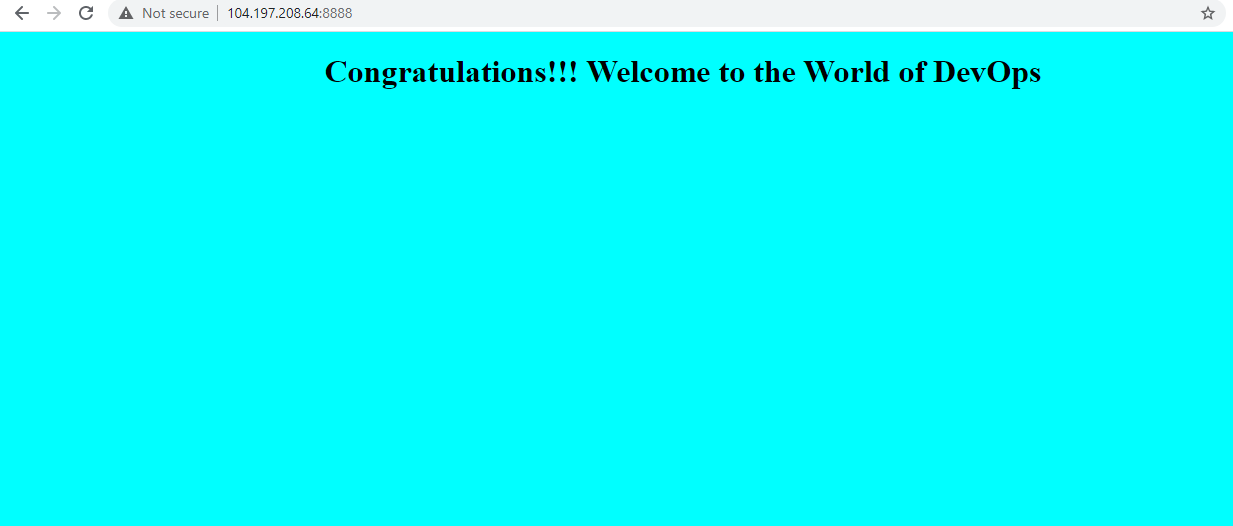
1. Build the docker image.



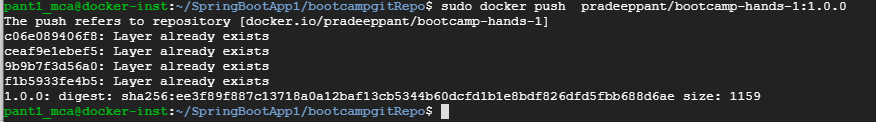
1. Run the container.

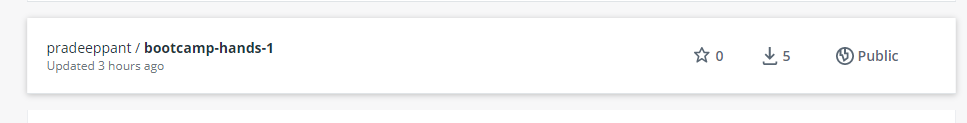


1. Check on the GCP server .

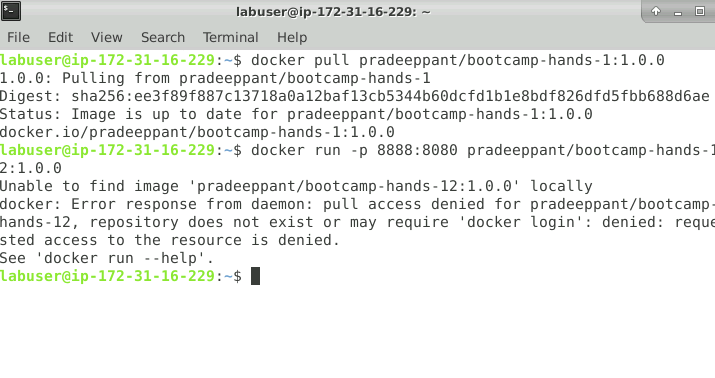


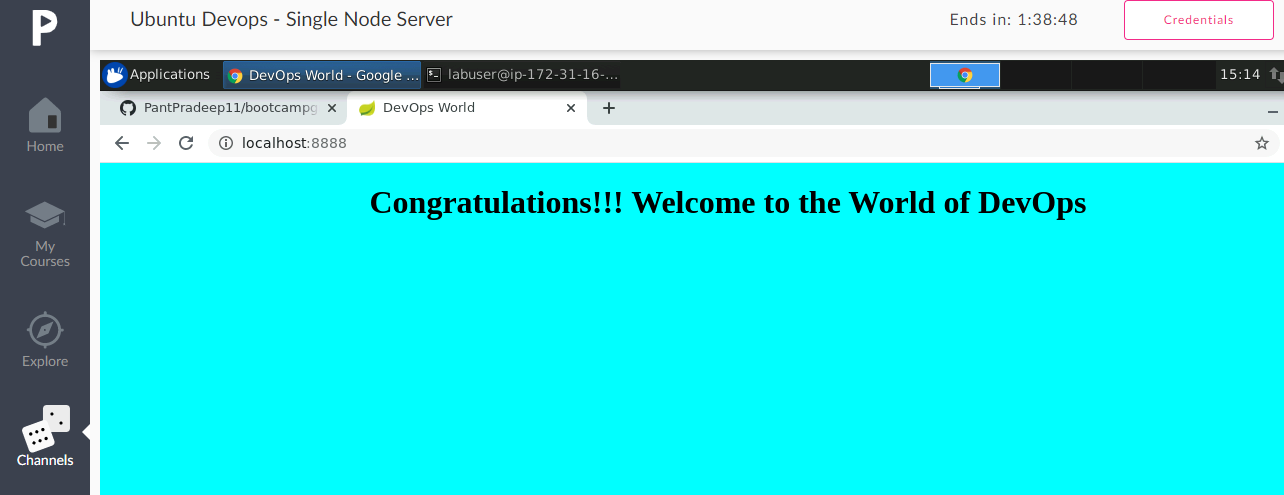
1. Push this to the docker hub.



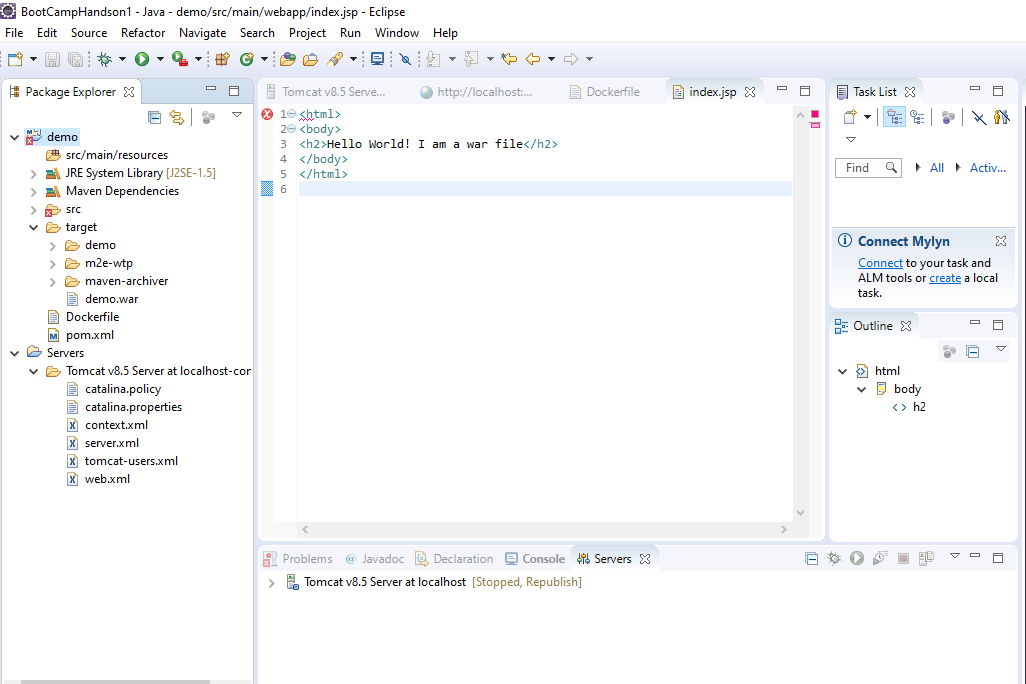


1. Pull and run the app on another machine

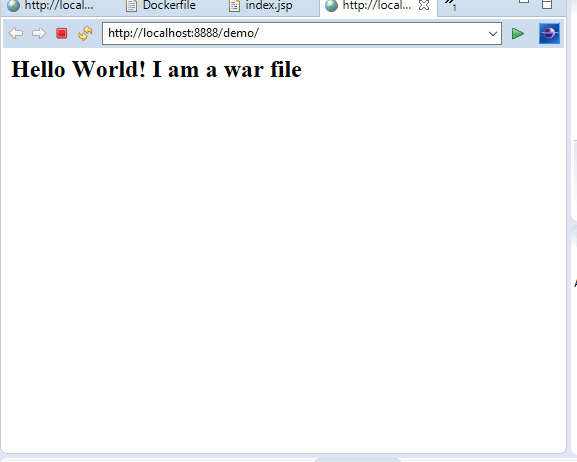




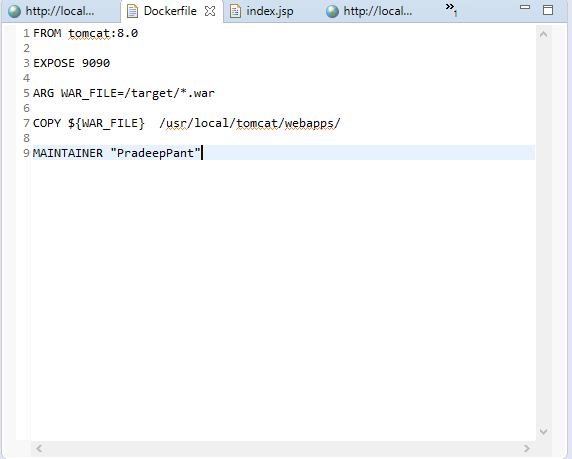
1. Write a Dockerfile to create docker image for a Dynamic Java Project(war file) and push that image to docker hub. Try running it in different environment by pulling it.
2. Create a war project using maven add configure tomcat server on it.



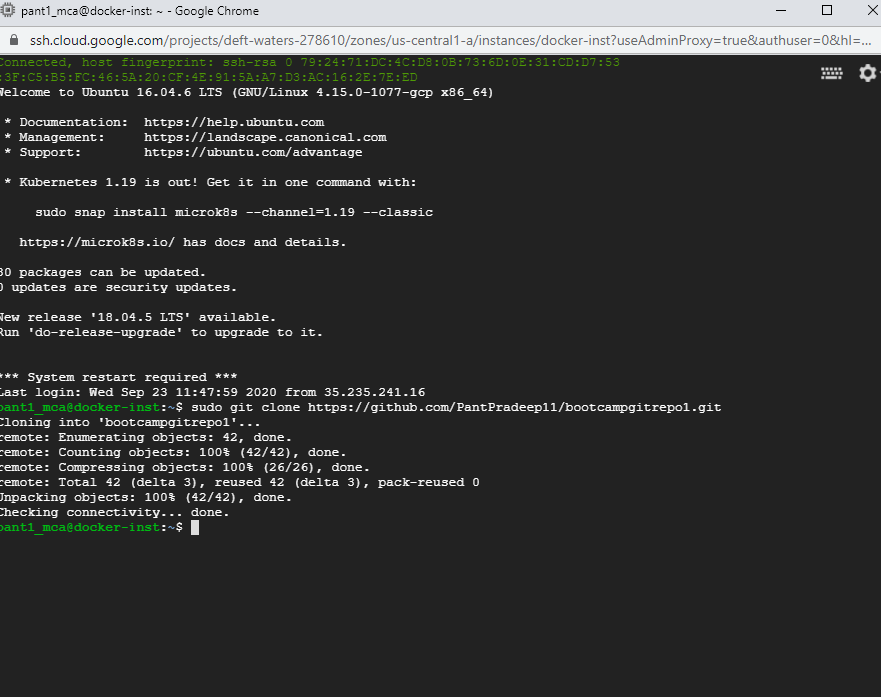
1. Run project using the eclipse



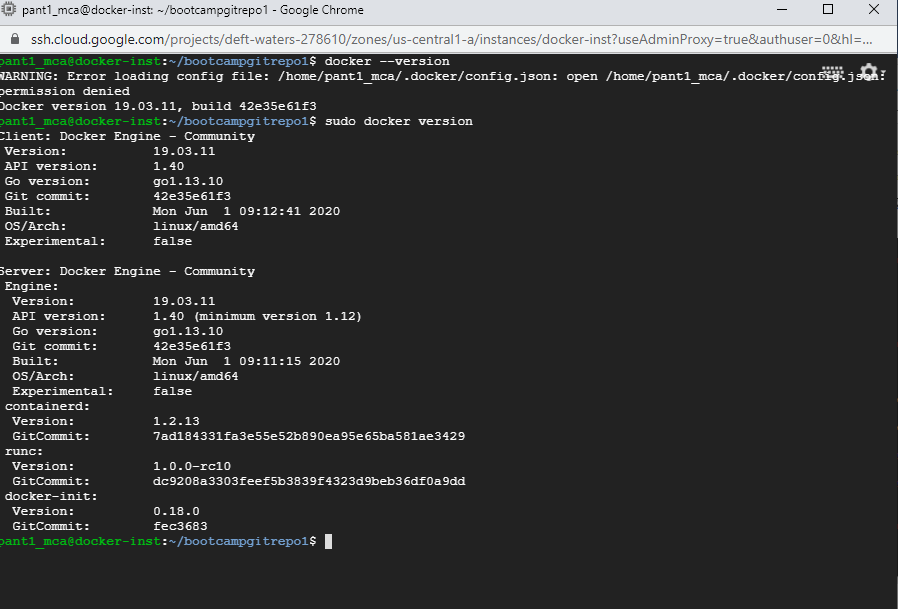
1. Create a docker file to make a container and deploy the war file on tomcat server.



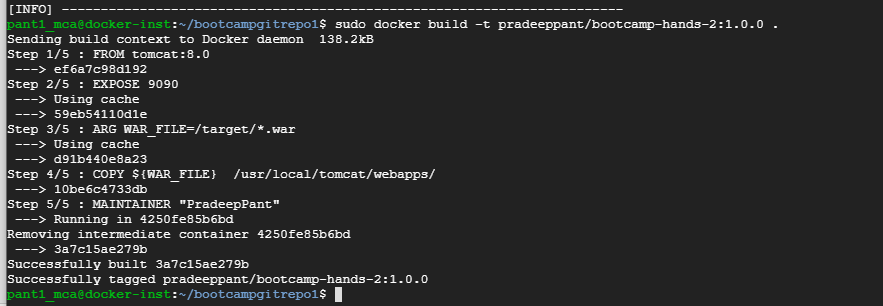
1. Push the solution to git and clone that repo in GCP machine.



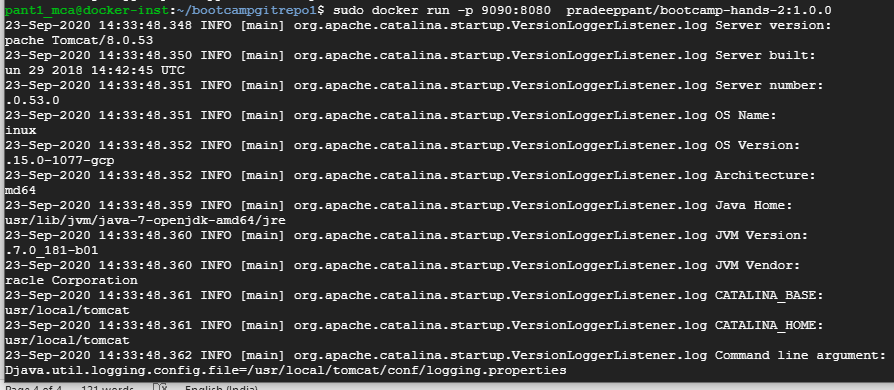
1. Check docker version



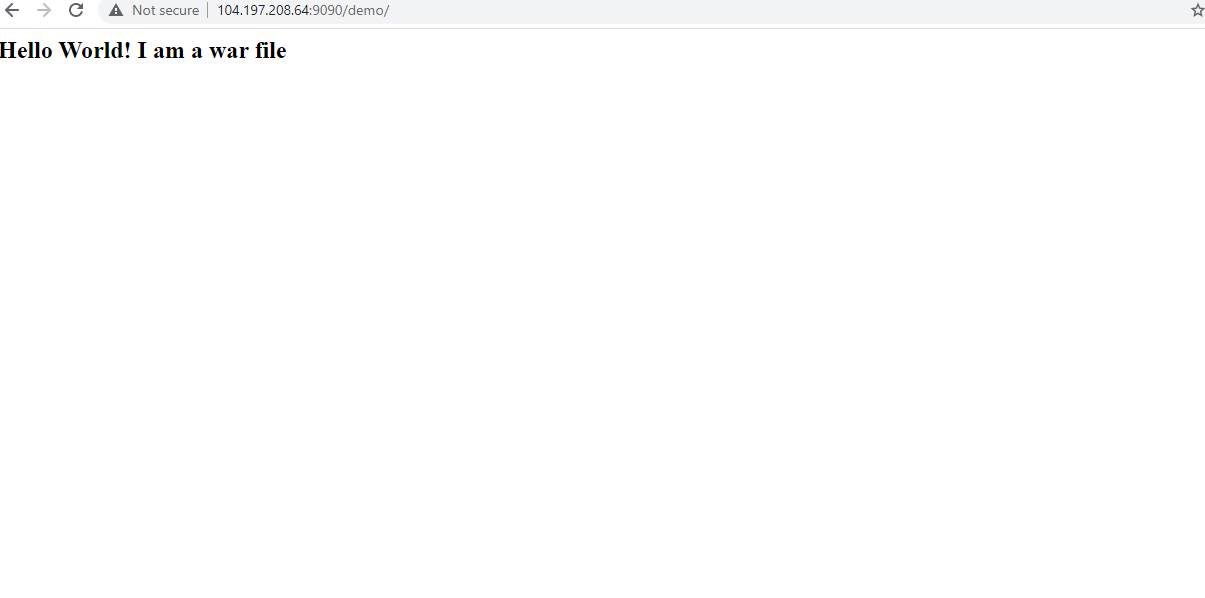
1. Create the docker image with tag name



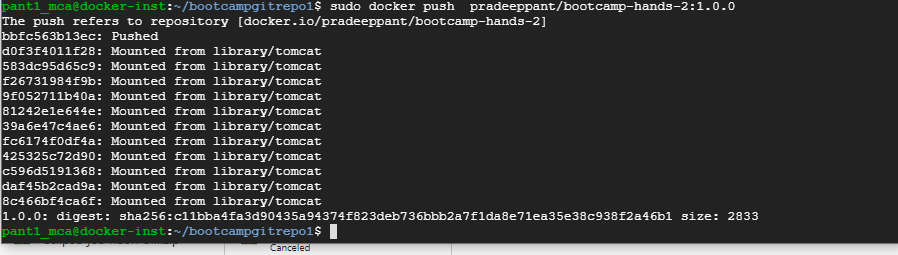
e. run the docker image

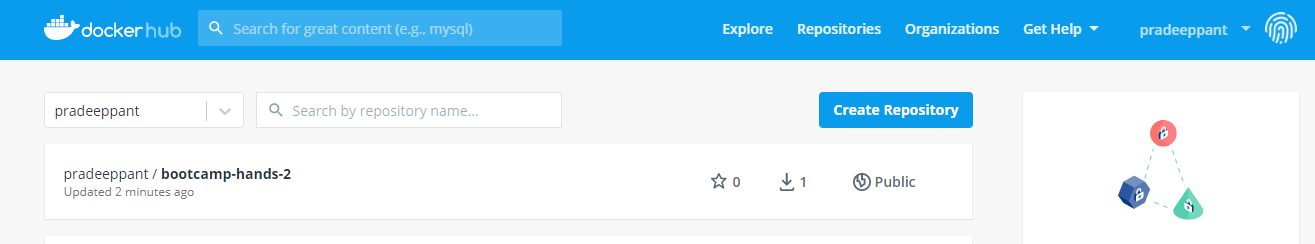


1. Run the application on browser

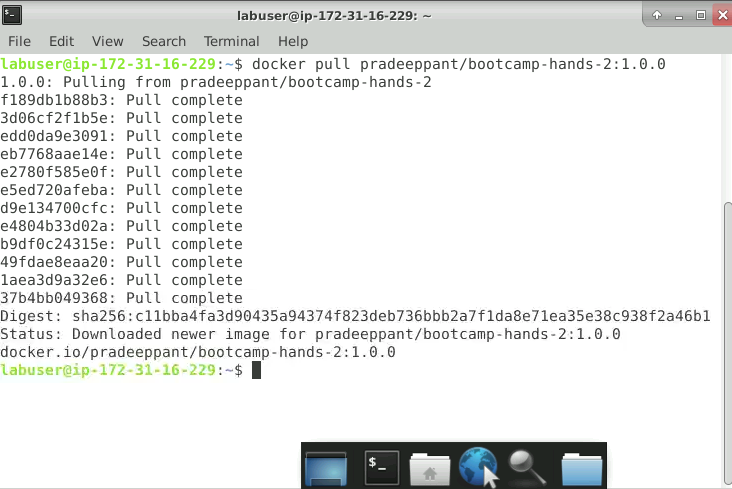


1. Push the image into docker hub.

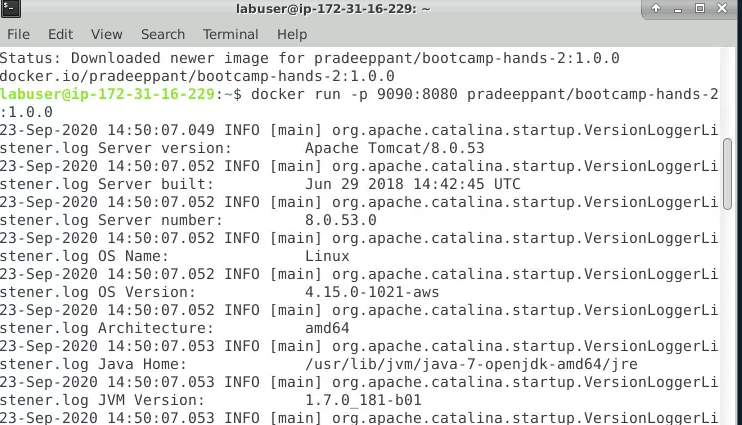


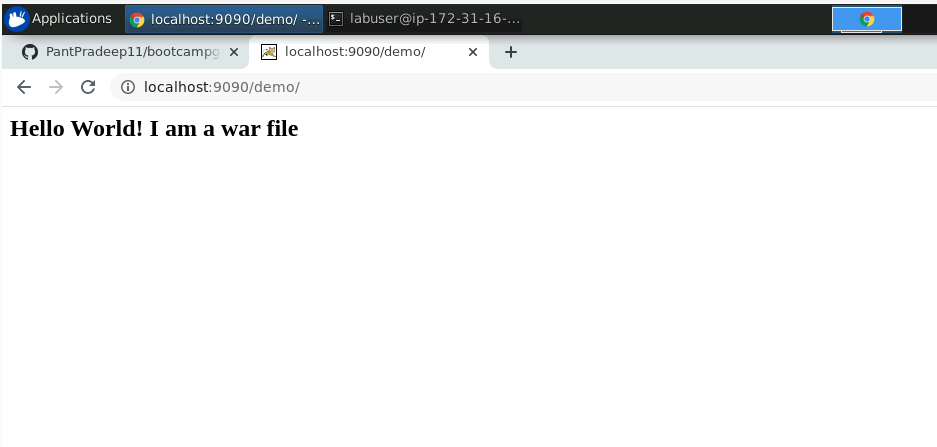


1. Pull the docker image on another machine



1. Run the application another app





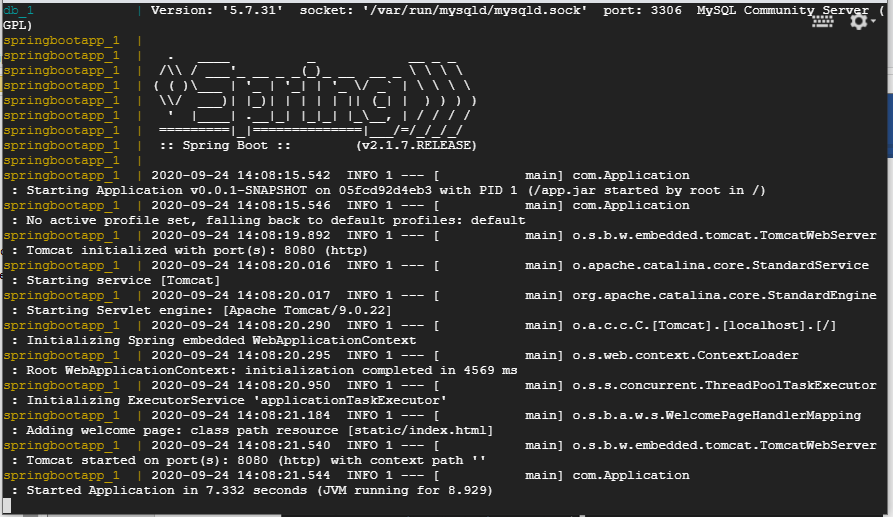
1. Write a DockerCompose file to run your SpringBoot Application along with mysql db container and establish the connectivity between application and db.
2. Add a docker compose file in root folder

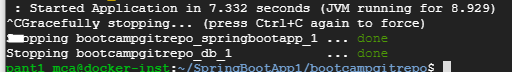


1. Install the mysql on the vm.
2. Pull the code in vm and insatll the docker compose.
3. Test the version to check the installation.



1. Run the docker compose file both application are running on the same network





1. Both application are running on same network

