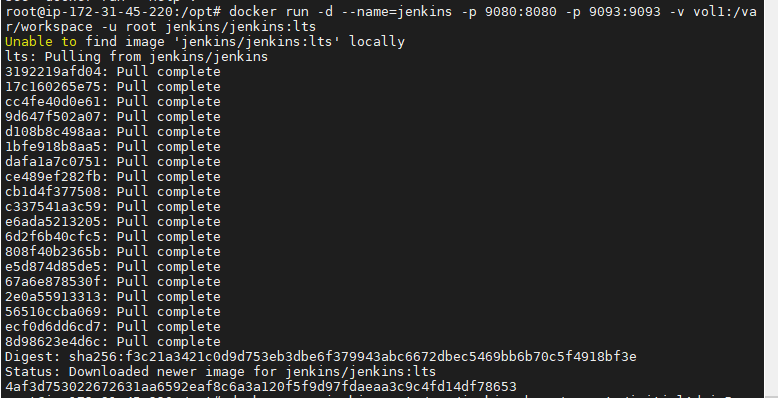
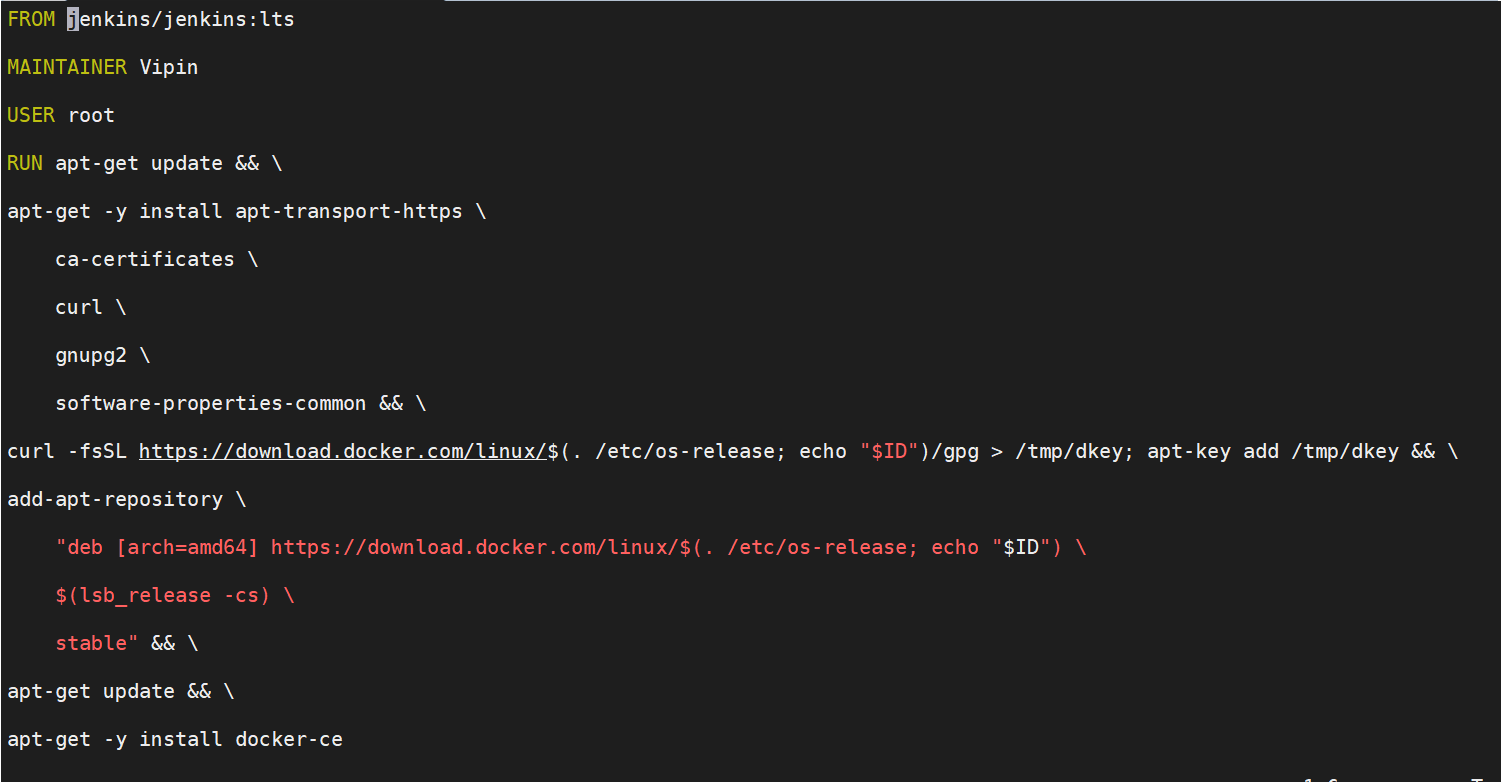
Use-Case3 : DOCKERIZED JENKINS

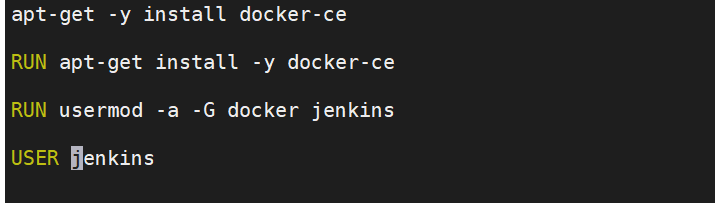
Create a Jenkins container after installing docker on Ubuntu.

**docker run -d --name=jenkins -p 9080:8080 -p 9093:9093 -v vol1:/var/jenkins\_home/workspace -u root jenkins/jenkins:lts**



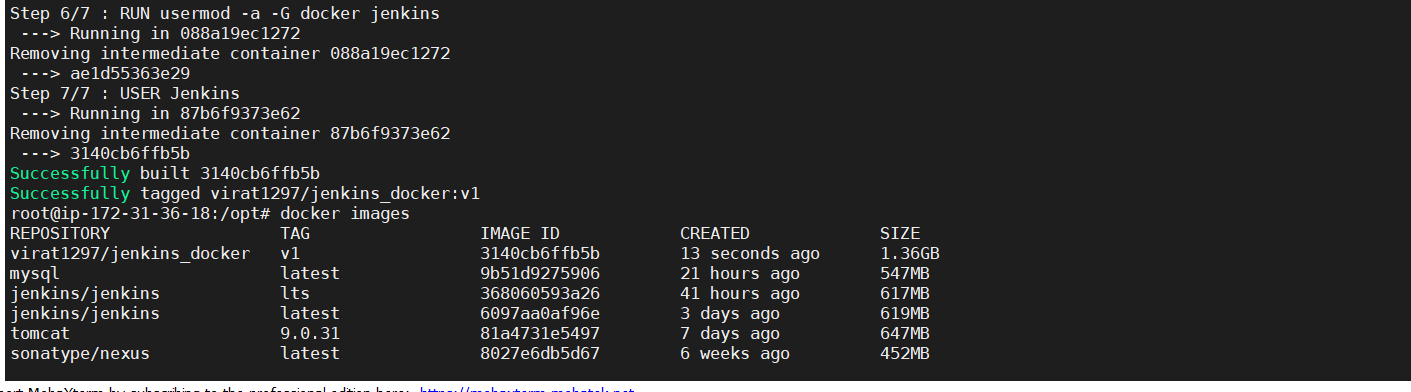
Now create a docker file which has base image of your Jenkins container:lts.





Now build the image using this docker file.



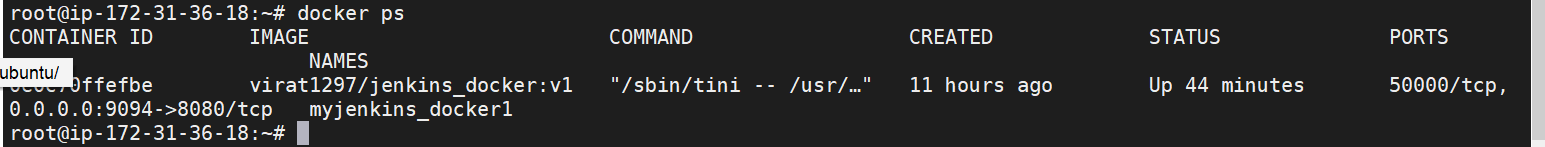


After successful build push the image to dockerhub.



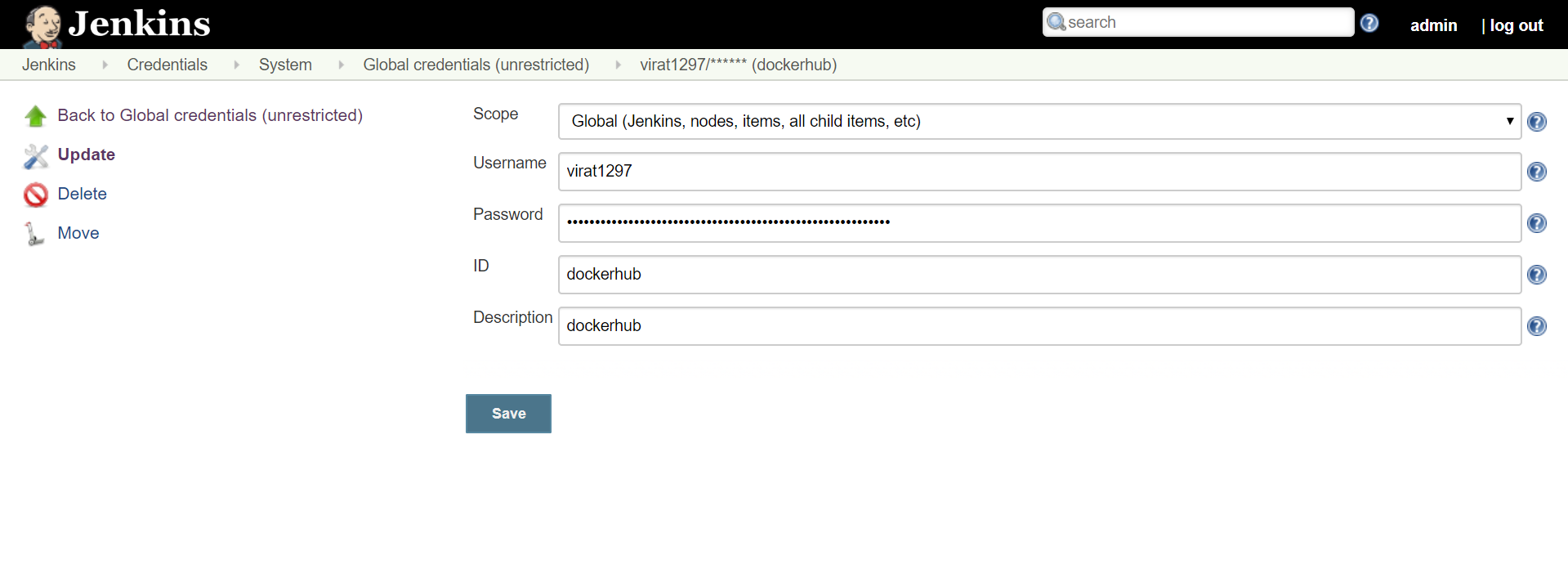
Now create a container using the same image.

**docker run -d --name=myjenkins-docker1 -p 9094:8080 -v /var/run/docker.sock:/var/run/docker.sock -u root virat1297/Jenkins\_docker:v1**

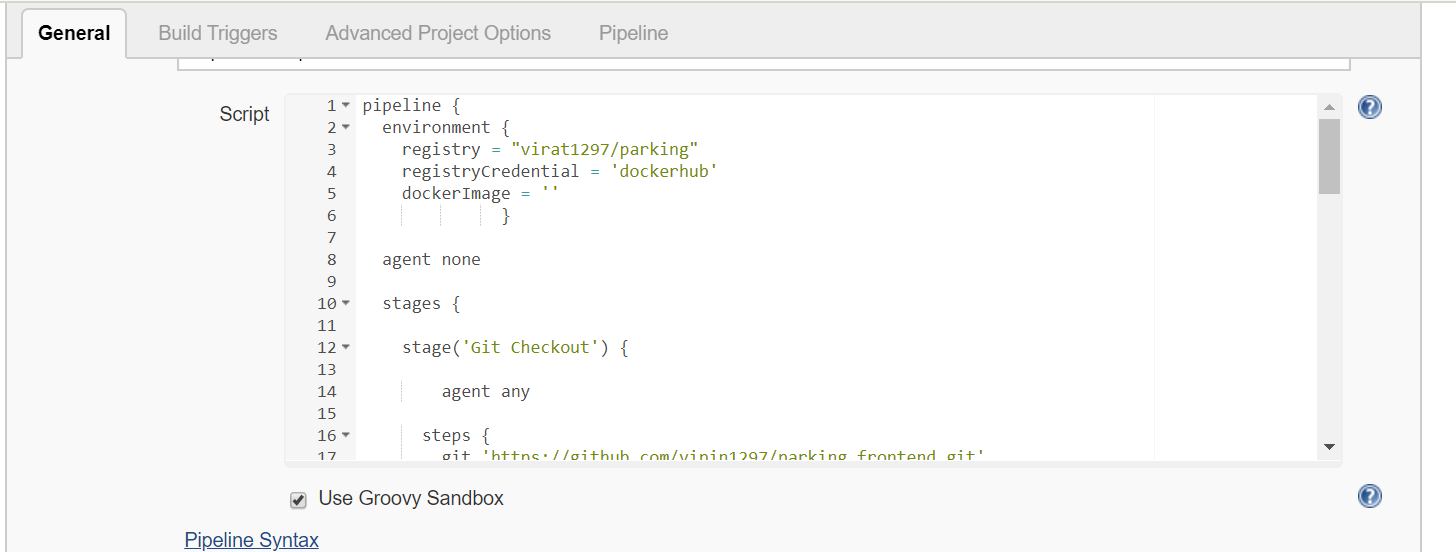
****

Now setup Jenkins on this myjenkins\_docker container.

Before creating a job add credentials on Jenkins of your dockerhub account.

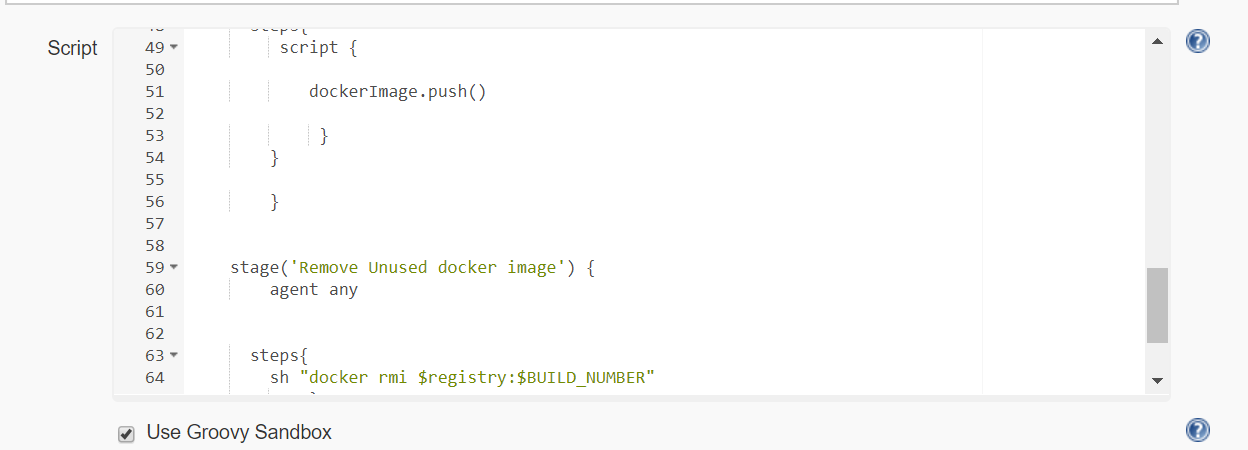


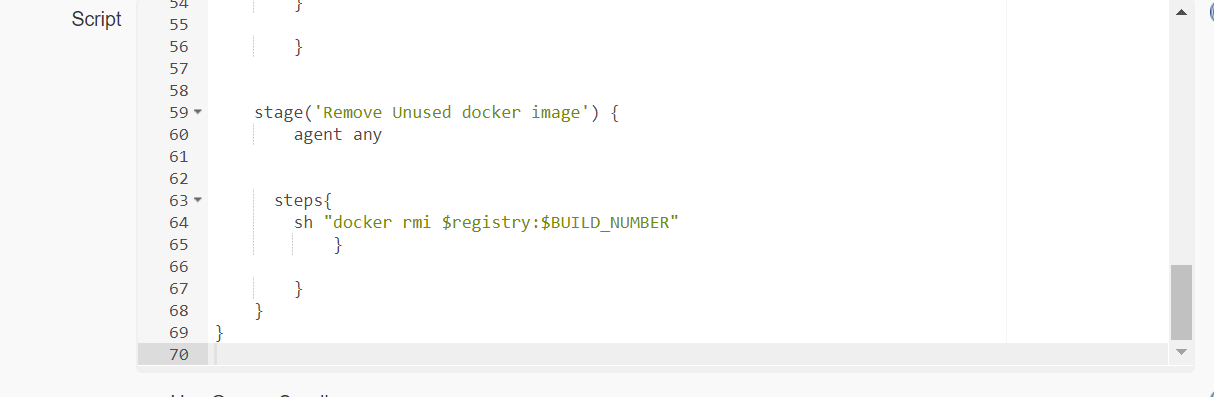
Now build a job using pipeline script.











Note- In the pipeline script you need to provide agents for npm otherwise it will not recognize npm command.

 agent {

     docker { image 'node:12-alpine'}

              }

You need to also use

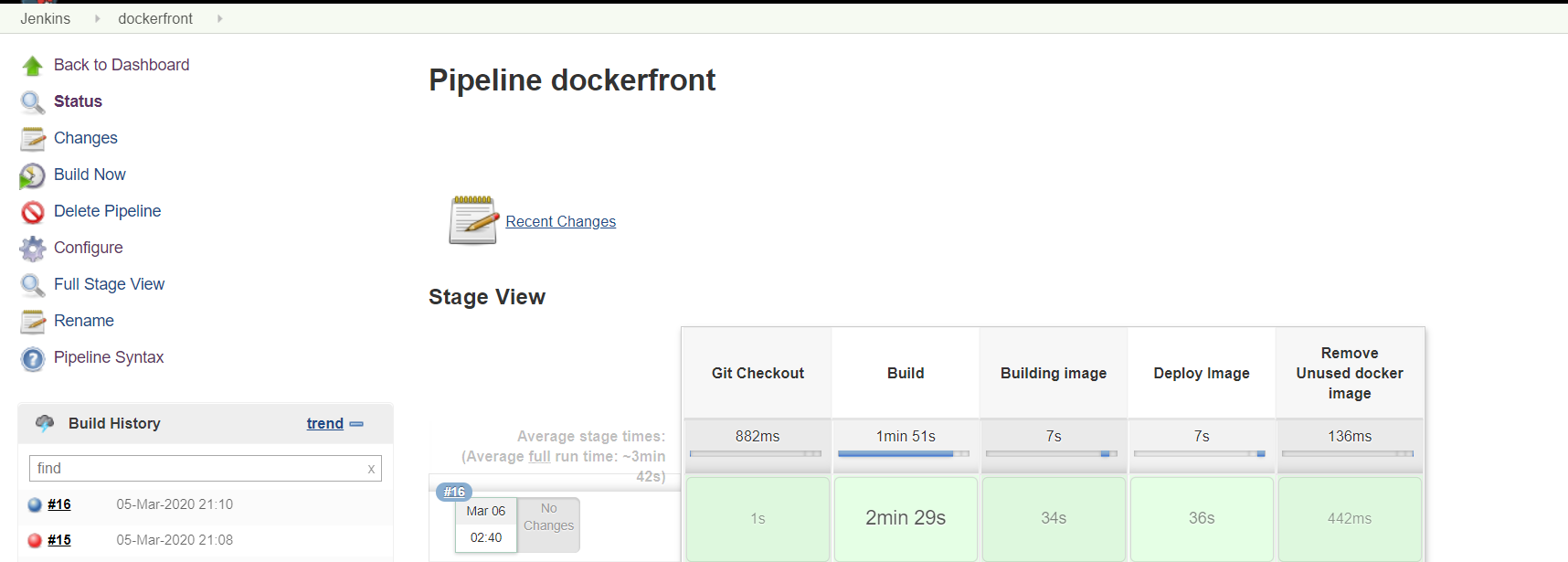
agent {

dockerfile true

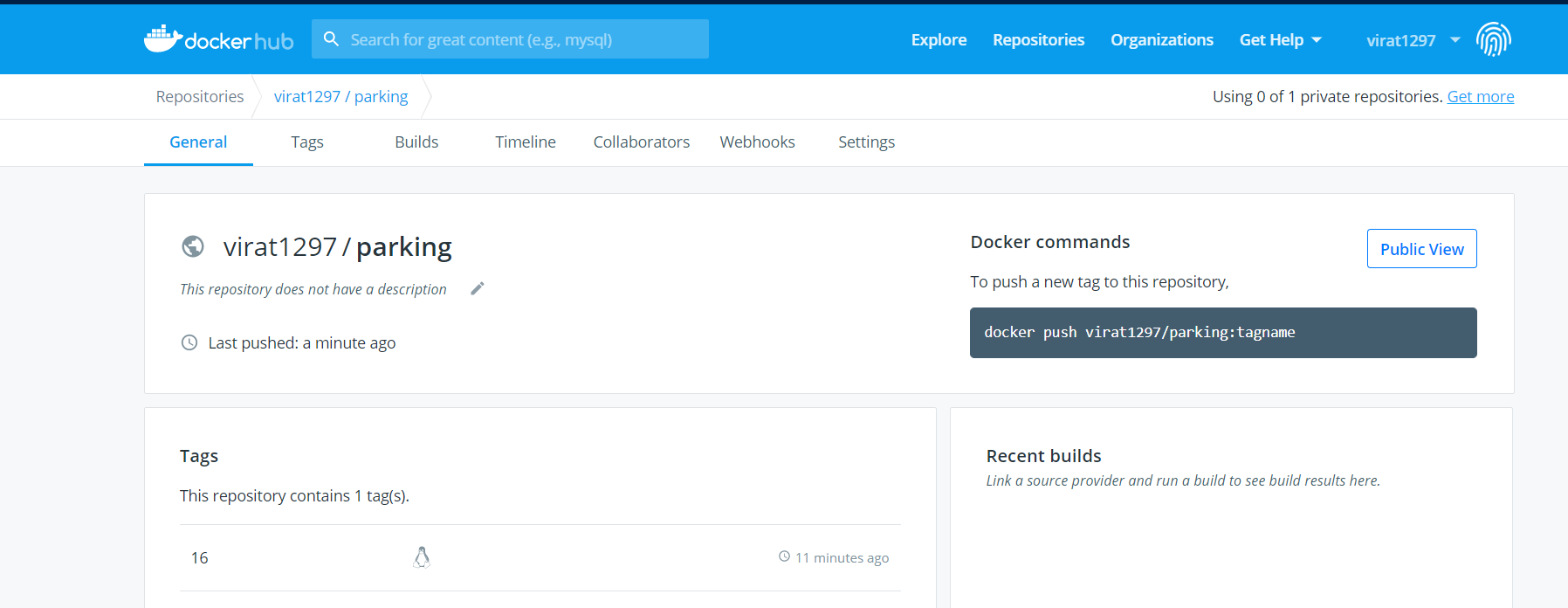
}

before building image stage otherwise it will not find dockerfile.

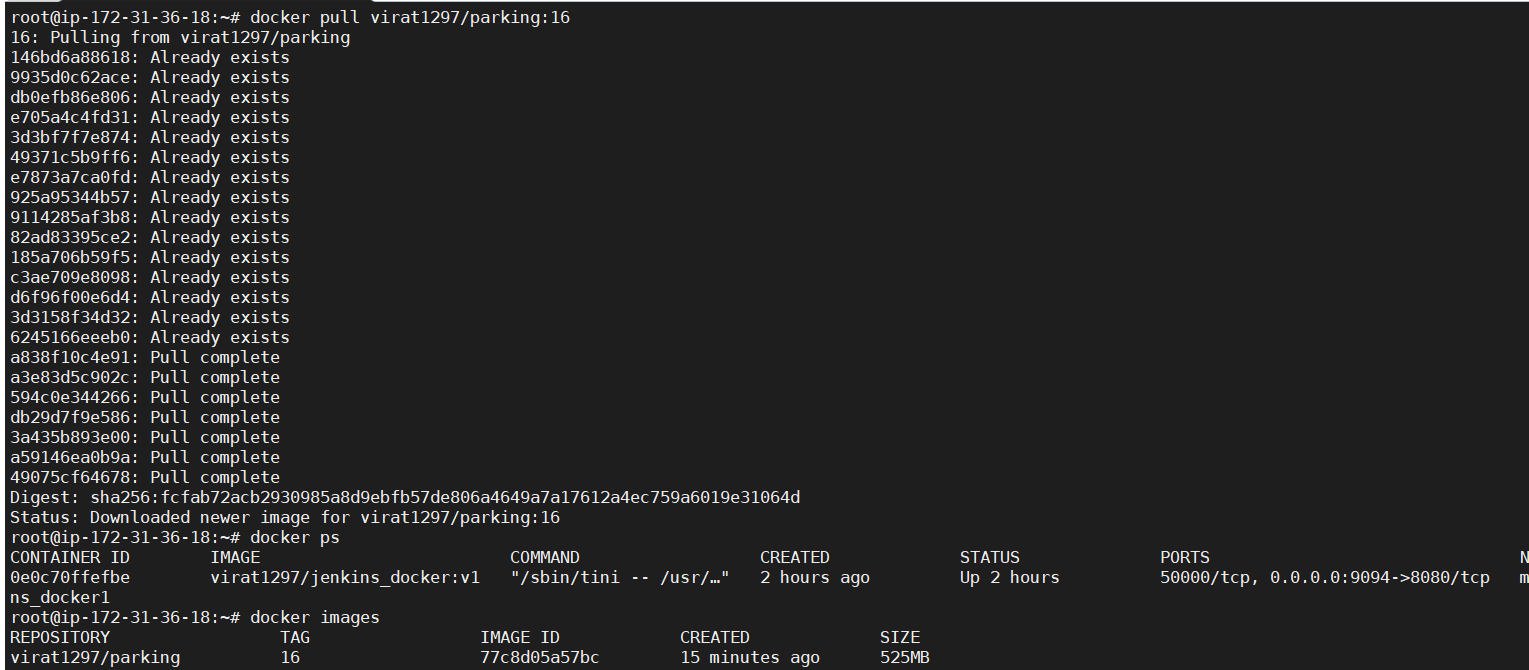
Now Build the job.



Now check the image on your dockerhub.

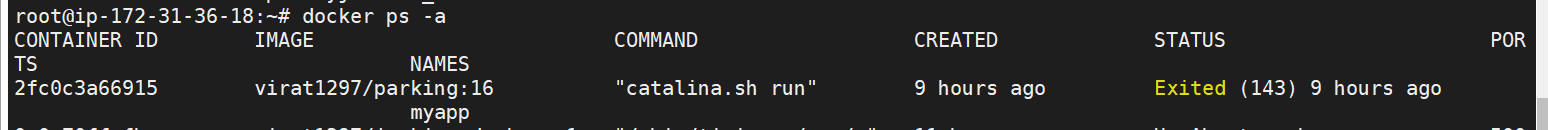


Now pull your dockerhub image.

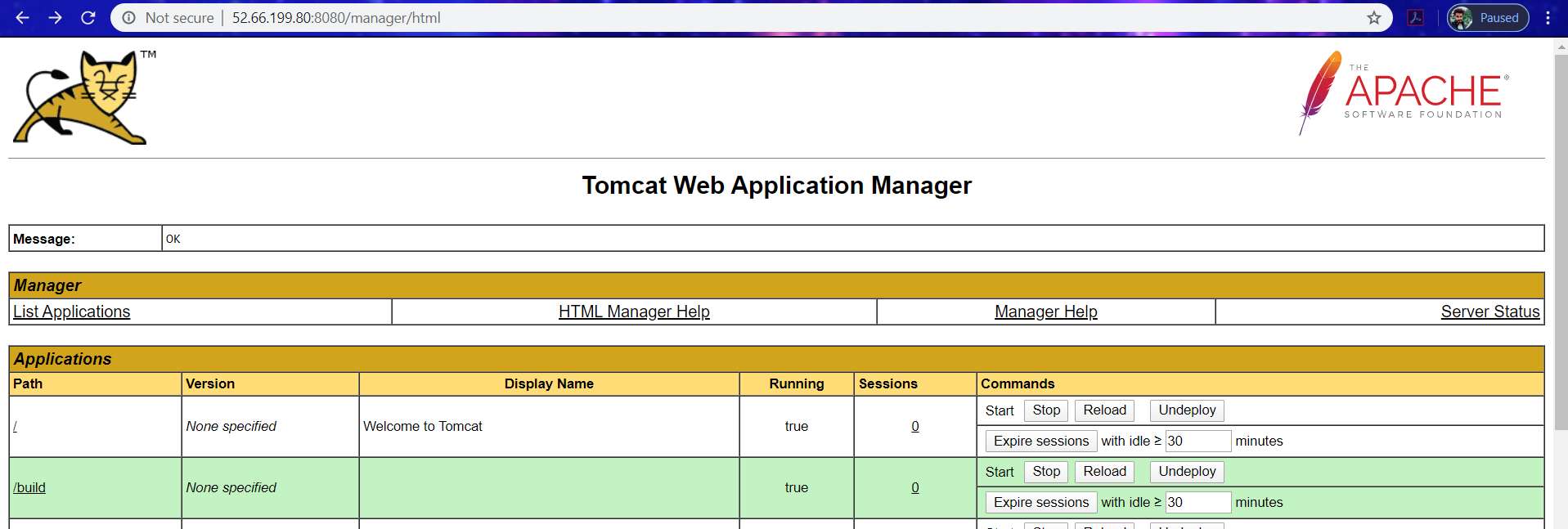


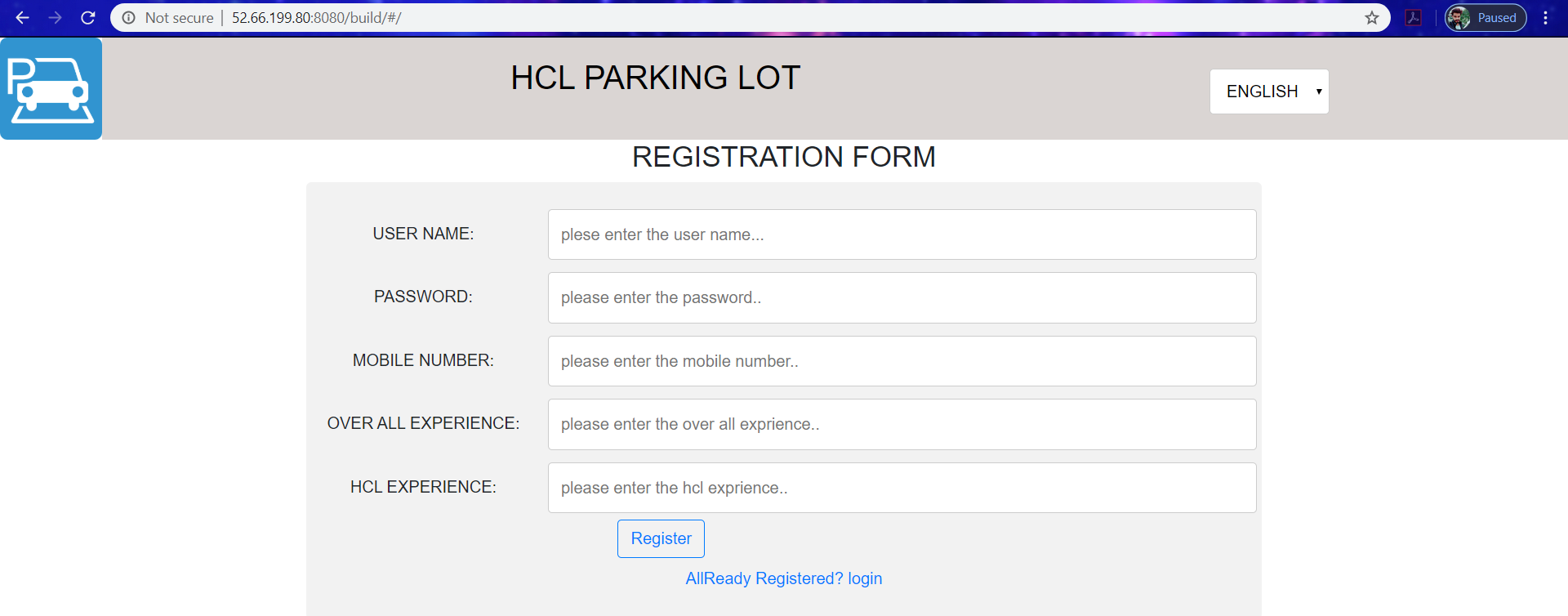
Create a container from this image.

**docker run –d –name=myapp –p 8080:8080 virat1297/parking:16**



Run this container see the output on tomcat as <yourIP>:8080.





Now you have successfully used docker in docker.