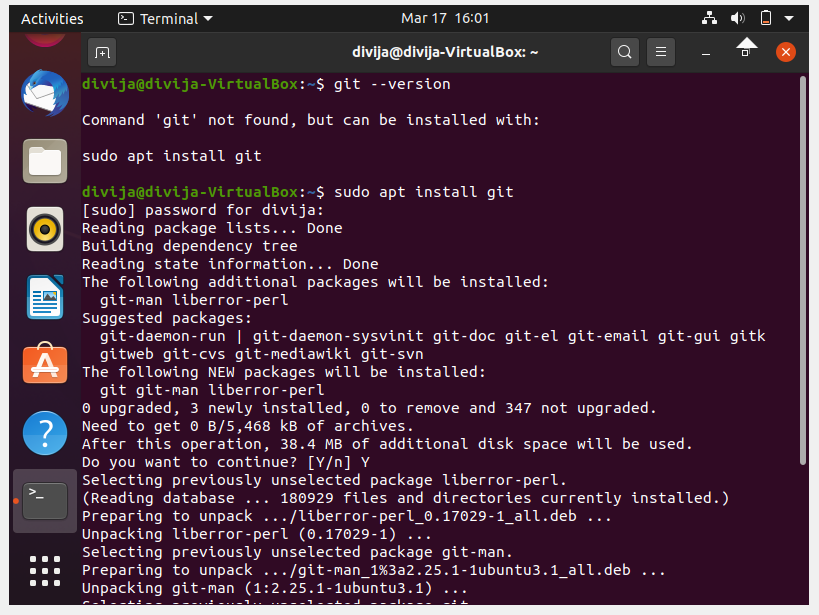
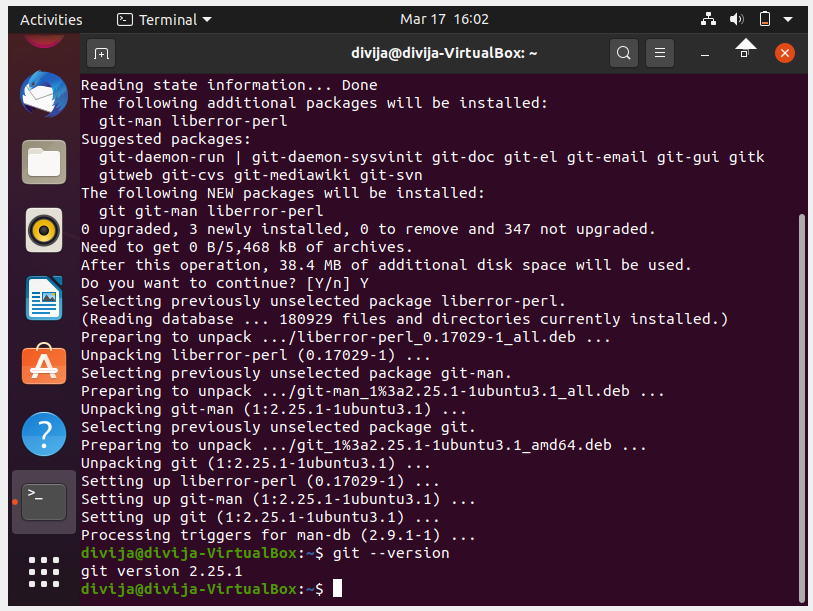
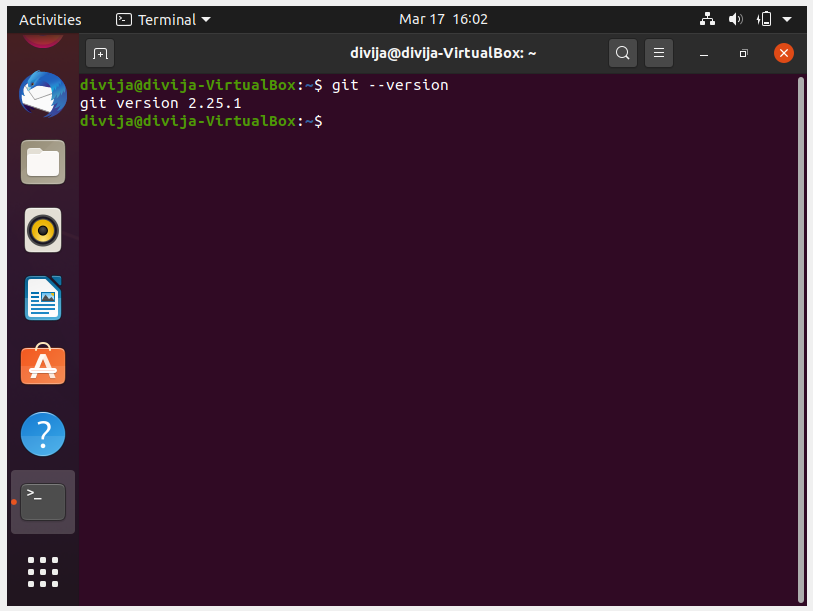
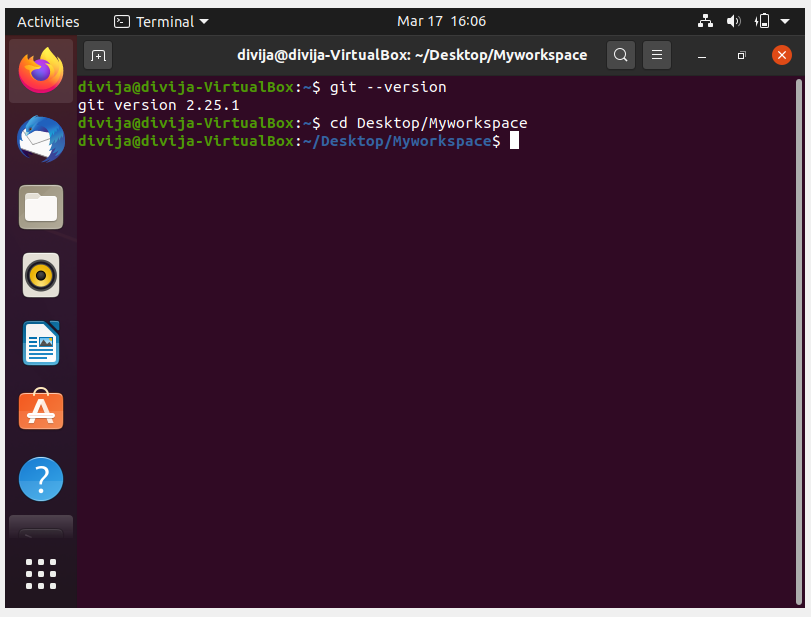
Installing git on Virtual Machine.

# sudo apt install git-all

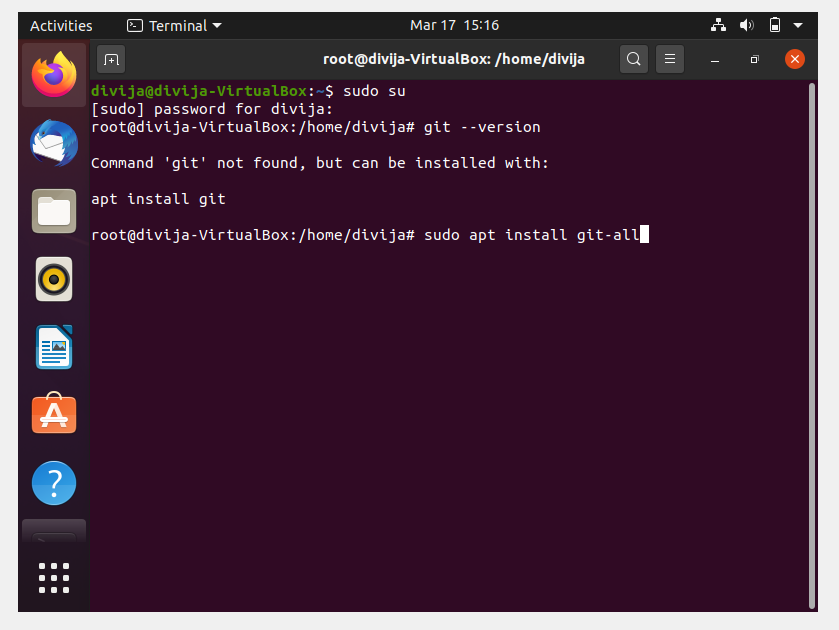




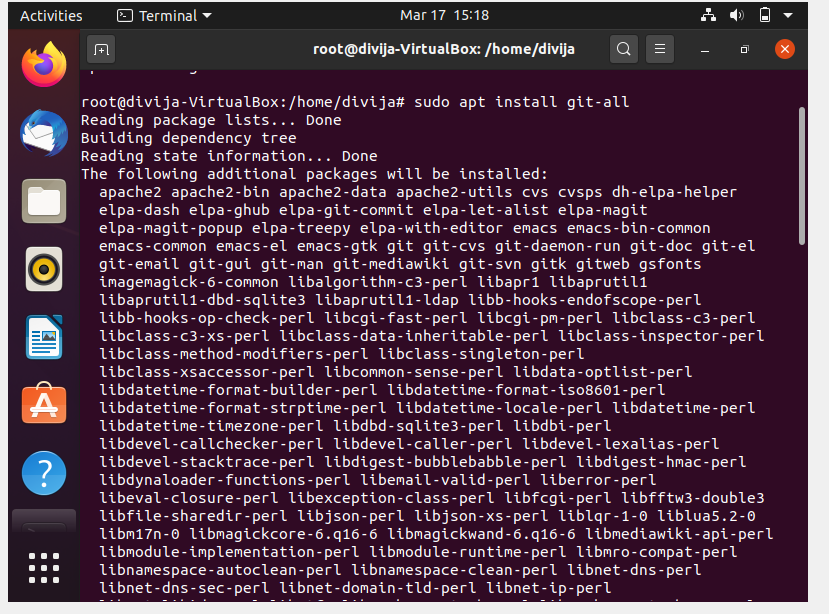


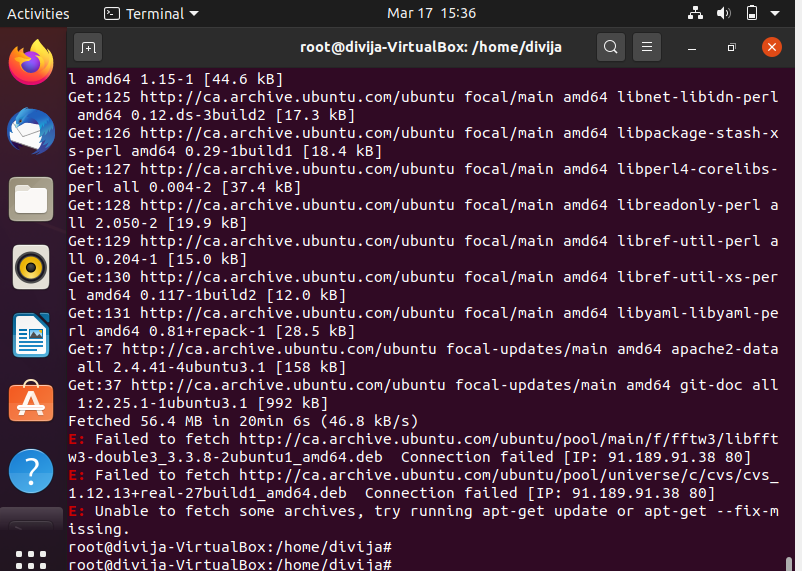


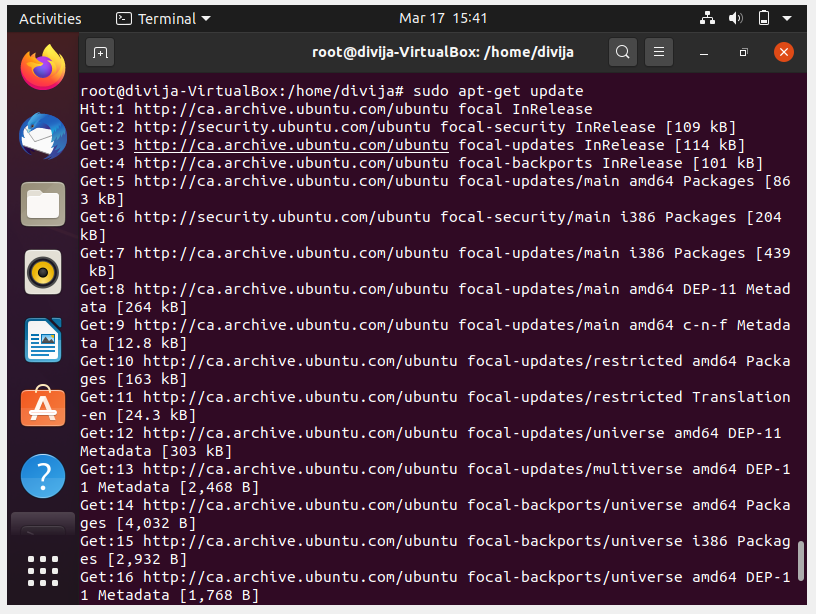
sudo apt install git-all

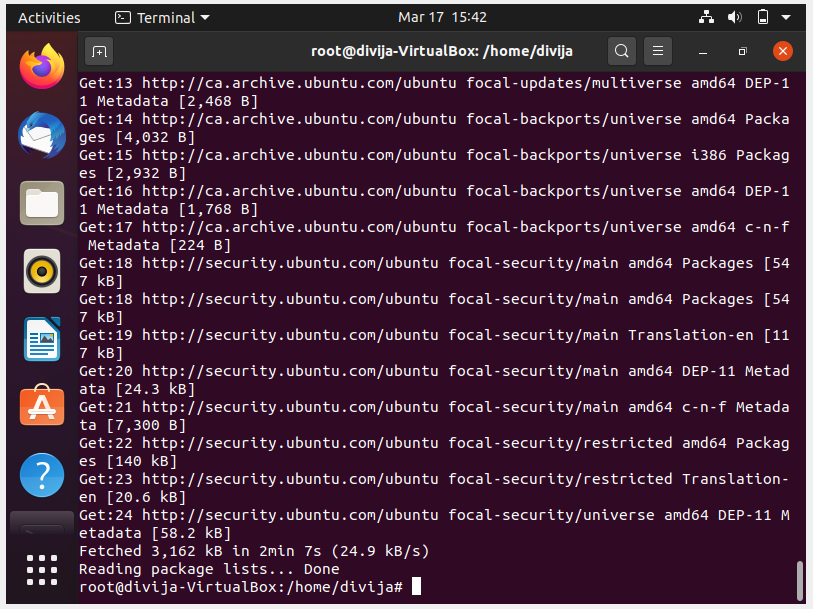


Installing git

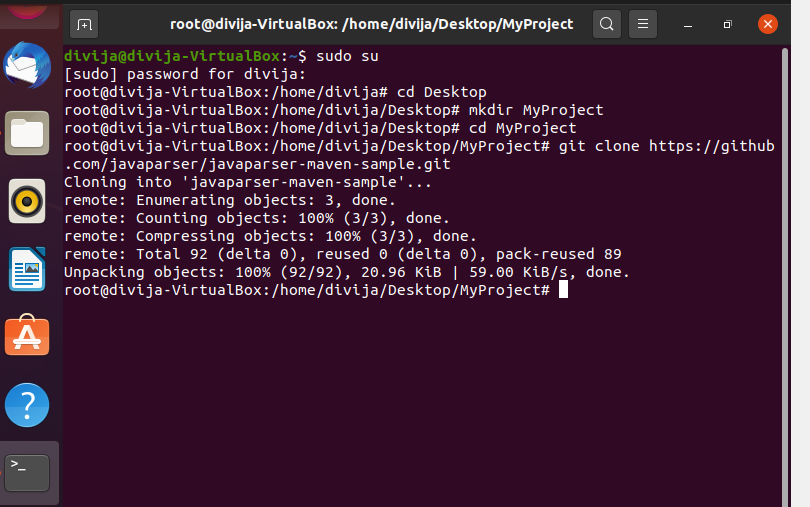


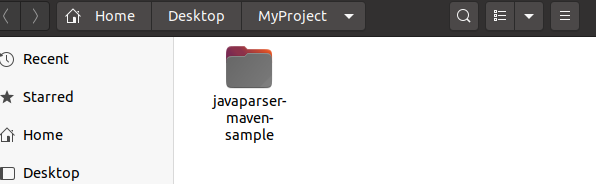






Created a Folder on Desktop “MyProject” - > Git Clone Sample Project from <https://github.com/javaparser/javaparser-maven-sample.git> to “/Desktop/MyProject/”

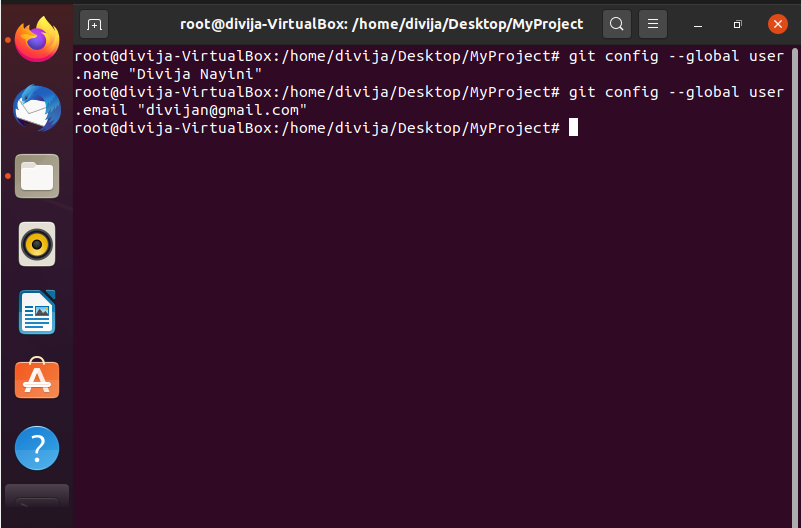


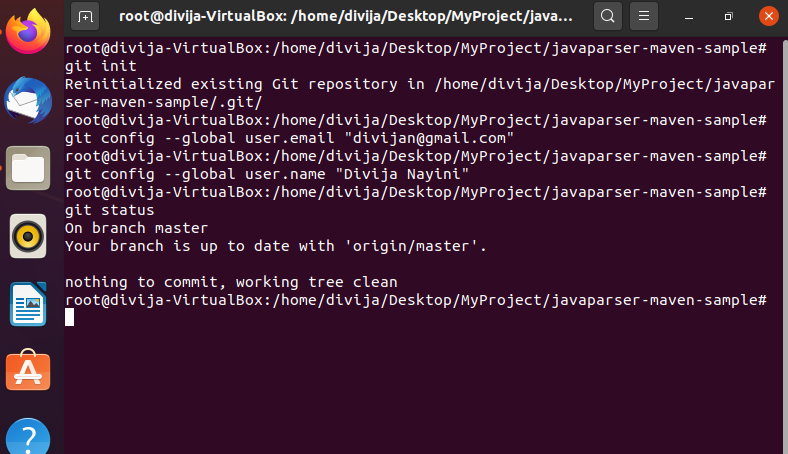


Now Git is installed the Github account is created.

Local Repo git commands:

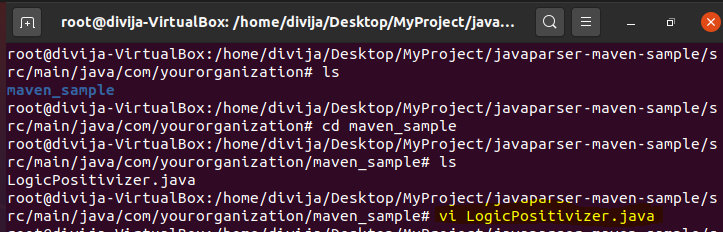
Git init & config accounts

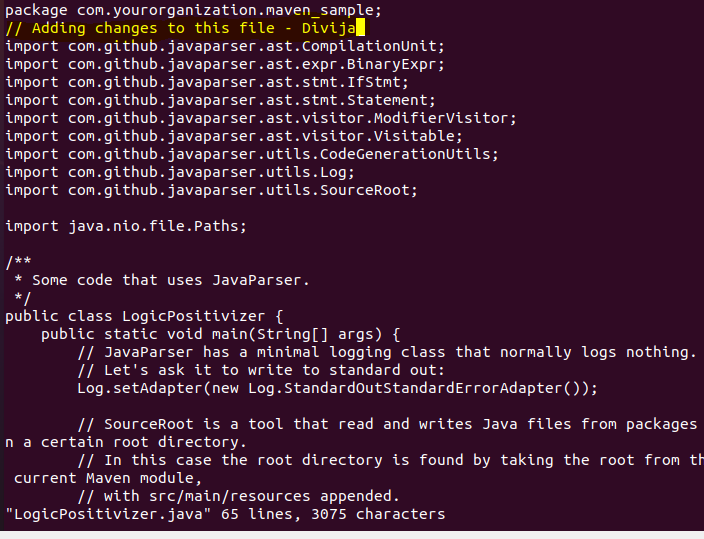


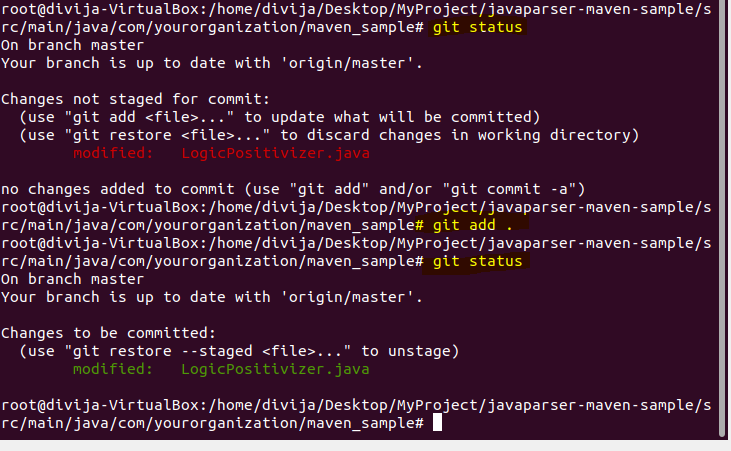


Vi

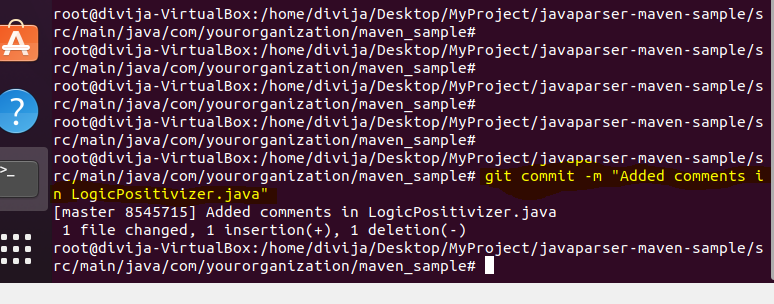
Vi the .java file





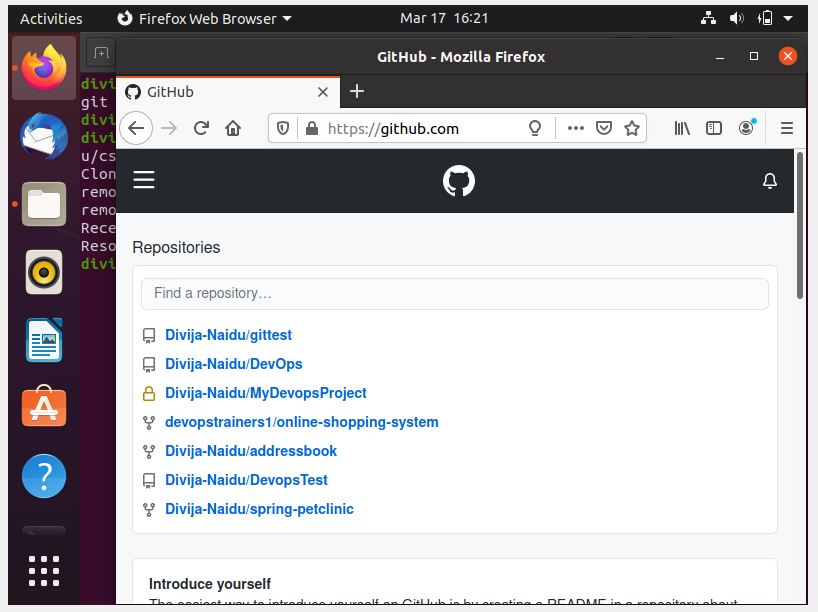


Commit the changes to local repo

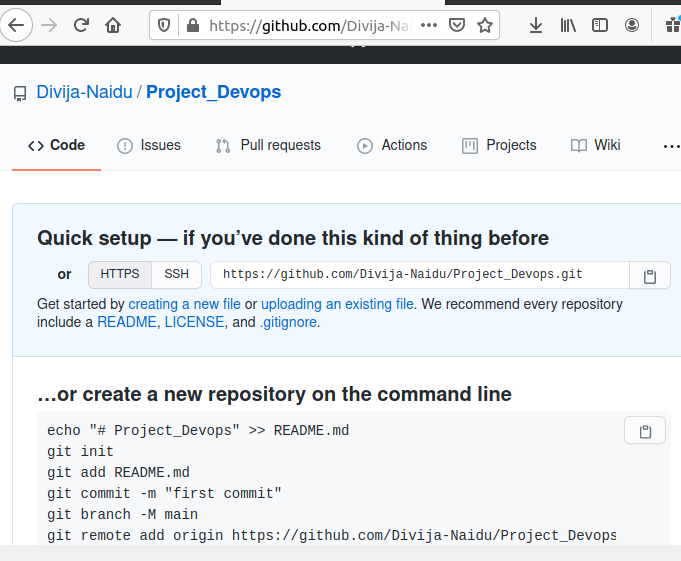


Now we are done with commit to local Repo.

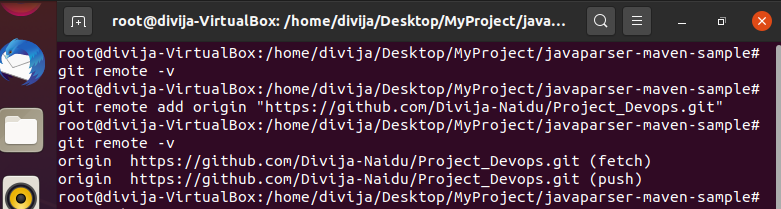
Sign-account Github - working with Remote repository.



Sign-up to github and Created a new pubic repository **“Project\_Devops”:**



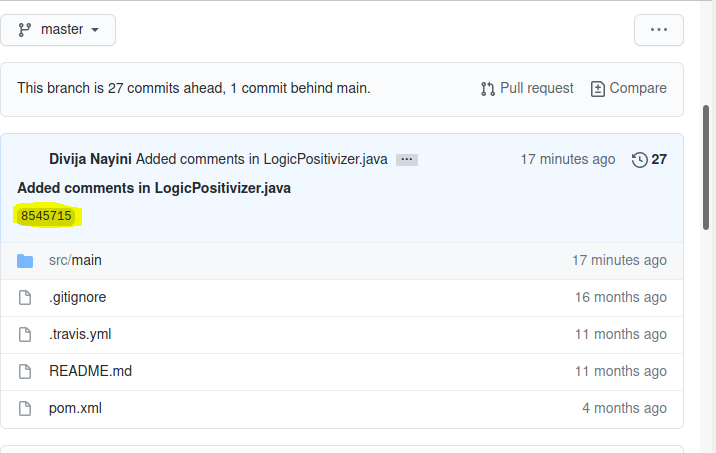
Connect to remote - Push files from the local system to GitHub.



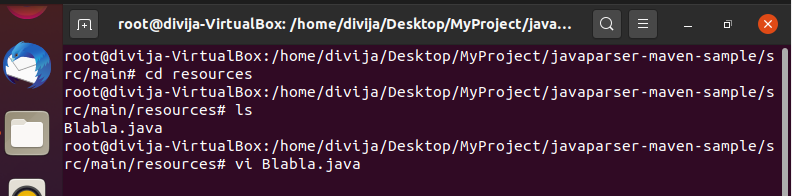


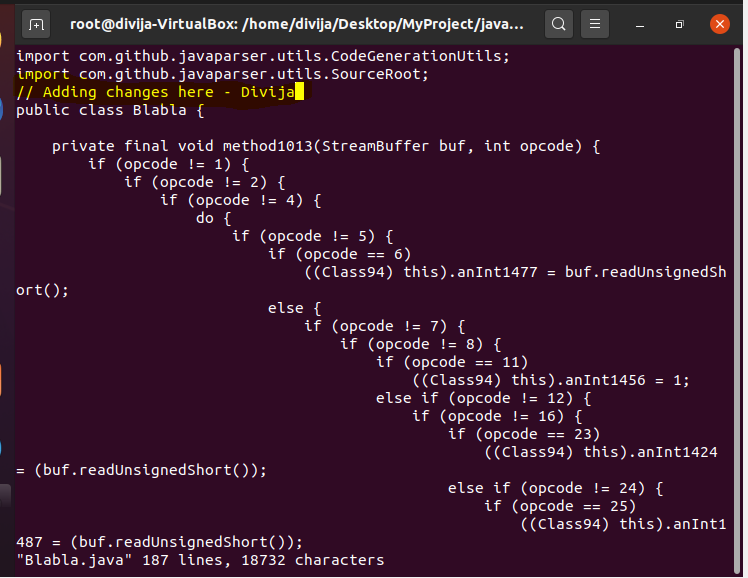
Track their versions every time a code is committed to the repository.

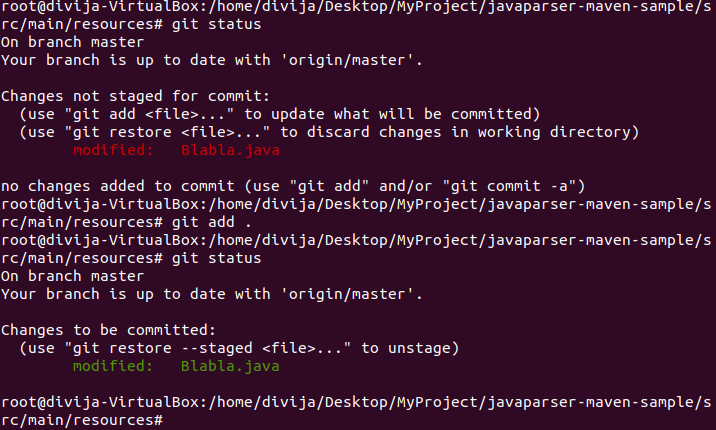
The version# 8545715

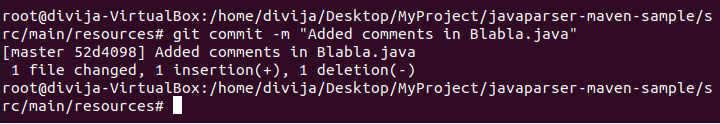


Repeated the same steps to modify other files in csharp-tutorial. Commit, push



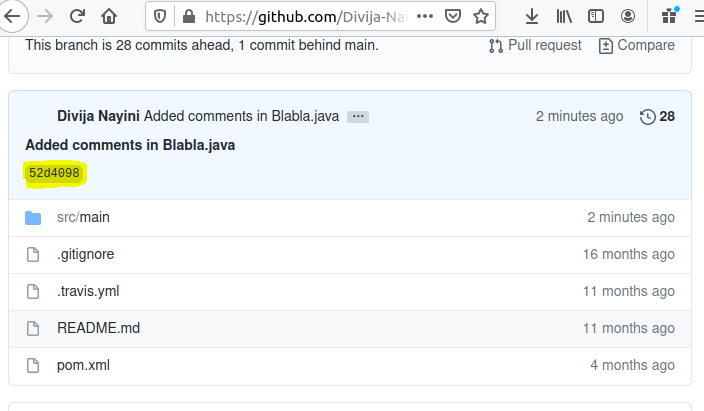




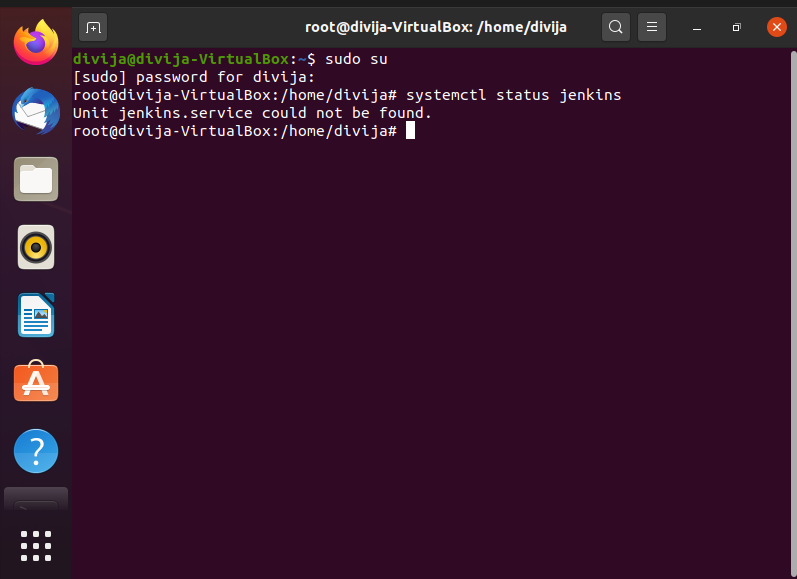


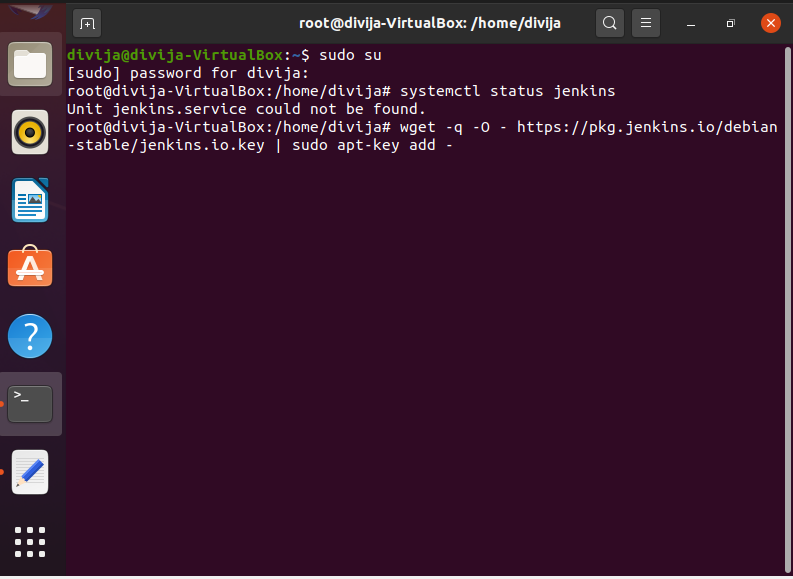


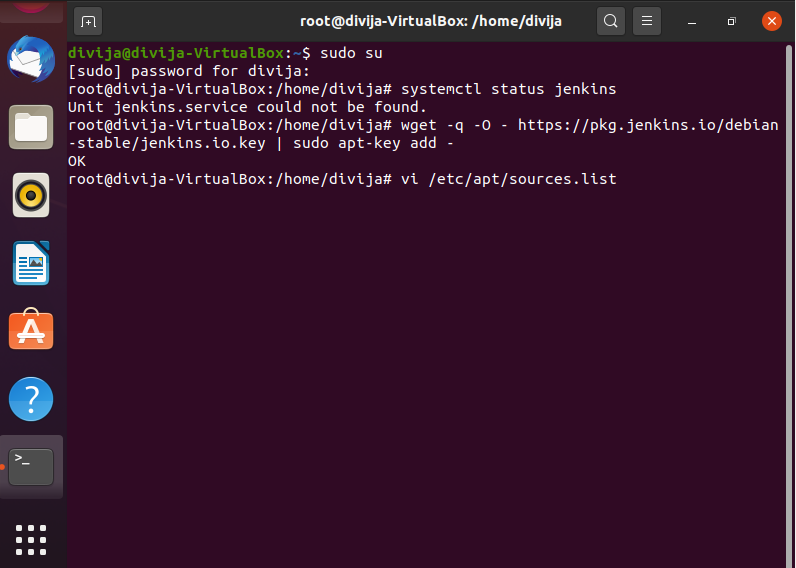
The version# 52d4098



Installing Jenkins on ubuntu







wget -q -O - | sudo apt-key add -

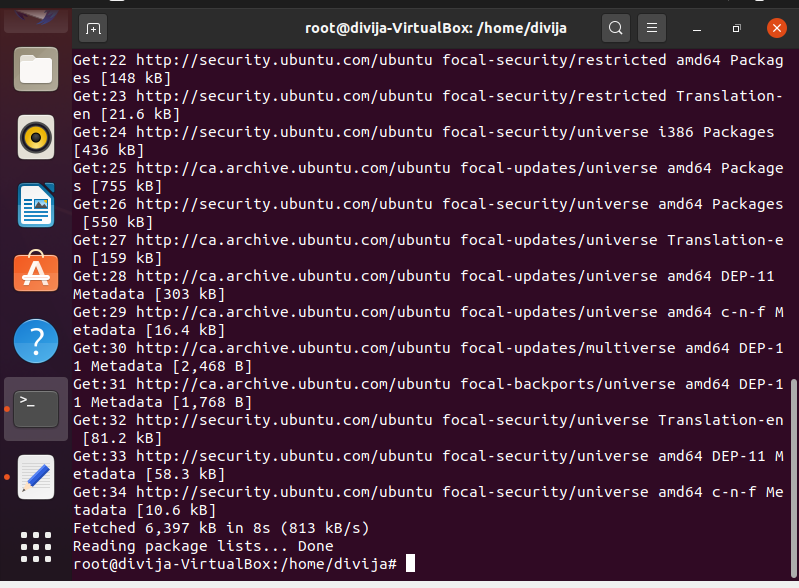
Then add the following entry in your /etc/apt/sources.list:

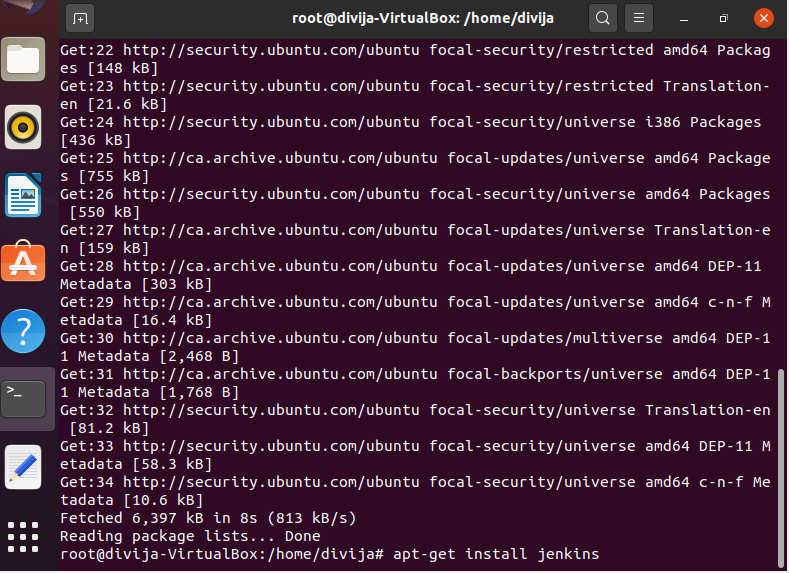
deb https://pkg.jenkins.io/debian-stable binary/

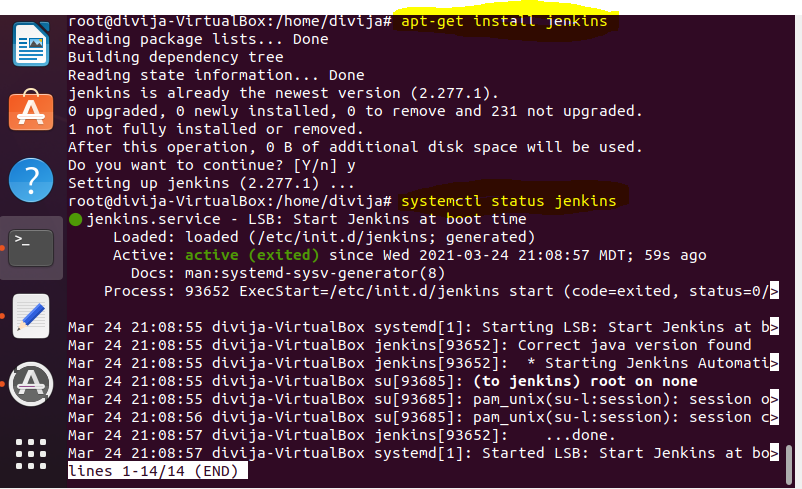
Update your local package index, then finally install Jenkins:

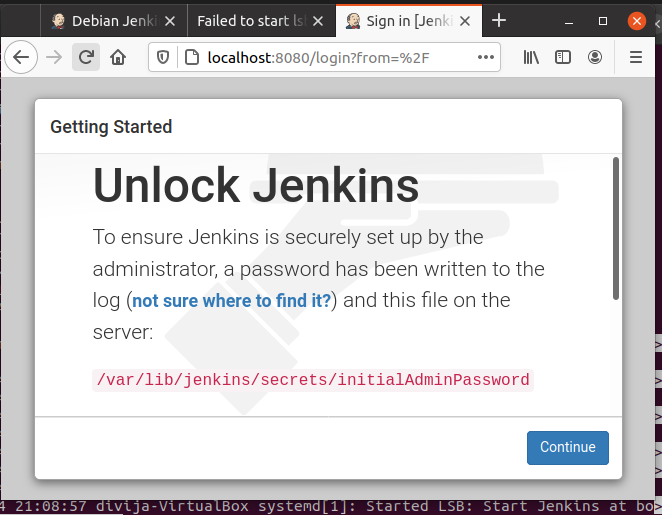
sudo apt-get update

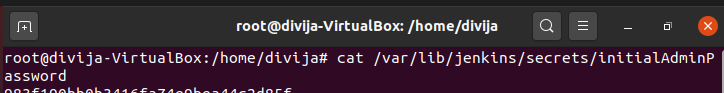
sudo apt-get install jenkins

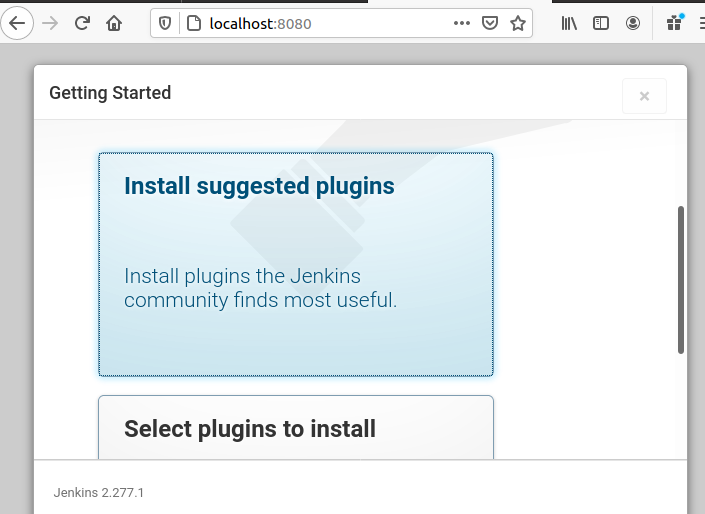


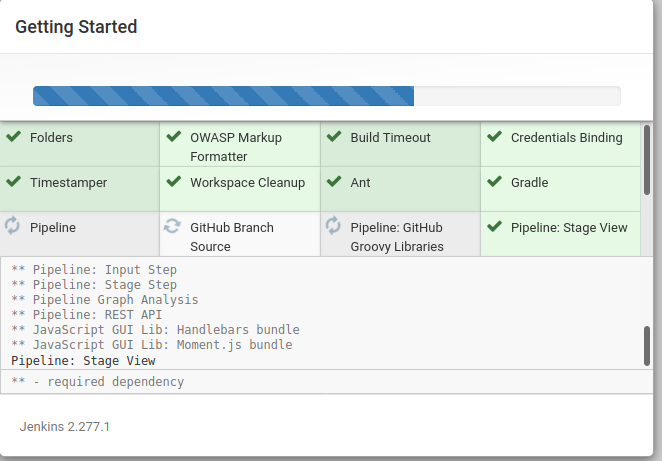


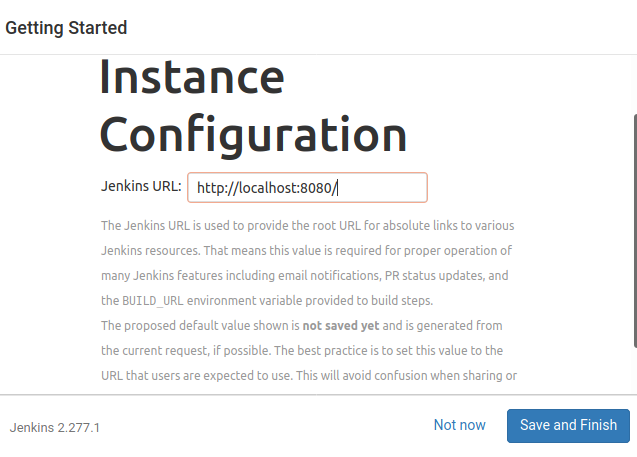






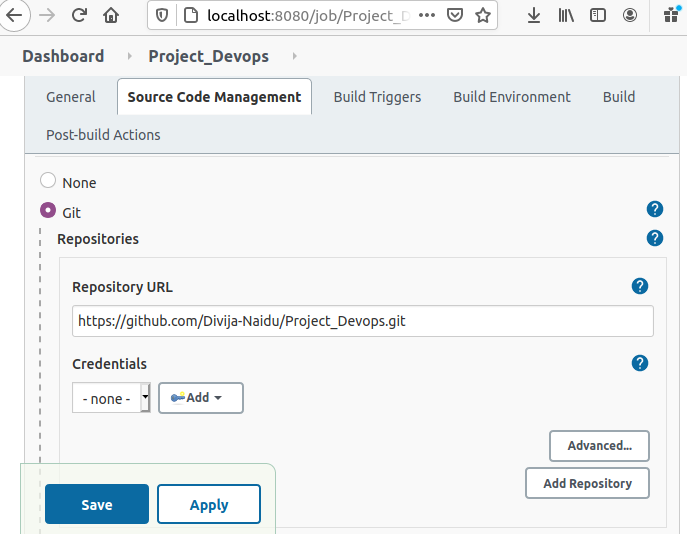


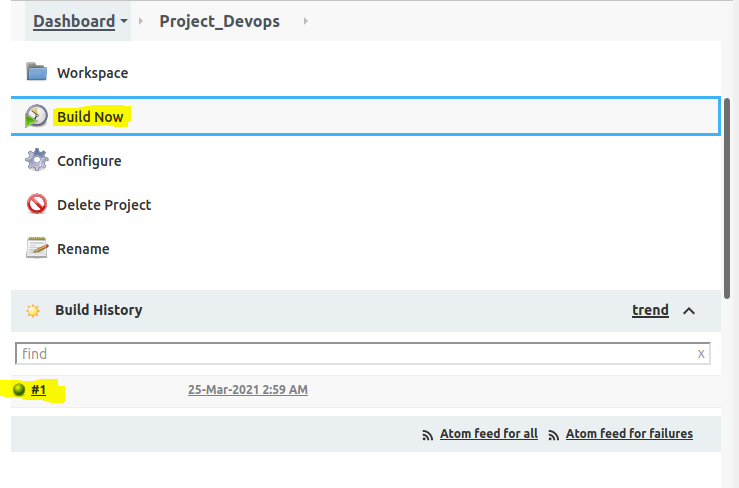


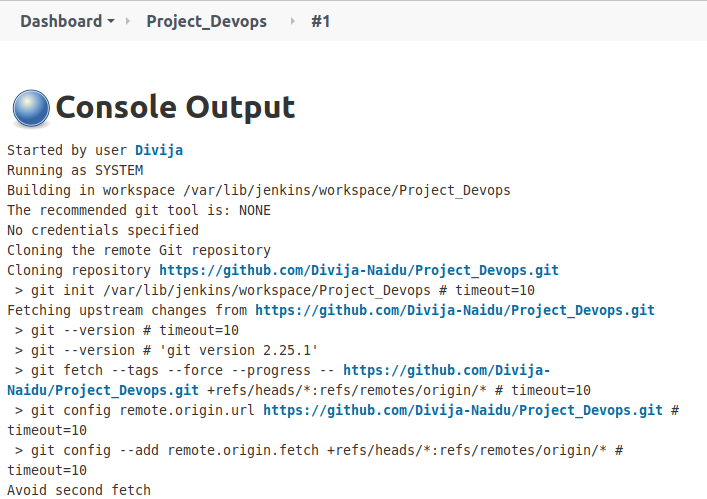




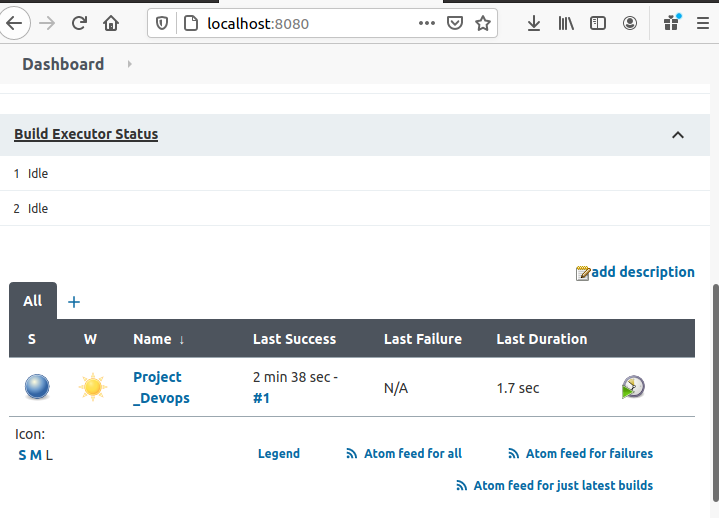
Created a Free Style project “Project\_Devops” and configured gitrepo in Source Code Management.



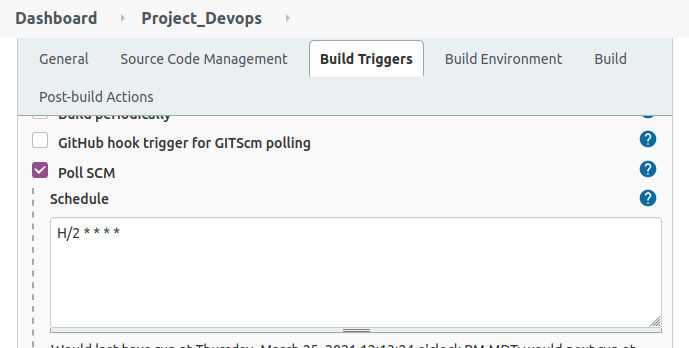




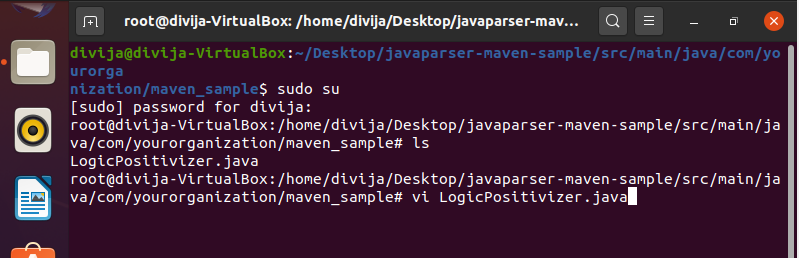


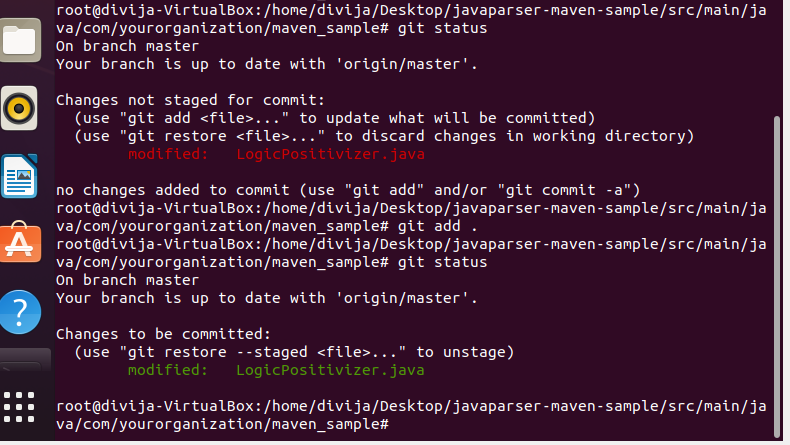


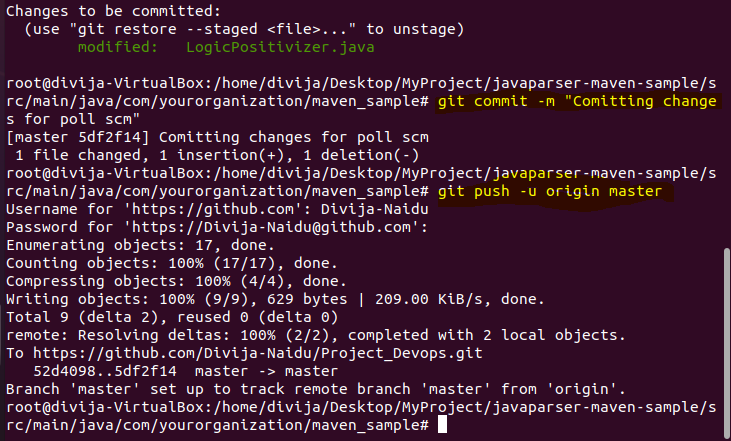
Trying poll scm build trigger option. Poll scm – check if there is a new update and then build trigger.

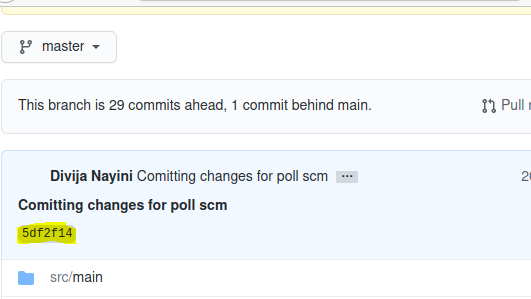


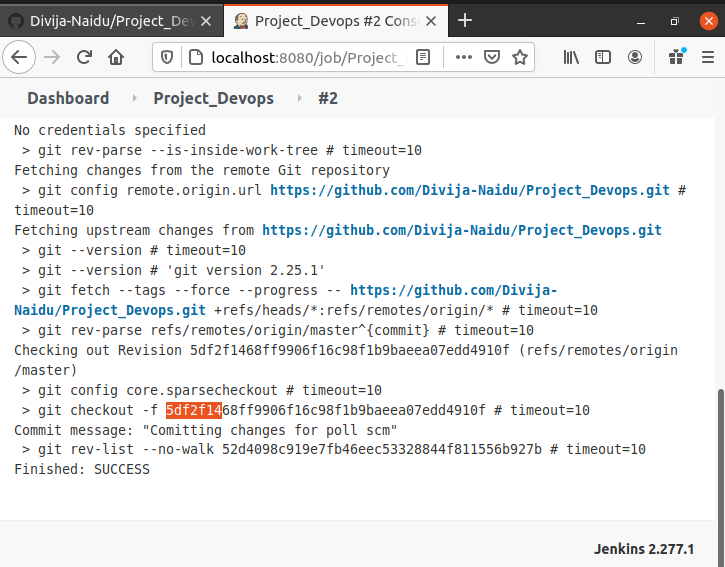
Making changes to files and commit, push to repo to run build in Jenkins.



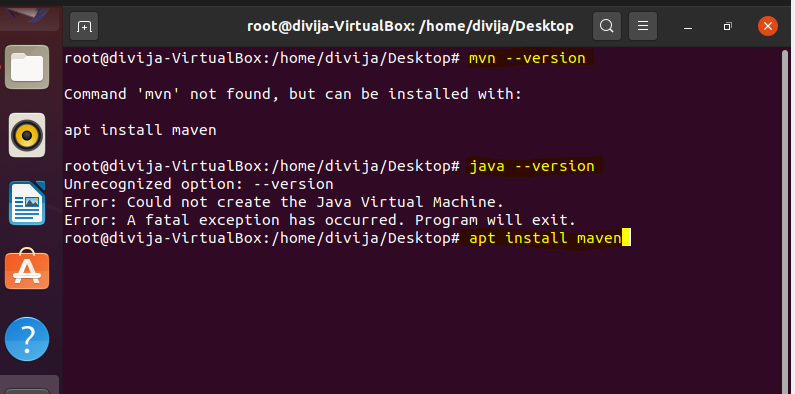


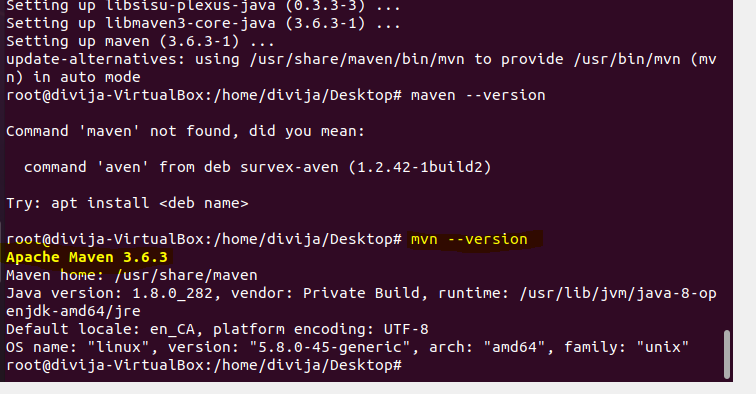




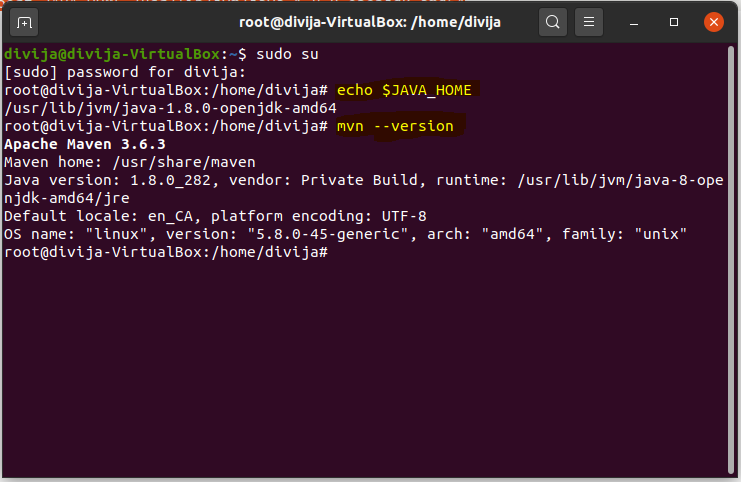


Java and maven installation for project from Manage Jenkins -> Manage Plugins

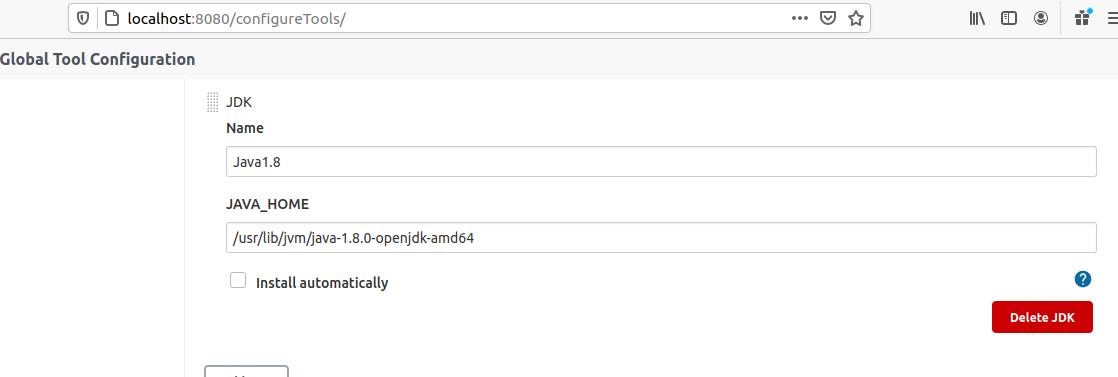


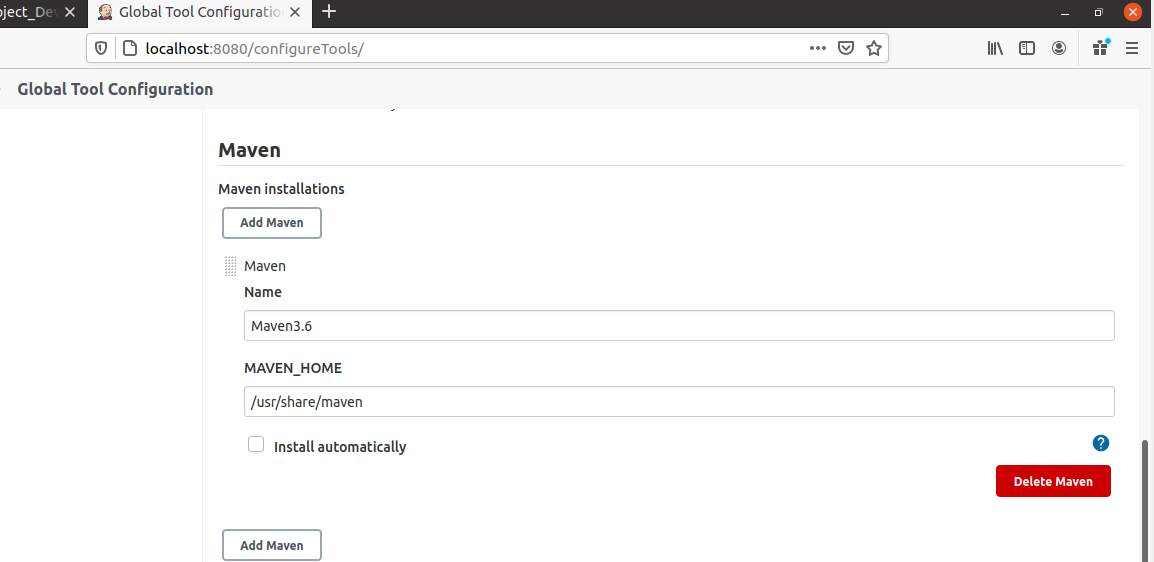


Java and Maven are installed.

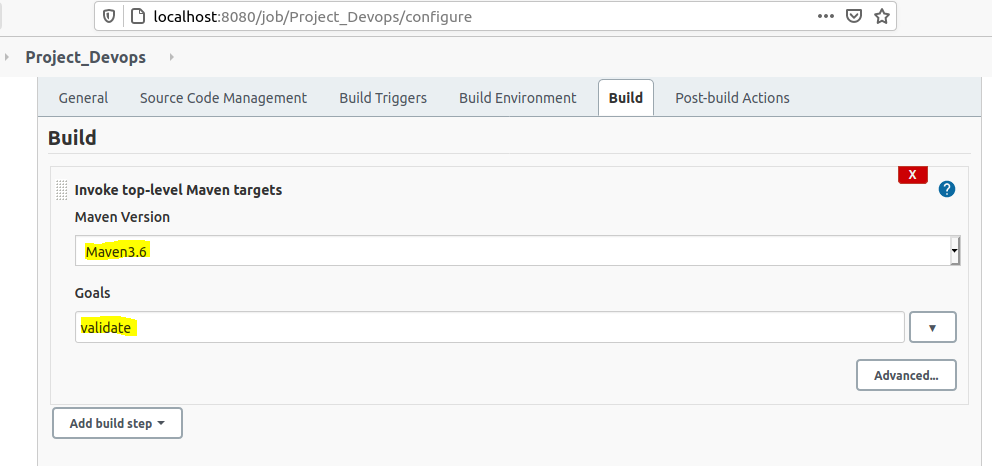


Configure Java and Maven plug-ins in Jenkins from Manage Jenkins -> Global Configuration Tool





Configure build for maven “validate”.





**Installing Docker**

https://docs.docker.com/engine/install/

# apt-get update

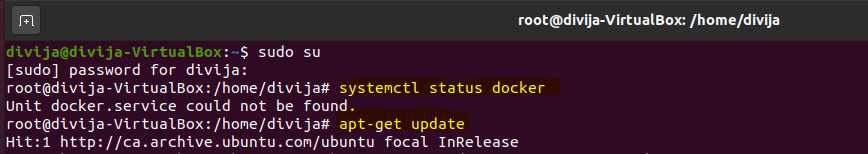
# apt-get install apt-transport-https ca-certificates curl gnupg lsb-release

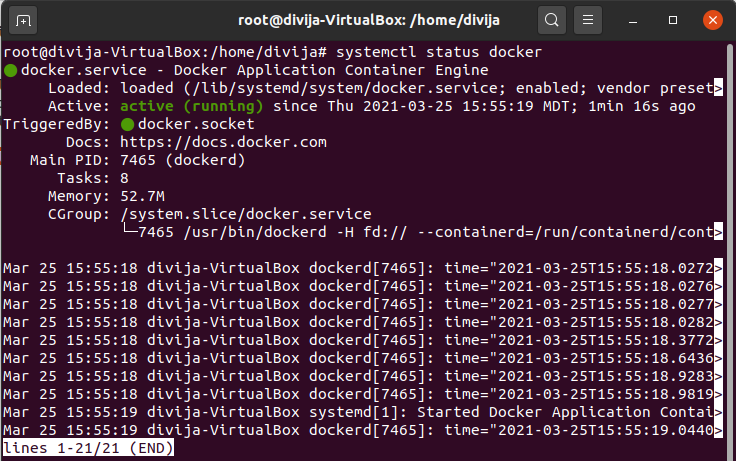
# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

# add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

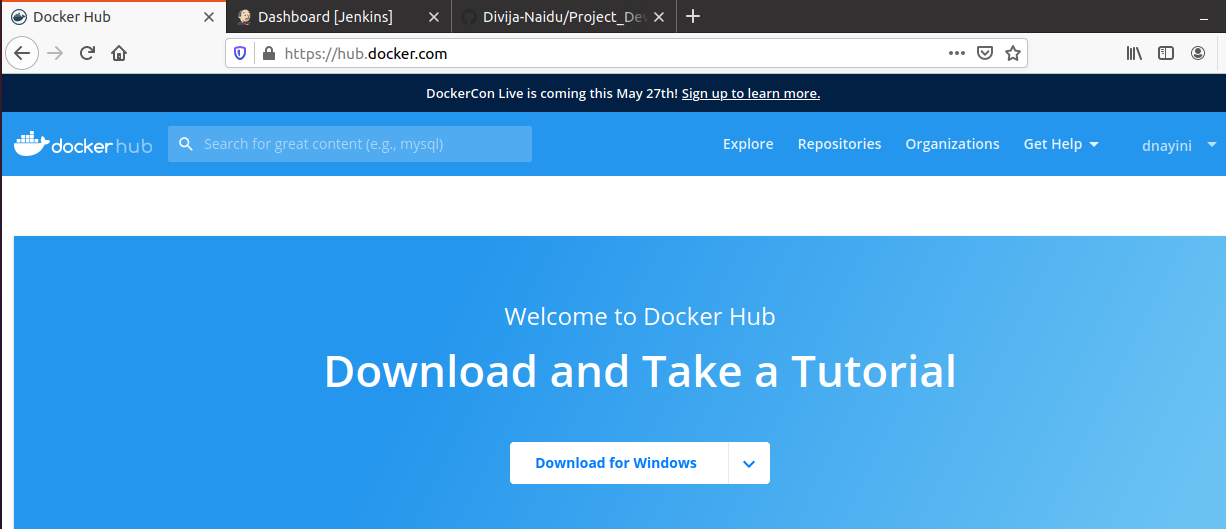
# apt-get update

# apt-get install docker-ce docker-ce-cli containerd.io

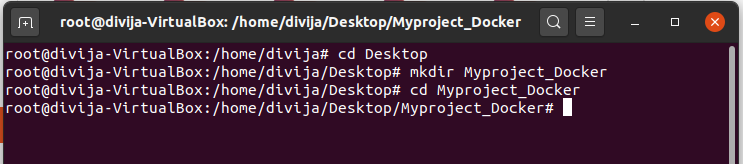




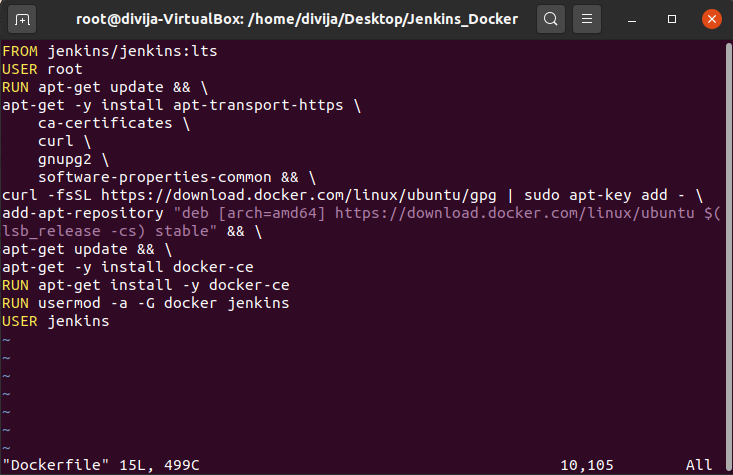
**Docker Hub Sign-in**



**Creating Docker Image:**

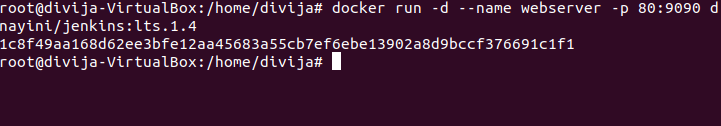


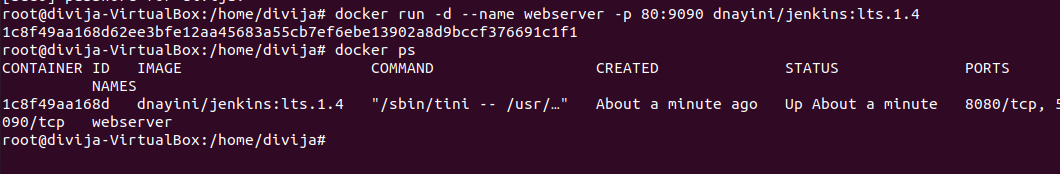
**Dockerfile**

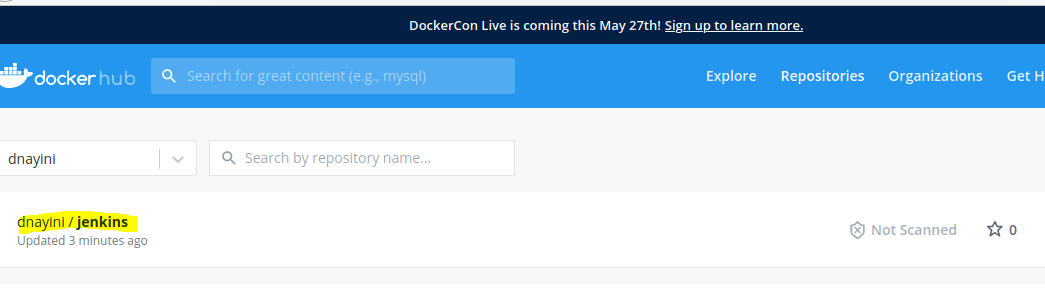




**Creating Container and Pushing image to dockerhub**



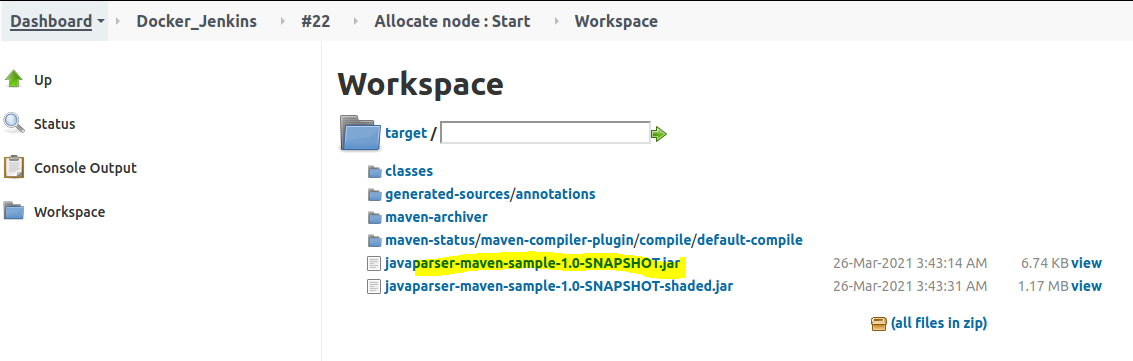




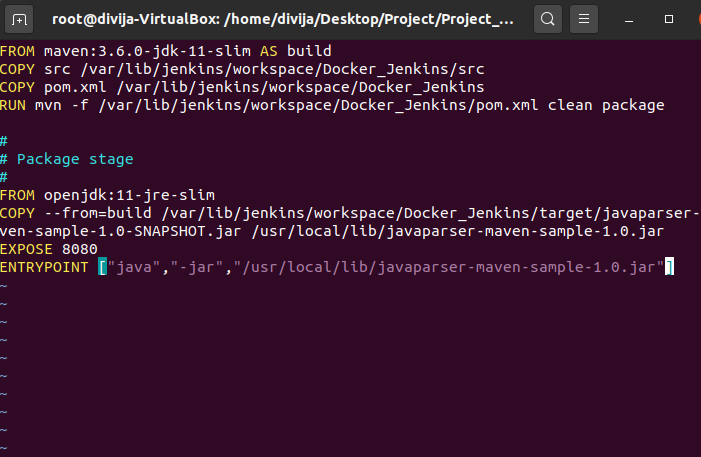
**We need Maven for the Project: This section describes Jenkins pipeline for maven stages for the continuous deploy.**



**Target/.jar file has been created**

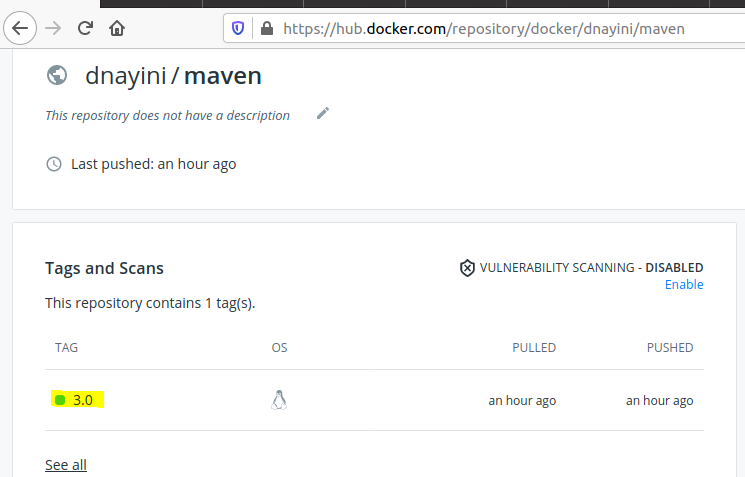


**Creating a Dokerfile in the root project directory for Maven Image & Build:**

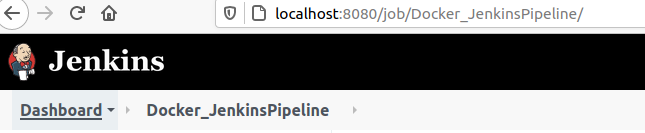


**Build and pushed the Maven image to Dockerhub**

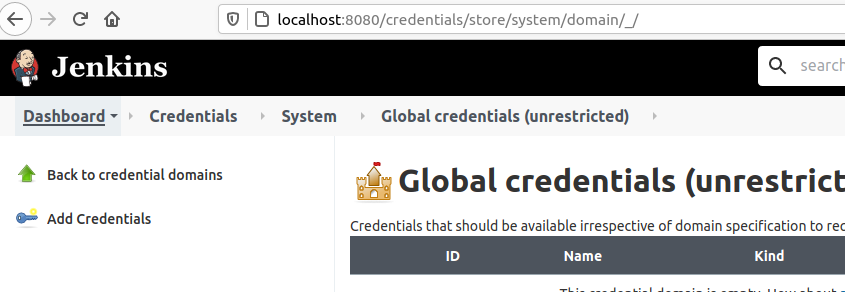


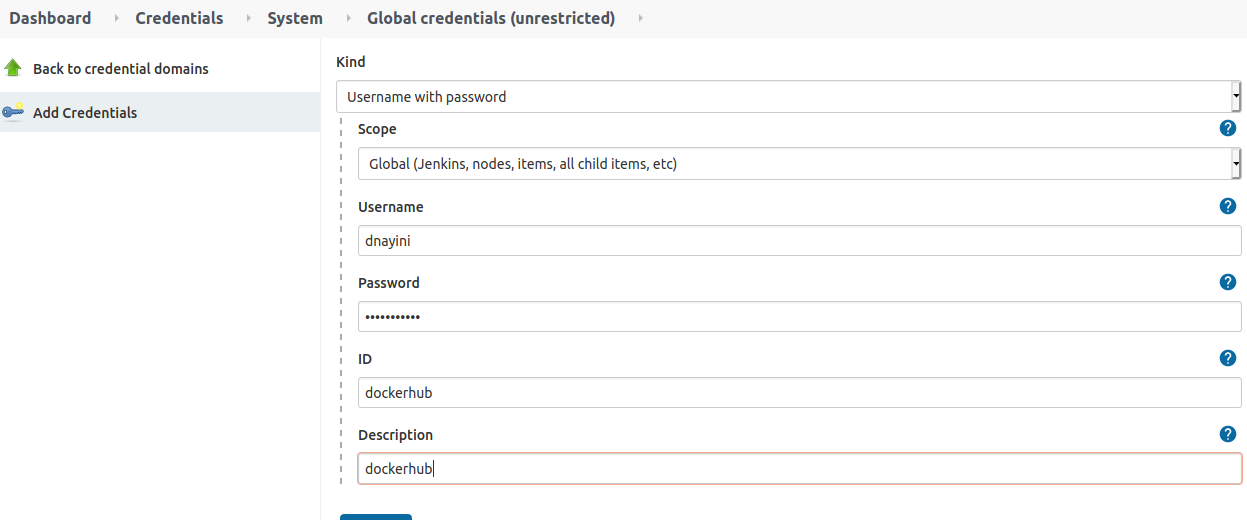


**Creating a Pipeline Project on Jenkins to test the Docker Image build, deploy with maven.**

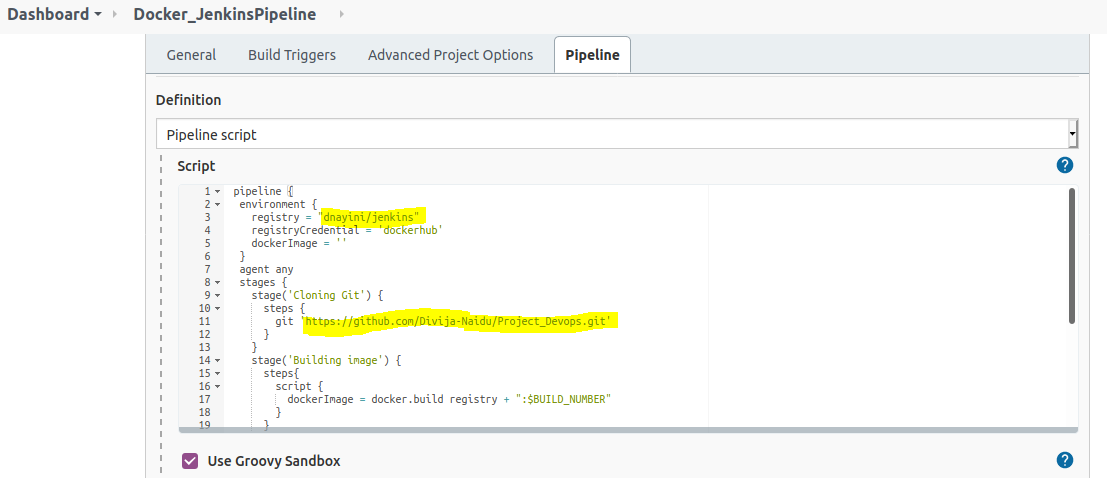


**Creating “dockerhub” user on Jenkins -> Manage Jenkins -> Manage Credentials -> Jenkins -> Add credentials**



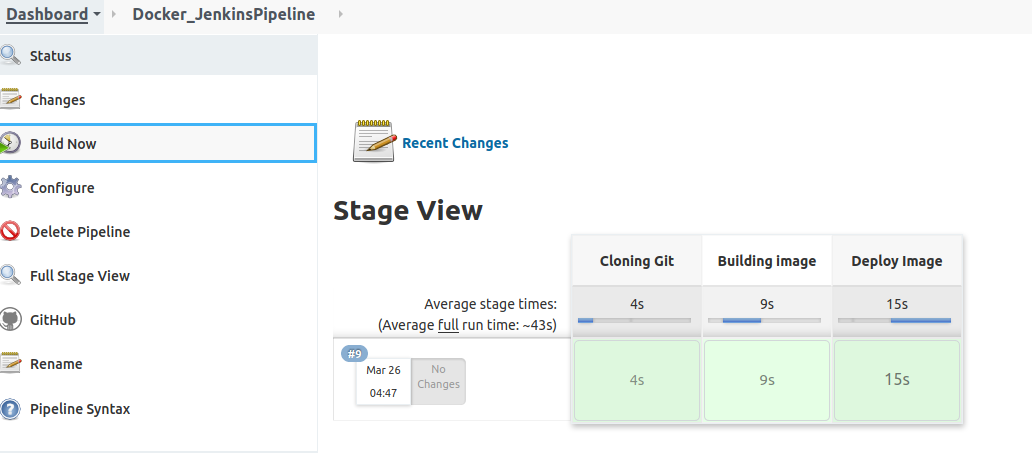


**Adding pipeline script to test docker build**



**Save -> Build Now**

**And the build is successful**



**Deployed Image**





