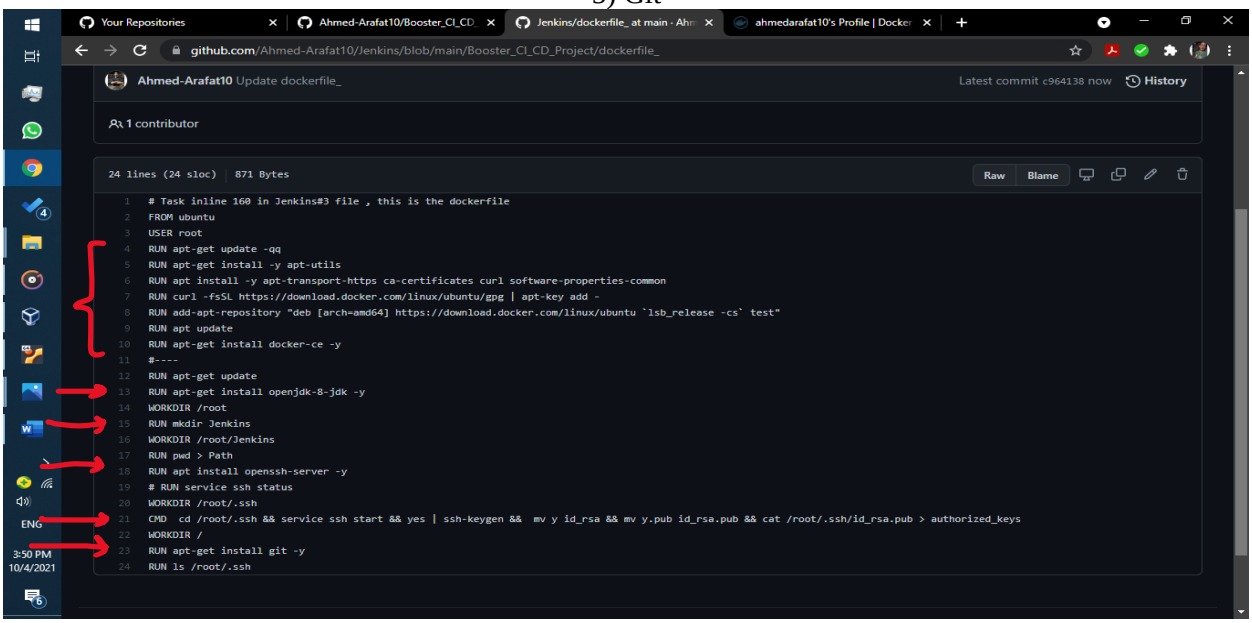
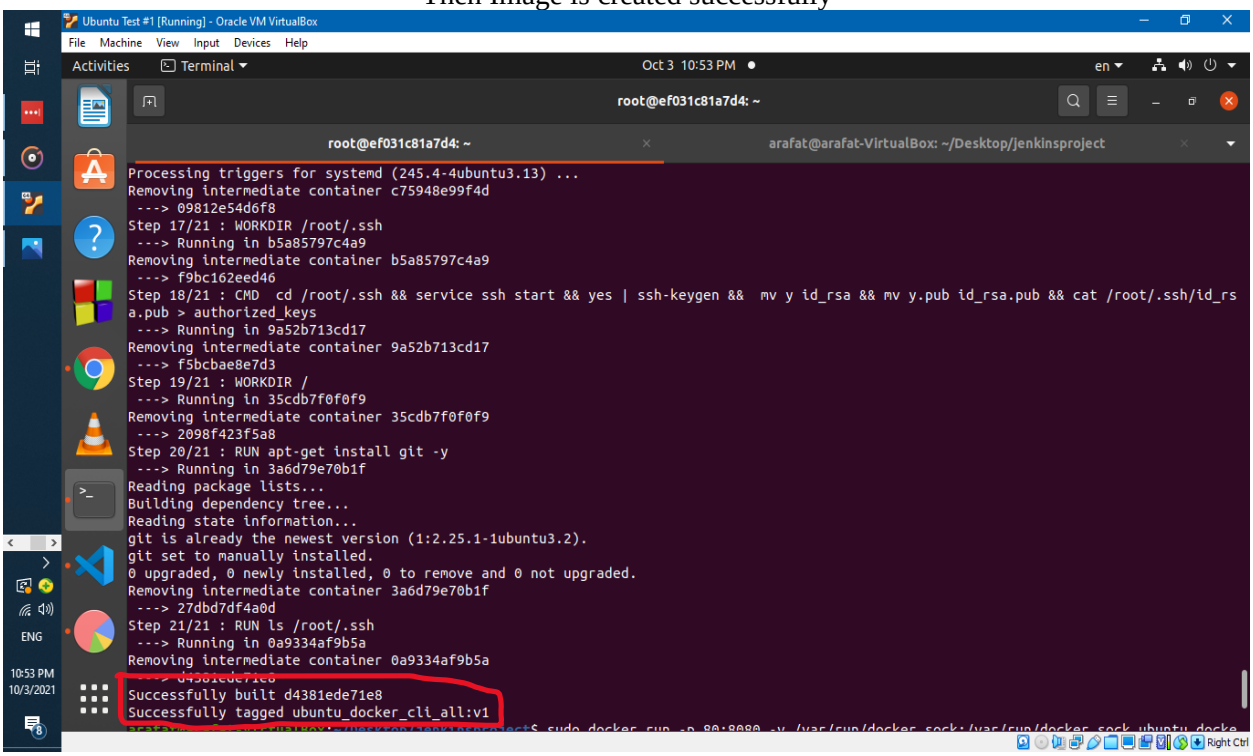
GitHub Repo's Link : https://github.com/Ahmed-Arafat10/Booster_CI_CD_Project.git

First of all, I have to create a {dockerfile} to modify image of ubuntu in which slave node in Jenkins Server will run commands of {Jenkinsfile} on it, this image must contains :

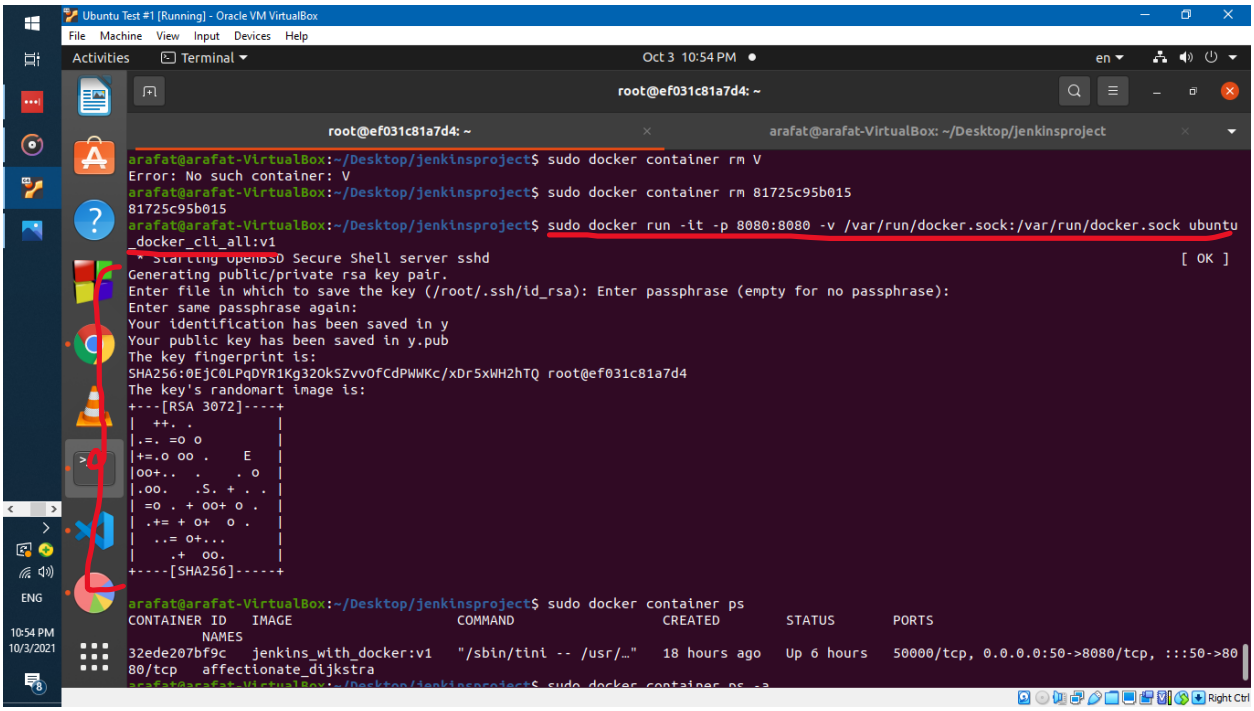
1) Docker client CLI {to run docker commands}
2) OpenJDK
3) A created Directory called {Jenkins} to save files in it
4) SSH {install it then run service when image is build into a container}
5)Authorized_keys file that contains public key in path {/root/.ssh}
5) Git



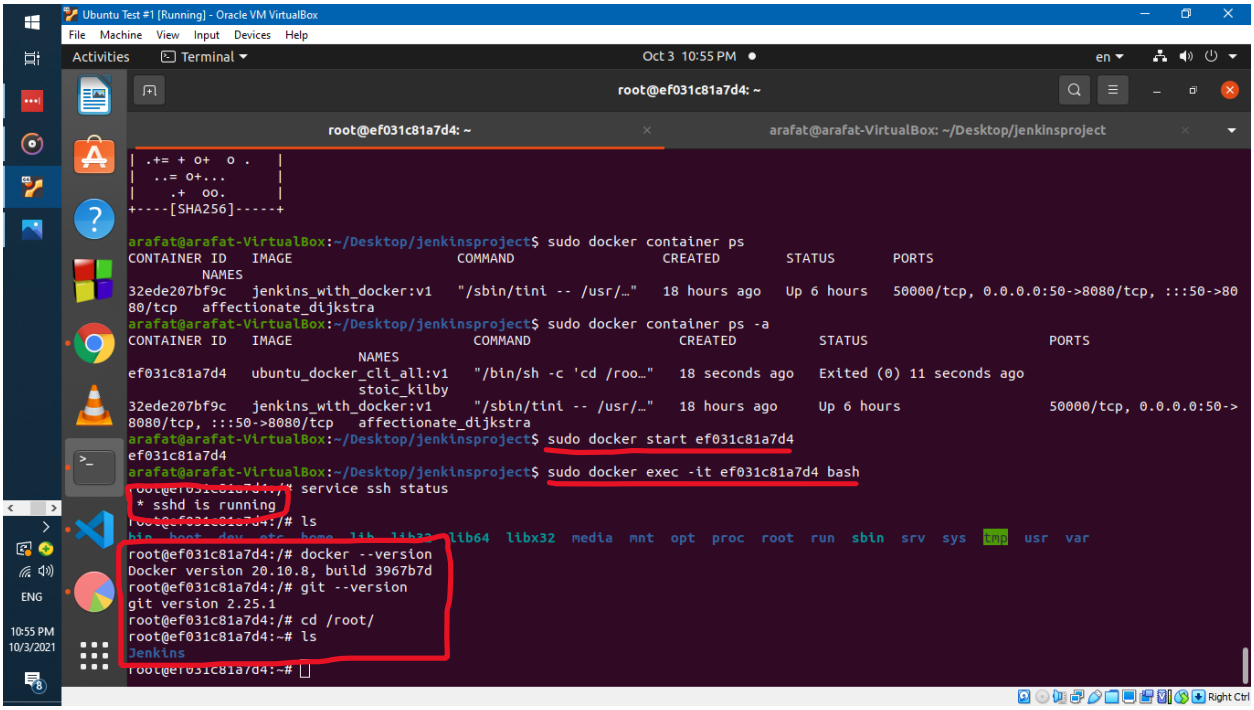
Creating an image from above dockerfile using command:
$ sudo docker build . -f dockerfile_ -t ubuntu_docker_cli_all:v1
Then Image is created successfully



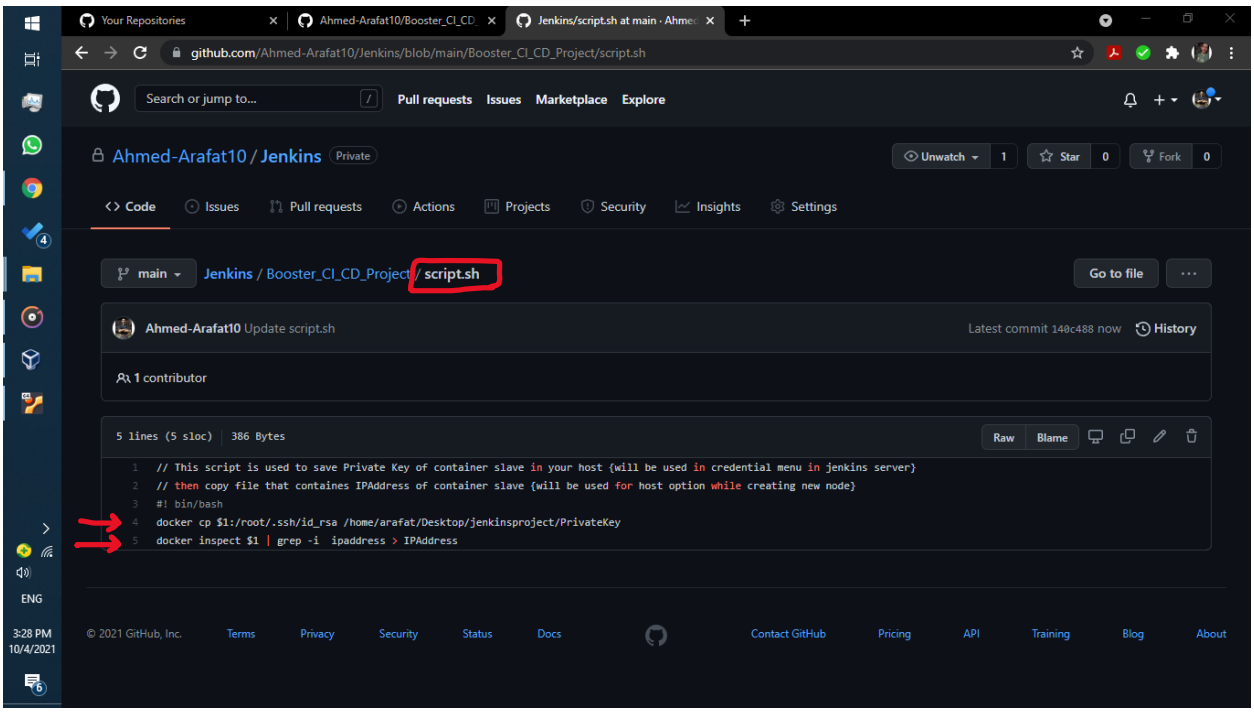
Now I can create a container from image with port [8080] refering to [8080] and a volume mapping of path /var/run/docker.sock to be able to run Docker CLI
Note: as in Line #21 in dockerfile, I used [CMD] that will run a series of commands when docker container is created These command are: enable SSH service, generating a public & private key and copying public key in Authorized_keys file



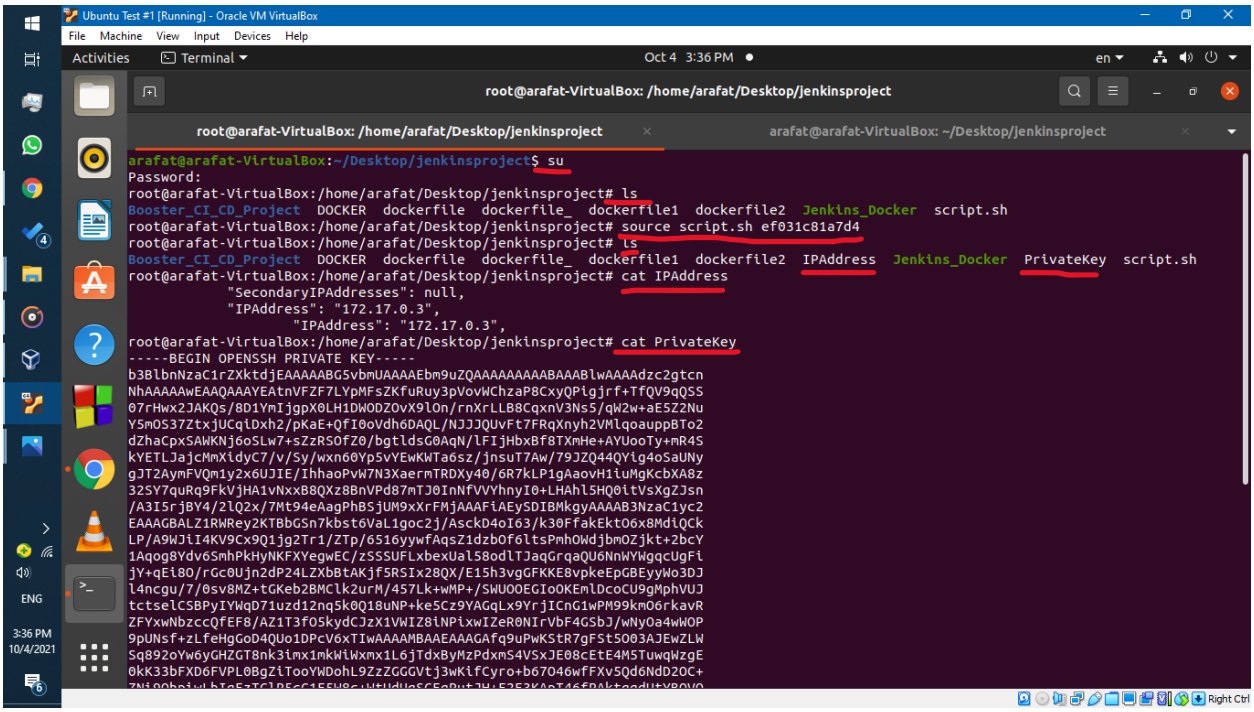
As I have exited from container & container has stopped, I have first to start it again then to enter in it's bash terminal {Enter inside container} then make sure that SSH is running, docker & git are installed and Jenkins Directory is created. Everything is working perfectly 😊



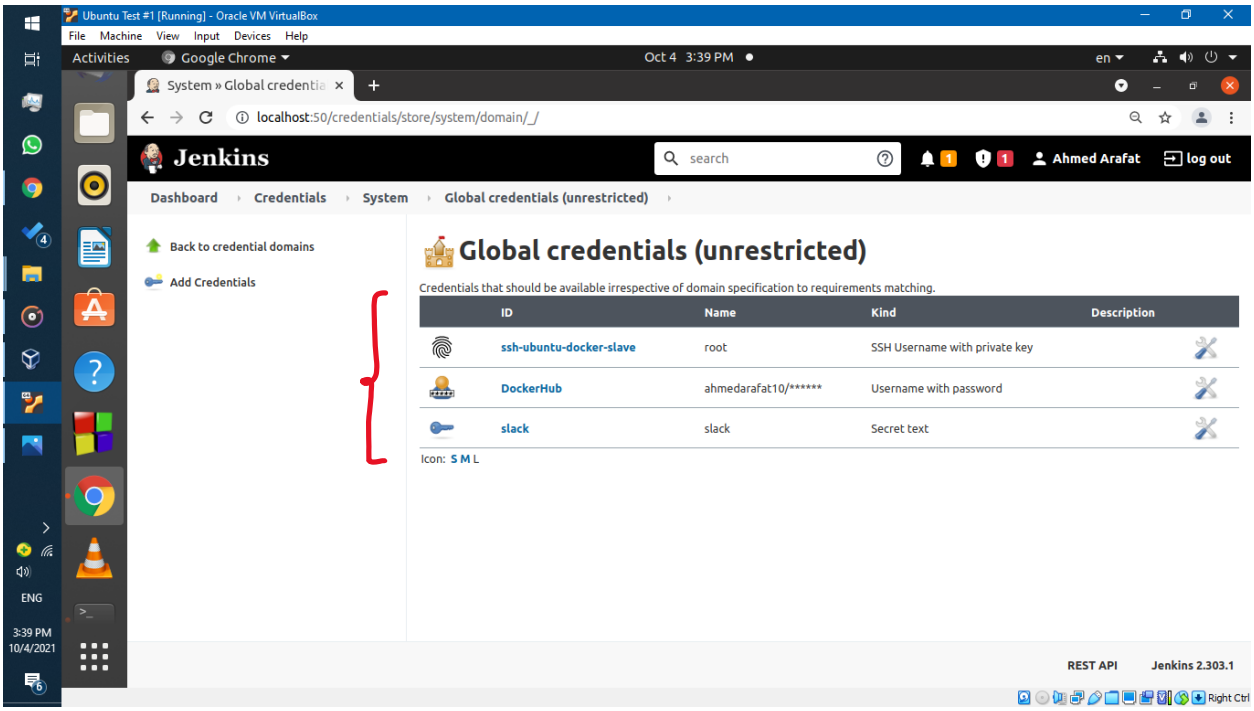
I have created a script to automatically copy Private key from container to host & then copy just lines that containes word [IPAddress] from output of $ docker inspect command {which shows all metadata of a container}and save it in a file called IPAddress. All of this can be done just by passing <Container_ID> as a parameter while executing script file



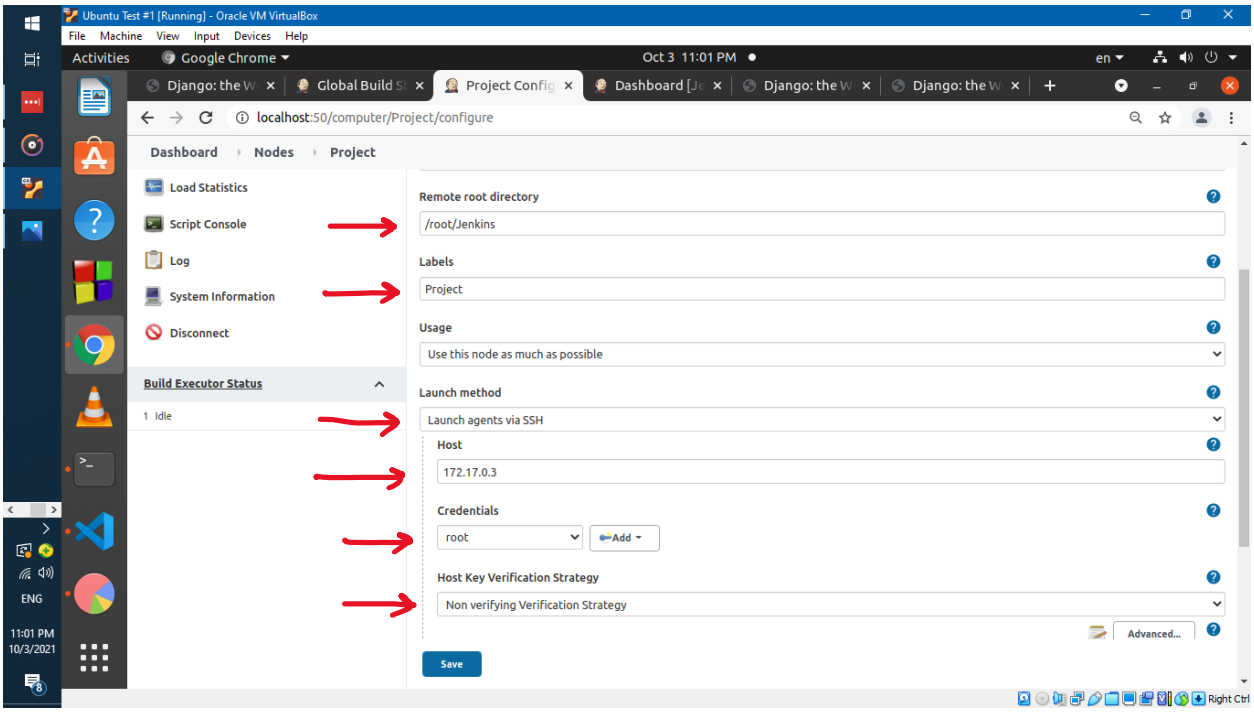
This is just implementing previous step on terminal, now I have Private key and IPAddress of Ubuntu Docker Container which will be used to create a slave node inside Jenkins Server



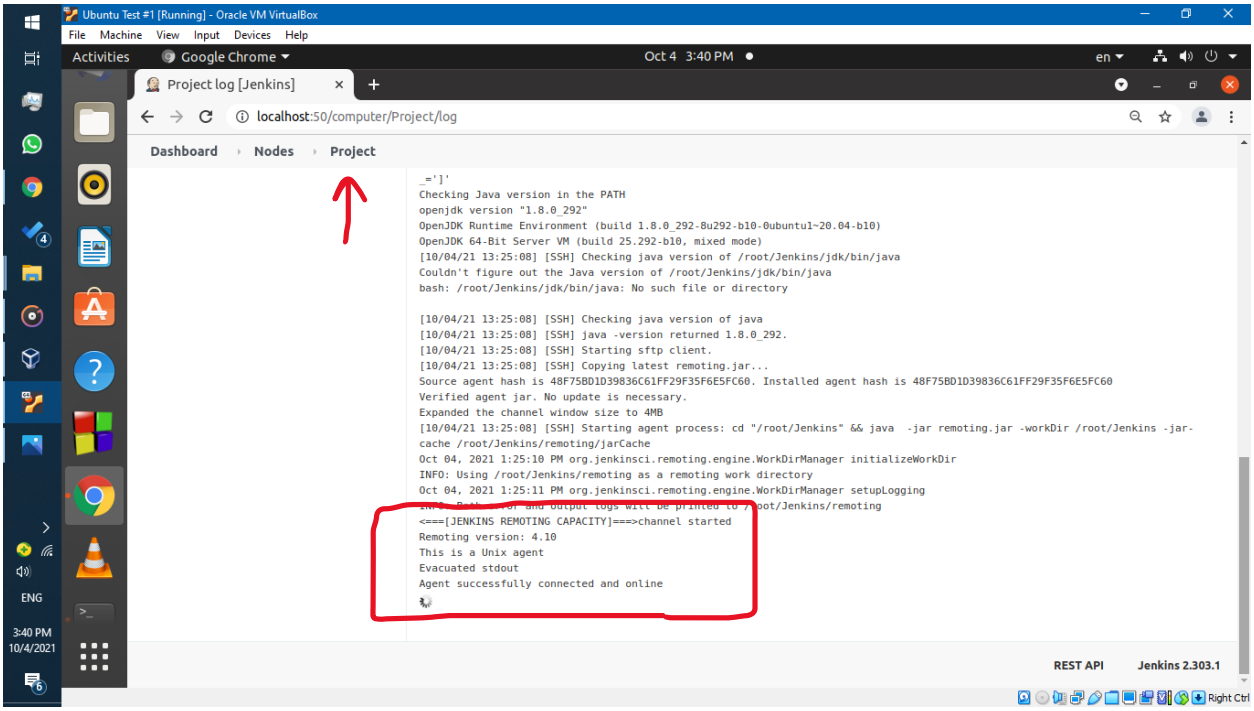
I have added all required credentials from{Credentials Menu} that will be used along project
Note: Since my GitHub repo of project is public, I don't have to add my credentials of my GitHub Account



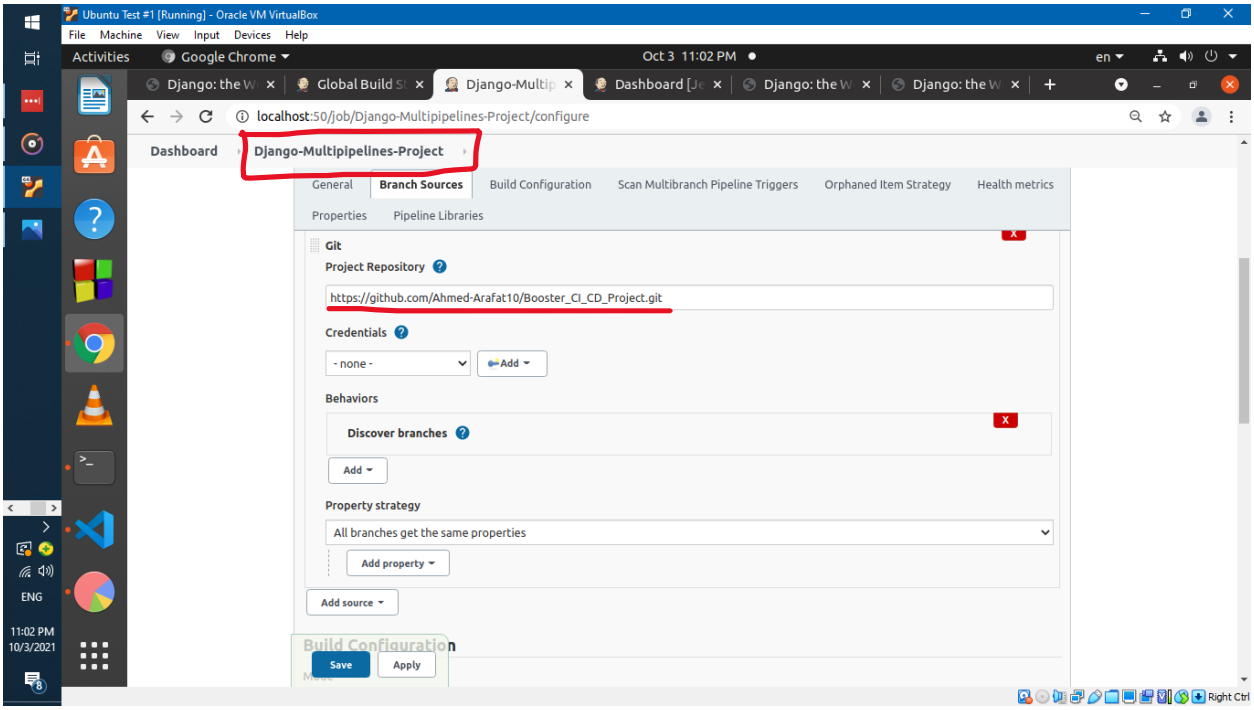
After going to {manage node menu} and then creating a new node called "Project". Now changing remote Dir. To path /root/Jenkins then copying IPAddress to host field
And then choosing credential named {root}
Note: Host key verification strategy option -> Non verifying verification strategy



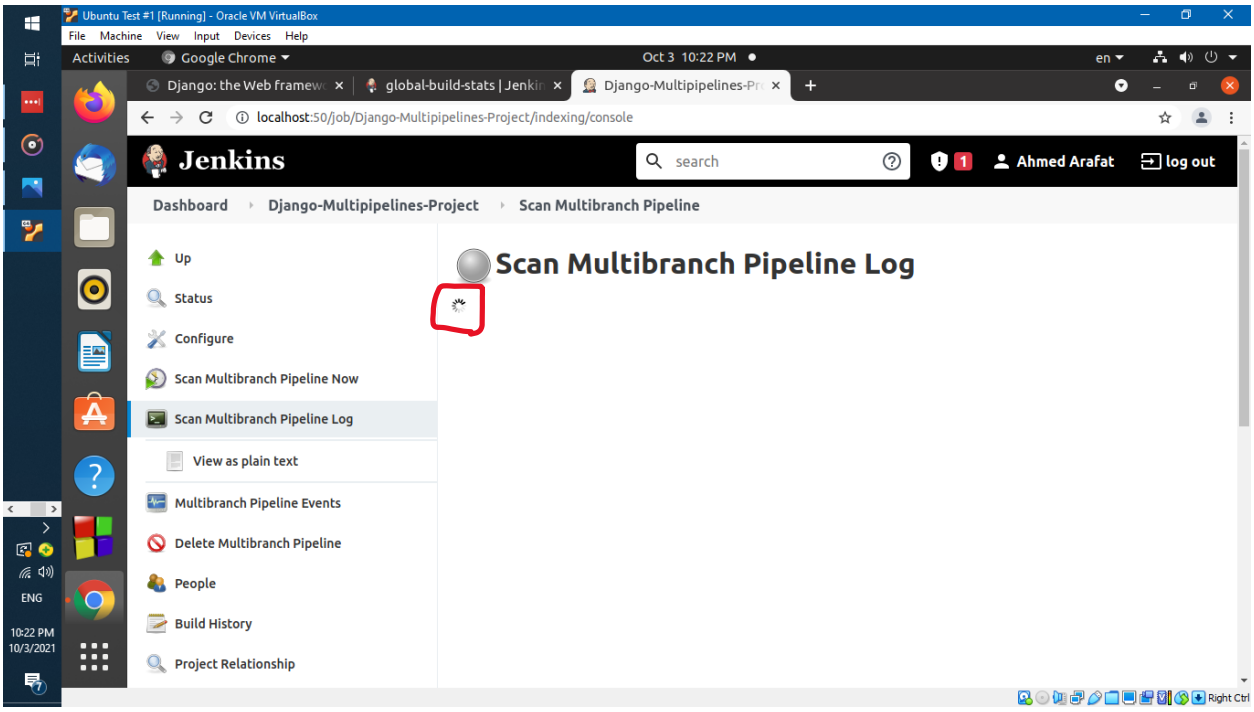
After saving and then creating “Project” slave node
Node Is created & online successfully 😊



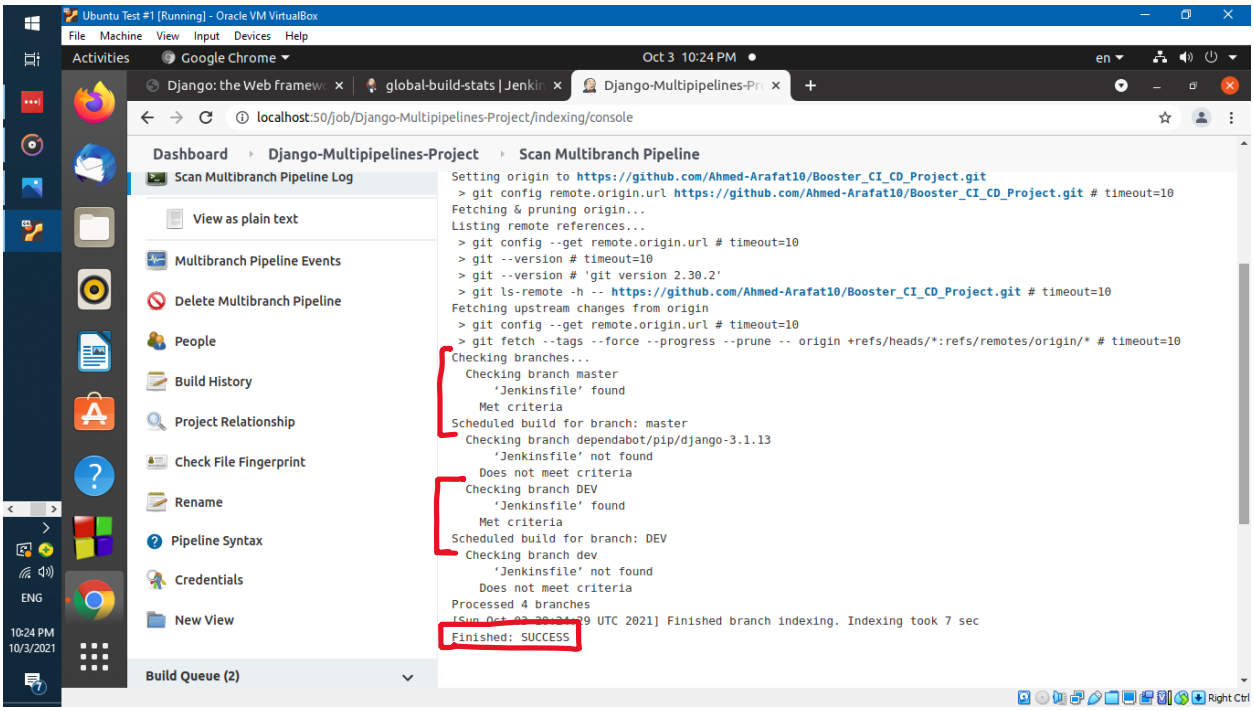
Now go to {New item menu} then create a Multibranch Pipeline called “Django-Multipipelines-Project”
With git option of Project’s GithHub repo link



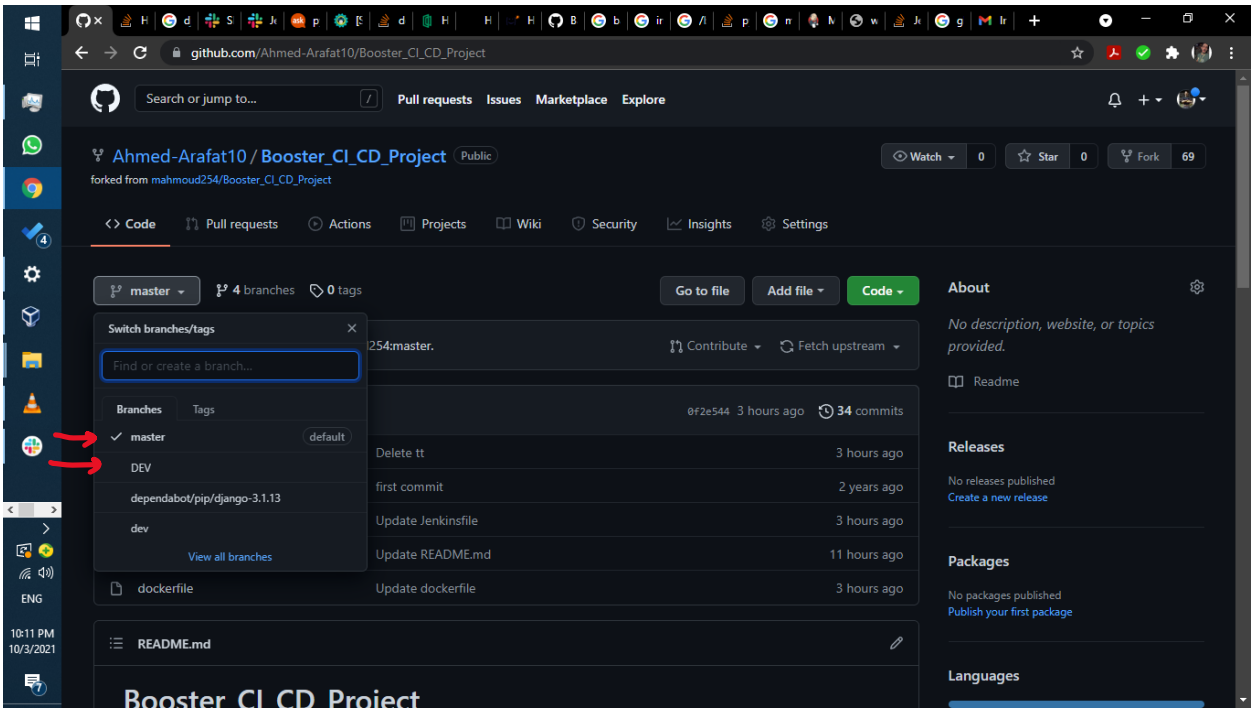
Then hit save and wait until it finishes loading



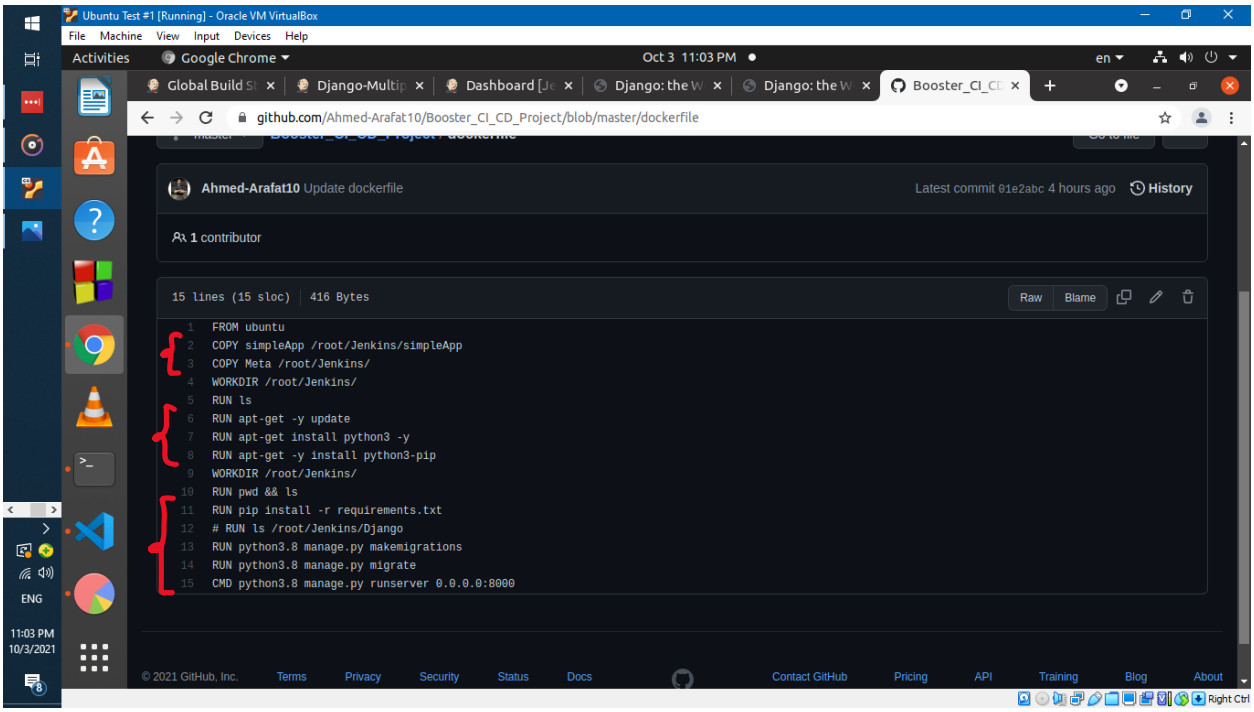
Multibranch pipeline automaticlly created Two pipelines from branches Master/DEV as both have {Jenkinsfile} in them while neglecting other branches



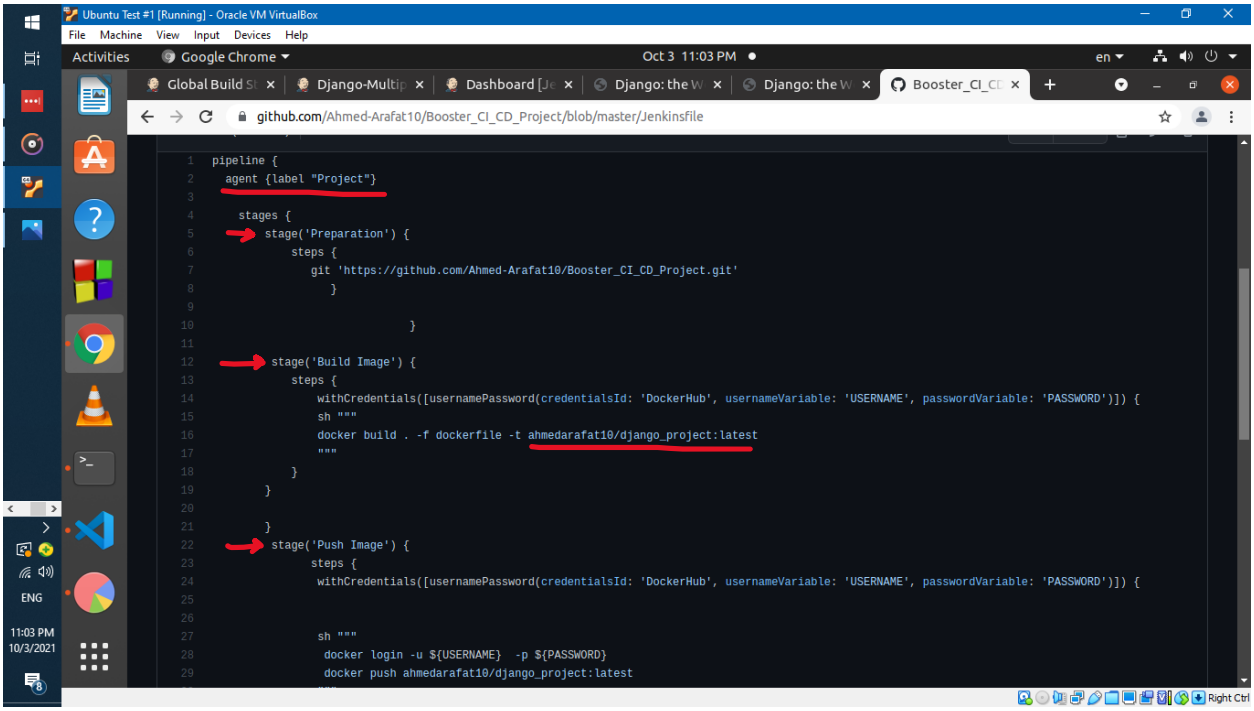
Here are Branches in Project's repo



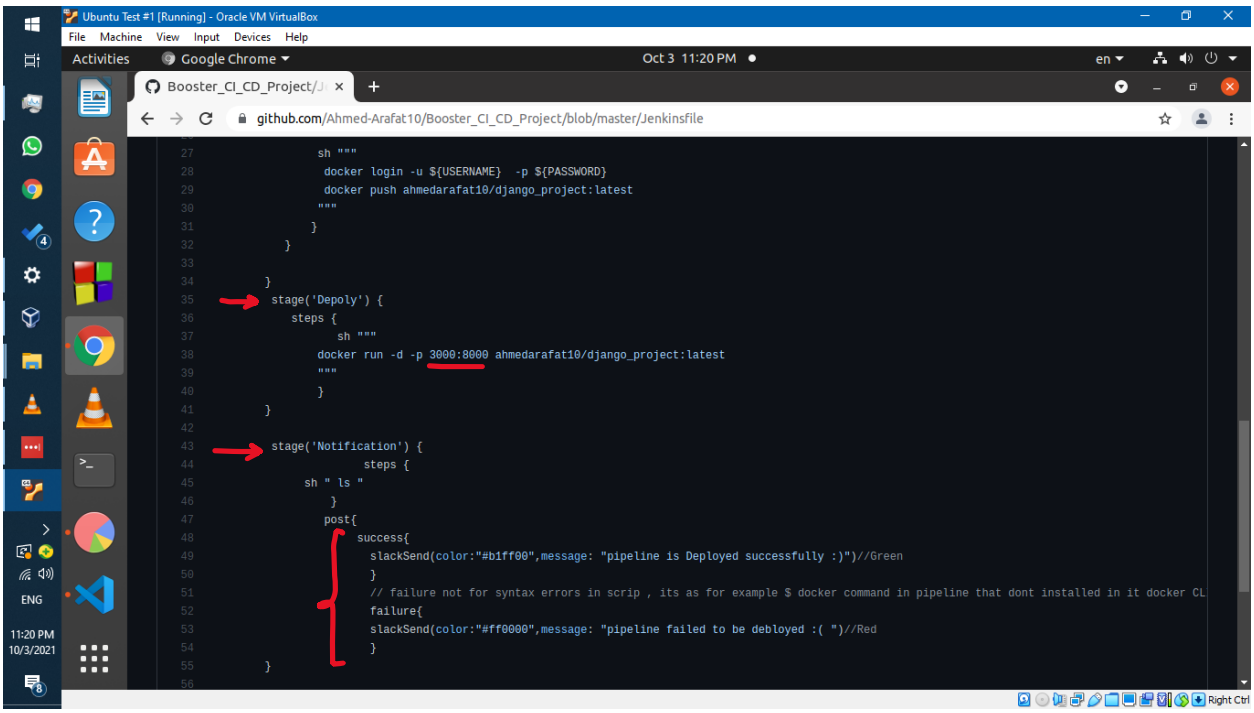
Master/DEV branches each has a {dockerfile} that will create a modified image from ubuntu image base
This modified image will install Python3 & Pip3, install required pakages for django, make migration for DataBase, apply the migrations then start the server
Note: I also can install Docker CLI in image if I want



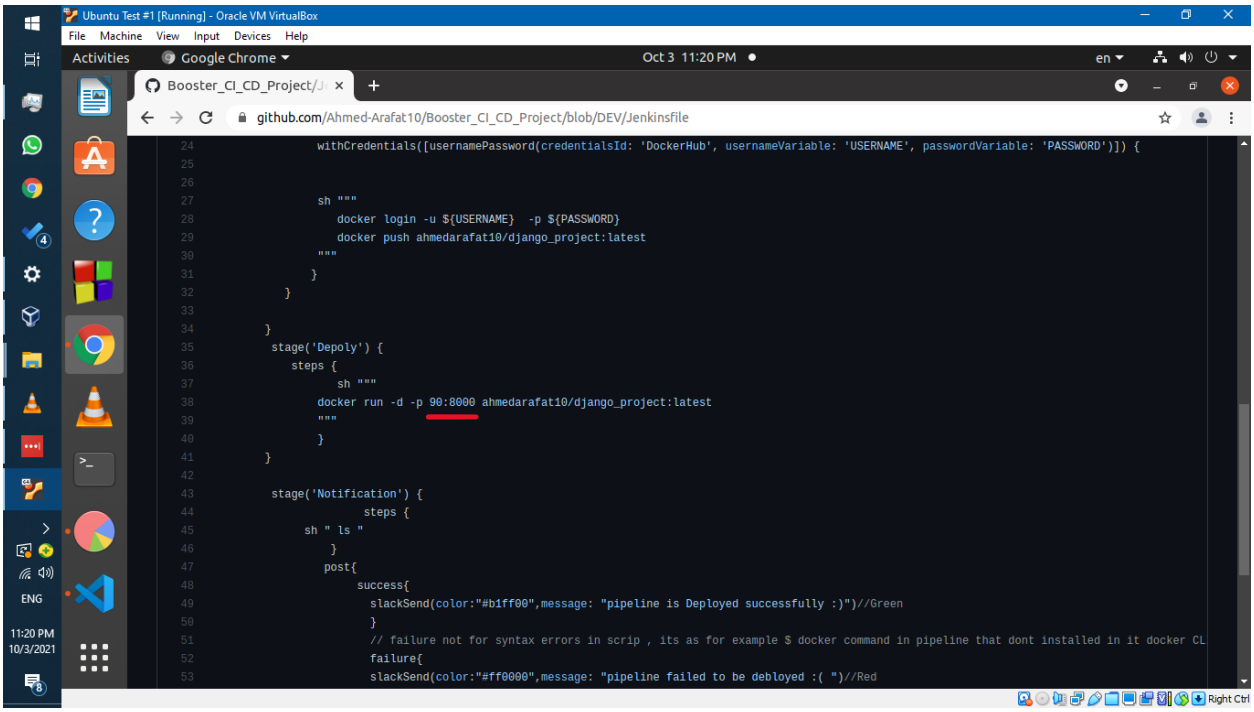
This is sample of {Jenkinsfile} that is found in both mater/DEV branches
{Just Check Github repo 😊}



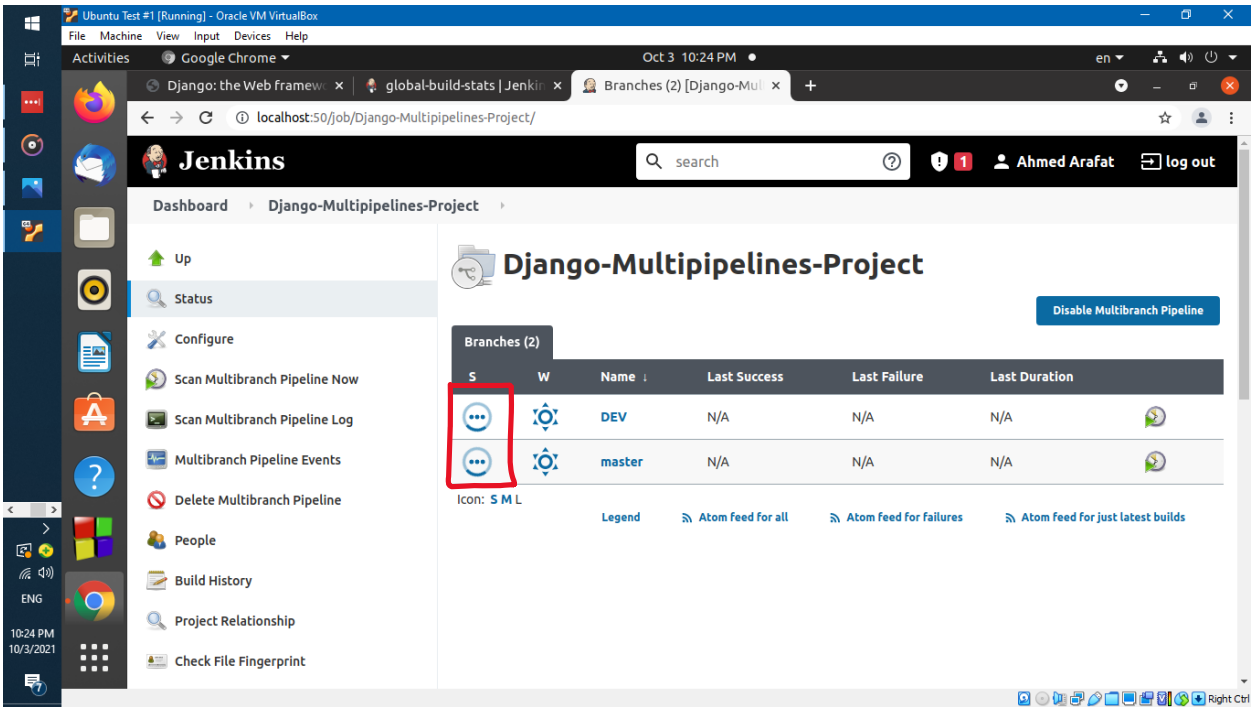
An Important Note: as I'm going to build *TWO* pipelines one for master branch and the other for DEV branch, each pipeline must have a different port referring to default Django port {8000}
So, for master branch's pipeline will have {3000} port



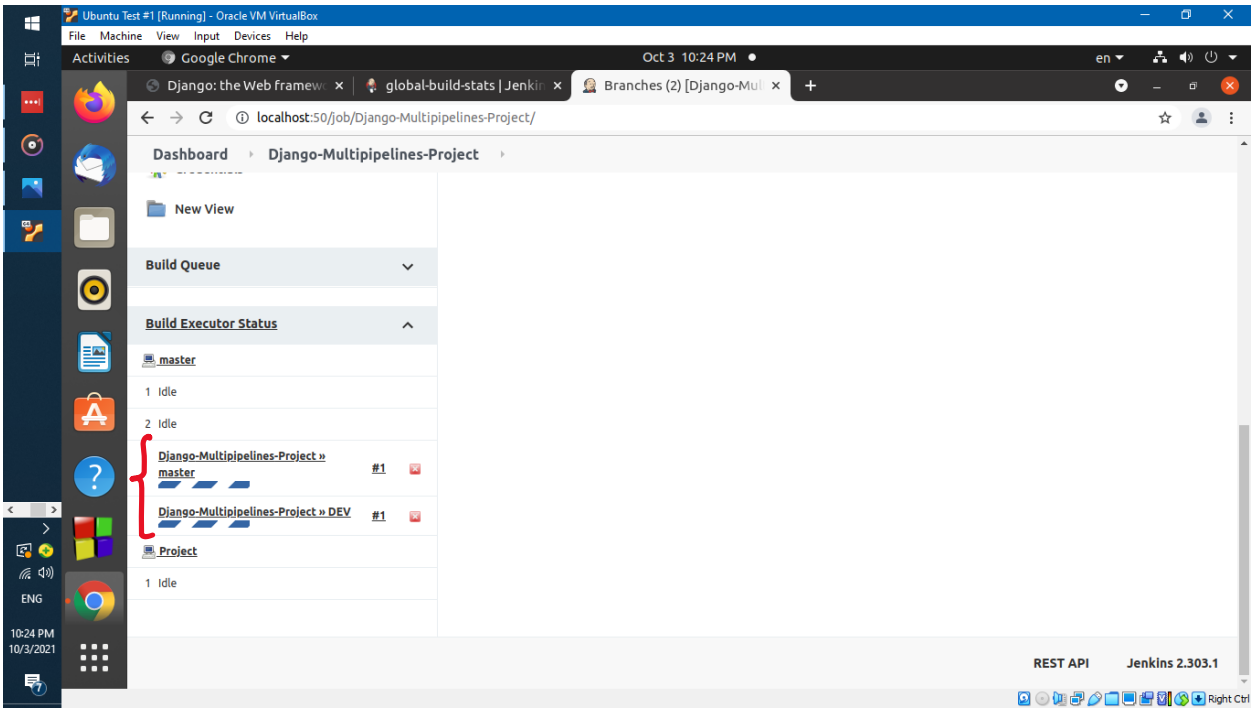
While DEV branch’s pipeline will have {90} port



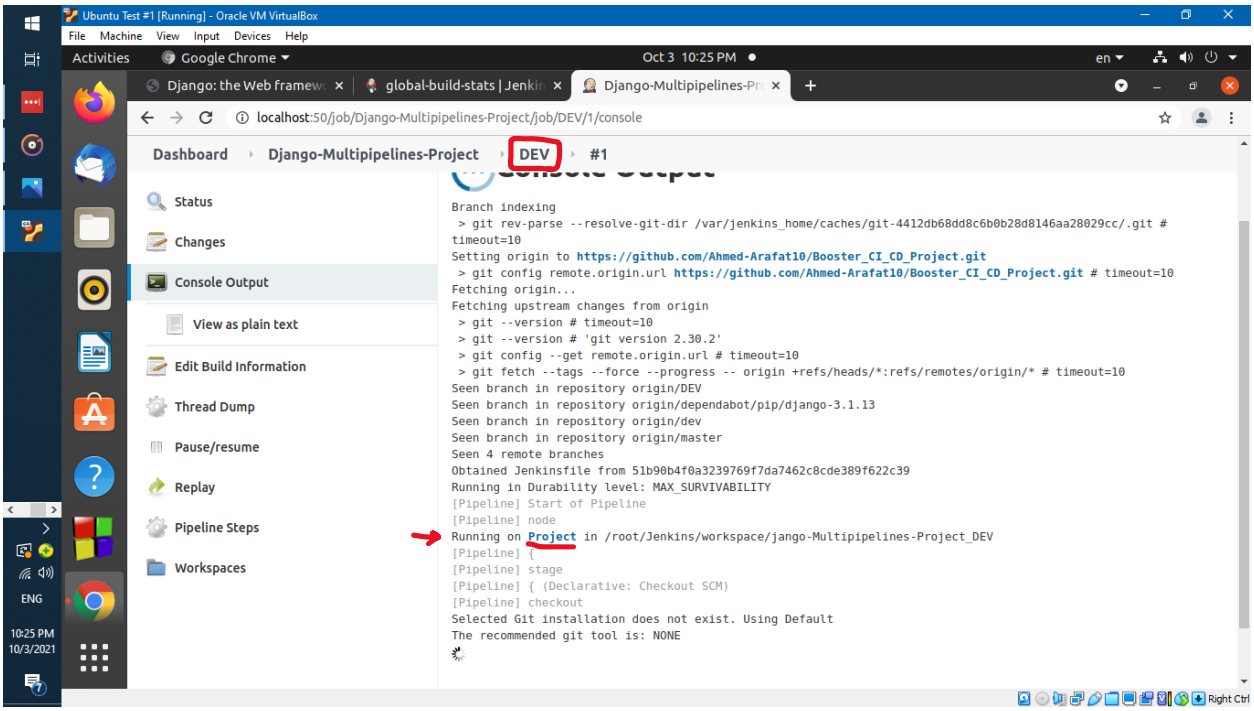
Now both pipelines are being built



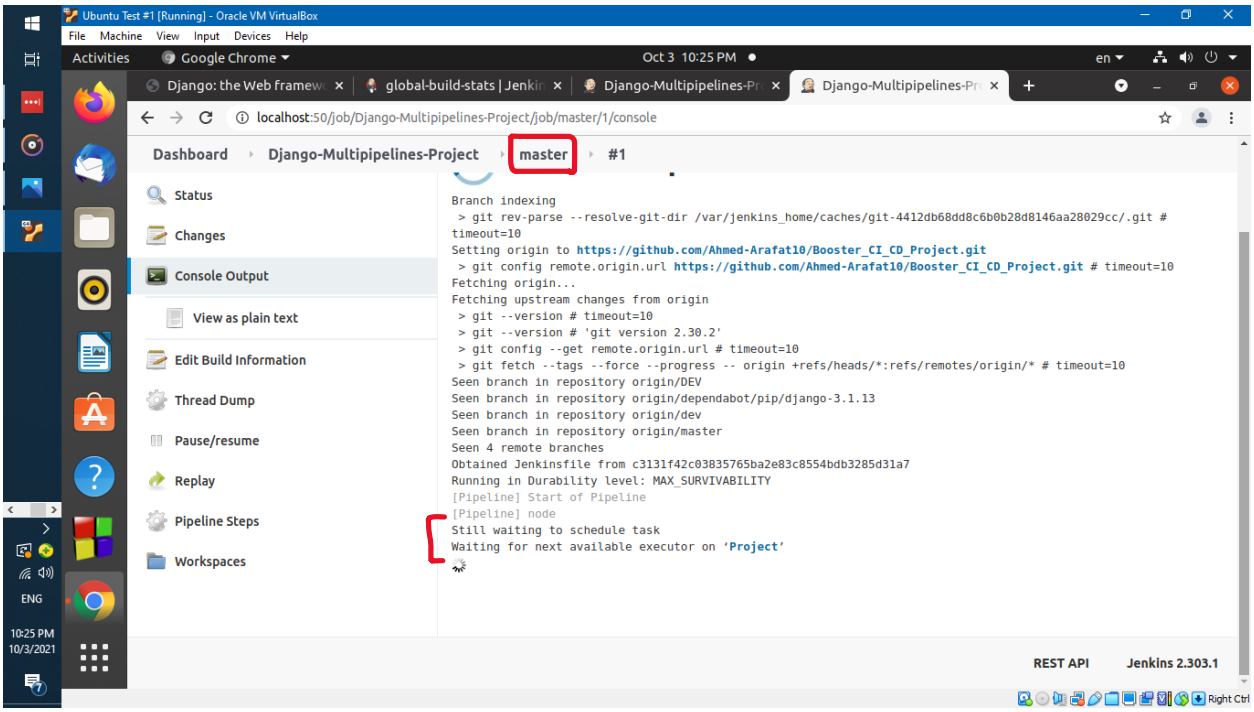
Still Loading …



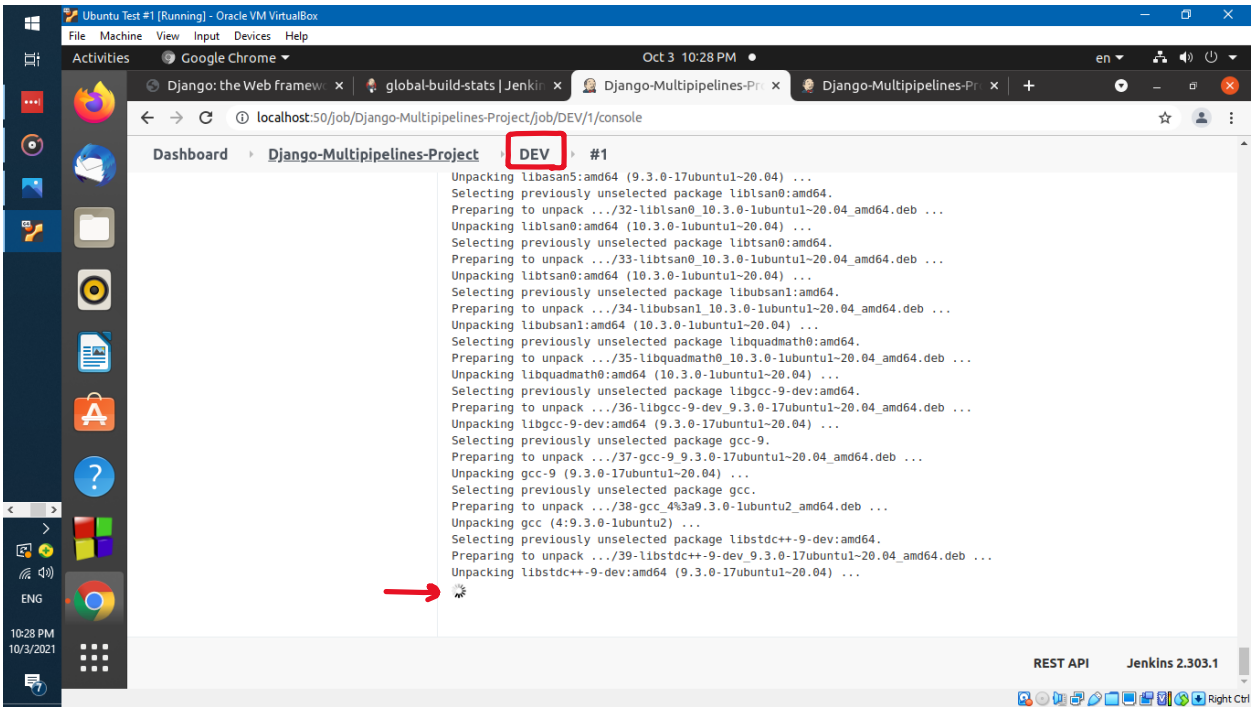
In {Console output Menu}, Jenkins Server is deploying DEV branch's pipeline first on "Project" node



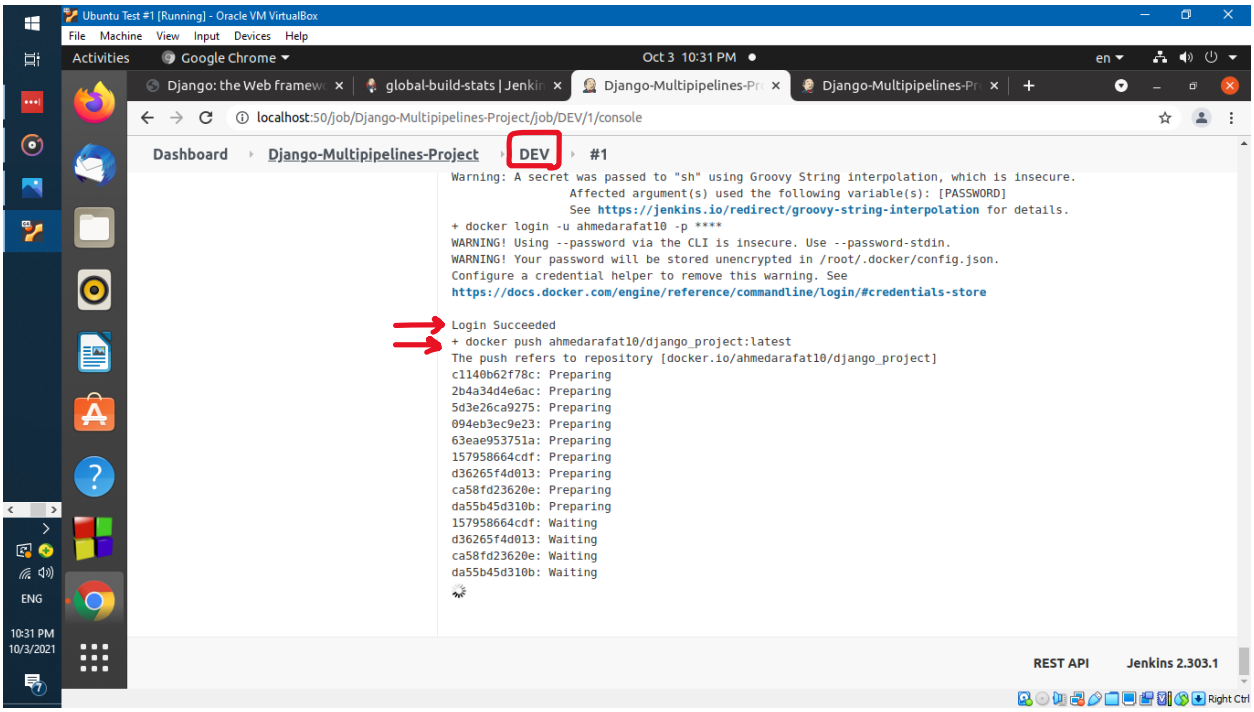
While DEV branch's pipeline is in queue waiting for the other pipeline to finish deploying



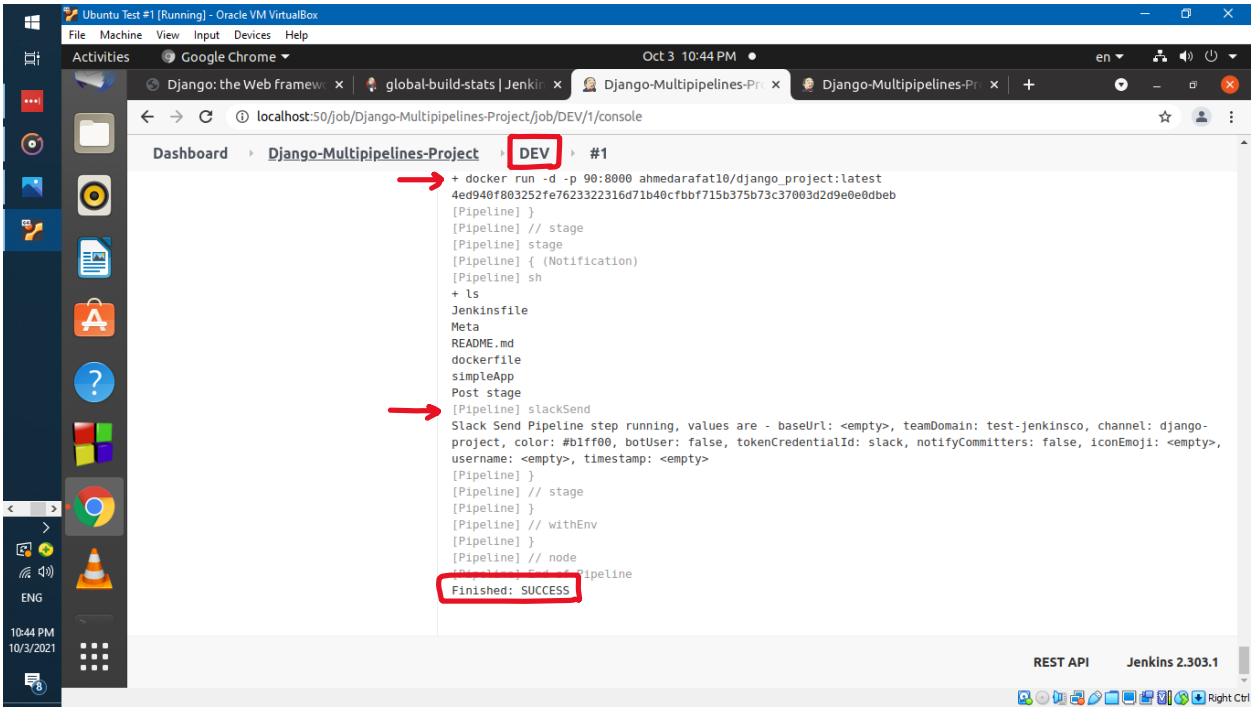
DEV branch's pipeline is still deploying {Executing Jenkinsfile's stages > steps}



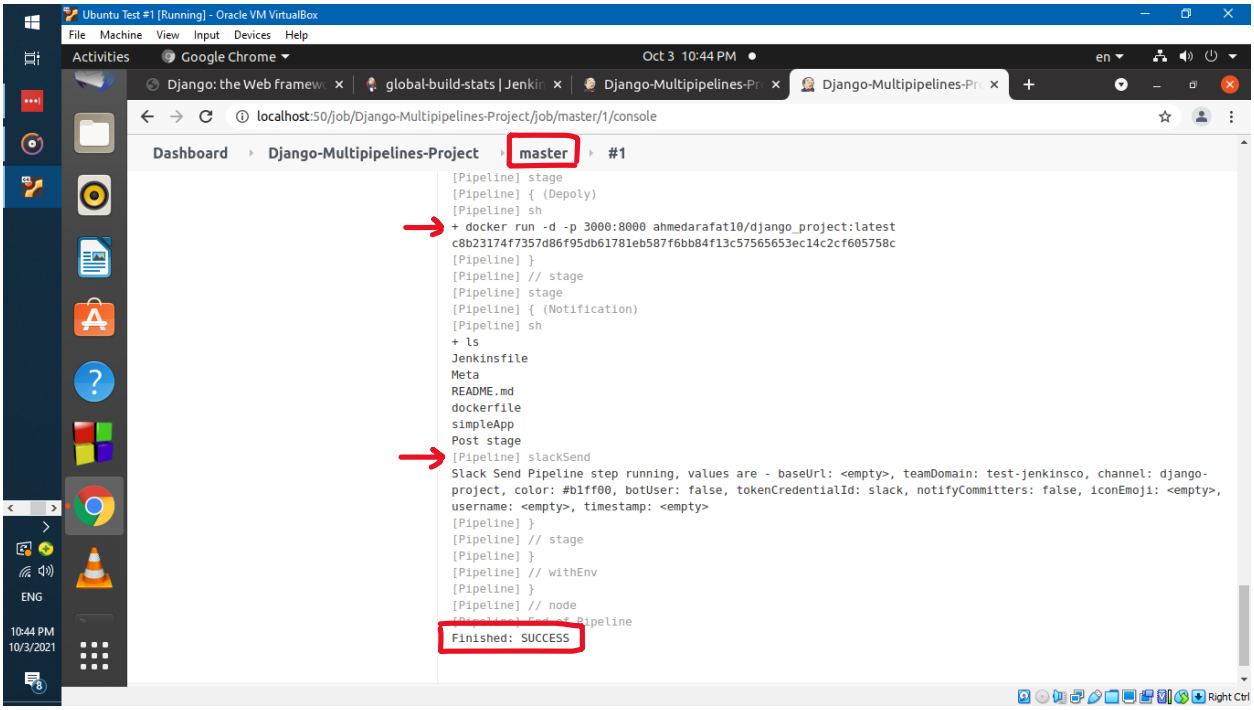
Now Jenkins Server is pushing created Django Image into my Docker Hub Account



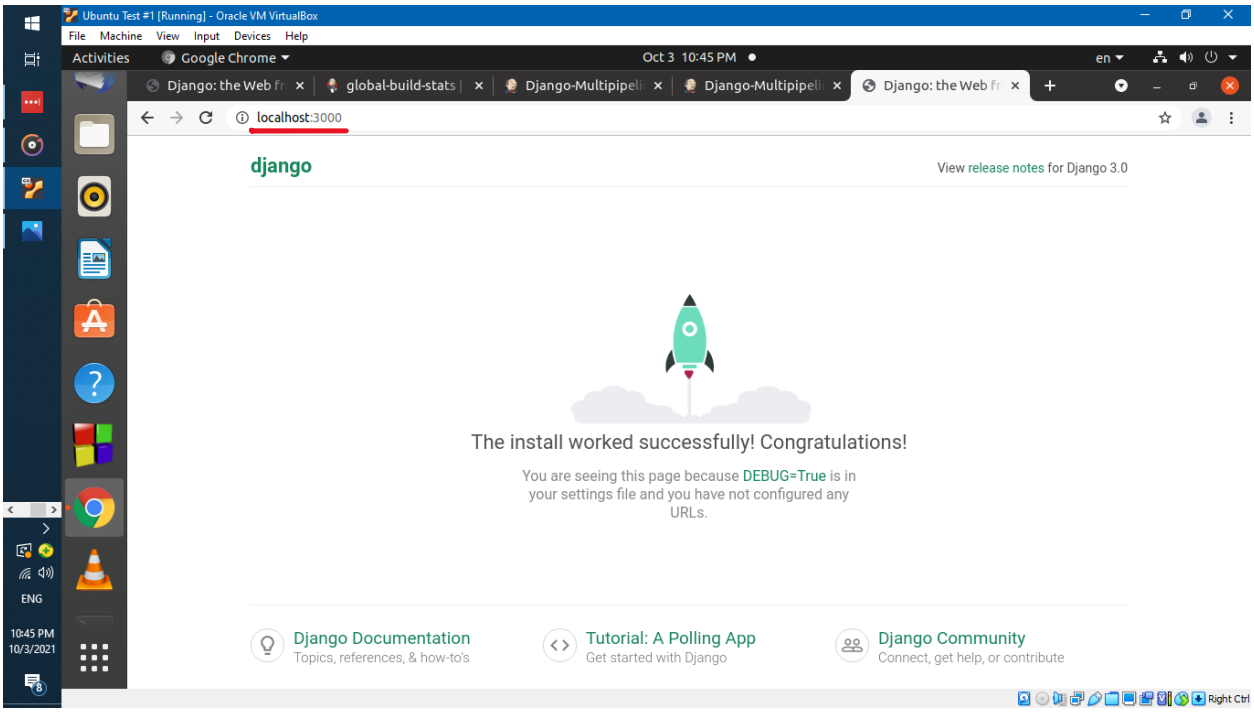
Now DEV branch's pipeline is deployed successfully 😊



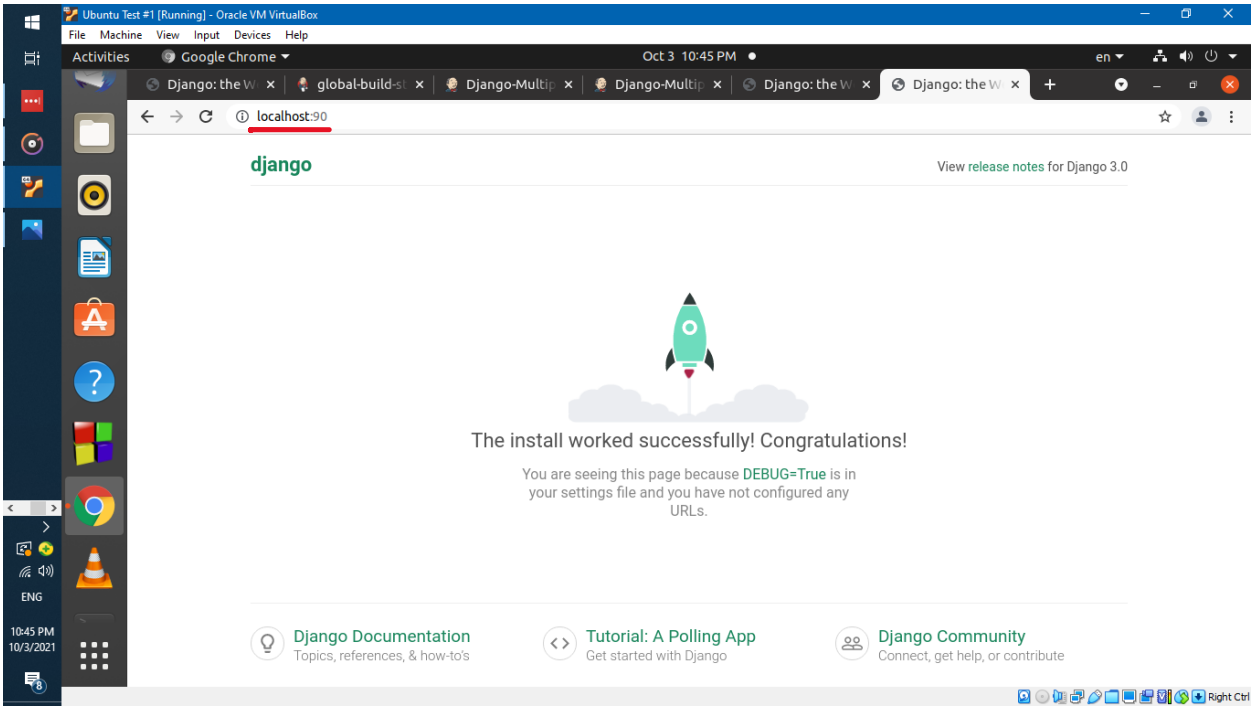
After finishing DEV branch's pipeline, Jenkins Server deploys Master branch's pipeline.
Now Master branch's pipeline is also deployed successfully 😊



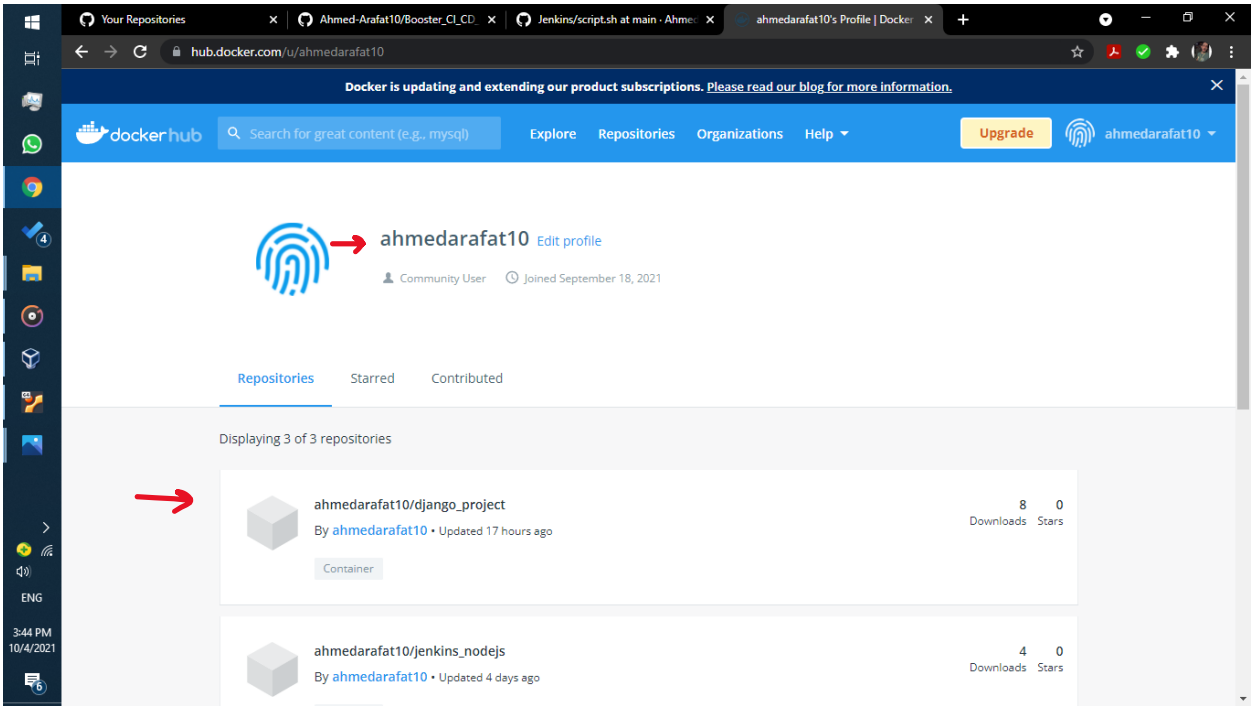
Checking {localhost:3000} -> For Master branch's pipeline
It Works 😊



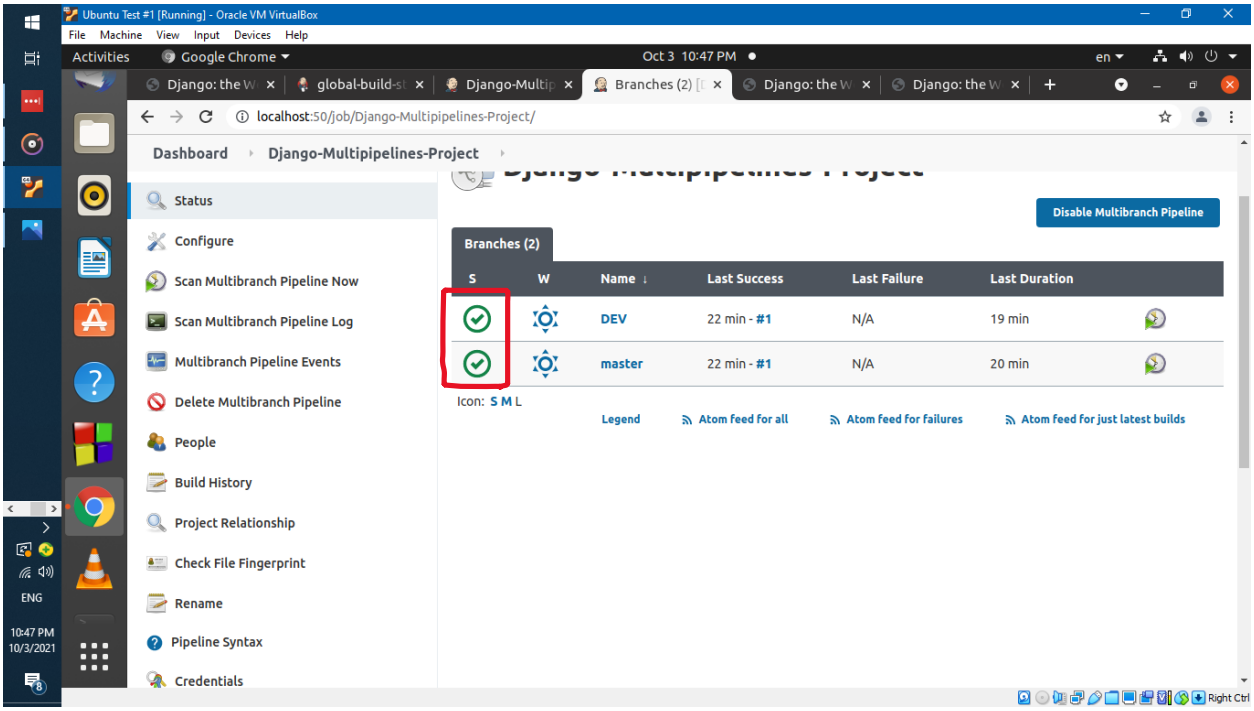
Checking {localhost:90} -> For DEV branch's pipeline
It Works 😊



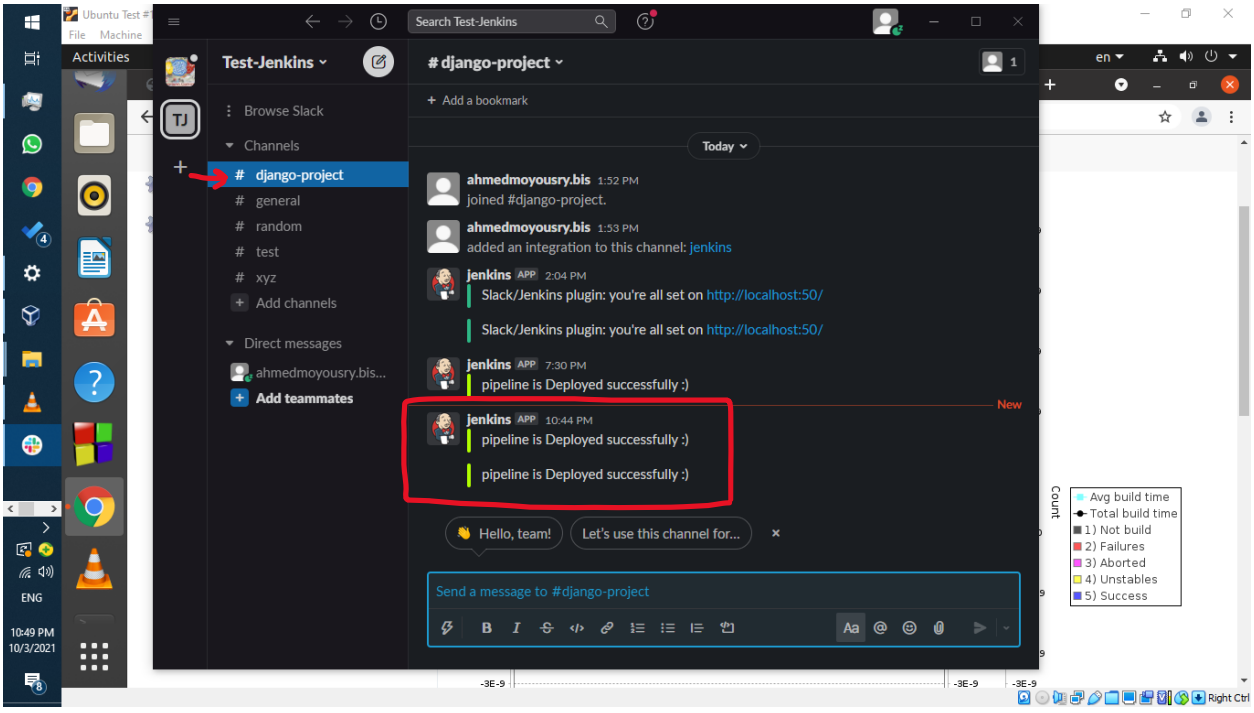
Created Django Image is pushed into my Docker Hub's Account



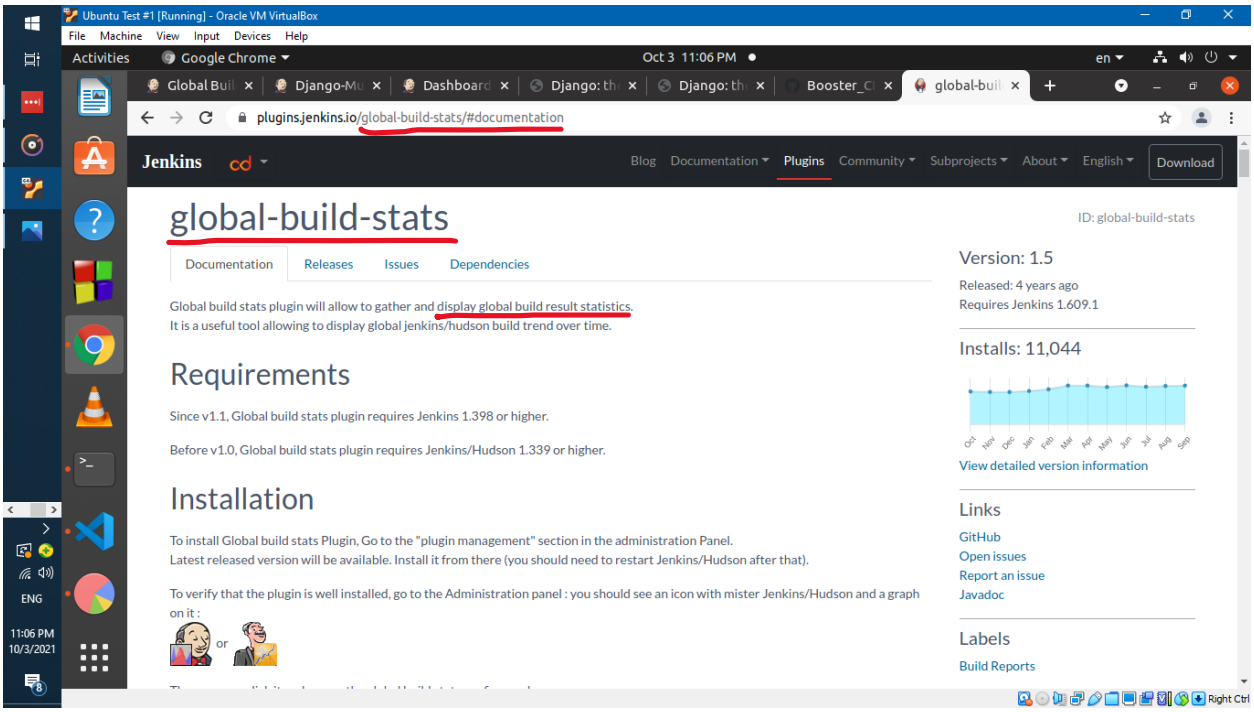
Both are working perfectly 😊



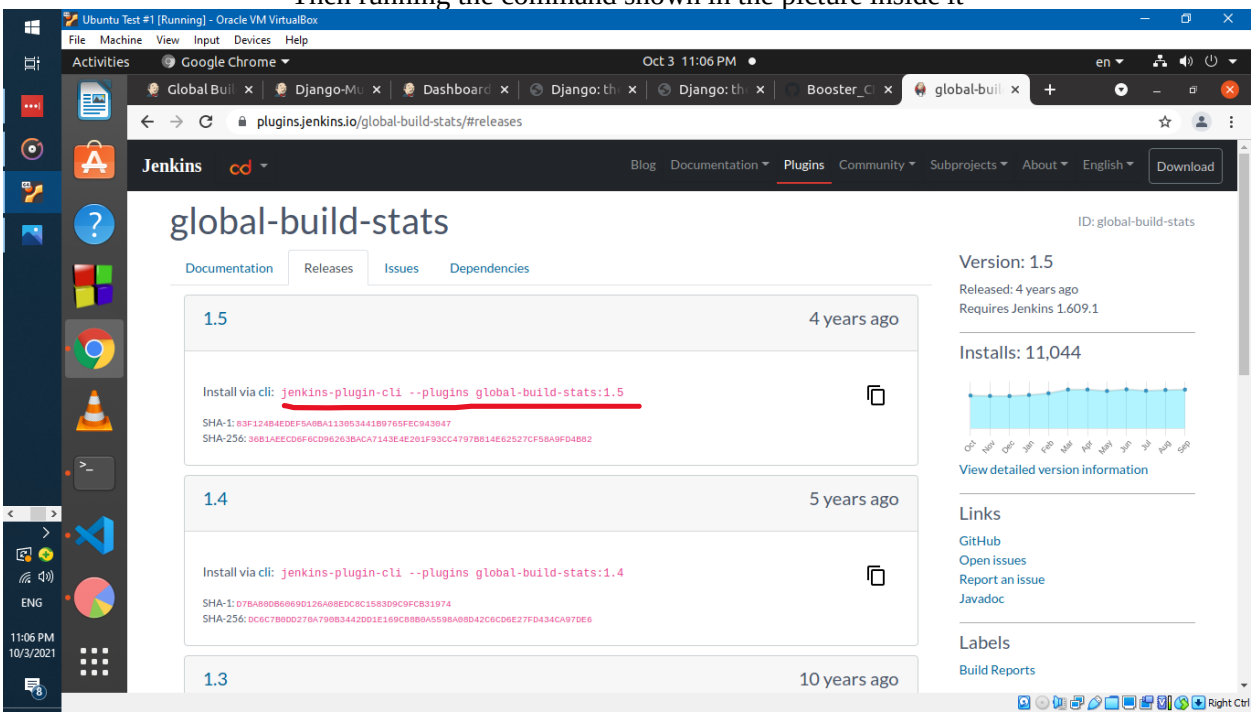
A message has been sent to my slack {Django-project} channel when both pipelines are successfully deployed



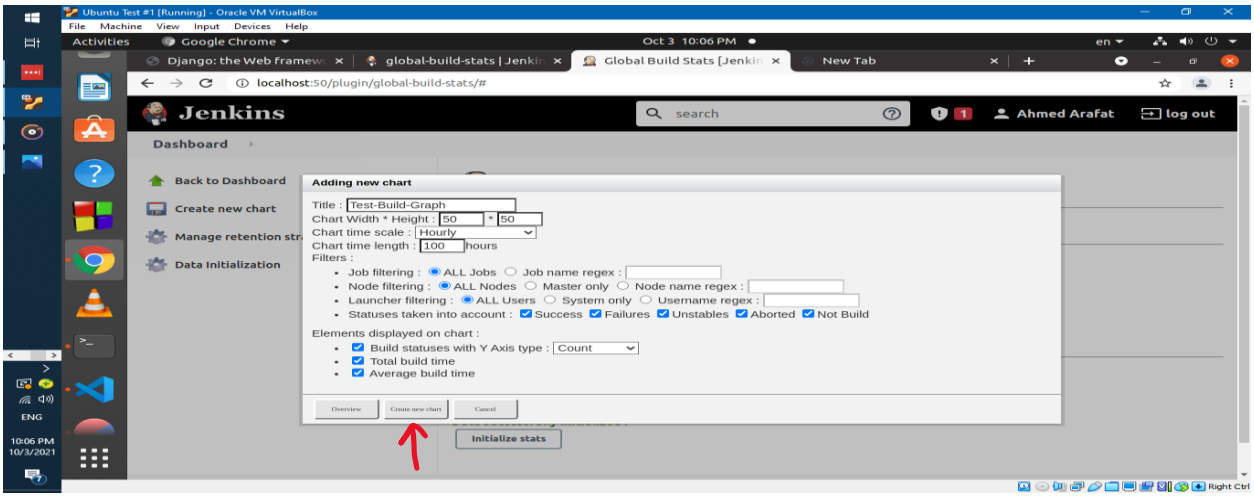
Now I have to install a Plugin called {global-build-stats} to show statistics about builds



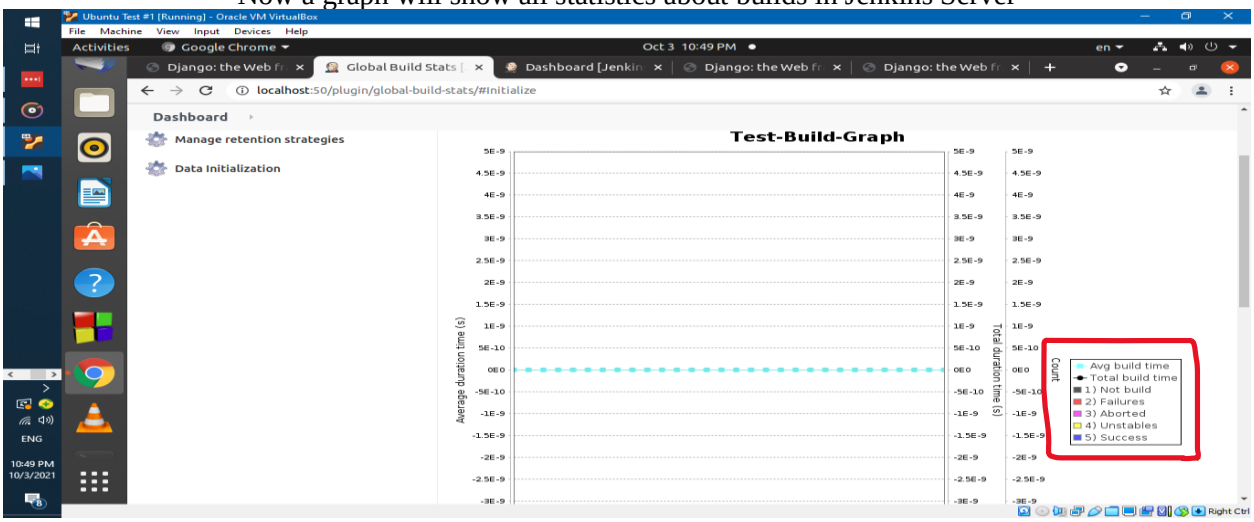
I will install it using terminal, but I will have first to enter Jenkins's docker container by running following command:
$ sudo docker exec -it <JenkinsContainer_ID> bash
Then running the command shown in the picture inside it



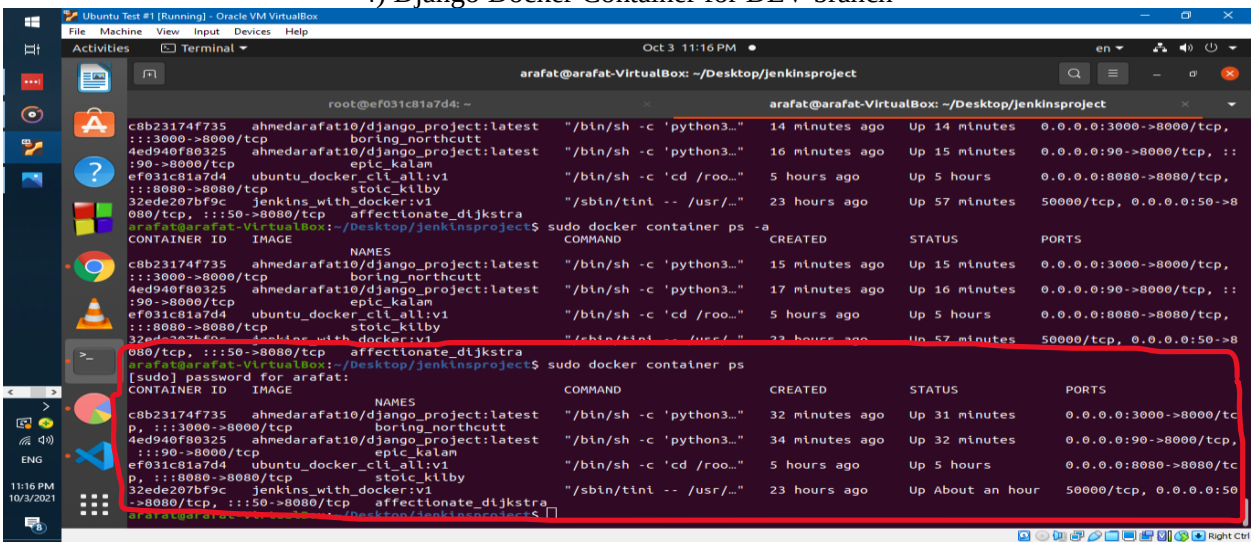
Configuring new installed plugin in Jenkins Server



Now a graph will show all statistics about builds in Jenkins Server



Now checking all running container in host terminal which are :
1) Jenkins server Docker Container
2) Ubuntu Docker Container
3) Django Docker Container for Master branch
4) Django Docker Container for DEV branch



That’s It 😊