Lab Documentation

How to use Jenkins to Automate Docker build and push to Docker hub.

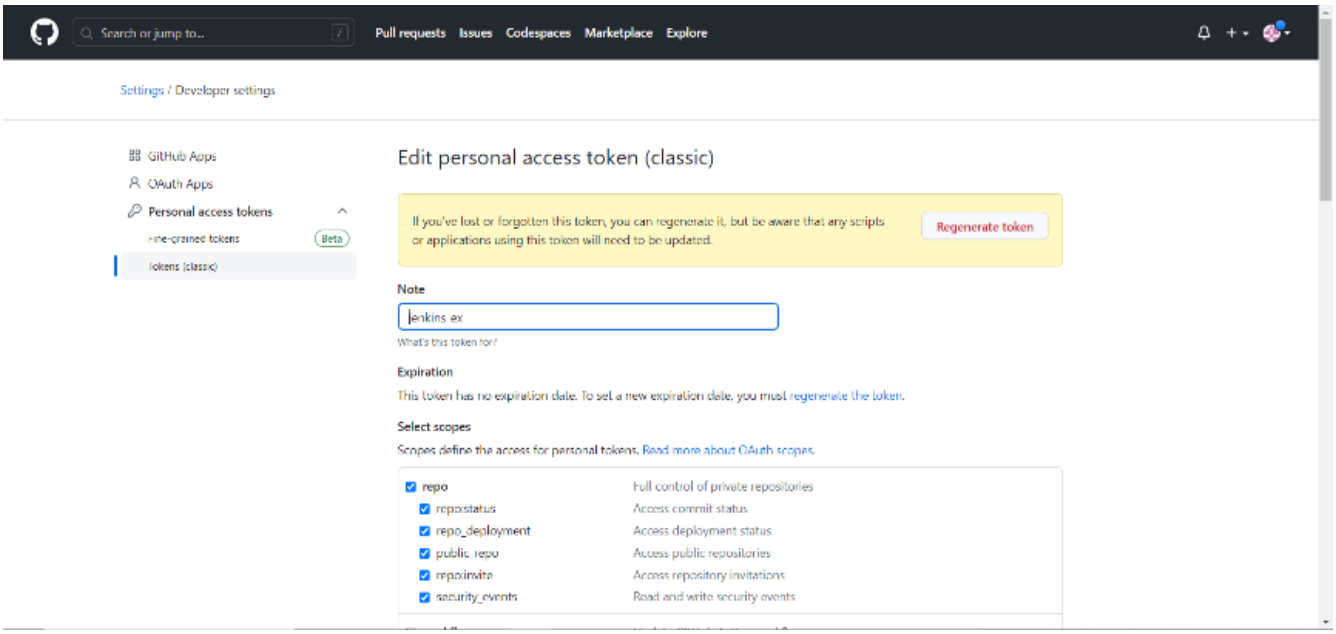
Note: download and run Jenkins using Docker image: “jenkins/jenkins:lts”

Step 1: clone github repository and push it to my github account as private repository

```
git clone https://github.com/Ahmad-Aladdin/jenkins_nodejs_example
cd .\jenkins_nodejs_example\ (the project folder)
git init
git add .
git remote add origin https://github.com/AlShaymaaHamdan/jenkins_nodeapp.git
git commit  -m “first commit”
git push –u origin master
```

Step 2: Create Personal access token in github

Open https://github.com/ and sign in to your account.
Go to “Settings > Developer Settings > Generate new Token”



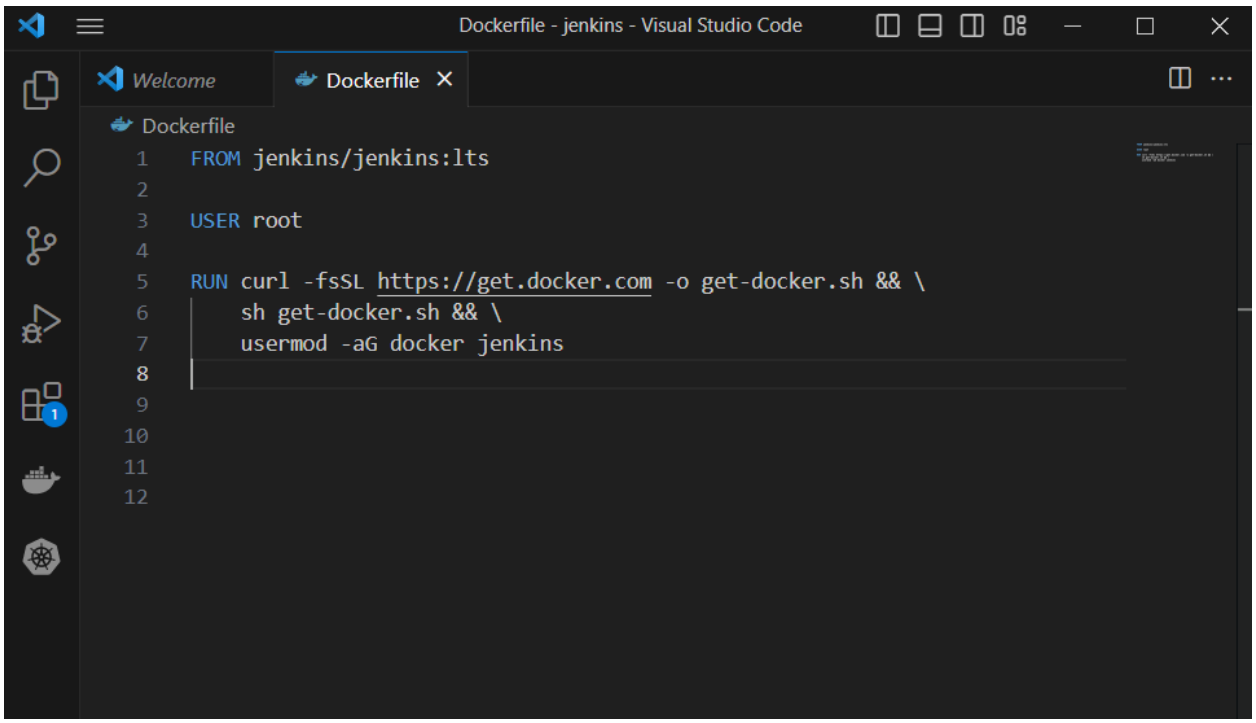
Step 3: Pull Jenkins image

After starting “Docker Desktop”, run this command to pull public Jenkins image from “DocekrHub”:

```
docker pull jenkins/Jenkins:lts
```

Step 4: Build Jenkins image from Dockerfile to run docker in Jenkins, so we can use docker commands in Jenkins project

- Open Dockerfile to view commands



- Use Dockerfile to build image named Jenkins:lts, using command:

```
docker build -t jenkins:lts .
```

- Run container from jenkins:lts image that I built before.
- Map the /var/jenkins_home directory inside the container to the local directory /Desktop/Training/jenkins/jenkins-home
- Use volume mount to let jenkins run the docker client
  Using command:

```
docker run -d -p 8080:8080 -v /Desktop/Training/jenkins/jenkins-home:/var/jenkins-home -v /var/run/docker.sock:/var/run/docker.sock jenkins:lts
```

- Start Jenkins:

Go to “localhost:8080”, jenkins will open and ask for admin password

Get it by running this cmd:

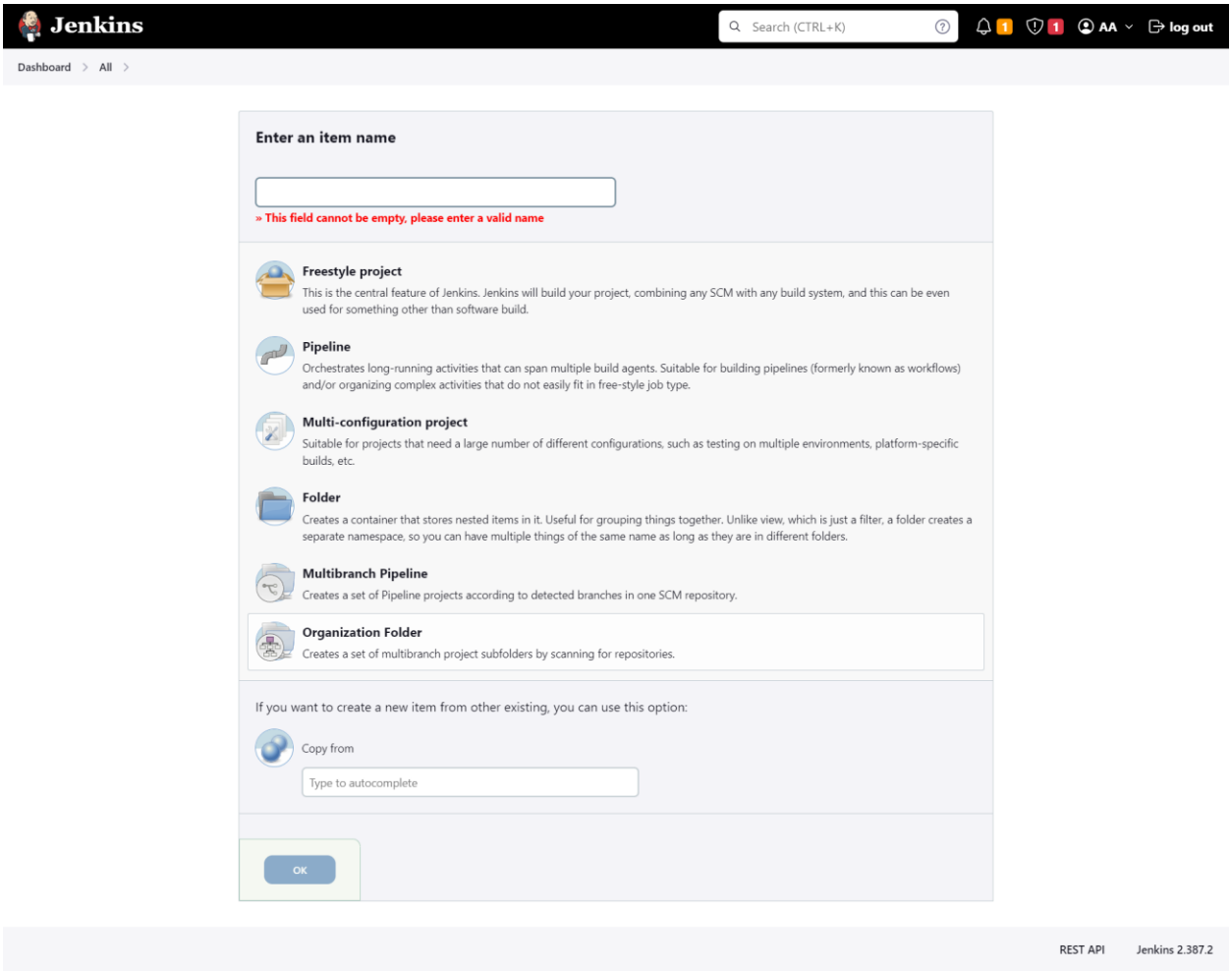
```
docker exec -it ab51  cat /var/jenkins_home/secrets/initialAdminPassword
```

Note: ab51 is container ID (Jenkins container)

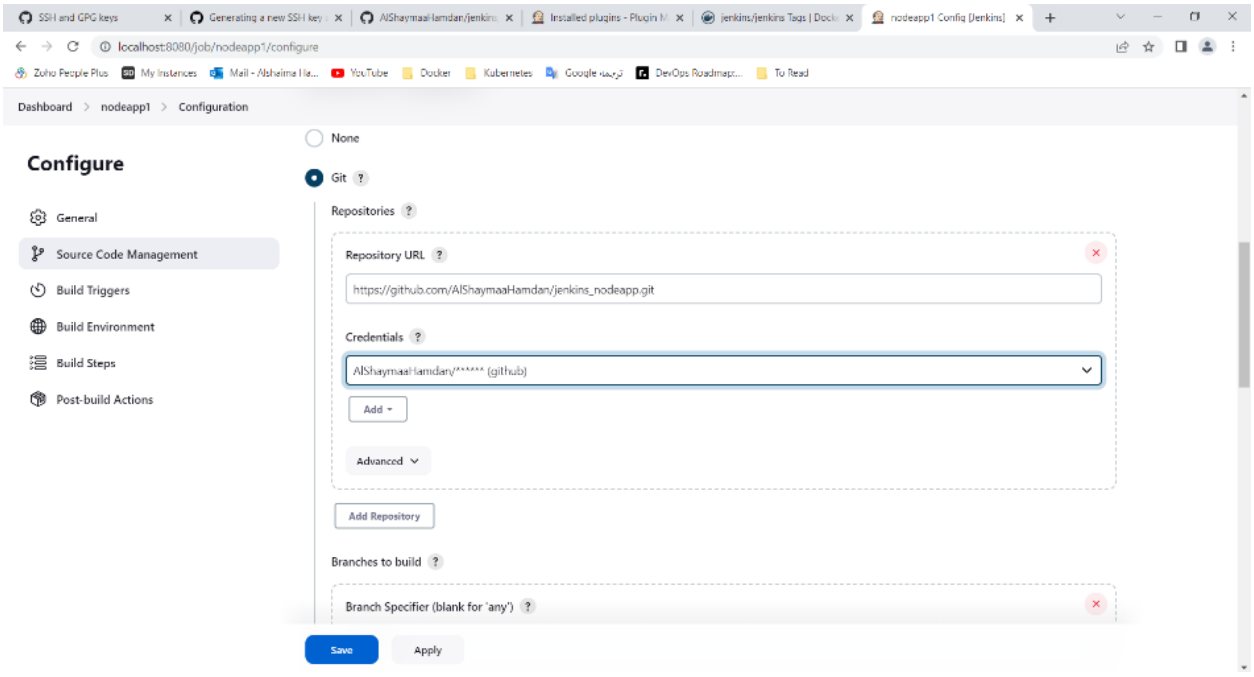
Setup “Jenkins” and create admin user

Step 5: Create a freestyle job on Jenkins

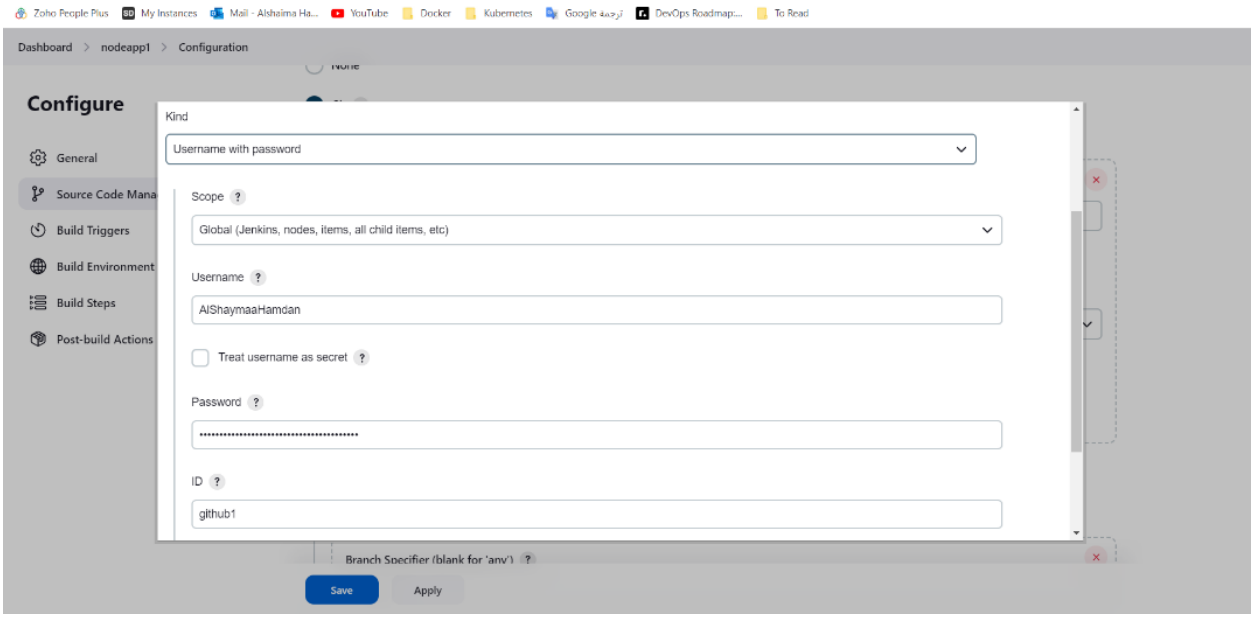
- Click “+ New Item” and choose “Freestyle Project”



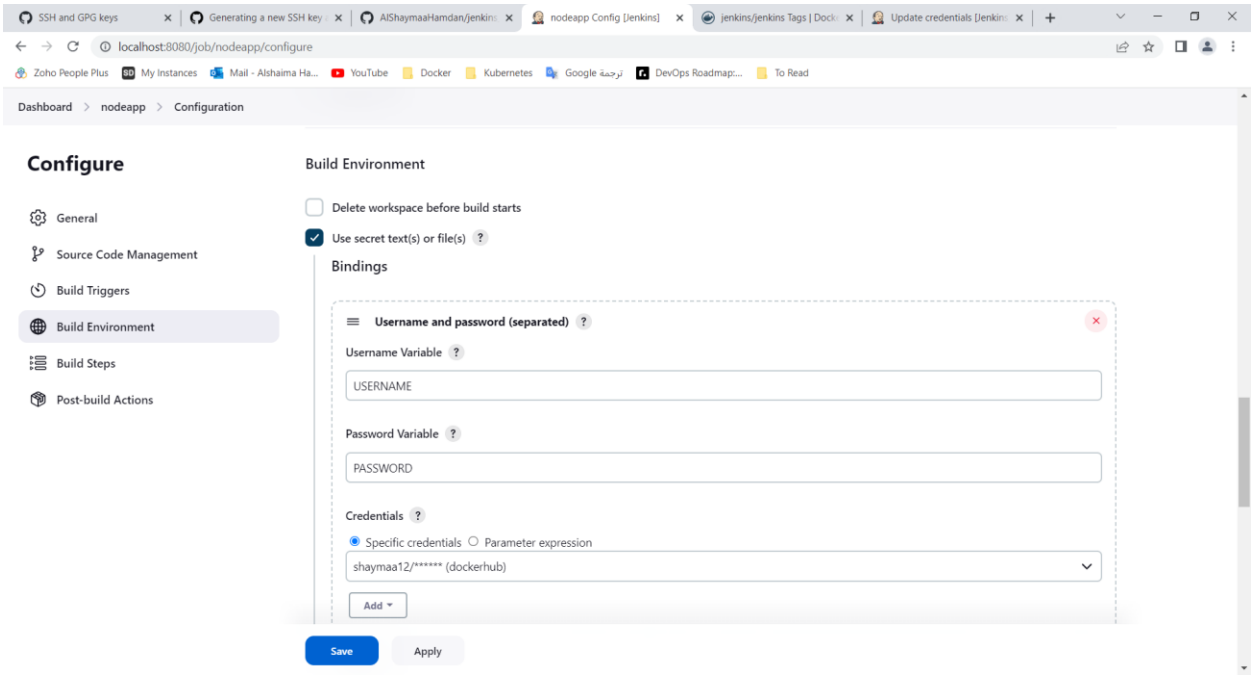
- Add your “Github” private repository and your “Github” credentials.



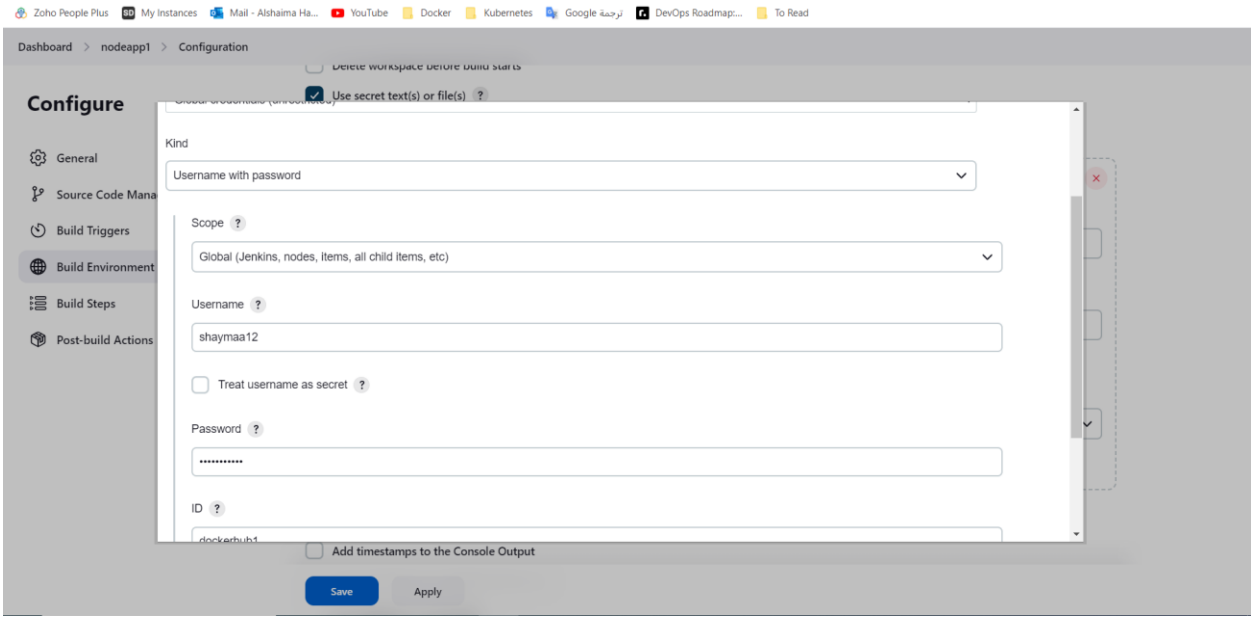
- To create “Github” credential click on Add and choose username and password (separated) then fill your username and password (your “Github access token” you created before)



- Then, add your “DockerHub” credentials. Choose “Use secret text(s) or file(s)”

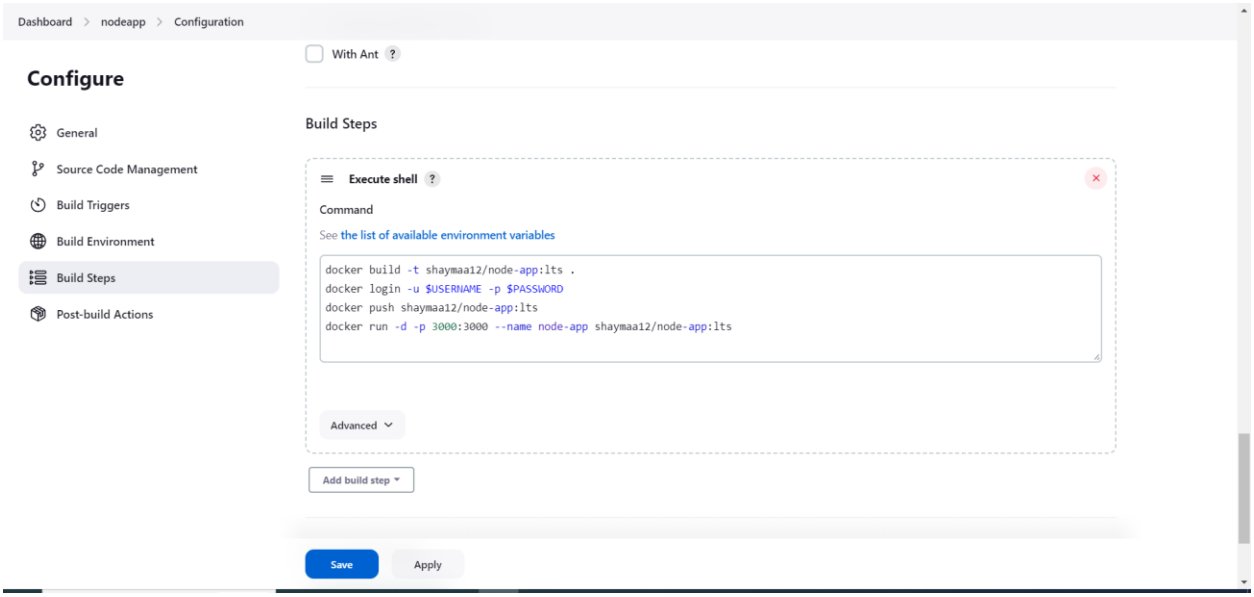


- To create “DockerHub” credentials you should choose “Username and password (separated)”



- Choose “Execute shell” to add build steps

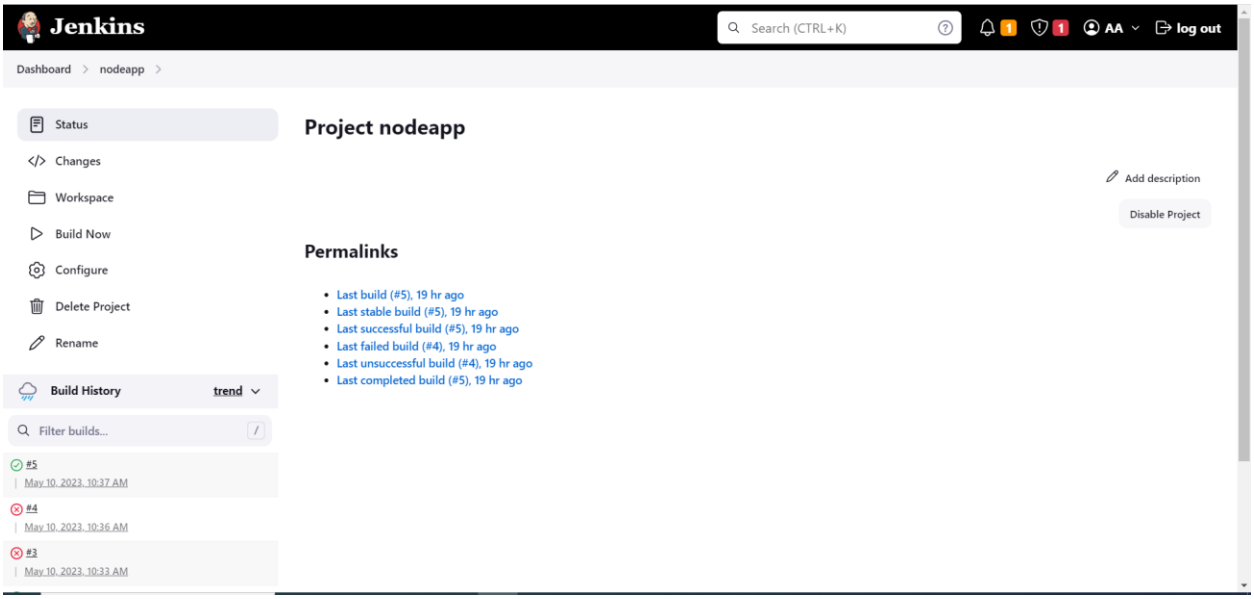
docker build -t shaymaa12/node-app:lts .	//use Dockerfile to build image named node-app and tag it image with your “DockerHub” account name
docker login -u $USERNAME -p $PASSWORD	//login to “DockerHub” with the “Secret” you created before, in order to push it there
docker push shaymaa12/node-app:lts	//push your image to your “DockerHub” account
docker run -d -p 3000:3000 --name node-app shaymaa12/node-app:lts	//create a container from your image and expose it to port 3000



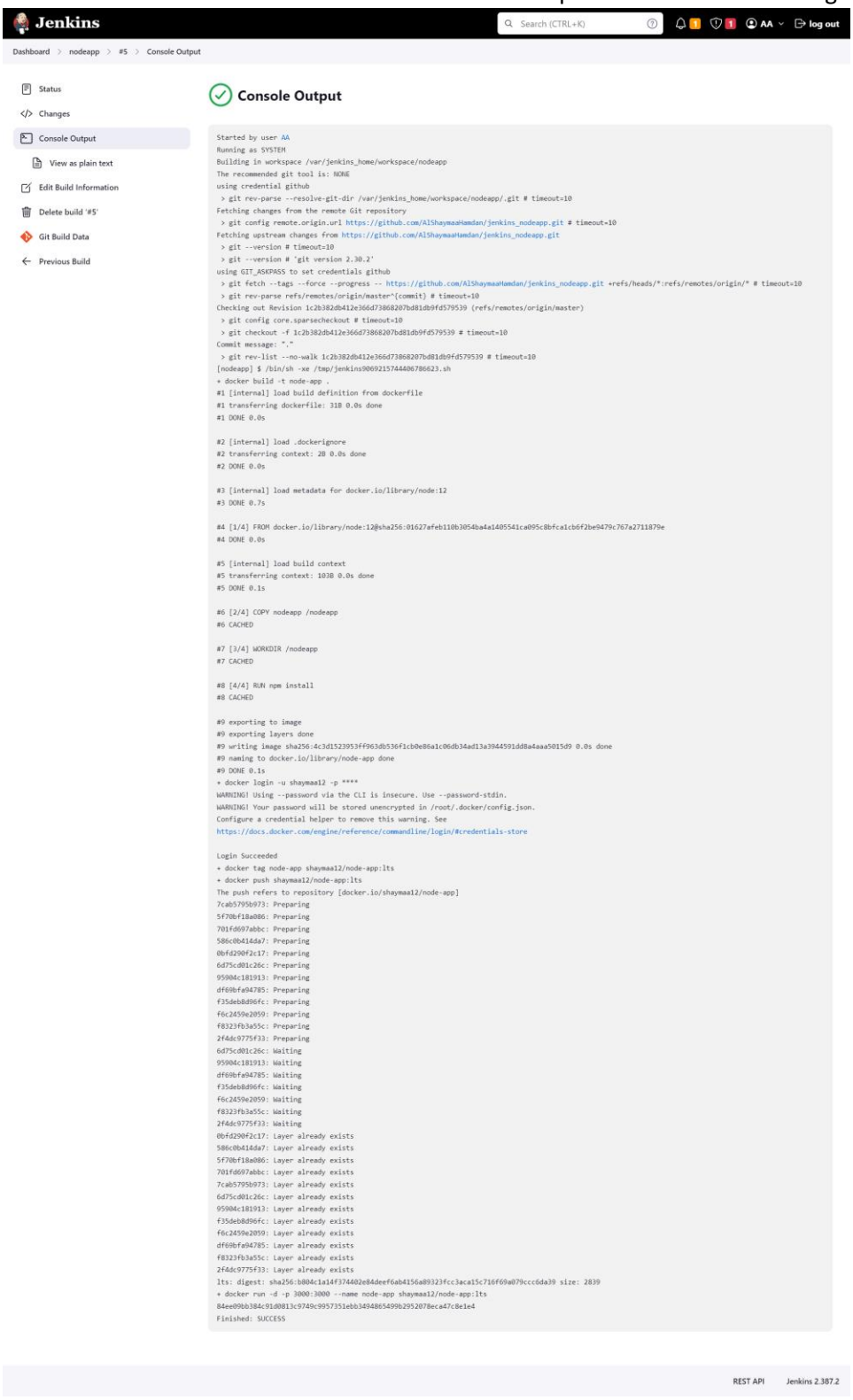
- Save your project

Step 6: Build “nodeapp” job

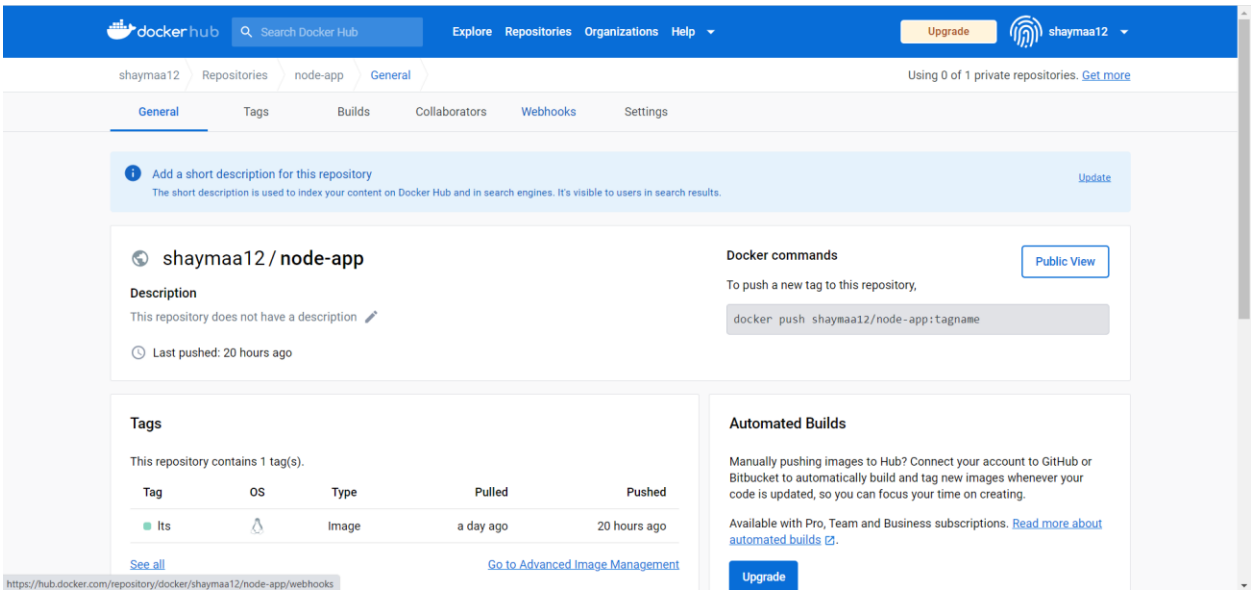
- Click on “Build Now”



- Choose the build number then Click on “Console Output” to view a detailed log of the build.



Step 7: Check your “DockerHub” account, a new repository (image) should be added



Step 8: Check if “nodeapp” application is running

- Go to browser and write “localhost:3000”