Creating CI CD Pipeline in Jenkins with Docker

In master ec2 instance

* Install Jenkins
* Install Maven
* Install Github
* Install Docker

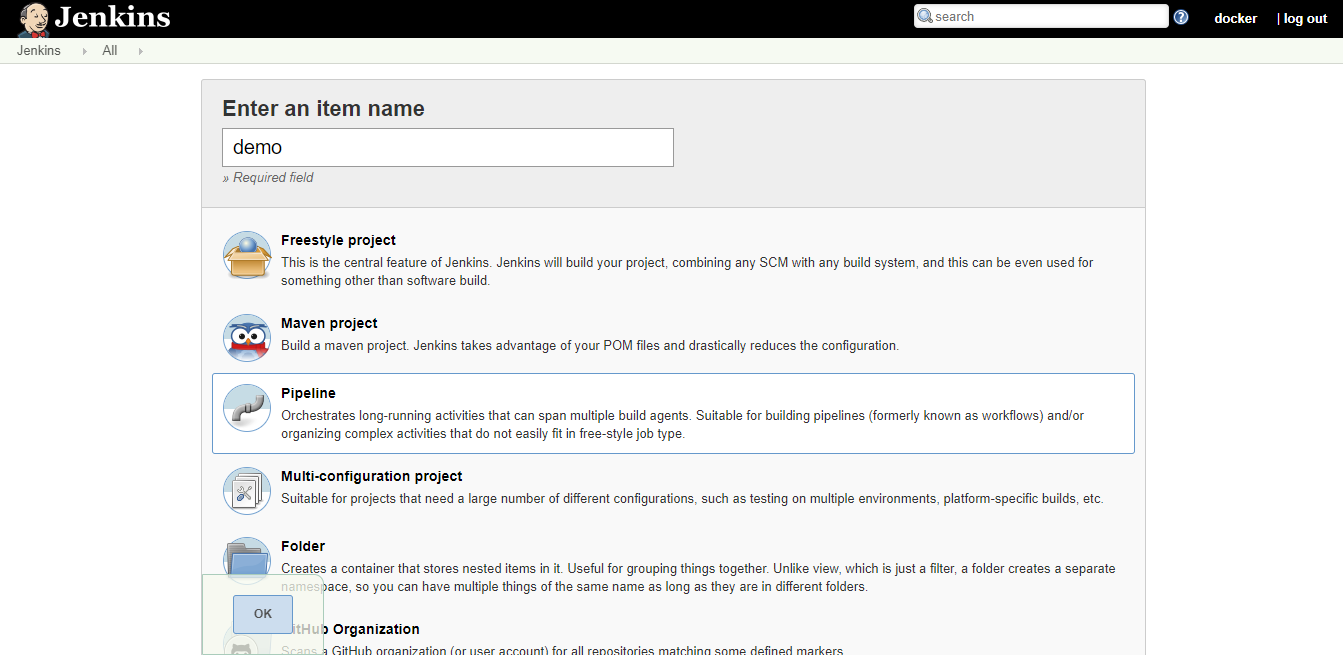
After logging in the Jenkins from master ec2

Install plugins of

* Docker
* Github
* Maven
* Sshagent

In Jenkins

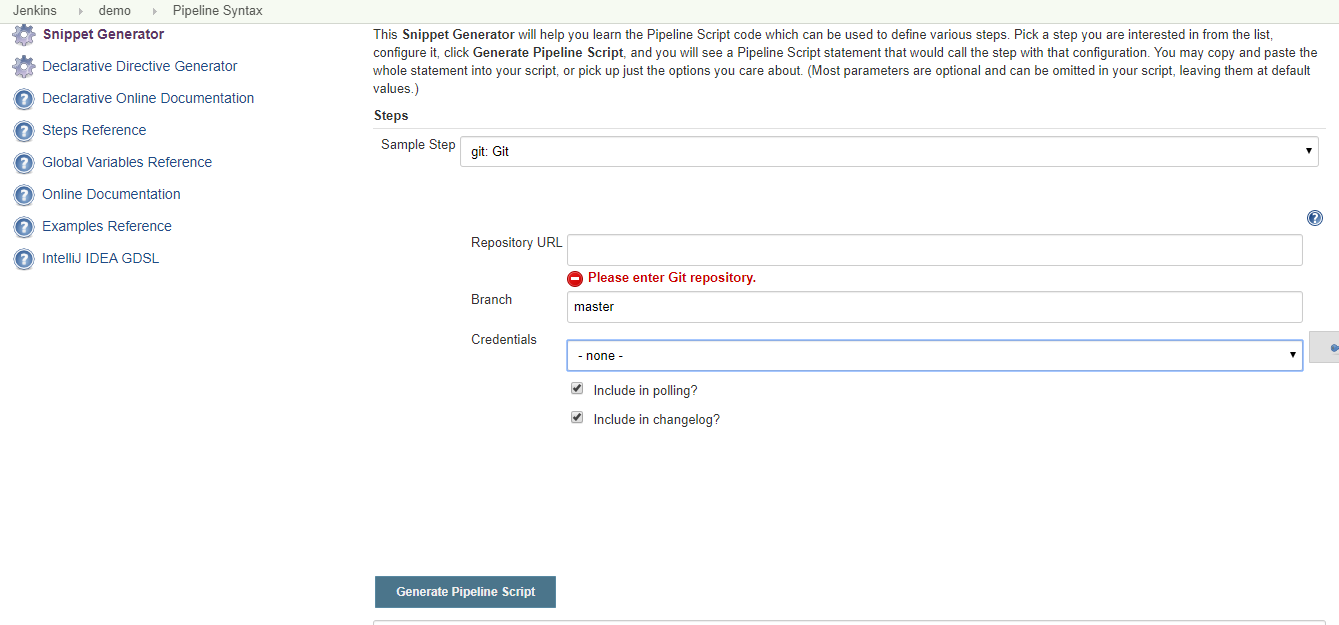
Step1: We need to create the Piple Line Project



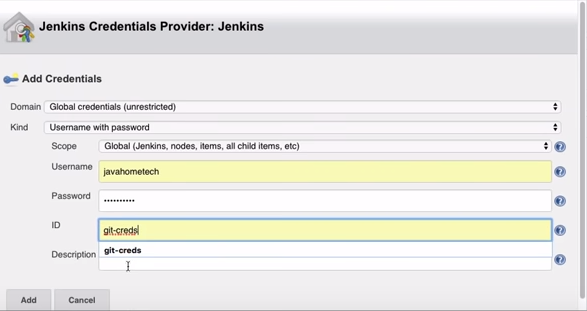
Step2: After creating the project

In pipeline we need to write the script for processing

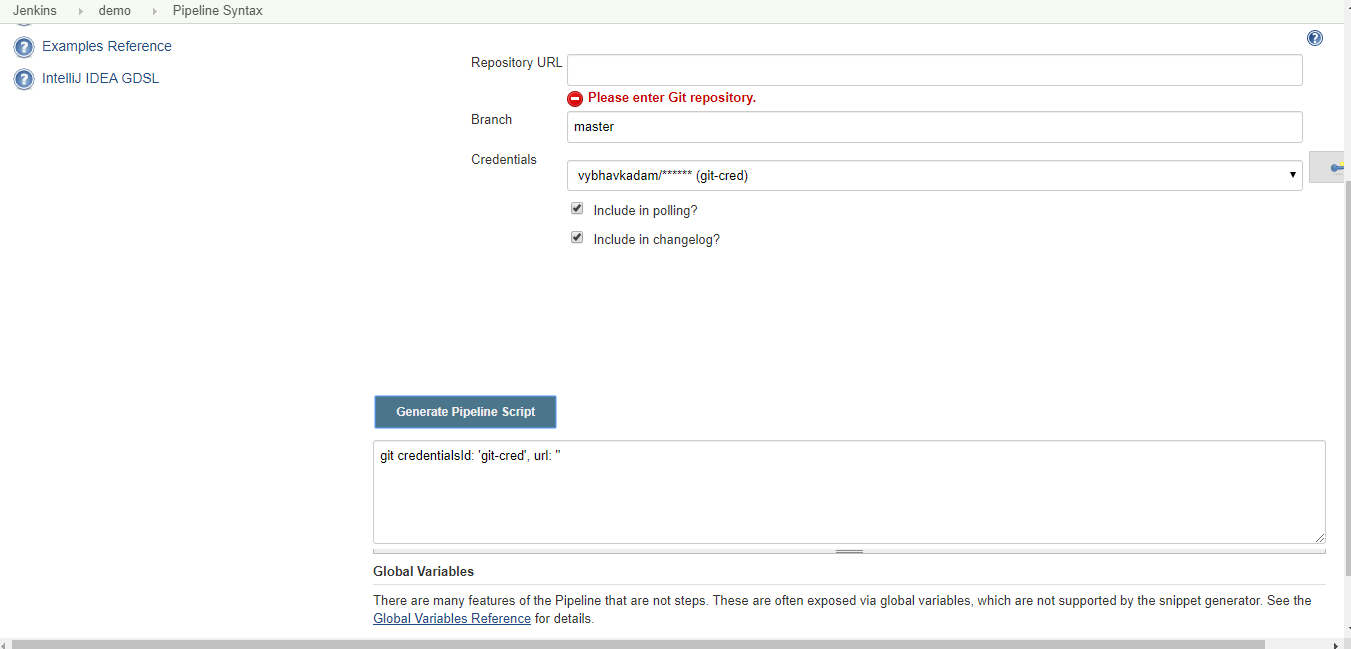
* Stage 1 : we clone the Github repository
* We need to bind the credentials of Github



Select the github and copy the url of the repository .

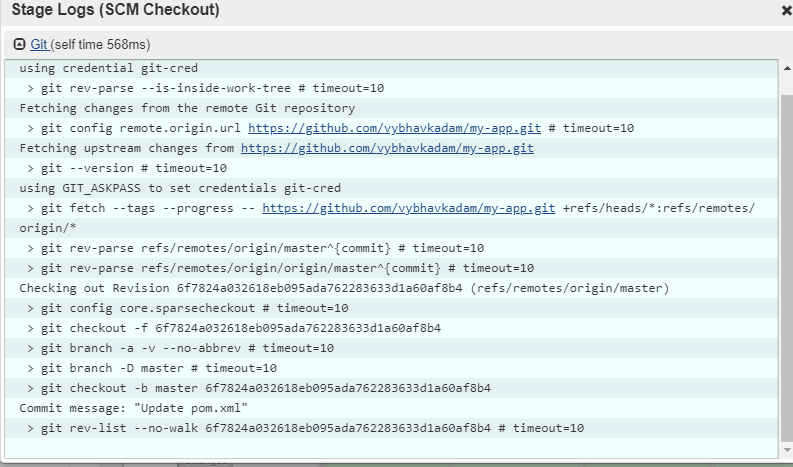


Binding the credentials of the git hub



Generating the pipeline script

After the building the project this is the out put



Script

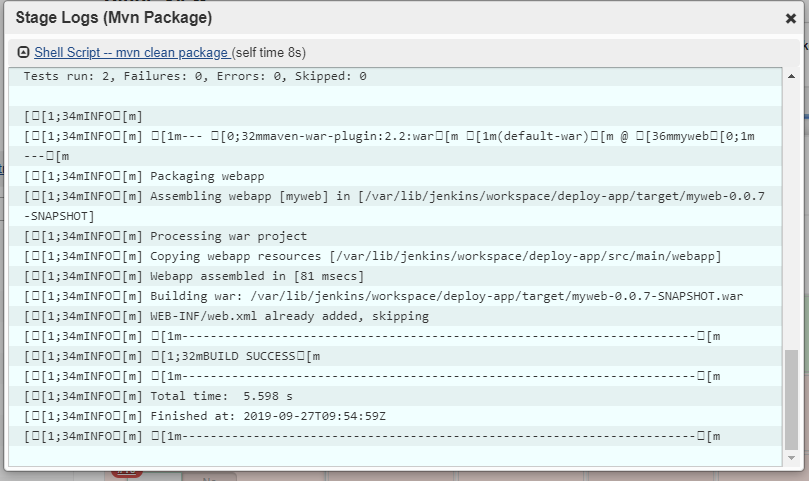
Node {

stage('SCM Checkout'){

git credentialsId: 'git-cred', url: 'https://github.com/vybhavkadam/my-app.git'

}

* Stage2: Maven Clean and install



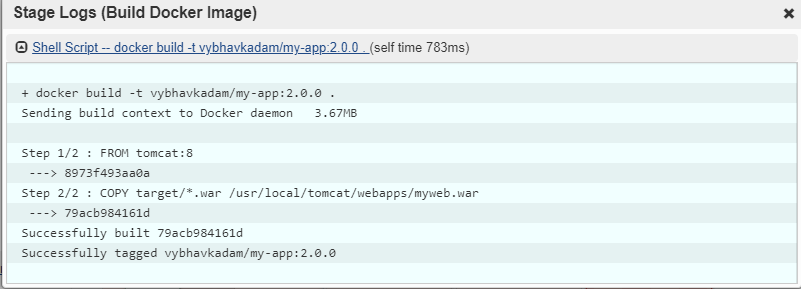
Script

stage('Mvn Package'){

sh 'mvn clean package'

}

* Stage3 Build Docker Image



Script

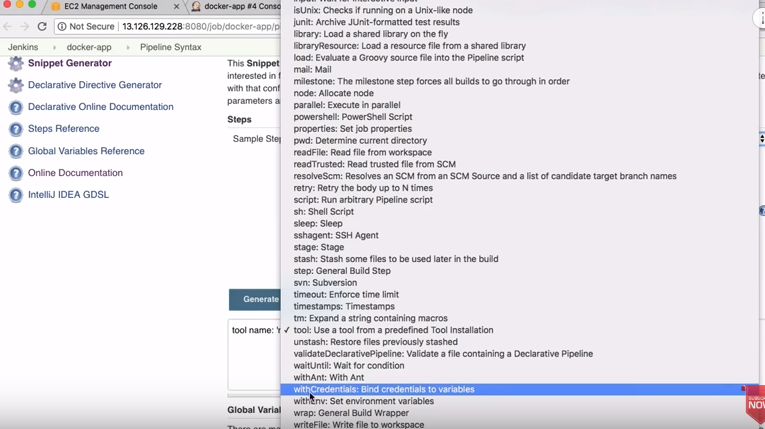
stage('Build Docker Image'){

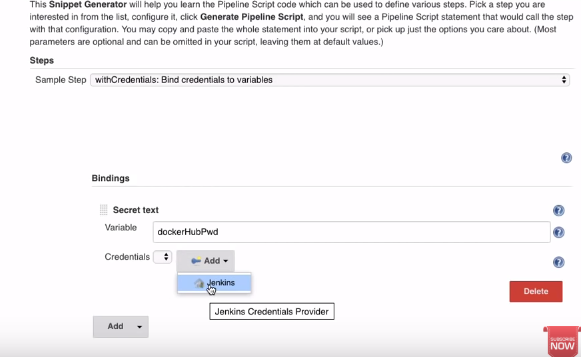
sh 'docker build -t vybhavkadam/my-app:2.0.0 .'

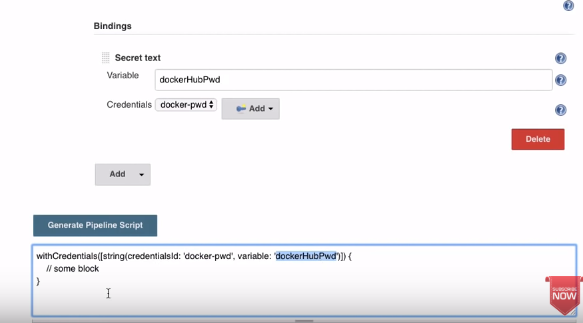
}

* Stage4 Push docker image to docker hub

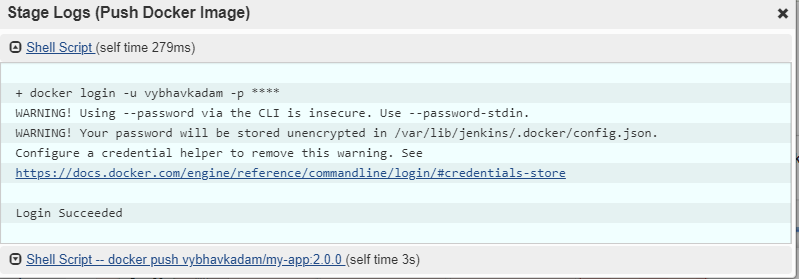
We need to bind the docker hub Credentials in the Jenkins



Binding the docker hub credentials 

Generating the pipeline script

This is the output of the successful push docker image



stage('Push Docker Image'){

withCredentials([string(credentialsId: 'docker-pwd', variable: 'dockerHubPwd')]) {

sh "docker login -u vybhavkadam -p ${dockerHubPwd}"

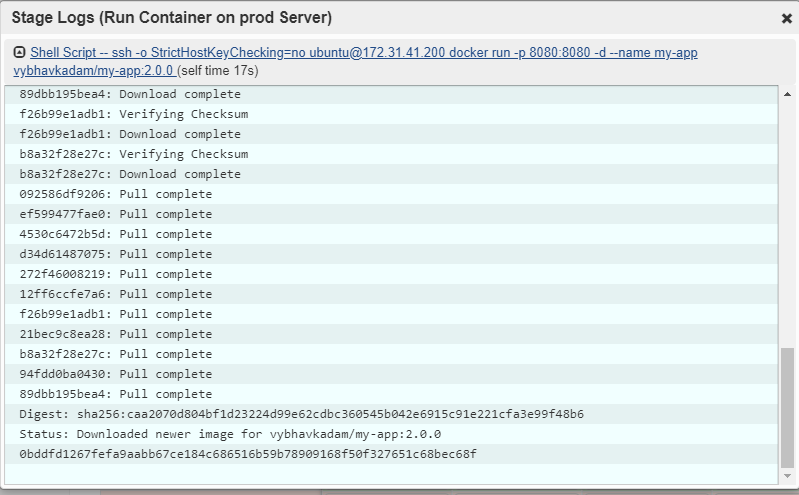
}

sh 'docker push vybhavkadam/my-app:2.0.0'

}

* Stage5 Runing the image in the other ec2 or Production server

We need to install the Docker in the production server



Reference Links

Installing All the tools

<https://linuxize.com/post/how-to-install-and-use-docker-on-ubuntu-18-04/>

Reference Video

<https://www.youtube.com/watch?v=gdbA3vR2eDs&t=1798s>