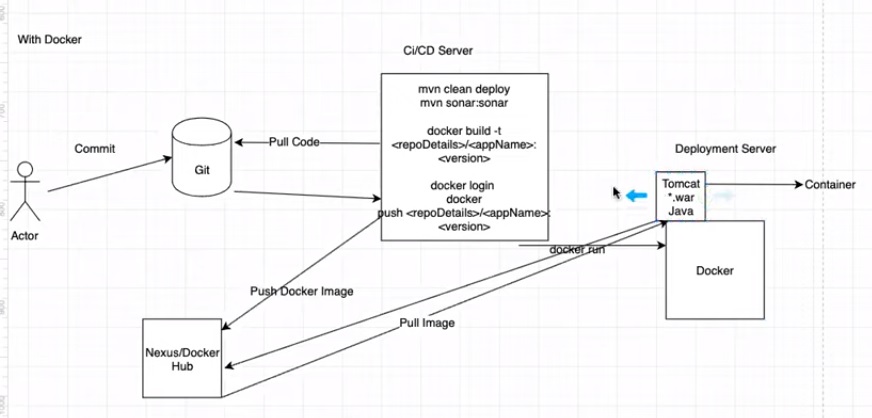
**Jenkins docker CICD project**

**reference:** [**https://github.com/sandeepdurai/java-web-app-docker.git**](https://github.com/sandeepdurai/java-web-app-docker.git)

**youtube:** [**https://youtu.be/hHZoV3LBIwE**](https://youtu.be/hHZoV3LBIwE)

****

**step 1**

create two EC2 instance[ ubuntu]---> 1.jenkins java and dockers

2. only docker

==========================================================================

**install java and jenkins, docker: EC2-1**

**\*** sudo apt update

\* sudo apt install openjdk-8-jdk

\* sudo wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -

\* sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

\* sudo apt update

\* sudo apt install jenkins

\* sudo systemctl status jenkins

\* sudo apt install docker.io -y

**# add jenkins user to docker**

\* sudo usermod -aG docker jenkins

\*sudo systemctl restart Jenkins

**install docker in ec2-2:**

\* sudo apt update

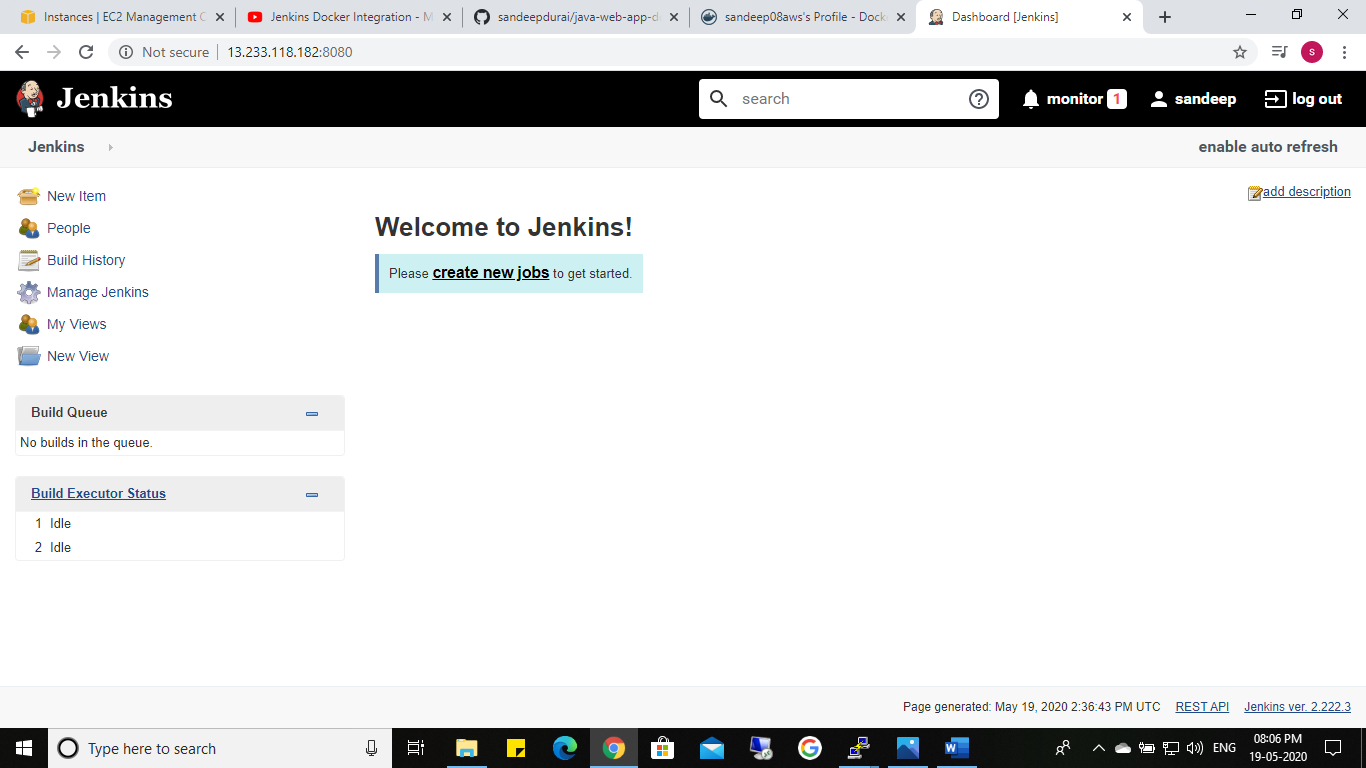
\* sudo apt install docker.io

\*docker info

\*sudo docker info

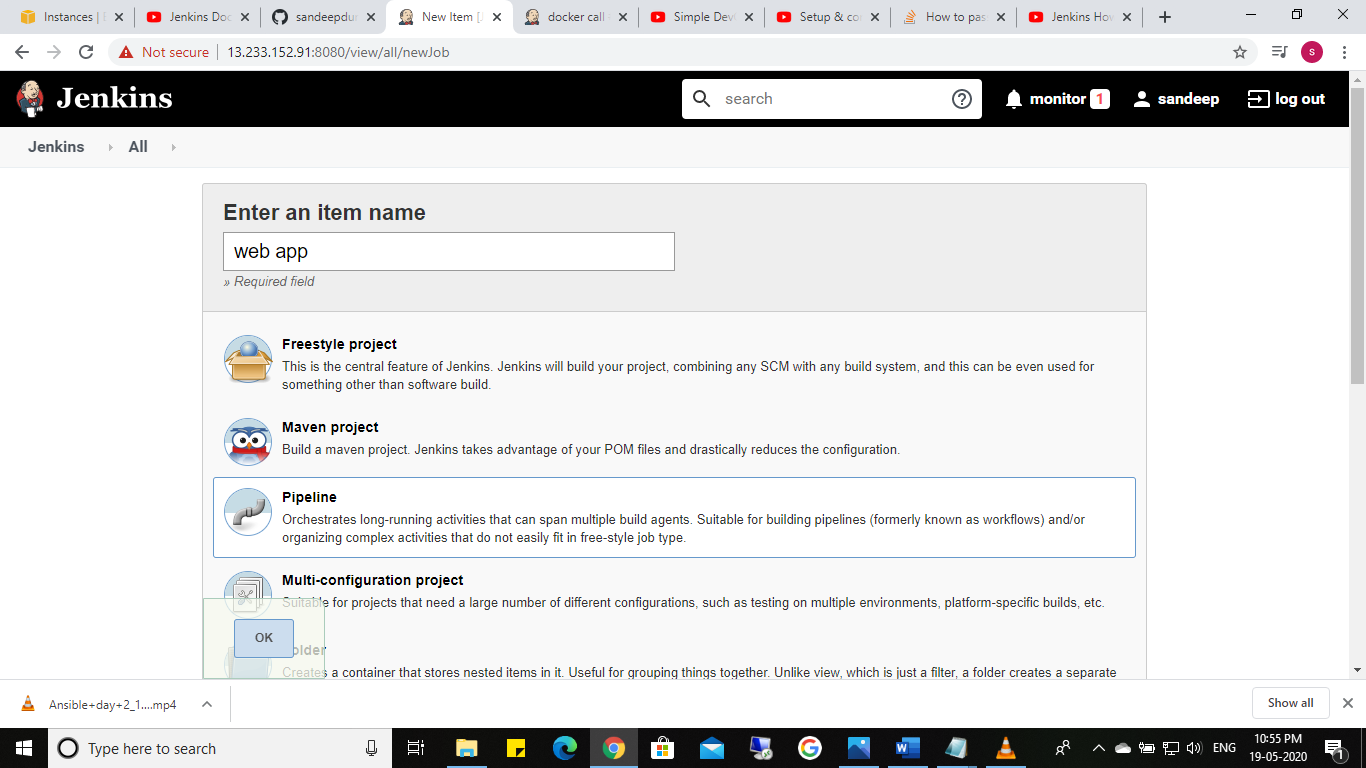
\* sudo usermod -aG docker ubuntu

**Configer Jenkins**



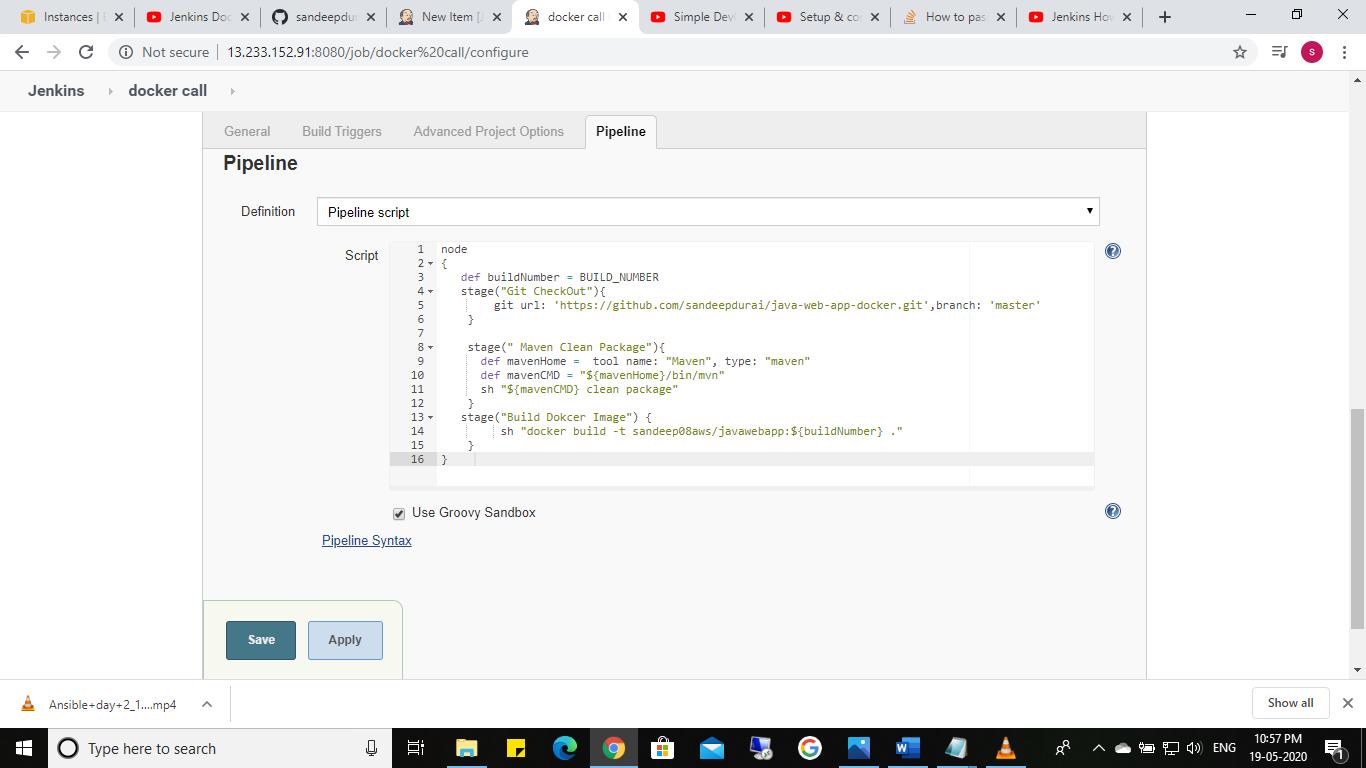
**Create job in jenkin:**

**Using pipeline**

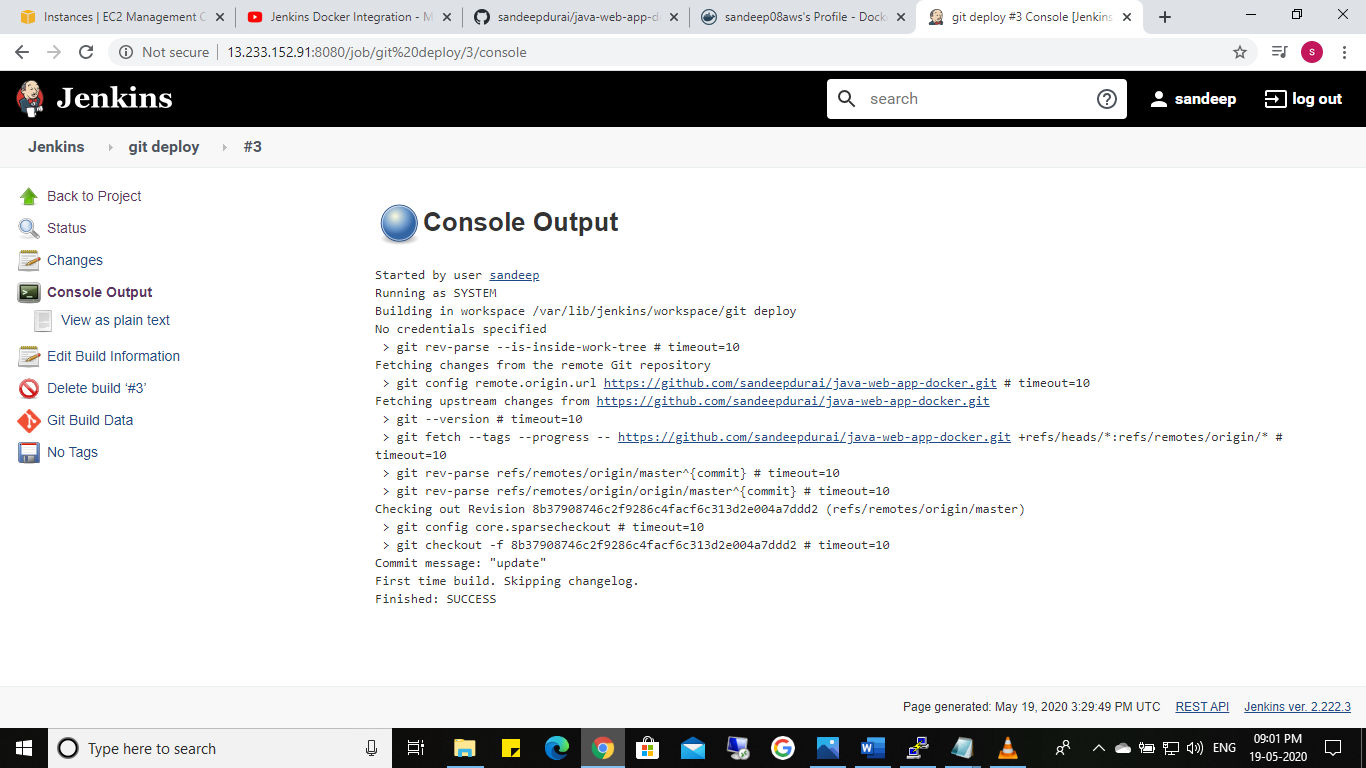


**Stage 1**

**Check the git repository**



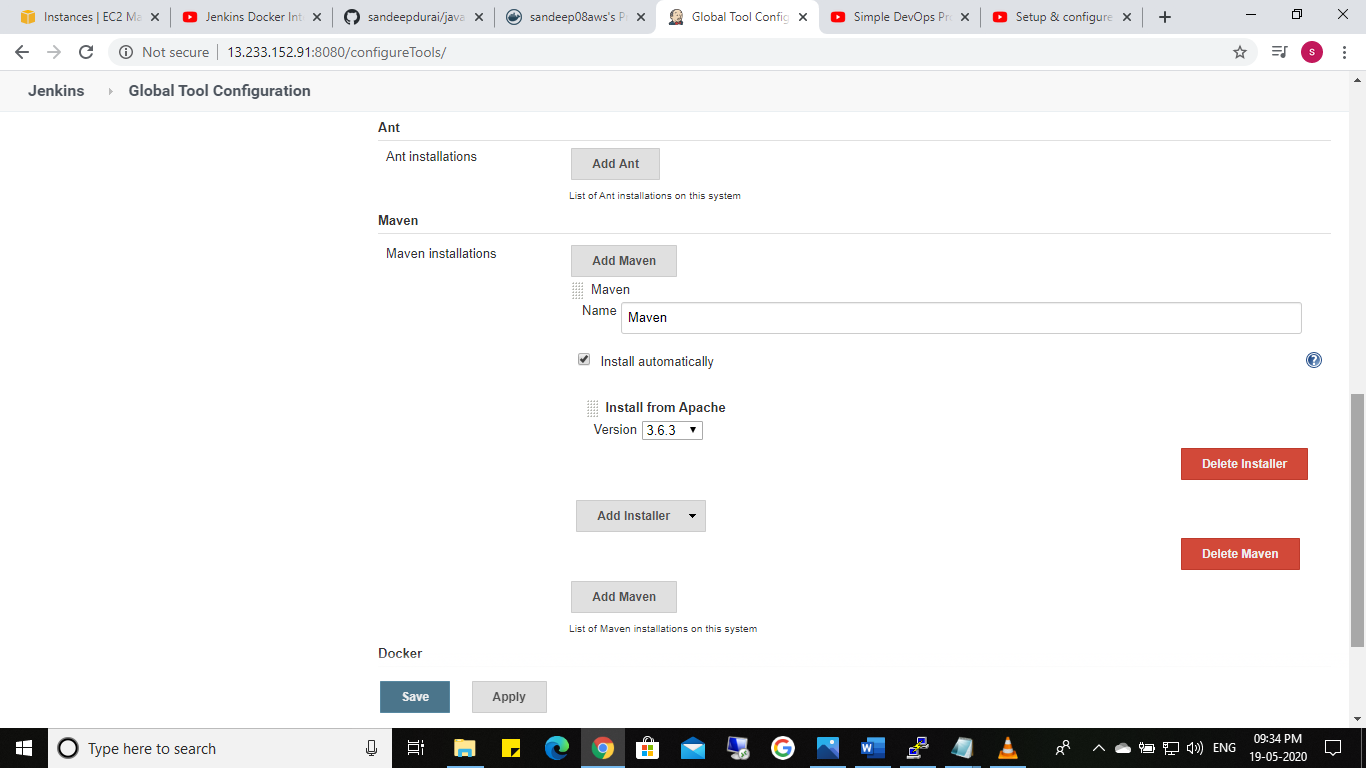
**Built – console output—success**



**Install maven**

**Stage 2**

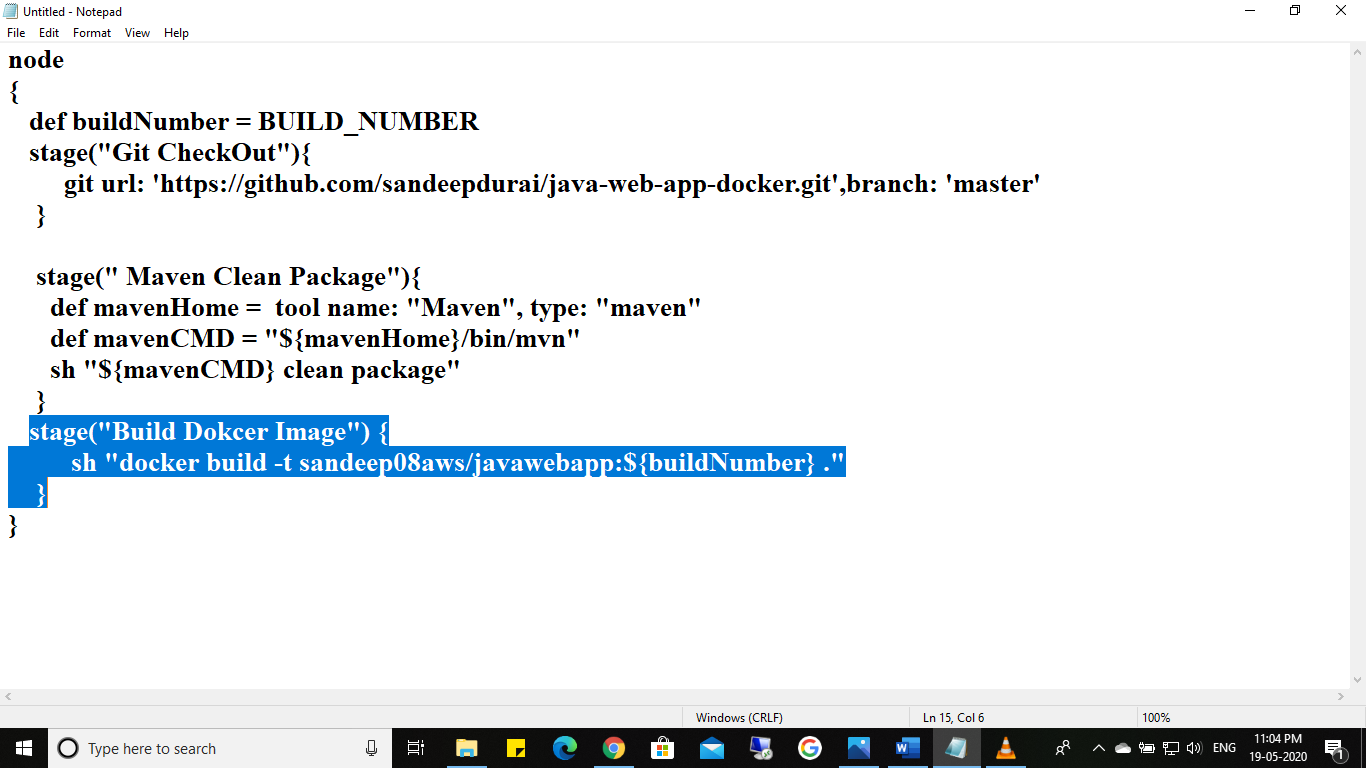
**Install maven using global tool configuration**



**Check using build stage 2**

**Stage 3**

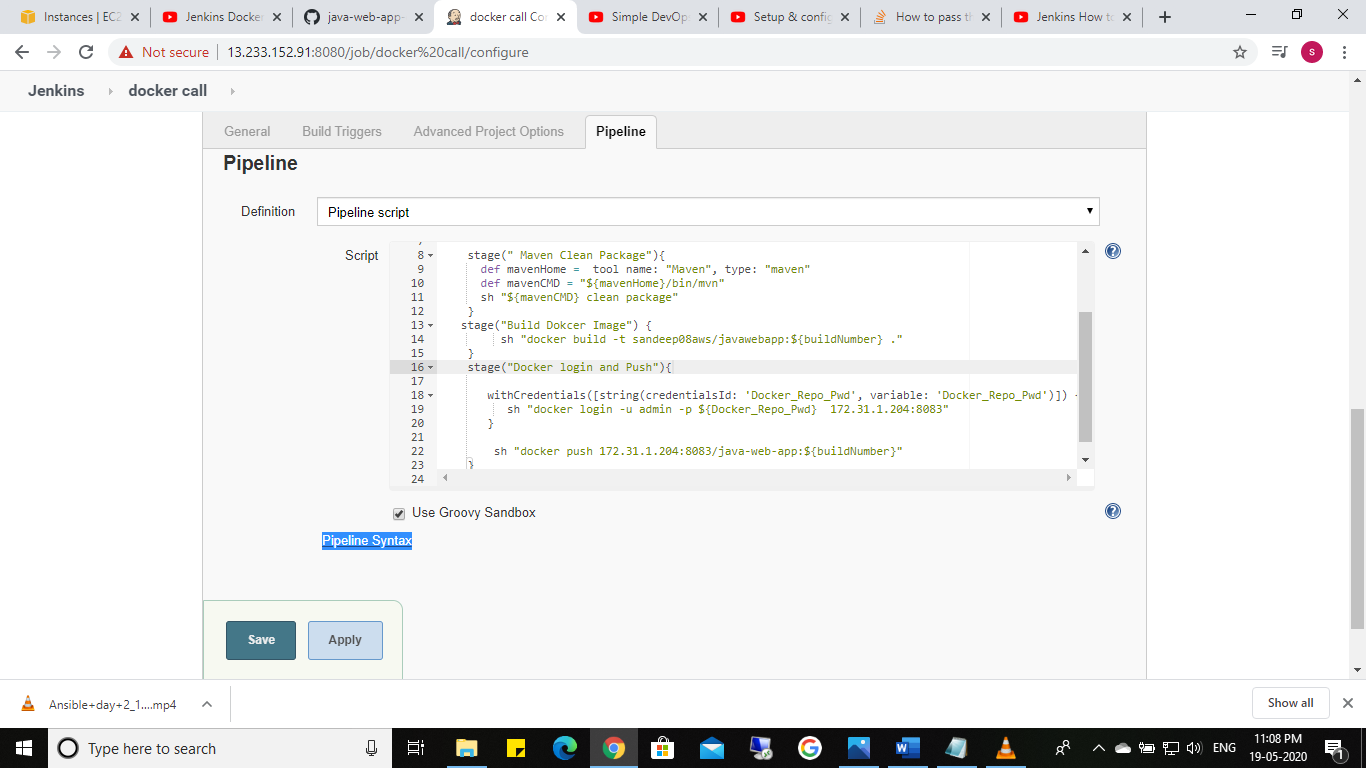
**Create docker image using docker file in github : version as build number**



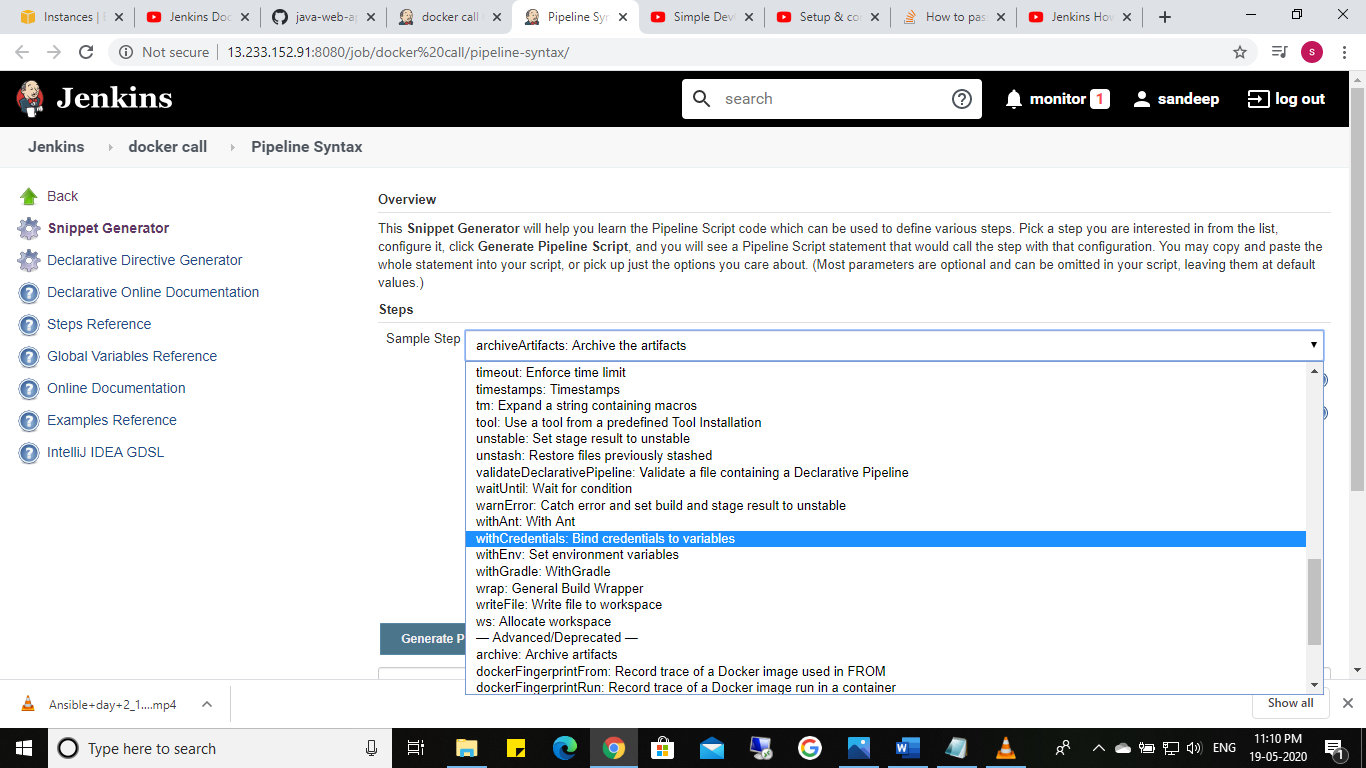
**Stage 4**

**For puch image to docker hub and log in :**

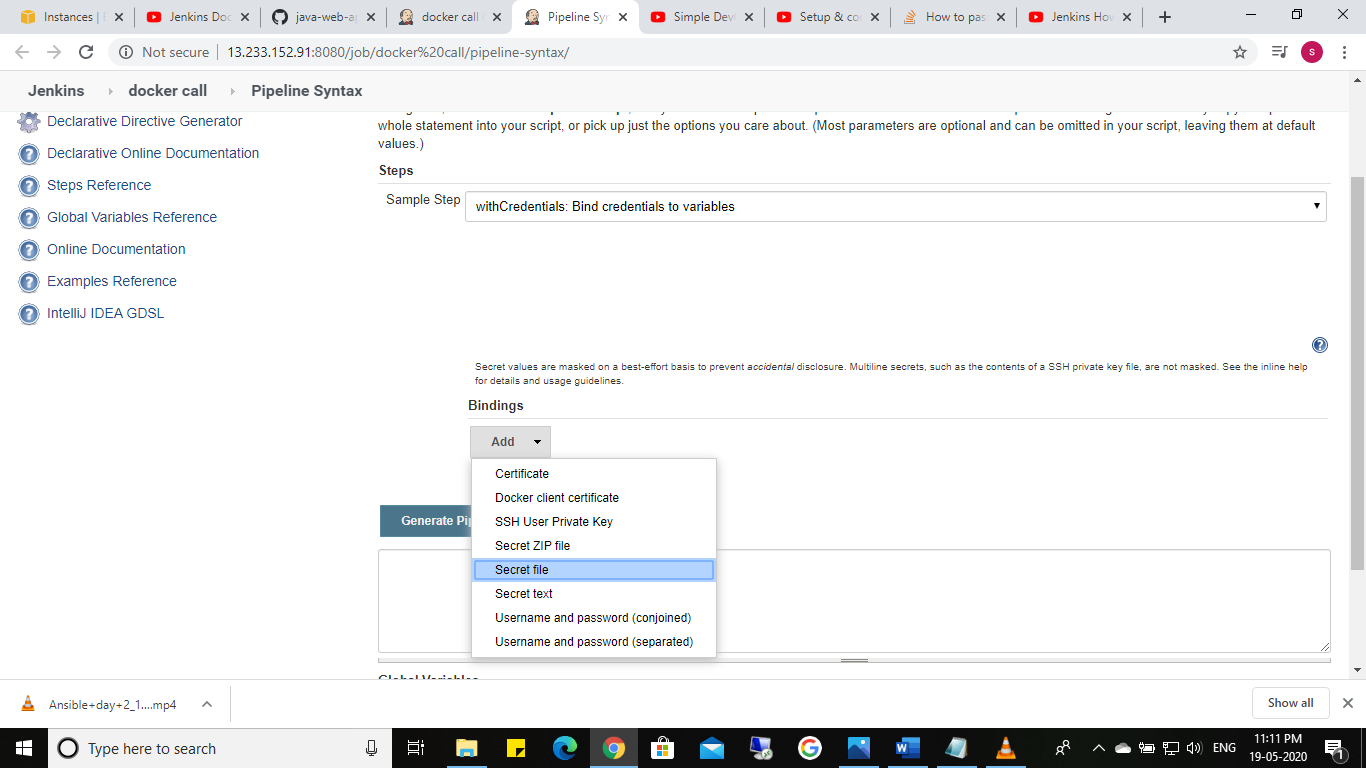
**Give credential as secret : using piplinesysntex**

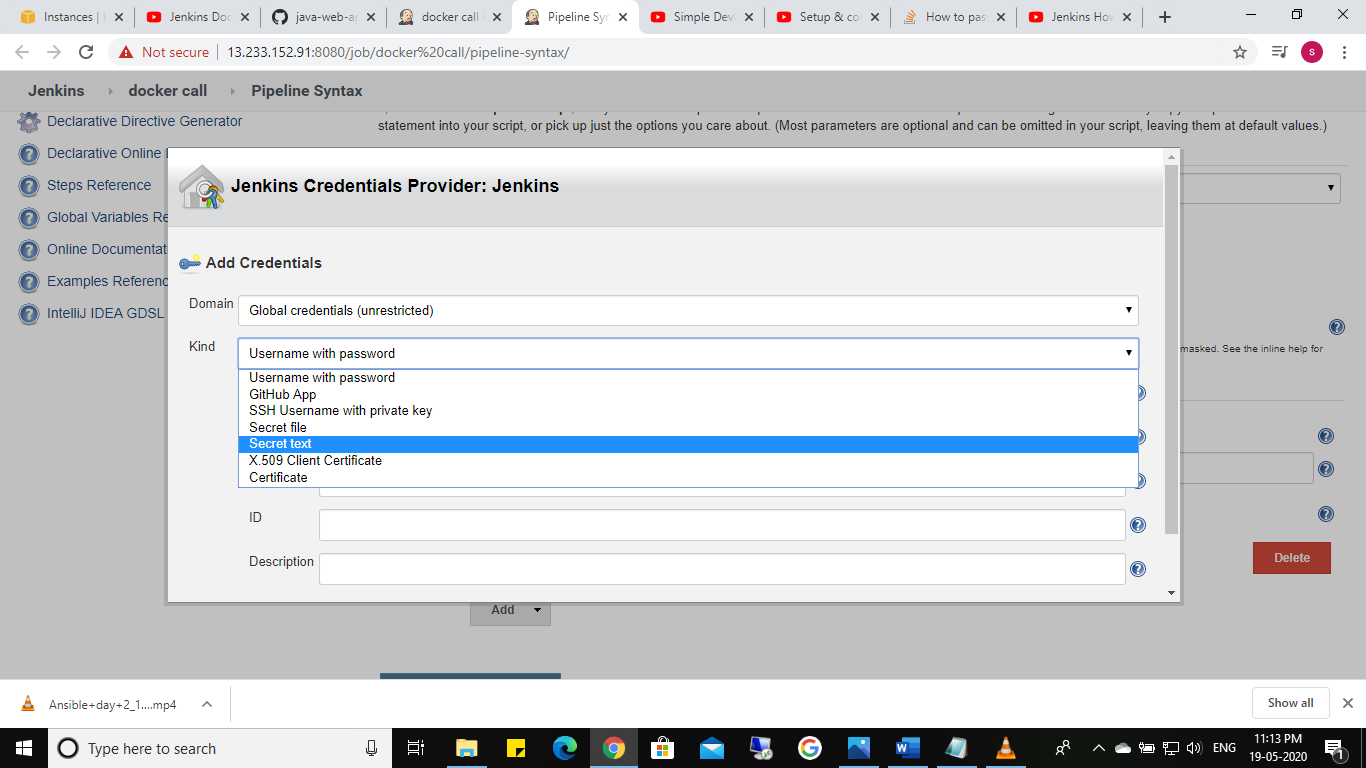


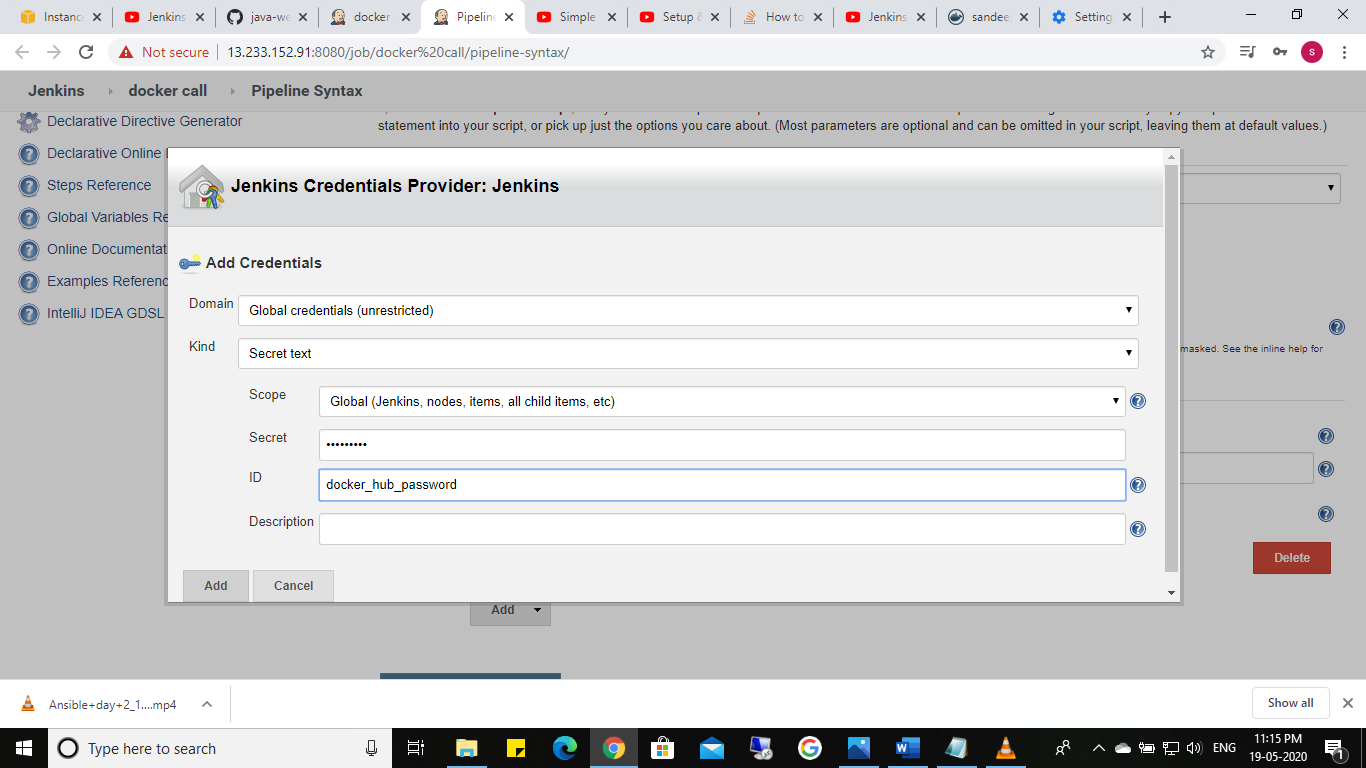
**With credential**



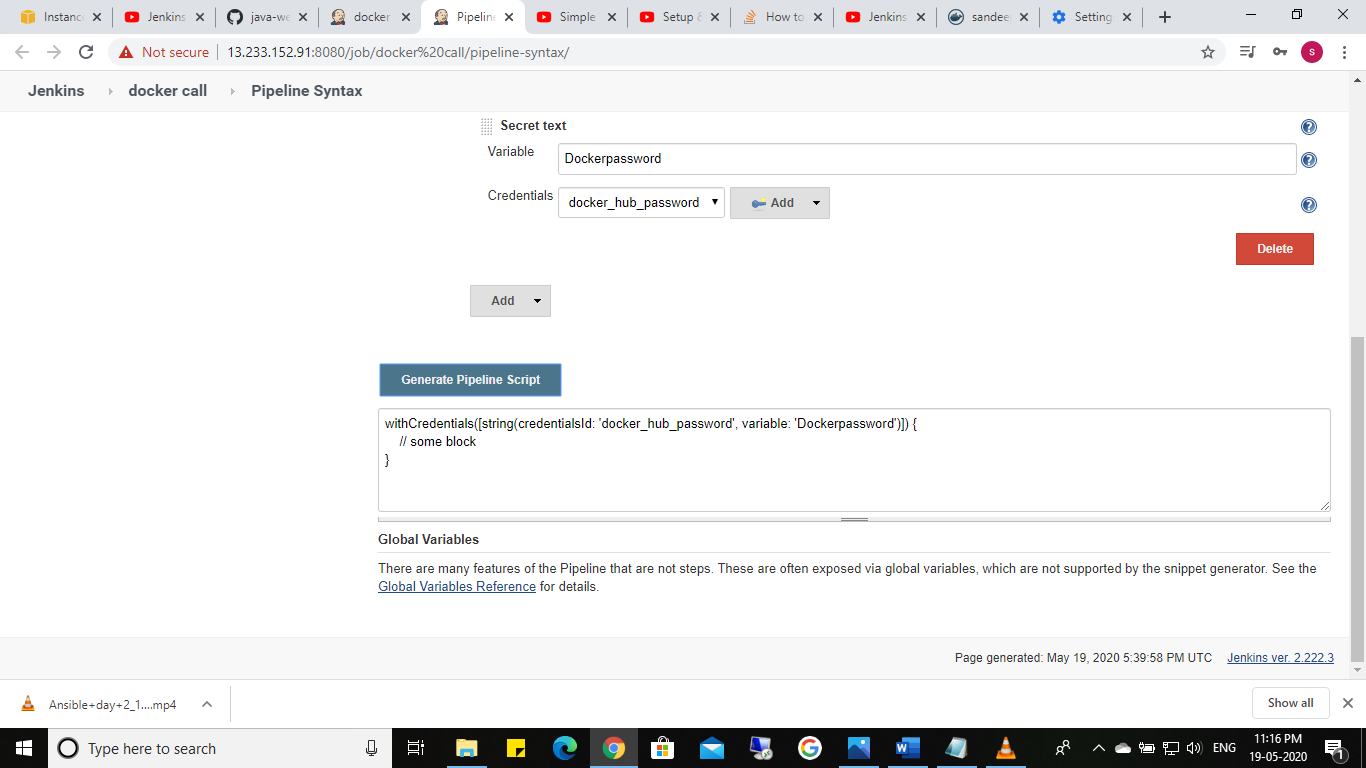
**Secret txt:**

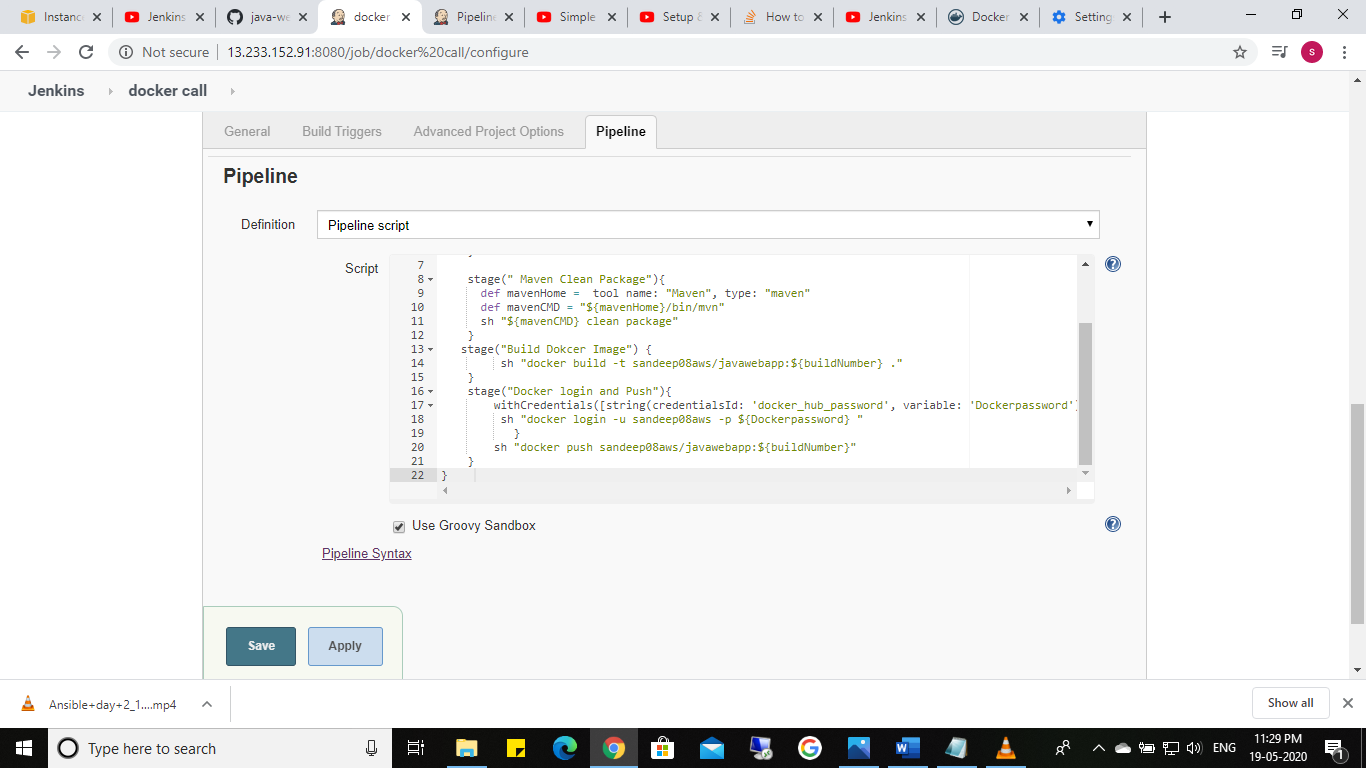






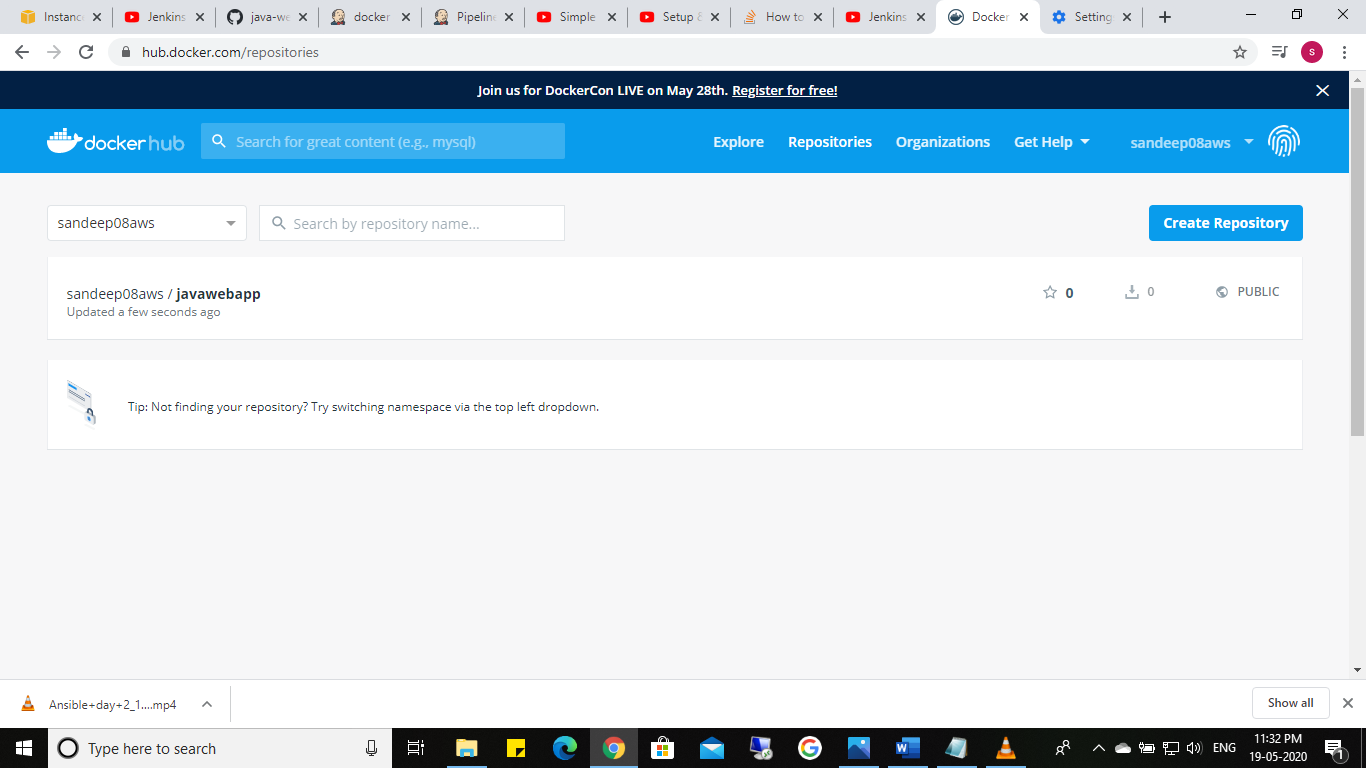
**Click generate pipeline script and copy the script**





To check the stage 4 buld and check in docker hub:

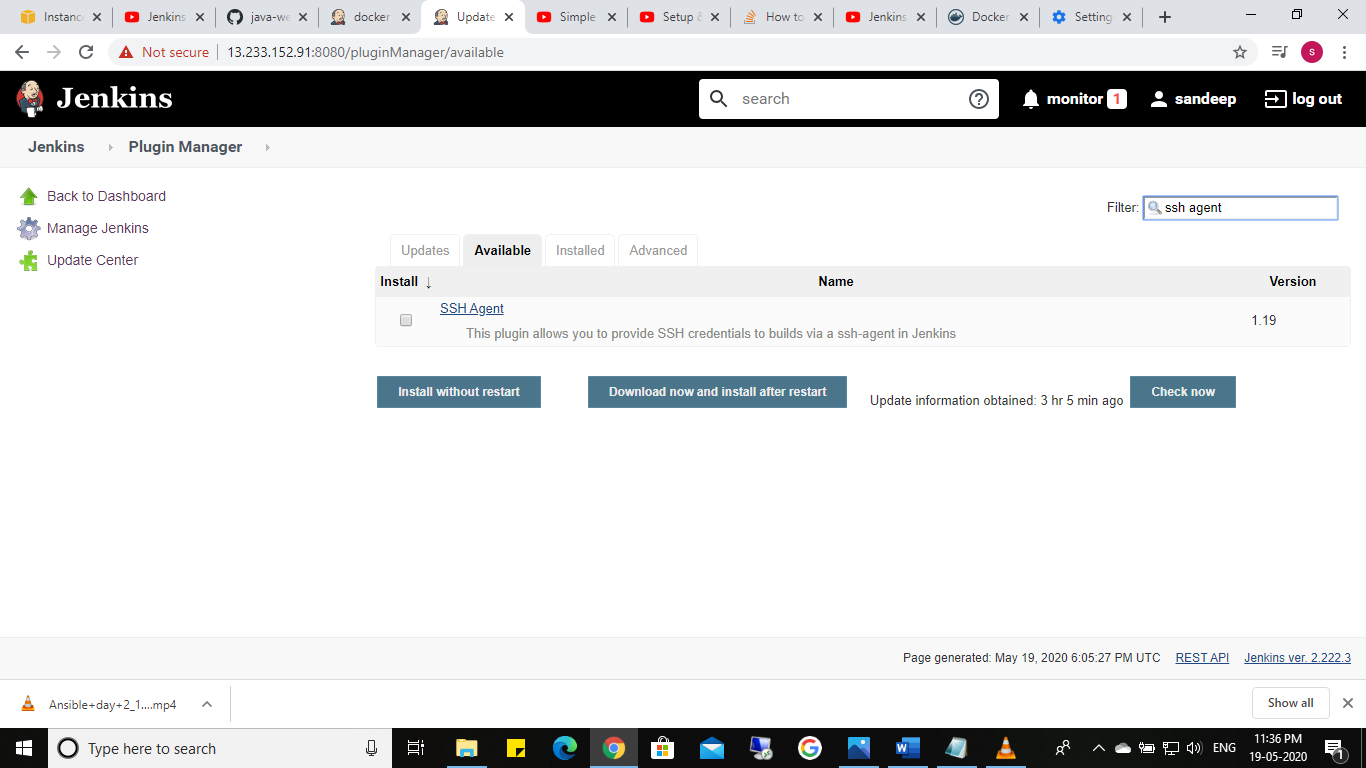
Image in pushed to docker repository



Stage 5

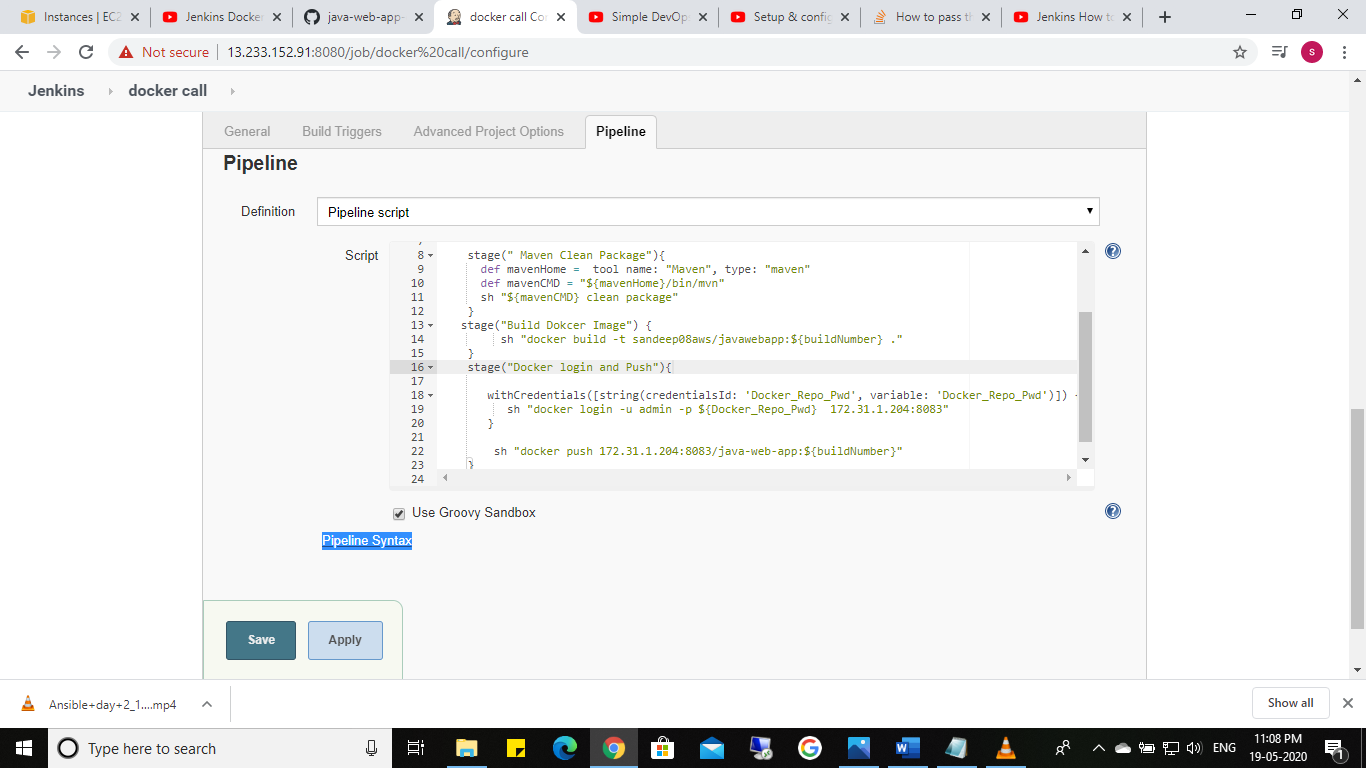
Deploy container to deployment server :

1. Install ssh user plugin :ssh agent

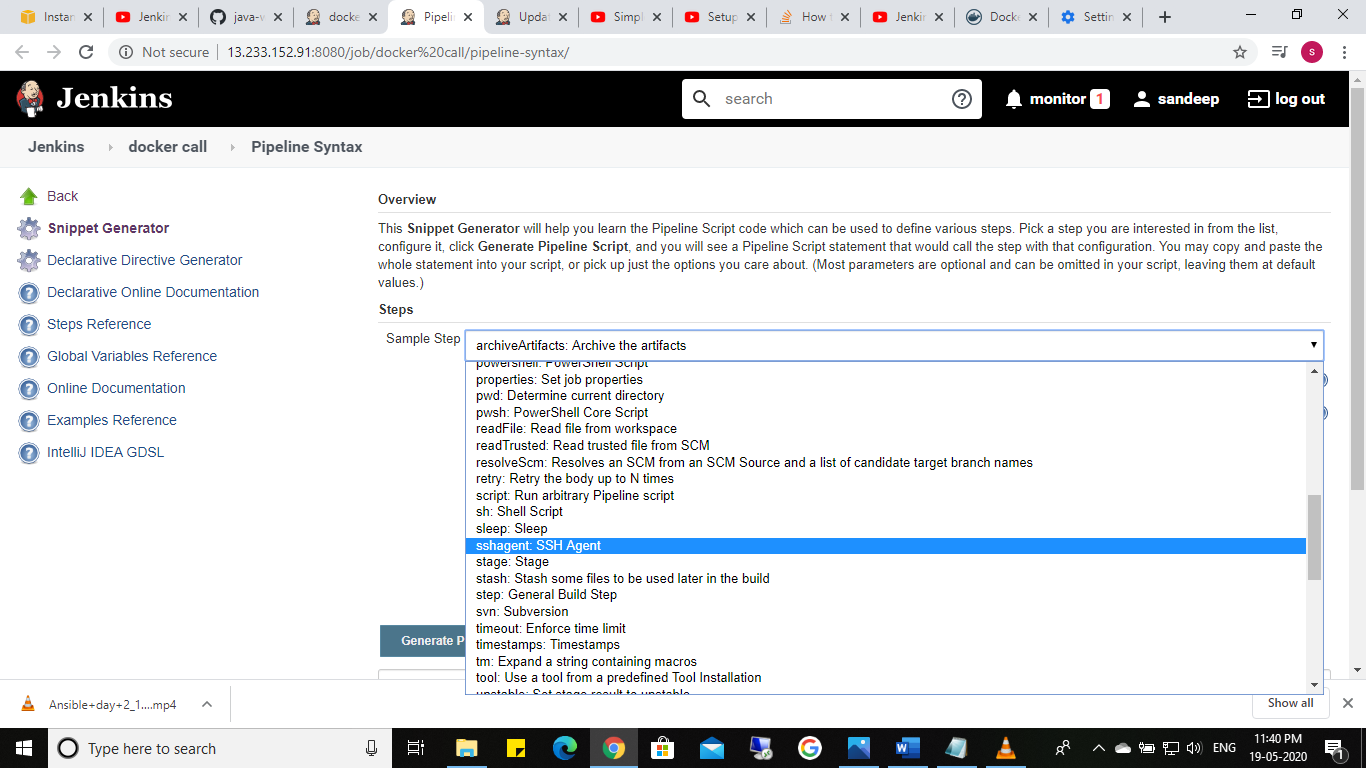


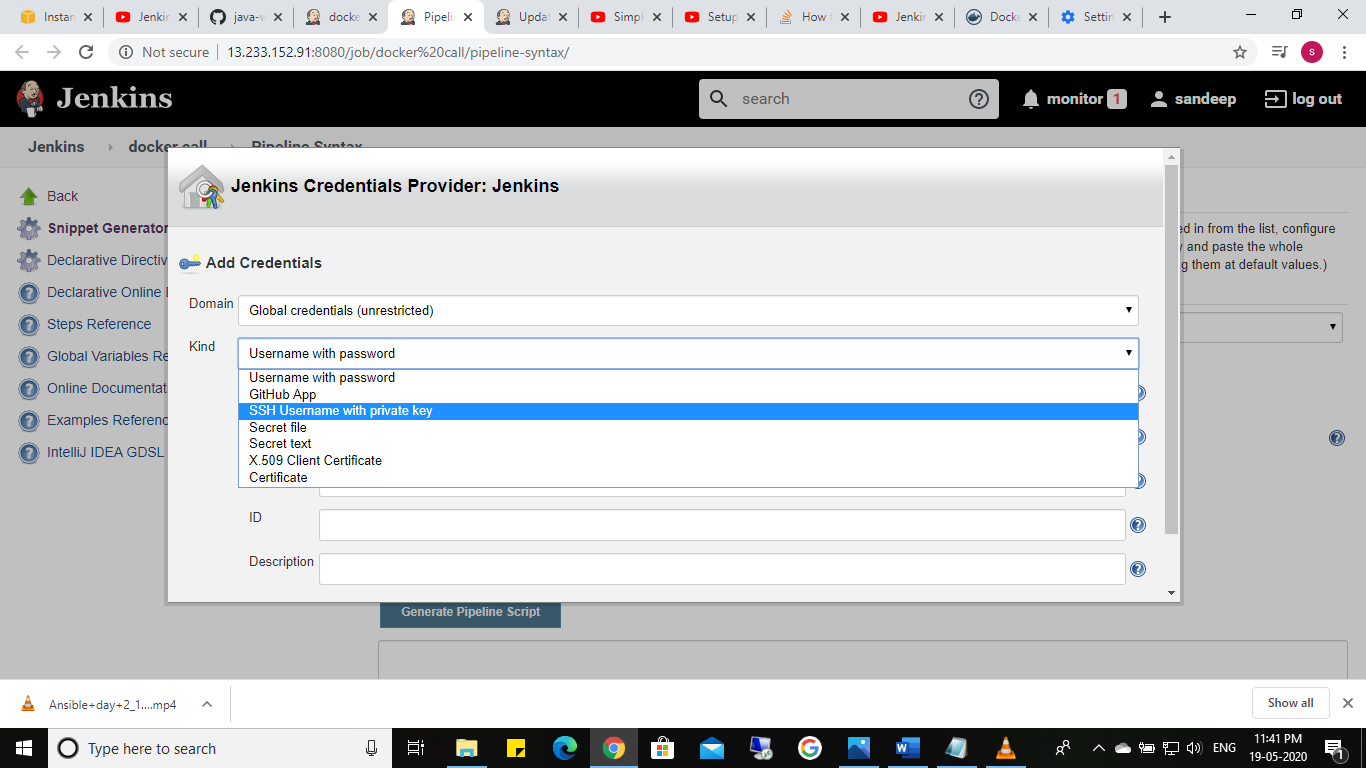
2.Connect Jenkins with docker deployer sever :

1. piplinesyntex:

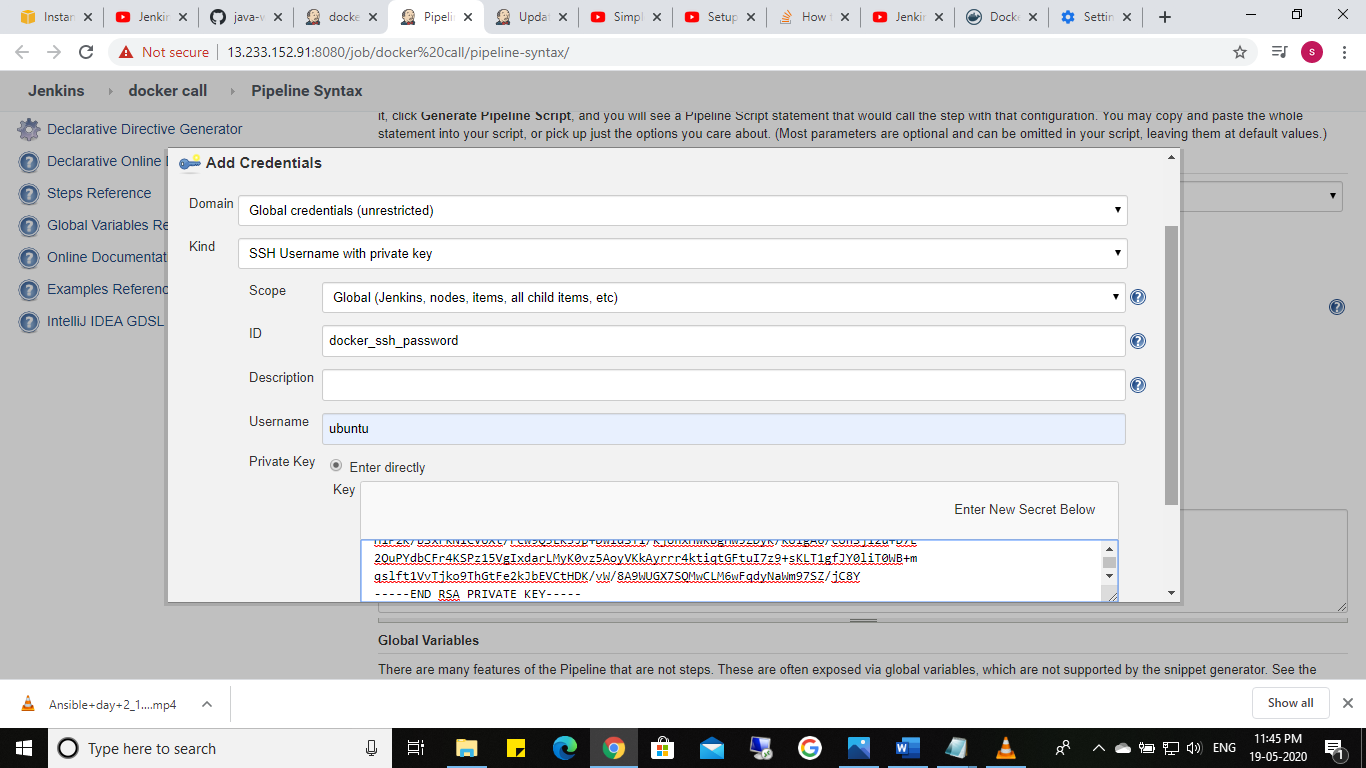


1. **Sshagent**

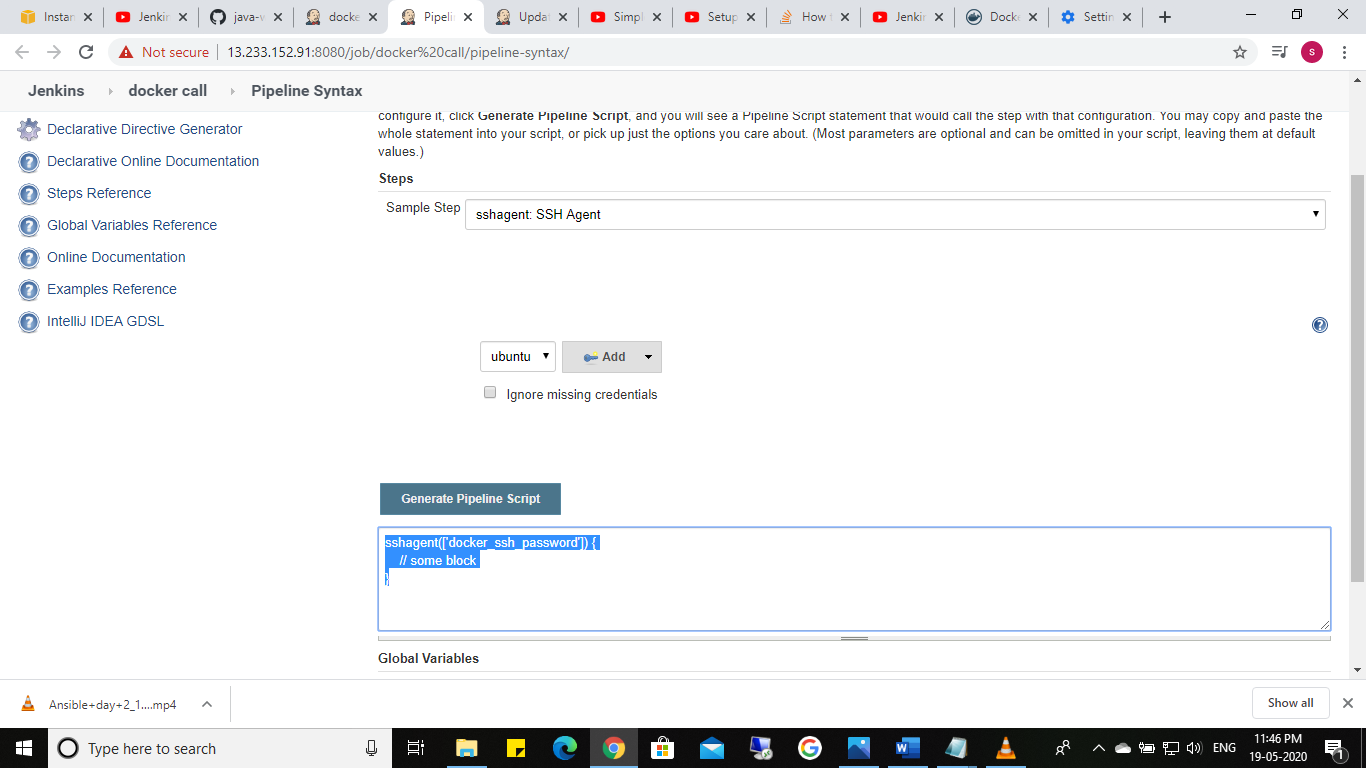


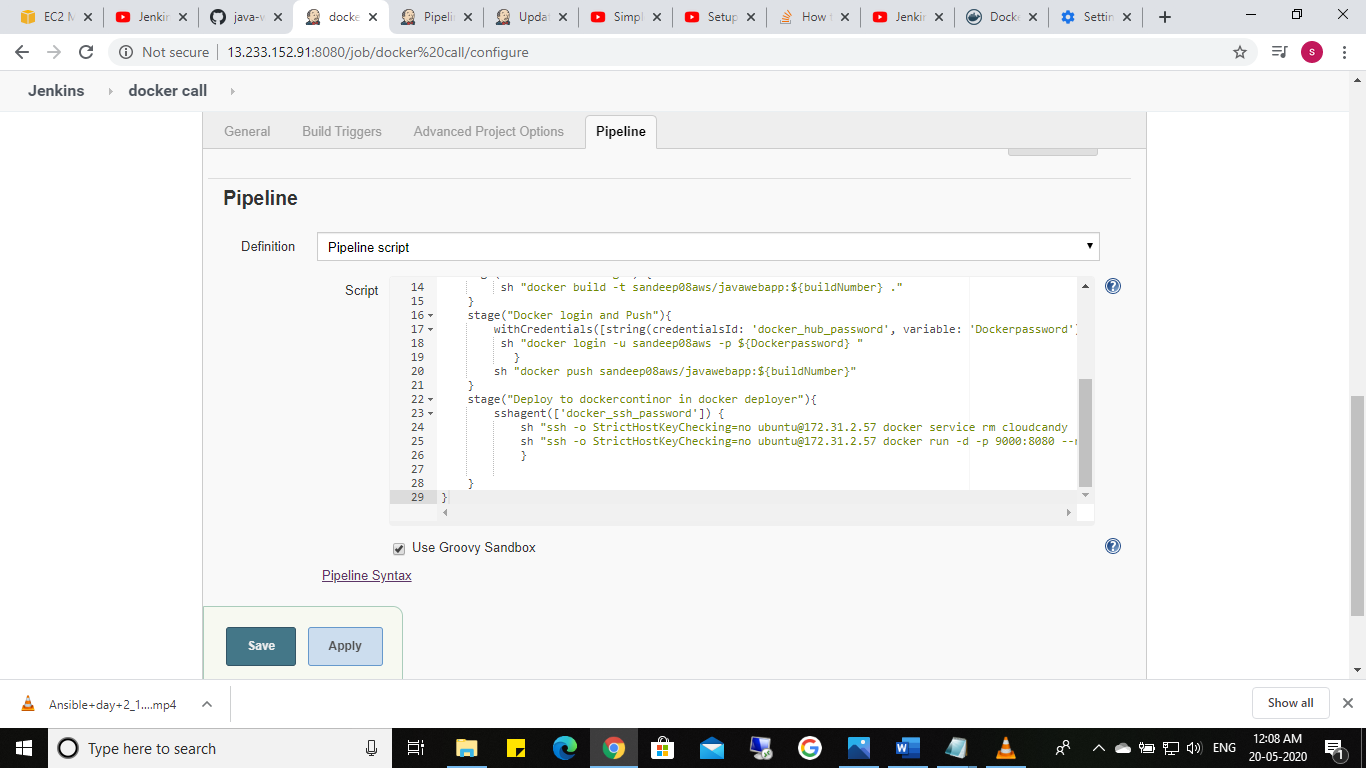


Add user name and .pem file

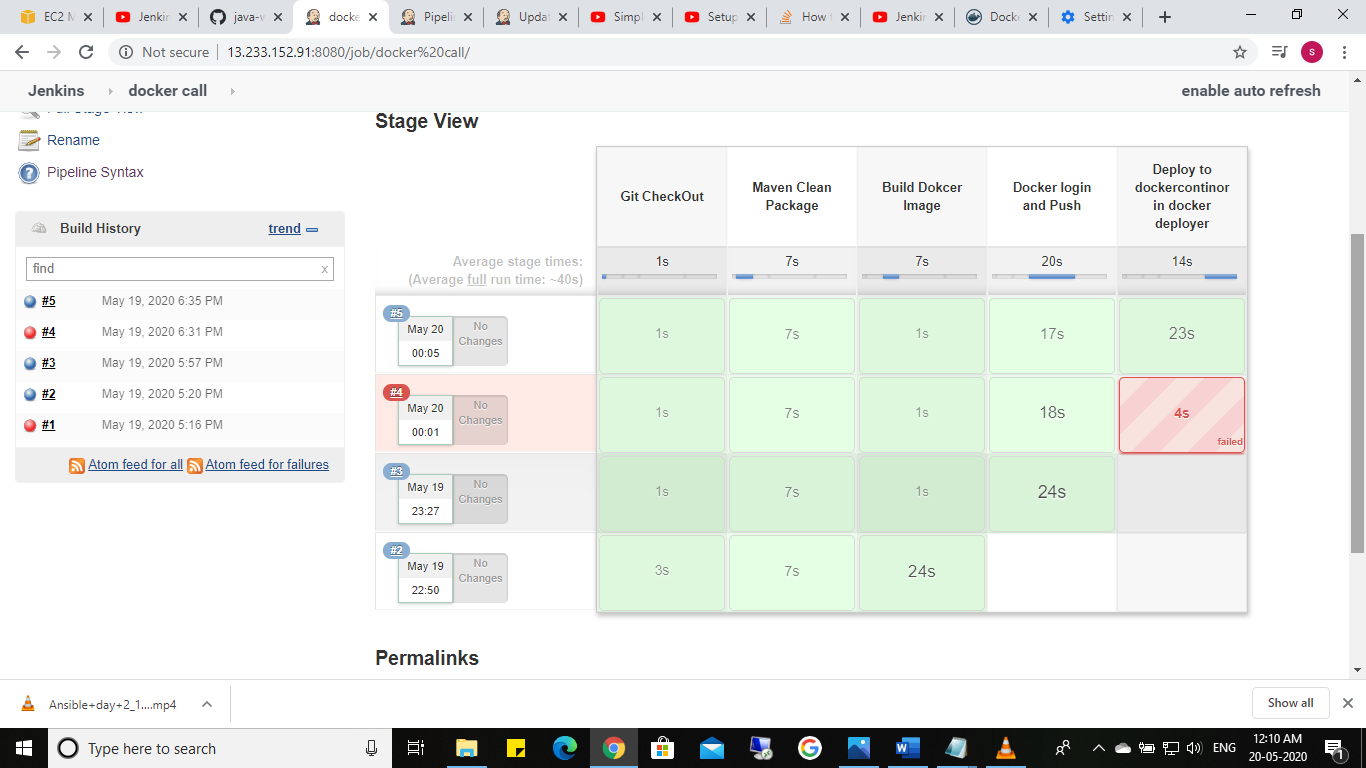


**Click generate pipeline script and copy the script**



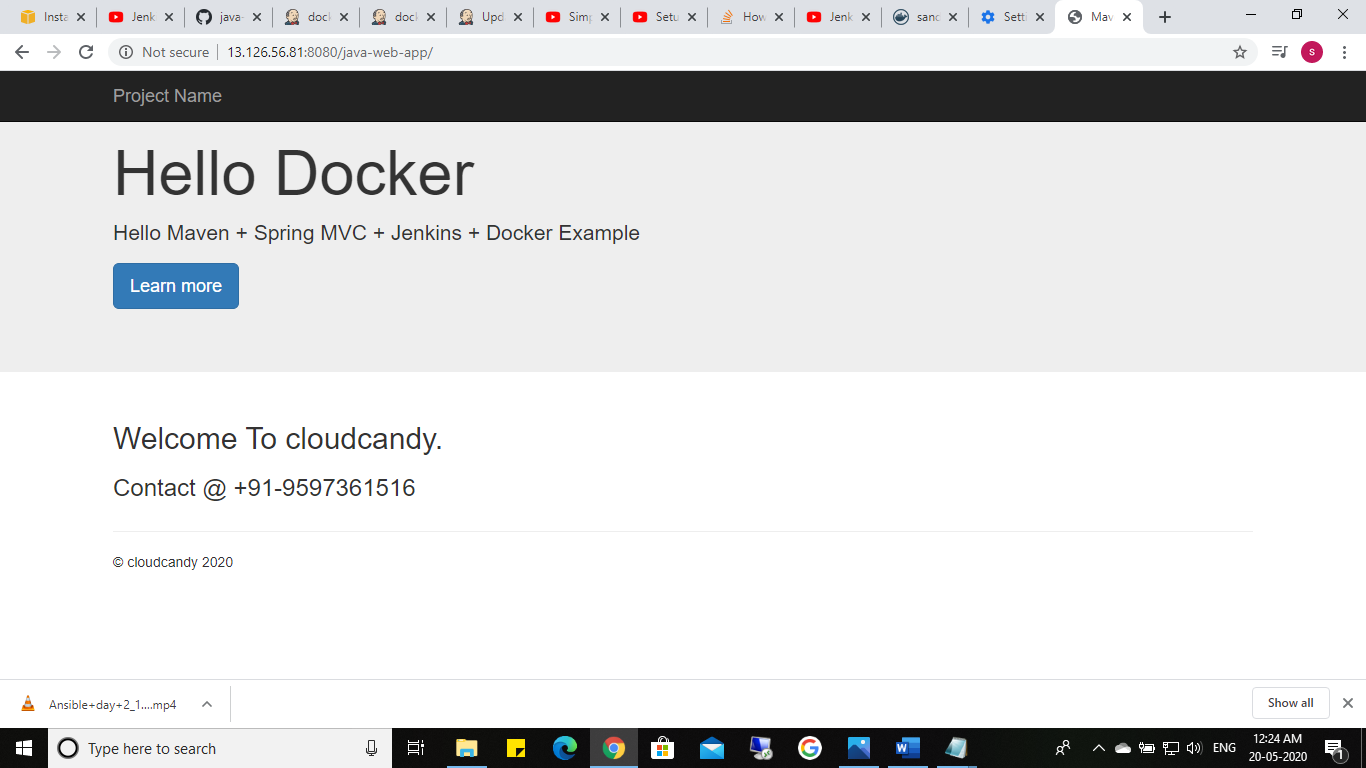
1. Run docker container from docker repository with pot number 8080
2. 

Build the script to run container:



Check the with<deployer public ip>:8080:

<http://13.126.56.81:8080/java-web-app/>



project Script:

node

{

def buildNumber = BUILD\_NUMBER

stage("Git CheckOut"){

git url: 'https://github.com/sandeepdurai/java-web-app-docker.git',branch: 'master'

}

stage(" Maven Clean Package"){

def mavenHome= tool name: "Maven", type: "maven"

def mavenCMD = "${mavenHome}/bin/mvn"

sh "${mavenCMD} clean package"

}

stage("Build Dokcer Image") {

sh "docker build -t sandeep08aws/javawebapp:${buildNumber} ."

}

stage("Docker login and Push"){

withCredentials([string(credentialsId: 'docker\_hub\_password', variable: 'Dockerpassword')]){

sh "docker login -u sandeep08aws -p ${Dockerpassword} "

}

sh "docker push sandeep08aws/javawebapp:${buildNumber}"

}

stage("Deploy to dockercontinor in docker deployer"){

sshagent(['docker\_ssh\_password']) {

sh "ssh -o StrictHostKeyChecking=no ubuntu@172.31.2.57 docker rm -f cloudcandy || true"

sh "ssh -o StrictHostKeyChecking=no ubuntu@172.31.2.57 docker run -d -p 8080:8080 --name cloudcandy sandeep08aws/javawebapp:${buildNumber}"

}

}