How to Integrate jenkins with docker ci/cd

Step1: **Launch Ubuntu 18.0 instance for Jenkins**

- **Installing the Default JRE/JDK**
  - sudo apt update
  - sudo apt install default-jdk
  - `java -version`

- Installing Jenkins
  - **First, add the repository key to the system:**
    wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -

  - **append the Debian package repository address to the server's `sources.list`:**
    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`

  - **`Starting Jenkins`**
    `systemctl start jenkins`

  - **`To check the status`**

```
root@ip-172-31-8-229:/home/ubuntu# systemctl status jenkins
● jenkins.service - LSB: Start Jenkins at boot time
   Loaded: loaded (/etc/init.d/jenkins; generated)
   Active: active (exited) since Sun 2021-12-26 16:11:20 UTC; 2min 59s ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 0 (limit: 2347)
   CGroup: /system.slice/jenkins.service

Dec 26 16:11:19 ip-172-31-8-229 systemd[1]: Starting LSB: Start Jenkins at boot
Dec 26 16:11:19 ip-172-31-8-229 jenkins[7599]: Correct java version found
Dec 26 16:11:19 ip-172-31-8-229 jenkins[7599]:  * Starting Jenkins Automation Se
Dec 26 16:11:19 ip-172-31-8-229 su[7650]: Successful su for jenkins by root
Dec 26 16:11:19 ip-172-31-8-229 su[7650]: + ??? root:jenkins
Dec 26 16:11:19 ip-172-31-8-229 su[7650]: pam_unix(su:session): session opened f
Dec 26 16:11:19 ip-172-31-8-229 su[7650]: pam_unix(su:session): session closed f
Dec 26 16:11:20 ip-172-31-8-229 jenkins[7599]:    ...done.
Dec 26 16:11:20 ip-172-31-8-229 systemd[1]: Started LSB: Start Jenkins at boot t
lines 1-16/16 (END)
```

  - **`Install docker`**

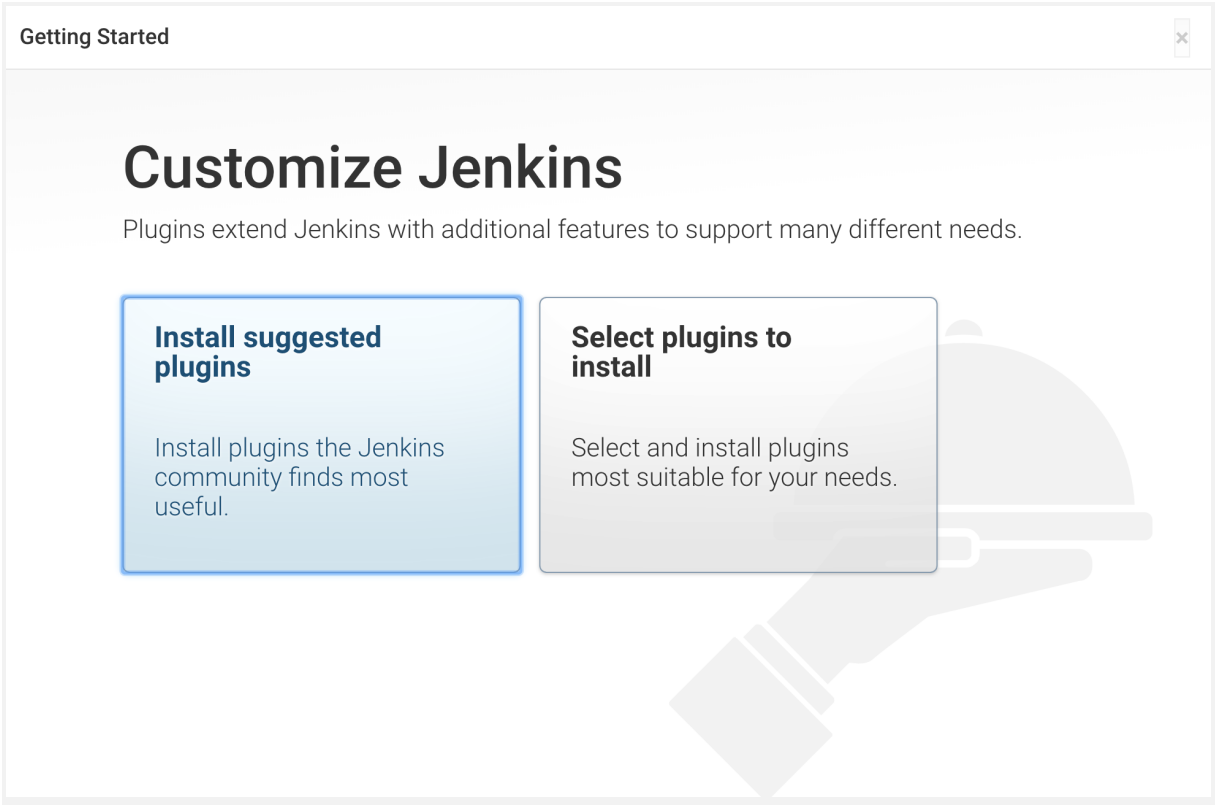
- `apt install docker.io -y`
- `usermod -aG docker jenkins`
- `systemctl restart jenkins`

- Opening the Firewall:
  - `ufw allow 8080`
  - `ufw status`
  - `ufw allow OpenSSH`
  - `ufw enable`

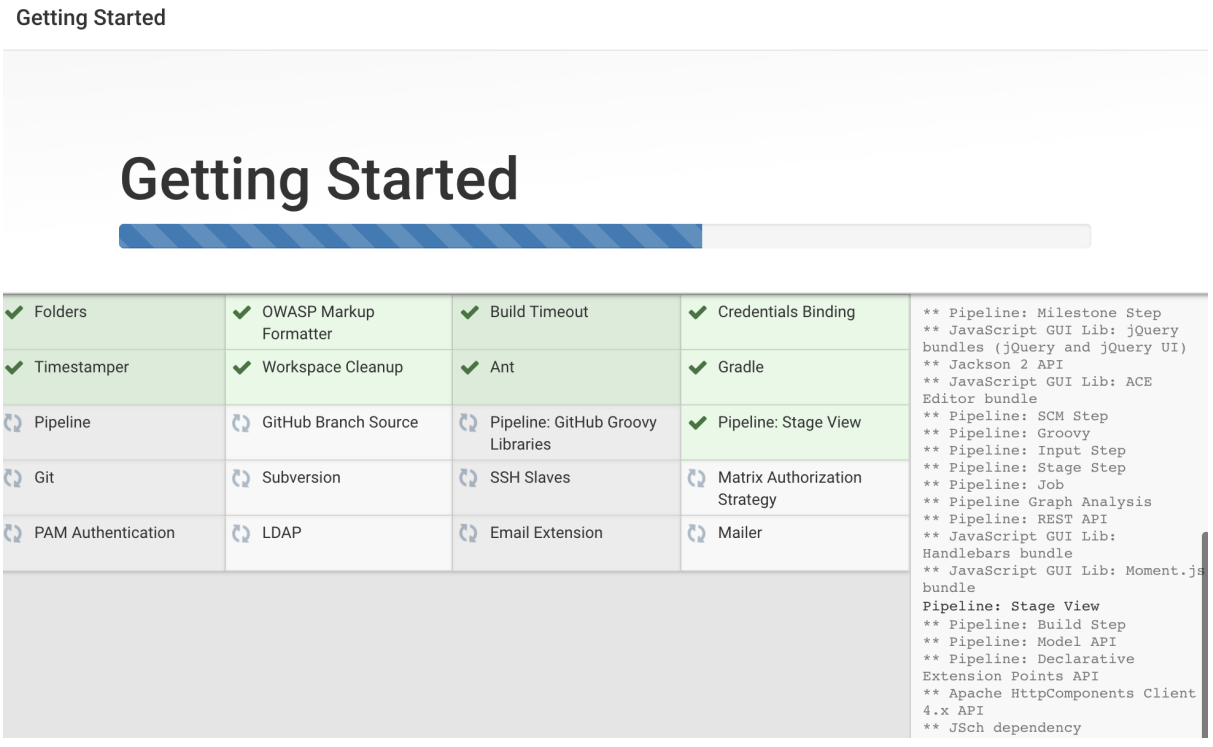
- **Setting Up Jenkins:**
  To set up your installation, visit Jenkins on its default port, 8080, using your server domain name or IP address: http://your_server_ip_or_domain:8080

  - **use the `cat` command to display the password:**
    `cat /var/lib/jenkins/secrets/initialAdminPas`

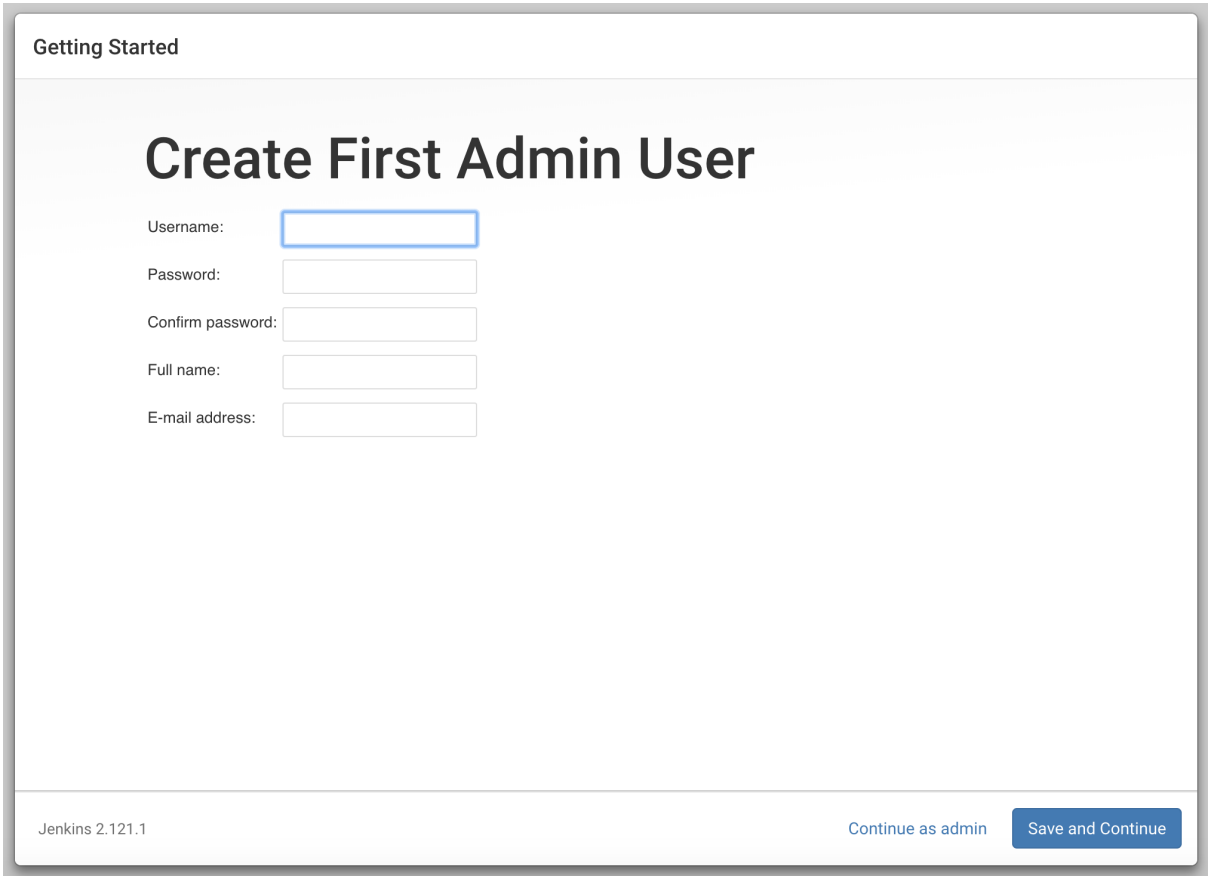
  - Copy the 32-character alphanumeric password from the terminal and paste it into the Administrator password field, then click Continue.
  - The next screen presents the option of installing suggested plugins or selecting



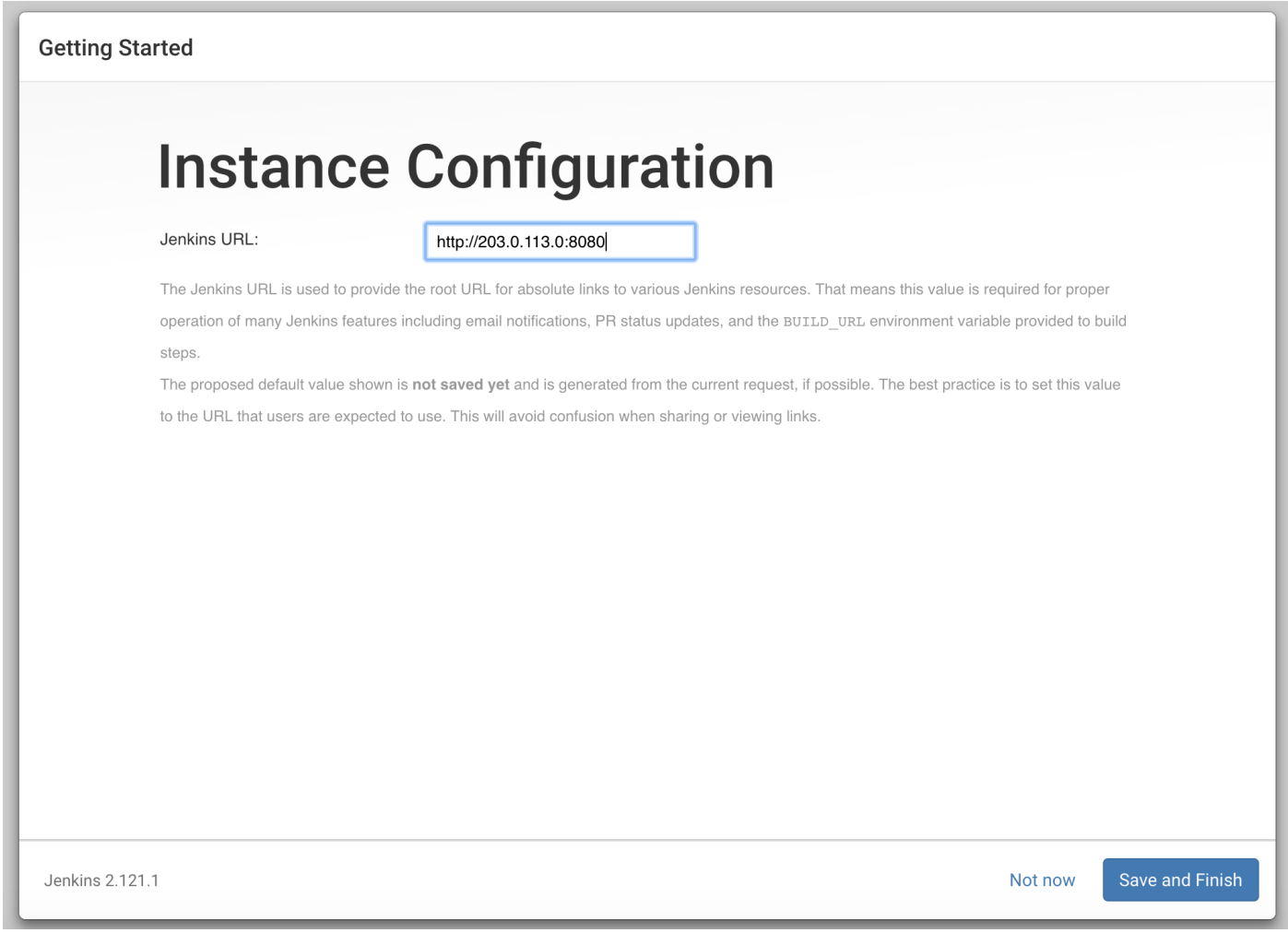
❖ We'll click the **Install suggested plugins** option, which will immediately begin the installat



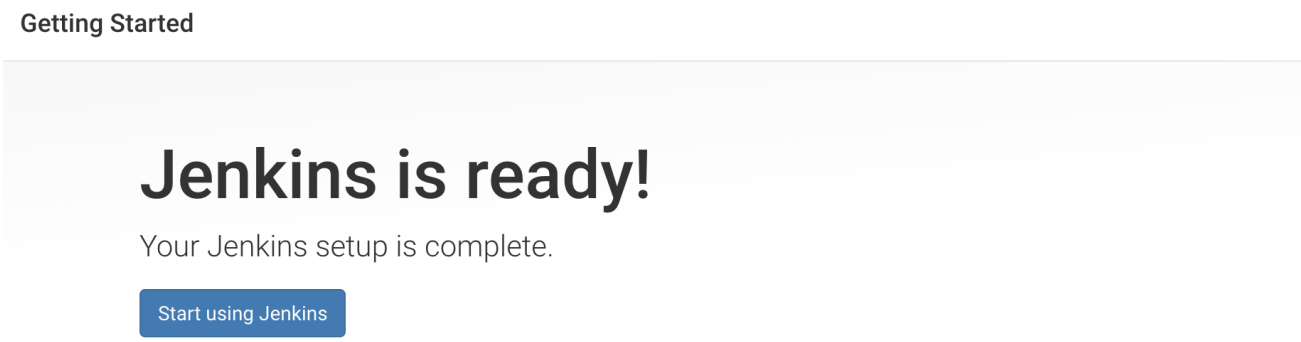
❖ **create the user:**



- You will see an **Instance Configuration** page that will ask you to confirm the preferred URL fo instance. Confirm either the domain name for your server or your server's IP address:



- After confirming the appropriate information, click **Save and Finish**. You will see a confirmati confirming that **"Jenkins is Read**



❖

❖ **Launch ubuntu instance for Docker:**

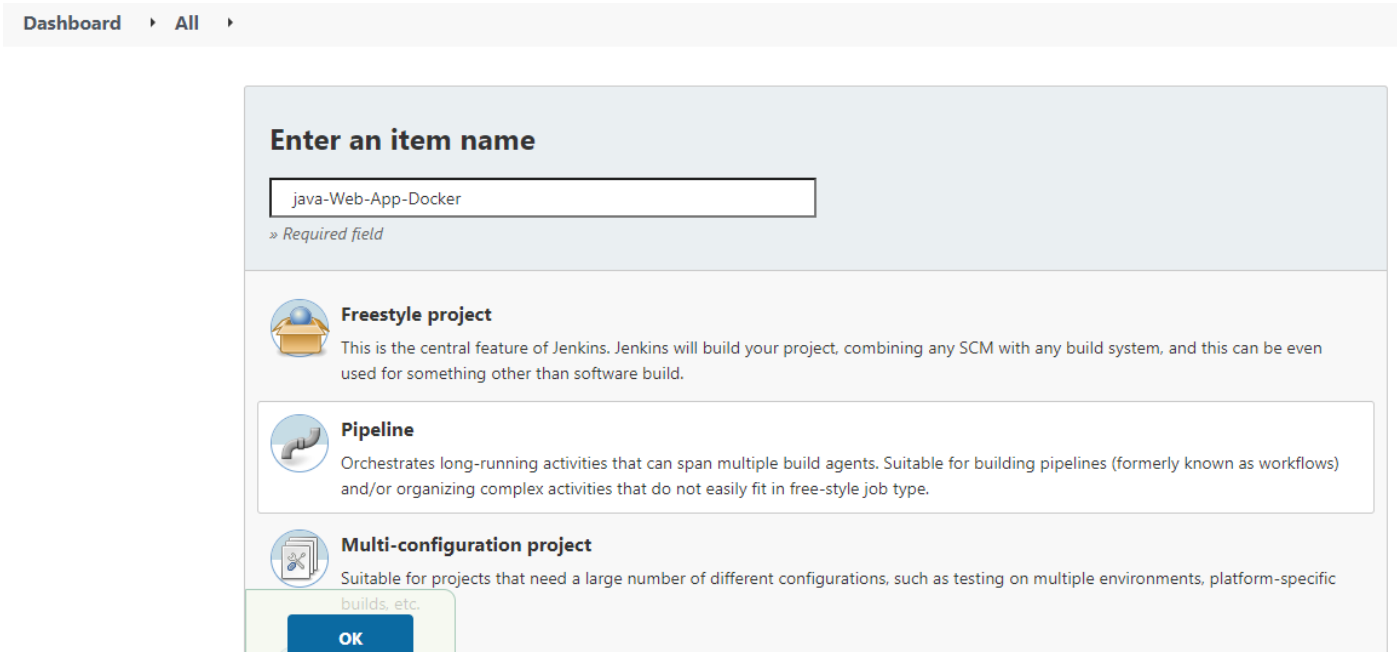
➢ Install of Docker:

➢ Apt update

➢ apt install docker.io -y

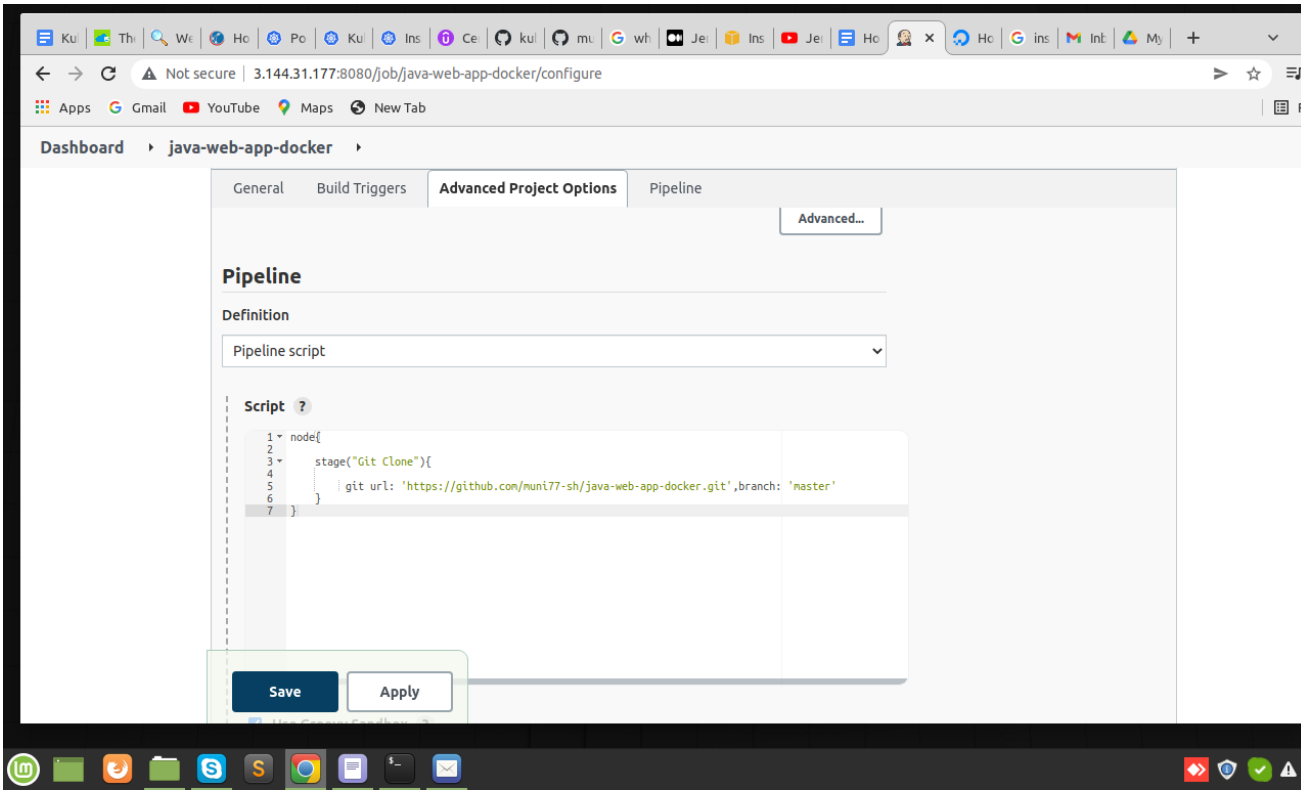
➢ usermod -aG docker ubuntu

❖ **Create CI/CD pipeline Job:**



❖ Write a pipeline Script:

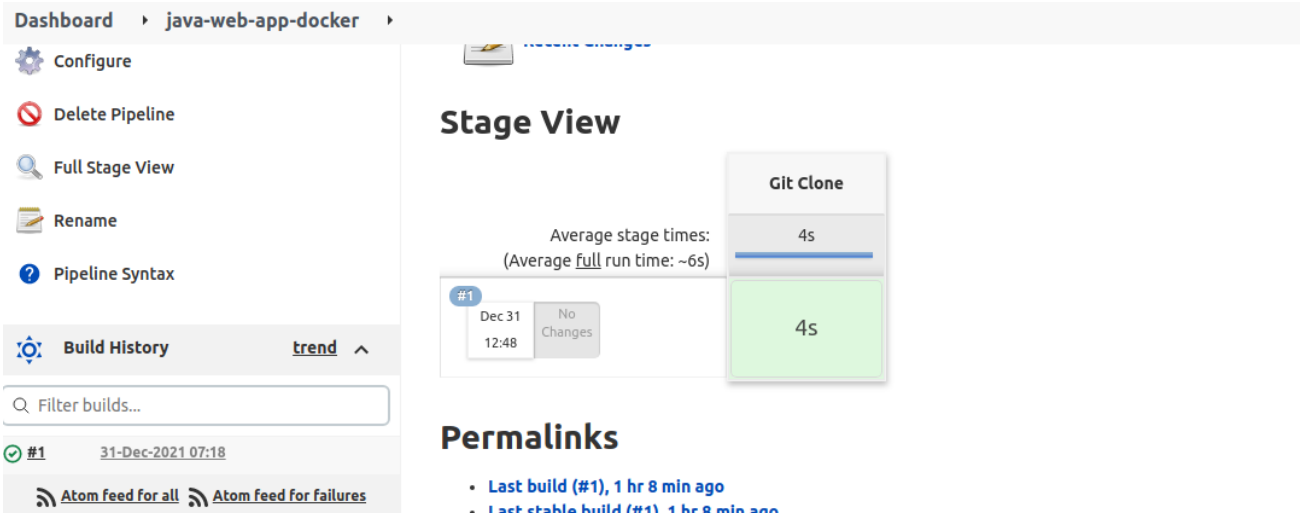
Step1: Git Clone:



→ Save it

→ trigger the Build

→ Build is Success



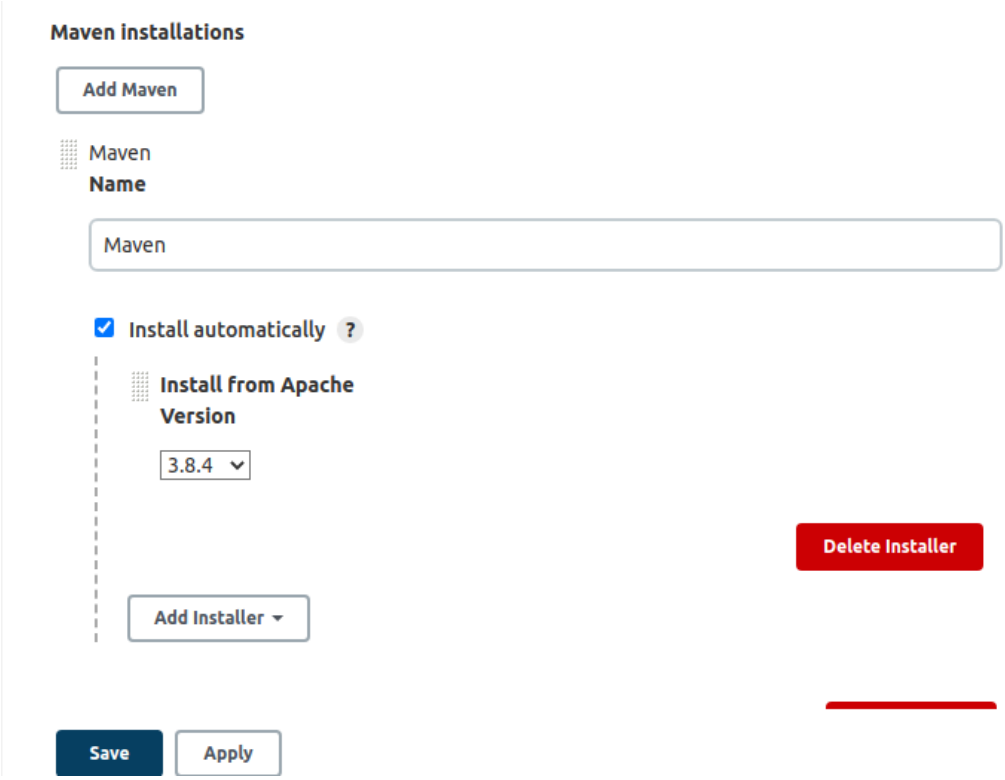
Step2: Maven Package:

Global Tool Configuration:

→ Click the "Manage Jenkins"

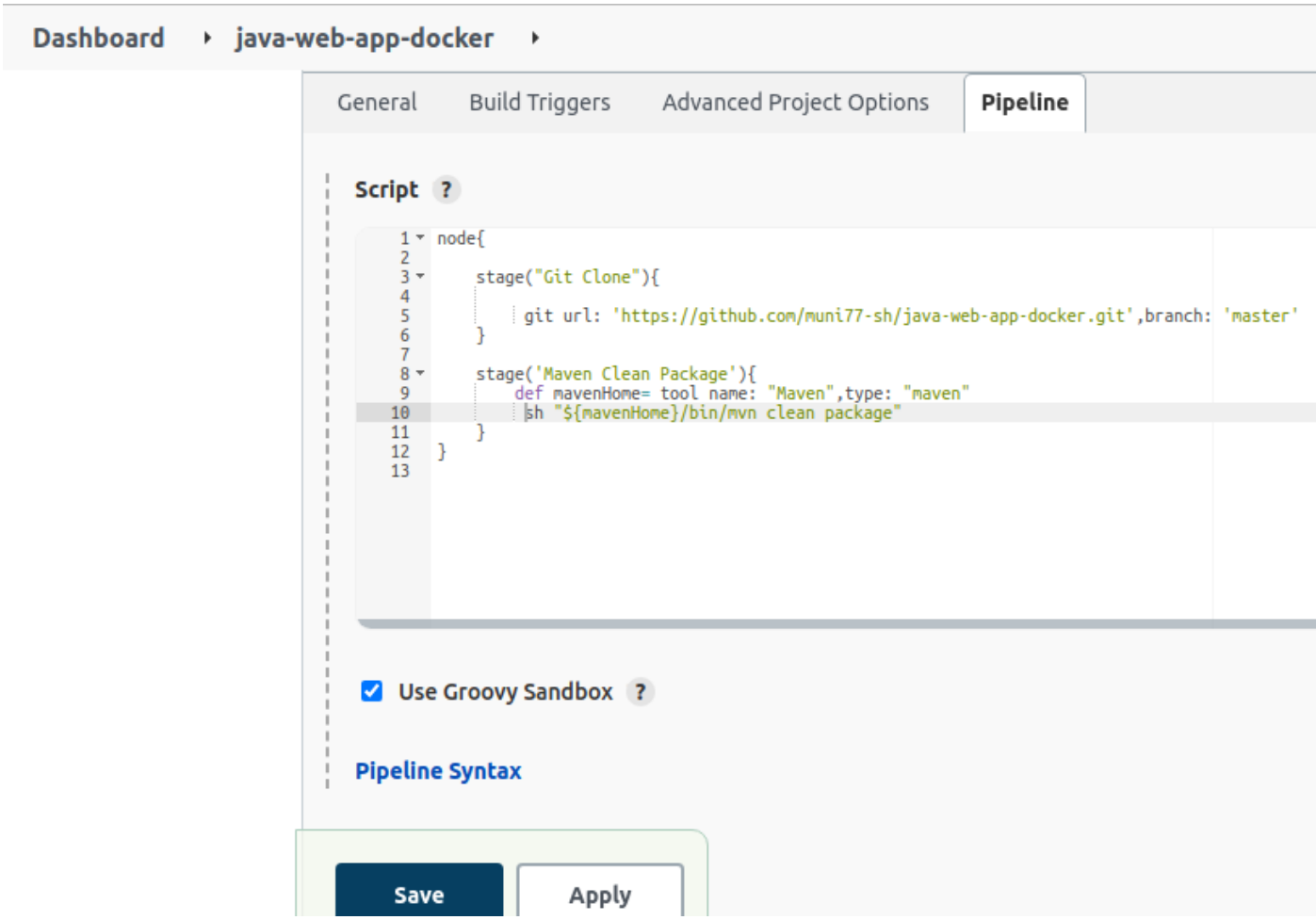
→ Click the "Global Tool Configuration"

→ Click on Maven Installation and set the Maven path.

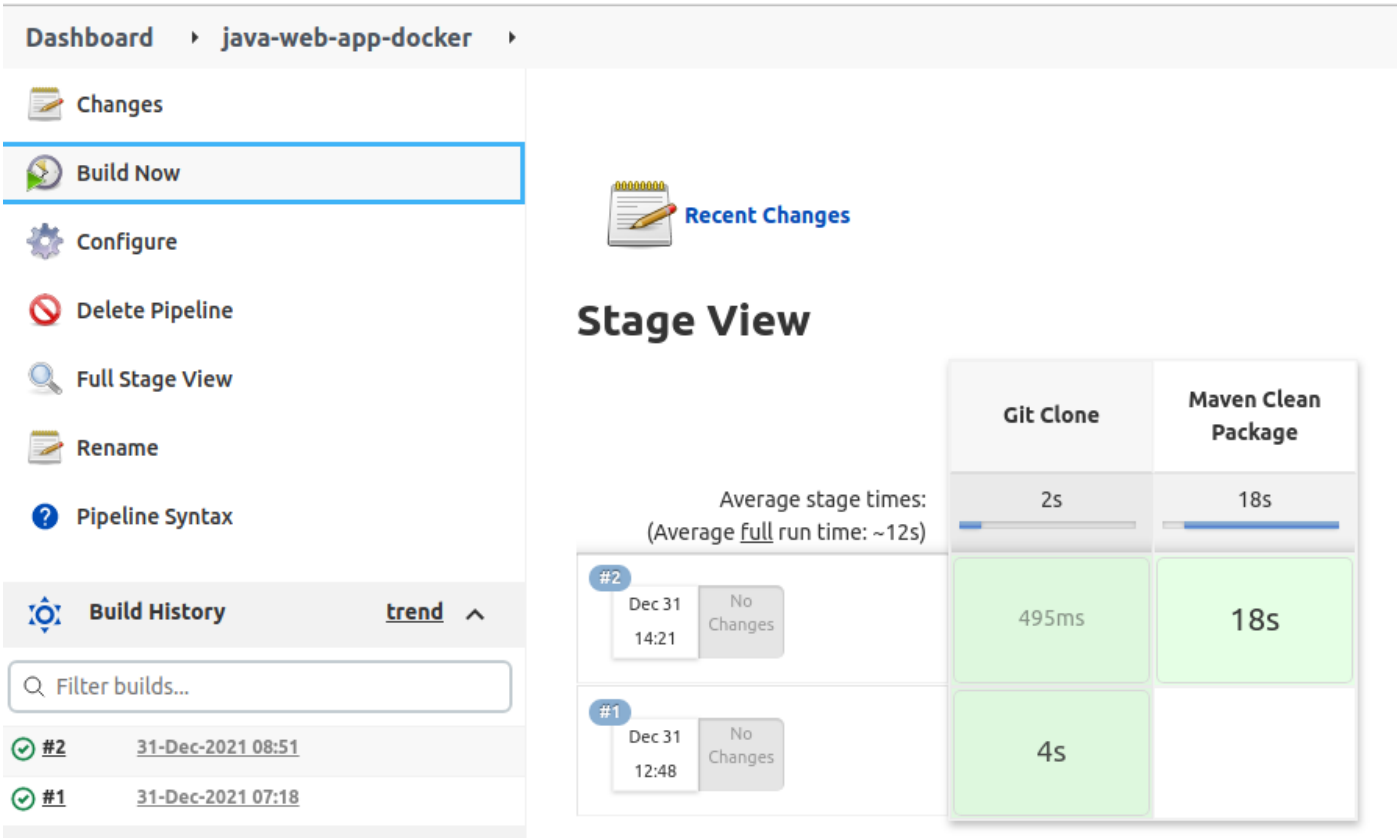


→ Finally click on save and your Maven configuration is done.

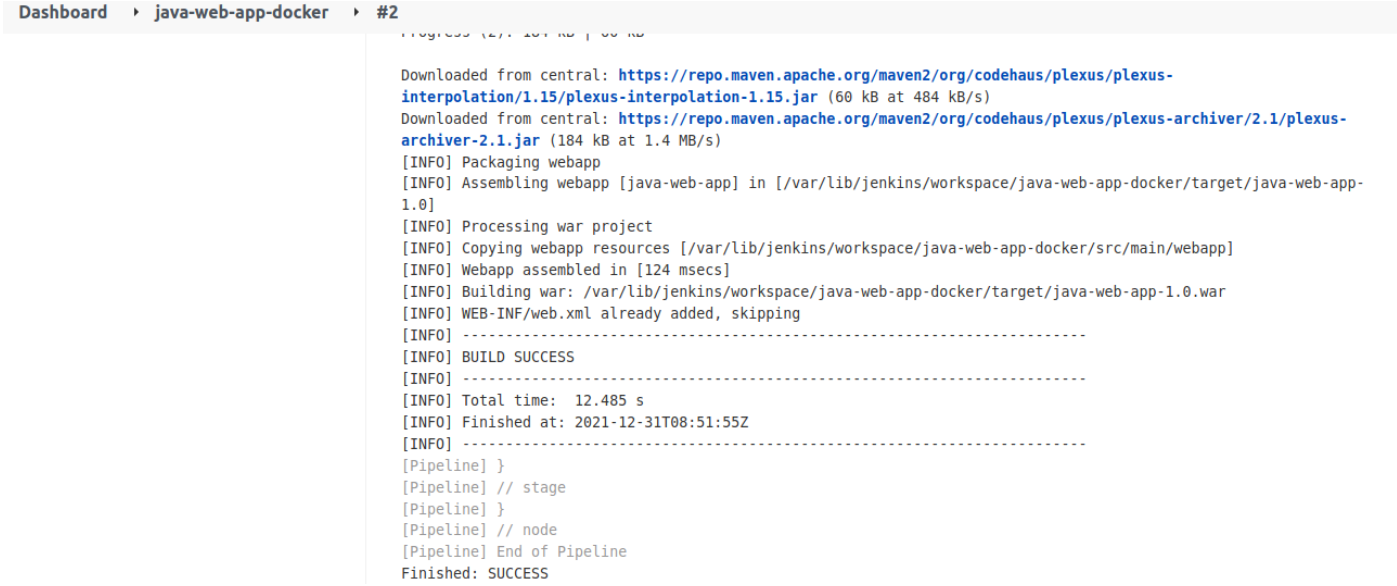
Then, goto “pipeline job” → Click on “Configure” →



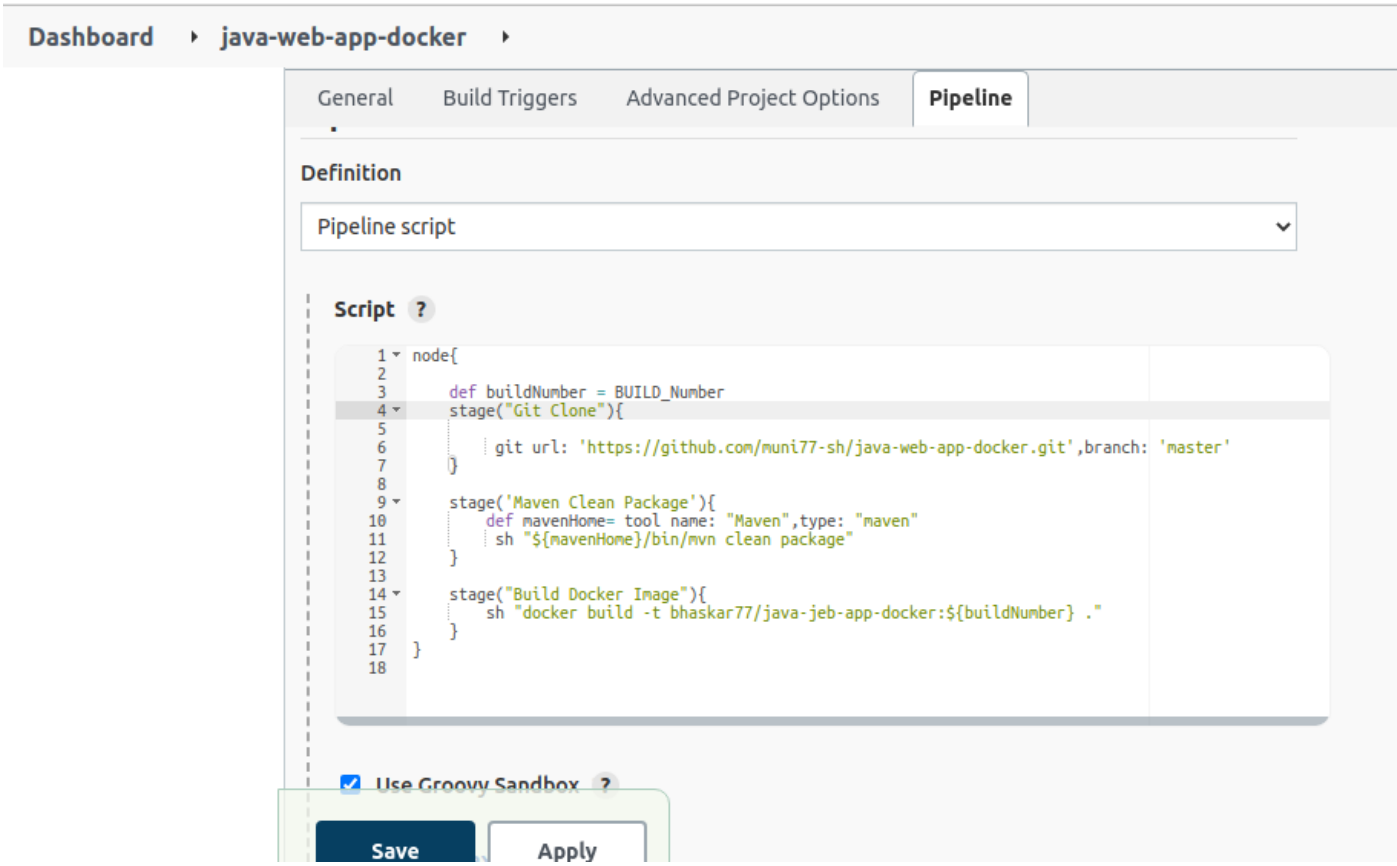
Save it → Trigger the build → Build is Success



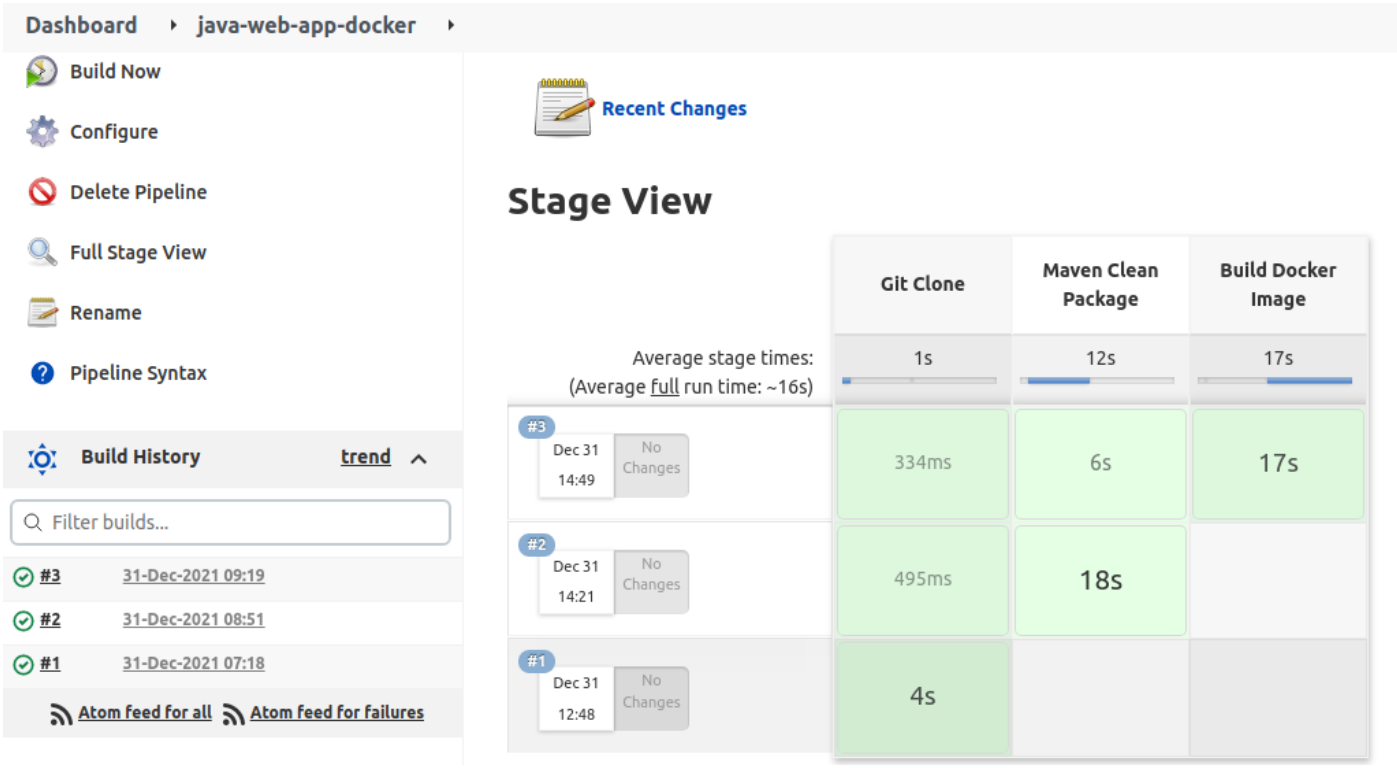
Go to build job to verify to build is success → Click on “Console”



Step 3: Create a Docker Image: (in Pipeline Job)



→ save → Build is Success



❖ Check the images are created in jenkins server:

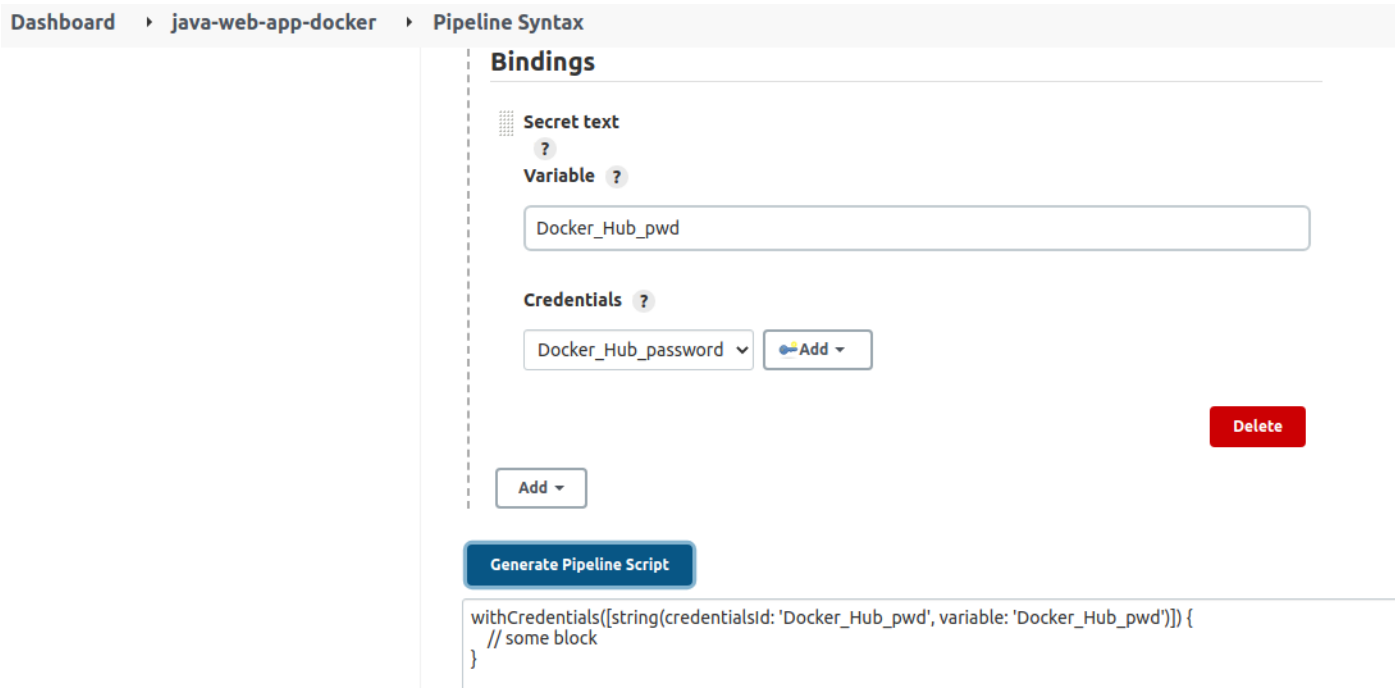
```
ubuntu@ip-172-31-2-207:~$ sudo docker images
REPOSITORY                        TAG           IMAGE ID       CREATED          SIZE
bhaskar77/java-jeb-app-docker     3             b2a3d3ef0640   6 minutes ago    499MB
tomcat                            8.0.20-jre8   e88a065848be   6 years ago      493MB
ubuntu@ip-172-31-2-207:~$
```

Step4: **To Login In Docker Hub And Push :**

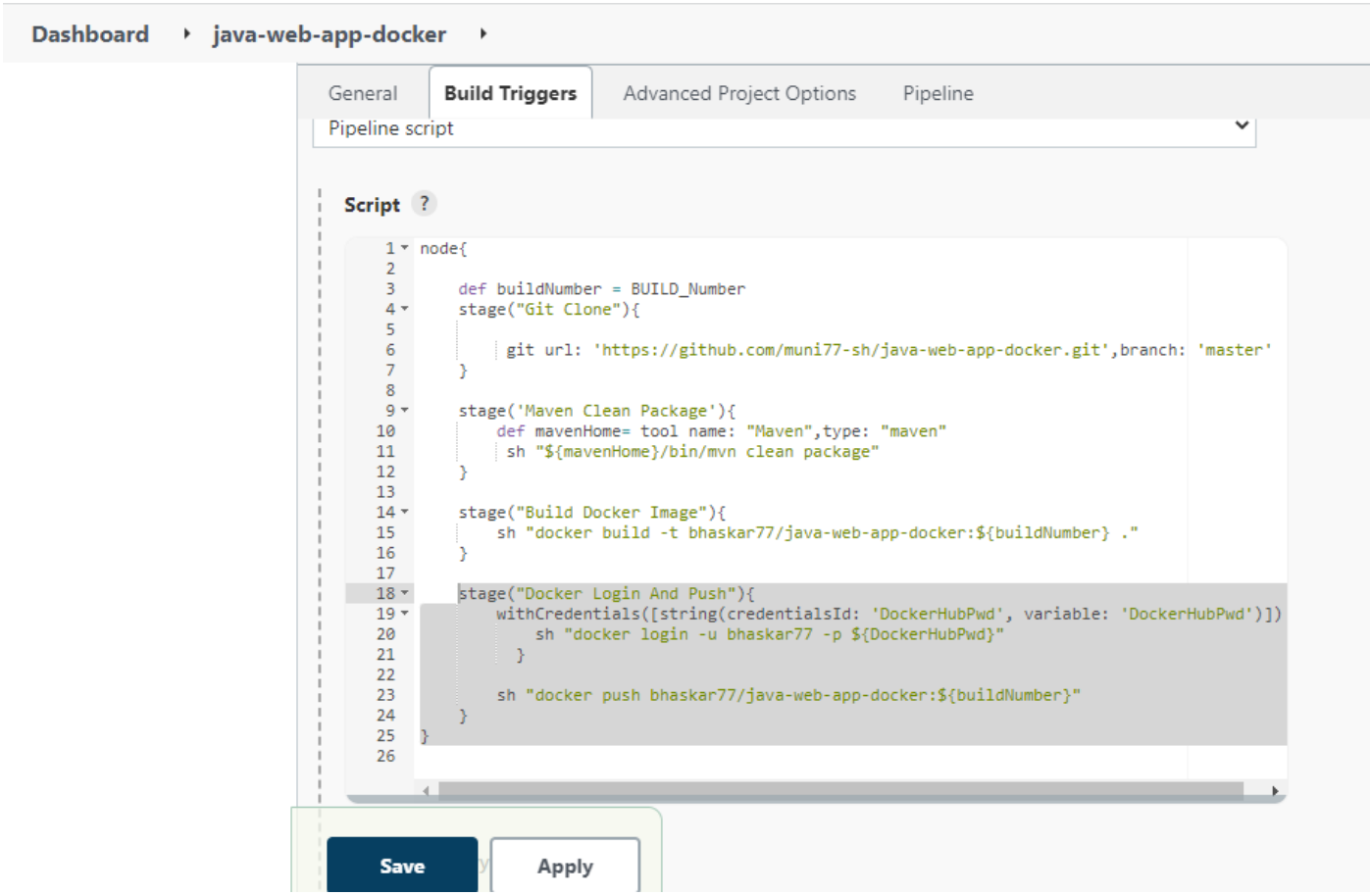
Click on, Pipeline Syntax → To create Credentials (Secret) → Sample Step →

“withCredentials: Bind credentials to variables” → Create Secret Text → then,

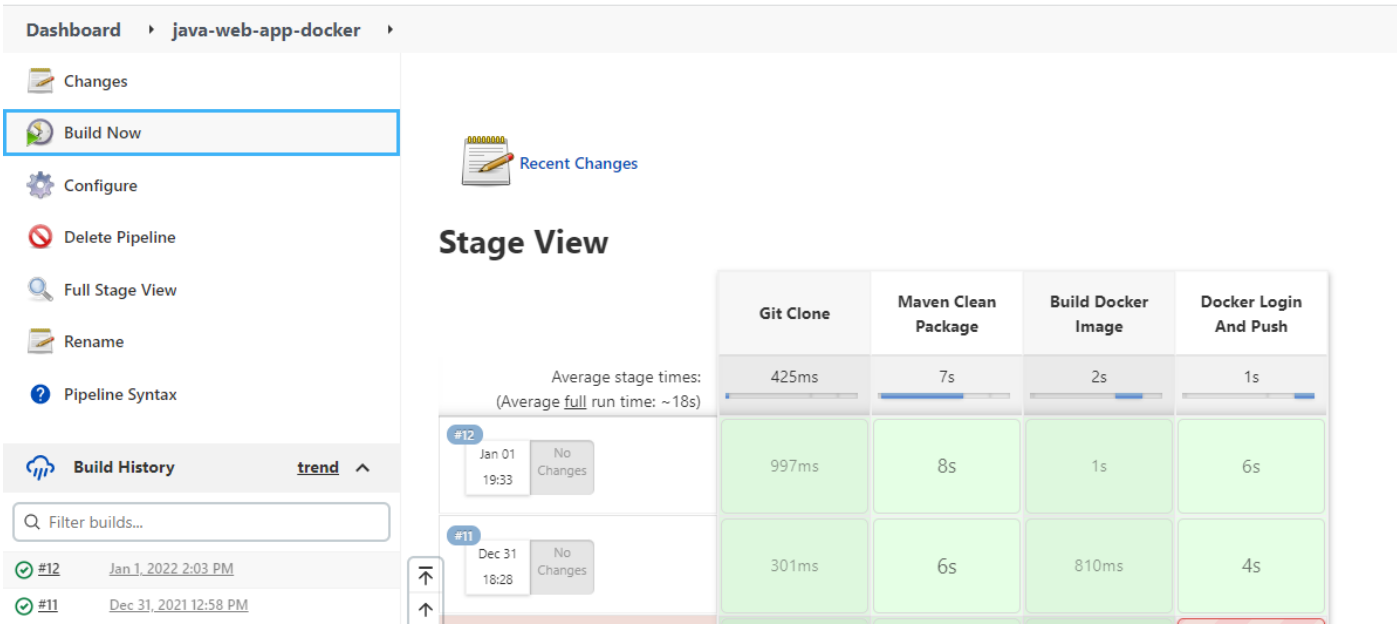
“Generate pipeline script”



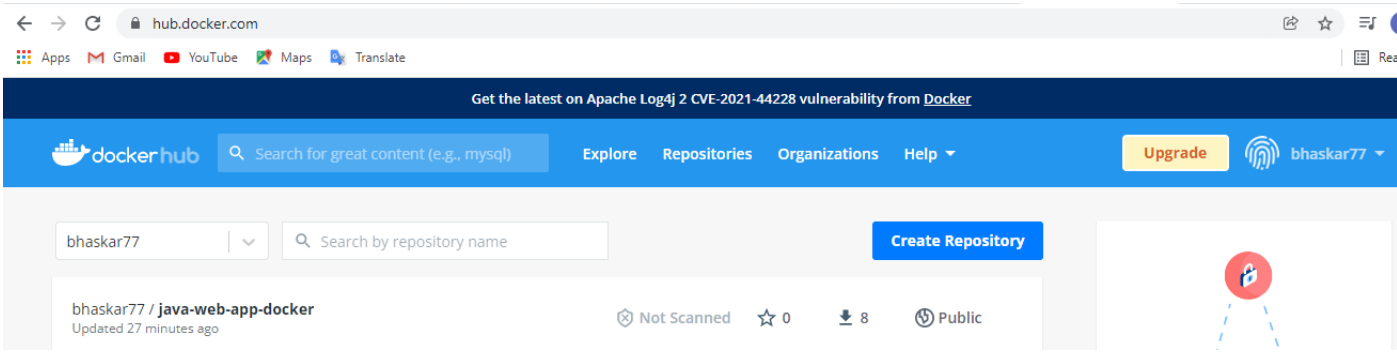
Again, goto “pipeline job” → Click on “Configure” →



→ save it → trigger the build → build is success



→ check images have been pushed into Docker Hub.



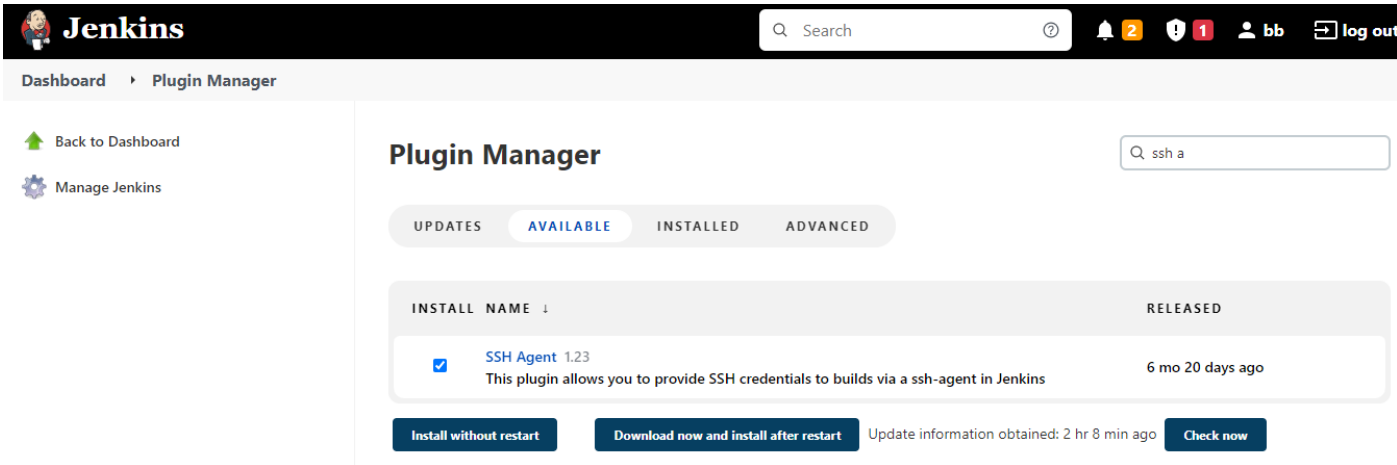
→ check if CLI docker images are created.

```
ubuntu@ip-172-31-2-207:~$ sudo docker images
REPOSITORY                    TAG          IMAGE ID       CREATED        SIZE
bhaskar77/java-web-app-docker 12           c09794d40cdf   2 hours ago    499MB
tomcat                        8.0.20-jre8  e88a065848be   6 years ago    493MB
ubuntu@ip-172-31-2-207:~$
```

Step5: To Connect to Deployment server From Jenkin_Server

→ Jenkins Dashboard → Manage Jenkins → Manage Plugin →

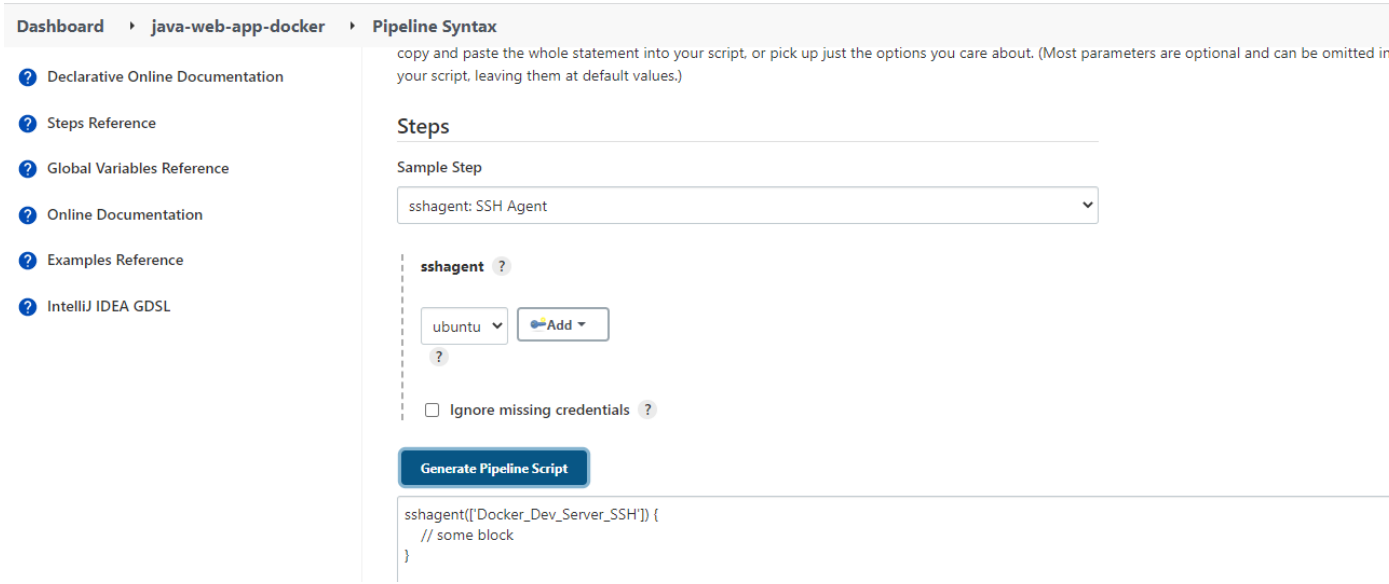
Click on “Available” → type “SSH Agent” install without restart.



Click on, Pipeline Syntax → Sample Step →

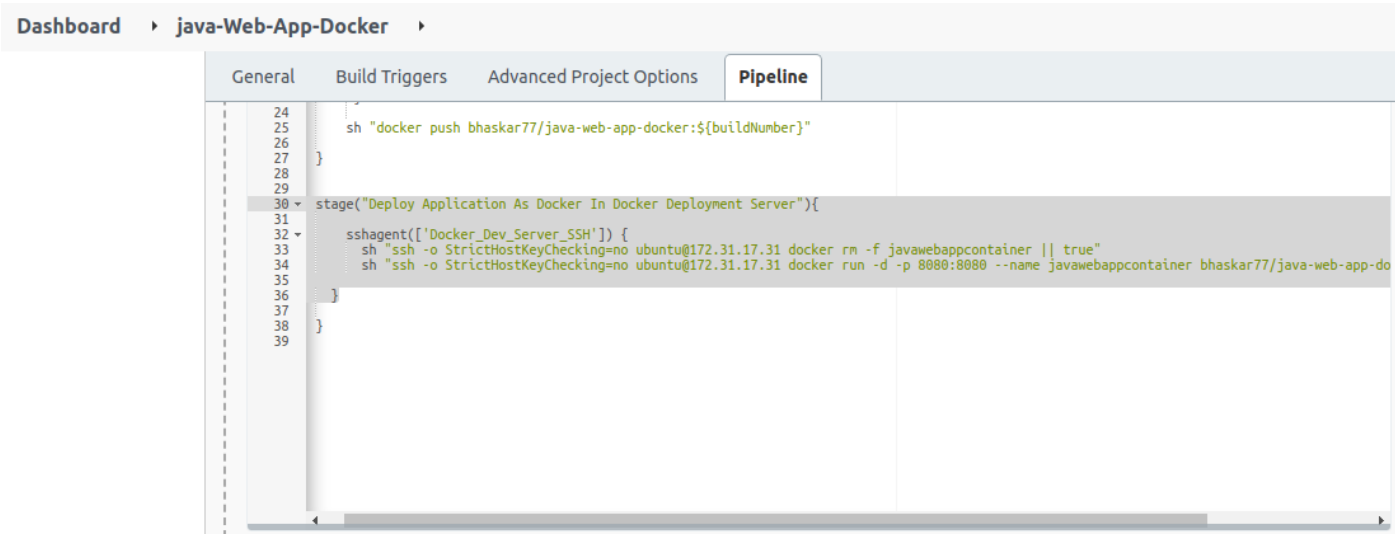
“sshagent:SSH Agent” → To connect from One to another server → then,

“Generate pipeline script”



→ Go to back pipeline job → Click on “Configure” write the script for → Deploy Applic

Docker Deployment Server:



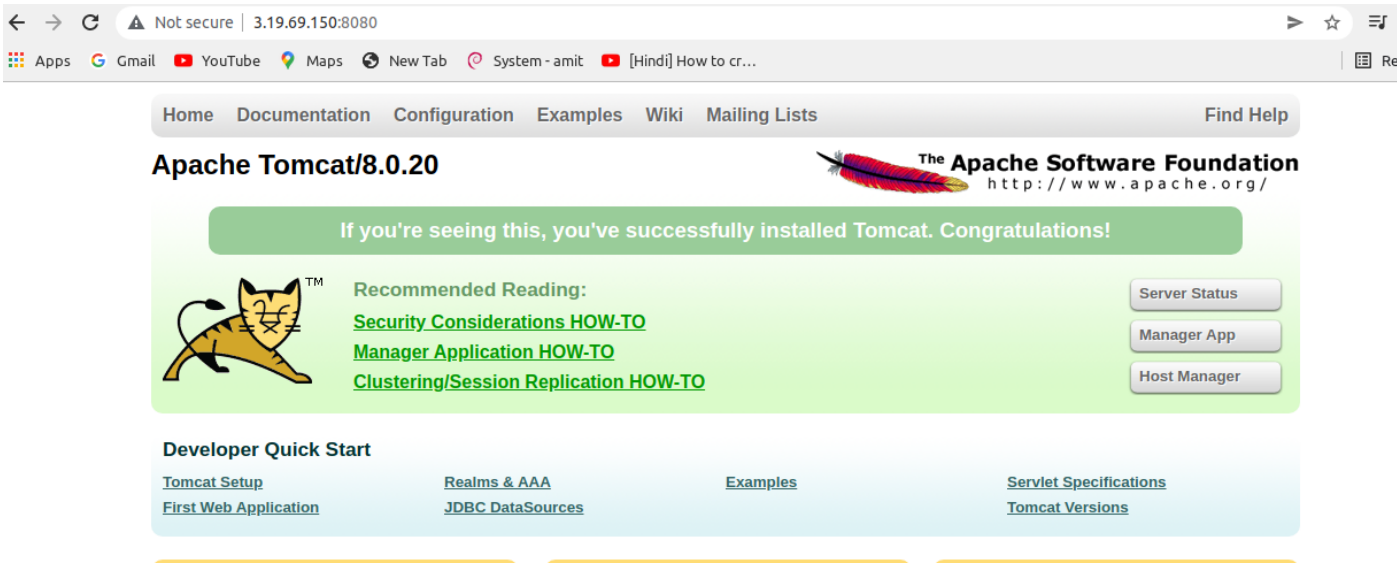
```
stage("Deploy Application As Docker In Docker Deployment Server"){

    sshagent(['Docker_Dev2_Server_SSH']) {
      sh "ssh -o StrictHostKeyChecking=no ubuntu@172.31.2.250 docker rm -f javawebappcontainer || true"
      sh "ssh  -o StrictHostKeyChecking=no ubuntu@172.31.2.250 docker run -d -p 8080:8080 --name javawebappcontainer bhaskar77/java-web-app-docker:${buildNumber}"

  }
}
```

→ we are trying access the Application→ from " Deployment-server-docker"

With public-IP "http://3.19.69.150:8080/"



To check my application context is "http://3.19.69.150:8080/java-web-app/"

Again, make changes in code and again rebuild the your “jenkins job”