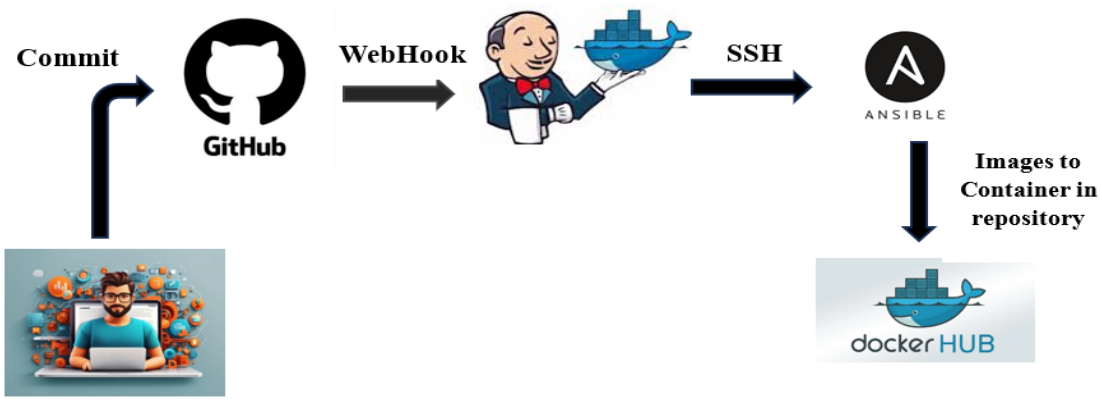
Automated Image Build and Deployment Pipeline to Docker Hub via Git, Jenkins & Ansible

by

PL. Nagarathinam

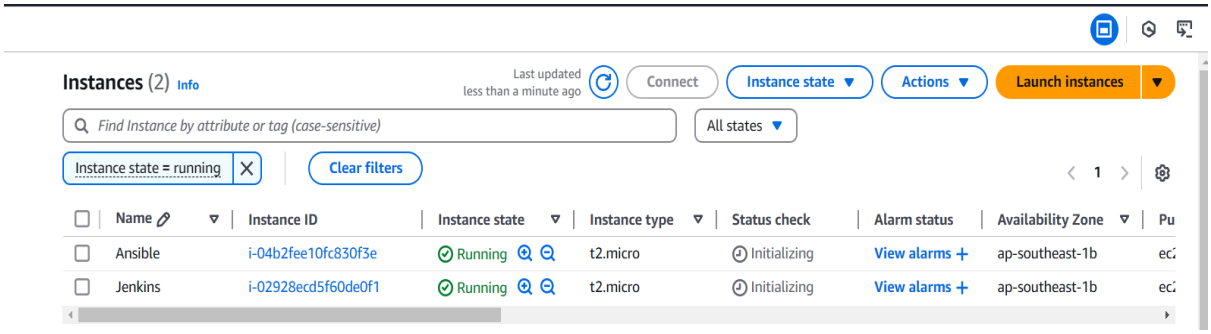
Workflow:



“This project incorporates various components to streamline the building and deployment of a Docker image to Container. It starts with the developer creating a Dockerfile and pushing it to GitHub. Jenkins triggers an automated pipeline upon receiving commits and connects to an Ansible server via SSH to build the image based on the Dockerfile.”

Step 1:

Launch Two Instances and Name it as Jenkins and Ansible



Step 2:

In Jenkins Instance install Java and Jenkins Tools

In Ansible Instance install Ansible and Docker Tools

```
ubuntu@ip-172-31-16-110:~$ sudo apt install fontconfig openjdk-17-jre
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  adwaita-icon-theme alsa-topology-conf alsa-ucm-conf at-spi2-common at-spi2-core ca-certificates-java dconf-gsettings-backend dconf-service fontconfig-config
  fonts-dejavu-core fonts-dejavu-extra fonts-dejavu-mono gsettings-desktop-schemas gtk-update-icon-cache hicolor-icon-theme humanity-icon-theme java-common
  libasound2-data libasound2t64 libatk-bridge2.0-0t64 libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0t64 libatspi2.0-0t64 libavahi-client3
  libavahi-common-data libavahi-common3 libcairo-gobject2 libcairo2 libcups2t64 libdatrie1 libdconf1 libdeflate0 libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2
  libdrm-radeon1 libfontconfig1 libgail-common libgail18t64 libgdk-pixbuf-2.0-0 libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-common libgif7 libgl1 libgl1-amber-dri
  libgl1-mesa-dri libglapi-mesa libglvnd0 libglx-mesa0 libglx0 libgraphite2-3 libgtk2.0-0t64 libgtk2.0-bin libgtk2.0-common libharfbuzz0b libice6 libjbig0
  libjpeg-turbo8 libjpeg8 liblcms2-2 liblerc4 libllvm17t64 libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0 libpciaccess0 libpcsclite1 libpixman-1-0
  librsvg2-2 librsvg2-common libsharpyuv0 libsm6 libthai-data libthai0 libtiff6 libvulkan1 libwayland-client0 libwebp7 libx11-xcb1 libxaw7 libxcb-dri2-0
  libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-render0 libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libxcomposite1 libxcursor1
  libxdamage1 libxfixes3 libxft2 libxi6 libxinerama1 libxkbfile1 libxmu6 libxpm4 libxrandr2 libxrender1 libxshmfence1 libxt6t64 libxtst6 libxv1 libxxf86dga1
  libxxf86vm1 mesa-vulkan-drivers openjdk-17-jre-headless session-migration ubuntu-mono x11-common x11-utils
Suggested packages:
```

i-01b6b4b3e107104fd (Jenkins)

PublicIPs: 34.221.166.111 PrivateIPs: 172.31.16.110

```
ubuntu@ip-172-31-16-110:~$ java --version
openjdk 17.0.13 2024-10-15
OpenJDK Runtime Environment (build 17.0.13+11-Ubuntu-2ubuntu124.04)
OpenJDK 64-Bit Server VM (build 17.0.13+11-Ubuntu-2ubuntu124.04, mixed mode, sharing)
ubuntu@ip-172-31-16-110:~$
```

```
ubuntu@ip-172-31-16-110:~$ sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package jenkins is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

E: Package 'jenkins' has no installation candidate
ubuntu@ip-172-31-16-110:~$ sudo apt update
```

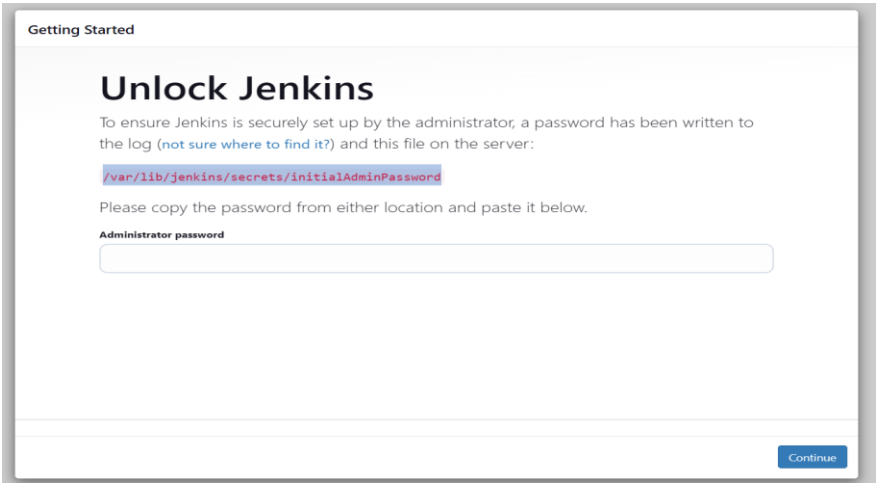
```
ubuntu@ip-172-31-16-110:~$ sudo systemctl start jenkins
ubuntu@ip-172-31-16-110:~$ sudo systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins
ubuntu@ip-172-31-16-110:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
     Active: active (running) since Wed 2024-11-20 09:38:48 UTC; 1min 59s ago
   Main PID: 4307 (java)
      Tasks: 38 (limit: 1130)
     Memory: 299.4M (peak: 355.3M)
        CPU: 16.562s
     CGroup: /system.slice/jenkins.service
             └─4307 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war

Nov 20 09:38:40 ip-172-31-16-110 jenkins[4307]: 57470e009a684627918a041ed44d4fd3
Nov 20 09:38:40 ip-172-31-16-110 jenkins[4307]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Nov 20 09:38:40 ip-172-31-16-110 jenkins[4307]: ************************************************************
Nov 20 09:38:40 ip-172-31-16-110 jenkins[4307]: ************************************************************
Nov 20 09:38:40 ip-172-31-16-110 jenkins[4307]: ************************************************************
Nov 20 09:38:48 ip-172-31-16-110 jenkins[4307]: 2024-11-20 09:38:48.559+0000 [id=30]        INFO        jenkins.InitReactorR
Nov 20 09:38:48 ip-172-31-16-110 jenkins[4307]: 2024-11-20 09:38:48.587+0000 [id=23]        INFO        hudson.lifecycle.Li
Nov 20 09:38:48 ip-172-31-16-110 systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Nov 20 09:38:49 ip-172-31-16-110 jenkins[4307]: 2024-11-20 09:38:49.459+0000 [id=46]        INFO        h.m.DownloadServices
Nov 20 09:38:49 ip-172-31-16-110 jenkins[4307]: 2024-11-20 09:38:49.460+0000 [id=46]        INFO        hudson.util.Retrier
lines 1-20/20 (END)
```

i-01b6b4b3e107104fd (Jenkins)

PublicIPs: 34.221.166.111 PrivateIPs: 172.31.16.110

After install Jenkins in the Instance copy the IP Address of the instance and paste to web browser with port number **8080**, Then the Jenkins web page will be open Copy that path from the page and paste to instance console then click enter, we got one password for the Jenkins tool to open.




```
Nov 20 09:38:48 ip-172-31-16-110 systemd[1]: Started jenkins.service - Jenkins Continuous Integ
Nov 20 09:38:49 ip-172-31-16-110 jenkins[4307]: 2024-11-20 09:38:49.459+0000 [id=46]        INF
Nov 20 09:38:49 ip-172-31-16-110 jenkins[4307]: 2024-11-20 09:38:49.460+0000 [id=46]        INF
lines 1-20/20 (END)
[1]+  Stopped                 sudo systemctl status jenkins
ubuntu@ip-172-31-16-110:~$ sudo su
root@ip-172-31-16-110:/home/ubuntu# cat /var/lib/jenkins/secrets/initialAdminPassword
lines 1-20/20 (END)
[1]+  Stopped                 sudo systemctl status jenkins
ubuntu@ip-172-31-16-110:~$ sudo su
root@ip-172-31-16-110:/home/ubuntu# cat /var/lib/jenkins/secrets/initialAdminPassword
57470e009a684627918a041ed44d4fd3
root@ip-172-31-16-110:/home/ubuntu#
```

```
ubuntu@ip-172-31-16-219:~$ sudo apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ansible-core python3-jmespath python3-kerberos python3-nacl python3-ntlm-auth python3-packaging python3-paramiko p
  python3-winrm python3-xmltodict sshpass
Suggested packages:
  python-nacl-doc python3-gssapi python3-invoke
The following NEW packages will be installed:
  ansible ansible-core python3-jmespath python3-kerberos python3-nacl python3-ntlm-auth python3-packaging python3-pa
  python3-resolvelib python3-winrm python3-xmltodict sshpass
0 upgraded, 13 newly installed, 0 to remove and 57 not upgraded.
Need to get 19.0 MB of archives.
After this operation, 208 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-16-219:~$ ansible --version
ansible [core 2.17.6]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/ubuntu/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.12.3 (main, Sep 11 2024, 14:17:37) [GCC 13.2.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
ubuntu@ip-172-31-16-219:~$
```

i-0d885328ba81da5aa (Ansible)

PublicIPs: 35.91.221.232 PrivateIPs: 172.31.16.219

```
ubuntu@ip-172-31-16-219:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compo
The following NEW packages will be installed:
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-16-219:~$ docker --version
Docker version 24.0.7, build 24.0.7-0ubuntu4.1
ubuntu@ip-172-31-16-219:~$
```

```
root@ip-172-31-1-55:/home/ubuntu# sudo apt install -y apt-transport-https ca-certificates curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
The following additional packages will be installed:
  libcurl3t64-gnutls libcurl4t64
The following NEW packages will be installed:
  apt-transport-https
The following packages will be upgraded:
  curl libcurl3t64-gnutls libcurl4t64
3 upgraded, 1 newly installed, 0 to remove and 54 not upgraded.
Need to get 904 kB of archives.
After this operation, 35.8 kB of additional disk space will be used.
Get:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 apt-transport-https all 2.7.14build2
Get:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 curl amd64 8.5.0-2ubuntu10.5 [22
Get:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libcurl4t64 amd64 8.5.0-2ubuntu1
Get:4 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libcurl3t64-gnutls amd64 8.5.0-2
Fetched 904 kB in 0s (21.5 MB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 68203 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.7.14build2_all.deb ...
Unpacking apt-transport-https (2.7.14build2) ...
Preparing to unpack .../curl_8.5.0-2ubuntu10.5_amd64.deb ...
Unpacking curl (8.5.0-2ubuntu10.5) over (8.5.0-2ubuntu10.4) ...
```

Step 3:

Next Open VS Code Tool to Push the Docker File into Git Repository

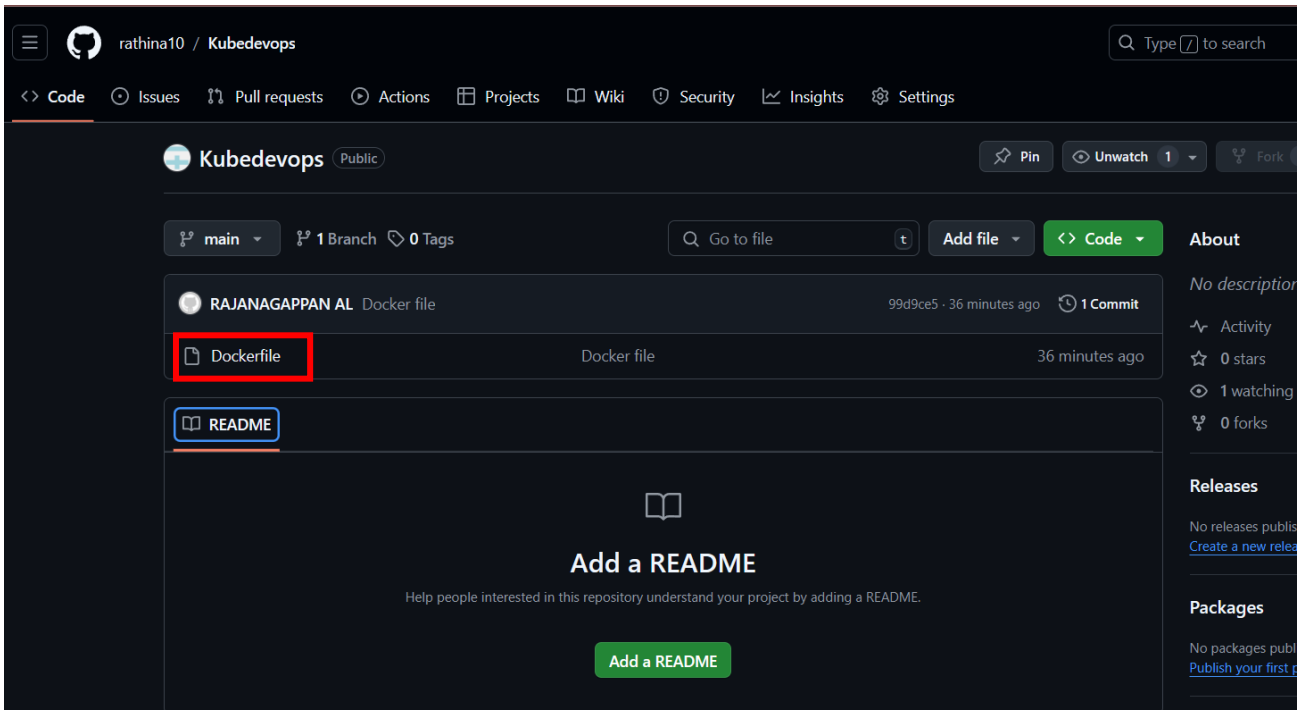
Dockerfile > ...

```
1   # Use AlmaLinux 8 as the base image
2   FROM almalinux:8
3
4   # Install Apache (httpd) web server
5   RUN yum install -y httpd && yum clean all
6
7   # Create a directory for serving the image
8   # Use the standard Apache web directory instead of /home/ubuntu/
9   RUN mkdir -p /home/ubuntu
10
11  # Expose port 80 to allow web access
12  EXPOSE 80
13
14  # Start Apache in the foreground
15  CMD ["httpd", "-D", "FOREGROUND"]
16
17
18
```

In the Terminal enter the Git commands

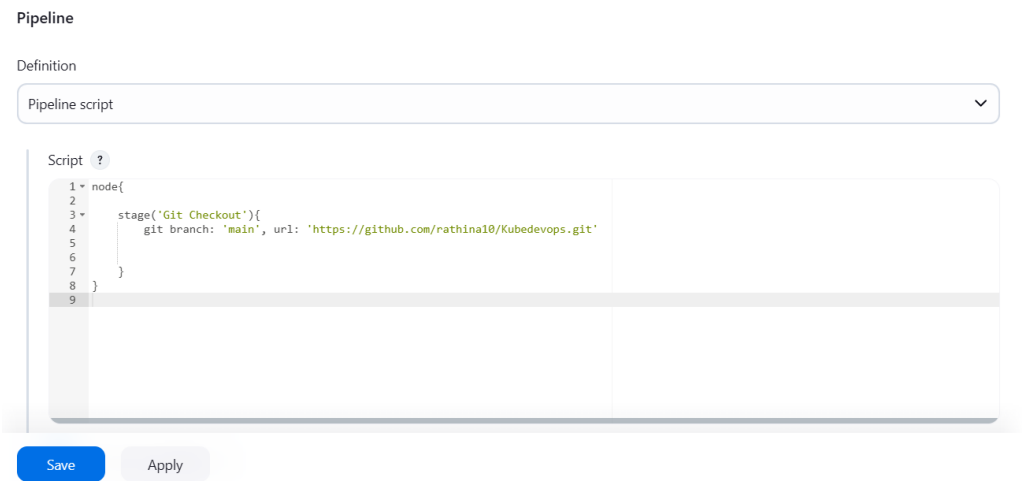
```
PS C:\Users\rajan\OneDrive\Notes\New Devops\Kubedevops> git init
Reinitialized existing Git repository in C:/Users/rajan/OneDrive/Notes/New Devops/Kubedevops/.git/
PS C:\Users\rajan\OneDrive\Notes\New Devops\Kubedevops> git add .
PS C:\Users\rajan\OneDrive\Notes\New Devops\Kubedevops> git commit -m "Docker images"
```

```
PS C:\Users\rajan\OneDrive\Notes\New Devops\Kubedevops> git commit -m "Docker file"
[main (root-commit) 99d9ce5] Docker file
 1 file changed, 12 insertions(+)
 create mode 100644 Dockerfile
PS C:\Users\rajan\OneDrive\Notes\New Devops\Kubedevops> git remote add origin https://github.com/rathina10/Kubedevops.git
error: remote origin already exists.
PS C:\Users\rajan\OneDrive\Notes\New Devops\Kubedevops> git push origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 459 bytes | 459.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/rathina10/Kubedevops.git
 * [new branch]      main -> main
```

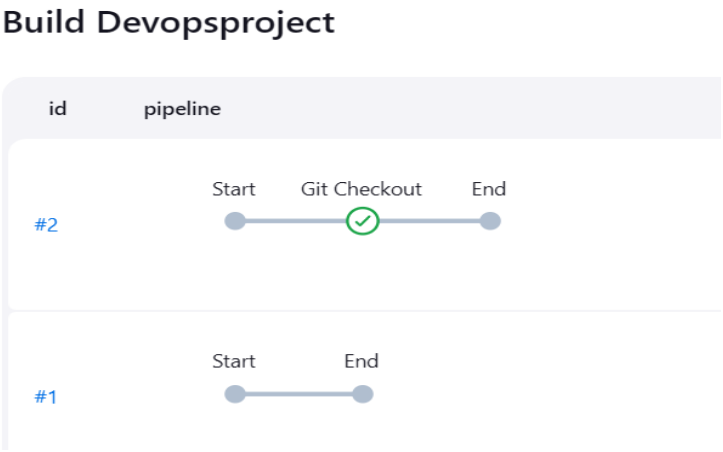


Step 4:

Next Go to Jenkins > Built Job > Pipeline style > Script, Text the scripting line to use the GitHub. So, Paste the URL to access the GitHub Repository.

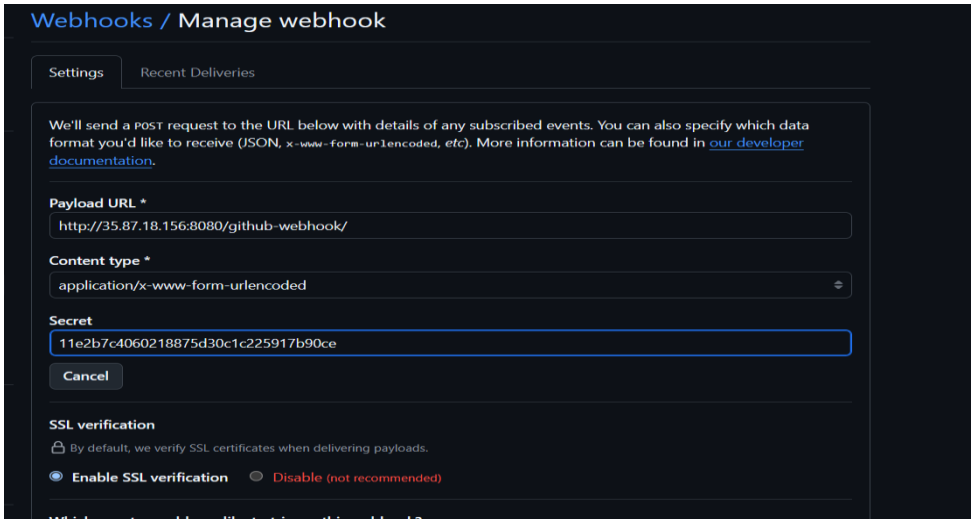


Then Click Apply and Save to Build the Script

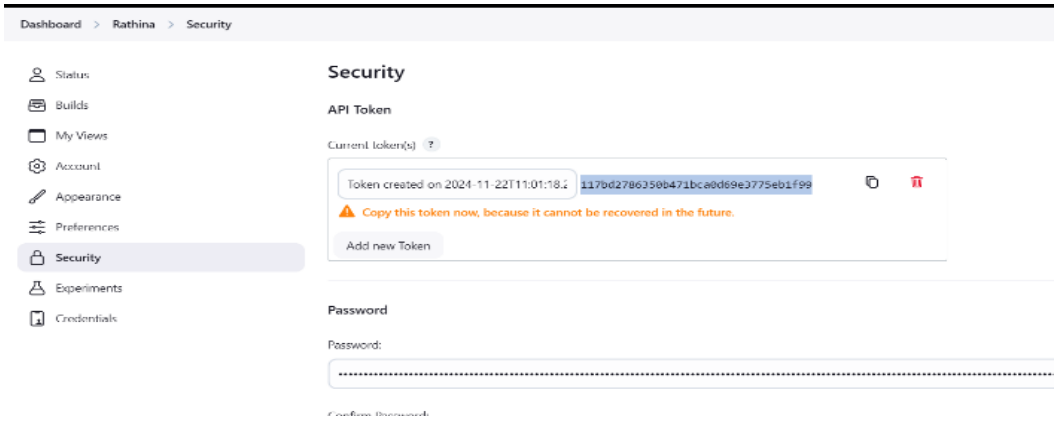


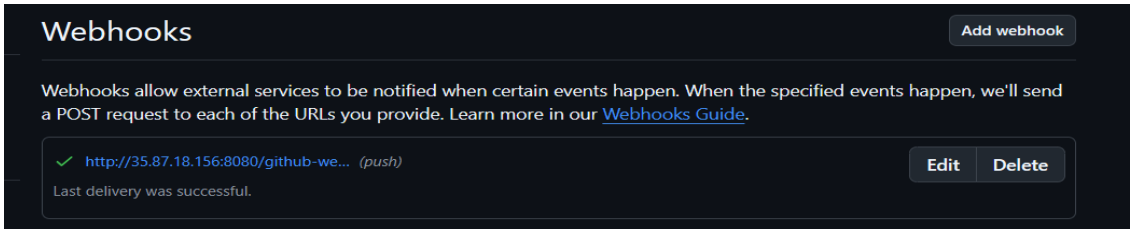
Step 5:

This Manual Build process is not easy to a very big scripts, So for we can make it automatic build and run process using GitHub's Webhook



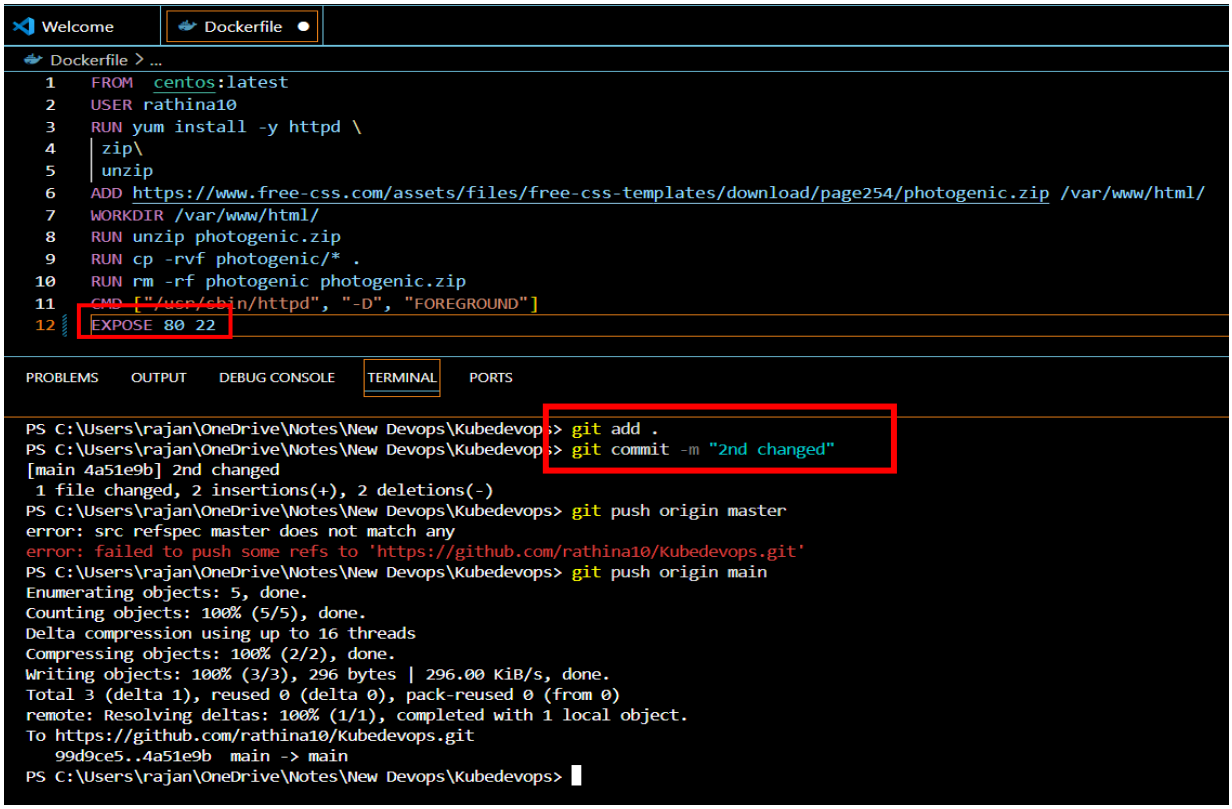
Go to GitHub>Setting>Webhook and enter the URL of the Jenkins IP Address and secret token from the Jenkins.

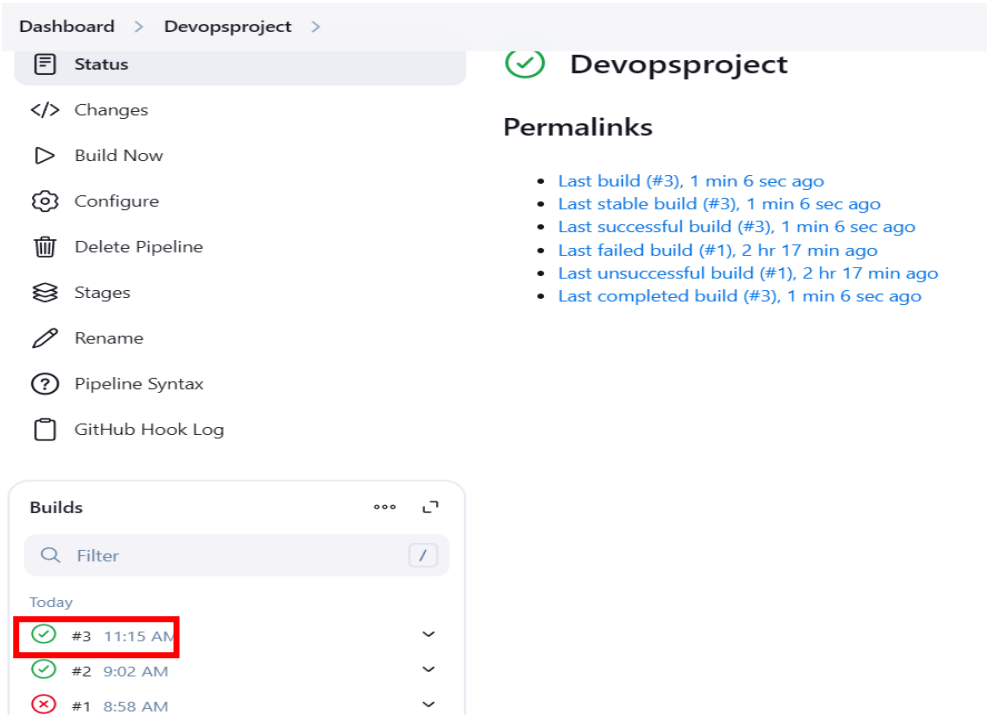




Step 6:

Then give some change in the Docker file and commit to the GitHub repository it will automatically change without build the script Jenkins





Definition

Pipeline script

Script ?

```
node{

    stage('Git Checkout'){
        git branch: 'main', url: 'https://github.com/rathina10/Kubedevops.git'

    }
    stage('Sending Docker to Ansible Server over the SSH'){
        sshagent(['ansible_access']){
        sh 'ssh -o StrictHostKeyChecking=no ubuntu 172.31.16.219'
        sh'scp /var/lib/jenkins/workspace/Devopsproject/* ubuntu 172.31.16.219:/home/ubuntu '
}
}
    }
}
```

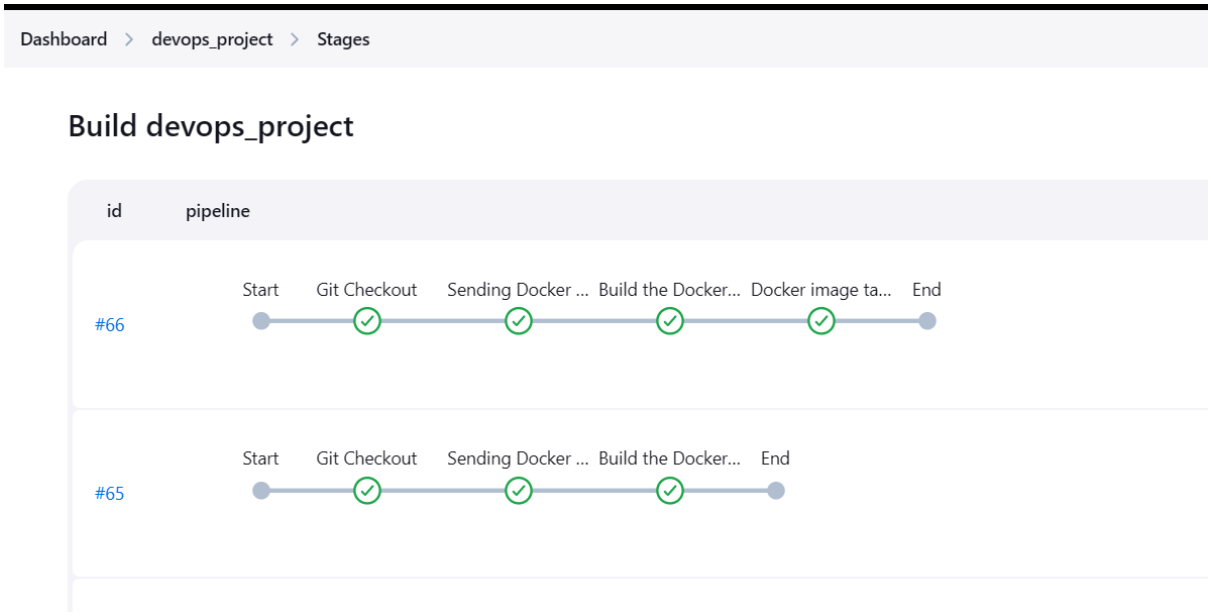
Use Groovy Sandbox ?

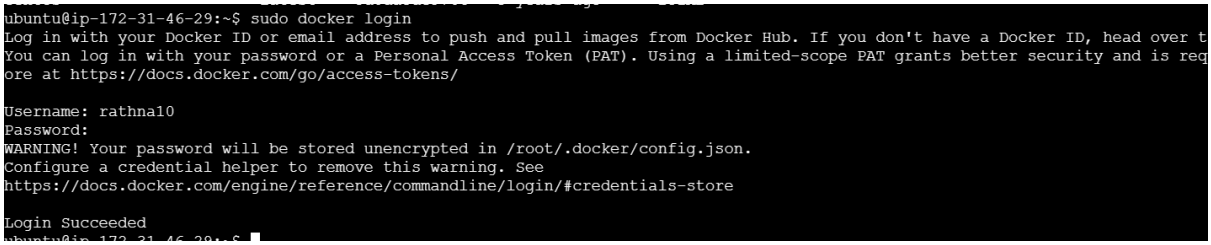
Pipeline Syntax

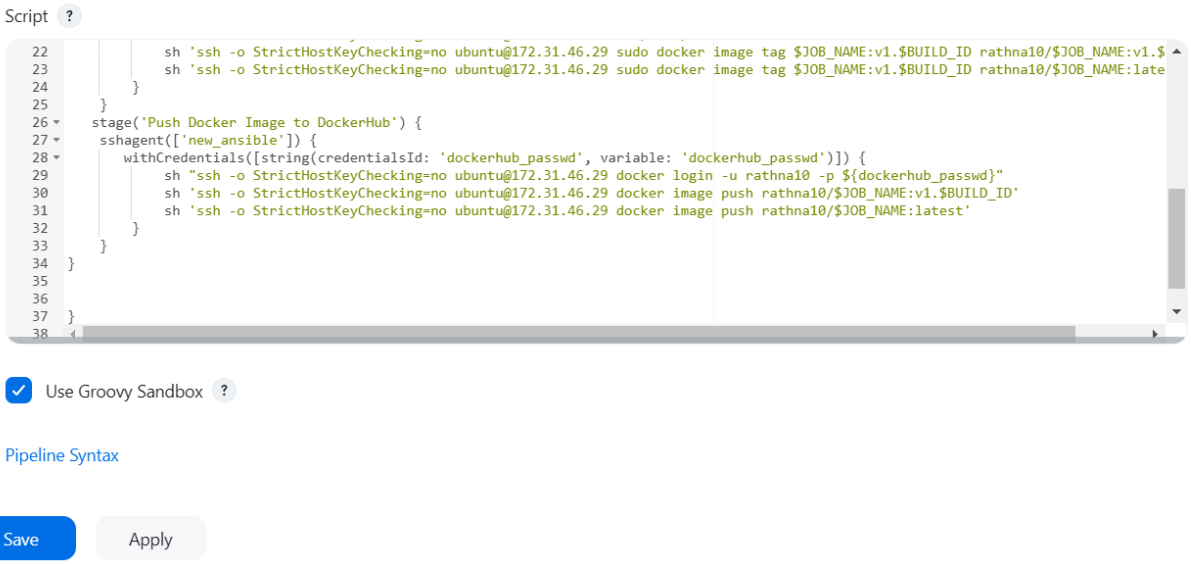
Save Apply

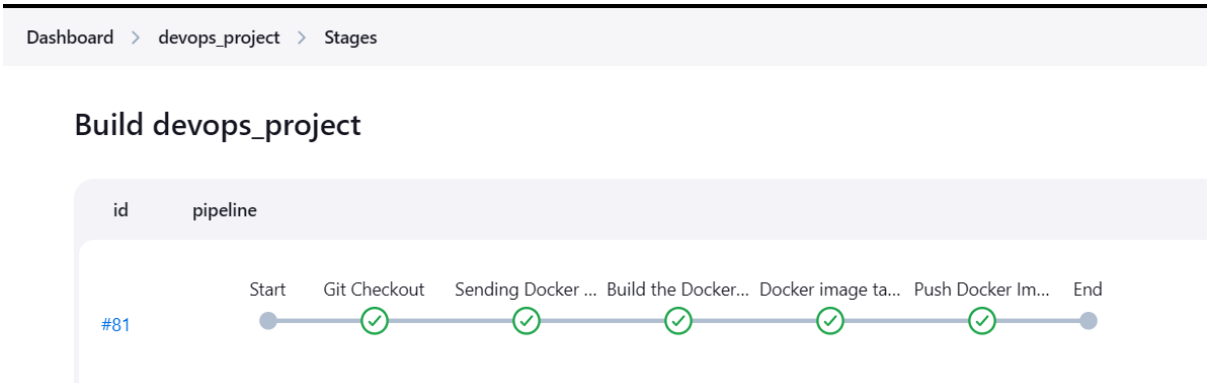
```
}
stage("Build the Docker image") {
    sshagent(['new_ansible']){
      sh 'ssh -o StrictHostKeyChecking=no ubuntu@172.31.46.29 cd /home/ubuntu/'
      sh 'ssh -o StrictHostKeyChecking=no ubuntu@172.31.46.29 docker image build -t $JOB_NAME:v1.$BUILD_ID .'
    }
}
stage('Docker image tagging') {
sshagent(['new_ansible']) {
    sh 'ssh -o StrictHostKeyChecking=no ubuntu@172.31.46.29 cd /home/ubuntu'
    sh 'ssh -o StrictHostKeyChecking=no ubuntu@172.31.46.29 sudo docker image tag $JOB_NAME:v1.$BUILD_ID rathina10/$JOB_NAME:v1.$BU
    sh 'ssh -o StrictHostKeyChecking=no ubuntu@172.31.46.29 sudo docker image tag $JOB_NAME:v1.$BUILD_ID rathina10/$JOB_NAME:latest
}
```

```
ubuntu@ip-172-31-46-29:~$ sudo docker images
REPOSITORY       TAG       IMAGE ID       CREATED         SIZE
devops_project   v1.65     ff7f572d5a54   2 minutes ago   210MB
<none>           <none>    bbf241f545ed   5 hours ago     211MB
<none>           <none>    6a6125ede348   5 hours ago     211MB
<none>           <none>    5a70b814bb26   5 hours ago     211MB
<none>           <none>    412a4d45d59a   5 hours ago     211MB
<none>           <none>    b37e76c665d6   5 hours ago     211MB
<none>           <none>    78e66d2cc2dd   2 days ago      231MB
almalinux        8         f1218159f16a   2 months ago    190MB
centos           latest    5d0da3dc9764   3 years ago     231MB
ubuntu@ip-172-31-46-29:~$ sudo docker images
REPOSITORY                   TAG       IMAGE ID       CREATED         SIZE
devops_project               v1.65     ff7f572d5a54   7 minutes ago   210MB
devops_project               v1.66     ff7f572d5a54   7 minutes ago   210MB
rathina10/devops_project     latest    ff7f572d5a54   7 minutes ago   210MB
rathina10/devops_project     v1.66     ff7f572d5a54   7 minutes ago   210MB
<none>                       <none>    bbf241f545ed   5 hours ago     211MB
<none>                       <none>    6a6125ede348   5 hours ago     211MB
<none>                       <none>    5a70b814bb26   5 hours ago     211MB
<none>                       <none>    412a4d45d59a   5 hours ago     211MB
<none>                       <none>    b37e76c665d6   5 hours ago     211MB
<none>                       <none>    78e66d2cc2dd   2 days ago      231MB
almalinux                    8         f1218159f16a   2 months ago    190MB
centos                       latest    5d0da3dc9764   3 years ago     231MB
ubuntu@ip-172-31-46-29:~$
```



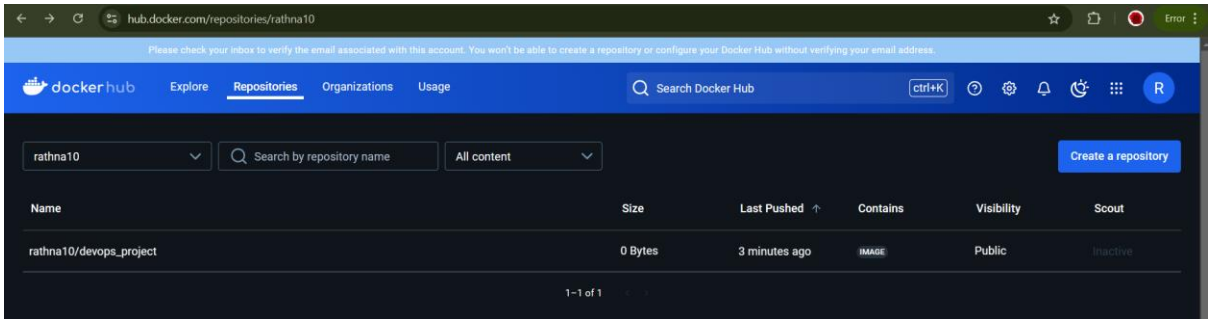


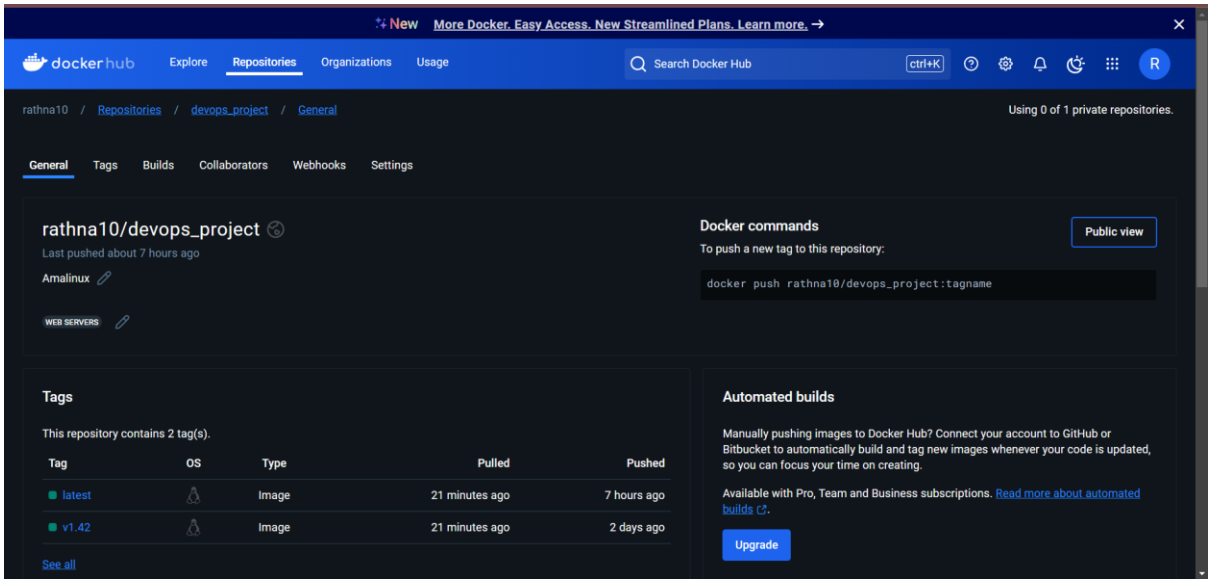


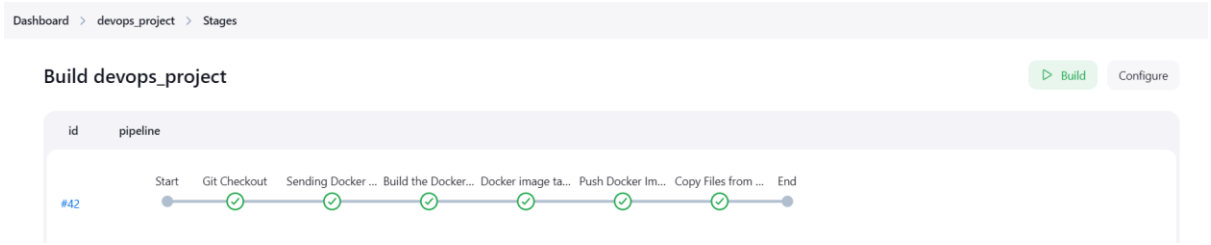


Step:7

Now, All steps are done and go to Docker hub and the image will appear successfully. Now any one can pull this image and use in their local devices







Final Step:

Now want to test the image, whether it is running or not

1)Launch the new instance and install the docker

2) Then, pull the rathna10/devop_project:lastest copy from Dockerhub

3)It is pulling, check using the command sudo docker images

4) To run the images using the command sudo docker run -d --name almalinux_container -p 8080:80 rathna10/devops_project:latest

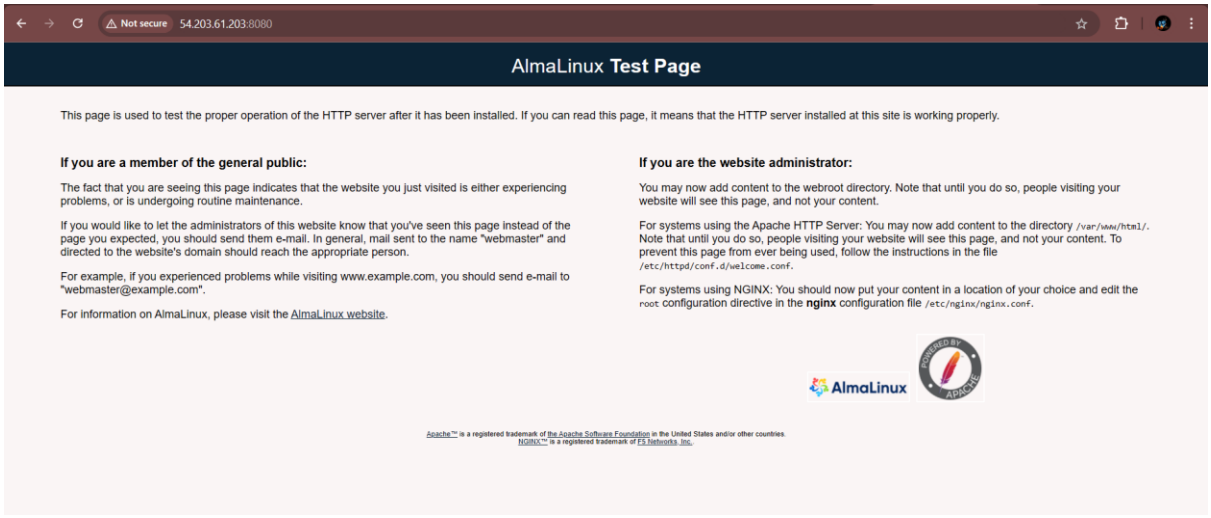
5) Press Enter button, you get the random alpha number

6) Then, copy the public IP Address of the instance and paste the web browser with port number (:8080), successfully we get the image of the almalinux page

```
ubuntu@ip-172-31-24-68:~$ sudo docker pull rathna10/devops_project:latest
latest: Pulling from rathna10/devops_project
2609da11fd88: Pull complete
165da094b92d: Pull complete
4789af970148: Pull complete
c37966faaa3c: Pull complete
a1a182728d93: Pull complete
cc4d991c7e7f: Pull complete
Digest: sha256:968d7554c900437a479f4fe7503fa403bc846e9e5b0ea6ab383ac626d4c9eeb3
Status: Downloaded newer image for rathna10/devops_project:latest
docker.io/rathna10/devops_project:latest
ubuntu@ip-172-31-24-68:~$ sudo docker run -d --name almalinux_container -p 8080:80 rathna10/devops_project:latest
f939dcc000bc47ee2e6cbe854901895792c1fd34962758451dd3850674d5a9bd
ubuntu@ip-172-31-24-68:~$
```

i-022c9e6174b7b7052 (Testing)

PublicIPs: 54.203.61.203 PrivateIPs: 172.31.24.68



Pipeline Script:

```
node{
   stage('Git Checkout'){
      git branch: 'main', url: 'https://github.com/rathina10/realtime_project.git'
   }
  stage('Sending Docker to Ansible Server over the SSH'){
      sshagent(['ansi_ssh']){
        sh 'ssh -o StrictHostKeyChecking=no ubuntu@172.31.31.34'
```

```
        sh 'scp /var/lib/jenkins/workspace/devops_project/* ubuntu@172.31.31.34:/home/ubuntu/'
      }
    }
      stage("Build the Docker image") {
      sshagent(['ansi_ssh']){
        sh 'ssh -o StrictHostKeyChecking=no ubuntu@172.31.31.34 cd /home/ubuntu/'
        sh 'ssh -o StrictHostKeyChecking=no ubuntu@172.31.31.34 docker image build -t $JOB_NAME:v1.$BUILD_ID .'
      }
    }
    stage('Docker image tagging') {
      sshagent(['ansi_ssh']) {
         sh 'ssh -o StrictHostKeyChecking=no ubuntu@172.31.31.34 cd /home/ubuntu/'
         sh 'ssh -o StrictHostKeyChecking=no ubuntu@172.31.31.34 sudo docker image tag $JOB_NAME:v1.$BUILD_ID rathna10/$JOB_NAME:v1.$BUILD_ID'
         sh 'ssh -o StrictHostKeyChecking=no ubuntu@172.31.31.34 sudo docker image tag $JOB_NAME:v1.$BUILD_ID rathna10/$JOB_NAME:latest'
      }
    }
   stage('Push Docker Image to DockerHub') {
     sshagent(['ansi_ssh']) {
       withCredentials([string(credentialsId: 'dhub_password', variable: 'dhub_password')]) {
          sh "ssh -o StrictHostKeyChecking=no ubuntu@172.31.31.34 docker login -u rathna10 -p ${dhub_password}"
          sh 'ssh -o StrictHostKeyChecking=no ubuntu@172.31.31.34 docker image push rathna10/$JOB_NAME:v1.$BUILD_ID'
```

```
            sh 'ssh -o StrictHostKeyChecking=no ubuntu@172.31.31.34 docker image push rathna10/$JOB_NAME:latest'
        }
      }
   }
      stage('Copy Files from Ansible to Kube Server') {
        sshagent(['kube_ssh']){
          sh 'ssh -o StrictHostKeyChecking=no ubuntu@172.31.35.205 '
          sh 'scp /var/lib/jenkins/workspace/devops_project/* ubuntu@172.31.31.34:/home/ubuntu/'
        }
      }
}
```