

DEVOPS Capstone Projects-01

Assignment Submitted By:-Hitesh Chauhan

Course Offered: -Advanced Cloud Computing and Devops

Assignment By: -Intellipaat

Trainer: -Kumar

Date Of Submission: -17/03/2025

CAPSTONE PROJECTS

You have been Hired Sr. DevOps Engineer in Abode Software. They want to implement DevOps Lifecycle in their company. You have been asked to implement this lifecycle as fast as possible. Abode Software is a product-based company, their product is available on this GitHub link.

<https://github.com/hshar/website.git>

Following are the specifications of the lifecycle:

1. Install the necessary software on the machines using a configuration management tool.
2. Git Workflow has to be implemented.

3.Code Build should automatically be triggered once commit is made to master branch or develop branch.

- If commit is made to master branch, test and push to prod.
- If commit is made to develop branch, just test the product, do not push to prod.

4.The Code should be containerized with the help of a Dockerfile. The Dockerfile should be built every time there is a push to Git-Hub. Use the following pre-built container for your application:

hshar/webapp

The code should reside in '/var/www/html'

5.The above tasks should be defined in a Jenkins Pipeline, with the following jobs:

Job1 : build

Job2: test

Job3 : prod

Prerequisites

1. Infrastructure Setup

- Ensure that you have a server (EC2 instance) to install and run Jenkins.
- Install a configuration management tool (**Ansible**) to automate software installation.
- Ensure that Docker is installed on the machine running Jenkins.

2. Version Control System

- Set up a **GitHub repository** for the source code.
- Implement a **Git workflow**, ensuring branches like master and develop exist.
- Developers should commit and push their code to these branches following best practices.

3. Jenkins Installation and Configuration

- Install **Jenkins** on the server.
- Install necessary plugins:
 - **Pipeline Plugin**
- Configure **webhook integration** between GitHub and Jenkins to trigger builds on commits.

4. Docker Setup

- Ensure **Docker Engine** is installed on the Jenkins server.
- Verify that the user running Jenkins has permissions to execute Docker commands.

5. Jenkins Pipeline Dependencies

- Ensure the Jenkins pipeline script (Jenkinsfile) includes three jobs:
 - **Job 1: Build** – Pull code from GitHub, create a Docker image.
 - **Job 2: Test** – Run test cases inside the container.
 - **Job 3: Prod** – Deploy the containerized application (only for master branch commits).

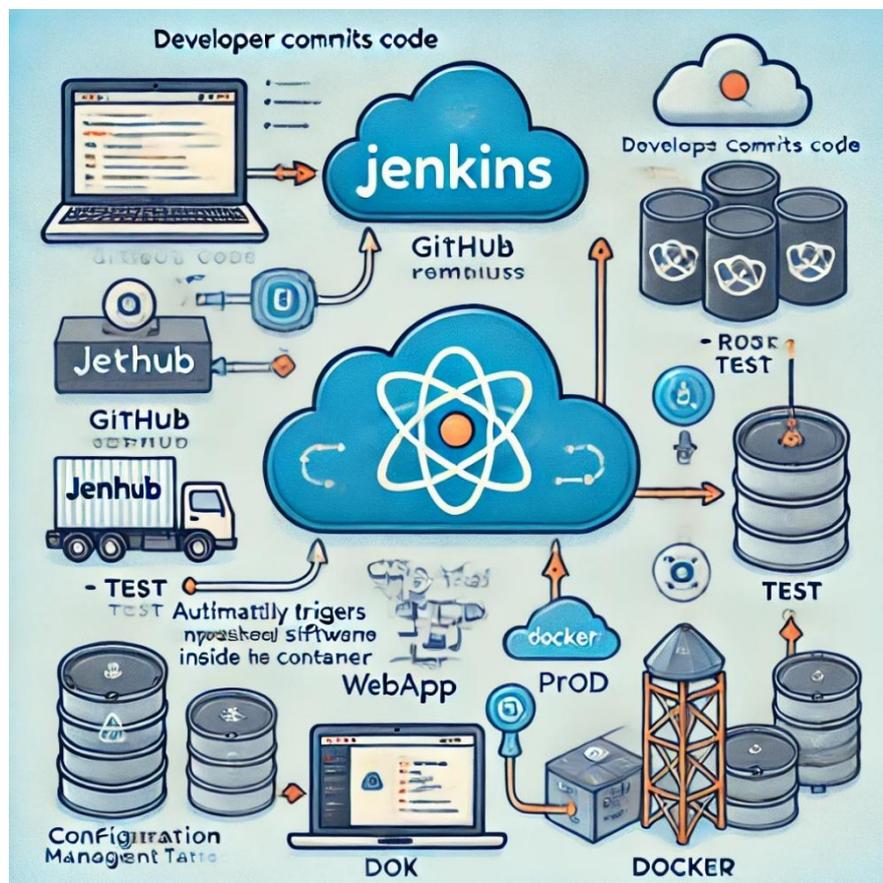
6. Web Application Code Placement

- Ensure that the application code is placed in `/var/www/html` inside the container as per the requirement.
-

ARCHITECTURE OF THIS PROJECT

Components in the Architecture:

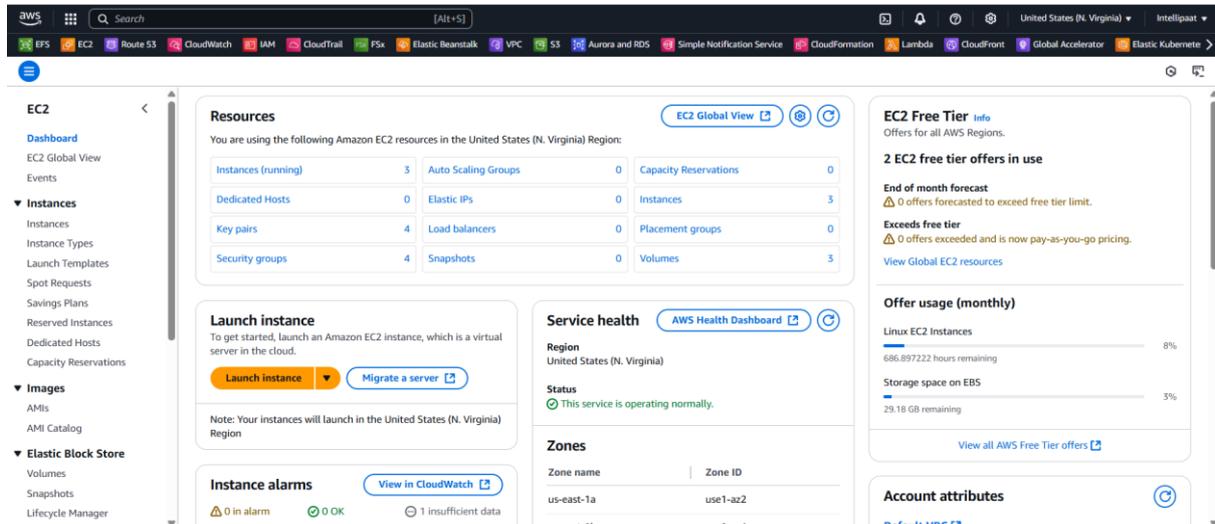
1. Developer commits code to GitHub (master or develop branch).
2. GitHub Webhook triggers Jenkins Pipeline, which executes:
 - Job 1: Build – Clones the repository and builds a Docker image.
 - Job 2: Test – Runs tests inside the container.
 - Job 3: Prod (only for master branch) – Pushes the container to production.
3. Dockerized Application Deployment using the pre-built container (hshar/webapp).
4. Configuration Management Tool installs necessary software on target machines.



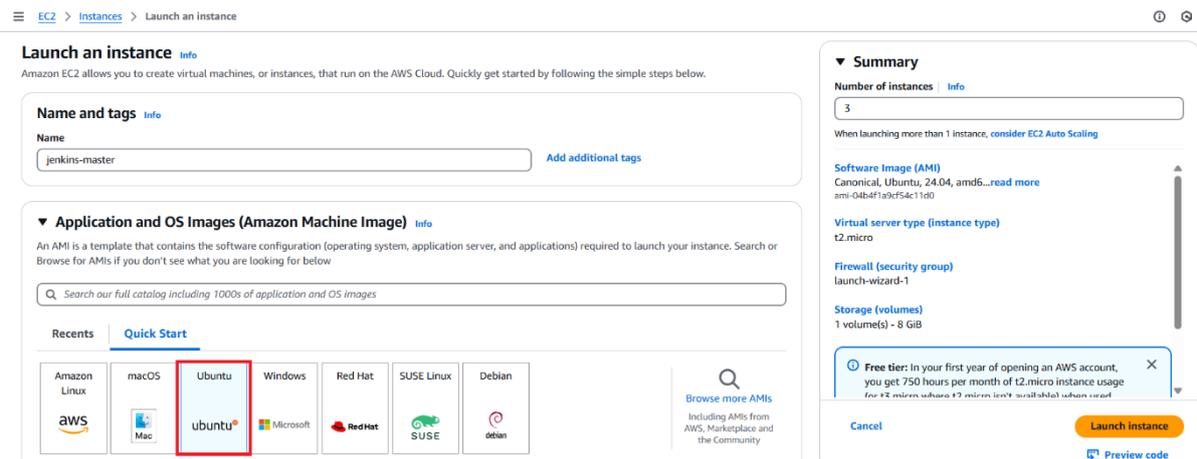
SOLUTION

1. Install the necessary software on the machines using a configuration management tool.

First We need to Infrastructure Setup and Log in to AWS Management Console.



Step 1: Launch an EC2 Instance And Choose Ubuntu And Number Of Instances Will Be 3.



Step 2: Choose Amazon Machine Image.

Note :When you install the ansible in ubuntu it is require 22.04 version and above.

In this Case I choose Ubuntu Server 24.04. LTS Version.

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-04b4f1a9cf54c11d0 (64-bit (x86)) / ami-0a7a4e87939439934 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture

64-bit (x86)

AMI ID

ami-04b4f1a9cf54c11d0

Publish Date

2025-01-15

Username

ubuntu

Verified provider

Instance type

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

Free tier eligible

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

Step 3: Choose Key Pair

Key pair (login)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

docker

Create new key pair

Network settings

Network

vpc-056bf1d89dcf9098e

Subnet

No preference (Default subnet in any availability zone)

Auto-assign public IP

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group

Select existing security group

Common security groups

Select security groups

launch-wizard-1 sg-047bfe49f4b83fc4a

VPC: vpc-056bf1d89dcf9098e

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Step 4: Configure Storage.

Configure storage

1x 8 GiB gp3 Root volume, 3000 IOPS, Not encrypted

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Add new volume

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

Click refresh to view backup information

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 File systems

Advanced details

Virtual server type (instance type)

t2.micro

Firewall (security group)

launch-wizard-1

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Launch instance

Preview code

Step 5: Launch Instance

Instances (1/3)

Find Instance by attribute or tag (case-sensitive)

All states

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Put
jenkins-master	i-044dc88a5365b1e7b	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-54-173-187-32.co...	54:
jenkins-slave1	i-082374ce4e9c83636	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-3-88-116-24.comp...	3.8i
jenkins-slave2	i-068dc2e3ddc6ad4ef	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-54-175-11-165.co...	54:

Now Install the Ansible in Ansible Master Server

First we need to go to ansible documentation for install the ansible.

https://docs.ansible.com/ansible/latest/installation_guide/installation_on_distros.html

Installing Ansible on Ubuntu

Ubuntu builds are available in a PPA [here](#).

To configure the PPA on your system and install Ansible run these commands:

```
$ sudo apt update
$ sudo apt install software-properties-common
$ sudo add-apt-repository --yes --update ppa:ansible/ansible
$ sudo apt install ansible
```

Note

On older Ubuntu distributions, "software-properties-common" is called "python-software-properties". You may want to use `apt-get` rather than `apt` in older versions. Also, be aware that only newer distributions (that is, 18.04, 18.10, and later) have a `-u` or `--update` flag. Adjust your script as needed.

File any issues in the PPA's [issue tracker](#).

For this installation now we need to create script file.

Created `install.sh` for ansible installation and this files as a `.sh`.

```
GNU nano 7.2                                install.sh *
sudo apt update -y
sudo apt install software-properties-common
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible -y
```

```
sudo apt update -y
```

```
sudo apt install software-properties-common
```

```
sudo add-apt-repository --yes --update ppa:ansible/ansible
```

```
sudo apt install ansible -y
```

Now We need to run this script so command is

```
bash install.sh
```

```
ubuntu@ip-172-31-87-210:~$ sudo touch install.sh
ubuntu@ip-172-31-87-210:~$ sudo nano install.sh
ubuntu@ip-172-31-87-210:~$ bash install.sh
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [364 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [20.0 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Fetched 789 kB in 1s (1040 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
140 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
```

```

Unpacking python3-ntlm-auth (1.5.0-1) ...
Selecting previously unselected package python3-paramiko.
Preparing to unpack .../06-python3-paramiko_2.12.0-2ubuntu4.1_all.deb ...
Unpacking python3-paramiko (2.12.0-2ubuntu4.1) ...
Selecting previously unselected package python3-requests-ntlm.
Preparing to unpack .../07-python3-requests-ntlm_1.1.0-3_all.deb ...
Unpacking python3-requests-ntlm (1.1.0-3) ...
Selecting previously unselected package python3-xmltodict.
Preparing to unpack .../08-python3-xmltodict_0.13.0-1_all.deb ...
Unpacking python3-xmltodict (0.13.0-1) ...
Selecting previously unselected package python3-winrm.
Preparing to unpack .../09-python3-winrm_0.4.3-2_all.deb ...
Unpacking python3-winrm (0.4.3-2) ...
Selecting previously unselected package sshpass.
Preparing to unpack .../10-sshpass_1.09-1_amd64.deb ...
Unpacking sshpass (1.09-1) ...
Setting up python3-ntlm-auth (1.5.0-1) ...
Setting up python3-resolvelib (1.0.1-1) ...
Setting up python3-kerberos (1.1.14-3.1build9) ...
Setting up ansible-core (2.17.9-1ppa~noble) ...
Setting up sshpass (1.09-1) ...
Setting up python3-xmltodict (0.13.0-1) ...
Setting up ansible (10.7.0-1ppa~noble) ...
Setting up python3-nacl (1.5.0-4build1) ...
Setting up python3-requests-ntlm (1.1.0-3) ...
Setting up python3-winrm (0.4.3-2) ...
Setting up python3-paramiko (2.12.0-2ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-87-210:~$ █

```

After Ansible Successfully install.we need to check the ansible version

The command is

ansible --version

```

ubuntu@ip-172-31-87-210:~$ ansible --version
ansible [core 2.17.9]
  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, Nov  6 2024, 18:32:19) [GCC 13.2.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
ubuntu@ip-172-31-87-210:~$ █

```

Now We need to install in the ansible master

1.JAVA 2.ANSIBLE 3.JENKINS

ANSIBLE MASTER	ANSIBLE SLAVE 1	ANSIBLE SLAVE 2
JAVA	JAVA	JAVA
ANSIBLE	DOCKER	DOCKER
JENKIN		

Now we need to create a connection b/w master and slaves1,slave2 for the connection, we need to generate a keypair in master and pasting the keypair will help us to connect with slave1,slave2

ssh-keygen -t rsa in ansible master.

key will be saved in :

/home/ubuntu/.ssh/id_rsa.pub

Reference Ansible Assignment 01(Page No 10)

```
ubuntu@ip-172-31-87-210:~/.ssh$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:kK0u80CK6q/kDtg7LD0s0pjgFBjKnG0AnGWZeZiGFx8 ubuntu@ip-172-31-87-210
The key's randomart image is:
+---[RSA 3072]-----+
|+ ++BE
|.=.0...o
|=.* ..o .
|o= o o
| =+ . S
|o++++.
|B=o.o..
|Xo.. .
|B*+o
+----[SHA256]-----+
ubuntu@ip-172-31-87-210:~/.ssh$ █
```

Go to this path **/home/ubuntu/.ssh**

Here key has been generated

```
ubuntu@ip-172-31-87-210:~/.ssh$ ll
total 20
drwx----- 2 ubuntu ubuntu 4096 Mar 16 18:33 ./
drwxr-x--- 5 ubuntu ubuntu 4096 Mar 16 18:21 ../
-rw----- 1 ubuntu ubuntu 388 Mar 16 17:14 authorized_keys
-rw----- 1 ubuntu ubuntu 2610 Mar 16 18:33 id_rsa
-rw-r--r-- 1 ubuntu ubuntu 577 Mar 16 18:33 id_rsa.pub
ubuntu@ip-172-31-87-210:~/.ssh$ █
```

This key require **id_rsa.pub** for password less authentication.

Now we need to run this command

cat id_rsa.pub

```
ubuntu@ip-172-31-87-210:~/.ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAmxCtD+DVGz3LYX5b+DmZbw+2mo+Co6r0zCCUDi11W/o0EtgsbIzg8ozSwuK10D1XxaVMLxInNhvwHxfSnmMCWtb5MyLveP/p4LYBfrzoBIIyYpoF9uPPqtJvWt6fR50
LmGfuhYZSvwRwB0inMeZAJprC07fByL8IISLf9ujnhICJ8C53v1Eq7IQLOPW9Pj rmmVt:DNc1Bw6pqt37PYKwrxrJrGnCK0mY7poPspxnjLUtLL0bjX5IFIJsfshkJPtswEh iQfnpEnChtLDRDsbEu12n5dEJ+uYEL0CpR
EmJv5KeXQ//mCc0xf5vML2Z9McR/6BUipsT9FcWQnk6JEH3fEREQ0INzkvB329AkyufynIYcHaRVtdiHw0bmW/8P4DRJnXXNbuDnCc8QmjboXL95TuwYoY5noOUIiUaaLmsa/gCY0UuNH5HfFP+YYi0JFNgpESHENwEJa7SZ1
2GRMEjW0rv09zoVLGwjyFIx9ZHJGLBwdz0C9rEQ0FoP0= ubuntu@ip-172-31-87-210
```

Copy this key and go to both slave server and paste the same path
But we need to paste in `authorized_key` under `.ssh` folder.

```
ubuntu@ip-172-31-81-95:~$ cd .ssh/
ubuntu@ip-172-31-81-95:~/.ssh$ ll
total 12
drwx----- 2 ubuntu ubuntu 4096 Mar 16 17:14 ./
drwxr-x--- 4 ubuntu ubuntu 4096 Mar 16 17:42 ../
-rw----- 1 ubuntu ubuntu 388 Mar 16 17:14 authorized_keys
ubuntu@ip-172-31-81-95:~/.ssh$ █
```

Now check through ansible master server.

My Slave 2 has been successfully connected through passwordless.

Same process with slave 2

```
ubuntu@ip-172-31-87-210:~/.ssh$ ssh 172.31.81.95
The authenticity of host '172.31.81.95 (172.31.81.95)' can't be established.
ED25519 key fingerprint is SHA256:mjaQjmgwVULfQzp96TVJM4taH4sL2rSgmfcipYaFRk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? █
```

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.31.81.95' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)
```

```
* Documentation: https://help.ubuntu.com
* Management:   https://landscape.canonical.com
* Support:      https://ubuntu.com/pro
```

System information as of Sun Mar 16 18:42:26 UTC 2025

```
System load: 0.0          Processes:             110
Usage of /: 28.5% of 6.71GB Users logged in:      1
Memory usage: 21%        IPv4 address for enX0: 172.31.81.95
Swap usage: 0%
```

Expanded Security Maintenance for Applications is not enabled.

```
146 updates can be applied immediately.
47 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
```

```
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
```

```
Last login: Sun Mar 16 17:15:56 2025 from 106.222.209.100
ubuntu@ip-172-31-81-95:~$ █
```

```
ubuntu@ip-172-31-87-210:~/ssh$ ssh 172.31.87.60
The authenticity of host '172.31.87.60 (172.31.87.60)' can't be established.
ED25519 key fingerprint is SHA256:ehdQs4o/IIeGZQJRbWQHMIzOZ0lSzJ8spZpU6GUhP8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? █
```

This Is ansible slave-2.this server also connected through passwordless authentication

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.31.87.60' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Sun Mar 16 18:44:15 UTC 2025

System load:  0.0                Processes:            110
Usage of /:   28.5% of 6.71GB    Users logged in:    1
Memory usage: 22%                IPv4 address for enX0: 172.31.87.60
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

146 updates can be applied immediately.
47 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sun Mar 16 17:16:33 2025 from 106.222.209.100
ubuntu@ip-172-31-87-60:~$ █
```

we need to add the slave Private-IP int the ansible host file

path of the ansible host file: cd /etc/ansible/

```
ubuntu@ip-172-31-87-210:/etc/ansible$ ll
total 20
drwxr-xr-x  3 root root 4096 Mar 16 18:19 ./
drwxr-xr-x 109 root root 4096 Mar 16 18:19 ../
-rw-r--r--  1 root root  614 Feb 25 16:59 ansible.cfg
-rw-r--r--  1 root root 1175 Feb 25 16:59 hosts
drwxr-xr-x  2 root root 4096 Feb 25 16:59 roles/
ubuntu@ip-172-31-87-210:/etc/ansible$ █
```

sudo nano hosts.We need to grouping the server.

[prod]

172.31.23.185

[test]

172.31.25.172

NOTE:-If you want to add more any of server in this prod and test section just add the ip address.

```
GNU nano 7.2                               hosts
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern, you can specify
# them like this:

## www[001:006].example.com

# You can also use ranges for multiple hosts:

## db-[99:101]-node.example.com

# Ex 3: A collection of database servers in the 'dbservers' group:

## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

# Ex4: Multiple hosts arranged into groups such as 'Debian' and 'openSUSE':

## [Debian]
## alpha.example.org
## beta.example.org

## [openSUSE]
## green.example.com
## blue.example.com

172.31.81.95
172.31.87.60
```

to check whether the connection is successful or not

```
ubuntu@ip-172-31-87-210:/etc/ansible$ ansible -m ping all
[WARNING]: Platform linux on host 172.31.87.60 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
172.31.87.60 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "changed": false,
  "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.81.95 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
172.31.81.95 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu@ip-172-31-87-210:/etc/ansible$
```

After Done we need to create the one yaml file so this yaml file helps to install java and rest server install the docker and java. we need to mention what need to be give task.

```
GNU nano 7.2 install.yaml
--
- name: installing java and jenkins
  host: localhost
  tasks:
    - name: installing java and Jenkins
      script: localhost.sh

- name: installing java and docker by ansible slave 1 and slave 2 machine
  host: all
  tasks:
    - name: installing java and docker
      script: slave.sh
```

```
GNU nano 7.2 localhost.sh *
sudo apt update -y
sudo apt install openjdk-17-jdk -y

sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null" > /dev/null
sudo apt-get update -y
sudo apt-get install jenkins -y
```

```
GNU nano 7.2 slave.sh
sudo apt update -y
sudo apt install openjdk-17-jdk -y
sudo apt install docker.io -y
```

Now Run this playbook.

```
ubuntu@ip-172-31-87-210:/etc/ansible$ ansible-playbook install.yaml

PLAY [Installing Java and Jenkins on Master] *****
TASK [Gathering Facts] *****
ok: [localhost]

TASK [Running local setup script] *****
changed: [localhost]

PLAY [Installing Java and Docker on Slave Machines] *****
TASK [Gathering Facts] *****
[WARNING]: Platform linux on host 172.31.87.60 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.87.60]
[WARNING]: Platform linux on host 172.31.81.95 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.81.95]

TASK [Running slave setup script] *****
changed: [172.31.81.95]
changed: [172.31.87.60]

PLAY RECAP *****
172.31.81.95      : ok=2   changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.87.60      : ok=2   changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
localhost        : ok=2   changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-87-210:/etc/ansible$
```

Now Check the Version in both slaves

```
ubuntu@ip-172-31-81-95:/etc$ java --version
openjdk 17.0.14 2025-01-21
OpenJDK Runtime Environment (build 17.0.14+7-Ubuntu-124.04)
OpenJDK 64-Bit Server VM (build 17.0.14+7-Ubuntu-124.04, mixed mode, sharing)
ubuntu@ip-172-31-81-95:/etc$ docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
```

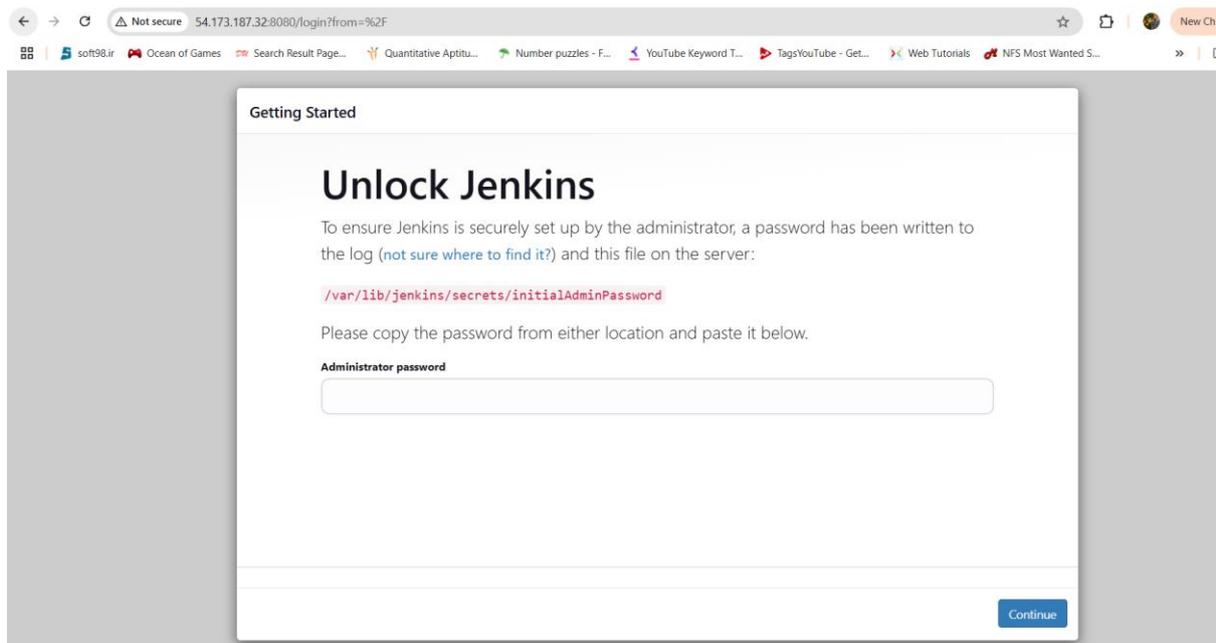
```
ubuntu@ip-172-31-87-60:~/ssh$ java --version
openjdk 17.0.14 2025-01-21
OpenJDK Runtime Environment (build 17.0.14+7-Ubuntu-124.04)
OpenJDK 64-Bit Server VM (build 17.0.14+7-Ubuntu-124.04, mixed mode, sharing)
ubuntu@ip-172-31-87-60:~/ssh$ docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
ubuntu@ip-172-31-87-60:~/ssh$
```

Jenkin version Check in ansible master server

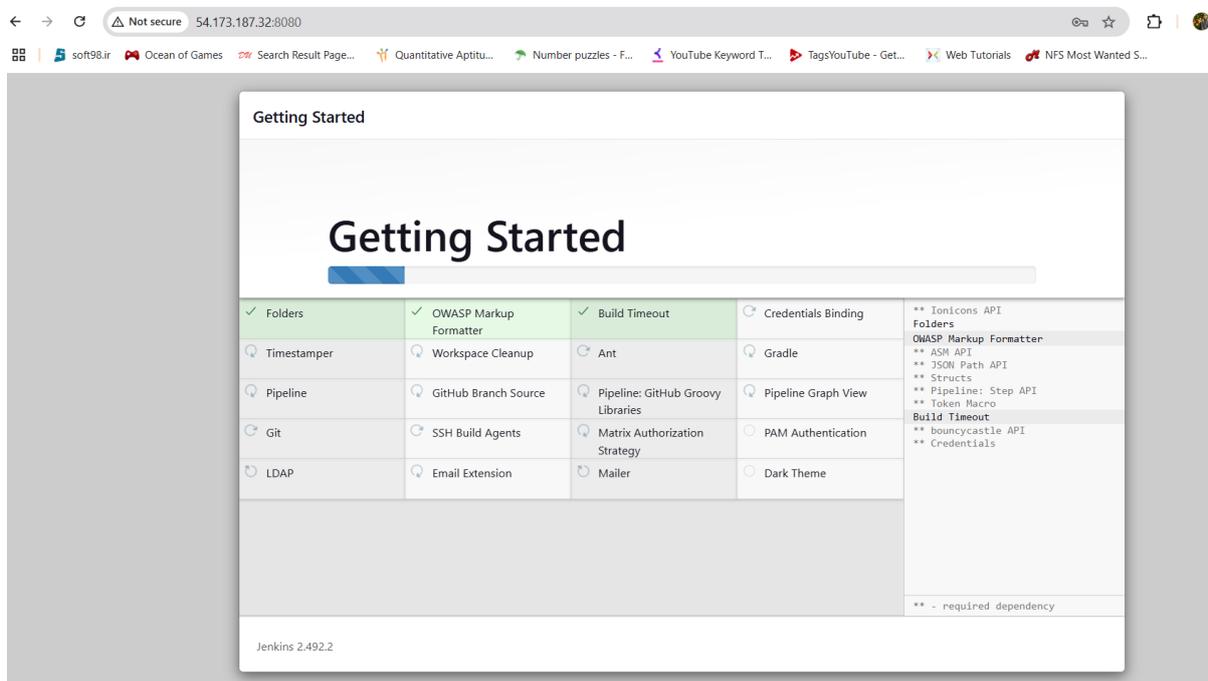
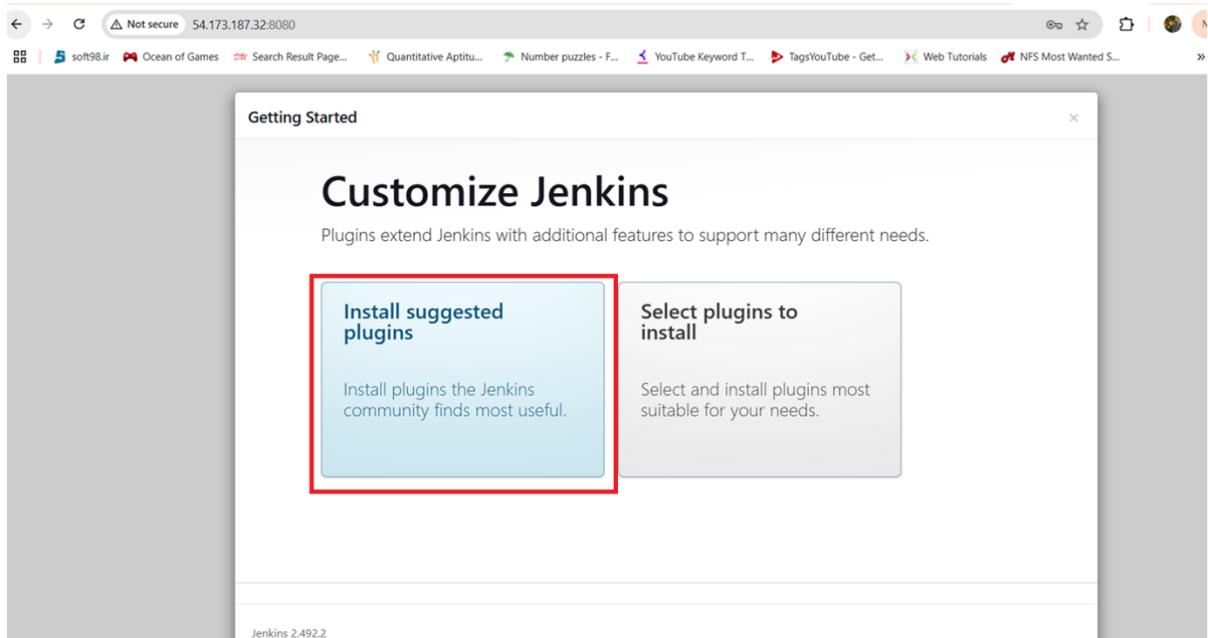
```
ubuntu@ip-172-31-87-210:/etc/ansible$ jenkins --version
2.492.2
```

Now We need to configure the Jenkins dashboard

Go to **sudo cat /var/lib/jenkins/secrets/initialAdminPassword**

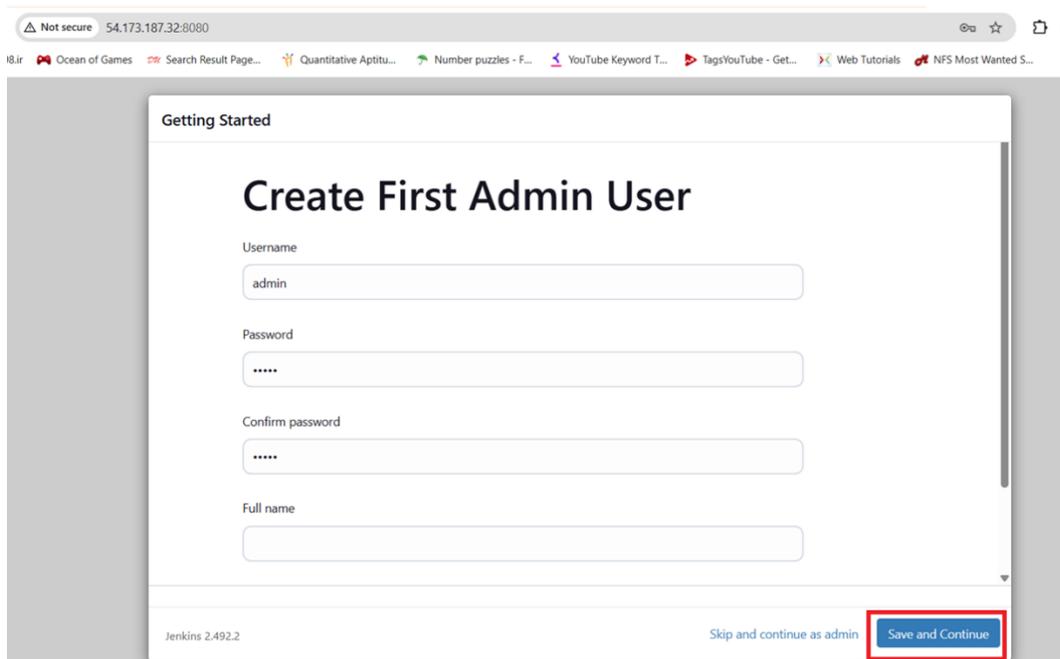
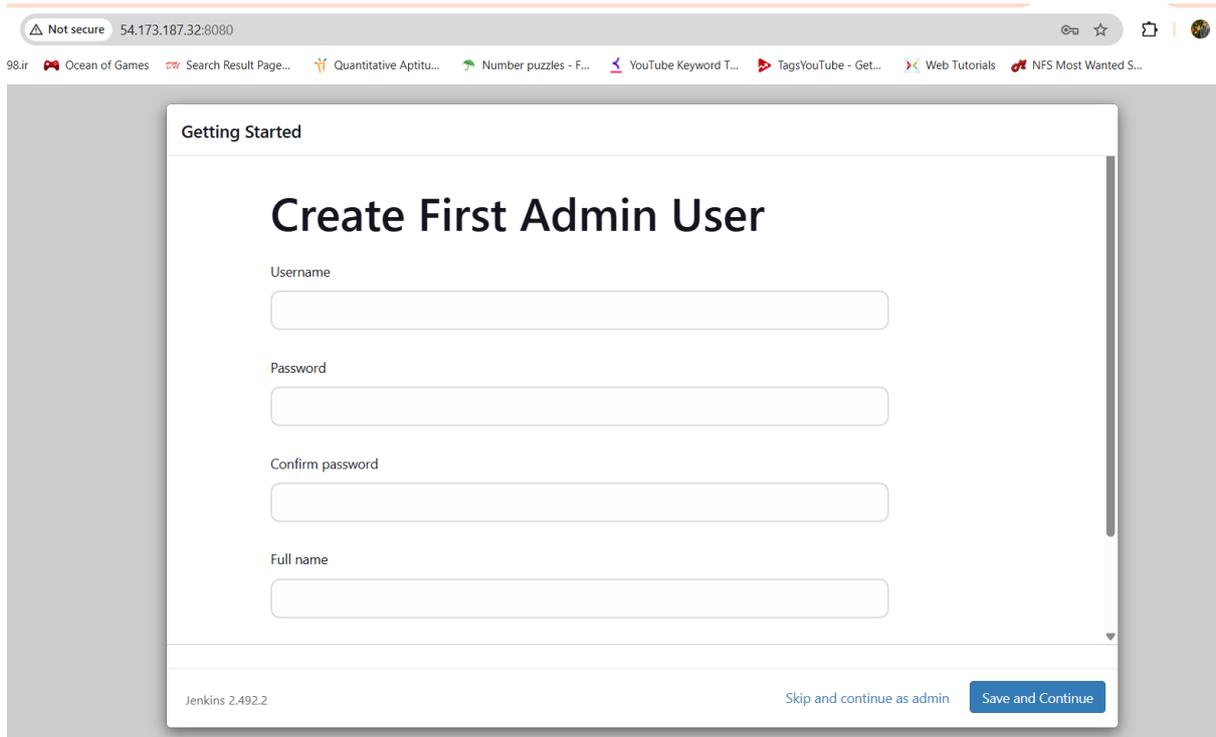


```
ubuntu@ip-172-31-87-210:/etc/ansible$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
34ef7814079c424ba28f5d713fce259b
```

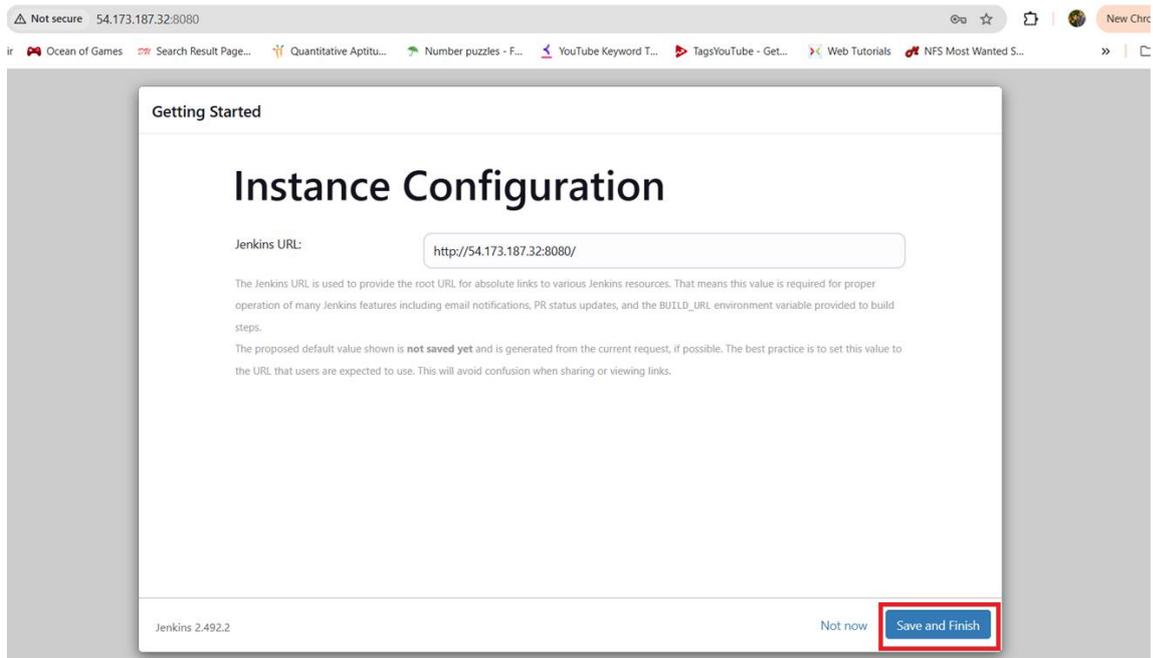


Now After Installed The Plugins.

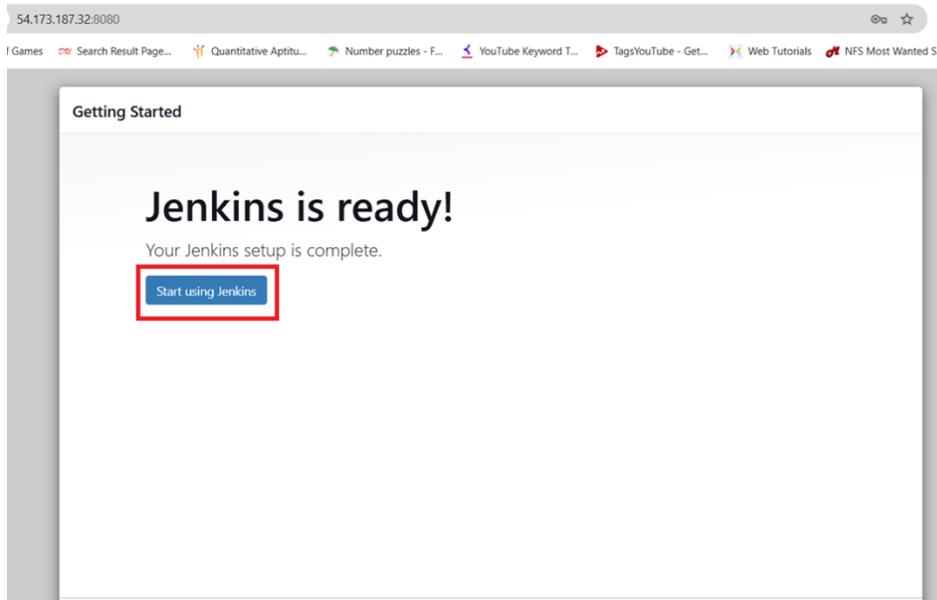
You need to fill your user name, password and email id in this form as per your choice.



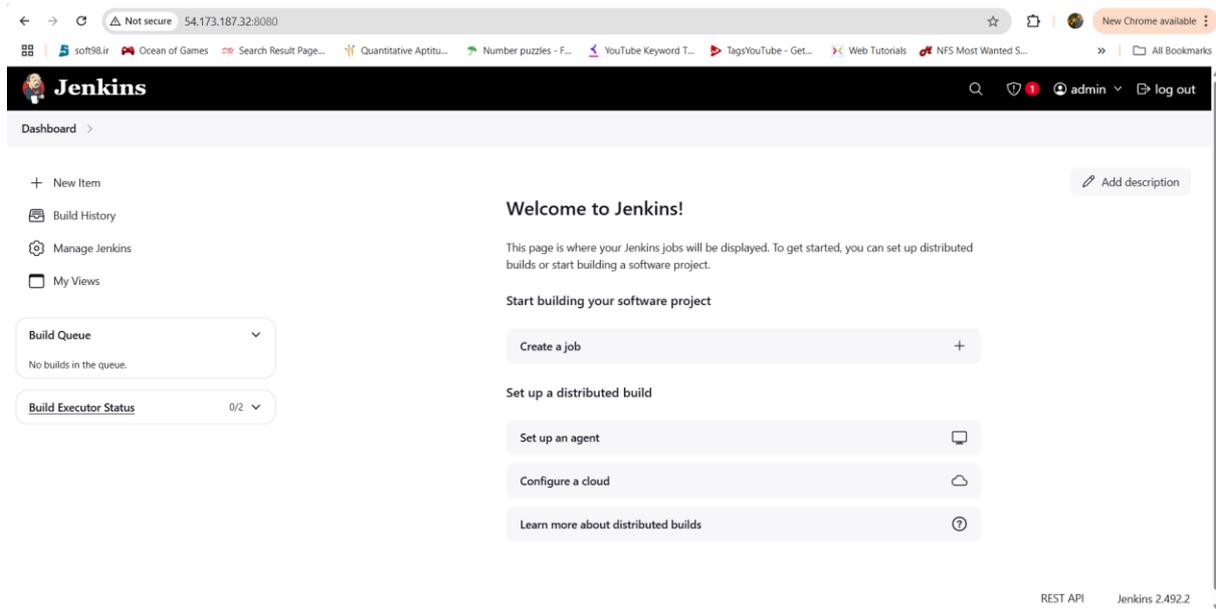
After Setup Your username and password.you need to configure your instance with jenkins urls.



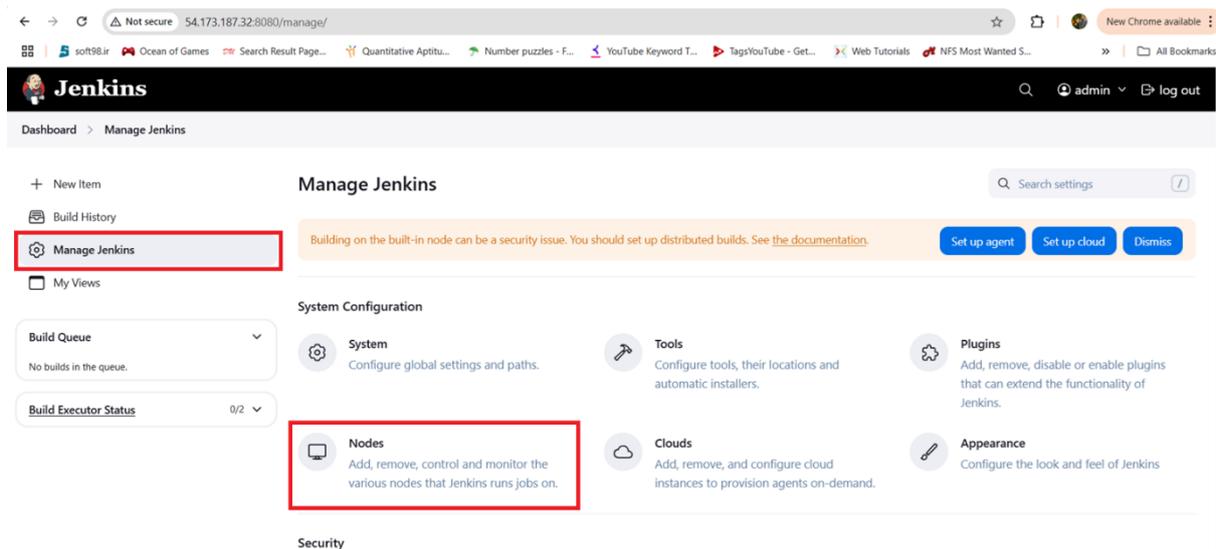
Then Click Start using Jenkin



After click the Start using Jenkin you will see jenkins dashboard.

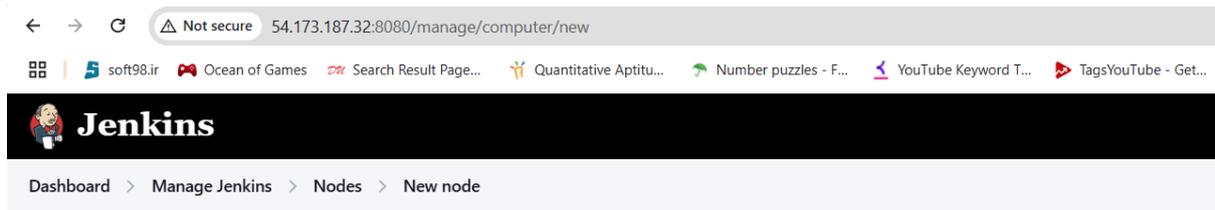


Now Need to configure nodes in Jenkins.
Jenkins > Mange Jenkins > Click Nodes.



After Click Nodes.

You will need to fill the all information regarding the nodes.



New node

Node name

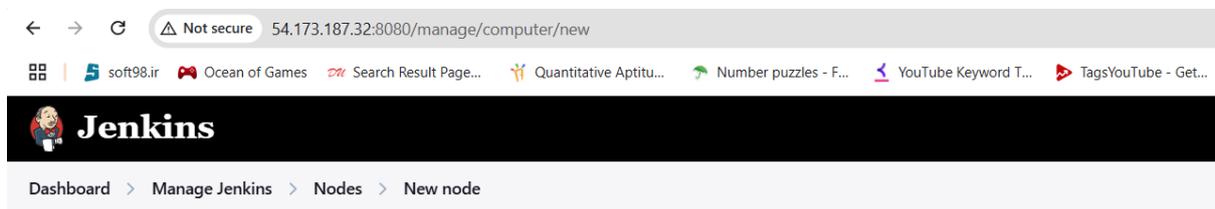
Type

Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

Create

Node Name:-slave-1 then click Create.



New node

Node name

Type

Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

Create

After Click Create You need to put the remote root directory.
In My Case.this is my jenkins workspace path.
/home/ubuntu/jenkins

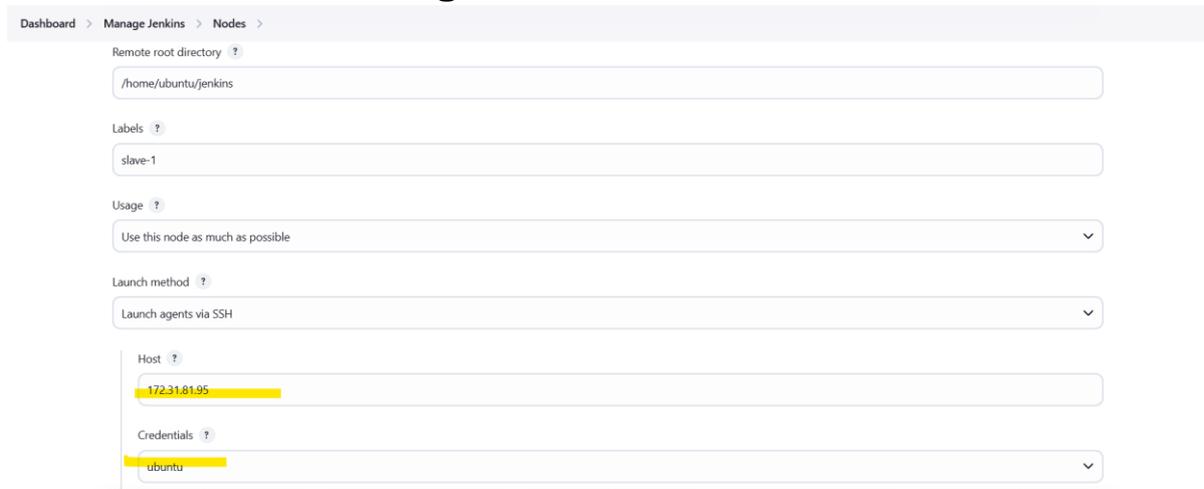


The screenshot shows the Jenkins configuration page for a node named 'slave-1'. The 'Remote root directory' field is highlighted with a red box and contains the value '/home/ubuntu/jenkins'. Other fields include 'Name' (slave-1), 'Description', 'Number of executors' (1), and 'Plain text' (Preview).

Host:172.31.81.95

Username:ubuntu

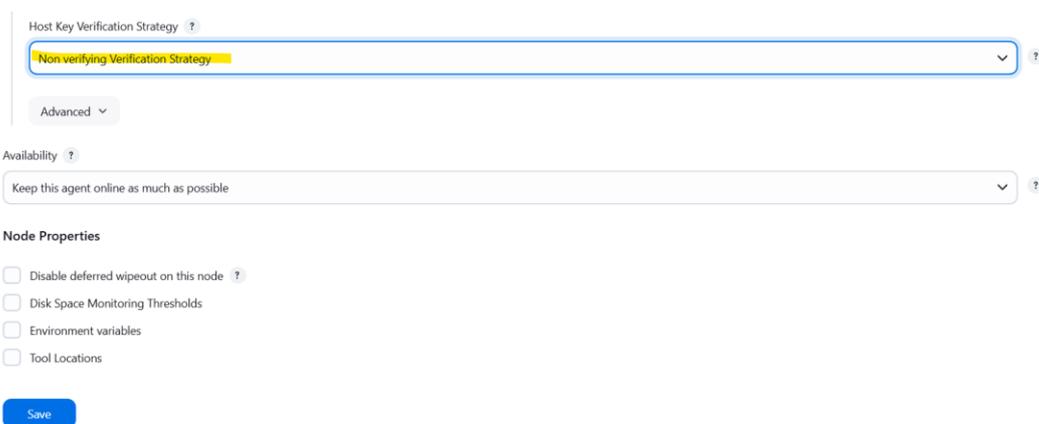
Launch Method:Launch agents via SSH.



The screenshot shows the Jenkins configuration page for a node named 'slave-1'. The 'Remote root directory' field contains '/home/ubuntu/jenkins'. The 'Labels' field contains 'slave-1'. The 'Usage' dropdown is set to 'Use this node as much as possible'. The 'Launch method' dropdown is set to 'Launch agents via SSH'. The 'Host' field contains '172.31.81.95' and the 'Credentials' dropdown is set to 'ubuntu'.

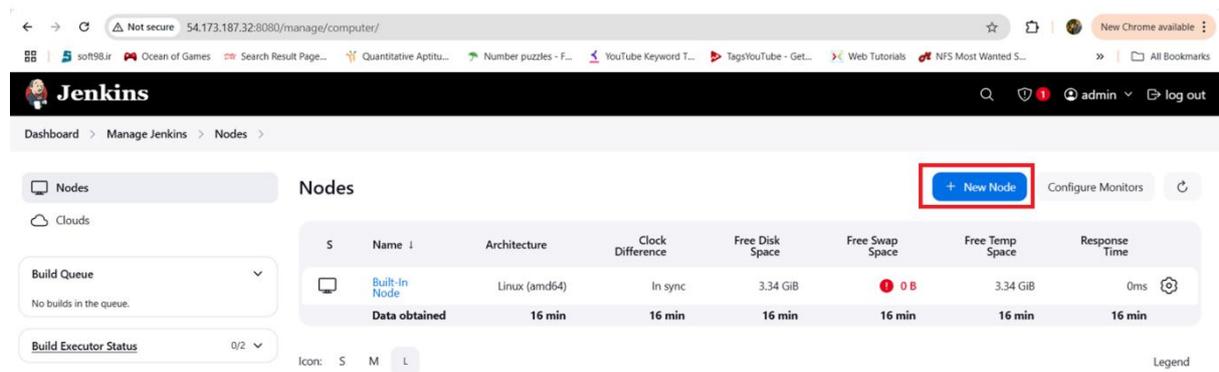
Here we need to put the .pem key.

Host Key Verification Strategy: Non Verifying Verification Strategy



The screenshot shows the 'Host Key Verification Strategy' configuration page. The 'Host Key Verification Strategy' dropdown is set to 'Non verifying Verification Strategy'. The 'Availability' dropdown is set to 'Keep this agent online as much as possible'. Under 'Node Properties', there are four checkboxes: 'Disable deferred wipeout on this node', 'Disk Space Monitoring Thresholds', 'Environment variables', and 'Tool Locations'. A 'Save' button is at the bottom.

After Click Save



The screenshot shows the Jenkins 'Nodes' page. A red box highlights the '+ New Node' button. The table below shows the 'Built-In Node' with a status of 'In sync'.

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	3.34 GiB	0 B	3.34 GiB	0ms
Data obtained		16 min	16 min	16 min	16 min	16 min	16 min

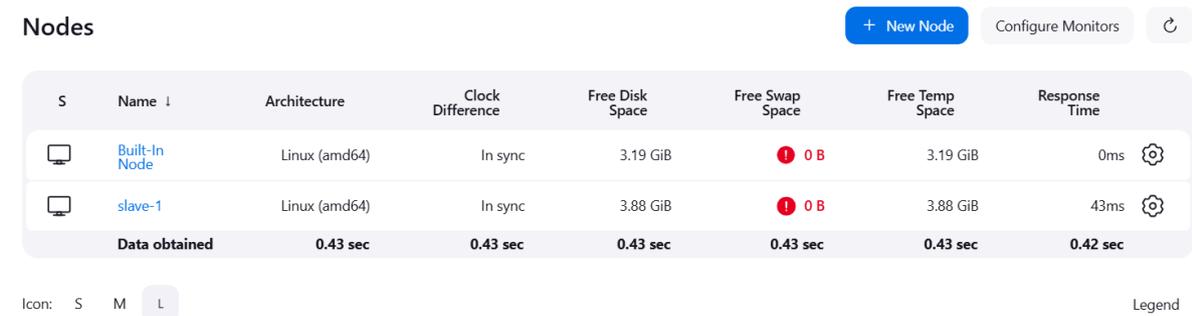
You will See the node which is create showing here.



The screenshot shows the Jenkins 'Nodes' page with two nodes listed in the table.

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	3.34 GiB	0 B	3.34 GiB	0ms
	slave-1	N/A	N/A	N/A	N/A	N/A	N/A
Data obtained		6 ms	3 ms	19 min	1 ms	19 min	19 min

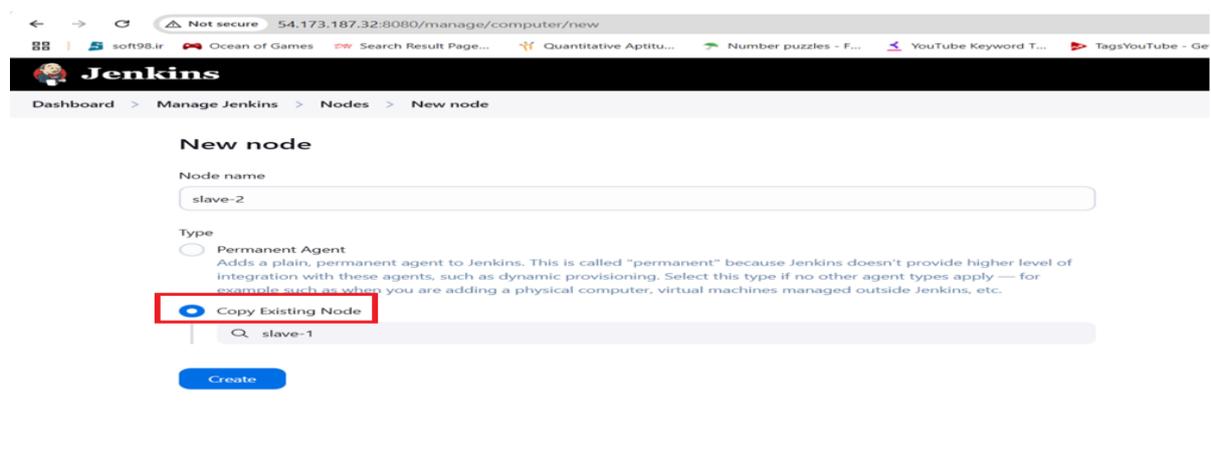
Node has been sync successfully.



The screenshot shows the Jenkins 'Nodes' page with three nodes listed in the table.

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	3.19 GiB	0 B	3.19 GiB	0ms
	slave-1	Linux (amd64)	In sync	3.88 GiB	0 B	3.88 GiB	43ms
Data obtained		0.43 sec	0.43 sec	0.43 sec	0.43 sec	0.43 sec	0.42 sec

Same Process We need to as it is slave-1



The screenshot shows the 'New node' form in Jenkins. The 'Node name' field contains 'slave-2'. The 'Type' section has 'Copy Existing Node' selected, and a search box below it contains 'slave-1'.

Node name:

Type: Copy Existing Node

Search:

Create

Dashboard > Manage Jenkins > Nodes > slave-2 > Configure

status

Delete Agent

Configure

Build History

Load Statistics

Log

Build Executor Status 0/1

slave-2 (offline)

slave-2

Description ?

Plain text Preview

Number of executors ?

1

Remote root directory ?

/home/ubuntu/jenkins

Labels ?

slave-2

Dashboard > Manage Jenkins > Nodes > slave-2 > Configure

Remote root directory ?

/home/ubuntu/jenkins

Labels ?

slave-2

Usage ?

Use this node as much as possible

Launch method ?

Launch agents via SSH

Host ?

172.31.87.60

Credentials ?

ubuntu

Save Apply

After Node Has been sync.

Jenkins

Dashboard > Manage Jenkins > Nodes >

Nodes

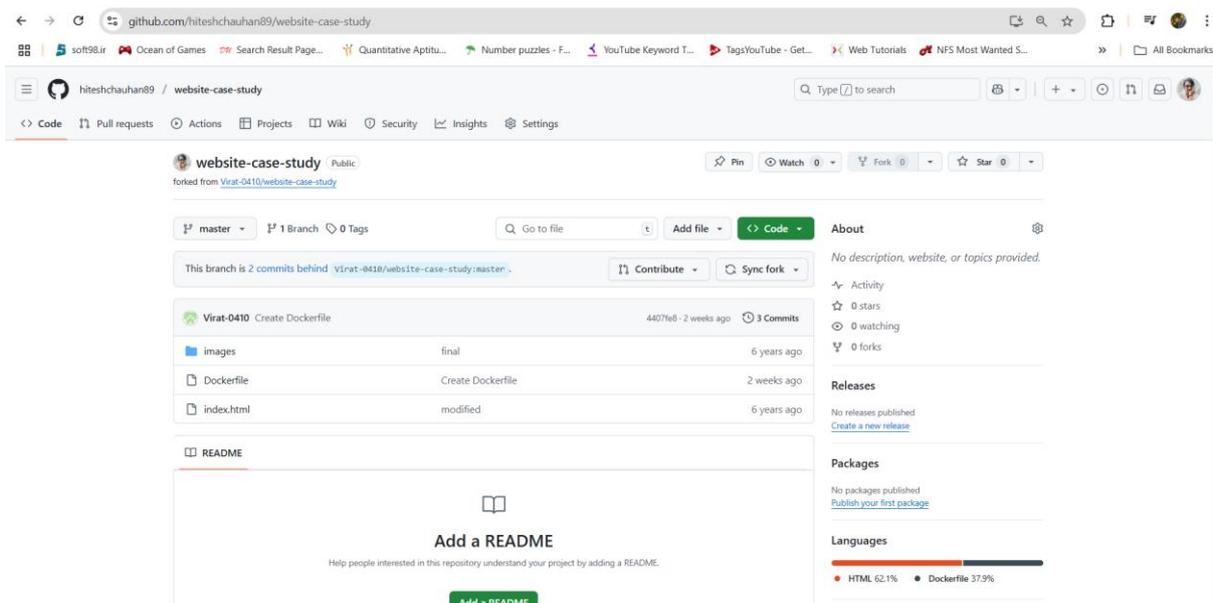
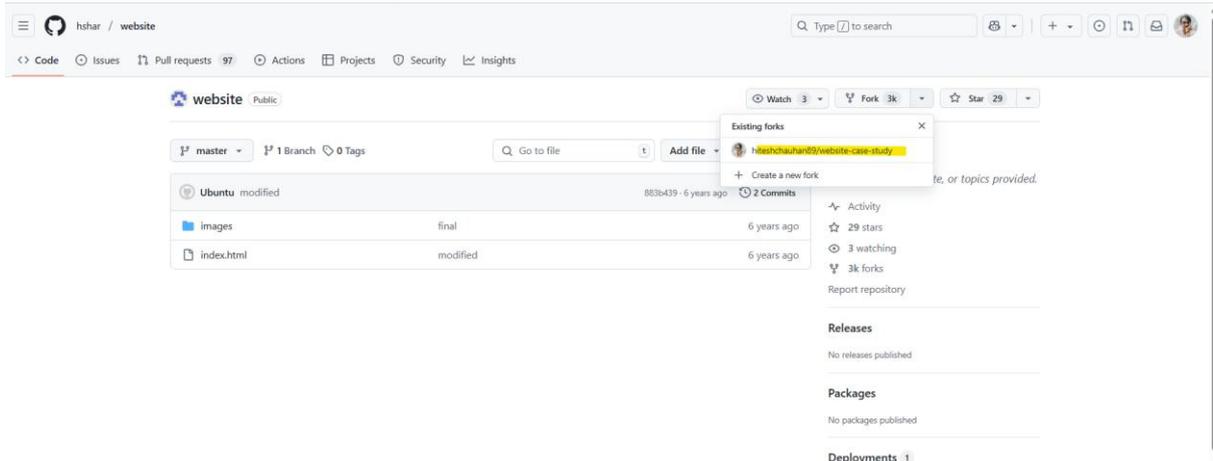
+ New Node Configure Monitors

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	3.19 GiB	0 B	3.19 GiB	0ms
	slave-1	Linux (amd64)	In sync	3.88 GiB	0 B	3.88 GiB	28ms
	slave-2	Linux (amd64)	In sync	3.88 GiB	0 B	3.88 GiB	23ms
Data obtained		0.28 sec	0.28 sec	0.28 sec	0.28 sec	0.28 sec	0.27 sec

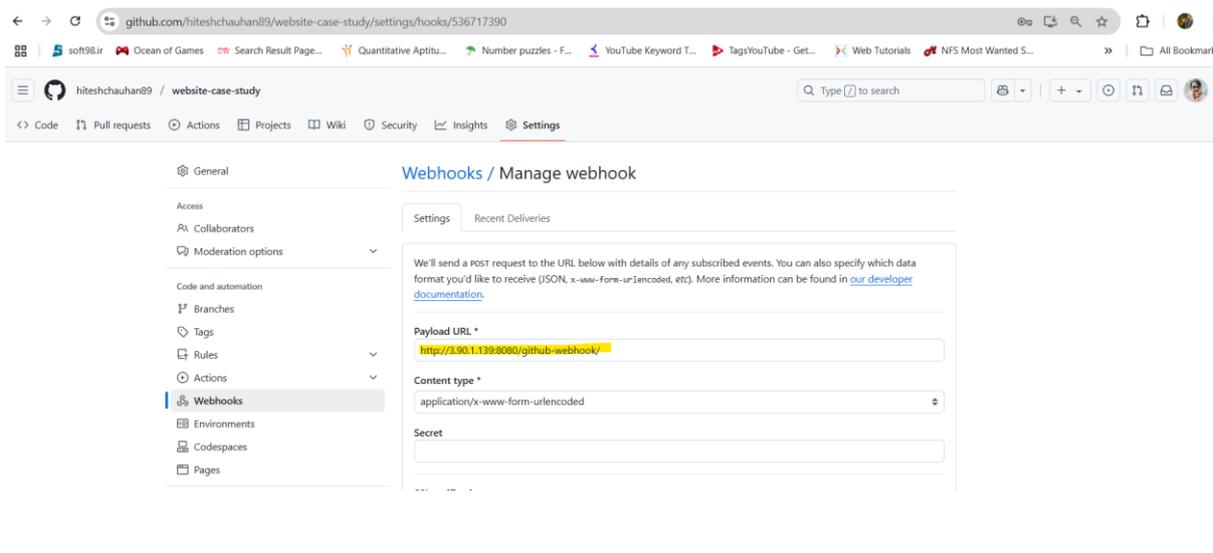
Icons: S M L Legend

Go To This Url and Fork this repositories in our github account
<https://github.com/hshar/website>

Fork This Url with your github.



After Fork Your Url Then go to same repository and click settings then click webhook.



Then Click Add Webhook

The screenshot shows the 'Add Webhook' configuration page in GitHub. On the left is a sidebar menu with options: Webhooks (selected), Environments, Codespaces, Pages, Security (Code security, Deploy keys, Secrets and variables), and Integrations (GitHub Apps, Email notifications). The main content area includes a text input field, 'SSL verification' options (Enable SSL verification is selected), 'Which events would you like to trigger this webhook?' (Just the push event is selected), and an 'Active' checkbox which is checked. Below the 'Active' checkbox is a note: 'We will deliver event details when this hook is triggered.' At the bottom, a green 'Add webhook' button is highlighted with a red rectangular box.

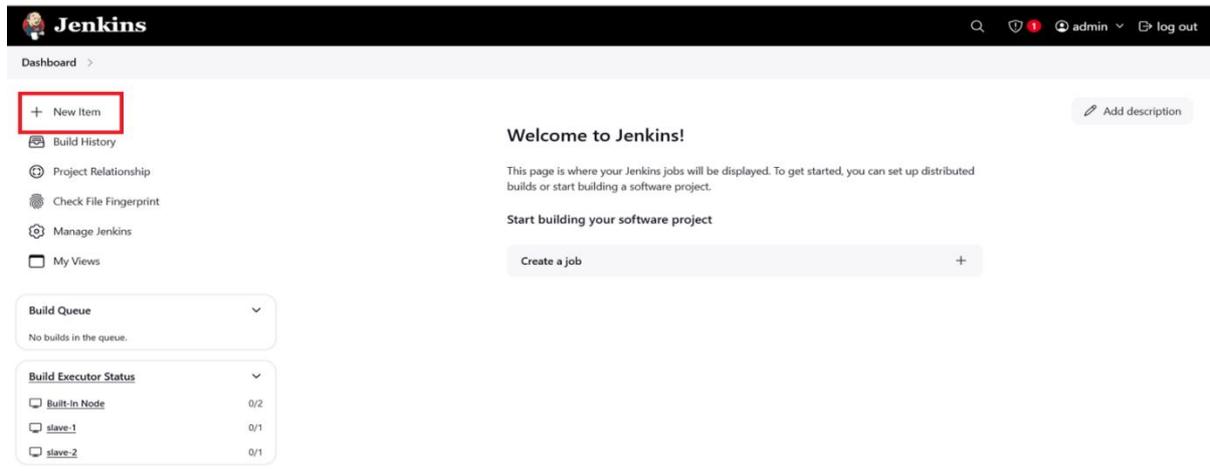
You will see like this.

The screenshot shows the 'Webhooks' section of the repository settings for 'hiteshchauhan89 / website-case-study'. The left sidebar has 'Webhooks' selected. The main area shows a list of webhooks with one entry: 'http://590.119940000/gmhub-webh... (push)'. The entry has a green checkmark and the text 'Last delivery was successful.' Below the entry are 'Edit' and 'Delete' buttons. An 'Add webhook' button is visible in the top right corner of the webhooks list.

Job1 : build

Go To Jenkins Url.We need to Create Jobs.

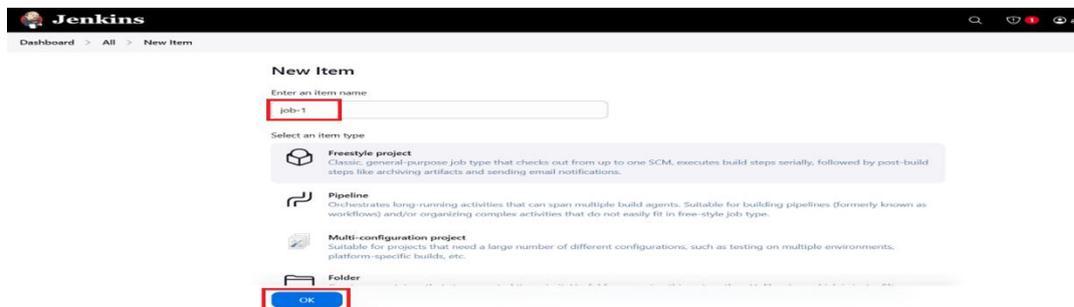
JOB-1



Now To Create job-1

Click **Dashboard >All>New Item** Then

Click OK

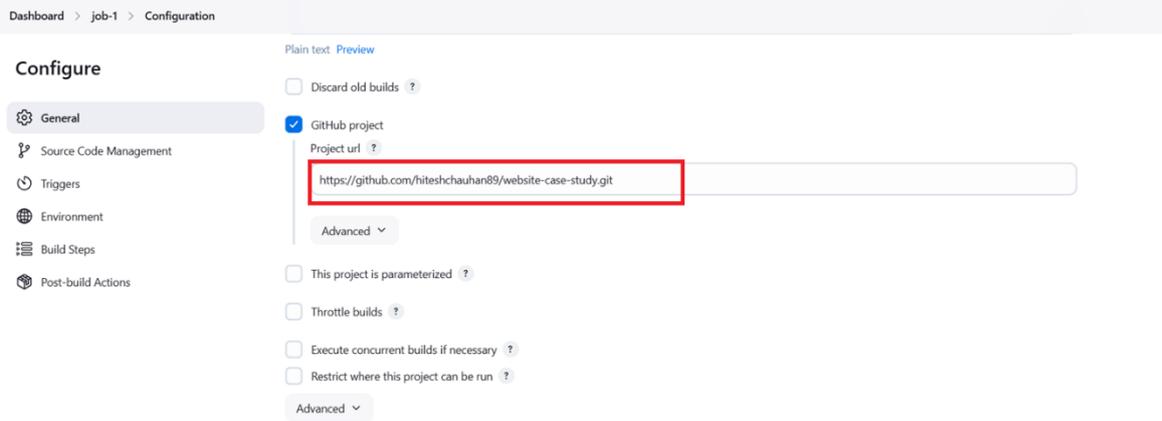


After You Need to choose GitHub Projects Url.Copy the Url we have added in forks.

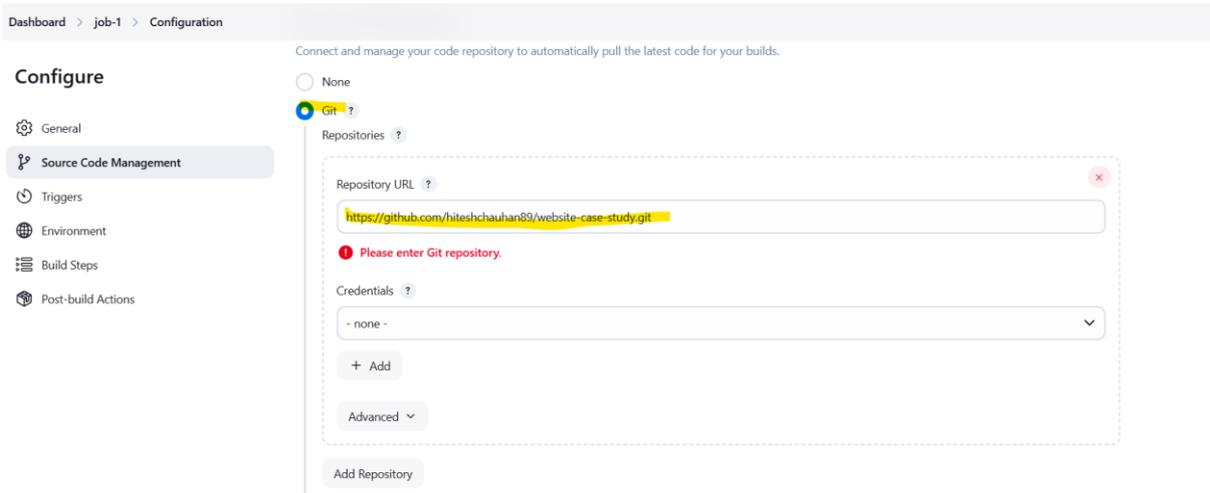
Project Url:- <https://github.com/hiteshchauhan89/website-case-study.git>

Check Box Restrict Where This Project can be run so mentioned **built-in**

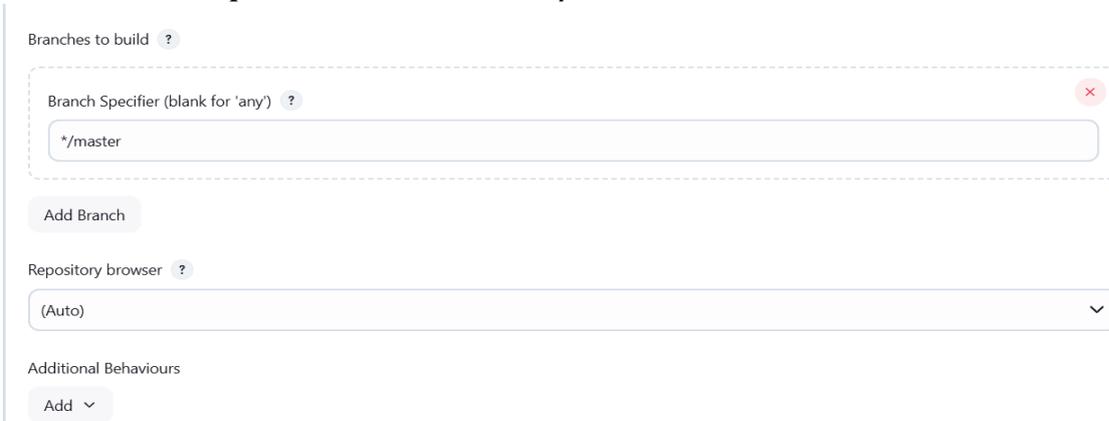




Then Source Code Management Choose Git same url need to mentioned here <https://github.com/hiteshchauhan89/website-case-study.git>



Then Next, We need to code which Branch To Build Here Branch Specifier mentioned ***/master**



Now Save This Settings

Dashboard > job-1 > Configuration

Configure

- General
- Source Code Management
- Triggers
- Environment**
- Build Steps
- Post-build Actions

- Add timestamps to the Console Output
- Inspect build log for published build scans
- Terminate a build if it's stuck
- With Ant ?

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

Add build step ▾

Post-build Actions

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

Add post-build action ▾

Save Apply

Then Click Build Now

← → ↻ ⚠ Not secure 54.173.187.32:8080/job/job-1/

soft98.ir Ocean of Games Search Result Page... Quantitative Aptitu... Number puzzles - F... YouTube Keyword T... TagsYouTub

Jenkins

Dashboard > job-1 >

- Status**
- Changes
- Workspace
- Build Now**
- Configure
- Delete Project
- GitHub Hook Log
- GitHub
- Rename

job-1

Permalinks

You will see the below Job-1 has been successfully Created.

Status

Changes

Workspace

Build Now

Configure

Delete Project

GitHub Hook Log

GitHub

Rename

job-1

Permalinks

Builds

Today
#1 7:44 PM

Now We need to check the outputs.

The screenshot shows the Jenkins Dashboard for job-1. On the left, there is a sidebar with a 'New Item' button highlighted in a red box. Below it are links for 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. The main area displays a table of build history for job-1. The table has columns for 'S' (Status), 'W' (Weather icon), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. A single build is listed with a green checkmark in the 'S' column, a sun icon in the 'W' column, and a duration of 3.5 sec. Below the table, there are sections for 'Build Queue' (showing no builds) and 'Build Executor Status' (showing 0/2 built-in nodes and 0/1 slave nodes).

This is the Job-1 Output.

The screenshot shows the Jenkins Console Output for job-1 build #1. The output text is as follows:

```
Started by user admin
Running as SYSTEM
Building on the built-in node in workspace /var/lib/jenkins/workspace/job-1
The recommended git tool is: NONE
No credentials specified
Cloning the remote git repository
Cloning repository https://github.com/hiteshchauan89/website-case-study.git
> git init /var/lib/jenkins/workspace/job-1 # timeout=10
Fetching upstream changes from https://github.com/hiteshchauan89/website-case-study.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/hiteshchauan89/website-case-study.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
> git config remote.origin.url https://github.com/hiteshchauan89/website-case-study.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 4407fe81b88d12338bdecf95472a6eaa2f91280c (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 4407fe81b88d12338bdecf95472a6eaa2f91280c # timeout=10
Commit message: "Create Dockerfile"
First time build. Skipping changelog.
Finished: SUCCESS
```

Now We can also Cross-very through server.

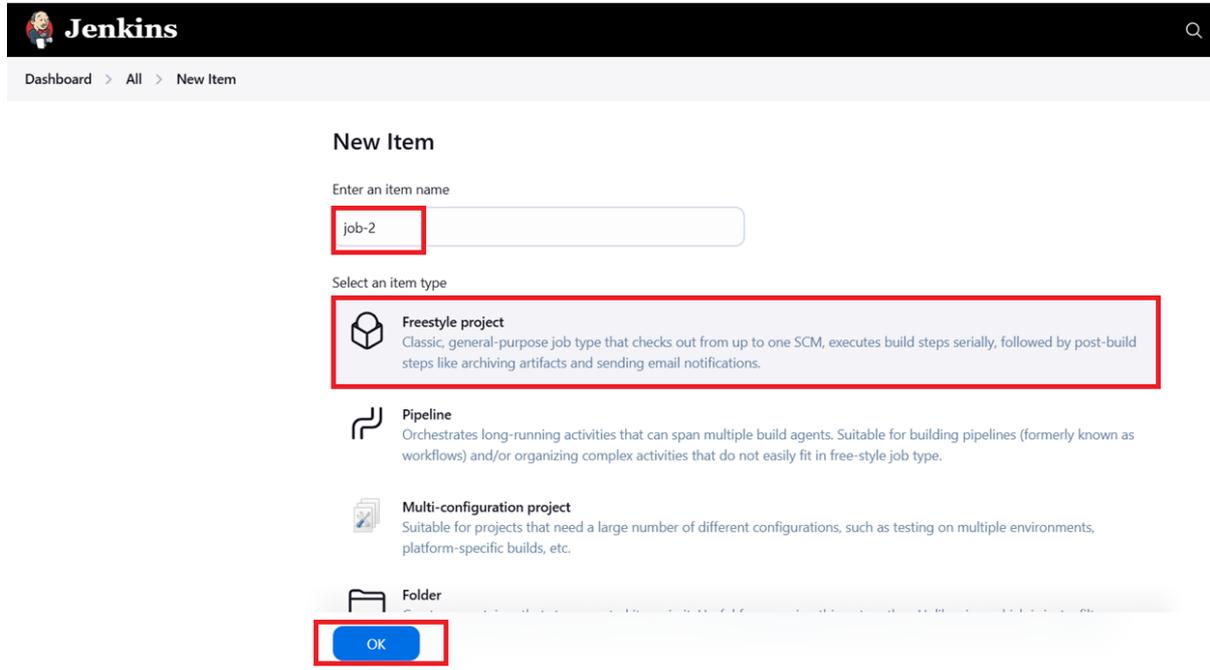
Go to `cd /var/lib/Jenkins/workspaces/job-1`.
You will see the Job-1 has been created the jobs.

```
ubuntu@ip-172-31-95-109:~$ cd /var/lib/jenkins/workspace/job-1
ubuntu@ip-172-31-95-109:/var/lib/jenkins/workspace/job-1$ ll
total 24
drwxr-xr-x 4 jenkins jenkins 4096 Mar 22 10:37 ./
drwxr-xr-x 3 jenkins jenkins 4096 Mar 22 10:37 ../
drwxr-xr-x 8 jenkins jenkins 4096 Mar 22 11:13 .git/
-rw-r--r-- 1 jenkins jenkins  118 Mar 22 10:37 Dockerfile
drwxr-xr-x 2 jenkins jenkins 4096 Mar 22 10:37 images/
-rw-r--r-- 1 jenkins jenkins  193 Mar 22 10:37 index.html
ubuntu@ip-172-31-95-109:/var/lib/jenkins/workspace/job-1$ █
```

Go To Jenkins Url.We need to Create Jobs

Job2: test

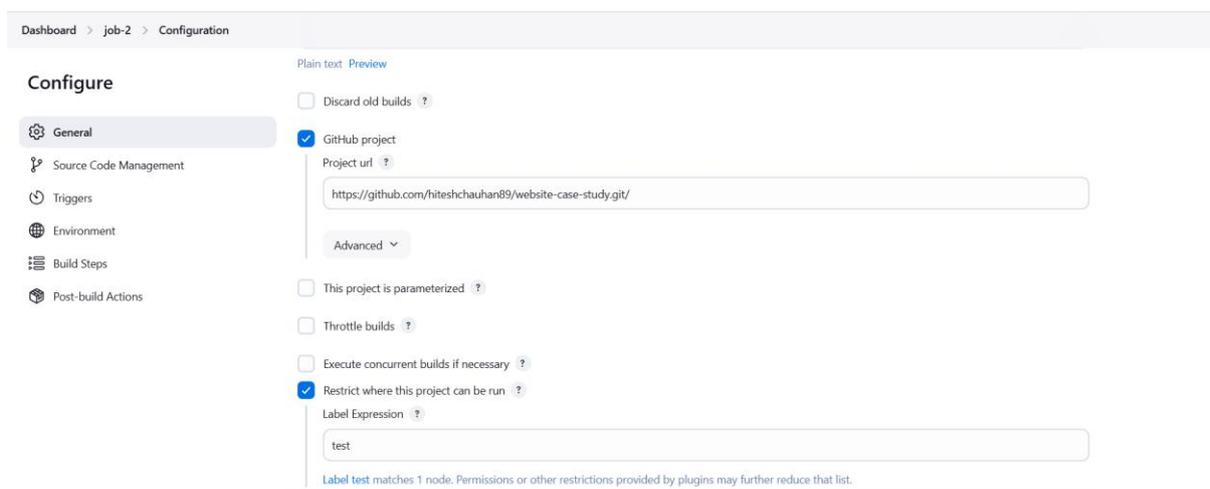
Click **Dashboard >All>New Item** Then
Click **OK**

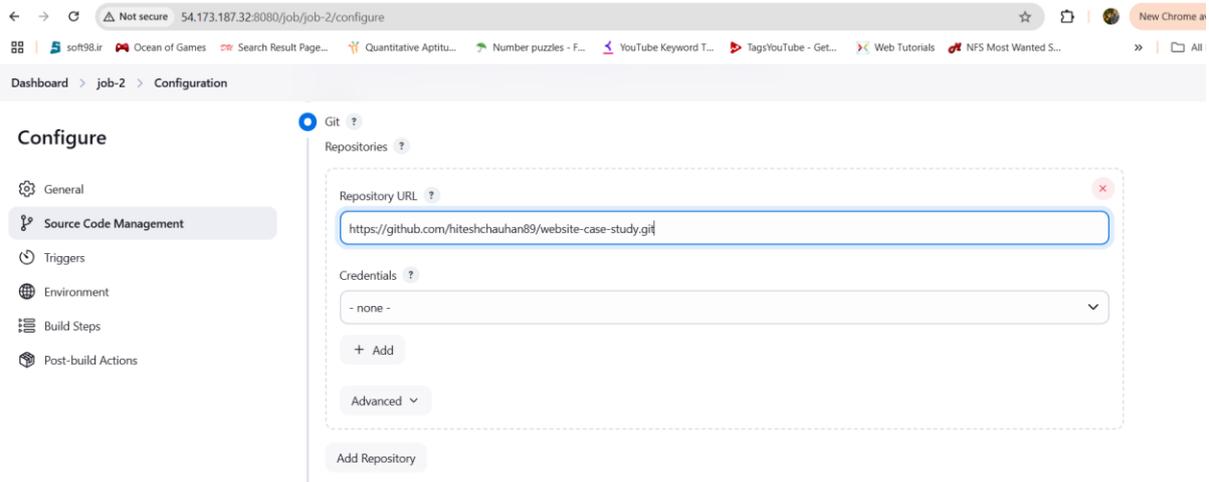


After You Need to choose GitHub Projects Url.Copy the Url we have added in forks.

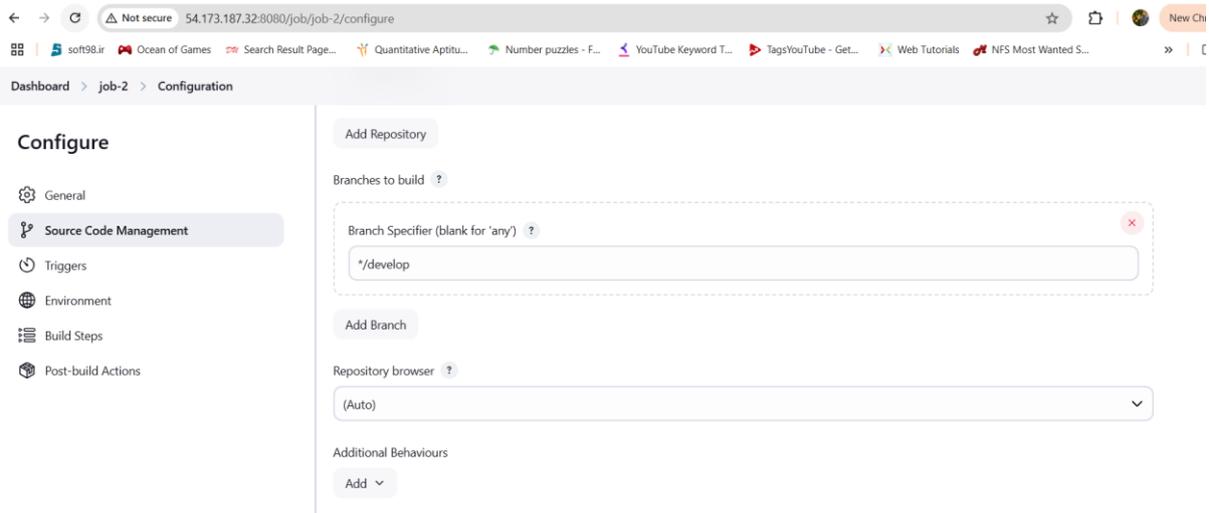
Project Url:- <https://github.com/hiteshchauhan89/website-case-study.git>

Check Box Restrict Where This Project can be run so mentioned **test**



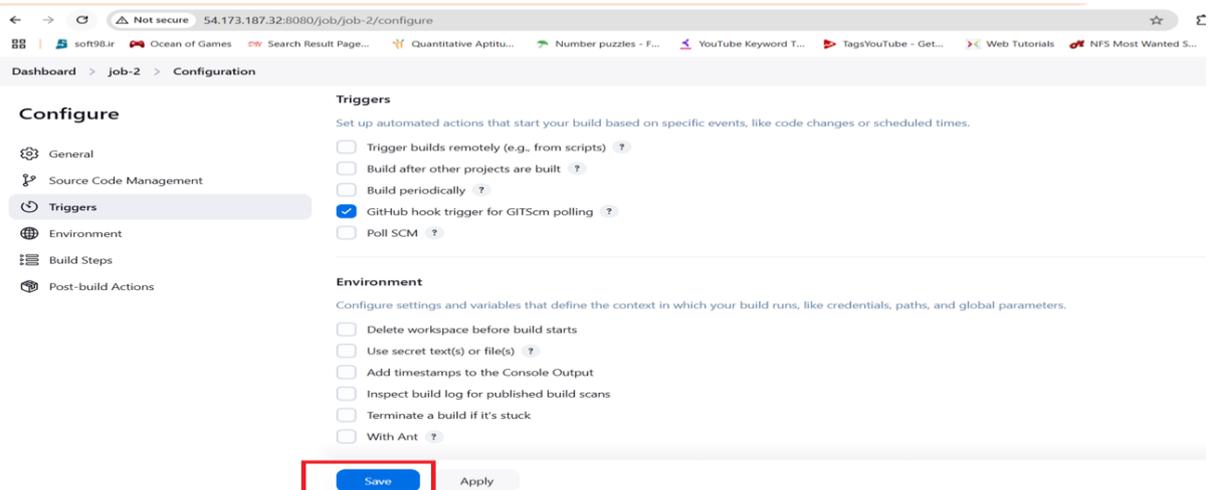


Now We need to change the Branch here mentioned ***/develop**

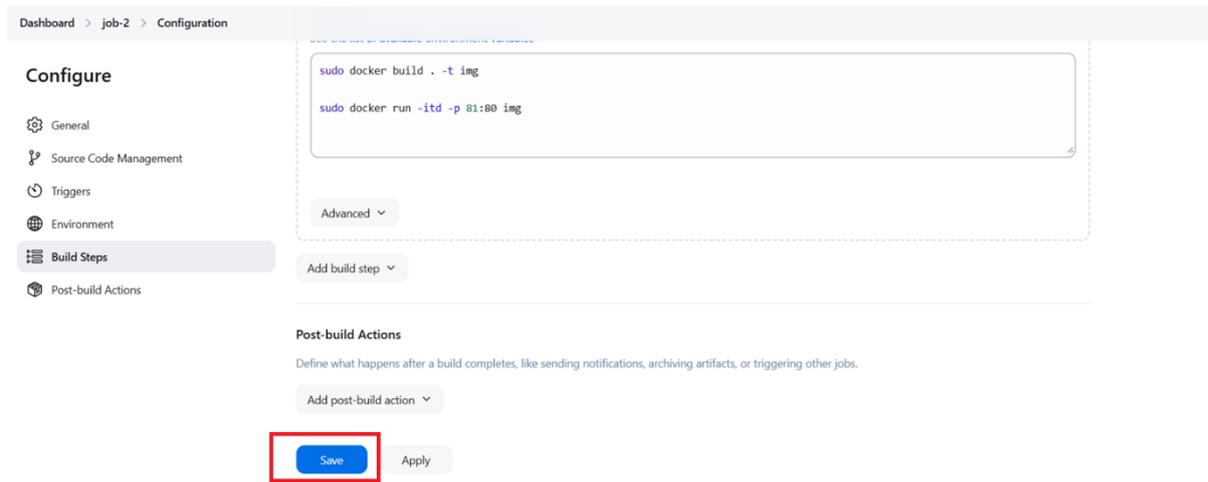


Triggers

We need trigger this url through webhook. we have already added this webhook url in Job-1.



After there is Build Steps
Click **Build Steps>Execute Shell**.
Here we need to execute command.
sudo docker build . -t img
sudo docker run -itd -p 81:80 img



Dashboard > job-2 > Configuration

Configure

- General
- Source Code Management
- Triggers
- Environment
- Build Steps**
- Post-build Actions

```
sudo docker build . -t img
sudo docker run -itd -p 81:80 img
```

Advanced ▾

Add build step ▾

Post-build Actions

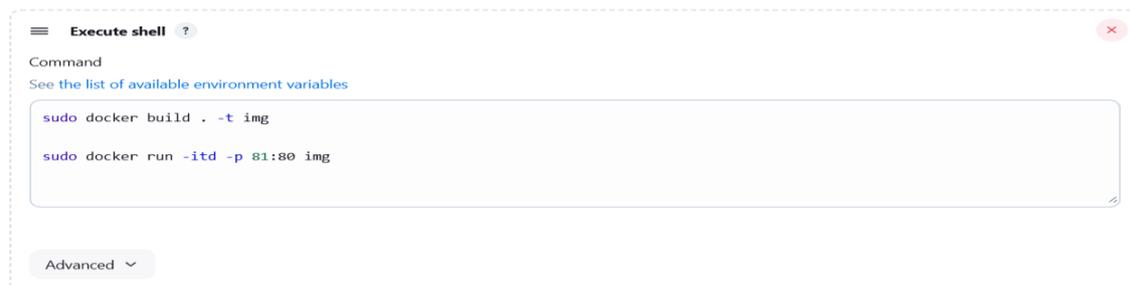
Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

Add post-build action ▾

Save Apply

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.



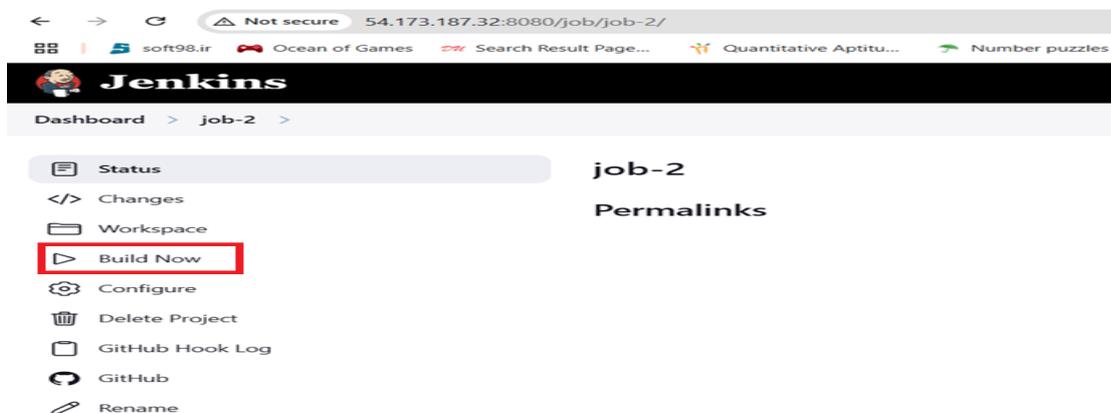
Execute shell ?

Command

See [the list of available environment variables](#)

```
sudo docker build . -t img
sudo docker run -itd -p 81:80 img
```

Advanced ▾



Not secure 54.173.187.32:8080/job/job-2/

soft98.ir Ocean of Games Search Result Page... Quantitative Aptitu... Number puzzles

Jenkins

Dashboard > job-2 >

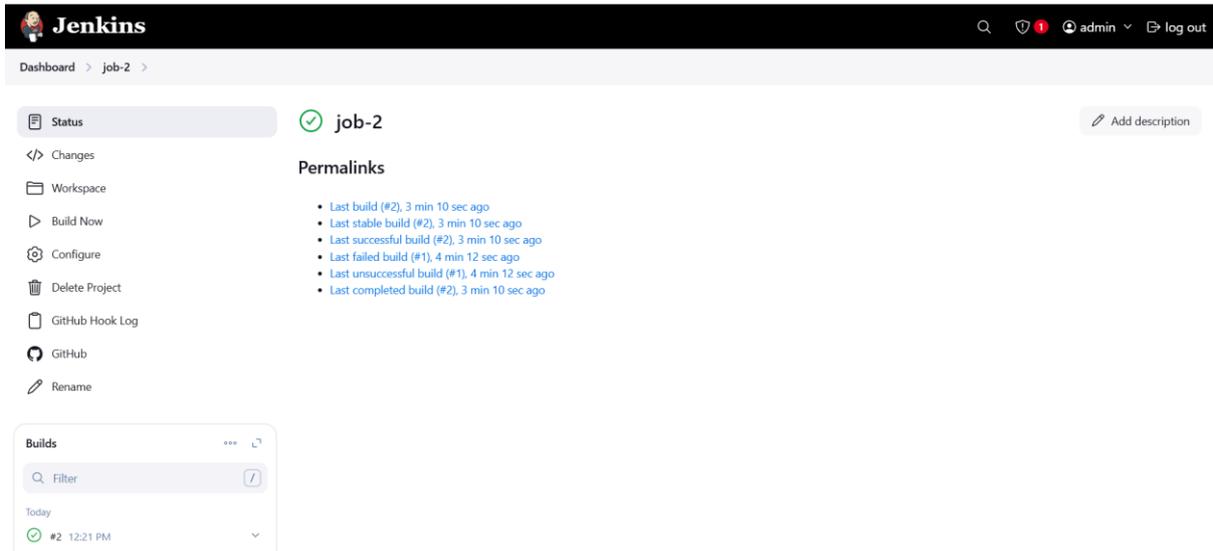
job-2

Permalinks

- Status
- Changes
- Workspace
- Build Now**
- Configure
- Delete Project
- GitHub Hook Log
- GitHub
- Rename

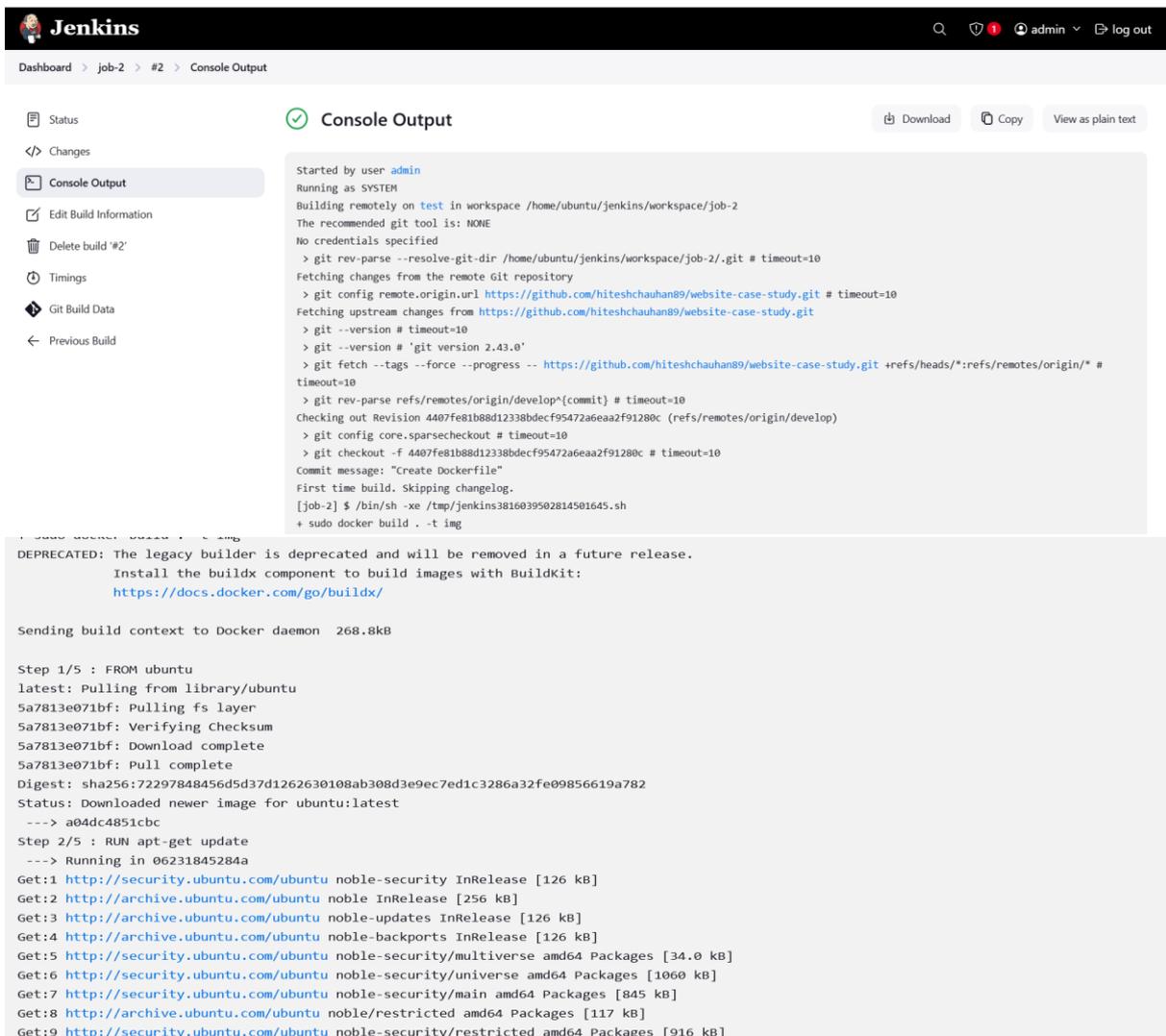
Then Click Build Now.

Now Job-2 Has been successfully Created.



The screenshot shows the Jenkins dashboard for a job named 'job-2'. The top navigation bar includes the Jenkins logo, a search icon, a shield icon with a red exclamation mark, a user profile icon for 'admin', and a 'log out' button. Below the navigation bar, the breadcrumb path is 'Dashboard > job-2 >'. The main content area is divided into two columns. The left column contains a sidebar with the following items: 'Status' (selected), 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Project', 'GitHub Hook Log', 'GitHub', and 'Rename'. The right column shows the job's status as 'job-2' with a green checkmark icon and an 'Add description' button. Below the status, there is a 'Permalinks' section with a list of build links: 'Last build (#2), 3 min 10 sec ago', 'Last stable build (#2), 3 min 10 sec ago', 'Last successful build (#2), 3 min 10 sec ago', 'Last failed build (#1), 4 min 12 sec ago', 'Last unsuccessful build (#1), 4 min 12 sec ago', and 'Last completed build (#2), 3 min 10 sec ago'. At the bottom left, there is a 'Builds' section with a search filter and a dropdown menu showing 'Today' and a single build entry: '#2 12:21 PM' with a green checkmark icon.

This is the output of job-2.



The screenshot shows the Jenkins console output for job-2. The top navigation bar is identical to the previous screenshot. The breadcrumb path is 'Dashboard > job-2 > #2 > Console Output'. The main content area is divided into two columns. The left column contains a sidebar with the following items: 'Status', 'Changes', 'Console Output' (selected), 'Edit Build Information', 'Delete build '#2'', 'Timings', 'Git Build Data', and 'Previous Build'. The right column shows the console output with a green checkmark icon and a 'Console Output' title. Above the output, there are buttons for 'Download', 'Copy', and 'View as plain text'. The console output text is as follows:

```
Started by user admin
Running as SYSTEM
Building remotely on test in workspace /home/ubuntu/jenkins/workspace/job-2
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /home/ubuntu/jenkins/workspace/job-2/.git # timeout=10
Fetching changes from the remote git repository
> git config remote.origin.url https://github.com/hiteshchauhan89/website-case-study.git # timeout=10
Fetching upstream changes from https://github.com/hiteshchauhan89/website-case-study.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/hiteshchauhan89/website-case-study.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
> git rev-parse refs/remotes/origin/develop^[commit] # timeout=10
Checking out Revision 4407fe81b88d12338bdecf95472a6eaa2f91280c (refs/remotes/origin/develop)
> git config core.sparsecheckout # timeout=10
> git checkout -f 4407fe81b88d12338bdecf95472a6eaa2f91280c # timeout=10
Commit message: "Create Dockerfile"
First time build. Skipping changelog.
[job-2] $ /bin/sh -xe /tmp/jenkins3816039502814501645.sh
+ sudo docker build . -t img

DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 268.8kB

Step 1/5 : FROM ubuntu
latest: Pulling from library/ubuntu
5a7813e071bf: Pulling fs layer
5a7813e071bf: Verifying Checksum
5a7813e071bf: Download complete
5a7813e071bf: Pull complete
Digest: sha256:72297848456d5d37d1262630108ab308d3e9ec7ed1c3286a32fe09856619a782
Status: Downloaded newer image for ubuntu:latest
--> a04dc4851cbc
Step 2/5 : RUN apt-get update
--> Running in 06231845284a
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:5 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [34.0 kB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1060 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [845 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [916 kB]
```

```

Get:10 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:12 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1171 kB]
Get:14 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1351 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [38.7 kB]
Get:16 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [962 kB]
Get:17 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [48.0 kB]
Get:18 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [30.9 kB]
Fetched 28.7 MB in 2s (13.4 MB/s)
Reading package lists...
--> Removed intermediate container 06231845284a
--> 8ae5dd4d6c21
Step 3/5 : RUN apt-get install apache2 -y
--> Running in 08ae5b6169e2
Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  adduser apache2-bin apache2-data apache2-utils ca-certificates krb5-locale
  libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64
  libbrotli1 libcurl4t64 libexpat1 libgdbm-compat4t64 libgdbm6t64
  libgssapi-krb5-2 libicu74 libjansson4 libk5crypto3 libkeyutils1 libkrb5-3
  libkrb5support0 libldap-common libldap2 liblua5.4-0 libnghttp2-14
  libperl5.38t64 libpsl5t64 librtmp1 libsasl2-2 libsasl2-modules
  libsasl2-modules-db libsasl2-modules-gssapi-mit
  libsasl2-modules-gssapi-heimdal libsasl2-modules-ldap libsasl2-modules-otp
  libsasl2-modules-sql perl-doc libterm-readline-gnu-perl
  | libterm-readline-perl-perl make libtap-harness-archive-perl
The following NEW packages will be installed:
  adduser apache2 apache2-bin apache2-data apache2-utils ca-certificates
  krb5-locale libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  libaprutil1t64 libbrotli1 libcurl4t64 libexpat1 libgdbm-compat4t64
  libgdbm6t64 libgssapi-krb5-2 libicu74 libjansson4 libk5crypto3 libkeyutils1
  libkrb5-3 libkrb5support0 libldap-common libldap2 liblua5.4-0 libnghttp2-14
  libperl5.38t64 libpsl5t64 librtmp1 libsasl2-2 libsasl2-modules
  libsasl2-modules-db libsasl2-modules-gssapi-mit
  libsasl2-modules-gssapi-heimdal libsasl2-modules-ldap
  libsasl2-modules-otp libsasl2-modules-sql perl-doc libterm-readline-gnu-perl
  | libterm-readline-perl-perl make libtap-harness-archive-perl
The following packages will be upgraded:
  libssl3t64 perl-base
2 upgraded, 43 newly installed, 0 to remove and 16 not upgraded.
Need to get 30.0 MB of archives.
After this operation, 109 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 perl-base amd64 5.38.2-3.2build2.1 [1823 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 perl-modules-5.38 all 5.38.2-3.2build2.1 [3110 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble/main amd64 libgdbm6t64 amd64 1.23-5.1build1 [34.4 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble/main amd64 libgdbm-compat4t64 amd64 1.23-5.1build1 [6710 B]
Get:5 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libperl5.38t64 amd64 5.38.2-3.2build2.1 [4867 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 perl amd64 5.38.2-3.2build2.1 [231 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libapr1t64 amd64 1.7.2-3.1ubuntu0.1 [108 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libexpat1 amd64 2.6.1-2ubuntu0.2 [87.4 kB]

```

Dashboard > job-2 > #2 > Console Output

```

liblocale-gettext-perl cron quota ecryptfs-utils apache2-doc
apache2-suexec-pristine | apache2-suexec-custom www-browser ufw gdbm-l10n
krb5-doc krb5-user libsasl2-modules-gssapi-mit
| libsasl2-modules-gssapi-heimdal libsasl2-modules-ldap libsasl2-modules-otp
libsasl2-modules-sql perl-doc libterm-readline-gnu-perl
| libterm-readline-perl-perl make libtap-harness-archive-perl
The following NEW packages will be installed:
  adduser apache2 apache2-bin apache2-data apache2-utils ca-certificates
  krb5-locale libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  libaprutil1t64 libbrotli1 libcurl4t64 libexpat1 libgdbm-compat4t64
  libgdbm6t64 libgssapi-krb5-2 libicu74 libjansson4 libk5crypto3 libkeyutils1
  libkrb5-3 libkrb5support0 libldap-common libldap2 liblua5.4-0 libnghttp2-14
  libperl5.38t64 libpsl5t64 librtmp1 libsasl2-2 libsasl2-modules
  libsasl2-modules-db libsasl2-modules-gssapi-mit
  libsasl2-modules-gssapi-heimdal libsasl2-modules-ldap
  libsasl2-modules-otp libsasl2-modules-sql perl-doc libterm-readline-gnu-perl
  | libterm-readline-perl-perl make libtap-harness-archive-perl
The following packages will be upgraded:
  libssl3t64 perl-base
2 upgraded, 43 newly installed, 0 to remove and 16 not upgraded.
Need to get 30.0 MB of archives.
After this operation, 109 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 perl-base amd64 5.38.2-3.2build2.1 [1823 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 perl-modules-5.38 all 5.38.2-3.2build2.1 [3110 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble/main amd64 libgdbm6t64 amd64 1.23-5.1build1 [34.4 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble/main amd64 libgdbm-compat4t64 amd64 1.23-5.1build1 [6710 B]
Get:5 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libperl5.38t64 amd64 5.38.2-3.2build2.1 [4867 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 perl amd64 5.38.2-3.2build2.1 [231 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libapr1t64 amd64 1.7.2-3.1ubuntu0.1 [108 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libexpat1 amd64 2.6.1-2ubuntu0.2 [87.4 kB]

```

Dashboard > job-2 > #2 > Console Output

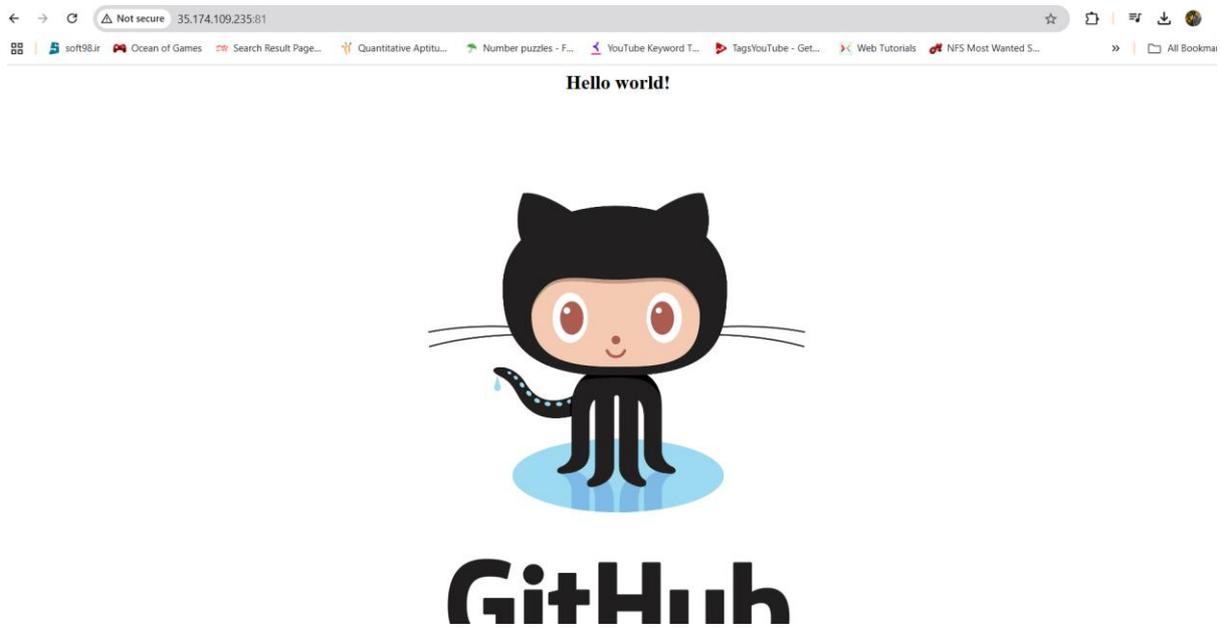
```

Enabling site 000-default.
invoke-rc.d: could not determine current runlevel
invoke-rc.d: policy-rc.d denied execution of start.
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Processing triggers for ca-certificates (20240203) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
--> Removed intermediate container 08ae5b6169e2
--> 0ae7cd346a24
Step 4/5 : ADD . /var/www/html/
--> c6e40c501b66
Step 5/5 : ENTRYPOINT apachectl -D FOREGROUND
--> Running in 22f891e0b16c
--> Removed intermediate container 22f891e0b16c
--> 284fbd145e98
Successfully built 284fbd145e98
Successfully tagged img:latest
+ sudo docker run -itd -p 81:80 img
5560223ff05ea382442ce1456139d8d4025d079b2cad7a5a15e9a961a21f1b7
Finished: SUCCESS

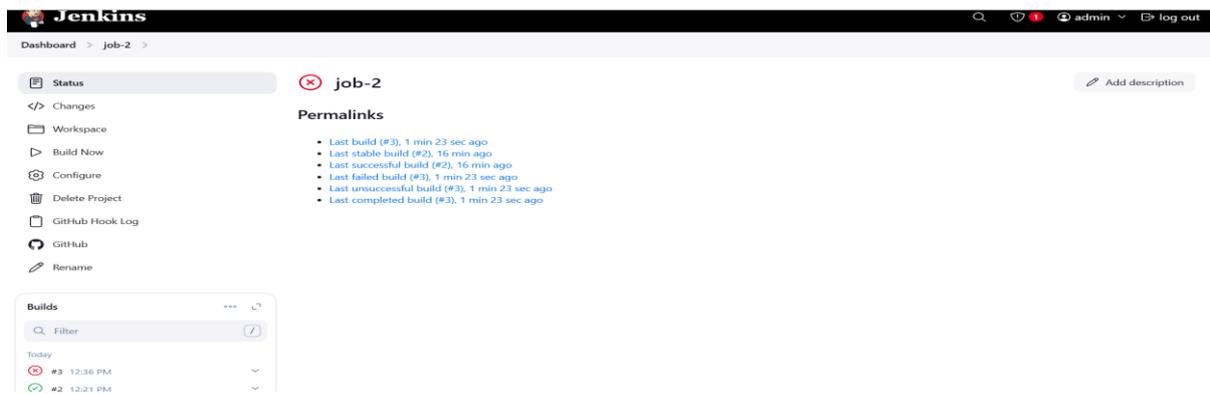
```

After Deployed the Job-2.

Copy the public ip from test machine.you will the output with port no 81.



Now In this job 2.you will again getting the error when you build the images.



```
Dashboard > job-2 > #3 > Console Output

Step 1/5 : FROM ubuntu
--> a04dc4851cbc
Step 2/5 : RUN apt-get update
--> Using cache
--> 8ae5dd4d6c21
Step 3/5 : RUN apt-get install apache2 -y
--> Using cache
--> 0ae7cd346a24
Step 4/5 : ADD . /var/www/html/
--> Using cache
--> c6e40c501b66
Step 5/5 : ENTRYPOINT apachectl -D FOREGROUND
--> Using cache
--> 284fbd145e98
Successfully built 284fbd145e98
Successfully tagged img:latest
+ sudo docker run -itd -p 81:80 img
143520b9331b049890d5e0700de3a9e410d8efc4f13fb455b4ea3ac8ae107e8c
docker: Error response from daemon: driver failed programming external connectivity on endpoint magical_black
(666f25ff5ba475deaab02a6e0d3fda42557c71e3ff927a8b7f2289d779fa7a): Bind for 0.0.0.0:81 failed: port is already allocated.
Build step 'Execute shell' marked build as failure
Finished: FAILURE
```

Now We need to some changes in execute shell command.

Sudo docker rm -f hitesh

Sudo docker build . -t project1

Sudo docker run -itd -p 100:80 --name=hitesh project1

After change the execute shell,you need to save the configuration.

Dashboard > job-2 > Configuration

Configure

- General
- Source Code Management
- Triggers
- Environment
- Build Steps**
- Post-build Actions

Execute shell

Command

See the list of available environment variables

```
sudo docker rm -f hitesh
sudo docker build . -t project1
sudo docker run -itd -p 100:80 --name=hitesh project1
```

Advanced

Add build step

Post-build Actions

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

Add post-build action

Save Apply

Now you need to create again build job after some chages.

Jenkins

Dashboard > job-2

Status **job-2** Add description

Changes

Workspace

Build Now

Configure

Delete Project

GitHub Hook Log

GitHub

Rename

Builds

Filter

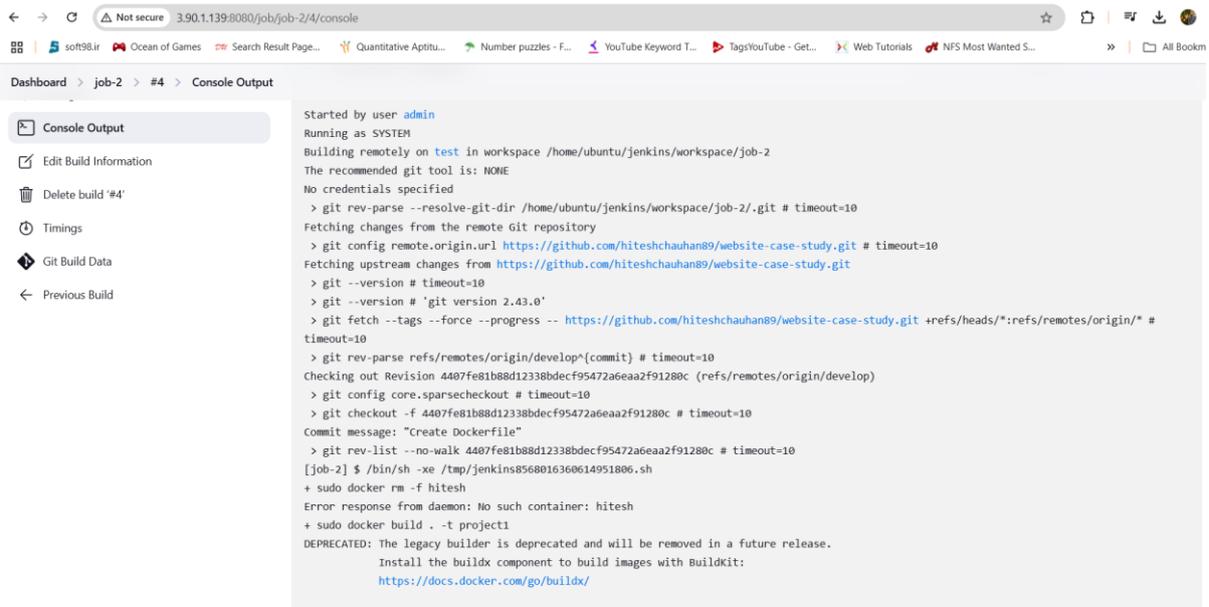
Today

- #4 12:46 PM

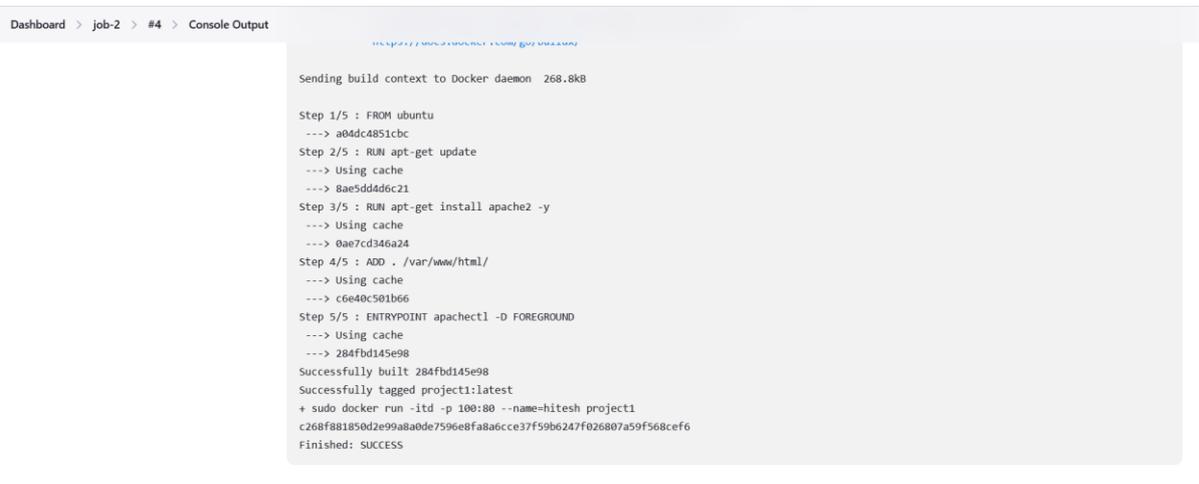
Permalinks

- Last build (#4), 2 min 7 sec ago
- Last stable build (#4), 2 min 7 sec ago
- Last successful build (#4), 2 min 7 sec ago
- Last failed build (#3), 12 min ago
- Last unsuccessful build (#3), 12 min ago
- Last completed build (#4), 2 min 7 sec ago

Now You can see the expected output from job-2.



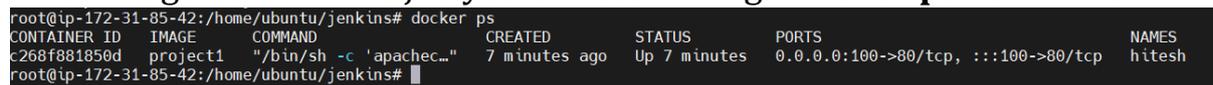
```
Started by user admin
Running as SYSTEM
Building remotely on test in workspace /home/ubuntu/jenkins/workspace/job-2
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /home/ubuntu/jenkins/workspace/job-2/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/hiteshchauhan89/website-case-study.git # timeout=10
Fetching upstream changes from https://github.com/hiteshchauhan89/website-case-study.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/hiteshchauhan89/website-case-study.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
> git rev-parse refs/remotes/origin/develop^{commit} # timeout=10
Checking out Revision 4407fe81b88d12338bdecf95472a6eaa2f91280c (refs/remotes/origin/develop)
> git config core.sparsecheckout # timeout=10
> git checkout -f 4407fe81b88d12338bdecf95472a6eaa2f91280c # timeout=10
Commit message: "Create Dockerfile"
> git rev-list --no-walk 4407fe81b88d12338bdecf95472a6eaa2f91280c # timeout=10
[job-2] $ /bin/sh -xe /tmp/jenkins8568016360614951806.sh
+ sudo docker rm -f hitesh
Error response from daemon: No such container: hitesh
+ sudo docker build . -t project1
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/
```



```
Sending build context to Docker daemon 268.8kB

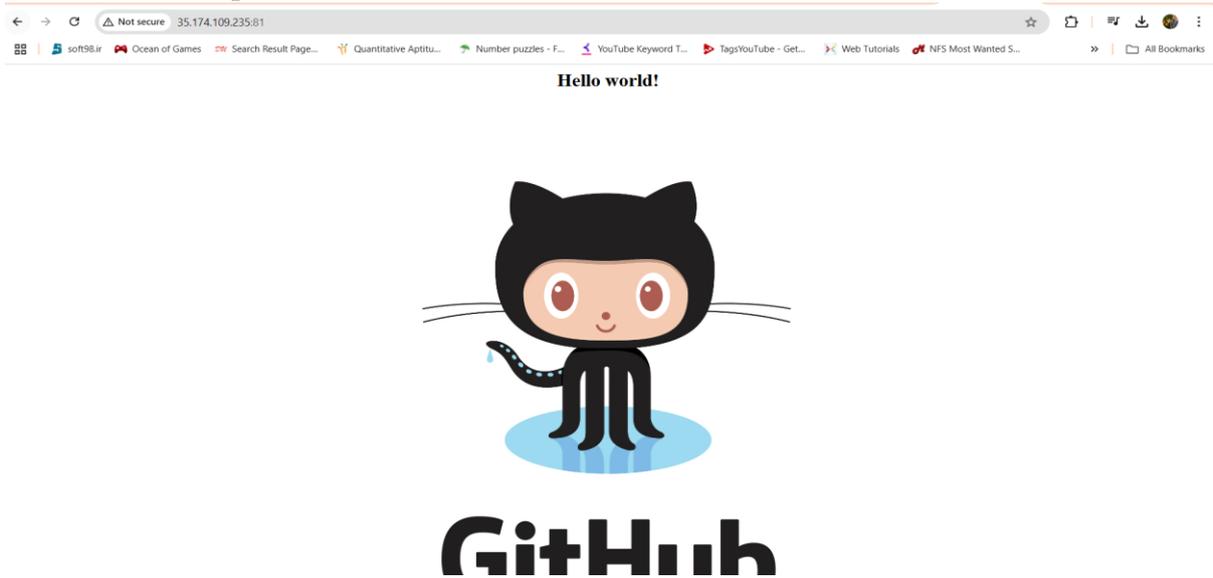
Step 1/5 : FROM ubuntu
--> a04dc4851cbc
Step 2/5 : RUN apt-get update
--> Using cache
--> 8ae5dd4d6c21
Step 3/5 : RUN apt-get install apache2 -y
--> Using cache
--> 0ae7cd346a24
Step 4/5 : ADD . /var/www/html/
--> Using cache
--> c6e40c501b66
Step 5/5 : ENTRYPOINT apache2l -D FOREGROUND
--> Using cache
--> 284fbd145e98
Successfully built 284fbd145e98
Successfully tagged project1:latest
+ sudo docker run -itd -p 100:80 --name=hitesh project1
c268f881850d2e99a8a0de7596e8fa8a6cce37f59b6247f026807a59f568cef6
Finished: SUCCESS
```

After Image build in this job,you can see through **docker ps** command.



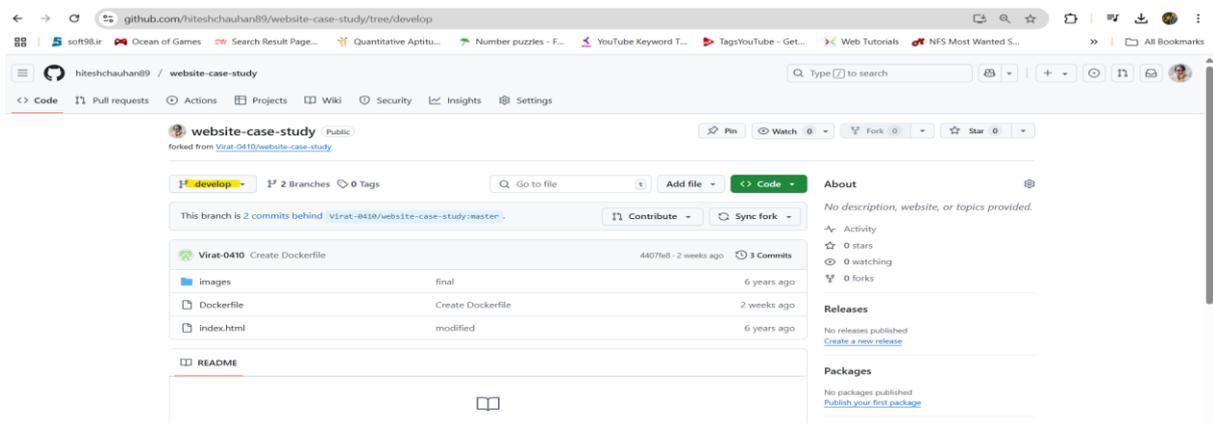
```
root@ip-172-31-85-42:/home/ubuntu/jenkins# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
c268f881850d   project1  "/bin/sh -c 'apachec..." 7 minutes ago  Up 7 minutes  0.0.0.0:100->80/tcp, :::100->80/tcp  hitesh
root@ip-172-31-85-42:/home/ubuntu/jenkins#
```

This is the output

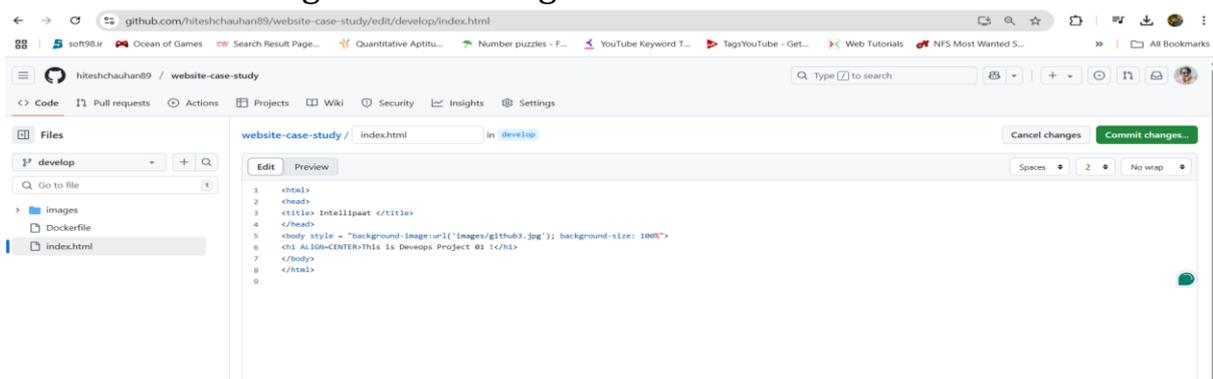


Now In this repository you do some changes in same repositories.

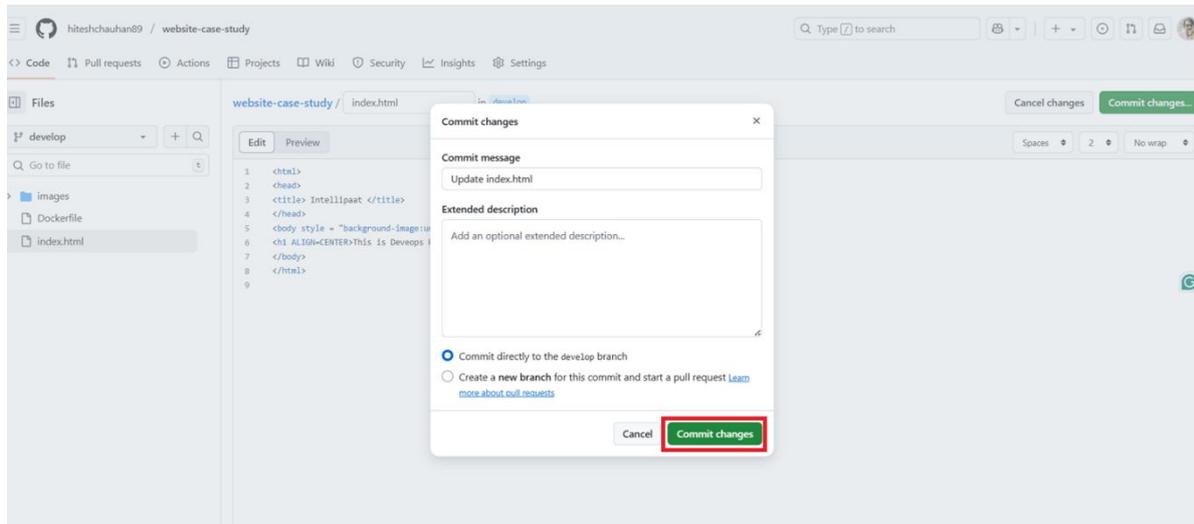
For example I need to change some index.html file.



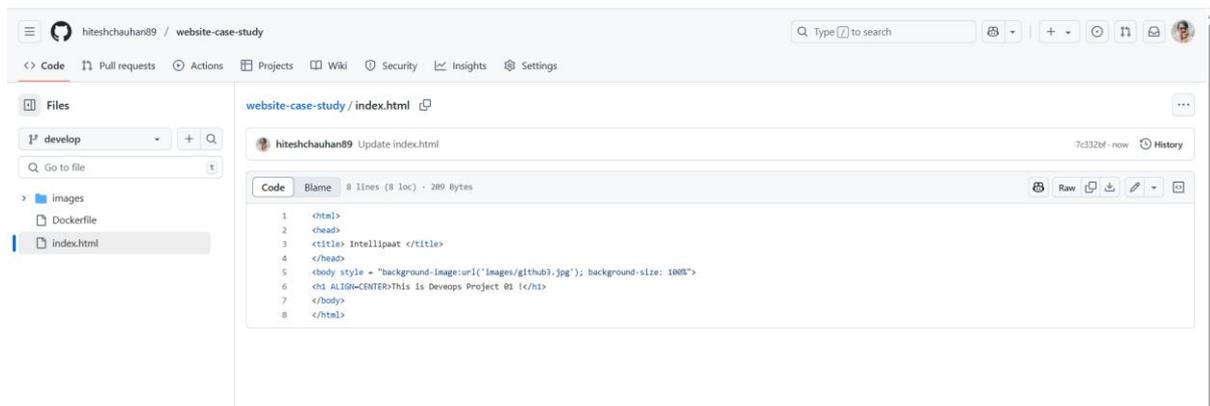
Now we have changed some changed in index.html.



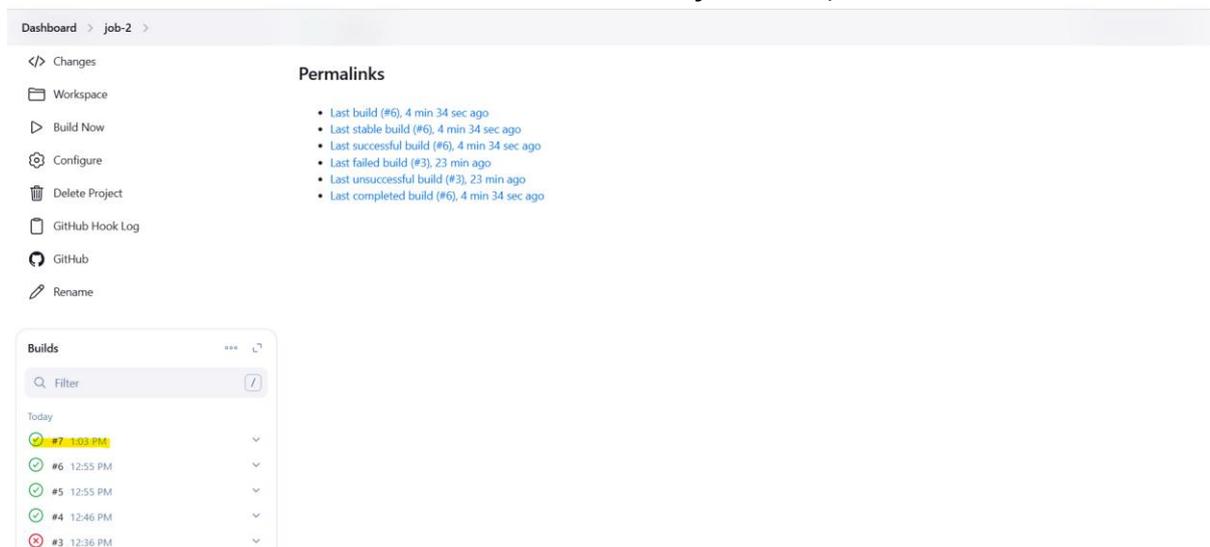
Now Commit this changes through github.



Click Commit Changes.



Now You can see Task no 07 is automatically run in Jenkins.



This is out put with port no 81.

This is Deveops Project 01 !



GitHuh

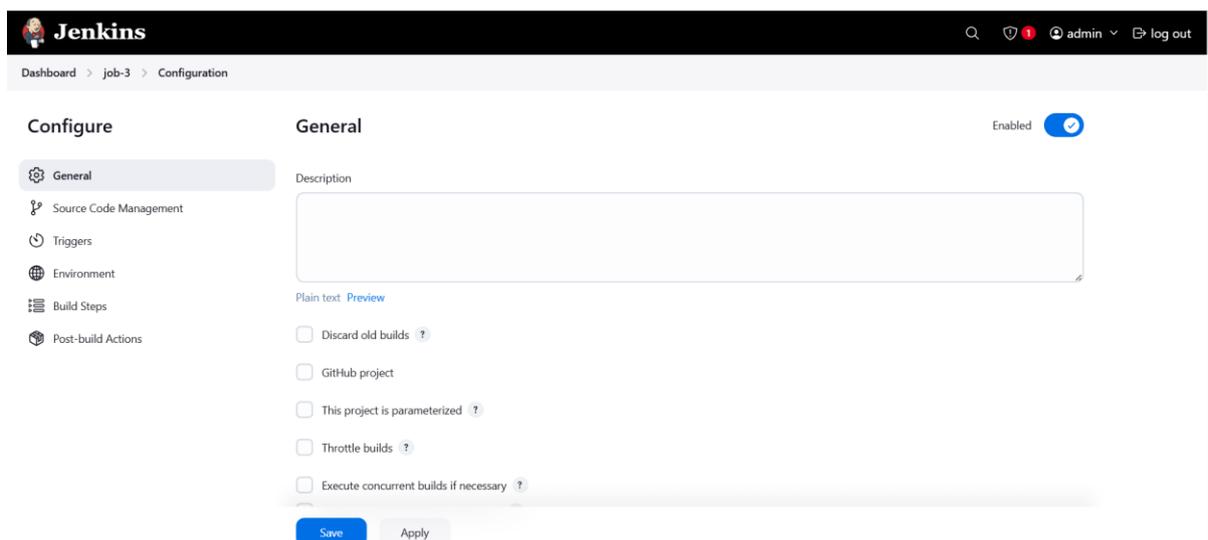
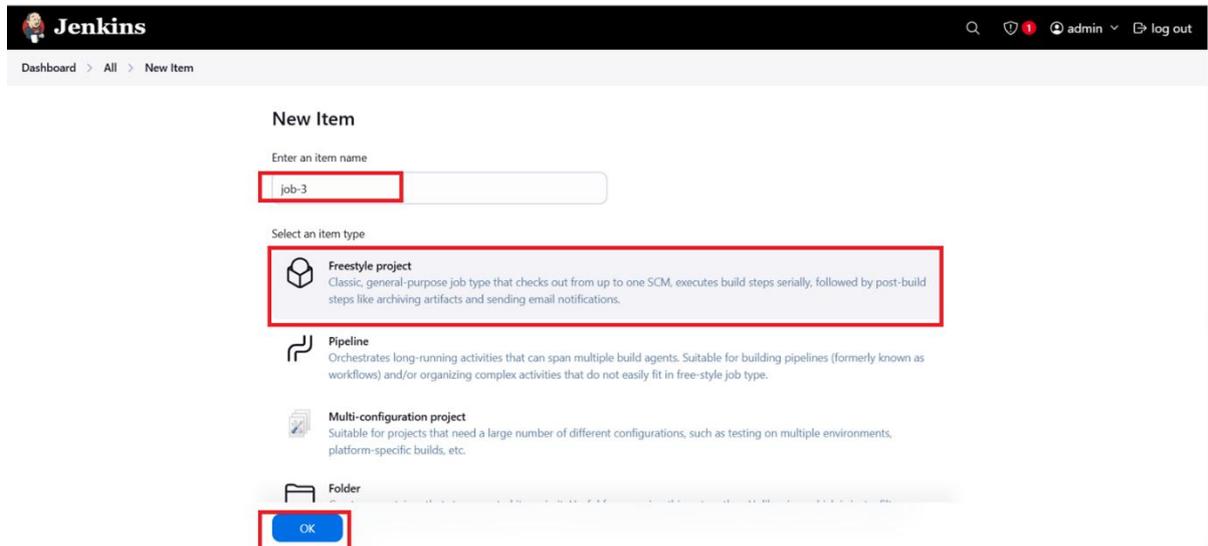
Job3 : prod

Go To Jenkins Url.We need to Create Jobs

JOB-3

Click **Dashboard >All>New Item** Then

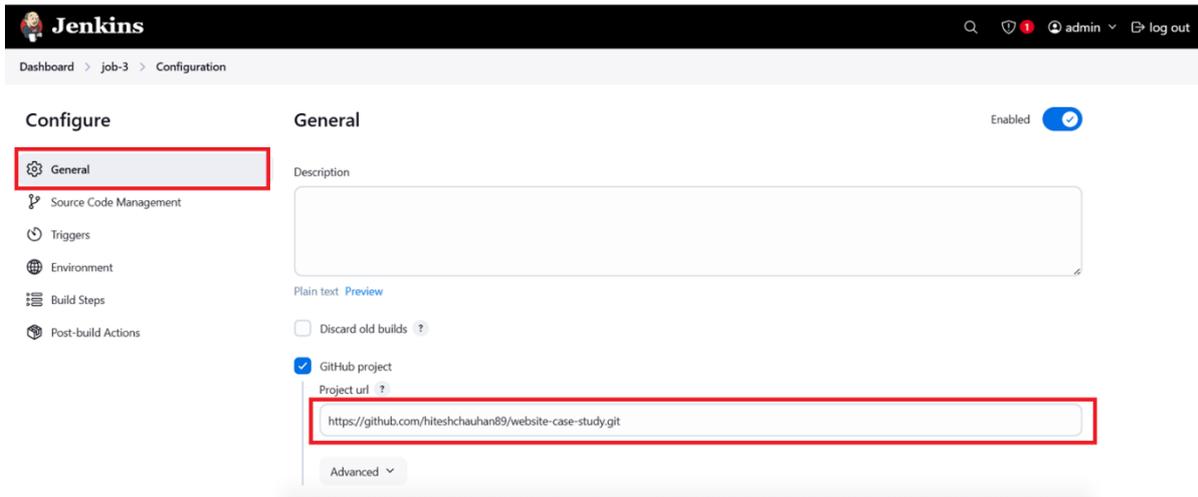
Click OK



After You Need to choose GitHub Projects Url.Copy the Url we have added in forks.

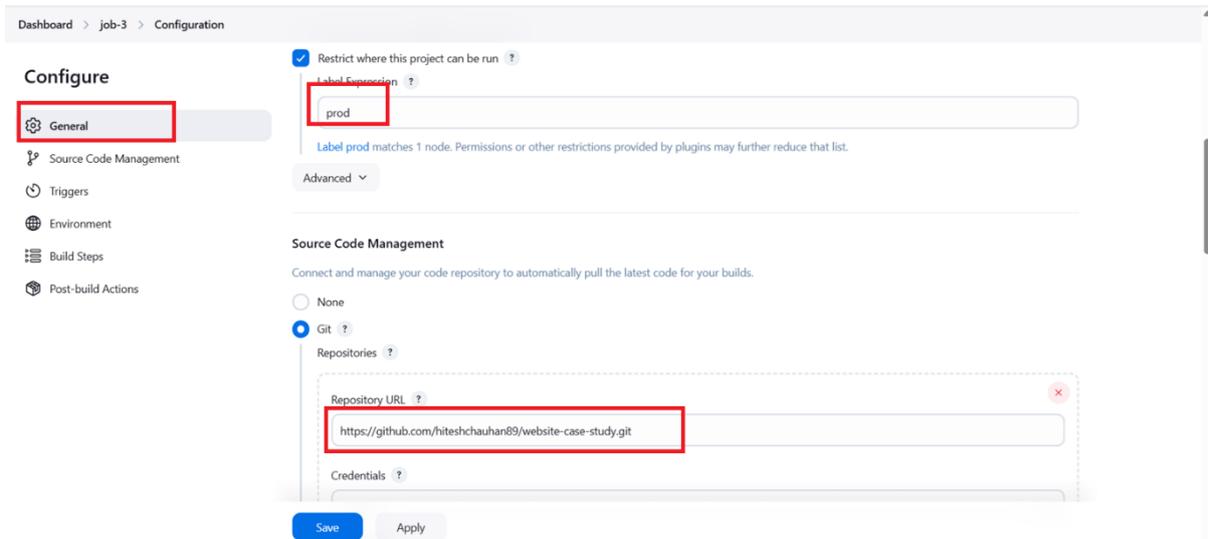
Project Url:- <https://github.com/hiteshchauhan89/website-case-study.git>

Check Box Restrict Where This Project can be run so mentioned **prod**



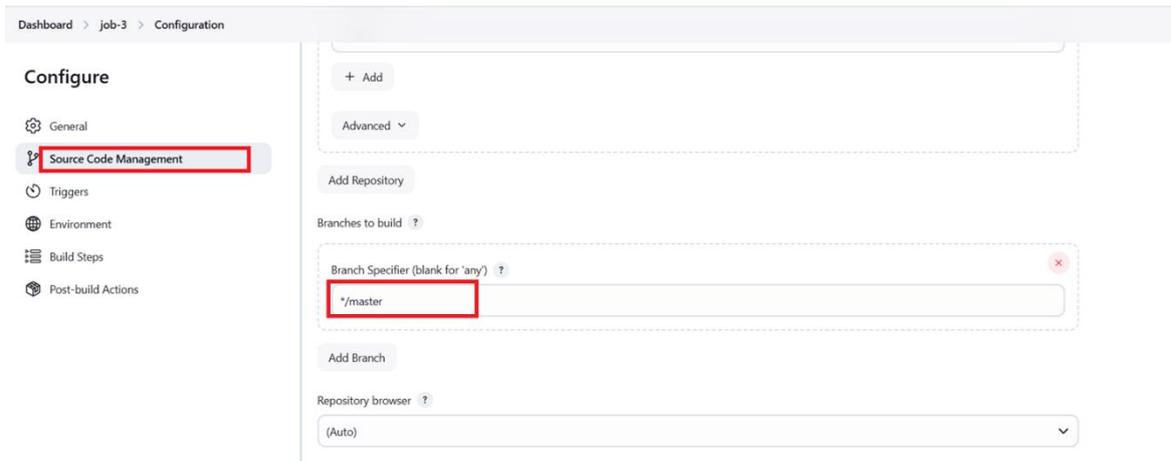
The screenshot shows the Jenkins configuration page for a job named 'job-3'. The 'General' tab is selected and highlighted with a red box. The 'Project url' field is also highlighted with a red box and contains the text 'https://github.com/hiteshchauhan89/website-case-study.git'. The 'Discard old builds' checkbox is unchecked, and the 'GitHub project' checkbox is checked. The 'Enabled' toggle is turned on.

Label Name is prod.



The screenshot shows the 'Source Code Management' tab selected and highlighted with a red box. The 'Restrict where this project can be run' checkbox is checked, and the 'Label Expression' field contains 'prod', also highlighted with a red box. Below this, the 'Git' option is selected under 'Source Code Management', and the 'Repository URL' field contains 'https://github.com/hiteshchauhan89/website-case-study.git', highlighted with a red box. 'Save' and 'Apply' buttons are visible at the bottom.

Source Code Management in Branch should master ***/master**



The screenshot shows the 'Branches to build' section. The 'Branch Specifier (blank for 'any')' field contains '*/master', highlighted with a red box. The 'Repository browser' dropdown is set to '(Auto)'.

In Execute Shell We need to use same command which is already used in job-2.

The screenshot shows the Jenkins Configuration page for job-3. The left sidebar has 'Build Steps' highlighted with a red box. The main content area shows the 'Execute shell' step configuration, with the 'Command' field containing the following text, also highlighted with a red box:

```
sudo docker rm -f hitesh  
sudo docker build . -t hello1  
sudo docker run -itd -p 100:80 --name=hitesh hello1
```

Below the command field, there is an 'Advanced' dropdown menu and an 'Add build step' button. At the bottom of the configuration area, there is a 'Post-build Actions' section with an 'Add post-build action' button. At the very bottom, there are 'Save' and 'Apply' buttons, with the 'Save' button highlighted by a red box.

Build Now

The screenshot shows the Jenkins job-3 page. At the top, there is a black header with the Jenkins logo and the word 'Jenkins'. Below this is a breadcrumb trail: 'Dashboard > job-3 >'. The main content area is divided into two columns. The left column contains a list of actions: 'Status', 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Project', 'GitHub Hook Log', 'GitHub', and 'Rename'. The 'Build Now' button is highlighted with a red box. The right column contains the text 'job-3' and 'Permalinks'.

Job-3 creating.

← → ↻ Not secure 3.90.1.139:8080/job/job-3/

soft98.ir Ocean of Games Search Result Page... Quantitative Aptitu... Number puzzles - F... YouTube Keywo

Jenkins

Dashboard > job-3 >

- Status
- Changes
- Workspace
- Build Now
- Configure
- Delete Project
- GitHub Hook Log
- GitHub
- Rename

job-3

Permalinks

- Last build (#1), 15 sec ago

Builds ⋮ ↗

Filter /

Today

⋮ #1 1:20 PM ⏏ ⌵

Here Job-3 is created.

← → ↻ Not secure 3.90.1.139:8080/job/job-3/

soft98.ir Ocean of Games Search Result Page... Quantitative Aptitu... Number puzzles - F... YouTube Keywo

Dashboard > job-3 >

- Status
- Changes
- Workspace
- Build Now
- Configure
- Delete Project
- GitHub Hook Log
- GitHub
- Rename

job-3

Permalinks

- Last build (#1), 15 sec ago

Builds ⋮ ↗

Filter /

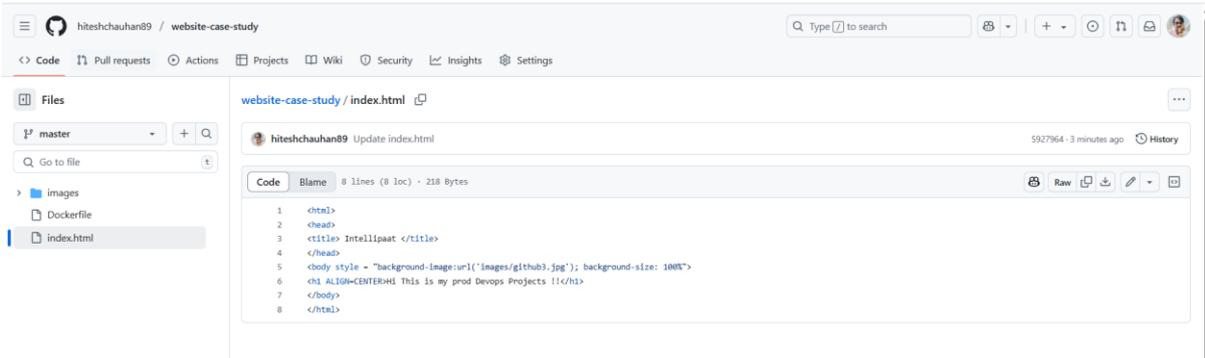
Today

✔ #1 1:20 PM ⌵

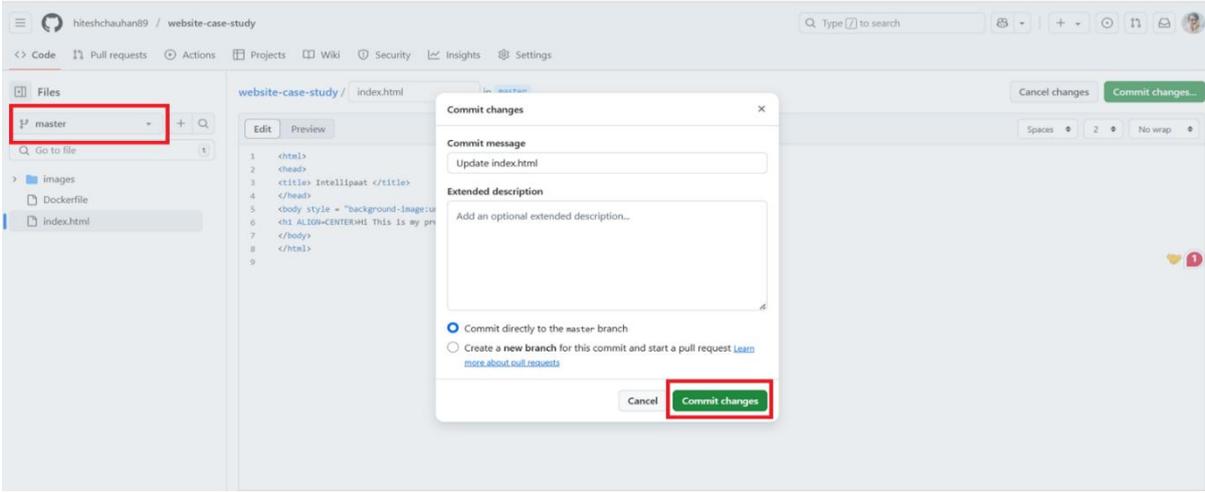

```
Dashboard > job-3 > #1 > Console Output
#####
Enabling site 000-default.
invoke-rc.d: could not determine current runlevel
invoke-rc.d: policy-rc.d denied execution of start.
Processing triggers for libc-bin (2.39-0ubuntu3) ...
Processing triggers for ca-certificates (20240203) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
--> Removed intermediate container 884f9ce22d4e
--> e2da602db7d2
Step 4/5 : ADD ./var/www/html/
--> 7897973e3ed6
Step 5/5 : ENTRYPOINT apache2ctl -D FOREGROUND
--> Running in aed90781b702
--> Removed intermediate container aed90781b702
--> 4806f692c43c
Successfully built 4806f692c43c
Successfully tagged hello1:latest
+ sudo docker run -itd -p 100:80 --name=hitesh hello1
73d24711020381091b5a475ac99ad54e80878645a1bfd1cedde92754cfd3a35
Finished: SUCCESS
```

REST API Jenkins 2.492.2

Now Here we need to change the some code



Commit changes the code.



Now Here in master branch code has been deployed automatically. The Task no 07 has been automatically created after github master branch code changed.

The screenshot shows a web browser window with a dashboard for a CI/CD pipeline. The address bar shows '3.90.1.139:8080/job-3/'. The dashboard has a sidebar with options: Workspace, Build Now, Configure, Delete Project, GitHub Hook Log, GitHub, and Rename. The main area lists build history:

- Last build (#5), 1 min 16 sec ago
- Last stable build (#1), 9 min 30 sec ago
- Last successful build (#1), 9 min 30 sec ago
- Last failed build (#5), 1 min 16 sec ago
- Last unsuccessful build (#5), 1 min 16 sec ago
- Last completed build (#5), 1 min 16 sec ago

Below this is a 'Builds' section with a filter input and a list of builds for 'Today':

- #7 1:31 PM
- #6 1:30 PM

This is output of Master Branch code changes.

The screenshot shows a web browser window with the address bar '44.211.217.204:100'. The page content includes the text 'Hi This is my prod Devops Projects !!' and the GitHub logo (Octocat) above the word 'GitHub'.

Hi This is my prod Devops Projects !!



GitHub