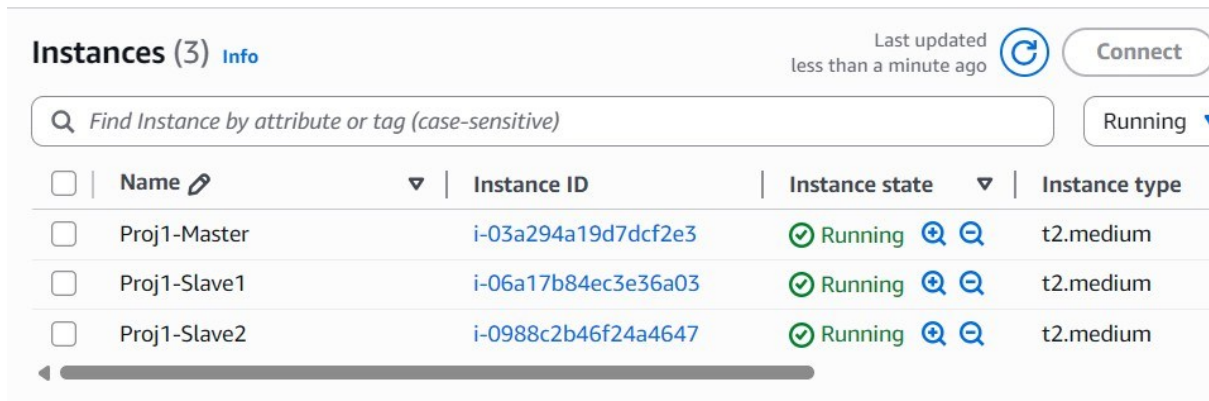


Capstone Project 1:-

Step1:- Launch 3 EC2 Instance – Proj1-Slave1 , Proj1-Slave2, Proj1-Master

Version – Ubuntu 22.0 LTS

t2.medium – default security group



The screenshot displays the AWS Management Console 'Instances' page. It shows three EC2 instances in a 'Running' state. The instances are named 'Proj1-Master', 'Proj1-Slave1', and 'Proj1-Slave2'. All three instances are using the 't2.medium' instance type. The 'Instance ID' for each instance is displayed in blue text. The 'Instance state' column shows a green checkmark and the word 'Running' for each instance. The 'Instance type' column shows 't2.medium' for each instance. The page includes a search bar, a 'Connect' button, and a 'Running' filter dropdown.

	Name	Instance ID	Instance state	Instance type
<input type="checkbox"/>	Proj1-Master	i-03a294a19d7dcf2e3	Running	t2.medium
<input type="checkbox"/>	Proj1-Slave1	i-06a17b84ec3e36a03	Running	t2.medium
<input type="checkbox"/>	Proj1-Slave2	i-0988c2b46f24a4647	Running	t2.medium

Step2:- Connect all three nodes and update it

Sudo apt update -y

Step3:- Install Ansible in master node using ansible documentation.

Create **install.sh** script file and paste commands in script file

Implement the script file using **Bash install.sh**

Step4:- Go in ssh folder – **cd .ssh**

There is one file `authorized_keys`

Generate key - **ssh-keygen**

sudo cat /home/ubuntu/.ssh/id_rsa.pub

Copy the key contain.

```
ubuntu@ip-172-31-26-199:~/ssh$ ssh-keygen
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:gM352OWFTYrQnEQMlwnus1hVW2hVREhcaF7ulg2N15U ubuntu@ip-172-31-26-199
The key's randomart image is:
+---[RSA 3072]-----+
|
| ..+==*O= . .|
| . *o*X..= E.|
| . o.*o.o+o.o .|
| . . =.o+.o .|
| + . S..=|
| o o + .|
| . .|
|
+---+
[SHA256]-----+
ubuntu@ip-172-31-26-199:~/ssh$ sudo cat /home/ubuntu/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCQKu/cLElMrjHATfV5OAX+3wu5r15NHfttZ7CpbEnEYbJVGKobDmc9BtZJyEk+WATg93MukxsInVfLVMkmZkrAydhCZQg3zOc2bqCnhI5hhtJF5sYmJ5scds93T9
1u0raE/VpuMmHAQR2mHEDEl8hFzudKBRAad38EuXX4Wpu5JePuIjWB9mj10+0csFwnYkVgZu8YyBmPLK15GpnsObF10JlUQewcstAo7kvKm/ag/EVChfJ8ye2Eu9CylCaoUo+eoXJB+WOO2Em8zFeox8yZJB5WNTLxd
BwrcXVKRaoIymYrYlbrvRo5G531ptQlnaumiNrvgmtPDDpM22Jj1867cNeyH4xjqr41YentwZtZjhoLUCuxdGHQ7o4QZM5Nm9y95rP92409XrMcPp1rrjVqtesfgtIRGs8VHg4eNlBDNkdmbegm1ctkfwY95JtZ6WJ
0htR1XulouIrf6hewEpp+r4tqmImqPa9o3UBGGd4S9KRqdIE= ubuntu@ip-172-31-26-199
ubuntu@ip-172-31-26-199:~/ssh$ ^[[200-cd /etc/ansible-
```

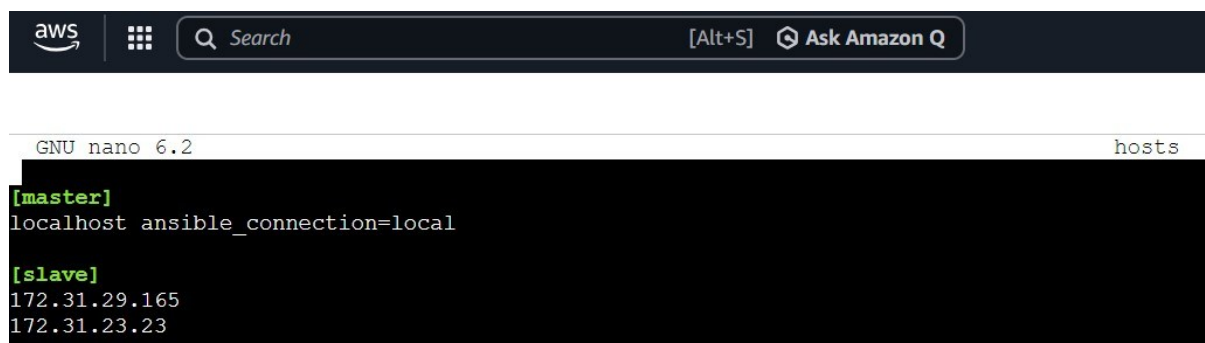
Step5:- Go in Slave1 Node and go inside ssh folder and in authorized_keys folder – **sudo nano authorized_keys**

Paste the key contain in it save and exite.

Step6:- Same steps with Slave 2

Step7:- Master Node – **cd /etc/ansible/** path and do **ls** and check available file init.

There is **hosts** file , Copy Slave1 and Slave2 PrivateIPs and paste in host file.



Step8:-Now create a playbook for installing other things like java and Jenkins.

sudo nano play.yaml create and write a code inside the file.

- name: installing java and jenkins inside the machine where

ansible is installed

hosts: master

become: yes

tasks:

- name: install java and jenkins

script: master.sh

- name: installing java and docker inside the other 2 machines

hosts: all

become: yes

tasks:

- name: install java and docker

script: slave.sh

```
aws [Search] [Alt+S] Ask Amazon Q

GNU nano 6.2 play.yaml
--
- name: installing java and jenkins inside the machine where ansible is installed
  hosts: master
  become: yes
  tasks:
    - name: install java and jenkins
      script: master.sh
- name: installing java and docker inside the other 2 machines
  hosts: slave
  become: yes
  tasks:
    - name: install java and docker
      script: slave.sh

[ Read 14 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line M-

i-03a294a19d7dcf2e3 (Proj1-Master)
PublicIPs: 34.224.7.181 PrivateIPs: 172.31.23.181
```

Step9:-Create **master.sh** scrip file **sudo nano master.sh**

```
sudo apt update
```

```
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
```

```
sudo apt-get update
```

```
sudo apt-get install fontconfig openjdk-17-jre -y
```

```
sudo apt-get install jenkins -y
```

Step 10:-Create **slave.sh** file **sudo nano slave.sh**

```
Sudo apt update
```

```
Sudo apt install openjdk-17-jdk -y
```

```
Sudo apt install docker.io -y
```

```
GNU nano 6.2
sudo apt update
sudo apt install openjdk-17-jre -y
sudo apt install docker.io -y
```

^G Help ^O Write Out ^W Where Is
^X Exit ^R Read File ^\ Replace

i-0cc839da40c06a092 (Proj1-Master)
PublicIPs: 54.159.26.132 PrivateIPs: 172.31.26.199

Step 11:-Execute the playbook – **ansible-playbook play.yaml**

aws Search [Alt+S]

```
ok: [localhost]
TASK [install java and jenkins] *****
changed: [localhost]

PLAY [installing java and docker inside the other 2 machines] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host 172.31.17.97 is using the discovered Python interpreter at /usr/bin/python3.10, 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: [172.31.17.97]
[WARNING]: Platform linux on host 172.31.30.94 is using the discovered Python interpreter at /usr/bin/python3.10, 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: [172.31.30.94]

TASK [install java and docker] *****
changed: [172.31.30.94]
changed: [172.31.17.97]

PLAY RECAP *****
172.31.17.97      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.30.94     : 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-172-31-26-199:/etc/ansible$
```

i-0cc839da40c06a092 (Proj1-Master)
PublicIPs: 54.159.26.132 PrivateIPs: 172.31.26.199

Step 12:- Go to given git link

Create a new fork – **Dockerfile**

FROM ubuntu

RUN apt-get update

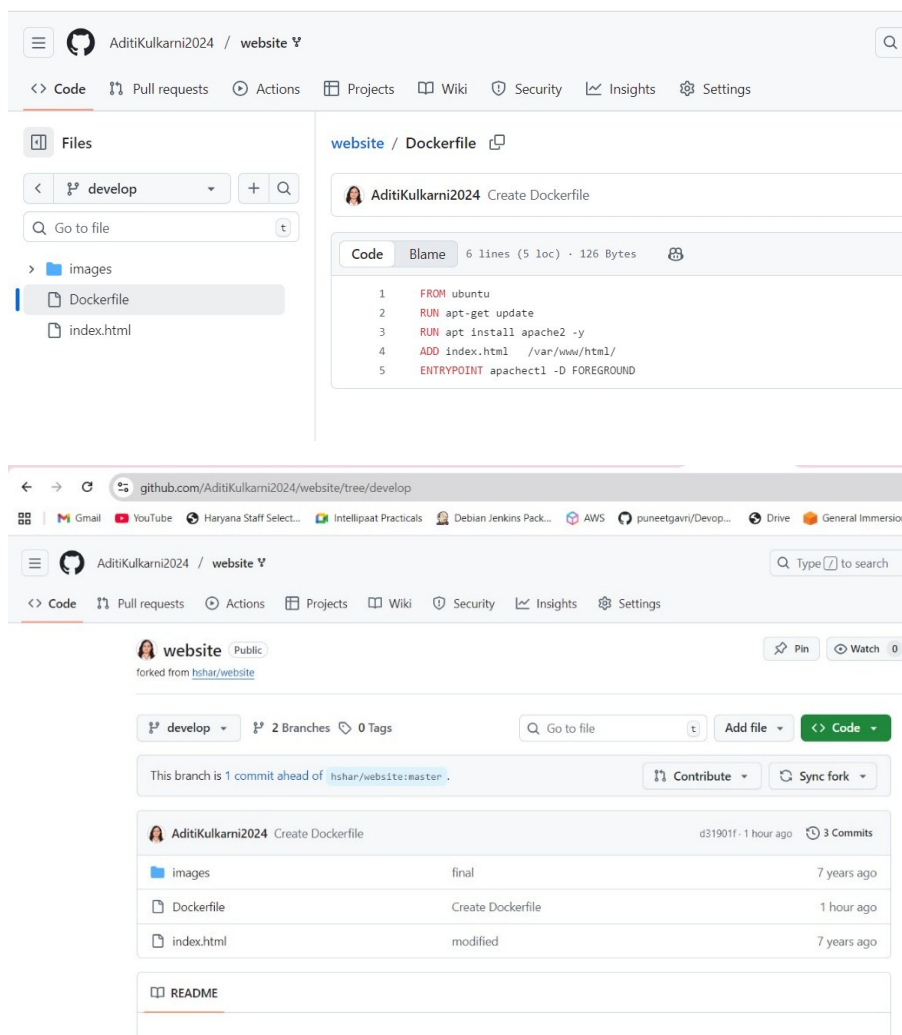
RUN apt install apache2 -y

ADD index.html /var/www/html/

ENTRYPOINT apachectl -D FOREGROUND

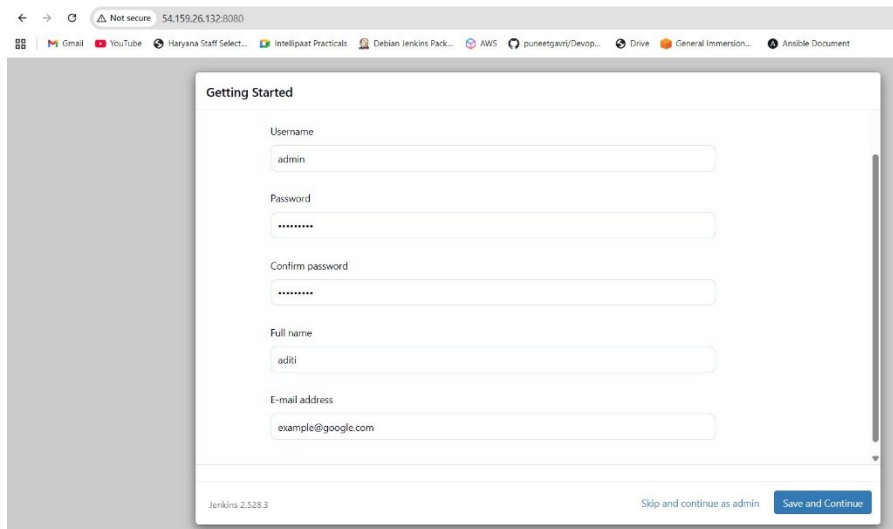
Commit the changes and

Create one branch **develop**



Step 13:- Go back to Master node and check Jenkins version.

Copy the PublicIP of master node and open Jenkins and setup the Jenkins.

A screenshot of the Jenkins 'Getting Started' configuration screen. The browser address bar shows '54.159.26.132:8080'. The form contains the following fields: Username (admin), Password (masked with dots), Confirm password (masked with dots), Full name (aditi), and E-mail address (example@google.com). At the bottom, there is a 'Skip and continue as admin' link and a 'Save and Continue' button. The Jenkins version 'Jenkins 2.528.3' is displayed in the bottom left corner.

Getting Started

Username
admin

Password

Confirm password

Full name
aditi

E-mail address
example@google.com

Jenkins 2.528.3

[Skip and continue as admin](#) [Save and Continue](#)

Step 14:-Manage Jenkins – Nodes – create new Node – **test-S1**

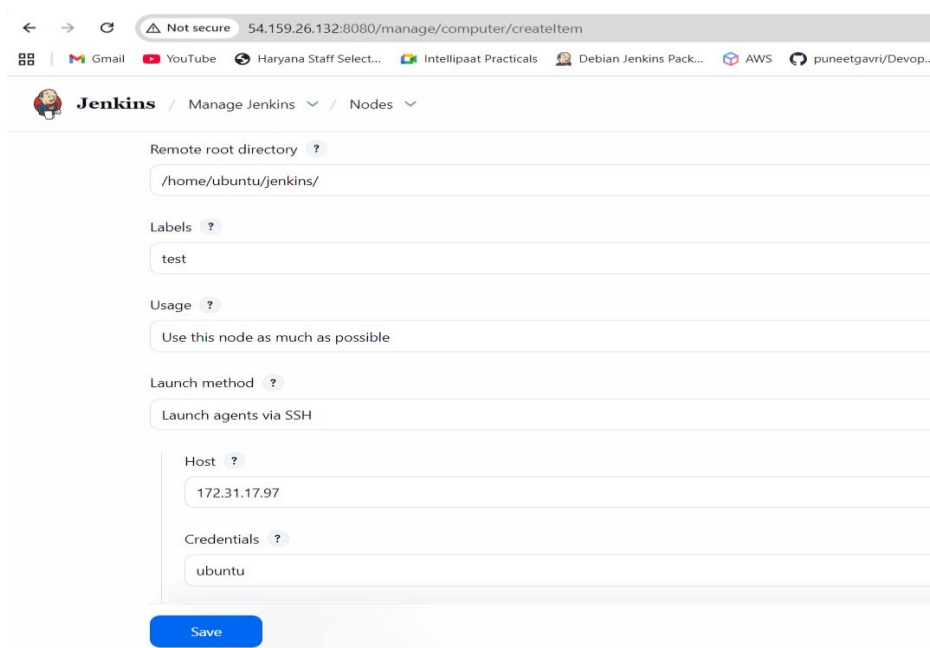
Remote root directory **-/home/ubuntu/Jenkins**

Labels – **test**

Copy Slave1 Private IP and paste in Host column

Credentials – Add **username - ubuntu** and **key-pem** file in secret key.

Save and create Node.

A screenshot of the Jenkins 'Manage Jenkins' - 'Nodes' - 'Create Item' configuration screen. The browser address bar shows '54.159.26.132:8080/manage/computer/createItem'. The form contains the following fields: Remote root directory (/home/ubuntu/jenkins/), Labels (test), Usage (Use this node as much as possible), Launch method (Launch agents via SSH), Host (172.31.17.97), and Credentials (ubuntu). A 'Save' button is at the bottom.

Jenkins / Manage Jenkins / Nodes

Remote root directory ?
/home/ubuntu/jenkins/

Labels ?
test

Usage ?
Use this node as much as possible

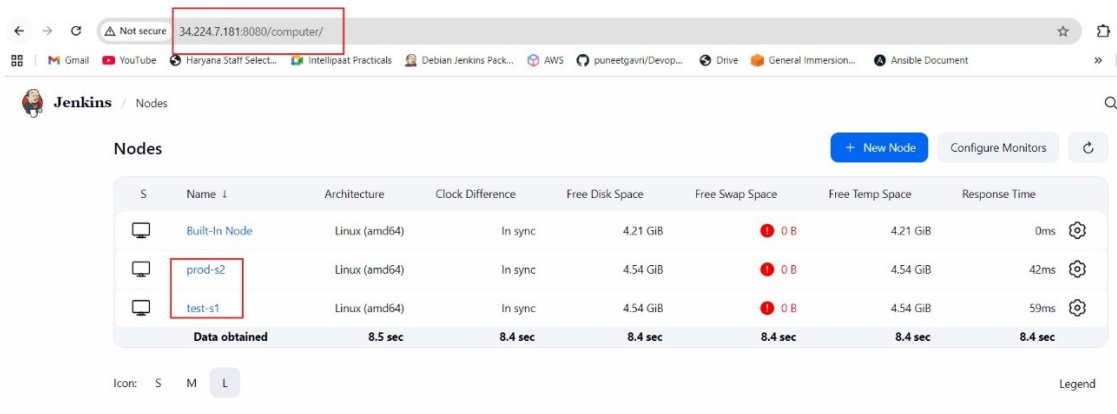
Launch method ?
Launch agents via SSH

Host ?
172.31.17.97

Credentials ?
ubuntu

[Save](#)

Step 14:- Create Node 2 name- **prod-S2** using same steps



The screenshot shows the Jenkins 'Nodes' page. The browser address bar displays '34.224.7.181:8080/computer/'. The page title is 'Jenkins / Nodes'. There are buttons for '+ New Node', 'Configure Monitors', and a refresh icon. A table lists the nodes:

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	4.21 GiB	0 B	4.21 GiB	0ms
	prod-s2	Linux (amd64)	In sync	4.54 GiB	0 B	4.54 GiB	42ms
	test-s1	Linux (amd64)	In sync	4.54 GiB	0 B	4.54 GiB	59ms
Data obtained		8.5 sec	8.4 sec	8.4 sec	8.4 sec	8.4 sec	8.4 sec

At the bottom, there are icons for 'Icon: S M L' and a 'Legend' link.

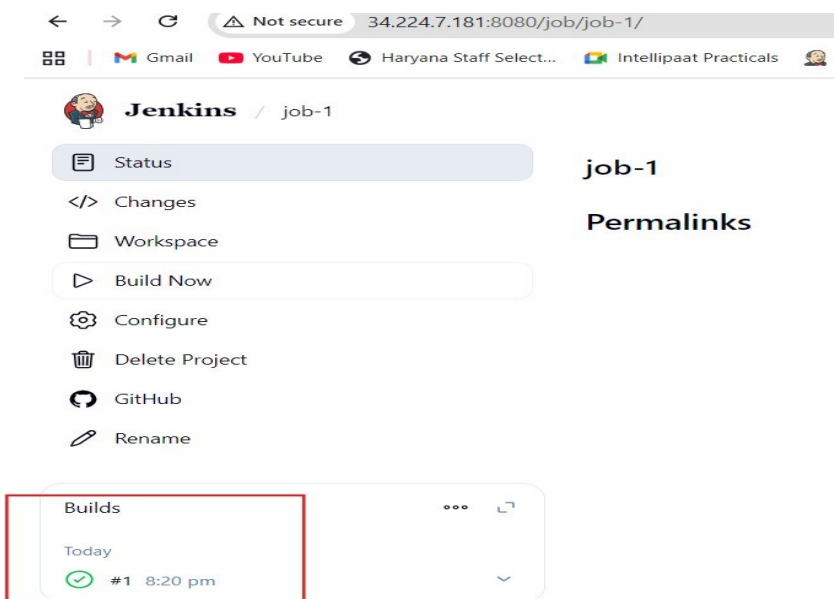
Step 15:-Create a Job – **Job-1** – freestyle project

GitHub project column – paste github repo url

Label expression – test

Source code Management – same url

Branch Specifire - */develop - save and build the job.



The screenshot shows the Jenkins 'job-1' page. The browser address bar displays '34.224.7.181:8080/job/job-1/'. The page title is 'Jenkins / job-1'. On the left, there is a sidebar with links: 'Status', 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Project', 'GitHub', and 'Rename'. On the right, there is a section titled 'job-1' with a sub-section 'Permalinks'. At the bottom, there is a 'Builds' section with a table showing the build history:

Builds
Today
#1 8:20 pm

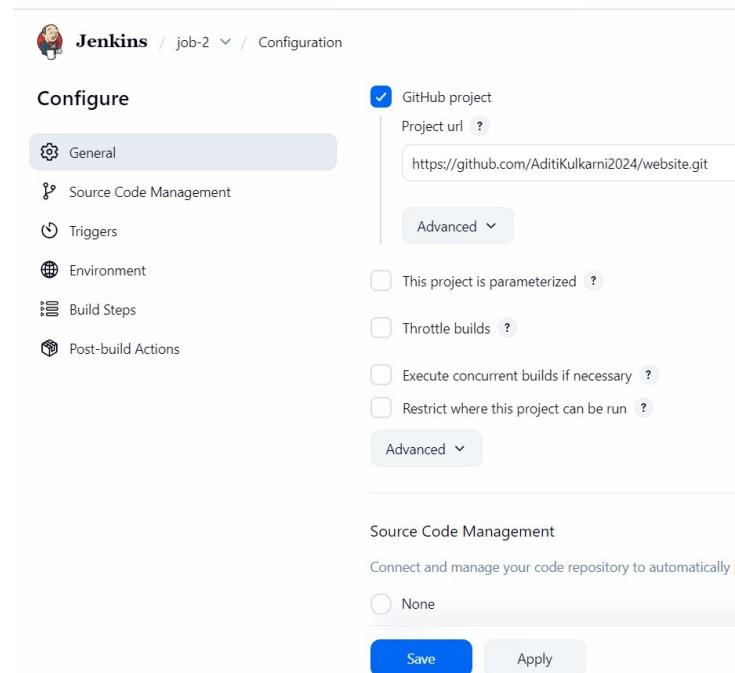
Step 17:-Create a **Job-2** in Jenkins – freestyle

GitHub – Git URL

Label – **prod**

Branch Specifire - */master

Save and create



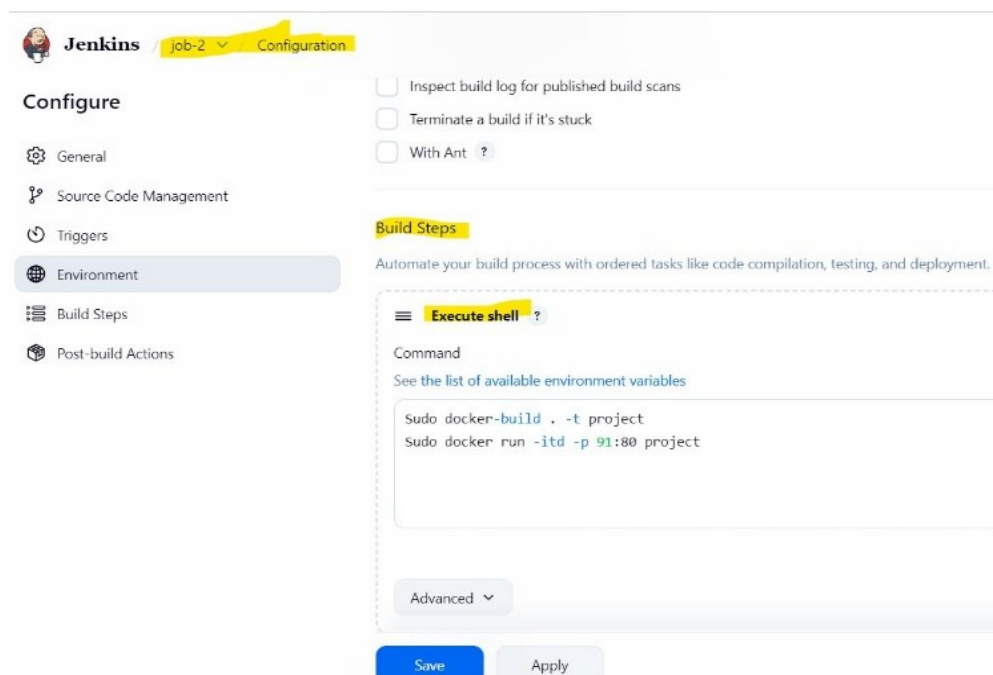
The screenshot shows the Jenkins configuration page for 'job-2'. The 'Configure' section is active, with 'General' selected in the left sidebar. The 'GitHub project' checkbox is checked, and the 'Project url' is set to 'https://github.com/AditiKulkarni2024/website.git'. The 'Advanced' dropdown is expanded, showing options like 'This project is parameterized', 'Throttle builds', 'Execute concurrent builds if necessary', and 'Restrict where this project can be run'. The 'Source Code Management' section is also visible, with 'None' selected for the repository type. The 'Save' and 'Apply' buttons are at the bottom.

Again go to Job-2 -and configure and go to **Build Steps**

In Build Steps – Execute shell

Sudo docker-build . -t project

Sudo docker run -itd -p 91:80 project



The screenshot shows the Jenkins configuration page for 'job-2', specifically the 'Build Steps' section. The 'Execute shell' checkbox is checked, and the 'Command' field contains the following text: 'Sudo docker-build . -t project' and 'Sudo docker run -itd -p 91:80 project'. The 'Advanced' dropdown is expanded, showing options like 'Inspect build log for published build scans', 'Terminate a build if it's stuck', and 'With Ant'. The 'Save' and 'Apply' buttons are at the bottom.

The screenshot shows the Jenkins web interface for a job named 'job-2'. The browser address bar indicates the URL is '34.224.7.181:8080/job/job-2/'. The Jenkins logo and 'job-2' are at the top. On the left, a sidebar contains links: Status, Changes, Workspace, Build Now, Configure, Delete Project, GitHub, and Rename. The main area shows a green checkmark icon next to 'job-2', which is highlighted by a red box. Below this is a 'Permalinks' section with a list of build links: Last build (#5), 50 sec ago; Last stable build (#5), 50 sec ago; Last successful build (#5), 50 sec ago; Last failed build (#4), 6 min 20 sec ago; Last unsuccessful build (#4), 6 min 20 sec ago; and Last completed build (#5), 50 sec ago. At the bottom, a 'Builds' panel is also highlighted by a red box, showing a search filter and a list of builds under the heading 'Today', with the first entry being a successful build #5 at 8:41 pm.

Save and Build – go to Slave2 copy PublicIp and check on browser.