

Capstone Project 1:-

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

Version – Ubuntu 22.0 LTS

t2.medium – default security group

Instances (3) Info		Last updated less than a minute ago	Connect
<input type="text"/> Find Instance by attribute or tag (case-sensitive)			
<input type="checkbox"/>	Name 🔗	Instance ID	Instance state 🕒
<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's fingerprint is:
SHA256:qM3520WFTYQNEQMIwuslhVW2hvREhcaF7ulg2N15U ubuntu@ip-172-31-26-199
The key's randomart image is:
+---[RSA 3072]---+
| ...%*o= . . |
| ..*o*X.. E. |
| ..o.o+oo. . |
| . =.o+.o . |
| + .S.= |
| o o + . |
| . . . |
| . . . |
+---[SHA256]---+
ubuntu@ip-172-31-26-199:~/.ssh$ sudo cat /home/ubuntu/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAQABAAQgQCQRu/CIEImrjHAtIFVSOAX+3wuSzyMTi5NHfttZP7CpbEnYEJVVGkobDmc98tZJyEk+WATg93MUksxInVfLVKmzkrAydHCZQg3z0c2bQcNh15hhTJF5sYMJ5scds93T9
lotra8vPwMmHQR2mfleDEL8hJFzukKBRaad38EuXX4WPu5JePuIJBWB9m1o+0csPwNYNKVZgU8yBmPlK15GBnSobf1oJ1UQEWcstAo7kvKm/aq/EVChrfJ8ye2Eu9CY1CaoU+eoXJB+WO02Em8zFexQ8yZuB5WNTLxd
bWfcXVKRaOlyMr1DrvxRo5G53lpTqlnaumiNrvzgmt.PDDQMZZJ1l867NcyH4xjqR41YEntwTZzJholUCUxdQHO7o4QZM50Nmy95fpR924U9XrMcepIrrjVqtesfgtIRGS8gVHg4eNlBcDNkbmebGmictkfWYE9JtZ6WJ
0htRIXIu1oulf6Hew8Mp+r4tqmlmopYa9o3UBogd4S9kRqdIB= ubuntu@ip-172-31-26-199
ubuntu@ip-172-31-26-199:~/.ssh$ ^[[200~cd /etc/ansible~

```

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

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 exit.

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.



```

GNU nano 6.2                                         hosts

[master]
localhost ansible_connection=local

[slave]
172.31.29.165
172.31.23.23

```

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

The screenshot shows a terminal window with the AWS logo at the top. The title bar says "play.yaml". The main area contains the Ansible playbook code. At the bottom, there's a status bar with various keyboard shortcuts and the command "i-03a294a19d7dcf2e3 (Proj1-Master)".

```
GNU nano 6.2
---
- 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)

Public IPs: 34.224.7.181 Private IPs: 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
```

```

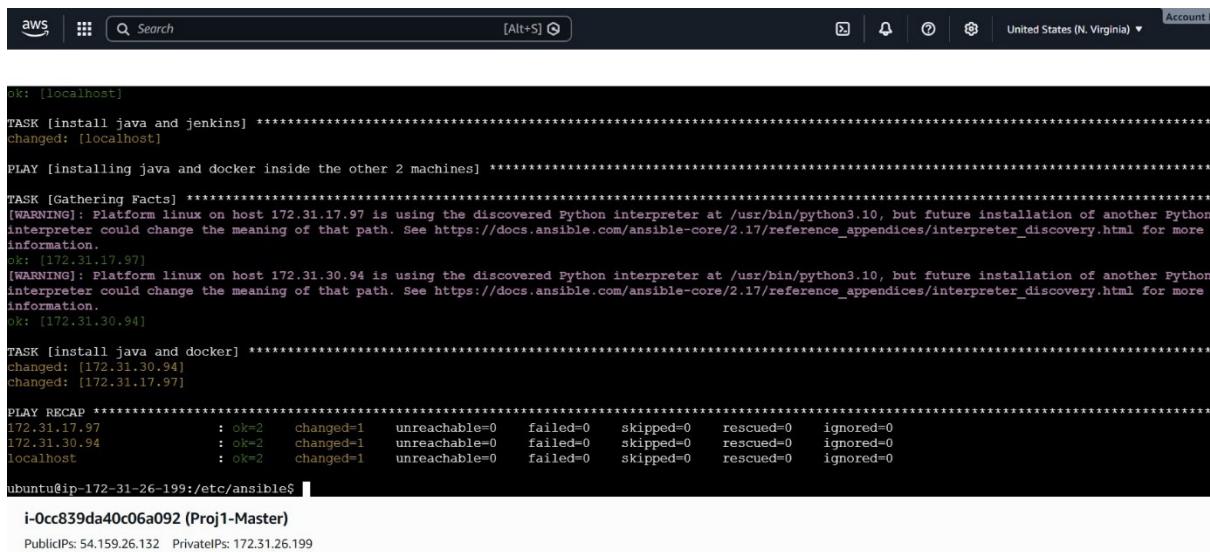
GNU nano 6.2
sudo apt update
sudo apt install openjdk-17-jre -y
sudo apt install docker.io -y

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

i-Occ839da40c06a092 (Proj1-Master)
Public IPs: 54.159.26.132  Private IPs: 172.31.26.199

```

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



```

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-Occ839da40c06a092 (Proj1-Master)
 Public IPs: 54.159.26.132 Private IPs: 172.31.26.199

Step 12:- Go to given git link

Create a new fork – **Dockerfile**

FROM ubuntu

RUN apt-get update

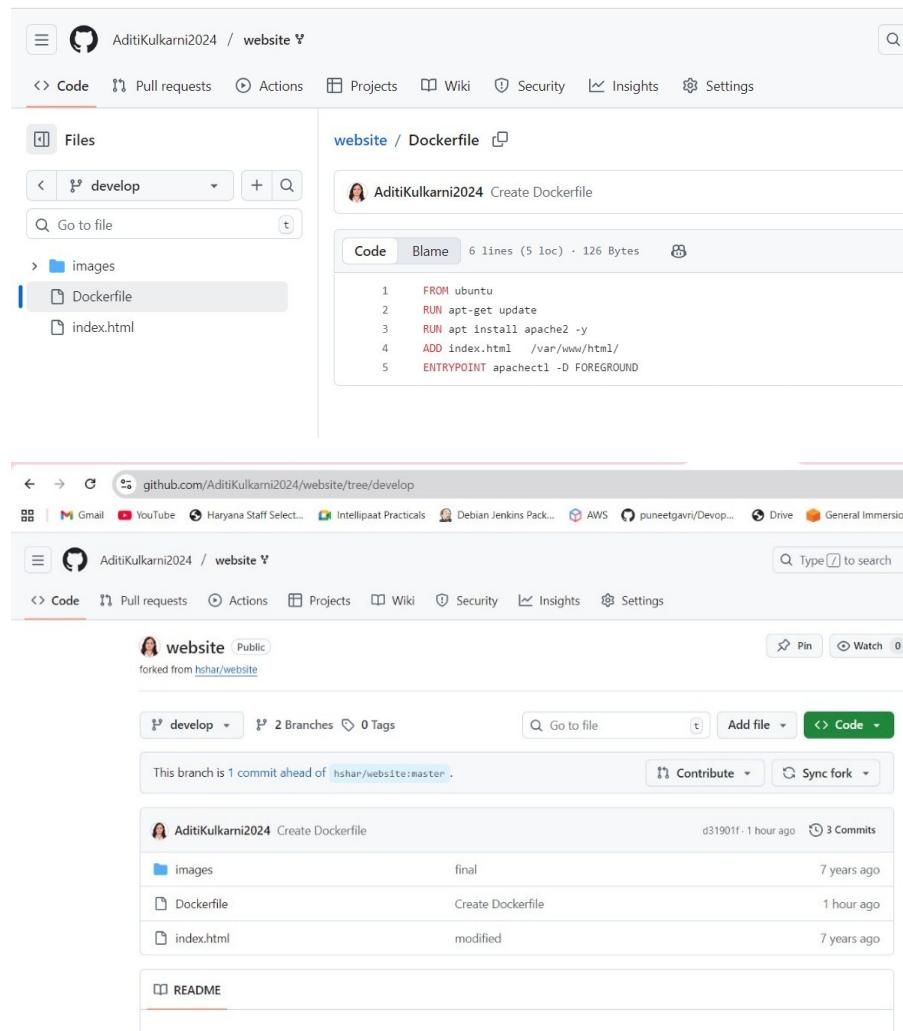
```

RUN apt install apache2 -y
ADD index.html /var/www/html/
ENTRYPOINT apachectl -D FOREGROUND

```

Commit the changes and

Create one branch **develop**



The screenshot shows two GitHub pages. The top page displays the Dockerfile content:

```

FROM ubuntu
RUN apt-get update
RUN apt install apache2 -y
ADD index.html /var/www/html/
ENTRYPOINT apachectl -D FOREGROUND

```

The bottom page shows the commit history for the 'develop' branch of the 'website' repository. It includes a summary table of changes:

File	Change	Time
index.html	modified	7 years ago
Dockerfile	Create Dockerfile	1 hour ago
images	final	7 years ago

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

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

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.

Manage Jenkins / Nodes

Remote root directory ? /home/ubuntu/Jenkins/

Labels ? test

Usage ? Use this node as much as possible

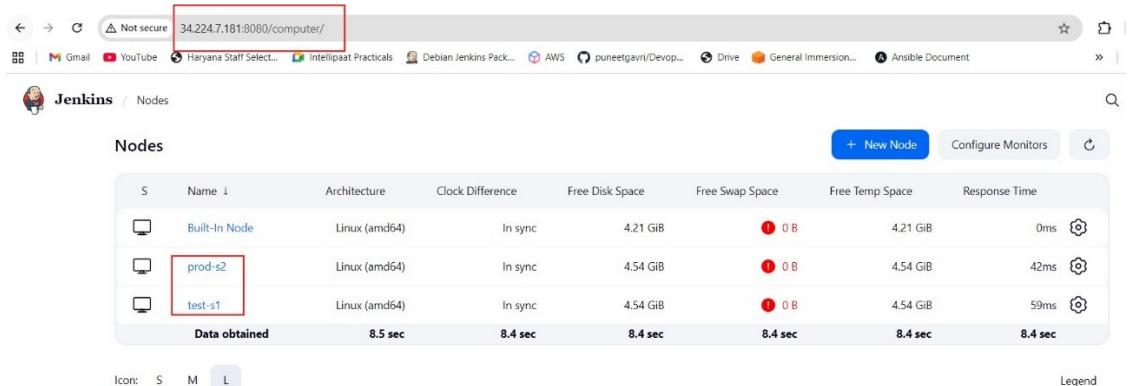
Launch method ? Launch agents via SSH

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. At the top, there's a navigation bar with icons for Gmail, YouTube, Haryana Staff Select..., Intellipaat Practicals, Debian Jenkins Pack..., AWS, puneetgavri/Devop..., Drive, General Immersion..., and Ansible Document. Below the navigation bar is the Jenkins logo and the word 'Nodes'. A search bar is on the right. The main area is titled 'Nodes' and contains a table with columns: S, Name, Architecture, Clock Difference, Free Disk Space, Free Swap Space, Free Temp Space, and Response Time. The table has three rows: 'Built-In Node' (Linux (amd64)), 'prod-s2' (Linux (amd64)), and 'test-s1' (Linux (amd64)). The 'prod-s2' row is highlighted with a red box. The table footer shows 'Data obtained' and times for each column. At the bottom left is a legend with icons for S, M, and L. At the bottom right is a 'Legend' link.

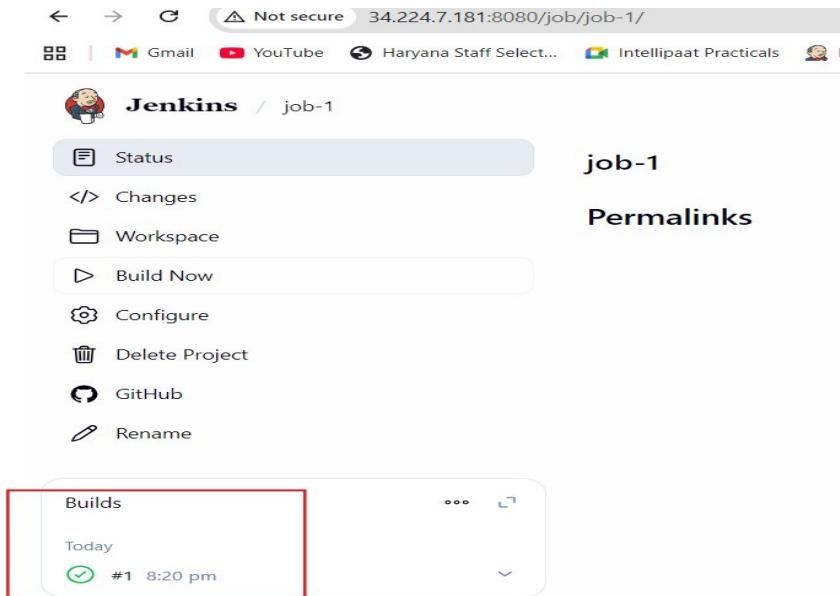
Step 15:-Create a Job – **Job-1** – freestyle project

GitHub project column – paste githup 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' status page. At the top, there's a navigation bar with icons for Gmail, YouTube, Haryana Staff Select..., Intellipaat Practicals, and a Jenkins icon. Below the navigation bar is the Jenkins logo and the word 'job-1'. On the left, there's a sidebar with options: Status (highlighted), Changes, Workspace, Build Now (button), Configure, Delete Project, GitHub, and Rename. The main area is titled 'job-1' and 'Permalinks'. It shows a 'Builds' section with a red box around it. Inside the 'Builds' section, it says 'Today' and shows a green checkmark next to '#1 8:20 pm'. There are also 'More' and 'Less' buttons.

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

GitHub – Git URL

Label – **prod**

Branch Specifire - */master

Save and create

Configure

GitHub project
Project url ?
https://github.com/AditiKulkarni2024/website.git

This project is parameterized ?

Throttle builds ?

Execute concurrent builds if necessary ?

Restrict where this project can be run ?

Advanced ▾

Source Code Management

Connect and manage your code repository to automatically |

None

Save Apply

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

Configure

General

Source Code Management

Triggers

Environment

Build Steps

Post-build Actions

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 project
Sudo docker run -itd -p 91:80 project

Advanced ▾

Save Apply

The screenshot shows a Jenkins job page for 'job-2'. The URL is 34.224.7.181:8080/job/job-2/. The page includes a sidebar with options like Status, Changes, Workspace, Build Now, Configure, Delete Project, GitHub, and Rename. A red box highlights the 'Status' button. Another red box highlights the 'job-2' link under the 'Permalinks' section. A third red box highlights the 'Builds' section, which shows a table with one row: 'Today #5 8:41 pm'. The table has columns for 'Builds', '...', and a search/filter icon.

Permalinks

- 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
- Last completed build (#5), 50 sec ago

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