

Application Deployment Using CI-CD

You have been hired as a 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 and 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. CodeBuild should automatically be triggered once a commit is made to master branch or develop branch.
 - a. If a commit is made to master branch, test and push to prod.
 - b. If a 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 GitHub. 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:

The screenshot shows the AWS EC2 Home page for the US East (N. Virginia) Region. The left sidebar includes links for EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances (selected), Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, and New. The main content area displays a summary of resources: Instances (running) 0, Auto Scaling Groups 0, Dedicated Hosts 0, Elastic IPs 0, Instances 0, Key pairs 1, Load balancers 0, Placement groups 0, Security groups 8, Snapshots 0, and Volumes 0. Below this is a 'Launch instance' button and a 'Service health' section. A sidebar on the right provides EC2 Free Tier Info, stating there are 3 EC2 free tier offers in use, and details about end-of-month forecasts and usage. The bottom navigation bar includes CloudShell, Feedback, a search bar, and system status indicators.

Go to AWS resources – Go to instances resources.

The screenshot shows the AWS EC2 Instances page. The left sidebar is identical to the previous screenshot. The main content area features an 'Instances Info' section with a search bar ('Find Instance by attribute or tag (case-sensitive)'), a 'Launch instances' button, and a table header for Name, Instance ID, Instance state, Instance type, Status check, and Alarm status. A message below the table states 'No matching instances found'. The bottom navigation bar is identical to the previous screenshot.

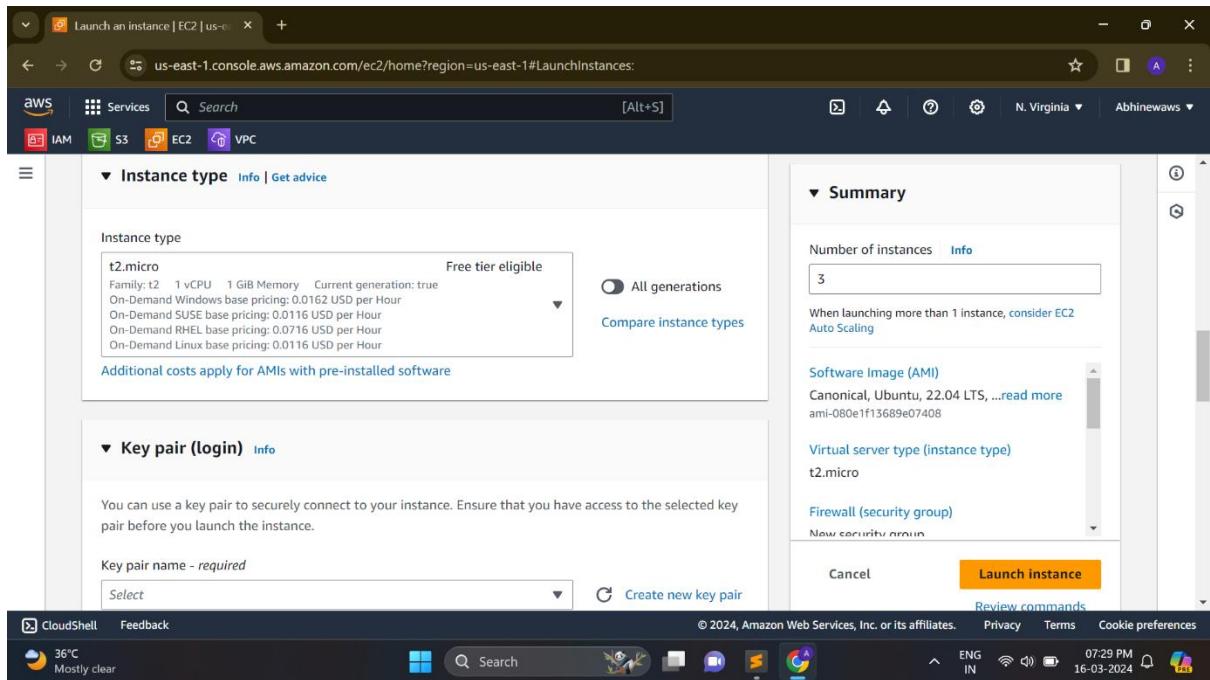
Click on lauch instances.

The screenshot shows the AWS EC2 'Launch an instance' page. In the 'Name and tags' section, the name 'Proj-' is entered. Under 'Software Image (AMI)', 'Amazon Linux 2023 AMI 2023.3.2...' is selected. The 'Virtual server type (instance type)' is set to 't2.micro'. The 'Launch instance' button is highlighted in orange at the bottom right.

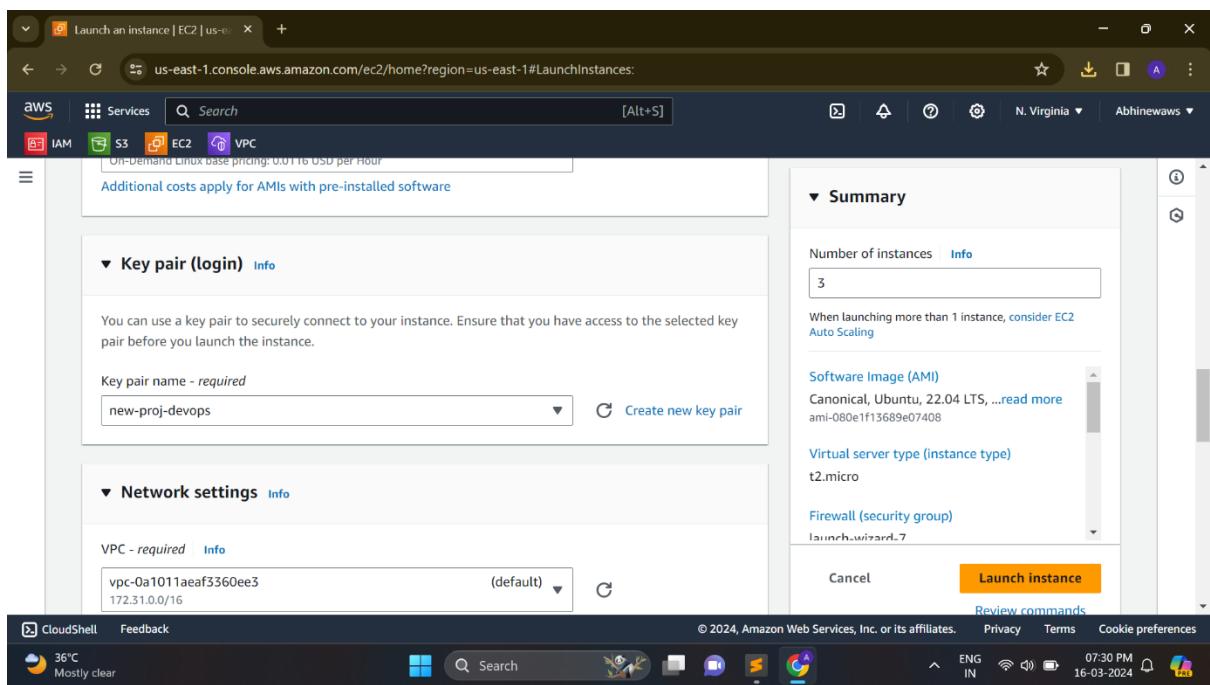
Name – project – no of instances – 03.

The screenshot shows the AWS EC2 'Launch an instance' page with a search bar containing 'Search our full catalog including 1000s of application and OS images'. In the 'Quick Start' section, 'Ubuntu' is selected. The 'Ubuntu Server 22.04 LTS (HVM), SSD Volume Type' AMI is chosen. The 'Launch instance' button is highlighted in orange at the bottom right.

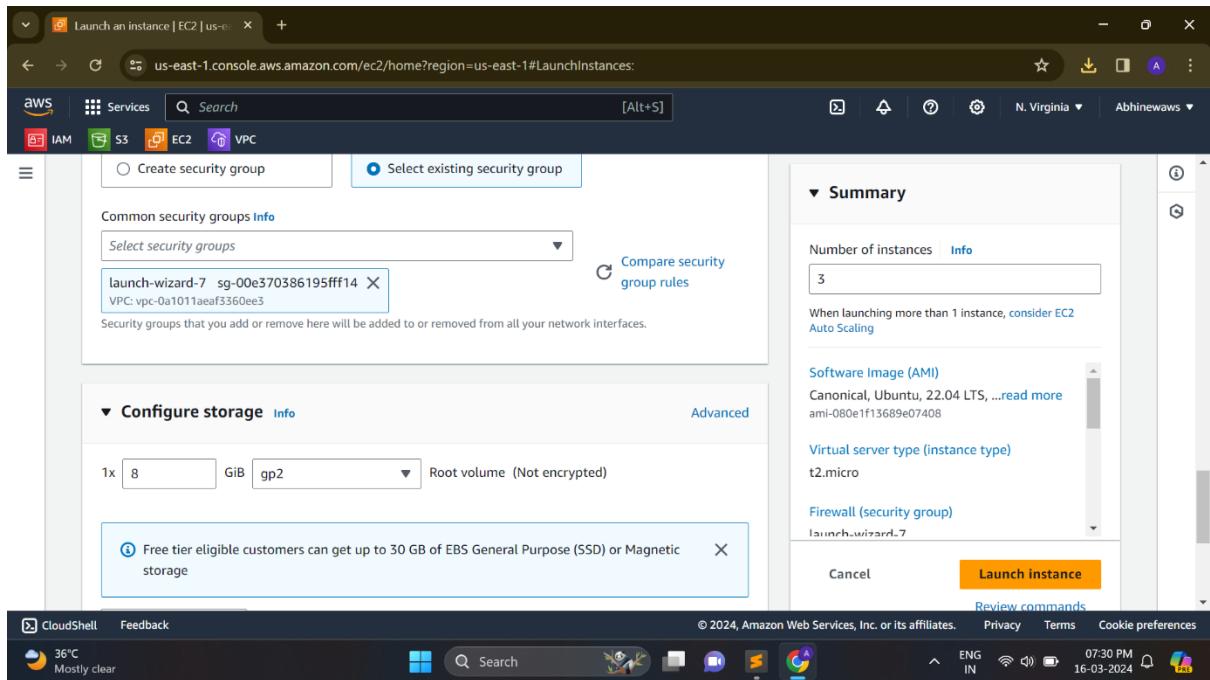
AMI – ubuntu – ubuntu server 22.04 LTS



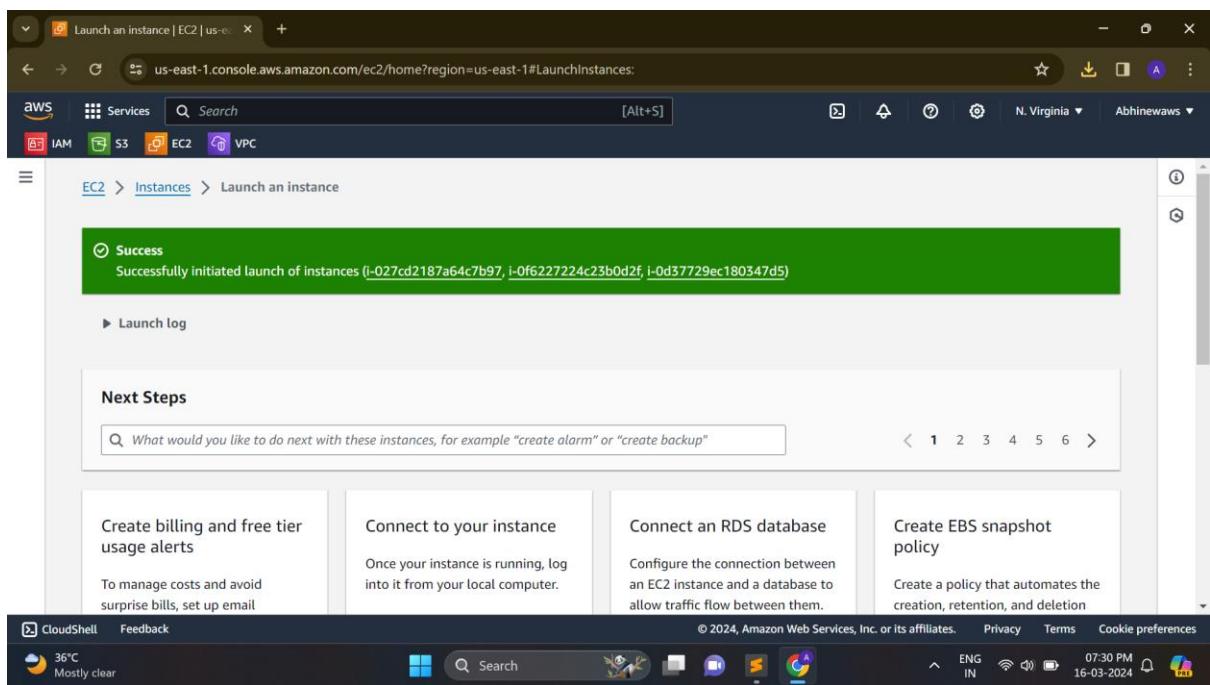
Instance type – t2.micro



Keypair.



Click on lauch instance.



Instance lauched successfully.

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main area displays a table of instances with the following data:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
Proj-M	i-0d37729ec180347d5	Running	t2.micro	Initializing	...	us-e
Proj-S1	i-0f6227224c23b0d2f	Running	t2.micro	Initializing	...	us-e
Proj-S2	i-027cd2187a64c7b97	Running	t2.micro	Initializing	...	us-e

A modal window titled "Select an instance" is open at the bottom, listing the three instances.

3 instances lauched successfully.

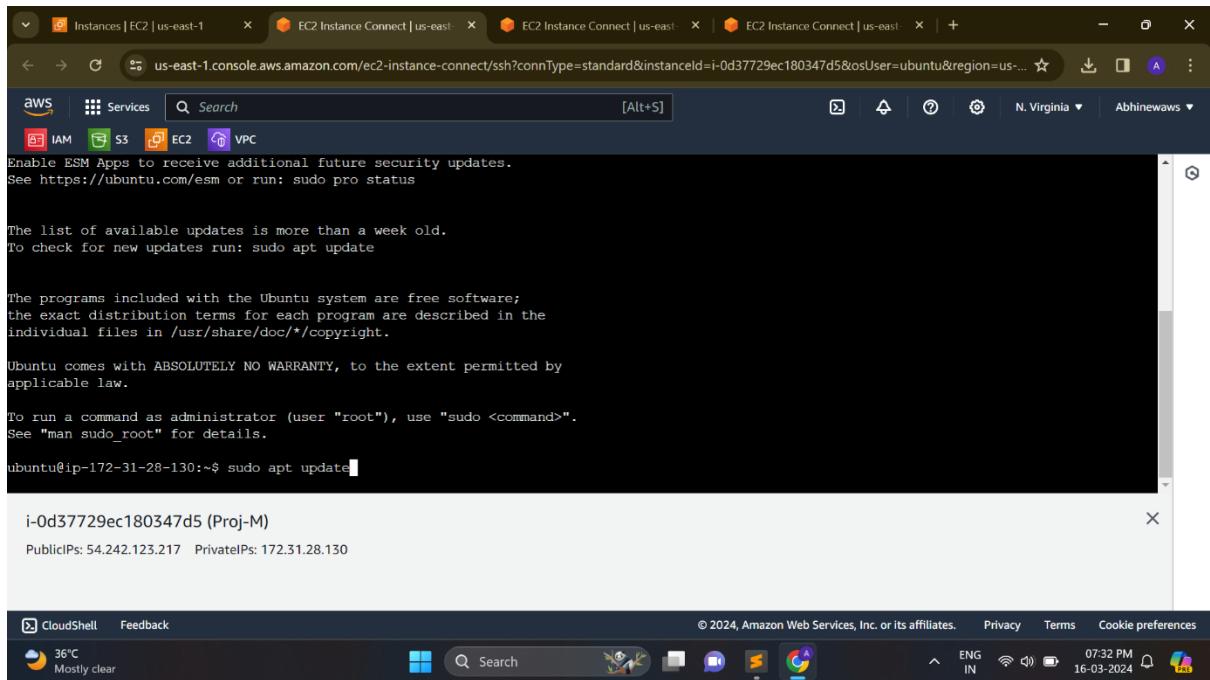
The screenshot shows the "Connect to instance" page for instance `i-0d37729ec180347d5 (Proj-M)`. The "Connection Type" section is expanded, showing two options:

- Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.
- Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

The "Public IP address" field contains `54.242.123.217`. The "Username" field contains `ubuntu`. A note at the bottom states: "Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username."

At the bottom right are "Cancel" and "Connect" buttons.

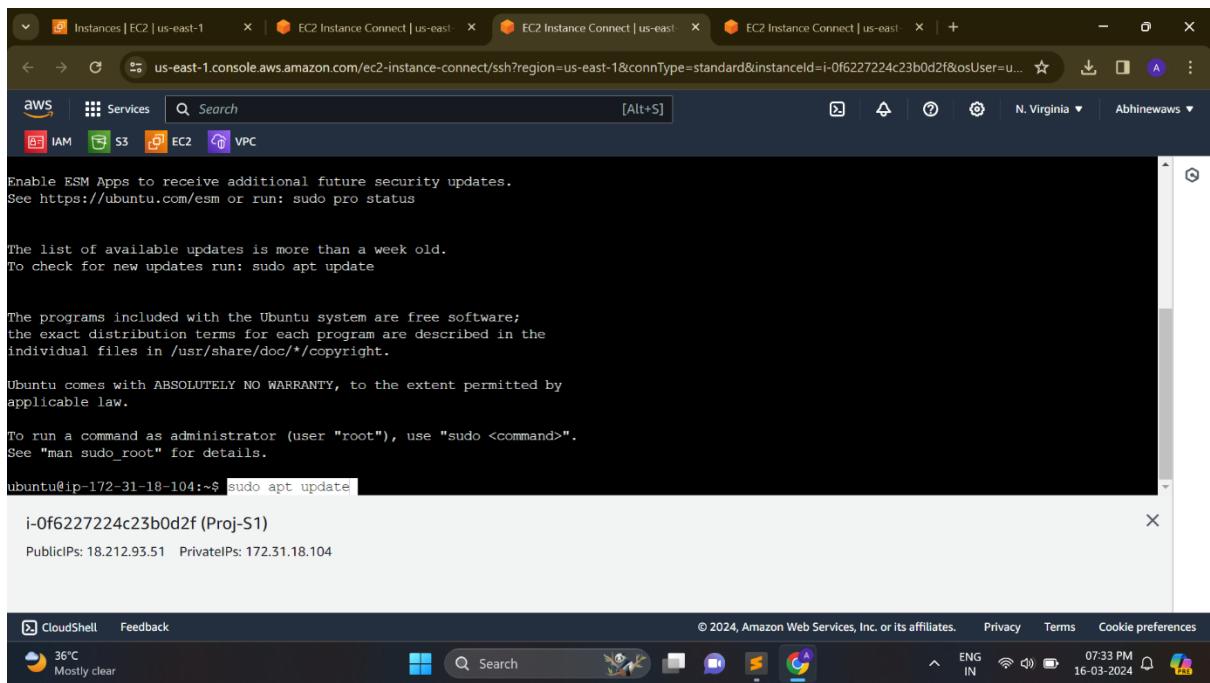
Click on connect.



```
Instances | EC2 | us-east-1 | EC2 Instance Connect | us-east-1 | EC2 Instance Connect | us-east-1 | EC2 Instance Connect | us-east-1 | +  
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0d37729ec180347d5&osUser=ubuntu&region=us-...  
aws Services Search [Alt+S]  
IAM S3 EC2 VPC  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
ubuntu@ip-172-31-28-130:~$ sudo apt update  
  
i-0d37729ec180347d5 (Proj-M)  
PublicIPs: 54.242.123.217 PrivateIPs: 172.31.28.130
```

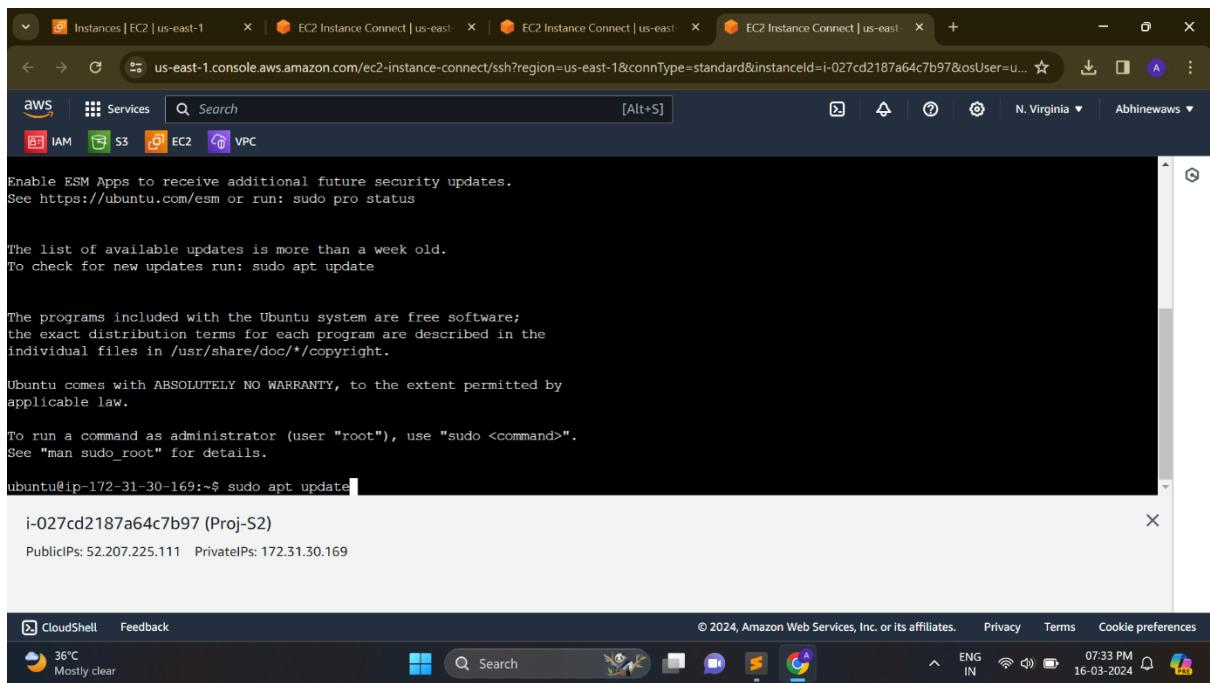
Update the instance.

\$ sudo apt update



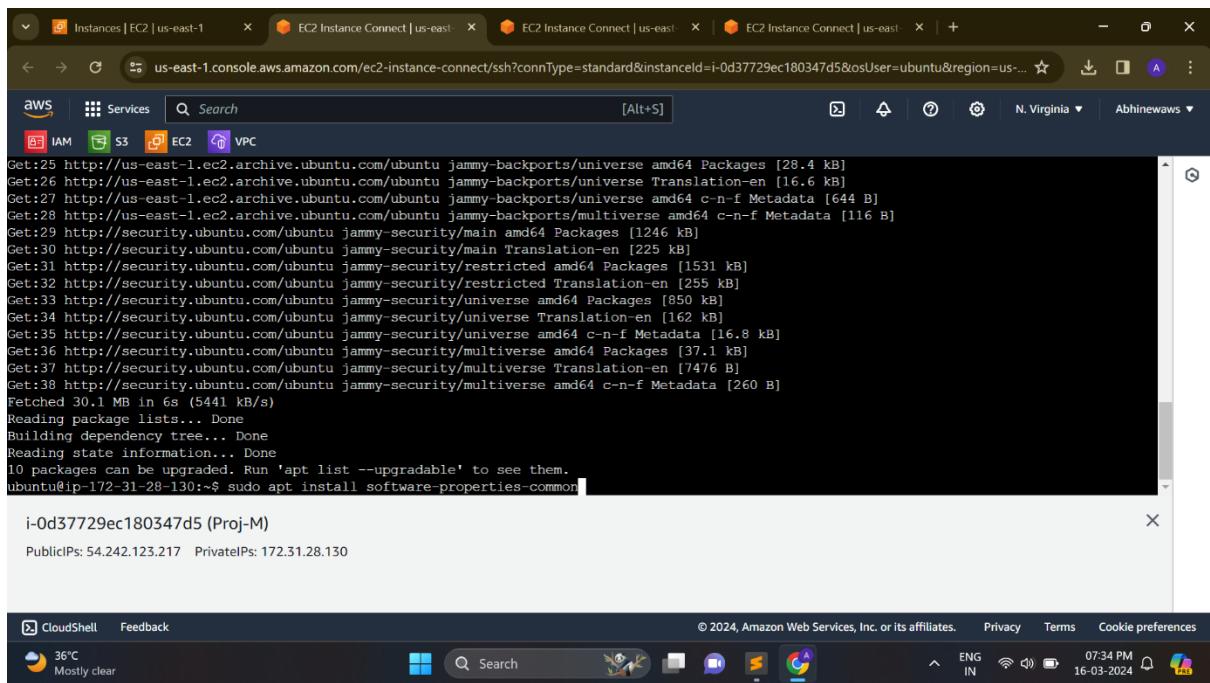
```
Instances | EC2 | us-east-1 | EC2 Instance Connect | us-east-1 | EC2 Instance Connect | us-east-1 | EC2 Instance Connect | us-east-1 | +  
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-0f6227224c23b0d2f&osUser=u...  
aws Services Search [Alt+S]  
IAM S3 EC2 VPC  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
ubuntu@ip-172-31-18-104:~$ sudo apt update  
  
i-0f6227224c23b0d2f (Proj-S1)  
PublicIPs: 18.212.93.51 PrivateIPs: 172.31.18.104
```

Update the proj -s1 instance.



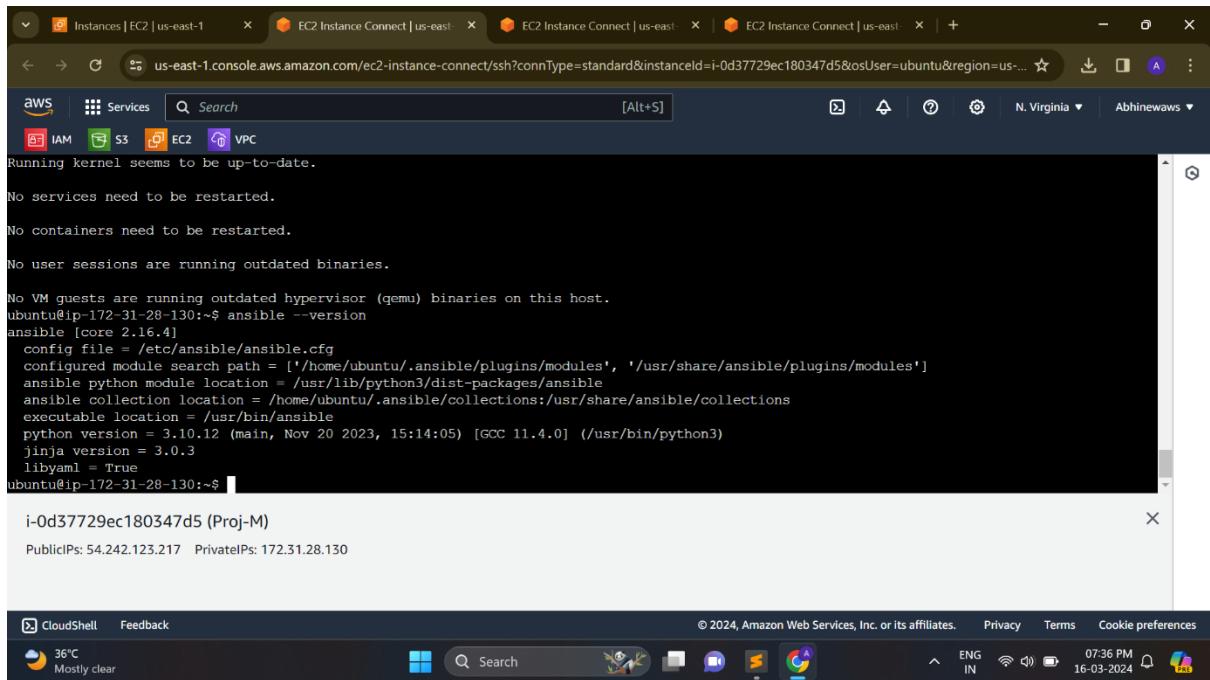
```
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-30-169:~$ sudo apt update  
i-027cd2187a64c7b97 (Proj-S2)  
Public IPs: 52.207.225.111 Private IPs: 172.31.30.169  
  
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences  
36°C Mostly clear
```

Update proj-s2 instance.



```
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [28.4 kB]  
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.6 kB]  
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [644 B]  
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]  
Get:29 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1246 kB]  
Get:30 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [225 kB]  
Get:31 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1531 kB]  
Get:32 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [255 kB]  
Get:33 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [850 kB]  
Get:34 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [162 kB]  
Get:35 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.8 kB]  
Get:36 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37.1 kB]  
Get:37 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7476 B]  
Get:38 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]  
Fetched 30.1 MB in 6s (5441 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
10 packages can be upgraded. Run 'apt list --upgradable' to see them.  
ubuntu@ip-172-31-28-130:~$ sudo apt install software-properties-common  
  
i-0d37729ec180347d5 (Proj-M)  
Public IPs: 54.242.123.217 Private IPs: 172.31.28.130  
  
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences  
36°C Mostly clear
```

Install ansible.



Instances | EC2 | us-east-1 | EC2 Instance Connect | us-east-1 | EC2 Instance Connect | us-east-1 | EC2 Instance Connect | us-east-1 | +

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0d37729ec180347d5&osUser=ubuntu®ion=us-east-1

aws Services Search [Alt+S]

N. Virginia Abhinewaws

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.

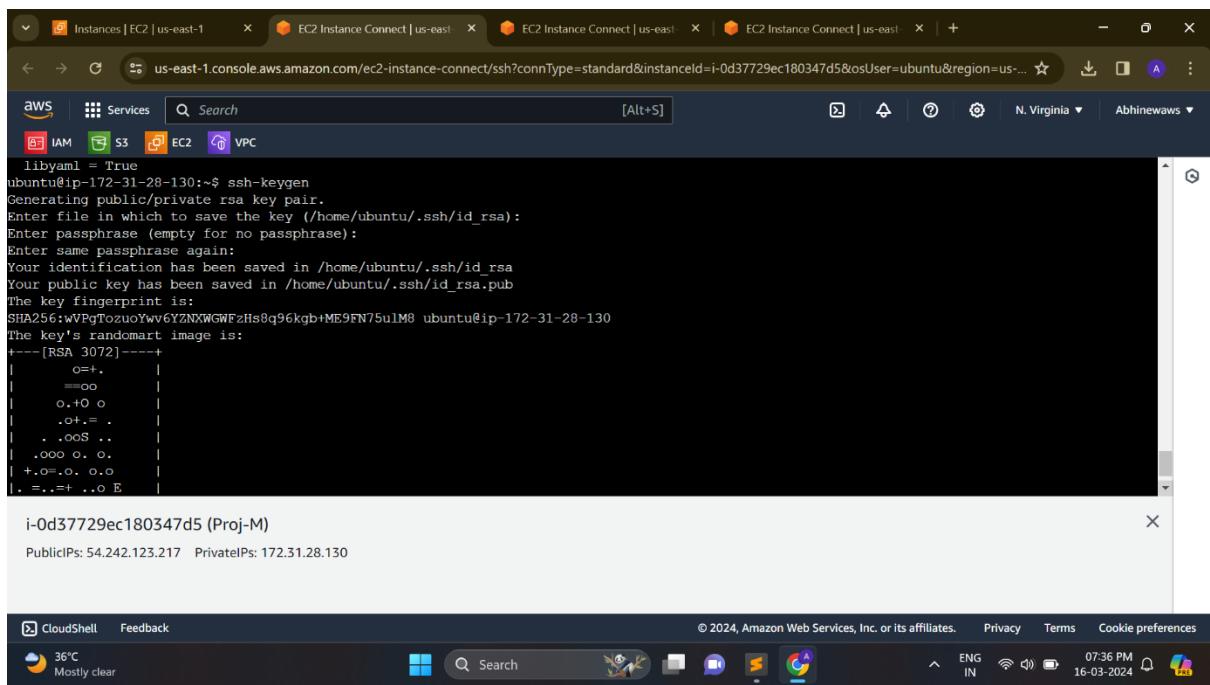
No VM guests are running outdated hypervisor (qemu) binaries on this host.

```
ubuntu@ip-172-31-28-130:~$ ansible --version
ansible [core 2.16.4]
  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 = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] (/usr/bin/python3)
  jinja version = 3.0.3
  libyaml = True
ubuntu@ip-172-31-28-130:~$
```

i-0d37729ec180347d5 (Proj-M)

Public IPs: 54.242.123.217 Private IPs: 172.31.28.130

Ansible installed successfully on master.



Instances | EC2 | us-east-1 | EC2 Instance Connect | us-east-1 | EC2 Instance Connect | us-east-1 | EC2 Instance Connect | us-east-1 | +

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0d37729ec180347d5&osUser=ubuntu®ion=us-east-1

aws Services Search [Alt+S]

N. Virginia Abhinewaws

libyaml = True

```
ubuntu@ip-172-31-28-130:~$ 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:wPgT0zuoYvv6YZNWXGFzHs8q96kgb+ME9FN75uLM0 ubuntu@ip-172-31-28-130
The key's randomart image is:
+---[RSA 3072]----+
|      o=+.      |
|      ==o       |
|      o.+O o     |
|      .o+=.      |
|      .ooS ..    |
|      .ooo o. o.  |
|      +.o=.o. o.o |
|      .=.+=..o E  |
```

i-0d37729ec180347d5 (Proj-M)

Public IPs: 54.242.123.217 Private IPs: 172.31.28.130

```
ubuntu@ip-172-31-28-130:~$ cd .ssh/
ubuntu@ip-172-31-28-130:~/ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
ubuntu@ip-172-31-28-130:~/ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAQABgQCrl+StQYsxStuDaDrYXGbZebB9UnBjJ+AtAJwIKFSevT/2LYY2V4ZHNaqEHTHlTDzbia6WlaRrw94TKRnEHg4iIEeoSQ7K7Ky]2Up9N1ZxAtQHRFkgf1CJ/hLcnzakhHP/x1hD0eUWeMvxEeqtTaSKaGn14/8TExVldtAF/tpvMv3hO0M2Yx20Ap6NQHsa7C+gw/th4EbJMF+SBrq1c1QU8rLSaUkWI2t+uuztgkTFQvgen/mGn25258woFnfb0aoYC/Youx22myC6H+mdnKBvEMeEnlyYi5DWhOem1XNUSgPphfsF4wGh1qyqglX1WrxXoh35aDeTk/3Efq5d/o2Us1lms1ZJWsJDUTXN5APC61qdTh1MDexwR0xa/0/Rb910n]/YWeZ?AlloxnP+pg5KiU+6XRosQ1Y60PULkJks6WE8KS7YgTPGLPSQTt9fLmkT1gdyGIRoW/mS3/fUq0sEwC4C2CFN]M7wzGje3qyPdj1Gq9X8= ubu
ntu@ip-172-31-28-130
ubuntu@ip-172-31-28-130:~/ssh$
```

i-0d37729ec180347d5 (Proj-M)
PublicIPs: 54.242.123.217 PrivateIPs: 172.31.28.130

Create a ssh keygen. Copy key.

```
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:29 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1246 kB]
Get:30 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [225 kB]
Get:31 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1531 kB]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [255 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [850 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [162 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.8 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37.1 kB]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7476 B]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
Fetched 30.1 MB in 6s (5426 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
10 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-18-104:~$ cd .ssh/
ubuntu@ip-172-31-18-104:~/ssh$ ls
authorized_keys
ubuntu@ip-172-31-18-104:~/ssh$ sudo nano authorized_keys
```

i-0f6227224c23b0d2f (Proj-S1)
PublicIPs: 18.212.93.51 PrivateIPs: 172.31.18.104

Paste the key in proj-s1 instance.

A screenshot of an AWS CloudShell terminal window. The title bar shows four tabs: 'Instances | EC2 | us-east-1', 'EC2 Instance Connect | us-east-1', 'EC2 Instance Connect | us-east-1', and 'EC2 Instance Connect | us-east-1'. The main content area is a terminal window with the following text:

```
authorized keys *
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQD5Hd8V7vB16Onun10N1deIP/t7NoimLgafASS/JrKiRbJ9WR6UhEKpEqobOV40tCERBFatZadaksV5ct6wAfaZTcXbKFse/T9x5>
<C2CFNjM7wzgQje3qyPdj1GqX8= ubuntu@ip-172-31-28-130
```

The terminal has a dark background and uses the nano text editor. The bottom status bar shows the instance ID 'i-Of6227224c23b0d2f (Proj-S1)', public and private IP addresses, and a help menu with keyboard shortcuts for various functions like Help, Write Out, Where Is, Cut, Execute, Location, Undo, Set Mark, Read File, Replace, Paste, Justify, Go To Line, Redo, and Copy.

Paste the key in proj-s2 instance.

```
Instances | EC2 | us-east-1      EC2 Instance Connect | us-east-1      EC2 Instance Connect | us-east-1      EC2 Instance Connect | us-east-1      +  
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-027cd2187a64c7b97&osUser=u...  
  
aws Services Search [Alt+S] N. Virginia ▾ Abhinewaws ▾  
AWS IAM S3 EC2 VPC  
GNU nano 6.2 authorized_keys *  
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQD5hd8VvB16Onunl0N1deLP/t7NoimLgafASS/JrKiRbJ9WR6UhEKpEqobOV40tCERBfAtZadaksV5ct6WafaZTcXbKFse/T9x5  
<C2CFNJM7wzgOje3qyPdj1Gq9X8= ubuntu@ip-172-31-28-130  
  
^G Help ^C Write Out ^W Where Is ^K Cut ^I Execute ^O Location M-U Undo M-A Set Mark  
^X Exit ^R Read File ^Y Replace ^U Paste ^J Justify ^G Go To Line M-E Redo M-C Copy  
  
i-027cd2187a64c7b97 (Proj-S2)  
PublicIPs: 52.207.225.111 PrivateIPs: 172.31.30.169
```

```
| . .ooS .. |
| .ooo o. o. |
| +o=o.o. o.o |
| . =+ ..o E |
| o...o++o. |
+---[SHA256]-----+
ubuntu@ip-172-31-28-130:~$ cd .ssh/
ubuntu@ip-172-31-28-130:~/ssh$ ls
authorized_keys id_rsa id_rsa.pub
ubuntu@ip-172-31-28-130:~/ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQGCrIu+SIQYsxStuDaDrYXGbZebB9UunBjJ+AtAJw1KFSevT72LYY2V4ZHNaqEHTHltDzbia6WlaRrw94TKRnEHg4iIEeoSQ7K7Kyj
2UP9N1ZXatQHRFkgf1CJ/hLcnzakhHP/xLhd0eUWeMxvEeqTaSKaGn14/8TeXVldtAF/tpvMy3hO0M2Yx2OAp6NQHsa7C+gw/th4EbJMf+SBrqIcIQU0rLSaUkWI2t+uuztgkTfQ
vgen/mGn25258woFNbdoaoYC/Youx22myC6H+mdhKBvBMeEnly1q5DWhoem1XNUsGPphfsF4wGhIqygkLX1W/rX0h35aDeTk/3Ef5d/o2Uss1mS1ZJPwSJDUtJXN5APC61gdTh
IMDexwRUx0a/7Rb910nj/YWeZ7ailoxnP+pqP5KiU+6XR0sQiYB6OPULkJks6WE8KS7YgTPGip5QTT9fLmkT1gDyGTRoW/mS3/fUq0sEwC4C2CFNjM7wzgGje3qyPdj1Gq9X8= ubu
ntu@ip-172-31-28-130
ubuntu@ip-172-31-28-130:~/ssh$ cd
ubuntu@ip-172-31-28-130:~$ cd /etc/ansible/
ubuntu@ip-172-31-28-130:~/etc/ansible$ ls
ansible.cfg hosts roles
ubuntu@ip-172-31-28-130:~/etc/ansible$ sudo nano hosts
```

i-0d37729ec180347d5 (Proj-M)

Public IPs: 54.242.123.217 Private IPs: 172.31.28.130

Edit the hosts file.

```
GNU nano 6.2
hosts
172.31.18.104
172.31.30.169
```

i-0d37729ec180347d5 (Proj-M)

Public IPs: 54.242.123.217 Private IPs: 172.31.28.130

Add your private ip addresses of your slave instances.

```
ubuntu@ip-172-31-28-130:~$ ansible -m ping all
172.31.30.169 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
172.31.18.104 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-28-130:~$ █
```

i-0d37729ec180347d5 (Proj-M)
PublicIPs: 54.242.123.217 PrivateIPs: 172.31.28.130

Ping all slave intances.

```
ubuntu@ip-172-31-28-130:~$ ansible -m ping all
172.31.30.169 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
172.31.18.104 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-28-130:~$ ls
ubuntu@ip-172-31-28-130:~$ sudo nano master.sh █
```

i-0d37729ec180347d5 (Proj-M)
PublicIPs: 54.242.123.217 PrivateIPs: 172.31.28.130

Create a script file for master instance.

A screenshot of a browser window titled "us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0d37729ec180347d5&osUser=ubuntu®ion=us-...". The address bar shows the URL. The page displays session information: "i-0d37729ec180347d5 (Proj-M)", "Public IPs: 54.242.123.217", and "Private IPs: 172.31.28.130". Below this, there's a terminal window showing command-line interactions. The terminal output includes:

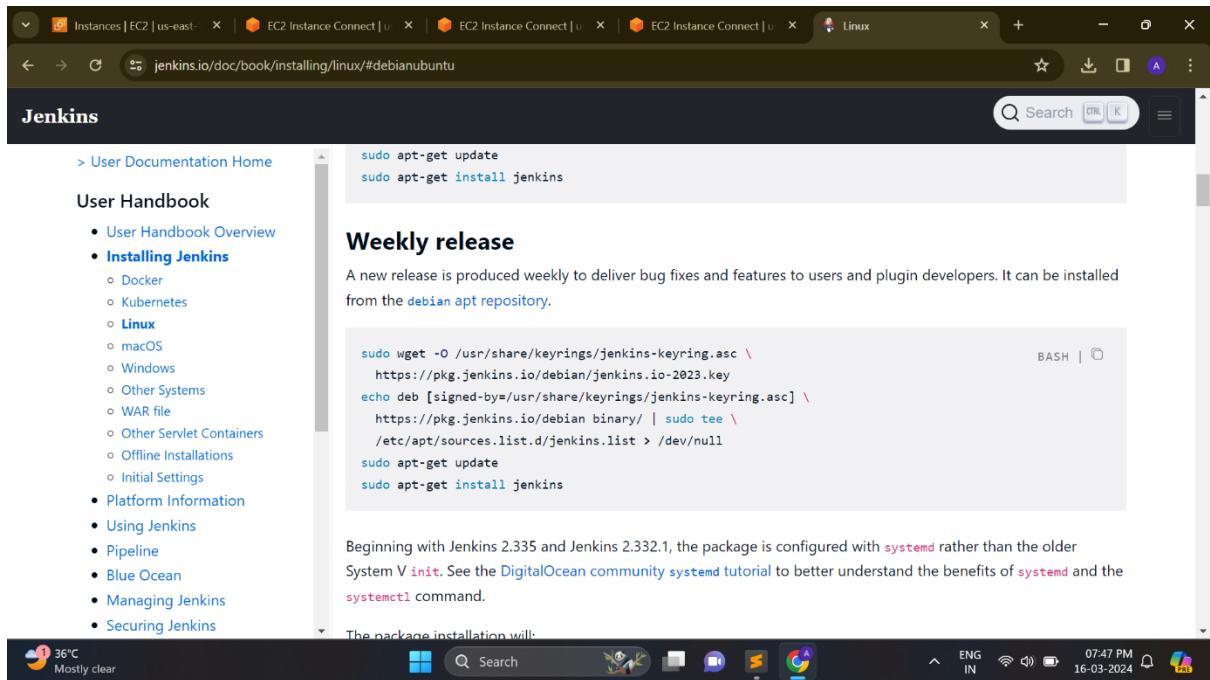
```
aws
Instances | EC2 | us-east-1
EC2 Instance Connect | us-east
N. Virginia
Abhinewaws

ping": "pong"
)
ubuntu@ip-172-31-28-130:~$ ansible -m ping all
172.31.30.169 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
172.31.18.104 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu@ip-172-31-28-130:~$ ls
ubuntu@ip-172-31-28-130:~$ sudo nano master.sh
ubuntu@ip-172-31-28-130:~$ sudo nano slave.sh
```

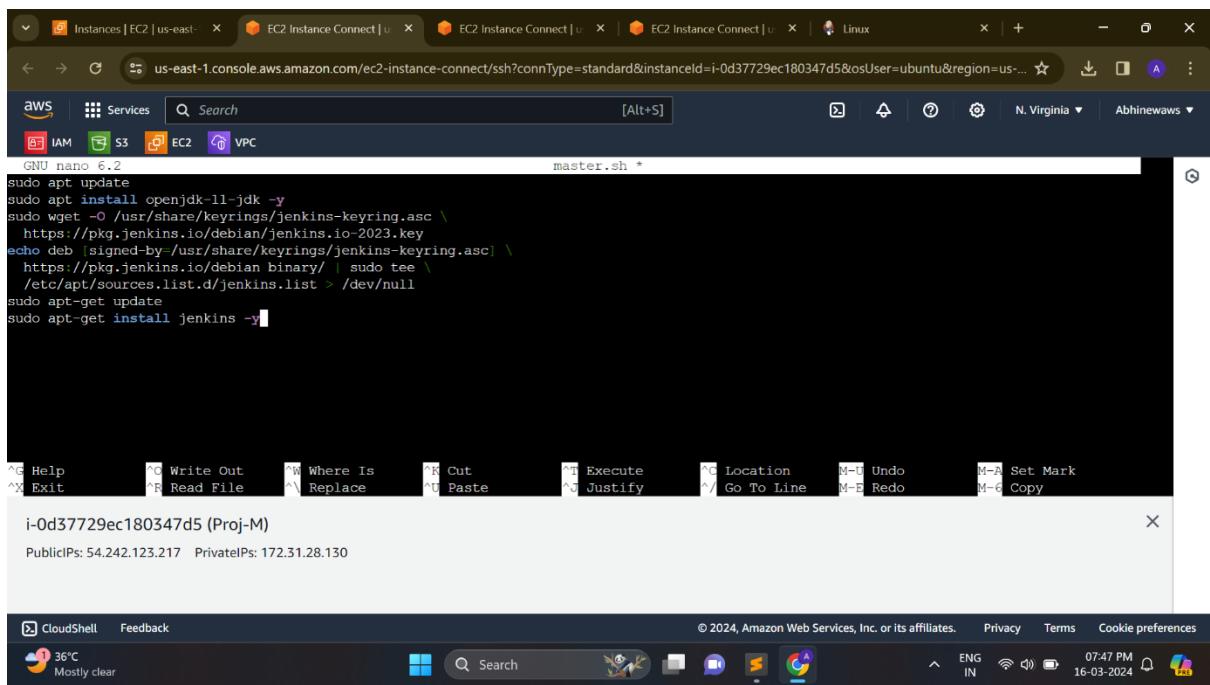
Create a script file for slave instance.

A screenshot of a browser window titled "us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0d37729ec180347d5&osUser=ubuntu®ion=us-...". The address bar shows the URL. The page displays session information: "i-0d37729ec180347d5 (Proj-M)", "Public IPs: 54.242.123.217", and "Private IPs: 172.31.28.130". Below this, there's a terminal window showing command-line interactions. The terminal output includes:

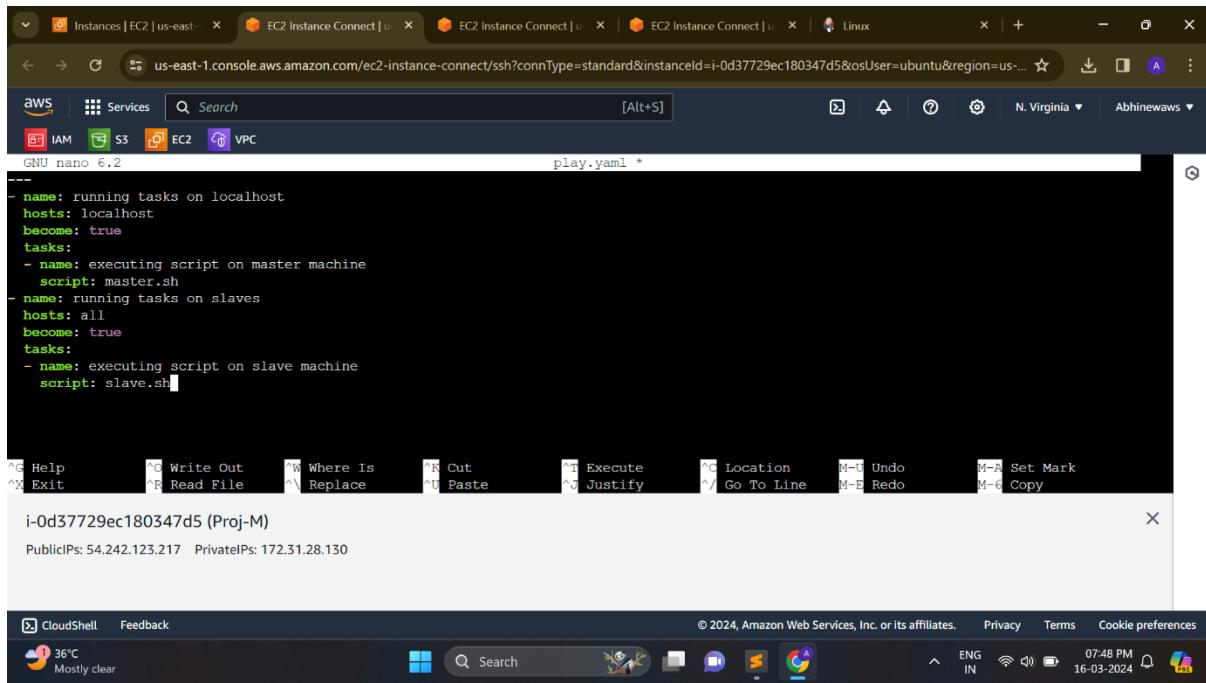
```
GNU nano 6.2
slave.sh *
sudo apt update
sudo apt install openjdk-11-jdk -y
sudo apt install docker.io -y
```



Copy the weekly release commands for jenkins installation.



Script file for master instance.



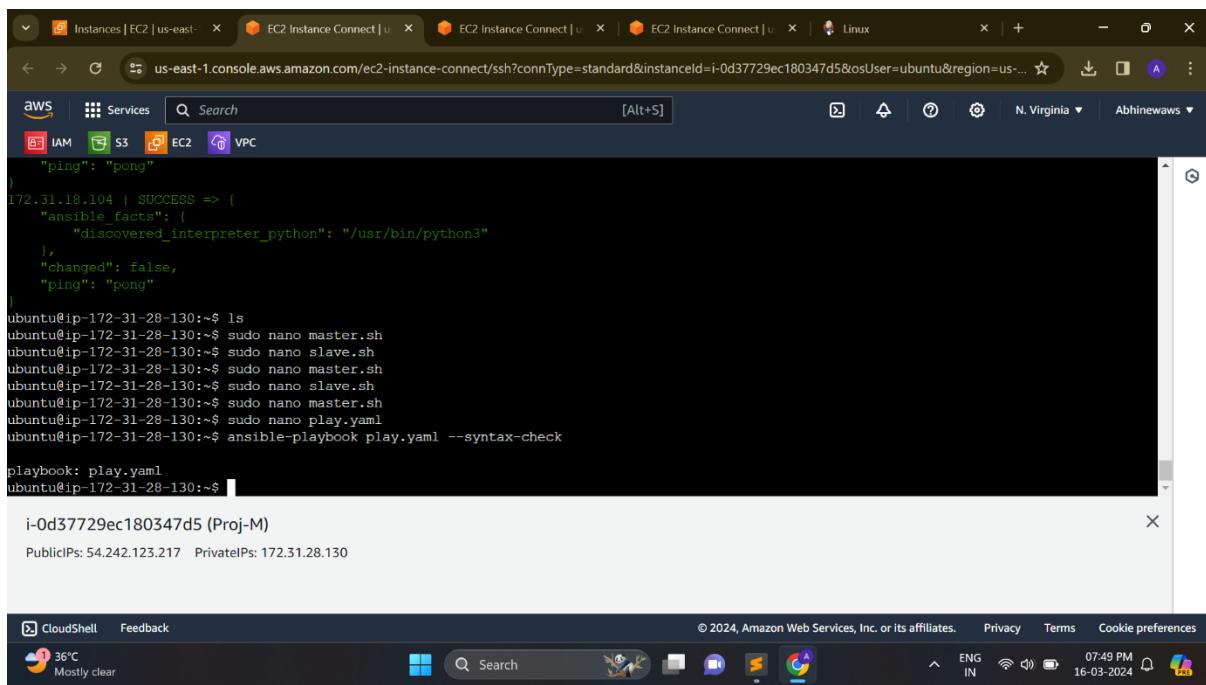
```
play.yaml *  
---  
- name: running tasks on localhost  
  hosts: localhost  
  become: true  
  tasks:  
    - name: executing script on master machine  
      script: master.sh  
- name: running tasks on slaves  
  hosts: all  
  become: true  
  tasks:  
    - name: executing script on slave machine  
      script: slave.sh
```

GNU nano 6.2

^G Help ^C Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo ^X Exit ^R Read File ^V Paste ^T Justify ^A Go To Line M-P Redo M-B Set Mark M-C Copy

i-0d37729ec180347d5 (Proj-M)
Public IPs: 54.242.123.217 Private IPs: 172.31.28.130

Yaml file to install the required softwares.



```
"ping": "pong"  
}  
172.31.10.4 | SUCCESS => {  
  "ansible_facts": {  
    "discovered_interpreter_python": "/usr/bin/python3"  
  },  
  "changed": false,  
  "ping": "pong"  
}  
ubuntu@ip-172-31-28-130:~$ ls  
ubuntu@ip-172-31-28-130:~$ sudo nano master.sh  
ubuntu@ip-172-31-28-130:~$ sudo nano slave.sh  
ubuntu@ip-172-31-28-130:~$ sudo nano master.sh  
ubuntu@ip-172-31-28-130:~$ sudo nano slave.sh  
ubuntu@ip-172-31-28-130:~$ sudo nano master.sh  
ubuntu@ip-172-31-28-130:~$ sudo nano play.yaml  
ubuntu@ip-172-31-28-130:~$ ansible-playbook play.yaml --syntax-check  
  
playbook: play.yaml  
ubuntu@ip-172-31-28-130:~$
```

i-0d37729ec180347d5 (Proj-M)
Public IPs: 54.242.123.217 Private IPs: 172.31.28.130

Play the ansible playbooks by syntax and dr run.

The screenshot shows a CloudShell window with the following terminal session:

```
Instances | EC2 | us-east-1 | EC2 Instance Connect | EC2 Instance Connect | EC2 Instance Connect | Linux
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0d37729ec180347d5&cosUser=ubuntu&region=us-... ★
AWS Services Search [Alt+S] N. Virginia Abhinewaws
IAM S3 EC2 VPC

"ping": "pong"
}
172.31.10.104 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu@ip-172-31-28-130:~$ ls
ubuntu@ip-172-31-28-130:~$ sudo nano master.sh
ubuntu@ip-172-31-28-130:~$ sudo nano slave.sh
ubuntu@ip-172-31-28-130:~$ sudo nano master.sh
ubuntu@ip-172-31-28-130:~$ sudo nano slave.sh
ubuntu@ip-172-31-28-130:~$ sudo nano play.yaml
ubuntu@ip-172-31-28-130:~$ ansible-playbook play.yaml --syntax-check

playbook: play.yaml
ubuntu@ip-172-31-28-130:~$ ansible-playbook play.yaml --check

i-0d37729ec180347d5 (Proj-M)
PublicIPs: 54.242.123.217 PrivateIPs: 172.31.28.130
```

```
TASK [executing script on master machine] ****
skipping: [localhost]

PLAY [running tasks on slaves] ****

TASK [Gathering Facts] ****
ok: [172.31.18.104]
ok: [172.31.30.169]

TASK [executing script on slave machine] ****
skipping: [172.31.18.104]
skipping: [172.31.30.169]

PLAY RECAP ****
172.31.18.104      : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
172.31.30.169      : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
localhost          : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

ubuntu@ip-172-31-28-130:~$
```

Ansible playbook runned successfully.

```
ok: [172.31.30.169]

TASK [executing script on slave machine] *****
skipping: [172.31.18.104]
skipping: [172.31.30.169]

PLAY RECAP *****
172.31.18.104 : ok=1    changed=0      unreachable=0      failed=0      skipped=1      rescued=0      ignored=0
172.31.30.169 : ok=1    changed=0      unreachable=0      failed=0      skipped=1      rescued=0      ignored=0
localhost      : ok=1    changed=0      unreachable=0      failed=0      skipped=1      rescued=0      ignored=0

ubuntu@ip-172-31-28-130:~$ ansible-playbook play.yaml

PLAY [running tasks on localhost] *****
TASK [Gathering Facts] *****
ok: [localhost]

TASK [executing script on master machine] *****

i-0d37729ec180347d5 (Proj-M)
PublicIPs: 54.242.123.217 PrivateIPs: 172.31.28.130
```

```
changed: [localhost]

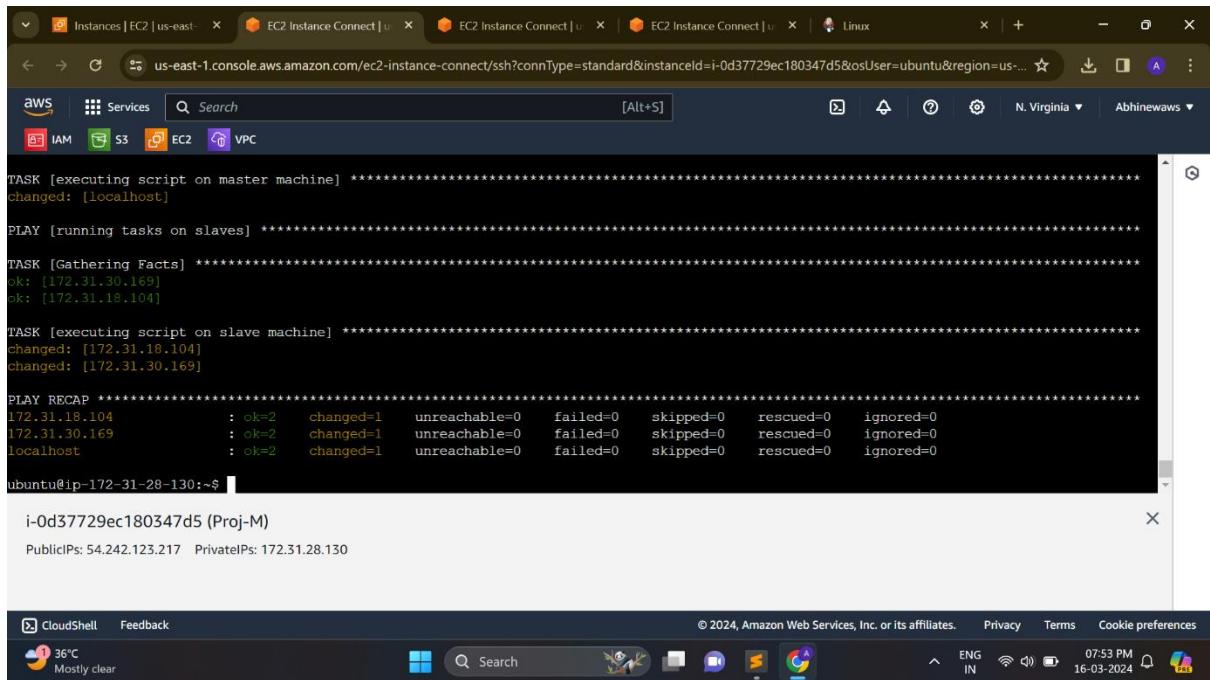
TASK [executing script on master machine] *****
changed: [localhost]

PLAY [running tasks on slaves] *****
TASK [Gathering Facts] *****
ok: [172.31.30.169]
ok: [172.31.18.104]

TASK [executing script on slave machine] *****
changed: [172.31.18.104]
changed: [172.31.30.169]

PLAY RECAP *****
172.31.18.104 : ok=2    changed=1      unreachable=0      failed=0      skipped=0      rescued=0      ignored=0
172.31.30.169 : 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-172-31-28-130:~$ [REDACTED]
i-0d37729ec180347d5 (Proj-M)
PublicIPs: 54.242.123.217 PrivateIPs: 172.31.28.130
```



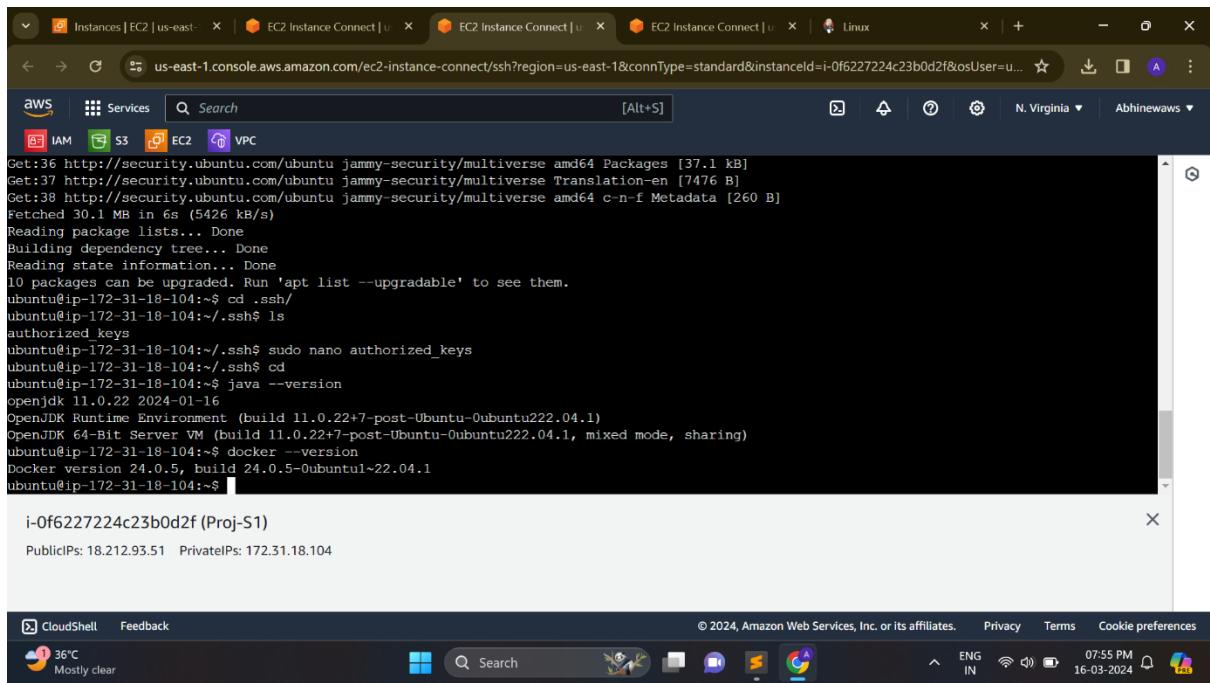
AWS CloudShell interface showing multiple tabs for EC2 Instance Connect. The main terminal window displays the output of an Ansible playbook running across three EC2 instances. The log shows tasks like 'Gathering Facts' and 'executing script on slave machine'. It also lists hosts and their status (ok or changed). The final 'PLAY RECAP' summary shows the count of tasks for each host.

```
TASK [executing script on master machine] ****
changed: [localhost]

PLAY [running tasks on slaves] ****
TASK [Gathering Facts] ****
ok: [172.31.30.169]
ok: [172.31.18.104]

TASK [executing script on slave machine] ****
changed: [172.31.18.104]
changed: [172.31.30.169]

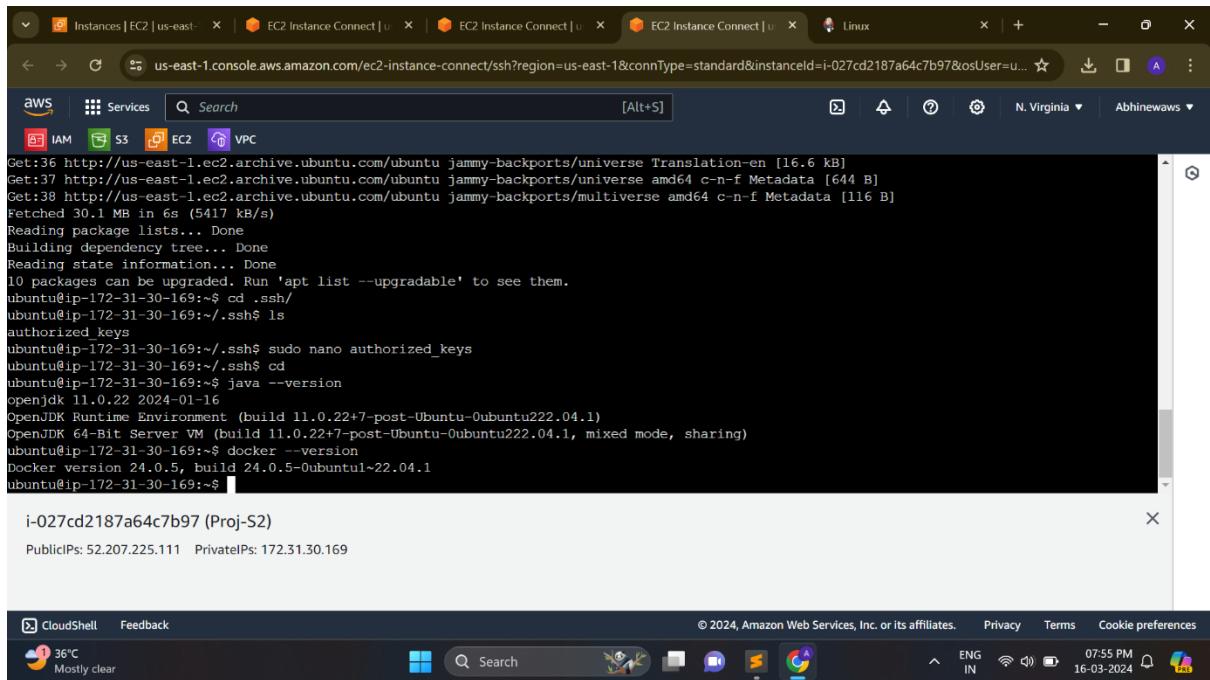
PLAY RECAP ****
172.31.18.104      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.30.169      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
localhost          : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```



AWS CloudShell interface showing multiple tabs for EC2 Instance Connect. The main terminal window displays the output of an 'apt-get update' command followed by the installation of Java. The log shows the download of security packages, the building of dependency trees, and the successful installation of Java 11.

```
Get:36 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37.1 kB]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7476 B]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
Fetched 30.1 MB in 6s (5426 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
10 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-18-104:~$ cd .ssh
ubuntu@ip-172-31-18-104:~/ssh$ ls
authorized_keys
ubuntu@ip-172-31-18-104:~/ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-18-104:~/ssh$ cd
ubuntu@ip-172-31-18-104:~$ java --version
openjdk 11.0.22 2024-01-16
OpenJDK Runtime Environment (build 11.0.22+7-post-Ubuntu-0ubuntu22.04.1)
OpenJDK 64-Bit Server VM (build 11.0.22+7-post-Ubuntu-0ubuntu22.04.1, mixed mode, sharing)
ubuntu@ip-172-31-18-104:~$ docker --version
Docker version 24.0.5, build 24.0.5-0ubuntu1~22.04.1
ubuntu@ip-172-31-18-104:~$
```

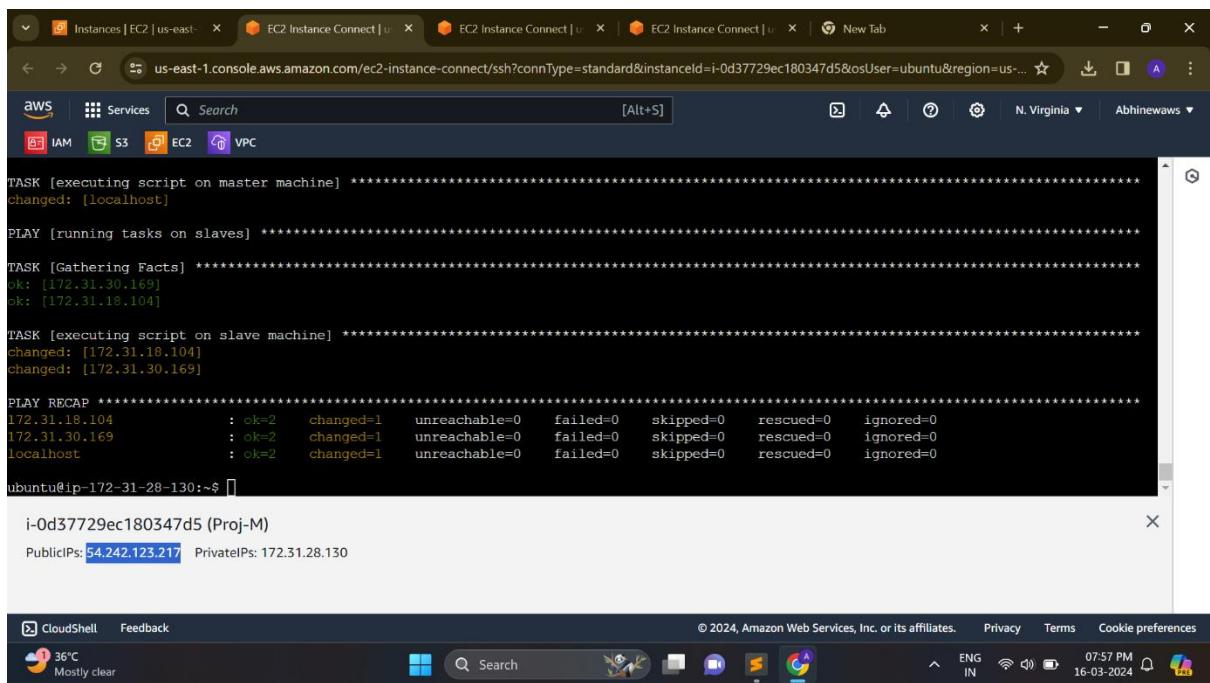
You can check java installed successfully.



```
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.6 kB]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [644 B]
Get:38 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 30.1 MB in 6s (5417 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
10 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-30-169:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-30-169:~/.ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-30-169:~/.ssh$ cd
ubuntu@ip-172-31-30-169:~$ java --version
openjdk 11.0.22 2024-01-16
OpenJDK Runtime Environment (build 11.0.22+7-post-Ubuntu-0ubuntu22.04.1)
OpenJDK 64-Bit Server VM (build 11.0.22+7-post-Ubuntu-0ubuntu22.04.1, mixed mode, sharing)
ubuntu@ip-172-31-30-169:~$ docker --version
Docker version 24.0.5, build 24.0.5-0ubuntu1~22.04.1
ubuntu@ip-172-31-30-169:~$
```

i-027cd2187a64c7b97 (Proj-S2)
PublicIPs: 52.207.225.111 PrivateIPs: 172.31.30.169

You can check docker installed successfully.



```
TASK [executing script on master machine] *****
changed: [localhost]

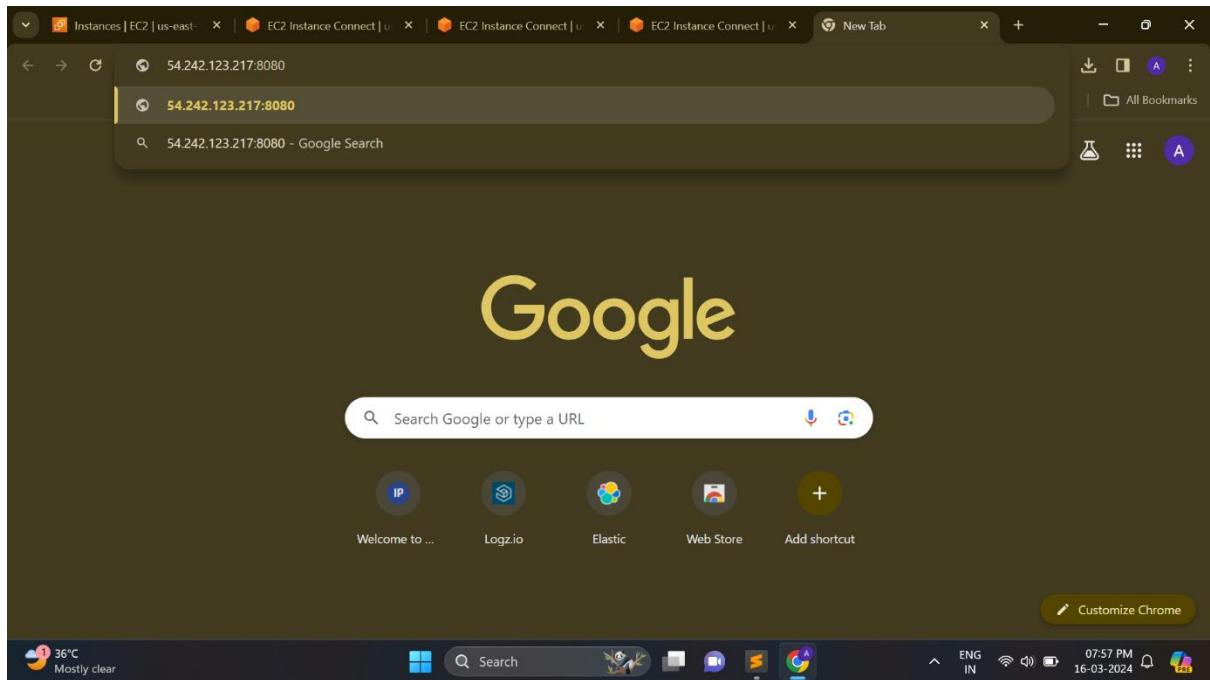
PLAY [running tasks on slaves] *****
TASK [Gathering Facts] *****
ok: [172.31.30.169]
ok: [172.31.18.104]

TASK [executing script on slave machine] *****
changed: [172.31.18.104]
changed: [172.31.30.169]

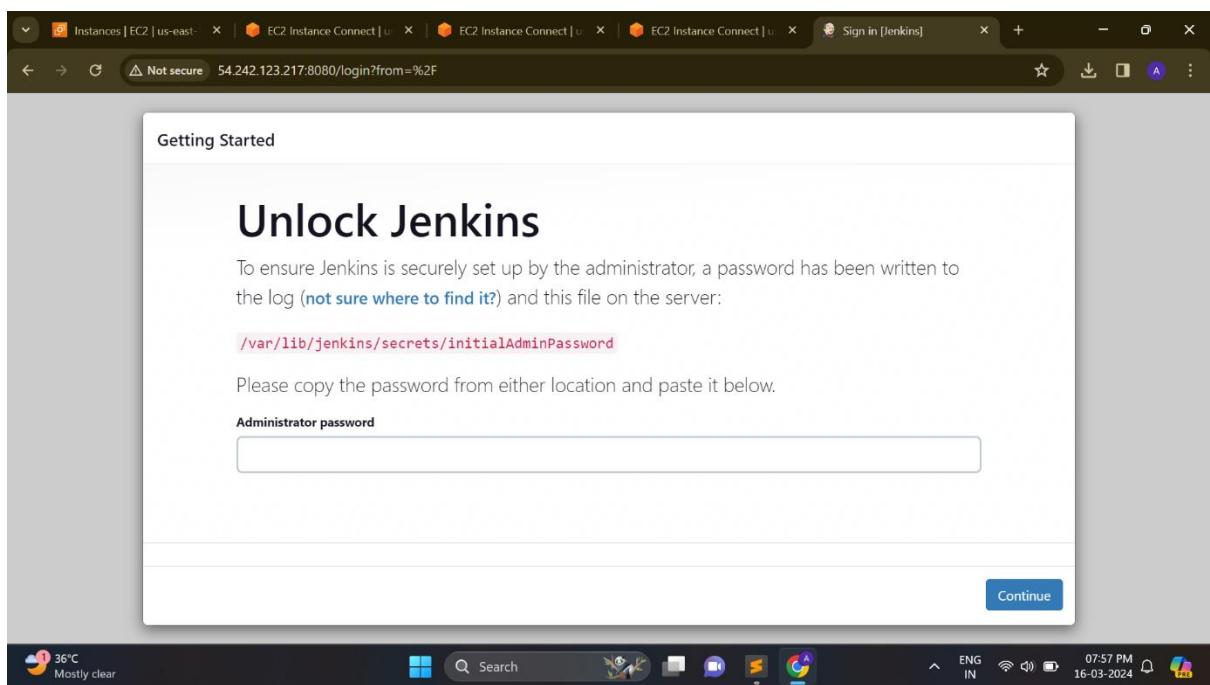
PLAY RECAP *****
172.31.18.104 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.30.169 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
localhost       : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

ubuntu@ip-172-31-28-130:~\$

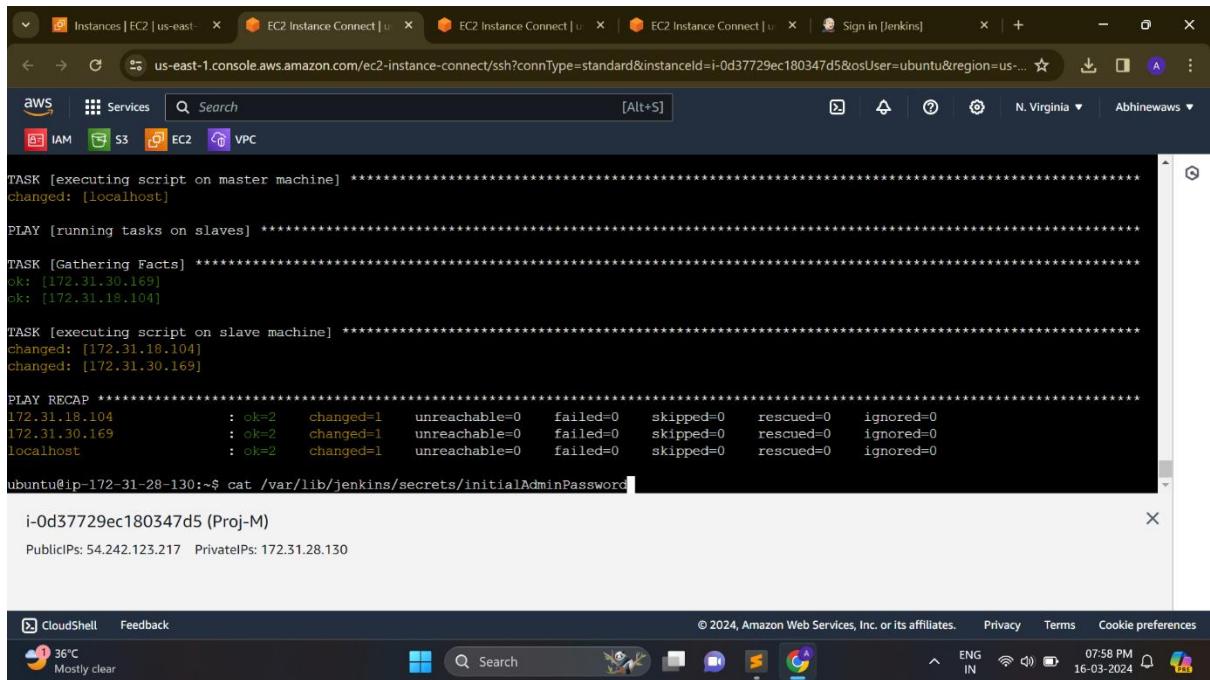
i-0d37729ec180347d5 (Proj-M)
PublicIPs: 54.242.123.217 PrivateIPs: 172.31.28.130



Paste the master instance public ip address to install jenkins.



Unlock the jenkins using jenkins password.



```
TASK [executing script on master machine] ****
changed: [localhost]

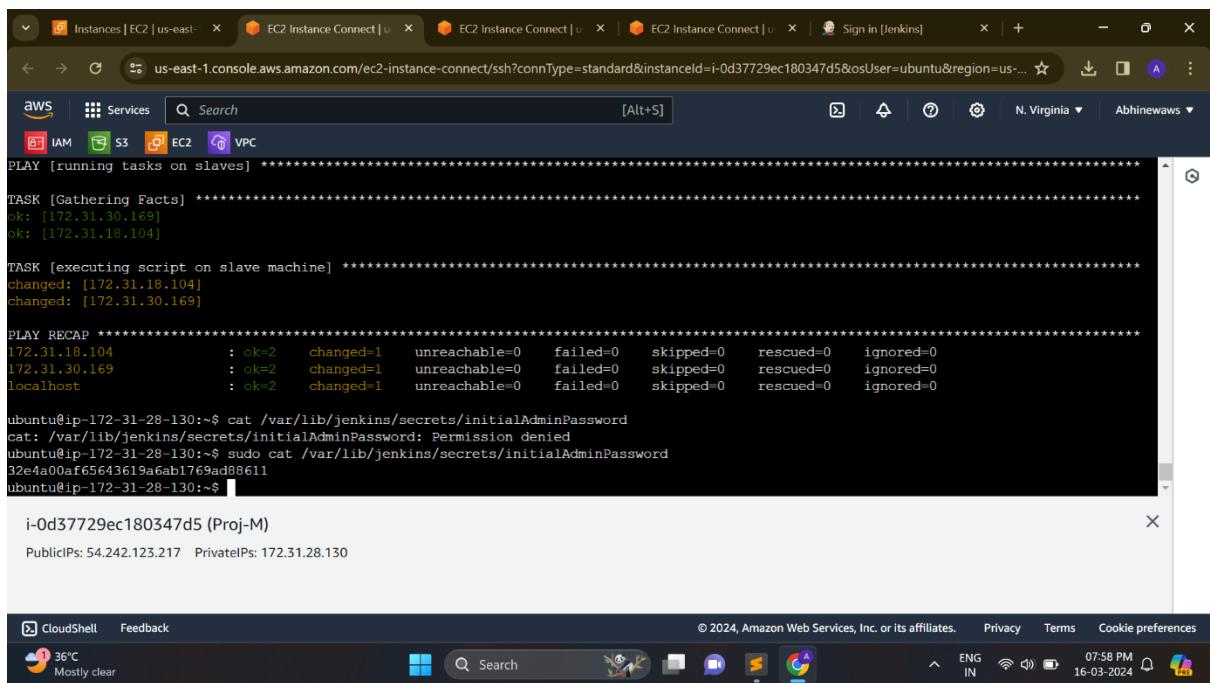
PLAY [running tasks on slaves] ****
TASK [Gathering Facts] ****
ok: [172.31.30.169]
ok: [172.31.18.104]

TASK [executing script on slave machine] ****
changed: [172.31.18.104]
changed: [172.31.30.169]

PLAY RECAP ****
172.31.18.104      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.30.169      : 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-172-31-28-130:~$ cat /var/lib/jenkins/secrets/initialAdminPassword
```

To get jenkins password we need to go into some directories.



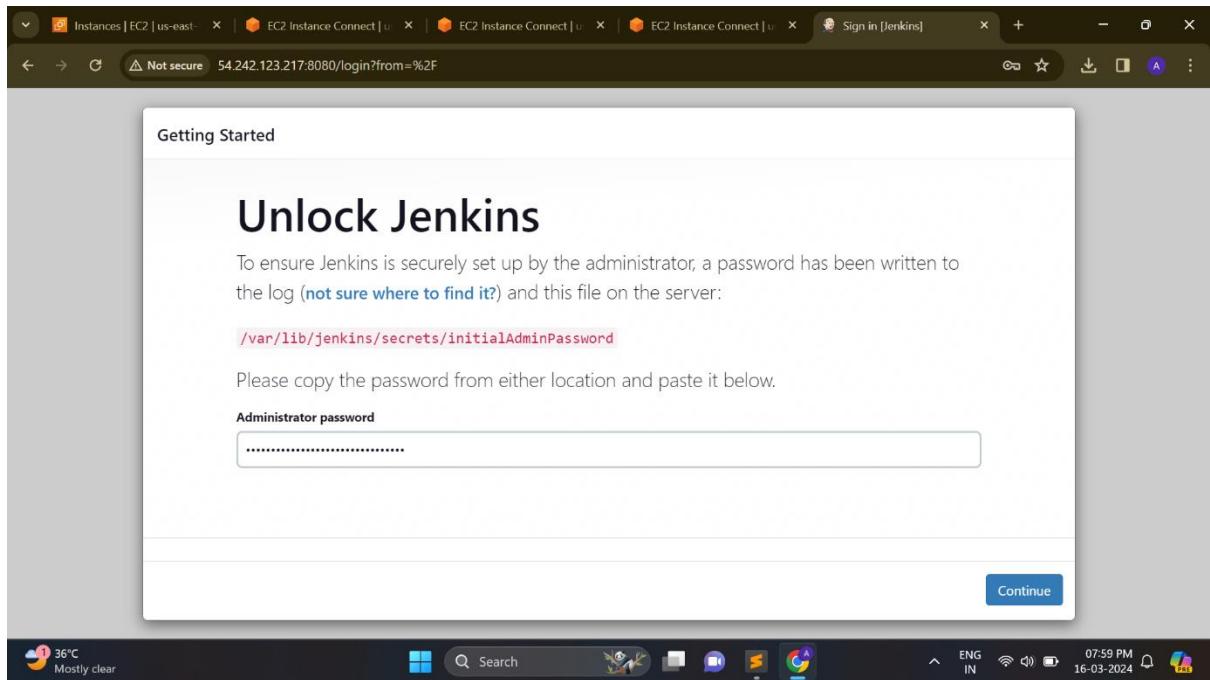
```
PLAY [running tasks on slaves] ****
TASK [Gathering Facts] ****
ok: [172.31.30.169]
ok: [172.31.18.104]

TASK [executing script on slave machine] ****
changed: [172.31.18.104]
changed: [172.31.30.169]

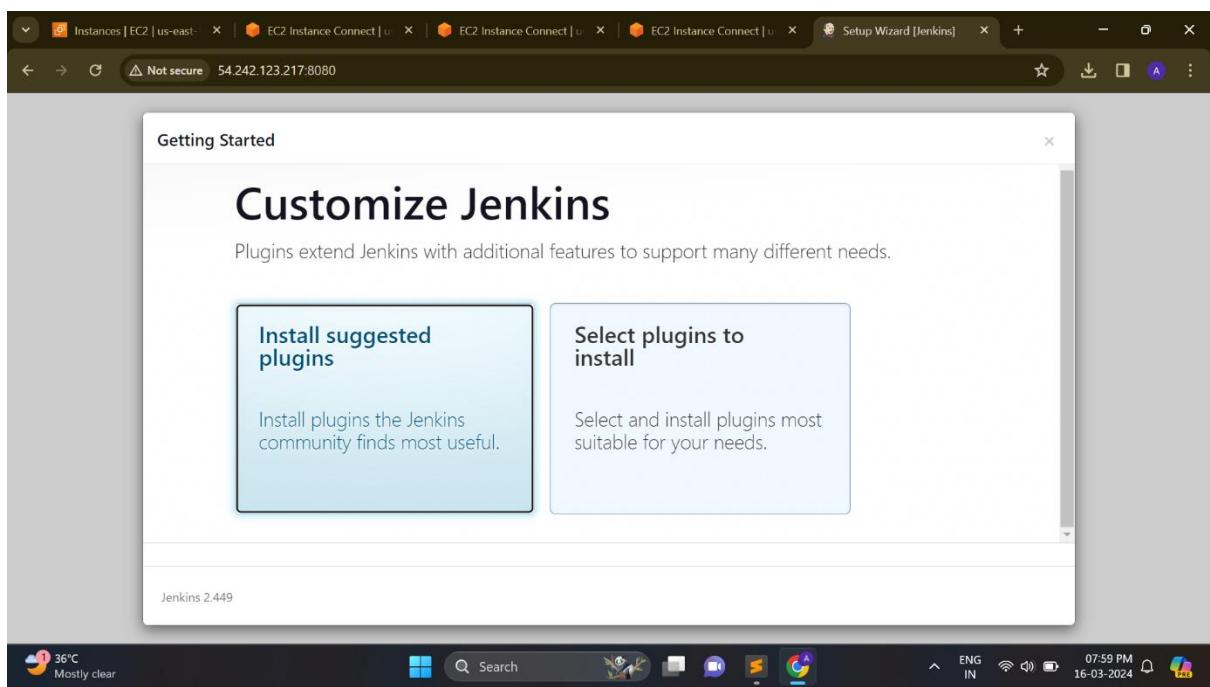
PLAY RECAP ****
172.31.18.104      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.30.169      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
localhost          : ok=2    changed=1    unreachable=0    failed=0   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

ubuntu@ip-172-31-28-130:~$ cat /var/lib/jenkins/secrets/initialAdminPassword
cat: /var/lib/jenkins/secrets/initialAdminPassword: Permission denied
ubuntu@ip-172-31-28-130:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
32ea0af65643619a6ab1769ad8611
ubuntu@ip-172-31-28-130:~$
```

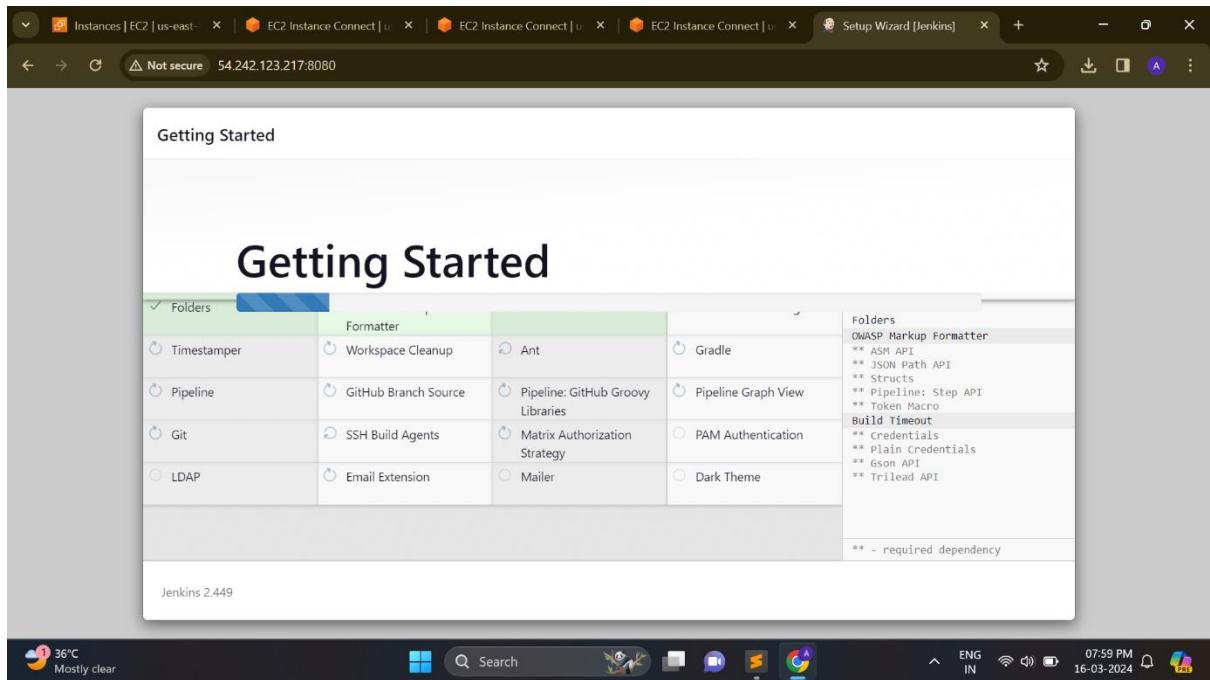
Copy the jenkins password.



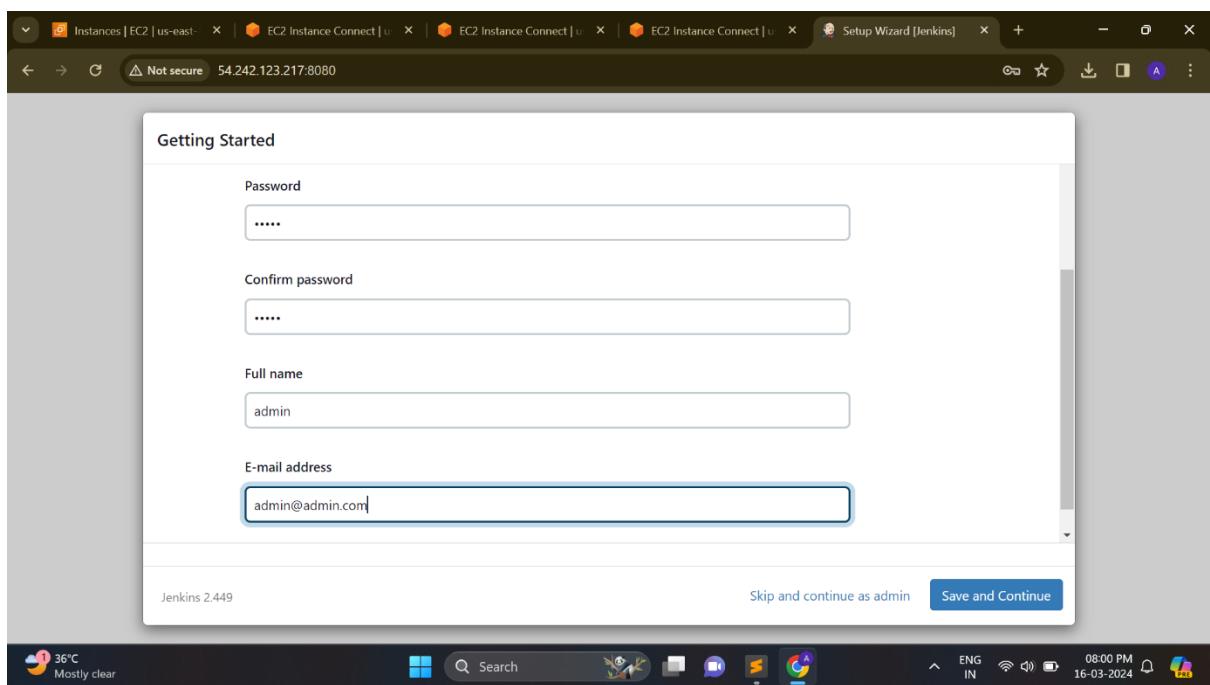
Paste the password and click on continue.



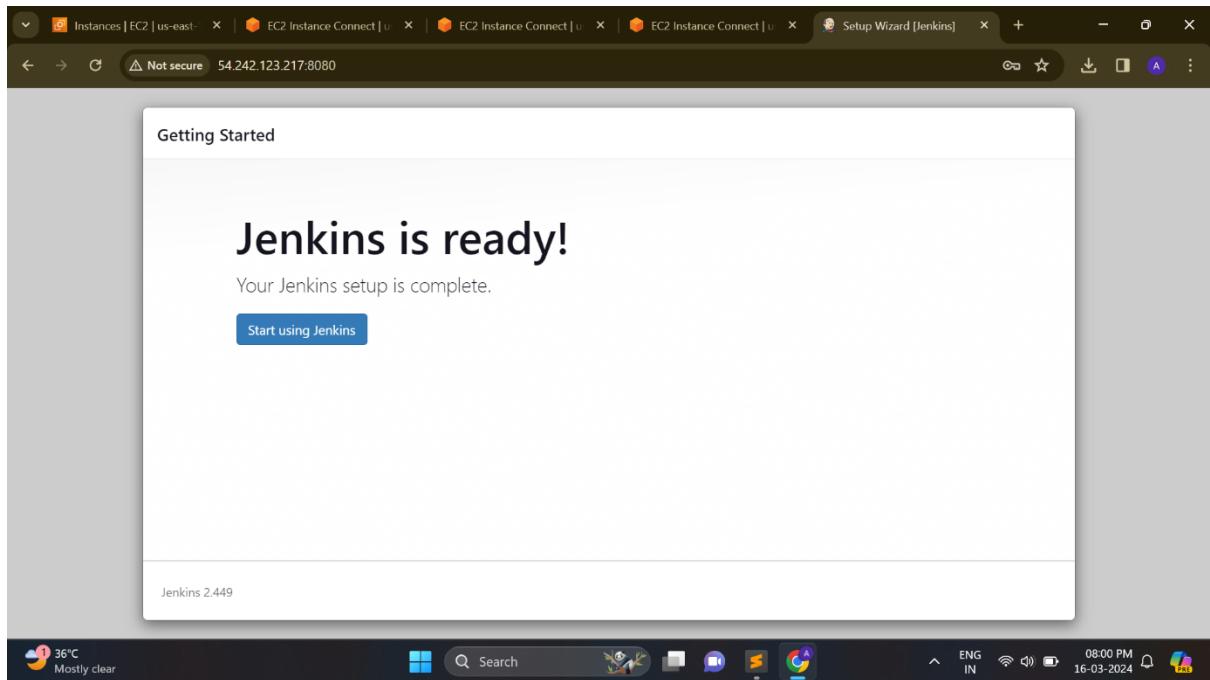
Click on install suggested plugins.



Getting started with jenkins.



Confirm the password and click on save and continue.



Jenkins is ready!

Click on manage jenkins.

The screenshot shows the Jenkins Manage Jenkins interface. At the top, there is a yellow banner warning about Java 11 end-of-life, stating: "Java 11 end of life in Jenkins" and "You are running Jenkins on Java 11, support for which will end on or after Sep 30, 2024. Refer to [the documentation](#) for more details." Below the banner, there are several configuration sections:

- Build Queue:** No builds in the queue.
- Build Executor Status:** 1 Idle, 2 Idle.
- System Configuration:**
 - System:** Configure global settings and paths.
 - Tools:** Configure tools, their locations and automatic installers.
- Plugins:** Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- Clouds:** Add, remove, and configure cloud instances to provision agents on-demand.
- Nodes:** Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Appearance:** Configure the look and feel of Jenkins.

At the bottom of the page, there is a navigation bar with links like "Dashboard", "Manage Jenkins", and "Nodes". The status bar at the bottom right shows the date and time as "16-03-2024 08:04 PM".

Click on nodes.

The screenshot shows the Jenkins Nodes page. The main title is "Nodes". There is a table displaying the current nodes:

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
1	Built-In Node	Linux (amd64)	In sync	4.73 GiB	1.0 B	4.73 GiB	0ms
	Data obtained	11 min	11 min	11 min	11 min	11 min	11 min

Below the table, there is a "New Node" button and a "Configure Monitors" link. On the left side, there are sections for "Build Queue" (No builds in the queue) and "Build Executor Status" (1 Idle, 2 Idle). The status bar at the bottom right shows the date and time as "16-03-2024 08:04 PM".

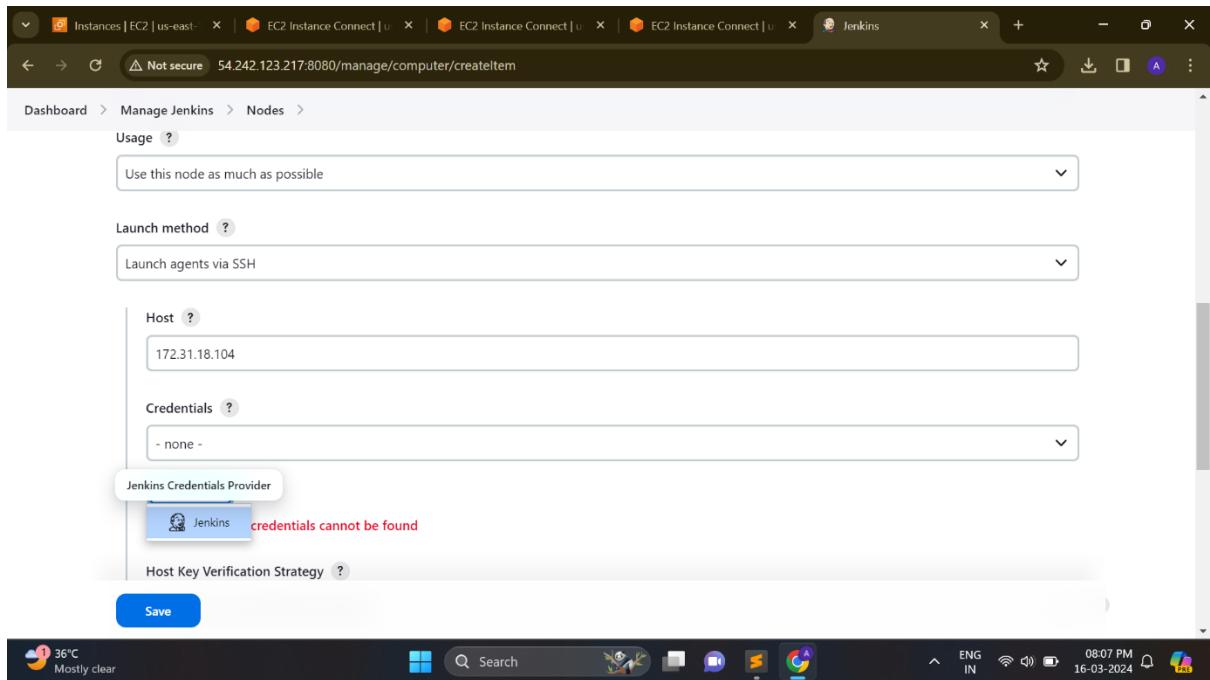
Click on create new node.

The screenshot shows the Jenkins 'New node' configuration page. The node name is set to 'slave1'. The type is selected as 'Permanent Agent', which is described as adding a plain, permanent agent to Jenkins. A 'Create' button is present at the bottom.

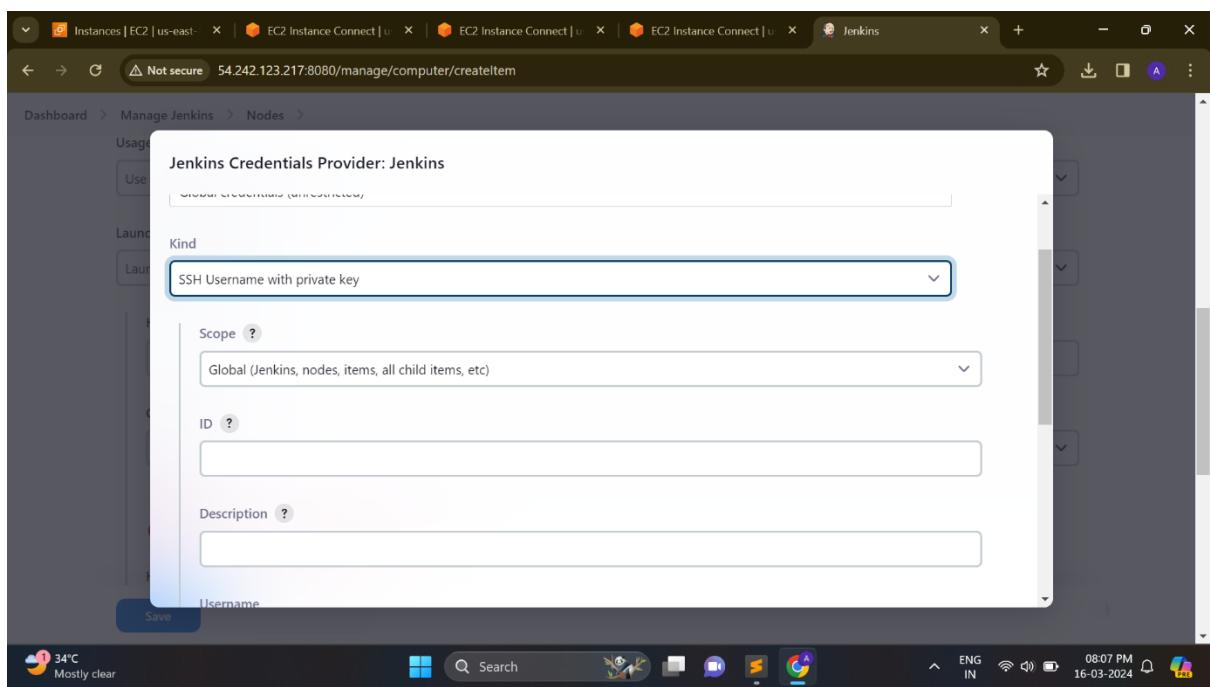
Node name – slave1 – type – permanent agent – click on create.

The screenshot shows the Jenkins 'createItem' configuration page. It includes fields for 'Number of executors' (set to 1), 'Remote root directory' (set to '/home/ubuntu/jenkins'), and 'Usage' (set to 'Use this node as much as possible'). A 'Save' button is located at the bottom.

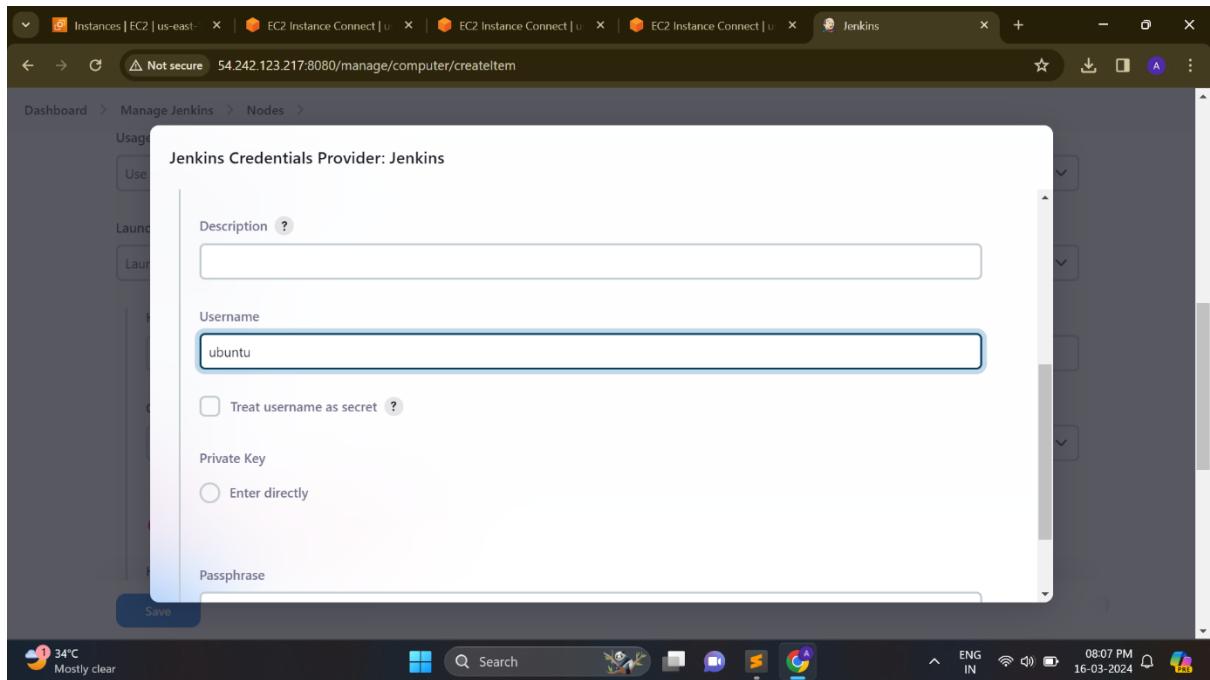
Provide remote root directory - /home/ubuntu/jenkins, click on save.



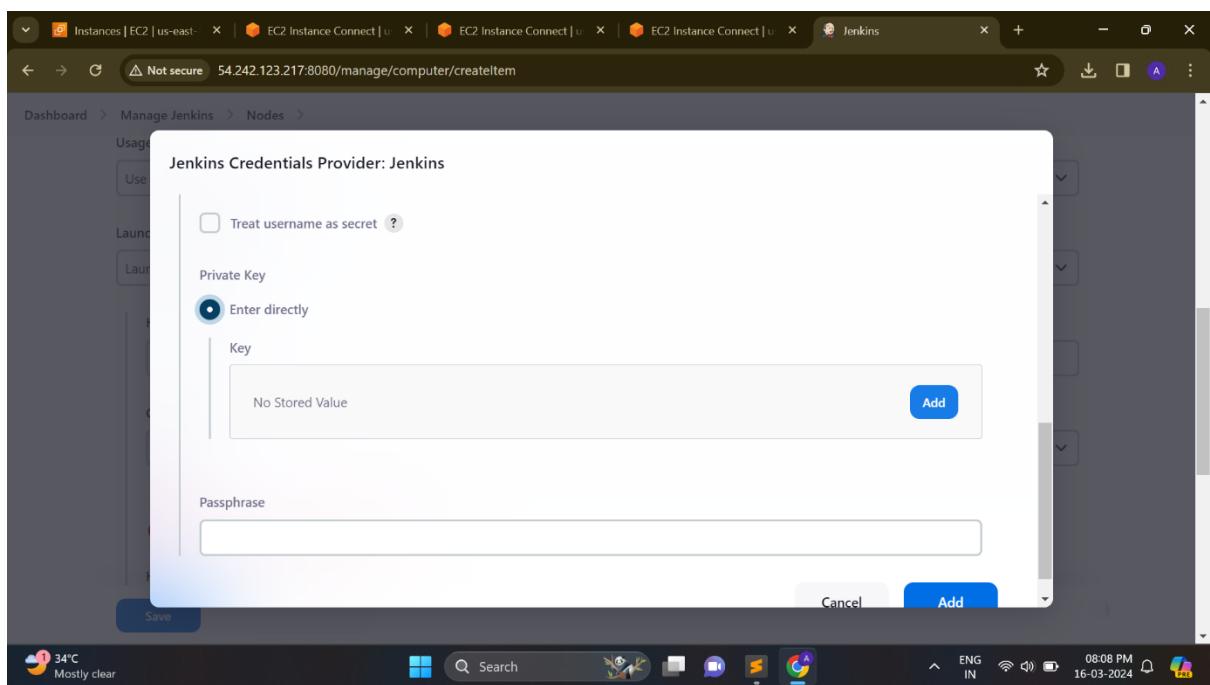
Click on credentials – click on jenkins.



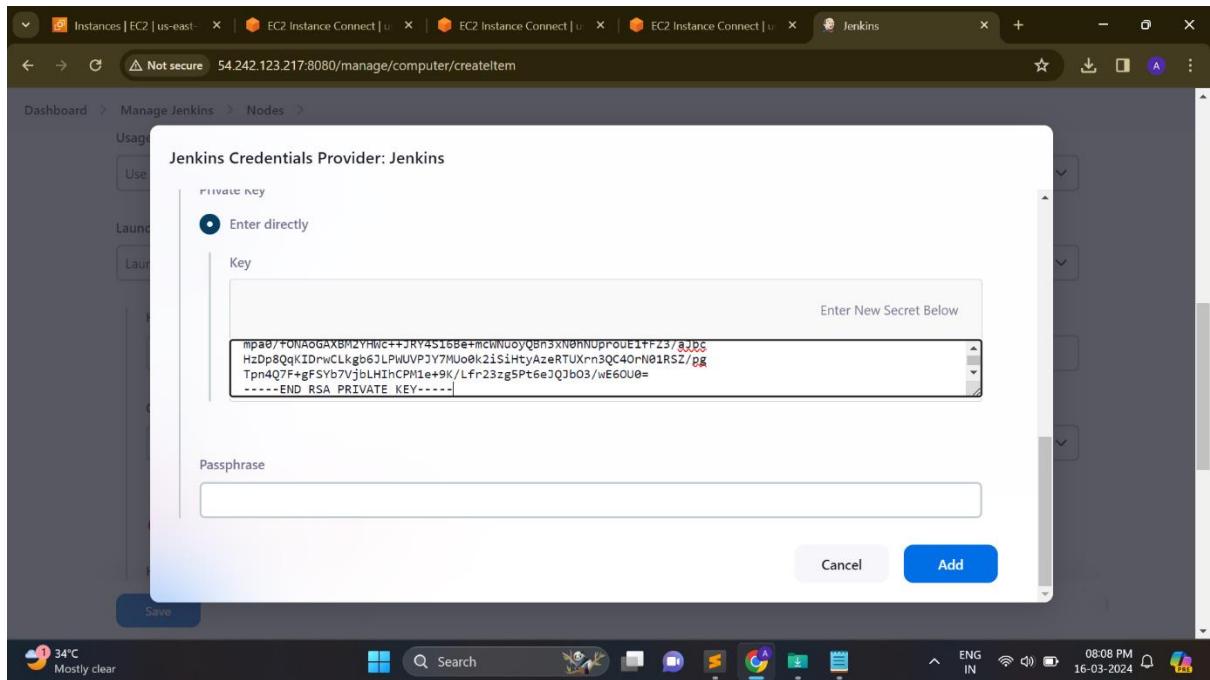
Click on SSH username with private key.



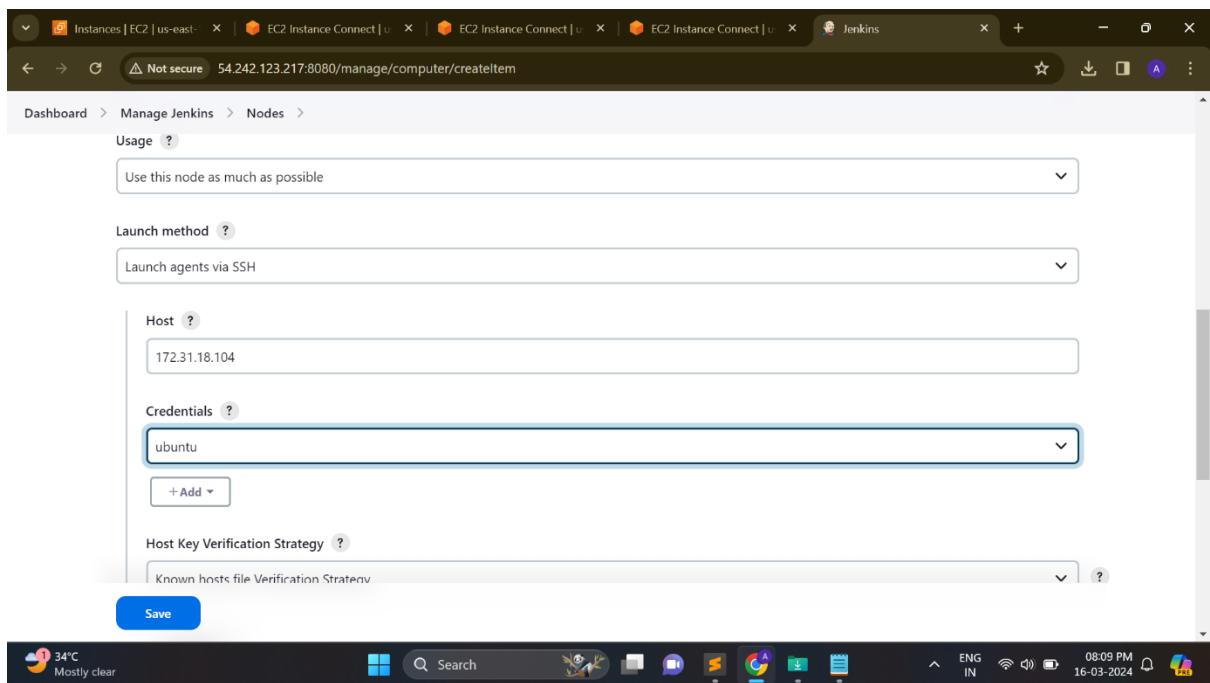
Username - ubuntu



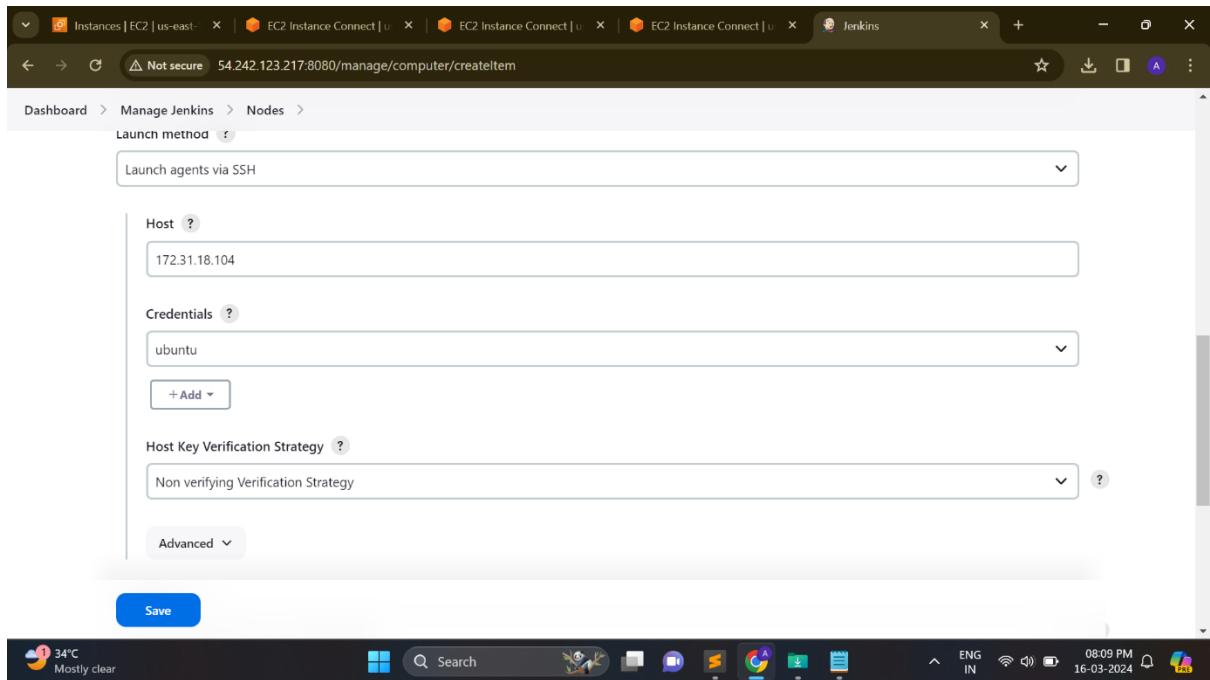
Private key – click on enter directly – click on add.



Add key - click on add.



Click on ubuntu.

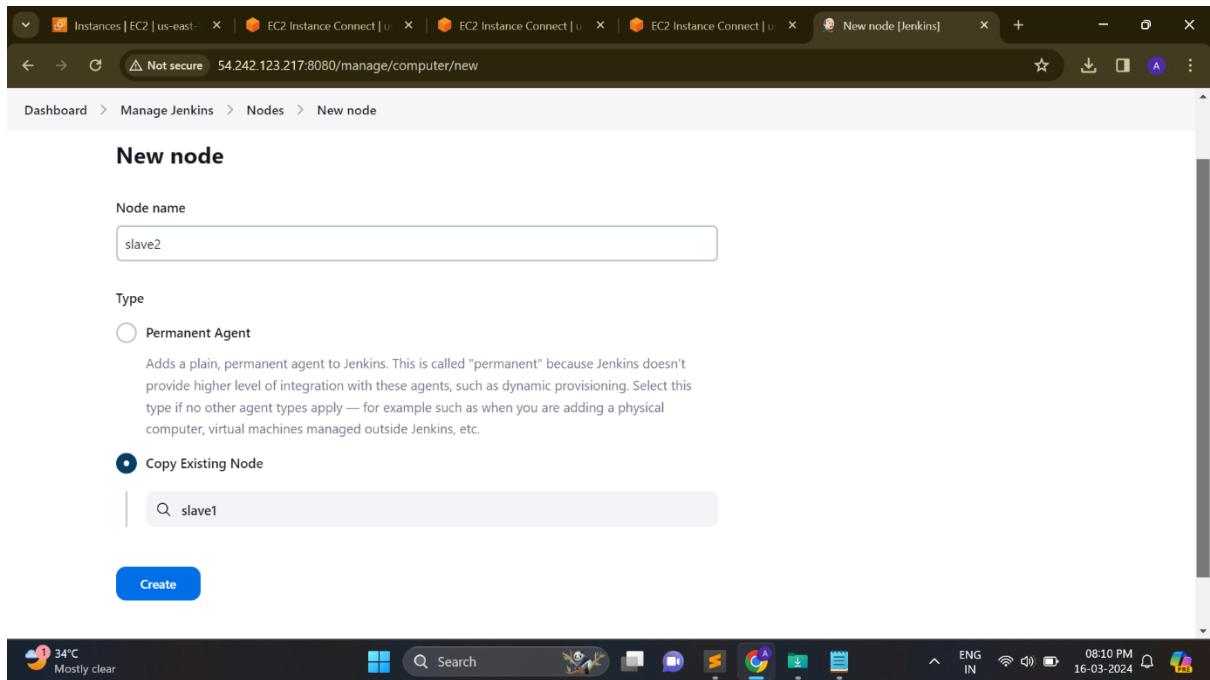


Click on save.

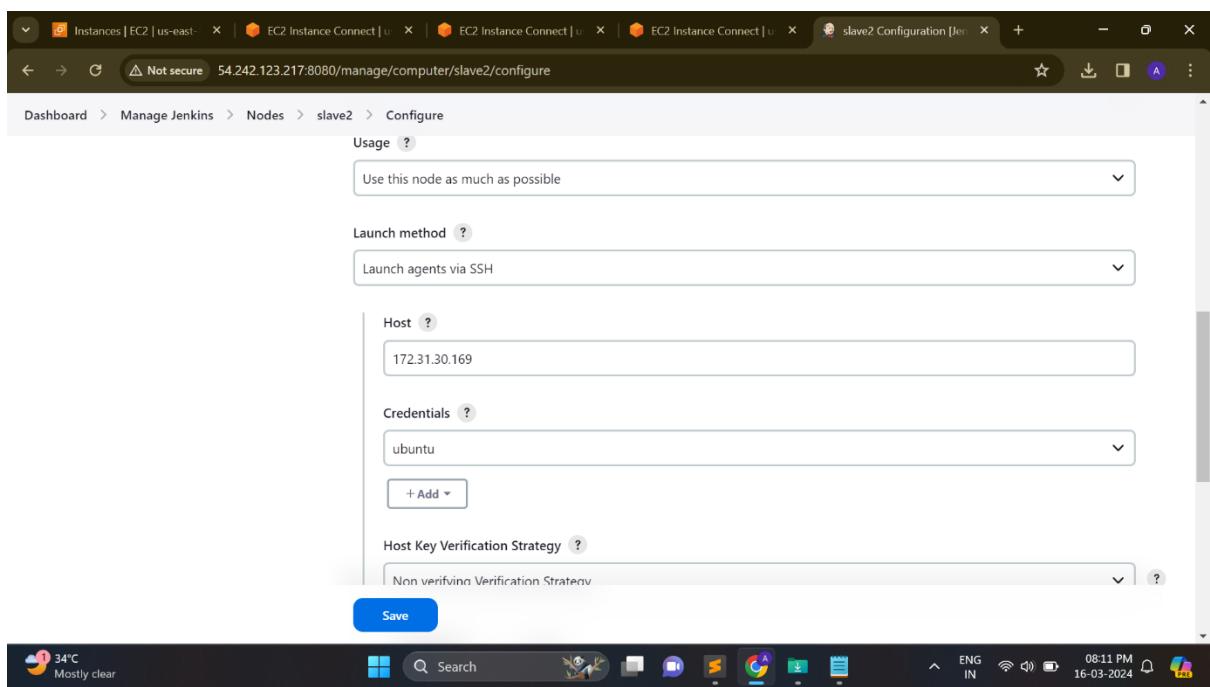
The screenshot shows the Jenkins 'Nodes' management page. It lists two nodes: 'Built-In Node' and 'slave1'. Both nodes are shown as healthy (green) with 0ms response time. A 'New Node' button is visible at the top right.

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
1	Built-In Node	Linux (amd64)	In sync	4.49 GiB	0 B	4.49 GiB	0ms
2	slave1	Linux (amd64)	In sync	5.10 GiB	0 B	5.10 GiB	19ms

Slave 1 attached successfully.



Click on new node – node name – slave 2 - type – copy exixting node.



Change host provider. Click on save.

The screenshot shows the Jenkins 'Nodes' page. On the left, there's a sidebar with 'Build Queue' (empty), 'Build Executor Status' (showing 'Built-In Node' with 1 idle and 2 idle slaves), and 'Clouds' (empty). 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. It lists four nodes: 'Built-In Node' (Linux amd64, In sync, 4.49 GiB free disk, 0B swap, 4.49 GiB temp, 0ms response), 'slave1' (Linux amd64, In sync, 5.10 GiB free disk, 0B swap, 5.10 GiB temp, 2ms response), 'slave2' (Linux amd64, In sync, 5.10 GiB free disk, 0B swap, 5.10 GiB temp, 40ms response), and 'Data obtained' (0.34 sec, 0.34 sec, 0.34 sec, 0.34 sec, 0.34 sec, 0.34 sec). A toolbar at the top right includes '+ New Node' and 'Configure Monitors'. The bottom status bar shows the date and time as 16-03-2024 08:11 PM.

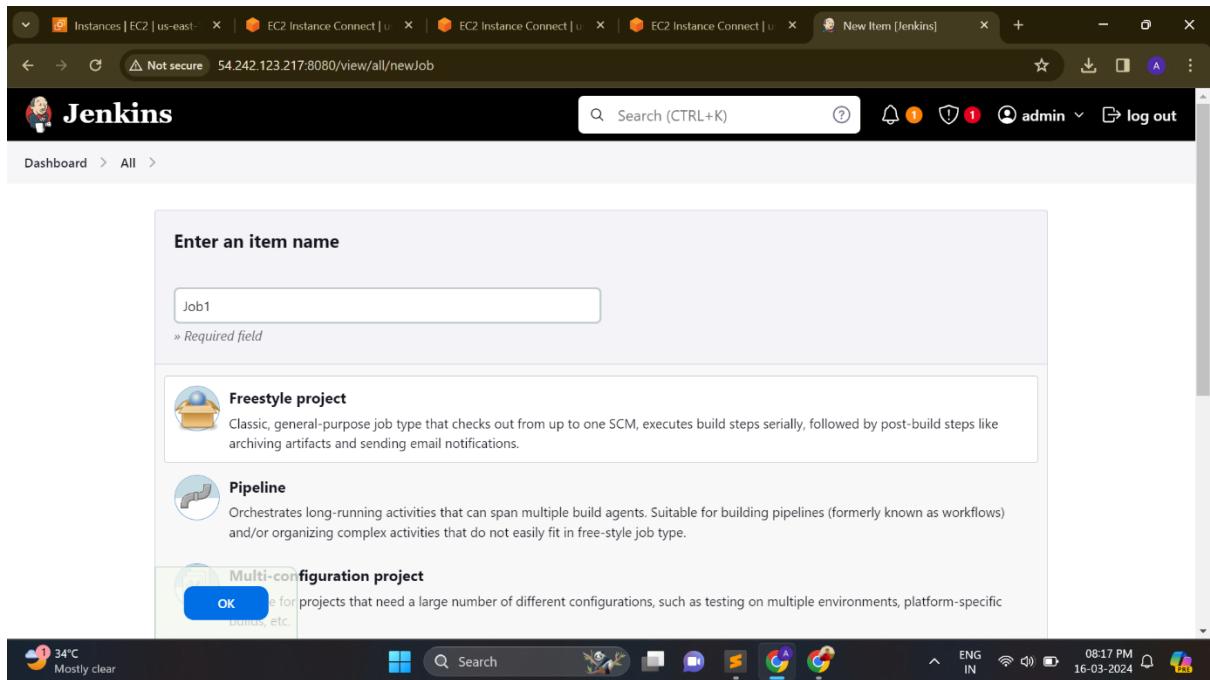
Slave 2 attached successfully.

The screenshot shows a GitHub repository named 'AbhiGundim / website'. The 'Code' tab is selected. On the left, the file tree shows 'master' (containing 'images', 'Dockerfile', and 'index.html'), 'actions', 'issues', and 'pull requests'. The main area is an editor for the 'Dockerfile' in the 'master' branch. The code is as follows:

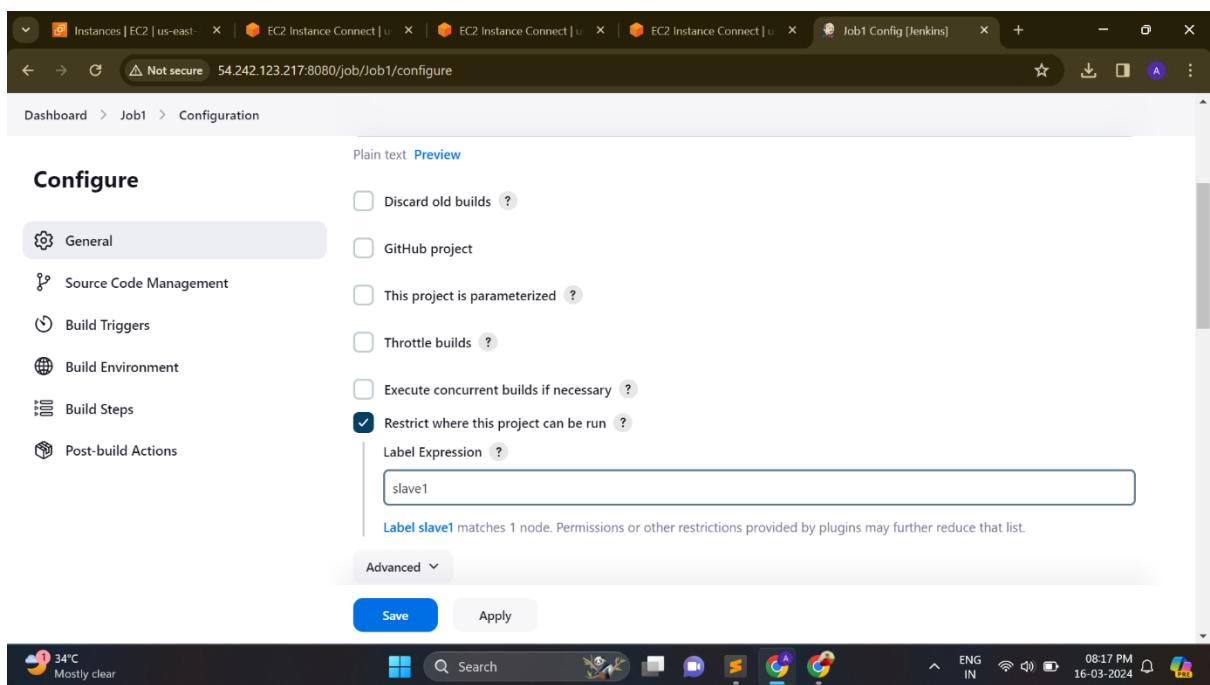
```
1 FROM ubuntu
2 RUN apt update
3 RUN apt install apache2 -y
4 ENTRYPOINT apache2 -D FOREGROUND
5 ADD . /var/www/html/
6
```

The GitHub Copilot interface shows 'Code 55% faster with GitHub Copilot' above the editor. The bottom status bar shows the date and time as 16-03-2024 08:14 PM.

Create a dockerfile in your repository.



Create a job – enter an item name – Job1 – click on freestyle project.



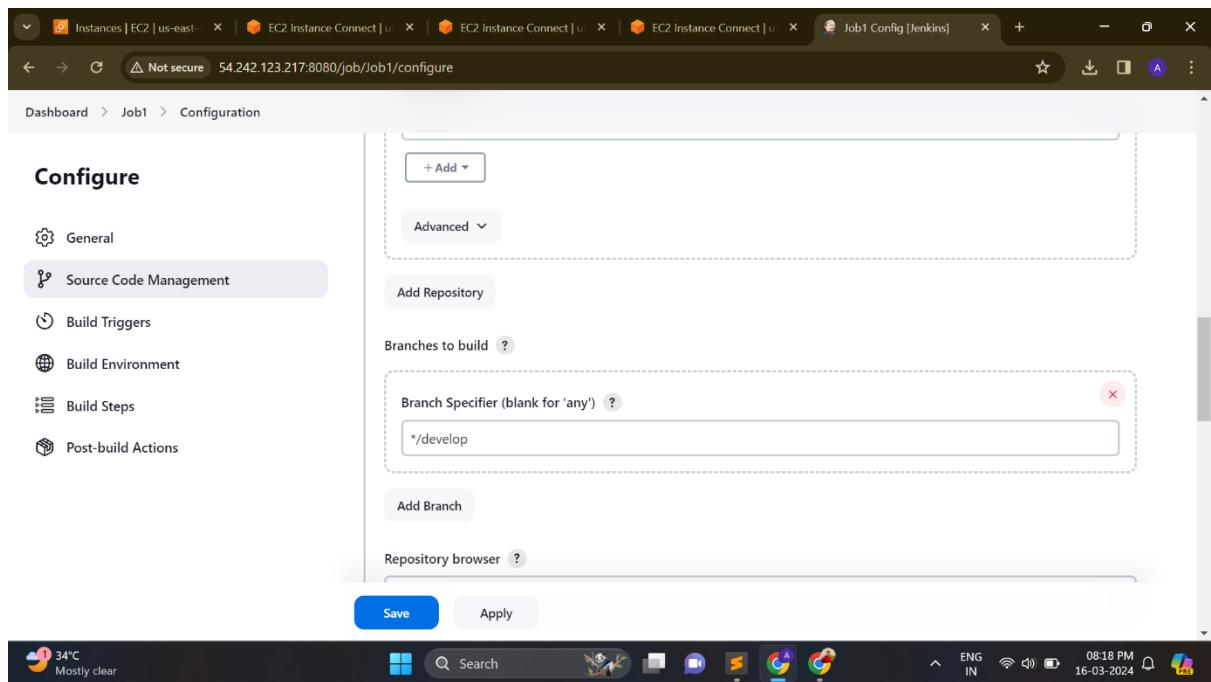
Click on restrict where this project can be run. – slave 1.

The screenshot shows the Jenkins job configuration interface for a job named 'Job1'. The 'Source Code Management' section is selected. It is configured to use 'Git' with the repository URL set to <https://github.com/AbhiGundim/website.git>. There are no credentials defined. The 'Save' and 'Apply' buttons are at the bottom.

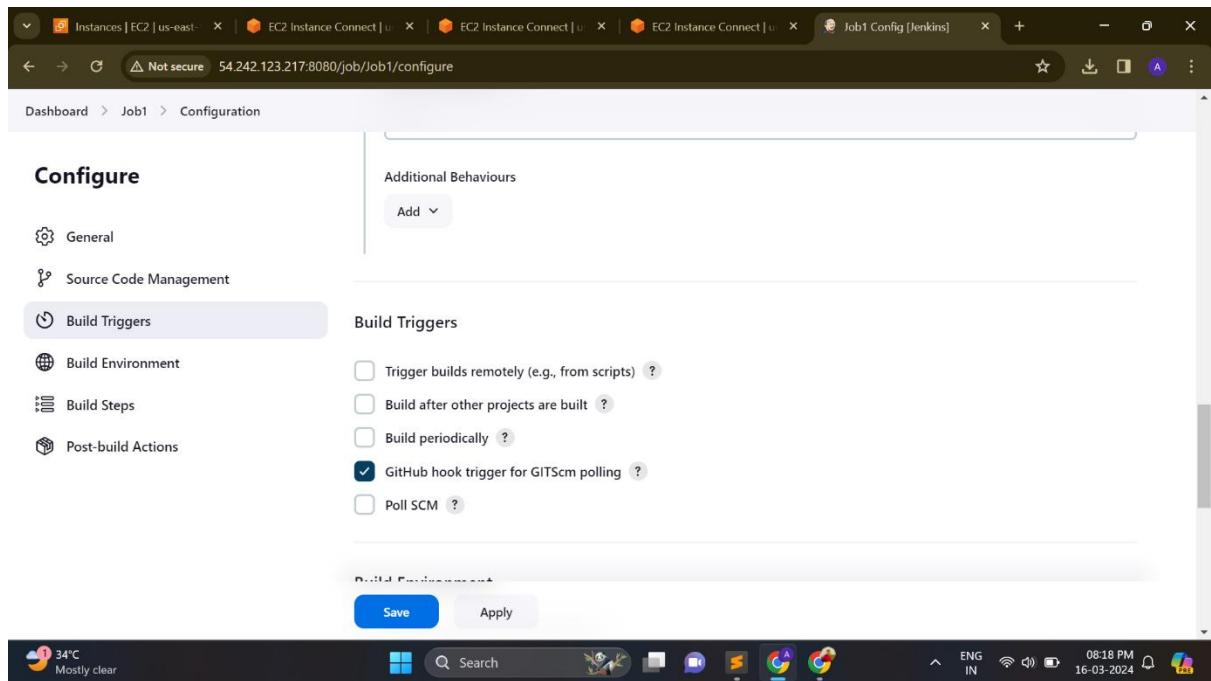
Source code management – Git – provide repository URL.

The screenshot shows the Jenkins job configuration interface for a job named 'Job1'. The 'Source Code Management' section is selected. Under 'Branches to build', the 'Branch Specifier' is set to `*/develop`. The 'Save' and 'Apply' buttons are at the bottom.

Branches to build - `*/develop`

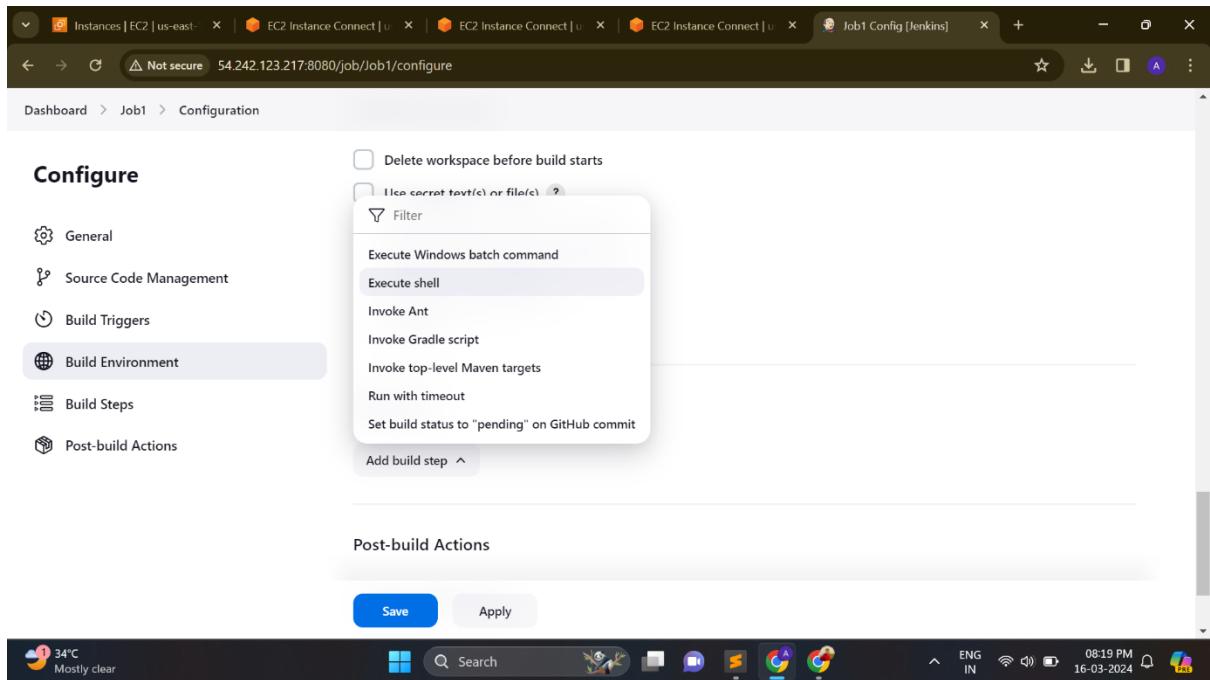


The screenshot shows the Jenkins configuration interface for a job named 'Job1'. The left sidebar lists configuration sections: General, Source Code Management (selected), Build Triggers, Build Environment, Build Steps, and Post-build Actions. The main panel is titled 'Configure' and displays the 'Source Code Management' tab. It includes fields for 'Branch Specifier (blank for 'any')' containing '/develop' and an 'Add Branch' button. At the bottom are 'Save' and 'Apply' buttons.

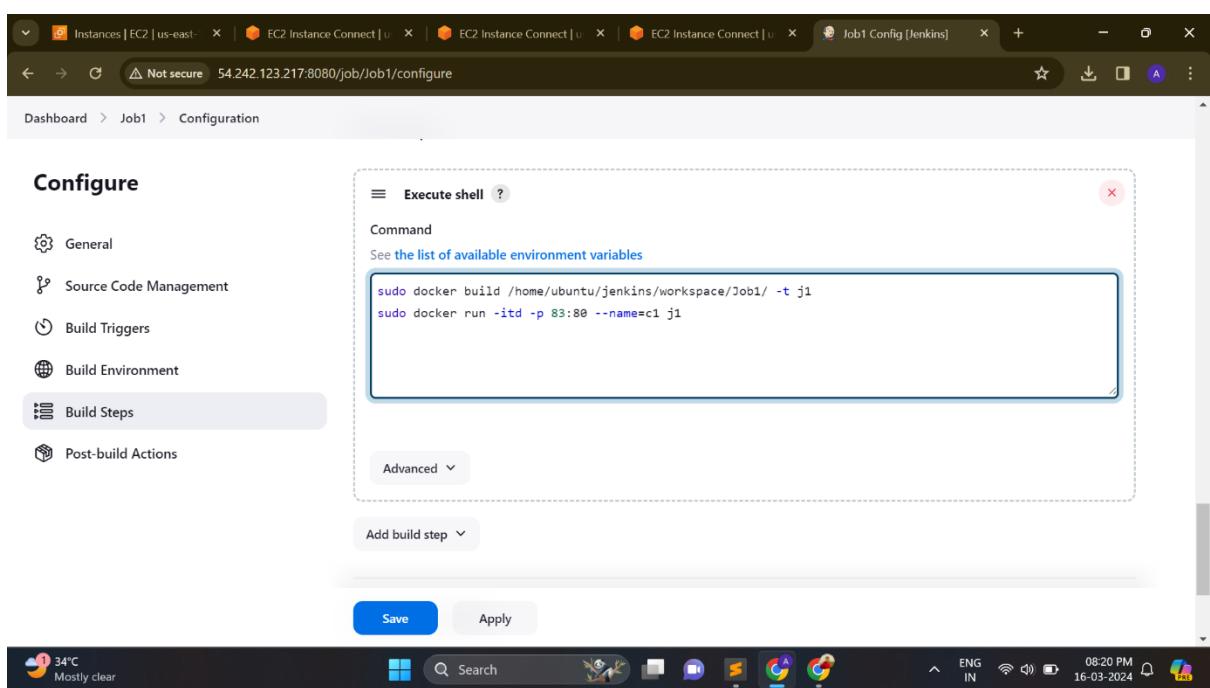


The screenshot shows the Jenkins configuration interface for a job named 'Job1'. The left sidebar lists configuration sections: General, Source Code Management, Build Triggers (selected), Build Environment, Build Steps, and Post-build Actions. The main panel is titled 'Configure' and displays the 'Build Triggers' tab. It includes checkboxes for 'Trigger builds remotely (e.g., from scripts)', 'Build after other projects are built', 'Build periodically', 'GitHub hook trigger for GITScm polling' (which is checked), and 'Poll SCM'. At the bottom are 'Save' and 'Apply' buttons.

In build triggers – click on github hook trigger for GITSCM polling.



Click on execute shell.



Provide your commands. Click on save.

Instances | EC2 | us-east- | EC2 Instance Connect | EC2 Instance Connect | EC2 Instance Connect | Job1 [Jenkins]

Not secure 54.242.123.217:8080/job/Job1/

Dashboard > Job1 > workspace

Disable Project

Build Now

Configure

Delete Project

GitHub Hook Log

Rename

Build History trend

#1 Mar 16, 2024, 2:50 PM

Atom feed for all Atom feed for failures

REST API Jenkins 2.449

34°C Mostly clear Search ENG IN 08:21 PM 16-03-2024

Click on build now.

Instances | EC2 | us-east- | EC2 Instance Connect | EC2 Instance Connect | EC2 Instance Connect | Job1 #1 Console [Jenkins]

Not secure 54.242.123.217:8080/job/Job1/console

Dashboard > Job1 > #1 > Console Output

```
Enabling conf serve-cgi-bin.
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 (2.35-0ubuntu3.6) ...
Processing triggers for ca-certificates (20230311ubuntu0.22.04.1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
Removing intermediate container 9058bb1b43e2
--> 804e913e76e3
Step 4/5 : ENTRYPOINT apachectl -D FOREGROUND
--> Running in 66261825f8f7
Removing intermediate container 66261825f8f7
--> 034f7a79e9ac
Step 5/5 : ADD . /var/www/html/
--> ff137d2e36b4
Successfully built ff137d2e36b4
Successfully tagged j1:latest
+ sudo docker run -itd -p 83:80 --name=c1 j1
ff04497f10042996e12f89a7ae8330d178ef283fbce5c83ef2b082f26daa5821
Finished: SUCCESS
```

ENG IN 08:22 PM 16-03-2024

34°C Mostly clear Search ENG IN 08:22 PM 16-03-2024

The screenshot shows the Jenkins configuration interface for a job named 'Job1'. On the left, a sidebar lists various configuration sections: General, Source Code Management, Build Triggers, Build Environment, Build Steps (which is currently selected), and Post-build Actions. The main area is titled 'Configure' and contains a 'Build Steps' section. Under 'Execute shell', there is a command box containing the following Docker commands:

```
sudo docker rm -f c1
sudo docker build /home/ubuntu/jenkins/workspace/Job1/ -t j1
sudo docker run -itd -p 83:80 --name=c1 j1
```

At the bottom of the configuration page are 'Save' and 'Apply' buttons.

Again give remove command. And click on save.

The screenshot shows the Jenkins 'Permalinks' page for job 'Job1'. It displays a list of recent builds:

- Last build (#1), 2 min 18 sec ago
- Last stable build (#1), 2 min 18 sec ago
- Last successful build (#1), 2 min 18 sec ago
- Last completed build (#1), 2 min 18 sec ago

Below this, the 'Build History' section shows two entries:

- #2 Mar 16, 2024, 2:53 PM
- #1 Mar 16, 2024, 2:50 PM

At the bottom of the page are links for 'Atom feed for all' and 'Atom feed for failures'. The status bar at the bottom right indicates 'Jenkins 2.449'.

Again click on build now.

```
> git fetch --tags --force --progress -- https://github.com/AbhiBounalim/website.git  
+refs/heads/*:refs/remotes/origin/* # timeout=10  
> git rev-parse refs/remotes/origin/develop^{commit} # timeout=10  
Checking out Revision 4e0e78bb64ee4bf68f2f08d84daf09eafe64992c (refs/remotes/origin/develop)  
> git config core.sparsecheckout # timeout=10  
> git checkout -f 4e0e78bb64ee4bf68f2f08d84daf09eafe64992c # timeout=10  
Commit message: "Update Dockerfile"  
> git rev-list --no-walk 4e0e78bb64ee4bf68f2f08d84daf09eafe64992c # timeout=10  
[Job1] $ /bin/sh -xe /tmp/jenkins7412140733262760279.sh  
+ sudo docker rm -f c1  
c1  
+ sudo docker build /home/ubuntu/jenkins/workspace/Job1/ -t j1  
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 294.9kB  
  
Step 1/5 : FROM ubuntu  
--> ca2b0f26964c  
Step 2/5 : RUN apt update  
--> Using cache  
--> 8d69e5308758  
Step 3/5 : RUN apt install apache2 -y  
--> Using cache
```

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Build Steps

Execute shell

Command

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

```
sudo docker rm -f c1  
sudo docker build /home/ubuntu/jenkins/workspace/Job1/ -t j1  
sudo docker run -itd -p 83:80 --name=c1 j1
```

Advanced ▾

Save Apply

```
> git fetch --tags --force --progress -- https://github.com/AbhiBunalmi/website.git  
+refs/heads/*:refs/remotes/origin/* # timeout=10  
> git rev-parse refs/remotes/origin/develop^{commit} # timeout=10  
Checking out Revision 4e0e78bb64ee4bf68f2f08d84daf09eafe64992c (refs/remotes/origin/develop)  
> git config core.sparsecheckout # timeout=10  
> git checkout -f 4e0e78bb64ee4bf68f2f08d84daf09eafe64992c # timeout=10  
Commit message: "Update Dockerfile"  
> git rev-list --no-walk 4e0e78bb64ee4bf68f2f08d84daf09eafe64992c # timeout=10  
[Job1] $ /bin/sh -xe /tmp/jenkins7412140733262760279.sh  
+ sudo docker rm -f c1  
c1  
+ sudo docker build /home/ubuntu/jenkins/workspace/Job1/ -t j1  
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 294.9kB  
  
Step 1/5 : FROM ubuntu  
--> ca2b0f26964c  
Step 2/5 : RUN apt update  
--> Using cache  
--> 8d69e5308758  
Step 3/5 : RUN apt install apache2 -y  
--> Using cache
```

```
Step 1/2 : FROM ubuntu  
--> ca2b0f26964c  
Step 2/5 : RUN apt update  
--> Using cache  
--> 8d69e5308758  
Step 3/5 : RUN apt install apache2 -y  
--> Using cache  
--> 804e913e76e3  
Step 4/5 : ENTRYPOINT apachectl -D FOREGROUND  
--> Using cache  
--> 034f7a79e9ac  
Step 5/5 : ADD . /var/www/html/  
--> Using cache  
--> ff137d2e36b4  
Successfully built ff137d2e36b4  
Successfully tagged j1:latest  
+ sudo docker run -itd -p 83:80 --name=c1 j1  
c410163d1be75dc9c502a0c44f015dc715d5e32c8792be1e69317494870867b  
Finished: SUCCESS
```

Instances | EC2 | us-east- | EC2 Instance Connect | EC2 Instance Connect | EC2 Instance Connect | New Item [Jenkins] | +

Jenkins

Dashboard > All >

Enter an item name

Job2

» Required field

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
OK for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

34°C Mostly clear

Search

08:24 PM 16-03-2024

Again create a new job – enter an item name – Job2 – click on freestyle project.

Instances | EC2 | us-east- | EC2 Instance Connect | EC2 Instance Connect | EC2 Instance Connect | Job2 Config [Jenkins] | +

Dashboard > Job2 > Configuration

Plain text Preview

Configure

Discard old builds ?

General

GitHub project

Source Code Management

This project is parameterized ?

Build Triggers

Throttle builds ?

Build Environment

Build Steps

Post-build Actions

Restrict where this project can be run ?

Label Expression ?

slave1

Label slave1 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Advanced ▾

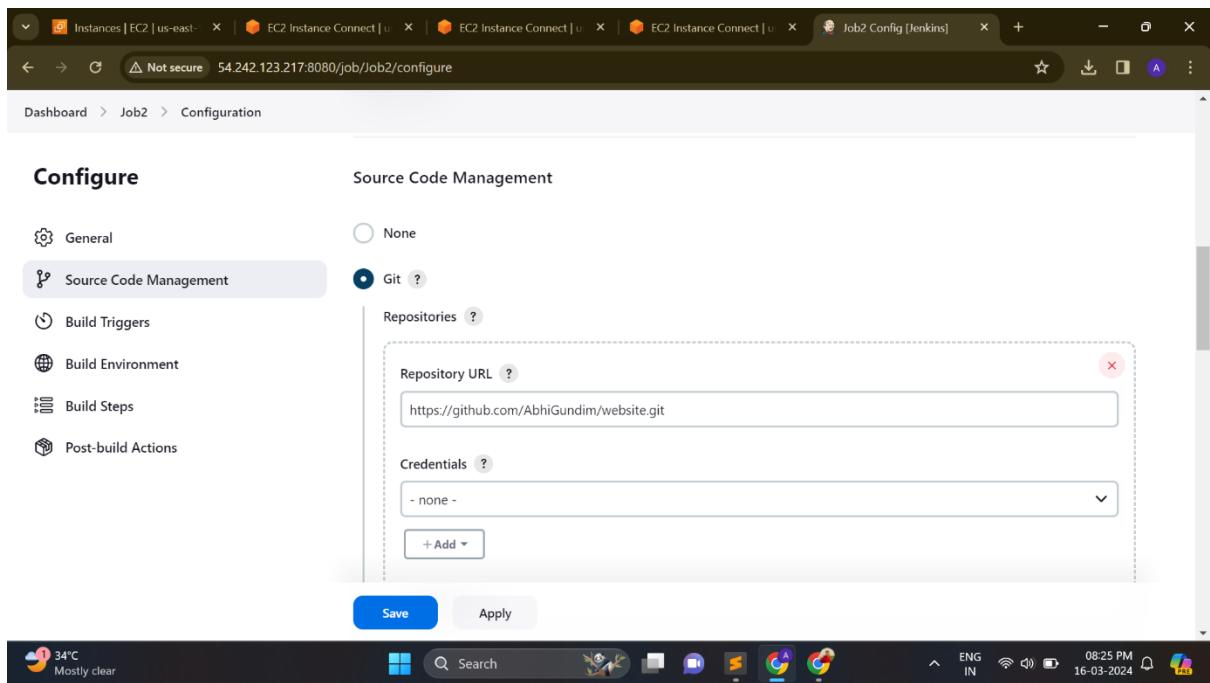
Save Apply

34°C Mostly clear

Search

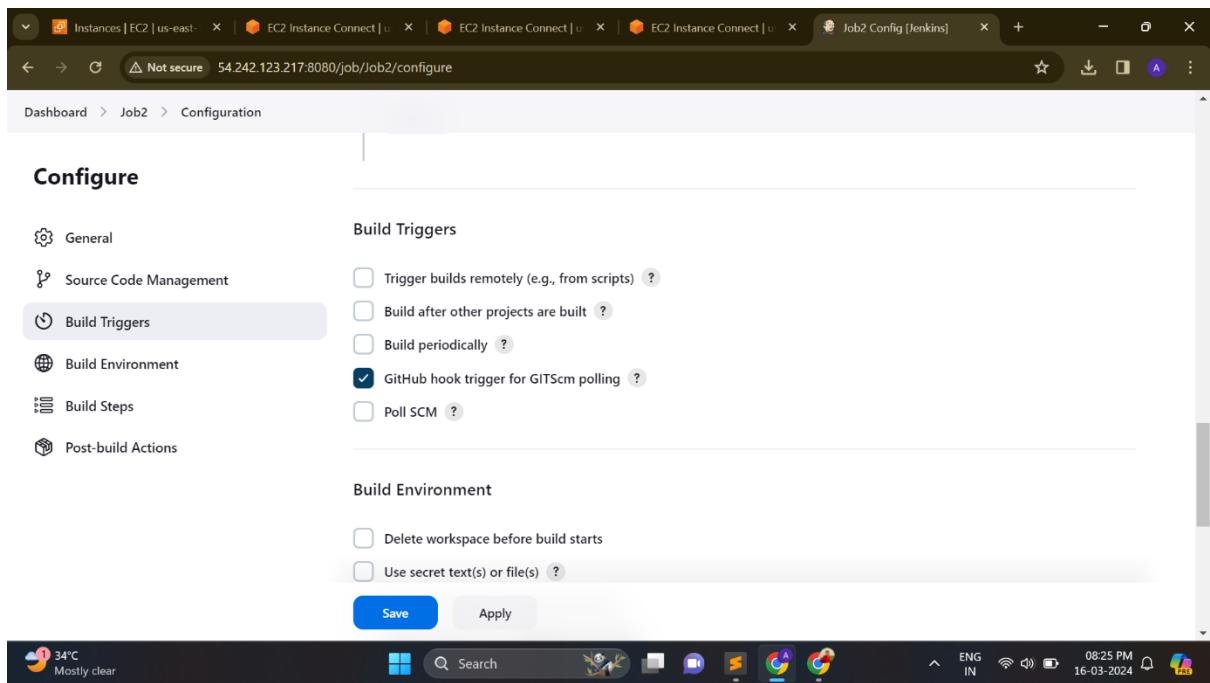
08:24 PM 16-03-2024

Click on restrict where this project can be run.



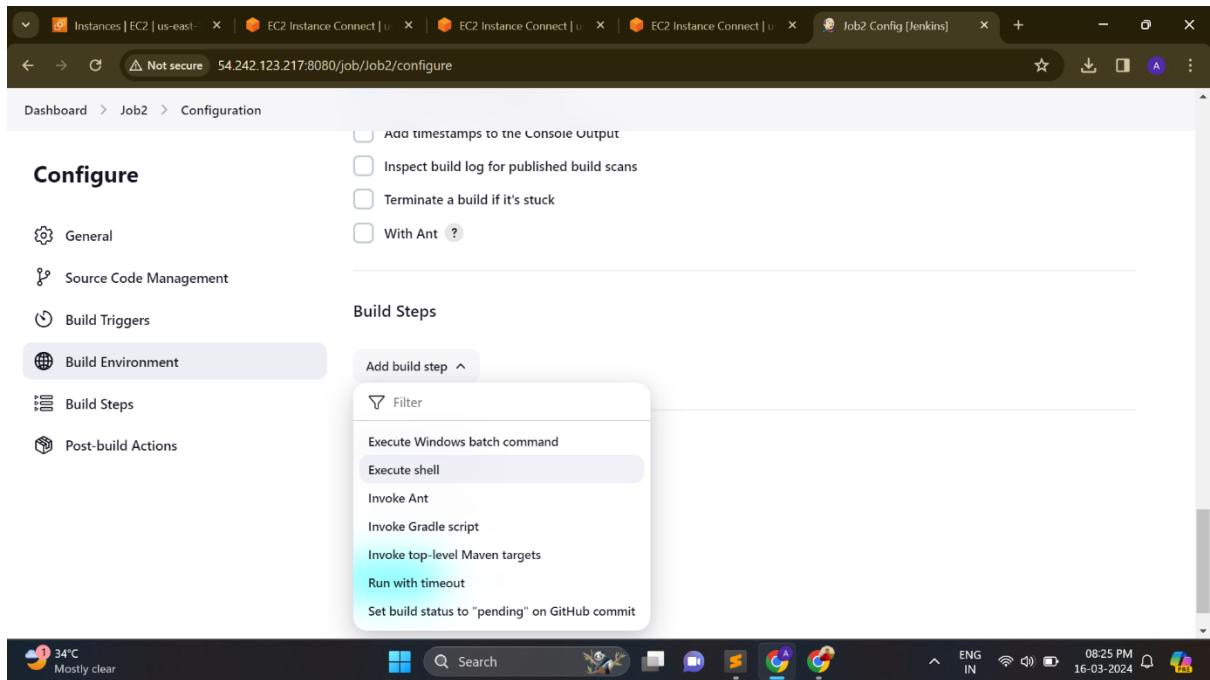
The screenshot shows the Jenkins configuration interface for a job named 'Job2'. The left sidebar lists 'General', 'Source Code Management' (which is selected), 'Build Triggers', 'Build Environment', 'Build Steps', and 'Post-build Actions'. The main panel is titled 'Source Code Management' and shows the 'Git' tab selected. It contains a 'Repositories' section with a 'Repository URL' input field containing 'https://github.com/AbhiGundim/website.git' and a 'Credentials' dropdown set to '- none -'. At the bottom are 'Save' and 'Apply' buttons.

Source code management - select Git – provide repository URL – click on save.

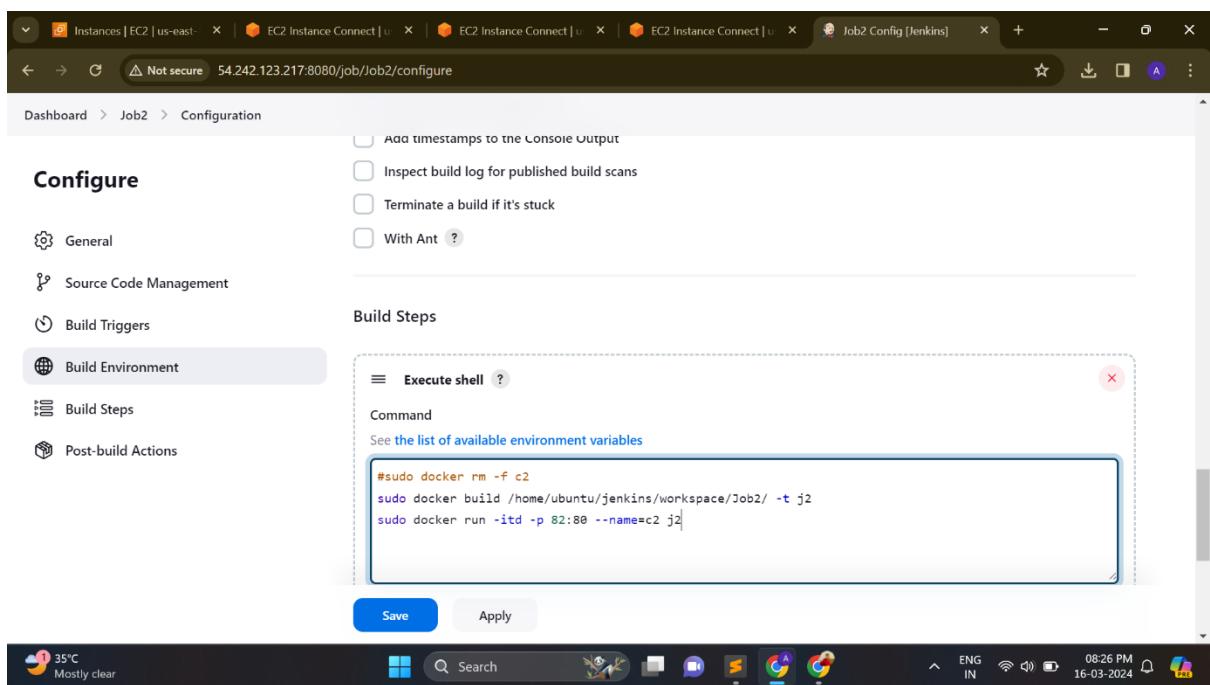


The screenshot shows the Jenkins configuration interface for a job named 'Job2'. The left sidebar lists 'General', 'Source Code Management', 'Build Triggers' (which is selected), 'Build Environment', 'Build Steps', and 'Post-build Actions'. The main panel is titled 'Build Triggers' and shows several options: 'Trigger builds remotely (e.g., from scripts)', 'Build after other projects are built', 'Build periodically', 'GitHub hook trigger for GITScm polling' (which is checked), and 'Poll SCM'. Below this is a 'Build Environment' section with 'Delete workspace before build starts' and 'Use secret text(s) or file(s)'. At the bottom are 'Save' and 'Apply' buttons.

Click on github hook trigger for GITScm polling.



Click on execute shell.



Build steps – command.

Dashboard > Job2 > workspace

Disable Project

Build Now

Configure

Delete Project

GitHub Hook Log

Rename

Build History

trend

#1 Mar 16, 2024, 2:57 PM

Atom feed for all Atom feed for failures

REST API Jenkins 2.449

Click on build now.

Dashboard > Job2 > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Execute shell

Command

See the list of available environment variables

```
sudo docker rm -f c2
sudo docker build /home/ubuntu/jenkins/workspace/Job2/ -t j2
sudo docker run -itd -p 82:80 --name=c2 j2
```

Advanced

Add build step

Save Apply

Again modify some commands and click on save.

Dashboard > Job2 >

Permalinks

- Configure
- Delete Project
- GitHub Hook Log
- Rename

Build History **trend** **Filter...**

- #2 | Mar 16, 2024, 3:00 PM
- #1 | Mar 16, 2024, 2:57 PM

Atom feed for all Atom feed for failures

REST API Jenkins 2.449

Again click on build now.

Dashboard > All >

Enter an item name

Job3 » Required field

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
OK for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific bundles, etc.

35°C Mostly clear Search ENG IN 08:31 PM 16-03-2024

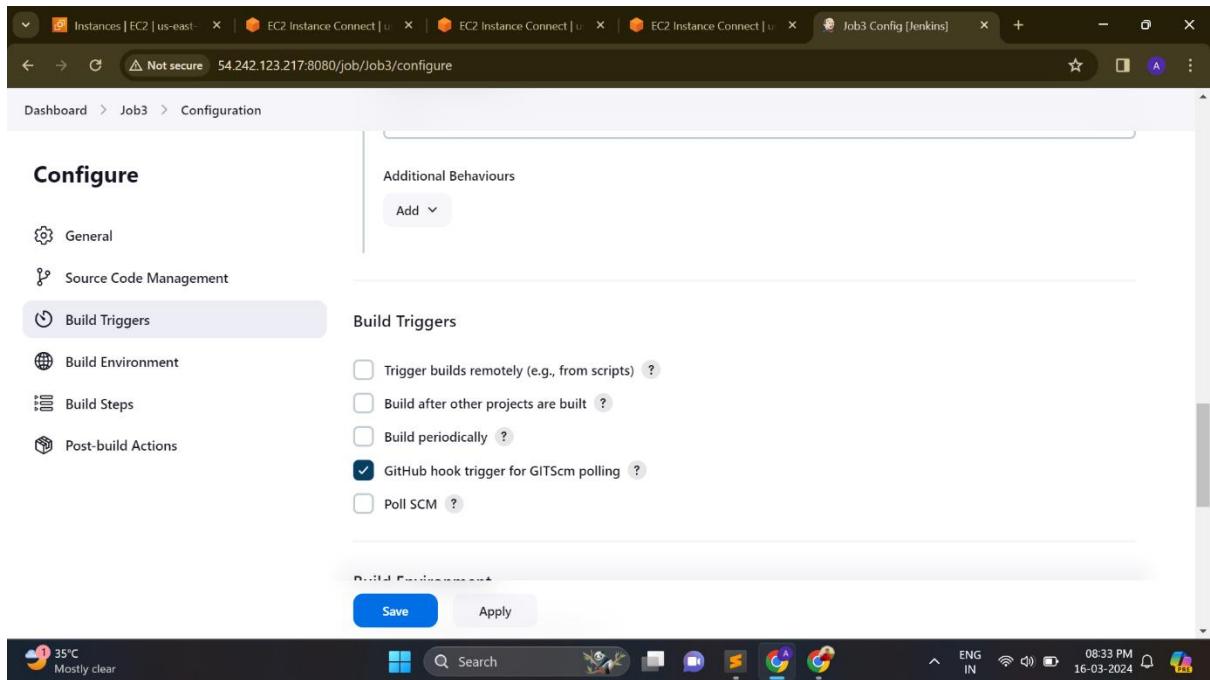
Create a new job – click on create a new item – enter an item name – Job3 – click on freestyle project.

The screenshot shows the Jenkins configuration page for Job3. In the 'General' section, the 'Restrict where this project can be run' checkbox is checked, and the 'Label Expression' field contains 'slave2'. Below this, there is a 'Source Code Management' section with 'Git' selected. The 'Repository URL' field contains 'https://github.com/AbhiGundim/website.git'. At the bottom, there are 'Save' and 'Apply' buttons.

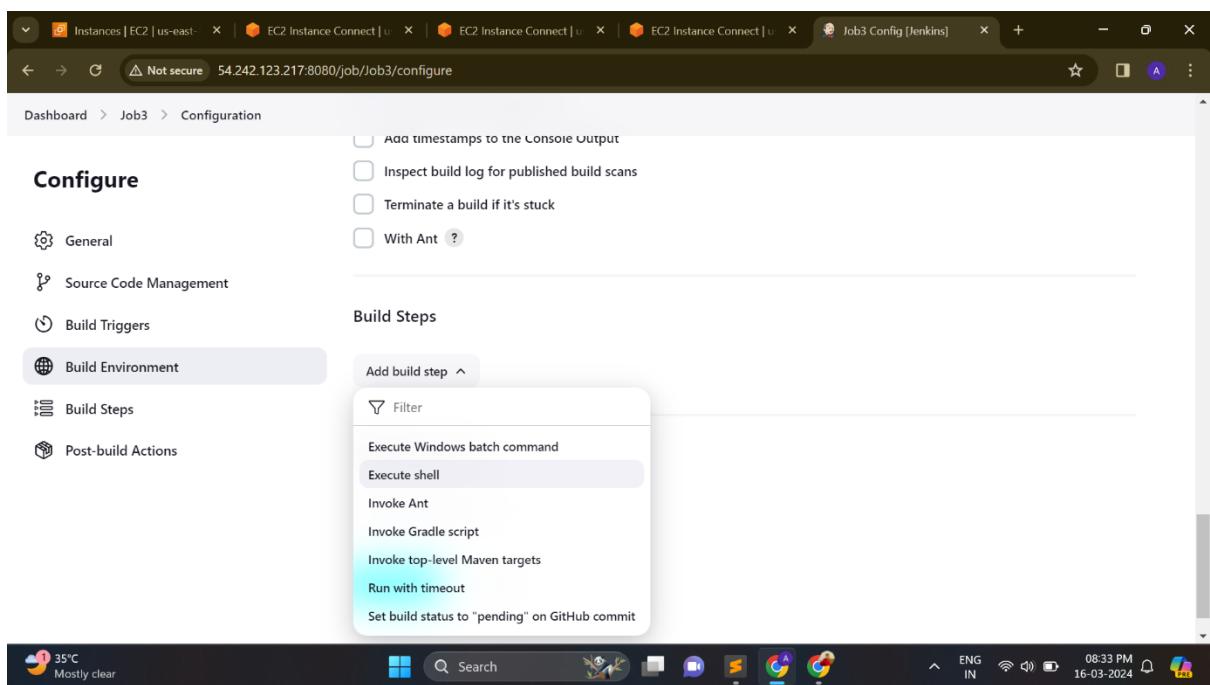
Click on restrict where this project can be run.

The screenshot shows the Jenkins configuration page for Job3. In the 'Source Code Management' section, 'Git' is selected. The 'Repository URL' field contains 'https://github.com/AbhiGundim/website.git'. At the bottom, there are 'Save' and 'Apply' buttons.

Source code management – Click on Git – provide Repository URL.



Click on github hook trigger for gitscm polling.



In build steps – click on execute shell.

Dashboard > Job3 > workspace

Disable Project

Build Now

Configure

Delete Project

GitHub Hook Log

Rename

Build History

trend

#1 Mar 16, 2024, 3:05 PM

Atom feed for all Atom feed for failures

REST API Jenkins 2.449

Click on build now.

Dashboard > Job3 > Configuration

With Ant ?

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Build Steps

Execute shell

Command

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

```
sudo docker rm -f c3
sudo docker build /home/ubuntu/jenkins/workspace/Job3/ -t j3
sudo docker run -itd -p 80:80 --name=c3 j3
```

Advanced ▾

Save Apply

35°C Mostly clear

Click on save.

Jenkins

Dashboard > Job2 >

Status: Job2

Changes, Workspace, Build Now, Configure, Delete Project, GitHub Hook Log, Rename.

Add description, Disable Project.

Permalinks:

- Last build (#2), 13 min ago
- Last stable build (#2), 13 min ago
- Last successful build (#2), 13 min ago
- Last completed build (#2), 13 min ago

Build History: trend ▾ Filter... /

35°C Air: Moderate

Search, Taskbar icons, ENG IN, 08:44 PM, 16-03-2024, Notifications.

Copy the ip address and webpage runned successfully.

Hello There!

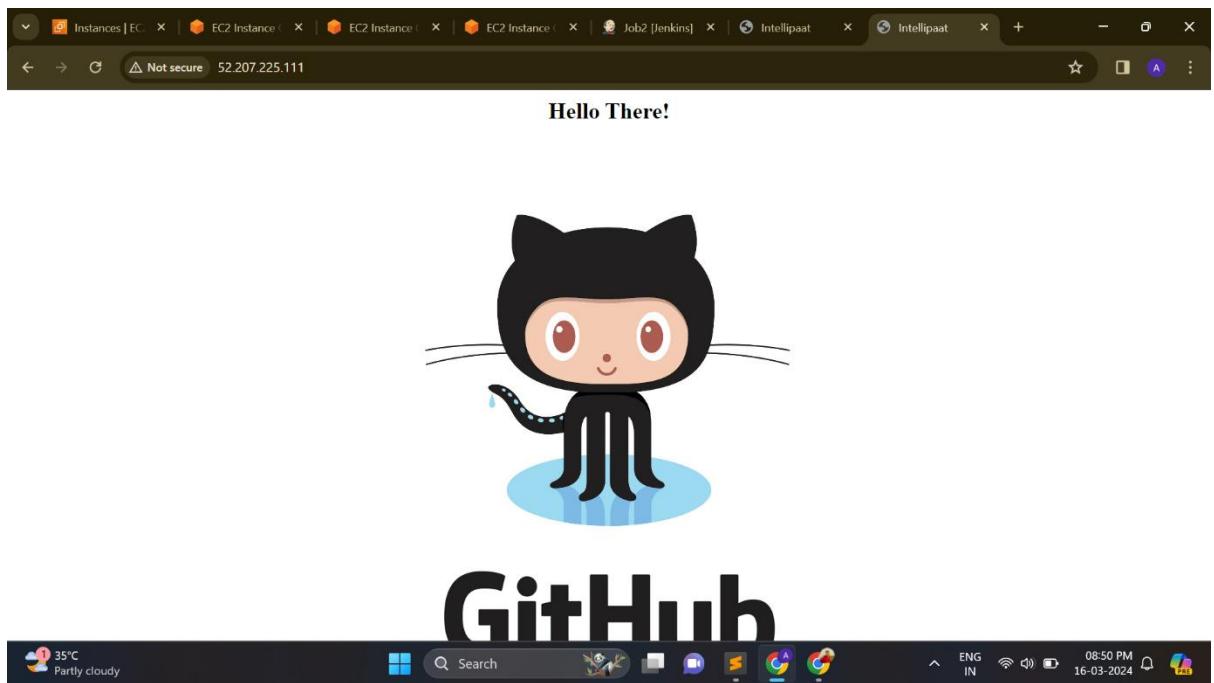
Hello There Once Again myself Abhinandan!



GitHub

35°C Partly cloudy

Search, Taskbar icons, ENG IN, 08:49 PM, 16-03-2024, Notifications.



Two webpages runned successfully.