

Course: Online Session DevOps Weekday BC = 231107 B2

Project 2

NAME: SONU NIGAM U S

Mail id: sonunigamn55@gmail.com

1 we create our 1st instance manually

The screenshot shows the AWS Management Console interface for the EC2 service. The left sidebar is collapsed, and the main content area displays the 'Instances' section. A single instance, 'machine1', is listed in the table. The instance details are shown in a modal window below the table.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
machine1	i-07343977e00c80bf7	Running	t2.medium	-	No alarms	us-west-2c

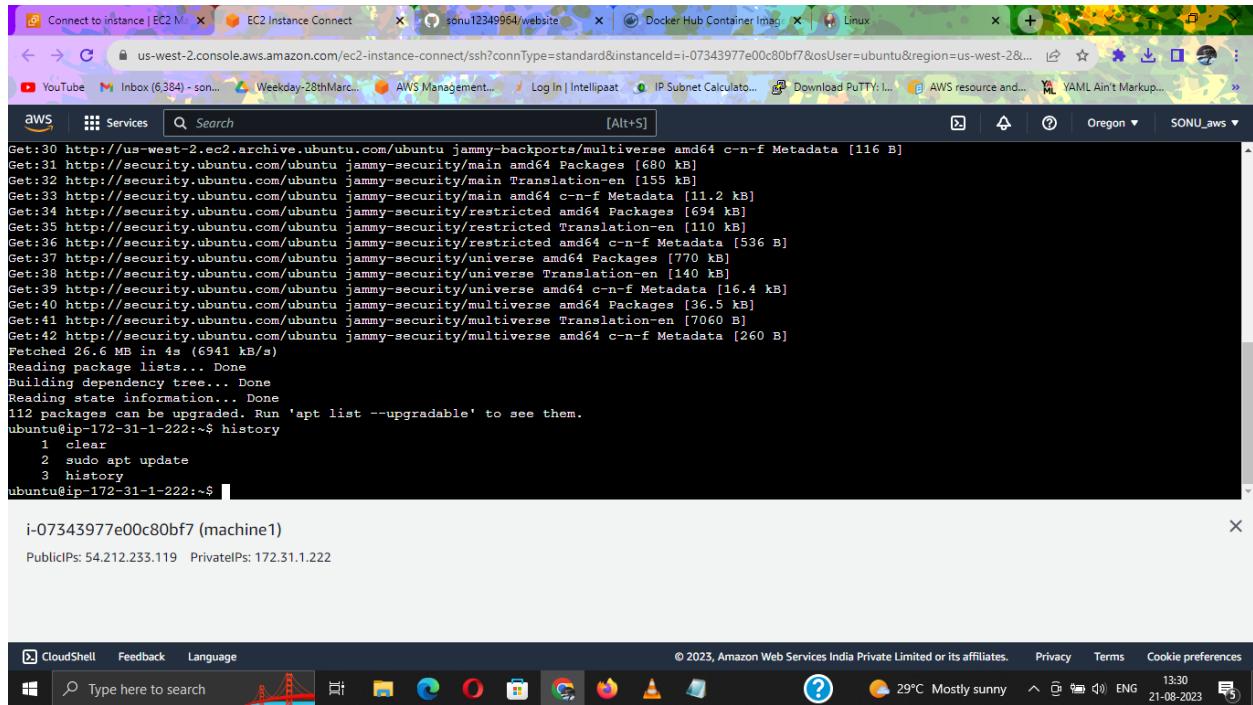
Instance: i-07343977e00c80bf7 (machine1)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary

Instance ID	Public IPv4 address	Private IPv4 addresses
i-07343977e00c80bf7 (machine1)	54.212.233.119 open address	172.31.1.222
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-54-212-233-119.us-west-2.compute.amazonaws.com open address

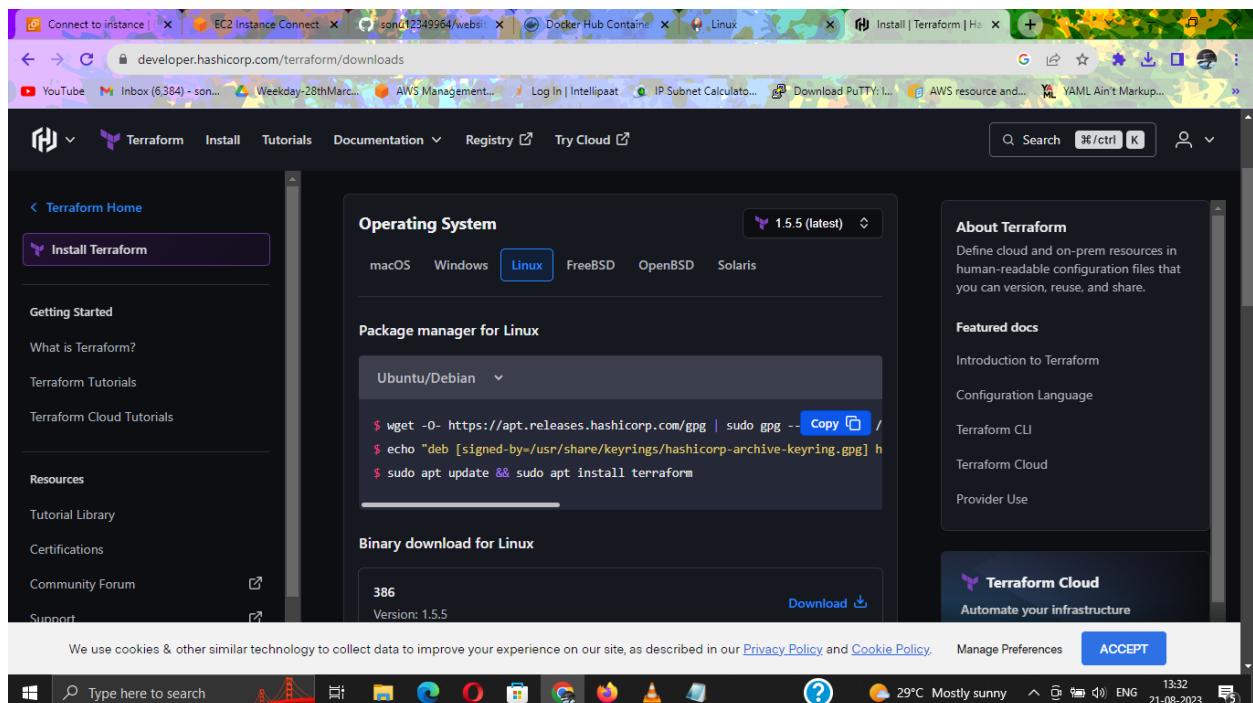
2.we update



```
Get:30 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:31 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [680 kB]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [155 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [11.2 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [694 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [110 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [536 B]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [770 kB]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [140 kB]
Get:39 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.4 kB]
Get:40 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [36.5 kB]
Get:41 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7060 B]
Get:42 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
Fetched 26.6 MB in 4s (6941 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
112 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-1-222:~$ history
1  clear
2  sudo apt update
3  history
ubuntu@ip-172-31-1-222:~$
```

i-07343977e00c80bf7 (machine1)
Public IPs: 54.212.233.119 Private IPs: 172.31.1.222

3.we install terraform



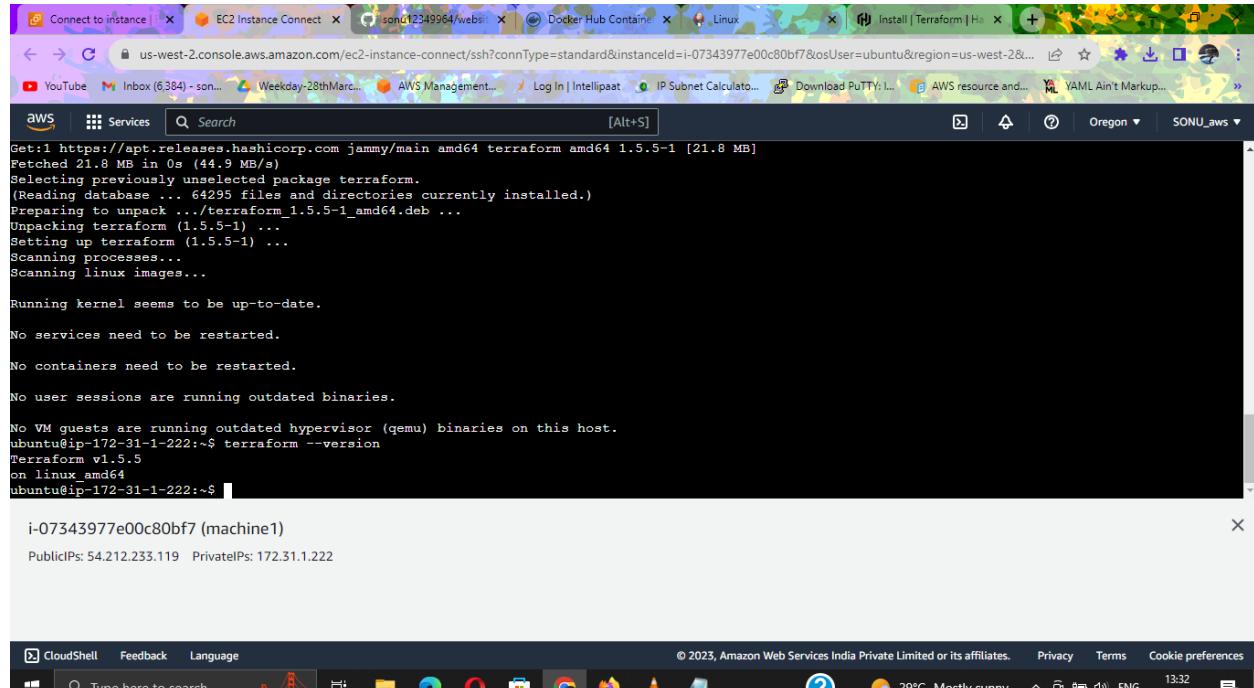
The screenshot shows a Windows desktop environment with a browser window open to the HashiCorp Terraform website. The URL in the address bar is developer.hashicorp.com/terraform/downloads. The page content is as follows:

- Operating System**: A dropdown menu set to Linux, with options for macOS, Windows, FreeBSD, OpenBSD, and Solaris.
- Package manager for Linux**: A section for Ubuntu/Debian with a copy-and-paste command:

```
$ wget -O https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor - | sudo apt-key add -
$ echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] h
$ sudo apt update && sudo apt install terraform
```
- Binary download for Linux**: A section for 386 architecture, Version 1.5.5, with a download button.
- About Terraform**: A brief description: "Define cloud and on-prem resources in human-readable configuration files that you can version, reuse, and share."
- Featured docs**: Links to Introduction to Terraform, Configuration Language, Terraform CLI, Terraform Cloud, and Provider Use.
- Terraform Cloud**: A sidebar with the text "Automate your infrastructure".

At the bottom of the browser window, there is a cookie consent banner from the website.

4.



```
Get:1 https://apt.releases.hashicorp.com jammy/main amd64 terraform amd64 1.5.5-1 [21.8 MB]
Fetched 21.8 MB in 0s (44.9 MB/s)
Selecting previously unselected package terraform.
(Reading database ... 64295 files and directories currently installed.)
Preparing to unpack .../terraform_1.5.5-1_amd64.deb ...
Unpacking terraform (1.5.5-1) ...
Setting up terraform (1.5.5-1) ...
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.

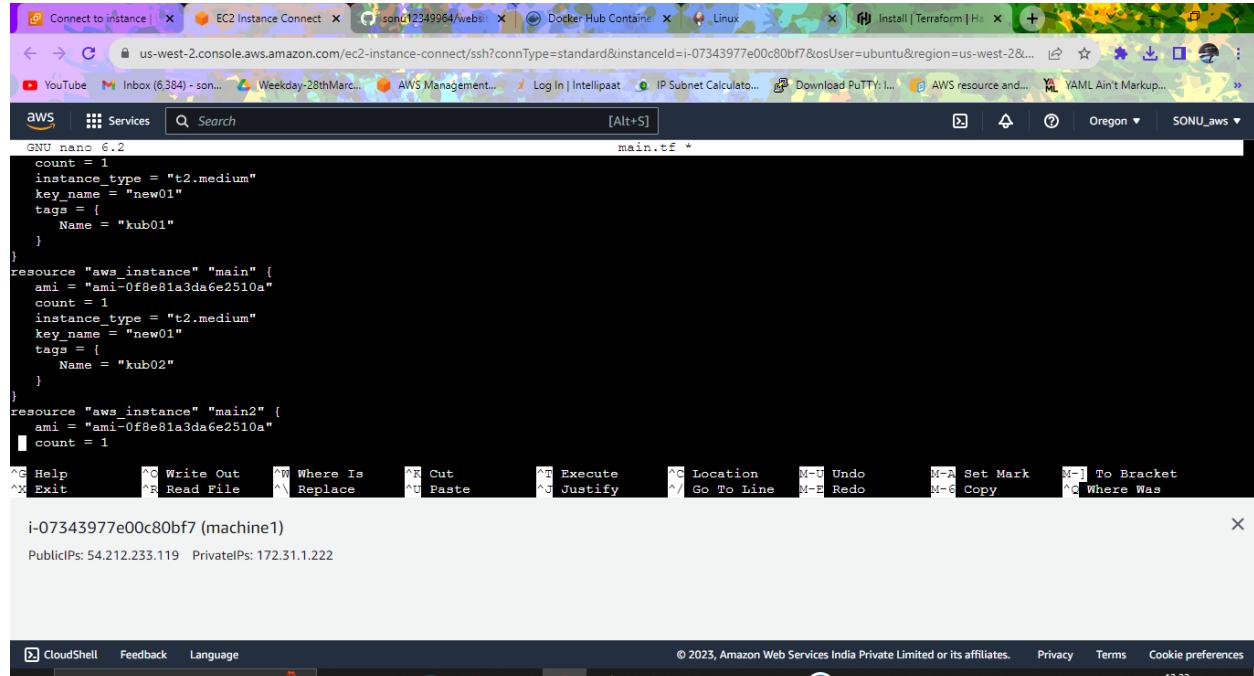
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-1-222:~$ terraform --version
Terraform v1.5.5
on linux_amd64
ubuntu@ip-172-31-1-222:~$
```

i-07343977e00c80bf7 (machine1)

PublicIPs: 54.212.233.119 PrivateIPs: 172.31.1.222

5. we create 3 machine using the code



```
GNU nano 6.2
```

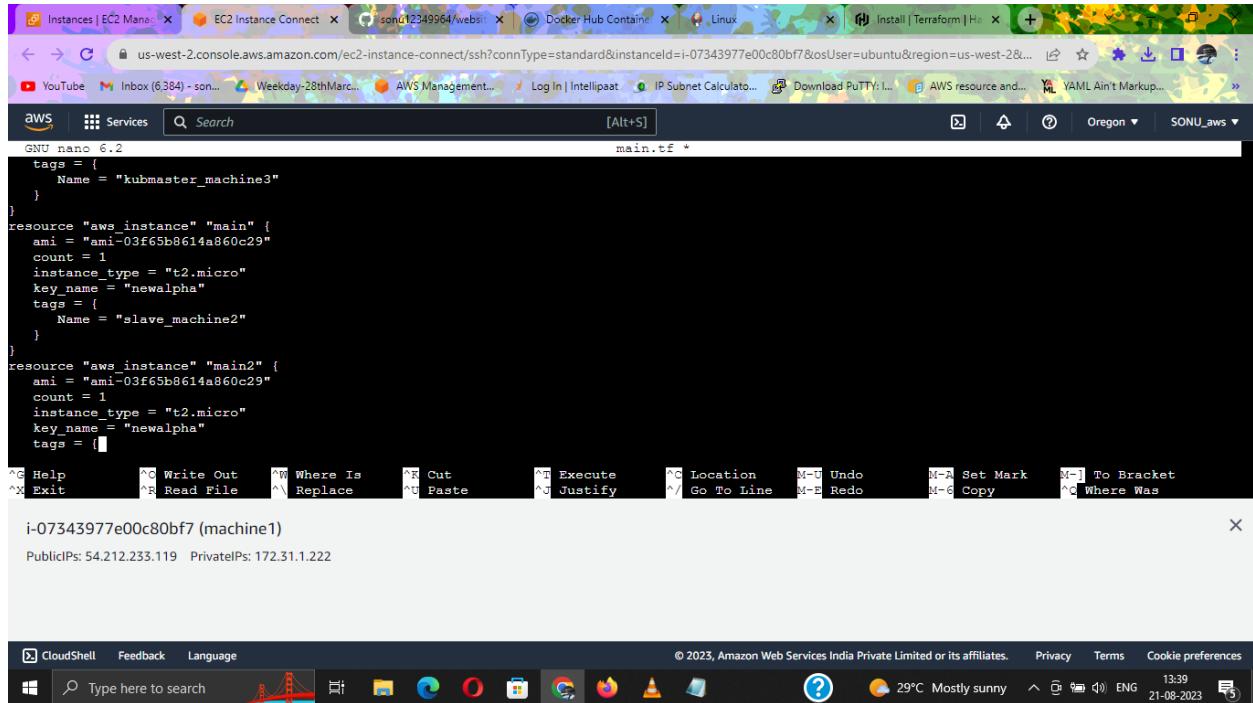
```
main.tf *
```

```
count = 1
resource "aws_instance" "main" {
  ami = "ami-0f8e81a3da6e2510a"
  count = 1
  instance_type = "t2.medium"
  key_name = "new01"
  tags = [
    Name = "kub01"
  ]
}
resource "aws_instance" "main2" {
  ami = "ami-0f8e81a3da6e2510a"
  count = 1
  instance_type = "t2.medium"
  key_name = "new01"
  tags = [
    Name = "kub02"
  ]
}
```

i-07343977e00c80bf7 (machine1)

PublicIPs: 54.212.233.119 PrivateIPs: 172.31.1.222

6.we now create action group



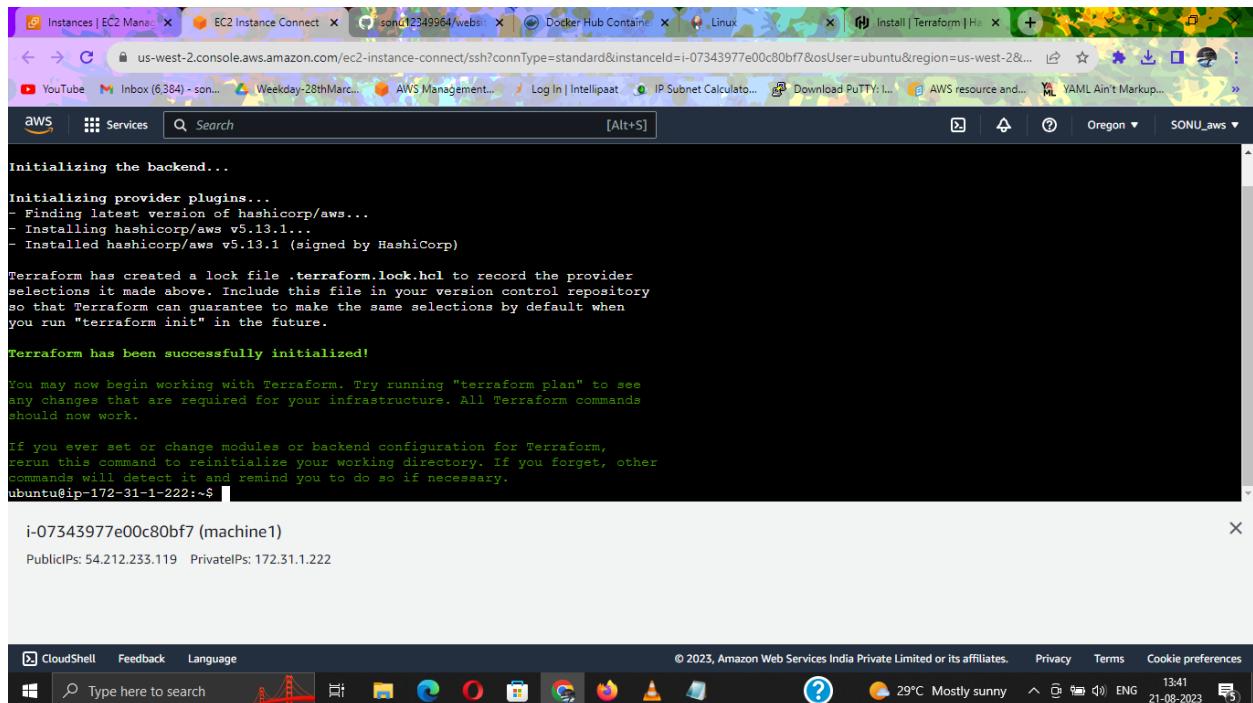
```
GNU nano 6.2
tags = {
    Name = "kubmaster_machine3"
}
resource "aws_instance" "main" {
    ami = "ami-03f65b8614a860c29"
    count = 1
    instance_type = "t2.micro"
    key_name = "newalpha"
    tags = {
        Name = "slave_machine2"
    }
}
resource "aws_instance" "main2" {
    ami = "ami-03f65b8614a860c29"
    count = 1
    instance_type = "t2.micro"
    key_name = "newalpha"
    tags = {}
```

i-07343977e00c80bf7 (machine1)

PublicIPs: 54.212.233.119 PrivateIPs: 172.31.1.222

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 29°C Mostly sunny 13:39 21-08-2023

7.



```
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.13.1...
- Installed hashicorp/aws v5.13.1 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
run this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
ubuntu@ip-172-31-1-222:~$
```

i-07343977e00c80bf7 (machine1)

PublicIPs: 54.212.233.119 PrivateIPs: 172.31.1.222

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 29°C Mostly sunny 13:41 21-08-2023

8.

```
i-07343977e00c80bf7 (machine1)
PublicIPs: 54.212.233.119 PrivateIPs: 172.31.1.222
```

9.

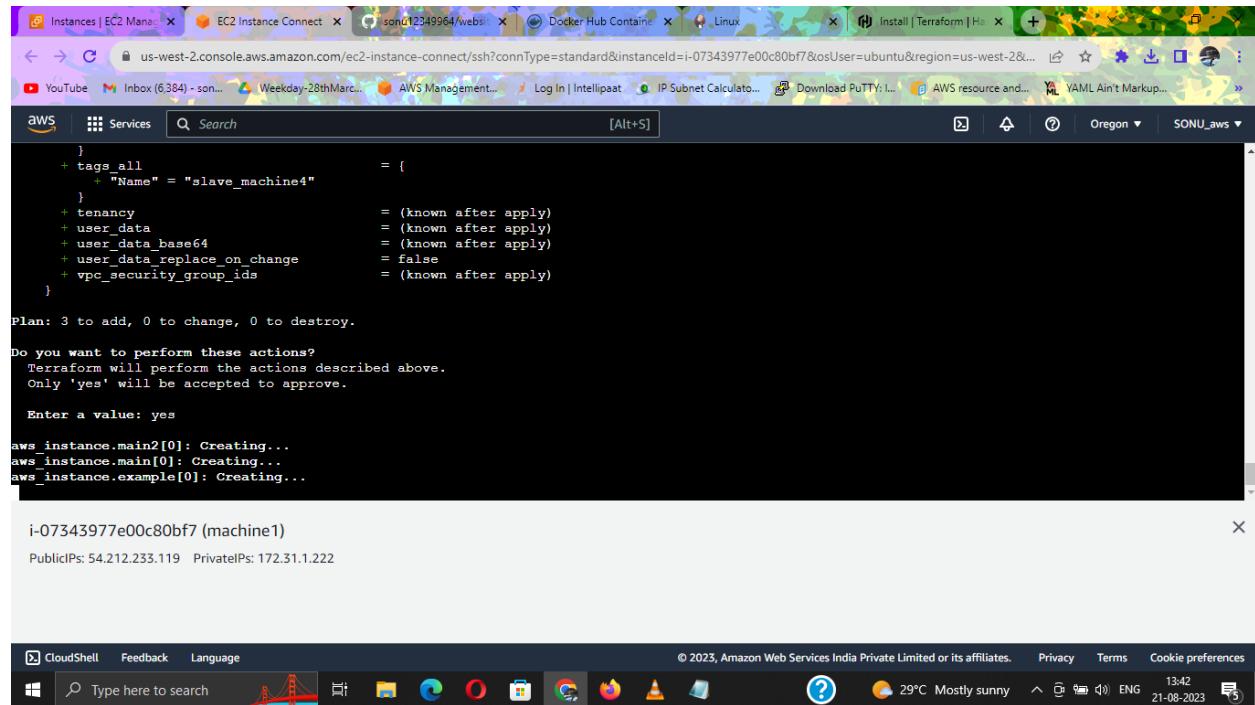
```
+ security_groups          = (known after apply)
+ source_dest_check        = true
+ spot_instance_request_id = (known after apply)
+ subnet_id                 = (known after apply)
+ tags                      =
  + "Name" = "slave_machine4"
}
+ tags_all                  = {
  + "Name" = "slave_machine4"
}
+ tenancy                   = (known after apply)
+ user_data                 = (known after apply)
+ user_data_base64          = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids    = (known after apply)
}

Plan: 3 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
ubuntu@ip-172-31-1-222:~$
```

```
i-07343977e00c80bf7 (machine1)
PublicIPs: 54.212.233.119 PrivateIPs: 172.31.1.222
```

10. we run our code



```
Instances | EC2 Manager X EC2 Instance Connect X sonu12349964/webs X Docker Hub Container X Linux X Install | Terraform | Ha X
us-west-2.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-07343977e00c80bf7&lostUser=ubuntu&region=us-west-2&... YouTube Inbox (6,384) - son... Weekday-28thMarch... AWS Management... Log In | Intellipaat IP Subnet Calculato... Download PuTTY: I... AWS resource and... YAML Ain't Markup...
AWS Services Search [Alt+S] X ASONU_aws ▾
}
+ tags_all = {
  + "Name" = "slave_machine4"
}
+ tenancy = (known after apply)
+ user_data = (known after apply)
+ user_data_base64 = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
}

Plan: 3 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

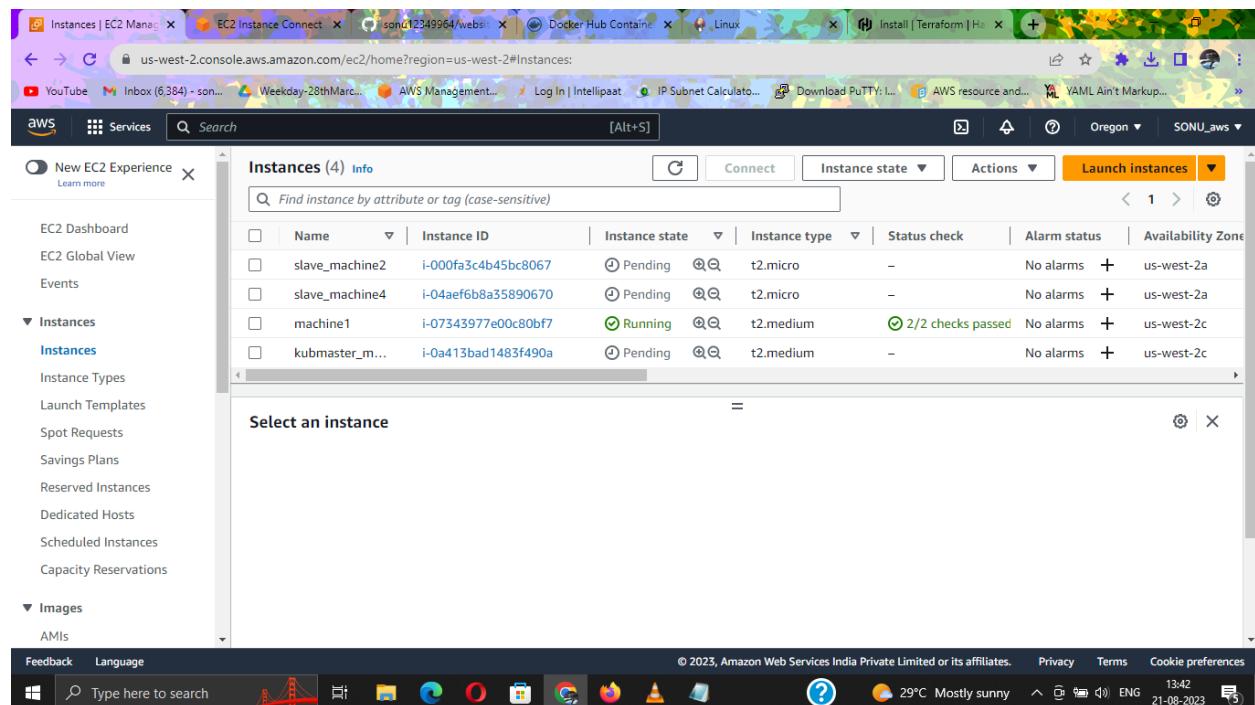
Enter a value: yes

aws_instance.main[0]: Creating...
aws_instance.main[0]: Creating...
aws_instance.example[0]: Creating...

i-07343977e00c80bf7 (machine1)
PublicIPs: 54.212.233.119 PrivateIPs: 172.31.1.222

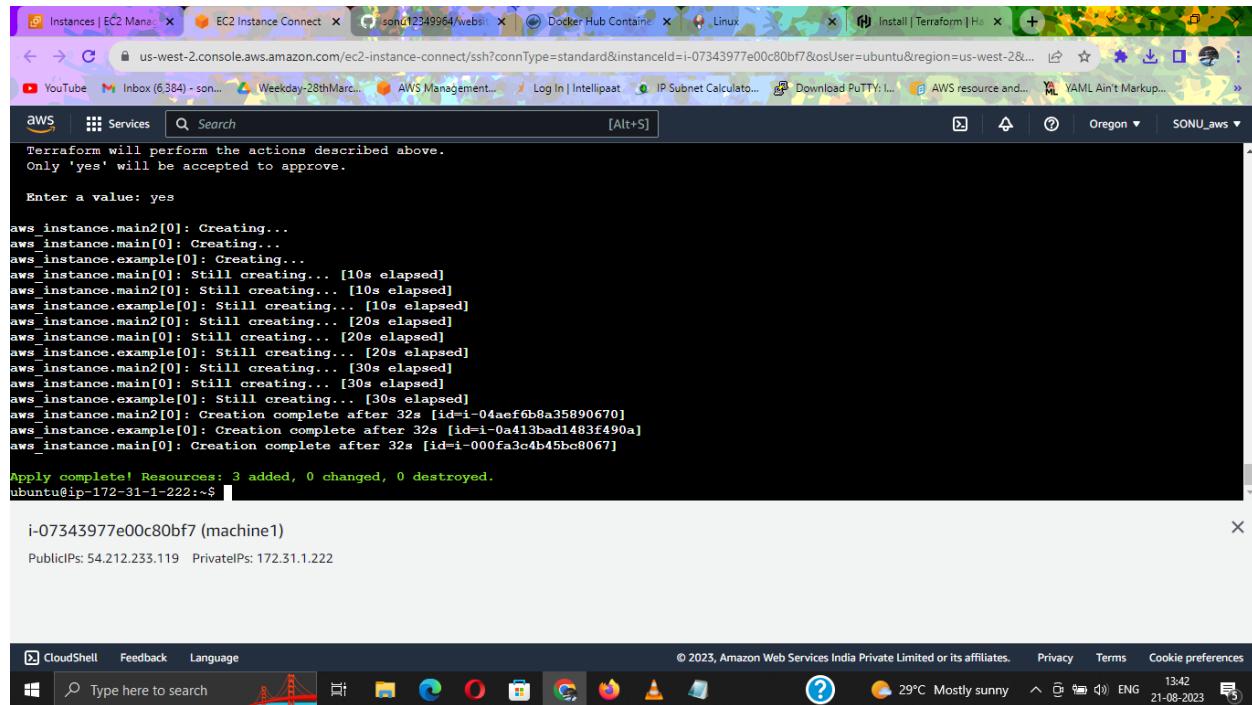
CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences
Type here to search 29°C Mostly sunny 13:42 ENG 21-08-2023
```

11. our machine are installed



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
slave_machine2	i-000fa3c4b45bc8067	Pending	t2.micro	-	No alarms	+ us-west-2a
slave_machine4	i-04aef6b8a35890670	Pending	t2.micro	-	No alarms	+ us-west-2a
machine1	i-07343977e00c80bf7	Running	t2.medium	2/2 checks passed	No alarms	+ us-west-2c
kubmaster_m...	i-0a413bad1483f490a	Pending	t2.medium	-	No alarms	+ us-west-2c

12. we got our email



```
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws instance.main2[0]: Creating...
aws instance.main[0]: Creating...
aws instance.example[0]: Creating...
aws instance.main[0]: Still creating... [10s elapsed]
aws instance.main2[0]: Still creating... [10s elapsed]
aws instance.example[0]: Still creating... [10s elapsed]
aws instance.main2[0]: Still creating... [20s elapsed]
aws instance.main[0]: Still creating... [20s elapsed]
aws instance.example[0]: Still creating... [20s elapsed]
aws instance.main2[0]: Still creating... [30s elapsed]
aws instance.main[0]: Still creating... [30s elapsed]
aws instance.example[0]: Still creating... [30s elapsed]
aws instance.main2[0]: Creation complete after 32s [id=i-04acf6b8a35890670]
aws instance.example[0]: Creation complete after 32s [id=i-0x413bad1483f490a]
aws instance.main[0]: Creation complete after 32s [id=i-000fa3c4b45bc8067]

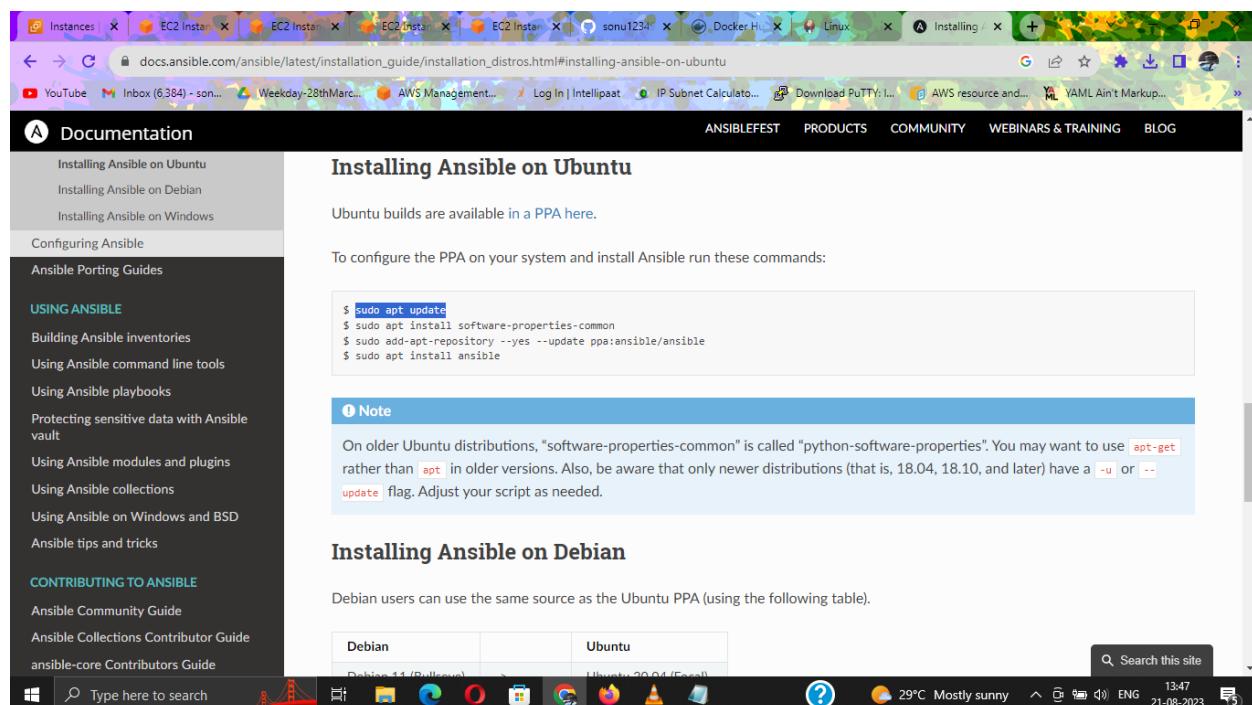
Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-1-222:~$
```

i-07343977e00c80bf7 (machine1)

PublicIPs: 54.212.233.119 PrivateIPs: 172.31.1.222



13 we install ansible in our 1rd machine



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.

Installing Ansible on Debian

Debian users can use the same source as the Ubuntu PPA (using the following table).

Debian	Ubuntu
Distro 11 (Bullseye)	Ubuntu 20.04 (Focal)



14

A screenshot of a Windows desktop environment. At the top, there is a taskbar with several icons and an open browser window showing the AWS CloudShell interface. The main focus is a terminal window titled 'Linux' which displays the following command and its output:

```
ubuntu@ip-172-31-1-222:~$ ansible --version
ansible [core 2.15.2]
  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 = 3.10.6 (main, Mar 10 2023, 10:55:28) [GCC 11.3.0] (/usr/bin/python3)
  jinja version = 3.0.3
  libyaml = True
```

Below the terminal, a message box provides details about the EC2 instance:

i-07343977e00c80bf7 (machine1)
PublicIPs: 54.212.233.119 PrivateIPs: 172.31.1.222

The system tray at the bottom shows various icons and the current date and time: 21-08-2023.

15

A screenshot of a Windows desktop environment, similar to the previous one. The taskbar and browser window are identical. The terminal window titled 'Linux' shows the execution of the `ssh-keygen` command:

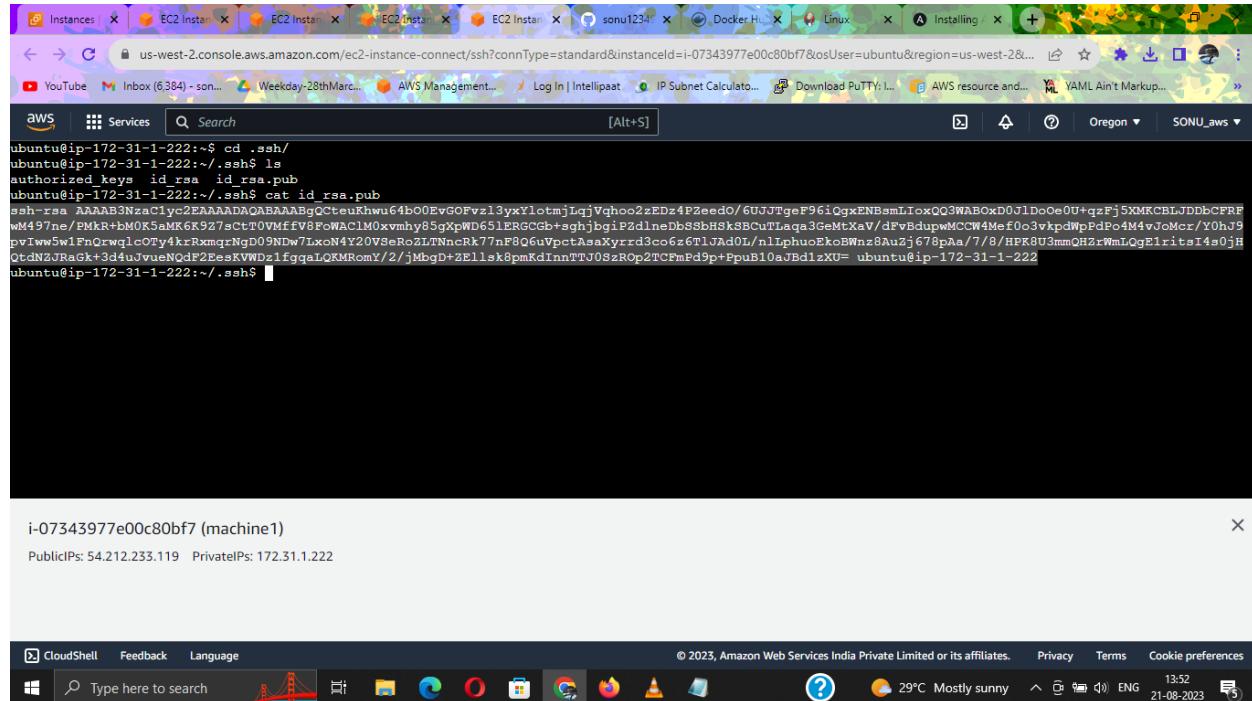
```
ubuntu@ip-172-31-1-222:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter same passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:9lu0ejaj8u5pp87qyTB0Si6b70ykIoBzyt2yrfQgHj8 ubuntu@ip-172-31-1-222
The key's randomart image is:
+---[RSA 3072]---+
| . |
| .o |
| +...+ o |
| =oo+.. S + |
| ++o.=. o o . |
| =o+Eto. . . |
| .++=to. o |
| +**o. . |
+---[SHA256]---+
```

Below the terminal, a message box provides details about the EC2 instance:

i-07343977e00c80bf7 (machine1)
PublicIPs: 54.212.233.119 PrivateIPs: 172.31.1.222

The system tray at the bottom shows various icons and the current date and time: 21-08-2023.

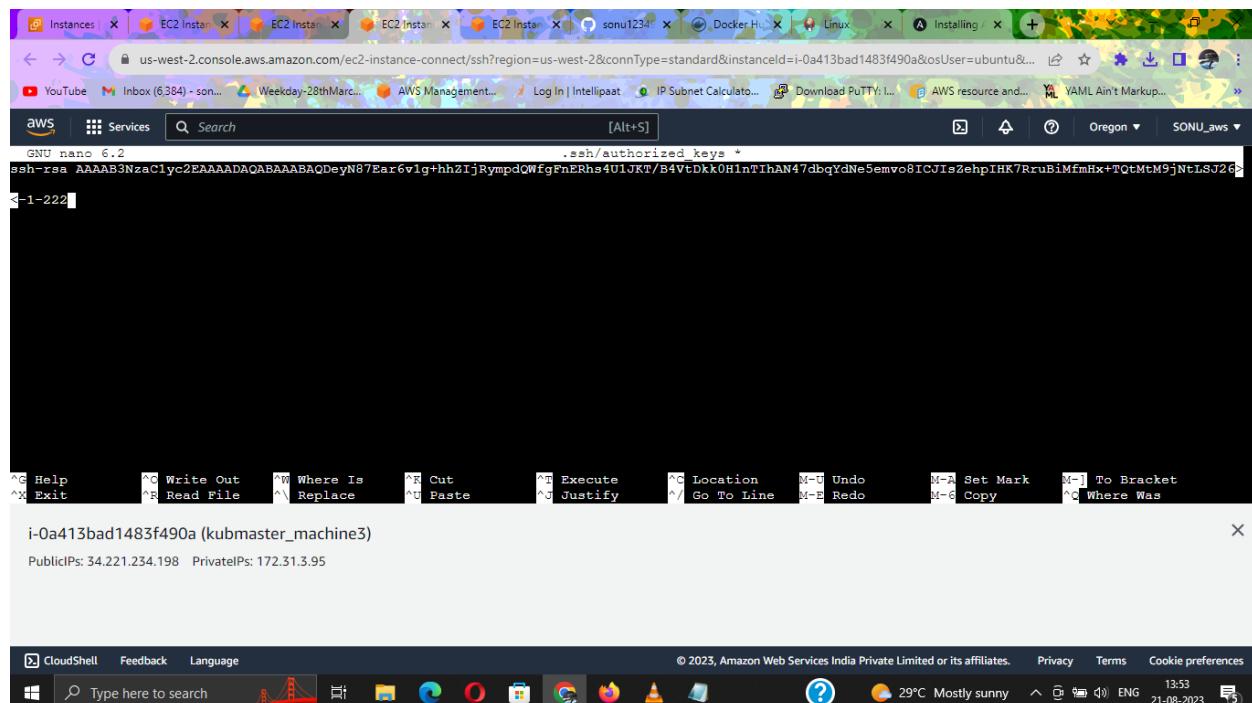
16



```
ubuntu@ip-172-31-1-222:~$ cd .ssh/
ubuntu@ip-172-31-1-222:~/ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
ubuntu@ip-172-31-1-222:~/ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDeyN87Ear6vlghhZijRympdOWfgFnERhs4U1JKt/84VtDkk0H1nTThAN47dbqYdNe5emvo8ICJIsZehpIHK7RruBiMfmHx+TQtMtM9jNtISJ26
ubuntu@ip-172-31-1-222:~/ssh$
```

i-07343977e00c80bf7 (machine1)
PublicIPs: 54.212.233.119 PrivateIPs: 172.31.1.222

17

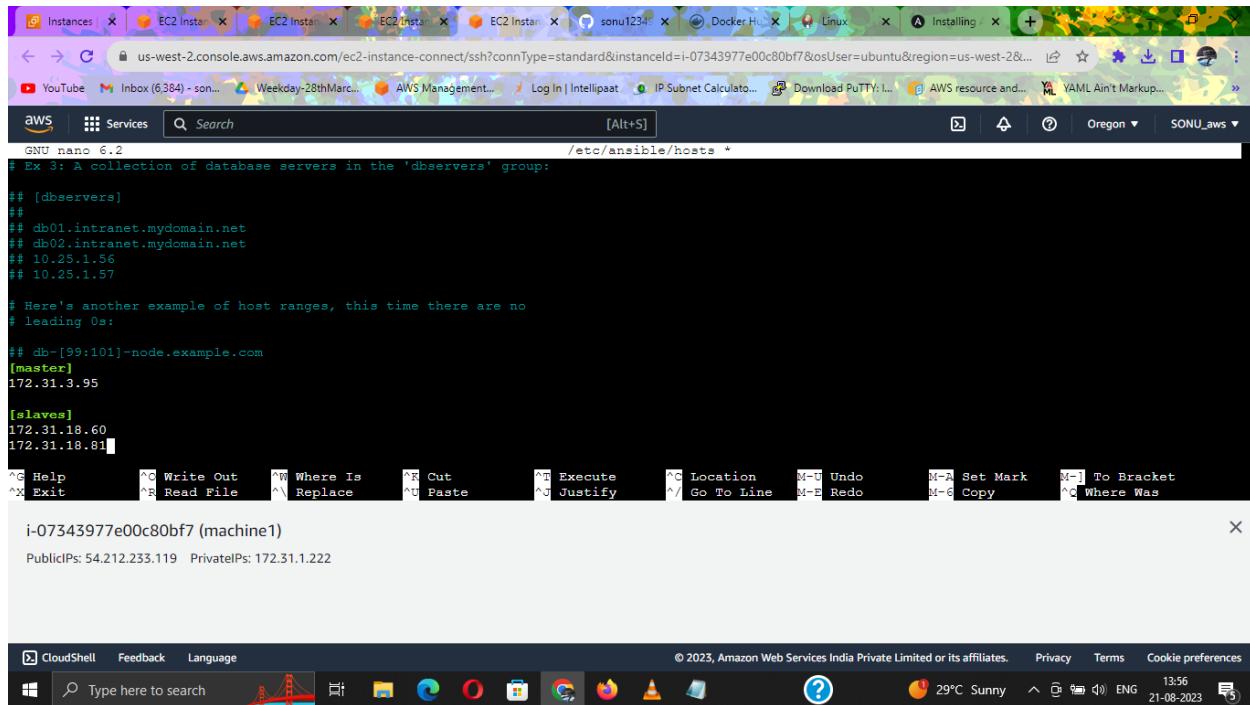


```
GNU nano 6.2 .ssh/authorized_keys +
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDeyN87Ear6vlghhZijRympdOWfgFnERhs4U1JKt/84VtDkk0H1nTThAN47dbqYdNe5emvo8ICJIsZehpIHK7RruBiMfmHx+TQtMtM9jNtISJ26
-1-222
```

i-0a413bad1483f490a (kubmaster_machine3)

PublicIPs: 34.221.234.198 PrivateIPs: 172.31.3.95

18 we create master and slaves



The screenshot shows a Windows desktop environment. In the foreground, a terminal window is open, displaying Ansible configuration code for database servers. The code defines groups for db servers, a master node, and slave nodes. It also includes host ranges and specific IP addresses. The terminal window has a standard Windows-style menu bar at the top.

```
GNU nano 6.2 /etc/ansible/hosts
# 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

# Here's another example of host ranges, this time there are no
# leading Os:

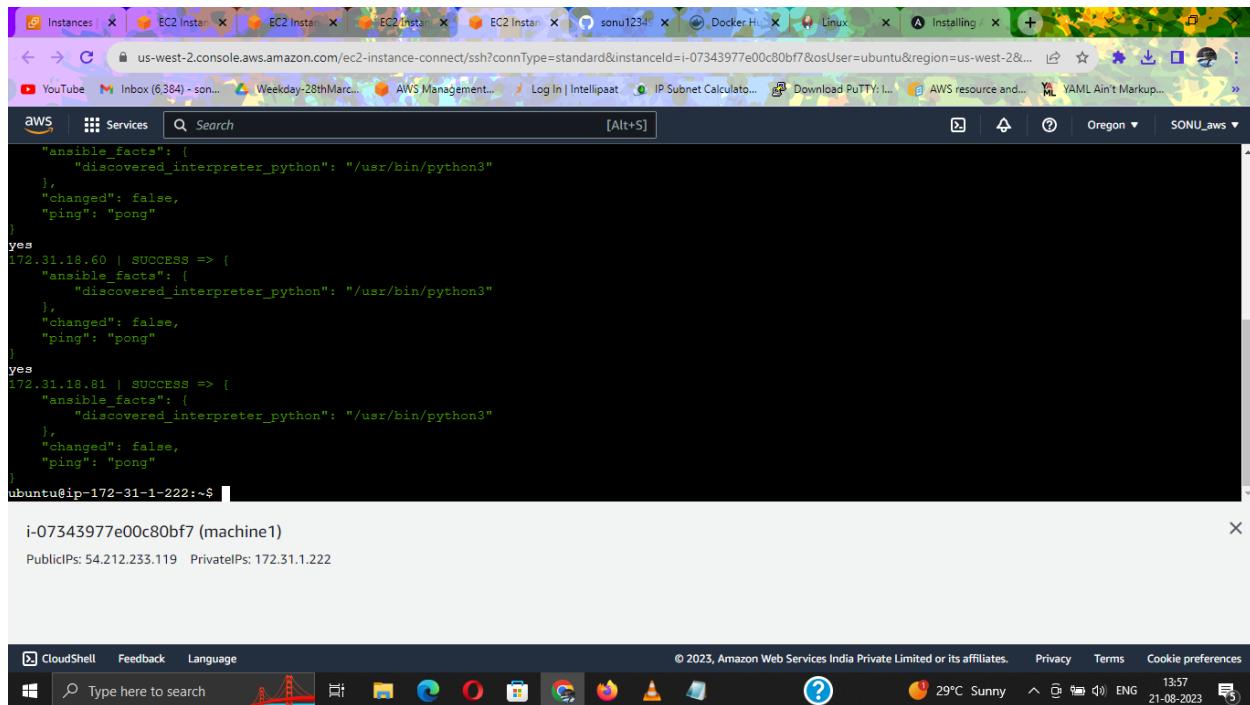
## db-[99:101]-node.example.com
[master]
172.31.3.95

[slaves]
172.31.18.60
172.31.18.81

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

i-07343977e00c80bf7 (machine1)
PublicIPs: 54.212.233.119 PrivateIPs: 172.31.1.222
```

19 we ping to our slaves



The screenshot shows a Windows desktop environment. In the foreground, a terminal window is open, displaying the output of an Ansible ping command. The command was run on the master node (172.31.18.60) to check connectivity to two slave nodes (172.31.18.81 and 172.31.1.222). The output shows successful ping responses from both slave nodes. The terminal window has a standard Windows-style menu bar at the top.

```
"ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
},
"changed": false,
"ping": "pong"
}
yes
172.31.18.60 | SUCCESS => {
"ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
},
"changed": false,
"ping": "pong"
}
yes
172.31.18.81 | SUCCESS => {
"ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
},
"changed": false,
"ping": "pong"
}
ubuntu@ip-172-31-1-222:~$
```

20 we now write scripts

A screenshot of a terminal window titled "script1.sh". The terminal is running on an AWS EC2 instance. The command entered is:

```
sudo apt update  
sudo apt install openjdk-11-jdk -y  
curl -fsSL https://pkg.jenkins.io/debian/jenkins.io-2023.key | sudo tee \  
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null  
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \  
  https://pkg.jenkins.io/debian binary/ | sudo tee \  
  /etc/apt/sources.list.d/jenkins.list > /dev/null  
sudo apt-get update  
sudo apt-get install jenkins -y
```

The terminal shows the command being run and its output. At the bottom, there is a standard Linux terminal key binding menu.

21

A screenshot of a terminal window titled "script2.sh". The terminal is running on an AWS EC2 instance. The command entered is:

```
sudo apt update  
sudo apt install openjdk-11-jdk -y  
sudo apt install docker.io -y  
sudo apt update  
sudo apt upgrade -y  
sudo apt install -y curl apt-transport-https ca-certificates software-properties-common  
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -  
sudo add-apt-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"  
sudo swapoff -a  
sudo apt update  
sudo apt install -y kubelet kubeadm kubectl
```

The terminal shows the command being run and its output. At the bottom, there is a standard Linux terminal key binding menu.

22

The screenshot shows a terminal window titled "script2.sh" running on an AWS CloudShell session. The terminal displays the following command sequence:

```
sudo apt update
sudo apt install openjdk-11-jdk -y
sudo apt install docker.io -y
sudo apt update
sudo apt upgrade -y
sudo apt install -y curl apt-transport-https ca-certificates software-properties-common
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
sudo add-apt-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"
sudo swapoff -a
sudo apt update
sudo apt install -y kubelet kubeadm kubectl
```

At the bottom of the terminal, the instance ID "i-07343977e00c80bf7 (machine1)" and its IP addresses ("PublicIPs: 54.212.233.119 PrivateIPs: 172.31.1.222") are displayed. The terminal window has a standard Linux-style menu bar at the top.

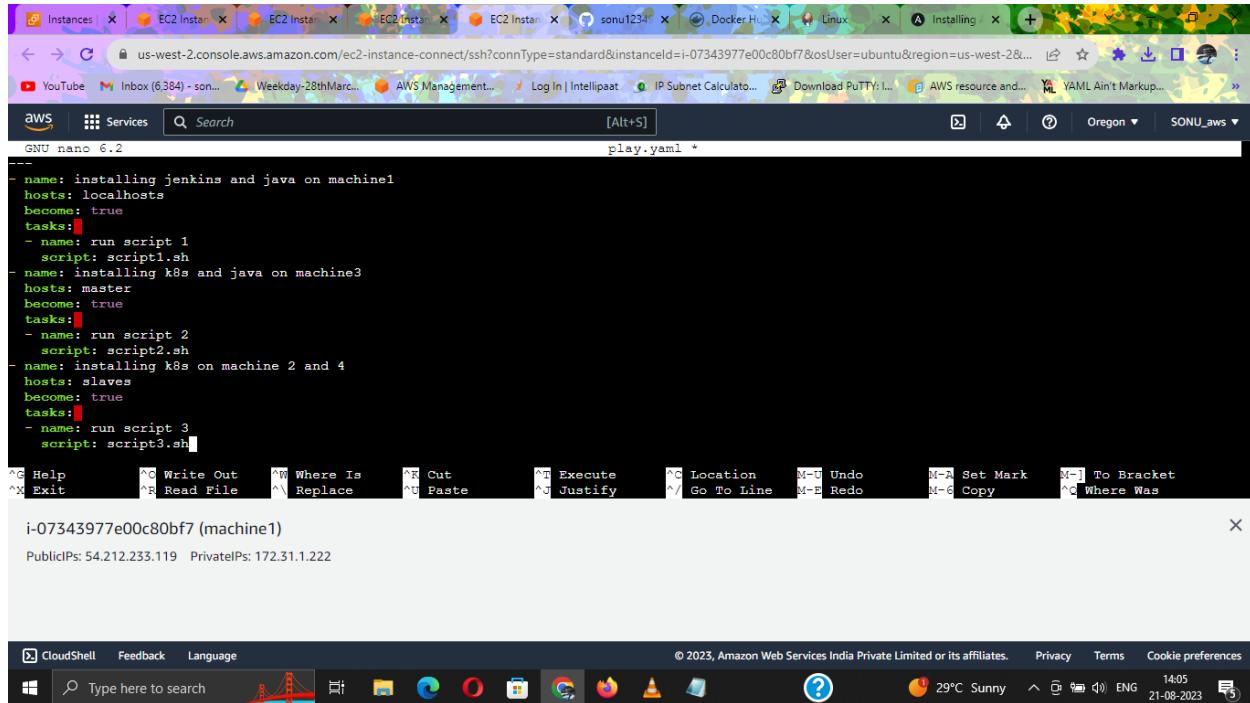
23

The screenshot shows a terminal window titled "script3.sh" running on an AWS CloudShell session. The terminal displays the same command sequence as in the previous screenshot:

```
sudo apt update
sudo apt install docker.io -y
sudo apt update
sudo apt upgrade -y
sudo apt install -y curl apt-transport-https ca-certificates software-properties-common
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
sudo add-apt-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"
sudo swapoff -a
sudo apt update
sudo apt install -y kubelet kubeadm kubectl
```

At the bottom of the terminal, the instance ID "i-07343977e00c80bf7 (machine1)" and its IP addresses ("PublicIPs: 54.212.233.119 PrivateIPs: 172.31.1.222") are displayed. The terminal window has a standard Linux-style menu bar at the top.

24 we run required packages using playbooks



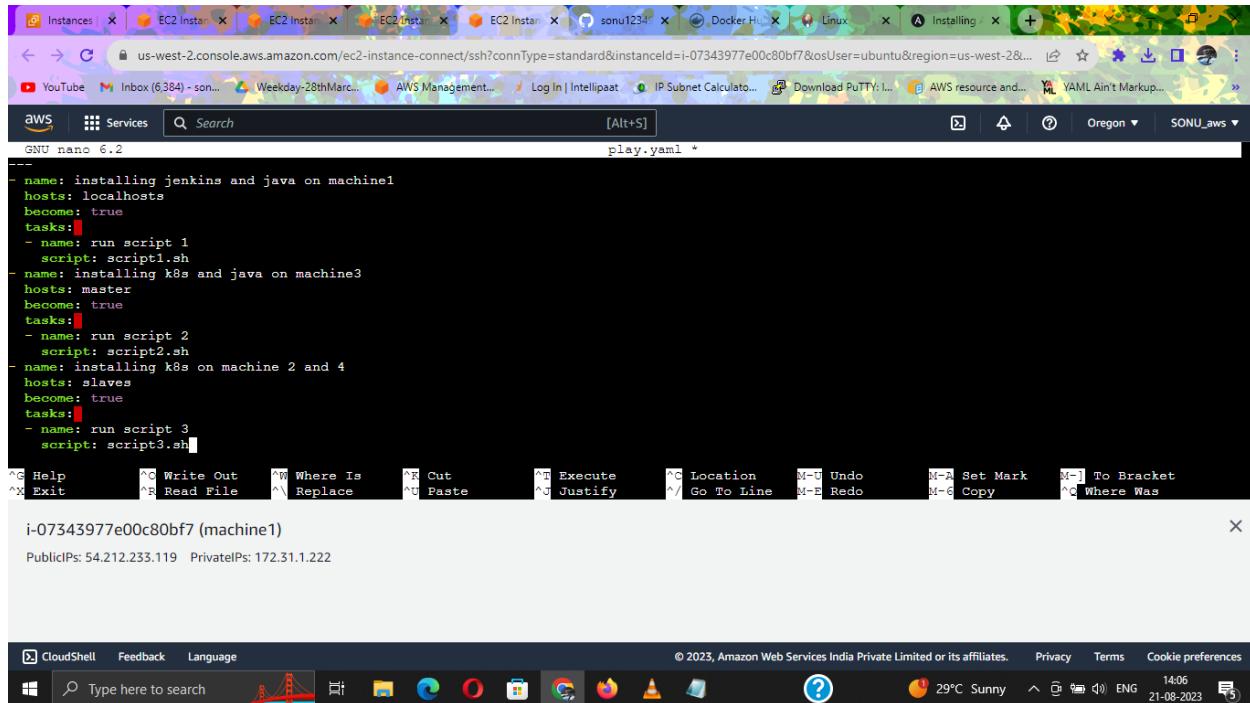
```
GNU nano 6.2                               play.yaml *
```

```
---  
- name: installing jenkins and java on machine1  
  hosts: localhost  
  become: true  
  tasks:  
    - name: run script 1  
      script: script1.sh  
- name: installing k8s and java on machine3  
  hosts: master  
  become: true  
  tasks:  
    - name: run script 2  
      script: script2.sh  
- name: installing k8s on machine 2 and 4  
  hosts: slaves  
  become: true  
  tasks:  
    - name: run script 3  
      script: script3.sh
```

i-07343977e00c80bf7 (machine1)
PublicIPs: 54.212.233.119 PrivateIPs: 172.31.1.222

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 29°C Sunny 14:05 21-08-2023

25



```
GNU nano 6.2                               play.yaml *
```

```
---  
- name: installing jenkins and java on machine1  
  hosts: localhost  
  become: true  
  tasks:  
    - name: run script 1  
      script: script1.sh  
- name: installing k8s and java on machine3  
  hosts: master  
  become: true  
  tasks:  
    - name: run script 2  
      script: script2.sh  
- name: installing k8s on machine 2 and 4  
  hosts: slaves  
  become: true  
  tasks:  
    - name: run script 3  
      script: script3.sh
```

i-07343977e00c80bf7 (machine1)
PublicIPs: 54.212.233.119 PrivateIPs: 172.31.1.222

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 29°C Sunny 14:06 21-08-2023

26

A screenshot of a browser window displaying a terminal session on an AWS EC2 instance. The terminal shows the user running several commands to check for files and run an Ansible playbook:

```
ubuntu@ip-172-31-1-222:~$ sudo nano script1.sh
ubuntu@ip-172-31-1-222:~$ sudo nano script2.sh
ubuntu@ip-172-31-1-222:~$ sudo nano script3.sh
ubuntu@ip-172-31-1-222:~$ sudo nano play.yaml
ubuntu@ip-172-31-1-222:~$ ansible-playbook play.yaml --syntax-check
[WARNING]: Could not match supplied host pattern, ignoring: localhosts

playbook: play.yaml
ubuntu@ip-172-31-1-222:~$
```

The terminal window has a title bar "i-07343977e00c80bf7 (machine1)" and a status bar at the bottom indicating "PublicIPs: 54.212.233.119 PrivateIPs: 172.31.1.222".

27

A screenshot of a browser window displaying the output of an Ansible playbook run across multiple hosts. The terminal session shows the following tasks being executed:

```
TASK [Gathering Facts] *****
ok: [172.31.3.95]

TASK [run script 2] *****
skipping: [172.31.3.95]

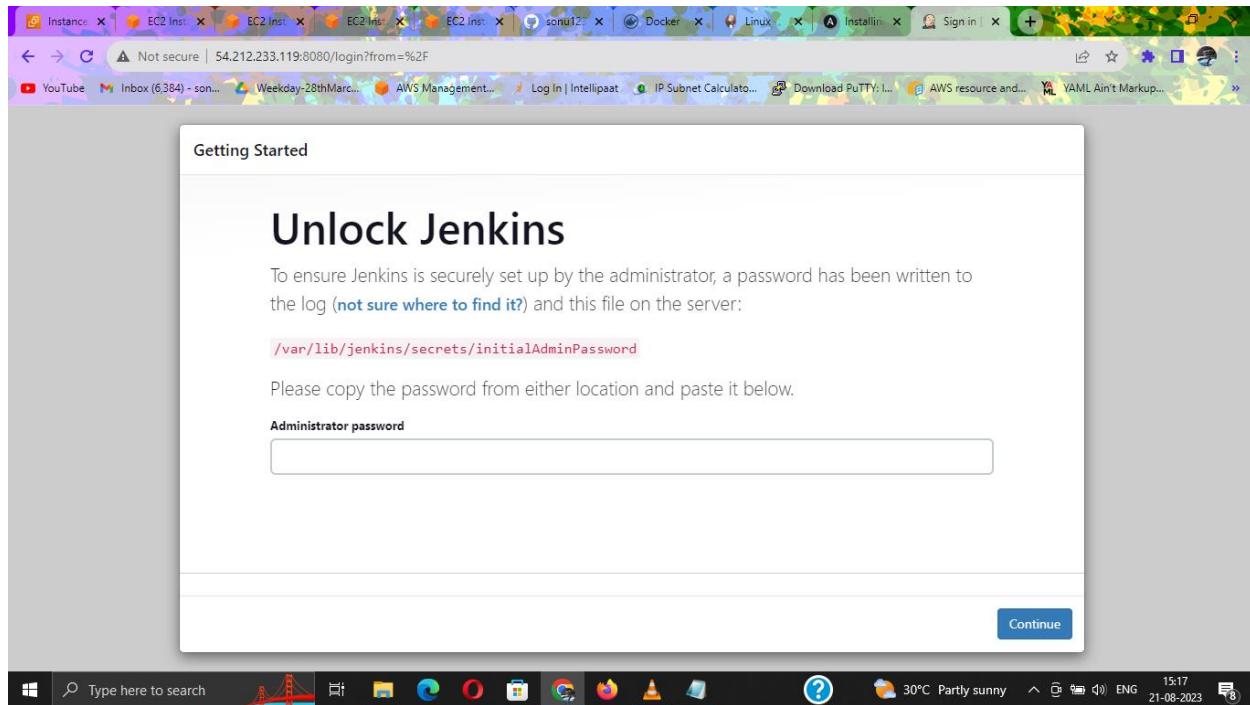
PLAY [installing k8s on machine 2 and 4] *****
TASK [Gathering Facts] *****
ok: [172.31.18.60]
ok: [172.31.18.81]

TASK [run script 3] *****
skipping: [172.31.18.60]
skipping: [172.31.18.81]

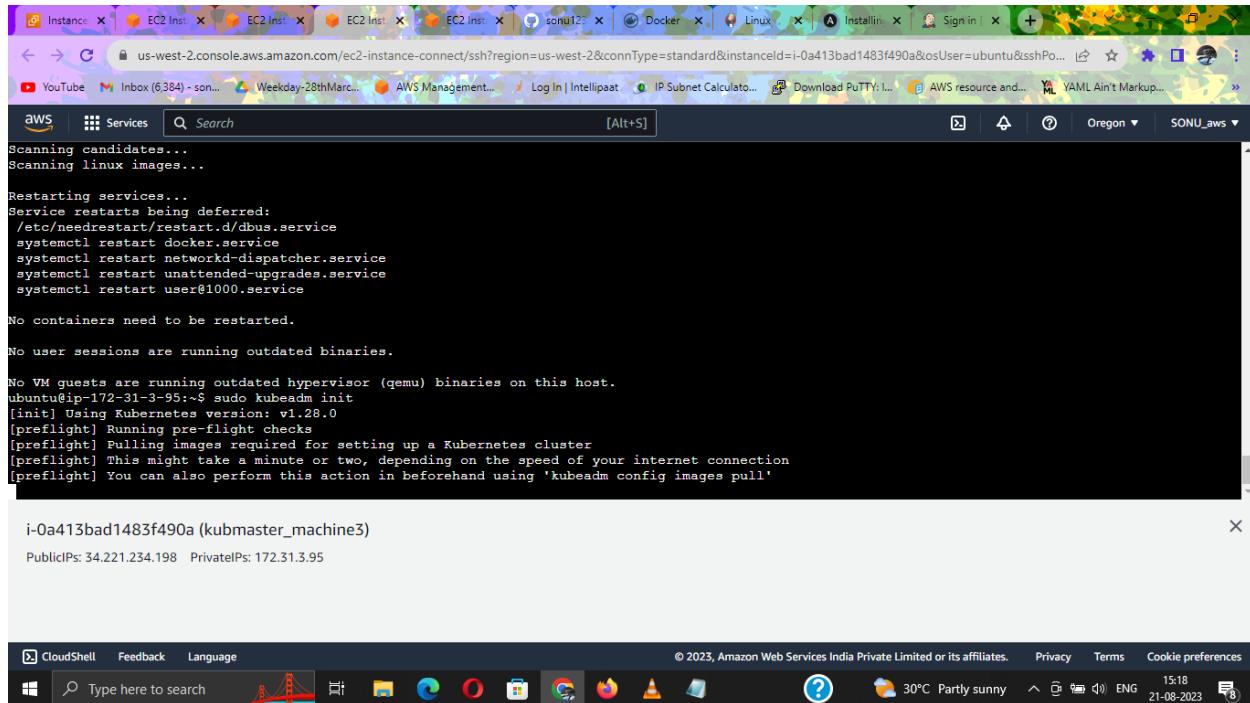
PLAY RECAP *****
172.31.18.60      : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
172.31.18.81      : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
172.31.3.95      : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

The terminal window has a title bar "i-07343977e00c80bf7 (machine1)" and a status bar at the bottom indicating "PublicIPs: 54.212.233.119 PrivateIPs: 172.31.1.222".

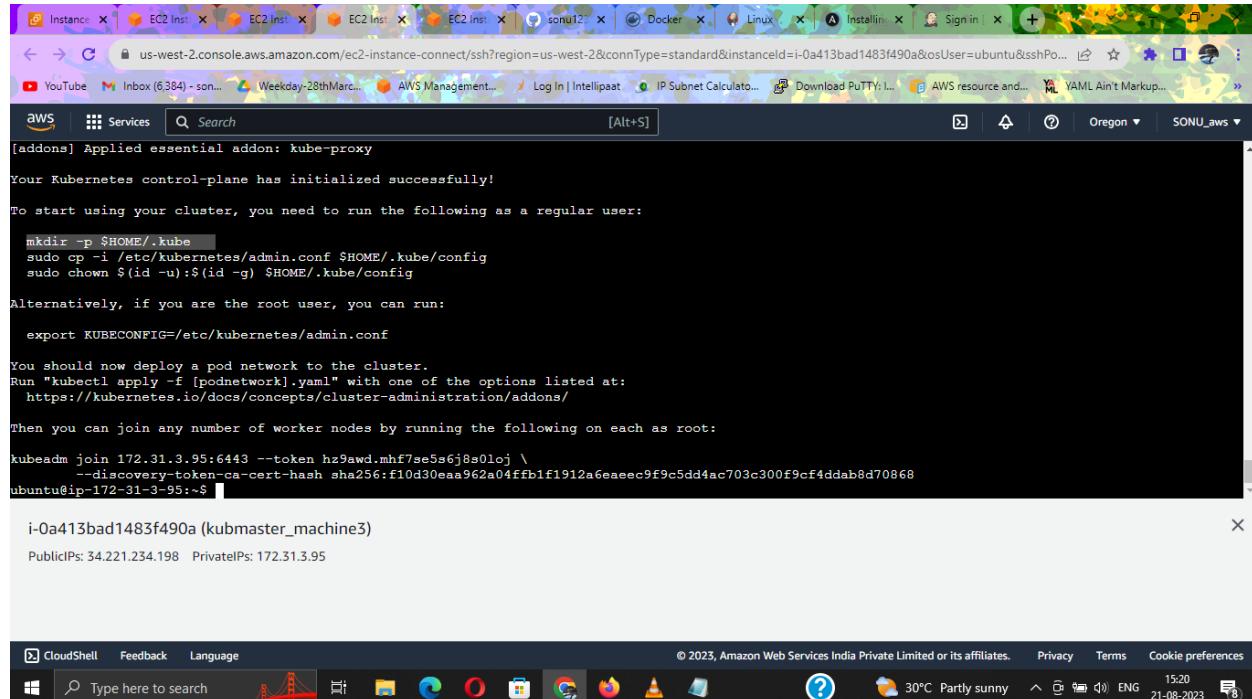
28 our jenkins is intalled



29



30



```
[addons] Applied essential addon: kube-proxy
Your Kubernetes control-plane has initialized successfully!
To start using your cluster, you need to run the following as a regular user:
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config

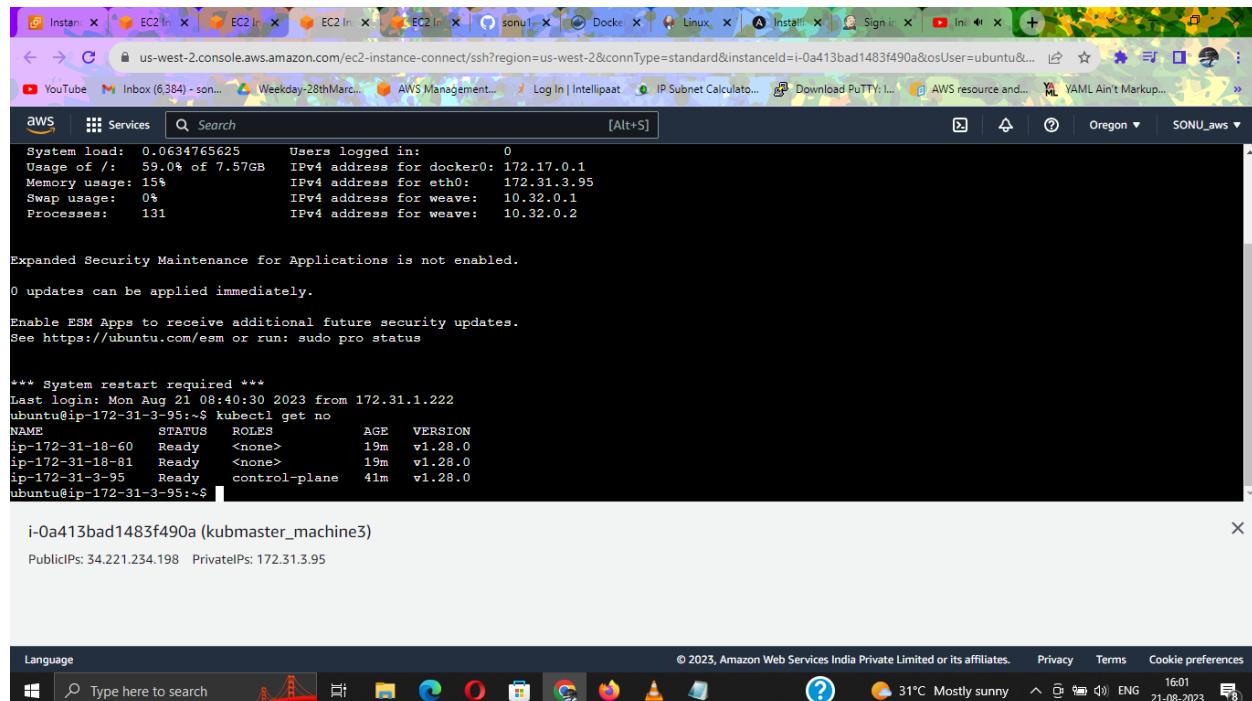
Alternatively, if you are the root user, you can run:
export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:
kubeadm join 172.31.3.95:6443 --token hz9awd.mhf7se5s6j8s0loj \
    --discovery-token-ca-cert-hash sha256:f10d30eaa962a04ffbf1f1912a6eaec9f9c5dd4ac703c300f9cf4ddab8d70868
ubuntu@ip-172-31-3-95:~$
```

i-0a413bad1483f490a (kubmaster_machine3)
PublicIPs: 34.221.234.198 PrivateIPs: 172.31.3.95

31



```
System load: 0.0634765625  Users logged in: 0
Usage of /: 59.0% of 7.57GB  IPv4 address for docker0: 172.17.0.1
Memory usage: 15%
Swap usage: 0%
Processes: 131  IPv4 address for eth0: 172.31.3.95
                IPv4 address for weave: 10.32.0.1
                IPv4 address for weave: 10.32.0.2

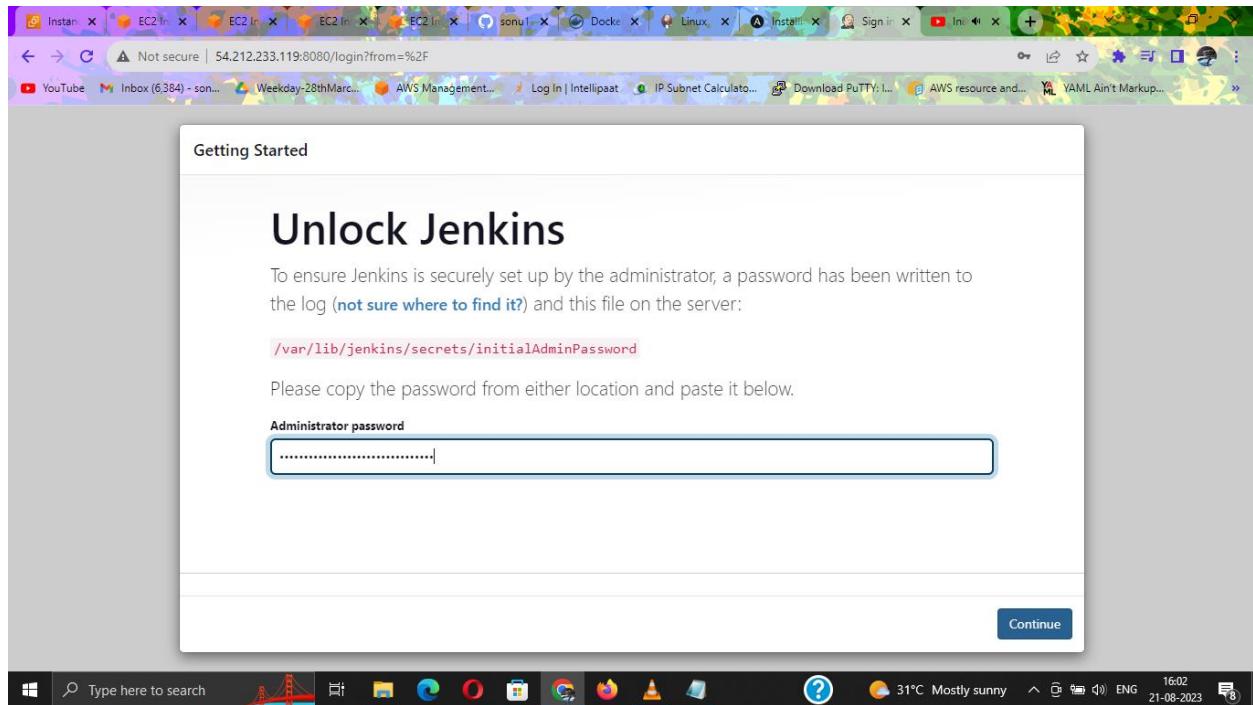
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

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

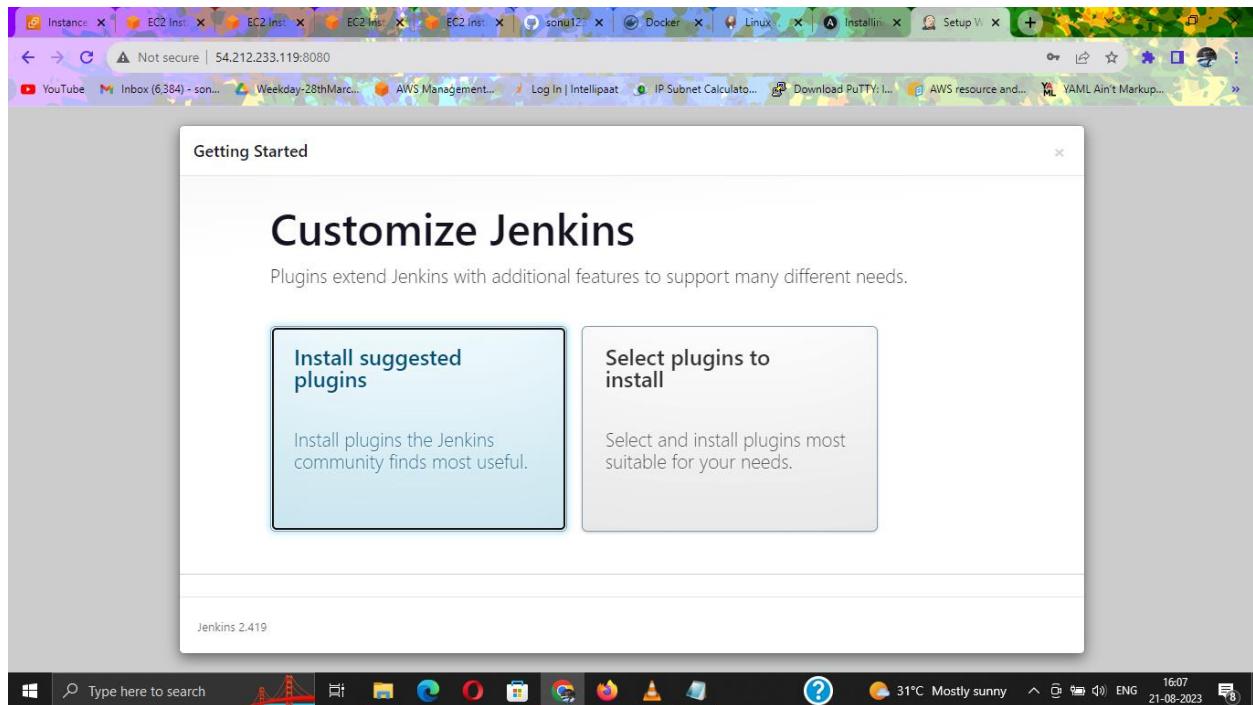
*** System restart required ***
Last login: Mon Aug 21 08:40:30 2023 from 172.31.1.222
ubuntu@ip-172-31-3-95:~$ kubectl get no
NAME      STATUS   ROLES      AGE     VERSION
ip-172-31-18-60 Ready    <none>    19m    v1.28.0
ip-172-31-18-81 Ready    <none>    19m    v1.28.0
ip-172-31-3-95 Ready    control-plane 41m    v1.28.0
ubuntu@ip-172-31-3-95:~$
```

i-0a413bad1483f490a (kubmaster_machine3)
PublicIPs: 34.221.234.198 PrivateIPs: 172.31.3.95

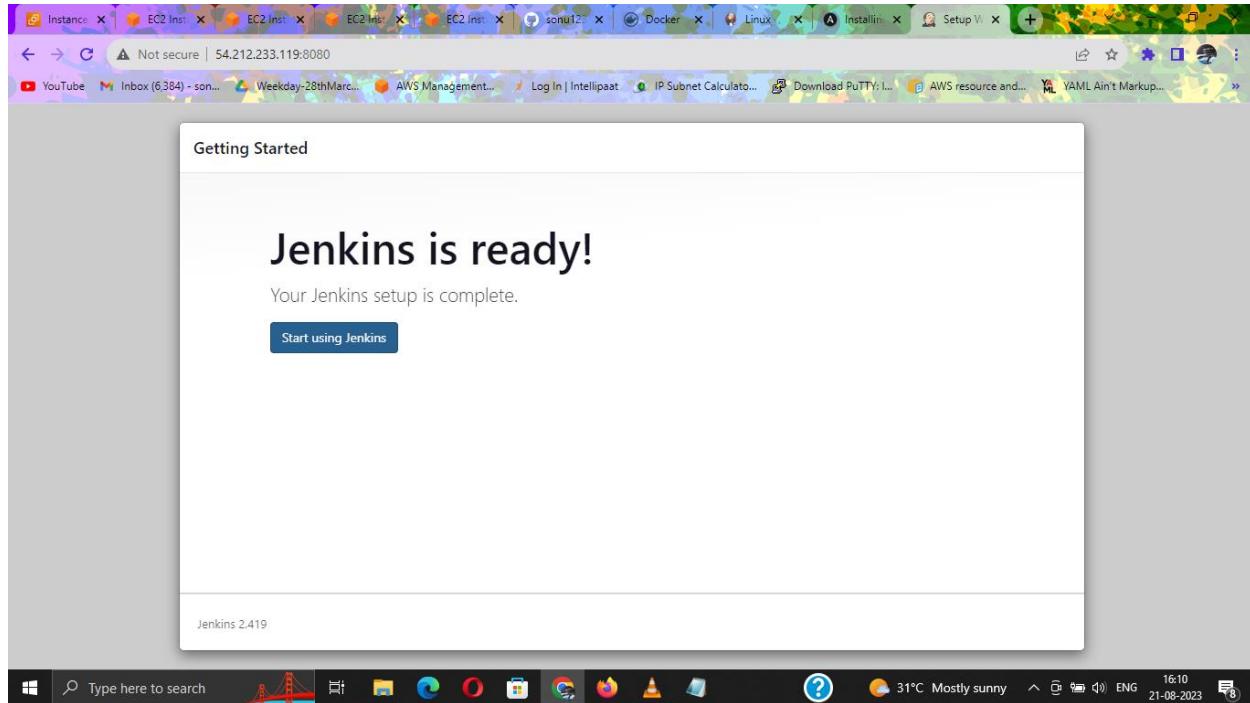
32



33



34



35

The screenshot shows the Jenkins 'Nodes' management page. The top navigation bar includes links for 'Dashboard', 'Manage Jenkins', and 'Nodes'. The main content area is titled 'Nodes' and contains a table with the following data:

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
1	Built-In Node	Linux (amd64)	In sync	3.76 GB	0 B	3.76 GB	0ms
	Data obtained	2 min 40 sec	2 min 40 sec	2 min 40 sec	2 min 40 sec	2 min 40 sec	2 min 40 sec

Below the table, there is a section titled 'Build Executor Status' showing '1 Idle' and '2 Idle'. The bottom right corner of the page indicates 'REST API' and 'Jenkins 2.419'. The browser's address bar shows the URL '54.212.233.119:8080/manage/computer/new'.

36

The screenshot shows the Jenkins 'New node' configuration page. The 'Node name' field contains 'k8s-master'. The 'Type' section is set to 'Permanent Agent'. A detailed description explains that this adds a plain, permanent agent to Jenkins. Below the description is a 'Create' button. At the top right, there are links for 'Search (CTRL+K)', 'admin', and 'log out'. The browser's address bar shows 'Not secure | 54.212.233.119:8080/manage/computer/new'. The taskbar at the bottom includes icons for File Explorer, Edge, File, Task View, Start, Taskbar settings, and a search bar.

37

The screenshot shows the Jenkins 'Create Item' dialog for adding a new node. The 'Type' dropdown is set to 'Private Key'. The 'Key' field contains an RSA private key:

```
-----BEGIN RSA PRIVATE KEY-----  
MIIEowIxDjQBgkqhkiG9w0BAQEwEQDfJ...  
-----END RSA PRIVATE KEY-----
```

The 'Passphrase' field is empty. At the bottom are 'Add' and 'Cancel' buttons. Below the dialog, the 'Node Properties' section has a 'Save' button. The browser's address bar shows 'Not secure | 54.212.233.119:8080/manage/computer/createtem'. The taskbar at the bottom includes icons for File Explorer, Edge, File, Task View, Start, Taskbar settings, and a search bar.

38

The screenshot shows a web browser window with multiple tabs open. The active tab is 'Manage Jenkins > Nodes'. The form fields are as follows:

- Host:** 172.31.3.95
- Credentials:** ubuntu
- Host Key Verification Strategy:** Known hosts file Verification Strategy
- Availability:** Keep this agent online as much as possible

A 'Node Properties' section is visible at the bottom, containing checkboxes for 'Disable deferred wipeout on this node', 'Environment variables', and 'Tool Locations'. A 'Save' button is located at the bottom left of the properties section.

39

The screenshot shows a web browser window with multiple tabs open. The active tab is 'Manage Jenkins > Nodes'. The form fields are as follows:

- Host Key Verification Strategy:** Non verifying Verification Strategy
- Availability:** Keep this agent online as much as possible

A 'Node Properties' section is visible at the bottom, containing checkboxes for 'Disable deferred wipeout on this node', 'Environment variables', and 'Tool Locations'. A 'Save' button is located at the bottom left of the properties section.

40

The screenshot shows the Jenkins Nodes management interface. On the left, there are navigation links for Dashboard, Manage Jenkins, and Nodes. Below these are sections for Clouds, Node Monitoring, Build Queue (empty), and Build Executor Status (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. Two nodes are listed:

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
1	Built-In Node	Linux (amd64)	In sync	3.73 GB	0 B	3.73 GB	0ms
2	k8s-master	Linux (amd64)	In sync	3.06 GB	0 B	3.06 GB	79ms

A "New Node" button is located at the top right of the table. The bottom of the screen shows a Windows taskbar with various icons and system status.

41

The screenshot shows the Jenkins Nodes management interface. The layout is identical to the previous one, with navigation links for Dashboard, Manage Jenkins, and Nodes. The "Build Queue" and "Build Executor Status" sections remain empty. The main "Nodes" table shows the same two nodes: Built-In Node and k8s-master. The table structure and data are identical to the previous screenshot. A "New Node" button is present at the top right. The bottom of the screen shows a Windows taskbar with various icons and system status.

42

The screenshot shows a web browser window with the Jenkins interface. The address bar indicates the URL is [Not secure | 54.212.233.119:8080/manage/credentials/](http://54.212.233.119:8080/manage/credentials/). The Jenkins logo is at the top left, and the user 'admin' is logged in. The main content area is titled 'Credentials' and shows a table with one row:

T	P	Store	Domain	ID	Name
		System	(global)	33fceef4f-5724-40f1-a03c-5c310e15b9a3	ubuntu

Below the table, a section titled 'Stores scoped to Jenkins' lists a single store named 'System' under the domain '(global)'. The Windows taskbar at the bottom shows various pinned icons and the date/time as 21-08-2023.

43

The screenshot shows a web browser window with the Jenkins interface. The address bar indicates the URL is [Not secure | 54.212.233.119:8080/manage/credentials/store/system/domain/_/newCredentials](http://54.212.233.119:8080/manage/credentials/store/system/domain/_/newCredentials). The Jenkins logo is at the top left, and the user 'admin' is logged in. The main content area is titled 'Global credentials (unrestricted)' and contains a form for creating a new credential:

Scope ?
Global (Jenkins, nodes, items, all child items, etc)

Username ?
sonu1234996

Treat username as secret ?

Password ?

ID ?

Description ?

Create

The Windows taskbar at the bottom shows various pinned icons and the date/time as 21-08-2023.

44

The screenshot shows the Jenkins Global credentials (unrestricted) page. It lists two entries:

ID	Name	Kind	Description
33fce4f-5724-40f1-a03c-5c310e15b9a3	ubuntu	SSH Username with private key	
65ba6a62-f8c6-427d-9f7c-7b54664e73d3	sonu12349964*****	Username with password	

Icon: S M L

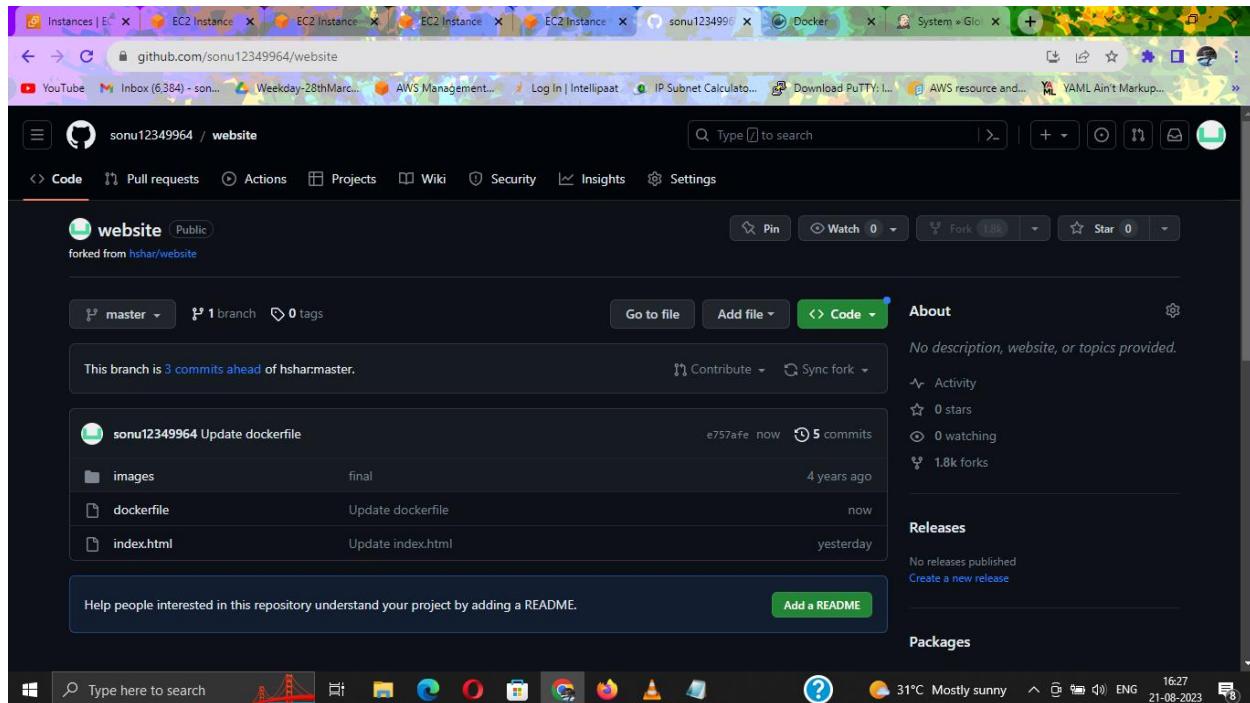
REST API Jenkins 2.419

45

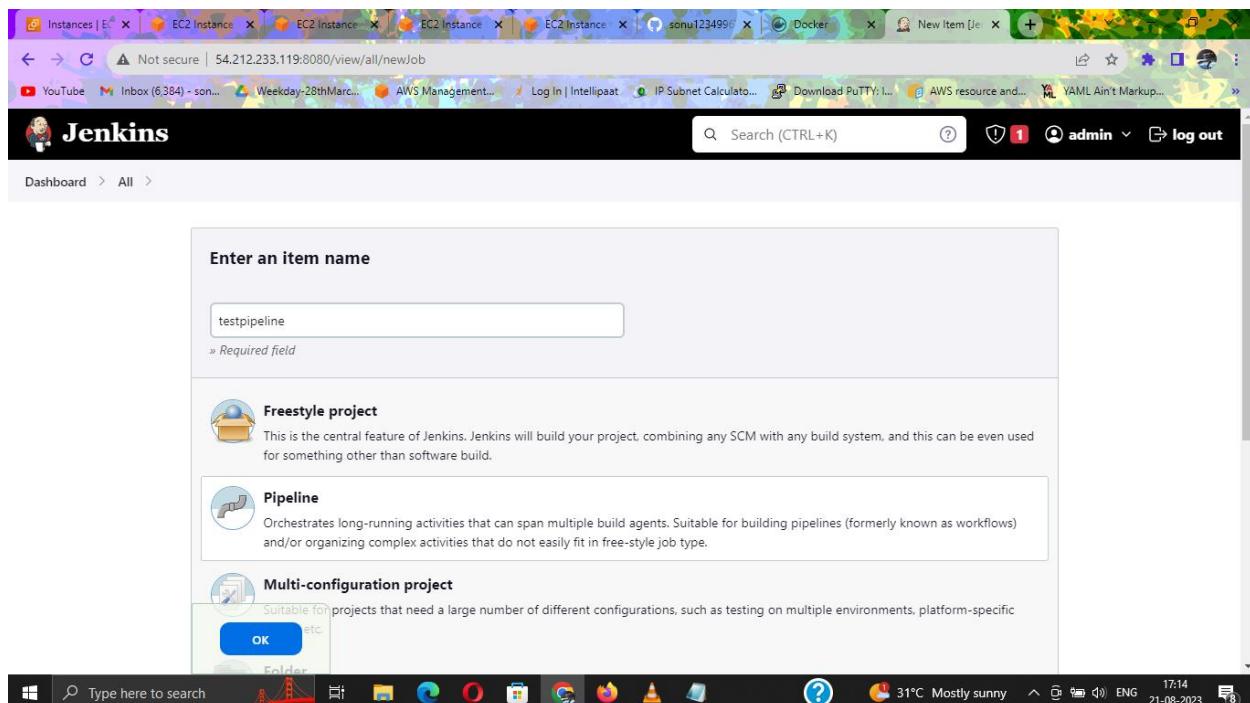
The screenshot shows the GitHub website editor for the repository sonu12349964 / website. The Dockerfile content is:

```
FROM ubuntu/apache2
COPY . /var/www/html/
```

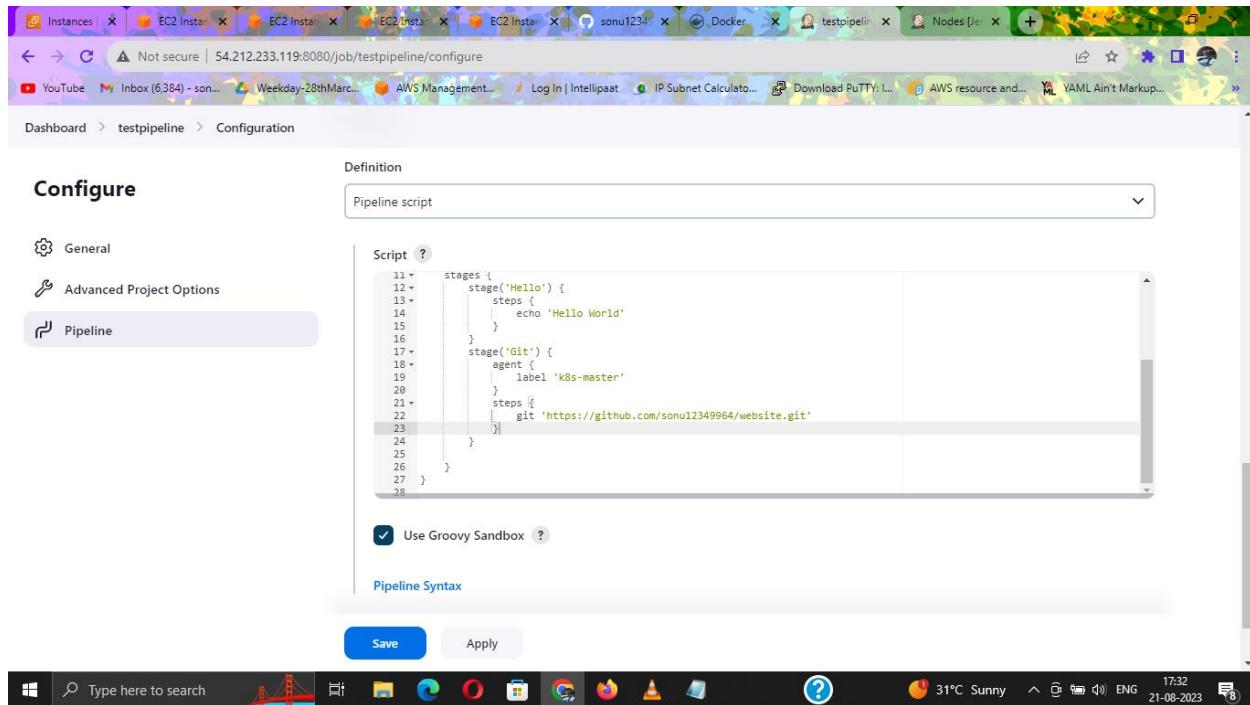
46



47

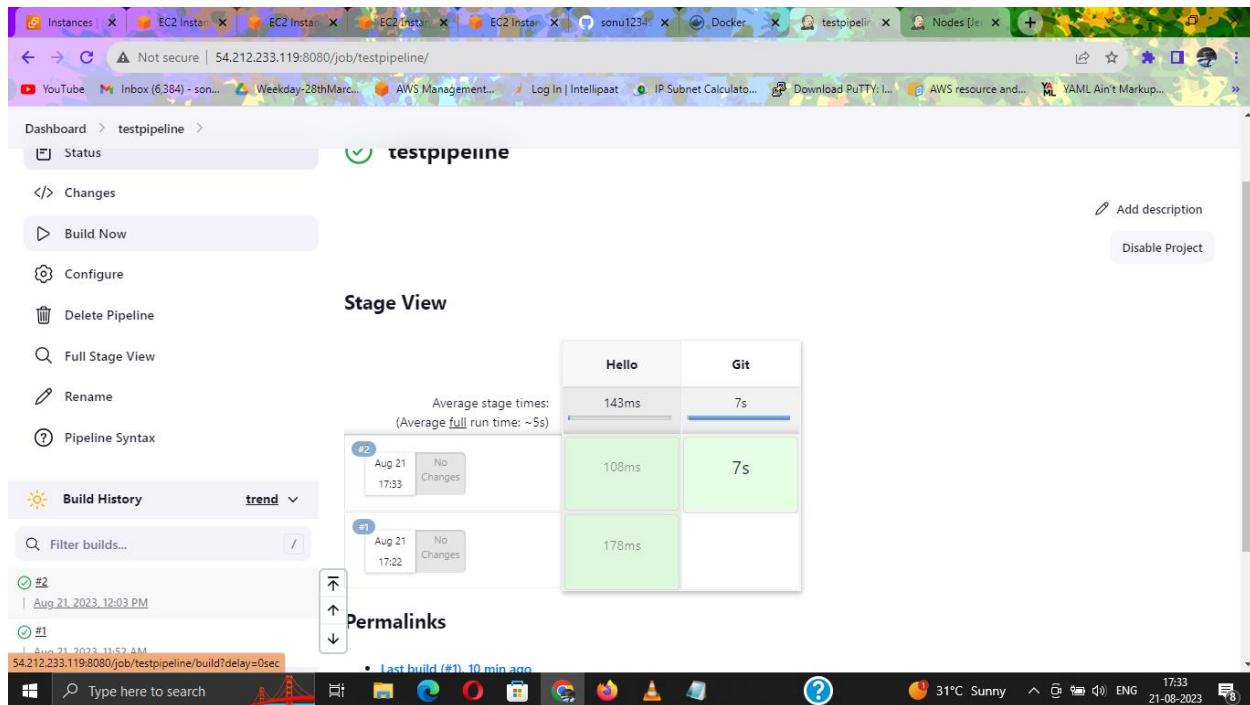


48 we create our jobs and test



```
stages {
    stage('Hello') {
        steps {
            echo 'Hello World'
        }
    }
    stage('Git') {
        agent {
            label 'k8s-master'
        }
        steps {
            git 'https://github.com/sonu12349964/website.git'
        }
    }
}
```

49



50

Dashboard > testpipeline >

Status

testpipeline

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

Rename

Pipeline Syntax

Build History trend

Filter builds...

#2 | Aug 21, 2023, 12:03 PM

#1 | Aug 21, 2023, 11:52 AM

Last build (#1), 10 min ago

Average stage times:
(Average full run time: ~5s)

Hello	Git
143ms	7s
108ms	7s
178ms	

Permalinks

31°C Sunny 17:34 21-08-2023

51

```
*** System restart required ***
Last login: Mon Aug 21 08:40:30 2023 from 172.31.1.222
ubuntu@ip-172-31-3-95:~$ kubectl get no
NAME        STATUS   ROLES      AGE     VERSION
ip-172-31-18-60 Ready    <none>    19m    v1.28.0
ip-172-31-18-81 Ready    <none>    19m    v1.28.0
ip-172-31-3-95 Ready    control-plane 41m    v1.28.0
ubuntu@ip-172-31-3-95:~$ ls
jenkins
ubuntu@ip-172-31-3-95:~$ cd jenkins
ubuntu@ip-172-31-3-95:~/jenkins$ ls
remoting  remoting.jar  workspace
ubuntu@ip-172-31-3-95:~/jenkins$ cd workspace
ubuntu@ip-172-31-3-95:~/jenkins/workspace$ ls
testpipeline
ubuntu@ip-172-31-3-95:~/jenkins/workspace$ cd testpipeline
-bash: cd: testpipeline: No such file or directory
ubuntu@ip-172-31-3-95:~/jenkins/workspace$ cd testpipeline
ubuntu@ip-172-31-3-95:~/jenkins/workspace/testpipeline$ ls
dockerrfile  images  index.html
ubuntu@ip-172-31-3-95:~/jenkins/workspace/testpipeline$
```

i-0a413bad1483f490a (kubmaster_machine3)

Public IPs: 34.221.234.198 Private IPs: 172.31.3.95

Language

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

31°C Sunny 17:36 21-08-2023

52

The screenshot shows the Jenkins Pipeline configuration page. The pipeline script is defined as follows:

```
1> }  
15 }  
16 } stage('Git') {  
17 agent {  
18 | label 'k8s-master'  
19 }  
20 steps {  
21 | git 'https://github.com/sonu12349964/website.git'  
22 }  
23 }  
24 stage('Docker') {  
25 steps {  
26 | bash 'sudo docker build /home/ubuntu/jenkins/workspace/testpipeline -t sonu12349964/project2'  
27 | bash 'sudo echo $DOCKERHUB_CREDENTIALS_PSW | sudo docker login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin'  
28 | bash 'sudo docker push sonu12349964/project2'  
29 }  
30 }  
31 }  
32 }
```

A checkbox for "Use Groovy Sandbox" is checked.

53

The screenshot shows the Jenkins Pipeline configuration page. The pipeline script is defined as follows:

```
1> }  
15 }  
16 stage('Git') {  
17 agent {  
18 | label 'k8s-master'  
19 }  
20 steps {  
21 | git 'https://github.com/sonu12349964/website.git'  
22 }  
23 }  
24 stage('Docker') {  
25 steps {  
26 | sh 'sudo docker build /home/ubuntu/jenkins/workspace/testpipeline -t sonu12349964/project2'  
27 | sh '$DOCKERHUB_CREDENTIALS_PSW | sudo docker login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin'  
28 | sh 'sudo docker push sonu12349964/project2'  
29 }  
30 }  
31 }  
32 }  
33 }
```

A checkbox for "Use Groovy Sandbox" is checked.

54

The screenshot shows a CI/CD pipeline interface. On the left, there's a sidebar with options like 'Configure', 'Delete Pipeline', 'Full Stage View', 'Rename', and 'Pipeline Syntax'. Below that is a 'Build History' section with a dropdown set to 'trend'. It lists several builds: #9 (Aug 21, 18:10, No Changes), #8 (Aug 21, 18:04, No Changes), #7 (Aug 21, 18:04, No Changes), #6 (Aug 21, 18:01, No Changes), and #5. To the right is a 'Stage View' section titled 'Stage View' with three columns: 'Hello', 'Git', and 'Docker'. Below each column is a horizontal bar showing average stage times: 124ms for Hello, 4s for Git, and 15s for Docker. The 'Git' bar is highlighted in blue.

55

The screenshot shows the Docker Hub user profile page for 'sonu12349964'. The top navigation bar includes links for 'Explore', 'Repositories', 'Organizations', 'Help', and 'Upgrade'. A search bar is present at the top. The main content area displays two repositories: 'sonu12349964 / project2' (inactive, 0 stars, 0 forks, public) and 'sonu12349964 / assignment3' (inactive, 0 stars, 2 forks, public). To the right, there's a promotional box for creating an organization, featuring icons for a lock, a key, and a padlock, with the text 'Create an Organization Manage Docker Hub repositories with your team'. At the bottom, there's a banner for 'DockerCon 2023: Our annual developer event is back – online & in person. Learn more.' The taskbar at the bottom shows various application icons and system status.

56

The screenshot shows a Microsoft Edge browser window with multiple tabs open, including EC2 instances and AWS Management Console. The main content area displays the Kubernetes documentation for Deployment controllers. On the left, there is a navigation sidebar with links like Home, Getting started, Concepts, Workloads, Deploy, and others. The central content area shows a YAML configuration for a Deployment named 'nginx-deployment' with three replicas, each labeled 'app: nginx'. The right sidebar contains links for editing the page, creating child pages, and reporting issues, along with a 'Use Case' section listing various deployment scenarios.

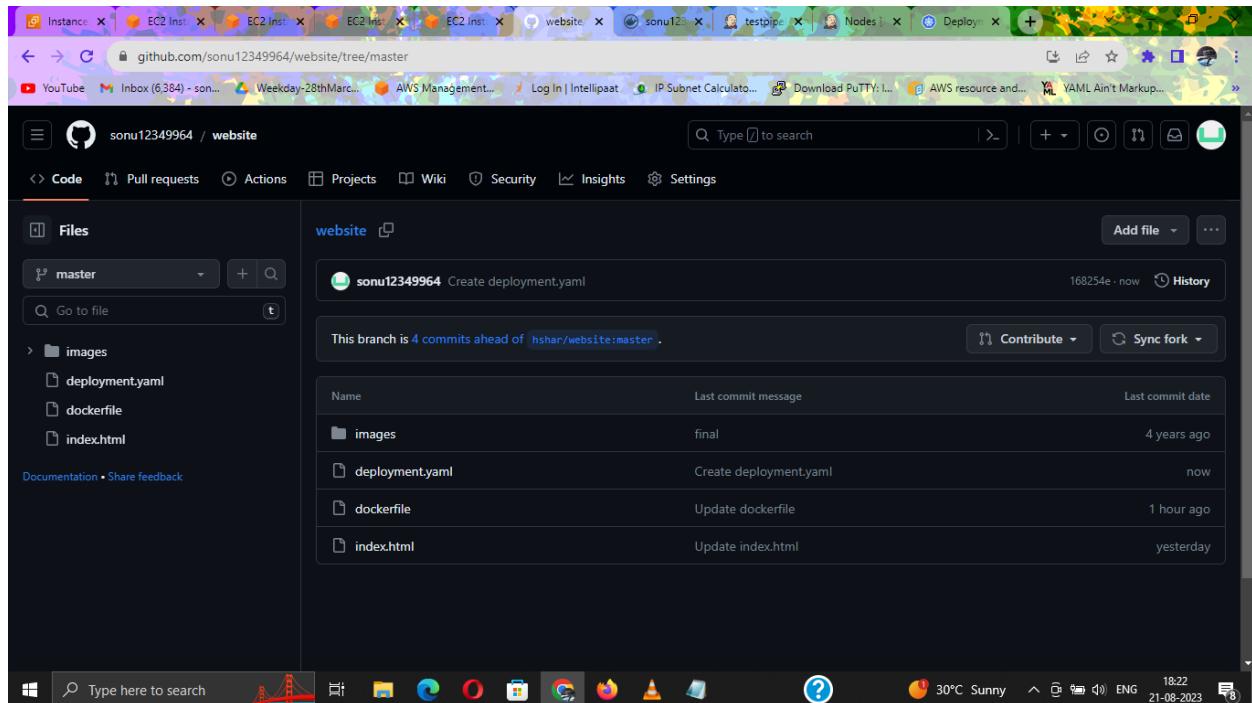
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.14.2
          ports:
            - containerPort: 80
```

57

The screenshot shows a Microsoft Edge browser window with multiple tabs open, including EC2 instances and AWS Management Console. The main content area displays a GitHub repository for a website. The repository structure on the left includes 'master' and files for 'images', 'dockerfile', and 'index.html'. The central content area shows the 'deployment.yaml' file being edited. The file content is identical to the one shown in the previous screenshot, defining a Deployment for 'nginx-deployment' with three replicas. A note at the bottom of the code editor indicates keyboard shortcuts for navigating between tabs and interactive elements.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
```

58



59

A screenshot of the Kubernetes documentation page for 'Services'. It shows an example YAML manifest for a NodePort service:

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  type: NodePort
  selector:
    app.kubernetes.io/name: MyApp
  ports:
    # By default and for convenience, the `targetPort` is set to the same value as the `port` field.
    - port: 80
      targetPort: 80
      # Optional field
      # By default and for convenience, the Kubernetes control plane will allocate a port from a range (nodePort: 30007
      nodePort: 30007
```

The page also includes a note: "Reserve Nodeport Ranges to avoid collisions when port assigning".

60

The screenshot shows a GitHub repository interface for a project named 'sonu12349964 / website'. The 'service.yaml' file is open in the editor. The code content is as follows:

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  type: NodePort
  selector:
    app: nginx
  ports:
    - port: 80
      targetPort: 80
      # Optional field
    # By default and for convenience, the 'nodePort' is set to the same value as the 'port' field.
    # By default and for convenience, the Kubernetes control plane will allocate a port from a range (default: 30000-32767)
    nodePort: 30008
```

Below the code, there is a note: "Use **Control + Shift + m** to toggle the **tab** key moving focus. Alternatively, use **esc** then **tab** to move to the next interactive element on the page."

61

The screenshot shows the same GitHub repository interface. The commit history for the 'service.yaml' file is displayed. The commit message is "Create service.yaml". The commit was made by user 'sonu12349964' at 00:03 on 21-08-2023. The commit message is "Create service.yaml". The commit was made by user 'sonu12349964' at 00:03 on 21-08-2023.

Name	Last commit message	Last commit date
final		4 years ago
Create deployment.yaml		3 minutes ago
Update dockerfile		2 hours ago
Update index.html		yesterday
Create service.yaml		now

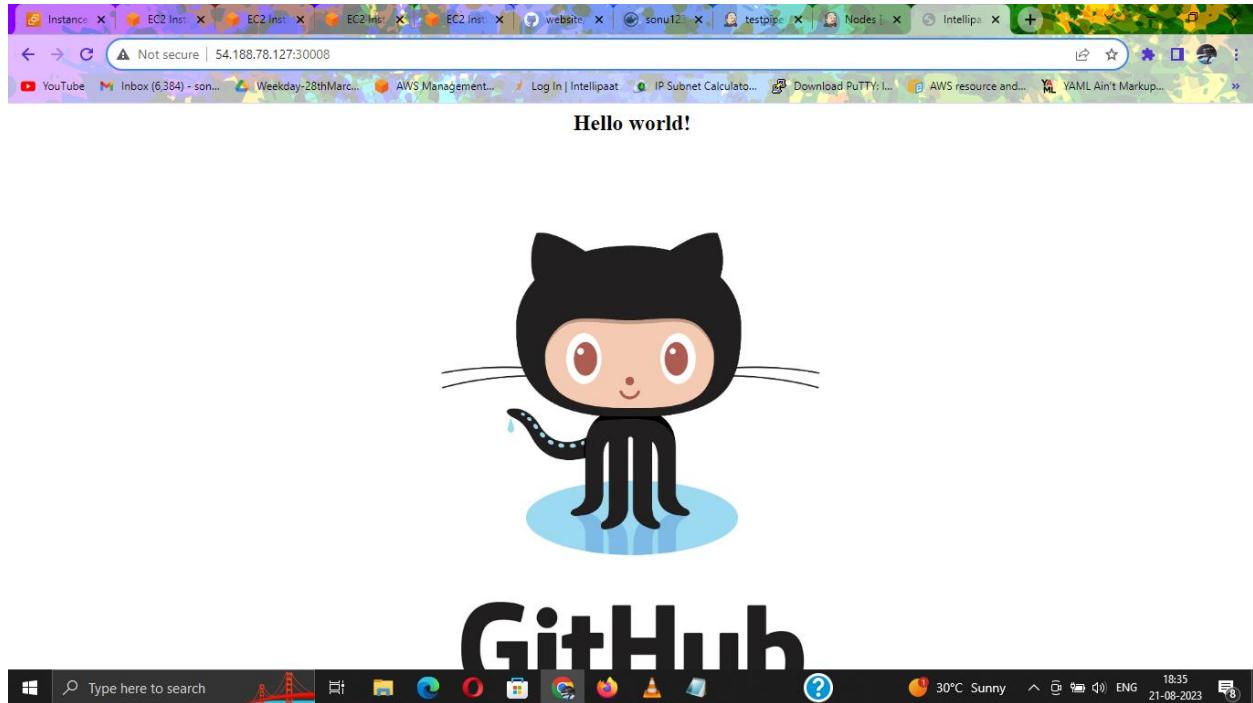
62

The screenshot shows a Jenkins pipeline interface titled "Stage View". It displays three stages: "Hello", "Git", and "Docker". The "Hello" stage has an average time of 113ms. The "Git" stage has an average time of 2s. The "Docker" stage has an average time of 10s. Below the stages, there is a table showing build history for builds #10, #9, #8, and #7. Build #10 is the most recent, showing 2 commits and a duration of 81ms for "Hello", 739ms for "Git", and 5s for "Docker". Build #9 shows 86ms for "Hello", 714ms for "Git", and 15s for "Docker". Builds #8 and #7 show "No Changes". On the left sidebar, there are options like "Configure", "Delete Pipeline", "Full Stage View", "Rename", and "Pipeline Syntax". A "Build History" section lists builds #10 through #7. The bottom of the screen shows a Windows taskbar with various icons and system status.

63

The screenshot shows a Jenkins pipeline interface titled "testpipeline". It displays four stages: "Hello", "Git", "Docker", and "k8s". The "Hello" stage has an average time of 105ms. The "Git" stage has an average time of 3s. The "Docker" stage has an average time of 10s. The "k8s" stage has an average time of 1s. Below the stages, there is a table showing build history for builds #11 and #10. Build #11 is the most recent, showing 72ms for "Hello", 4s for "Git", 7s for "Docker", and 1s for "k8s". Build #10 shows 81ms for "Hello", 739ms for "Git", 7s for "Docker", and 576ms for "k8s", which is highlighted in red with the word "failed". On the left sidebar, there are options like "Status", "Changes", "Build Now", "Configure", "Delete Pipeline", "Full Stage View", "Rename", and "Pipeline Syntax". A "Build History" section lists builds #11 and #10. The bottom of the screen shows a Windows taskbar with various icons and system status.

64



65

