

## 1<sup>st</sup> create manual ec2 instance and install terraform and ansible

After that create terrafolder > main.tf

```
ubuntu@ip-172-31-83-59:~$ sudo mkdir terrafolder
ubuntu@ip-172-31-83-59:~$ sudo chown -R ubuntu:ubuntu ~/terrafolder
ubuntu@ip-172-31-83-59:~$ cd terrafolder/
ubuntu@ip-172-31-83-59:~/terrafolder$ sudo nano main.tf
ubuntu@ip-172-31-83-59:~/terrafolder$ terraform init
Terraform has no command named "inie". Did you mean "init"?
To see all of Terraform's top-level commands, run:
  terraform -help

Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.97.0...
- Installed hashicorp/aws v5.97.0 (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,
rerun 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-172-31-83-59:~/terrafolder$ terraform plan
Terraform has no command named "pln". Did you mean "plan"?
To see all of Terraform's top-level commands, run:
  terraform -help

ubuntu@ip-172-31-83-59:~/terrafolder$ terraform plan
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

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-172-31-83-59:~/terrafolder$ terraform apply
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_instance.k8s_master_jenkins_slave will be created
+ resource "aws_instance" "k8s_master_jenkins_slave" {
  + ami                                = "ami-0f9de6e2d2f067fca"
  + arn                                = "(known after apply)"
  + associate_public_ip_address        = "(known after apply)"
  + availability_zone                  = "(known after apply)"
  + cpu_core_count                     = "(known after apply)"
  + cpu_threads_per_core              = "(known after apply)"
  + disable_api_stop                  = "(known after apply)"
  + disable_api_termination           = "(known after apply)"
  + ebs_optimized                      = "(known after apply)"
  + enable_primary_ipv6                = "(known after apply)"
  + get_password_data                 = false
  + host_id                            = "(known after apply)"
  + host_resource_group_arn            = "(known after apply)"
  + iam_instance_profile               = "(known after apply)"
  + id                                 = "(known after apply)"
  + instance_initiated_shutdown_behavior = "(known after apply)"
  + instance_lifecycle
```

```

Jenkins-master/Ansible-maste X K8s-Master/Jenkins-Slave X K8s-Slave1 X K8s-Slave2 X
+ cpu_options (known after apply)
+ ebs_block_device (known after apply)
+ enclave_options (known after apply)
+ ephemeral_block_device (known after apply)
+ instance_market_options (known after apply)
+ maintenance_options (known after apply)
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (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.k8s_slave1: Creating...
aws_instance.k8s_master_jenkins_slave: Creating...
aws_instance.k8s_slave2: Creating...
aws_instance.k8s_slave1: Still creating... [10s elapsed]
aws_instance.k8s_master_jenkins_slave: Still creating... [10s elapsed]
aws_instance.k8s_slave2: Still creating... [10s elapsed]
aws_instance.k8s_slave1: Creation complete after 13s [id=i-0936fa04acc084fee]
aws_instance.k8s_slave2: Creation complete after 13s [id=i-09c7ead9be60210ed]
aws_instance.k8s_master_jenkins_slave: Creation complete after 13s [id=i-0ea40f59c5481b8b1]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-83-59:~/terraformer$ |
```

applicable law.

To run a command as administrator (user "root"), use "sudo <command>".

See "man sudo\_root" for details.

```

ubuntu@ip-172-31-22-48:~$ sudo apt update -
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2547 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [414 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [18.5 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [3473 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [618 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [676 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1201 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [296 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [28.7 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [46.5 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [11.8 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [592 B]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [68.4 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [11.1 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [392 B]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [30.0 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.5 kB]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [672 B]
Get:30 http://us-east-1.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 [2306 kB]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [351 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [13.6 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [3353 kB]
92% [34 Packages 2311 kB/3353 kB 69%]
```

415 kB/s 6s



```

Jenkins-master/Ansible-maste X K8s-Master/Jenkins-Slave X K8s-Slave1 X K8s-Slave2 X
+ ephemeral_block_device (known after apply)
+ instance_market_options (known after apply)
+ maintenance_options (known after apply)
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (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.k8s_slave1: Creating...
aws_instance.k8s_master_jenkins_slave: Creating...
aws_instance.k8s_slave2: Creating...
aws_instance.k8s_slave1: Still creating... [10s elapsed]
aws_instance.k8s_master_jenkins_slave: Still creating... [10s elapsed]
aws_instance.k8s_slave2: Still creating... [10s elapsed]
aws_instance.k8s_slave1: Creation complete after 13s [id=i-0936fa04acc084fee]
aws_instance.k8s_slave2: Creation complete after 13s [id=i-09c7ead9be60210ed]
aws_instance.k8s_master_jenkins_slave: Creation complete after 13s [id=i-0ea40f59c5481b8b1]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-83-59:~/terrafolder$ cd
ubuntu@ip-172-31-83-59:~$ cd /ter
-bash: cd: /ter: No such file or directory
ubuntu@ip-172-31-83-59:~/terrafolder/
ubuntu@ip-172-31-83-59:~/terrafolder$ ls
main.tf  terraform.tfstate
ubuntu@ip-172-31-83-59:~/terrafolder$ |

```

### create ssh key pair for ansible connection

```

|oos..          |
| B++.. .       |
|**+=o o .S    |
|[E=+oB . . o  |
|+. =o= . + .  |
|...o + o+ .   |
|....+...=o     |
+---[SHA256]----+
ubuntu@ip-172-31-83-59:~/.ssh$ ls
authorized_keys  l  l.pub
ubuntu@ip-172-31-83-59:~/.ssh$ sudo nano l.pub
ubuntu@ip-172-31-83-59:~/.ssh$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ZdMHMj/Kku2tVorFVFn9jh4qxWdL1m06AbnQURFBgkNQ ubuntu@ip-172-31-83-59
The key's randomart image:
+---[RSA 3072]----+
 ...+o+ .+|.
 . E. B *.=+|.
 . * =*+. |.
 o =+=* o|.
 S + +o. o|.
 o =.+ oo|.
 .+ o|.
 .. o|.
+---[SHA256]----+
ubuntu@ip-172-31-83-59:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub  l  l.pub
ubuntu@ip-172-31-83-59:~/.ssh$ sudo cat id_rsa.pub
ssh-rsa AAAAB3nza1c1yc2EAAAQABgQC3Z+f4L2ziyE0+QTDgb7+31kqgE20y603lzwafMrbuKaGdE5qHLDH3W/bFd73JJd65yCTqoEHRBPo503xjl9rG1Mg9Rus9Bxdp0Y41TMjkpJol228HXNo
phxe7AqdGF0KE/mWR6DFYZvXw1WQ2yifjejyVszXlt+554+1dFJVkbSpiROp8kMgp05xCzxHohuV+SuvwbUWey8ZB+ASDyJF+fehgqE3jool/SyPArdnYkliaZUEpmCY8LBmp8KU/r4dOC3oSRLxsL
F7911MSEb00WJgq1528XAzMDf4QTHwhilwsZCyn9AOfwYhekrilij1xH5auq4ESfd7wK628W1RTjz8zm2i/RgJkoWuNLmwj5j1VScAyJe6mx8kfvtUKdPqBiqnWErb2DL98C6FvhjaU9aLm#48//1Y5Tx
KxgDwKh1pqAgcSF4OHr52vSzc/ktc7SH0lNmee1xwQ+Z8TwHuts/MqW4Tgf9ll7R4ujwKSNvtSsEnRTJcGCTk= ubuntu@ip-172-31-83-59
ubuntu@ip-172-31-83-59:~$ cd
ubuntu@ip-172-31-83-59:~$ |

```

```

Jenkins-master/Ansible-master  X K8s-Master/Jenkins-Slave  X K8s-Slave1  X K8s-Slave2  X
ubuntu@ip-172-31-22-48:~$ cd .ssh
ubuntu@ip-172-31-22-48:~/ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-22-48:~/ssh$ 

Jenkins-master/Ansible-master  X K8s-Master/Jenkins-Slave  X K8s-Slave1  X K8s-Slave2  X
Get:18 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [2306 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [296 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [28.7 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [46.5 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [11.8 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [592 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [68.4 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [11.1 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [392 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [30.0 kB]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.5 kB]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [672 B]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [351 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [13.6 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [3353 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [600 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [624 B]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [974 kB]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [210 kB]
Get:39 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [21.7 kB]
Get:40 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [39.6 kB]
Get:41 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [8716 B]
Get:42 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [368 B]
Fetched 37.4 MB in 6s (6787 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
65 packages can be upgraded. Run 'apt list --upgradable' to see them.

ubuntu@ip-172-31-24-64:~$ sudo cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAQADQcIGQIizxFUPw5ihJ4+tEMUuuRjJ3RWkpo4jj+gt+06gDWP2vyXBz7+j+7ECpkWVGISGUJvTiI681d/ivg/igjmdlyJTrMHAmo06shA78qCJOzaAUycWLW7
QbhjhdtVcBqP0zNYZB8rpKexJLpd0QGkxZ2A92+dix+Xg9spzQzF9/WBrqOBALhwoFOLBngBchJWjj5bngfu3Eh3+Ie2QlUwka6uSfne72MyzNtmM/S9eKXAjMbChqizaSu8wSDs3Hgm1er15tg1/nS3
il8asGWQuuX9C+etKgl+sjuPejalalyrVWe+G0kfV/MltTqDsuLX56F1wm9gYnneI NV-Keypair
ssh-rsa AAAAB3NzaC1yc2EAAQADQcIGQIizxFUPw5ihJ4+tEMUuuRjJ3RWkpo4jj+gt+06gDWP2vyXBz7+j+7ECpkWVGISGUJvTiI681d/ivg/igjmdlyJTrMHAmo06shA78qCJOzaAUycWLW7
QbhjhdtVcBqP0zNYZB8rpKexJLpd0QGkxZ2A92+dix+Xg9spzQzF9/WBrqOBALhwoFOLBngBchJWjj5bngfu3Eh3+Ie2QlUwka6uSfne72MyzNtmM/S9eKXAjMbChqizaSu8wSDs3Hgm1er15tg1/nS3
pMr+7AdhGF9xE/mWR6DF7YVv/M1WQH2iifjejyVszXLt+55H+1df3Vkb5PiR0P8MhGp85xCz2hohv/+5uvwbUle/8ZB+ASDYJF+fEHQgE3jool/SyAArduYLiAZUEpmCY8LBmep8VU/r4dOC3oSRLXsL
F791lMSEf000WJg0q1528ZXAzMDFa4QTWh1lWS2Lcyh9AOFWYhekr4iij1xH5auq4ESFd7Wk628W1RTj+8xm2i/RgJkoWuNLmmj5jiVSACayJed6mx8kfvtkDpQb1qnWerb2Dl98C6FvhjaU9aLm#40//1Y5TXe
KxgDWXhLpqAGcSF40HRS2vSrc/ktc7SHOLNme1xwQ+Z8TWHuts/MqW4Tgf9ll7R4ujwKSNvtSsEnFDcGCTk= ubuntu@ip-172-31-83-59
ubuntu@ip-172-31-24-64:~$ ssh$ 

```

```

Jenkins-master/Ansible-maste X K8s-Master/Jenkins-Slave X K8s-Slave1 X K8s-Slave2 X
ubuntu@ip-172-31-83-59:$ cd /etc/ansible/
ubuntu@ip-172-31-83-59:/etc/ansible$ ls
ansible.cfg hosts roles
ubuntu@ip-172-31-83-59:/etc/ansible$ sudo nano hosts
ubuntu@ip-172-31-83-59:/etc/ansible$ sudo nano ansible.cfg
ubuntu@ip-172-31-83-59:/etc/ansible$ ansible-config init --disabled > ansible.cfg
-bash: ansible.cfg: Permission denied
ubuntu@ip-172-31-83-59:/etc/ansible$ sudo ansible-config init --disabled > ansible.cfg
-bash: ansible.cfg: Permission denied
ubuntu@ip-172-31-83-59:/etc/ansible$ sudo ansible-config init --disabled | sudo tee ansible.cfg > /dev/null
ubuntu@ip-172-31-83-59:/etc/ansible$ sudo nano ansible.cfg
ubuntu@ip-172-31-83-59:/etc/ansible$ ansible -m ping all
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
The authenticity of host '172.31.22.48 (172.31.22.48)' can't be established.
ED25519 key fingerprint is SHA256:ymDGzOoguDd7TcZNmVb0CbqRbrbPBwEpejyVdittMw.
This key is not known by any other names
The authenticity of host '172.31.24.64 (172.31.24.64)' can't be established.
ED25519 key fingerprint is SHA256:549rD8jj6tMKes0ZNM2k8U85T7j4QR9YfhLsJndlXo.
This key is not known by any other names
The authenticity of host '172.31.23.96 (172.31.23.96)' can't be established.
ED25519 key fingerprint is SHA256:NRfFpvwgKE0YlGZ9eUpGuLjSu4Lg0TC18PqCt1QcLM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Please type 'yes', 'no' or the fingerprint: yes
[ERROR]: User interrupted execution
ubuntu@ip-172-31-83-59:/etc/ansible$ sudo nano ansible.cfg
ubuntu@ip-172-31-83-59:/etc/ansible$ ansible -m ping all
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
[WARNING]: Platform linux on host 172.31.23.96 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
172.31.23.96 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.10"
    },
    "changed": false,
    "ping": "pong"
}

Jenkins-master/Ansible-maste X K8s-Master/Jenkins-Slave X K8s-Slave1 X K8s-Slave2 X
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Please type 'yes', 'no' or the fingerprint: yes
[ERROR]: User interrupted execution
ubuntu@ip-172-31-83-59:/etc/ansible$ sudo nano ansible.cfg
ubuntu@ip-172-31-83-59:/etc/ansible$ ansible -m ping all
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
[WARNING]: Platform linux on host 172.31.23.96 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
172.31.23.96 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.10"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.22.48 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
172.31.22.48 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.10"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.24.64 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
172.31.24.64 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.10"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-83-59:/etc/ansible$

```

using ansible I am going to install kubernetes

```

Jenkins-master/Ansible-maste X K8S-Master/Jenkins-Slave X K8S-Slave1 X K8S-Slave2 X + -
Please type 'yes', 'no' or the fingerprint: yes
Please type 'yes', 'no' or the fingerprint: yes
Please type 'yes', 'no' or the fingerprint: yes
Please type 'yes', 'no' or the fingerprint: 'C [ERROR]: User interrupted execution
ubuntu@ip-172-31-83-59:/etc/ansible$ sudo nano ansible.cfg
ubuntu@ip-172-31-83-59:/etc/ansible$ ansible -m ping all
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
[WARNING]: Platform linux on host 172.31.23.96 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
172.31.23.96 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.10"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.22.48 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
172.31.22.48 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.10"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.24.64 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
172.31.24.64 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.10"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-83-59:/etc/ansible$ sudo nano play.yaml
ubuntu@ip-172-31-83-59:/etc/ansible$ sudo nano localhost.sh
ubuntu@ip-172-31-83-59:/etc/ansible$ sudo nano k8sinstallation.sh
ubuntu@ip-172-31-83-59:/etc/ansible$ ansible-playbook play.yaml --syntax-check

```

ENG IN 22:08 05-05-2025

```

Jenkins-master/Ansible-maste X K8S-Master/Jenkins-Slave X K8S-Slave1 X K8S-Slave2 X + -
ubuntu@ip-172-31-83-59:/etc/ansible$ sudo nano ansible.cfg
ubuntu@ip-172-31-83-59:/etc/ansible$ ansible -m ping all
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
[WARNING]: Platform linux on host 172.31.23.96 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
172.31.23.96 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.10"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.22.48 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
172.31.22.48 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.10"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.24.64 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
172.31.24.64 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.10"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-83-59:/etc/ansible$ sudo nano play.yaml
ubuntu@ip-172-31-83-59:/etc/ansible$ sudo nano localhost.sh
ubuntu@ip-172-31-83-59:/etc/ansible$ sudo nano k8sinstallation.sh
ubuntu@ip-172-31-83-59:/etc/ansible$ ansible-playbook play.yaml --syntax-check
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
playbook: play.yaml
ubuntu@ip-172-31-83-59:/etc/ansible$ |

```

ENG IN 23:18 05-05-2025

```
Jenkins-master/Ansible-maste X K8s-Master/Jenkins-Slave X K8s-Slave1 X K8s-Slave2 X + - X
PLAY [executing script for Enhanced Kubernetes Node Bootstrap Script] ****
TASK [Gathering Facts] ****
[WARNING]: Platform linux on host 172.31.22.48 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.22.48]

TASK [script] ****
skipping: [172.31.22.48]

PLAY [executing script for k8s installation on slave1] ****
TASK [Gathering Facts] ****
[WARNING]: Platform linux on host 172.31.24.64 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.24.64]

TASK [script] ****
skipping: [172.31.24.64]

PLAY [executing script for k8s installation on slaves] ****
TASK [Gathering Facts] ****
[WARNING]: Platform linux on host 172.31.23.96 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.23.96]

TASK [script] ****
skipping: [172.31.23.96]

PLAY RECAP ****
172.31.22.48 : ok=1    changed=0   unreachable=0 failed=0   skipped=1  rescued=0  ignored=0
172.31.23.96 : ok=1    changed=0   unreachable=0 failed=0   skipped=1  rescued=0  ignored=0
172.31.24.64 : 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-172-31-83-59:/etc/ansible$ |
```

```
Jenkins-master/Ansible-maste X K8s-Master/Jenkins-Slave X K8s-Slave1 X K8s-Slave2 X + - X
PLAY [executing script for Enhanced Kubernetes Node Bootstrap Script] ****
TASK [Gathering Facts] ****
[WARNING]: Platform linux on host 172.31.22.48 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.22.48]

TASK [script] ****
changed: [172.31.22.48]

PLAY [executing script for k8s installation on slave1] ****
TASK [Gathering Facts] ****
[WARNING]: Platform linux on host 172.31.24.64 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.24.64]

TASK [script] ****
changed: [172.31.24.64]

PLAY [executing script for k8s installation on slaves] ****
TASK [Gathering Facts] ****
[WARNING]: Platform linux on host 172.31.23.96 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.23.96]

TASK [script] ****
changed: [172.31.23.96]

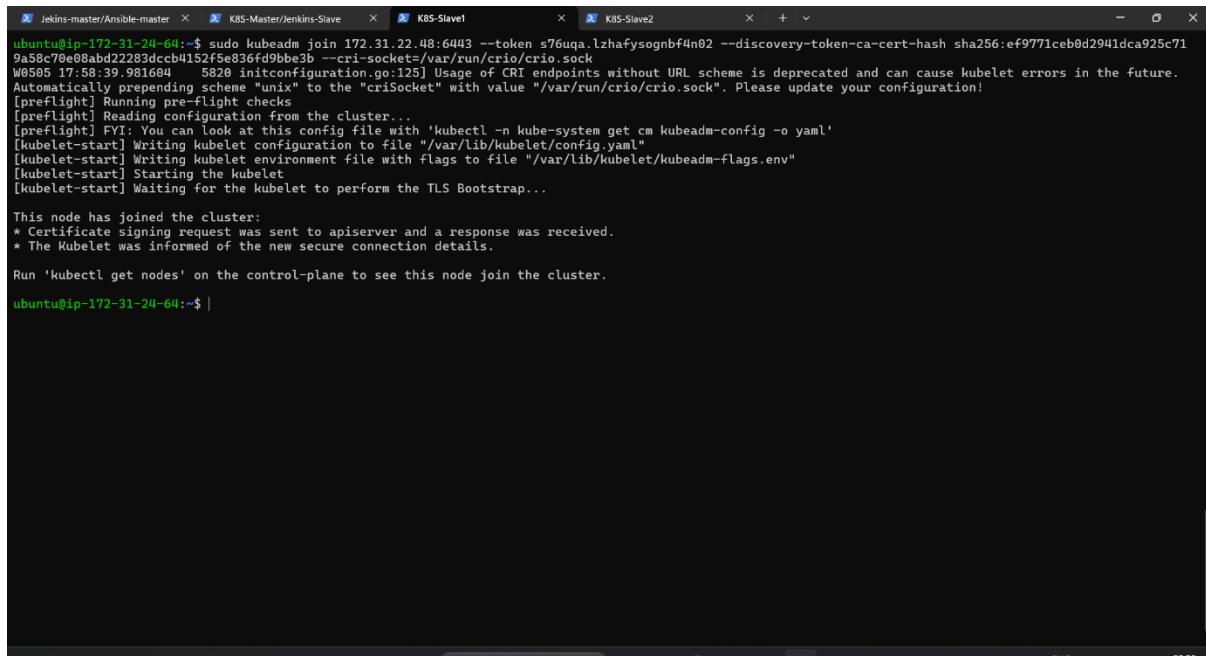
PLAY RECAP ****
172.31.22.48 : ok=2    changed=1   unreachable=0 failed=0   skipped=0  rescued=0  ignored=0
172.31.23.96 : ok=2    changed=1   unreachable=0 failed=0   skipped=0  rescued=0  ignored=0
172.31.24.64 : ok=2    changed=1   unreachable=0 failed=0   skipped=0  rescued=0  ignored=0
localhost     : ok=2    changed=1   unreachable=0 failed=0   skipped=0  rescued=0  ignored=0
ubuntu@ip-172-31-83-59:/etc/ansible$ |
```

```

Jenkins-master/Ansible-master K8S-Master/Jenkins-Slave K8S-Slave1 K8S-Slave2
ubuntu@ip-172-31-22-48:~$ sudo kubeadm init --cri-socket=/var/run/crio/crio.sock
W0505 17:54:37.064009      5669 initconfiguration.go:125] Usage of CRI endpoints without URL scheme is deprecated and can cause kubelet errors in the future.
Automatically prepending scheme "unix" to the "criSocket" with value "/var/run/crio/crio.sock". Please update your configuration!
I0505 17:54:37.211109      5669 version.go:256] remote version is much newer: v1.33.0; falling back to: stable-1.29
[init] Using Kubernetes version: v1.29.15
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
W0505 17:54:47.995658      5669 checks.go:835] detected that the sandbox image "registry.k8s.io/pause:3.10" of the container runtime is inconsistent with that used by kubeadm. It is recommended that using "registry.k8s.io/pause:3.9" as the CRI sandbox image.
[certs] Using certificatebin folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-22-48 kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 172.31.22.48]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [ip-172-31-22-48 localhost] and IPs [172.31.22.48 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [ip-172-31-22-48 localhost] and IPs [172.31.22.48 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "super-admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
[kubeconfig] Writing "scheduler.conf" kubeconfig file
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
[control-plane] Using manifest folder "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for "kube-apiserver"
[control-plane] Creating static Pod manifest for "kube-controller-manager"
[control-plane] Creating static Pod manifest for "kube-scheduler"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Starting the kubelet

Jenkins-master/Ansible-master K8S-Master/Jenkins-Slave K8S-Slave1 K8S-Slave2
serviceaccount/calico-node created
serviceaccount/calico-cni-plugin created
configmap/calico-config created
customresourcedefinition.apilextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apilextensions.k8s.io/bgpfilters.crd.projectcalico.org created
customresourcedefinition.apilextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apilextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apilextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apilextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apilextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apilextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apilextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apilextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apilextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apilextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apilextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apilextensions.k8s.io/irpervations.crd.projectcalico.org created
customresourcedefinition.apilextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apilextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apilextensions.k8s.io/networksets.crd.projectcalico.org created
customresourcedefinition.apilextensions.k8s.io/stagedglobalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apilextensions.k8s.io/stagedkubernetesnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apilextensions.k8s.io/tiers.crd.projectcalico.org created
customresourcedefinition.apilextensions.k8s.io/adminnetworkpolicies.policy.networking.k8s.io created
customresourcedefinition.apilextensions.k8s.io/baselineadminnetworkpolicies.policy.networking.k8s.io created
clusterrole:rba.authorization.k8s.io/calico-kube-controllers created
clusterrole:rba.authorization.k8s.io/calico-node created
clusterrole:rba.authorization.k8s.io/calico-cni-plugin created
clusterrole:rba.authorization.k8s.io/calico-tier-getter created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-cni-plugin created
clusterrolebinding.rbac.authorization.k8s.io/calico-tier-getter created
daemonset.apps/calico-node created
deployment.apps/calico-hub-controllers created
ubuntu@ip-172-31-22-48:~$ kubeadm token create --print-join-command
KubeADM join 172.31.22.48:6443 --token s76uqa.lzhafysognbfb4n02 --discovery-token-ca-cert-hash sha256:ef9771ceb0d2941dca925c719a58c70e08abd22283dccb4152f5e83
6fd9bbe3b
ubuntu@ip-172-31-22-48:~|

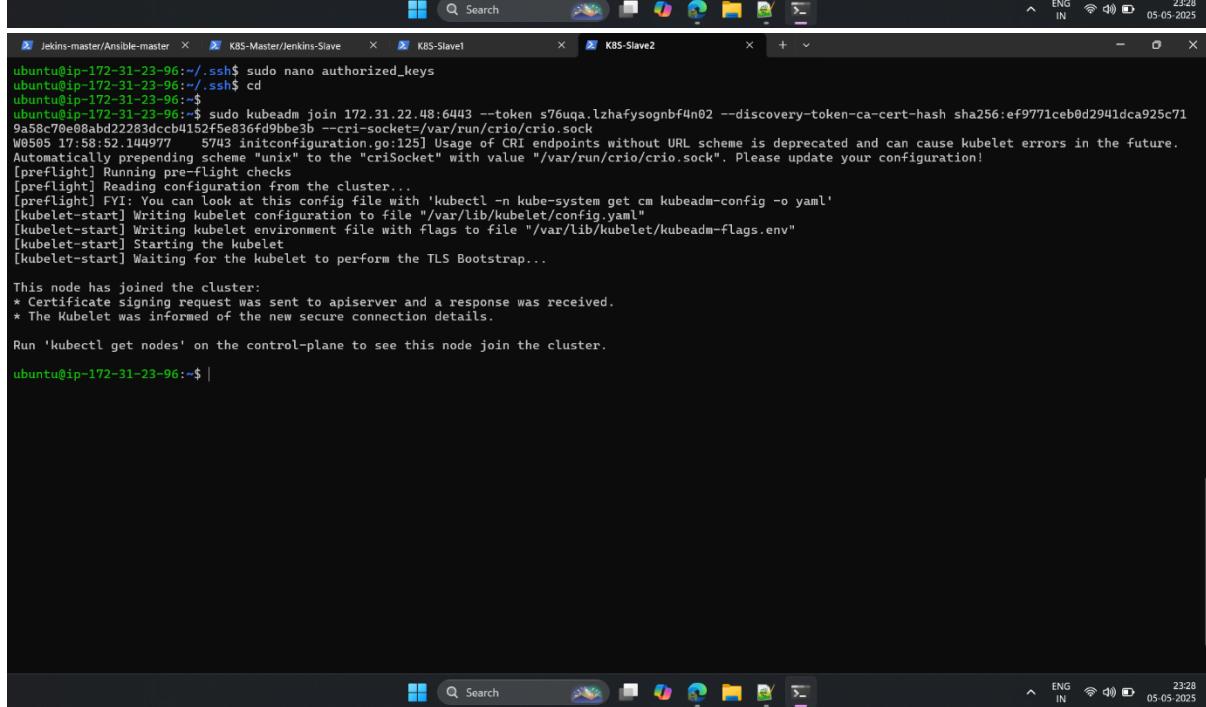
```



```
ubuntu@ip-172-31-24-64:~$ sudo kubeadm join 172.31.22.48:6443 --token s76uqa.lzhafysognbf4n02 --discovery-token-ca-cert-hash sha256:ef9771ceb0d2941dca925c71
9a58c70e08abd22283dc4152f5e836fd9bbe3b --cri-socket=/var/run/crio/crio.sock
W0505 17:58:39.981604      5820 initconfiguration.go:125] Usage of CRI endpoints without URL scheme is deprecated and can cause kubelet errors in the future.
Automatically prepending scheme "unix" to the "criSocket" with value "/var/run/crio/crio.sock". Please update your configuration!
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
ubuntu@ip-172-31-24-64:~$ |
```



```
ubuntu@ip-172-31-23-96:~/.ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-23-96:~/.ssh$ cd
ubuntu@ip-172-31-23-96:~$ sudo kubeadm join 172.31.22.48:6443 --token s76uqa.lzhafysognbf4n02 --discovery-token-ca-cert-hash sha256:ef9771ceb0d2941dca925c71
9a58c70e08abd22283dc4152f5e836fd9bbe3b --cri-socket=/var/run/crio/crio.sock
W0505 17:58:52.144977      5743 initconfiguration.go:125] Usage of CRI endpoints without URL scheme is deprecated and can cause kubelet errors in the future.
Automatically prepending scheme "unix" to the "criSocket" with value "/var/run/crio/crio.sock". Please update your configuration!
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
ubuntu@ip-172-31-23-96:~$ |
```

```

Jenkins-master/Ansible-master K8s-Master/Jenkins-Slave K8s-Slave1 K8s-Slave2
customresourcedefinition.apixtensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/irpresaevations.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/networksets.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/stagedglobalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/stagedkubernetesnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/tiers.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/adminnetworkpolicies.policy.networking.k8s.io created
customresourcedefinition.apixtensions.k8s.io/baselineadminnetworkpolicies.policy.networking.k8s.io created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrole.rbac.authorization.k8s.io/calico-cni-plugin created
clusterrole.rbac.authorization.k8s.io/calico-tier-getter created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-cni-plugin created
clusterrolebinding.rbac.authorization.k8s.io/calico-tier-getter created
daemonset.apps/calico-node created
deployment.apps/calico-kube-controllers created
ubuntu@ip-172-31-22-48:~$ kubeadm token create --print-join-command
kubeadm join 172.31.22.48:6443 --token s76qua.lzhafysognbf4n82 --discovery-token-ca-cert-hash sha256:ef9771ceb0d2941dca925c719a58c70e08abd22283dccb4152f5e83
6fd9bbe3b
ubuntu@ip-172-31-22-48:~$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-22-48 Ready control-plane 4m25s v1.29.0
ip-172-31-23-96 Ready <none> 31s v1.29.0
ip-172-31-24-64 Ready <none> 44s v1.29.0
ubuntu@ip-172-31-22-48:~$ |

```

## install docker in all machines

```

Jenkins-master/Ansible-master K8s-Master/Jenkins-Slave K8s-Slave1 K8s-Slave2
customresourcedefinition.apixtensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/irpresaevations.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/networksets.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/stagedglobalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/stagedkubernetesnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/tiers.crd.projectcalico.org created
customresourcedefinition.apixtensions.k8s.io/adminnetworkpolicies.policy.networking.k8s.io created
customresourcedefinition.apixtensions.k8s.io/baselineadminnetworkpolicies.policy.networking.k8s.io created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrole.rbac.authorization.k8s.io/calico-cni-plugin created
clusterrole.rbac.authorization.k8s.io/calico-tier-getter created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-cni-plugin created
clusterrolebinding.rbac.authorization.k8s.io/calico-tier-getter created
daemonset.apps/calico-node created
deployment.apps/calico-kube-controllers created
ubuntu@ip-172-31-22-48:~$ kubeadm token create --print-join-command
kubeadm join 172.31.22.48:6443 --token s76qua.lzhafysognbf4n82 --discovery-token-ca-cert-hash sha256:ef9771ceb0d2941dca925c719a58c70e08abd22283dccb4152f5e83
6fd9bbe3b
ubuntu@ip-172-31-22-48:~$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-22-48 Ready control-plane 4m25s v1.29.0
ip-172-31-23-96 Ready <none> 31s v1.29.0
ip-172-31-24-64 Ready <none> 44s v1.29.0
ubuntu@ip-172-31-22-48:~$ sudo apt-get install docker.io -y

```

```
Jenkins-master/Ansible-master x K8s-Master/Jenkins-Slave x K8s-Slave1 x K8s-Slave2 x + - Jenkins Master Node  
customresourcedefinition.apixensions.k8s.io/globalnetworksets.crd.projectcalico.org created  
customresourcedefinition.apixensions.k8s.io/hostendpoints.crd.projectcalico.org created  
customresourcedefinition.apixensions.k8s.io/ipamblocks.crd.projectcalico.org created  
customresourcedefinition.apixensions.k8s.io/ipamconfigs.crd.projectcalico.org created  
customresourcedefinition.apixensions.k8s.io/ipamhandles.crd.projectcalico.org created  
customresourcedefinition.apixensions.k8s.io/ippools.crd.projectcalico.org created  
customresourcedefinition.apixensions.k8s.io/ipservations.crd.projectcalico.org created  
customresourcedefinition.apixensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created  
customresourcedefinition.apixensions.k8s.io/networkpolicies.crd.projectcalico.org created  
customresourcedefinition.apixensions.k8s.io/stagedglobalnetworkpolicies.crd.projectcalico.org created  
customresourcedefinition.apixensions.k8s.io/stagedkubernetesnetworkpolicies.crd.projectcalico.org created  
customresourcedefinition.apixensions.k8s.io/stagednetworkpolicies.crd.projectcalico.org created  
customresourcedefinition.apixensions.k8s.io/tiers.crd.projectcalico.org created  
customresourcedefinition.apixensions.k8s.io/adminnetworkpolicies.policy.networking.k8s.io created  
customresourcedefinition.apixensions.k8s.io/baselineadminnetworkpolicies.policy.networking.k8s.io created  
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created  
clusterrole.rbac.authorization.k8s.io/calico-node created  
clusterrole.rbac.authorization.k8s.io/calico-cni-plugin created  
clusterrole.rbac.authorization.k8s.io/calico-tier-getter created  
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created  
clusterrolebinding.rbac.authorization.k8s.io/calico-node created  
clusterrolebinding.rbac.authorization.k8s.io/calico-cni-plugin created  
clusterrolebinding.rbac.authorization.k8s.io/calico-tier-getter created  
daemonset.apps/calico-node created  
deployment.apps/calico-kube-controllers created  
ubuntu@ip-172-31-22-48:~$ kubeadm token create --print-join-command  
kubeadm join 172.31.22.48:6443 --token s76uqa.lzhafysognbf4n02 --discovery-token-ca-cert-hash sha256:ef9771ceb0d2941dca925c719a58c70e08abd22283dccb4152f5e83  
6fd9bbe3b  
ubuntu@ip-172-31-22-48:~$ kubectl get nodes  
NAME STATUS ROLES AGE VERSION  
ip-172-31-22-48 Ready control-plane 4m25s v1.29.0  
ip-172-31-23-96 Ready <none> 31s v1.29.0  
ip-172-31-24-64 Ready <none> 44s v1.29.0  
ubuntu@ip-172-31-22-48:~$ sudo apt-get install docker.io -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
docker.io is already the newest version (26.1.3-0ubuntu1~22.04.1).  
0 upgraded, 0 newly installed, 0 to remove and 57 not upgraded.  
ubuntu@ip-172-31-22-48:~$ |
```

```
Jenkins-master/Ansible-master x K8s-Master/Jenkins-Slave x K8s-Slave1 x K8s-Slave2 x + - Jenkins Master Node  
ubuntu@ip-172-31-22-48:~$ sudo kubeadm join 172.31.22.48:6443 --token s76uqa.lzhafysognbf4n02 --discovery-token-ca-cert-hash sha256:ef9771ceb0d2941dca925c719a58c70e08abd22283dccb4152f5e83  
9a5c70e08abd22283dccb4152f5e836fd9bbe3b --cri-socket=/var/run/crio/crio.sock  
W0505 17:58:39.981604      5820 initconfiguration.go:125] Usage of CRI endpoints without URL scheme is deprecated and can cause kubelet errors in the future.  
Automatically prepending scheme "unix" to the "criSocket" with value "/var/run/crio/crio.sock". Please update your configuration!  
[preflight] Running pre-flight checks  
[preflight] Reading configuration from the cluster...  
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'  
[kubebundle-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"  
[kubebundle-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"  
[kubebundle-start] Starting the kubelet  
[kubebundle-start] Waiting for the kubelet to perform the TLS Bootstrap...  
  
This node has joined the cluster:  
* Certificate signing request was sent to apiserver and a response was received.  
* The Kubelet was informed of the new secure connection details.  
  
Run 'kubectl get nodes' on the control-plane to see this node join the cluster.  
ubuntu@ip-172-31-24-64:~$ sudo apt-get install docker.io -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
docker.io is already the newest version (26.1.3-0ubuntu1~22.04.1).  
0 upgraded, 0 newly installed, 0 to remove and 57 not upgraded.  
ubuntu@ip-172-31-24-64:~$ |
```

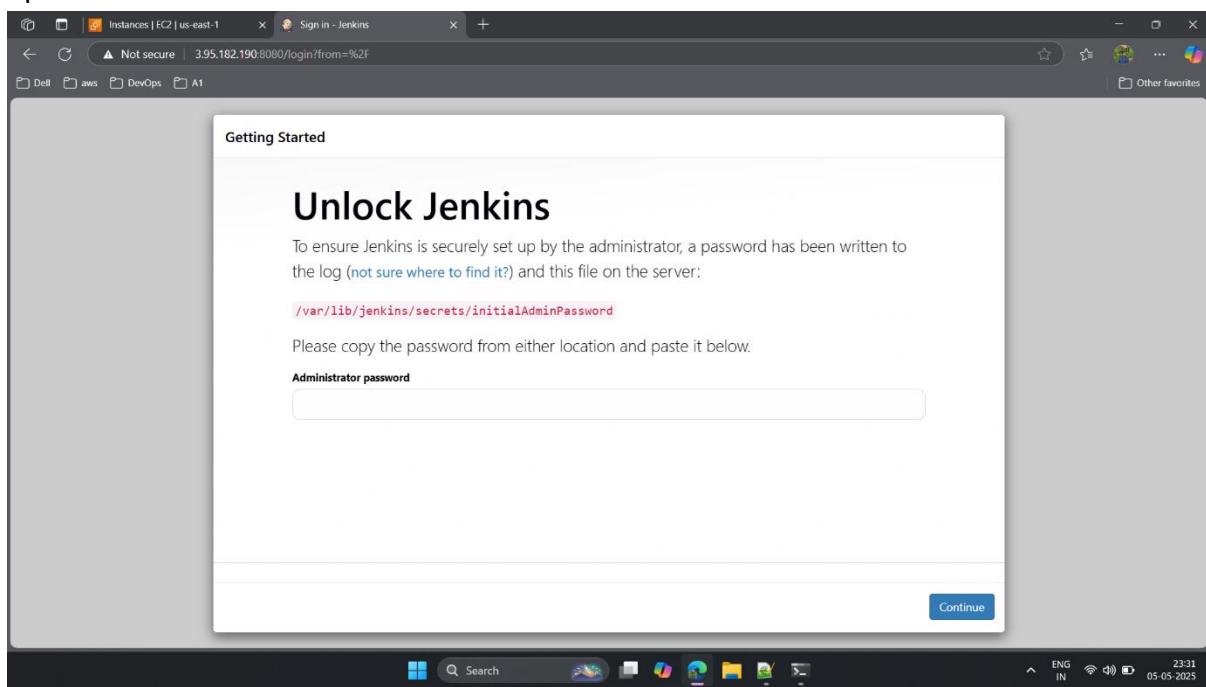
```
Jenkins-master/Ansible-master  X  K8s-Master/Jenkins-Slave  X  K8s-Slave1  X  K8s-Slave2  X  +  ~
ubuntu@ip-172-31-23-96:~$ sudo kubeadm join 172.31.22.48:6443 --token s76uqa.lzhafysognbf4n02 --discovery-token-ca-cert-hash sha256:ef9771ceb0d2941dca925c71
9a58c70e08abd22283dc4152f5e836fd9bbe3b --cri-socket=/var/run/crio/crio.sock
W0505 17:58:52.144977      5743 initconfig.go:125] Usage of CRI endpoints without URL scheme is deprecated and can cause kubelet errors in the future.
Automatically prepending scheme "unix" to the "criSocket" with value "/var/run/crio/crio.sock". Please update your configuration!
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

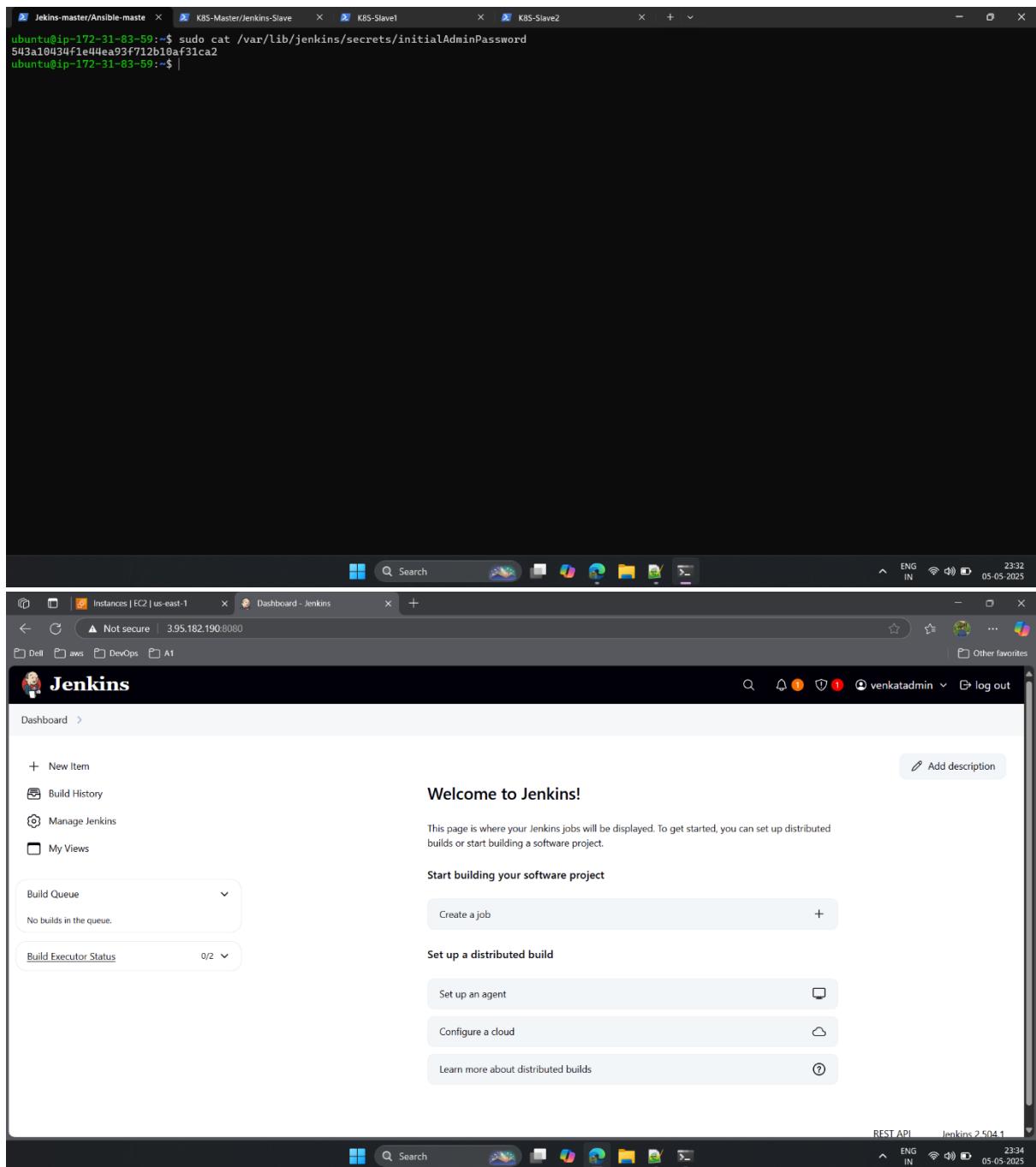
This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

ubuntu@ip-172-31-23-96:~$ sudo apt-get install docker.io -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker.io is already the newest version (26.1.3-0ubuntu1-22.04.1).
0 upgraded, 0 newly installed, 0 to remove and 57 not upgraded.
ubuntu@ip-172-31-23-96:~$ |
```

open Jenkins dashboard





giving credentials

The image shows two screenshots of the Jenkins web interface, likely from a Windows desktop environment.

**Screenshot 1: New credentials creation**

This screenshot shows the 'New credentials' page under 'Manage Jenkins > Credentials > System > Global credentials (unrestricted)'. The 'Kind' dropdown is set to 'Username with password'. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc.)'. The 'Username' field contains 'venkatesh1503'. A checkbox labeled 'Treat username as secret' is unchecked. The 'Password' field is empty. The 'ID' field is also empty. A blue 'Create' button is at the bottom.

**Screenshot 2: Global credentials (unrestricted) list**

This screenshot shows the 'Global credentials (unrestricted)' page. It lists two credentials:

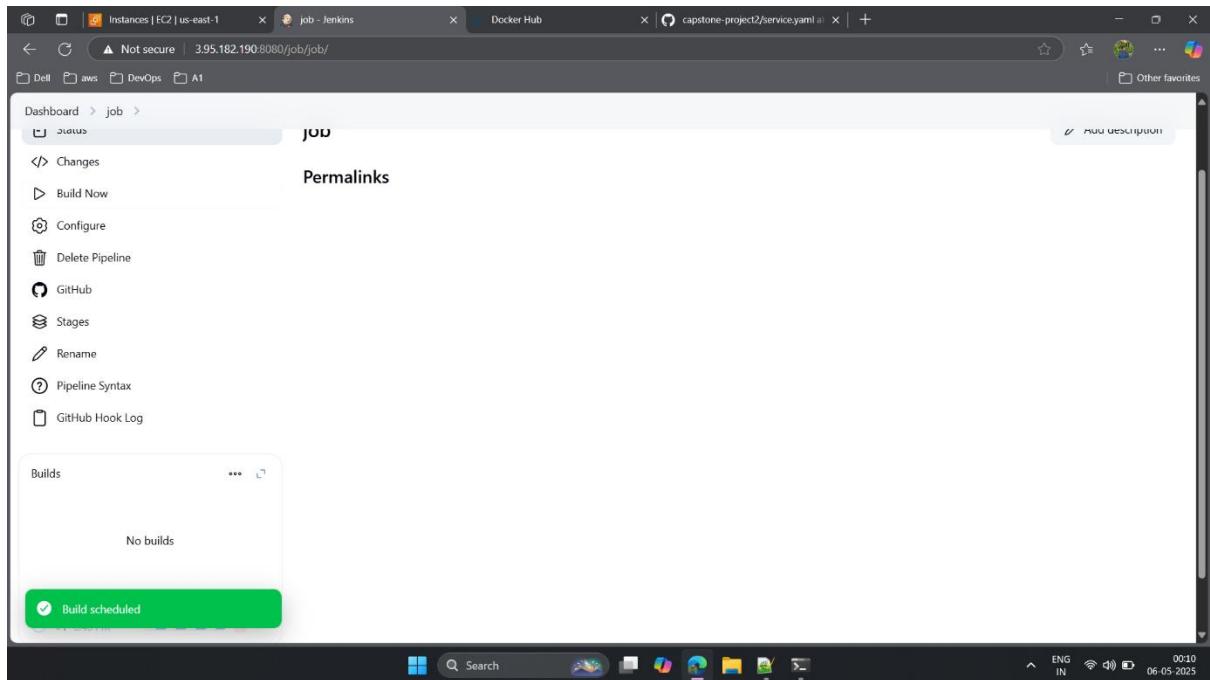
ID	Name	Kind	Description
4a13c60a-a9d8-495f-ae39-cd2a678a5be0	ubuntu	SSH Username with private key	
b2bba9da-626d-4dca-bdf2-a7b3a58e2a0d	venkatesh1503/*****	Username with password	

At the top right of this page is a blue '+ Add Credentials' button. The bottom right corner of the interface displays 'REST API Jenkins 2.504.1'.

The image shows two screenshots of a Jenkins interface. The top screenshot displays the 'Global credentials (unrestricted)' page, listing two entries: 'ubuntu' (SSH Username with private key) and 'venkatesh1503\*\*\*\*\*' (Username with password). The bottom screenshot shows the 'Configuration' page for a Jenkins job, specifically the 'Pipeline' tab. The pipeline script is defined as follows:

```
steps {
    script {
        git 'https://github.com/venkatesh1715/capstone-project2.git'
    }
}
stage('docker') {
    agent {
        label "K8S-Master-Jenkins-Slave"
    }
    steps {
        script {
            // git 'https://github.com/venkatesh1715/capstone-project2.git'
            sh "sudo docker build . -t venkatesh1503/capstone-project2"
            sh "sudo docker login -u ${DOCKERHUB_CREDENTIALS_USR} -p ${DOCKERHUB_CREDENTIALS_PSW}"
        }
    }
}
```

The Jenkins interface includes standard browser navigation bars at the top and bottom.



## troubleshooting steps

```

Jenkins-master//Ansible-master x K8s-Master/Jenkins-Slave x K8s-Slave1 x K8s-Slave2 x + -
Setting up humanity-icon-theme (0.6.16) ...
Setting up ubuntu-mono (20.10-0ubuntu2) ...
Processing triggers for man-db (2.1.0-2-1) ...
Processing triggers for libglib2.0-0:amd64 (2.72.4-0ubuntu2.4) ...
Setting up libgtk-3-0:amd64 (3.24.33-1ubuntu2.2) ...
Processing triggers for libc-bin (2.35-0ubuntu3.9) ...
Setting up libgtk-3-bin (3.24.33-1ubuntu2.2) ...
Setting up openjdk-21-jre:amd64 (21.0.6+7-1~22.04.1) ...
Setting up openjdk-21-jdk:amd64 (21.0.6+7-1~22.04.1) ...
update-alternatives: using /usr/lib/jvm/java-21-openjdk-amd64/bin/jconsole to provide /usr/bin/jconsole (jconsole) in auto mode
Processing triggers for libgdk-pixbuf2.0-0:amd64 (2.42.8+dfsg-1ubuntu0.3) ...
Scanning processes...
Scanning linux images...

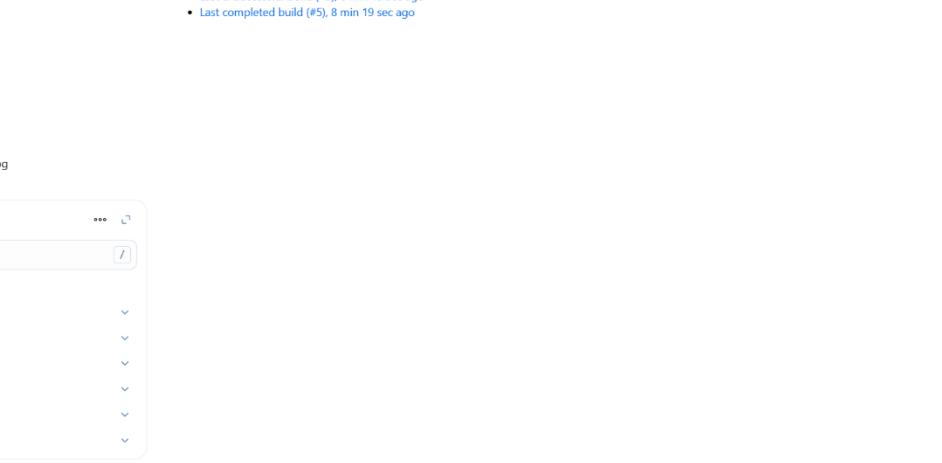
Running kernel seems to be up-to-date.

No services need to be restarted.

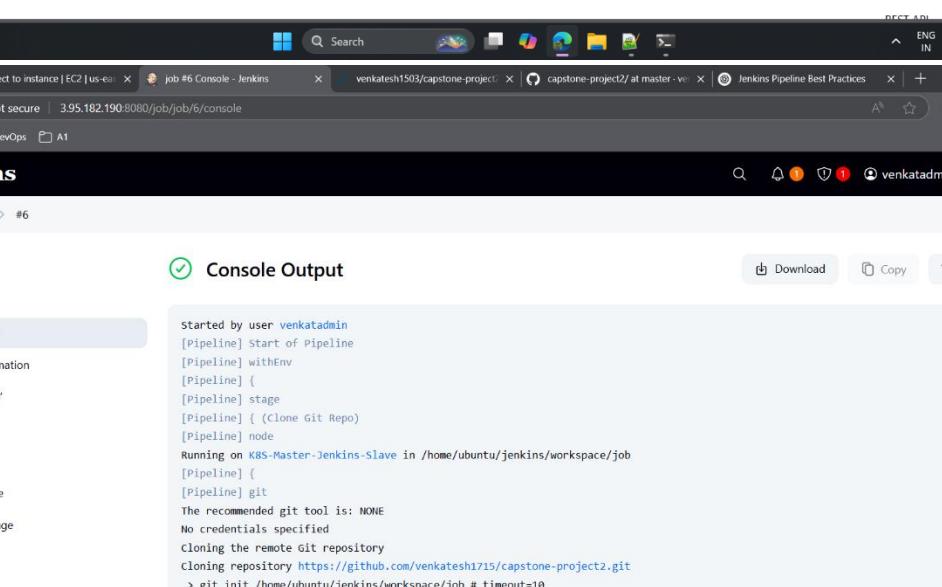
No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-22-48:~$ sudo usermod -G docker jenkins
usermod: user 'jenkins' does not exist
ubuntu@ip-172-31-22-48:~$ sudo groupadd docker
groupadd: group 'docker' already exists
ubuntu@ip-172-31-22-48:~$ sudo usermod -G docker jenkins
usermod: user 'jenkins' does not exist
ubuntu@ip-172-31-22-48:~$ ps aux | grep jenkins
ubuntu   18760  1.0  4.1 3638076 164348 ?  Ssl 18:23  0:20 java -jar remotng.jar -workDir /home/ubuntu/jenkins -jar-cache /home/ubuntu/jenkins/remotng/jarCache
ubuntu   27855  0.0  0.0   7008 2304 pts/0  S+ 18:56  0:00 grep --color=auto jenkins
ubuntu@ip-172-31-22-48:~$ sudo systemctl status jenkins
Unit jenkins.service could not be found.
ubuntu@ip-172-31-22-48:~$ sudo usermod -G docker ubuntu
ubuntu@ip-172-31-22-48:~$ sudo reboot
Connection to ec2-54-90-72-122.compute-1.amazonaws.com closed by remote host.
Connection to ec2-54-90-72-122.compute-1.amazonaws.com closed.
PS C:\Users\svatj\Downloads> ssh -i "NV-Keypair.pem" ubuntu@ec2-54-90-72-122.compute-1.amazonaws.com
|
```



The screenshot shows the Jenkins Pipeline Best Practices dashboard. On the left, there's a sidebar with options like Delete Pipeline, GitHub, Stages, Rename, Pipeline Syntax, and GitHub Hook Log. The main area displays a 'Builds' section with a filter input. A list of builds is shown, with build #6 being the last successful one at 7:00 PM. Below this is a 'Console Output' section for build #6, which shows the Jenkins pipeline script and its execution logs. The logs indicate the pipeline was started by user 'venkatadmin', used 'Pipeline' and 'withEnv' steps, and cloned a repository from 'https://github.com/venkatesh1715/capstone-project2.git'. It also shows the execution environment as 'Running on K8S-Master-Jenkins-Slave'.



The screenshot shows two windows side-by-side. The left window is a Jenkins job console output for build #6, showing the pipeline stages and command logs. The right window is a Docker Hub repository page for 'venkatesh1503/capstone-project2/general', showing tags and Docker commands.

**Jenkins Console Output:**

```
Started by user venkatadmin
[Pipeline] Start of Pipeline
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Clone Git Repo)
[Pipeline] node
[Pipeline] git
Running on K8S-Master-Jenkins-Slave in /home/ubuntu/jenkins/workspace/job
[Pipeline] {
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
cloning repository https://github.com/venkatesh1715/capstone-project2.git
> git init /home/ubuntu/jenkins/workspace/job # timeout=10
Fetching upstream changes from https://github.com/venkatesh1715/capstone-project2.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
> git fetch --tags --force --progress -- https://github.com/venkatesh1715/capstone-project2.git +refs/heads/*:refs/remotes/origin/*
timeout=10
> git config remote.origin.url https://github.com/venkatesh1715/capstone-project2.git # timeout=10
```

**Docker Hub Repository:**

venkatesh1503/capstone-project2

Tags:

Tag	OS	Type	Pulled	Pushed
6	Image		less than 1 day	2 minutes

Docker commands:

```
docker push venkatesh1503/capstone-project2:tagname
```

Build with Docker Build Cloud

Accelerate image build times with access to cloud-based builders and shared cache.

Docker Build Cloud executes builds on optimally-dimensioned cloud infrastructure with dedicated per-organization isolation.

Get faster builds through shared caching across your team, native multi-platform support, and encrypted data transfer - all

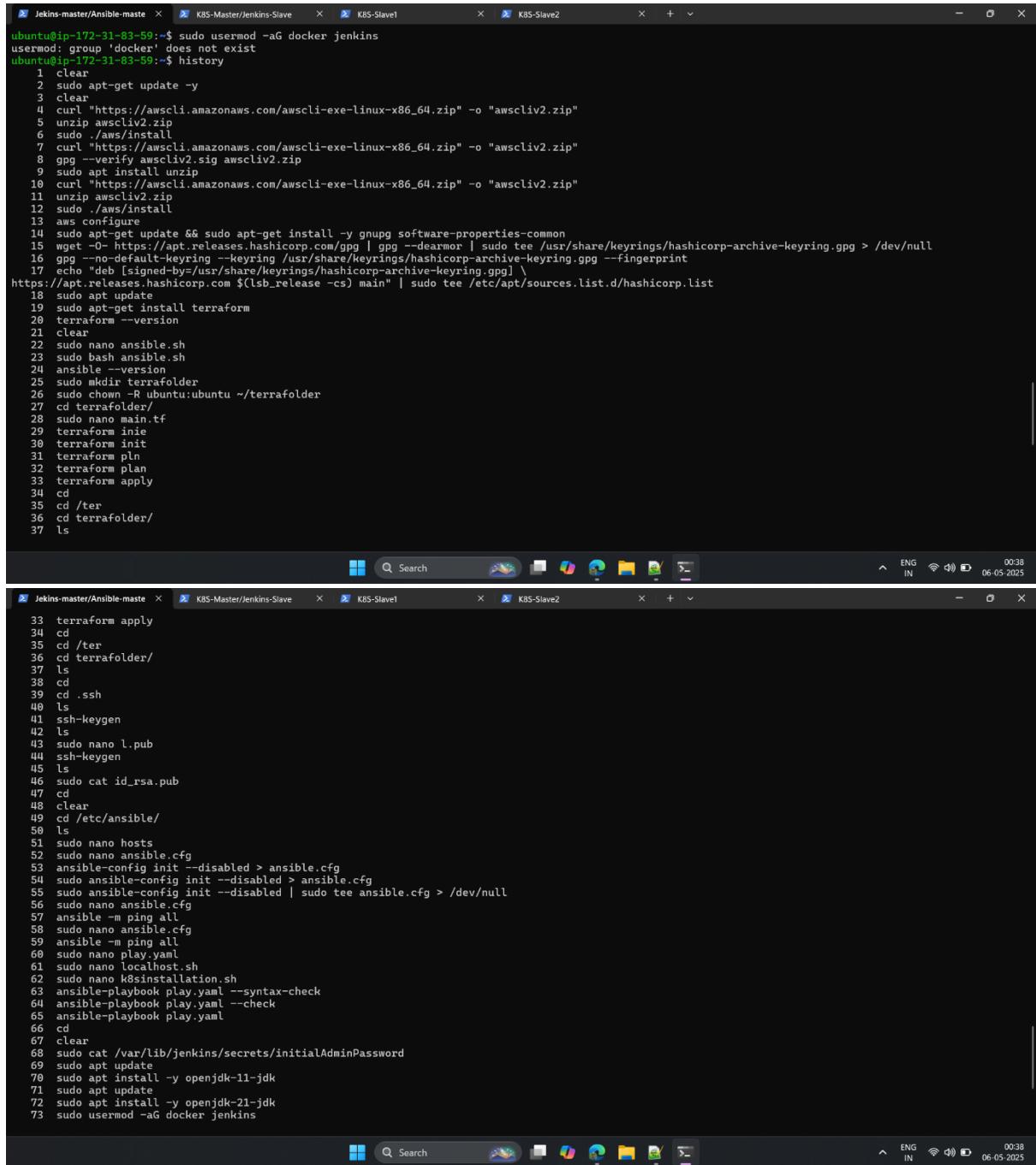
The image displays a Windows desktop environment with two open browser windows.

The top browser window shows the Jenkins job #6 console output. The log output is as follows:

```
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] node
Running on K8S-Master-Jenkins-Slave in /home/ubuntu/jenkins/workspace/job
[Pipeline] {
[Pipeline] cleanWs
[WS-CLEANUP] Deleting project workspace...
[WS-CLEANUP] Deferred wipeout is used...
[WS-CLEANUP] done
[Pipeline] }
[Pipeline] // node
[Pipeline] echo
Pipeline completed successfully.
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] End of Pipeline
Finished: SUCCESS
```

The bottom browser window shows a GitHub page with the text "Hello world!" and the GitHub logo.

## Used commands:



The screenshot shows a terminal window with multiple tabs open, displaying a history of Ansible commands. The commands are numbered from 1 to 73. The terminal interface includes a top bar with tabs for Jenkins-master/Ansible-maste, K8S-Master/Jenkins-Slave, K8S-Slave1, K8S-Slave2, and K8S-Slave3. Below the tabs is a scrollable command history area. At the bottom of the window is a Windows-style taskbar with icons for various applications like File Explorer, Task View, and Start.

```
ubuntu@ip-172-31-83-59:~$ sudo usermod -aG docker jenkins
usermod: group 'docker' does not exist
ubuntu@ip-172-31-83-59:~$ history
  1  clear
  2  sudo apt-get update -y
  3  clear
  4  curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
  5  unzip awscliv2.zip
  6  sudo ./aws/install
  7  curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
  8  gpg --verify awscliv2.sig awscliv2.zip
  9  sudo apt install unzip
 10 curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
 11 unzip awscliv2.zip
 12 sudo ./aws/install
 13 aws configure
 14 sudo apt-get update && sudo apt-get install -y gnupg software-properties-common
 15 wget -O https://apt.releases.hashicorp.com/gpg | gpg --dearmor | sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg > /dev/null
 16 gpg --no-default-keyring --keyring /usr/share/keyrings/hashicorp-archive-keyring.gpg --fingerprint
 17 echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] "
https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list
 18 sudo apt update
 19 sudo apt-get install terraform
 20 terraform --version
 21 clear
 22 sudo nano ansible.sh
 23 sudo bash ansible.sh
 24 ansible --version
 25 sudo mkdir terrafolder
 26 sudo chown -R ubuntu:ubuntu ~/terrafolder
 27 cd terrafolder/
 28 sudo nano main.tf
 29 terraform init
 30 terraform init
 31 terraform plan
 32 terraform plan
 33 terraform apply
 34 cd
 35 cd /ter
 36 cd terrafolder/
 37 ls
 38 cd
 39 cd .ssh
 40 ls
 41 ssh-keygen
 42 ls
 43 sudo nano l.pub
 44 ssh-keygen
 45 ls
 46 sudo cat id_rsa.pub
 47 cd
 48 clear
 49 cd /etc/ansible/
 50 ls
 51 sudo nano hosts
 52 sudo nano ansible.cfg
 53 ansible-config init --disabled > ansible.cfg
 54 sudo ansible-config init --disabled > ansible.cfg
 55 sudo ansible-config init --disabled | sudo tee ansible.cfg > /dev/null
 56 sudo nano ansible.cfg
 57 ansible -m ping all
 58 sudo nano ansible.cfg
 59 ansible -m ping all
 60 sudo nano play.yaml
 61 sudo nano localhost.sh
 62 sudo nano k8sinstallation.sh
 63 ansible-playbook play.yaml --syntax-check
 64 ansible-playbook play.yaml --check
 65 ansible-playbook play.yaml
 66 cd
 67 clear
 68 sudo cat /var/lib/jenkins/secrets/initialAdminPassword
 69 sudo apt update
 70 sudo apt install -y openjdk-11-jdk
 71 sudo apt update
 72 sudo apt install -y openjdk-21-jdk
 73 sudo usermod -aG docker jenkins
```

```

42 ls
43 sudo nano l.pub
44 ssh-keygen
45 ls
46 sudo cat id_rsa.pub
47 cd
48 clear
49 cd /etc/ansible/
50 ls
51 sudo nano hosts
52 sudo nano ansible.cfg
53 ansible--config init --disabled > ansible.cfg
54 sudo ansible--config init --disabled > ansible.cfg
55 sudo ansible--config init --disabled | sudo tee ansible.cfg > /dev/null
56 sudo nano ansible.cfg
57 ansible -m ping all
58 sudo nano ansible.cfg
59 ansible -m ping all
60 sudo nano play.yaml
61 sudo nano localhost.sh
62 sudo nano k8sinstallation.sh
63 ansible-playbook play.yaml --syntax-check
64 ansible-playbook play.yaml --check
65 ansible-playbook play.yaml
66 cd
67 clear
68 sudo cat /var/lib/jenkins/secrets/initialAdminPassword
69 sudo apt update
70 sudo apt install -y openjdk-11-jdk
71 sudo apt update
72 sudo apt install -y openjdk-21-jdk
73 sudo usermod -aG docker jenkins
74 history
ubuntu@ip-172-31-83-59:~$ java --version
openjdk 21.0.6 2025-01-21
OpenJDK Runtime Environment (build 21.0.6+7-Ubuntu-122.04.1)
OpenJDK 64-Bit Server VM (build 21.0.6+7-Ubuntu-122.04.1, mixed mode, sharing)
ubuntu@ip-172-31-83-59:~$ jenkins --version
2.594.1
ubuntu@ip-172-31-83-59:~$ terraform --version
Terraform v1.11.4

-----[Windows Terminal]-----
ubuntu@ip-172-31-22-48:~$ cd /home/ubuntu/jenkins/workspace
ubuntu@ip-172-31-22-48:~/jenkins/workspace$ ls
job@tmp
ubuntu@ip-172-31-22-48:~/jenkins/workspace$ cd job@tmp/
ubuntu@ip-172-31-22-48:~/jenkins/workspace/job@tmp$ ls
ubuntu@ip-172-31-22-48:~/jenkins/workspace/job@tmp$ cd
ubuntu@ip-172-31-22-48:~$ history
1 sudo apt update -y
2 cd .ssh
3 sudo nano authorized_keys
4 cd
5 sudo kubeadm init --cri-socket=/var/run/crio/crio.sock
6 mkdir -p "$HOME"/.kube
7 sudo cp -i /etc/kubernetes/admin.conf "$HOME"/.kube/config
8 sudo chown "$(id -u)": "$(id -g)" "$HOME"/.kube/config
9 kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/master/manifests/calico.yaml
10 kubectl token create --print-join-command
11 kubectl get nodes
12 sudo apt-get install docker.io -y
13 sudo apt update
14 sudo apt install -y openjdk-11-jdk
15 sudo apt update
16 sudo apt install -y openjdk-21-jdk
17 sudo usermod -aG docker jenkins
18 sudo groupadd docker
19 sudo usermod -aG docker jenkins
20 ps aux | grep jenkins
21 sudo systemctl status jenkins
22 sudo usermod -aG docker ubuntu
23 sudo reboot
24 docker ps
25 cd /home/ubuntu/jenkins/workspace/job
26 cat /home/ubuntu/jenkins/workspace/job
27 cat /home/ubuntu/jenkins/workspace/
28 cd /home/ubuntu/jenkins/workspace
29 ls
30 cd job@tmp/
31 ls
32 cd
33 history
ubuntu@ip-172-31-22-48:~$ |

```

```

Jenkins-master/Ansible-maste X K8S-Master/Jenkins-Slave X K8S-Slave1 X K8S-Slave2 X
69 sudo apt update
70 sudo apt install -y openjdk-11-jdk
71 sudo apt update
72 sudo apt install -y openjdk-21-jdk
73 sudo usermod -aG docker jenkins
74 history
ubuntu@ip-172-31-83-59:~$ java --version
openjdk 21.0.6 2025-01-21
OpenJDK Runtime Environment (build 21.0.6+7-Ubuntu-122.04.1)
OpenJDK 64-Bit Server VM (build 21.0.6+7-Ubuntu-122.04.1, mixed mode, sharing)
ubuntu@ip-172-31-83-59:~$ jenkins --version
2.594.1
ubuntu@ip-172-31-83-59:~$ terraform --version
Terraform v1.11.4
on linux_amd64
ubuntu@ip-172-31-83-59:~$ ansible --version
Command 'ansible' not found, did you mean:
  command 'ansible' from deb ansible-core (2.12.0-1ubuntu0.1)
  command 'ansible' from deb ansible (2.10.7+merged+base+2.10.8+dfsg-1)
Try: sudo apt install <deb name>
ubuntu@ip-172-31-83-59:~$ ansible --version
Command 'ansible' not found, did you mean:
  command 'ansible' from deb ansible-core (2.12.0-1ubuntu0.1)
  command 'ansible' from deb ansible (2.10.7+merged+base+2.10.8+dfsg-1)
Try: sudo apt install <deb name>
ubuntu@ip-172-31-83-59:~$ terraform destroy
Terraform has no command named "destroy". Did you mean "destroy"?
To see all of Terraform's top-level commands, run:
  terraform -help

ubuntu@ip-172-31-83-59:~$ terraform destroy
No changes. No objects need to be destroyed.

Either you have not created any objects yet or the existing objects were already deleted outside of Terraform.

Destroy complete! Resources: 0 destroyed.
ubuntu@ip-172-31-83-59:~$ ls
ansible.sh aws awscliv2.zip terrafolder terraform.tfstate
ubuntu@ip-172-31-83-59:~|

```

Name	Instance ID	Instance state	Type	Status check	Alarm status	Availability
Jenkins-master/Ansible-master	i-094ce057dc46e7069	Terminated	t2.micro	-	<a href="#">View alarms</a>	us-east-1b
K8S-Slave1	i-0936fa04acc084fee	Terminated	t2.medium	-	<a href="#">View alarms</a>	us-east-1c
K8S-Master/Jenkins-Slave	i-0ea40f59c5481b8b1	Terminated	t2.medium	-	<a href="#">View alarms</a>	us-east-1c
K8S-Slave2	i-09c7ead9be60210ed	Terminated	t2.medium	-	<a href="#">View alarms</a>	us-east-1c

Instances | EC2 | us-east-1 | venkatesh1503/capstone-project | venkatesh1715/capstone-project | +

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:v=3;\$case=tags:true%5C;client:false;\$regex=tags:false%5C;client:false

Dell aws DevOps A1 Other favorites

Search [Alt+S]

EC2 Instances

Successfully initiated termination (deletion) of i-094ce057dc46e7069

Notifications 0 0 0 3 0 0 0 0

Instances (4) Info Last updated less than a minute ago Connect Instance state Actions Launch instances

Find Instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
Jenkins-master/Ansible-master	i-094ce057dc46e7069	Terminated	t2.micro	-	<a href="#">View alarms</a> +	us-east-1b
K8S-Slave1	i-0936fa04acc084fee	Terminated	t2.medium	-	<a href="#">View alarms</a> +	us-east-1c
K8S-Master/Jenkins-Slave	i-0ea40f59c5481b8b1	Terminated	t2.medium	-	<a href="#">View alarms</a> +	us-east-1c
K8S-Slave2	i-09c7ead9be60210ed	Terminated	t2.medium	-	<a href="#">View alarms</a> +	us-east-1c

Select an instance

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 00:56 06-05-2025

CloudShell Feedback

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
Jenkins-master/Ansible-master	i-094ce057dc46e7069	Terminated	t2.micro	-	<a href="#">View alarms</a> +	us-east-1b
K8S-Slave1	i-0936fa04acc084fee	Terminated	t2.medium	-	<a href="#">View alarms</a> +	us-east-1c
K8S-Master/Jenkins-Slave	i-0ea40f59c5481b8b1	Terminated	t2.medium	-	<a href="#">View alarms</a> +	us-east-1c
K8S-Slave2	i-09c7ead9be60210ed	Terminated	t2.medium	-	<a href="#">View alarms</a> +	us-east-1c