

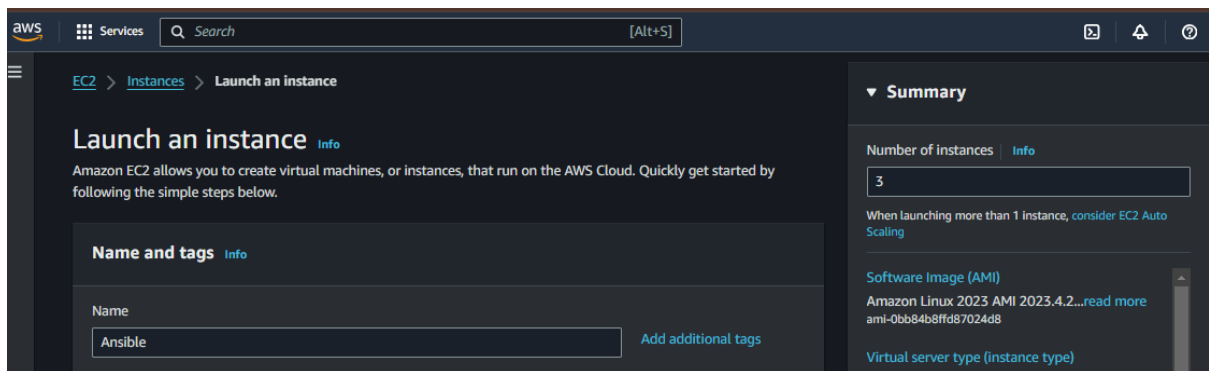
Module-6: Ansible Assignment - 1

You have been asked to:

- Setup Ansible cluster with 3 nodes
- On slave1 install java
- On slave 2 install mysql-server

Do the above tasks using Ansible playbooks

1. Launch 3 Ubuntu Instances for this Assignment



Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: Ansible

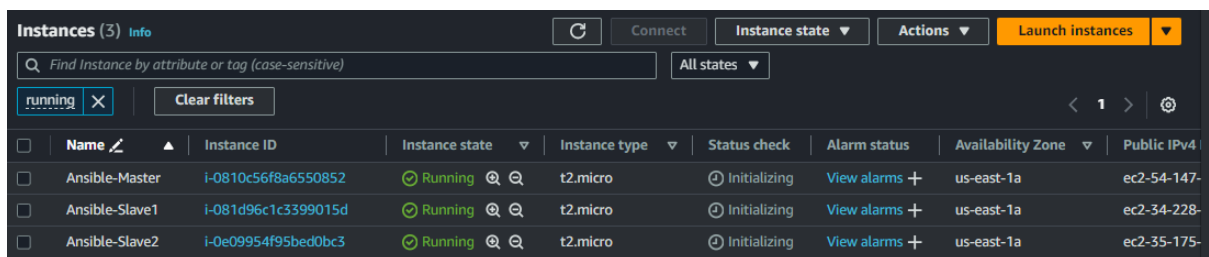
Summary

Number of instances: 3

Software Image (AMI): Amazon Linux 2023 AMI 2023.4.2...

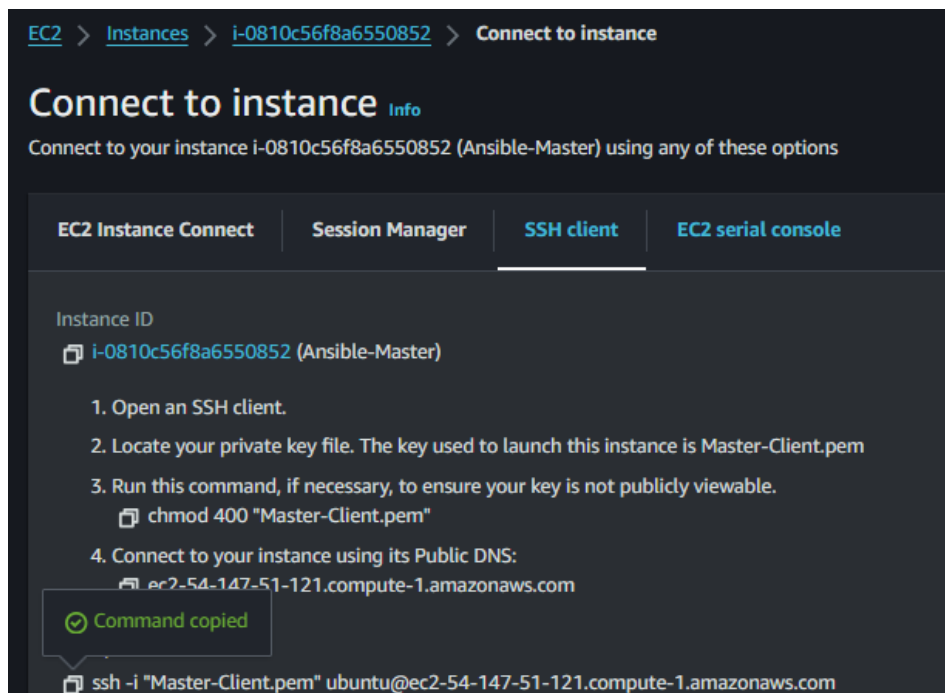
Virtual server type (instance type):

2. And Rename as your wish Now its Running Status

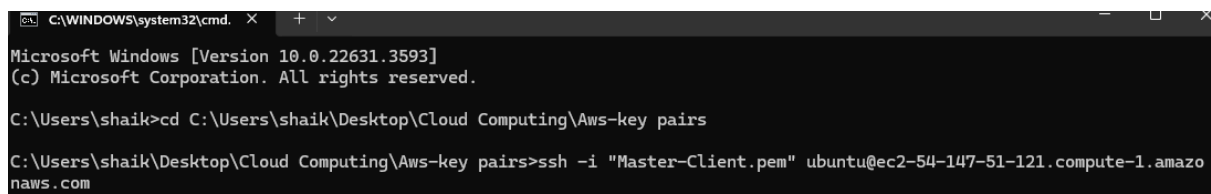


	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
<input type="checkbox"/>	Ansible-Master	i-0810c56f8a6550852	Running	t2.micro	Initializing	View alarms	us-east-1a	ec2-54-147-
<input type="checkbox"/>	Ansible-Slave1	i-081d96c1c3399015d	Running	t2.micro	Initializing	View alarms	us-east-1a	ec2-34-228-
<input type="checkbox"/>	Ansible-Slave2	i-0e09954f95bed0bc3	Running	t2.micro	Initializing	View alarms	us-east-1a	ec2-35-175-

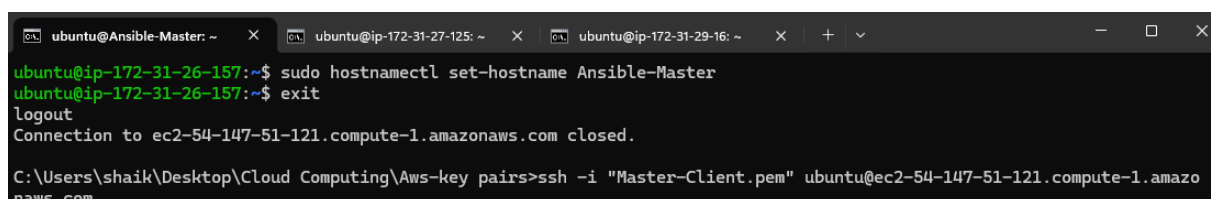
3. Click Connect Instances and Go to SSH Client and Copy it.



4. Go to Key-Pair Path and Paste SSH Client Path and Login it. And follow the same process to other 2 Instances.



5. Successfully Logged In and I change the Host Name as Ansible-Master for Remind Purpose And follow the same process to other 2 Instances.



6. Successfully changed the Host Name for 3 Instances



7. First Thing Before starting the Assignment Task we need to **Install Ansible in Ansible-Master.**

```
ubuntu@Ansible-Master: ~$ nano Ansible_install.sh
```

8. Copy the command from the **drive.**

Ansible installation ubuntu

```
sudo apt-add-repository ppa:ansible/ansible
sudo apt update
sudo apt install ansible -y
```

9. **Create a Ansible_install.sh** and Paste the Installation command and save it.

```
GNU nano 7.2 Ansible_install.sh *
sudo apt-add-repository ppa:ansible/ansible
sudo apt update
sudo apt install ansible -y
```

10. And **run** the shell file.

```
ubuntu@Ansible-Master: ~$ nano Ansible_install.sh
ubuntu@Ansible-Master: ~$ bash Ansible_install.sh
```

11. Check the **key-pair**

Instances (1/3) Info								
Find Instance by attribute or tag (case-sensitive)								
All states								
running								
Clear filters								
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
<input checked="" type="checkbox"/>	Ansible-Master	i-0810c56f8a6550852	Running	t2.micro	2/2 checks p:	View alarms +	us-east-1a	ec2-54-147-
<input type="checkbox"/>	Ansible-Slave1	i-081d96c1c3399015d	Running	t2.micro	2/2 checks p:	View alarms +	us-east-1a	ec2-34-228-
<input type="checkbox"/>	Ansible-Slave2	i-0e09954f95bed0bc3	Running	t2.micro	2/2 checks p:	View alarms +	us-east-1a	ec2-35-175-

i-0810c56f8a6550852 (Ansible-Master)		
AMI Launch index	Key pair assigned at launch	State transition reason
2	Master-Client	-

12. Go to **Key-Pair Location** and copy it.

```
-----BEGIN RSA PRIVATE KEY-----
MIIeowIBAAKCAQEAiRLbt+42IpYpwVWuKMzLKHYr73m8YBa+f2mXWzw5/Vh4+pu4
bnDxccK+xj0/e8608ZiglgNWTEB2s1HGc3B3exKpv/VLP0Dfj4jppjN4R0WoapYHX
RXeZ5EVmfenJ15WBeUAYr7y8Jf9vRbBc2SM0k4R51uRN71gseEmaq7fSF/uumxf+
NH2HQsFuyVcLqQqGvDRoIW9Reoo0JipX8/M+RCqVQwlm2XU2vhRUMNoY0S7bSM3D
3aMZow5RpAP5UmxN+7tsiZad910sI/7bjiAeP/Bla/d7bIns1fMB47D3uVFXAev/
5PqnpZG3uQjWzp0t7Q1JqBFWv89TH7KR978sLwIDAQABAoIBABgCRJff0n2WXbPu
ZPXigQcSQ7nLUdJmyXEZtpgVYXbdgzsqAVDe2Xt6h93BFJ2b3W+1sXLzfk8Yb3Hm
XuMjZNoat77WoRpwK3hBm96/mtFaS1aH+oPB YiQYf844H8Y80pRZSUd8wFvLdNr4
/BpQT0ozXv/jtz7f69IZ2kyPgu40NjjS74tx7X7nKFHzkEgr1IQyJqF4A0CdUVAL
+cWtrbyjLnIWBULIPcwI2d9frWmTleZGssCLf1PKCxbqW0EsA+sK0psnfVn47fM
sM/byb9HHu8/T0qnD8YeCM5w+vimmUdkA5Ibi26A07gGo4lFryAtk/zoAMYnch0Q
k0wc/QECgYEA04HLfKR/szwgt0BkgNuSzmx8Qdp2NsD9N1Tc0Nw00Edilfo2CEI
LM/y5A7YKt8vAJVPzPHNSnClui0BTlMBL2jF6VSqz4i4686/gDDKdh0AHu02FZ/R
/fpBSyFI50xTjGS3Ko/FONpwn4YwWOTH+HdGnsUvbWC8I2M5y30USq8CgYEApei
HRuLWv+sC5/gRaEGDkfTin1+kzcHwK5RtLUKYa7MHstb1oZY2qUIqbub0LPHLOK2
p5EVo64hKXvEEaMFOL+DMgQECie7wUEZm+SltLxD1a96n2fRdXdz6vqn6eho30yG
0JoppUC9+zXQ8A3t6gzCPmaHazzlmJZA+AKtloECgYEAkDihGIzSxtTx892kqnk0
0dCdquz0z/oH6KBClwYVmpd9vSfUjgt4F5Z0eS3mGNW2px8pGwaGhYvqddjaWx4e
YMWArmodTwZbke+YhUC/zWpY+r2lc3LXJ2fNvf0kR3UeQ00p2MvQH8RTp3M3KcZo
j5RFg13RyTN9FmRctIhLReECgYBF80hGtRHuhtIspF17fQJ8irMXMwhlL1GeNF4m
-----END RSA PRIVATE KEY-----
```

13. Create a Key-Pair **name.pem** in Ansible-Master

```
ubuntu@Ansible-Master: ~$ nano Master-Slave1-Slave2.pem
```

14. And paste and save it.

```
GNU nano 7.2 Master-Slave1-Slave2.pem *
LM/y5A7YKt8vAJVPzPHNSnClui0BTlMBL2jF6VSqz4i4686/gDDKdh0AHu02FZ/R
/fpBSyFI50xTjGS3Ko/FONpwn4YwWOTH+HdGnsUvbWC8I2M5y30USq8CgYEApei
HRuLWv+sC5/gRaEGDkfTin1+kzcHwK5RtLUKYa7MHstb1oZY2qUIqbub0LPHLOK2
p5EVo64hKXvEEaMFOL+DMgQECie7wUEZm+SltLxD1a96n2fRdXdz6vqn6eho30yG
0JoppUC9+zXQ8A3t6gzCPmaHazzlmJZA+AKtloECgYEAkDihGIzSxtTx892kqnk0
0dCdquz0z/oH6KBClwYVmpd9vSfUjgt4F5Z0eS3mGNW2px8pGwaGhYvqddjaWx4e
YMWArmodTwZbke+YhUC/zWpY+r2lc3LXJ2fNvf0kR3UeQ00p2MvQH8RTp3M3KcZo
j5RFg13RyTN9FmRctIhLReECgYBF80hGtRHuhtIspF17fQJ8irMXMwhlL1GeNF4m
-----END RSA PRIVATE KEY-----
```

15. And Give read permission to owner and Create inventory file

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Master:~$ nano Master-Slave1-Slave2.pem  
ubuntu@Ansible-Master:~$ chmod 400 Master-Slave1-Slave2.pem  
ubuntu@Ansible-Master:~$ nano inventory
```

16. Copy Public Ip of Slave1 which means Target-Server

The screenshot shows the AWS Management Console interface for an EC2 instance named 'Ansible-Slave1' with ID 'i-081d96c1c3399015d'. The instance is running on a 't2.micro' instance type in the 'us-east-1a' availability zone. The 'Instance summary' section shows the public IPv4 address '34.228.43.162' highlighted in yellow, with a link to 'open address'. The private IPv4 address is '172.31.27.125'.

17. Copy Public Ip of Slave2 which means Target-Server

The screenshot shows the AWS Management Console interface for an EC2 instance named 'Ansible-Slave2' with ID 'i-0e09954f95bed0bc3'. The instance is running on a 't2.micro' instance type in the 'us-east-1a' availability zone. The 'Instance summary' section shows the public IPv4 address '35.175.196.16' highlighted in yellow, with a link to 'open address'. The private IPv4 address is '172.31.29.16'.

18. Like we need written the Inventory file and Key-Pair Must be Same Both Master and Slave Instances

```
GNU nano 7.2  
inventory *  
[slave1]  
34.228.43.162 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem  
[slave2]  
35.175.196.16 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem
```

19. After that `ansible all -i inventory -m ping`, if it will success then master and slave instances connected otherwise it will show an error.

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Master:~$ nano inventory  
ubuntu@Ansible-Master:~$ cat inventory  
[slave1]  
34.228.43.162 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem"  
  
[slave2]  
35.175.196.16 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem"  
ubuntu@Ansible-Master:~$ ansible all -i inventory -m ping  
The authenticity of host '34.228.43.162 (34.228.43.162)' can't be established.  
ED25519 key fingerprint is SHA256:YRpSAvNf8s5efy3iiTpXSV80XS4b6j77+rTOBO/vFUs.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? 35.175.196.16 | SUCCESS => {  
  "ansible_facts": {  
    "discovered_interpreter_python": "/usr/bin/python3"  
  },  
  "changed": false,  
  "ping": "pong"  
}  
yes  
34.228.43.162 | SUCCESS => {  
  "ansible_facts": {  
    "discovered_interpreter_python": "/usr/bin/python3"  
  },  
  "changed": false,  
  "ping": "pong"  
}
```

20. Now come to the Assignment Task According to Task we have to perform with Ansible Playbook so I am Creating Name as `setup.yml`

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Master:~$ nano setup.yml
```

- **Hosts:** slave1 and slave2 (This play will run only on the host or group named slave1, slave2)
- **Become:** true (Run the tasks with elevated privileges)
- **Task:**
 - **Name:** Install Java on slave1 and Install mysql on slave2
 - **Module:** apt (Used for managing packages with the apt package manager)
 - **Option:**
 - name: default-jdk, mysql (Specifies the package to install)
 - state: present (Ensures that the package is installed)

```

GNU nano 7.2                                     setup.yml *
-----
- name: Perform package updates and installations
  hosts: all
  become: true
  tasks:
    - name: Update package lists
      apt:
        update_cache: yes

- name: Install Java on slave1
  hosts: slave1
  become: true
  tasks:
    - name: Install Java
      apt:
        name: default-jdk
        state: present

- name: Install MySQL server on slave2
  hosts: slave2
  become: true
  tasks:
    - name: Install MySQL server
      apt:
        name: mysql-server
        state: present

```

21. It will check Syntax of file it is correct or not, now its fine

```

ubuntu@Ansible-Master:~$ nano setup.yml
ubuntu@Ansible-Master:~$ ansible-playbook -i inventory setup.yml --syntax -check

playbook: setup.yml
ubuntu@Ansible-Master:~$

```

22. Now If we see it will Successfully Perform the Updated the Package Lists and **Play Name** Install Java on Slave1 and **Task** Install Java on Slave1 and Same as it Installed MySQL server on slave2

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Master:~$ ansible-playbook -i inventory setup.yml  
  
PLAY [Perform package updates and installations] *****  
  
TASK [Gathering Facts] *****  
ok: [35.175.196.16]  
ok: [34.228.43.162]  
  
TASK [Update package lists] *****  
changed: [35.175.196.16]  
changed: [34.228.43.162]  
  
PLAY [Install Java on slave1] *****  
  
TASK [Gathering Facts] *****  
ok: [34.228.43.162]  
  
TASK [Install Java] *****  
changed: [34.228.43.162]  
  
PLAY [Install MySQL server on slave2] *****  
  
TASK [Gathering Facts] *****  
ok: [35.175.196.16]  
  
TASK [Install MySQL server] *****  
changed: [35.175.196.16]  
  
PLAY RECAP *****  
34.228.43.162      : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0  
35.175.196.16     : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0  
  
ubuntu@Ansible-Master:~$
```

23. See on Slave1 Its Installed java

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Slave1: ~  
ubuntu@Ansible-Slave2: ~  
  
ubuntu@Ansible-Slave1:~$ java --version  
openjdk 21.0.3 2024-04-16  
OpenJDK Runtime Environment (build 21.0.3+9-Ubuntu-1ubuntu1)  
OpenJDK 64-Bit Server VM (build 21.0.3+9-Ubuntu-1ubuntu1, mixed mode, sharing)  
ubuntu@Ansible-Slave1:~$ systemctl status mysql.service  
Unit mysql.service could not be found.  
ubuntu@Ansible-Slave1:~$
```


24. And we can see on Slave 2 Mysql was Installed Successfully

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Slave1: ~  
ubuntu@Ansible-Slave2: ~  
ubuntu@Ansible-Slave2:~$ systemctl status mysql.service  
● mysql.service - MySQL Community Server  
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)  
   Active: active (running) since Tue 2024-05-28 15:25:19 UTC; 2min 35s ago  
     Process: 2786 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)  
    Main PID: 2795 (mysqld)  
      Status: "Server is operational"  
     Tasks: 37 (limit: 1130)  
    Memory: 352.1M (peak: 380.2M)  
       CPU: 1.435s  
    CGroup: /system.slice/mysql.service  
            └─2795 /usr/sbin/mysqld  
  
May 28 15:25:17 Ansible-Slave2 systemd[1]: Starting mysql.service - MySQL Community Server...  
May 28 15:25:19 Ansible-Slave2 systemd[1]: Started mysql.service - MySQL Community Server.  
ubuntu@Ansible-Slave2:~$ java --version  
Command 'java' not found, but can be installed with:  
sudo apt install default-jre # version 2:1.17-75, or  
sudo apt install openjdk-17-jre-headless # version 17.0.10~6ea-1  
sudo apt install openjdk-11-jre-headless # version 11.0.21+9-0ubuntu1  
sudo apt install openjdk-19-jre-headless # version 19.0.2+7-4  
sudo apt install openjdk-20-jre-headless # version 20.0.2+9-1  
sudo apt install openjdk-21-jre-headless # version 21.0.1+12-3  
sudo apt install openjdk-22-jre-headless # version 22~22ea-1  
sudo apt install openjdk-8-jre-headless # version 8u392-ga-1  
ubuntu@Ansible-Slave2:~$
```

OPTIONAL TASK:

25. I am Uninstalling Both Java and My Sql on Slave1 and Slave2, Creating uninstall.yml File

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Slave1: ~  
ubuntu@Ansible-Slave2: ~  
ubuntu@Ansible-Master:~$ nano uninstall.yml  
  
GNU nano 7.2  
uninstall.yml  
- name: Uninstall Java on slave1  
  hosts: slave1  
  become: true  
  tasks:  
    - name: Uninstall Java  
      apt:  
        name: default-jdk  
        state: absent  
        purge: yes  
        autoremove: yes  
  
- name: Uninstall MySQL server on slave2  
  hosts: slave2  
  become: true  
  tasks:  
    - name: Uninstall MySQL server  
      apt:  
        name: mysql-server  
        state: absent  
        purge: yes  
        autoremove: yes
```

26. Now We Can See Its Successfully ran

```
ubuntu@Ansible-Master: ~$ nano uninstall.yml
ubuntu@Ansible-Master:~$ ansible-playbook -i inventory uninstall.yml --syntax -check

playbook: uninstall.yml
ubuntu@Ansible-Master:~$ ansible-playbook -i inventory uninstall.yml

PLAY [Uninstall Java on slave1] *****

TASK [Gathering Facts] *****
ok: [34.228.43.162]

TASK [Uninstall Java] *****
changed: [34.228.43.162]

PLAY [Uninstall MySQL server on slave2] *****

TASK [Gathering Facts] *****
ok: [35.175.196.16]

TASK [Uninstall MySQL server] *****
changed: [35.175.196.16]

PLAY RECAP *****
34.228.43.162      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
35.175.196.16    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@Ansible-Master:~$
```

27. Now Checking On Slave1 its shown not found

```
ubuntu@Ansible-Master: ~$ 
ubuntu@Ansible-Slave1: ~$ java -version
Command 'java' not found, but can be installed with:
sudo apt install default-jre          # version 2:1.17-75, or
sudo apt install openjdk-17-jre-headless # version 17.0.10~6ea-1
sudo apt install openjdk-11-jre-headless # version 11.0.21+9-0ubuntu1
sudo apt install openjdk-19-jre-headless # version 19.0.2+7-4
sudo apt install openjdk-20-jre-headless # version 20.0.2+9-1
sudo apt install openjdk-21-jre-headless # version 21.0.1+12-3
sudo apt install openjdk-22-jre-headless # version 22~22ea-1
sudo apt install openjdk-8-jre-headless  # version 8u392-ga-1
ubuntu@Ansible-Slave1:~$
```

28. On Slave2 also, Assignment Task 1 Completed.

```
ubuntu@Ansible-Master: ~$ 
ubuntu@Ansible-Slave1: ~$ 
ubuntu@Ansible-Slave2: ~$ systemctl status mysql.service
Unit mysql.service could not be found.
ubuntu@Ansible-Slave2:~$
```

Module-6: Ansible Assignment - 2

You have been asked to:

- Create a script which can add text "This text has been added by custom script" to /tmp.1.txt
- Run this script using Ansible on all the hosts

1. If You Stop Instances then Replace New Public IP Addresses

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Master:~$ cat inventory  
[slave1]  
34.228.43.162 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem  
  
[slave2]  
35.175.196.16 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem  
ubuntu@Ansible-Master:~$
```

2. So I Stopped Instances and Now I need to Update New IP Address

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Master:~$ nano inventory
```

3. This is New IP of Slave1

Instance	ID	Status	Type	Checks	Region	Subnet
Ansible-Slave1	i-051a1d40514cf61c8	Running	t2.micro	2/2 checks passed	us-east-1a	ec2-54
Ansible-Slave2	i-0cdcf4ed7ee38a75	Running	t2.micro	2/2 checks passed	us-east-1a	ec2-54

i-051a1d40514cf61c8 (Ansible-Slave1)
Instance ID: i-051a1d40514cf61c8 (Ansible-Slave1)
Public IPv4 address: **54.173.233.13** [open address](#)
Private IPv4 addresses: 172.31.19.56

4. This is New IP of Slave2

Instance	ID	Status	Type	Checks	Region	Subnet
Ansible-Slave2	i-0cdcf4ed7ee38a75	Running	t2.micro	2/2 checks passed	us-east-1a	ec2-54

i-0cdcf4ed7ee38a75 (Ansible-Slave2)
Instance ID: i-0cdcf4ed7ee38a75 (Ansible-Slave2)
Public IPv4 address: **54.196.234.163** [open address](#)
Private IPv4 addresses: 172.31.24.188

5. This is Updated IP's

```
ubuntu@Ansible-Master: ~ X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~
GNU nano 7.2 inventory *
[all]
54.173.233.13 ansible_ssh_private_key_file=~/.ssh/known_hosts
54.196.234.163 ansible_ssh_private_key_file=~/.ssh/known_hosts
```

6. I got an Error so I Go with this command `ssh-keyscan -H ip_address >> ~/.ssh/known_hosts` manually adds the SSH host key of a server to your local `known_hosts` file.

```
ubuntu@Ansible-Master: ~ X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~ X + v
ubuntu@Ansible-Master:~$ ansible all -i inventory -m ping

The authenticity of host '54.173.233.13 (54.173.233.13)' can't be established.
ED25519 key fingerprint is SHA256:2fb+oQ9KPyGmpraeyi3aIYUo09xG/SlqAU/coUdJhzA.
This key is not known by any other names.
The authenticity of host '54.196.234.163 (54.196.234.163)' can't be established.
ED25519 key fingerprint is SHA256:SK9WaJ4xyEXRRdsUMx13EVNzULUDeXYEiP8SLBKcXak.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
54.173.233.13 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
54.196.234.163 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: Host key verification failed.",
  "unreachable": true
}
ubuntu@Ansible-Master:~$ ssh-keyscan -H 54.196.234.163 >> ~/.ssh/known_hosts
# 54.196.234.163:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13
# 54.196.234.163:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13
# 54.196.234.163:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13
# 54.196.234.163:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13
# 54.196.234.163:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13
ubuntu@Ansible-Master:~$ ansible all -i inventory -m ping
54.173.233.13 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
54.196.234.163 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

7. Create Script File

```
ubuntu@Ansible-Master: ~ X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~ X +
ubuntu@Ansible-Master:~$ nano add_text.sh
```

8. Write the Script as per the Assignment and save in tmp file/tmp.1.txt
9. What will do this script, `#!/bin/bash` Mandatory
10. `Echo "text"` it will Print the text and it will save `>> /tmp/tmp.1.txt` into this path

```
ubuntu@Ansible-Master: ~ X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~ X + v
GNU nano 7.2 add_text.sh *
#!/bin/bash
echo "This text has been added by custom script" >> /tmp/tmp.1.txt
```

11. Give Executable permission and Create Playbook with name of run script.yml

```
ubuntu@Ansible-Master: ~ X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~ X +
ubuntu@Ansible-Master:~$ nano add_text.sh
ubuntu@Ansible-Master:~$ chmod +x add_text.sh
ubuntu@Ansible-Master:~$ nano run_script.yml
```

12. Copy the custom script file
13. Will take Src: `add_text.sh`
14. And Going to Copy this dest: `/tmp/add_text.sh`
15. Run the custom Script
16. It will ran

```
ubuntu@Ansible-Master: ~ X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~ X
GNU nano 7.2 run_script.yml
---
- name: Add text to /tmp/tmp.1.txt on all hosts
  hosts: all
  become: true

  tasks:
    - name: Copy the custom script to the remote hosts
      copy:
        src: add_text.sh
        dest: /tmp/add_text.sh
        mode: '0755'

    - name: Run the custom script on the remote hosts
      command: /tmp/add_text.sh
```

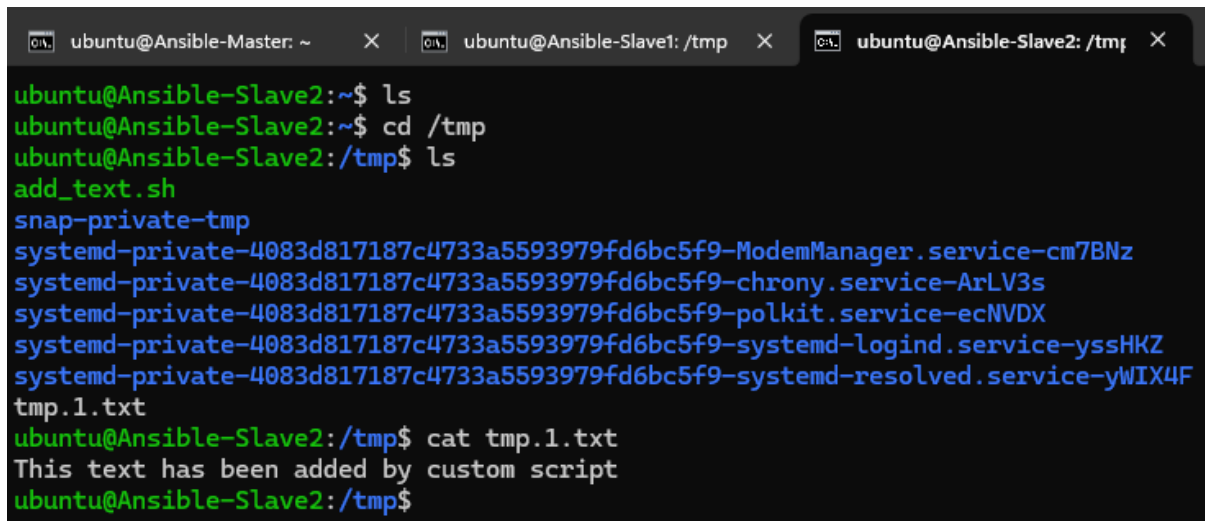
17. Syntax check executed successfully
18. And performed tasks successfully
19. We can see it gathering the facts and copying the script and running the script on remote hosts

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Master:~$ ansible-playbook -i inventory run_script.yml --syntax -check  
playbook: run_script.yml  
ubuntu@Ansible-Master:~$ ansible-playbook -i inventory run_script.yml  
PLAY [Add text to /tmp/tmp.1.txt on all hosts] *****  
TASK [Gathering Facts] *****  
ok: [54.196.234.163]  
ok: [54.173.233.13]  
TASK [Copy the custom script to the remote hosts] *****  
changed: [54.173.233.13]  
changed: [54.196.234.163]  
TASK [Run the custom script on the remote hosts] *****  
changed: [54.196.234.163]  
changed: [54.173.233.13]  
PLAY RECAP *****  
54.173.233.13      : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0  
54.196.234.163    : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0  
ubuntu@Ansible-Master:~$
```

20. Go to Ansible-Slave1 and ls(list of contains) there is no lists because it is hidden file so cd /tmp ls u can see the ran scripted and file

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Slave1: /tmp  
ubuntu@Ansible-Slave2: /tmp  
ubuntu@Ansible-Slave1:~$ ls  
ubuntu@Ansible-Slave1:~$ cd /tmp  
ubuntu@Ansible-Slave1:/tmp$ ls  
add_text.sh  
snap-private-tmp  
systemd-private-763e7213321d4f49b3e34a63089bd056-ModemManager.service-ru048G  
systemd-private-763e7213321d4f49b3e34a63089bd056-chrony.service-EidM9X  
systemd-private-763e7213321d4f49b3e34a63089bd056-polkit.service-qhKAXM  
systemd-private-763e7213321d4f49b3e34a63089bd056-systemd-logind.service-mWQ09j  
systemd-private-763e7213321d4f49b3e34a63089bd056-systemd-resolved.service-VE8qAR  
tmp.1.txt  
ubuntu@Ansible-Slave1:/tmp$ cat tmp.1.txt  
This text has been added by custom script  
ubuntu@Ansible-Slave1:/tmp$
```

21. Go to `Ansible-Slave2` and `ls` (list of contains) there is no lists because it is hidden file so `cd /tmp` `ls` u can see the ran scripted and file



```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Slave1: /tmp  
ubuntu@Ansible-Slave2: /tmp  
ubuntu@Ansible-Slave2:~$ ls  
ubuntu@Ansible-Slave2:~$ cd /tmp  
ubuntu@Ansible-Slave2:/tmp$ ls  
add_text.sh  
snap-private-tmp  
systemd-private-4083d817187c4733a5593979fd6bc5f9-ModemManager.service-cm7BNz  
systemd-private-4083d817187c4733a5593979fd6bc5f9-chrony.service-ArLV3s  
systemd-private-4083d817187c4733a5593979fd6bc5f9-polkit.service-ecNVDX  
systemd-private-4083d817187c4733a5593979fd6bc5f9-systemd-logind.service-yssHKZ  
systemd-private-4083d817187c4733a5593979fd6bc5f9-systemd-resolved.service-yWIX4F  
tmp.1.txt  
ubuntu@Ansible-Slave2:/tmp$ cat tmp.1.txt  
This text has been added by custom script  
ubuntu@Ansible-Slave2:/tmp$
```

22. Stop Instances, Don't Terminate it will use for next Upcoming Tasks

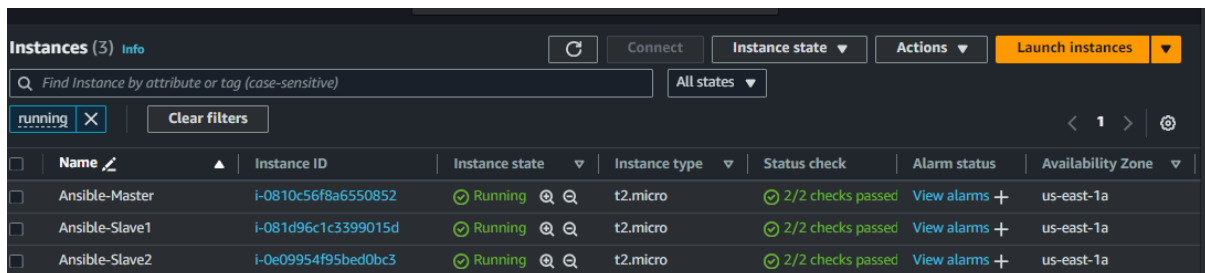
Module-6: Ansible Assignment - 3

You have been asked to:

- Create 2 Ansible Roles
- Install apache2 on slave1 using one role and nginx on slave2 using the other role

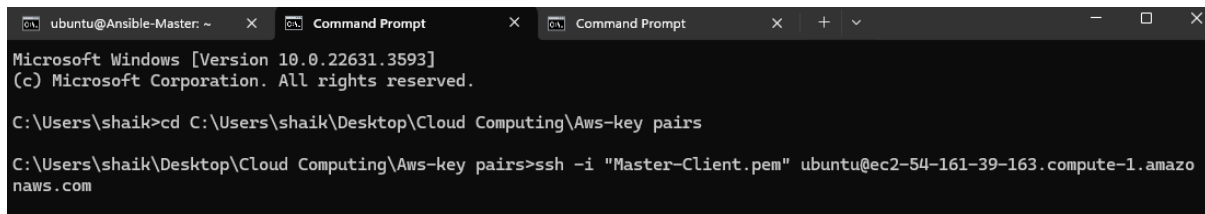
Above should be implemented using different Ansible Roles

1. Enable 3 Running Instances



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	Ansible-Master	i-0810c56f8a6550852	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a
<input type="checkbox"/>	Ansible-Slave1	i-081d96c1c3399015d	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a
<input type="checkbox"/>	Ansible-Slave2	i-0e09954f95bed0bc3	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a

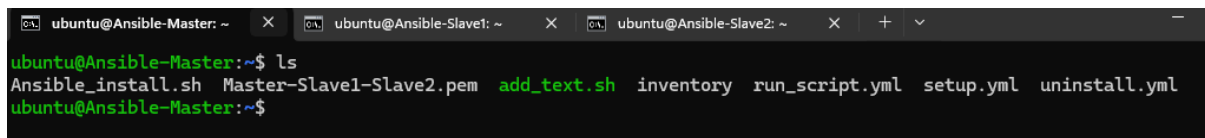
2. Go to Key-Pair Path and Paste SSH client Path to Connect Instance



```
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.

C:\Users\shaik>cd C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs
C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs>ssh -i "Master-Client.pem" ubuntu@ec2-54-161-39-163.compute-1.amazonaws.com
```

3. ls



```
ubuntu@Ansible-Master:~$ ls
Ansible_install.sh  Master-Slave1-Slave2.pem  add_text.sh  inventory  run_script.yml  setup.yml  uninstall.yml
ubuntu@Ansible-Master:~$
```


4. For this Assignment We Need 2 Ansible Role, Previously We Performed Task with Ansible Play-Book, So Creating apache2 and nginx, 2 different role

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Master:~$ ansible-galaxy init apache2  
- Role apache2 was created successfully  
ubuntu@Ansible-Master:~$ ansible-galaxy init nginx  
- Role nginx was created successfully  
ubuntu@Ansible-Master:~$ sudo apt install tree  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following NEW packages will be installed:  
  tree
```

5. we able to see the apache2 and nginx role

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Master:~$ ls  
Ansible_install.sh  add_text.sh  inventory  run_script.yml  uninstall.yml  
Master-Slave1-Slave2.pem  apache2      nginx      setup.yml  
ubuntu@Ansible-Master:~$
```

6. go to apache2 and nginx we can see the ansible role directories.

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Master:~$ cd apache2  
ubuntu@Ansible-Master:~/apache2$ ls  
README.md  defaults  files  handlers  meta  tasks  templates  tests  vars  
ubuntu@Ansible-Master:~/apache2$ cd ..  
ubuntu@Ansible-Master:~$ cd nginx  
ubuntu@Ansible-Master:~/nginx$ ls  
README.md  defaults  files  handlers  meta  tasks  templates  tests  vars  
ubuntu@Ansible-Master:~/nginx$
```

7. Optional: I Installed tree(sudo apt install tree) for Visual files of inside the directories total 9 directories and 8 files is there.

```
ubuntu@Ansible-Master: ~/a| X  ubuntu@Ansible-Slave1: ~ X
ubuntu@Ansible-Master:~/apache2$ tree
.
├── README.md
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml

9 directories, 8 files
ubuntu@Ansible-Master:~/apache2$
```

8. Same for **nginx**

```
ubuntu@Ansible-Master: ~/n  X  ubuntu@Ansible-Slave1: ~  X
ubuntu@Ansible-Master:~/nginx$ tree
.
├── README.md
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml

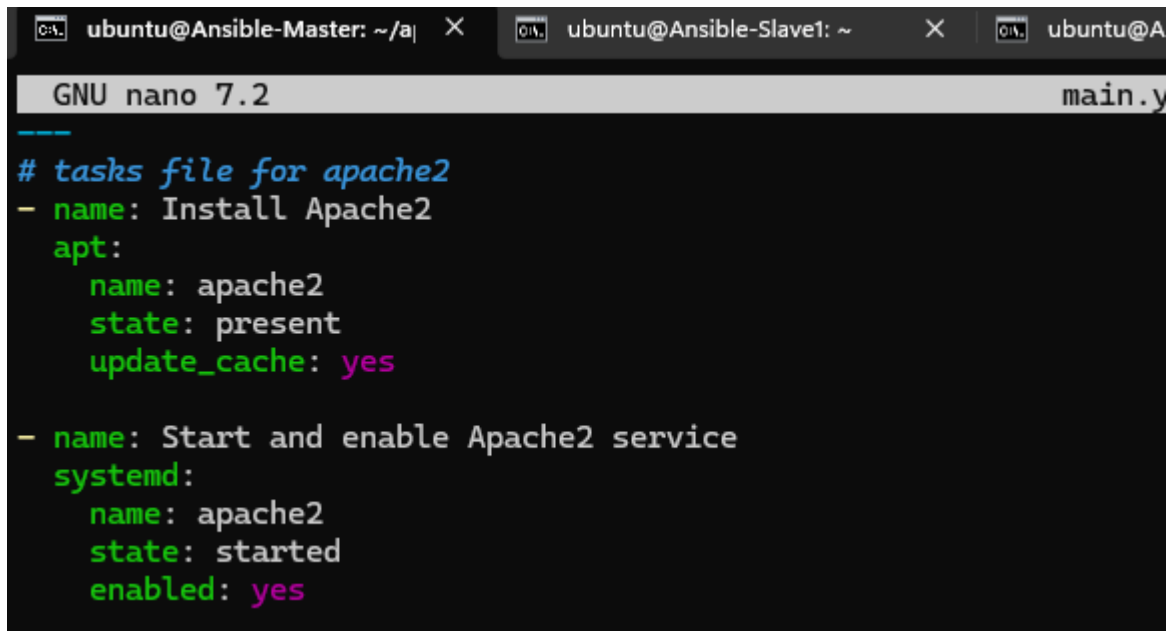
9 directories, 8 files
```

9. Now Start from **Apache2 Role**, Whatever we write regarding task we go inside the tasks in that main.yml file we are going to write the task

```
ubuntu@Ansible-Master: ~/a/  X  ubuntu@Ansible-Slave1: ~  X  ubuntu@Ansible-Slave2: ~
ubuntu@Ansible-Master:~/apache2$ ls
README.md  defaults  files  handlers  meta  tasks  templates  tests  vars
ubuntu@Ansible-Master:~/apache2$ cd tasks
ubuntu@Ansible-Master:~/apache2/tasks$ ls
main.yml
ubuntu@Ansible-Master:~/apache2/tasks$ nano main.yml
```

10. here we are written the script `install apache` and `apt(Package)` name `apache2` and state `present` and updating the cache and start and enable apache service

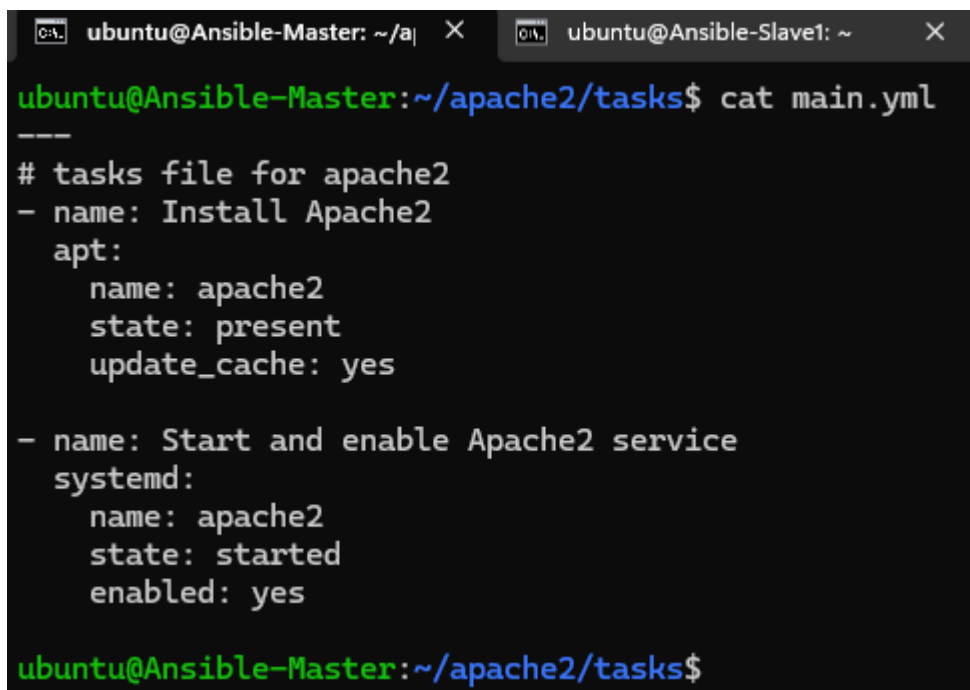
Note: Here we are not Mentioning the Host because `Ansible Role` not `Playbook`



```
GNU nano 7.2 main.y
---
# tasks file for apache2
- name: Install Apache2
  apt:
    name: apache2
    state: present
    update_cache: yes

- name: Start and enable Apache2 service
  systemd:
    name: apache2
    state: started
    enabled: yes
```

11. Cat view



```
ubuntu@Ansible-Master: ~/apache2/tasks$ cat main.yml
---
# tasks file for apache2
- name: Install Apache2
  apt:
    name: apache2
    state: present
    update_cache: yes

- name: Start and enable Apache2 service
  systemd:
    name: apache2
    state: started
    enabled: yes

ubuntu@Ansible-Master: ~/apache2/tasks$
```

12. Now Start from nginx Role, Whatever we write regarding task we go inside the tasks in that main.yml file we are going to write the task

```
ubuntu@Ansible-Master: ~/n  X  ubuntu@Ansible-Slave1: ~  X  ubuntu@Ansible-Slave2: ~  
ubuntu@Ansible-Master:~/nginx$ ls  
README.md  defaults  files  handlers  meta  tasks  templates  tests  vars  
ubuntu@Ansible-Master:~/nginx$ cd tasks  
ubuntu@Ansible-Master:~/nginx/tasks$ ls  
main.yml  
ubuntu@Ansible-Master:~/nginx/tasks$ nano main.yml
```

13. here we are written the script install nginx and apt(Package) name nginx and state present and updating the cache and start and enabled nginx service

```
GNU nano 7.2  
---  
# tasks file for nginx  
- name: Install Nginx  
  apt:  
    name: nginx  
    state: present  
    update_cache: yes  
  
- name: Start and enable Nginx service  
  systemd:  
    name: nginx  
    state: started  
    enabled: yes
```

14. cat view

```
ubuntu@Ansible-Master: ~/n X ubuntu@Ansible-Slave1: ~ X u
ubuntu@Ansible-Master:~/nginx/tasks$ cat main.yml
---
# tasks file for nginx
- name: Install Nginx
  apt:
    name: nginx
    state: present
    update_cache: yes

- name: Start and enable Nginx service
  systemd:
    name: nginx
    state: started
    enabled: yes
ubuntu@Ansible-Master:~/nginx/tasks$
```

15. Go to home

```
ubuntu@Ansible-Master: ~/n X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~
ubuntu@Ansible-Master:~/nginx/tasks$ cd ~
```

16. I am Copying Public IP of Slave 1

<input checked="" type="checkbox"/>	Ansible-Slave1	i-081d96c1c3399015d	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a
<input type="checkbox"/>	Ansible-Slave2	i-0e09954f95bed0bc3	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a

i-081d96c1c3399015d (Ansible-Slave1)		
Details	Status and alarms <small>New</small>	Monitoring Security Networking Storage Tags
▼ Instance summary <small>Info</small>		
Instance ID i-081d96c1c3399015d (Ansible-Slave1)	Public IPv4 address 54.161.39.163 open address	Private IPv4 addresses 172.31.27.125

17. I am Copying Public IP of Slave 2

<input type="checkbox"/>	Ansible-Slave1	i-081d96c1c3399015d	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a
<input checked="" type="checkbox"/>	Ansible-Slave2	i-0e09954f95bed0bc3	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a

i-0e09954f95bed0bc3 (Ansible-Slave2)		
Details	Status and alarms <small>New</small>	Monitoring Security Networking Storage Tags
▼ Instance summary <small>Info</small>		
Instance ID i-0e09954f95bed0bc3 (Ansible-Slave2)	Public IPv4 address 18.214.98.150 open address	Private IPv4 addresses 172.31.29.16

17. Cat Inventory, for what the IP's there in inventory file

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Slave1: ~  
ubuntu@Ansible-Slave2: ~  
ubuntu@Ansible-Master:~$ cat inventory  
[all]  
34.228.43.162 ansible_ssh_private_key_file=~/.ssh/ansible-ssh-key.pem  
35.175.196.16 ansible_ssh_private_key_file=~/.ssh/ansible-ssh-key.pem  
ubuntu@Ansible-Master:~$
```

18. Replace new public IP and Remember key-pair have to same both Ansible-master and Ansible slaves

```
GNU nano 7.2 inventory *  
[all]  
54.161.39.163 ansible_ssh_private_key_file=~/.ssh/ansible-ssh-key.pem  
18.214.98.150 ansible_ssh_private_key_file=~/.ssh/ansible-ssh-key.pem
```

19. And Now we Need to change according to task Because we have to install different packages in both slaves

```
GNU nano 7.2 inventory *  
[slave1]  
54.161.39.163 ansible_ssh_private_key_file=~/.ssh/ansible-ssh-key.pem  
[slave2]  
18.214.98.150 ansible_ssh_private_key_file=~/.ssh/ansible-ssh-key.pem
```

20. Now Create Play-Book

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Slave1: ~  
ubuntu@Ansible-Slave2: ~  
ubuntu@Ansible-Master:~$ nano inventory  
ubuntu@Ansible-Master:~$ nano double-role.yml
```

21. Write Hosts and add which role you have to Assign

22. Here I attached `apache2` role for slave1 server

23. and I attach `nginx` role for slave 2 server

```
ubuntu@Ansible-Master: ~  
GNU nano 7.2  
# doublerole.yml  
-----  
- hosts: slave1  
  become: yes  
  roles:  
    - apache2  
  
- hosts: slave2  
  become: yes  
  roles:  
    - nginx
```

22. syntax check

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Master:~$ ansible-playbook -i inventory double-role.yml --syntax -check  
playbook: double-role.yml  
ubuntu@Ansible-Master:~$
```

23. Before executing the task, I am showing in the `ansible slave 1 and slave 2` which is `not installed`

```
ubuntu@Ansible-Slave1:~$ systemctl status apache2  
Unit apache2.service could not be found.  
ubuntu@Ansible-Slave1:~$  
  
ubuntu@Ansible-Slave2:~$ systemctl status nginx  
Unit nginx.service could not be found.  
ubuntu@Ansible-Slave2:~$
```


24. Now we are able to see successfully completed

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Master:~$ ansible-playbook -i inventory double-role.yml --syntax -check  
playbook: double-role.yml  
ubuntu@Ansible-Master:~$ ansible-playbook -i inventory double-role.yml  
  
PLAY [slave1] *****  
TASK [Gathering Facts] *****  
ok: [54.161.39.163]  
  
TASK [apache2 : Install Apache2] *****  
changed: [54.161.39.163]  
  
TASK [apache2 : Start and enable Apache2 service] *****  
ok: [54.161.39.163]  
  
PLAY [slave2] *****  
TASK [Gathering Facts] *****  
The authenticity of host '18.214.98.150 (18.214.98.150)' can't be established.  
ED25519 key fingerprint is SHA256:lbiygpYdrhyWoPKKLIsqIFNW/GiZj+mu+OEAnE9JuyI.  
This host key is known by the following other names/addresses:  
  ~/.ssh/known_hosts:1: [hashed name]  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
ok: [18.214.98.150]  
  
TASK [nginx : Install Nginx] *****  
changed: [18.214.98.150]  
  
TASK [nginx : Start and enable Nginx service] *****  
ok: [18.214.98.150]  
  
PLAY RECAP *****  
18.214.98.150      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0  
54.161.39.163     : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0  
  
ubuntu@Ansible-Master:~$
```

25. We can see apache2 server on slave1

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Slave1: ~  
ubuntu@Ansible-Slave2: ~  
ubuntu@Ansible-Slave1:~$ systemctl status apache2  
● apache2.service - The Apache HTTP Server  
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)  
   Active: active (running) since Wed 2024-05-29 16:08:10 UTC; 1min 40s ago  
     Docs: https://httpd.apache.org/docs/2.4/  
  Main PID: 9618 (apache2)  
    Tasks: 55 (limit: 1130)  
   Memory: 5.5M (peak: 5.5M)  
      CPU: 42ms  
   CGroup: /system.slice/apache2.service  
           └─9618 /usr/sbin/apache2 -k start  
             └─9621 /usr/sbin/apache2 -k start  
               └─9622 /usr/sbin/apache2 -k start  
  
May 29 16:08:10 Ansible-Slave1 systemd[1]: Starting apache2.service - The Apache HTTP Server...  
May 29 16:08:10 Ansible-Slave1 systemd[1]: Started apache2.service - The Apache HTTP Server.  
ubuntu@Ansible-Slave1:~$
```

26. We can see nginx on slave2

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Slave1: ~  
ubuntu@Ansible-Slave2: ~  
ubuntu@Ansible-Slave2:~$ systemctl status nginx  
● nginx.service - A high performance web server and a reverse proxy server  
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)  
   Active: active (running) since Wed 2024-05-29 16:08:33 UTC; 1min 45s ago  
     Docs: man:nginx(8)  
  Process: 8768 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)  
  Process: 8770 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)  
 Main PID: 8771 (nginx)  
    Tasks: 2 (limit: 1130)  
  Memory: 1.7M (peak: 1.9M)  
     CPU: 11ms  
   CGroup: /system.slice/nginx.service  
           └─8771 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"  
             └─8772 "nginx: worker process"  
  
May 29 16:08:33 Ansible-Slave2 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...  
May 29 16:08:33 Ansible-Slave2 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.  
ubuntu@Ansible-Slave2:~$
```

DevOps Certification Training



Module-6: Ansible Assignment - 4

You have been asked to:

- Use the previous deployment of ansible cluster
- Configure the files folder in the role with index.html which should be replaced with the original index.html

All of the above should only happen on the slave which has nginx installed using the role

1. I am continuing the Assignment 4, Using previous deployment so its related to nginx server so I am going inside the nginx role

```
ubuntu@Ansible-Master: ~/n X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~ X +
ubuntu@Ansible-Master:~$ cd nginx
ubuntu@Ansible-Master:~/nginx$ ls
README.md  defaults  files  handlers  meta  tasks  templates  tests  vars
ubuntu@Ansible-Master:~/nginx$ tree
.
├── README.md
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml

9 directories, 8 files
ubuntu@Ansible-Master:~/nginx$
```

2. Go to Files folder here we creating a index.html file

```
ubuntu@Ansible-Master: ~/n X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~ X +
ubuntu@Ansible-Master:~/nginx$ cd files
ubuntu@Ansible-Master:~/nginx/files$ ls
ubuntu@Ansible-Master:~/nginx/files$ nano index.html
```

3. Write a Html Script, Here I Written Title "Welcome to Nginx on Slave2"

Body "Success! Nginx is installed on Slave2"

```
GNU nano 7.2 index.html *
<html>
<head>
  <title>Welcome to Nginx on Slave2</title>
</head>
<body>
  <h1>Success! Nginx is installed on Slave2</h1>
</body>
</html>
```

4. **Optional** Or Otherwise we can give **destination location** on task, Or we go Vars folder and **edit main.yml**

```
ubuntu@Ansible-Master: ~/n X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~ X
ubuntu@Ansible-Master:~/nginx/tasks$ cd ..
ubuntu@Ansible-Master:~/nginx$ cd vars
ubuntu@Ansible-Master:~/nginx/vars$ ls
main.yml
ubuntu@Ansible-Master:~/nginx/vars$ nano main.yml
```

5. Giving **destination path** was **/var/www/html**

```
ubuntu@Ansible-Master: ~/n X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~ X
GNU nano 7.2
---
# vars file for nginx
destpath: /var/www/html
```

6. go to **tasks file** and **edit main file**

```
ubuntu@Ansible-Master: ~/n X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~ X
ubuntu@Ansible-Master:~/nginx/tasks$ ls
main.yml
ubuntu@Ansible-Master:~/nginx/tasks$ nano main.yml
```

7. **systemd**: Uses the Ansible **systemd** module to start the Nginx service and enable it to start at boot.

copy: Uses the Ansible **copy** module to transfer **index.html** to the destination path specified by **destpath**.

owner, group, mode: Set the file's owner, group, and permissions.

```
ubuntu@Ansible-Master: ~/n X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~ X +
GNU nano 7.2 main.yml
---
# tasks file for nginx
- name: Install Nginx
  apt:
    name: nginx
    state: present

- name: Start and enable Nginx service
  systemd:
    name: nginx
    state: started
    enabled: yes

- name:
  copy:
    src: index.html
    dest: "{{ destpath }}"
    owner: www-data
    group: www-data
    mode: '0644'
```

8. Cat View

```
ubuntu@Ansible-Master: ~/n X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~ X
ubuntu@Ansible-Master:~/nginx/tasks$ cat main.yml
---
# tasks file for nginx
- name: Install Nginx
  apt:
    name: nginx
    state: present

- name: Start and enable Nginx service
  systemd:
    name: nginx
    state: started
    enabled: yes

- name:
  copy:
    src: index.html
    dest: "{{ destpath }}"
    owner: www-data
    group: www-data
    mode: '0644'
ubuntu@Ansible-Master:~/nginx/tasks$
```

9. Now Configure the playbook

```
ubuntu@Ansible-Master: ~ X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~
ubuntu@Ansible-Master:~$ nano double-role.yml
```

10. I left has it is.

```
ubuntu@Ansible-Master: ~  X  ubuntu@Ansible-Slave1: ~  X  ubuntu@Ansible-Slave2: ~
GNU nano 7.2 double-
# doublerole.yml
- hosts: slave1
  become: yes
  roles:
    - apache2

- hosts: slave2
  become: yes
  roles:
    - nginx
```

11. All Ok one thing. it performed that is nginx copying to slave2

```
ubuntu@Ansible-Master: ~  X  ubuntu@Ansible-Slave1: ~  X  ubuntu@Ansible-Slave2: ~  X  +  v
ubuntu@Ansible-Master:~$ ansible-playbook -i inventory double-role.yml --syntax -check
playbook: double-role.yml
ubuntu@Ansible-Master:~$ ansible-playbook -i inventory double-role.yml

PLAY [slave1] *****

TASK [Gathering Facts] *****
ok: [54.161.39.163]

TASK [apache2 : Install Apache2] *****
ok: [54.161.39.163]

TASK [apache2 : Start and enable Apache2 service] *****
ok: [54.161.39.163]

PLAY [slave2] *****

TASK [Gathering Facts] *****
ok: [18.214.98.150]

TASK [nginx : Install Nginx] *****
ok: [18.214.98.150]

TASK [nginx : Start and enable Nginx service] *****
ok: [18.214.98.150]

TASK [nginx : copy] *****
changed: [18.214.98.150]

PLAY RECAP *****
18.214.98.150      : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
54.161.39.163    : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu@Ansible-Master:~$
```

12. we can see the **index.html** and I deleted Default html file(No Need to Delete)

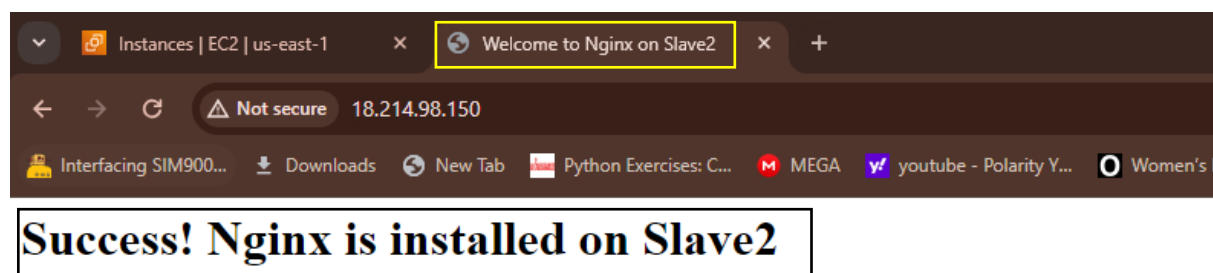
```
ubuntu@Ansible-Master: ~ X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: /var, >
ubuntu@Ansible-Slave2:~$ cd /var/www/html
ubuntu@Ansible-Slave2:/var/www/html$ ls
index.html
ubuntu@Ansible-Slave2:/var/www/html$ cat index.html
<html>
<head>
  <title>Welcome to Nginx on Slave2</title>
</head>
<body>
  <h1>Success! Nginx is installed on Slave2</h1>
</body>
</html>

ubuntu@Ansible-Slave2:/var/www/html$
```

13. Notice the slave 2 IP and copy it

```
ubuntu@Ansible-Master: ~ X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~
GNU nano 7.2 inventory *
[slave1]
54.161.39.163 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem"
[slave2]
18.214.98.150 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem"
```

14. Paste it in Browser, see the **title and Body**



OPTIONAL TASK:

15. I am Uninstalling Both Servers

```
ubuntu@Ansible-Master: ~ X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: /var/ X +
ubuntu@Ansible-Master:~$ nano uninstall.yml
```

16. State: absent and Purge: yes and autoremove:yes

```
ubuntu@Ansible-Master: ~ X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: /var/ X + v
GNU nano 7.2 uninstall.yml
- name: Uninstall apache2 on slave1
  hosts: slave1
  become: true
  tasks:
    - name: Uninstall apache2
      apt:
        name: apache2
        state: absent
        purge: yes
        autoremove: yes

- name: Uninstall nginx on slave2
  hosts: slave2
  become: true
  tasks:
    - name: Uninstall nginx
      apt:
        name: nginx
        state: absent
        purge: yes
        autoremove: yes
```

17. Perform the task and its changed on both IPS and its performed successfully.

```
ubuntu@Ansible-Master: ~ X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: /var/ X + v
ubuntu@Ansible-Master:~$ ansible-playbook -i inventory uninstall.yml --syntax -check
playbook: uninstall.yml
ubuntu@Ansible-Master:~$ ansible-playbook -i inventory uninstall.yml

PLAY [Uninstall apache2 on slave1] *****
TASK [Gathering Facts] *****
ok: [54.161.39.163]

TASK [Uninstall apache2] *****
changed: [54.161.39.163]

PLAY [Uninstall nginx on slave2] *****
TASK [Gathering Facts] *****
ok: [18.214.98.150]

TASK [Uninstall nginx] *****
changed: [18.214.98.150]

PLAY RECAP *****
18.214.98.150      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
54.161.39.163    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu@Ansible-Master:~$
```


18. Check on Slave1, apache2 not found

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Slave1: ~  
ubuntu@Ansible-Slave2: ~  
ubuntu@Ansible-Slave1:~$ systemctl status apache2  
Unit apache2.service could not be found.  
ubuntu@Ansible-Slave1:~$
```

19. Check on Slave2, nginx not found

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Slave1: ~  
ubuntu@Ansible-Slave2: ~  
ubuntu@Ansible-Slave2:~$ systemctl status nginx  
Unit nginx.service could not be found.  
ubuntu@Ansible-Slave2:~$
```

DevOps Certification Training



Module-6: Ansible Assignment - 5

You have been asked to:

- Create a new deployment of ansible cluster of 5 nodes
- Label 2 nodes as test and other 2 as prod
- Install java on test nodes
- Install mysql-server on prod nodes

Use Ansible roles for the above, group the hosts under test and prod

1. Enable the Instances

Instances (3/3) Info									
<input type="text" value="Find instance by attribute or tag (case-sensitive)"/>									
<input type="text" value="ansible"/> <input type="button" value="Clear filters"/>									
<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Instance state	Alarm status	Availability Zone	Public IP	
<input checked="" type="checkbox"/>	Ansible-Master	i-0810c56f8a6550852	Stopped	t2.micro	Stopped	View alarms	us-east-1a	-	
<input checked="" type="checkbox"/>	Ansible-Slave1	i-081d96c1c3399015d	Stopped	t2.micro	Stopped	View alarms	us-east-1a	-	
<input checked="" type="checkbox"/>	Ansible-Slave2	i-0e09954f95bed0bc3	Stopped	t2.micro	Stopped	View alarms	us-east-1a	-	

2. Now its Running

<input type="text" value="ansible"/> <input type="button" value="Clear filters"/>									
<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP	
<input checked="" type="checkbox"/>	Ansible-Master	i-0810c56f8a6550852	Running	t2.micro	-	View alarms	us-east-1a	ec2-	
<input checked="" type="checkbox"/>	Ansible-Slave1	i-081d96c1c3399015d	Running	t2.micro	-	View alarms	us-east-1a	ec2-	
<input checked="" type="checkbox"/>	Ansible-Slave2	i-0e09954f95bed0bc3	Running	t2.micro	-	View alarms	us-east-1a	ec2-	

3. Go to Key-Pair Path

```
C:\WINDOWS\system32\cmd. X + v

Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.

C:\Users\shaik>cd C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs
C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs>
```

4. Connected to Ansible-Master and Slave1 and Slave2, But According to Assignment Task we Need 5 Instances

```
ubuntu@Ansible-Master: ~ X Command Prompt X Command Prompt X

ubuntu@Ansible-Master:~$
```

5. I am Launching Another 2 Instances and Named as Ansible-Slave3 and Slave4

Instances (11) Info

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

<1>

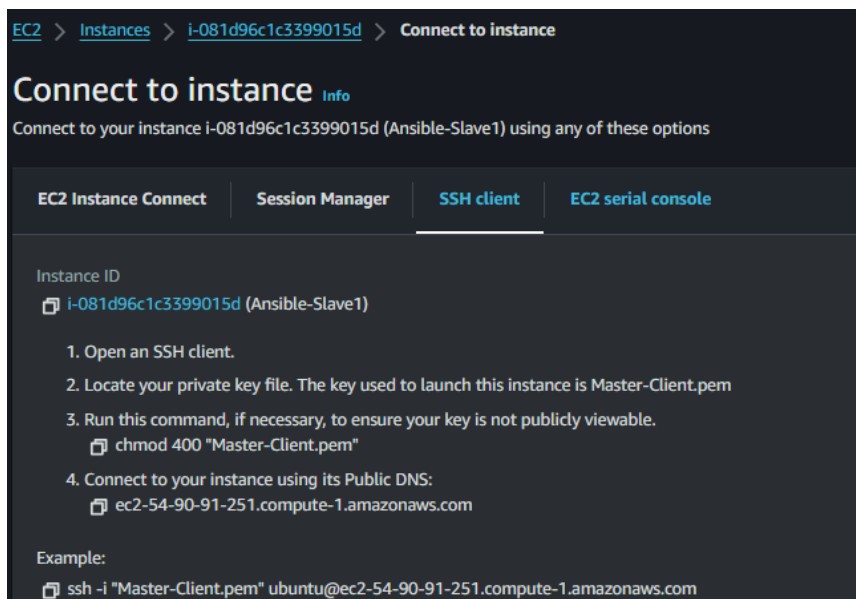
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	Ansible-Master	i-0810c56f8a6550852	<div>Running</div>	t2.micro	<div>Initializing</div>	<div>View alarms</div>	us-east-1a
<input type="checkbox"/>	Ansible-Slave1	i-081d96c1c3399015d	<div>Running</div>	t2.micro	<div>Initializing</div>	<div>View alarms</div>	us-east-1a
<input type="checkbox"/>	Ansible-Slave2	i-0e09954f95bed0bc3	<div>Running</div>	t2.micro	<div>Initializing</div>	<div>View alarms</div>	us-east-1a
<input type="checkbox"/>	Ansible-Slave3	i-0a1097c86cdd12efc	<div>Running</div>	t2.micro	<div>Initializing</div>	<div>View alarms</div>	us-east-1a
<input type="checkbox"/>	Ansible-Slave4	i-04c145c6d8a6ffbb1	<div>Running</div>	t2.micro	<div>Initializing</div>	<div>View alarms</div>	us-east-1a

6. See the 5 Instances logged successfully and change host name as per ur need

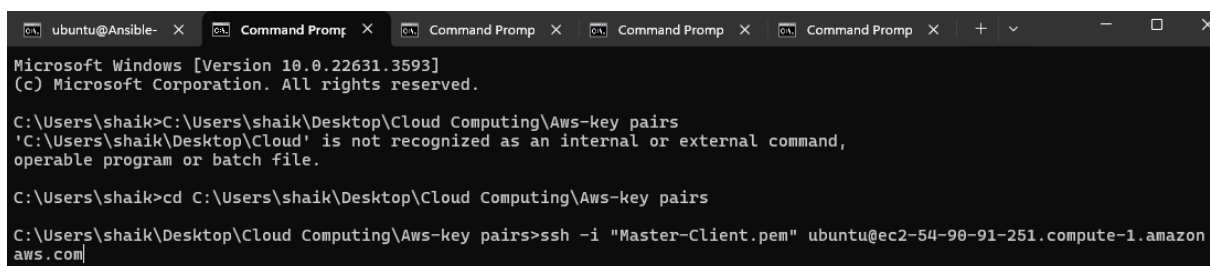
```
ubuntu@Ansible-Master: ~ X Command Prompt X Command Prompt X Command Prompt X Command Prompt X

ubuntu@Ansible-Master:~$
```

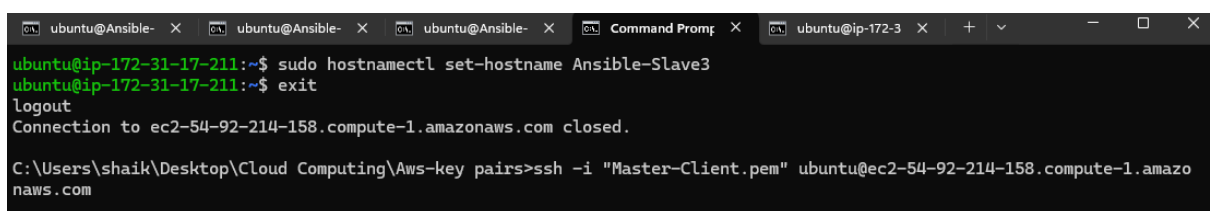
7. Go **Instances > connect > SSH Client** copy the path



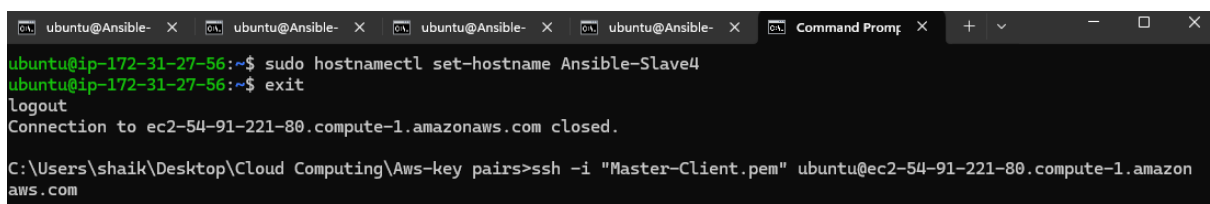
8. paste in **key-pair location** and connect this is slave1



9. Connect like all instances, **Now connecting slave3**, Change the Hostname as your wish



10. same as slave4



11. Go **Ansible-Master** and Check the IP's and Change it

```
ubuntu@Ansible-Master:~$ cat inventory
```

12. This is **Past Ip's** and Now we need to Modify it

```
ubuntu@Ansible-Master:~$ cat inventory
[slave1]
54.161.39.163 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem

[slave2]
18.214.98.150 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem
ubuntu@Ansible-Master:~$
```

13. Copying **slave1 Public IP**

<input checked="" type="checkbox"/>	Ansible-Slave1	i-081d96c1c3399015d	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a
<input type="checkbox"/>	Ansible-Slave2	i-0e09954f95bed0bc3	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a
<input type="checkbox"/>	Ansible-Slave3	i-0a1097c86cdd12efc	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a

i-081d96c1c3399015d (Ansible-Slave1)	
Details	Status and alarms New
▼ Instance summary Info	
Instance ID i-081d96c1c3399015d (Ansible-Slave1)	Public IPv4 address 54.90.91.251 open address

14. Open **file in editor**

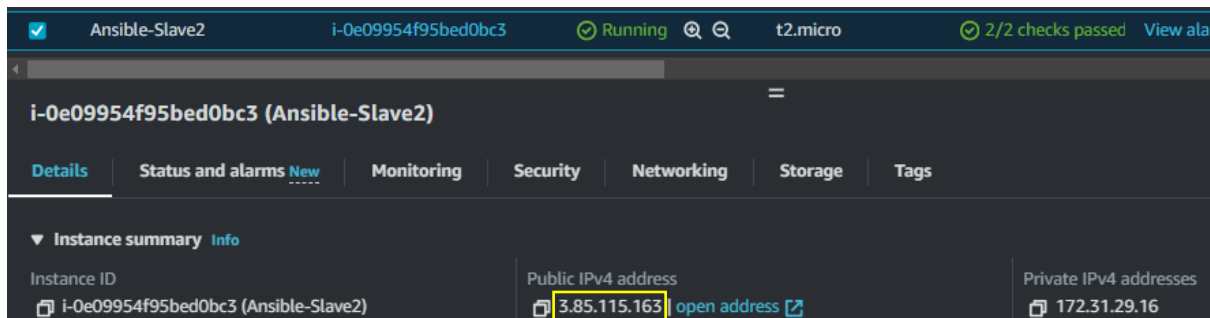
```
ubuntu@Ansible-Master:~$ nano inventory
```

15. Paste Copied IP of Slave1

```
GNU nano 7.2 inventory *
[test]
54.90.91.251 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem

[prod]
```

16. Copying slave2 Public IP



Ansible-Slave2 i-0e09954f95bed0bc3 Running t2.micro 2/2 checks passed View alarms

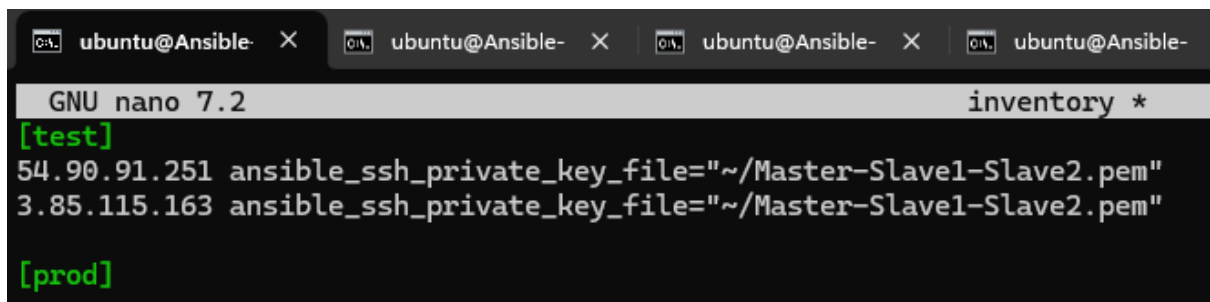
i-0e09954f95bed0bc3 (Ansible-Slave2)

Details Status and alarms New Monitoring Security Networking Storage Tags

▼ Instance summary Info

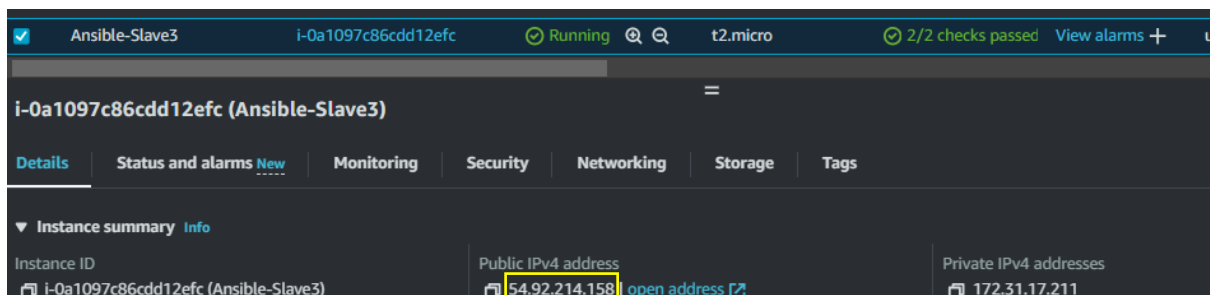
Instance ID	Public IPv4 address	Private IPv4 addresses
i-0e09954f95bed0bc3 (Ansible-Slave2)	3.85.115.163 open address	172.31.29.16

17. Paste Copied IP of Slave2



```
GNU nano 7.2 inventory *
[test]
54.90.91.251 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem"
3.85.115.163 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem"
[prod]
```

18. Copying slave3 Public IP



Ansible-Slave3 i-0a1097c86cdd12efc Running t2.micro 2/2 checks passed View alarms +

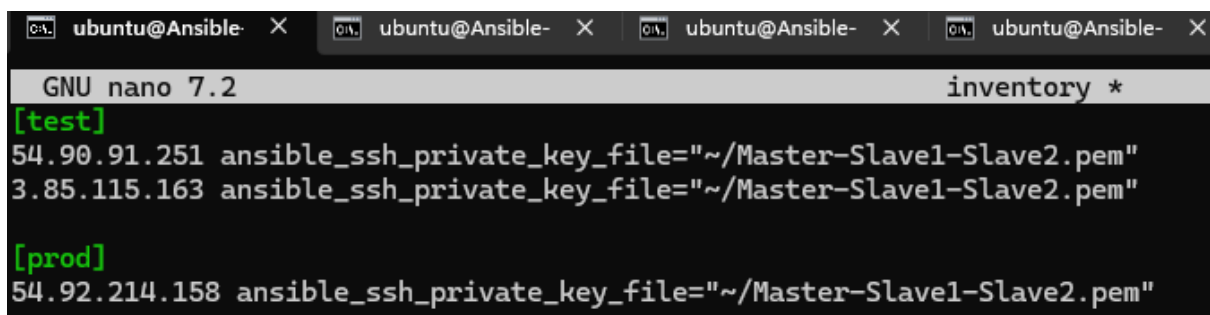
i-0a1097c86cdd12efc (Ansible-Slave3)

Details Status and alarms New Monitoring Security Networking Storage Tags

▼ Instance summary Info

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0a1097c86cdd12efc (Ansible-Slave3)	54.92.214.158 open address	172.31.17.211

19. Paste Copied IP of Slave3



```
GNU nano 7.2 inventory *
[test]
54.90.91.251 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem"
3.85.115.163 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem"
[prod]
54.92.214.158 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem"
```

20. Copying slave 4 Public IP

The screenshot shows the AWS Management Console for an EC2 instance named 'i-04c145c6d8a6ffbb1 (Ansible-Slave4)'. The instance is in the 'Running' state. The 'Public IPv4 address' is highlighted as 54.91.221.80. The 'Private IPv4 addresses' are listed as 172.31.27.56.

21. Paste Copied IP of Slave4

```
GNU nano 7.2 inventory *
[test]
54.90.91.251 ansible_ssh_private_key_file=~/.ssh/Ansible-Slave2.pem
3.85.115.163 ansible_ssh_private_key_file=~/.ssh/Ansible-Slave2.pem

[prod]
54.92.214.158 ansible_ssh_private_key_file=~/.ssh/Ansible-Slave2.pem
54.91.221.80 ansible_ssh_private_key_file=~/.ssh/Ansible-Slave2.pem
```

22. I am pinging but sometimes, it showing error only first IP going to be success and other will be failed.

```
ubuntu@Ansible-Master:~$ ansible all -i inventory -m ping
The authenticity of host '3.85.115.163 (3.85.115.163)' can't be established.
ED25519 key fingerprint is SHA256:lbiygpYdrhyWoPKKLIsqIFNW/GiZj+mu+OEAnE9JuyI.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [hashed name]
  ~/.ssh/known_hosts:8: [hashed name]
  ~/.ssh/known_hosts:16: [hashed name]
The authenticity of host '54.92.214.158 (54.92.214.158)' can't be established.
ED25519 key fingerprint is SHA256:sFSC07HDLQruNqaNsKPCCdT/ekxu2KD69dRD0eSjL8k.
This key is not known by any other names.
The authenticity of host '54.91.221.80 (54.91.221.80)' can't be established.
ED25519 key fingerprint is SHA256:u6hPi0IZvPsbJccgf1bCbshCvW8lk0YxQKqhb62ho0A.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? 54.90.91.251 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
3.85.115.163 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: Host key verification failed.",
  "unreachable": true
}
54.92.214.158 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: Host key verification failed.",
  "unreachable": true
}
54.91.221.80 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: Host key verification failed.",
  "unreachable": true
}
```

Manually Update Known Hosts

1. **Remove the offending host entries:** You can manually remove the old or offending entries from your `~/.ssh/known_hosts` file. This can be done by opening the file and deleting the lines corresponding to the problematic hosts.
2. **Automatically accept the new host keys:** You can use the `ssh-keyscan` command to fetch and add the host keys to your `known_hosts` file. For example:

```
ssh-keyscan -H 3.85.115.163 >> ~/.ssh/known_hosts
ssh-keyscan -H 54.92.214.158 >> ~/.ssh/known_hosts
ssh-keyscan -H 54.91.221.80 >> ~/.ssh/known_hosts
```

22. So I am Manually Scanning the keys and its ran successfully

```
ubuntu@Ansible-Master: ~$ ssh-keyscan -H 3.85.115.163 >> ~/.ssh/known_hosts
ssh-keyscan -H 54.92.214.158 >> ~/.ssh/known_hosts
ssh-keyscan -H 54.91.221.80 >> ~/.ssh/known_hosts
# 3.85.115.163:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13
# 3.85.115.163:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13
# 3.85.115.163:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13
# 3.85.115.163:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13
# 3.85.115.163:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13
# 54.92.214.158:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13
# 54.92.214.158:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13
# 54.92.214.158:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13
# 54.92.214.158:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13
# 54.92.214.158:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13
# 54.91.221.80:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13
# 54.91.221.80:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13
# 54.91.221.80:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13
# 54.91.221.80:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13
# 54.91.221.80:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13
ubuntu@Ansible-Master:~$ ansible all -i inventory -m ping
54.90.91.251 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
54.91.221.80 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
54.92.214.158 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

23. cat view

```
ubuntu@Ansible-Master: ~$ cat inventory
[test]
54.90.91.251 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem
3.85.115.163 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem

[prod]
54.92.214.158 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem
54.91.221.80 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem
```

24. Now for 2 Nodes we have to install java and 2 nodes we have install mysql

```
ubuntu@Ansible-Master: ~$ ansible-galaxy init install_java
- Role install_java was created successfully
ubuntu@Ansible-Master: ~$ ansible-galaxy init install_mysql
- Role install_mysql was created successfully
ubuntu@Ansible-Master: ~$ ls
Ansible_install.sh  add_text.sh  double-role.yml  install_mysql  nginx  setup.yml
Master-Slave1-Slave2.pem  apache2  install_java  inventory  run_script.yml  uninstall.yml
ubuntu@Ansible-Master: ~$
```

25. tree view

```
ubuntu@Ansible-Master: ~$ cd install_java
ubuntu@Ansible-Master: ~/install_java$ ls
README.md  defaults  files  handlers  meta  tasks  templates  tests  vars
ubuntu@Ansible-Master: ~/install_java$ tree
.
├── README.md
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml

9 directories, 8 files
ubuntu@Ansible-Master: ~/install_java$
```

26. Go to Tasks Edit main file

```
ubuntu@Ansible-Master: ~/install_java$ ls tasks
main.yml
ubuntu@Ansible-Master: ~/install_java$ nano main.yml
```


27. So Updating the Package lists and Name: Install java and state: present

```
ubuntu@Ansible-Master: ~ X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~ X ubuntu@
GNU nano 7.2 main.yml *
---
# tasks file for install_java
- name: Update package lists
  apt:
    update_cache: yes
- name: Install Java
  apt:
    name: default-jdk
    state: present
```

28. Cat View

```
ubuntu@Ansible-Master: ~ X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~
ubuntu@Ansible-Master:~/install_java/tasks$ cat main.yml
---
# tasks file for install_java
- name: Update package lists
  apt:
    update_cache: yes
- name: Install Java
  apt:
    name: default-jdk
    state: present
ubuntu@Ansible-Master:~/install_java/tasks$
```

29. Now Go Mysql Role and tree view

```
ubuntu@Ansible-Master: ~ X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~ X ubuntu@Ansible-Slave3: ~ X
ubuntu@Ansible-Master:~$ cd install_mysql
ubuntu@Ansible-Master:~/install_mysql$ ls
README.md defaults files handlers meta tasks templates tests vars
ubuntu@Ansible-Master:~/install_mysql$ tree
.
├── README.md
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml

9 directories, 8 files
ubuntu@Ansible-Master:~/install_mysql$
```

30. go to tasks edit main.yml file

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Master:~/install_mysql$ cd tasks  
ubuntu@Ansible-Master:~/install_mysql/tasks$ ls  
main.yml  
ubuntu@Ansible-Master:~/install_mysql/tasks$ nano main.yml
```

31. So Updating the Package lists and Name: Install mysql-server and state: present

```
GNU nano 7.2 main.yml *  
---  
# tasks file for installing MySQL server  
  
- name: Update package lists  
  apt:  
    update_cache: yes  
  
- name: Install MySQL server  
  apt:  
    name: mysql-server  
    state: present
```

32. Cat View

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Master:~/install_mysql/tasks$ cat main.yml  
---  
# tasks file for installing MySQL server  
  
- name: Update package lists  
  apt:  
    update_cache: yes  
  
- name: Install MySQL server  
  apt:  
    name: mysql-server  
    state: present  
  
ubuntu@Ansible-Master:~/install_mysql/tasks$
```

33. Modify the double-role.yml file it is a play book we are replacing the roles

```
ubuntu@Ansible-Master: ~  
ubuntu@Ansible-Master:~$ ls  
Ansible_install.sh  add_text.sh  double-role.yml  install_mysql  nginx  setup.yml  
Master-Slave1-Slave2.pem  apache2  install_java  inventory  run_script.yml  uninstall.yml  
ubuntu@Ansible-Master:~$ nano double-role.yml
```

34. According to task Install java on test host and install_mysql on prod host

```
ubuntu@Ansible-Master: ~
ubuntu@Ansible-Slave1: ~
ubuntu@Ansible-Slave2: ~
ubuntu@Ansible-Slave3: ~

GNU nano 7.2 double-role.yml *
---
- hosts: test
  become: yes
  roles:
    - install_java

- hosts: prod
  become: yes
  roles:
    - install_mysql
```

35. Check the inventory and playbook

```
ubuntu@Ansible-Master: ~
ubuntu@Ansible-Slave1: ~
ubuntu@Ansible-Slave2: ~
ubuntu@Ansible-Slave3: ~

ubuntu@Ansible-Master:~$ ls
Ansible_install.sh  add_text.sh  double-role.yml  install_mysql  nginx  setup.yml
Master-Slave1-Slave2.pem  apache2  install_java  inventory  run_script.yml  uninstall.yml
ubuntu@Ansible-Master:~$ cat inventory
[test]
54.90.91.251 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem
3.85.115.163 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem

[prod]
54.92.214.158 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem
54.91.221.80 ansible_ssh_private_key_file=~/.Master-Slave1-Slave2.pem
ubuntu@Ansible-Master:~$ cat double-role.yml
---
- hosts: test
  become: yes
  roles:
    - install_java

- hosts: prod
  become: yes
  roles:
    - install_mysql
ubuntu@Ansible-Master:~$
```

36. syntax check

```
ubuntu@Ansible-Master: ~
ubuntu@Ansible-Slave1: ~
ubuntu@Ansible-Slave2: ~
ubuntu@Ansible-Slave3: ~

ubuntu@Ansible-Master:~$ ansible-playbook -i inventory double-role.yml --syntax -check

playbook: double-role.yml
ubuntu@Ansible-Master:~$
```

37. Performed the tasks and its executed successfully

```
ubuntu@Ansible-Master: ~$ ansible-playbook -i inventory double-role.yml

PLAY [test] *****

TASK [Gathering Facts] *****
ok: [54.90.91.251]
ok: [3.85.115.163]

TASK [install_java : Update package lists] *****
changed: [54.90.91.251]
changed: [3.85.115.163]

TASK [install_java : Install Java] *****
changed: [54.90.91.251]
changed: [3.85.115.163]

PLAY [prod] *****

TASK [Gathering Facts] *****
ok: [54.92.214.158]
ok: [54.91.221.80]

TASK [install_mysql : Update package lists] *****
changed: [54.92.214.158]
changed: [54.91.221.80]

TASK [install_mysql : Install MySQL server] *****
changed: [54.92.214.158]
changed: [54.91.221.80]

PLAY RECAP *****
3.85.115.163      : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
54.90.91.251     : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
54.91.221.80    : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
54.92.214.158   : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@Ansible-Master:~$
```

38. check in slave1, its installed java successfully

```
ubuntu@Ansible-Slave1:~$ java --version
openjdk 21.0.3 2024-04-16
OpenJDK Runtime Environment (build 21.0.3+9-Ubuntu-1ubuntu1)
OpenJDK 64-Bit Server VM (build 21.0.3+9-Ubuntu-1ubuntu1, mixed mode, sharing)
ubuntu@Ansible-Slave1:~$
```

39. Check in slave2, its installed java successfully

```
ubuntu@Ansible-Slave2:~$ java --version
openjdk 21.0.3 2024-04-16
OpenJDK Runtime Environment (build 21.0.3+9-Ubuntu-1ubuntu1)
OpenJDK 64-Bit Server VM (build 21.0.3+9-Ubuntu-1ubuntu1, mixed mode, sharing)
ubuntu@Ansible-Slave2:~$
```

40. Check in slave3, its installed mysql successfully

```
ubuntu@Ansible-Slave3:~$ systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
   Active: active (running) since Fri 2024-05-31 06:53:34 UTC; 6min ago
     Process: 3018 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
    Main PID: 3027 (mysqld)
      Status: "Server is operational"
     Tasks: 37 (limit: 1130)
    Memory: 352.1M (peak: 367.3M)
       CPU: 2.250s
    CGroup: /system.slice/mysql.service
            └─3027 /usr/sbin/mysqld

May 31 06:53:32 Ansible-Slave3 systemd[1]: Starting mysql.service - MySQL Community Server...
May 31 06:53:34 Ansible-Slave3 systemd[1]: Started mysql.service - MySQL Community Server.
ubuntu@Ansible-Slave3:~$
```

41. check in slave4, its installed mysql successfully

```
ubuntu@Ansible-Master: ~ X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~ X ubuntu@Ansible-Slave3: ~ X ubuntu@Ansible-Slave4: ~ X
ubuntu@Ansible-Slave4:~$ systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
   Active: active (running) since Fri 2024-05-31 06:53:39 UTC; 6min ago
     Process: 2985 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
    Main PID: 2993 (mysqld)
      Status: "Server is operational"
        Tasks: 37 (limit: 1130)
      Memory: 353.1M (peak: 376.0M)
         CPU: 2.550s
       CGroup: /system.slice/mysql.service
              └─2993 /usr/sbin/mysqld

May 31 06:53:37 Ansible-Slave4 systemd[1]: Starting mysql.service - MySQL Community Server...
May 31 06:53:39 Ansible-Slave4 systemd[1]: Started mysql.service - MySQL Community Server.
ubuntu@Ansible-Slave4:~$
```

Optional Task:

42. Uninstalling the servers

```
ubuntu@Ansible-Master: ~ X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~ X ubuntu@Ansible-Slave3: ~ X ubuntu@Ansible-Slave4: ~ X
ubuntu@Ansible-Master:~$ nano uninstall.yml
```

43. Using Ansible Playbook uninstalling the java and mysql server

```
GNU nano 7.2                                uninstall.yml *
- name: Uninstall Java on test
  hosts: test
  become: true
  tasks:
    - name: Uninstall Java
      apt:
        name: default-jdk
        state: absent
        purge: yes
        autoremove: yes

- name: Uninstall MySQL Server on prod
  hosts: prod
  become: true
  tasks:
    - name: Uninstall MySQL server
      apt:
        name: mysql-server
        state: absent
        purge: yes
        autoremove: yes
```

44. syntax check

```
ubuntu@Ansible-Master: ~ X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~ X ubuntu@Ansible-Slave3: ~ X ubuntu@Ansible-Slave4: ~ X
ubuntu@Ansible-Master:~$ nano uninstall.yml
ubuntu@Ansible-Master:~$ ansible-playbook -i inventory uninstall.yml --syntax -check

playbook: uninstall.yml
ubuntu@Ansible-Master:~$
```

45. successfully executed

```
ubuntu@Ansible-Master: ~$ ansible-playbook -i inventory uninstall.yml

PLAY [Uninstall Java on test] *****

TASK [Gathering Facts] *****
ok: [54.90.91.251]
ok: [3.85.115.163]

TASK [Uninstall Java] *****
changed: [3.85.115.163]
changed: [54.90.91.251]

PLAY [Uninstall MySQL Server on prod] *****

TASK [Gathering Facts] *****
ok: [54.92.214.158]
ok: [54.91.221.80]

TASK [Uninstall MySQL server] *****
changed: [54.92.214.158]
changed: [54.91.221.80]

PLAY RECAP *****
3.85.115.163      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
54.90.91.251     : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
54.91.221.80     : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
54.92.214.158   : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@Ansible-Master:~$
```

46. Now, There is No Java on Slave1

```
ubuntu@Ansible-Slave1:~$ java --version
Command 'java' not found, but can be installed with:
sudo apt install default-jre          # version 2:1.17-75, or
sudo apt install openjdk-17-jre-headless # version 17.0.10~6ea-1
sudo apt install openjdk-11-jre-headless # version 11.0.21+9-0ubuntu1
sudo apt install openjdk-19-jre-headless # version 19.0.2+7-4
sudo apt install openjdk-20-jre-headless # version 20.0.2+9-1
sudo apt install openjdk-21-jre-headless # version 21.0.1+12-3
sudo apt install openjdk-22-jre-headless # version 22~22ea-1
sudo apt install openjdk-8-jre-headless  # version 8u392-ga-1
ubuntu@Ansible-Slave1:~$
```

47. Now, There is No Java on Slave2

```
ubuntu@Ansible-Slave2:~$ java --version
Command 'java' not found, but can be installed with:
sudo apt install default-jre          # version 2:1.17-75, or
sudo apt install openjdk-17-jre-headless # version 17.0.10~6ea-1
sudo apt install openjdk-11-jre-headless # version 11.0.21+9-0ubuntu1
sudo apt install openjdk-19-jre-headless # version 19.0.2+7-4
sudo apt install openjdk-20-jre-headless # version 20.0.2+9-1
sudo apt install openjdk-21-jre-headless # version 21.0.1+12-3
sudo apt install openjdk-22-jre-headless # version 22~22ea-1
sudo apt install openjdk-8-jre-headless  # version 8u392-ga-1
ubuntu@Ansible-Slave2:~$
```

48. Now, There is No MySQL on Slave3

```
ubuntu@Ansible-Master: ~ X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~ X ubuntu@Ansible-Slave3: ~ X
ubuntu@Ansible-Slave3:~$ systemctl status mysql
Unit mysql.service could not be found.
ubuntu@Ansible-Slave3:~$
```

49. Now, There is No MySQL on Slave4

```
ubuntu@Ansible-Master: ~ X ubuntu@Ansible-Slave1: ~ X ubuntu@Ansible-Slave2: ~ X ubuntu@Ansible-Slave3: ~ X ubuntu@Ansible-Slave4: ~
ubuntu@Ansible-Slave4:~$ systemctl status mysql
Unit mysql.service could not be found.
ubuntu@Ansible-Slave4:~$
```