

Ansible

Assignment - 1



















Tasks To Be Performed:

1. Setup Ansible cluster with 3 nodes
2. On slave 1 install Java
3. On slave 2 install MySQL server

Do the above tasks using Ansible Playbooks

Solutions:

Launch 3 instances

Name 	Instance ID	Instance state 	Instance type 	Status check	Alarm status	Availability
ansible-master	i-0eef2307b21dc9b6f	 Running  	t2.micro	 Initializing	View alarms 	ap-southe
ansible-slave1	i-010bce9e31481d143	 Running  	t2.micro	 Initializing	View alarms 	ap-southe
ansible-slave2	i-0c4fc5810a9ed8aec	 Running  	t2.micro	 Initializing	View alarms 	ap-southe

Connect and update all the 3 instances

Ansible master node

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.
```

```
To run a command as administrator (user "root"), use "sudo <command>". See "man sudo_root" for details.
```

```
ubuntu@ip-172-31-20-121:~$
```

i-0eef2307b21dc9b6f (ansible-master)

PublicIPs: 54.251.90.128 PrivateIPs: 172.31.20.121

Ansible slave 1

```
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.
```

```
ubuntu@ip-172-31-21-228:~$
```

i-010bce9e31481d143 (ansible-slave1)

PublicIPs: 13.229.115.186 PrivateIPs: 172.31.21.228

Ansible slave 2

```
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.
```

```
ubuntu@ip-172-31-29-200:~$
```

i-0c4fc5810a9ed8aec (ansible-slave2)

PublicIPs: 18.142.231.118 PrivateIPs: 172.31.29.200

Create Sh file for installation of ansible(Install ansible only on master server)

```
ubuntu@ip-172-31-20-121:~$ sudo nano a.sh  
ubuntu@ip-172-31-20-121:~$ bash a.sh  
Hit:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:2 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:3 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
49 packages can be upgraded. Run 'apt list --upgradable' to see them.  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
software-properties-common is already the newest version (0.99.22.9).  
software-properties-common set to manually installed.  
0 upgraded, 0 newly installed, 0 to remove and 49 not upgraded.
```

Check the version and confirm the ansible installed

```
ubuntu@ip-172-31-20-121:~$ ansible --version
ansible [core 2.16.7]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ubuntu/.ansible/plugins/modul
  ansible python module location = /usr/lib/python3/dist-packages/ansib
  ansible collection location = /home/ubuntu/.ansible/collections:/usr/
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] (
  jinja version = 3.0.3
  libyaml = True
ubuntu@ip-172-31-20-121:~$
```

We have to setup a cluster so we need to generate a key using command: ssh-keygen. It will generate 2 keys, public and private keys.

```
ubuntu@ip-172-31-20-121:~$ 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:eAchkp4fBbDUTaiOkpM4ROdo5JCCRd6mf4GXy2VCD8A ubuntu@ip-172-31-20-121
The key's randomart image is:
+---[RSA 3072]-----+
|.to.=+o=o          |
|=+ +E+o.o.         |
|* =.+oo..          |
| = ++o.= .         |
|+o.o..*.S .        |
|B ...o.B .         |
| + . +             |
|                   |
|                   |
+----[SHA256]-----+
ubuntu@ip-172-31-20-121:~$
```

cd. ssh, check the keys with the list command

```
ubuntu@ip-172-31-20-121:~$ cd .ssh
ubuntu@ip-172-31-20-121:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
ubuntu@ip-172-31-20-121:~/.ssh$ sudo cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
```

Sudo cat id_rsa.pub – will give you the public key token, which you need to copy to the slaves.

```
ubuntu@ip-172-31-20-121:~/.ssh$ sudo cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGC+UQvLSa1+DglzNPuBvuietv2cRktqq49l5yAULIn9sjz730fv/rkwc2
CuACnsFA2ncRlTOZOWXVr387+6K3M8aUq9LgRuoY6AjPK6I8BQn7xAYYt86QrNlEx9EgPcoItuu+VvzLgF920i7rd7SIk3
0A9QTfuGXjdeXQ7Rn2fD3tqEZq8j2A/ntG9nksFPO889Lf52dBHH05Nsy5UalAuFJzhMj61CJmVV226prOTcDEAZ5mvNk0
B2/7ec5e2F4/e9aywCIJGfqOq1F6TaxoT+HcOrlf9p2mnHA95NAwxXpYmYjWCTgKFmc9DqhABsHisCd7UIy3g2WSA/XKR8
-121
ubuntu@ip-172-31-20-121:~/.ssh$
```

Go to slaves and go to ssh location you will see authorized_keys. Inside that you need to paste the token.

```
ubuntu@ip-172-31-21-228:~$ cd .ssh
ubuntu@ip-172-31-21-228:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-21-228:~/.ssh$ sudo nano authorized_keys
```

Now edit the host file in master node.

```
ubuntu@ip-172-31-20-121:~/.ssh$ cd /etc/ansible/
ubuntu@ip-172-31-20-121:/etc/ansible$ ls
ansible.cfg  hosts  roles
ubuntu@ip-172-31-20-121:/etc/ansible$ sudo nano hosts
```

Add the private ip of both the slaves

```
aws Services Search
GNU nano 6.2
[group]
slave1 ansible_host=172.31.21.228
slave2 ansible_host=172.31.29.200
```

Now you need to make a ping request for slaves to connect within the cluster.

\$ ansible -m ping all

It will reflect ping : pong

```
ubuntu@ip-172-31-20-121:/etc/ansible$ ansible -m ping all
The authenticity of host '172.31.21.228 (172.31.21.228)' can't be established.
ED25519 key fingerprint is SHA256:/HOa+BfjnLe6zfXfyFGpjGwNegAE7UWziSTH2x9tWMS.
This key is not known by any other names
The authenticity of host '172.31.29.200 (172.31.29.200)' can't be established.
ED25519 key fingerprint is SHA256:r62HWhwPWTFysyFJF0Ghla09hF8vx2lzeipniZvgm5c.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
slave1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
yes
slave2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu@ip-172-31-20-121:/etc/ansible$
```

Now write a playbook for installing java and mysql on slaves

\$ sudo nano play1.yml

```
GNU nano 6.2                                                                    P.
---
- name: executing task on slave1
  hosts: slave1
  become: true
  tasks:
    - name: installing java
      apt: name=openjdk-11-jdk state=present
- name: executing task on slave2
  hosts: slave2
  become: true
  tasks:
    - name: installing mysql on slave2
      apt: name=mysql-server state=present
```

Do the syntax check first and do the dry run before the actual run.

```
ubuntu@ip-172-31-20-121:~$ ansible-playbook play1.yml --syntax-check
playbook: play1.yml
ubuntu@ip-172-31-20-121:~$ ansible-playbook play1.yml --check

PLAY [executing task on slave1] *****

TASK [Gathering Facts] *****
ok: [slave1]

TASK [installing java] *****
changed: [slave1]

PLAY [executing task on slave2] *****

TASK [Gathering Facts] *****
ok: [slave2]

TASK [installing mysql on slave2] *****
changed: [slave2]

PLAY RECAP *****
slave1                : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
slave2                : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Check the java version on slave1 to confirm the installation.

```
ubuntu@ip-172-31-21-228:~$ java --version
openjdk 11.0.23 2024-04-16
OpenJDK Runtime Environment (build 11.0.23+9-post-Ubuntu-1ubuntu122.04.1)
OpenJDK 64-Bit Server VM (build 11.0.23+9-post-Ubuntu-1ubuntu122.04.1, mixed mode, sharing)
ubuntu@ip-172-31-21-228:~$
```

Check the status of mysql server on slave2 for confirmation.

```
ubuntu@ip-172-31-29-200:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2024-06-16 17:14:57 UTC; 4min 1s ago
     Process: 3056 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
    Main PID: 3064 (mysqld)
      Status: "Server is operational"
        Tasks: 37 (limit: 1121)
      Memory: 354.1M
         CPU: 1.950s
    CGroup: /system.slice/mysql.service
            └─3064 /usr/sbin/mysqld

Jun 16 17:14:56 ip-172-31-29-200 systemd[1]: Starting MySQL Community Server...
Jun 16 17:14:57 ip-172-31-29-200 systemd[1]: Started MySQL Community Server.
ubuntu@ip-172-31-29-200:~$
```

Assignment – 2

Tasks To Be Performed:

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

Solutions:

We need to create script in /tmp file location

```
ubuntu@ip-172-31-20-121:~$ sudo nano b.sh
ubuntu@ip-172-31-20-121:~$
```

Write a script followed by the locations, save and exit

```
GNU nano 6.2
echo "This text has been added by custom script" >/tmp/1.txt
```

Create a playbook

```
ubuntu@ip-172-31-20-121:~$ sudo nano play2.yml
```

Write a playbook for executing shell script which is written

```
---
- name: executing script on slaves
  hosts: all
  become: true
  tasks:
    - name: executing b.sh script n slaves
      script: b.sh
```

Check the indentations.

```
ubuntu@ip-172-31-20-121:~$ sudo cat b.sh
echo "This text has been added by custommm script" >/tmp/1.txt
ubuntu@ip-172-31-20-121:~$ sudo cat play2.yml
---
- name: executing script on slaves
  hosts: all
  become: true
  tasks:
    - name: executing b.sh script n slaves
      script: b.sh
```

Do the syntax check

```
ubuntu@ip-172-31-20-121:~$ ansible-playbook play2.yml --syntax-check
playbook: play2.yml
```

Do the dry run

```
ubuntu@ip-172-31-20-121:~$ ansible-playbook play2.yml --check
PLAY [executing script on slaves] *****

TASK [Gathering Facts] *****
ok: [slave1]
ok: [slave2]

TASK [executing b.sh script n slaves] *****
skipping: [slave1]
skipping: [slave2]

PLAY RECAP *****
slave1      : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
slave2      : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Do the actual run

```
ubuntu@ip-172-31-20-121:~$ ansible-playbook play2.yml
PLAY [executing script on slaves] *****

TASK [Gathering Facts] *****
ok: [slave1]
ok: [slave2]

TASK [executing b.sh script n slaves] *****
changed: [slave1]
changed: [slave2]

PLAY RECAP *****
slave1      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
slave2      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu@ip-172-31-20-121:~$
```


Go to slave 1

Go inside tmp file location → Cd /tmp/1.txt

You can see the shell script written on master.

```
ubuntu@ip-172-31-29-200:~$ cd /tmp/  
ubuntu@ip-172-31-29-200:/tmp$ ls  
1.txt  
snap-private-tmp  
systemd-private-95588d522c2e4d80913e6694c1dcd55b-chrony.ser  
systemd-private-95588d522c2e4d80913e6694c1dcd55b-systemd-ld  
systemd-private-95588d522c2e4d80913e6694c1dcd55b-systemd-re  
tmp.Gj872Enviw  
ubuntu@ip-172-31-29-200:/tmp$ cat 1.txt  
This text has been added by customm script  
ubuntu@ip-172-31-29-200:/tmp$
```

Similarly check on Slave2

```
ubuntu@ip-172-31-29-200:~$ cd /tmp/  
ubuntu@ip-172-31-29-200:/tmp$ ls  
1.txt  
snap-private-tmp  
systemd-private-95588d522c2e4d80913e6694c1dcd55b-chrony.ser  
systemd-private-95588d522c2e4d80913e6694c1dcd55b-systemd-ld  
systemd-private-95588d522c2e4d80913e6694c1dcd55b-systemd-re  
tmp.Gj872Enviw  
ubuntu@ip-172-31-29-200:/tmp$ cat 1.txt  
This text has been added by customm script  
ubuntu@ip-172-31-29-200:/tmp$
```

Assignment - 3

Tasks To Be Performed:

1. Create 2 Ansible roles
2. Install Apache2 on slave1 using one role and NGINX on slave2 using the other role
3. Above should be implemented using different Ansible roles

Solutions:

To create roles for apache2 & nginx - go to /etc/ansible/ location

sudo ansible-galaxy init apache2

```
ubuntu@ip-172-31-20-121:~$ cd /etc/ansible/
ubuntu@ip-172-31-20-121:/etc/ansible$ ls
ansible.cfg  hosts  roles
ubuntu@ip-172-31-20-121:/etc/ansible$ cd roles
ubuntu@ip-172-31-20-121:/etc/ansible/roles$ ls
ubuntu@ip-172-31-20-121:/etc/ansible/roles$ sudo ansible-galaxy init apache2
- Role apache2 was created successfully
ubuntu@ip-172-31-20-121:/etc/ansible/roles$ ls
apache2
```

Sudo ansible-galaxy init nginx

```
ubuntu@ip-172-31-20-121:/etc/ansible/roles$ sudo ansible-galaxy init nginx
- Role nginx was created successfully
ubuntu@ip-172-31-20-121:/etc/ansible/roles$ ls
apache2  nginx
ubuntu@ip-172-31-20-121:/etc/ansible/roles$
```

Go to task location create install.yml file

```
ubuntu@ip-172-31-20-121:/etc/ansible/roles$ cd apache2
ubuntu@ip-172-31-20-121:/etc/ansible/roles/apache2$ ls
README.md  defaults  files  handlers  meta  tasks  templates  tests  vars
ubuntu@ip-172-31-20-121:/etc/ansible/roles/apache2$ cd tasks/
ubuntu@ip-172-31-20-121:/etc/ansible/roles/apache2/tasks$ ls
main.yml
ubuntu@ip-172-31-20-121:/etc/ansible/roles/apache2/tasks$ sudo nano install.yml
```

Write a file to install apache2 server

```
---  
- name: installing apache2 on slave1  
  apt: name=apache2 state=latest
```

Include the task on main.yml file

```
ubuntu@ip-172-31-20-121:/etc/ansible/roles/apache2/tasks$ sudo nano main.yml  
ubuntu@ip-172-31-20-121:/etc/ansible/roles/apache2/tasks$ sudo cat main.yml  
---  
# tasks file for apache2  
- include_tasks: install.yml
```

Similarly write install.yml

```
ubuntu@ip-172-31-20-121:/etc/ansible/roles$ ls  
apache2  nginx  
ubuntu@ip-172-31-20-121:/etc/ansible/roles$ cd nginx  
ubuntu@ip-172-31-20-121:/etc/ansible/roles/nginx$ ls  
README.md  defaults  files  handlers  meta  tasks  templates  tests  vars  
ubuntu@ip-172-31-20-121:/etc/ansible/roles/nginx$ cd tasks  
ubuntu@ip-172-31-20-121:/etc/ansible/roles/nginx/tasks$ sudo nano install.yml  
ubuntu@ip-172-31-20-121:/etc/ansible/roles/nginx/tasks$ sudo cat install.yml  
---  
- name: installing nginx on slave2  
  apt: name=nginx state=present  
ubuntu@ip-172-31-20-121:/etc/ansible/roles/nginx/tasks$
```

main.yml file

```
ubuntu@ip-172-31-20-121:/etc/ansible/roles/nginx/tasks$ ls  
install.yml  main.yml  
ubuntu@ip-172-31-20-121:/etc/ansible/roles/nginx/tasks$ sudo nano main.yml  
ubuntu@ip-172-31-20-121:/etc/ansible/roles/nginx/tasks$ sudo cat main.yml  
---  
# tasks file for nginx  
- include_tasks: install.yml  
ubuntu@ip-172-31-20-121:/etc/ansible/roles/nginx/tasks$
```

Write a playbook for execution

```
ubuntu@ip-172-31-20-121:/etc/ansible/roles/nginx/tasks$ cd  
ubuntu@ip-172-31-20-121:~$ ls  
a.sh  b.sh  play1.yml  play2.yml  
ubuntu@ip-172-31-20-121:~$ sudo nano play3.yml  
ubuntu@ip-172-31-20-121:~$
```

Play3.yml

```
GNU nano 6.2
---
- name: executing role on slave1
  hosts: slave1
  become: true
  roles:
    - apache2
- name: executing role on slave2
  hosts: slave2
  become: true
  roles:
    - nginx
```

Syntax check

```
ubuntu@ip-172-31-20-121:~$ ansible-playbook play3.yml --syntax-check
playbook: play3.yml
```

Dry run

```
ubuntu@ip-172-31-20-121:~$ ansible-playbook play3.yml --check

PLAY [executing role on slave1] *****

TASK [Gathering Facts] *****
ok: [slave1]

TASK [apache2 : include_tasks] *****
included: /etc/ansible/roles/apache2/tasks/install.yml for slave1

TASK [apache2 : installing apache2 on slave1] *****
changed: [slave1]

PLAY [executing role on slave2] *****

TASK [Gathering Facts] *****
ok: [slave2]

TASK [nginx : include_tasks] *****
included: /etc/ansible/roles/nginx/tasks/install.yml for slave2
```

Actual run

```
ubuntu@ip-172-31-20-121:~$ ansible-playbook play3.yml

PLAY [executing role on slave1] *****

TASK [Gathering Facts] *****
ok: [slave1]

TASK [apache2 : include_tasks] *****
included: /etc/ansible/roles/apache2/tasks/install.yml for slave1

TASK [apache2 : installing apache2 on slave1] *****
changed: [slave1]

PLAY [executing role on slave2] *****

TASK [Gathering Facts] *****
ok: [slave2]

TASK [nginx : include_tasks] *****
included: /etc/ansible/roles/nginx/tasks/install.yml for slave2

TASK [nginx : include_tasks] *****
included: /etc/ansible/roles/nginx/tasks/install.yml for slave2

TASK [nginx : installing nginx on slave2] *****
changed: [slave2]

PLAY RECAP *****
slave1      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignore=0
slave2      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignore=0
```

Go to slave1 and check apache2 service status

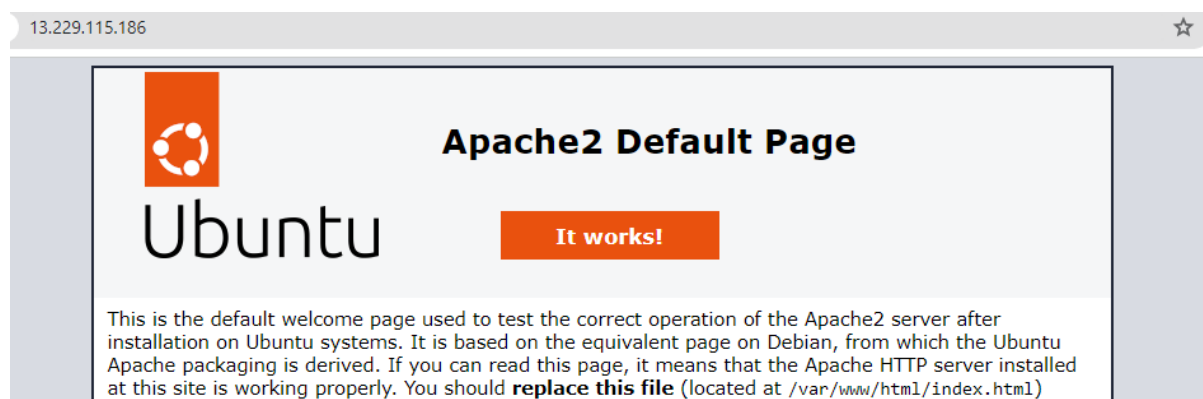
```
ubuntu@ip-172-31-21-228:~$ service apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enab
   Active: active (running) since Sun 2024-06-16 18:35:04 UTC; 2min 27s ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 6086 (apache2)
      Tasks: 55 (limit: 1121)
   Memory: 4.9M
      CPU: 39ms
    CGroup: /system.slice/apache2.service
            └─6086 /usr/sbin/apache2 -k start
              └─6088 /usr/sbin/apache2 -k start
                └─6089 /usr/sbin/apache2 -k start

Jun 16 18:35:04 ip-172-31-21-228 systemd[1]: Starting The Apache HTTP Server...
Jun 16 18:35:04 ip-172-31-21-228 systemd[1]: Started The Apache HTTP Server.
ubuntu@ip-172-31-21-228:~$
```

On slave2 check nginx status

[illegible]

Ip address of slave1 followed by port 80



Ip address of slave2 followed by port 80



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Assignment - 4

Tasks To Be Performed:

1. Use the previous deployment of Ansible cluster
2. 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.

Solutions:

Go to apache2 file folder and write one index.html code

```
ubuntu@ip-172-31-20-121:/etc/ansible/roles/apache2$ cd files
ubuntu@ip-172-31-20-121:/etc/ansible/roles/apache2/files$ ls
index.html
ubuntu@ip-172-31-20-121:/etc/ansible/roles/apache2/files$ cat index.html
<!DOCTYPE html>
<html>
<head>
    <title>
        Scientific Calculator using HTML, CSS and Js
    </title>
    <script src=
"https://cdnjs.cloudflare.com/ajax/libs/mathjs/10.6.4/math.js"
        integrity=
```

Go to task folder and write copy file which says copying new index.html file to default location

Give the source and destination locations

```
ubuntu@ip-172-31-20-121:/etc/ansible/roles/apache2$ cd tasks
ubuntu@ip-172-31-20-121:/etc/ansible/roles/apache2/tasks$ ls
copy.yml  install.yml  main.yml
ubuntu@ip-172-31-20-121:/etc/ansible/roles/apache2/tasks$ cat copy.yml
---
- name: copying new index.html file to default location
  copy: src=/etc/ansible/roles/apache2/files/index.html
        dest=/var/www/html/
```

Include the task on main.yml file for execution

```
ubuntu@ip-172-31-20-121:/etc/ansible/roles/apache2/tasks$ cat main.yml
---
# tasks file for apache2
# - include_tasks: install.yml
- include_tasks: copy.yml
```

Similarly create in the nginx folder

```
ubuntu@ip-172-31-20-121:/etc/ansible/roles/apache2/tasks$ cd ../../
ubuntu@ip-172-31-20-121:/etc/ansible/roles$ ls
apache2  nginx
ubuntu@ip-172-31-20-121:/etc/ansible/roles$ cd nginx
ubuntu@ip-172-31-20-121:/etc/ansible/roles/nginx$ ls
README.md  defaults  files  handlers  meta  tasks  templates  tests  vars
```

```
ubuntu@ip-172-31-20-121:/etc/ansible/roles/nginx/files$ sudo nano index.nginx-debian.html
```

```
ubuntu@ip-172-31-20-121:/etc/ansible/roles/nginx/files$ cat index.nginx-debian.html
<html>
<head>
<title> Yes it works!!! </title>
</head>
<body>
<h1> This is customized nginx webpage </h1>
</body>
</html>
```

```
ubuntu@ip-172-31-20-121:/etc/ansible/roles/nginx/tasks$ sudo nano copy.yml
ubuntu@ip-172-31-20-121:/etc/ansible/roles/nginx/tasks$ sudo cat copy.yml
---
- name: copying new index.nginx-debian.html file to default location
  copy: src=/etc/ansible/roles/nginx/files/index.nginx-debian.html
        dest=/var/www/html/
```

Write a playbook for executing slaves

```
ubuntu@ip-172-31-20-121:~$ sudo nano play4.yml
ubuntu@ip-172-31-20-121:~$ sudo cat play4.yml
---
- name: executing role on slave1
  hosts: slave1
  become: true
  roles:
    - apache2
- name: executing role on slave2
  hosts: slave2
  become: true
  roles:
    - nginx
```


Run the playbook.

```
ubuntu@ip-172-31-20-121:~$ ansible-playbook play4.yml

PLAY [executing role on slave1] *****

TASK [Gathering Facts] *****
ok: [slave1]

TASK [apache2 : include_tasks] *****
included: /etc/ansible/roles/apache2/tasks/copy.yml for slave1

TASK [apache2 : copying new index.html file to default location] *****
changed: [slave1]

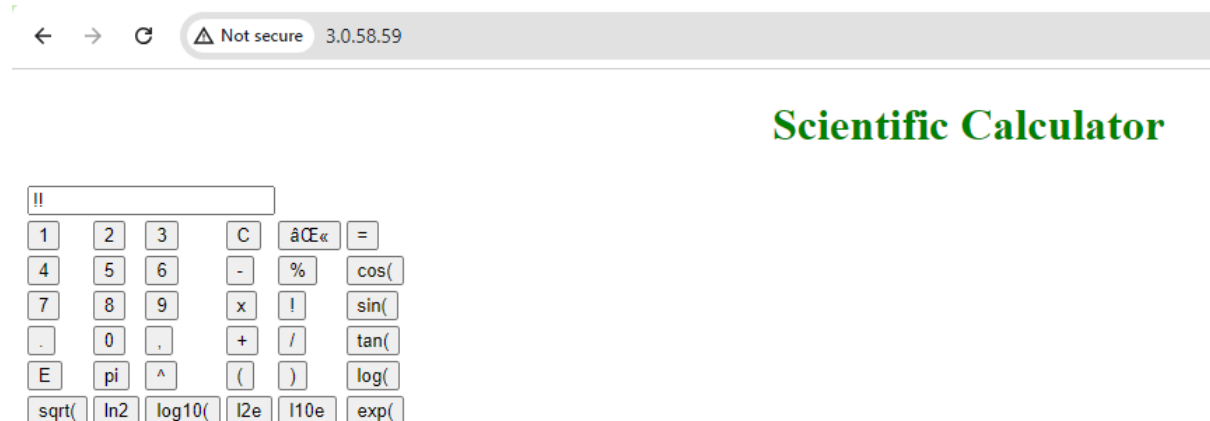
PLAY [executing role on slave2] *****

TASK [Gathering Facts] *****
ok: [slave2]

TASK [nginx : include_tasks] *****
included: /etc/ansible/roles/nginx/tasks/copy.yml for slave2

TASK [nginx : copying new index.nginx-debian.html file to default locatio
changed: [slave2]
```

Slave1 html code



Slave 2 html code



Assignment – 5

Tasks To Be Performed:

1. Create a new deployment of Ansible cluster of 5 nodes
2. Label 2 nodes as test and other 2 as prod
3. Install Java on test nodes
4. Install MySQL server on prod nodes

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

Launch 5 instances for creating 5 nodes

<input type="checkbox"/>	ansible-master	i-0eef2307b21dc9b6f	Running	t2.micro	2/2 checks passed View alarms +	ap-sou
<input type="checkbox"/>	ansible-prod1	i-010bce9e31481d143	Running	t2.micro	2/2 checks passed View alarms +	ap-sou
<input type="checkbox"/>	ansible-prod2	i-0c4fc5810a9ed8aec	Running	t2.micro	2/2 checks passed View alarms +	ap-sou
<input type="checkbox"/>	ansible-test1	i-0e091d84af4ab9f2a	Running	t2.micro	- View alarms +	ap-sou
<input type="checkbox"/>	ansible-test2	i-07594cdf69beae247	Running	t2.micro	- View alarms +	ap-sou

Create a token on master node and paste to all other slaves

```
ubuntu@ip-172-31-28-129:~$ cd .ssh
ubuntu@ip-172-31-28-129:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-28-129:~/.ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-28-129:~/.ssh$ sudo cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAha9lPDdaSWXxfAL/v61BpNUoZV6D8QmKbn1ipj7XT8nkI9Ih08W8PlfC7wg/uef
jJqPCglo+QWVC/ntZZVcoecAg/AQCoxXlHB1YB0WSl/qk08qj653Ce9+ZEHBG6dtcxwnpjKAKOZryZhmUxynFjsAuDMv64jIcK7wcHC
/TM1sZSscxJShd/nPBQL5ZEgrvR+M1EC8OVOzTbx/E/hTdyn73ZPIEmpUuIKnwcH4ts0FpYo0yb1uz61NcQ7UbX2cgErrkJ devops
ubuntu@ip-172-31-28-129:~/.ssh$ sudo nano authorized_keys
```

Write the host in master node

```
[group1]
prod1 ansible_host=172.31.21.228
prod2 ansible_host=172.31.29.200
[group2]
test1 ansible_host=172.31.28.129
test2 ansible_host=172.31.19.173
```

Pin the request for all 4 slaves

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? prod2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
prod1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
yes
test1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

Go to ansible location and create 2 roles prod and test

```
ubuntu@ip-172-31-20-121:~$ cd /etc/ansible
ubuntu@ip-172-31-20-121:/etc/ansible$ ls
ansible.cfg  hosts  roles
ubuntu@ip-172-31-20-121:/etc/ansible$ cd roles
ubuntu@ip-172-31-20-121:/etc/ansible/roles$ ls
apache2  nginx
ubuntu@ip-172-31-20-121:/etc/ansible/roles$ sudo ansible-galaxy init prod
- Role prod was created successfully
ubuntu@ip-172-31-20-121:/etc/ansible/roles$ ls
apache2  nginx  prod
ubuntu@ip-172-31-20-121:/etc/ansible/roles$ sudo ansible-galaxy init test
- Role test was created successfully
ubuntu@ip-172-31-20-121:/etc/ansible/roles$ ls
apache2  nginx  prod  test
```

In prod folder write install.yml file and main.yml file

```
ubuntu@ip-172-31-20-121:/etc/ansible/roles$ cd prod
ubuntu@ip-172-31-20-121:/etc/ansible/roles/prod$ ls
README.md  defaults  files  handlers  meta  tasks  templates  tests  vars
ubuntu@ip-172-31-20-121:/etc/ansible/roles/prod$ cd tasks
ubuntu@ip-172-31-20-121:/etc/ansible/roles/prod/tasks$ ls
main.yml
ubuntu@ip-172-31-20-121:/etc/ansible/roles/prod/tasks$ sudo nano install.yml
ubuntu@ip-172-31-20-121:/etc/ansible/roles/prod/tasks$ sudo nano main.yml
```

Same for the test folder

```
ubuntu@ip-172-31-20-121:~$ cd /etc/ansible/roles/test/tasks
ubuntu@ip-172-31-20-121:/etc/ansible/roles/test/tasks$ ls
main.yml
ubuntu@ip-172-31-20-121:/etc/ansible/roles/test/tasks$ sudo nano install.yml
ubuntu@ip-172-31-20-121:/etc/ansible/roles/test/tasks$ sudo nano main.yml
ubuntu@ip-172-31-20-121:/etc/ansible/roles/test/tasks$
```

Create a playbook and run

```
ubuntu@ip-172-31-20-121:~$ ansible-playbook play5.yml --syntax -check
playbook: play5.yml
```

Go to prod 1 and prod2 slaves check java installation reflecting

```
ubuntu@ip-172-31-21-228:~$ java --version
openjdk 11.0.23 2024-04-16
OpenJDK Runtime Environment (build 11.0.23+9-post-Ubuntu-1ubuntu122.04.1)
OpenJDK 64-Bit Server VM (build 11.0.23+9-post-Ubuntu-1ubuntu122.04.1, mixed mode, sharing)
ubuntu@ip-172-31-21-228:~$
```

Check mysql on test1 and test2 slaves

```
ubuntu@ip-172-31-28-129:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-06-17 16:37:58 UTC; 19s ago
     Process: 3511 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
    Main PID: 3519 (mysqld)
      Status: "Server is operational"
        Tasks: 38 (limit: 1121)
       Memory: 352.7M
          CPU: 1.076s
      CGroup: /system.slice/mysql.service
              └─3519 /usr/sbin/mysqld

Jun 17 16:37:57 ip-172-31-28-129 systemd[1]: Starting MySQL Community Server...
Jun 17 16:37:58 ip-172-31-28-129 systemd[1]: Started MySQL Community Server.
ubuntu@ip-172-31-28-129:~$
```
