ANSIBLE PROJECT ON AWS

You are a Devops Engineer and the organization you are working on needs to set up two configuration management server groups. One for Apache another for Nginx. Being a Devops Engineer it is your task to deal with this configuration management issue.

Let us see the tasks that you need to perform using Ansible.

1. Create two Server Groups. One for Apache and another for Nginx.
2. Push two html files with their server information.

Make sure that you don't forget to start the services once the installation is done. Also send post installation messages for both the server groups.

Using Ansible Roles accomplish the above the tasks.

Also, once the Apache server configuration is done you need to install Java on that server group using ansible role in a playbook.

WHAT IS ANSIBLE?

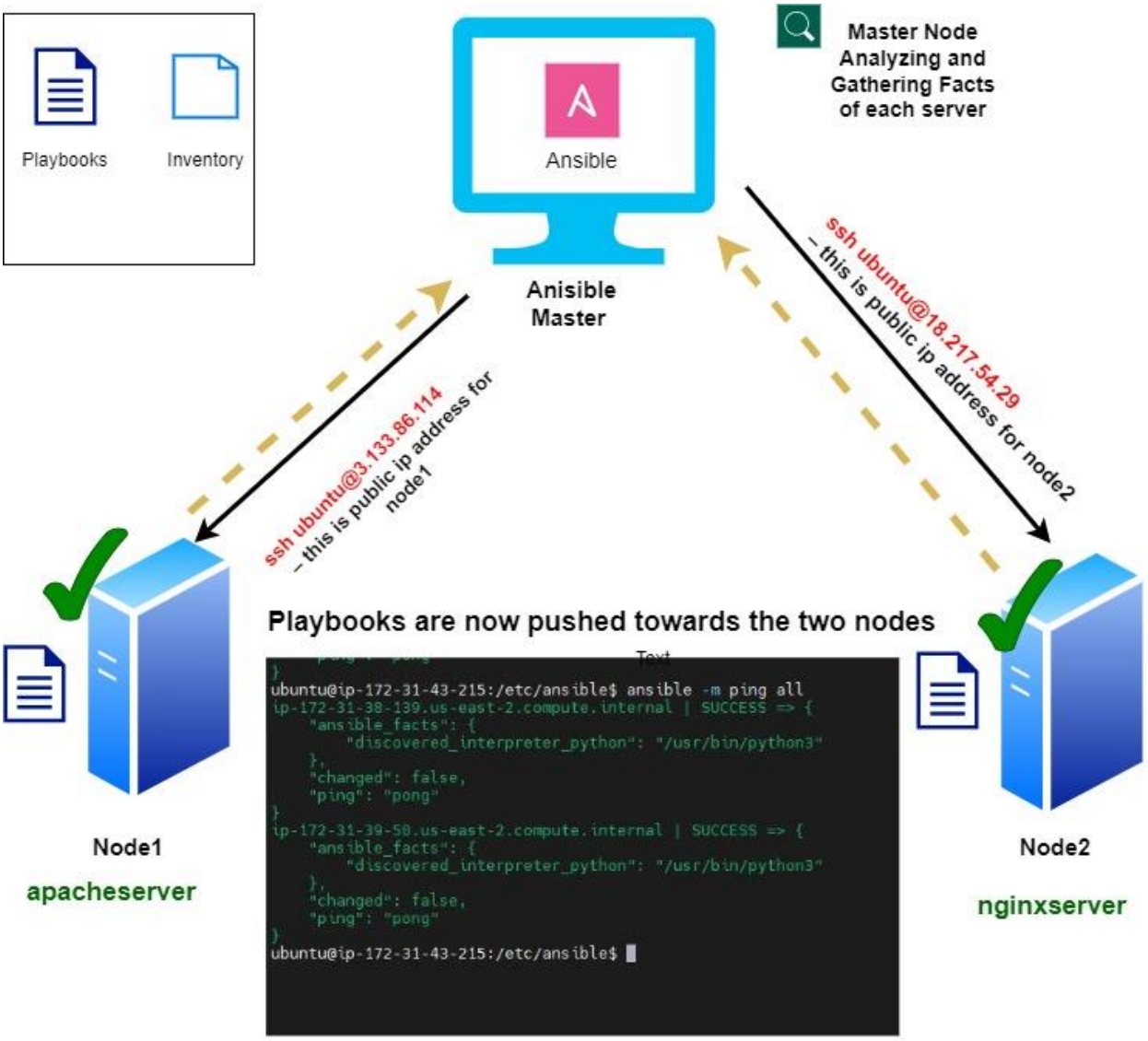
Ansible is an automation and orchestration tool popular for its simplicity of installation, ease of use in what concerns the connectivity to clients, its lack of agent for ansible clients and the multitude of skills.

Ansible supports the deployment and infrastructure management phases of the DevOps lifecycle, automating such tasks as configuration management, infrastructure provisioning, and application deployment.

Ansible functions by connecting via SSH to the clients, so it doesn't need a special agent on the client-side, and by pushing modules to the clients. The modules are then executed locally, on the client-side, and the output is pushed back to the Ansible server.

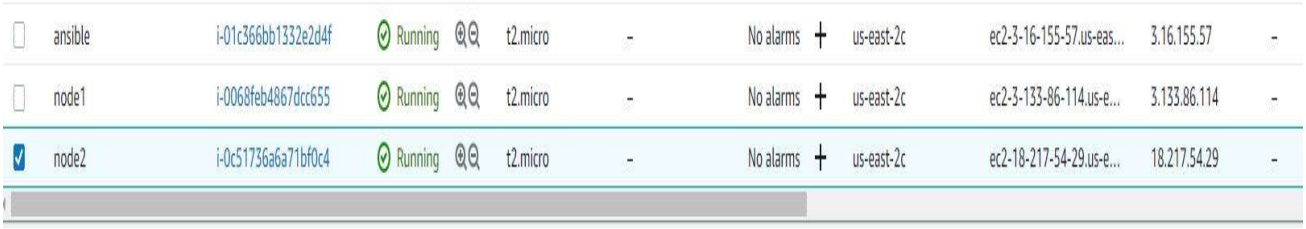
Since it uses SSH, it can very easily connect to clients using SSH-Keys, simplifying though the whole process. Client details, like hostnames or IP addresses and SSH ports, are stored in files called inventory files. Once you have created an inventory file and populated it, ansible can use it.

How Ansible Will Work In This Project



Ensure comminication between Master and Worker Nodes (Node1 and Node2)

Step 1. Configure 3 VMs on AWS. One ansible (master) and two worker nodes (apache (node1) and nginx (node2))



ansible	i-01c366bb1332e2d4f	Running	t2.micro	-	No alarms	us-east-2c	ec2-3-16-155-57.us-eas...	3.16.155.57	-
node1	i-0068feb4867dcc655	Running	t2.micro	-	No alarms	us-east-2c	ec2-3-133-86-114.us-e...	3.133.86.114	-
node2	i-0c51736a6a71bf0c4	Running	t2.micro	-	No alarms	us-east-2c	ec2-18-217-54-29.us-e...	18.217.54.29	-

Step 2. Install ansible by including the following commands in a script file, ansible.sh in the master (ansible) and run script file via bash ansible.sh

Check node1 and node2 to verify that python3 --version is installed.

```
sudo apt update
sudo apt install software-properties-common
sudo apt-add-repository --yes --update ppa:ansible/ansible
sudo apt install ansible
```

Step 3. ssh connection into node1 and node2 from the master. See commands below

```
cd
cd .ssh
ls
ssh-keygen
cat id_rsa.pub
```

On node:

```
cd .ssh
sudo nano authorized_key
```

Note: paste key save & exit

Step 4. Check ssh connection – both were successful

ssh ubuntu@3.133.86.114 – this is public ip address for node1

ssh ubuntu@18.217.54.29 – this is public ip address for node2

Step 5. Configure host file for two server groups.

Change to ansible directory and the host file
cd /etc/ansible
ls
Sudo nano hosts

Create two server groups.
-- Get apacheserver (node1) private ip address using hostname –f and paste in host file on master

[apacheserver]
ip-172-31-39-50.us-east-2.compute.internal

-- Get nginxserver (node2) private ip address using hostname –f and paste in host file on master

[nginxserver]
ip-172-31-38-139.us-east-2.compute.internal

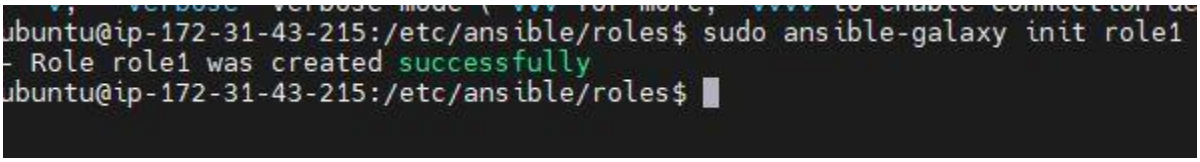
--Save & Exit

Step 6. Check to see if ansible can connect to the other two machines (node1 & node2) using the ping command

From the ansible directory cd /etc/ansible
ansible –m ping all

```
}
ubuntu@ip-172-31-43-215:/etc/ansible$ ansible -m ping all
ip-172-31-38-139.us-east-2.compute.internal | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ip-172-31-39-50.us-east-2.compute.internal | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-43-215:/etc/ansible$
```

Step 7. Create role1 & role2 using the command:



Navigate to tasks directory and create multiple files

Roles:
Role1: apacheserver

- Main.yml – currently there is nothing in this file
- Install.yml – to install apache2
- Config.yml – taking the file from the location and pasting it in the location /var/www/html
- Service.yml – make the apache service running
- Java.yml – install java

Files

- Index.html --------> /var/www/html

Create the following files under tasks using sudo nano <file_name> (where file name is as shown below)

sudo nano **install.yml**

```
---
  - name: Install Apache2
    apt: name=apache2 state=latest
    become: true
```

sudo nano **config.yml**

```
---
  - name: Install Apache2
    copy: src=index.html  dest=/var/www/html
    become: true
```

sudo nano **service.yml**

```
---
 - name: service
   service: name=apache2 state=started
   become: true
```

Step 8. Create a playbook

Go to /etc/ansible/

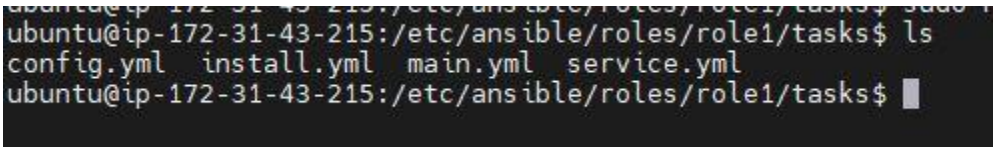
Sudo nano playbook.yml

playbook.yml

```
---
 - hosts: apacheserver
   roles:
     - role1
```

Step 9. Go to main.yml and attach the above files so playbook will read them

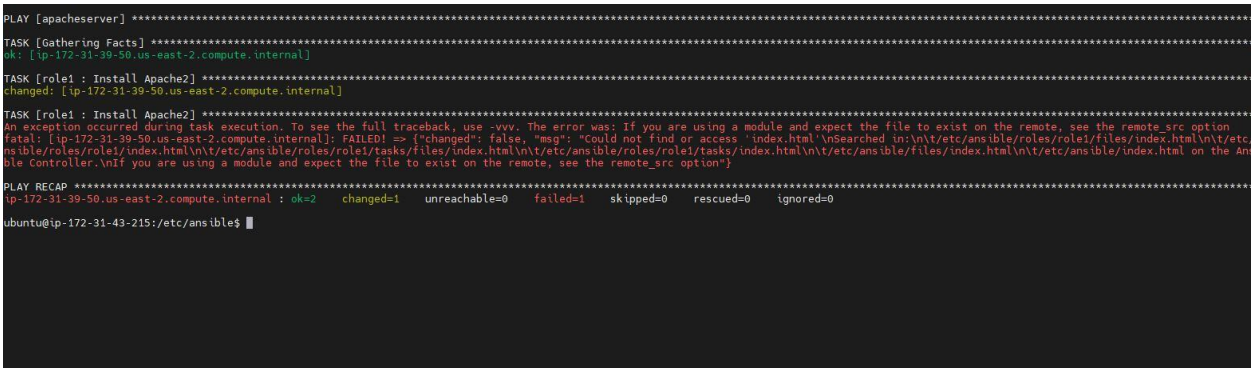
/etc/ansible/roles/role1/tasks



Step 10. Run playbook

etc/ansible

Using command: ansible-playbook playbook.yml



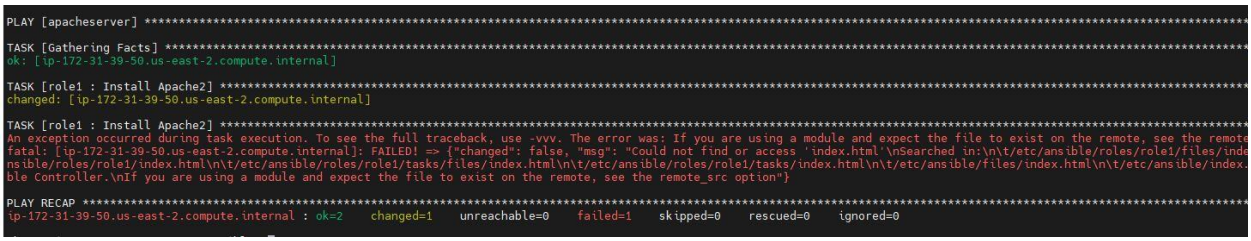
The above error requires me to update the machines (master, node1, & node2)

Step 11. Run playbook again

etc/ansible

Using command: ansible-playbook playbook.yml

Error:



html source is empty which I am trying to copy does not exit. So I need to create an index file as shown below.

/etc/ansible/roles/role1/files

Create index file: sudo nano index.html

Hi Everyone!

Step 12.

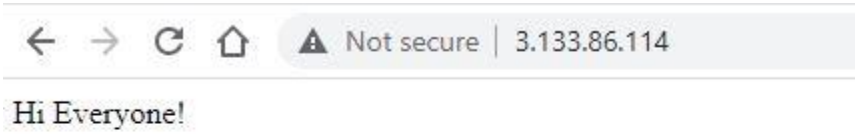
/etc/ansible

Run playbook again
Using command: ansible-playbook playbook.yml

```
PLAY [apacheserver] ************************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************************
ok: [ip-172-31-39-50.us-east-2.compute.internal]

TASK [role1 : Install Apache2] *************************************************************************************************************************
ok: [ip-172-31-39-50.us-east-2.compute.internal]

TASK [role1 : Install Apache2] *************************************************************************************************************************
changed: [ip-172-31-39-50.us-east-2.compute.internal]

TASK [role1 : service] *********************************************************************************************************************************
ok: [ip-172-31-39-50.us-east-2.compute.internal]

PLAY RECAP *********************************************************************************************************************************************
ip-172-31-39-50.us-east-2.compute.internal : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-43-215:/etc/ansible$
```

Step 13. Check from the browser

Node1 public IP address. The playbook runs successfully!



Step 14. Download CSS Template & then deploy

/etc/ansible/roles/role1/files

/etc/ansible/roles/role1/files$ sudo wget https://www.free-css.com/assets/files/free-css-templates/download/page276/spicy.zip

/etc/ansible/roles/role1/files$ sudo apt install unzip

/etc/ansible/roles/role1/files$ sudo unzip spicy.zip

Run playbook again
Using command: ansible-playbook playbook.yml

```
PLAY [apacheserver] ************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************
ok: [ip-172-31-39-50.us-east-2.compute.internal]

TASK [role1 : Install Apache2] *************************************************************************************************************
ok: [ip-172-31-39-50.us-east-2.compute.internal]

TASK [role1 : Install Apache2] *************************************************************************************************************
ok: [ip-172-31-39-50.us-east-2.compute.internal]

TASK [role1 : service] *********************************************************************************************************************
ok: [ip-172-31-39-50.us-east-2.compute.internal]

PLAY RECAP *********************************************************************************************************************************
ip-172-31-39-50.us-east-2.compute.internal : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-43-215:/etc/ansible$
```

We have only specified to copy index.html in the config.yml file so need to change to:

```
---
  - name: Install Apache2
    copy: src=spicy dest=/var/www/html
    become: true
```

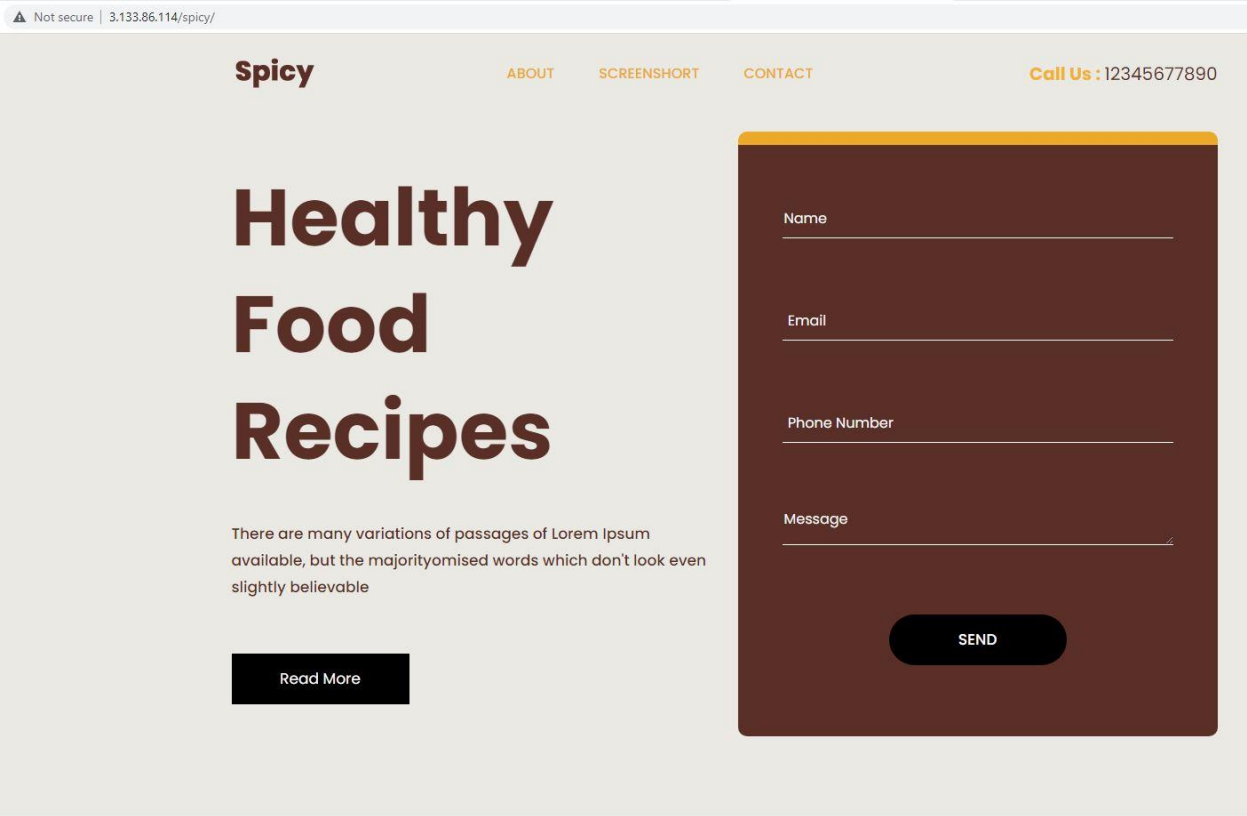
/etc/ansible

Run playbook again
Using command: ansible-playbook playbook.yml

```
PLAY [apacheserver] ************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************
ok: [ip-172-31-39-50.us-east-2.compute.internal]

TASK [role1 : Install Apache2] *************************************************************************************************************
ok: [ip-172-31-39-50.us-east-2.compute.internal]

TASK [role1 : Install Apache2] *************************************************************************************************************
changed: [ip-172-31-39-50.us-east-2.compute.internal]

TASK [role1 : service] *********************************************************************************************************************
ok: [ip-172-31-39-50.us-east-2.compute.internal]

PLAY RECAP *********************************************************************************************************************************
ip-172-31-39-50.us-east-2.compute.internal : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-43-215:/etc/ansible$
```

Changes have occurred as expected.

Step 15. Check results from browser which is working on node1 (worker node)



Playbook ran successfully!

Step 16. Now add role2 (node2)

ubuntu@ip-172-31-43-215:/etc/ansible/roles/role2/tasks$ ls
main.yml

/etc/ansible/roles/role2/tasks$ sudo nano install.yml

```
---
  - name: Install nginx
    apt: name=nginx state=latest
    become: true
```

/etc/ansible/roles/role2/tasks$ sudo nano config.yml

```
---
  - name: config indexhtml
    copy: src=index.html dest=/var/www/html
    become: true
```

/etc/ansible/roles/role2/tasks$ sudo nano service.yml

```
---
  - name: service
    service: name=nginx state=started
    become: true
```

/etc/ansible/roles/role2/tasks$ sudo nano main.yml

```
---
# tasks file for role2
  - include : install.yml
  - include : config.yml
  - include : service.yml
```

/etc/ansible/roles/role2/files$ sudo nano index.html

How's everyone doing?

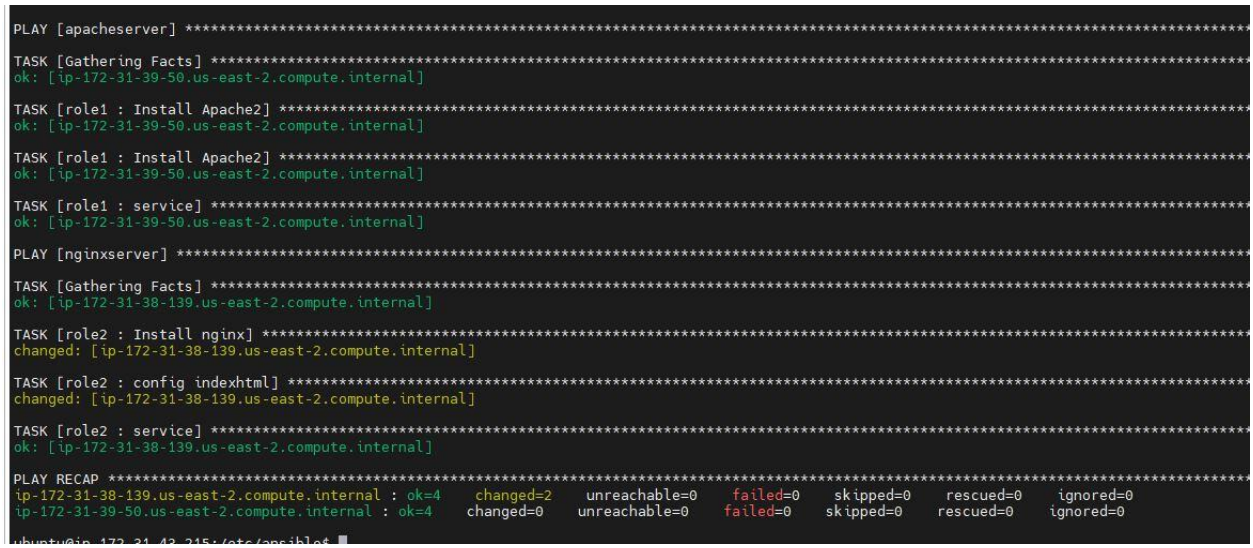
Now we need to make changes in the playbook

/etc/ansible$ sudo nano playbook.yml

```
---
  - hosts: apacheserver
    roles:
      - role1
  - hosts: nginxserver
    roles:
      - role2
```

Step 17. Now run playbook

/etc/ansible$ ansible-playbook playbook.yml



Two changes occurred as expected.

Step 18. Install Java

ubuntu@ip-172-31-43-215:/etc/ansible/roles/role1/tasks$ ls
config.yml install.yml main.yml service.yml

/etc/ansible/roles/role1/tasks$ sudo nano install java.yml

```
---
  - name: Install java
    apt: name=openjdk-11-jdk state=latest
    become: true
```

/etc/ansible/roles/role1/tasks$ sudo nano main.yml

```
---
# tasks file for role1
 - include: install.yml
 - include: config.yml
 - include: service.yml
 - include: java.yml
```

Need an html file for role2:

ubuntu@ip-172-31-43-215:/etc/ansible/roles/role2/files

Get template from internet

ubuntu@ip-172-31-43-215:/etc/ansible/roles/role2/files$ sudo wget https://www.free-css.com/assets/files/free-css-templates/download/page278/buzzed.zip

Install & Unzip file:

/etc/ansible/roles/role1/files$ sudo apt install unzip

/etc/ansible/roles/role1/files$ sudo buzzed.zip

ubuntu@ip-172-31-43-215:/etc/ansible/roles/role2/tasks$ sudo nano config.yml

change config.yml to:

```
---
  - name: config indexhtml
    copy: src=buzzed-html-template dest=/var/www/html
    become: true
```

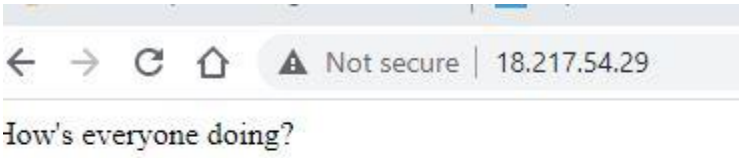
Step 19: Run playbook again

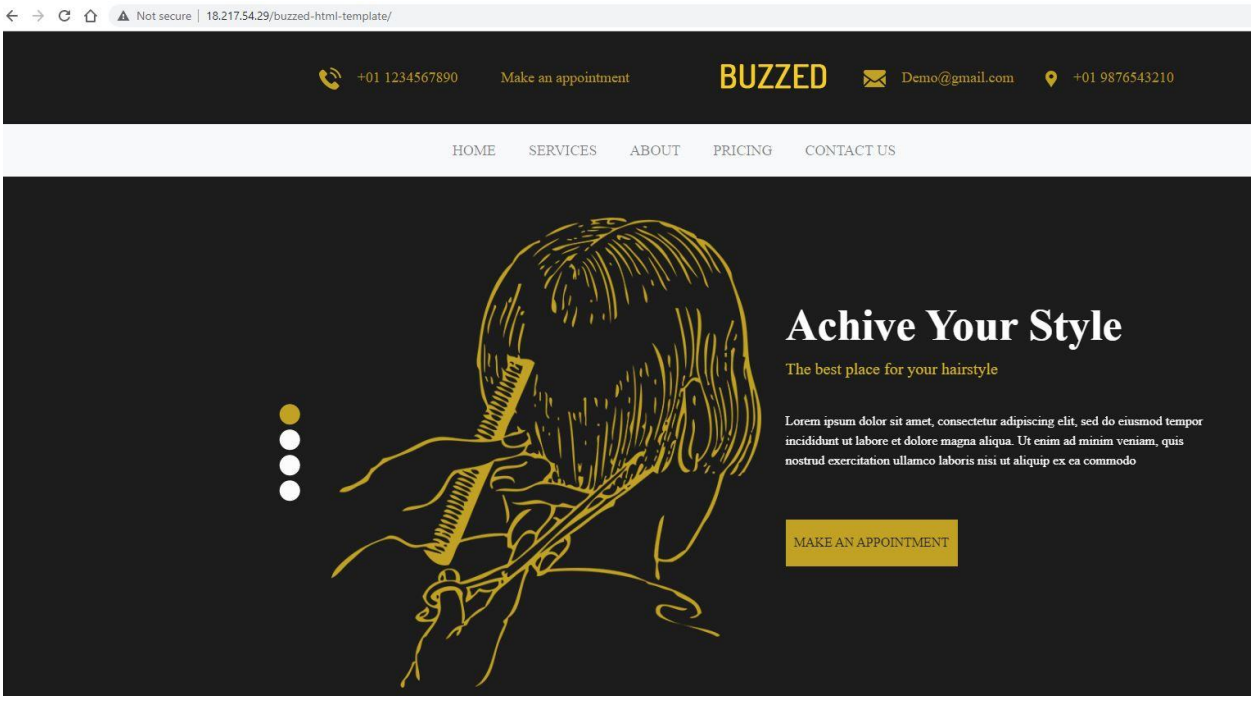
ubuntu@ip-172-31-43-215:/etc/ansible$ ansible-playbook playbook.yml

```
PLAY [apacheserver] ************************************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************************************
ok: [ip-172-31-39-50.us-east-2.compute.internal]

TASK [role1 : Install Apache2] *************************************************************************************************************************************
ok: [ip-172-31-39-50.us-east-2.compute.internal]

TASK [role1 : Install Apache2] *************************************************************************************************************************************
ok: [ip-172-31-39-50.us-east-2.compute.internal]

TASK [role1 : service] *********************************************************************************************************************************************
ok: [ip-172-31-39-50.us-east-2.compute.internal]

TASK [role1 : Install java] ****************************************************************************************************************************************
changed: [ip-172-31-39-50.us-east-2.compute.internal]

PLAY [nginxserver] *************************************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************************************
ok: [ip-172-31-38-139.us-east-2.compute.internal]

TASK [role2 : Install nginx] ***************************************************************************************************************************************
ok: [ip-172-31-38-139.us-east-2.compute.internal]

TASK [role2 : config indexhtml] ************************************************************************************************************************************
changed: [ip-172-31-38-139.us-east-2.compute.internal]

TASK [role2 : service] *********************************************************************************************************************************************
ok: [ip-172-31-38-139.us-east-2.compute.internal]

PLAY RECAP *********************************************************************************************************************************************************
ip-172-31-38-139.us-east-2.compute.internal : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ip-172-31-39-50.us-east-2.compute.internal : ok=5    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Two changes occurred as expected.

Step 20: Results of node2





Node2 playbook ran successfully!