ANSIBLE CASE STUDY

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.

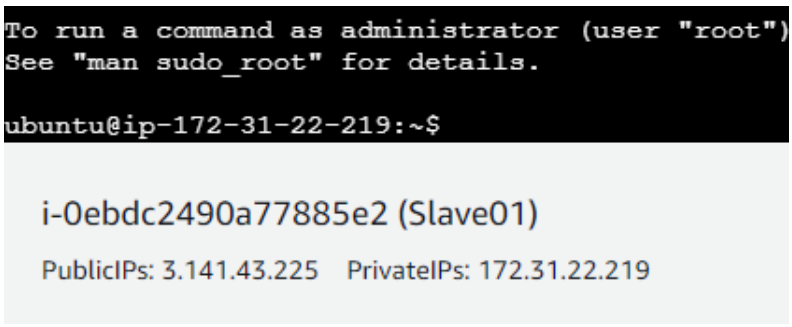
Solutions:

Create 1 Master node and 2 slaves for Apache and 2 slaves for Nginx

☐	salve03	i-017659d89146fcb66	Running	t2.micro
☐	salve04	i-023102d1b3418c024	Running	t2.micro
☐	Master	i-01f4d04667162e0c7	Running	t2.micro
☐	Slave01	i-0ebdc2490a77885e2	Running	t2.micro
☐	Slave02	i-02610ad3c2f10b426	Running	t2.micro

Connect and update all the nodes:

Ansible-Master node



Install Ansible in master node only

```
ubuntu@AnsibleMaster:~$ sudo cat a.sh
sudo apt-add-repository ppa:ansible/ansible
sudo apt update
sudo apt install ansible -y
ubuntu@AnsibleMaster:~$
```

i-01f4d04667162e0c7 (Master)

$ ansible --version

```
ubuntu@ip-172-31-17-95:~$ ansible --version
ansible [core 2.16.10]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/ubuntu/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.12.3 (main, Apr 10 2024, 05:33:47) [GCC 13.2.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
ubuntu@ip-172-31-17-95:~$
```

i-05f2f94026f8cb82b (Ansible-master)

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.

```
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_ed25519
Your public key has been saved in /home/ubuntu/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:8nVgM131rpmZMOayjAUcQIRsYn5oSGEWq55maSTjSIE ubuntu@ip-172-31-17
The key's randomart image is:
+--[ED25519 256]--+
| =o. ++       ...|
|+.+ +  .    . .  .|
|E* +    . = .    .|
|o = .  . o +    . |
|++ .  . S . =    .|
|B.o    o o + o * |
|.O      . o . *  |
|+        + o     |
|         . o     |
+----[SHA256]-----+
ubuntu@ip-172-31-17-95:~$
```

i-05f2f94026f8cb82b (Ansible-master)

Go to .ssh path, copy the public key which is generated

```
ubuntu@ip-172-31-17-95:~$ cd .ssh
ubuntu@ip-172-31-17-95:~/.ssh$ ls
authorized_keys  id_ed25519  id_ed25519.pub
ubuntu@ip-172-31-17-95:~/.ssh$ sudo cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAICCZQfyW8P1F0zSaIebwwFo1R9MlhIbzxt+HHvF
ubuntu@ip-172-31-17-95:~/.ssh$
```

i-05f2f94026f8cb82b (Ansible-master)

Go to the slaves paste it in the authorized keys file

```
ubuntu@ip-172-31-28-117:~$ cd .ssh
ubuntu@ip-172-31-28-117:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-28-117:~/.ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-28-117:~/.ssh$
```

i-0350d5ae72d3ae32c (Slave01)

PublicIPs: 18.142.47.130 PrivateIPs: 172.31.28.117

```
  GNU nano 7.2                                                    autho
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAICCZQfyW8P1F0zSaIebwwFo1R9MlhIbzxt+HHvFANdl4
```

Slave-2

```
ubuntu@ip-172-31-21-136:~$ cd .ssh
ubuntu@ip-172-31-21-136:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-21-136:~/.ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-21-136:~/.ssh$
```

i-04b7aee1c5cbbac1f (Slave02)

PublicIPs: 18.139.225.158 PrivateIPs: 172.31.21.136

Slave-3

```
ubuntu@ip-172-31-23-39:~$ cd .ssh
ubuntu@ip-172-31-23-39:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-23-39:~/.ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-23-39:~/.ssh$
```

i-0a9dc3f4afc85764b (slave03)

PublicIPs: 47.128.251.123 PrivateIPs: 172.31.23.39

Slave-4

```
ubuntu@ip-172-31-19-74:~$ cd .ssh
ubuntu@ip-172-31-19-74:~/.ssh$ la
authorized_keys
ubuntu@ip-172-31-19-74:~/.ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-19-74:~/.ssh$
```

i-003a9ccb2fab14e60 (slave04)

PublicIPs: 13.215.178.0 PrivateIPs: 172.31.19.74

Now go to ansible location for creating hosts

```
ubuntu@ip-172-31-17-95:~/.ssh$ cd /etc/ansible/
ubuntu@ip-172-31-17-95:/etc/ansible$ ls
ansible.cfg  hosts  roles
ubuntu@ip-172-31-17-95:/etc/ansible$ sudo nano hosts
ubuntu@ip-172-31-17-95:/etc/ansible$
```

i-05f2f94026f8cb82b (Ansible-master)

Put the first two slaves (Slave1 & Slave2) machines in apache & the last two slaves (Slave3 & Slave4) machines in the nginx group.

```
  GNU nano 7.2
[apache]
Slave1 ansible_host=172.31.28.117
Slave2 ansible_host=172.31.21.136

[nginx]
Slave3 ansible_host=172.31.23.39
Slave4 ansible_host=172.31.19.74
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
```

$ ansible –m ping all

If this command gives error, you follow the below method

```
ubuntu@ip-172-31-17-95:/etc/ansible$ ssh-keyscan -H 172.31.28.117 >> ~/.ssh/known_hosts
# 172.31.28.117:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.4
# 172.31.28.117:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.4
# 172.31.28.117:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.4
# 172.31.28.117:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.4
# 172.31.28.117:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.4
ubuntu@ip-172-31-17-95:/etc/ansible$ ssh-keyscan -H 172.31.21.136 >> ~/.ssh/known_hosts
# 172.31.21.136:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.4
# 172.31.21.136:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.4
# 172.31.21.136:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.4
# 172.31.21.136:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.4
# 172.31.21.136:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.4
ubuntu@ip-172-31-17-95:/etc/ansible$ ssh-keyscan -H 172.31.19.74 >> ~/.ssh/known_hosts
# 172.31.19.74:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.4
# 172.31.19.74:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.4
# 172.31.19.74:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.4
# 172.31.19.74:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.4
# 172.31.19.74:22 SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.4
ubuntu@ip-172-31-17-95:/etc/ansible$ ansible -m ping all
Slave03 | SUCCESS => {
```

Again give ping command, it works successfully

```
ubuntu@ip-172-31-17-95:/etc/ansible$ ansible -m ping all
Slave03 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
Slave04 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
Slave01 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
Slave02 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
```

Now create two roles for apache and nginx

List the files

```
ubuntu@ip-172-31-17-95:/etc/ansible$ cd roles/
ubuntu@ip-172-31-17-95:/etc/ansible/roles$ ls
ubuntu@ip-172-31-17-95:/etc/ansible/roles$ sudo ansible-galaxy init apache
- Role apache was created successfully
ubuntu@ip-172-31-17-95:/etc/ansible/roles$ sudo ansible-galaxy init nginx
- Role nginx was created successfully
ubuntu@ip-172-31-17-95:/etc/ansible/roles$ ls
apache  nginx
ubuntu@ip-172-31-17-95:/etc/ansible/roles$
```

i-05f2f94026f8cb82b (Ansible-master)

Go inside apache, tasks and create install.yml file

```
/apache/tasks$ sudo nano install.yml
```

Give the command for installing apache 2 server

```
  GNU nano 7.2
- name: installing apache2 on apache server group
  apt: name=apache2 update_cache=yes state=latest
  become: true
```

Inside tasks create configure.yml file

```
les/apache/tasks$ sudo nano configure.yml
```

Write the commands for copying html file from default location and notify for restarting

```
  GNU nano 7.2
- name: copy the html file
  become: true
  copy: src=sample.html dest=/var/www/html
  notify:
  - restart the apache web server
```

Now create main.yml file to include the tasks

```
le/roles/apache/tasks$ sudo nano main.yml
```

Mention the yaml files here for execution

```
  GNU nano 7.2
---
- include_tasks: install.yaml
- include_tasks: configure.yaml
# tasks file for apache
```

Go back from the task folder, enter to handlers folder and create one main.yml file

```
ubuntu@ip-172-31-17-95:/etc/ansible/roles/apache/tasks$ cd ..
ubuntu@ip-172-31-17-95:/etc/ansible/roles/apache$ ls
README.md  defaults  files  handlers  meta  tasks  templates  tests  vars
ubuntu@ip-172-31-17-95:/etc/ansible/roles/apache$ cd handlers/
ubuntu@ip-172-31-17-95:/etc/ansible/roles/apache/handlers$ sudo nano main.yml
```

Mention the commands for restarting the apache server

```
---
# handlers file for apache
- name: restart the apache web server
  service: name=apache2 state=restarted
  become: true
```

Now go under the files folder and create one sample.html file

```
/etc/ansible/roles/apache/handlers$ cd ..
/etc/ansible/roles/apache$ ls
les  handlers  meta  tasks  templates  tests  vars
/etc/ansible/roles/apache$ cd files/
/etc/ansible/roles/apache/files$ ls
/etc/ansible/roles/apache/files$ sudo nano sample.html
```

Create one simple html code

```
  GNU nano 7.2
<h1>Welcome to Apache Website</h1>
```

Similarly go the nginx server path and create tasks

```
ubuntu@ip-172-31-17-95:/etc/ansible/roles/nginx$ cd tasks/
ubuntu@ip-172-31-17-95:/etc/ansible/roles/nginx/tasks$ ls
main.yml
ubuntu@ip-172-31-17-95:/etc/ansible/roles/nginx/tasks$ sudo nano install.yml
ubuntu@ip-172-31-17-95:/etc/ansible/roles/nginx/tasks$
```

In the install.yml file, write the command for installing nginx server

```
  GNU nano 7.2
- name: installing nginx on nginx server
  apt: name=nginx update_cache=yes state=latest
  become: true
```

Under the tasks folder only, create configure.yml file

```
/nginx/tasks$ sudo nano configure.yml
```

Write the code for copying the html file

```
  GNU nano 7.2
- name: copy the html file
  become: true
  copy: src=sample.html dest=/var/www/html
  notify:
  - restart the nginx web server
```

Create main.yml file to include the created tasks

```
roles/nginx/tasks$ sudo nano main.yml
```

Mention install.yaml file and configure.yaml file

```
  GNU nano 7.2
---
# tasks file for nginx
- include_tasks: install.yaml
- include_tasks: configure.yaml
```

Create main.yml file inside handlers folder path

```
/roles/nginx$ cd handlers/
/roles/nginx/handlers$ ls

/roles/nginx/handlers$ sudo nano main.yml
```

Give commands for restarting nginx

```
---
# handlers file for nginx
- name: restart the nginx web server
  service: name=nginx state=restarted
  become: true
```

Go for the file section and create another sample.html file for nginx server

```
-95:/etc/ansible/roles/nginx/handlers$ cd ..
-95:/etc/ansible/roles/nginx$ ls
 files  handlers  meta  tasks  templates  tests  vars
-95:/etc/ansible/roles/nginx$ cd files/
-95:/etc/ansible/roles/nginx/files$ ls
-95:/etc/ansible/roles/nginx/files$ sudo nano sample.html
```

Write a html code for nginx

```
  GNU nano 7.2
<h1>Welcome to NGINX Website</h1>
```

As per the question we need to install java as well so initialize the java resource under roles

Go inside java> tasks> create main.yml file

```
ubuntu@ip-172-31-17-95:/etc/ansible/roles$ sudo ansible-galaxy init java
- Role java was created successfully
ubuntu@ip-172-31-17-95:/etc/ansible/roles$ ls
apache  java  nginx
ubuntu@ip-172-31-17-95:/etc/ansible/roles$ cd java/
ubuntu@ip-172-31-17-95:/etc/ansible/roles/java$ ls
README.md  defaults  files  handlers  meta  tasks  templates  tests  vars
ubuntu@ip-172-31-17-95:/etc/ansible/roles/java$ cd tasks/
ubuntu@ip-172-31-17-95:/etc/ansible/roles/java/tasks$ ls
main.yml
ubuntu@ip-172-31-17-95:/etc/ansible/roles/java/tasks$ sudo nano main.yml
```

give commands for installing Java.

```
---
# tasks file for java
- name: install java on webserver
  apt: name=openjdk-18-jre-headless state=present
  become: true
```

now create the playbook yaml file for execution of apache, nginx and java servers.

```
ubuntu@ip-172-31-17-95:/etc/ansible/roles$ ls
apache  java  nginx
ubuntu@ip-172-31-17-95:/etc/ansible/roles$ sudo nano playbook.yml
ubuntu@ip-172-31-17-95:/etc/ansible/roles$
```

i-05f2f94026f8cb82b (Ansible-master)

Write the playbook in which it executes all the created roles

```
---
- hosts: apache
  roles:
  - apache
- hosts: nginx
  roles:
  - nginx

- hosts: apache
  roles:
  - java
- hosts: nginx
  roles:
  - java
```

Give the execution command

```
ubuntu@ip-172-31-17-95:/etc/ansible/roles$ ansible-playbook playbook.yml

PLAY [apache] ******************************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [Slave01]
ok: [Slave02]

TASK [apache : include_tasks] **************************************************************************
included: /etc/ansible/roles/apache/tasks/install.yml for Slave01, Slave02

TASK [apache : installing apache2 on apache server group] **********************************************
ok: [Slave01]
ok: [Slave02]

TASK [apache : include_tasks] **************************************************************************
included: /etc/ansible/roles/apache/tasks/configure.yml for Slave01, Slave02

TASK [apache : copy the html file] *********************************************************************
ok: [Slave01]
ok: [Slave02]
```

```
PLAY [nginx] *******************************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [Slave03]
ok: [Slave04]

TASK [nginx : include_tasks] ***************************************************************************
included: /etc/ansible/roles/nginx/tasks/install.yml for Slave03, Slave04

TASK [nginx : installing nginx on nginx server] ********************************************************
ok: [Slave03]
ok: [Slave04]

TASK [nginx : include_tasks] ***************************************************************************
included: /etc/ansible/roles/nginx/tasks/configure.yml for Slave03, Slave04

TASK [nginx : copy the html file] **********************************************************************
ok: [Slave03]
ok: [Slave04]

PLAY RECAP *********************************************************************************************
Slave01                    : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
Slave02                    : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
Slave03                    : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
Slave04                    : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

You can check java installation on all slaves

```
TASK [java : install java on webserver] ****************************************************************************************************
changed: [Slave02]
changed: [Slave01]

PLAY [nginx] *******************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************
ok: [Slave04]
ok: [Slave03]

TASK [java : install java on webserver] ****************************************************************************************************
changed: [Slave04]
changed: [Slave03]

PLAY RECAP *********************************************************************************************************************************
Slave01                    : ok=7    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
Slave02                    : ok=7    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
Slave03                    : ok=7    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
Slave04                    : ok=7    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Go to slave 1 and slave 2 and check apache2 status and confirm it is running

```
ubuntu@ip-172-31-28-117:~/.ssh$ service apache2 status
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Sun 2024-09-01 07:43:06 UTC; 23min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 4223 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 4226 (apache2)
      Tasks: 55 (limit: 1130)
     Memory: 5.3M (peak: 5.5M)
        CPU: 84ms
     CGroup: /system.slice/apache2.service
             ├─4226 /usr/sbin/apache2 -k start
             ├─4228 /usr/sbin/apache2 -k start
             └─4229 /usr/sbin/apache2 -k start

Sep 01 07:43:06 ip-172-31-28-117 systemd[1]: Starting apache2.service - The Apache HTTP Ser
Sep 01 07:43:06 ip-172-31-28-117 systemd[1]: Started apache2.service - The Apache HTTP Serv
```

Go to slave 3 and 4, check the active status of nginx server

```
ubuntu@ip-172-31-23-39:~/.ssh$ service nginx status
● 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 Sun 2024-09-01 07:46:17 UTC; 24min ago
       Docs: man:nginx(8)
    Process: 3721 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code
    Process: 3722 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited,
   Main PID: 3724 (nginx)
      Tasks: 2 (limit: 1130)
     Memory: 1.9M (peak: 2.1M)
        CPU: 6ms
     CGroup: /system.slice/nginx.service
             ├─3724 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             └─3725 "nginx: worker process"

Sep 01 07:46:17 ip-172-31-23-39 systemd[1]: Starting nginx.service - A high performance web
Sep 01 07:46:17 ip-172-31-23-39 systemd[1]: Started nginx.service - A high performance web
ubuntu@ip-172-31-23-39:~/.ssh$
```

i-0a9dc3f4afc85764b (slave03)

Also check the java installed in all the slaves – 1,2,3 and 4 slaves

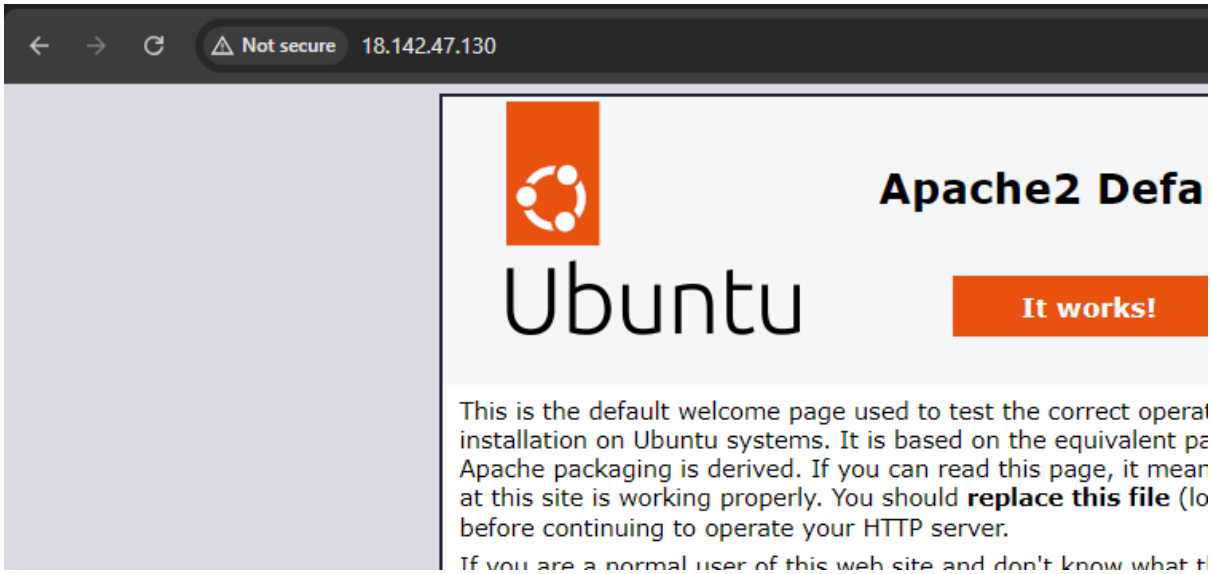
,

```
ubuntu@ip-172-31-28-117:~$ java --version
openjdk 11.0.24 2024-07-16
OpenJDK Runtime Environment (build 11.0.24+8-post-Ubuntu-1ubuntu324.04.1)
OpenJDK 64-Bit Server VM (build 11.0.24+8-post-Ubuntu-1ubuntu324.04.1, mixed mode, s
ubuntu@ip-172-31-28-117:~$
```

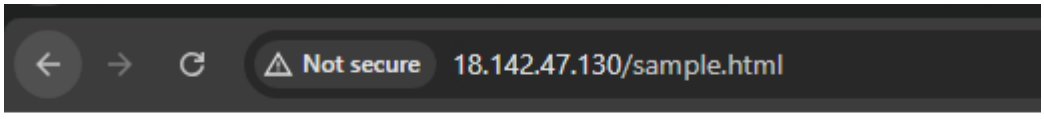
i-0350d5ae72d3ae32c (Slave01)

PublicIPs: 18.142.47.130 PrivateIPs: 172.31.28.117

Public IP of slave 1 to see the webpage of apache 2 server

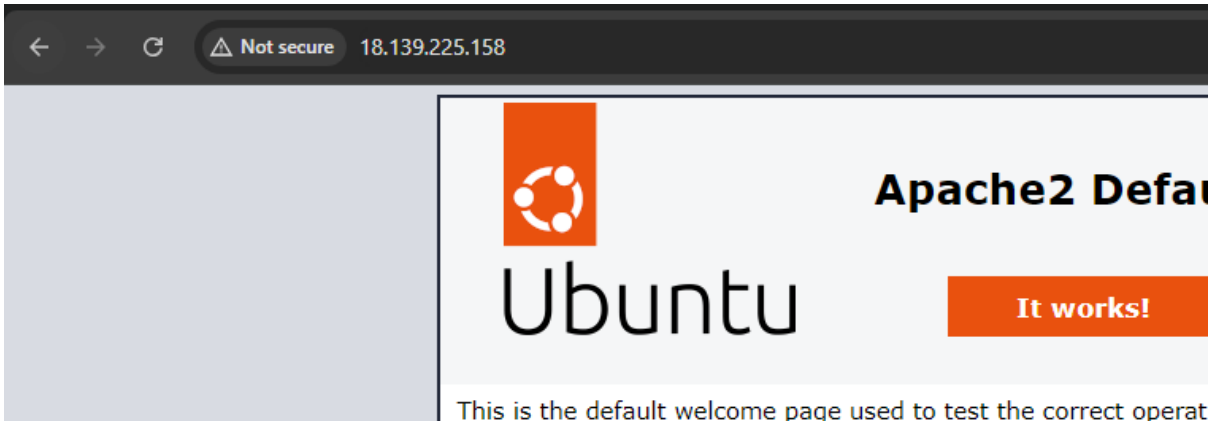


When you provide /sample.html your code will be displayed



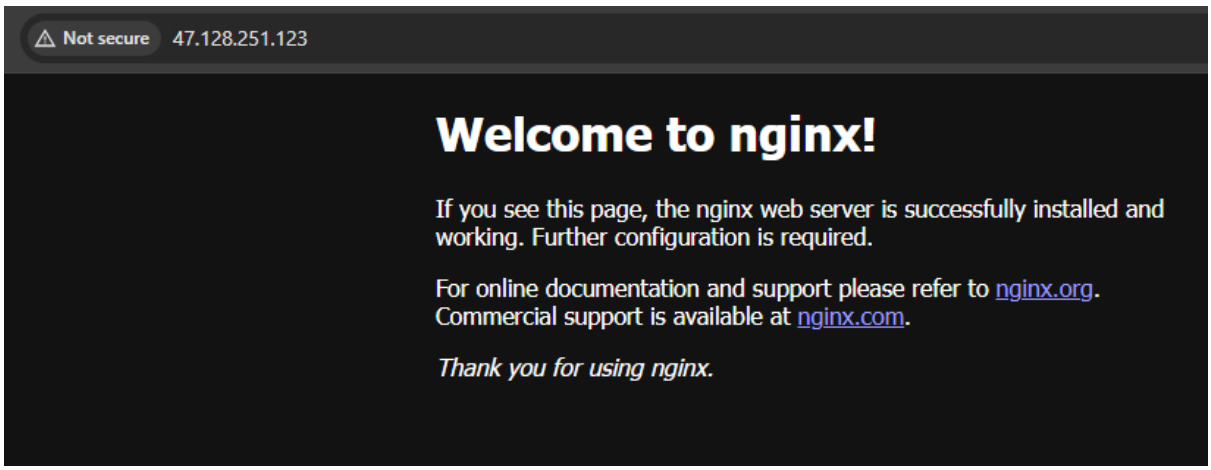
Welcome to Apache Website

Slave2 webpage



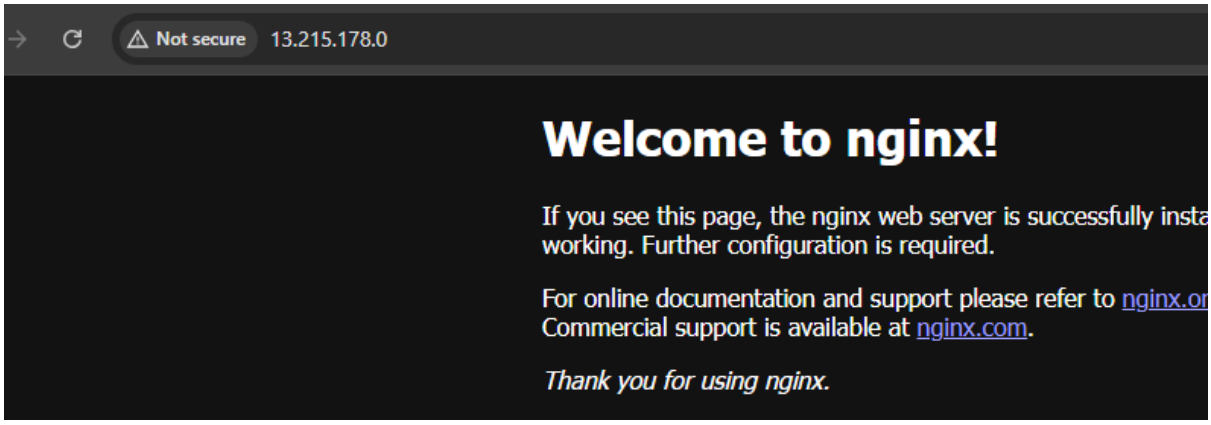
Html page of slave 2

Web page of slave 3



Html page of slave 3

Web page of slave 4



Html page of slave 4