Working with Ubuntu and Linux Web Servers

Muhannad Sinan

> 1. Add two hosts Ubuntu servers and amazon Linux webservers with one instances. Write a playbook to install apache on both of the hosts.
> 2. Add two hosts Ubuntu servers and amazon Linux webservers with one instances. Create a text file sample1.txt in master node of ansible. Now write playbook to copy this to both the hosts.
> 3. Take screen shots of each configuration above

Step1:

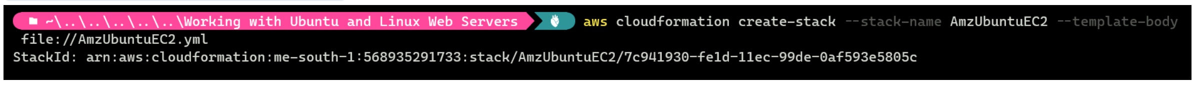
Creating three EC2 instances, one Linux server as an Ansible master, one Amazon Linux web servers and one
Ubuntu server using this CloudFormation template:

`amz-ubuntu-ec2.yml`

```
AWSTemplateFormatVersion: "2010-09-09"
Metadata:
  License: Apache-2.0
Description: "Core Assignment 1 - Working with Ubuntu and Linux Web Servers"
Parameters:
  KeyName:
    Description: Name of an existing EC2 KeyPair to enable SSH access to the
instance
    Type: AWS::EC2::KeyPair::KeyName
    Default: main
    ConstraintDescription: must be the name of an existing EC2 KeyPair.
  InstanceType:
    Description: WebServer EC2 instance type
    Type: String
    Default: t3.micro
    AllowedValues: [t3.nano, t3.micro, t3.small, t3.medium]
    ConstraintDescription: must be a valid EC2 instance type.
  SSHLocation:
    Description: The IP address range that can be used to SSH to the EC2 instances
    Type: String
    MinLength: 9
    MaxLength: 18
    Default: 0.0.0.0/0
    AllowedPattern: (\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})/(\d{1,2})
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
  HTTPLocation:
    Description: The IP address range that can be used to SSH to the EC2 instances
    Type: String
    MinLength: 9
    MaxLength: 18
    Default: 0.0.0.0/0
    AllowedPattern: (\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})/(\d{1,2})
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
  LatestAmzLinuxAmiId:
    Type: "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>"
    Default: "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2"
  LatestUbuntuAmiId:
    Type:  'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>'
    Default:
'/aws/service/canonical/ubuntu/server/20.04/stable/current/amd64/hvm/ebs-gp2/ami-
id'
  InstanceCount:
    Description: Number of EC2 instances (must be between 1 and 5).
    Type: Number
    Default: 1
    MinValue: 1
    MaxValue: 3
    ConstraintDescription: Must be a number between 1 and 5.

Resources:
  MasterSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupName: ansbile-master-sg
      GroupDescription: Enable SSH access via port 22
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: !Ref "SSHLocation"

  NodeSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupName: ansbile-node-sg
      GroupDescription: Enable SSH access via port 22 and HTTP access via port 80
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: !Ref "SSHLocation"
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: !Ref "HTTPLocation"

  EC2InstanceMaster:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: !Ref "InstanceType"
      SecurityGroups: [!Ref "MasterSecurityGroup"]
      KeyName: !Ref "KeyName"
      ImageId: !Ref "LatestAmzLinuxAmiId"
      UserData:
        Fn::Base64: !Sub |
          #!/bin/bash
          sudo yum update -y
          sudo amazon-linux-extras install ansible2 -y
      Tags:
        - Key: Name
          Value: ansible-master
  EC2Instancehost1:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: !Ref "InstanceType"
      SecurityGroups: [!Ref "NodeSecurityGroup"]
      KeyName: !Ref "KeyName"
      ImageId: !Ref "LatestAmzLinuxAmiId"
      Tags:
        - Key: Name
          Value: host1
  EC2Instancehost2:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: !Ref "InstanceType"
      SecurityGroups: [!Ref "NodeSecurityGroup"]
      KeyName: !Ref "KeyName"
      ImageId: !Ref "LatestUbuntuAmiId"
      Tags:
        - Key: Name
          Value: web2

Outputs:
  AnsibleMasterPublicIP:
    Description: Public IP address of the newly created Ansible master EC2 instance
    Value: !GetAtt [EC2InstanceMaster, PublicIp]

  Host1PublicIP:
    Description: Public IP address of the newly created Amazon Linux EC2 instance
    Value: !GetAtt [EC2Instancehost1, PublicIp]

  Host2PublicIP:
    Description: Public IP address of the newly created Ubuntu EC2 instance
```

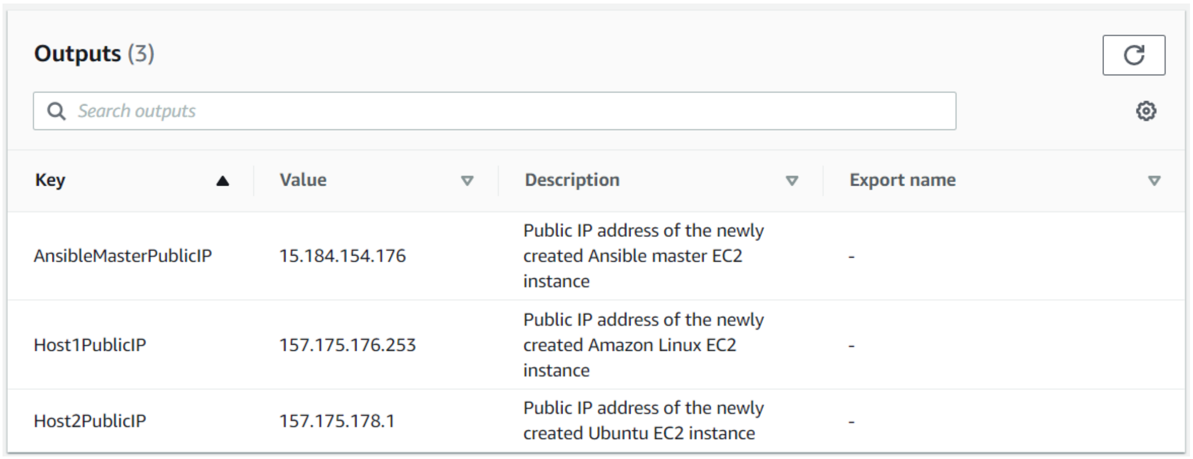
```
        Value: !GetAtt [EC2Instancehost2, PublicIp]
```

Creating the CloudFormation stack using aws cli:
`aws cloudformation create-stack --stack-name AmzUbuntuEC2 --template-body file://AmzUbuntuEC2.yml`



```
~\..\..\..\..\..\Working with Ubuntu and Linux Web Servers  aws cloudformation create-stack --stack-name AmzUbuntuEC2 --template-body
 file://AmzUbuntuEC2.yml
StackId: arn:aws:cloudformation:me-south-1:568935291733:stack/AmzUbuntuEC2/7c941930-fe1d-11ec-99de-0af593e5805c
```

The outputs is:

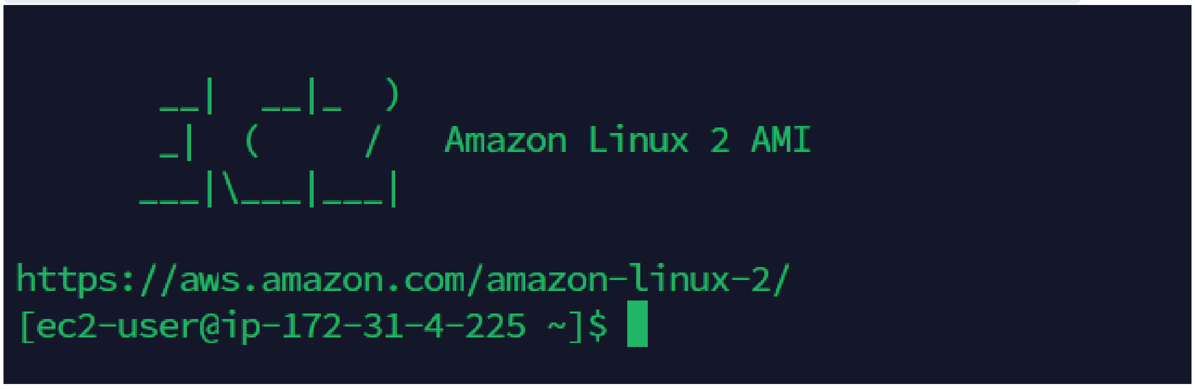


Outputs (3)

Key	Value	Description	Export name
AnsibleMasterPublicIP	15.184.154.176	Public IP address of the newly created Ansible master EC2 instance	-
Host1PublicIP	157.175.176.253	Public IP address of the newly created Amazon Linux EC2 instance	-
Host2PublicIP	157.175.178.1	Public IP address of the newly created Ubuntu EC2 instance	-

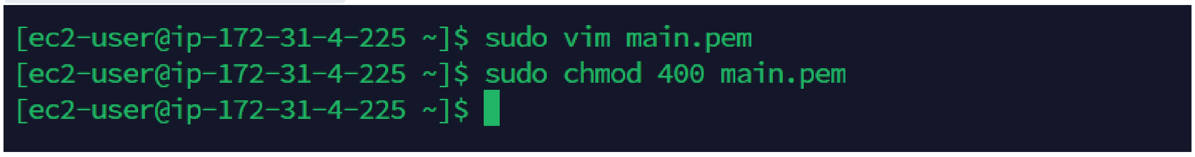
Step2:

Connect to `ansible-master` server using ssh protocol:
`ssh -i "main.pem" ec2-user@ec2-15-184-154-176.me-south-1.compute.amazonaws.com`



```
       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-4-225 ~]$ 
```

Add my private ssh key to the Ansible master server to this directory `/home/ec2-user/main.pem` and change the permissions to 400:
`sudo vim main.pem`
`sudo chmod 400 main.pem`

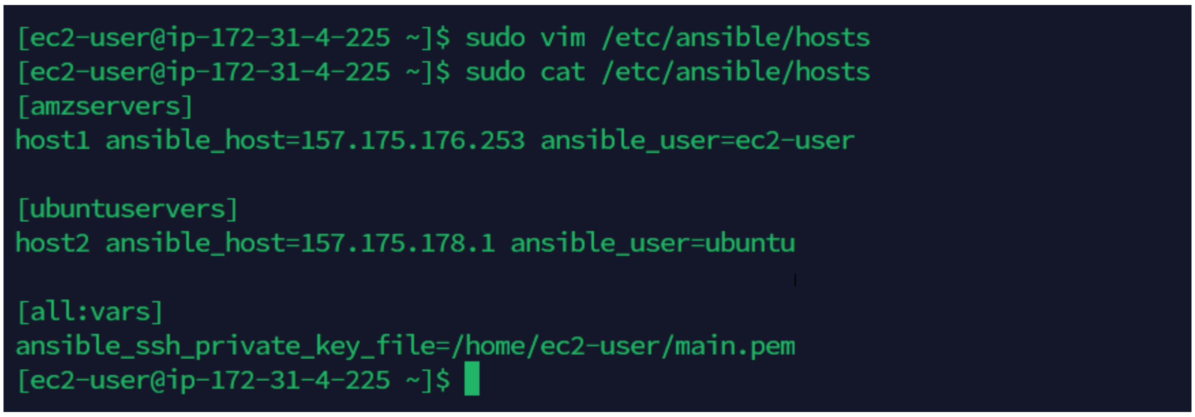


```
[ec2-user@ip-172-31-4-225 ~]$ sudo vim main.pem
[ec2-user@ip-172-31-4-225 ~]$ sudo chmod 400 main.pem
[ec2-user@ip-172-31-4-225 ~]$ 
```

Step3:

Creating an inventory file in the path `/etc/ansible/hosts`
`sudo vim /etc/ansible/hosts`
with the following content:

```
[amzservers]
host1 ansible_host=157.175.176.253 ansible_user=ec2-user

[ubuntuservers]
host2 ansible_host=157.175.178.1 ansible_user=ubuntu

[all:vars]
ansible_ssh_private_key_file=/home/ec2-user/main.pem
```



```
[ec2-user@ip-172-31-4-225 ~]$ sudo vim /etc/ansible/hosts
[ec2-user@ip-172-31-4-225 ~]$ sudo cat /etc/ansible/hosts
[amzservers]
host1 ansible_host=157.175.176.253 ansible_user=ec2-user

[ubuntuservers]
host2 ansible_host=157.175.178.1 ansible_user=ubuntu

[all:vars]
ansible_ssh_private_key_file=/home/ec2-user/main.pem
[ec2-user@ip-172-31-4-225 ~]$
```

Step4:

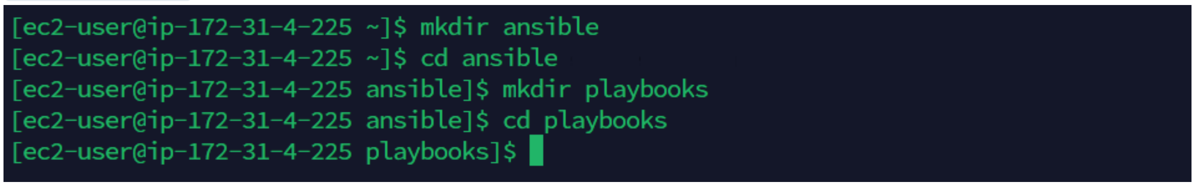
Let us create a new directory with name `playbooks` under `ansible` folder

`mkdir ansible`

`cd ansible`

`mkdir playbooks`

`cd playbooks`

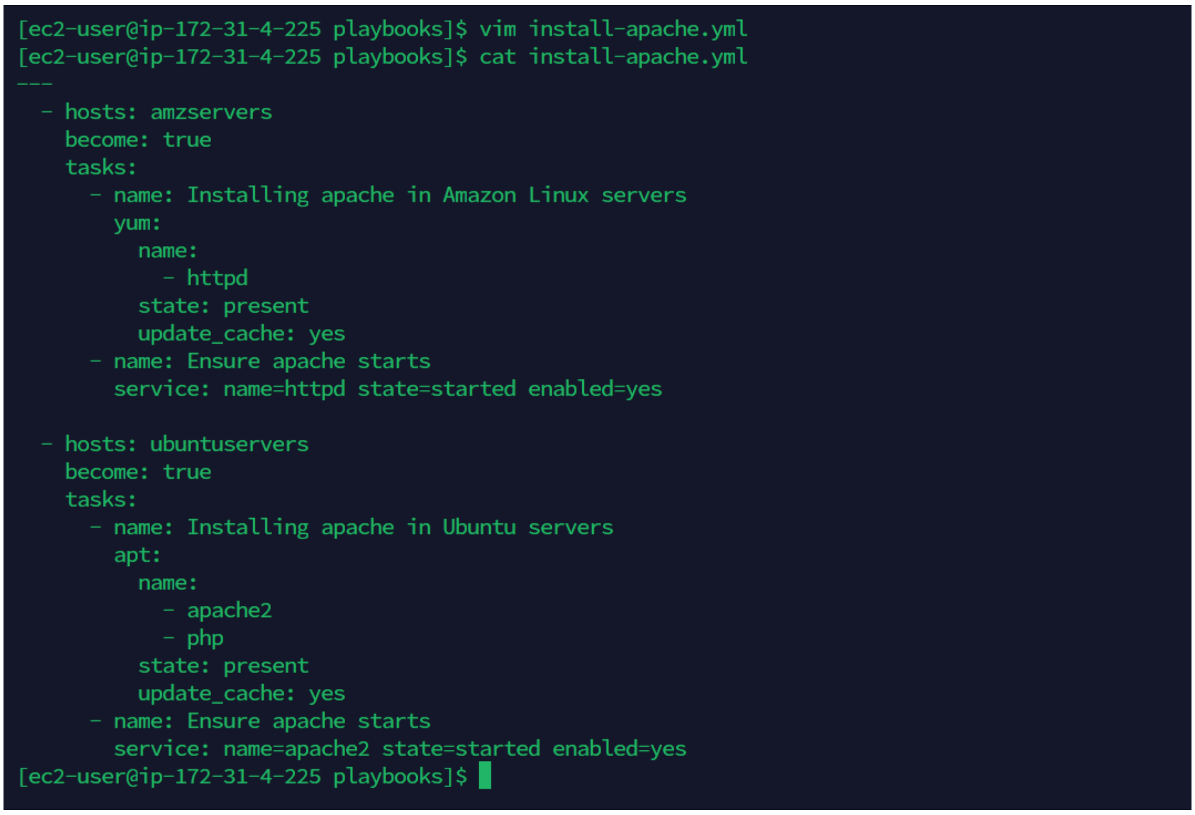


```
[ec2-user@ip-172-31-4-225 ~]$ mkdir ansible
[ec2-user@ip-172-31-4-225 ~]$ cd ansible
[ec2-user@ip-172-31-4-225 ansible]$ mkdir playbooks
[ec2-user@ip-172-31-4-225 ansible]$ cd playbooks
[ec2-user@ip-172-31-4-225 playbooks]$
```

Writing a playbook to install apache on both of the hosts

`vim install-apache.yml`

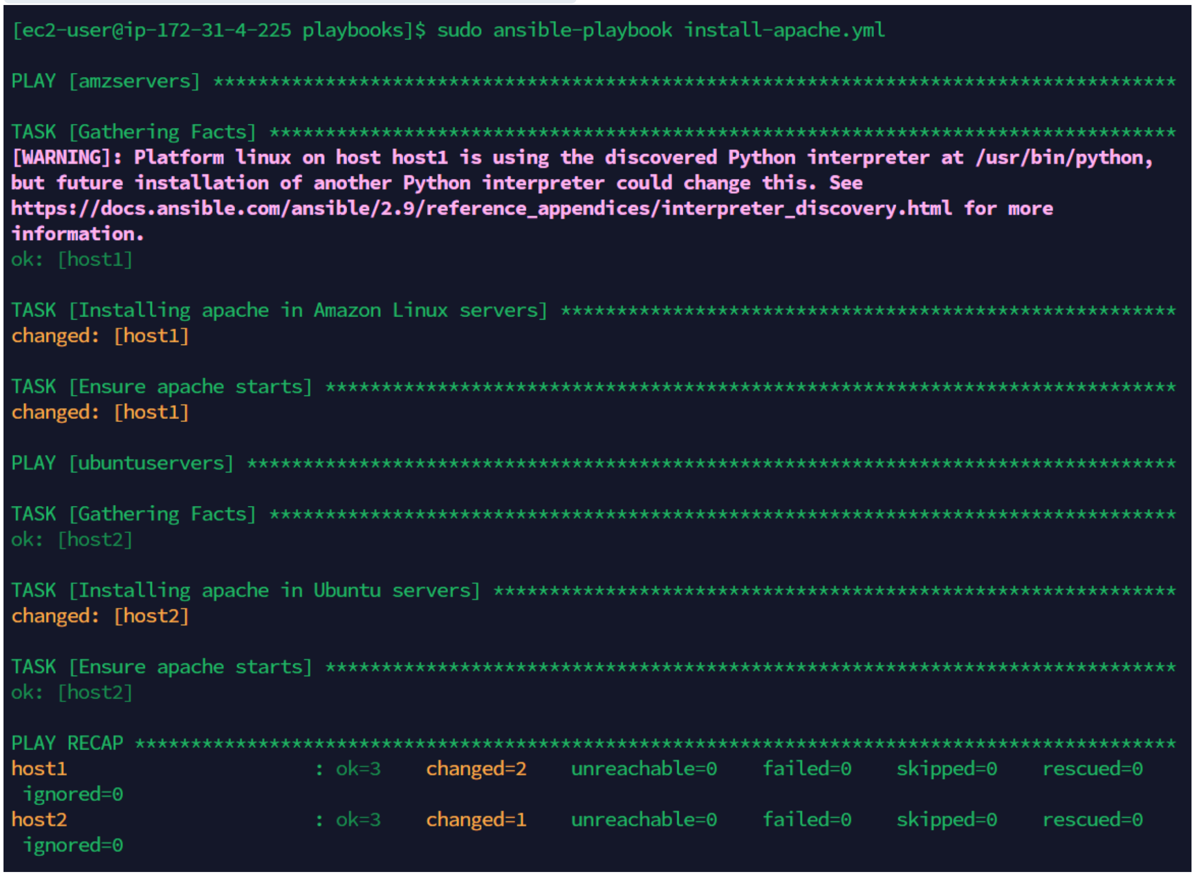
```yaml
---
 - hosts: amzservers
   become: true
   tasks:
     - name: Installing apache in Amazon Linux servers
       yum:
         name:
           - httpd
         state: present
         update_cache: yes
     - name: Ensure apache starts
       service: name=httpd state=started enabled=yes

 - hosts: ubuntuservers
   become: true
   tasks:
     - name: Installing apache in Ubuntu servers
       apt:
         name:
           - apache2
           - php
         state: present
         update_cache: yes
     - name: Ensure apache starts
       service: name=apache2 state=started enabled=yes
```



```
[ec2-user@ip-172-31-4-225 playbooks]$ vim install-apache.yml
[ec2-user@ip-172-31-4-225 playbooks]$ cat install-apache.yml
---
  - hosts: amzservers
    become: true
    tasks:
      - name: Installing apache in Amazon Linux servers
        yum:
          name:
            - httpd
          state: present
          update_cache: yes
      - name: Ensure apache starts
        service: name=httpd state=started enabled=yes

  - hosts: ubuntuservers
    become: true
    tasks:
      - name: Installing apache in Ubuntu servers
        apt:
          name:
            - apache2
            - php
          state: present
          update_cache: yes
      - name: Ensure apache starts
        service: name=apache2 state=started enabled=yes
[ec2-user@ip-172-31-4-225 playbooks]$
```

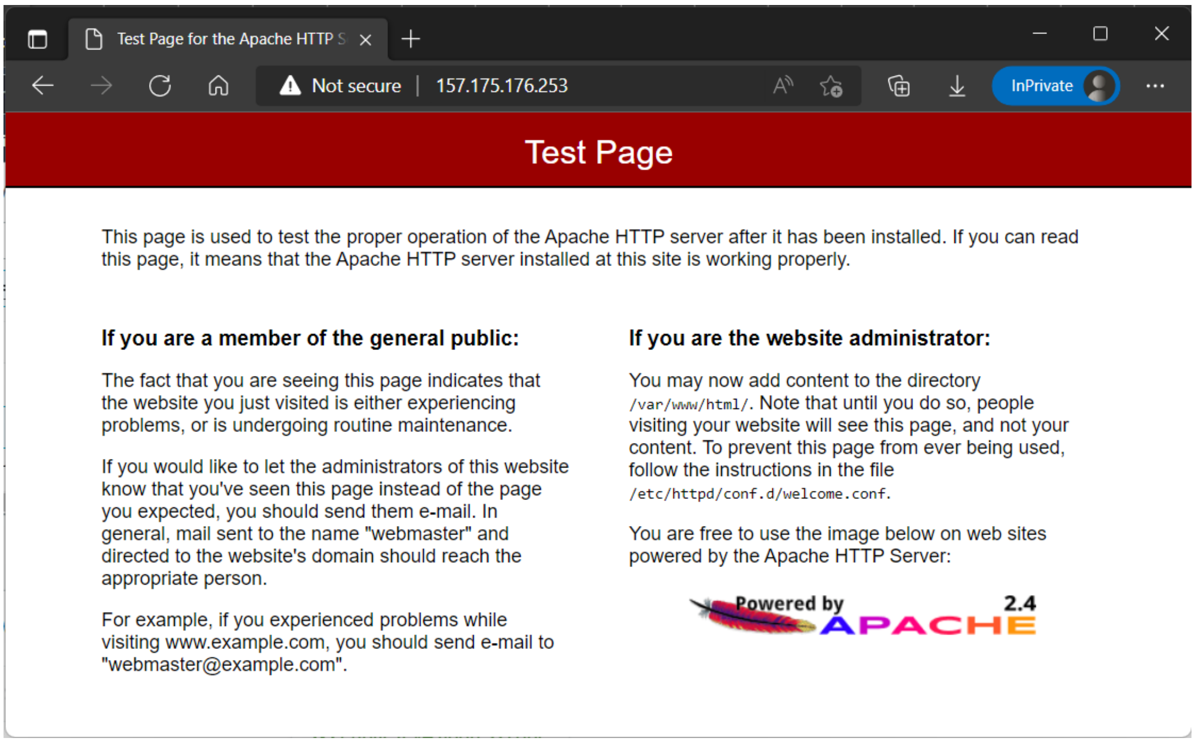
Run the following command in the folder where `install-apache.yml` file is saved

`sudo ansible-playbook install-apache.yml`



```
[ec2-user@ip-172-31-4-225 playbooks]$ sudo ansible-playbook install-apache.yml

PLAY [amzservers] ***********************************************************************************

TASK [Gathering Facts] ******************************************************************************
[WARNING]: Platform linux on host host1 is using the discovered Python interpreter at /usr/bin/python,
but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more
information.
ok: [host1]

TASK [Installing apache in Amazon Linux servers] ****************************************************
changed: [host1]

TASK [Ensure apache starts] *************************************************************************
changed: [host1]

PLAY [ubuntuservers] ********************************************************************************

TASK [Gathering Facts] ******************************************************************************
ok: [host2]

TASK [Installing apache in Ubuntu servers] **********************************************************
changed: [host2]

TASK [Ensure apache starts] *************************************************************************
ok: [host2]

PLAY RECAP ******************************************************************************************
host1                      : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0
 ignored=0
host2                      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
 ignored=0
```

Host1 Amazon Linux



Test Page

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

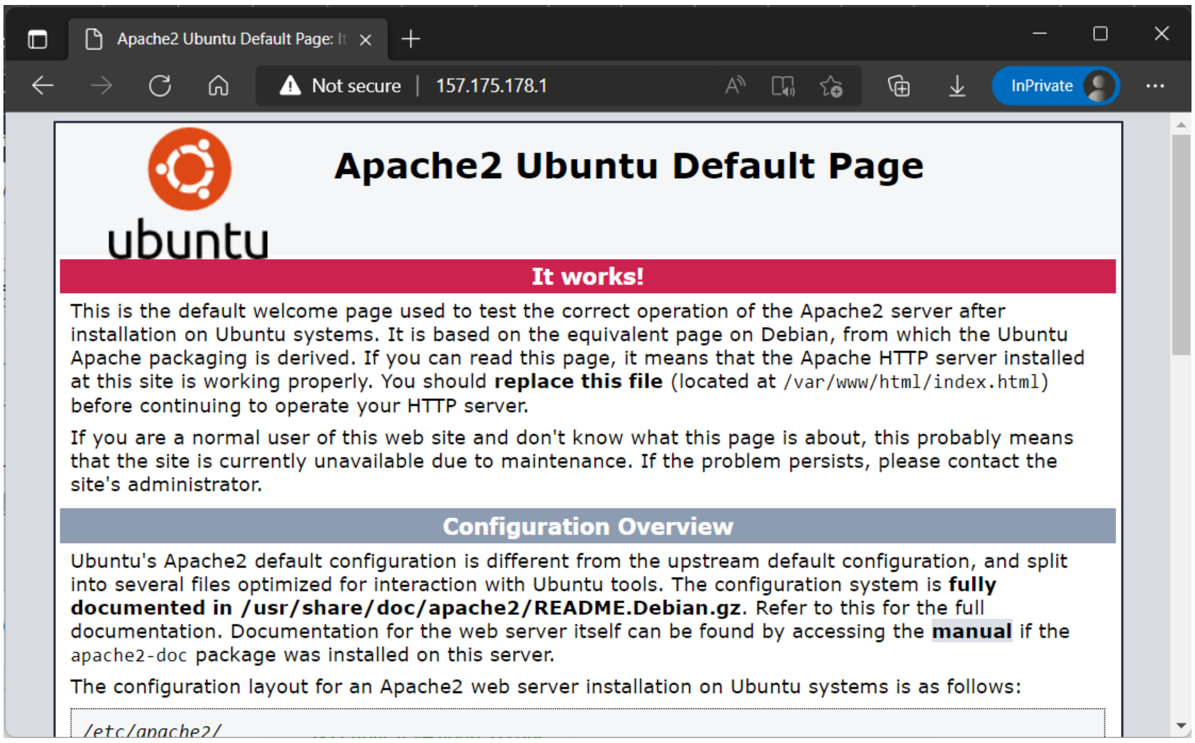
For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server:

Host2 Ubuntu



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

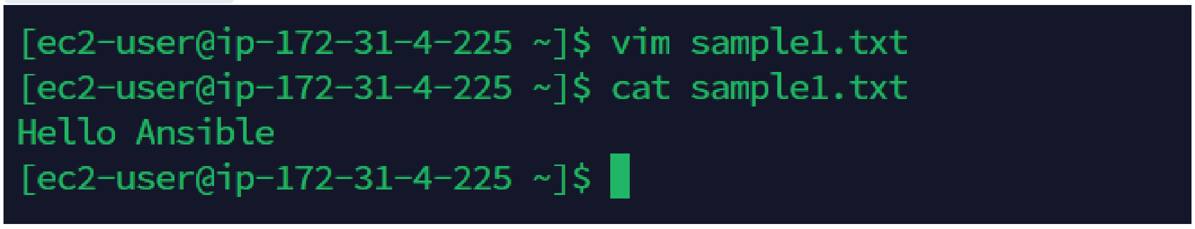
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
```

Step5:

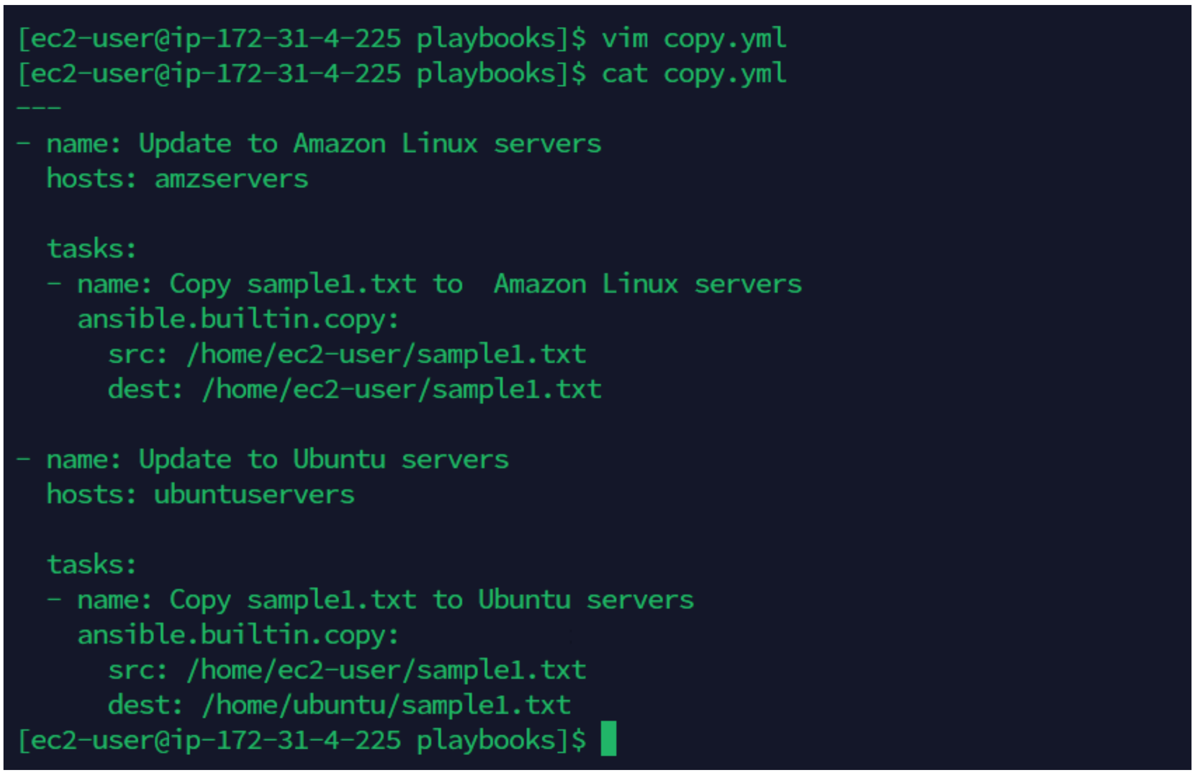
Creating a text file `sample1.txt`:
`vim sample1.txt`



```
[ec2-user@ip-172-31-4-225 ~]$ vim sample1.txt
[ec2-user@ip-172-31-4-225 ~]$ cat sample1.txt
Hello Ansible
[ec2-user@ip-172-31-4-225 ~]$
```

Writing a playbook to copy this to both the hosts:
`vim copy.yml`

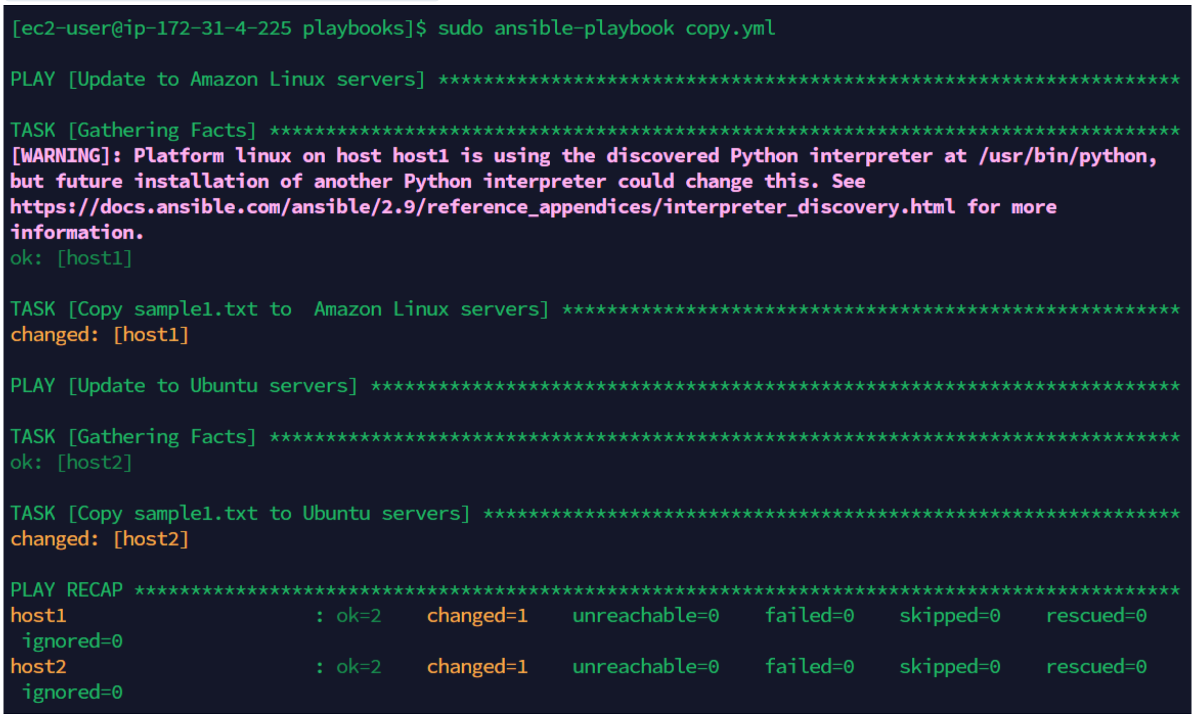
```yaml
---
- name: Update to Amazon Linux servers
  hosts: amzservers

  tasks:
  - name: Copy sample1.txt to  Amazon Linux servers
    ansible.builtin.copy:
      src: /home/ec2-user/sample1.txt
      dest: /home/ec2-user/sample1.txt

- name: Update to Ubuntu servers
  hosts: ubuntuservers

  tasks:
  - name: Copy sample1.txt to Ubuntu servers
    ansible.builtin.copy:
      src: /home/ec2-user/sample1.txt
      dest: /home/ubuntu/sample1.txt
```



```
[ec2-user@ip-172-31-4-225 playbooks]$ vim copy.yml
[ec2-user@ip-172-31-4-225 playbooks]$ cat copy.yml
---
- name: Update to Amazon Linux servers
  hosts: amzservers

  tasks:
  - name: Copy sample1.txt to  Amazon Linux servers
    ansible.builtin.copy:
      src: /home/ec2-user/sample1.txt
      dest: /home/ec2-user/sample1.txt

- name: Update to Ubuntu servers
  hosts: ubuntuservers

  tasks:
  - name: Copy sample1.txt to Ubuntu servers
    ansible.builtin.copy:
      src: /home/ec2-user/sample1.txt
      dest: /home/ubuntu/sample1.txt
[ec2-user@ip-172-31-4-225 playbooks]$
```

Run the following command in the folder where `copy.yml` file is saved

`sudo ansible-playbook copy.yml`



```
[ec2-user@ip-172-31-4-225 playbooks]$ sudo ansible-playbook copy.yml

PLAY [Update to Amazon Linux servers] *********************************************************************

TASK [Gathering Facts] ************************************************************************************
[WARNING]: Platform linux on host host1 is using the discovered Python interpreter at /usr/bin/python,
but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more
information.
ok: [host1]

TASK [Copy sample1.txt to  Amazon Linux servers] **********************************************************
changed: [host1]

PLAY [Update to Ubuntu servers] ***************************************************************************

TASK [Gathering Facts] ************************************************************************************
ok: [host2]

TASK [Copy sample1.txt to Ubuntu servers] *****************************************************************
changed: [host2]

PLAY RECAP ************************************************************************************************
host1                      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
 ignored=0
host2                      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
 ignored=0
```