

Homework 1

Kevin Ma - 010200411

Github link: <https://github.com/vetag/cmpe-172>

The purpose of this assignment was to create a website utilizing AWS and Ansible, reading "Hello World!".

To start, Ansible was installed and an AWS EC2 instance was set up.

Screenshot of Ansible setup confirmation:

```
kevin@kevin-VirtualBox:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/kevin/.ssh/id_rsa):
Created directory '/home/kevin/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/kevin/.ssh/id_rsa.
Your public key has been saved in /home/kevin/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:nhxLXlbxzjRITKWh8YlK4ktI52+AwsVf703CQEqznY kevin@kevin-VirtualBox
The key's randomart image is:
+---[RSA 2048]---+
|      .    ..00=..      |
|      o . . B.B        |
|    + = o = * +        |
|  + = B + . + = .      |
|  = * E S + + o        |
|    o o O * . .        |
|      . O              |
|      .                |
+-----[SHA256]-----+
```

Screenshot of AWS EC2 connection success:

```
kevin@kevin-VirtualBox:~$ ssh -i /home/kevin/Downloads/cmpe172.pem ec2-user@ec2-13-57-184-10.us-west-1.compute.amazonaws.com
The authenticity of host 'ec2-13-57-184-10.us-west-1.compute.amazonaws.com (13.57.184.10)' can't be established.
ECDSA key fingerprint is SHA256:zhPgSntKcQsoxFsb00uLmtxmQIusI7sCIsnJdsK0Kj0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-13-57-184-10.us-west-1.compute.amazonaws.com,13.57.184.10' (ECDSA) to the list of known hosts.

  ____|_____|_____)
  ____|_____|_____| /   Amazon Linux AMI
  ____|_____|_____|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
2 package(s) needed for security, out of 3 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-15-73 ~]$
```

After setting up the AWS EC2 connection, we want to set up the Ansible Hosts file to include the IPv4 public IP and also connect the ssh private key file.

Screenshot of hosts file:

```
[webservers:vars]
ansible_ssh_private_key_file=./cmpe172.pem

[webservers]
13.57.184.10
```

Now, we can confirm the link between AWS and Ansible by pinging our host.

Screenshot of successful ansible ping:

```
kevin@kevin-VirtualBox:/etc/ansible$ ansible -i hosts all -m ping -u ec2-user
13.57.184.10 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
kevin@kevin-VirtualBox:/etc/ansible$
```

Because our ansible ping is successful, we can start setting up the Ansible playbook to set up the server.

Screenshot of Ansible Playbook:

```
---

- hosts: webservers
  remote_user: ec2-user

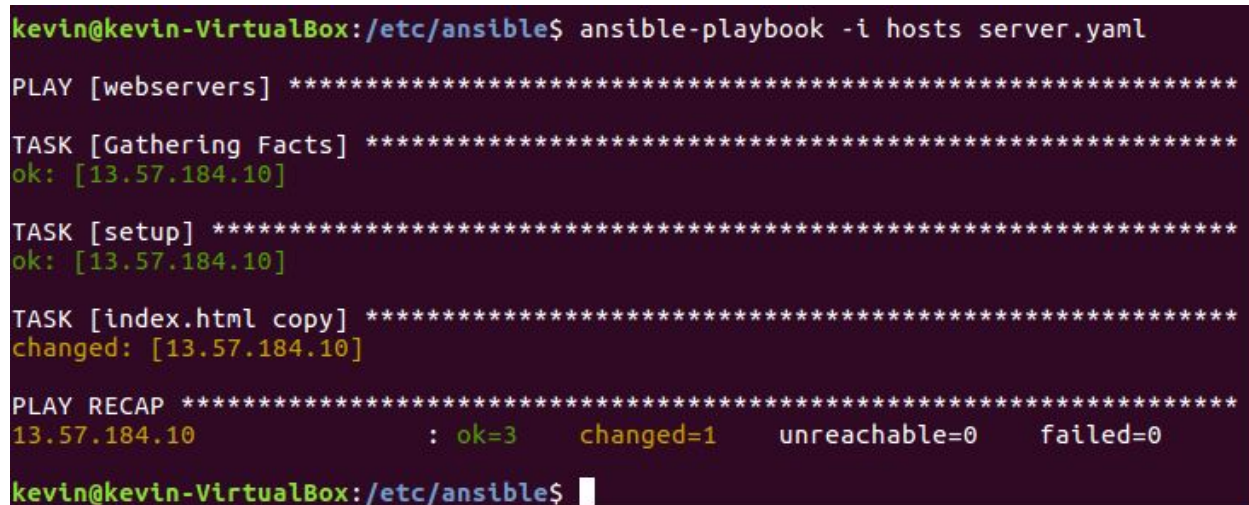
  vars:
    - WebsiteText: "Hello World!"

  tasks:
    - name: setup
      become: true
      yum: pkg=nginx state=installed update_cache=true

    - name: index.html copy
      become: true
      template: src=index.html.j2 dest=/home/ec2-user/nginx/index.html

---
```

Screenshot of Ansible Playbook in action:

A terminal window with a dark purple background and light green text. The prompt is 'kevin@kevin-VirtualBox:/etc/ansible\$'. The command 'ansible-playbook -i hosts server.yaml' has been executed. The output shows the playbook 'webserver' running on host '13.57.184.10'. It includes tasks for 'Gathering Facts' (ok), 'setup' (ok), and 'index.html copy' (changed). A recap line shows 'ok=3', 'changed=1', 'unreachable=0', and 'failed=0'. The prompt returns to 'kevin@kevin-VirtualBox:/etc/ansible\$'.

```
kevin@kevin-VirtualBox:/etc/ansible$ ansible-playbook -i hosts server.yaml
PLAY [webservers] *****
TASK [Gathering Facts] *****
ok: [13.57.184.10]
TASK [setup] *****
ok: [13.57.184.10]
TASK [index.html copy] *****
changed: [13.57.184.10]
PLAY RECAP *****
13.57.184.10 : ok=3    changed=1    unreachable=0    failed=0
kevin@kevin-VirtualBox:/etc/ansible$
```

Now that the playbook works, the server should be able to run by typing the IP address into the address bar.

Screenshot of website working:



Ansible Demo

Hello World!

Now that resources are deployed, we can write a playbook to undeploy the resources.

Screenshot of ansible playbook to undeploy:

```
---

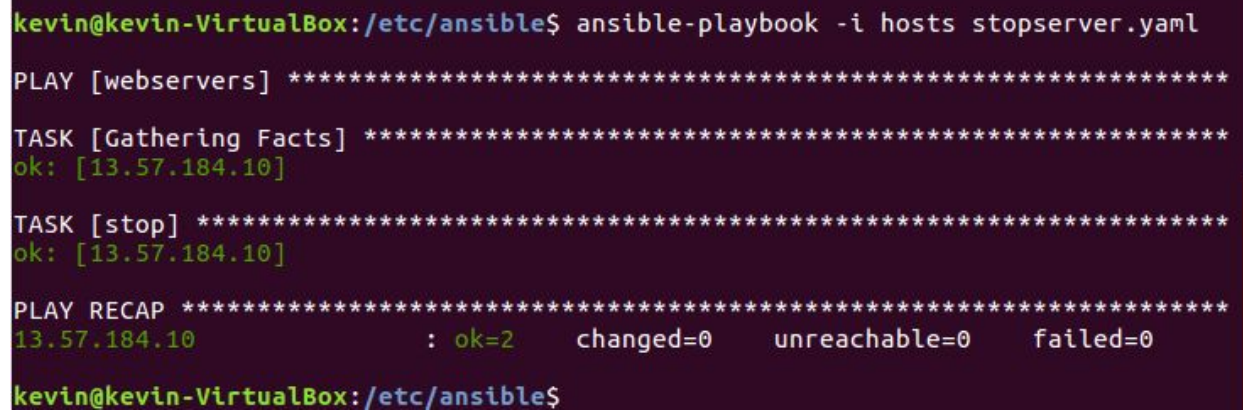
- hosts: webservers
  remote_user: ec2-user

  tasks:
    - name: stop
      become: true
      service: name=nginx state=stopped

...

```

Screenshot of undeploy in terminal:



```
kevin@kevin-VirtualBox:/etc/ansible$ ansible-playbook -i hosts stopserver.yaml
PLAY [webservers] *****
TASK [Gathering Facts] *****
ok: [13.57.184.10]
TASK [stop] *****
ok: [13.57.184.10]
PLAY RECAP *****
13.57.184.10      : ok=2    changed=0    unreachable=0    failed=0
kevin@kevin-VirtualBox:/etc/ansible$
```

Screenshot of webpage not working:

