HW #1 – Ansible Submission Question

- Configure two VMs, **VM1** and **VM2** either on your own hardware, or in a cloud environment. Configure Ansible to deploy a webserver on ***VM1*** and ***VM2*** on port 8080 with a web page that is accessible from a web browser, and displays the message: "Hello World from SJSU-X" where X is 1 or 2 depending on which webserver instance, VM1 or VM2.
- Include in the Ansible playbook, plays to **deploy** and **un-deploy** the webserver resources
- Submit a Word document, with screenshots showing your work, and a demo, and all ansible code/scripts via github

Submitted By:- Praful John

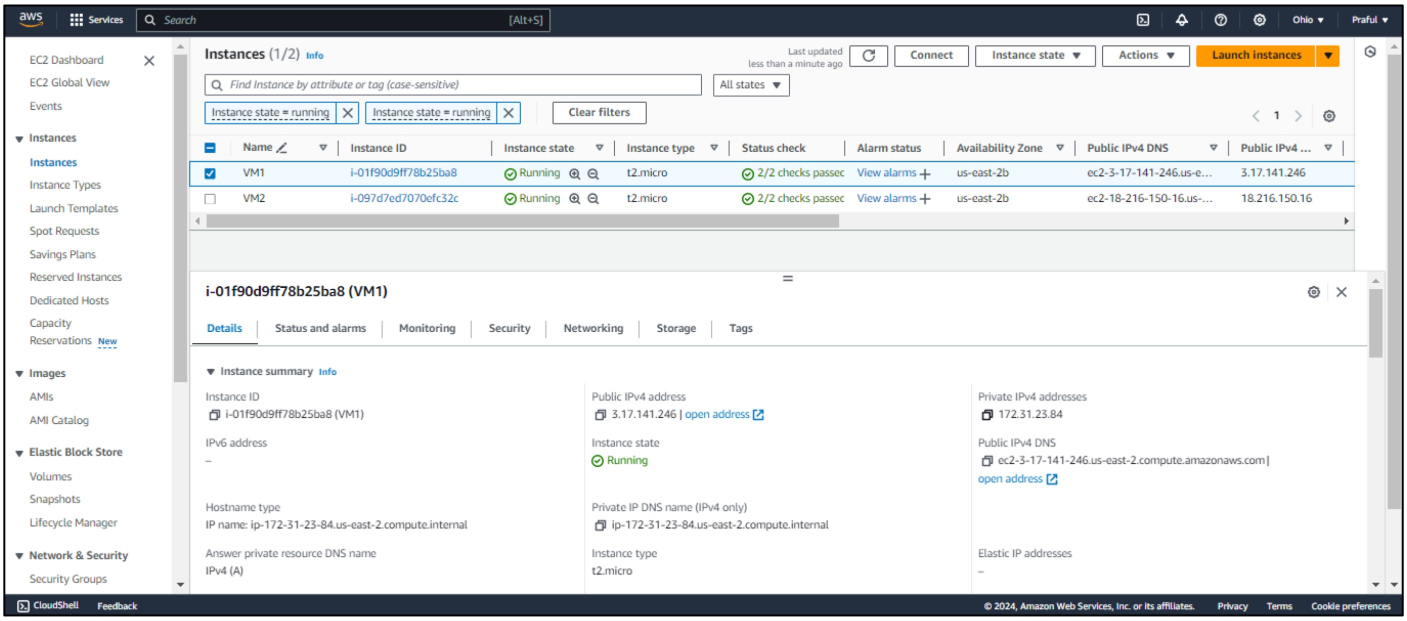
FA24: CMPE-272 Sec 49

GitHub repo link:- https://github.com/Praful-John2409/Ansible.git

1. Configuring the two Virtual Machines in a cloud environment: (I have used AWS EC2.)

steps used by me to configure the cloud settings:-

i. First, created EC2 instances.
   Go to EC2 after logging into the AWS Console.
   Make two virtual machines (VM1 and VM2):
ii. System of Operation: Any compatible OS (I chose Ubuntu).
iii. Type of Instance: Select t2.micro or a comparable type.
iv. Security Unit: Permit incoming connections on SSH port 22 and HTTP port 8080.
    Key Pair: To access the VMs, generate a new SSH key pair or utilize an existing one.
v. Obtain the public IPs.
vi. Once the instances are running, see what their public IP addresses are. These IP addresses are what Ansible will use to connect and configure.



Instances (1/2) Info — Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive)

Instance state = running ☒ Instance state = running ☒

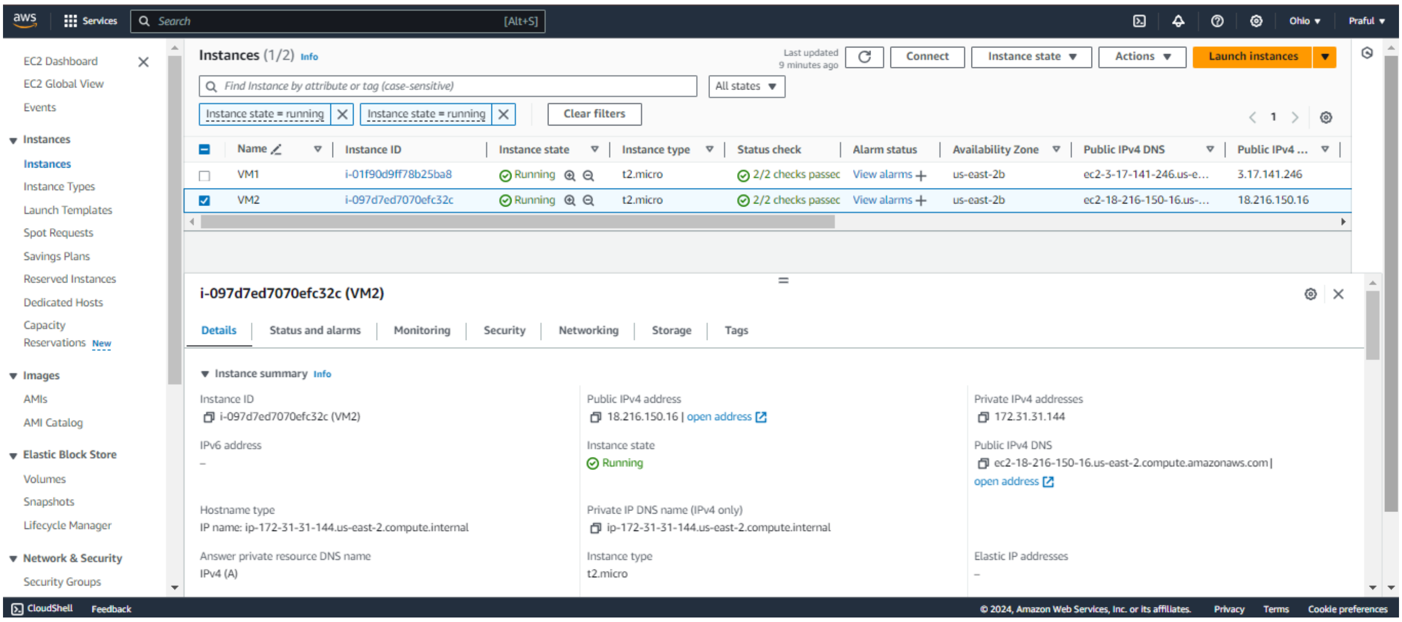
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
☑	VM1	i-01f90d9ff78b25ba8	Running	t2.micro	2/2 checks passec	View alarms +	us-east-2b	ec2-3-17-141-246.us-e...	3.17.141.246
□	VM2	i-097d7ed7070efc32c	Running	t2.micro	2/2 checks passec	View alarms +	us-east-2b	ec2-18-216-150-16.us-...	18.216.150.16

i-01f90d9ff78b25ba8 (VM1)

Instance summary Info

Instance ID: i-01f90d9ff78b25ba8 (VM1)
Public IPv4 address: 3.17.141.246 | open address
Private IPv4 addresses: 172.31.23.84
IPv6 address: –
Instance state: Running
Public IPv4 DNS: ec2-3-17-141-246.us-east-2.compute.amazonaws.com | open address
Hostname type: IP name: ip-172-31-23-84.us-east-2.compute.internal
Private IP DNS name (IPv4 only): ip-172-31-23-84.us-east-2.compute.internal
Answer private resource DNS name: IPv4 (A)
Instance type: t2.micro
Elastic IP addresses: –

Fig1: Details of the VM1



Instances (1/2) Info — Last updated 9 minutes ago

Find Instance by attribute or tag (case-sensitive)

Instance state = running ☒ Instance state = running ☒

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
□	VM1	i-01f90d9ff78b25ba8	Running	t2.micro	2/2 checks passec	View alarms +	us-east-2b	ec2-3-17-141-246.us-e...	3.17.141.246
☑	VM2	i-097d7ed7070efc32c	Running	t2.micro	2/2 checks passec	View alarms +	us-east-2b	ec2-18-216-150-16.us-...	18.216.150.16

i-097d7ed7070efc32c (VM2)

Instance summary Info

Instance ID: i-097d7ed7070efc32c (VM2)
Public IPv4 address: 18.216.150.16 | open address
Private IPv4 addresses: 172.31.31.144
IPv6 address: –
Instance state: Running
Public IPv4 DNS: ec2-18-216-150-16.us-east-2.compute.amazonaws.com | open address
Hostname type: IP name: ip-172-31-31-144.us-east-2.compute.internal
Private IP DNS name (IPv4 only): ip-172-31-31-144.us-east-2.compute.internal
Answer private resource DNS name: IPv4 (A)
Instance type: t2.micro
Elastic IP addresses: –

Fig2: Details of the VM2

2) Configure Ansible to Deploy a Web Server on VM1 and VM2

Ansible will help automate the deployment of the web server like NGINX or Apache(I used Apache) on both instances. The server will run on port 8080, displaying a custom message based on the instance.

Step 1: Set up Ansible on Windows with Ubuntu WSL

1. Install Windows Subsystem for Linux (WSL)

   - Opening PowerShell as an Administrator and running

     wsl --install

2. Installing Ubuntu from the Microsoft Store.

3. Installing Ansible on Ubuntu WSL (on ubuntu's bash)

   sudo apt update

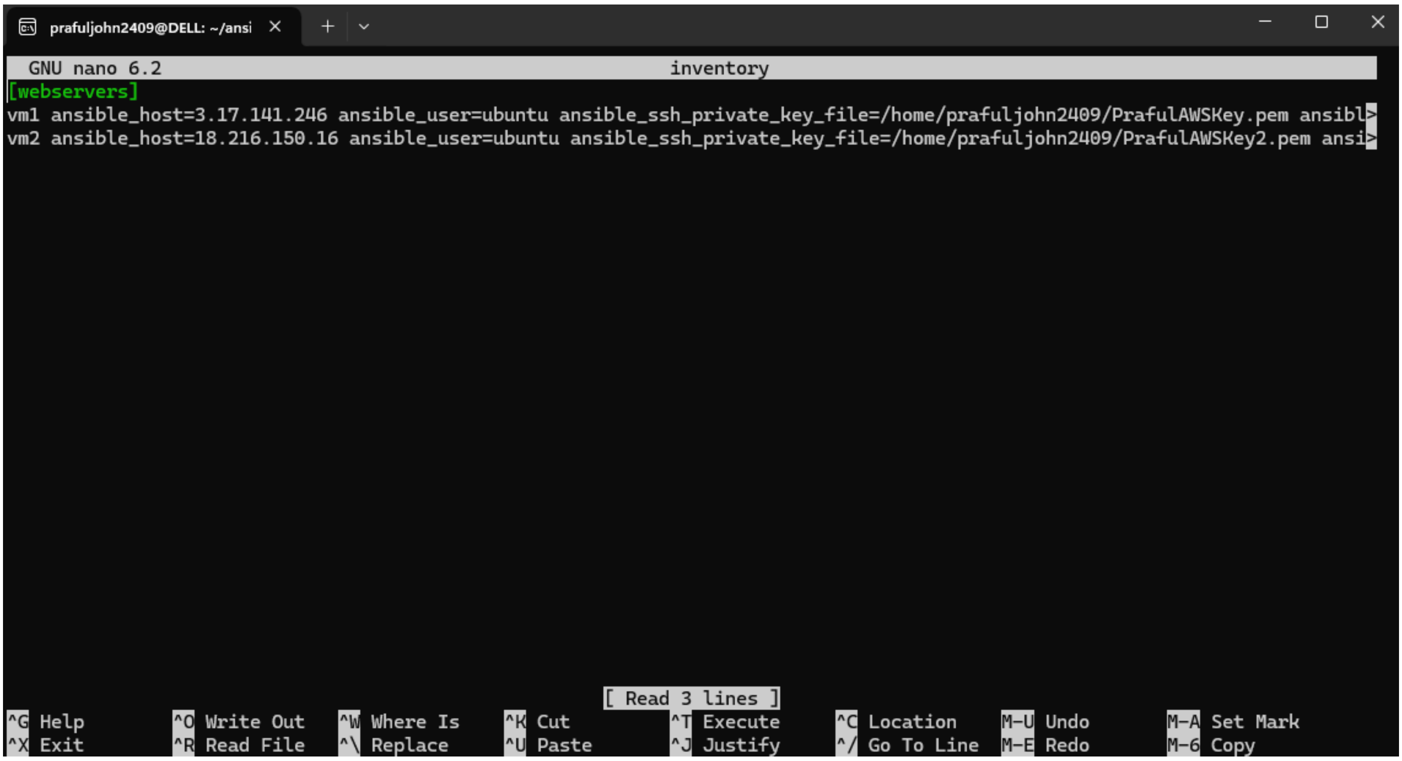
   sudo apt install ansible

Step 2: Configure Ansible Inventory

Creating an Ansible inventory file (inventory) to define your EC2 instances (VM1 and VM2).

The command used to do so:-

nano inventory

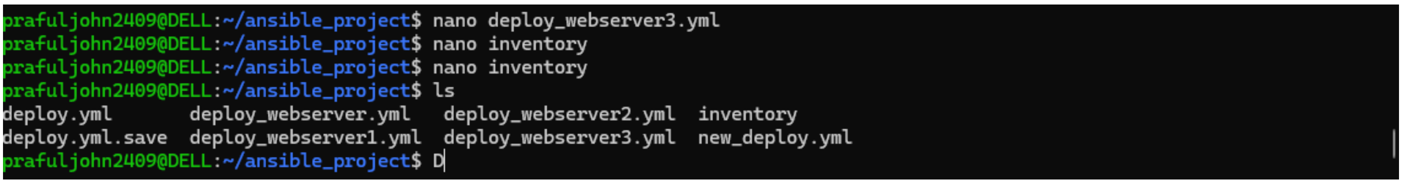


```
  GNU nano 6.2                                              inventory
[webservers]
vm1 ansible_host=3.17.141.246 ansible_user=ubuntu ansible_ssh_private_key_file=/home/prafuljohn2409/PrafulAWSKey.pem ansibl>
vm2 ansible_host=18.216.150.16 ansible_user=ubuntu ansible_ssh_private_key_file=/home/prafuljohn2409/PrafulAWSKey2.pem ansi>

                                                        [ Read 3 lines ]
^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location     M-U Undo        M-A Set Mark
^X Exit         ^R Read File    ^\ Replace      ^U Paste        ^J Justify      ^/ Go To Line   M-E Redo        M-6 Copy
```

Fig 3: Creating inventory for Ansible

Step 3: Create Ansible Playbook for Web Server Deployment

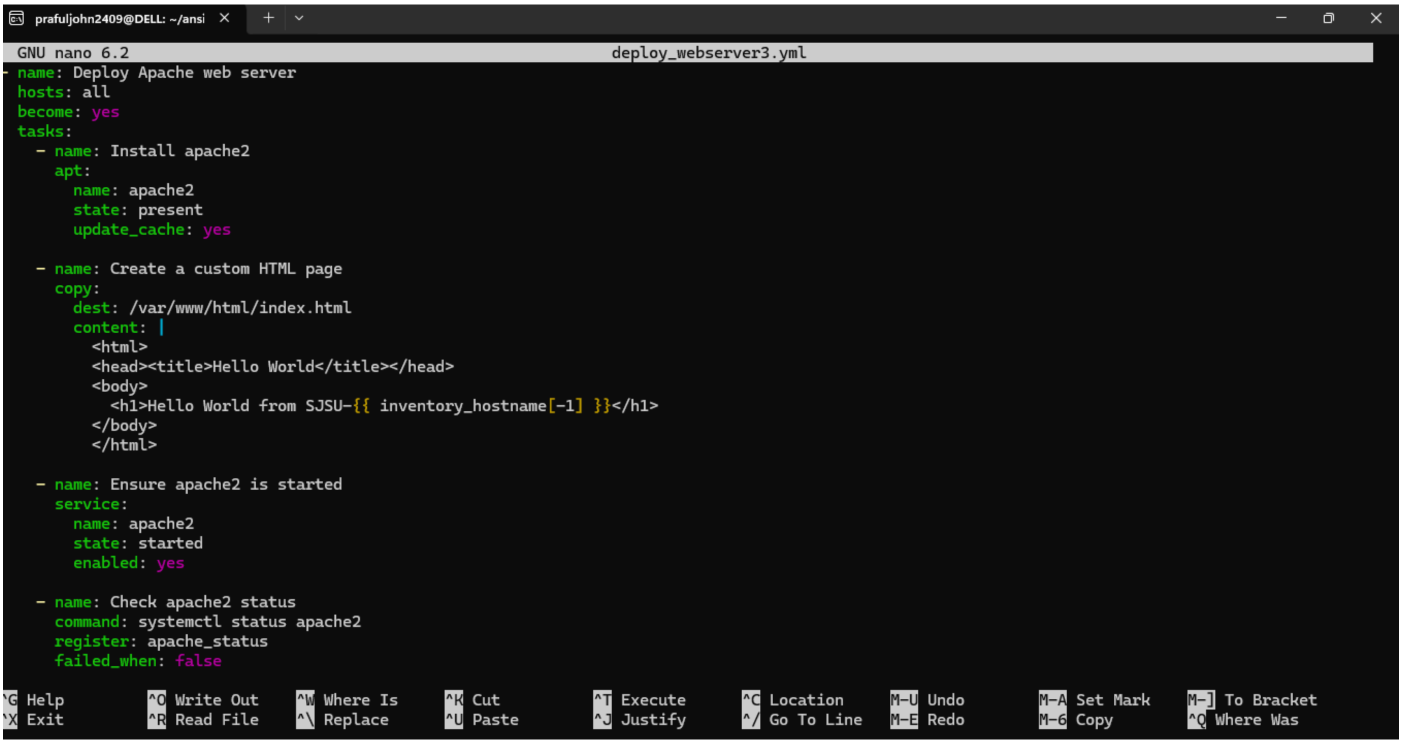
Create a playbook file ‘deploy_webserver3.yml’ that installs a web server (Apache in this example) and configures it to display a customized message on port 8080. I played around with ansible by making multiple playbooks they will be enlisted In the screenshots



```
prafuljohn2409@DELL:~/ansible_project$ nano deploy_webserver3.yml
prafuljohn2409@DELL:~/ansible_project$ nano inventory
prafuljohn2409@DELL:~/ansible_project$ nano inventory
prafuljohn2409@DELL:~/ansible_project$ ls
deploy.yml       deploy_webserver.yml   deploy_webserver2.yml  inventory
deploy.yml.save  deploy_webserver1.yml  deploy_webserver3.yml  new_deploy.yml
prafuljohn2409@DELL:~/ansible_project$ D
```

Fig4:list of files in Ansible Project

The playbook used for deployment:-



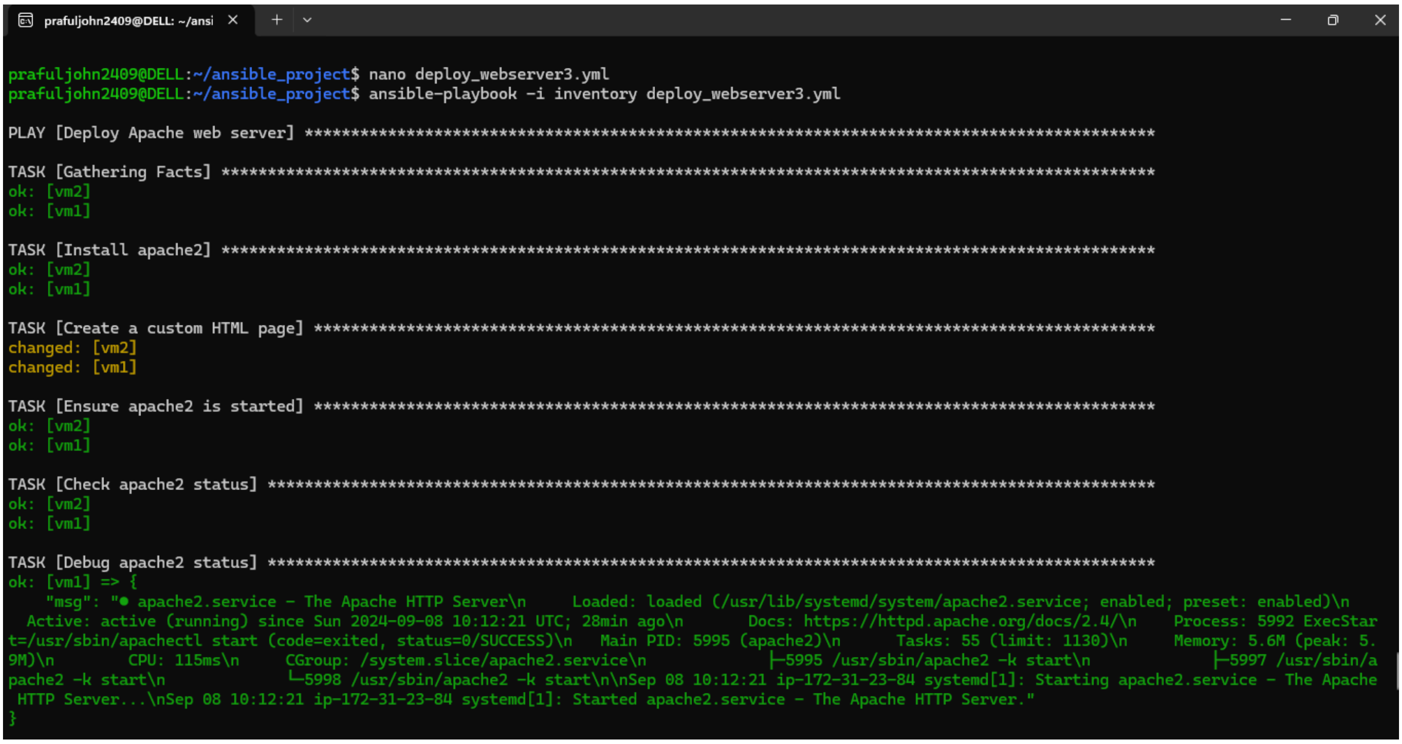
```
- name: Deploy Apache web server
  hosts: all
  become: yes
  tasks:
    - name: Install apache2
      apt:
        name: apache2
        state: present
        update_cache: yes

    - name: Create a custom HTML page
      copy:
        dest: /var/www/html/index.html
        content: |
          <html>
          <head><title>Hello World</title></head>
          <body>
            <h1>Hello World from SJSU-{{ inventory_hostname[-1] }}</h1>
          </body>
          </html>

    - name: Ensure apache2 is started
      service:
        name: apache2
        state: started
        enabled: yes

    - name: Check apache2 status
      command: systemctl status apache2
      register: apache_status
      failed_when: false
```

Fig5: The playbook ‘deploy_webserver3.yml’

Step 4: Run the Playbook

To deploy the web server(command used):

ansible-playbook -i inventory deploy_webserver3.yml

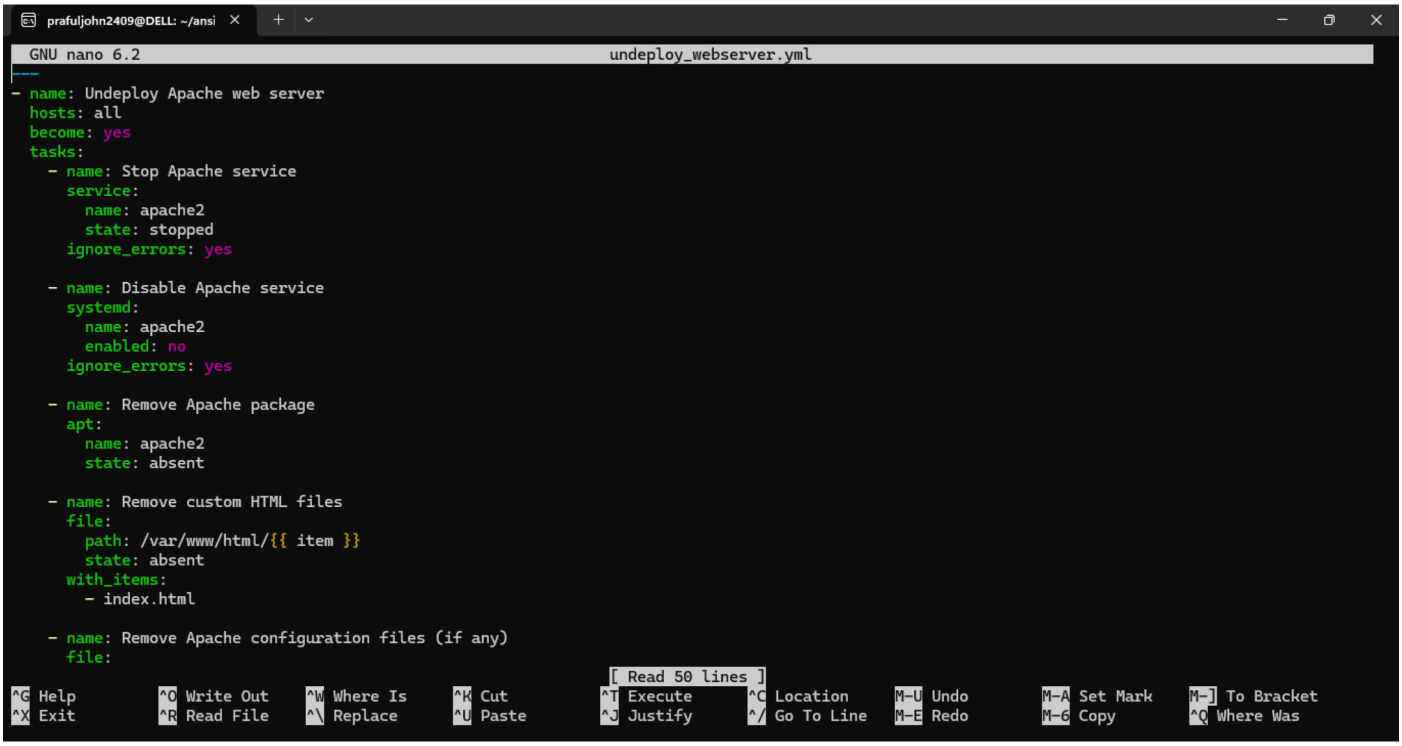


```
prafuljohn2409@DELL:~/ansible_project$ nano deploy_webserver3.yml
prafuljohn2409@DELL:~/ansible_project$ ansible-playbook -i inventory deploy_webserver3.yml

PLAY [Deploy Apache web server] ********************************************************************************************

TASK [Gathering Facts] *****************************************************************************************************
ok: [vm2]
ok: [vm1]

TASK [Install apache2] *****************************************************************************************************
ok: [vm2]
ok: [vm1]

TASK [Create a custom HTML page] *******************************************************************************************
changed: [vm2]
changed: [vm1]

TASK [Ensure apache2 is started] *******************************************************************************************
ok: [vm2]
ok: [vm1]

TASK [Check apache2 status] ************************************************************************************************
ok: [vm2]
ok: [vm1]

TASK [Debug apache2 status] ************************************************************************************************
ok: [vm1] => {
    "msg": "● apache2.service - The Apache HTTP Server\n     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)\n
  Active: active (running) since Sun 2024-09-08 10:12:21 UTC; 28min ago\n       Docs: https://httpd.apache.org/docs/2.4/\n    Process: 5992 ExecStar
t=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)\n   Main PID: 5995 (apache2)\n      Tasks: 55 (limit: 1130)\n     Memory: 5.6M (peak: 5.
9M)\n        CPU: 115ms\n     CGroup: /system.slice/apache2.service\n             ├─5995 /usr/sbin/apache2 -k start\n             ├─5997 /usr/sbin/a
pache2 -k start\n             └─5998 /usr/sbin/apache2 -k start\n\nSep 08 10:12:21 ip-172-31-23-84 systemd[1]: Starting apache2.service - The Apache
 HTTP Server...\nSep 08 10:12:21 ip-172-31-23-84 systemd[1]: Started apache2.service - The Apache HTTP Server."
}
```

Fig 6:- Deploying the HTML page

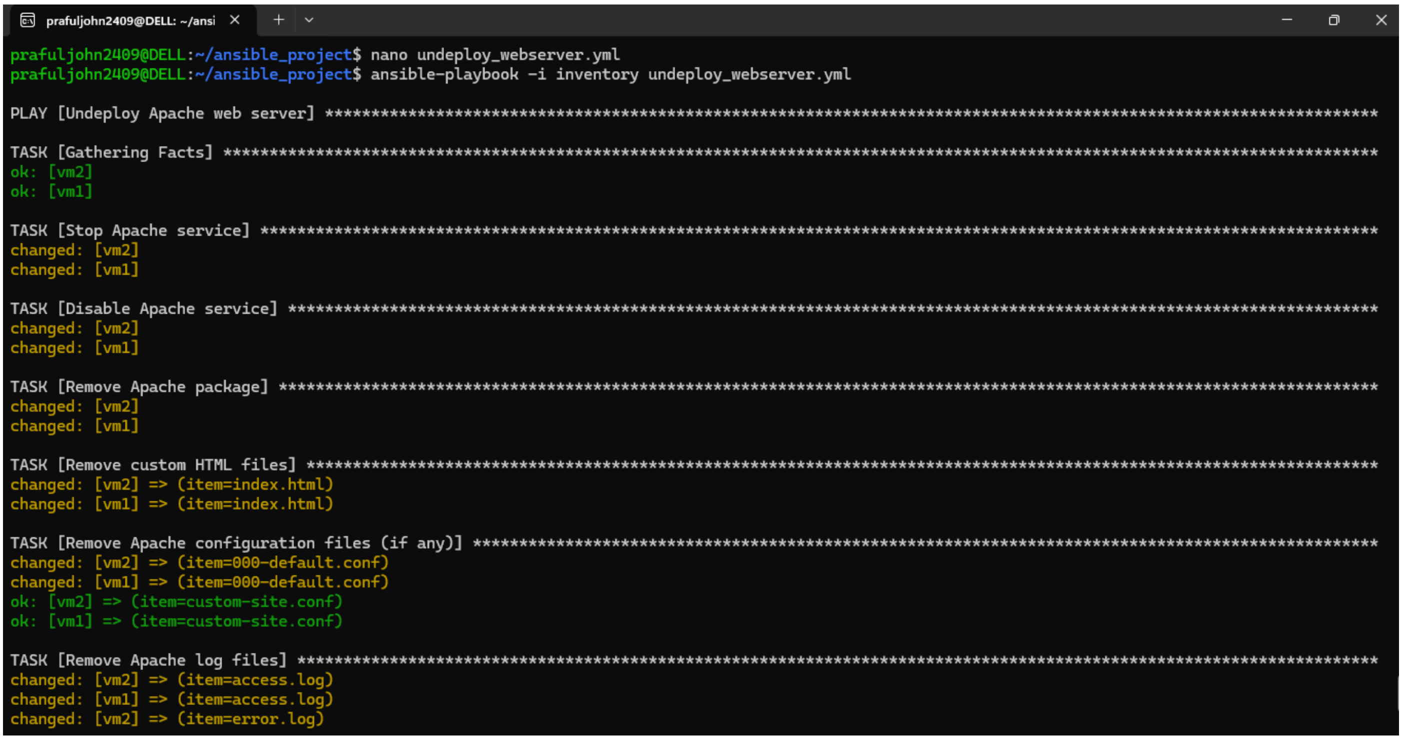
To undeploy:

I created an undeploy playbook ‘undeploy_webservers.yml’ to undeploy the HTML page.



```
  GNU nano 6.2                                            undeploy_webserver.yml
---
- name: Undeploy Apache web server
  hosts: all
  become: yes
  tasks:
    - name: Stop Apache service
      service:
        name: apache2
        state: stopped
      ignore_errors: yes

    - name: Disable Apache service
      systemd:
        name: apache2
        enabled: no
      ignore_errors: yes

    - name: Remove Apache package
      apt:
        name: apache2
        state: absent

    - name: Remove custom HTML files
      file:
        path: /var/www/html/{{ item }}
        state: absent
      with_items:
        - index.html

    - name: Remove Apache configuration files (if any)
      file:
                                                    [ Read 50 lines ]
```

Fig7: undeploy_webservers.yml’

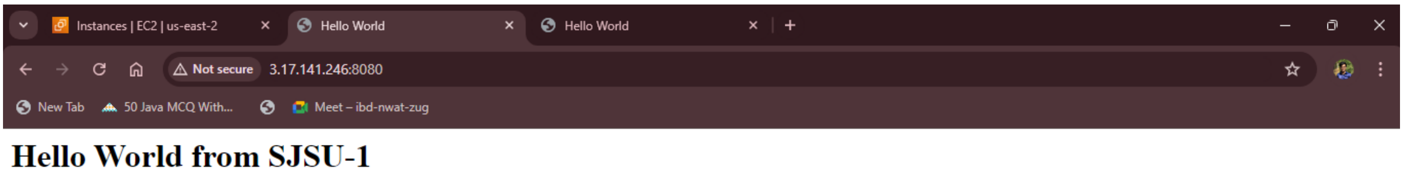


```
prafuljohn2409@DELL:~/ansible_project$ nano undeploy_webserver.yml
prafuljohn2409@DELL:~/ansible_project$ ansible-playbook -i inventory undeploy_webserver.yml

PLAY [Undeploy Apache web server] ***********************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************
ok: [vm2]
ok: [vm1]

TASK [Stop Apache service] ******************************************************************************************************
changed: [vm2]
changed: [vm1]

TASK [Disable Apache service] ***************************************************************************************************
changed: [vm2]
changed: [vm1]

TASK [Remove Apache package] ****************************************************************************************************
changed: [vm2]
changed: [vm1]

TASK [Remove custom HTML files] *************************************************************************************************
changed: [vm2] => (item=index.html)
changed: [vm1] => (item=index.html)

TASK [Remove Apache configuration files (if any)] *******************************************************************************
changed: [vm2] => (item=000-default.conf)
changed: [vm1] => (item=000-default.conf)
ok: [vm2] => (item=custom-site.conf)
ok: [vm1] => (item=custom-site.conf)

TASK [Remove Apache log files] **************************************************************************************************
changed: [vm2] => (item=access.log)
changed: [vm1] => (item=access.log)
changed: [vm2] => (item=error.log)
```

Fig8: successful undeployment of the HTML pages

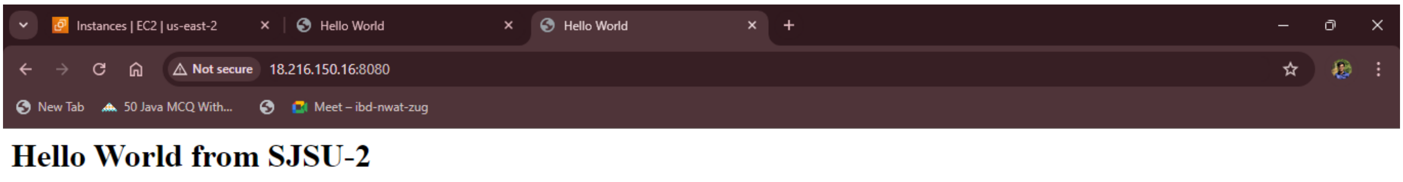
Results:

- The HTML pages were deployed on the VMs with the the message: “Hello World from SJSU-X” where X is 1 or 2 depending on which webserver instance, VM1 or VM2.



Hello World from SJSU-1

Fig9: HTML on VM1

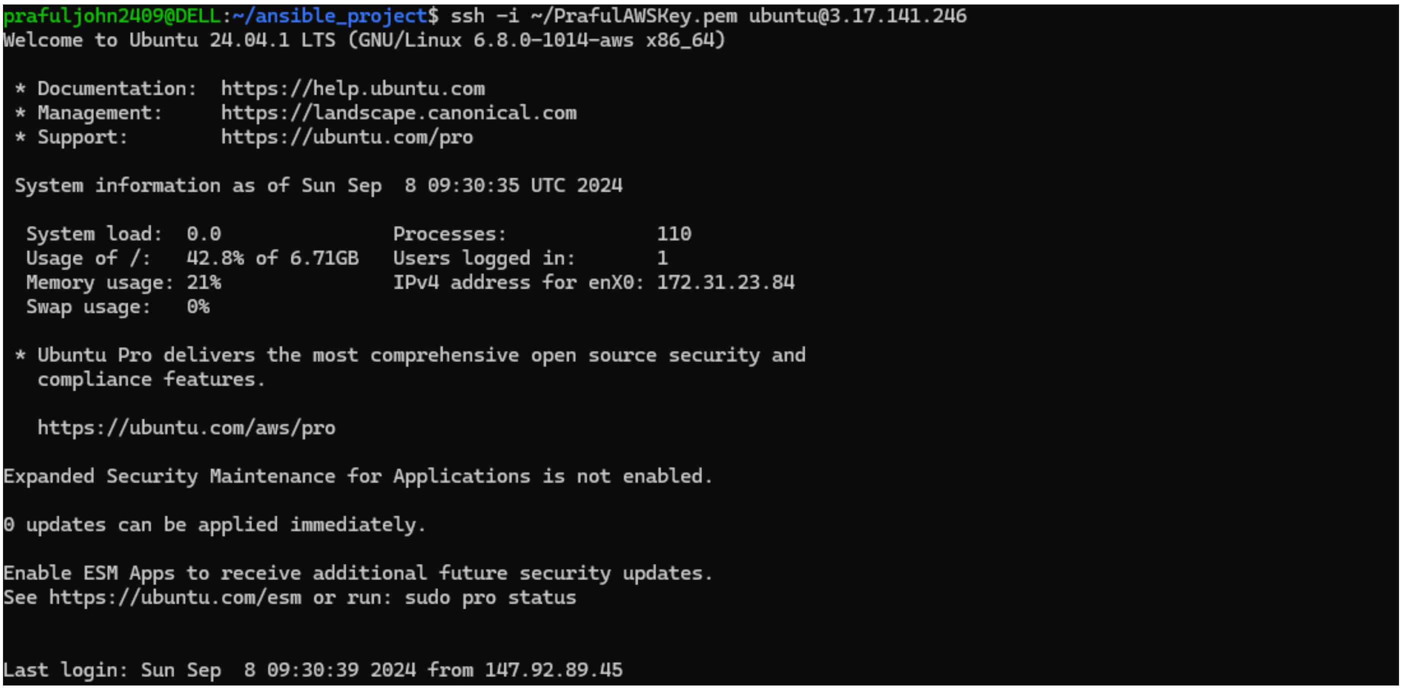


Hello World from SJSU-2

Fig10: HTML on VM2

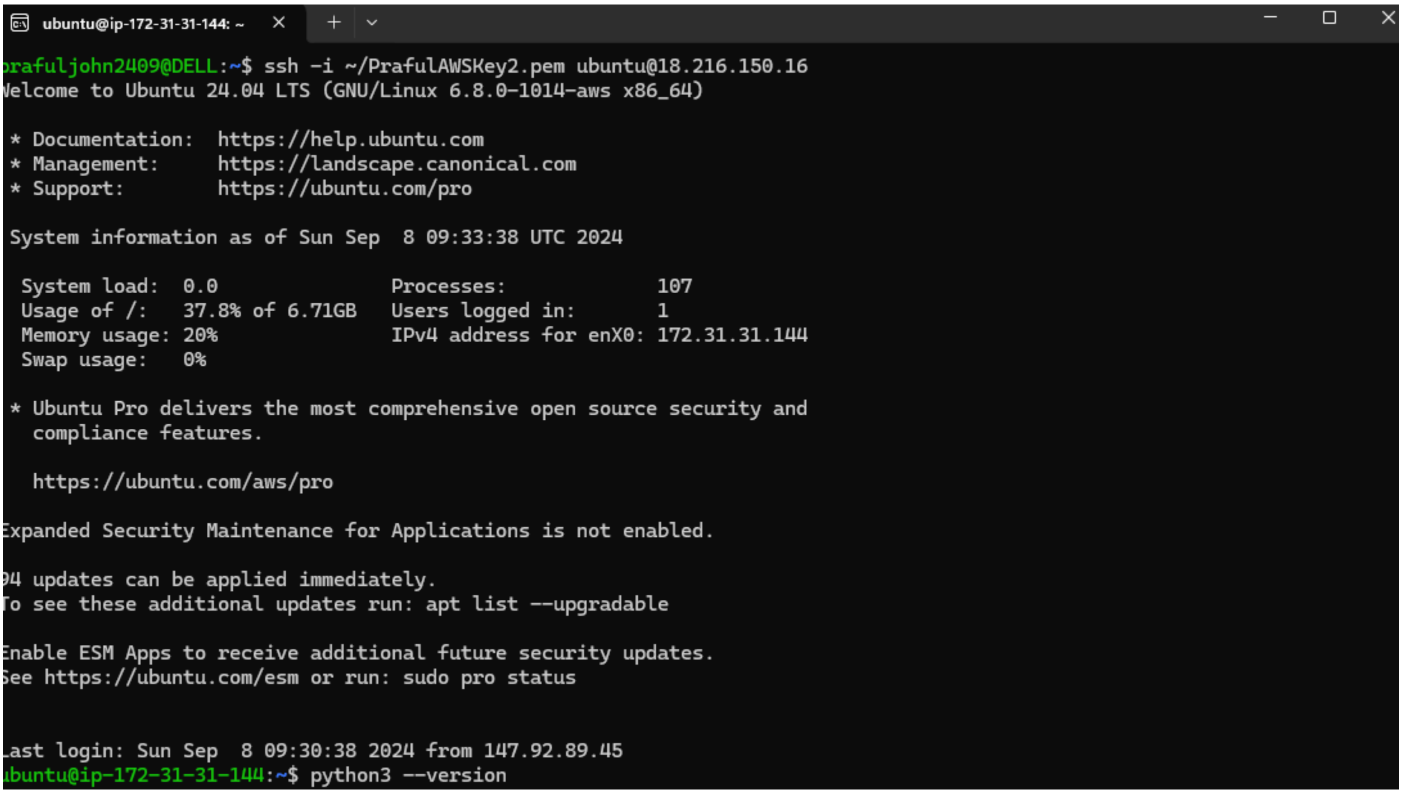
Testing:

There were times when the connection was not establishing then SSHing the server(s).



```
prafuljohn2409@DELL:~/ansible_project$ ssh -i ~/PrafulAWSKey.pem ubuntu@3.17.141.246
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1014-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sun Sep  8 09:30:35 UTC 2024

  System load:  0.0               Processes:             110
  Usage of /:   42.8% of 6.71GB   Users logged in:       1
  Memory usage: 21%               IPv4 address for enX0: 172.31.23.84
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Sun Sep  8 09:30:39 2024 from 147.92.89.45
```

Fig11: SSHing the VM1



```
prafuljohn2409@DELL:~$ ssh -i ~/PrafulAWSKey2.pem ubuntu@18.216.150.16
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1014-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sun Sep  8 09:33:38 UTC 2024

  System load:  0.0               Processes:             107
  Usage of /:   37.8% of 6.71GB   Users logged in:       1
  Memory usage: 20%               IPv4 address for enX0: 172.31.31.144
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

94 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Sun Sep  8 09:30:38 2024 from 147.92.89.45
ubuntu@ip-172-31-31-144:~$ python3 --version
```

Fig12: SSHing VM2