

## Assignment – 1 Ansible

**Ask:** Configure two VMs, VM1 and VM2 either on your own hardware, or in a cloud environment. Configure ansible to deploy a web server 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 web server instance, VM1 or VM2. Include in the ansible playbook, plays to deploy and un deploy the web server resources.

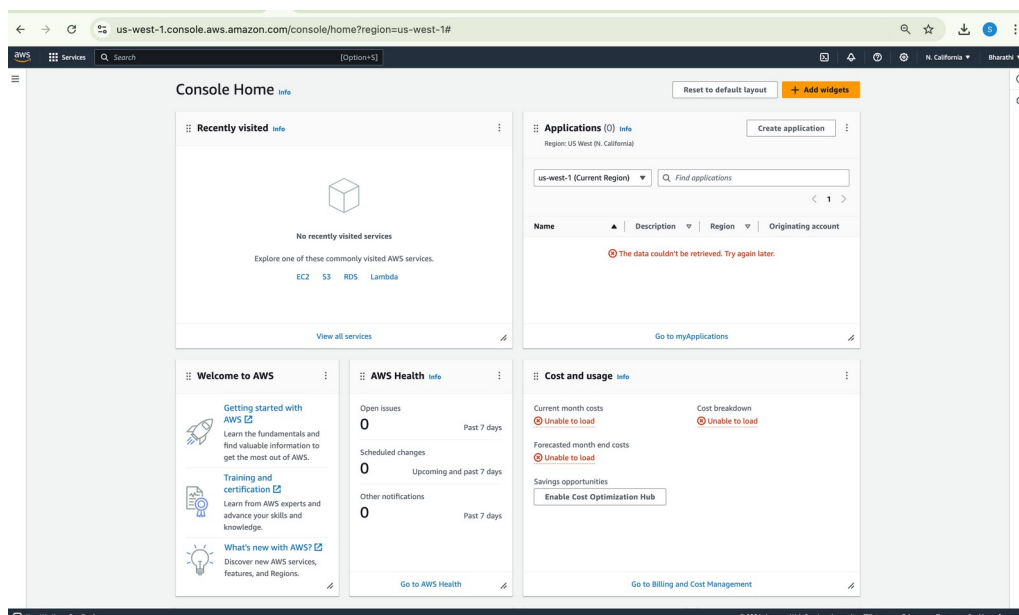
**\*\*GITHUB URL:** The following URL has the ansible session logs and Playbooks used for the assignment.

[https://github.com/BharathiVetukuri/CMPE-272\\_EnterpriseSoftwarePlatforms](https://github.com/BharathiVetukuri/CMPE-272_EnterpriseSoftwarePlatforms)

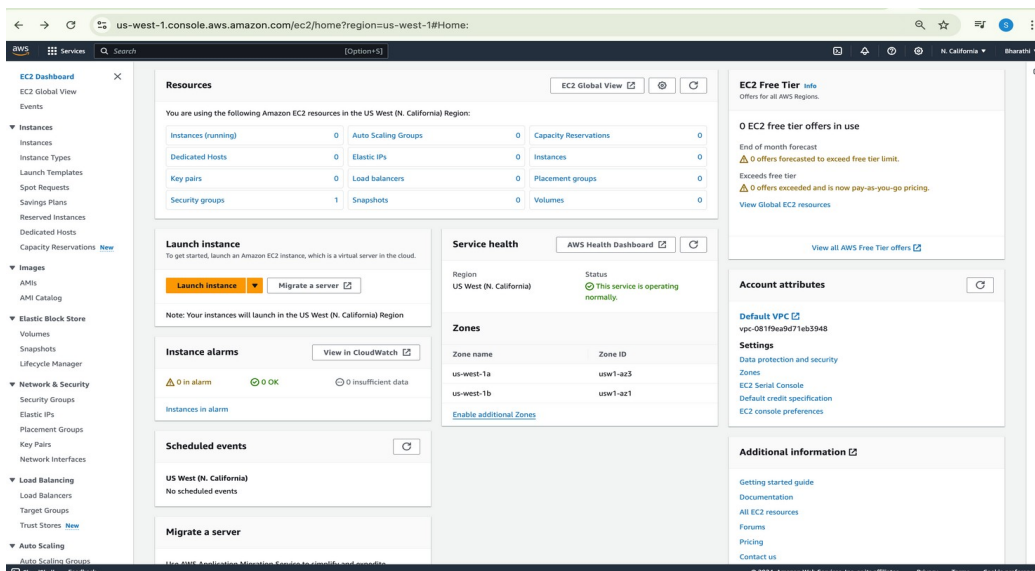
### 1. Configuration of VMs in Cloud Environment:

I have chosen Amazon Web Services(AWS) as the cloud environment to create and configure my two virtual machines. The following steps below with screenshots describe the process of VMs creation in AWS.

- **Login to AWS:** Below is the AWS Dashboard or the Console Home page which appears after creating an account and logging in to the cloud.



- **Creation of Instances:** From the Console, select EC2 and then click on the Launch Instance button in the EC2 Dashboard. Instances are nothing but the VMs.



- **Configuration:** In the Launch an Instance page, I have given the below details to configure my VM.

Name and Tags: Ubuntu VM1

Application and OS Images (AMI): Ubuntu

Instance Type: t2.micro

Key Pair: click on create new pair. In the create key pair page, give the name, type(RSA) and format(PEM) and click on create key pair button. A '.pem' file with key will be downloaded to machine.

Network Settings: Either a default security group can be selected by choosing 'select existing security group' option or 'create security group' can be chosen to define new rules.

Configuration Storage: Keep default and click Create Instance.

**Launch an instance** [info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags** [info](#)

Name  
Ubuntu VM1 [Add additional tags](#)

**Application and OS Images (Amazon Machine Image)** [info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

**Quick Start**

Amazon Linux, Ubuntu, Windows, Red Hat, SUSE Linux, Debian

**Amazon Machine Image (AMI)**

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type [Free tier eligible](#)

ami-0d53d72369335a9d6 (64-bit (x86)) / ami-06098d756d39a2267 (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

**Description**

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

**Architecture** 64-bit (x86) **AMI ID** ami-0d53d72369335a9d6 [Verified provider](#)

**Instance type** [info](#) [Get advice](#)

**Instance type**

t2.micro [Free tier eligible](#)

Family: t2 1 vCPU 1 GiB Memory Current generation: true  
On-Demand RHEL base pricing: 0.0282 USD per Hour  
On-Demand SUSE base pricing: 0.0138 USD per Hour  
On-Demand Windows base pricing: 0.0184 USD per Hour  
On-Demand Linux base pricing: 0.0158 USD per Hour

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

**Quick Start**

Amazon Linux, Ubuntu, Windows, Red Hat, SUSE Linux, Debian

**Amazon Machine Image (AMI)**

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type [Free tier eligible](#)

ami-0d53d72369335a9d6 (64-bit (x86)) / ami-06098d756d39a2267 (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

**Description**

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

**Architecture** 64-bit (x86) **AMI ID** ami-0d53d72369335a9d6 [Verified provider](#)

**Instance type** [info](#) [Get advice](#)

**Instance type**

t2.micro [Free tier eligible](#)

Family: t2 1 vCPU 1 GiB Memory Current generation: true  
On-Demand RHEL base pricing: 0.0282 USD per Hour  
On-Demand SUSE base pricing: 0.0138 USD per Hour  
On-Demand Windows base pricing: 0.0184 USD per Hour  
On-Demand Linux base pricing: 0.0158 USD per Hour

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

**Create key pair** [X](#)

**Key pair name**

Key pairs allow you to connect to your instance securely.

KEY-2

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

**Key pair type**

☒ RSA  
RSA encrypted private and public key pair

☐ ED25519  
ED25519 encrypted private and public key pair

**Private key file format**

☒ .pem  
For use with OpenSSH

☐ .ppk  
For use with PuTTY

**When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)**

[Cancel](#) [Create key pair](#)

**Network settings** [info](#) [Edit](#)

**Network** [info](#)

vpc-081f9ea9d71eb3948

**Subnet** [info](#)

No preference (Default subnet in any availability zone)

**Auto-assign public IP** [info](#)

Enable

**Additional charges apply when outside of free tier allowance**

**Firewall (security groups)** [info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group ☒ Select existing security group

**Common security groups** [info](#)

Select security groups

default sg-00a640f56ab09f585 [X](#)

VPC: vpc-081f9ea9d71eb3948

Security groups that you add or remove here will be added to or removed from all your network interfaces.

[Compare security group rules](#)

**Configure storage** [info](#) [Advanced](#)

1x 8 GiB gp3 [Root volume \(Not encrypted\)](#)

[Free tier eligible customers can get up to 30 GiB of EBS General Purpose \(SSD\) or Magnetic storage](#) [X](#)

[Add new volume](#)

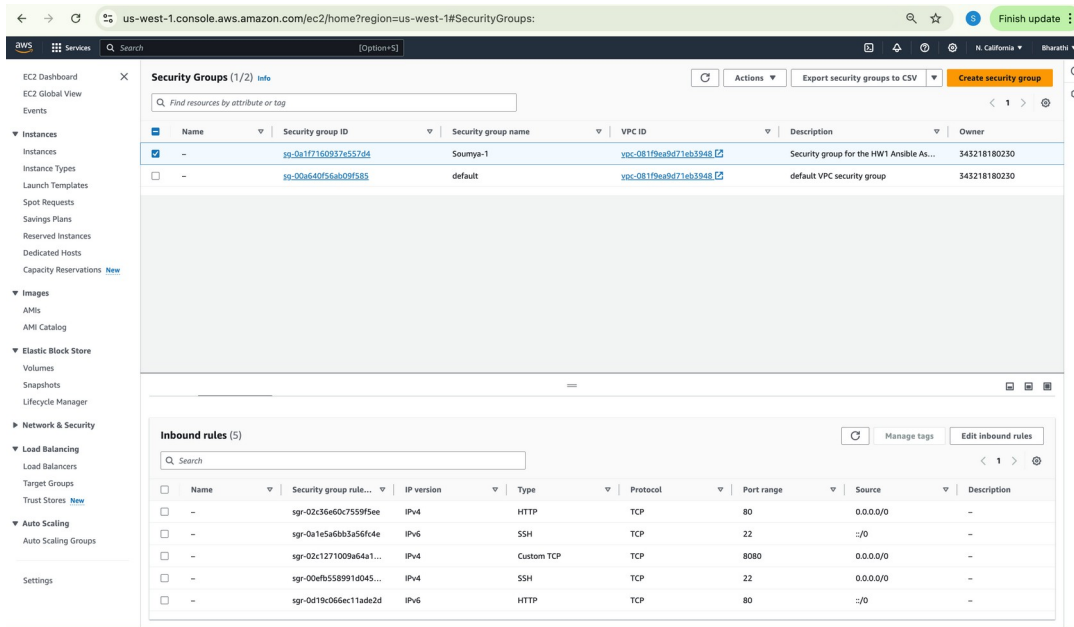
The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance.

[Click refresh to view backup information](#)

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

**File systems** [Edit](#)

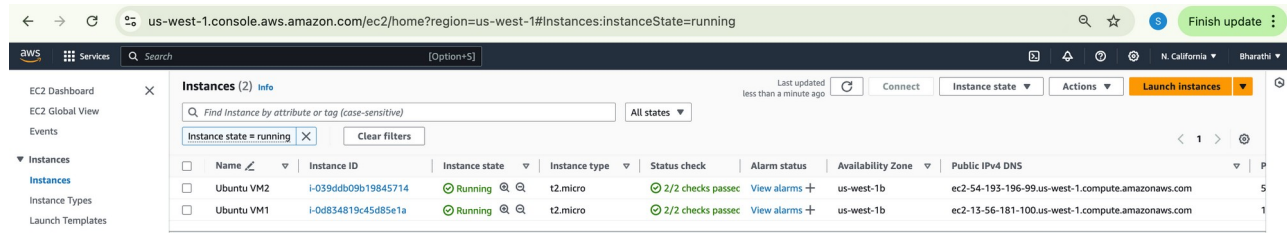
\*Create Security Group: I chose to create a new security group 'Soumya-1' and set the rules mentioned in the screenshot below to allow ports 80 and 8080 in the Inbound rules.



Once, all the configuration is verified and 'Launch Instance' button is clicked, the virtual machine will be created and the following success message will appear.



- **Repeat for VM2:** After the creation of VM1, repeated the same steps for creating the second virtual machine using the same key pair and security group. The EC2 Dashboard will have the information of number of Instances, key pairs and security groups created. The Instances tab will have the information about the VMs created.



## 2. Connect to VMs with SSH and Key Pair:

```
ssh -i KEY-1.pem ubuntu@13.56.181.100
ssh -i KEY-1.pem ubuntu@54.193.196.99
```

```

bharathi@Bharathi-MacBook-Air ~ % ssh -i /Users/bharathi/KEY-1.pem ubuntu@13.56.181.100
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

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

System information as of Wed Sep  4 05:40:28 UTC 2024

System load:  0.04               Processes:           106
Usage of /:   22.9% of 6.71GB    Users logged in:    0
Memory usage: 30%               IPv4 address for enX0: 172.31.24.21
Swap usage:   0%

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

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Wed Sep  4 05:40:29 2024 from 24.130.212.211
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-24-21:~$

```

### 3. Install Ansible:

- **Install HomeBrew:** I installed ansible in my system using Homebrew. The commands prompts for sudo access, password entry is required. So these are the commands used to install homebrew:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

to verify homebrew installation, I checked the version using:

```
brew --version
```

- **Install Ansible:** After the Brew installation is complete used the following command to install ansible:

```
brew install ansible
```

to verify ansible installation, I checked the version using:

```
ansible --version
```

**\*\*The terminal logs for these commands were huge. Please find in the logs file being submitted along with the doc.**

### 4. Configure Ansible:

- **Create Ansible directory:** mkdir creates the ansible directory and cd changes the directory to ansible.

```
mkdir ansible_asgn1
```

```
cd ansible_asgn1
```

- **Inventory File:** After navigating to the ansible directory, create an inventory file which contains IP addresses of the Virtual Machines.

```
nano hosts.ini
```

- **Hosts.ini:** the inventory file should have the following webservers information like IP, type(ubuntu) and key pair.

```
[webservers]
vm1 ansible_host=54.193.196.99 ansible_user=ubuntu ansible_ssh_private_key_file=/kEY-1.pem
vm2 ansible_host=13.56.181.100 ansible_user=ubuntu ansible_ssh_private_key_file=/kEY-1.pem
```

```
-----
bharathi@Bharathis-MacBook-Air ~ % mkdir ansible_asgn1
bharathi@Bharathis-MacBook-Air ~ % cd ansible_asgn1
bharathi@Bharathis-MacBook-Air ansible_asgn1 % nano hosts.ini
bharathi@Bharathis-MacBook-Air ansible_asgn1 % cat hosts.ini
[webservers]
vm1 ansible_host=13.56.181.100 ansible_user=ubuntu ansible_ssh_private_key_file=~/.ssh/your-aws-key.pem
vm2 ansible_host=54.193.196.99 ansible_user=ubuntu ansible_ssh_private_key_file=~/.ssh/your-aws-key.pem
bharathi@Bharathis-MacBook-Air ansible_asgn1 % pwd
/Users/bharathi/ansible_asgn1
```

## 5. Ansible Communication to VMs: Use the following command to ping to the VMs from ansible:

```
ansible -i hosts.ini webservers -m ping
```

If the connection is successful, servers will respond with “pong” (This response made me happy!!)

```
bharathi@Bharathis-MacBook-Air ansible_asgn1 % ansible -i hosts.ini webservers -m ping
[WARNING]: Platform linux on host vm1 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
vm1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "changed": false,
  "ping": "pong"
}
[WARNING]: Platform linux on host vm2 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
vm2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "changed": false,
  "ping": "pong"
}
```

## 6. Ansible Playbook to Deploy Webserver:

- **Create yaml file:** The ansible directory should be open and now give the following command to create a .yaml file which will be the playbook. The deploy\_undeploy file will open and write the playbook for deploying server.

```
nano deploy_undeploy_web.yaml
```

```
UW PICO 5.89 File: deploy_undeploy_web.yaml
--
- name: Deploy Webserver on VM1 and VM2
  hosts: webservers
  become: yes
  tasks:
    # Install Apache web server
    - name: Install Apache web server
      apt:
        name: apache2
        state: present
        update_cache: yes

    # Ensure Python is installed
    - name: Ensure Python is installed
      apt:
        name: python3
        state: present

    # Configure Apache to listen on port 8080
    - name: Change Apache to listen on port 8080
      lineinfile:
        path: /etc/apache2/ports.conf
        regexp: '^Listen 80'
        line: 'Listen 8080'

    - name: Update default virtual host to use port 8080
      lineinfile:
        path: /etc/apache2/sites-available/000-default.conf
        regexp: '^<VirtualHost *:80>'
        line: '<VirtualHost *:8080>'

    # Restart Apache to apply changes
    - name: Restart Apache service
      service:
        name: apache2
        state: restarted
        become: yes

    # Create index.html with custom message
    - name: Create index.html on VM1
      copy:
        dest: /var/www/html/index.html
        content: "Hello World from SJSU-1"
        when: "'vm1' in inventory_hostname"

    - name: Create index.html on VM2
      copy:
        dest: /var/www/html/index.html
        content: "Hello World from SJSU-2"
        when: "'vm2' in inventory_hostname"

    # Enable Apache service on boot
    - name: Ensure Apache is enabled on boot
      systemd:
        name: apache2
        enabled: yes
```

- **Execute .yaml file:** Now, to run the playbook created:

ansible -playbook -i hosts.ini deploy\_undeploy\_web.yaml

```
bharathi@Bharathi-MacBook-Air ansible_asgn1 % ansible-playbook -i hosts.ini deploy_undeploy_web.yaml

PLAY [Deploy Webserver on VM1 and VM2] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host vm1 is using the discovered Python
interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [vm1]
[WARNING]: Platform linux on host vm2 is using the discovered Python
interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [vm2]

TASK [Install Apache web server] *****
ok: [vm1]
ok: [vm2]

TASK [Ensure Python is installed] *****
ok: [vm1]
ok: [vm2]

TASK [Change Apache to listen on port 8080] *****
ok: [vm1]
ok: [vm2]

TASK [Update default virtual host to use port 8080] *****
ok: [vm1]
ok: [vm2]

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

TASK [Create index.html on VM1] *****
skipping: [vm2]
ok: [vm1]

TASK [Create index.html on VM2] *****
skipping: [vm1]
ok: [vm2]

TASK [Ensure Apache is enabled on boot] *****
ok: [vm1]
ok: [vm2]

PLAY RECAP *****
vm1      : ok=8    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
vm2      : ok=8    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

bharathi@Bharathi-MacBook-Air ansible_asgn1 %
```

The playbook will install Apache on the VMs and

configure to listen to port 8080. The HTML file, index.html will be created and the requested “Hello World from SJSU” will be displayed.

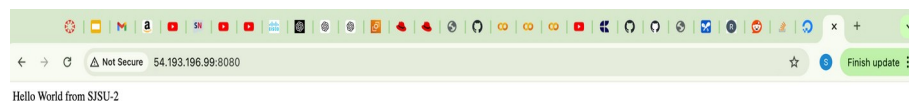
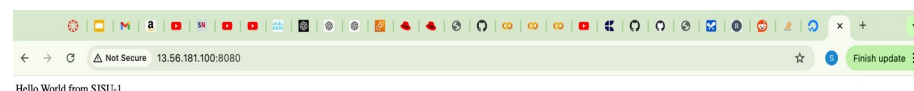
be created and the requested “Hello World from

- **HTML Page:** To check the messages displayed from both servers on port 8080, open a browser tab and give the following URL's for both the VMs:

<http://13.56.181.100:8080/>

<http://54.193.196.99:8080/>

Find the screenshots of the pages with Hello World messages for the respective VMs.



## 7. Ansible Playbook to UnDeploy Webserver:

- **Create yaml file:** The ansible directory should be open and now give the following command to create a .yaml file which will be the playbook to un deploy the servers. The undeploy.yaml file will open and write the playbook for un deploying the deployed servers.

nano undeploy.yaml

```
UN PICO 9.09 File: undeploy.yaml
--
- name: Un-deploy Webserver on VM1 and VM2
  hosts: webservers
  become: yes
  tasks:
    # Stop the Apache service
    - name: Stop Apache service
      service:
        name: apache2
        state: stopped

    # Disable Apache service on boot
    - name: Disable Apache service on boot
      systemd:
        name: apache2
        enabled: no

    # Uninstall Apache web server
    - name: Uninstall Apache web server
      apt:
        name: apache2
        state: absent

    # Remove the index.html file
    - name: Remove index.html file
      file:
        path: /var/www/html/index.html
        state: absent

Get Help
Exit
WriteOut
Justify
Read File
Where is
Prev Pg
Next Pg
Cut Text
Undo Cut Text
Cur Pos
To Spell
```

- **Execute undeploy.yaml file:** Now, to run the playbook created:

ansible -playbook -i hosts.ini undeploy.yaml

```
bharathi@Bharathi-MacBook-Air ansible_asgn1 % nano undeploy.yaml
bharathi@Bharathi-MacBook-Air ansible_asgn1 % ansible-playbook -i hosts.ini undeploy.yaml

PLAY [Un-deploy Webserver on VM1 and VM2] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host vm1 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [vm1]
[WARNING]: Platform linux on host vm2 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [vm2]

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

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

TASK [Uninstall Apache web server] *****
changed: [vm1]
changed: [vm2]

TASK [Remove index.html file] *****
changed: [vm1]
changed: [vm2]

PLAY RECAP *****
vm1 : ok=5 changed=4 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
vm2 : ok=5 changed=4 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

bharathi@Bharathi-MacBook-Air ansible_asgn1 %
```

**GITHUB URL:** The following URL has the ansible session logs and Playbooks used for the assignment.

[https://github.com/BharathiVetukuri/CMPE-272\\_EnterpriseSoftwarePlatforms](https://github.com/BharathiVetukuri/CMPE-272_EnterpriseSoftwarePlatforms)

**\*\*References:** Referred to AWS Tutorials from Youtube and ChatGPT for the Ansible installation and bash commands.