HW #1 – Ansible

Aarsh Sheth || Pooja Sindham || Shivani Jariwala || Aishwarya Indi

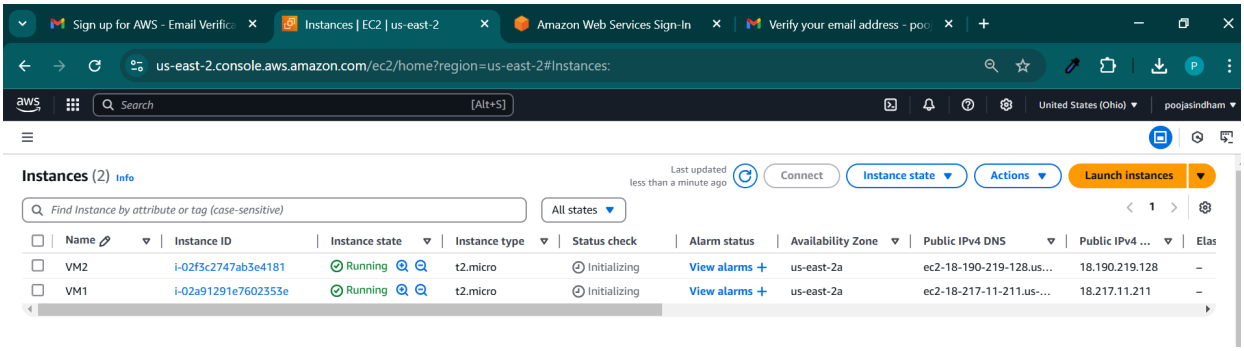
Video Demo Link: https://drive.google.com/file/d/1fgGfNt-3Ul56iy984sOEWaSoRqel8FTS/view?usp=sharing

GitHub Repo: https://github.com/wh0th3h3llam1/cmpe-272/

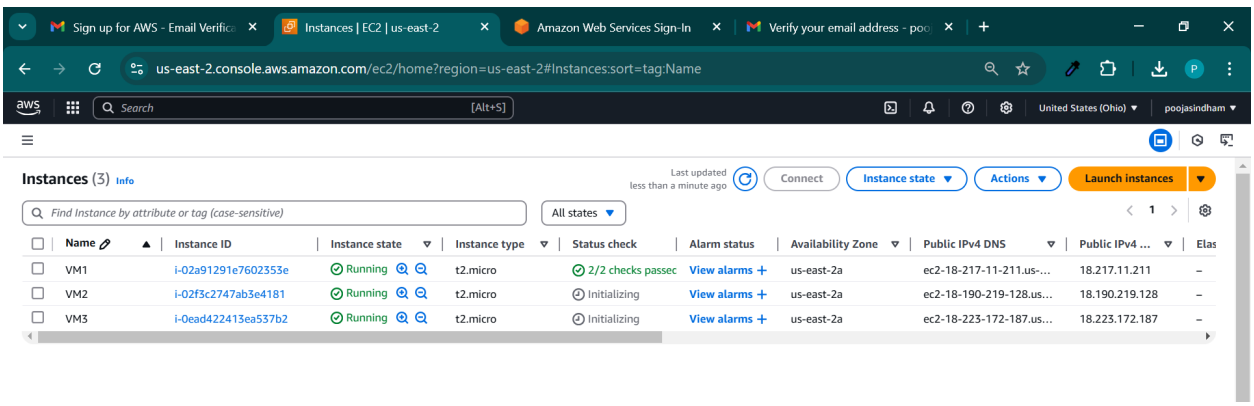
Ansible-Based Webserver Deployment on AWS VMs

This project will configure and deploy Apache web server on multiple AWS EC2 instances using Ansible. The setup includes a control node (VM3) managing two target nodes (VM1 and VM2). The Ansible playbook performs tasks such as installing Apache, modifying its configuration to run on port 8080, and deploying custom web pages to each VM. Security groups are configured to allow inbound traffic on port 8080, ensuring accessibility. The project demonstrates infrastructure automation, remote configuration management, and cloud-based deployment using Ansible.

1. Created Two Virtual Machines (VM1 and VM2)



2. Created VM3 as Control Node.



3. Connect to control Node from local through SSH.

```
ssh -i sjsu-ansible-key.pem ubuntu@18.223.172.187
sudo apt update
sudo apt install -y ansible
```

4. Copy key file to control node.

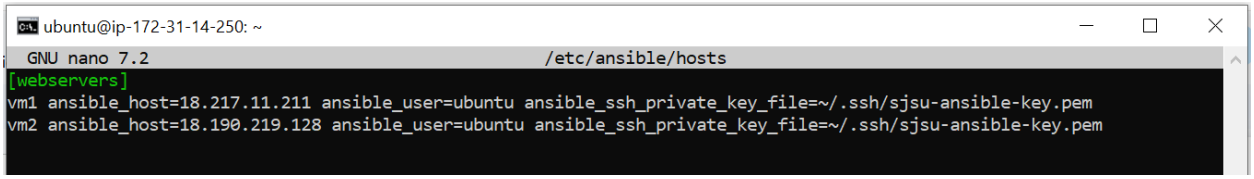
```
scp -i sjsu-ansible-key.pem sjsu-ansible-key.pem
ubuntu@18.223.172.187:~
ssh -i sjsu-ansible-key.pem ubuntu@18.223.172.187
```

5. Testing ssh connectivity to VM1 and VM2 from VM3 (Control Node)

```
ssh -i ~/.ssh/sjsu-ansible-key.pem ubuntu@18.190.219.128
ssh -i ~/.ssh/sjsu-ansible-key.pem ubuntu@18.217.11.211
```

6. Create Inventory File

```
sudo mkdir -p /etc/ansible
sudo touch /etc/ansible/hosts
sudo nano /etc/ansible/hosts
```



7. Test Ansible Ping

```
ansible all -m ping
```

```
Last login: Fri Feb  7 07:27:28 2025 from 104.50.4.18
ubuntu@ip-172-31-14-250:~$ sudo nano /etc/ansible/hosts
ubuntu@ip-172-31-14-250:~$ ansible all -m ping
vm1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
vm2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-14-250:~$
```

8. Create the Ansible Playbook

```
nano deploy_webserver.yml
```

```
---
- name: Deploy Webserver on VM1 and VM2
  hosts: webservers
  become: true
  tasks:
    - name: Install Apache Web Server
      apt:
        name: apache2
        state: present
        update_cache: yes

    - name: Change Apache port to 8080
      lineinfile:
        path: /etc/apache2/ports.conf
        regexp: "^Listen 80"
        line: "Listen 8080"

    - name: Change Virtual Host to 8080
      replace:
        path: /etc/apache2/sites-available/000-default.conf
        regexp: "VirtualHost \\*:80"
        replace: "VirtualHost *:8080"

    - name: Restart Apache service
      service:
        name: apache2
        state: restarted

    - name: Deploy custom index.html for VM1
      copy:
        content: "Hello World from SJSU-1"
        dest: /var/www/html/index.html
      when: "'vm1' in inventory_hostname"

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

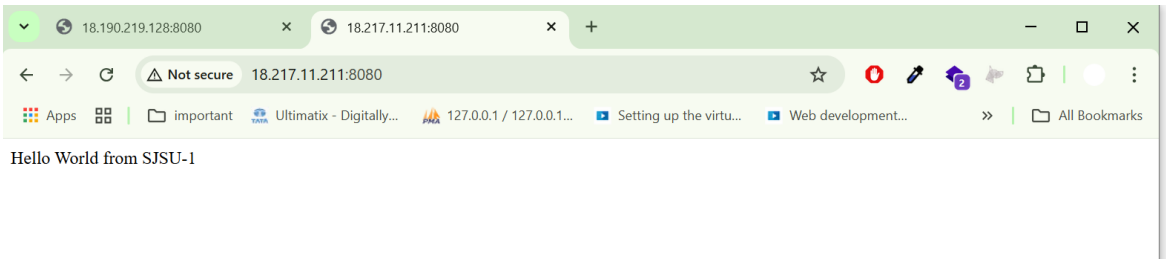
9. Run the Ansible Playbook

```
ansible-playbook deploy_webserver.yml
```

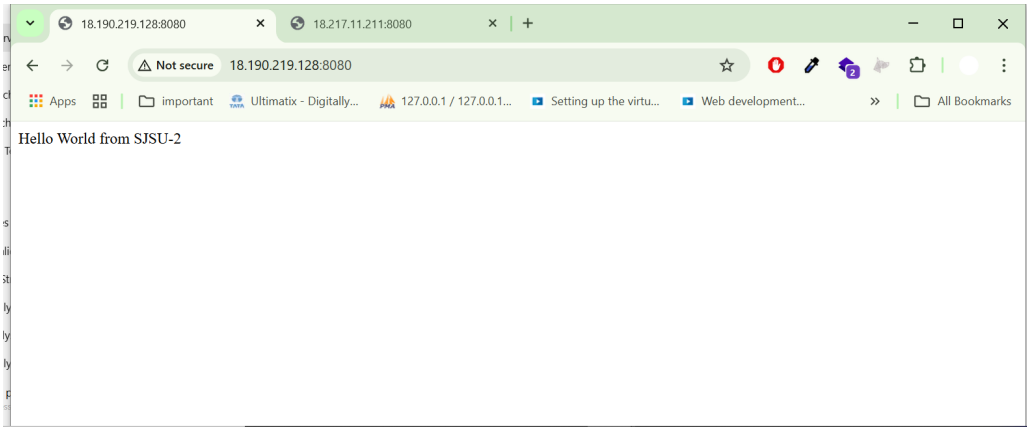
```
}
ubuntu@ip-172-31-14-250:~$ nano deploy_webserver.yml
ubuntu@ip-172-31-14-250:~$ ubuntu@ip-172-31-14-250:~$ ansible-playbook deploy_webserver.yml

PLAY [Deploy Webserver on VM1 and VM2] ****************************************************************************
TASK [Gathering Facts] ********************************************************************************************ok: [vm1]
ok: [vm2]

TASK [Install Apache Web Server] **********************************************************************************changed: [vm1]
changed: [vm2]

TASK [Change Apache port to 8080] *********************************************************************************changed: [vm1]
changed: [vm2]

TASK [Change Virtual Host to 8080] ********************************************************************************changed: [vm2]
changed: [vm1]

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

TASK [Deploy custom index.html for VM1] ***************************************************************************skipping: [vm2]
changed: [vm1]

TASK [Deploy custom index.html for VM2] ***************************************************************************skipping: [vm1]
changed: [vm2]

PLAY RECAP ********************************************************************************************************vm1                        : ok=6    changed=5    un
reachable=0    failed=0    skipped=1    rescued=0    ignored=0
vm2                        : ok=6    changed=5    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

ubuntu@ip-172-31-14-250:~$
```

10.Verify in the Browser

http://18.217.11.211:8080/



http://18.190.219.128:8080/



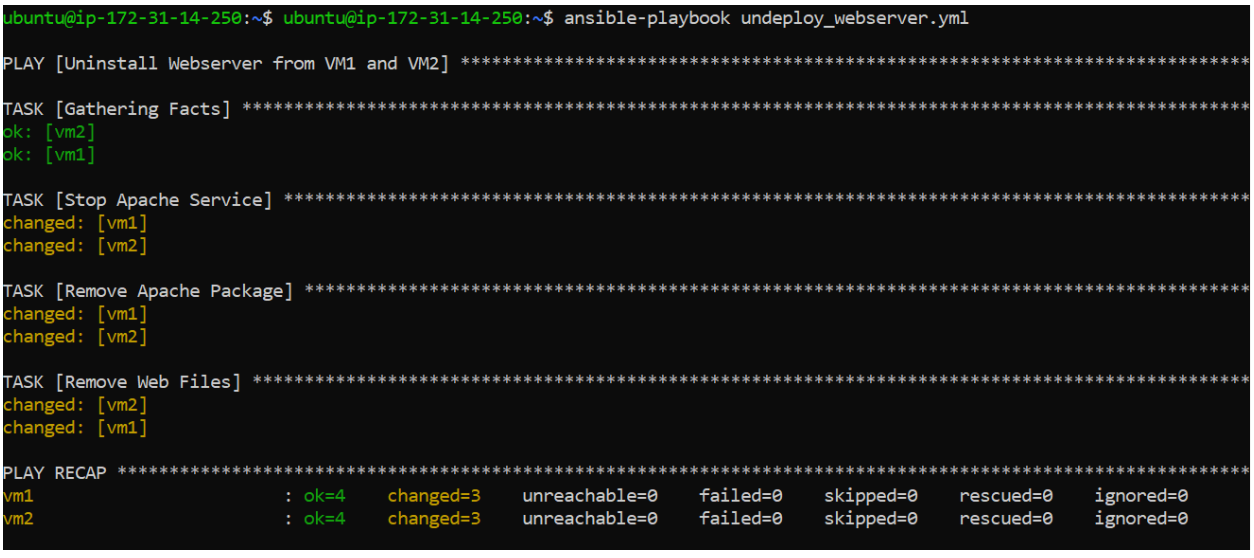
11. Undeploy Webserver Ansible Playbook

```
nano undeploy_webserver.yml
```

```
---
- name: Uninstall Webserver from VM1 and VM2
  hosts: webservers
  become: true
  tasks:
    - name: Stop Apache Service
      service:
        name: apache2
        state: stopped

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

    - name: Remove Web Files
      file:
        path: /var/www/html/index.html
        state: absent
```

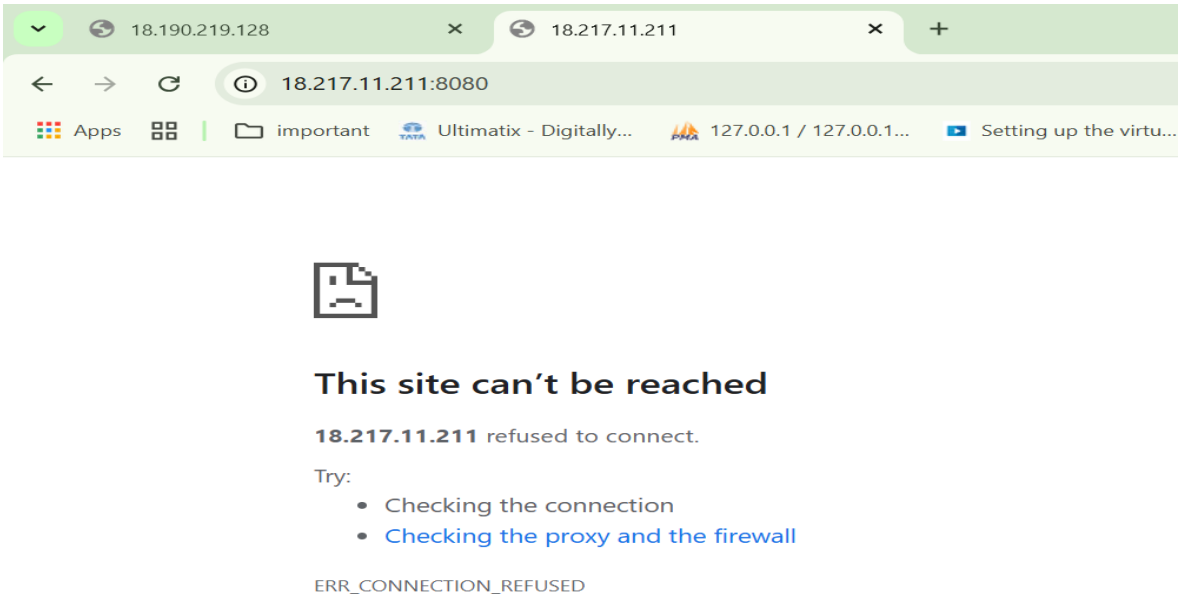
12. Run the playbook.

```
ansible-playbook undeploy_webserver.yml
```

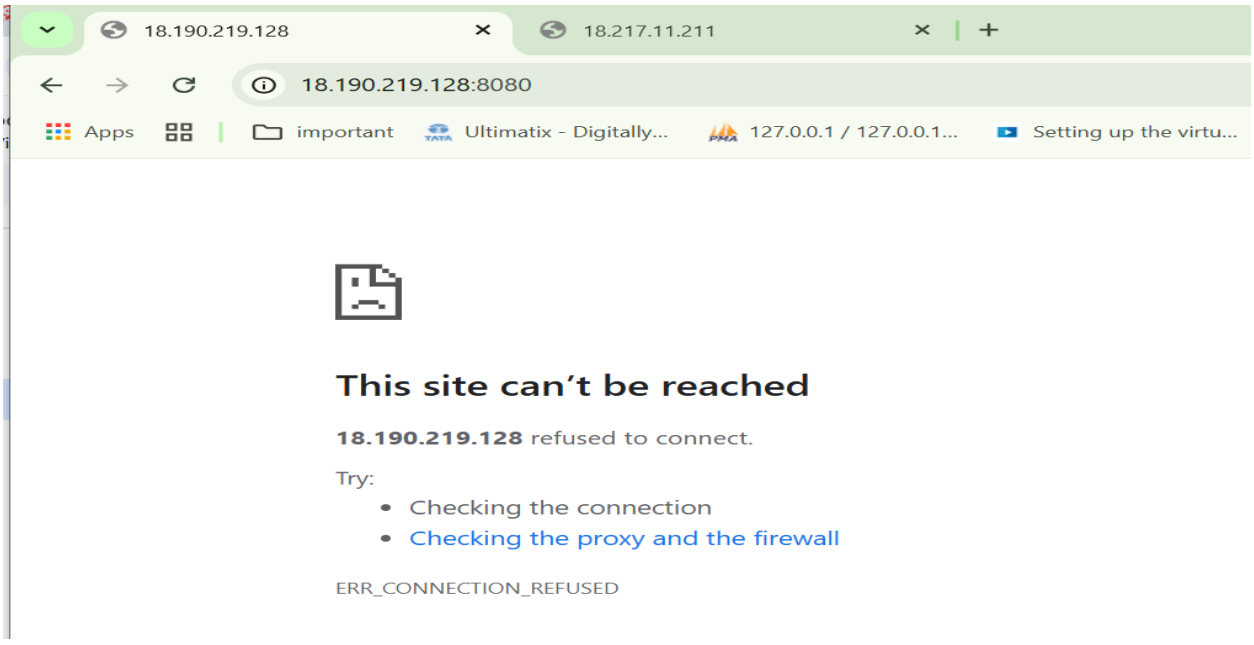


13. Verify in the Browser

http://18.217.11.211:8080/



http://18.190.219.128:8080/



Issues Faced -

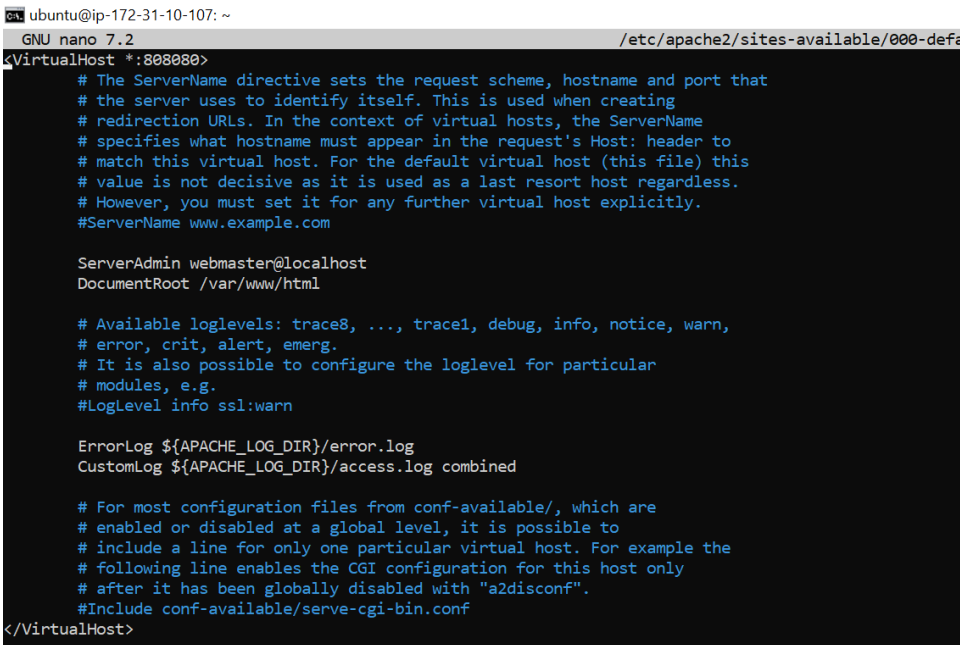
Issue 1 – When trying to rerun deploy_webserver.yml it is appending 8080 to the config file of apache

Below are steps to rectify it.

1. **Open the Apache port configuration file:**
   `sudo nano /etc/apache2/ports.conf`
   Make sure it contains:
   `Listen 8080`
   If there's a Listen 80 line, replace it with Listen 8080.

2. **Open the default virtual host configuration:**
   `sudo nano /etc/apache2/sites-available/000-default.conf`
   Change:
   `<VirtualHost *:80>`
   To:
   `<VirtualHost *:8080>`

   Save and exit (Ctrl + X → Y → Enter).



Issue with Apache Config File

3. **Restart Apache**

After making the above changes, restart Apache manually:

```
sudo systemctl restart apache2
```

If the restart fails again, check which process is using port `8080`:

```
sudo netstat -tulnp | grep 8080
```

If another process is running on port `8080`, stop it and restart Apache.

Issue 2 - `apache not found`

Check if Apache is Installed

Run this command to verify:

```
apache2 -v
```

If Apache is **not found**, install it using:

```
sudo apt update && sudo apt install -y apache2
```

After installation, start Apache:]

```
sudo systemctl start apache2
sudo systemctl enable apache2
```

Now check its status:

```
sudo systemctl status apache2
```

Issue 3 – `sudo apachectl configtest AH00526: Syntax error on line 1 of /etc/apache2/sites-enabled/000-default.conf: The address or port is invalid`

there's an issue with the Virtual Host configuration in /etc/apache2/sites-enabled/000-default.conf. Let's fix it step by step.

1. Open the Virtual Host Configuration File

Run:

```
sudo nano /etc/apache2/sites-enabled/000-default.conf
```

Make sure it has the **correct syntax** like this:

```
<VirtualHost *:8080>
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

- Ensure **<VirtualHost *:8080>** is correctly written.
- Remove any extra characters, invalid IP addresses, or typos.

Save the file (CTRL + X, then Y, then ENTER).

2. Check Apache Ports Configuration

```
Run: sudo nano /etc/apache2/ports.conf
```

Make sure it contains:

```
Listen 8080
```

Save the file and exit.

3. Disable and Re-enable the Site

Run these commands to apply the changes:

```
sudo a2dissite 000-default
sudo a2ensite 000-default
```

4. Restart Apache

Now, restart Apache:

```
sudo systemctl restart apache2
```

Check the status:

```
sudo systemctl status apache2
```