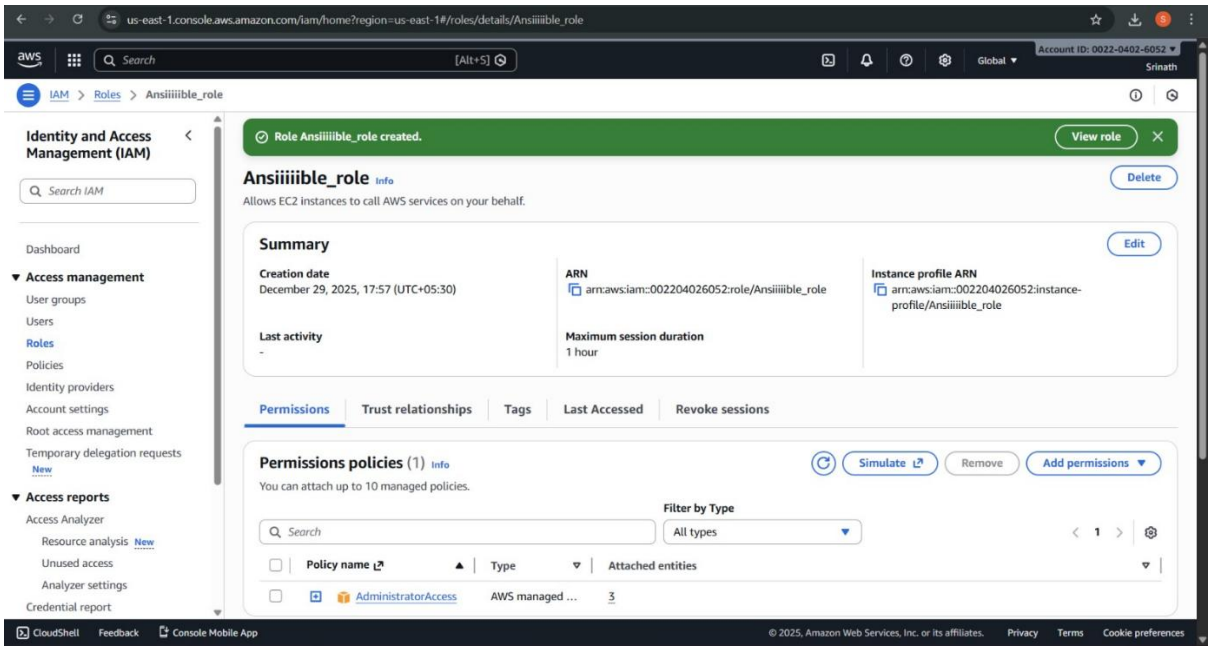
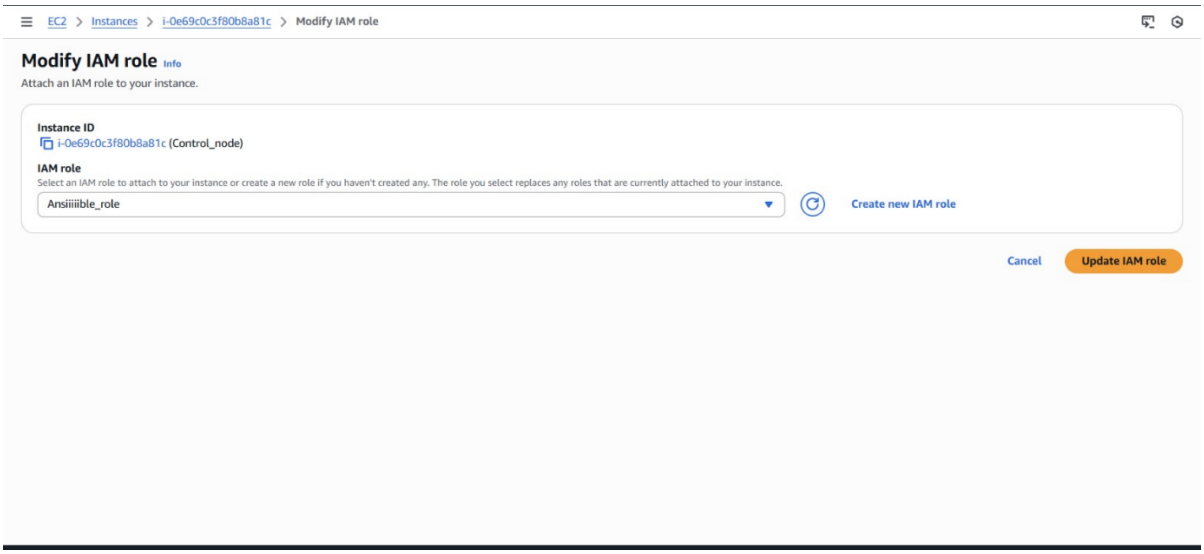
TASK – 26

ANSIBLE WITH LOOP

EC2 deployment through ansible:

Creating a IAM role and attach it to control node through actions in console





Installing boto3 and botocore

```
[ec2-user@ip-172-31-3-249 ~]$ pip3 install --user boto3 botocore
Collecting boto3
  Downloading boto3-1.42.17-py3-none-any.whl (140 kB)
     |████████████████████████████████| 140 kB 16.3 MB/s
Collecting botocore
  Downloading botocore-1.42.17-py3-none-any.whl (14.6 MB)
     |████████████████████████████████| 14.6 MB 425 kB/s
Collecting s3transfer<0.17.0,>=0.16.0
  Downloading s3transfer-0.16.0-py3-none-any.whl (86 kB)
     |████████████████████████████████| 86 kB 10.3 MB/s
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in /usr/lib/python3.9/site-packages (fr
om boto3) (0.10.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/lib/python3.9/site-package
s (from botocore) (2.8.1)
Requirement already satisfied: urllib3<1.27,>=1.25.4 in /usr/lib/python3.9/site-packages (fro
m botocore) (1.25.10)
Requirement already satisfied: six>=1.5 in /usr/lib/python3.9/site-packages (from python-date
util<3.0.0,>=2.1->botocore) (1.15.0)
Installing collected packages: botocore, s3transfer, boto3
Successfully installed boto3-1.42.17 botocore-1.42.17 s3transfer-0.16.0
```

Creating a yaml file for deploying the instance through ansible and run the playbook

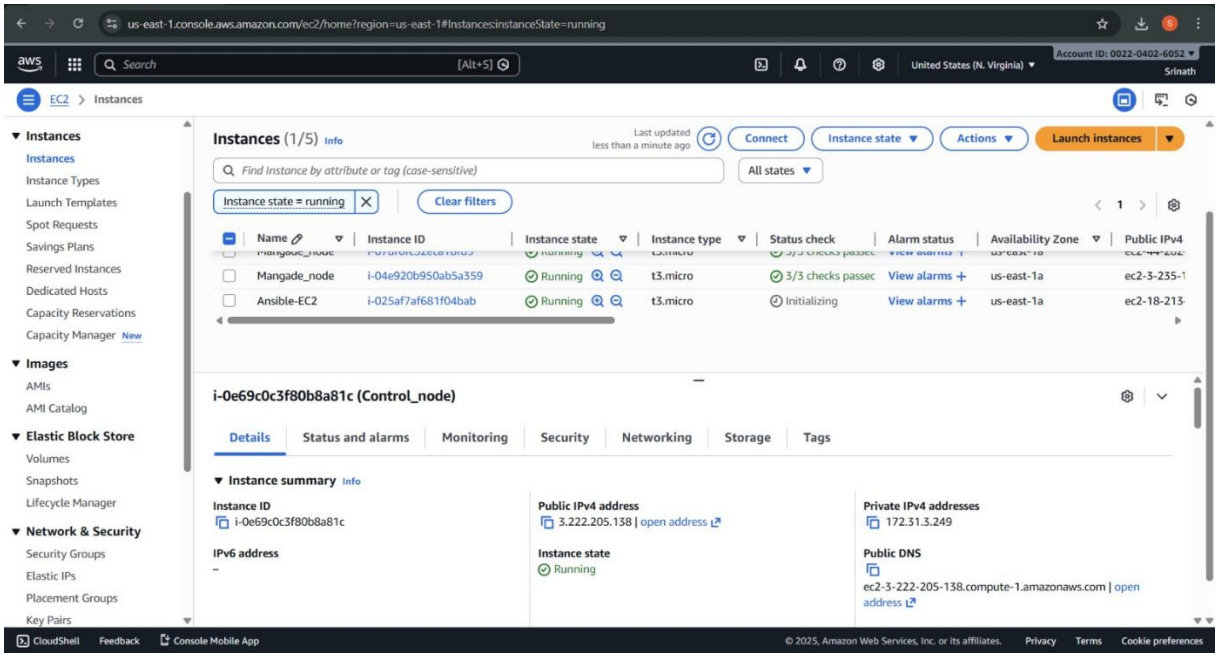
```
[ec2-user@ip-172-31-3-249 ~]$ vi instance.yaml
[ec2-user@ip-172-31-3-249 ~]$ ansible-playbook -i inventory.txt instance.yaml

PLAY [Create EC2 Instance using Ansible] ***************************************

TASK [Create EC2 instance] *****************************************************
changed: [localhost]

PLAY RECAP *********************************************************************
localhost                  : ok=1    changed=1    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
```

Code inside instance.yaml:

```
- name: Create EC2 Instance using Ansible
  hosts: localhost
  connection: local
  gather_facts: no

  vars:
    region: us-east-1
    instance_type: t3.micro
    key_name: ansiii
    image_id: ami-068c0051b15cdb816   # Amazon Linux 2 (Mumbai)
    security_group: my_sg
    instance_name: Ansible-EC2

  tasks:
    - name: Create EC2 instance
      amazon.aws.ec2_instance:
        name: "{{ instance_name }}"
        key_name: "{{ key_name }}"
        instance_type: "{{ instance_type }}"
        image_id: "{{ image_id }}"
        wait: yes
        region: "{{ region }}"
```

```
security_group: "{{ security_group }}"
count: 1
tags:
  Environment: Dev
  CreatedBy: Ansible
```

Instance created successfully



Tree:

"Ansible tree" refers to the standard directory structure for an Ansible project (especially roles) or the ansible.builtin.tree callback plugin that saves execution history as files, helping visualize playbooks or track events in a hierarchical format, crucial for understanding complex automation.

```
[ec2-user@ip-172-31-3-249 ~]$ sudo yum install tree
Last metadata expiration check: 1:03:52 ago on Mon Dec 29 11:49:43 2025.
Dependencies resolved.
================================================================================
 Package     Architecture  Version                    Repository          Size
================================================================================
Installing:
 tree        x86_64        1.8.0-6.amzn2023.0.2       amazonlinux         56 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 56 k
Installed size: 113 k
Is this ok [y/N]: y
Downloading Packages:
tree-1.8.0-6.amzn2023.0.2.x86_64.rpm           1.3 MB/s |  56 kB     00:00
--------------------------------------------------------------------------------
Total                                          686 kB/s |  56 kB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : tree-1.8.0-6.amzn2023.0.2.x86_64                       1/1
  Running scriptlet: tree-1.8.0-6.amzn2023.0.2.x86_64                       1/1
  Verifying        : tree-1.8.0-6.amzn2023.0.2.x86_64                       1/1

Installed:
  tree-1.8.0-6.amzn2023.0.2.x86_64

Complete!
[ec2-user@ip-172-31-3-249 ~]$ tree
.
├── ansible.cfg
├── instance.yaml
├── inventory.txt
└── {
```

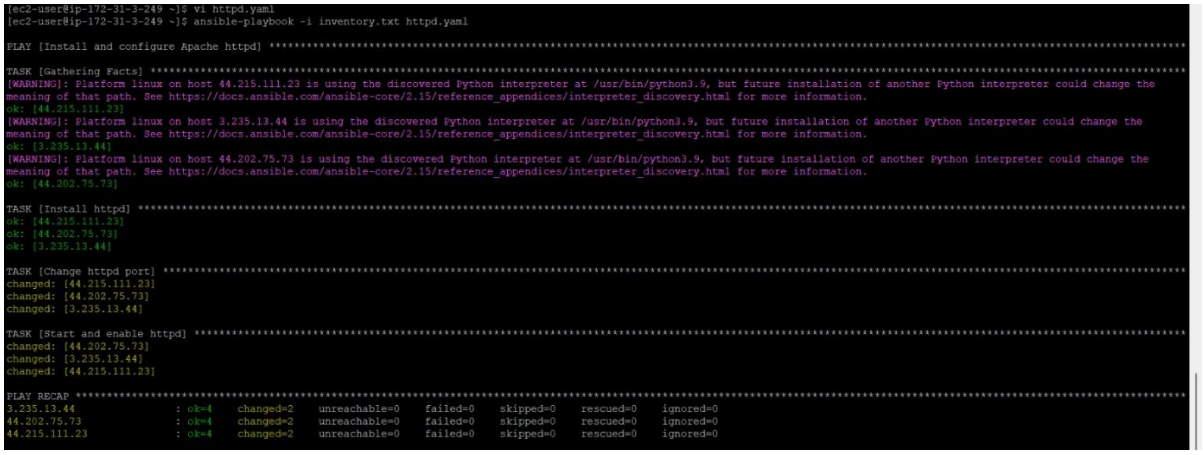
Changing the port number of httpd using ansible:

Code of httpd.yaml

```
---
- name: Install and configure Apache httpd
  hosts: all
  become: yes

  vars:
    httpd_port: 8080   # Change port here

  tasks:
    - name: Install httpd
      yum:
        name: httpd
        state: present

    - name: Change httpd port
      lineinfile:
        path: /etc/httpd/conf/httpd.conf
        regexp: '^Listen '
        line: "Listen {{ httpd_port }}"
```

```
    backup: yes

- name: Start and enable httpd
  service:
    name: httpd
    state: restarted
    enabled: yes
```

Creating a yaml file httpd.yaml for changing port number to 8080



Output of managed nodes after changing the port number of httpd

It works!

It works!

Ansible loops:

Ansible loops are a core feature for running a single task multiple times with different input values, which helps reduce code duplication and improves playbook readability and scalability. The modern and recommended keyword for this is loop.

Code of dev.yaml:

```
---
- hosts: all
  remote_user: ec2-user
  become: yes

  tasks:
    - name: Install packages one by one
      yum:
        name: "{{ item }}"
        state: present
      loop:
        - php
        - unzip
        - wget
```

Created a yaml file of loops and installed the packages