Web Application Using Ansible

Objective

To create an automation script to deploy an application using Ansible and Jinja2 template

Problem Statement and Motivation

Real-time scenario:

You have joined as a DevOps engineer in XYZ Pvt. Ltd. It is a platform where individuals can create their profile and start blogging on various topics. The application is ready to be hosted on a server.

You are tasked with implementing an Ansible script to deploy this application on a remote Nginx server.

Tasks

The following tasks outline the process of deploying web application on a remote server:

1. Create an inventory file to define the remote server(s)
2. Write a YAML playbook with tasks for installing Nginx, copying web application files, deploying the Nginx configuration, and enabling the site
3. Create a directory for templates and a Jinja2 template for the Nginx configuration
4. Define variables in the playbook for application details and Nginx configuration
5. Include tasks in the playbook for installing Nginx, copying application files, deploying Nginx configuration, and enabling the Nginx site
6. Execute the playbook to deploy the web application on the remote server

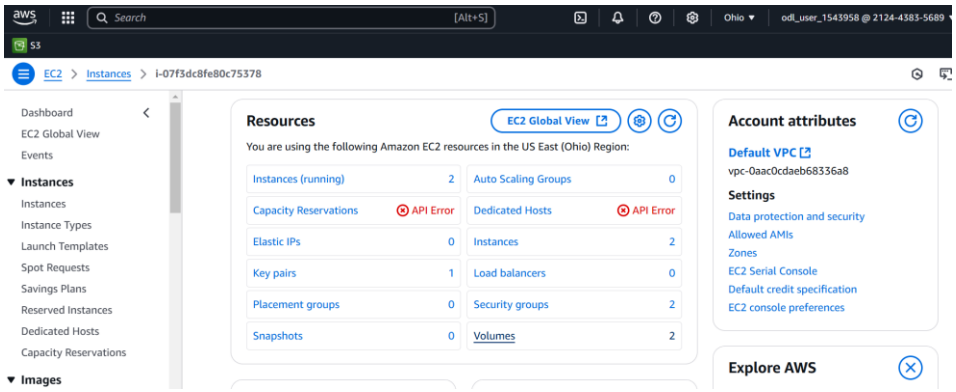
Procedure

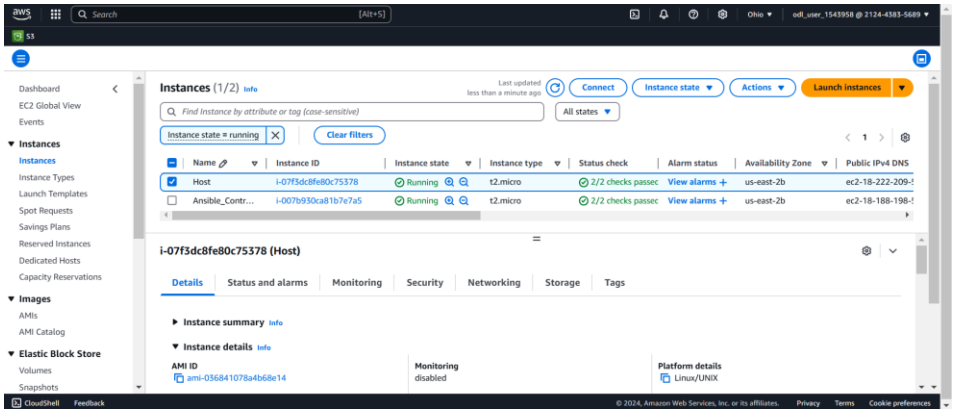
1.Log in to Amazon Web Services

2.Launcing 2 EC2 Instances

1.Ansible Controller

2.Host

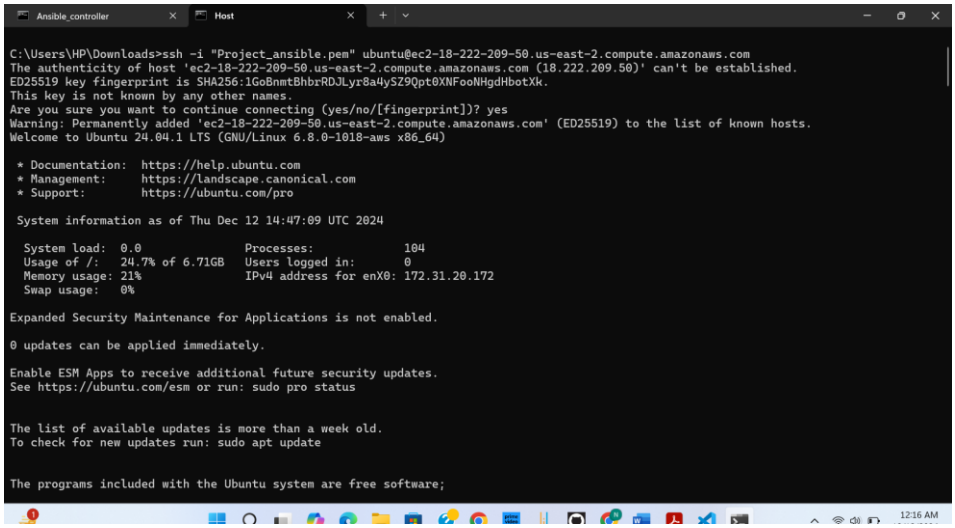




3.Connecting to instances through Command prompt and SSH

```
ubuntu@ip-172-31-19-125:~$ client_loop: send disconnect: Connection reset

C:\Users\HP\Downloads>ssh -i "Project_ansible.pem" ubuntu@ec2-18-188-198-52.us-east-2.compute.amazonaws.com
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

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

 System information as of Thu Dec 12 18:46:03 UTC 2024

  System load:  0.0               Processes:             123
  Usage of /:   35.3% of 6.71GB   Users logged in:       1
  Memory usage: 33%               IPv4 address for enX0: 172.31.19.125
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

46 updates can be applied immediately.
31 of these updates are standard security updates.
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: Thu Dec 12 18:29:47 2024 from 152.59.212.126
ubuntu@ip-172-31-19-125:~$
```

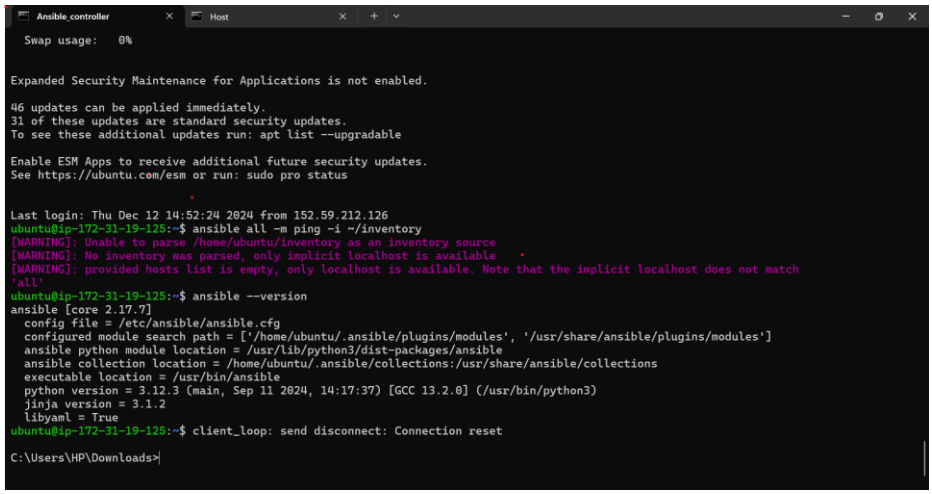
On Host Machine



4.Install Ansible on 1 machine i.e on Ansible Controller

```
sudo apt update
sudo apt install -y software-properties-common
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install -y ansible
```

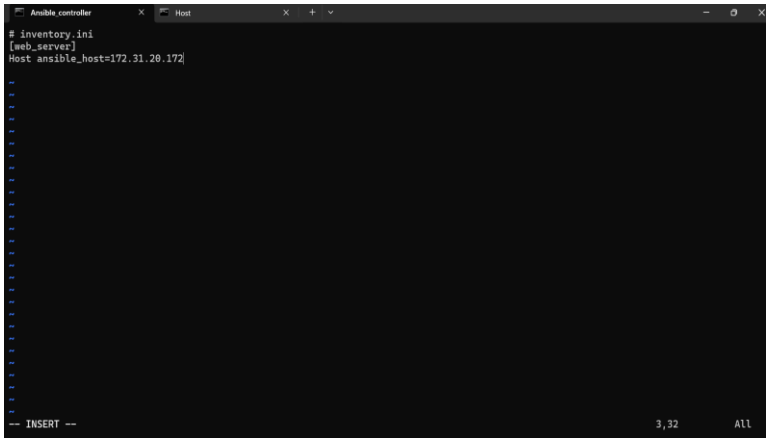
Verify version



5.Setup Host and inventories

Create an inventory

Create a file vi ~/inventory



5.Connecting Ansible and host machine

Setup SSH connection between Ansible and host machines

- On the **control** machine
  - Check is a SSH keypair exists already
    - ll ~/.ssh
    - If SSH key-pair don't exist, generate SSH keys
      - ssh-keygen -t rsa
      - (just press enter for any input prompts)
  - Print SSH public Key
    - cat ~/.ssh/id_rsa.pub
  - Copy the above printed key

- On **each Host** machine

a. Connect to each host instance

b. Add public key (copied from the master machine) to the end of below file as a new line (Please DO NOT delete any existing data from the below file)

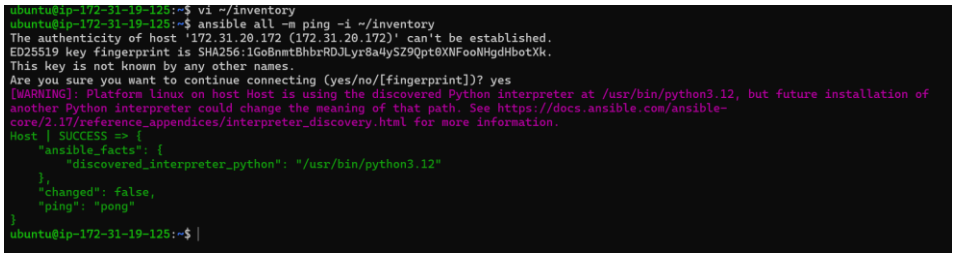
i.nano ~/.ssh/authorized_keys

6.Verify the SSH connection to hosts

On Ansible machine

```
ansible all -m ping -i ~/inventory
```

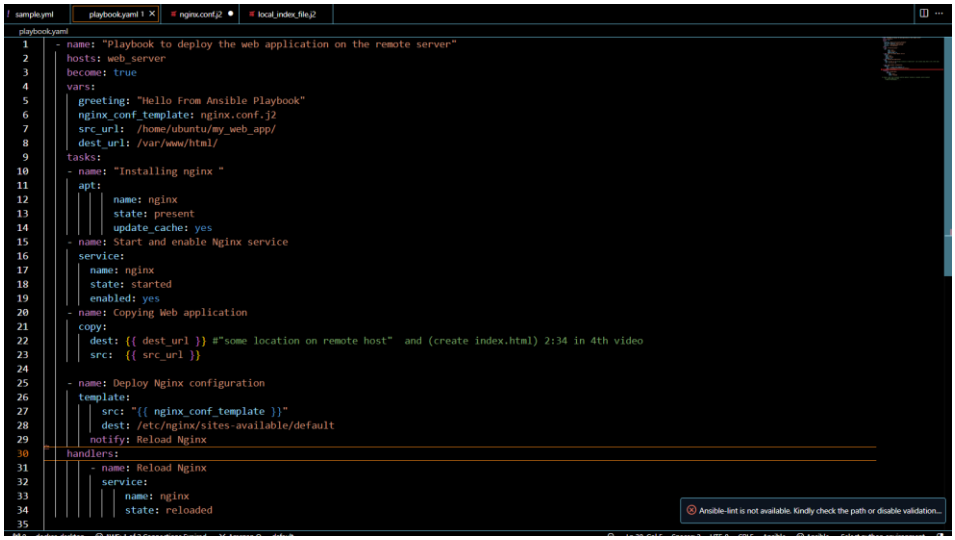
Expected output



Commented [NN1]: Pinging all hosts

Write a YAML Playbook (Playbook.yaml)

Next, write the playbook with tasks for installing Nginx, copying web application files, deploying the Nginx configuration, and enabling the site.



Create a Directory for Templates and a Jinja2 Template

Create a directory for templates and add a Jinja2 template for the Nginx configuration

$ mkdir templates

Create the Jinja2 template file nginx.conf.j2 in the templates directory:

$ Vi nginx.conf.j2

nginx.conf.j2

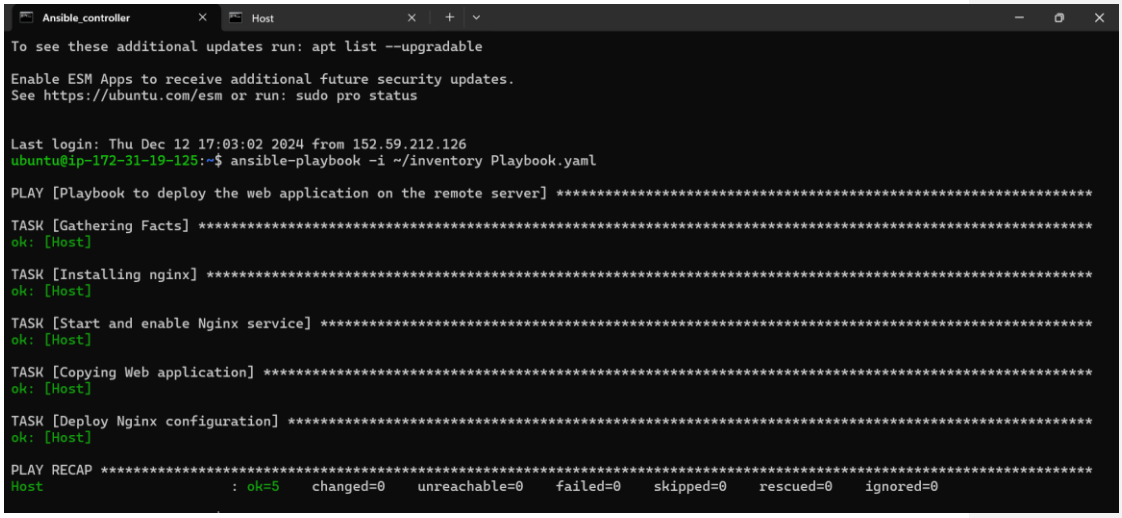
```
# templates/nginx.conf.j2
server { listen 80;
        server_name http://18.222.209.50/;
        root {{ dest_url }};
        index index.html index.htm;
        location / {
        try_files $uri $uri/ =404;
         } }
```

Create sample Web app

Created an index.html file in /home/ubuntu/my_web_app/index.html

Executed Command (Output)

Ansible-playbook -I ~/inventory Playbook.yaml

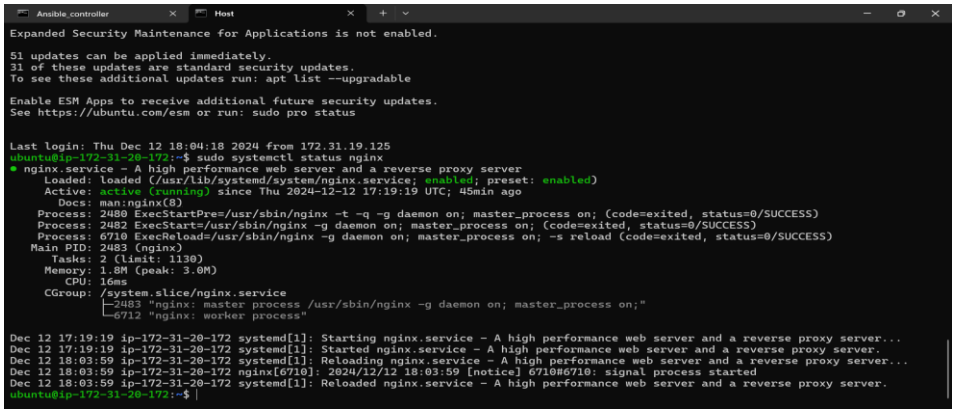


Steps To check NGINX Server deployed on webserver

To verify that your web application has been deployed successfully on your web server

1.Verify Nginx Status

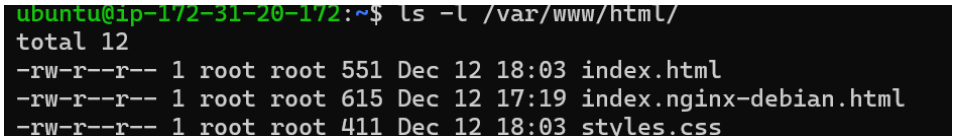
sudo systemctl status nginx



2. **Check the Web Root Directory**

Ensure that the web application files have been copied to the correct directory (/var/www/html/):

ls -l /var/www/html/



We should see Our web application files listed here.

3. Check Nginx Configuration

Verify that the Nginx configuration is set up correctly. You can test the configuration with the following command:

sudo nginx -t

```
ubuntu@ip-172-31-20-172:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
ubuntu@ip-172-31-20-172:~$
```

If the configuration is correct, you should see a message indicating that the test was successful.

4.Access the Web Application

http://18.222.209.50/