Name : Komal Mhetre

Role : DevOps Engineer

Ansible

Introduction :-

What is ansible and why we need to use?

Ansible, as an open-source automation tool, or platform, is used for IT tasks such as configuration management, application deployment, intraservice orchestration, and provisioning.

You can use ansible to manage one or many computers at once.

Can do task like configuration management, application deployment, infrastructure setup etc.

The detailed about resources to manage (hosts) kept in file called Ansible inventory.

This instructions to run on hosts are kept in a YAML files called playbooks.

When playbook is run, connection made to the host systems and playbook instructions are executed on the hosts.

Ansible is written in Python, Powershell, Shell.

Ansible Architecture :-

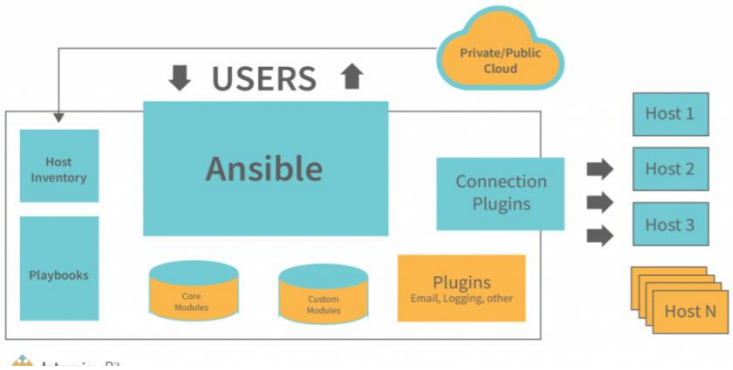
Control Node: The machine from which Ansible is run. It stores the inventory of managed nodes, playbooks, and configuration files. Ansible commands and playbooks are executed from the control node.

Managed Nodes: The remote servers or devices that are managed by Ansible. These nodes can be configured and controlled by the control node using SSH or WinRM.

Inventory: The inventory is a file or directory containing a list of managed nodes. It defines the hosts and groups of hosts that Ansible will manage. The inventory can be static or dynamic, allowing for dynamic discovery of hosts.

Modules: Ansible uses modules to perform specific tasks on managed nodes. Modules are small programs that are executed on the managed nodes and perform tasks such as installing packages, copying files, and managing services**.**

Playbooks: Playbooks are YAML files that define a set of tasks to be executed on managed nodes. Playbooks consist of one or more plays, each of which consists of a list of tasks and configurations.



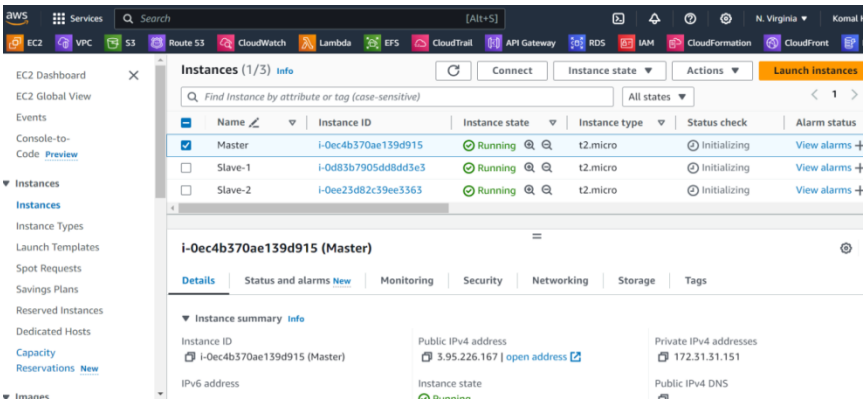
Ansible Documentaion :-

- https://docs.ansible.com/ansible/latest/index.html

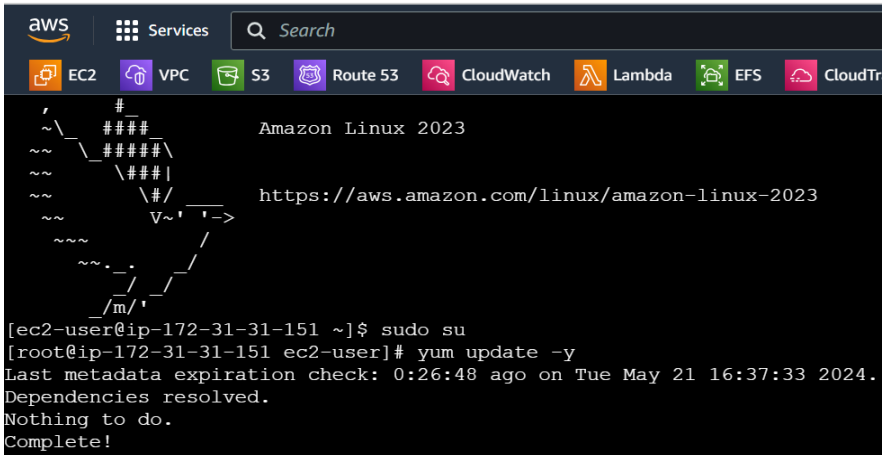
Practical :-

1. Launch three EC2 instances:
   Name one instance master.
   Name the other two instances slave-1 and slave-2.
   Choose an appropriate Amazon Machine Image

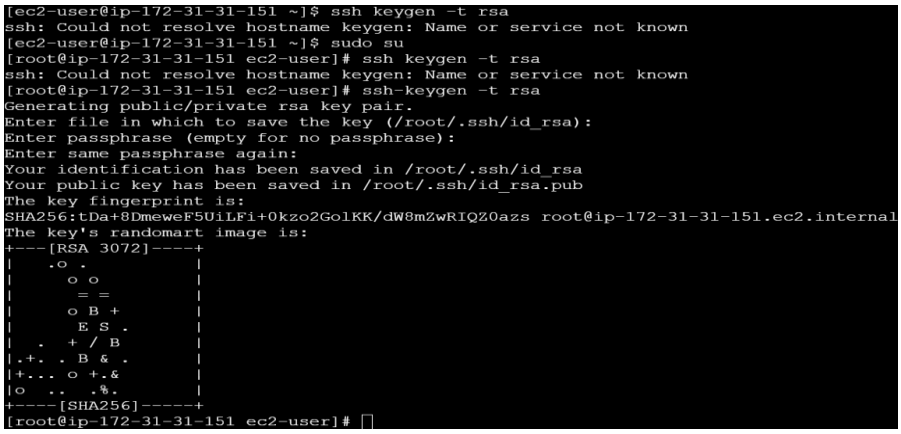
Select an instance type (t2.micro)
Configure instance details, add storage, and configure security groups to allow SSH access (port 22).
Create or select a key pair to access the instances.

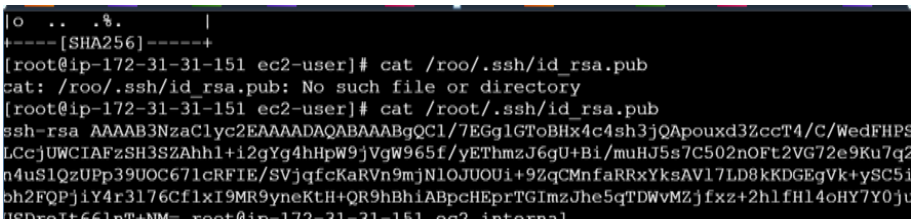


2. Connect Master Instance.
   Update the package and intall ansible
   Sudo yum update -y
   Sudo yum install python3
   Sudo yum install ansible -y



3. Generate SSH Key on Master Node:-
   ssh-keygen –t rsa
   cat /root/.ssh/id_rsa.pub
   /root/.ssh/id_rsa.pub: This is the path to the file containing the public SSH key for the root user.





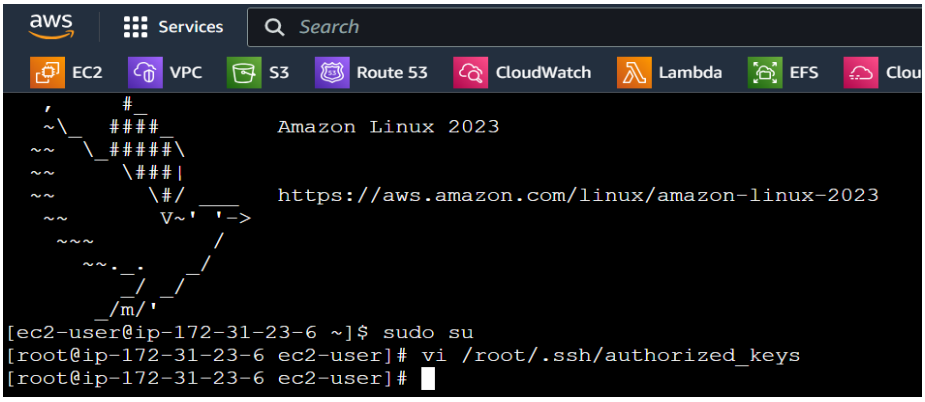
After that Connect slave-1.

Sudo su

Yum update –y

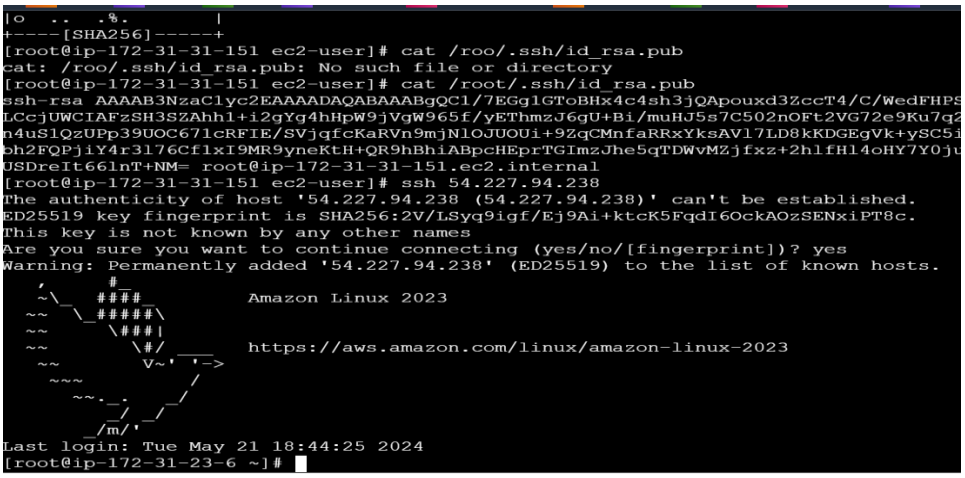
And then go to the authorized file…

Vi /root/.ssh/authorized_keys

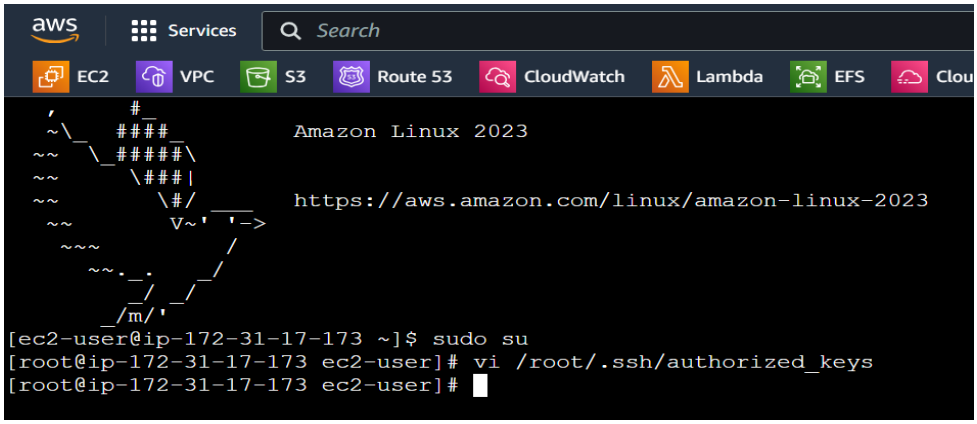


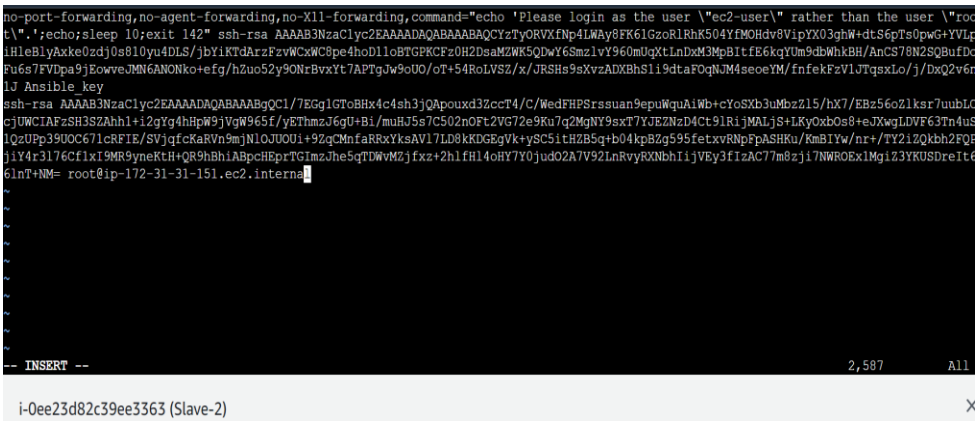
Copy that ssh key and paste the slave1 server and save it.

Run the ssh slave1_public_IP command on your master server. When you execute this command, you will log in to the slave 1 server.



You will follow the same process for the slave 2 server





Configure Ansible Inventory File on Master Node

Create and edit the inventory file:

Add your instances to the inventory file use private IP for security

For this use below command…

Vi /etc/ansible/hosts

Test the Connection

ansible all -m ping

```
[root@ip-172-31-31-151 ec2-user]# sudo vi /etc/ansible/hosts
[root@ip-172-31-31-151 ec2-user]# ansible all -m ping
The authenticity of host '172.31.23.6 (172.31.23.6)' can't be established.
ED25519 key fingerprint is SHA256:2V/LSyq9igf/Ej9Ai+ktcK5FqdI6OckAOzSENxiPT8c.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: 54.227.94.238
The authenticity of host '172.31.17.173 (172.31.17.173)' can't be established.
ED25519 key fingerprint is SHA256:a8y6C4NXJu1sD6X+mVCisRDS9p4cWQOlx+KbDIYTqh8.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:3: 3.88.180.140
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
[WARNING]: Platform linux on host 172.31.23.6 is using the discovered Python interpreter at /usr/bin/python
n
of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
172.31.23.6 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
```

here we can see a successful response from each slave node.

```
[root@ip-172-31-31-151 ec2-user]# sudo ansible all -m ping
[WARNING]: Platform linux on host 172.31.23.6 is using the discovered Python int
installation of another Python interpreter could change the meaning of that path
core/2.15/reference_appendices/interpreter_discovery.html for more information.
172.31.23.6 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.17.173 is using the discovered Python i
installation of another Python interpreter could change the meaning of that path
core/2.15/reference_appendices/interpreter_discovery.html for more information.
172.31.17.173 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
[root@ip-172-31-31-151 ec2-user]#
```

Here we will write an ansible playbook before this you can install nginx in master server.

Sudo yum install nginx –y

Sudo systemctl enable nginx

Sudo systemctl start nginx

Sudo systemctl status nginx

create a playbook file

vi nginx.yml

add tasks to the playbook

Adjust the playbook tasks according to your requirements.

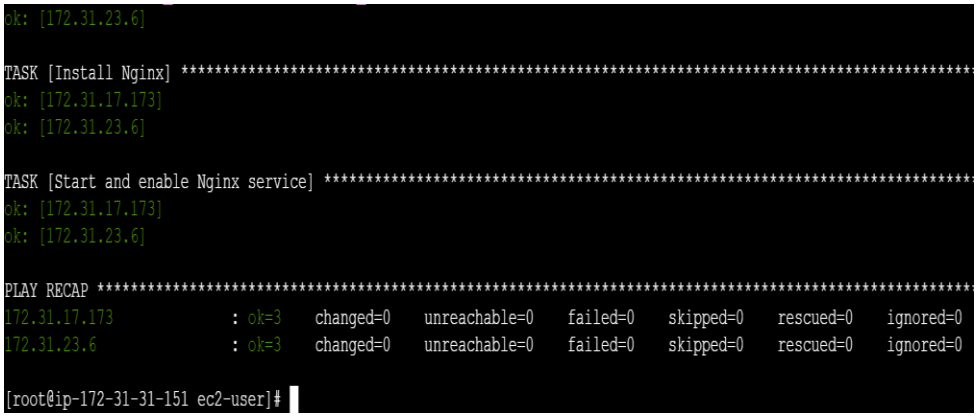
```
---
- name: Install and start Nginx on web servers
  hosts: web
  become: yes
  tasks:
    - name: Install Nginx
      yum:
        name: nginx
        state: present

    - name: Start and enable Nginx service
      service:
        name: nginx
        state: started
        enabled: yes
~
~
~
~
~
-- INSERT --
```

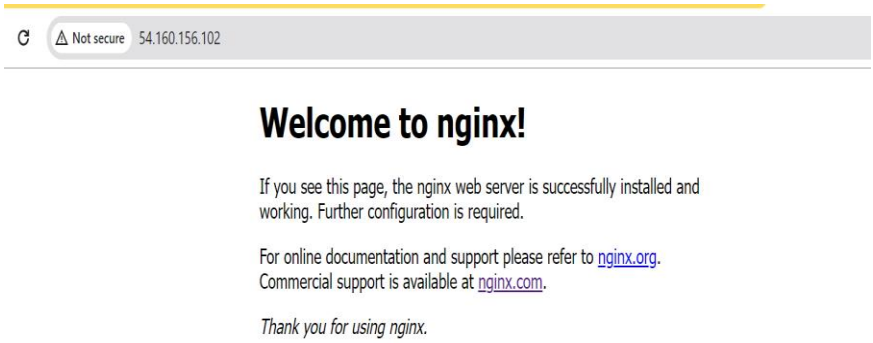
Run the Playbook using below command

ansible-playbook nginx.yml

This command will apply the tasks defined in site.yml to all servers listed in the slaves group in your inventory.



Copy the public IP of your slave-1 server. Open a web browser and paste the IP press Enter to see the result.



Do the same for the slave-2 server and see the result.