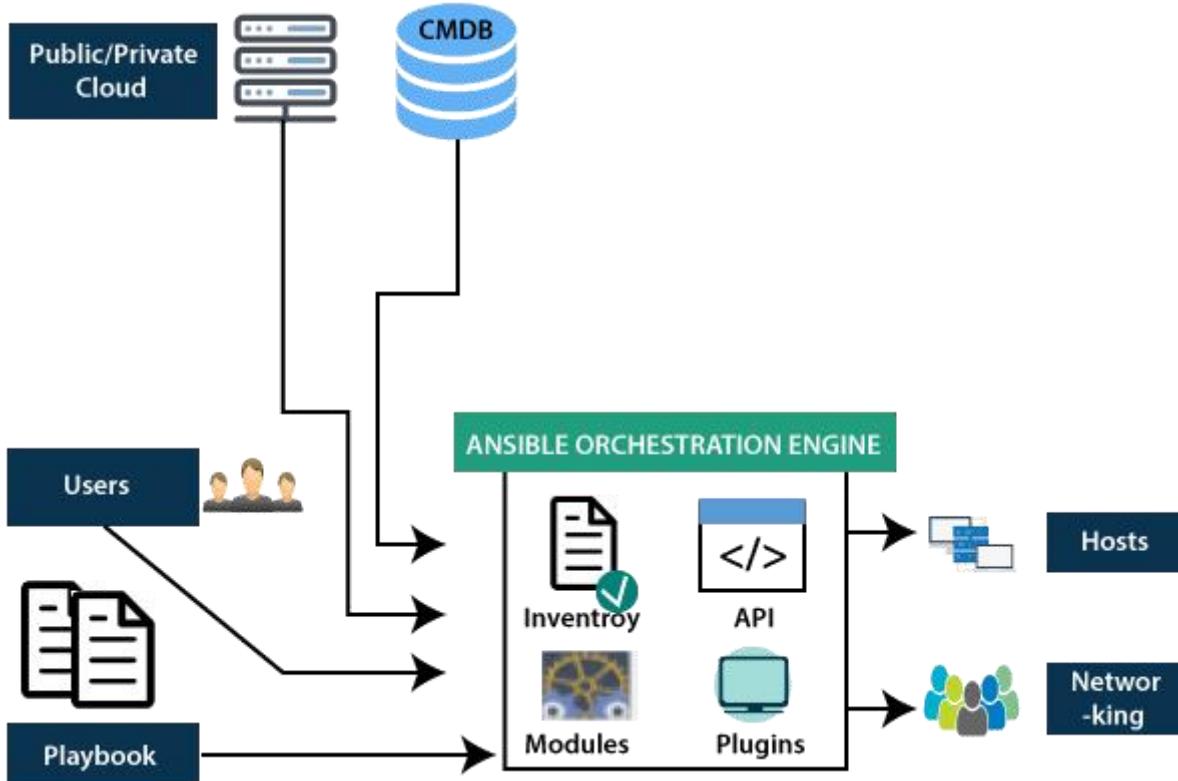


Ansible Masterclass with Handson

Ojas Jawale



Ansible Architecture-



The Ansible orchestration engine interacts with a user who is writing the Ansible playbook to execute the Ansible orchestration and interact along with the services of private or public cloud and configuration management database.

1. Inventory - Inventory is lists of nodes or hosts having their IP addresses, databases, servers, etc. which are need to be managed.

2. API's - The Ansible API's works as the transport for the public or private cloud services.

3. Modules - Ansible connected the nodes and spread out the Ansible modules programs. Ansible executes the modules and removed after finished. These modules can reside on any machine; no database or servers are required here. You can work with the chose text editor or a terminal or version control system to keep track of the changes in the content.

4. Plugins - Plugins is a piece of code that expends the core functionality of Ansible. There are many useful plugins, and you also can write your own.

5. Playbooks - Playbooks consist of your written code, and they are written in YAML format, which describes the tasks and executes through the Ansible. Also, you can launch the tasks synchronously and asynchronously with playbooks.

6. Hosts - In the Ansible architecture, hosts are the node systems, which are automated by Ansible, and any machine such as RedHat, Linux, Windows, etc.

7. Networking - Ansible is used to automate different networks, and it uses the simple, secure, and powerful agentless automation framework for IT operations and development. It uses a type of data model which separated from the Ansible automation engine that spans the different hardware quite easily.

8. Cloud - A cloud is a network of remote servers on which you can store, manage, and process the data. These servers are hosted on the internet and storing the data remotely rather than the local server. It just launches the resources and instances on the cloud, connect them to the servers, and you have good knowledge of operating your tasks remotely.

9. CMDB - CMDB is a type of repository which acts as a data warehouse for the IT installations.

1. Create 3 AWS EC2 instance for Ansible-

- One for MASTER
- Two for Nodes

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

[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*

Recents

Quick Start



[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

Free tier eligible

ami-0e86e20dae9224db8 (64-bit (x86)) / ami-096ea6a12ea24a797 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

All generations

[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.

Ansible-key

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 ↗](#)

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

We'll create a new security group called '**launch-wizard-5**' with the following rules:

Allow SSH traffic from

Helps you connect to your instance

Anywhere

0.0.0.0/0



Allow HTTPS traffic from the internet

To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet

To set up an endpoint, for example when creating a web server

▼ Summary

Number of instances | [Info](#)

3

When launching more than 1 instance, [consider EC2 Auto Scaling](#)

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64...[read more](#)
ami-0e86e20dae9224db8

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 20 GiB

- Three EC2 instance launch is in progress, then change their names accordingly.

Instances (3/3) Info		Last updated less than a minute ago	Connect	Instance state ▾	Actions ▾	Launch instances	▼
				Running ▾			
<input checked="" type="checkbox"/>	Name ↴	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone ▾
<input checked="" type="checkbox"/>	Ansible-master	i-0e7c5603efce3b1d9	🕒 Running ⓘ ⓘ	t2.micro	⌚ Initializing	View alarms +	us-east-1b
<input checked="" type="checkbox"/>	node-1	i-002fb0da058eca1a1	🕒 Running ⓘ ⓘ	t2.micro	⌚ Initializing	View alarms +	us-east-1b
<input checked="" type="checkbox"/>	node-2	i-0e8550bcb1965ab85	🕒 Running ⓘ ⓘ	t2.micro	⌚ Initializing	View alarms +	us-east-1b

- Connect master using private key.

Installation of Ansible on master-

- Add Ansible repository,
- `sudo apt-add-repository ppa:ansible/ansible`

```
ubuntu@ip-172-31-93-109:~$ sudo apt-add-repository ppa:ansible/ansible
Repository: 'Types: deb
URIs: https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/
Suites: noble
Components: main
'
Description:
Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts or custom code to deploy and update your applications— automate in a language that approaches plain English, using S
SH, with no agents to install on remote systems.

http://ansible.com/

If you face any issues while installing Ansible PPA, file an issue here:
https://github.com/ansible-community/ppa/issues
More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.|
```

- Update repository
 - sudo apt update
- Install Ansible,
 - sudo apt install ansible

```
ubuntu@ip-172-31-93-109:~$ sudo apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ansible-core python3-jmespath python3-kerberos python3-nacl python3-ntlm-auth python3-packaging python3-paramiko
  python3-requests-ntlm python3-resolvelib python3-winrm python3-xmlltodict sshpass
Suggested packages:
  python-nacl-doc python3-gssapi python3-invoke
The following NEW packages will be installed:
  ansible ansible-core python3-jmespath python3-kerberos python3-nacl python3-ntlm-auth python3-packaging python3-paramiko
  python3-requests-ntlm python3-resolvelib python3-winrm python3-xmlltodict sshpass
0 upgraded, 13 newly installed, 0 to remove and 86 not upgraded.
Need to get 18.5 MB of archives.
After this operation, 203 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

- Verify installation,
 - ansible --version

```
ubuntu@ip-172-31-93-109:~$ ansible --version
ansible [core 2.16.10]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/ubuntu/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.12.3 (main, Apr 10 2024, 05:33:47) [GCC 13.2.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
ubuntu@ip-172-31-93-109:~$ |
```

- Ansible runs on SSH and built on Python.

Now, we need private on master node to communicate with nodes. Hence create private key while launch instance needs to be copy on master.

- Copy SSH private key from local machine to master.
- Go to directory where key is stored and open CMD prompt.
- Run below command,

- scp -i "Ansible-key.pem" Ansible-key.pem ubuntu@Public_IP_of_master

```
PS C:\Users\ojash\OneDrive\Desktop\Study\Keys> scp -i "Ansible-key.pem" Ansible-key.pem ubuntu@107.21.44.9:/home/ubuntu/keys/
The authenticity of host '107.21.44.9 (107.21.44.9)' can't be established.
ED25519 key fingerprint is SHA256:igvbIez7E5cERBTI0LDPmMWIGEtndVm6IFxhVS09RMc.
This host key is known by the following other names/addresses:
  C:\Users\ojash\.ssh\known_hosts:70: ec2-107-21-44-9.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Please type 'yes', 'no' or the fingerprint:
Warning: Permanently added '107.21.44.9' (ED25519) to the list of known hosts.
Ansible-key.pem                                         100% 1678     4.7KB/s   00:00
PS C:\Users\ojash\OneDrive\Desktop\Study\Keys> |
```

```

ubuntu@ip-172-31-93-109:~/keys$ pwd
/home/ubuntu/keys
ubuntu@ip-172-31-93-109:~/keys$ ls
Ansible-key.pem
ubuntu@ip-172-31-93-109:~/keys$ |

```

Set Up Ansible Inventory-

- The Ansible hosts file (/etc/ansible/hosts) is where you define your inventory of servers.
- Ensure that private key file has appropriate permissions,
 - **chmod 400 Ansible-key.pem**
 - sudo nano /etc/ansible/hosts

```

GNU nano 7.2                               /etc/ansible/hosts
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers:

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

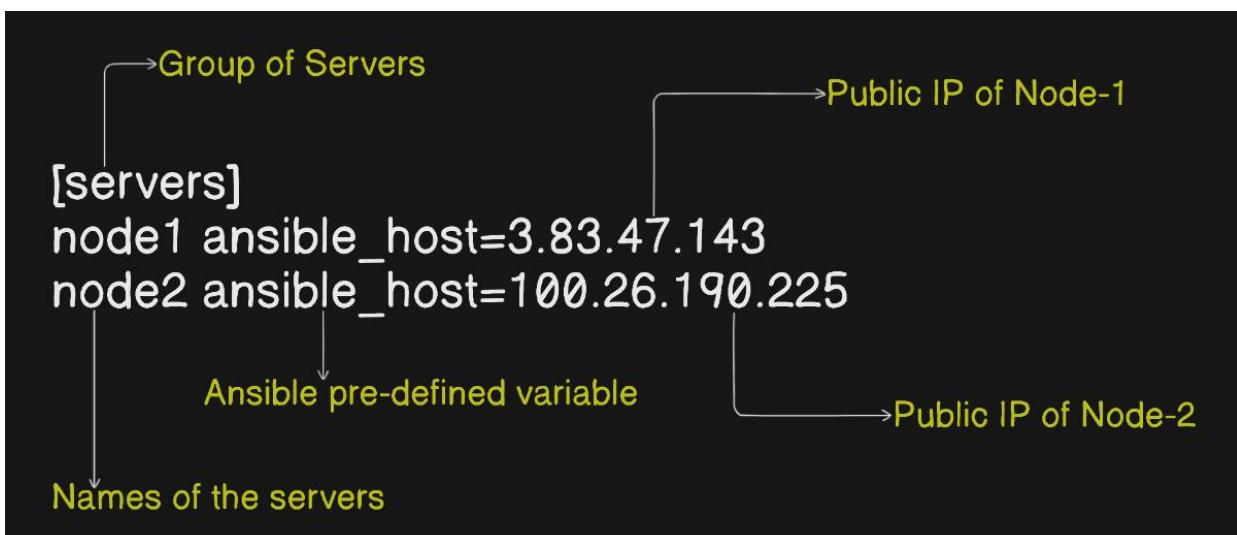
# Ex 2: A collection of hosts belonging to the 'webservers' group:

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

```

[Read 54 lines]

- Add your servers specific group,



```

[servers]
node1 ansible_host=3.83.47.143
node2 ansible_host=100.26.190.225

[all:vars]
ansible_python_interpreter=/usr/bin/python3
ansible_user=ubuntu
ansible_ssh_private_key_file=/home/ubuntu/keys/Ansible-key.pem

```

- Save and close the /etc/ansible/hosts file.

- Verify the inventory file,

- ansible-inventory --list

```

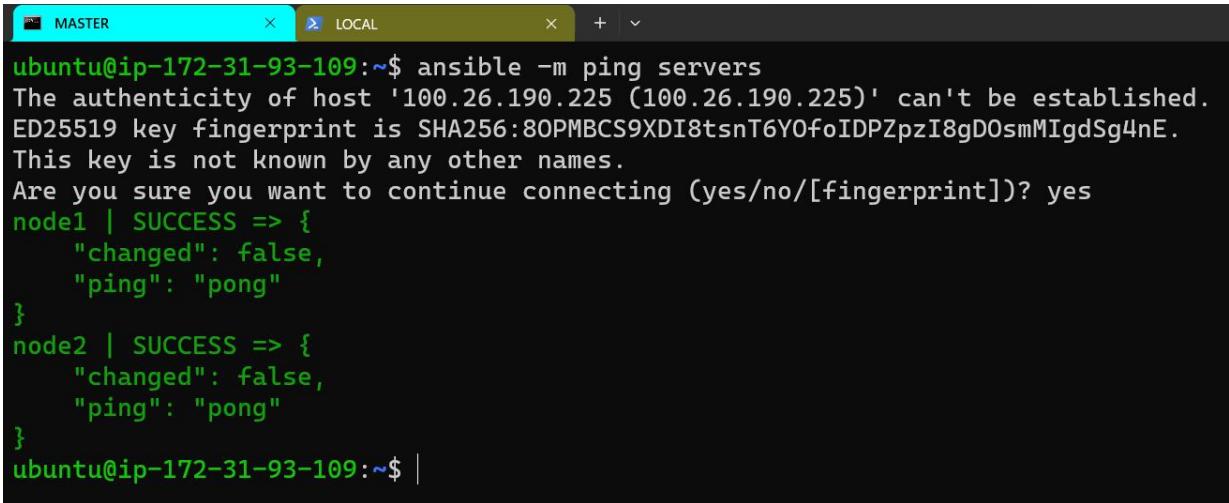
ubuntu@ip-172-31-93-109:~$ ansible-inventory --list
{
    "_meta": {
        "hostvars": {
            "node1": {
                "ansible_host": "3.83.47.143",
                "ansible_python_interpreter": "/usr/bin/python3",
                "ansible_ssh_private_key_file": "/home/ubuntu/keys/Ansible-key.pem",
                "ansible_user": "ubuntu"
            },
            "node2": {
                "ansible_host": "100.26.190.225",
                "ansible_python_interpreter": "/usr/bin/python3",
                "ansible_ssh_private_key_file": "/home/ubuntu/keys/Ansible-key.pem",
                "ansible_user": "ubuntu"
            }
        }
    },
    "all": {
        "children": [
            "ungrouped",
            "servers"
        ]
    },
    "servers": {
        "hosts": [
            "node1",
            "node2"
        ]
    }
}

```

- Now, try connectivity between master and nodes,

- Using ping module,

- ansible -m ping servers/node1/node2 (-m : built-in modules)



```

ubuntu@ip-172-31-93-109:~$ ansible -m ping servers
The authenticity of host '100.26.190.225 (100.26.190.225)' can't be established.
ED25519 key fingerprint is SHA256:80PMBCS9XDI8tsnT6Y0foIDPZpzI8gD0smMIgdSg4nE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
node1 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
node2 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-93-109:~$ |

```

- Check system resources of nodes using *Ansible Adhoc commands*,

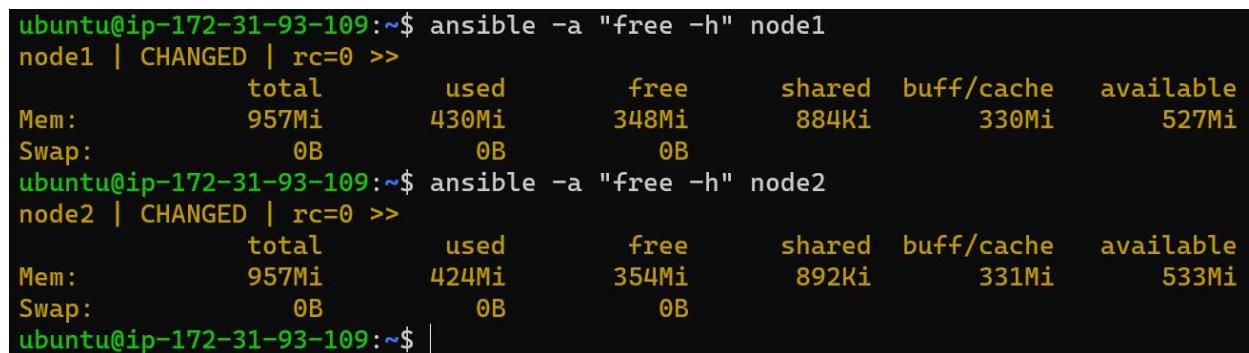
- ansible -a "free -h" servers (-a : Adhoc command)



```

ubuntu@ip-172-31-93-109:~$ ansible -a "free -h" servers
node2 | CHANGED | rc=0 >>
      total        used        free      shared  buff/cache   available
Mem:   957Mi       424Mi     354Mi     892Ki       331Mi      533Mi
Swap:      0B         0B         0B
node1 | CHANGED | rc=0 >>
      total        used        free      shared  buff/cache   available
Mem:   957Mi       430Mi     348Mi     884Ki       330Mi      527Mi
Swap:      0B         0B         0B
ubuntu@ip-172-31-93-109:~$ |

```



```

ubuntu@ip-172-31-93-109:~$ ansible -a "free -h" node1
node1 | CHANGED | rc=0 >>
      total        used        free      shared  buff/cache   available
Mem:   957Mi       430Mi     348Mi     884Ki       330Mi      527Mi
Swap:      0B         0B         0B
ubuntu@ip-172-31-93-109:~$ ansible -a "free -h" node2
node2 | CHANGED | rc=0 >>
      total        used        free      shared  buff/cache   available
Mem:   957Mi       424Mi     354Mi     892Ki       331Mi      533Mi
Swap:      0B         0B         0B
ubuntu@ip-172-31-93-109:~$ |

```

- ansible -a "sudo apt update" servers

Above adhoc command will update the repository on nodes mentioned in /etc/ansible/hosts file.

```

ubuntu@ip-172-31-93-109:~$ ansible -a "df -h" servers
node2 | CHANGED | rc=0 >>
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       19G   1.8G   17G  10% /
tmpfs          479M    0  479M  0% /dev/shm
tmpfs          192M  888K  191M  1% /run
tmpfs          5.0M    0  5.0M  0% /run/lock
/dev/xvda16     881M   76M  744M 10% /boot
/dev/xvda15     105M   6.1M  99M  6% /boot/efi
tmpfs          96M   12K   96M  1% /run/user/1000
node1 | CHANGED | rc=0 >>
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       19G   1.8G   17G  10% /
tmpfs          479M    0  479M  0% /dev/shm
tmpfs          192M  884K  191M  1% /run
tmpfs          5.0M    0  5.0M  0% /run/lock
/dev/xvda16     881M   76M  744M 10% /boot
/dev/xvda15     105M   6.1M  99M  6% /boot/efi
tmpfs          96M   12K   96M  1% /run/user/1000
ubuntu@ip-172-31-93-109:~$ 

```

Custom Inventory in Ansible-

- Create directory for custom inventories
- mkdir inventories

Now, create two custom inventories for **Dev** and **Prod** environment.

Custom inventory : Dev,

- *node1 will be dev environment*
- cd inventories
- nano dev

```

GNU nano 7.2                                         dev
[dev]
node1 ansible_host=3.83.47.143

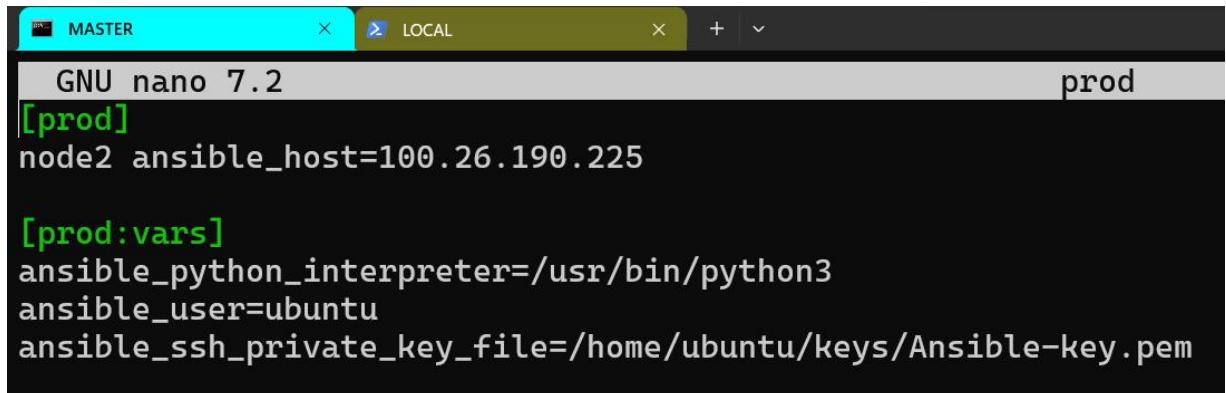
[dev:vars]
ansible_python_interpreter=/usr/bin/python3
ansible_user=ubuntu
ansible_ssh_private_key_file=/home/ubuntu/keys/Ansible-key.pem

```

- Above custom inventory is for **dev** environment only and **node1** will be **dev**.

Custom inventory : Prod,

- node2 will be prod environment
- cd inventories
- nano prod



```
MASTER × LOCAL × + ▾
GNU nano 7.2
prod
[prod]
node2 ansible_host=100.26.190.225

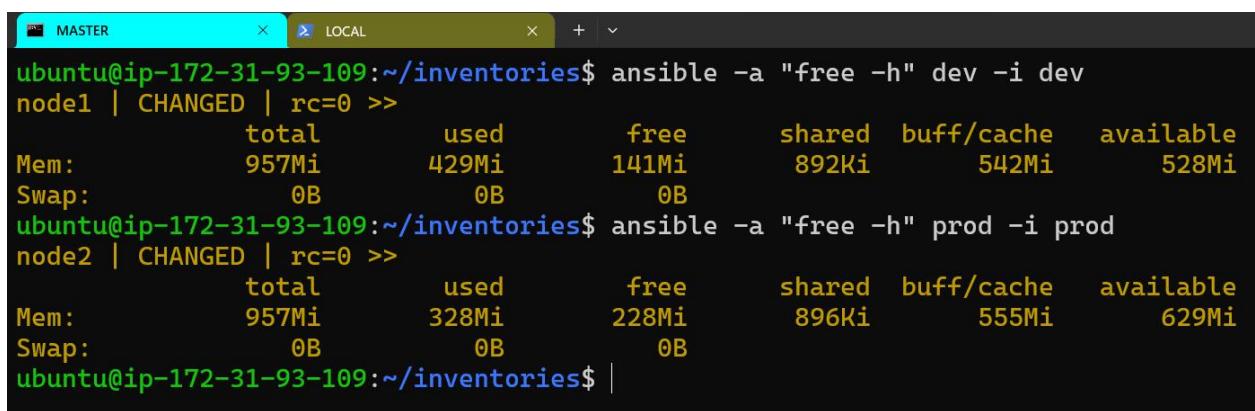
[prod:vars]
ansible_python_interpreter=/usr/bin/python3
ansible_user=ubuntu
ansible_ssh_private_key_file=/home/ubuntu/keys/Ansible-key.pem
```

Trying to get details of nodes using custom inventories,

- pwd : /home/ubuntu/inventories
- ansible -a "free -h" dev -i dev (-i : for inventory)

Or

- ansible -a "free -h" dev -i /home/ubuntu/inventories/dev



```
MASTER × LOCAL × + ▾
ubuntu@ip-172-31-93-109:~/inventories$ ansible -a "free -h" dev -i dev
node1 | CHANGED | rc=0 >>
      total        used         free        shared      buff/cache   available
Mem:       957Mi       429Mi       141Mi       892Ki       542Mi       528Mi
Swap:          0B          0B          0B
ubuntu@ip-172-31-93-109:~/inventories$ ansible -a "free -h" prod -i prod
node2 | CHANGED | rc=0 >>
      total        used         free        shared      buff/cache   available
Mem:       957Mi       328Mi       228Mi       896Ki       555Mi       629Mi
Swap:          0B          0B          0B
ubuntu@ip-172-31-93-109:~/inventories$ |
```

- To get the details of inventory **dev**,
 - ansible-inventory --list -i dev/prod

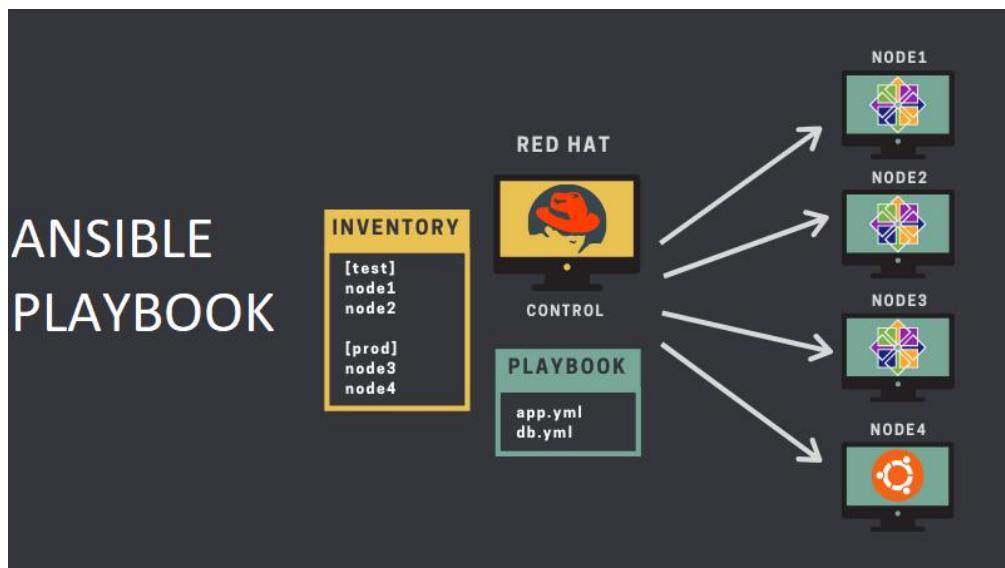
```

MASTER          LOCAL          + 
ubuntu@ip-172-31-93-109:~/inventories$ ansible-inventory --list -i dev
{
    "_meta": {
        "hostvars": {
            "node1": {
                "ansible_host": "3.83.47.143",
                "ansible_python_interpreter": "/usr/bin/python3",
                "ansible_ssh_private_key_file": "/home/ubuntu/keys/Ansible-key.pem",
                "ansible_user": "ubuntu"
            }
        }
    },
    "all": {
        "children": [
            "ungrouped",
            "dev"
        ]
    },
    "dev": {
        "hosts": [
            "node1"
        ]
    }
}
ubuntu@ip-172-31-93-109:~/inventories$ |

```

Playbooks in Ansible-

- Ansible Playbooks offer a repeatable, reusable, simple configuration management and multi-machine deployment system, one that is well suited to deploying complex applications.
- If you need to execute a task with Ansible more than once, write a playbook and put it under source control. Then you can use the playbook to push out new configuration or confirm the configuration of remote systems.



- pwd : /home/ubuntu
- mkdir playbooks
- nano playbook.yml

```

MASTER LOCAL + -
GNU nano 7.2
-
  name: Show date & kernel version Playbook
  hosts: servers
  tasks:
    - name: Show date
      command: date
    - name: Show kernel version
      command: uname -r

```

- Run the playbook,
- ansible-playbook playbook.yml

```

ubuntu@ip-172-31-93-109:~/playbooks$ ansible-playbook playbook.yml
PLAY [Show date & kernel version Playbook] ****
TASK [Gathering Facts] ****
ok: [node2]
ok: [node1]

TASK [Show date] ****
changed: [node2]
changed: [node1]

TASK [Show kernel version] ****
changed: [node2]
changed: [node1]

PLAY RECAP ****
node1              : ok=3    changed=2    unreachable=0    failed=0     skipped=0    rescued=0    ignor
ed=0
node2              : ok=3    changed=2    unreachable=0    failed=0     skipped=0    rescued=0    ignor
ed=0

ubuntu@ip-172-31-93-109:~/playbooks$ |

```

- Above shows the commands executed using playbook, but if you want to view detailed data, use -v : verbose (means elaborate the output).

```

TASK [Gathering Facts] ****
ok: [node2]
ok: [node1]

TASK [Show date] ****
changed: [node2] => {"changed": true, "cmd": ["date"], "delta": "0:00:00.004157", "end": "2024-08-25 12:49:47.556446", "msg": "", "rc": 0, "start": "2024-08-25 12:49:47.552289", "stderr": "", "stderr_lines": [], "stdout": "Sun Aug 25 12:49:47 UTC 2024", "stdout_lines": ["Sun Aug 25 12:49:47 UTC 2024"]}
changed: [node1] => {"changed": true, "cmd": ["date"], "delta": "0:00:00.004399", "end": "2024-08-25 12:49:47.604729", "msg": "", "rc": 0, "start": "2024-08-25 12:49:47.600330", "stderr": "", "stderr_lines": [], "stdout": "Sun Aug 25 12:49:47 UTC 2024", "stdout_lines": ["Sun Aug 25 12:49:47 UTC 2024"]}

TASK [Show kernel version] ****
changed: [node2] => {"changed": true, "cmd": ["uname", "-r"], "delta": "0:00:00.003793", "end": "2024-08-25 12:49:47.956814", "msg": "", "rc": 0, "start": "2024-08-25 12:49:47.953021", "stderr": "", "stderr_lines": [], "stdout": "6.8.0-1012-aws", "stdout_lines": ["6.8.0-1012-aws"]}
changed: [node1] => {"changed": true, "cmd": ["uname", "-r"], "delta": "0:00:00.004179", "end": "2024-08-25 12:49:47.975941", "msg": "", "rc": 0, "start": "2024-08-25 12:49:47.975941", "stderr": "", "stderr_lines": [], "stdout": "6.8.0-1012-aws", "stdout_lines": ["6.8.0-1012-aws"]}

PLAY RECAP ****
node1              : ok=3    changed=2    unreachable=0    failed=0     skipped=0    rescued=0    ignor
ed=0
node2              : ok=3    changed=2    unreachable=0    failed=0     skipped=0    rescued=0    ignor
ed=0

```

Install Nginx using Ansible playbook-

- pwd : /home/ubuntu/playbooks
- nano install-nginx.yml

```
MASTER          X  LOCAL          X  +  ▾
GNU nano 7.2          install-nginx.yml

name: Install Nginx & Ensure that its enabled
hosts: dev
become: yes
tasks:
- name: Install Nginx
  ansible.builtin.apt:
    name: nginx
    state: latest

- name: Start and Enable Nginx
  ansible.builtin.service:
    name: nginx
    state: started
    enabled: yes
```

Explanation of each entry in playbook,

```
name: Install Nginx & Ensure that its enabled  
hosts: dev → Execute on Dev Server  
become: yes → Means, use Root priviledge as Sudo  
tasks:  
- name: Install Nginx  
  ansible.builtin.apt: → Ansible builtin Module  
    name: nginx  
    state: latest → Nginx latest version  
  
- name: Start and Enable Nginx  
  ansible.builtin.service: → Ansible builtin Module  
    name: nginx  
    state: started → Start nginx service  
    enabled: yes → Enable nginx service so that, even if server reboots the nginx should be automatically started
```

- Run playbook with the custom inventory using -i flag and absolute path of inventory.

```
ubuntu@ip-172-31-93-109:~/playbooks$ ansible-playbook install-nginx.yml -i /home/ubuntu/inventories/dev -v
Using /etc/ansible/ansible.cfg as config file

PLAY [Install Nginx & Ensure that its enabled] ****
TASK [Gathering Facts] ****
OK: [node1]

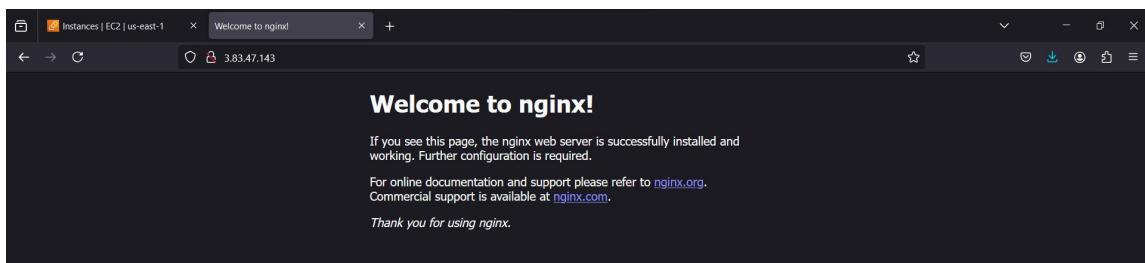
TASK [Install Nginx] ****
changed: [node1] => {"cache_update_time": 1724585741, "cache_updated": false, "changed": true, "stderr": "", "stderr_lines": [], "stdout": "Reading package lists...\nBuilding dependency tree...\nReading state information...\n\nThe following additional packages will be installed:\n  nginx-common\nSuggested packages:\n  fcgiwrap\n  nginx-doc\n  ssl-cert\nThe following NEW packages will be installed:\n  nginx\n  nginx-common\n0 upgraded, 2 newly installed, 0 to remove and 86 not upgraded.\nNeed to get 552 kB of archives.\nAfter this operation, 1596 kB of additional disk space will be used.\nGet:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 nginx-common all 1.24.0-2ubuntu7 [31.2 kB]\nGet:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 nginx amd64 1.24.0-2ubuntu7 [521 kB]\nPreconfiguring packages...\nFetched 552 kB in 0s (18.2 MB/s)\nSelecting previously unselected package nginx-common.\n(Reading database ...)\nReading database ... 5%\n(Reading database ... 10%)\n(Reading database ... 15%)\n(Reading database ... 20%)\n(Reading database ... 25%)\n(Reading database ... 30%)\n(Reading database ... 35%)\n(Reading database ... 40%)\n(Reading database ... 45%)\n(Reading database ... 50%)\n(Reading database ... 55%)\n(Reading database ... 60%)\n(Reading database ... 65%)\n(Reading database ... 70%)\n(Reading database ... 75%)\n(Reading database ... 80%)\n(Reading database ... 85%)\n(Reading database ... 90%)\n(Reading database ... 95%)\n(Reading database ... 100%)\n67741 files and directories currently installed.\nPreparing to unpack .../nginx-common_1.24.0-2ubuntu7_all.deb...\nUnpacking nginx-common (1.24.0-2ubuntu7)...\nSelecting previously unselected package nginx.\nPreparing to unpack .../nginx_1.24.0-2ubuntu7_amd64.deb...\nUnpacking nginx (1.24.0-2ubuntu7)...\nSetting up nginx (1.24.0-2ubuntu7)...\nSetting up nginx-common (1.24.0-2ubuntu7)...\nCreated symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.\nProcessing triggers for ufw (0.36.2-6)...\n"
}
```

- Check Nginx is installed or not in node1(dev server),

- ansible -a "nginx -v" dev -i /home/ubuntu/inventories/dev

```
ubuntu@ip-172-31-93-109:~/playbooks$ ansible -a "nginx -v" dev -i /home/ubuntu/inventories/dev
node1 | CHANGED | rc=0 >>
nginx version: nginx/1.24.0 (Ubuntu)
ubuntu@ip-172-31-93-109:~/playbooks$ |
```

- Access Nginx on browser using node1's public IP,



Install Apache using Ansible playbook-

- pwd : /home/ubuntu/playbooks
- nano install-apache.yml

```

MASTER X LOCAL X + -
GNU nano 7.2           install-apache.yml

name: Install Apache and ensure its enabled
hosts: prod
become: yes
tasks:
  - name: Install Apache
    apt:
      name: apache2
      state: latest

  - name: Start and enable Apache
    service:
      name: apache2
      state: started
      enabled: yes

```

```

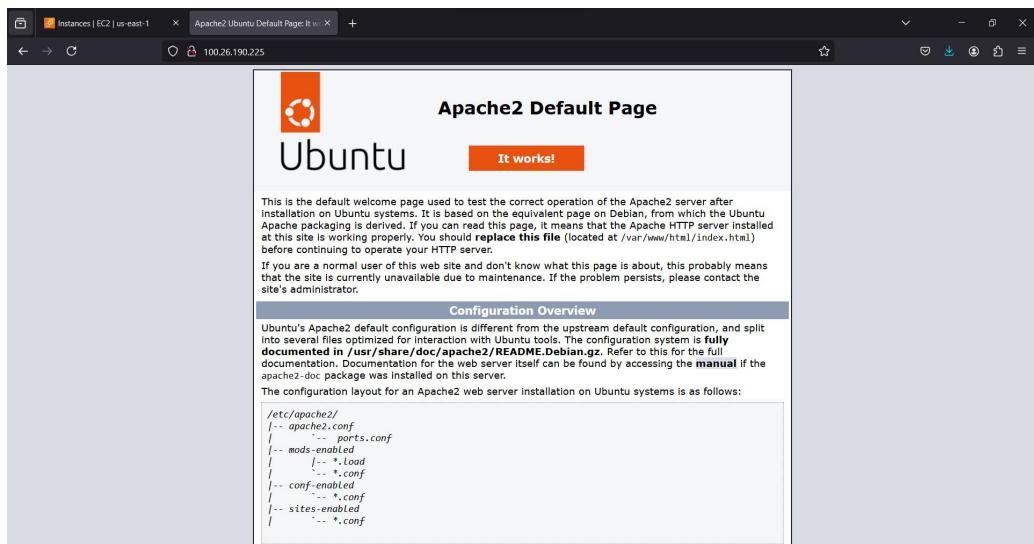
ubuntu@ip-172-31-93-109:~/playbooks$ ansible-playbook install-apache.yml -i /home/ubuntu/inventories/prod -v
Using /etc/ansible/ansible.cfg as config file

PLAY [Install HTTPD and ensure its enabled] ****
TASK [Gathering Facts] ****
ok: [node2]

TASK [Install httpd] ****
changed: [node2] => {"cache_update_time": 1724585740, "cache_updated": false, "changed": true, "stderr": "", "stderr_lines": [], "stdout": "Reading package lists...\nBuilding dependency tree...\nReading state information...\n\nThe following additional packages will be installed:\n  apache2-bin apache2-data apache2-utils libapr1t64\n  libaprutil1-db-sqlite3\n  libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert\nSuggested packages:\n  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser\nThe following NEW packages will be installed:\n  apache2 apache2-bin apache2-data apache2-utils libapr1t64\n  libaprutil1t64 liblua5.4-0 ssl-cert\n0 upgraded, 10 newly installed, 0 to remove and 86 not upgraded.\nNeed to get 2083 kB of archives.\nAfter this operation, 8094 kB of additional disk space will be used.\nGet:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libapr1t64 amd64 1.7.2-3.1build2 [107 kB]\nGet:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1t64 amd64 1.6.3-1.1ubuntu7 [91.9 kB]\nGet:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-db-sqlite3 amd64 1.6.3-1.1ubuntu7 [11.2 kB]\nGet:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-ldap amd64 1.6.3-1.1ubuntu7 [9116 B]\nGet:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblua5.4-0 amd64 5.4.6-3build2 [166 kB]\nGet:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-bin amd64 2.4.58-1ubuntu8.4 [1329 kB]\nGet:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-data all 2.4.58-1ubuntu8.4 [163 kB]\nGet:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-utils amd64 2.4.58-1ubuntu8.4 [97.1 kB]\nGet:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2 amd64 2.4.58-1ubuntu8.4 [90.2 kB]\nGet:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 ssl-cert all 1.1.2ubuntu1 [17.8 kB]\nPreconfiguring packages ...

```

- Access Apache on browser using node2's public IP,



Managing different types of OS servers,

- Launch instance using Amazon Linux AMI.

The screenshot shows the AWS Lambda console. In the top left, there's a 'Name and tags' section with a 'Info' link. Below it, a 'Name' input field contains 'node-3'. To the right is a 'Add additional tags' button. The main area has tabs for 'Recents' and 'Quick Start'. Under 'Quick Start', there are six cards: 'Amazon Linux' (selected), 'macOS', 'Ubuntu', 'Windows', 'Red Hat', and 'SUSE Li'. To the right of these is a search icon and a 'Browse more AMIs' link, which includes 'Including AMIs from AWS, Marketplace and the Community'. Below this is a section titled 'Amazon Machine Image (AMI)' with a card for 'Amazon Linux 2023 AMI'. The card details include: 'Free tier eligible', 'ami-066784287e358dad1 (64-bit (x86), uefi-preferred) / ami-023508951a94f0c71 (64-bit (Arm), uefi)', 'Virtualization: hvm', 'ENA enabled: true', and 'Root device type: ebs'.

The screenshot shows a terminal window with three tabs: 'MASTER', 'Node-3 Amazon Linux', and 'LOCAL'. The 'Node-3 Amazon Linux' tab is active, displaying the following command and output:

```
PS C:\Users\ojash\OneDrive\Desktop\Study\Keys> ssh -i "Ansible-key.pem" ec2-user@ec2-35-171-8-165.compute-1.amazonaws.com
```

The output continues with:

```
The authenticity of host 'ec2-35-171-8-165.compute-1.amazonaws.com (35.171.8.165)' can't be established.  
ED25519 key fingerprint is SHA256:S2ShVxPuva3fjs1CrzillIrUYGx9mS7HXmfeqEWgTHE.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'ec2-35-171-8-165.compute-1.amazonaws.com' (ED25519) to the list of known hosts.  
#_ _ _ _ _  
~~~ \_ #####\ Amazon Linux 2023  
~~~ \###|  
~~~ \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023  
~~~ V~'`-->  
~~~ /  
~~~ .-. /  
~~~ /- /  
~~~ /m/ |  
[ec2-user@ip-172-31-94-85 ~]$ |
```

- Now, create SSH key pair on master,
 - ssh-keygen -t rsa

```

MASTER                               Node-3 Amazon Linux      LOCAL
ubuntu@ip-172-31-93-109:~/ssh$ ls
authorized_keys  known_hosts  known_hosts.old
ubuntu@ip-172-31-93-109:~/ssh$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:/MLmiX1dYFy7bkFNsf9AEH9bqhVZOYOBnLYhhEY5w7Y  ubuntu@ip-172-31-93-109
The key's randomart image is:
+---[RSA 3072]----+
|   o+.. =oo.+|
|   O . * +.*o|
|   o + o.o==|=|
|   E   .+.o+=|
|   S   . o+o.|
|   .   . o+..|
|   +   . o+..|
|   =   o. . o|
|   .   +. . |
+---[SHA256]----+
ubuntu@ip-172-31-93-109:~/ssh$ |

```

- Now, copy Public key generated into node-3's authorized_key file,

```

MASTER                               Node-3 Amazon Linux      LOCAL
GNU nano 5.8                         authorized_keys          Modified
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCLHa2ZnJ2SqrqHXULRaUF2orI02JClOzsQz+sVylfp1nJS1njB969ymijV/b3FWTLG3ibBS4p7r>
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDgJY5aIZK5Pr4cF7xKDfkjhyhokYjCB/xWT+qVN/hisa/G/S377CS3NM5YdSC5uY+sGfLI6iPf1whs>

```

- Copy private key to /home/ubuntu/keys directory,
- Create new custom inventory for staging environment,
 - pwd : /home/ubuntu/inventories
 - nano stage

```

MASTER                               Node-3 Amazon Linux      LOCAL
GNU nano 7.2                         stage
[stage]
node-3 ansible_host=35.171.8.165

[stage:vars]
ansible_python_interpreter=/usr/bin/python3
ansible_user=ec2-user
ansible_ssh_private_key_file=/home/ubuntu/keys/id_rsa

```

- Trying to ping to node-3 which is Amazon Linux server,

```

MASTER                               Node-3 Amazon Linux      LOCAL
ubuntu@ip-172-31-93-109:~/inventories$ ansible -m ping stage -i stage
The authenticity of host '35.171.8.165 (35.171.8.165)' can't be established.
ED25519 key fingerprint is SHA256:S2ShVxPuva3fjs1CrzillIrUYGx9mS7HXmfeqEWgTHE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
node-3 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-93-109:~/inventories$ |

```

- Update /etc/ansible/hosts file to add fedora server,

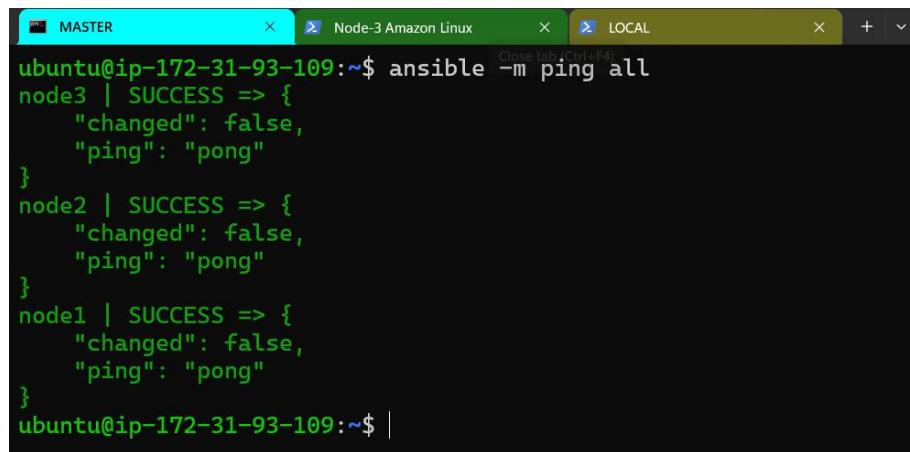
```
[ubuntusersvers]
node1 ansible_host=3.83.47.143
node2 ansible_host=100.26.190.225

[fedoraservers]
node3 ansible_host=35.171.8.165

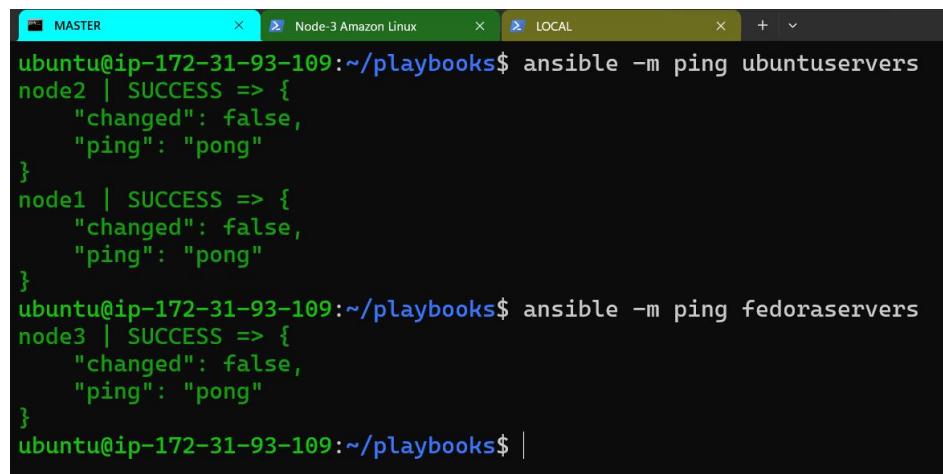
[fedoraservers:vars]
ansible_python_interpreter=/usr/bin/python3
ansible_user=ec2-user
ansible_ssh_private_key_file=/home/ubuntu/keys/id_rsa

[all:vars]
ansible_python_interpreter=/usr/bin/python3
ansible_user=ubuntu
ansible_ssh_private_key_file=/home/ubuntu/keys/Ansible-key.pem
```

- Try to ping to all server, Ubuntu and Fedora,



```
ubuntu@ip-172-31-93-109:~$ ansible -m ping all
node3 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
node2 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
node1 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-93-109:~$ |
```



```
ubuntu@ip-172-31-93-109:~/playbooks$ ansible -m ping ubuntusersvers
node2 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
node1 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-93-109:~/playbooks$ ansible -m ping fedoraservers
node3 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-93-109:~/playbooks$ |
```

Install Docker on only Ubuntu servers except Fedora server,

- Create new playbook to install docker,

- pwd : /home/ubuntu/playbooks
- nano install-docker.yml

```
GNU nano 7.2                               install-docker.yml

name: Install Docker and ensure enabled
hosts: all
become: yes
tasks:
- name: Install Docker
  apt:
    name: docker.io
    state: latest
    when: ansible_distribution == 'Debian' or ansible_distribution == 'Ubuntu'

- name: Start and Enable Docker
  service:
    name: docker.io
    state: started
    enabled: yes
    when: ansible_distribution == 'Debian' or ansible_distribution == 'Ubuntu'
```

- Check syntax of playbook,

- ansible-playbook --syntax-check install-docker.yml

```
ubuntu@ip-172-31-93-109:~/playbooks$ ansible-playbook --syntax-check install-docker.yml
playbook: install-docker.yml
ubuntu@ip-172-31-93-109:~/playbooks$ |
```

- Dry run playbook,

- ansible-playbook install-docker.yml -C

```
ubuntu@ip-172-31-93-109:~/playbooks$ ansible-playbook install-docker.yml -C
PLAY [Install Docker and ensure enabled] ****
TASK [Gathering Facts] ****
ok: [node3]
ok: [node2]
ok: [node1]

TASK [Install Docker] ****
skipping: [node3]
changed: [node2]
changed: [node1]
```

- It says, skipping : [node3] means, node3 is **Fedora** based server. We have added condition that install docker only if server is **Debain** or **Ubuntu** based.

```

ubuntu@ip-172-31-93-109:~/playbooks$ ansible-playbook install-docker.yml

PLAY [Install Docker and ensure enabled] ****
TASK [Gathering Facts] ****
ok: [node3]
ok: [node2]
ok: [node1]

TASK [Install Docker] ****
skipping: [node3]
changed: [node2]
changed: [node1]

TASK [Start and Enable Docker] ****
skipping: [node3]
ok: [node2]
ok: [node1]

PLAY RECAP ****
node1              : ok=3    changed=1      unreachable=0      failed=0      skipped=0      rescued=0      ignor
ed=0
node2              : ok=3    changed=1      unreachable=0      failed=0      skipped=0      rescued=0      ignor
ed=0
node3              : ok=1    changed=0      unreachable=0      failed=0      skipped=2      rescued=0      ignor
ed=0

```

- Docker is installed on Ubuntu servers only and skipped Fedora server(node3).

Roles in Ansible-

- Roles let you automatically load related vars, files, tasks, handlers, and other Ansible artifacts based on a known file structure. After you group your content into roles, you can easily reuse them and share them with other users.

- Create a new role,

- ansible-galaxy init nginx

```

ubuntu@ip-172-31-93-109:~$ ansible-galaxy init nginx
- Role nginx was created successfully
ubuntu@ip-172-31-93-109:~$ |

```

```

ubuntu@ip-172-31-93-109:~/nginx$ tree
.
├── README.md
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
├── templates
└── tests
    ├── inventory
    │   └── test.yml
    └── vars
        └── main.yml

8 directories, 8 files
ubuntu@ip-172-31-93-109:~/nginx$ |

```

- In /etc/ansible we have roles directory

```
MASTER Node-3 Amazon Linux LOCAL
ubuntu@ip-172-31-93-109:~$ cd /etc/ansible/
ubuntu@ip-172-31-93-109:/etc/ansible$ ls
ansible.cfg hosts roles
ubuntu@ip-172-31-93-109:/etc/ansible$ cd roles/
ubuntu@ip-172-31-93-109:/etc/ansible/roles$ |
```

- We need to move create role called nginx into /etc/ansible/roles folder.

```
MASTER Node-3 Amazon Linux LOCAL
ubuntu@ip-172-31-93-109:~$ sudo mv nginx /etc/ansible/roles/
ubuntu@ip-172-31-93-109:~$ ls /etc/ansible/roles/
nginx
ubuntu@ip-172-31-93-109:~$ |
```

- Go to /etc/ansible/roles/nginx

- To install Nginx as a task, need to edit file called nginx/tasks/main.yml

```
MASTER Node-3 Amazon Linux LOCAL
GNU nano 7.2 main.yml
---
# tasks file for nginx
- name: Install Nginx
  apt:
    name: nginx
    state: present

- name: Enable nginx
  service:
    name: nginx
    state: started
    enabled: yes

- name: Deploy webpage
  copy:
    src: index.html
    dest: /var/www/html/index.html
```

- Create file index.html in files directory and add dummy content,
 - Go to playbooks directory and create new playbook called install-nginx-via-role.yml.

```
MASTER Node-3 Amazon Linux LOCAL
GNU nano 7.2           install-nginx-via-role.yml

name: Install Nginx via Role
hosts: dev
become: yes
roles:
  - nginx
```

- Run the playbook,

```
ubuntu@ip-172-31-93-109:~/playbooks$ ansible-playbook install-nginx-via-role.yml -i /home/ubuntu/inventories/dev

PLAY [Install Nginx via Role] ****
TASK [Gathering Facts] ****
ok: [node1]

TASK [nginx : Install Nginx] ****
ok: [node1]

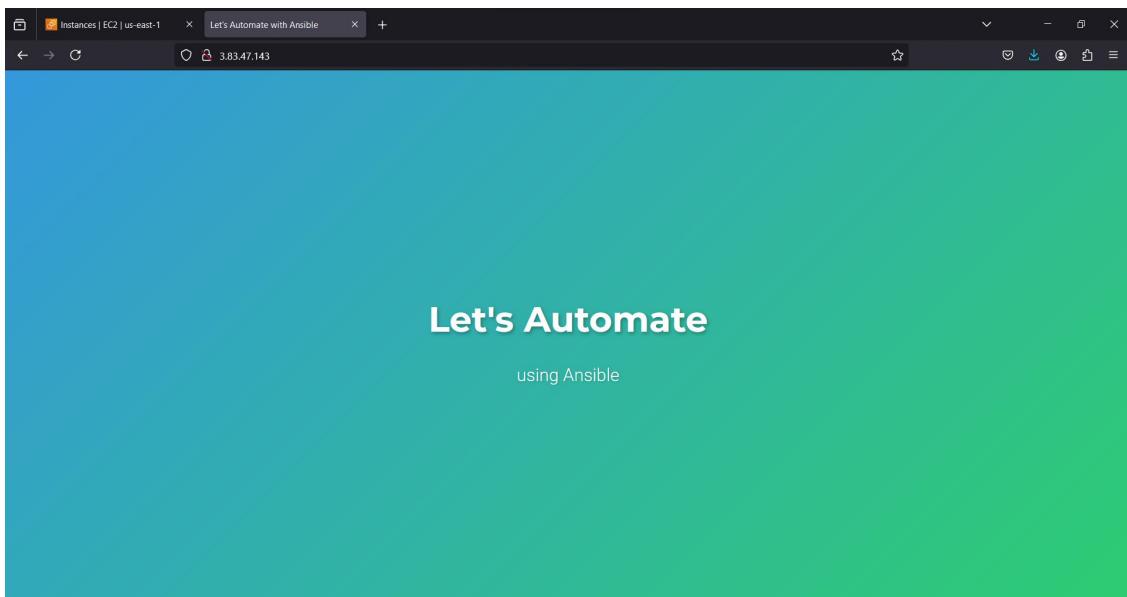
TASK [nginx : Enable nginx] ****
ok: [node1]

TASK [nginx : Deploy webpage] ****
changed: [node1]

PLAY RECAP ****
node1                  : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0

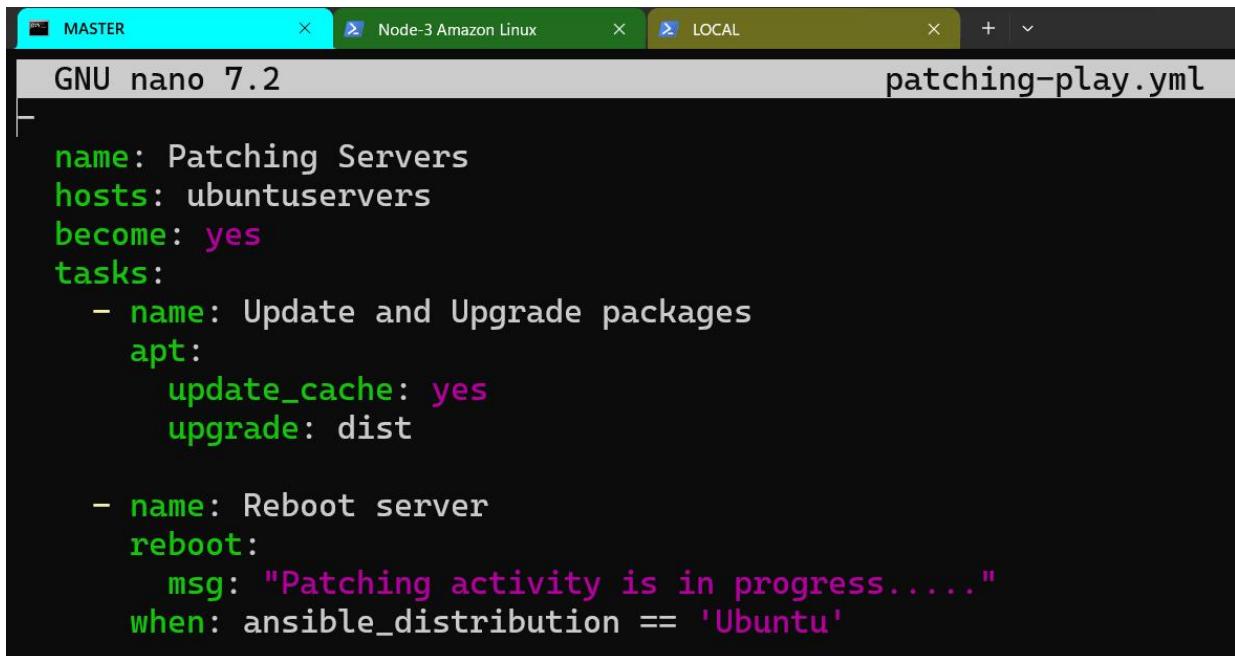
ubuntu@ip-172-31-93-109:~/playbooks$ |
```

- Go to browser and try to access nginx new webpage of node1,



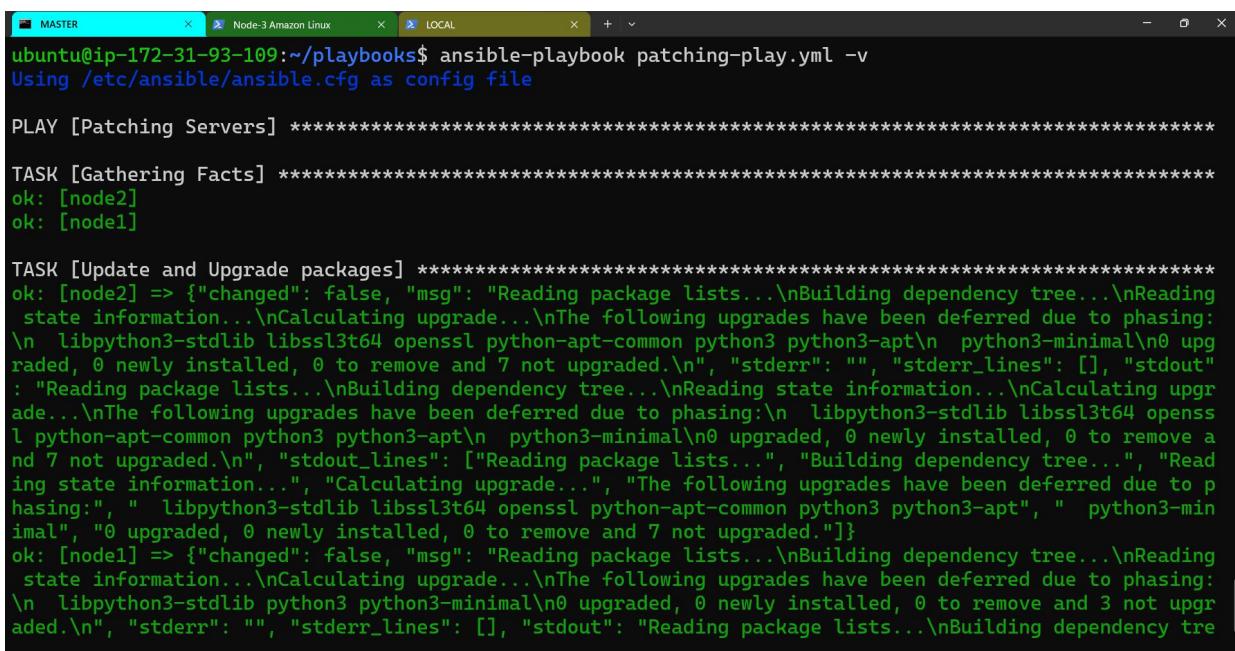
Patching of servers using Ansible playbook-

- Create new playbook called patching-play.yml



```
MASTER Node-3 Amazon Linux LOCAL
GNU nano 7.2 patching-play.yml
-
name: Patching Servers
hosts: ubuntuservers
become: yes
tasks:
- name: Update and Upgrade packages
  apt:
    update_cache: yes
    upgrade: dist

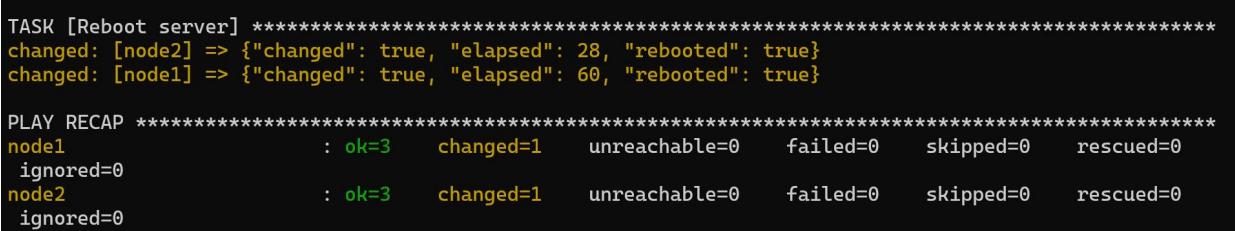
- name: Reboot server
  reboot:
    msg: "Patching activity is in progress....."
    when: ansible_distribution == 'Ubuntu'
```



```
ubuntu@ip-172-31-93-109:~/playbooks$ ansible-playbook patching-play.yml -v
Using /etc/ansible/ansible.cfg as config file

PLAY [Patching Servers] ****
TASK [Gathering Facts] ****
ok: [node2]
ok: [node1]

TASK [Update and Upgrade packages] ****
ok: [node2] => {"changed": false, "msg": "Reading package lists...\nBuilding dependency tree...\nReading state information...\nCalculating upgrade...\nThe following upgrades have been deferred due to phasing:\n\n libpython3-stdlib libssl3t64 openssl python-apt-common python3 python3-apt\n python3-minimal\n0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.\n", "stderr": "", "stderr_lines": [], "stdout": "Reading package lists...\nBuilding dependency tree...\nReading state information...\nCalculating upgrade...\nThe following upgrades have been deferred due to phasing:\n\n libpython3-stdlib libssl3t64 openssl python-apt-common python3 python3-apt", "stdout_lines": ["Reading package lists...", "Building dependency tree...", "Reading state information...", "Calculating upgrade...", "The following upgrades have been deferred due to phasing:", " libpython3-stdlib libssl3t64 openssl python-apt-common python3 python3-apt", " python3-minimal", "0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded."]}
ok: [node1] => {"changed": false, "msg": "Reading package lists...\nBuilding dependency tree...\nReading state information...\nCalculating upgrade...\nThe following upgrades have been deferred due to phasing:\n\n libpython3-stdlib python3 python3-minimal\n0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.\n", "stderr": "", "stderr_lines": [], "stdout": "Reading package lists...\nBuilding dependency tree..."}
```



```
TASK [Reboot server] ****
changed: [node2] => {"changed": true, "elapsed": 28, "rebooted": true}
changed: [node1] => {"changed": true, "elapsed": 60, "rebooted": true}

PLAY RECAP ****
node1 : ok=3    changed=1    unreachable=0    failed=0     skipped=0    rescued=0
      ignored=0
node2 : ok=3    changed=1    unreachable=0    failed=0     skipped=0    rescued=0
      ignored=0
```