

# Ansible

## Introduction to Ansible

Ansible is an open-source IT automation tool used to automate software provisioning, configuration management, application deployment, and task automation. It simplifies tasks like provisioning servers, configuring systems, and managing configurations across infrastructure.

## History of Ansible

Ansible was created by Michael DeHaan in 2012. It was developed as a simple, agentless configuration management tool with an emphasis on simplicity and ease of use. Ansible was later acquired by Red Hat in 2015. Since then, it has gained significant popularity in DevOps environments.

## Advantages of Configuration Management Tools

Configuration Management (CM) tools like Ansible provide the following advantages:

- **Consistency:** Ensure configurations are the same across environments.
- **Automation:** Automates repetitive tasks like software installation, updates, and system configurations.
- **Scalability:** CM tools scale across hundreds or thousands of machines with minimal human intervention.
- **Auditing:** CM tools can track changes, making auditing easier.
- **Error reduction:** Minimize manual errors in system configurations and application setups.

## Why Ansible?

- **Agentless:** Ansible does not require agents to be installed on target machines. It uses SSH or WinRM for communication.
- **Declarative Syntax:** Users declare what they want to happen, not how it should happen.
- **Simple and Easy:** Ansible uses YAML, a human-readable language, making it simple for users to write automation scripts.
- Idempotence is a technical term that means a function or operation can be repeated without changing the result.
- **Idempotency:** Ansible ensures that tasks are executed only when necessary, avoiding redundant operations.
- **Extensible:** Ansible provides an easy mechanism for integrating with other tools and creating custom modules.

## Ansible Advantages

- **Ease of Use:** Simple, clear syntax in YAML.
- **No Need for Agents:** Direct communication via SSH.
- **Cross-platform:** Works on Linux, Windows, and macOS.

- **Extensible:** Ansible can be extended with custom modules.
  - **Declarative:** You define the desired state, not the steps to reach that state.
  - **Powerful Features:** Support for roles, playbooks, and dynamic inventories.
- 

## Install & Configure Ansible

1. **Install Ansible:**
  - On Linux: `sudo apt-get install ansible` (Ubuntu/Debian)
  - On RedHat/CentOS: `sudo yum install ansible`
  - On macOS: `brew install ansible`
2. **Configure Ansible:**
  - Edit the `ansible.cfg` file to configure settings like default inventory file, SSH settings, etc.

## Use Cases of Ansible

- **Infrastructure Provisioning:** Automate server provisioning across cloud and on-premise environments.
  - **Configuration Management:** Ensure consistent configuration across machines.
  - **Application Deployment:** Automate the deployment of software applications across many servers.
  - **Continuous Integration/Continuous Deployment (CI/CD):** Automate workflows for deploying and testing applications.
- 

## What Can Ansible Do in a Production Environment?

- **Provision Servers:** Create and configure new machines in your infrastructure.
- **Automate Software Deployment:** Install applications like web servers, databases, etc.
- **Manage Configurations:** Ensure system configurations are correct and consistent.
- **Orchestrate Tasks:** Automate complex multi-machine tasks, including load balancing, service discovery, etc.

## Ansible Documentation

Ansible provides detailed documentation on its website, including guides on installation, modules, playbooks, and advanced usage. It's a comprehensive resource for troubleshooting and understanding concepts.

<https://docs.ansible.com/>

---

## How Ansible is Different from Configuration Management Tools

- **Agentless:** Unlike tools like Chef and Puppet, Ansible does not require agents installed on managed nodes. It uses SSH to communicate directly with the nodes.
- **Simplicity:** Ansible has a simpler syntax (YAML) compared to the domain-specific languages used by other CM tools.
- **Declarative Approach:** Ansible allows you to declare the desired state without defining step-by-step procedures.

## Ansible Architecture

- **Control Node:** The machine from which Ansible commands are run.
  - **Managed Nodes:** The systems being managed.
  - **Modules:** Reusable pieces of code used to carry out tasks (e.g., installing a package).
  - **Inventory:** A list of all systems to be managed.
- 

## Ansible Control Machine Requirements

- **Operating System:** Linux, macOS, or Windows (via WSL).
- **Dependencies:** Python (on the control node), SSH (for communication with managed nodes).

## Ansible Installation Process

- Install Ansible via package managers (**apt**, **yum**, **brew**) or from source.
  - Ensure SSH access to target machines.
- 

## Ansible Terminologies

- **Playbook:** A YAML file containing a series of tasks to be executed on managed nodes.
  - **Inventory:** A file that lists managed nodes and their attributes.
  - **Module:** A single unit of work (e.g., installing a package, configuring a service).
  - **Task:** A single action in a playbook.
  - **Role:** A reusable set of playbooks and tasks for managing a specific functionality.
- 

## How Ansible Works

1. **Playbook Execution:** Ansible executes a series of tasks defined in a playbook.
2. **Communication:** Ansible communicates with managed nodes via SSH or WinRM.
3. **Modules:** Tasks are carried out by running Ansible modules on the nodes.
4. **Idempotency:** Ansible only executes tasks when necessary, ensuring consistency across environments.

## Ansible Lab Setup

To set up an Ansible lab, you need:

- A control node where Ansible is installed.
  - Managed nodes that can be accessed over SSH.
  - A proper inventory file listing all managed nodes.
- 

## Ansible Inventory

Inventory files can be static (written manually) or dynamic (generated by scripts or cloud APIs).

- **Group Inventory File:** A file grouping hosts for easy management.
- **Inventory Parameters:** Define settings like user, port, etc.

## Test Environment Setup

To test an Ansible setup:

1. Define a simple inventory file with a few test hosts.
2. Run the `ansible all -m ping` command to ensure connectivity.

## Host Patterns

Host patterns define which hosts in the inventory should be targeted by a playbook or command.

Examples:

- `all`: All hosts in the inventory.
  - `group1`: All hosts in the `group1` group.
  - `host1`: A specific host named `host1`.
- 

## Ad-Hoc Commands

Ansible ad-hoc commands allow you to run simple commands directly from the CLI without writing a full playbook. For example:

- `ansible all -m ping`: Ping all hosts in the inventory.
- `ansible web -m apt -a "name=nginx state=latest"`: Install or update nginx on the `web` group.

## Modules

Ansible modules are the smallest unit of work. Examples:

- `apt`: Package management module for Debian-based systems.
- `yum`: Package management module for RedHat-based systems.

- **service**: Manage system services.

## Gathering Facts

Ansible gathers system information (facts) about the target machine, which can be used in playbooks (e.g., **ansible\_facts**).

---

## Playbooks

A Playbook is a file containing one or more plays, written in YAML. Each play runs on specific hosts and executes a series of tasks.

### Sample Playbook:

```
- name: Install nginx

hosts: web servers

become: yes

tasks:
  - name: Install nginx
    apt:
      name: nginx
      state: present
```

- **Playbook Format:**
  - **name**: Describes what the playbook does.
  - **hosts**: Defines which hosts this playbook applies to.
  - **tasks**: A list of tasks to run on the hosts.

## How to Run Playbooks

Run playbooks with the following command:

- **ansible-playbook** `playbook.yml`

## Check Syntax of a Playbook

To check the syntax without running the playbook:

- **ansible-playbook** `--syntax-check` `playbook.yml`

## Run Playbook on Multiple Hosts

Specify multiple hosts using a comma:

- `ansible-playbook -i inventory playbook.yml`
- 

## Roles

Ansible roles are a way to organize playbooks. They contain files, tasks, templates, and other components to modularize the playbook. Example structure:

```
roles/  
  common/  
    tasks/  
      main.yml  
    templates/  
      config.j2
```

To use a role:

```
- hosts: web  
  
  roles:  
    - common
```

---

## Install Tomcat and Jenkins using Ansible

1. **Install Tomcat:**
  - Define a playbook with tasks to install Java, download Tomcat, and configure it.
2. **Install Jenkins:**
  - Define tasks to install Java and Jenkins using the package manager.

## Practicals

### lab setup

launch 3 instance

1. install ansible on master
2. configure ssh access to ansible host
3. setting up ansible host and testing connection

one is master and both are client

launch 3 ubuntu server with normal configuration

**Instances (1/3)** Info Last updated less than a minute ago Connect Instance state Actions Launch instances

Find Instance by attribute or tag (case-sensitive) All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input type="checkbox"/>	master	i-0513347c90a0fb528	Running	t2.micro	Initializing	<a href="#">View alarms +</a>	us-east-2b
<input type="checkbox"/>	slave1	i-0212538490fb5f3fa	Running	t2.micro	Initializing	<a href="#">View alarms +</a>	us-east-2b
<input checked="" type="checkbox"/>	slave2	i-07940b191a63cab9d	Running	t2.micro	Initializing	<a href="#">View alarms +</a>	us-east-2b

create a sg with all traffic allow in it and connect it to your server

**Inbound rules (3)** Manage tags Edit inbound rules

Search

	Name	Security group rule ID	IP version	Type	Protocol
<input type="checkbox"/>	-	sgr-02892dae4176e6e5c	IPv4	SSH	TCP
<input type="checkbox"/>	-	sgr-02bac543a514e7db9	IPv4	All traffic	All
<input type="checkbox"/>	-	sgr-03bd0f5c04f6cb3e2	IPv4	HTTP	TCP

one client and one master

>connect your master server

```
Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-31-209:~$

i-0513347c90a0fb528 (master)
```

>sudo apt update && sudo apt upgrade

```
ubuntu@ip-172-31-31-209:~$ sudo apt update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
```

>sudo apt install software-properties-common

```
ubuntu@ip-172-31-31-209:~$ sudo apt install software-properties-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
software-properties-common is already the newest version (0.99.49.1).
software-properties-common set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-31-209:~$ sudo apt-add-repository ppa:ansible/ansible
Repository: 'Types: deb
URIs: https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/
```

> sudo add-apt-repository --yes --update ppa:ansible/ansible

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-30-40:~$ sudo add-apt-repository --yes --update ppa:ansible/ansible
Repository: 'Types: deb
URIs: https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/
Suites: noble
Components: main
'
```

>sudo apt install ansible -y

```
ubuntu@ip-172-31-31-209:~$ 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-kerberos python3-nacl python3-ntlm-auth python3-py
python3-xmltodict sshpass
```

>ansible --version

```
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-19-45:~$ ansible --version
ansible [core 2.16.3]
  config file = None
  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, Nov 6 2024, 18:32:19) [GCC 13.2.0] (/usr/bin/python3)
  jinja version = 3.1.2
```

2. step

ssh access

now when you try to ssh your slave it will not happen

>cd .ssh >ls >ssh-keygen

```
ubuntu@ip-172-31-38-32:~$ cd .ssh
ubuntu@ip-172-31-38-32:~/.ssh$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_ed25519
Your public key has been saved in /home/ubuntu/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:JceUxsd0LdHo/cNxlpbZlGsiSmqAjB5ugy2m8u16Yk8 ubuntu@ip-172-31-38-32
The key's randomart image is:
+--[ED25519 256]--+

```

>ls



>cat id\_rsa.pub

```
ubuntu@ip-172-31-31-209:~/.ssh$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIApR6Y8mVpHc+V4S4epUhwti2VnaNObt0BcM9W20dKRI ubuntu@ip-172-31-31-209
ubuntu@ip-172-31-31-209:~/.ssh$
```

on the both node server

cd .ssh copy that ssh.pub file here

sudo vi auth

```
ubuntu@ip-172-31-37-108:~$ cd .ssh
ubuntu@ip-172-31-37-108:~/.ssh$ sudo vi authorized_keys
ubuntu@ip-172-31-37-108:~/.ssh$
```

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQKD5eIcbPFKFC63fEwwOgLaO+9/IqBFILHQKTvni0JXGFLIQUbuY721NwDHFMcYur5YiPhYQVkpEHHih+saGwyr1EtqrRk+BDhAluyJn4kC/z47/lmz
vF886KfMDQwE20h0th58sMSxttoiweCYGmjRucS2Mllhgc7Ocfh0qRbFiUrP8dFGdKLgdmxwk5Hs6sPJowcwUmGv1/ZbVSANjdeuMbaQ6K47OX4Wy1I4mfcSfShLXQ6C8BWLARM5OH9GYtDDRv224W
pgg88g3GbechtIAIM01q0wNMf+6Hd4i/gJb7b36qmKME/r0PCaAPD2FgIPBE3KEkEjullm ansible
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOXeMO2Kp0LKncRQ37JbH3ISx11QEKRMUAmMaBhVaM ubuntu@ip-172-31-38-82
```

copy that file and paste it inside authorized\_keys

now on the slave

>sudo apt update && sudo apt upgrade -y

>sudo apt-get install python3

```
ubuntu@ip-172-31-19-45:~$ sudo apt upgrade && sudo apt install python -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
#
```

```
ubuntu@ip-172-31-37-108:~/.ssh$ sudo apt-get install python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.12.3-0ubuntu2).
python3 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
ubuntu@ip-172-31-37-108:~/.ssh$
```

create a .pem file of your node server and add it in the control panel

>sudo vi demo.pem

pasted your key here from the window

>sudo chmod 400 demo.pem

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOXeMOZKpQlKnrcHQ37JbH3ISx
ubuntu@ip-172-31-38-32:~/.ssh$ cd
ubuntu@ip-172-31-38-32:~$ sudo vi demo.pem
ubuntu@ip-172-31-38-32:~$ sudo chmod 400 demo.pem
ubuntu@ip-172-31-38-32:~$
```

now try to do ssh

>ssh ubuntu@ip

```
ubuntu@ip-172-31-38-32:~$ ssh ubuntu@16.171.31.125
The authenticity of host '16.171.31.125 (16.171.31.125)' can't be established.
ED25519 key fingerprint is SHA256:DPjV7hILX853twEGSQu3PplBULs9RYjQCFke6/gJUaE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '16.171.31.125' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

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

to test your ansible

>ansible localhost -m ping

```
ubuntu@ip-172-31-19-45:~$ ansible localhost -m ping
[WARNING]: No inventory was parsed, only implicit localhost is available
localhost | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

>sudo chmod 766 ansible.cfg

now if you read its config file you won't see much for that

>sudo ansible-config init --disable -t all > ansible.cfg

```
ubuntu@ip-172-31-38-32:/etc/ansible$ sudo chmod 766 ansible.cfg
ubuntu@ip-172-31-38-32:/etc/ansible$ sudo ansible-config init --disable -t all > ansible.cfg
ubuntu@ip-172-31-38-32:/etc/ansible$ ls
ansible.cfg  hosts  roles
ubuntu@ip-172-31-38-32:/etc/ansible$
```

## create user and group

>vi /etc/ansible/host

add your slave ip here

[prod]

mantu ansible\_ssh\_host=51.20.85.255 ansible\_user=ubuntu

ansible\_ssh\_private\_key\_file=/home/ubuntu/demo.pem

[test]

chantu ansible\_ssh\_host=16.171.31.125 ansible\_user=ubuntu  
ansible\_ssh\_private\_key\_file=/home/ubuntu/demo.pem

```
[prod]
mantu ansible_ssh_host=51.20.85.255 ansible_user=ubuntu ansible_ssh_private_key_file=/home/ubuntu/demo.pem

[test]
chantu ansible_ssh_host=16.171.31.125 ansible_user=ubuntu ansible_ssh_private_key_file=/home/ubuntu/demo.pem

## [webservers]
```

save it

>ansible-inventory -list

```
ubuntu@ip-172-31-38-32:~$ ansible-inventory --list
{
  "_meta": {
    "hostvars": {
      "chantu": {
        "ansible_ssh_host": "16.171.31.125",
        "ansible_ssh_private_key_file": "/home/ubuntu/demo.pem",
        "ansible_user": "ubuntu"
      },
      "mantu": {
        "ansible_ssh_host": "51.20.85.255",
        "ansible_ssh_private_key_file": "/home/ubuntu/demo.pem",
        "ansible_user": "ubuntu"
      }
    }
  },
  "all": {
    "children": [
      "ungrouped",

```

>sudo chown ubuntu:ubuntu /home/ubuntu/demo.pem

> chmod 400 /home/ubuntu/demo.pem

>ansible -m ping all

```
ubuntu@ip-172-31-30-40:/etc/ansible$ ansible -m ping all
[WARNING]: Platform linux on host mantu is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for
information.
mantu | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "changed": false,
  "ping": "pong"
}
[WARNING]: Platform linux on host demo is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for
information.
demo | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "changed": false,
  "ping": "pong"
}
```

if its success then its mean its done

>ansible -m ping production (group name )

>ansible mantu -m ping

```

ubuntu@ip-172-31-38-32:~$ ansible prod -m ping
[WARNING]: Platform linux on host mantu is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
mantu | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu@ip-172-31-38-32:~$ ansible mantu -m ping
[WARNING]: Platform linux on host mantu is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
mantu | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu@ip-172-31-38-32:~$

```

>ansible -m ping slave1 (server name )

```

ubuntu@ip-172-31-30-40:/etc/ansible$ ansible -m ping demo
[WARNING]: Platform linux on host demo is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
demo | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu@ip-172-31-30-40:/etc/ansible$

```

if you dont have anyone so

```

ubuntu@ip-172-31-30-40:/etc/ansible$ ansible -m ping he
[WARNING]: Could not match supplied host pattern, ignoring: he
[WARNING]: No hosts matched, nothing to do
ubuntu@ip-172-31-30-40:/etc/ansible$

```

## lets create a file for ping only

>mkdir playbooks

>cd playbooks

ansible-playbook --syntax-check ping.yml

```

ubuntu@ip-172-31-38-32:/etc/ansible/playbook$ ansible-playbook --syntax-check ping.yml
playbook: ping.yml
ubuntu@ip-172-31-38-32:/etc/ansible/playbook$

```

## Ansible ad-hoc

>ansible localhost -m ping

>ansible all -m ping

To ping all the server that are add in hosts file

>ansible 10.0.0.1 -m ping

> ansible all -m copy -a "src=/home/ubuntu/demo.pem dest=/tmp/"

```
ubuntu@ip-172-31-38-32:~$ ansible all -m copy -a "src=/home/ubuntu/demo.pem dest=/tmp/"
[WARNING]: Platform linux on host mantu is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of this interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
mantu | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "changed": true,
  "checksum": "8714aedd7a85e8ed595a11e24cee75d334c0329",
  "dest": "/tmp/demo.pem",
  "gid": 1000,
  "group": "ubuntu",
  "md5sum": "59faaf1041e121c06991c365bd57c917",
  "mode": "0664",
  "owner": "ubuntu",
  "size": 1679,
  "src": "/home/ubuntu/.ansible/tmp/ansible-tmp-1738668599.6884708-5333-268534181142919/.source.pem",
  "state": "file",
  "uid": 1000
}
```

>ansible all -m copy -a "src=/home/ubuntu/demo.pem dest=/tmp/" -b --ask-become-pass

```
ubuntu@ip-172-31-38-32:~$ ansible all -m copy -a "src=/home/ubuntu/demo.pem dest=/tmp/" -b --ask-become-pass
BECOME password:
[WARNING]: Platform linux on host chantu is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of this interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
chantu | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "changed": false,
  "checksum": "8714aedd7a85e8ed595a11e24cee75d334c0329",
  "dest": "/tmp/demo.pem",
  "gid": 1000,
  "group": "ubuntu",
  "md5sum": "59faaf1041e121c06991c365bd57c917",
  "mode": "0664",
  "owner": "ubuntu",
  "size": 1679,
  "src": "/home/ubuntu/.ansible/tmp/ansible-tmp-1738668599.6884708-5333-268534181142919/.source.pem",
  "state": "file",
  "uid": 1000
}
```

It will ask you password

-b become true

-m for command

-a arguments

>ansible all -m service -a "name=nginx state=reloaded" --become

```
ubuntu@ip-172-31-38-32:~$ ansible all -m service -a "name=nginx state=reloaded" --become
[WARNING]: Platform linux on host mantu is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of this interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
mantu | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "changed": true,
  "command": "systemctl reload nginx",
  "rc": 0
}
```

ansible all -m apt -a "name=nginx state=present"

```
ubuntu@ip-172-31-38-32:~$ ansible all -m apt -a "name=nginx state=present"
[WARNING]: Platform linux on host mantu is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of this interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
mantu | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "cache_update_time": 1738651300,
  "cache_updated": false,
  "changed": false,
  "name": "nginx",
  "state": "present"
}
```

## Role

## Roles

It helps to structure a way of grouping together various functionalities and making it easier to reuse and share common setup task.

Main file it call the roles

>cd ..

>cd roles

>sudo ansible-galaxy init httpd\_setup

>it will create the whole dir with different files

>ls cd /roles/httpd\_setup

```
[root@localhost ansible]# cd roles/
[root@localhost roles]# ls
[root@localhost roles]# ansible-galaxy init httpd_setup
- Role httpd_setup was created successfully
[root@localhost roles]# ls
httpd_setup
[root@localhost roles]# S
```

>cd task

>vi main.yml

---

- name: Install Apache (httpd)

apt:

name: nginx

state: present

when: ansible\_os\_family == "Debian"

- name: Start and enable Apache service

systemd:

name: nginx

state: started

enabled: yes

to run your file or role

>cd playbook

>sudo vi inventory.ini

[webservers]

mantu ansible\_host=16.171.31.125 ansible\_user=ubuntu

ansible\_ssh\_private\_key\_file=/home/ubuntu/demo.pem

>sudo vi role.yml

---

- name: Install and configure Apache using Ansible Role
- hosts: webserver
- become: yes # Run as sudo
- roles:
  - httpd\_setup

to run your ansible role

> ansible-playbook -i inventory.ini ans.yml

```
ans.yml  create.yml  inventory.ini  jenkins.yml  test.yml
ubuntu@ip-172-31-38-32:/etc/ansible/playbook$ ansible-playbook -i inventory.ini ans.yml

PLAY [Install and configure Apache using Ansible Role] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host mantu is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for mo
information.
ok: [mantu]

TASK [httpd_setup : Install Apache (httpd)] *****
ok: [mantu]

TASK [httpd_setup : Start and enable Apache service] *****
ok: [mantu]

PLAY RECAP *****
i-020a0c80b11c39b32 (Control_panel)
```

and you are done!!!