Ansible

* **What is Ansible**?

Ansible is an open source automation tool or platform used for IT task such as configuration management, application deployment, intraservice orchestration, and provisioning. Ansible uses a simple, human-readable syntax called YAML (YAML Language) to describe automation tasks. Ansible is widely used in IT operations, system administration, and DevOps practices to automate routine tasks, manage infrastructure, and streamline the deployment of applications across large-scale environments.

Ansible is a free and open source automation that allow system administrators to configure and control hundreds of node form a central server ***without the need installing any agents on the node.***

It relies in **SSH** protocol to communicate with the remote nodes. Compared to other management tool such as puppet and chef, Ansible come out as the favourite due to it ease of use, and installation.

If you take other configuration management tool like puppet, chef , and CFEngine , server software is installed on one machine , and client machine are managed through the agent. *Wherein Ansible are node are manged by controlling machine (Ansible Server) over SSH, so there won’t be any agent running on node machines.*

* **Benefits of Ansible**:

1. Free: Ansible is an open-source tool. Very simple to set up and use: No special coding skills are necessary to use Ansible’s playbooks (more on playbooks later).
2. Powerful: Ansible lets you model even highly complex IT workflows.
3. Flexible: You can orchestrate the entire application environment no matter where it’s deployed. You can also customize it based on your needs.
4. Agentless: You don’t need to install any other software or firewall ports on the client systems you want to automate. You also don’t have to set up a separate management structure.

**Note: Ansible is also agent oriented configuration tool, but it uses SSH protocol**

**For communication with client machine.**

**EX: we do have agent base configuration tool such as puppet chef.**

1. Efficient: Because you don’t need to install any extra software, there’s more room for application resources on your server.

* **Configuration Management with Ansible**:
* Ansible is designed very simple, reliable, and consistent for configuration management.
* If you are already working IT you can get up running quickly.
* Ansible configuration simple data description of infrastructure which can be easily readable by humans and parcable by machine.
* **Application Deployment**: Ansible let you quickly and easily deploy with multi-tier apps you won’t read write to custum code to automate your system.
* **security & compliance**: If you configure to security detail on the control machine and run they associate playbook all the remote host will automatically, be updated those details. That means you won't need to monitor each machine for security compliance continually manually.
* **cloud provisioning**: To automate your application you should automate your infrastructure provisioning with ansible you can provisioning cloud platforms virtualize host, Network devices & mental server hosts.
* **Components Of Ansible**:

1. **Task**:

Any actual we to for slave node is called task.

1. **Inventory file**:

It is an file store on controller node which contain details like worker node information, Ip, hostname, username, password

1. **how to see Inventory file**:

Ansible –version

* **Setting the lab for Ansible:**

1. Create three Linux instances one master and other slaves.
2. Install ansible on master instance only not on slave instances.
3. This command should in Master.

sudo apt-add-repository ppa:ansible/ansible

sudo apt-get update

sudo apt-get install ansible

ansible –version

sudo vi /etc/ansible/hosts

(Edit the ansible hosts file to add the Ip addresses of the users and groups to the hosts configuration file.)

1. On slave instances edit the sshd configuration file and Permitrootlogin to yes and password authentication as yes.

**vi /etc/ssh/sshd\_config**

**systemctl restart sshd**

1. . Also set root password for slave instances.
2. On master instance run the following command and connect to the slave instance from master instance.

ssh root@(pub IP add. of slave inst)

* **List**:
* To list a specific host of an Ip address

**ansible (IP address) list-host**

* To list out all hosts on master server

**ansible all --list-host**

* To list out group of host on master = **ansible (group name) --list-host**

We can create a inventory file for ansible containing users or groups with IP address and list them on ansible using:

**ansible all --list-hosts -i (inventoryfilename).cfg**

If you dont want to use default inventory file you can create you own inventory file and use it.

If you want to list our hosts under your inventory file you have to run above command

* **Ansible Module**:

A module is a reusable standalone script that ansible runs on your behalf either locally or remotely, modules interact with your local machines on API or a remote system to perform specific tasks like changing a database password, changing a configuration file or spinning of a cloud instance.

Basic Ansible Module:

1. Command Raw Modules:

With the help of command raw module we can run or perform any Linux command on slave machine. Syntax for running a module:

**ansible [group name/IP] (option) (modulename) (argument)**

eg. ansible web -m command -a "uptime"

eg. ansible web -m command "date' –k

(use option -k for using the password based connection with slave machines)

(argument option stands for any command that we want to run on all slaves at the same time)

(we can use any Linux command like date, Cal, install, remote. Etc as an argument)

here web is target slaves group name which we have defined under /etc/ansible/hosts file. -**m stands for module**,

-**a stands for argument**

To disable host key checking under inventory file make an entry like below

**host\_key\_checking=false**

You can use command module for single command only, to use multiple commands we need to go for raw module

eg. ansible web -m raw -a "ls;date;pwd"

**Passwordless authetication**:

Generate a key for slave machines: ssh-keygen

ssh-copy-id root @private.Ip (Try to connect without password)

ssh root@prv.ip

1. **Ansible copy module**:

The copy module copies a file from the local or remote machine to a location on a remote machine.

Whenever we want to copy a file from master node to slave node we need to use copy module.

eg. **ansible web -m copy -a 'src=/etc/passwd dest=/tmp"**

note: whenever you run ansible module it will provide output and output will have different colours those are yellow green red.

Yellow- It means whatever ansible command or ansible playbook you have run got executed successfully and it changed some configuration in slave nodes.

Green- It means your ansible command or file executed successfully but no changes happened ( It means those changes were already present on the slave.

Red- It means there is an error occurred while running ansible command or file.

**Copy Module Options**:

If a file is already present in worker (slave node), you want to keep that file as well the n set backup=yes

**ansible web -m copy -a "src = /etc/passwd dest=/tmp owner=rahul group=aws mode=777 backup=yes" -k --(will keep older file as back up )**

In above both scenario file was present in controller and we were copying it in worker. But now if file is present in worker only and we want to copy it in another location of worker only then

**ansible web -m copy -a "src = /etc/passwd remote\_src=yes dest=/tmp owner=rahul group=aws mode=777 " -k**

To write any content inside a file

**ansible web -m copy -a 'content="welcome to Ansible " dest=/tmp/newfile.txt'**

1. Fetch module:

In Ansible, the "fetch" module is used to retrieve files from remote hosts and copy them to the local machine. This module is essentially the opposite of the "copy" module, which copies files from the local machine to remote hosts.

In this example:

src: Specifies the path to the file on the remote host that you want to fetch.

dest: Specifies the destination path on the local machine where the file will be copied.

We use fetch module to copy files from worker node to controller node.

Example:

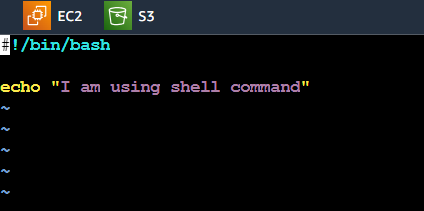
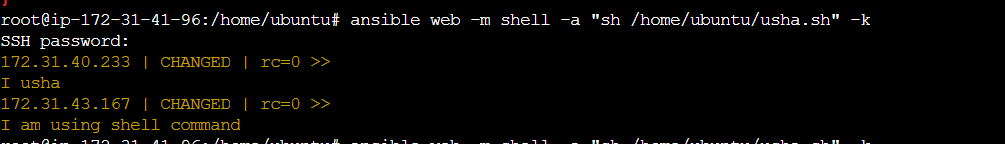
First Create file in any node then use command :

***ansible web -m fetch -a 'src=home/ubuntu/172.31.43.167/tmp dest=/home/ubuntu/'***

1. Shell command:

In the context of Ansible, a "shell module" refers to the shell module that Ansible provides. Ansible is an open-source automation tool that allows you to define and manage infrastructure as code. The shell module in Ansible allows you to execute shell commands on remote hosts.

Fist Slave create file extension with **.sh** and then run command in Master.

1. File Module:

The file module in Ansible is used to manage files and directories on remote hosts.

We use this module to create update or delete a file or directory.

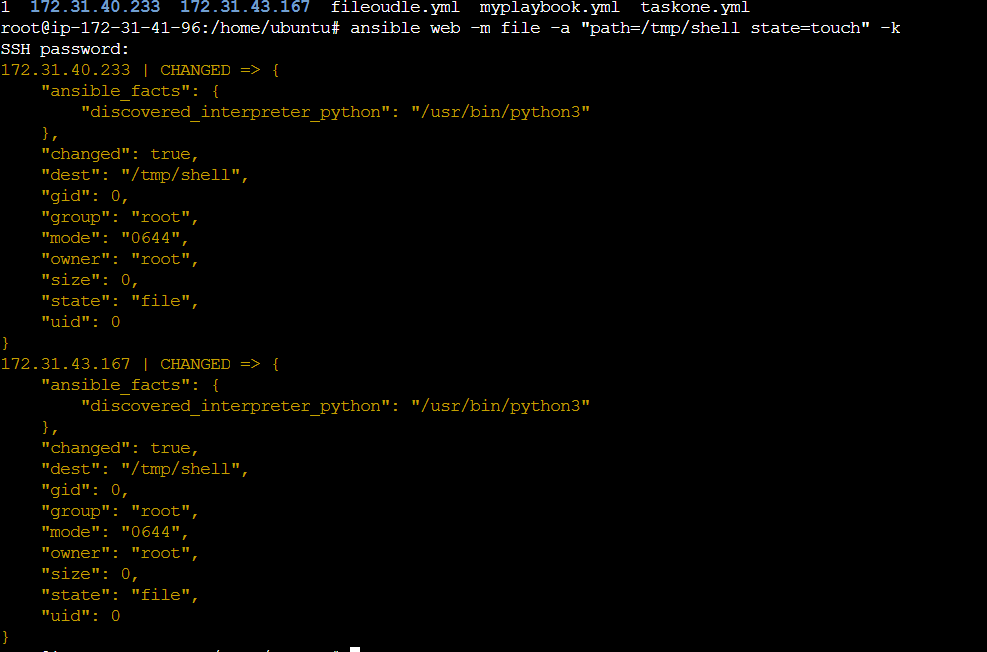
Ex:

***ansible web -m file -a "path=/home/Ubuntu/India state=directory" –k***



File module using create file :

***ansible web -m file -a "path=/home/Ubuntu/India state=touch " –k***



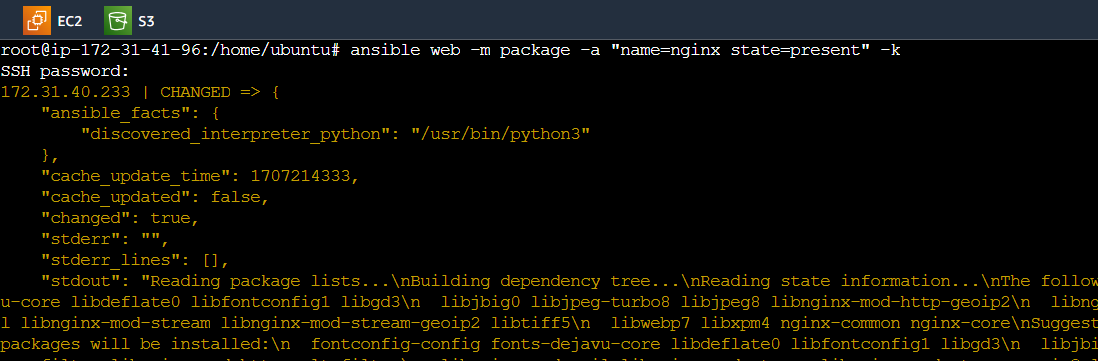
Check the node in path: check folder comes under the file.

1. Package Module:

Package module is a combination of all repositories which we use to install and remove a software, if you are looking for specific yum or apt module those are also available.

Present= install Package

Absent = uninstall package



1. Service Module:

Service module is useful to control services on remote machines i.e slaves

**ansible web -m service -a "name=nginx state=started/stopped”**

1. User and Group Module:

We use this module for creating new users and groups on remote machines (slaves).

command= (user)

ansible all -m user -a "name=wipro state=present" –k

check the slave machine use the command *cat /etc/passwd*

command= (group)

ansible all -m group -a "name=wipro state=present" –k

check the slave machine use the command *cat /etc/gshadow*

1. Line in file module :

To add a line in file

ansible all -m lineinfile -a "path=/etc/sudoers line='rahul ALL=(ALL)ALL'"

To delete a line in file

ansible all -m lineinfile -a "path=/etc/sudoers line='rahul ALL=(ALL)ALL' state=absent "

ansible all -m lineinfile -a "path=/etc/sudoers line='rahul ALL=(ALL)ALL' insertafter=BOF"

ansible all -m lineinfile -a "path=/etc/sudoers line='rahul ALL=(ALL)ALL' insertafter=EOF"

ansible all -m lineinfile -a "path=/etc/sudoers line='rahul ALL=(ALL)ALL' insertafter='^root'"

ansible all -m lineinfile -a "path=/etc/sudoers line='rahul ALL=(ALL)ALL' insertbefore='^root'"

ansible all -m lineinfile -a "path=/etc/sudoers line='rahul ALL=(ALL)ALL' regexp='^root'"

ansible all -m lineinfile -a "path=/etc/sudoers line='rahul ALL=(ALL)ALL' regexp='^root' state

* **Ansible Playbook** :

Ansible playbooks are list of tasks that automatically executes for your specified inventory or group of hosts, one or more ansible tasks can be combined to make a play. An ordered grouping of tasks map to specific hosts. Tasks are executed in the order in which they are written. A playbook can include one or more play.

Tasks are executed by modules each of which performs a specific tasks in a playbook, a module contains metadata that determines when and where a task is executed as well as which user executes it.

There are thousands of ansible modules that can perform all kinds of IT tasks such as:

1. Cloud Management=>

It creates deletes or updates virtual cloud networks in cloud infrastructure environments. It can add remove or

update VMware clusters or Vspare clusture.

2. User Management=>

With the help of ansible we can manage users on Linux operating system, git, gitlab.

3. Networking=>

Ansible provides multiple modules handles API, CISCO iOS, NXOS, iOS XR devices.

4. Security=>

Open SSH\_cert generates an open ssh host or user certificates, IPA\_config manages global preIPA configuration settings.

5. Configuration management=>

With the help of pip ansible manage configurations of applications or systems effectively.

pip manages python library dependencies.

6. Communication=>

Ansible can send emails based on certain criteria and shows created, updated, deleted records in service now.

There are two ways to run ansible modules;

1. Adoc

2. Playbook

We can write a playbook in two different language:

1. YAML 2. JSON

We can write a playbook in three different formats:

1. Single line format

2. Multi line format

3. Dictionary format

1. Single Line format:

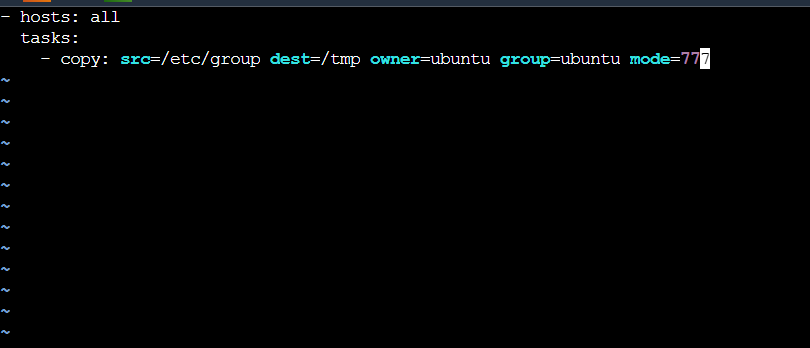
In single line format arguments will be present along with module

**command:**

- hosts: all

tasks:

-copy: src=/etc/group dest=/tmp owner=ubuntu group=ubuntu mode=777



Then run on Master ***ansible-playbook filename & check slave***

2. Multi line format:

In multi line format arguments will start from module name and each new argument will be on new line.

EX:

- hosts: all

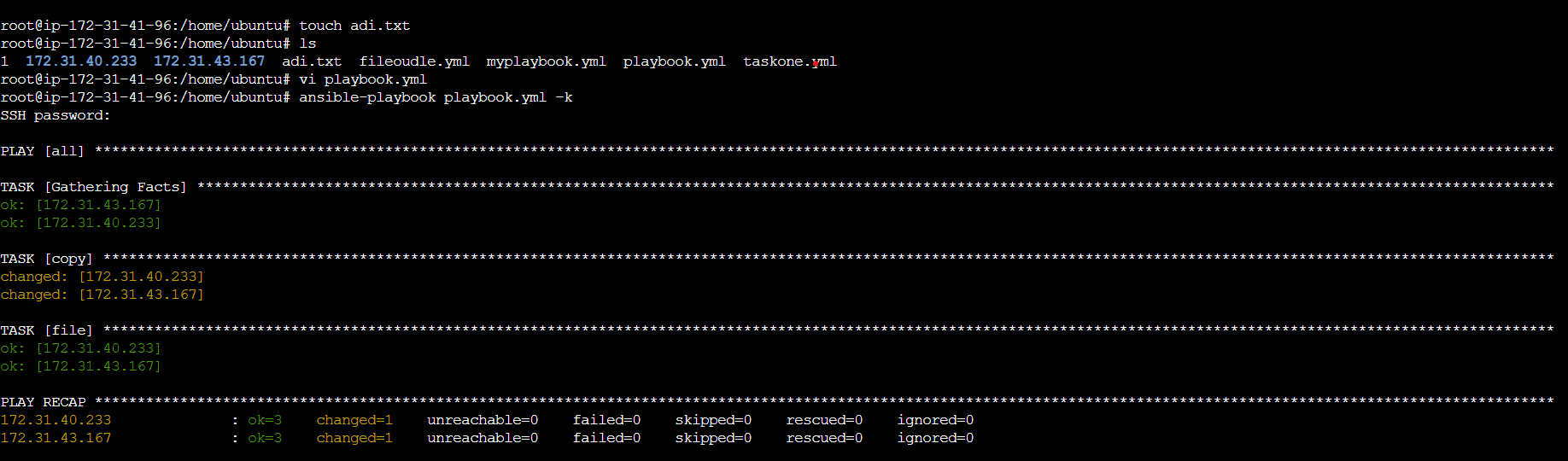
tasks:

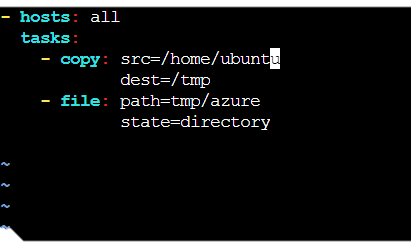
- copy: src=/etc/shadow

dest=/tmp

- file: path=/tmp/azure

state=directory



In Multiline command run Master check the slave machine.

*ansible-playbook multine.yml*

3. **Dictionary format:**

Command:

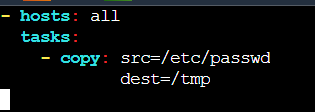
-hosts: all

tasks:

-copy:

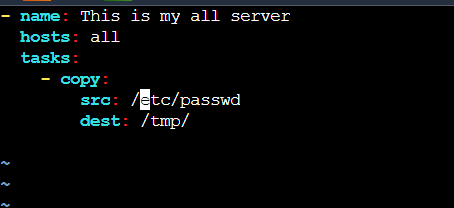
src: /etc/passwd

dest: /tmp/



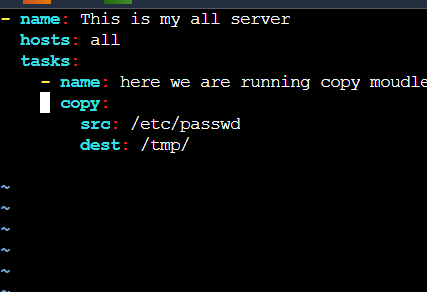
* if you want to give a message to play you need to write name tag above hosts tag

for eg.



* if you want to give a message inside module or task

for eg.



* **Defining task in separate file & calling them**.

we can define our task in separate file for simplicity & reliability purpose we need to import those task main playbook.

eg.

**How to define separate task in file?**

Taskone.yml comes under the code

- name: This is my all server

copy:

src: /etc/passed

dest: /tmp/

Importing task into main playbook

