**ANSIBLE**

Ansible is a configuration tool and also, agent less such that it requires nothing to be installed on target host machine except SSH and Python. It is push based methodology.

To check the ansible master is connected to the server: ansible group-name or ip address -m ping

**Difference between chef and ansible**

|  |  |
| --- | --- |
| **Chef** | **Ansible** |
| Chef is client server architecture | pushes based architecture |
| chef works with ruby | it works with python |
| we need to register each and every host individually | it supports dynamic inventory |
| chef is more secure than ansible | it is not secured by default, but we can use external module called vault |
| chef is not easy to setup | Ansible is easy to set up |

Note: Raw Module:

Raw module is also called as dirty module. If there is no phyton in target machine, we use raw module to install python or run some shell commands

**Ansible inventory:**

**Static inventory**: it is just a file containing list of ip address of target host machine and group it together based on the user requirements in “ini” format. The default location of the inventory file is /etc/ansible/host

**Dynamic inventory:** cloud itself will provide the files script and ini file. If we run script it will automatically fetch ip address of target machine and store in a ini file

**Ansible Modules:**

Ping module: it is used to check connection of host machine on inventory it will get reply with pong.

Ansible ipadress -m ping or ansible ipadress groupname -m ping.

**Package module:** it is same as yum in centos or redhat & apt in ubuntu (Debian)

Repository path in Linux: ubuntu /etc/apt/sources.list.d for yum: /etc/yum/repos.d

State: Ansible behave as a hydopotential behaviour

state: present: it will check for the package installed or not. If it is not installed, then package module is executed

State: latest: it will check whether the package is installed or not if it is installed it will ensure upto date or it will be installed with the latest version.

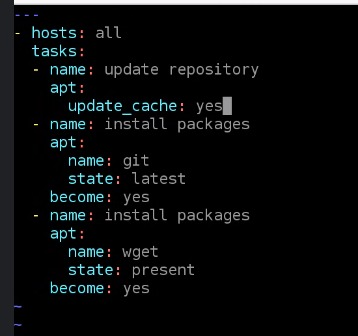
State: remove or State: absent: it will uninstall the installed package.

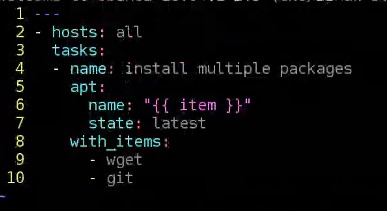
**Privilege access (become module):**

If I want to run with root permission or user with root permission, then we are going to use become module

become: yes – to run with root permission

become\_usr: username (ex: become\_usr: ubuntu) – to run the commands as a specific user





**Copy module:** it is used to copy a file from the host machine to the multiple target machines.

Syntax:

copy:

src: path of a file

dest: path of a destination

File Module: how to change the permission of a file.

File:

Path: path of a file

Mode: 0655

Group: username

Owner: username

Ansible vault: allows us to keep sensitive data such as password or keys in encrypted files rather than plane text in playbooks or rules.

Commands:

Create: it is used to create ansible vault file in the encrypted format “Ansible-vault create filename (yaml file)

“

View: it is used to view the data of encrypted files “ansible-vault view filename”

Encrypt: used to encrypt the unencrypted files “ansible-vault encrypt filename”

Decrypt: used to decrypt an encrypted file “ansible-vault decrypt filename”

--ask-vault-pass: while running playbooks if I need to provide passwords then we will use this.

Ansible-playbook filename –ask-vault-pass   
--vault-password-file: used to pass a password through a file

Ansible-playbook filename --vault-password-filename (file which contains password).

Ansible-galaxy:

It is a command used to create and manage the roles.

It is a public repository of ansible roles.

Ansible-galaxy list: it displays a list of installed roles with version number

Ansible galaxy remove role\_name: it will remove the installed roles

Ansible galaxy info: it will provide information about the ansible galaxy

Ansible galaxy init rolename: it will create a role

**Roles:** it will split the single playbook into multiple files. Ansible roles is defined with 8 directories and 8 files