

## **ANSIBLE**

### **(Push Based Methodology)**

Ansible works on push based methodology. Ansible is agentless such that it requires nothing to be installed on the target host machine except ssh connection and python.

#### **Difference between chef and ansible**

##### **Chef →**

- It works with client server architecture.
- It works with ruby.
- We need to register each and every host individually to the host machine.
- Chef is more secure than ansible.
- It is difficult to set up.

##### **Ansible →**

- It works with push based methodology. It works with python.
- It supports dynamic inventory.
- Ansible is not secure by default. We need to add an external module called vault.
- It is easy to set up.

#### **RAW Module**

It is also called a dirty module. If there is no python in the target machine we use raw module to install python or run it by some shell commands.

#### **Ansible Inventory    (/etc/ansible/hosts)**

We have 2 types of inventory

- **Static Inventory**
- **Dynamic Inventory**

#### **Static Inventory:**

Static inventory is a file which contains the list of IP addresses of target host machines and groups them together based on user requirements in json or ini format. Default location of inventory is </etc/ansible/hosts>

#### **Dynamic Inventory :**

Ansible team will provide two files, python script and ini file. When we execute the script it will automatically fetch the address of target host machines and store it in the inventory file. Only thing we have to update is the path of the inventory file in the python script.

## Modules:



- ❖ **Ping Module** : It is used to check connection of target host machines. It will reply with pong.

Syntax → *ansible ip\_address(target m/c) -m ping*

*ansible group\_name -m ping*

*ansible -i inventory\_filename -m ping*

- ❖ **Packaging Module** : It is used to install packages. It is the same as yum in centos/rhel and apt in ubuntu/debian.

- ❖ **copy module** ---> it is used to copy a file from the host machine to the multiple target machines.

### **Syntax:**

copy:

src: source path

dest: destination path

- ❖ how to change the permission of a file?

**sol.** Using file module

syntax:

file:

path: file path

mode: 0744

owner:

group:

- ❖ **State Module:**

*state: present* → It will check for the package is already installed or not. If not installed then only it will install the package.

*state: latest* → It will check whether the package is installed or not, if it is installed it will ensure up to date or install with the latest version.

*state: removed* → It will uninstall the package.

*state: absent* → It will uninstall the package.

### ❖ Become Module :

**Privilege Access :** If i want to run with root permissions or other user privilege permission.

For this we will use the *become* module

**Syntax** → *become: yes*  
*become\_user: user\_name*

❖ *update\_cache*: used to update repository

*update\_cache: yes*

**Ansible-vault:** (how you are going to secure ansible)

Sol: using ansible-vault

Ansible vault is a feature of ansible that allows you to keep sensitive data such as password or keys in encrypted files, rather than plaintext in playbooks or roles.

**Commands:**

- ❑ **Create:** it is used to create ansible vault file in the encrypted format.

**ansible-vault create 1.yml**

- ❑ **View:** To view the data of encrypted files.

**ansible-vault view 1.yml**

- ❑ **Edit:** It is used to edit the encrypted file

**ansible-vault view 1.yml**

- ❑ **Encrypt:** used to encrypt the unencrypted files

**ansible-vault encrypt 1.yml**

- ❑ **Decrypt:** used to decrypt an encrypted files

**ansible-vault decrypt 1.yml**

- ❑ **--ask-vault-pass:** used to provide passwords while running playbooks

**Ansible-playbooks 1.yml --ask-vault-pass**

- ❑ **--vault-password-file:** used to pass a vault password through a file

**Ansible-galaxy:** it is a command used to create and manage the roles. Ansible galaxy is a large public repository of ansible roles.

**ansible-galaxy list :** it display a list of installed roles with version number

**ansible-galaxy remove role\_name :** it will remove an installed role

**ansible-galaxy info :** it will provide a information about ansible-galaxy

**ansible-galaxy init Role\_name :** to create a role

**Roles:** Roles splits the single playbook into multiple files. Ansible roles are defined with 8 directory and with 8 files, roles must include at least one of the above directories.

#### Directory structure of roles:

sample

```
|— defaults
|   |— main.yml
|— files
|— handlers
|   |— main.yml
|— meta
|   |— main.yml
|— README.md
|— tasks
|   |— main.yml
|— templates
|— tests
|   |— inventory
|   |— test.yml
|— vars
|— main.yml
```

**default:** default variable for the roles

**files:** contain files which can be deployed through roles

**handlers:** which contains handlers which may be used by this role or even anywhere outside this role

**meta:** Define some meta data for this role

**tasks:** contain the main list of tasks to be executed by the roles

**templates:** it contains templates(files) which can be deployed through this roles

**tests:** it may contain simple inventory and a test, it may be useful if u on automated testing process which is build through roles

**vars:** variables used in the role

### **Difference B/w default and vars**

Default have lowest priority value than any other variables in ansible. Vars will have the highest priority than default

### **Difference B/w files and templates:**

Ansible files are used to copy static files without changing any contents from host machine to target machine.

Templates take the files from the host machine and change the contents of files (value of variable defined in templates) before copying. It can be done using a jinja filter (.j2 format) and then it will copy a file to the target machine.

Handlers: it are special type of tasks which is executed only when there is change in task to which handler is defined

How will you call handlers?

Using notify

**Syntax:** **- notify: handlers name**

How will you include a yml (playbooks) file within another yml (playbooks) file or how will you call yml files from another yml file.

Using include

**Syntax:** **- include: yml\_file\_name**