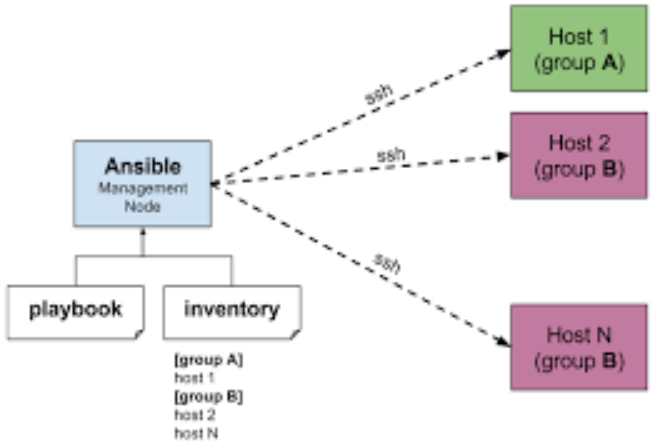
ANSIBLE

What is Ansible

Ansible is a popular tool for automating and managing cloud and on-premises infrastructure, Ansible is a flexible, powerful tool for automating infrastructure management, configuration, networking, and deployment tasks.

How Ansible Work



Playbook

Actions like INSTALL, UNINSTALL, UPDATE

Inventory

Environments like DEV, PROD, TEST and the IP address of HOST

ANSIBLE SETUP

Follow the steps:

Master Server

1. yum install python-pip -y -----→ Install python dependency for Ansible
2. amazon-linux-extras install ansible2 -y ----→ Install Ansible
3. ansible –-version ----→ check the Version
4. ssh-keygen -----→ generate key
5. passwd root ------→set Password for root user
6. vi /etc/ssh/sshd_config ---→ change the permissions
   PermitRootLogin yes
   PasswordAuthentication yes
7. systemctl restart ssh ------ > restart the configs

Node Server

1passwd root ------→set Password for root user

2vi /etc/ssh/sshd_config ---→ change the permissions in config

   PermitRootLogin yes
   PasswordAuthentication yes

3. systemctl restart ssh ------ > restart the configs

Connect the MASTER To NODE server ---- SSH

ssh-copy-id root@IP address of Node server ---→execute in Master server

Set the environment to the NODE server

etc/ansible/host ---→ inventory config file

[dev]

IP address of node server

To check the ansible setup

ansible all -m ping

To the HOSTS in Environment

ansible all --list-hosts ------→ all servers

ansible dev –-list-hosts -----→ all the servers of dev environment

ansible all[0:9] –-list-hosts -----→ top 10 servers

What is Ansible Playbook

Ansible Playbook is a script file - written in YAML format - that defines the tasks required to configure servers. Ansible executes the listed tasks on the target machine sequentially.

Playbooks are mainly divided into sections like

TARGET SECTION: defines host server

VARIABLE SECTION: defines variables

TASK SECTION: action you are performing

name. yaml or name.yml -----→ Name of the Playbook

```
---
    - hosts: environment_name  ------→ (dev, test, prod)
      connection: ssh
       tasks:
          -  name: mention the task that you are doing
             action: yum name=git state=present
…
```

Execute the playbook

ansible-playbook name.yaml

To check the playbook

ansible-playbook name.yaml --check

States in Ansible

Install = present

Uninstall =absent

Start = started

Stop = stopped

Restart = restarted

Modules in Ansible Playbook

- yum
- action
- command
- service

Variables in Ansible Playbook

```
---
        - hosts: environment_name   ------→ (dev, test, prod)
          connection: ssh
           vars:
              variable_name: git
           tasks:
              -  name: mention the task that you are doing
                 action: yum name={{ variable_name }} state=present
    …
```

Loops in Ansible playbook

```
---
        - hosts: environment_name   ------→ (dev, test, prod)
          connection: ssh
          tasks:
              -  name: mention the task that you are doing
                 user: name={{item}} state=present
```

```
        with_items:
        - abc
        - xyz
…
```

Create a File and add the data

```
---
    - hosts: environment_name   ------→ (dev, test, prod)
      connection: ssh
      tasks:
        - name: create a file
          file: path="book" state=touch
        - name: add some data
          copy: dest="book" content= | Hi welcome to the class
  …
```

Download File from online

```
---
    - hosts: environment_name   ------→ (dev, test, prod)
      connection: ssh
      tasks:
        - name: download file
          get_url: url="https://media.geeksforgeeks.org" dest="root"
  …
```

To Skip the tasks

ansible-playbook name.yml –-skip-tags=" tag_name"

ansible-playbook name.yml –-skip-tags=" tag_name1,tag_name2"

ansible-playbook name.yml tags=" tag_name" --→ only this task will execute

When condition in Ansible playbook

```
---
    - hosts: environment_name   ------→ (dev, test, prod)
      connection: ssh
       tasks:
          -  name: install httpd in redhat
             action: yum name=httpd state=present
             when: ansible_os_family == "RedHat"
          -  name: install httpd in ubuntu
             action: yum name=apache2 state=present
             when: ansible_os_family == "Ubuntu"
  …
```

Change Owners and permissions

```
---
    - hosts: environment_name   ------→ (dev, test, prod)
      connection: ssh
       tasks:
          -  name: create a user
             user: name=TCS state=present
          -  name: create a group
             group: name=Wipro state=present
          -  name: create a file
             file: path="office" state=present owner=TCS group=Wipro
  mode=777
```

…

Encrypt

ansible-vault encrypt name.yml -------→to encrypt the file

ansible-vault view name.yml -----→ to view the data

ansible-vault decrypt name.yml -----→ to decrypt the file

ansible-vault rekey name.yml ---→ to change password

ansible-vault create file_name -----→ It will create new file and encrypt it

To print msg

```
---
    - hosts: environment_name   ------→ (dev, test, prod)
      connection: ssh

      tasks:
        - debug:
            msg: “welcome”
```

…

Commands

- ansible_os_family
- ansible_kernel
- ansible_processor_cores
- ansible_default_ipv4
- ansible_memory_mb.real
- ansible_memory_mb.real.free
- ansible_memory_mb.real.used

To store and print

```
---
    - hosts: environment_name   ------→ (dev, test, prod)
```

```
connection: ssh

 tasks:

   -  name: to know git version

      command: git -v

      register: xyz

   - name:

     debug:

      msg: “{{xyz.stdout}}”
```

…

ANSIBLE GALAXY

Ansible Galaxy is a website and command-line tool for sharing and managing collections of playbooks.

- ansible-galaxy search playbook_name
- ansible-galaxy install playbook_name

What is Ansible Tower

Ansible Tower is a commercial, web-based graphical user interface (GUI) and automation platform that provides enhanced user experience for managing Ansible automation at scale.