## **Ansible installation ubuntu**

sudo apt-add-repository ppa:ansible/ansible

sudo apt update

sudo apt install ansible -y

**SSH Connection between Ansible Control node to target servers:**

Private key file Method:

Use the private key of existing target servers.

Step1: download the private key of target servers

Step 2: chmod 400 privatekeyname

Step3: give the target server ip and location of private key in inventory file like below

Step 4: test connection using command,

ansible all -i inventoryfilelocation -m ping  
  
Pls refer to recording for any queries on ssh connection

## **basic inventory file**

If the you are using private key to connect to target servers

[all]

172.31.33.232 ansible\_ssh\_private\_key\_file="location of privatekey"

## **Execution of playbook**

Ansible-playbook playbookname.yml --syntax-check

ansible-playbook playbookname.yml --check

ansible-playbook playbookname.yml -vvv ( To troubleshoot the playbook if it fails for any version vvv(Verbose) will print everything what is happening with the playbook in the background)

ansible-playbook playbookname.yml (if you are using default inventory location (/etc/ansible/hosts)

Ansible-playbook -i inventoryfilelocation playbookname.ymal (custom inventory file)

## **Playbook 1 - To install apache2 on the ubuntu server**

---

- hosts: all

become: yes

tasks:

- name: Demo Playbook

apt:

name: apache2

state: present

- name: Demo for starting service

service:

name: apache2

state: started

## **Playbook2 - To uninstall apache2 and its configuration**

---

- name: Uninstall Apache2 and its dependencies

hosts: all

become: yes

tasks:

- name: Uninstall Apache2 and dependencies

apt:

name: apache2

state: absent

purge: yes

autoremove: yes

## **Playbook3 - To install multiple packages together on Ubuntu**

- hosts: all

become: yes

tasks:

- name: Install packages

apt:

name:

- ntpdate

- nmap

- nginx

state: present

## **Playbook4 - To uninstall multiple packages together on Ubuntu**

- hosts: all

become: yes

tasks:

- name: Uninstall packages

apt:

name:

- ntpdate

- nmap

- nginx

state: absent

purge: yes

autoremove: yes

## **Playbook 5 - Use ansible tags**

---

- hosts: all

become: yes

tasks:

- name: Demo Playbook

tags: first

apt:

name: apache2

state: present

- name: Demo for starting service

tags: Second

service:

name: apache2

state: started

## **execute tags playbook**

ansible-playbook playbookname.yml --tags first (Executing the first tag only)

ansible-playbook playbookname.yml --tags second (Executing second tag only)

ansible-playbook playbookname.yml --list-tags

## **Playbook 6 - Execute shell scripts**

shell script

#!/bin/bash

sudo apt update -y

sudo apt install apache2 -y

---

- name: Install Apache2 using Script

hosts: all

become: yes

tasks:

- name: Copy the script file

copy:

src: /home/ubuntu/install.sh

dest: /tmp/

- name: Execute the script

shell: sh /tmp/install.sh

## **Playbook 7 - Replace module**

---

- name: Replace Text in a File

hosts: all

become: yes

tasks:

- name: Replace the file

replace:

dest: /tmp/a.txt

regexp: 'Thursday'

replace: 'Wednesday'

On remote Machine

Create file on the path /tmp by the a.txt

vi a.txt

Monday

Tuesday

Wednesday

Thursday

Friday

Saturday

Sunday

## **Playbook 8 - notfier and handler (trigger specific tasks only if there is change of events)**

---

- name: Example playbook with notify and handler

hosts: all

become: yes

tasks:

- name: Install Apache2 package

package:

name: apache2

state: present

notify: Restart Apache

handlers:

- name: Restart Apache

service:

name: apache2

state: restarted

## **Playbook 9 - Sample Playbook using Handlers, Notifier, Variables, apt and copy ansible modules**

---

- name: Deploy Web Content

hosts: all

become: yes

vars:

srcpath: /root/index.html

destpath: /var/www/html

tasks:

- name: Install Apache2 package

apt:

name: apache2

state: present

notify: Restart Apache

- name: Copy the file

copy:

src: "{{ srcpath }}"

dest: "{{ destpath }}"

notify: Restart Apache

handlers:

- name: Restart Apache

service:

name: apache2

state: restarted

index.html

<html>

<h1> Hello ansible </h1>

<body> sample playbook </body>

</html>

aws ec2 slave instance:80

**Ansible Roles:**

ansible-galaxy init rolename

Step1: Create roles folder

Step2: inside roles folder run command (ansible-galaxy init rolename)

Step3: paste specific logic in specific folders main.yml files

Step4: paste below logic in tasks folder -> main.yml file

- name: Install Apache2 package

apt:

name: apache2

state: present

notify: Restart Apache

- name: Copy the file

templates:

src: "{{ srcpath }}"

dest: "{{ destpath }}"

notify: Restart Apache

Step5: Paste below logic in handlers folder -> main.yml

- name: Restart Apache

service:

name: apache2

state: restarted

Step6: paste below file and its logic in templates folder

index.html.j2

<html>

<h1> Hello ansible </h1>

<body> sample playbook </body>

</html>

Step7: paste below in defaults folder main.yml

srcpath: templates/index.html.j2

destpath: /var/www/html/index.html

Step8: come outside roles folder and create below file.yml and run it

—--

* hosts: all

become: yes

roles:

- rolename

Please check the recording once in case of issues.

Extra playbook:

---

- name: Sample Playbook with Skipped and Ignored Tasks

hosts: all

tasks:

- name: Update packages cache

yum:

name: '\*'

state: latest

when: ansible\_os\_family == 'RedHat'

- name: Start nginx service

service:

name: nginx

state: started

ignore\_errors: yes

**System admin playbook:**

---

- name: system admin tasks

hosts: all

become: yes

tasks:

- name: Create a folder

file:

path: /home/ubuntu/test

state: directory

mode: '0755'

- name: Create a file

file:

path: /home/ubuntu/testfile.txt

state: touch

- name: create user

user:

name: newuser

state: present

create\_home: yes

shell: /bin/bash

- name: change file permissions

file:

path: /home/ubuntu/testfile.txt

mode: '0755'

**Windows** **playbook**:

Config:

Ansible control server:

sudo apt install python3-pip

pip3 install pywinrm>=0.3.0

Sample inventory file:

[windowsserver]

172.31.6.144 ansible\_user=Administrator ansible\_password=MnC8KDgQ02KB;.GE=aiTpHr3Zz-Jwx(8 ansible\_connection=winrm ansible\_winrm\_server\_cert\_validation=ignore

Windows OS server:

Open powershell as administrator

$url=”<https://raw.githubusercontent.com/ansible/ansible-documentation/ae8772176a5c645655c91328e93196bcf741732d/examples/scripts/ConfigureRemotingForAnsible.ps1>”

Invoke-Webrequest -Uri $url -OutFile ConfigureRemotingForAnsible.ps1

.\ConfigureRemotingForAnsible.ps1

Verify connection:

$ansible windowsserver -i hosts –m win\_ping

---

- name: Playbook example for windows

hosts: windowsserver

tasks:

- name: create folder

win\_file:

path: C:\ansible\_download

state: directory

- name: Download software setup

win\_get\_url:

url: https://archive.apache.org/dist/httpd/binaries/win32/httpd-2.2.25-win32-x86-no\_ssl.msi

dest: C:\ansible\_download

- name: Install MSI

win\_package:

path: C:\ansible\_download\httpd-2.2.25-win32-x86-no\_ssl.msi

state: present