PLAYBOOK --YAML CODE

* **A play** is an ordered set of tasks which should be run against hosts selected from your inventory.
* **A playbook** is a text file that contains a list of one or more plays to run in order.
* Playbooks are written in YAML format.
* YAML stands for Yet Another Markup Language.
* **Playbooks** are one of the core features of Ansible and tell Ansible what to execute.

Like the name is saying, a playbook is a collection of plays. Through a playbook, you can designate specific roles to some of the hosts and other roles to other hosts. By doing so, you can orchestrate multiple servers in very diverse scenarios, all in one playbook.

**Different YAML Tags**

1) name: pecifies the name of the Ansible playbook

2) hosts: targeting IP or group or all

3) vars: lets you define the variables

4) task: tasks are a list of actions one needs to perform.

**Sample Format of Playbook**

**# vi test1.yml**

--- Playbook start

- hosts: webservers Specify the group or servers as per inventory to execute tasks

become: true

tasks:

- name: Copy Tomcat ZIP file to install location Short description of the task

copy: src=/home/ansible/deepak/apache-tomcat-8.5.31.tar.gz dest=/opt/deepak/tomcat

**To run any playbook**

#ansible-playbook test1.yml

**To check the playbook for syntax errors**

**#**ansible-playbook test1.yml --syntax-check

**To view hosts list**

ansible-playbook test1.yml --list-hosts

**Example 1:** Create the file on the target machines or servers as mentioned in the inventory file and the webserver's group

# vi test.yml

---

- hosts: webservers

become: true

tasks:

- name: Create a file

file: path=/home/ansible/deepak.txt state=touch

**Example 2:** Create a directory with the mode as 775 and owner/group as Ansible.

---

- hosts: webservers

become: true

tasks:

- name: Create directory

file: path=/home/ansible/niranjan state=directory mode=775 owner=ansible group=ansible

**Example 3:**Create a user.

---

- hosts: webservers

become: true

tasks:

- name: Create User

user: name=deepak password=deepak groups=wheel shell=/bin/bash

**Example 4:** Remove user.

---

- hosts: webservers

become: true

tasks:

- name: Remove User

user:

name=deepak state=absent remove=yes force=yes

**Example 5:** Copy content to a file using the copy module.

---

- hosts: webservers

become: true

tasks:

- name: Copy content to file

copy: content="Hello World deepak \n" dest=/home/ansible/deepak.txt

or

copy: src=/home/ansible/deepak.txt dest=/tmp/deepak.txt

**Example 6:**Archive or ZIP files and Folders

1. zip the file niranjan.txt to niranjan.zip file

---

- hosts: all

become: true

tasks:

- name: Ansible zip file example

archive:

path: /home/ansible/niranjan.txt

dest: /home/ansible/niranjan.zip

format: zip

1. zip multiple files to niranjan.zip file.

---

- hosts: all

tasks:

- name: Ansible zip multiple files example

archive:

path:

- /home/ansible/niranjan1.txt

- /home/ansible/niranjan2.txt

dest: /home/ansible/niranjan.zip

format: zip

1. zip all files in the /home/ansible directory.

- hosts: all

tasks:

- name: Ansible zip directory example

archive:

path:

- /home/ansible

dest: /home/ansible/niranjan.zip

format: zip

**Example 7:**Working with date and timestamp

1. displays the date.

---

- hosts: webservers

become: true

tasks:

- name: Date and Time Example in Ansible

debug:

var=ansible\_date\_time.date

1. displays the time.

---

- hosts: webservers

become: true

tasks:

- name: Date and Time Example in Ansible

debug:

var=ansible\_date\_time.time

1. create a dynamic file based on the current date for **E.g**. deepak2020-09-10.log

- hosts: all

tasks:

- name: Ansible timestamp filename example

command: touch niranjan{{ansible\_date\_time.date}}.log

**Example 8:** Install vim editor and GIT on the target servers

---

- hosts: webservers

become: true

tasks:

- name: Install Package

yum: name=vim,git state=latest

----------------------------------------------------------------------------------------------------------------

1. Install httpd in centos

# nano test1.yml

---

- hosts: all

tasks:

- name: Install httpd

yum: name=httpd state=present

How to run

#ansible-playbook test1.yml

1. Uninstall httpd in centos

# nano test1.yml

---- hosts: all

tasks:

- name: Install httpd

yum: name=httpd state=absent

**Same program in other way**

# nano test1.yml

---

- hosts: all

tasks:

- name: Install httpd

yum:

name: httpd

state: present

1. installing and starting httpd service

---

- hosts: webservers

become: true

tasks:

- name: Install Package

yum: name=httpd state=present

- name: Start httpd service

service: name=httpd state=started

12) Install Apache in ubuntu

# nano test1.yml

---- hosts: all

become: true

tasks:

- name: Update apt-cache

apt: update\_cache=yes

- name: Install apache2

apt: name=apache2 state=latest or present

1. For uninstallation

Change the state : absent

How to run

#ansible-playbook test1.yml

1. To allow http in ubuntu firewall

#nano test2.yml

---

- hosts: all

become: true

- name: Allow all access to tcp port 80

ufw:

rule: allow

port: '80'

proto: tcp

**How to run**

#ansible-playbook test1.yml

15) HTTP web server configuration in AWS cloud(RHEL OS)

# vi httpd.yaml

---

- name: This sets up an httpd webserver

hosts: webservers

tasks:

- name: Install apache packages

yum:

name: httpd

state: present

- name: copying index.html file

template: src=/etc/ansible/index.html dest=/var/www/html

- name: ensure httpd is running

service:

name: httpd

state: started

16) HTTP web server configuration in On premise Linux Server

# vi httpd.yaml

---

- name: This sets up an httpd webserver

hosts: 192.168.43.101

tasks:

- name: Install apache packages

yum:

name: httpd

state: present

- name: copying index.html file

template: src=/etc/ansible/index.html dest=/var/www/html

- name: ensure httpd is running

service:

name: httpd

state: started

- name: Open port 80 for http access

firewalld:

service: http

permanent: true

state: enabled

- name: Restart the firewalld service to load in the firewall changes

service:

name: firewalld

state: restarted

# ansible-playbook httpd.yml

17) Apache web server configuration in AWS cloud

# vi apache.yaml

---

- hosts: apache

tasks:

- name: install apache2

apt: name=apache2 update\_cache=yes state=latest

- name: copying index.html file

template: src=/etc/ansible/mobile.html dest=/var/www/html

- name: enabled mod\_rewrite

apache2\_module: name=rewrite state=present

notify:

- restart apache2

handlers:

- name: restart apache2

service: name=apache2 state=restarted

18) Create the EC2 Instance

# yum -y install python-pip

# pip install boto

# vi ec2.yml

---

- name: Launching the AWS Instance

hosts: localhost

tasks:

- name: Launch the AWS Instance

ec2:

key\_name: mysql

region: us-east-1

instance\_type: t2.micro

image: ami-00eb20669e0990cb4

count: 2

vpc\_subnet\_id: subnet-2100387d

assign\_public\_ip: yes

aws\_access\_key: AKIAIR7Q6ABR572FMODA

aws\_secret\_key: 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

**same but in another way**

19) Create the EC2 Instance

a) create one iam role having ec2 full permission and add it to the ansible server

b)# yum -y install python-pip

c) # pip install boto

d) vi ec2.yml

---

- name: Launching the AWS Instance

hosts: localhost

tasks:

- name: Launch the AWS Instance

ec2:

key\_name: deepakawskey

region: ap-south-1

instance\_type: t2.micro

image: ami-0e306788ff2473ccb

count: 2

vpc\_subnet\_id: subnet-fe3144b2

assign\_public\_ip: yes

20) Stop the EC2 Instance

#vi ec2stop.yaml

---

- name: Stop the Ec2 Instance

hosts: localhost

tasks:

- name: Stop the Ec2 Instance

ec2:

instance\_ids: i-0d65770e13c2e1445

region: us-east-1

state: stopped

aws\_access\_key: AKIAIR7Q6ABR572FMODA

aws\_secret\_key: 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

21) Start the EC2 Instance

#vi ec2start.yaml

---

- name: Stop the Ec2 Instance

hosts: localhost

tasks:

- name: Stop the Ec2 Instance

ec2:

instance\_ids: i-0d65770e13c2e1445

region: us-east-1

state: running

aws\_access\_key: AKIAIR7Q6ABR572FMODA

aws\_secret\_key: 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

22) Playbooks can contain multiple plays.

**---**

**-** hosts**:** webservers

remote\_user**:** root

tasks**:**

**-** name**:** ensure apache is at the latest version

yum**:**

name**:** httpd

state**:** latest

**-** name**:** write the apache config file

template**:**

src**:** /srv/httpd.j2

dest**:** /etc/httpd.conf

**-** hosts**:** databases

remote\_user**:** root

tasks**:**

**-** name**:** ensure postgresql is at the latest version

yum**:**

name**:** postgresql

state**:** latest

**-** name**:** ensure that postgresql is started

service**:**

name**:** postgresql

state**:** started

23) Install and Setup Docker Using Ansible on Ubuntu

1. Vi test 1.yml

---

- hosts: all

tasks:

- name: Add Docker GPG key

apt\_key: url=https://download.docker.com/linux/ubuntu/gpg

- name: Add Docker APT repository

apt\_repository:

repo: deb [arch=amd64] https://download.docker.com/linux/ubuntu {{ansible\_distribution\_release}} stable

- name: Install list of packages

apt:

name: "{{ item }}"

state: present

update\_cache: yes

with\_items:

- apt-transport-https

- ca-certificates

- curl

- software-properties-common

- docker-ce

- python3-pip

1. Pulling image from dockerhub

---

- hosts: all

tasks:

- name: Install Docker Module for Python

pip:

name: docker

- name: Pull CentOS Docker image

docker\_image:

name: centos

source: pull

1. Downloading multiple images ( centos, Ubuntu, wordpress, mysql)

---

- hosts: 172.31.0.96

tasks:

- name: Install Docker Module for Python

pip:

name: docker

- name: Pull CentOS Docker image

docker\_image:

name: "{{ item }}"

source: pull

with\_items:

- centos

- ubuntu

- wordpress

1. Running container

---

- hosts: all

tasks:

- name: Install Docker Module for Python

pip:

name: docker

- name: Create default containers

docker\_container:

name: depweb1

image: centos

command: "docker run -it -d"

state: present