# 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/niranjan.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=ansible 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

a) 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
```

b) 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
```

c) 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

a) displays the date.

```
---
- hosts: webservers
  become: true
  tasks:
  - name: Date and Time Example in Ansible
    debug:
     var=ansible_date_time.date
```

b) displays the time.

```
---
- hosts: webservers
  become: true
  tasks:
  - name: Date and Time Example in Ansible
  debug:
   var=ansible_date_time.time
```

c) 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
```

---

9) 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

10)Uninstall httpd in centos

# nano test1.yml

```
---- hosts: all
      tasks:
          - name: Install httpd
            yum: name=httpd state=absent
```

11)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
```

13)For uninstallation

Change the state : absent

How to run

#ansible-playbook test1.yml

14) 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 Centos

# 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

16) HTTP web server configuration in AWS cloud

# 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
# ansible-playbook   httpd.yml
```

17) 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
```

18) 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
```

19) Install JDK

```
---
- hosts: webservers
  become: true
  vars:
   download_url: http://download.oracle.com/otn-pub/java/jdk/8u171-b11/512cd62ec5174c3487ac17c61aaa89e8/jdk-8u171-linux-x64.rpm

  tasks:
  - name: Download JDK 8 RPM file
    command: "wget --no-check-certificate --no-cookies --header 'Cookie: oraclelicense=accept-securebackup-cookie' {{download_url}} "
  - name: Install JDK 8
    command: "rpm -ivh jdk-8u171-linux-x64.rpm"
```

20) Install Tomcat 8

```
---
- hosts: webservers
  become: true
  gather_facts: no
  tasks:
  - name: Download Tomcat
    get_url: url=http://www-us.apache.org/dist/tomcat/tomcat-8/v8.5.32/bin/apache-tomcat-8.5.32.tar.gz dest=/home/ansible

  - name: Extract the file downloaded tomcat file
    command: tar xvf apache-tomcat-8.5.32.tar.gz

  - name: Move the Tomcat directory to a smaller one
    command: mv apache-tomcat-8.5.32 tomcat

  - name: Change Ownership and group of the Tomcat directory
    file: path=/home/ansible/tomcat owner=ansible group=ansible mode=775 state=directory recurse=yes
```

```
  - name: Start Tomcat
    command: nohup /home/ansible/tomcat/bin/startup.sh # Execute command even after you have exited from the shell prompt
    become: true
    become_user: ansible
```

21) 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

21) 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
```

22) 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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
```

## 23) configuring docker in ec2 instances.

```
name: configure yum repo for docker
hosts: ec2-instance
become: yes
become_user: root
tasks:
#Add Repo for docker software
- name: Add repository
yum_repository:
name: docker
description: Docker YUM repo
baseurl: https://download.docker.com/linux/centos/7/x86_64/stable/
gpgcheck: no
- name: Install docker
package:
name: docker-ce-18.09.1-3.el7.x86_64
state: present
#Enable docker service
- name: enable Docker services
service:
name: "docker"
state: started
enabled: yes
```

## 24) Run an apache web server in a docker container

- name: launching apache server in docker

docker_container:

name: webserver

image: httpd

state: started

exposed_ports:

- "80"

ports:

- "80:80"

volumes:

- /root:/usr/local/apache2/htdocs/

**23) 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
```

# HTTP web server configuration in AWS cloud

For RHEL-7/CentOS-7/Amazon Linux AMI2

```
# 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
```

# ansible-playbook   httpd.yml

## HTTP web server configuration in Onpremise 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

## Apache web server configuration in AWS cloud

## For Ubuntu

```
---
- 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
```

# ansible-playbook httpd.yml

## Create ngnix server

# vi ngnix.yml

```
---
- name: Install nginx
hosts: testserver
become: true
tasks:
- name: Add epel-release repo
yum:
name: epel-release
state: present
- name: Install nginx
yum:
name: nginx
state: present
- name: Start NGiNX
service:
name: nginx
state: started
```

#ansible-playbook nginx.yml

# Create ec2 instance through YAML

```
# vi ec2test.yaml

---
- name: Launching the AWS Instance
hosts: localhost
tasks:
- name: Launch the AWS Instance
ec2:
key_name: mysql
region: ap-south-1
instance_type: t2.micro
image: ami-00eb20669e0990cb4
count: 1

vpc_subnet_id: subnet-2100387d
assign_public_ip: yes
aws_access_key: AKIAIR7Q6ABR572FMODA
aws_secret_key: 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
```