Playbooks :

Playbooks are the files where Ansible code is written. 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. They are like a to-do list for Ansible that contains a list of tasks.

Playbooks contain the steps which the user wants to execute on a particular machine. Playbooks are run sequentially. Playbooks are the building blocks for all the use cases of Ansible.

Playbook Structure :

Each playbook is an aggregation of one or more plays in it. Playbooks are structured using Plays. There can be more than one play inside a playbook.

The function of a play is to map a set of instructions defined against a particular host.

YAML is a strict typed language; so, extra care needs to be taken while writing the YAML files. There are different YAML editors but we will prefer to use a simple editor like vim.

A YAML starts with --- (3 hyphens)

The Different YAML Tags :

Let us now go through the different YAML tags. The different tags are described below -

name

This tag specifies the name of the Ansible playbook. As in what this playbook will be doing. Any logical name can be given to the playbook.

hosts

This tag specifies the lists of hosts or host group against which we want to run the task. The hosts field/tag is mandatory. It tells Ansible on which hosts to run the listed tasks. The tasks can be run on the same machine or on a remote machine. One can run the tasks on multiple machines and hence hosts tag can have a group of hosts' entry as well.

vars

Vars tag lets you define the variables which you can use in your playbook. Usage is similar to variables in any programming language.

tasks

All playbooks should contain tasks or a list of tasks to be executed. Tasks are a list of actions one needs to perform. A tasks field contains the name of the task. This works as the help text for the user. It is not mandatory but proves useful in debugging the playbook. Each task internally links to a piece of code called a module. A module that should be executed, and arguments that are required for the module you want to execute.

Now Lets start writing Some Sample Plays

#mkdir /Plays

#cd /Plays

1) Writing Sample Play

#vim play1.yml

```
---
 - name : Sample Play to Create User ( Play 1 )
   hosts : db
   become : true
   tasks :
     - name : This Task is to create user RAVI ( Task 1 of Play 1 )
       user :
         name : ravi
         uid : 9001
         state : present
```

:wq!

#ansible-playbook --syntax-check play1.yml **(To Perform the Syntax Check on play1.yml)**

#ansible-playbook play1.yml --list-hosts **(To see what hosts would be affected by a playbook before you run it)**

#ansible-playbook -C play1.yml **(Executing a playbook with out doing any changes ..Dry RUN)**

#ansible-playbook play1.yml **(Executing Play Book)**

2) Writing Play With multiple tasks

#vim play2.yml

```
---
 - name : Installing Few Packages & Ensuring Selected Services are UP ( Play 1 )
   hosts : db
   become : true
   tasks :
     - name : Installing chrony Package ( Task 1 of Play 1 )
       yum :
         name : chrony
         state : latest
     - name : Installing Apache Package ( Task 2 of Play 1 )
       yum :
         name : httpd
         state : latest
     - name : Starting & Enabling NTP Service ( Task 3 of Play 1 )
       service :
         name : chronyd
```

```
      state : started
      enabled : true
  - name : Starting & Enabling Apache Service ( Task 4 of Play 1 )
    service :
      name : httpd
      state : started
      enabled : true
:wq!
```

3)Writing a Playbook to Setup Webserver

```
#vim apache.yml
---
 - name : Setting Up Webserver ( Play 1 )
   hosts : db
   become : true
   tasks :
    - name : Installing Apache Package ( Task 1 of Play 1 )
      yum :
        name : httpd
        state : latest
    - name : Defining Index.html file ( Task 2 of Play 1 )
      copy :
        src : /apache/index.html
        dest : /var/www/html/index.html
    - name : Ensure Apache Service i Started & Enabled ( Task 3 of Play 1 )
```

```
    service :
      name : httpd
      state : started
      enabled : true
:wq!
```

4)Writing Playbook with Multiple Plays :

```
#vim multiplay.yml
---
 - name : Setting up Few Services ( Play 1 )
   hosts : db
   become : true
   tasks :
     - name : Installing Dovecot Service ( Task 1 of Play 1 )
       yum :
         name : dovecot
         state : latest
     - name : Installing DNS Package ( Task 2 of Play 1 )
       yum :
         name : bind
         state : latest
 - name : Ensuring Services are UP ( Play 2 )
   hosts : db
```

```
  become : true
  tasks :
   - name : Starting Dovecot Service ( Task 1 of Play 2 )
     service :
       name : dovecot
       state : started
       enabled : true
   - name : Starting DNS Service ( Task 2 of Play 2 )
     service :
       name : named
       state : started
       enabled : true
- name : Manually Resolving with out DNS ( Play 3 )
  hosts : db
  become : true
  tasks :
   - name : Adding Entries to /etc/hosts ( Task 1 of Play 3 )
     lineinfile :
       path : /etc/hosts
       line : '192.168.1.200  ae'
       state : present
:wq!
```

Booleans Variations in Playbook

True | true | YES | yes | 1

False | false | NO | no | 0 (Zero)

PlayBook Syntax Variations :

```
hosts :
    - 192.168.1.201
    - 192.168.1.205
    - 192.168.1.210
    - 192.168.1.215
    - 192.168.1.220
    - db
    - mail_servers
    - middle_ware
```

Short Form :

```
hosts : [192.168.1.201, 192.168.1.205, 192.168.1.210, 192.168.1.215, 192.168.1.220, db, mail_servers, middle_ware]
tasks :
    - name : Starting DNS Service ( Task 2 of Play 2 )
      service :
        name : named
        state : started
        enabled : true
```

Short Form :

```
- name : Some task Description
  service : name=named, state=started, enabled=true
```

5) Writing Playbook with New Variations in Syntax

#vim webserver_user.yml

```
---
 - name : Setting up Webserver Locally
   hosts :
    - 192.168.1.108
    - 192.168.1.109
    - 192.168.1.110
   tasks :
    - name : Installing the latest Packages req to setup Webserver
      yum :
       name :
        - httpd
        - firewalld
        - vim
        - wget
        - yp-tools
       state : latest
    - name : Defining INDEX.html file
      copy :
       content : "Welcome to my Ansible Training"
```

```
    dest : /var/www/html/index.html
  - name : Ensure Firewall Service is Started & Enabled
    service :
      name : firewalld
      state : started
      enabled : true
  - name : Adding Port to the Firewall Rule
    firewalld :
      port : 80/tcp
      permanent : true
      state : enabled
      immediate : yes
  - name : Ensuring Apache Service is Running & Enabled
    service : name=httpd state=started enabled=true
- name : Creating Users from Selected Group Memebers
  hosts : db
  tasks :
    - name : This task is to create user
      user :
        name : clouduser
        comment : userforCLOUD
        uid : 6798
        groups : aws,azure,gcp
        append : yes
```

shell : /bin/sh

generate_ssh_key : true

ssh_key_bits : 2048

ssh_key_file : .ssh/id_rsa

expires : -1

state : present

:wq!