# Ansible Scripting - Writing Playbooks in Ansible

Ansible playbooks are scripts containing ansible code in files. It is a way of performing complex automation tasks. It is similiar to a shell script which has features like variables, templates, loops, if-else.

Ansible Playbooks are written in a specific markup language which is YAML. YAML is similiar to that of XML, JSON but with a strict liking of indentation.

All playbooks are written in YAML and has .yml extension.

## Components of Playbook

To understand the components of a playbook lets see a sample playbook -

---

- hosts: web tasks:

- name: Install Nginx

apt: pkg=nginx state=installed update\_cache=true

Every playbook starts with **---** 3 dashes

**hosts** keyword in the playbook is used to mention the target host group. This will be read from the inventory file

**tasks** keyword will signify that start of the actual task we would like to perform. It has many sub-parts. In above example, **name** and **apt** are falling under tasks.

**name** keyword is used to mention the name of the task. It is readble plain english and must be clearly headlining the action we are trying to acheive.

**apt** is an ansible module which is used to install the packages

**pkg** is an argument required for apt module which will tell which package is to be installed

**state and update\_cache** are other two arguments passed with apt module

## Executing an ansbile playbook

Below command is used to execute an ansbile playbook -

ansible-playbook <playbook-name>

For example -

ansible-playbook vim-install.yml

# Frequently used modules of Ansible

Ansible has special functions/modules using which automation can be performed. Let us first understand the major routine works done by System administrtor - a. Installing packages b. Provisioning Servers c. working with files, directories d. working with services e. Exeucting commands using ansible, etc.

## Install packages using apt module

apt module is used to install packages in Ubuntu/Debian based OS. A sample playbook is given below

---

- hosts: localhost tasks:

- name: Install vim editor

apt: pkg=vim state=present update\_cache=true

## Install pakages using yum module

yum module is used to install packages in CentOS. A sample playbooks is given below -

---

- hosts: web tasks:

- name: Install vim editory

yum: pkg=vim state=present update\_cache=yes

## Deal with files using Ansible

Ansible file modules can be used to deal with file operations like creation deleting, adding file contents, etc.

---

- hosts: all tasks:

- name: Creates a file if it does not exist file:

path: "/home/aashay/Documents/devops-schedule.txt" state: touch

The above playbooks looks for a file an in case it is not found, it creates it.**state: touch**\_ ensures that it creates a file

---

- hosts: all tasks:

- name: Create a file with some content copy:

dest: "/home/aashay/Documents/devops-schedule.txt" content: |

dog1 tiger

the above playbook will create a file with content mentioned in the content section of playbook

---

- hosts: all tasks:

- name: Create our directory and manage permissions file: path=/var/www/codereview

state=directory owner={{user\_name}} group={{user\_name}} mode=0700

The above playbooks crreates a directory and changes its owner, group and permissions

## Ansible copy module

Ansible copy module is used to copy files to remote desitnations. the playbook is self explanative where it uses copy module of ansbile and pass parameters for source and destination

Snippet for the same is given.

- name: Copy our 10mb file to our server being managed by Ansible

copy: src=10mb-file.bin dest=/home/codereview/10mb-file.bin

## Ansible find modules to find files

Ansibe find module can be used to find files in the target hosts

---

- hosts: all tasks:

- name: Ansible find file examples find:

paths: /tmp patterns: "\*.txt"

register: files\_matched

- debug:

msg: "{{ files\_matched.files }}"

the above playbook uses find module to mention the path and pattern we want to use to match files. Also it uses the following two modules - a. **register** register module is used to save the output of he execution result of the playbook. In the

above given example, register will hold the name of all the files which were found using the pattern \*.txt

b. **debug** module is used to print the variable which stores the names of the files found. The value is stored in the variable mentioned in the register module

## Ansbile service module to deal with services

You can deal with service status like that of apache2, nginx, sshd, docker, etc.

The playbook given below exaplains a sample usage of the service module

- name: stop nginx service service: name=nginx state=stopped

the above playbook ensures that the service nginx on the remote host is stopped after the execution.

## Ansible Shell module to execute shell commands

Ansible shell module is used to execute shell command in linux systems The example playbooks is given below -

* name: Executing a Command Using Shell Module of Ansible shell: ls -lrt > temp.-file.txt
* name: Executing a command using command module shell: cat hello.txt

the above playbook will execute shell command mentioned in the **shell**

## Ansible command module to execute commands on linux

- name: Executing a command using command module command: cat hello.txt

**Note - difference between Shell and command** Shell and command modules can be used to execute commands on linux. In shell module one can mention the type of shell which they want to use to execute command. This option is not available in the shell.

The sample playbook to demonstrate the same is given below -

- hosts: loc tasks:

- name: Ansible Shell chdir and executable parameters shell: ls -lrt > temp.txt

args:

chdir: /root/ansible/shell\_chdir\_example executable: /bin/bash

The **args** section mention the added parameters to change the directory first and then use **/bin/bash/**