Ansible Tutorial Part 2 – Implementing Playbooks

Ansible Playbooks and Ad-Hoc commands

Ad-Hoc commands can run a single, simple task against a set of targeted hosts as a one-time command. The real power of Ansible however is in learning how to use playbooks to run multiple, complex tasks against a set of targeted hosts in an easily repeatable manner.

A Play in an ordered set of tasks run against hosts selected from your inventory. A Playbook is a text file containing a list of one or more plays to run in a specific order.

Plays allow to you to change a lengthy complex set of manual administrative tasks into an easily repeatable routine with predictable and successful outcomes. In a playbook, you can save the sequence of tasks in a play into a human-readable and immediately runnable form. The tasks themselves, because of the way in which they are written, document the steps needed to deploy your application or infrastructure.

Below are few important points –

- Playbooks are written in YAML language (Yet Another Markup Language)
- YAML is a data structure representation language
- Playbooks will have extensions as .yml or .yaml
- We run a playbook with command **ansible-playbook <Playbook Name>**
- For syntax checking we use **ansible-playbook – -syntax-check<Playbook Name>**
- For Dry run we use **ansible-playbook <Playbook Name> -C or ansible-playbook <Playbook Name> – -check**
- A playbook begins with three dashes (—) and should end with three dots (…)
- In between those markers, the playbook is defined as a list of plays. An item in a YAML list starts with a single dash followed by a space.

For example a YAML list might appear as follows –

```
– Apple
– Orange
– Grape
```

- Maintaining Indentation in a playbook is very important.
- A play is a collection of Key-Value pairs. Keys in the same play should have same indentation.
- A Playbook has three keys – **name**, **hosts** and **tasks**. These keys should have same indentation.

Example –

```
—
– name: Example
  hosts: webservers
  tasks:
    – name: First Task
      Module:
        – Attributes

    – name: Second Task
```

Module:
– Attributes

- The first line of the example play starts with a dash and a space (indicating the play is the first item of a list) and then the first key , the **name** attribute . The name key associates an arbitrary string with the play as a label. This identifies what the play is for. The name key is optional but is recommended because it helps to document your playbook. This is especially useful when a playbook contains multiple plays.
- The second key in the play is **hosts** attribute which specifies the hosts against which the play's tasks are run.
- Finally the last key in the play is the **tasks** attribute whose value specifies a list of tasks to run for this play.

A Sample Playbook

```
—
– name: Create a user john with UID 4000
  hosts: servera.lab.example.com
  tasks:
– name: John user exists with UID 4000
 user:
name: John
uid: 4000
state: present
```

Output Verbosity of Playbook Execution

-v à The task results are displayed
-vv à Both task results and tasks configurations are displayed
-vvv à Includes information about connection to managed hosts
-vvvv à Adds extra verbosity options to the connection plug-ins, including users being used in the managed hosts to execute scripts, and what scripts have been executed.

Lab 1

1. Write a playbook to install httpd package in all managed hosts
2. The local file files/index.html should be copied to /var/www/html/index.html in the managed hosts
3. The httpd service should be started and enabled

Lab 2

Write a playbook to define 2 plays –

- First play should install httpd package, copied index.html to managed hosts , start and enable httpd daemon and allow httpd service through firewall
- Second play should contain a task to test intranet web server