Ansible – PLAYBOOK YAML Explained

VISHWANATH M S

VISHWACLOUDLAB.ORG

What is YAML?

YAML, which stands for Yet Another Markup Language or
YAML Ain’t Markup Language (depending who you ask)

Using YAML for definitions gives you a number of advantages, including:

- **Convenience:** You’ll no longer have to add all of your parameters to the command line
- **Maintenance:** YAML files can be added to source control, so you can track changes
- **Flexibility:** You’ll be able to create much more complex structures using YAML than you can on the command line

What is YAML?

- The YAML is a scripting language, means we can communicate with other languages using yaml.
- Strictly speaking YAML is a superset of JSON with additional features like **new line** and **indentation**.

- YAML is a case sensitive scripting language
- YAML does not allow the use of **tabs for indentation like python.**
- Alternatively **space** is used for indentation.
- There are three editions in YAML scripting:

What is YAML?

- There are three editions in YAML scripting:
  - 1.2 → third edition
  - 1.1 → second edition
  - 1.0 → first edition

- YAML script extension:
  - .yaml
  - .yml

What is Data Types?

- Data Types also called as Key.
- Key is used to store any value
- Value can change depending on condition
- Example:
  - xyz:340
  - test_int: 59
  - testname: “vishwacloudlab”
  - test_name: vishwacloudlab
  - testfloat: 39.0
  - testboolean: true
  - null_value: null

Data Types Continued…

- These type of Data collections are called Scalar representation of data.
- Theses are rarely used in real time
- Commonly used is Multiple Key value pair
- Two types are
  - Sequential Data Collection.
  - Map data Collection

Data Types - Sequential Data Collection.

Sequential Data collection are also called YAML lists.

Example1 representation:

```
Chess players:
- player1
- “player2”
- player3
```

Data Types - Map Data Collection.

Map Data collection are also called YAML MAPS.

Example2 representation:

```
Chess_players_age:
- player1: 56
- player2: 33
- player3: 42
```

Example3 representation:

```
Chess_players_Details:
    player1:
        - Expert Level
        - age: 56
    player2:
        - Beginner Level
        - age: 33
    player3:
        - Legend Level
        - age:42
```

Data Types Conclusion:

There are only **two** types of structures you need to know about in YAML:

- Lists
- Maps

That’s it. You might have maps of lists and lists of maps and so on……

Review on YAML

- Maps, which are groups of name-value pairs
- Lists, which are individual items
- Maps of maps
- Maps of lists
- Lists of lists
- Lists of maps

Basic Steps to write playbook

1) Starts with --- → This Represents the beginning of the script
2) Target selection list (Like hosts, user etc)
3) Variable List (optional)
4) Tasks list
    1) List all the modules that needs to run in the particular order

Note: These are steps for one play

Each play is a sequence and sequence values have maps.

YAML Playbook example

```
---
- name: Install httpd
  hosts: web
  become: true

  tasks:
  - name: Install httpd on web server
    yum:
      name: httpd
      state: present
  - name: Insert index page
    template:
      src: index.html
      dest: /var/www/html/index.html
  - name: start the httpd service
    service:
      name: httpd
      state: started
```

Now lets understand the playbook in parts

Let’s look at each piece closer

```
---
- name: Install httpd
  hosts: web
  become: true
  tasks:
  - name: Install httpd
    yum:
      name: httpd
      state: present
```

Delimiter between sets of YAML script

Description

Define the hosts

Run as root

POINTS TO REMEMBER FOR YAML SCRIPTS

Indenting the lines are important
Min is “1” space
Indenting should be CONSISTENT

*****NEVER USE TABS in a YAML file*****

Let’s look at each piece closer

```
---
- name: Install httpd
  hosts: web
  become: true
  tasks:
  - name: Install httpd on web server
    yum:
      name: httpd
      state: present
```

Define tasks to be executed

Yum module

Name of the App to be installed

Present , latest, absent

Let's look at each piece closer

```
tasks:
  - name: Install httpd on web server
    yum:
      name: httpd
      state: present
  - name: Insert index page
    template:
      src: index.html
      dest: /var/www/html/index.html
  - name: start the httpd service
    service:
      name: httpd
      state: started
```

This would copy the file from the Ansible server to the destination Host in the particular folder

Let’s look at each piece closer

```
tasks:

  ……….. .
  - name: Insert index page
    template:
      src: index.html
      dest: /var/www/html/index.html
  - name: start the httpd service
    service:
      name: httpd
      state: started
```

Service module in linux, to start, enable, stop , restart the app/service.

Running the Ansible Playbook

```
$ ansible-playbook <filename.yml>
```

Now that we have learned the playbook to be executed on a single group of host.

Let's further understand a complex playbook with multiple group and failure conditions.

Tasks and Play

Example

```
---
- name: Create a file
  hosts: app_server
  tasks:
  - command: touch file1.txt

- name: install web server
  hosts: web
  tasks:
  - yum:
      name: httpd
      state: present
```

Tasks

Play - 1

Play - 2

Questions……….