Ansible-Day-4

13 December 2023 13:07

==========

LOOP in ansible

==========

- loop keyword used to execute the task multiple times.
  Example:
  - Creating multiple users with user module
  - Changing the ownership for multiple files

- Just few versions before instead of loop, with_items was used to support loop functionality & now its depricated.

- **usecase1: create testuser1,testuser2 & testuser3 using loop in the single task**

```
---
- hosts: all
  become: yes
  gather_facts: no
  tasks:
    - name: "Creating user"
      user:
        name: "{{ item }}"
        state: present
      loop:
        - testuser1
        - testuser2
        - testuser3
```

- **usecase2: create 3 users(testuser1,testuser2 & testuser3) in three different groups(dev,test,prod) using loops**

```
---
- hosts: all
  become: yes
  gather_facts: no
  tasks:
    - name: "Creating user"
      user:
        name: {{ item.user }}
        group: {{ item.group }}
        state: present
      loop:
        - { name: 'testuser1', groups: 'dev' }
        - { name: 'testuser2', groups: 'test' }
        - { name: 'testuser3', groups: 'prod' }
```

- **usecase3: Install multiple packages using loop {git,tree,wget}**

==========

When condition

==========

- "When" statement is a conditional statement that runs the particular task if the condition is met.

- **usecase1:- Create a playbook install apache webserver setup in Linux{httpd} & Ubuntu{apache2} machines**

```
---
- hosts: all
  become: yes
  gather_facts: no
  tasks:
    - name: "Installing httpd"
      yum:
        - name: httpd
          state: present

    - name: "Start httpd service"
      service:
        name: httpd
        state: started
```

```
- name: "Configure httpd"
  copy:
    src: index.html
    dest: /var/www/html/index.html

- name: "Installing apache2"
  apt:
    name: apache2
    state: present

- name: "start apache2 service"
  service:
    name: apache2
    state: started

- name: "configure apache2"
  copy:
    src: index.html
    dest: /var/www/html/index.html
```

► Once I execute above playbook all the tasks will run in sequential mode & execution failed on all ubuntu & rhel servers.
To overcome this is issue I am going to use when condition particular task, so when condition met then only respective task will executed.

► Capture OS family details,
ansible -m setup -a "filter=*os*" all

gather_facts set to be true.

```
---
- hosts: all
  become: yes
  gather_facts: yes
  tasks:
    - name: "Installing httpd"
      yum:
        - name: httpd
          state: present
      when: ansible_os_family == "RedHat"

    - name: "Start httpd service"
      service:
        name: httpd
        state: started
      when: ansible_os_family == "RedHat"

    - name: "Configure httpd"
      copy:
        src: index.html
        dest: /var/www/html/index.html
      when: ansible_os_family == "RedHat"

    - name: "Installing apache2"
      apt:
        name: apache2
        state: present
      when: ansible_os_family == "Ubuntu"

    - name: "start apache2 service"
      service:
        name: apache2
        state: started
      when: ansible_os_family == "Ubuntu"

    - name: "configure apache2"
      copy:
        src: index.html
        dest: /var/www/html/index.html
      when: ansible_os_family == "Ubuntu"
```

==========

Grouping of Inventory

==========

==========

Handlers

==========

- It's a feature of ansible, appears similar to the task.
- Handlers will works only when it's called from another task using notify keyword.
- When there is a need of running repetitive task after running particular task then we will use handlers.
- Assume we have Apache installed & configured, after the configuration normally we required to restart apache
- In this case we will use handlers.
- **Usecase4:- Create playbook Apache setup using handlers**

```
---
- hosts: all
  become: yes
  gather_facts: no
  tasks:
    - name: "Installing httpd"
      yum:
        name: httpd
        state: present

    - name: "Configuring httpd"
      copy:
        src: index.html
        dest: /var/www/html/index.html
      notify: Restart apache

  handlers:
    - name: "Restart apache"
      service:
        name: httpd
        state: restarted
```