Ansible-Day-3

13 December 2023 12:05

==========

Ansible modules

==========

- Modules are predefined commands/units which are used inside the playbooks or commands in order to execute specific tasks on target machines.
- Ansible have large extent number of modules, we can view it on ansible community side.
- Explain user module & required parameters

==========

Ansible playbook syntax

==========

- Ansible playbook is a text file written in YAML(Ain't markup language) & stored with format .yml
- The sample format of that ansible playbook appears as

```
---
- hosts: all
  become: yes
  gather_facts: yes
  tasks:
    - task1
    - task2
    - task3
```

Where

- --- Indicates the starting of the playbook
- - hosts: Points the list of hostnames that we are targeting to configure.
- gather_facts: Capture the facts about the target machines
- become: The ansible user becomes the root.
- tasks: List of the activities that we are going to do.
- task1: activity that we are going to do

```
Ex:
- name: "Installing git"
  yum:
    name: git
    state: present
```

- **Create a playbook for userid creation**

```
---
- hosts: all
  become: yes
  gather_facts: yes
  tasks:
    user:
      name: dhoni
      state: present
```

- ansible-playbook -i ~/hosts -v user_creation.yml
  The output appears in the following format

```
 PLAY RECAP
*********************************************************************************************************************************
*********************************************************************************
 client-0x0001          : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

- changed ==> The state of the target machine modified after executing particular task.
- ok ==> means there was no change on target machine state after executing particular task.

- unreachable ==> Not able to connect target machine, try with wrong ip address
- failed ==> Specific task failed due to some errors , try with error wrong package name
- skipped ==> Particular task skipped to run on specific host machine, run a task only on Ubuntu {Discuss during conditional statements}

```
tasks:
  - name: "Install git on rhel"
    yum:
      name: git
      state: present
    when: ansible_os_family == rhel8

  - name: "Install tree on ubuntu"
    apt:
      name: tree
      state: present
    when: ansible_os_family == ubuntu
```

rescued: Need to work on this. {Discuss during conditional statements}

Explain every line ansible-day-3.png image

- The playbooks uses the indentation with spaces in order to indicate structure of the data.

```
---
- hosts: all
  become: yes
tasks:
    - user:
        name: john
        state: present
```

Since we didn't followed indentation in right format we received syntax errors.

- gather_facts ==>
  - Used to fetch target machine meta data details/facts
    Example:
    "ansible_nodename": "ip-172-31-21-77.ec2.internal",
    "ansible_os_family": "RedHat",
    "ansible_pkg_mgr": "yum"
  - By default gather_facts are enabled
  - When we don't have any need of facts we can disable it by setting gather_facts to no

```
---
- hosts: all
  become: yes
  gather_facts: no
  tasks:
    - user:
        name: john
        state: present
```

- **become**
  - This used to get privilege escalation option & converts ansible user to get sudo privilege's,
  - similarly that we used "-b" option in ad-hoc commands during group creation.
- **UC:- Create a user with become no option.**

```
---
- hosts: all
  gather_facts: no
  become: no
```

```
tasks:
 - name: "create user"
   user:
    name: john
    state: present
```

- ansible-playbook -i ~/hosts -v user_creation.yml
- The execution will get failed due to the permissions issue & we can solve it by setting become to yes.
- By default become is set to no, if we don't use become option in our playbooks.

► **--check**:

- When ansible-playbook is executed with --check it will not make any changes on remote systems.
  It will display report what changes they would have made rather than making them.
- **UC:- Create a playbook for kohli user creation & before execute the playbook run it on check mode**
  On target machine we don't see the kohli user after mode, since mode id dry run,

It just report user creation changes is going to happen

```
---
- hosts: all
  become: yes
  gather_facts: no
  tasks:
    - name: "create a user"
      user:
        name: kohli
        state: present
```

- ansible-playbook -i ~/hosts -v user_creation.yml --check
  Even we have executed our playbook, it hasn't created kohli user, It's just reported the changes what are all going happen.
- Remove the --check option & execute playbook, now kohli user added on target system.
  ansible-playbook -i ~/hosts -v user_creation.yml

► **UC1:- Create a playbook for installing "git" package on ubuntu & rhel machine.**

```
---
- hosts: all
  gather_facts: no
  become: yes
  tasks:
        - name: "Install git"
          yum:
                name: git
                state: present
```

- ansible-playbook -i ~/hosts -v git_install.yml
- The execution will be failed due to By default
  "yum" will acts package repo for RHEL machines &
  "apt" will acts package repo for Ubuntu machines.
- To overcome this issue we use "package" module to support installation of packages on any Linux flavours when package name same for all flavours.

```
---
- hosts: all
  gather_facts: no
  become: yes
  tasks:
    - name "Installing git"
      package:
        name: git
        state: present
```

ansible-playbook -i ~/hosts -v git_install.yml

- **UC2:- Playbook for Install apache(httpd) webserver, start it & later configure it**

```
yum install httpd
service httpd start
echo "Welcome to ansible" > /var/www/html/index.html
```

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

    - name: "starting httpd service"
      service:
        name: httpd
        state: started

    - name: "Configure the httpd service"
      command: echo "Welcome to ansible" > /var/www/html/index.html
```

ansible-playbook -i ~/hosts -v httpd_setup.yml

- Grouping & subgrouping of servers

- LAMP stack setup with installation commands & customize it with loop