

AdHoc commands(shell, command, raw), copy (src, content), play, playbook

Command-Shell-Raw difference

Command modulu shell-in yeni bash-in xüsusiyyətlərindən istifadə edə bilmir. Məsələn environment variable, pipe, redirection. Əgər bunlardan istifadə edəcəyiksə shell modulundan istifadə etmək lazımdır. Command və Shell-də manage etdiyimiz hostların hamısında python yüklü olmalıdır. Rawda isə python olmasa belə işləyəcək.

```
ansible prod -m copy -a 'src=/root/ansible/engineer dest=/opt/engineer'
```

```
[root@master ansible]# ansible prod -m copy -a 'src=/root/ansible/engineer dest=/opt/engineer'
192.168.149.136 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "checksum": "7b3f2afd5722fe3afa73df44721ca9ebafc734db",
  "dest": "/opt/engineer",
  "gid": 0,
  "group": "root",
  "md5sum": "2cef5b3b49d31c28f0baeab85ff31108",
  "mode": "0644",
  "owner": "root",
  "size": 6,
  "src": "/root/.ansible/tmp/ansible-tmp-1713800032.350957-31175-260252128241304/source",
  "state": "file",
  "uid": 0
}
```

copy modulundan istifadə edərək öz hostundakı /root/ansible/engineer faylını manage etdiyimiz prod qrupundakı /opt/engineer at

```
ansible prod -m file -a 'src=/usr/lib/systemd/system/multi-user.target dest = /etc /systemd/
system/default.target state=link'
```

```
[root@master ansible]# ansible prod -m file -a 'src=/usr/lib/systemd/system/multi-user.target dest=/etc/sys
temd/system/default.target state=link'
192.168.149.136 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "dest": "/etc/systemd/system/default.target",
  "gid": 0,
  "group": "root",
  "mode": "0777",
  "owner": "root",
  "size": 41,
  "src": "/usr/lib/systemd/system/multi-user.target",
  "state": "link",
  "uid": 0
}
```

prod qrupunda file modulundan istifadə edərək /usr/lib/systemd/system/multi-user.target faylını /etc /systemd/system/default .target -ə linklə

```
ansible prod -m service -a 'name=httpd state=started'
```

```
[root@master ansible]# ansible prod -m service -a 'name=httpd state=started'
192.168.149.136 | FAILED! => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "msg": "Unable to start service httpd: Job for httpd.service failed because the control process exited
with error code.\nSee \"systemctl status httpd.service\" and \"journalctl -xeu httpd.service\" for details.
\n"
}
```

prod qrupunda service modulundan istifadə edərək adı httpd olan servisi start et

```
ansible prod -m shell -a 'rpm -qa | grep nano'
```

```
[root@master ansible]# ansible prod -m shell -a 'rpm -qa | grep nano'
192.168.149.136 | FAILED | rc=1 >>
non-zero return code
```

shell modulundan istifadə edərək prod qrupunda nano paketinin olub-olmadığını yoxla

```
ansile prod -m yum_repository -a 'name=test baseurl=anylink description=anydescribtion
gpgcheck=any gpgkey='asfhosaf' enabled=no '
```

```
[root@master ansible]# ansible prod -m yum_repository -a 'name=test baseurl=https://download.fedoraproject.
org/pub/epel/$releasever/$basearch/ description="TEST REPO" gpgcheck=yes gpgkey="kjelfllldflf" enabled=no'
192.168.149.136 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "repo": "test",
  "state": "present"
}
```

yum_repository modulunda istifadə edərək prod qrupunda verilən xüsusiyyətlərdə repo qur.

Playbook

```
vim ali.yml
```

```
---
- name: play1
  hosts: all
  tasks:
    - name: ping et
      ping:

    - name: debug mesajı goster
      debug:
        msg: echo salam
```

```
ansible-playbook ali.yml --syntax-check (syntax -ı yoxlayır)
```

```
ansible-playbook ali.yml --check (simulasiya edir)
```

```
vim ali.yml
```

```
---
- name: play1
  hosts: all
  gather_facts: no
  tasks:
    - name: ping et
      ping:

    - name: debug mesajı goster
      debug:
        msg: echo salam
```

```
gather-facts: no --(node-lərdə factları toplamağın qarşısını alır)
```

vim ali.yml

```
- name: play2
  hosts: prod
  gather_facts: no
  tasks:
    - name: user yarat
      user:
        name: riyad
        shell: /sbin/nologin
        uid: 2457
    - name: user haqqında info ver
      command: .
        - id riyad
```

Prod qrupunda riyad adında, shell-i /sbin/nologin, uid-si 2457 olan user qur. Daha sonra riyad useri üçün id commandını run et.