

# 第1章 Ansible Role角色介绍

## 1.为什么需要使用角色

1. 不太灵活,臃肿
2. 全部写在一起,修改不方便
3. 配置文件随便放,不标准

## 2.角色解决了什么问题

1. 把剧本 拆分 拆分 拆分
2. 解耦,结构更清晰,调试更方便

## 3.编写角色的最佳实践

1. 初级阶段,不要直接写角色,先写好剧本,然后再拆分
2. 一开始不要想一步到位,不用拆的很细,尤其是变量

# 第2章 角色目录规划

## 0.官方说明

- 1 [https://docs.ansible.com/ansible/latest/user\\_guide/playbooks\\_reuse\\_roles.html](https://docs.ansible.com/ansible/latest/user_guide/playbooks_reuse_roles.html)

## 1.目录说明

注意! 这里的目录结构必须按照官方定义的要求来做! 不是自己随便乱起!

```
1  site.yml          #调用总入口
2  nfs_servers.yml   #角色
3  rsync_servers.yml #角色
4  roles/            #角色具体的目录
5      nfs_servers/
6          tasks/      #剧本里的任务
7          handlers/   #剧本里的handlers
8          files/      #存放需要拷贝的文件或压缩包
9          templates/  #存放配置文件的目录
10         vars/        #存放变量
11     rsync_servers/
12         tasks/      #剧本里的任务
13         handlers/   #剧本里的handlers
14         files/      #存放需要拷贝的文件或压缩包
15         templates/  #存放配置文件的目录
16         vars/        #存放变量
```

# 第3章 编写rsync角色

## 0.编写思路

- 1.先写好剧本
- 2.创建角色目录
- 3.拷贝需要发送的文件到指定目录
- 4.拆分剧本

## 1.编写剧本

```
1 - hosts: backup
2   vars:
3     user_id: '666'
4     rsync_user: 'www'
5
6   tasks:
7     #1.创建www组和www用户
8     - name: 01-create_group
9       group:
10        name: "{{ rsync_user }}"
11        gid: "{{ user_id }}"
12
13    #2.创建www用户
14    - name: 02-create_user
15      user:
16        name: "{{ rsync_user }}"
17        uid: "{{ user_id }}"
18        group: "{{ rsync_user }}"
19        create_home: no
20        shell: /sbin/nologin
21
22    #3.创建数据目录并更改授权
23    - name: 03-create_data
24      file:
25        path: "{{ item }}"
26        state: directory
27        owner: "{{ rsync_user }}"
28        group: "{{ rsync_user }}"
29        mode: '755'
30      loop:
31        - /data/
32        - /backup/
33
34    #4.安装rsync软件
35    - name: 04-install_rsync
36      yum:
37        name: rsync
38        state: latest
39
40    #5.复制配置文件和密码文件
41    - name: 05-copy_pwd&conf
42      copy:
43        src: "{{ item.src }}"
44        dest: /etc/
45        mode: "{{ item.mode }}"
46      notify:
47        - restart rsyncd
```

```

48     loop:
49         - { src: /root/script/rsync/rsyncd.conf, mode: '644'}
50         - { src: /root/script/rsync/rsync.passwd, mode: '600'}
51
52     #6.启动服务
53     - name: 06-start
54       systemd:
55         name: rsyncd
56         state: started
57         enabled: yes
58
59     #7.重启服务
60     handlers:
61         - name: restart rsyncd
62           systemd:
63             name: rsyncd
64             state: restarted

```

## 2.创建角色目录

```

1 [root@m-61 ~]# cd /etc/ansible/roles/
2 [root@m-61 /etc/ansible/roles]# mkdir
  rsync_server/{tasks,handlers,files,templates,vars} -p
3 [root@m-61 /etc/ansible/roles]# tree rsync_server/
4 rsync_server/
5 |— files
6 |— handlers
7 |— tasks
8 |— templates
9 |— vars

```

## 3.把剧本复制到tasks目录

```

1 |— tasks
2 |   |— main.yaml

```

## 4.把配置文件复制到file目录

```

1 cp script/rsync/* /etc/ansible/roles/rsync_server/files/

```

## 5.拆分handlers

```

1 [root@m-61 ~]# cat /etc/ansible/roles/rsync_server/handlers/main.yaml
2 - name: restart rsyncd
3   systemd:
4     name: rsyncd
5     state: restarted

```

## 6.拆分vars

```
1 [root@m-61 ~]# cat /etc/ansible/roles/rsync_server/vars/main.yaml
2 user_id: '666'
3 rsync_user: 'www'
```

## 7.精简tasks任务文件

```
1 [root@m-61 ~]# cat /etc/ansible/roles/rsync_server/tasks/main.yaml
2 #1.创建www组和www用户
3 - name: 01-create_group
4   group:
5     name: "{{ rsync_user }}"
6     gid: "{{ user_id }}"
7
8 #2.创建www用户
9 - name: 02-create_user
10  user:
11    name: "{{ rsync_user }}"
12    uid: "{{ user_id }}"
13    group: "{{ rsync_user }}"
14    create_home: no
15    shell: /sbin/nologin
16
17 #3.创建数据目录并更改授权
18 - name: 03-create_data
19   file:
20     path: "{{ item }}"
21     state: directory
22     owner: "{{ rsync_user }}"
23     group: "{{ rsync_user }}"
24     mode: '755'
25   loop:
26     - /data/
27     - /backup/
28
29 #4.安装rsync软件
30 - name: install_rsync
31   yum:
32     name: 04-rsync
33     state: latest
34
35 #5.复制配置文件和密码文件
36 - name: 05-copy_pwd&conf
37   copy:
38     src: "{{ item.src }}"
39     dest: /etc/
40     mode: "{{ item.mode }}"
41   notify:
42     - restart rsyncd
43   loop:
44     - { src: rsyncd.conf, mode: '644'}
45     - { src: rsync.passwd, mode: '600'}
46
47 #6.启动服务
48 - name: start
49   systemd:
50     name: rsyncd
```

```
51 | state: started
52 | enabled: yes
```

## 8.编写调用文件

```
1 | [root@m-61 ~]# cat /etc/ansible/rsync_server.yaml
2 | - hosts: rsync_server
3 |   roles:
4 |     - rsync_server
```

## 9.编写主机清单

```
1 | [root@m-61 ~]# cat /etc/ansible/hosts
2 | [rsync_server]
3 | 172.16.1.41
```

## 10.调试运行

```
1 | cd /etc/ansible/
2 | ansible-playbook -C rsync_server.yaml
3 | ansible-playbook rsync_server.yaml
```

# 第4章 编写sshd角色

## 1.编写思路

```
1 | 1.先拷贝配置文件到template目录下并重命名为j2
2 | 2.编写tasks文件
3 | 3.调试运行
```

## 2.创建角色目录

```
1 | cd /etc/ansible/roles/
2 | mkdir sshd/{tasks,handlers,files,templates,vars} -p
```

## 3.编写jinja模版文件

jinja模板注意：

```
1 | 1.模块必须是template
2 | 2.模版文件必须以.j2结尾
3 | 3.模版文件必须放在template目录下
```

关键配置：

```
1 | #复制sshd配置文件到template文件夹下
2 | Port {{ ssh_port }}
3 | ListenAddress {{ ansible_facts.eth1.ipv4.address }}
```

## 4.编写变量文件

```
1 [root@m-61 /etc/ansible/roles/sshd]# cat vars/main.yml
2 ssh_port: '22'
```

## 5.编写handlers文件

```
1 [root@m-61 /etc/ansible/roles/sshd]# cat handlers/main.yml
2 - name: restart sshd
3   systemd:
4     name: sshd
5     state: restarted
```

## 6.编写主任务文件

```
1 [root@m-61 /etc/ansible/roles/sshd]# cat tasks/main.yml
2 #1.复制配置文件和密码文件
3 - name: 01_copy_sshd
4   template:
5     src: sshd_config.j2
6     dest: /etc/ssh/sshd_config
7     mode: '600'
8     backup: yes
9   notify:
10    - restart sshd
11
12 #2.启动服务
13 - name: start
14   systemd:
15     name: sshd
16     state: started
17     enabled: yes
```

## 7.查看最终的目录

```
1 [root@m-61 /etc/ansible/roles]# tree sshd/
2 sshd/
3 ├── files
4 ├── handlers
5 |   └── main.yml
6 ├── tasks
7 |   └── main.yml
8 ├── templates
9 |   └── sshd_config.j2
10 └── vars
11     └── main.yml
```

## 8.编写主调用文件

```
1 [root@m-61 /etc/ansible/roles]# cat ../sshd.yaml
2 - hosts: ssh
3   roles:
4     - sshd
```

## 第5章 编写nfs角色

### 1.编写思路

```
1 1.先拷贝配置文件到template目录下并重命名为j2
2 2.编写handlers
3 3.编写tasks
```

### 2.创建角色目录

```
1 cd /etc/ansible/roles/
2 mkdir nfs_server/{tasks,handlers,files,templates,vars} -p
```

### 3.编写jinja模版文件

```
1 [root@m-61 ~]# cat /etc/ansible/roles/nfs_server/templates/exports.j2
2 /data 172.16.1.0/24(rw,sync,all_squash,anonuid=666,anongid=666)
```

### 4.编写handlers文件

```
1 [root@m-61 ~]# cat /etc/ansible/roles/nfs_server/handlers/main.yaml
2 - name: restart nfs
3   systemd:
4     name: nfs
5     state: restarted
```

### 5.编写主任务文件

```
1 [root@m-61 ~]# cat /etc/ansible/roles/nfs_server/tasks/main.yaml
2 #1.创建www组和www用户
3 - name: create_group
4   group:
5     name: www
6     gid: 666
7
8 #2.创建www用户
9 - name: create_user
10  user:
11    name: www
12    uid: 666
13    group: www
14    create_home: no
15    shell: /sbin/nologin
16
17 #3.创建数据目录并更改授权
18 - name: create_data
```

```
19 file:
20     path: "{{ item }}"
21     state: directory
22     owner: www
23     group: www
24     mode: '755'
25 loop:
26     - /data/
27     - /backup/
28
29 #4.安装nfs软件
30 - name: install_nfs
31   yum:
32     name: nfs-utils
33     state: latest
34
35 #5.复制配置文件和密码文件
36 - name: copy_exports
37   template:
38     src: exports.j2
39     dest: /etc/exports
40   notify:
41     - restart nfs
42
43 #6.启动服务
44 - name: start
45   systemd:
46     name: nfs
47     state: started
48     enabled: yes
```

## 6.编写调用文件

```
1 [root@m-61 ~]# cat /etc/ansible/nfs_server.yaml
2 - hosts: nfs
3   roles:
4     - nfs_server
```

## 第六章 编写lsyncd服务

```
1
```

## 第6章 拆分init角色

### 0.编写思路

- 1.先分析以前写过所有的角色里重复的操作
- 2.把重复的操作内容单独写一个角色，例如：init
- 3.先备份一份以前写好的角色文件
- 4.精简以前的角色文件，删除重复的内容
- 5.调试，运行，检查

### 1.找出重复的操作



- 1 1.创建www组和www用户
- 2 2.创建www用户
- 3 3.创建数据目录并更改授权
- 4 4.安装rsync软件
- 5 4.安装nfs软件

## 2.创建角色目录

- 1 cd /etc/ansible/roles/
- 2 mkdir init/{tasks,handlers,files,templates,vars} -p

## 3.编写jinja模版文件

1 |

## 4.编写handlers文件

1 |

## 5.编写主任务文件

```
1 [root@m-61 /etc/ansible]# cat /etc/ansible/roles/init/tasks/main.yaml
2 #1.创建www组和www用户
3 - name: create_group
4   group:
5     name: www
6     gid: 666
7
8 #2.创建www用户
9 - name: create_user
10  user:
11    name: www
12    uid: 666
13    group: www
14    create_home: no
15    shell: /sbin/nologin
16
17 #3.创建数据目录并更改授权
18 - name: create_data
19   file:
20     path: "{{ item }}"
21     state: directory
22     owner: www
23     group: www
24     mode: '755'
25   loop:
26     - /data/
27     - /backup/
28
29 #4.安装nfs软件
30 - name: install_soft
31   yum:
32     name: "{{ item }}"
```

```
33     state: latest
34     loop:
35         - rsync
36         - nfs-utils
```

## 第八章 拆分后的各个服务角色文件

### 1. 拆分后的rsync角色

```
1 [root@m-61 ~]# cat /etc/ansible/roles/rsync_server/tasks/main.yml
2 #1.复制配置文件和密码文件
3 - name: copy pwd&conf
4   copy:
5     src: "{{ item.src }}"
6     dest: /etc/
7     mode: "{{ item.mode }}"
8   notify:
9     - restart rsyncd
10  loop:
11    - { src: rsyncd.conf, mode: '644'}
12    - { src: rsync.passwd, mode: '600'}
13
14 #2.启动服务
15 - name: start
16   systemd:
17     name: rsyncd
18     state: started
19     enabled: yes
```

### 2. 拆分后的nfs角色

```
1 [root@m-61 ~]# cat /etc/ansible/roles/nfs_server/tasks/main.yml
2 #1.复制配置文件和密码文件
3 - name: copy_exports
4   template:
5     src: exports.j2
6     dest: /etc/exports
7   notify:
8     - restart nfs
9
10 #2.启动服务
11 - name: start
12   systemd:
13     name: nfs
14     state: started
15     enabled: yes
```

### 3. 拆分后的lsyncd角色

```
1
```

### 4. 调用文件

## rsync

```
1 [root@m-61 ~]# cat /etc/ansible/rsync_server.yaml
2 - hosts: rsync_server
3   roles:
4     - init
5     - rsync_server
```

## nfs

```
1 [root@m-61 ~]# cat /etc/ansible/nfs_server.yaml
2 - hosts: nfs
3   roles:
4     - init
5     - nfs_server
```