[Top](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/ARCHITECTURE/DAY02/CASE/01/index.html" \l "page_top_case)

# NSD ARCHITECTURE DAY02

1. [练习1：playbook练习](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/ARCHITECTURE/DAY02/CASE/01/index.html" \l "case1)
2. [案例2：变量练习](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/ARCHITECTURE/DAY02/CASE/01/index.html" \l "case2)
3. [案例3：handlers练习](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/ARCHITECTURE/DAY02/CASE/01/index.html" \l "case3)
4. [案例4：编写playbook](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/ARCHITECTURE/DAY02/CASE/01/index.html" \l "case4)

## 1 练习1：playbook练习

### 1.1 问题

本案例要求：

* 安装Apache
* 修改监听端口为8080
* 设置默认主页hello world
* 启动服务
* 设置开机自启

### 1.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：playbook的ping脚本检测

1. [root@ansible ansible]# vim ping.yml
2. ---
3. - hosts: all
4. remote\_user: root
5. tasks:
6. - ping:
7. [root@ansible ansible]# ansible-playbook ping.yml //输出结果
8. PLAY [all] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
9. TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
10. ok: [web1]
11. ok: [web2]
12. ok: [cache]
13. ok: [db1]
14. ok: [db2]
15. TASK [ping] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
16. ok: [db1]
17. ok: [web2]
18. ok: [cache]
19. ok: [web1]
20. ok: [db2]
21. PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
22. cache : ok=2 changed=0 unreachable=0 failed=0
23. db1 : ok=2 changed=0 unreachable=0 failed=0
24. db2 : ok=2 changed=0 unreachable=0 failed=0
25. web1 : ok=2 changed=0 unreachable=0 failed=0
26. web2 : ok=2 changed=0 unreachable=0 failed=0

注意：如果检测的时候出错，会在当前的目录生成一个新的文件（以.retry结尾），可以去这个文件里面看是哪个主机的错

步骤二：用playbook安装Apache,修改端口，配置ServerName，修改主页，设置开机自启

1. [root@ansible ansible]# vim http.yml
2. ---
3. - hosts: cache
4. remote\_user: root
5. tasks:
6. - name: install one specific version of Apache
7. yum:
8. name: httpd        //安装Apache
9. state: installed
10. - lineinfile:
11. path: /etc/httpd/conf/httpd.conf
12. regexp: '^Listen '
13. line: 'Listen 8080'        //修改端口为8080
14. - service:
15. name: httpd
16. enabled: yes        //开机自启
17. state: restarted
18. - copy:
19. src: /root/index.html        //修改主页，可以自己写个页面
20. dest: /var/www/html/index.html
21. [root@ansible ansible]# ansible-playbook http.yml
22. [root@ansible ansible]# curl 192.168.1.45:8080
23. hello world
24. [root@ansible ansible]# ssh cache
25. Last login: Fri Sep 7 09:32:05 2018 from 192.168.1.40
26. [root@cache ~]# apachectl -t
27. Syntax OK

## 2 案例2：变量练习

### 2.1 问题

本案例要求熟悉playbook进阶：

* 练习使用user模块添加用户
* 练习使用变量简化task，让play通用性更强
* 练习使用过滤器

### 2.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：使用user模块添加用户，并修改密码

1. [root@ansible ansible]# vim user.yml
2. ---
3. - hosts: cache
4. remote\_user: root
5. vars:
6. username: xiaoming
7. tasks:
8. - name: create user "{{username}}"
9. user: group=wheel uid=1000 name={{username}}
10. - shell: echo 123456 | passwd --stdin xiaoming
11. - shell: chage -d 0 {{username}}
12. [root@ansible ansible]# ansible-playbook user.yml //执行结果
13. PLAY [cache] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
14. TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
15. ok: [cache]
16. TASK [create user " xiaoming "] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
17. changed: [cache]
18. TASK [command] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
19. changed: [cache]
20. TASK [command] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
21. changed: [cache]
22. PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
23. cache : ok=4 changed=3 unreachable=0 failed=0

步骤二：变量过滤器，创建一个用户，设置密码

1. [root@ansible ansible]# vim user1.yml
2. ---
3. - hosts: cache
4. remote\_user: root
5. tasks:
6. - user:
7. name: lisi
8. group: root
9. password: "{{'123456' | password\_hash('sha512')}}"
10. - shell: chage -d 0 lisi
11. [root@ansible ansible]# ansible-playbook user1.yml
12. PLAY [cache] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
13. TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
14. ok: [cache]
15. TASK [user] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
16. changed: [cache]
17. TASK [command] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
18. changed: [cache]
19. PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
20. cache : ok=3 changed=2 unreachable=0 failed=0

步骤三：定义一个变量创建用户

1. [root@ansible ansible]# vim user2.yml
2. ---
3. - hosts: cache
4. remote\_user: root
5. vars:
6. user: zhangs
7. tasks:
8. - user:
9. name: "{{user}}"
10. group: root
11. password: "{{'123456' | password\_hash('sha512')}}"
12. - shell: chage -d 0 "{{user}}"
13. [root@ansible ansible]# ansible-playbook user2.yml
14. PLAY [cache] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
15. TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
16. ok: [cache]
17. TASK [user] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
18. changed: [cache]
19. TASK [command] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
20. changed: [cache]
21. PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
22. cache : ok=3 changed=2 unreachable=0 failed=0

## 3 案例3：handlers练习

### 3.1 问题

本案例要求：

* 安装Apache软件
* 配置文件，重新载入配置文件让服务生效
* 使用handlers来实现

### 3.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：error

playbook从上往下顺序执行，若报错，后面的命令不会在执行，若想解决可以使用ignoring\_errors:True(使用这个，会有报错信息，告诉你错误忽略，继续执行下面的命令)

1. [root@ansible ansible]# vim error.yml
2. ---
3. - hosts: web
4. remote\_user: root
5. tasks:
6. - shell: mkdir /tmp/cache
7. - name: ReStart service httpd
8. service:
9. name: httpd
10. state: restarted
11. - name: run some command
12. shell: /usr/bin/somecommand
13. ignore\_errors: True
14. [root@ansible ansible]# ansible-playbook error.yml
15. PLAY [web] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
16. TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
17. ok: [web2]
18. ok: [web1]
19. TASK [command] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
20. [WARNING]: Consider using file module with state=directory rather than running mkdir
21. fatal: [web1]: FAILED! => {"changed": true, "cmd": "mkdir /tmp/cache", "delta": "0:00:00.004799", "end": "2019-04-11 14:57:11.582036", "msg": "non-zero return code", "rc": 1, "start": "2019-04-11 14:57:11.577237", "stderr": "mkdir: \u65e0\u6cd5\u521b\u5efa\u76ee\u5f55\"/tmp/cache\": \u6587\u4ef6\u5df2\u5b58\u5728", "stderr\_lines": ["mkdir: \u65e0\u6cd5\u521b\u5efa\u76ee\u5f55\"/tmp/cache\": \u6587\u4ef6\u5df2\u5b58\u5728"], "stdout": "", "stdout\_lines": []}
22. changed: [web2]
23. TASK [ReStart service httpd] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
24. fatal: [web2]: FAILED! => {"changed": false, "msg": "Could not find the requested service httpd: host"}
25. to retry, use: --limit @/etc/ansible/error.retry
26. PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
27. web1 : ok=1 changed=0 unreachable=0 failed=1
28. web2 : ok=2 changed=1 unreachable=0 failed=1
29. [root@ansible ansible]#

步骤二：tags给指定的任务定义一个调用标识

1）tags 样例

1. [root@ansible ansible]# vim adhttp.yml
2. ---
3. - hosts: cache
4. remote\_user: root
5. tasks:
6. - copy:
7. src: /root/httpd.conf
8. dest: /etc/httpd/conf/httpd.conf
9. owner: apache
10. group: apache
11. mode: 0644
12. tags: syncconf

2）调用方式

1. [root@ansible ansible]# ansible-playbook tags.yml -t syncconf
2. PLAY [cache] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
3. TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
4. ok: [cache]
5. TASK [copy] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
6. changed: [cache]
7. PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
8. cache : ok=2 changed=1 unreachable=0 failed=0

步骤三： handlers

关注的资源发生变化时采取的操作

1) 使用handlers来配置文件，重新载入配置文件让服务生效

1. [root@ansible ansible]# vim handers.yml
2. ---
3. - hosts: cache
4. remote\_user: root
5. tasks:
6. - copy:
7. src: /root/httpd.conf
8. dest: /etc/httpd/conf/httpd.conf
9. owner: apache
10. group: apache
11. mode: 0644
12. tags: syncconf
13. notify:
14. - restart httpd
15. handlers:
16. - name: restart httpd
17. service:
18. name: httpd
19. state: restarted
20. [root@ansible ansible]# ansible-playbook handers.yml
21. PLAY [cache] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
22. TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
23. ok: [cache]
24. TASK [copy] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
25. ok: [cache]
26. PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
27. cache : ok=2 changed=0 unreachable=0 failed=0

## 4 案例4：编写playbook

### 4.1 问题

本案例要求：

* 把系统负载太高的Apache服务停止

### 4.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：把系统负载太高的Apache服务停止

1）当系统负载超过0.7时，则关掉httpd

1. [root@ansible ansible]# vim when.yml
2. ---
3. - hosts: cache
4. remote\_user: root
5. tasks:
6. - shell: uptime | awk '{printf("%.2f\n",$(NF-2))}'
7. register: result
8. - service:
9. name: httpd
10. state: stopped
11. when: result.stdout|float > 0.7
12. [root@ansible ansible]# ansible-playbook when.yml
13. PLAY [cache] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
14. TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
15. ok: [cache]
16. TASK [command] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
17. changed: [cache]
18. TASK [service] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
19. changed: [cache]
20. PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
21. cache : ok=3 changed=2 unreachable=0 failed=0

步骤三：with\_items标准循环

1）with\_item创建多用户

1. [root@ansible ansible]# vim adduser.yml
2. ---
3. - hosts: web2
4. remote\_user: root
5. tasks:
6. - name: add users
7. user: group=wheel password={{'123456' | password\_hash('sha512')}} name={{item}}
8. with\_items: ["nb", "dd", "plj", "lx"]
9. [root@ansible ansible]# ansible-playbook adduser.yml
10. PLAY [web2] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
11. TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
12. ok: [web2]
13. TASK [add users] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
14. changed: [web2] => (item=nb)
15. changed: [web2] => (item=dd)
16. changed: [web2] => (item=plj)
17. changed: [web2] => (item=lx)
18. PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
19. web2 : ok=2 changed=1 unreachable=0 failed=0

2）为不同用户定义不同组

1. [root@ansible ansible]# vim adduser1.yml
2. ---
3. - hosts: web2
4. remote\_user: root
5. tasks:
6. - name: add users
7. user: group={{item.group}} password={{'123456' | password\_hash('sha512')}} name={{item.name}}
8. with\_items:
9. - {name: 'nb', group: 'root'}
10. - {name: 'dd', group: 'root'}
11. - {name: 'plj', group: 'wheel'}
12. - {name: 'lx', group: 'wheel'}
13. [root@ansible ansible]# ansible-playbook adduser1.yml
14. PLAY [web2] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
15. TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
16. ok: [web2]
17. TASK [add users] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
18. changed: [web2] => (item={u'group': u'root', u'name': u'nb'})
19. changed: [web2] => (item={u'group': u'root', u'name': u'dd'})
20. changed: [web2] => (item={u'group': u'wheel', u'name': u'plj'})
21. changed: [web2] => (item={u'group': u'wheel', u'name': u'lx'})
22. PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
23. web2 : ok=2 changed=1 unreachable=0 failed=0

3）include and roles

在编写playbook的时候随着项目越来越大，playbook越来越复杂。可以把一些play、task 或 handler放到其他文件中，通过包含进来是一个不错的选择

roles像是加强版的include，它可以引入一个项目的文件和目录

一般所需的目录层级有

vars：变量层

tasks：任务层

handlers：触发条件

files：文件

template：模板

default：默认，优先级最低

1. ...
2. tasks:
3. - include: tasks/setup.yml
4. - include: tasks/users.yml user=plj
5. //users.yml 中可以通过{{ user }}来使用这些变量
6. handlers:
7. - include: handlers/handlers.yml

步骤五：debug检测

1. [root@ansible ansible]# ansible-playbook --syntax-check http.yml //检测语法
2. playbook: http.yml
3. [root@ansible ansible]# ansible-playbook -C http.yml //测试运行
4. [root@ansible ansible]# ansible-playbook http.yml --list-tasks
5. //显示要执行的工作
6. playbook: http.yml
7. play #1 (cache): cache    TAGS: []
8. tasks:
9. install one specific version of Apache    TAGS: []
10. lineinfile    TAGS: []
11. replace    TAGS: []
12. service    TAGS: []
13. copy    TAGS: []
14. [root@ansible ansible]# vim debug.yml
15. ---
16. - hosts: cache
17. remote\_user: root
18. tasks:
19. - shell: uptime |awk '{printf("%f\n",$(NF-2))}'
20. register: result
21. - service:
22. name: httpd
23. state: stopped
24. when: result.stdout|float > 0.7
25. - name: Show debug info
26. debug: var=result
27. [root@ansible ansible]# ansible-playbook debug.yml         //运行
28. PLAY [cache] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
29. TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
30. ok: [cache]
31. TASK [command] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
32. changed: [cache]
33. TASK [command] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
34. skipping: [cache]
35. TASK [Show debug info] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
36. ok: [cache] => {
37. "result": {
38. "changed": true,
39. "cmd": "uptime |awk '{printf(\"%f\\n\",$(NF-2))}'",
40. "delta": "0:00:00.005905",
41. "end": "2018-09-07 12:57:51.371013",
42. "failed": false,
43. "rc": 0,
44. "start": "2018-09-07 12:57:51.365108",
45. "stderr": "",
46. "stderr\_lines": [],
47. "stdout": "0.000000",
48. "stdout\_lines": [
49. "0.000000"
50. ]
51. }
52. }
53. PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
54. cache : ok=3 changed=1 unreachable=0 failed=0