

- 简介
- 为task打tag
- 使用tag
 - 执行指定tag的task
 - 排除指定tag的task
 - 查看playbook中的所有tag
- 打tag的几种方式
- ansible内置tag

简介

在大型项目当中，通常一个playbook会有非常多的task。而我们每次执行这个playbook时，都会将所有task运行一遍。而事实上，在实际使用过程中，我们可能只是想要执行其中的一部分任务而已，并不想把整个playbook完整跑一遍。这个时候就需要用到tags。

通过tags，我们可以给playbook中的某一些任务打上“标签”，而在执行playbook的时候，我们可以通过选定标签的方式指定只执行哪一些任务或者不执行哪一些任务。

为task打tag

下面是一个安装httpd的简单示例：

```
# cat /etc/ansible/playbook/install_web.yml
- name: configure webserver
  hosts: all
  remote_user: ansible

  tasks:
    - name: Install httpd
      yum:
        name: httpd
        state: present
        tags: install_httpd

    - name: Configuration httpd
      copy:
        src: /root/httpd.conf
        dest: /etc/httpd/conf/httpd.conf
        tags: conf_httpd
      notify:
        - restart httpd

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

```
    enabled: no
    tags: start_httpd

handlers:
  - name: restart httpd
    service: name=httpd state=restart
```

在上面的示例中，我们为两个task定义了三个tags: `install_httpd`、`conf_httpd` 以及 `start_httpd`。

使用tag

执行指定tag的task

有了tags之后，我们就可以只运行playbook中指定标签的task了：

```
# ansible-playbook --tags="start_httpd" install_web.yml

PLAY [configure webserver]
*****

TASK [Gathering Facts]
*****

ok: [10.1.61.187]

TASK [Start httpd]
*****

changed: [10.1.61.187]

PLAY RECAP
*****

10.1.61.187      : ok=2    changed=1    unreachable=0    failed=0    skipped=0
rescued=0       ignored=0
```

也可以一次指定多个tag执行：

```
# ansible-playbook --tags="conf_httpd,start_httpd" install_web.yml

PLAY [configure webserver]
*****

TASK [Gathering Facts]
*****
```

```

ok: [10.1.61.187]

TASK [Cofiguration httpd]
*****
*****
ok: [10.1.61.187]

TASK [Start httpd]
*****
*****
ok: [10.1.61.187]

PLAY RECAP
*****
*****
10.1.61.187      : ok=3    changed=0    unreachable=0    failed=0    skipped=0
rescued=0       ignored=0

```

排除指定tag的task

通过下面的方式可以排除指定了tag的task，即除了指定tag的task不执行，其他task都执行：

```

# ansible-playbook --skip-tags="install_httpd" install_web.yml

PLAY [configure webserver]
*****
*****

TASK [Gathering Facts]
*****
*****
ok: [10.1.61.187]

TASK [Cofiguration httpd]
*****
*****
ok: [10.1.61.187]

TASK [Start httpd]
*****
*****
ok: [10.1.61.187]

PLAY RECAP
*****
*****
10.1.61.187      : ok=3    changed=0    unreachable=0    failed=0    skipped=0
rescued=0       ignored=0

```

执行效果跟上面一样。

查看playbook中的所有tag

可以通过 `--list-tags` 参数列出指定的playbook中所有的tag

```
# ansible-playbook --list-tags install_web.yml

playbook: install_web.yml

play #1 (all): configure webserver TAGS: []
TASK TAGS: [conf_httpd, install_httpd, start_httpd]
```

打tag的几种方式

1. 为一个任务指定一个标签

这种方式就是上面示例中的方法：

```
tags: conf_httpd
```

2. 为一个任务指定多个标签

可以通过列表的方式为一个任务指定多个标签：

```
tags:
  - install_httpd
  - install_web

tags: ['install_httpd','install_web']

tags: install_httpd,install_web
```

3. 为一个play指定一组标签

当为一个play指定一组标签后，该play下的所有task都会自动继承该标签，各task也可以自定义自己的标签。

```
- name: configure webserver
  hosts: all
  remote_user: ansible
  tags:
    - httpd
  tasks:
    ...
```

ansible内置tag

除了用户自定义tag，ansible也内置了几个tag，这些tag都包含特殊含义：

- **always**：一旦某个task被打上了always的tag，则无论是playbook的完整执行，还是指定tag执行，不管你指定的tag是啥，该任务总是会被执行。除非明确指定"--skip-tags=always"选项，才不会执行该task。
- **never**：该标签与always正好相反，总是不会执行，除非明确指定"--tags=never"选项。
- **tagged**：在调用时使用

```
# 所有打了tag的任务都会被执行，包含never tag的除外，没有标签的不会被执行
ansible-playbook --tags tagged install_web.yaml

# 所有打了tag的任务都不会被执行，包括always tag也不会被执行
ansible-playbook --skip-tags tagged install_web.yaml
```

- **untagged**：在调用时使用

```
# 所有未打tag的任务都会被执行，打了always tag的也会被执行
ansible-playbook --tags untagged install_web.yaml

# 所有未打tag的任务都不会被执行
ansible-playbook --skip-tags untagged install_web.yaml
```

- **all**：表示所有任务都会被执行，在默认情况下，不指定任何标签，则使用的就是该标签