

ansible常用模块

user模块

模块动作

1. name: 指定用户名
2. comment: 用户描述
3. createhome: 是否创建用户家目录, 默认为yes
4. home: 设置用户家目录
5. uid: 指定用户UID
6. shell: 指定用户登录shell类型, 例如shell: /bin/bash
7. state: 可选present和absent
8. system: 是否创建为系统用户
9. password: 设置密码

```
1 # user模块使用
2 - hosts: all
3   remote_user: root
4   tasks:
5     - name: User Management
6       user:
7         name: ansibleuser
8         comment: Ansible related users
9         createhome: yes
10        shell: /usr/sbin/nologin
11        uid: 515
12        state: absent
```

cron模块

模块动作

1. name: 任务计划名称
2. state: 允许任务计划存在或不存在, present和absent
3. job: 要执行的命令
4. backup: 对远程主机上的原任务计划内容修改之前做备份
5. day|hour|minute|month|weekday|: 指定时间

```
1 # cron模块使用
2 - hosts: all
3     remote_user: root
4     tasks:
5     - name: Task plan
6       cron:
7         name: date
8         state: present
9         minute: 0-59
10        job: date >> /root/date.txt
```

copy模块

模块动作

1. backup: 复制不会覆盖源文件, 将复制之前, 将源文件作一次备份。如test文件会备份为test.4846.2017-04-11@20:22:48~
2. dest: 目标地址, 要复制文件到远程服务器的绝对路径。如果src是一个目录, 这这也是一个目录。
3. src: 源地址。如果是目录, 将递归复制。如果 '/' 结尾, 则复制目录下的内容, 如果没有, 则连同目录一起复制
4. force: 如果目标主机包含该文件, 但内容不同, 如果设置为yes, 则强制覆盖, 如果为no, 则只有当目标主机的目标位置不存在该文件时, 才复制。默认为yes

5. mode: 指定复制之后的文件的权限。例如命令行模式下:
mode=u=rw,g=r,o=r
6. group: 复制之后的文件的属组
7. owner: 复制之后的文件的属主

```
1 # copy模块
2 - hosts: 192.168.1.151
3   remote_user: root
4   tasks:
5     - name: copy a file
6       copy:
7         backup: yes
8         dest: /root/works/
9         src: /root/works/
10        force: yes
11        mode: u=rwx,g=rw,o=r
12        group: xuekaixin
13        owner: xuekaixin
```

file模块

模块动作: 创建、修改、删除文件操作

1. group: 创建之后文件的属组
2. owner: 创建只有文件的属主
3. mode: 指定创建之后的文件的权限。例如命令行模式下:
mode=u=rw,g=r,o=r
4. state: file | link | firectory | hard | touch | absent, 指定文件状态
5. path: 指明要操作的文件路径
6. src: 要操作的源文件, 在创建链接文件时用到
7. dest: 要操作的目标文件, 在创建链接时用到

```
1 # file模块
```

```
2 - hosts: test
3   remote_user: root
4   tasks:
5     - name: touch a file
6       file:
7         path: /root/works/nginx.conf
8         group: xuekaixin
9         owner: xuekaixin
10        mode: 0644
11        state: touch
12  # 创建一个链接文件
13  - file:
14    src: /file/to/link/to
15    dest: /path/to/symlink
16    owner: foo
17    group: foo
18    state: link
```

ping模块

```
1 ansible all -m ping
```

yum模块

1. name: 用于安装的包名称
2. state: present|installed|latest|absent|removed
3. enablerepo: 指定yum源
4. list: 列出指定的包的列表及信息

```
1 - hosts: test
2   remote_user: root
3   tasks:
4     - name: yum install
5       file:
6         name: httpd
7         state: present
```

service模块

1. state: started|stoped|restarted|reloaded
2. name: 指定要管理的服务名称
3. enabled: yes|no, 开机自动启动

```
1 # service模块
2 - hosts: test
3   remote_user: root
4   tasks:
5     - name: service
6       service:
7         name: httpd
8         state: restarted
```

shell模块和command模块

command和shell模块的区别的shell模块支持管道符, 和变量的使用。

script模块

```
1 - hosts: test
2   remote_user: root
3   tasks:
4     - script: /root/works/script.sh
```

setup模块

获取服务器信息, 这些信息都保存在变量中。

```
ansible all -m setup
```

参考: [All Modules](#)