

- [Ad-hoc](#)
  - [简介](#)
  - [命令说明](#)
  - [用户说明](#)
- [命令执行模块](#)
  - [shell模块](#)
  - [script模块](#)

# Ad-hoc

---

## 简介

Ad-Hoc 是指ansible下临时执行的一条命令，并且不需要保存的命令，对于复杂的命令会使用playbook。Ad-hoc的执行依赖于模块，ansible官方提供了大量的模块。如：command、raw、shell、file、cron等，具体可以通过ansible-doc -l 进行查看。可以使用ansible-doc -s module来查看某个模块的参数，也可以使用ansible-doc help module来查看该模块更详细的信息。

## 命令说明

一个ad-hoc命令的执行，需要按以下格式进行执行：

```
ansible 主机或组 -m 模块名 -a '模块参数' ansible参数
```

- 主机和组，是在/etc/ansible/hosts 里进行指定的部分，当然动态Inventory 使用的是脚本从外部应用里获取的主机；
- 模块名，可以通过ansible-doc -l 查看目前安装的模块，默认不指定时，使用的是command模块，具体可以查看/etc/ansible/ansible.cfg 的 “#module\_name = command” 部分，默认模块可以在该配置文件中进行修改；
- 模块参数，可以通过 “ansible-doc -s 模块名” 查看具体的用法及后面的参数；
- ansible参数，可以通过ansible命令的帮助信息里查看到，这里有很多参数可供选择，如是否需要输入密码、是否sudo等。

## 用户说明

ansible在执行ad-hoc的时候，需要使用远程管理机上的一个用户身份，默认这个用户是root，而在实际生产环境中，不建议直接使用root，而推荐使用一个普通用户，需要配置该普通用户可通过sudo提权。

下面是一个配置示例：

1. 在被管理机上创建一个普通用户ansible，并配置sudo提权：

```
# 创建ansible用户
useradd ansible

# 设置ansible用户可免密提权
vim /etc/sudoers.d/ansible
ansible    ALL=(ALL)      NOPASSWD:ALL

chmod 400 /etc/sudoers.d/ansible

mkdir /home/ansible/.ssh
chown ansible.ansible /home/ansible/.ssh
chmod 700 /home/ansible/.ssh
```

2. 在管理端修改ansible.cfg配置文件如下：

```
[default]
remote_user = ansible
ask_pass = False

[privilegeEscalation]
become=True
become_method=sudo
become_user=root
become_ask_pass=False
```

3. 将管理端用户的 id\_rsa.pub 复制到被管理端

的 /home/ansible/.ssh//home/ansible/.ssh/authorized\_keys 文件中，并设置该文件的权限为 400

```
# 管理端执行:
ssh-copy-id -i .ssh/id_rsa.pub ansible@db1.example.com
```

需要说明的是，通过上面的操作，需要为ansible用户先创建密码，在实际生产当中，不建议为该用户设置密码。可在安装系统时，提前将管理端公钥直接写入到被管理节点的ansible用户下

## 命令执行模块

命令执行模块包含如下 四个模块：

- command模块：该模块通过-a跟上要执行的命令可以直接执行，不过命令里如果有带有如下字符部分则执行不成功 “<”, “>”, “|”, “&”；
- shell 模块：用法基本和command一样，不过其是通过/bin/sh进行执行，所以shell 模块可以执行任何命令，就像在本机执行一样；

- raw模块：用法和shell模块一样，也可以执行任意命令，就像在本机执行一样；和command、shell模块不同的是其没有chdir、creates、removes参数
- script模块：将管理端的shell 在被管理主机上执行，其原理是先将shell 复制到远程主机，再在远程主机上执行，原理类似于raw模块。

## shell模块

shell模块包含如下选项：

- creates: 一个文件名，当该文件存在，则该命令不执行
- free\_form: 要执行的linux指令
- chdir: 在执行指令之前，先切换到该指定的目录
- removes: 一个文件名，当该文件不存在，则该选项不执行

chdir示例：

```
# 三个命令都会返回执行成功的状态。不过实际上只有前两个文件会被创建成功。使用raw模块的执行的结果文件事实上也被正常创建了，不过不是在chdir指定的目录，而是在当前执行用户的家目录
ansible 192.168.1.1 -m command -a 'chdir=/tmp/test.txt touch test.file'
ansible 192.168.1.1 -m shell -a 'chdir=/tmp/test.txt touch test2.file'
ansible 192.168.1.1 -m raw -a 'chdir=/tmp/text.txt touch test3.file'
```

creates与removes示例：

```
ansible 192.168.1.1 -a 'creates=/tmp/server.txt uptime' #当/tmp/server.txt文件存在时，则不执行uptime指令
ansible 192.168.1.1 -a 'removes=/tmp/server.txt uptime' #当/tmp/server.txt文件不存在时，则不执行uptime指令
```

## script模块

示例：

```
#要执行的脚本文件script.sh内容如下：
#!/bin/bash
ifconfig
df -hT

# 执行ansible指令：
ansible 10.212.52.252 -m script -a 'script.sh'
```