

# ansible剧本编写

原创

GeorgeKai

2018-02-01 18:43:42

评论(0)

578人阅读

## ansible剧本编写

### 1.1.1 ansible基于口令方式免交互登录

1. 编辑/etc/ansible/hosts管理配置文件，实现基于口令免交互登录

```
vim /etc/ansible/hosts
[client]
172.16.1.7
172.16.1.31 ansible_ssh_user=root ansible_ssh_pass=123456
172.16.1.41
```

2. 测试是否需要进行交互登录

```
[root@m01 ~]# ansible george -m ping
172.16.1.41 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

### 1.1.2. ansible以交互方式输入密码登录（所有远程主机密码必须一致）

1. 不配置密码

```
vim /etc/ansible/hosts
[oldboy]
172.16.1.7
172.16.1.31 ansible_ssh_user=root
172.16.1.41
```

2. -k参数实现以交互方式远程登录

```
[root@m01 ~]# ansible george -m ping -k #以交互方式输入密码进行远程管理
SSH password:
172.16.1.7 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

## 1.2 ansible剧本编写规则说明

### 1.2.1 语法格式

ansible剧本格式：遵循yaml语法格式（类似python脚本编写格式）

rsync配置文件格式：ini语法格式

sersync配置文件格式：xml语法格式（标签格式）

### 1.2.2 pyYAML语法规则：

规则一：缩进

yaml使用一个固定的缩进风格表示数据层结构关系，Saltstack需要每个缩进级别由两个空格组成。一定不能使用tab键

强调\*\*\*\*\*：当编写ansible剧本时，请把键盘的tab键扣掉

规则二：冒号

```
CMD="echo"
yaml:
mykey:
```

每个冒号后面一定要有一个空格（以冒号结尾不需要空格，表示文件路径的模版可以不需要空格）

### 规则三：短横线

想要表示列表项，使用一个短横杠加一个空格。多个项使用同样的缩进级别作为同一个列表的一部分

核心规则：有效的利用空格进行剧本的编写，剧本编写是不支持tab的

## 1.2.3 ansible-playbook编写格式

### 1. 创建部署rsync服务剧本

```
vim /etc/ansible/ansible_play_book/rsync.yaml
```

```
- hosts: server
  tasks:
    - name: 01.push rsyncd.conf
      copy: src=/tmp/rsyncd.conf dest=/etc/
    - name: 02.create user-rsync
      user:
        name: rsync
        shell: /sbin/nologin
        createhome: no
        state: present
    - name: 03.create dir_backup,auth
      file: dest=/backup state=directory owner=rsync group=rsync
    - name: 04.create rsync.pass,auth
      copy: src=/tmp/rsync.pass dest=/etc/ mode=0600
```

-空hosts: 空yunwei : 指定主机组或单个主机，也可以是all所有主机

空空tasks: : 固定格式

空空空空-空file: 空dest=src=/etc/hosts..... : 指定模块

PS:下一个级别要比上一个级别多两个空格

### 2. 对剧本语法检测:

```
ansible-playbook --syntax-check /etc/ansible/ansible-play_book/rsync.yaml
```

### 3. -C模拟执行剧本:

```
ansible-playbook -C /etc/ansible/ansible-play_book/rsync.yaml
```

### 4. 执行剧本:

```
ansible-playbook /etc/ansible/ansible-play_book/rsync.yaml
```

## 1.2.4 多剧本编写

### 1. 编写多个剧本

### 剧本的开头，可以不写

```
- hosts: all          <- all表示所有主机，可以写某个主机组；-(空格)hosts:(空格)all
  task:              <- 剧本所要干的事情；(空格)(空格)task:
    - command: echo hello oldboy linux.      (空格)(空格)(空格)(空格)-(空格)模块名称:(空格)模块中对应的功能
```

### 2. 剧本编写内容扩展: 剧本任务定义名称

```
- hosts: 172.16.1.7    <- 处理指定单台服务器          -(空格)hosts:(空格)all
  task:                <- 剧本所要干的事情;          (空格)(空格)task:
    - name:            <- 注释信息                      (空格)(空格)(空格)(空格)-(空格)模块名称:
      command: echo hello oldboy linux.                (空格)模块中对应的功能
```

### 3. 剧本编写内容扩展：剧本任务编写定时任务

```
# ansible all -m cron -a "name='restart network' minute=00 hour=00 job='/usr/sbin/ntpdate
time.nist.gov >/dev/null 2>&1'"
- hosts: all
  tasks:
    - name: restart-network
      cron: name='restart network' minute=00 hour=00 job='/usr/sbin/ntpdate time.nist.gov >/dev/null
2>&1'
```

说明：测试剧本命令后面可以跟多个-v进行调试检查

### 4. 剧本编写内容扩展：剧本任务编写多个任务

```
- hosts: all
  tasks:
    - name: restart-network
      cron: name='restart network' minute=00 hour=00 job='/usr/sbin/ntpdate time.nist.gov
>/dev/null 2>&1'
    - name: sync time
      cron: name='sync time' minute=*/5 job='/usr/sbin/ntpdate pool.ntp.com >/dev/null 2>&1'
```

### 5. 剧本编写内容扩展：剧本任务编写多个主机

```
- hosts: 172.16.1.7
  tasks:
    - name: restart-network
      cron: name='restart network' minute=00 hour=00 job='/usr/sbin/ntpdate time.nist.gov >/dev/null
2>&1'
    - name: sync time
      cron: name='sync time' minute=*/5 job='/usr/sbin/ntpdate pool.ntp.com >/dev/null 2>&1'
- hosts: 172.16.1.31
  tasks:
    - name: show ip addr to file
      shell: echo $(hostname -i) >> /tmp/ip.txt
```

#### 1.2.5 剧本编写方式

1. 多主机单任务编写方式
2. 多主机多任务编写方式
3. 不同主机多任务编写方式

#### 1.2.6 ansible编写剧本拍错思路：

1. ansible-playbook编写完，检查语法和模式测试运行
2. 打开脚本，根据-name定位异常原因，然后将剧本中的内容转换为命令执行一次
3. 将参数中的脚本文件推送到远程服务器，在远程服务器本地执行一次
4. ansible执行时，加上-vvvv（显示详细执行过程），也可以定位异常原因

小伙伴们可以关注我的微信公众号：linux运维菜鸟之旅



关注“中国电信天津网厅”公众号，首次绑定可免费领2G流量，为你的学习提供流量！



ansible更多知识——友情链接：<http://blog.51cto.com/breezey>

---

版权声明：原创作品，如需转载，请注明出处。否则将追究法律责任