

原创

Ansible之playbook的使用



karony1

2018-02-03 18:38:44 328327人阅读 3人评论

playbook介绍

一. 为什么引入playbook

我们完成一个任务，例如安装部署一个httpd服务，我们需要多个模块（一个模块也可以称之为task）提供功能来完成。而playbook就是组织多个task的容器，他的实质就是一个文件，有着特定的组织格式，它采用的语法格式是YAML（Yet Another Markup Language）。YAML语法能够简单的表示散列表，字典等数据结构。具体请参考[YAML详细语法](#)

YAML基本语法

列表：每一个列表成员前面都要有一个短横线和一个空格

```
fruits:
  - Apple
  - Orange
  - Strawberry
  - Mango
```

或者：

```
fruits: ['Apple', 'Orange', 'Strawberry', 'Mango']
```

字典：每一个成员由键值对组成，注意冒号后面要有空格

```
martin:
  name: Martin D'vloper
  job: Developer
  skill: Elite
或者
martin: {name: Martin D'vloper, job: Developer, skill: Elite}
```

列表和字典可以混合使用

```
- martin:
  name: Martin D'vloper
  job: Developer
  skills:
    - python
    - perl
    - pascal
- tabitha:
  name: Tabitha Bitumen
  job: Developer
  skills:
    - lisp
    - fortran
    - erlang
```

[在线客服](#)

- Hosts：运行执行任务（task）的目标主机
- remote_user：在远程主机上执行任务的用户
- tasks：任务列表
- handlers：任务，与tasks不同的是只有在接受到通知时才会被触发
- templates：使用模板语言的文本文件，使用jinja2语法。
- variables：变量，变量替换{{ variable_name }}

整个playbook是以task为中心，表明要执行的任务。hosts和remote_user表明在哪些远程主机以何种身份执行。其他组件让其能够更加灵活。

下面详细介绍某些组件

1. variable

变量定义在资产（inventory）中：

```
主机变量：
192.168.200.136 http_port=808 maxRequestsPerChild=808
192.168.200.137 http_port=8080 maxRequestsPerChild=909

主机组变量：
[webservers]
192.168.200.136
192.168.200.137

[webservers:vars]
ntp_server=ntp.exempl.com
proxy=proxy.exempl.com
```

变量定义在playbook中

```
- hosts: webservers
  vars:
    http_port: 80
```

使用facts变量：



```
facts变量是由setup模块获取远程主机的信息。
用法：
ansible 192.168.200.136 -m setup
```

在roles中定义变量

后面介绍

ansible-playbook 命令中传入参数

```
使用 -e选项传入参数

ansible-playbook 192.168.200.136 -e "httpd_port=808" httpd04.yml
```

变量的引用

```
{{ var_name }}
```

2. templates

它是一个模块功能，与copy不同的是他的文本文件采用了jinja2语法，

在线客服

jinja2基本语法如下，

```
字面量：
字符串：使用单引号或双引号
数字：整型，浮点数
```



算术运算：
+, -, *, /, //, %, **
比较运算：
==, !=, >, >=, <, <=
逻辑运算：
and, or, not

注意：template只能在palybook中使用。

3. tasks

执行的模块命令

格式：
action: 模块参数（此种方式只在较新的版本中出现）
module: 参数(已键值对的形式出现)

每一个task都有一个名称，用于标记此任务。任务示例：

```
name: install httpd
yum: name=httpd state=present
```

注意：shell和command没有参数，可在后面直接跟命令

```
shell: ss -tnl | grep :80
```

- (1) 某任务的运行状态为**changed**后，可通过相应的notify通知相应的handlers
- (2) 任务可以通过tags打标签，然后通过palybook命令-t选项调用。

三. playbook调用方式

用法：
ansible-playbook <filename.yml> ... [options]

<filename.yml>:yaml格式的playbook文件路径，必须指明
[options]: 选项
-C, --check: 并不在远程主机上执行，只是测试。
-i PATH, --inventory=PATH: 资产的文件路径
--flush-cache: 清楚fact缓存
--list-hosts: 列出匹配的远程主机，并不执行任何动作
-t, TAGS, --tags=TAGS: 运行指定的标签任务
--skip-tags: 跳过指定的notify，后面详细介绍。



四. palybook示例文件解析

1. 安装部署httpd服务-version1

资产文件

```
~] cat /etc/ansible/Hosts

[webservers]
192.168.200.136 httpd_port=808
192.168.200.137 httpd_port=8088
[test]
192.168.200.13[6:7]
```

httpd配置文件

```
~] grep ^Listen /etc/httpd/conf/httpd.conf

Listen 808
```

palybook文件

```
~] cat /root/httdn01.vml
```



```
tasks:
- name: install httpd
  yum: name=httpd state=present
- name: install configure file
  copy: src=httpd.conf dest=/etc/httpd/conf/
- name: start httpd service
  service: name=httpd state=started
```

测试playbook

```
[root@node2 ~]# ansible-playbook --check httpd01.yml

PLAY [webservers] *****
TASK [Gathering Facts] *****
ok: [192.168.200.136]
ok: [192.168.200.137]
TASK [install httpd] *****
ok: [192.168.200.136]
changed: [192.168.200.137]
TASK [install configure file] *****
changed: [192.168.200.137]
changed: [192.168.200.136]
TASK [start httpd service] *****
ok: [192.168.200.136]
changed: [192.168.200.137]
PLAY RECAP *****
192.168.200.136 : ok=4 changed=1 unreachable=0 failed=0
192.168.200.137 : ok=4 changed=3 unreachable=0 failed=0
```

运行playbook

```
[root@node2 ~]# ansible-playbook httpd01.yml

PLAY [webservers] *****
TASK [Gathering Facts] *****
ok: [192.168.200.136]
ok: [192.168.200.137]
TASK [install httpd] *****
ok: [192.168.200.136]
ok: [192.168.200.137]
TASK [install configure file] *****
fatal: [192.168.200.137]: FAILED! => ("changed": false, "checksum": "2986a97d1bd31843262c9756cf838504f889ed12", "failed": true, "msg": "Aborting. target uses selinux but py
thon bindings (libselinux-python) aren't installed")
TASK [start httpd service] *****
ok: [192.168.200.136]
to retry, use: --limit @/root/httpd01.retry
PLAY RECAP *****
192.168.200.136 : ok=4 changed=0 unreachable=0 failed=0
192.168.200.137 : ok=2 changed=0 unreachable=0 failed=1
```

再次执行playbook

```
~]# yum install libselinux-python
```

```
[root@node2 ~]# ansible-playbook httpd01.yml

PLAY [webservers] *****
TASK [Gathering Facts] *****
ok: [192.168.200.136]
ok: [192.168.200.137]
TASK [install httpd] *****
ok: [192.168.200.136]
ok: [192.168.200.137]
TASK [install configure file] *****
changed: [192.168.200.137]
TASK [start httpd service] *****
ok: [192.168.200.136]
fatal: [192.168.200.137]: FAILED! => ("changed": false, "failed": true, "msg": "httpd: Syntax error on line 56 of /etc/httpd/conf/httpd.conf: Include directory '/etc/httpd/
conf.modules.d' not found\n")
to retry, use: --limit @/root/httpd01.retry
PLAY RECAP *****
192.168.200.136 : ok=4 changed=0 unreachable=0 failed=0
192.168.200.137 : ok=3 changed=1 unreachable=0 failed=1
```

从上图可以看出，192.168.200.136执行成功，而192.168.200.137启动服务时配置文件错误，这是因为拷贝过去的配置文件是centos7上的，而137这台主机是centos6，它安装的是httpd-2.2配置文件不兼容。此问题后面解决。

查看服务启动时的端口

```
[root@node2 ~]# ansible 192.168.200.136 -m shell -a "ss -tnl | grep :80"
192.168.200.136 | SUCCESS | rc=0 >>
LISTEN 0 128 :::80
```



copy命令拷贝配置文件时，无法对配置文件进行修改，不够灵活。接下来我们使用template拷贝文件，并使用主机变量设置httpd端口号

资产文件

```
~]# cat /etc/ansible/hosts

[webservers]
192.168.200.136 httpd_port=8088
[test]
192.168.200.13[6:7]
```

配置文件

```
~]# grep ^Listen httpd.conf.j2

Listen {{ httpd_port }}
```

playbook文件

```
~]# cat httpd02.yml

- hosts: 192.168.200.136
  remote_user: root
  tasks:
    - name: install httpd
      yum: name=httpd state=present
    - name: install configure file
      template: src=httpd.conf.j2 dest=/etc/httpd/conf/httpd.conf
    - name: start httpd service
      service: name=httpd state=started
```

运行playbook

```
[root@node2 ~]# ansible-playbook httpd02.yml

PLAY [192.168.200.136] *****

TASK [Gathering Facts] *****
ok: [192.168.200.136]

TASK [install httpd] *****
ok: [192.168.200.136]

TASK [install configure file] *****
changed: [192.168.200.136]

TASK [start httpd service] *****
ok: [192.168.200.136]

PLAY RECAP *****
192.168.200.136 : ok=4 changed=1 unreachable=0 @51CTO博客

[root@node2 ~]# ansible 192.168.200.136 -m shell -a "ss -tnl | grep :80"
192.168.200.136 | SUCCESS | rc=0 >>
LISTEN 0 128 :::8088 :::* @51CTO博客
```



由图可以看出，playbook中只将文件拷贝过去了，并没有重启服务，所以我们可以看到端口监听的依然是808而不是8088。此时我们需要在配置文件修改时触发一个任务，这就是handlers的用法，重新修改playbook文件。

在线客服

重新修改YAML文件

```
~]# cat httpd02.yml
```



```

- name: install httpd
  yum: name=httpd state=present
- name: install configure file
  notify: restart httpd service
  template: src=httpd.conf.j2 dest=/etc/httpd/conf/httpd.conf
- name: start httpd service
  service: name=httpd state=started
handlers:
- name: restart httpd service
  service: name=httpd state=restarted

```

notify表明此task改变时，它会触发一个事件，此事件会调用name为restart httpd service的handlers task。

再次运行playbook

```

[root@node2 ~]# ansible-playbook httpd02.yml

PLAY [192.168.200.136] *****

TASK [Gathering Facts] *****
ok: [192.168.200.136]

TASK [install httpd] *****
ok: [192.168.200.136]

TASK [install configure file] *****
changed: [192.168.200.136]

TASK [start httpd service] *****
ok: [192.168.200.136]

RUNNING HANDLER [restart httpd service] *****
changed: [192.168.200.136]

PLAY RECAP *****
192.168.200.136      : ok=5    changed=2    unreachable=0    failed=0

[root@node2 ~]# ansible 192.168.200.136 -m shell -a "ss -tnl | grep :80"
192.168.200.136 | SUCCESS | rc=0 >>
LISTEN          0            128          :::8088          :::*             @51CTO博客

```



可以看到，此playbook执行了两个任务，一个是拷贝文件，一个是重启服务。也可以看到此时监听的端口不再是80而是8088

我们修改配置文件并重启服务，这是一个非常常见的操作。由上面我们看到，尽管我们只需执行两个task但我们把所有的任务都执行一遍，这会降低效率，我们可以使用tags来指定执行那个任务。

3. 指定执行任务

资产文件

注意，修改了端口号

```

~] cat /etc/ansible/hosts

[webservers]
192.168.200.136 http_port=8080
[test]
192.168.200.13[6:7]

```

修改playbook文件

```

~]# cat httpd03.yml

- hosts: 192.168.200.136
  remote user: root

```

在线客服



```

- name: install configure file
  notify: restart httpd service
  tags: reinstall configure file
  template: src=httpd.conf.j2 dest=/etc/httpd/conf/httpd.conf
- name: start httpd service
  service: name=httpd state=started
handlers:
- name: restart httpd service
  service: name=httpd state=restarted

```

执行playbook

由图可以看到，playbook只执行了拷贝文件，以及拷贝文件触发的重启服务事件。也可以看到服务开启的是8080端口。

五. ansible特性--判断和循环

在前面有一个问题没有解决就是centos6和centos7配置文件不兼容的问题，我们需要对其进行判断，不同的版本拷贝同的文件。

我们将lamp放在一台主机上时，我们需要安装多个程序包，写成一个一个的task显得效率不高，写的臃肿。我们可以使用循环来进行安装。

1 判断

以解决上面遗留的问题为例，如何让centos6的主机拷贝centos6的文件，让centos7的主机拷贝centos7的文件。

也可以使用变量来实现，此处不做演示

资产文件

```

~]# cat /etc/ansible/hosts

[webservers]
192.168.200.136 httpd_port=8088
192.168.200.137 httpd_port=8080
[test]
192.168.200.13[6:7]

```



playbook文件

```

[root@node2 ~]# cat httpd04.yml
- hosts: webservers
  remote_user: root
  tasks:
    - name: install httpd
      yum: name=httpd state=present
    - name: install configure httpd
      template: src=httpd.conf.j2 dest=/etc/httpd/conf/httpd.conf
      tags: reinstall configure file
      notify: restart httpd service
      when: ansible_distribution_major_version == "7"
    - name: install configure httpd
      template: src=httpd6.conf.j2 dest=/etc/httpd/conf/httpd.conf
      tags: reinstall configure file
      notify: restart httpd service
      when: ansible_distribution_major_version == "6"
    - name: start httpd service
      service: name=httpd state=started
  handlers:
    - name: restart httpd service
      service: name=httpd state=restarted

```

判断对应的主版本号，实现不同文件的拷贝

@51CTO博客

在线
客服

其中的ansible_distribution_major_version是ansible收集的facts

测试playbook文件



```

[root@node2 ~]# ansible-playbook --check httpd04.yml

PLAY [webservers] *****

TASK [Gathering Facts] *****
ok: [192.168.200.136]
ok: [192.168.200.137]

TASK [install httpd] *****
changed: [192.168.200.136]
changed: [192.168.200.137]

TASK [install configure httpd] *****
skipping: [192.168.200.137]
changed: [192.168.200.136]

TASK [install configure httpd] *****
skipping: [192.168.200.136]
changed: [192.168.200.137]

TASK [start httpd service] *****
changed: [192.168.200.136]
changed: [192.168.200.137]

RUNNING HANDLER [restart httpd service] *****
changed: [192.168.200.137]
changed: [192.168.200.136]

PLAY RECAP *****
192.168.200.136      : ok=5    changed=4    unreachable=0    failed=0
192.168.200.137      : ok=5    changed=4    unreachable=0    failed=0  @51CTO博客

```

可以看到不符合条件的跳过了

重启了服务，对于刚安装的服务时不必要的

跳过notify，让其不执行重启服务的操作，在执行playbook时使用 --skip-tags选项
值得注意的是，虽然没有执行重启，但拷贝配置文件也没有执行。

```

[root@node2 ~]# ansible-playbook --check --skip-tags="restart configure file" httpd04.yml

PLAY [webservers] *****

TASK [Gathering Facts] *****
ok: [192.168.200.136]
ok: [192.168.200.137]

TASK [install httpd] *****
changed: [192.168.200.136]
changed: [192.168.200.137]

TASK [start httpd service] *****
changed: [192.168.200.136]
changed: [192.168.200.137]

PLAY RECAP *****
192.168.200.136      : ok=3    changed=2    unreachable=0    failed=0
192.168.200.137      : ok=3    changed=2    unreachable=0    failed=0  @51CTO博客

```



执行playbook

在线
客服




```
[root@node2 ~]# ansible-playbook httpd04.yml

PLAY [webservers] *****

TASK [Gathering Facts] *****
ok: [192.168.200.136]
ok: [192.168.200.137]

TASK [install httpd] *****
changed: [192.168.200.137]
changed: [192.168.200.136]

TASK [install configure httpd] *****
skipping: [192.168.200.137]
changed: [192.168.200.136]

TASK [install configure httpd] *****
skipping: [192.168.200.136]
changed: [192.168.200.137]

TASK [start httpd service] *****
changed: [192.168.200.136]
changed: [192.168.200.137]

RUNNING HANDLER [restart httpd service] *****
changed: [192.168.200.137]
changed: [192.168.200.136]

PLAY RECAP *****
192.168.200.136 : ok=5 changed=4 unreachable=0 failed=0
192.168.200.137 : ok=5 changed=4 unreachable=0 failed=0

[root@node2 ~]# ansible webservers -m shell -a "ss -tnlp | grep :80"
192.168.200.137 | SUCCESS | rc=0 >>
ISTEN 0 128 :::8080 :::* users:({ "httpd",5355,6},{ "httpd",5361,6},{ "httpd",5362,6},{ "httpd",5363,6},{ "httpd",5364,6})
192.168.200.136 | SUCCESS | rc=0 >>
ISTEN 0 128 :::8080 :::* users:({ "httpd",5355,6},{ "httpd",5361,6},{ "httpd",5362,6},{ "httpd",5363,6},{ "httpd",5364,6})
```

并不影响结果

与我们在资产中定义的http 端口一致

@51CTO博客

注意:他的语法格式类似jinja2。

2. 循环

playbook文件

```
[root@node2 ~]# cat loop.yml
- hosts: 192.168.200.136
  remote_user: root
  tasks:
  - name: create users
    user: name={{ item }}
    with_items: [ test1, test2, test3 ]
```

调用item变量

此组件用于存放列表或者字典，它会每此取一个元素放入item变量中，以便调用

@51CTO博客



执行playbook文件并查看

```
[root@node2 ~]# ansible-playbook loop.yml

PLAY [192.168.200.136] *****

TASK [Gathering Facts] *****
ok: [192.168.200.136]

TASK [create users] *****
changed: [192.168.200.136] => (item=test1)
changed: [192.168.200.136] => (item=test2)
changed: [192.168.200.136] => (item=test3)

PLAY RECAP *****
192.168.200.136 : ok=2 changed=1 unreachable=0 failed=0

[root@node1 httpd]# id test1
uid=1001(test1) gid=100(users) groups=100(users)
[root@node1 httpd]# id test2
uid=1002(test2) gid=1002(test2) groups=1002(test2)
[root@node1 httpd]# id test3
uid=1003(test3) gid=1003(test3) groups=1003(test3)
```

执行成功，并没有报错

三个用户都已添加

@51CTO博客

判断和循环的详细请点击[这](#)

在线
客服



上一篇：Ansible常用命令 下一篇：Ansible之roles使用



karony1
27篇文章, 106W+人气, 0粉丝



提问和评论都可以，用心的回复会被更多人看到和认可

Ctrl+Enter 发布 取消 发布

3条评论 按时间正序 | 按时间倒序



wx5ba4bf93068d4
1楼 2018-09-21 17:54:31
请问下，为啥我在执行： ansible-playbook --check httpd01.yml 的时候会报： Could not find or access 'httpd.conf'的异常



wx5948b6eef3573
2楼 2018-12-05 10:08:58
好文章

1



188801943
3楼 2019-03-19 15:26:46
好文，不罗嗦。

推荐专栏 更多



基于Python的DevOps实战
自动化运维开发新概念
共20章 | 抚琴煮酒
¥51.00 387人订阅 订阅



全局视角看大型园区网
路由交换+安全+无线+优化+运维
共40章 | 51CTO夏杰
¥51.00 1214人订阅 订阅

在线客服



网工2.0晋级攻略 ——零基础入门Python/A...
网络工程师2.0进阶指南





负载均衡高手炼成记

高并发架构之路

共15章 | sery

¥51.00 473人订阅

订 阅



带你玩转高可用

前百度高级工程师的架构高可用实战

共15章 | 曹林华

¥51.00 444人订阅

订 阅

猜你喜欢

Nginx系列--04HTTP常用指令及常用模块

运维自动化-Ansible (一)

Java内部类的使用小结

自动化运维工具Ansible实战（五）Playbooks剧本使用

使用Ambari搭建Hadoop集群

Java日期时间使用总结

CentOS7 搭建企业级NFS网络文件服务器

Django+Django-Celery+Celery的整合实战

Nginx10m+高并发内核优化详解

部署SaltStack及批量安装httpd服务

安装mysql-proxy实现mysql读写分离

自动化运维工具Ansible详细部署

python下的MySQLdb使用

Zabbix中文使用手册

ansible基础学习，常用模块概述

Ansible入门与playbook实战

k8s实践6:从解决报错开始入门RBAC

Linux 性能测试工具 sysbench 的安装与简单使用

架构师的操作系统

Kubernetes共享存储之Glusterfs+Heketi



在线
客服

