

06.Ansible jinja2模板

06.Ansible jinja2模板

- 1.Ansible Jinja2模板概述
- 2.Ansible Jinja2基本使用
- 3.Ansible Jinja2管理Nginx

1.Ansible Jinja2模板概述

1.什么是jinja2

Jinja2是Python的全功能模板引擎

2.jinja2模板与Ansible有什么关系

Ansible通常会使用jinja2模板来修改被管理主机的配置文件。例如给10台远程主机都装上httpd服务，但是要求每个服务器的端口不一样，如何解决？

3.Ansible如何使用jinja2模板

使用ansible的jinja2模板，也就是template模块。该模块和copy模块一样，都是将文件复制到远端主机上去，但是区别在于template模块可以获取要复制的文件中变量的值，而copy则是原封不动的把文件内容复制过去。比如：针对不同的主机定义不同的变量，template会在将配置文件分发出去前读取变量到jinja2模板，然后分发到不同的被管理主机上。

Ansible使用jinja注意事项

Ansible允许jinja2模板中使用条件判断和循环，但是jinja判断循环语法不允许在playbook中使用。

注意：不是每个管理员都需要这个特性，但是有些时候jinja2模板能大大提高效率。

2.Ansible Jinja2基本使用

1.jinja模板基本语法

- 1) playbook文件使用template参数
- 2) 模板文件里面变量使用{名称}, 比如{PORT}}或使用facts。

2.jinja模板逻辑关系

{% for i in EXPR %}...{% endfor %} 作为循环表达式

{% if EXPR %}...{% elif EXPR %}...{% endif %} 作为条件判断

{# COMMENT #} 表示注释

3.jinja模板使用示例、使用fact变量的示例

- 1) 使用Playbook推送文件

```
[root@m01 playbook]# cat jinja2.yml
- hosts: web
  tasks:
    - name: Copy Template File /etc/motd
      template: src=./motd.j2 dest=/etc/motd
```

- 2) 准备motd.j2文件

```
[root@m01 playbook]# cat motd.j2
welcome to {{ ansible_hostname }}
This system total Memory is: {{ ansible_memtotal_mb }} MB
This system free Memory is: {{ ansible_memfree_mb }} MB
```

- 3) 执行playbook

```
[root@m01 playbook]# ansible-playbook jinja2.yml

PLAY [web]
*****
*****

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

ok: [172.16.1.8]
ok: [172.16.1.7]
```

```
TASK [Copy Template File /etc/motd]
*****

*****

changed: [172.16.1.8]
changed: [172.16.1.7]

PLAY RECAP
*****

*****

172.16.1.7                : ok=2    changed=1
unreachable=0    failed=0
172.16.1.8                : ok=2    changed=1
unreachable=0    failed=0
```

4. 检查执行后的状态

```
[root@m01 playbook]# ssh root@172.16.1.7
welcome to web01
This system total Memory is: 470 MB
This system free Memory is: 193 MB

[root@m01 playbook]# ssh root@172.16.1.8
welcome to web02
This system total Memory is: 470 MB
This system free Memory is: 198 MB
```

5)总结 上面的例子展示了如何使用facts变量, 当相关的playbook被执行后, `ansible_hostname`和`ansible_memtotal_mb`将会被替换成被管理主机上搜集的facts变量的值

3.Ansible Jinja2管理Nginx

ansible使用jinja2的for循环表达式渲染出nginx负载均衡的配置文件

1) 使用Playbook推送文件

```
[root@m01 playbook]# cat proxy.yml
- hosts: web
  vars:
    http_port: 80
    server_name: www.oldboy.com
```

```

tasks:
  - name: Copy Template Nginx Configure
    template: src=blog.conf.j2
    dest=/etc/nginx/conf.d/blog.oldboy.com.conf
    notify: Reload Nginx Server

handlers:
  - name: Reload Nginx Server
    service: name=nginx state=reloaded

```

2) 准备blog.conf.j2配置文件

```

[root@m01 playbook]# cat blog.conf.j2
upstream {{ server_name }} {
#设置变量，并进行循环赋值，渲染配置
  {% for i in range(7,10) %}
    server 172.16.1.{{i}}:{{ http_port }};
  {% endfor %}
}

server {
  listen  {{ http_port }};
  server_name {{ server_name }};
  location / {
    proxy_pass http://{{ server_name }};
    proxy_set_header Host $http_host;
  }
}

```

3) 执行playbook

```

[root@m01 playbook]# ansible-playbook proxy.yml

PLAY [web]
*****
*****

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

```

```
ok: [172.16.1.7]
```

```
ok: [172.16.1.8]
```

```
TASK [Copy Template Nginx Configure]
```

```
*****
```

```
*****
```

```
changed: [172.16.1.7]
```

```
changed: [172.16.1.8]
```

```
PLAY RECAP
```

```
*****
```

```
*****
```

```
172.16.1.7 : ok=2 changed=1
```

```
unreachable=0 failed=0
```

```
172.16.1.8 : ok=2 changed=1
```

```
unreachable=0 failed=0
```

4.检查使用jinja模板渲染出来的配置文件

```
[root@web02 ~]# cat /etc/nginx/conf.d/blog.oldboy.com.conf
```

```
upstream www.oldboy.com {
```

```
#设置变量，并进行循环赋值,渲染配置
```

```
server 172.16.1.7:80;
```

```
server 172.16.1.8:80;
```

```
server 172.16.1.9:80;
```

```
}
```

```
server {
```

```
listen 80;
```

```
server_name www.oldboy.com;
```

```
location / {
```

```
proxy_pass http://www.oldboy.com;
```

```
proxy_set_header Host $http_host;
```

```
}
```

```
}
```