

第3天-自动化运维利器 Ansible-Jinja2模板

一、Ansible Jinja2模板背景介绍

目前Nginx的配置文件在所有的服务器上都是相同的,但我希望能根据每一台服务器的性能去定制服务的启动进程。同时定制每一台Nginx服务的响应头,以便于当某台服务出现问题时能快速定位到具体的服务器。要做这样的定制势必会导致一个问题, Nginx 在每台物理服务器上的配置文件都不一样,这样的配置文件如何管理呢? 再使用copy 模块去做管理显然已经不合适。此时使用Ansible 提供的另一个模板(template) 功能, 它可以帮助我们完美的解决问题。

二、Jinja2 模板

要学会Ansible 中的模板(template)使用, 前提我们必须要学会Jinja2模板。学会了它, 就相当于我们学会了Ansible 模板。

1. Jinja2 是什么

Jinja2是基于Python书写的模板引擎。功能比较类似于PHP的smarty模板。

2. Jinja2 必知必会

1. jinja2 文件以 `.j2` 为后缀, 也可以不写后缀。
2. jinja2 中存在 三种定界符
 - 注释: `{# 注释内容 #}`

- 变量引用: `{{ var }}`
- 逻辑表达: `{% %}`

3. Jinja2 逻辑控制

3.1 条件表达

```
1  {% if %}  
2  ...  
3  {% elif %}  
4  ...  
5  {% else %}  
6  ...  
7  {% endif %}
```

Example

```
1  {# 如果定义了 idc 变量, 则输出 #}  
2  {% if idc is defined %}  
3  {{ idc }}  
4  {% elif %}  
5      没有定义  
6  {% endif %}
```

3.2 循环控制

```
1  {% for %}  
2  ...  
3  ...  
4  {% endfor %}
```

Example

```
1  {# 列举出 dbservers 这个 group 中的所有主机 #}  
2  {% for host in groups['dbservers'] %}  
3  {{ host }}  
4  {% endfor %}
```

```
1  {#与Python 语法不通，模板中的循环内不能break或continue。但  
    你可以在迭代中过滤序列来跳过某些项#}  
2  {#打印dbservers 组中的所有主机，但是不打印1.1.1.1 这台主机  
    #}  
3  {% for host in groups['dbservers'] if host !=  
    "1.1.1.1" %}  
4  {{host}}  
5  {% endfor %}
```

三、如何使用模板

一个基于Facts的Jinja2 实例

```
1  # cat config.j2  
2  {# use variable example #}  
3  welcome host {{ ansible_hostname }}, os is {{  
    ansible_os_family }}  
4  today is {{ ansible_date_time.date }}  
5  cpucore numbers {{ ansible_processor_vcpus }}  
6  
7  {# use condition example #}  
8  {% if ansible_processor_vcpus > 1 %}  
9  OS CPU more than one core  
10 {% endif %}  
11  
12 {% for m in ansible_mounts if m['mount'] != "/" %}
```

```
13 mount {{ m['mount'] }}, total size is
    {{m['size_total']}}, free size is
    {{m['size_available']}}
14 {% endfor %}
15
```

在Ansible 中使用模板

```
1 ---
2 - name: a template example
3   hosts: all
4   remote_user: root
5   tasks:
6     - name: update jinja2 config
7       template: src=config.j2 dest=/tmp/config.conf
```

四、 实例演示

Jinja2 模板以及如何在Ansible中使用模板，已经介绍完了。那么如何去实现我们的需求呢？

```
1 # cat nginx.conf.j2
2 user          nginx;
3 {# start process equal cpu cores #}
4 worker_processes {{ ansible_processor_vcpus }};
5
6 error_log      /var/log/nginx/error.log;
7 pid           /var/run/nginx.pid;
8
9 events {
10     worker_connections 1024;
11 }
12
13 http {
```

```

14     include          /etc/nginx/mime.types;
15     default_type     application/octet-stream;
16
17     log_format main '$remote_addr - $remote_user
[$time_local] "$request" '
18                     '$status $body_bytes_sent
"$http_referer" '
19                     '"$http_user_agent"
"$http_x_forwarded_for"';
20
21     sendfile          on;
22     tcp_nopush        on;
23
24     keepalive_timeout 0;
25
26     gzip on;
27     gzip_min_length   1k;
28     gzip_buffers       8 64k;
29     gzip_http_version 1.0;
30     gzip_comp_level    5;
31     gzip_types         text/plain application/x-javascript
text/css application/json application/xml
application/x-shockwave-flash application/javascript
image/svg+xml image/x-icon;
32     gzip_vary on;
33     {# add_header {{ ansible_hostname }}; #}
34     add_header x-hostname {{ ansible_hostname }};
35
36     include /etc/nginx/conf.d/*.conf;
37 }

```

继续优化我们的PlayBook, 让它支持模板

```

1 - name: template playbook example
2     hosts: webservers

```

```
3 vars:
4   createuser:
5     - tomcat
6     - www
7     - mysql
8 tasks:
9   - name: create user
10     user: name={{ item }} state=present
11     with_items: "{{ createuser }}"
12
13   - name: yum nginx webserver
14     yum: name=nginx state=present
15
16     # use ansible template
17   - name: update nginx main config
18     template:
19       src: nginx.conf.j2
20       dest: /etc/nginx/nginx.conf
21       tags: updateconfig
22       notify: reload nginx server
23
24   - name: add virtualhost config
25     copy:
26       src: www.qfedu.com.conf
27       dest: /etc/nginx/conf.d/
28       tags: updateconfig
29       notify: reload nginx server
30
31   - name: check nginx syntax
32     shell: /usr/sbin/nginx -t
33     register: nginxsyntax
34     tags: updateconfig
35
36   - name: check nginx running
37     stat: path=/var/run/nginx.pid
```

```
38     register: nginxrunning
39     tags: updateconfig
40
41     - name: print nginx syntax
42       debug: var=nginxsyntax
43
44     - name: start nginx server
45       service: name=nginx state=started
46       when:
47         - nginxsyntax.rc == 0
48         - nginxrunning.stat.exists == false
49 handlers:
50     - name: reload nginx server
51       service: name=nginx state=started
52       when:
53         - nginxsyntax.rc == 0
54         - nginxrunning.stat.exists == true
```

执行还是按照原来的方式执行即可

```
1 | ansible-playbook -i hosts site.yml
```