PLAYBOOK配置文件

1.执行配置文件

2.触发器

前期环境配置

3.角色 --关于某一个服务的配置放到一起，就形成了角色

实验案例 mariadb

4.变量

4.1、在playbook中使用自定义变量

4.2、在playbook中使用ansible 的内置变量

5.Template模板

5.1、利用template模块下发可变的配置文件

5.2、下发配置文件里面使用判断语法

5.3、Playbook 的notify通知和下发nginx配置

6.基于Playbook部署Nginx综合案例

1、创建目录结构

2、查看目录结构

3、定义一个主调用文件

4、files:存储由copy或script等模块调用的文件;

5、handlers: 此目录中至少应该有一个名为main.yml的文件，用于定义各handler;其它的文件需要由main.yml进行“包含”调用;

6、tasks:目录至少应该有一个名为main.yml的文件，用于定义各task;其它的文件需要由main.yml进行“包含”调用;

7、templates: 存储由template模块调用的模板文本;

8、vars: 此目录中至少应该有一个名为main.yml的文件，用于定义各variable; 其它的文件而; 要由main.yml进行“包含”调用;

9、其他:

10、测设部署

PHP

PLAYBOOK配置文件

1.执行配置文件

2.触发器

3. 角色

4. 变量

5. TEMPLATE模板

6. 基于PLAYBOOK部署NGINX综合案例

前期环境配置

```
[root@ansible ~]# vim /etc/ansible/hosts
[sofia]
192.168.200.108
192.168.200.109
[root@ansible ~]# ssh-keygen -t rsa
[root@ansible ~]# ssh-copy-id root@192.168.200.108
[root@ansible ~]# ssh-copy-id root@192.168.200.109
[root@ansible ~]# vim /etc/hosts
192.168.200.107 ansible
192.168.200.107 client1
192.168.200.107 client2
[root@ansible ~]# ansible sofia -m copy -a "src=/etc/hosts dest=/etc/"
```

改主机名：

```
[root@ansible ~]# ansible 192.168.200.108 -m hostname -a "name=client1"
[root@ansible ~]# ansible 192.168.200.109 -m hostname -a "name=client2"
```

3.角色　--关于某一个服务的配置放到一起，就形成了角色

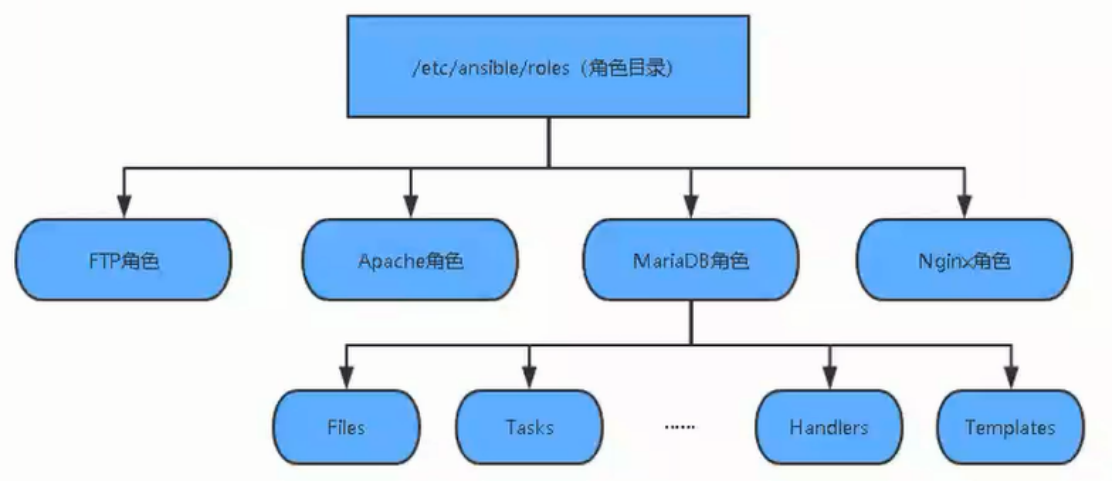
将多种不同的tasks的文件集中存储在某个目录下，则该目录就是角色

角色一般存放在/etc/ansible/roles/目录中

可通过ansible 的配置文件来调整默认的角色目录

/etc/ansible/roles目录下有很多的子目录，其中每一个子目录对应一个角色。

每个角色也有自己的目录结构



/etc/ansible/roles/为角色集合，该目录下有自定义的各个子目录

1. mariadb: mysql 角色
2. apache: httpd 角色
3. nginx: nginx 角色

每个角色的定义，以特定的层级目录结构进行组织，以Mariadb (mysql 角色)为例

1. files: 存放copy或script等模块调用的文件

（nginx.conf,nginx源码包）--固定，静态文件，脚本，一般不会改变

2. templates: 存放template模块查找所需要的*模板文件*的目录，如mysql配置文件等模板
3. tasks: 任务存放目录
4. handlers: 存放相关触发执行器的目录
5. vars: 变量存放的目录
6. meta:用于存放此角色元数据
7. default: 默认变量存放目录，文件中定义了此角色使用的默认变量

上述目录中tasks, handlers， vars， meta， default至少应该包含一个main.yml， 该目录

下也可以有其他的.yml文件，但是需要在main.yml文件中用include指定将其他.yml文件包含进来。

有了角色之后，可以直接在 yaml文件中(playbook 配置文件)中调用角色示例如下

```
- hosts: test01
  remote user: root
  roles:
     - mysql                  //调用角色名
     - httpd                   //调用角色名
```

可以只调用一个角色，也可以调用多个角色，当定义了角色后，

用ansible-playbookPLAYBOOK文件执行即可，

此时ansible会到角色集合的目录(/etc/ansible/roles) 去找mysql和httpd目录，

然后依次运行mysql目录和httpd目录下的所有代码

实验案例　　mariadb

下面通过一个实例配置数据库角色，要求被管理主机自动安装Mariadb

安装完成后上传提前准备好的配置文件到远端主机

重启服务，然后新建testdb数据库，并允许test用户对其拥有所有权限

1.被管理端配置yum源

```
[root@client1 ~]# cd /etc/yum.repos.d/
[root@client1 yum.repos.d]# yum clean all && yum makecache fast
```

2.配置数据库角色

```
[root@ansible ~]# mkdir -pv /etc/anxible/roles/mariadb/{files,tasks,handlers}
[root@ansible ansible]# mkdir -pv
/etc/ansible/roles/mariadb/{files,tasks,handlers}
mkdir: 已创建目录 "/etc/ansible/roles/mariadb"
mkdir: 已创建目录 "/etc/ansible/roles/mariadb/files"
mkdir: 已创建目录 "/etc/ansible/roles/mariadb/tasks"
mkdir: 已创建目录 "/etc/ansible/roles/mariadb/handlers"
[root@ansible ~]# cd /etc/ansible/
[root@ansible ansible]# vim mariadb.yml          # 统一入口文件，总文件
- hosts: sofia
  remote_user: root
  roles:
    - mariadb
```

检查语法：

```
[root@ansible ansible]# ansible-playbook --syntax-check /etc/ansible/mariadb.yml
playbook: /etc/ansible/mariadb.yml
[root@ansible ansible]# vim roles/mariadb/tasks/main.yml
```

```
- name: install mariadb
  yum: name=mariadb-server state=present
- name: move config file
  shell: "[ -e /etc/my.cnf ] && mv /etc/my.cnf /etc/my.cnf.bak"
- name: provide a new config file
  copy: src=my.cnf dest=/etc/my.cnf
- name: reload mariadb
  shell: systemctl restart mariadb
- name: create database testdb
  shell: mysql -u root -e "create database testdb;grant all privileges on
testdb.* to 'test'@'192.168.200.%' identified by 'test123';flush privileges;"
notify:
  - restart mariadb
```

触发器：

```
[root@ansible ansible]# vim roles/mariadb/headlers/main.yml
- name: restart mariadb
  service: name=mariadb state=restarted
[root@ansible ansible]# cd roles/mariadb/files/
[root@ansible files]# cp /etc/my.cnf ./
[root@ansible ansible]# tree
.
├── ansible.cfg
├── hosts
├── mariadb.yml
└── roles
    └── mariadb
        ├── files
        │   └── my.cnf
        ├── handlers
        │   └── main.yml
        └── tasks
            └── main.yml
```

预执行：

```
[root@ansible ansible]# ansible-playbook -C /etc/ansible/mariadb.yml
```

```
PLAY [sofia] ***********************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************
ok: [192.168.200.109]
ok: [192.168.200.108]

TASK [mariadb : install mariadb] ***************************************************************************************
changed: [192.168.200.109]
changed: [192.168.200.108]

TASK [mariadb : move config file] **************************************************************************************
skipping: [192.168.200.108]
skipping: [192.168.200.109]

TASK [mariadb : provide a new config file] *****************************************************************************
changed: [192.168.200.108]
changed: [192.168.200.109]

TASK [mariadb : reload mariadb] ****************************************************************************************
skipping: [192.168.200.109]
skipping: [192.168.200.108]

TASK [mariadb : create database testdb] ********************************************************************************
skipping: [192.168.200.108]
skipping: [192.168.200.109]

PLAY RECAP *************************************************************************************************************
192.168.200.108            : ok=3    changed=2    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.200.109            : ok=3    changed=2    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
```

执行：

```
[root@ansible ansible]# ansible-playbook /etc/ansible/mariadb.yml
```

测试查看：

```
[root@client1 ~]# mysql
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| test               |
| testdb             |
+--------------------+
5 rows in set (0.00 sec)
MariaDB [(none)]> show grants for test@'192.168.200.%';
+-----------------------------------------------------------------------------------------------------------------+
| Grants for test@192.168.200.%
|
+-----------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO 'test'@'192.168.200.%' IDENTIFIED BY PASSWORD
'*676243218923905CF94CB52A3C9D3EB30CE8E20D' |
| GRANT ALL PRIVILEGES ON `testdb`.* TO 'test'@'192.168.200.%'
|
```

```
+-----------------------------------------------------------------------------------
----------------------------------+
2 rows in set (0.00 sec)
```

4.变量

4.1、在playbook中使用自定义变量

```
[root@ansible ansible]# vim /etc/ansible/test_vars.yml
- hosts: sofia
  vars:                              # 定义变量
    - name: "cloud"                 # 第一个name变量
      age: "3"                       # 第二个name变量
  tasks:
    - name: "{{name}}"                # {{}} 两对大括号引用变量，变量名两头空格，好像没有也没关系
      shell: echo "myname {{name}}, myage {{age}}"
      register: var_result
    - debug: var=var_result
```

语法检查：

```
[root@ansible ansible]# ansible-playbook --syntax-check
/etc/ansible/test_vars.yml
playbook: /etc/ansible/test_vars.yml
[root@ansible ansible]# ansible-playbook -C /etc/ansible/test_vars.yml
```

---

ps ： 我发现语法检查通过，预执行通过，但真正执行时会报错，

```
TASK [cloud] *******************************************************************************************************
fatal: [192.168.200.108]: FAILED! => {"changed": true, "cmd": "myname cloud, myage 3", "delta": "0:00:00.004501", "end": "2
020-05-11 00:46:05.946342", "msg": "non-zero return code", "rc": 127, "start": "2020-05-11 00:46:05.941841", "stderr": "/bi
n/sh: myname: 未找到命令", "stderr_lines": ["/bin/sh: myname: 未找到命令"], "stdout": "", "stdout_lines": []}
fatal: [192.168.200.109]: FAILED! => {"changed": true, "cmd": "myname cloud, myage 3", "delta": "0:00:00.004729", "end": "2
020-05-10 16:46:06.617970", "msg": "non-zero return code", "rc": 127, "start": "2020-05-10 16:46:06.613241", "stderr": "/bi
n/sh: myname: 未找到命令", "stderr_lines": ["/bin/sh: myname: 未找到命令"], "stdout": "", "stdout_lines": []}

PLAY RECAP *********************************************************************************************************
192.168.200.108            : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
192.168.200.109            : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
```

这里的原因是：少敲了“echo”

执行：

```
[root@ansible ansible]# ansible-playbook /etc/ansible/test_vars.yml
```

```
[root@ansible ansible]# ansible-playbook /etc/ansible/test_vars.yml
[WARNING]: Found variable using reserved name: name

PLAY [sofia] ***************************************************************************************************************

TASK [Gathering Facts] *****************************************************************************************************
ok: [192.168.200.108]
ok: [192.168.200.109]

TASK [cloud] ***************************************************************************************************************
changed: [192.168.200.109]
changed: [192.168.200.108]

TASK [debug] ***************************************************************************************************************
ok: [192.168.200.108] => {
    "var_result": {
        "changed": true,
        "cmd": "echo \"myname cloud, myage 3\"",
        "delta": "0:00:00.004547",
        "end": "2020-05-11 00:47:23.016060",
        "failed": false,
        "rc": 0,
        "start": "2020-05-11 00:47:23.011513",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "myname cloud, myage 3",
        "stdout_lines": [
            "myname cloud, myage 3"
        ]
    }
}
ok: [192.168.200.109] => {                          这两个地方还有点问题，需要后期改
    "var_result": {
        "changed": true,
        "cmd": "echo \"myname cloud, myage 3\"",
        "delta": "0:00:00.005127",
        "end": "2020-05-10 16:47:23.686733",
        "failed": false,
        "rc": 0,
        "start": "2020-05-10 16:47:23.681606",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "myname cloud, myage 3",
        "stdout_lines": [
            "myname cloud, myage 3"
        ]
    }
}

            "myname cloud, myage 3"
        ]
    }
}
```

4.2、在playbook中使用ansible 的内置变量

使用ansible all -m setup | more 查看ansible内置变量

```
[root@ansible ~]# cat /etc/ansible/test_setupvars.yml
- hosts: sofia
  gather_facts: True            #使用ansible内置变量
  tasks:
  - name: setup var
    shell: echo "ip {{ ansible_all_ipv4_addresses[1] }} {{
ansible_processor_count }}"
    register: var_result
  - debug: var=var_result
[root@ansible ~]# ansible-playbook --syntax-check
/etc/ansible/test_setupvars.yml
```

有时候还是要看一下报错的。。。

预执行：

```
[root@ansible ~]# ansible-playbook -C /etc/ansible/test_setupvars.yml
```

执行：

```
[root@ansible ~]# ansible-playbook /etc/ansible/test_setupvars.yml
```

```
PLAY RECAP *********************************************************************************************************
192.168.200.108            : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.200.109            : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

[root@ansible ~]# ansible-playbook --syntax-check /etc/ansible/test_setupvars.yml

playbook: /etc/ansible/test_setupvars.yml
[root@ansible ~]# ansible-playbook /etc/ansible/test_setupvars.yml

PLAY [sofia] *******************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.200.108]
ok: [192.168.200.109]

TASK [setup var] ***************************************************************************************************
changed: [192.168.200.108]
changed: [192.168.200.109]

TASK [debug] *******************************************************************************************************
ok: [192.168.200.108] => {
    "var_result": {
        "changed": true,
        "cmd": "echo \"ip 192.168.200.108 1\"",
        "delta": "0:00:00.004518",
        "end": "2020-05-11 01:09:45.381975",
        "failed": false,
        "rc": 0,
        "start": "2020-05-11 01:09:45.377457",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "ip 192.168.200.108 1",
        "stdout_lines": [
            "ip 192.168.200.108 1"
        ]
    }
}
ok: [192.168.200.109] => {
    "var_result": {
        "changed": true,
        "cmd": "echo \"ip 192.168.200.109 1\"",
        "delta": "0:00:00.005369",
        "end": "2020-05-10 17:09:46.057422",
        "failed": false,
        "rc": 0,
        "start": "2020-05-10 17:09:46.052053",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "ip 192.168.200.109 1",
        "stdout_lines": [
            "ip 192.168.200.109 1"
        ]
    }
}

PLAY RECAP *********************************************************************************************************
192.168.200.108            : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.200.109            : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[root@ansible ~]#
```

[root@ansible ~]# vim /etc/ansible/test1_setupvars.yml

[root@ansible ~]# ansible-playbook /etc/ansible/test1_setupvars.yml

```
- hosts: sofia
  gather_facts: True
  tasks:
  - name: setup var
    shell: echo "ip {{ ansible_all_ipv4_addresses[1] }} cpu {{ ansible_processor_count }}" >> /tmp/test
  - name: setup var2
    shell: echo "time {{ ansible_date_time['date']}}" >> /tmp/test
    register: var_result
  - debug: var=var_result
```

```
PLAY [sofia] ***************************************************************************************************

TASK [Gathering Facts] *****************************************************************************************
ok: [192.168.200.109]
ok: [192.168.200.108]

TASK [setup var] ***********************************************************************************************
changed: [192.168.200.109]
changed: [192.168.200.108]

TASK [setup var2] **********************************************************************************************
changed: [192.168.200.109]
changed: [192.168.200.108]

TASK [debug] ***************************************************************************************************
ok: [192.168.200.108] => {
    "var_result": {
        "changed": true,
        "cmd": "echo \"time 2020-05-11\" >> /tmp/test",
        "delta": "0:00:00.004602",
        "end": "2020-05-11 01:23:13.214050",
        "failed": false,
        "rc": 0,
        "start": "2020-05-11 01:23:13.209448",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "",
        "stdout_lines": []
    }
}
ok: [192.168.200.109] => {
    "var_result": {
        "changed": true,
        "cmd": "echo \"time 2020-05-10\" >> /tmp/test",
        "delta": "0:00:00.004478",
        "end": "2020-05-10 17:23:13.884114",
        "failed": false,
        "rc": 0,
        "start": "2020-05-10 17:23:13.879636",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "",
        "stdout_lines": []
    }
}

PLAY RECAP *****************************************************************************************************
192.168.200.108            : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.200.109            : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[root@ansible ~]# vim /etc/ansible/test1_setupvars.yml
[root@ansible ~]#
```

5.Template模板

配置文件如果使用copy模块去下发的话

那所有主机的配置都是一样的;如果下发的配置文件里

有可变的配置，需要用到template模块

5.1、利用template模块下发可变的配置文件

```
[root@ansible ~]# vim /tmp/test
my name is {{ myname }}                    # 自定义变量
my ip is {{ ansible_all_ipv4_addresses[1] }}          # 系统变量
[root@ansible ~]# vim /etc/ansible/filevars.yml
- hosts: sofia
  gather_facts: True
  vars:
  - myname: "cloud"
  tasks:
  - name: xxx
    template: src=/tmp/test dest=/root/test
[root@ansible ~]# ansible-playbook --syntax-check /etc/ansible/filevars.yml
playbook: /etc/ansible/filevars.yml
[root@ansible ~]# ansible-playbook -C /etc/ansible/filevars.yml
[root@ansible ~]# ansible-playbook /etc/ansible/filevars.yml
```

```
PLAY [sofia] ***************************************************************************************************

TASK [Gathering Facts] *****************************************************************************************
ok: [192.168.200.109]
ok: [192.168.200.108]

TASK [xxx] *****************************************************************************************************
changed: [192.168.200.109]
changed: [192.168.200.108]

PLAY RECAP *****************************************************************************************************
192.168.200.108            : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.200.109            : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

```
[root@client1 ~]# cat test
my name is cloud
my ip is 192.168.200.108
[root@client2 ~]#  cat test
my name is cloud
my ip is 192.168.200.109
```

5.2、下发配置文件里面使用判断语法

```
[root@ansible ~]# vim /tmp/if.j2                # 模板 jinjia2
{% if PORT %}                    # if PORT 存在
ip=0.0.0.0:{{ PORT }}
{% else %}                            # 否则的话
ip=0.0.0.0:80
{% endif %}
```

```
[root@ansible ~]# vim /etc/ansible/test_ifvars.yml
- hosts: all
  gather_facts: True            # 开启系统内置变量
  vars:
  - PORT: 90              # 自定义变量
  tasks:
  - name: jinjia2 if test
    template: src=/tmp/if.j2 dest=/root/test
[root@ansible ~]# ansible-playbook --syntax-check /etc/ansible/test_ifvars.yml
playbook: /etc/ansible/test_ifvars.yml
[root@ansible ~]# ansible-playbook -C /etc/ansible/test_ifvars.yml
[root@ansible ~]# ansible-playbook /etc/ansible/test_ifvars.yml
```

```
PLAY [all] *****************************************************************************************************

TASK [Gathering Facts] *****************************************************************************************
ok: [192.168.200.108]
ok: [192.168.200.109]

TASK [jinjia2 if test] *****************************************************************************************
changed: [192.168.200.108]
changed: [192.168.200.109]

PLAY RECAP *****************************************************************************************************
192.168.200.108            : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.200.109            : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

测试查看：

```
[root@client1 ~]# cat test
ip=0.0.0.0:90
[root@client2 ~]# cat test
ip=0.0.0.0:90
```

如果端口不设值：

```
[root@ansible ~]# vim /etc/ansible/test_ifvars.yml
- hosts: all
  gather_facts: True
  vars:
  - PORT:
  tasks:
  - name: jinjia2 if test
    template: src=/tmp/if.j2 dest=/root/test
[root@ansible ~]# ansible-playbook /etc/ansible/test_ifvars.yml
```

测试查看：

```
[root@client1 ~]# cat test
ip=0.0.0.0:80
[root@client2 ~]# cat test
ip=0.0.0.0:80
```

5.3、Playbook 的notify通知和下发nginx配置

实战下发可执行动作的可变的nginx配置文件

```
[root@ansible ~]# cp nginx.conf /tmp/nginx.j2
[root@ansible ~]# head -3 /tmp/nginx.j2

#user  nobody;
worker_processes  {{ ansible_processor_vcpus }};  #可变的参数

[root@ansible ~]# cat /etc/ansible/test_nginxvars.yml
---
- hosts: all
  gather_facts: True      #开启系统内置变量
  tasks:
  - name: nginx conf
    template: src=/tmp/nginx.j2 dest=/usr/local/nginx/conf/nginx.conf
    notify:
    - reload nginx   #下发通知给 handlers 模块执行名字叫做 reload nginx 的动作
  handlers: #定义动作
  - name: reload nginx   #动作的名字
    shell: /usr/local/nginx/sbin/nginx -s reload
...
[root@ansible ~]# ansible-playbook /etc/ansible/test_nginxvars.yml
```

6.基于Playbook部署Nginx综合案例

1、创建目录结构

```
[root@ansible ~]# mkdir -pv
/etc/ansible/roles/nginx/{files,handlers,tasks,templates,vars}
mkdir: 已创建目录 "/etc/ansible/roles/nginx"
mkdir: 已创建目录 "/etc/ansible/roles/nginx/files"
mkdir: 已创建目录 "/etc/ansible/roles/nginx/handlers"
mkdir: 已创建目录 "/etc/ansible/roles/nginx/tasks"
mkdir: 已创建目录 "/etc/ansible/roles/nginx/templates"
mkdir: 已创建目录 "/etc/ansible/roles/nginx/vars"
```

2、查看目录结构

```
[root@ansible ~]# tree /etc/ansible/roles/nginx/
/etc/ansible/roles/nginx/
├── files
│   └── nginx-1.16.0.tar.gz
├── handlers
│   └── main.yml
├── tasks
│   └── main.yml
```

```
├──── templates
│       └──── nginx.conf
└──── vars
    └──── main.yml
```

3、定义一个主调用文件

```
[root@ansible ~]# vim /etc/ansible/nginx.yml
- hosts: sofia
  gather_facts: True
  remote_user: root
  roles:
  - nginx
```

4、files:存储由copy或script等模块调用的文件;

```
[root@ansible ~]# vim /etc/ansible/nginx.yml
[root@ansible ~]# cd /etc/ansible/roles/nginx/
[root@ansible nginx]# cd files/
[root@ansible files]# rz
[root@ansible files]# ls
nginx-1.16.0.tar.gz
```

5、handlers: 此目录中至少应该有一个名为main.yml的文件，用于定义各handler;其它的文件需要由main.yml进行“包含”调用;

```
[root@ansible files]# cd ../
[root@ansible nginx]# vim handlers/main.yml
- name: start nginx
  raw: /usr/local/nginx/sbin/nginx
```

6、tasks:目录至少应该有一个名为main.yml的文件，用于定义各task;其它的文件需要由main.yml进行“包含”调用;

```
[root@ansible nginx]# vim tasks/main.yml
- name: yum install
  yum: name={{ item }} state=latest
  with_items:
    - openssl-devel
    - pcre-devel
```

```
    - zlib-devel
    - gcc
    - gcc-c++
    - make
- name: user nginx
  shell: useradd -M -s /sbin/nologin nginx
- name: package
  copy: src=nginx-1.16.0.tar.gz dest=/usr/src
- name: install nginx
  shell: cd /usr/src ; tar xf nginx-1.16.0.tar.gz -C /usr/src/ ; cd
/usr/src/nginx-1.16.0 ; ./configure --prefix=/usr/local
/nginx --user=nginx --group=nginx --with-http_ssl_module --with-http_flv_module
--with-http_stub_status_module --with-http_gzip_static_module --with-pcre &&
make && make install
- name: copy conf file
  template: src=nginx.conf dest=/usr/local/nginx/conf/nginx.conf
  notify:
    - start nginx
```

7、templates: 存储由template模块调用的模板文本;

```
[root@ansible nginx]# vim templates/nginx.conf
user  nginx;
worker_processes  {{ ansible_processor_vcpus }};
{% if ansible_processor_vcpus == 1 %}
worker_cpu_affinity 10;
{% elif ansible_processor_vcpus == 2 %}
worker_cpu_affinity 01 10;
{% elif ansible_processor_vcpus == 4 %}
worker_cpu_affinity 0001 0010 0100 1000;
{% elif ansible_processor_vcpus == 8 %}
worker_cpu_affinity 00000001 00000010 00000100 00001000 00010000 00100000
01000000 10000000;
{% else %}
worker_cpu_affinity 0001 0010 0100 1000;
{% endif %}
```

```
error_log  logs/error.log;
pid        logs/nginx.pid;


events {
    use epoll;
    worker_connections  65535;
}


http {
    include       mime.types;
    default_type  application/octet-stream;

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  logs/access.log  main;

    sendfile        on;
    keepalive_timeout  65;
    gzip  on;

    server {
        listen       {{ nginxport }};
        server_name  {{ server_name}};

        location / {
            root   html;
            index  index.html index.htm;
        }

        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
```

```
            root   html;
        }
    }
}
```

8、vars: 此目录中至少应该有一个名为main.yml的文件，用于定义各variable; 其它的文件而; 要由main.yml进行“包含”调用;

```
[root@ansible nginx]# vim vars/main.yml
nginxport: "80"
server_name: "www.sofia.com"
```

9、其他:

meta:此目录中至少应该有--一个名为main.yml的文件，定义当前角色的特殊设定及其依赖关系;其它的文件需要由main.yml进行“包含”调用;

default:此目录中至少应该有一个名为main.yml的文件，用于设定默认变量;

10、测设部署

预执行

```
[root@ansible ~]# ansible-playbook -C /etc/ansible/nginx.yml
```

执行：

```
[root@ansible ~]# ansible-playbook /etc/ansible/nginx.yml

PLAY [sofia] *******************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************
ok: [192.168.200.108]
ok: [192.168.200.109]

TASK [nginx : yum install] *****************************************************************************************************
[DEPRECATION WARNING]: Invoking "yum" only once while using a loop via squash_actions is deprecated. Instead of using a loop to
supply multiple items and specifying `name: "{{ item }}"`, please use `name: ['openssl-devel', 'pcre-devel', 'zlib-devel', 'gcc',
'gcc-c++', 'make']` and remove the loop. This feature will be removed in version 2.11. Deprecation warnings can be disabled by
setting deprecation_warnings=False in ansible.cfg.
[DEPRECATION WARNING]: Invoking "yum" only once while using a loop via squash_actions is deprecated. Instead of using a loop to
supply multiple items and specifying `name: "{{ item }}"`, please use `name: ['openssl-devel', 'pcre-devel', 'zlib-devel', 'gcc',
'gcc-c++', 'make']` and remove the loop. This feature will be removed in version 2.11. Deprecation warnings can be disabled by
setting deprecation_warnings=False in ansible.cfg.
ok: [192.168.200.108] => (item=[u'openssl-devel', u'pcre-devel', u'zlib-devel', u'gcc', u'gcc-c++', u'make'])
ok: [192.168.200.109] => (item=[u'openssl-devel', u'pcre-devel', u'zlib-devel', u'gcc', u'gcc-c++', u'make'])

TASK [nginx : user nginx] ******************************************************************************************************
changed: [192.168.200.109]
changed: [192.168.200.108]

TASK [nginx : package] *********************************************************************************************************
changed: [192.168.200.108]
changed: [192.168.200.109]

TASK [nginx : install nginx] ***************************************************************************************************
changed: [192.168.200.109]
changed: [192.168.200.108]

TASK [nginx : copy conf file] **************************************************************************************************
changed: [192.168.200.109]
changed: [192.168.200.108]

RUNNING HANDLER [nginx : start nginx] ******************************************************************************************
changed: [192.168.200.108]
changed: [192.168.200.109]

PLAY RECAP *********************************************************************************************************************
192.168.200.108            : ok=7    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.200.109            : ok=7    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[root@ansible ~]#
```


访问测试

PHP

```
[root@ansible ~]# cd /etc/ansible/
[root@ansible ansible]# cp nginx.yml php.yml
[root@ansible ansible]# vim php.yml
- hosts: sofia
  gather_facts: True
  remote_user: root
  roles:
  - php
[root@ansible ansible]# cd roles/
[root@ansible roles]# ls
mariadb  nginx
[root@ansible roles]# cp -r nginx/ php
[root@ansible roles]# cd php/
[root@ansible php]# ls
files  handlers  tasks  templates  vars
[root@ansible php]# vim tasks/main.yml

[root@ansible php]# cd ../
[root@ansible roles]# cd nginx
[root@ansible nginx]# cd files
[root@ansible files]# rz -E
```

```
rz waiting to receive.
[root@ansible files]# ls
bbs.tar.gz  nginx-1.16.0.tar.gz
[root@ansible files]# cd ../
[root@ansible nginx]# vim tasks/main.yml
[root@ansible nginx]# vim templates/nginx.conf
[root@ansible nginx]# cd ../php
[root@ansible php]# vim tasks/main.yml
[root@ansible php]# vim handlers/main.yml
- name: start php
  raw: systemctl start php-fpm
[root@ansible php]# cd ../nginx/
[root@ansible nginx]# vim tasks/main.yml

[root@ansible nginx]# cd ../mariadb/
[root@ansible mariadb]# ls
files  handlers  tasks
[root@ansible mariadb]# vim tasks/main.yml
- name: install mariadb
  yum: name=mariadb-server state=present
- name: move config file
  shell: "[ -e /etc/my.cnf ] && mv /etc/my.cnf /etc/my.cnf.bak"
- name: provide a new config file
  copy: src=my.cnf dest=/etc/my.cnf
- name: reload mariadb
  shell: systemctl restart mariadb
- name: create database testdb
  shell: mysql -u root -e "create database bbs;grant all on bbs.* to
'bbs'@'192.168.200.%' identified by '123456' ;flush privileges;"
notify:
  - restart mariadb
```

还原及准备客户端：

```
[root@client1 ~]# killall -9 nginx
```

```
[root@client1 ~]# rm -rf /usr/local/nginx/
[root@client1 ~]# rm -rf /usr/src/nginx-1.16.0*
[root@client1 ~]# systemctl stop mariadb.service
[root@client1 ~]# netstat -lnpt
[root@client1 ~]# cd /etc/yum.repos.d
[root@client1 yum.repos.d]# mv backup/CentOS-Base.repo ./

[root@ansible mariadb]# cd ../../
[root@ansible ansible]# cat nginx.yml php.yml mariadb.yml
- hosts: sofia
  gather_facts: True
  remote_user: root
  roles:
  - nginx
- hosts: sofia
  gather_facts: True
  remote_user: root
  roles:
  - php
- hosts: sofia
  remote_user: root
  roles:
    - mariadb
[root@ansible ansible]# vim lnmp.yml
- hosts: sofia
  gather_facts: True
  remote_user: root
  roles:
  - nginx
  - mariadb
  - php
```

预执行：【web01】

```
[root@ansible ~]# ansible-playbook -C /etc/ansible/lnmp.yml
```

```
[root@ansible ~]# ansible-playbook -C /etc/ansible/lnmp.yml

PLAY [web01] ***********************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************
ok: [192.168.200.108]

TASK [nginx : yum install] *********************************************************************************************************
[DEPRECATION WARNING]: Invoking "yum" only once while using a loop via squash_actions is deprecated. Instead of using a loop to
supply multiple items and specifying `name: "{{ item }}"`, please use `name: ['openssl-devel', 'pcre-devel', 'zlib-devel', 'gcc',
'gcc-c++', 'make']` and remove the loop. This feature will be removed in version 2.11. Deprecation warnings can be disabled by
setting deprecation_warnings=False in ansible.cfg.
changed: [192.168.200.108] => (item=[u'openssl-devel', u'pcre-devel', u'zlib-devel', u'gcc', u'gcc-c++', u'make'])

TASK [nginx : user nginx] **********************************************************************************************************
skipping: [192.168.200.108]

TASK [nginx : package] *************************************************************************************************************
changed: [192.168.200.108]

TASK [nginx : bbs] *****************************************************************************************************************
changed: [192.168.200.108]

TASK [nginx : install nginx] *******************************************************************************************************
skipping: [192.168.200.108]

TASK [nginx : copy conf file] ******************************************************************************************************
changed: [192.168.200.108]

TASK [nginx : install bbs] *********************************************************************************************************
skipping: [192.168.200.108]

TASK [mariadb : install mariadb] ***************************************************************************************************
ok: [192.168.200.108]

TASK [mariadb : move config file] **************************************************************************************************
skipping: [192.168.200.108]

TASK [mariadb : provide a new config file] *****************************************************************************************
ok: [192.168.200.108]

TASK [mariadb : reload mariadb] ****************************************************************************************************
skipping: [192.168.200.108]

TASK [mariadb : create database testdb] ********************************************************************************************
skipping: [192.168.200.108]

TASK [php : yum install] ***********************************************************************************************************
[DEPRECATION WARNING]: Invoking "yum" only once while using a loop via squash_actions is deprecated. Instead of using a loop to
supply multiple items and specifying `name: "{{ item }}"`, please use `name: ['php', 'php-devel', 'php-mysql', 'php-fpm']` and
remove the loop. This feature will be removed in version 2.11. Deprecation warnings can be disabled by setting
deprecation_warnings=False in ansible.cfg.
changed: [192.168.200.108] => (item=[u'php', u'php-devel', u'php-mysql', u'php-fpm'])

RUNNING HANDLER [php : start php] **************************************************************************************************
skipping: [192.168.200.108]

PLAY RECAP *************************************************************************************************************************
192.168.200.108            : ok=8    changed=5    unreachable=0    failed=0    skipped=7    rescued=0    ignored=0
```

预执行：【sofia】

[root@ansible ~]# ansible-playbook -C /etc/ansible/lnmp.yml

```
[root@ansible tasks]# ansible-playbook -C /etc/ansible/lnmp.yml

PLAY [sofia] *******************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************
ok: [192.168.200.108]
ok: [192.168.200.109]

TASK [nginx : yum install] *****************************************************************************************************
[DEPRECATION WARNING]: Invoking "yum" only once while using a loop via squash_actions is deprecated. Instead of using a loop to
supply multiple items and specifying `name: "{{ item }}"`, please use `name: ['openssl-devel', 'pcre-devel', 'zlib-devel', 'gcc',
'gcc-c++', 'make']` and remove the loop. This feature will be removed in version 2.11. Deprecation warnings can be disabled by
setting deprecation_warnings=False in ansible.cfg.
[DEPRECATION WARNING]: Invoking "yum" only once while using a loop via squash_actions is deprecated. Instead of using a loop to
supply multiple items and specifying `name: "{{ item }}"`, please use `name: ['openssl-devel', 'pcre-devel', 'zlib-devel', 'gcc',
'gcc-c++', 'make']` and remove the loop. This feature will be removed in version 2.11. Deprecation warnings can be disabled by
setting deprecation_warnings=False in ansible.cfg.
changed: [192.168.200.108] => (item=[u'openssl-devel', u'pcre-devel', u'zlib-devel', u'gcc', u'gcc-c++', u'make'])
changed: [192.168.200.109] => (item=[u'openssl-devel', u'pcre-devel', u'zlib-devel', u'gcc', u'gcc-c++', u'make'])

TASK [nginx : user nginx] ******************************************************************************************************
skipping: [192.168.200.108]
skipping: [192.168.200.109]

TASK [nginx : package] *********************************************************************************************************
changed: [192.168.200.108]
changed: [192.168.200.109]

TASK [nginx : bbs] *************************************************************************************************************
changed: [192.168.200.108]
changed: [192.168.200.109]

TASK [nginx : install nginx] ***************************************************************************************************
skipping: [192.168.200.108]
skipping: [192.168.200.109]

TASK [nginx : copy conf file] **************************************************************************************************
changed: [192.168.200.109]
changed: [192.168.200.108]

TASK [nginx : install bbs] *****************************************************************************************************
skipping: [192.168.200.108]
skipping: [192.168.200.109]

TASK [mariadb : install mariadb] ***********************************************************************************************
ok: [192.168.200.108]
changed: [192.168.200.109]

TASK [mariadb : move config file] **********************************************************************************************
skipping: [192.168.200.109]
skipping: [192.168.200.108]

TASK [mariadb : provide a new config file] *************************************************************************************
changed: [192.168.200.109]
ok: [192.168.200.108]

TASK [mariadb : reload mariadb] ************************************************************************************************
skipping: [192.168.200.108]
skipping: [192.168.200.109]

TASK [mariadb : create database testdb] ****************************************************************************************
skipping: [192.168.200.109]
skipping: [192.168.200.108]

TASK [php : yum install] *******************************************************************************************************
[DEPRECATION WARNING]: Invoking "yum" only once while using a loop via squash_actions is deprecated. Instead of using a loop to
supply multiple items and specifying `name: "{{ item }}"`, please use `name: ['php', 'php-devel', 'php-mysql', 'php-fpm']` and
remove the loop. This feature will be removed in version 2.11. Deprecation warnings can be disabled by setting
deprecation_warnings=False in ansible.cfg.
[DEPRECATION WARNING]: Invoking "yum" only once while using a loop via squash_actions is deprecated. Instead of using a loop to
supply multiple items and specifying `name: "{{ item }}"`, please use `name: ['php', 'php-devel', 'php-mysql', 'php-fpm']` and
remove the loop. This feature will be removed in version 2.11. Deprecation warnings can be disabled by setting
deprecation_warnings=False in ansible.cfg.
changed: [192.168.200.109] => (item=[u'php', u'php-devel', u'php-mysql', u'php-fpm'])
changed: [192.168.200.108] => (item=[u'php', u'php-devel', u'php-mysql', u'php-fpm'])

RUNNING HANDLER [php : start php] **********************************************************************************************
skipping: [192.168.200.109]
skipping: [192.168.200.108]

PLAY RECAP *********************************************************************************************************************
192.168.200.108            : ok=8    changed=5    unreachable=0    failed=0    skipped=7    rescued=0    ignored=0
192.168.200.109            : ok=8    changed=7    unreachable=0    failed=0    skipped=7    rescued=0    ignored=0

[root@ansible tasks]#
```

查看测试：

```
[root@client1 ~]# netstat -lnpt | grep -E "80|9000|3306"
tcp        0      0 127.0.0.1:9000          0.0.0.0:*               LISTEN
6893/php-fpm: maste
```

```
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN
6832/mysqld
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
6506/nginx: master
[root@client2 ~]# netstat -lnpt | grep -E "80|9000|3306"
tcp        0      0 127.0.0.1:9000          0.0.0.0:*               LISTEN
8321/php-fpm: maste
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN
8260/mysqld
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
7934/nginx: master
```

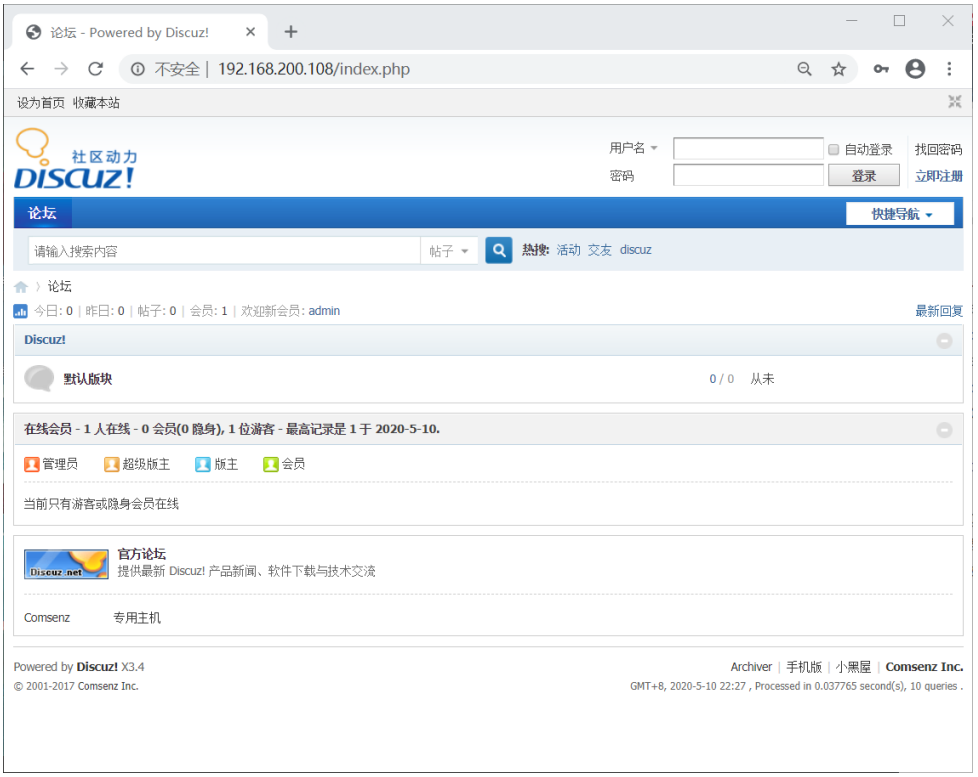
后期排错：

```
[root@ansible ~]# vim /etc/ansible/roles/mariadb/tasks/main.yml
- name: install mariadb
  yum: name=mariadb-server state=present
- name: move config file
  shell: "[ -e /etc/my.cnf ] && mv /etc/my.cnf /etc/my.cnf.bak"
- name: provide a new config file
  copy: src=my.cnf dest=/etc/my.cnf
- name: reload mariadb
  shell: systemctl restart mariadb
- name: create database testdb
  shell: mysql -u root -e "create database bbs;grant all on bbs.* to
'bbs'@'localhost' identified by '123456' ;flush privileges;"
notify:
  - restart mariadb
```

注意：前面授权的时候是mysql -u root -e "create database bbs;grant all on bbs.* to 'bbs'@'192.168.200.%' identified by '123456' ;flush privileges;"

在登录数据库的时候，名字得写192.168.200.108/109

后面改了后就是默认的'localhost'了


论坛 - Powered by Discuz!
不安全 | 192.168.200.108/index.php
设为首页 收藏本站
社区动力
DISCUZ!
用户名
自动登录
找回密码
密码
登录
立即注册
论坛
快捷导航
请输入搜索内容
帖子
热搜: 活动 交友 discuz
论坛
今日: 0 | 昨日: 0 | 帖子: 0 | 会员: 1 | 欢迎新会员: admin
最新回复
Discuz!
默认版块
0 / 0 从未
在线会员 - 1 人在线 - 0 会员(0 隐身), 1 位游客 - 最高记录是 1 于 2020-5-10.
管理员
超级版主
版主
会员
当前只有游客或隐身会员在线
官方论坛
提供最新 Discuz! 产品新闻、软件下载与技术交流
Comsenz
专用主机
Powered by Discuz! X3.4
© 2001-2017 Comsenz Inc.
Archiver | 手机版 | 小黑屋 | Comsenz Inc.
GMT+8, 2020-5-10 22:27 , Processed in 0.037765 second(s), 10 queries .