nsd1902_devops_day03

ansible基础应用

在虚拟环境下安装ansible

```
# cd ansible_pkg/
# pip3 install *

# 或 在线安装
# pip3 install ansible==2.7.2

# yum install -y sshpass
```

准备运行环境

```
# 创建工作目录并创建配置文件
# mkdir myansible
# cd myansible
# vim ansible.cfg
[defaults]
inventory = hosts
remote_user = root

# vim hosts
[dbservers]
node5.tedu.cn

[webservers]
node6.tedu.cn
node7.tedu.cn

# 配置名称解析
[root@room8pc16 nsd2019]# for i in {1..254}
> do
> echo -e "192.168.4.$i\tnode$i.tedu.cn\tnode$i" >> /etc/hosts
> done


# 收集远程主机的密钥并保存。用户登陆时，不再提示是否接受密钥
[root@room8pc16 nsd2019]# ssh-keyscan node{5..7} node{5..7}.tedu.cn 192.168.4.{5..7} >>
~/.ssh/known_hosts


# 测试环境
# ansible all -m ping -k
```

ansible之adhoc（临时命令）

```
# ansible 主机  -m 模块  -a "参数"
```

ansible之playbook

- 配置远程主机的密钥

```
# 查模块帮助
# ansible-doc authorized_key

# vim auth_key.yml
---
- name: configure ssh key
  hosts: all
  tasks:
    - name: upload key
      authorized_key:
        user: root
        state: present
        key: "{{ lookup('file', '/root/.ssh/id_rsa.pub') }}"
# 检查语法
# ansible-playbook --syntax-check auth_key.yml

# 执行playbook
# ansible-playbook auth_key.yml -k
```

- 配置yum

```
# mkdir files
# vim files/servers.repo
# 编写playbook
# vim  mkrepo.yml
---
- name: configure yum
  hosts: all
  tasks:
    - name: upload repo file
      copy:
        src: files/servers.repo
        dest: /etc/yum.repos.d/server.repo

# ansible-playbook mkrepo.yml
```

配置lamp分离结构

```
# vim lamp.yml
---
- name: configure webservers
  hosts: webservers
  tasks:
    - name: install web pkgs
      yum:
        name: [httpd, php, php-mysql]
        state: present
    - name: configure web service
      service:
        name: httpd
        state: started
        enabled: yes

- name: configure dbservers
  hosts: dbservers
  tasks:
    - name: install db pkgs
      yum:
        name: mariadb-server
        state: present
    - name: configure db service
      service:
        name: mariadb
        state: started
        enabled: yes
# ansible-playbook lamp.yml
```

命名元组

命名元组是对元组的一个扩展，它也支持原始的元组的操作，同时它给每个元素起名。访问元组的元素时，既可以通过下标访问，也可以通过名称访问。

```
>>> from collections import namedtuple
>>> Point = namedtuple('Point', ['x', 'y', 'z'])
>>> p1 = Point(10, 20, 30)
>>> p1
Point(x=10, y=20, z=30)
>>> p1.x
10
>>> p1.y
20
>>> p1.z
30
>>> p1[0]
10
>>> p1[:2]
(10, 20)
```

ansible编程

ansible官方手册：https://docs.ansible.com

找到ansible2.7的手册页：https://docs.ansible.com/ansible/2.7/index.html，搜索python api，找到：https://docs.ansible.com/ansible/2.7/dev_guide/developing_api.html?highlight=python%20api

yaml对应成python的数据类型

```
# vim  mkrepo.yml
---
- name: configure yum
  hosts: all
  tasks:
    - name: upload repo file
      copy:
        src: files/servers.repo
        dest: /etc/yum.repos.d/server.repo
# 对应的数据类型如下：
[
    {
        'name': 'configure yum',
        'hosts': 'all',
        'tasts': [
            {
                'name': 'upload repo file',
                'copy': {
                    'src': 'files/servers.repo',
                    'dest': '/etc/yum.repos.d/server.repo'
                },
            },
            {},
        ]
    },
    {},
]
```

ansible-cmdb

可以将服务器的信息以web形式展现

```
# 在虚拟环境中安装ansible-cmdb
# pip3 install ansible-cmdb_pkgs/*
# 或在线安装
# pip3 install ansible-cmdb

# 获取远程主机的信息
# ansible all -m setup --tree /tmp/servers
# ls /tmp/servers

# ansible-cmdb分析获取的信息文件，生成html文件
# ansible-cmdb /tmp/servers > /tmp/servers.html
# firefox /tmp/servers.html
```

开发ansible模块

指定自己编写模块的文件路径是/opt/myansible_lib/

```
# export ANSIBLE_LIBRARY=/opt/myansible_lib/
# mkdir /opt/myansible_lib/
```

编写模块文件rcopy.py，实现远程主机在自己的系统内执行拷贝操作

```
# vim /opt/myansible_lib/rcopy.py
#!/usr/bin/env python

from ansible.module_utils.basic import AnsibleModule
import shutil

def main():
    module = AnsibleModule(
        argument_spec=dict(
            yuan=dict(required=True, type='str'),
            mubiao=dict(required=True, type='str')
        )
    )
    shutil.copy(module.params['yuan'], module.params['mubiao'])
    module.exit_json(changed=True)

if __name__ == '__main__':
    main()
```

调用自己写的模块，将服务器上/etc/hosts拷贝到/tmp/zhuji

```
# ansible webservers -m rcopy -a "yuan=/etc/hosts mubiao=/tmp/zhuji"
```

练习：编写模块

- 模块名为download
- 接受两个参数
    - url：指定一个网络路径
    - path：指定本地路径
- 执行指令ansible all -m download -a "url=http://xxxx path=/path/to/local/file"，可以将网上的资源下载到本地