nsd1912-devops-day03

ansible应用

- ansible是纯python编写的
- 使用yum进行安装，默认安装的是基于python2的
- 使用pip3安装基于python3的ansible

```
# 查看ansible所有可用版本
[root@localhost day02]# pip3 install ansible==
# 安装2.7.17版本
[root@localhost day02]# pip3 install ansible==2.7.17
```

- 配置ansible工作环境

```
# 创建工作目录
[root@localhost day03]# mkdir myansible
[root@localhost day03]# cd myansible/

# 创建配置文件
[root@localhost myansible]# ansible --version
[root@localhost myansible]# vim ansible.cfg
[defaults]
inventory = hosts
remote_user = root

# 创建主机清单文件
[root@localhost myansible]# vim hosts
[dbservers]
db1.tedu.cn

[webservers]
web1.tedu.cn

# 配置名称解析
[root@localhost myansible]# vim /etc/hosts
192.168.81.133  db1.tedu.cn
192.168.81.134  web1.tedu.cn

# 收集远程主机密钥并保存
[root@localhost myansible]# ssh-keyscan {db1,web1}.tedu.cn >> ~/.ssh/known_hosts

# 测试到远程主机的连接
[root@localhost myansible]# ansible all -m ping -k
# 如果报错，不能使用密码连接，则需要安装 sshpass
[root@localhost myansible]# yum install -y sshpass
# 再次测试
[root@localhost myansible]# ansible all -m ping -k
```

> sshpass使用
>
> ```
> # 在远程主机执行命令，需要输入密码
> [root@localhost myansible]# ssh db1.tedu.cn 'id root'
> root@db1.tedu.cn's password:
> uid=0(root) gid=0(root) 组=0(root)
>
> # 可以通过sshpass为ssh命令传递所需的密码
> [root@localhost myansible]# sshpass -predhat ssh db1.tedu.cn 'id root'
> uid=0(root) gid=0(root) 组=0(root)
> ```

- 使用模块

```
# 查看所有模块
[root@localhost myansible]# ansible-doc -l
[root@localhost myansible]# ansible-doc -l | grep auth
[root@localhost myansible]# ansible-doc authorized_key
```

ansible应用方式

adhoc模式

- 临时命令模式
- 执行的指令很简单，或不经常使用的配置都可以采用adhoc模式

```
[root@localhost myansible]# ansible all -m ping
```

playbook模式

- 修改vim配置

```
[root@localhost myansible]# vim ~/.vimrc
filetype plugin on
syntax on
" set cursorline
set scrolloff=7
set ai
set ts=4
set et
set encoding=utf8
autocmd FileType yaml setlocal sw=2 ts=2 et ai
nnoremap <C-a> <Home>
nnoremap <C-e> <End>
nnoremap <F2> :set nu! nu?<CR>
nnoremap ; :
```

- 实现ssh免密登陆

```
# 创建密钥对
[root@localhost myansible]# ssh-keygen
# 编写playbook
[root@localhost myansible]# vim auth.yml
---
- name: configure ssh key
  hosts: all
  tasks:
    - name: upload root key
      authorized_key:
        user: root
        state: present
        key: "{{ lookup('file', '/root/.ssh/id_rsa.pub') }}"

# 检查语法
[root@localhost myansible]# ansible-playbook auth.yml --syntax-check

# 执行playbook
[root@localhost myansible]# ansible-playbook auth.yml -k
```

- 编写playbook，实现在webservers组中安装httpd / php / php-mysql并启动httpd服务；实现在dbservers组中安装mariadb-server，并启动mariadb服务

```
[root@localhost myansible]# vim lamp.yml
---
- name: configure web service
  hosts: webservers
  tasks:
    - name: install web pkgs
      yum:
        name: httpd, php, php-mysql
        state: present

    - name: start web service
      service:
        name: httpd
        state: started
        enabled: yes

- name: configure db service
  hosts: dbservers
  tasks:
    - name: install db pkgs
      yum:
        name: mariadb-server
        state: latest

    - name: start db service
      service:
        name: mariadb
        state: started
        enabled: yes
[root@localhost myansible]# ansible-playbook lamp.yml --syntax-check
[root@localhost myansible]# ansible-playbook lamp.yml
```

ansible编程

https://docs.ansible.com/ -> Ansible Documentation -> 修改查看的版本，然后搜索 python api -> https://docs.ansible.com/ansible/2.7/dev_guide/developing_api.html?highlight=python%20api -> 将example中的代码复制出来，并执行。

命名的元组

- 仍然是元组。元组相关的操作同样适用
- 为元组的下标命名，可以通过下标的名字取出相应的数据

```
>>> from functools import namedtuple
>>> Point = namedtuple('Point', ['x', 'y', 'z'])
>>> p1 = Point(10, 20, 35)
>>> type(p1)
<class '__main__.Point'>
>>> p1[0]
10
>>> p1[1:]
(20, 35)
>>> len(p1)
3
>>> p1.x
10
>>> p1.y
20
>>> p1.z
35
```

手工将yaml转为python的类型

```yaml
---
- name: configure web service
  hosts: webservers
  tasks:
    - name: install web pkgs
      yum:
        name: httpd, php, php-mysql
        state: present

    - name: start web service
      service:
        name: httpd
        state: started
        enabled: yes

- name: configure db service
  hosts: dbservers
  tasks:
    - name: install db pkgs
      yum:
        name: mariadb-server
        state: latest

    - name: start db service
      service:
        name: mariadb
        state: started
        enabled: yes
```

转为python数据类型：

```
[
    {
        'name': 'configure web service',
        'hosts': 'webservers',
        'tasks': [
            {
                'name': 'install web pkgs'
                'yum': {
                    'name': ['httpd', 'php', 'php-mysql']
                    'state': 'present'
                }
            },
            {
                'name': 'start web service',
                'service': {
                    'name': 'httpd',
                    'state': 'started',
                    'enabled': 'yes'
                }
            }
        ]
    },
    {
        'name': 'configure db service',
        'hosts': 'dbservers'
        'tasks': [
            {
                'name': 'install db pkgs'
                'yum': {
                    'name': 'mariadb-server',
                    'state': 'latest'
                }
            },
            {
                'name': 'start db service',
                'service': {
                    'name': 'mariadb',
                    'state': 'started',
                    'enabled': 'yes'
                }
            }
        ]
    }
]
```

- ansible vault：实现文件加解密

```
[root@localhost myansible]# cp /etc/hosts /tmp/
# 加密
[root@localhost myansible]# ansible-vault encrypt /tmp/hosts
New Vault password: redhat
Confirm New Vault password: redhat
Encryption successful
[root@localhost myansible]# cat /tmp/hosts

# 解密
[root@localhost myansible]# ansible-vault decrypt /tmp/hosts
Vault password:
[root@localhost myansible]# cat /tmp/hosts
```

ansible-cmdb插件

- 用于将收集到的远程主机信息以web页面的方式展示

```
# 收集信息
[root@localhost myansible]# ansible all -m setup --tree /tmp/out
[root@localhost myansible]# ls /tmp/out
db1.tedu.cn  web1.tedu.cn

# 安装ansible-cmdb插件
[root@localhost myansible]# pip3 install ansible-cmdb
[root@localhost myansible]# ansible-cmdb /tmp/out/ > /tmp/hosts.html
# 报错 ImportError: No module named mako
# 采用以下方式解决
[root@localhost myansible]# which ansible-cmdb
/usr/local/bin/ansible-cmdb
[root@localhost myansible]# vim /usr/local/bin/ansible-cmdb
修改第8行为以下内容
PY_BIN=$(which python3)
[root@localhost myansible]# ansible-cmdb /tmp/out/ > /tmp/hosts.html
[root@localhost myansible]# firefox /tmp/hosts.html
```

ansible模块

- 创建自定义模块路径

```
[root@localhost myansible]# mkdir /opt/mylibs
[root@localhost myansible]# export ANSIBLE_LIBRARY=/opt/mylibs
```

- 编写模块

```
[root@localhost myansible]# vim /opt/mylibs/rcopy.py
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()
```

- 执行命令

```
[root@localhost myansible]# ansible db1.tedu.cn -m rcopy -a "yuan=/etc/passwd
mubiao=/tmp/mima"
```

> 创建download模块，通过命令下载文件
>
> ```
> [root@localhost myansible]# ansible db1.tedu.cn -m download -a "url=http://xxxxx
> dest=/tmp/xxxx"
> [root@localhost myansible]# vim /opt/mylibs/download.py
> from ansible.module_utils.basic import AnsibleModule
> import wget
>
> def main():
>     module = AnsibleModule(
>         argument_spec=dict(
>             url=dict(required=True, type='str'),
>             dest=dict(required=True, type='str'),
>         )
>     )
>     wget.download(module.params['url'], module.params['dest'])
>     module.exit_json(changed=True)
>
> if __name__ == '__main__':
>     main()
>
> # 注意，需要在被管理端安装 wget。并且是为python2安装
> [root@localhost ~]# wget
> https://files.pythonhosted.org/packages/47/6a/62e288da7bcda82b935ff0c6cfe542970f04e2
> 9c756b0e147251b2fb251f/wget-3.2.zip
> [root@localhost ~]# unzip wget-3.2.zip
> [root@localhost ~]# cd wget-3.2/
> [root@localhost wget-3.2]# python setup.py install
> ```