

nsd1903_devops_day03

ansible应用

安装

```
(nsd1903) [root@room8pc16 day03]# pip install zzg_pypkgs/ansible_pkg/*
```

配置基础应用环境

```
(nsd1903) [root@room8pc16 day03]# mkdir myansible
(ns1903) [root@room8pc16 day03]# cd myansible
(ns1903) [root@room8pc16 myansible]# vim ansible.cfg
[defaults]
inventory = hosts
remote_user = root
(ns1903) [root@room8pc16 myansible]# vim hosts
[dbservers]
node5.tedu.cn

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

# 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

# 收集主集密钥，在首次ssh远程主机时，就不需要回答yes了
[root@room8pc16 nsd2019]# ssh-keyscan 192.168.4.{5..7} node{5..7} node{5..7}.tedu.cn >>
~/.ssh/known_hosts

# 免密登陆
[root@room8pc16 nsd2019]# for i in {5..7}
> do
> ssh-copy-id node${i}
> done
```

应用之adhoc

- adhoc是临时命令，命令比较简单又不是周期性使用时，采用此方式

```
(nsd1903) [root@room8pc16 myansible]# ansible all -m ping
(nsd1903) [root@room8pc16 myansible]# ansible-doc -l | grep copy
(nsd1903) [root@room8pc16 myansible]# ansible-doc copy
(nsd1903) [root@room8pc16 myansible]# ansible all -m copy -a "src=/etc/hosts
dest=/etc/hosts"
```

应用之playbook

```
# 修改vim编辑器，可以支持yaml输入格式
# vim ~/.vimrc
autocmd FileType yaml setlocal sw=2 ts=2 et ai

# 上传yum仓库文件
# 1. 在ansible管理端创建yum文件
(nsd1903) [root@room8pc16 myansible]# mkdir files
(nsd1903) [root@room8pc16 myansible]# vim files/server.repo
[server]
name=server
baseurl=ftp://192.168.4.254/centos7.4
gpgcheck=0
# 2. 编写playbook
(nsd1903) [root@room8pc16 myansible]# vim yumrepo.yml
---
- name: configure yum
  hosts: all
  tasks:
    - name: upload yum repo file
      copy:
        src: files/server.repo
        dest: /etc/yum.repos.d/
# 3. 检查语法
[root@room8pc16 myansible]# ansible-playbook --syntax-check yumrepo.yml

# 4. 执行playbook
[root@room8pc16 myansible]# ansible-playbook yumrepo.yml
```

练习：playbook

- web服务器上安装httpd / php / php-mysql
- web服务器启用httpd服务
- db服务器安装mariadb-server
- db服务器启动mariadb服务

```
[root@room8pc16 myansible]# vim lamp.yml
---
- name: configure web servers
  hosts: web servers
  tasks:
    - name: install web pkgs
      yum:
        name: [httpd, php, php-mysql]
```

```

    state: present
- name: enable web service
  service:
    name: httpd
    state: started
    enabled: yes

- name: configure db servers
  hosts: db servers
  tasks:
    - name: install db pkgs
      yum:
        name: mariadb-server
        state: latest
    - name: enable db service
      service:
        name: mariadb
        state: started

```

ansible编程

ansible官方文档: <https://docs.ansible.com/ansible/2.7/index.html> -> 搜索 python api。将example中的代码复制，执行。

命名的元组

- 本质上还是元组
- 只是给元组的下标添加了名称

```

>>> from collections import namedtuple
# 创建名为Point的命名元组，它接受3个参数
>>> Point = namedtuple('Point', ['x', 'y', 'z'])
>>> p1 = Point(10, 15, 28)
>>> p1[0]
10
>>> p1[1:]
(15, 28)
>>> p1.x
10
>>> p1.y
15
>>> p1.z
28

```

将yaml文件手工转成python数据类型

- 以"- "开始的行，转成列表项
- 以"key: val"作为结构的行，转为字典

```

[
  {

```

```

        'name': 'configure webservers',
        'hosts': 'webservers',
        'tasks': [
            {
                'name': 'install web pkgs',
                'yum': {
                    'name': ['httpd', 'php', 'php-mysql'],
                    'state': 'present'
                }
            },
            {
                'name': 'enable web service',
                'service': {
                    'name': 'httpd',
                    'state': 'started',
                    'enabled': 'yes'
                }
            }
        ]
    },
    {
        'name': 'configure dbservers',
        'hosts': 'dbservers',
        'tasks': [
            {},
            {}
        ]
    }
]

```

ansible模块开发

- 自定义的模块可以存放到一个目录后，设置ANSIBLE_LIBRARY环境变量指向它

```

(nsd1903) [root@room8pc16 day03]# mkdir /tmp/mylibs
(nsd1903) [root@room8pc16 day03]# export ANSIBLE_LIBRARY=/tmp/mylibs

```

编写模块，用于在远程主机上实现拷贝操作

```

# rcopy.py
"用于在远程主机上进行拷贝操作"
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()

# 调用模块，执行拷贝
(nsd1903) [root@room8pc16 myansible]# ansible dbservers -m rcopy -a "yuan=/etc/passwd
mubiao=/tmp/mima"

# 注意，ansible在执行命令时，将会把模块进行配置拷贝到远程主机上执行。远程主机如果没有python3，则不支持中文

```

模块练习：

- 编写模块download用于下载
- 有两个参数
 - url: 定义网络源
 - dest: 用于定义本机目录

```
# ansible all -m download -a "url=http://xxxx dest=/path/to/file"
```

1. 编写模块

```

# /tmp/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()

```

2. 在远程主机安装wget模块

```
# 在本地先将wget下载
[root@room8pc16 tmp]# pip download wget --trusted-host pypi.douban.com
# 拷贝下载的文件到远程主机
[root@room8pc16 tmp]# scp wget-3.2.zip node5:/tmp
# 在远程主机安装wget
[root@room8pc16 tmp]# ssh node5
[root@node5 ~]# cd /tmp/
[root@node5 tmp]# unzip wget-3.2.zip
[root@node5 tmp]# cd wget-3.2/
[root@node5 wget-3.2]# python setup.py install # python包都可以如此安装
```

3. 执行下载操作

```
(nsd1903) [root@room8pc16 myansible]# ansible dbservers -m download -a  
"url=http://192.168.4.254/server.repo dest=/tmp/"
```

ansible-cmdb插件

该插件可以将收集下来的主机信息显示为web页面。

```
# 收集主机信息，存到/tmp/out目录
(nsd1903) [root@room8pc16 myansible]# ansible all -m setup --tree /tmp/out/
# 安装ansible-cmdb
(nsd1903) [root@room8pc16 myansible]# pip install ansible-cmdb_pkgs/*
# 生成web页面
(nsd1903) [root@room8pc16 myansible]# ansible-cmdb /tmp/out/ > /tmp/hosts.html
# 查看结果
(nsd1903) [root@room8pc16 myansible]# firefox /tmp/hosts.html &
```