

nsd1907_devops_day03

ansible

```
# 本地安装
(nsd1907) [root@room8pc16 day03]# pip install /var/ftp/pub/zzg_pypkgs/ansible_pkg/*
# 在线安装
(nsd1907) [root@room8pc16 day03]# pip install ansible==2.7.2
(nsd1907) [root@room8pc16 day03]# ansible --version
ansible 2.7.2
```

ansible应用

- adhoc临时命令
- playbook

```
# 创建工作目录
(nsd1907) [root@room8pc16 day03]# mkdir myansible
(nsd1907) [root@room8pc16 day03]# cd myansible/
# 编写配置文件
(nsd1907) [root@room8pc16 myansible]# vim ansible.cfg
[defaults]
inventory = hosts
remote_user = bob    # ssh到目标主机的用户

[privilege_escalation]    # 提权设置
become = true
become_user = root
become_method = sudo
become_ask_pass = flase

(nsd1907) [root@room8pc16 myansible]# vim ansible.cfg
[defaults]
inventory = hosts
remote_user = root

#[privilege_escalation]
#become = true
#become_user = root
#become_method = sudo
#become_ask_pass = flase

(nsd1907) [root@room8pc16 myansible]# vim hosts
[dbservers]
node4
```

```
[webservers]
node5
node6

# 配置名称解析
[root@room8pc16 day02]# for i in {1..254}
> do
> echo -e "192.168.4.${i}\tnode${i}.tedu.cn\tnode${i}" >> /etc/hosts
> done

# 收集密钥，加入到信任列表
[root@room8pc16 day02]# ssh-keyscan node{4..6} >> ~/.ssh/known_hosts
# 测试连通性
(nsd1907) [root@room8pc16 myansible]# ansible all -m ping -k
# 配置免密登陆
(nsd1907) [root@room8pc16 day02]# ansible-doc authorized_key
(nsd1907) [root@room8pc16 day02]# ansible all -m authorized_key -a "user=root state=present
key='{lookup(\'file\', \'/root/.ssh/id_rsa.pub\') }{' -k
```

配置vim支持yaml语法

```
(nsd1907) [root@room8pc16 myansible]# vim ~/.vimrc
autocmd FileType yaml setlocal sw=2 ts=2 et ai
```

通过playbook配置yum

```
(nsd1907) [root@room8pc16 myansible]# ansible-doc yum_repository
(nsd1907) [root@room8pc16 myansible]# vim yum_repos.yml
---
- name: configure yum repository
  hosts: all
  tasks:
    - name: config yum
      yum_repository:
        file: centos7
        baseurl: ftp://192.168.4.254/centos7.4
        name: centos7
        description: local centos7 repository
        gpgcheck: no
        enabled: yes
(nsd1907) [root@room8pc16 myansible]# ansible-playbook --syntax-check yum_repos.yml
(nsd1907) [root@room8pc16 myansible]# ansible-playbook yum_repos.yml
```

通过playbook完成lamp环境

```
(nsd1907) [root@room8pc16 myansible]# vim lamp.yml
---
- name: configure dbservers
  hosts: dbservers
  tasks:
    - name: install db pkgs
```

```

    yum:
      name: mariadb-server
      state: present
  - name: config db service
    service:
      name: mariadb
      state: started
      enabled: yes

- name: configure web servers
  hosts: webservers
  tasks:
    - name: install web pkgs
      yum:
        name: httpd, php, php-mysql
        state: present
    - name: config web service
      service:
        name: httpd
        state: started
        enabled: yes

(nsd1907) [root@room8pc16 myansible]# ansible-playbook --syntax-check lamp.yml
(nsd1907) [root@room8pc16 myansible]# ansible-playbook lamp.yml

```

ansible编程

<https://docs.ansible.com/> -> Ansible Documentation -> 选2.7版本 -> 搜索python api -> https://docs.ansible.com/ansible/2.7/dev_guide/developing_api.html?highlight=python%20api

命名元组

- 本质上还是元组
- 扩展了元组的属性，为元组下标添加名称

```

>>> from collections import namedtuple
>>> Point = namedtuple('Point', ('x', 'y', 'z'))
>>> p1 = Point(10, 15, 23)
>>> p1[0]
10
>>> p1[1:]
(15, 23)
>>> p1.x
10
>>> p1.y
15
>>> p1.z
23

```

将playbook转换成python数据类型

```

# lamp.yml
[
  {
    'name': 'configure dbservers',
    'hosts': 'dbservers',
    'tasks': [
      {
        'name': 'install db pkgs',
        'yum': {
          'name': 'mariadb-server',
          'state': 'present'
        }
      },
      {
        'name': 'config db service',
        'service': {
          'name': 'mariadb',
          'state': 'started',
          'enabled': 'yes'
        }
      }
    ]
  },
  {
    'name': 'configure webservers',
    'hosts': 'webservers',
    'tasks': [
      {
        'name': 'install web pkgs',
        'yum': {
          'name': [httpd, php, php-mysql],
          'state': present
        }
      },
      {
        'name': 'config web service',
        'service': {
          'name': 'httpd',
          'state': 'started',
          'enabled': 'yes'
        }
      }
    ]
  }
]

```

ansible加密

```
(nsd1907) [root@room8pc16 myansible]# echo 'hello world' > /tmp/hello.txt
(nsd1907) [root@room8pc16 myansible]# ansible-vault encrypt /tmp/hello.txt # 加密
(nsd1907) [root@room8pc16 myansible]# cat /tmp/hello.txt
(nsd1907) [root@room8pc16 myansible]# ansible-vault decrypt /tmp/hello.txt # 解密
(nsd1907) [root@room8pc16 myansible]# cat /tmp/hello.txt
```

ansible-cmdb

```
(nsd1907) [root@room8pc16 day03]# pip install /var/ftp/pub/zzg_pypkgs/ansible-cmdb_pkgs/*
# 收集远程主机的信息，保存到一个目录中
(nsd1907) [root@room8pc16 myansible]# ansible all -m setup --tree /tmp/nsd1907out/
# 将收集下来的信息通过ansible-cmdb转成web页面
(nsd1907) [root@room8pc16 myansible]# ansible-cmdb /tmp/nsd1907out/ > /tmp/myhosts.html
(nsd1907) [root@room8pc16 myansible]# firefox /tmp/myhosts.html
```

ansible模块

自定义模块

```
# 创建环境变量，指定自定义ansible模块的位置
(nsd1907) [root@room8pc16 myansible]# mkdir /tmp/mylibs
(nsd1907) [root@room8pc16 myansible]# export ANSIBLE_LIBRARY=/tmp/mylibs

# 在模块目录下创建模块文件
(nsd1907) [root@room8pc16 myansible]# cp ../rcopy.py /tmp/mylibs/
(nsd1907) [root@room8pc16 myansible]# cat /tmp/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()

# 执行自定义的模块
(nsd1907) [root@room8pc16 myansible]# ansible webservers -m rcopy -a "yuan=/etc/passwd
mubiao=/tmp/mima"

# download模块
(nsd1907) [root@room8pc16 myansible]# cat /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()
```

执行模块报错。因为远端主机没有wget模块

```
(nsd1907) [root@room8pc16 myansible]# ansible webservers -m download -a
"url=http://192.168.4.254/zabbix.png dest=/tmp/zabbix.png"
```

不使用pip安装wget模块(安装所有的python模块都可以采用这样的方法)

1. 访问<https://pypi.org/>，搜索wget并下载

2. 在目标主机上安装wget

```
[root@node5 ~]# unzip wget-3.2.zip
```

```
[root@node5 ~]# cd wget-3.2/
```

```
[root@node5 wget-3.2]# python setup.py install
```

再次执行下载任务，成功

```
(nsd1907) [root@room8pc16 myansible]# ansible webservers -m download -a
"url=http://192.168.4.254/zabbix.png dest=/tmp/zabbix.png"
```