

nsd1906_devops_day03

嵌套的数据类型

- 只要记住，容器类型可以包含其他类型的数据
- 字典通过key取出值，列表和元组能通过下标取值。取出的值如果还是字典或列表元组，可以继续通过key和下标取值。

```
>>> users = {'bob': {'email': 'bob@qq.com', 'qq': '123423454'}, 'alice': {'email': 'alice@tedu.cn', 'qq': '98792342'}}
# 取出bob的信息
>>> users['bob']
{'email': 'bob@qq.com', 'qq': '123423454'}
# 取出bob的email
{'email': 'bob@qq.com', 'qq': '123423454'}
>>> users['bob']['email']
'bob@qq.com'

>>> users2 = {'bob': ['bob@qq.com', '2344525'], 'alice': ['alice@tedu.cn', '4523426364']}
>>> users2['alice']
['alice@tedu.cn', '4523426364']
>>> users2['alice'][-1]
'4523426364'
```

ansible

```
# 本地安装方式
(nsd1906) [root@room8pc16 day03]# pip install /var/ftp/pub/zzg_pypkgs/ansible_pkg/*
# 在线安装方式
(nsd1906) [root@room8pc16 day03]# pip install ansible==2.7.2

# 初始化ansible环境
[root@room8pc16 day03]# mkdir myansible
[root@room8pc16 day03]# cd myansible
[root@room8pc16 myansible]# vim ansible.cfg
[defaults]
inventory = hosts
remote_user = root

[root@room8pc16 myansible]# vim hosts
[dbservers]
node4

[webservers]
node5
node6
```

```
# 创建三台虚拟机，配置好IP地址，可用于管理。实现免密登陆
# 实现名称解析
[root@room8pc16 day01]# for i in {1..254}
> do
> echo -e "192.168.4.$i\tnode$i.tedu.cn\tnode$i" >> /etc/hosts
> done

# 测试
[root@room8pc16 myansible]# ansible all -m ping
```

ansible应用

管理远程主机的两种方式：

- adhoc临时命令
- playbook

```
# adhoc方式配置yum
[root@room8pc16 myansible]# ansible all -m yum_repository -a "name=Server
baseurl=ftp://192.168.4.254/centos7.4 enabled=yes gpgcheck=no description='Centos 7.4'"

# 修改vim配置，适应yaml语法
[root@room8pc16 myansible]# vim ~/.vimrc
autocmd FileType yaml setlocal sw=2 ts=2 et ai

# 通过playbook创建lamp环境
[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: configure 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: present
    - name: configure db service
      service:
        name: mariadb
        state: started
```

```
enabled: yes
# 检查语法
[root@room8pc16 myansible]# ansible-playbook --syntax-check lamp.yml
# 执行playbook
[root@room8pc16 myansible]# ansible-playbook lamp.yml
```

ansible编程

ansible官方网站：<https://docs.ansible.com/ansible/2.7/index.html> -> 搜索 python api -> 将示例代码拷贝出来，执行。

命名的元组

- 本质上还是元组
- 元组通过下标取值
- 命名的元组只是给下标起名。可以通过名字进行取值

```
>>> from collections import namedtuple
>>> Point = namedtuple('Point', ['x', 'y', 'z'])
>>> p1 = Point(10, 20, 15)
>>> p1[1:]
(20, 15)
>>> p1[-1]
15
>>> p1.x
10
>>> p1.y
20
>>> p1.z
15
```

将yaml文件转成python数据类型

- 以-开头的是列表
- 以:分隔的是字典

```
---
- name: configure web servers
  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

```

将以上playbook改为python数据类型

一个playbook由两个play构成，每个play都是一个列表项

每个play又是一个字典结构

```

[
  {
    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
    }
}
],
},
]

```

加密

```

# 加密文件
[root@room8pc16 myansible]# echo 'hello world' > hi.txt
[root@room8pc16 myansible]# cat hi.txt
hello world
[root@room8pc16 myansible]# ansible-vault encrypt hi.txt

# 解密
[root@room8pc16 myansible]# ansible-vault decrypt hi.txt

```

自定义模块

```

# 创建ansible模块目录
(nsd1906) [root@room8pc16 day03]# mkdir /tmp/mylib
(nsd1906) [root@room8pc16 day03]# export ANSIBLE_LIBRARY=/tmp/mylib
(nsd1906) [root@room8pc16 day03]# vim /tmp/mylib/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
(nsd1906) [root@room8pc16 myansible]# ansible dbservers -m rcopy -a "yuan=/etc/hosts
mubiao=/tmp/zhuji"

```

安装python软件包

```
# 在http://pypi.org查找并下载wget
# 拷贝wget到目标主机
[root@node4 ~]# unzip wget-3.2.zip
[root@node4 ~]# cd wget-3.2/
[root@node4 wget-3.2]# python setup.py install
```