

信息采集

硬件采集这部分内容不是浏览器端做的事情了，但是以后可以用它实现安排编排Job，后台启动执行安排好的作业。

CMDB的资源数据来源

1. 人工录入
 - 做好CMDB系统，提供UI界面，手工录入数据提交入库
2. 表格采集导入
 - 制定各种资源的excel表格，发给各个部门填写，收集上来后，采用专用程序提取excel中合格数据入库
3. 程序采集

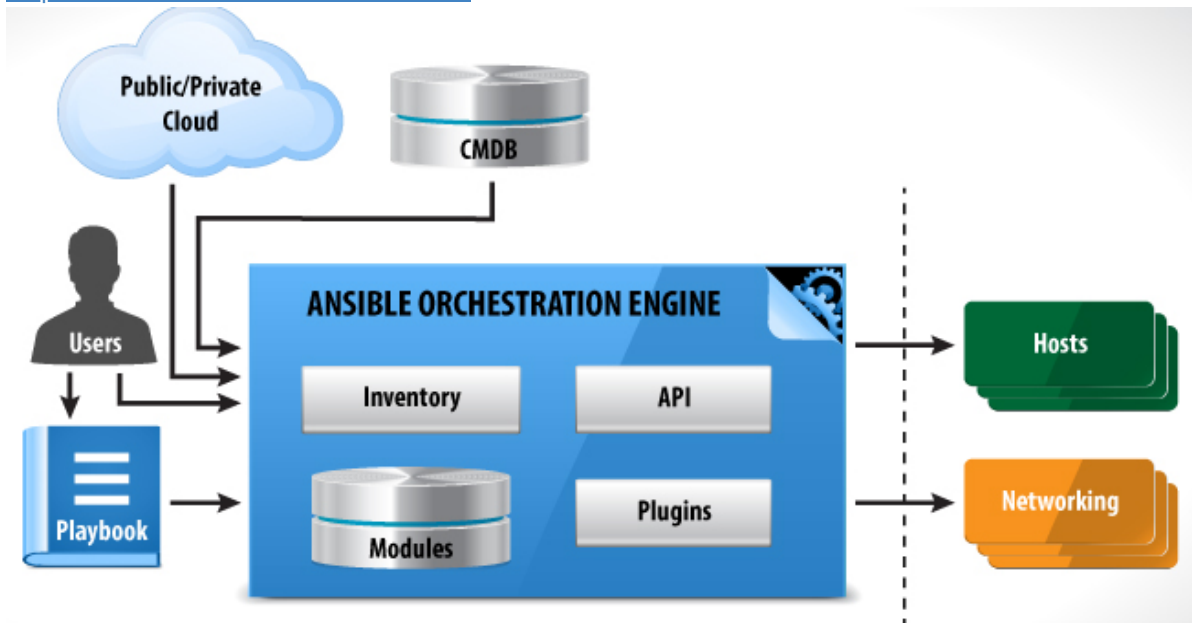
如何采集一个服务器的相关信息？可以人工去查看服务器所有相关信息，然后填写表单提交。也可以写一个程序自动在服务器上搜索所有信息。

程序采集方式

- 无Agent：不需要在目标安装特别制作的软件来收集信息
 - SSH：通过SSH命令在远程主机运行命令或脚本返回执行结果
- ```
1 | $ ssh python@192.168.142.130 pwd
```
- Ansible
  - 有Agent：必须在目标运行Agent进程来收集信息
    - SNMP (Simple Network Management Protocol) 简单网络管理协议
      - 在目标设备上安装net-snmp，运行SNMP Agent的服务器snmpd
      - NMS (Network Management System)，负责访问、收集、设置设备数据，并管理、分析这些数据的系统

## Ansible

---



Ansible 是Python开发的，主要使用paramiko和pyYaml库。

Ansible 2.12开始 需要 Python3.8+。可以在Python虚拟环境中使用Ansible。

```
1 | $ pip install -v ansible
```

## Inventory文件

将主机Host分组Group。

/home/python/ansible\_hosts, 内容如下

```
1 | [remotes]
2 | 192.168.142.130
```

段remote表示分组。

## 配置文件

先来配置下配置项~/ansible.cfg。ansible执行的时候会按照以下顺序查找配置项:

- ANSIBLE\_CONFIG (环境变量)
- ansible.cfg (当前目录下)
- ~/.ansible.cfg (用户家目录下)
- /etc/ansible/ansible.cfg

创建~/ansible.cfg, 内容如下

```
1 | [defaults]
2 | inventory = /home/python/ansible_hosts
```

## 运行

```
1 $ ansible all -m ping
```

all 所有分组。

-m 指定module模块

-a 指定args模块参数

```
1 192.168.142.130 | UNREACHABLE! => {
2 "changed": false,
3 "msg": "Failed to connect to the host via ssh: Warning: Permanently added
 '192.168.142.130' (ECDSA) to the list of known
 hosts.\r\npython@192.168.142.130: Permission denied (publickey,gssapi-
 keyex,gssapi-with-mic,password).",
4 "unreachable": true
5 }
```

很明显，SSH权限问题，配置免密登录。

python@192.168.142.130 说明默认使用python用户，有些信息需要使用root用户才能获取，我们指定root用户。

修改/home/python/ansible\_hosts，内容如下

```
1 [remotes]
2 192.168.142.130 ansible_ssh_user=root
```

```
1 $ ssh-keygen -t rsa
2 $ ssh-copy-id -i ~/.ssh/id_rsa.pub root@192.168.142.130
3 Now try logging into the machine, with: "ssh 'root@192.168.142.130'"
4 and check to make sure that only the key(s) you wanted were added.
```

出现了上面的提示，也就是当前用户可以免密登录192.168.142.130且使用root身份。

```
1 $ ansible remotes -m ping
2 192.168.142.130 | SUCCESS => {
3 "ansible_facts": {
4 "discovered_interpreter_python": "/usr/libexec/platform-python"
5 },
6 "changed": false,
7 "ping": "pong"
8 }
```

## 模块

<https://docs.ansible.com/ansible-core/devel/collections/ansible/builtin/index.html#modules>

- ping 测试连接到主机
- shell 在目标主机运行shell 命令
- setup 收集远程主机信息。可以收集主机名、内核版本、网卡接口、IP 地址、操作系统版本、环境变量、CPU及核数、可用内存、可用磁盘等等信息

```
1 $ ansible remotes -m shell -a hostname
2 192.168.142.130 | CHANGED | rc=0 >>
3 localhost.localdomain
4
5 $ ansible remotes -m shell -a pwd
6 192.168.142.130 | CHANGED | rc=0 >>
7 /root
8
9 $ ansible remotes -m setup
```

## 编程开发

使用Ubuntu开发。

```
1 官方文档中
2 If you want to use Python API only for executing playbooks or modules,
 consider ansible-runner first.
```

<https://ansible-runner.readthedocs.io/en/latest/>

Ansible Runner负责运行Ansible和Playbook并收集输出

在Pycharm项目的虚拟环境中安装

```
1 $ pip install ansible ansible-runner
```

配置文件在/etc/ansible/ansible.cfg，其中inventory文件是同一个目录里的hosts。

在hosts文件中增加一个remotes组

```
1 [remotes]
2 192.168.142.130 ansible_ssh_user=root
```

编程参考 [https://ansible-runner.readthedocs.io/en/latest/python\\_interface/](https://ansible-runner.readthedocs.io/en/latest/python_interface/)

```
1 import ansible_runner
2
3 r = ansible_runner.run(host_pattern="remotes", module="shell",
4 module_args="hostname")
5 print(type(r))
6 print(r.status)
7 print(r.rc)
```

要报错，因为还没有把自己的公钥放到目标主机上。 `ssh-copy-id -i ~/.ssh/id_rsa.pub`

`root@192.168.142.130`

```
1 import ansible_runner
2 import json
3
4 # r = ansible_runner.run(host_pattern="remotes", module="shell",
5 module_args="hostname")
6 r = ansible_runner.run(host_pattern="remotes", module="setup")
7 print(type(r))
8 print(r.status) # 成功失败等状态
9 print(r.rc) # return code 0
10 print(r.stdout.read())
11 print('=====')
12 with open('/tmp/out', 'wt') as f:
13 for e in r.events:
14 # print(type(e), e)
15 print(e['event']) # runner_on_ok/ansible_facts
16 print('*****')
17 # s = json.dumps(e)
18 # f.write(s + '\n')
```

这样就可以获得主机数据，再提取其中关心的数据存入数据库即可。这些代码同学们根据需求自行完成。