

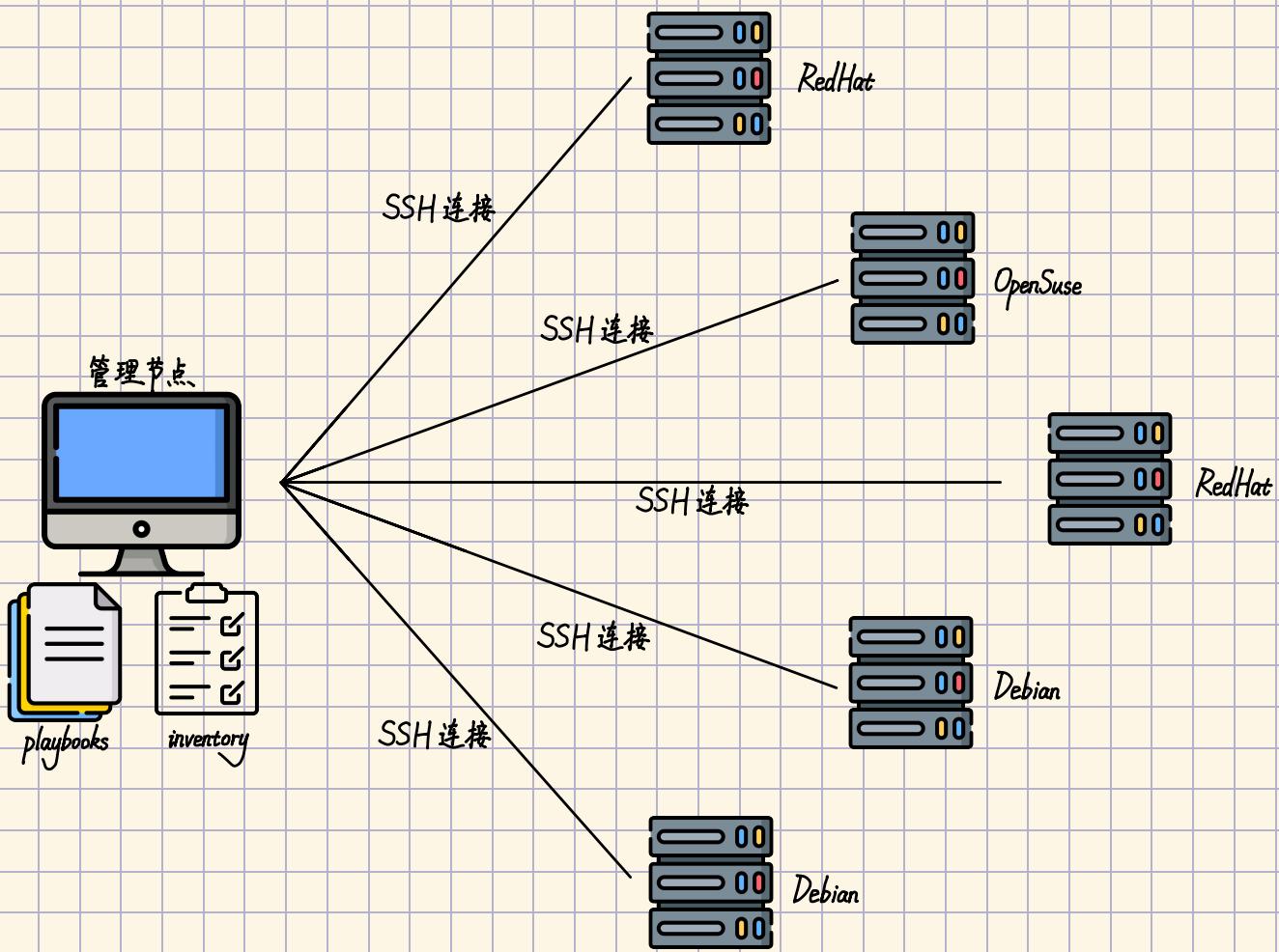
## Ansible 简介

Ansible 是一个部署一群远程机器的工具。这里的“远程机器”是指任何可以通过 SSH 登录的机器。

Ansible 解决了

- 大批量
- 自动化
- 减少重复劳动
- 支持多种操作系统

等运维痛点。





# Hello, Ansible !

## Host Inventory

Ansible 管理节点需要知道自己管理的主机有哪些，以及一些连接参数、分类。这些由 主机目录配置文件 来管理。

默认的文件是： /etc/ansible/hosts 。

```
vsa13700500:/home/i542016 # cat /etc/ansible/hosts
[admiring]
100.70.225.1
100.70.224.87

[rejected]
100.70.224.81
```

## # 用命令管理主机

先不细究语法，仅仅从感观上体会一下。

在所有的远程主机上，以 i542016 用户，执行 /bin/echo hello：

```
vsa13700500:/home/i542016 # ansible all -u i542016 -a "/bin/echo hello"
[WARNING]: Platform linux on host 100.70.225.1 is using the discovered Python interpreter at /usr/bin/python3.6, but
future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.11/reference_appendices/interpreter_discovery.html for more information.
100.70.225.1 | CHANGED | rc=0 >>
hello
[WARNING]: Platform linux on host 100.70.224.81 is using the discovered Python interpreter at /usr/bin/python3.6, but
future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.11/reference_appendices/interpreter_discovery.html for more information.
100.70.224.81 | CHANGED | rc=0 >>
hello
[WARNING]: Platform linux on host 100.70.224.87 is using the discovered Python interpreter at /usr/bin/python3.6, but
future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.11/reference_appendices/interpreter_discovery.html for more information.
100.70.224.87 | CHANGED | rc=0 >>
hello
```

发现了 warning(s)！分析 warning(s) 后，

参考 [https://docs.ansible.com/ansible-core/2.11/reference\\_appendices/interpreter\\_discovery.html#interpreter-discovery](https://docs.ansible.com/ansible-core/2.11/reference_appendices/interpreter_discovery.html#interpreter-discovery)，

知道要指定 Python 解释器，

再根据 [https://docs.ansible.com/ansible/latest/reference\\_appendices/config.html#the-configuration-file](https://docs.ansible.com/ansible/latest/reference_appendices/config.html#the-configuration-file)，

在当前路径下新建文件 ansible.cfg：

```
[defaults]
interpreter_python = /usr/bin/python3
```

再执行一遍，warning(s) 就消失了：

```
vsa13700500:/home/i542016 # ansible all -u i542016 -a "/bin/echo hello"
100.70.224.87 | CHANGED | rc=0 >>
hello
100.70.225.1 | CHANGED | rc=0 >>
hello
100.70.224.81 | CHANGED | rc=0 >>
hello
```

再多玩玩：

```
vsa13700500:/home/i542016 # ansible all -u i542016 -a "pwd"
100.70.224.87 | CHANGED | rc=0 >>
/home/i542016
100.70.225.1 | CHANGED | rc=0 >>
/home/i542016
100.70.224.81 | CHANGED | rc=0 >>
/home/i542016

vsa13700500:/home/i542016 # ansible all -u i542016 -a "uptime"
100.70.225.1 | CHANGED | rc=0 >>
12:45:33 up 2 days 3:57, 1 user,  load average: 0.14, 0.13, 0.09
100.70.224.87 | CHANGED | rc=0 >>
12:45:34 up 0:58, 1 user,  load average: 0.12, 0.15, 0.11
100.70.224.81 | CHANGED | rc=0 >>
12:45:34 up 2 days 4:10, 1 user,  load average: 0.65, 0.24, 0.14
```

```
vsa13700500:/home/i542016 # ansible all -u i542016 -m ping
100.70.224.81 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
100.70.225.1 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
100.70.224.87 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

```

vsa13700500:/home/i542016 # ansible rejected -u i542016 -m copy -a "src=/home/i542016/temp.txt dest=/tmp/temp.txt"
100.70.224.81 | CHANGED => {
    "changed": true,
    "checksum": "da39a3ee5e6b4b0d3255bfef95601890af80709",
    "dest": "/tmp/temp.txt",
    "gid": 58000,
    "group": "ADSUSER",
    "md5sum": "d41d8cd98f00b204e9800998ecf8427e",
    "mode": "0640",
    "owner": "i542016",
    "size": 0,
    "src": "/home/i542016/.ansible/tmp/ansible-tmp-1737003985.7966099-16059-13190324105499/source",
    "state": "file",
    "uid": 3542016
}

```

## # 用脚本管理主机

*YAML Ain't a Markup Language.*

Ansible 的脚本名字叫 Playbook，使用的是 YAML 格式，文件以 .yml 或 .yaml 为后缀。

执行 Playbook

编写 Playbook →

以某个用户身份连接远程机器

IP/主机组名

定义顺序执行的任务。  
任务中有动作(Action)，  
每个动作调用一个Ansible模块。

```

# Play: to present VM's info
- hosts: admiring
  remote_user: i542016
  tasks:
    - name: INFO
      shell: gmp_info

```

```

# Play: to copy a file
- hosts: rejected
  remote_user: i542016
  tasks:
    - name: COPY
      copy:
        src: /home/i542016/temp.txt
        dest: /tmp/temp_1.txt

```

```

vsa13700500:/home/i542016 # ansible-playbook playbooks/playbook_helloWorld.yml
[WARNING]: ansible.utils.display.initialize_locale has not been called, this may result in incorrectly calculated text widths that can cause Display to print incorrect line lengths

PLAY [admiring] ****
TASK [Gathering Facts] ****
ok: [100.70.225.1]
ok: [100.70.224.87]

TASK [INFO] ****
changed: [100.70.225.1]
changed: [100.70.224.87]

PLAY [rejected] ****
TASK [Gathering Facts] ****
ok: [100.70.224.81]

TASK [COPY] ****
ok: [100.70.224.81]

PLAY RECAP ****
100.70.224.81 : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
100.70.224.87 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
100.70.225.1  : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

常用的模块有 yum、copy、template 等，模块在 Ansible 中的作用，相当于 bash 脚本中的 yum、copy 这样的命令。  
之后会详细介绍。

Playbook 是指一个可被 Ansible 执行的 YAML 文件。

在一个 Playbook 文件中，可以针对多组主机进行不同的操作，针对每一组主机的所有操作就组成一个 Play。

一般一个 Playbook 中只包含一个 Play。

# Ansible 模块

相关文档 (All Modules) :

[https://docs.ansible.com/ansible/2.8/modules/list\\_of\\_all\\_modules.html](https://docs.ansible.com/ansible/2.8/modules/list_of_all_modules.html)

[https://docs.ansible.com/ansible/latest/collections/index\\_module.html](https://docs.ansible.com/ansible/latest/collections/index_module.html)



bash 无论是在命令行上执行，还是在脚本中，都需要调用 cd、ls、copy、yum 等命令。

模块就是 Ansible 的“命令”。模块是 Ansible 命令行和脚本中都需要调用的。常用的 Ansible 模块有 yum, copy, template 等。

bash 在调用命令时可以跟不同的参数，每个命令的参数都是该命令自定义的。

同样，Ansible 中调用模块也可以跟不同的参数，每个模块的参数也都是由模块自定义的。

模块的详细用法可以查阅官方文档，也可以通过命令 `ansible-doc <module_name>` 查看其用法。

## # 在命令行中使用模块

在命令行中：

- m 后面接调用模块的名字；

- a 后面接调用模块的参数；

vsa13700500:/home/i542016 # ansible rejected -u i542016 -m copy -a "src=/home/i542016/temp.txt dest=/tmp/temp.txt"

## # 在 Playbook 脚本中使用模块

在 Playbook 脚本中，任务中的每一个动作 (Action) 都是对模块的一次调用。

在每个动作中，

```

# Play: to present VM's info
- hosts: admiring
  remote_user: i542016
  tasks:
    - name: INFO
      shell: gmp_info
  
```

冒号前是模块的名字，  
后面是调用模块的参数。

Ansible 提供一些常用功能的模块，同时 Ansible 也提供 API，让用户可以自己写模块，使用的编程语言是 Python。

## # 常用模块

### - ping 模块

先检查能否通过 SSH 登录远程机器，再检查其 Python 版本能否满足要求。如果都满足则返回 pong。

### - debug 模块

打印输出信息。

#### ① 通过参数 msg 定义打印的字符串

msg 中可以嵌入变量，下面的例子注入了系统变量 (Ansible 在执行 Playbook 之前会收集一些比较常用的系统变量，这样在 Playbook 中不需要定义、直接使用即可)。

```

- hosts: rejected
  remote_user: i542016
  tasks:
    - name: Debug Module
      debug:
        msg: "System {{ inventory_hostname }} has gateway {{ ansible_default_ipv4.gateway }}"
  
```

vsa13700500:/home/i542016 # ansible-playbook playbooks/playbook\_helloWorld.yml  
[WARNING]: ansible.utils.display.initialize\_locale has not been called, this may result in incorrectly calculated text widths that can cause Display to print incorrect line lengths

```

PLAY [rejected] ****
TASK [Gathering Facts] ****
ok: [100.70.224.81]

TASK [Debug Module] ****
ok: [100.70.224.81] => {
    "msg": "System 100.70.224.81 has gateway 100.70.224.1"
}

PLAY RECAP ****
100.70.224.81 : ok=2    changed=0    unreachable=0   failed=0    skipped=0   rescued=0   ignored=0
  
```

## ② 通过参数 var 定义需要打印的变量

变量可以是系统变量，也可以是动态的执行结果（通过关键字 register 注入变量中）。

```
- hosts: admiring
  remote_user: i542016
  tasks:
    - name: Debug Module
      debug:
        var: hostvars[inventory_hostname]["ansible_default_ipv4"]["gateway"]
```

```
vsa13700500:/home/i542016 # ansible-playbook playbooks/playbook_helloWorld.yml
[WARNING]: ansible.utils.display.initialize_locale has not been called, this may result in incorrectly calculated text widths that can cause Display to print incorrect line lengths

PLAY [admiring] ****
TASK [Gathering Facts] ****
ok: [100.70.225.1]
ok: [100.70.224.87]

TASK [Debug Module] ****
ok: [100.70.225.1] => {
  "hostvars[inventory_hostname][\"ansible_default_ipv4\"][\\"gateway\\"]": "100.70.224.1"
}
ok: [100.70.224.87] => {
  "hostvars[inventory_hostname][\"ansible_default_ipv4\"][\\"gateway\\"]": "100.70.224.1"
}

PLAY RECAP ****
100.70.224.87 : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
100.70.225.1  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

```
- hosts: rejected
  remote_user: i542016
  tasks:
    - shell: uptime
      register: result
    - debug:
        var: result
```

```
vsa13700500:/home/i542016 # ansible-playbook playbooks/playbook_helloWorld.yml
[WARNING]: ansible.utils.display.initialize_locale has not been called, this may result in incorrectly calculated text widths that can cause Display to print incorrect line lengths

PLAY [rejected] ****
TASK [Gathering Facts] ****
ok: [100.70.224.81]

TASK [shell] ****
changed: [100.70.224.81]

TASK [debug] ****
ok: [100.70.224.81] => {
  "result": {
    "changed": true,
    "cmd": "uptime",
    "delta": "0:00:00.006074",
    "end": "2025-01-16 05:59:33.214695",
    "failed": false,
    "msg": "",
    "rc": 0,
    "start": "2025-01-16 05:59:33.208621",
    "stderr": "",
    "stderr_lines": [],
    "stdout": " 05:59:33  up 11:58, 1 user,  load average: 0.31, 0.22, 0.24",
    "stdout_lines": [
      " 05:59:33  up 11:58, 1 user,  load average: 0.31, 0.22, 0.24"
    ]
  }
}

PLAY RECAP ****
100.70.224.81 : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

## - copy 模块

remote\_src 参数另当别论

从管理节点上复制静态文件到远程主机上，并设置合理的文件权限。该模块在复制文件时，会先比较一下文件的 checksum，如果相同则不会复制，返回状态为 OK；如果不同才会复制，返回状态为 changed。

```

---  

- hosts: 100.70.224.87  

  remote_user: i542016  

  tasks:  

  - copy:  

    src: /lib/atm.txt  

    dest: /tmp/atm.txt  

    mode: 0644

```

复制管理节点上的某文件到远程机器上，并设置文件权限。

利用 mode 设置权限可以是数字(必须以0开头), 也可以是符号的形式(如 "u=rx,g=r,o=r")。

```
vsa13700500:/home/i542016 # ansible-playbook playbooks/playbook_helloWorld.yml  
[WARNING]: ansible.utils.display.initialize_locale has not been called, this may result in incorrectly calculated text widths that can cause Display to print incorrect line lengths
```

```
PLAY [100.70.224.87] ****
```

```
TASK [Gathering Facts] ****  
ok: [100.70.224.87]
```

checksum  
不同!

```
TASK [copy] ****  
changed: [100.70.224.87]
```

```
PLAY RECAP ****  
100.70.224.87 : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

再执行一遍：

```
vsa13700500:/home/i542016 # ansible-playbook playbooks/playbook_helloWorld.yml  
[WARNING]: ansible.utils.display.initialize_locale has not been called, this may result in incorrectly calculated text widths that can cause Display to print incorrect line lengths
```

```
PLAY [100.70.224.87] ****
```

```
TASK [Gathering Facts] ****  
ok: [100.70.224.87]
```

checksum  
相同!

```
TASK [copy] ****  
ok: [100.70.224.87]
```

```
PLAY RECAP ****  
100.70.224.87 : ok=2 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

在远程机器上确认：

```
10.47.19.91 ✓ 100.70.224.87 ✓ Connection Successful!
```

```
Command ls -l /tmp/atm.txt
```

```
Stdout -rw-r--r-- 1 i542016 ADSUSER 19 Jan 19 13:07 /tmp/atm.txt
```

6 4 4

```

---  

- hosts: 100.70.224.87  

  remote_user: i542016  

  tasks:  

  - copy:  

    src: /lib/atm.txt  

    dest: /tmp/atm.txt  

    backup: yes

```

backup 参数为 yes 的时候，

如果两个文件的 checksum 相同，不再进行复制操作；

如果不同，则发生复制操作，那么会先复制目标节点(远程机器)上的源文件。

(备份)

为了演示 "backup"，先改一下远程机器上的 dest 文件：

```
10.47.19.91 ✓ 100.70.224.87 ✓ Connection Successful!
```

```
Command echo "real madrid" > /tmp/atm.txt
```

执行 Playbook：

```
vsa13700500:/home/i542016 # ansible-playbook playbooks/playbook_helloWorld.yml  
[WARNING]: ansible.utils.display.initialize_locale has not been called, this may result in incorrectly calculated text widths that can cause Display to print incorrect line lengths
```

```
PLAY [100.70.224.87] ****
```

```
TASK [Gathering Facts] ****  
ok: [100.70.224.87]
```

✓ TASK [copy] \*\*\*\*  
changed: [100.70.224.87]

```
PLAY RECAP ****
```

```
100.70.224.87 : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

在远程机器上确认。

10.47.19.91 ✓ 100.70.224.87 ✓ Connection Successful!

Command ls /tmp/

atm.txt

atm.txt.24748.2025-01-19@13:16:33~

← 备份

10.47.19.91 ✓ 100.70.224.87 ✓ Connection Successful!

Command cat /tmp/atm.txt\*

atletico de madrid

Stdout real madrid

copy 模块中的 validate 参数接需要验证的命令。

当 copy 模块中加入了 validate 参数后，不仅需要成功复制文件，还需要 validate 命令返回成功的状态，整个模块的执行状态才算成功。

## - template 模块

如果复制的是静态文件，那么 copy 模块就足够了。

但是如果在复制的同时需要根据实际情况修改部分内容，那么就要用到 template 模块。（变量的表示法是 {{ }}）

## - file 模块

该模块可以用来创建和删除文件、文件夹和软链接（symlinks），也可以用来设置它们的权限。

```
---  
- hosts: 100.70.224.87  
  remote_user: i542016  
  tasks:  
    - file:  
        path: /tmp/bilibili.txt  
        state: touch  
        mode: "u=rw,g=r,o=r"
```

← 创建新文件

→ 创建新文件夹

```
---  
- hosts: 100.70.224.87  
  remote_user: i542016  
  tasks:  
    - file:  
        path: /tmp/bilibili  
        state: directory  
        mode: 0755
```

验证：

10.47.19.91 ✓ 100.70.224.87 ✓ Connection Successful!

Command ls -l /tmp/ | grep bilibili

```
7 5 5  
Stdout drwxr-xr-x 2 i542016 ADSUSER 4096 Jan 19 13:36 bilibili  
-rw-r--r-- 1 i542016 ADSUSER 0 Jan 19 13:33 bilibili.txt  
rw r r
```

改变文件的权限：

```
---  
- hosts: 100.70.224.87  
  remote_user: i542016  
  tasks:  
    - file:  
        path: /tmp/bilibili.txt  
        mode: 0777
```

10.47.19.91 ✓ 100.70.224.87 ✓ Connection Successful!

Command ls -l /tmp/ | grep bilibili.txt

```
7 7 7  
Stdout -rwxrwxrwx 1 i542016 ADSUSER 0 Jan 19 13:33 bilibili.txt
```

## - user 模块

该模块可以增、删、改远程机器的用户账户，并为其设置账户的属性。

## - yum/dnf/zypper 模块

该模块可用于安装最新版本/指定版本的包、删除包等等。

## - service 模块

```
tasks:  
- service:  
  name: victoriameetrics  
  state: started
```

开启服务

```
tasks:  
- service:  
  name: victoriameetrics  
  state: restarted
```

重启服务

```
tasks:  
- service:  
  name: victoriameetrics  
  state: reloaded
```

重载服务

```
tasks:  
- service:  
  name: victoriameetrics  
  state: stopped
```

关服务

```
tasks:  
- service:  
  name: victoriameetrics  
  enabled: yes
```

设置开机启动

## - firewalld 模块

该模块为某服务和端口添加 firewalld 规则。

## - shell 模块

在远程机器上通过 /bin/sh 执行命令。

但如果一个操作可以通过模块实现，那么建议不要使用 shell 或者 command 这样通用的命令模块。因为通用的命令模块不会根据具体操作的特点进行状态 (status) 判断。即使在没有必要再重新执行的时候，它还是会重新执行一遍。

该模块：

- ① 支持 ~、<、>、!、;、&。
- ② 支持调用脚本。
- ③ 支持在执行命令前改变工作目录。

```
---  
- hosts: admiring  
  remote_user: i542016  
  tasks:  
    - shell: echo "hello2025" > temp.txt  
      args:  
        chdir: /tmp/
```

10.47.19.91 ✓ 100.70.224.87 ✓ Connection Successful!

Command cat /tmp/temp.txt

Stdout hello2025

- ④ 支持在执行命令前改变工作目录，且仅在文件 <something.txt> 不存在时才执行命令。

args:  
 ↗ chdir: /tmp/  
 ↗ creates: <something.txt>

- ⑤ 指定用 bash 运行命令。

## Ansible 的配置文件

1.5 版本以前是另外一种顺序

Ansible 会按照下面的顺序查找配置文件，并使用第一个发现的：\$ANSIBLE\_CONFIG → ansible.cfg → ~/.ansible.cfg → /etc/ansible/ansible.cfg。  
([https://docs.ansible.com/ansible/latest/reference\\_appendices/config.html#the-configuration-file](https://docs.ansible.com/ansible/latest/reference_appendices/config.html#the-configuration-file))

配置文件被分为很多的 sections，如 [defaults]，[ssh\_connection] 等等，每个 section 都包含了相关的 settings。

阅读：

<https://www.ansiblepilot.com/articles/ansible.cfg/>

[https://tekeneed.com/creating-and-managing-ansible-configuration-file-linux/#google\\_vignette](https://tekeneed.com/creating-and-managing-ansible-configuration-file-linux/#google_vignette)

## 主机清单

官方文档：[https://docs.ansible.com/ansible/latest/intro\\_inventory.html](https://docs.ansible.com/ansible/latest/intro_inventory.html)

主机清单 (Host Inventory) 告诉 Ansible 需要管理哪些主机，以及这些主机的分类及相关信息。

地域/功能

主机清单的配置文件默认是 /etc/ansible/hosts，也可以在 Ansible 的配置文件 (/etc/ansible/ansible.cfg) 中修改其路径：inventory = <path/to/file>。

还可以通过参数 -i 或 --inventory-file 来传递主机清单的配置文件：ansible-playbook -i <path/to/file> playbook.yml。（可以传多个 -i）

主机清单甚至可以是一个文件夹，而不仅仅是单个文件。

## Organizing inventory in a directory

You can consolidate multiple inventory sources in a single directory. The simplest version of this is a directory with multiple files instead of a single inventory file. A single file gets difficult to maintain when it gets too long. If you have multiple teams and multiple automation projects, having one inventory file per team or project lets everyone easily find the hosts and groups that matter to them.

You can also combine multiple inventory source types in an inventory directory. This can be useful for combining static and dynamic hosts and managing them as one inventory. The following inventory directory combines an inventory plugin source, a dynamic inventory script, and a file with static hosts:

```

inventory/
openstack.yml      # configure inventory plugin to get hosts from OpenStack cloud
dynamic-inventory.py # add additional hosts with dynamic inventory script
on-prem            # add static hosts and groups
parent-groups      # add static hosts and groups

```

官方文档：  
[https://docs.ansible.com/ansible/latest/intro\\_inventory.html#organizing-inventory-in-a-directory](https://docs.ansible.com/ansible/latest/intro_inventory.html#organizing-inventory-in-a-directory)

You can target this inventory directory as follows:

```
ansible-playbook example.yml -i inventory
```

给远程主机分组，以便于在 Playbook 中使用。

[ ] 内是组名

```
vsa13700500:/home/i542016 # cat /etc/ansible/hosts
```

[admiring]

100.70.225.1

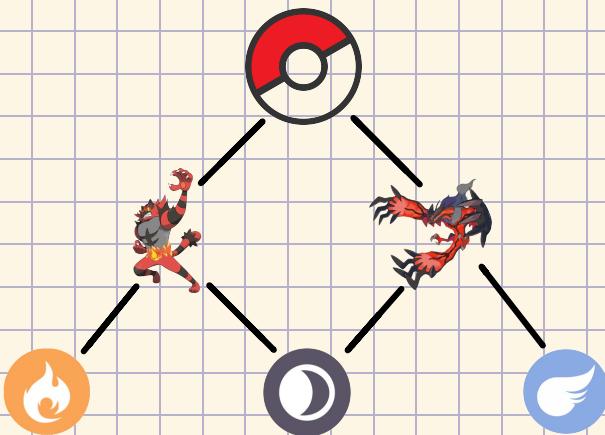
100.70.224.87

[rejected]

100.70.224.81

官方文档：[https://docs.ansible.com/ansible/latest/intro\\_inventory.html#inventory-basics-formats-hosts-and-groups](https://docs.ansible.com/ansible/latest/intro_inventory.html#inventory-basics-formats-hosts-and-groups)

分组也支持 parent - children 关系。



[ Incineroar ]

Dark

Fire

[ Yveltal ]

Dark

Flying

[ Pokemon : children ]

Incineroar suffix

Yveltal

官方文档：

[https://docs.ansible.com/ansible/latest/inventory\\_guide/intro\\_inventory.html#grouping-groups-parent-child-group-relationships](https://docs.ansible.com/ansible/latest/inventory_guide/intro_inventory.html#grouping-groups-parent-child-group-relationships)

Ansible 可以在主机清单中指定主机的连接参数，比如连接方法、用户等，用空格分隔多个参数。 参数的含义

官方文档：[https://docs.ansible.com/ansible/latest/inventory\\_guide/intro\\_inventory.html#connecting-to-hosts-behavioral-inventory-parameters](https://docs.ansible.com/ansible/latest/inventory_guide/intro_inventory.html#connecting-to-hosts-behavioral-inventory-parameters)

[https://docs.ansible.com/ansible/latest/inventory\\_guide/intro\\_inventory.html#adding-variables-to-inventory](https://docs.ansible.com/ansible/latest/inventory_guide/intro_inventory.html#adding-variables-to-inventory)

Ansible 支持在主机清单文件中，或与主机清单文件同目录的特定子目录和文件中，声明/定义变量。

官方文档：[https://docs.ansible.com/ansible/latest/inventory\\_guide/intro\\_inventory.html#adding-variables-to-inventory](https://docs.ansible.com/ansible/latest/inventory_guide/intro_inventory.html#adding-variables-to-inventory)

为单个远程主机指定参数：

```
[atlanta]
host1 http_port=80 maxRequestsPerChild=808
host2 http_port=303 maxRequestsPerChild=909
```

为一个组的远程主机指定参数：

```
[atlanta]
host1
host2

[atlanta:vars]
ntp_server=ntp.atlanta.example.com
proxy=proxy.atlanta.example.com
```

假设主机清单文件为 /etc/ansible/hosts，

那么相关的 Host 和 Group 变量可以放在 /etc/ansible/host\_vars/ 和 /etc/ansible/group\_vars/ 下的 主机/组 同名的文件（支持 .yml, .yaml, .json, 无文件扩展名）或文件夹。

原文：Ansible will read all the files in these directories in lexicographical order.

原文：Ansible loads host and group variable files by searching paths relative to the inventory or the playbook file.

除了主机清单文件的同级目录下，group\_vars/ 与 host\_vars/ 还可以放在 Playbook 文件的同级目录下。如果两个目录下都存在变量文件，那么 Playbook 目录下的值会 覆盖 主机清单文件目录下的变量的值。

官方文档：[https://docs.ansible.com/ansible/latest/inventory\\_guide/intro\\_inventory.html#organizing-host-and-group-variables](https://docs.ansible.com/ansible/latest/inventory_guide/intro_inventory.html#organizing-host-and-group-variables)

变量可以被 parent - children 关系继承。

# Ansible 的脚本：Playbook

如何执行 Ansible 的 Playbook 呢？Ansible 提供了一个单独的命令：`ansible-playbook`。以此 playbook 为例：

基本用法：

```
vsa13700500:/home/i542016 # ansible-playbook playbooks/playbook_helloWorld.yml
[WARNING]: ansible.utils.display.initialize_locale has not been called, this may result in incorrectly calculated text
widths that can cause Display to print incorrect line lengths

PLAY [all] ****
TASK [Gathering Facts] ****
ok: [100.70.224.81]
ok: [100.70.225.1]
ok: [100.70.224.87]

TASK [shell] ****
changed: [100.70.224.81]
changed: [100.70.224.87]
changed: [100.70.225.1]

PLAY RECAP ****
100.70.224.81      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
100.70.224.87      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
100.70.225.1      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

查看输出的细节：

```
vsa13700500:/home/i542016 # ansible-playbook playbooks/playbook_helloWorld.yml --verbose
Using /home/i542016/ansible.cfg as config file
[WARNING]: ansible.utils.display.initialize_locale has not been called, this may result in incorrectly calculated text
widths that can cause Display to print incorrect line lengths

PLAY [all] ****
TASK [Gathering Facts] ****
ok: [100.70.224.81]
ok: [100.70.225.1]
ok: [100.70.224.87]

TASK [shell] ****
changed: [100.70.224.81] => {"changed": true, "cmd": "echo \\\"hello2025\\\"", "delta": "0:00:00.003559", "end": "2025-01-24
02:27:06.461150", "msg": "", "rc": 0, "start": "2025-01-24 02:27:06.457591", "stderr": "", "stderr_lines": [], "stdout"
: "hello2025", "stdout_lines": ["hello2025"]}
changed: [100.70.225.1] => {"changed": true, "cmd": "echo \\\"hello2025\\\"", "delta": "0:00:00.003702", "end": "2025-01-24
02:27:06.477609", "msg": "", "rc": 0, "start": "2025-01-24 02:27:06.473907", "stderr": "", "stderr_lines": [], "stdout"
: "hello2025", "stdout_lines": ["hello2025"]}
changed: [100.70.224.87] => {"changed": true, "cmd": "echo \\\"hello2025\\\"", "delta": "0:00:00.003784", "end": "2025-01-24
02:27:06.481676", "msg": "", "rc": 0, "start": "2025-01-24 02:27:06.477892", "stderr": "", "stderr_lines": [], "stdout"
: "hello2025", "stdout_lines": ["hello2025"]}

PLAY RECAP ****
100.70.224.81      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
100.70.224.87      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
100.70.225.1      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

查看该 playbook 影响哪些主机(hosts)。

```
vsa13700500:/home/i542016 # ansible-playbook playbooks/playbook_helloWorld.yml --list-hosts
```

playbook: playbooks/playbook\_helloWorld.yml

```
play #1 (all): all    TAGS: []
  pattern: ['all']
  hosts (3):
    100.70.224.81
    100.70.225.1
    100.70.224.87
```

另外，`-f` 选项/参数的含义是：“specify number of parallel processes to use (default = 5)”。

更多的关于 `ansible-playbook` 的用法可通过命令 `ansible-playbook --help` 查看。

另外，还有一个工具 `ansible-lint`，它用来 validate playbooks，可帮助 CI。

最基本的 Playbook 分为三个部分

主机和用户	在哪些机器上、以什么身份执行	hosts & users
任务列表	都有什么任务要执行	tasks
响应事件	可被触发哪些事件	handlers

示例：

```
1 ---  
2   - hosts: webServers  
3     user: root  
4   vars:  
5     http_post: 80  
6     max_clients: 200  
7  
8   tasks:  
9     - name: to ensure that apache is at the latest version  
10    | yum: pkg=httpd state=latest  
11     - name: to write the apache config file  
12    | template: src=/srv/httpd.j2 dest=/etc/httpd.conf  
13     notify:  
14       - restart apache  
15     - name: to ensure that apache is running  
16       | service: name=httpd state=running  
17  
18   handlers:  
19     - name: restart apache  
20       | service: name=httpd state=restarted
```

} hosts & users

} tasks

} handlers

hosts

主机的 ip, 或者主机组名, 或者关键字 all.

注意区别

remote\_user

"An Ansible setting that controls the SSH user whom Ansible is using to connect (ssh <remote-user>@<host>)."

在远程机器上以哪个用户身份执行。

become\*

与切换用户身份相关。

每个动作(action)

任务列表里的任务(tasks)是从上至下顺序执行的, 如果中间发生错误, 那么整个 Playbook 会中止。任务中的都是对模块的一次调用, 只是使用的参数和变量不同而已。每个任务最好有 name 属性, 这是供人读的, 没有实际的操作, 然后会在命令行里输出, 提示用户执行的情况。

关于参数的写法,

上面的示例使用的是 "`<key> = <value>`" 的传参方式,

tasks:

```
- name: to ensure that apache is running  
  service: name=httpd state=running
```

也可以分到多行去,

tasks:

```
- name: to ensure that apache is running  
  service:  
    name=httpd  
    state=running
```

或用 YAML 的字典格式。

tasks:

```
- name: to ensure that apache is running  
  service:  
    name: httpd  
    state: running
```

任务(tasks)中每个动作(Action)都会调用一个模块，然后在模块中检查当前系统状态是否需要执行。

如果本次执行了，那么Action会得到返回值 changed。

如果不需要执行，那么Action会得到返回值 ok。

模块的执行状态的具体判断规则由各个模块自己决定和实现。例如，copy模块的判断方法是比较文件的 checksum，

代码在 <https://github.com/ansible/ansible/blob/devel/lib/ansible/modules/copy.py>。

最后，学习一下响应事件(handlers)。

Handlers 里有多个 handler，handler 中的每个动作也是对模块的一次调用。

每个 handler，仅在任务(tasks)中被 notify 时，才会被执行。

任务何时会 notify 其 handler 呢？前面提过，任务列表中的任务都是有状态的(changed/ok)，当且仅当 任务的执行状态为 changed 时，该任务才会 notify 其 handler。但一个 handler 最多只执行一次，不论它被 notify 了多少次。

Handlers，会在所有的 tasks 都执行完之后才开始执行。且 handlers 按定义的顺序，而非被 notify 的顺序，执行。

Playbook 中，tasks 与 handlers 的执行顺序，写成伪代码大概是：

```
tasks = [...]
handlers = [...]

if_notified = {
    h: False for h in handlers
}

for t in tasks:
    status = exec(t)
    if status is changed:
        if_notified[t.h] = True

for h in handlers:
    if if_notified[h] is True:
        exec(h)
```

Ansible 官方提供了一些较常用的、经过测试的 Playbook 例子：<https://github.com/ansible/ansible-examples>。

此外，Ansible 提供了一个 Playbook 的分享平台（用户自己上传、未必经过测试）：<https://galaxy.ansible.com/ui/>。

## Playbook的复用

不能站在巨人肩膀上的编程语言不是好语言。支持复用机制可以节省重复工作的时间，还能提高可维护性（避免臃肿）。

Ansible 支持两种代码复用机制，

一是复用单个 Playbook 脚本，`include` 类似于编程语言中的 `include`，是复用单个文件的；

二是复用实现了特定功能的文件夹，`role` 类似于编程语言中的 `import`，是复用一组文件、形成完整功能的。

### # include

官方文档：[https://docs.ansible.com/ansible/2.9/user\\_guide/playbooks\\_reuse\\_includes.html](https://docs.ansible.com/ansible/2.9/user_guide/playbooks_reuse_includes.html)。

基本用法：

```
vsa13700500:/home/i542016 # ls playbooks/
hello.yml main.yml
vsa13700500:/home/i542016 #
vsa13700500:/home/i542016 # cat playbooks/hello.yml
---
- name: say hello
  shell: echo "hello"

vsa13700500:/home/i542016 # cat playbooks/main.yml
---
- hosts: rejected
  remote_user: i542016
  gather_facts: False
  tasks:
    - name: to include
      include: hello.yml
```

在 main Playbook 中调用 `include` 方法即可

```
vsa13700500:/home/i542016 # ansible-playbook playbooks/main.yml --verbose
Using /home/i542016/ansible.cfg as config file
[WARNING]: ansible.utils.display.initialize_locale has not been called, this may result in incorrectly calculated text widths that can cause Display to print incorrect line lengths

PLAY [rejected] ****
TASK [say hello] ****
changed: [100.70.224.81] => {"changed": true, "cmd": "echo \"hello\"", "delta": "0:00:00.006524", "end": "2025-01-29 02:13:31.743423", "msg": "", "rc": 0, "start": "2025-01-29 02:13:31.736899", "stderr": "", "stderr_lines": [], "stdout": "hello", "stdout_lines": ["hello"]}

PLAY RECAP ****
100.70.224.81 : ok=1    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

使用参数：

```
vsa13700500:/home/i542016 # ls playbooks/
hello.yml main.yml
vsa13700500:/home/i542016 # cat playbooks/hello.yml
---
- name: say something
  shell: echo {{ something }}
```

定义参数

```
vsa13700500:/home/i542016 # cat playbooks/main.yml
---
- hosts: rejected
  remote_user: i542016
  gather_facts: False
  tasks:
    - name: to include
      include: hello.yml something="Anhao"
```

传入参数

只能有一个空格

```
TASK [say something] ****
changed: [100.70.224.81] => {"changed": true, "cmd": "echo Anhao", "delta": "0:00:00.004758", "end": "2025-01-29 02:24:18.069347", "msg": "", "rc": 0, "start": "2025-01-29 02:24:18.064589", "stderr": "", "stderr_lines": [], "stdout": "Anhao", "stdout_lines": ["Anhao"]}
```

此外，还可以通过变量传参。

```
vsa13700500:/home/i542016 # cat playbooks/main.yml
---
- hosts: rejected
  remote_user: i542016
  gather_facts: False

  tasks:
    - name: to include
      include: hello.yml
    vars:
      something: Anhao
```

```
vsa13700500:/home/i542016 # cat playbooks/main.yml
---
- hosts: rejected
  remote_user: i542016
  gather_facts: False

  vars:
    something: Anhao

  tasks:
    - name: to include
      include: hello.yml
```

在 Ansible 2.0 之后，handlers 里面也可以加入 include。

## # role

官方文档：[https://docs.ansible.com/ansible/2.9/user\\_guide/playbooks\\_reuse\\_roles.html](https://docs.ansible.com/ansible/2.9/user_guide/playbooks_reuse_roles.html)

例：安装和配置 Apache，需要

- ① 安装包
- ② 复制模板（需模板文件）
- ③ 启动

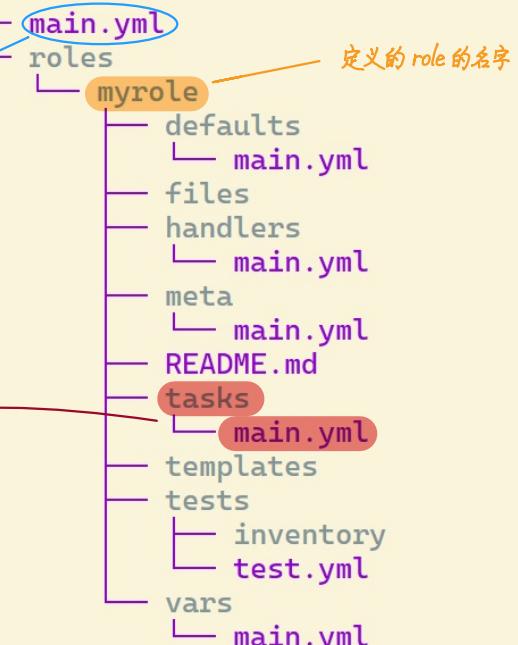
这些文件，都可放在一个 role 里，以供不同的 Playbook 文件复用。

Ansible 十分提倡在 Playbook 中使用 role，并且提供了一个分享 role 的平台 Ansible Galaxy，<https://galaxy.ansible.com/>。

在 Ansible 中，通过遵循特定的目录结构，就可以实现对 role 的定义。

```
vsa13700500:/home/i542016/playbooks # tree
```

```
---  
- hosts: rejected  
  remote_user: i542016  
  gather_facts: False  
  
  roles:  
    - myrole
```



在写 role 的时候，

一般都要包含 role 的入口文件 `roles/<role_name>/tasks/main.yml`，

其他的文件和目录，可以根据需求选择是否加入。

11 directories, 8 files

如果文件 `roles/<role_name>/tasks/main.yml` 存在，则其中列出的所有任务都将被添加到 Play 中。

如果文件 `roles/<role_name>/handlers/main.yml` 存在，则其中列出的所有响应都将被添加到 Play 中。

如果文件 `roles/<role_name>/vars/main.yml` 存在，则其中列出的所有变量都将被添加到 Play 中。

如果文件 `roles/<role_name>/defaults/main.yml` 存在，则其中列出的所有变量都将被添加到 Play 中。

如果文件 `roles/<role_name>/meta/main.yml` 存在，则其中列出的所有依赖的 role 都将被添加到 Play 中。

此外，下面的文件不需要绝对或相对路径，和放在同一个目录下的文件一样，直接使用即可：

- △ copy 或者 script 使用 roles/<role\_name>/files/ 下的文件；
- △ template 使用 roles/<role\_name>/templates/ 下的文件；
- △ include 使用 roles/<role\_name>/tasks/ 下的文件；

接着，学习一下 role 中的参数的用法（定义与使用）。

```
vsa13700500:/home/i542016/playbooks # tree
.
└── main.yml
    ├── roles
    │   └── myrole
    │       ├── defaults
    │       │   └── main.yml
    │       ├── files
    │       ├── handlers
    │       │   └── main.yml
    │       ├── meta
    │       │   └── main.yml
    │       ├── README.md
    │       ├── tasks
    │       │   └── main.yml
    │       ├── templates
    │       ├── tests
    │       │   └── inventory
    │       │       └── test.yml
    │       └── vars
    │           └── main.yml
11 directories, 8 files

vsa13700500:/home/i542016/playbooks # ansible-playbook main.yml
[WARNING]: ansible.utils.display.initialize_locale has not been called, this may result in incorrectly calculated text widths that can cause Display to print incorrect line lengths

PLAY [rejected] ****
TASK [myrole : to use parameters] ****
ok: [100.70.224.81] => {
    "msg": "XY"
}

TASK [myrole : to use parameters] ****
ok: [100.70.224.81] => {
    "msg": "ORAS"
}

PLAY RECAP ****
100.70.224.81 : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

使用方法一

```
---  
- hosts: rejected  
  remote_user: i542016  
  gather_facts: False  
  
  roles:  
    - { role: myrole, param: "XY" }  
    - { role: myrole, param: "ORAS" }
```

使用方法二

```
---  
- hosts: rejected  
  remote_user: i542016  
  gather_facts: False  
  
  roles:  
    - role: myrole  
      param: "XY"  
    - role: myrole  
      param: "ORAS"
```

一旦指定默认参数，若在调用时传参数了，那么就使用传入的参数值，否则就使用默认值。

```
vsa13700500:/home/i542016/playbooks # tree
.
└── main.yml
    ├── roles
    │   └── myrole
    │       ├── defaults
    │       │   └── main.yml
    │       ├── files
    │       ├── handlers
    │       │   └── main.yml
    │       ├── meta
    │       │   └── main.yml
    │       ├── README.md
    │       ├── tasks
    │       │   └── main.yml
    │       ├── templates
    │       ├── tests
    │       │   └── inventory
    │       │       └── test.yml
    │       └── vars
    │           └── main.yml
11 directories, 8 files
```

param: HGSS

同时，role 还可以与条件语句 when 一起。

```
vsa13700500:/home/i542016/playbooks # tree
.
├── main.yml
└── roles
    └── myrole
        ├── defaults
        │   └── main.yml
        ├── files
        ├── handlers
        │   └── main.yml
        ├── meta
        │   └── main.yml
        ├── README.md
        ├── tasks
        │   └── main.yml
        ├── templates
        ├── tests
        │   └── inventory
        │       └── test.yml
        └── vars
            └── main.yml

11 directories, 8 files

vsa13700500:/home/i542016/playbooks # ansible-playbook main.yml
[WARNING]: ansible.utils.display.initialize_locale has not been called, this may result in incorrectly calculated text widths that can cause Display to print incorrect line lengths

PLAY [rejected] ****
TASK [Gathering Facts] ****
[WARNING]: Platform linux on host 100.70.224.81 is using the discovered Python interpreter at /usr/bin/python3.6, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.11/reference_appendices/interpreter_discovery.html for more information.
ok: [100.70.224.81]

TASK [myrole : to use parameters] ****
ok: [100.70.224.81] => {
    "msg": "BW"
}

TASK [myrole : to use parameters] ****
ok: [100.70.224.81] => {
    "msg": "HGSS"
}

PLAY RECAP ****
100.70.224.81 : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

如果一个 Playbook 中同时出现 roles 和 tasks，则调用顺序为 pre\_tasks → roles → tasks → post\_tasks。

```
vsa13700500:/home/i542016 # cat playbooks/main.yml
---
- hosts: rejected
  remote_user: i542016
  gather_facts: False

  pre_tasks:
    - shell: echo "FRLG"
  roles:
    - myrole
  tasks:
    - shell: echo "BW2"
  post_tasks:
    - shell: echo "ORAS"
```

指 <main>.yml 文件的 working dictionary

除了放在当前子目录 roles 下，role 文件夹还可以放在其他位置，以便让更多 Playbook 共享同一组 role。

(1) 无论 Ansible 中对 roles path 是如何设置的，放在当前子目录 roles 文件夹下的 role 都会被找到；

(2) 如果定义了环境变量 ANSIBLE\_ROLES\_PATH，那么 Ansible 也会搜索该文件夹下的 role；

(3) 在 Ansible 配置文件中的 roles\_path 定义（如果有多个目录，则使用冒号分隔）；

但如果定义了(2)，则(3)会失效。

(4) Ansible 还提供了一个默认的文件夹 — /etc/ansible/roles/。

[https://docs.ansible.com/ansible/latest/reference\\_appendices/config.html#default-roles-path](https://docs.ansible.com/ansible/latest/reference_appendices/config.html#default-roles-path)

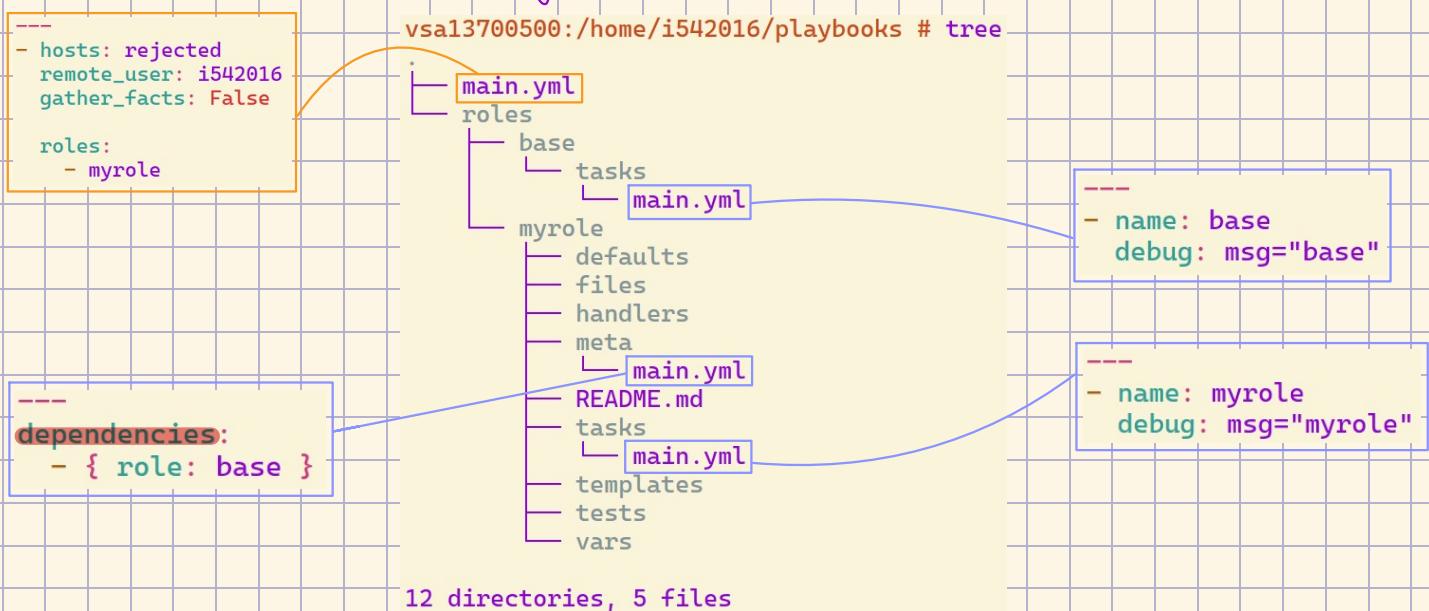
如果需对任务或变量进一步划分

(而非全部丢进 `main` 里)，  
则可以通过 `include` 与 `include- vars` 引入  
`<nonMain>` 文件中的任务和变量。

```
vsa13700500:/home/i542016/playbooks/roles/myrole # tree
.
+- defaults
+- files
+- handlers
+- meta
+- README.md
+- tasks
   +- configure.yml
   +- install.yml
   +- main.yml
+- templates
+- tests
+- vars
   +- Debian.yml
   +- main.yml
   +- RedHat.yml

8 directories, 7 files
```

Ansible 也支持 roles 之间的依赖（前驱关系）：`base → myrole`。



```
vsa13700500:/home/iz42016/playbooks # ansible-playbook main.yml
[WARNING]: ansible.utils.display.initialize_locale has not been called, this may result in incorrectly calculated text
widths that can cause Display to print incorrect line lengths

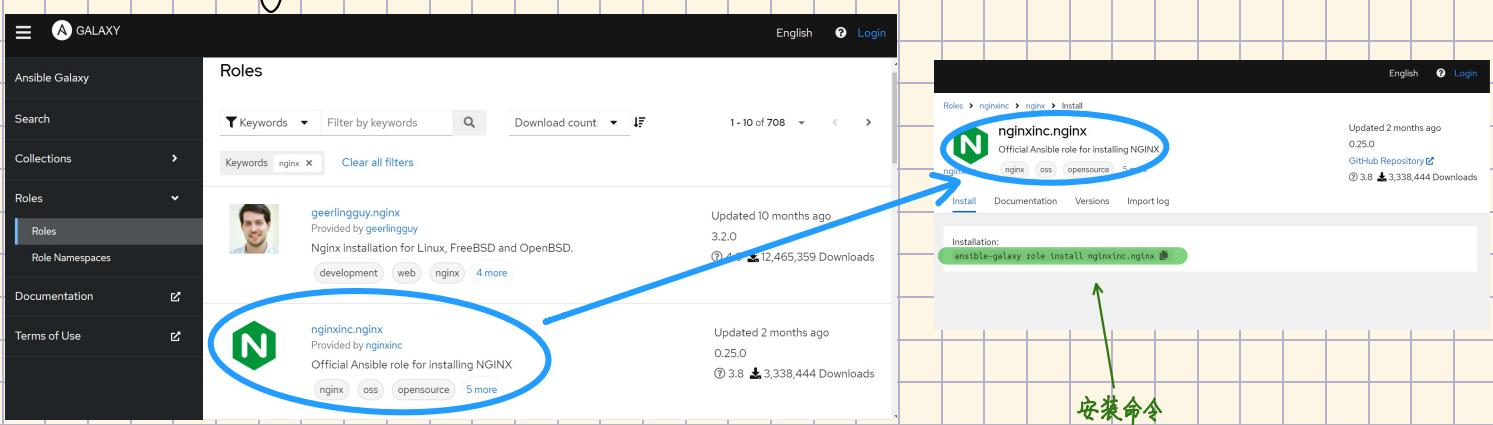
PLAY [rejected] ****
TASK [base : base] ****
ok: [100.70.224.81] => {
    "msg": "base"
}

TASK [myrole : myrole] ****
ok: [100.70.224.81] => {
    "msg": "myrole"
}

PLAY RECAP ****
100.70.224.81 : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

roles 的 dependency 相当于隐式地加入了 include

最后，介绍一下 Ansible Galaxy 的用法。



Ansible 提供了 `ansible-galaxy install` 命令来下载 roles 到本地, <https://docs.ansible.com/ansible/latest/cli/ansible-galaxy.html>。  
除了一个个地下载之外, 还可以批量下载 (-r 参数)。

`ansible-galaxy install -r roles.txt`

```
user1.role1,v1.0.0
user2.role2,v0.5
user2.role3
```

`ansible-galaxy install -r requirement.yml`

```
# from galaxy
- src: yatesr.timezone

# from github
- src: https://github.com/bennojoy/nginx

# from github installing to a relative path
- src: https://github.com/bennojoy/nginx
  path: vagrant/roles/

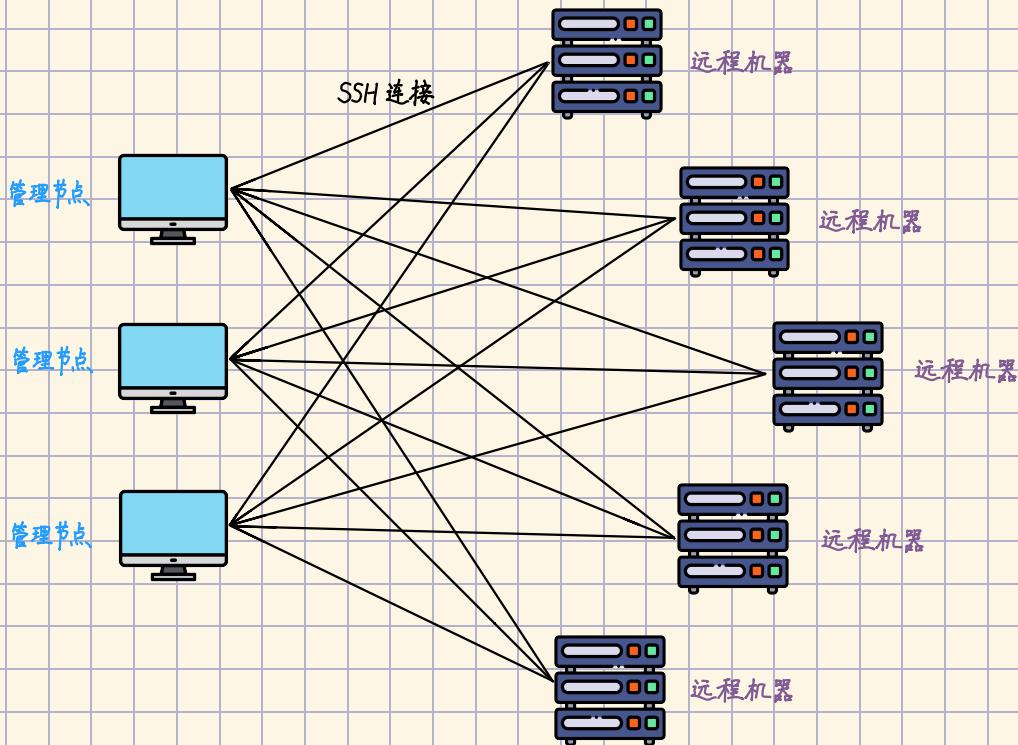
# from github, overriding the name and specifying a specific tag
- src: https://github.com/bennojoy/nginx
  version: master
  name: nginx_role
```

如何写 role 呢?

大致思路是, 先写一个单个的 Playbook, 再改造单个的 Playbook 为 role。

## Ansible Tower

个人见解：相当于 Platform



每位管理员都需要配置所有远程机器的 SSH 连接。

Ansible Tower 是中心化的 Ansible 管理节点，远程机器的管理员通过登录 Tower 来运行 Playbook、无需每位管理员都在自己的电脑上配置 SSH 连接。

