

TS-ansible 常用参数

示例: `ansible-playbook -i ~/workers/inventory -l pluto113 ~/scripts/ansible/opa/start_worker.yml`

`-playbook`

你可以为 `playbook` 中的每一个 `play`，个别地选择操作的目标机器是哪些，以哪个用户身份去完成要执行的步骤。

`playbook`是由一个或多个`play`组成的列表，`play`的主要功能在于将事先归并为一组的主机装扮成事先通过`ansible`中的`task`定义好的角色。从根本上来讲，所谓的`task`无非是调用`ansible`的一个`module`。将多个`play`组织在一个`playbook`中，即可以让它们联合起来按事先编排机制完成某一任务。自然也可以在其中设置选择使用什么用户去跑脚本。

```
cat ssh_copy.yml
---
- hosts: all
  user: xg
  tasks:
    - name: ssh-copy
      authorized_key: user=xg key="{{ lookup('file', '~/.ssh/id_rsa.pub') }}"
    - name: ssh-root-copy
      authorized_key: user=root key="{{ lookup('file', '~/.ssh/id_rsa.pub') }}"
```

`-i` (INVENTORY)

指定host文件，即工作的对象的地址

`-l` (limit SUBSET)

进一步限制所选的主机对象

示例: `ansible -i ~/workers/inventory p_worker -m shell -a "reboot" -b`

`-m` (MODULE_NAME)

模块名称。这里有些会使用`shell`或者`command`模。`shell` 模块通过`shell`程序执行，默认是`/bin/sh<>| ; , &`可用。但这样有潜在的 `shell` 注入风险。而 `command` 模块命令不会使用 `shell` 执行，像`$HOME`这样的变量是不可用的，还有像`<>| , &`都将不可用。所以最好是使用`ansible`本身自带的模块。

`-a` (MODULE_ARGS)

模块参数，即所需执行命令

`-b` (become)

不返回密码提示