

- [配置文件优先级](#)
- [配置文件详解](#)
 - [配置文件分段说明](#)
 - [配置参数说明](#)
- [关于ssh连接一些常见的错误说明](#)

配置文件优先级

ansible的配置文件名为ansible.cfg，它一般会存在于四个地方：

- ANSIBLE_CONFIG：首先，Ansible命令会检查该环境变量，及这个环境变量将指向的配置文件
- ./ansible.cfg：当前工作目录，即当前执行ansible指令的目录，如果ANSIBLE_CONFIG环境变量未定义，则优先使用该配置文件
- ~/.ansible.cfg：当前用户家目录下的一个隐藏文件，如果当前工作目录下不存在ansible.cfg配置文件，则会查找用户家目录下的该隐藏文件
- /etc/ansible/ansible.cfg：默认配置文件，如果上面两个路径下的ansible.cfg都不存在，则使用该文件

需要说明的是，配置文件中所有的配置项都可以通过环境变量的方式来定义，而环境变量定义的配置项具有最高优先级，会覆盖掉所有配置文件中的配置项

配置文件详解

配置文件分段说明

ansible.cfg的配置默认分为十段：

- [defaults]：通用配置项
- [inventory]：与主机清单相关的配置项
- [privilege_escalation]：特权升级相关的配置项
- [paramiko_connection]：使用paramiko连接的相关配置项，Paramiko在RHEL6以及更早的版本中默认使用的ssh连接方式
- [ssh_connection]：使用OpenSSH连接的相关配置项，OpenSSH是Ansible在RHEL6之后默认使用的ssh连接方式
- [persistent_connection]：持久连接的配置项
- [accelerate]：加速模式配置项
- [selinux]：selinux相关的配置项
- [colors]：ansible命令输出的颜色相关的配置项
- [diff]：定义是否在运行时打印diff（变更前与变更后的差异）

配置参数说明

```
[default]
inventory      = /etc/ansible/hosts
remote_user    = root
ask_pass       = false
log_path       = /var/log/ansible.log

[privilege_escalation]
become=True
become_method=sudo
become_user=root
become_ask_pass=False

[ssh_connection]
ssh_args = -C -o ControlMaster=auto -o ControlPersist=60s
host_key_checking = False
```

配置项说明：

- inventory: 定义默认使用的主机清单
- remote_user: ansible在操作远程主机时, 使用远程主机上的哪个用户身份, 默认是root
- ask_pass: ansible在操作远程主机时, 获取远程主机上的用户身份, 是否交互提示密码验证, 默认为true。如果使用密钥认证的话, 建议将其设置为false
- log_path: 默认ansible 执行的时候, 并不会输出日志到文件, 打开该配置项, 所有的命令执行后, 都会将日志输出到 /var/log/ansible.log 文件。
- become: 如果ansible在操作远程主机时, 使用的是远程主机上的普通用户, 该普通用户是否允许提权
- become_method: 如果允许提权, 使用何种提权方式, 默认是sudo
- become_user: 提权到哪个用户身份, 默认是root
- become_ask_pass: 提权时, 是否交互提示密码验证, 默认为False
- ssh_args: ansible通过ssh连接远程被管理机, 这里用于定义一些ssh连接时的参数, 如-C启用压缩传输, ControlPersist用于提升性能。
- host_key_checking: 通过ssh首次连接远程主机时, 由于在本机的 ~/.ssh/known_hosts 文件中并有 fingerprint key 串, ssh第一次连接的时候一般会提示输入yes/no进行确认将key字符串加入到 ~/.ssh/known_hosts 文件中。将此项设置为False将跳过该确认过程。

关于ssh连接一些常见的错误说明

1. ERROR! to use the 'ssh' connection type with passwords, you must install the sshpass program

完整错误示例如下：

```
root@ctnr:/etc/ansible# ansible '*.a32-168-1.*' -m ping
ctnr.a32-168-1.prod.yiz | FAILED! => {
  "failed": true,
  "msg": "ERROR! to use the 'ssh' connection type with passwords, you must install the
sshpass program"
}
```

一般出现这种错误，是在通过密码验证远程被管理机的时候，需要在server端安装sshpass：

```
yum install sshpass -y
```

2. Using a SSH password instead of a key is not possible because Host Key checking is enabled and sshpass does not support this. Please add this host's fingerprint to your known_hosts file to manage this host

完整错误如下：

```
ansible test -a 'uptime'

192.168.1.1 | FAILED => Using a SSH password instead of a key is not possible because
HostKey checking is enabled and sshpass does not support this. Please add this host's
fingerprint to your known_hosts file to manage this host.
192.168.1.2 | FAILED => Using a SSH password instead of a key is not possible because
Host Key checking is enabled and sshpass does not support this. Please add this host's
fingerprint to your known_hosts file to manage this host.
```

这种错误通常就出现在server端第一次连接被管理机的时候，就是上面说到的需要通过输入yes/no进行确认将key字符串加入到 ~/.ssh/known_hosts 文件中。

解决办法有两个：

- 通过修改上面提到的host_key_cheking，将其设置为false（在实际测试中，似乎并没有效果）
- 通过修改ssh_args参数，修改如下：

```
ssh_args = -C -o ControlMaster=auto -o ControlPersist=60s -o StrictHostKeyChecking=no
```