



链滴

# Ansible 从入门到实战 (2) - Ansible 初体验

作者: [Smiteli](#)

原文链接: <https://ld246.com/article/1567408262199>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# Ansible从入门到实战 (2) -Ansible初体验

Ansible是一款基于python的自动化运维工具，既然是自动化运维工具，那么就是用来管理其他服务的。上一节中，我们已经在主控节点安装好了Ansible，那么我们还需要一台被管理的服务器。下面学习本节的要求：

- 安装了Ansible的主控节点，上一节你已经安装好了
- 被管理服务器，可以再新建一台Centos7或者其他Linux系统的虚拟机

## 一、Ansible自动化运维原理

- Ansible是通过SSH协议与被管理的节点通信的，过程是怎样的呢？

下面为官方的回答：

默认情况下，Ansible将尽可能使用本机OpenSSH进行远程通信。这样可以启用ControlPersist（性功能），Kerberos和~/.ssh/config跳转主机设置等选项。但是，当使用Enterprise Linux 6操作系统为控制机器（Red Hat Enterprise Linux和CentOS等衍生产品）时，OpenSSH的版本可能太旧而无支持ControlPersist。在这些操作系统上，Ansible将回归使用名为'paramiko'的OpenSSH的高质量Python实现。如果您希望使用Kerberized SSH等功能，请考虑使用Fedora，macOS或Ubuntu作为控制计算机，直到您的平台可以使用较新版本的OpenSSH。

有时您会遇到不支持SFTP的设备。这种情况很少见，但如果发生这种情况，您可以在配置Ansible中换到SCP模式。

与远程计算机通话时，Ansible默认假设您使用的是SSH密钥。鼓励使用SSH密钥，但也可以通过提选项在需要时使用密码验证--ask-pass。如果使用sudo功能，当sudo需要密码时，也提供--ask-become-pass（之前--ask-sudo-pass已弃用）。

## 二、配置被管理的主机清单

如你所想，既然Ansible用来管理众多的服务器，那么肯定会有一个文件记录着那些被管理的主机。文件/etc/ansible/hosts就是默认的记录着被管理的主机清单文件。

cat /etc/ansible/hosts，我们可以看到以下内容：

```
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers.

## green.example.com
## blue.example.com
## 192.168.100.1
```

```
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern you can specify
# them like this:

## www[001:006].example.com

# Ex 3: A collection of database servers in the 'dbservers' group

## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

# Here's another example of host ranges, this time there are no
# leading 0s:

## db-[99:101]-node.example.com
```

下面我们编辑`/etc/ansible/hosts`文件，在文本最后添加如下内容：

```
[node1]
apple ansible_host=192.168.153.129 ansible_user=root ansible_password=root
```

上面的配置中：

- node1 是主机组名，我这里只配置了192.168.153.129这台主机
- apple 是主机名称，根据实际情况起一个简单易记的名字，也可以填写IP
- ansible\_host=192.168.153.129 ， 配置主机的IP地址。如果DNS能解析的话，也可以配置hostname,
- ansible\_user=root ， Ansible访问时使用的用户
- ansible\_password=root， Ansible访问时使用的用户的密码

### 三、Ansible初体验

- 3.1 现在ping所有节点： `$ ansible all -m ping`

我的主控节点执行上面的命令后得到的结果如下：

```
[root@localhost ~]# ansible all -m ping
apple | SUCCESS => {
  "ansible_facts": {
```

```
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
```

#### 四、配置免密登陆

在第三节中，我们在`/etc/ansible/hosts`文件中配置主机清单时也配置了用户名和密码，在生产环境，这是十分不安全的。正如第二节，官方说到的，鼓励使用SSH密钥进行通信，那么我们就配置好主节点和被管理节点的SSH免密登陆，那样就无需再配置用户名和密码了。

#### 配置免密登陆参考链接

根据上面的教程配置好了免密登陆后，`/etc/ansible/hosts`的配置修改为下面的样子：

```
[node1]
apple ansible_host=192.168.153.129
#apple ansible_host=192.168.153.129 ansible_user=root ansible_password=root
```

执行以下命令验证：`ansible all -a "/bin/echo hello"`

我的机器得到结果如下：

```
[root@localhost ~]# ansible all -a "/bin/echo hello"
apple | CHANGED | rc=0 >>
hello
```

可以看到已经成功了。

**注意：Ansible会尝试使用您当前的用户名远程连接到计算机，就像SSH一样。要覆盖远程用户名，需使用“-u”参数。**

**如果您想访问sudo模式，还有一些标志可以做到这一点：**

```
# as bruce
$ ansible all -m ping -u bruce
# as bruce, sudoing to root (sudo is default method)
$ ansible all -m ping -u bruce --become
# as bruce, sudoing to batman
$ ansible all -m ping -u bruce --become --become-user batman
```

#### 五、一个有用的提示

运行命令时，可以使用“localhost”或“127.0.0.1”指定本地服务器作为服务器名称。

例：

```
$ ansible localhost -m ping -e 'ansible_python_interpreter="/usr/bin/env python"'
```

我的服务器执行结果如下：

```
[root@localhost ~]# ansible localhost -m ping -e 'ansible_python_interpreter="/usr/bin/env p
thon"'
localhost | SUCCESS => {
```

```
"changed": false,  
"ping": "pong"  
}
```

当然，我们也可以配置`/etc/ansible/hosts`文件来明确指定localhost。在`/etc/ansible/hosts`文件最添加以下配置：

```
localhost ansible_connection=local ansible_python_interpreter="/usr/bin/env python"
```

再次执行`ansible all -a "/bin/echo hello"`命令，结果如下：

```
[root@localhost ~]# ansible all -a "/bin/echo hello"  
localhost | CHANGED | rc=0 >>  
hello
```

```
apple | CHANGED | rc=0 >>  
hello
```

可见配置已经生效。

## 六、下一节是？

你或许对上面执行的命令还不是很理解，不过没有关系。我提倡的是在实践中学习，在学习中总结，面的教程会——答疑的，别急，跟着教程一步一步来就好。