

Ansible 基础



千易雲(北京)教育科技有限公司

安装

- 需要安装些什么

Ansible默认通过 SSH 协议管理机器。

安装Ansible之后，不需要启动或运行一个后台进程,或是添加一个数据库.只要在一台电脑(可以是一台笔记本)上安装好,就可以通过这台电脑管理一组远程的机器。在远程被管理的机器上,不需要安装运行任何软件，因此升级Ansible版本不会有太多问题。

安装

- 选择哪一个版本？

因为**Ansible**可以很简单的从源码运行,且不必在远程被管理机器上安装任何软件，很多**Ansible**用户会跟进使用开发版本。

Ansible一般每两个月出一个发行版本。小**bugs**一般在下一个发行版本中修复，并在稳定分支中做**backports**。大**bugs**会在必要时出一个维护版本，不过这不是很频繁。

若你希望使用**Ansible**的最新版本，并且你使用的操作系统是 **Red Hat Enterprise Linux (TM)**, **CentOS**, **Fedora**, **Debian**, **Ubuntu**，我们建议使用系统的软件包管理器。

另有一种选择是通过“**pip**”工具安装，“**pip**”是一个安装和管理**Python**包的工具。

若你希望跟进开发版本，想使用和测试最新的功能特性,我们会分享如何从源码运行**Ansible**的方法.从源码运行程序不需要进行软件安装。

安装

- 对管理主机的要求

目前，只要机器上安装了 Python 2.6 或 Python 2.7 （**windows**系统不可以做控制主机），都可以运行Ansible。主机的系统可以是 Red Hat， Debian， CentOS， OS X， BSD的各种版本，等等。

自2.0版本开始，**ansible**使用了更多句柄来管理它的子进程，对于**OS X**系统，你需要增加**ulimit**值才能使用**15**个以上子进程，方法 **sudo launchctl limit maxfiles 1024 2048**，否则你可能会看见 “Too many open file” 的错误提示。

安装

- 对托管节点的要求

通常我们使用 `ssh` 与托管节点通信，默认使用 `sftp`。如果 `sftp` 不可用，可在 `ansible.cfg` 配置文件中配置成 `scp` 的方式。在托管节点上也需要安装 Python 2.4 或以上的版本。如果版本低于 Python 2.5，还需要额外安装一个模块：

`python-simplejson`

安装

- 通过Yum安装最新发布版本

通过Yum安装RPMs适用于 EPEL 6, 7, 以及仍在支持中的Fedora发行版. 托管节点的操作系统版本可以是更早的版本(如 EL5), 但必须安装 Python 2.4 或更高版本的Python。

Fedora 用户可直接安装Ansible, 但RHEL或CentOS用户, 需要 配置 EPEL
install the epel-release RPM if needed on CentOS, RHEL, or Scientific Linux:

```
$ sudo yum install ansible
```

你也可以自己创建RPM软件包。在Ansible项目的checkout的根目录下, 或是在一个tarball中, 使用 `make rpm` 命令创建RPM软件包.。然后可分发这个软件包或是使用它来安装Ansible.在创建之前, 先确定你已安装了 `rpm-build`, `make`, and `python2-devel` 。

```
$ git clone git://github.com/ansible/ansible.git
```

```
$ cd ./ansible
```

```
$ make rpm
```

```
$ sudo rpm -Uvh ~/rpmbuild/ansible-*.noarch.rpm
```


安装

- 通过**Apt (Ubuntu)**安装最新发布版本

Ubuntu 编译版可在PPA中获得:

配置PPA及安装ansible, 执行如下命令:

```
$ sudo apt-get install software-properties-common  
$ sudo apt-add-repository ppa:ansible/ansible  
$ sudo apt-get update  
$ sudo apt-get install ansible
```

安装

- 通过 **Portage (Gentoo)** 安装最新发布版本

```
$ emerge -av app-admin/ansible
```

要安装最新版本，你可能需要在执行 **emerge** 之前，先做如下操作
(unmask ansible)

```
$ echo 'app-admin/ansible'  
>>/etc/portage/package.accept_keywords
```

若在 **Gentoo** 托管节点中，已经安装了 **Python 3** 并将之作为默认的 **Python slot**（这也是默认设置），则你必须在 **组变量** 或 **inventory** 变量中设置如下变量

```
ansible_python_interpreter = /usr/bin/python2
```


安装

- 通过 **pkg (FreeBSD)** 安装最新发布版本

```
$ sudo pkg install ansible
```

你或许想从ports中安装:

```
$ sudo make -C /usr/ports/sysutils/ansible  
install
```

新手上路

- 你的第一条命令

现在你已经安装了Ansible,是时候从一些基本知识开始了. 编辑（或创建）`/etc/ansible/hosts` 并在其中加入一个或多个远程系统.你的public SSH key必须在这些系统的`authorized_keys`中:

```
192.168.1.50  
aserver.example.org  
bserver.example.org
```

这里有个节点设置文件(inventory file)将会在 *Inventory* 文件中得到深入说明. 我们假定你使用SSH Key来授权.为了避免在建立SSH连接时,重复输入密码你可以这么做:

```
$ ssh-agent bash  
$ ssh-add ~/.ssh/id_rsa
```

（根据你的建立方式,你也许希望使用Ansible的 `--private-key` 选项,通过指定pem文件来代替SSH Key来授权）现在ping 你的所有节点:

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

新手上路

- 你的第一条命令

Ansible会像**SSH**那样试图用你的当前用户名来连接你的远程机器.要覆写远程用户名只需使用'-u'参数.如果你想访问 **sudo**模式,这里也有标识(flags)来实现:

```
$ ansible all -m ping -u bruce
```

```
$ ansible all -m ping -u bruce --sudo
```

```
$ ansible all -m ping -u bruce --sudo --sudo-user batman
```

(如果你碰巧想要使用其他**sudo**的实现方式,你可以通过修改**Ansible**的配置文件来实现.也可以通过传递标识给**sudo**(如-H)来设置.) 现在对你的所有节点运行一个命令:

```
$ ansible all -a "/bin/echo hello"
```

恭喜你! 你刚刚通过**Ansible**连接了你的所有节点。

新手上路

- 公钥认证

Ansible1.2.1及其之后的版本都会默认启用公钥认证。如果有个主机重新安装并在“**known_hosts**”中有了不同的**key**,这会提示一个错误信息直到被纠正为止.在使用**Ansible**时,你可能不想遇到这样的情况:如果有个主机没有在“**known_hosts**”中被初始化将会导致在交互使用**Ansible**或定时执行**Ansible**时对**key**信息的确认提示.如果你想禁用此项行为并明白其含义,你能够通过编辑 `/etc/ansible/ansible.cfg` or `~/.ansible.cfg`来实现:

```
[defaults]  
host_key_checking = False
```

新手上路

- 公钥认证

Ansible 1.2.1 及其之后的版本都会默认启用公钥认证。如果有个主机重新安装并“known_hosts”中有了不同的key,这会提示一个错误信息直到被纠正为止.在使用 Ansible 时, 你可能不想遇到这样的情况:如果有个主机没有在“known_hosts”中被初始化将会导致在交互使用 Ansible 或定时执行 Ansible 时对 key 信息的确认提示。

如果你想禁用此项行为并明白其含义,你能够通过编辑 `/etc/ansible/ansible.cfg` or `~/.ansible.cfg` 来实现:

```
[defaults]
host_key_checking = False
```

或者你也可以通过设置环境变量来实现:

```
$ export ANSIBLE_HOST_KEY_CHECKING=False
```

同样注意在 paramiko 模式中 公钥认证 相当的慢。因此,当使用这项特性时, 切换至 ‘SSH’ ‘是推荐做法。

Ansible 将会对远程系统模块参数记录在远程的 syslog 中,除非一个任务或者 play 被标记了 “no_log: True” 属性,稍后解释. 在主机上启用基本的日志功能参见 *Ansible 的配置* 文件文档 并在配置文件中设置 ‘log_path’。企业用户可能也对 *Ansible Tower* 感兴趣。

塔提供了非常实用数据库日志.它使一次次向下钻取并查看基于主机, 项目, 和特定的结果集成为可能——同时提供了图形和 RESTful API。

谢谢，待续