

Ansible 模块



千易雲(北京)教育科技有限公司

简介

模块(也被称为“task plugins”或“library plugins”)是在 Ansible 中实际在执行的.它们就 是在每个 playbook 任务中被执行的.你也可以仅仅通过 ‘ansible’ 命令来运行它们。

让我们回顾一下我们是如何通过命令行来执行三个不同的模块:

```
ansible webservers -m service -a "name=httpd state=started"
```

```
ansible webservers -m ping
```

```
ansible webservers -m command -a "/sbin/reboot -t now"
```

每个模块都能接收参数。几乎所有的模块都接受键值对(key=value)参数，空格分隔。一些模块不接收参数，只需在命令行输入相关的命令就能调用。

核心模块

这些模块是 **ansible** 团队维护的核心模块，同样也是**ansible**自带的模块，在收到的请求中，它们有比 “**extras**” 源更高的优先级核心模块的源码托管在 Github 的 [ansible-modules-core](https://github.com/ansible-modules-core) repo。

如果你确信你在核心模块上发现了一个 **bug**，同时你使用的是最新的稳定版或者开发版本的 **Ansible**，首先你需要看[github.com/ansible/ansible-modules-core](https://github.com/ansible-modules-core)上的 “**issue tracker**”，确保你的 **bug** 还没有被提交。如果还没被提交，非常感谢你的提交你可能更想去问问题，而不是提交**bug**，欢迎在**Ansible-project**的**google-group**里咨询

<https://groups.google.com/forum/#!forum/ansible-project>，**Ansible**的 **#ansible_chanel**咨询也是可以的，它们位于[irc.freenode.net](https://freenode.net)。开发方向的应该用类似的讨论组，位于

<https://groups.google.com/forum/#!forum/ansible-devel>

这些模块的文档更新可以直接编辑在模块自身里面,然后提交一个**pull**请求到源码，不过你需要找到源码树的 “**DOCUMENTATION**” 块。

额外模块

这些模块是当前**ansible**附带的，但是也可能在以后被分开.额外模块主要被社区人员维护。非核心模块仍然是完全可用的，但是在发出和拉取请求时可能收到稍微低的响应速率受欢迎的“**extras**”模块将来可能会提升为核心模块这些额外的模块托管在Github上的**ansible-modules-extras** <http://github.com/ansible/ansible-modules-extras> repo。

如果你确信你在额外模块上发现了一个 **bug**，同时你使用的是最新的稳定版或者开发版本的 **Ansible** ,首先你需要看github.com/ansible/ansible-modules-extras上的“**issue tracker**”，确保你的**bug**还没有被提交。如果还没被提交，非常感谢你的提交你可能更想去问问题,而不是提交 **bug**，欢迎在 **Ansible-project** 的 **google-group**里咨询

<https://groups.google.com/forum/#!forum/ansible-project>，在 **Ansible**的 **#ansible_chanel**咨询也是可以的,它们位于 **irc.freenode.net** .开发方向的应该用类似的讨论组，位于

<https://groups.google.com/forum/#!forum/ansible-devel>这些模块的文档更新可以直接编辑在模块自身里面，然后提交一个**pull**请求到源码，不过你需要找到源码树的“**DOCUMENTATION**”块。

模块相关命令

- 列出当前环境中安装的所有模块：
`ansible-doc -l`
- 获得指定模块的帮助文档：
`ansible-doc -s <module_name>`
- 调用指定模块：
`ansible -m <module_name>`

常用模块简介

ping -- 尝试连接主机，验证一个可用的python和返回一个成功的pong

command -- 在远程节点上执行一个命令

raw -- 在远程节点上执行一个命令（可支持管道、通配符等脏命令）

shell -- 执行远程主机的shell/python脚本

script -- 在远程主机执行主控端的shell/python脚本

copy -- 将文件复制到远程位置

user -- 用户账户管理

group -- 用户组管理

filesystem -- 在块设备上创建文件系统

cron -- 管理cron.d和crontab条目

firewalld -- 使用firewalld管理任意端口/服务

iptables -- 修改系统iptables

parted -- 管理块设备分区

selinux -- 更改SELinux的策略和状态

sysctl -- 管理sysctl.conf中的条目

常用模块简介（续）

yum -- 通过yum管理软件包（RHEL、CentOS、Fedora）

yum-repository -- 添加或删除YUM仓库（RHEL、CentOS、Fedora）

rpm_key -- 从rpm数据库中添加或删除一个gpg key（RHEL、CentOS、Fedora、SuSE）

apt -- 通过apt管理软件包（Debian、Ubuntu）

apt_repository -- 添加或删除APT仓库（Debian、Ubuntu）

apt_key -- 添加或删除一个apt key（Debian、Ubuntu）

dpkg_selections -- dpkg包选择（Debian、Ubuntu）

apt_rpm -- apt和rpm包管理（混合环境）

zypper -- 通过zypper管理软件包（SuSE）

zypper_repository -- 添加或删除zypper仓库（SuSE）

pip -- 管理Python库依赖关系

easy_install -- 安装Python库

常用模块简介（续）

mysql_db -- 从远程主机中添加或删除MySQL数据库
mysql_replication -- 管理MySQL复制
mysql_user -- 从MySQL数据库中添加或删除一个用户
mysql_variables -- 管理MySQL全局变量
mongodb_paramete -- 更改MongoDB服务器上的管理参数
mongodb_user -- 从MongoDB数据库中添加或删除一个用户
rabbitmq_bindin -- 管理rabbitMQ绑定
rabbitmq_exchang -- 管理rabbitMQ交换
rabbitmq_parameter -- 添加或删除RabbitMQ的参数
rabbitmq_plugin -- 管理RabbitMQ插件
rabbitmq_policy -- 管理RabbitMQ中的策略状态
rabbitmq_queue -- 管理rabbitMQ队列
rabbitmq_user -- 从rabbitMQ服务器上添加或删除用户
rabbitmq_vhost -- 管理RabbitMQ中的虚拟主机的状态

更多模块

Ansible还有很多的实用模块，由于篇幅问题，无法一一介绍，详情可以参考以下官网地址：

http://docs.ansible.com/ansible/latest/modules_by_category.html

谢谢，待续