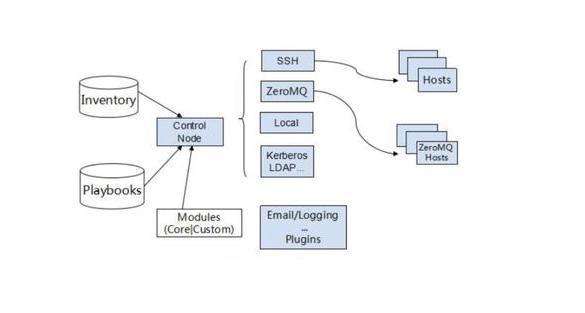
我们知道现在自动化工具中，最简单、易于上手，而且最流行的当属Ansible莫属了。和Chef、Puppet等Ruby工具甚至同时Python系的Saltstack等CS架构的自动化工具相比虽然执行性能上可能会稍一点，但是无需客户端，只需SSH就可以管理优势还是相当明显的。服务器上架后无需额外操作就可以直接进行操作，比如服务器的初始化标准配置等。本文虫虫给大家介绍就是Ansible入门教程，如果此前已有对应的基础可以忽略本文。



**概述**

Ansible是一个开源配置管理工具，可以使用它来自动化任务，部署应用程序实现IT基础架构。Ansible可以用来自动化日常任务，比如，服务器的初始化配置、安全基线配置、更新和打补丁系统，安装软件包等。Ansible架构相对比较简单，仅需通过SSH连接客户机执行任务即可：



Ansible使用过程中会用到一些概念术语，我们先介绍一下。

Ansible的与节点有关的重要术语包括控制节点，受管节点，清单和主机文件：

控制节点（Control node）：指安装了Ansible的主机，也叫Ansible服务器端,管理机。 Ansible控制节点主要用于发布运行任务，执行控制命令。Ansible的程序都安装在控制节点上，控制节点需要安装Python和Ansible所需的各种依赖库。注意：目前Ansible还不能安装在Windows下。

受控节点（Managed nodes）：也叫客户机，就是想用Ansible执行任务的客户服务器。

清单（Inventory）：受控节点的列表，就是所有要管理的主机列表。

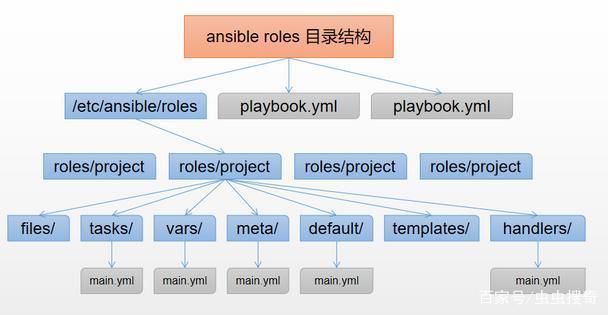
host文件：清单列表通常保存在一个名为host文件中。在host文件中，可以使用IP地址或者主机名来表示具体的管理主机和认证信息，并可以根据主机的用户进行分组。缺省文件：/etc/ansible/hosts，可以通过-i指定自定义的host文件。

模块（Modules）：模块是Ansible执行特定任务的代码块。比如：添加用户，上传文件和对客户机执行ping操作等。Ansible现在默认自带450多个模块，，Ansible Galaxy公共存储库则包含大约1600个模块。

任务（Task）：是Ansible客户机上执行的操作。可以使用ad-hoc单行命令执行一个任务。

剧本(Playbook):是利用YAML标记语言编写的可重复执行的任务的列表，playbook实现任务的更便捷的读写和贡献。比如，在Github上有大量的Ansible playbooks共享，你要你有一双善于发现的眼睛你就能找到大量的宝藏。

角色（roles）：角色是Ansible 1.2版本引入的新特性，用于层次性、结构化地组织playbook。roles能够根据层次型结构自动装载变量文件、tasks以及handlers等。



**Ansible的优势**

Ansible作为最受欢迎的自动化配置工具，主要得益于其设计上的优势。

**无需客户端**

与Chef、Puppet以及Saltstack（现在也支持Agentless方式salt-ssh）不同，Ansible是无客户端Agent的，所以无需在客户机上安装或配置任何程序，就可以运行Ansible任务。由于Ansible不会在客户机上安装任何软件或运行监听程序，因此消除了许多管理开销，我们可以在即可上手使用Ansible管理服务器，同时Ansible的更新也不会影响任何客户机。

**使用SSH进行通讯**

默认情况下，Ansible使用SSH协议在管理机和客户机之间进行通信。可以使用SFTP与客户机进行安全的文件传输。

**并行执行**

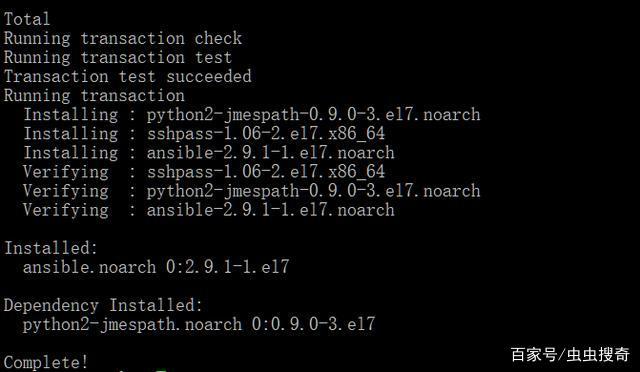
Ansible与客户机并行通信，可以更快地运行自动化任务。默认情况下，forks值为5，可以按需，在配置文件中增大该值。

**安装Ansible**

Ansible可以运行在任何机器上，但是对管理机有一定要求。管理机应安装Python 2（2.7）或Python 3（3.5或更高版本），另外，管理机不支持Windows控制节点。我们可以使用Linux发行版包管理器、源码安装或者Python包管理器（PIP）来安装Ansible。

**redhat系**

sudo yum install ansible 或者dnf install ansible



**ubuntu**

sudo apt update

sudo apt install software-properties-common

sudo apt-add-repository --yes --update ppa:ansible/ansible

sudo apt install ansible

**通过Pip安装**

pip install --user ansible

或者全局性安装

sudo pip install ansible

**源码安装**

git clone github.com/ansible/ansible.git

cd ./ansible

source ./hacking/env-setup

安装成功后，可以使用下面命令检查Ansible的安装版本：

ansible –version

ansible 2.9.1

config file = /etc/ansible/ansible.cfg

configured module search path = [u'/home/xx/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']

ansible python module location = /usr/lib/python2.7/site-packages/ansible

executable location = /usr/bin/ansible

python version = 2.7.5 (default, Oct 30 2018, 23:45:53) [GCC 4.8.5 20150623 (Red Hat 4.8.5-36)]

**安全配置**

为了使Ansible与客户端通信，需要使用用户帐户配置管理机和客户机。为了方便快捷安全，一般会配置证书方式连接客户机。

**帐号规划**

假设我们有四台客户机（node1，node2，node3和node4）和管理机（manage）。为了方便，我们创建一个ansible用户帐户，将该ansible用户添加到wheel组，然后配置SSH身份验证。在配置新用户帐户时，请在所有节点上创建该帐户（注意可以使用ansible批量创建）：

sudo useradd ansible

然后将ansible用户添加到wheel组：

sudo usermod -aG wheel ansible

为用户设置密码：

sudo passwd ansible

接下来，使用visudo配置/etc/sudoers文件,赋予ansible用户使用sudo执行特权命令：

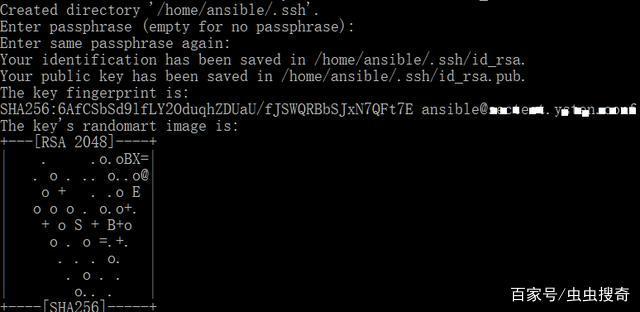
%wheel ALL=(ALL) NOPASSWD: ALL

**配置证书登陆**

在所有客户机和管理上创建新的ansible用户之后，我们在管理机（ansible用户）生成SSH密钥，然后将SSH公钥复制到所有客户机。

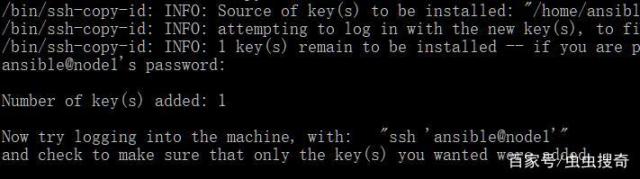
sudo su - ansible

ssh-keygen



现在，将SSH公钥复制到所有客户机，这使管理机ansible用户无需输入密码即可登录客户机：

ssh-copy-id ansible@node1



第一次使用，完成以后，就可以直接用证书登陆了。

**Ansible配置**

默认的配置文件位于/etc/ansible/ansible.cfg下。可以使用此配置文件来修改绝Ansible大多数设置，一般无需额外多配置，默认配置应能满足大多数使用情况。关于Ansible配置文件，其执行程序会按照一定顺序搜索配置：

Ansible按照以下顺序搜索配置文件，优先配置优先使用，而忽略其余配置文件：

$ANSIBLE\_CONFIG环境变量。

任务当前目录下的：ansible.cfg（如果在当前目录中）。

当前用户下的ansible.cfg：~/.ansible.cfg

默认配置文件：/etc/ansible/ansible.cfg。

默认清单配置文件位于/etc/ansible/hosts中，但是通过ansible.cfg配置文件中修改此位置。也可以通过-i自定义hosts清单。

为了安全起见，虫虫建议你，不要直接在/etc/ansible/host配置清单，尤其是有用户认账等信息时候。对于长期不执行ansible可以将host文件加密锁定，防止信息泄露，引起安全事故。

清单文件格式如下：

[nodes]

node1

node2

node3

node4

[web]

node2

node3

可以对不同用途分组，用[]指定分组名。

**Ansible ad-hoc单行命令执行**

ad-hoc命令行是我们可以随手执行的单个ansible任务，是ansible任务快速执行方式。对于一些快速获取的任务使用ad-doc命令非常简便有效，而且有助于我们学习和熟悉Ansible的使用。

**命令行选项**

常用的Ansible命令行选项如下：

-b，--become：特权方式运行命令。

-m：要使用的模块名称。

-a，--args：制定模块所需的参数。

-u：制定连接的用户名。

-h，--help显示帮助内容。

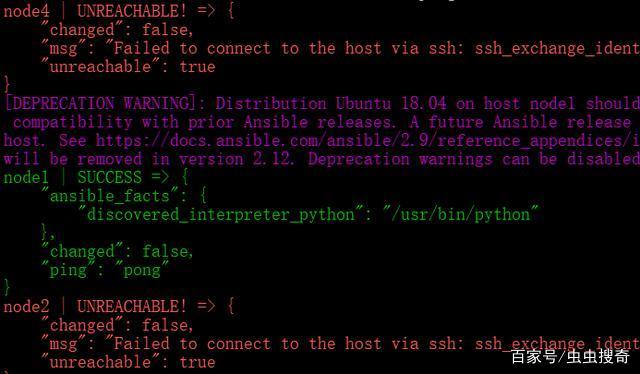
-v，--verbose以详细信息模式运行命令，可以用来调试错误。

更多选项和说明详见Ansible官方文档。下面我们介绍几个简单实例示范：

使用ping模块检查与客户机的连接性，请执行以下操作：

ansible all -m ping

在上面的命令中，all指定Ansible应该在所有主机上运行此命令，也可以按照分组比如nodes 或者主机 node1等。执行结果如下：



可见node1是通的，而node2和node4不通。

**使用ad-hoc命令管理软件包**

使用Ansible的ad-hoc命令，可以给客户端安装软件包。我们只需执行一个单行命令，然后实现安装。

比如：在分组客户机安装Apache只需要执行：

ansible web -m yum yum -a "name=httpd state=present" -b

**使用ad-hoc命令管理服务**

上面我们给web组的客户机批量安装了Apache服务器，下面我们来看怎么启动httpd服务。

启动httpd服务只需要执行：

ansible web -b -m service -a "name=httpd enabled=yes"

要启动，重新启动和停止该服务，只需state参数的值更改为started以启动服务，restarted以重新启动服务，stop来停止服务：

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

Ansible ad-hoc命令非常出色，适合于运行单个任务。具体任务模块可以参考官方文档。

**Ansible Playbook**



Playbook 是Ansible提供的最强大的任务执行方法。与ad-hoc命令不同，Playbooks配置在文件中，可以重用和共享给其他人。

playbooks是以YAML标记语言来定义的。每个playbook由一个或多个play组成。play的目标是将一组主机映射到任务中去。每个play包含一个或多个任务，这些任务每次执行一次。

下面是一个简单的Ansible httpd.yaml playbook，用于给web组上安装Apache服务器：

---

- hosts: web

remote\_user: ansible

tasks:

- name: Ensure apache is installed and updated

yum:

name: httpd

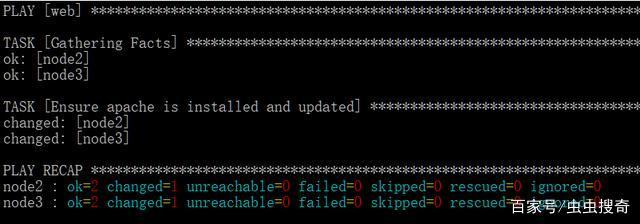
state: latest

become: yes

执行playbook使用ansible-playbook命令，格式如下：

ansible-playbook -i <hostfile> <playbook.yaml>

ansible-playbook httpd.yaml结果：



下面是一个综合多个任务的多个play的playbook，实现

安装apche服务器。

在Web服务器组中启用并启动httpd服务。

在所有客户机上安装git。

其内容如下：

---

- hosts: web

remote\_user: ansible

become: yes

tasks:

- name: Installing apache

yum:

name: httpd

state: latest

- name: Enabling httpd service

service:

name: httpd

enabled: yes

notify:

- name: restart httpd

handlers:

- name: restart httpd

service:

name: httpd

state: restarted

- hosts: all

remote\_user: ansible

become: yes

tasks:

- name: Installing git

yum:

name: git

state: latest

**总结**

Ansible是一个方便快捷又功能强大的自动化执行和配置管理工具。本文只是Ansible的入门教程，介绍了Ansible的基本概念、优势、安装，配置和简单的使用。师傅领进门修行靠自己，Ansible更强大的功能和实现自己基于它的全任务自动化需要自己使用探索。当然前面也提到了网络上尤其是Github上已经有大量Ansible的技术积累(ansible-role-xxxx)可供大家学习参考。