Ansible 自动化运维

Ansible 简介

Ansible 安装使用非常简单，而且基于上千个插件和模块，实现各种软件、平台、版本的管理，支持虚拟容器多层级的部署。

Ansible 自动化工具原理

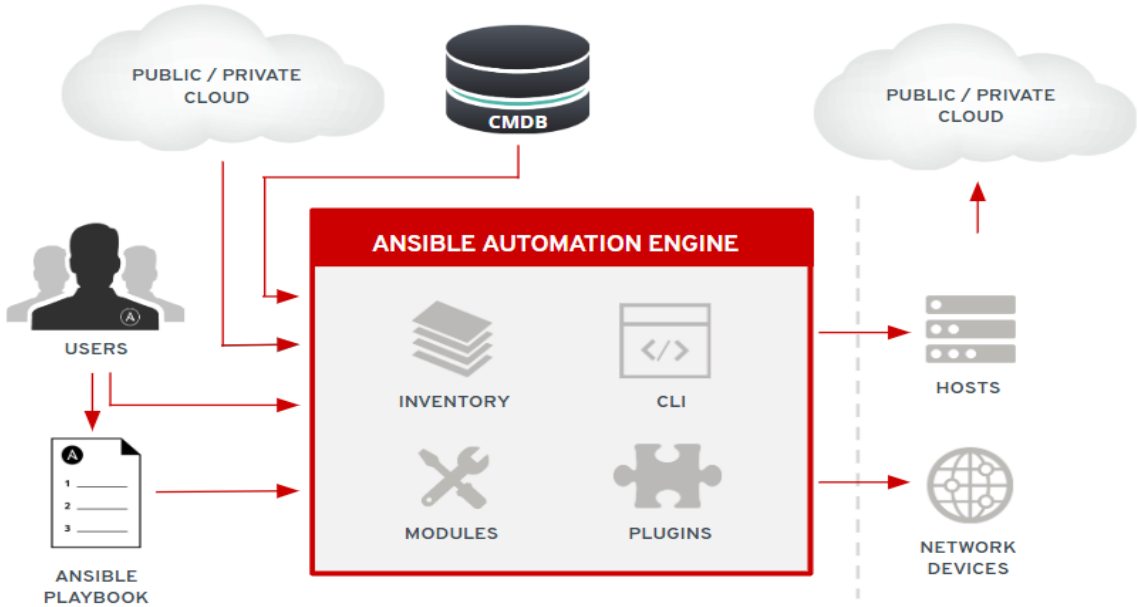
Ansible 是一款极为灵活的开源工具套件，能够大大简化 UNIX 管理元的自动化配置管理与流程控制方式，他利用推送方式对客户系统加以配置，这样所有工作都可在服务器端完成。命令行机制同样非常强大，允许大家利用商业许可 WEB UI 实现授权管理与配置，通过 ansible 进行管理的服务。

Ansible 与 2015 年被 Red Hat 公司以 1.5 亿美元收购；

Ansible 自动运维管理工具优点：

- 轻量级，更新时只需要在操作机上进行一次更新即可；
- 采用 SSH 协议；
- 不需要客户端安装 agent；
- 批量任务执行可以写成脚本，而且不用分发到远程客户端；
- 使用 python 编写，维护更简单；
- 支持 sudo 普通用户命令；
- 去中心化管理。

Ansible 自动化管理工具原理拓补图：



Ansible 管理工具安装配置

Ansible 可以工作在 Linux、 BSD、 Mac OS 、Windows 等平台，对 python 环境要求在 python2.6 以上；
Red Hat、CentOS 操作系统可以直接机遇 YUM 工具自动安装 ansible。再安装前需要先安装 epel 扩展源；

安装方法：（一）注意：需要连接网络

（1） 安装 epel 源

- wget -O /etc/yum.repos.d/epel.repo http://mirrors.aliyun.com/repo/epel-7.repo
- yum install -y https://mirrors.aliyun.com/epel/epel-release-latest-8.noarch.rpm
- 将 repo 配置中的地址替换为阿里云镜像站地址
  a) sed -i 's|^#baseurl=https://download.fedoraproject.org/pub|baseurl=https://mirrors.aliyun.com|' /etc/yum.repos.d/epel*
  b) sed -i 's|^metalink|#metalink|' /etc/yum.repos.d/epel*

安装 ansible

yum -y install ansible

安装方法：（二）基于没有本地安装（需提前准备安装包）

ansible 配置文件：

- /etc/ansible/ansible.cfg. //主配置文件
- /etc/ansible/hosts //hosts 文件为被管理机 IP 或者主机列表
- /etc/ansible/roles。 //为角色或者插件路径 默认该目录为空目录

Ansible 远程批量管理执行命令时通过 Ad-Hoc 来完成，也即点对点执行命令，能狗快速执行，而且不需要保存执行的命令，默认 hosts 文件配置主机列表，也可以配置分组。

```
# 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' gr

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

环境部署表

角色	主机名	IP	组名	OS version
master	automation	192.168.18.95		Redhat 8
node	Node	192.168.18.155	web	Centos 7

安装 epel 源:

automation~]#sed -i 's|^#baseurl=https://download.fedoraproject.org/pub|baseurl=https://mirrors.aliyun.com|' /etc/yum.repos.d/epel.repo
automation ~]# sed -i 's|^metalink|#metalink|' /etc/yum.repos.d/epel*

安装 ansible:

master 主机上安装 ansible
[root@automation ~]# dnf -y install ansible

Ansible 配置及测试:

配置主机的清单
[root@automation ~]# vim /etc/ansible/hosts

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

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10
192.168.18.155
```

通过 ping 模块测试主机的连通性，分别对单主机及组进行 ping 操作

```
[root@automation ~]# ansible node -m ping
node | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: root@node: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).",
    "unreachable": true
}
```

出现红色时报错：原因是因为由于主控端与被控端主机未配置 SSH 证书信任，需要免密码连接；

配置 Linux 主机 SSH 无密码访问

为了避免 ansible 发送指令时输入目标主机密码，通过证书签名达到 SSH 无密码是一个好的方案，推荐使用 ssh-keygen 与 ssh-copy-id 来实现快速证书的生成及公钥下发。

（1）在主控端用 ssh-keygen 命令申请私钥和公钥

```
[root@automation ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:j4VH9cX6QSGvpDtE+AtvqzZrbtKV6sqWD1x0wCMZBuQ root@automation
The key's randomart image is:
+---[RSA 2048]----+
|    .o.o+.  .. oo|
|    . .o oo. .oo.|
|     E  .ooo .oo |
|        .o+ o o. |
|        Soo+.. ..|
|      . .=+oo   .|
|       o+ +*     |
|      .+.B. o    |
|      .oX*+.     |
+-----[SHA256]-----+
```

（2） 同步公钥文件 id_rsa.pub 到目标主机，使用 ssh-copy-id 公钥拷贝工具

```
[root@automation ~]# ssh-copy-id root@192.168.18.155
```

（3） 再次测试出现 ping pong 说明 ping 通了

```
[root@automation ~]# ansible 192.168.18.155 -m ping
192.168.18.155 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
```

定义主机与组规则

Ansible 通过定义好的主机与组规则（inventory）对匹配的目标主机进行远程操作，配

置文件默认是/etc/ansible/hosts；两种方式定义主机和组规则的方式；

（1） vim /etc/ansible/hosts

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

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10
192.168.18.155
# 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
```

（2） 自定义 inventory 文件

```
[root@master opt]# cat inventory
node    ansible_host=192.168.18.155 ansible_port=22
node1   ansible_host=192.168.18.150 ansible_port=22
```

inventoty 文件选项

ansible_host	连接目标主机的地址	
ansible_port	连接目标主机 SSH 端口，端口默认无需指定	
ansible_user	连接目标主机默认用户	
ansible_pass	连接目标主机默认用户密码	
ansibale_connection	连接目标主机的连接类型	
ansible_private_key_file	连接目标主机的 SSH 密钥	
ansible_*_interpreter	指定采用非 python 的其他脚本语言*代表替换语言	

Ansible 模块

1、 ping 模块

a) 作用：

i. 判断远程客户端是否子在线

b) 示例；

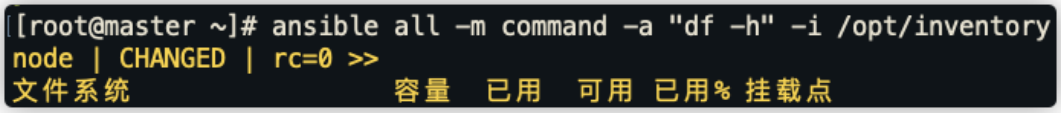
```
[root@master ~]# ansible all -m ping -i /opt/inventory
node | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
```

2、 command 模块

a) 作用：

Ansible command 模块为 ansible 默认模块，主要用于执行 Linux 基础命令，可以执行远程服务器命令执行、任务执行等操作。

b) 示例：



3、 copy 模块

a) 作用：

Ansible copy 模块主要用于文件或者目录复制，支持文件、目录、权限、用户组等；

copy 模块使用详解：

- src: ansible 端源文件或者目录，空文件夹不复制
- content：用来代替 src，用于指定文件的内容复制到远程文件内；
- dest：客户端目标目录或者文件，需要绝对路径；
- backup：复制之前，先备份远程节点上的原始文件；
- directory_mode：用于复制文件夹，新建的文件会被复制，而老旧的不会被复制；
- follow：支持 link 文件复制；
- force：覆盖远程主机不一致的内容；
- group：设定远程主机文件夹的组名；
- mode：指定远程主机文件夹的用户名；
- owner：设定远程主机文件夹的用户用；

b) 示例：

4、