**ansible基础**

ansible简介

什么是ansible

ansible是2013年推出的一款IT自动化和DevOps软件，目前由redhat已签署ansible收购协议。其基于python开发，揉合了很多老运维工具的优点实现了批量操作系统配置，批量程序的部署，批量运行命令等功能

ansible可以让我们实现

自动化部署APP

自动化管理配置项

自动化的持续交付

自动化的（AWS）云服务管理

为什么要选择anisble

选择一款配置管理软件总的来说，无外乎从以下几点来权衡利弊

活跃度（社区活跃度）

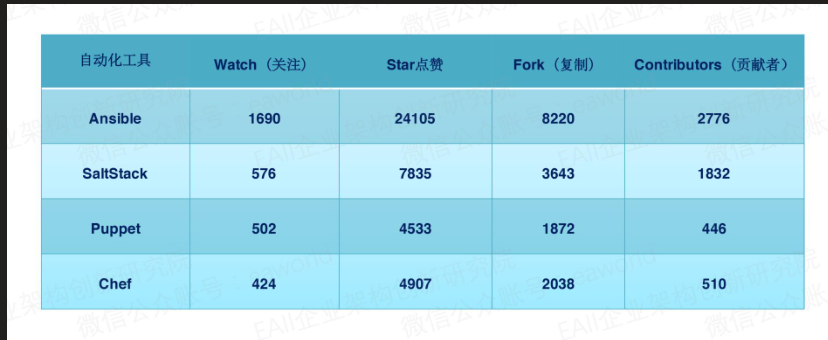
学习成本

使用成本

编码语言

性能

使用是否广泛



ansible优点

是仅需要ssh和python即可使用

无客户端

ansible功能强大，模块丰富

上手容易门槛低

基于python开发，做二次开发更容易

使用公司比较多，社区活跃

ansible缺点

对于几千台、上万台机器的操作，还不清楚性能、效率情况如何，需要进一步了解

ansible特性

模块化设计，调用特定的模块来完成特定的任务

基于python语言实现

paramiko

pyYAML（半结构化语言）

jinjia2

其模块支持JSON等标准输出格式，可采用任何编程语言重写

部署简单

主从模式工作

支持自定义模块

支持playbook

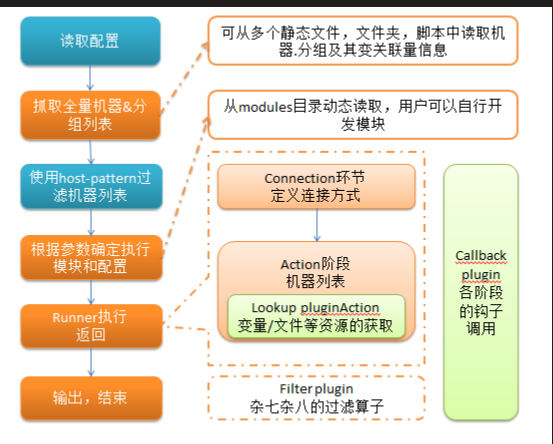
易于使用

支持多局部署

支持异构IT环境

工作流程

ansible大体执行过程



ansible安装

软件依赖关系

对管理主机的要求

要求python2.6或python2.7

ansible使用了以下模块，都需要安装

paramiko

PyYAML

jinja2

httplib2

six

对于被托管主机

ansible默认通过SSH协议管理主机

被管理主机要开启SSH服务，允许ansible主机登录

在托管主机上也需要安装python2.5或以上的版本

如果托管节点上开启了selinux，需要安装libselinux-python

安装ansible

ansible可以基于源码运行

源码安装

pip，需要配置扩展软件包源extras

git

yum -y install epel-release

yum -y install git python2-pip

pip安装依赖模块

pip install paramiko PyYAML Jinja2 httplib2 six

源码下载

ansible源码下载

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

yum install python-setuptools python-devel

python setup.py build

python setup.py install

pip方式安装

pip install ansible

安装ansible

yum扩展源安装简单，自动解决依赖关系（推荐）

<http://mirror.centos.org/../../extras/>

yum install ansible

安装完以后验证

ansible -version

**ad-hoc**

主机管理

主机定义与分组

安装好了ansible之后就可以开始一些简单的任务了

ansible配置文件查找顺序

首先检测ANSIBLE\_CONFIG变量定义的配置文件

其次检查当前目录下的./ansible.cfg文件

再次检查当前用户家目录下~/ansible.cfg文件

最后检查/etc/ansible/ansible.cfg文件

/etc/ansible/ansible.cfg默认配置文件路径

ansible.cfg配置文件

inventory是定义托管主机地址配置文件

首先编辑/etc/ansible/hosts文件，写入一些远程主机地址

格式

#表示注释

[组名称]

主机名称或ip地址，登录用户名，密码、端口等信息

测试

ansible [组名称] --list-hosts

inventory参数说明

ansible\_ssh\_host

将要连接的远程主机名。与想要设定的主机的别名不同的话，可用过此变量设置

ansible\_ssh\_port

ssh端口号.如果不是默认的端口号，通过此变量设置

ansible\_ssh\_user

默认的ssh用户名

ansible\_ssh\_pass

ssh密码（这种方式并不安全，强烈建议使用--ask-pass或SSH密码）

ansible\_sudo\_pass

sudo密码（建议使用 --ask-sudo-pass）

ansible\_sudo\_exe(new in version1.8)

sudo命令路径（适用于1.8及以上版本）

ansible\_connection

与主机连接的类型。比如:local,ssh或者paramiko。ansible1.2以前默认使用paramiko.1.2，以后默认使用’smart’,’smart’方式会提示是否支持controlpersist,来判断’ssh’方式

ansible\_ssh\_private\_key\_file

ssh使用的私钥文件。适用于有多个密钥，而不想使用SSH代理的情况

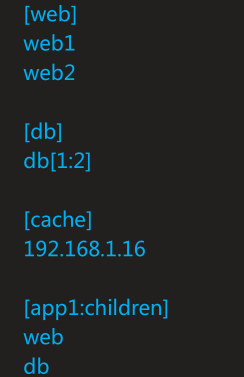
ansible\_sheel\_type

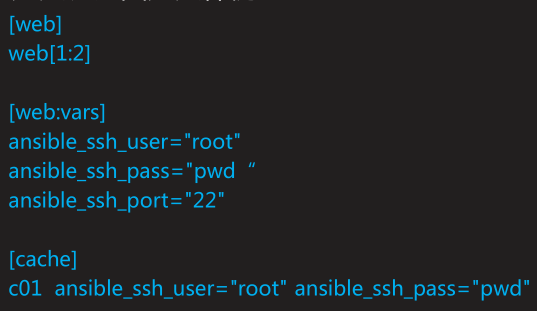
目标系统的shell类型.默认情况下，命令的执行使用’sh’语法，可设置为’csh’或’fish’;

ansible\_python\_interpreter

目标主机的python路径.适用于的情况：系统中有多个python，或者命令路径不是”/usr/bin/python”

分组定义、范围定义样例





自定义配置文件

创建文件夹myansible

创建配置ansible.cfg

[defaults]

inventory=myhost

配置主机文件

[nginx]

192.168.1.11

192.168.1.12

192.168.1.13

ansible nginx --list-hosts

动态主机

无限可能

ansible inventory实际上是包含静态inventory和动态inventory两部分，静态inventory指的是在文件/etc/ansible/hosts中指定的主机和组，dynamic inventory指通过外部脚本获取主机列表，并按照andible所要求的格式返回给ansible命令的

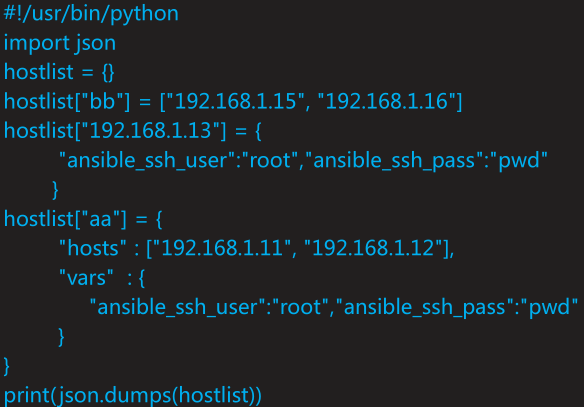
json

json的全称是”javascript object notation”,意思是javascript对象表示法，它是一种基于文本，独立于语言的轻量级数据交换格式

注意事项：

1. 主机部分必须是列表格式的；
2. hostdata行，其中”hosts”部分可以忽略，但如果使用时，必须是hosts

脚本输出主机列表



脚本输出样例



批量执行

ansible命令基础

ansible <host-parttern> [options]

host-pattern 主机或定义的分组

-M 指定模块路径

-m 使用模块，默认command模块

-a or --args模块参数

-i inventory文件路径，或可执行脚本

-k 使用交互式登录密码

-e 定义变量

-v详细信息，-vvvv开启debug模式

ansible all --list-hosts列出要执行的主机，不执行任何操作

ansible all -m ping批量检测主机

ansible all -m command -a ‘id’ -k

批量部署证书文件

每次交互输入密码比较麻烦

密码写入配置文件安全性很差

不同主机不同密码，配置文件要上天

使用key方式验证，是一个不错的选择

给所有主机部署公钥

ansible all -m authorized\_key -a “user=root exclusive=true manage\_dir=true key=’$(</root/.ssh/authorized\_keys)’” -k -v

**批量配置管理**

模块

ansible-doc

模块的手册，相当于shell的man

非常重要

ansible-doc -l列出所有模块

ansible-doc modulename 查看帮助

ping模块

测试网络连通性,ping模块没有参数

注：测试ssh的联通性

ansible host-pattern -m ping

command模块

默认模块，远程执行命令

用法

ansible host-pattern -n command -a ‘[args]’

查看所有机器负载

ansible all -m command -a ‘uptime’

查看日期和时间

ansible all -m command -a ‘date +%F\_%T’

注意事项：

该模块通过-a跟上要执行的命令可以直接执行，不过命令里如果有如下部分则执行不成功

< > | &

该模块不启动shell直接在ssh进程中执行，所有使用到sheell特性的命令都会失败

下列命令执行会失败

ansible all -m command -a ‘ps aux|grep ssh’

ansible all -m command -a ‘set’

shell | raw模块

shell模块用法基本和command一样，区别是shell模块是通过/bin/sh进行执行命令，可以执行任意命令

raw模块，用法和shell模块一样，可以执行任意命令

区别是raw没有cchdir、creates、removes参数

执行以下命令查看结果

ansible t1 -m command -a ‘chdir=/tmp touch f1’

ansible t1 -m shell -a ‘chdir=/tmp touch f2’

ansible t1 -m raw -a ‘chdir=/tmp touch f3’

script模块

复杂命令直接在本地写脚本，然后使用script模块批量执行

ansible t1 -m script -a ‘urscript’

友情提示：该脚本包含但不限于shell脚本，只要指定sha-bang解释器的脚本都可以运行

copy模块

复制文件到远程主机

src：要复制到远程主机的文件在本地的地址，可以是绝对路径，也可以是相对路径。如果路径是一个目录，它将递归复制。在这种情况下，如果路径使用“/”来结尾，则只复制目录里的内容，如果没有使用”/”来结尾，则包含目录在内的整个内容全部复制，类似于rsync

dest：必选项。远程主机的绝对路径，如果源文件是一个目录，那么该路径也必须是个目录

backup：在覆盖之前将原文件备份，备份文件包含时间信息。有两个选项：yes|no

force：如果目标主机包含该文件，但内容不同，如果设置为yes，则强制覆盖，如果为no，则只有当目标主机的目标位置不存在该文件时，才复制。默认为yes

复制文件

ansible t1 -m copy -a ‘src=/root/alog dest=/root/a.log’

复制目录

ansible t1 -m copy -a ‘src=urdir dest=/root/’

lineinfile | replace 模块

类似sed的一种行编辑替换模式

path目的文件

regexp正则表达式

line替换后的结果

ansible t1 -m lineinfile -a 'path="/etc/selinux/config"

regexp="^SELINUX=" line="SELINUX=disabled"'

替换指定的字符串

ansible t1 -m replace -a 'path="/etc/selinux/config"

regexp="^(SELINUX=).\*" replace="\1disabled"'

yum模块

使用yum包管理器来管理软件包

config\_file：yum的配置文件

disable\_gpg\_check：关闭gpg\_check

disablerepo：不启用某个源

enablerepo：启用某个源

name：要进行操作的软件包的名字，也可以传递一个url或者一个本地的rpm包的路径

state：状态（present，absent，latest）

删除软件包

ansible t1 -m yum -a 'name="lrzsz" state=absent'

删除多个软件包

ansible t1 -m yum -a 'name="lrzsz,lftp" state=absent'

安装软件包

ansible t1 -m yum -a 'name="lrzsz"'

安装多个软件包

ansible t1 -m yum -a 'name="lrzsz,lftp"'

service模块

name：必选项，服务名称

enabled：是否开机启动yes|no

sleep：如果执行了restarted，在则stop和start之间沉睡几秒钟

state：对当前服务执行启动，停止、重启、重新加载等操作（started，stopped,restarted,reloaded）

ansible t1 -m service -a 'name="sshd" enabled="yes" state="started"'

setup模块

主要用于获取主机信息，在playbooks里经常会用到的一个参数gather\_facts就与该模块相关。setup模块下经常使用的一个参数是filter参数

filter可以国旅到我们需要的信息

ansible t1 -m setup -a 'filter=ansible\_distribution'