**Morning：**

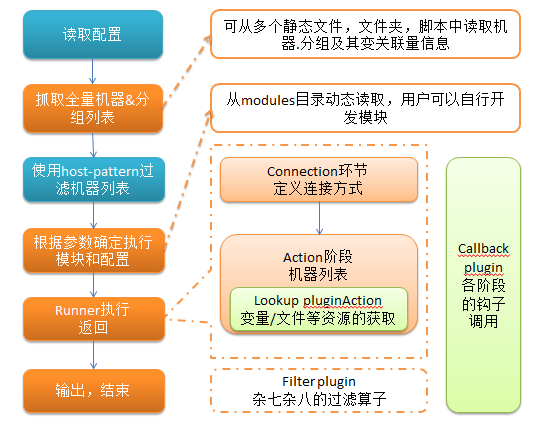
1、ansible简介

**-What’s ansible**

DevOps软件，基于Python开发。

可实现自动化部署APP、自动化管理配置项、自动化云服务管理等。

**-工作流程**



主机集合+命令集合

1. ansible安装

**步骤1：环境准备**

6台虚拟机：2G内存、2核CPU、20G硬盘

192.168.1.10 ansible #安装ansible

192.168.1.11 web1

192.168.1.12 web2

192.168.1.13 db1 #被托管

192.168.1.14 db2

192.168.1.15 cache

**步骤2：软件依赖关系**

管理主机：Python 2.6/2.7，需安装模块：paramiko、PyYAML、Jinja2、httplib2、six

被托管主机：Python 2.5以上，开启sshd

**步骤3：安装ansible**

1. 源码安装：

yum install epel-release git python2-pip #安装pip、git工具包

pip install paramiko PyYAML Jinja2 httplib2 six #pip安装依赖模块

git clone git://github.com/ansible/ansible.git #ansible源码下载

yum install python-setuptools python-devel #安装编译工具

python setup.py build #编译

python setup.py install #安装

pip install ansible #pip方式安装

2）自定义yum安装

[root@room12pc2 ansible]# ls ansible\_soft/

#包括依赖包： ansible、python2-jmespath、python-httplib2、 python-paramiko、python-passlib、sshpass

[root@room12pc2 ansible]# createrepo ansible\_soft

#自定义yum源

[root@ansible ~]# yum -y install ansible #安装ansible

[root@ansible ~]# ansible --version #查看版本

1. ad-hoc（使用ansible指令）

**-主机管理**

1. 配置文件查找顺序：

- ANSIBLE\_CONFIG变量定义的配置文件

- ./ansible.cfg

- ~/ansible.cfg

- /etc/ansible/ansible.cfg （默认）

1. 主机定义与分组

-ansible配置文件与inventory文件

[root@ansible ~]# vim /etc/ansible/ansible.cfg

[default]

inventory = /etc/ansible/hosts #定义托管主机地址配置文件

host\_key\_checking = false #不作ssh key检测

... ...

[root@ansible ~]# vim /etc/ansible/hosts

[web] #定义组名

web1 ansible\_ssh\_pass=”123456” #定义单个成员，配置个性参数

[db]

db[1..10] #定义范围成员

[db:vars] #为组db配置共性参数

ansible\_ssh\_pass=”123456”

[app:children] #使用app、db组作为成员

web

db

**inventory参数说明：**

|  |  |
| --- | --- |
| ansible\_ssh\_host | 要连接的进程主机名 |
| ansible\_ssh\_port | ssh端口号 |
| ansible\_ssh\_user | ssh 用户名 |
| ansible\_ssh\_pass | ssh 密码 |
| ansible\_sudo\_pass | sudo 密码 |
| ansible\_sudo\_exe | sudo 命令路径 |
| ansible\_connection | 与主机的连接类型 |
| ansible\_ssh\_private\_key\_file | ssh 使用的私钥文件 |
| ansible\_shell\_type | 目标系统的shell类型 |
| ansible\_python\_interpreter | 目标主机的 python 路径 |

3）动态主机

通过外部脚本获取主机列表,并按照ansible 所要求的格式(JSON)返回给ansilbe命令。

-将inventory指定为一个脚本文件

-脚本文件固定输出JSON格式的主机列表

**-批量执行**

1）基础命令的使用

命令格式：**ansible 主机或分组 参数 模块名 “模块的参数”**

-M 指定模块路径

-m 使用模块,默认 command 模块

-a或--args 模块参数

-i inventory文件路径,或可执行脚本

-k 使用交互式登彔密码

-e 定义变量

-v 详细信息 -vvvv 开启 debug 模式

[root@ansible ~]# ansible web|web,db|all --list-hosts

#检查组成员列表

[root@ansible ~]# ansible all -m authorized\_key -a "user=root \

>exclusive=true manage\_dir=true key='$(< /root/.ssh/id\_rsa.pub)'" -k -v

#批量部署ssh公钥

#exclusive覆盖原公钥、manage\_dir创建.ssh目录

**Afternoon**

1. 批量配置管理

**-模块的使用**

1）**ansible-doc**（相当于shell的man）

[root@ansible ~]# ansible-doc -l #列出所有模块

[root@ansible ~]# ansible-doc 模块名 #查看帮助

2）ping模块

测试ssh的连通性，没有参数。

#每次建立连接后，会在/root/.ansible/cp建立socket文件

[root@ansible ~]# ansible all -m ping

1. command模块（不写-m默认）

直接执行系统命令，不开启bash子进程。

因此"<", ">", "|", "&"都不支持

4）**shell** | raw模块

shell模块与command一样，区别是通过/bin/bash执行命令；

row模块与shell一样，区别是没有chdir、creates、removes参数。

例子：web1、web2创建z3用户，密码改为123，首次登陆需更改密码

[root@ansible ~]# ansible web -m shell -a 'useradd z3; \

> echo 123 | passwd --stdin z3;chage -d 0 z3'

1. **script模块**

批量执行本地脚本。不限于shell脚本（只要指定sha-bang解释器）

[root@ansible ~]# ansible web -m script -a 'script.sh'

6）**copy模块**

复制文件/目录到进程主机。(复制目录的”/”类似rsync)，参数如下：

src=本地地址(必填)

dest=远程地址(必填)

bachup=yes|no (覆盖前是否备份)

force=yes|no (yes强制覆盖文件，no仅文件不存在才复制)

-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/'

7）lineinfile | replace模块

行编辑替换模块。(类似sed -i) 。参数如下：

path=“要修改的文件”

regexp=”正则”

line=”修改后的内容” | replace=”分组匹配修改的样子”

ansible db1 -m lineinfile -a 'path="xx" regexp="^ONBOOT" line="ONBOOT=no"'

ansible db1 -m replace -a 'path="xx" regexp="^(ONBOOT).\*" replace="\1=yes"'

1. **yum模块**

管理软件包(安装、删除)，要求目标主机yum源可用。参数如下：

name=“包1,包2..”(url或本地rpm包)

state=”状态”(installed、removed)

ansible all -m yum -a 'name=httpd state=installed'

1. service模块

启动或关闭服务，设置开机自启。参数如下：

name=”服务名”

state=”动作”(started,stopped,restarted,reloaded)

enabled=”yes|no”(开机自启)

ansible all -m service -a 'name="httpd" state="started" enabled="yes"'

1. setup模块

用于过滤主机信息。（相当于grep）

ansible all -m setup -a 'filter=’ #查看全部参数

ansible all -m setup -a 'filter=ansible\_distribution'#过滤系统发行版

ansible all -m setup --tree out #把主机信息保存到当前out目录