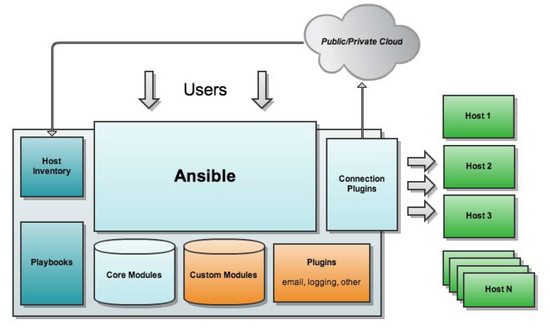
# Ansible 学习心得

## 一、 ansible 简单介绍

Ansible 基于python研发的自动化运维工具, ansible是无客户端也不需要启服务端工具,十分方便，主要基于openssl所以安全性也比较高，但是因为任务按队列依次执行,所以并没有saltsatck那样并发的快.特别是维护上百台机子后会感觉到明显慢很多。

Ansible架构



ansible core ： ansible 自身核心模块

host inventory： 主机库，定义可管控的主机列表

connection plugins： 连接插件，一般默认基于 ssh 协议连接

modules：core modules ( 自带模块 ) 、 custom modules ( 自定义模块 )

playbooks ：剧本，按照所设定编排的顺序执行完成安排任务

## 二、 playbooks

Playbooks 相当于saltstack的状态模块，可以重复执行，管理配置主机。其中目录结构为

.

├── ansible.cfg

├── conf

├── environments

├── roles

├── run.sh

└── ssh-agent.sh

* ansible.cfg主要用于配置ansible
* conf 目录针对主机或者组的变量做定制与分配角色，
* Enviroments可以分地区，分项目通过IP对主机分组，并增加组的变量
* Roles 在里面定义角色用于对应主机的配置与软件安装，主要的任务都是从这里执行。
* run.sh 用于运行ansible playbook

下面就一个个简述一下各目录下的内容

### conf 目录

conf主要针对envirment下面的组进行分配角色，所以变量定义，可以分区域来定制

conf/

└── site

└── aliyun

├── site.zabbix-agent.yml

├── site.zabbix-proxy.yml

└── site.zabbix-server.yml

下面可以简单看其中一个文件的内容

cat conf/site/aliyun/site.zabbix-agent.yml

- hosts: zabbix-agent

remote\_user: yulei

sudo: yes

vars:

zabbix\_server: '192.168.56.32,192.168.56.33'

zabbix\_server\_active: '192.168.56.32,192.168.56.33'

zabbix\_hostmetadataite: "{{ ansible\_hostname }}"

zabbix\_agent\_nginx: true

zabbix\_agent\_redis: false

zabbix\_agent\_mysql: true

zabbix\_agent\_memcached: false

zabbix\_server\_url: http://192.168.56.31/zabbix

zabbix\_login\_user: Admin

zabbix\_login\_password: zabbix

roles:

- zabbix-agent

这里就是对所有的名字zabbix-agent的组或者主机的配置与角色分配，变量以后在template中需要用的。

### environments 目录

environments主要针对每个主机进行分组命名，以便于管理。

environments/

└── aliyun

├── group\_vars

│ └── all

└── hosts

[yulei@localhost ~/ansible]$ cat environments/aliyun/hosts

[zabbix-server]

192.168.56.33

[zabbix-agent]

192.168.56.34

[zabbix-proxy]

192.168.56.32

以后对组定义变量

[yulei@localhost ~/ansible]$ cat environments/aliyun/group\_vars/all

# group\_vars/all

env\_name: aliyun

### roles 目录

roles目录主要用于描述状态，针对目标主机需要完成什么任务，启什么服务等

roles

├── zabbix-agent

│ ├── files

│ ├── handlers

│ ├── meta

│ ├── tasks

│ └── templates

Roles下的目录需要与上面conf上roles下面的名字一致，针对每一个角色下面可以有面列目录

* default 为当前角色设置默认变量 (main.yml)
* files 存放文件，执行状态时分发到目的主机
* handlers 主要用于启动或者重启服务（main.yml）
* meta 定义角色与角色之间的依赖关系（main.yml）
* template jinja模版，分发到目标主机
* task 主要用于描述状态，任务列表，安装软件，修改配置都在这里写

cat roles/zabbix-agent/tasks/mail.yml

---

- stat: path=/home/{{ansible\_env.SUDO\_USER}}/software/zabbix-agent

register: dir\_rules

- stat: path=/etc/zabbix/scripts

register: scripts\_rules

- name: check if zabbix is installed

command: rpm -q zabbix

ignore\_errors: True

changed\_when: no

register: zabbix\_check

- name: check if zabbix-agent is installed

command: rpm -q zabbix-agent

ignore\_errors: True

changed\_when: no

register: zabbix\_agent\_check

- name: make-dest-dir

command: mkdir -p /home/{{ansible\_env.SUDO\_USER}}/software/zabbix-agent

when: dir\_rules.stat.exists == False

- name: copy-zabbix-agent-needed-files

copy: src={{item}} dest=/home/{{ansible\_env.SUDO\_USER}}/software/zabbix-agent/{{item}} mode=0644

with\_items:

- zabbix-agent-2.4.3-1.el6.x86\_64.rpm

- zabbix-2.4.3-1.el6.x86\_64.rpm

when: dir\_rules.stat.exists == False

- name: install-zabbix-agent-1

command: rpm -ivh /home/{{ansible\_env.SUDO\_USER}}/software/zabbix-agent/zabbix-2.4.3-1.el6.x86\_64.rpm

sudo: True

when: zabbix\_check.stdout.find('is not installed') != -1

- name: install-zabbix-agent-2

command: rpm -ivh /home/{{ansible\_env.SUDO\_USER}}/software/zabbix-agent/zabbix-agent-2.4.3-1.el6.x86\_64.rpm

sudo: True

when: zabbix\_agent\_check.stdout.find('is not installed') != -1

register: agent\_pkg

- include: mysql.yml

- name: copy-zabbix-agentd-conf

template: src=etc/zabbix/zabbix\_agentd.conf.j2 dest=/etc/zabbix/zabbix\_agentd.conf mode=0644

notify: restart zabbix-agent

sudo: True

register: agent\_conf

- name: make-zabbix-agent-scripts-dir

command: mkdir -p /etc/zabbix/scripts

sudo: True

when: scripts\_rules.stat.exists == False

- name: copy-scripts-to-agent

template: src={{item}} dest=/etc/zabbix/scripts/ mode=0744 group=zabbix owner=zabbix

with\_items:

- etc/zabbix/scripts/zabbix\_nginx\_status

- etc/zabbix/scripts/zabbix\_pdmi\_application

- etc/zabbix/scripts/zabbix\_redis\_status

sudo: True

register: scripts\_conf

- name: copy-parameter-to-agent

template: src={{item}} dest=/etc/zabbix/zabbix\_agentd.d/ mode=0644

with\_items:

- etc/zabbix/zabbix\_agentd.d/userparameter\_memcached.conf

- etc/zabbix/zabbix\_agentd.d/userparameter\_mysql.conf

- etc/zabbix/zabbix\_agentd.d/userparameter\_nginx.conf

- etc/zabbix/zabbix\_agentd.d/userparameter\_pdmi\_application.conf

- etc/zabbix/zabbix\_agentd.d/userparameter\_percona\_mysql.conf

- etc/zabbix/zabbix\_agentd.d/userparameter\_redis.conf

sudo: True

register: parameter\_conf

- name: start-zabbix-agent

service: name=zabbix-agent state=started enabled=yes

when: agent\_pkg.changed

- name: restart-agent-0

command: echo "restart agent"

when: agent\_conf.changed or scripts\_conf.changed or parameter\_conf.changed

notify: restart zabbix-agent

- name: Create a new host or update an existing host's info

local\_action:

module: zabbix\_host

server\_url: http://192.168.56.33/zabbix

login\_user: Admin

login\_password: zabbix

host\_name: "{{ ansible\_hostname }}"

host\_groups:

- Zabbix servers

# - jd

link\_templates:

- Template App FTP Service

- Template App Zabbix Agent

status: enabled

state: present

inventory\_mode: automatic

interfaces:

- type: 1

main: 1

useip: 1

ip: "{{ ansible\_eth0.ipv4.address}}"

dns: ""

port: 10050

- type: 4

main: 1

useip: 1

ip: '{{ ansible\_eth0.ipv4.address }}'

dns: ""

port: 18080

proxy: zabbix\_proxy

上面主要就是针对安装zabbix-agent描述的状态，但是因为ansible本身效率的问题，依次一个个执行，并且执行错便不会再向下执行，所以需要对所有的name做好判断这样提高执行效率这里就需要使用register对状态时行判断，然后对满足条件的再执行状态。

{{item}} 使用时需要慎重，因为with\_items是for循环，会挨个执行。