Ansible

简单易用的配置部署管理工具

wangxinyu

2013.10

IT运维和开源工具

硬件 / 网络

云 / 虚拟机

操作系统 / 应用软件（安装，配置，初始化）

（Puppet, Chef, CFEngine, Ansible, SaltStack, …）

自动任务（DB备份，数据同步，日志清理等）

（Fabric, Ansible, SaltStack, …）

手动任务（部署应用，升级，重启，增加用户，巡检等）

（Fabric, Rake, Func, Rundeck, Ansible, SaltStack, …）

监控告警

（Nagios,Pingdom,Zabbix,Zenoss,...）

性能报表

（Ganglia,Cacti,Graphite,Zabbix,Zenoss,...）

...

Ansible简介

Ansible是一种IT运维自动化的工具，具有配置管理 / 应用部署 / 流程编排的功能。主要特征：

- 简单，极低的学习曲线
- Python编写
- 基于SSH，Push模块到远程机器执行，不需要Agent
- 支持单独命令执行和指令编排
- 内置大量常用模块，支持模块二次开发
- YAML语言进行指令编排

为什么使用Ansible

手工->shell script->配置管理工具(puppet,saltstack,ansible)

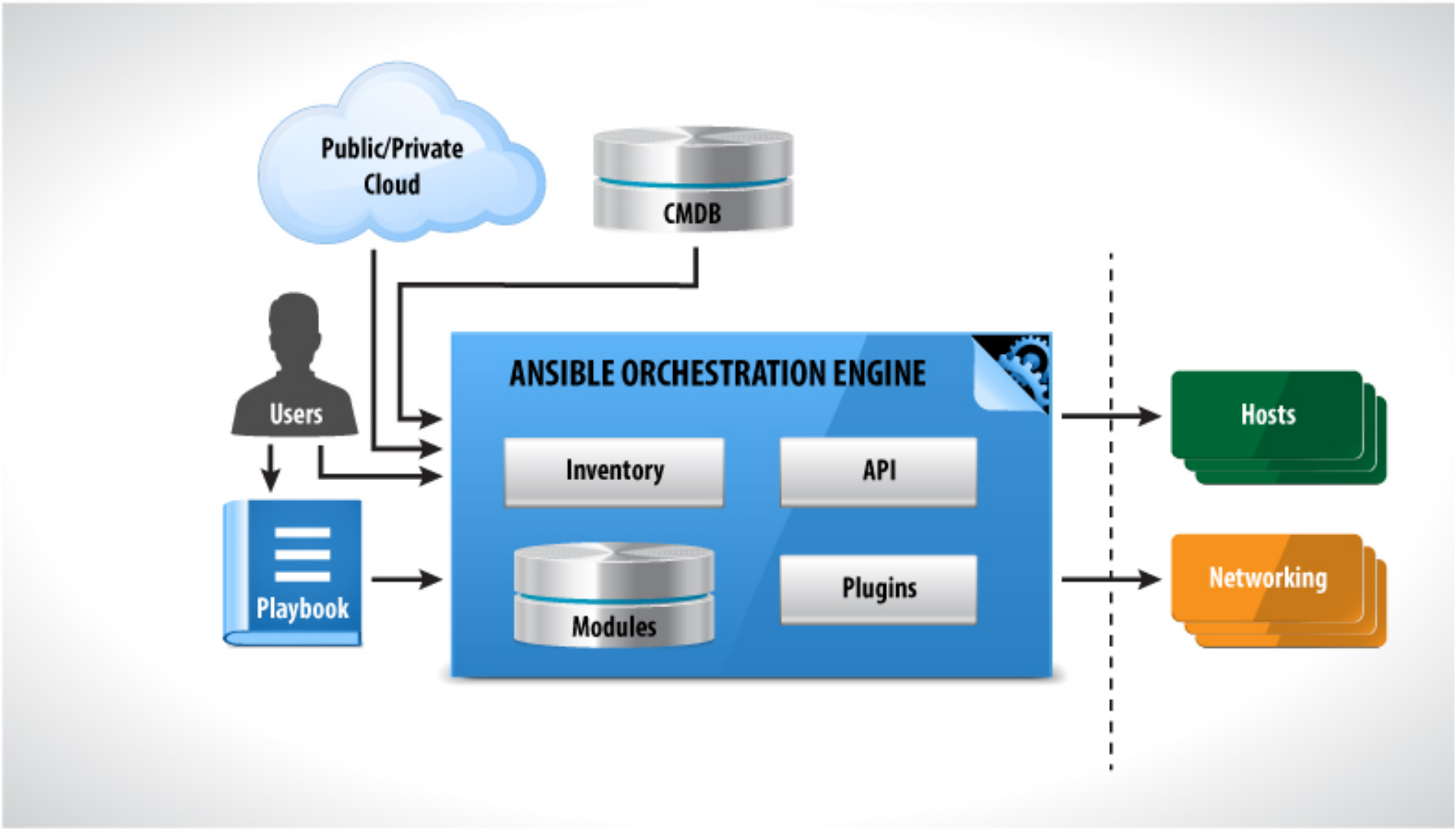
A huge advantage of using a CM tool is that they are “idempotent”.Idempotency basically means that you can run the directives over and over again safely.

相比puppet / saltstack，ansible有更简洁的配置描述语言，极容易使用，没有angent，核心代码量更少 / 容易理解。

Major features of ansible are that it has a minimal learning curve, requires no server side software, and also requires no software to run on remote machines. Machines can be managed simply over SSH, using the Python stacks that are already included in every major Linux distribution by default.

- - Michael DeHaan（Author of Ansible）

Ansible架构


Public/Private
Cloud
CMDB
Users
ANSIBLE ORCHESTRATION ENGINE
Inventory
API
Playbook
Modules
Plugins
Hosts
Networking

Ansible安装

RPM包安装(CentOS)：

$ sudo yum install ansible

源码安装：

$ sudo easy_install pip
$ sudo pip install paramiko PyYAML jinja2
$ git clone git://github.com/ansible/ansible.git
$ cd ./ansible
$ source ./hacking/env-setup

验证：

$ echo "127.0.0.1" > ~/my_hosts
$ export ANSIBLE_HOSTS=~/my_hosts
$ ansible all -m ping –ask-pass

安装后默认主机配置文件/etc/ansible/hosts，这里也可以直接编辑这个文件增加主机。

/etc/ansible/ansible.cfg是ansible运行配置文件。

Ansible - 一个简单示例

$vi my_hosts
$cat my_hosts

```
[dev_group1]
192.168.1.11 ansible_ssh_port=22 ansible_ssh_user=dev ansible_ssh_pass=123456
192.168.1.12 ansible_ssh_port=22 ansible_ssh_user=dev ansible_ssh_pass=123456
```

$ansible dev_group1 -m ping

```
192.168.1.11 | success >> {
    "changed": false,
    "ping": "pong"
}
192.168.1.12 | success >> {
    "changed": false,
    "ping": "pong"
}
```

Ansible组成

Inventory

Ansible (module, command)

Playbook(task, handlers)

Files / Templates / Variables

Inventory

```
# /etc/ansible/hosts
[nginx-servers]
nginx01.example.org
nginx02.example.org
[testbo-services]
testbo01 ansible_ssh_host=192.168.1.11 ansible_ssh_port=7022
testbo02 ansible_ssh_host=192.168.1.12 ansible_ssh_port=7022
```

除了支持静态配置Inventory，还支持动态Inventory（例如通过脚本从CMDB获取服务部署信息）。

Ansible command示例

$ ansible <host-pattern> [-f forks] [-m module_name] [-a args]

使用command模块远程执行命令
$ansible dev_group1 -m command -a 'uname -a'

使用shell模块远程执行命令
$ansible dev_group1 -m shell -a 'ps -ef | grep BUS_BO'

使用copy模块上传一个文件
$ansible dev_group1 -m copy -a 'src=/etc/ansible/hosts dest=/tmp/hosts'

详细命令参考：
$ansible-doc -l
$ansible-doc copy
$ansible-doc yum

Ansible Modules

大量内置模块

add_host,debug,get_url,mount,postgresql_user,slurp,
apt,django_manage,git,mysql_db,rabbitmq_parameter,
subversion,apt_key,easy_install,group,mysql_user,
rabbitmq_plugin,supervisorctl,apt_repository,ec2,group_by,
nagios,rabbitmq_user,svr4pkg,assemble,ec2_facts,hg,ohai,
rabbitmq_vhost,sysctl,async_status,ec2_vol,ini_file,opkg,raw,
template,async_wrapper,facter,libr,pacman,script,uri,
authorized_key,fail,lineinfile,pause,seboolean,user,
cloudformation,fetch,lvol,ping,selinux,virt,command,file,
macports,pip,service,wait_for,copy,fireball,mail,pkgin,setup,
yum,cron,gem,mongodb_user,postgresql_db,shell,zfs

Playbook-nginx安装的一个完整示例

nginx.yml

```
---
- hosts: nginx_server
  sudo: True
  vars:
        domain: example.com
  tasks:
   - name: Install nginx
     yum: name=nginx state=present
   - name: configure nginx
     template: src=templates/nginx.conf.j2 dest=/usr/local/etc/nginx.conf
     notify: nginx-restart
  handlers:
   - name: nginx-restart
     service: name=nginx state=restarted
```

```
$ansible-playbook -K nginx.yml
```

Playbook-后台模块的一个安装示例

```
$cat test_bo_install.yml
---
# This playbook would deploy TEST_BO application
- hosts: test_bo_servers
  roles:
   - role: test_bo
```

Playbook是Ansible的配置 / 部署 / 编排语言，采用YAML语法格式进行描述。Playbook描述了需要远程机器执行的一种策略或者进行IT维护的步骤集合。

Playbook-后台模块的一个安装示例

$cat roles/test_bo/tasks/main.yml

```
---
- name: create install path
  file: path={{test_bo_installdir }} state=directory owner=dev group=dev
- name: upload install pkg
  copy: src=roles/test_bo/files/test_bo.tar.gz dest={{ test_bo_installdir }}/test_bo.tar.gz
- name: unzip install pkg
  shell: tar zxvf {{ test_bo_installdir }}/mileage_stat.tar.gz -C {{test_bo_installdir }}
- name: modi start.sh
  template: src=start.sh dest={{test_bo_installdir }}/bin/start.sh
- name: modi stop.sh
  template: src=stop.sh dest={{test_bo_installdir }}/bin/stop.sh
- name: chmod file
  shell: chmod +x {{ test_bo_installdir }}/bin/*.sh
```

Playbook(YAML)

YAML是一种容易阅读和编写的数据格式协议，和XML / JSON相似。

```
---
# A list
- Apple
- Orange
- Strawberry
- Mango

---
# An dictionary
name: Example Developer
job: Developer
skill: Elite
```

Playbook(Roles)

Roles对变量 / 模板进行了自动处理。

预定义好的文件结构：

```
|-auto_install/
   |---group_vars/
   |---roles/
   |——test_bo/
   |-------files/
   |-------tasks/
   |-------templates/
   |-------handlers/
   |-------vars/
   |-------meta/
```

Files / Templates / Variables

文件是本地机器的规则文件。例如需要上传到目标机器的安装包，密码文件，配置文件等。

模板一般是有变量定义需要部署时修改的文件。

变量在Ansible中用于区分不同系统 / 条件判断和模板配置等，有组变量和全局变量。

Files / Templates / Variables

$ls -l roles/test_bo/files/

-rw-rw-r-- 1 dev dev 2413305 Oct 23 11:23 test_bo.tar.gz

$ls -l roles/test_bo/templates/

-rw-rw-r-- 1 dev dev 395 Oct 23 14:04 start.sh

$grep '{{' roles/test_bo/templates/start.sh

TEST_BO_PATH={{ test_bo_bindir }}

$cat group_vars/test_bo_servers

test_bo_installdir: /opt/bo/test_bo

Ansible Developing

- Developing Dynamic Inventory Sources

通过开发脚本实现动态生成远程管理的主机信息。

- Developing Modules

通过自定义开发的模块整合到playbook中实现针对业务的部署 / 配置 / 控制功能。

- Developing Plugins

Ansible插件包括通信协议插件开发和增加日志 / 监控能力的callback插件开发。例如可以通过二次开发将告警上报到Nagios。

实现业务模块部署自动化

1.针对每个模块编写Playbook，一个业务所有模块的playbook汇集起来组成业务部署包。安装和升级包分开。

2.一些共性的动作编写为Ansible Module。

3.完善已有的RPM打包机制，Playbook通过rpm包进行安装和升级。

4.通过编写动态Inventory脚本实现从CMDB获取主机信息。

参考资料

1. http://www.ansibleworks.com/docs/

需要全面了解ansible，可以从这里开始。

1. https://github.com/mpdehaan/ansible-examples

Playbook示例，参考示例可以得到playbook编写的最佳实践。

1. http://missingm.co/2013/06/ansible-and-salt-a-detailed-comparison/

此文将salt和ansible进行了对比，主要观点：

salt由于采用0mq通信，速度上比ansible快一些。

安全性上ansible稍微好些。

安装 / 部署 / 维护上ansible好于salt，ansible更简洁一些。

作者也倾向于ansible，但是salt也很优秀，会同时使用两者。

1. http://www.slideshare.net/gnosek/ansible
2. http://flocktofedora.org/wp-content/uploads/2013/08/flock-2013-Ansible_Flock_Michael_DeHaan.pdf
3. https://speakerdeck.com/ausmarton/ansible-configuration-management-simplified
4. http://www.nycbug.org/events/10335/ansible_config_mgmt.pdf

参考资料

http://en.wikipedia.org/wiki/Comparison_of_open_source_configuration_management_software

维基上进行了常用配置工具的比较。

	Language	License	Lines of code	Encryps	Verify mode	First release	Lasted statble release
Ansible	python	GPL	10,932	Yes	Yes	2012-03	2013-07
Puppet	ruby	Apache from 2.7.0	323,329	Yes	Yes	2005-08	2013-06
Salt	python	Apache	97,046	Yes	Yes	2011-03	2013-09