

# AUTOMATION FOR EVERYONE

Ansible is designed around the way people work and the way people work together.

WHY ANSIBLE?

GET STARTED

## Ansible在线训练营之 基础篇



高效运维社区  
GreatOps Community

孙博



## 孙博

- 高级运维开发工程师
- 专注于Linux系统集成，分布式存储，虚拟化，容器化
- 热爱开源软件



高效运维社区  
GreatOps Community

# 课程大纲

- Ansible介绍
- Ansible基本使用
- Playbook入门
- Playbook进阶
- Ansible最佳实践
- Roles



高效运维社区  
GreatOps Community



ANSIBLE

# 学习目标及环境要求

- 学习目标：

- 掌握ansible基本原理
- 掌握inventory的写法
- 掌握模块的使用方法
- 掌握playbook的写法

- 环境要求：

- 3台可上网的Linux系统环境 ( Hosted/Virtualbox/KVM/VMware)
  - 1 x ansible-master
  - 2 x client



高效运维社区  
GreatOps Community

# Ansible介绍

- 什么是Ansible
- 为什么选择Ansible
- 工作原理
- Ansible的构成



高效运维社区  
GreatOps Community



ANSIBLE

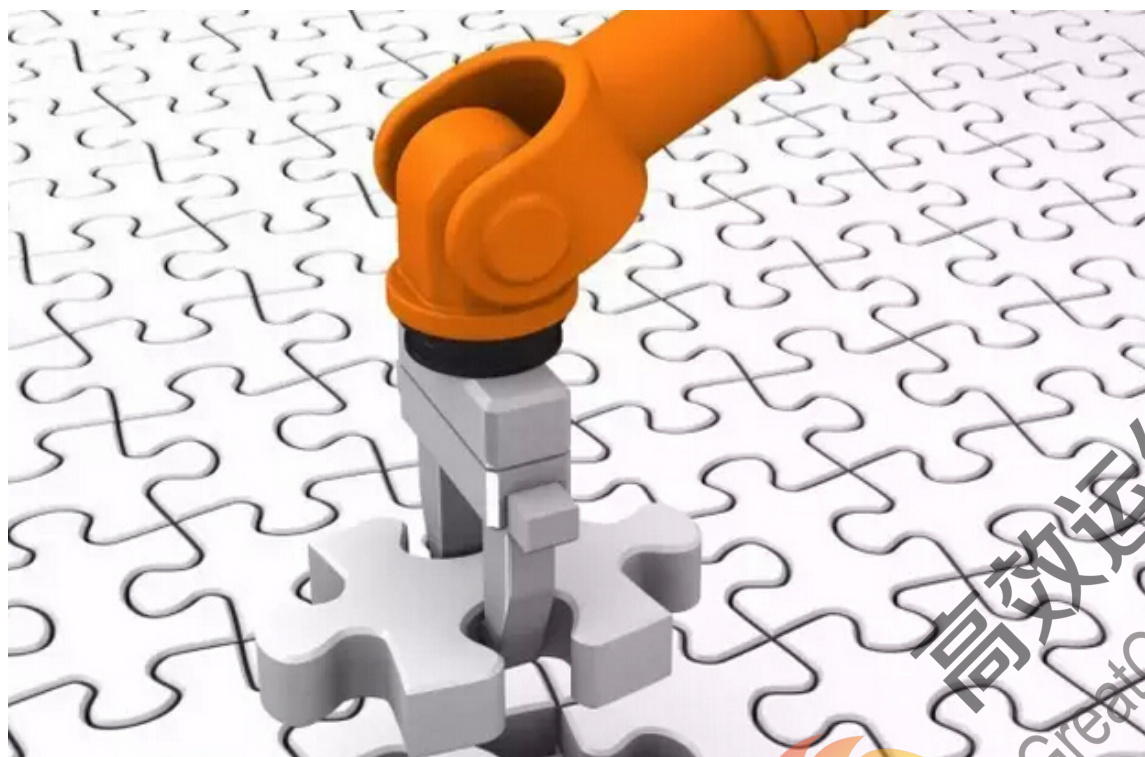
# 什么是Ansible

运维自动化的发展

纯手工阶段

脚本阶段

工具阶段



高效运维社区  
GreatOps Community

## Ansible : “Ansible is Simple IT Automation”

- 一款简单的自动化工具
- 无代理（无需在所管理的系统上安装任何软件）
- 开源，免费使用

### Ansible 提供：

- 应用部署
- 配置管理
- 任务自动化

### 目标用户：

- 运维&开发



高效运维社区  
GreatOps Community

## Ansible 作者：

Michael DeHaan (creator cobbler and func)

<https://www.ansible.com/blog/2013/12/08/the-origins-of-ansible>

## SIMPLE

Can manage almost any \*IX through SSH

requires Python 2.4

Windows (powershell, winrm python module)



“ Ansible owes much of it's origins to time I spent at Red Hat's Emerging Technologies group, which was an R&D unit under Red Hat's CTO ”

- Michael DeHaan

“...because Puppet was too declarative you couldn't use it to do things like reboot servers or do all the "ad hoc" tasks in between... ”

- Michael DeHaan





# 为什么选择Ansible

## puppet VS saltstack VS ansible

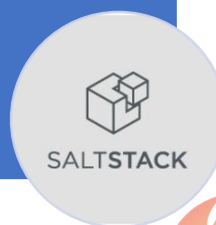
- 开发语言：Ruby
- 是否有客户端：有
- 通信协议：SSL
- 配置文件格式：Ruby
- Web UI: 提供
- 命令行执行: 不支持
- 是否支持二次开发: 不支持

Puppet



- 开发语言：Python
- 是否有客户端：有
- 通信协议：AES加密
- 配置文件格式：YAML
- Web UI: 提供
- 命令行执行: 支持
- 是否支持二次开发: 支持

Saltstack



- 开发语言：Python
- 是否有客户端：无
- 通信协议：OpenSSH
- 配置文件格式：YAML
- Web UI: 只提供商业版本
- 命令行执行: 支持
- 是否支持二次开发: 支持

Ansible



# 特点：



无客户端



简单，易懂



扩展性强，可管理从几十台到数千台节点



ssh连接，安全



强大社区，大量module，role拿来即用

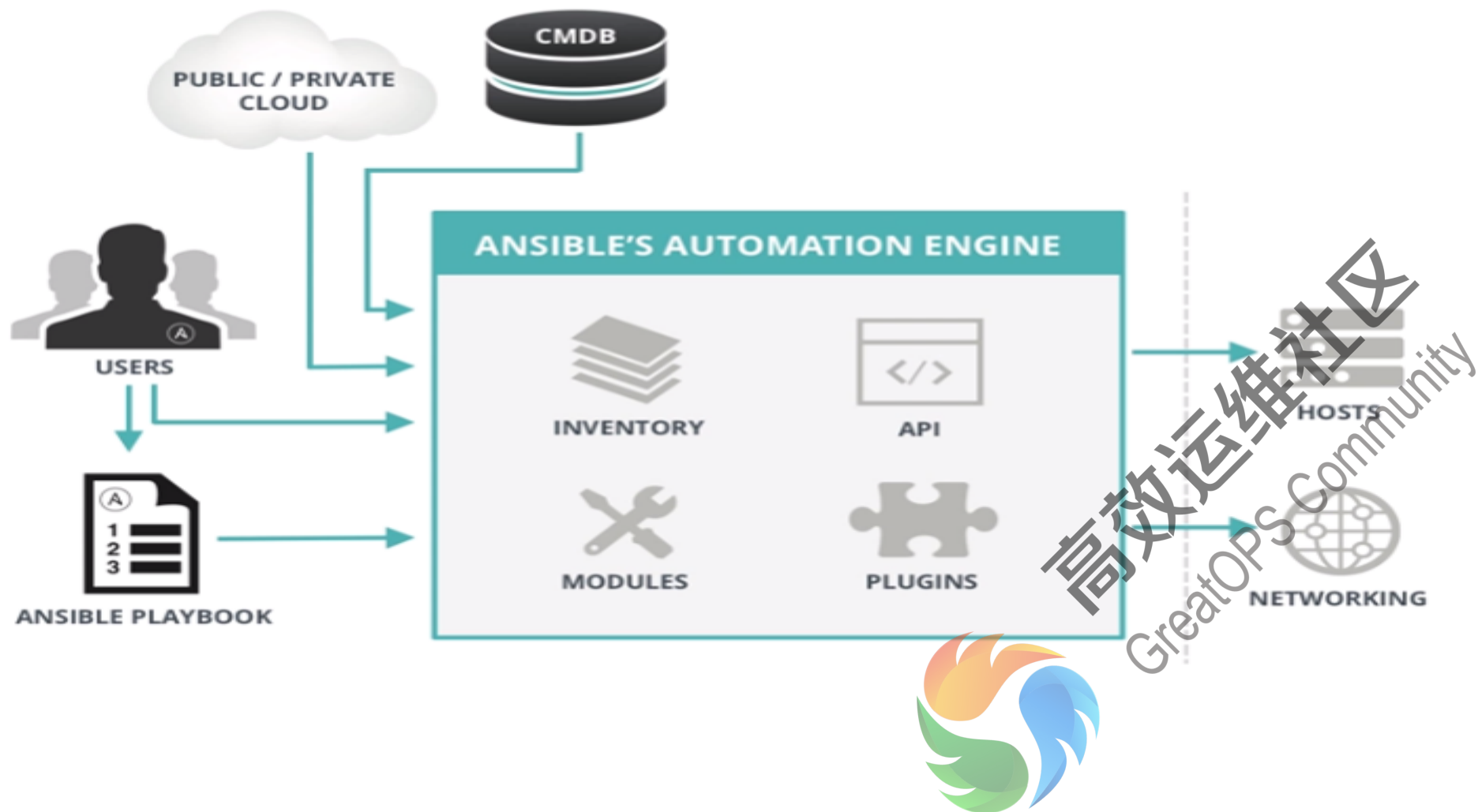


幂等性



高效运维社区  
GreatOps Community

# Ansible工作原理



# Ansible核心构成

- Inventory：定义 Ansible 被管理主机的清单
- Modules：包括 Ansible 自带的核心模块及自定义模块
- Playbooks：剧本，定义 Ansible 多任务配置文件，由 Ansible 自动执行
- API：python接口，提供二次开发及系统整合能力
- Plugins：完成模块功能的补充，包括连接插件、邮件插件等



# Ansible 基本使用

- 如何安装Ansible
- Inventory资源清单
- 初识ad-hoc命令
- 常用模块介绍



高效运维社区  
GreatOps Community



ANSIBLE

# Ansible安装

- yum方式

```
yum install epel-release
yum install ansible -y
```
- apt-get方式

```
apt-get install software-properties-common
apt-add-repository ppa:ansible/ansible
apt-get update
apt-get install ansible
```
- pip方式

```
pip install ansible
```



## 文件组成：

- 可执行文件： /usr/bin/ansible\*
- 配置文件目录： /etc/ansible/
  - ansible-config
  - hosts
  - roles/
- Python lib文件： /usr/lib/python2.7/site-packages/ansible
- Help文档



重要：

在/etc/ansible/ansible-config里关闭  
**host\_key\_checking**

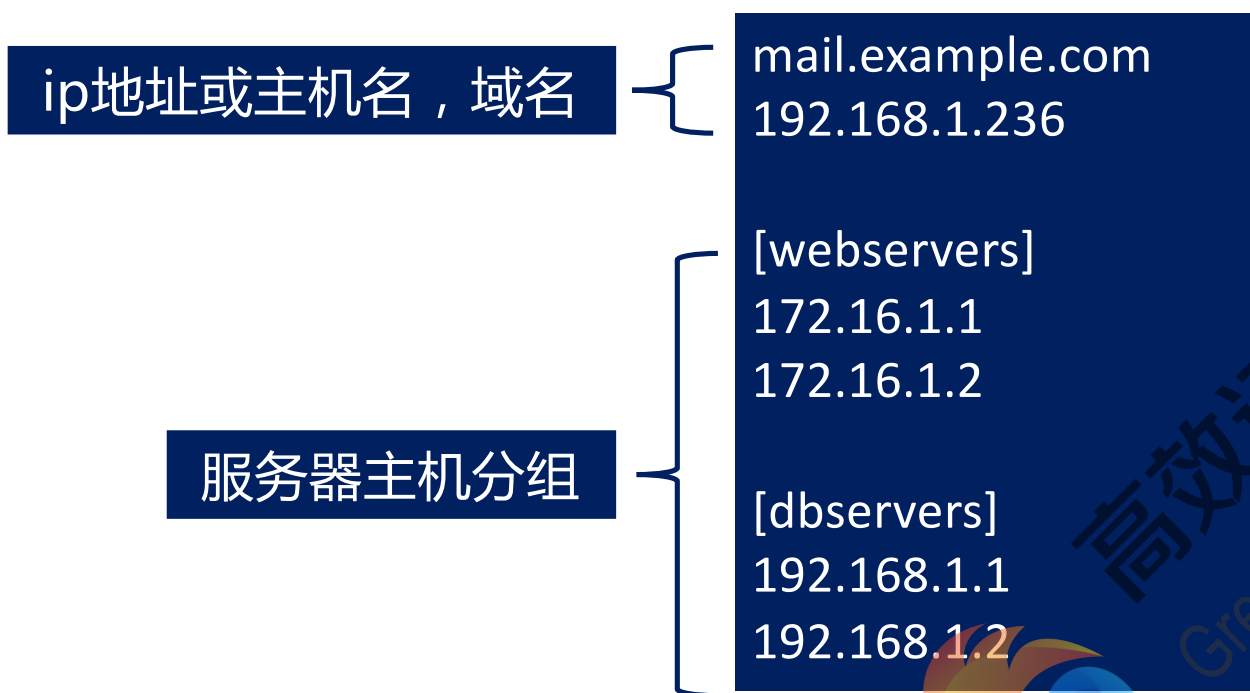


高效运维社区  
GreatOps Community



# Inventory

Ansible 通过读取默认的主机清单配置 `/etc/ansible/hosts`, 可以同时连接到多个远程主机上执行任务, 也可以同时使用多个清单文件, 甚至可以动态地或者从云资源中拉取清单。



# Inventory

使用[start:end]表示连续地址段

db-[a:f].example.com  
www[01:50].example.com  
192.168.1.[10-50]

指定ssh连接端口参数

[dbservers]  
bar.example.com:5309  
  
[targets]  
localhost ansible\_ssh\_port=8022



高效运维社区  
GreatOps Community

# Inventory

主机变量

组变量

## 主机变量和组变量

[alpha]

192.168.1.1 http\_port=80 name=apple

host2 http\_port=8080 name=orange

[beta]

192.168.1.1 ansible\_ssh\_host=192.168.1.1

host2 ansible\_ssh\_host=192.168.1.2

[beta:vars]

ntp\_server=ntp.atlanta.example.com

ansible\_ssh\_user=root

ansible\_ssh\_pass=shroot



高效运维社区  
GreatOps Community

# Ansible ad-hoc

- ad-hoc 是什么  
完成实时的，一次性的，简单的工作

- ad-hoc 语法规则

ansible **-i** inventory group **-m** module **-a** "mod\_args"

主机文件

组名

模块

模块参数



高维运维社区  
GreatOps Community

## 常用参数：

- -i 指定inventory文件
- -m 指定模块
- -a 参数命令
- -u 指定访问用户
- -k 输入密码
- -K 输入sudo密码
- -f 指定并行数量（默认5）



# ansible常用模块

- ping : 用于检测远程主机是否存活
- shell : 在远程主机上执行shell命令
- script : 在远端机器执行本地脚本
- yum/apt : 用于安装软件包
- service : 用于管理服务
- file:用于配置文件属性
- copy : 复制文件到远程主机
- setup : 查看远程主机的基本信息



# ping

检查目标主机是否存活

示例：

- `ansible -i gxyw all -m ping`



# shell

在目标主机执行shell命令

- chdir : 运行shell之前cd到某个目录

示例：

- `ansible all -m shell -a "cat /etc/centos-release"`
- `ansible all -m shell -a "cat centos-release chdir=/etc"`



高效运维社区  
GreatOps Community



# script

在目标机器上执行本地脚本

- chdir : 先进入目标文件夹再执行脚本

示例：

```
ansible db -m script -a "/tmp/add_env.sh"
```

```
ansible db -m script -a "/root/test.py chdir=/var/www/"
```



高效运维社区  
GreatOps Community

# apt/yum

apt、yum 模块分表用于管理Ubuntu 系列和RedHat 系列系统软件包

- name : 软件包名
- state : 软件包的状态 ( present/installed/absent/removed)

示例 :

- 确保httpd 包已经安装  
`ansible webservers -m apt -a "name=httpd state=present"`
- 确保安装包到一个特定的版本  
`ansible webservers -m apt -a "name=httpd-1.5 state=present"`
- 确保一个软件包没有被安装  
`ansible webservers -m apt -a "name=httpd state=absent"`



# service

管理目标机器服务的状态

- name : 必选项 , 服务名称
- 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 )
- enabled : 是否开机启动 yes|no
- runlevel : 运行级别
- arguments : 给命令行提供一些选项

示例 :

- `ansible test -m service -a "name=httpd state=started enabled=yes"`
- `ansible test -m service -a "name=network state=restarted args=eth0 "`
- `ansible test -m service -a "name=httpd state=stopped"`



# copy

将文件从ansible管理机拷贝到目标机器

- src:源文件
- dest:目标路径
- backup:覆盖之前，是否备份原文件
- owner:设定文件/目录的属主
- group:设定文件/目录的属组
- mode:设定文件/目录的权限

示例：

- `ansible test -m copy -a "src=/srv/myfiles/foo.conf dest=/etc/foo.conf owner=foo group=foo mode=644"`
- `ansible test -m copy -a "src=/mine/ntp.conf dest=/etc/ntp.conf owner=root group=root mode=644 backup=yes"`



高效运维社区  
GreatOps Community

# file

操作目标机器文件属性，新建/删除文件，文件夹及链接文件

- group：定义文件/目录的属组
- owner：定义文件/目录的属主
- mode：定义文件/目录的权限
- path：必选项，定义文件/目录的路径
- state:定义文件状态(directory/link/absent/touch/hard)

示例：

- `ansible test -m file -a "src=/etc/fstab dest=/tmp/fstab state=link"`
- `ansible test -m file -a "path=/tmp/ansible mode=755 group=test owner=test"`
- `ansible test -m file -a "path=/tmp/test state=directory"`



高效运维社区  
GreatOps Community

# setup

## 搜集系统信息

- 搜集主机的所有系统信息

```
ansible all -m setup
```

- 搜集系统信息并以主机名为文件名分别保存在/tmp/facts 目录

```
ansible all -m setup --tree /tmp/facts
```

- 搜集和内存相关的信息

```
ansible all -m setup -a "filter=ansible_*_mb"
```

- 搜集网卡信息

```
ansible all -m setup -a "filter=ansible_eth[0-2]"
```



高效运维社区  
GreatOps Community

# 模块帮助

- `ansible-doc -l` #获取所有模块列表
- `ansible-doc module_name` #获取模块的详细信息
- 在线帮助：  
[http://docs.ansible.com/ansible/latest/modules\\_by\\_category.html](http://docs.ansible.com/ansible/latest/modules_by_category.html)



高效运维社区  
GreatOps Community