

Playbook 入门

- 什么是Playbook
- yaml语言介绍
- Playbook小试牛刀



高效运维社区
GreatOps Community



ANSIBLE

什么是Playbook



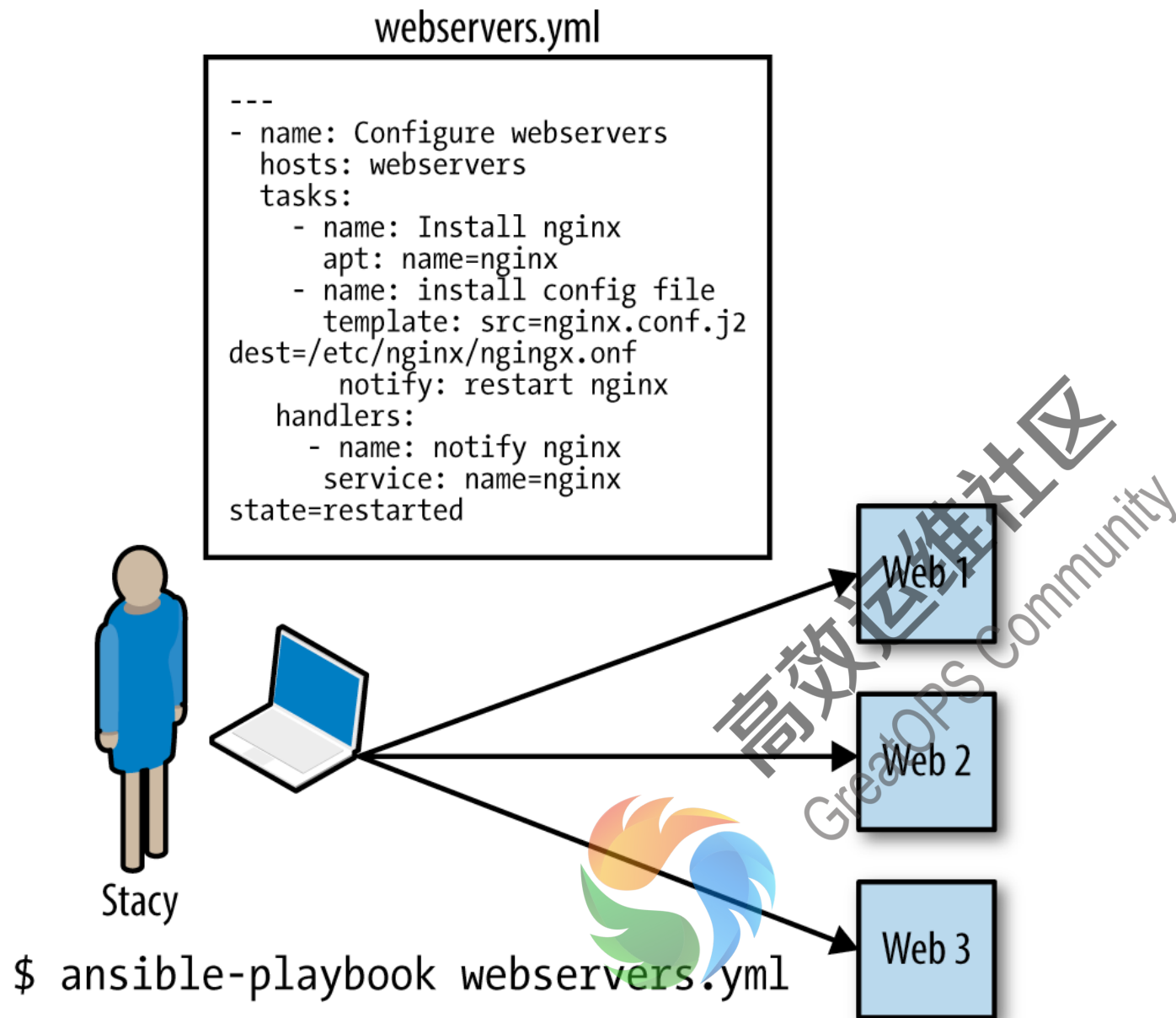
Playbook是

- 以YAML文件作为载体
- 描述多个任务的集合
- 使目标主机按照既定顺序执行任务以达到期望状态
- 通过ansible-playbook命令执行



高效运维社区
GreatOps Community

举个例子



什么是YAML语言

一种人性化的数据格式定义语言

YAML : "YAML Ain't a Markup Language"

- 结构感强，可读性好
- 通用的数据串行化格式



这是一个yaml :

```
1  ---
2  # 一位职工记录
3  name: Daivd
4  job: Developer
5  skill: Elite
6  employed: True
7  foods:
8      - Apple
9      - Orange
10 languages:
11     ruby: Elite
12     python: Elite
```

语法要求：

- 以 '---' 作为起始句
- 大小写敏感
- 使用缩进表示层级关系
- 缩进时不允许使用Tab键，只允许使用空格
- 缩进的空格数目不重要，相同层级的元素左侧对齐即可



数据类型：

- 对象：键值对的集合，又称为映射（ mapping ） / 哈希（ hashes ） / 字典（ dictionary ）
- 数组：一组按次序排列的值，又称为序列（ sequence ） / 列表（ list ）
- 纯量（ scalars ）：单个的、不可再分的值



高效运维社区
GreatOps Community

对象：

```
1  name: David  
2  job: Developer
```

`{"name": "David", "job": "Developer"}`

数组：

```
1  - apple  
2  - orange
```

`["apple", "orange"]`



高效运维社区
GreatOps Community

复合结构：

```
1  students:
2    - name: Lucy
3      age: 15
4    - name: Lily
5      age: 13
```

{students: [{"name": "Lucy", "age": 15}, {"name": "Lily", "age": 13}]}



高效运维社区
GreatOps Community

布尔值：

```
1 create_key: yes/no
2 gather_facts: True/False
3 sudoer: T/F
```



高效运维社区
GreatOps Community

再看yaml：

```
1  ---
2  #John的个人信息
3  name: John Smith
4  age: 37
5  spouse:
6      name: Nini Smith
7      age: 25
8  children:
9      - name: Jimmy Smith
10      age: 15
11      - name: Jenny Smith
12      age: 12
```

YAML 语法检查

Python

```
python -c 'import yaml,sys; print yaml.load(sys.stdin)' < test.yaml
```

YAML Lint

<http://www.yamllint.com/>

Ansible 命令工具

```
ansible-playbook test.yaml --syntax-check
```



高效运维社区
GreatOps Community

A Playbook

```
---  
- name: install and start apache  
  hosts: webservers  
  user: root  
  
  tasks:  
    - name: install httpd  
      yum: name=httpd state=latest  
  
    - name: start httpd  
      service: name=httpd state=running
```

Playbook

Play

Tasks



高效运维社区
GreatOps Community

Play

```
1  ---
2  - hosts: webserver
3    remote_user: root
4    vars:
5      - http_port: 8080
6      - max_clients: 200
7    tasks:
8      - name: ensure httpd is installed
9        yum:
10          name: httpd
11          state: present
12      - name: update httpd config
13        template:
14          src: /srv/httpd.conf.j2
15          dest: /etc/httpd.conf
16      - name: restart httpd service
17        service:
18          name: httpd
19          state: restarted
```

Target Section

Variable Section

Task Section

高效运维社区
GreatOps Community



Play组成部分：

- Target section：定义将要执行playbook的远程主机组及属性
- Variable section：定义playbook运行时需要使用的变量
- Task section：定义将要在远程主机上执行的任务列表



高效运维社区
GreatOps Community

Target section :

- hosts : 定义远程的主机组
- remote_user : 执行该任务组的用户
- sudo : 如果设置为yes , 以sudo权限执行命令
- gather_facts : 默认收集setup模块传递来的变量, 可禁止

```
2  - hosts: webservers
3      remote_user: root
4      sudo: yes
5      gather_facts: no
```



高效运维社区
GreatOps Community

Variable section :

- vars: 直接写入对象形式的变量名及其值
- vars_files : 引入变量所在的文件

```
4  vars:
5      - http_port: 8080
6      - max_clients: 200
7  vars_files:
8      - /tmp/vars
```



高效运维社区
GreatOps Community

系统变量：

- Setup模块（默认gather_facts = yes）

自定义变量：

- Inventory文件
- Playbook文件
- 变量文件

如何引用：`{{ var }}` 双大括号



高效运维社区
GreatOps Community

Task section :

- 由任务组成的列表，其中包含任务名，模块以及模块参数

```
11 tasks:
12   - name: aaaa
13     module 1:
14       ..... #module 1 args
15   - name: bbbb
16     module 2:
17       ..... #module 2 args
18   - name: cccc
19     module 3:
20       ..... #module 3 args
```



高效运维社区
GreatOps Community

运行Playbook

ansible-playbook -i hosts playbook.yml

ansible-playbook -i hosts playbook.yml --syntax-check

ansible-playbook -i hosts playbook.yml --list-hosts

ansible-playbook -i hosts playbook.yml --list-tasks



高可用运维社区
GreatOps Community

Playbook小试牛刀

搭建HTTP服务器

- 要求：在目标机器中安装http服务，修改主页，使其能显示自己的IP及主机名，运行httpd并设置为开机启动
- 步骤：
 1. 安装httpd包
 2. 将index.html传到远端服务器
 3. 使用setup获取的变量，修改index.html的内容
 4. 启动httpd服务



高效运维社区
GreatOps Community

运行结果颜色说明

- 黄色：执行成功，目标机器状态改变
- 绿色：执行成功，目标机器状态不变
- 红色：执行失败



高效运维社区
GreatOps Community