

ansible

是一款 IT 自动化和 DevOps 软件, 自动化部署 app 自动化管理配置项 自动化持续交付 自动化 (AWS) 云服务管理

特性: 主从模式工作 支持自定义模块 支持 playbook 支持异构 IT 环境 支持多层部署

工作流程:

读取配置 → 抓取全量机器&分组列表 → 使用 host-pattern 过滤机器列表 → 根据参数确定执行模块和配置 → Runner 执行返回 → 输出, 结束

1. 安装软件包

```
ansible-2.4.2.0-2.el7.noarch.rpm
python2-jmespath-0.9.0-3.el7.noarch.rpm
python-httpplib2-0.9.2-1.el7.noarch.rpm
python-paramiko-2.1.1-4.el7.noarch.rpm
python-passlib-1.6.5-2.el7.noarch.rpm
```

2. 主机定义与分组

```
vim /etc/ansible/ansible.cfg
```

```
inventory 定义托管主机文件路径 //去掉注释
```

```
host_key_checking=False //去掉注释
```

```
]# ssh-keygen -t rsa -b 2048 -N '' -f key
```

```
]# ssh-copy-id -i key.pub 主机名
```

```
vim /etc/ansible/hosts
```

```
[组名称]
```

```
主机名称或者 IP 地址, 其他参数
```

```
如:
```

```
[web]
```

```
web1
```

```
web2
```

```
[db]
```

```
db1
```

```
db2
```

```
[app:children] //子组定义
```

```
web
db
```

```
[other]    //更改单台主机 ssh 端口, 需在客户端主机修改/etc/ssh/sshd_config
配置端口 222
```

```
cache ansible_ssh_port=222
```

```
[all:vars]    //变量引用
ansible_ssh_private_key_file="/root/.ssh/key"
```

ansible 命令

```
]#ansible all --list-hosts    //检测所有主机
```

```
]#ansible 组名 -m ping -k //使用交互式登录密码
```

```
]#ansible all -m ping
```

自定义配置文件

ansible 配置文件查找顺序

- 首先检测 ansible_config 变量定义的配置文件
- 其次检查当前目录下的./ansible.cfg 文件
- 再次检查当前用户家目录下~/ansible.cfg 文件
- 最后检查/etc/ansible/ansible.cfg 文件

如:

```
]# mkdir myansible
```

```
]#cd myansible
```

```
]#vim ansible.cfg
```

```
[defaults]
```

```
inventory = /root/myansible/myhost
```

```
host_key_checking = False
```

```
]#vim myhost
```

```
[app1]
```

```
web1
```

```
db1
```

```
[app2]
```

```
web2
```

```
db2
```

```
[app:children]
```

```
app1
app2
```

```
]# ansible app2 --list-hosts
hosts (2):
    web2
    db2
```

模块

```
]# ansible-doc -l //列出所有模块
```

```
]# ansible-doc 模块名称 //查看模块帮助
```

command 模块 默认模块, 远程执行命令

```
]# ansible all -m command -a 'uptime' //查看所有机器负载
```

command 模块不能解析系统变量 不能执行"> < | &"的命令 该模块不启动 shell, 直接在 ssh 进程中执行
所有使用到的 shell 命令执行都会失败

shell 模块

shell 模块用法基本与 command 一样, 区别是 shell 模块能通过/bin/sh 进行执行命令, 可以执行任意命令
不能执行交互式的命令, 如 vim top 等

```
]#ansible all -m shell -a 'chdir=/tmp touch testfile'
```

script 模块

在本地写脚本, 然后使用 script 模块批量执行
该脚本包含但不限于 shell 脚本, 只要指定 sha-bang 解释器的脚本都可运行

```
]# vim ur.sh
]# ansible web -m script -a './ur.sh'
```

yum 模块

```
]# ansible all -m yum -a 'name="mariadb-server" state=installed' //给所有主机安装 mariadb
```

service 模块

```
]#ansible all -m service -a 'name="mariadb" enabled="yes" state="started"'
//给所有主机开启 mariadb 服务并设置成开机自启
```

copy 模块

src:复制本地文件到远程主机

dest:远程主机的绝对路径

复制文件

```
]#ansible all -m copy -a 'src=/etc/resolv.conf dest=/etc/resolv.conf'
```

复制目录

```
]# ansible all -m copy -a 'src=/root/my.cnf dest=/root/'
```

lineinfile 模块

-类似 sed 的一种行编辑替换模块

path 目标文件

regexp 正则表达式,要修改的行

line 最终修改的结果

```
]#ansible db -m lineinfile -a 'path="/etc/my.cnf" regexp="^binlog-format"
line="binlog-format=row"'
```

replace 模块

类似 lineinfile, 匹配更改更精确, 适用于更改很长的行

```
]# ansible db -m replace -a 'path="/etc/my.cnf" regexp="=row$"
replace="=mixed"'
```

setup 模块

主要用于获取主机信息,playbooks 里经常会用的另一个参数 gather_facts 与该模块相关,setup 模块下

经常用的是 filter 参数

```
]# ansible cache -m setup -a 'filter="ansible_default_ipv4"'
```

