ansible 基础

一、安装ansible

1-1、先安装epel第三方源（我们这里选择华中科技大学镜像站）

```
wget http://mirror.hust.edu.cn/epel/epel-release-latest-8.noarch.rpm
rpm -ivh epel-release-latest-8.noarch.rpm
```

1-2、 安装ansible

```
yum install -y ansible
```

二、 ansible的简单了解（入门知识）

2-1、 主机清单

```
/etc/ansible/hosts
定义主机和主机组，以及选择主机和主机组
//主机（hosts）部分可以使用域名，主机名（ps.当使用这两种需主机能解析到相应的ip地址），ip地址(ip里面可以使用通配符)表示。
examples：
vim /etc/ansible/hosts
    192.168.1.11
    [mysql]
    192.168.1.100
    192.168.1.101
    [httpd]
    192.168.1.200
    192.168.1.201
    [nginx]
    192.168.1.[15:30]       //若ip多可以这样写 代表从192.168.1.15到192.168.1.30
    [ftp]
    node1
    node2       //可以写在/etc/hosts 中解析过的主机名
    [wuhan:children]            //主机组嵌套
    mysql
    httpd                       //组包含组 用children表示wuhan这个组里包含mysql和httpd组内的ip 而且这里面只能写组名
    [wuhan]
    192.168.1.31             //children表示wuhan的子组 若还需添加IP需重新写wuhan组 这样才不会报错



ansible ip(all) --list-hosts 查看主机清单里是否有该ip存在
// ip为all时列出所有ip ip为分组时 列出该组的ip  // 若ip即在组内有，也单独列出 被视为只在组有

examples: ansible 192.168.1.100,192.168.1.101 --list-hosts //匹配多个ip
          ansible mysql,192.168.1.101 --list-hosts
```

```
          ansible mysql,httpd --list-hosts  //匹配多个组
          ansible ungrouped --list-hosts //列出不属于任何组的ip
          ansible all --list-hosts      //遇到重复IP会自动去重
// 通配符以及通配符组合匹配
          ansible '*.com' --list-hosts  //使用通配符匹配所有.com 结尾的主机
          ansible 's*' --list-hosts // 匹配s开头的主机 和 s开头的主机组内所有主机
          ansible '*.examples.com,!*.lab.examples.com' --list-hosts // 匹配*.examples.com，但是不匹配*.lab.examples.com的主机
          ansible 'db,&wuhan' --list-hosts // 匹配属于db组同时也属于武汉组的主机
          ansible 'boston,london,&prod,!lb' --list-hosts    // 匹配在london组或者boston组，还必须在prod组中且必须不在lb组中的主机
//正则表达式匹配
          ansible '~(s|db).*\.example\.com' --list-hosts
          //开头以~来表示这是一个正则表达式。意思是匹配以s或者db开头，中间包含任意长度字符串并以example.com结尾的主机或者组
//通过 --limit明确指定主机或组
          ansible all --limit srv1.examples.com --list-hosts  //选定指定的主机或组（指定srv1.examples.com）
          ansible all --limit @retry.txt --list-hosts  //选定retry.txt 文件里面的主机ip
```

2-2、配置文件

```
/etc/ansible/ansible.cfg
##优先级##
    ① ANSIBLE_CONFIG                //ANSIBLE_CONFIG环境变量指定配置文件
    ② ./ansible.cfg                 //当前运行应用级配置文件
    ③ ~/.ansible.cfg                //用户级配置文件
    ④ /etc/ansible/ansible.cfg      //系统级配置文件

##ansible.cfg常用配置##
[defaults]
#inventory      = /etc/ansible/hosts                    //定义默认使用的主机清单路径
#library        = /usr/share/my_modules/                //定义默认使用的模块文件路径
#remote_tmp     = ~/.ansible/tmp                        //ansible远程主机脚本临时存放目录
#local_tmp      = ~/.ansible/tmp                        //ansible管理节点脚本临时存放目录
#forks          = 5                                     //ansible执行并发数
#poll_interval  = 15                                    //ansible异步任务查询间隔
#sudo_user      = root                                  //ansible sudo用户
#ask_sudo_pass = True                                   //运行ansible是否提示输入sudo密码
#ask_pass      = True                                   //运行ansible是否提示输入密码
#transport      = smart                                 //ansible远程传输模式
#remote_port    = 22                                    //远程主机SSH端口
#module_lang    = C
#remote_user    = root                                  //默认登录用户可以修改为其他用户
[inventory]
[privilege_escalation]
#become=True                                            //如果远程主机连接的为普通用户，是否提权
```

```
#become_method=sudo                                    //如果允许提取，定义提权方式，默认sudo
#become_user=root                                      //定义提权到哪个用户，默认root
#become_ask_pass=False                                 //提权时，是否交互提示密码验证，默认false
[paramiko_connection]
[ssh_connection]
[persistent_connection]
[accelerate]
#accelerate_port = 5099
#accelerate_timeout = 30
#accelerate_connect_timeout = 5.0
[selinux]
#host_key_checking = False                             //默认false，跳过第一次ssh连接输入yes/no选项
[colors]
#highlight = white
#verbose = blue
#warn = bright purple
#error = red
#debug = dark gray
#deprecate = purple
#skip = cyan
#unreachable = red
#ok = green
#changed = yellow
#diff_add = green
#diff_remove = red
#diff_lines = cyan
[diff]
examples:
    ansible 被控端选择其他用户然后提权
    1、修改ansible.cfg 中的 #remote_user    = root
    2、将[privilege_escalation]选项下的所有注释取消
    3、ssh-keygen做免密
    4、在主控端键入ansible agent -m shell -a 'echo ansible ALL=\(ALL\) NOPASSWD:ALL > /etc/sudoers.d/ansible' -u root
    修改被控端sudo权限即可
```

2-3、ad-hoc语法

```
##ad-hoc命令说明
    ansible 主机或者组 -m 模块名 -a '模块参数' ansible参数
//主机或者组
    在/etc/ansible/hosts 里进行指定
//模块名
    可通过 ansible-doc -l 里查看目前安装的模块
    默认不指定时使用command模块
    可在配置文件/etc/ansible/ansible.cfg 中"#module_name = command" 修改
//模块参数
    可以通过" ansible-doc 模块名 "查看具体用法和参数
//ansible参数
    可以通过ansible命令帮助信息里查看


##ansible-doc用法
```

```
    ansible-doc -l //查看所有模块
    ansible-doc -s module // 查看某个模块的参数
    ansible-doc module //查看该模块更详细的参数信息
example：
    ansible-doc -l | grep yum //查询yum模块
    ansible-doc -s yum  //查询yum模块的参数 里面带例子examples
    ansible-doc yum //查询yum所有参数，以及在playbook的用法 带实例
```