## ****Ansible****

# 安装部分

* 1. **Ansible安装**
     1. **安装**

|  |
| --- |
| * yum install http://mirrors.163.com/centos/7.4.1708/extras/x86\_64/Packages/epel-release-7-9.noarch.rpm * yum install ansible -y |

* + 1. **配置**

**vim /etc/hosts #增加**

|  |
| --- |
| 192.168.193.63 k8s-master  192.168.193.65 k8s-node1  192.168.193.66 k8s-node2 |

**vim /etc/ansible/ansible.cfg #打开注释**

|  |
| --- |
| host\_key\_checking = False |

**vim /etc/ansible/hosts #增加**

|  |
| --- |
| [k8s]  k8s-master ansible\_ssh\_user="root" ansible\_ssh\_pass="eccom"  k8s-node1 ansible\_ssh\_user="root" ansible\_ssh\_pass="eccom"  k8s-node2 ansible\_ssh\_user="root" ansible\_ssh\_pass="eccom" |

* + 1. **验证**

**ansible k8s -m ping**

|  |
| --- |
| [root@localhost etc]# ansible k8s -m ping  k8s-master | SUCCESS => {  "changed": false,  "ping": "pong"  }  k8s-node1 | SUCCESS => {  "changed": false,  "ping": "pong"  }  k8s-node2 | SUCCESS => {  "changed": false,  "ping": "pong"  } |

* 1. **Ansible Tower安装**
     1. **下载安装包 ansible-tower-setup**

https://www.ansible.com/products/tower/trial

or

<https://pan.baidu.com/share/init?surl=ebc2tBhC8XSz-Ycq1fkxgA> 提取码： xq2n

* + 1. **安装前配置及检查**
* 内存4G以上
* 创建目录文件：mkdir -p  /var/log/tower
* 配置修改

vim  inventory

|  |
| --- |
| [tower] localhost ansible\_connection=local [database]  [all:vars] admin\_password= 'eccom' pg\_host='localhost' pg\_port='5432'  pg\_database='awx' pg\_username='awx' pg\_password='admin' |

  关于集群版参考：

https://docs.ansible.com/ansible-tower/latest/html/quickinstall/install\_script.html

* + 1. **创建本地数据库**
* mkdir -p /data/pgdata
* docker pull postgres:9.6 （需要安装docker）
* docker run --name postgres\_ansible -d -p 5432:5432 -e POSTGRES\_PASSWORD=admin -e PGDATA=/data/pgdata -e POSTGRES\_USER=awx -e POSTGRES\_DB=awx postgres:9.6

**验证操作**

|  |
| --- |
| docker exec -it postgres\_ansible bash  psql -U awx -d awx -h localhost -p 5432 ctrl+p+q #退出容器 |

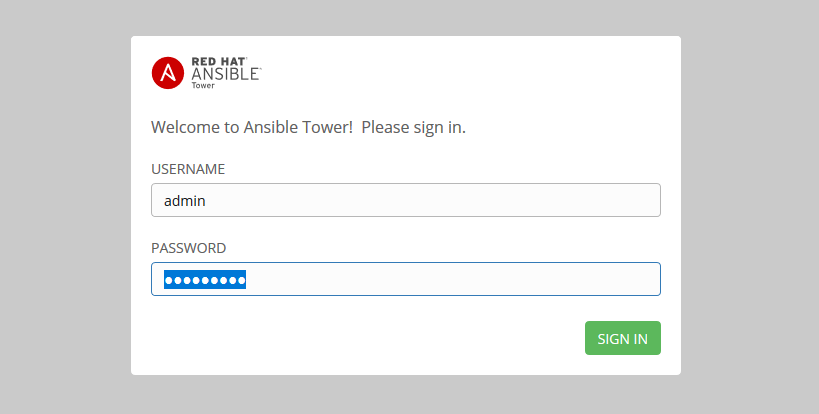
* + 1. **执行安装脚本**

sh setup.sh

安装成功输出日志

|  |
| --- |
| PLAY [Install Tower isolated node(s)] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  skipping: no hosts matched  PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  localhost : ok=123 changed=50 unreachable=0 failed=0  The setup process completed successfully.  Setup log saved to /var/log/tower/setup-2019-04-12-17:55:10.log |

**可以通过80端口进行ansible tower 访问**

****

 license 官方免费申请 <https://www.ansible.com/license>

无限node破解 /var/lib/awx/i18n.db

# Ansible概述

* 1. **特点**
* 部署简单，只需在主控端部署 Ansible 环境，被控端无需做任何操作。
* 默认使用 SSH（Secure Shell）协议对设备进行管理。
* 主从集中化管理。
* 配置简单、功能强大、扩展性强。
* 支持 API 及自定义模块，可通过 Python 轻松扩展。
* 通过 Playbooks 来定制强大的配置、状态管理。
* 对云计算平台、大数据都有很好的支持。
* 提供一个功能强大、操作性强的 Web 管理界面和 REST API 接口 —- AWX 平台。
  1. **常用命令**
     1. **ansible-doc**

ansible-doc -l 列出支持模块

ansible-doc ping 查看ping模块用法

**PS ：以下Client 是配置文件定义的组**

**学习定义组可以参照/etc/ansible/hosts 中的例子，相当于集合，便于批量操作**

* + 1. **script**

script 的功能是在远程主机执行主控端存储的 shell 脚本文件，相当于 scp + shell 组合。

shell > ansible Client -m script -a "/home/test.sh 12 34" # 远程执行本地脚本

* + 1. **shell**

shell模块基本和command相同，但是shell支持管道符

shell > ansible Client -m shell -a "/home/test.sh" # 执行远程脚本

* + 1. **copy**

实现主控端向目标主机拷贝文件，类似于 scp 功能

shell > ansible Client -m copy -a "src=/home/test.sh dest=/tmp/ owner=root group=root mode=0755" # 向 Client 组中主机拷贝 test.sh 到 /tmp 下，属主、组为 root ，权限为 0755

* + 1. **stat**

获取远程文件状态信息，atime/ctime/mtime/md5/uid/gid 等信息

shell > ansible Client -m stat -a "path=/etc/syctl.conf"

* + 1. **get\_url**

实现在远程主机下载指定 URL 到本地，支持 sha256sum 文件校验

shell > ansible Client -m get\_url -a "url=http://www.baidu.com dest=/tmp/index.html mode=0440 force=yes"

* + 1. **yum**

软件包管理

shell > ansible Client -m yum -a "name=curl state=latest"

* + 1. **corn**

远程主机 crontab 配置

shell > ansible Client -m cron -a "name='check dirs' hour='5,2' job='ls -alh > /dev/null'"

* + 1. **mount**

远程主机分区挂载

shell > ansible Client -m mount -a "name=/mnt/data src=/dev/sd0 fstype=ext4 opts=ro state=present"

* + 1. **service**

远程主机系统服务管理

shell > ansible Client -m service -a "name=nginx state=stoped"

shell > ansible Client -m service -a "name=nginx state=restarted"

shell > ansible Client -m service -a "name=nginx state=reloaded"

* + 1. **user**

远程主机用户管理

shell > ansible Client -m user -a "name=wang comment='user wang'"

shell > ansible Client -m user -a "name=wang state=absent remove=yes" # 添加删除用户

* 1. **单命令模式ansible Ad-Hoc**

用法 ansible <hostname pattern> <options>

|  |  |
| --- | --- |
| 常用Option | describe |
| -f NUM | 并发线程数 默认5 |
| -m NAME | 模块 如 ping yum |
| -a ‘args’ | ansible apps -a ‘df -lh’’ |
| -K | 用户密码sudo时使用 |
| -t dir | 输出信息到dir下 |
| -T NUM | 远程最大超时间 |
| -B NUM | 超过NUM后终止正在执行任务 |
| -P NUM | --poll 定期返回后台执行进度 |
| -u username | 用户名 |
| --list | 列出符合条件的主机列表 |
| -l ~REGEX | 正则运行指定主机 |
| -l SUBSET | 运行指定主机 |

更多见：<https://docs.ansible.com/ansible/latest/user_guide/intro_adhoc.html>

* 1. **批量命令模式-剧本 ansible playbook**
* 基于yaml格式 .yml ,
* 剧本执行出错时自动停止
* 系统现有状态与剧本内容一致时默认跳过
* 限制执行剧本对象
* 自定义组下的管理变量替代hosts 中的group var
* ansible-vault 加密解密机制
* Block块编程，增强脚本的逻辑与异常处理
* Include封装执行过程，以供重复调用
  1. **项目级命令模式Ansible Roles**

基于目录文件结构构建ansible