



实验室

云实验室(5) - kubesphere&k8s

2021-11-01 / 0 评论 / 0 点赞 / 145 阅读 / 2,345 字

 温馨提示：
本文为原创作品，感谢您喜欢~

1. 与rancher对比

1.1 功能体验

「1.1.1 rancher 2.6.2 && kubernetes 1.2.1」

劣势：过时的流水线,改用fleet进行devops尝试,暂不是很成熟

优点：rancher更偏向运维使用,多集群管理和基于web UI 半自动快速建立集群相当的方便,便于大规模集群使用.

「1.1.2 kubesphere 3.1.1 && kubernetes 1.2.0」

劣势：部署完全基于shell自动,没有webUI,部署,另外部署过程中还有些小坑,小规模使用还是勉强可以

优点：kubesphere比较均衡,[集群管理]模块偏向运维使用,[项目管理]研发使用,运维配合即可,整体UI风格也要优于rancher.

上一篇记录了部署rancher和初体验,本篇记录kubesphere的搭建.

1.2 环境准备

「1.2.1 实验虚拟机 centos7.9 (基于proxmox-debian金属机)」



Plain text



```
1 name: node1, address: 192.168.0.123, cpu: 4,memory: 4G,disk: 50G
2 name: node2, address: 192.168.0.124, cpu: 4,memory: 4G,disk: 50G
3 name: node3, address: 192.168.0.125, cpu: 4,memory: 4G,disk: 50G
```

```
3   name: node3, address: 192.168.0.123, cpu: 4,memory: 4G,disk: 50G
4   name: node4, address: 192.168.0.126, cpu: 4,memory: 4G,disk: 50G
5   name: node5, address: 192.168.0.127, cpu: 4,memory: 4G,disk: 50G
6   name: node6, address: 192.168.0.128, cpu: 4,memory: 4G,disk: 50G
```

「 1.2.2 网络 」

虚拟机虚拟网卡桥接宿主机物理网卡,访问公网.

「 1.2.3 初始化 」

脚本/命令 与rancher前期准备相同,见 : <https://www.buukle.top/archives/si-you-yun-4-rancherk8s>
另外执行:

Plain text

```
1  yum install -y openssl openssl-devel
2
```

「 1.2.4 节点免密 」

① 设置host

Plain text

```
1  vi /etc/hosts
```

完成后 :

Plain text

```
1  192.168.0.123 node1
2  192.168.0.124 node2
3  192.168.0.125 node3
4  192.168.0.126 node4
5  192.168.0.127 node5
6  192.168.0.128 node6
```

② 生成公私证书

Plain text

```
1  mkdir .ssh
2  cd .ssh
3  ssh-keygen -t rsa
```

```
4 ssh-copy-id -i ~/.ssh/id_rsa.pub node1
```

③ 推送证书

在6台节点分别执行

```
1 ssh-copy-id -i ~/.ssh/id_rsa.pub node1 ## 输入密码
2 ssh-copy-id -i ~/.ssh/id_rsa.pub node2 ## 输入密码
3 ssh-copy-id -i ~/.ssh/id_rsa.pub node3 ## 输入密码
4 ssh-copy-id -i ~/.ssh/id_rsa.pub node4 ## 输入密码
5 ssh-copy-id -i ~/.ssh/id_rsa.pub node5 ## 输入密码
6 ssh-copy-id -i ~/.ssh/id_rsa.pub node6 ## 输入密码
```

「 1.2.5 检查resolv.conf文件,清除其中当前网络不能识别的记录 」

```
1 vim /etc/resolv.conf
```

示例：

```
nameserver 8.8.8.8
search localdomain
~
```

2. 下载脚本准备配置

参照官方文档：<https://kubesphere.io/docs/installing-on-linux/introduction/multioverview/>

```
1 # 下载执行脚本
2 curl -sfL https://get-kk.kubesphere.io | VERSION=v1.1.1 sh -
3 # 生成配置文件
4 ./kk delete cluster -f ./config-kubesphere.yaml
5 # 修改配置文件
6 vim config-kubesphere.yaml
```

局部示例：

```
name: sample
spec:
  hosts:
    - {name: node1, address: 192.168.0.123, internalAddress: 192.168.0.123, privateKeyPath: "~/ssh/id_rsa"}
    - {name: node2, address: 192.168.0.124, internalAddress: 192.168.0.124, privateKeyPath: "~/ssh/id_rsa"}
    - {name: node3, address: 192.168.0.125, internalAddress: 192.168.0.125, privateKeyPath: "~/ssh/id_rsa"}
    - {name: node4, address: 192.168.0.126, internalAddress: 192.168.0.126, privateKeyPath: "~/ssh/id_rsa"}
    - {name: node5, address: 192.168.0.127, internalAddress: 192.168.0.127, privateKeyPath: "~/ssh/id_rsa"}
    - {name: node6, address: 192.168.0.128, internalAddress: 192.168.0.128, privateKeyPath: "~/ssh/id_rsa"}
  roleGroups:
    etcd:
      - node1
    master:
      - node1
    worker:
      - node1
      - node2
      - node3
      - node4
      - node5
      - node6
  controlPlaneEndpoint:
    domain: lb.kubesphere.local
    address: ""
```

3. 部署

Plain text

```
1 ./kk create cluster -f ./config-kubesphere.yaml
```

观察日志,直到部署成功;

4. 卸载

「 4.1 卸载集群 」

Plain text

```
1 ./kk delete cluster -f ./config-kubesphere.yaml
```

「 4.2 清理 」

Plain text

```
1 docker stop $(docker ps -a -q)
2 docker rm $(docker ps -a -q)
3 docker rmi $(docker images -q)
4 #删除/var/etcd目录
```

```

5  sudo rm -rf /var/etcd
6  #删除/var/lib/kubelet/目录, 删除前先卸载
7  for m in $(sudo tac /proc/mounts | sudo awk '{print $2}'|sudo grep /var/lib/kub
8  sudo rm -rf /var/lib/kubelet/
9  #删除/var/lib/rancher/目录, 删除前先卸载
10 for m in $(sudo tac /proc/mounts | sudo awk '{print $2}'|sudo grep /var/lib/ran
11 sudo rm -rf /var/lib/rancher/
12 #删除/run/kubernetes/ 目录
13 sudo rm -rf /run/kubernetes/
14 #删除所有的数据卷
15 sudo docker volume rm $(sudo docker volume ls -q)
16 #再次显示所有的容器和数据卷, 确保没有残留
17 sudo docker ps -a
18 sudo docker volume ls

```

5. 效果



平台管理

工作台

KUBESPHERE

admin

default

集群管理

概览

节点管理

集群节点

服务组件

项目管理

应用负载

配置中心

自定义资源 CRD

存储管理

监控告警

集群设置

集群节点

集群节点提供了当前集群下节点的运行状态。以及可以编辑删除节点

了解更多

> 集群节点的类型?

> 什么是节点污点?

6

1

6

节点数量

Master 节点

工作节点

输入查询条件进行过滤

<input type="checkbox"/>	名称	状态	角色	CPU	内存	容器组	已分配 CPU	已分配内存	
<input type="checkbox"/>	node6 192.168.0.128	运行中	worker	1% 0.05/4 Core	40% 1.48/3.7 Gi	5% 5/110	0.562 Core (15%) 资源预留	1.729 Gi (57%) 资源预留	⋮
<input type="checkbox"/>	node5 192.168.0.127	运行中	worker	1% 0.05/4 Core	16% 0.61/3.7 Gi	5% 5/110	0.562 Core (15%) 资源预留	0.283 Gi (9%) 资源预留	⋮
<input type="checkbox"/>	node4 192.168.0.126	运行中	worker	1% 0.05/4 Core	16% 0.58/3.7 Gi	5% 5/110	0.712 Core (19%) 资源预留	0.514 Gi (16%) 资源预留	⋮
<input type="checkbox"/>	node3 192.168.0.125	运行中	worker	2% 0.09/4 Core	32% 1.19/3.7 Gi	12% 13/110	1.132 Core (31%) 资源预留	1.245 Gi (41%) 资源预留	⋮
<input type="checkbox"/>	node2 192.168.0.124	运行中	worker	2% 0.09/4 Core	26% 0.96/3.7 Gi	9% 10/110	0.687 Core (19%) 资源预留	0.776 Gi (25%) 资源预留	⋮
<input type="checkbox"/>	node1 192.168.0.123	运行中	control-plane, master, worker	3% 0.13/4 Core	44% 1.63/3.7 Gi	13% 14/110	1.107 Core (30%) 资源预留	0.581 Gi (19%) 资源预留	⋮

共 6 个条目

People meet and leave
人们相遇又告别

1 / 1



0

架构实验室



- 👤

版权归属： 惊羽
- 🔗

本文链接： <https://i.buukle.top/archives/si-you-yun-5-kubespherek8s>
- 📄

许可协议： 本文使用《署名-非商业性使用-相同方式共享 4.0 国际（CC BY-NC-SA 4.0）》协议授权

广告

buukle : generator-plus

介绍

布壳儿脚手架服务

原型管理 : 集中管理自定义的archetype ,并能一键在线直接根据archetype生成项目代码

逆向管理 : 可自由灵活配置多数据源, 自定义模板, 并根据模板一键在线生成前后端代码, 支持一键导入本地项目结构

[上一篇](#)

[下一篇](#)

评论区

大神, 请留下你的哲思 ...

戳我试试 OwO

* 昵称


* 电子邮件


提交

暂无评论

2022 © 惊羽 - 京ICP备18025122号-1

本站点由  阿里云 提供云服务

 已运行 1999 天 15 时 54 分 48 秒

 [RSS](#) [站点地图](#)