

# 单节点rancher快速部署

我们知道kubernetes学习成本非常高,如果想要快速体验Paas云平台的话,连部署搭建对于新手来说都是难以下手。



rancher是一个 kubernetes-as-a-service 解决方案,也就是说rancher将kubernetes集群搭建与应用编排实现了打包一体化,可以通过图形界面的操作就可以来部署kubernetes集群与应用了。

参考官网: <https://rancher.com/>

下面我们来使用容器方式快速部署单节点rancher

## 环境准备

系统	docker版本	rancher版本	IP
centos7.6	19.03.1	2.X	192.168.122.12

准备工作:

1, 配置静态IP地址(要求能上公网,因为要拉取镜像)

```
# vim /etc/sysconfig/network-scripts/ifcfg-eth0
BOOTPROTO="static"
NAME="eth0"
DEVICE="eth0"
ONBOOT="yes"
IPADDR=192.168.122.12
NETMASK=255.255.255.0
GATEWAY=192.168.122.1
DNS1=192.168.122.1
```

```
# systemctl stop NetworkManager
# systemctl disable NetworkManager
# systemctl restart network
# systemctl enable network
```

## 2, 主机名和主机名绑定

```
# hostnamectl set-hostname --static rancher
```

```
# vim /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4
localhost4.localhost4
::1         localhost localhost.localdomain localhost6
localhost6.localhost6
192.168.122.12 rancher
```

## 3, 所有节点关闭centos7的firewalld防火墙,打开iptables并保存为空规则

```
# systemctl stop firewalld
# systemctl disable firewalld

# yum install iptables-services -y
# systemctl restart iptables
# systemctl enable iptables

# iptables -F
# iptables -F -t nat
# iptables -F -t mangle
# iptables -F -t raw

# service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[
OK  ]
```

## 4, 所有节点关闭selinux

```
# vim /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of
#                 enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled          确认为disabled,如果不是请改成
disabled(需要reboot重启生效)
# SELINUXTYPE= can take one of three values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only
#                 selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

## 5, 所有节点 时间同步

```
# systemctl restart ntpd
# systemctl enabled ntpd
```

## 需要保证安装docker,并启动服务

```
# wget https://download.docker.com/linux/centos/docker-
ce.repo -O /etc/yum.repos.d/docker-ce.repo
# yum install docker-ce -y
# systemctl restart docker
# systemctl enable docker
```

## 配置镜像加速器

```
# vim /etc/docker/daemon.json
{
    "registry-mirrors":
    ["https://42h8kzrh.mirror.aliyuncs.com"]
}
# systemctl daemon-reload
# systemctl restart docker
```

# 启动rancher

---

```
[root@rancher ~]# docker run -d --restart=unless-stopped -p
80:80 -p 443:443 --name rancher -v
/var/lib/rancher/:/var/lib/rancher/ rancher/rancher:stable
Unable to find image 'rancher/rancher:stable' locally
stable: Pulling from rancher/rancher
35c102085707: Pull complete
251f5509d51d: Pull complete
8e829fe70a46: Pull complete
6001e1789921: Pull complete
190724123869: Pull complete
f144ec07b677: Pull complete
eb837ce35a44: Pull complete
71ab38270b07: Pull complete
d69cad2e554a: Pull complete
33d6cdb7458b: Pull complete
f5a0fc4d0fec: Pull complete
2bd480581d18: Pull complete
d81d5f05f9e8: Pull complete
Digest:
sha256:b85f1dd239d2555ef438f46790642334e0c75f314a35047b93acb1
b457b4cd09
Status: Downloaded newer image for rancher/rancher:stable
3ff8ca28cede2b119ecceb27337748631ebc2659846d20671fbfdda456ec
433
```

## 通过浏览器访问配置

---

直接访问节点IP即可: <https://192.168.122.12>

注意: 因为我们没有配置SSL证书, 所以浏览器会报不安全的连接, 如果要对公网使用, 建议使用CA认证的证书。

## Welcome to Rancher

The first order of business is to set a strong password for the default `admin` user.

☒ Allow collection of anonymous statistics [Learn More](#)

---

☒ Set a specific password to use:

New Password

Confirm Password

☐ Use a new randomly generated password:

[Continue](#)

指定admin管理员的密码

## Rancher Server URL

What URL should be used for this installation? All the nodes in your clusters will need to be able to reach this.

URL

[https://](#)

我还没有域名,直接使用IP访问

**i**

Are you sure all the hosts you will create will be able to reach `192.168.122.12`? It looks like a private IP or local network.

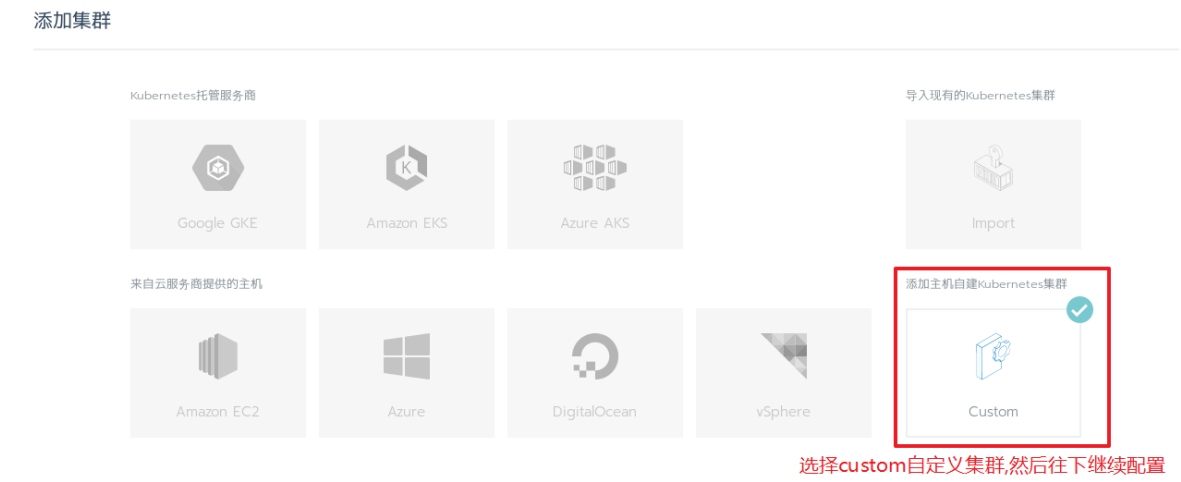
[Save URL](#)

# 添加kubernetes集群

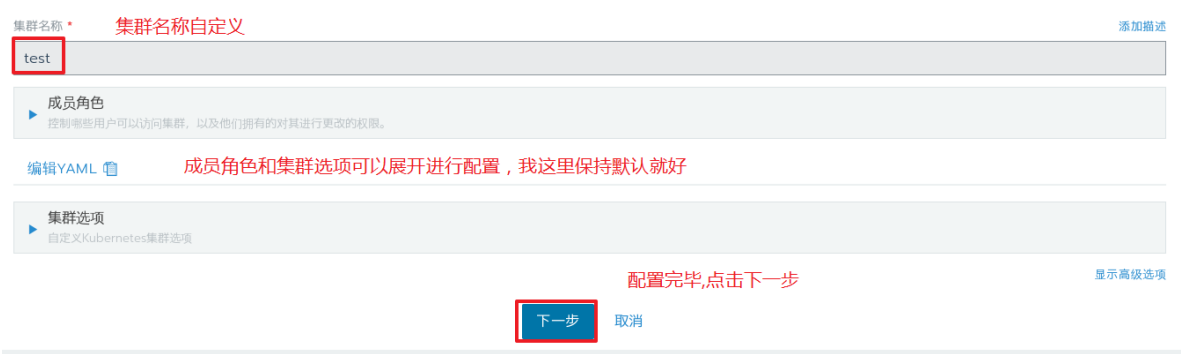
直接在主页面添加集群



## 选择集群类型



## 集群名称与选项自定义



# 定义集群主机选项

添加集群: test

编辑YAML

自定义主机运行命令

编辑主机选项将更新主机注册命令

## 1 主机选项

选择主机角色,端口放行请参考: <https://rancher.com/docs/rancher/v2.x/en/installation/references/>

角色选择(一台主机可以对应多个角色,每个集群至少需要一个Etcd角色、一个Control角色、一个Worker角色)

☒ Etcd

☒ Control

☒ Worker

主机地址

我们是单节点环境, 所以三个角色都要打勾

为主机配置公网地址和内网地址, 如果为VPC网络的云服务器, 如果不指定公网地址节点将无法获取到对应公网IP。

公网地址

例如: 1.2.3.4

内网地址

192.168.122.12

节点名称

(可选)自定义节点显示的名称, 不显示实际的主机名

本机IP地址, 本环境没有公网IP

node1

自定义的节点名称

## 安装rancher-agent

主机标签

应用于节点的可选标签

+ 添加标签

## 2 复制以下命令在主机的SSH终端运行。

```
sudo docker run -d --privileged --restart=unless-stopped --net=host -v /etc/kubernetes:/etc/kubernetes -v /var/run:/var/run rancher/rancher-agent:v2.2.8 --server https://192.168.122.12 --token 66bknpgf44ghdfxsx7rmsmhdf2ggpkvpbz6mr8cfxd8s2684mms2l2 --ca-checksum 68fae84b07cb71dd7dfd3c1edb86d2972049bfb93011e65f2c3ba39a48e2f4ee --node-name node1 --internal-address 192.168.122.12 --etcd --controlplane --worker
```

复制此命令在主机上执行  
执行完后点完成

完成

如果有多节点, 都要执行

## 通过下面命令安装rancher-agent后点完成

```
[root@rancher ~]# sudo docker run -d --privileged --restart=unless-stopped --net=host -v /etc/kubernetes:/etc/kubernetes -v /var/run:/var/run rancher/rancher-agent:v2.2.8 --server https://192.168.122.12 --token 66bknpgf44ghdfxsx7rmsmhdf2ggpkvpbz6mr8cfxd8s2684mms2l2 --ca-checksum 68fae84b07cb71dd7dfd3c1edb86d2972049bfb93011e65f2c3ba39a48e2f4ee --node-name node1 --internal-address 192.168.122.12 --etcd --controlplane --worker
Unable to find image 'rancher/rancher-agent:v2.2.8' locally
```

```
v2.2.8: Pulling from rancher/rancher-agent
35c102085707: Already exists
251f5509d51d: Already exists
8e829fe70a46: Already exists
6001e1789921: Already exists
357cc14e23eb: Pull complete
1a45097802d6: Pull complete
767c96a0eb35: Pull complete
2d7f44528c06: Pull complete
03c89d0b79d9: Pull complete
9cf93329bea9: Pull complete
Digest:
sha256:6c1e84e208b87912683d1c68c80476d48150480a7f8819e8931c06
3492621d70
Status: Downloaded newer image for rancher/rancher-
agent:v2.2.8
2a6d16e36a5fc735df0e0a03089ecbdab9e464750ef6a638ce9b470466328
372
```

2 复制以下命令在主机SSH终端运行。

```
sudo docker run -d --privileged --restart=unless-stopped --net=host -v /etc/kubernetes:/etc/kubernetes -v /var/run:/var/run rancher/rancher-agent:v2.2.8 --server https://192.168.122.12 --token
66bknpqf44ghdfxsx7rmsmhdf2ggpkvpbz6mr8cfxd8s2684mms2lz --ca-checksum
68fae84b07cb71dd7dfd3c1edb86d2972049bf93011e65f2c3ba39a48e2f4ee --node-name node1 --internal-address 192.168.122.12
--etcd --controlplane --worker
```

1台新主机注册成功

执行完命令后就注册好了1台新主机

完成

2.2.8 帮助&文档 论坛 Slack 中文社区 微信 提交问题

简体中文 下载 Rancher



全局 集群 多集群应用 用户 系统设置 安全 工具

添加集群

集群列表

删除 自

搜索

状态	集群名称	供应商	主机数	处理器	内存
Active	test	自定义 v1.14.6	1	0.4/4 Cores 10%	0.1/19 GiB 5%

过一段时间后，等待为Active状态

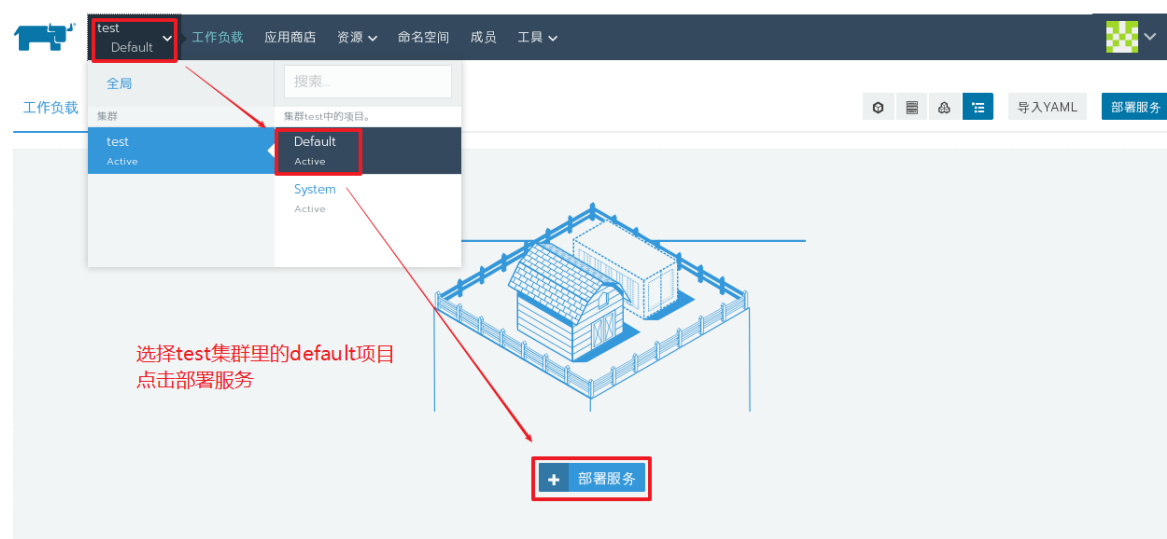




我们在rancher中分别部署一个nginx与mariadb应用来体验一下容器时代的快速部署

## rancher部署nginx示例应用

### 集群中部署服务



### 自定义deployment部署

test

Default

工作负载

应用商店

资源

命名空间

成员

工具

### 部署工作负载

名称 \*

nginx-text

自定义名称

添加描述

类型

部署

副本数

1

↑Pod

Docker镜像 \*

nginx:1.13-alpine

镜像名称

▼

命名空间 \*

default

我这里为默认命名空间

创建新的命名空间

端口映射

容器端口 \*

80

协议

TCP

网络模式

NodePort(所有主机端口均可访问)

主机监听端口 \*

30000

+

添加规则

容器内80端口使用NodePort映射到主机的30000端口（此端口可以随机）

## 启动

环境变量

设置容器可见的环境变量，包括从其他资源(如密文等)注入的值。

主机调度

配置Pod对应的主机调度规则。

健康检查

周期性向容器发出请求，以检测其健康状态。默认设置下，readiness和liveness使用相同的配置参数。对于应用初始化较长的容器，需要增加就绪检测时间。

数据卷

持久化及共享数据并与独立容器的生命周期分离。

缩放/升级策略

配置升级过程中替换Pod的策略。

在这里可以对pod进行自定义,但要求会对kubernetes的pod熟悉才能配置  
在这里我们直接保持默认

启动

取消

## 状态验证

test

Default

工作负载

应用商店

资源

命名空间

成员

工具

工作负载

负载均衡

服务发现

PVC

流水线

重新部署

暂停编排

下载YAML

删除

搜索

状态

名称

镜像

Pod副本数

命名空间: default

nginx-text

30000/tcp

nginx:1.13-alpine

1↑Pod / 创建时间: a few seconds ago

1

部署OK，我们使用访问节点的30000端口就可以访问到此nginx应用了

工作负载: nginx-text

Active

默认default命名空间

镜像

pod控制器类型

命名空间: default

镜像名: nginx:113-alpine

类型: Deployment

访问端口: 30000/tcp

Pod配置副本数: 1  
Pod可用副本数: 1

创建时间: 3:09 PM

NodePort类型service端口pod副本数可以扩容或裁减

全部展开

Pods

此工作负载中的Pods

下载YAML

删除

状态

名称

镜像

主机

Running

nginx-text-5c456f5dc-mg5wf

nginx:113-alpine  
10.42.0.7 / 创建时间: 7 hours ago / 重启次数: 0

node1  
192.168.122.12

# 客户端浏览器验证

192.168.122.12:30000

110%

访问成功

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](#). Commercial support is available at [nginx.com](#).

Thank you for using nginx.

# rancher应用商店部署mariadb示例

## 启动应用商店

test

Default

工作负载

应用商店

资源

命名空间

成员

工具

应用商店

管理

启动

点击应用商店  
启动应用商店

+ 启动

# 查找mariadb应用


test  
Default


工作负载应用商店资源命名空间成员工具


应用商店


刷新所有分类全部应用商店搜索

在这里搜索mariadb

  
官方  
artifactory-ha  
(来自 Library)  
Universal Repository Manager supporting all major packaging formats, build tools and CI servers.  
查看详情

  
cert-manager  
(来自 Library)  
A Helm chart for cert-manager  
查看详情


  
chartmuseum  
(来自 Library)  
Helm Chart Repository with support for Amazon S3 and Google Cloud Storage  
查看详情

  
datadog  
(来自 Library)  
DataDog Agent  
查看详情


## 配置应用参数

应用商店

刷新所有分类全部应用商店mariadb

  
mariadb  
(来自 Library)  
Fast, reliable, scalable, and easy to use open-source relational database system. MariaDB Server is intended for mission-critical, heavily-loaded production systems.  
查看详情

点击查看详情

  
MariaDB  
MariaDB is one of the most popular database servers in the world. It's made by the original developers of MySQL and guaranteed to stay open source. Notable users include Wikipedia, Facebook and Google.  
MariaDB is developed as open source software and as a relational database it provides an SQL interface for accessing data. The latest versions of MariaDB also include GIS and JSON features.

全部展开

详细描述  
应用程序信息和用户guid

配置选项  
粘贴和读取操作要求应否为yaml/yaml格式

名称 \*  
mariadb 名称可自定义

添加描述 模板版本  
5.0.9 模板版本,当前这已经是最新模板版本,保持默认

选择模板版本部署

NAMESPACE

NAMESPACE

命名空间 \*

使用现有的名称空间

mariadb命名空间保持默认

编辑YAML

CONTAINER IMAGES

Use Default Image

☒是☐否

使用默认镜像

Use default Docker image

MASTER SETTINGS

MariaDB Database User \*

adminmariadb的用户，保持默认

Username of new user to create.

MariaDB Database Password

密码自定义一下

生成

Password for the new user, defaults to a random 10-character alphanumeric string if not set

MariaDB Database User \*

admin

Username of new user to create.

MariaDB Database Password

生成

Password for the new user, defaults to a random 10-character alphanumeric string if not set

MariaDB Database Name \*

my\_database库名，保持默认

Name of database to create

Enable Persistent Volume for MariaDB Master Pod \*

☐是☒否

主从复制,默认不打开主pod的持久卷,我这里没有配置过持久存储卷

Enable persistent volume for MariaDB Master Pod

REPLICA SETTINGS

Enable MariaDB Replication \*

☒是☐否

打开主从复制功能

Enable MariaDB Replication

MariaDB Replica Count \*

2pod副本数为2

Desired number of slave replicas

Enable Persistent Volume for MariaDB Slave \*

☐是☒否

也不打开从pod的持久卷

Enable persistent volume for MariaDB Slave

SERVICES AND LOAD BALANCING

MariaDB Service Type \*

NodePortservice类型为NodePort端口不填,使用随机端口

MariaDB KBs Service type

Service NodePort number

NodePort port number(to set explicitly, choose port between 30000-32767)

Service NodePort number

NodePort port number(to set explicitly, choose port between 30000-32767)

METRICS SETTINGS

Enable Metrics \*

☐是☒否

暂不打开prometheus监控

Start a side-car prometheus exporter

预览

启动

取消

五

# 启动部署



test

Default

工作负载

应用商店


资源

命名空间

成员

工具

## 应用商店



mariadb

已经是最新版本 (5.0.9)

Deploying

30352/tcp, 30754/tcp

2

部署中, 请等待



test

Default

工作负载

应用商店

资源

命名空间

成员

工具

## 应用商店



mariadb

已经是最新版本 (5.0.9)

Active

30352/tcp, 30754/tcp

3

active

## 简单验证

使用下面命令访问OK

```
[root@rancher ~]# mysql -h 192.168.122.12 -u admin -p123 -P 30352
```

```
[root@rancher ~]# mysql -h 192.168.122.12 -u admin -p123 -P 30754
```