Rancher 快速上手指南（虚拟机篇）

前言

云舒网络携手 Rancher Labs 推出【Rancher | 实战群】，在线为您分享 Docker 技术干货，更有往期回顾精选期刊等你拿！

本群汇集了 Rancher 中国最强技术精英团队及业内技术派高人，宗旨是为了大家拥有更专业的平台交流 Rancher 实战技术，实时与 Rancher 创始团队面对面！同时欢迎各位分享自己的经验、疑难问题，我们将定期邀请分享嘉宾做各类话题分享及回顾，共同实践研究 Docker 容器生态圈。

对 Rancher 和 Docker 技术感兴趣、或对本文中细节需继续探讨的朋友，欢迎加入本群参与讨论！

加群方法：

1.关注【云舒网络】公众号

2.留言”我要加群“

注：因版本更新，文中 UI 界面与最新版本略有差异。

通过一个您已经熟悉的任何一种主流的发行版 Linux 虚拟机，就可以开始一个快速简单的 Rancher 测试体验。

建议虚拟机的规格：1vcpu，不小于 4GB 内存，一块能够连通互联网的网卡。本文编写的测试机是 AWS 虚拟机上的 Amazon Linux AMI，对 CentOS/RHEL 有直接参考意义。

Linux 主机准备

安装和运行 Docker 命令和服务，这基本是 Rancher 对于操作系统的最小化需求了。如果您还不太熟悉 Linux 或者 Docker 可以参考以下文档：

- Ubuntu 用户参考文档:https://docs.docker.com/engine/installation/ubuntulinux/
- CentOS/RHEL 用户参考文档:http://www.dedoimedo.com/computers/docker-guide.html

Docker 命令和服务安装好之后需要确认：

```
#确认 docker 的版本，下面是 centos 的输出
[ec2-user@ip-172-31-30-38 ~]$ sudo docker version
Client:
 Version:      1.9.1
 API version:  1.21
 Go version:   go1.4.2
```

```
 Git commit:     a34a1d5/1.9.1
 Built:
 OS/Arch:        linux/amd64
Server:
 Version:        1.9.1
 API version:    1.21
 Go version:     go1.4.2
 Git commit:     a34a1d5/1.9.1
 Built:
 OS/Arch:        linux/amd64
#确认 docker 服务已经启动并且运行，下面是以 centos 为例
[ec2-user@ip-172-31-30-38 ~]$ sudo service docker status
docker (pid  7652) is running...
```

启动 Rancher 服务器

Rancher 服务器是一个 Docker image，所以其软件本身不需要安装，只需执行 Docker 命令下载并且运行 Rancher 服务器的镜像即可。

```
[ec2-user@ip-172-31-30-38 ~]$ sudo docker run -d --restart=always -p 8080:8080 rancher/server
```

```
Unable to find image 'rancher/server:latest' locally
latest: Pulling from rancher/server
0bf056161913: Pull complete
1796d1c62d0c: Pull complete
e24428725dd6: Pull complete
89d5d8e8bafb: Pull complete
a31a85515ea3: Pull complete
c2fd2bef635f: Pull complete
cb545eb6ebd1: Pull complete
7beaeed203e7: Pull complete
f483a41462cd: Pull complete
2fd8dc138841: Pull complete
a4e1df2cafae: Pull complete
5f632b46feff: Pull complete
a4ff409fd1b0: Pull complete
8713e0a3f956: Pull complete
7f6c235d968a: Verifying Checksum
c074ec496974: Download complete
390a2453f500: Download complete
```

```
c7f9c84ef74a: Download complete
Status: Downloaded newer image for rancher/server:latest
docker.io/rancher/server: this image was pulled from a legacy registry.  Important: This registry version will not be supported in future versions of docker.
7c41a0a1a9c79842bca53c19e4ec106b0c2dc6469baec6077a40405f80b26963
[ec2-user@ip-172-31-30-38 ~]$
```

命令行参数解释：

- docker run [OPTIONS] IMAGE [COMMAND] [ARG...] 运行一个 docker 容器
- -d 在后台运行 docker 容器，并且打印出它的容器 ID （Run container in background and print container ID）
- --restart=always 当容器存在时所应用的重启策略，总是重启。
- -p 8080:65432 容器端口在虚拟机本机上使用 8080 端口，Rancher 服务器的 UI 对外服务的端口是 8080，如果您的服务器是远程的服务器，还需要考虑到你的测试客户机和虚拟机之间的防火墙策略，确保所使用的 Rancer 服务器 UI 对外服务端口不是防火墙阻止的端口。
- rancher/server 这里声明让 docker 去 docker hub 下载并且运行名称为 rancher/server 的 docker 镜像到本地。

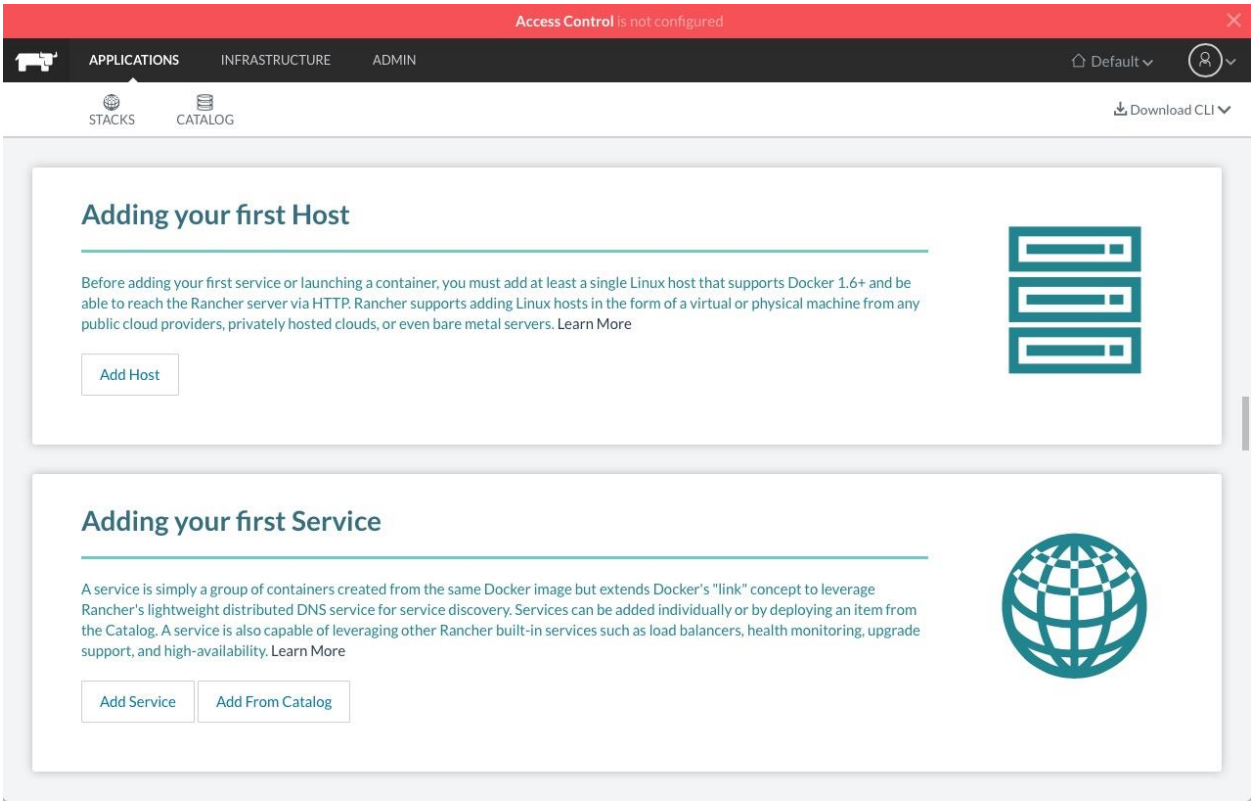
#检查 docker 已经正确下载了 rancher/server 镜像到本地

```
[ec2-user@ip-172-31-30-38 ~]$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
VIRTUAL SIZE
rancher/server      latest              25c20134881a        3 days ago
845 MB
<none>              <none>              8713e0a3f956        4 days ago
473.9 MB
[ec2-user@ip-172-31-30-38 ~]$
```

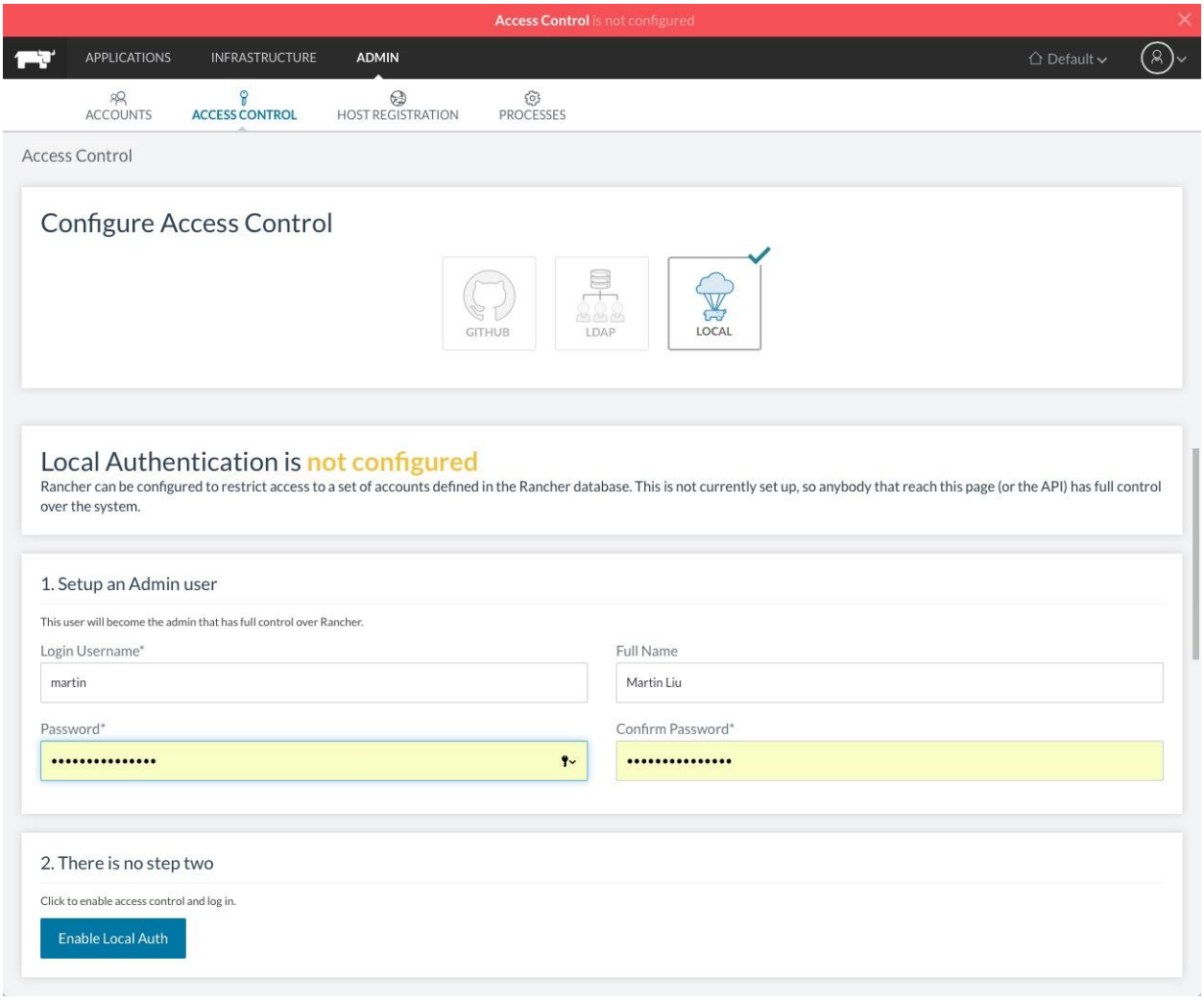
#检查 Rancher 服务器容器已经正常运行

```
[ec2-user@ip-172-31-30-38 ~]$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND
CREATED             STATUS              PORTS
NAMES
7c41a0a1a9c7        rancher/server      "/usr/bin/s6-svscan /"   3 minutes ago
Up 3 minutes        3306/tcp, 8080/tcp, 0.0.0.0:8080->8080/tcp   tiny_kalam
[ec2-user@ip-172-31-30-38 ~]$
```

国内的服务器的下载速度可能会比较慢，需要等待大约 30 分钟左右。用浏览器打开 Rancher 服务器 UI 界面，并且确认是否可以正常登陆。



首次访问还没有配置访问权限的页面如上图所示。为了安全起见，点击上面的 Access Control 来设置一个本地账号和密码。



如上图，使用设置的账号和密码再次登陆确认，配置的信息正确，继续下面的测试。

还可以在命令行开启 Rancher 容器运行日志监控，如下所示：

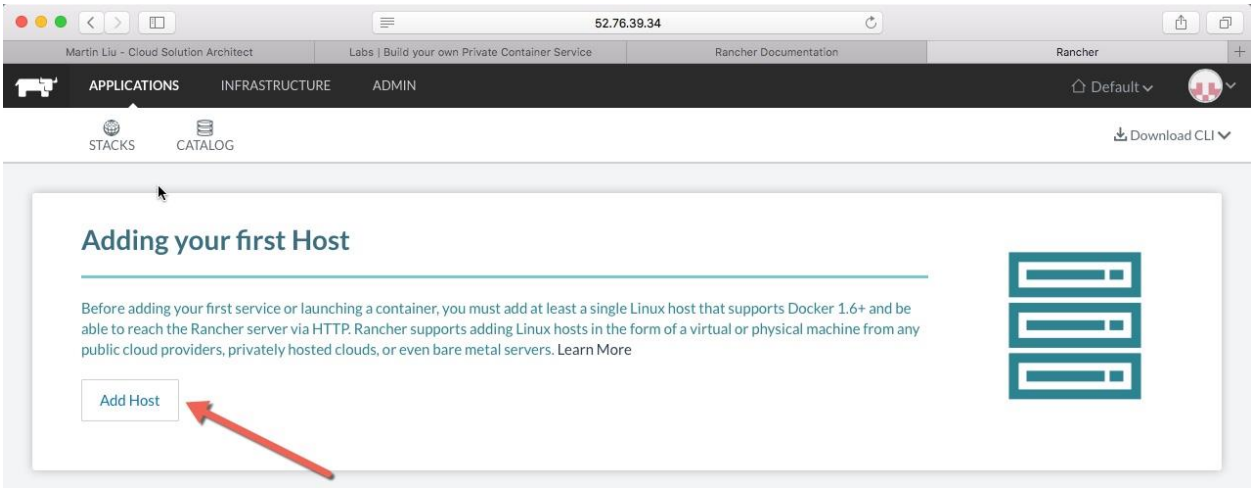
```
#替换下面红字部分的 docker 容器 id，你的 id 可以从 docker ps 命令查到

[ec2-user@ip-172-31-21-99 ~]$ sudo docker logs -f 988003c02bcd
```

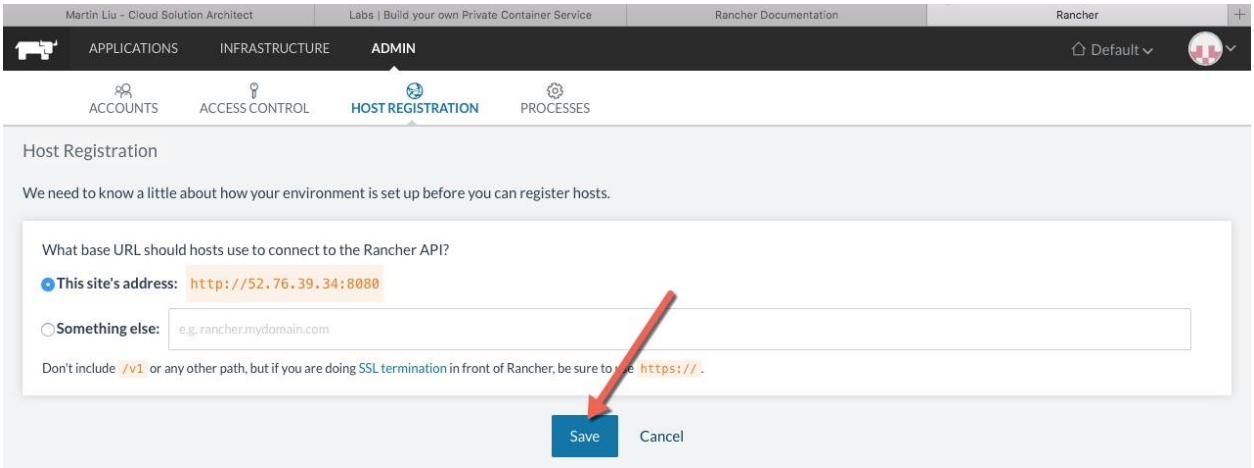
到目前，你已经完成了 Rancher 服务器的部署和基础配置。

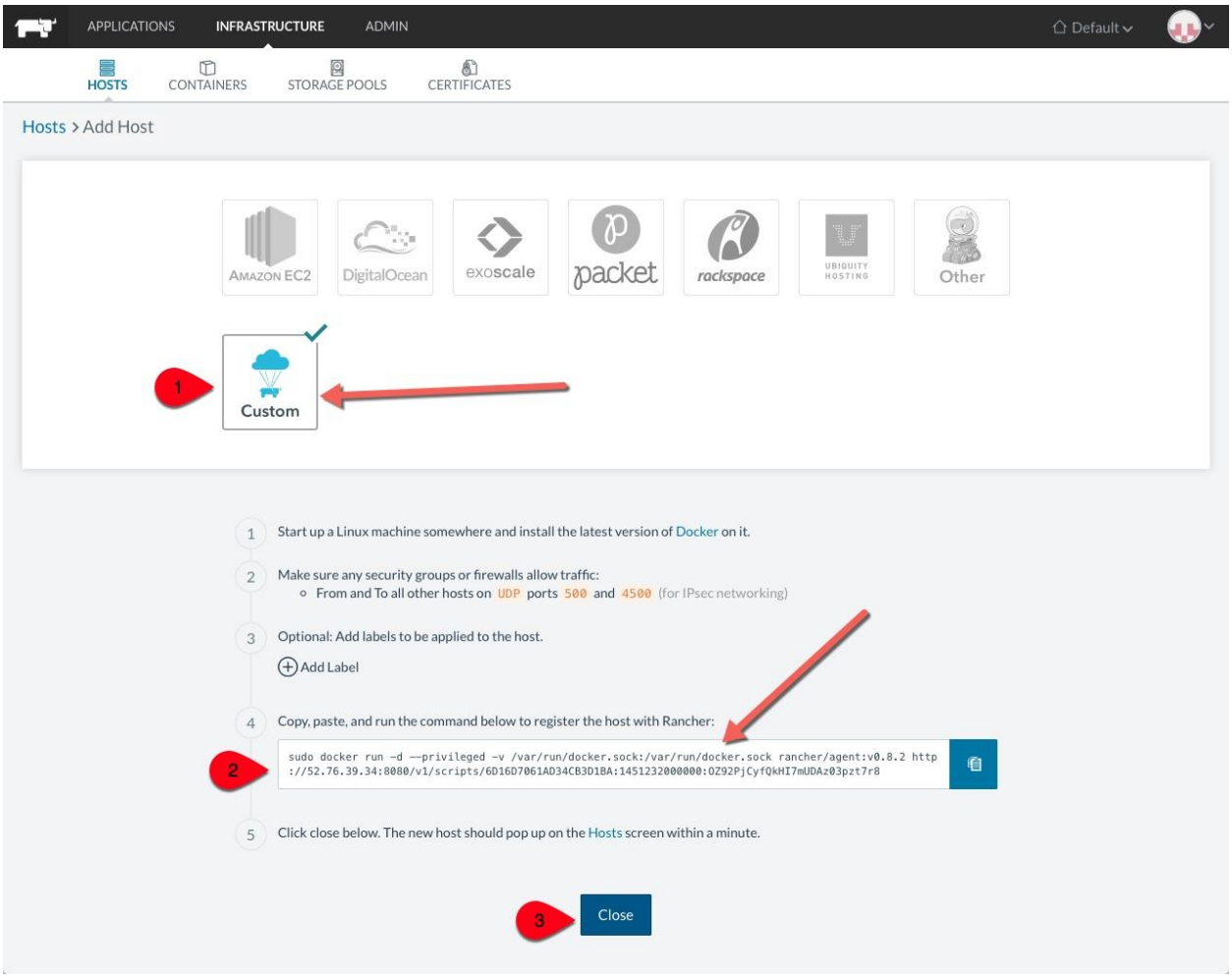
添加主机

主机是 Rancher 的工作节点，类似服务器虚拟化的 Hypervisor；在本实验中我们在运行 Rancher 服务器容器的管理节点上（虚拟机）做 All-in-One 的测试，因此下面把测试所用的虚拟机添加为运行工作负载容器的工作主机。

点击登陆以后界面上的 add host 按钮。







1. 点击 Customer，默认的选项是 DigtalOcean

2. 复制文本框中的代码，在虚拟机的命令里面粘贴运行。

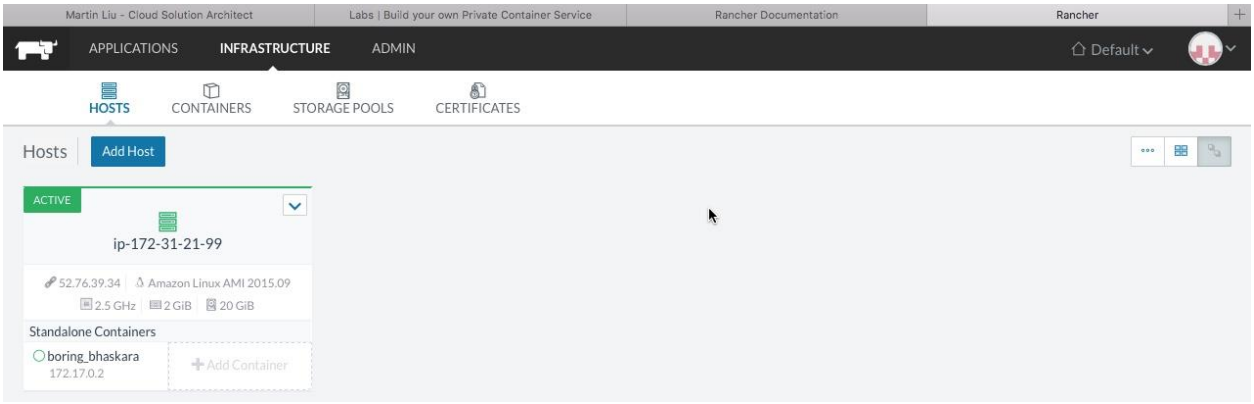
3. 点击 Close 按钮

运行以上命令之后，在命令行可以用 docker ps 命令再次查看容器运行的状态，如下所示：

```
[ec2-user@ip-172-31-21-99 ~]$ sudo docker ps
CONTAINER ID IMAGE COMMAND
CREATED STATUS PORTS
```

```
NAMES
6ab206a64a68        rancher/agent:v0.8.2   "/run.sh run"          2 minutes ago
Up 2 minutes                                                      rancher-agent
988003c02bcd        rancher/server         "/usr/bin/s6-svscan /"  28 minutes ago
Up 28 minutes       0.0.0.0:8080->8080/tcp, 3306/tcp              boring_bhaskara
[ec2-user@ip-172-31-21-99 ~]$
```

我们可以看到多了一个名字为 rancher/agent 的容器。过几分钟之后在回到 Web 控制台中。查看 Host 添加之后的结果。



如上图所示，我们看到了一台活动的主机，并且该主机上运行着一个容器，就是 Rancher 服务器自己。

可以继续重用上面的命令，来把其他的测试虚拟机也添加为 Host，如下所示：

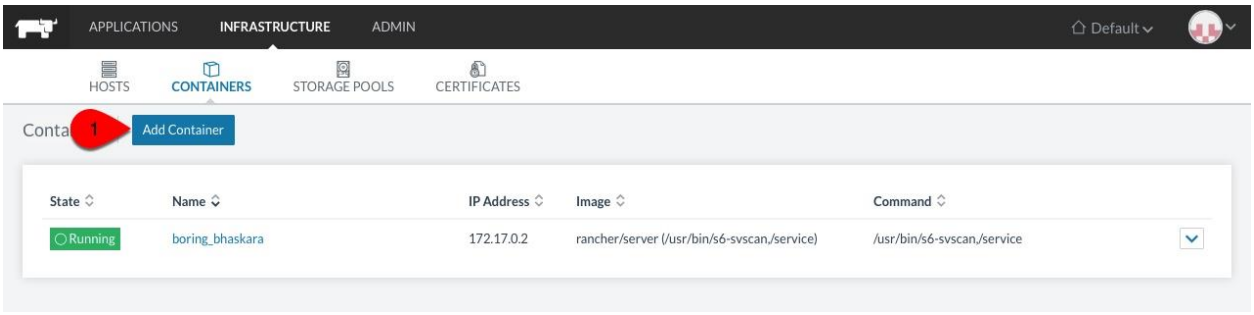
```
[ec2-user@ip-172-31-21-99 ~]$ sudo docker run  -e CATTLE_AGENT_IP=X.X.X.X -d --
privileged  -v  /var/run/docker.sock:/var/run/docker.sock  rancher/agent:v0.8.2  http://
X.X.X.X:8080/v1/scripts/6D16D7061AD34CB3D1BA:1451232000000:OZ92PjCyfQkHI7
mUDAz03pzt7r8
```

替换以上红色部分，主要是 IP 地址需要修改为运行 Rancher 服务器容器的服务器的 IP，

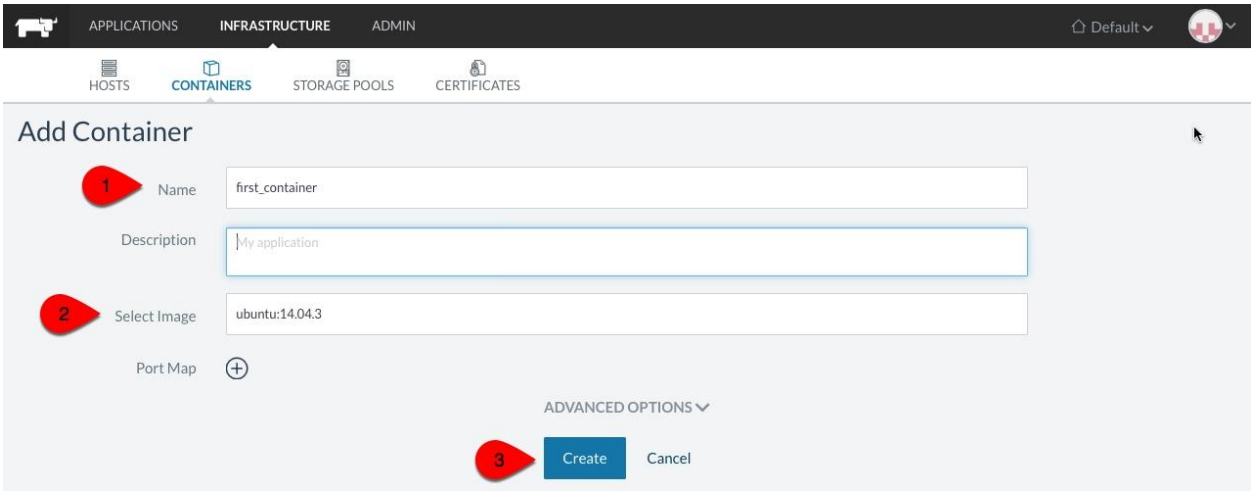
当然，运行此条命令的前提条件如第一节所述，与 Rancher 服务器的准备工作相同。

用 Web UI 运行容器

现在来通过图形界面来运行第一个容器。通过简单地几下鼠标点击即可为完成一个容器的运行动作。



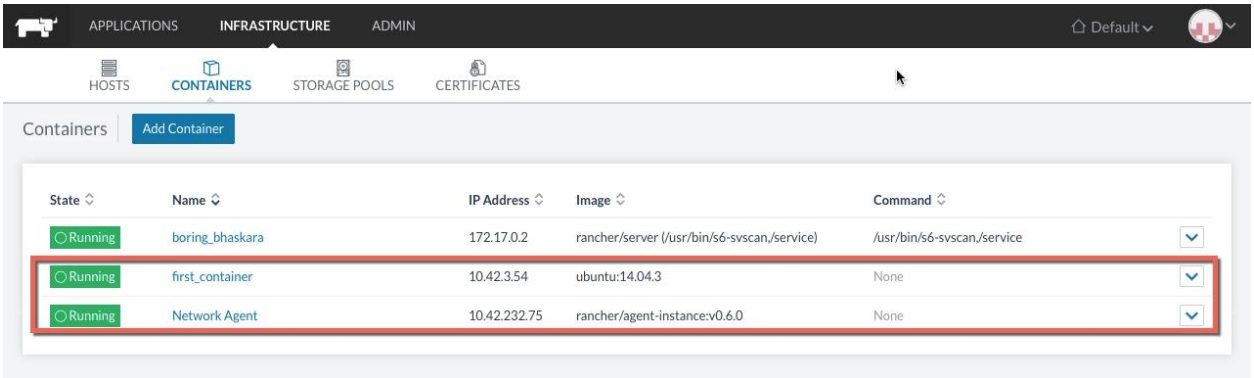
1. 点击 Add Container 按钮增加新的容器



1. 输入 first_container 作为容器的名称

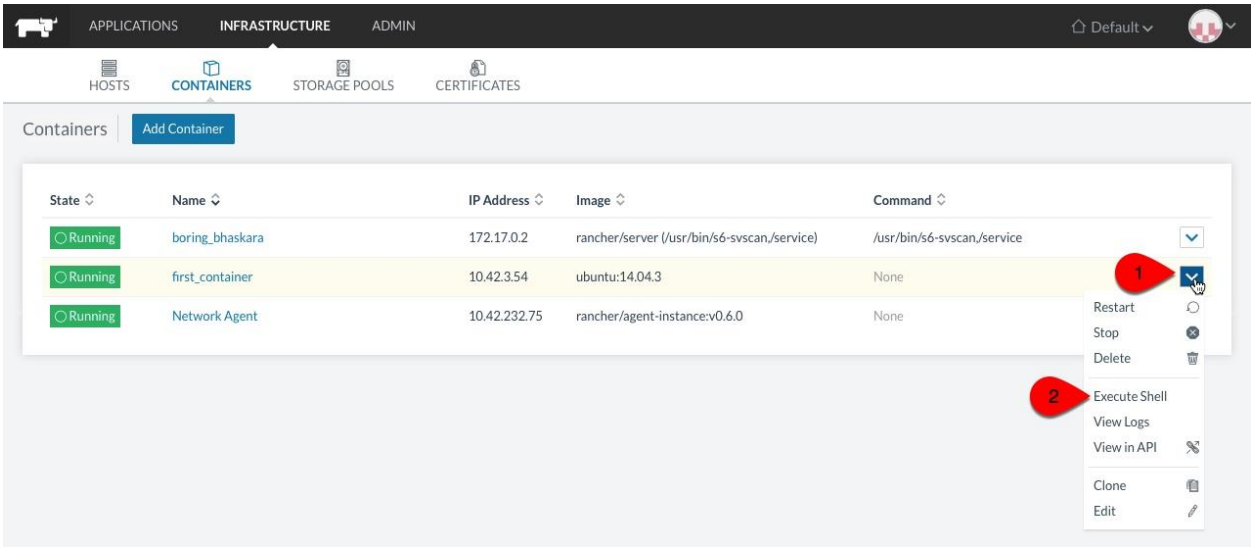
2. 使用默认的 Ubuntu14 镜像

3. 点击 create 创建按钮



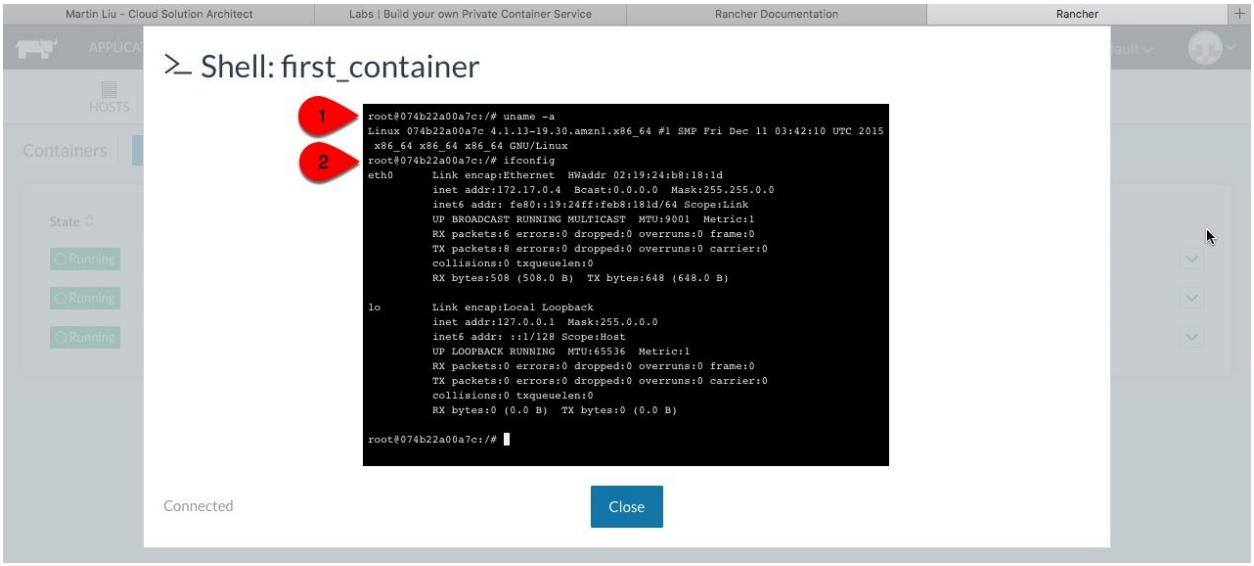
1. 经过大约几秒钟，出现了两个运行的容器，firs_container 为本测试创建的容器，下面的那个 Network Agent 容器的作用是用户主机之间的网络通信。

2. 上图可以看出主机间会使用 Rancher 服务器所管理的 10.42.0.0 内网通信。



1. 在新建的 Ubuntu 容器的菜单上点击向下按钮

2. 选择在 Web 页面中运行 Shell 选项



1. 经过几秒钟的链接，网页上出现了 Ubuntu 容器的 root 命令行提示符，运行第一个命令 uname –a
2. 运行第二个命令行 ifconfig

到此为止你已经成功创建了新的容器，在命令行里面，如果你是 developer 的话，你已经可以开工了，或是部署测试代码，或者打造应用服务的 image。

用命令行运行容器

如果您偏爱 CLI 命令行，可以在 host 上执行容器的创建和管理动作，如下所示：

#下面命令为运行一个新的名为 second_container 的 Ubuntu14.04.2 容器，并在容器运行之后，直接进入该容器的命令行

[ec2-user@ip-172-31-21-99 ~]$ docker run -it --name=second_container ubuntu:14.04.2

Unable to find image 'ubuntu:14.04.2' locally

```
14.04.2: Pulling from library/ubuntu
Digest:
sha256:a1cec70421f71f00c8bdb0adf0226dc548ff5ba9699cbd5fa09acdb68df82a02
Status: Downloaded newer image for ubuntu:14.04.2
root@be607f589023:/#
#这里是第二个容器的命令行，下面执行 id 命令
root@be607f589023:/# id
uid=0(root) gid=0(root) groups=0(root)
#在该容器里做网络测试
root@be607f589023:/# ping rancher.com
PING rancher.com (104.24.18.49) 56(84) bytes of data.
64 bytes from 104.24.18.49: icmp_seq=1 ttl=59 time=1.75 ms
64 bytes from 104.24.18.49: icmp_seq=2 ttl=59 time=1.97 ms
^C
--- rancher.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.758/1.866/1.975/0.116 ms
#退出该容器命令行
root@be607f589023:/# exit
```

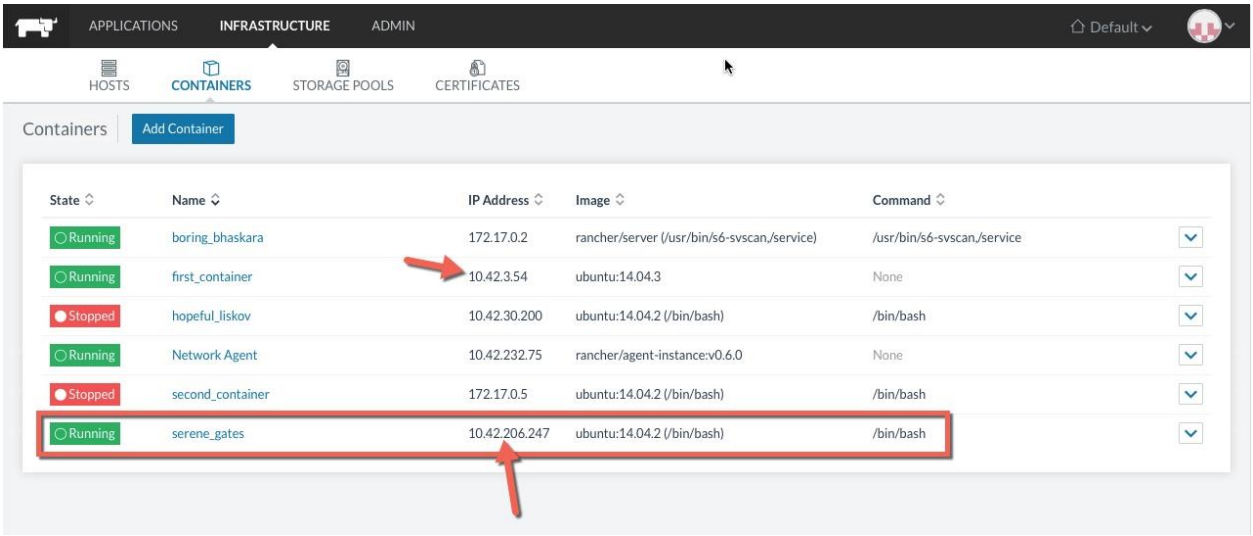
```
exit
#在 host 执行 docker 容器运行状态查看命令
[ec2-user@ip-172-31-21-99 ~]$ sudo docker ps
CONTAINER ID IMAGE COMMAND
CREATED STATUS PORTS
NAMES
074b22a00a7c ubuntu:14.04.3 "/bin/bash"
15 minutes ago Up 15 minutes
37695985-73e7-4e54-be90-870c86d4cef3
9132d138e896 rancher/agent-instance:v0.6.0 "/etc/init.d/agent-in" 15
minutes ago Up 15 minutes 0.0.0.0:500->500/udp, 0.0.0.0:4500->4500/udp
ce65d23e-2f55-4497-9ffe-38ae1bcedf9f
6ab206a64a68 rancher/agent:v0.8.2 "/run.sh run" 39
minutes ago Up 39 minutes
rancher-agent
988003c02bcd rancher/server "/usr/bin/s6-svscan /"
About an hour ago Up About an hour 0.0.0.0:8080->8080/tcp, 3306/tcp
boring_bhaskara
[ec2-user@ip-172-31-21-99 ~]$
```

以上测试结束后，可以在 Web UI 中查看该容器的状态，可以看到它已经处于 stop 的状态，它所使用网络地址与 Rancher 服务器容器同一个网段，这是默认执行 docker run 命令的默认结果；还可以在命令行制定使用 Rancher 服务器所管理的叠加的内部私有网络，命令行如下：

```
#注意一下 docker run 命令使用了 --label io.rancher.container.network=true 参数
[ec2-user@ip-172-31-21-99 ~]$ docker run -it --label io.rancher.container.network=true
ubuntu:14.04.2
#进入新容器的命令行之后查看网络地址
root@a12455b64a7b:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:ac:11:00:05
          inet addr:172.17.0.5  Bcast:0.0.0.0  Mask:255.255.0.0
          inet6 addr: fe80::42:acff:fe11:5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:9001  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:438 (438.0 B)  TX bytes:508 (508.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
```

```
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@a12455b64a7b:/# ping rancher.com
PING rancher.com (104.24.18.49) 56(84) bytes of data.
64 bytes from 104.24.18.49: icmp_seq=1 ttl=59 time=1.75 ms
64 bytes from 104.24.18.49: icmp_seq=2 ttl=59 time=1.95 ms
```

在不退出以上容器命令行的情况下，进入 Web UI 查看此容器的状态。



最下面一个是使用了内部叠加网的容器，它的 Ip 地址从 Web UI 中看和第一个 first_container 的 IP 为同一个网段。

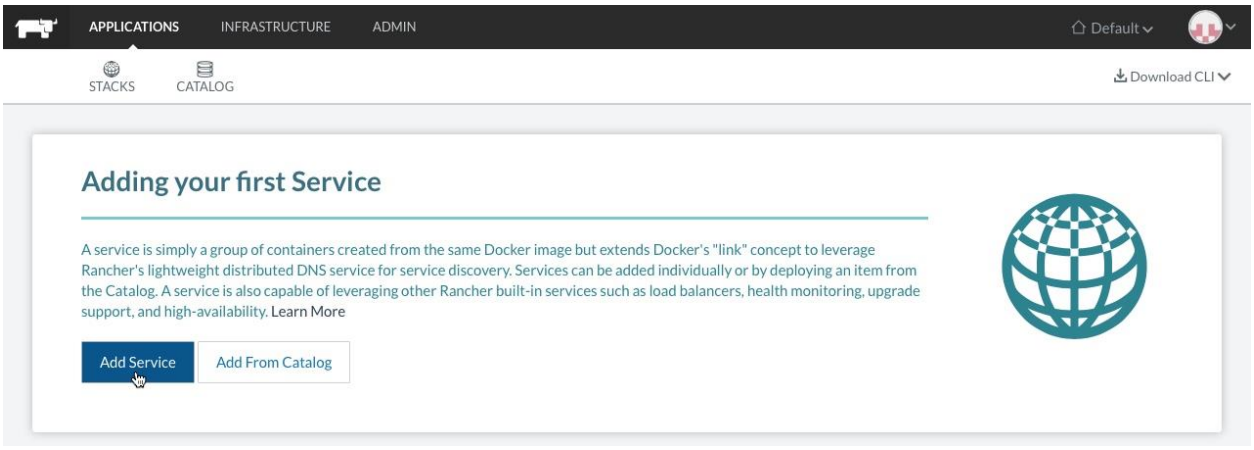
至此搞清楚了多个容器的运行和基本的网络概念之后，下面可以做更为复杂的测试。

用 Web UI 完成多层应用架构的部署

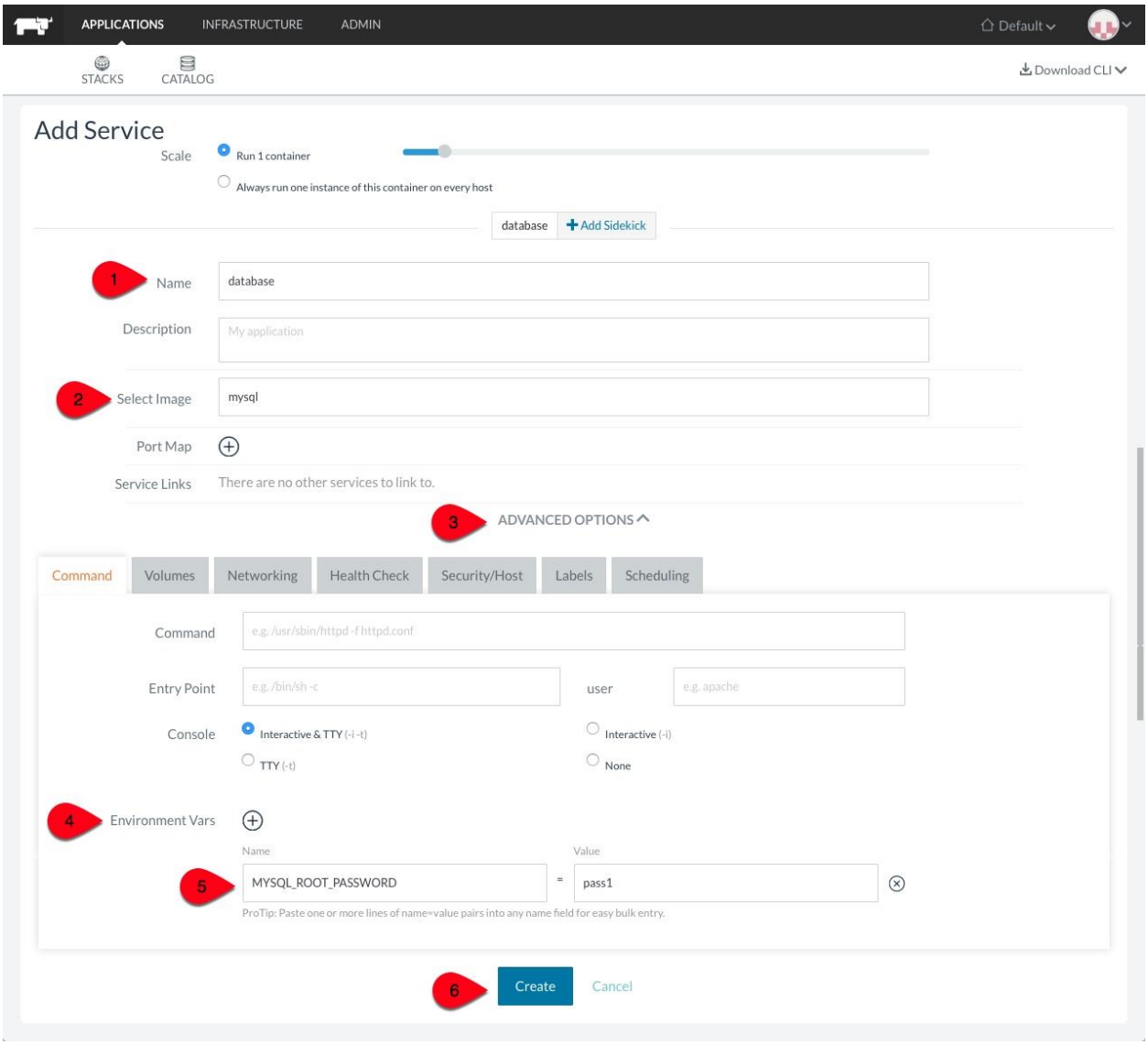
本测试需要完成 WordPress 应用的多层部署，期望的部署模式如下：

1. 前端负载均衡服务，一个 LB 用来接入来自互联网的流量
2. WordPress 应用层，有两个运行 WordPress 软件的容器组成
3. 数据库服务层，有一个 MySQL 服务器容器

具体的操作步骤如下所示。

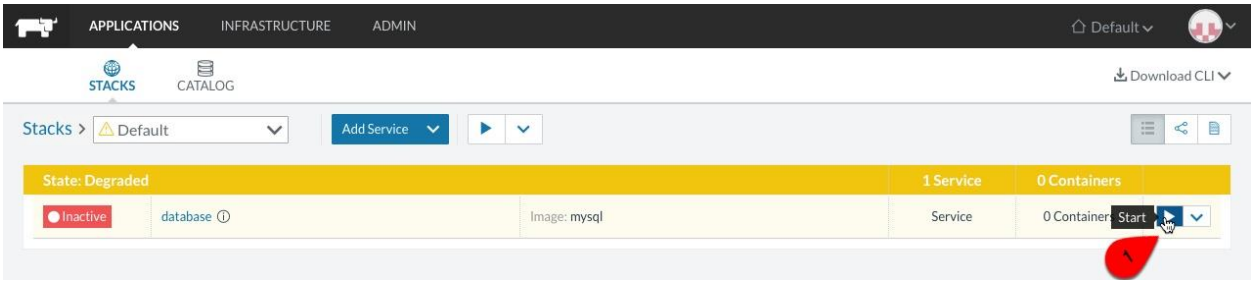


在登录后的首页，点击 Add Service 按钮开始创建整个堆栈。

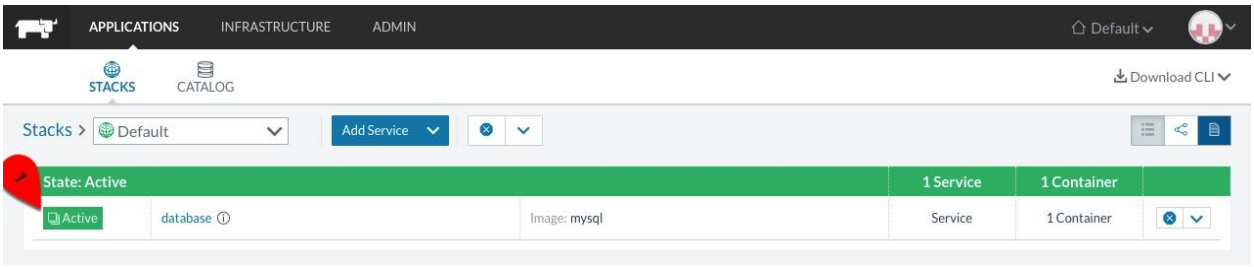


1. 选择 Add Service 增加 数据库服务，输入名称 database
2. 选择使用 mysql 镜像
3. 点击高级选项卡
4. 点击添加环境变量
5. 数据环境变量的内容，mysql 数据库的 root 密码为 pass1

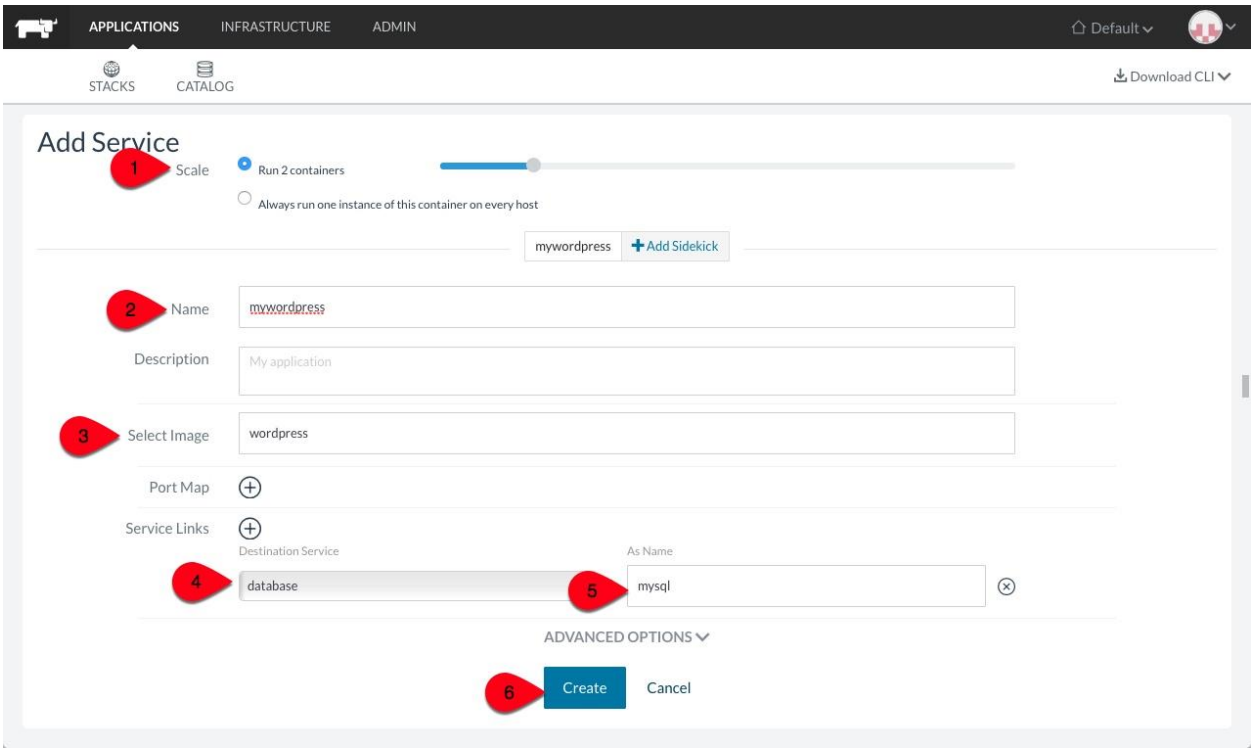
6. 点击创建此服务



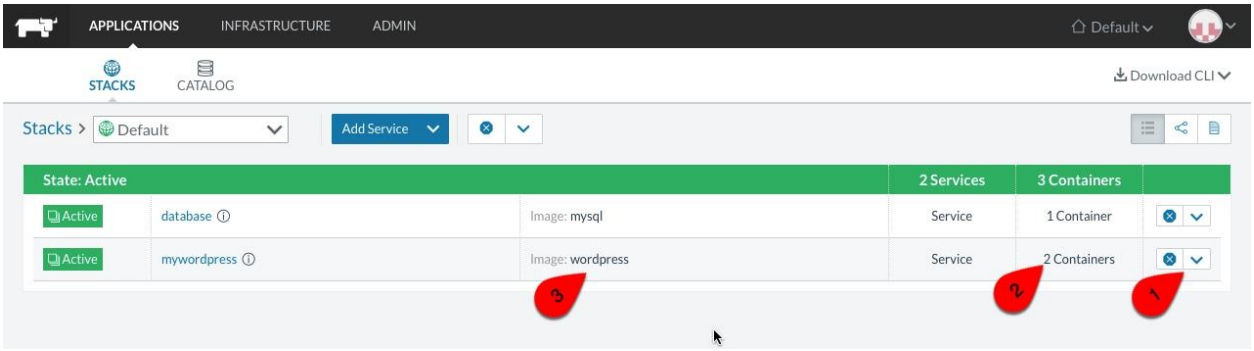
1. 创建后的服务默认为非活动状态，点击 Start 按钮，启动数据库服务



1. 启动之后的 mysql 数据库服务状态正常，点击上面的 Add Service 按钮添加服务



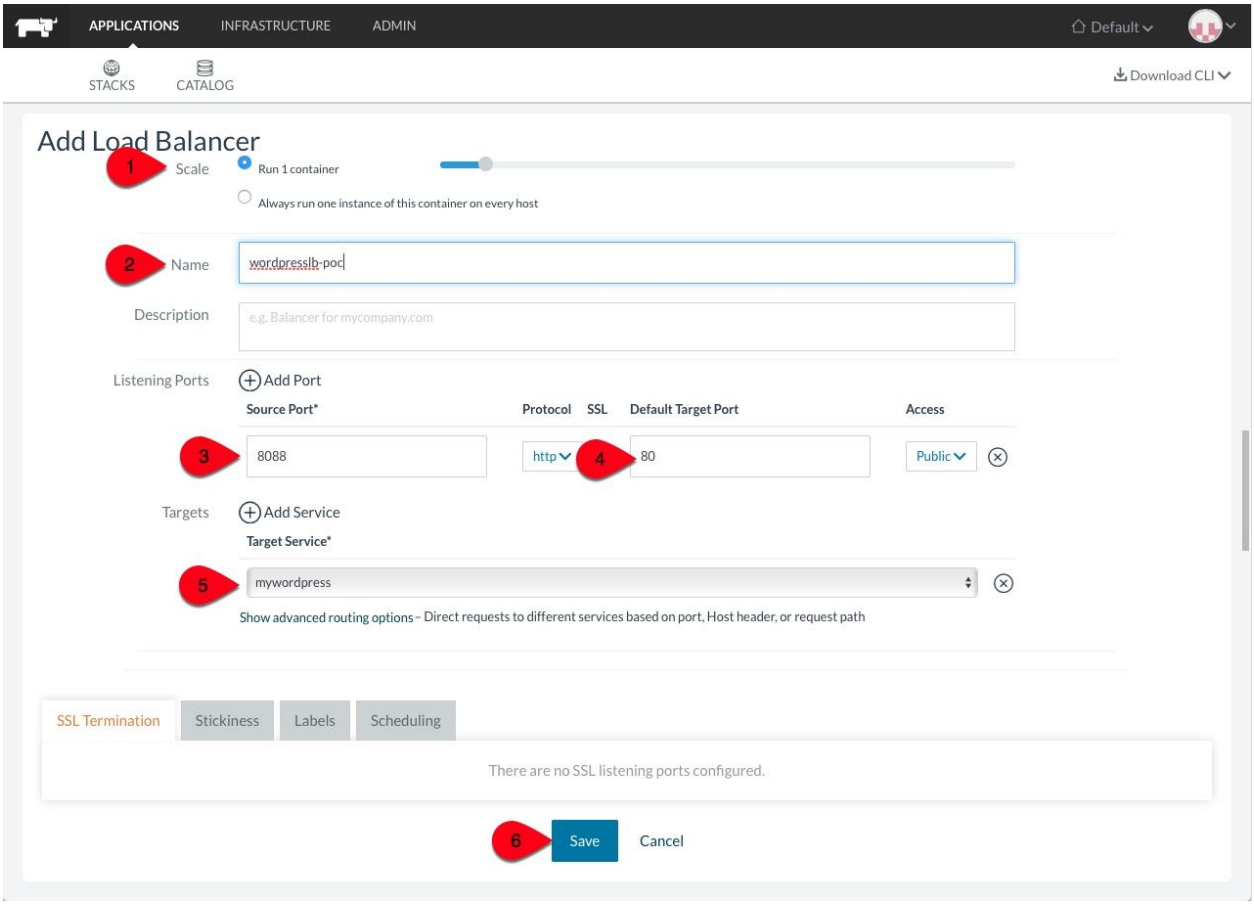
1. 拖动圆点，使本层服务的容器数量为 2
2. 输入名称为 mywordpress
3. 输入所需要使用的 Wordpress 镜像
4. 选择它所依赖的数据库服务
5. 输入名称 mysql
6. 点击创建按钮



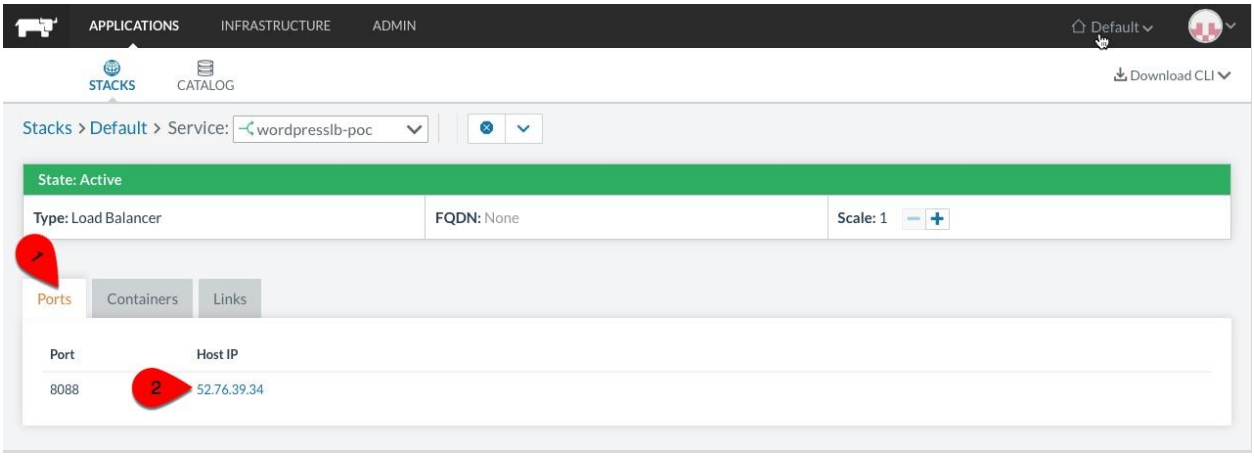
1. 点击菜单中的运行按钮，启动新建容器

2. 观察容器的数量为 2

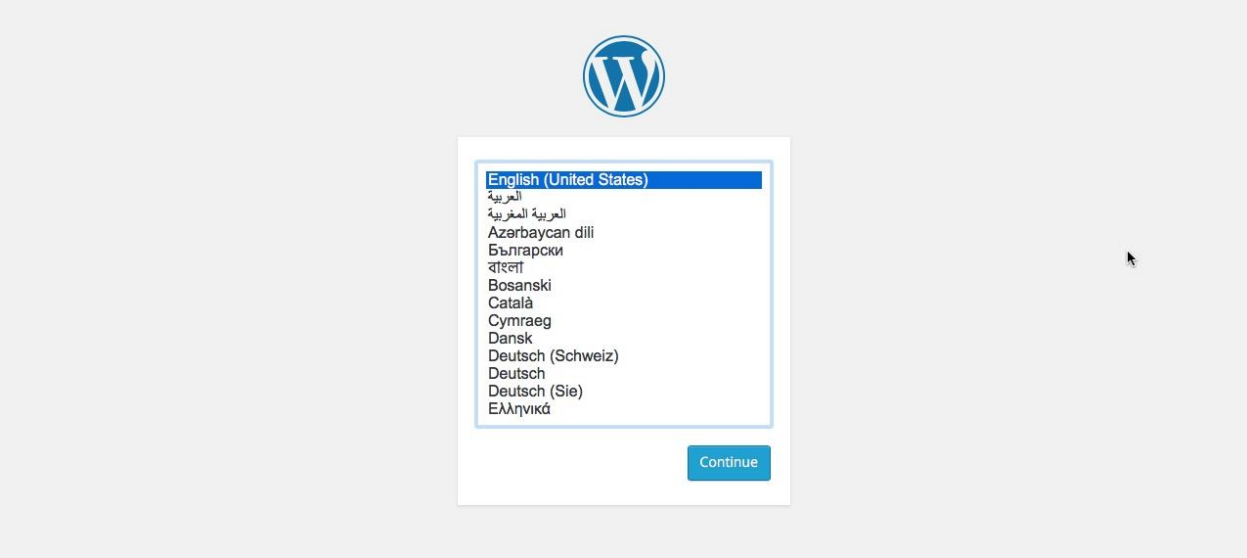
3. 观察容器所使用的镜像名称，点击 Add Service 下拉菜单，选择创建负载均衡器



1. 使用默认 1 为 LB 的数量
2. 输入 LB 的名称
3. 输入在 Host 上 LB 对外服务的端口为 8088
4. 输入 Wordpress 容器的服务端口 80
5. 选择对象服务，为 myWordpress
6. 点击 Save 创建此容器和相关配置



1. 运行创建好的 LB 容器，点击该容器，查看它的状态，点击 Ports 选项卡
2. 点击 Host IP，浏览器就会连接到 http://your_host_ip:port 打开负载均衡的服务网址



如上图所示 wordPress 的安装页面正常打开，可以继续完成 WordPress 的安装和配置。至此您已经顺利完成了多层应用的部署和搭建。

后记

如果您再次刷新浏览器，或者用新的窗口打开该网址，有可能再次进入此安装页面，这是由于您连接到了第二个为曾执行过安装程序的 WordPress 容器，手工把 wp-config.php 复制到该容器，再次刷新即可，看到安装好之后的页面。

如果您是用的是 AWS 的主机，或者其他云主机，默认情况下 8080，8088 这些服务端口是不通的，需要在使用前，先进入安全组管理，打开这些端口。

本文参考了 Rancher 官方文档：http://docs.rancher.com/rancher/quick-start-guide/

但是不包含 docker-compse 命令行工具 + yml 配置文件的创建方式，建议可以参考该

文档完成完整的测试。

本测试把 Rancher 的基础用法做了一个初级的尝试，希望对新手有所帮助。

Rancher 官方网站：http://rancher.com/