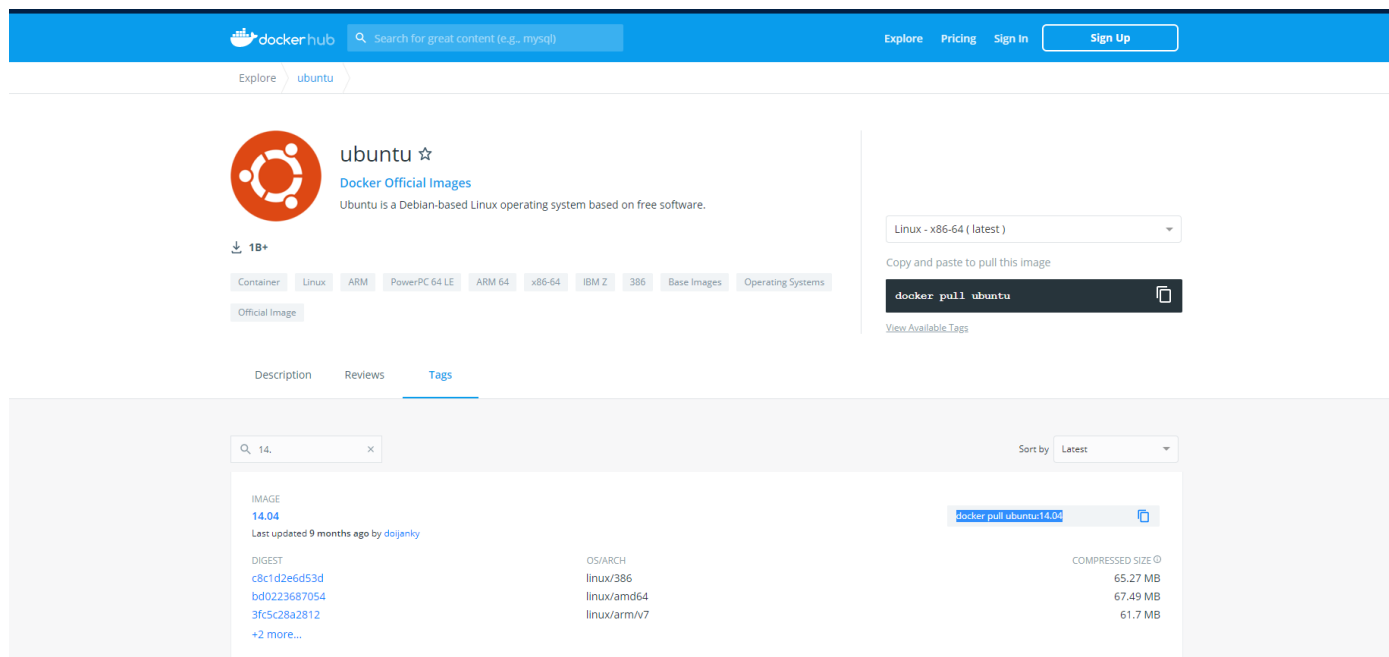


# 利用docker搭建MPI环境并进行MPI Reduce计算

## 一、打造镜像

### 1、拉取基础镜像

本文以ubuntu14.04作为OS系统，因此拉取ubuntu14.04 docker镜像作为基础镜像，在docker hub上搜索ubuntu并找到对应版本



直接运行 `docker pull ubuntu:14.04` 拉取镜像

### 2、运行镜像

拉取镜像之后，需要对基础镜像进行环境的配置，在此之前先进行换源

开启一个容器

```
docker run -dit --hostname origin --name origin ubuntu:14.04
```

-dit 使得该容器保持后台运行

进入该容器

```
docker exec -it origin bash
```

进行换源

备份

```
sudo cp /etc/apt/sources.list /etc/apt/sources.list.bak
```

打开文件

```
sudo vi /etc/apt/sources.list
```

把文件内容替换为以下内容

```
deb http://mirrors.aliyun.com/ubuntu/ trusty main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu/ trusty-security main restricted universe
multiverse
deb http://mirrors.aliyun.com/ubuntu/ trusty-updates main restricted universe
multiverse
deb http://mirrors.aliyun.com/ubuntu/ trusty-proposed main restricted universe
multiverse
deb http://mirrors.aliyun.com/ubuntu/ trusty-backports main restricted universe
multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ trusty main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ trusty-security main restricted universe
multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ trusty-updates main restricted universe
multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ trusty-proposed main restricted universe
multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ trusty-backports main restricted universe
multiverse
```

再执行update操作更新源

```
sudo apt-get update
```

### 3、配置mpi环境

运行命令安装MPI

```
sudo apt-get install libopenmpi-dev
sudo apt-get install openmpi-bin
```

安装后查看是否安装成功

```
mpirun --version
# 输出:
# mpirun (Open MPI) 1.6.5

# Report bugs to http://www.open-mpi.org/community/help/
```

有如下输出就是安装成功了，然后再运行安装openssh-server

```
sudo apt-get install openssh-server
```

安装后生成公钥密钥

```
ssh-keygen -t rsa
```

如果所有的容器都是一个镜像生成的，那么他们的公钥密钥一致，将公钥加入自己的authorized\_keys文件中来，这样同一镜像的容器可以互相ssh访问

```
root@host1:~/.ssh# cat id_rsa.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQACnJ9DhnC4kyZfhZi0uXUCApwWkMN7a+KgJ2TarlsV5NZGSUEL7MmpBsuz
2l4dtLJh+QwS0qQFbUeuAmYIcSMcDoJPr6aiHg1fUDyktkQ/ZyHYXA/LKf6MOceBVlusKq1bITqvo9WIX30U2Q
HsdDyqp6E/PQBi9Pf69gA3iUGkjQzltTlY+7/1N62iOTTIUHB81p5X910JYzgGZ6qO2U1CT5Hq/OY3xABmmrzBE
BgNdtZ8op70GYBNRjtjBqdMi7TB9xyOb2+nhc2Q1+jjnJ8RW9hlq/2gbwdtRxdX0QX3VAH6iry2j/IL42uM5DEw
IEhHrM8tdCR+NunysRNNsWql root@75201ce87a9b
root@host1:~/.ssh# vi authorized_keys
# 直接复制粘贴上面的公钥
```

记得为每个容器开启ssh服务，镜像开启的容器不会自动开启服务

```
service ssh start
```

## 4、创建新镜像

```
docker commit [-a (your name)] <container name or id> <image name>[:(tag)]
```

-a是表明作者名，后面两个顾名思义，本例中将镜像名命名为 mpi-ubuntu

```
docker commit -a werllen origin mpi-ubuntu:latest
```

罗列出本地镜像，看是否创建成功

```
docker images
```

## 二、运行MPI

## 1、运行镜像

```
docker run -dit --name mpi-host1 --hostname host1 mpi-ubuntu:latest
```

name是容器名，hostname是主机名

再开第二个

```
docker run -dit --name mpi-host2 --hostname host2 mpi-ubuntu:latest
```

## 2、配置hosts

打开各个容器的hosts文件

```
docker exec -i mpi-host1 bash #进入容器
sudo vi /etc/hosts #打开hosts文件
```

文本中可以看到最后一行为

```
172.17.0.2      host1
```

前面是ip地址（也是本容器的地址），后面是主机名（可以看作是ip的别名，可以用它来代替ip地址的书写）

同样mpi-host2容器也有，不过内容不同

```
172.17.0.3      host2
```

将这行声明写到host1的hosts文件中，当再host1里访问host2即可写为

```
ssh root@host2
```

## 3、打开ssh服务

利用镜像开启的容器并不会开启ssh服务，需要手动开启

```
service ssh start # 在每个容器中运行
```

然后来到主节点容器（host1），分别连接其他容器保存known\_hosts信息

```
ssh root@host2
# 第一次连接时会询问你如下
The authenticity of host 'host2 (172.17.0.3)' can't be established.
ECDSA key fingerprint is 58:41:1c:8e:05:4a:f0:23:82:41:67:06:0f:17:18:59.
Are you sure you want to continue connecting (yes/no)?
# 因为没有默认回答，所以应该输入yes
```

## 4、编译mpi代码并运行

编译行为要对所有服务器执行，要确保各个服务器同样的地方都有可执行文件

```
mpicc hellompi.c -o hellompi
```

```
mpirun -np 10 -host host1,host2 hellompi
```

也可以通过所有容器挂载同一本地文件夹来实现共享代码和可执行文件

即在开启镜像时，运行

```
docker run -dit --hostname host1 --name mpi-host1 -v ~/mpidir:/root/mpidir mpi-ubuntu:latest
```

## 5、快速开启脚本

可以通过脚本快捷创建容器和开启ssh服务

```
docker rm -f $(docker ps -a|grep mpi-host)
```

```
docker run -dit --hostname host1 --name mpi-host1 -v ~/mpidir:/root/mpidir mpi-ubuntu:lcmd
```

```
docker exec mpi-host1 service ssh start
```

```
echo ""
```

```
docker exec mpi-host1 tail -1 /etc/hosts
```

```
echo ""
```

```
docker run -dit --hostname host2 --name mpi-host2 -v ~/mpidir:/root/mpidir mpi-ubuntu:lcmd
```

```
docker exec mpi-host2 service ssh start
```

```
echo ""
```

```
docker exec mpi-host2 tail -1 /etc/hosts
```

```
echo ""
```

```
docker run -dit --hostname host3 --name mpi-host3 -v ~/mpidir:/root/mpidir mpi-ubuntu:lcmd
```

```
docker exec mpi-host3 service ssh start
```

```
echo ""
```

```
docker exec mpi-host3 tail -1 /etc/hosts
```

```
echo ""
```

```
docker exec -it mpi-host1 bash
```

```
# 然后根据ifconfig后的输出将各个host加入host1的/etc/hosts文件中
```

