1、docker default machine IP:

192.168.99.100

默认用户名密码：

docker

tcuser

sudo passwd root

root

root123

https://hub.docker.com/

rogers

Roger123@

docker修改root密码

在docker用户可以直接用

sudo passwd root

来修改

/var/lib/boot2docker/profile

export HTTP\_PROXY=http://web-proxy.oa.com:8080

export HTTPS\_PROXY=http://web-proxy.oa.com:8080

HTTP\_PROXY=http://[web-proxyhk.oa.com](http://www.google.com):8080

HTTPS\_PROXY=[https://](https://web-proxy.oa.com:8080)[web-proxyhk.oa.com](http://www.google.com):8080

export HTTP\_PROXY=http://[web-proxyhk.oa.com](http://www.google.com):8080

export HTTPS\_PROXY=[https://](https://web-proxy.oa.com:8080)[web-proxyhk.oa.com](http://www.google.com):8080

export HTTP\_PROXY=http://web-proxy.oa.com:8080

export HTTPS\_PROXY=http://web-proxy.oa.com:8080

docker create -ti --name="mona" centos bash

sudo /etc/init.d/docker restart

dev net:

set HTTP\_PROXY=http://dev-proxy.oa.com:8080

set HTTPS\_PROXY=http[://dev-proxy.oa.com:8080](https://dev-proxy.oa.com:8080)

export HTTP\_PROXY=http://dev-proxy.oa.com:8080

export HTTPS\_PROXY=http://dev-proxy.oa.com:8080

docker run hello-world

debian sources:

/etc/apt/sources.list

deb http://ftp.cn.debian.org/debian stable main

deb-src http://ftp.cn.debian.org/debian stable main

apt-get update

apt-get install vim

docker stop imagename

docker rm containername

2、mysql

install:

docker run --name wfd\_mysqlContainer -e MYSQL\_ROOT\_PASSWORD=rootroot -d mysql:latest

run:

docker start ee41bdc6f234

docker exec -it wfd\_mysqlContainer bash

3、nginx

install:

docker run --name wfd\_nginxContainer -v /some/content:/usr/share/nginx/html:ro -d nginx

run:

docker start 718

docker exec -it wfd\_nginxContainer bash

ps -ef|grep nginx|grep -v grep|cut -c 9-15|xargs kill -9

/usr/sbin/nginx -c /etc/nginx/nginx.conf

/usr/sbin/nginx start

/usr/sbin/nginx stop

/usr/sbin/nginx -s reload

4、ApachePHP

curl -v https://index.docker.io

/var/lib/boot2docker/profile

export HTTP\_PROXY=http://web-proxy.oa.com:8080

export HTTPS\_PROXY=https://web-proxy.oa.com:8080

重启docker

sudo /etc/init.d/docker restart

Dockerfile:

FROM php:5.6-apache

COPY src/ /var/www/html/

docker build -t my-php-app .

docker run -d --name my-running-app imageid（docker images查看）

ifconfig

ip address(ip a)

docker exec -it 2070 bash

root@2070c05c3909:/var/www/html# cat Dockerfile

FROM php:5.6-apache

COPY config/php.ini /usr/local/etc/php/

COPY src/ /var/www/html/

docker-php-ext-configure mysql

docker-php-ext-install mysql

container 里：

apt-get install python-pip

apt-get install python-dev

apt-get install samba

apt-get install smbclient

{

启动Samba服务器只需执行如下命令：

#sudo /etc/init.d/samba start

关闭Samba服务器：

#sudo /etc/init.d/samba stop

重新启动Samba服务器：

#sudo /etc/init.d/samba restart

ps -aux

/etc/samba/smb.conf

security=share

[share]

comment=this is docker share dir

path=/home

public=yes

writable=yes

docker run --rm -ti -v <host dir>:<container dir> dmxc-build bash

docker run -v /home/share/ --name wfd\_container1 mysql true

docker run --rm -v /usr/local/bin/docker:/docker -v /var/run/docker.sock:/docker.sock svendowideit/samba wfd\_container1

docker run --rm -ti --name wfd\_container1 -v /E/dockershare/:/home/share/ 0ffe8484635b bash

}

{

<https://dashboard.daocloud.io/>

rogers

[sdjnwfd@126.com](mailto:sdjnwfd@126.com)

Roger123@

sudo echo "DOCKER\_OPTS=\"$DOCKER\_OPTS --registry-mirror=http://3d24766e.m.daocloud.io\"" >> */var/lib/boot2docker/profile*

sudo /etc/init.d/docker restart

docker-machine ssh default
sudo sed -i "s|EXTRA\_ARGS='|EXTRA\_ARGS='--registry-mirror=http://3d24766e.m.daocloud.io |g" /var/lib/boot2docker/profile
exit
docker-machine restart default

docker-machine create --driver virtualbox dev --engine-registry-mirror ["http://3d24766e.m.daocloud.io"]

docker-machine create --engine-registry-mirror=http://3d24766e.m.daocloud.io -d virtualbox dev

**cat ubuntu-14.04-x86\_64.tar.gz |docker import - ubuntu:ubuntu14**

docker-machine ssh default
sudo sed -i "s|EXTRA\_ARGS='|EXTRA\_ARGS='--registry-mirror=http://3d24766e.m.daocloud.io |g" /var/lib/boot2docker/profile
exit
docker-machine restart default

/var/lib/boot2docker/profile

export HTTP\_PROXY=http://web-proxy.oa.com:8080

export HTTPS\_PROXY=<http://web-proxy.oa.com:8080>

需要安装一下我们的证书（需要root权限）：

mkdir -p /etc/docker/certs.d/docker.oa.com
mkdir -p /etc/docker/certs.d/registry.oa.com
wget docker.oa.com/cert/gaia.crt -O /etc/docker/certs.d/docker.oa.com/ca.crt
cp /etc/docker/certs.d/docker.oa.com/ca.crt /etc/docker/certs.d/registry.oa.com/

然后就可以愉快的使用https了。再也不用加--insecure-registry启动参数了。

docker search docker.oa.com/xxx // 其中xxx是用来模糊查找的字符串
docker login docker.oa.com // 仅限docker1.7及以上版本，否则往下看
docker pull docker.oa.com/library/centos7
docker tag CUSTOM\_IMAGE docker.oa.com/blizard/warcraft:v1.1
docker push docker.oa.com/blizard/warcraft:v1.1

docker run --net=isolated\_nw -itd --name=mycontainer2 mysql

docker run -p 8080:1000/tcp --name=ngixcontainer1 0d409d33b27e

docker run -p 8080:2000/tcp --name=mysqlcontainer1 43c70d2fa278

docker inspect $container\_id

docker port $container\_id

/etc/init.d/nginx reload

ps -ef|grep nginx|grep -v grep|cut -c 9-15|xargs kill -9

/etc/init.d/nginx -c /etc/nginx/nginx.conf

<http://192.168.99.100:8080/>

/usr/sbin/nginx -t -c nginx.conf

docker run -p 8090:2000/tcp --name=ngixcontainer1 0d409d33b27e

docker run -p 8091:2001/tcp --name=ngixcontainer2 0d409d33b27e

docker run -p [8091:2000,8092:2001,8092:2002] --name=ngixcontainer2 0d409d33b27e

docker run -p 8091:2001/tcp --name=mycontainer1 -e MYSQL\_ROOT\_PASSWORD=rootroot -d 43c70d2fa278

echo -e "deb http://ftp.cn.debian.org/debian stable main \ndeb-src http://ftp.cn.debian.org/debian stable main"> /etc/apt/sources.list

apt-get update

apt-get install ssh

apt-get install python-pip

apt-get install python-dev

apt-get install python-mysqldb

pip install web.py

/etc/init.d/nginx reload

/etc/services

mysql -uroot -prootroot

mysql -h172.17.0.3 -uroot -prootroot

create table testtable1(id int(11) not null primary key auto\_increment, name varchar(20) not null);

insert into testtable1(id,name) values(1001, 'rogers1');

test.php:

<?php

$con = mysql\_connect("172.17.0.3","root","rootroot");

if (!$con)

  {

  die('Could not connect: ' . mysql\_error());

  }

mysql\_select\_db("mysql", $con);

$result = mysql\_query("SELECT \* FROM test\_user");

    while($row = mysql\_fetch\_array($result))

    {

        echo $row['id'] . " " . $row['name'] . " " . $row['password'] . " " . $row['registertime'] . " " . $row['lastaccesstime'];

        echo "\n";

    }

// some code

?>



--------------------------------------------------------------------------------------------------------

mahout：

docker search ubuntu

docker run -p 10001:10001/tcp --name=mahout1 ubuntu:latest

docker run -it -p 10001:10001/tcp --name=mahout1 ff6011336327

docker run -it -p 10002:10002/tcp --name=mahout2 ff6011336327

docker run -it -p 10003:10003/tcp --name=mahout3 ff6011336327

docker run -it -p 10004:10004/tcp --name=mahout4 ff6011336327

docker run -it -p 10005:10005/tcp --name=mahout5 ff6011336327

echo -e "deb http://ftp.cn.debian.org/debian stable main \ndeb-src http://ftp.cn.debian.org/debian stable main"> /etc/apt/sources.list

apt-get update

apt-get install vim

apt-get install ssh

apt-get install python-pip

apt-get install python-dev

apt-get install python-mysqldb

pip install web.py

samba:

/etc/samba/smb.conf

contents:

[global]

workgroup = TENCENT

netbios name = TENCENT

server string = Linux Samba Server TestServer

security = share

[share]

path = /data

writeable = yes

browseable = yes

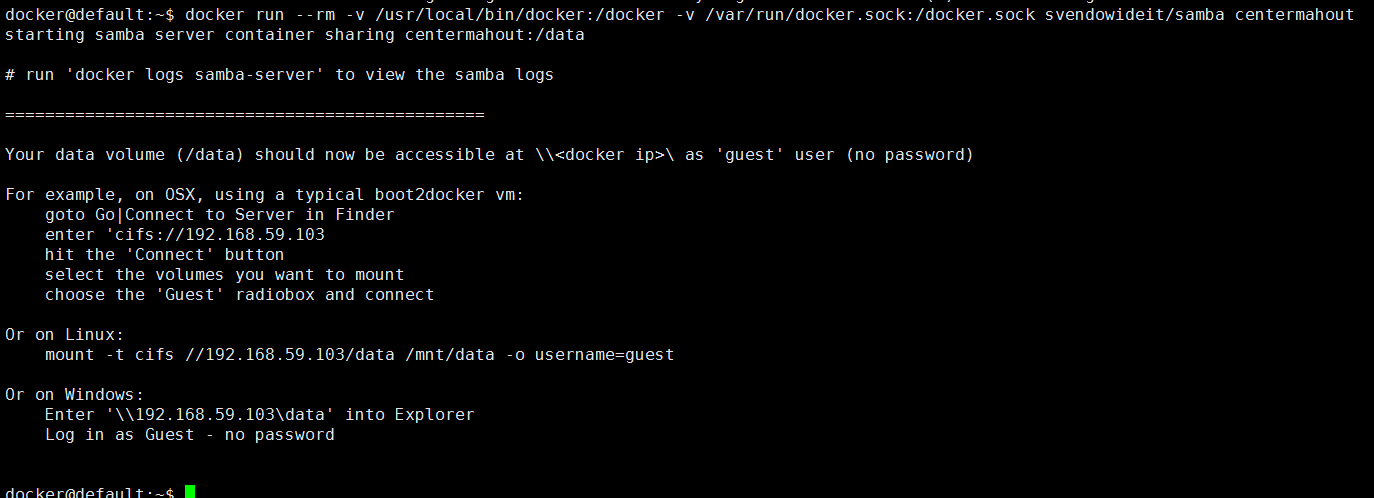
guest ok = yes

/etc/init.d/samba start

data volume:

docker run -it -p 139:139 -p 445:445 -v /data --name mahout ff6011336327

docker run --rm -v /usr/local/bin/docker:/docker -v /var/run/docker.sock:/docker.sock svendowideit/samba mahout



id nobody

chown -R nobody:nogroup /data

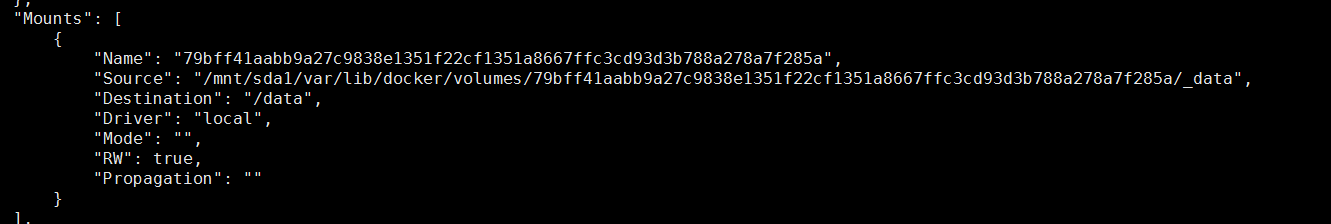
\\192.168.99.100\data

docker run -v /data --name my-data busybox true

docker run --rm -v /usr/local/bin/docker:/docker -v /var/run/docker.sock:/docker.sock svendowideit/samba my-data

docker run -it -p 139:139 -p 445:445 -p 10000:10000/tcp --name mahout0 --volumes-from my-data ff6011336327 /bin/bash

docker inspect my-data mahout



其他容器只要挂载这个volume就可以共享这个目录：

docker run -it -p 10000:10000/tcp --name=mahout0 --volumes-from my-data ff6011336327 /bin/bash

(docker run -dit --restart=always -p 139:139 -p 445:445 -p 10000:10000/tcp --name mahout0 --volumes-from my-data ff6011336327 -s 'data;/my-data;yes;no;yes;all' /bin/bash

**groupadd**dockeruser -g 6000

**useradd** dockeruser -u 6000 -g 6000 -s /sbin/nologin -d /dev/null

sudo smbpasswd -a dockeruser

)

can work:

docker pull dperson/samba

docker run -v /data --name my-data busybox true

\\192.168.99.100\data

docker run -it -p 10000:10000/tcp --name=mahout0 --volumes-from my-data ff6011336327 /bin/bash

docker run -it -p 10001:10001/tcp --name=mahout1 --volumes-from my-data ff6011336327 /bin/bash

docker run -it -p 10002:10002/tcp --name=mahout2 --volumes-from my-data ff6011336327 /bin/bash

docker run -it -p 10003:10003/tcp --name=mahout3 --volumes-from my-data ff6011336327 /bin/bash

docker run -it -p 10004:10004/tcp --name=mahout4 --volumes-from my-data ff6011336327 /bin/bash

docker run -it -p 10005:10005/tcp --name=mahout5 --volumes-from my-data ff6011336327 /bin/bash

docker run --name samba -p 139:139 -p 445:445 -v /mnt/sda1/var/lib/docker/volumes/cbcf59073cd466750dcc4d5a1887df7b1cbc65f6762d2c2fdea1ecc9f472f136/\_data/:/my-data -d dperson/samba samba.sh -s 'data;/my-data;yes;no;yes;all'

echo -e "deb http://ftp.cn.debian.org/debian stable main \ndeb-src http://ftp.cn.debian.org/debian stable main"> /etc/apt/sources.list

apt-get update

apt-get install ssh

apt-cache showpkg tzdata

apt-get install tzdata=2016d-0+deb8u1

apt-get install tzdata-java

apt-get install openjdk-7-jre

apt-get install openjdk-7-jdk

apt-get install maven

apt-get install subversion

export HTTP\_PROXY=http://web-proxy.oa.com:8080

export HTTPS\_PROXY=<http://web-proxy.oa.com:8080>

ssh-keygen -t rsa -P ""

cat ~/.ssh/id\_rsa.pub >>~/.ssh/authorized\_keys

sudo apt-get install rsync

**~/.bashrc:**

export HADOOP\_HOME=/usr/local/hadoop

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

export YARN\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native

export PATH=$PATH:$HADOOP\_HOME/sbin:$HADOOP\_HOME/bin

export HADOOP\_INSTALL=$HADOOP\_HOME

source ~/.bashrc

etc/hadoop/hadoop-env.sh

/usr/lib/jvm/java-7-openjdk-amd64

**core-site.xml**

<configuration>

<property>

<name>fs.default.name</name>

<value>hdfs://localhost:10000</value>

</property>

</configuration>

**hdfs-site.xml**

<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.name.dir</name>
<value>file:///home/hadoop/hadoopinfra/hdfs/namenode</value>
</property>
<property>
<name>dfs.data.dir</name>
<value>file:///home/hadoop/hadoopinfra/hdfs/datanode</value>
</property>
</configuration>

**yarn-site.xml**

<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce\_shuffle</value>
</property>
</configuration>

cp mapred-site.xml.template mapred-site.xml

**mapred-site.xml**

<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>

cd ~

hdfs namenode -format

sbin/start-dfs.sh

sbin/start-yarn.sh

<dependency>
<groupId>org.apache.mahout</groupId>
<artifactId>mahout-core</artifactId>
<version>0.9</version>
</dependency>

mahout:

pom:
<dependency>
<groupId>org.apache.mahout</groupId>
<artifactId>mahout-math</artifactId>
<version>${mahout.version}</version>
</dependency>
<dependency>
<groupId>org.apache.mahout</groupId>
<artifactId>mahout-integration</artifactId>
<version>${mahout.version}</version>
</dependency>

check gtk:

dpkg -l libgtk[0-9]\* | grep ^i

docker run -it -p 8080:8080 -p 49152-49162:49152-49162 --name=eclipse --volumes-from my-data cda93ea38352

apt-get update

apt-get install vim

apt-get install ssh

apt-get install python-pip

apt-get install python-dev

apt-get install python-mysqldb

pip install web.py

(~/.profile

source /etc/profile)

scikit-learn:

echo -e "deb http://ftp.cn.debian.org/debian stable main \ndeb-src http://ftp.cn.debian.org/debian stable main"> /etc/apt/sources.list

apt-get install python

python get-pip.py

pip install NumPy

pip install SciPy

pip install -U scikit-learn

pip install jieba

使用中文时这样用：(locale)

docker exec -it mahout0 env LANG=C.UTF-8 bash

locale-gen en\_US.UTF-8

export LANG=en\_US.UTF-8

export LANGUAGE=en\_US:en