2.获取和创建镜像

五、获取镜像

1.镜像是一个模板，可以从镜像服务器上获取做好的镜像。docker 使用 docker pull命令来获取需要的镜像。

```
#会到镜像仓库中下载镜像到docker
docker pull mysql
#mariadb就是mysql的代替品
docker pull mariadb

#查看docker中的镜像
docker images

docker stop $(docker ps -aq)


#删除所有容器
docker rm `docker ps -a -q`
docker rm $(docker ps -aq)


#删除所有镜像
docker rmi `docker images -q`
docker rmi $(docker images -q)
```

```
[OK]
[root@bigdata-4 ~]# docker search mysql
^Z
[1]+  Stopped                 docker search mysql
[root@bigdata-4 ~]# docker search mysql
INDEX       NAME                                                DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
docker.io   docker.io/mysql                                     MySQL is a widely used, open-source relati...   8193      [OK]
docker.io   docker.io/mariadb                                   MariaDB is a community-developed fork of M...   2790      [OK]
docker.io   docker.io/mysql/mysql-server                        Optimized MySQL Server Docker images. Crea...   610                 [OK]
docker.io   docker.io/percona                                   Percona Server is a fork of the MySQL rela...   434       [OK]
docker.io   docker.io/centurylink/mysql                         Image containing mysql. Optimized to be li...   60                  [OK]
docker.io   docker.io/centos/mysql-57-centos7                   MySQL 5.7 SQL database server                   53
docker.io   docker.io/mysql/mysql-cluster                       Experimental MySQL Cluster Docker images. ...   44
docker.io   docker.io/deitch/mysql-backup                       Automated and scheduled mysql database dum...   36                  [OK]
docker.io   docker.io/tutum/mysql                               Base docker image to run a MySQL database ...   32
docker.io   docker.io/bitnami/mysql                             Bitnami MySQL Docker Image                      27                  [OK]
docker.io   docker.io/schickling/mysql-backup-s3                Backup MySQL to S3 (supports periodic back...   27                  [OK]
docker.io   docker.io/linuxserver/mysql                         A Mysql container, brought to you by Linux...   20
docker.io   docker.io/prom/mysqld-exporter                      17                  [OK]
docker.io   docker.io/centos/mysql-56-centos7                   MySQL 5.6 SQL database server                   13
docker.io   docker.io/circleci/mysql                            MySQL is a widely used, open-source relati...   11
docker.io   docker.io/mysql/mysql-router                        MySQL Router provides transparent routing ...   11
docker.io   docker.io/arey/mysql-client                         Run a MySQL client from a docker container      9                   [OK]
docker.io   docker.io/openshift/mysql-55-centos7                DEPRECATED: A Centos7 based MySQL v5.5 ima...   6
docker.io   docker.io/yloeffler/mysql-backup                    This image runs mysqldump to backup data u...   6                   [OK]
docker.io   docker.io/fradelg/mysql-cron-backup                 MySQL/MariaDB database backup using cron t...   4                   [OK]
docker.io   docker.io/jelastic/mysql                            An image of the MySQL database server main...   1
docker.io   docker.io/ansibleplaybookbundle/mysql-apb           An APB which deploys RHSCL MySQL                0                   [OK]
docker.io   docker.io/astronomerio/mysql-sink                   MySQL sink                                      0                   [OK]
docker.io   docker.io/monasca/mysql-init                        A minimal decoupled init container for mysql    0
docker.io   docker.io/widdpim/mysql-client                      Dockerized MySQL Client (5.7) including Cu...   0                   [OK]
[root@bigdata-4 ~]#
```

2.例如我们获取一个mysql镜像。

```
#获取镜像
docker pull mysql

#查看docker中的镜像
docker images
```

```
#运行docker 容器
docker run -t -i xxx  /bin/bash
-t   分配一个终端到docker容器的标准输入上
-i   让容器的标准输入保持打开

#退出docker容器
exit


#查看运行中的容器
docker ps
```

```
[root@bigdata-4 ~]# docker pull mysql
Using default tag: latest
Trying to pull repository docker.io/library/mysql ...
latest: Pulling from docker.io/library/mysql
743f2d6c1f65: Pull complete
3f0c413ee255: Pull complete
aef1ef8f1aac: Pull complete
f9ee573e34cb: Pull complete
3f237e01f153: Pull complete
f9da32e8682a: Pull complete
4b8da52fb357: Pull complete
3416ca8f6890: Pull complete
786698c2d5de: Pull complete
4ddf84d07bd1: Pull complete
cd3aa23461b6: Pull complete
9f287a2a95ad: Pull complete
Digest: sha256:711df5b93720801b3a727864aba18c2ae46c07f9fe33d5ce9c1f5cbc2c035101
Status: Downloaded newer image for docker.io/mysql:latest
```

3.我们可以使用docker images显示本地已经有的镜像。

```
docker images
```

```
[root@bigdata-4 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
docker.io/mysql     latest              990386cbd5c0        2 weeks ago         443 MB
spark               v1.0                61bc8fb93bb5        2 months ago        763 MB
[root@bigdata-4 ~]#
```

4.我们再获取一个centos的镜像。

```
docker pull centos
```

5.使用刚刚获取的centos镜像，创建一个容器，在容器中运行bash应用

```
[root@bigdata-4 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
docker.io/ubuntu    latest              7698f282e524        11 days ago         69.9 MB
docker.io/mysql     latest              990386cbd5c0        2 weeks ago         443 MB
spark               v1.0                61bc8fb93bb5        2 months ago        763 MB
[root@bigdata-4 ~]# docker run -t -i docker.io/ubuntu /bin/bash
WARNING: IPv4 forwarding is disabled. Networking will not work.
root@b47f7fadbead:/# ls -l
total 4
drwxr-xr-x   2 root root 4096 May 15 14:07 bin
drwxr-xr-x   2 root root    6 Apr 24  2018 boot
drwxr-xr-x   5 root root  360 May 27 03:54 dev
drwxr-xr-x   1 root root   66 May 27 03:54 etc
drwxr-xr-x   2 root root    6 Apr 24  2018 home
drwxr-xr-x   8 root root   96 May 23  2017 lib
drwxr-xr-x   2 root root   34 May 15 14:06 lib64
drwxr-xr-x   2 root root    6 May 15 14:06 media
drwxr-xr-x   2 root root    6 May 15 14:06 mnt
drwxr-xr-x   2 root root    6 May 15 14:06 opt
dr-xr-xr-x 119 root root    0 May 27 03:54 proc
drwx------   2 root root   37 May 15 14:07 root
drwxr-xr-x   1 root root   21 May 27 03:54 run
drwxr-xr-x   1 root root   21 May 15 21:20 sbin
drwxr-xr-x   2 root root    6 May 15 14:06 srv
dr-xr-xr-x  13 root root    0 May 27 03:36 sys
drwxrwxrwt   2 root root    6 May 15 14:07 tmp
drwxr-xr-x   1 root root   18 May 15 14:06 usr
drwxr-xr-x   1 root root   17 May 15 14:07 var
root@b47f7fadbead:/# exit
exit
[root@bigdata-4 ~]#
```

六、创建镜像

1. 创建镜像有很多种方法，用户可以更新镜像，也可以利用本地文件系统创建镜像。 a.修改已有镜像, 先使用docker pull training/sinatra下载镜像

```
#获取镜像
docker pull training/sinatra
docker images
```

```
[root@bigdata-4 ~]# docker pull training/sinatra
Using default tag: latest
Trying to pull repository docker.io/training/sinatra ...
latest: Pulling from docker.io/training/sinatra
a3ed95caeb02: Pull complete
6e71c809542e: Pull complete
d196a7609355: Pull complete
08f6dff5acea: Pull complete
ce65532003d0: Pull complete
54bcaa4d1a10: Pull complete
8572ad96f6e1: Pull complete
Digest: sha256:03fc0cd265cbc28723e4efd446f9f2f37b4790cf9cc12f1b9203c79fb86b6772
Status: Downloaded newer image for docker.io/training/sinatra:latest
```

b.启动镜像

```
#进入容器交互接口
docker run -t -i training/sinatra /bin/bash

#ls -la 类似于简化版的linux操作系统
```

```
[root@bigdata-4 ~]# docker run -t -i training/sinatra /bin/bash
WARNING: IPv4 forwarding is disabled. Networking will not work.
root@e2ba5ff4b89d:/# ls -la
total 4
drwxr-xr-x   1 root root   17 May 27 07:12 .
drwxr-xr-x   1 root root   17 May 27 07:12 ..
-rwxr-xr-x   1 root root    0 May 27 07:12 .dockerenv
drwxr-xr-x   3 root root   19 Jun  9  2014 .gem
drwxr-xr-x   2 root root 4096 Apr 16  2014 bin
drwxr-xr-x   2 root root    6 Apr 10  2014 boot
drwxr-xr-x   5 root root  360 May 27 07:12 dev
drwxr-xr-x   1 root root   66 May 27 07:12 etc
drwxr-xr-x   2 root root    6 Apr 10  2014 home
drwxr-xr-x   1 root root   56 Jun  9  2014 lib
drwxr-xr-x   2 root root   34 Apr 16  2014 lib64
drwxr-xr-x   2 root root    6 Apr 16  2014 media
drwxr-xr-x   2 root root    6 Apr 10  2014 mnt
drwxr-xr-x   2 root root    6 Apr 16  2014 opt
dr-xr-xr-x 123 root root    0 May 27 07:12 proc
drwx------   2 root root   37 Apr 16  2014 root
drwxr-xr-x   1 root root   21 May 27 07:12 run
drwxr-xr-x   1 root root   44 May 29  2014 sbin
drwxr-xr-x   2 root root    6 Apr 16  2014 srv
dr-xr-xr-x  13 root root    0 May 27 03:36 sys
drwxrwxrwt   1 root root    6 Jun  9  2014 tmp
drwxr-xr-x   1 root root   19 Jun  9  2014 usr
drwxr-xr-x   1 root root   17 Jun  9  2014 var
root@e2ba5ff4b89d:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:ac:11:00:02
          inet addr:172.17.0.2  Bcast:0.0.0.0  Mask:255.255.0.0
          inet6 addr: fe80::42:acff:fe11:2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

c. 在镜像中添加json和gem两个应用

```
#
gem install json
```

添加完成后，使用exit退出容器，我们在容器中添加了新的应用，容器被改变。

```
[root@bigdata-4 ~]# docker run -t -i training/sinatra /bin/bash
root@01d2b9da9de9:/# ping www.baidu.com
PING www.a.shifen.com (14.215.177.38) 56(84) bytes of data.
64 bytes from 14.215.177.38: icmp_seq=1 ttl=127 time=27.2 ms
64 bytes from 14.215.177.38: icmp_seq=2 ttl=127 time=28.1 ms
64 bytes from 14.215.177.38: icmp_seq=3 ttl=127 time=27.0 ms
64 bytes from 14.215.177.38: icmp_seq=4 ttl=127 time=28.1 ms
64 bytes from 14.215.177.38: icmp_seq=5 ttl=127 time=29.3 ms
^Z
[1]+  Stopped                 ping www.baidu.com
root@01d2b9da9de9:/# gem install json
Fetching: json-2.2.0.gem (100%)
Building native extensions.  This could take a while...
Successfully installed json-2.2.0
1 gem installed
Installing ri documentation for json-2.2.0...
Installing RDoc documentation for json-2.2.0...
root@01d2b9da9de9:/# exit
exit
There are stopped jobs.
root@01d2b9da9de9:/#
```

d.我们使用docker commit 命令来提交更新后的副本。

```
docker commit -m 'add json gem' -a 'Docker Container'  7b789b19757d my/sinatra:v2
```

-m参数指定提交的说明，和Git中的git commit -m的参数是一样的；-a指定更新的用户信息，后面是用来创建镜像的容器ID，最后是创建镜像的仓库名和tag信息。创建成功后，命令会返回这个镜像的ID信息。

```
root@01d2b9da9de9:/# gem install json
Fetching: json-2.2.0.gem (100%)
Building native extensions.  This could take a while...
Successfully installed json-2.2.0
1 gem installed
Installing ri documentation for json-2.2.0...
Installing RDoc documentation for json-2.2.0...
root@01d2b9da9de9:/# exit
exit
There are stopped jobs.
root@01d2b9da9de9:/# exit
exit
[root@bigdata-4 ~]# docker commit -m 'add json gem' -a 'Docker Container' 01d2b9da9de9 my/sinatra:v2
sha256:01caf5f11ab1f0b0f321b9be0e8a8a3d0320e63bb780f0602052235d3190178
[root@bigdata-4 ~]#
```

e.接着我们可以使用我们自己创建的镜像来启动容器了

```
docker run -t -i my/sinatra:v2 /bin/bash
ls -la
ifconfig
```

我们可以看到，一个容器就是一个微型的LINUX系统

```
[root@bigdata-4 ~]# docker run -t -i my/sinatra:v2 /bin/bash
root@611e561d72d1:/# ls -a
.   .bash_history  .gem  boot  etc   lib    media  opt   root  sbin  sys  usr
..  .dockerenv     bin   dev   home  lib64  mnt    proc  run   srv   tmp  var
root@611e561d72d1:/# ls -l
total 4
drwxr-xr-x   2 root root 4096 Apr 16  2014 bin
drwxr-xr-x   2 root root    6 Apr 10  2014 boot
drwxr-xr-x   5 root root  360 May 27 07:38 dev
drwxr-xr-x   1 root root   66 May 27 07:38 etc
drwxr-xr-x   2 root root    6 Apr 10  2014 home
drwxr-xr-x   1 root root   56 Jun  9  2014 lib
drwxr-xr-x   2 root root   34 Apr 16  2014 lib64
drwxr-xr-x   2 root root    6 Apr 16  2014 media
drwxr-xr-x   2 root root    6 Apr 10  2014 mnt
drwxr-xr-x   2 root root    6 Apr 16  2014 opt
dr-xr-xr-x 124 root root    0 May 27 07:38 proc
drwx------   2 root root   37 Apr 16  2014 root
drwxr-xr-x   1 root root   21 May 27 07:33 run
drwxr-xr-x   1 root root   44 May 29  2014 sbin
drwxr-xr-x   2 root root    6 Apr 16  2014 srv
dr-xr-xr-x  13 root root    0 May 27 03:36 sys
drwxrwxrwt   1 root root    6 May 27 07:33 tmp
drwxr-xr-x   1 root root   19 Jun  9  2014 usr
drwxr-xr-x   1 root root   17 Jun  9  2014 var
root@611e561d72d1:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:ac:11:00:02
          inet addr:172.17.0.2  Bcast:0.0.0.0  Mask:255.255.0.0
          inet6 addr: fe80::42:acff:fe11:2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

2. 使用Dockerfile来创建镜像，使用docker commit虽然很容易扩展镜像，但不便于团队分享，我们可以使用docker build来创建一个新的镜像，为此首先要创建一个Dockerfile文件(名字不要搞错)，这个文件中包含如何创建镜像的指令。 a.首先新建一个目录和Dockerfile

```
mkdir centos
vi Dockerfile
#注解
FROM centos
MAINTAINER REGAN 191801055@qq.com
RUN yum -qqy install python
```

```
[root@bigdata-4 ~]# mkdir centos
[root@bigdata-4 ~]# cd centos
[root@bigdata-4 centos]# vi Dockerfile
[root@bigdata-4 centos]# cat Dockerfile
#注解
FROM centos
MAINTAINER REGAN 191801055@qq.com
RUN yum -qqy install mysql
[root@bigdata-4 centos]# docker build -t='my/centos_with_mysql:v1'
```

b。Dockerfile中输入指令，每条指令都创建镜像的一层。 Dockerfile中内容如下：

```
#注解
FROM centos
MAINTAINER REGAN 191801055@qq.com
RUN yum -qqy install mysql
```

```
#构建docker镜像
docker build -t='my/centos_with_mysql:v1' .
docker images
```

-其中 FROM 告诉Docker使用哪个镜像作为基础;接着是维护者的信息; RUN开头的指令会在创建中运行，例如安装一些软件包，这里使用yum 安装python. 注意使用yum 需要制定参数 -qqy，不然呢可能会报错的。 然后使用docker build构建镜像：

```
docker build -t='my/centos_with_mysql:v1' .
```

```
[root@bigdata-4 centos]# docker build -t='my/centos_with_mysql:v1' .
Sending build context to Docker daemon 2.048 kB
Step 1/3 : FROM centos
Trying to pull repository docker.io/library/centos ...
latest: Pulling from docker.io/library/centos
8ba884070f61: Pull complete
Digest: sha256:b5e66c4651870a1ad435cd75922fe2cb943c9e973a9673822d1414824a1d0475
Status: Downloaded newer image for docker.io/centos:latest
 ---> 9f38484d220f
Step 2/3 : MAINTAINER REGAN 191801055@qq.com
 ---> Running in f3a13a3ef396
 ---> 91ba90041367
Removing intermediate container f3a13a3ef396
Step 3/3 : RUN yum -qqy install mysql
 ---> Running in bbfc0c3be3be

Loaded plugins: fastestmirror, ovl
Ignored option -q, -v, -d or -e (probably due to merging: -yq != -y -q)
Determining fastest mirrors
 * base: mirrors.aliyun.com
 * extras: mirrors.aliyun.com
 * updates: mirrors.huaweicloud.com
Resolving Dependencies
--> Running transaction check
---> Package mariadb.x86_64 1:5.5.60-1.el7_5 will be installed
--> Processing Dependency: mariadb-libs(x86-64) = 1:5.5.60-1.el7_5 for package: 1:mariadb-5.5.60-1.el7_5.x86_64
--> Processing Dependency: perl(Sys::Hostname) for package: 1:mariadb-5.5.60-1.el7_5.x86_64
--> Processing Dependency: perl(IPC::Open3) for package: 1:mariadb-5.5.60-1.el7_5.x86_64
--> Processing Dependency: perl(Getopt::Long) for package: 1:mariadb-5.5.60-1.el7_5.x86_64
--> Processing Dependency: perl(File::Temp) for package: 1:mariadb-5.5.60-1.el7_5.x86_64
--> Processing Dependency: perl(Fcntl) for package: 1:mariadb-5.5.60-1.el7_5.x86_64
--> Processing Dependency: perl(Exporter) for package: 1:mariadb-5.5.60-1.el7_5.x86_64
--> Processing Dependency: /usr/bin/perl for package: 1:mariadb-5.5.60-1.el7_5.x86_64
--> Running transaction check
---> Package mariadb-libs.x86_64 1:5.5.60-1.el7_5 will be installed
---> Package perl.x86_64 4:5.16.3-294.el7_6 will be installed
```

其中 -t 标记来添加 tag，指定新的镜像的用户信息。 “.” 是 Dockerfile 所在的路径（当前目录），也可以替 换为一个具体的 Dockerfile 的路径。 可以看到 build 进程在执行操作。它要做的第一件事情就是上传这个 Dockerfile 内容，因为所有的操作都要依据 Dockerfile 来进行。 然后，Dockfile 中的指令被一条一条的执行。每一步都创建了一个新的容器，在容器中执行指令并提交修改（就跟之前介绍过的 docker commit 一样）。当所有的指令都执行

完毕之后，返回了最终的镜像 id。所有的中间步骤所产生的容器都被删除和清理了。 需要注意的是：一个镜像是不能操作127层的，否则会报错！ 当然Dockerfile中还可以输入其他的命来，例如可以使用ADD命令复制本地文件到镜像中；用EXPOSE命令对外开放端口；用CMD命令描述容器启动后运行的程序。

```
--> Processing Dependency: perl-libs = 4:5.16.3-294.el7_6 for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: perl(Socket) >= 1.3 for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: perl(Scalar::Util) >= 1.10 for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: perl-macros for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: perl-libs for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: perl(threads::shared) for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: perl(threads) for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: perl(constant) for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: perl(Time::Local) for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: perl(Time::HiRes) for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: perl(Storable) for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: perl(Socket) for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: perl(Scalar::Util) for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: perl(Pod::Simple::XHTML) for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: perl(Pod::Simple::Search) for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: perl(Filter::Util::Call) for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: perl(File::Spec::Unix) for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: perl(File::Spec::Functions) for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: perl(File::Spec) for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: perl(File::Path) for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: perl(Cwd) for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: perl(Carp) for package: 4:perl-5.16.3-294.el7_6.x86_64
--> Processing Dependency: libperl.so()(64bit) for package: 4:perl-5.16.3-294.el7_6.x86_64
---> Package perl-Exporter.noarch 0:5.68-3.el7 will be installed
---> Package perl-File-Temp.noarch 0:0.23.01-3.el7 will be installed
---> Package perl-Getopt-Long.noarch 0:2.40-3.el7 will be installed
--> Processing Dependency: perl(Pod::Usage) >= 1.14 for package: perl-Getopt-Long-2.40-3.el7.noarch
--> Processing Dependency: perl(Text::ParseWords) for package: perl-Getopt-Long-2.40-3.el7.noarch
--> Running transaction check
---> Package perl-Carp.noarch 0:1.26-244.el7 will be installed
---> Package perl-File-Path.noarch 0:2.09-2.el7 will be installed
---> Package perl-Filter.x86_64 0:1.49-3.el7 will be installed
---> Package perl-PathTools.x86_64 0:3.40-5.el7 will be installed
---> Package perl-Pod-Simple.noarch 1:3.28-4.el7 will be installed
--> Processing Dependency: perl(Pod::Escapes) >= 1.04 for package: 1:perl-Pod-Simple-3.28-4.el7.noarch
--> Processing Dependency: perl(Encode) for package: 1:perl-Pod-Simple-3.28-4.el7.noarch
---> Package perl-Pod-Usage.noarch 0:1.63-3.el7 will be installed
```

```
  perl-Pod-Usage.noarch 0:1.63-3.el7
  perl-Scalar-List-Utils.x86_64 0:1.27-248.el7
  perl-Socket.x86_64 0:2.010-4.el7
  perl-Storable.x86_64 0:2.45-3.el7
  perl-Text-ParseWords.noarch 0:3.29-4.el7
  perl-Time-HiRes.x86_64 4:1.9725-3.el7
  perl-Time-Local.noarch 0:1.2300-2.el7
  perl-constant.noarch 0:1.27-2.el7
  perl-libs.x86_64 4:5.16.3-294.el7_6
  perl-macros.x86_64 4:5.16.3-294.el7_6
  perl-parent.noarch 1:0.225-244.el7
  perl-podlators.noarch 0:2.5.1-3.el7
  perl-threads.x86_64 0:1.87-4.el7
  perl-threads-shared.x86_64 0:1.43-6.el7

Complete!
 ---> 01f9e71db666
Removing intermediate container bbfc0c3be3be
Successfully built 01f9e71db666
[root@bigdata-4 centos]# docker images
REPOSITORY                  TAG                 IMAGE ID            CREATED             SIZE
my/centos_with_mysql        v1                  01f9e71db666        About a minute ago  387 MB
my/sinatra                  v2                  01ca75f11ab1        36 minutes ago      453 MB
docker.io/ubuntu            latest              7698f282e524        11 days ago         69.9 MB
docker.io/mysql             latest              990386cbd5c0        2 weeks ago         443 MB
docker.io/centos            latest              9f38484d220f        2 months ago        202 MB
spark                       v1.0                61bc8fb93bb5        2 months ago        763 MB
docker.io/training/sinatra  latest              49d952a36c58        4 years ago         447 MB
[root@bigdata-4 centos]# docker run -t -i my/centos_with_mysql /bin/bash
Unable to find image 'my/centos_with_mysql:latest' locally
Trying to pull repository docker.io/my/centos_with_mysql ...
/usr/bin/docker-current: repository docker.io/my/centos_with_mysql not found: does not exist or no pull access.
See '/usr/bin/docker-current run --help'.
```

c.现在可以使用创建好的额镜像来启动容器了

```
    docker run -t -i my/centos_with_mysql /bin/bash
```

```
[root@bigdata-4 centos]# docker run -t -i my/centos_with_mysql /bin/bash
Unable to find image 'my/centos_with_mysql:latest' locally
Trying to pull repository docker.io/my/centos_with_mysql ...
/usr/bin/docker-current: repository docker.io/my/centos_with_mysql not found: does not exist or no pull access.
See '/usr/bin/docker-current run --help'.
[root@bigdata-4 centos]# docker run -t -i 01f9e71db666 /bin/bash
[root@4759170eb4fa /]#
[root@4759170eb4fa /]#
[root@4759170eb4fa /]#
```

d.现在觉的镜像的标签不好，可以使用docker tag 命来来修改镜像的标签。

```
  docker tag image_id  xxxx
```

```
[root@bigdata-4 centos]# docker tag 01f9e71db666 my/centos_mysql:v1
[root@bigdata-4 centos]# docker images
REPOSITORY                    TAG                 IMAGE ID            CREATED             SIZE
my/centos_mysql               v1                  01f9e71db666        17 minutes ago      387 MB
my/centos_with_mysql          v1                  01f9e71db666        17 minutes ago      387 MB
my/sinatra                    v2                  01caf5f11ab1        52 minutes ago      453 MB
docker.io/ubuntu              latest              7698f282e524        11 days ago         69.9 MB
docker.io/mysql               latest              990386cbd5c0        2 weeks ago         443 MB
docker.io/centos              latest              9f38484d220f        2 months ago        202 MB
spark                         v1.0                61bc8fb93bb5        2 months ago        763 MB
docker.io/training/sinatra    latest              49d952a36c58        4 years ago         447 MB
[root@bigdata-4 centos]#
```