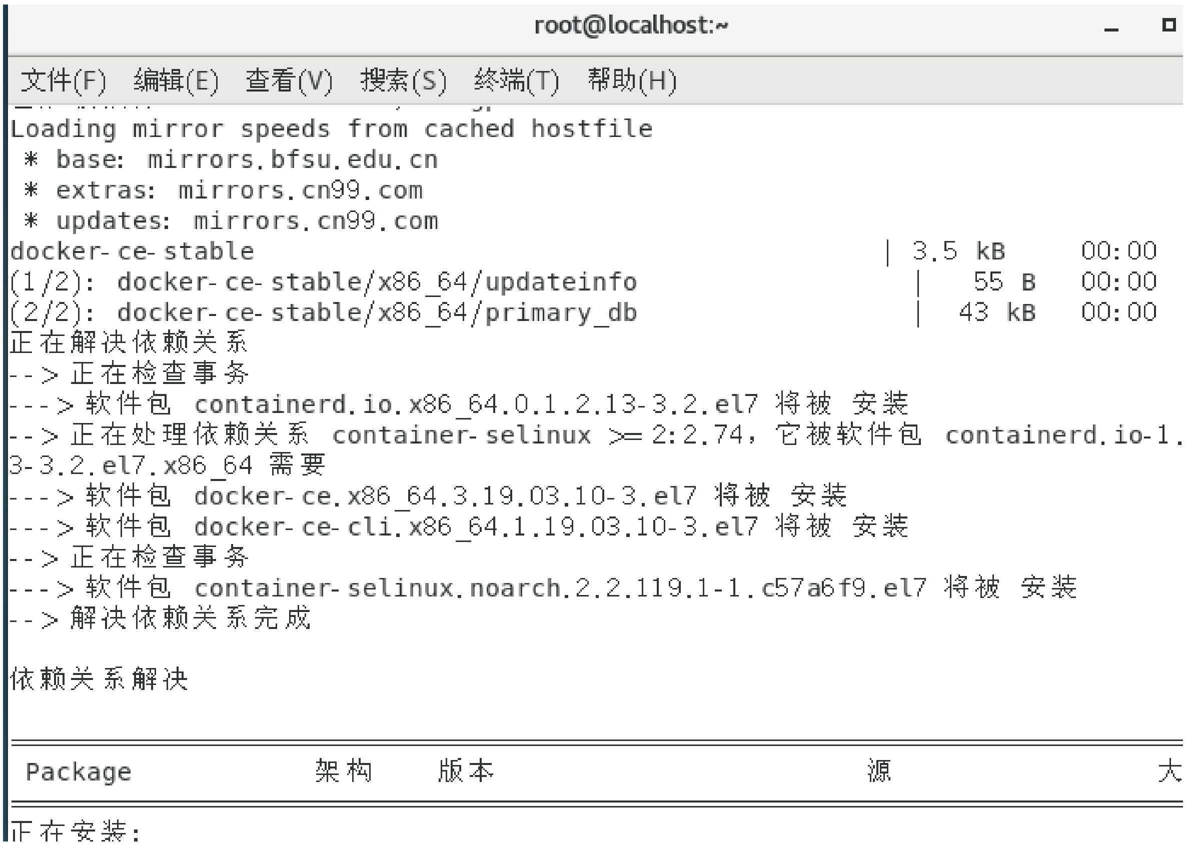
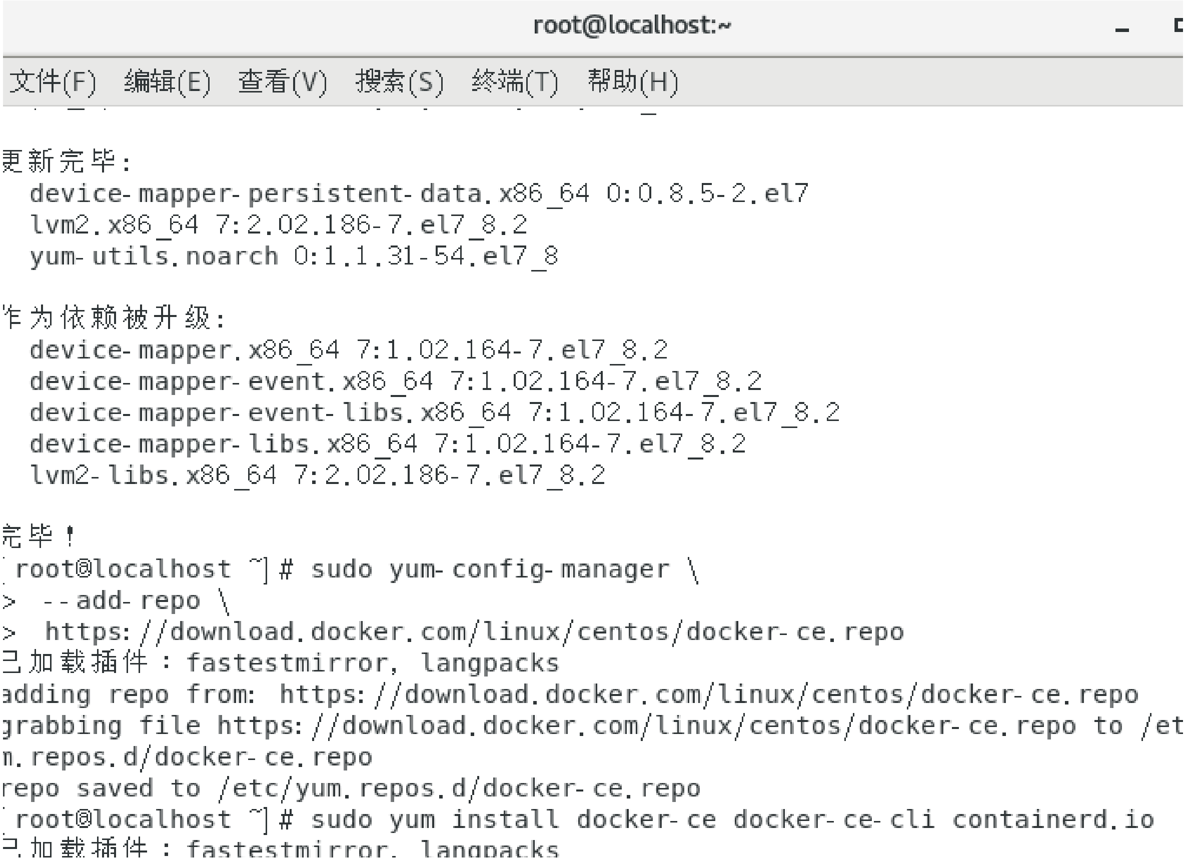
一、完成centos上安装Docker

1、安装 Docker Engine-Community



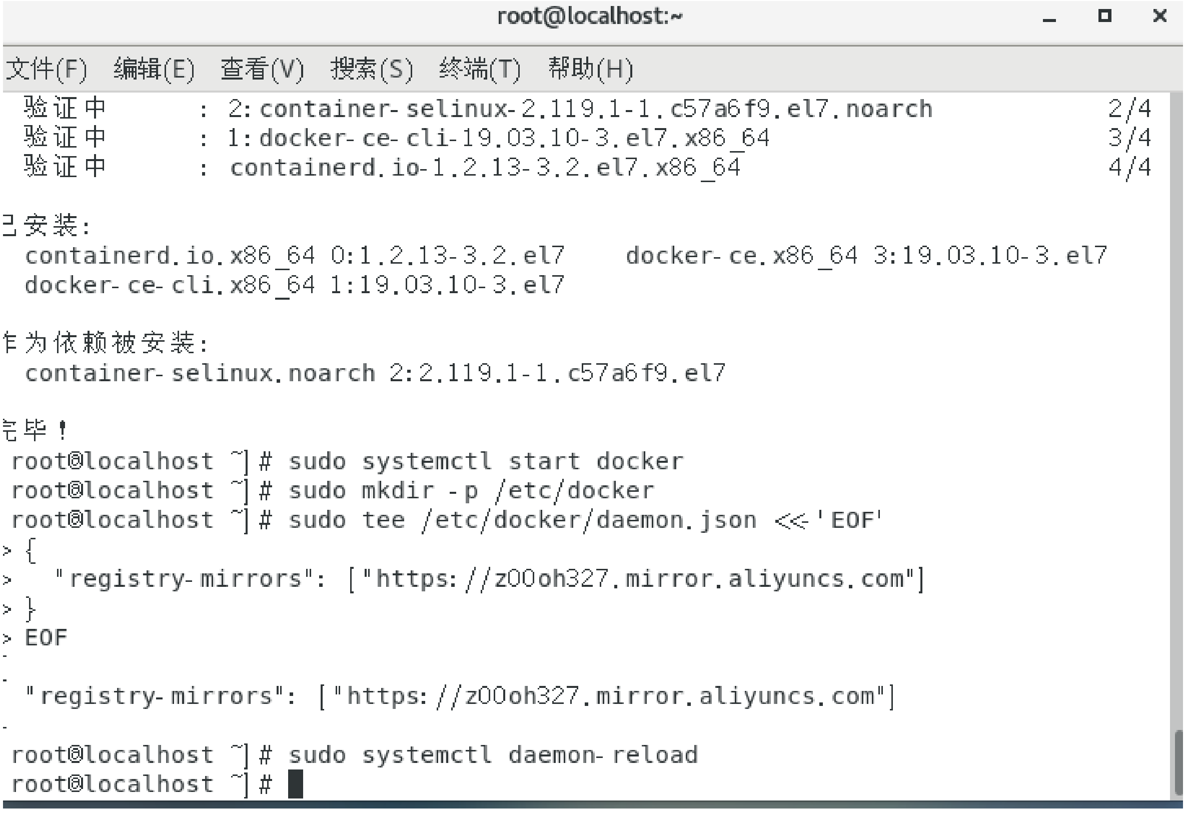
```
Loading mirror speeds from cached hostfile
 * base: mirrors.bfsu.edu.cn
 * extras: mirrors.cn99.com
 * updates: mirrors.cn99.com
docker-ce-stable                                            | 3.5 kB     00:00
(1/2): docker-ce-stable/x86_64/updateinfo                   |   55 B     00:00
(2/2): docker-ce-stable/x86_64/primary_db                   |   43 kB    00:00
正在解决依赖关系
--> 正在检查事务
---> 软件包 containerd.io.x86_64.0.1.2.13-3.2.el7 将被 安装
--> 正在处理依赖关系 container-selinux >= 2:2.74，它被软件包 containerd.io-1.
3-3.2.el7.x86_64 需要
---> 软件包 docker-ce.x86_64.3.19.03.10-3.el7 将被 安装
---> 软件包 docker-ce-cli.x86_64.1.19.03.10-3.el7 将被 安装
--> 正在检查事务
---> 软件包 container-selinux.noarch.2.2.119.1-1.c57a6f9.el7 将被 安装
--> 解决依赖关系完成

依赖关系解决

================================================================================
 Package                架构      版本                           源          大
================================================================================
正在安装:
```

2、设置仓库（这里设置的官网）



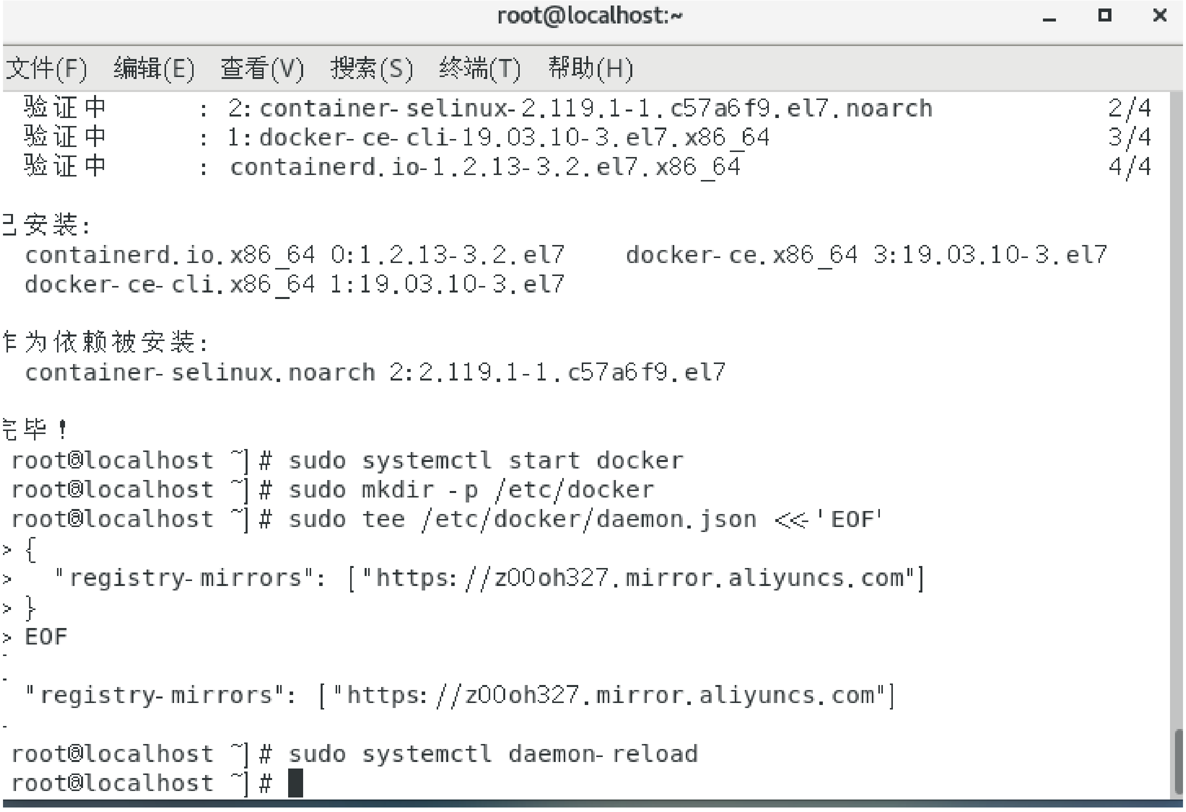
```
更新完毕:
  device-mapper-persistent-data.x86_64 0:0.8.5-2.el7
  lvm2.x86_64 7:2.02.186-7.el7_8.2
  yum-utils.noarch 0:1.1.31-54.el7_8

乍为依赖被升级:
  device-mapper.x86_64 7:1.02.164-7.el7_8.2
  device-mapper-event.x86_64 7:1.02.164-7.el7_8.2
  device-mapper-event-libs.x86_64 7:1.02.164-7.el7_8.2
  device-mapper-libs.x86_64 7:1.02.164-7.el7_8.2
  lvm2-libs.x86_64 7:2.02.186-7.el7_8.2

完毕!
 root@localhost ~]# sudo yum-config-manager \
>  --add-repo \
>  https://download.docker.com/linux/centos/docker-ce.repo
己加载插件: fastestmirror, langpacks
adding repo from: https://download.docker.com/linux/centos/docker-ce.repo
grabbing file https://download.docker.com/linux/centos/docker-ce.repo to /et
n.repos.d/docker-ce.repo
repo saved to /etc/yum.repos.d/docker-ce.repo
 root@localhost ~]# sudo yum install docker-ce docker-ce-cli containerd.io
己加载插件: fastestmirror, langpacks
```

3、安装 Docker 最新版本



```
 验证中      : 2:container-selinux-2.119.1-1.c57a6f9.el7.noarch          2/4
 验证中      : 1:docker-ce-cli-19.03.10-3.el7.x86_64                     3/4
 验证中      : containerd.io-1.2.13-3.2.el7.x86_64                       4/4

己安装:
  containerd.io.x86_64 0:1.2.13-3.2.el7     docker-ce.x86_64 3:19.03.10-3.el7
  docker-ce-cli.x86_64 1:19.03.10-3.el7

乍为依赖被安装:
  container-selinux.noarch 2:2.119.1-1.c57a6f9.el7

完毕!
 root@localhost ~]# sudo systemctl start docker
 root@localhost ~]# sudo mkdir -p /etc/docker
 root@localhost ~]# sudo tee /etc/docker/daemon.json <<'EOF'
> {
>   "registry-mirrors": ["https://z00oh327.mirror.aliyuncs.com"]
> }
> EOF
-
  "registry-mirrors": ["https://z00oh327.mirror.aliyuncs.com"]
-
 root@localhost ~]# sudo systemctl daemon-reload
 root@localhost ~]# 
```

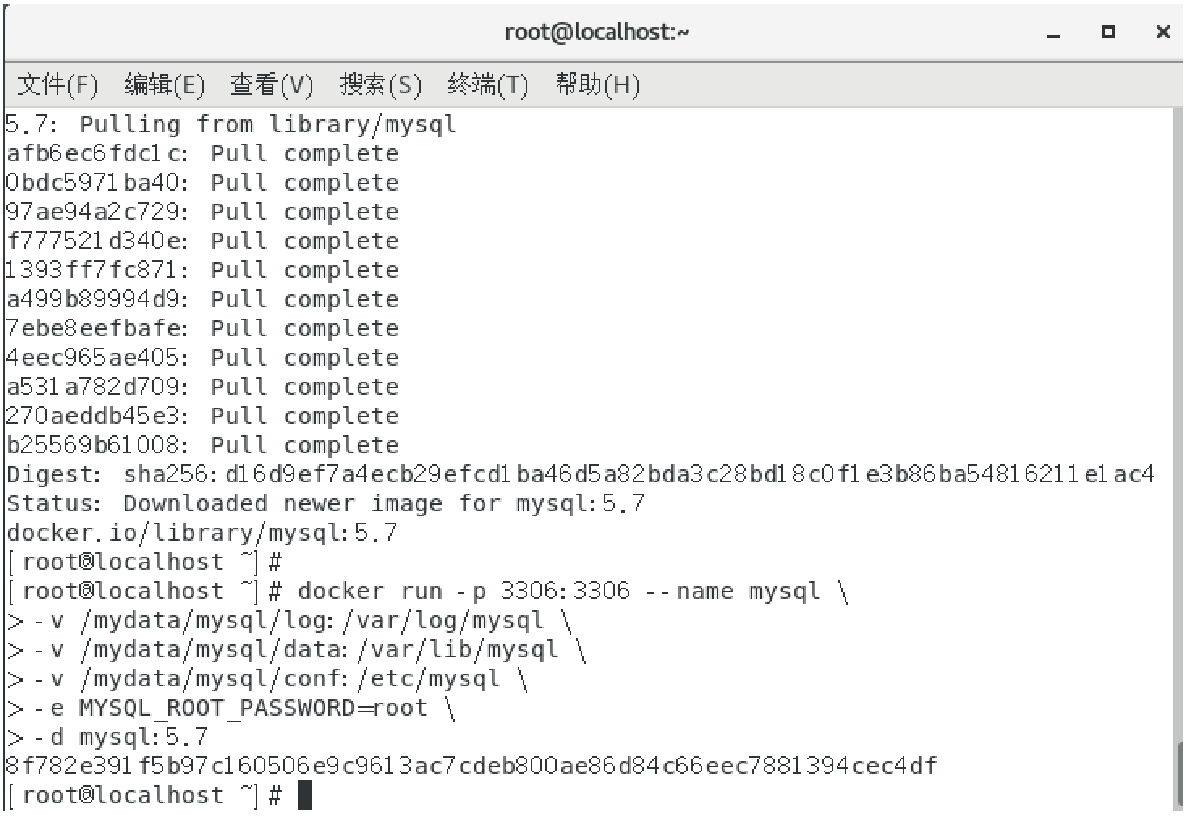
4、配置阿里云的镜像加速并重启服务



```
 验证中      : 2:container-selinux-2.119.1-1.c57a6f9.el7.noarch               2/4
 验证中      : 1:docker-ce-cli-19.03.10-3.el7.x86_64                          3/4
 验证中      : containerd.io-1.2.13-3.2.el7.x86_64                            4/4

己安装:
 containerd.io.x86_64 0:1.2.13-3.2.el7    docker-ce.x86_64 3:19.03.10-3.el7
 docker-ce-cli.x86_64 1:19.03.10-3.el7

乍为依赖被安装:
 container-selinux.noarch 2:2.119.1-1.c57a6f9.el7

宅毕!
 root@localhost ~]# sudo systemctl start docker
 root@localhost ~]# sudo mkdir -p /etc/docker
 root@localhost ~]# sudo tee /etc/docker/daemon.json <<'EOF'
> {
>   "registry-mirrors": ["https://z00oh327.mirror.aliyuncs.com"]
> }
> EOF
-
-
 "registry-mirrors": ["https://z00oh327.mirror.aliyuncs.com"]
-
 root@localhost ~]# sudo systemctl daemon-reload
 root@localhost ~]# 
```

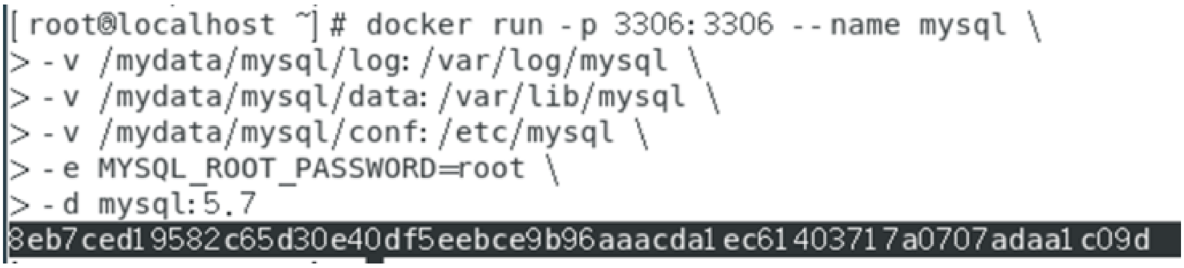
二、在docker中安装mysql

1、安装mysql镜像



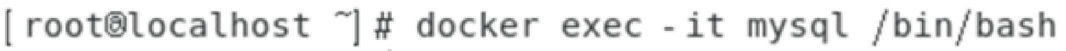
```
5.7: Pulling from library/mysql
afb6ec6fdc1c: Pull complete
0bdc5971ba40: Pull complete
97ae94a2c729: Pull complete
f777521d340e: Pull complete
1393ff7fc871: Pull complete
a499b89994d9: Pull complete
7ebe8eefbafe: Pull complete
4eec965ae405: Pull complete
a531a782d709: Pull complete
270aeddb45e3: Pull complete
b25569b61008: Pull complete
Digest: sha256:d16d9ef7a4ecb29efcd1ba46d5a82bda3c28bd18c0f1e3b86ba54816211e1ac4
Status: Downloaded newer image for mysql:5.7
docker.io/library/mysql:5.7
[root@localhost ~]#
[root@localhost ~]# docker run -p 3306:3306 --name mysql \
> -v /mydata/mysql/log:/var/log/mysql \
> -v /mydata/mysql/data:/var/lib/mysql \
> -v /mydata/mysql/conf:/etc/mysql \
> -e MYSQL_ROOT_PASSWORD=root \
> -d mysql:5.7
8f782e391f5b97c160506e9c9613ac7cdeb800ae86d84c66eec7881394cec4df
[root@localhost ~]# 
```

2、使用dockerm命令启动mysql



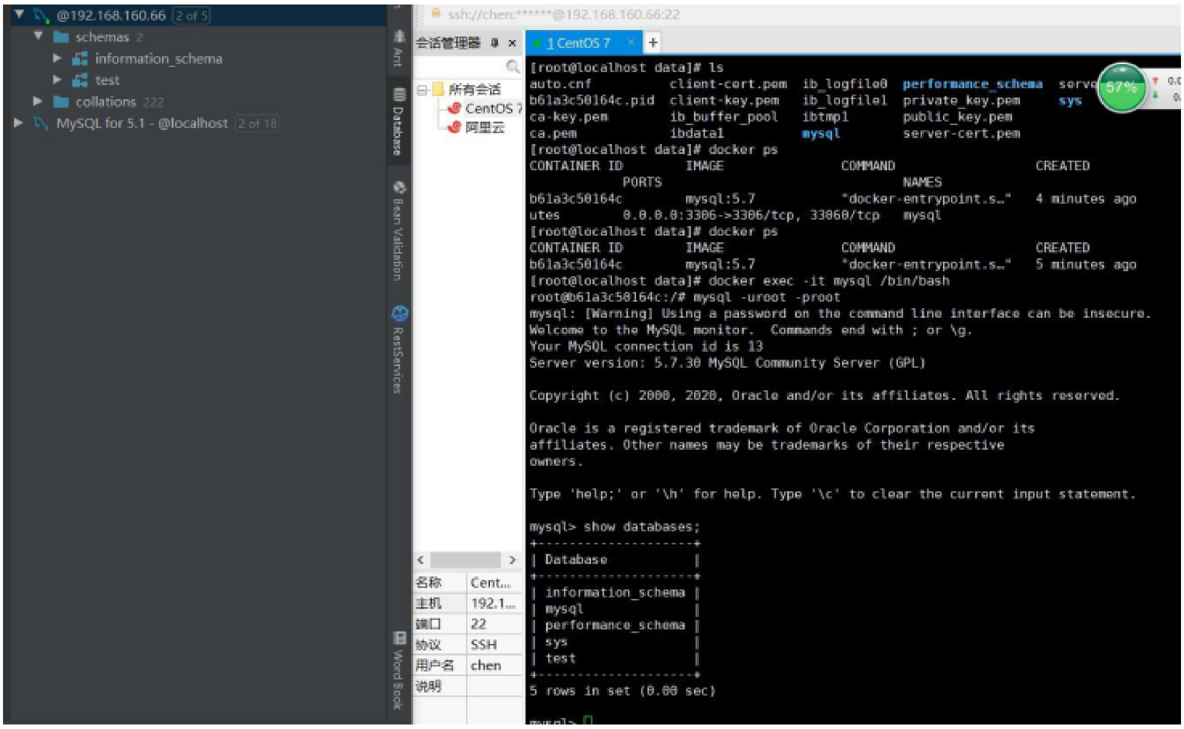
```
[root@localhost ~]# docker run -p 3306:3306 --name mysql \
> -v /mydata/mysql/log:/var/log/mysql \
> -v /mydata/mysql/data:/var/lib/mysql \
> -v /mydata/mysql/conf:/etc/mysql \
> -e MYSQL_ROOT_PASSWORD=root \
> -d mysql:5.7
8eb7ced19582c65d30e40df5eebce9b96aaacdal ec61403717a0707adaal c09d
```

3、进入运行mysql的docker容器



```
[root@localhost ~]# docker exec -it mysql /bin/bash
```

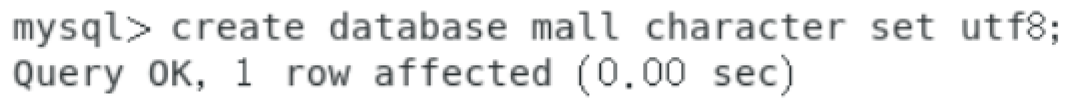
4、idea连接虚拟机中的mysql--并创建数据库--进入Docker与查看



名称 Cent...
主机 192.1...
端口 22
协议 SSH
用户名 chen
说明

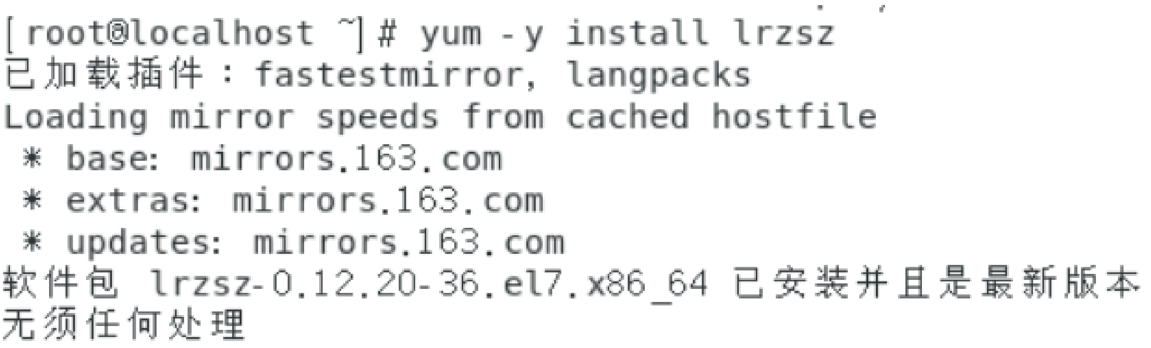
```
[root@localhost data]# ls
auto.cnf            client-cert.pem  ib_logfile0  performance_schema  serve
b61a3c50164c.pid    client-key.pem   ib_logfile1  private_key.pem     sys
ca-key.pem          ib_buffer_pool   ibtmp1       public_key.pem
ca.pem              ibdata1          mysql        server-cert.pem
[root@localhost data]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
          PORTS                               NAMES
b61a3c50164c        mysql:5.7           "docker-entrypoint.s…"   4 minutes ago
utes      0.0.0.0:3306->3306/tcp, 33060/tcp   mysql
[root@localhost data]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
b61a3c50164c        mysql:5.7           "docker-entrypoint.s…"   5 minutes ago
[root@localhost data]# docker exec -it mysql /bin/bash
root@b61a3c50164c:/# mysql -uroot -proot
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 5.7.30 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| test               |
+--------------------+
5 rows in set (0.00 sec)

mysql> 
```

5、创建mall数据库



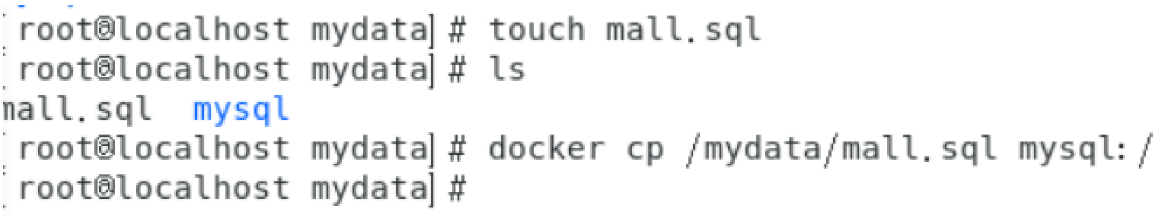
```
mysql> create database mall character set utf8;
Query OK, 1 row affected (0.00 sec)
```

6、安装上传下载插件，并将docment/sql/mall.sql上传到Linux服务器上



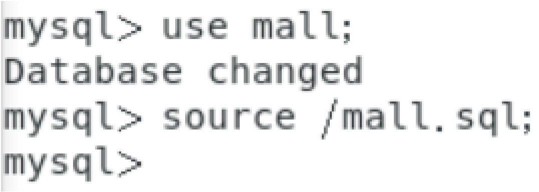
```
[root@localhost ~]# yum -y install lrzsz
已加载插件：fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.163.com
 * extras: mirrors.163.com
 * updates: mirrors.163.com
软件包 lrzsz-0.12.20-36.el7.x86_64 已安装并且是最新版本
无须任何处理
```

7.将mall.sql文件拷贝到mysql容器的/目录下



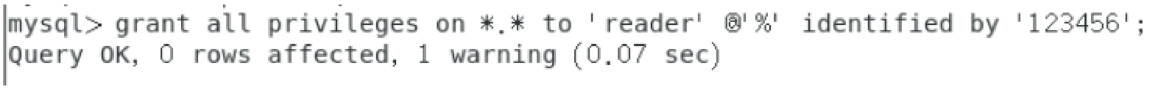
```
root@localhost mydata] # touch mall.sql
root@localhost mydata] # ls
nall.sql  mysql
root@localhost mydata] # docker cp /mydata/mall.sql mysql:/
root@localhost mydata] #
```

8.将sql文件导入到数据库



```
mysql> use mall;
Database changed
mysql> source /mall.sql;
mysql>
```

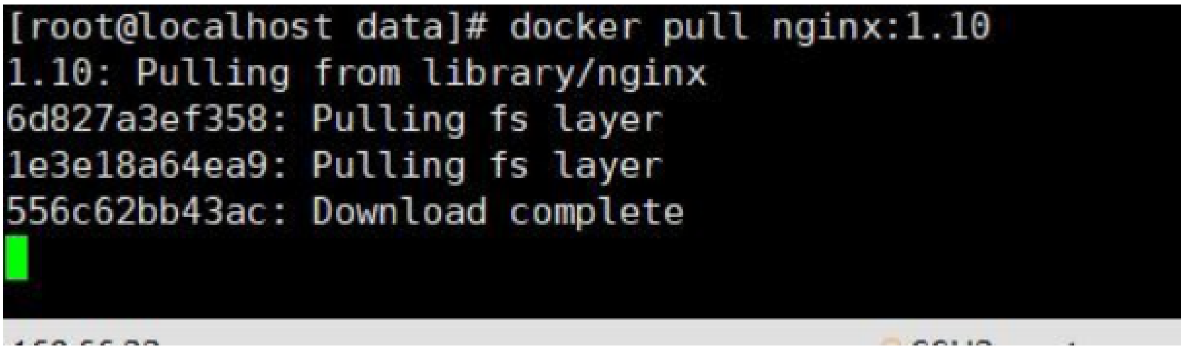
9.创建一个reader帐号使得任何ip都能访问



```
mysql> grant all privileges on *.* to 'reader' @'%' identified by '123456';
Query OK, 0 rows affected, 1 warning (0.07 sec)
```

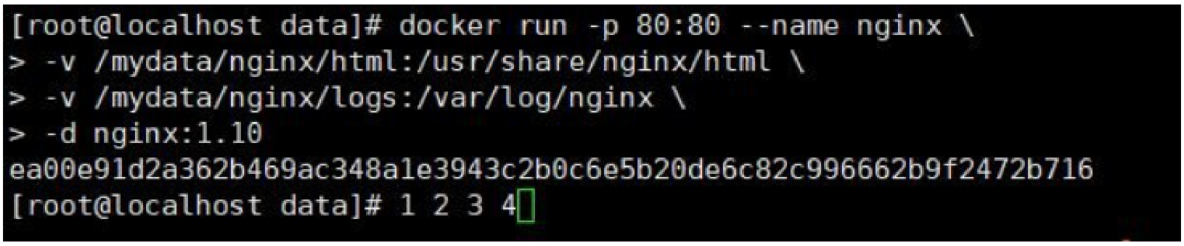
三、在docker中安装Nginx

1、下载Nginx1.10的docker镜像



```
[root@localhost data]# docker pull nginx:1.10
1.10: Pulling from library/nginx
6d827a3ef358: Pulling fs layer
1e3e18a64ea9: Pulling fs layer
556c62bb43ac: Download complete
```

2、将容器内的配置文件拷贝到指定目录：



```
[root@localhost data]# docker run -p 80:80 --name nginx \
> -v /mydata/nginx/html:/usr/share/nginx/html \
> -v /mydata/nginx/logs:/var/log/nginx \
> -d nginx:1.10
ea00e91d2a362b469ac348a1e3943c2b0c6e5b20de6c82c996662b9f2472b716
[root@localhost data]# 1 2 3 4
```