

Docker 安装 MySQL

方法一、docker pull mysql

查找Docker Hub上的mysql镜像

```
runoob@runoob:/mysql$ docker search mysql
```

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
mysql	MySQL is a widely used, open-source relati...	2529	[OK]	
mysql/mysql-server	Optimized MySQL Server Docker images. Crea...	161		[OK]
centurylink/mysql	Image containing mysql. Optimized to be li...	45		[OK]
sameersbn/mysql		36		[OK]
google/mysql	MySQL server for Google Compute Engine	16		[OK]
appcontainers/mysql	Centos/Debian Based Customizable MySQL Con...	8		[OK]
marvambass/mysql	MySQL Server based on Ubuntu 14.04	6		[OK]
drupaldocker/mysql	MySQL for Drupal	2		[OK]
azukiapp/mysql	Docker image to run MySQL by Azuki - http:...	2		[OK]
...				

这里我们拉取官方的镜像,标签为5.6

```
runoob@runoob:~/mysql$ docker pull mysql:5.6
```

等待下载完成后,我们就可以在本地镜像列表里查到REPOSITORY为mysql,标签为5.6的镜像。

```
runoob@runoob:~/mysql$ docker images |grep mysql
```

mysql	5.6	2c0964ec182a	3 weeks ago	329 MB
-------	-----	--------------	-------------	--------

方法二、通过 Dockerfile构建

创建Dockerfile

首先, 创建目录mysql,用于存放后面的相关东西。

```
runoob@runoob:~$ mkdir -p ~/mysql/data ~/mysql/logs ~/mysql/conf
```

data目录将映射为mysql容器配置的数据文件存放路径

logs目录将映射为mysql容器的日志目录

conf目录里的配置文件将映射为mysql容器的配置文件

进入创建的mysql目录, 创建Dockerfile

```
FROM debian:jessie

# add our user and group first to make sure their IDs get assigned consistently, regardless of whatever
dependencies get added
RUN groupadd -r mysql && useradd -r -g mysql mysql

# add gosu for easy step-down from root
ENV GOSU_VERSION 1.7
RUN set -x \
    && apt-get update && apt-get install -y --no-install-recommends ca-certificates wget && rm -rf /var/
lib/apt/lists/* \
    && wget -O /usr/local/bin/gosu "https://github.com/tianon/gosu/releases/download/$GOSU_VERSION/gosu
-$(dpkg --print-architecture)" \
    && wget -O /usr/local/bin/gosu.asc "https://github.com/tianon/gosu/releases/download/$GOSU_VERSION/g
osu-$(dpkg --print-architecture).asc" \
    && export GNUPGHOME="$(mktemp -d)" \
    && gpg --keyserver ha.pool.sks-keyservers.net --recv-keys B42F6819007F00F88E364FD4036A9C25BF357DD4 \
    && gpg --batch --verify /usr/local/bin/gosu.asc /usr/local/bin/gosu \
    && rm -r "$GNUPGHOME" /usr/local/bin/gosu.asc \
    && chmod +x /usr/local/bin/gosu \
    && gosu nobody true \
    && apt-get purge -y --auto-remove ca-certificates wget

RUN mkdir /docker-entrypoint-initdb.d

# FATAL ERROR: please install the following Perl modules before executing /usr/local/mysql/scripts/mysql
_install_db:
# File::Basename
# File::Copy
# Sys::Hostname
# Data::Dumper
RUN apt-get update && apt-get install -y perl pwgen --no-install-recommends && rm -rf /var/lib/apt/list
s/*

# gpg: key 5072E1F5: public key "MySQL Release Engineering <mysql-build@oss.oracle.com>" imported
RUN apt-key adv --keyserver ha.pool.sks-keyservers.net --recv-keys A4A9406876FCBD3C456770C88C718D3B5072E
1F5

ENV MYSQL_MAJOR 5.6
ENV MYSQL_VERSION 5.6.31-1debian8

RUN echo "deb http://repo.mysql.com/apt/debian/ jessie mysql-${MYSQL_MAJOR}" > /etc/apt/sources.list.d/m
ysql.list

# the "/var/lib/mysql" stuff here is because the mysql-server postinst doesn't have an explicit way to d
isable the mysql_install_db codepath besides having a database already "configured" (ie, stuff in /var/l
ib/mysql/mysql)
# also, we set debconf keys to make APT a little quieter
```

```

RUN { \
    echo mysql-community-server mysql-community-server/data-dir select ''; \
    echo mysql-community-server mysql-community-server/root-pass password ''; \
    echo mysql-community-server mysql-community-server/re-root-pass password ''; \
    echo mysql-community-server mysql-community-server/remove-test-db select false; \
} | debconf-set-selections \
&& apt-get update && apt-get install -y mysql-server="${MYSQL_VERSION}" && rm -rf /var/lib/apt/lists/* \
&& rm -rf /var/lib/mysql && mkdir -p /var/lib/mysql /var/run/mysqld \
&& chown -R mysql:mysql /var/lib/mysql /var/run/mysqld \
# ensure that /var/run/mysqld (used for socket and lock files) is writable regardless of the UID our mysqld instance ends up having at runtime
&& chmod 777 /var/run/mysqld

# comment out a few problematic configuration values
# don't reverse lookup hostnames, they are usually another container
RUN sed -Ei 's/^(bind-address|log)/#&/' /etc/mysql/my.cnf \
&& echo 'skip-host-cache\nskip-name-resolve' | awk '{ print } $1 == "[mysqld]" && c == 0 { c = 1; system("cat") }' /etc/mysql/my.cnf > /tmp/my.cnf \
&& mv /tmp/my.cnf /etc/mysql/my.cnf

VOLUME /var/lib/mysql

COPY docker-entrypoint.sh /usr/local/bin/
RUN ln -s usr/local/bin/docker-entrypoint.sh /entrypoint.sh # backwards compat
ENTRYPOINT ["docker-entrypoint.sh"]

EXPOSE 3306
CMD ["mysqld"]

```

通过Dockerfile创建一个镜像，替换成你自己的名字

```
runoob@runoob:~/mysql$ docker build -t mysql .
```

创建完成后，我们可以在本地的镜像列表里查找到刚刚创建的镜像

```
runoob@runoob:~/mysql$ docker images |grep mysql
```

mysql	5.6	2c0964ec182a	3 weeks ago	329 MB
-------	-----	--------------	-------------	--------

使用mysql镜像

运行容器

```
runoob@runoob:~/mysql$ docker run -p 3306:3306 --name mymysql -v $PWD/conf:/etc/mysql/conf.d -v $PWD/logs:/logs -v $PWD/data:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=123456 -d mysql:5.6
```

```
21cb89213c93d805c5bacf1028a0da7b5c5852761ba81327e6b99bb3ea89930e
runoob@runoob:~/mysql$
```

命令说明：

- **-p 3306:3306**：将容器的 3306 端口映射到主机的 3306 端口。
- **-v -v \$PWD/conf:/etc/mysql/conf.d**：将主机当前目录下的 conf/my.cnf 挂载到容器的 /etc/mysql/my.cnf。
- **-v \$PWD/logs:/logs**：将主机当前目录下的 logs 目录挂载到容器的 /logs。
- **-v \$PWD/data:/var/lib/mysql**：将主机当前目录下的 data 目录挂载到容器的 /var/lib/mysql。
- **-e MYSQL_ROOT_PASSWORD=123456**：初始化 root 用户的密码。

查看容器启动情况

```
runoob@runoob:~/mysql$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	...	PORTS	NAMES
21cb89213c93	mysql:5.6	"docker-entrypoint.sh"	...	0.0.0.0:3306->3306/tcp	mymysql

← Docker 安装 Nginx

Docker 安装 PHP →



2 篇笔记

写笔记



最新官方MySQL(5.7.19)的docker镜像在创建时映射的配置文件目录有所不同，在此记录并分享给大家：

官方原文：

The MySQL startup configuration is specified in the file `/etc/mysql/my.cnf`, and that file in turn includes any files found in the `/etc/mysql/conf.d` directory that end with `.cnf`. Settings in files in this directory will augment and/or override settings in `/etc/mysql/my.cnf`. If you want to use a customized MySQL configuration, you can create your alternative configuration file in a directory on the host machine and then mount that directory location as `/etc/mysql/conf.d` inside the `mysql` container.

大概意思是说：

MySQL(5.7.19)的默认配置文件是 `/etc/mysql/my.cnf` 文件。如果想要自定义配置，建议向 `/etc/mysql/conf.d` 目录中创建 `.cnf` 文件。新建的文件可以任意起名，只要保证后缀名是 `cnf` 即可。新建的文件中的配置项可以覆盖 `/etc/mysql/my.cnf` 中的配置项。

具体操作：

首先需要创建将要映射到容器中的目录以及 `.cnf` 文件，然后再创建容器

```
# pwd
/opt
# mkdir -p docker_v/mysql/conf
# cd docker_v/mysql/conf
# touch my.cnf
# docker run -p 3306:3306 --name mysql -v /opt/docker_v/mysql/conf:/etc/mysql/conf.d -e MYSQL_ROOT_PASSWORD=123456 -d imageID
4ec4f56455ea2d6d7251a05b7f308e314051fdad2c26bf3d0f27a9b0c0a71414
```

命令说明：

- **-p 3306:3306**：将容器的3306端口映射到主机的3306端口
- **-v /opt/docker_v/mysql/conf:/etc/mysql/conf.d**：将主机/opt/docker_v/mysql/conf目录挂载到容器的/etc/mysql/conf.d
- **-e MYSQL_ROOT_PASSWORD=123456**：初始化root用户的密码
- **-d**：后台运行容器，并返回容器ID
- **imageID**：mysql镜像ID

查看容器运行情况

```
# docker ps
```

CONTAINER ID	IMAGE	COMMAND	... PORTS	NAMES
4ec4f56455ea	c73c7527c03a	"docker-entrypoint.sh"	... 0.0.0.0:3306->3306/tcp	mysql

Brian 2年前 (2017-09-08)



docker 安装 mysql 8 版本

```
# docker 中下载 mysql
docker pull mysql

#启动
docker run --name mysql -p 3306:3306 -e MYSQL_ROOT_PASSWORD=Lzslov123! -d mysql

#进入容器
docker exec -it mysql bash

#登录mysql
mysql -u root -p
ALTER USER 'root'@'localhost' IDENTIFIED BY 'Lzslov123!';

#添加远程登录用户
CREATE USER 'liao zesong'@'%' IDENTIFIED WITH mysql_native_password BY 'Lzslov123!';
GRANT ALL PRIVILEGES ON *.* TO 'liao zesong'@'%';
```

liao zesong 8个月前 (07-30)

