



链滴

详解使用 DockerHub 官方的 mysql 镜像生成容器

作者: [unhappydepig](#)

原文链接: <https://ld246.com/article/1482733643679>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<div>

<p>写在前面：看到网上关于利用DockerHub官方的mysql镜像生成容器之类的文档比较少，故结合自身实践分享给大家，还望多多指教。</p>

<p>我的需求：利用docker 镜像快速建立一个mysql容器。</p>

<p>于是我去找度娘，谷歌，看了好多以官方centos镜像为基础，制作基于centos的mysql镜像，但制作后发现镜像大小已经超过1G，对于一般的docker镜像来说太臃肿了。</p>

<p>自己私下想：不至于吧，mysql的镜像这么大？然后去仓库看了下</p>

<p>https://hub.docker.com/_/mysql/ 英文水平比较好的直接看这链接文档来就好了，嘿嘿）</p>

<p>言归正传 -></p>

<p> 我自己隐约觉得，用官方的就好了，因为这已经暂时满足我的需求了。其他人自己制作mysql镜像有他们的特殊需求吧！ 所以我就先使用官方的了！ </p>

<p>拉下来一看Docker Hub目前官方提供的mysql镜像才380.2MB。</p>

<p>一、下载官方的mysql镜像</p>

```
class="hljs elixir"><code class="language-bash hljs">[root@localhost ~]<span class="hljs-comment"><span class="hljs-comment"># docker pull mysql</span></span></code></pre>
```

<p>下载完成后查看镜像：</p>

```
class="hljs elixir"><code class="language-bash hljs">[root@localhost ~]<span class="hljs-comment"><span class="hljs-comment"># docker images</span></span>
REPOSITORY          TAG          IMAGE ID      CREATED      SIZE
docker.io/mysql     latest      <span class="hljs-number">1195</span>b21c3a45
                   <span class="hljs-number">10</span> weeks ago   <span class="hljs-number">380.2</span>
span> MB</code></pre>
```

<p> </p>

<p>二、生成容器 </p>

<p>生成容器时，调用的启动脚本是/entrypoint.sh;通过查看/entrypoint.sh脚本内容，总结参数如</p>

```
class="hljs php"><code class="language-bash hljs">mysqld <span class="hljs-comment"><span class="hljs-comment">#启动mysql服务，必须使用</span></span>
```

```
MYSQL_ROOT_PASSWORD <span class="hljs-comment"><span class="hljs-comment">#设mysql的root密码，必须使用</span></span>
```

```
<span class="hljs-comment"><span class="hljs-comment">#以下二个参数添加除root之外的户并设置密码，可选。</span></span>
```

```
MYSQL_USER
```

```
MYSQL_PASSWORD
```

```
<span class="hljs-comment"><span class="hljs-comment">#设置生成容器时需要新建的数据，可选</span></span>
```

```
MYSQL_DATABASE
```

```
<span class="hljs-comment"><span class="hljs-comment">#容器的mysql数据库默认的保存径是:</span></span>
```

```
<span class="hljs-keyword">var</span>/lib/mysql
```

```
<span class="hljs-comment"><span class="hljs-comment">#容器的配置文件my.cnf的路径为</span></span>
```

```
<span class="hljs-keyword">var</span>/etc/mysql/my.cnf</code></pre>
```

<p>使用上要的参数生成新的容器：</p>

```
class="hljs elixir"><code class="language-bash hljs">[root@localhost ~]<span class="hljs-comment"><span class="hljs-comment"># docker run -d -p 3307:3306 --name mysql -P -e mysqld -e MYSQL_ROOT_PASSWORD=12356 -e MYSQL_USER=yope -e MYSQL_PASSWORD=yope -e MYSQL_DATABASE=testDb -v /mysql_data:/var/lib/mysql mysql </span></span>
```

```
<span class="hljs-number">492</span> ffa26d8653561208aed28eb62c61e9bae0de91ef9117
```

9c35f9e4eae6b272

```
[root@localhost ~]# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
492	ffa26d865	mysql	seconds ago	Up 7 seconds

```
mysql
```

重要参数说明:

`--name mysql`

指定容器名称为 `mysql`;

`-p 3307:3306`

指定映射端口 将宿主主机端口3307映射到容器3306端口;

`-v /mysql_data:/var/lib/mysql`

映射数据库存放位置。将宿主机的目录 `/mysql_data` 映射到容器的 `/var/lib/mysql` 目录这是因为默认情况下数据库的数据库文件和日志文件都会存放于容器的AUFS文件层，这不仅不使得器变得越来越臃肿，不便于迁移、备份等管理，而且数据库的性能也会受到影响。因此建议挂载到宿机的目录到容器内。

查看本地的 `mysql_data` 是否生成容器内的数据库:

```
root@localhost /# ll /mysql_data
```

总用量 188452

```
-rw-r----- 1 systemd-bus-proxy ssh_keys 56 8月 20 22:25 auto.cnf
```

```
-rw-r----- 1 systemd-bus-proxy ssh_keys 1329 8月 20 22:25 ib_buffer_pool
```

```
-rw-r----- 1 systemd-bus-proxy ssh_keys 79691776 8月 20 22:25 ibdata1
```

```
-rw-r----- 1 systemd-bus-proxy ssh_keys 50331648 8月 20 22:25 ib_logfile0
```

```
-rw-r----- 1 systemd-bus-proxy ssh_keys 50331648
```

```

tor-tag">8</span>月 <span class="hljs-selector-tag">20</span> <span class="hljs-selector-tag">22</span><span class="hljs-selector-pseudo">:25</span> <span class="hljs-selector-tag">ib_logfile1</span>
<span class="hljs-selector-tag">-rw-r-----</span> <span class="hljs-selector-tag">1</span>
<span class="hljs-selector-tag">systemd-bus-proxy</span> <span class="hljs-selector-tag">ssh_keys</span> <span class="hljs-selector-tag">12582912</span> <span class="hljs-selector-tag">8</span>月 <span class="hljs-selector-tag">20</span> <span class="hljs-selector-tag">22</span><span class="hljs-selector-pseudo">:25</span> <span class="hljs-selector-tag">ibtmp1</span>
<span class="hljs-selector-tag">drwxr-x---</span> <span class="hljs-selector-tag">2</span>
<span class="hljs-selector-tag">systemd-bus-proxy</span> <span class="hljs-selector-tag">ssh_keys</span> <span class="hljs-selector-tag">4096</span> <span class="hljs-selector-tag">8</span>月 <span class="hljs-selector-tag">20</span> <span class="hljs-selector-tag">22</span><span class="hljs-selector-pseudo">:25</span> <span class="hljs-selector-tag">mysql</span>
<span class="hljs-selector-tag">drwxr-x---</span> <span class="hljs-selector-tag">2</span>
<span class="hljs-selector-tag">systemd-bus-proxy</span> <span class="hljs-selector-tag">ssh_keys</span> <span class="hljs-selector-tag">8192</span> <span class="hljs-selector-tag">8</span>月 <span class="hljs-selector-tag">20</span> <span class="hljs-selector-tag">22</span><span class="hljs-selector-pseudo">:25</span> <span class="hljs-selector-tag">performance_schema</span>
<span class="hljs-selector-tag">drwxr-x---</span> <span class="hljs-selector-tag">2</span>
<span class="hljs-selector-tag">systemd-bus-proxy</span> <span class="hljs-selector-tag">ssh_keys</span> <span class="hljs-selector-tag">8192</span> <span class="hljs-selector-tag">8</span>月 <span class="hljs-selector-tag">20</span> <span class="hljs-selector-tag">22</span><span class="hljs-selector-pseudo">:25</span> <span class="hljs-selector-tag">sys</span>
<span class="hljs-selector-tag">drwxr-x---</span> <span class="hljs-selector-tag">2</span>
<span class="hljs-selector-tag">systemd-bus-proxy</span> <span class="hljs-selector-tag">ssh_keys</span> <span class="hljs-selector-tag">19</span> <span class="hljs-selector-tag">8</span>月 <span class="hljs-selector-tag">20</span> <span class="hljs-selector-tag">22</span><span class="hljs-selector-pseudo">:25</span> <span class="hljs-selector-tag">test</span></span></span><span class="hljs-selector-tag">Db</span>

```

```

<span class="hljs-selector-attr">[root@localhost /]</span><span class="hljs-comment"># </span>

```

已经生成相关的文件和默认数据库，同时也新建了"testDb"数据库。

测试方案一：

在宿主机上通过端口访问容器中的mysql服务

```

[root@localhost /]<span class="hljs-comment"># yum -y i
stall mysql

```

测试登录：

```

[root@localhost mysql_data]<span class="hljs-comment">mysql -u root -p -h 127.0.0.1 -P 3307

```

```

Enter password:

```

```

Welcome to the MariaDB monitor. Commands end with
; or \g.

```

```

Your MySQL connection id is 2

```

```

Server version: 5.7.13
MySQL Community Server (GPL)

```

```

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

```

Type `'help;'/> or 'h' for help. Type 'c' to clear the current input statement.`

MySQL [(none)]> </code> </pre>

<p>测试方案二: </p>

<p>自己本地利用mysql 连接工具 (eg: Navicat 等) 进行连接, 注意映射出来的真实端口不一定是306就好了。</p>

<p>需要注意的是: 宿主机上的UID和GID是999。这是容器里用户mysql的UID和GID。千万进行权限变更, 否则容器对这个目录进行读写就会出现问題。如果觉得不舒服, 可以在本地新建一个mysql_docker的用户指定UID和GID是999。</p>

<p> (这个注意本人没有落实验证, 如果描写信息有错, 还望指出) </p>

<p> </p>

<p>三、进入到新生成的容器 (名为mysql的容器) </p>

<p>使用exec进入容器, 同时进行相关操作: </p>

```
class="hljs elixir"><code class="language-bash hljs">[root<span class="hljs-variable">@localhost</span> /]<span class="hljs-comment"><span class="hljs-comment"># docker exec -it mysql /bin/bash</span></span></code></pre>
```

<p>查看进程: </p>

```
class="hljs less"><code class="language-bash hljs">root<span class="hljs-variable">@92ffa26d865:</span>/<span class="hljs-comment"># ps -aux</span>
USER      PID %CPU %MEM  VSZ  RSS TTY   STAT START  TIME COMMAND
mysql     <span class="hljs-number">1</span> <span class="hljs-number">0.2</span> <span class="hljs-number">9.7</span> <span class="hljs-number">1262696</span> <span class="hljs-number">182200</span> ?    Ssl <span class="hljs-number">14</span>:<span class="hljs-number">52</span> <span class="hljs-number">0</span>:<span class="hljs-number">00</span> mysql
root      <span class="hljs-number">127</span> <span class="hljs-number">0.2</span> <span class="hljs-number">0.1</span> <span class="hljs-number">20216</span> <span class="hljs-number">1884</span> ?    Ss <span class="hljs-number">14</span>:<span class="hljs-number">57</span> <span class="hljs-number">0</span>:<span class="hljs-number">00</span> /bin/bash
root      <span class="hljs-number">133</span> <span class="hljs-number">0.0</span> <span class="hljs-number">0.0</span> <span class="hljs-number">17492</span> <span class="hljs-number">1148</span> ?    R+ <span class="hljs-number">14</span>:<span class="hljs-number">58</span> <span class="hljs-number">0</span>:<span class="hljs-number">00</span> ps -aux</code></pre>
```

<p>查看数据库所在的文件夹: </p>

```
class="hljs less"><code class="language-bash hljs">root<span class="hljs-variable">@92ffa26d865:</span>/<span class="hljs-comment"># ls -l /var/lib/mysql/</span>
total <span class="hljs-number">188452</span>
-rw-r----- <span class="hljs-number">1</span> mysql mysql <span class="hljs-number">56</span> Aug <span class="hljs-number">20</span> <span class="hljs-number">14</span>:<span class="hljs-number">52</span> auto.cnf
-rw-r----- <span class="hljs-number">1</span> mysql mysql <span class="hljs-number">329</span> Aug <span class="hljs-number">20</span> <span class="hljs-number">14</span>:<span class="hljs-number">52</span> ib_buffer_pool
-rw-r----- <span class="hljs-number">1</span> mysql mysql <span class="hljs-number">5031648</span> Aug <span class="hljs-number">20</span> <span class="hljs-number">14</span>:<span class="hljs-number">52</span> ib_logfile0
-rw-r----- <span class="hljs-number">1</span> mysql mysql <span class="hljs-number">5031648</span> Aug <span class="hljs-number">20</span> <span class="hljs-number">14</span>:<span class="hljs-number">52</span> ib_logfile1
```



```
-rw-r----- <span class="hljs-number">1</span> mysql mysql <span class="hljs-number">79
91776</span> Aug <span class="hljs-number">20</span> <span class="hljs-number">14</span>
<span class="hljs-number">52</span> ibdata1
-rw-r----- <span class="hljs-number">1</span> mysql mysql <span class="hljs-number">12
82912</span> Aug <span class="hljs-number">20</span> <span class="hljs-number">14</span>
<span class="hljs-number">52</span> ibtmp1
drwxr-x--- <span class="hljs-number">2</span> mysql mysql <span class="hljs-number">
4096</span> Aug <span class="hljs-number">20</span> <span class="hljs-number">14</span>
<span class="hljs-number">52</span> mysql
drwxr-x--- <span class="hljs-number">2</span> mysql mysql <span class="hljs-number">
8192</span> Aug <span class="hljs-number">20</span> <span class="hljs-number">14</span>
<span class="hljs-number">52</span> performance_schema
drwxr-x--- <span class="hljs-number">2</span> mysql mysql <span class="hljs-number">
8192</span> Aug <span class="hljs-number">20</span> <span class="hljs-number">14</span>
<span class="hljs-number">52</span> sys
drwxr-x--- <span class="hljs-number">2</span> mysql mysql <span class="hljs-number">
19</span> Aug <span class="hljs-number">20</span> <span class="hljs-number">14</span>
<span class="hljs-number">52</span> <span class="hljs-built_in">test</span><span class="hljs-built_in">Db</span></code></pre>
```

<p>进入mysql:</p>

```
<pre class="hljs elixir"><code class="language-bash hljs">root<span class="hljs-variable">
492ffa26d865</span> <span class="hljs-symbol">:</span> <span class="hljs-comment"><
an class="hljs-comment"># mysql -u root -p</span></span>
Enter <span class="hljs-symbol">password:</span>
Welcome to the MySQL monitor. Commands <span class="hljs-keyword">end</span> with ;
<span class="hljs-keyword">or</span> \g.
Your MySQL connection id is <span class="hljs-number">3</span>
Server <span class="hljs-symbol">version:</span> <span class="hljs-number">5.7</span>.
<span class="hljs-number">13</span> MySQL Community Server (GPL)
Copyright (c) <span class="hljs-number">2000</span>, <span class="hljs-number">2016</span>
<span class="hljs-keyword">and</span>/<span class="hljs-keyword">or</span>
n> its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation <span class="hljs-keyword">and</span>
n>/<span class="hljs-keyword">or</span> its
affiliates. Other names may be trademarks of their respective
owners.
Type <span class="hljs-string"><span class="hljs-string">'help;'</span></span> <span class="hljs-keyword">or</span> <span class="hljs-string"><span class="hljs-string">'\h'</span>
</span> <span class="hljs-keyword"><span class="hljs-keyword">for</span></span> help.
Type <span class="hljs-string"><span class="hljs-string">'\c'</span></span> to clear the cu
rent input statement.
mysql</code></pre>
```

<p> </p>

<p> </p>

<p>四、修改容器中mysql的配置文件 </p>

<p>默认镜像中并没有安装vi或者vim，需要手工安装</p>

```
<pre class="hljs less"><code class="language-bash hljs">root<span class="hljs-variable">@
92ffa26d865:</span> <span class="hljs-comment"># apt-get update &amp;&amp; apt-get
yq install vim</span></code></pre>
```

<p>然后就可以使用vim 来修改mysql的配置文件: </p>

```
<pre class="hljs less"><code class="language-bash hljs">root<span class="hljs-variable">@
92ffa26d865:</span> <span class="hljs-comment"># vim /etc/mysql/my.cnf</span></cod
></pre>
```

<p>vim编辑器在此不做详述。 </p>

<p>如果已经有一个比较成熟的my.cnf的配置方案，可以在宿主机上新建一个文件夹将已经设置好的

y.cnf和conf.d这二个文件放到里面, </p>

<p>然后在新建容器的时候, 直接使用参数" -v" 将这个文件夹映射到容器的" /etc/mysql" 目录即可。 </p>

<p>The End </p>

</div>

<div> </div>