

Docker 常用安装与镜像发布

作者: [zouchanglin](#)

原文链接: <https://ld246.com/article/1564887771946>

来源网站: [链滴](#)

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

Docker安装MySQL

下面将使用Docker来安装Mysql, 先搜索, 然后拉取:

```
[root@localhost ~]# docker pull mysql
```

```
[root@localhost ~]# docker run -p 3306:3306 --restart=always --name mysql -v /root/mysql/conf:/etc/mysql/conf.d -v /root/mysql/logs:/logs -v /root/mysql/data:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=123456 -d mysql
```

```
77de7fb81728c289744d4adba8db2d4f60124dd0955eea5a7b6bcd4e6277b138
```

```
[root@localhost ~]# docker exec -it x
```

```
"docker exec" requires at least 2 arguments.
```

```
See 'docker exec --help'.
```

```
□
```

```
Usage: docker exec [OPTIONS] CONTAINER COMMAND [ARG...]
```

```
□
```

Run a command in a running container

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

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
77de7fb81728	mysql	"docker-entrypoint.s..."	18 seconds ago	Up 17 seconds
0.0.0.0:3306->3306/tcp, 33060/tcp	mysql			

```
□
```

进入mysql容器

```
[root@localhost ~]# docker exec -it 77de7fb81728 /bin/bash
```

```
root@77de7fb81728:/# mysql -uroot -p
```

```
Enter password:
```

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

```
Your MySQL connection id is 8
```

```
Server version: 8.0.17 MySQL Community Server - GPL
```

```
□
```

```
Copyright (c) 2000, 2019, 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.
```

```
□
```

解决外部工具连接不上的问题

```
mysql> ALTER USER 'root'@'%' IDENTIFIED BY '123456' PASSWORD EXPIRE NEVER;  
Query OK, 0 rows affected (0.02 sec)
```

□

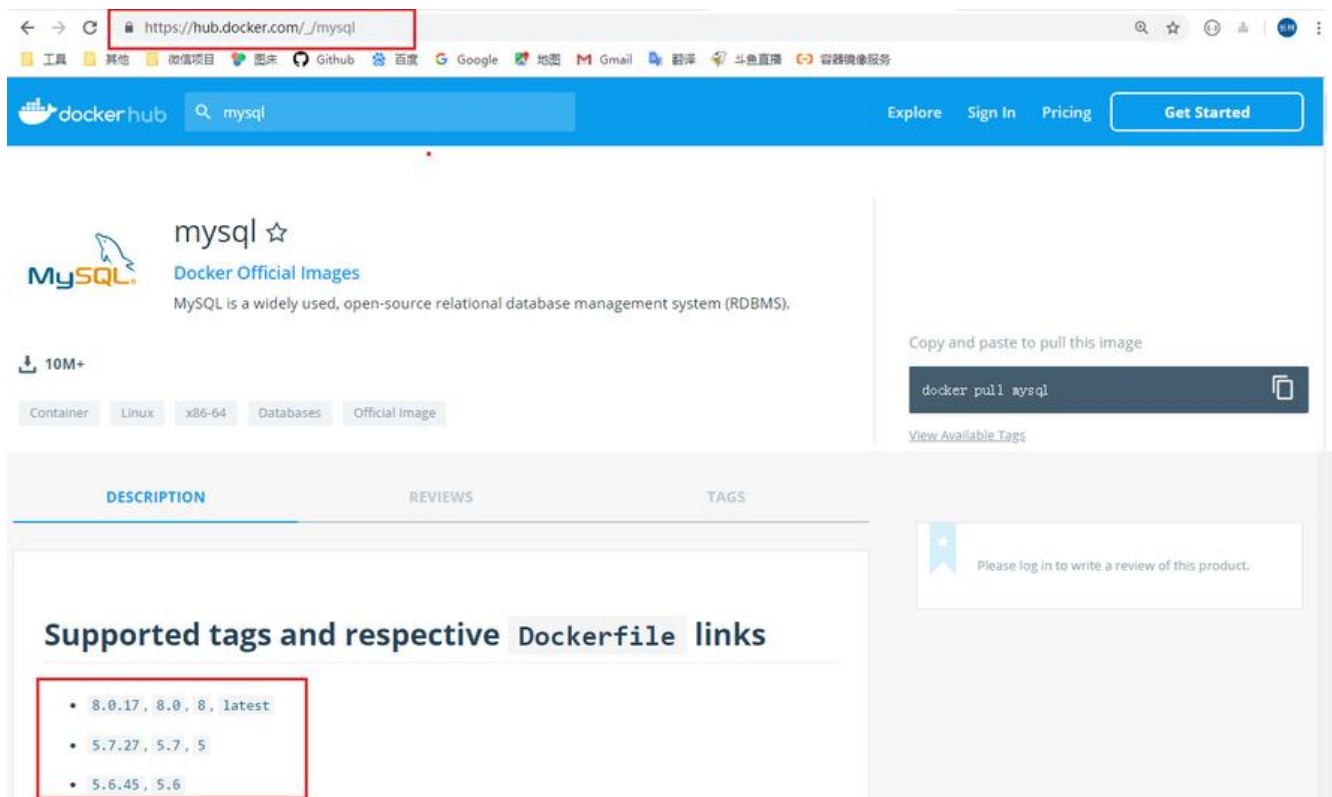
```
mysql> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.01 sec)
```

□

```
mysql>
```

Docker安装MySQL需要注意的问题:

1、搜索时在 <https://hub.docker.com/> 这个网站进行搜索，比如搜索MySQL



下面就有版本号，默认是最新的，也就是latest

2、`docker run -p 3306:3306 --restart=always --name mysql -v /root/mysql/conf:/etc/mysql/conf.d -v /root/mysql/logs:/logs -v /root/mysql/data:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=123456 -d mysql` 这条命令的意思：其实很简单--restart=always是让容器在Docker服务启动也跟随启动，我的Docker设置的是开机启动，所以开机后一旦Docker服务启动，mysql容器也会启动，通过-v命令添加了三个容器卷，分别是~/mysql/conf、~/mysql/logs、mysql/data，这样的话我们在宿主机上就可以配置MySQL、查看log、以及备份数据！

Docker安装Tomcat

同样的安装Tomcat也是先查找，然后直接根据自己的版本号直接`docker pull tomcat:标签`

```
docker run -it -p 8080:8080 --restart=always --name tomcat -v /root/tomcat/webapps:/usr/local/tomcat/webapps -v /root/tomcat/logs:/usr/local/tomcat/logs tomcat /bin/bash
```

不知道为啥，这个Tomcat需要手动启动，所以使用 -it 模式进入bin目录start.sh才可以开启Tomcat
另外，如果像上面这个路径去添加容器卷的话，webapp和logs目录会清空，我是直接把webapps的西copy进去就好了！

Docker安装Redis

同样的安装redis也是先查找，然后直接根据自己的版本号直接`docker pull redis:标签`

```
docker run -p 6379:6379 -v /root/redis/data:/data -v /root/redis/conf/redis.conf:/usr/local/etc/redis/redis.conf -d redis:3.2 redis-server /usr/local/etc/redis/redis.conf --appendonly yes
```

在主机/root/redis/conf/redis.conf目录下新建redis.conf文件 vim /root/redis/conf/redis.conf/redis.conf

配置文件如下：

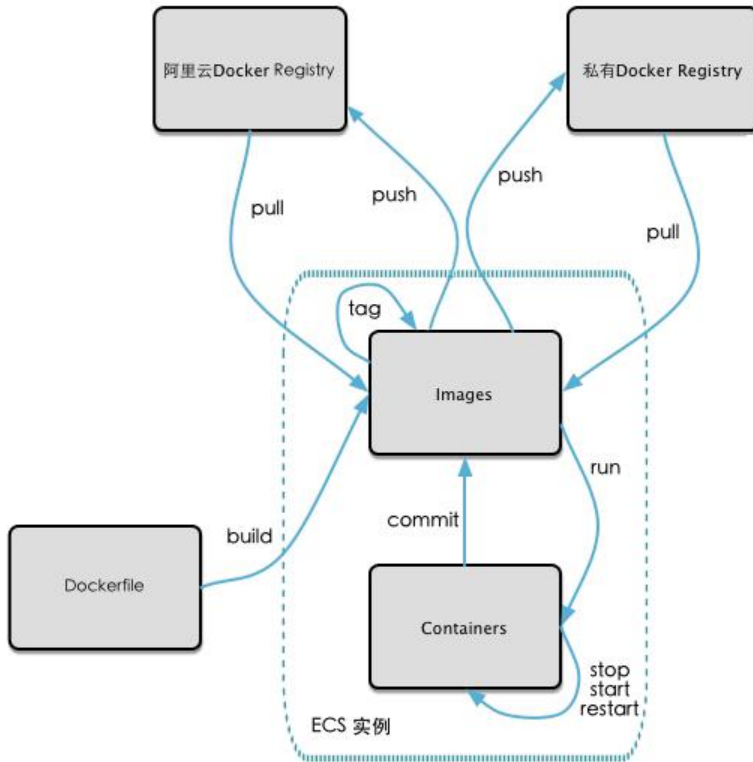
测试redis-cli连接和测试持久化文件生成

```
docker exec -it 容器编号 redis-cli
```

```
[root@localhost ~]# docker exec -it 7f9c53bde0c6 redis-cli
127.0.0.1:6379> set k1 v1
OK
127.0.0.1:6379> set k2 v2
OK
127.0.0.1:6379> set k3 v3
OK
127.0.0.1:6379> get k3
"v3"
127.0.0.1:6379> SHUTDOWN
[root@localhost data]# ls
appendonly.aof
[root@localhost data]# cat appendonly.aof
*2
$6
SELECT
$1
0
*3
$3
set
$2
k1
$2
v1
```

Docker镜像发布到阿里云

阿里云ECS Docker生态如下图所示：



镜像的生成方法，编写DockerFile，这个在上一篇文章DockerFile已经说过了。或者从容器生成一个的镜像：

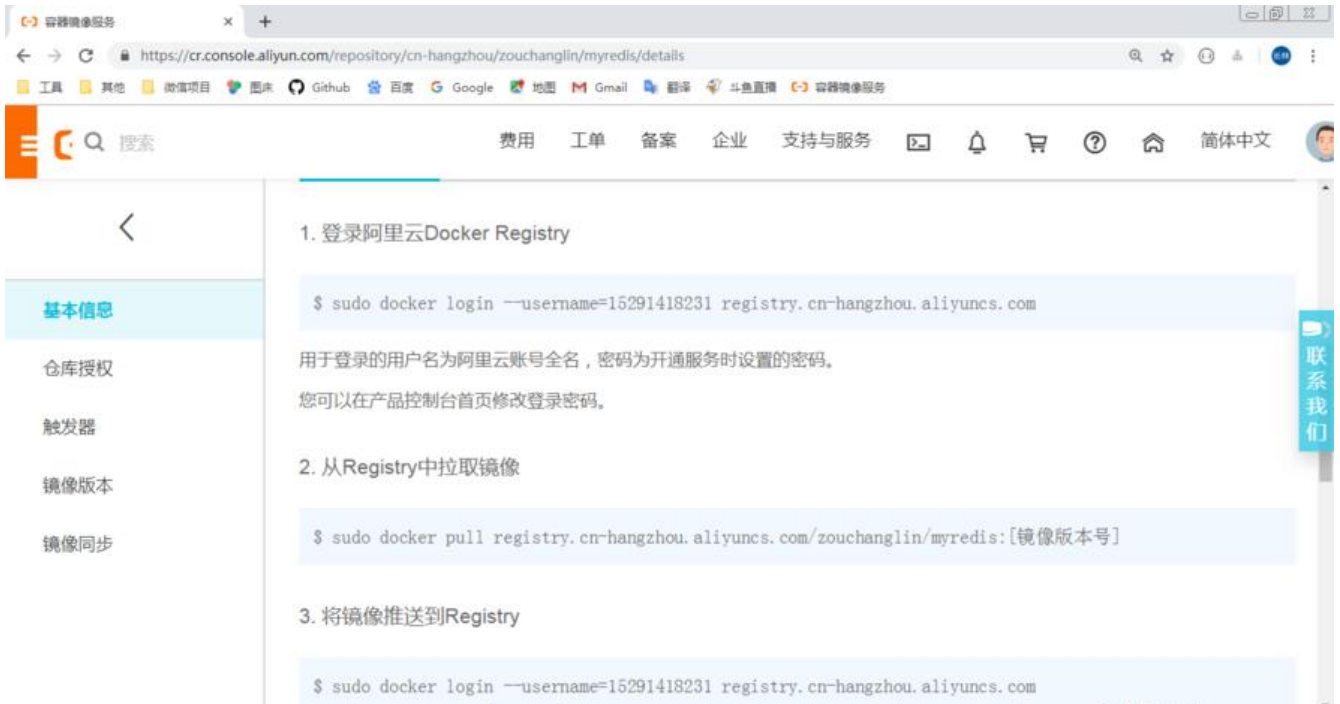
```
docker commit [OPTIONS] 容器ID [REPOSITORY[:TAG]]
```

登录到阿里云的镜像服务：<https://cr.console.aliyun.com/cn-hangzhou/instances/repositories>

创建一个命名空间，创建本地仓库：



将镜像推送到阿里云，推送方式阿里云文档里面就有



按照提示操作，先登录

```
[root@localhost ~]# docker login --username=15291418231 registry.cn-hangzhou.aliyuncs.com
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

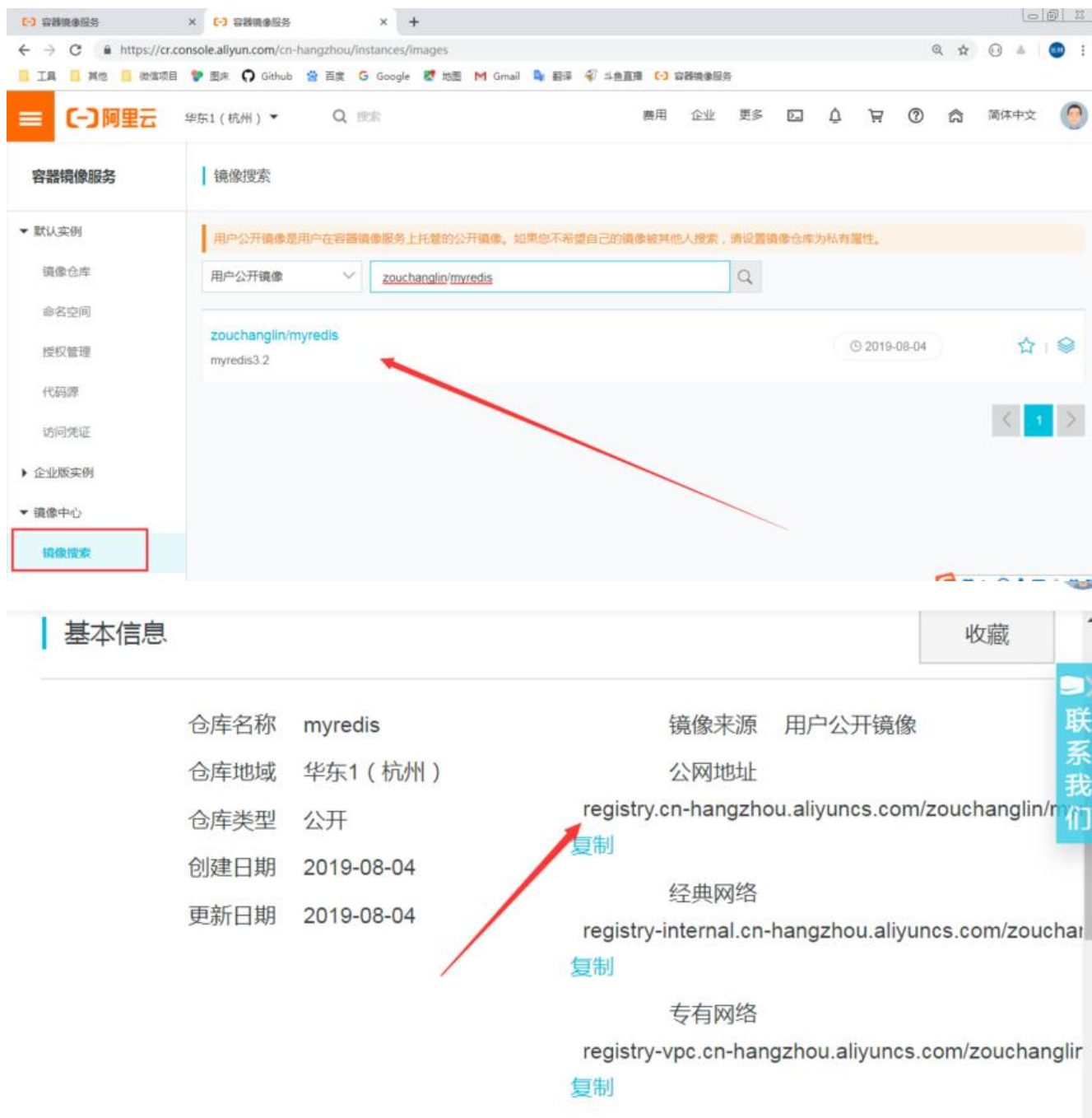
登录成功后，由于我之前没还制作镜像，现在开始制作镜像并且推送镜像

```
[root@localhost ~]# docker commit -m "test aliyun push redis3.2" -a="zouchanglin" 7f9c53bde0c6 zouchanglin/myredis:1.1
sha256:a9545e9e7754e691f8612e92d14a42c642b39b925896b9df84ca08df4dcb837a
[root@localhost ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
zouchanglin/myredis 1.1                a9545e9e7754       6 seconds ago     76MB
mysql               latest             2151acc12881       12 days ago       445MB
tomcat              latest             238e6d7313e3       2 weeks ago       506MB
redis               3.2                87856cc39862       9 months ago      76MB
[root@localhost ~]# docker tag a9545e9e7754 registry.cn-hangzhou.aliyuncs.com/zouchanglin/myredis:1.1
[root@localhost ~]# docker push registry.cn-hangzhou.aliyuncs.com/zouchanglin/myredis:1.1
The push refers to repository [registry.cn-hangzhou.aliyuncs.com/zouchanglin/myredis]
9556e83fbb15: Pushed
56431c543d6c: Pushed
bb617143299d: Pushed
cfe17e3394d7: Pushing 17.43MB
aa1a19279a9a: Pushed
197ffb073b01: Pushed
237472299760: Pushing 525.3kB/55.27MB
```

出现下面这种，即说明推送完毕（其实这和Pull是一样的，镜像分层都是一层一层的结构）

```
[root@localhost ~]# docker push registry.cn-hangzhou.aliyuncs.com/zouchanglin/myredis:1.1
The push refers to repository [registry.cn-hangzhou.aliyuncs.com/zouchanglin/myredis]
9556e83fbb15: Pushed
56431c543d6c: Pushed
bb617143299d: Pushed
cfe17e3394d7: Pushed
aa1a19279a9a: Pushed
197ffb073b01: Pushed
237472299760: Pushed
1.1: digest: sha256:26a1056a645db729f0d47cf6c2137b6b62233ca5462c6b4963657a4b62c82933 size: 1778
[root@localhost ~]#
```

在阿里云执行搜索，一定要是公有仓库才能搜到，下面看看我的Redis



这样就和那些mysql镜像一样了，可以直接下载，先删除本地的镜像，然后再拉取

```
1 Local_CentOS7 [0] 2 Local_CentOS7 [1] 3 Local_CentOS7 [2] +
[root@localhost ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
mysql                latest              2151acc12881       12 days ago        445MB
tomcat               latest              238e6d7313e3       2 weeks ago        506MB
redis                3.2                 87856cc39862       9 months ago       76MB
[root@localhost ~]# docker pull registry.cn-hangzhou.aliyuncs.com/zouchanglin/myredis:1.1
1.1: Pulling from zouchanglin/myredis
f17d81b4b692: Already exists
b32474098757: Already exists
8980cabe8bc2: Already exists
58af19693e78: Already exists
a977782cf22d: Already exists
9c1e268980b7: Already exists
b08192a79931: Pull complete
Digest: sha256:26a1056a645db729f0d47cf6c2137b6b62233ca5462c6b4963657a4b62c82933
Status: Downloaded newer image for registry.cn-hangzhou.aliyuncs.com/zouchanglin/myredis:1.1
registry.cn-hangzhou.aliyuncs.com/zouchanglin/myredis:1.1
[root@localhost ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
registry.cn-hangzhou.aliyuncs.com/zouchanglin/myredis    1.1                 a9545e9e7754       21 minutes ago     76MB
```