



链滴

# linux 环境下 docker 的安装与使用

作者: [WangShijie](#)

原文链接: <https://ld246.com/article/1537803448859>

来源网站: 链滴

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

安装docker:

由于国内访问不到docker官方镜像的缘故,可以通过aliyun的源来完成:

```
yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.
epo
```

查看docker版本:

```
yum list docker-ce --showduplicates
```

安装最新版docker:

```
sudo yum install docker-ce
```

设置docker开机启动:

```
systemctl enable docker
cp /usr/lib/systemd/system/docker.service /etc/systemd/system/
systemctl daemon-reload
systemctl restart docker
```

或者:

```
chkconfig docker on
```

启动/停止/状态docker服务:

```
service docker start/stop/status
```

搜索镜像:

```
docker search {}
```

拉取镜像:

```
docker pull {}(mysql/tomcat/redis/nginx等)
```

查看本机所有镜像:

```
docker images
```

启动容器:

```
docker run -d -e MYSQL_ROOT_PASSWORD=sj19940429 --restart=always --name mysql -v /
ata/mysql/data:/var/lib/mysql -p 3306:3306 mysql:8.0
```

Docker容器的重启策略如下:

no, 默认策略, 在容器退出时不重启容器

on-failure, 在容器非正常退出时(退出状态非0), 才会重启容器

on-failure:3, 在容器非正常退出时重启容器, 最多重启3次

always, 在容器退出时总是重启容器

unless-stopped, 在容器退出时总是重启容器, 但是不考虑在Docker守护进程启动时就已经停止了容器

如果在run时未指定, 可以用update指定:

```
docker update --restart=always {container_id}
```

查看运行中的容器:

```
docker ps
```

查看所有容器:

```
docker ps -a
```

停止所有容器:

```
docker stop $(docker ps -a -q)
```

运行/停止某个容器:

```
docker start/stop {repository}
```

删除容器:

```
docker rm {container_id}
```

-f, --force=false:强制终止并删除一个运行中的容器。

-l, --link=false:删除容器的连接, 但保留容器。

-v, --volumes=false:删除容器挂载的数据卷。

删除镜像:

```
docker rmi {image_id}
```

删除image\_id相同的镜像:

```
docker rmi {repository}:{tag}
```

进入容器:

```
docker exec -it {container_id} /bin/bash
```

-t: 进入终端

-i: 获得一个交互式的连接, 通过获取container的输入

/bin/bash: 在container中启动一个bash shell

退出容器:

```
exit
```

启动docker tomcat:

```
启动tomcat: docker run -d --name my-tomcat --restart=always -p 8080:8080 tomcat:latest
```

# 若端口被占用, 可以指定容器和主机的映射端口 前者是外围访问端口: 后者是容器内部端口

```
部署war包: docker cp {*}.war my-tomcat:/usr/local/tomcat/webapps
```

启动docker nginx:

```
启动nginx: docker run -p 80:80 --name nginx --restart=always -v /usr/local/nginx/www:/www
```

```
v /usr/local/nginx/conf/nginx.conf:/etc/nginx/nginx.conf --privileged=true -v /usr/local/nginx
```

```
logs:/wwwlogs -v /usr/local/nginx/html:/etc/nginx/html -d nginx:latest
```

在docker中安装vim:

```
apt-get update
```

```
apt-get install vim
```