## Docker 使用说明

**环境**: fedola 23

**Docker 版本:** 17.06.0-ce

**镜像地址:** https://hub.docker.com

## 常用参数:

-i (interractive)打开STDIN用于控制台交互

-t (tty)分配tty设备,用户控制台交互

-u (user)指定容器用户

-e (env)指定环境变量

-p (publish)暴露端口

-h (hostname)指定容器主机名

-v (volume)挂载目录

**常用命令:**

启动/停用docker服务: service start/stop docker

查看镜像: docker images

查看容器: docker ps

查看未启动容器:docker ps -a

进入容器: docker exec 容器名 -it bash

开启/停止容器 docker start/stop 容器名

删除容器 docker rm 容器名

**Mysql 安装(按步骤执行以下命令):**

**1.(**创建容器**)** docker run -it --name=mysql -p 3306:3306 -v /home/mysql/data:/var/lib/mysql -e MYSQL\_ROOT\_PASSWORD=root mysql

2.(乱码问题) 进入mysql 容器 执行:

echo "character-set-server=utf8" >> /etc/mysql/mysql.conf.d/mysqld.cnf 更改编码为utf-8

**3.**重启容器 (docker restart mysql)或者mysql(在mysql容器内执行 service mysql restart)

**Zookeeper安装(按步骤执行以下命令):**

**1.**(创建容器) docker run -it --name=zookeeper -p 2181:2181  zookeeper

**Redis安装(按步骤执行以下命令):**

1.(创建容器) docker run -it --name=redis -p 6379:6379 -v /home/redis/data:/data redis

**Kafka容器:**

由于kafka暂时没有适用镜像,可自己制作容器(也可制作自己的镜像,有兴趣可自己试验,见后文),并且kafka依赖java,以及zookeeper环境 ,所以我们下载一个java镜像作为环境(zookeeper我们已经安装过了)**]**

**下载**[kafka](https://www.apache.org/dyn/closer.cgi?path=/kafka/0.10.2.1/kafka_2.12-0.10.2.1.tgz" \t "/home/lgq/文档\\x/_blank)**:wget**[http://apache.mirror.colo-serv.net/kafka/0.10.2.1/kafka\_2.12-0.10.2.1.tgz](http://apache.mirror.colo-serv.net/kafka/0.10.2.1/kafka_2.12-0.10.2.1.tgz" \t "/home/lgq/文档\\x/_blank)**到本地的目录/home/kafka ,并执行 tar -zxvf xxx.tgz解压到当前目录**

1. **修改配置文件:config/server.properties**

**zookeeper.connection.timeout.ms=5000 [连接超时]**

**zookeeper.connect=localhost:2181 [localhost改为zookeeper地址]**

**advertised.host.name=localhost [kafka服务地址,如果kafka消费者跟提供者不同在一台服务器,该地址不能指定为localhost,必须为对外可访问ip或域名]**

**advertised.port=9092**

1. (下载镜像创建容器)docker run -it --name=kafka -p 9092:9092 -v /home/kafka:/root java ;
2. 进入容器进入kafka所在目录,执行 ./bin/kafka-server-start.sh config/server.properties

启动;

(现有kafka服务镜像)

镜像地址:[dounine/goddess-kafka-provider:latest](https://portainer.bjike.com/" \l "/images/dounine~2Fgoddess-kafka-provider:latest/)

Kafka 服务器域名: kafka.host.bjike.com

创建容器步骤(portainer)

1.Add Container :

2.Container name : kafka

3.Tag name : [dounine/goddess-kafka-provider:latest](https://portainer.bjike.com/" \l "/images/dounine~2Fgoddess-kafka-provider:latest/)

**4.Port mapping : 9092:9092**

**5.Command : /kafka/bin/start.sh**

**6.Volume mapping: [**container: /kafka/config: **/root/config] 映射kafka服务器目录上的配置文件方便修改**

**Config说明:**

**/kafka/config/server.properties:**

**检查并修改以下配置**

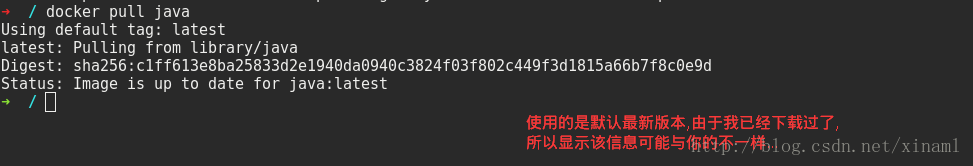
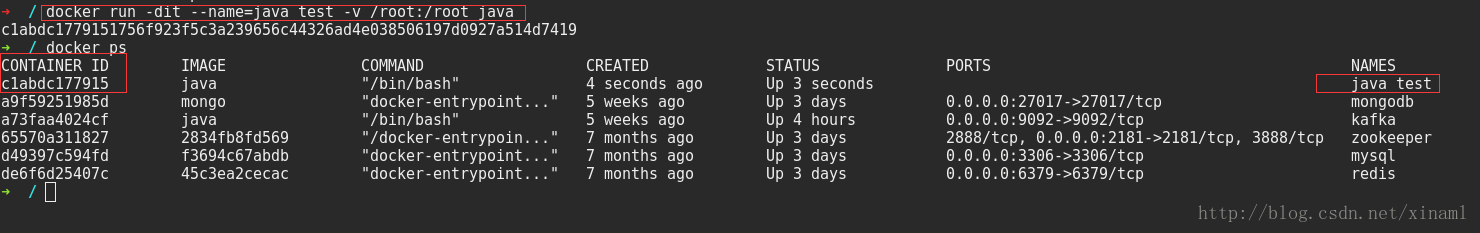
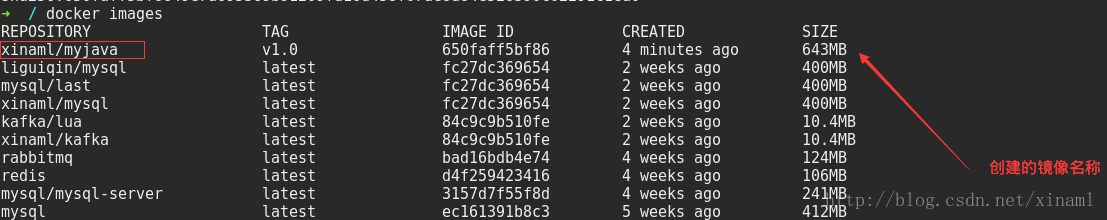
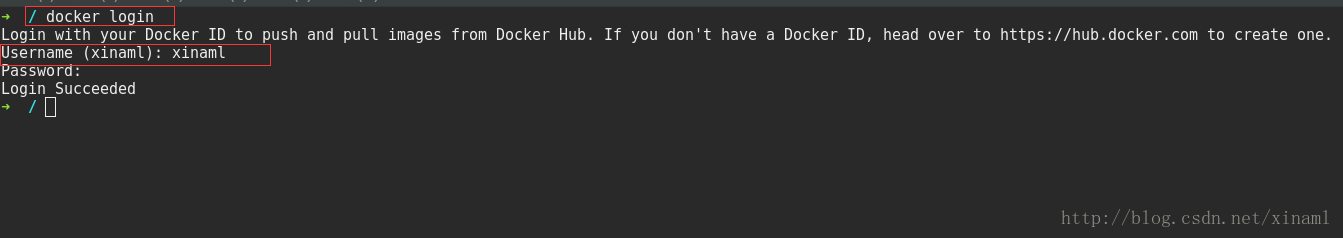
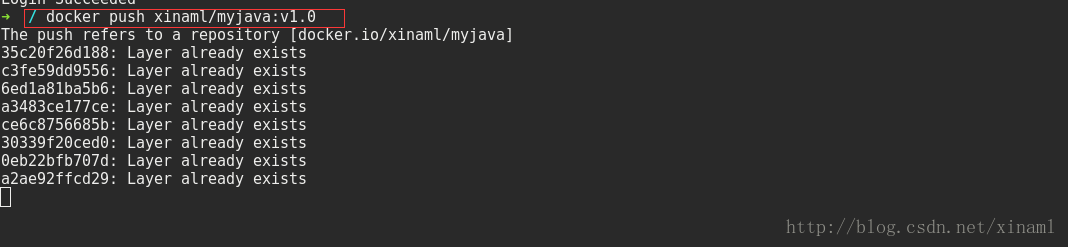
**zookeeper.connect=zookeeper.host.bjike.com:2181 //zookeeper 地址**

**zookeeper.connection.timeout.ms=6000**

**advertised.host.name=kafka.host.bjike.com //kafka提供给消费者的地址**

**advertised.port=9092 //kafka端口**

**创建镜像步骤:**

1.下载镜像:执行[Docker](http://lib.csdn.net/base/docker" \o "Docker知识库" \t "/home/lgq/文档\\x/_blank) pull [Java](http://lib.csdn.net/base/java" \o "Java 知识库" \t "/home/lgq/文档\\x/_blank)命令下载好java镜像   
   
2.创建容器:执行[docker](http://lib.csdn.net/base/docker" \o "Docker知识库" \t "/home/lgq/文档\\x/_blank) run -dit –name=java\_test -v /root:/root java命令   
创建了一个名为 java\_test的容器,映射挂载/root目录并后台运行,创建成功后找到容器id: docker ps 命令查看现有运行的容器及容器基本信息   
   
3.提交镜像: 执行docker commit -a “lgq” -m “test commit” c1abdc177915 xinaml/myjava:v1.0命令提交镜像到本地(这个跟[Git](http://lib.csdn.net/base/git" \o "Git知识库" \t "/home/lgq/文档\\x/_blank)的其实是一样的,先提交镜像到本地,才能推送到你的远程镜像仓库,**一定要注意提交的镜像名格式 帐号/名字:如 xinaml/myjava,否则无法推送**)   
解释以下上面命令的信息:   
-a “lgq”为创建镜像的作者名字 -m “test commit”为提交信息 c1abdc177915 为容器id,xinaml/myjava为镜像名,v1.0为tag   
   
4.执行docker images命令可见镜像创建成功:   
   
5.执行命令:docker login 登录你的 hub.docker 帐号   
   
6.提交到远程仓库(由于镜像较大,就不真正的提交了,提示的信息是我之前已经创建过了一个空的仓库)   
   
7.前往hub仓库查看是否提交成功