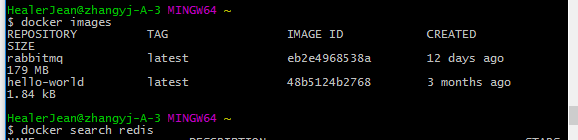
## 1.镜像下载

### docker pull rabbitmq



## 2.查看镜像

## $ docker images



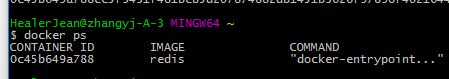
## 3.运行一个redis容器（edis-name自己随意取名）

#### 解释：redis为镜像的名称

## $ docker run -name redis-name -d redis

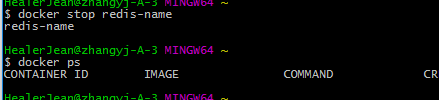
## 4.查看容器列表

### docker ps



## 5.停止容器redis

### $ docker stop redis-name

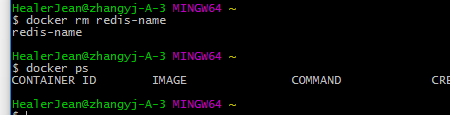


## 6.启动容器(和上面的run其实是一样的)

### docker start redis-name

## 7.删除单个容器

### docker rm redis-name



## 8.端口映射(将容器的6379端口映射到本机的6378端口)

### docker run –d –p 6378:6379 –name redis-name redis

# 9.云部署 docker部署

## 1. 首先打包一个项目 ch10docker-0.0.1-SNAPSHOT.jar

## 2.在jar同级目录下创建一个 Dockerfile文件 (都放到项目根目录下面)

FROM java:8 //java版本为8  
  
MAINTAINER wyf //作者为wrf  
  
ADD ch10docker-0.0.1-SNAPSHOT.jar app.jar //将jar包添加到镜像中，并重命名为app.jar  
  
EXPOSE 8080 //运行镜像中的容器，端口为800  
  
ENTRYPOINT ["java","-jar","/app.jar"] 容器启动时候执行 这个命令

## 3.编译镜像

### $ docker build -t wisely/ch10docker .

#### 解释; wisely为前缀，可以自己取名称 ch10docker为项目的名字，后面的.表示Dockerfile 路径的，表示在当前路径下面

### 3.2测试 docker images

## 3 docker运行 （和基本命令中是一样的，只是顺序不一样而已）

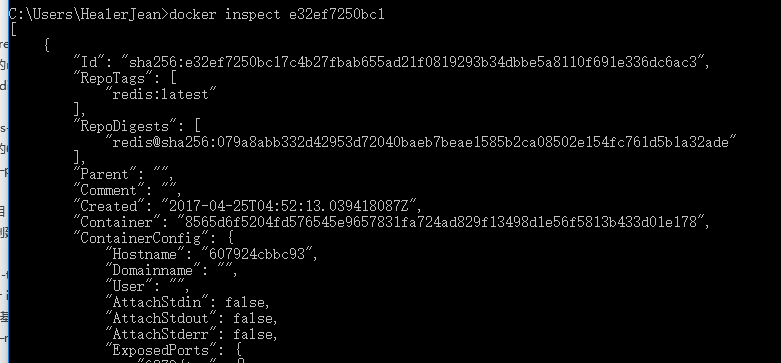
### docker run –d –name ch10 –p 8080:8080 wisely/ch10docker

## 4.测试

## http://localhost:8080/

# 10 获取镜像详细信息 （后面的乱码表示它的唯一id 可以通过images查看）

## docker inspect e32ef7250bc1

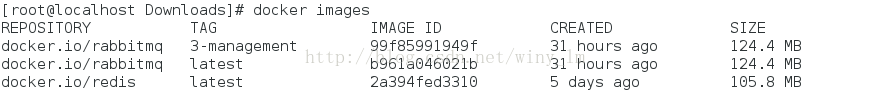


# 11.docker删除镜像

本以为删除镜像会很简单，但是删除过程中并不是那么顺利。

## ****1. 查询镜像****

### ****docker images****



现在想删除第一个，ID为 99f85991949f 的镜像。

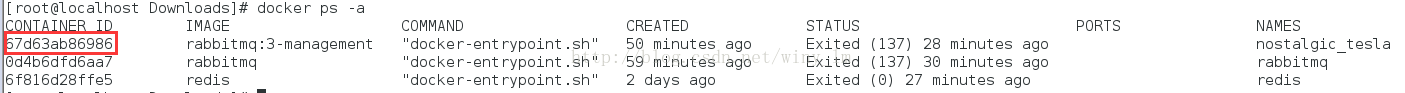
### 删除镜像，docker rmi 99f85991949f

http://img.blog.csdn.net/20170914151152134?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvd2lueV9sbQ==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center

## 解释：从上面看出，需要先停到ID为 67\*\*\* 的容器。

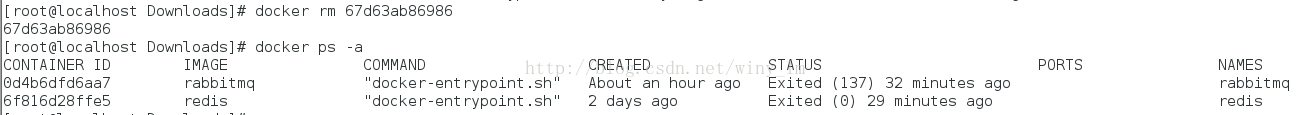
## ****2. 查询容器****

### docker ps -a



## ****3. 先删除容器****

#### docker rm 8464184651

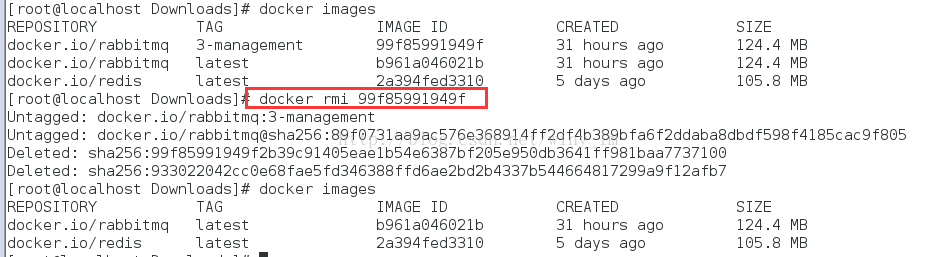


上面可以看出，容器已经删除。

## ****4. 删除镜像****

#### docker images

docker rmi 565656



**注意点：**

**1. 删除前需要保证容器是停止的  stop**

**2. 需要注意删除镜像和容器的命令不一样。 docker rmi ID  ,其中 容器(rm)  和 镜像(rmi)**

**3. 顺序需要先删除容器**