# Docker

## 概念

### 镜像

用于创建容器的只读模板。

### 容器

简易版Linux系统环境与应用程序打包而成的盒子。

基于镜像创建，启动时在镜像的最上层创建一个可写层。

容器是无状态的，不会保存至原有镜像，需要基于该容器重新构建一个镜像。

### 仓库

默认使用DockerHub仓库，概念类似于Github，但服务器在国外，上传下载速度慢。

可使用第三方云平台作为仓库，如[阿里云](https://cs.console.aliyun.com/)。



## 安装Docker

### 2.1 Centos7下安装Docker

**1.centos7安装docker ce**

# 安装依赖包

sudo yum install -y yum-utils device-mapper-persistent-data lvm2

# 添加软件源

sudo yum-config-manager --add-repo https://mirrors.ustc.edu.cn/docker-ce/linux/centos/docker-ce.repo

# 更新缓存/安装docker ce

sudo yum makecache fast

sudo yum install docker-ce

**2.启动docker ce**

sudo systemctl enable docker

sudo systemctl start docker

**3.将docker加入用户组**

sudo groupadd docker

sudo usermod -aG docker $USER

**4.测试docker**

docker run hello-world

参考：[Centos下安装docker](https://yeasy.gitbooks.io/docker_practice/install/centos.html)

## 镜像

### 3.1 镜像加速

利用阿里云容器镜像服务中的加速器来下载镜像。

可以通过docker info来查看是否启用了该加速器。



### 3.2 搜索镜像

在官方提供的镜像中所有你所需要的镜像

docker search --filter=is-official=true nginx

### 3.3 获取镜像

下载镜像至本地

docker pull 镜像名称:版本标签

### 3.4 查看镜像信息

docker inspect 镜像id 或者 镜像名称:标签

### 3.5 删除镜像

建议先删除依赖该镜像的所有容器，再来删除镜像

docker rm 容器id 或者 容器名称

docker rmi 镜像id 或者 镜像名称

或者，强制删除镜像，依赖它的容器会一并删除

docker rmi 镜像id 或者 镜像名称 -f

### 3.6 清理镜像

强制清理镜像

docker image prune -f

## 容器

### 4.1 新建并启动容器

docker run -it 镜像id 或者 镜像名称:便签

-i 标准输入

-t 伪终端

-d 守护状态运行

--name 命名容器

-p 端口映射(宿主机端口:容器端口)

-P 端口映射(随机端口)

-v 文件挂载 (-v $PWD/redis/conf/redis.conf:/etc/redis/redis.conf)

本地文件/路径:容器文件/路径

不论修改宿主机还是容器内挂载的文件，会同时变化，避免要修改容器中文件内容

而需要安装vim或者复制文件至容器的操作。

--link 容器间通信(--link django\_tsdrm:django\_server)

关联的容器名称/id:关联容器服务器地址的别名(自动加载至hosts文件)

### 4.2 进入容器

进入的容器必须已经启动

docker exec -it 容器id 或者 容器名称 /bin/bash

### 4.3 启动容器

docker start 容器id 或者 容器名称

### 4.4 暂停容器

docker pause 容器id 或者 容器名称(unpause取消暂停)

### 4.5 关闭容器

docker stop 容器id 或者 容器名称

### 4.6 查看容器详情

docker inspect 容器id 或者 容器名称

### 4.7 删除容器

docker rm 容器id 或者 容器名称(只删除已经关闭或者退出的容器，-f表示强制)

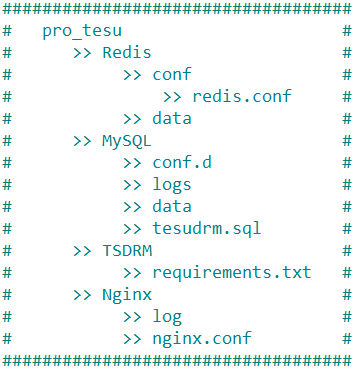
### 4.8 复制文件至容器

docker cp 本地命令执行所在相对路径 容器id/名称：/root/(容器绝对路径)

## 示例

### 5.1 \*本地文件预置

准备的文件目录(在/www/路径下创建)



mkdir /www/pro\_tesu/Redis/{conf, data} -p

mkdir /www/pro\_tesu/MySQL/{conf.d, logs, data} -p

mkdir /www/pro\_tesu/Nginx/log -p

redis.conf，my.conf以及nginx.conf文件的配置查看**附录**。

### 5.2 \*拉取并启动MySQL容器

# 创建容器:宿主机3306端口可能被宿主机占用

docker pull mysql:5.6

cd /www/pro\_tesu/

docker run -p 3307:3306 --name tesudrm\_mysql -v $PWD/MySQL/conf.d:/etc/mysql/conf.d \

-v $PWD/MySQL/logs:/logs \

-v $PWD/MySQL/data:/var/lib/mysql \

-e MYSQL\_ROOT\_PASSWORD=password \

-d mysql:5.6

# 拷贝sql文件至容器

docker cp /www/pro\_tesu/MySQL/tesudrm.sql 容器ID:/root/

# 进入容器

docker exec -it tesudrm\_mysql /bin/bash # 容器开启的状态下

cd /root/

# 创建数据库/导入数据

mysql -uroot -p

create dababase example charset=utf8;

mysql -uroot -p example < tesudrm.sql

# 退出容器

Ctrl+D

### 5.3 \*Redis容器

docker pull redis:3.2

cd /www/pro\_tesu/

# 创建容器:宿主机6379端口可能被宿主机占用

docker run -p 6378:6379 --name tesudrm\_redis -v $PWD/Redis/conf/redis.conf:/etc/redis/redis.conf \

-v $PWD/Redis/data:/data \

-d redis:3.2 redis-server /etc/redis/redis.conf

# 附件的配置文件需要修改：

# 注释掉127.0.0.1

# 找到requirepass字段，取消注释，字段后填写redis密码

### 5.4 搭建项目python环境镜像并上传阿里云

#### 5.4.1 \*拉取centOS6.8镜像，并创建容器

docker pull centos:6.8

docker run -itd centos:6.8 /bin/bash

#### 5.4.2 编译的的方式安装python3.5.7

1.官网下载python3.5.7(手动迅雷下载快)

2.安装软件管理包和可能使用的依赖

yum -y groupinstall "Development tools"

yum install openssl-devel bzip2-devel expat-devel gdbm-devel readline-devel sqlite-devel gcc

3.安装python3

# 解压

tar -zxvf Python-3.5.7.tgz

cd Python-3.5.7

# 编译

./configure --prefix=/usr/local/python3

make

make install

# 建立软链接

ln -s /usr/local/python3/bin/python3.5 /usr/bin/python3

ln -s /usr/local/python3/bin/pip3.5 /usr/bin/pip3

# 查看python3与pip3版本

python3

pip3 -V

# 上传requirements.txt依赖文件

docker cp /www/pro\_tesu/TSDRM/requirements.txt 容器ID:/root/

# 从清华大小pip源安装依赖

pip3 install -r /root/requirements.txt -i https://pypi.tuna.tsinghua.edu.cn/simple/

#### 5.4.3 安装pdf生成软件wkhtmltox

在https://github.com/wkhtmltopdf/wkhtmltopdf/releases/ 下载对应的安装包，在这里我选择：wkhtmltox-0.12.5-1.centos6. x86\_64.rpm

rpm -ivh wkhtmltox-0.12.5-1.centos6. x86\_64.rpm

报错：

error: Failed dependencies:

fontconfig is needed by wkhtmltox-1:0.12.5-1.centos6.x86\_64

freetype is needed by wkhtmltox-1:0.12.5-1.centos6.x86\_64

libX11 is needed by wkhtmltox-1:0.12.5-1.centos6.x86\_64

libXext is needed by wkhtmltox-1:0.12.5-1.centos6.x86\_64

libXrender is needed by wkhtmltox-1:0.12.5-1.centos6.x86\_64

libjpeg is needed by wkhtmltox-1:0.12.5-1.centos6.x86\_64

libpng is needed by wkhtmltox-1:0.12.5-1.centos6.x86\_64

xorg-x11-fonts-75dpi is needed by wkhtmltox-1:0.12.5-1.centos6.x86\_64

xorg-x11-fonts-Type1 is needed by wkhtmltox-1:0.12.5-1.centos6.x86\_64

解决：

# 通过yum安装

yum install -y fontconfig freetype libX11 libXext libXrender libjpeg libpng xorg-x11-fonts-75dpi xorg-x11-fonts-Type1

# 下载中文字体simsun.ttc复制到 linux系统 /usr/share/fonts

docker cp /root/Downloads/simsun.ttc 容器ID:/usr/share/fonts

测试使用：

wkhtmltopdf http://www.baidu.com ./test.pdf

#### 5.4.4 打包自定义镜像上传阿里云

# 先关闭容器

docker stop 容器ID

# 基于当前容器重新创建镜像

docker commit -m "在centos6.8下搭建python环境" -a "root" -p 容器ID tesudrm\_pro:1.0

# -p表示提交时停止容器

# 重新命名

docker tag 镜像ID registry.cn-hangzhou.aliyuncs.com/tesunet/ tesudrm\_pro:1.0

# 上传至阿里云

docker push registry.cn-hangzhou.aliyuncs.com/tesunet/ tesudrm\_pro:1.0

#### 5.4.5 \*下拉自定义镜像并创建项目Django容器

# 拉取镜像

docker pull registry.cn-hangzhou.aliyuncs.com/tesunet/ tesudrm\_pro:1.0

# 命名镜像

docker tag registry.cn-hangzhou.aliyuncs.com/tesunet/ tesudrm\_pro:1.0 tesudrm\_pro:1.0

# 运行容器

cd /www/pro\_tesu/

docker run -p 8000:8000 -v $PWD/TSDRM:/TSDRM \

--link tesudrm\_mysql:mysql\_server \

--link tesudrm\_redis:redis\_server \

--name tesudrm\_pro \

-itd tesudrm\_pro:1.0

#### 5.4.6 \*处理静态文件并配置uwsgi

1.collectstatic命令收集静态文件至项目目录下

python manage.py collectstatic

2.注释掉STATIC\_ROOT，添加

STATICFILES\_DIRS = [

os.path.join(BASE\_DIR, "static")

]

3.配置uwsgi.ini

[uwsgi]

socket=0.0.0.0:8000 # 配置nginx代理时使用

#http=0.0.0.0:8000

chdir=/TSDRM

#module=TSDRM.wsgi:application

wsgi-file=/TSDRM/TSDRM/wsgi.py

master=True

process=4

pidfile=/TSDRM/TSDRM-master.pid

vacuum=True

max-requests=5000

daemonize=/TSDRM/log/wsgi.log

static-map=/static=/TSDRM/static

4.uwsgi运行项目

cd /www/pro\_tesu/TSDRM

uwsgi --ini uwsgi.ini

### 5.5 \*Nginx容器

#### 5.5.1 下载镜像/安装容器

docker run -v $PWD/Nginx/nginx.conf:/etc/nginx/nginx.conf \

-v $PWD/Nginx/log:/var/log/nginx \

-v $PWD/TSDRM/static:/static \

--link tesudrm\_pro:django\_server \

--name tesudrm\_nginx \

-d -p 8888:80 nginx

### 5.6 访问项目

IP:8888

#### 5.7 Docker容器启动/重启脚本