

# Docker 安装使用

编写： 张帅男

邮箱： [zhangshuainan@tbkt.cn](mailto:zhangshuainan@tbkt.cn)

版本	修改人	修改时间	注释
V0.0	张帅男	2017-12-11	初稿

声明：以下说明，属作者个人理解，若出现不符想象，概不负责

# Docker 安装与使用

## (一) Docker 安装

### 1. docker 下载

toolbox 官方下载地址: <https://www.docker.com/toolbox>

或者在咱们公司的机器上有安装包

1) win + R

\\192.168.0.130 点击确定后选择 DockerToolbox-1.12.0

### 2. docker 安装

**windows 下 docker-machine 创建 docker 环境不成功的一般解决方法:**

测试发现 win7 系统不支持最新的 virtualbox, 所以在 win7 下 安装 virtualbox 4.3.12

也可以选择升级系统到 win10, 可以使用最新的 virtualbox 版本

### 3. docker 关联公司内部镜像仓库

**virtual-machine** 名称可以自定义, 作用是用来定义 一个 **docker** 环境

一台电脑上可以有多个 docker 环境, 用 docker-machine ls 命令  
可以查看, 每个 docker 环境可以运行多个 container

### 4. 创建 docker 机器 dev

```
docker-machine.exe create --driver virtualbox  
--engine-insecure-registry 122.114.40.73:5001 dev
```

来创建一个默认镜像仓库为公司地址的 docker 环境 (dev)

### 5. docker 操作注意事项

windows 下用 `cmd` 或者 `git bash` 进入命令行窗口进行 docker 或 git 的操作。docker 进入的是 docker shell 的命令窗口

进入 docker shell 方式:

在 `cmd` 或 `git bash` 命令行下 输入 `docker-machine ssh dev` 进入 `docker shell`

建议与 docker 环境交互的命令全在 `docker shell` 中执行

# Docker 安装与使用

## (二) Docker 使用

### 1. 打开 git bash

### 2. 查看 docker 机器的状态：可查看机器的 ip，可以创建多个机器 `docker-machine.exe ls`

### 3. 启动并进入所选择的 docker 机器

`docker-machine.exe start dev`

`docker-machine.exe ssh dev`

### 4. 依据上面创建 docker 机器的命令已经拉取过镜像了。 在机器里面创建镜像

#### A. 远程拉取

`docker pull 122.114.40.73:5001/django:1.8.7`

B. 本地上传镜像（需要在设置共享目录之后）

#### 2) win + R

\\192.168.0.130 点击确定后选择要使用的镜像进行复制，  
复制到 docker 的共享目录下 Work

#### 3) 进入容器 `docker-machine.exe ssh dev`

进入容器中的共享目录，ls，找到你粘贴的镜像，

`cat django1_2_3.tar.gz | docker load`

#### C. 查看该机器所有的镜像

`docker images [options "o">] [name]`

`docker search 镜像名字`

### 5. 设置共享目录

#### A. 建立文件夹，作为共享目录的文件夹

`sudo mkdir -p /mnt/code`

#### B. 设置 VirtualBox 的共享文件夹

打开桌面上打开 VirtualBox，选择对应的机器，点击设置->  
选择共享文件夹，设置共享文件夹的路径以及名称，选择固定分配，确定保存

#### C. 设置自动挂载

进入对应的机器(ssh)，打开以下文件

`sudo vi /var/lib/boot2docker/profile`

最后一行增加：（work 为 B 中共享文件夹的名称）

`sudo mkdir -p /mnt/code && sudo mount -t vboxsf Work /mnt/code`

#### D. 退出重启

`docker-machine.exe restart dev`

## Docker 安装与使用

### E. 进入镜像查看共享目录

```
docker run -it -v /mnt/code/:/code
122.114.40.73:5001/django:1.8.7 /bin/bash
cd /code
```

### 6. 创建对应项目的容器， 并启用别名（django 镜像为例子）:

#### A.

```
docker create --name 容器名 -it --net=host -v 共享目
录: 挂载路径 122.114.40.73:5001/django:1.8.7 项目启
动命令
```

例:

```
docker create --name media -it --net=host -v
/mnt/code/:/code 122.114.40.73:5001/django:1.8.7 python
/code/GitProject/tbkt_media/manage.py runserver
0.0.0.0:808
```

通过上面启动的容器， 只能通过 docker 机器的 ip 对应上面的端口进行访问。 可以写个 sh 脚本进行映射， 映射到本地的指定端口

例:

```
docker-machine ssh dev -L :808:localhost:8009
```

#### B. 直接端口映射创建容器

```
docker run -d -v /Users/yy_stu_web_dj/tbkt:/code -p
8000:8099 --entrypoint=python
hub.tbkt.cn:5001/django:1.8.7 /code/manage.py runserver
0.0.0.0:8000
```

查找对应的进程（容器）， 并设置名字

```
docker ps -a
```

```
docker rename 进程 id 容器名字
```

### 7. 启动容器（项目）

```
docker start 容器名字
```

```
docker restart 容器名字 重启容器
```

### 8. 根据名字查看项目运行的日志

```
docker logs -f --tail=50 容器名称
```

### 9. 通过容器名进入容器

```
docker exec -it 容器名字 bash
```

或

```
docker exec -it 容器名字 /bin/sh
```

### 10. 删除容器

## Docker 安装与使用

查看机器里面创建的所有容器

```
docker ps -a
```

```
docker rm -f 容器名
```

```
docker rm -f 容器进程号
```