# Docker使用说明

版本 v0.2 - 初始版

版本 v0.3 - 加入MySQL镜像创建、应用服务器与MySQL服务器连接

## 为什么使用Docker

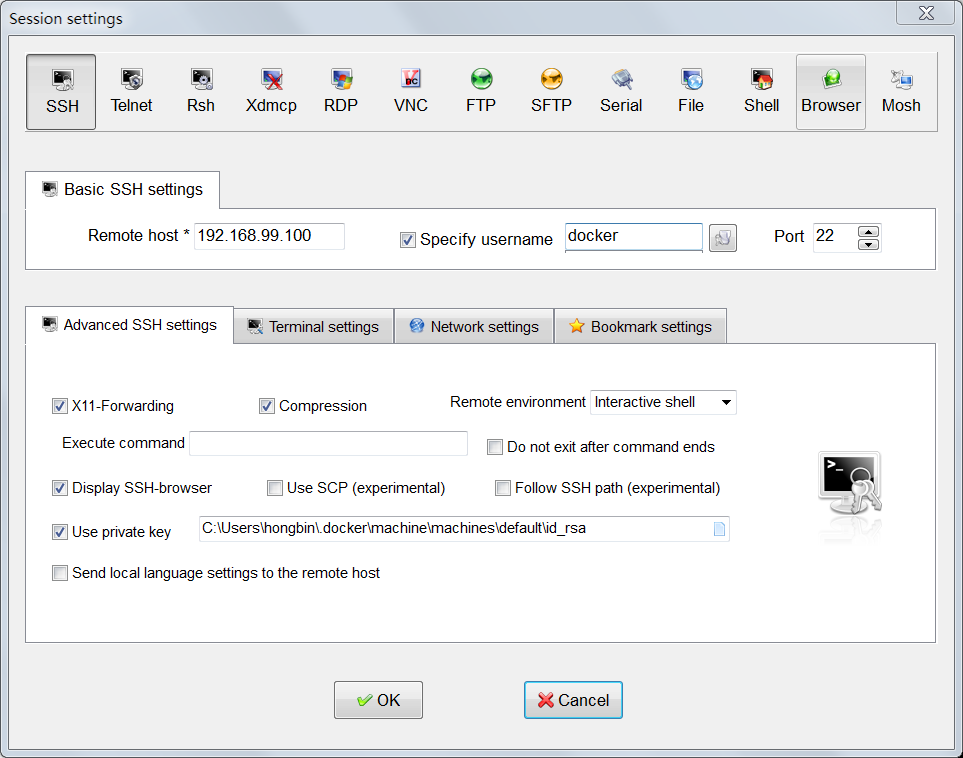
略

## Docker搭建Linux/Python开发环境（Windows）

### 基础开发环境搭建

下面介绍在Windows电脑上搭建Linux Python Web开发环境的过程

1. 下载安装Git for Windows：<http://www.git-scm.com/>
2. 下载安装Docker Toolbox：<http://get.daocloud.io/> （由于docker.io官方网站速度太慢，建议使用道客的下载链接）
3. 下载安装Mobaxterm或PuTTY（建议前者）: <http://mobaxterm.mobatek.net/>
4. Docker Toolbox安装完成之后，可以使用传统的CMD，PowerShell或者SSH客户端连接，推荐使用Mobaxterm连接，连接配置如图所示：



1. 在MobaXterm中敲入命令：sudo docker run –i –t ubuntu /bin/bash即可下载Ubuntu镜像并启动Ubuntu Linux的终端，其中默认安装了Python 3.4.3，敲入命令python3可知
2. 在<https://hub.docker.com/>上注册一个账户，例如freelemon
3. 登陆 sudo docker login
4. 执行下面的步骤（配置Django）可以加入Dockerfile，以便后续使用
   1. apt-get update
   2. apt-get install curl wget
   3. cd
   4. mkdir dmp
   5. cd dmp
   6. wget <https://bootstrap.pypa.io/get-pip.py>
   7. python3 get-pip.py
   8. /root/dmp/projects/tanzhen/mysite
   9. virtualenv --python=/usr/bin/python3 env
   10. source env/bin/activate
   11. pip install django

说明：如果服务器上只有一个版本，可以省略virtualenv的使用，减少复杂性。

1. 提交本地修改 sudo docker commit <image-id> freelemon/django
2. 使用命令sudo docker ps –l –q查看<image-id>
3. 使用命令sudo docker push freelemon/django推送到dockerhub，此处因为网络原因，很难成功，推送不成功但是本地可以commit保存。这种保持镜像的方法并不推荐，推荐使用Dockerfile的方式。
4. 应用服务器需要连接数据库，此处使用MySQL，但是MySQL-python模块还不支持Python 3(如果使用Python2.7则可以直接安装MySQL-python模块)，所以只能使用MySQL的官方驱动，下载MySQL Python Connector之后并不能使用，使用下述方法解决（见参考链接4和5）：
   1. pip search mysql-connector
   2. pip install mysql-connector-python
   3. pip install mysql-connector-python --allow-external mysql-connector-python
   4. 测试：在python窗口中敲入命令import mysql，没有错误即可。
   5. 安装MySQL client用来测试：apt-get install mysql-client

### 示例docker image

我目前维护的Django镜像地址：<https://hub.docker.com/r/freelemon/django/>

可以直接下载使用

### Django Dockerfile的内容和使用说明

@TODO

## Docker搭建MySQL开发环境（Windows）

完全使用参考链接7文中的内容。

*注意：不要改成centos7，因为centos7使用的是MariaDB而不是MySQL，也不要使用CentOS 6.5，因为docker hub上面没有centos6.5*

Dockerfile内容

FROM centos:centos6

MAINTAINER Fanbin Kong "kongxx@hotmail.com"

RUN yum install -y mysql-server mysql

RUN /etc/init.d/mysqld start &&\

mysql -e "grant all privileges on \*.\* to 'root'@'%' identified by 'letmein';"&&\

mysql -e "grant all privileges on \*.\* to 'root'@'localhost' identified by 'letmein';"&&\

mysql -u root -pletmein -e "show databases;"

EXPOSE 3306

CMD ["/usr/bin/mysqld\_safe"]

命令如下：

* 生成镜像：sudo docker build -t mysql\_server .
* 启动容器：sudo docker run --name=mysqlserver -d -P mysql\_server
* 使用命令检查安装是否成功：mysql -h <宿主机器> -u root -pletmein -P 49153

## Docker链接Web和数据库服务

主要使用**link**命令进行连接，主要的好处是：“Docker通过在 两个container建立了一个安全通道, 这样container就不用对外暴露端口了. 你可能已经注意到了我们在启动 db container的时候没有使用 -p 或者 -P。 因为我们已经吧两个container通过 link连接起来了， 所以没必要通过端口暴露数据库的服务了.”

参见链接8。

命令如下：

1. 重命名数据库服务器：sudo docker run -d --name db mysql\_server
2. 重命名freelemon/django为web(作为mysql的client),建立web和db的链接并且打开终端：sudo docker run --name=web --link=db:db -t -i freelemon/django /bin/bash
3. 使用命令env列出环境

# env | grep DB

DB\_NAME=/web/db

DB\_PORT=tcp://172.17.0.4:3306

DB\_PORT\_3306\_TCP\_PORT=3306

DB\_PORT\_3306\_TCP\_PROTO=tcp

DB\_PORT\_3306\_TCP\_ADDR=172.17.0.4

DB\_PORT\_3306\_TCP=tcp://172.17.0.4:3306

1. 启动MySQL客户端验证连接：mysql -h db -u root -pletmein
2. 熟悉一下docker start/stop, docker exec, docker ps, docker rm等常用命令来控制容器

## Docker在阿里云ECS上的使用sudo docker build -t mysql\_server .

@TODO

## Docker在内部服务器上的使用

@TODO

## 参考链接

1. Django开发指南：<http://masteringdjango.com/django-models/>
2. 开箱即用的简单Django: <http://www.pedaldrivenprogramming.com/2015/10/sane-django-development-with-docker/>
3. Django生产部署镜像参考：<https://realpython.com/blog/python/django-development-with-docker-compose-and-machine/>
4. 安装MySQL的Python驱动：<http://stackoverflow.com/questions/22100757/can-not-get-mysql-connector-python-to-install-in-virtualenv>
5. MySQL-Python不支持Python3: <http://smilejay.com/2013/03/python3-mysql-connector/>
6. MySQL Python Connector指南：<http://dev.mysql.com/doc/connector-python/en/connector-python-example-connecting.html>
7. Docker创建MySQL容器 ：<http://blog.csdn.net/kongxx/article/details/38579723>
8. Docker链接容器：<http://www.oschina.net/translate/dockerlinks>
9. 第一本Docker书：<http://item.jd.com/11600997.html>