

# Ron's Blog

## 在Windows上玩TensorFlow(一)——安装Docker

📅 2016-05-09 | 📁 Docker | 👁 2300

欢迎转载，转载请在醒目处注明出处，Ron's Blog: <http://www.neilron.xyz/set-up-docker-on-windows/>

“谷歌”+“深度学习”，两个标签让2015年12月才由谷歌开源的深度学习工具TensorFlow在其发布之后就迅速地成为了全球最为炙手可热的开源项目，2016年4月，开源的TensorFlow又支持了分布式特性，向着生产环境下的应用更进一步。

TensorFlow API支持Python 2.7和Python 3.3+，共支持4种安装方式。

- Pip install
- Virtualenv install
- Anaconda install
- Docker install

其中大部分支持Linux和Mac OS，由于主要开发环境是Windows，我选择了最为灵活的docker方式安装TensorFlow。TensorFlow还有GPU支持版本，本文仅探索CPU-Only版本。

我计划完成一系列3篇博文，第一步安装docker，第二步单机版TensorFlow的Demo，第三步分布式TensorFlow的Demo，争取在1个月内完成。

### 1 Docker是什么？

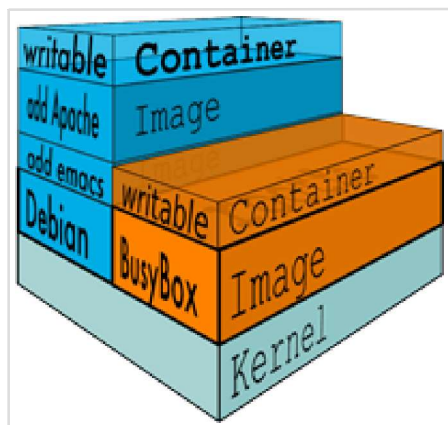
借用Docker官网最大的一行字。

Docker allows you to package an application with all of its dependencies into a standardized container for software development.

从功能上讲，Docker也可以理解为一种虚拟化的方案，可以通过构建包含不同软件的镜像，来

速部署开发环境的目的。

再借用官网的一张图，左边蓝色的部分从kernel开始一层层加了debian, emacs, apache形成了一个Image，每一层都是只读的，我们运行这个Image的时候，上面盖上了一层可读写的Container，让我们做一些编辑和修改，一个简单的服务器就可以用了；又如右边橘红色的部分，Kernel的上面加上了BusyBox就形成了Image，运行起来之后就可以以非常轻量级的方式运行起busybox中支持的命令。再有我们接下来要学习的TensorFlow镜像，就包含了运行它所需要的全部依赖，简单操作就可以完成TensorFlow开发环境的搭建。



更多的Docker理解推荐阅读[10张图带你深入理解Docker容器和镜像](#)

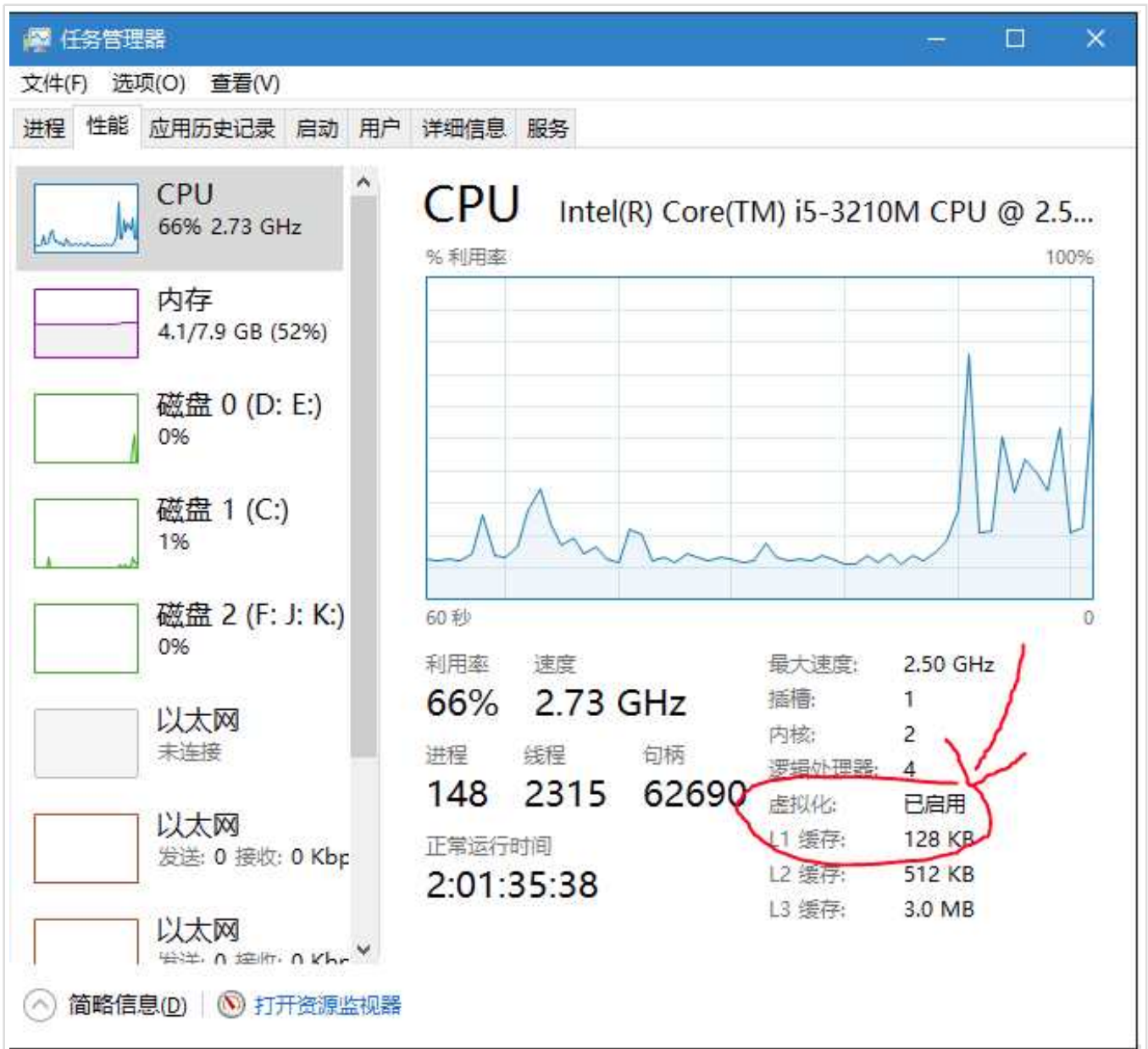
## 2 安装Docker

要在Windows上运行Docker，首先需要下载和安装[Docker Toolbox](#)。顺便奉上[Docker Windows文档](#)，更喜欢官方文档的话可以看[这里](#)，也可以按照本文的步骤继续。

### 2.1 确认系统版本

首先确认自己的系统是WIN 7或更新的64位系统，且需要支持硬件虚拟化技术。

Win 8及以上查看方法：



## Win 7查看方法

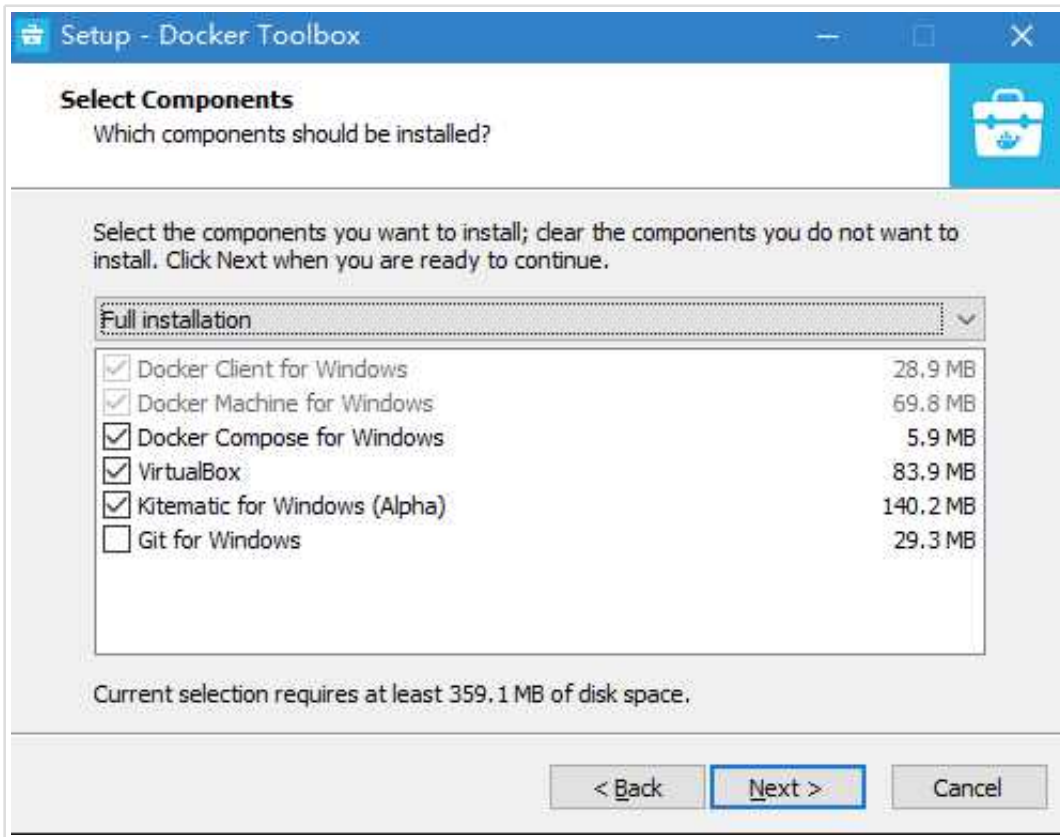
Win 7 运行 Microsoft® Hardware-Assisted Virtualization Detection Tool 这一工具执行检查。

完成之后就可以继续下一步了。

## 2.2 安装Docker Toolbox

点击 Docker Toolbox 下载，在本文写作时的最新版本为1.11.1。

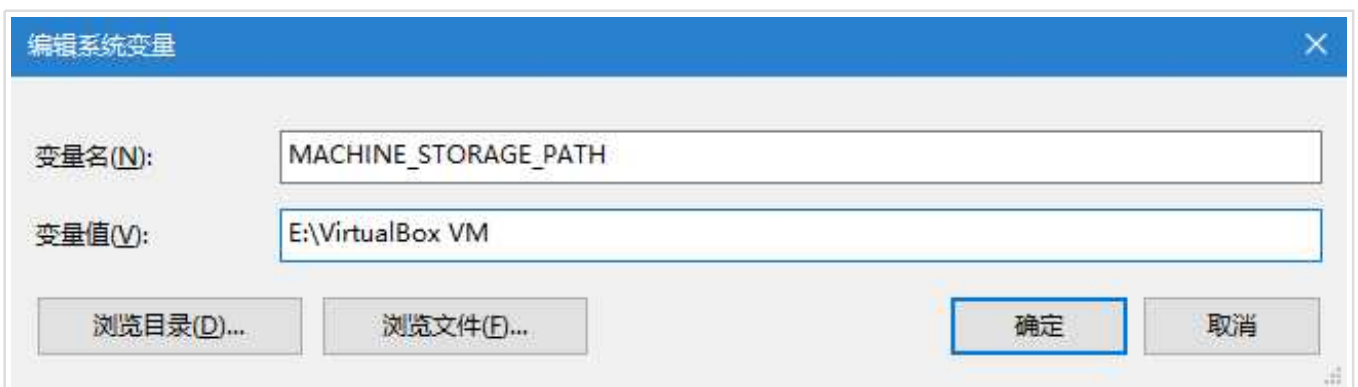
安装过程会安装Docker的各个组件和Oracle VirtualBox，因为Docker需要依赖Linux内核的一些特性，因此Mac和Windows都需要在机器上运行一个小型的Linux系统作为Host系统。如果已经安装过VirtualBox的最新版本，则无需重新安装。



自行选择安装路径，如果已经安装过Git也可以去掉该工具的勾选，VirtualBox也是一样，其它可一路Next。

## 2.3 配置Docker

安装完成后，**建议先配置一个环境变量MACHINE\_STORAGE\_PATH**，来自定义虚拟机保存的位置，因为之后下载的镜像越来越多，都是放在虚拟机的虚拟磁盘文件中，虚拟磁盘文件会越来越大，放在默认的C盘用户目录下可能会在以后造成一些麻烦，参考下图，设置为任意你喜欢的位置。



设置完成之后，WIN+R键输入cmd，打开一个windows命令行窗口。

输入下面的命令新建一个虚拟机iron-docker，使用virtualbox作为driver，这个虚拟机会自动创的MACHINE\_STORAGE\_PATH配置的目录中，安装过程会从github上拉取boot2docker.iso，可

较慢，可以手动下载，也可以直接将toolbox安装目录下的boot2docker.iso复制到%MACHINE\_STORAGE\_PATH%/cache中。

```
1 docker-machine create ron-docker -d virtualbox
```

还要注意过程中可能会出现一些要求Windows执行权限的窗口，有些会最小化在任务栏上，请——赋予权限。

建立完成之后键入 `docker-machine ls` 查看刚刚新建的虚拟机，下面的命令都会以**ron-docker**为例，读者请按照自己的配置修改。

```
1 C:\Users\ron>docker-machine ls
2 NAME          ACTIVE  DRIVER      STATE    URL                     SWARM   DOC
3 ron-docker    -       virtualbox   Running  tcp://192.168.99.100:2376           v1.
```

这时候我们使用的CMD还不能与Docker Engine建立连接，可以通过 `docker-machine env mydocker` 查看如何设置。

```
1 C:\Users\ron>docker-machine env ron-docker
2 SET DOCKER_TLS_VERIFY=1
3 SET DOCKER_HOST=tcp://192.168.99.100:2376
4 SET DOCKER_CERT_PATH=E:\VirtualBox VM\machines\ron-docker
5 SET DOCKER_MACHINE_NAME=ron-docker
6 REM Run this command to configure your shell:
7 REM     @FOR /f "tokens=*" %i IN ('docker-machine env ron-docker') DO @%i
```

键入下面的命令完成环境变量配置


```
1 FOR /F "tokens=*" %i IN ('docker-machine env ron-docker') DO %i
```

### 3 运行镜像

下面我们下载最新的TensorFlow镜像来体验一下在docker下运行镜像。

### 3.1 下载镜像

```
1  docker pull gcr.io/tensorflow/tensorflow
2
3  docker images
4  REPOSITORY                                TAG          IMAGE ID          CREATED
5  gcr.io/tensorflow/tensorflow               latest       ae5ff5a9860a3     2 weeks ago
```



当然有可能有网络问题，请根据自己的具体状况自行解决，挂代理，VPN，手动下载都可以。

### 3.2 启动镜像和探索

使用下面的命令启动镜像。

```
1  docker run -it gcr.io/tensorflow/tensorflow bash
```

选项-i用于保持STDIN在当前的窗口上，-t用于分配一个pseudo-tty，两个选项使得当前的窗口可以像一个linux的bash一样运行，丝毫没有违和感。第一个参数指定了使用的镜像，第二个参数指定了启动这个镜像后启用的命令，用bash启动方便我们开始第一次的探索。

- o uname -a，查看信息
- o cd到/目录下，查看目录结构
- o 运行tensorflow的hello world

```
1  root@ec9bfd276082:/# uname -a
2  Linux ec9bfd276082 4.4.8-boot2docker #1 SMP Mon Apr 25 21:57:27 UTC 2016 x86_64 x86_64
3  root@ec9bfd276082:/# ls /
4  bin  dev  home  lib64  mnt      opt    root  run_jupyter.sh  srv  tmp  var
5  boot  etc  lib   media  notebooks  proc   run   sbin            sys  usr
6  root@ec9bfd276082:/# python
7  Python 2.7.6 (default, Jun 22 2015, 17:58:13)
8  [GCC 4.8.2] on linux2
9  Type "help", "copyright", "credits" or "license" for more information.
10 >>> import tensorflow as tf
11 >>> hello = tf.constant('Hello, TensorFlow!')
12 >>> sess =tf.Session()
13 >>> sess.run(hello)
14 'Hello, TensorFlow!'
```

```
15 >>> a = tf.constant(10)
16 >>> b = tf.constant(32)
17 >>> sess.run(a+b)
18 42
19 >>>
```

注意到/目录下的run\_jupyter.sh，这事实上是当前版本tensorflow启动的默认命令，也就是说，如果我们在启动镜像时没有指定bash，就会默认运行这个脚本，这与一些稍早一些版本的tensorflow不同，许多教程中也还没有提到，可能会造成困惑，读者可以尝试一下 `docker run -it gcr.io/tensorflow/tensorflow`，它会启动一个notebook的服务，运行在本地的8888端口上，但这样就想从windows的浏览器上打开notebook是不行的，这与docker本身的机制和运行在虚拟上两个原因有关，如果读者现在就想看到notebook，点击[这里](#)立即带你去到完成更多必要配置的端口转发部分。

## 4 配置自己的快速启动窗口

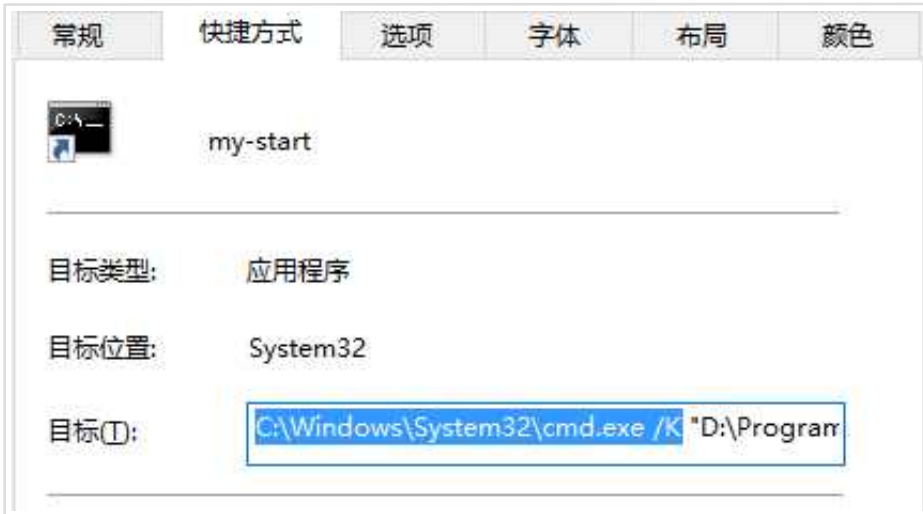
回顾一下上面的过程，要将一个Windows的CMD窗口变为一个运行特定容器的窗口，需要3步。

1. 启动虚拟机
2. 配置虚拟机对应的环境变量
3. 运行镜像

下面我自己用的一个启动脚本，需要的话仅需修改相应的machine-name，然后将脚本保存为my-start.bat，放在toolbox安装目录下。之后右键my-start.bat->发送到桌面快捷方式，按代码下的图配置该快捷方式，/K参数可以使bat运行完之后不自动关闭。这样，需要时双击运行就可以获得一个新的可运行镜像的CMD窗口。

```
1 @ECHO off
2
3 SET machine-name=ron-docker
4
5 ECHO Init...
6
7 FOR /F %i IN ('docker-machine status %machine-name%') DO SET status=%i
8
9 IF %status%==Running (
10     ECHO %machine-name% is running.
11 ) ELSE (
12     ECHO %machine-name% is shutdown.
13     docker-machine start %machine-name%
```

```
14 )  
15  
16 ECHO Configuring Environment...  
17 FOR /F "tokens=*" %%i IN ('docker-machine env %machine-name%') DO %%i  
18  
19 ECHO Init Finished.
```



## 5 完成更多必要配置

上面我们完成了镜像从下载到启动的过程，但在将docker用作开发环境之前，还有一些很有必要的配置工作要做，让我们一起来完成下面3步。

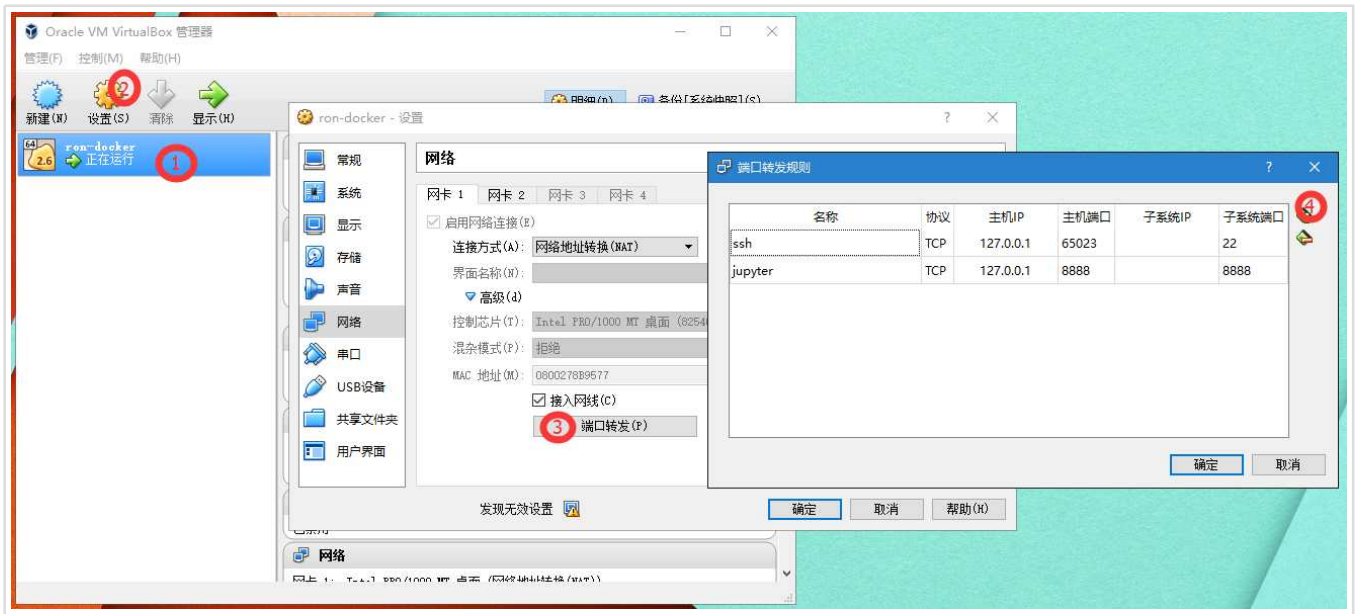
### 5.1 端口转发

我们的容器运行在一个小的linux虚拟机上，而虚拟机又运行在Windows系统的VirtualBox上，自然的，运行在容器上的服务不能在Windows上打开浏览器直接访问。端口转发就是要解决这个问题，共有两步：

#### 建立Windows和虚拟机之间的端口转发

可以通过VirtualBox的管理界面完成，如图，然后重启虚拟机。





## 配置虚拟机和容器之间的端口转发

使用-p选项

```
1 docker run -it -p 8888:8888 gcr.io/tensorflow/tensorflow
```

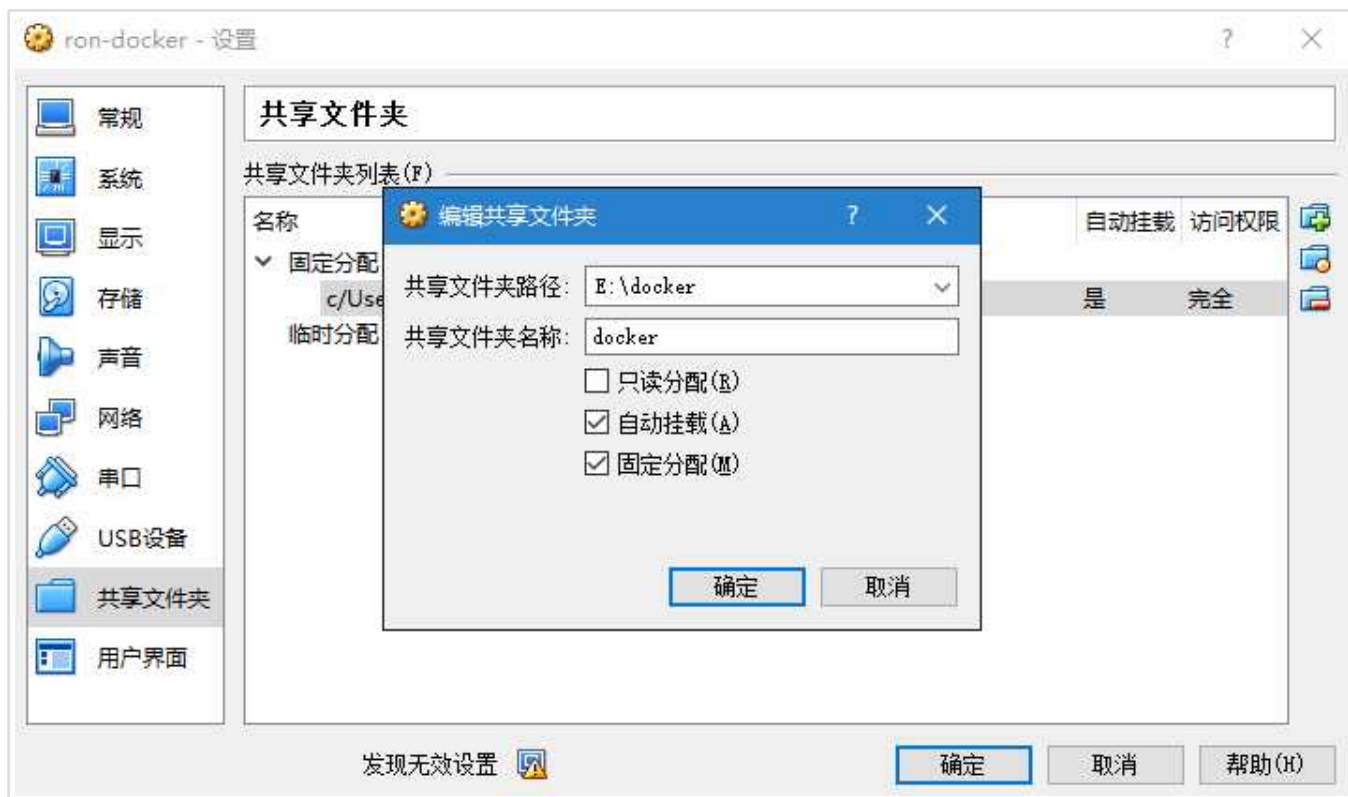
键入上面这条命令之后打开浏览器，访问localhost:8888，看到运行在容器中的notebook服务了吧

## 5.2 磁盘映射

磁盘映射是非常有用的一步配置，可以将Windows上的磁盘直接映射到容器中，这样在Windows上开发代码，直接在容器中运行，避免了写完复制的大麻烦。磁盘映射的处理方法与端口转发类似，磁盘的映射关系需要完成Windows到虚拟机，虚拟机到容器的两步配置，请跟随下面3个步骤：

### 建立Windows和虚拟机之间的磁盘映射

打开虚拟机的设置页面后，在共享文件夹中设置你想要共享的文件夹和它的名称，如图，重启虚拟机。



## 在虚拟机中挂载磁盘

这一步要用到刚才设置的名称，我这里就是docker。

```
1 mkdir -p /home/docker/data
2 mount -t vboxsf -o uid=1000,gid=50 docker /home/docker/data
```

到这里，我们在Windows中所做的修改就可以实时地反映到虚拟机中了。

## 建立虚拟机和容器之间的磁盘映射

使用-v选项，建立两个目录的映射关系

```
1 docker run -it -v /home/docker/data:/data gcr.io/tensorflow/tensorflow bash
```

好了，试试在Windows的共享文件夹中添加一个python的hello world，然后在docker中 `python /data/hello.py` 试试吧

## 5.3 配置启动脚本

我们上面用docker-machine创建的虚拟机，它的大部分目录在重启之后都会复原，除了/mnt/sc

个目录也就是虚拟机的虚拟磁盘文件disk.vmdk所挂载的位置，我们可以修改里面的/mnt/sda1/var/lib/boot2docker/profile文件，在文件最后添加我们自定义的启动命令，比如说添加前面的磁盘映射，就是在profile文件最后加上下面这两句。

```
1  mkdir -p /home/docker/data
2  mount -t vboxsf -o uid=1000,gid=50 docker /home/docker/data
```

这样每次虚拟机启动就会完成磁盘的挂载了。

## 6 结语

到这里，使用tensorflow之前安装docker的工作我尽己所能最详细地描述了，读者您如果能看到这里，我真的感到非常荣幸，第一次写这么长的教程，希望能带来一些帮助，最后，感谢下面参考资料的贡献者们，请期待tensorflow三部曲第二篇吧~

## 7 参考资料

1. 利用Docker构建开发环境：<http://tech.uc.cn/?p=2726>
2. 10张图带你深入理解Docker容器和镜像：<http://dockone.io/article/783>
3. How to install and run TensorFlow on a Windows PC: <http://www.netinstructions.com/how-to-install-and-run-tensorflow-on-a-windows-pc/>
4. Install Docker for Windows: <https://docs.docker.com/windows/>
5. 5 Useful Docker Tips and Tricks on Windows : <http://blog.pavelsklenar.com/5-useful-docker-tip-and-tricks-on-windows/>

#linux   #windows   #tensorflow   #google   #deep learning   #深度学习   #docker  
#virtualbox

---

◀ Spark中的Join操作

停止更新的tensorflow系列 ▶

快来点击分享吧   微博   微信   QQ空间   豆瓣   QQ   Google   Facebook  
Twitter   LinkedIn

## 1 条评论



赵越 Windows 7 Chrome 45.0.2454.101

谢谢提供这么详细的教程，我是搞期货 黄金 股票领域的 tensorflow 人工智能交易分析数据的。  
我微信号：ac4134134,希望有机会合作。

6月9日 回复 顶 转发

社交帐号登录:

微信

微博

QQ

人人

更多»



说点什么吧...

发布

Ron's Blog正在使用多说

© 2016 ♥ Neil Ron

由 [Hexo](#) 强力驱动 | 主题 - [NexT.Mist](#)

👁 3755