

第二部分

TensorFlow 初接触



扫描二维码

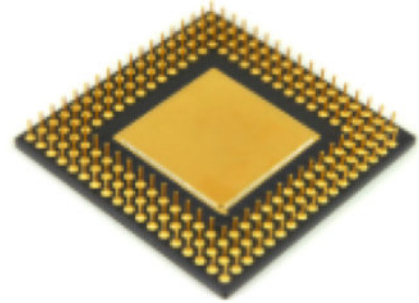
试看/购买《TensorFlow 快速入门与实战》视频课程

第二部分 目录

- 搭建你的 TensorFlow 开发环境
- “Hello TensorFlow”
- 在交互式环境中使用 TensorFlow
- 在容器中使用 TensorFlow

搭建你的 TensorFlow 开发环境

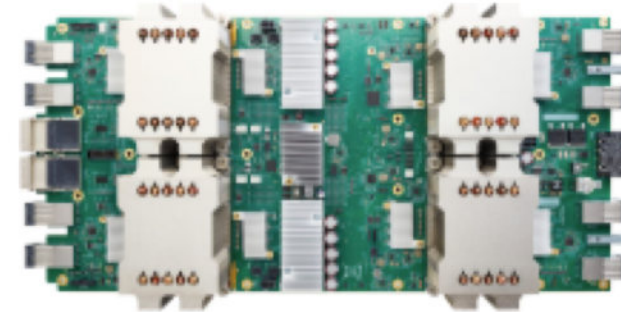
TensorFlow 支持的硬件平台



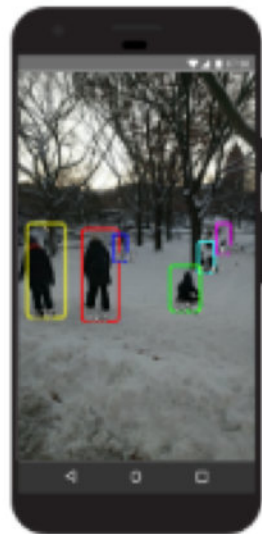
CPU



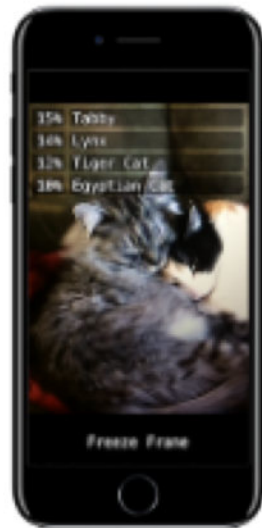
GPU



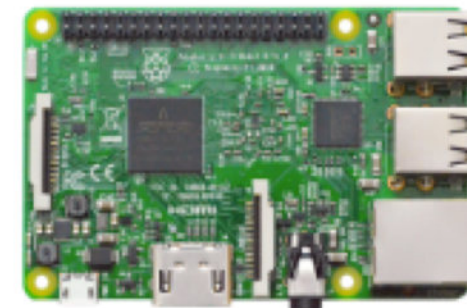
Cloud TPU



Android



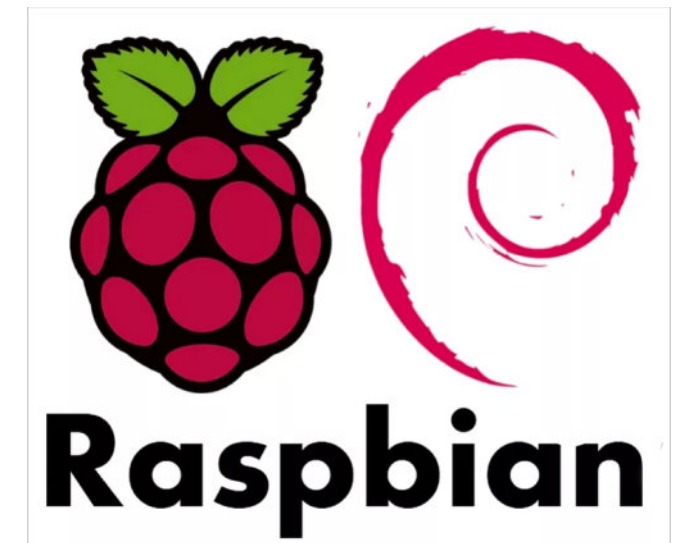
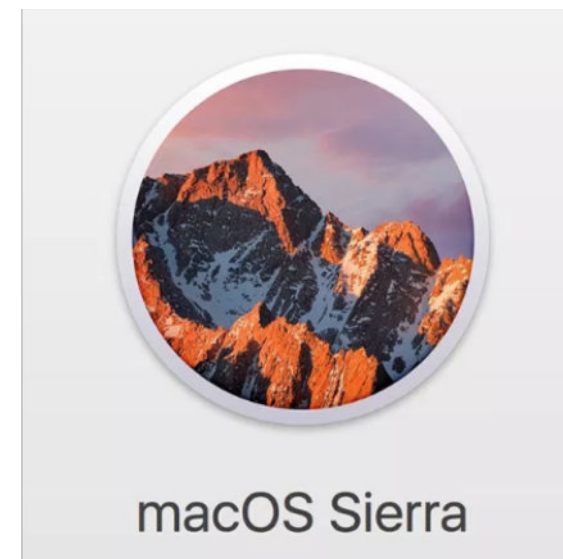
iOS



Embedded
Devices

TensorFlow 支持的操作系统

- Ubuntu 16.04 or later
- Windows 7 or later
- macOS 10.12.6 (Sierra) or later (no GPU support)
- Raspbian 9.0 or later



使用 pip 安装 TensorFlow

tensorflow —Current release for CPU-only (*recommended for beginners*)

tensorflow-gpu —Current release with GPU support (*Ubuntu and Windows*)

tf-nightly —Nightly build for CPU-only (*unstable*)

tf-nightly-gpu —Nightly build with GPU support (*unstable, Ubuntu and Windows*)

1. 安装 Python 开发环境

```
$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
$ export PATH="/usr/local/bin:/usr/local/sbin:$PATH"
$ brew update
$ brew install python@2 # Python 2
$ curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py # get get-pip.py
$ python get-pip.py # install pip
$ sudo pip install -U virtualenv # system-wide install
```

2. 创建 Python 虚拟环境 (*)

```
$ virtualenv --system-site-packages -p python2.7 ./venv
$ source ./venv/bin/activate # activate virtual env
(venv) $ pip list # show packages installed within the virtual environment
```

3. 安装适配的 TensorFlow 软件包

```
(venv) $ pip install --upgrade tensorflow
(venv) $ python -c "import tensorflow as tf"
```

Try it

“Hello TensorFlow”

```
import tensorflow as tf
# 定义常量操作 hello
hello = tf.constant("Hello TensorFlow")
# 创建一个会话
sess = tf.Session()
# 执行常量操作 hello 并打印到标准输出
print(sess.run(hello))
```

Output:

2018-12-19 02:00:58.943154: I

tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports

instructions that this TensorFlow binary was not compiled to use: **AVX2 FMA**

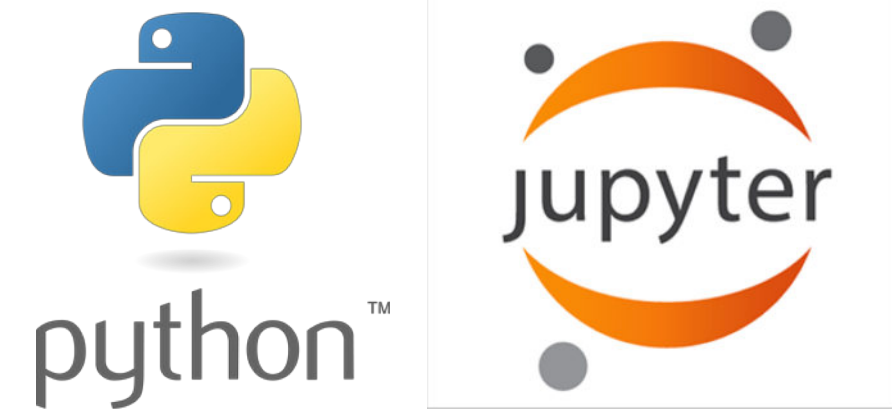
Hello TensorFlow

支持 AVX2 指令集的 CPUs

- Intel
 - Haswell processor, Q2 2013
 - Haswell E processor, Q3 2014
 - Broadwell processor, Q4 2014
 - Broadwell E processor, Q3 2016
 - Skylake processor, Q3 2015
 - Kaby Lake processor, Q3 2016(ULV mobile)/Q1 2017(desktop/mobile)
 - Skylake-X processor, Q2 2017
 - Coffee Lake processor, Q4 2017
 - Cannon Lake processor, expected in 2018
 - Cascade Lake processor, expected in 2018
 - Ice Lake processor, expected in 2018
- AMD
 - Excavator processor and newer, Q2 2015
 - Zen processor, Q1 2017
 - Zen+ processor, Q2 2018

在交互式环境中使用 TensorFlow

Jupyter Notebook 交互式开发环境



```
(venv) $ pip install jupyter
```

```
(venv) $ python -m ipykernel install --user --name=venv
```

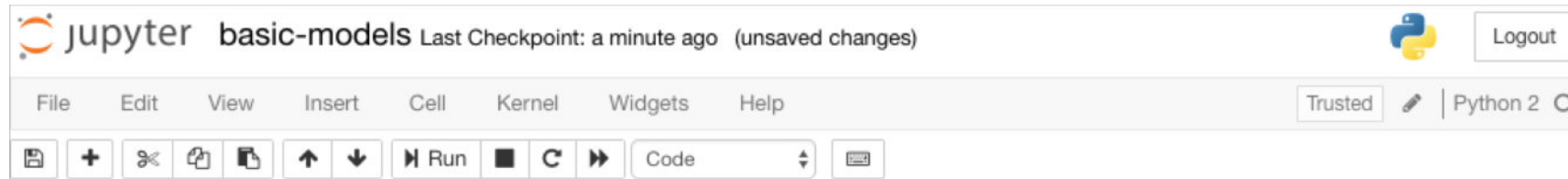
 Quit Logout

Files Running Clusters

Select items to perform actions on them. Upload New ▾ ↻

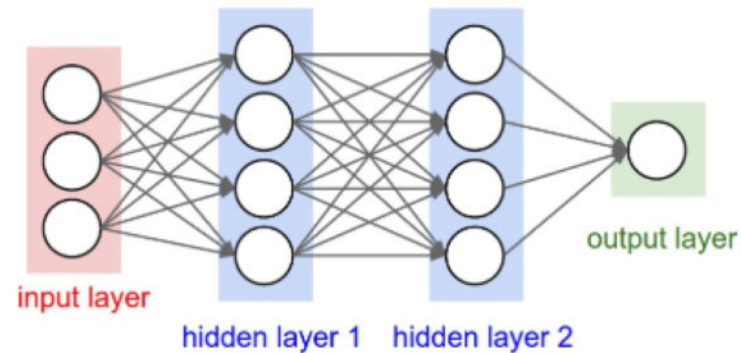
<input type="checkbox"/> 0 ▾	📁 / notebook-examples	Name ▾	Last Modified	File size
	📁 ..		seconds ago	
<input type="checkbox"/>	📄 array-ops.ipynb		8 hours ago	1.3 kB
<input type="checkbox"/>	📄 basic-models.ipynb		seconds ago	6.8 kB
<input type="checkbox"/>	📄 basic-ops.ipynb		7 minutes ago	4.94 kB

在 Jupyter Notebook 中使用 TensorFlow



多层感知机模型示例

Neural Network Overview



MNIST Dataset Overview

MNIST图像数据集使用形如 $[28, 28]$ 的二阶数组来表示每张图像，数组中的每个元素对应一个像素点。该数据集中的图像都是256阶灰度图，像素值0表示白色（背景），255表示黑色（前景）。由于每张图像的尺寸都是28x28像素，为了方便连续存储，我们可以将形如 $[28, 28]$ 的二阶数组“摊平”成形如 $[784]$ 的一阶数组。数组中的784个元素共同组成了一个784维的向量。



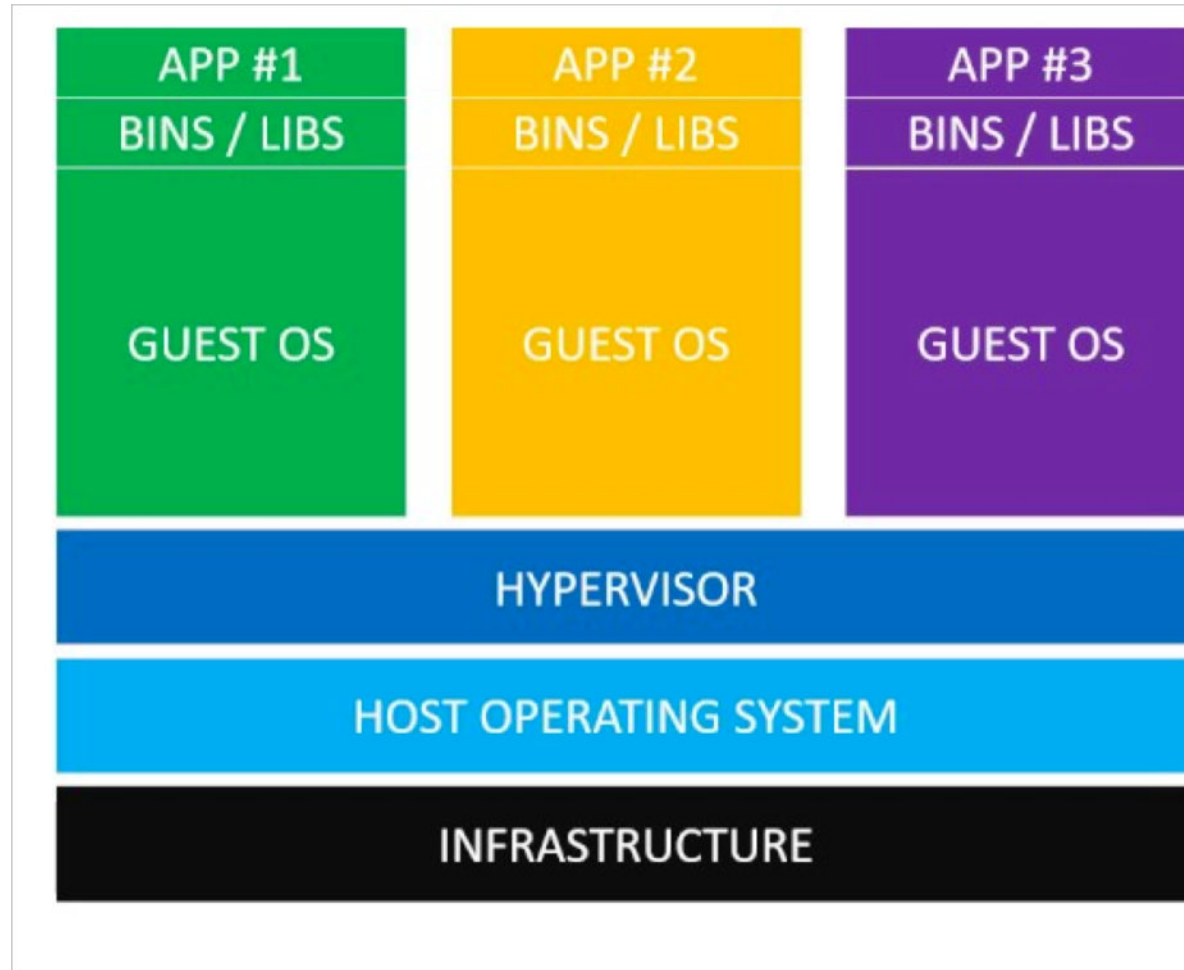
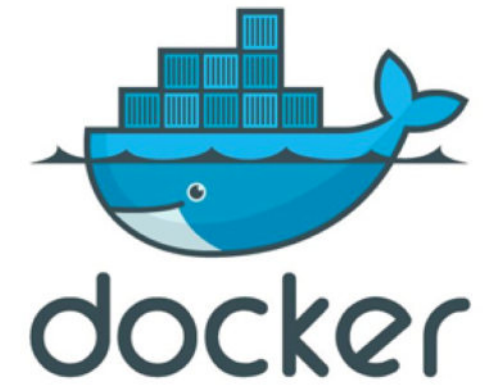
More info: <http://yann.lecun.com/exdb/mnist/>



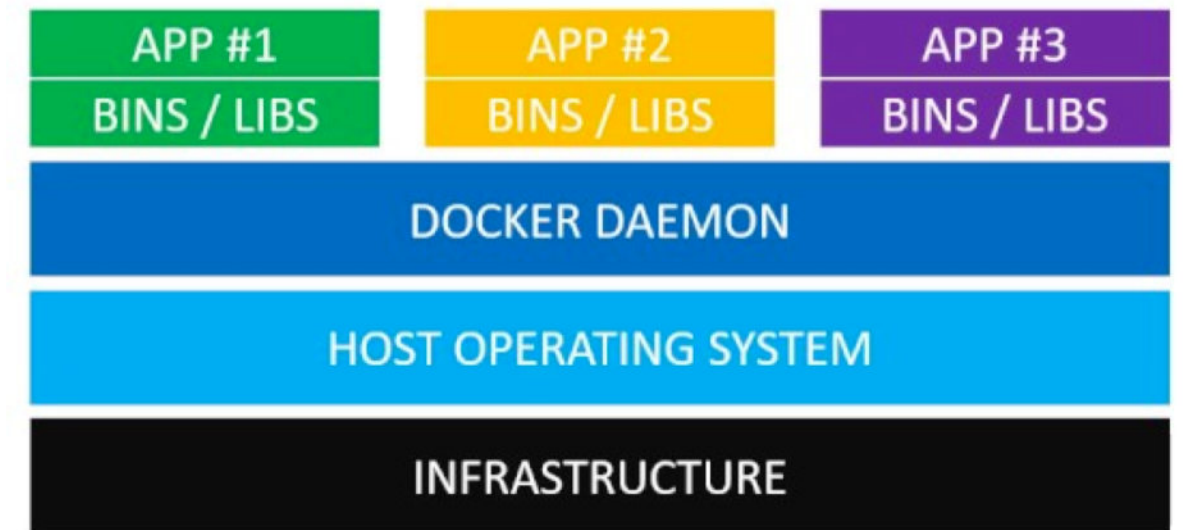
Try it

在容器中使用 TensorFlow

VM vs Docker Container



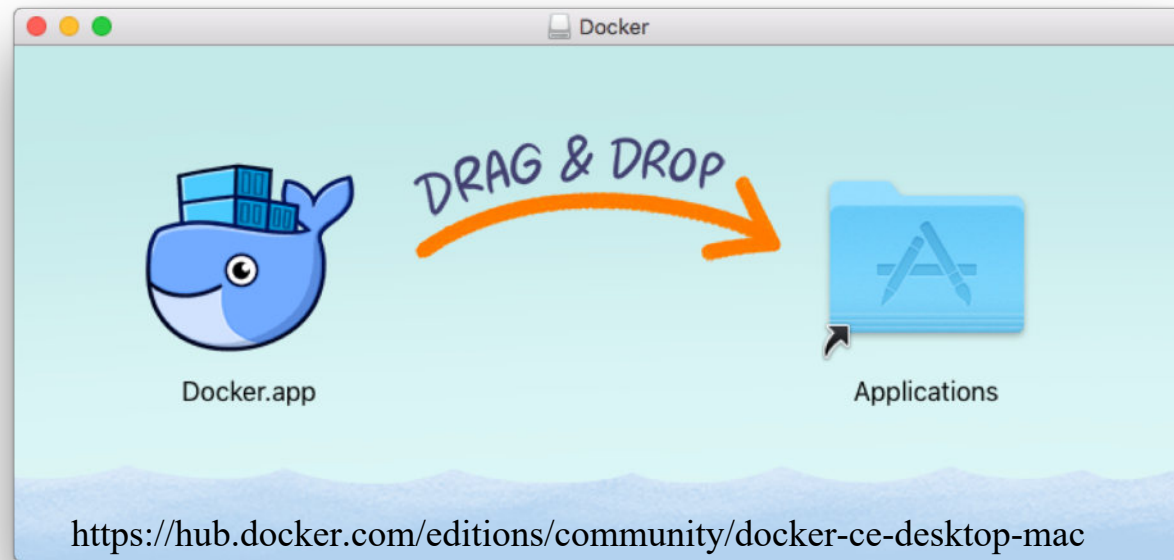
Virtual Machine



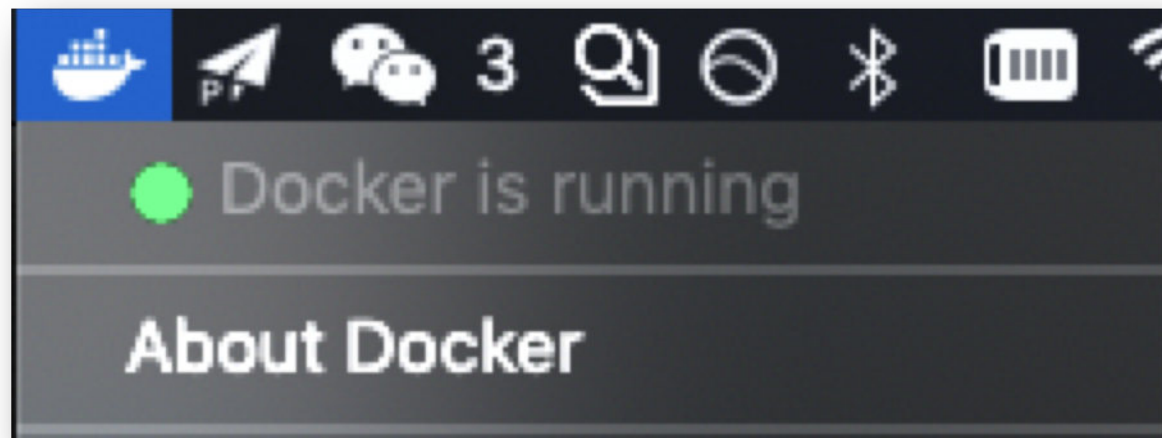
Docker Container

在 Docker 中使用 TensorFlow

1. Install Docker for Mac



2. Run Docker for Mac

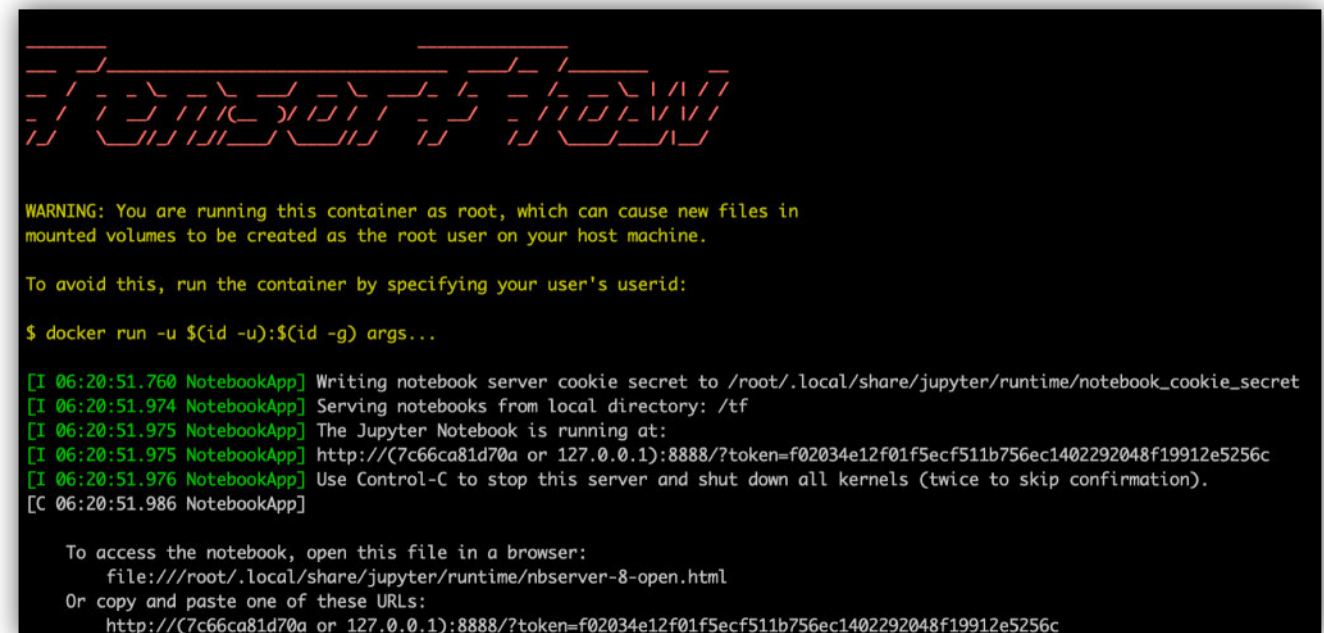


3. Pull a TensorFlow Docker image

```
$ docker pull  
tensorflow/tensorflow:nightly-jupyter
```

4. Start a TensorFlow Docker container

```
$ docker run -it -p 8888:8888 -v  
$(notebook-examples-path):/tf/notebooks  
tensorflow/tensorflow:nightly-jupyter
```



Try it



扫描二维码

试看/购买《TensorFlow 快速入门与实战》视频课程