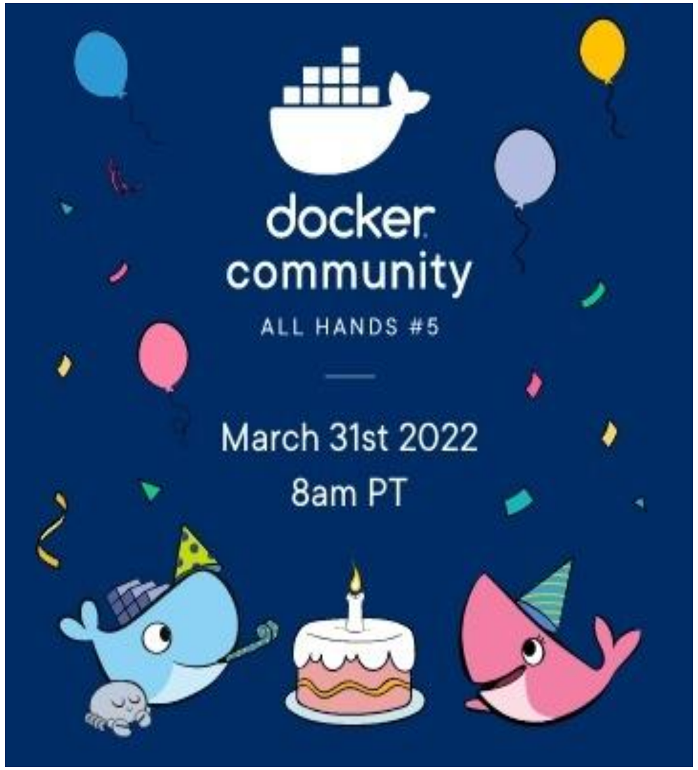
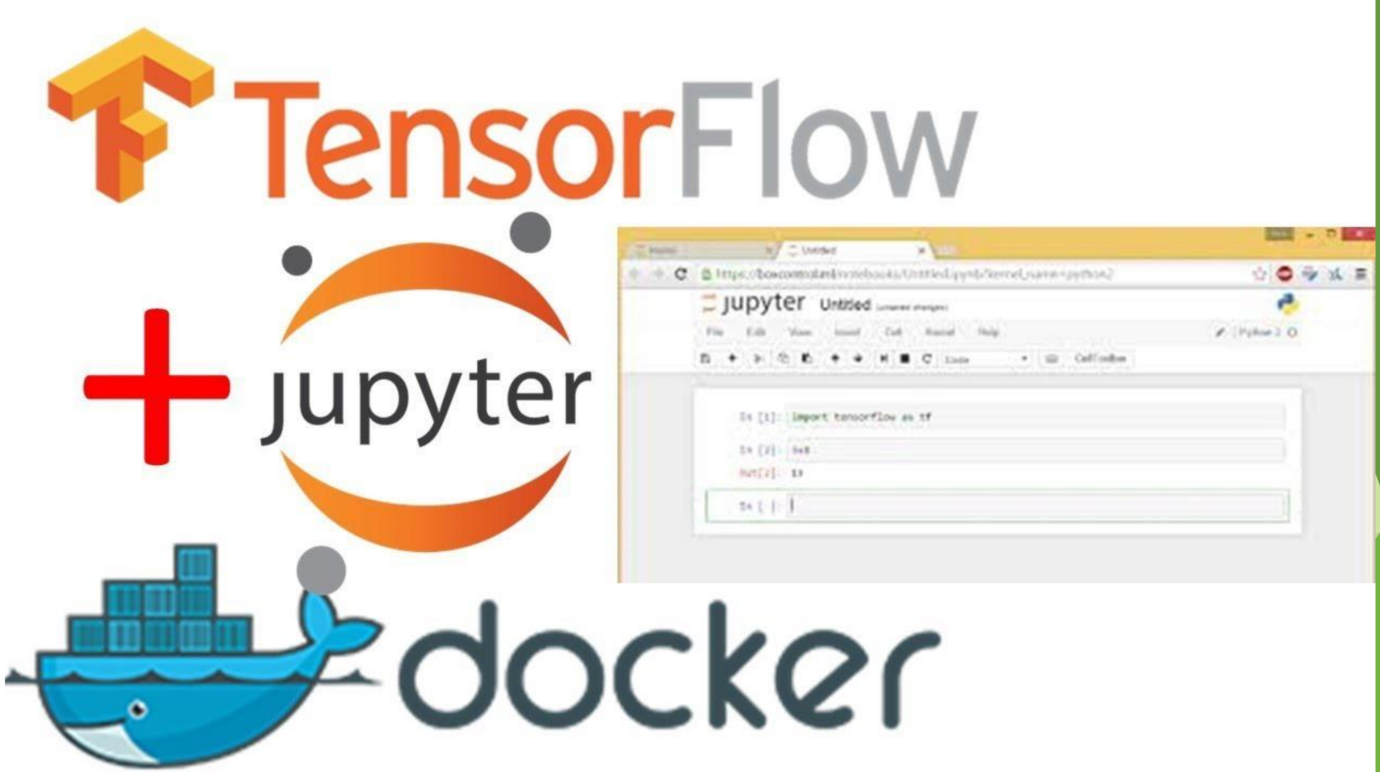
Machine Learning with TensorFLow Object Detection running on Docker

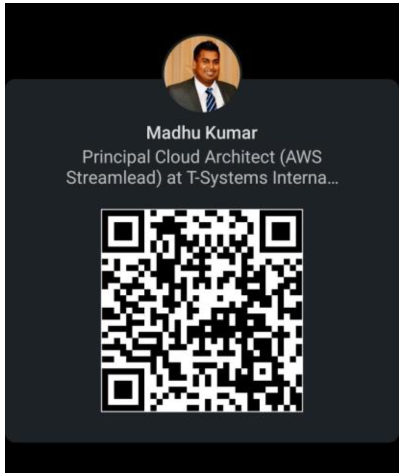




```
In [1]: import tensorflow as tf

In [2]: 5+8
Out[2]: 13

In [ ]:
```

About the Speaker

Yeluri, Madhu Kumar
Principal Cloud Architect
@ T-Systems International
Budapest, Hungary



Madhu Kumar
Principal Cloud Architect (AWS Streamlead) at T-Systems Interna...

- Yeluri, Madhu Kumar
- Madhu is a qualified Principal Cloud Architect and Senior DevSecOps Consultant with over 21 years of IT experience working across multiple regions including Asia, Middle East, the US, Europe and the UK. He is helping many customers transform their business using the cloud. He is leading diverse teams to driving change and deliver business value at scale.
- A certified Amazon Web Services (AWS) Solution Architect and Security Specialist. Product Owner for Container services (Docker, K8s, AWS ECS and EKS). He has worked with many Cloud Partners/Providers (AWS, Rackspace, Wipro, Google, Oracle, Azure, IBM and Vodafone) and successfully managed and implemented multiple Cloud migration projects replacing business-critical core legacy systems across the Telecom, Financial, Banking, Insurance, Retail, and Government sectors.

Agenda

- Surprising Facts about Docker Adoption
- TensorFlow
- Create the Dockerfile
- Build the Dockerfile (Demo)
- Run the Docker Image (Demo)
- Object Detection API (Demo)
- Conclusion
- Questions

Surprising Facts about Docker Adoption

- Nearly One Quarter of Companies Have Adopted Docker
- Docker Now Runs on More Than 20% of Hosts
- Docker Usage Rates Increase with Infrastructure Size
- Half of Docker Environments Are Orchestrated
- The Average Size of a Docker Deployment Has Grown 75% in One Year
- The Most Widely Used Images Are NGINX, Redis, and Postgres
- The Median Docker Organization Runs Eight Containers per Host
- Orchestrated Containers Churn 12x Faster

Source: https://www.datadoghq.com/docker-adoption

TensorFlow

- **TensorFlow** is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.

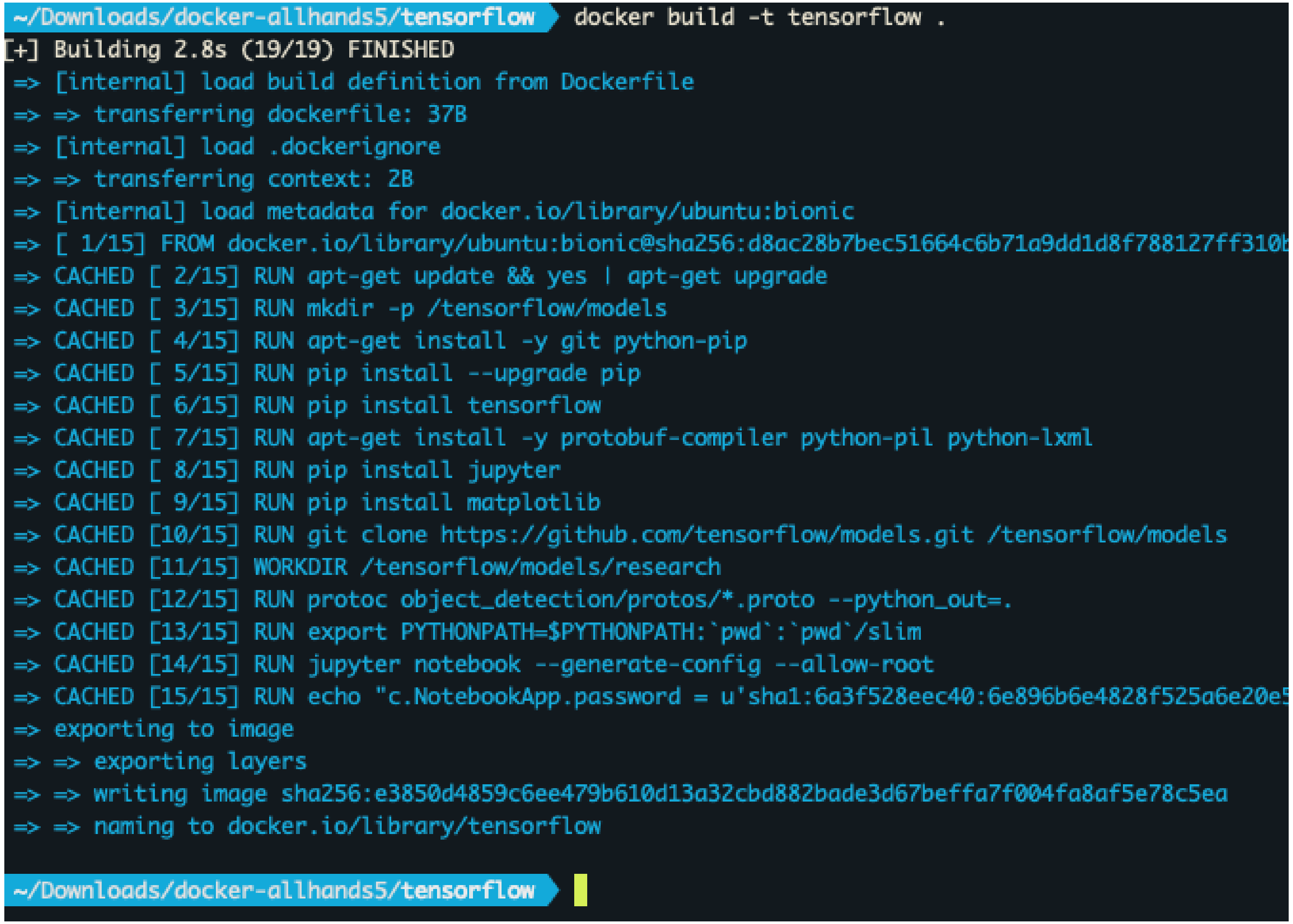
- TensorFlow was developed by the Google Brain team for internal Google use in research and production. The initial version was released under the Apache License 2.0 in 2015. Google released the updated version of TensorFlow, named TensorFlow 2.0, in September 2019.

Create the Dockerfile

```
FROM "ubuntu:bionic"
RUN apt-get update && yes | apt-get upgrade
RUN mkdir -p /tensorflow/models
RUN apt-get install -y git python-pip
RUN pip install --upgrade pip
RUN pip install tensorflow
RUN apt-get install -y protobuf-compiler python-pil python-lxml
RUN pip install jupyter
RUN pip install matplotlib
RUN git clone https://github.com/tensorflow/models.git /tensorflow/models
WORKDIR /tensorflow/models/research
RUN protoc object_detection/protos/*.proto --python_out=.
RUN export PYTHONPATH=$PYTHONPATH:`pwd`:`pwd`/slim
RUN jupyter notebook --generate-config --allow-root
RUN echo "c.NotebookApp.password = u'sha1:6a3f528eec40:6e896b6e4828f525a6e20e5411cd1c8075d68619'" >> /root/.jupyter/jupyter_notebook_config.py
EXPOSE 8888
CMD ["jupyter", "notebook", "--allow-root", "--notebook-dir=/tensorflow/models/research/object_detection", "--ip=0.0.0.0", "--port=8888", "--no-browser"]
```

Demo

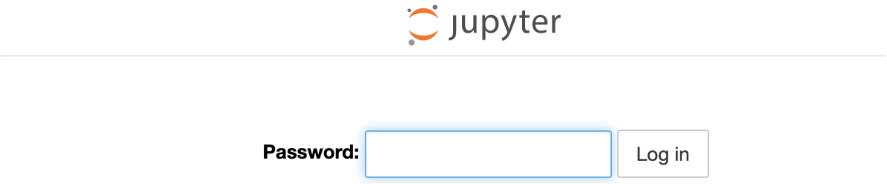
Build the Docker image



```
~/Downloads/docker-allhands5/tensorflow  docker build -t tensorflow .
[+] Building 2.8s (19/19) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 37B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/ubuntu:bionic
 => [ 1/15] FROM docker.io/library/ubuntu:bionic@sha256:d8ac28b7bec51664c6b71a9dd1d8f788127ff310b
 => CACHED [ 2/15] RUN apt-get update && yes | apt-get upgrade
 => CACHED [ 3/15] RUN mkdir -p /tensorflow/models
 => CACHED [ 4/15] RUN apt-get install -y git python-pip
 => CACHED [ 5/15] RUN pip install --upgrade pip
 => CACHED [ 6/15] RUN pip install tensorflow
 => CACHED [ 7/15] RUN apt-get install -y protobuf-compiler python-pil python-lxml
 => CACHED [ 8/15] RUN pip install jupyter
 => CACHED [ 9/15] RUN pip install matplotlib
 => CACHED [10/15] RUN git clone https://github.com/tensorflow/models.git /tensorflow/models
 => CACHED [11/15] WORKDIR /tensorflow/models/research
 => CACHED [12/15] RUN protoc object_detection/protos/*.proto --python_out=.
 => CACHED [13/15] RUN export PYTHONPATH=$PYTHONPATH:`pwd`:`pwd`/slim
 => CACHED [14/15] RUN jupyter notebook --generate-config --allow-root
 => CACHED [15/15] RUN echo "c.NotebookApp.password = u'sha1:6a3f528eec40:6e896b6e4828f525a6e20e5
 => exporting to image
 => => exporting layers
 => => writing image sha256:e3850d4859c6ee479b610d13a32cbd882bade3d67beffa7f004fa8af5e78c5ea
 => => naming to docker.io/library/tensorflow

~/Downloads/docker-allhands5/tensorflow 
```

- **Build the Docker image**
- docker build -t tensorflow .

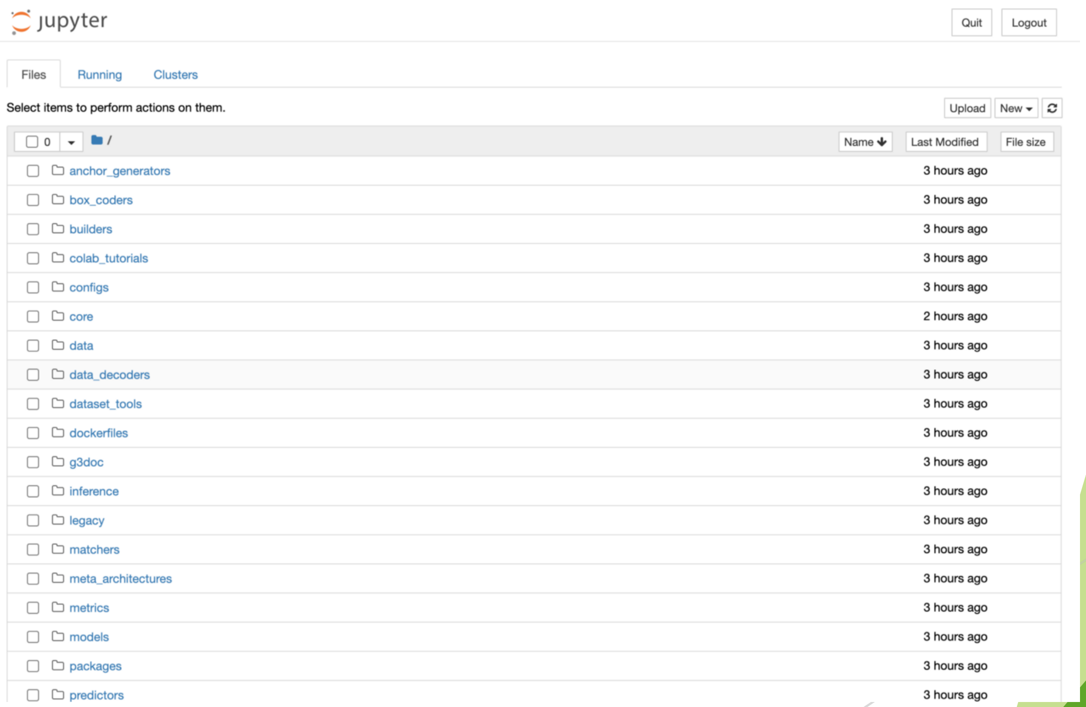
Run the Docker Image



Password: Log in

- Run the Docker container
  - docker run --rm --name tensorflow -p 8888:8888 -d tensorflow

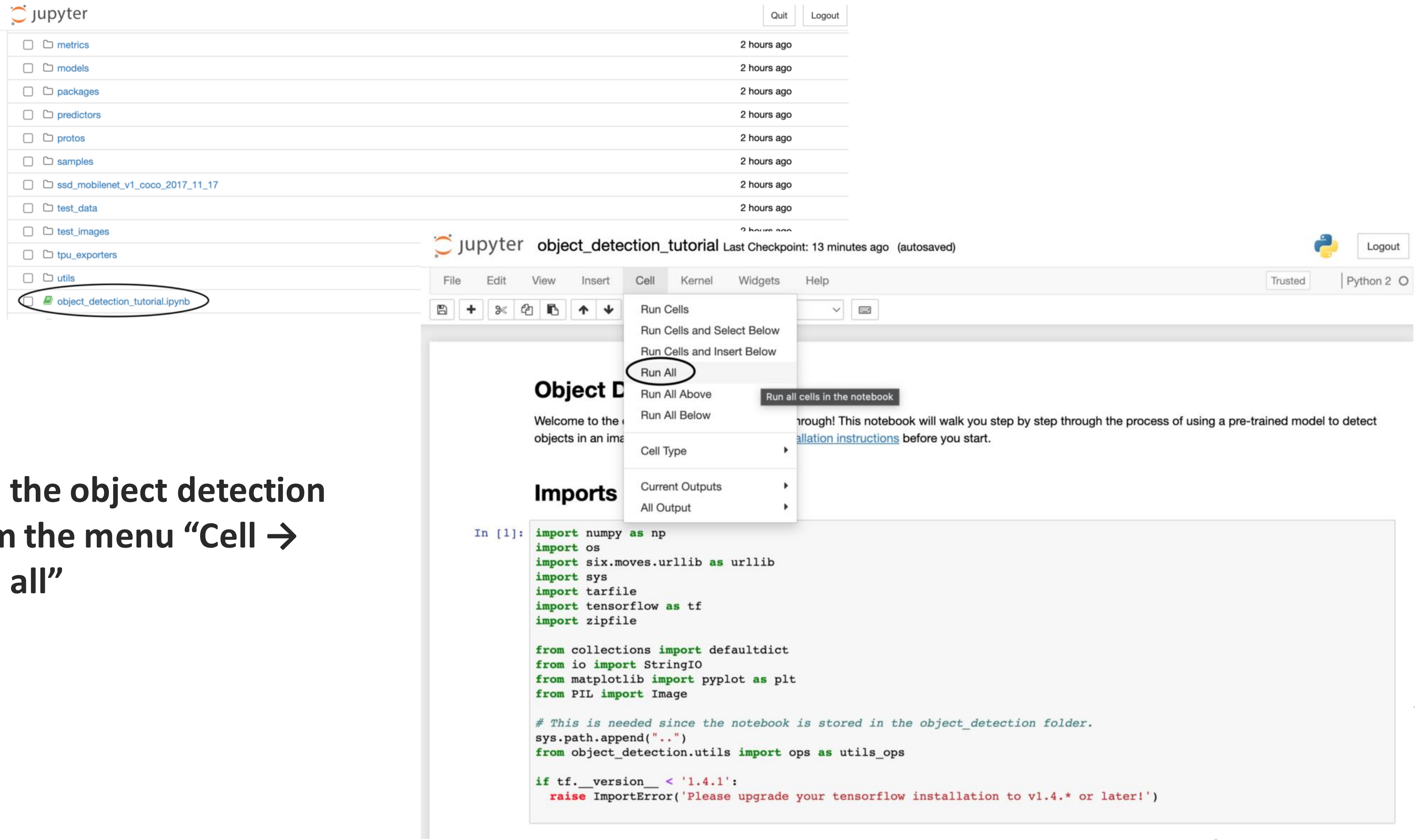
- Run the Docker container
  - Open http://localhost:8888



Select items to perform actions on them.

□ 0 ▾ /	Name ↓	Last Modified	File size
□ anchor_generators		3 hours ago	
□ box_coders		3 hours ago	
□ builders		3 hours ago	
□ colab_tutorials		3 hours ago	
□ configs		3 hours ago	
□ core		2 hours ago	
□ data		3 hours ago	
□ data_decoders		3 hours ago	
□ dataset_tools		3 hours ago	
□ dockerfiles		3 hours ago	
□ g3doc		3 hours ago	
□ inference		3 hours ago	
□ legacy		3 hours ago	
□ matchers		3 hours ago	
□ meta_architectures		3 hours ago	
□ metrics		3 hours ago	
□ models		3 hours ago	
□ packages		3 hours ago	
□ predictors		3 hours ago	

Object Detection API Demo


jupyter
Quit
Logout
jupyter object_detection_tutorial Last Checkpoint: 13 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 2
Run Cells
Run Cells and Select Below
Run Cells and Insert Below
Run All
Run All Above
Run All Below
Cell Type
Current Outputs
All Output
Run all cells in the notebook

metrics	2 hours ago
models	2 hours ago
packages	2 hours ago
predictors	2 hours ago
protos	2 hours ago
samples	2 hours ago
ssd_mobilenet_v1_coco_2017_11_17	2 hours ago
test_data	2 hours ago
test_images	2 hours ago
tpu_exporters	
utils	
object_detection_tutorial.ipynb	

Object D

Welcome to the … rough! This notebook will walk you step by step through the process of using a pre-trained model to detect objects in an ima… allation instructions before you start.

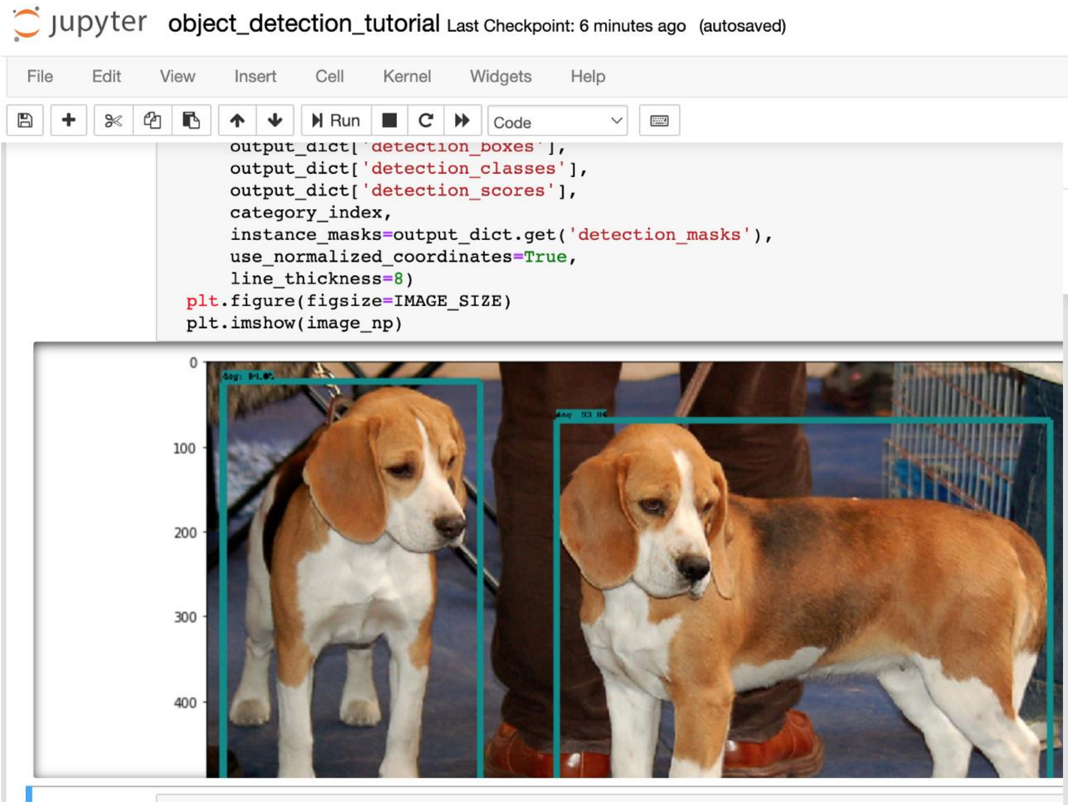
Imports

In [1]:

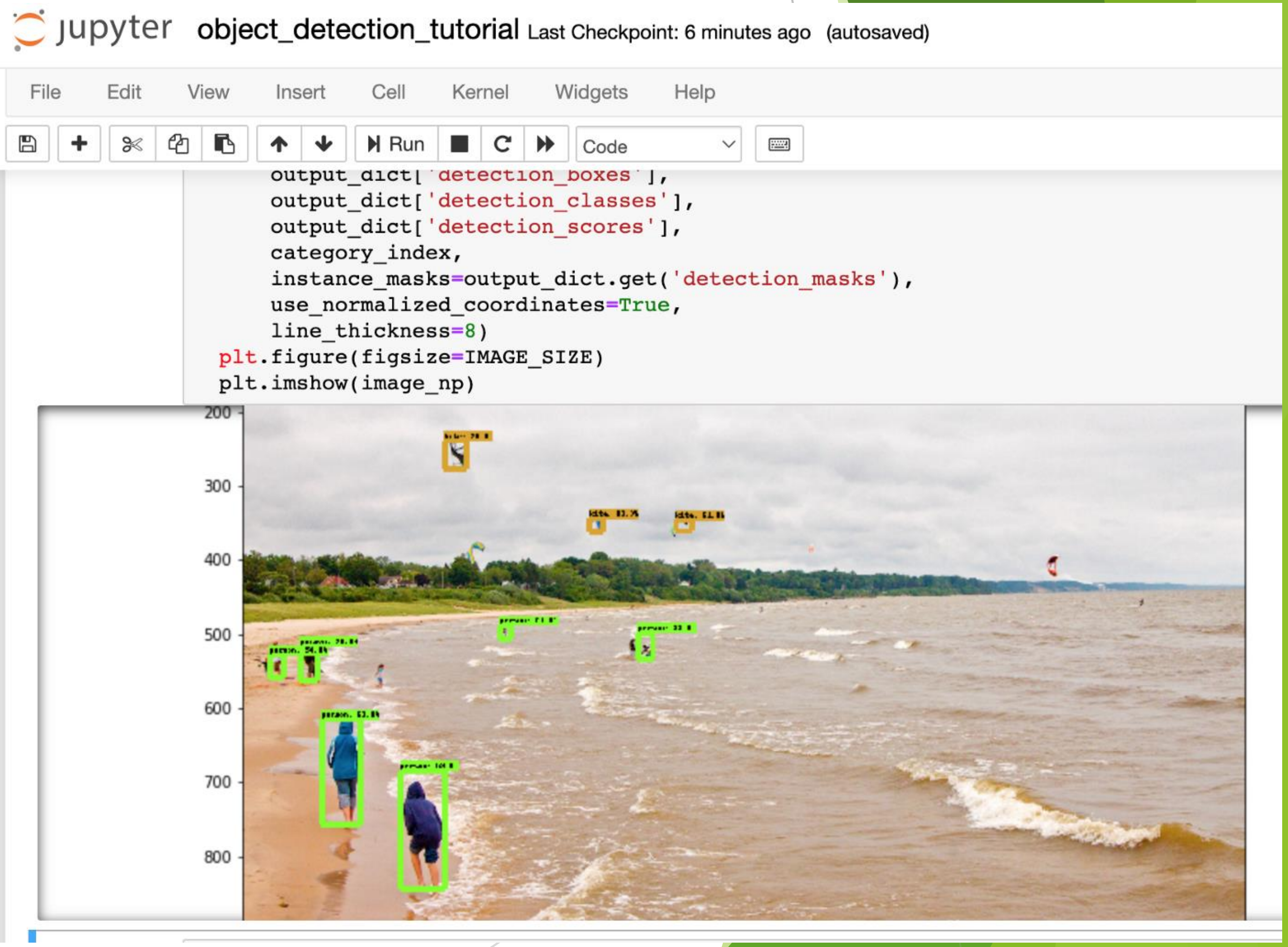
```
import numpy as np
import os
import six.moves.urllib as urllib
import sys
import tarfile
import tensorflow as tf
import zipfile

from collections import defaultdict
from io import StringIO
from matplotlib import pyplot as plt
from PIL import Image

# This is needed since the notebook is stored in the object_detection folder.
sys.path.append("..")
from object_detection.utils import ops as utils_ops

if tf.__version__ < '1.4.1':
  raise ImportError('Please upgrade your tensorflow installation to v1.4.* or later!')
```

Run the object detection from the menu “Cell → Run all”

Results of the object detection


jupyter
object_detection_tutorial Last Checkpoint: 6 minutes ago (autosaved)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Code
0
100
200
300
400

```
        output_dict['detection_boxes'],
        output_dict['detection_classes'],
        output_dict['detection_scores'],
        category_index,
        instance_masks=output_dict.get('detection_masks'),
        use_normalized_coordinates=True,
        line_thickness=8)
    plt.figure(figsize=IMAGE_SIZE)
    plt.imshow(image_np)
```



```
        output_dict['detection_boxes'],
        output_dict['detection_classes'],
        output_dict['detection_scores'],
        category_index,
        instance_masks=output_dict.get('detection_masks'),
        use_normalized_coordinates=True,
        line_thickness=8)
    plt.figure(figsize=IMAGE_SIZE)
    plt.imshow(image_np)
```

Stop the TensorFlow Docker container

```
docker rm -f tensorflow
```

Conclusion

Why Docker?

Docker is my favourite containerisation platform. Why? Docker provides a way to run applications securely isolated in a container, packaged with all its dependencies and libraries which are required for the application to run. Please refer to my other Docker blogs for more learning on Docker.

Why TensorFlow?

TensorFlow supports various platforms for deploying ML models, be it desktop, mobile, web, or even cloud. Models can be trained using different programming languages like Python, JavaScript, or Swift.

As per the StackOverflow Developers Survey 2020, TensorFlow is one of the most popular frameworks among developers. Around 65% of the surveyed respondents have expressed their interest in continuing to develop models using TensorFlow. Also, with Google's support, the library will be enhanced regularly to fulfill the growing needs of developers.

Questions