

# Object Detection API on Tensorflow for Demo

## Main Repositories

DockerHub : <https://hub.docker.com/u/qhub> Github : [https://github.com/qpjkw/tfod\\_ces2019.git](https://github.com/qpjkw/tfod_ces2019.git) Tensorflow  
Object Detection API : [https://github.com/tensorflow/models/tree/master/research/object\\_detection](https://github.com/tensorflow/models/tree/master/research/object_detection)

## Build container images

Make sure you have already imported the customized docker image into container station. You can pull images from Dockerhub repository. In this tutorial, we recommend GPU-based docker image.

### CUDA 9 :

```
docker pull qhub/tfod-ces2019:1.0-gpu
```

### CUDA 10 :

Download dockerfile and put to NAS `Public` folder : [DockerFile-CUDA10](#)

```
cd /share/Public  
docker build --rm --tag=qhub/tfod-ces2019:1.13-gpu --file=Dockerfile-GPU .
```

## Starting a container

### Use command

And then instantiate a container via the below command.

### CUDA 9 :

```
docker run --rm -it --name ces2019gpu --ipc=host -p 28888:8888 -p 26006:6006 --device  
/dev/nvidia0:/dev/nvidia0 --device /dev/nvidiactl:/dev/nvidiactl --device /dev/nvidia-  
uvm:/dev/nvidia-uvm -v /share/CACHEDEV1_DATA/.qpkg/NVIDIA_GPU_DRV/usr/:/usr/local/nvidia  
qhub/tfod-ces2019:1.0-gpu
```

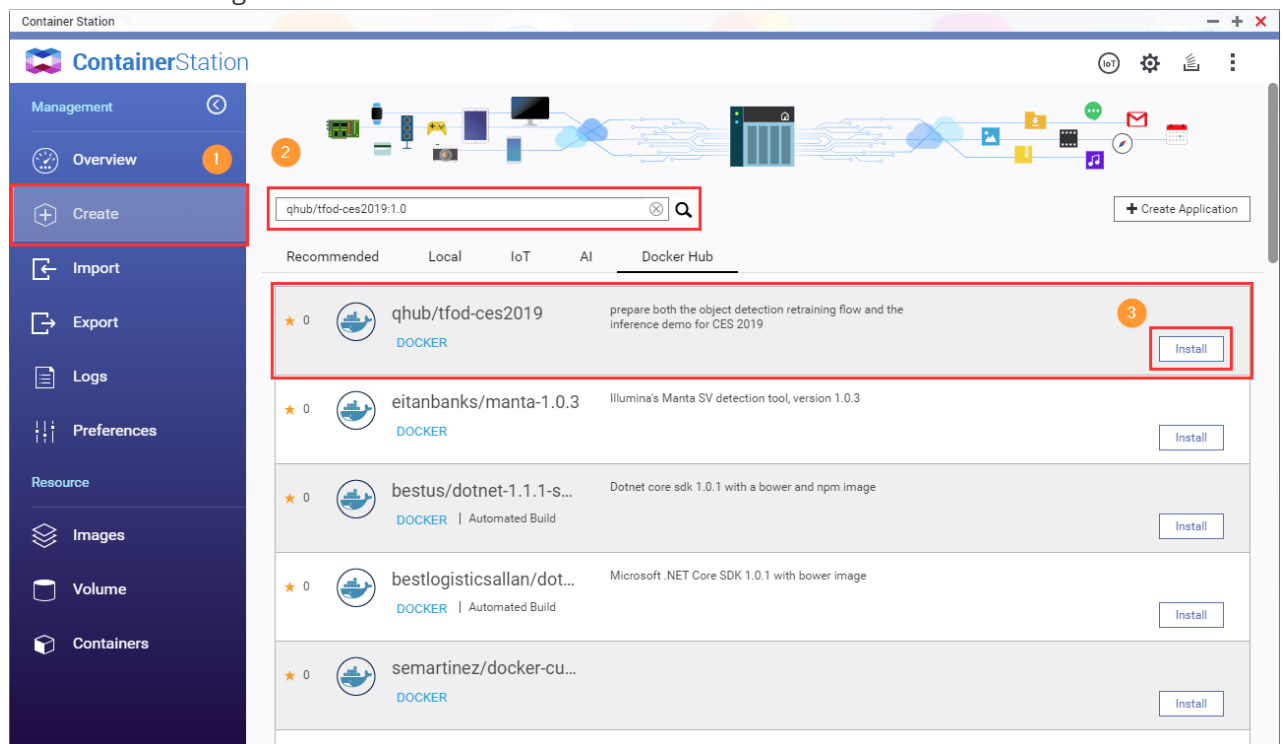
### CUDA 10 :

```
docker run --rm -it --name ces2019gpu --ipc=host -p 28888:8888 -p 26006:6006 --device  
/dev/nvidia0:/dev/nvidia0 --device /dev/nvidiactl:/dev/nvidiactl --device /dev/nvidia-  
uvm:/dev/nvidia-uvm -v /share/CACHEDEV1_DATA/.qpkg/NVIDIA_GPU_DRV/usr/:/usr/local/nvidia  
qhub/tfod-ces2019:1.13-gpu
```

## Use web GUI

## CUDA 9 :

### 1. Search docker image from docker hub



### 2. Choice docker image tag (version)



### 3. Change page to **Advanced Settings**

## Create Container

Image : qhub/tfod-ces2019

Name : tfod-ces2019-2

Command : /bin/bash -c "source /etc/bash.bashrc && jupyter notebook --notebook-dir=/notebooks --ip 0.0.0.0 --no-browser --allow-root"


Entrypoint :

Auto start : ☒

CPU Limit :  100 %

Memory Limit :  31869 MB

The CPU limit must be within 10-100 %. The memory limit must be within 64-31869MB.


 [Advanced Settings >>](#)

Create

Cancel

### 4. Setting port forwarding

## Create Container

 [Advanced Settings >>](#)

Environm...

Network

Device

Shared F...

Container Hostname :




Container MAC Address : 02:42:3C:54:C8:FC

Network Mode : NAT

Use default NAT network (command: --net=bridge)

Port Forwarding

Add

Host	Container	Protocol	
25000	5000	TCP	
26006	6006	TCP	
28888	8888	TCP	

Create

Cancel

## 5. Mount GPU card

### Create Container

Advanced Settings >>

Environ...

1 Network

Device

Shared F...

☐ Run containers in privileged mode.

Due to the compatibility issues of various vendors and developers, successfully mounting graphics card to container is not guaranteed. QNAP is not responsible for any consequence of the use of the graphic card service. If graphics card setting was changed (ex, Driver update), all containers that use graphics card resource may need to be recreated or unable to start.

GPU

GP106 [GeForce GTX 1060 6GB], PCIe Slot=1

Add

Device

Permission

No Data

Add

3 Create

Cancel

## 6. Created container

Container Station									
Management									
Overview									
Create									
Import									
Export									
Logs									
Preferences									
Resource									
Images									
Volume									
Containers									
<input type="checkbox"/>	<input type="checkbox"/>	DOCKER	pytorch/pytorch:1.0.1-cuda10.0-cud...						
<input type="checkbox"/>	<input type="checkbox"/>	DOCKER	PYTORCH-CPU-TESTING pytorch/pytorch:1.0.1-cuda10.0-cud...	0 %	0 %	0 B / s	0 B / s		
<input type="checkbox"/>	<input type="checkbox"/>	DOCKER	pytorch-TT pytorch/pytorch:1.0.1-cuda10.0-cud...						
<input type="checkbox"/>	<input type="checkbox"/>	DOCKER	serene_mayer tensorflow/tensorflow:1.11.0-gpu-py3						
<input type="checkbox"/>	<input type="checkbox"/>	DOCKER	tensorflow tensorflow/tensorflow:1.12.0-gpu						
<input type="checkbox"/>	<input type="checkbox"/>	DOCKER	tensorflow-1 tensorflow/tensorflow:0.11.0-gpu						
<input type="checkbox"/>	<input type="checkbox"/>	DOCKER	tensorflow-2 tensorflow/tensorflow:1.4.1-gpu-py3						
<input type="checkbox"/>	<input type="checkbox"/>	DOCKER	tensorflow-2.0a-gpu-dennistest tensorflow/tensorflow:2.0.0a0-gpu-...						
<input type="checkbox"/>	<input type="checkbox"/>	DOCKER	tensorflow-3 tensorflow/tensorflow:1.10.0-rc0-gpu						
<input type="checkbox"/>	<input type="checkbox"/>	DOCKER	tensorflow-4 tensorflow/tensorflow:1.11.0-gpu-py3						
<input type="checkbox"/>	<input type="checkbox"/>	DOCKER	tensorflow-5 tensorflow/tensorflow:1.11.0-gpu-py3						
<input type="checkbox"/>	<input type="checkbox"/>	DOCKER	tfod-ces2019-1 qhub/tfod-ces2019:1.0-gpu-data-1	0 %	5 %	0 B / s	0 B / s		

CUDA 10 :

1. Change to **images** page and create `qhub/tfod-ces2019:1.13-gpu` container

ContainerStation

Management

- Overview
- Create
- Import
- Export
- Logs
- Preferences

Resource

- Images**
- Volume
- Containers

Images

Search name

Pull Remove

Type	Name	Version	ID	Size	Image Created	Actions	
<input checked="" type="checkbox"/>	Docker	qhub/tfod-ces2019	1.13-gpu	d4fc510c75f6	5.06 GB	2019/04/02 06:42:35	<b>+</b> ⬆️ ➡️ ✕
<input checked="" type="checkbox"/>	Docker	tensorflow/tensorflow	2.0.0a0-gpu-py3-...	1e236feb24d3	3.26 GB	2019/03/06 17:00:41	+ ⬆️ ➡️ ✕
<input type="checkbox"/>	Docker	tensorflow/tensorflow	1.13.1-gpu-py3-j...	8c77abe6b462	3.33 GB	2019/02/27 17:01:07	+ ⬆️ ➡️ ✕
<input type="checkbox"/>	Docker	wordpress	undefined-data-1	69ab83f49c1c	401.01 MB	2019/02/15 02:05:11	+ ⬆️ ➡️ ✕
<input checked="" type="checkbox"/>	Docker	wordpress	latest	7539ce0f28d0	401.01 MB	2019/02/09 04:13:53	+ ⬆️ ➡️ ✕
<input checked="" type="checkbox"/>	Docker	pytorch/pytorch	1.0.1-cuda10.0-c...	1d67d1a473d2	5.75 GB	2019/02/08 20:21:52	+ ⬆️ ➡️ ✕
<input checked="" type="checkbox"/>	Docker	pytorch/pytorch	latest	e19f3b87dbf3	3.18 GB	2019/02/08 20:19:54	+ ⬆️ ➡️ ✕
<input checked="" type="checkbox"/>	Docker	mysql	5.7	e47e309f72c8	354.94 MB	2019/02/06 07:06:37	+ ⬆️ ➡️ ✕
<input checked="" type="checkbox"/>	Docker	paperspace/fastai	1.0-CUDA9.2-bas...	16b32f6b557f	5.32 GB	2019/01/25 19:40:29	+ ⬆️ ➡️ ✕
<input checked="" type="checkbox"/>	Docker	paperspace/fastai	1.0-release	1c176231a53d	12.65 GB	2019/01/25 17:40:21	+ ⬆️ ➡️ ✕
<input checked="" type="checkbox"/>	Docker	qhub/tfod-ces2019	1.0-gpu-data-1	32011cb28280	5.70 GB	2019/01/02 07:11:24	+ ⬆️ ➡️ ✕
<input checked="" type="checkbox"/>	Docker	qhub/tfod-ces2019	1.0-gpu	938cc64503c6	5.70 GB	2018/12/28 08:04:22	+ ⬆️ ➡️ ✕
<input type="checkbox"/>	Docker	qhub/tfod-ces2019	1.0	c496bda5052e	3.85 GB	2018/12/28 07:55:19	+ ⬆️ ➡️ ✕

2. Change page to **Advanced Settings**

Create Container

Image : qhub/tfod-ces2019

Name : tfod-ces2019-2

Command : /bin/bash -c "source /etc/bash.bashrc && jupyter notebook --notebook-dir=/notebooks --ip 0.0.0.0 --no-browser --allow-root"

Entrypoint :

Auto start : ☒

CPU Limit : 100 %

Memory Limit : 31869 MB

The CPU limit must be within 10-100 %. The memory limit must be within 64-31869MB.

**⚙️ Advanced Settings >>**

Create Cancel

### 3. Setting port forwarding

## Create Container

⚙️ [Advanced Settings >>](#)

Environm...

**Network**

Device

Shared F...

Container Hostname :

Container MAC Address : 02:42:3C:54:C8:FC

Network Mode : NAT

Use default NAT network (command: --net=bridge)

Port Forwarding

Host	Container	Protocol	
25000	5000	TCP	🗑️
26006	6006	TCP	🗑️
28888	8888	TCP	🗑️

CreateCancel

### 4. Mount GPU card

## Create Container

⚙️ [Advanced Settings >>](#)

Environm...

**1 Network**

**2 Device**

Shared F...

☐ Run containers in privileged mode.

Due to the compatibility issues of various vendors and developers, successfully mounting graphics card to container is not guaranteed. QNAP is not responsible for any consequence of the use of the graphic card service. If graphics card setting was changed (ex, Driver update), all containers that use graphics card resource may need to be recreated or unable to start.

GPU

GP106 [GeForce GTX 1060 6GB], PCIe Slot=1

Add

Device

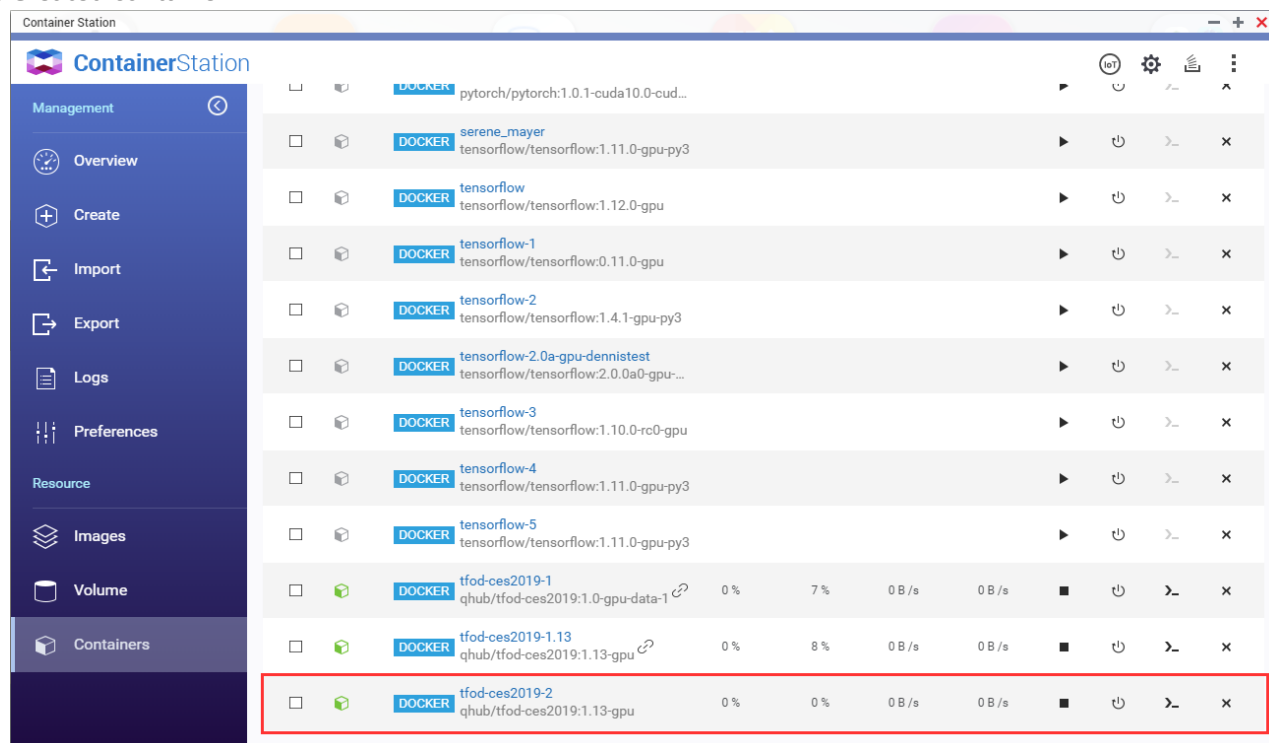
Permission

No Data

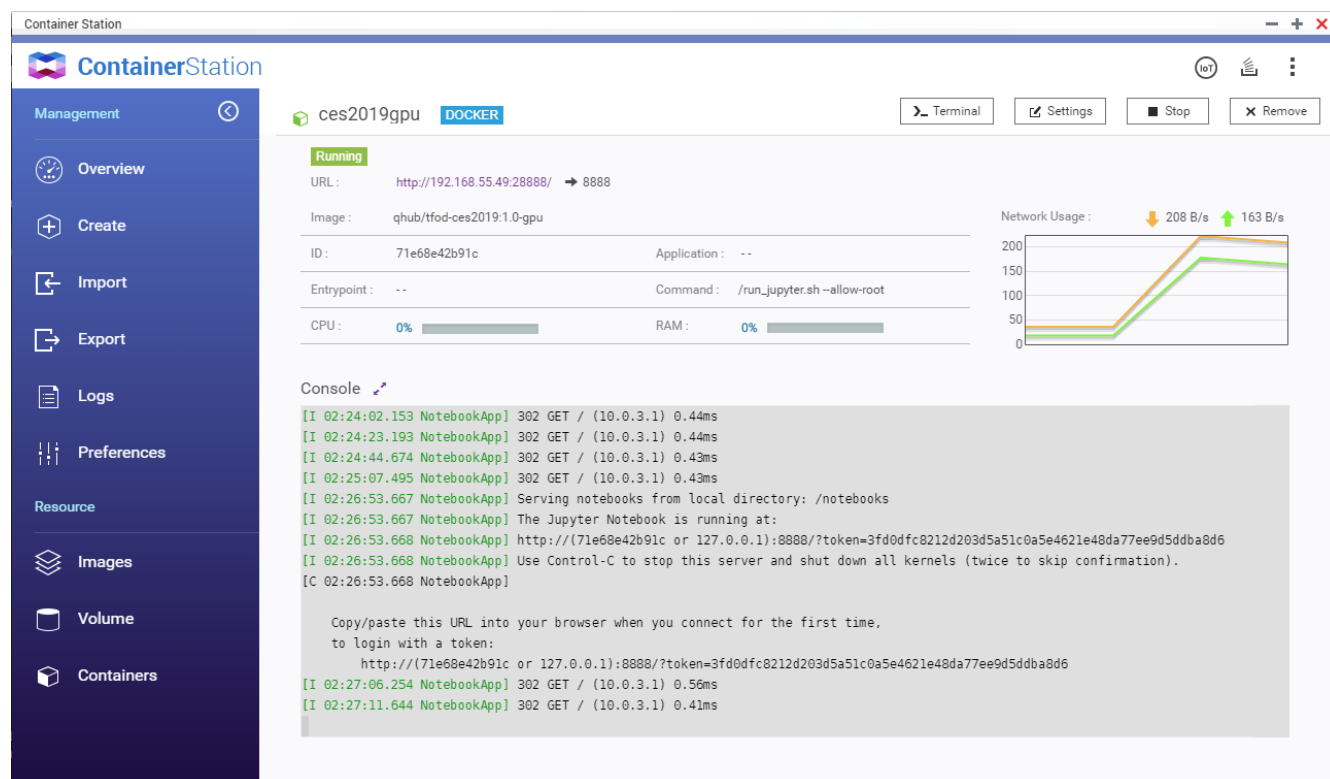
3 Create

Cancel

## 5. Created container



You now can surf the web link to use jupyter notebook (online IDE). <http://<IP>:28888/?token=>"(fetch from terminal)"



## Training

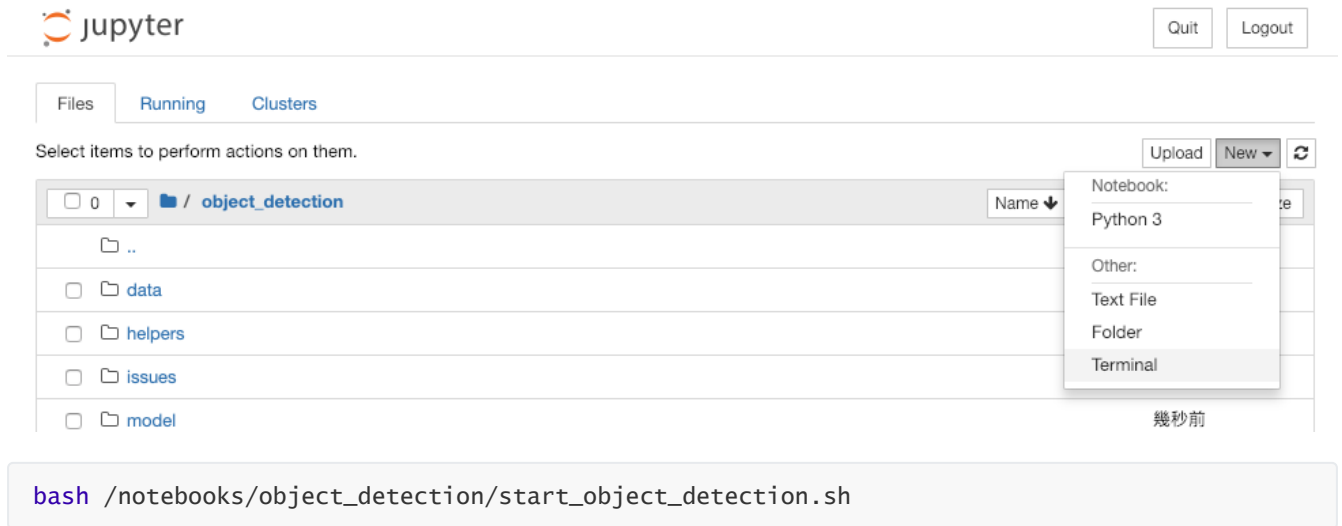
The `/object_detection/data` folder contains the training dataset ( `train.tfrecords` ) and the validation dataset ( `val.tfrecords` ) after preprocessing (classification mark and encoding into TFRecord format).

Script reference : [https://github.com/qpjkw/tfod\\_ces2019/blob/master/object\\_detection/start\\_object\\_detection.sh](https://github.com/qpjkw/tfod_ces2019/blob/master/object_detection/start_object_detection.sh)

Executing `start_object_detection.sh` will first:

1. Generate training config ( `set_training_configuration.py` ) for Dataset pre-processing
2. Execute Tensorboard to view the training status ( `start_tensorboard.sh` )
3. Start retraining. ( `model_main.py` )
4. Generate new **model** + **labelfile**

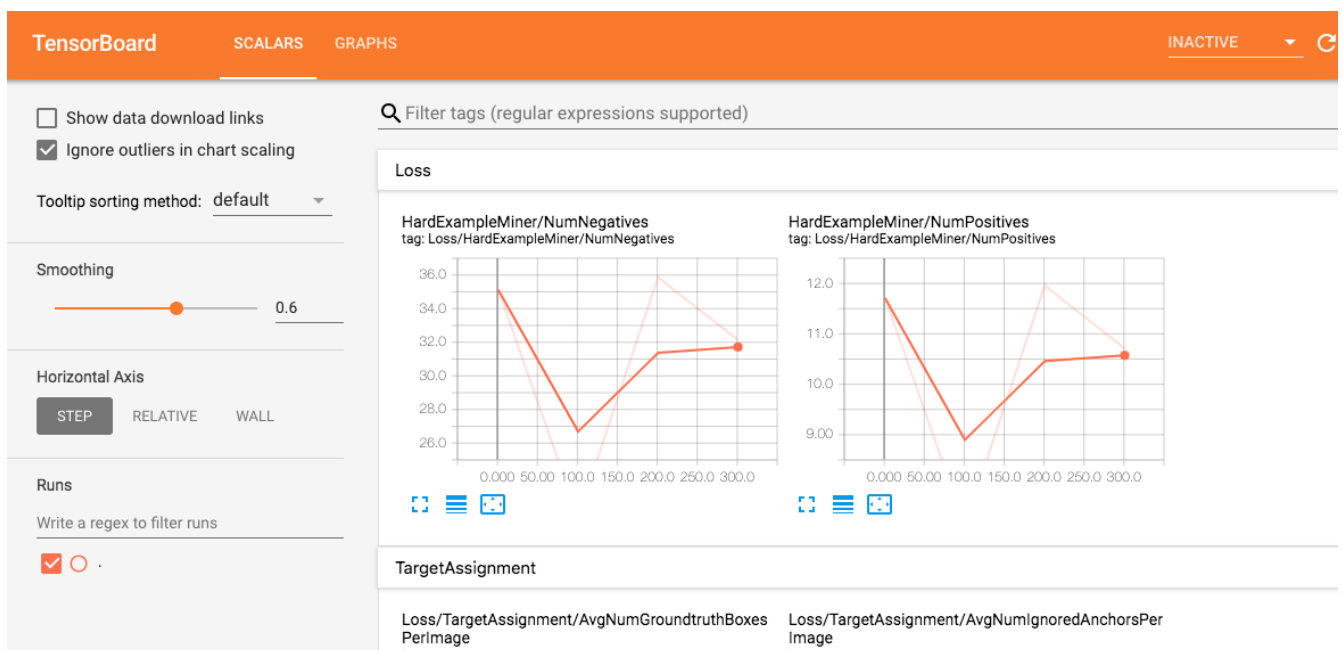
Execute the bash script to start a training. You can open a terminal by clicking the buttons [ `new > terminal` ]. After you open a terminal, copy the below command and paste on it to start a retraining task.



The image shows the JupyterLab interface. At the top, there's a 'Quit' and 'Logout' button. Below that, there are tabs for 'Files', 'Running', and 'Clusters'. The 'Files' tab is active, showing a file browser for the '/ object\_detection' directory. A dropdown menu is open, showing options: 'Notebook:', 'Python 3', 'Other:', 'Text File', 'Folder', and 'Terminal'. The 'Terminal' option is selected. Below the file browser, there's a terminal window with the command: `bash /notebooks/object_detection/start_object_detection.sh`.

You now can surf the web link to monitor training progresses via Tensorboard.

**`http://<IP>:26006"`**



After the training, you can find the model (.pb) on `/notebooks/object_detection/model`.

**If you stop the training unexpectedly, you can type the above starting training command to continue the training.**



# Inference

Back to jupyter notebook editor, you can edit the notebook `object_detection_demo.ipynb` to demo the object detection on images (the below image is the example).

