# Lab 02-03: Docker Basic

By Nick Wang, Allen Ou, last modified on 02/20, 2020.

The objective of the tutorial is to introduce docker engine and the common usages and assumes students are already already have virtualbox image. Also, before this tutorial, you need to complete [prerequisite document.](https://drive.google.com/open?id=1Yh9grt8vt3CNTDNImxa_gncA7kj1Nq6fW1EV5YD4Llo)

* For anyone has not finished setup-virtualbox users, please finish [02-01.setup-virtualbox](https://docs.google.com/document/d/1xAL16HPAZG5-Ds1CTURTWKHfRzyLuE9n3iKquUrwuJo/edit?usp=sharing).
* For anyone wants to run docker on your native Ubuntu, you could take a look at [Native-Ubuntu-docker](https://drive.google.com/open?id=1ey-CzgqlLGiII0q1LYrR5whvjlUzYmJXwOm5ubCEgRo).

## Hardware and Software Setup

Personal PC with virtualbox.

[**Lab 02-03: Docker Basic**](#_74gram21yqax) **1**

[Hardware and Software Setup](#_chc57ow4x0s1) 1

[Overview](#_x5puhc1p2jjo) 1

[Topics and Activities](#_g0fw0bgwxhdt) 2

[Topic/Activity 1 Get Docker Images](#_9w7jwenc3hbu) 2

[Topic/Activity 2 Run Docker Container](#_60rts91qlmd) 2

[Topic/Activity 3 DockerFile](#_v4uchiwk8car) 4

[Assignment Tasks](#_xd5nvoo27ex0) 5

[Task 1 Get Caffe Docker Image and Run](#_f467azdvd3oa) 5

[MNIST prediction by Jupyter notebook:](#_pt9wsh8ij0pa) 5

[Task 2 Create your own Docker Hub account and build your first own docker image and push it to your slack profile](#_wcepkhty30sy) 6

[Reference](#_uwwglharlfb) 7

## Overview

Estimated Time to Finish: 0.5 hours

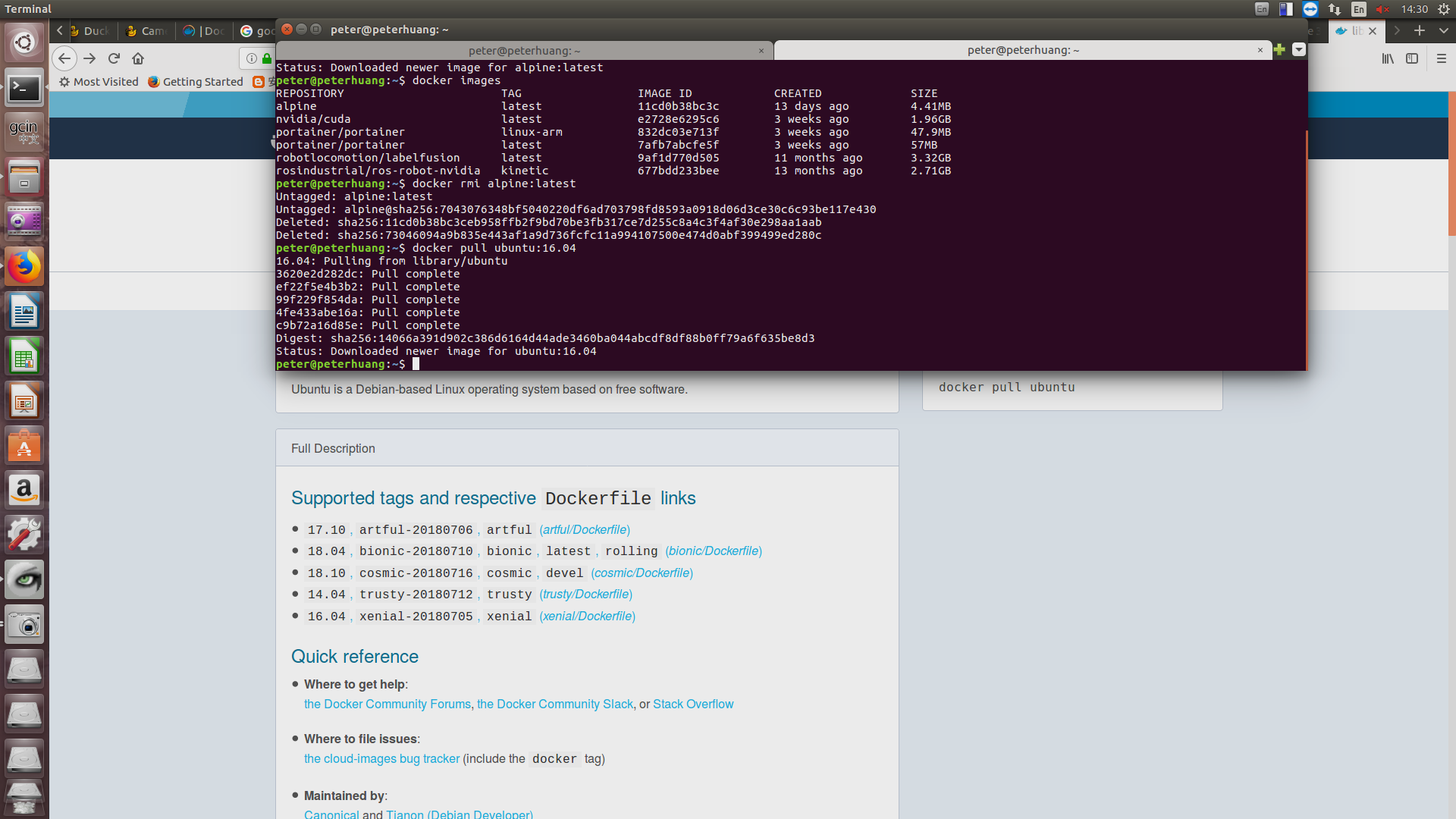
After completing this tutorial you should

* Understand how to use docker engine, and get familiar with several common commands.
* Be able to use existed docker image, and build your own docker image for development.

## Topics and Activities

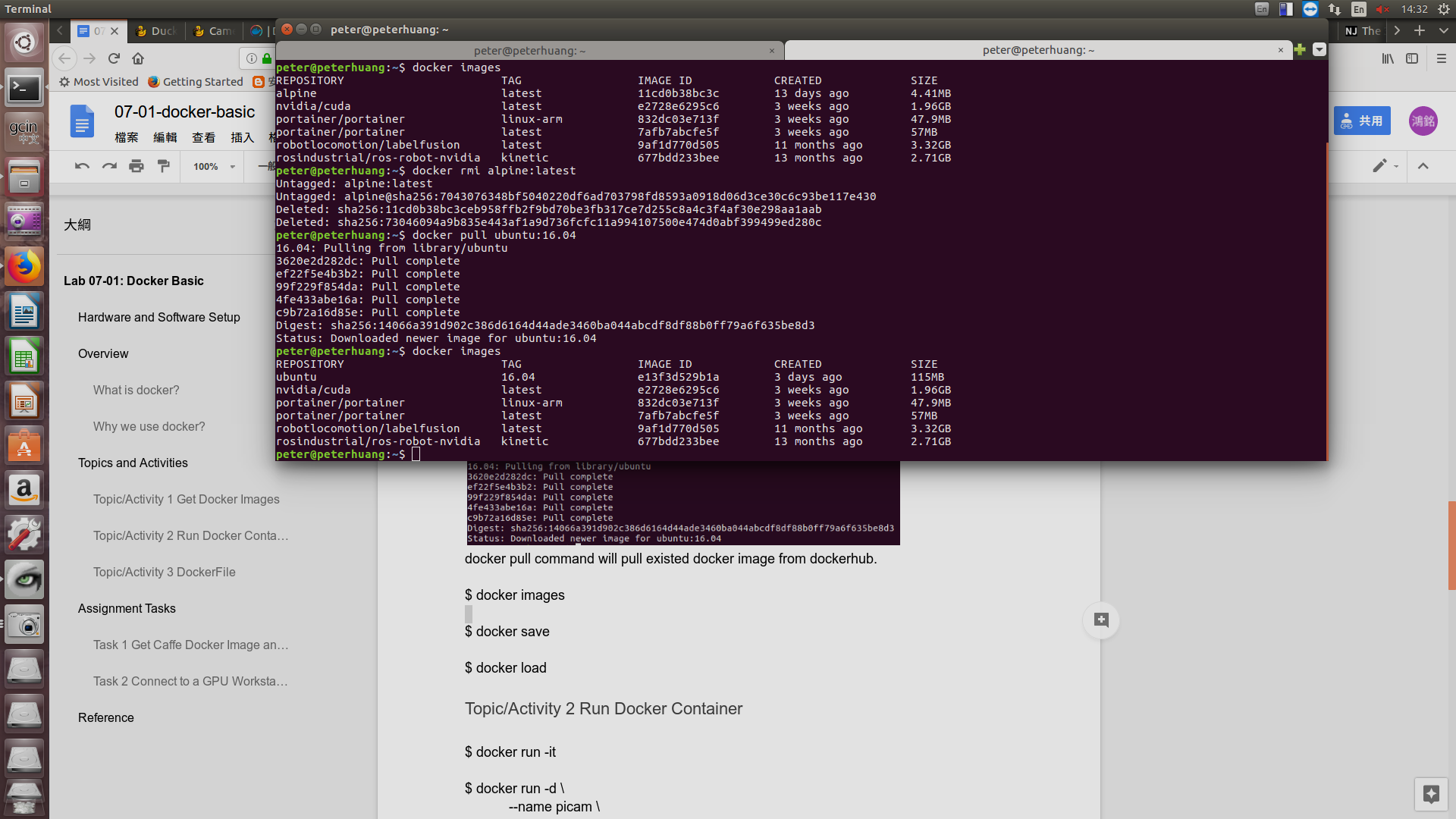
### Topic/Activity 1 Get Docker Images

**Laptop $ docker pull ubuntu:16.04**



“docker pull” command pull existed docker image from dockerhub.

**Laptop $ docker images**



“docker images” command show all the docker image info that you have download.

**Laptop $ cd ~/ && docker save -o ubuntu\_1604.tar ubuntu:16.04**

“docker save” command could export your docker image into a compressed file, and easy to save in any usb flash drive.

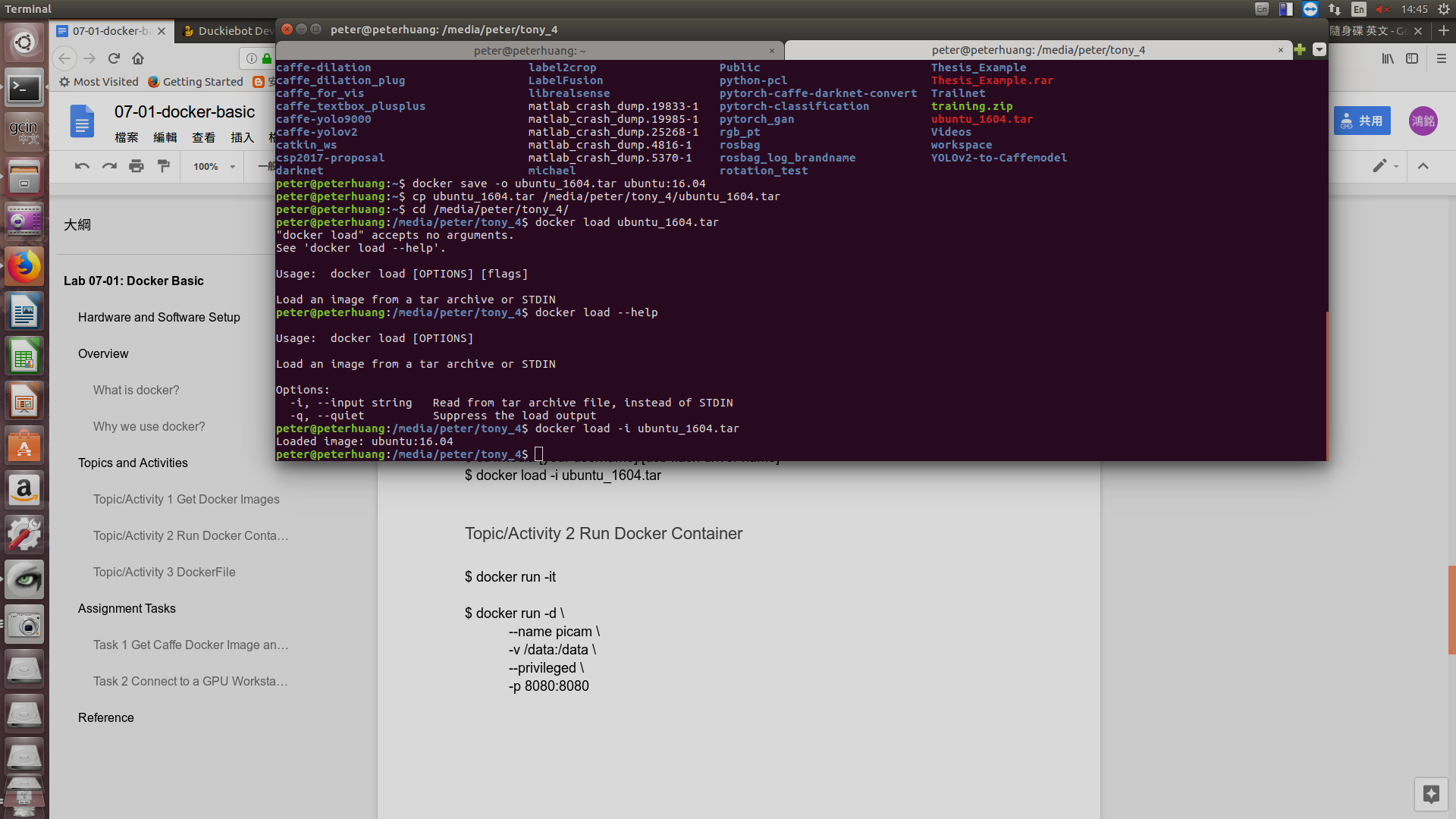
-o parameter: Output filename

**Laptop $ docker rmi ubuntu:16.04**

“docker rmi” command remove the docker image from local.

**Laptop $ docker images**

**Laptop $ cd ~/ && docker load -i ubuntu\_1604.tar**



“docker load” command import your compressed tar image.

-i parameter: Read from tar archive file, instead of STDIN

### Topic/Activity 2 Run Docker Container

Docker can access the external device like usb device, with giving the right parameters, we could directly using them inside of docker.

First we start a docker container that we already have by normal user model.

**Laptop $ docker run -it --rm --name ubuntu ubuntu:16.04**

-it parameter: Open a pseudo-TTY connected to the container’s stdin; creating an interactive bash shell in the container.

--rm parameter: Remove the container when it stopped.

--name parameter: Name of the opening container, this could help you command it easily.

**Container $ ls /dev**



you could see there are few devices could be accessed by a normal user container.

**Container $ exit**

**Laptop $ docker run -it --rm \**

**--name ubuntu \  
-v /home/[username]:/hosthome \  
--privileged \**

**-p 8080:8080 \**

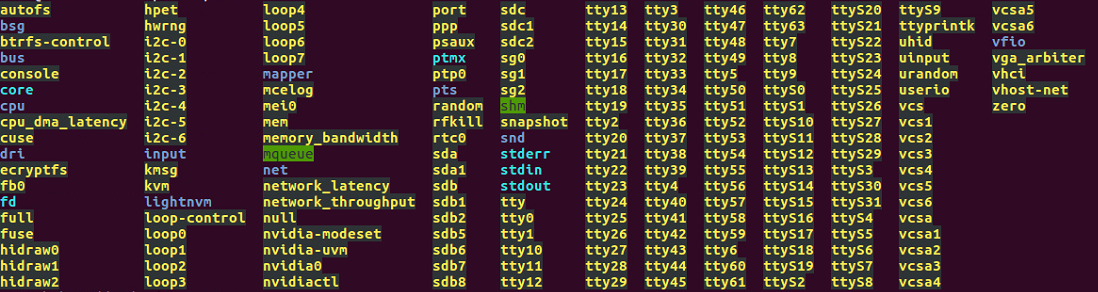
**ubuntu:16.04**

-v parameter : volume

--privileged parameter: privileged mode

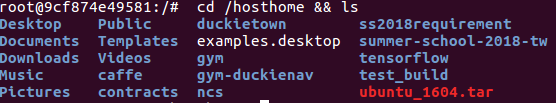
-p parameter: port mapping

**Container $ ls /dev**



You could see a lot of devices could be accessed by privileged mode.

**Container $ cd /hosthome && ls**



Access files located at host by mounted volume between host and container

Open another terminal

**Laptop $ docker port ubuntu**



Container port 8080 is mapping to host port 8080.

**Container $ exit**

### Topic/Activity 3 DockerFile

First download [test\_build.tar](https://drive.google.com/file/d/13gPjYG5RbFv1F6lADw25K6JAdKFTJvtF/view?usp=sharing) and untar it and move the folder to ~/

We could choose to build our image by starting from scratch. But normally building our own image from some official image is more convenient to set up our develop environment.

Dockerfile is the key part to build our own image.

**Laptop $ cd ~/test\_build**

**Laptop $ vim Dockerfile**

|  |
| --- |
| FROM: Creates a layer from the Docker image.  RUN: Execute any commands in a new layer on top of the current image and commit the results.  COPY: Adds files from your Docker client’s current directory.  EXPOSE: Indicates the ports on which a container listens for connections.  CMD: Specifies what command to run within the container by default. |

**Laptop $ docker build -t test\_ubuntu -f Dockerfile .**

**Laptop $ docker images**



Now you have a new images with new packages installed. Also this image will execute default command if you not give the specific command.

**Laptop $ docker run -it --rm \**

**--name ubuntu \**

**test\_ubuntu**



## Assignment Tasks

### Task 1 Get Caffe Docker Image and Run

**Laptop $ docker pull peterx7803/base\_image:cpu\_ros\_caffe**

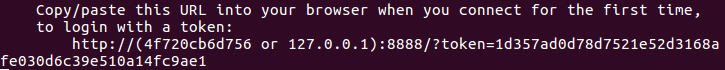
**Laptop $ docker run -it --name caffe -p 8888:8888 peterx7803/base\_image:cpu\_ros\_caffe bash**

**Container $ cd /opt/caffe/examples/mnist**

**Container $ jupyter notebook --allow-root --no-browser --ip="\*"**

Copy link from terminal and paste on your web browser, like following:

*127.0.0.1:8888/?token=xxxxxxxxxxxxxxxxxxxxxxx*



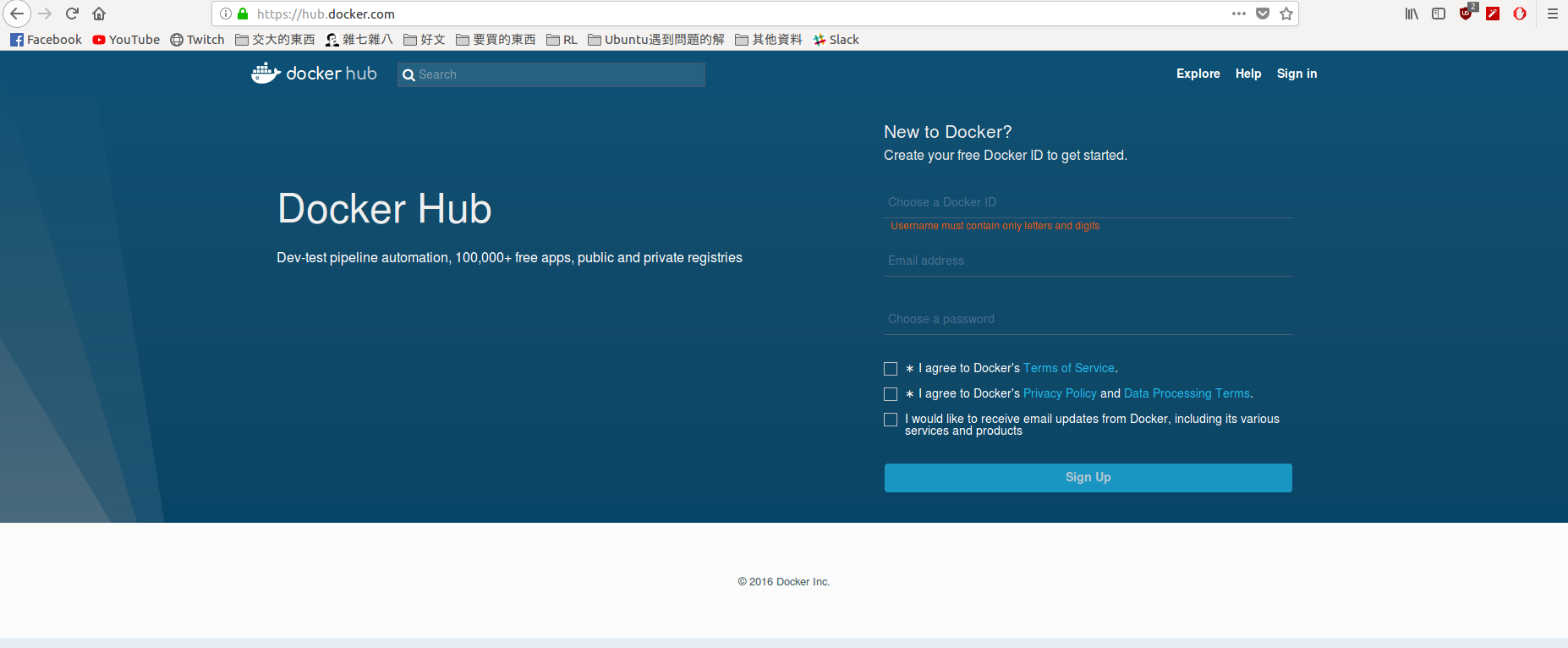
### MNIST prediction by Jupyter notebook:

[Checkpoint]

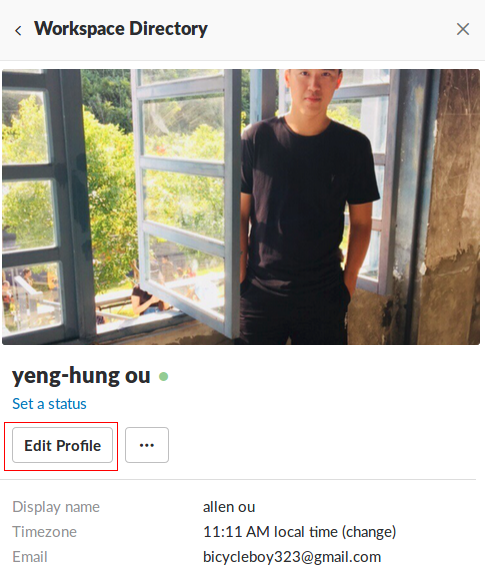
Predict MNIST images with Caffe inside docker.

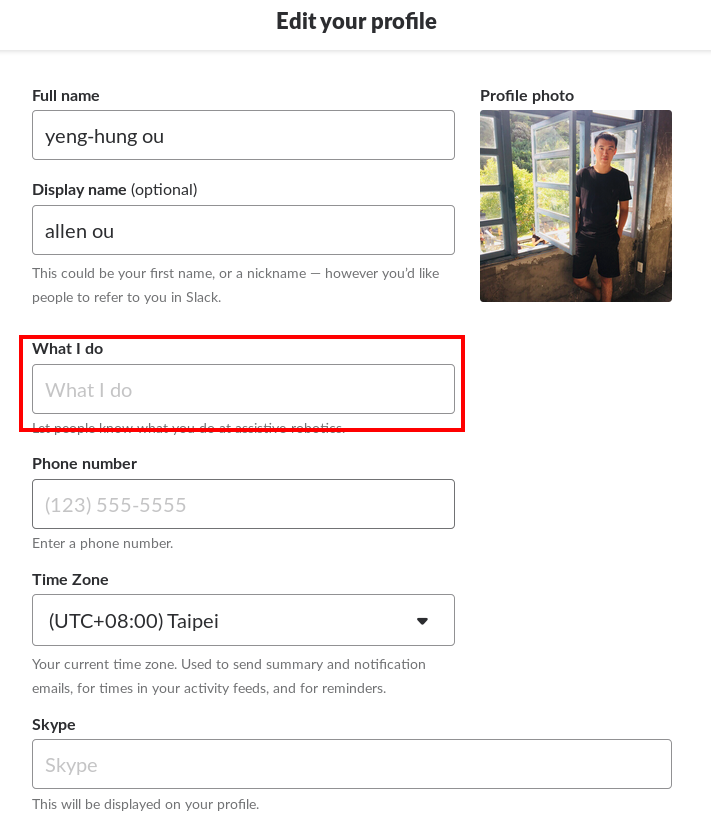
### 

### Task 2 Create your own Docker Hub account and build your first own docker image and push it to your slack profile

You should have your own docker hub account for the mini-RobotX competition, please go to [Docker Hub](https://hub.docker.com/) to create an account and build your first docker image. the content of your image can be anything, even not runnable :D. 

After your docker account and first image is set, update the link to your image to your slack profile





## Reference

* <http://docs.duckietown.org/software_devel/out/docker_intro.html>
* <https://docs.docker.com/engine/>
* <http://caffe.berkeleyvision.org/>