# Tsinghua 2018 Deep Learning Summer School
# Cluster Basic Use Guide

You will learn basic skills about cluster use by running MNIST classifier on TensorFlow.

Specifically, you will learn how to

1. log in cluster using **ssh**.
2. use **vim** to edit file.
3. or, edit locally and using **sftp** to download and upload files.
4. or, edit using **jupyter** notebook.
5. run **python** (TensorFlow) code using GPU.
6. monitor TensorFlow running using **TensorBoard**.

We are not going to explain every command we use, if you don't know what it's used for, Google it.

## 1 Preparation

Everyone will get something like

| | |
|---|---|
| IP | 166.111.69.245 |
| ssh port | 22001 |
| jupyter port | 23001 |
| tensorboard port | 24001 |
| username | test |
| password | 111111 |

they are the key to access the cluster and finish your tasks through this summer school.

We will use the above one as an example through this guide.

## 2 Login

### 2.1 Mac and Linux User

The first step to use the cluster is to log in. **ssh** is a method to access cluster remotely, which needs IP and ssh port of the cluster, your username and password.

**ssh** command basic usage like this:

```
ssh -p PORT USERNAME@IP
```

We got that the cluster IP is *166.111.69.245*, ssh port is *22001*, username is *test* and password is *111111*. You can log in using (Here we use Linux as an example. Mac is similar.)

```
ssh -p 22001 test@166.111.69.245
```

And **ssh** will let you enter password like fig 1.

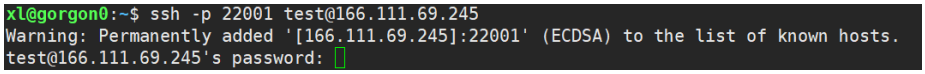


Figure 1: Enter password when using **ssh** to log in.

Then, enter password *111111*, you will see message like fig 2.

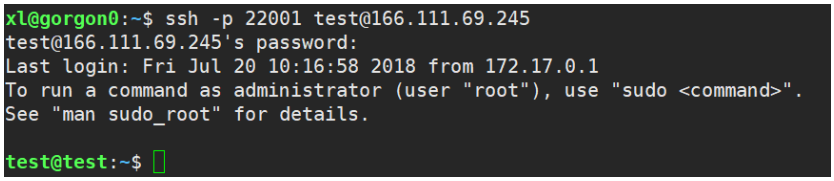


Figure 2: Log in successfully.

## 2.2 Windows User

We recommend to use **Xshell** as **ssh** client and **Xftp** as **ftp**/**sftp** client (we will review it later).

Download **Xshell6** from `https://www.netsarang.com/products/main.html` , home and school use is free, so is **Xftp6**.

Install and open it, you will see a window like fig 3.

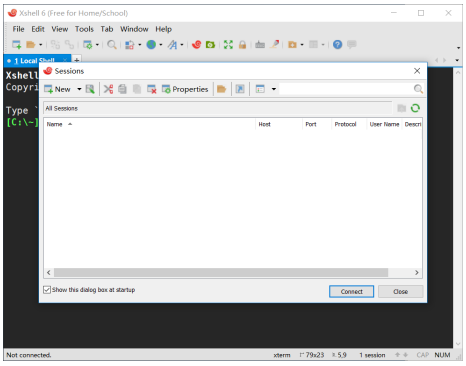


Figure 3: Xshell

Remember that the cluster IP is *166.111.69.245*, ssh port is *22001*, username is *test* and password is *111111*. You can *new* a new session like fig 4.

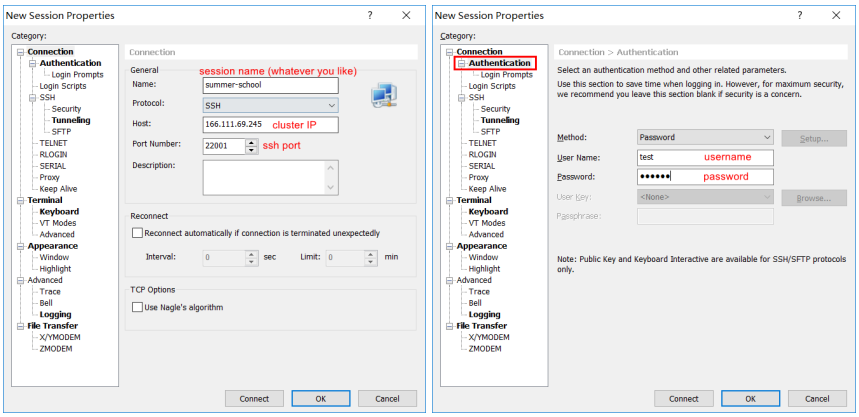


Figure 4: New session.

# 3 Change Your Password

For safety, we give everyone a unique and not-easy-to-remember password. Change password is necessary.

Using

```
passwd
```

command and follow the instructions.

Change it now and remember your new password and remember to modify your **Xshell** cnfiguration (for **Xshell** user).

# 4 What is the MNIST Classification Task?

You will learn it in class, you don't need to care about it now. : )

The solution to the task is written in python using TensorFlow as its backend, and we will try to run it on our cluster in the following sections.

Our cluster has installed python3, python2, each with TensorFlow latest. So you don't need to worry about installing them.

But, we still provide you the permission for you to be *root* (Google **sudo** command), which means you can do anything with the machine assigned to you, including installing any software.

# 5 Get the Code

For the sake of simplicity, we use TensorFlow official example.
Using

```
wget https://raw.githubusercontent.com/tensorflow/tensorflow/
    master/tensorflow/examples/tutorials/mnist/mnist_deep.py
```

to download code.

## 5.1 vim

You can view and edit it using **vim** command.

```
vim mnist_deep.py
```

You will see your screen like fig 5 (left one).

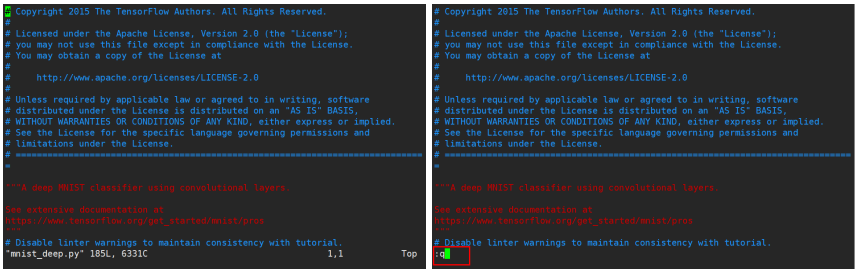


Figure 5: **vim** and how to quit it.

We won't teach you how to use **vim**, you can Google its usage. Here, we want to tell you how to quit it. (It's very important!)

Type *Esc* twice, then type *:q*, and then type *enter*, like fig 5 (right one).

## 5.2 sftp

You may feel uncomfortable with **vim**, especially when it's the first time you use it. (I won't tell you I still feel uncomfortable even after using it for years.)

Fortunately, we can download the file to local machine, edit and then upload it.

### 5.2.1 Mac and Linux user

Use **sftp** command, see fig 6.

```
sftp -oPort=PORT USERNAME@IP
```

An example,

```
sftp -oPort=22001 test@166.111.69.245
```

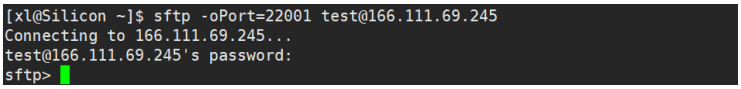


Figure 6: **sftp**

Type *help* to find usage.
But, we recommend you to use a **sftp** client such as FileZilla.

### 5.2.2 Windows User

If you use **Xshell**, remember to download **Xftp**.
If so, click a button like fig 7.

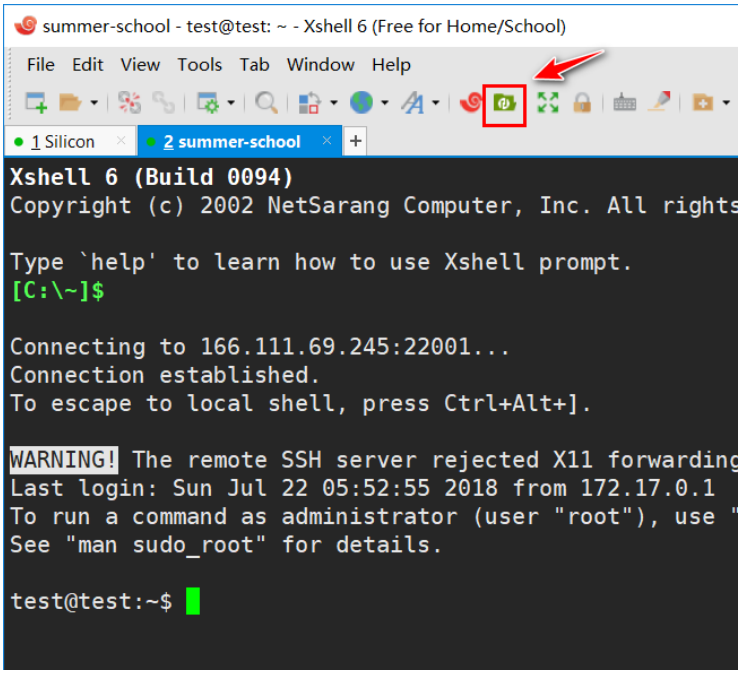


Figure 7: Open **Xftp** in **Xshell**.

It will open **Xftp** and connect automatically.

### 5.3 jupyter notebook

Another way to view and edit the python code is to use **jupyter** notebook. We have **jupyter** installed already too.

Open **jupyter** using

```
jupyter notebook --ip=0.0.0.0
```

*Attention*: DO NOT change the default port of **jupyter** notebook.

You will see message like fig 8.

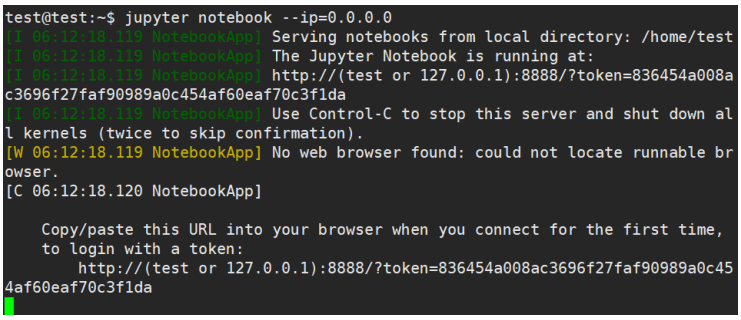


Figure 8: Run **jupyter** notebook.

Copy the last line

```
http://(test or 127.0.0.1):8888/?token=836454
    a008ac3696f27faf90989a0c454af60eaf70c3f1da
```

and replace `(test or 127.0.0.1)` to the cluster's IP `166.111.69.245`, and `8888` to **jupyter** port `23001`.

```
http://166.111.69.245:23001/?token=836454
    a008ac3696f27faf90989a0c454af60eaf70c3f1da
```

Open it in browser on your **local** machine, you will see page like fig 9.

## 6 Run!

First, we need to download dataset

```
mkdir data
cd data
wget http://yann.lecun.com/exdb/mnist/train-images-idx3-ubyte.
    gz
wget http://yann.lecun.com/exdb/mnist/train-labels-idx1-ubyte.
    gz
```

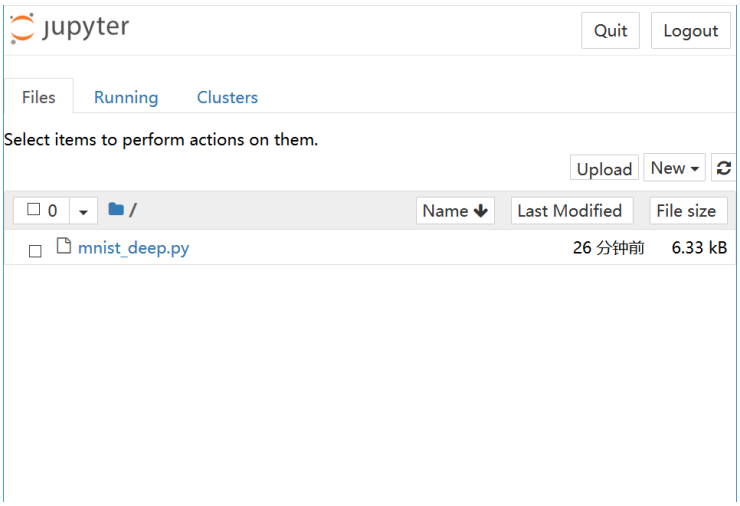


Figure 9: **jupyter** notebook.

```
wget http://yann.lecun.com/exdb/mnist/t10k-images-idx3-ubyte.gz
wget http://yann.lecun.com/exdb/mnist/t10k-labels-idx1-ubyte.gz
cd ..
```

then, run the code

```
python3 mnist_deep.py --data_dir=./data
```

or

```
python3 mnist_deep.py --data_dir=./data
```

You will see something like fig 10.

Find one line beginning with `Saving graph to:` like the one in the first red box of fig 10. Copy its directory.

And we can run TensorBoard using (Remember to replace the `logdir` using yours.)

```
tensorboard --logdir=/tmp/tmpbfchdbah
```

Open `http://(cluster's IP):(tensorbaord port)` (`http://166.111.69.245:24001` in our example) in browser on your **local** machine.

```
test@test:~$ python3 mnist_deep.py --data_dir=./data
WARNING:tensorflow:From mnist_deep.py:130: read_data_sets (from tensorflow.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a
future version.
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:260: maybe_download (from tensorflow.c
ontrib.learn.python.learn.datasets.base) is deprecated and will be removed in a future version.
Instructions for updating:
Please write your own downloading logic.
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:262: extract_images (from tensorflow.c
ontrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use tf.data to implement this functionality.
Extracting ./data/train-images-idx3-ubyte.gz
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:267: extract_labels (from tensorflow.c
ontrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use tf.data to implement this functionality.
Extracting ./data/train-labels-idx1-ubyte.gz
Extracting ./data/t10k-images-idx3-ubyte.gz
Extracting ./data/t10k-labels-idx1-ubyte.gz
WARNING:tensorflow:From /usr/local/lib/python3.5/dist-packages/tensorflow/contrib/learn/python/learn/datasets/mnist.py:290: DataSet.__init__ (from tensorflow
.contrib.learn.python.learn.datasets.mnist) is deprecated and will be removed in a future version.
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
Saving graph to: /tmp/tmpbfchdbah
2018-07-22 06:52:58.626965: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled
to use: AVX2 FMA
2018-07-22 06:52:59.566316: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1392] Found device 0 with properties:
name: Tesla P100-PCIE-16GB major: 6 minor: 0 memoryClockRate(GHz): 1.3285
pciBusID: 0000:04:00.0
totalMemory: 15.89GiB freeMemory: 15.60GiB
2018-07-22 06:52:59.566354: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1471] Adding visible gpu devices: 0
2018-07-22 06:52:59.893025: I tensorflow/core/common_runtime/gpu/gpu_device.cc:952] Device interconnect StreamExecutor with strength 1 edge matrix:
2018-07-22 06:52:59.893084: I tensorflow/core/common_runtime/gpu/gpu_device.cc:958]      0
2018-07-22 06:52:59.893093: I tensorflow/core/common_runtime/gpu/gpu_device.cc:971] 0:   N
2018-07-22 06:52:59.893544: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1084] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0
with 15123 MB memory) -> physical GPU (device: 0, name: Tesla P100-PCIE-16GB, pci bus id: 0000:04:00.0, compute capability: 6.0)
step 0, training accuracy 0.1
step 100, training accuracy 0.84
step 200, training accuracy 0.92
step 300, training accuracy 0.88
step 400, training accuracy 0.94
step 500, training accuracy 0.94
step 600, training accuracy 0.92
step 700, training accuracy 0.96
step 800, training accuracy 0.9
step 900, training accuracy 0.96
step 1000, training accuracy 0.96
```

Figure 10: Running.

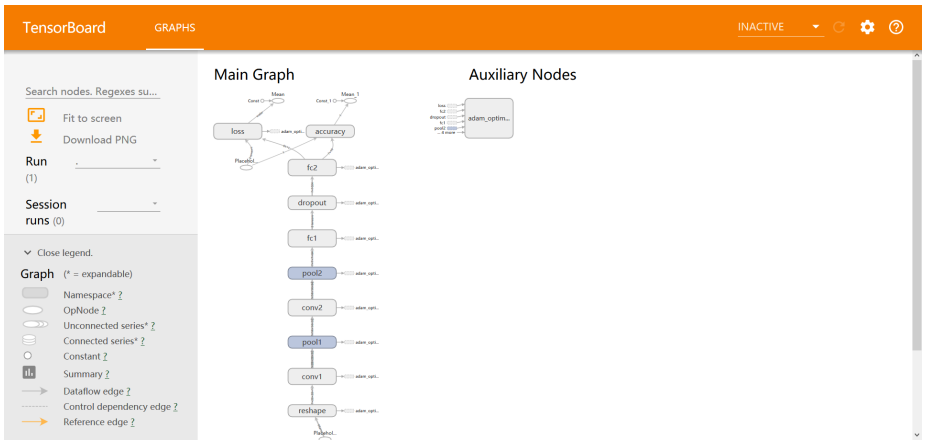


Figure 11: TensorBoard