Step by step guide on how to access NSCC supercomputer and open up jupyter notebook

This guide is done by Lim Zi Xiong and Bryan Wong Wen Ping from Singapore Polytechnic.

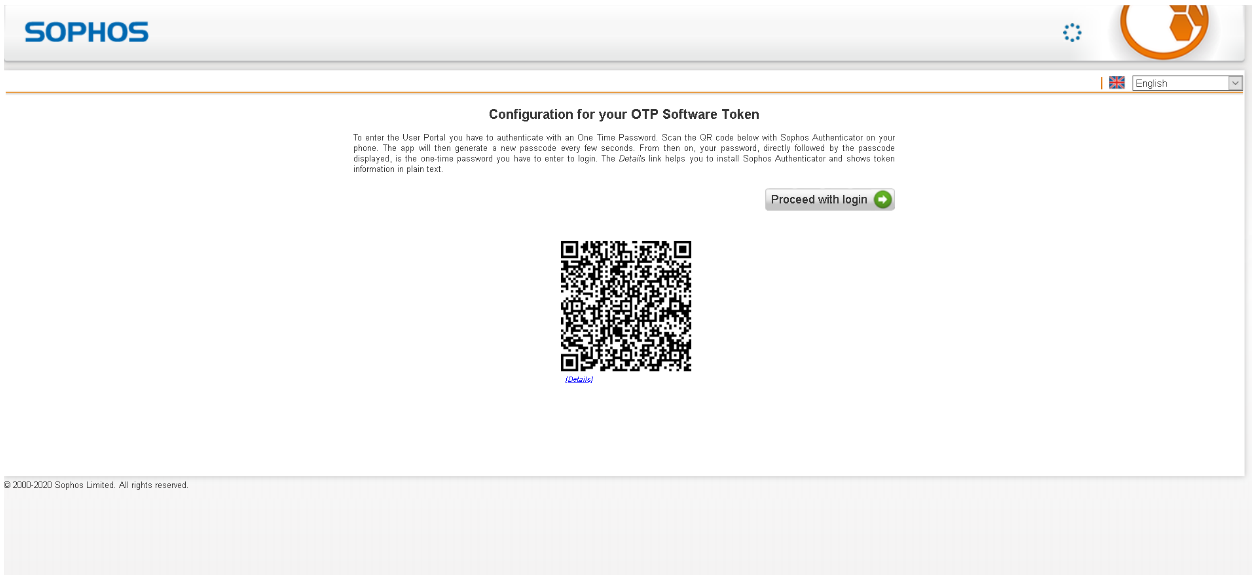
Required software

-Sophos SSL Vpn - download via NSCC website
-Sophos authenticator app - download on the app store on your mobile device.
-A terminal capable of ssh eg putty - https://www.putty.org/
-Remote file access software eg winscp - https://winscp.net/eng/download.php
-A linux environment if using windows, ubuntu for windows is available to download at windows store.

Step 1 - download the vpn and login to the vpn

-Visit https://vpn.nscc.sg/ to download the vpn client.

-When you first login using the password of your NSCC account, you will be prompted to scan the QR code. The QR code has to be scanned via the sophos vpn authenticator app so download the app from the App store.

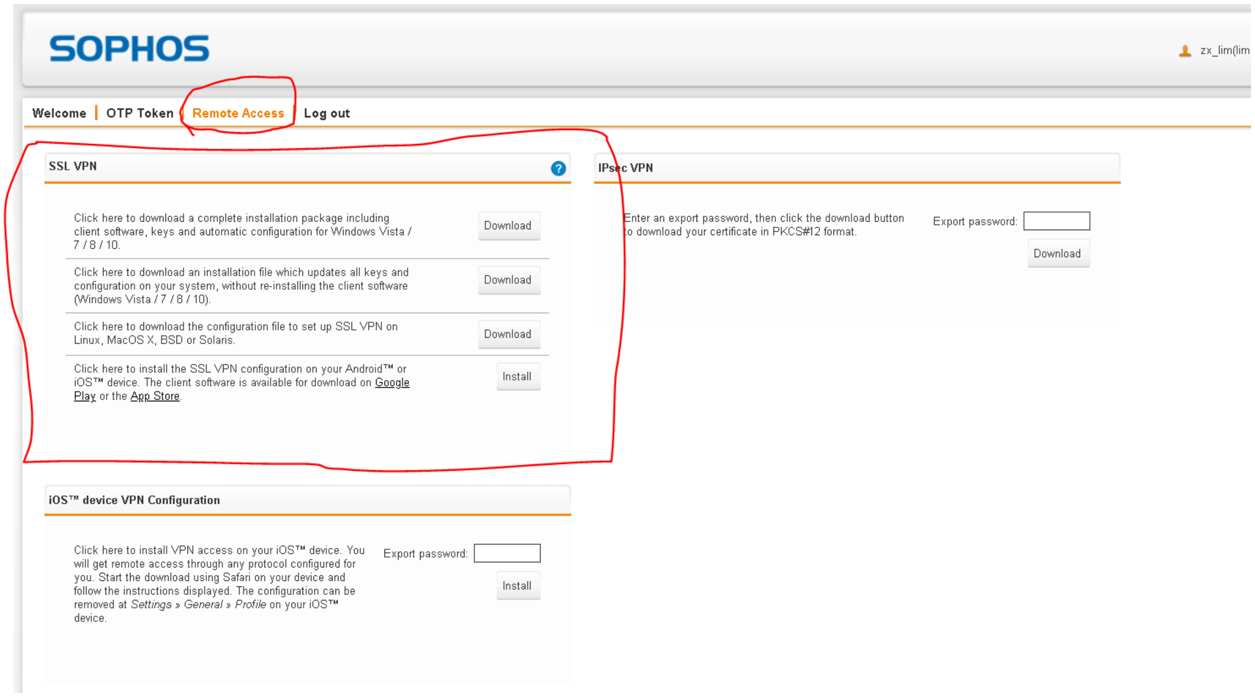


Configuration for your OTP Software Token

To enter the User Portal you have to authenticate with an One Time Password. Scan the QR code below with Sophos Authenticator on your phone. The app will then generate a new passcode every few seconds. From then on, your password, directly followed by the passcode displayed, is the one-time password you have to enter to login. The *Details* link helps you to install Sophos Authenticator and shows token information in plain text.

© 2000-2020 Sophos Limited. All rights reserved.

-Once the above step is done, you will have to log out and login again but this time in the password section you will have to include the number generated from the authenticator app. For example, my password from the NSCC is supercomputer123 and the code generated is 123456, my password will be supercomputer123123456. Once you have successfully logged in, you will see a new interface. Click on remote access on the top left corner and download the vpn client that is suitable for your OS.



Click here to download a complete installation package including client software, keys and automatic configuration for Windows Vista / 7 / 8 / 10.

Click here to download an installation file which updates all keys and configuration on your system, without re-installing the client software (Windows Vista / 7 / 8 / 10).

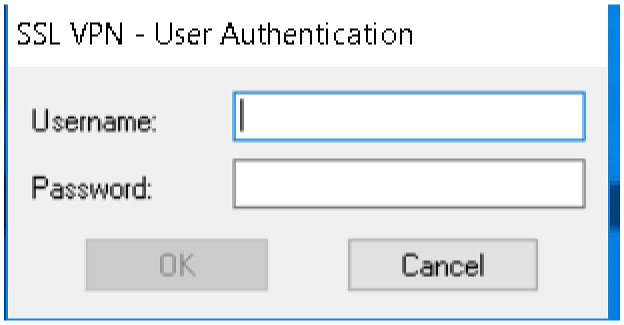
Click here to download the configuration file to set up SSL VPN on Linux, MacOS X, BSD or Solaris.

Click here to install the SSL VPN configuration on your Android™ or iOS™ device. The client software is available for download on Google Play or the App Store.

Enter an export password, then click the download button to download your certificate in PKCS#12 format.

Click here to install VPN access on your iOS™ device. You will get remote access through any protocol configured for you. Start the download using Safari on your device and follow the instructions displayed. The configuration can be removed at *Settings » General » Profile* on your iOS™ device.

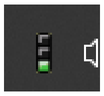
-Run the installer. Once the installer is completed, look for Sophos SSL VPN client. You will be prompted to login and the format is the same as the above step, username will be your NSCC username and the password will be the password with the code from the authenticator app.


SSL VPN - User Authentication

Username:

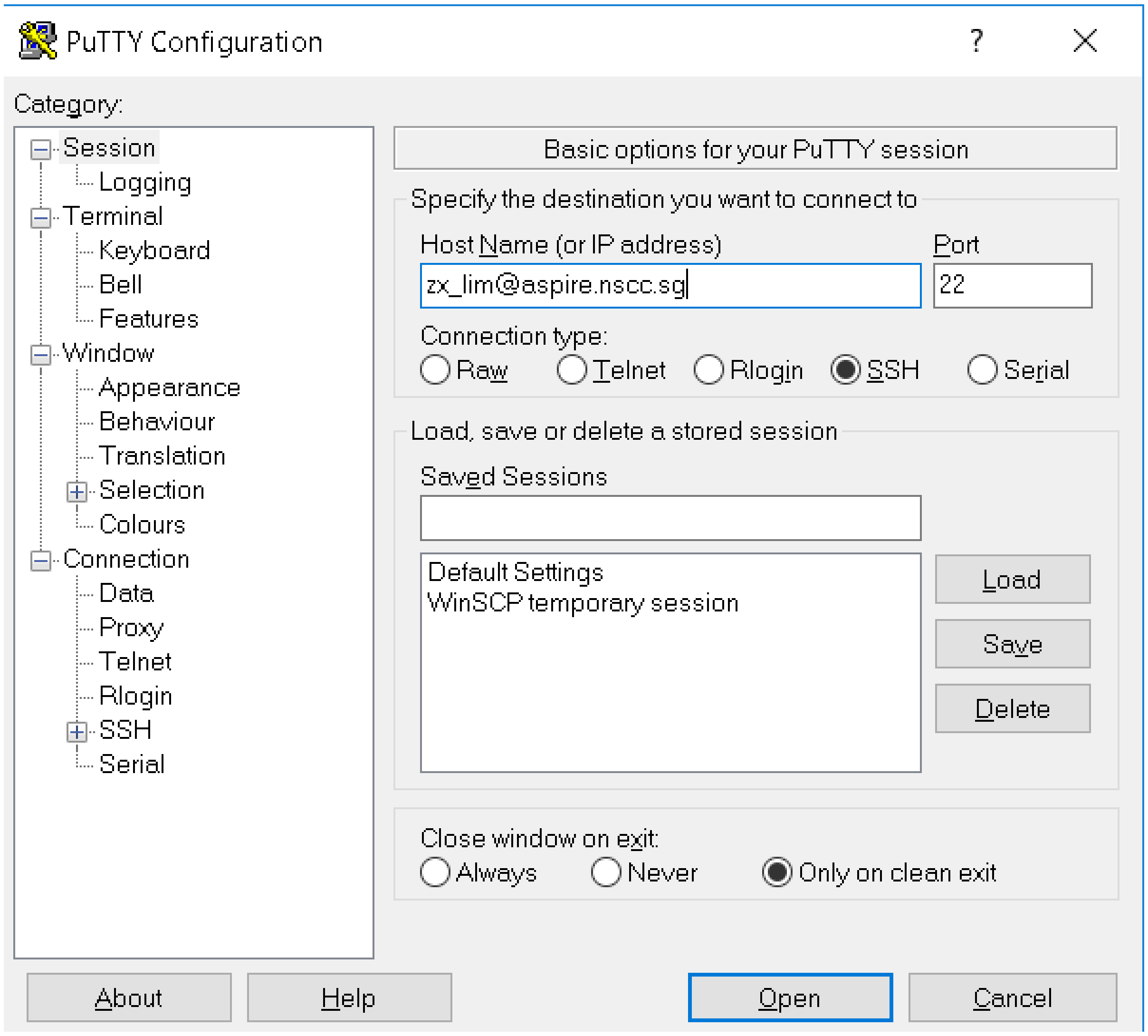
Password:

-If the login credentials are correct, the icon will turn from yellow-orange to green and this means that you have successfully connected to the NSCC vpn.



Step 2 - connecting to the nscc aspire facility via SSH

-In step 2, open up your terminal capable of SSH in this case i will be using putty.



Specify the destination you want to connect to

Host Name (or IP address): zx_lim@aspire.nscc.sg

Port: 22

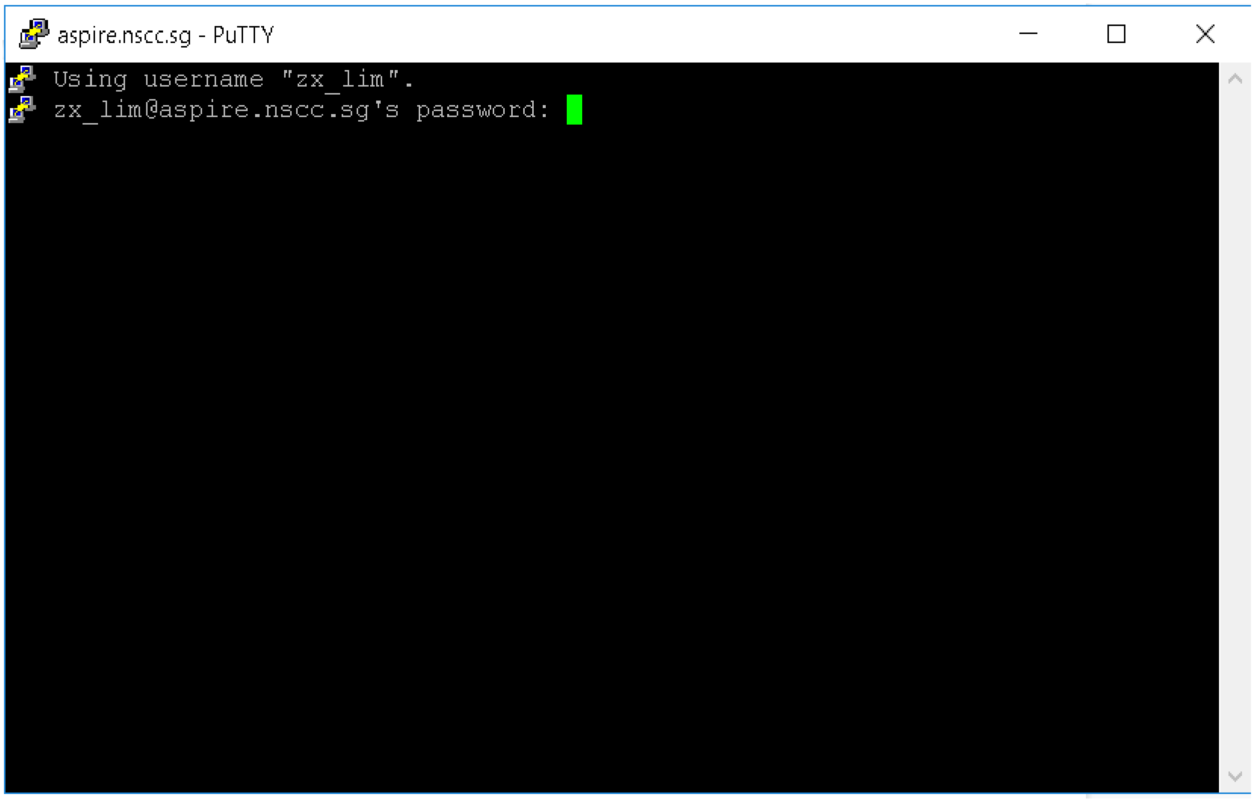
Connection type: Raw Telnet Rlogin [✓ SSH] Serial

Saved Sessions

Default Settings
WinSCP temporary session

Close window on exit: Always Never [✓ Only on clean exit]

-As you can see above, you are required to connect to a host name/address. The host name format is <NSCC username>@aspire.nscc.sg. After that click on open.



```
Using username "zx_lim".
zx_lim@aspire.nscc.sg's password:
```

-You can see from above that once you click on open, you are prompted to key in your NSCC password. The password will just be the password of your NSCC account without the authenticator code.

-Once you have successfully logged in, you will see the overview of your account eg usage the last 7 days. In this terminal we can do several things by sending a particular command. To start a new job, we use qsub, to delete a job, we use qdel, to view the status of the job, we use qstat.

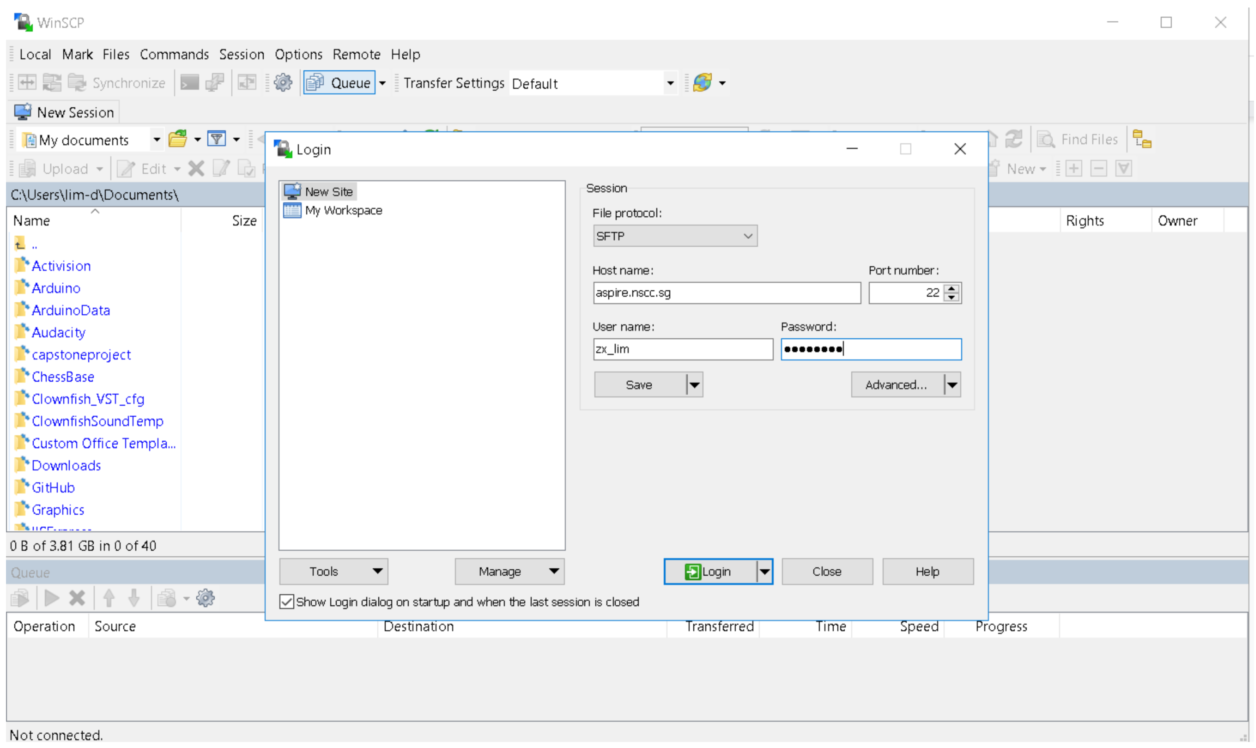
Step 2.9 - running MNIST Dataset

-NSCC uses the job management system, PBS Pro. A job is a request for processing power and commands on how to use acquired processing power.
To run a job, commands such as `qsub` are used and parameters stating required processing power are specified

-For ease, the required commands are already written in RunPythonSingularity.pbs file.

-The job script requests a GPU from nscc, starts a container that runs a python file which starts training on the MNIST Dataset.

-To run the job script, we must first transfer the job script and other files required to nscc.
To achieve this, we use WinSCP, transferring files through the SCP protocol.


WinSCP
Local Mark Files Commands Session Options Remote Help
Synchronize
Queue
Transfer Settings Default
New Session
My documents
Upload
Edit
Find Files
New
Login
New Site
My Workspace
Tools
Manage
Login
Close
Help
Save
Advanced...
Queue
Operation Source Destination Transferred Time Speed Progress

C:\Users\lim-d\Documents\

Name	Size
..	
Activision	
Arduino	
ArduinoData	
Audacity	
capstoneproject	
ChessBase	
Clownfish_VST_cfg	
ClownfishSoundTemp	
Custom Office Templa...	
Downloads	
GitHub	
Graphics	

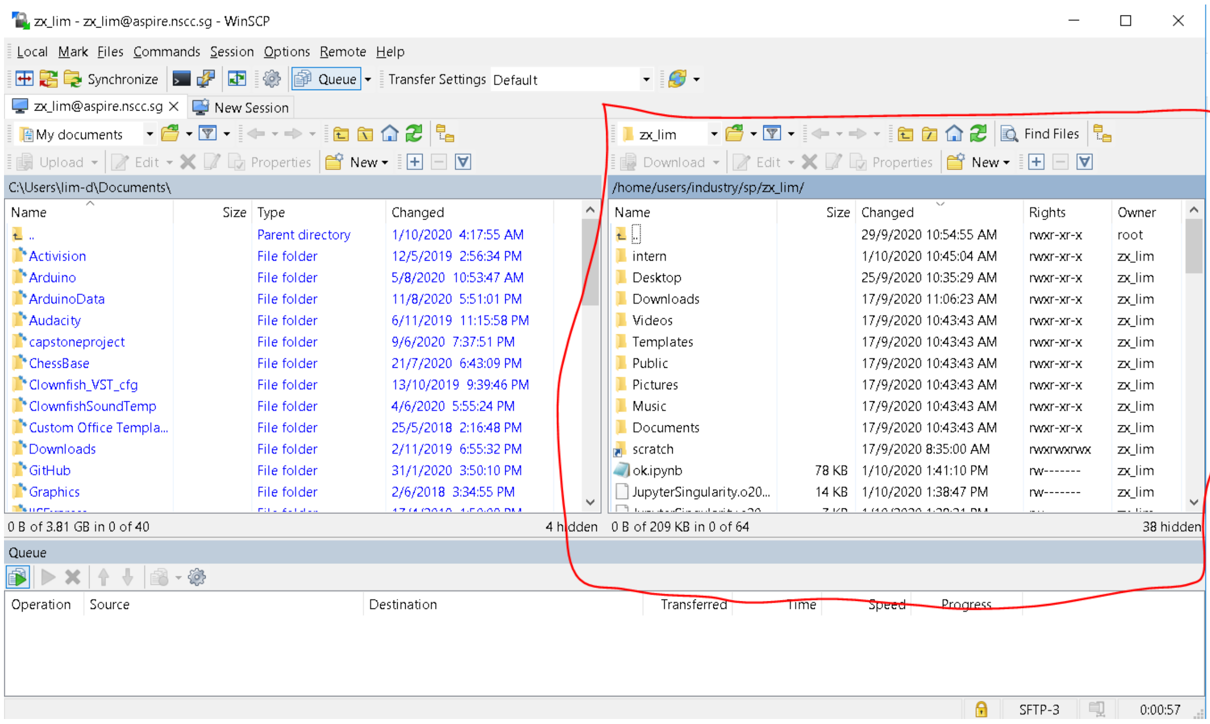
0 B of 3.81 GB in 0 of 40

Session
File protocol: SFTP
Host name: aspire.nscc.sg
Port number: 22
User name: zx_lim
Password: ••••••••

☑ Show Login dialog on startup and when the last session is closed

Not connected.

-Open a new session, and enter the following: (similar to establishing an SSH connection)
HOSTNAME: <username>@aspire.nscc.sg.
USERNAME: <your_nscc_username>
PASSWORD: <your_nscc_password>



C:\Users\lim-d\Documents\

Name	Size	Type	Changed
..		Parent directory	1/10/2020 4:17:55 AM
Activision		File folder	12/5/2019 2:56:34 PM
Arduino		File folder	5/8/2020 10:53:47 AM
ArduinoData		File folder	11/8/2020 5:51:01 PM
Audacity		File folder	6/11/2019 11:15:58 PM
capstoneproject		File folder	9/6/2020 7:37:51 PM
ChessBase		File folder	21/7/2020 6:43:09 PM
Clownfish_VST_cfg		File folder	13/10/2019 9:39:46 PM
ClownfishSoundTemp		File folder	4/6/2020 5:55:24 PM
Custom Office Templa...		File folder	25/5/2018 2:16:48 PM
Downloads		File folder	2/11/2019 6:55:32 PM
GitHub		File folder	31/1/2020 3:50:10 PM
Graphics		File folder	2/6/2018 3:34:55 PM

0 B of 3.81 GB in 0 of 40 — 4 hidden

/home/users/industry/sp/zx_lim/

Name	Size	Changed	Rights	Owner
..		29/9/2020 10:54:55 AM	rwxr-xr-x	root
intern		1/10/2020 10:45:04 AM	rwxr-xr-x	zx_lim
Desktop		25/9/2020 10:35:29 AM	rwxr-xr-x	zx_lim
Downloads		17/9/2020 11:06:23 AM	rwxr-xr-x	zx_lim
Videos		17/9/2020 10:43:43 AM	rwxr-xr-x	zx_lim
Templates		17/9/2020 10:43:43 AM	rwxr-xr-x	zx_lim
Public		17/9/2020 10:43:43 AM	rwxr-xr-x	zx_lim
Pictures		17/9/2020 10:43:43 AM	rwxr-xr-x	zx_lim
Music		17/9/2020 10:43:43 AM	rwxr-xr-x	zx_lim
Documents		17/9/2020 10:43:43 AM	rwxr-xr-x	zx_lim
scratch		17/9/2020 8:35:00 AM	rwxrwxrwx	zx_lim
ok.ipynb	78 KB	1/10/2020 1:41:10 PM	rw-------	zx_lim
JupyterSingularity.o20...	14 KB	1/10/2020 1:38:47 PM	rw-------	zx_lim

0 B of 209 KB in 0 of 64 — 38 hidden

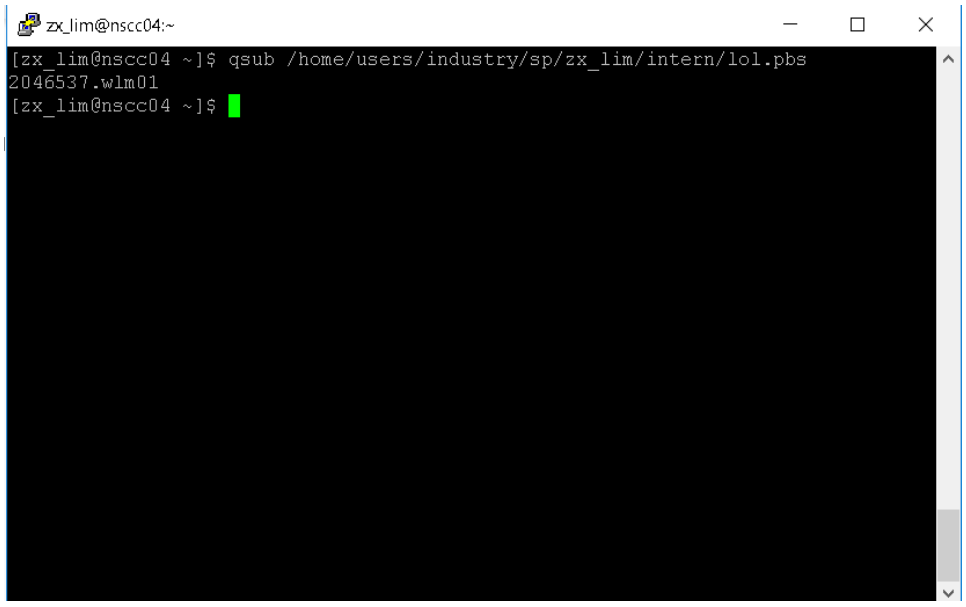
Operation	Source	Destination	Transferred	Time	Speed	Progress

-If successful, this is how your WinSCP should look.

-Copy over the script file ‘RunPythonSingularity.pbs’ and the python file ‘MNISTDataset.py’ To a folder on nscc. Note down the folder location you copied your file to.

-On your nscc ssh terminal, go to the location your RunPythonSingularity.pbs file is located. `cd <directory>` to change directory

-Start the script file by submitting a job with `qsub RunPythonSingularity.pbs`



```
[zx_lim@nscc04 ~]$ qsub /home/users/industry/sp/zx_lim/intern/lol.pbs
2046537.wlm01
[zx_lim@nscc04 ~]$
```

-You should see a 7 digit number .wlm01 and this means that your job is submitted with that ID.

-Check your job status using
` qstat -s `
S column -> Q -> job still in queue
S column -> R -> job running

-Once the job has finished running there will be 2 files.
<7 digit number>.e -> Error file
<7 digit number>.o -> Output file

-You can view the results of the training in the Output file, and any error logs in the Error file. There are example error and output files located in the 1. RunPythonSingularity folder for reference.

-Congratulations, you successfully ran MNIST dataset on nscc.

Step 3 - submitting a job script

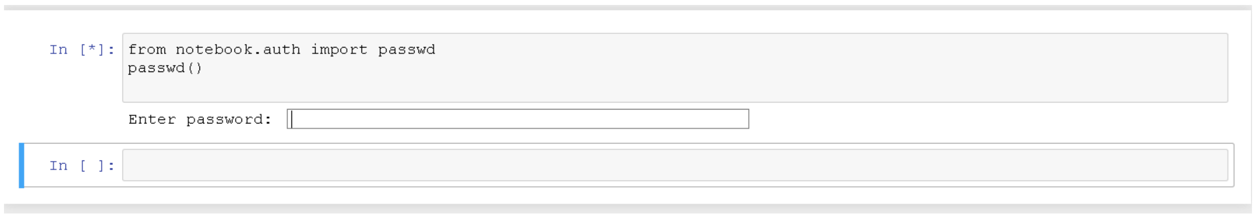
-Submitting a job requires you to use a script file in the format of SH or pbs. When submitting a job, a few things must be specified inside.

-Look for runjupytersingularity.pbs. Inside this script will have comments about what each line does so you can adjust accordingly if required.

-Before we move the script to the NSCC facility, we have to generate a hash in order to use the jupyter notebook.

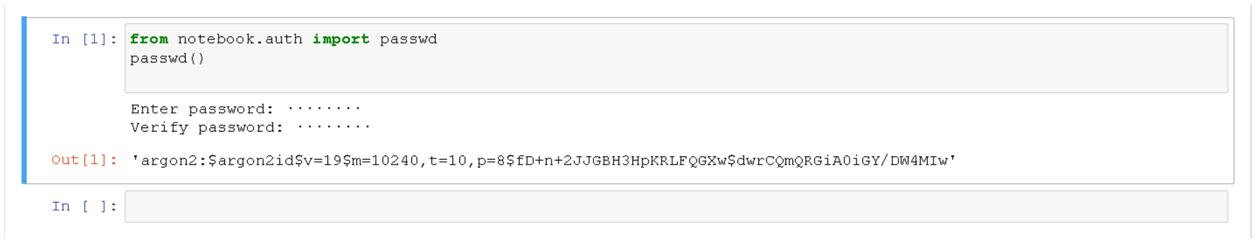
-Go into the jupyter notebook in your local computer and input this code and click on run.

from notebook.auth import passwd
passwd()



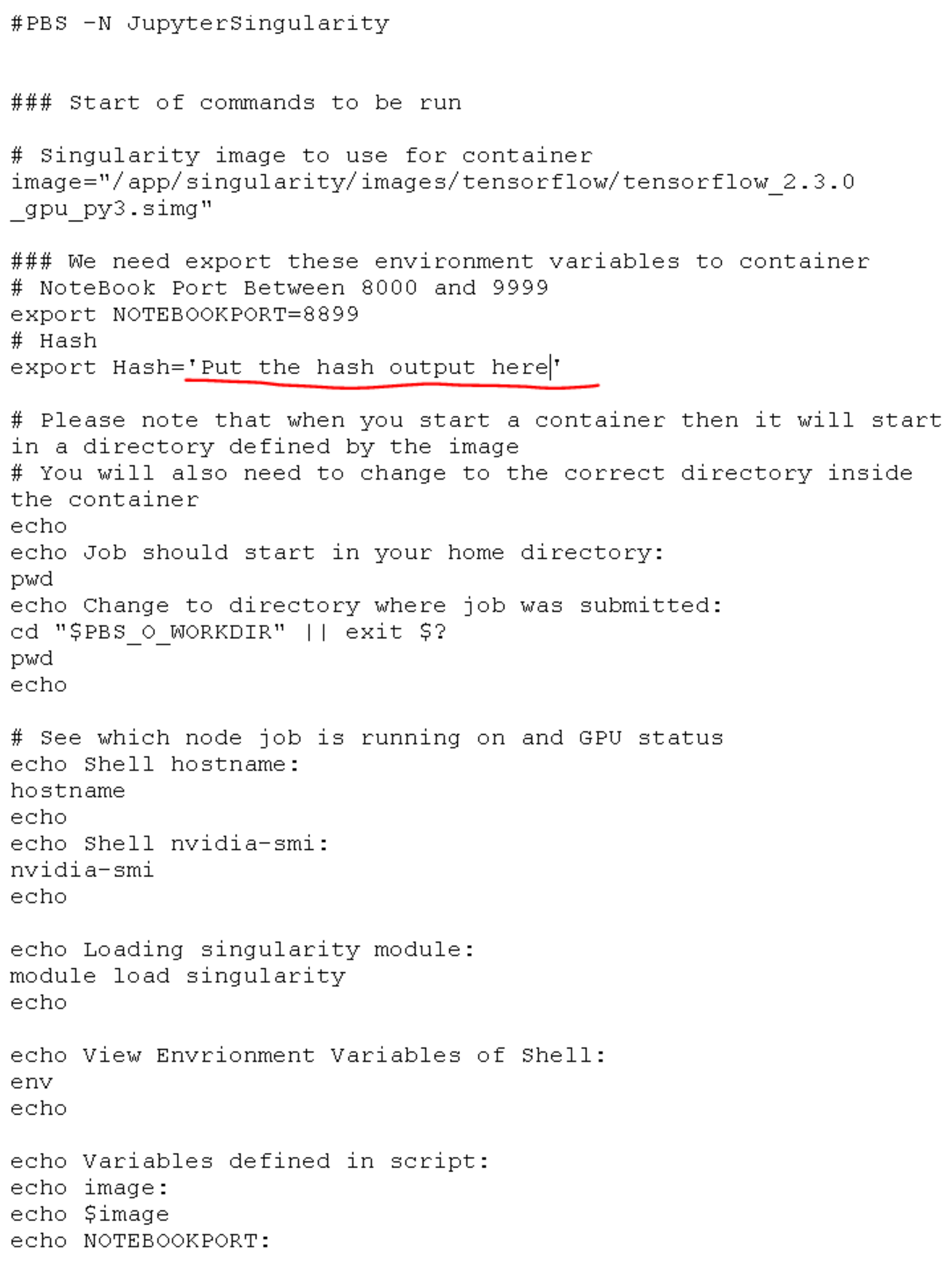
```
In [*]: from notebook.auth import passwd
        passwd()

        Enter password:

In [ ]:
```

-You will be prompted to key in a password just key in what you one but make sure you remember and note it somewhere. After entering and confirming the password, a hash will be generated.



```
In [1]: from notebook.auth import passwd
        passwd()

        Enter password: ········
        Verify password: ········

Out[1]: 'argon2:$argon2id$v=19$m=10240,t=10,p=8$fD+n+2JJGBH3HpKRLFQGXw$dwrCQmQRGiA0iGY/DW4MIw'

In [ ]:
```

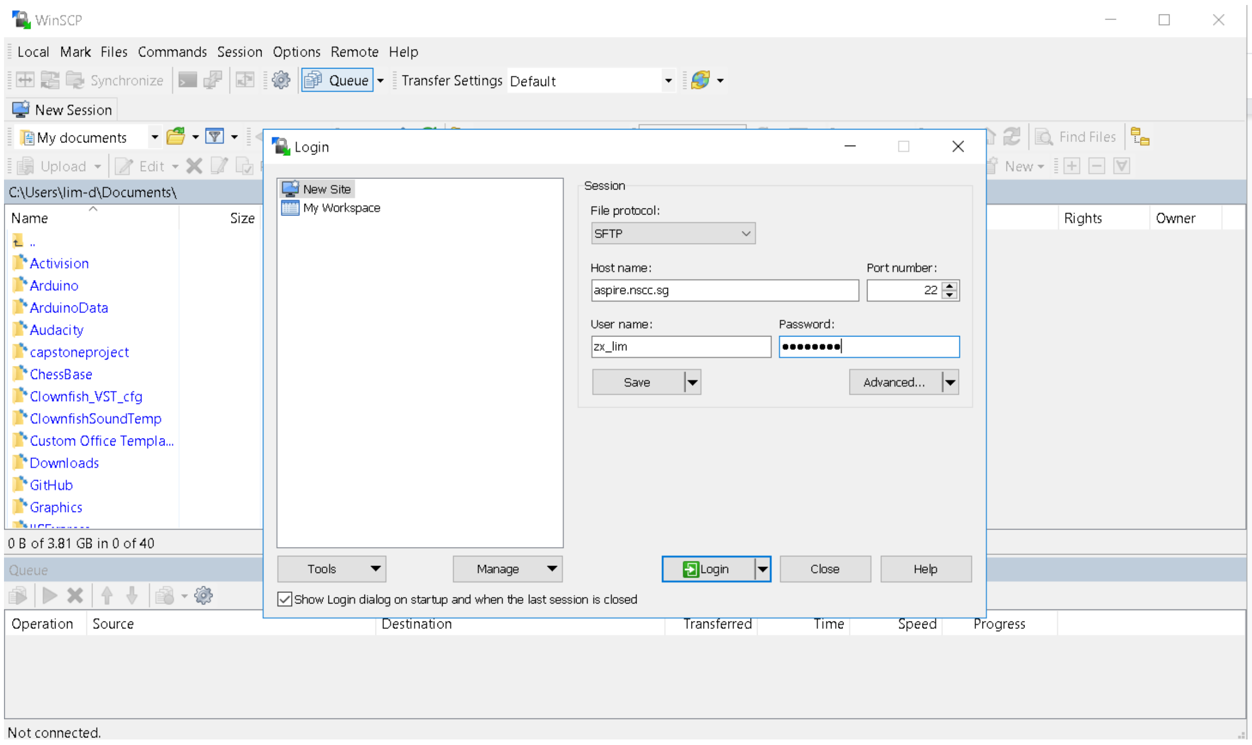
-Copy the output and paste it in the pbs script



```
#PBS -N JupyterSingularity


### Start of commands to be run

# Singularity image to use for container
image="/app/singularity/images/tensorflow/tensorflow_2.3.0
_gpu_py3.simg"

### We need export these environment variables to container
# NoteBook Port Between 8000 and 9999
export NOTEBOOKPORT=8899
# Hash
export Hash='Put the hash output here'

# Please note that when you start a container then it will start
in a directory defined by the image
# You will also need to change to the correct directory inside
the container
echo
echo Job should start in your home directory:
pwd
echo Change to directory where job was submitted:
cd "$PBS_O_WORKDIR" || exit $?
pwd
echo

# See which node job is running on and GPU status
echo Shell hostname:
hostname
echo
echo Shell nvidia-smi:
nvidia-smi
echo

echo Loading singularity module:
module load singularity
echo

echo View Envrionment Variables of Shell:
env
echo

echo Variables defined in script:
echo image:
echo $image
echo NOTEBOOKPORT:
```

-Now that we have done that, we can move the script over to NSCC facility in order to submit a job.

-In order to submit a job, we have to move the script from our local computer to the NSCC facility. To do so, we will open up winscp.



C:\Users\lim-d\Documents\

File protocol: SFTP

Host name: aspire.nscc.sg

Port number: 22

User name: zx_lim

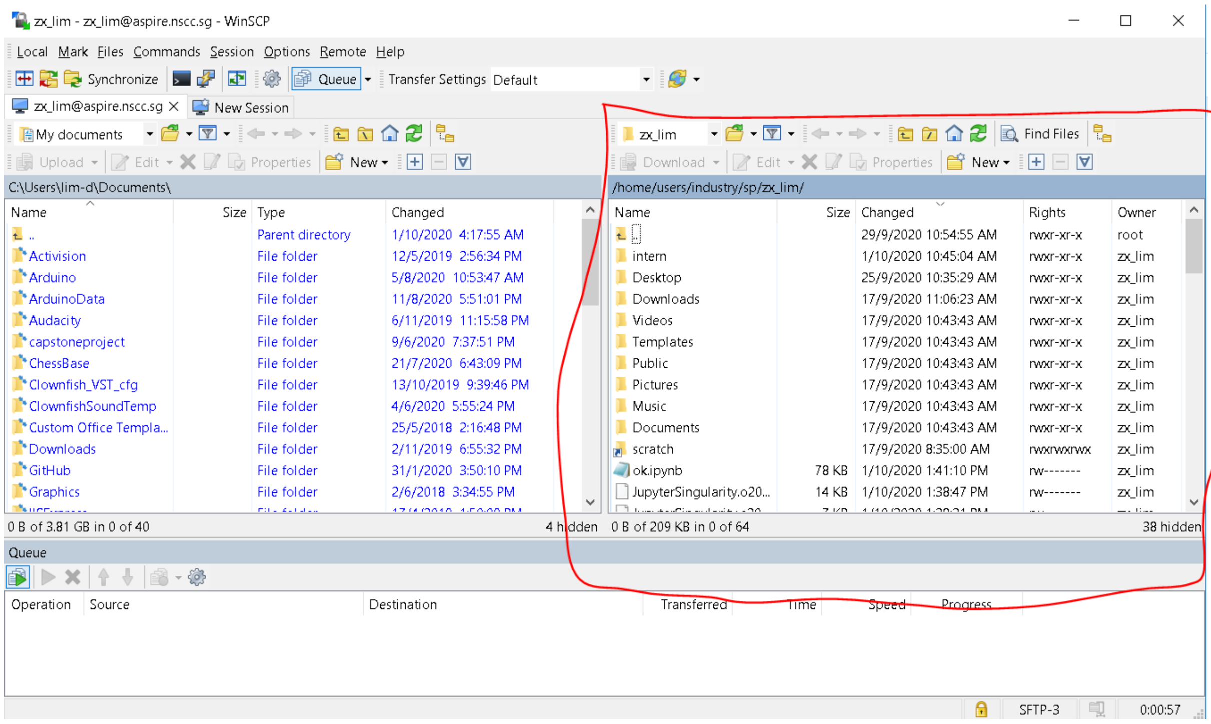
Password: ••••••••

☑ Show Login dialog on startup and when the last session is closed

0 B of 3.81 GB in 0 of 40

Not connected.

-Click on new session and you should see a new window popup prompting you to input hostname, username and password. Host name will be in the format of <username>@aspire.nscc.sg. Click on login and you should see that you have access to the files in your NSCC.


zx_lim - zx_lim@aspire.nscc.sg - WinSCP

C:\Users\lim-d\Documents\

Name	Size	Type	Changed
..		Parent directory	1/10/2020 4:17:55 AM
Activision		File folder	12/5/2019 2:56:34 PM
Arduino		File folder	5/8/2020 10:53:47 AM
ArduinoData		File folder	11/8/2020 5:51:01 PM
Audacity		File folder	6/11/2019 11:15:58 PM
capstoneproject		File folder	9/6/2020 7:37:51 PM
ChessBase		File folder	21/7/2020 6:43:09 PM
Clownfish_VST_cfg		File folder	13/10/2019 9:39:46 PM
ClownfishSoundTemp		File folder	4/6/2020 5:55:24 PM
Custom Office Templa...		File folder	25/5/2018 2:16:48 PM
Downloads		File folder	2/11/2019 6:55:32 PM
GitHub		File folder	31/1/2020 3:50:10 PM
Graphics		File folder	2/6/2018 3:34:55 PM

0 B of 3.81 GB in 0 of 40 4 hidden

/home/users/industry/sp/zx_lim/

Name	Size	Changed	Rights	Owner
..		29/9/2020 10:54:55 AM	rwxr-xr-x	root
intern		1/10/2020 10:45:04 AM	rwxr-xr-x	zx_lim
Desktop		25/9/2020 10:35:29 AM	rwxr-xr-x	zx_lim
Downloads		17/9/2020 11:06:23 AM	rwxr-xr-x	zx_lim
Videos		17/9/2020 10:43:43 AM	rwxr-xr-x	zx_lim
Templates		17/9/2020 10:43:43 AM	rwxr-xr-x	zx_lim
Public		17/9/2020 10:43:43 AM	rwxr-xr-x	zx_lim
Pictures		17/9/2020 10:43:43 AM	rwxr-xr-x	zx_lim
Music		17/9/2020 10:43:43 AM	rwxr-xr-x	zx_lim
Documents		17/9/2020 10:43:43 AM	rwxr-xr-x	zx_lim
scratch		17/9/2020 8:35:00 AM	rwxrwxrwx	zx_lim
ok.ipynb	78 KB	1/10/2020 1:41:10 PM	rw-------	zx_lim
JupyterSingularity.o20...	14 KB	1/10/2020 1:38:47 PM	rw-------	zx_lim

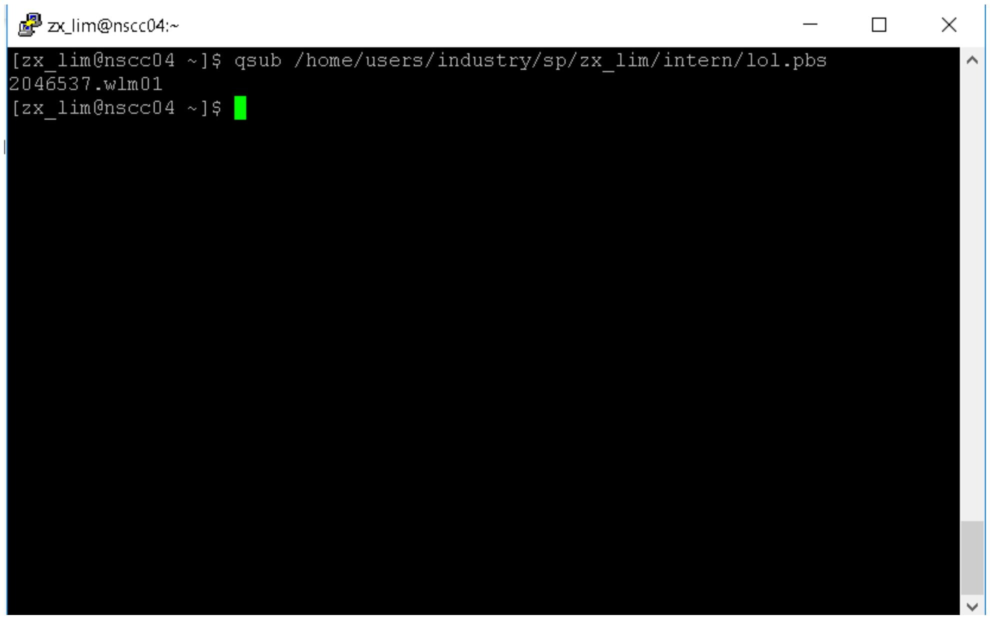
0 B of 209 KB in 0 of 64 38 hidden

SFTP-3 0:00:57

-Once we are here, move the pbs script into any directory and note down the directory. In this case I will place my script in /home/users/industry/sp/zx_lim/intern.

-Now that we have moved over the script, we can now submit a job using the script. Go to your terminal eg putty and type in the following

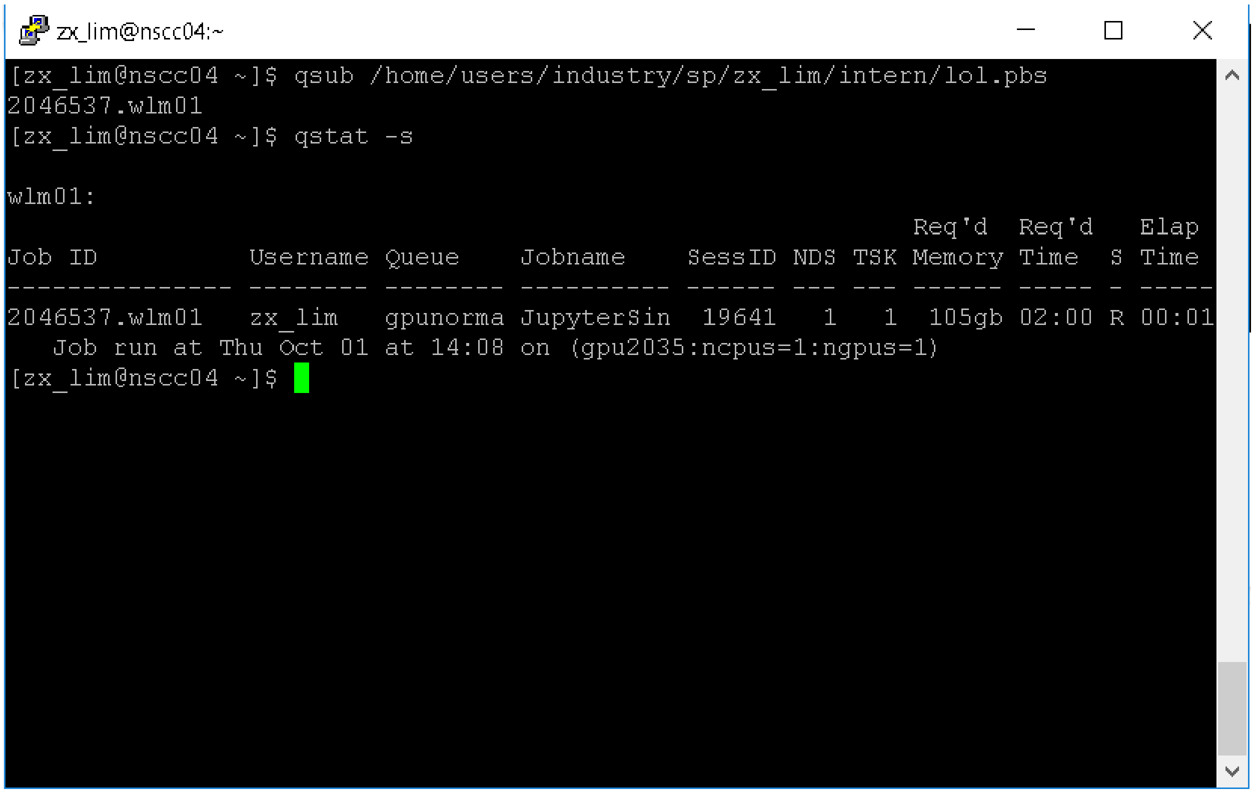
qsub /home/users/industry/sp/zx_lim/intern/nameofscript.pbs


zx_lim@nscc04:~

```
[zx_lim@nscc04 ~]$ qsub /home/users/industry/sp/zx_lim/intern/lol.pbs
2046537.wlm01
[zx_lim@nscc04 ~]$
```

-You should see a 7 digit number .wlm01 and this means that a job is submitted with that ID.

-Now that we have submitted a job, we have to track the status of the job and ensure that the job is running. The resources at NSCC are used by people all over singapore so there may be an instance where there is a shortage of CPU or GPU. To track our job status, type qstat -s on your putty terminal.



zx_lim@nscc04:~

```
[zx_lim@nscc04 ~]$ qsub /home/users/industry/sp/zx_lim/intern/lol.pbs
2046537.wlm01
[zx_lim@nscc04 ~]$ qstat -s

wlm01:
                                                            Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
2046537.wlm01   zx_lim   gpunorma JupyterSin  19641   1   1  105gb 02:00 R 00:01
   Job run at Thu Oct 01 at 14:08 on (gpu2035:ncpus=1:ngpus=1)
[zx_lim@nscc04 ~]$
```

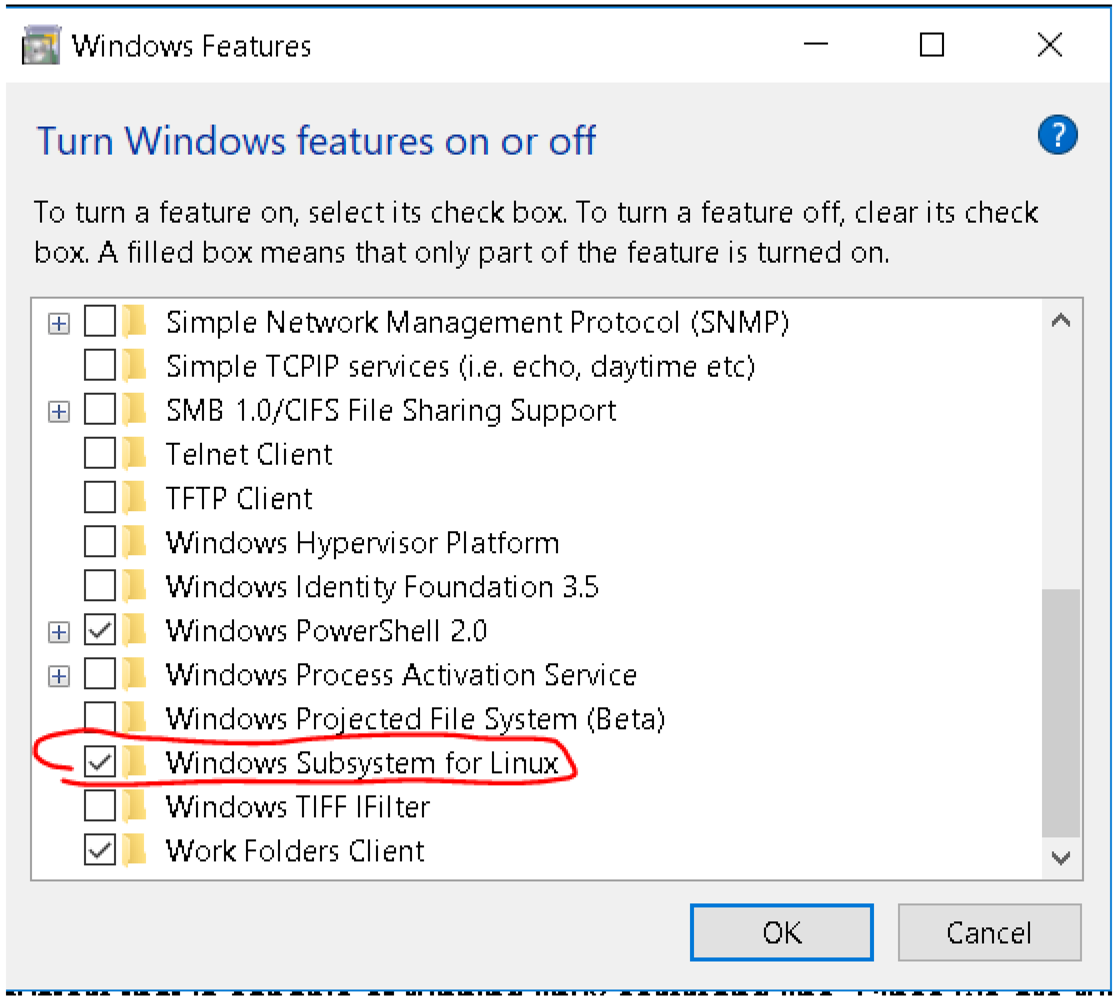
-You can see that our job is running as shown in the column ‘S’ which is good news but there may be an instance where in the ‘S’ column it shows ‘Q’, this means that the job is on queue and you will have to wait sometime before the system provision the resources you requested.

Step 3 - connecting to jupyter notebook via local computer

-In step 3, we will set up and run the jupyter notebook on our local computer but using the gpu and cpu from the NSCC to run our code.

-This step requires a system that is capable of running linux command line. Since we are running windows, we need to download ubuntu for windows in the microsoft store.

-Go to your windows start button and type in ‘Turn windows features on and off’ and look for Windows subsystem for linux. Enable it and close the window.

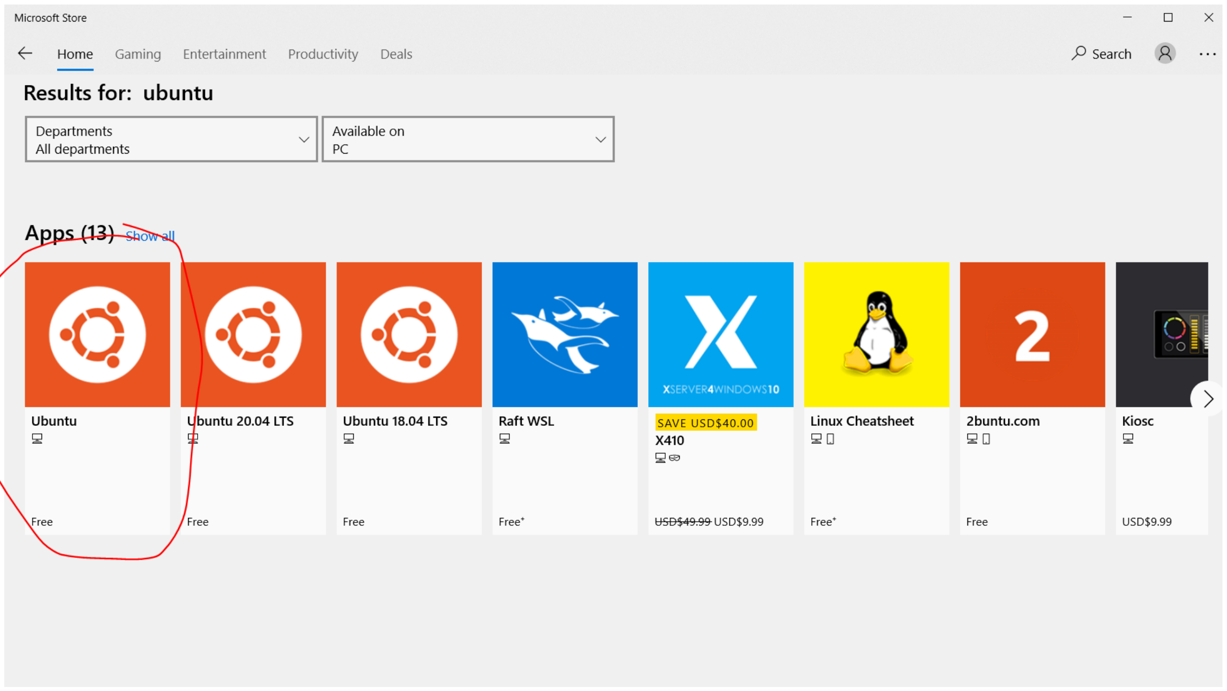


Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- [ ] Simple Network Management Protocol (SNMP)
- [ ] Simple TCPIP services (i.e. echo, daytime etc)
- [ ] SMB 1.0/CIFS File Sharing Support
- [ ] Telnet Client
- [ ] TFTP Client
- [ ] Windows Hypervisor Platform
- [ ] Windows Identity Foundation 3.5
- [✓] Windows PowerShell 2.0
- [ ] Windows Process Activation Service
- [ ] Windows Projected File System (Beta)
- [✓] Windows Subsystem for Linux
- [ ] Windows TIFF IFilter
- [✓] Work Folders Client

-You will be asked to restart but do not restart first as we have to install ubuntu for windows. On your start button type microsoft store and search for ubuntu. Install it and restart your computer.



Results for: ubuntu

Departments: All departments

Available on: PC

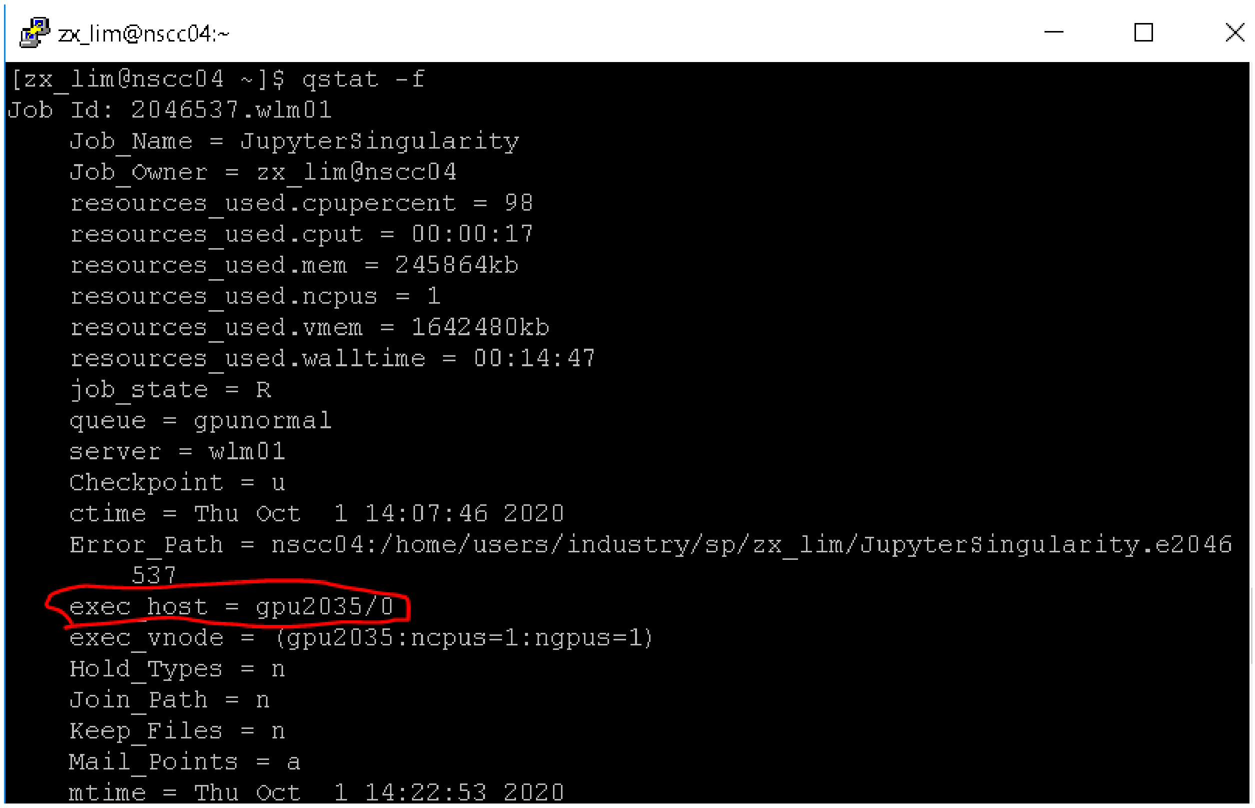
Apps (13) Show all

Ubuntu	Ubuntu 20.04 LTS	Ubuntu 18.04 LTS	Raft WSL	X410	Linux Cheatsheet	2buntu.com	Kiosc
Free	Free	Free	Free*	SAVE USD$40.00 ~~USD$49.99~~ USD$9.99	Free*	Free	USD$9.99

-Now that we have successfully installed ubuntu on our windows pc, we can proceed to establishing a tunnel connection from our local pc to NSCC facilities. In this step, open up your notepad++ and name it tunnel.sh. Inside, paste the content below to your notepad.

```
ssh -N -L localhost:<LOCAL_HOST_PORT>:<HOST_NAME>-ib0:<NOTEBOOK_PORT>
<USER_NAME>@aspire.nscc.sg
```

-Since in the above PBS script we specified that we want to run the jupyter notebook on port 8899 and we want to use 8899 on our local computer as well, we will key in 8899 as our local host port and notebook port. For the hostname, go to putty and type qstat -f <id of job> and look for ‘exec_host’.

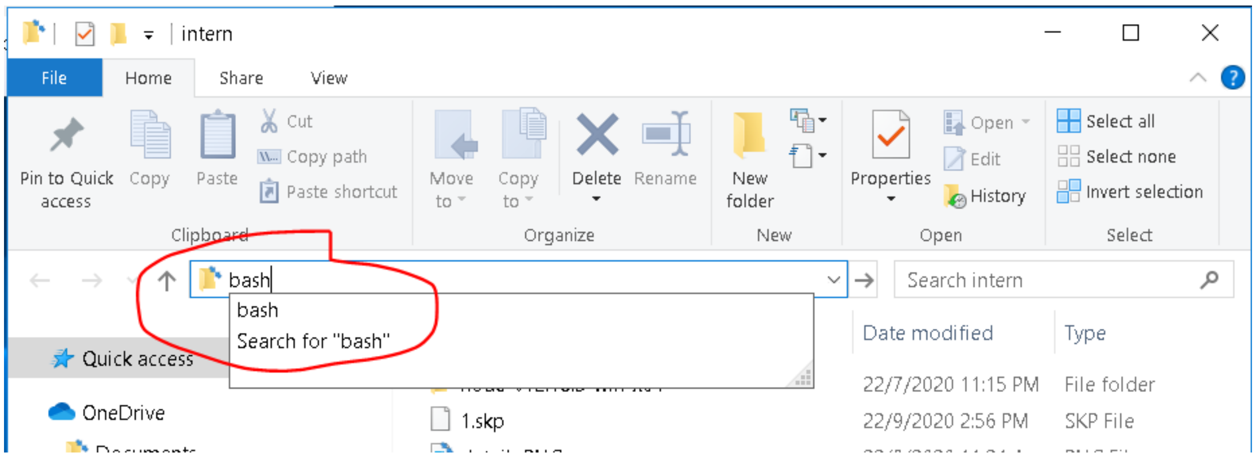


```
[zx_lim@nscc04 ~]$ qstat -f
Job Id: 2046537.wlm01
    Job_Name = JupyterSingularity
    Job_Owner = zx_lim@nscc04
    resources_used.cpupercent = 98
    resources_used.cput = 00:00:17
    resources_used.mem = 245864kb
    resources_used.ncpus = 1
    resources_used.vmem = 1642480kb
    resources_used.walltime = 00:14:47
    job_state = R
    queue = gpunormal
    server = wlm01
    Checkpoint = u
    ctime = Thu Oct  1 14:07:46 2020
    Error_Path = nscc04:/home/users/industry/sp/zx_lim/JupyterSingularity.e2046
        537
    exec_host = gpu2035/0
    exec_vnode = (gpu2035:ncpus=1:ngpus=1)
    Hold_Types = n
    Join_Path = n
    Keep_Files = n
    Mail_Points = a
    mtime = Thu Oct  1 14:22:53 2020
```

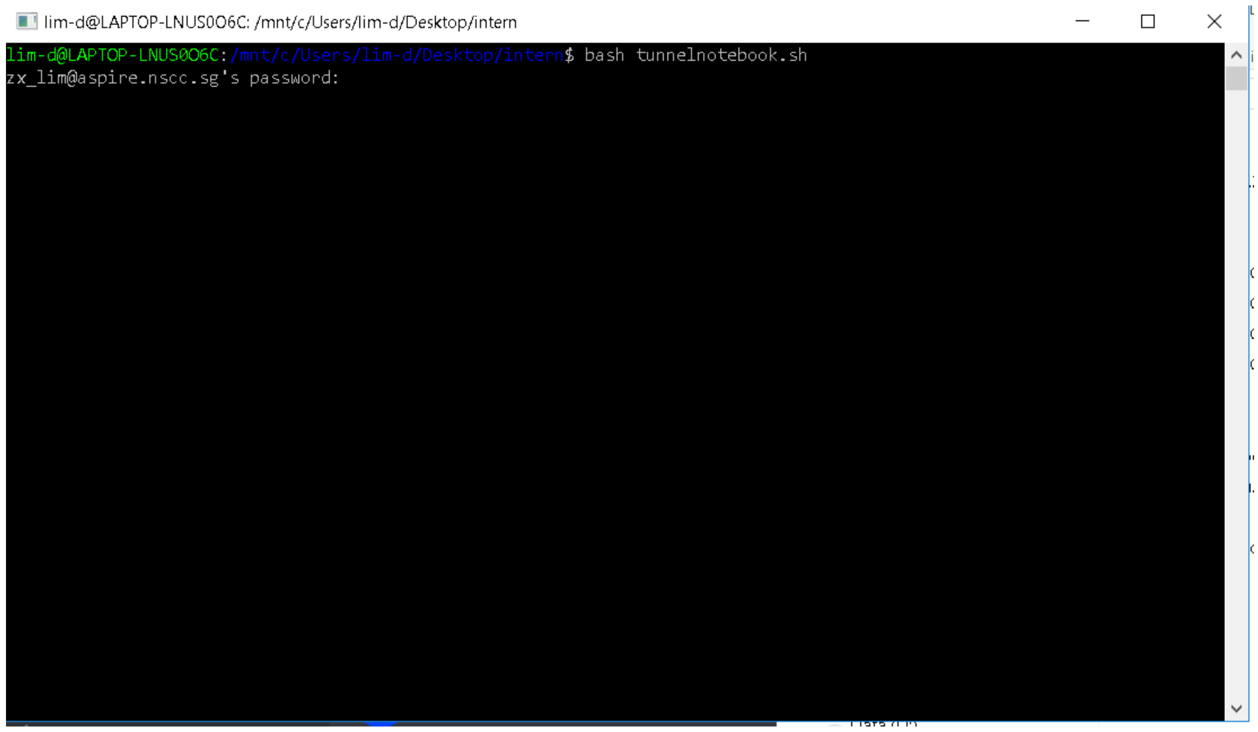
-We will ignore the /0 so our host name will be gpu2035. In this case, my script content will look like this

ssh -N -L localhost:8899:gpu2035-ib0:8899 zx_lim@aspire.nscc.sg

-Save the file in a folder. Now we have to run the script to establish the connection. Go to the folder and type bash.

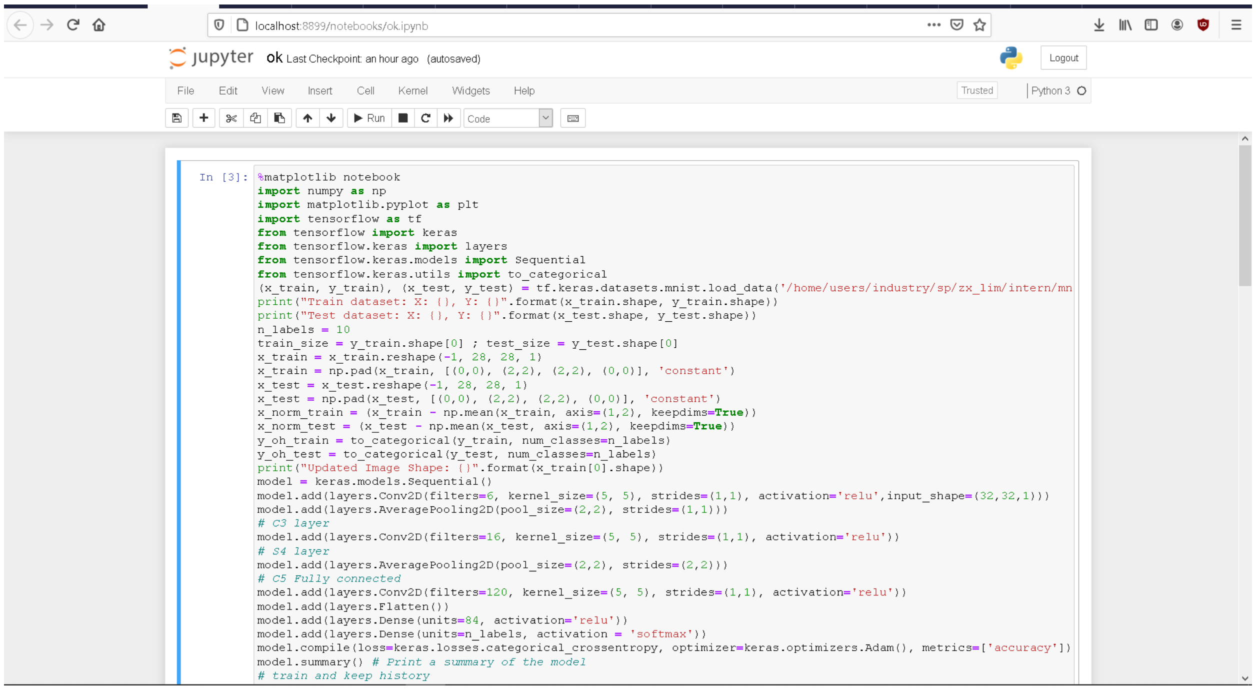


-A new window will popup. Type in ‘bash tunnel.sh’. You will be prompted for a password and the password will be your NSCC account password. Now that we have established a connection between NSCC and your local pc, it is time to test out if the connection is successful. Go to your browser and type localhost:8899. You should see jupyter notebook in the browser. You may be prompted to key in a password in order to access the notebook, the password will be the one that you have set on the earlier step in your local jupyter notebook.



```
lim-d@LAPTOP-LNUS0O6C:/mnt/c/Users/lim-d/Desktop/intern$ bash tunnelnotebook.sh
zx_lim@aspire.nscc.sg's password:
```

-If all is successful, you should be able to run the code in your jupyter notebook.



```
%matplotlib notebook
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
from tensorflow.keras.utils import to_categorical
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.mnist.load_data('/home/users/industry/sp/zx_lim/intern/mn
print("Train dataset: X: {}, Y: {}".format(x_train.shape, y_train.shape))
print("Test dataset: X: {}, Y: {}".format(x_test.shape, y_test.shape))
n_labels = 10
train_size = y_train.shape[0] ; test_size = y_test.shape[0]
x_train = x_train.reshape(-1, 28, 28, 1)
x_train = np.pad(x_train, [(0,0), (2,2), (2,2), (0,0)], 'constant')
x_test = x_test.reshape(-1, 28, 28, 1)
x_test = np.pad(x_test, [(0,0), (2,2), (2,2), (0,0)], 'constant')
x_norm_train = (x_train - np.mean(x_train, axis=(1,2), keepdims=True))
x_norm_test = (x_test - np.mean(x_test, axis=(1,2), keepdims=True))
y_oh_train = to_categorical(y_train, num_classes=n_labels)
y_oh_test = to_categorical(y_test, num_classes=n_labels)
print("Updated Image Shape: {}".format(x_train[0].shape))
model = keras.models.Sequential()
model.add(layers.Conv2D(filters=6, kernel_size=(5, 5), strides=(1,1), activation='relu',input_shape=(32,32,1)))
model.add(layers.AveragePooling2D(pool_size=(2,2), strides=(1,1)))
# C3 layer
model.add(layers.Conv2D(filters=16, kernel_size=(5, 5), strides=(1,1), activation='relu'))
# S4 layer
model.add(layers.AveragePooling2D(pool_size=(2,2), strides=(2,2)))
# C5 Fully connected
model.add(layers.Conv2D(filters=120, kernel_size=(5, 5), strides=(1,1), activation='relu'))
model.add(layers.Flatten())
model.add(layers.Dense(units=84, activation='relu'))
model.add(layers.Dense(units=n_labels, activation = 'softmax'))
model.compile(loss=keras.losses.categorical_crossentropy, optimizer=keras.optimizers.Adam(), metrics=['accuracy'])
model.summary() # Print a summary of the model
# train and keep history
```