

AI Deep Learning with TensorFlow on Google Cloud Platform (GCP)

Start and Use Jupyter Notebook in Deep Learning VM

Thuan L Nguyen, Ph.D.

1. GCP Virtual Machine Based on GCP Deep Learning Image

Google Cloud Platform now provides machine learning images designed for deep learning practitioners. The users can conveniently set up GCP deep learning virtual machine based on these images.

The Google Deep Learning images are a set of prepackaged VM images with a deep learning framework ready to be run out of the box. Currently, there are images supporting TensorFlow, PyTorch, and generic high-performance computing, with versions for both CPU-only and GPU-enabled workflows.

All the GCP deep learning images are based on [Debian](#), a Linux OS version.

All images come with python 2.7/3.5 with pre-installed core packages:

- numpy
- sklearn
- scipy
- pandas

Additionally, [Jupyter Notebook](#) is also a built-in core package available in the images. It means that the user can run Jupyter Notebook server in the remote VM and connect to it from the local computer.

To start and use Jupyter Notebook to train deep learning models in GCP virtual machines, the user needs to do three steps.

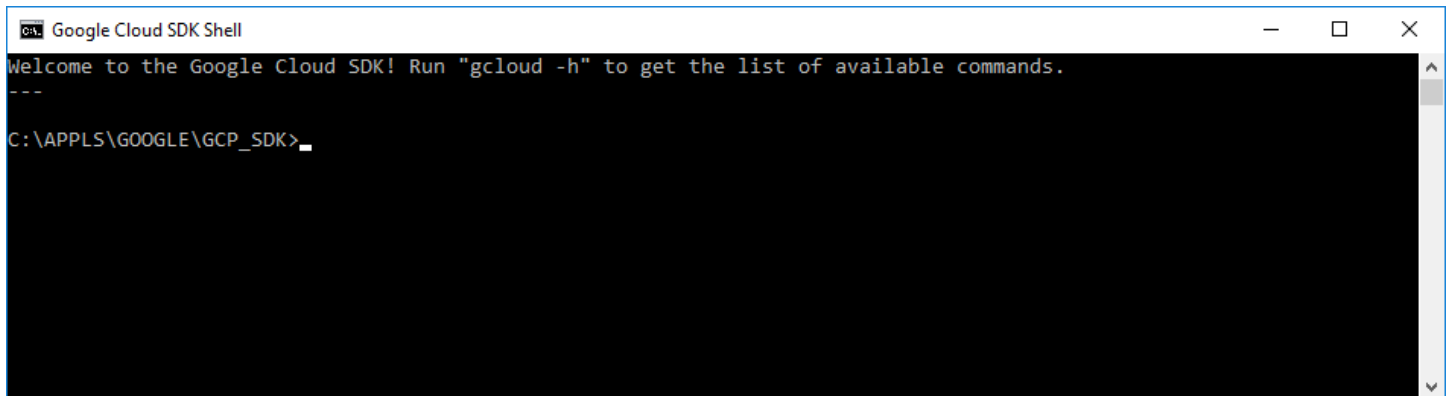
IMPORTANT NOTES:

--> It is assumed that the GCP deep learning virtual machine (VM) has been set up successfully as discussed in the document "HOWTO_GCP_setup_deep_learning_vm.pdf."

--> It is assumed that the GCloud SDK has been successfully installed in the user's local computer.

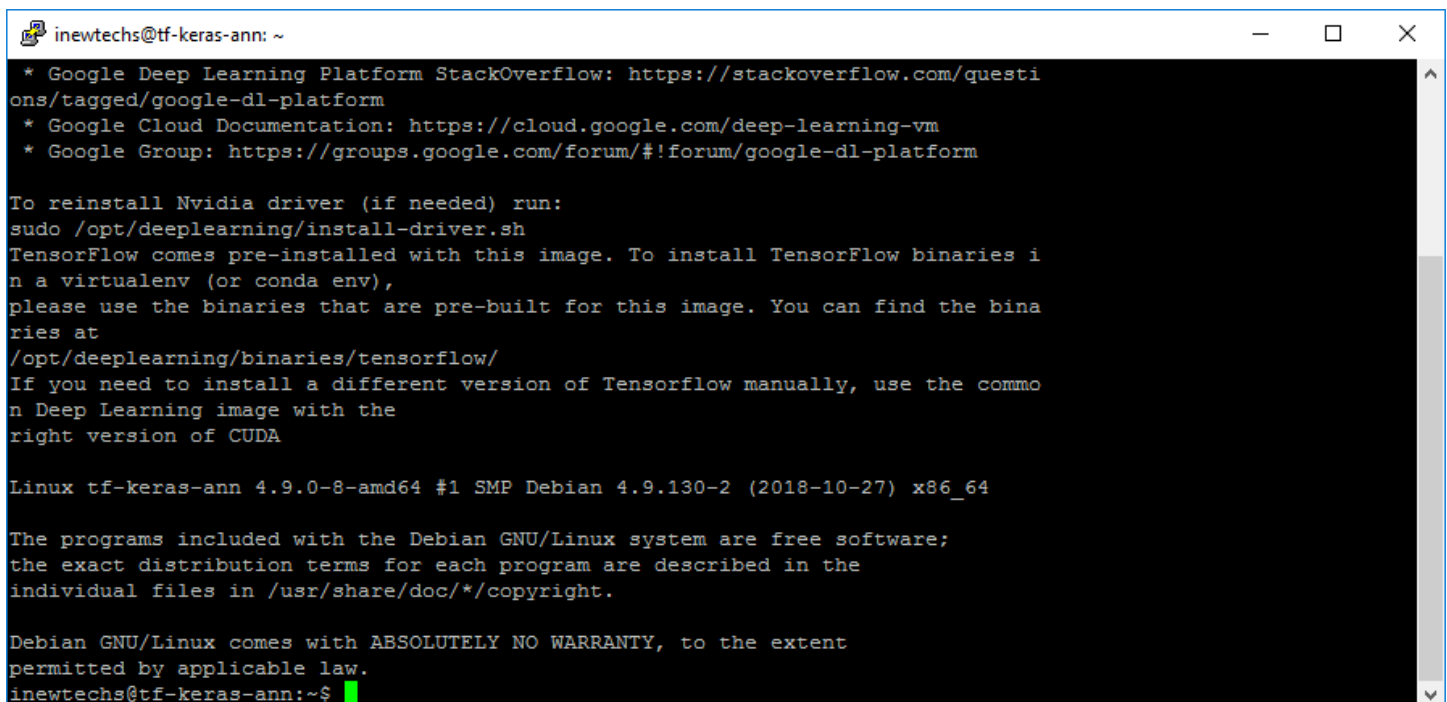
2. Step 1: Start Jupyter Notebook Server in GCP Remote VM

- Start a GCloud SDK terminal window (GCloud Terminal #1)



A screenshot of a terminal window titled "Google Cloud SDK Shell". The window has a black background with white text. The text inside reads: "Welcome to the Google Cloud SDK! Run 'gcloud -h' to get the list of available commands." followed by a separator line "----". Below that, the prompt "C:\APPLS\GOOGLE\GCP_SDK>" is visible with a cursor.

- SSH to the remote deep learning virtual machine
 - A SSH terminal window pops up
 - **NOTES:** This is SSH TERMINAL WINDOW #1



A screenshot of an SSH terminal window titled "inewtechs@tf-keras-ann: ~". The window has a black background with white text. The text inside includes several links: "Google Deep Learning Platform StackOverflow: https://stackoverflow.com/questions/tagged/google-dl-platform", "Google Cloud Documentation: https://cloud.google.com/deep-learning-vm", and "Google Group: https://groups.google.com/forum/#!forum/google-dl-platform". It then provides instructions for reinstalling the Nvidia driver and installing TensorFlow binaries. The prompt "inewtechs@tf-keras-ann:~\$" is visible at the bottom with a green cursor.

- **IMPORTANT NOTES:**
 - IF THIS IS THE FIRST TIME TO START JUPYTER NOTEBOOK IN THE VM
 - MAKE A NEW SUB FOLDER to store all jupyter notebooks documents
 - Run this command line inside the home directory
 - \$ **mkdir** <new folder name>

It is assumed that the new folder name is “**JP_NTBK**”

- Change the current directory to the newly created sub-folder
 - Run this command line:
 - `$ cd <new folder name>`
- Inside the **SSH terminal window #1**, START Jupyter Notebook server with the command line
 - It means that Jupyter Notebook server starts inside the new folder.
 - Run this command line:
 - `$ jupyter notebook --port=8888`

IMPORTANT NOTES:

--> Jupyter Notebook servers starts at the port 8888 by default. Therefore, if the port is not specified, 99.999% the server starts at this port. However, we would like to be sure that if the port was used for some other service, an error would occur, and the issue could be addressed accordingly.

```
inewtechs@tf-keras-ann: ~
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
inewtechs@tf-keras-ann:~$
inewtechs@tf-keras-ann:~$
inewtechs@tf-keras-ann:~$ jupyter notebook
[I 17:28:49.577 NotebookApp] Writing notebook server cookie secret to /run/user/1001/jupyter/notebook_cookie_secret
[I 17:28:51.731 NotebookApp] jupyter_tensorboard extension loaded.
[I 17:28:51.768 NotebookApp] JupyterLab extension loaded from /usr/local/lib/python3.5/dist-packages/jupyterlab
[I 17:28:51.768 NotebookApp] JupyterLab application directory is /usr/local/share/jupyter/lab
[I 17:28:51.770 NotebookApp] Serving notebooks from local directory: /home/inewtechs
[I 17:28:51.770 NotebookApp] The Jupyter Notebook is running at:
[I 17:28:51.770 NotebookApp] http://localhost:8888/?token=36d55ba0642017b941e81a69be6a6ae274e0665a1b3fed15
[I 17:28:51.770 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

[W 17:28:51.771 NotebookApp] No web browser found: could not locate runnable browser.
[C 17:28:51.771 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8888/?token=36d55ba0642017b941e81a69be6a6ae274e0665a1b3fed15
```

```
inewtechs@tf-keras-ann: ~
individual files in /usr/share/doc/*/copyright.

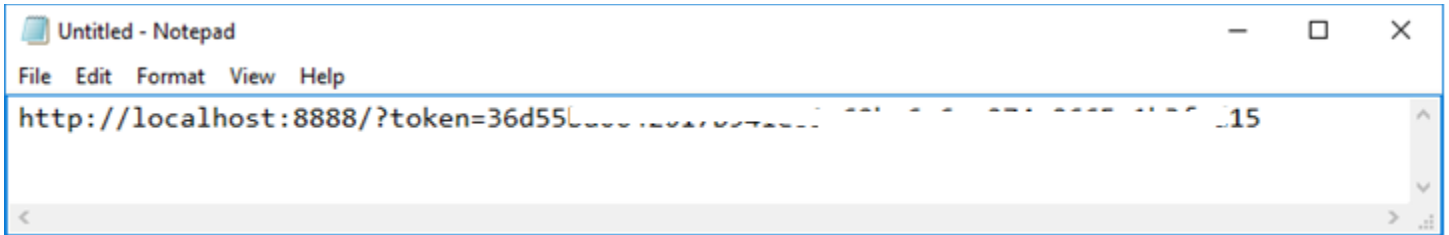
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
inewtechs@tf-keras-ann:~$
inewtechs@tf-keras-ann:~$
inewtechs@tf-keras-ann:~$ jupyter notebook
[I 17:28:49.577 NotebookApp] Writing notebook server cookie secret to /run/user/1001/jupyter/notebook_cookie_secret
[I 17:28:51.731 NotebookApp] jupyter_tensorboard extension loaded.
[I 17:28:51.768 NotebookApp] JupyterLab extension loaded from /usr/local/lib/python3.5/dist-packages/jupyterlab
[I 17:28:51.768 NotebookApp] JupyterLab application directory is /usr/local/share/jupyter/lab
[I 17:28:51.770 NotebookApp] Serving notebooks from local directory: /home/inewtechs
[I 17:28:51.770 NotebookApp] The Jupyter Notebook is running at:
[I 17:28:51.770 NotebookApp] http://localhost:8888/?token=36d55ba0642017b941e81a69be6a6ae274e0665a1b3fed15
[I 17:28:51.770 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

[W 17:28:51.771 NotebookApp] No web browser found: could not locate runnable browser.
[C 17:28:51.771 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8888/?token=36d55ba0642017b941e81a69be6a6ae274e0665a1b3fed15
```

IMPORTANT NOTES:

--> Copy this line by high-light it then paste it into a Notepad file (Windows) or any pure-text editor in any other OS platform.



--> **Pay attention to the details of this line:**

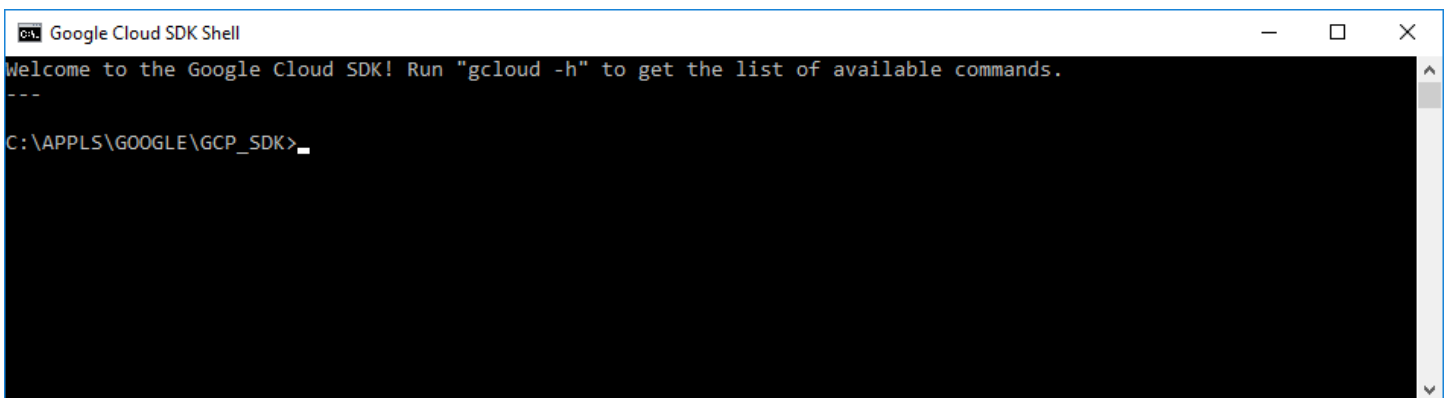
- This Jupyter Notebook server starts at the remote virtual machine (localhost: in the view of the remote VM).
- The server starts at the port 8888 by default.
- A token is issued for this Jupyter Notebook server.
- The token will be used for the authentication of connections to the remote Jupyter Notebook server.

IMPORTANT NOTES:

--> LEAVE THIS SSH TERMINAL WINDOW AS IS: DON'T CLOSE IT

3. Step 2: Forward a Local Port to the Port 8888 in GCP Remote VM

- Start a GCLOUD SDK terminal window (GCLOUD Terminal #2)



- In the GCLOUD terminal window #2, SSH to the remote deep learning virtual machine
 - Run this command line:

```
$ gcloud compute ssh USER@Instance_Name  
--project [PROJECT_ID]  
--zone [ZONE]  
-- -L 8000:localhost:8888
```

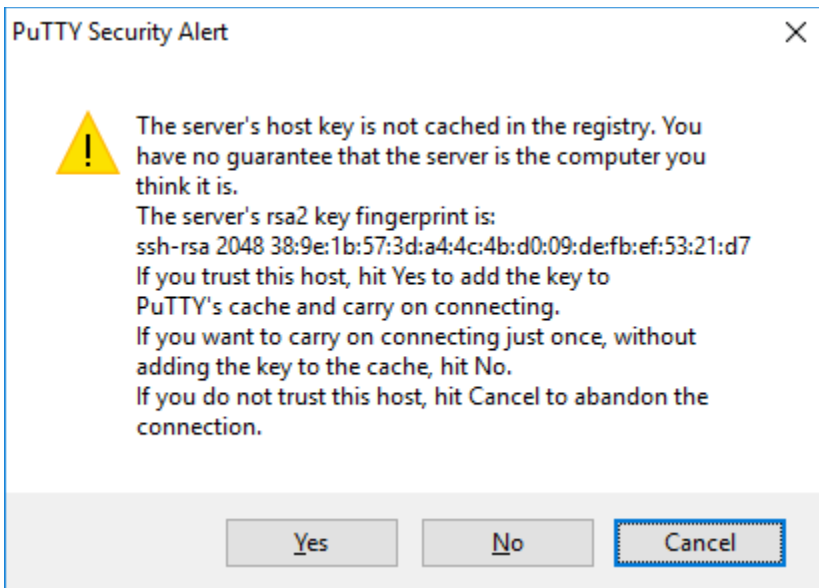
- USER: User name of the google account or the gmail account
 - E.g.: john@gmail.com → john: USER
- Instance_Name: The name of the instance
 - E.g.: gcpVM1 → john@gcpVM1
- PROJECT_ID: The project name
- ZONE: the region and zone
- 8000: port on the local computer
- 8888: port on the remote virtual machine

For example:

\$ gcloud compute ssh **john@**=gcpVM1 --project aProjName -- zone us-east1-c
-- -L 8000:localhost:8888

```

Google Cloud SDK Shell
Welcome to the Google Cloud SDK! Run "gcloud -h" to get the list of available commands.
---
C:\APPLS\GOOGLE\GCP_SDK>
C:\APPLS\GOOGLE\GCP_SDK>
C:\APPLS\GOOGLE\GCP_SDK>gcloud compute ssh inewtechs@tf-keras-ann --project dl-tf-cnn --zone us-east1-c -- -L 8000:localhost:8888_
  
```



- Click either YES or NO
 - A SSH terminal window pops up
 - **NOTES:** This is SSH TERMINAL WINDOW #2

```
inewtechs@tf-keras-ann: ~  
Authenticating with public key "DESKTOP-MIT2KAM\Admin@DESKTOP-MIT2KAM"  
=====Welcome to the Google Deep Learning VM=====
```

Version: ml3
Based on: Debian GNU/Linux 9.6 (stretch) (GNU/Linux 4.9.0-8-amd64 x86_64\n)

Resources:

- * Google Deep Learning Platform StackOverflow: <https://stackoverflow.com/questions/tagged/google-dl-platform>
- * Google Cloud Documentation: <https://cloud.google.com/deep-learning-vm>
- * Google Group: <https://groups.google.com/forum/#!forum/google-dl-platform>

To reinstall Nvidia driver (if needed) run:
sudo /opt/deeplearning/install-driver.sh
TensorFlow comes pre-installed with this image. To install TensorFlow binaries in a virtualenv (or conda env), please use the binaries that are pre-built for this image. You can find the binaries at
/opt/deeplearning/binaries/tensorflow/
If you need to install a different version of Tensorflow manually, use the command in Deep Learning image with the right version of CUDA

Linux tf-keras-ann 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64

The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

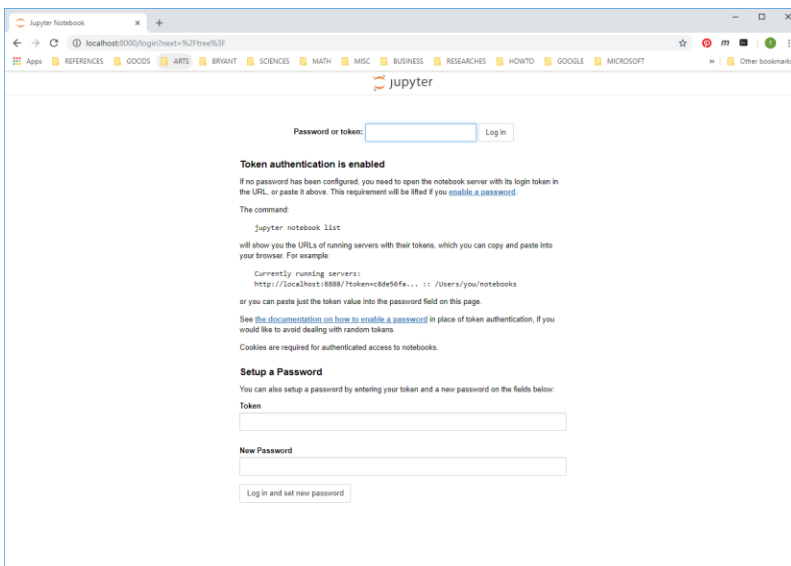
```
inewtechs@tf-keras-ann:~$
```

IMPORTANT NOTES:

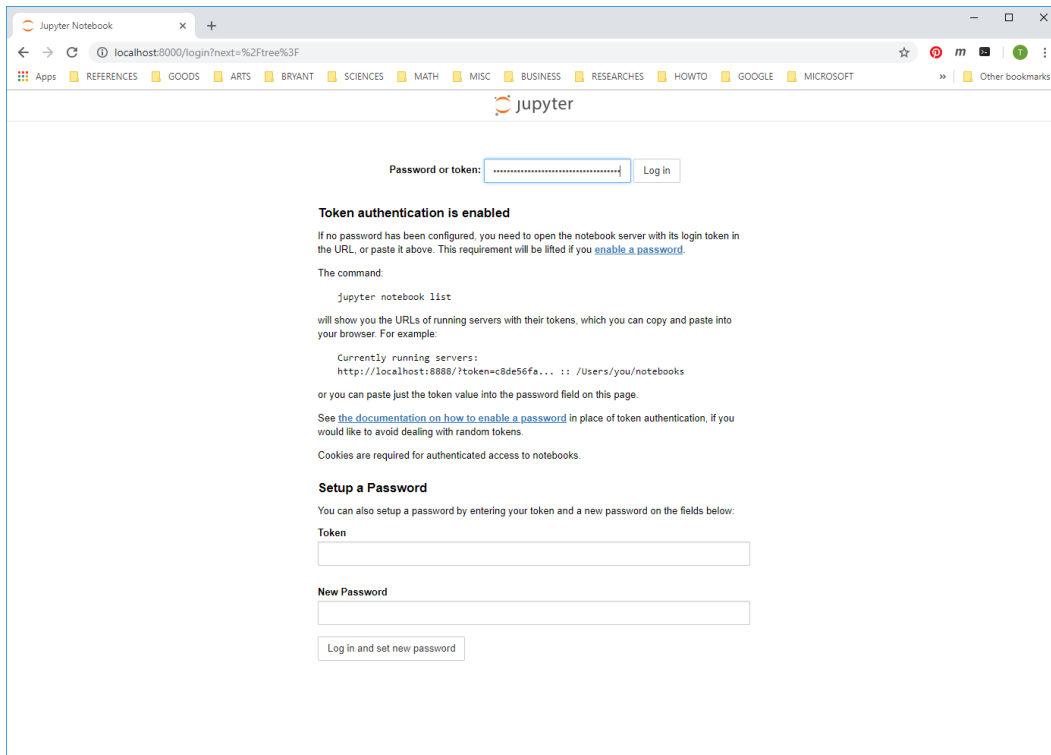
--> LEAVE THIS SSH TERMINAL WINDOW AS IS: DON'T CLOSE IT

4. Step 3: Connect to Jupyter Notebook Server in Remote Virtual Machine

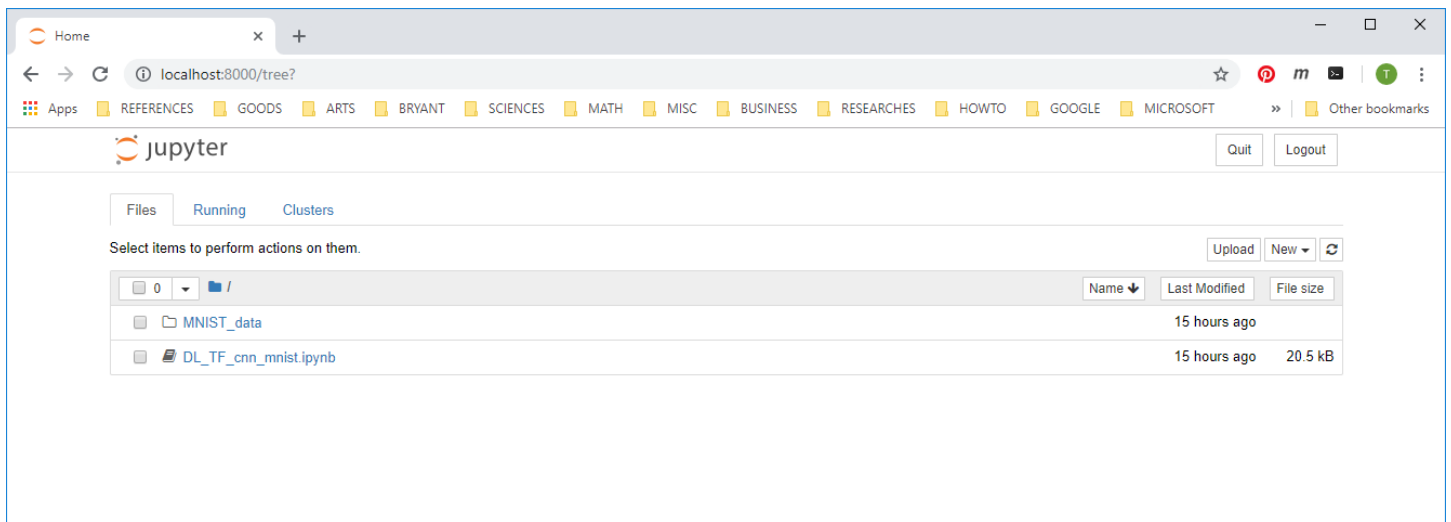
- Open Chrome browser
- Enter: <http://localhost:8000>
 - Into the URL search box



- Copy the token that has been copied into the aforementioned notepad file into the text field at the top



- Click Log in



SUCCESSFULLY CONNECT TO JUPYTER NOTEBOOK IN THE REMOTE VM!

JUPYTER NOTEBOOK IS READY FOR USE!

5. Stop Jupyter Notebook Server

- Inside the SSH terminal window #1 (in which Jupyter Notebook server starts:
 - Press **CTRL + C**
 - Be asked: y/n → Enter y (within 5 seconds)

```
inewtechs@tf-keras-ann: ~
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
inewtechs@tf-keras-ann:~$
inewtechs@tf-keras-ann:~$
inewtechs@tf-keras-ann:~$ jupyter notebook
[I 18:08:13.173 NotebookApp] Writing notebook server cookie secret to /run/user/1001/jupyter/notebook_cookie_secret
[I 18:08:14.848 NotebookApp] jupyter_tensorboard extension loaded.
[I 18:08:14.884 NotebookApp] JupyterLab extension loaded from /usr/local/lib/python3.5/dist-packages/jupyterlab
[I 18:08:14.884 NotebookApp] JupyterLab application directory is /usr/local/share/jupyter/lab
[I 18:08:14.885 NotebookApp] Serving notebooks from local directory: /home/inewtechs
[I 18:08:14.886 NotebookApp] The Jupyter Notebook is running at:
[I 18:08:14.886 NotebookApp] http://localhost:8888/?token=b74651caa7b42f13949251b3af78bdbfdd6b65a24f837ef0
[I 18:08:14.886 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[W 18:08:14.886 NotebookApp] No web browser found: could not locate runnable browser.
[C 18:08:14.886 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8888/?token=b74651caa7b42f13949251b3af78bdbfdd6b65a24f837ef0

^C[I 18:08:52.337 NotebookApp] interrupted
Serving notebooks from local directory: /home/inewtechs
0 active kernels
The Jupyter Notebook is running at:
http://localhost:8888/?token=b74651caa7b42f13949251b3af78bdbfdd6b65a24f837ef0
Shutdown this notebook server (y/[n])? y
[C 18:08:54.799 NotebookApp] Shutdown confirmed
[I 18:08:54.800 NotebookApp] Shutting down 0 kernels
inewtechs@tf-keras-ann:~$
inewtechs@tf-keras-ann:~$
inewtechs@tf-keras-ann:~$
inewtechs@tf-keras-ann:~$ ^C
inewtechs@tf-keras-ann:~$
```