Setting up Jupter notebooks

Restart VM instance, and login.

Repeating instructions from the assignment 0 GCP guide:
- Restart your instance with "Compute Engine > VM Instances" > click the instance > click the "start / play" button
- Get the ssh login command with "Compute Engine > VM Instances" > click the dropdown next to "SSH" > click "view gcloud command". Run the command on your local machine to ssh into the vm instance.
- If these steps aren't working, return to assignment 0 guide, section 6.

Set up Jupyter in the vm instance

After you've ssh'd into the vm instance run:
`jupyter notebook --generate-config`

Open the file `~/.jupyter/jupyter_notebook_config.py` with an editor like **vim** or **nano** (if you're not comfortable with command-line editors Nano is a good option).

Find the following lines, uncomment them by deleting the `#` character, and change the values to:

```
c.NotebookApp.ip = '*'
c.NotebookApp.port = 7000
```

Exit and save.

Run:

```
jupyter notebook
```

You'll see a lot of output, including some web addresses like this:

```
[I 21:32:06.425 NotebookApp] [nb_conda] enabled
/opt/conda/lib/python3.7/site-packages/jupyter_server/transutils.py:13: FutureWarning: The alias `_()` will be deprecated. Use `_i18n
  warnings.warn(warn_msg, FutureWarning)
[I 21:32:06.550 NotebookApp] Serving notebooks from local directory: /home/jamesburgess
[I 21:32:06.550 NotebookApp] Jupyter Notebook 6.4.0 is running at:
[I 21:32:06.551 NotebookApp] http://deeplearning-2-vm:7000/?token=bd9097cccd3d85fd5ff5abfe9ab12e8e867601bf8ab14f2b
[I 21:32:06.551 NotebookApp]  or http://127.0.0.1:7000/?token=bd9097cccd3d85fd5ff5abfe9ab12e8e867601bf8ab14f2b
[I 21:32:06.551 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[W 21:32:06.555 NotebookApp] No web browser found: could not locate runnable browser.
[C 21:32:06.556 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/jamesburgess/.local/share/jupyter/runtime/nbserver-31649-open.html
    Or copy and paste one of these URLs:
        http://deeplearning-2-vm:7000/?token=bd9097cccd3d85fd5ff5abfe9ab12e8e867601bf8ab14f2b
     or http://127.0.0.1:7000/?token=bd9097cccd3d85fd5ff5abfe9ab12e8e867601bf8ab14f2b
```
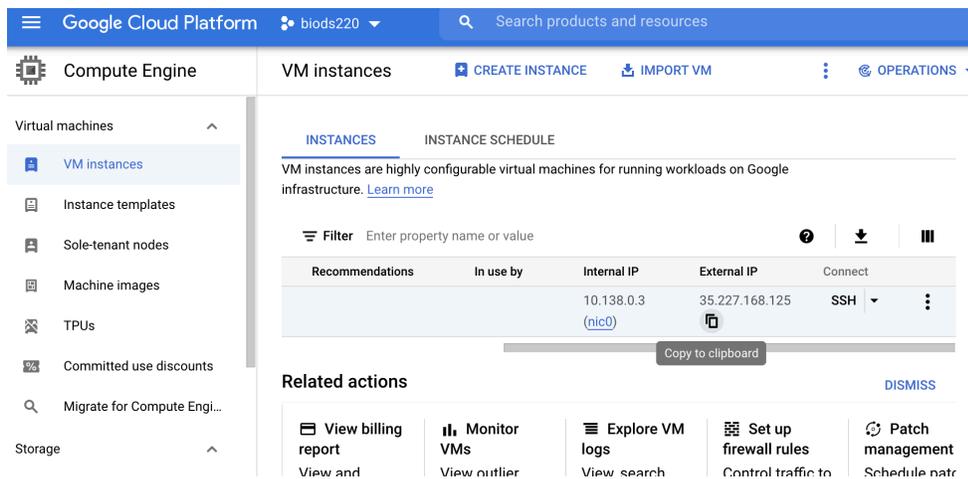
Don't try to join those links in the browser, it won't work.
We need to do another step:

Connect to Jupyter in the browser

Copy your IP address from the page you were just on, "External IP Addresses".

Alternatively, you can always find the IP address from your list of VM instances. That is, you go to "Compute Engine > VM Instance", and then it's under the column "External IP".

You can use this IP address to access the notebook on your local browser with
`http://<external-static-ip-address>:<port-number>`
(Important: this is "http", not "https").
In this case, the external IP is `35.227.168.125`, and in the config step earlier, we set the port number to 7000, so we want to go to:
`http://35.227.168.125:7000`

(WARNING: when you shut down and start the VM instance again, the IP address may change).

You may be taken to a page that asks you to enter a token. You can find that token in the output of the terminal. For example, if your terminal said this:

Then the token is the code starting with `bd9….`

Some common problems if you can't connect on your browser

Possible problem 1:
Cannot connect to the website, and you get an error message like this on the terminal:

```
[I 00:20:02.894 NotebookApp] Malformed HTTP message from 128.12.123.42: Malformed HTTP version in HTTP Request-Line: 'êê\x13\x01\x13\x02\x1
3\x03À+À/À,À0Ï©Ï¨À\x13À\x14\x00\x9c\x00\x9d\x00/\x005\x01\x00\x01\x93'
[I 00:20:02.894 NotebookApp] Malformed HTTP message from 128.12.123.42: no colon in header line
[I 00:20:39.906 NotebookApp] Malformed HTTP message from 128.12.123.42: Malformed HTTP version in HTTP Request-Line: '\x9a\x9a\x13\x01\x13\
x02\x13\x03À+À/À,À0Ï©Ï¨À\x13À\x14\x00\x9c\x00\x9d\x00/\x005\x01\x00\x01\x93'
```

This can be caused by using "https" instead of "http".
So instead of typing this in the browser:
https://35.227.168.125:7000
Instead type:
http://35.227.168.125:7000

Possible Problem 2:
Firstly check that the port was correct.In the above image, we see the line
`http://127.0.0.1:7000/`
Which indicates the port really was 7000. If it's a different number like
`http://127.0.0.1:8888/`,then change the port number in your browser (so in the above example,
it would be http://35.227.168.125:8888).

Possible Problem 3:
If you are getting an error like "No web browser found: could not locate runnable browser.
Jupyter-notebook", try launching jupyter with:

```
jupyter notebook --no-browser
```

Run jupyter in a `tmux` or `screen` session

By running `jupyter notebook`, we have created a server that lets us run notebooks.
If we kill it by running `ctrl-c` then if we go back to http://35.227.168.125:7000, we can't
access the notebook anymore.
Likewise, if we close the terminal window, then that kills the process as well, and we can't
access the notebook anymore.

To get around this problem, you can run Jupyter in a `tmux` or `screen` window. The end of
HW0 GCP instructions has some basic tmux instructions.

Run the Tensorflow "HelloWorld"

Ssh into your vm instance, and download the tensorflow "hello world" program that we've
modified slightly (taken from this article) by running

```
wget https://biods220.stanford.edu/notebooks/hello-world.ipynb
```

For the first cell of code, you should get something like the following, indicating that GPU's are available:

```
In [5]: import tensorflow as tf
        print("devices:", )
        [print(s) for s in tf.config.list_physical_devices()];

        devices:
        PhysicalDevice(name='/physical_device:CPU:0', device_type='CPU')
        PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')
```

The results here indicate that we can access both CPUs and GPUs from our VM.

We can also practice moving a tensor between CPU and GPU (tensors are data structures similar to arrays used in deep learning packages) …..

Run each cell in the code, check that the model trains. You should see similar results to the article.

As always, don't forget to stop your instances - instructions in the GCP guide