Guide to Running a Deep Learning VM in Google Cloud.

Contents

IMPORTANT NOTE:

Before you start this guide please log out of all existing google accounts except your mycit account.

Part 1: (Obtaining Coupons)

Dear MSc in AI Student,

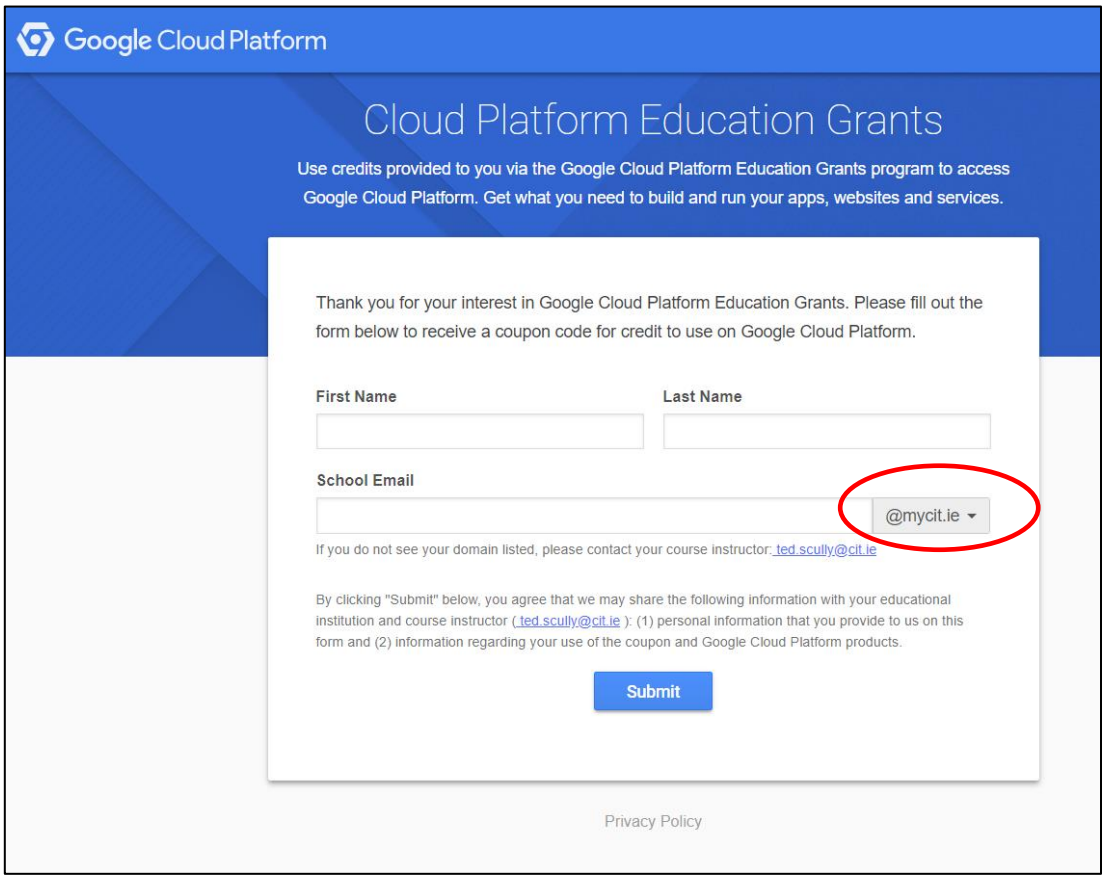
Below is the URL you will need to access in order to request a Google Cloud Platform coupon.

You will be asked to provide your college email address and name (please only use your **@mycit.ie email address**).

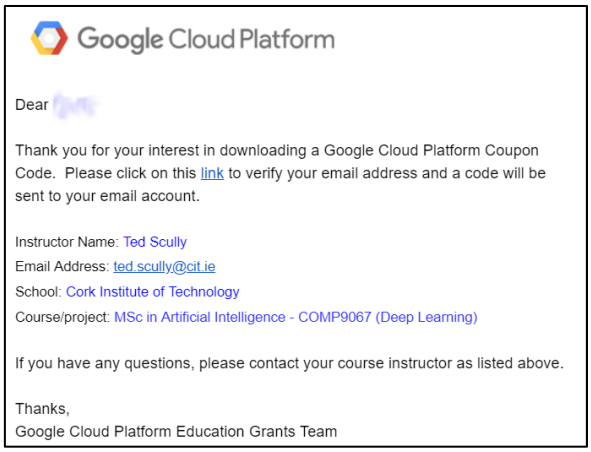
An email will be sent to you to confirm these details before a coupon is sent to you.

Student Coupon Retrieval Link

- You will be asked for a name and email address, which needs to match the domain (see the screen shot below). Again please remember you must select the **@mycit.ie** email from the drop down box (highlighted in red below)
- A confirmation email will be sent to you with a coupon code.
- You can only request ONE code per unique email address.



After submitting the above request you will receive an email asking you to verify your email address. See below.



Google Cloud Platform

Dear [redacted]

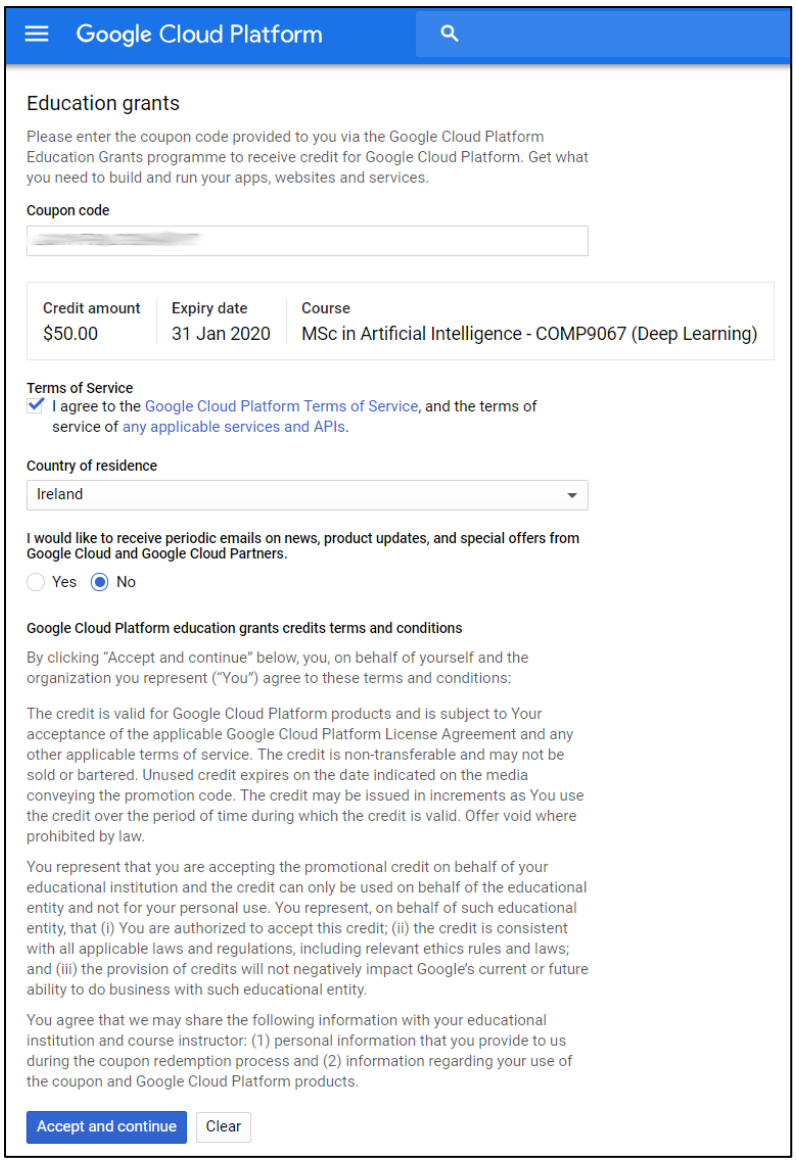
Thank you for your interest in downloading a Google Cloud Platform Coupon Code. Please click on this link to verify your email address and a code will be sent to your email account.

Instructor Name: Ted Scully
Email Address: ted.scully@cit.ie
School: Cork Institute of Technology
Course/project: MSc in Artificial Intelligence - COMP9067 (Deep Learning)

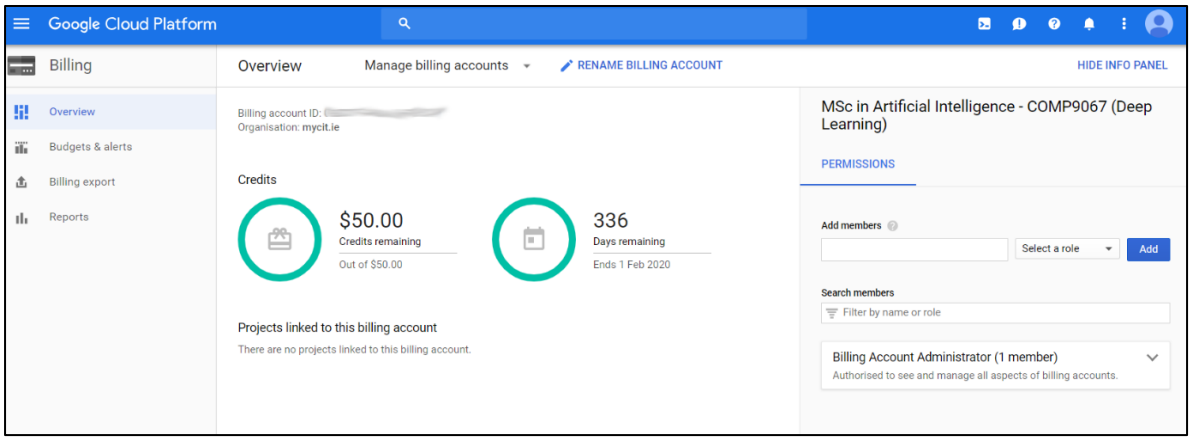
If you have any questions, please contact your course instructor as listed above.

Thanks,
Google Cloud Platform Education Grants Team

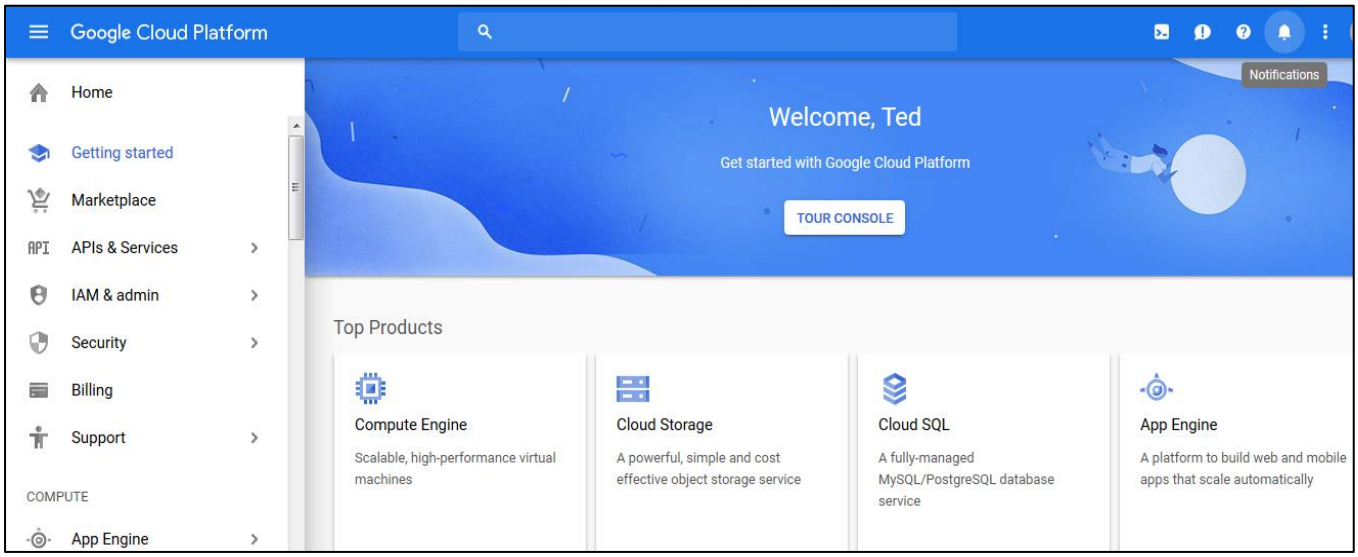
Once you verify your email address you should then receive another email containing the **coupon code** and a **link to click** that will allow you to redeem the code for credits. You will notice in the email that it contains the text “Click [here] to redeem”. When you click on this link it will take you to the page shown in the screen shot below. Here you can enter your coupon code. See screen shot below.



Once you accept and continue, you are taken to the billing page for the Google Cloud Platform (see below). It's a good idea to bookmark this page so you can return later.

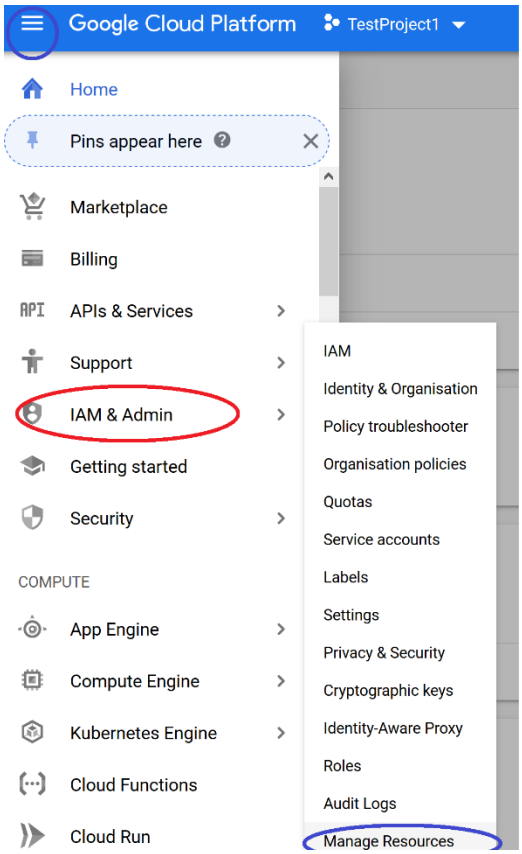


If you click on the “Google Cloud Platform“ on the top left hand corner of the page it should take you to a page similar to that shown below.

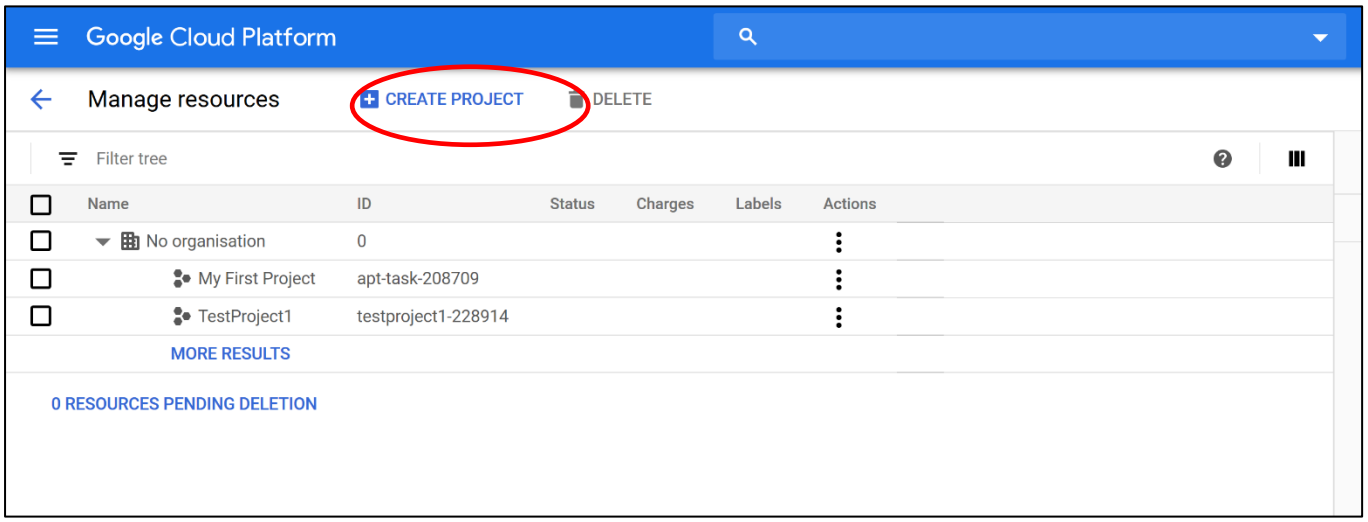


Part 2: (Create a Project)

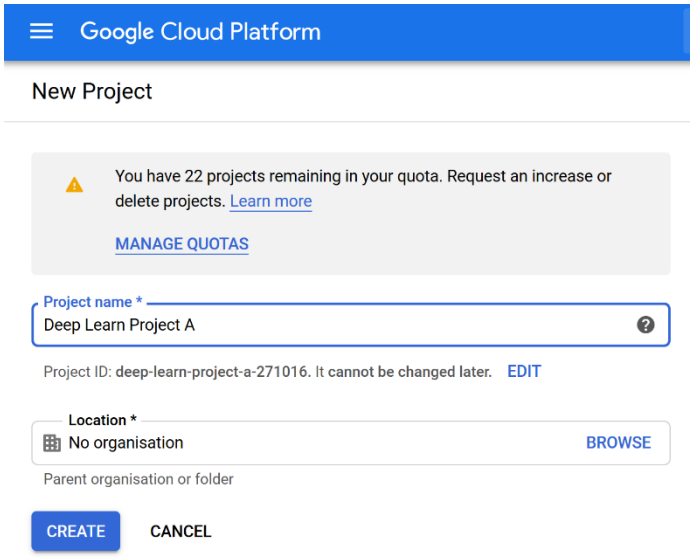
Go the Manage Resources page. Please note you can find the Manage Resources page in the side menu (highlighted in purple) under “IAM and Admin” (highlighted in red). The Manage Resources link is highlighted in blue below.



The ‘Manage Resources’ page will list your existing projects. By default a “My First Project” is created for you. We will create a new project by clicking the “Create Project” link (Highlighted below).



When creating the project just provide a project name (there is no need to specify organization). See screen shot below. Please note it may take a minute to create the new project. If you refresh your ‘manage resources’ page you should see the new project which I have called “Deep Learn Project A” listed.

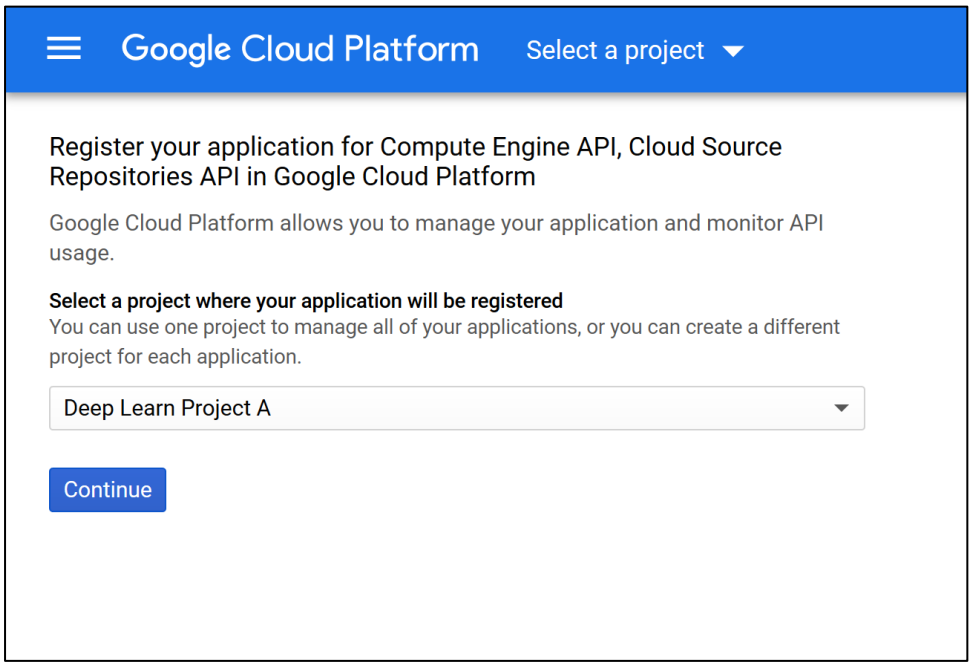


Part 3: (Enable APIs)

You will need to enable both the Google Compute Engine and Cloud Source Repositories APIs.

To do this you please go here. You should see the page shown below.

Click on the drop down menu and select a project. In this case I will be selecting “Deep Learn Project A”. Now click Continue. Your Google Cloud Platform is now enabling your APIs. This make take a few minutes to complete. Please wait until it completes.



Part 4: (Install and initialize the Cloud SDK)

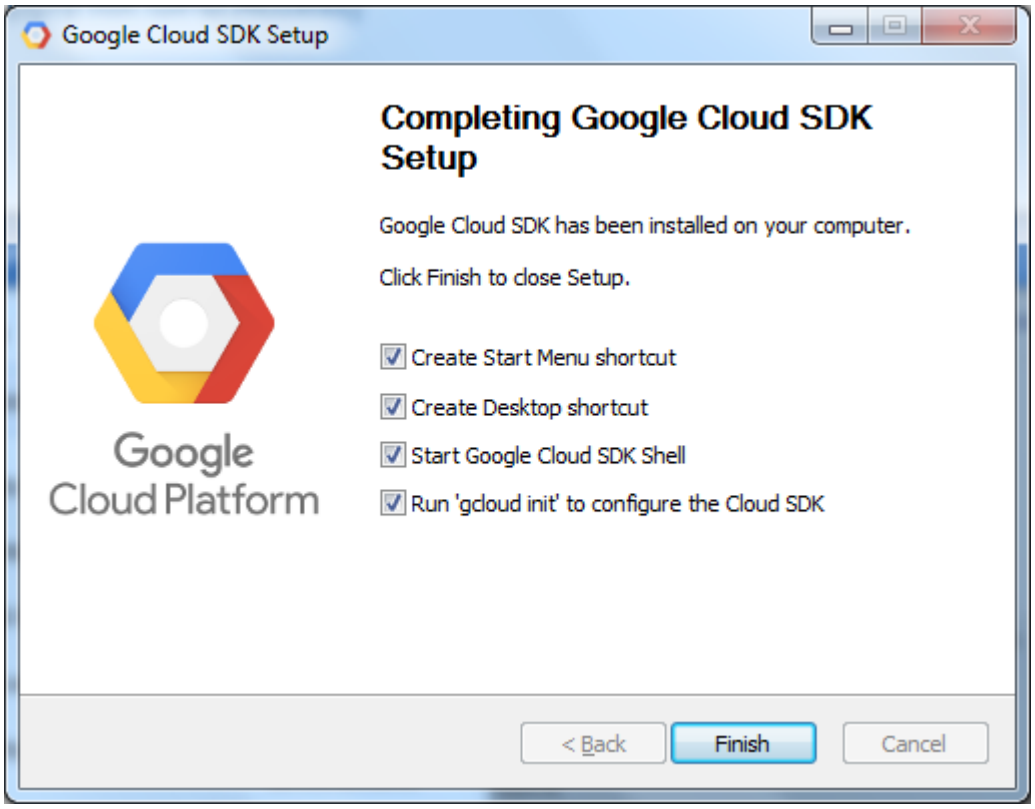
As most of you will be using Windows, Part 4 describes the Windows installation process. Google Cloud SDK is a set of tools that you can use to manage resources and applications hosted on Google Cloud Platform.

If you have multiple Google account please log out of all except the myCIT account.

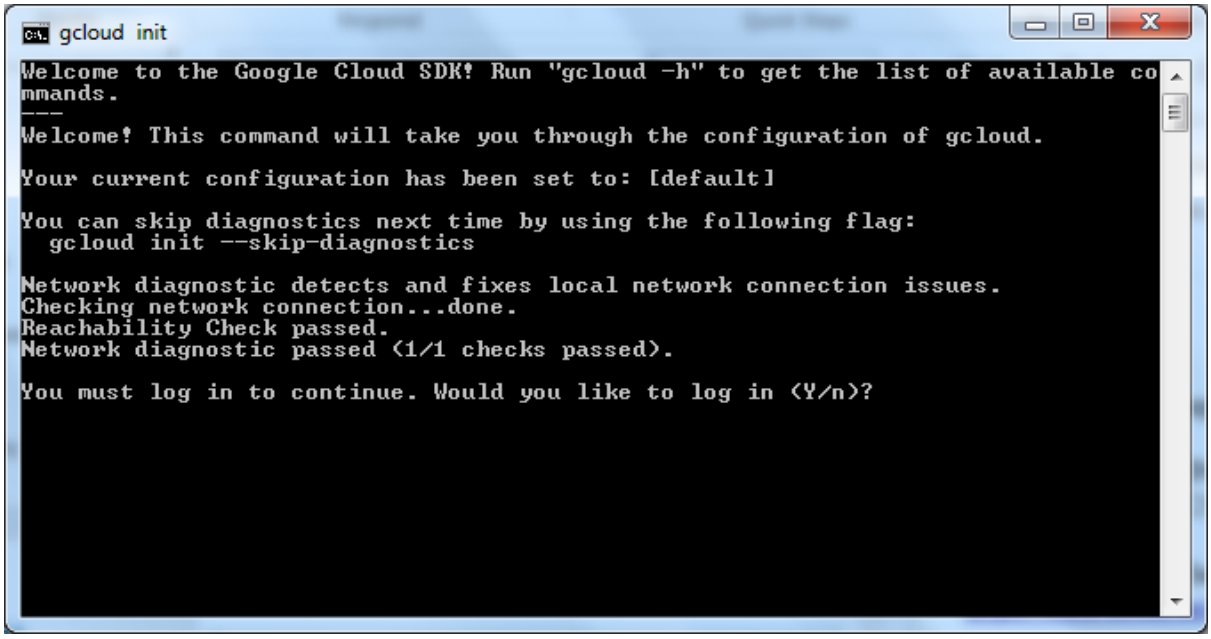
To install the Cloud SDK please follow the first three steps listed here. (When you go to this link you will notice there is also tabs available for Linux, Mac, etc).

1. The first step asks you to download the Could SDK installer.
2. In the second step you will launch the installer and follow the prompts. I would recommend adopting all default settings.
3. After installation has completed, accept the following options (see image below):
    a. Start Cloud SDK Shell
    b. Run gcloud init

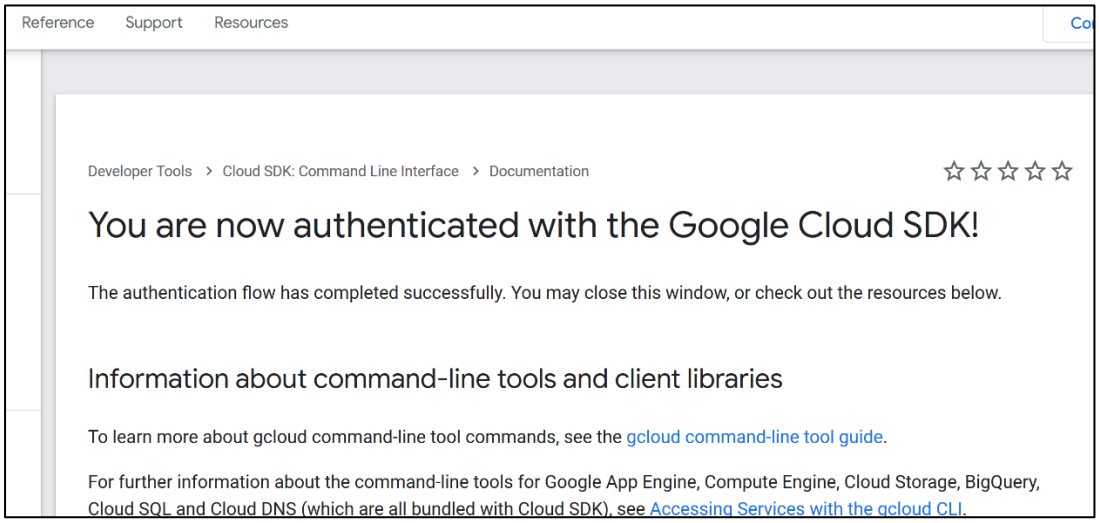
    Click Finish.



Next the installer starts a terminal window and may ask you to log in.

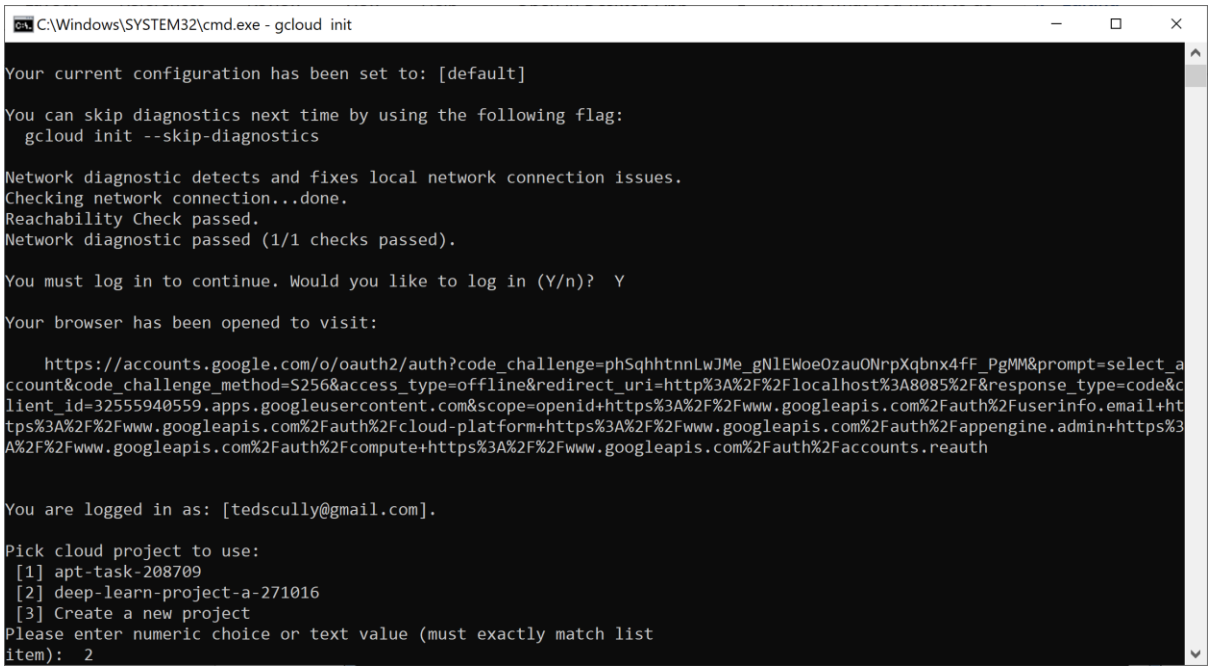


This will launch a browser where you are asked to accept certain terms and conditions. Click Allow. You should then receive a message as shown below saying you are authenticated.

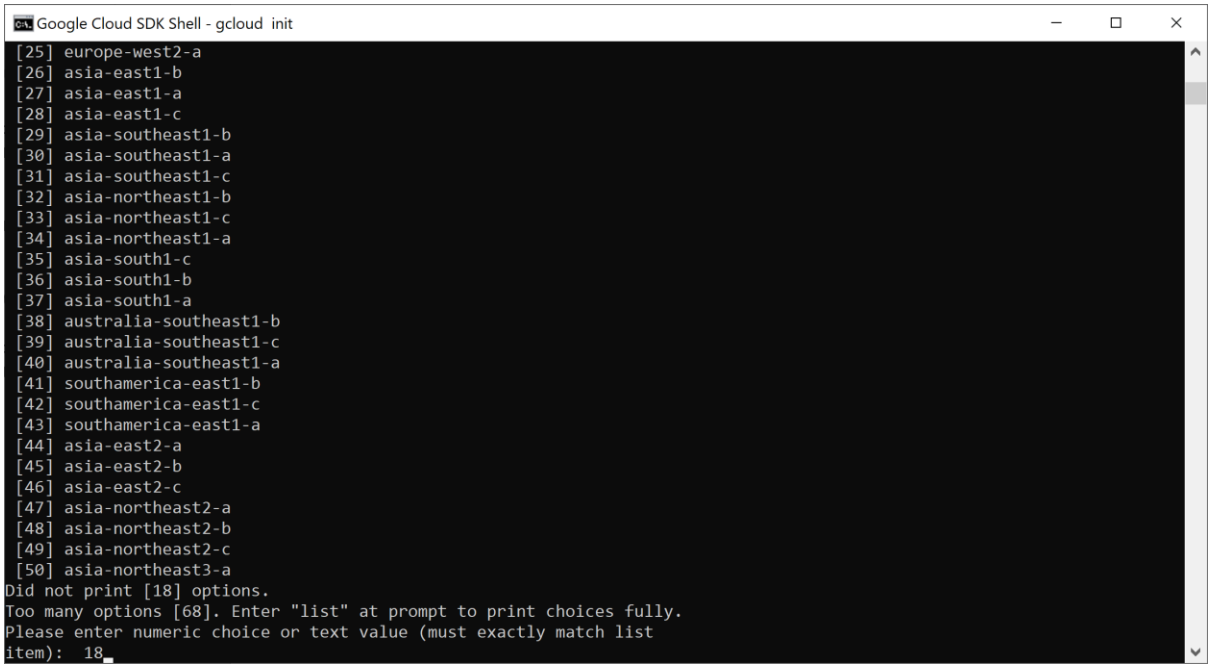


You should notice that that the terminal window indicates you are now logged in and asks you which of your projects you want to use. I want to use my “deep learn project a” so I select 2 (notice deep-

learn-project-a is listed as option 2 in the menu below). [Please be aware that it may not show the direct name of the project, instead it may show the ID of the project]



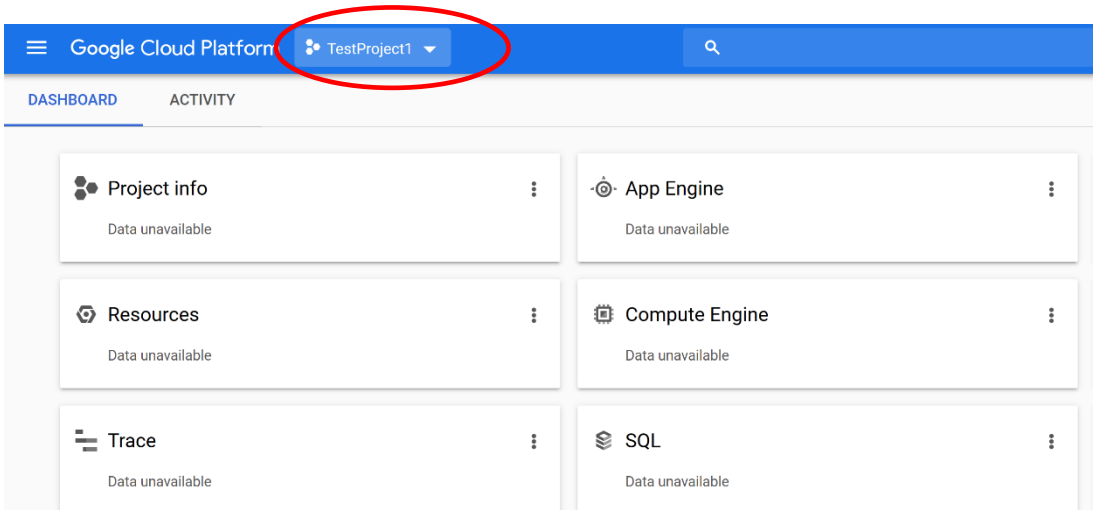
It may then ask you if you want to configure a default zone (Do you want to configure a default Compute Region and Zone?). You can say Yes to this. It will output a list of all available regions. Please enter 18 at the prompt (this is europe-west1-d, which is the region we are going to work with).



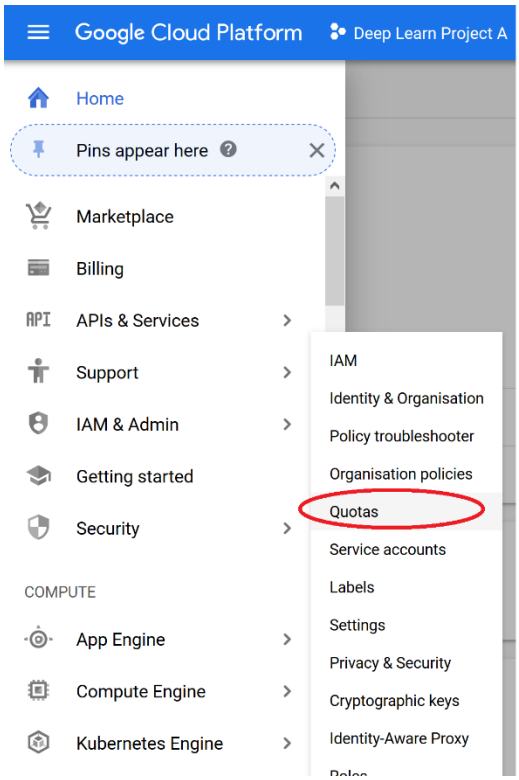
Part 5: (Apply for GPU Access)

You now need to apply for a GPU to be added to your quota in your Google Cloud Platform. Google specify that the application you submit can take up to two working days to be approved (However, each time I have applied it has been granted in less than 15 minutes).

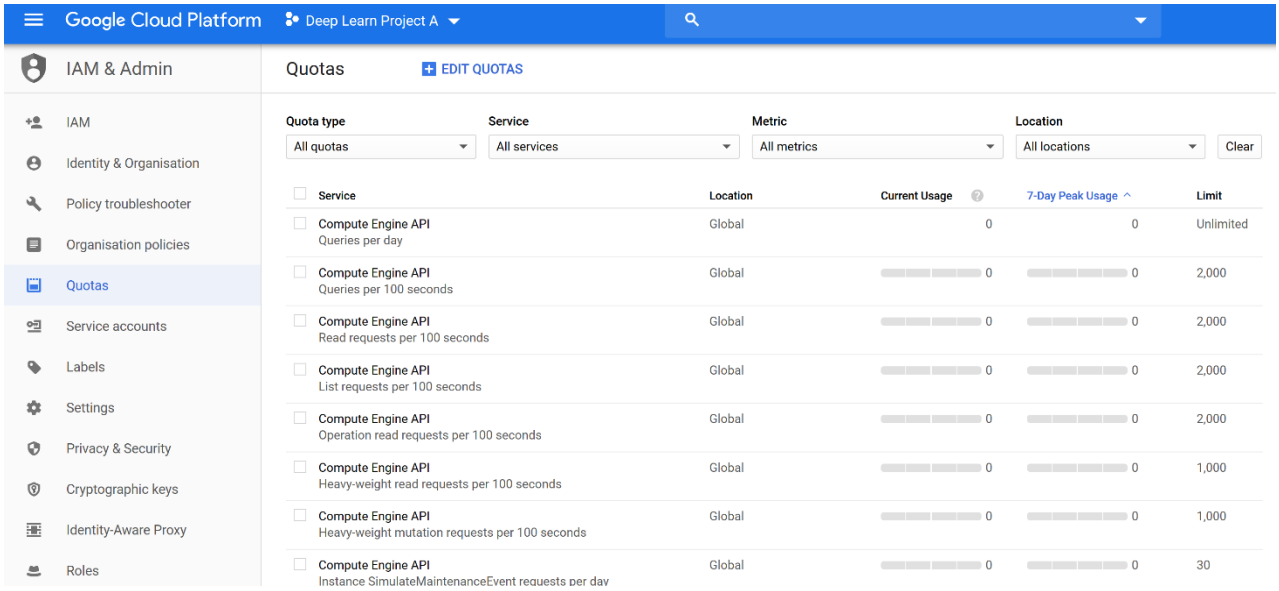
Adjacent to the Google Could Platform Title you should see a drop down menu with you current project selected (highlighted in red below). At the moment this is selected to the default TestProject1. Click the drop down menu and change the project to the one you created in Part 2. In this case I will select Deep Learn Project A. Please note you may have to enter the project name in the search field as not all projects are displayed.



Next you must apply for the GPU to be added to your quota. To do select the drop down menu (on the left of Google Cloud Platform title) and select “IAM and Admin” following by “Quotas”. See image below.



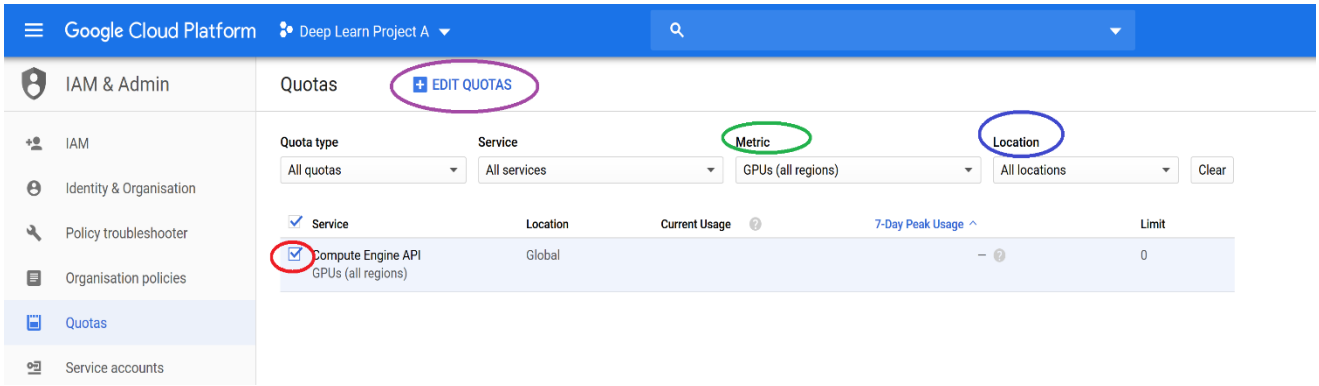
This will take you to the following page.



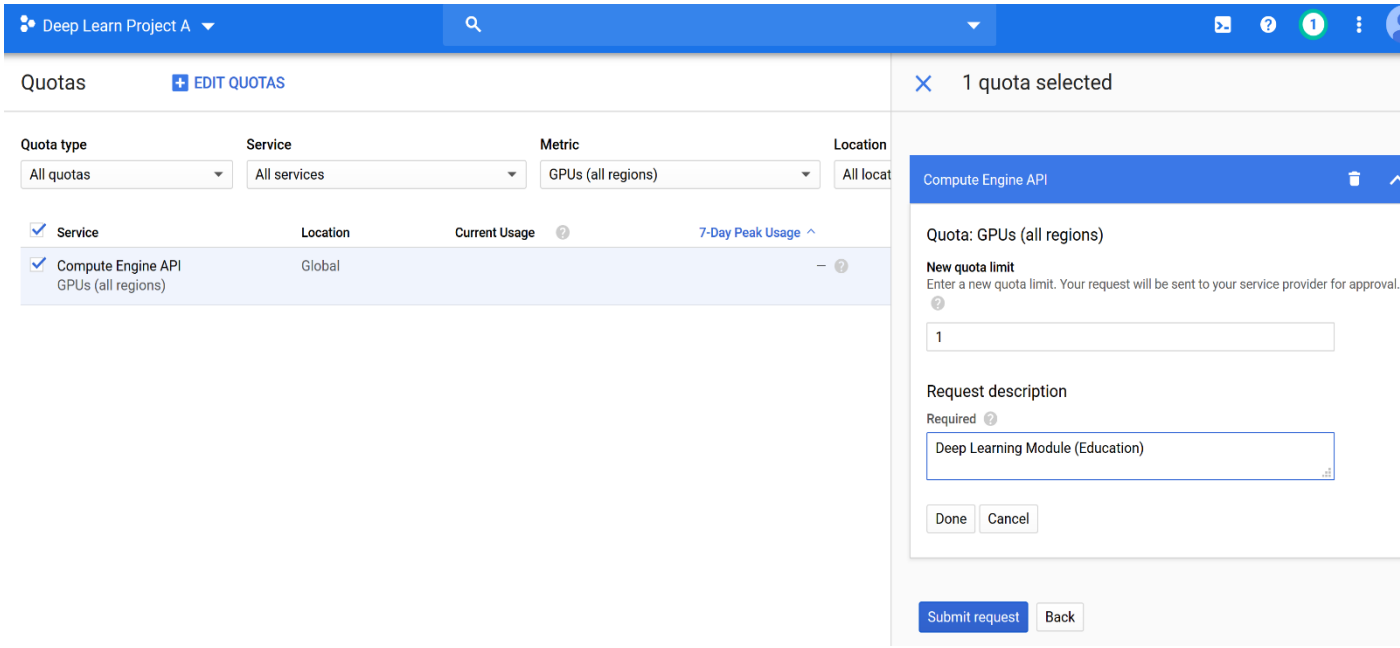
Select the Location as “All Locations” (highlighted blue). Under metric drop down just select “GPU (all regions)” (highlighted in green). (Please note you may have to “deselect” all the options first by selecting “None”, and then specifically select the “GPUs (all regions)” option.)

The list of options should now just show “Compute Engine API GPU (all regions)” as shown below. Select this option by clicking the checkbox (highlighted in red below).

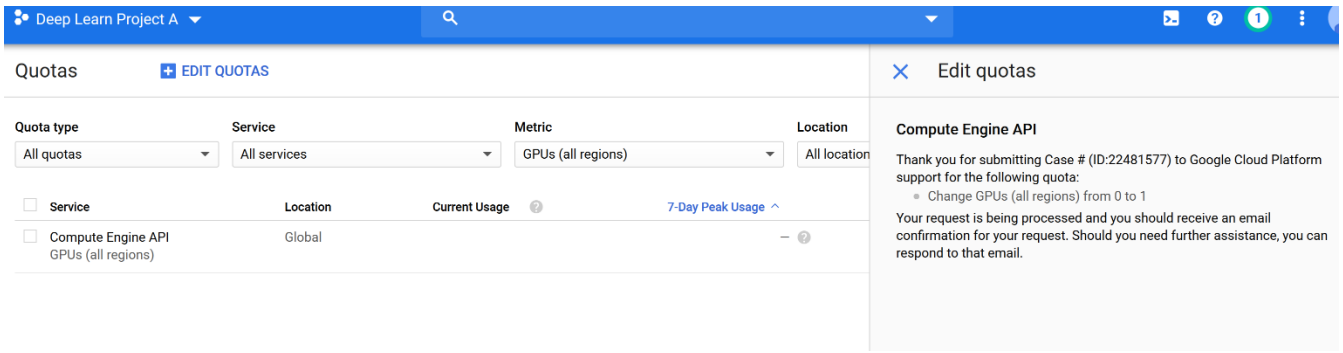
Next click Edit Quotas (highlighted in purple). This will just ask you to fill in some details.



When asked for the new quota limit just enter 1 (see below). In the Request Description just put “Deep Learning Module (Education)”. Click Done followed by Submit request.



Once you click Done and Submit request you should see the screen below (it may take a minute before this is processed and displayed).

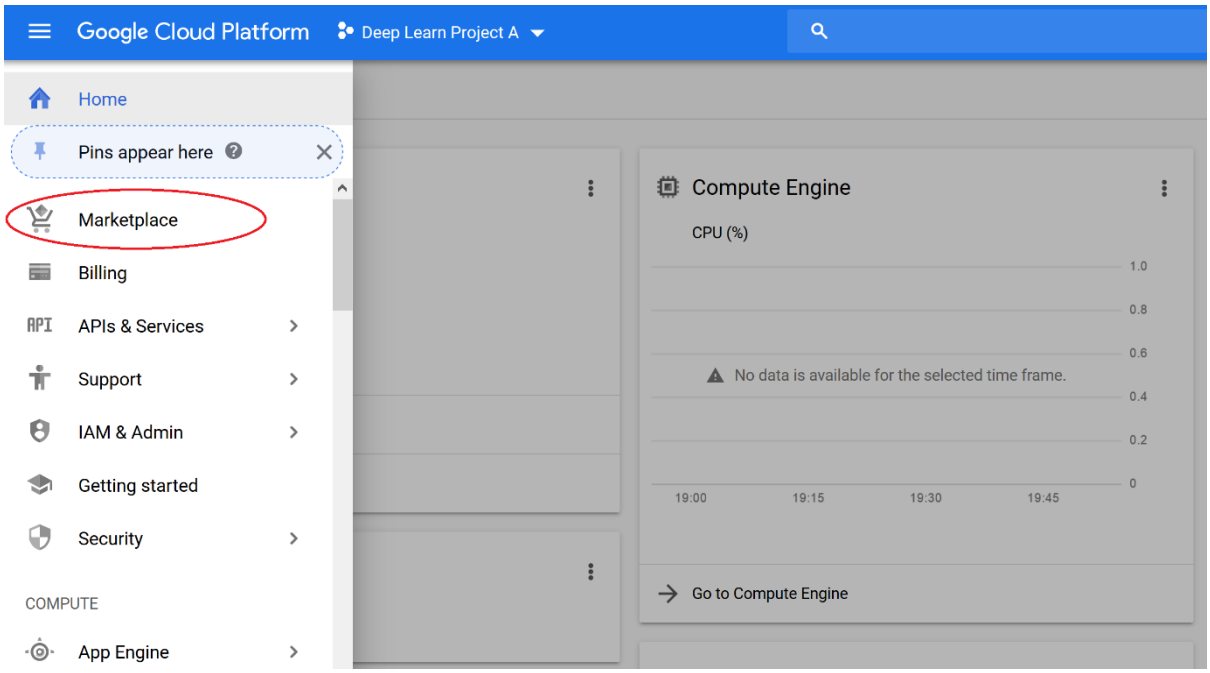


You will receive an email once the request has been submitted. You will then receive another email informing you when the GPU access has been granted (remember it may be anywhere from 10 minutes to 2 working days before this is approved).

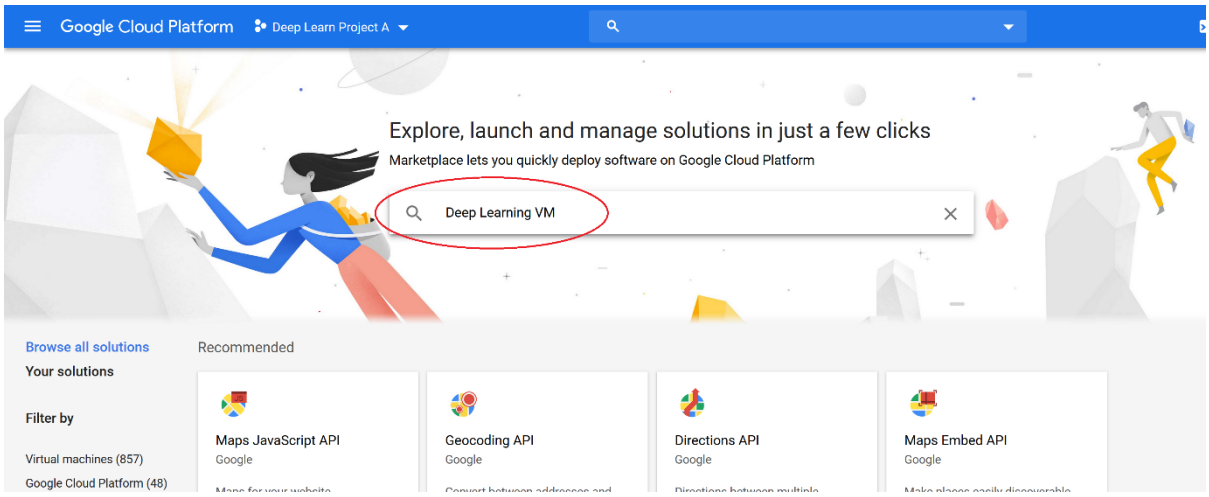
You will need to be granted GPU access before moving to the next step.

Part 6: (Deploying Deep Learning VM)

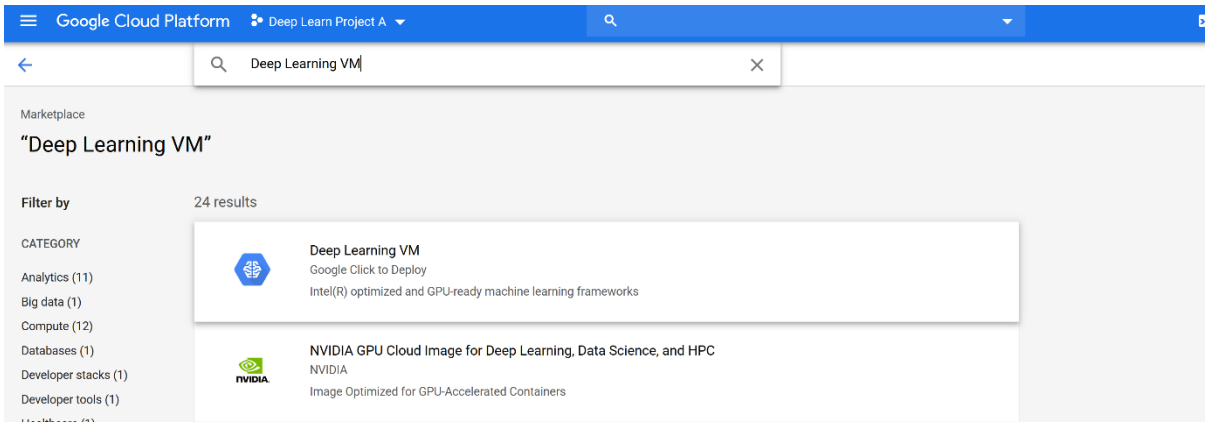
You now need to configure and launch a deep learning VM. Go the MarketPlace on the dropdown menu in the Google Cloud Platform as shown.



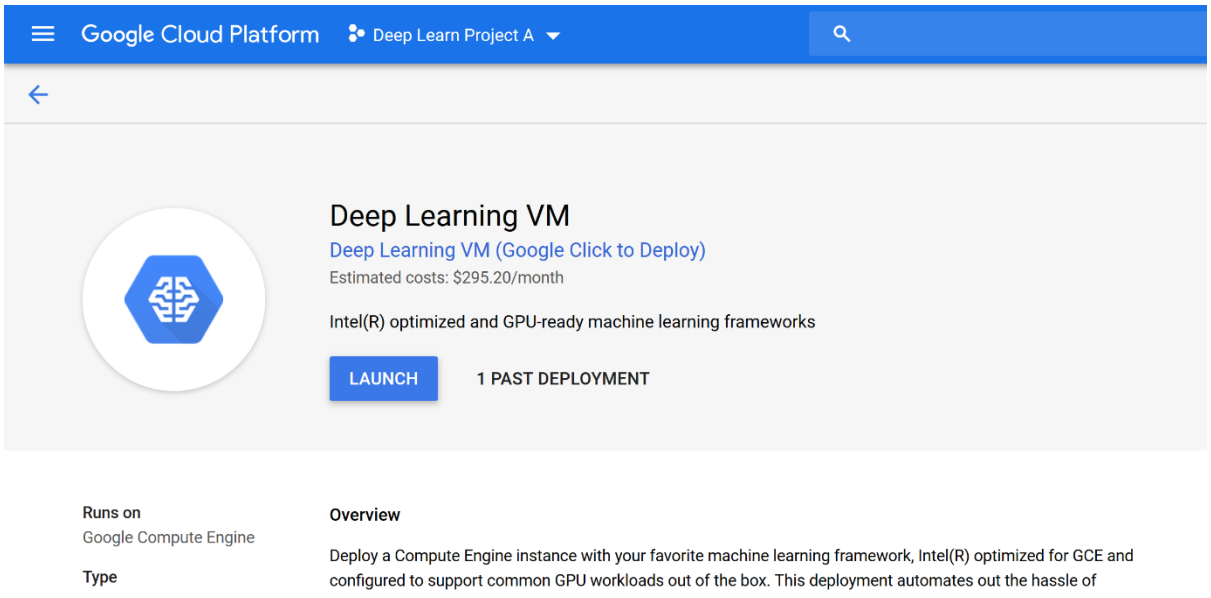
Type “Deep Learning VM” into the search field as shown.



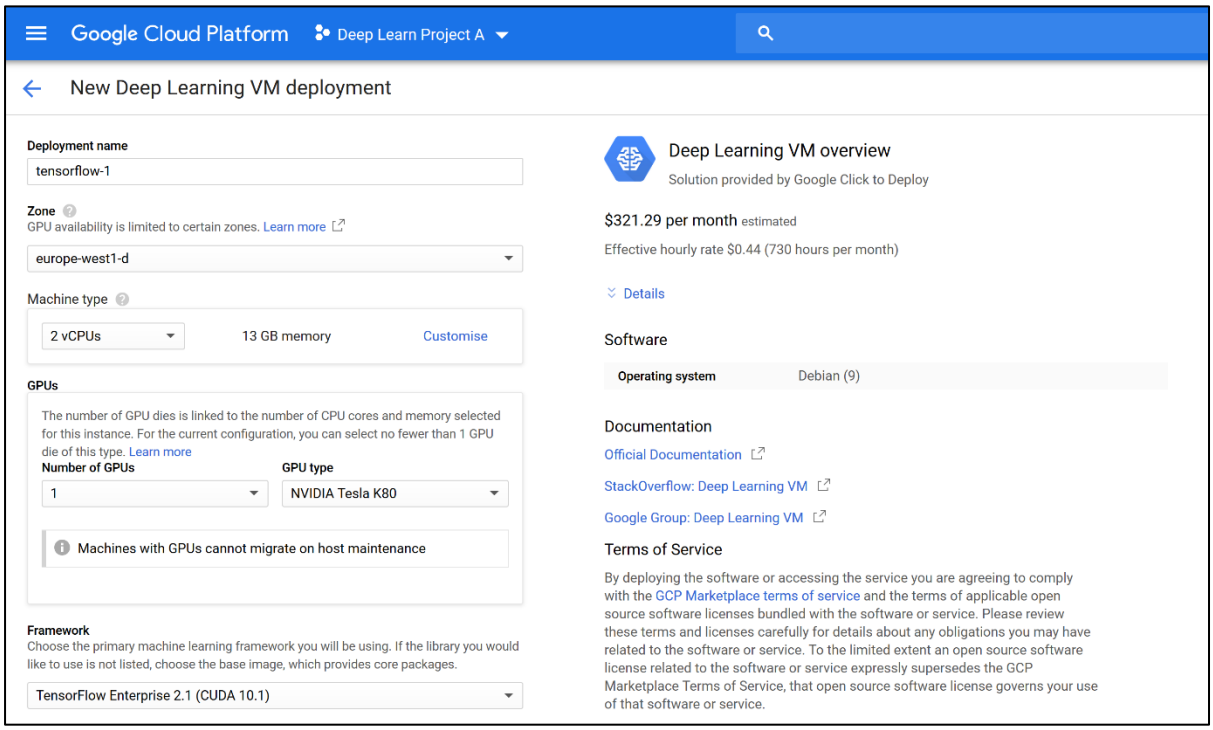
Click on the Deep Learning VM (Click to Deploy) option shown below.



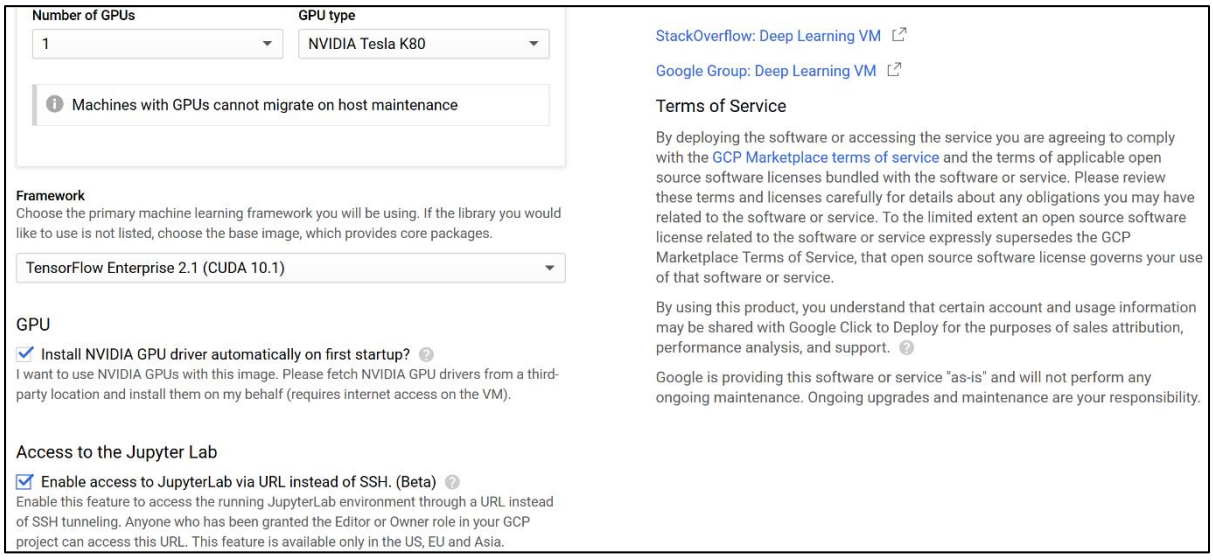
Before you go any further, make sure that the correct project is selected (this has to be same project where you applied for the additional GPUs). Notice below I have selected "Deep Learn Project A". Once you are sure the correct project is selected then next click "Launch"



Fill in the fields as shown below. Notice we call the VM tensorflow-1. We select the region as europe-west1-d (the region we have been using all along). Once all the fields have been filled in, at the end of the page click the Deploy button.

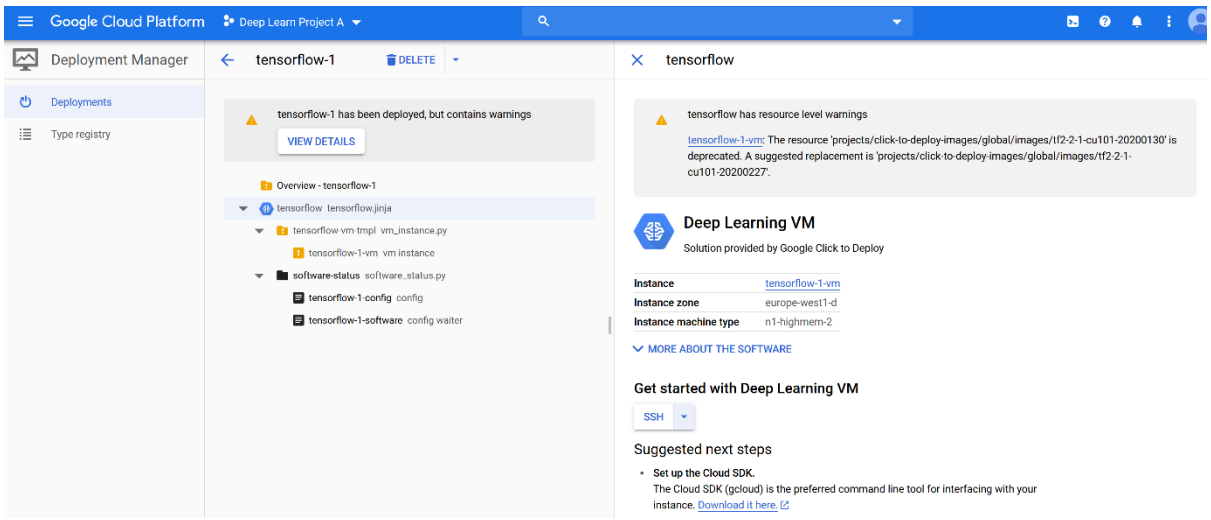


We select one GPU, the NVIDIA Tesla K80 (this is a basic GPU but is cheap to use and will be adequate for what we are going to cover). The framework is TensorFlow Enterprise 2.1. Also make sure you click the GPU checkbox and the Access to Jupyter Lab checkbox.



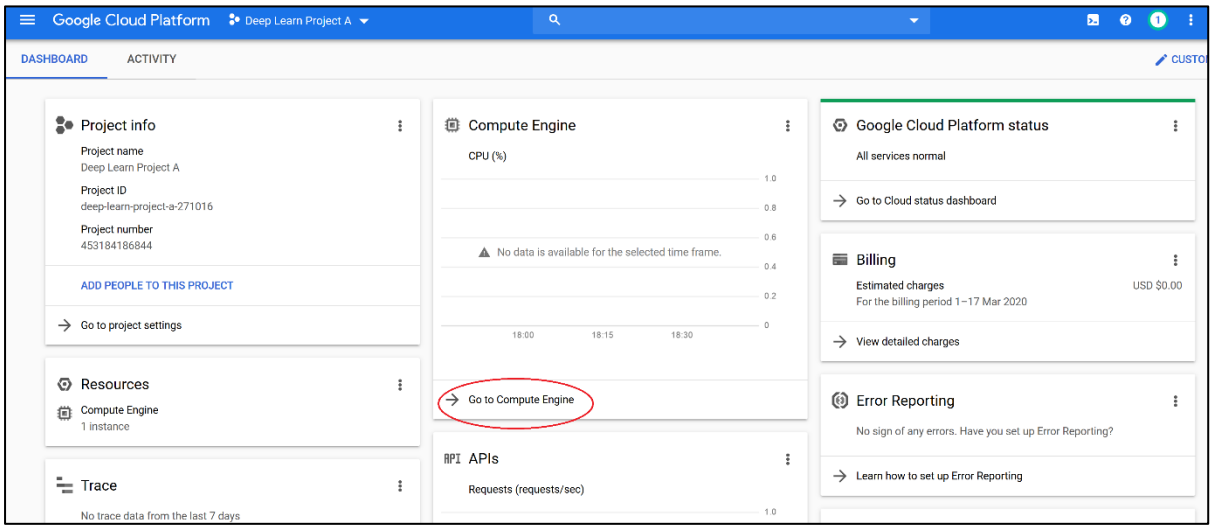
Next you should click the deploy button at the bottom of the page.

Once you click deploy you should see the page below. It may take several minutes for the deployment to finish. Please wait for this to finish.



Next let's check that this VM instance is actually running.

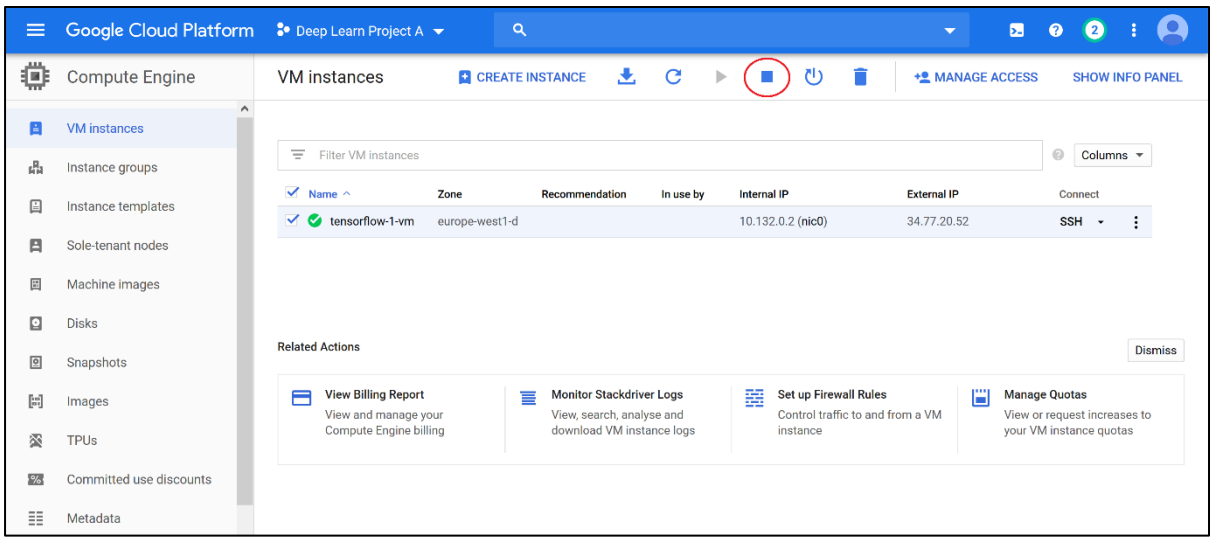
To do this we need to go back to Google Cloud Platform homepage (click Google Cloud Platform on top left of screen) as shown below. Then click on "Go to Compute Engine" link highlighted in red. Alternatively, you can also enter "Compute Engine" in the search box at the top of the page.



Once you click on this you should see the page below (note if you don't see this make sure the VM instance on the left hand menu is selected). You will notice our VM is listed as running below. Even though we named it tensorflow-1, it is named as tensorflow-1-vm below.

This page is very important as it is from this page that you can start and stop your VM instance. It is very important that you stop your VM instance when you are finished working. Otherwise you will continue to be charged. The stop button is highlighted in red below.

To restart the VM instance when you want to resume work just the play button (on the left side of the stop button). For the moment we will let the VM run but make sure to turn it off when finished working. In the next section we will connect to Jupyter notebook running on the VM instance.

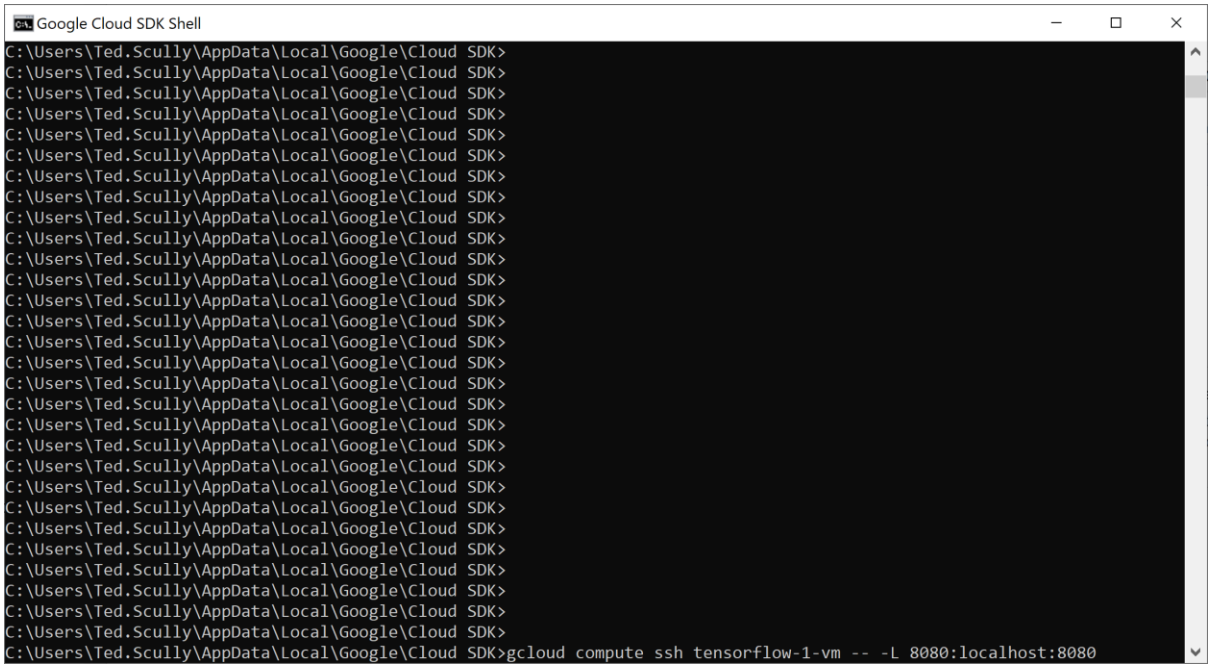


Part 7: (Connecting to Jupyter Notebook in VM instance)

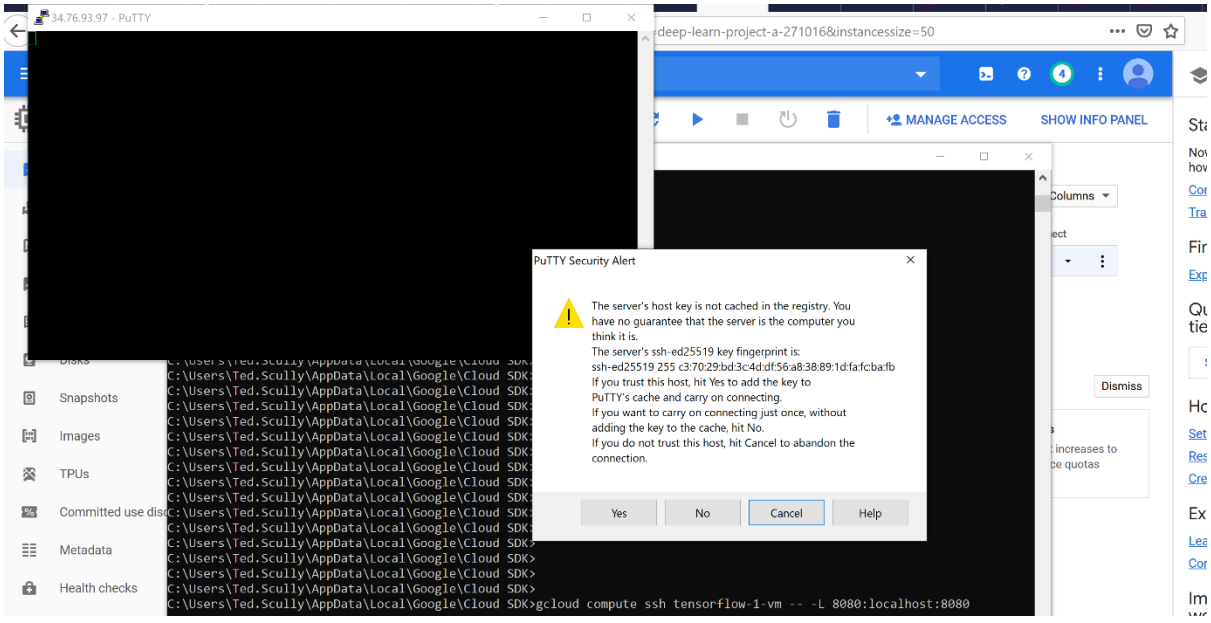
Go back to the console on your local machine. If you have already closed this then open your "Google Cloud SDK Shell". This was installed when you installed GCloud

Enter the following line of code in your console on your local machine. Notice we connect to tensorflow-1-vm, the name indicated in the list of VM instances above.

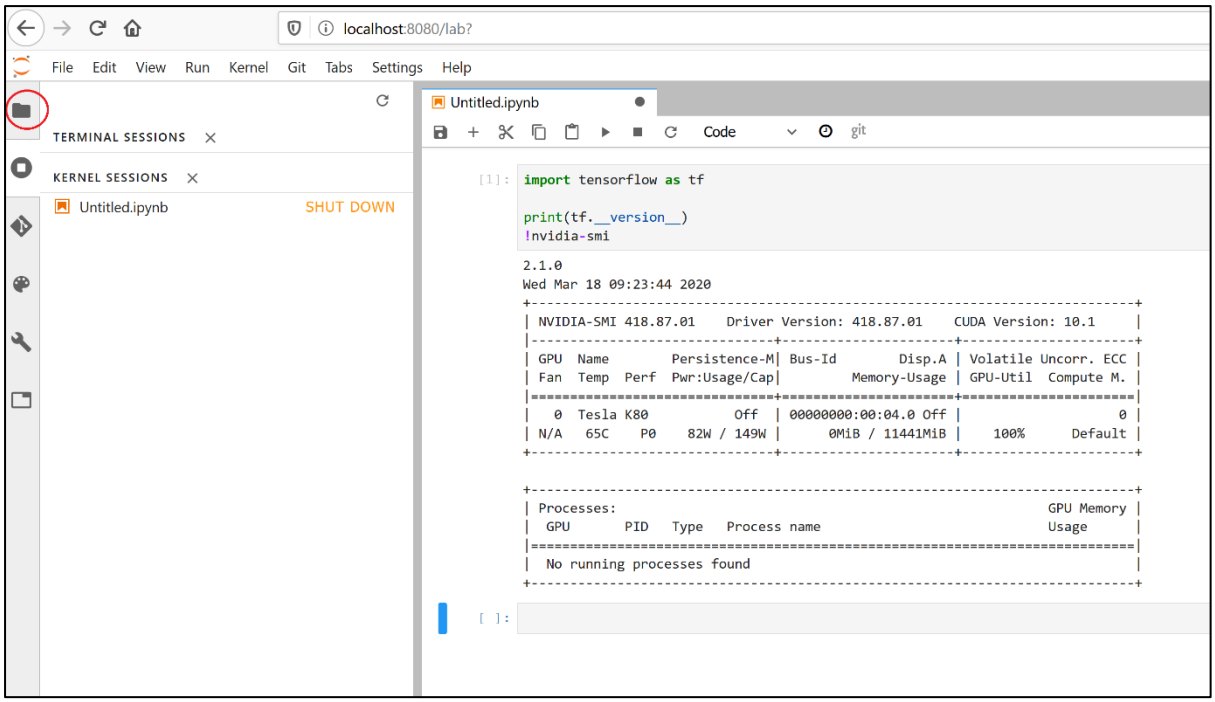
gcloud compute ssh **tensorflow-1-vm** -- -L 8080:localhost:8080



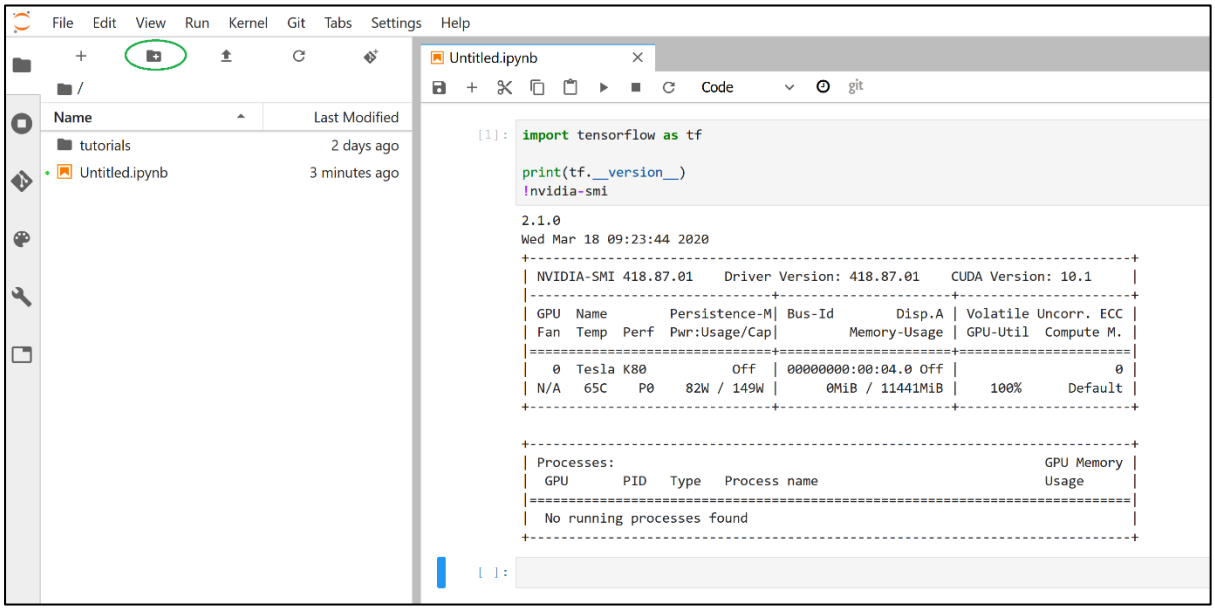
Once this line is executed you may see the following pop-up window asking if you trust the host. You should click Yes.



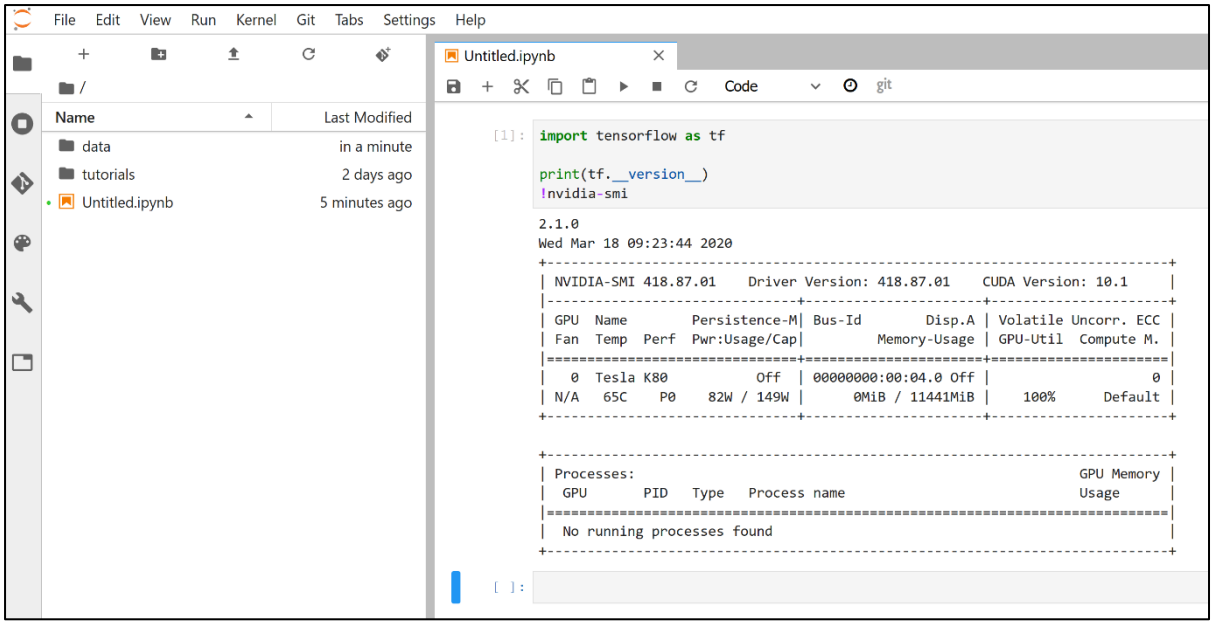
Once this line is executed and you click yes above, you should now be able to access the Jupyter notebook from a browser by going to the following page http://localhost:8080/ . You should see the page below. Notice I have entered some basic code in the first cell that confirms that I'm using TensorFlow 2.1 and the K80 GPU.



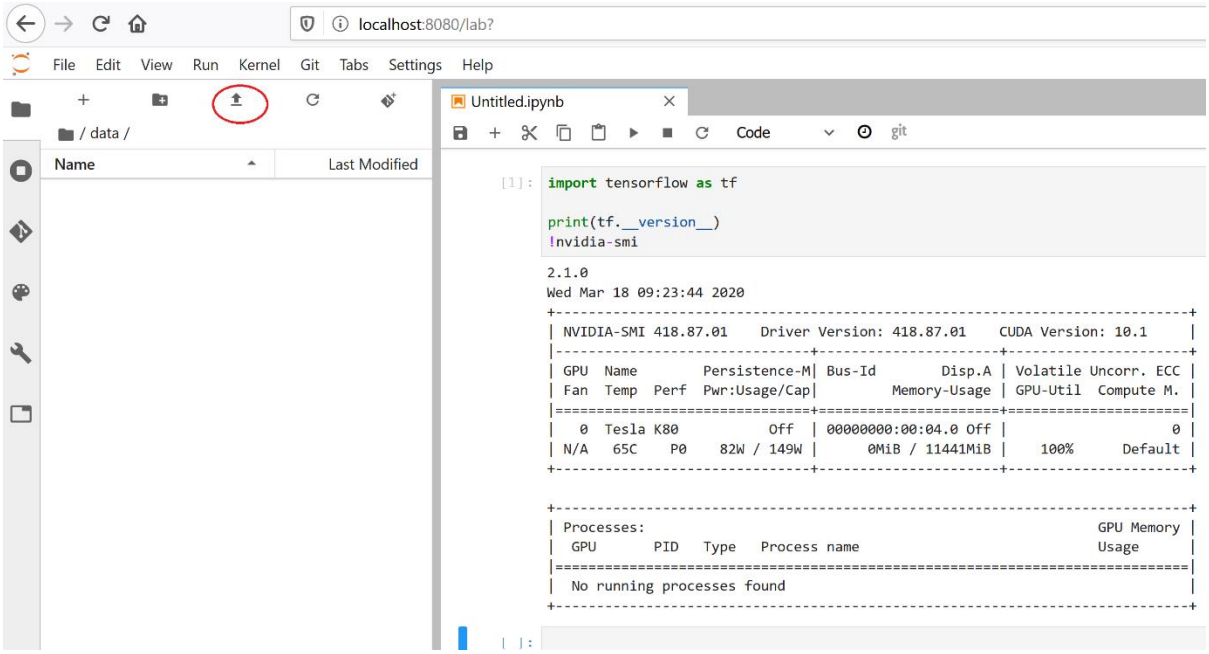
Over the following few lines I'm going to show how to upload and unzip a dataset. Click on the folder icon on the top left hand side of the screen highlighted in read above. This allows you to access the file system shown below. I'm going to create a folder called data by click the folder symbol highlighted in green below.



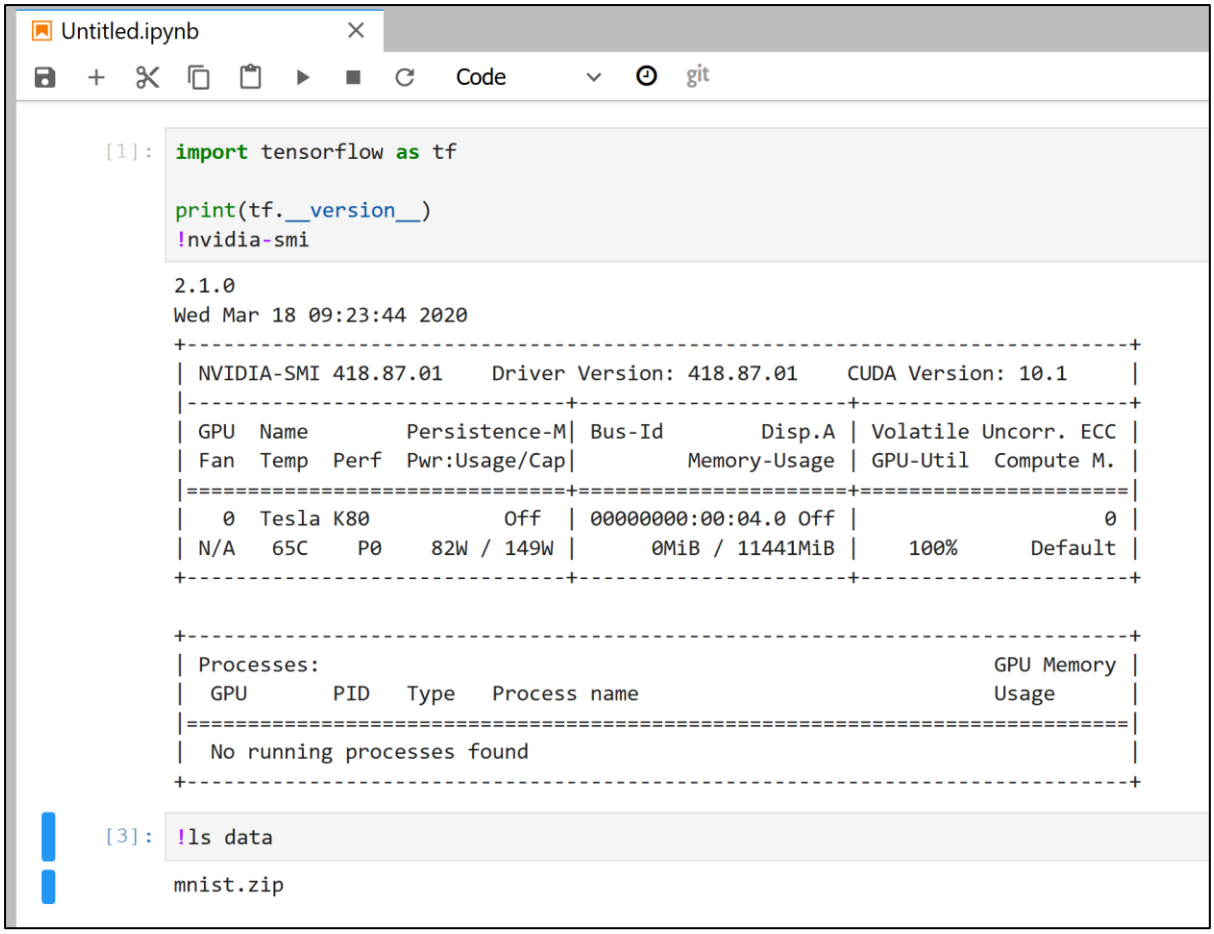
Once you have the data folder created, it will show below. Next double click on this folder to enter it.



Once we enter the data folder click the file upload button highlighted in red. You should upload the mnist.zip file (which you will find in the Colab unit on Canvas).



Once the upload of mnist.zip is completed we can then verify that it has been upload by checking the contents of the data folder using !ls command. See below. Once you type in !ls data into a cell and run you should see it print out the mnist.zip file.



Next we are going to upzip the file using using the following code.

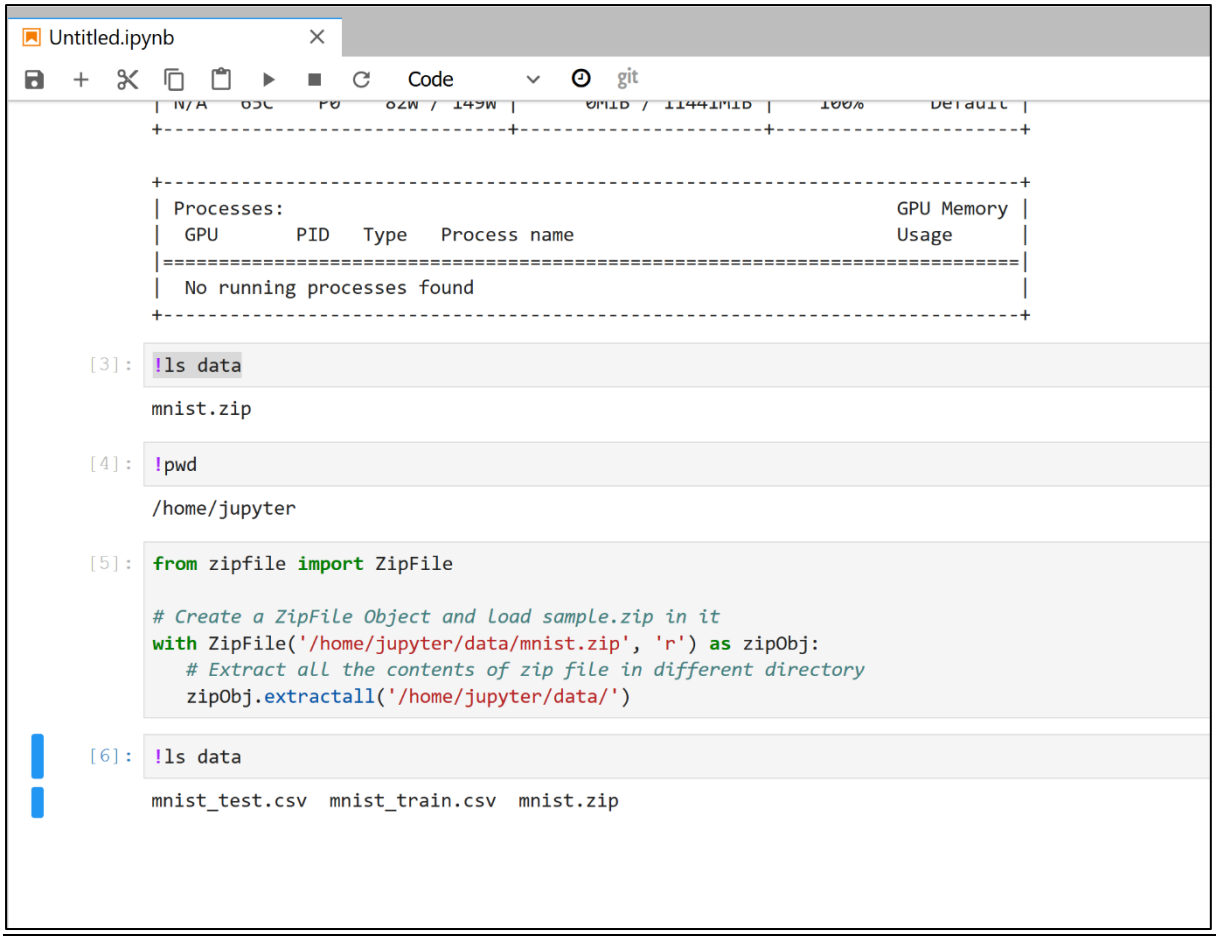
```
from zipfile import ZipFile


# Create a ZipFile Object and load sample.zip in it

with ZipFile('/home/jupyter/data/mnist.zip', 'r') as zipObj:

  # Extract all the contents of zip file in different directory
```

You can see below that when we run this code it unzips the original file and we now have a training and test file. Now that we have the training data we can build our model using the GPU.



Now that we have our data we can load using NumPy and build a model using the following code. You can find the full code for this example in this Colab notebook.

[8]:
```
# In this code we load the training and test data

import numpy as np
train = np.genfromtxt("/home/jupyter/data/mnist_train.csv", delimiter=",",skip_header=1)
test = np.genfromtxt("/home/jupyter/data/mnist_test.csv", delimiter=",",skip_header=1)

trainFeatures = train[:, 1:]
trainLabels = train[:,0]

testFeatures = test[:, 1:]
testLabels = test[:,0]
```

[9]:
```
# Normalize the input data
trainFeatures = trainFeatures/255
testFeatures = testFeatures /255
```

[10]:
```
import tensorflow as tf

# In the following we create a basic two layer network
# The first layer has 256 ReLu neurons. The second a softmax layer with 10 neurons
model = tf.keras.models.Sequential([

  tf.keras.layers.Dense(256, activation=tf.nn.relu, input_shape=(784,)),
  tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(trainFeatures, trainLabels, epochs=5, validation_split=0.1)
results = model.evaluate(testFeatures, testLabels)
print (results)
```

Remember to shut down your VM instance as shown in Section 6 when you are finished working.