

Guide to Running a Deep Learning VM in Google Cloud.

Contents

Part 1: (Obtaining Coupons)	1
Part 2: (Create a Project)	6
Part 3: (Enable APIs).....	8
Part 4: (Install and initialize the Cloud SDK).....	9
Part 5: (Apply for GPU Access).....	12
Part 6: (Deploying Deep Learning VM)	16
Part 7: (Connecting to Jupyter Notebook in VM instance)	20

IMPORTANT NOTE:

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

Part 1: (Obtaining Coupons)



Google Cloud Platform

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.

The screenshot shows a web page titled "Cloud Platform Education Grants" from the Google Cloud Platform. The page has a blue header and a white main content area. In the center, there is a form asking for personal information: First Name, Last Name, and School Email. The "School Email" field contains a dropdown menu with the option "@mycit.ie" highlighted by a red oval. Below the form, there is a note about contacting the course instructor if the domain is not listed. At the bottom, there is a "Submit" button and a link to "Privacy Policy".

Google Cloud Platform

Cloud Platform Education Grants

Use credits provided to you via the Google Cloud Platform Education Grants program to access Google Cloud Platform. Get what you need to build and run your apps, websites and services.

Thank you for your interest in Google Cloud Platform Education Grants. Please fill out the form below to receive a coupon code for credit to use on Google Cloud Platform.

First Name

Last Name

School Email @mycit.ie ▾

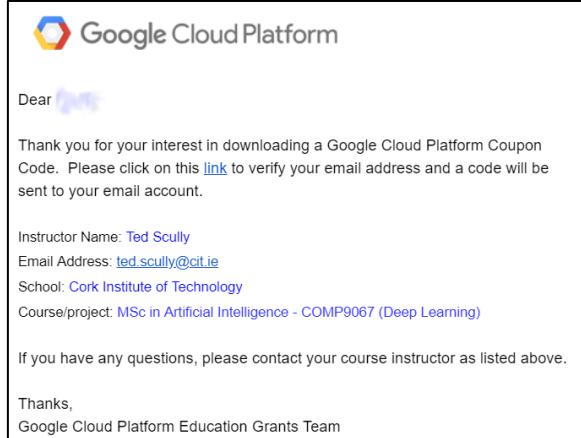
If you do not see your domain listed, please contact your course instructor: ted.scully@cit.ie

By clicking "Submit" below, you agree that we may share the following information with your educational institution and course instructor (ted.scully@cit.ie): (1) personal information that you provide to us on this form and (2) information regarding your use of the coupon and Google Cloud Platform products.

[Submit](#)

[Privacy Policy](#)

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



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.

≡ Google Cloud Platform 🔍

Education grants

Please enter the coupon code provided to you via the Google Cloud Platform Education Grants programme to receive credit for Google Cloud Platform. Get what you need to build and run your apps, websites and services.

Coupon code

Credit amount	Expiry date	Course
\$50.00	31 Jan 2020	MSc in Artificial Intelligence - COMP9067 (Deep Learning)

Terms of Service
 I agree to the [Google Cloud Platform Terms of Service](#), and the terms of service of [any applicable services and APIs](#).

Country of residence

I would like to receive periodic emails on news, product updates, and special offers from Google Cloud and Google Cloud Partners.
 Yes No

Google Cloud Platform education grants credits terms and conditions
By clicking "Accept and continue" below, you, on behalf of yourself and the organization you represent ("You") agree to these terms and conditions:

The credit is valid for Google Cloud Platform products and is subject to Your acceptance of the applicable Google Cloud Platform License Agreement and any other applicable terms of service. The credit is non-transferable and may not be sold or bartered. Unused credit expires on the date indicated on the media conveying the promotion code. The credit may be issued in increments as You use the credit over the period of time during which the credit is valid. Offer void where prohibited by law.

You represent that you are accepting the promotional credit on behalf of your educational institution and the credit can only be used on behalf of the educational entity and not for your personal use. You represent, on behalf of such educational entity, that (i) You are authorized to accept this credit; (ii) the credit is consistent with all applicable laws and regulations, including relevant ethics rules and laws; and (iii) the provision of credits will not negatively impact Google's current or future ability to do business with such educational entity.

You agree that we may share the following information with your educational institution and course instructor: (1) personal information that you provide to us during the coupon redemption process and (2) information regarding your use of the coupon and Google Cloud Platform products.

[Accept and continue](#) [Clear](#)

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.

≡ Google Cloud Platform 🔍 ✉️ ⚡ 🌐 🔍

Billing OVERVIEW Manage billing accounts RENAME BILLING ACCOUNT HIDE INFO PANEL

<ul style="list-style-type: none"> Overview Budgets & alerts Billing export Reports 	<p>Billing account ID: XXXXXXXXXX Organisation: mycit.ie</p> <p>Credits</p> \$50.00 <small>Credits remaining</small> <small>Out of \$50.00</small> 336 <small>Days remaining</small> <small>Ends 1 Feb 2020</small> <p>Projects linked to this billing account <small>There are no projects linked to this billing account.</small></p>	<p>MSc in Artificial Intelligence - COMP9067 (Deep Learning)</p> <p>PERMISSIONS</p> <p>Add members <input type="text"/> <input type="button" value="Select a role"/> <input type="button" value="Add"/></p> <p>Search members <input style="width: 100%; height: 20px; border: 1px solid #ccc; border-radius: 5px; margin-bottom: 5px;" type="text"/></p> <p>Billing Account Administrator (1 member) <small>Authorised to see and manage all aspects of billing accounts.</small></p>
---	---	--

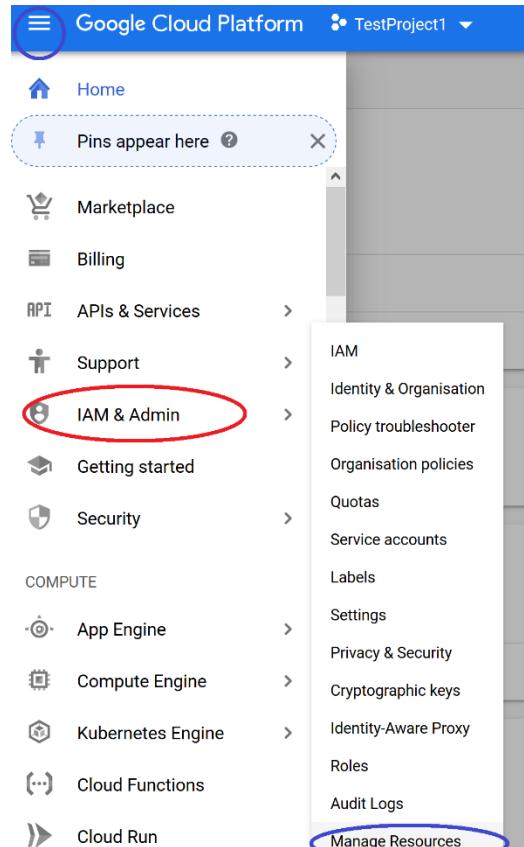
If you click on the “Google Cloud Platform” on the top left hand corner of the page it should take you to a page similar to that shown below.

The screenshot shows the Google Cloud Platform homepage. The navigation bar on the left includes links for Home, Getting started, Marketplace, APIs & Services, IAM & admin, Security, Billing, Support, COMPUTE, and App Engine. The main area features a "Welcome, Ted" message and a "Get started with Google Cloud Platform" button. Below this is a "TOUR CONSOLE" button. The "Top Products" section displays four services: Compute Engine (scalable, high-performance virtual machines), Cloud Storage (powerful, simple and cost-effective object storage service), Cloud SQL (fully-managed MySQL/PostgreSQL database service), and App Engine (platform for web and mobile apps that scale automatically).

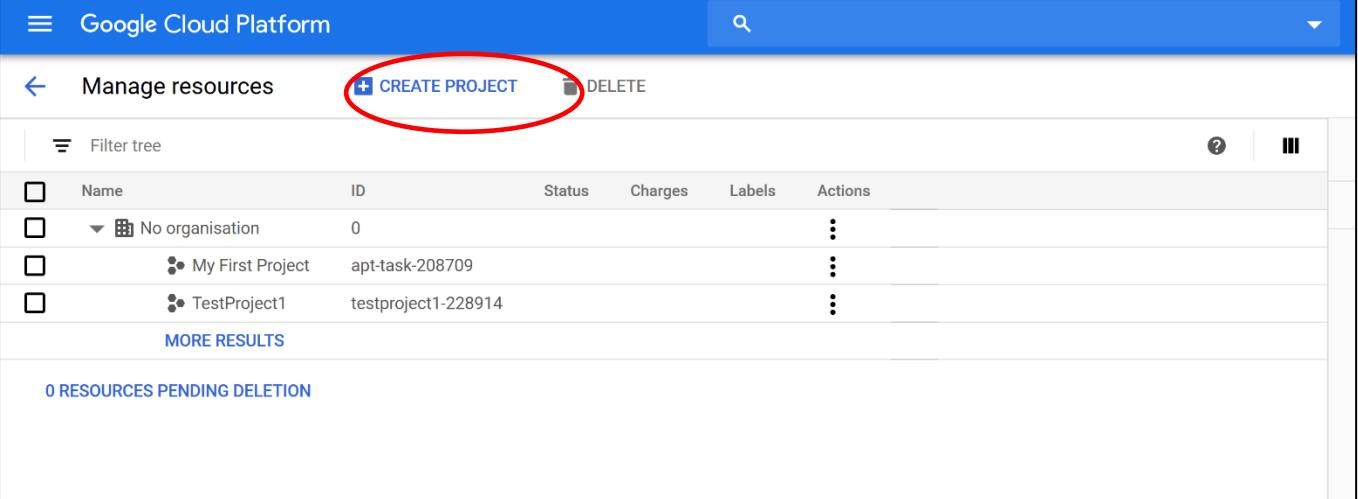
Product	Description
Compute Engine	Scalable, high-performance virtual machines
Cloud Storage	A powerful, simple and cost effective object storage service
Cloud SQL	A fully-managed MySQL/PostgreSQL database service
App Engine	A platform to build web and mobile apps that scale automatically

Part 2: (Create a Project)

Go to 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).

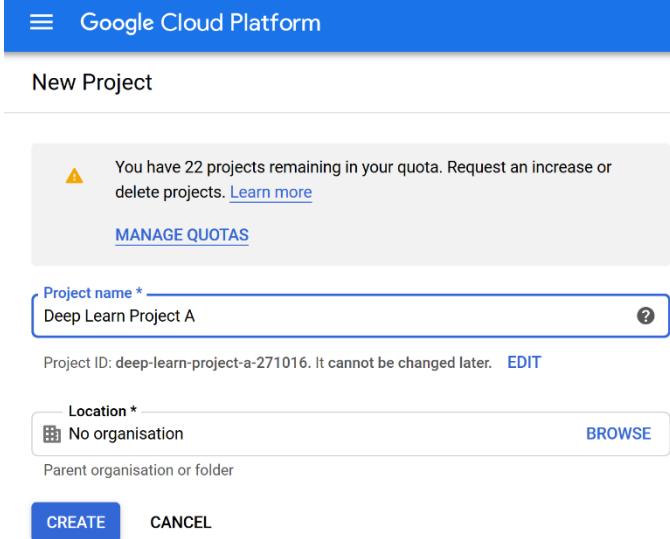


The screenshot shows the 'Manage resources' page in the Google Cloud Platform. At the top, there is a blue header bar with the 'Google Cloud Platform' logo and a search bar. Below the header, the title 'Manage resources' is displayed with a back arrow icon. To the right of the title are two buttons: '+ CREATE PROJECT' (highlighted with a red oval) and 'DELETE'. The main area is a table with columns: Name, ID, Status, Charges, Labels, and Actions. There are three rows listed:

Name	ID	Status	Charges	Labels	Actions
No organisation	0				⋮
My First Project	apt-task-208709				⋮
TestProject1	testproject1-228914				⋮

Below the table, there is a 'MORE RESULTS' link and a message '0 RESOURCES PENDING DELETION'.

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.



The screenshot shows the 'New Project' dialog. At the top, there is a blue header bar with the 'Google Cloud Platform' logo. Below the header, the title 'New Project' is displayed. The main form has several fields:

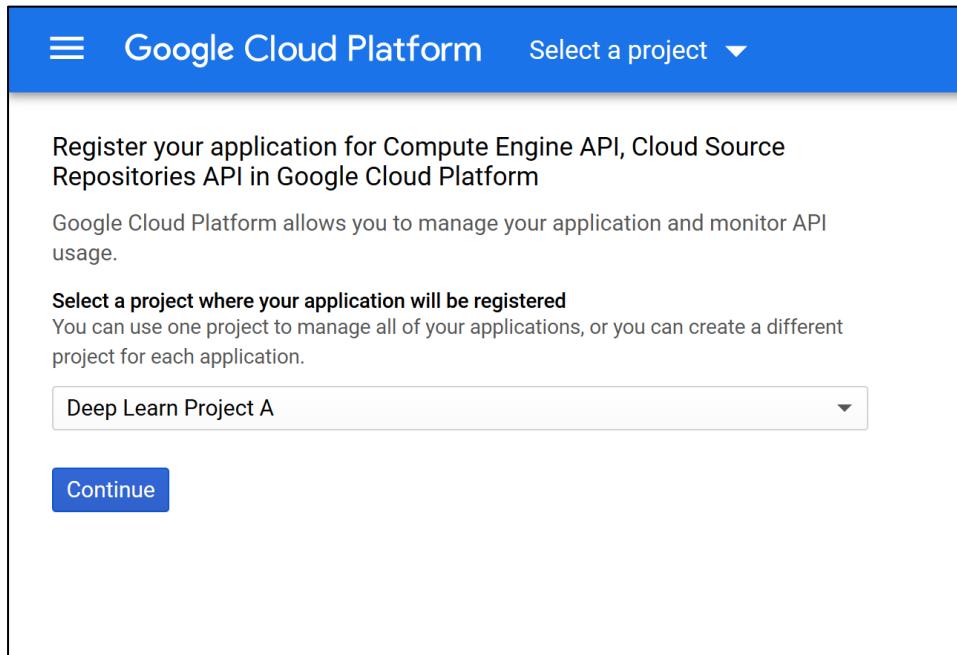
- A warning message: '⚠ You have 22 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)' with a 'MANAGE QUOTAS' link.
- A 'Project name *' input field containing 'Deep Learn Project A'.
- A note below the project name: 'Project ID: deep-learn-project-a-271016. It cannot be changed later. [EDIT](#)'.
- A 'Location *' dropdown menu showing 'No organisation' with a 'BROWSE' button.
- A note below the location: 'Parent organisation or folder'.
- Action buttons at the bottom: 'CREATE' (highlighted with a blue oval) and 'CANCEL'.

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)

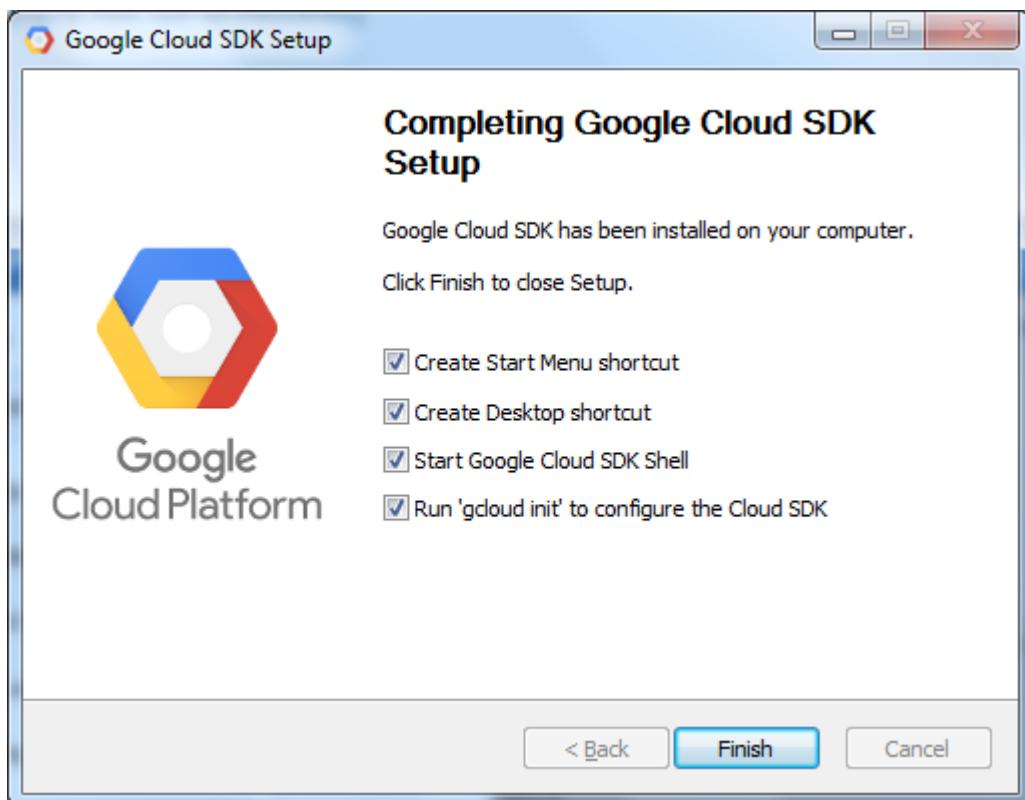
As most of you will be using Windows, Part 4 describes the Windows installation process. Google Cloud SDK is a set of tools that you can use to manage resources and applications hosted on Google Cloud Platform.

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

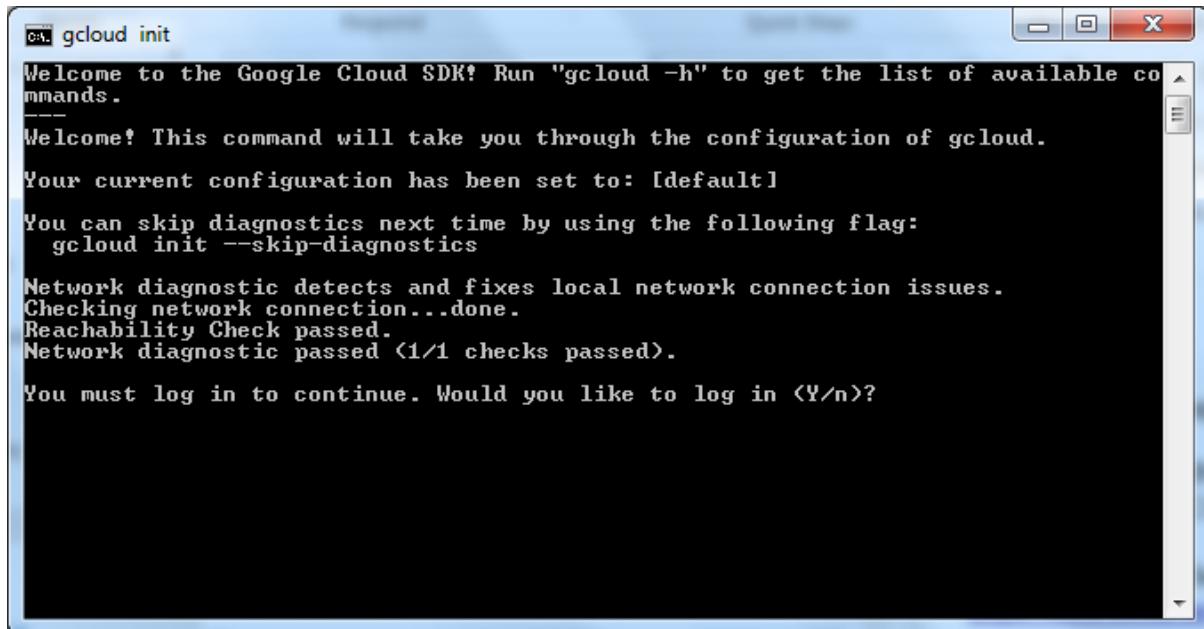
To install the Cloud SDK please follow the first three steps listed [here](#). (When you go to this link you will notice there is also tabs available for Linux, Mac, etc).

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.



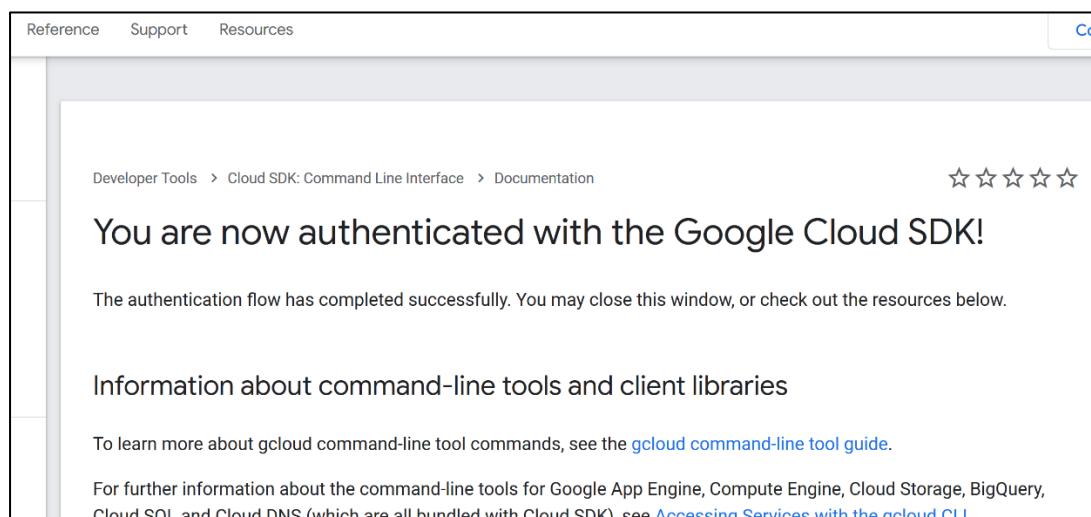
Next the installer starts a terminal window and may ask you to log in.



A screenshot of a terminal window titled "gcloud init". The window contains the following text:

```
Welcome to the Google Cloud SDK! Run "gcloud -h" to get the list of available commands.  
--  
Welcome! This command will take you through the configuration of gcloud.  
Your current configuration has been set to: [default]  
You can skip diagnostics next time by using the following flag:  
  gcloud init --skip-diagnostics  
Network diagnostic detects and fixes local network connection issues.  
Checking network connection...done.  
Reachability Check passed.  
Network diagnostic passed (1/1 checks passed).  
You must log in to continue. Would you like to log in (Y/n)?
```

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.



A screenshot of a web browser window. The top navigation bar includes "Reference", "Support", "Resources", and a search bar. The main content area shows the following:

Developer Tools > Cloud SDK: Command Line Interface > Documentation

You are now authenticated with the Google Cloud SDK!

The authentication flow has completed successfully. You may close this window, or check out the resources below.

Information about command-line tools and client libraries

To learn more about gcloud command-line tool commands, see the [gcloud command-line tool guide](#).

For further information about the command-line tools for Google App Engine, Compute Engine, Cloud Storage, BigQuery, Cloud SQL and Cloud DNS (which are all bundled with Cloud SDK), see [Accessing Services with the gcloud CLI](#).

You should notice 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]

```
C:\Windows\SYSTEM32\cmd.exe - gcloud init

Your current configuration has been set to: [default]

You can skip diagnostics next time by using the following flag:
gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.
Reachability Check passed.
Network diagnostic passed (1/1 checks passed).

You must log in to continue. Would you like to log in (Y/n)? Y

Your browser has been opened to visit:

https://accounts.google.com/o/oauth2/auth?code_challenge=phSqhhtnnLwJMe_gN1EWoeOzauONrpXqbnx4FF_PgMM&prompt=select_account&code_challenge_method=S256&access_type=offline&redirect_uri=http%3A%2F%2Flocalhost%3A8085%2F&response_type=code&client_id=32555940559.apps.googleusercontent.com&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth

You are logged in as: [tedscully@gmail.com].

Pick cloud project to use:
[1] apt-task-208709
[2] deep-learn-project-a-271016
[3] Create a new project
Please enter numeric choice or text value (must exactly match list item): 2
```

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).

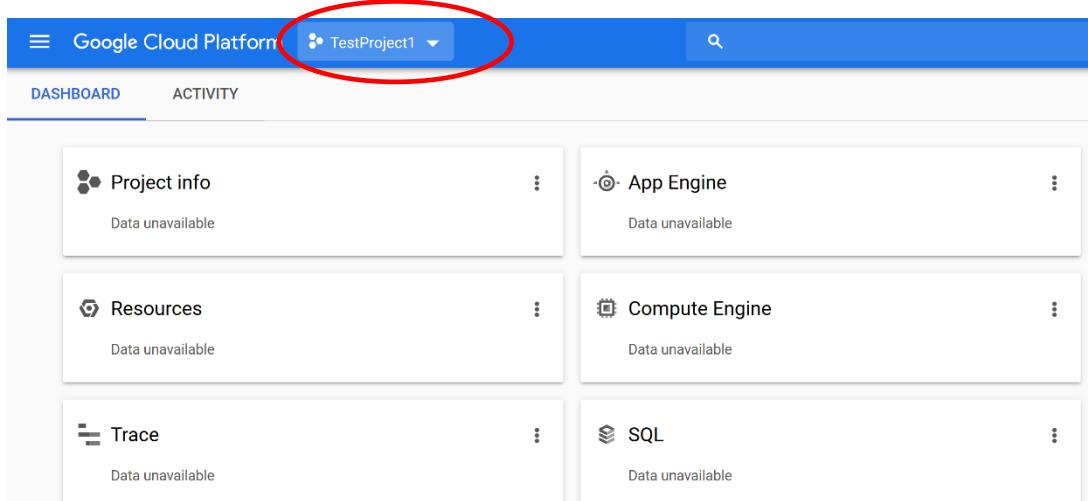
```
Google Cloud SDK Shell - gcloud init

[25] europe-west2-a
[26] asia-east1-b
[27] asia-east1-a
[28] asia-east1-c
[29] asia-southeast1-b
[30] asia-southeast1-a
[31] asia-southeast1-c
[32] asia-northeast1-b
[33] asia-northeast1-c
[34] asia-northeast1-a
[35] asia-south1-c
[36] asia-south1-b
[37] asia-south1-a
[38] australia-southeast1-b
[39] australia-southeast1-c
[40] australia-southeast1-a
[41] southamerica-east1-b
[42] southamerica-east1-c
[43] southamerica-east1-a
[44] asia-east2-a
[45] asia-east2-b
[46] asia-east2-c
[47] asia-northeast2-a
[48] asia-northeast2-b
[49] asia-northeast2-c
[50] asia-northeast3-a
Did not print [18] options.
Too many options [68]. Enter "list" at prompt to print choices fully.
Please enter numeric choice or text value (must exactly match list item): 18
```

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.

The screenshot shows the Google Cloud Platform navigation bar at the top with the title "Google Cloud Platform" and a project dropdown "Deep Learn Project A". Below the navigation bar is a sidebar with various menu items. One of the items, "APIs & Services", has a dropdown arrow indicating it has sub-options. The sub-menu for "APIs & Services" includes "IAM", "Identity & Organisation", "Policy troubleshooter", "Organisation policies", and "Quotas". The "Quotas" option is circled with a red marker. Other menu items in the sidebar include "Marketplace", "Billing", "Support", "Getting started", "Security", and sections for "COMPUTE" (App Engine, Compute Engine, Kubernetes Engine) and "Data" (Bigtable, Dataflow, Pub/Sub, Cloud Storage, Cloud ML Engine, Cloud Functions, Cloud Run).

This will take you to the following page.

The screenshot shows the "Quotas" page within the "IAM & Admin" section of the Google Cloud Platform. On the left is a sidebar with links to "Identity & Organisation", "Policy troubleshooter", "Organisation policies", "Quotas" (which is selected and highlighted in blue), "Service accounts", "Labels", "Settings", "Privacy & Security", "Cryptographic keys", "Identity-Aware Proxy", and "Roles". The main content area is titled "Quotas" and contains a header with "EDIT QUOTAS" and filter dropdowns for "Quota type" (set to "All quotas"), "Service" (set to "All services"), "Metric" (set to "All metrics"), and "Location" (set to "All locations"). Below the header is a table with columns: "Service", "Location", "Current Usage", "7-Day Peak Usage", and "Limit". The table lists several Compute Engine API quotas, all of which have a value of 0 for both current usage and 7-day peak usage, and an "Unlimited" or specific limit listed in the "Limit" column. The quotas listed are: "Compute Engine API Queries per day", "Compute Engine API Queries per 100 seconds", "Compute Engine API Read requests per 100 seconds", "Compute Engine API List requests per 100 seconds", "Compute Engine API Operation read requests per 100 seconds", "Compute Engine API Heavy-weight read requests per 100 seconds", "Compute Engine API Heavy-weight mutation requests per 100 seconds", and "Compute Engine API Instance SimulateMaintenanceEvent requests per day".

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.

The screenshot shows the Google Cloud Platform IAM & Admin Quotas page for project "Deep Learn Project A". The "Quotas" section is selected, and the "EDIT QUOTAS" button is highlighted with a purple oval. The "Metric" dropdown is highlighted with a green oval, showing "GPUs (all regions)". The "Location" dropdown is highlighted with a blue oval, showing "All locations". In the quota table, the "Service" column has a checked checkbox, and the "Compute Engine API" row is highlighted with a red oval. The "Location" column for this row shows "Global". The "Current Usage" and "7-Day Peak Usage" columns are present, and the "Limit" column shows a value of 0.

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.

The screenshot shows the Google Cloud Platform Quotas page for 'Deep Learn Project A'. On the left, there's a table with columns: Quota type, Service, Metric, and Location. The filters are set to 'All quotas', 'All services', 'GPUs (all regions)', and 'All locations'. Under the 'Service' column, 'Compute Engine API' and 'GPUs (all regions)' are selected. On the right, a modal window titled 'Compute Engine API' shows a quota request for 'GPUs (all regions)'. It has a 'New quota limit' field containing '1', a 'Request description' field with 'Deep Learning Module (Education)', and buttons for 'Done' and '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).

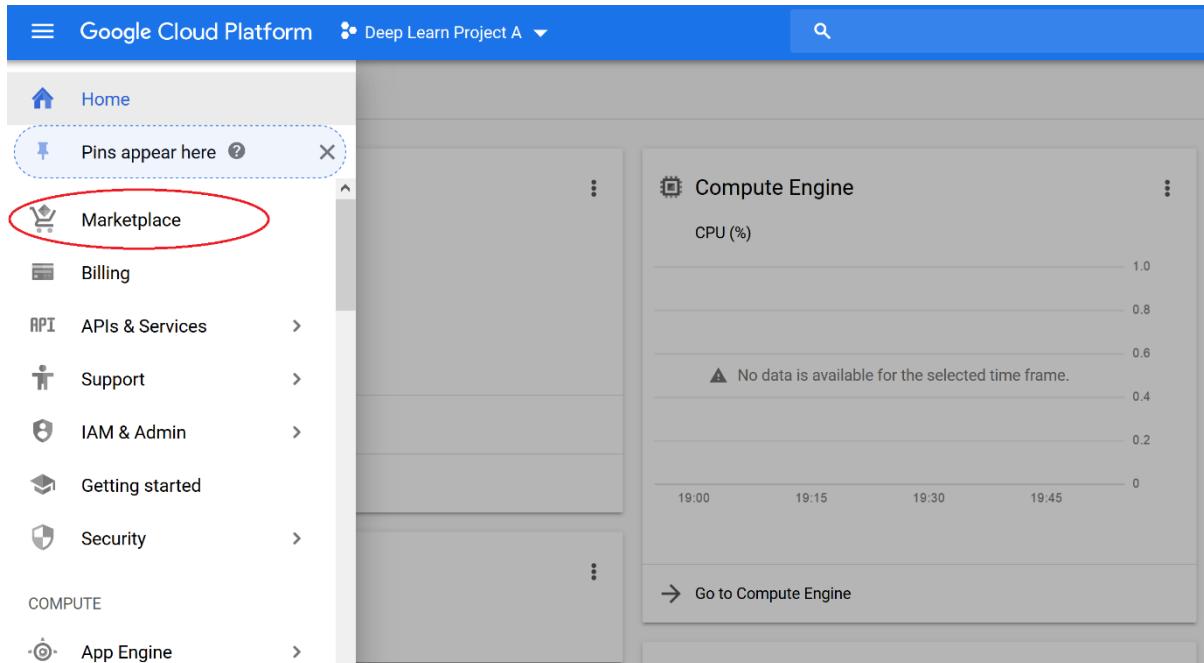
The screenshot shows the Google Cloud Platform Quotas page for 'Deep Learn Project A'. The interface is similar to the previous one, but the 'Compute Engine API' row in the table now has an unchecked checkbox next to 'Service'. On the right, a modal window titled 'Edit quotas' shows a message: 'Thank you for submitting Case # (ID:22481577) to Google Cloud Platform support for the following quota: Change GPUs (all regions) from 0 to 1. Your request is being processed and you should receive an email confirmation for your request. Should you need further assistance, you can respond to that email.'

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).

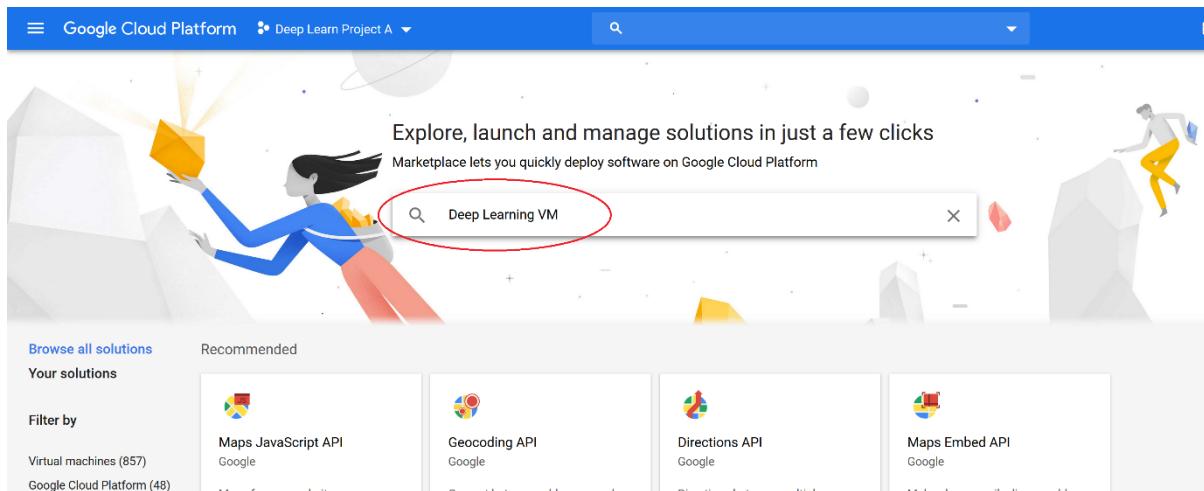
You will need to be granted GPU access before moving to the next step.

Part 6: (Deploying Deep Learning VM)

You now need to configure and launch a deep learning VM. Go to 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.

Marketplace

“Deep Learning VM”

Filter by

CATEGORY

- Analytics (11)
- Big data (1)
- Compute (12)
- Databases (1)
- Developer stacks (1)
- Developer tools (1)

24 results

Image	Name	Description
	Deep Learning VM	Google Click to Deploy Intel(R) optimized and GPU-ready machine learning frameworks
	NVIDIA GPU Cloud Image for Deep Learning, Data Science, and HPC	NVIDIA Image Optimized for GPU-Accelerated Containers

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”

Deep Learning VM

Deep Learning VM (Google Click to Deploy)

Estimated costs: \$295.20/month

Intel(R) optimized and GPU-ready machine learning frameworks

LAUNCH 1 PAST DEPLOYMENT

Runs on

Google Compute Engine

Type

Overview

Deploy a Compute Engine instance with your favorite machine learning framework, Intel(R) optimized for GCE and configured to support common GPU workloads out of the box. This deployment automates out the hassle of

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.

Deployment name: tensorflow-1

Zone: europe-west1-d

Machine type: 2 vCPUs, 13 GB memory

GPU: 1 NVIDIA Tesla K80

Software: Operating system: Debian (9)

Documentation: Official Documentation, StackOverflow: Deep Learning VM, Google Group: Deep Learning VM

Terms of Service: By deploying the software or accessing the service you are agreeing to comply with the GCP Marketplace terms of service and the terms of applicable open source software licenses bundled with the software or service. Please review these terms and licenses carefully for details about any obligations you may have related to the software or service. To the limited extent an open source software license related to the software or service expressly supersedes the GCP Marketplace Terms of Service, that open source software license governs your use of that software or service.

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.

Number of GPUs: 1

GPU type: NVIDIA Tesla K80

Framework: TensorFlow Enterprise 2.1 (CUDA 10.1)

Access to the Jupyter Lab:

- Install NVIDIA GPU driver automatically on first startup? ⓘ
- Enable access to JupyterLab via URL instead of SSH. (Beta) ⓘ

StackOverflow: Deep Learning VM, Google Group: Deep Learning VM

Terms of Service: By deploying the software or accessing the service you are agreeing to comply with the GCP Marketplace terms of service and the terms of applicable open source software licenses bundled with the software or service. Please review these terms and licenses carefully for details about any obligations you may have related to the software or service. To the limited extent an open source software license related to the software or service expressly supersedes the GCP Marketplace Terms of Service, that open source software license governs your use of that software or service.

By using this product, you understand that certain account and usage information may be shared with Google Click to Deploy for the purposes of sales attribution, performance analysis, and support. ⓘ

Google is providing this software or service "as-is" and will not perform any ongoing maintenance. Ongoing upgrades and maintenance are your responsibility.

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.

The screenshot shows the Google Cloud Platform Deployment Manager interface. On the left, under 'Deployments', there is a single entry for 'tensorflow-1'. The main pane displays the details of this deployment. A warning message states 'tensorflow-1 has been deployed, but contains warnings' with a 'VIEW DETAILS' button. Below this, the 'Overview - tensorflow-1' section shows a tree structure with nodes like 'tensorflow', 'tensorflow-vm', 'software-status', and 'tensorflow-1-software'. To the right, a separate panel titled 'tensorflow' provides more information about the 'Deep Learning VM' instance. It shows the instance name as 'tensorflow-1-vm', located in 'europe-west1-d' zone with 'n1-highmem-2' machine type. A 'MORE ABOUT THE SOFTWARE' section is partially visible. At the bottom, there's a 'Get started with Deep Learning VM' section with an 'SSH' button and a 'Suggested next steps' list.

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

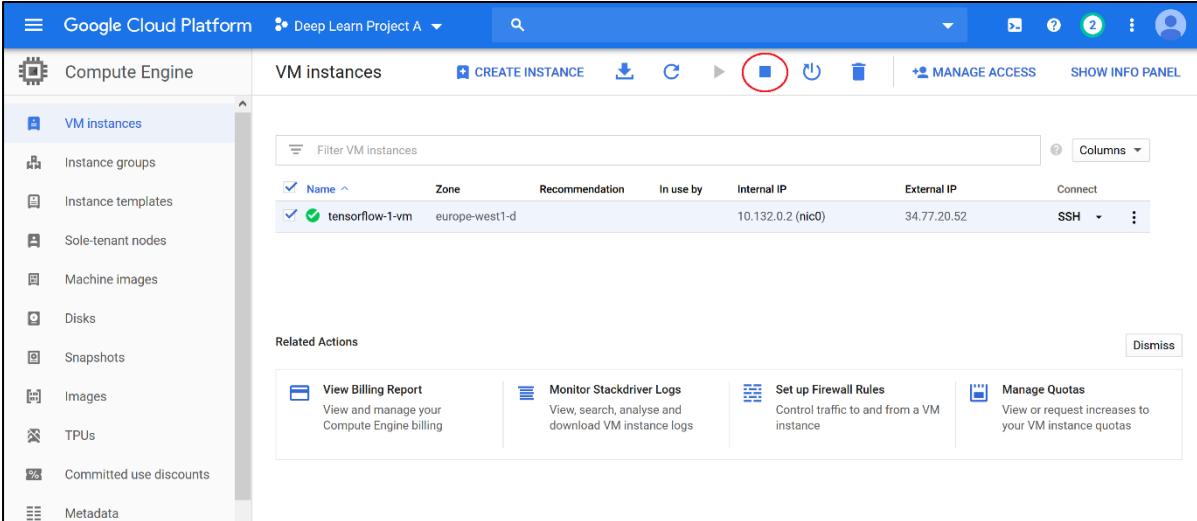
To do this we need to go back to Google Cloud Platform homepage (click Google Cloud Platform on top left of screen) as shown below. Then click on “Go to Compute Engine” link highlighted in red. Alternatively, you can also enter “Compute Engine” in the search box at the top of the page.

The screenshot shows the Google Cloud Platform homepage for 'Deep Learn Project A'. The 'DASHBOARD' tab is selected. On the left, there are sections for 'Project info', 'Resources' (with 'Compute Engine' listed), and 'Trace'. The 'Compute Engine' section contains a chart for CPU usage over time and a link '→ Go to Compute Engine'. This link is circled in red. To the right, there are other sections: 'Compute Engine' (with a chart showing 0.0 CPU usage), 'Google Cloud Platform status' (showing 'All services normal'), 'Billing' (showing 'Estimated charges USD \$0.00'), and 'Error Reporting'.

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.



The screenshot shows the Google Cloud Platform Compute Engine VM instances page. On the left, there's a sidebar with options like VM instances, Instance groups, Instance templates, Sole-tenant nodes, Machine images, Disks, Snapshots, Images, TPUs, Committed use discounts, and Metadata. The main area shows a table of VM instances. The first row has a checkbox, Name (tensorflow-1-vm), Zone (europe-west1-d), Recommendation, In use by, Internal IP (10.132.0.2 (nic0)), External IP (34.77.20.52), and a Connect button. Below the table, there's a 'Related Actions' section with four buttons: View Billing Report, Monitor Stackdriver Logs, Set up Firewall Rules, and Manage Quotas. The 'Stop' button for the tensorflow-1-vm instance is highlighted with a red circle.

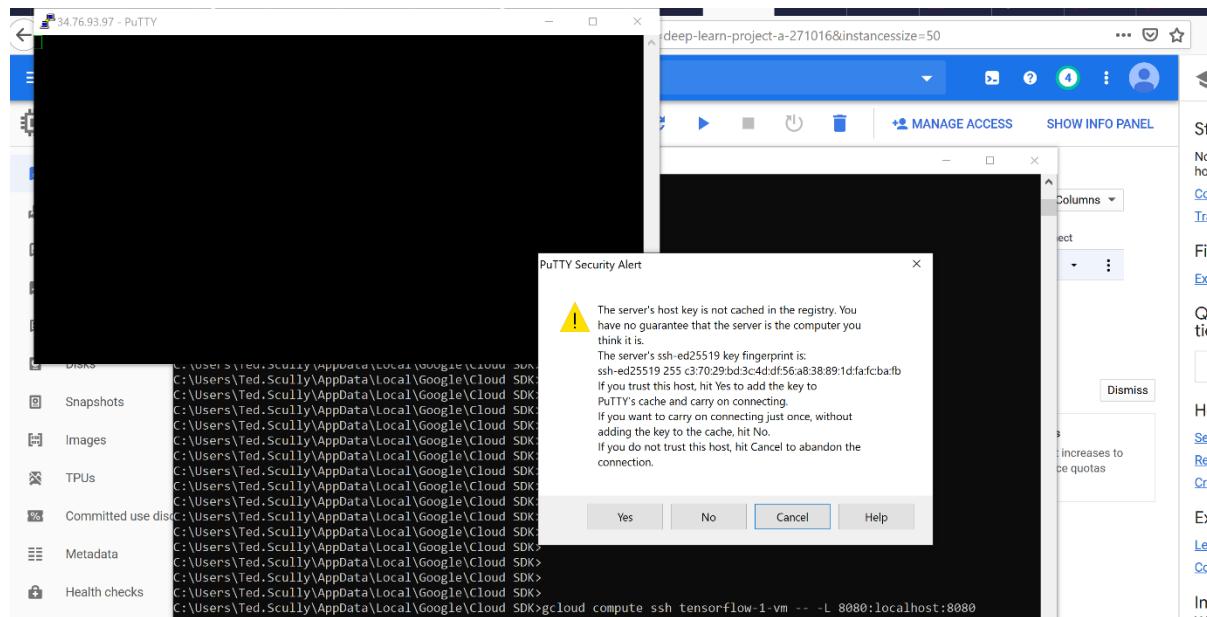
Part 7: (Connecting to Jupyter Notebook in VM instance)

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.

The screenshot shows the Jupyter Notebook interface. On the left, there are two sections: 'TERMINAL SESSIONS' and 'KERNEL SESSIONS'. The 'KERNEL SESSIONS' section contains a single tab labeled 'Untitled.ipynb'. On the right, the main area displays a terminal window with the following content:

```
[1]: import tensorflow as tf
print(tf.__version__)
!nvidia-smi

2.1.0
Wed Mar 18 09:23:44 2020
+-----+
| NVIDIA-SMI 418.87.01    Driver Version: 418.87.01    CUDA Version: 10.1 |
| Persistence-M| Bus-Id      Disp.A | Volatile Uncorr. ECC |
| Fan Temp Perf Pwr:Usage/Cap| Memory-Usage | GPU-Util Compute M. |
+-----+
| 0 Tesla K80          Off | 00000000:00:04.0 Off |          0 |
| N/A 65C   P0  82W / 149W |     0MiB / 11441MiB |  100% Default |
+-----+
+-----+
| Processes:                               GPU Memory |
| GPU PID Type Process name               Usage |
| ====== ===== ===== ===== |
| No running processes found               |
+-----+
```

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 red 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.

The screenshot shows the Jupyter Notebook interface. On the left, the file system pane shows a new folder named 'data' has been created. The 'KERNEL SESSIONS' section still contains the 'Untitled.ipynb' tab. The terminal window content remains the same as in the previous screenshot.

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

The screenshot shows the Jupyter Notebook interface. On the left is a file tree with a 'data' folder, a 'tutorials' folder, and an 'Untitled.ipynb' file. The main area contains a code cell with the following content:

```
[1]: import tensorflow as tf  
print(tf.__version__)  
!nvidia-smi
```

Output from the cell:

```
2.1.0  
Wed Mar 18 09:23:44 2020  
+-----+  
| NVIDIA-SMI 418.87.01 Driver Version: 418.87.01 CUDA Version: 10.1 |  
+-----+  
| GPU Name Persistence-M| Bus-Id Disp.A | Volatile Uncorr. ECC |  
| Fan Temp Perf Pwr:Usage/Cap| Memory-Usage | GPU-Util Compute M. |  
+-----+  
| 0 Tesla K80 Off | 00000000:00:04.0 Off | 0 |  
| N/A 65C P0 82W / 149W | 0MiB / 11441MiB | 100% Default |  
+-----+  
  
+-----+  
| Processes: GPU Memory |  
| GPU PID Type Process name Usage |  
+-----+  
| No running processes found |  
+-----+
```

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).

The screenshot shows the Jupyter Notebook interface. A red circle highlights the file upload button in the toolbar above the file tree. The file tree shows a 'data' folder. The main area contains a code cell with the same content as the previous screenshot:

```
[1]: import tensorflow as tf  
print(tf.__version__)  
!nvidia-smi
```

Output from the cell:

```
2.1.0  
Wed Mar 18 09:23:44 2020  
+-----+  
| NVIDIA-SMI 418.87.01 Driver Version: 418.87.01 CUDA Version: 10.1 |  
+-----+  
| GPU Name Persistence-M| Bus-Id Disp.A | Volatile Uncorr. ECC |  
| Fan Temp Perf Pwr:Usage/Cap| Memory-Usage | GPU-Util Compute M. |  
+-----+  
| 0 Tesla K80 Off | 00000000:00:04.0 Off | 0 |  
| N/A 65C P0 82W / 149W | 0MiB / 11441MiB | 100% Default |  
+-----+  
  
+-----+  
| Processes: GPU Memory |  
| GPU PID Type Process name Usage |  
+-----+  
| No running processes found |  
+-----+
```

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.

The screenshot shows a Jupyter Notebook window titled "Untitled.ipynb". The code cell [1] contains Python imports for TensorFlow and prints the version, followed by a command to check NVIDIA-SMI. The output shows TensorFlow version 2.1.0 and NVIDIA-SMI information for a Tesla K80 GPU. The code cell [3] lists the contents of a "data" directory, showing a single file named "mnist.zip".

```
[1]: import tensorflow as tf  
print(tf.__version__)  
!nvidia-smi  
  
2.1.0  
Wed Mar 18 09:23:44 2020  
+-----+  
| NVIDIA-SMI 418.87.01      Driver Version: 418.87.01      CUDA Version: 10.1 |  
+-----+  
| GPU  Name      Persistence-M| Bus-Id     Disp.A  | Volatile Uncorr. ECC | | | | |
| Fan  Temp  Perf  Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |  
|=====|=====|=====|=====|=====|=====|=====|  
| 0  Tesla K80          Off  | 00000000:00:04.0 Off |                0 |  
| N/A   65C   P0    82W / 149W |      0MiB / 11441MiB |    100%     Default |  
+-----+  
  
+-----+  
| Processes:                               GPU Memory |  
| GPU     PID  Type  Process name        Usage |  
|=====|=====|=====|=====|  
| No running processes found |  
+-----+  
  
[3]: !ls data  
mnist.zip
```

Next we are going to unzip 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.

The screenshot shows a Jupyter Notebook interface with the title bar 'Untitled.ipynb'. The notebook contains the following code cells:

```
[3]: !ls data
mnist.zip

[4]: !pwd
/home/jupyter

[5]: 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
    zipObj.extractall('/home/jupyter/data/')

[6]: !ls data
mnist_test.csv  mnist_train.csv  mnist.zip
```

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.**