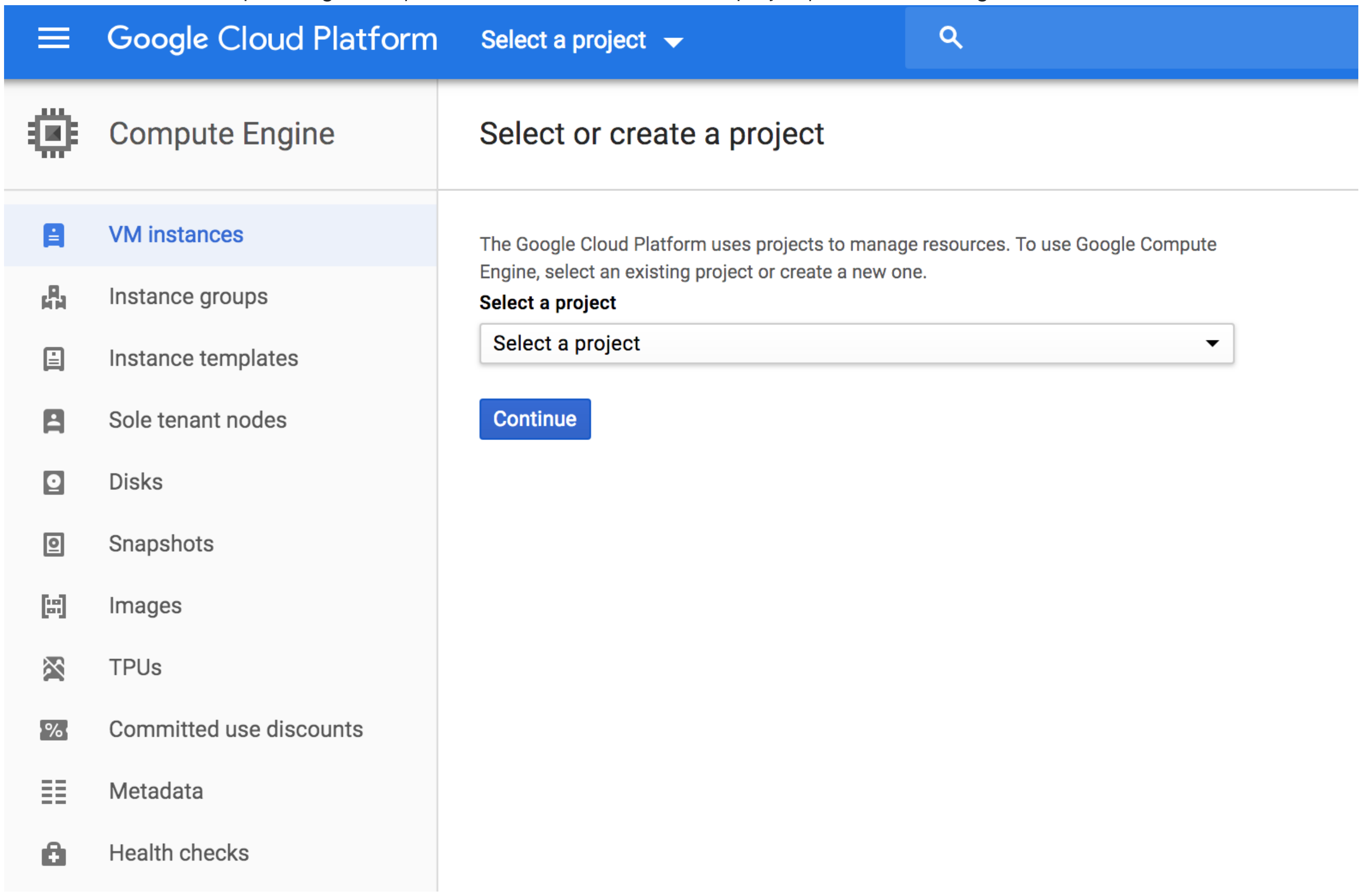
Google Cloud Coupons

Follow the instructions given to you earlier in order to request a $50 Google Cloud Platform coupon. You will be asked to provide your school email address and name. An email will be sent to you to confirm these details before a coupon is sent to you. WARNING: do not redeem your coupon with your Princeton accounts (this is especially true for graduate students for whom G-Suite account privileges are limited). Instead, even though you receive your coupon in your Princeton email, you should redeem it with a personal Google account. You can only request ONE code per unique email address. These coupons can only be used for your COS429 projects. Please do not share the URL with anyone not enrolled in this class.

Setting up a VM Instance

1. Follow the link given in the email (when you are logged into your personal Google account) in order to access the Billing page of the Google Cloud Platform. Check back here at any point in order to know how much you have left in compute resources.
2. Click on the three line icon in the top left hand corner in order to access the "Navigation Menu". If you have never used Google Cloud before, you may wish to read some of the resources provided under the "Getting Started" tab. The guide to using the Compute Engine may be especially helpful.
3. Scroll down to the Compute Engine tab, and click on this. Create a new project; call it something like "COS429":



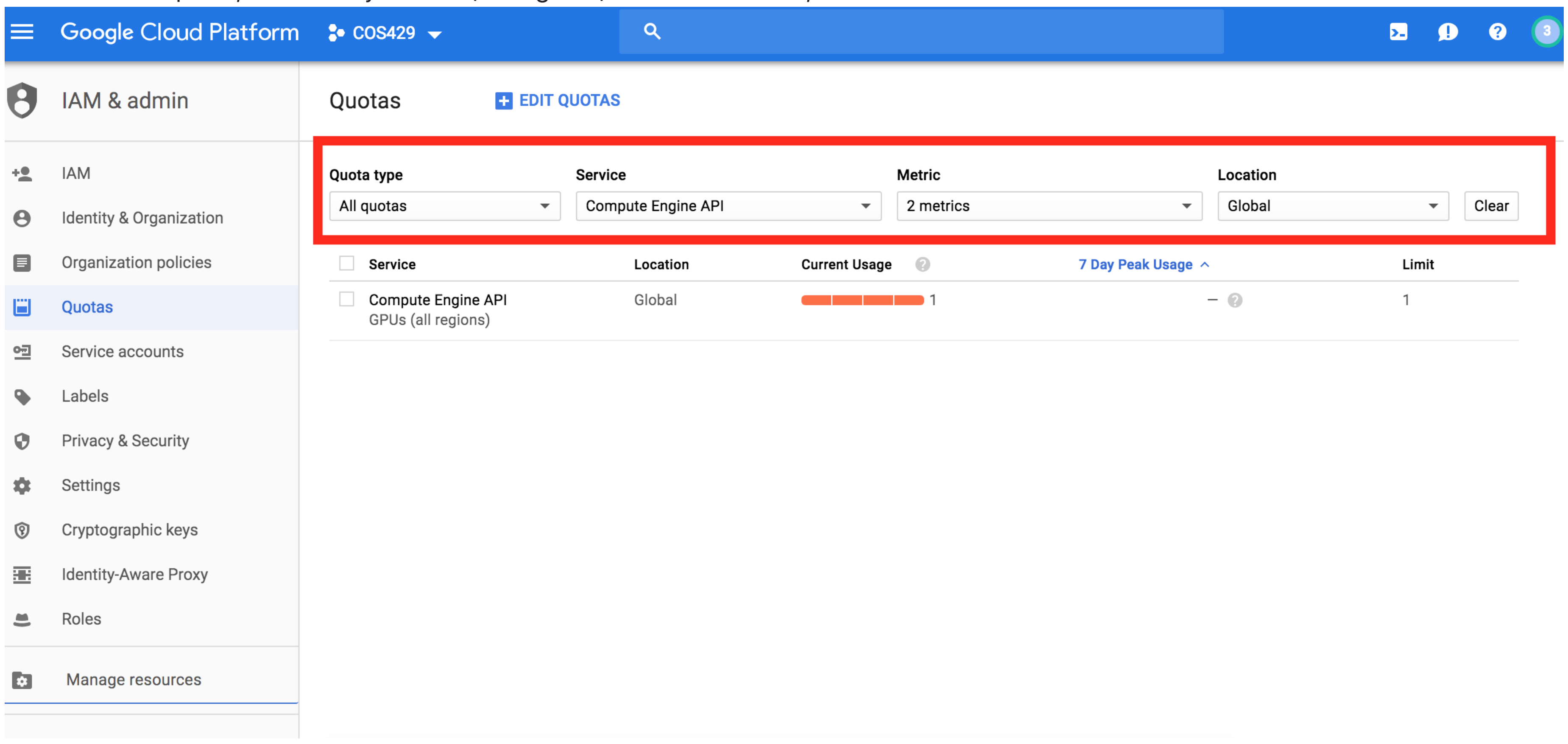
Select or create a project

The Google Cloud Platform uses projects to manage resources. To use Google Compute Engine, select an existing project or create a new one.

Select a project

Select a project

4. You then need to go back to the Navigation Menu, and select the "IAM & admin" option. Then select "Quotas" from the "IAM & admin" menu. In the Service option, select 'None' first and then scroll down and select "Compute Engine API". In the Metric option, select only "GPUs (all regions)" and in Location, select "Global".

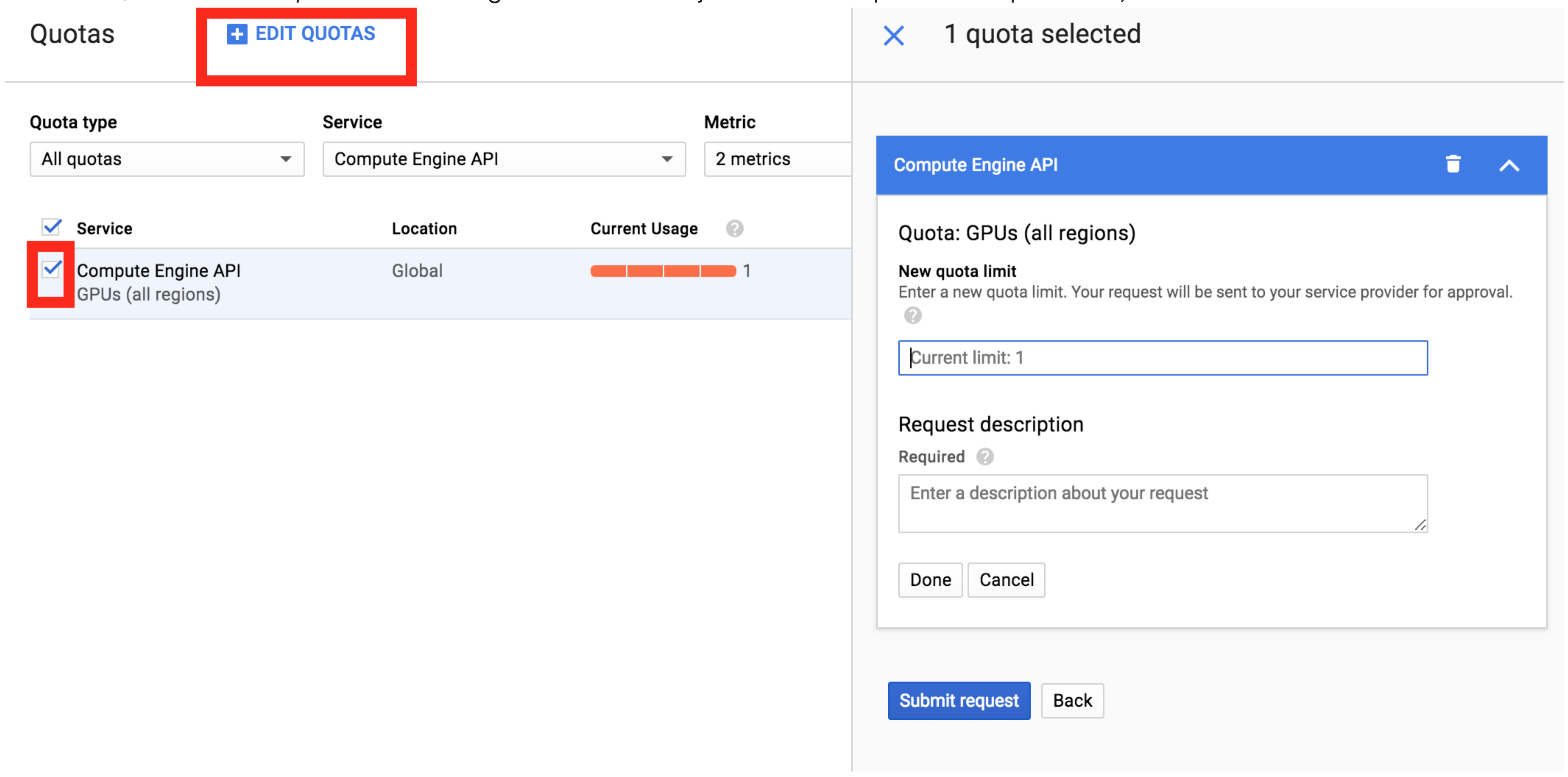


Quotas

Quota type: All quotas | Service: Compute Engine API | Metric: 2 metrics | Location: Global

	Service	Location	Current Usage	7 Day Peak Usage	Limit
☐	Compute Engine API GPUs (all regions)	Global	1	–	1

5. You want to increase your limit from 0 to 1 for this Service and you need to submit a request form to do this. (In the images provided, our limit is already at 1 as we already submitted the request form). Fill out the request form (enter 1 in the New Quota Limit box; enter something like "Course Project" in the Request description box).



Quota type: All quotas
Service: Compute Engine API
Metric: 2 metrics

Service	Location	Current Usage
✓ Compute Engine API GPUs (all regions)	Global	1

Quota: GPUs (all regions)

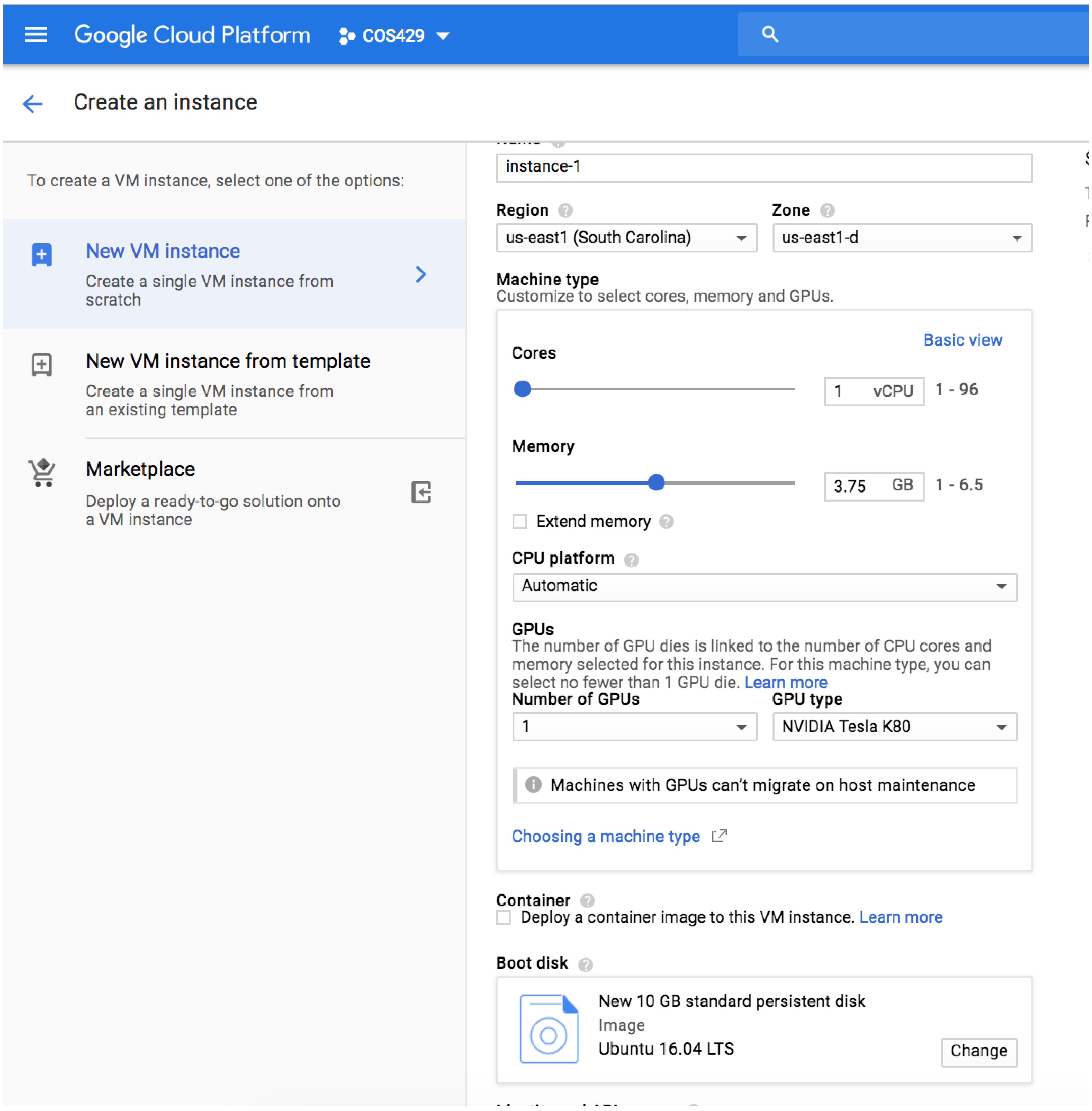
New quota limit
Enter a new quota limit. Your request will be sent to your service provider for approval.

Current limit: 1

Request description
Required

Enter a description about your request

6. Within a few minutes, you should receive another email from Google saying that your request has been granted.
7. Now go back to the Compute Engine tab, select "VM instances" and click to "Create Instance". When you create your VM instance, choose "us-east1" for Region, "us-east1-d" for Zone, click to Customize the Machine Type and change "Number of GPUs" to 1 and "GPU Type" to NVIDIA Tesla K80. You should also change the operating system of the VM via the "Boot Disk" option: set it to "Ubuntu 16.04 LTS" (this will help a lot later when you try to install CUDA).


Google Cloud Platform COS429

Create an instance

To create a VM instance, select one of the options:

New VM instance
Create a single VM instance from scratch

New VM instance from template
Create a single VM instance from an existing template

Marketplace
Deploy a ready-to-go solution onto a VM instance

instance-1

Region: us-east1 (South Carolina)
Zone: us-east1-d

Machine type
Customize to select cores, memory and GPUs.

Basic view

Cores: 1 vCPU 1 - 96

Memory: 3.75 GB 1 - 6.5

Extend memory

CPU platform: Automatic

GPUs
The number of GPU dies is linked to the number of CPU cores and memory selected for this instance. For this machine type, you can select no fewer than 1 GPU die. Learn more

Number of GPUs: 1
GPU type: NVIDIA Tesla K80

Machines with GPUs can't migrate on host maintenance

Choosing a machine type

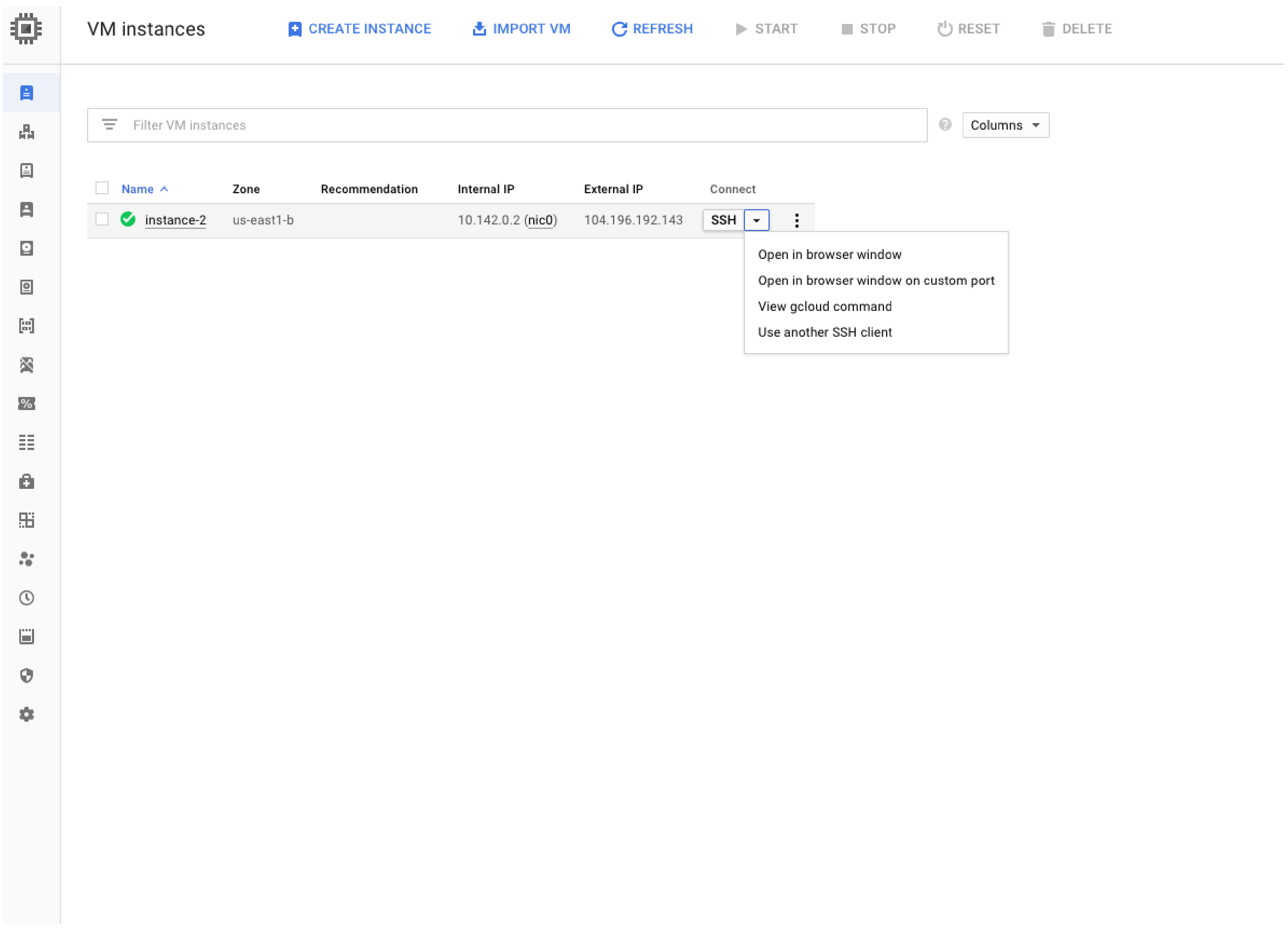
Container
Deploy a container image to this VM instance. Learn more

Boot disk
New 10 GB standard persistent disk
Image
Ubuntu 16.04 LTS
Change

Running the basis code

For running the basic code for training and testing a network in Tensorflow, you can follow the following steps:

1. **Access the VM Instance the you created:**
    1. There are multiple ways of accessing the VM instance that you created. We recommend you install the Google Cloud SDK as it provides a quick way to access the VM via terminal as well as transfer data. To install Google Cloud SDK follow the steps here.
    2. Once you have installed Google Cloud SDK, type "gcloud auth login" on the termial and authenticate on your browser with the account you used to create the VM instance.
    3. To access the VM instance, you can find the terminal command by clicking on the icon "View gcloud command" as shown below:



Name	Zone	Recommendation	Internal IP	External IP	Connect
instance-2	us-east1-b		10.142.0.2 (nic0)	104.196.192.143	SSH

   4. You can transfer data to your VM instance by using the "gcloud compute scp" command. More info about it can be found here.

2. **Install pip and virtual environment:**
   1. After you have terminal access the Google Cloud VM, type "sudo apt install python3-venv python3-pip" to install pip and virtual environment. Pip is used to install and manage various python packages like tensorflow, numpy. Virtual environments are used to separate out different python environments. Installing virtual environments is optional, however, it is highly recommended as it provides clean working environments for python. Virtual environments facilitate "sharing" your environment with others. A basic introduction to them could be found here.

3. **Install CUDA and CuDNN:**
   1. For faster training of deep neural networks, we use special hardware called GPU (Graphical Processing Unit). CUDA is a programming language used to run code on GPUs. CuDNN is a CUDA library specifically designed for running deep neural networks. We won't directly use CUDA or CuDNN, Python libraries like TensorFlow and PyTourch use them in the backend so they must be installed. A good guide for installing CUDA and CuDNN on the Google Cloud VM can be found here (Install CuDNN version 6.0 instead of version 7.3 as done in the guide). **WARNING:** The version of Tensorflow you install later should be compatible with the version of CUDA and CuDNN. We have tested that CUDA 8.0, CuDNN 6.0 and Tensorflow 1.4.1 is compatible and works with the starter code (explained later).
4. **Optional: Create and activate a Virtual Environment:**
   1. To create a virtual environment with name COS429 and default python as python3, use the command "virtualenv COS429 --python=/usr/bin/python3".
   2. To activate the virtual environment type "source activate COS429". Make sure to always activate the virtual environment whenever you restart your machine, so that all libraries (like tensorflow) installed inside that virtual environment are accessible to python.
5. **Install Tensorflow and Scikit-Image**
   1. If you followed all the instructions till this point correctly, installing tensorflow should be easy. To install tensorflow, just type "pip install tensorflow-gpu==1.4.1" in the terminal.
   2. To check whether everything worked, type "python -c 'import tensorflow as tf; print(tf.__version__)'". If you see an error, there is some mistake in your installation. If you see "1.4.1", it means everything worked fine (HURRAY!!).
   3. For the starter code, we also need to the Scikit-Image library. This library is used for reading/writing images. To install it, just type "pip install scikit-image" in the terminal.
6. **Run the starter code**
   1. Download the data and evaluation code in the VM instance using the command "wget http://www.cs.princeton.edu/~crosati/surface-normal-prediction-website-class-project/cos429.tgz". Remember that you need to uncompress the data. You can use the command "tax xvzf cos429.tgz". This will uncompress the data into a folder named cos429Challenge.
   2. The starter code is implemented in the script cos429Challenge/train.py. It implements a basic neural network for the project. You can build upon it for your project.

3. In the starter code suitably modify the variables SUMMARIES_PATH and DUMP_FOLDER. (These folders should exist in your file system. These are the location where tensorboard summaries and image predictions are stored.
4. Run the starter code as "python train.py".