Google Cloud Tutorial

Google Cloud Tutorial

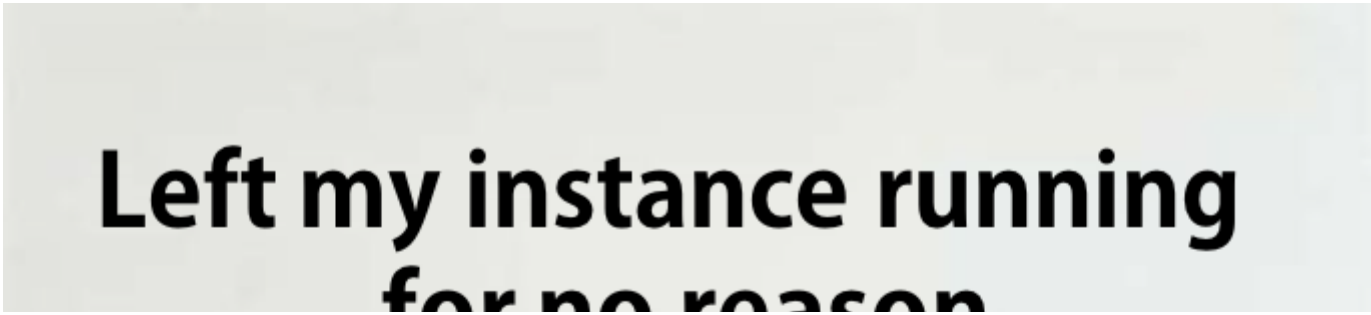
BEFORE WE BEGIN

BIG REMINDER: Make sure you stop your instances!

(We know you won't read until the very bottom once your assignment is running, so we are printing this at the top too since it is ***super important***)

Don't forget to ***stop your instance*** when you are done (by clicking on the stop button at the top of the page showing your instances), otherwise you will ***run out of credits*** and that will be very sad. :(

If you follow our instructions below correctly, you should be able to restart your instance and the downloaded software will still be available.



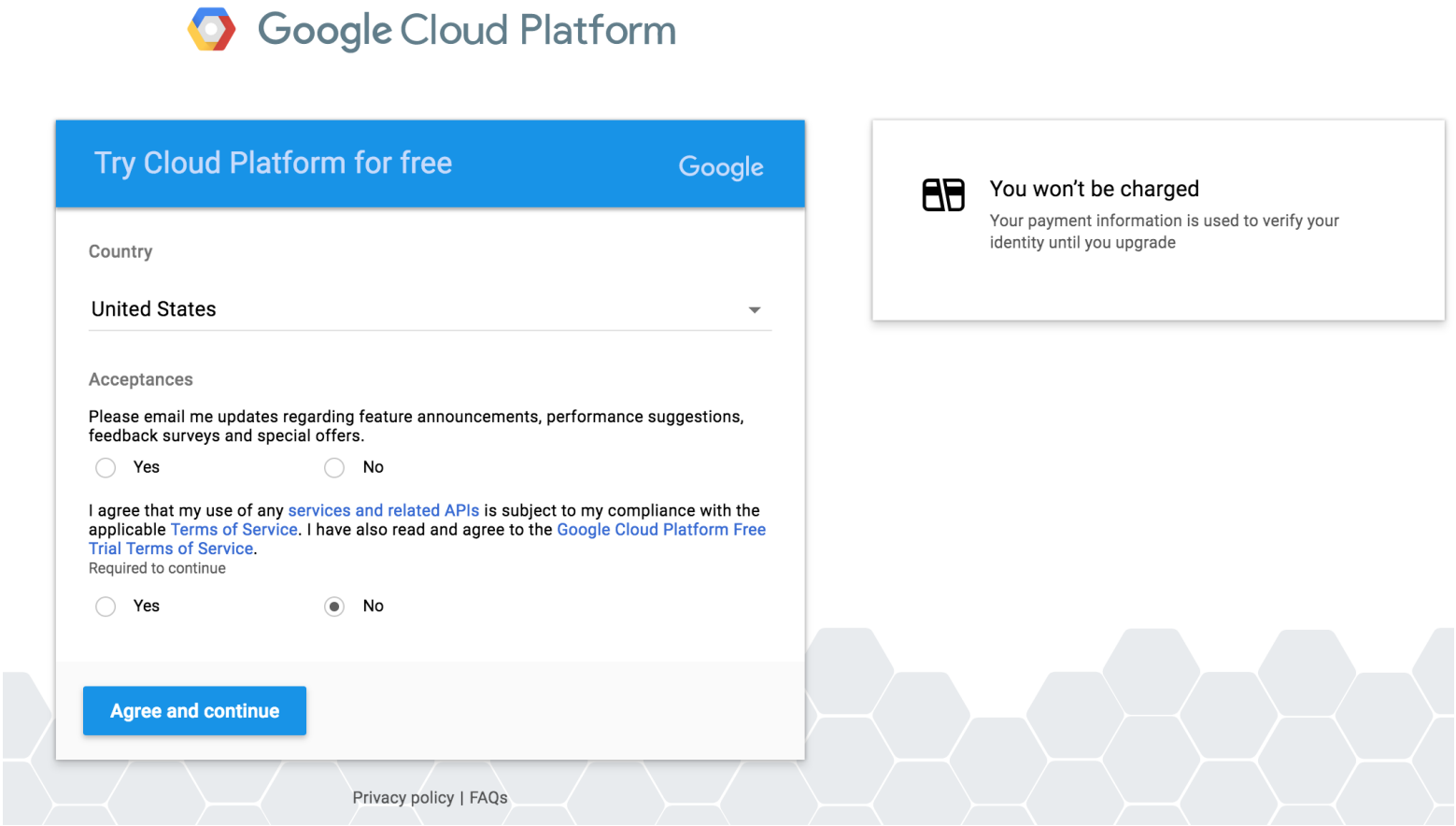

for no reason
No more Google Cloud credits

Create and Configure Your Account

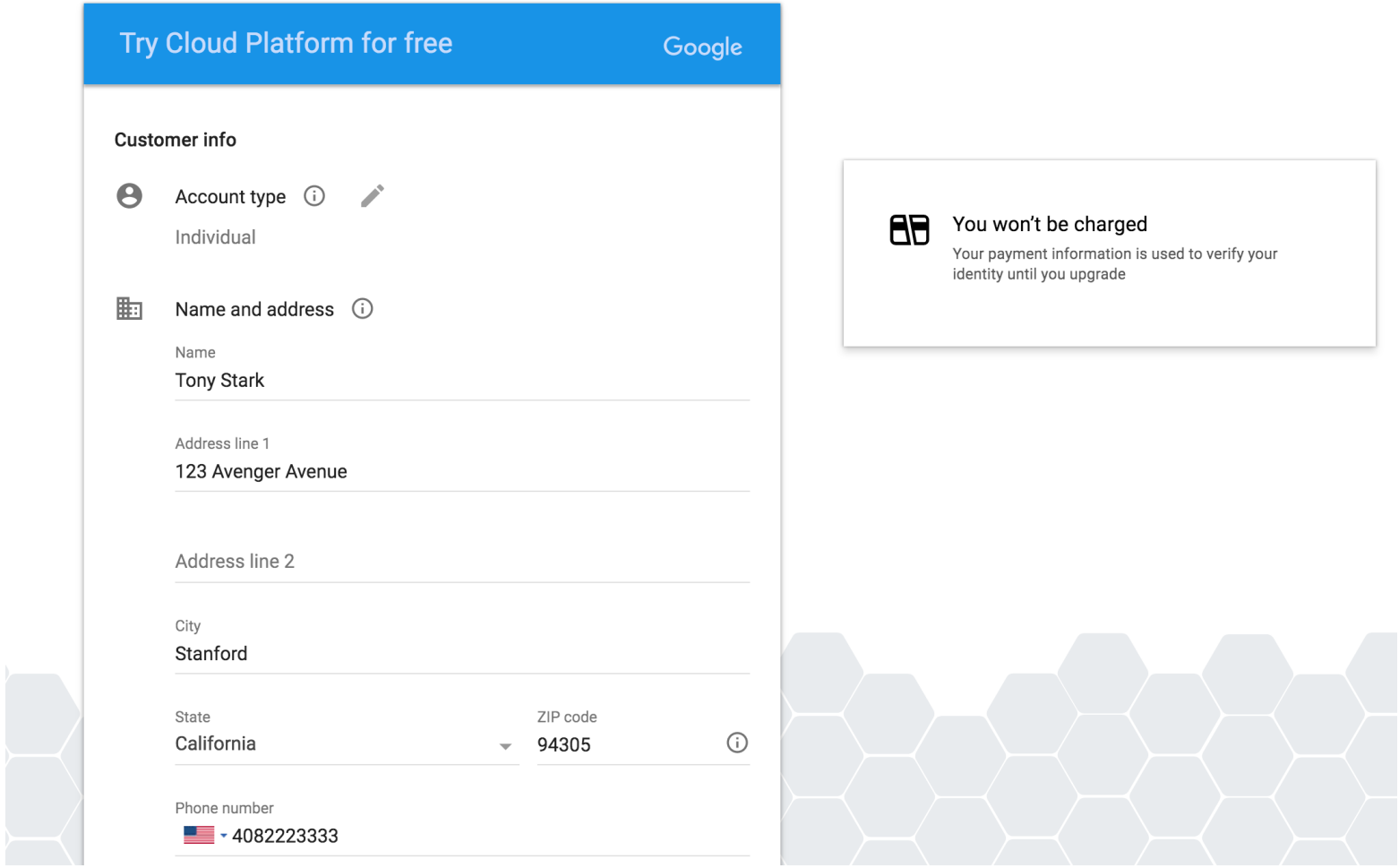
For the class project and assignments, we offer an option to use Google Compute Engine for developing and testing your implementations. This tutorial lists the necessary steps of working on the assignments using Google Cloud. **We expect this tutorial to take about an hour. Don't get intimidated by the steps, we tried to make the tutorial detailed so that you are less likely to get stuck on a particular step. Please tag all questions related to Google Cloud with google_cloud on Piazza.**

This tutorial goes through how to set up your own Google Compute Engine (GCE) instance to work on the assignments. Each student will have $100 in credit throughout the quarter. When you sign up for the first time, you also receive $300 credits from Google by default. Please try to use the resources judiciously. But if $100 ends up not being enough, we will try to adjust this number as the quarter goes on.

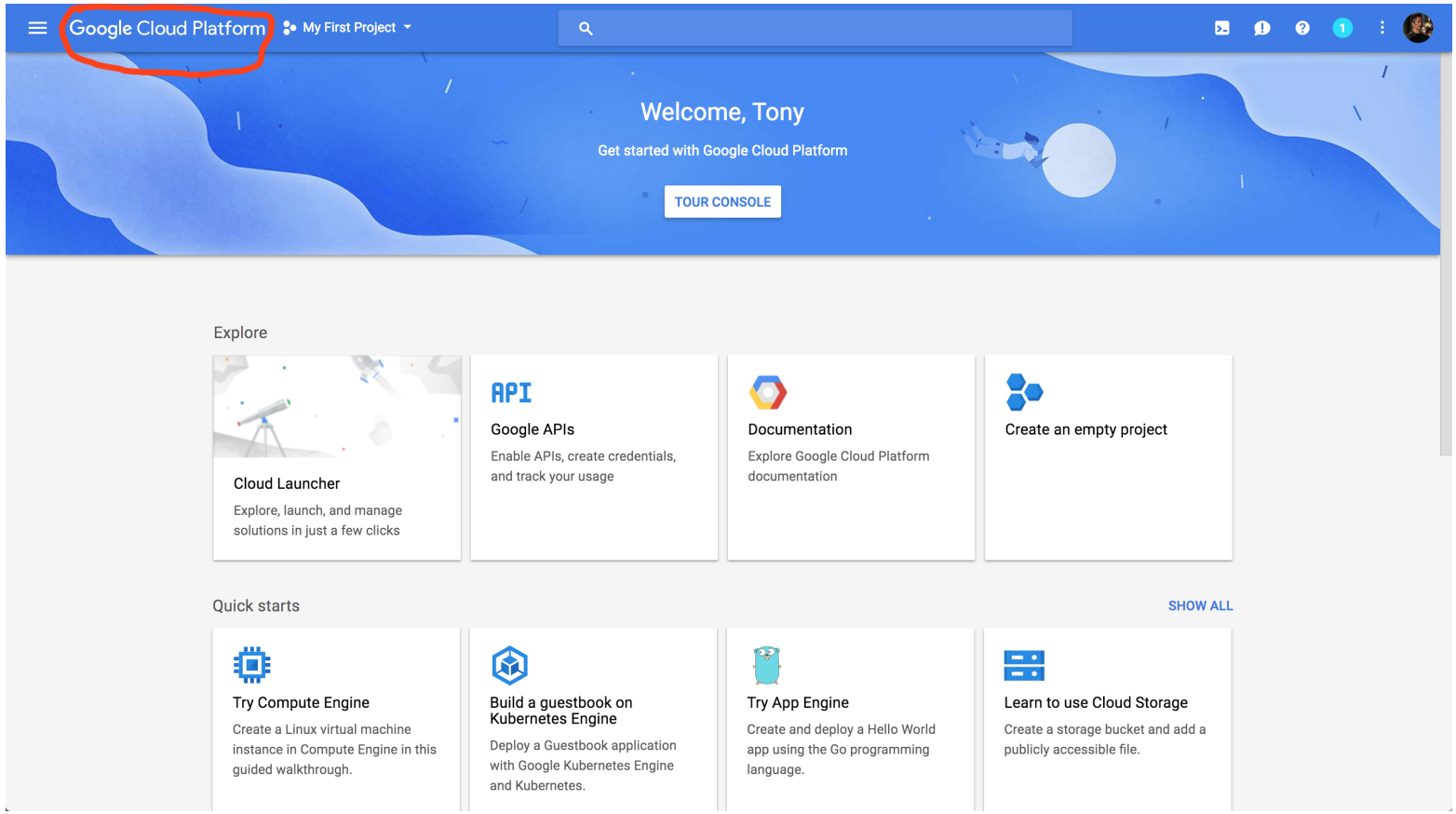
First, if you don't have a Google Cloud account already, create one by going to the Google Cloud homepage and clicking on **Compute**. When you get to the next page, click on the blue **TRY IT FREE** button. If you are not logged into gmail, you will see a page that looks like the one below. Sign into your gmail account or create a new one if you do not already have an account.



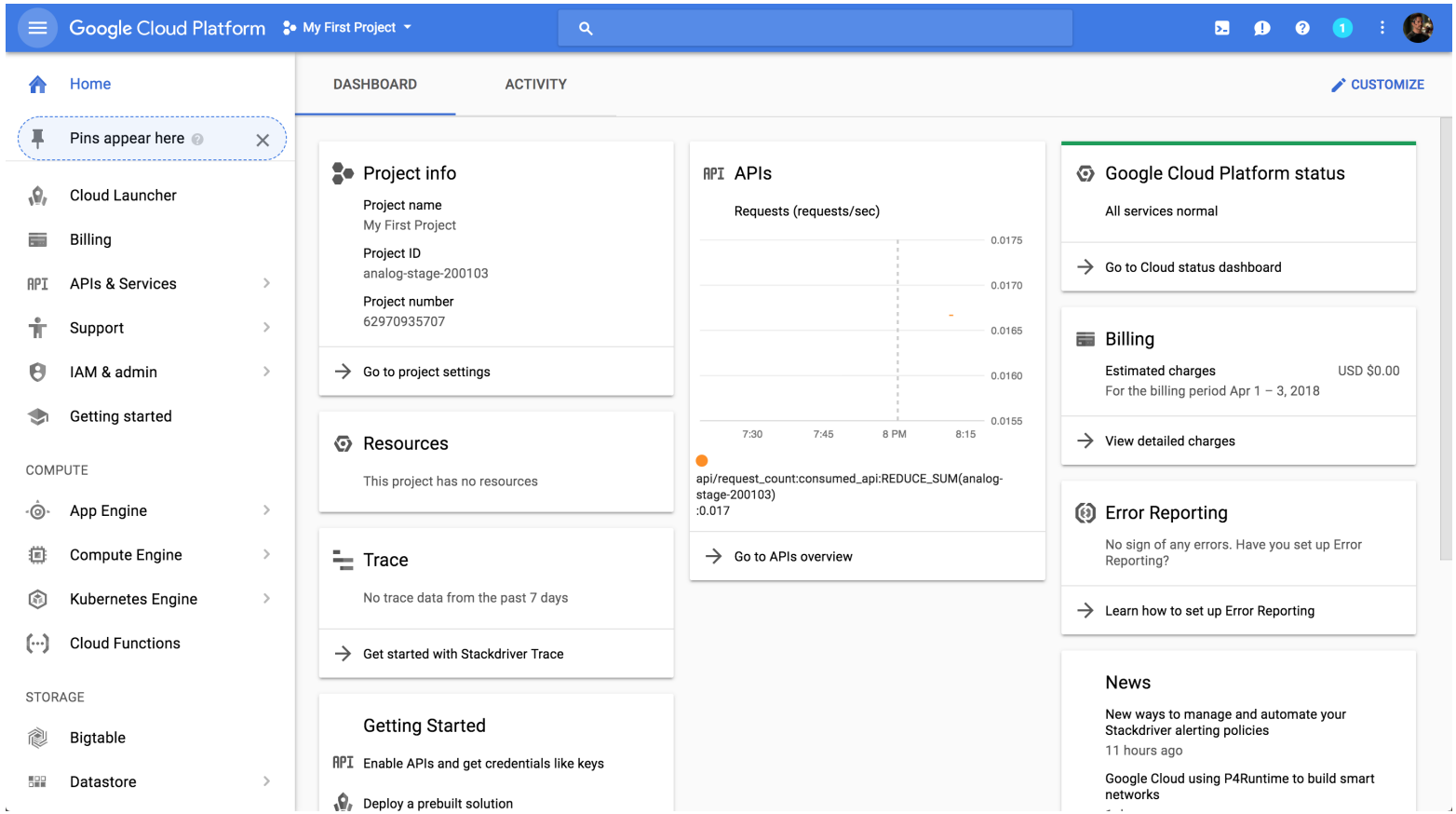
Click the appropriate **yes** or **no** button for the first option, and check **yes** for the second option after you have read the required agreements. Press the blue **Agree and continue** button to continue to the next page to enter the requested information (your name, billing address and credit card information). Remember to select "**Individual**" as "Account Type":



Once you have entered the required information, press the blue **Start my free trial** button. You will be greeted by a page like this:



Press the “Google Cloud Platform” (in red circle), and it will take you to the main dashboard:

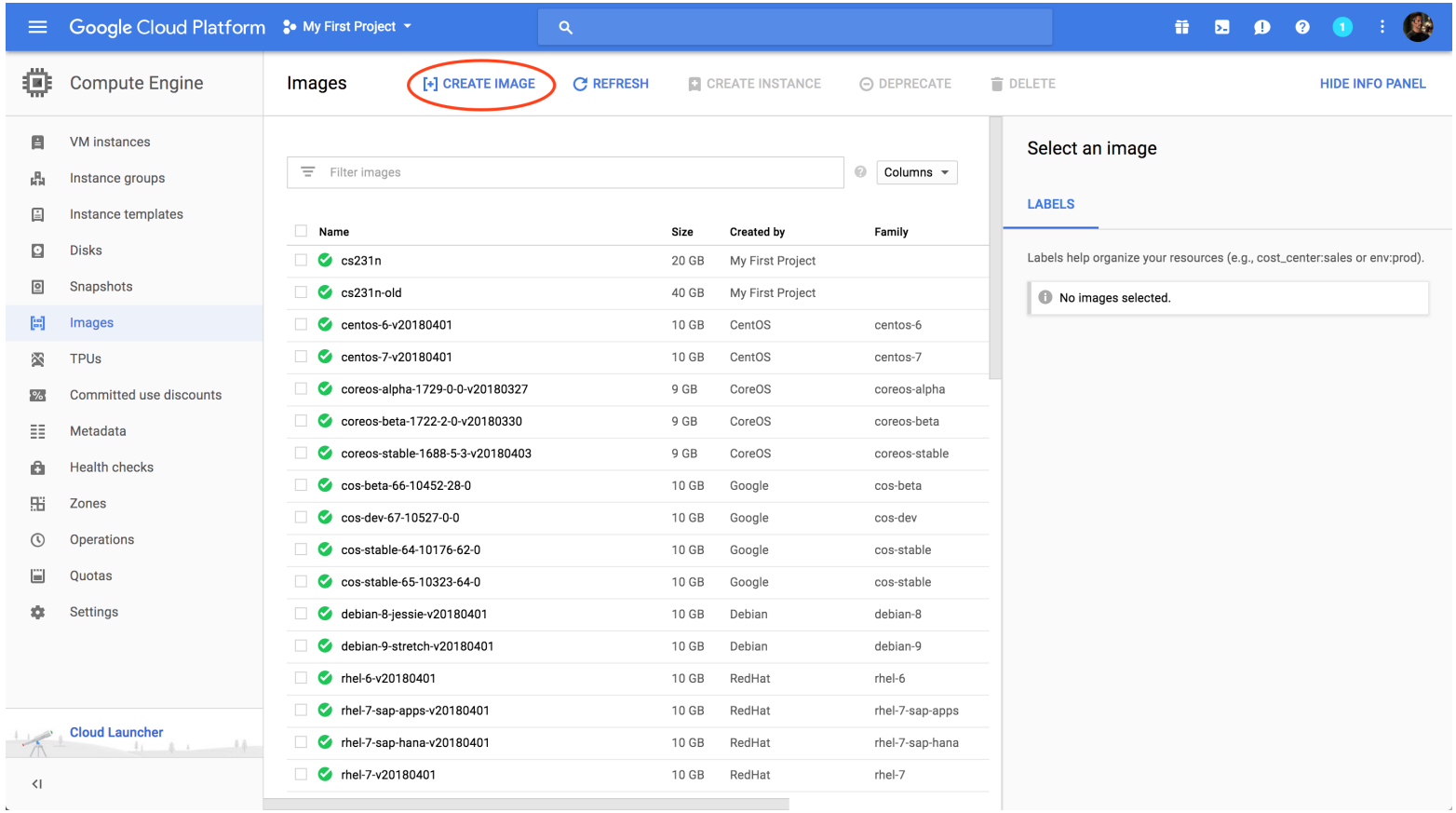


To change the name of your project, click on **Go to project settings** under the **Project info** section.

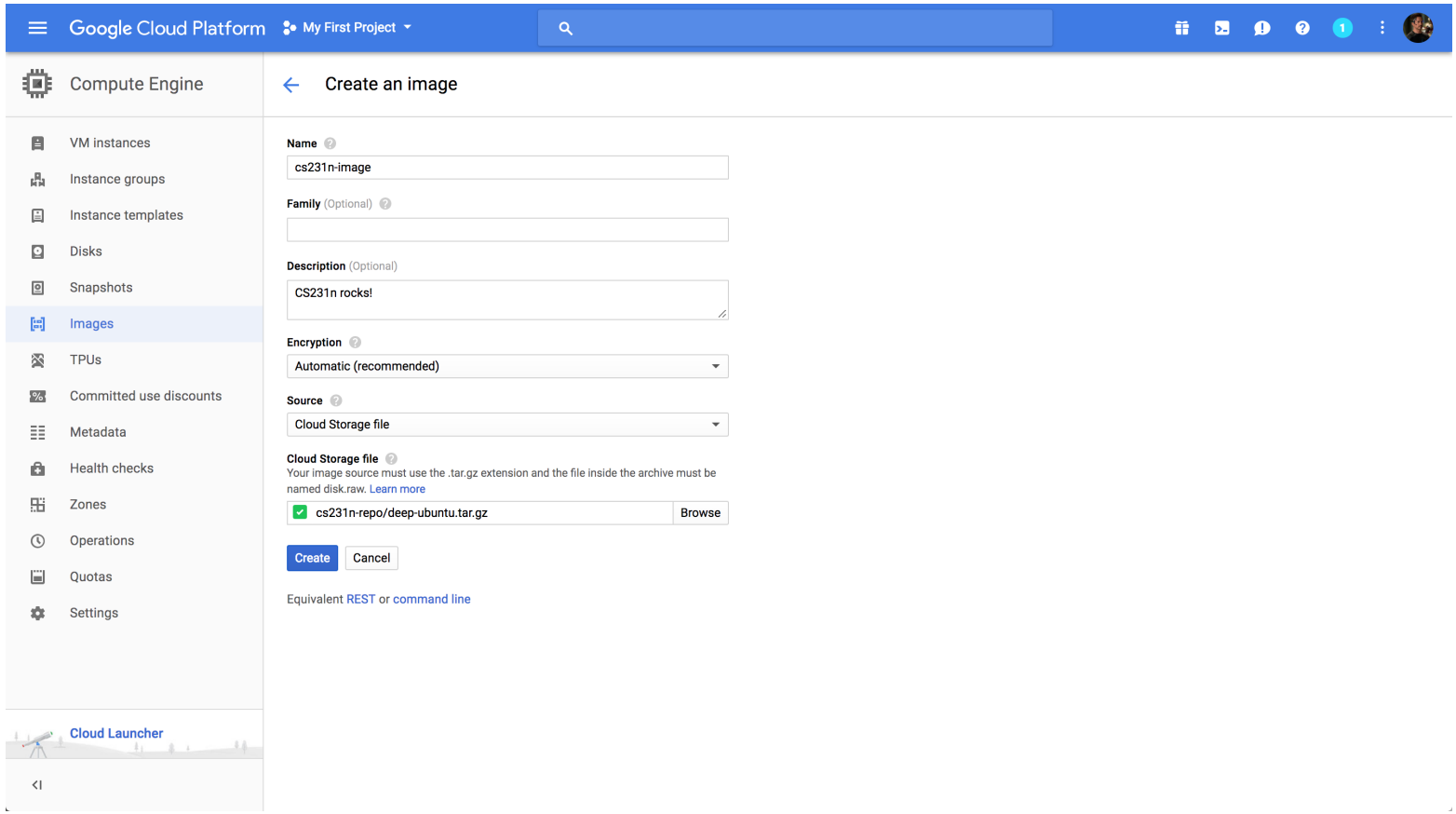
Create an image from our provided disk

For all assignments and the final project, we provide you with a pre-configured disk that contains the necessary environment and deep learning frameworks. To use our disk, you first need to create your own custom image using our file, and use this custom image as the boot disk for your new VM instance.

Go to **Compute Engine**, then **Images** and click on the blue **Create Image** button at the top of the page. See the screenshot below.



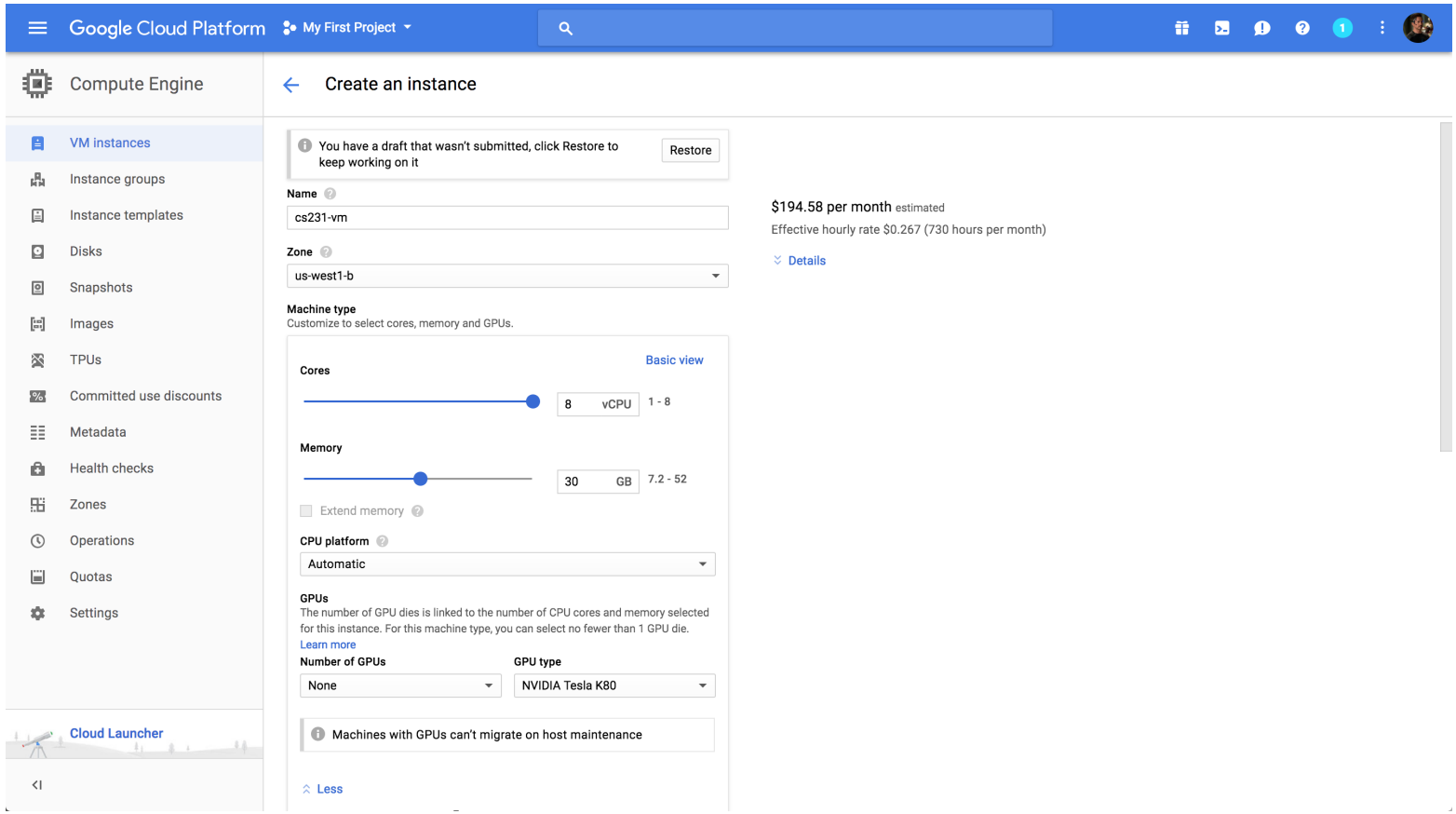
Enter your preferred name in the **Name** field. Mine is called **cs231n-image**. Select **Cloud Storage file** for **Source**, enter **cs231n-repo/deep-ubuntu.tar.gz** and click on the blue **Create** button. See the screenshot below. It will take a few minutes for your image to be created (about 10-15 in our experience, though your mileage may vary).



Launch a Virtual Instance

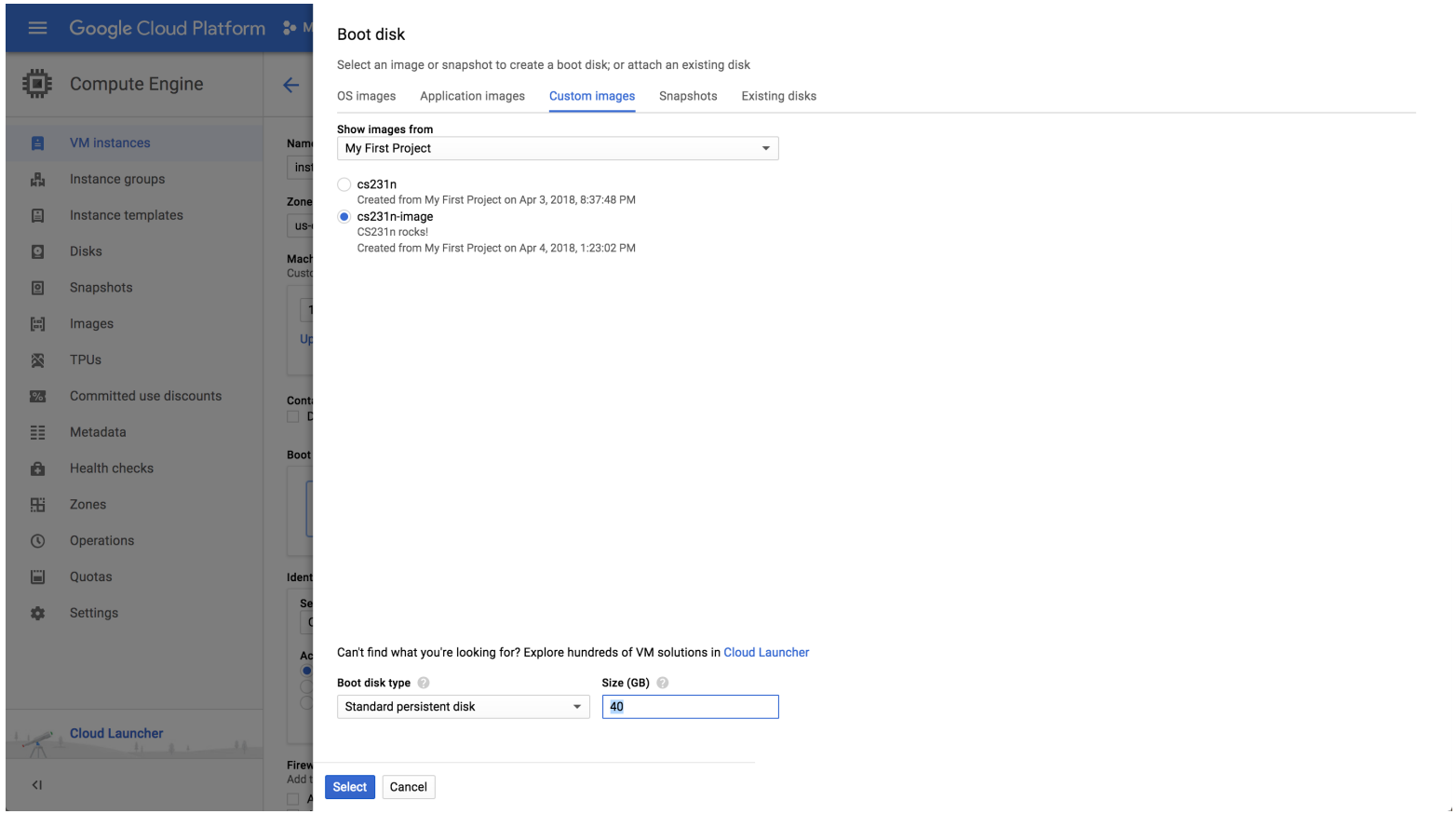
To launch a virtual instance, go to the **Compute Engine** menu on the left column of your dashboard and click on **VM instances**.

Then click on the blue **Create** button on the next page. This will take you to a page that looks like the screenshot below. **(NOTE: Please carefully read the instructions in addition to looking at the screenshots. The instructions tell you exactly what values to fill in).**



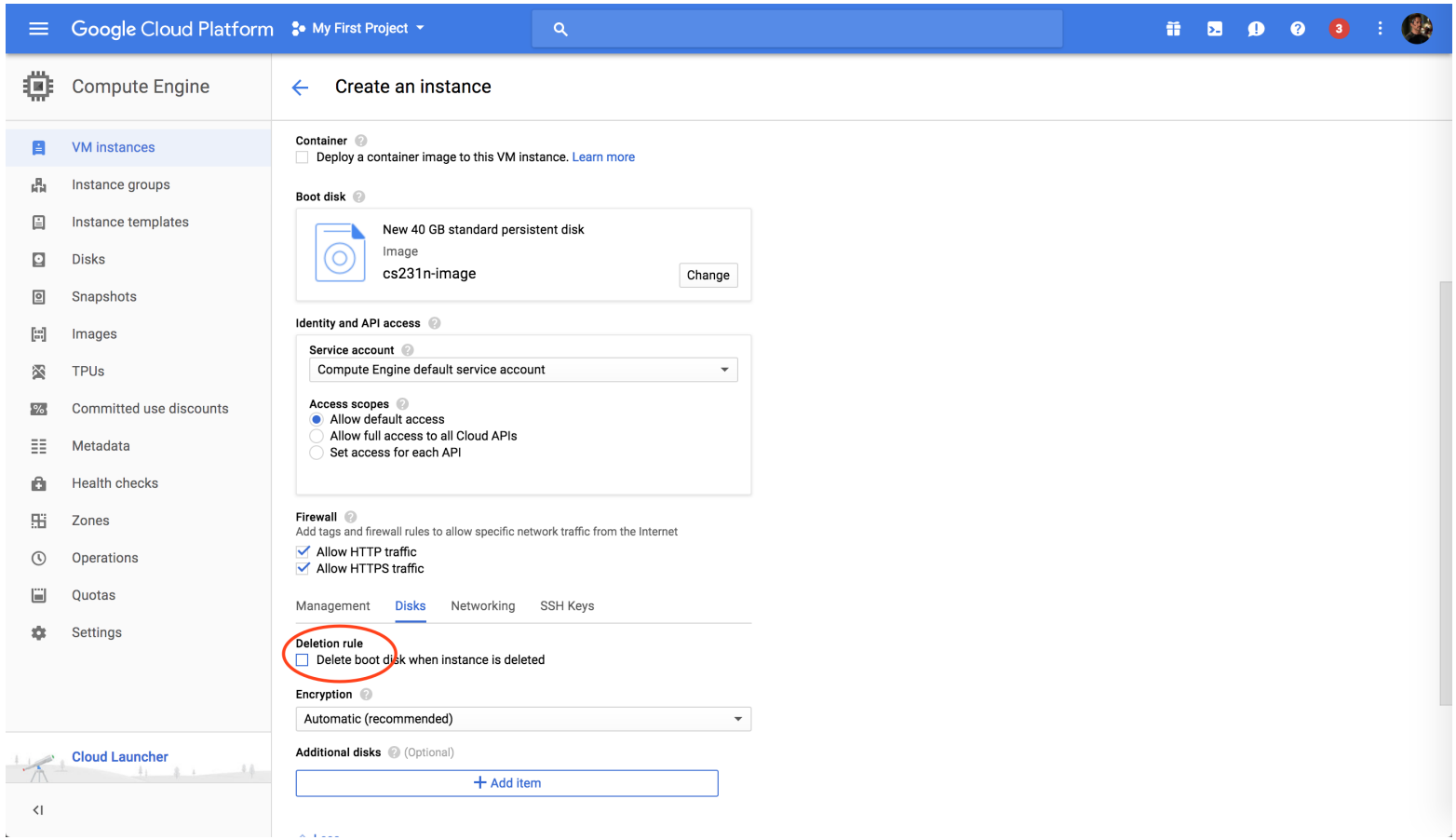
Make sure that the Zone is set to be **us-west1-b** (especially for assignments where you need to use GPU instances). Under **Machine type** pick the **8 vCPUs** option. Click on the **customize** button under **Machine type** and make sure that the number of cores is set to 8 and the number of GPUs is set to **None** (we will not be using GPUs in assignment 1. GPU will be covered later in this tutorial).

Click on the **Change** button under **Boot disk**, choose **Custom images**, you will see this screen:

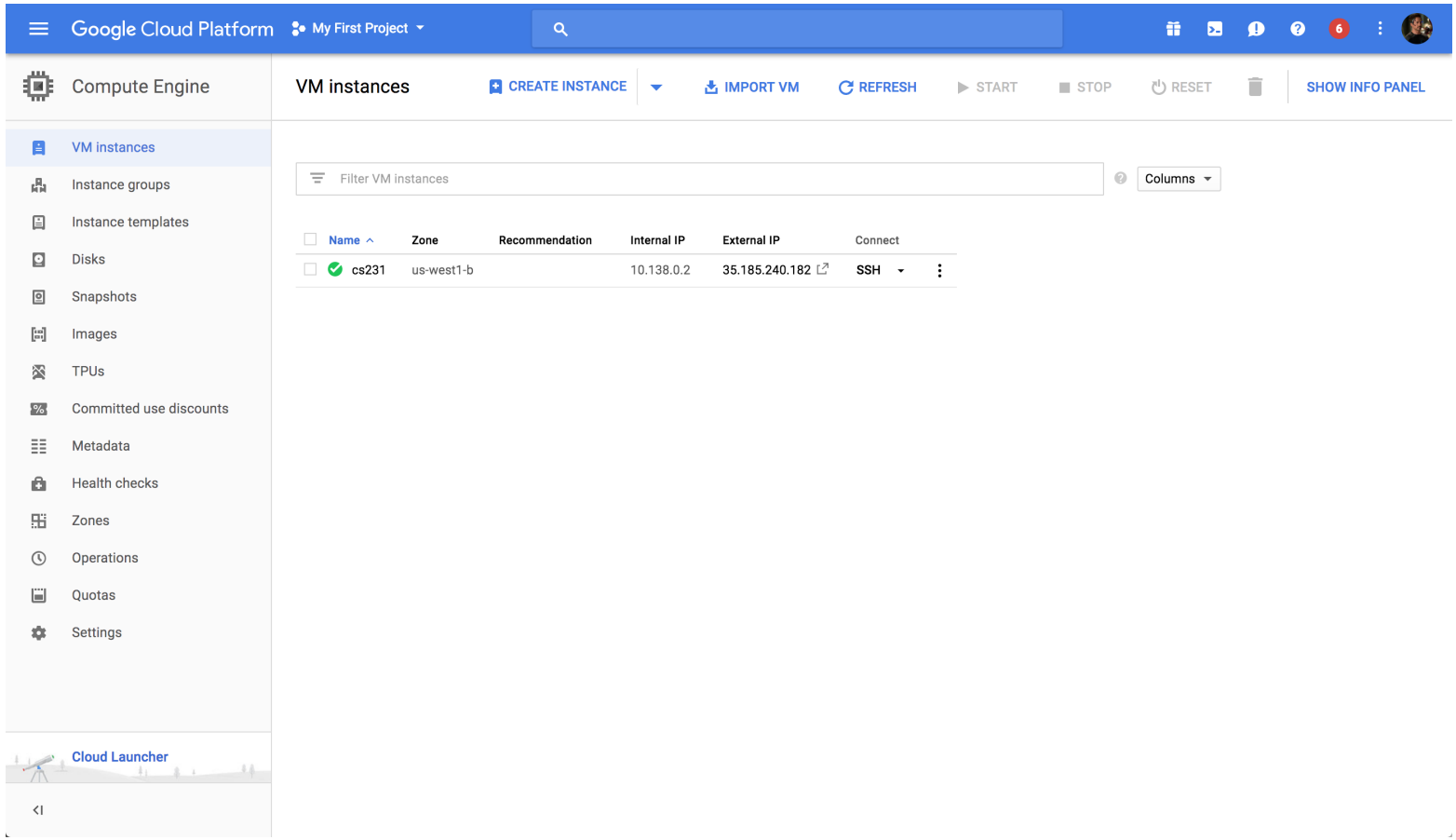


Select the image you created in the previous step, here it's **cs231n-image**. Also increase the boot disk size as you see fit. Click **Select** and you will get back to the "create instance" screen.

Check **Allow HTTP traffic** and **Allow HTTPS traffic**. Expand the **Management, disks, networking, SSH keys** menu if it isn't visible, select **Disks** tab, and uncheck **Delete boot disk when instance is deleted**.



Click on the blue **Create** button at the bottom of the page. You should have now successfully created a Google Compute Instance, it might take a few minutes to start running. When the instance is ready, your screen should look something like the one below. When you want to stop running the instance, click on the blue stop button above.



Take note of your instance name, you will need it to ssh from your laptop.

Connect to Your Virtual Instance

Now that you have created your virtual GCE, you want to be able to connect to it from your computer. The rest of this tutorial goes over how to do that using the command line. First, download the Google Cloud SDK that is appropriate for your platform from here and follow their installation instructions. **NOTE: this tutorial assumes that**

you have performed step #4 on the website which they list as optional. When prompted, make sure you select `us-west1-b` as the time zone.

The easiest way to connect is using the gcloud compute command below. The tool takes care of authentication for you. On your laptop (OS X for example), run:

```
gcloud compute ssh --zone=us-west1-b <YOUR-INSTANCE-NAME>
```

If `gcloud` command is not in system path, you can also reference it by its full path `/<DIRECTORY-WHERE-GOOGLE-CLOUD-IS-INSTALLED>/bin/gcloud`. See this page for more detailed instructions.

First time setup

Upon your first ssh, you need to run a one-time setup script and reload the `.bashrc` to activate the libraries. The exact command is

```
/home/shared/setup.sh && source ~/.bashrc
```

The command will download a git repo, patch your `.bashrc` and copy a jupyter notebook config file to your home directory. If you ever switch account/username, you will have to re-run the setup command. If you see any permission error, simply prepend `sudo` to the command.

When the command finishes without error, run `which python` on the command line and it should report `/home/shared/anaconda3/bin/python`. See screenshot:

```
tonystark@cs231:~$ which python
/home/shared/anaconda3/bin/python
tonystark@cs231:~$ python
Python 3.6.4 |Anaconda custom (64-bit)| (default, Jan 16 2018, 18:10:19)
[GCC 7.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow
/home/shared/anaconda3/lib/python3.6/site-packages/h5py/__init__.py:36: FutureWarning: Conversion of the second
 argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float
64 == np.dtype(float).type`.
  from ._conv import register_converters as _register_converters
>>> import torch
>>> import keras
Using TensorFlow backend.
>>> import caffe2
>>>
```

(don't worry about the Tensorflow warning message)

Our provided image supports the following frameworks:

- Anaconda3, a python package manager. You can think of it as a better alternative to `pip`.
- Numpy, matplotlib, and tons of other common scientific computing packages.
- Tensorflow 1.7, both CPU and GPU.
- PyTorch 0.3, both CPU and GPU.
- Keras that works with Tensorflow 1.7
- Caffe2, CPU only. Note that it is very different from the original Caffe.
- Nvidia runtime: CUDA 9.0 and cuDNN 7.0. They only work when you create a Cloud GPU instance, which we will cover later.

The `python` on our image is `3.6.4`, and has all the above libraries installed. It should work out of the box for all assignments unless noted otherwise. You don't need `virtualenv`, but if you insist, Anaconda has its own way. If you need libraries not mentioned above, you can always run `conda install <mylib>` yourself.

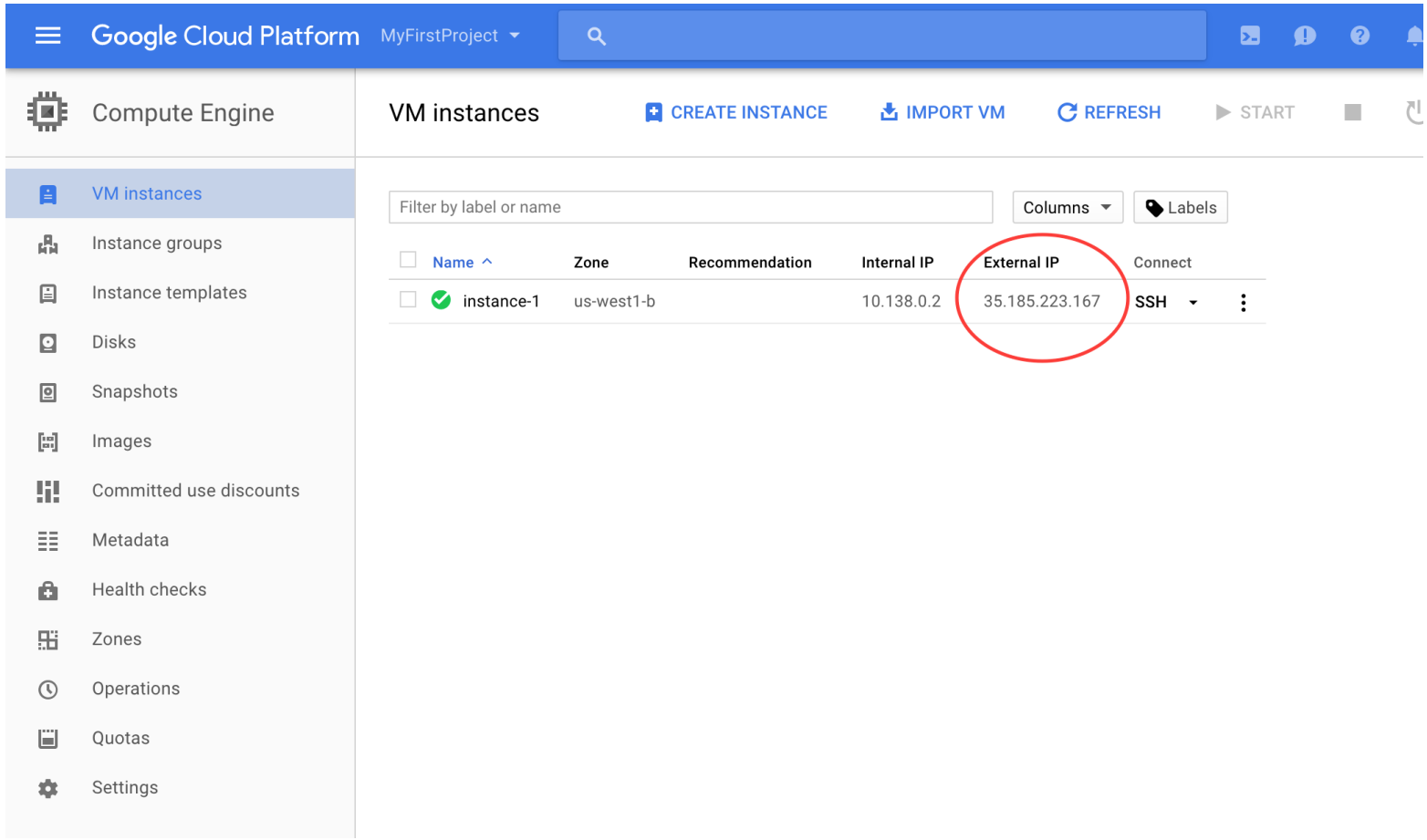
You are now ready to work on the assignments on Google Cloud!

Using Jupyter Notebook with Google Compute Engine

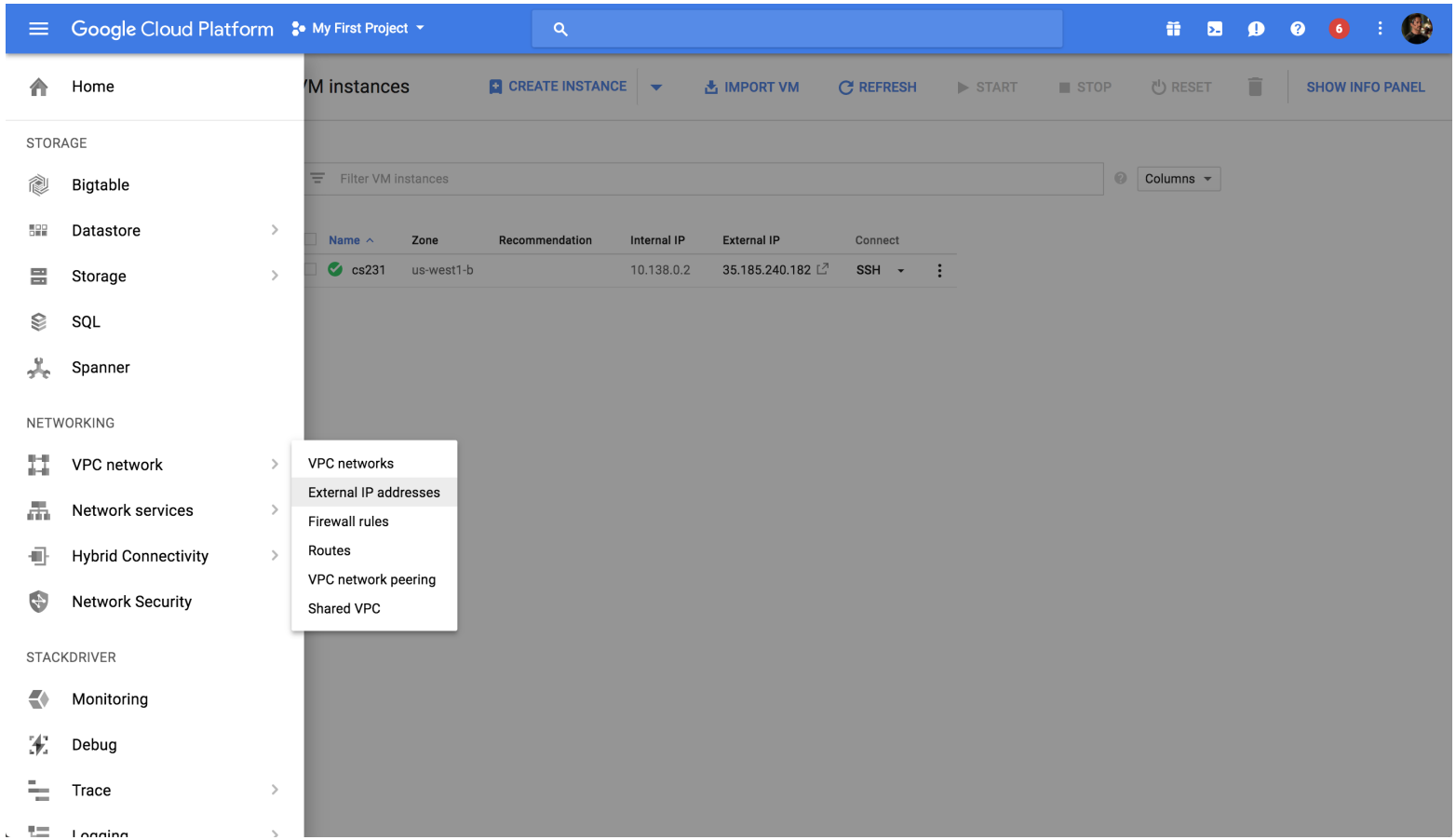
Many of the assignments will involve using Jupyter Notebook. Below, we discuss how to run Jupyter Notebook from your GCE instance and connect to it with your local browser.

Getting a Static IP Address

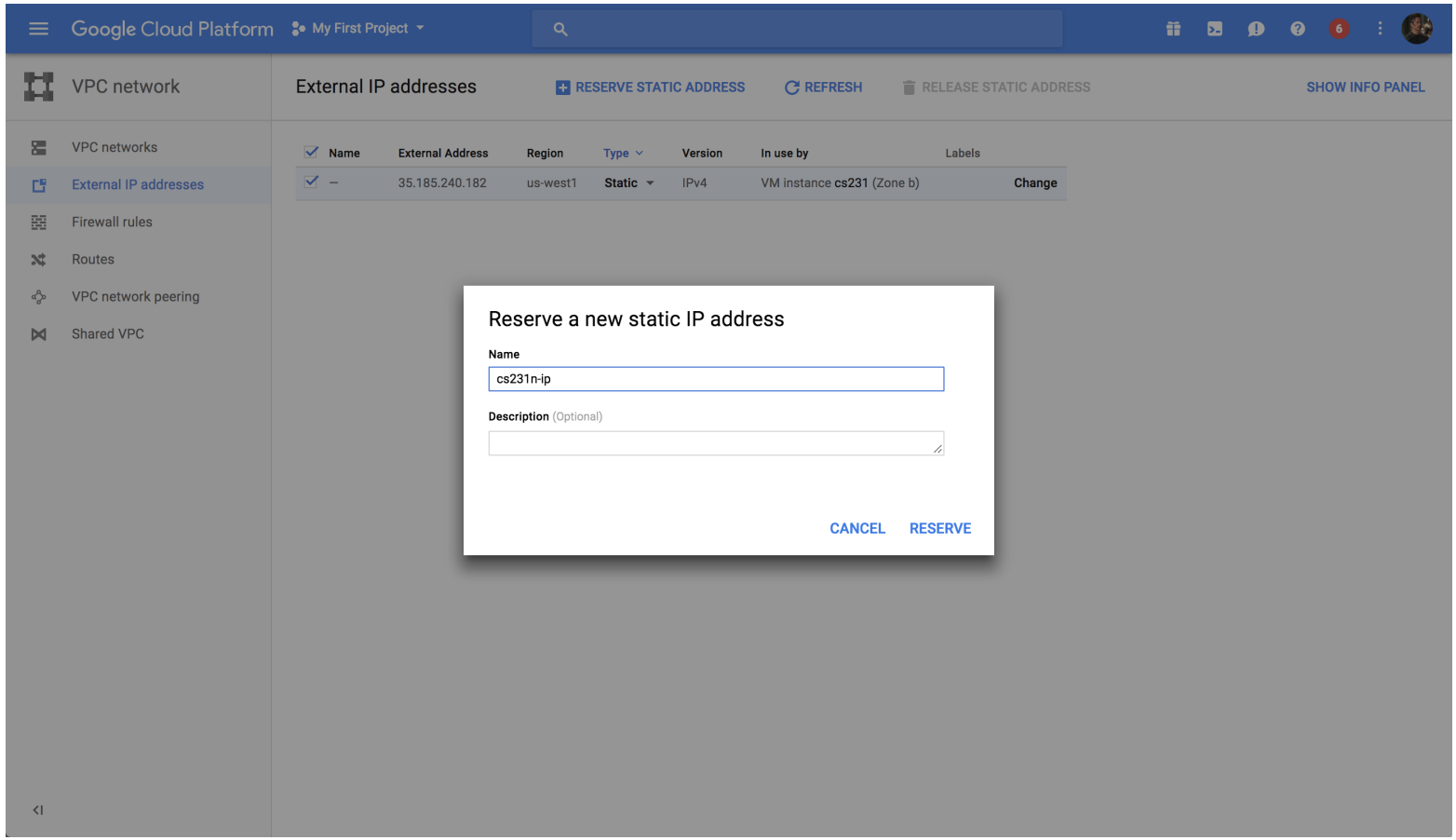
Change the Extenal IP address of your GCE instance to be static (see screenshot below).



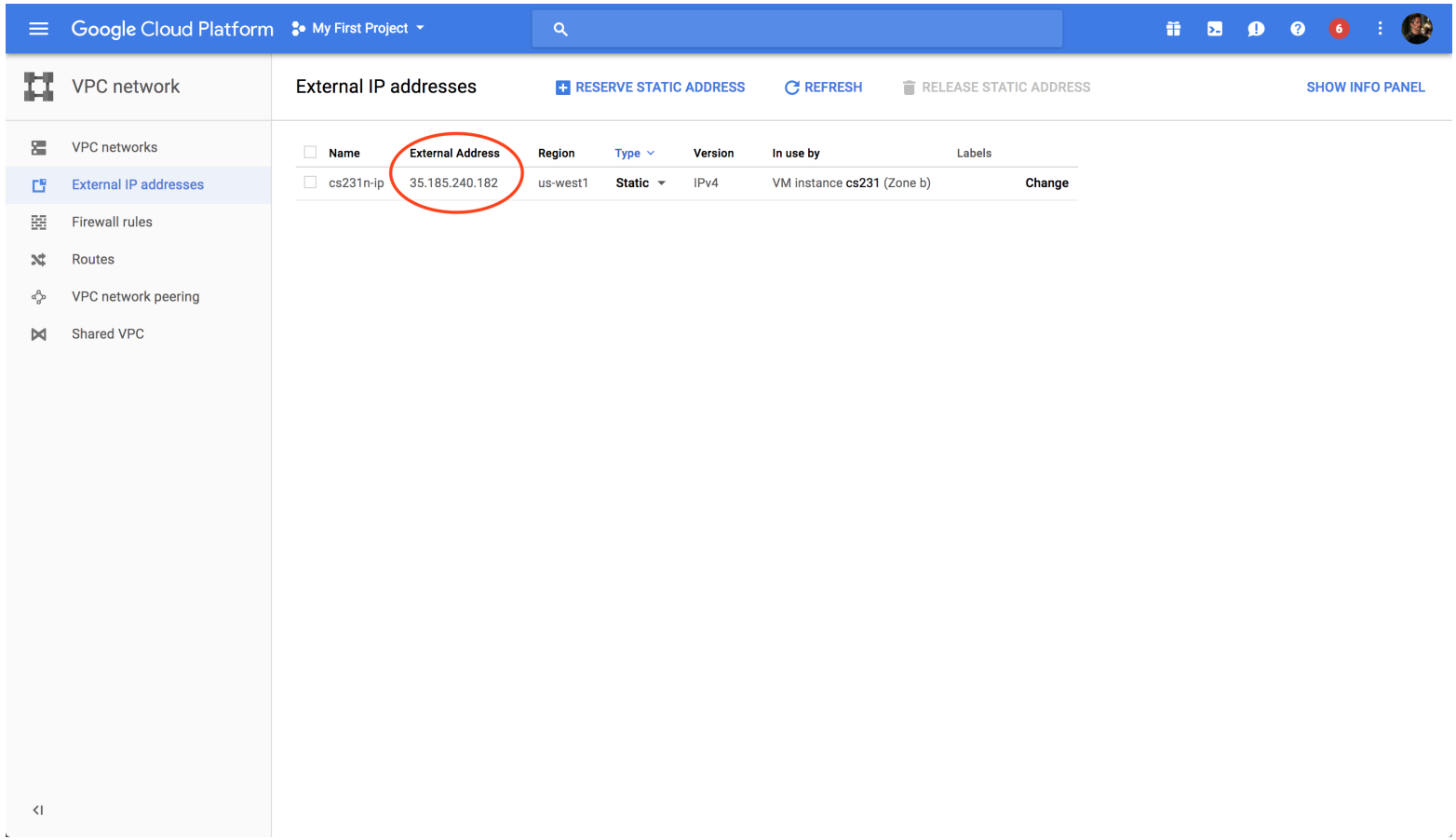
To Do this, click on the 3 line icon next to the **Google Cloud Platform** button on the top left corner of your screen, go to **VPC network** and **External IP addresses** (see screenshot below).



To have a static IP address, change **Type** from **Ephemeral** to **Static**. Enter your prefered name for your static IP, ours is `cs231n-ip` (see screenshot below). And click on Reserve. Remember to release the static IP address when you are done because according to this page Google charges a small fee for unused static IPs.



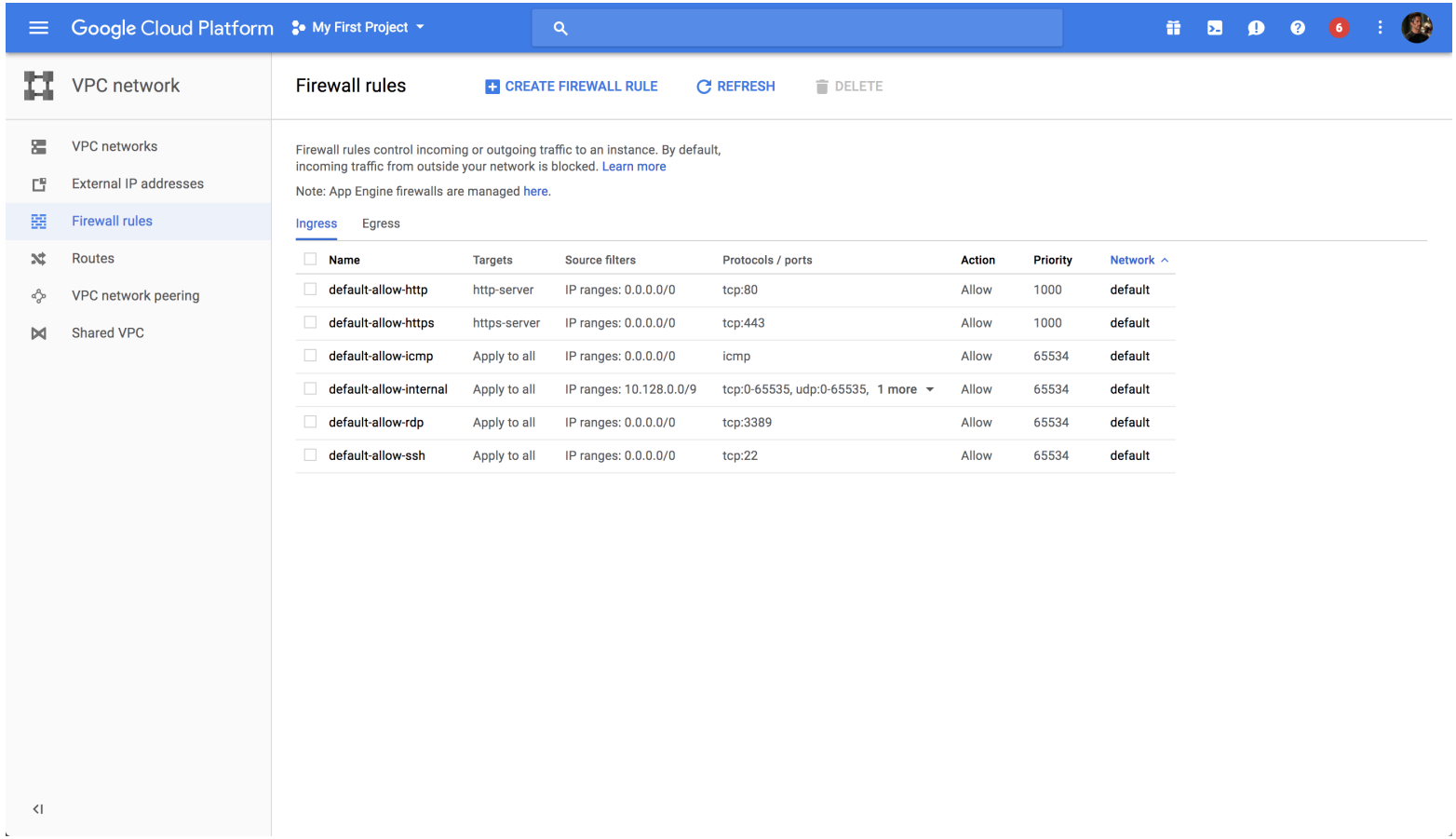
Take note of your Static IP address (circled on the screenshot below). We use 35.185.240.182 for this tutorial.



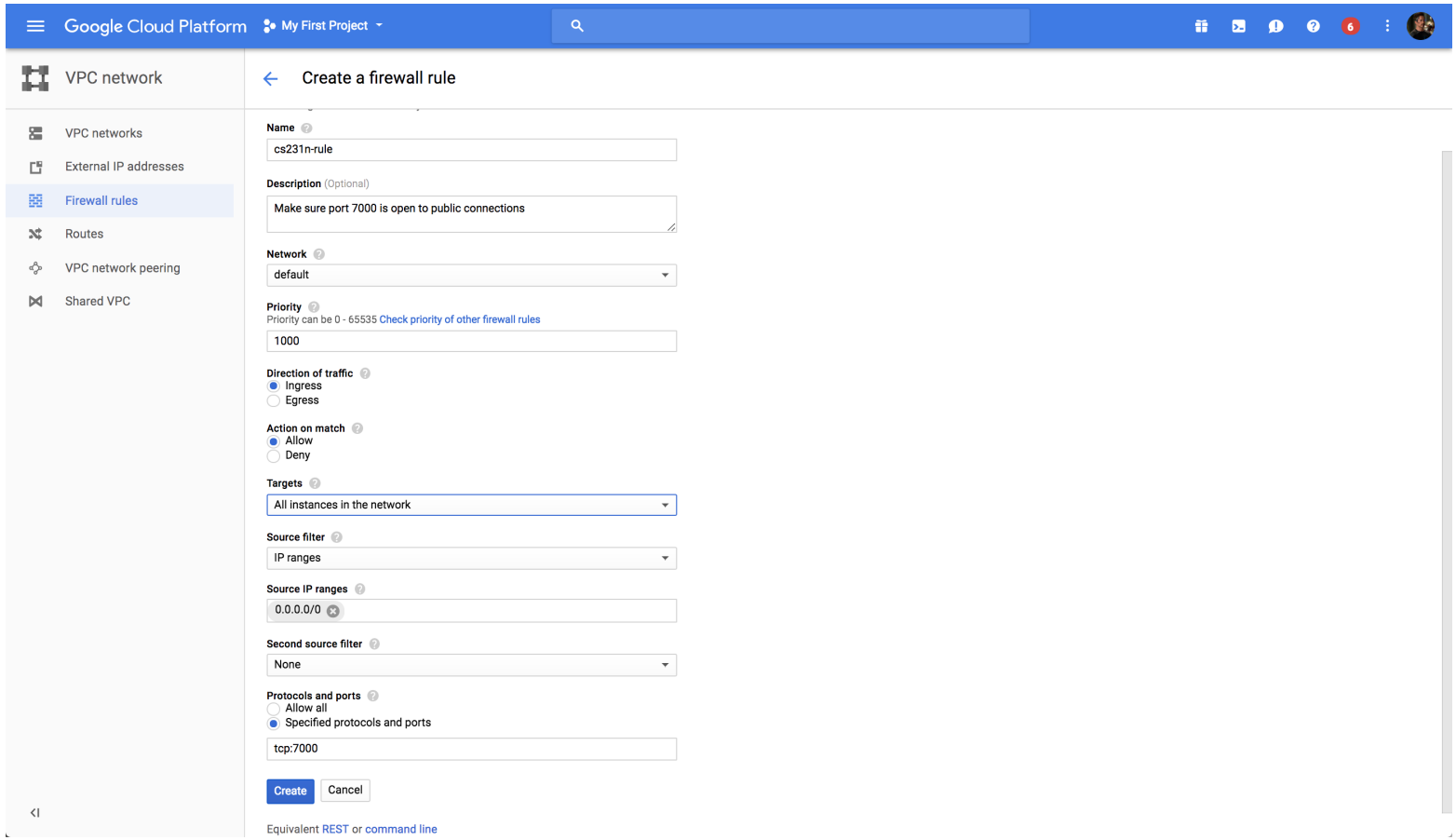
Adding a Firewall rule

One last thing you have to do is adding a new firewall rule allowing TCP acess to a particular port number. The default port we use for Jupyter is **7000**. You can find this default value in the config file generated at setup time (`~/.jupyter/jupyter_notebook_config.py`). Feel free to change it.

Click on the 3-line icon at the top of the page next to **Google Cloud Platform**. On the menu that pops up on the left column, go to **VPC network** and **Firewall rules** (see the screenshot below).



Click on the blue **CREATE FIREWALL RULE** button. Enter whatever name you want: we use cs231n-rule. Select "All instances in the network" for **Targets** (if the menu item exists). Enter `0.0.0.0/0` for **Source IP ranges** and `tcp:<port-number>` for **Specified protocols and ports** where `<port-number>` is the number you used above. Click on the blue **Create** button. See the screenshot below.



Launching and connecting to Jupyter Notebook

After you ssh into your GCE instance using the prior instructions, run Jupyter notebook from the folder with your assignment files. As a quick example, let's launch it from `/home/shared` folder.

```
cd /home/shared
jupyter-notebook --no-browser --port=7000
```

If you simply run `jupyter-notebook` without any command line arguments, it will pick up the default config values in `~/.jupyter/jupyter_notebook_config.py`. In our disk image, it is `no-browser` and port 7000 by default.

The command should block your stdin and display something like:

```
tonystark@cs231:/home/shared$ jupyter-notebook --no-browser --port=7000
[I 01:17:20.105 NotebookApp] Writing notebook server cookie secret to /run/user/1015/jupyter/notebook_cookie_secret
[W 01:17:20.519 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not recommended.
[I 01:17:20.550 NotebookApp] JupyterLab beta preview extension loaded from /home/shared/anaconda3/lib/python3.6/site-packages/jupyterlab
[I 01:17:20.550 NotebookApp] JupyterLab application directory is /home/shared/anaconda3/share/jupyter/lab
[I 01:17:20.555 NotebookApp] Serving notebooks from local directory: /home/shared
[I 01:17:20.555 NotebookApp] 0 active kernels
[I 01:17:20.555 NotebookApp] The Jupyter Notebook is running at:
[I 01:17:20.555 NotebookApp] http://[all ip addresses on your system]:7000/?token=aad408a5bcc56f8a7d79db4e144507537e4cf927bd1ab6bc
[I 01:17:20.555 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 01:17:20.556 NotebookApp]

    Copy/paste this URL into your browser when you connect for the first time,
    to login with a token:
        http://localhost:7000/?token=aad408a5bcc56f8a7d79db4e144507537e4cf927bd1ab6bc
```

The important line (underscored in red) has the token for you to login from laptop. Replace the “localhost” part with your external IP address created in prior steps. In our example, the URL should be

```
http://35.185.240.182:7000/?token=aad408a5bcc56f8a7d79db4e144507537e4cf927bd1ab6bc
```

If there is no token, simply go to `http://35.185.240.182:7000`.

If you visit the above URL on your local browser, you should see something like the screen below.

Submission: Transferring Files From Your Instance To Your Computer

When you are done with your assignments, run the submission script in your assignment folder to make a zip file. Please refer to specific instructions for each assignment.

Once you create the zip file, e.g. `assignment1.zip`, you will transfer the file from GCE instance to your local laptop. There is an easy command for this purpose:

```
gcloud compute scp <user>@<instance-name>:/path/to/assignment1.zip /local/path
```

For example, to download files from our instance to the current folder:

```
gcloud compute scp tonystark@cs231:/home/shared/assignment1.zip .
```

The transfer works in both directions. To upload a file to GCE:

```
gcloud compute scp /my/local/file tonystark@cs231:/home/shared/
```

Another (perhaps easier) option proposed by a student is to directly download the zip file from Jupyter. After running the submission script and creating assignment1.zip, you can download that file directly from Jupyter. To do this, go to Jupyter Notebook and click on the zip file, which will be downloaded to your local computer.

BIG REMINDER: Make sure you stop your instances!

Don't forget to stop your instance when you are done (by clicking on the stop button at the top of the page showing your instances). You can restart your instance and the downloaded software will still be available.

We have seen students who left their instances running for many days and ran out of credits. You will be charged per hour when your instance is running. This includes code development time. We encourage you to read up on Google Cloud, regularly keep track of your credits and not solely rely on our tutorials.

---

cs231n
cs231n
karpathy@cs.stanford.edu