

## Lecture 07

# Cloud Computing

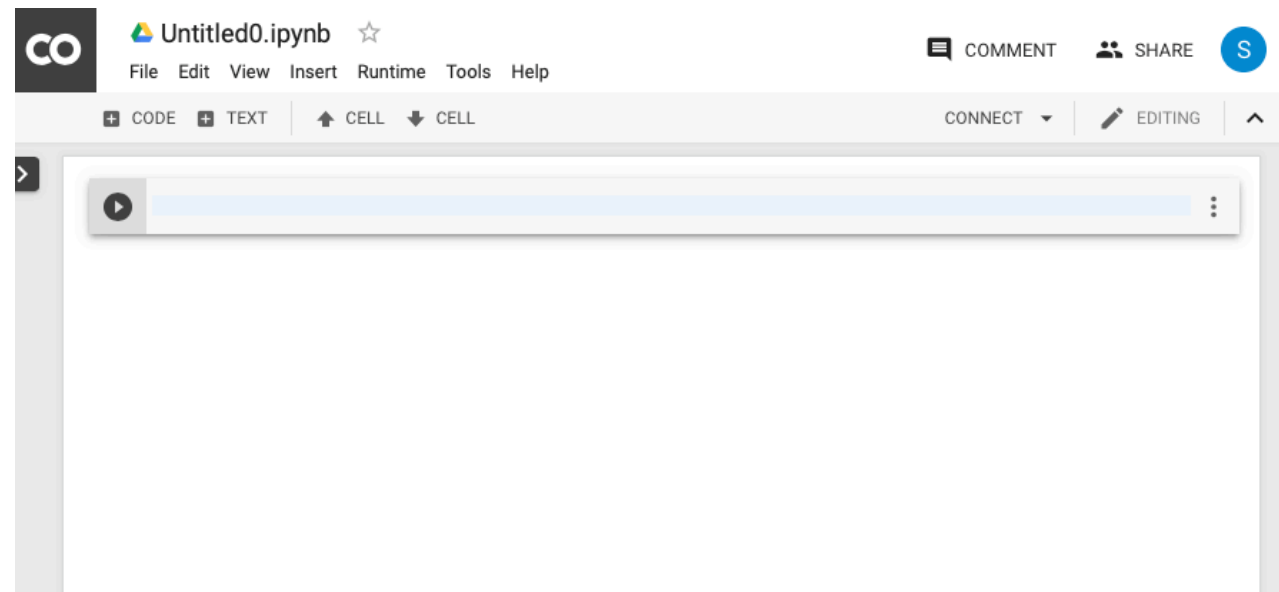
STAT 479: Deep Learning, Spring 2019

Sebastian Raschka

<http://stat.wisc.edu/~sraschka/teaching/stat479-ss2019/>

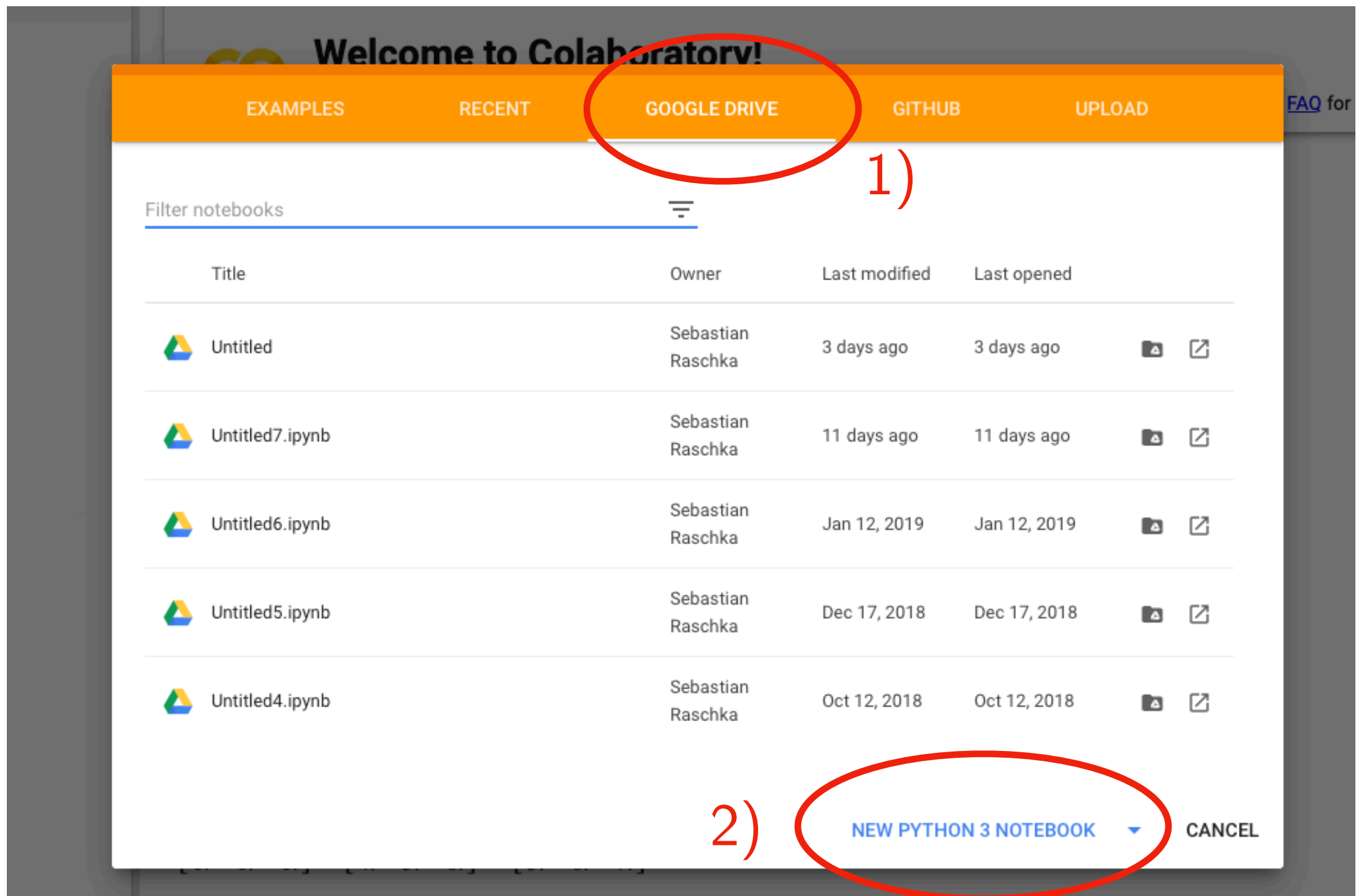
# Option 1: Google Colab

<https://colab.research.google.com>



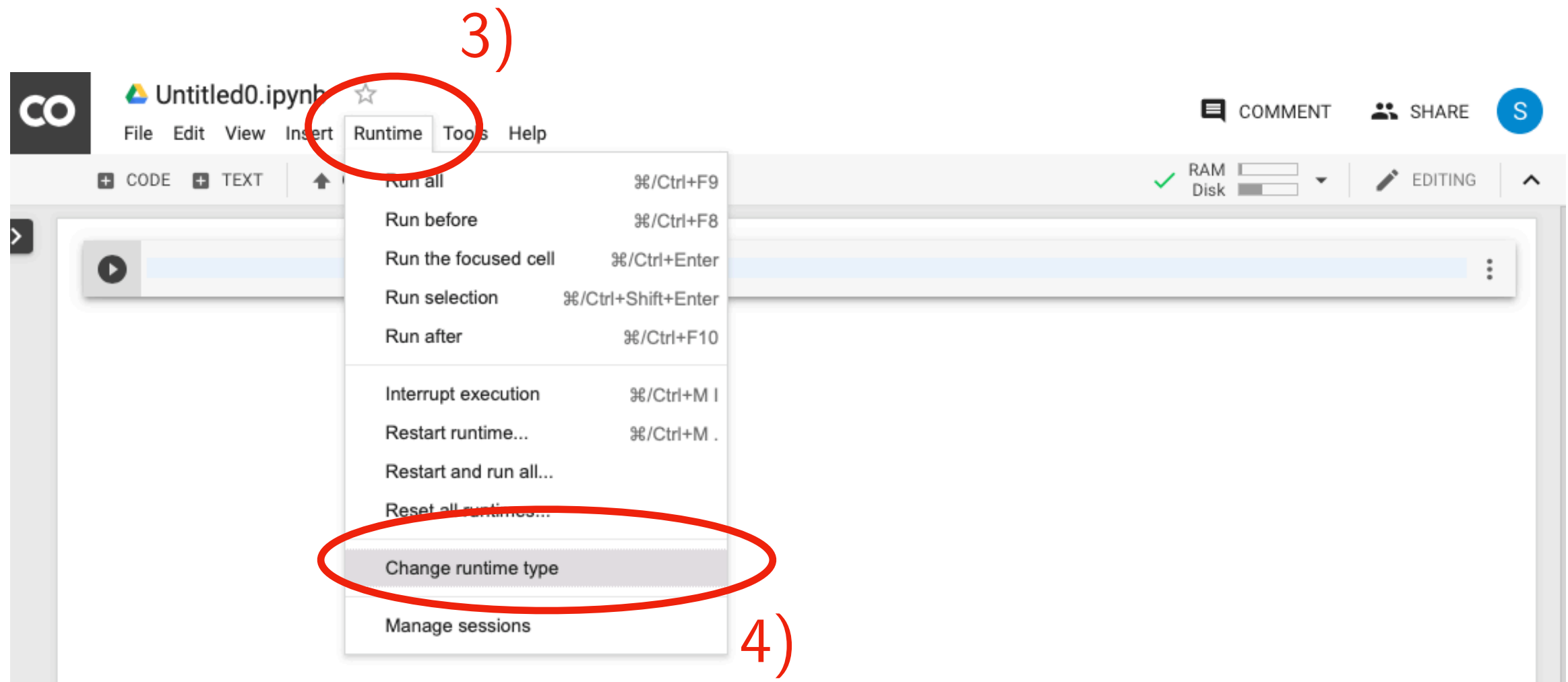
- Free Google-flavored Jupyter Notebooks in the Cloud
- For each notebook, they spin up a custom (Linux-based) computing instance
- Computations limited to ~12 h though; you won't lose your notebook, but computations will be interrupted
- Maybe useful for quick testing/experimenting/sharing (but maybe tedious as you need to reinstall packages each time)

# Option 1: Google Colab



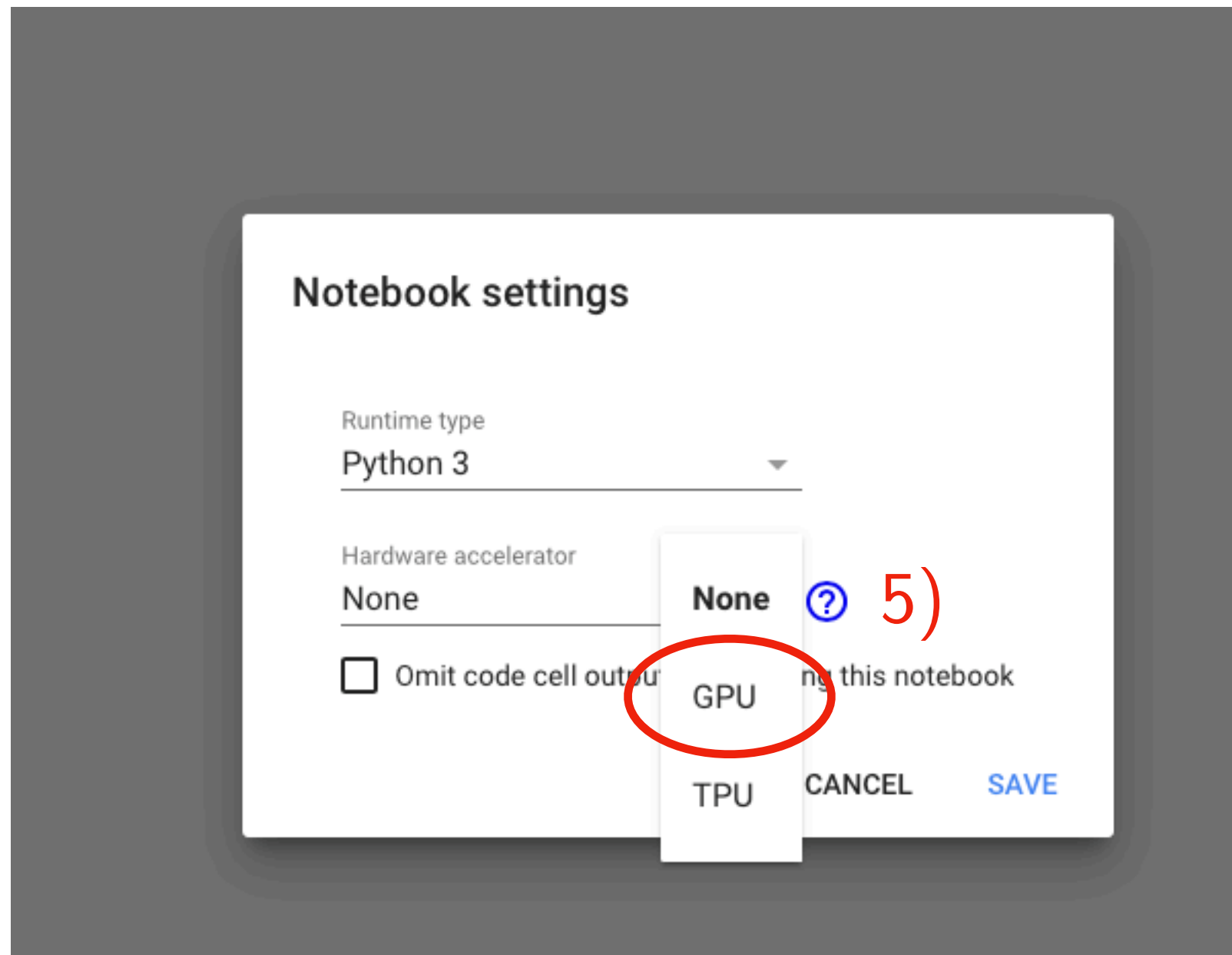
Menu appears if you visit <https://colab.research.google.com>

# Option 1: Google Colab



Follow these steps for running code on GPU later (default is CPU)

# Option 1: Google Colab



Follow these steps for running code on GPU later (default is CPU)

# Option 1: Google Colab



The screenshot shows the Google Colab interface. At the top, there's a header with the Colab logo, the file name 'Untitled0.ipynb', and a star icon. Below this is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. On the right side of the header, there are icons for 'COMMENT', 'SHARE', and a user profile icon with the letter 'S'. Below the header, there's a toolbar with buttons for '+ CODE', '+ TEXT', '↑ CELL', and '↓ CELL'. To the right of the toolbar, there's a status bar showing 'RAM' and 'Disk' usage, a green checkmark, and an 'EDITING' button. The main area of the notebook contains a code cell with the following code:

```
[1] import torch

[2] torch.__version__
```

The output of the second cell is displayed below the code: `'1.0.1.post2'`. At the bottom of the code cell, there's a play button icon and a progress bar.

- This is NEW! It appears that PyTorch is already pre-installed now (it wasn't always the case)

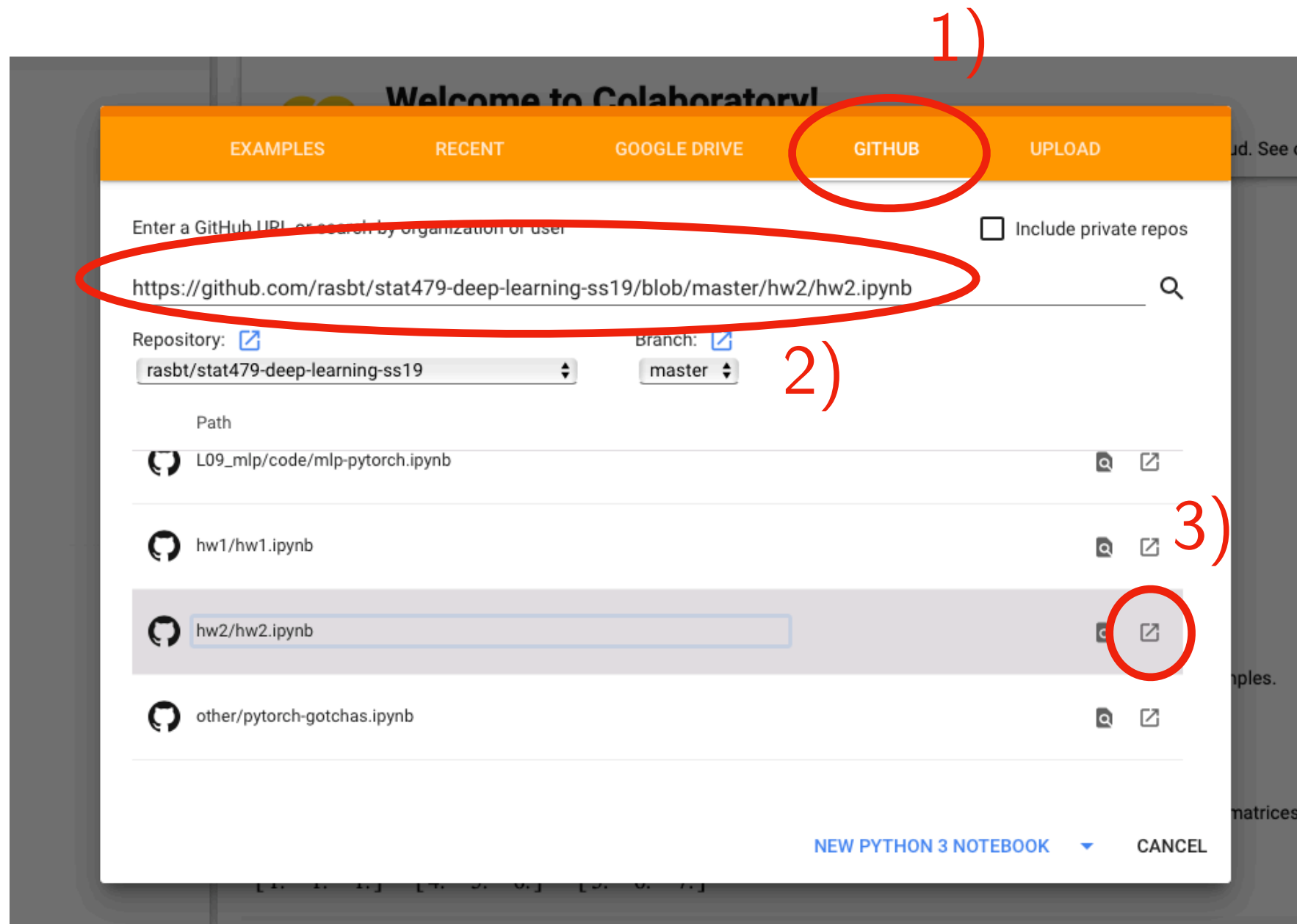
# Option 1: Google Colab



The screenshot shows the Google Colab interface for a notebook titled 'Untitled0.ipynb'. The top bar includes the Colab logo, a star icon, and a menu with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. On the right, there are 'COMMENT' and 'SHARE' buttons, and a user profile icon with the letter 'S'. Below the top bar, there's a toolbar with '+ CODE', '+ TEXT', '↑ CELL', and '↓ CELL' buttons. On the far right of the toolbar, there's a green checkmark, 'RAM' and 'Disk' usage indicators, and an 'EDITING' button. The notebook content consists of three code cells. The first cell contains `[1] import torch`. The second cell contains `[2] torch.__version__` and shows the output `'1.0.1.post2'`. The third cell contains `!pip install numpy` and has a play button icon on the left and a three-dot menu on the right.

- In any case, if you'd like/need to install packages, you can do it as shown in the example above
- Note that in Jupyter Notebooks, the "!" indicates that what follows on that line is a "shell command" (you can think of a "shell" as the Linux & macOS command-line terminal, e.g., a Bash Shell)

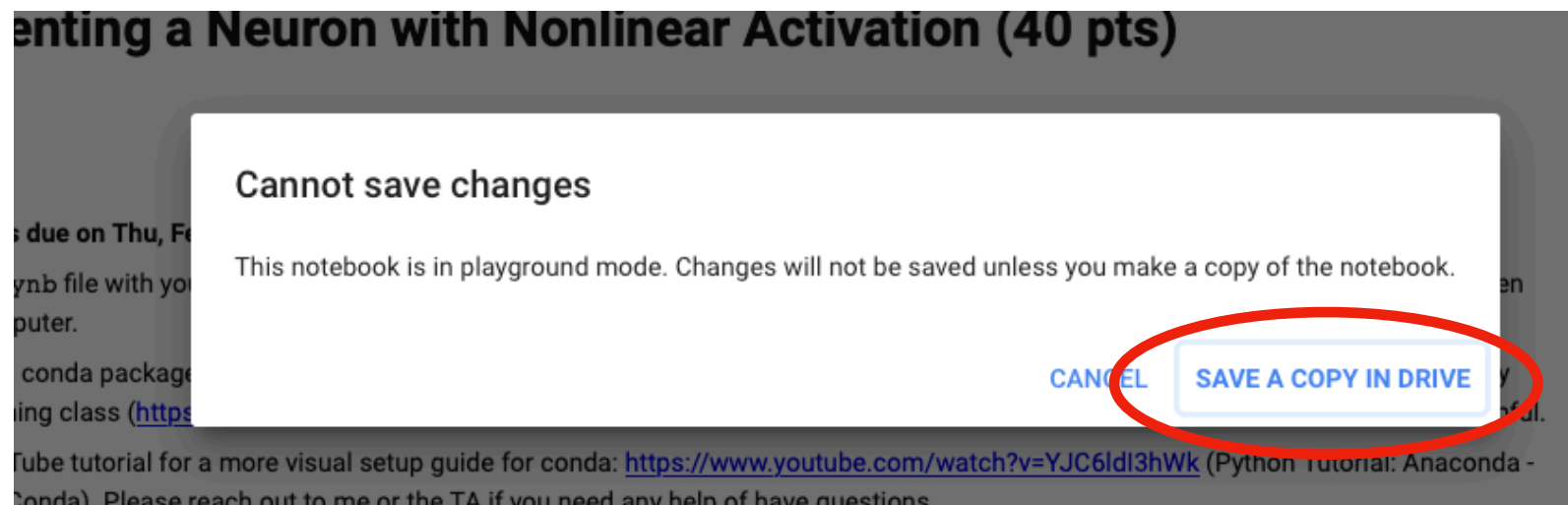
# Option 1: Google Colab



- You can also upload Notebooks or directly paste GitHub links to notebooks



# Option 1: Google Colab



4)

When you import a Notebook from a GitHub link, make sure to save it in your Google Drive if you plan to make edits, otherwise it will be gone later

# Option 1: Google Colab

```
/usr/local/lib/python3.6/dist-packages/pandas/io/parsers.py in __init__(self, f, engine, **kwds)
    816         self.options['has_index_names'] = kwds['has_index_names']
    817
--> 818         self._make_engine(self.engine)
    819
    820     def close(self):

/usr/local/lib/python3.6/dist-packages/pandas/io/parsers.py in _make_engine(self, engine)
    1047     def _make_engine(self, engine='c'):
    1048         if engine == 'c':
-> 1049             self._engine = CParserWrapper(self.f, **self.options)
    1050         else:
    1051             if engine == 'python':

/usr/local/lib/python3.6/dist-packages/pandas/io/parsers.py in __init__(self, src, **kwds)
    1693         kwds['allow_leading_cols'] = self.index_col is not False
    1694
-> 1695         self._reader = parsers.TextReader(src, **kwds)
    1696
    1697         # XXX

pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader.__cinit__()

pandas/_libs/parsers.pyx in pandas._libs.parsers.TextReader._setup_parser_source()

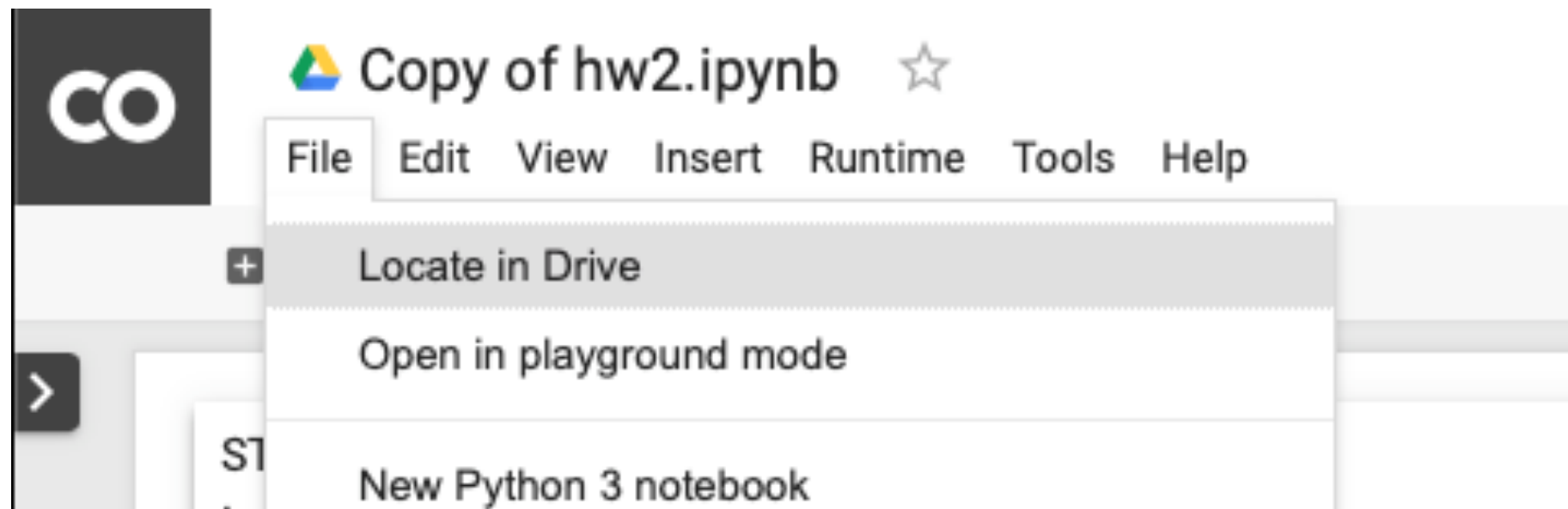
FileNotFoundError: File b'./datasets/iris.data' does not exist
```

SEARCH STACK OVERFLOW

If you'd run the HW2 notebook, you'd likely encounter this error.  
This is because it can't find the dataset via the specified, relative link ...

# Option 1: Google Colab

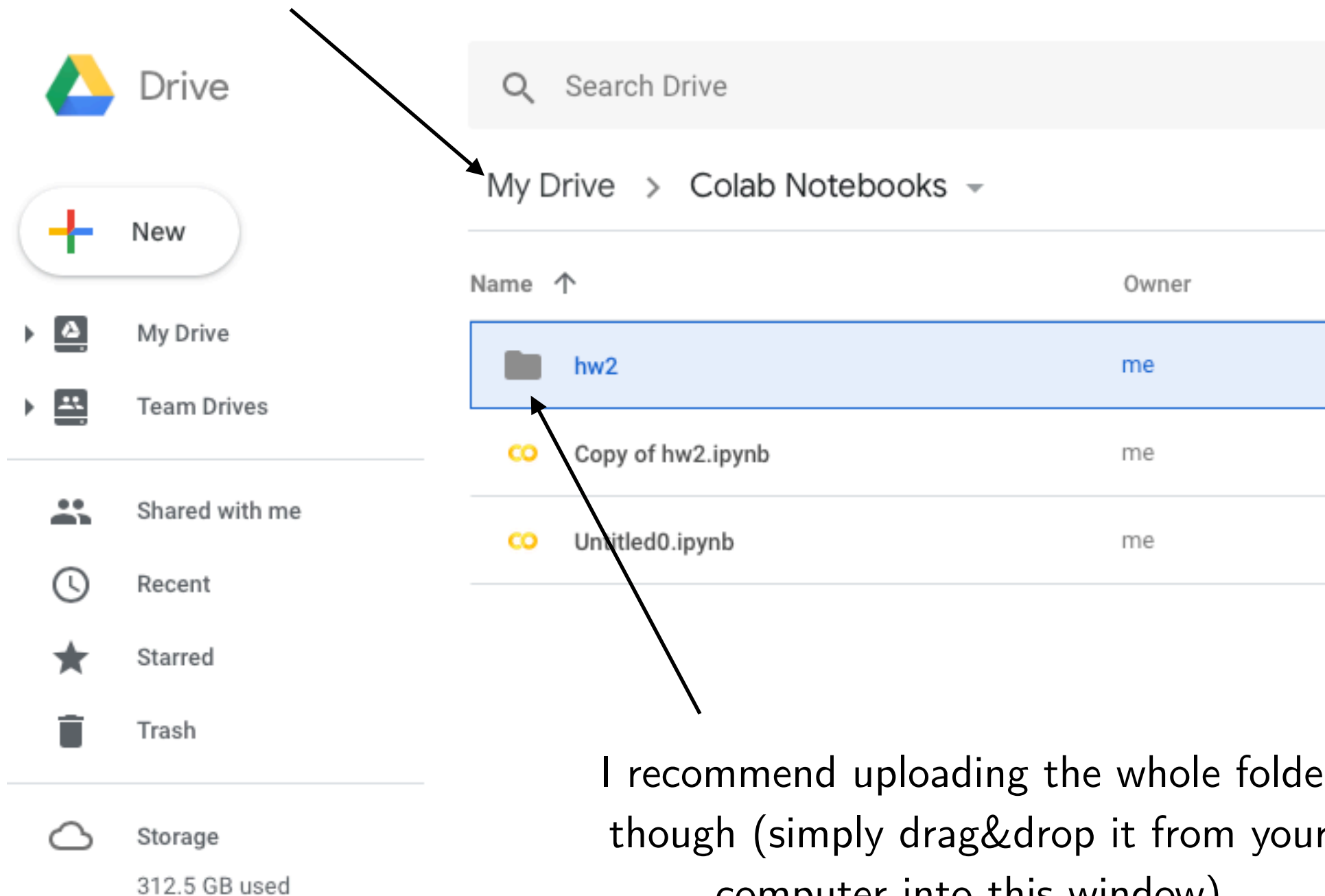
... you'd need to get the datafile into the same location as the notebook\*.  
First, locate the position of the saved Notebook in your Google Drive:



\*technically, it is also possible to load CSV files via `pandas.read_csv`, but getting the dataset onto Google Drive may be generally useful, e.g., for working with more complex datasets later.

# Option 1: Google Colab

Notebooks are usually in a directory called "My Drive/Colab Notebooks"



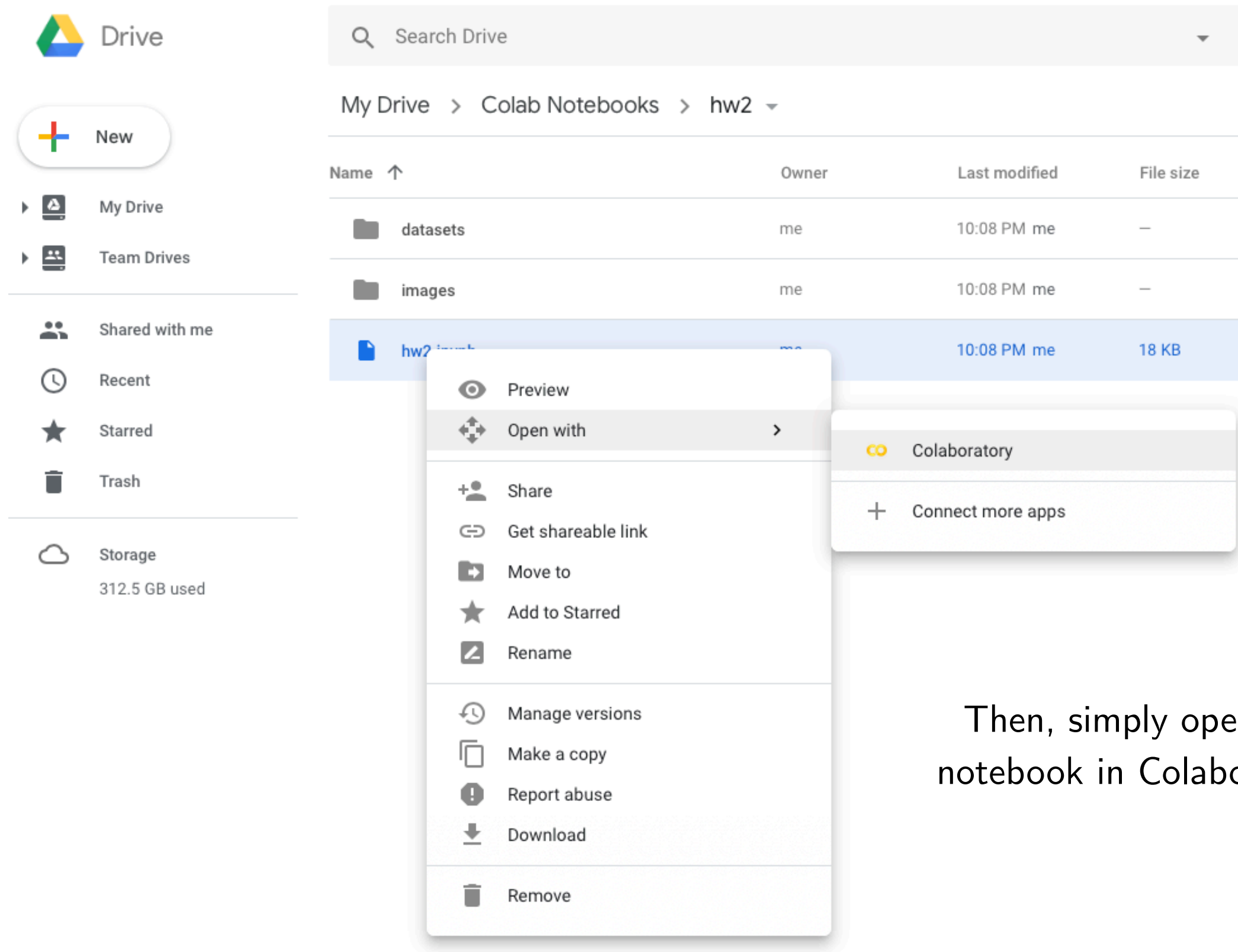
The screenshot shows the Google Drive interface. On the left is a sidebar with the 'Drive' logo, a 'New' button, and navigation links for 'My Drive', 'Team Drives', 'Shared with me', 'Recent', 'Starred', and 'Trash'. At the bottom of the sidebar, it shows 'Storage' with '312.5 GB used'. The main area has a search bar at the top. Below it, the breadcrumb path 'My Drive > Colab Notebooks' is shown. A table lists the contents of the 'Colab Notebooks' folder:

Name	Owner
hw2	me
Copy of hw2.ipynb	me
Untitled0.ipynb	me

Two arrows are present: one pointing from the text 'My Drive/Colab Notebooks' to the breadcrumb path, and another pointing from the text 'I recommend uploading the whole folder though...' to the 'hw2' folder entry in the table.

I recommend uploading the whole folder though (simply drag&drop it from your computer into this window)

# Option 1: Google Colab



The screenshot shows the Google Drive interface. On the left is a sidebar with navigation options: 'New', 'My Drive', 'Team Drives', 'Shared with me', 'Recent', 'Starred', 'Trash', and 'Storage' (312.5 GB used). The main area shows a breadcrumb path 'My Drive > Colab Notebooks > hw2'. Below this is a table of files:

Name	Owner	Last modified	File size
datasets	me	10:08 PM me	—
images	me	10:08 PM me	—
hw2	me	10:08 PM me	18 KB

A context menu is open over the 'hw2' file, listing various actions. The 'Open with' option is selected, and a sub-menu is displayed showing 'Colaboratory' as the chosen application.

- Preview
- Open with >
- Share
- Get shareable link
- Move to
- Add to Starred
- Rename
- Manage versions
- Make a copy
- Report abuse
- Download
- Remove

The sub-menu for 'Open with' includes:

- Colaboratory
- Connect more apps

Then, simply open the notebook in Colaboratory.

# Option 1: Google Colab

Unfortunately, there's some extra step required: mounting your Google Drive to the computer that now runs the Notebook. You need to execute the following code:

1)

```
from google.colab import drive
drive.mount('/content/drive')
```

... Go to this URL in a browser: [https://accounts.google.com/o/oauth2/auth?client\\_id=9473189](https://accounts.google.com/o/oauth2/auth?client_id=9473189)

Enter your authorization code:

Then, click on the link and enter it in the field above

2)

```
from google.colab import drive
drive.mount('/content/drive')
```

... Go to this URL in a browser: [https://accounts.google.com/o/oauth2/auth?client\\_id=9473189](https://accounts.google.com/o/oauth2/auth?client_id=9473189)

Enter your authorization code:

.....

3)

```
from google.colab import drive
drive.mount('/content/drive')
```

Go to this URL in a browser: [https://accounts.google.com/o/oauth2/auth?client\\_id=9473189](https://accounts.google.com/o/oauth2/auth?client_id=9473189)

Enter your authorization code:

.....

Mounted at /content/drive

Your Google Drive should now be finally mounted:

# Option 1: Google Colab

Now, you simply need to provide the correct address to the dataset inside the Notebook and it should work:

```
[16] df = pd.read_csv('/content/drive/My Drive/Colab Notebooks/hw2/datasets/iris.data', index_col=None, header=None)
      df.columns = ['x1', 'x2', 'x3', 'x4', 'y']
      df = df.iloc[50:150]
      df['y'] = df['y'].apply(lambda x: 0 if x == 'Iris-versicolor' else 1)
      df.tail()
```

	x1	x2	x3	x4	y
145	6.7	3.0	5.2	2.3	1
146	6.3	2.5	5.0	1.9	1
147	6.5	3.0	5.2	2.0	1
148	6.2	3.4	5.4	2.3	1
149	5.9	3.0	5.1	1.8	1

## Option 2: Google Cloud Instances

This is trickier and you don't have to use it for this class, but it's a useful skill and experience!

<https://console.cloud.google.com/education>

Will email a  
\$50 coupon code (per student) after class

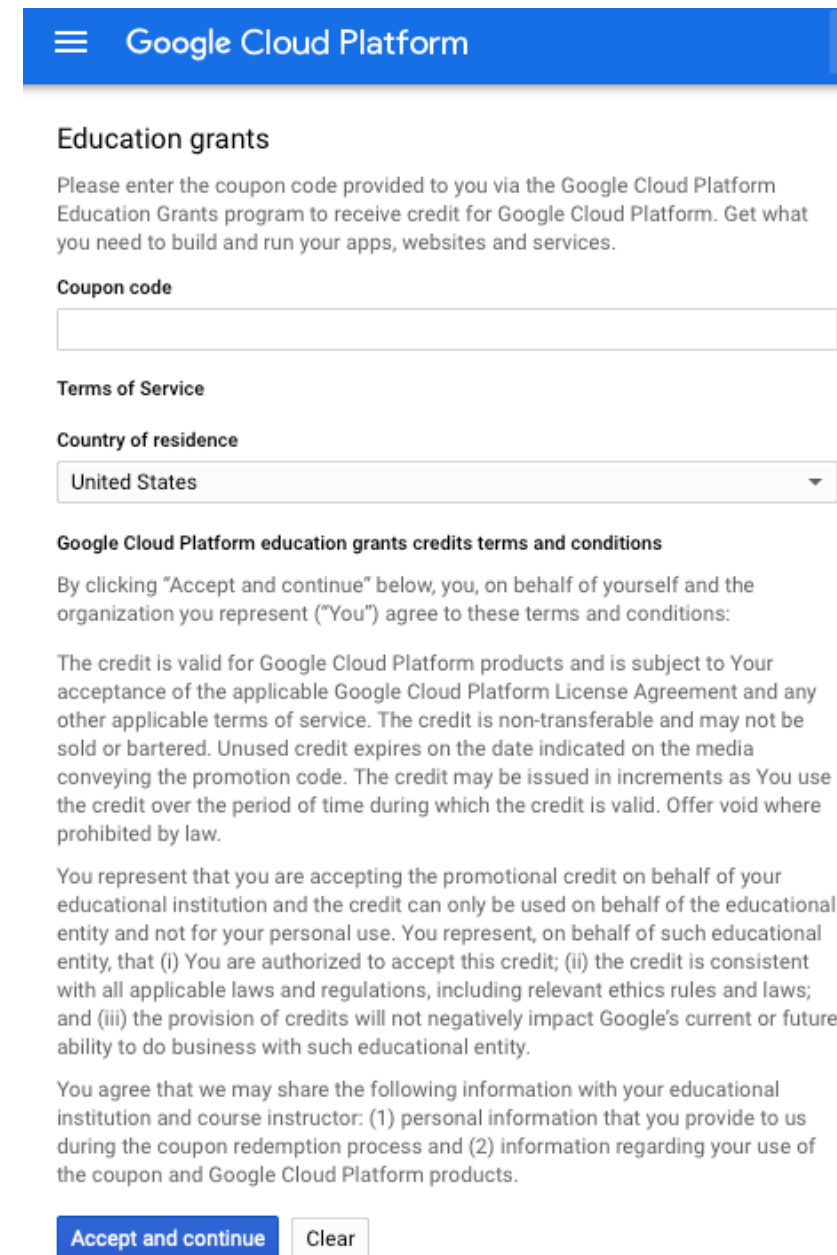


# Option 2: Google Cloud Instances

Go to the website

<https://console.cloud.google.com/education>

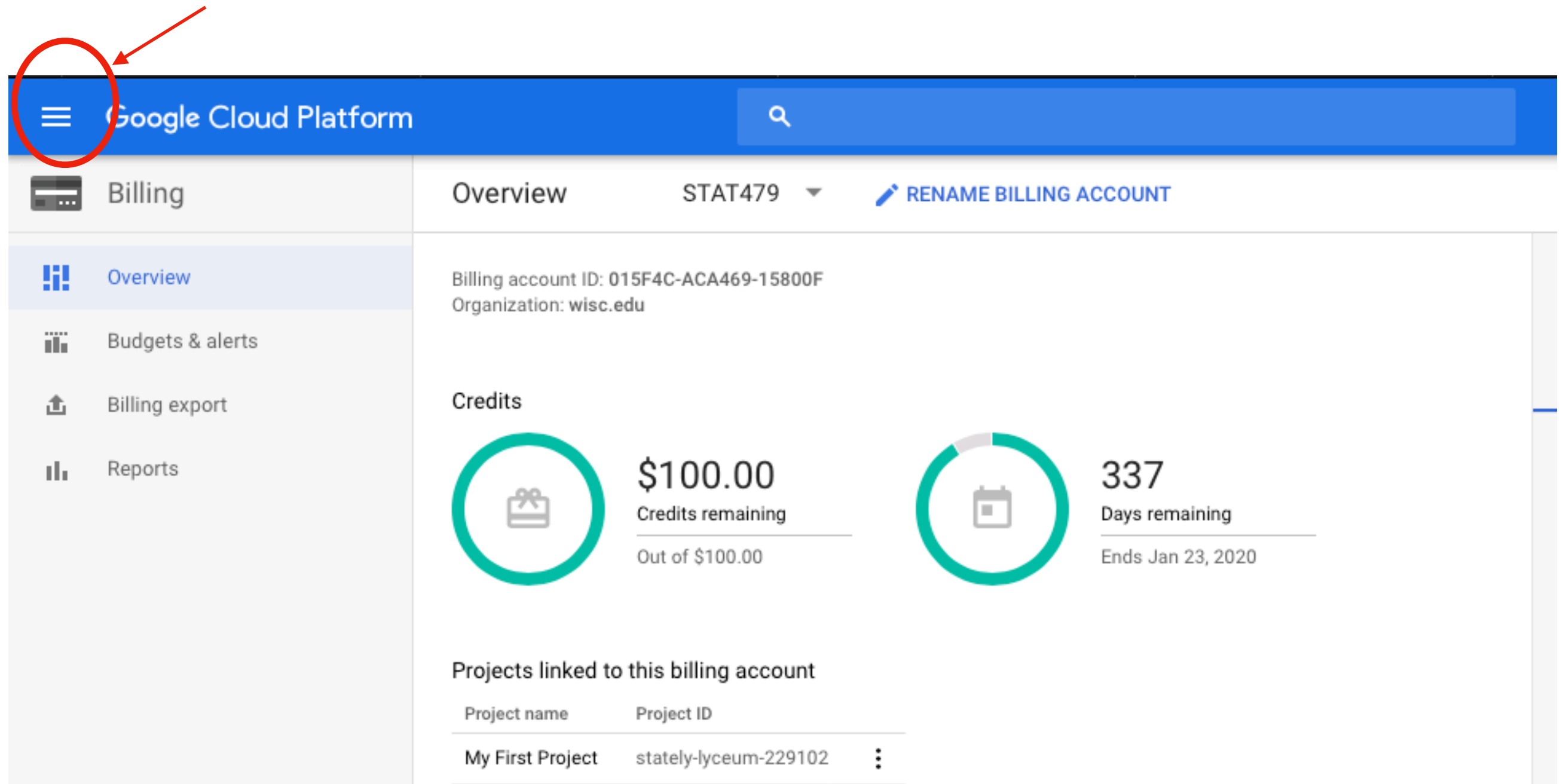
Read the terms, and accept if you agree  
(you don't have to use this platform for this class!)



The screenshot shows the Google Cloud Platform Education Grants page. At the top is a blue header with the Google Cloud Platform logo. Below the header, the page is titled "Education grants". A paragraph explains the program: "Please enter the coupon code provided to you via the Google Cloud Platform Education Grants program to receive credit for Google Cloud Platform. Get what you need to build and run your apps, websites and services." There is a text input field for the "Coupon code". Below this is a "Terms of Service" section, followed by a "Country of residence" dropdown menu currently set to "United States". The "Google Cloud Platform education grants credits terms and conditions" section follows, containing several paragraphs of legal text. At the bottom, there are two buttons: "Accept and continue" (in blue) and "Clear" (in white).

# Option 2: Google Cloud Instances

Check your credits periodically, via the billings menu that can be accessed from the main menu



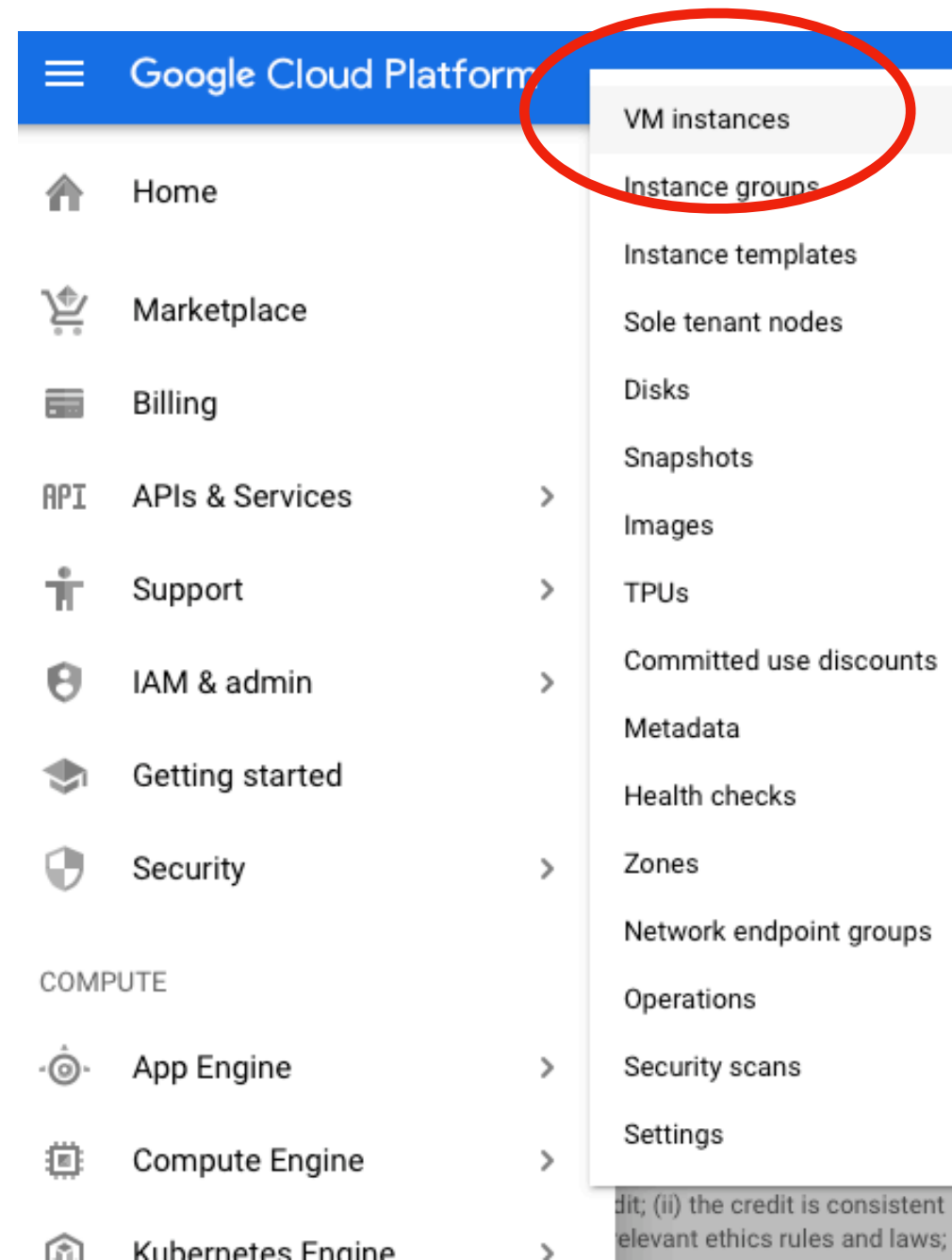
The screenshot shows the Google Cloud Platform Billing Overview page. The top navigation bar is blue and contains the Google Cloud Platform logo, a search bar, and a main menu icon (three horizontal lines) circled in red with an arrow pointing to it. Below the navigation bar, the left sidebar lists the following menu items: Billing, Overview (selected), Budgets & alerts, Billing export, and Reports. The main content area displays the following information:

- Overview** for project **STAT479**. A link to **RENAME BILLING ACCOUNT** is available.
- Billing account ID:** 015F4C-ACA469-15800F
- Organization:** wisc.edu
- Credits** section showing:
  - \$100.00** Credits remaining (Out of \$100.00)
  - 337** Days remaining (Ends Jan 23, 2020)
- Projects linked to this billing account** table:

Project name	Project ID	
My First Project	stately-lyceum-229102	⋮

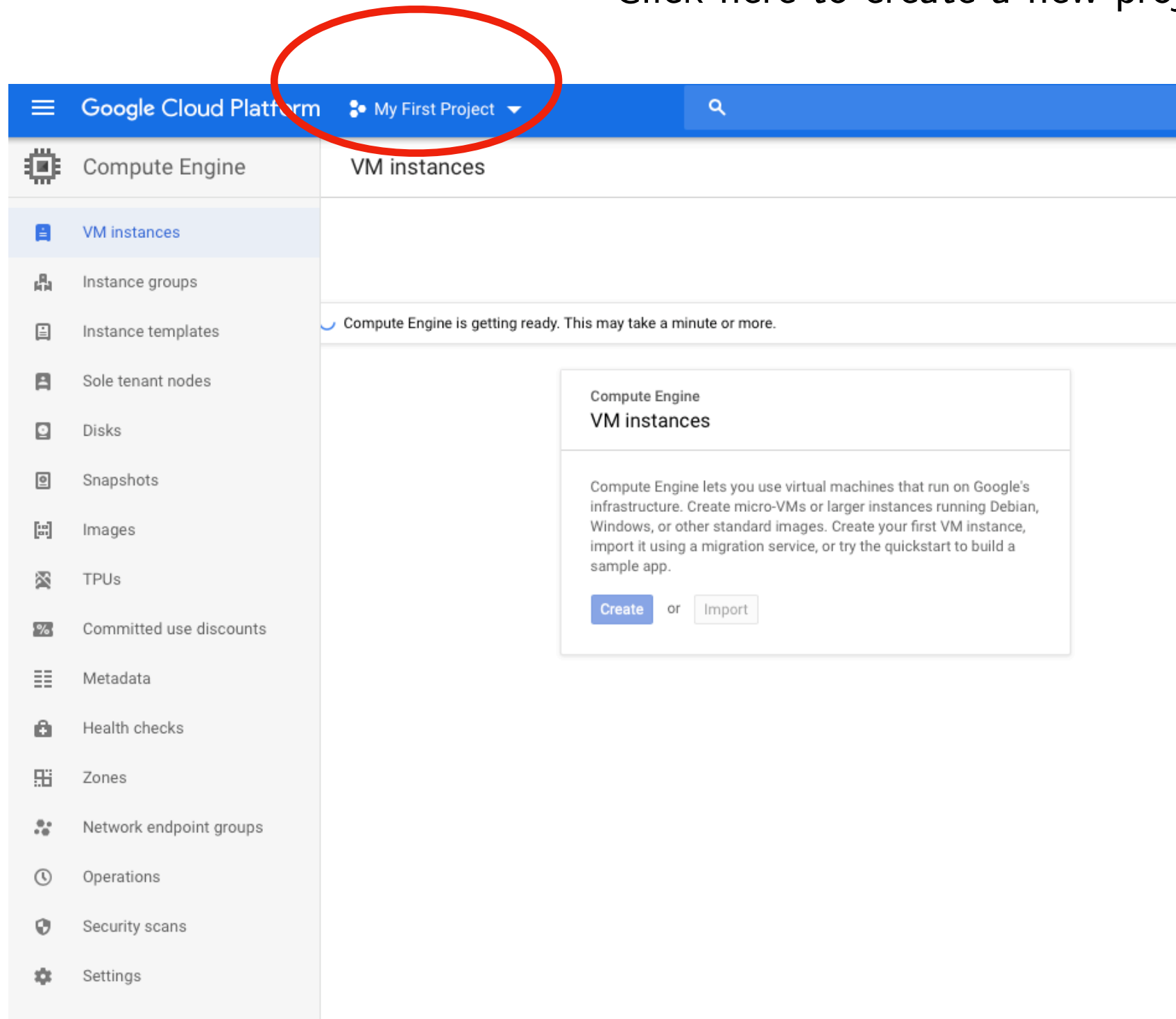
# Option 2: Google Cloud Instances

To create a new computing instance, click on VM instances



# Option 2: Google Cloud Instances

Click here to create a new project



# Option 2: Google Cloud Instances

My First Project

VM instances

You don't have permission to view the Compute project (compute.projects.get)

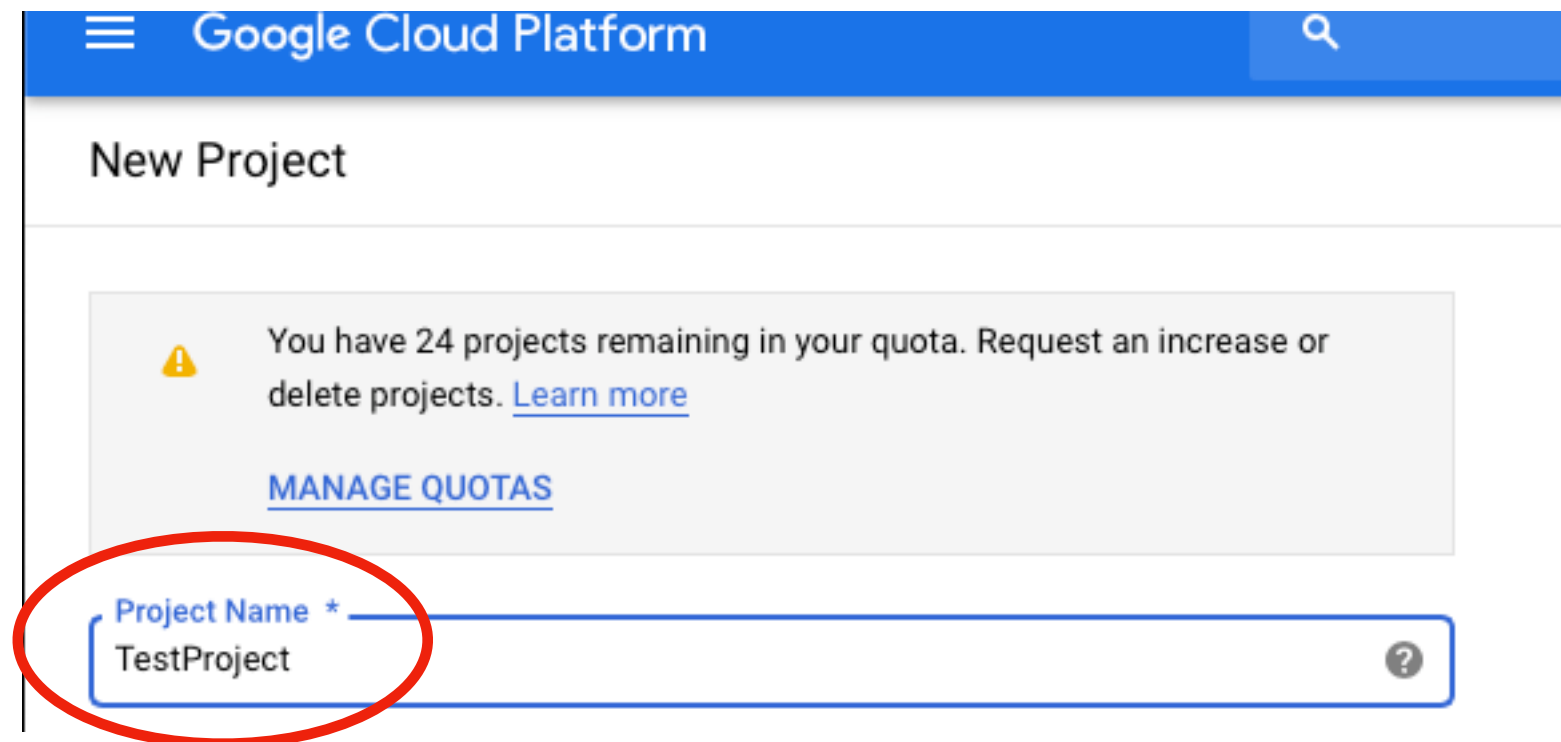
Select from **WISC.EDU** Then click here to create a new project **NEW PROJECT**

Search projects and folders

RECENT ALL

Name	ID
✓ My First Project ?	stately-lyceum-229102
📁 wisc.edu ?	532642321694

# Option 2: Google Cloud Instances



Google Cloud Platform

New Project

You have 24 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project Name \*  
TestProject

Enter a name for your project

# Option 2: Google Cloud Instances

Google Cloud Platform TestProject

## 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

**Name** myinstance-1

**Region** us-east1 (South Carolina) **Zone** us-east1-b

**Machine type**  
Customize to select cores, memory and GPUs.  
1 vCPU 3.75 GB memory [Customize](#)

**Container**  
☐ Deploy a container image to this VM instance. [Learn more](#)

**Boot disk**  
New 30 GB standard persistent disk  
Image: Ubuntu 18.04 LTS [Change](#)

**Identity and API access**  
**Service account** Compute Engine default service account  
**Access scopes**  
☒ Allow default access  
☐ Allow full access to all Cloud APIs  
☐ Set access for each API

**Firewall**  
Add tags and firewall rules to allow specific network traffic from the Internet  
☒ Allow HTTP traffic  
☒ Allow HTTPS traffic

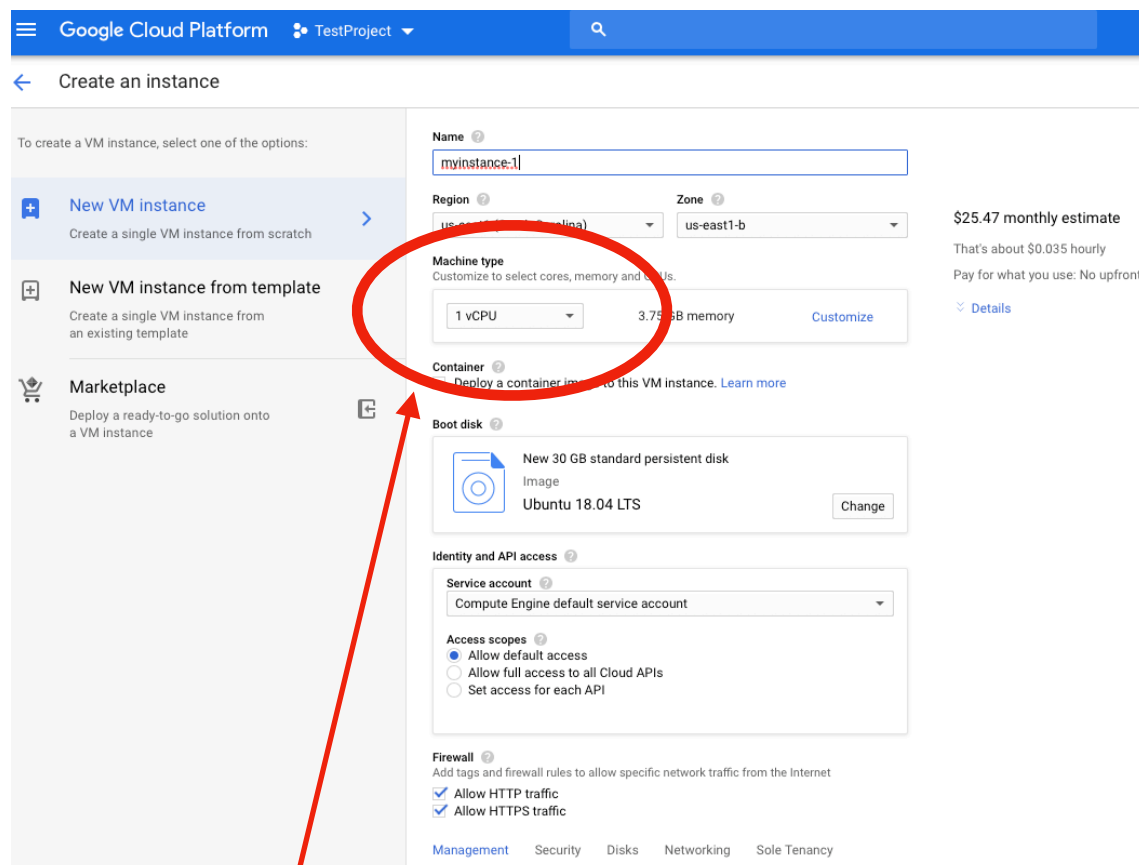
[Management](#) Security Disks Networking Sole Tenancy

**\$25.47 monthly estimate**  
That's about \$0.035 hourly  
Pay for what you use: No upfront c  
[Details](#)

I recommend trying CPUs first (cheaper) when experimenting.

Choose Ubuntu 18.04 LTS

# Option 2: Google Cloud Instances



Later, when you need it in the class, you can choose GPUs from here

Name <sup>?</sup>

myinstance-1

Region <sup>?</sup> us-east1 (South Carolina) Zone <sup>?</sup> us-east1-b

**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 <sup>?</sup>

**CPU platform** <sup>?</sup>

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 GPU type

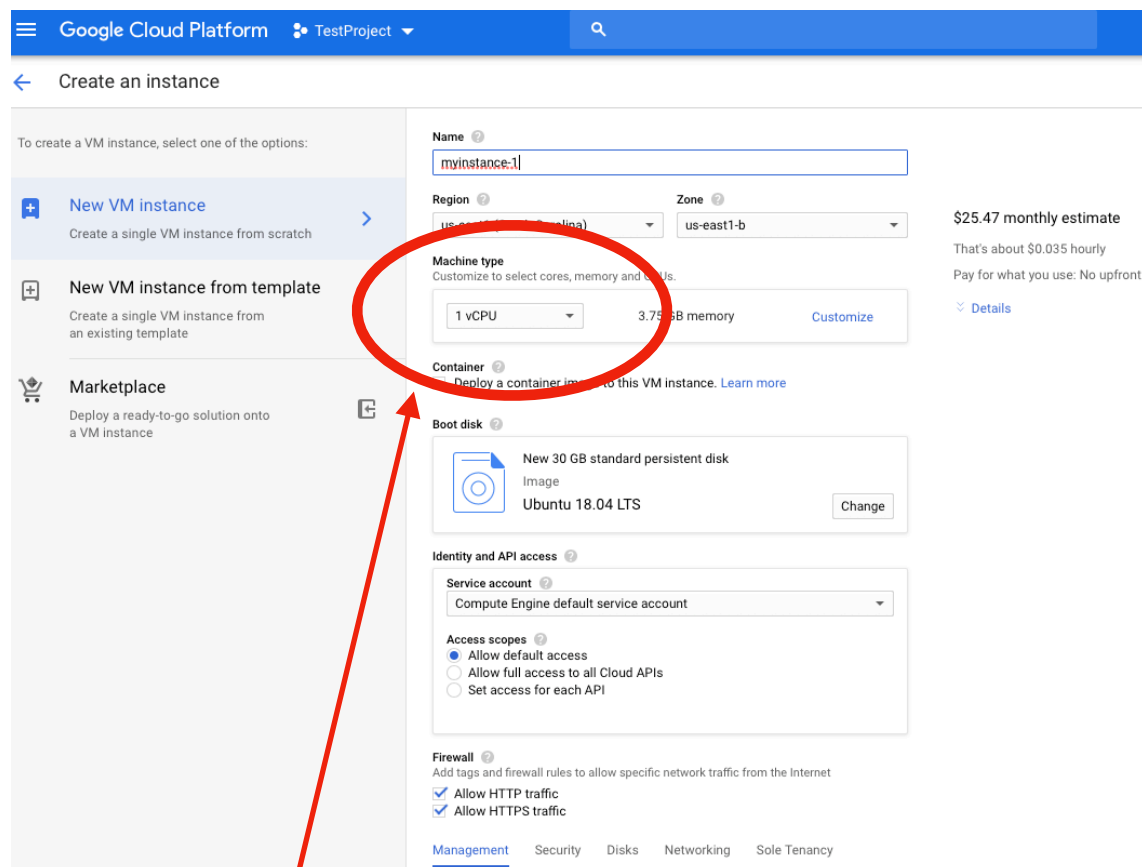
1 NVIDIA Tesla P100

**Machines with GPUs can't migrate on host maintenance**

[Choosing a machine type](#) ↗



# Option 2: Google Cloud Instances



Name <sup>?</sup>

myinstance-1

Region <sup>?</sup> Zone <sup>?</sup>

us-east1 (South Carolina) us-east1-b

**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 <sup>?</sup>

**CPU platform** <sup>?</sup>

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** **GPU type**

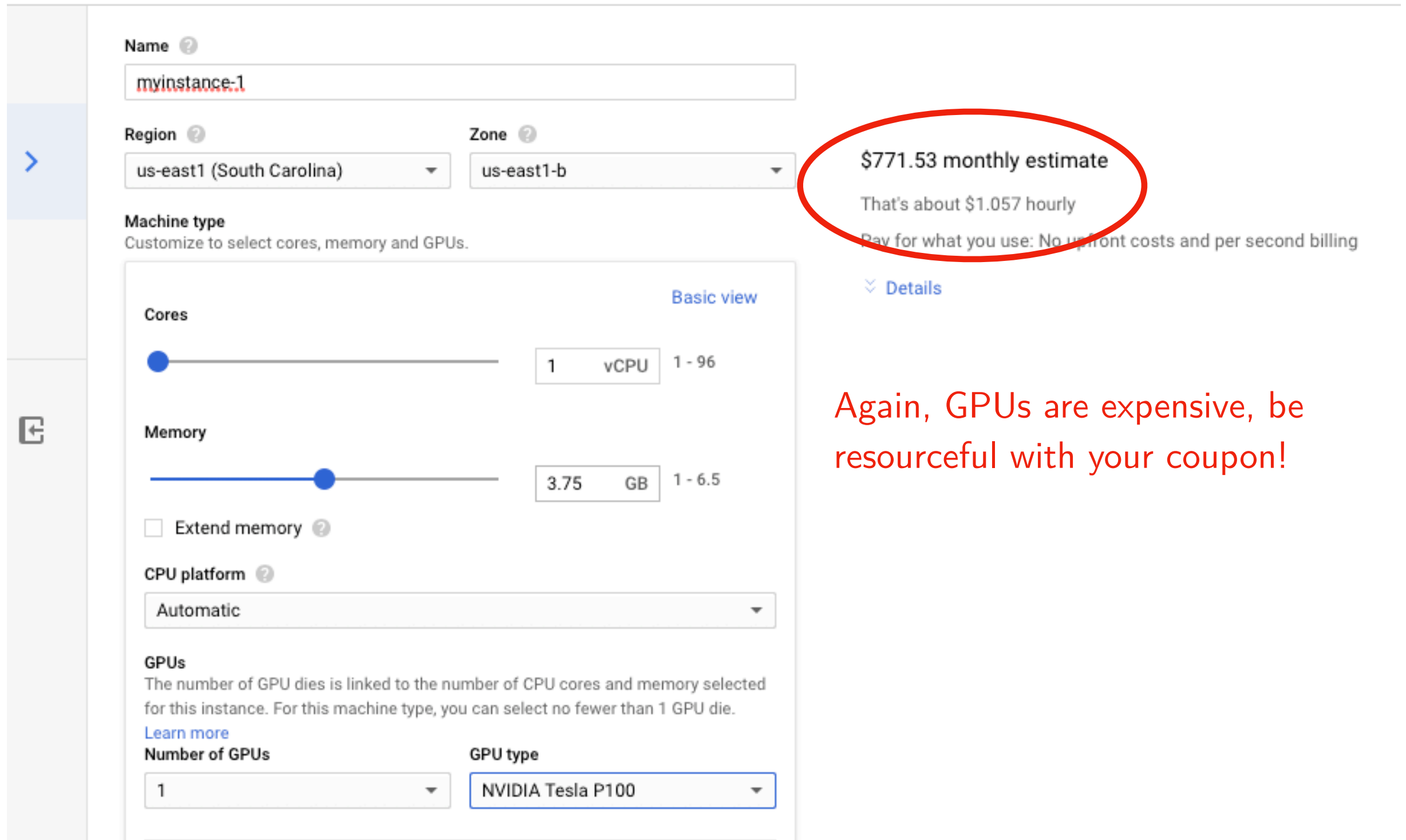
1 NVIDIA Tesla P100

**Machines with GPUs can't migrate on host maintenance**

[Choosing a machine type](#) ↗

Later, when you need it in the class, you can choose GPUs from here

# Option 2: Google Cloud Instances



The screenshot shows the Google Cloud Platform instance configuration interface. The 'Name' field is 'myinstance-1'. The 'Region' is 'us-east1 (South Carolina)' and the 'Zone' is 'us-east1-b'. The 'Machine type' section is expanded, showing 'Cores' set to 1 vCPU and 'Memory' set to 3.75 GB. The 'CPU platform' is 'Automatic'. The 'GPUs' section shows 'Number of GPUs' set to 1 and 'GPU type' set to 'NVIDIA Tesla P100'. A red circle highlights the pricing information: '\$771.53 monthly estimate', 'That's about \$1.057 hourly', and 'Pay for what you use: No upfront costs and per second billing'. A red text overlay on the right says 'Again, GPUs are expensive, be resourceful with your coupon!'.

Name <sup>?</sup>

myinstance-1

Region <sup>?</sup> us-east1 (South Carolina) Zone <sup>?</sup> us-east1-b

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 <sup>?</sup>

CPU platform <sup>?</sup>

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 GPU type

1 NVIDIA Tesla P100

\$771.53 monthly estimate


That's about \$1.057 hourly

Pay for what you use: No upfront costs and per second billing

[Details](#)

Again, GPUs are expensive, be resourceful with your coupon!

# Option 2: Google Cloud Instances

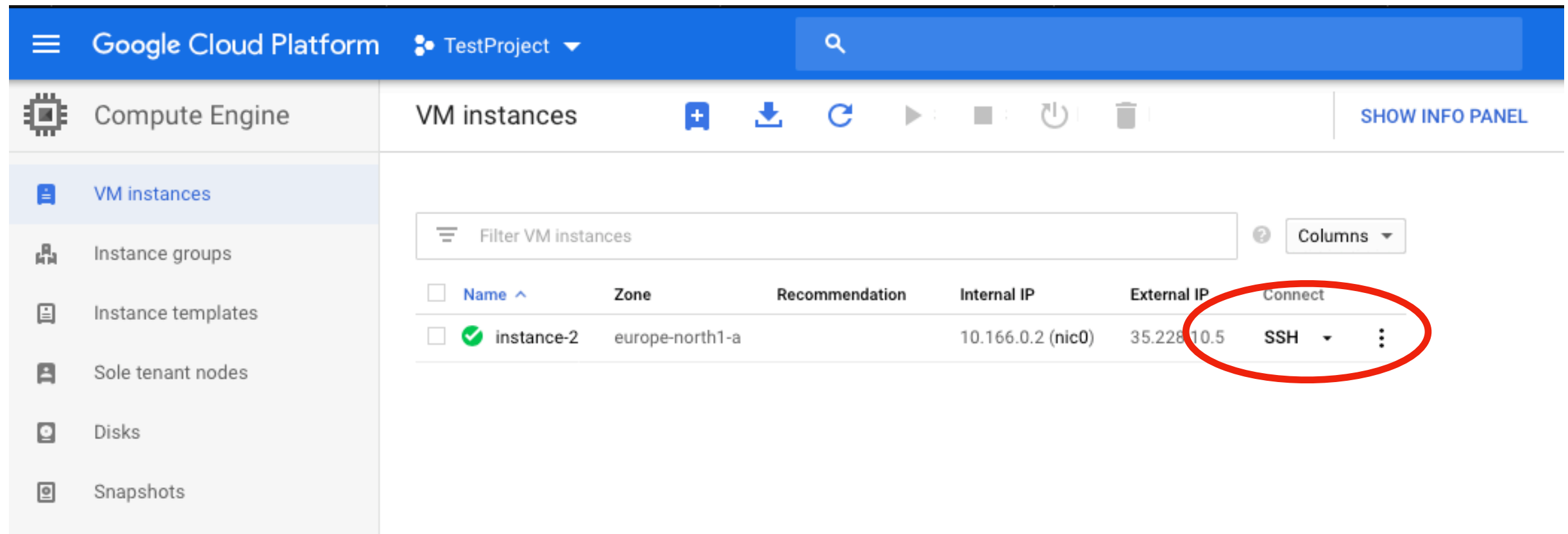


The screenshot shows the Google Cloud VM instances management interface. At the top, there is a search bar labeled 'Filter VM instances' and a 'Columns' dropdown menu. Below this is a table with columns: Name, Zone, Recommendation, Internal IP, External IP, and Connect. The table contains one instance named 'myinstance-1'. A red exclamation mark icon is next to the instance name, and a tooltip is displayed over it with the text: 'Quota 'GPUS\_ALL\_REGIONS' exceeded. Limit: 0.0 globally.' The 'External IP' column for this instance shows 'None'. The 'Connect' column has two icons: a refresh icon and a delete icon.

Name ^	Zone	Recommendation	Internal IP	External IP	Connect
! Quota 'GPUS_ALL_REGIONS' exceeded. Limit: 0.0 globally. ! myinstance-1				None	 
				None	 

It may unfortunately happen that sometimes all GPUs are busy (used by other people)

# Option 2: Google Cloud Instances

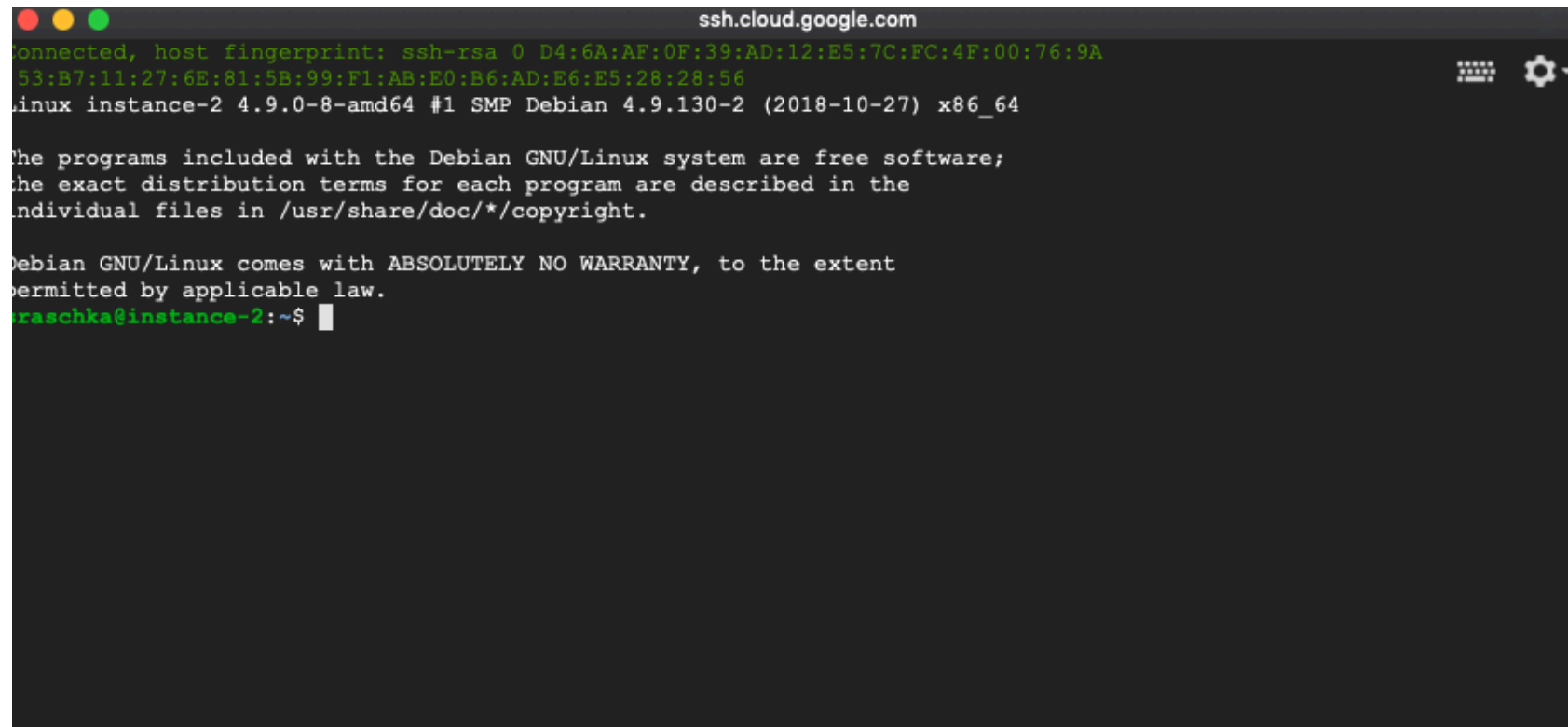


The screenshot shows the Google Cloud Platform interface for VM instances. The top navigation bar includes the Google Cloud Platform logo, the project name 'TestProject', and a search bar. The left sidebar lists various resources: Compute Engine, VM instances (selected), Instance groups, Instance templates, Sole tenant nodes, Disks, and Snapshots. The main content area is titled 'VM instances' and features a toolbar with icons for adding, downloading, refreshing, playing, pausing, and deleting instances, along with a 'SHOW INFO PANEL' link. Below the toolbar is a table of VM instances. The table has columns for Name, Zone, Recommendation, Internal IP, External IP, and Connect. A single instance, 'instance-2', is listed in the 'europa-north1-a' zone. The 'Connect' column for this instance shows an 'SSH' button, which is circled in red.

Name	Zone	Recommendation	Internal IP	External IP	Connect
<input type="checkbox"/> instance-2	europa-north1-a		10.166.0.2 (nic0)	35.228.10.5	SSH

Anyways, once your instance runs, you can click in SSH to log in

# Option 2: Google Cloud Instances

A screenshot of a terminal window titled 'ssh.cloud.google.com'. The terminal shows a connection to a Linux instance. The output includes a host fingerprint, the instance name 'linux instance-2', the kernel version '4.9.0-8-amd64', and the OS 'Debian 4.9.130-2'. It also displays the Debian GNU/Linux license text. The prompt is 'sebastian@instance-2:~\$' with a cursor. There are window control buttons (red, yellow, green) in the top left and a settings gear icon in the top right.

```
ssh.cloud.google.com
connected, host fingerprint: ssh-rsa 0 D4:6A:AF:0F:39:AD:12:E5:7C:FC:4F:00:76:9A
53:B7:11:27:6E:81:5B:99:F1:AB:E0:B6:AD:E6:E5:28:28:56
linux instance-2 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
sebastian@instance-2:~$
```

It will basically be a Linux terminal

# Option 2: Google Cloud Instances

First, I recommend installing conda. In this step, we download it first



```
ssh.cloud.google.com
sraschka@instance-2:~$ wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh
--2019-02-20 07:02:13-- https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh
Resolving repo.continuum.io (repo.continuum.io)... 104.16.19.10, 104.16.18.10, 2606:4700::6810:130a, ...
Connecting to repo.continuum.io (repo.continuum.io)|104.16.19.10|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 69826864 (67M) [application/x-sh]
Saving to: 'Miniconda3-latest-Linux-x86_64.sh'

Miniconda3-latest-Linux-x86_ 100%[=====>] 66.59M 157MB/s in 0.4s

2019-02-20 07:02:14 (157 MB/s) - 'Miniconda3-latest-Linux-x86_64.sh' saved [69826864/69826864]

sraschka@instance-2:~$
```

# Option 2: Google Cloud Instances

There is some package missing that we need for installing conda

```
sraschka@instance-2:~$ sudo apt-get install bzip2
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  bzip2-doc
The following NEW packages will be installed:
  bzip2
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 47.5 kB of archives.
After this operation, 188 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian stretch/main amd64 bzip2 amd64 1.0.6-8.1 [47.5 kB]
Fetched 47.5 kB in 0s (761 kB/s)
Selecting previously unselected package bzip2.
(Reading database ... 34432 files and directories currently installed.)
Preparing to unpack .../bzip2_1.0.6-8.1_amd64.deb ...
Unpacking bzip2 (1.0.6-8.1) ...
Setting up bzip2 (1.0.6-8.1) ...
Processing triggers for man-db (2.7.6.1-2) ...
```

# Option 2: Google Cloud Instances

After bzip2 is installed, you can run the installer for Miniconda:

A terminal window with a dark title bar containing three colored window control buttons (red, yellow, green) on the left and the text 'ssh.cloud.google.com' on the right. The terminal content shows a user 'sraschka' at 'instance-2' running a Miniconda3 installer script. The output includes a welcome message, a license agreement notice, and a prompt to press ENTER to continue, followed by a cursor and three greater-than signs '>>>'.



# Option 2: Google Cloud Instances

After completing the installation, source your `~/ .bashrc` file

A terminal window titled 'ssh.cloud.google.com' showing the installation of Miniconda3. The output lists the installation of various packages: setuptools, cryptography, wheel, pip, pyopenssl, urllib3, requests, and conda. After the installation is finished, it asks if the user wants to initialize Miniconda3 in their .bashrc file. The user responds 'yes'. The terminal then shows the initialization process, including a backup of the .bashrc file. It concludes with a message to open a new terminal for the changes to take effect and a thank you message. The prompt changes to 'sraschka@instance-2:~\$' and the user enters 'source ~/.bashrc'.

```
ssh.cloud.google.com
Installing: setuptools-40.6.3-py37_0 ...
Installing: cryptography-2.4.2-py37h1ba5d50_0 ...
Installing: wheel-0.32.3-py37_0 ...
Installing: pip-18.1-py37_0 ...
Installing: pyopenssl-18.0.0-py37_0 ...
Installing: urllib3-1.24.1-py37_0 ...
Installing: requests-2.21.0-py37_0 ...
Installing: conda-4.5.12-py37_0 ...
Installation finished.
Do you wish the installer to initialize Miniconda3
in your /home/sraschka/.bashrc ? [yes|no]
no] >>> yes

initializing Miniconda3 in /home/sraschka/.bashrc
A backup will be made to: /home/sraschka/.bashrc-miniconda3.bak

For this change to become active, you have to open a new terminal.

Thank you for installing Miniconda3!
sraschka@instance-2:~$ source ~/.bashrc
```

# Option 2: Google Cloud Instances

Next, you can conveniently install PyTorch via the command from PyTorch's main website <https://pytorch.org>

The image shows the PyTorch build configuration interface. On the left, a sidebar lists configuration categories: PyTorch Build, Your OS, Package, Language, CUDA, and Run this Command. The main area displays a grid of options with 'Stable (1.0)', 'Linux', 'Conda', 'Python 3.7', and '10.0' selected. Below the grid, the command to run is shown: `conda install pytorch torchvision cudatoolkit=10.0 -c pytorch`.

Below the website screenshot, a terminal window titled `ssh.cloud.google.com` shows the same command being executed in a shell: `sraschka@instance-2:~$ conda install pytorch torchvision cudatoolkit=10.0 -c pytorch`.

PyTorch Build	Stable (1.0)		Preview (Nightly)	
Your OS	Linux		Mac	Windows
Package	Conda		Pip	LibTorch
Language	Python 2.7	Python 3.5	Python 3.6	Python 3.7
CUDA	8.0	9.0	10.0	None

```
conda install pytorch torchvision cudatoolkit=10.0 -c pytorch
```

```
sraschka@instance-2:~$ conda install pytorch torchvision cudatoolkit=10.0 -c pytorch
```

# Option 2: Google Cloud Instances

Next, let's check that PyTorch works  
(you also may want to install ipython via conda):

```
Executing transaction: done
(base) sraschka@instance-2:~$ ipython
Python 3.7.1 (default, Dec 14 2018, 19:28:38)
Type 'copyright', 'credits' or 'license' for more information
Python 7.3.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: import torch

In [2]: █
```

# Option 2: Google Cloud Instances

How can we get data onto that instance now? This is a bit tricky, you would maybe need some understanding of Linux or macOS's Unix.

There are some tips here:

<https://cloud.google.com/compute/docs/instances/connecting-to-instance>

And here:

<https://cloud.google.com/compute/docs/instances/transfer-files>

# Option 2: Google Cloud Instances

There, you need to follow the instructions to create authentication files:

[SSH \(LINUX & MACOS\)](#) [PUTTY \(WINDOWS\)](#)

To connect to an instance using `ssh` :

1. [Provide your public SSH key to an instance](#) using one of the available options.
2. In the console, find the external IP address for the instance that you want to connect to. Go to the list of instances page.

GO TO THE INSTANCES PAGE

3. In a terminal, use the `ssh` command and your private SSH key file to connect to your instance. Specify the external IP address of the instance that you want to connect to.

```
ssh -i [PATH_TO_PRIVATE_KEY] [USERNAME]@[EXTERNAL_IP_ADDRESS]
```

where:

- `[PATH_TO_PRIVATE_KEY]` is [the path to your private SSH key file](#).
- `[USERNAME]` is the name of the user connecting to the instance. The username for your public key is the same as the username you used when the SSH key was created. You can connect to the instance as that user if the instance has that user and if you have the matching private SSH key.
- `[EXTERNAL_IP_ADDRESS]` is the external IP address for your instance.

If the connection is successful, you can use the terminal to run commands on your instance. When you are done, use the `exit` command to disconnect from the instance.

# Option 2: Google Cloud Instances

n	Internal IP	External IP	Connect
	10.166.0.2 (nic0)	35.228.10.5	SSH  
<div><div>Open in browser window</div><div>Open in browser window on custom port</div><div>View gcloud command</div><div>Use another SSH client</div></div>			

You can see that without key files, there's no access from your own terminal:

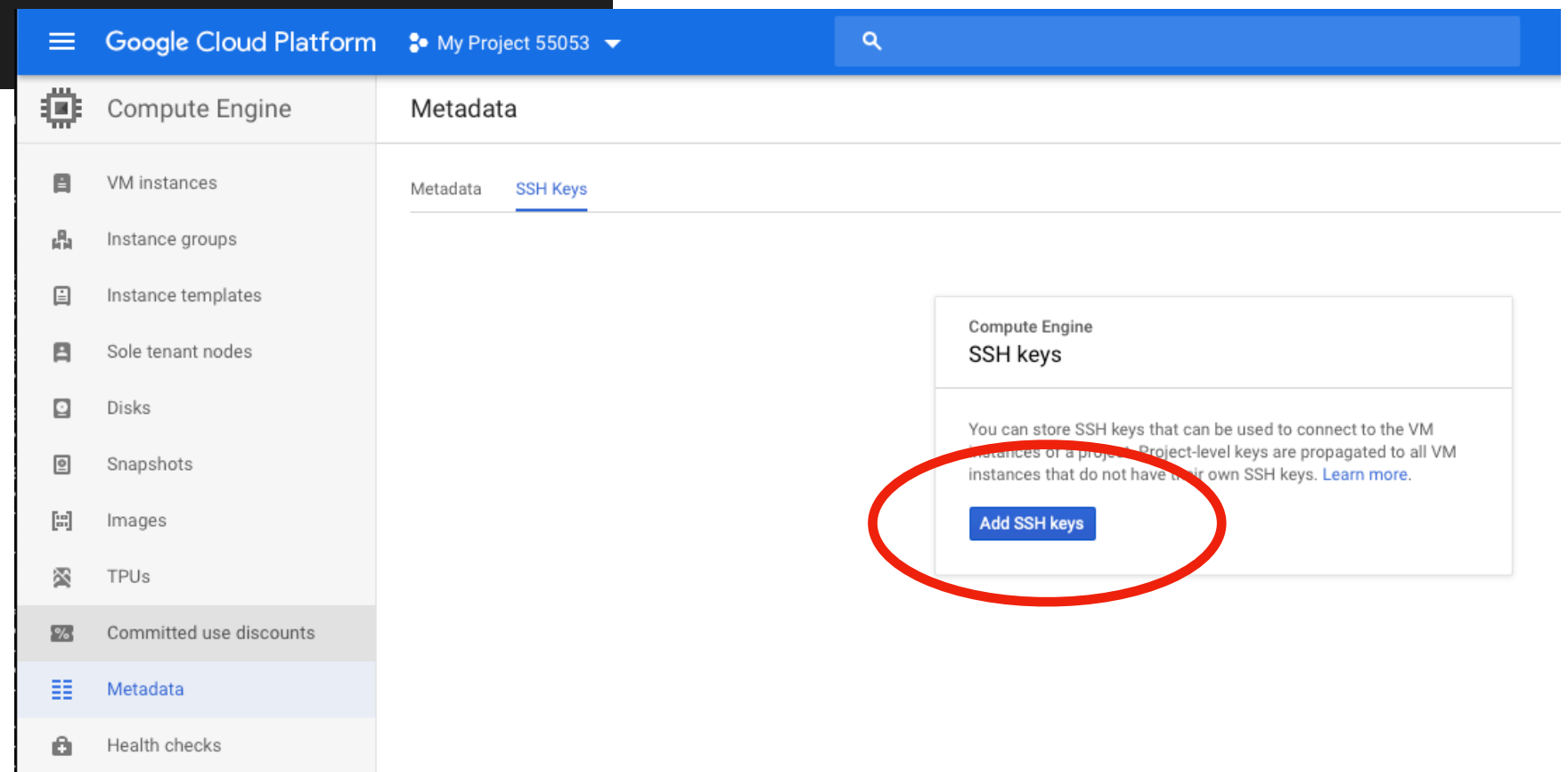
So, let's create a key pair:

```
sebastian — ssh-keygen -t rsa -f ~/.ssh/googlecloud -C sraschka — 80x24
Last login: Wed Feb 20 00:34:40 on ttys007
(base) sebastian@Sebastians-MacBook:~$ ssh sraschka@35.228.10.5
The authenticity of host '35.228.10.5 (35.228.10.5)' can't be established.
ECDSA key fingerprint is SHA256:E3SzAfk5pInnHdbnKkcAMNynaHvyzX5/UZN8OD4HExQ.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '35.228.10.5' (ECDSA) to the list of known hosts.
sraschka@35.228.10.5: Permission denied (publickey).
(base) sebastian@Sebastians-MacBook:~$ ssh-keygen -t rsa -f ~/.ssh/googlecloud -C sraschka
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
```

# Option 2: Google Cloud Instances

The public key (it's contents) is what needs to be entered online in your account:

```
.ssh — -bash — 80x24
(base) sebastian@Sebastians-MacBook:~/ssh$ ls
googlecloud  googlecloud.pub  known_hosts
(base) sebastian@Sebastians-MacBook:~/ssh$
```



# Option 2: Google Cloud Instances

Note that I blacked out my key for security reasons ;)

Be aware of Google's special key formatting requirements (i.e., the contents you see in the curly braces; you may have to add that manually)





# Option 2: Google Cloud Instances

Finally, we should be able to log in:

```
sebastian — ssh -i ~/.ssh/googlecloud sraschka@35.228.10.5 — 80x24
(base) sebastian@Sebastians-MacBook:~$ ssh -i ~/.ssh/googlecloud sraschka@35.228.10.5
Enter passphrase for key '/Users/sebastian/.ssh/googlecloud':
```

```
sebastian — sraschka@instance-2: ~ — ssh -i ~/.ssh/googlecloud sraschka@3...
[(base) sebastian@Sebastians-MacBook:~$ ssh -i ~/.ssh/googlecloud sraschka@35.228.10.5
[Enter passphrase for key '/Users/sebastian/.ssh/googlecloud':
Linux instance-2 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

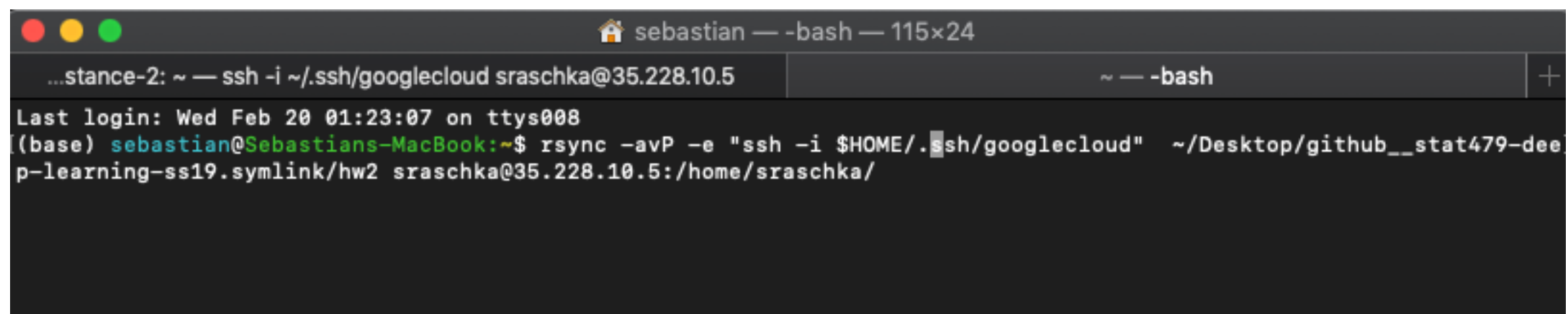
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Feb 20 06:59:47 2019 from 173.194.94.36
(base) sraschka@instance-2:~$
```

# Option 2: Google Cloud Instances

Next, you need to install rsync on the Google instance:

```
(base) sraschka@instance-2:~$ pwd
/home/sraschka
(base) sraschka@instance-2:~$ sudo apt-get install rsync
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  rsync
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 393 kB of archives.
After this operation, 703 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian stretch/main amd64 rsync amd64 3.1.2-1+deb9u1 [393 kB]
Fetched 393 kB in 0s (2,976 kB/s)
Selecting previously unselected package rsync.
(Reading database ... 34460 files and directories currently installed.)
Preparing to unpack .../rsync_3.1.2-1+deb9u1_amd64.deb ...
Unpacking rsync (3.1.2-1+deb9u1) ...
Setting up rsync (3.1.2-1+deb9u1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/rsync.service → /lib/systemd/system/rsync.service.
Processing triggers for systemd (232-25+deb9u8) ...
Processing triggers for man-db (2.7.6.1-2) ...
(base) sraschka@instance-2:~$
```

And after that, I can transfer files from my local machine to the Google instance:

A terminal window titled 'sebastian — -bash — 115x24' with a macOS-style title bar (red, yellow, green buttons). The terminal shows a session on a Google Cloud instance. The prompt is '(base) sebastian@Sebastians-MacBook:~\$'. The command entered is 'rsync -avP -e "ssh -i \$HOME/.ssh/googlecloud" ~/Desktop/github\_\_stat479-deep-learning-ss19.symlink/hw2 sraschka@35.228.10.5:/home/sraschka/'. The output shows the last login time and the command being executed. The terminal window has a tab titled '...stance-2: ~ — ssh -i ~/.ssh/googlecloud sraschka@35.228.10.5' and a '+' button to the right.

```
sebastian — -bash — 115x24
...stance-2: ~ — ssh -i ~/.ssh/googlecloud sraschka@35.228.10.5 ~ — -bash
Last login: Wed Feb 20 01:23:07 on ttys008
(base) sebastian@Sebastians-MacBook:~$ rsync -avP -e "ssh -i $HOME/.ssh/googlecloud" ~/Desktop/github__stat479-deep-learning-ss19.symlink/hw2 sraschka@35.228.10.5:/home/sraschka/
```

# Option 2: Google Cloud Instances

```
sebastian — -bash — 115x24
...stance-2: ~ — ssh -i ~/.ssh/googlecloud sraschka@35.228.10.5
~/rsync-52.200.1/rsync/io.c(453) [sender=2.6.9]
(base) sebastian@Sebastians-MacBook:~$ rsync -Pav -e "ssh -i $HOME/.ssh/googlecloud" ~/Desktop/github__stat479-dee
p-learning-ss19.symlink/hw2 sraschka@35.228.10.5:/home/sraschka/
[Enter passphrase for key '/Users/sebastian/.ssh/googlecloud':
building file list ...
9 files to consider
hw2/
hw2/.DS_Store
  6148 100%   0.00kB/s   0:00:00 (xfer#1, to-check=7/9)
hw2/hw2.ipynb
 18637 100% 17.77MB/s   0:00:00 (xfer#2, to-check=6/9)
hw2/.ipynb_checkpoints/
hw2/.ipynb_checkpoints/hw2-checkpoint.ipynb
 18637 100%   5.92MB/s   0:00:00 (xfer#3, to-check=4/9)
hw2/datasets/
hw2/datasets/iris.data
  4551 100% 493.82kB/s   0:00:00 (xfer#4, to-check=2/9)
hw2/images/
hw2/images/neuron.png
 47842 100%   4.15MB/s   0:00:00 (xfer#5, to-check=0/9)

sent 96376 bytes  received 154 bytes  9193.33 bytes/sec
total size is 95815  speedup is 0.99
(base) sebastian@Sebastians-MacBook:~$
```

As we can see, everything is on the Google instance now:

```
sebastian — sraschka@instance-2: ~/hw2 — ssh -i ~/.ssh/googlecloud sraschka@35.228.10.5 — 115x24
...ce-2: ~/hw2 — ssh -i ~/.ssh/googlecloud sraschka@35.228.10.5
(base) sraschka@instance-2:~$ ls
hw2  miniconda3  Miniconda3-latest-Linux-x86_64.sh  Miniconda3-latest-Linux-x86_64.sh.1
(base) sraschka@instance-2:~$ cd hw2/
(base) sraschka@instance-2:~/hw2$ ls
datasets  hw2.ipynb  images
(base) sraschka@instance-2:~/hw2$
```

# Option 2: Google Cloud Instances

Setting up Jupyter Notebook access is also tricky. I uploaded some instructions here, which also apply to Google Cloud:

Section H.10 (pp. 25-27)

[https://github.com/rasbt/stat479-deep-learning-ss19/blob/master/other/appendix\\_\\_cloud-computing.pdf](https://github.com/rasbt/stat479-deep-learning-ss19/blob/master/other/appendix__cloud-computing.pdf)

# Option 2: Google Cloud Instances

**Very Important: When you are done, stop or delete your instances!**

Compute Engine VM instances

Filter VM instances

Name	Zone	Recommendation	Internal IP	External IP	Connect
instance-2	europa-north1-a		10.166.0.2 (nic0)	35.228.10.5	SSH

- Start
- Stop
- Reset
- Delete
- View network details
- View logs

Once you are done, either stop or delete the instances. Stopped instances will cost some minor amount for storage, but you won't have to redo all the steps. Deleted instances are gone forever. I recommend stopping the instance until the end of the class if you like to reuse it.