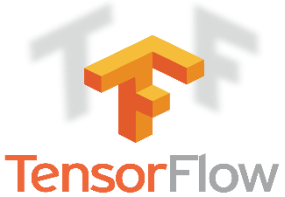
TensorFlow



- Google Colab is a cloud-based Juypter notebook style environment that comes with a free GPU or TPU (Brief introduction video here).
- A specific instance of Colab should allow you to run your code for 24 hrs without interruptions but each instance is limited to **24 hours**. Please note it can timeout.
- Colab has two versions of TensorFlow installed: a 1.x version (currently 1.15) and a 2.x version (currently 2.1).
- Colab currently uses TF 1.x by default. We will be using the newer TF2.1 in this module.

Short Tutorial on Using Colab

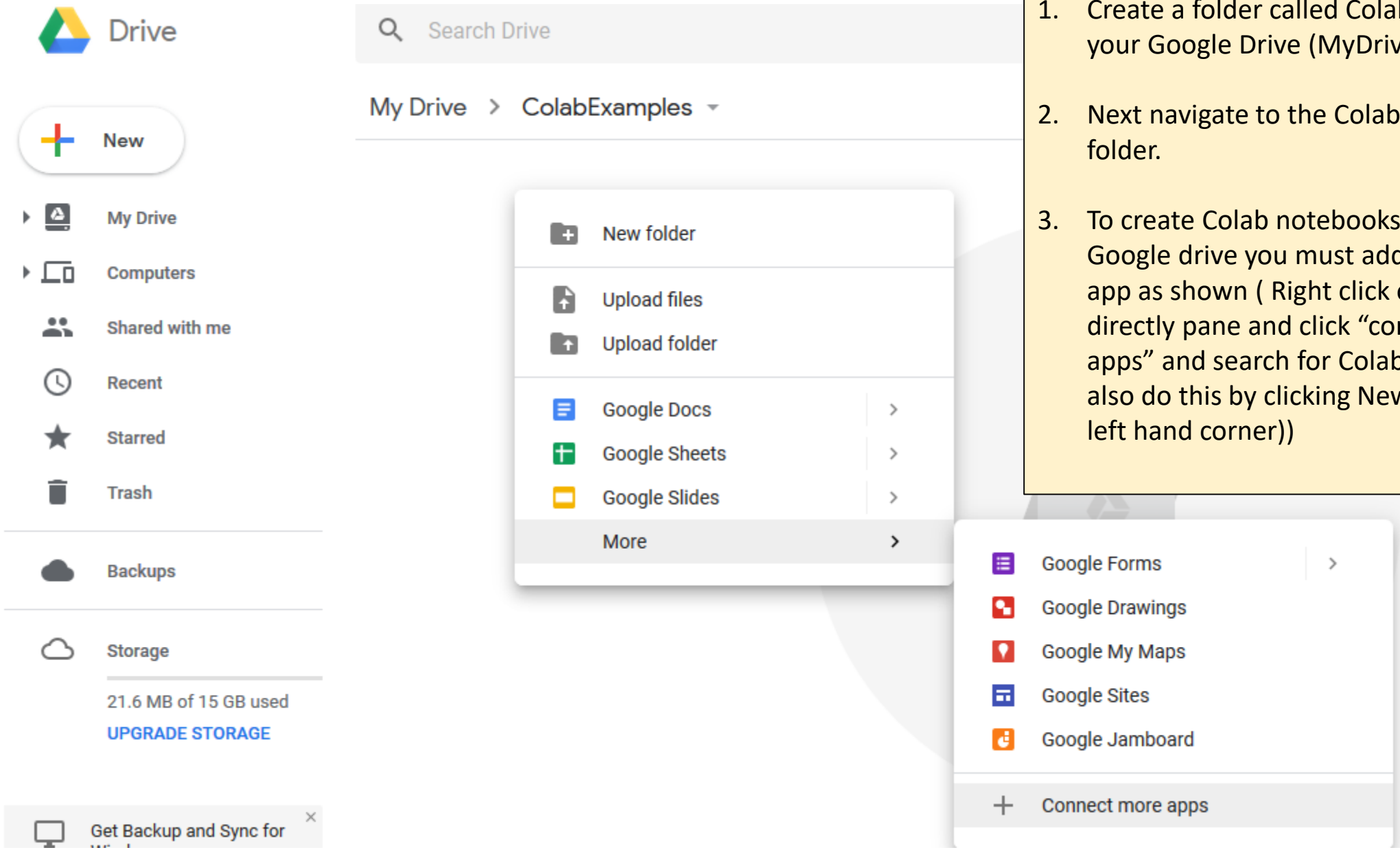
- Google Colab comes preinstalled with TensorFlow, Keras and a range of other packages such as NumPy.

- It is built on top of Juypter Notebooks and presents a similar interface consisting of cells.

- Over the next few slides we will create a Colab Notebook from your Google Drive, upload a dataset that resides in your drive and build a Keras Tensorflow model.

- First navigate to your Google Drive using your browser. You can directly create Colab Notebooks from there.

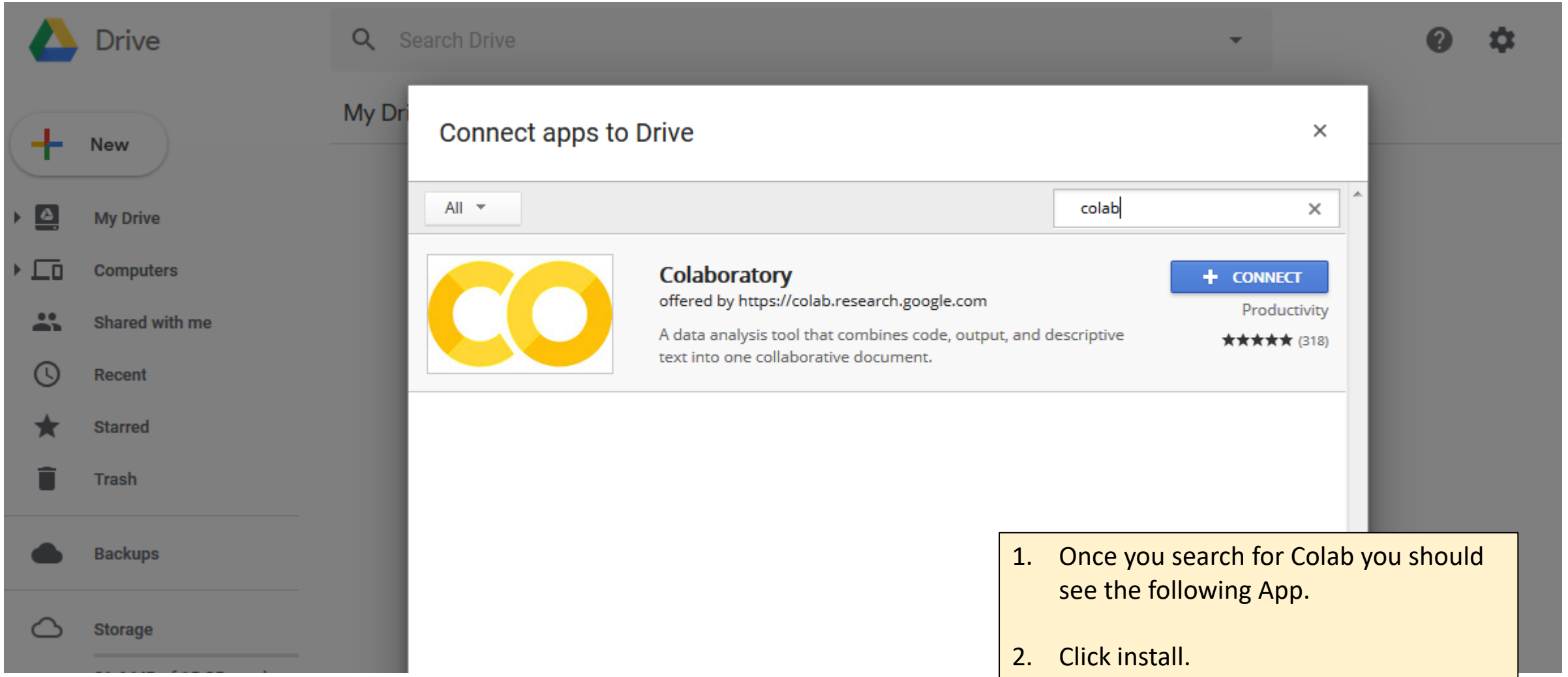
(Please note this is just one of many ways of creating Google Colab notebooks. I recommend that you use this as it allows you to easily organize all your files in a folder in your Google drive and also mount your Google Drive)



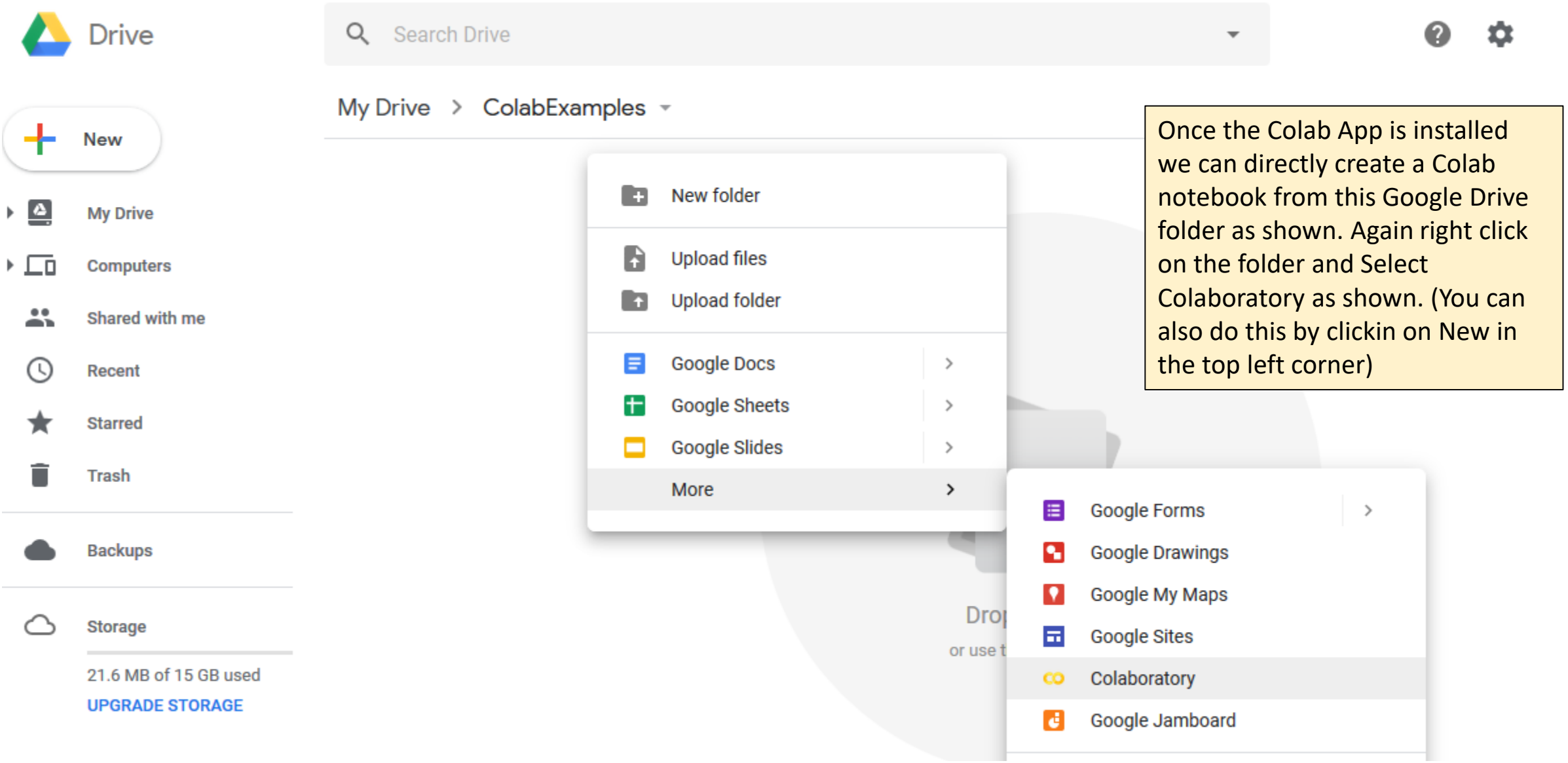
1. Create a folder called ColabExamples in your Google Drive (MyDrive).

2. Next navigate to the ColabExamples folder.

3. To create Colab notebooks from your Google drive you must add Colab as an app as shown (Right click on the directly pane and click “connect more apps” and search for Colab. (You can also do this by clicking New in the top left hand corner))



1. Once you search for Colab you should see the following App.
2. Click install.

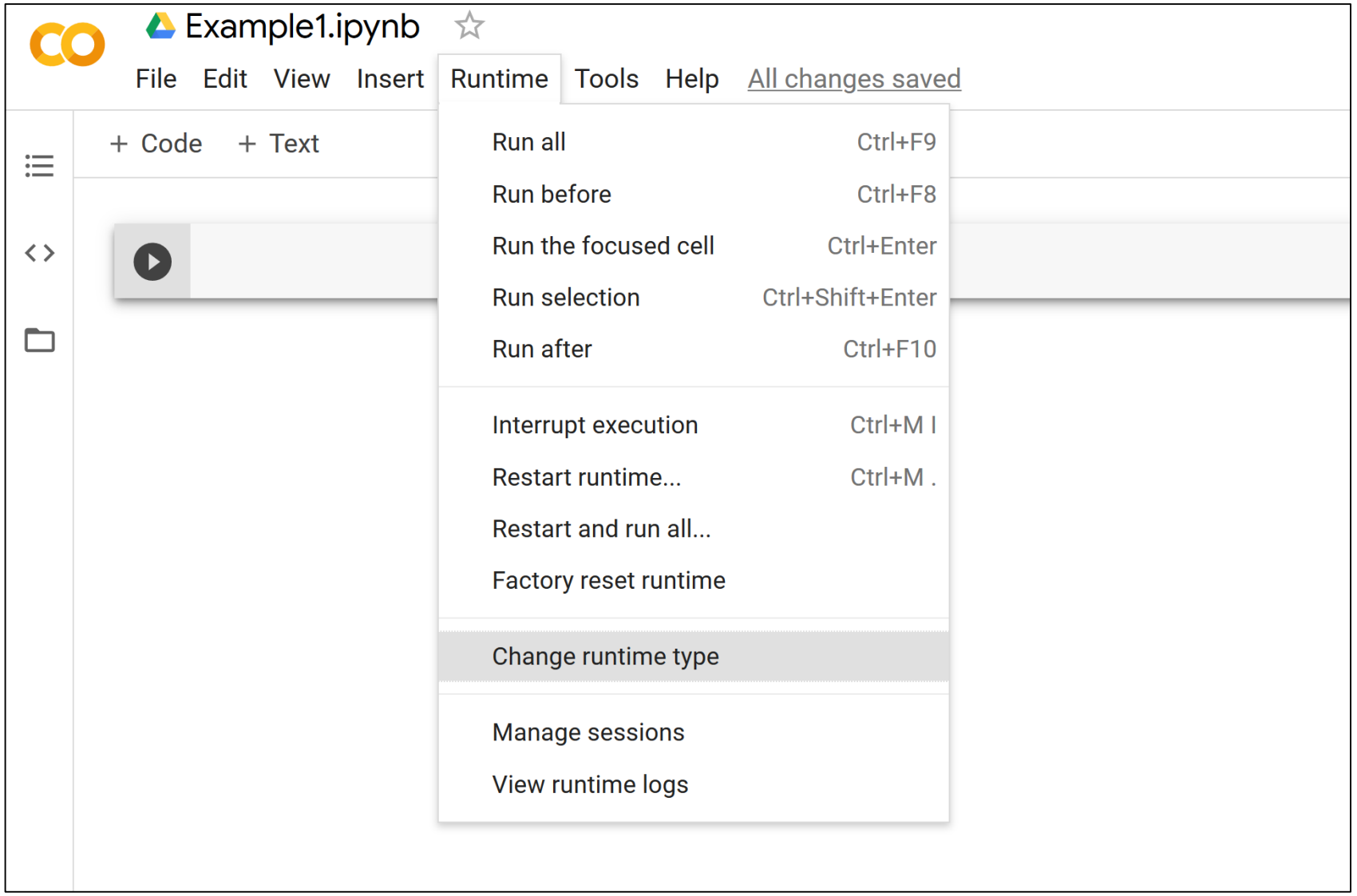

Drive
Search Drive
My Drive > ColabExamples
New
My Drive
Computers
Shared with me
Recent
Starred
Trash
Backups
Storage
21.6 MB of 15 GB used
UPGRADE STORAGE
New folder
Upload files
Upload folder
Google Docs
Google Sheets
Google Slides
More
Google Forms
Google Drawings
Google My Maps
Google Sites
Colaboratory
Google Jamboard
Once the Colab App is installed we can directly create a Colab notebook from this Google Drive folder as shown. Again right click on the folder and Select Colaboratory as shown. (You can also do this by clickin on New in the top left corner)

Please note you can find the full complete code (which we will be covering over the next few slides) here.

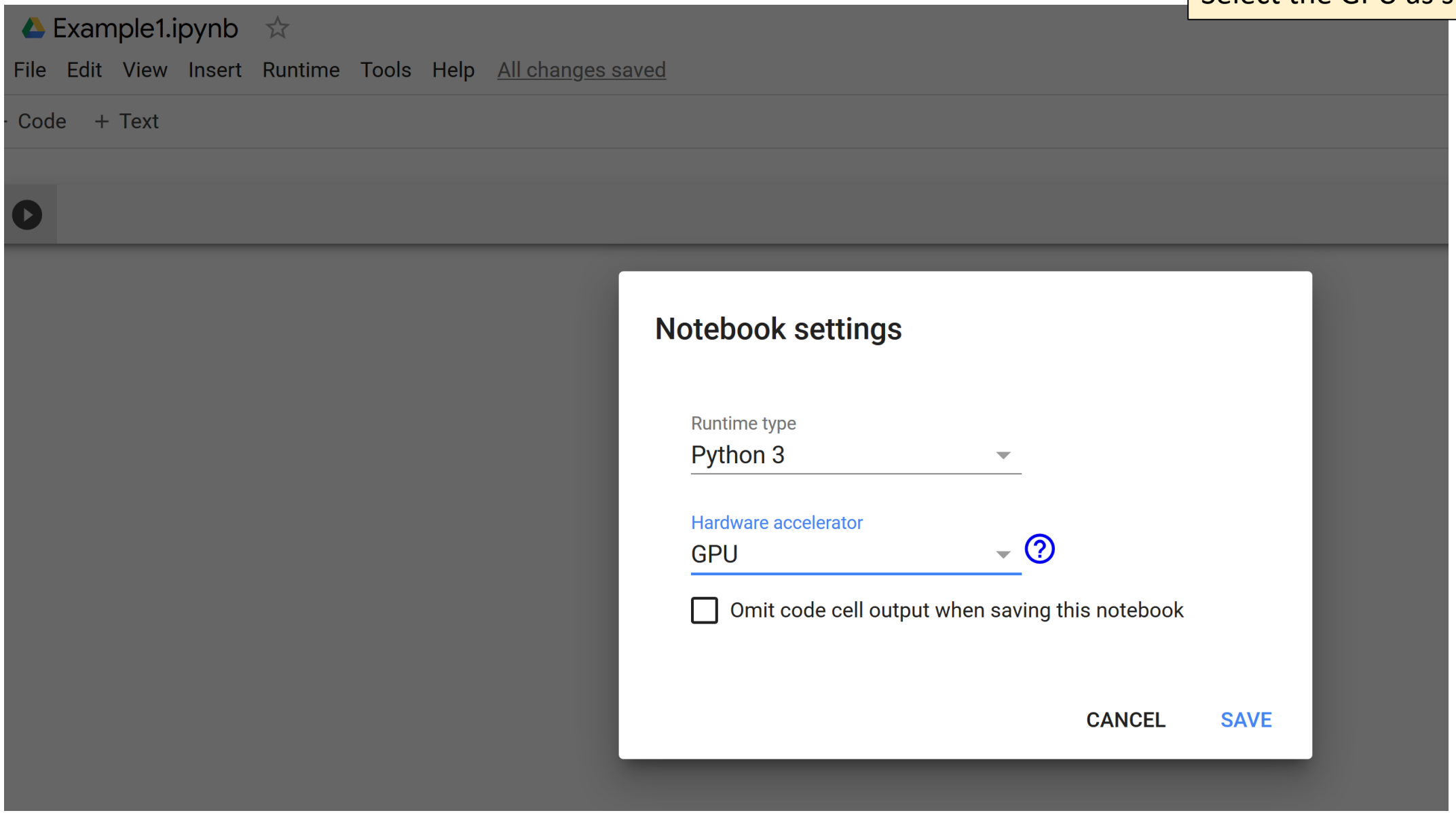
Once we create the Colab notebook it should open in Colab as shown below.

Next we are going to select a GPU runtime type.

Go to Runtime menu and Select Change runtime type (as shown).

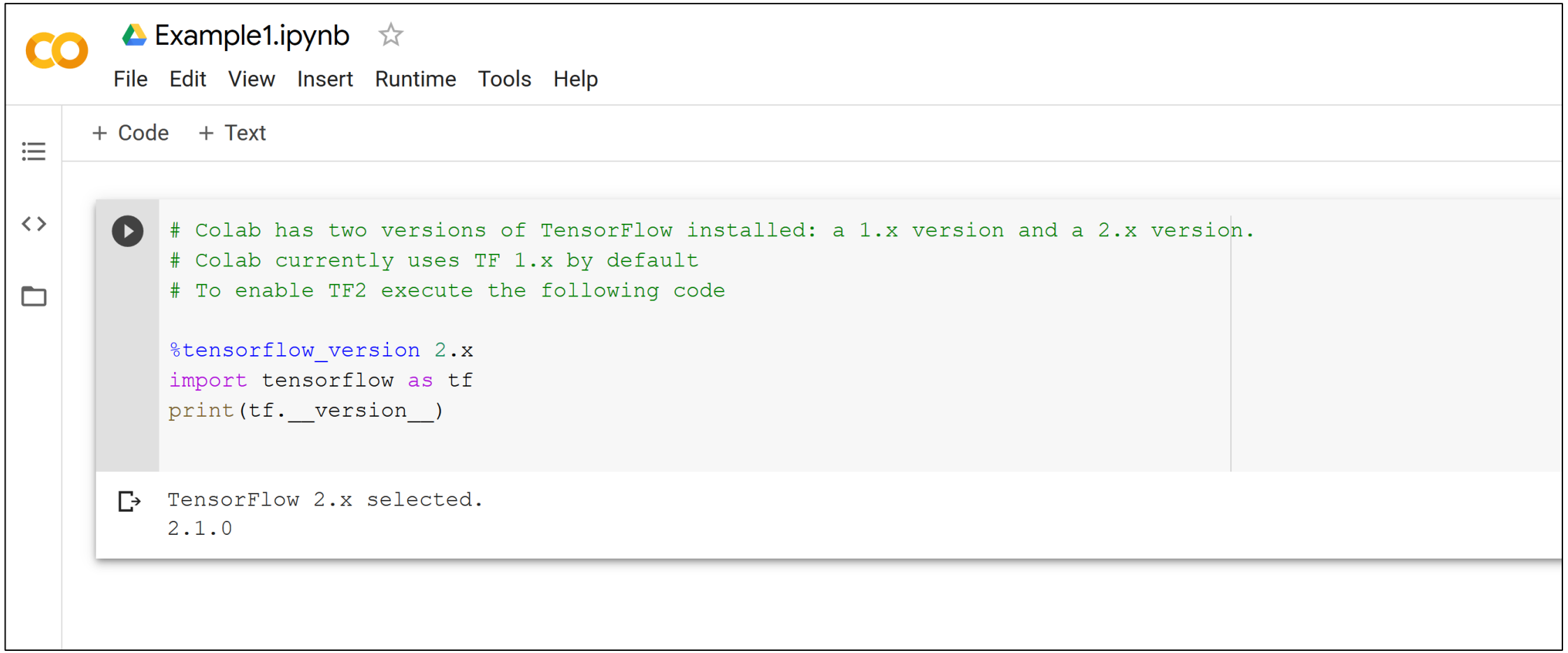


Select the GPU as shown.



Now it is important to enable TensorFlow 2.1 (at the moment TF 1.15 is the default version).

Type in the following into the first cell to enable TF2.1 (go here for full code.) When you run this cell you should see the following output.



TensorFlow using Colab

- Over the next few slides we will build a simple TensorFlow program as a classifier for MNIST.

- Rather than load the MNIST dataset automatically through TensorFlow we are going to save the MNIST data files to our Google Drive and show how to access this via Colab.

- 1. Go you the Colab unit in **Canvas**. There you will find a zip file called mnist.zip. Download this to your local machine. The zip file should contain a training and test file.

- 2. Next we want to **upload** the mnist folder to the ColabExamples folder on Google Drive (the same file containing your Colab files). You can do this as follows:

    - Unzip the mnist.zip file locally into a folder called mnist (so you should have one folder called mnist that contains the training and test data). Upload this mnist folder to your **ColabExamples** folder in Google Drive (one easy way of doing this is by dragging the folder to your ColabExamples directory).

    - * Alternatively if you are comfortable with using Google Drive then you can upload the mnist.zip file directly to your **ColabExamples** directory on your Google drive and unzip if using Zip Extractor.

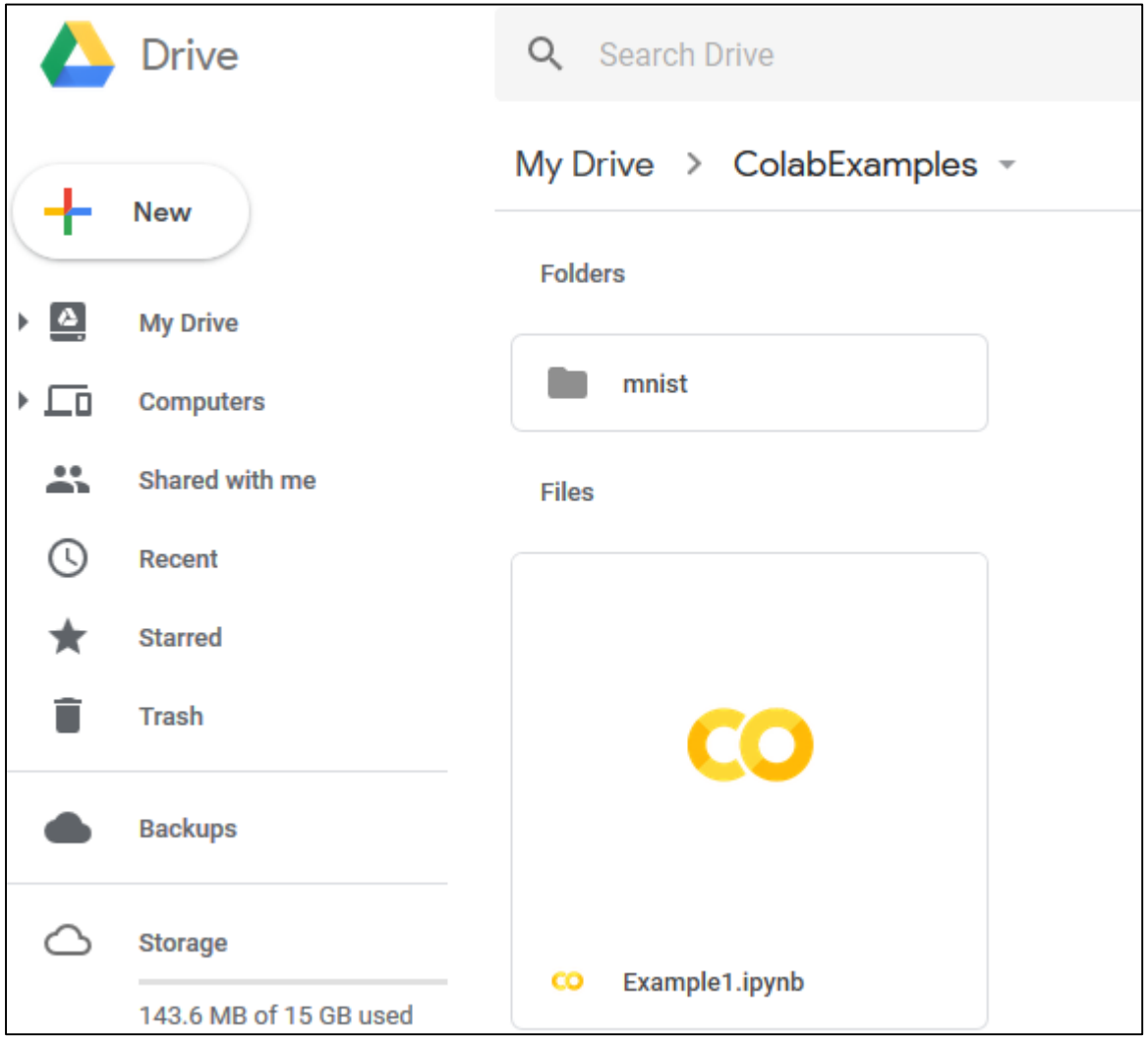
TensorFlow using Colab

- Before proceeding your should **make sure** the following two conditions are met:

1. Your ColabExamples directory (on Google Drive) will now contain your Example1 Colab notebook and your mnist folder (which contains the training and test data) [see image].

2. Your mnist folder contains a training and test csv file.

- The next step is to mount your Google drive so that you can access the mnist data from your Colab notebook.



Mounting Google Drive

- Enter the following code to mount your Google Drive.

```
from google.colab import drive

drive.mount('/content/gdrive')
```

- It will ask you to authenticate (click on the URL).
- Once you following the steps it will provide you with a code that you can enter, which will mount your Google drive.

```
[ ]  from google.colab import drive

     drive.mount('/content/gdrive')

     Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleuser

     Enter your authorization code:
     ··········
     Mounted at /content/gdrive
```

Mounting Google Drive and Reading Data

- Now that the drive is mounted we can read in our data file using the following code (notice NumPy is already installed).
- This may take some time depending on the dataset. The labels are the first column in the CSV file so we pull out the labels and features.
- We then normalize the feature data

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

     import numpy as np
     train = np.genfromtxt("/content/gdrive/My Drive/ColabExamples/mnist/mnist_train.csv", delimiter=",",skip_header=1)
     test = np.genfromtxt("/content/gdrive/My Drive/ColabExamples/mnist/mnist_test.csv", delimiter=",",skip_header=1)

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

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

```
[23]
     # Normalize the input data
     trainFeatures = trainFeatures/255
     testFeatures = testFeatures /255
```

Next we build our simple TF model and then subsequently train and evaluate the model.
Again you can find the full example code here.

```
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)
```

```
Train on 54000 samples, validate on 6000 samples
Epoch 1/5
54000/54000 [==============================] - 4s 81us/sample - loss: 0.2397 - accuracy: 0.9309 - val_loss: 0.1042 - val_accuracy: 0.9705
Epoch 2/5
54000/54000 [==============================] - 4s 77us/sample - loss: 0.0994 - accuracy: 0.9705 - val_loss: 0.0839 - val_accuracy: 0.9773
Epoch 3/5
54000/54000 [==============================] - 4s 76us/sample - loss: 0.0665 - accuracy: 0.9799 - val_loss: 0.0732 - val_accuracy: 0.9783
Epoch 4/5
54000/54000 [==============================] - 4s 78us/sample - loss: 0.0467 - accuracy: 0.9851 - val_loss: 0.0729 - val_accuracy: 0.9797
Epoch 5/5
54000/54000 [==============================] - 4s 77us/sample - loss: 0.0343 - accuracy: 0.9890 - val_loss: 0.0665 - val_accuracy: 0.9810
10000/10000 [==============================] - 1s 57us/sample - loss: 0.0698 - accuracy: 0.9795
```

Uploading Directly to Colab

- Please note there are multiple different methods for accessing data files on Colab.

- Another alternative is that you can directly upload the data to your Colab VM instance (This can take time depending on your connection speed and also the size of the data file).

- This will improve the performance but obviously you will have to reload this data each time you have to start the Colab instance. If it is a large data file this could take some time.

- You can find a version of the mnist example here where we directly upload the mnist.zip file to Colab.