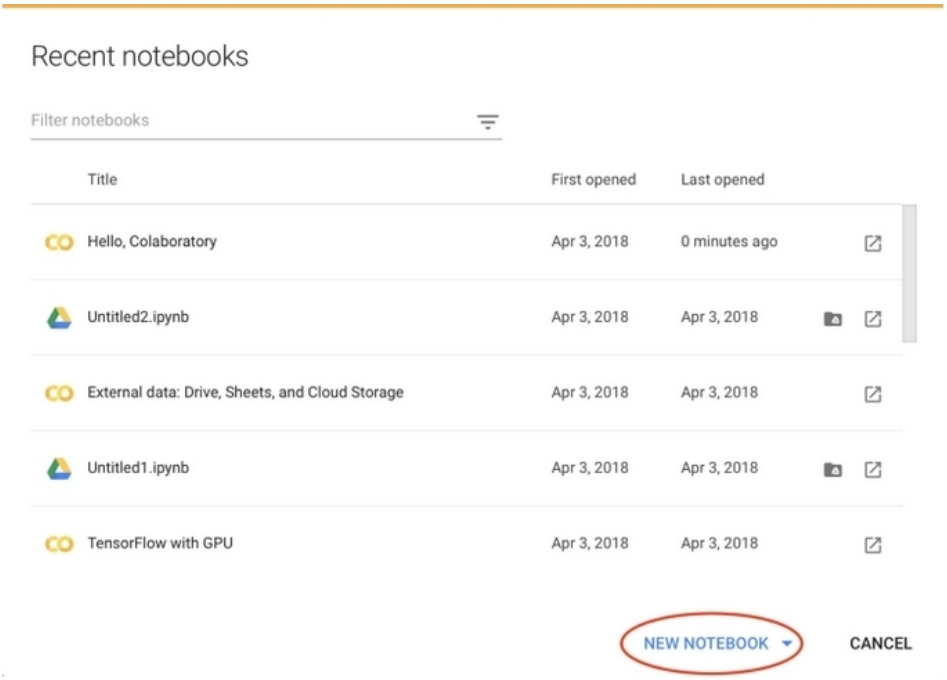
Running CUDA C/C++ in Jupyter or how to run `nvcc` in Google CoLab

1.1 Installation of compiler and runtime for CUDA

CUDA is NVIDIA's parallel computing architecture that enables dramatic increases in computing performance by harnessing the power of the GPU. With **Colab**, you can work with **CUDA C/C++** on the GPU for free. Imagine **11.5 GB Nvidia K80 GPU for free.**

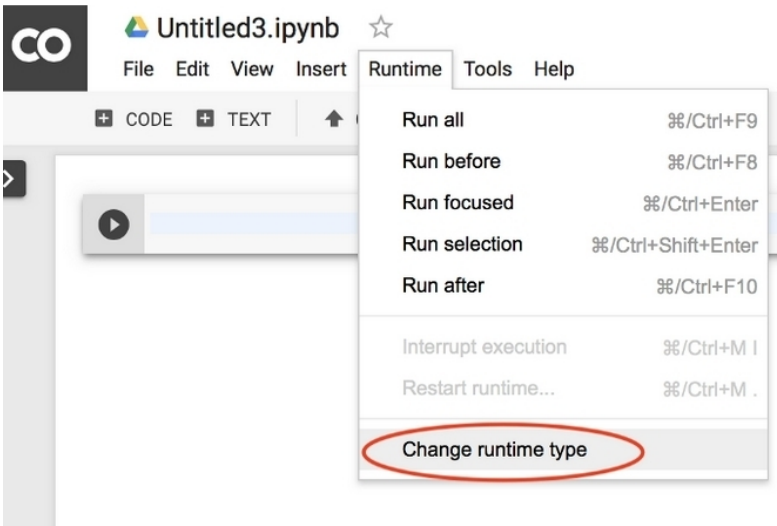
Not that long ago Nvidia announced its Deep Learning Institute where you can acquire basics of CUDA programming in both **Python** and **C/C++.**

First, create a new Notebook.



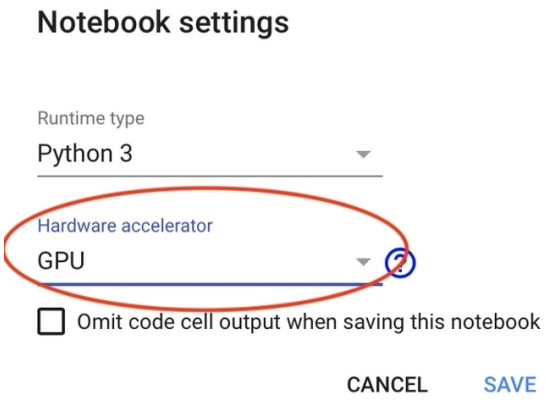
Please select **Python 3 Notebook** in the pop-up window.

If you have worked with Jupyter before, the interface will look familiar. A bit more stylish though. If you haven't, don't worry. It's a pretty simple and very powerful tool, that's way it is so popular.



Next, we need to switch our **runtime** from CPU to GPU. We just 2 clicks away.

Change **runtime** type in Notebook settings under Runtime tab on the upper menu:



And click **save**.

Despite that CUDA libs are available for the Tensorflow environment, **Colab** does not have NVCC (Nvidia CUDA Compiler) installed.

First uninstall any previous versions of CUDA completely.
Note that the **'!'** added at the beginning of a line allows it to be executed as a command line command.

```
!apt-get --purge remove cuda nvidia* libnvidia-*
!dpkg -l | grep cuda- | awk '{print $2}' | xargs -n1 dpkg --purge
!apt-get remove cuda-*
!apt autoremove
!apt-get update
```

Then install CUDA Version 9.

```
!wget https://developer.nvidia.com/compute/cuda/9.2/Prod/local_installers/cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64 -O cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64.deb
!dpkg -i cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64.deb
!apt-key add /var/cuda-repo-9-2-local/7fa2af80.pub
!apt-get update
!apt-get install cuda-9.2
```

You can just copy it in a cell in Notebook. Each line that starts with `!` is going to be executed as a command line command. Now you can test your CUDA installation by running

```
!nvcc --version
```

And the output should be something like

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2018 NVIDIA Corporation
Built on Wed_Apr_11_23:16:29_CDT_2018
Cuda compilation tools, release 9.2, V9.2.88
```

We are almost done. I created a small extension for running NVCC from Notebook cells. Install it with

```
!pip install git+git://github.com/andreinechaev/nvcc4jupyter.git
```

Now you need to load the installed extension, by running:

```
%load_ext nvcc_plugin
```

1.2 Compiling and running your first CUDA program on remote host

We are ready to run CUDA C/C++ code right in your Notebook.
For this we need explicitly say to the interpreter, that we want to use the extension by adding %cu at the beginning of each cell with CUDA code.

1.2.1 First test example

```
%%cu
#include <iostream>
int main() {
    std::cout << "Hello world\n";
    return 0;
}
```

1.3 Now you can edit and try your other programs !

1.3.1 `deviceStat.cu`

1.3.2 `addVector.cu`

1.3.3 `addVector4x4.cu`

1.3.4 `addVectorlong.cu`