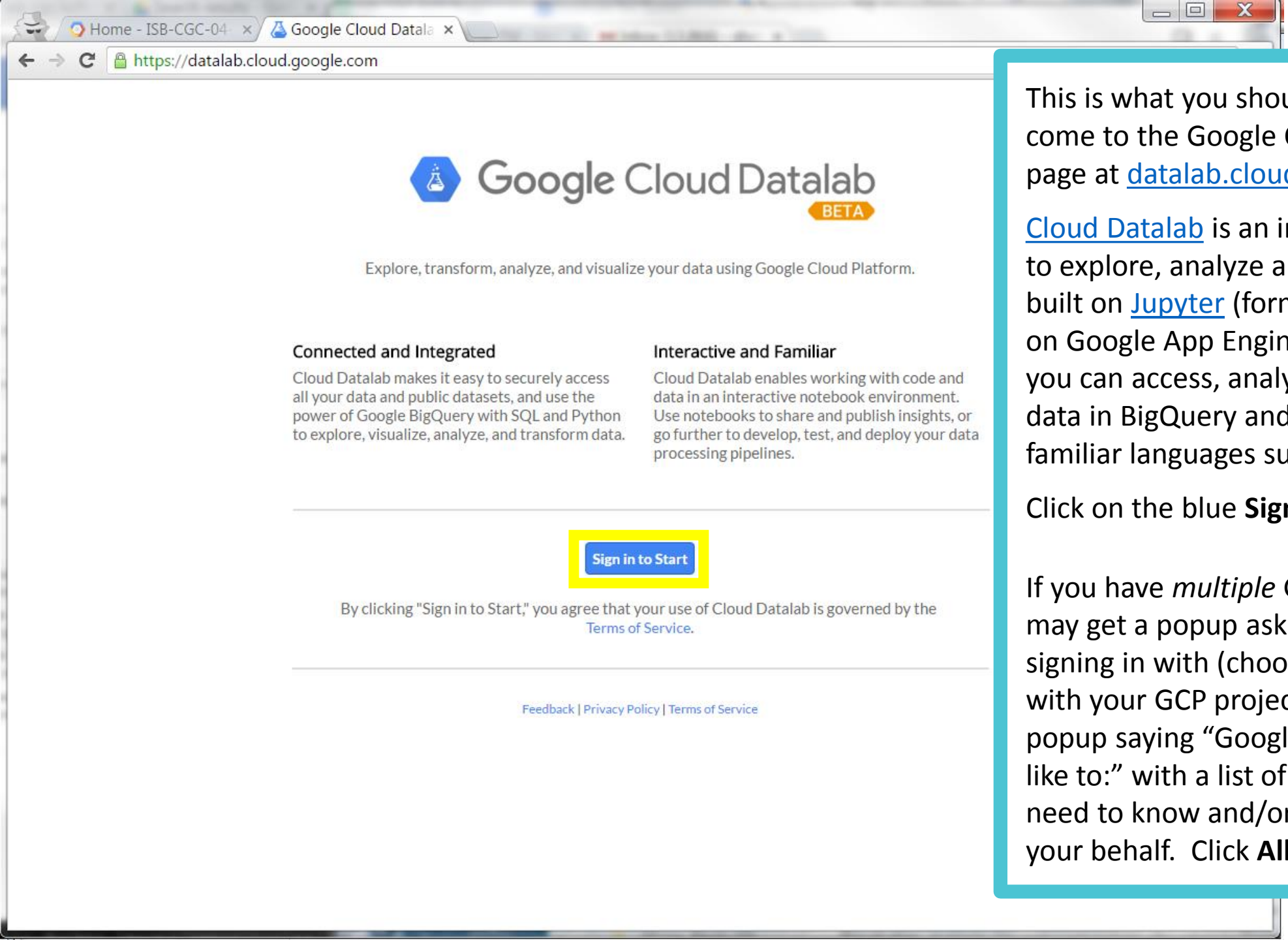
An Introduction to Cloud Datalab

(IPython/Jupyter)

(in less than 20 minutes)

brought to you by

The ISB Cancer Genomics Cloud

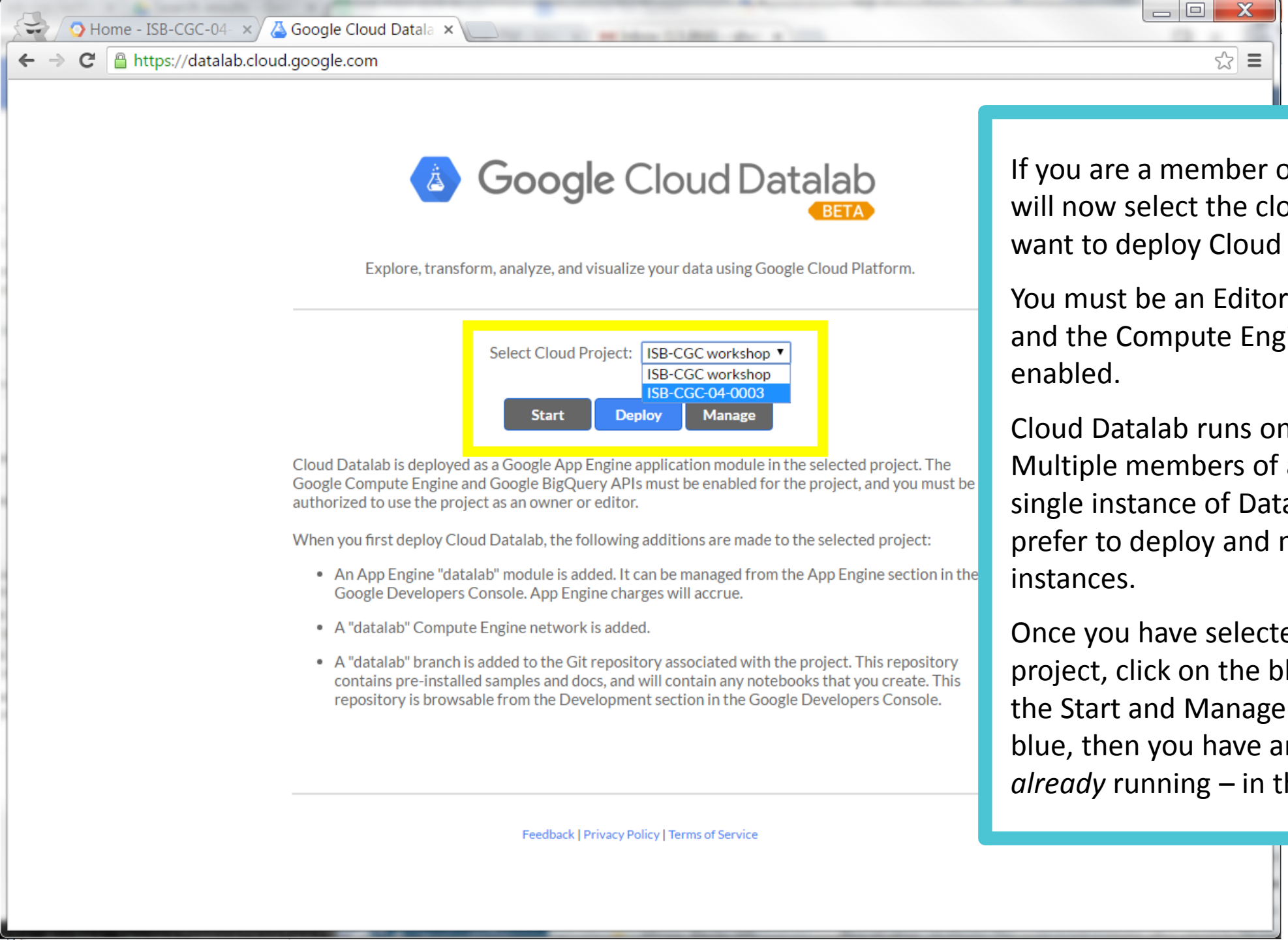


This is what you should see the first time you come to the Google Cloud Datalab landing page at datalab.cloud.google.com

Cloud Datalab is an interactive tool created to explore, analyze and visualize data. It is built on Jupyter (formerly IPython) and runs on Google App Engine. Using Cloud Datalab, you can access, analyze, and manipulate data in BigQuery and Cloud Storage using familiar languages such as Python and SQL.

Click on the blue **Sign in to Start** button.

If you have *multiple* Google identities, you may get a popup asking which one you are signing in with (choose the one associated with your GCP project). You may then see a popup saying "Google Cloud Datalab would like to:" with a list of things Datalab will need to know and/or be allowed to do on your behalf. Click **Allow**.

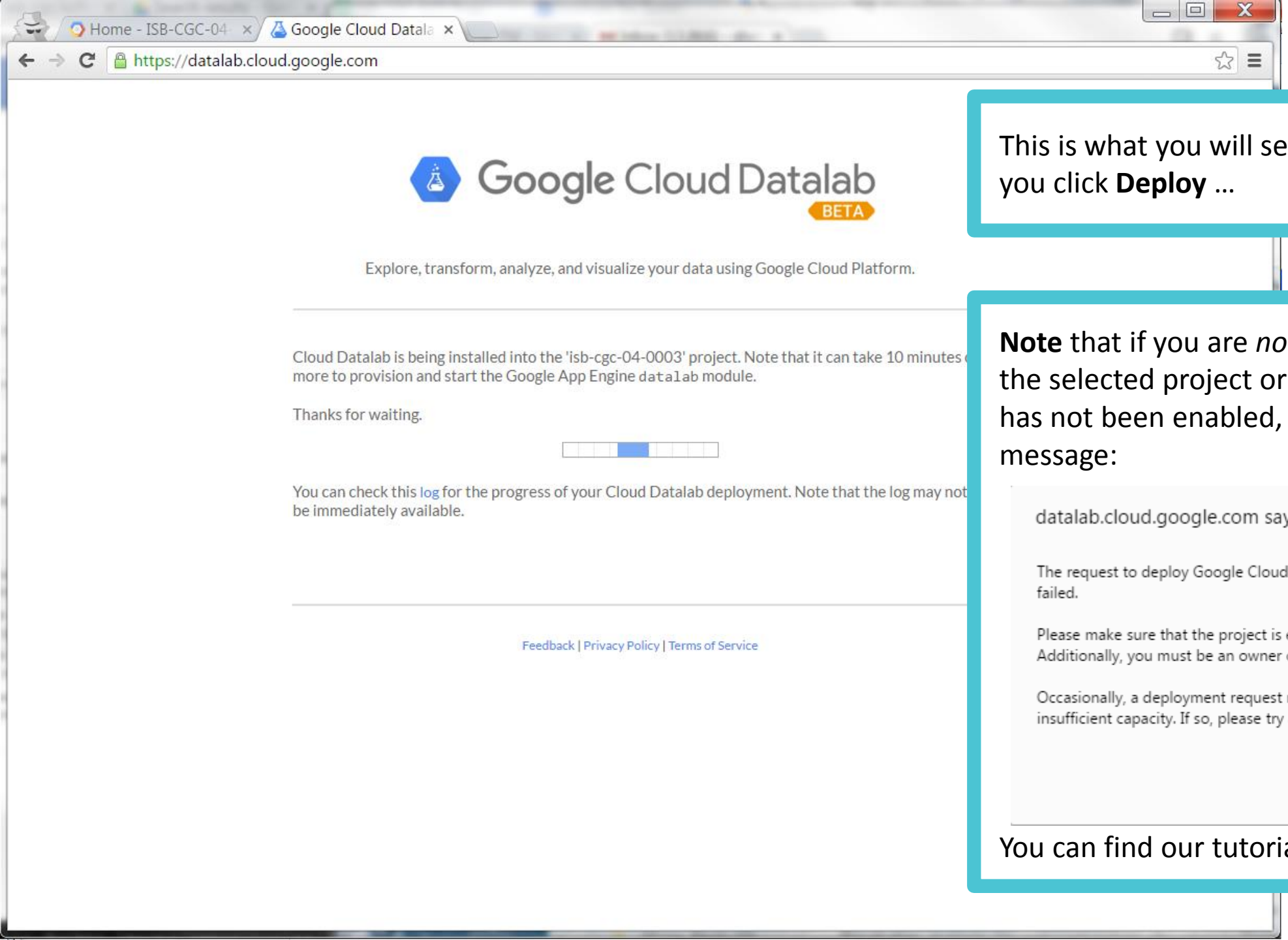


If you are a member of multiple projects you will now select the cloud project in which you want to deploy Cloud Datalab.

You must be an Editor or Owner of the project and the Compute Engine API must already be enabled.

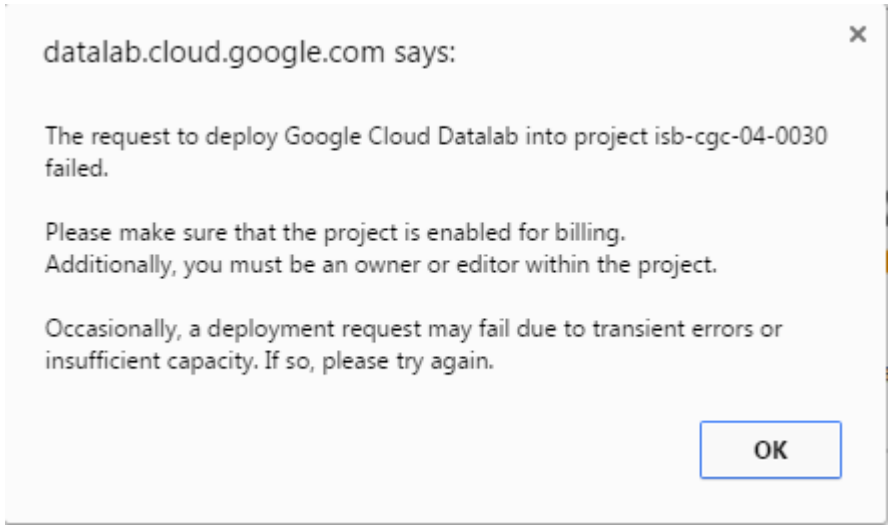
Cloud Datalab runs on a VM in your project. Multiple members of a project may access a single instance of Datalab, or individuals may prefer to deploy and manage personal instances.

Once you have selected the correct cloud project, click on the blue **Deploy** button. (If the Start and Manage buttons are already blue, then you have an instance of Datalab *already* running – in that case click **Start**.)

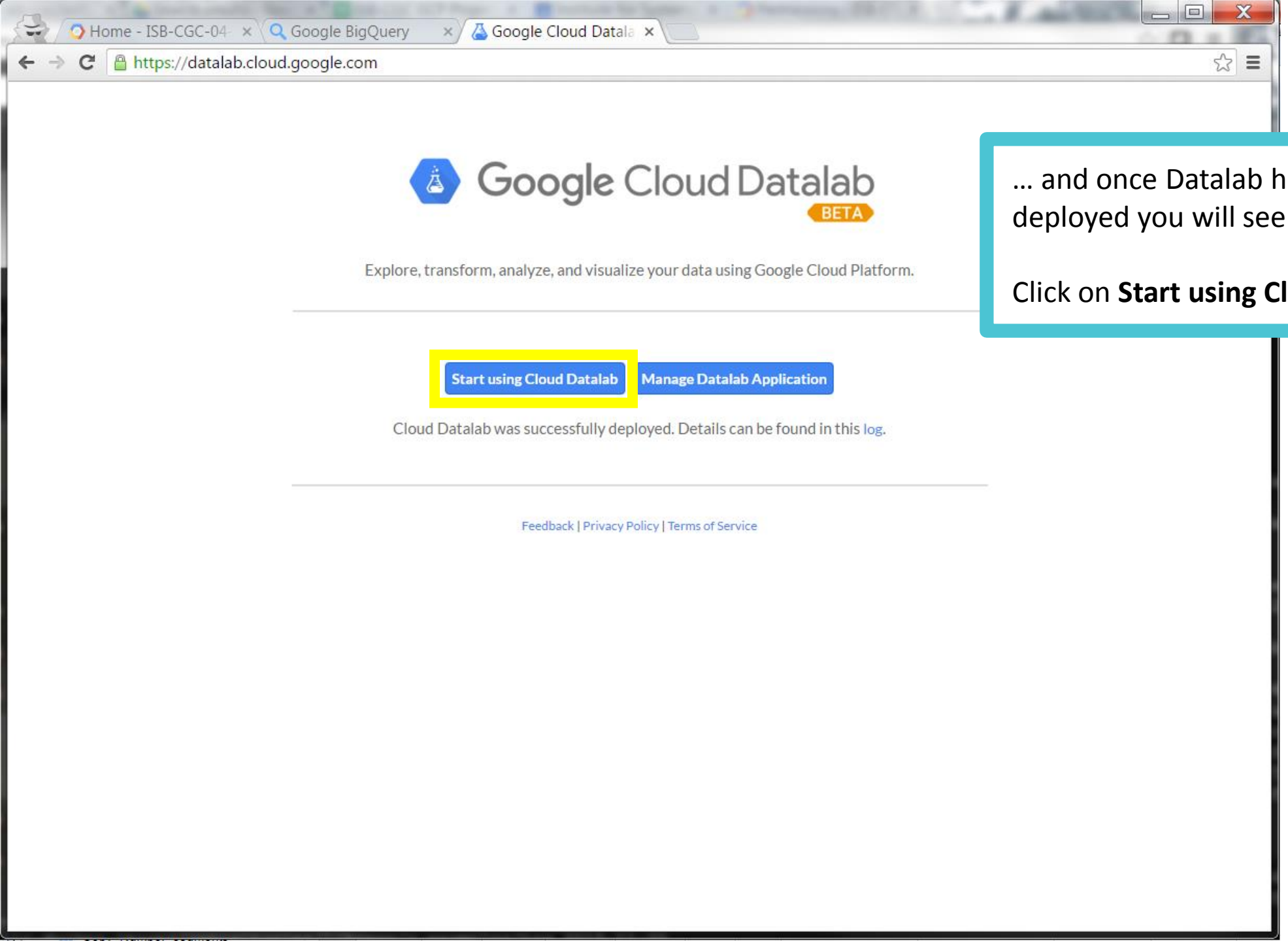


This is what you will see for 5-10 minutes after you click **Deploy** …

Note that if you are *not* an editor or owner on the selected project or the Compute Engine API has not been enabled, you will get this error message:

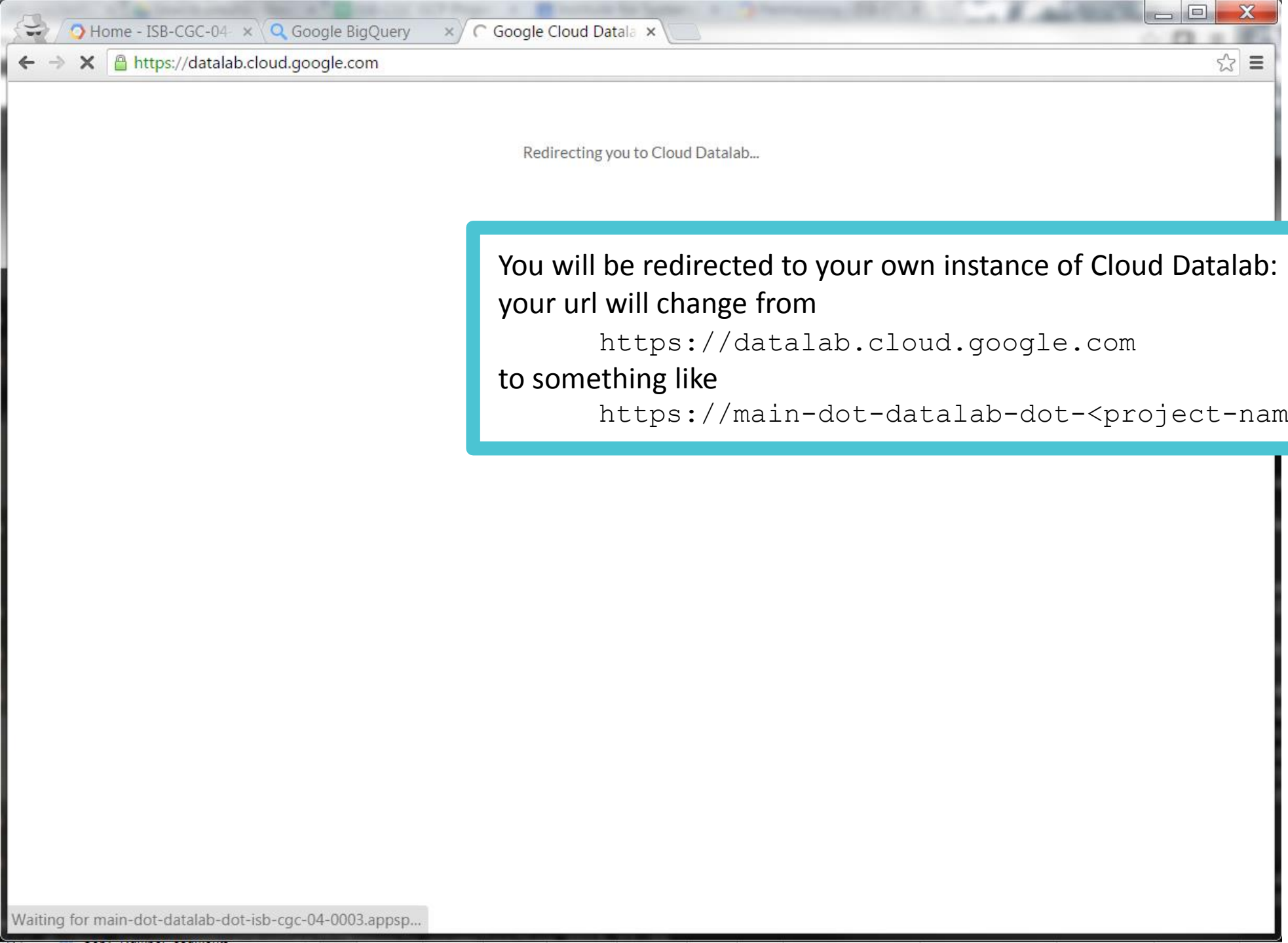


You can find our tutorial on enabling APIs here.



… and once Datalab has been successfully deployed you will see these two new options.

Click on **Start using Cloud Datalab**.

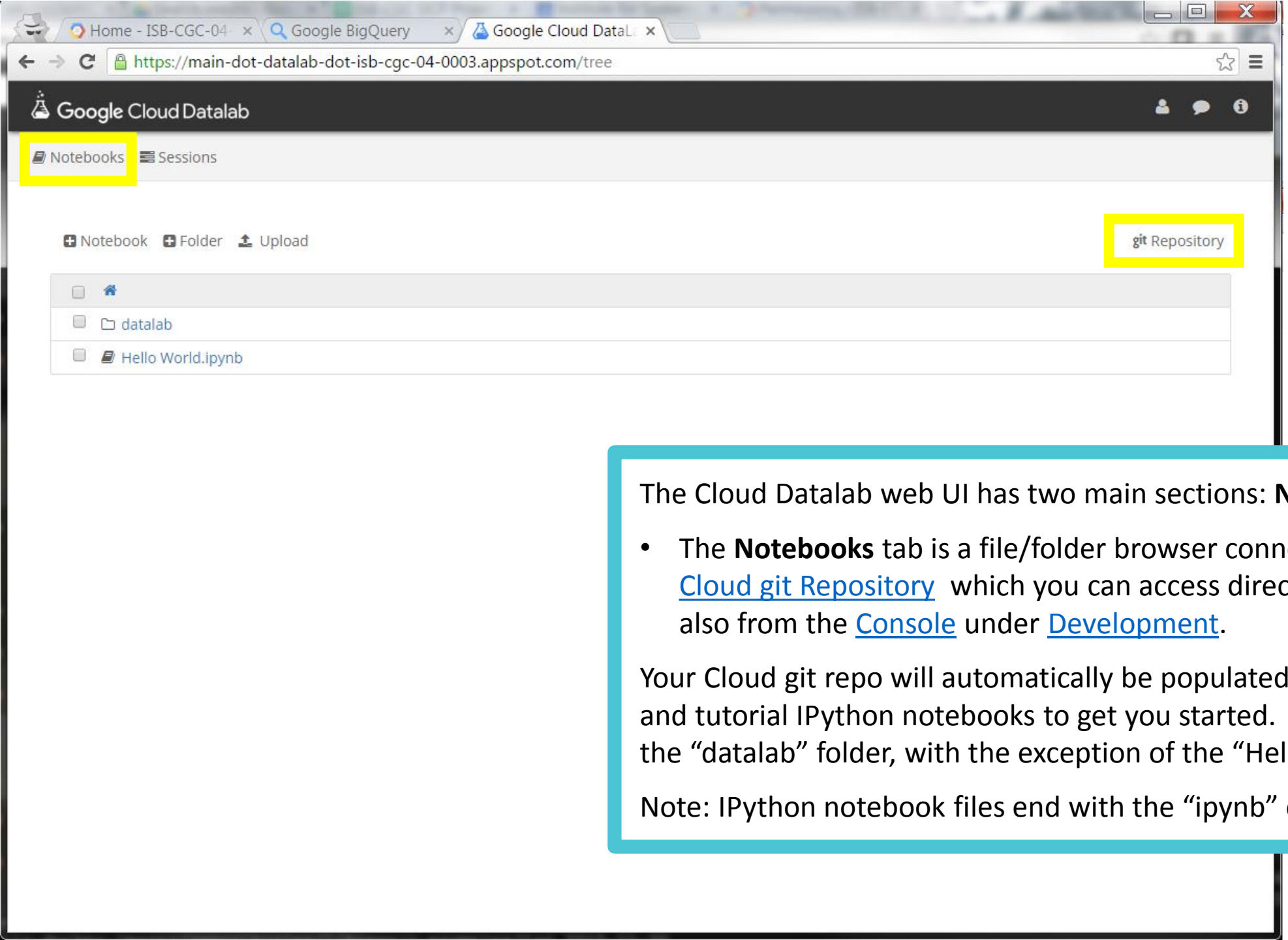


You will be redirected to your own instance of Cloud Datalab:
your url will change from

`https://datalab.cloud.google.com`

to something like

`https://main-dot-datalab-dot-<project-name>.appspot.com/tree`

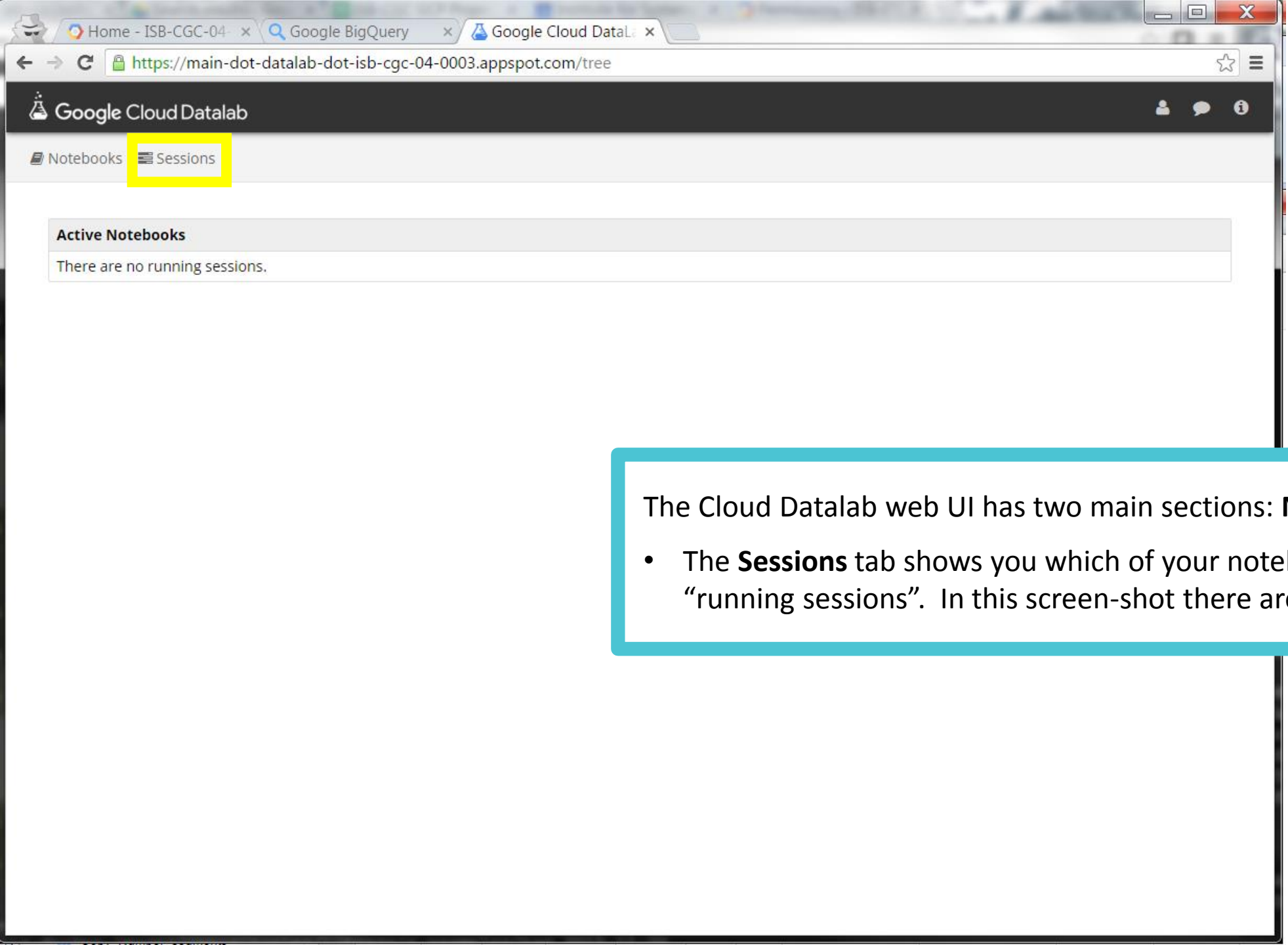


The Cloud Datalab web UI has two main sections: **Notebooks** and **Sessions**.

- The **Notebooks** tab is a file/folder browser connected to your Google Cloud git Repository which you can access directly from this page, and also from the Console under Development.

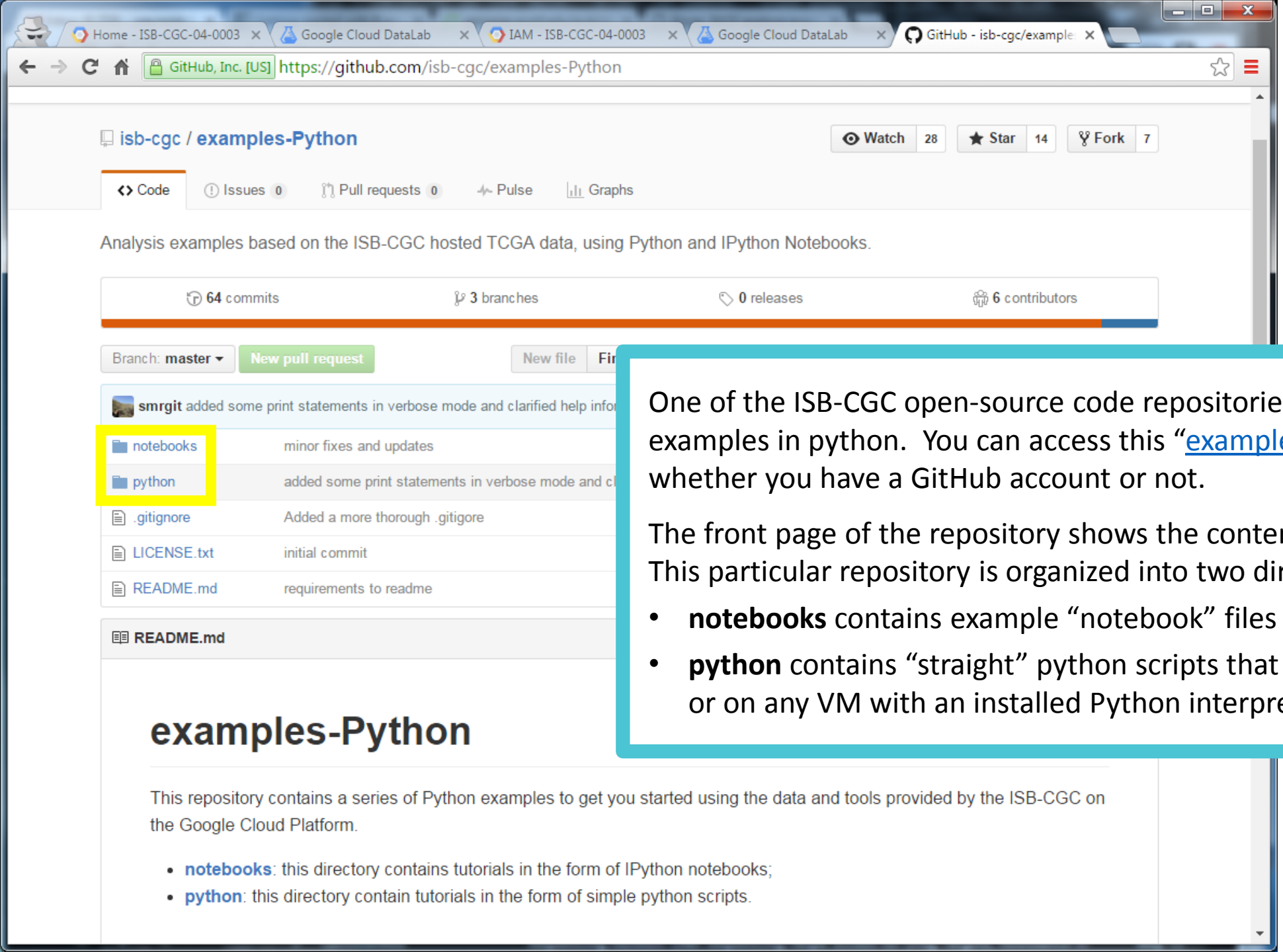
Your Cloud git repo will automatically be populated with a set of example and tutorial IPython notebooks to get you started. These notebooks are in the "datalab" folder, with the exception of the "Hello World" notebook.

Note: IPython notebook files end with the "ipynb" extension.



The Cloud Datalab web UI has two main sections: **Notebooks** and **Sessions**.

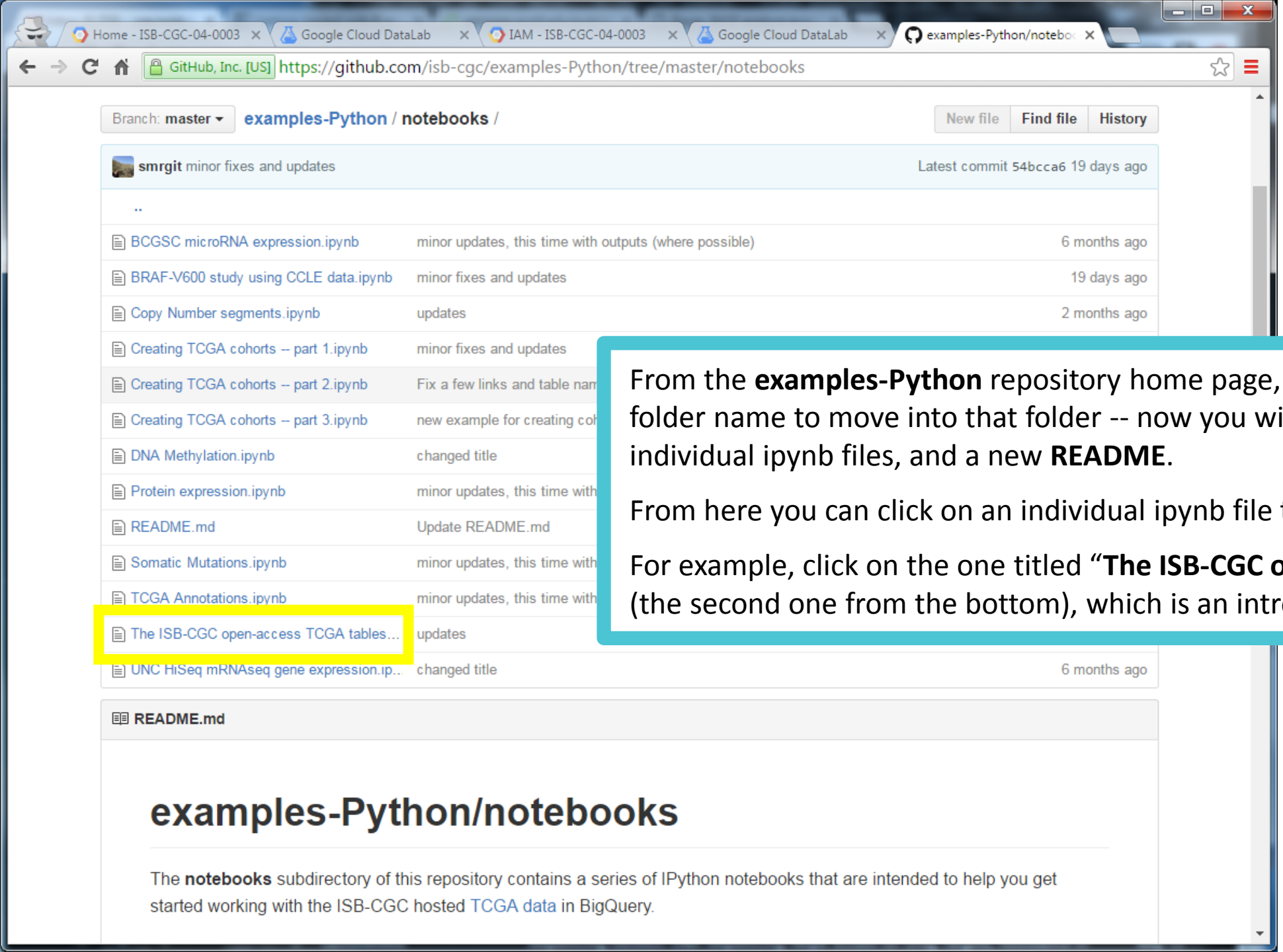
- The **Sessions** tab shows you which of your notebooks are “active” in “running sessions”. In this screen-shot there are no running sessions.



One of the ISB-CGC open-source code repositories on GitHub contains examples in python. You can access this "examples-Python" repository whether you have a GitHub account or not.

The front page of the repository shows the contents and the README. This particular repository is organized into two directories:

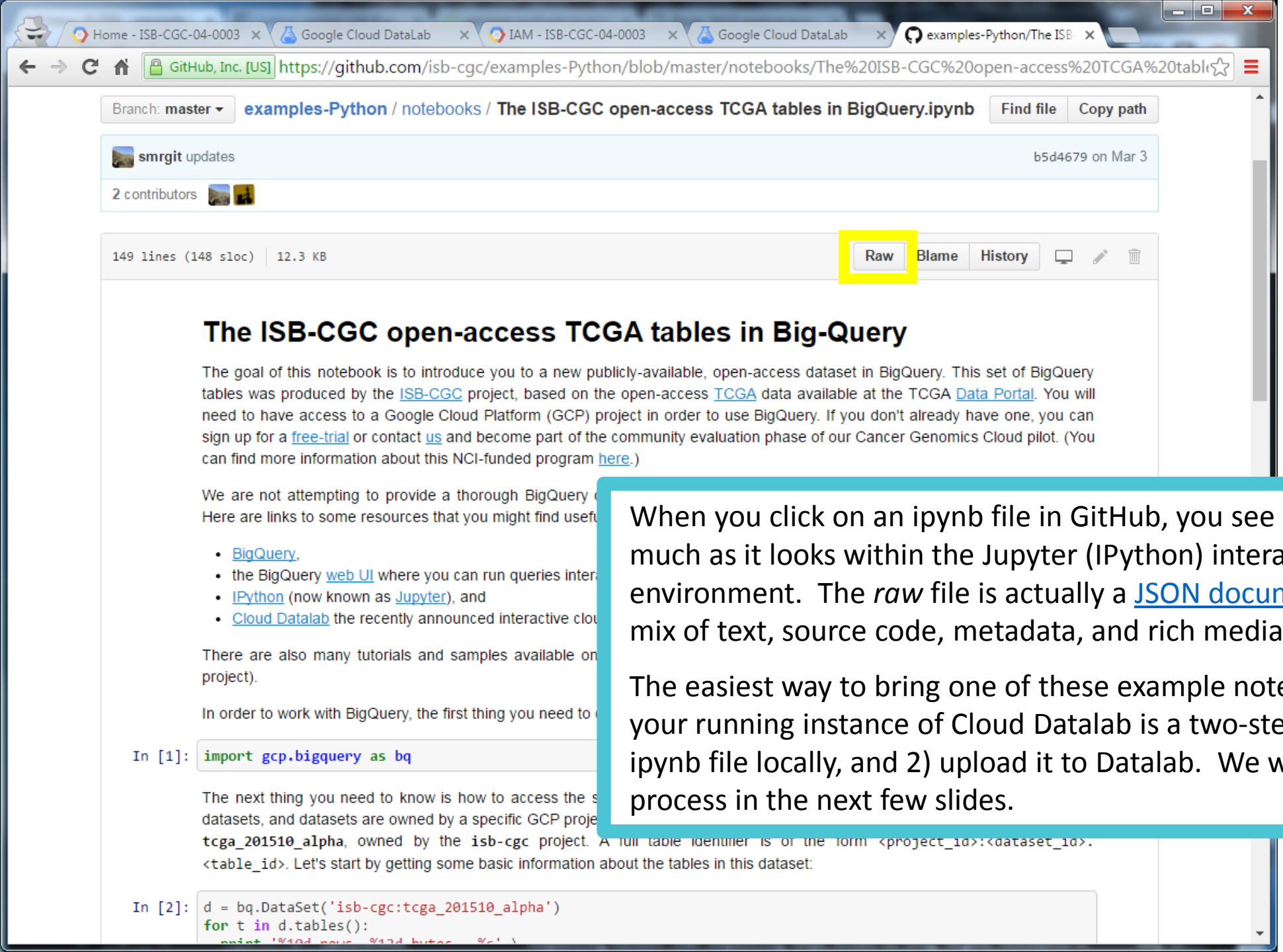
- **notebooks** contains example "notebook" files for use in Cloud Datalab;
- **python** contains "straight" python scripts that can be run in Cloud Shell or on any VM with an installed Python interpreter.



From the **examples-Python** repository home page, click on the **notebooks** folder name to move into that folder -- now you will see a list of the individual ipynb files, and a new **README**.

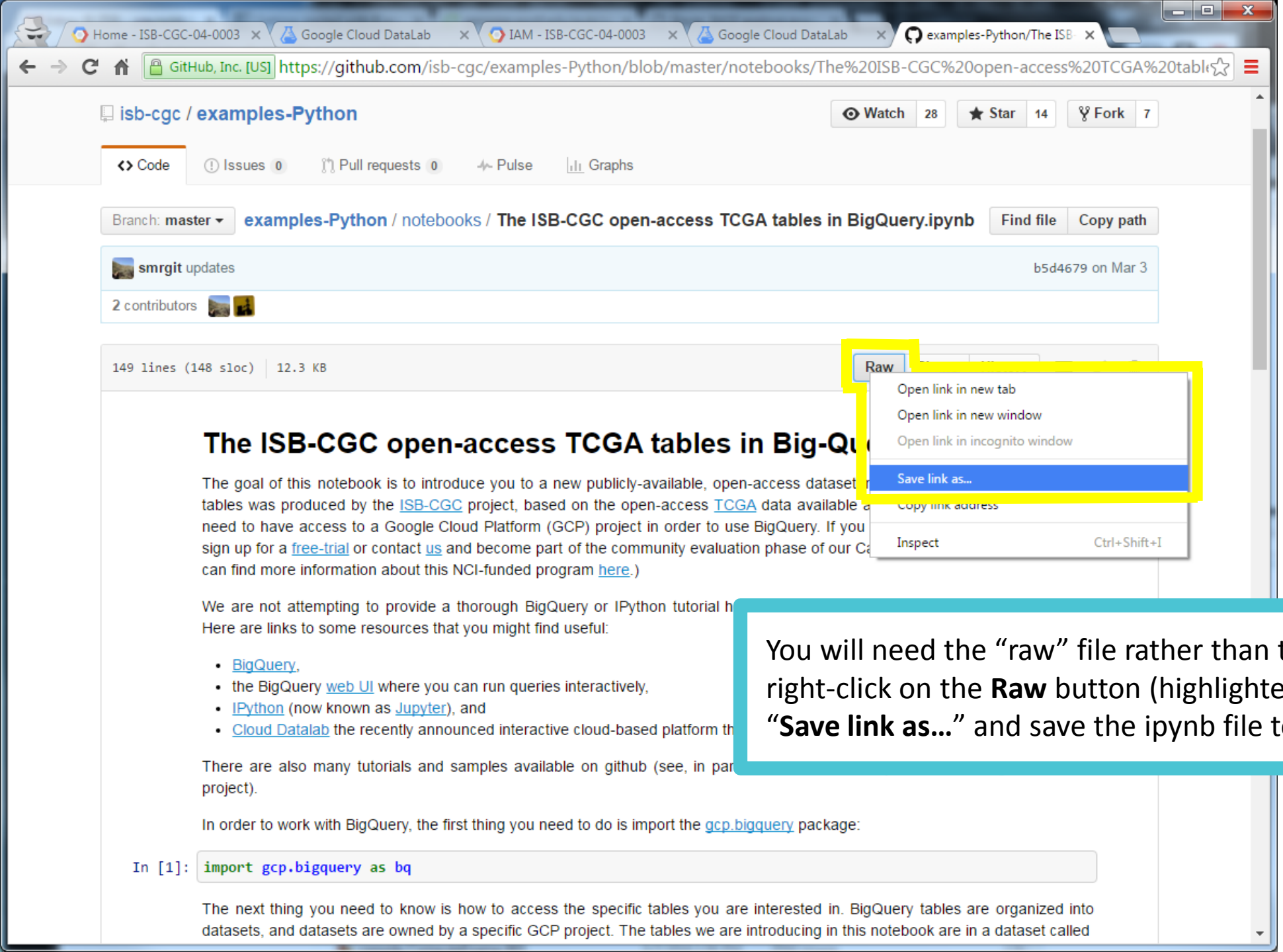
From here you can click on an individual ipynb file to see its contents.

For example, click on the one titled "**The ISB-CGC open-access TCGA tables**" (the second one from the bottom), which is an introductory notebook.

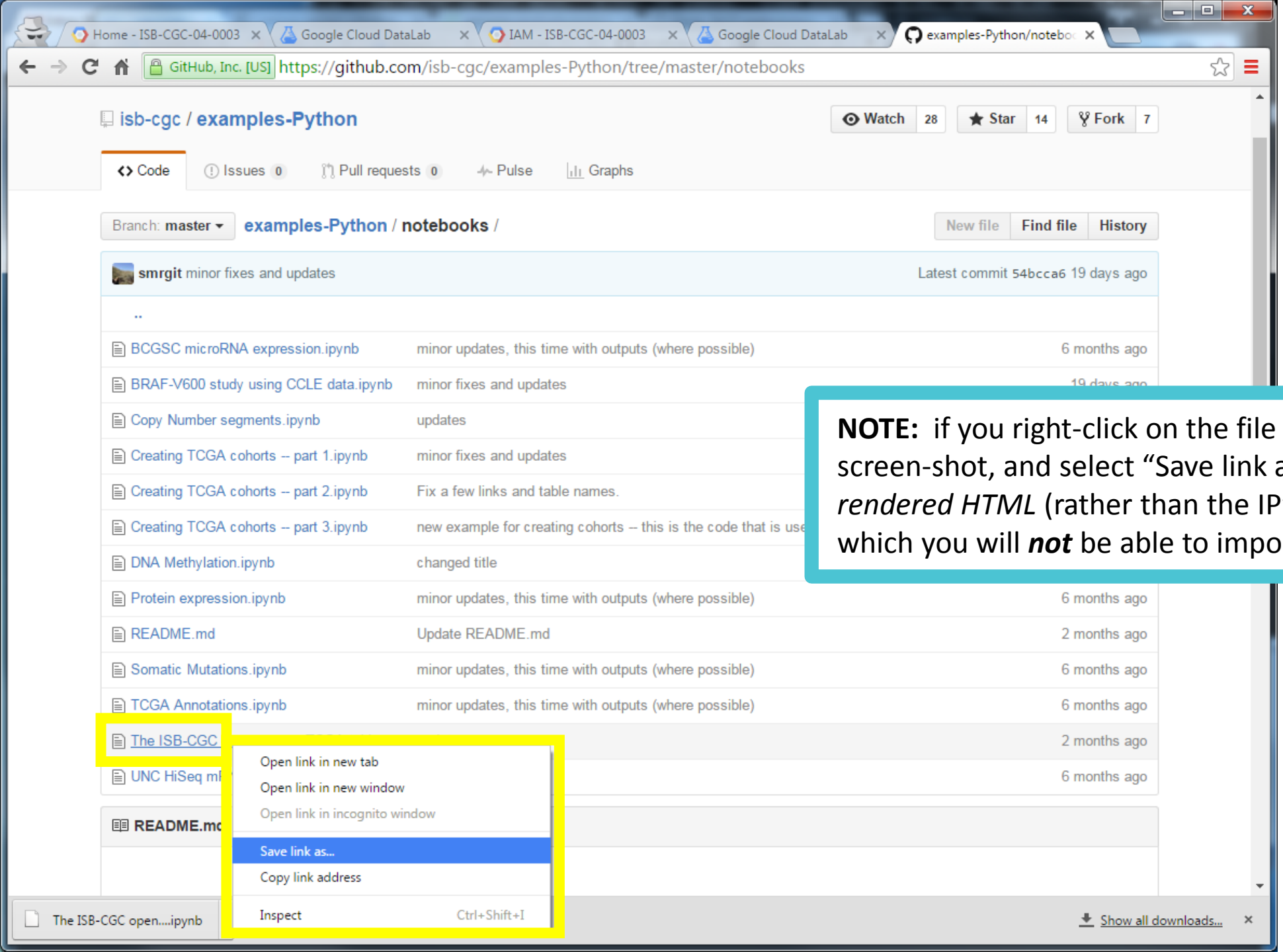


When you click on an ipynb file in GitHub, you see it *rendered* (as HTML) much as it looks within the Jupyter (IPython) interactive computing environment. The *raw* file is actually a JSON document which can contain a mix of text, source code, metadata, and rich media output.

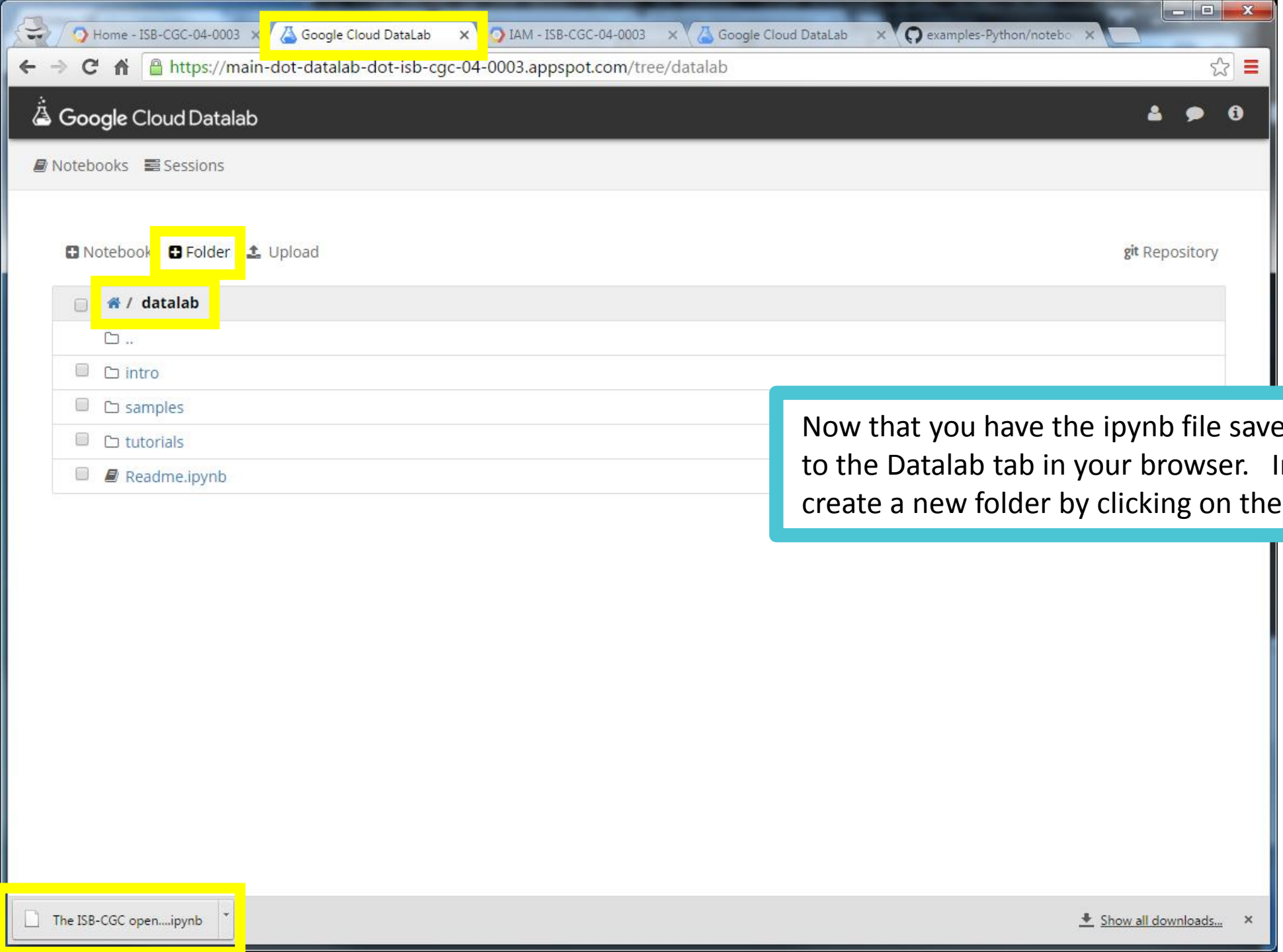
The easiest way to bring one of these example notebooks from GitHub into your running instance of Cloud Datalab is a two-step process: 1) save the ipynb file locally, and 2) upload it to Datalab. We will walk you through this process in the next few slides.



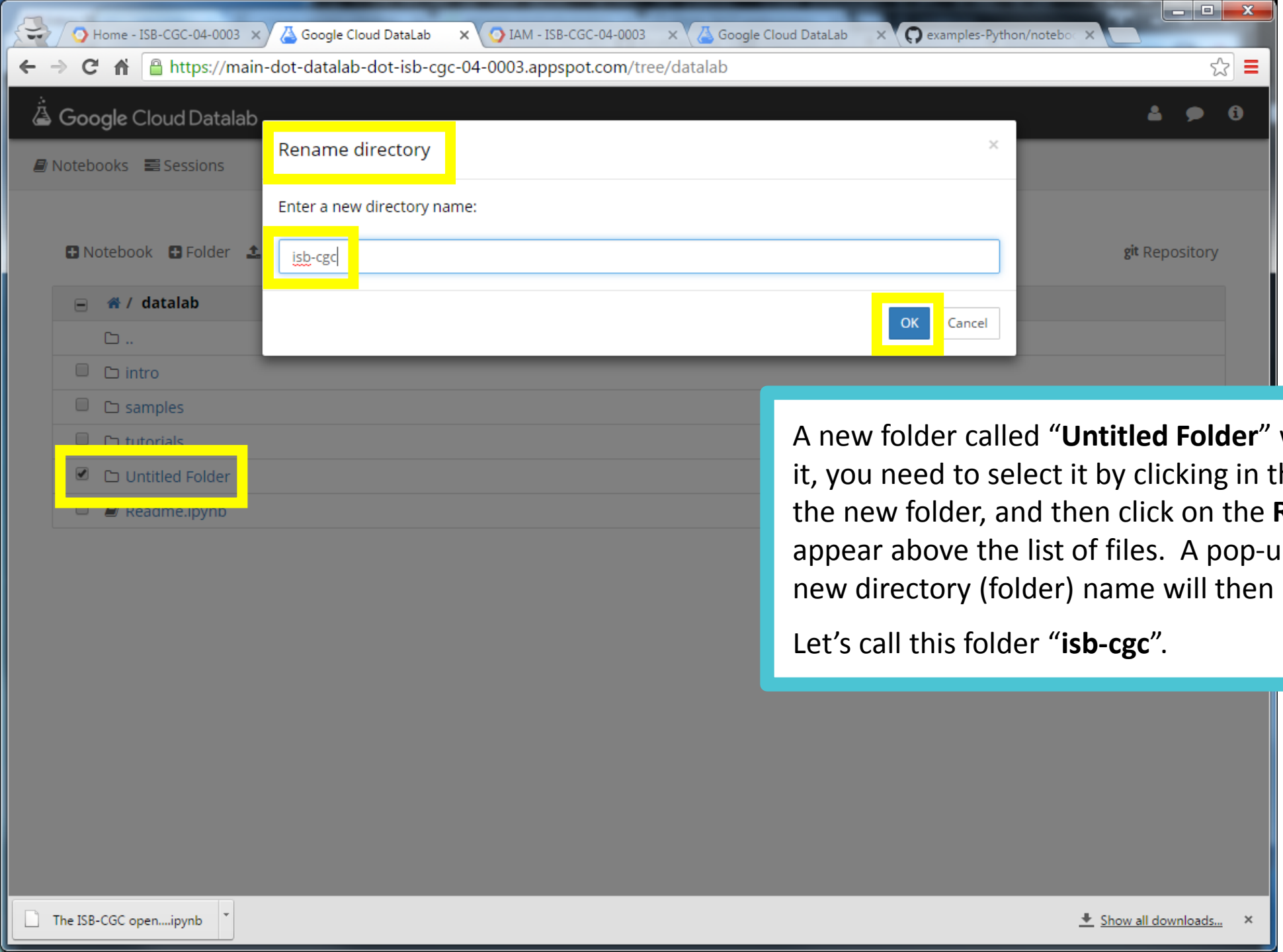
You will need the “raw” file rather than the rendered HTML, so right-click on the **Raw** button (highlighted in yellow above), select “**Save link as…**” and save the ipynb file to your local machine.



NOTE: if you right-click on the file name as shown in this screen-shot, and select “Save link as…” you will be saving the *rendered HTML* (rather than the IPython JSON document) which you will ***not*** be able to import into Datalab.

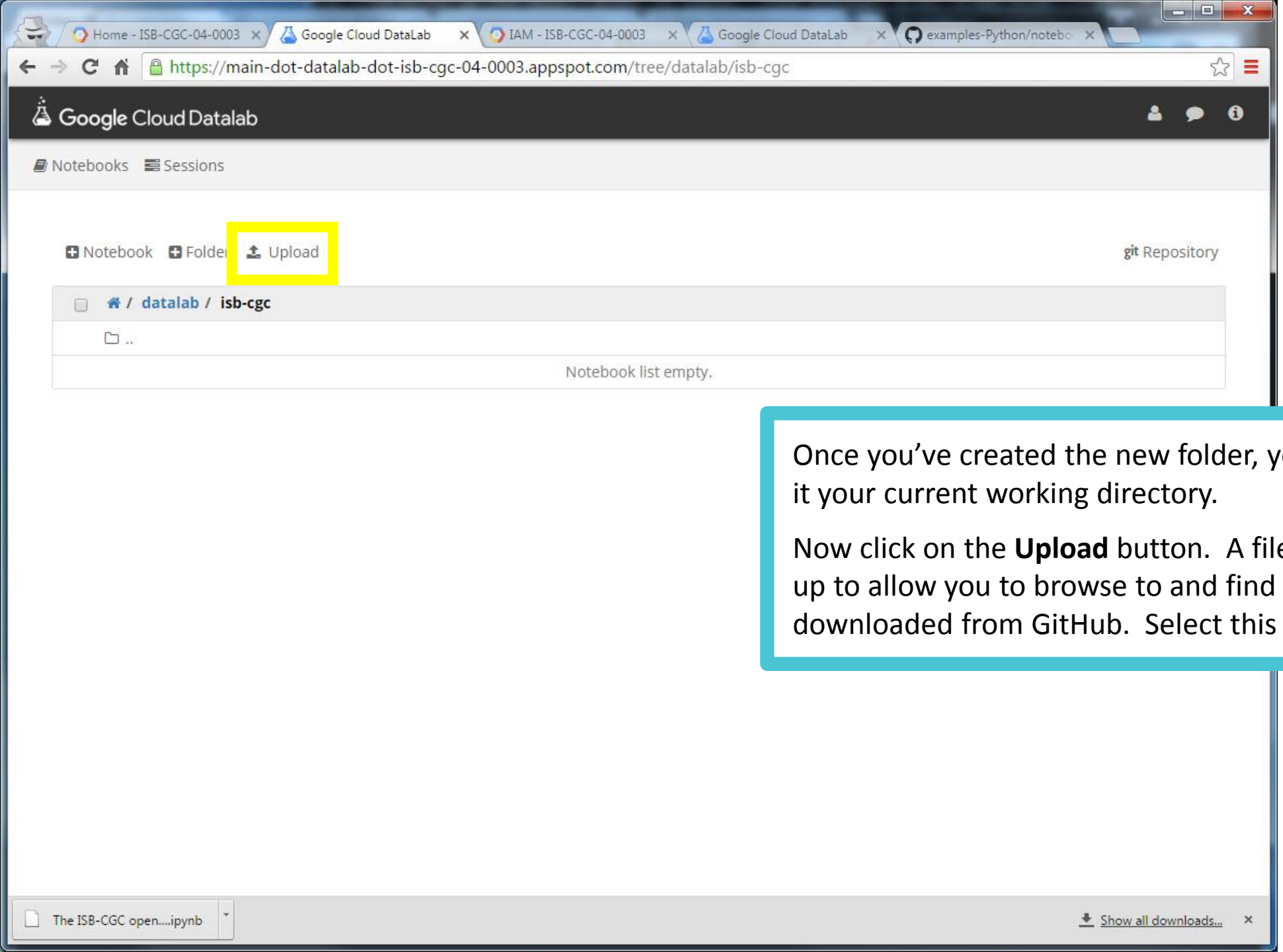


Now that you have the ipynb file saved locally, you can return to the Datalab tab in your browser. In the "datalab" folder let's create a new folder by clicking on the "**Add Folder**" button.



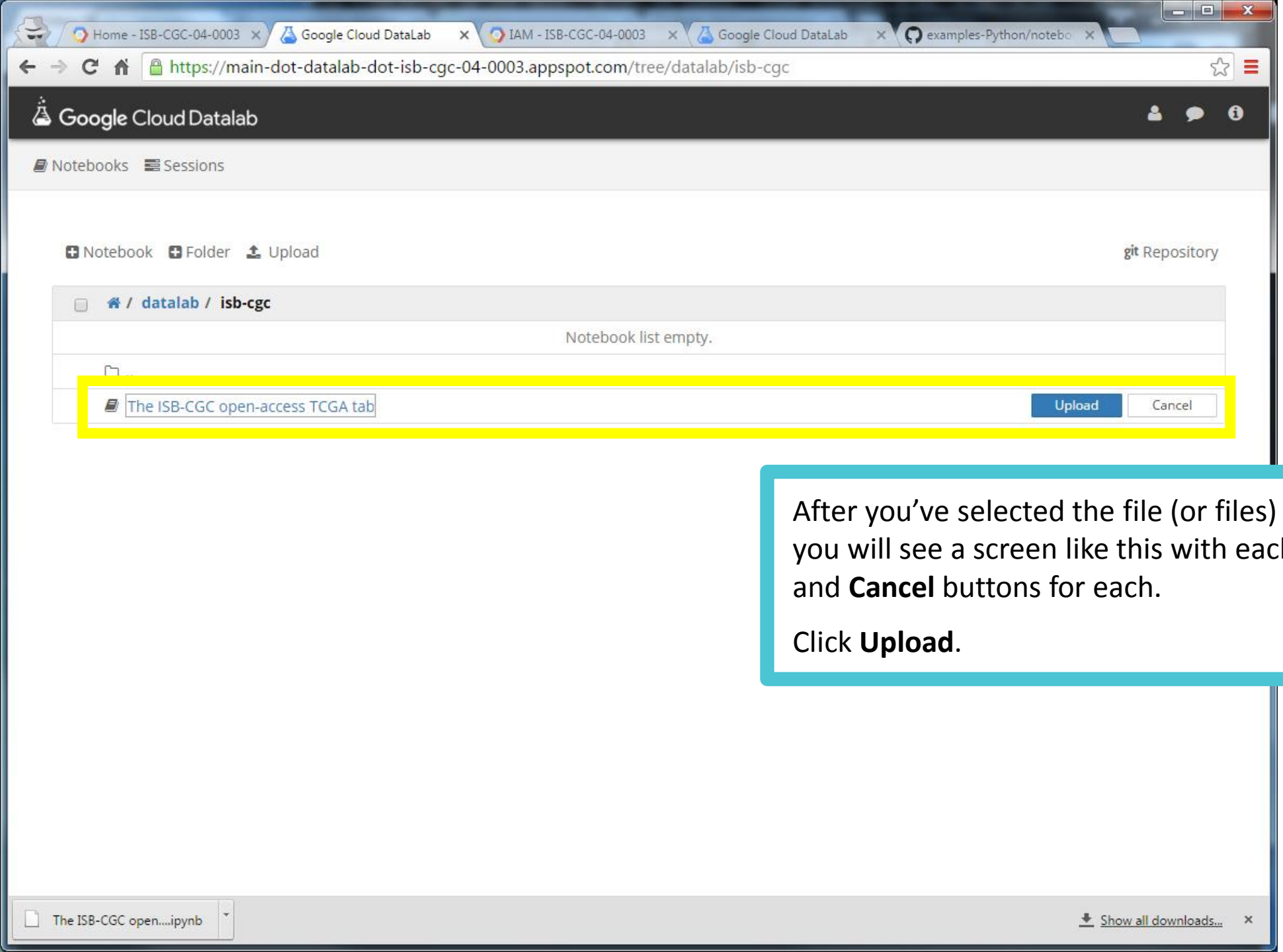
A new folder called "**Untitled Folder**" will be added. To rename it, you need to select it by clicking in the **checkbox** to the left of the new folder, and then click on the **Rename** button that will appear above the list of files. A pop-up prompting you for the new directory (folder) name will then appear.

Let's call this folder "**isb-cgc**".



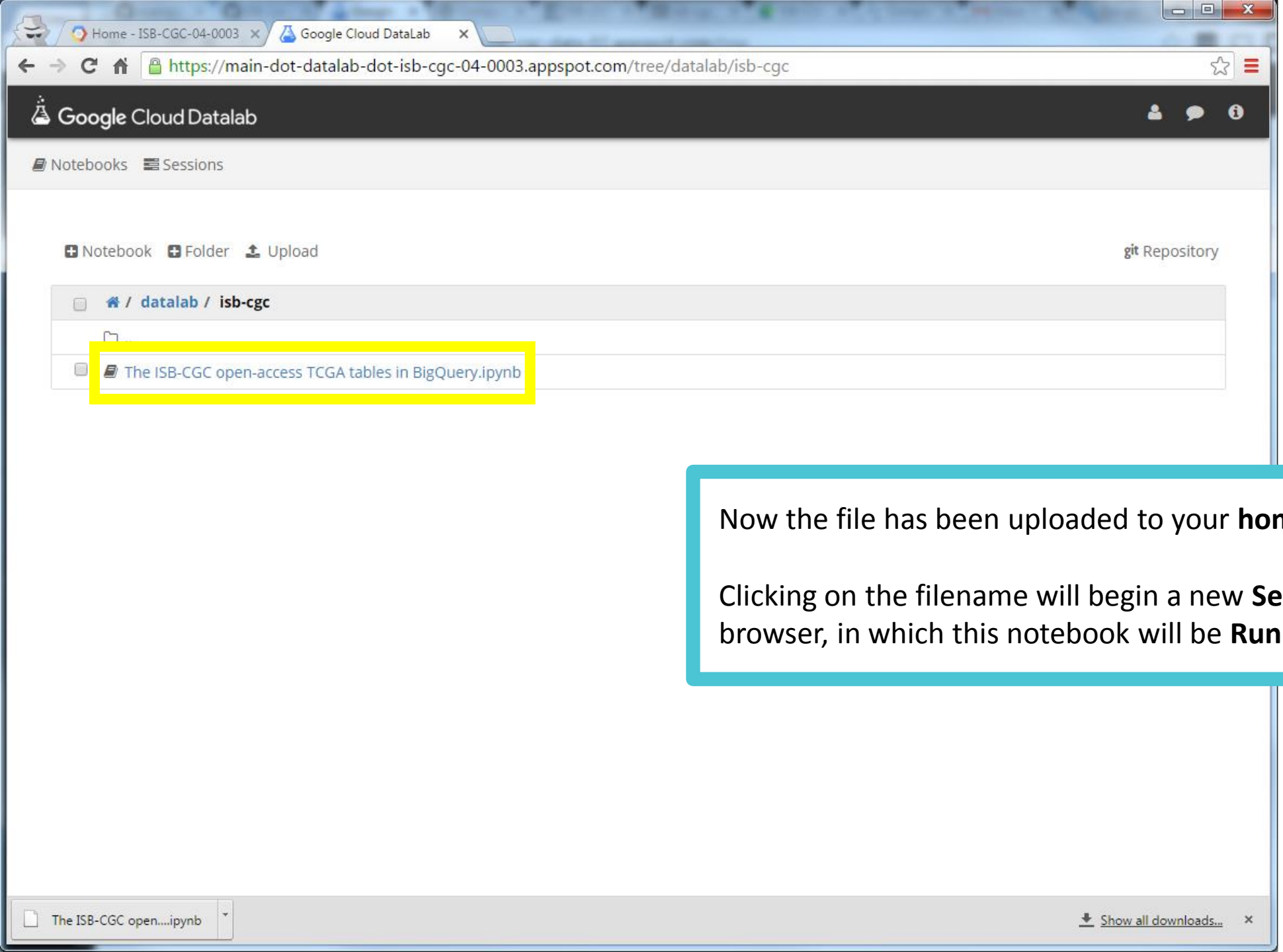
Once you've created the new folder, you can click on it to make it your current working directory.

Now click on the **Upload** button. A file-selection box will open up to allow you to browse to and find the ipynb file that you downloaded from GitHub. Select this file and click **Open**.



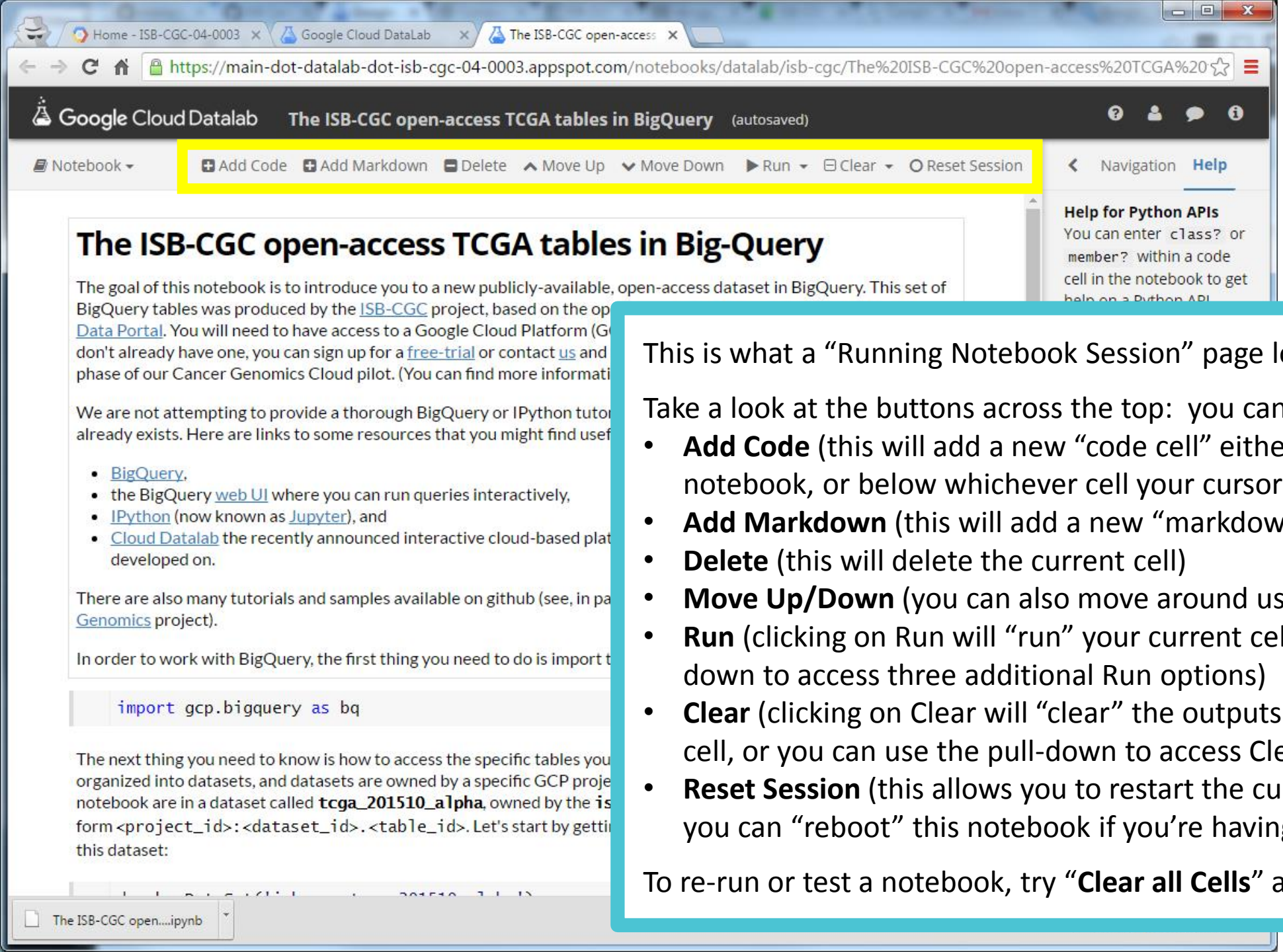
After you've selected the file (or files) that you want to upload, you will see a screen like this with each file listed and **Upload** and **Cancel** buttons for each.

Click **Upload**.



Now the file has been uploaded to your **home/datalab/isb-cgc** folder.

Clicking on the filename will begin a new **Session** in a new tab of your browser, in which this notebook will be **Running**.

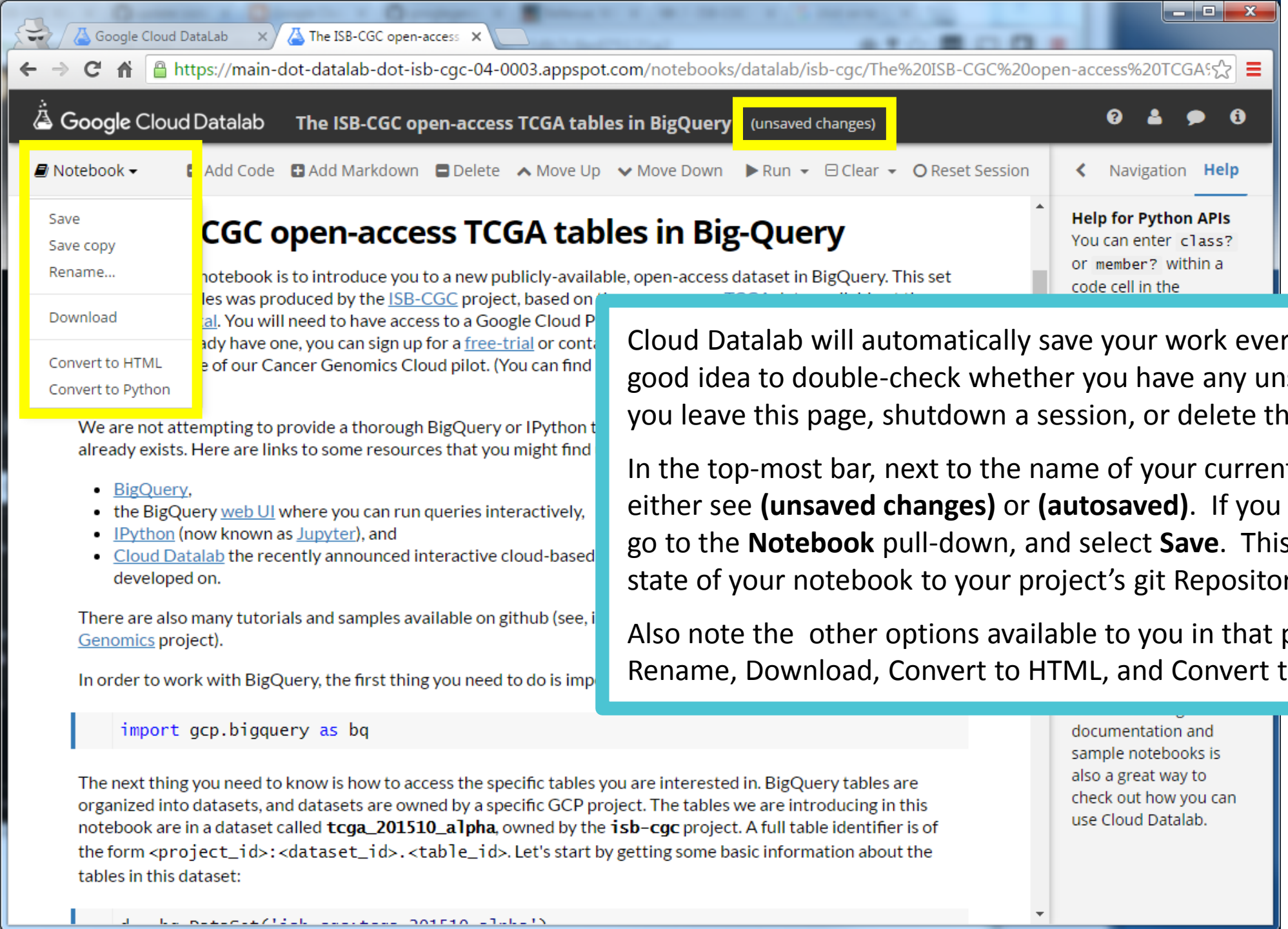


This is what a "Running Notebook Session" page looks like.

Take a look at the buttons across the top: you can

- **Add Code** (this will add a new "code cell" either at the bottom of the notebook, or below whichever cell your cursor is in)
- **Add Markdown** (this will add a new "markdown cell")
- **Delete** (this will delete the current cell)
- **Move Up/Down** (you can also move around using the mouse)
- **Run** (clicking on Run will "run" your current cell, or you can use the pull-down to access three additional Run options)
- **Clear** (clicking on Clear will "clear" the outputs only of your current code cell, or you can use the pull-down to access Clear all Cells)
- **Reset Session** (this allows you to restart the current kernel – essentially you can "reboot" this notebook if you're having problems).

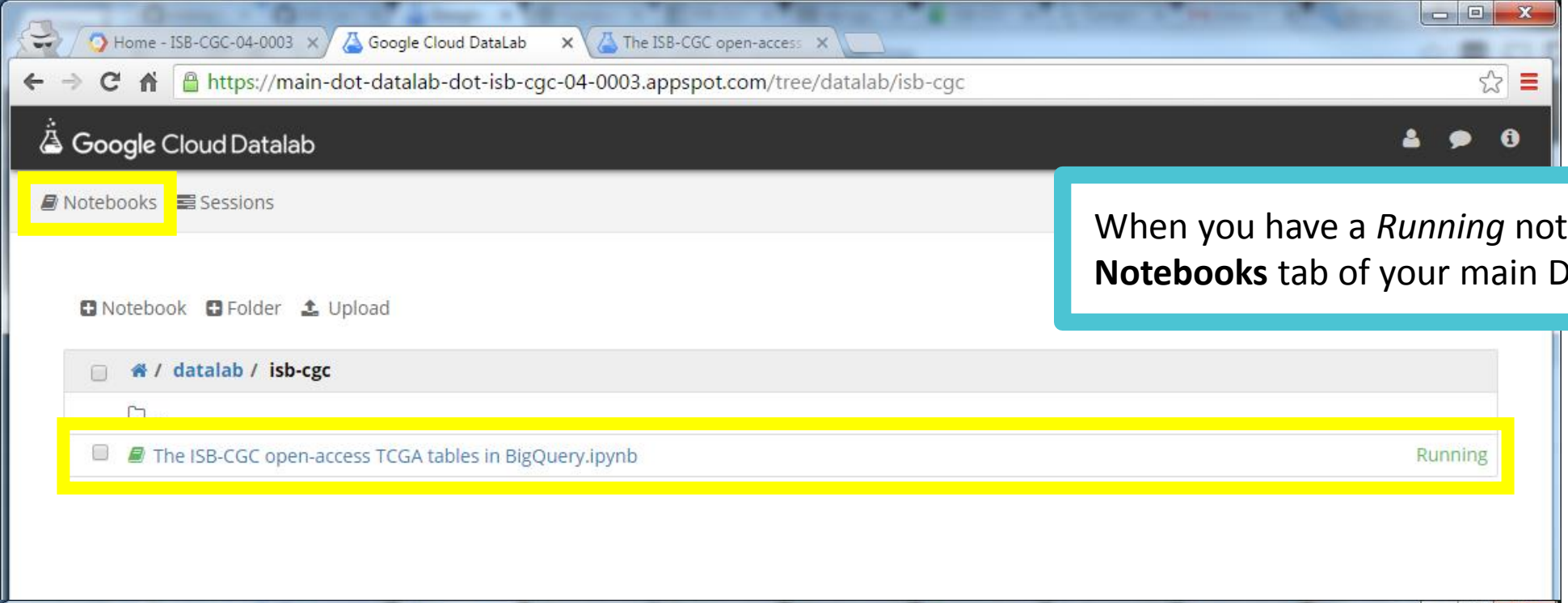
To re-run or test a notebook, try "**Clear all Cells**" and then "**Run all Cells**"



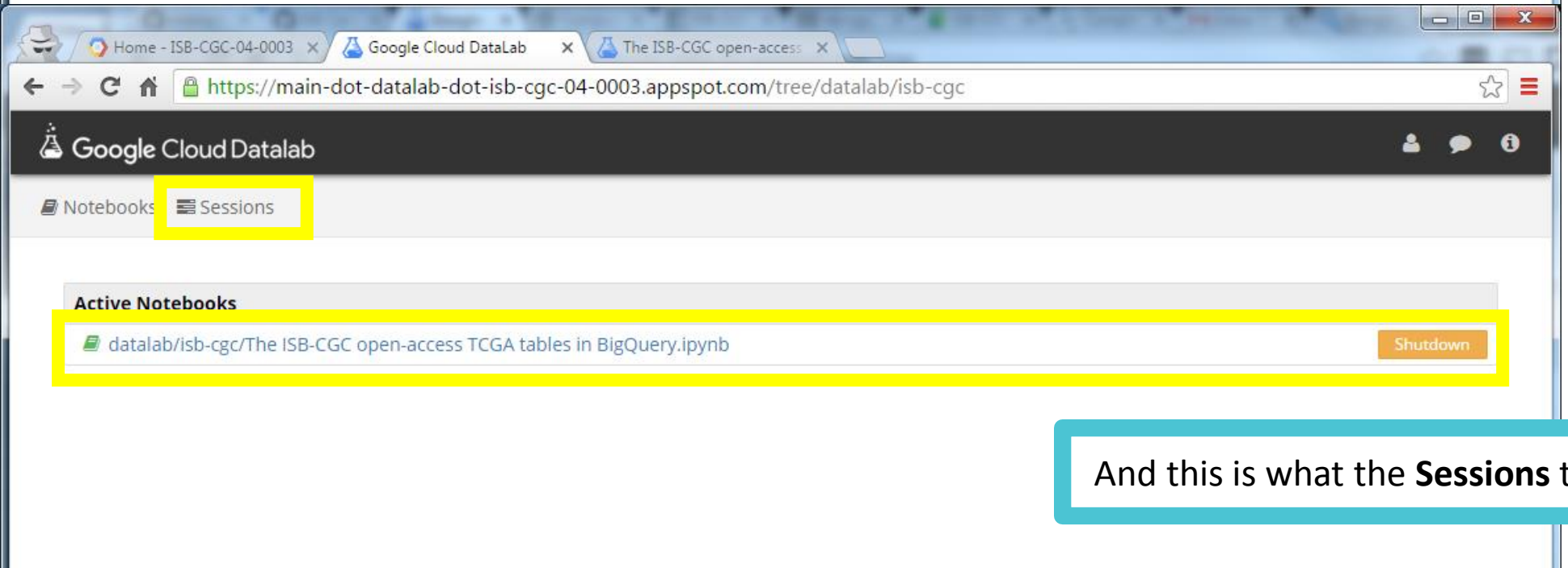
Cloud Datalab will automatically save your work every few minutes, but it's a good idea to double-check whether you have any unsaved changes before you leave this page, shutdown a session, or delete the Datalab VM.

In the top-most bar, next to the name of your current notebook, you will either see **(unsaved changes)** or **(autosaved)**. If you have unsaved changes, go to the **Notebook** pull-down, and select **Save**. This will save the current state of your notebook to your project's git Repository.

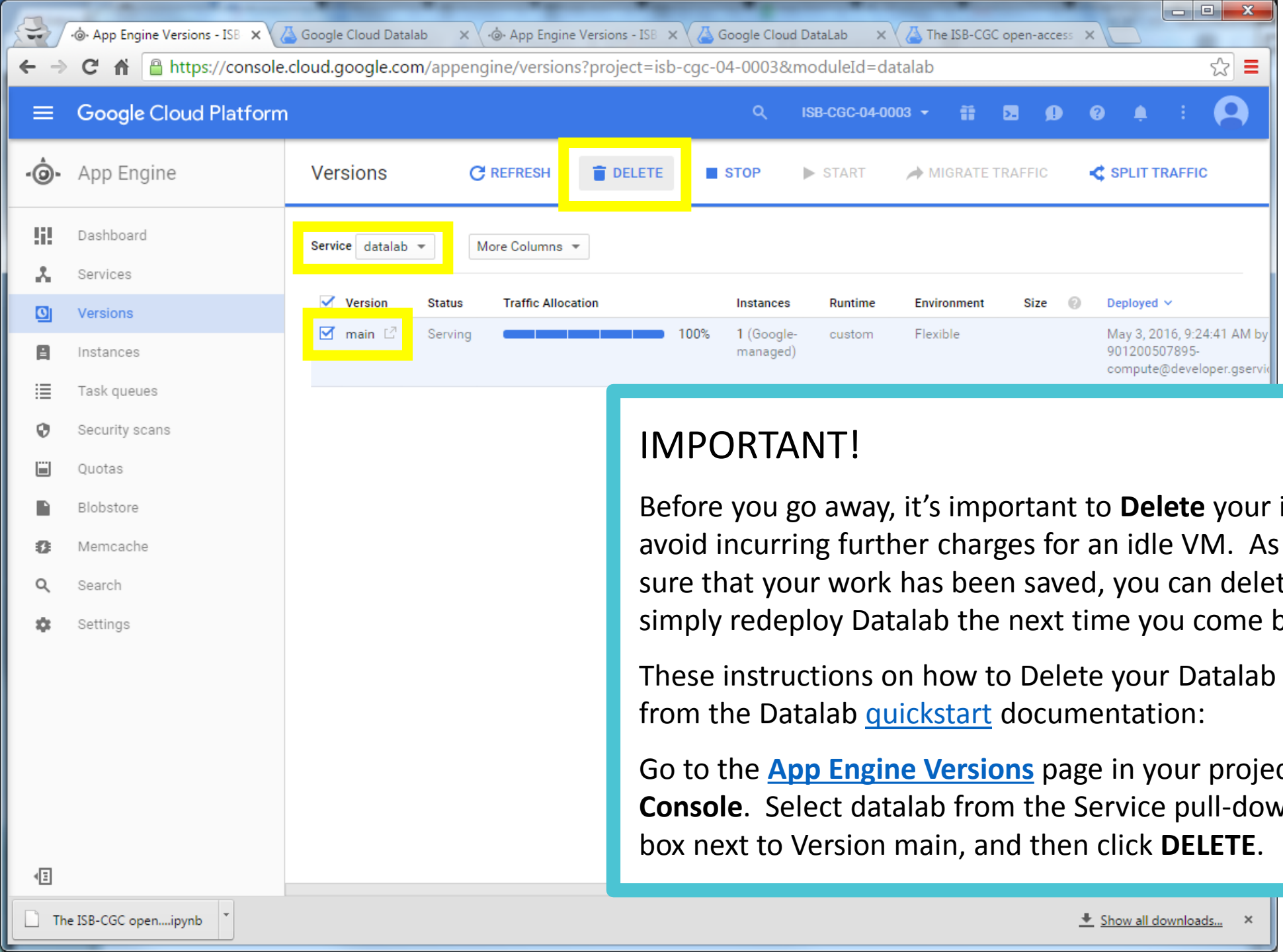
Also note the other options available to you in that pull-down: Save copy, Rename, Download, Convert to HTML, and Convert to Python.



When you have a *Running* notebook, this is what the **Notebooks** tab of your main Datalab page will look like.



And this is what the **Sessions** tab will look like.

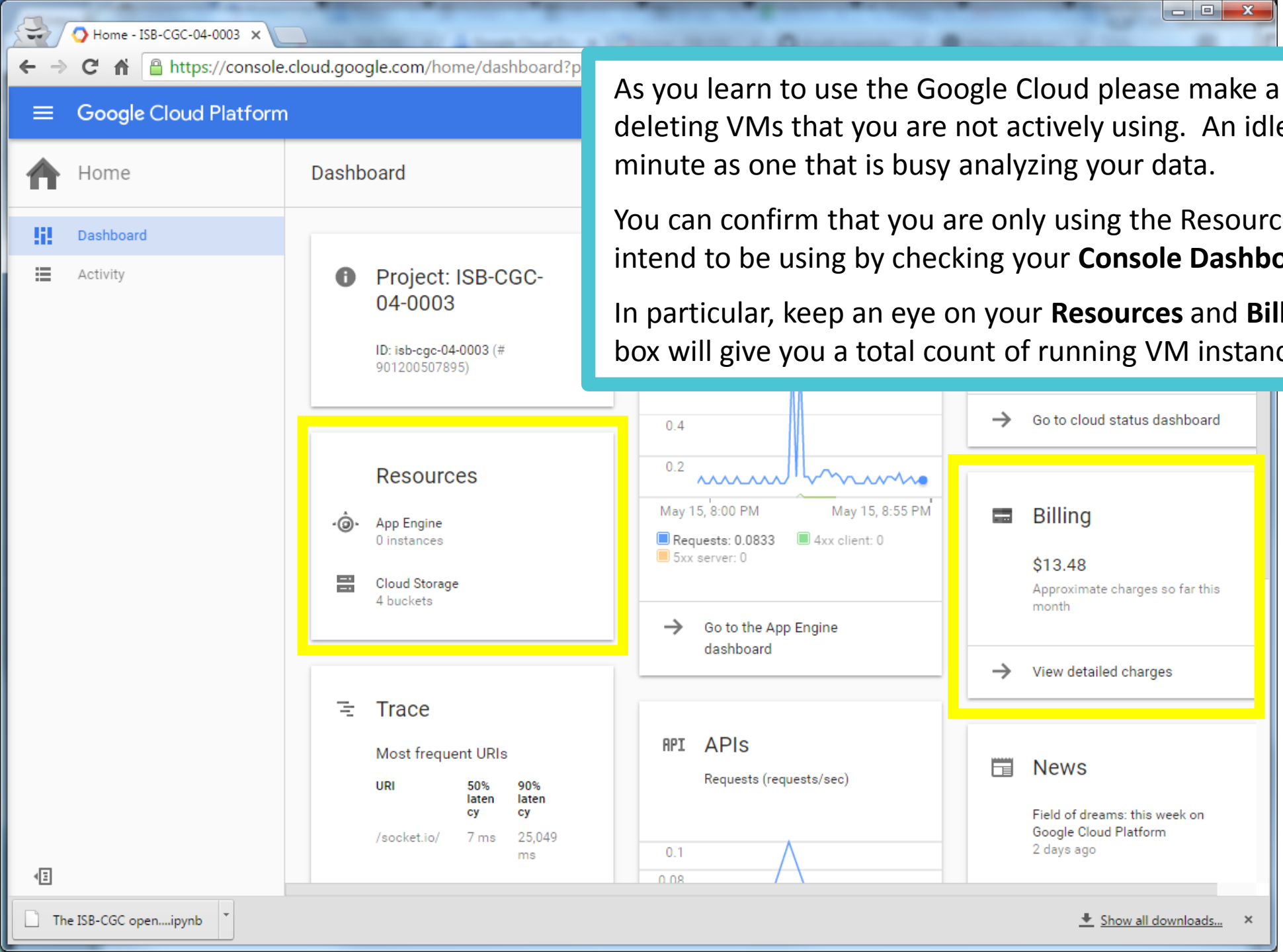


IMPORTANT!

Before you go away, it's important to **Delete** your instance of Datalab to avoid incurring further charges for an idle VM. As long as you have made sure that your work has been saved, you can delete the Datalab VM, and simply redeploy Datalab the next time you come back.

These instructions on how to Delete your Datalab VM instance are taken from the Datalab quickstart documentation:

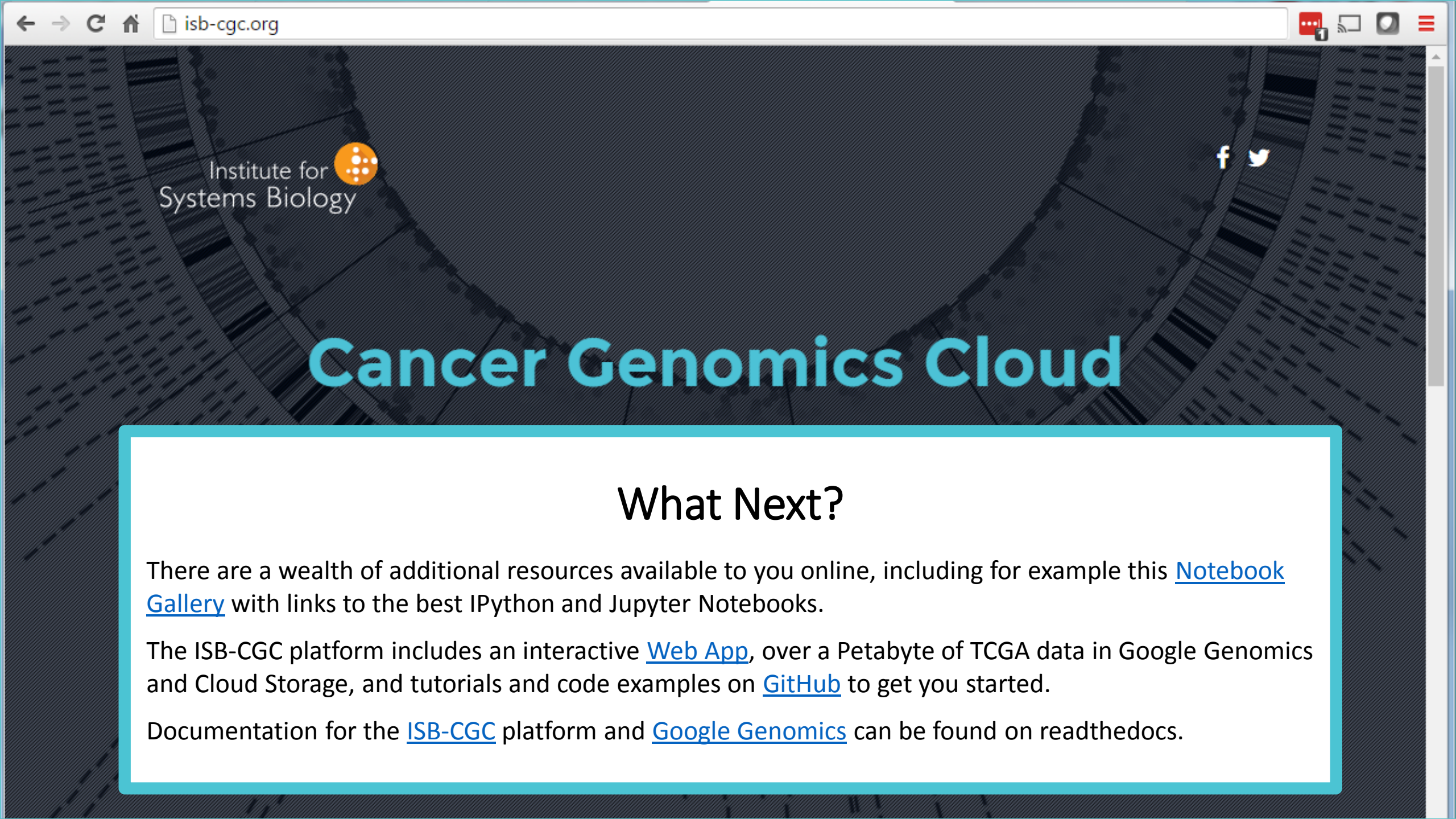
Go to the **App Engine Versions** page in your project's **Cloud Platform Console**. Select datalab from the Service pull-down, then click the check-box next to Version main, and then click **DELETE**.



As you learn to use the Google Cloud please make a habit of shutting down or deleting VMs that you are not actively using. An idle VM costs as much per minute as one that is busy analyzing your data.

You can confirm that you are only using the Resources that you expect and intend to be using by checking your **Console Dashboard** page daily.

In particular, keep an eye on your **Resources** and **Billing** details. The Resources box will give you a total count of running VM instances and Storage buckets.


isb-cgc.org
Institute for Systems Biology
Cancer Genomics Cloud
What Next?
There are a wealth of additional resources available to you online, including for example this Notebook Gallery with links to the best IPython and Jupyter Notebooks.
The ISB-CGC platform includes an interactive Web App, over a Petabyte of TCGA data in Google Genomics and Cloud Storage, and tutorials and code examples on GitHub to get you started.
Documentation for the ISB-CGC platform and Google Genomics can be found on readthedocs.