# How to run a notebook Locally?


**ZerocostDL4Mic | an open platform to use Deep Learning in microscopy**

**GitHub page** https://github.com/HenriquesLab/ZeroCostDL4Mic/wiki

**Video Tutorial:** https://github.com/HenriquesLab/ZeroCostDL4Mic/wiki

Google Colab lets users connect to a local runtime using Jupyter notebook on. This allows you to run Zerocost4DLMic notebooks on your local machine and have access to your local file system. This can be helpful for those who wish to train models using their own GPU power.

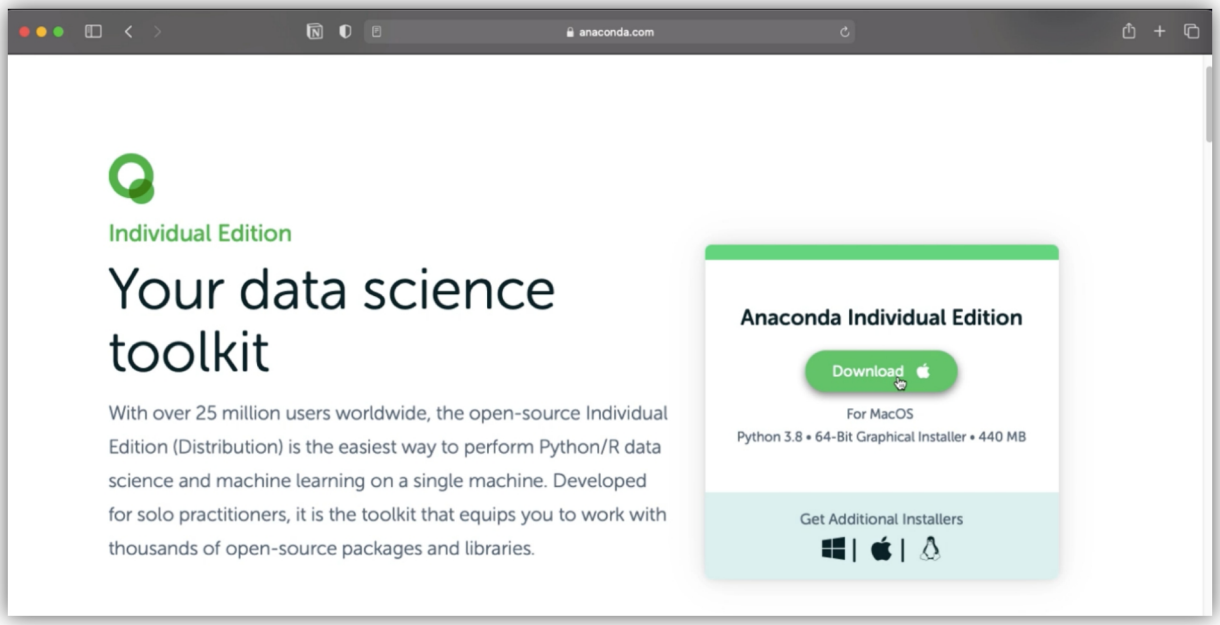
To run a notebook locally, first you should create an environment on which the notebook can work. This environment should include all the necessary packages and dependencies that each notebook requires for running. Here is a summary of all steps that you should take for connecting a notebook to local runtime, followed by detailed instructions of each step.

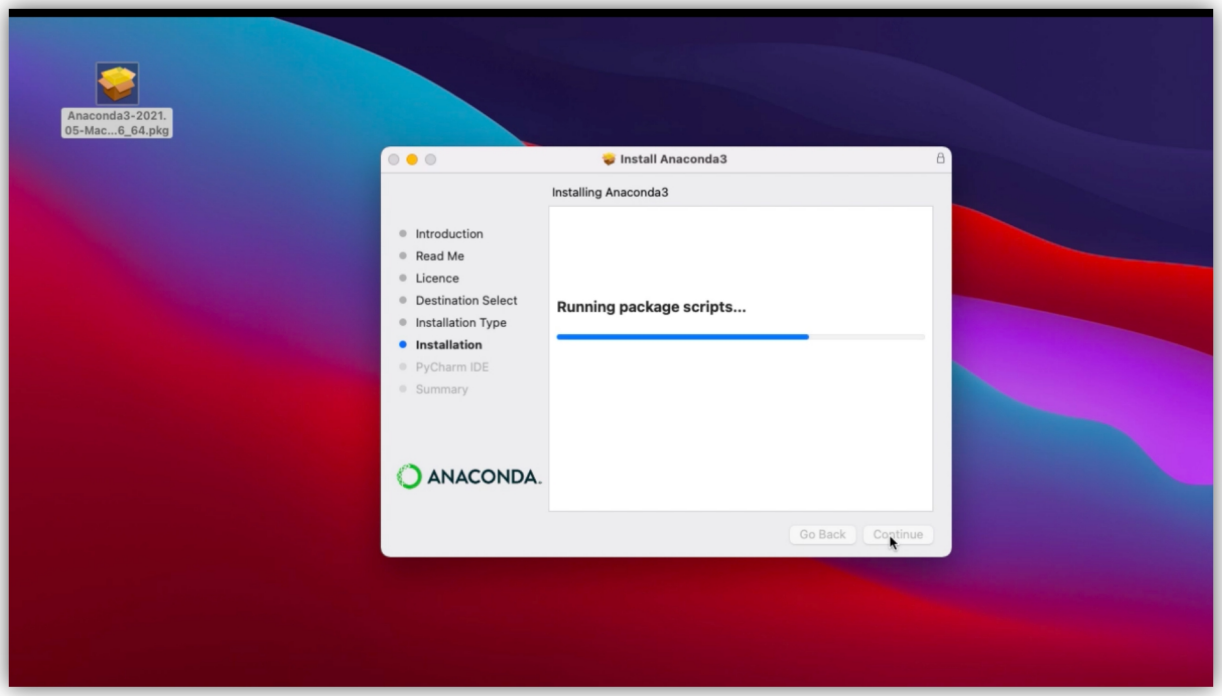
1. *Installing Anaconda*
2. *Creating a new Conda environment in Anaconda*
3. *Connecting Google Colab to Jupyter notebook*
4. *Installing Requirements.txt file*
   *(notebook specific packages and dependencies)*
5. *Running the notebook*

# Step 1: Download and Install Anaconda on your machine.

In this step you need to go to download the individual edition for your operating system. Then simply follow several steps for installation and you should have it on your machine.

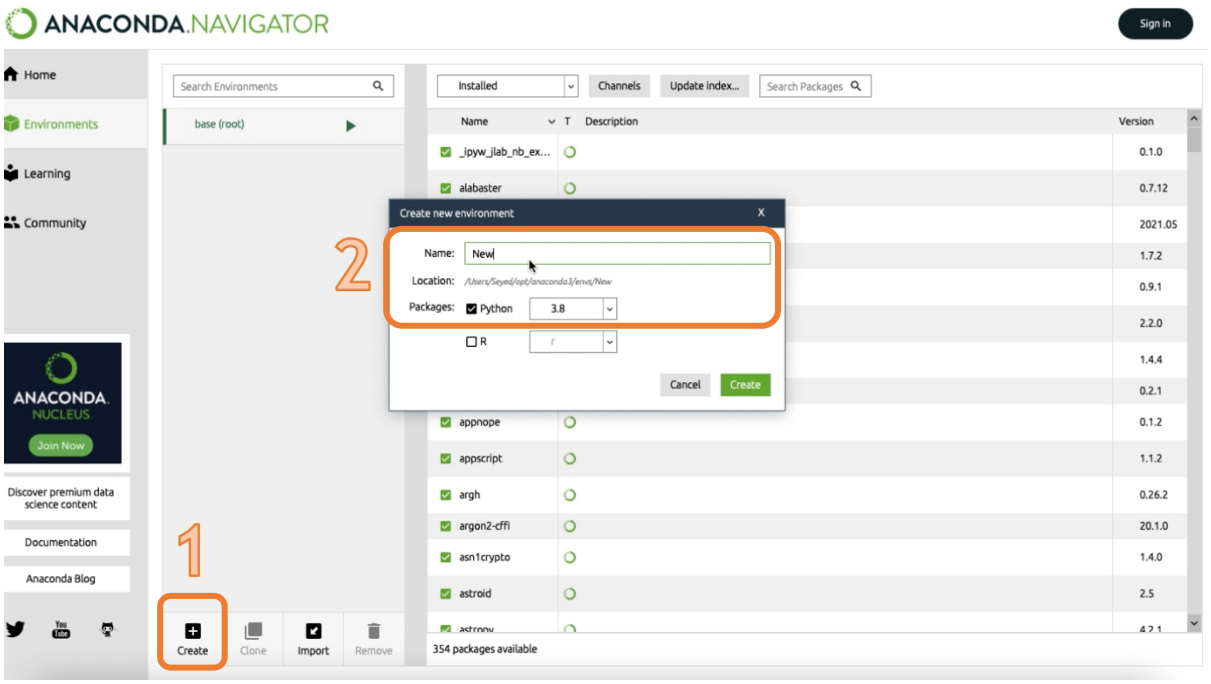
***Download link:*** *https://www.anaconda.com/products/individual*



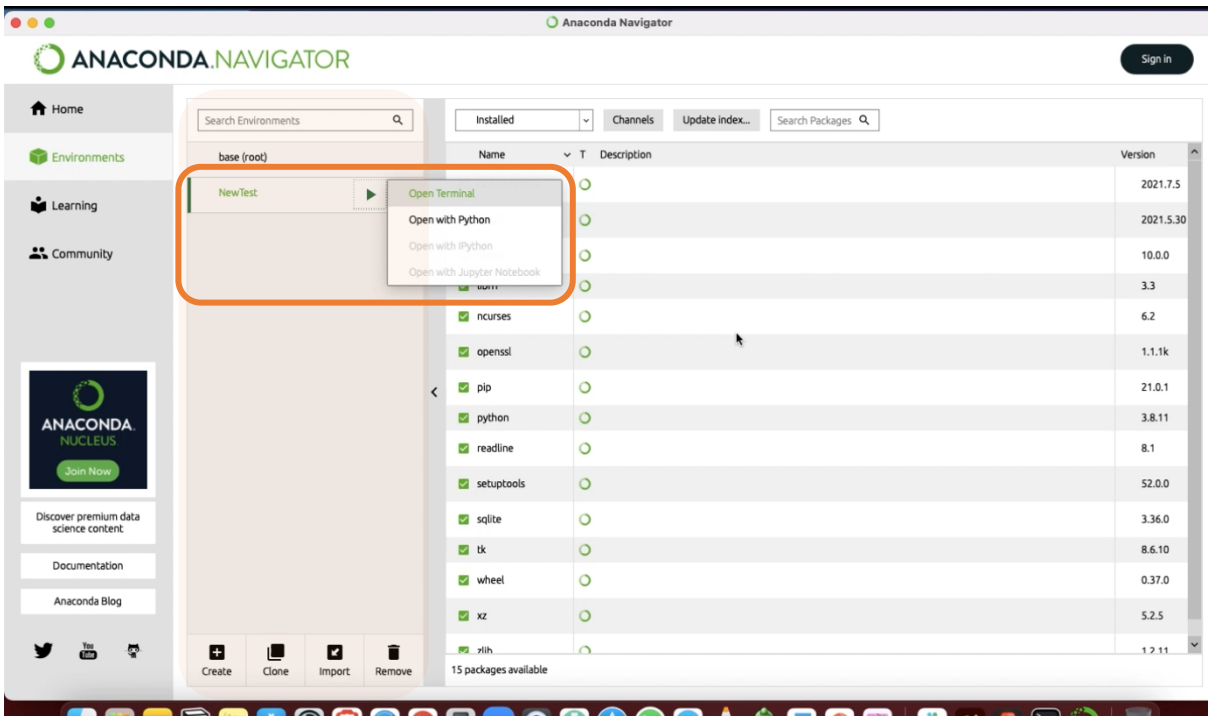



# Step 2: Create a new environment in Anaconda.

In this step you should create an environment for your notebook to be able to run on your local machine. Here, you need to pick a name for your new environment and more importantly to identify what version of Python you want to install on your environment.
*(you can have several environments with specific versions of Python, depending on the packages and dependencies of each notebook or any other codes you want to run.)*




After having your new Conda environment created you should be able to open its terminal for writing code and to see it under the environment list.



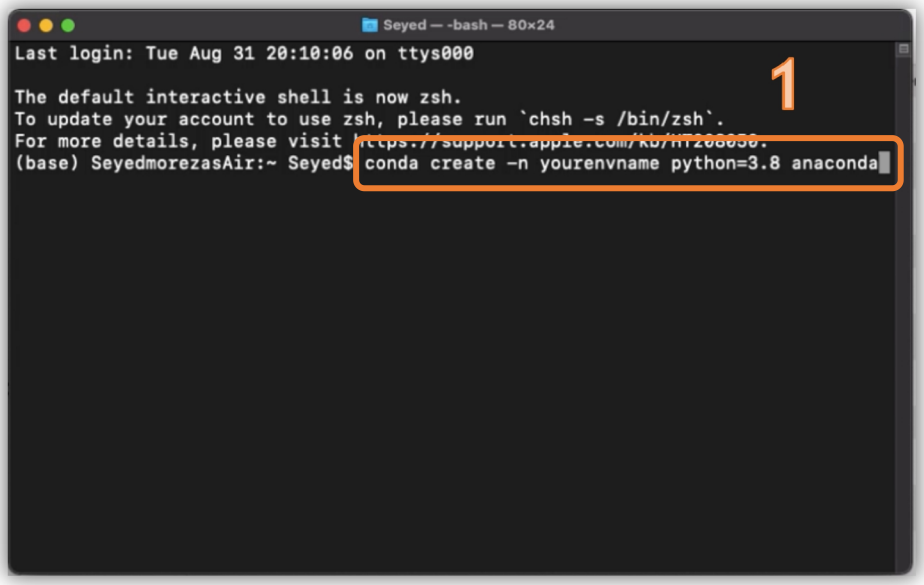
Another way to create a new Conda environment is to write a function in Terminal client. In the terminal client enter the following where ***yourenvname*** is the name you want to call your environment, and **replace *x.x* with the Python version** you want to use *(i.e 3.6.9)*.

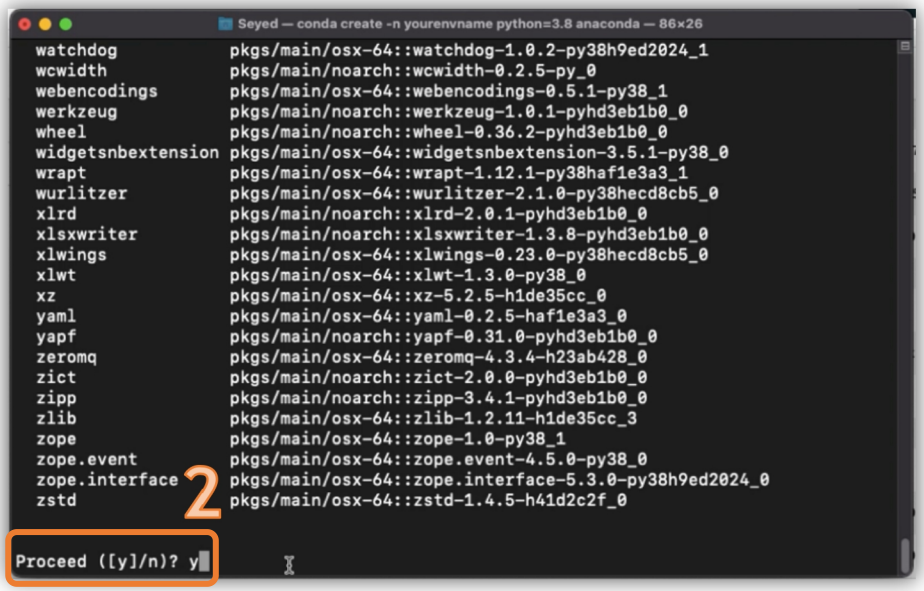
```
conda create -n yourenvname python=x.x anaconda
```

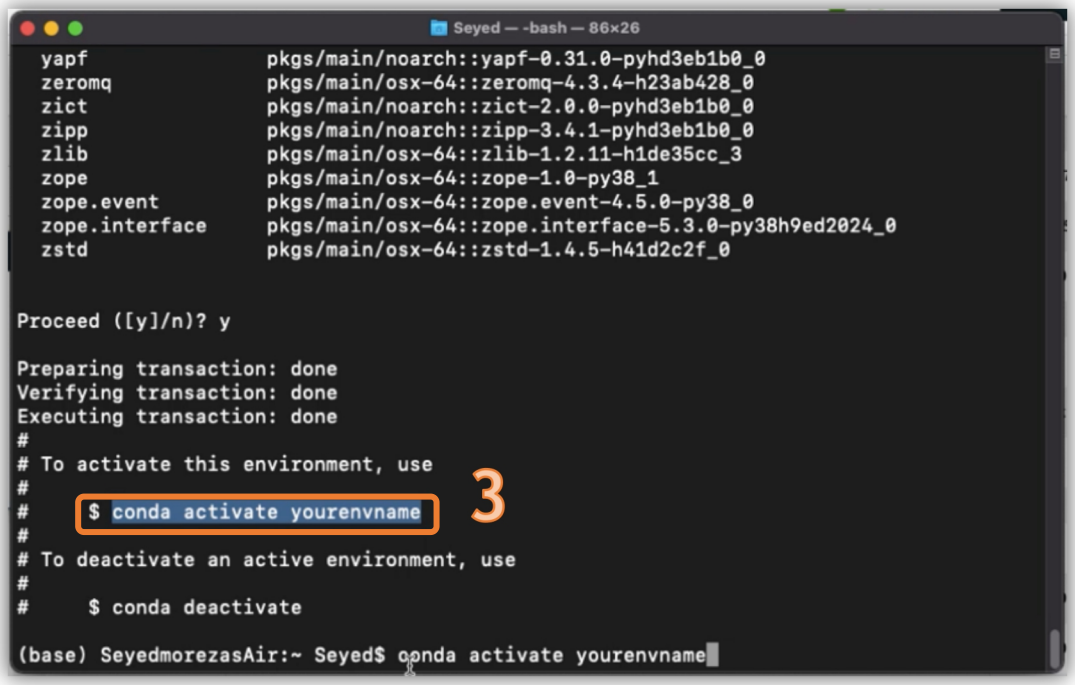
Press `Y` to proceed. This will install the Python version and all the associated anaconda packaged libraries at *"path_to_your_anaconda_location/anaconda/envs/yourenvname"*

After creating the new Conda environment, now you should activate it in the Terminal client. For this, You should run the following code:

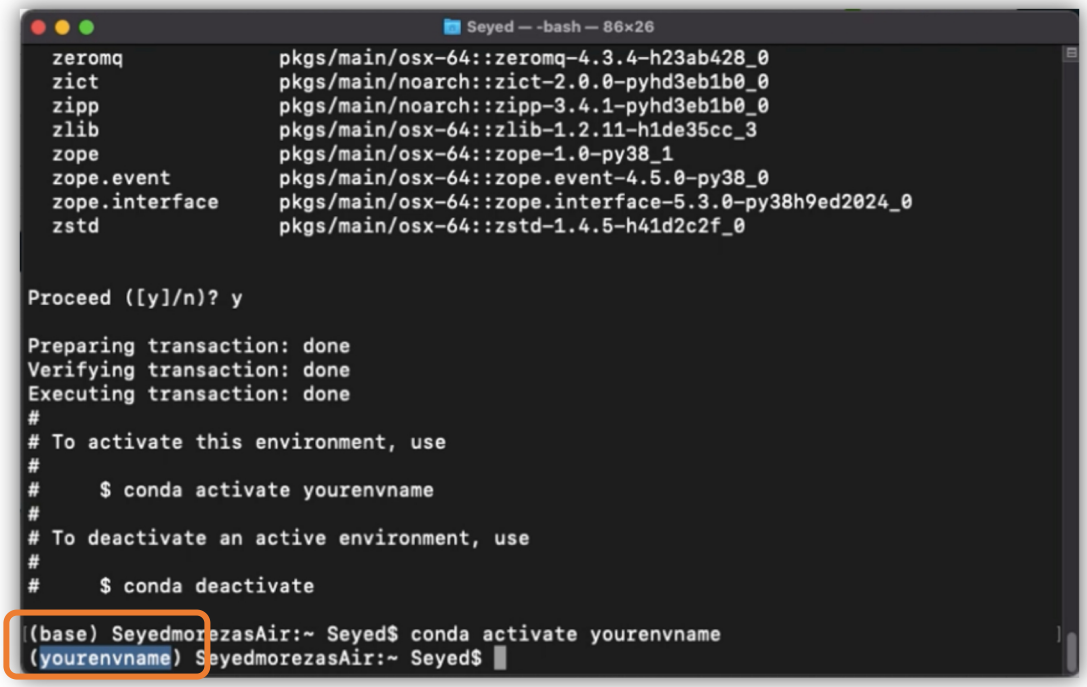
```
conda activate yourenvname
```



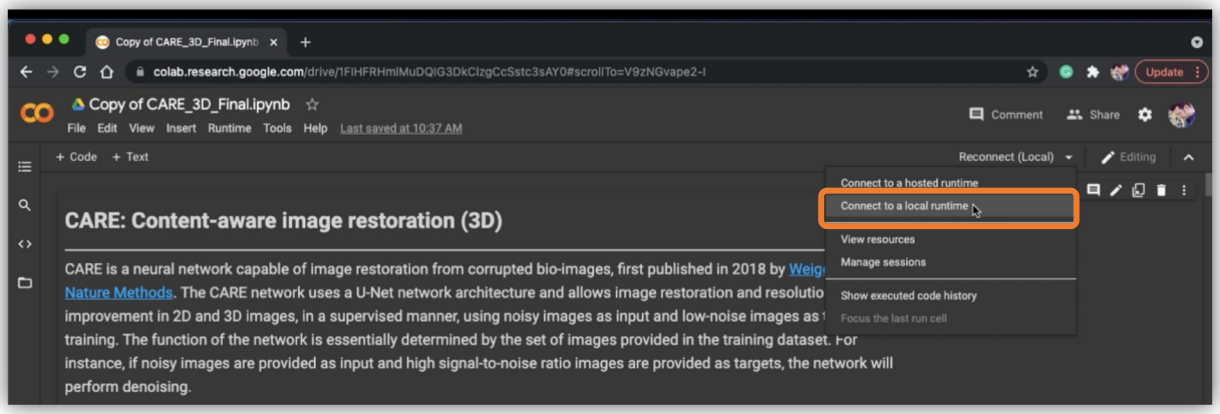



After activating your new Conda envirnonment you should be able to see it replaced the (base) environment.

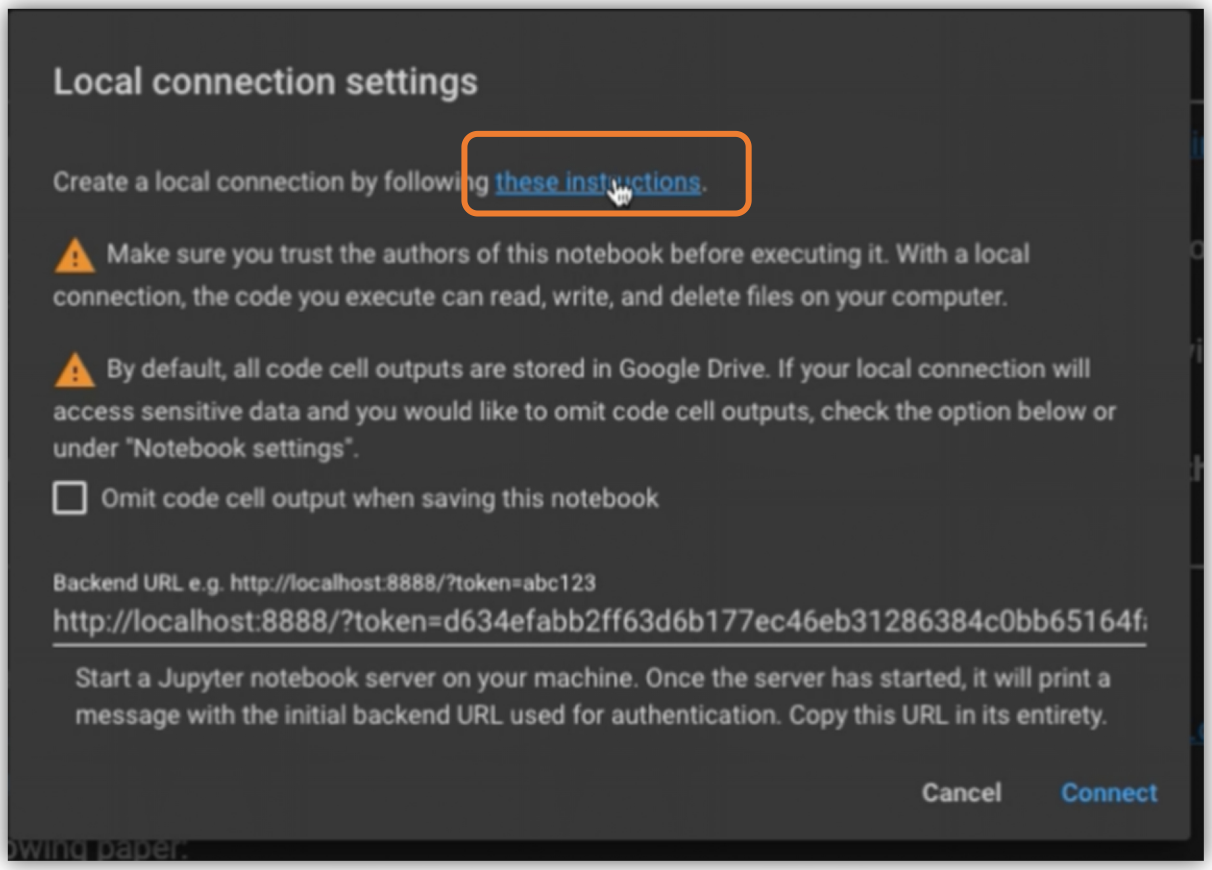


# Step 3: Connecting Google Colab to Local Run time.

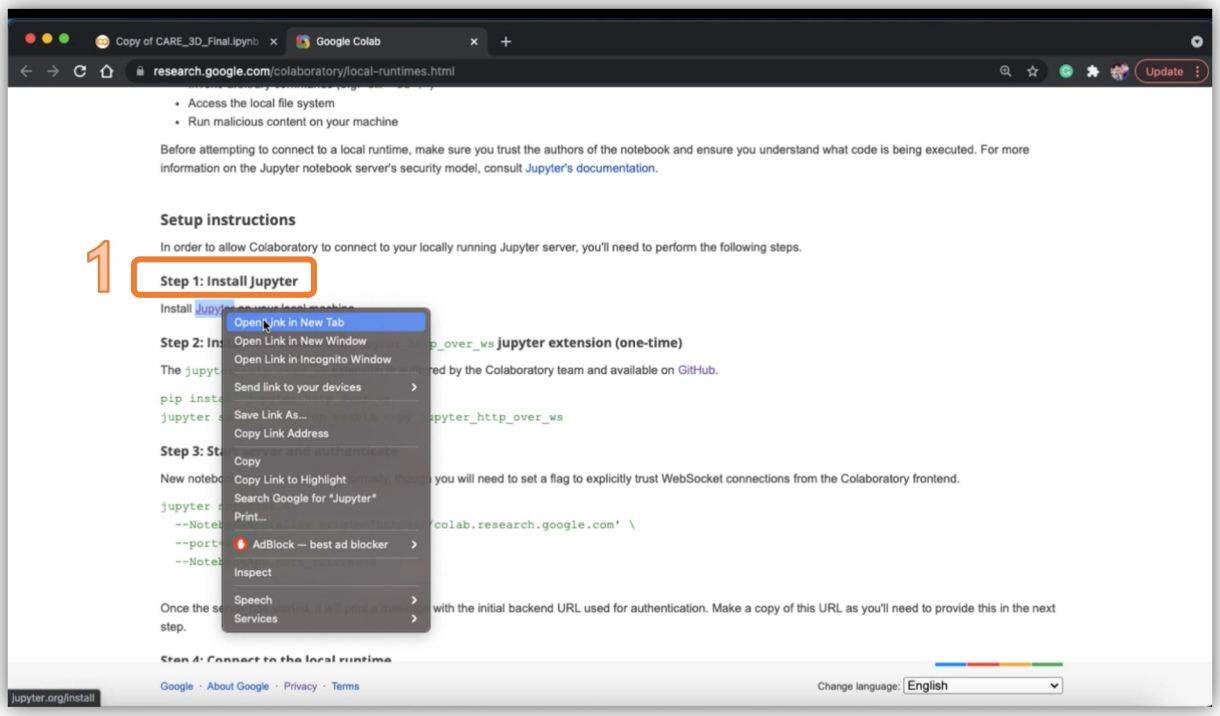
In this step you should connect the Google Colab (as front-end) to a Jupyter notebook server (as back-end) on your local machine. For this, open the drop down menu shown in the picture and choose “connect to local run time”.




Now you should see below pop-op window including a link to Google Colab instructions for local run time connection. Click on the link and follow the next steps.

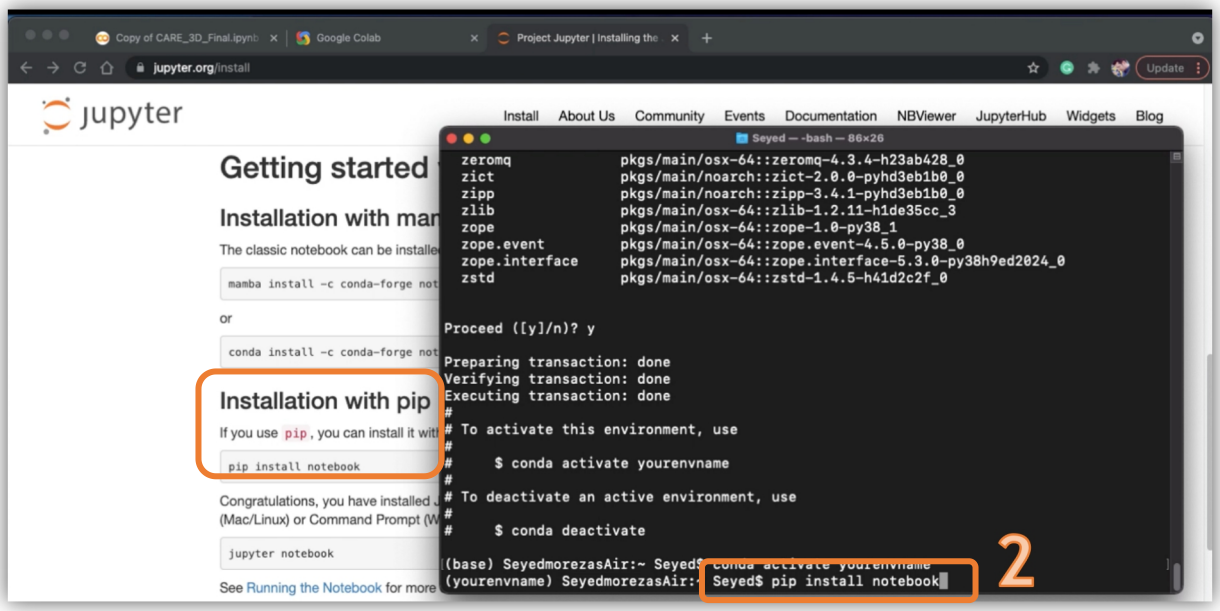


As you can read in the instruction webpage, you should first install Jupyter notebook on your previously made Conda environment. After that, you have to install a jupyter extension (for setting up a server on your machine) and activate it. Finally, you need to start the server and authenticate it to have your back-end ready for Google Colab to connect.



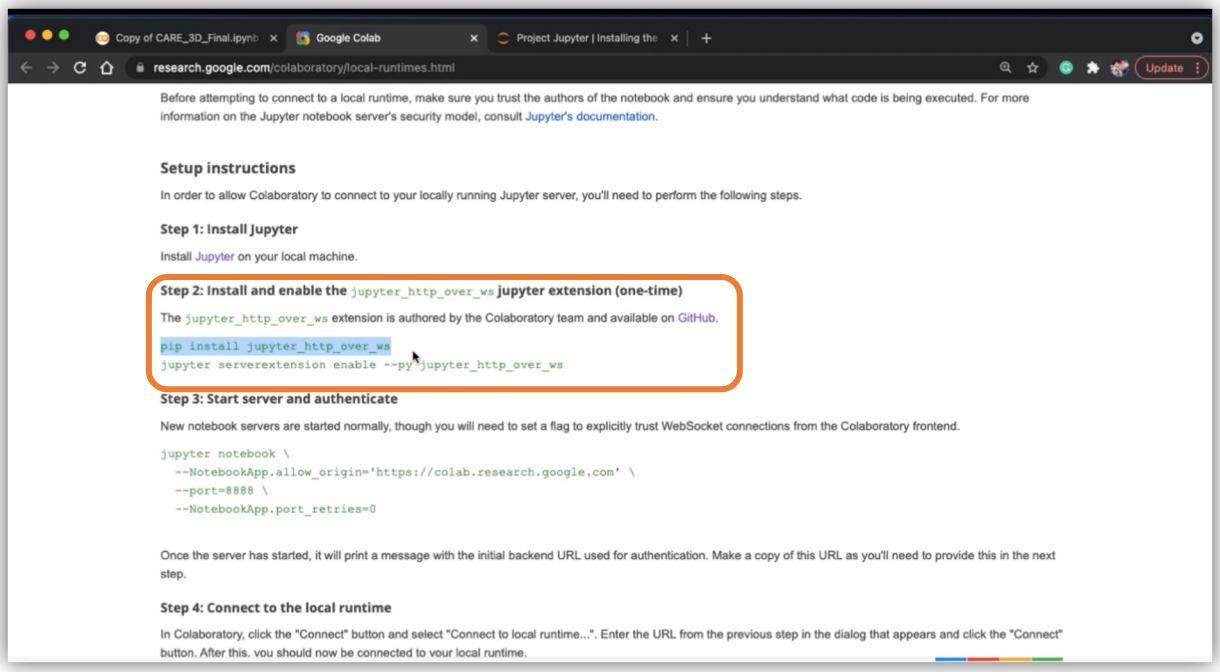

As you can see in the Jupyter website, one of the easiest ways to install it is using pip function.

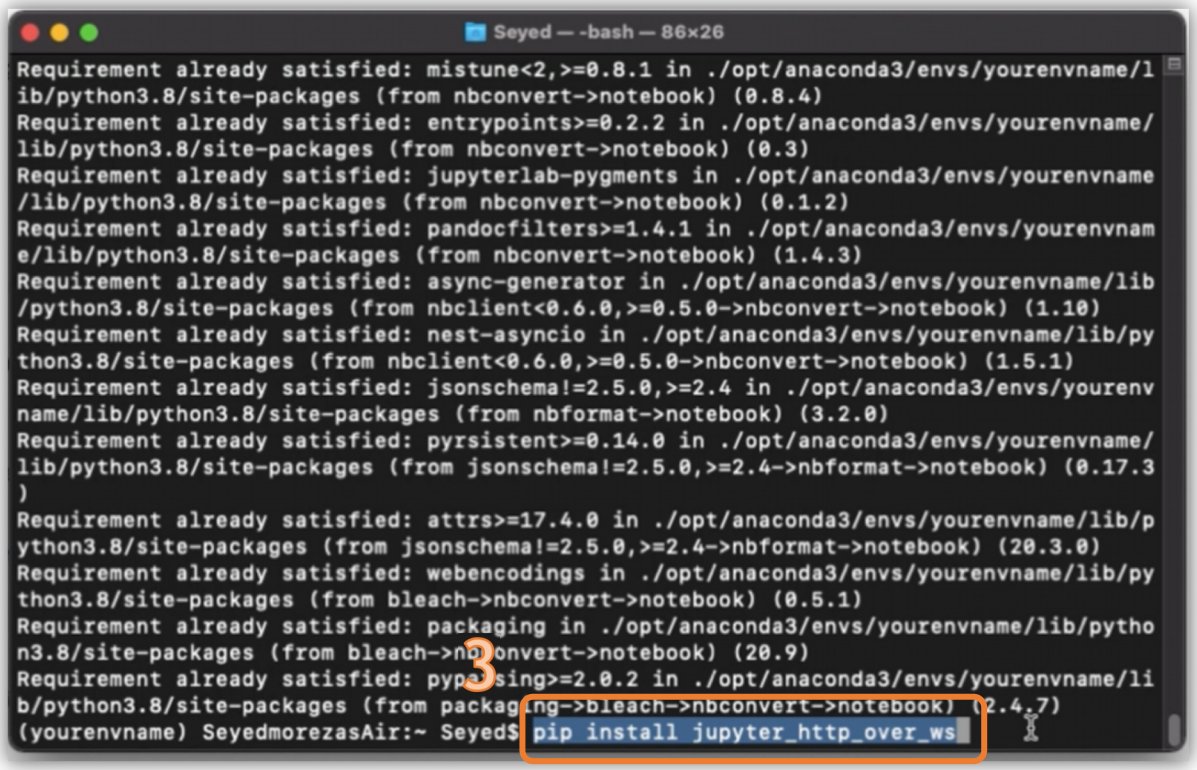
```
pip install notebook
```



Next, install the extension and enable it using the following function:

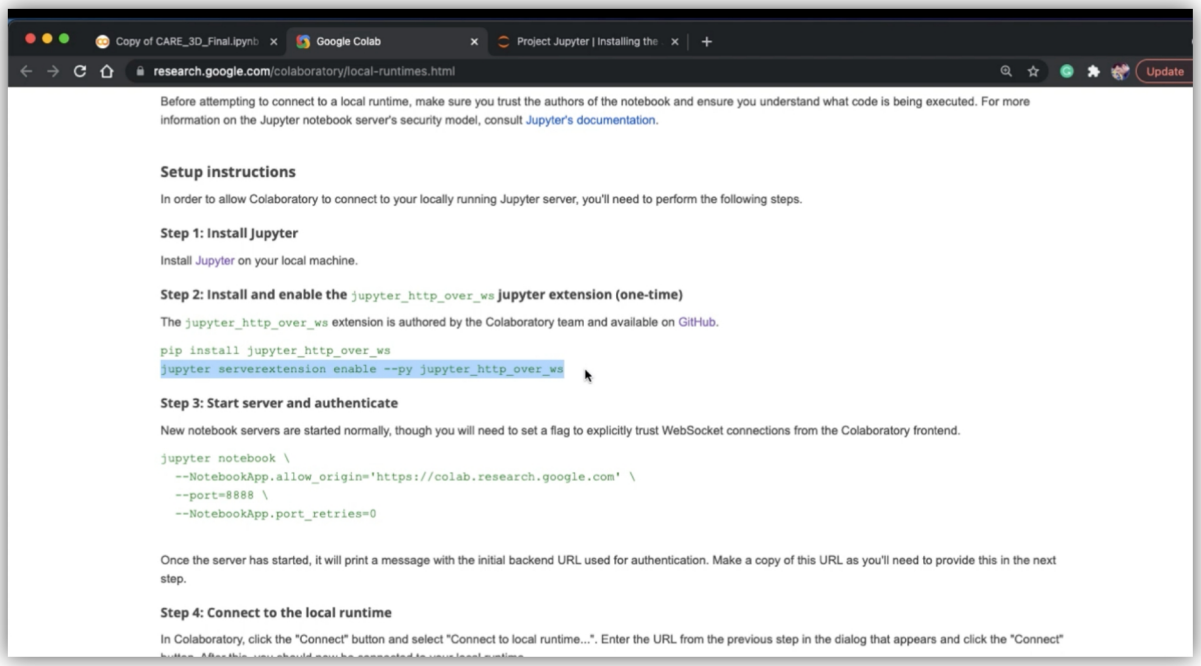
```
pip install jupyter_http_over_ws
```

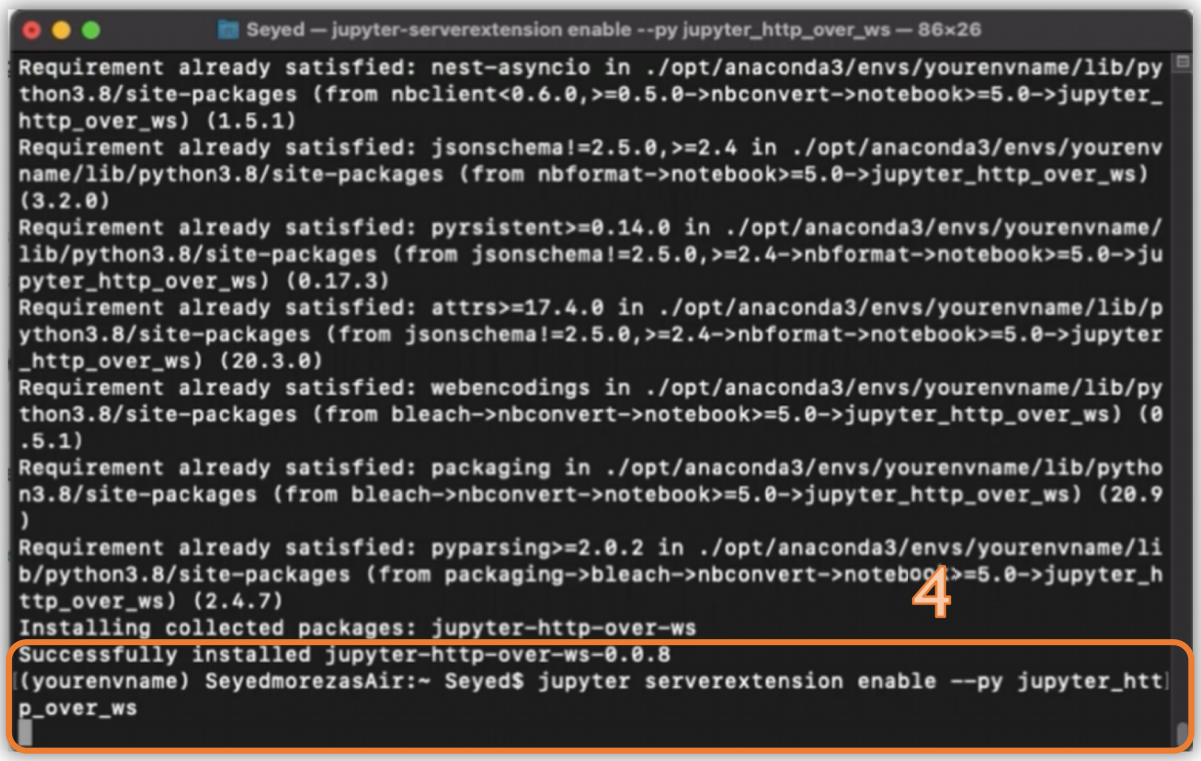


Now you should be able to enable the extension using this function:

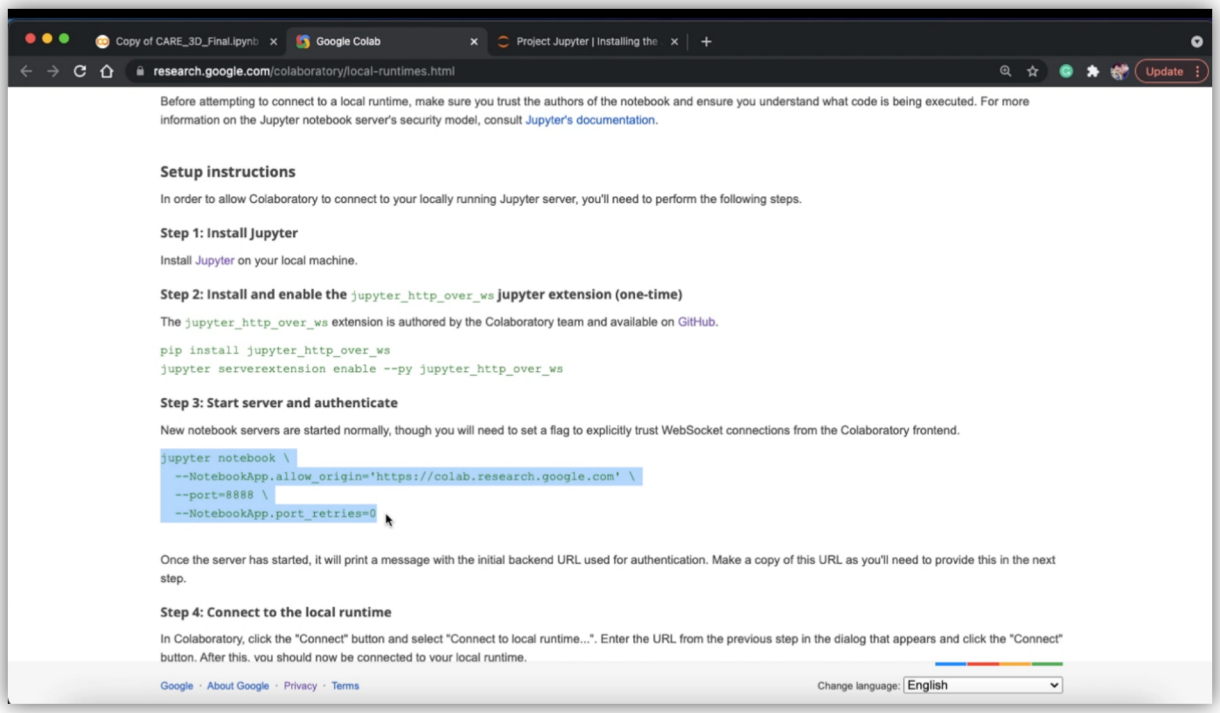
```
jupyter serverextension enable --py jupyter_http_over_ws
```

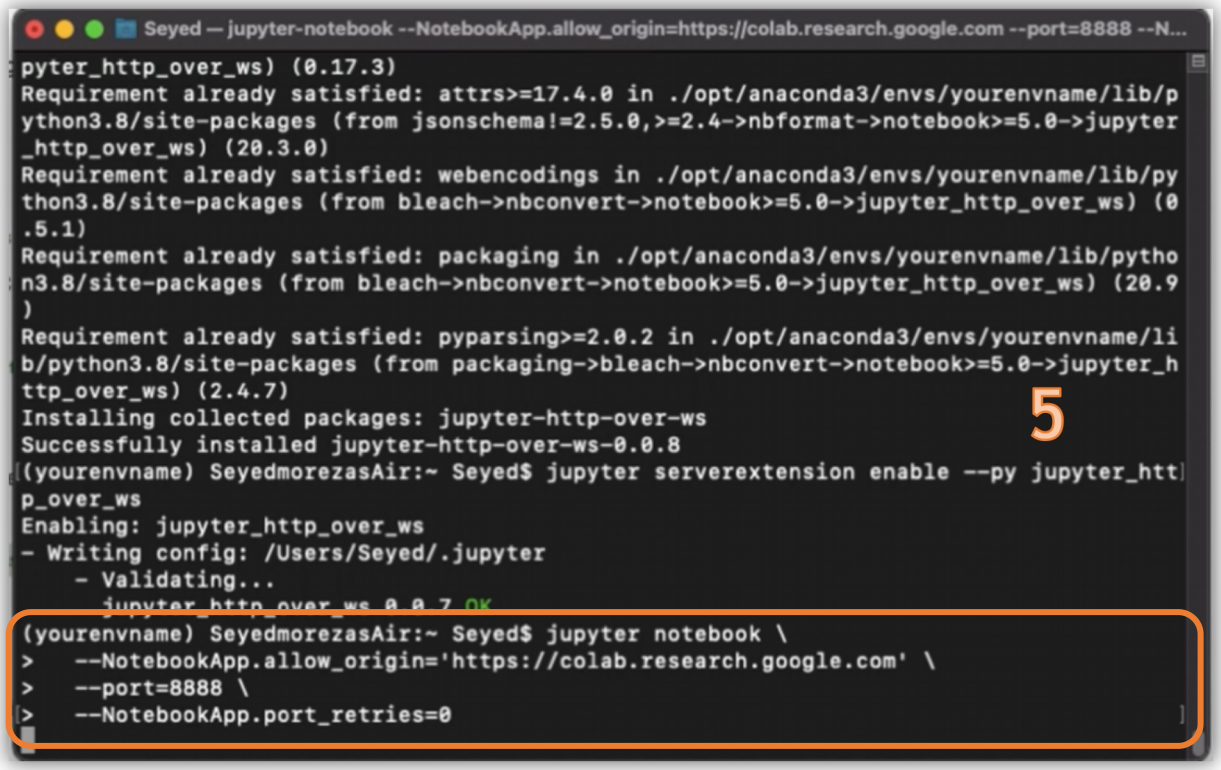


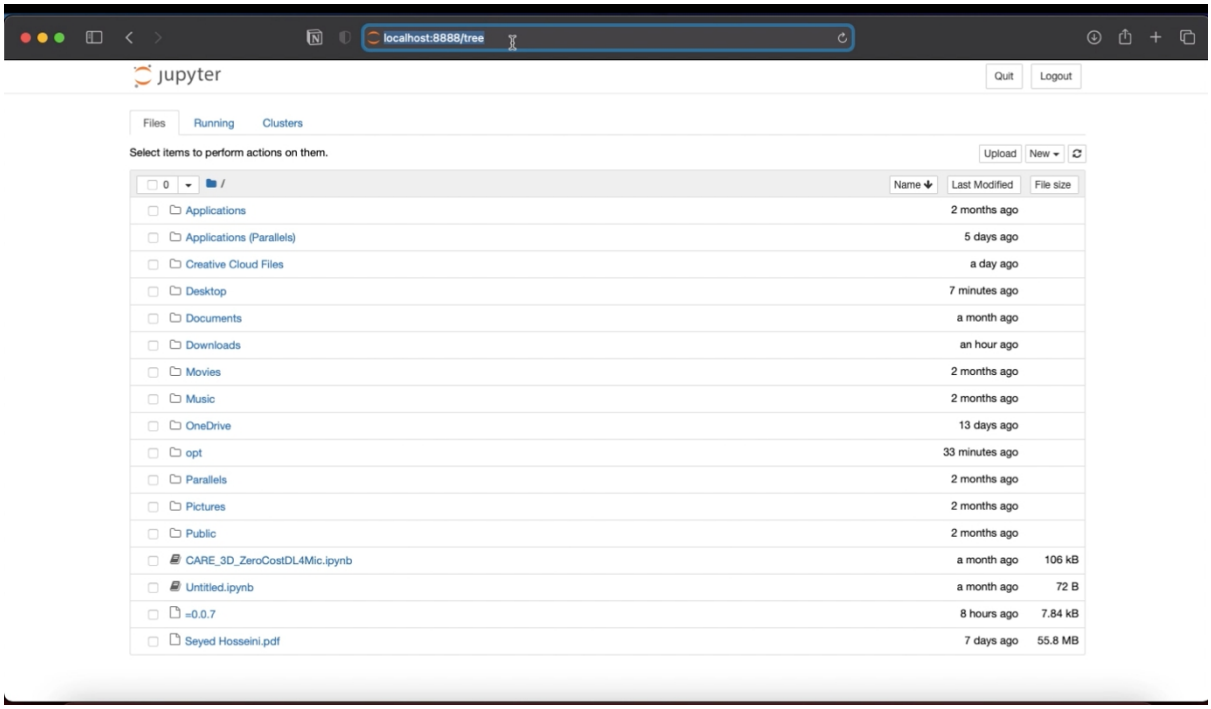
Now that you have your Jupyter notebook server ready, your last step is to start it and authenticate it using the following function:

```
jupyter notebook \
  --NotebookApp.allow_origin='https://colab.research.google.com'
\
  --port=8888 \
  --NotebookApp.port_retries=0
```

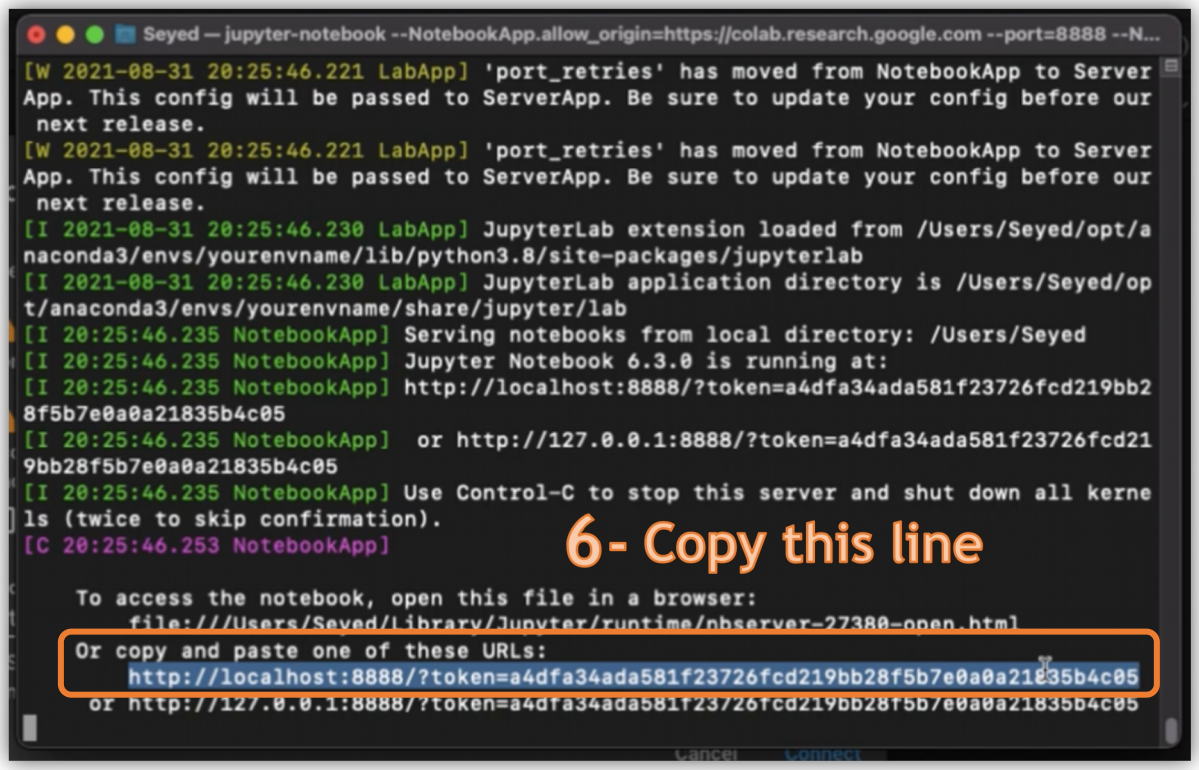


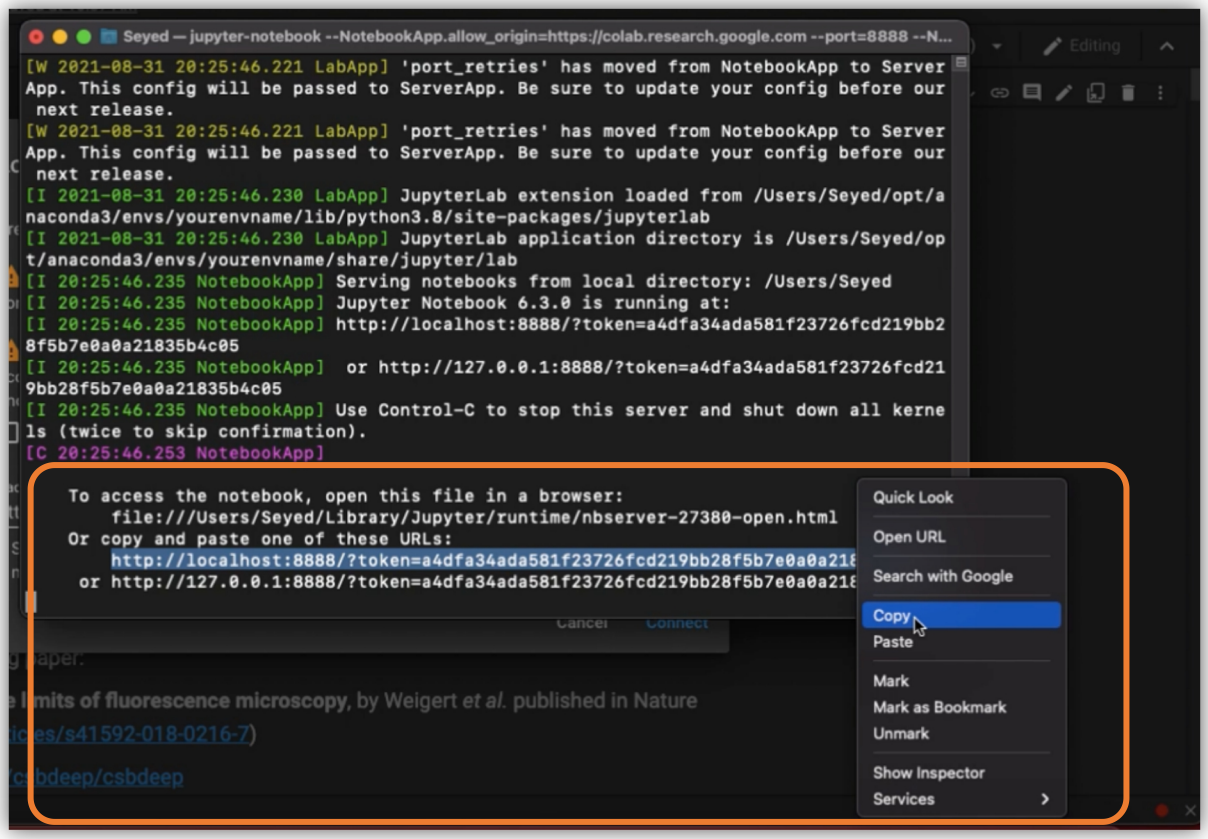



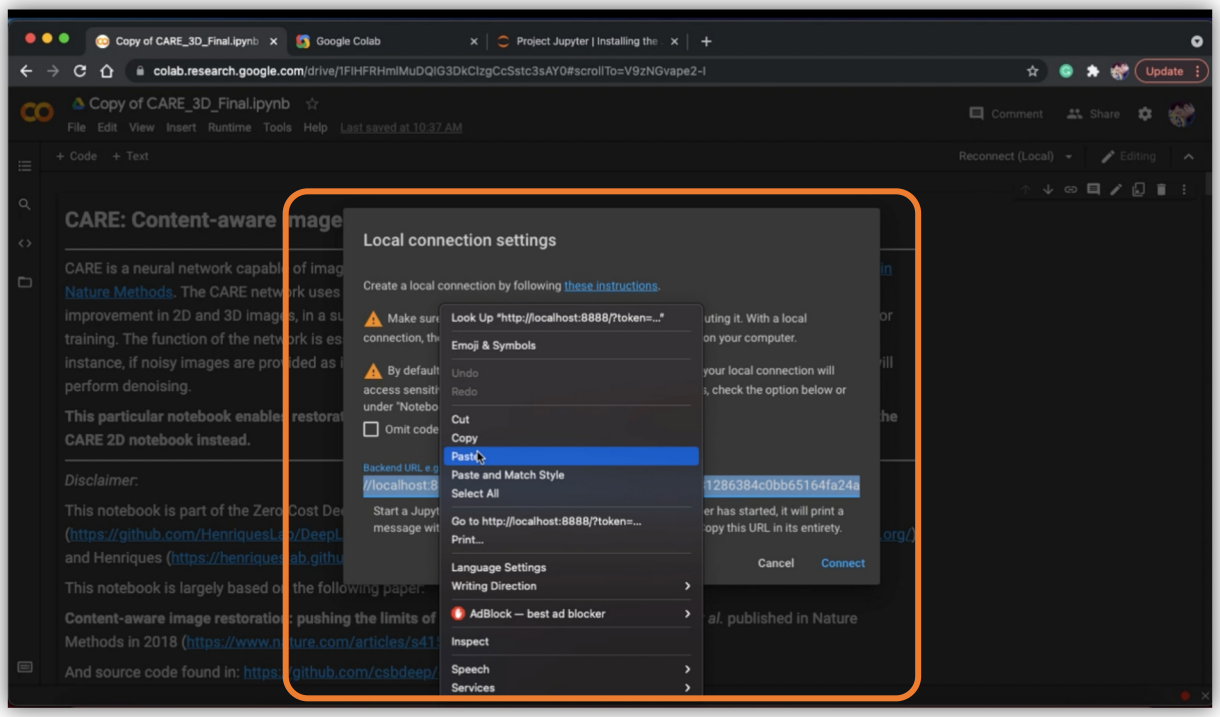
After starting the server it will automatically open a page like this on your browser. This page can be closed and is not needed during the next steps.




Coming back to your Terminal client, you should see your server address which should be **copied** into the pop-up window of Google Colab.




Seyed — jupyter-notebook --NotebookApp.allow_origin=https://colab.research.google.com --port=8888 --N...
[W 2021-08-31 20:25:46.221 LabApp] 'port_retries' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[W 2021-08-31 20:25:46.221 LabApp] 'port_retries' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[I 2021-08-31 20:25:46.230 LabApp] JupyterLab extension loaded from /Users/Seyed/opt/anaconda3/envs/yourenvname/lib/python3.8/site-packages/jupyterlab
[I 2021-08-31 20:25:46.230 LabApp] JupyterLab application directory is /Users/Seyed/opt/anaconda3/envs/yourenvname/share/jupyter/lab
[I 20:25:46.235 NotebookApp] Serving notebooks from local directory: /Users/Seyed
[I 20:25:46.235 NotebookApp] Jupyter Notebook 6.3.0 is running at:
[I 20:25:46.235 NotebookApp] http://localhost:8888/?token=a4dfa34ada581f23726fcd219bb28f5b7e0a0a21835b4c05
[I 20:25:46.235 NotebookApp] or http://127.0.0.1:8888/?token=a4dfa34ada581f23726fcd219bb28f5b7e0a0a21835b4c05
[I 20:25:46.235 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 20:25:46.253 NotebookApp]
To access the notebook, open this file in a browser:
file:///Users/Seyed/Library/Jupyter/runtime/nbserver-27380-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=a4dfa34ada581f23726fcd219bb28f5b7e0a0a218
or http://127.0.0.1:8888/?token=a4dfa34ada581f23726fcd219bb28f5b7e0a0a218
Quick Look
Open URL
Search with Google
Copy
Paste
Mark
Mark as Bookmark
Unmark
Show Inspector
Services
Editing
Cancel
Connect
mits of fluorescence microscopy, by Weigert et al. published in Nature
bdeep/csbdeep


Copy of CARE_3D_Final.ipynb
Google Colab
Project Jupyter | Installing the
colab.research.google.com/drive/1FIHFRHmlMuDQIG3DkClzgCcSstc3sAY0#scrollTo=V9zNGvape2-I
Update
Copy of CARE_3D_Final.ipynb
File Edit View Insert Runtime Tools Help Last saved at 10:37 AM
Comment
Share
+ Code + Text
Reconnect (Local)
Editing
CARE: Content-aware image
Local connection settings
Create a local connection by following these instructions.
Look Up "http://localhost:8888/?token=..."
Emoji & Symbols
Undo
Redo
Cut
Copy
Paste
Paste and Match Style
Select All
Go to http://localhost:8888/?token=...
Print...
Language Settings
Writing Direction
AdBlock — best ad blocker
Inspect
Speech
Services
Cancel
Connect
Nature Methods. The CARE network uses
improvement in 2D and 3D images, in a su
training. The function of the network is es
instance, if noisy images are provided as i
perform denoising.
This particular notebook enables restorat
CARE 2D notebook instead.
Disclaimer:
This notebook is part of the Zero-Cost De
and Henriques (https://henriqueslab.githu
This notebook is largely based on the following paper:
Content-aware image restoration: pushing the limits of
Methods in 2018 (https://www.nature.com/articles/s41
And source code found in: https://github.com/csbdeep/
al. published in Nature

Finally, you can click on "**Connect**" to have your Zerocost4DLMic connected to local runtime.