Installing Optimization Packages

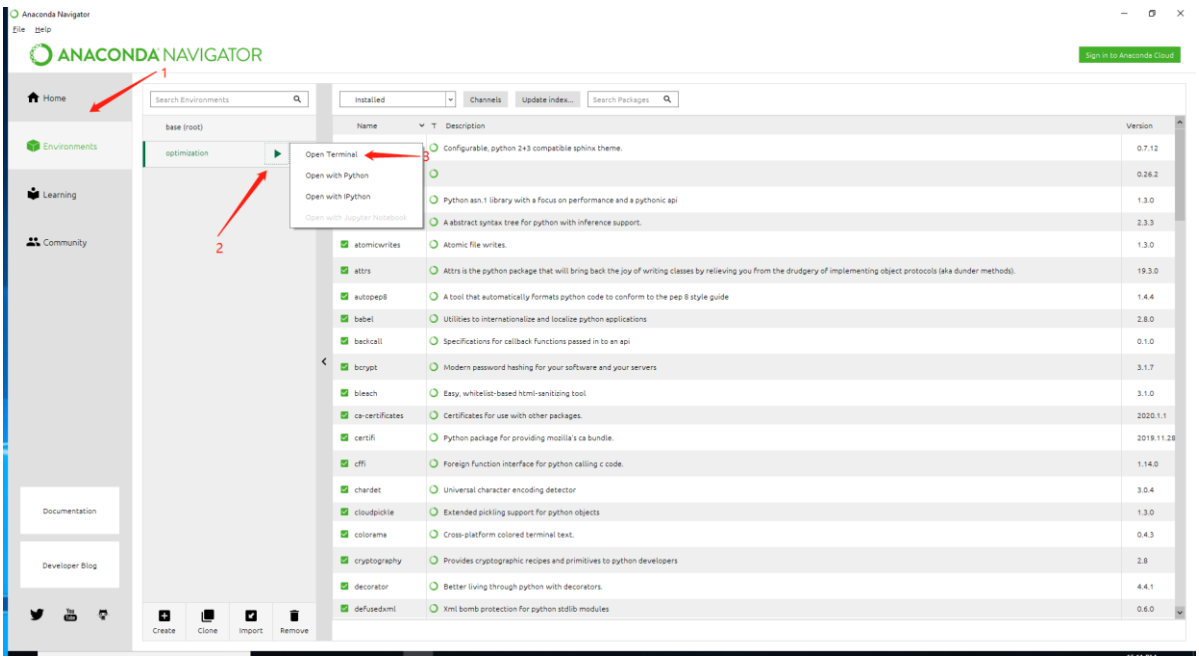
To finish the first (marked) workshop, one of the following optimization packages is required. We strongly encourage you to finish the installation in (Python) Workshop_0 and test your environment.

1. Package 1: *pyomo* (recommended) The installation instructions can be found here: https://pyomo.readthedocs.io/en/stable/installation.html
2. Package 2: You can also use *cvxpy* as an alternative for optimization problems. Here is the link: https://www.cvxpy.org/install/
3. Package 3: You can use the scipy.optimize package: https://docs.scipy.org/doc/scipy/reference/optimize.html

You can choose whichever package you like *without losing any marks*.

Here are some additional tips:

1. [Windows 10] For **cvxpy**, you may need to install the VS C++ build tools first. The instructions to install VS C++ build tools are in Appendix A.
2. To get **pyomo** work, you must install pyomo and solver packages, we recommend you use conda install for both pyomo and solver packages. Detailed instructions can be found in Appendix B.
3. The image below shows how to open the command window in *Anaconda*.

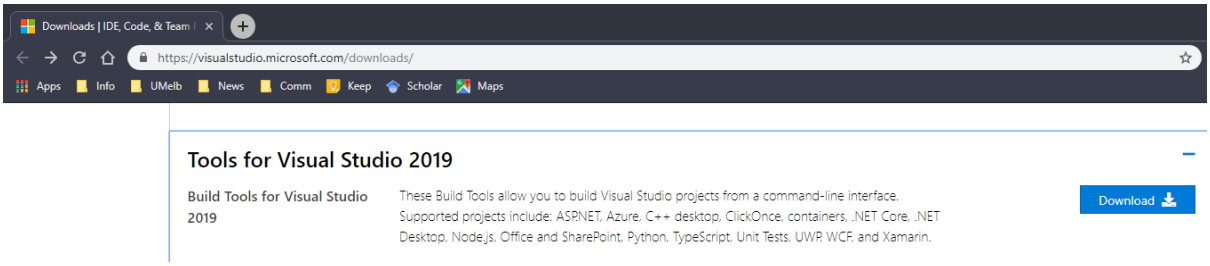


Appendix A

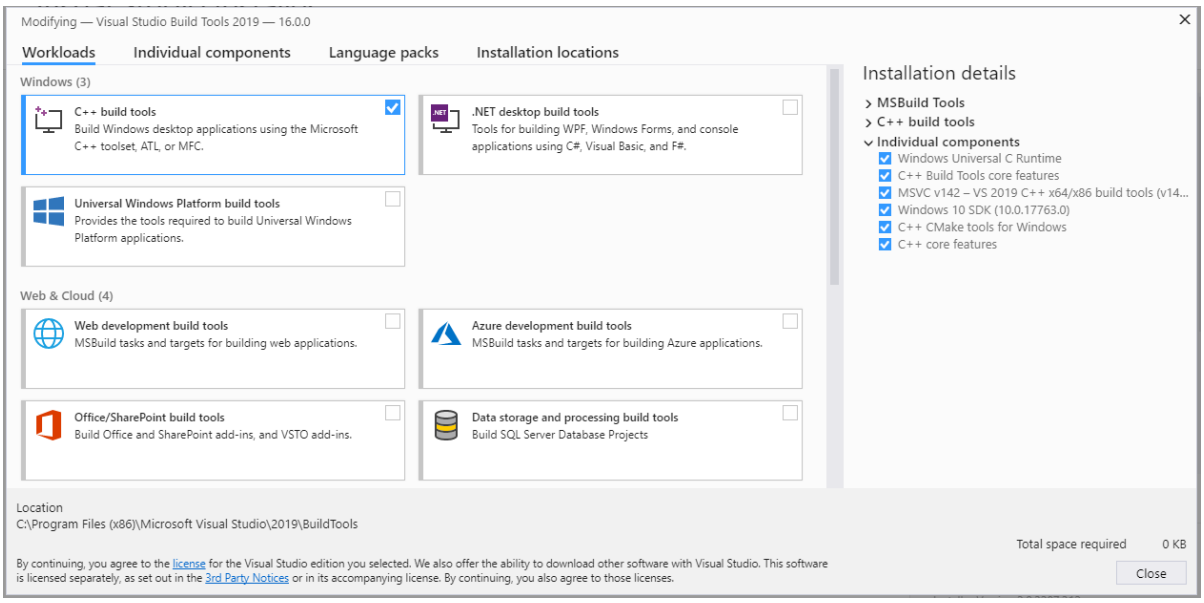
Installing Build Tools for VS 2019

If you have a **Windows 10** machine, then you should follow the instructions below to install *Build Tools for Visual Studio 2019.* These tools are sometimes required by some Python packages.
(The instruction below are probably not applicable to lab computers but may be worth trying.)

1. You need to download the *installer* for the **Build Tools for VS 2019**, see image below



2. Then, from the installer check C++ build tools. I made it work with that and individual components listed on screenshot. Individual components can be installed from the respective tab.



3. Once VS C++ build tools install on your windows machine, you can now go to the Anaconda environment and install the package that requires them.

Appendix B

For *pyomo* to solve (nonlinear optimisation) problems, you must install an optimization tool like *ipopt*. On MacOS and Linux, you can install *ipopt* in a terminal using: *conda install -c conda-forge ipopt*

2021 Installation Problem for Windows: the above *ipopt* installation is currently a bit broken under Windows. To manually install *ipopt*:

1. Locate Anaconda's *Library**bin directory* on your computer. This directory can be found using the command *where conda* in a cmd window
2. Copy the *ipopt.exe* we have provided in files (originally from pre-compiled sources), 3.11.1 64 or 32 bit version to the above *Library**bin* directory
3. Close and reopen Anaconda (and any Jupyter notebook) for changes to take effect.