Data Science Toolkit on Windows 10

Installation Instruction

1. Install Anaconda

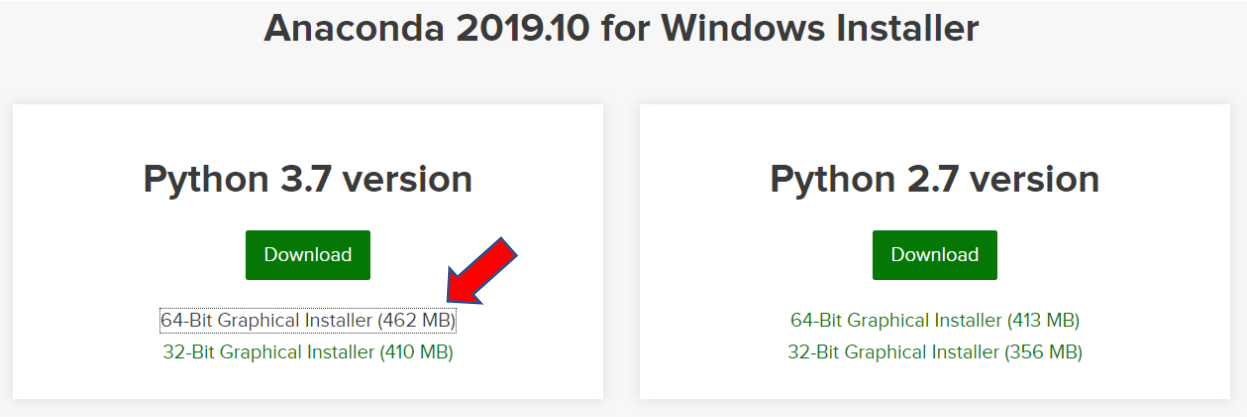
Designed for data science and machine learning workflows, Anaconda is an open-source package manager, environment manager, and distribution of the Python programming languages.

This section will guide you through installing Anaconda on a Windows 10 machine. Before installing software, you need to make sure you have administrator or other proper privileges, such as PowerBroker – Token Access Level 2.

Please Note: Uninstall Anaconda if you found your machine has an old version Anaconda. If you want to keep your existing Anaconda, you can ignore the first step of "Download and Install Anaconda" to avoid any conflictions.

Download and Install the Anaconda

From a web browser, go to the Anaconda Distribution page via the following link: https://www.anaconda.com/distribution/ to find the latest Windows version and download 64-bit Graphical Installer (assuming you have 64-bit computer).



After downloading, locate "**Anaconda3-2019.10-Windows-x86_64.exe**" executable file and double-click to open it. And then Follow the instructions on the screen. If you are unsure about any setting, accept the defaults. You can change them later. ...

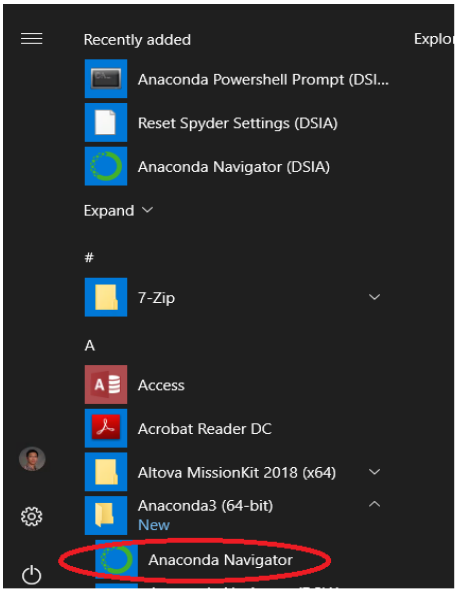
Note, the installation of Anaconda will require several GB sizes of storage. So, please make sure your laptop or VDI has sufficient disk free space.

Import Environment with an yml file

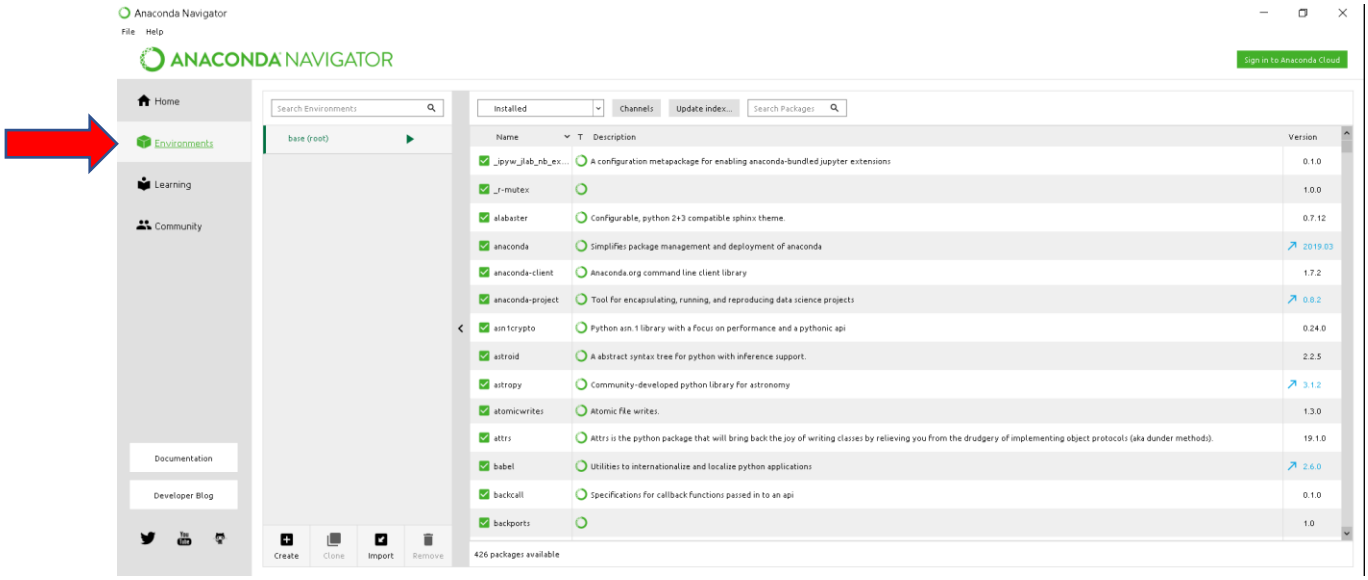
After the base Anaconda installed, you can import DSIA (Anaconda) environment to install additional packages by following the steps below.

Step 1: Download Anaconda environment yml file from my Github current folder and save it to the folder of "Downloads" at your laptop or VDI.

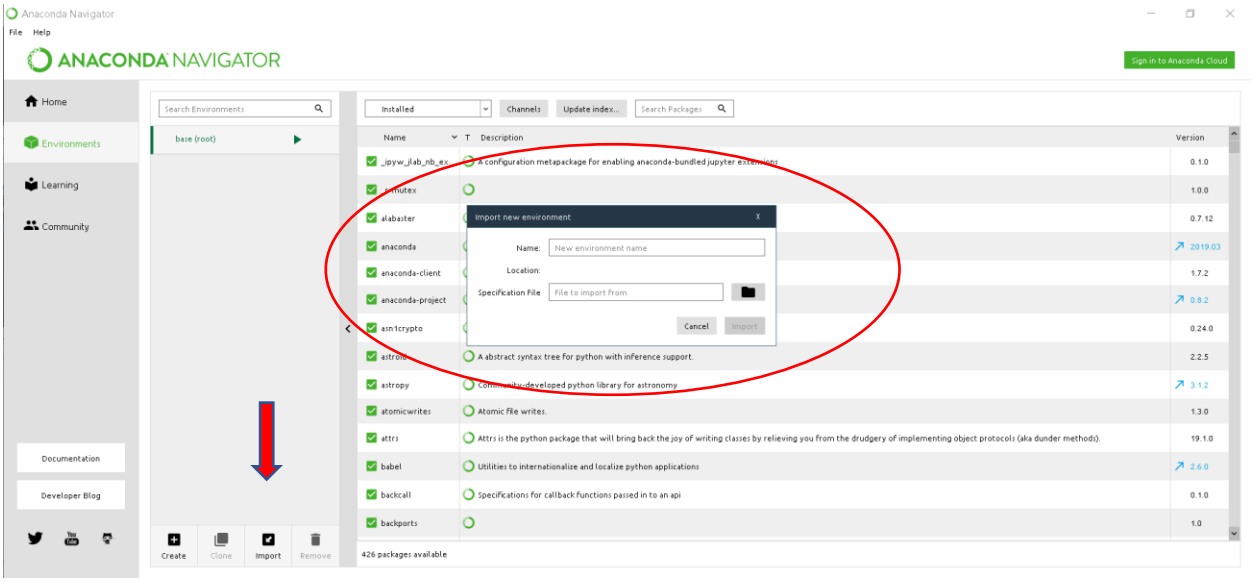
Step 2: Open Anaconda Navigator by clicking Start → Anaconda3 (64 bit) → Anaconda Navigator



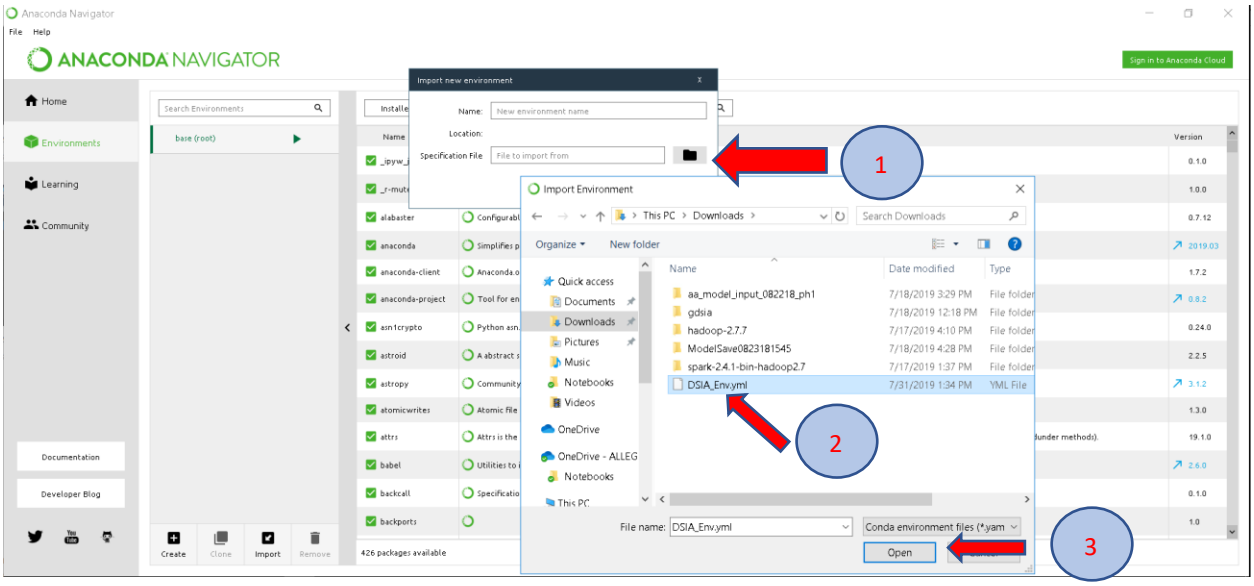
Step 3: Go to "Environments" tab on the left side by clicking it as shown below.



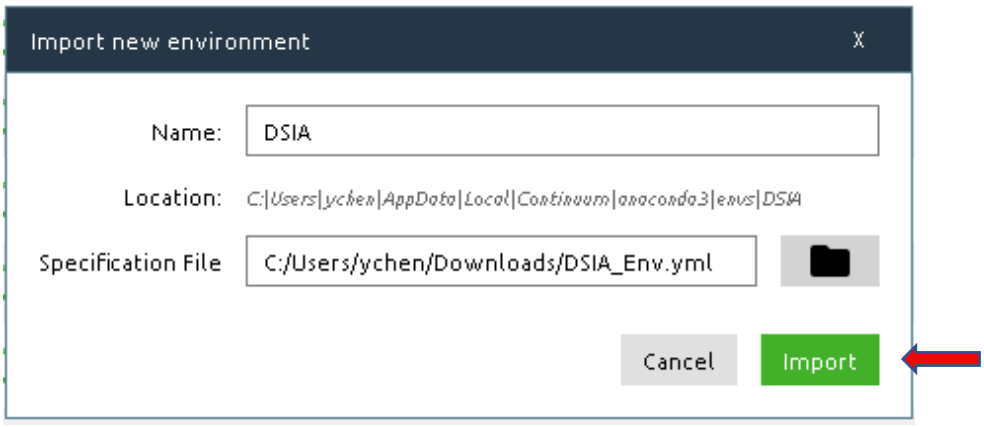
Step 4: Click “Import” at the bottom to open a prompt window to allow you to pick yml file.



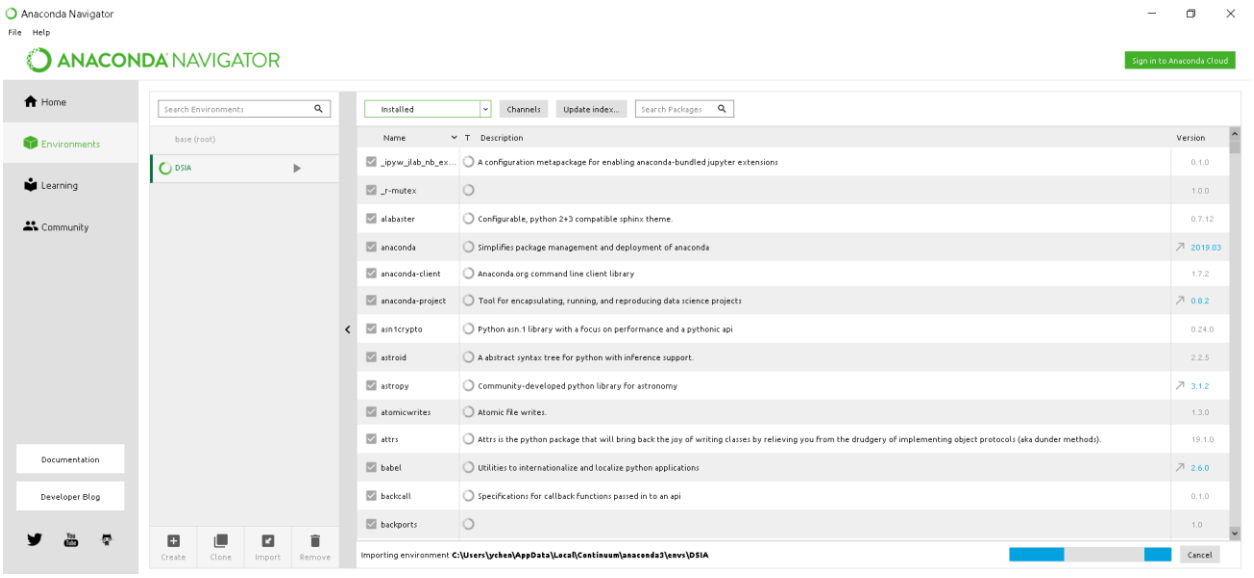
Step 5: Click folder icon on the prompt window, navigate the folder of “Downloads” and pick the yml file. And then click “Open” button.



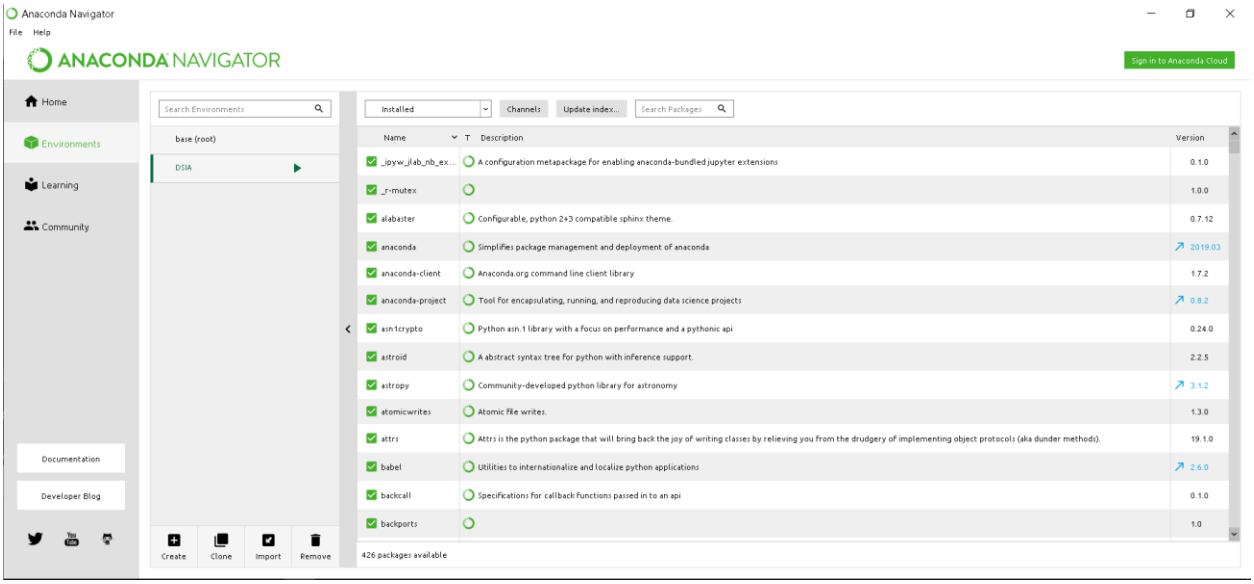
After picking the right yml file, you should see the prompt window with proper information filled, like the following screen shot.



Step 6: Click “Import” (green color) button to start the import process. You should see the importing progress bar like the following screen shot. The import process will take a while.



After import process finishes, your Anaconda environment should look like the following screen shot.



Close the Anaconda Navigator when you are done the importing.

Test Whole Installation

Use the `conda` command to test the installation and activation: For example,

```
C:\>conda config --set ssl_verify false

C:\>conda list

C:\>pip list
```

You'll receive output of all the packages you have available through the Anaconda and pip installation.

Update Installation

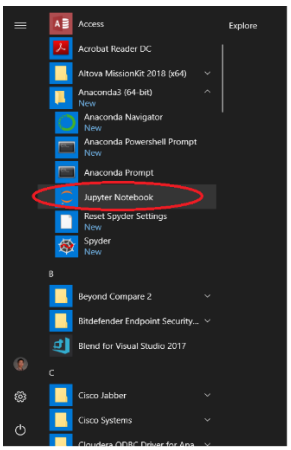
You can easily update Anaconda to the latest version.

```
C:\Users\ychen\anaconda3\bin>conda update --all --yes
```

Activate Jupyter Notebook Extensions

```
c:\>jupyter contrib nbextension install –user
```

Start Jupyter Notebook. You can launch Jupyter Notebook from **Start→Anaconda3 (64-bit)→Jupyter Notebook (DSIA)** like following screen shot:

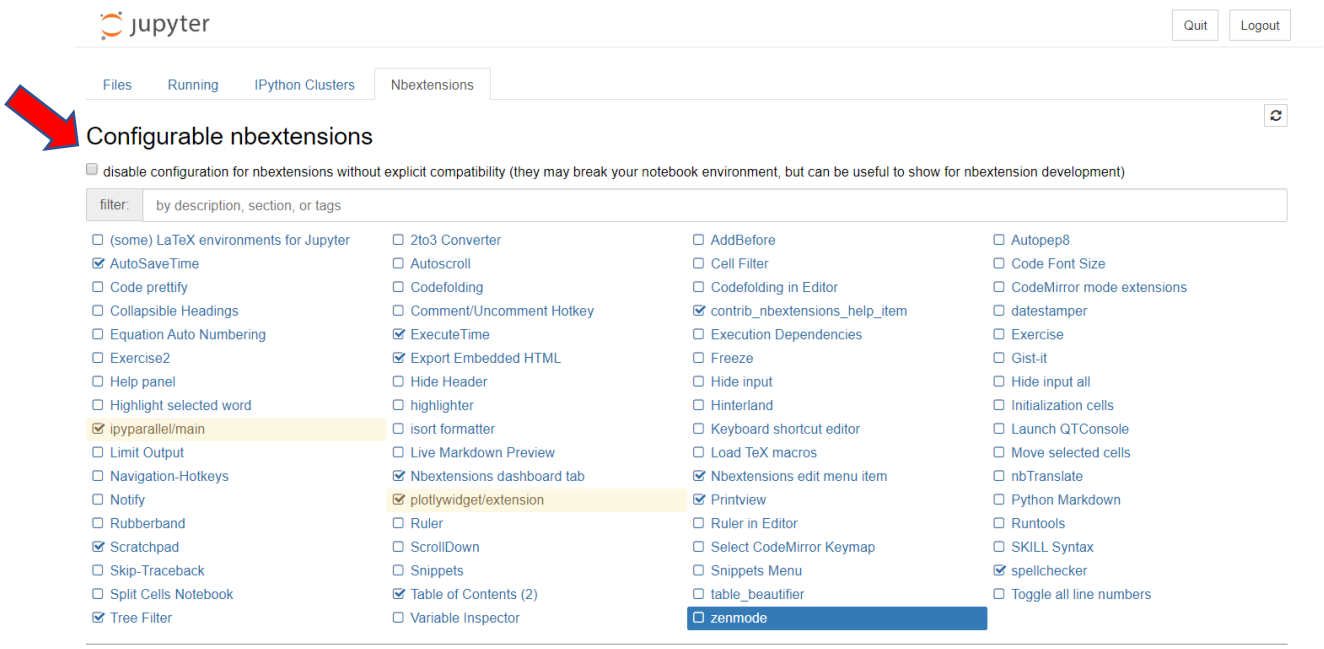


Or type the command from **cmd** windows prompt to start Jupyter Notebook:

```
C:\>jupyter notebook
```

If Jupyter notebook cannot be open in a web browser, you can copy the server url (e.g. http://localhost:8888/?token=0fa79120798f795452ebc143ce11d7f1f8ac73e5f860d551) and paste it on a open web browser, such as google chrome.

After open Jupyter Notebook, click the tab of "**Nbextensions**" to see the configurable nbextensions as shown below:

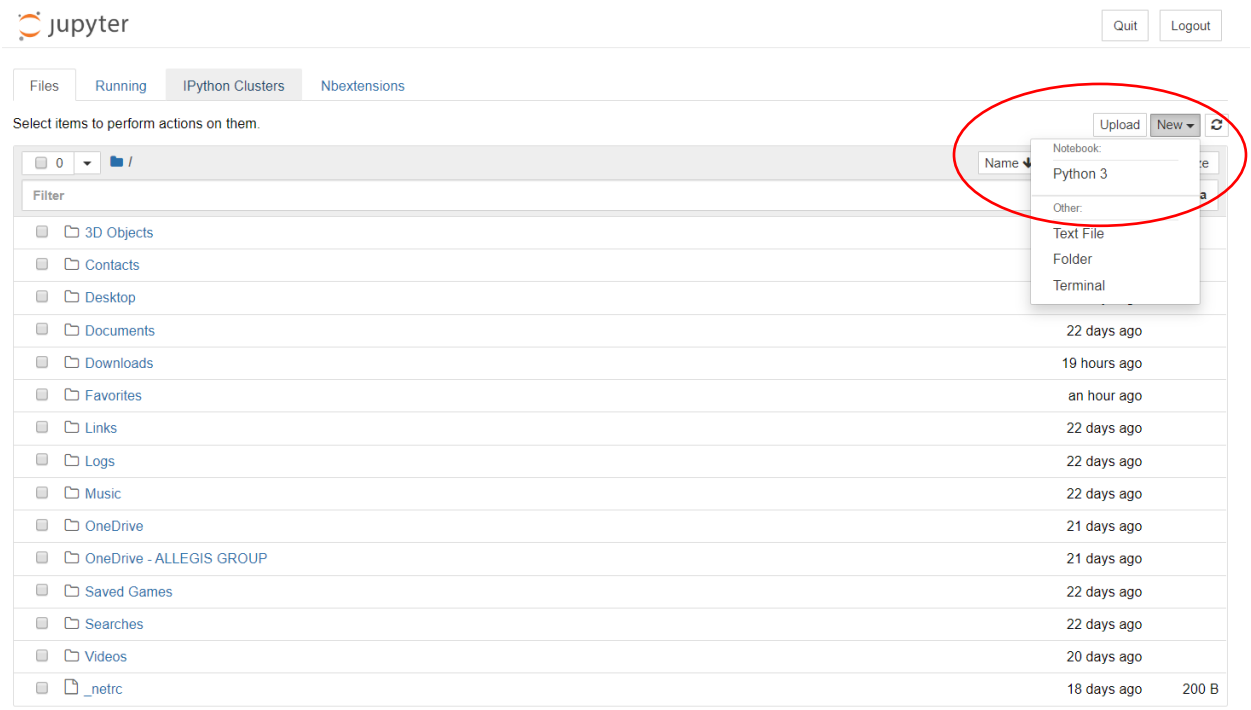


Unselect "disable configuration for nbextionsions without explicit…" to active the extension configuration and then choose the extensions you would like to use. The recommended list of extensions you should use at least includes:

- **ExecuteTime**: Display when each cell has been executed and how long it took
- **Export Embedded HTML**: Export to HTML with images embedded
- **plotlywidget/extension:** This nbextension is enabled in the notebook json config, but doesn't provide a yaml file to tell us how to configure it. You can still enable or disable it from here, though.
- **Table of Contents (2):** The toc2 extension enables to collect all running headers and display them in a floating window, as a sidebar or with a navigation menu. The extension is also draggable, resizable, collapsable, dockable and features automatic numerotation with unique links ids, and an optional toc cell.
- **Spellchecker:** Adds a CodeMirror overlay mode for Typo.js spellchecking.

- **Help panel:** Add a toolbar button to display a help panel showing shortcuts to the right side of the notebook.

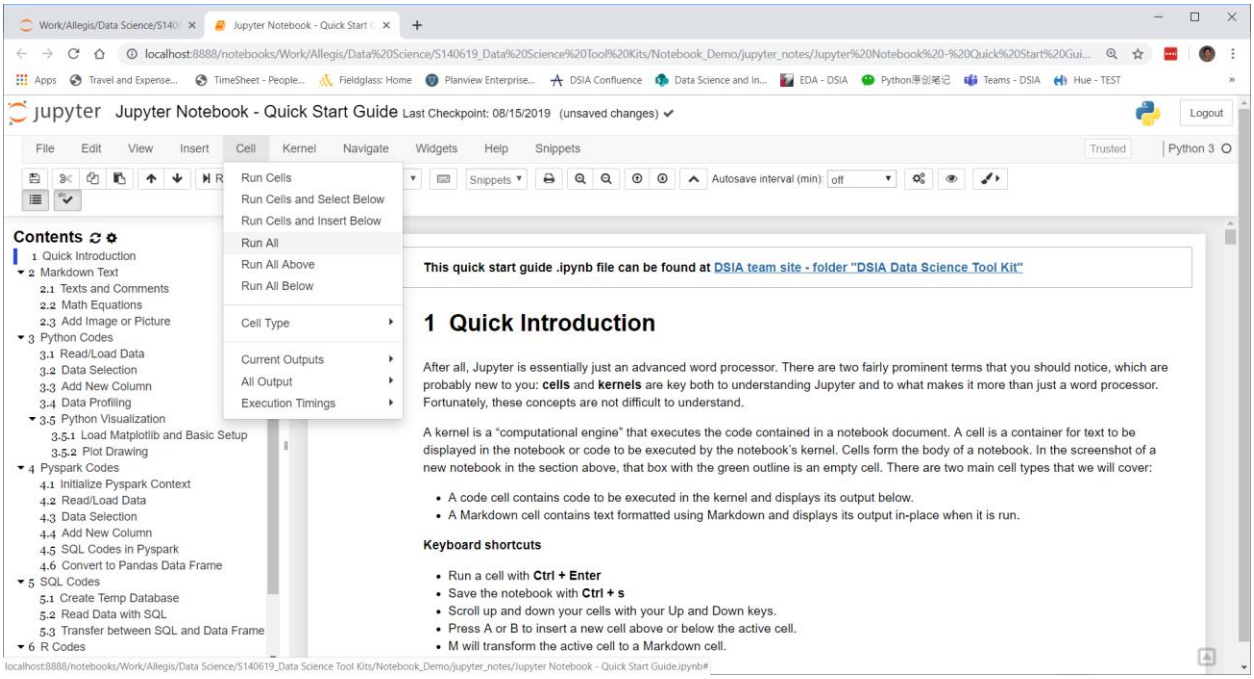
To create/open a new Python 3 notebook, you can click menu **New→Python3** like the following screen shots.



2.Run Quick Start Guide (Optional)

After all the steps above have been properly performed, you can run a jupyter notebook, called Jupyter Notebook – Quick Start Guide, to test and explore what you can do with this data science environment.

The jupyter notebook .ipynb file of the quick start guide can be downloaded from the Team site here. After it is downloaded, open jupyter notebook application from your start menu, navigate to the folder you have downloaded the ipynb file, and then click it to open. When this jupyter notebook is open in a separate tab, you can run all the contents by clicking **Cell → Runn All** menu item (see the following screen shot).

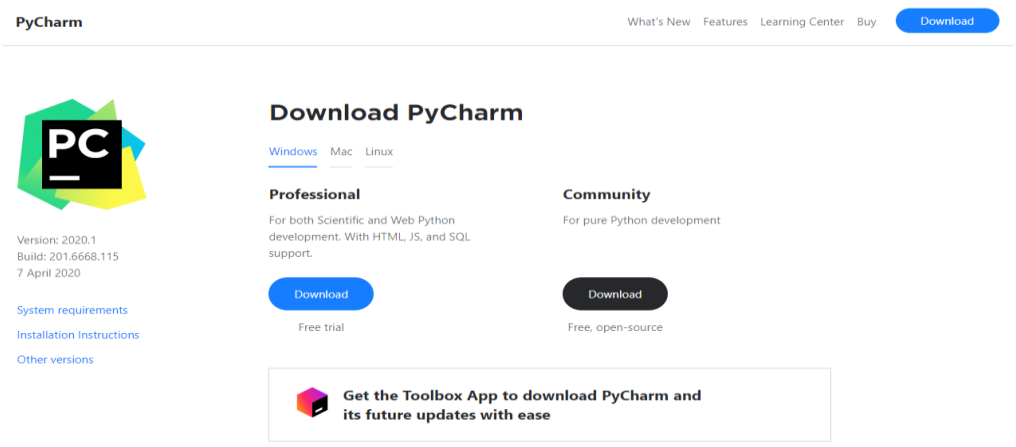


Please notify Data Science team or manager if you see any errors when running this jupyter notebook for trouble shooting. Otherwise, you can play this jupyter notebook a little bit to learn how you are able to perform data science and data engineering work with jupyter notebook and the environment. Enjoy! 😊

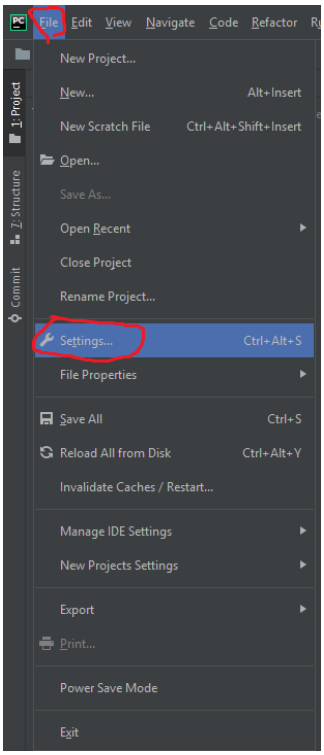
3. Install Python Development IDE

Two popular IDEs for python project development are Pycharm and Visual Studio Code. Pycharm is recommended as IDE for the team due to its simple installation and configuration processes. The following provides step by step instruction to install and config Pycharm.

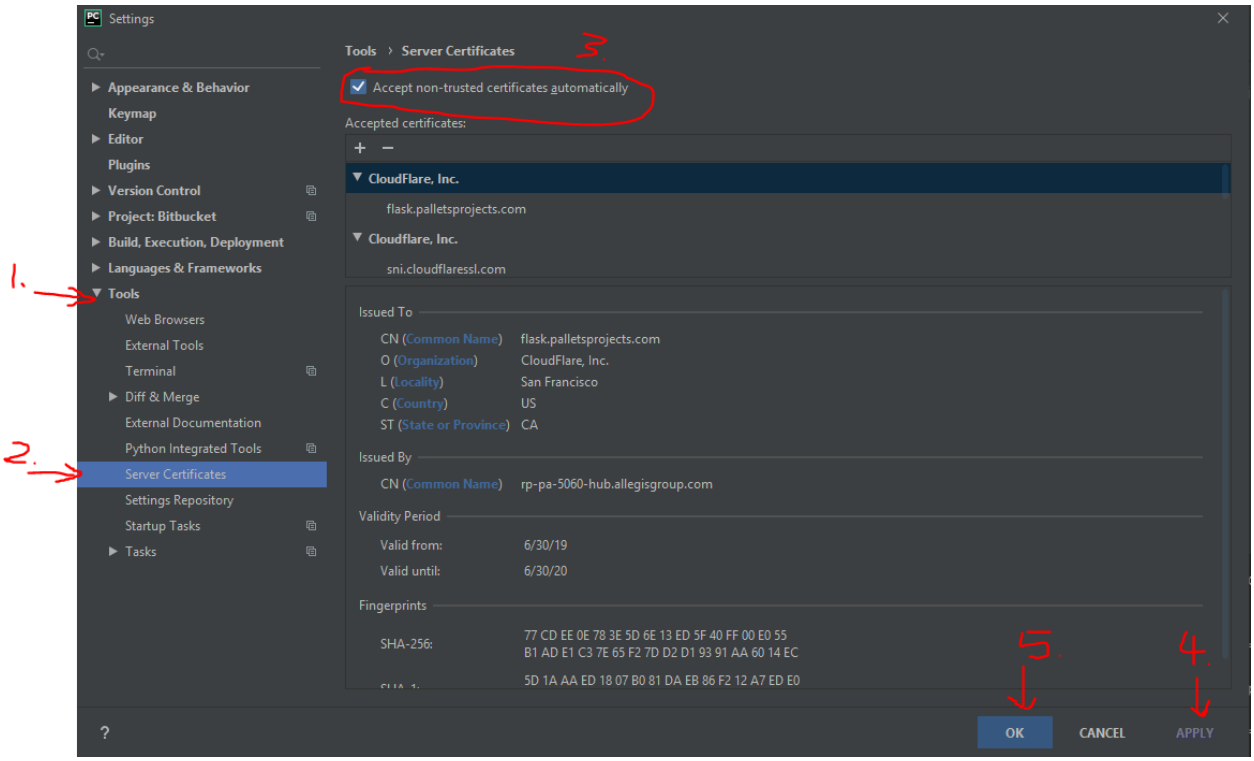
Step 1: Go to the website https://www.jetbrains.com/pycharm/download/#section=windows to download and install proper (professional if you have license) edition as below screen shot.



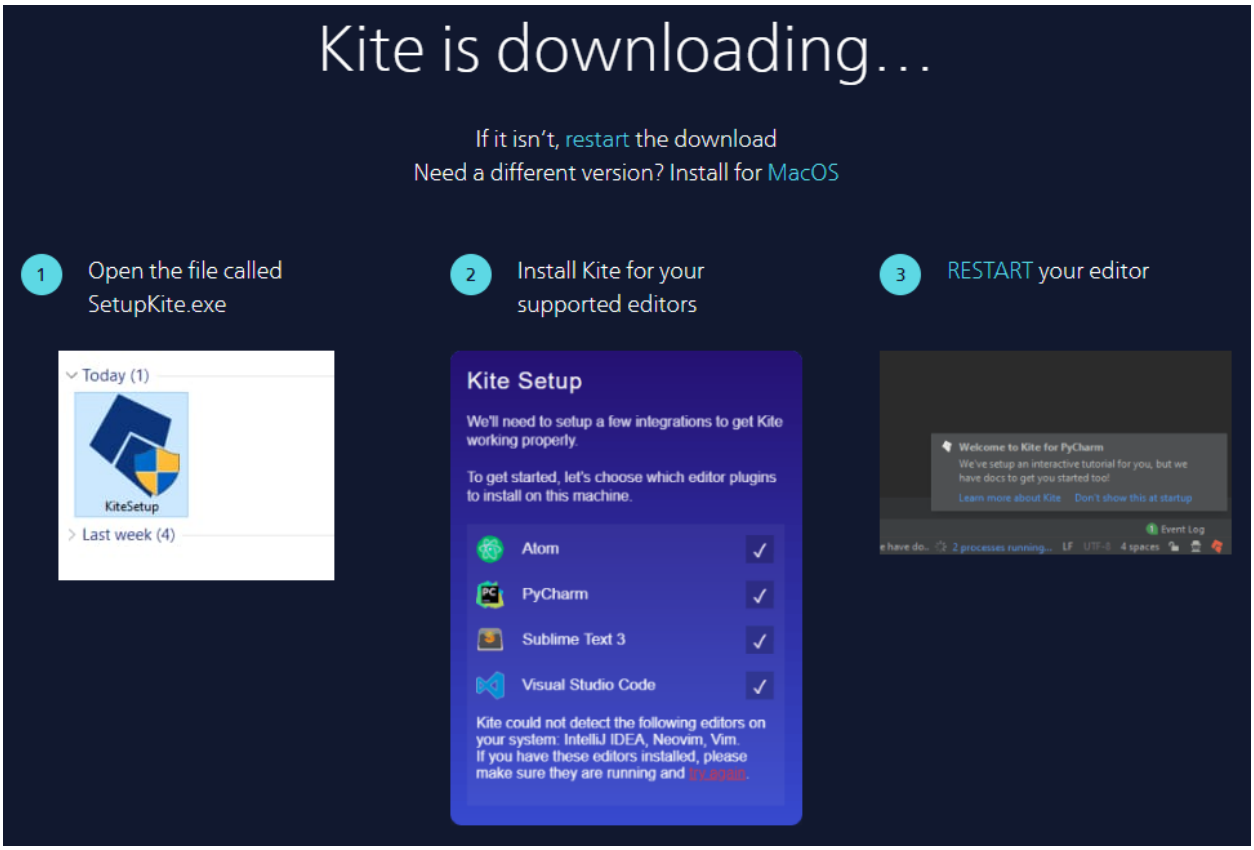
Step 2: After success installation, open Pycharm and go to the menu “File” → “Settings”



And follow the screen shot below to config “server certificates” to prevent unnecessary alarm window popup.



Step 3 (optional): Recommend installing the Kite plugin, a popular AI powered Python code autocompletion tool, on Pycharm to enable a better code autocomplete feature. The installation steps are very simple. Go to the website https://kite.com/ to download the executable file "SetupKite.exe". After downloading the Kite setup file, please follow the 3-step instruction shown as its webpage (see screen shot below):



Step 4 (optional): Use the link of "https://www.jetbrains.com/help/pycharm/quick-start-guide.html" to go through a quick start guide to understand how to use Pycharm to do Python development.

4. Appendix – Install R Packages

Install R Kernel and R Packages

To use R in an anaconda environment, all you need to do is to install the r-essentials bundle, which includes over 80 of the most popular scientific R packages.

```
C:\>conda install r-essentials
```

```
C:\>conda install r-irkernel
```

The R language packages are available to install with conda at http://repo.anaconda.com/pkgs/r/. You can install any of these R language packages into your current environment with the conda command

```
C:\>conda install -c r package-name
```

For more information, you can check this link.