

# 07. Installing Python

Monday, November 6, 2023

3:01 PM

## INSTALLATION CHECKLIST

- **Anaconda Distro (free Version)**

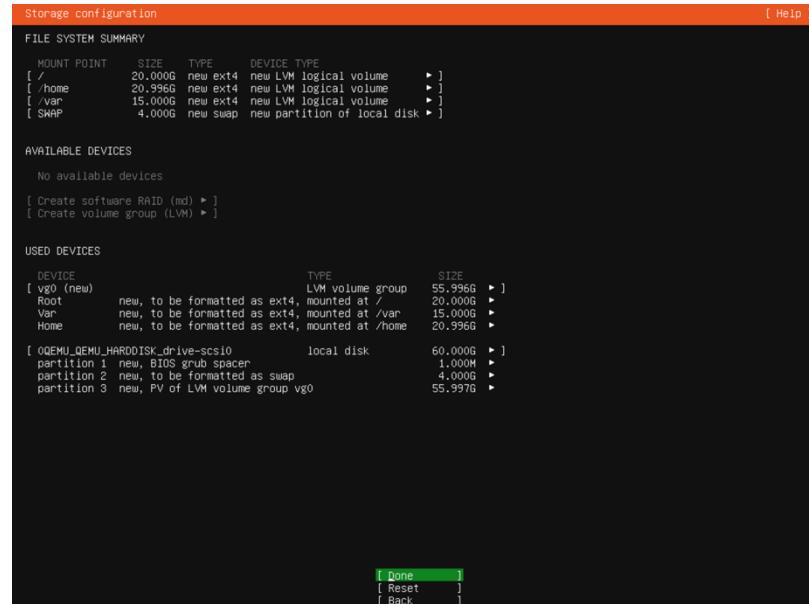


**ANACONDA®**

- **URL Main = <https://www.anaconda.com/download>**
- **URL Installation Guide = <https://docs.anaconda.com/free/anaconda/install/>**

- **Hardware Minimum Requirements**

- **Cpu :** 2 Core + 2 Hyperthreaded with 2.8Ghz
- **RAM :** 14GB 1600mhz DDR3 WINDOWS or 8GB 1600mhz DDR3 LINUX
  - 8 GB for Windows OS or 2 GB for Linux Server
  - 2 GB per person
  - 4 GB for MongoDB
- **STORAGE :** 100 GB for Windows OS & 60 GB for Linux Ubuntu server
  - 64 GB for Windows
  - 20 GB Linux Ubuntu Server      Linux : / (Root)
  - 2gb for Anaconda3 distro      Linux : 4gb /swap
  - 3.5GB for Ananconda3 Standard Install      Linux : 16gb /Home
  - 10 GB Free Disk Space MongoDB      Linux : /Var
  - 5 GB Database Storage      Linux : /Var



- **Ref URL : See below**

- <https://docs.anaconda.com/anaconda-repository/admin-guide/install/requirements/#directive-4>
- <https://docs.anaconda.com/ae-notebooks/system-requirements/>
- <https://www.mongodb.com/docs/manual/tutorial/install-mongodb-enterprise-on-ubuntu/#std-label-install-mdb-enterprise-ubuntu>
- <https://www.mongodb.com/docs/manual/administration/production-notes/#std-label-prod-notes-ram>
- [https://documentation.commvault.com/v11/essential/system\\_requirements\\_for\\_mongodb.html](https://documentation.commvault.com/v11/essential/system_requirements_for_mongodb.html)
- [https://www.reddit.com/r/learnpython/comments/d7d0kn/anaconda\\_disk\\_storage\\_requirements/](https://www.reddit.com/r/learnpython/comments/d7d0kn/anaconda_disk_storage_requirements/)

- **Anaconda Distro contain list**

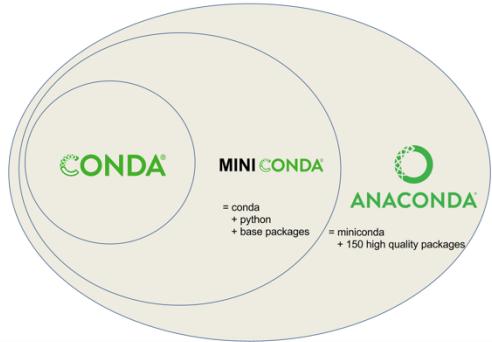
- Anaconda Navigator

*Note : It is a Graphical applications, packages, and environment management system.*

- Python
- Jupiter Notebook
- Bunch of other useful tools

## INFORMATION FOR REFERENCES

### Differences: Conda, Miniconda and Anaconda.



### Key insights

#### 1. What is Conda?



- Conda get install in computer with Miniconda Installer.
- Conda use `conda install` command to install packages available by default in Anaconda's public repository or from other channels, like conda-forge or bioconda.
- Conda is open source package and environment management system that runs on Windows, Mac OS and Linux.
- Conda can quickly find, then install, run, and update packages and their dependencies.
- Conda can create, save, load, and switch between project specific software environments.
- Conda can package and distribute software for any language such as R, Ruby, Lua, Scala, Java, JavaScript, C, C++, FORTRAN
- Conda was created to provide greater control over project environments, making it recommended choice for Advance Python programmer user.
- Conda is command line program and it works on the Linux Terminal, Windows Anaconda Prompt or Windows Anaconda PowerShell Prompt.
- NO GUI

#### 2. What is Miniconda?

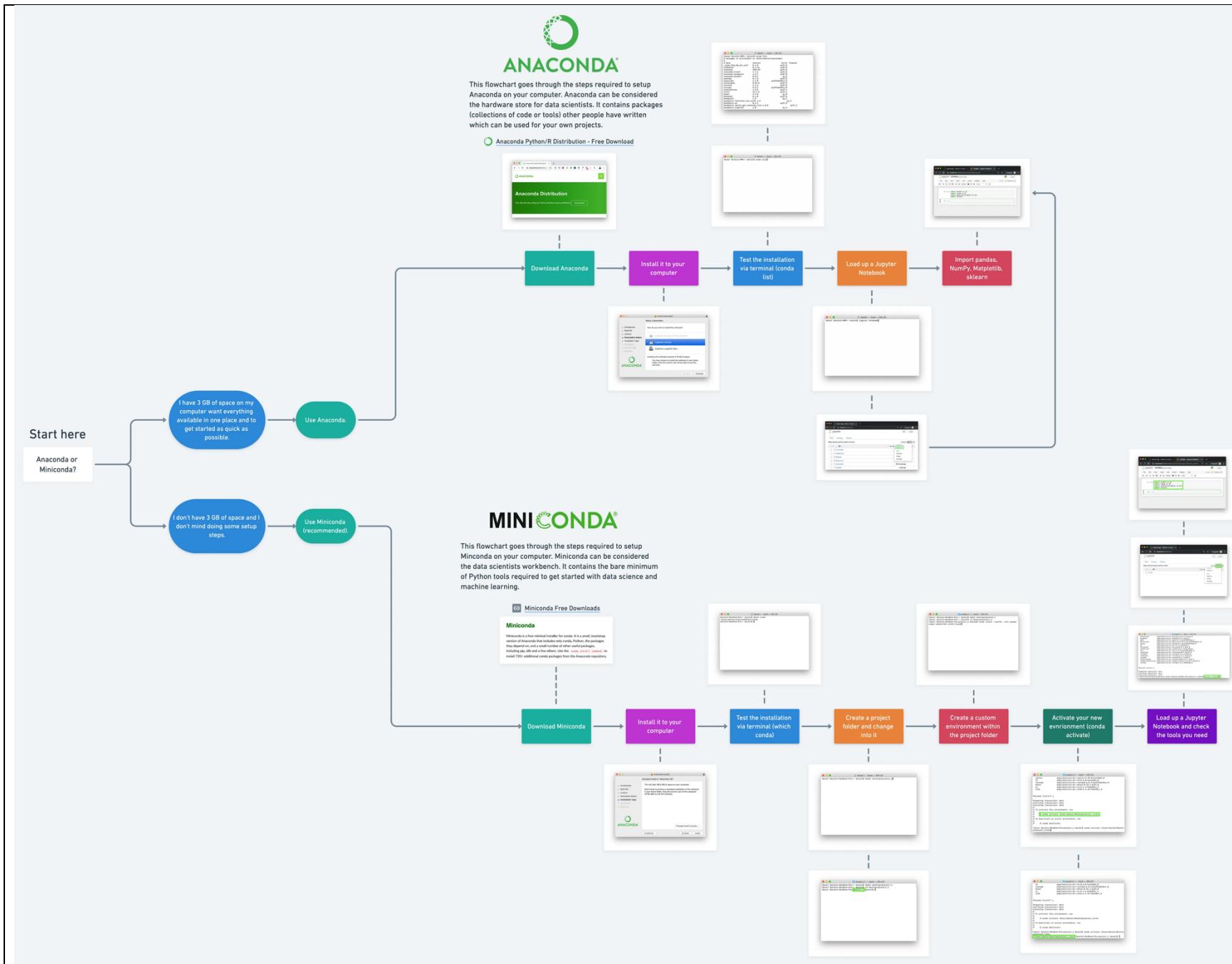


- Miniconda is a free minimal installer for conda.
- Minimum Hard Drive size 400mb free space for installation only.
- Install Python, Conda & Essential Basic Libraries for specific projects
- Uses the `conda install` command to install packages available by default in Anaconda's public repository or from other channels, like conda-forge or bioconda.
- Uses Conda, where conda created to provide greater control over project environments, making it recommended choice for Advance Python programmer user.
- Uses Conda command line program and it works on the Linux Terminal, Windows Anaconda Prompt or Windows Anaconda PowerShell Prompt.
- NO GUI

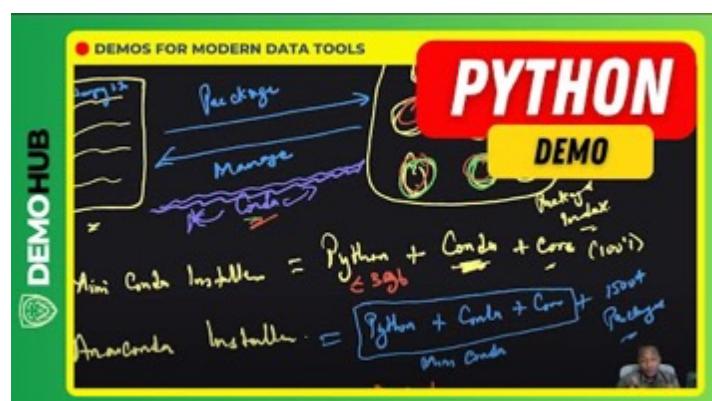
#### 3. What is Anaconda?



- **!!Recommend for this Course!!**
- Anaconda is a All-in-One Free distribution of Python
- Minimum Hard Drive size 3gb+ free space for installation only.
- Include Conda, Python and over 1500+ of fundamental DS, AI, and ML packages.
- Have over 8000+ packages in Anaconda's public repository.
- Support Linux Terminal & Windows PowerShell prompt.
  - Note : Linux need X11 ???
- Anaconda Navigator GUI for easily manage integrated applications, packages, and environments without using the command line.



[Demohub Tips // Python: Conda, Mini Conda vs. Anaconda Explained in Easy Language | www.demohub.dev](https://www.demohub.dev/tips/python-conda-miniconda-explained-in-easy-language)



#### <SOURCES>

- MAIN : <https://www.anaconda.com/download>
- GETTING STARTED CONDA : <https://kaust-vislab.github.io/python-novice-gapminder/00-getting-started-with-conda/index.html>
- MINICONDA : <https://docs.conda.io/projects/miniconda/en/latest/>
- INTRO TO CONDA : <https://geohackweek.github.io/Introductory/01-conda-tutorial/>
- DIFFERENCE CONDA VS ANACONDA : <https://stackoverflow.com/questions/30034840/what-are-the-differences-between-conda-and-anaconda>
- DIFFERENCE CONDA VS ANACONDA VS MINICONDA : <https://eightify.app/summary/programming/differences-conda-anaconda-miniconda-and-pip>



## WINDOWS INSTALLATION GUIDE

URL Installation Guide = <https://docs.anaconda.com/free/anaconda/install/windows/>

1. Go the url : [anaconda.com/download](https://anaconda.com/download) or Google search by using "Anaconda".

2. Scroll down to bottom of page and click on the windows as shown below:



3. Go to your Downloads folder and double-click the installer to launch.

4. Click **Next**.

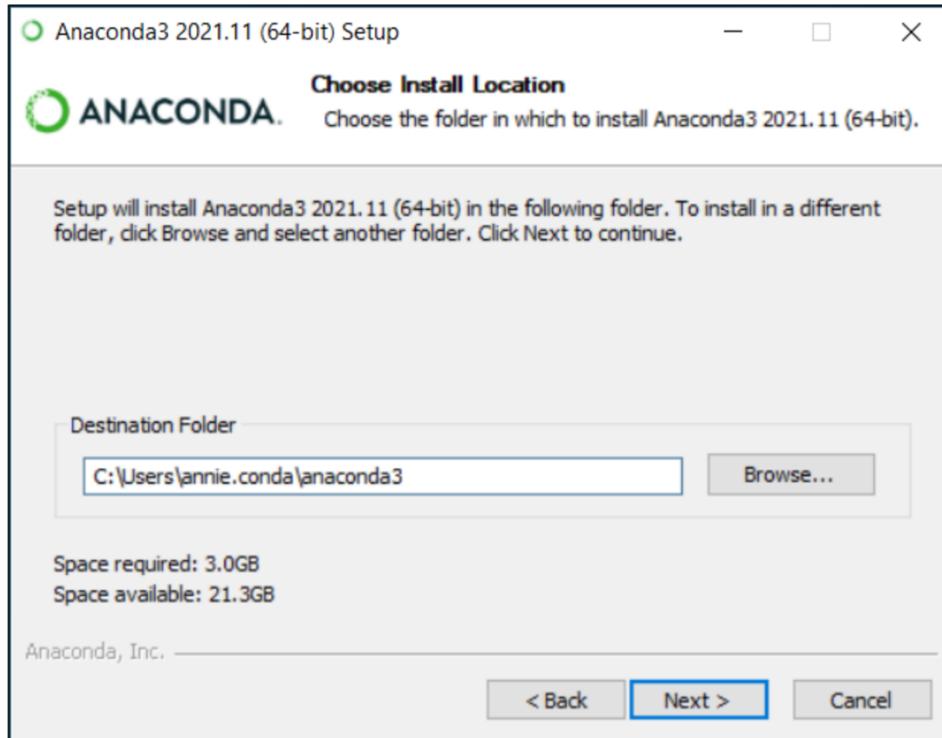
5. Read the licensing terms and click **I Agree**.

6. It is recommended that you install for **Just Me**, as for per user machine.

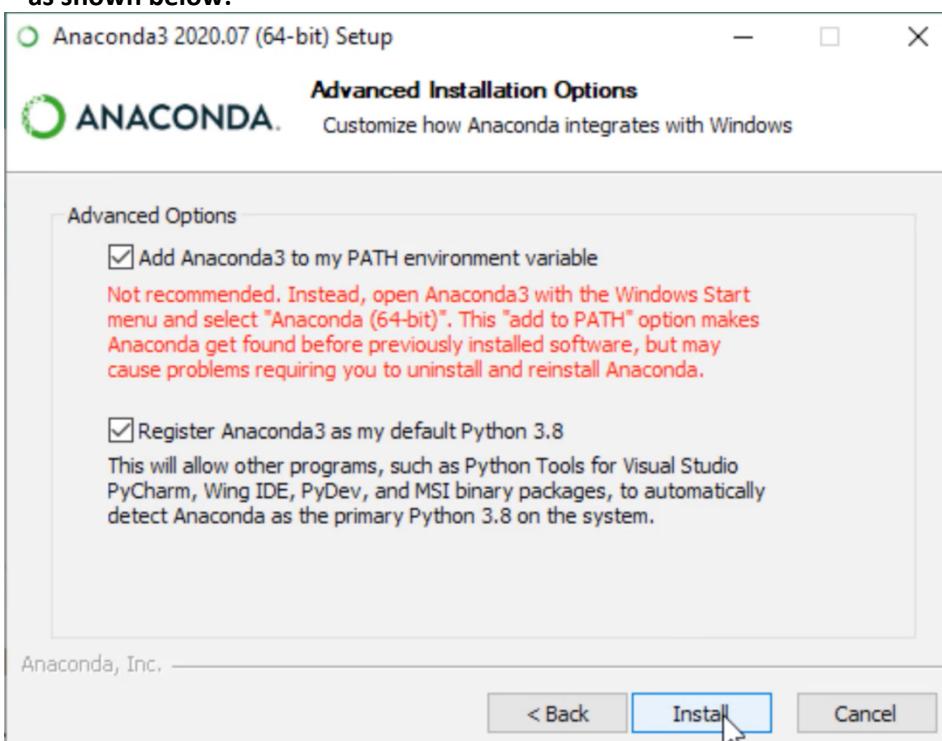
*Note : Only select an install for All Users if you need to install for multiple users' accounts on one computer (which requires Windows Administrator privileges)*

7. Click **Next**.

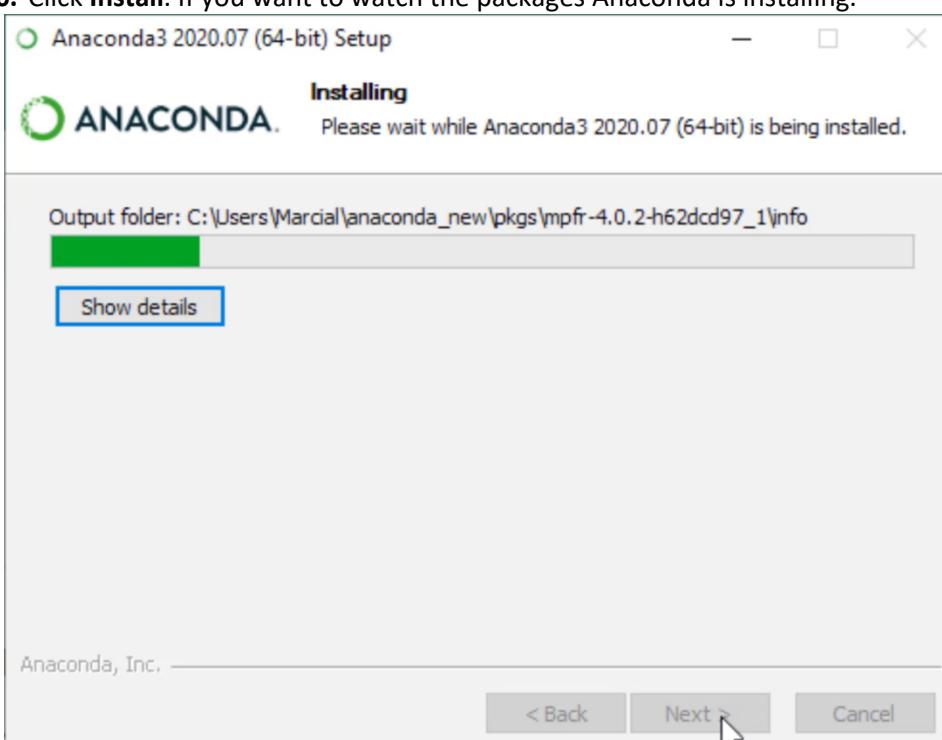
8. Suggest to keep default folder location to install Anaconda and click **Next**.



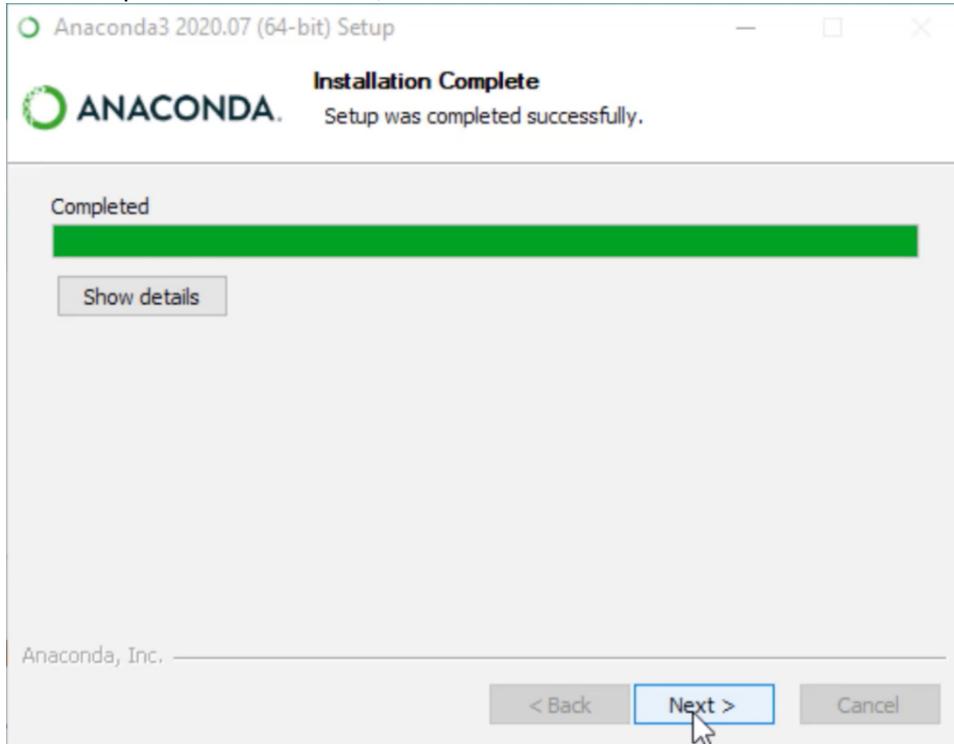
9. Recommended to check marked on box of "Add Anaconda3 to my PATH environment variable" as shown below:



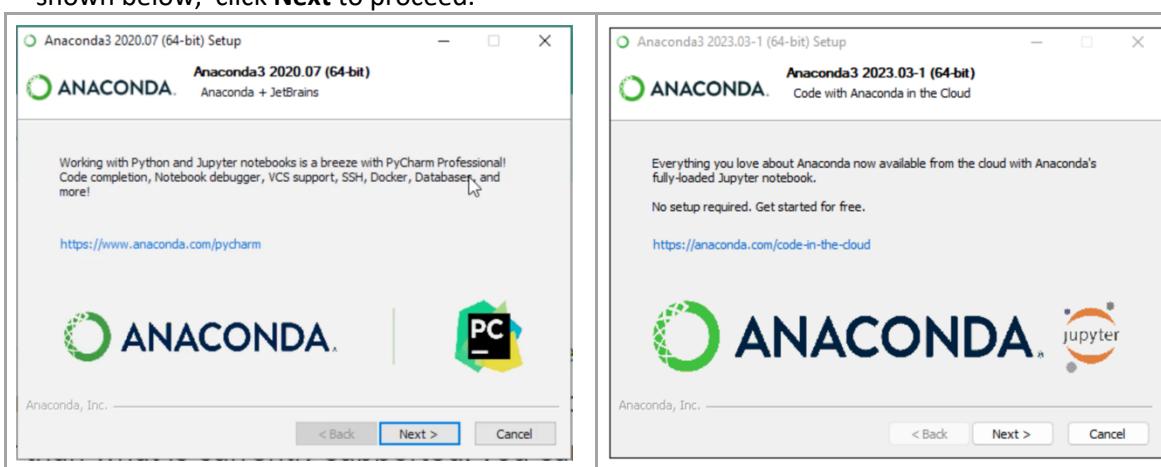
10. Click **Install**. If you want to watch the packages Anaconda is installing.



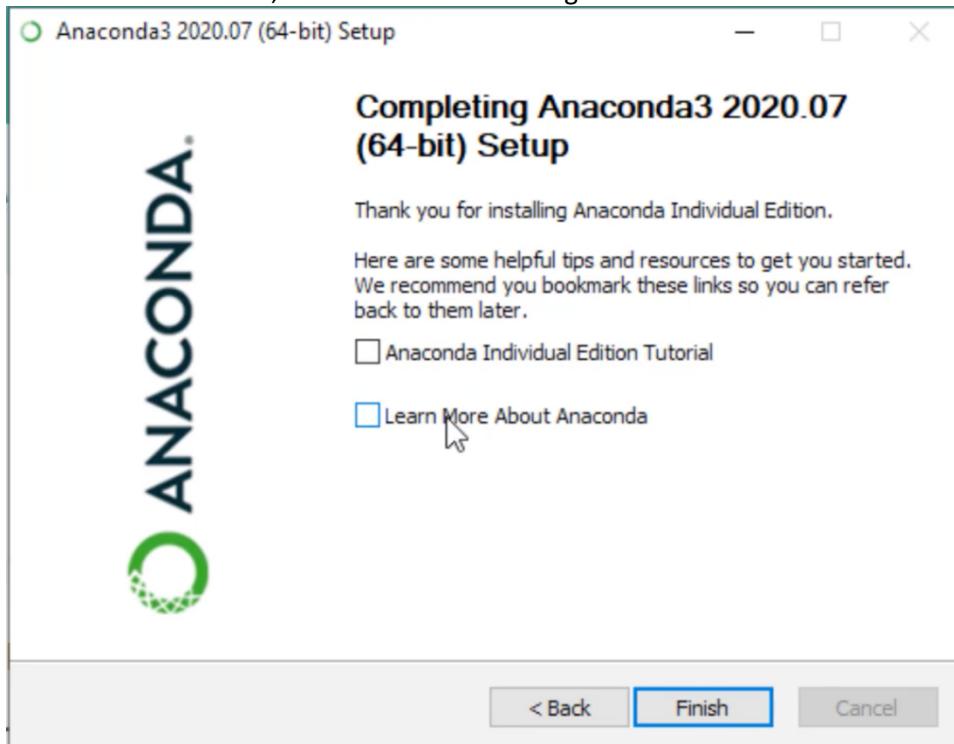
**11.** On completion of installation, Click **Next**.



**12.** On screen "[Anaconda PyCharm](#)" or "[To learn more about Anaconda's cloud notebook service](#)" as shown below, click **Next** to proceed.



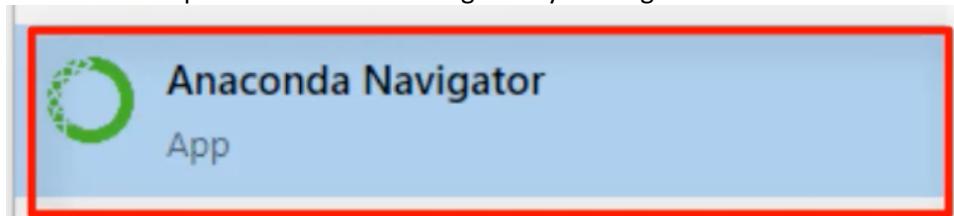
**13.** Uncheck both boxes, as shown on below image and Click the **Finish** button.



## ANACONDA NAVIGATOR INITIAL SETUP

### INITIAL SETUP

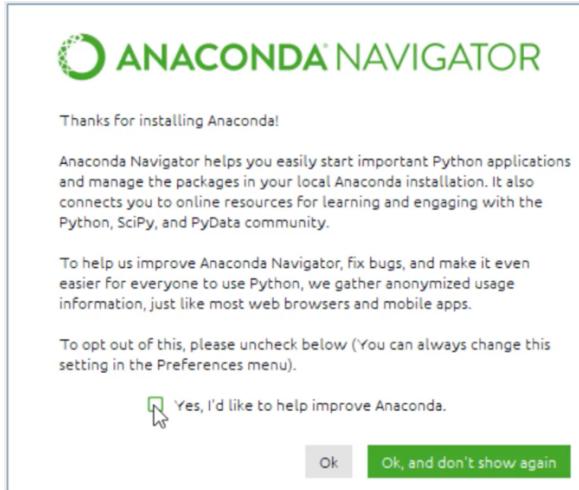
**1.** Search and open the Anaconda Navigator by clicking the icon as shown below:



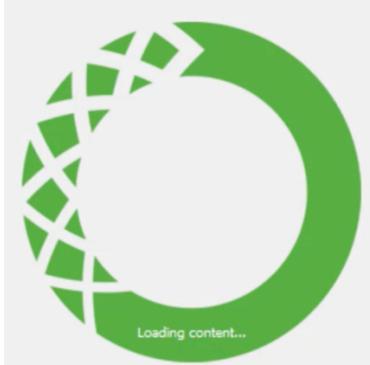
2. If you see on screen for Loading applications..., **So be patient** while it loads.



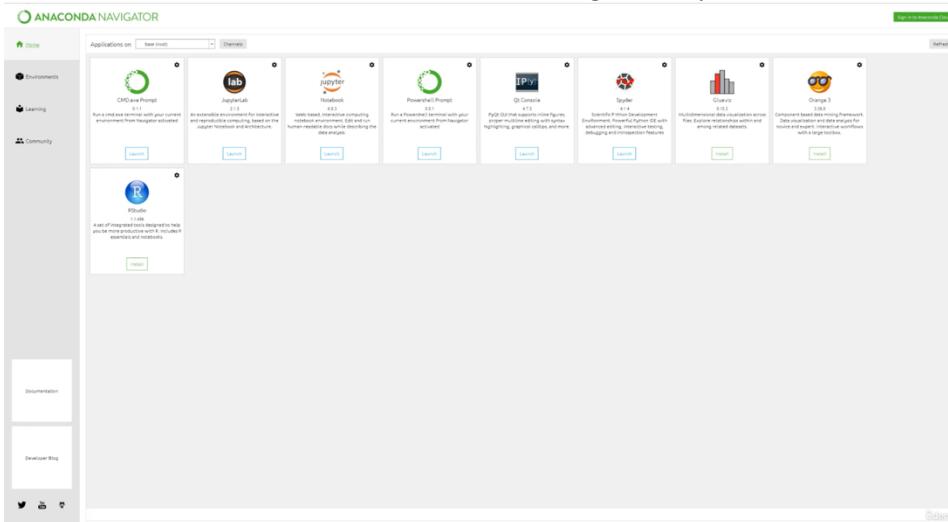
3. Once you see below screen, uncheck the box for sending report and click on **Ok** button.



4. Once again if you see screen of Loading content..., **So be patient** while it fully load the Anaconda Navigator.

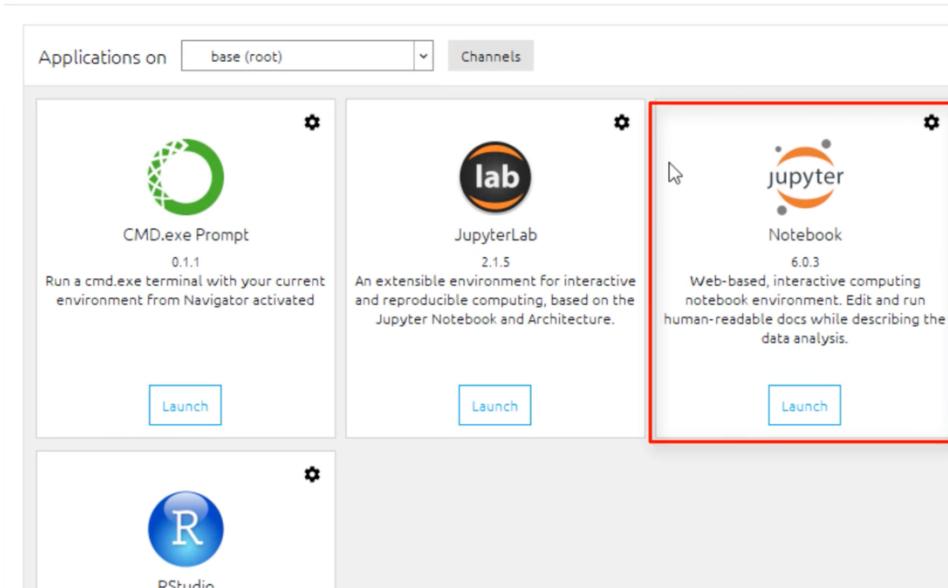


5. You should see below screen of Anaconda Navigator fully loaded.



**Note : this course only be using the Notebook classic as shown in below image.**

## IDA NAVIGATOR



\*\*\*\*\*END OF INSTALLATION\*\*\*\*\*



## LINUX INSTALLATION GUIDE

**NOTE : This is my Installation, not provided by course. Also there is many Linux OS and below guide is build on Linux Ubuntu Server v22.04.**

### INSTRUCTIONS

**Note:** Before proceeding understand there is GUI and Server version of Ubuntu. Also Server version do not support Anaconda Navigator.

**For Anaconda Navigator (GUI of Anaconda) & Jupyter Notebook (GUI over Firefox Browser), make sure you follow below a to c guides prior to following Prerequisites sections.**

- a. [Ubuntu server installation guide](#) <-- with storage requirement partition as define above in hardware requirement sections.
- b. [v22.4.3 : Install GNOME GUI guide](#)
- c. [v22.4.3 : GNOME to WINDOWS LOOK LIKE guide](#)

**For Jupyter Notebook ,make sure you follow below a & d guides prior to following Prerequisites sections.**

- a. [Ubuntu server installation guide](#) <-- with storage requirement partition as define above in hardware requirement sections.
- b. [GNIX : Installation Guide](#)
- c. [OpenSSL - Localhost Self-Sign Certification](#) (Optional and Required for SSL Jupyter)
- d. [Direct Approach Solution Guide](#) (Optional and Recommended for SSL Jupyter) **Note: for nginx v1.24.0, preset already.**

### Prerequisites

To use GUI (Graphical User Interface) packages with Linux, you will need to install the following extended dependencies for Qt:

```
libgl1-mesa-glx  
libegl1-mesa  
libxrandr2  
libxrandr2  
libxss1  
libxcursor1  
libcomposite1  
libasound2  
libxi6  
Libxtst6  
lsb-release  
Python3  
gnupg
```

### STEPS

1. Go to Ubuntu Sever v22.04 terminal and make sure it already uptodate by using below command:

```
$ sudo apt update && upgrade -y  
sysadmin@vm-u24projpython:~$ sudo apt update && upgrade -y  
Hit:1 http://us.archive.ubuntu.com/ubuntu jammy InRelease  
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]  
Hit:3 http://nginx.org/packages/ubuntu jammy InRelease  
Hit:4 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Get:5 http://us.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]  
Fetched 229 kB in 1s (201 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
All packages are up to date.  
upgrade: command not found  
sysadmin@vm-u24projpython:~$
```

2. Type the below command to install :

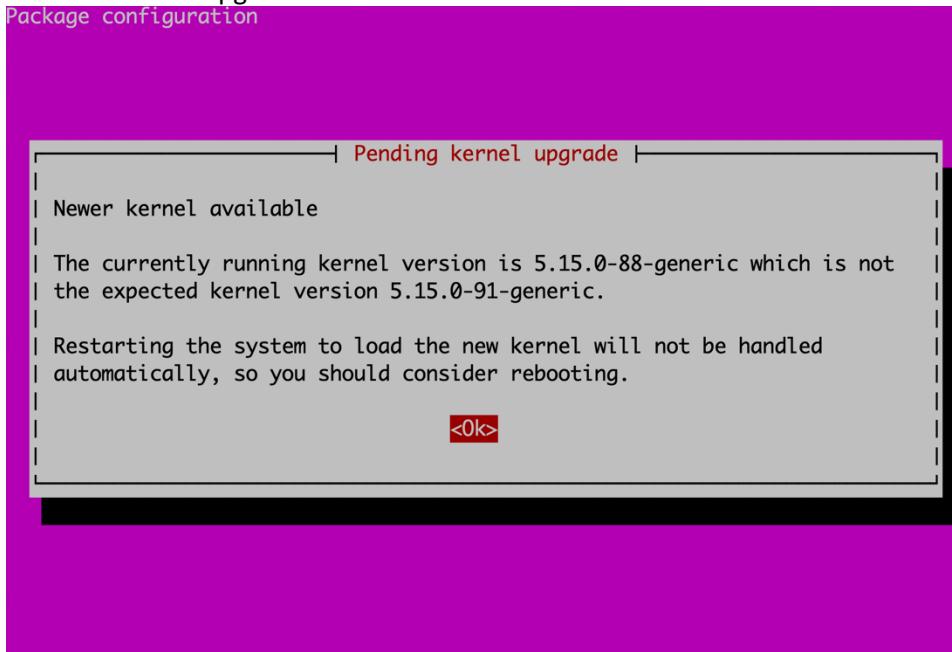
```
$ sudo apt-get install libgl1-mesa-glx libegl1-mesa libxrandr2 libxrandr2 libxss1 libxcursor1 libcomposite1 libasound2 libxi6 libxtst6 lsb-release python3 gnupg
```

3. Type in the password and "Y" without quote to install

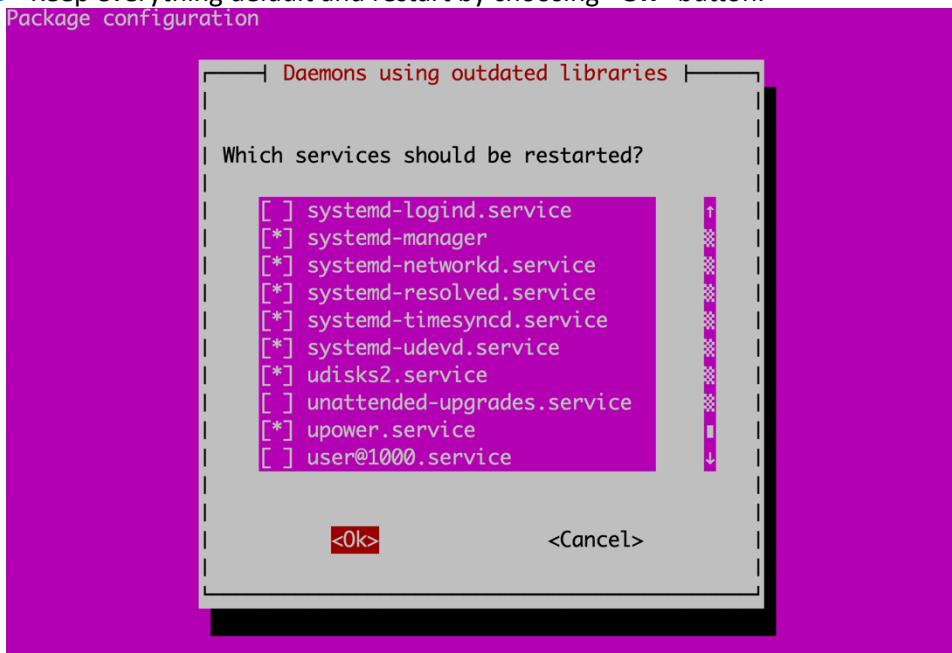
```
sysadmin@vm-u24projpython:~$ sudo apt-get install libgl1-mesa-glx libegl1-mesa libxrandr2 libxrandr2 libxss1 libxcursor1 libcomposite1 libasound2 libxi6 libxtst6 lsb-release python3 gnupg  
Reading package lists... Done
```

```
After this operation, 171 MB of additional disk space will be used.  
Do you want to continue? [Y/n] Y
```

4. Press <OK> to upgrade the kernel



5. Keep everything default and restart by choosing <OK> button.



6. Import the MongoDB public GPG key by using below command:

```
$ wget -qO - https://pgp.mongodb.com/server-7.0.asc | gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/mongodb-org-7.0.gpg
(base) sysadmin@vm-u24projpython:~$ wget -qO - https://pgp.mongodb.com/server-7.0.asc | gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/mongodb-org-7.0.gpg
```

To check key added successfully using below command:

```
$ sudo apt-key list
(base) sysadmin@vm-u24projpython:~$ sudo apt-key list
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead
(see apt-key(8)).
/etc/apt/trusted.gpg.d/mongodb-org-7.0.gpg
-----
pub    rsa4096 2023-01-18 [SC] [expires: 2028-01-17]
      E588 3020 1F7D D82C D808  AA84 160D 26BB 1785 BA38
uid          [ unknown] MongoDB 7.0 Release Signing Key <packaging@mongodb.c
om>

/etc/apt/trusted.gpg.d/nginx-archive-keyring.gpg
-----
pub    rsa2048 2011-08-19 [SC] [expires: 2024-06-14]
      573B FD6B 3D8F BC64 1079  A6AB ABF5 BD82 7BD9 BF62
uid          [ unknown] nginx signing key <signing-key@nginx.com>

/etc/apt/trusted.gpg.d/ubuntu-keyring-2012-cdimage.gpg
-----
pub    rsa4096 2012-05-11 [SC]
      8439 38DF 228D 22F7 B374  2BC0 D94A A3F0 EFE2 1092
uid          [ unknown] Ubuntu CD Image Automatic Signing Key (2012) <cdimag
e@ubuntu.com>

/etc/apt/trusted.gpg.d/ubuntu-keyring-2018-archive.gpg
-----
pub    rsa4096 2018-09-17 [SC]
      F6EC B376 2474 EDA9 D21B  7022 8719 20D1 991B C93C
uid          [ unknown] Ubuntu Archive Automatic Signing Key (2018) <ftpmast
er@ubuntu.com>

sysadmin@vm-u24projpython:~$
```

7. Create the list file for MongoDB, by using below command:

```
$ echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0 multiverse" | sudo tee
/etc/apt/sources.list.d/mongodb-org-7.0.list
(base) sysadmin@vm-u24projpython:~$ echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu jammy/
mongodb-org/7.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-7.0.list
deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0 multiverse
sysadmin@vm-u24projpython:~$
```

8. Update and Upgrade the system by using below command:

```
$ sudo apt update && upgrade -y  
sysadmin@VM-U24ProjPython:~$ sudo apt update && upgrade -y  
[sudo] password for sysadmin: [REDACTED]
```

9. Install Mongodb software by using below command:

```
$ sudo apt install mongodb-org -y  
sysadmin@vm-u24projpython:~$ sudo apt install mongodb-org -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  mongodb-database-tools mongodb-mongosh mongodb-org-database mongodb-org-database-tools-extra  
  mongodb-org-mongos mongodb-org-server mongodb-org-shell mongodb-org-tools  
The following NEW packages will be installed:  
  mongodb-database-tools mongodb-mongosh mongodb-org mongodb-org-database  
  mongodb-org-database-tools-extra mongodb-org-mongos mongodb-org-server mongodb-org-shell  
  mongodb-org-tools  
0 upgraded, 9 newly installed, 0 to remove and 3 not upgraded.  
Need to get 162 MB of archives.  
After this operation, 530 MB of additional disk space will be used.  
Get:1 https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0/multiverse amd64 mongodb-database-tools amd64 100.9.4 [51.9 MB]  
Get:2 https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0/multiverse amd64 mongodb-mongosh amd64 2.1.1 [47.9 MB]  
45% [2 mongodb-mongosh 35.3 MB/47.9 MB 74%] 11.0 MB/s 6s [REDACTED]  
No containers need to be restarted.  
No user sessions are running outdated binaries.  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
sysadmin@vm-u24projpython:~$ [REDACTED]
```

10. Reboot the machine

## Installation (Following AIO Package : Anaconda, Anaconda Navigator, Jupyter, etc..)

1. To download the installer, open a terminal and use the following command:

```
# Replace <INSTALLER_VERSION> with the version of the installer file you want to download# All installers can be found  
at repo.anaconda.com/archive OR "anaconda.com/download"  
  
$ curl -O https://repo.anaconda.com/archive/Anaconda3-<INSTALLER\_VERSION>-Linux-x86\_64.sh  
$ curl -O https://repo.anaconda.com/archive/Anaconda3-2023.09-0-Linux-x86\_64.sh  
sysadmin@VM-U24ProjPython:~$ curl -O https://repo.anaconda.com/archive/Anaconda3-  
-2023.09-0-Linux-x86_64.sh  
  % Total    % Received % Xferd  Average Speed   Time     Time     Time  Current  
          Dload  Upload Total   Spent    Left Speed  
100 1099M  100 1099M    0      0  10.9M      0  0:01:40  0:01:40  --:--:-- 11.1M  
sysadmin@VM-U24ProjPython:~$ [REDACTED]
```

2. Anaconda recommends verifying the integrity of the installer after downloading it by using below command:

```
$ shasum -a 256 Anaconda3-2023.09-0-Linux-x86_64.sh  
sysadmin@VM-U24ProjPython:~$ ls  
Anaconda3-2023.09-0-Linux-x86_64.sh  
sysadmin@VM-U24ProjPython:~$ shasum -a 256 Anaconda3-2023.09-0-Linux-x86_64.sh  
6c8a4abb36fb711dc055b7049a23bbfd61d356de9468b41c5140f8a11abd851  Anaconda3-2023  
.09-0-Linux-x86_64.sh  
sysadmin@VM-U24ProjPython:~$ [REDACTED]
```

3. To install, run the following command:

```
$ bash Anaconda3-2023.09-0-Linux-x86_64.sh  
sysadmin@vm-u24projpython:~$ ls  
Anaconda3-2023.09-0-Linux-x86_64.sh  
sysadmin@vm-u24projpython:~$ bash Anaconda3-2023.09-0-Linux-x86_64.sh  
  
Welcome to Anaconda3 2023.09-0  
[REDACTED]
```

4. Press **Enter** to review the license agreement.

```
sysadmin@vm-u24projpython:~$ bash Anaconda3-2023.09-0-Linux-x86_64.sh  
  
Welcome to Anaconda3 2023.09-0  
  
In order to continue the installation process, please review the license  
agreement.  
Please, press ENTER to continue  
>>> [REDACTED]
```

5. Then press and keep pressing Enter to scroll.

```
=====
End User License Agreement - Anaconda Distribution
=====

Copyright 2015-2023, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

This End User License Agreement (the "Agreement") is a legal agreement between you and Anaconda, Inc. ("Anaconda") and governs your use of Anaconda Distribution (which was formerly known as Anaconda Individual Edition).

Subject to the terms of this Agreement, Anaconda hereby grants you a non-exclusive, non-transferable license to:

    * Install and use the Anaconda Distribution (which was formerly known as Anaconda Individual Edition),
    * Modify and create derivative works of sample source code delivered in Anaconda Distribution from Anaconda's repository, and;
    * Redistribute code files in source (if provided to you by Anaconda as source) and binary forms, with or without modification subject to the requirements set forth below, and;

--More--
```

```
Export; Cryptography Notice
=====

You must comply with all domestic and international export laws and regulations that apply to the software, which include restrictions on destinations, end users, and end use. Anaconda Distribution includes cryptographic software. The country in which you currently reside may have restrictions on the import, possession, use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check your country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted. See the Wassenaar Arrangement http://www.wassenaar.org/ for more information.

Anaconda has self-classified this software as Export Commodity Control Number (ECCN) EAR99 which includes mass market information security software using or performing cryptographic functions with asymmetric algorithms. No license is required for export of this software to non-embargoed countries.

The Intel Math Kernel Library contained in Anaconda Distribution is classified by Intel as ECCN 5D992.c with no license required for export to non-embargoed countries.

--More--
```

```
n, use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check your country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted. See the Wassenaar Arrangement http://www.wassenaar.org/ for more information.

Anaconda has self-classified this software as Export Commodity Control Number (ECCN) EAR99 which includes mass market information security software using or performing cryptographic functions with asymmetric algorithms. No license is required for export of this software to non-embargoed countries.

The Intel Math Kernel Library contained in Anaconda Distribution is classified by Intel as ECCN 5D992.c with no license required for export to non-embargoed countries.

The following packages listed on https://www.anaconda.com/cryptography are included in the repository accessible through Anaconda Distribution that relate to cryptography.

Last updated February 25, 2022
```

Do you accept the license terms? [yes|no]  
[no] >>>

6. Type "yes" without quote and press "Enter" in keyboard, to agree to the license agreement.

```
The following packages listed on https://www.anaconda.com/cryptography are included in the repository accessible through Anaconda Distribution that relate to cryptography.
```

Last updated February 25, 2022

Do you accept the license terms? [yes|no]  
[no] >>> yes

7. Use **Enter** to accept the default install location to continues the installation. It may take a few minutes to complete

```
Do you accept the license terms? [yes|no]
[no] >>> yes

Anaconda3 will now be installed into this location:
/home/sysadmin/anaconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/home/sysadmin/anaconda3] >>> 
```

```
[/home/sysadmin/anaconda3] >>>
PREFIX=/home/sysadmin/anaconda3
Unpacking payload ... 
```

```
Unpacking payload ...
Extracting : prometheus_client-0.14.1-py311h06a4308_0.conda: 24%| | 124/517 [00
```

```
Unpacking payload ...
Extracting : black-23.3.0-py311h06a4308_0.conda: 73%| | 378/517 [02:26<00:56, ]
```

```
Unpacking payload ...
Extracting : pathlib-1.0.1-pyhd3eb1b0_1.conda: 93%| | 481/517 [03:42<00:29, 1.
```

```
Installing base environment... 
```

```
Downloading and Extracting Packages

Preparing transaction: done
Executing transaction: \

    Installed package of scikit-learn can be accelerated using scikit-learn-intel.
    More details are available here: https://intel.github.io/scikit-learn-intel
x

    For example:

        $ conda install scikit-learn-intel
        $ python -m sklearnex my_application.py

done
installation finished. 
```

8. Anaconda recommends you enter “**yes**” to initialize Anaconda Distribution by running “**conda init**” command, as define below:

```
done
installation finished.
Do you wish to update your shell profile to automatically initialize conda?
This will activate conda on startup and change the command prompt when activated
.
If you'd prefer that conda's base environment not be activated on startup,
run the following command when conda is activated:

conda config --set auto_activate_base false

You can undo this by running `conda init --reverse $SHELL`? [yes|no]
[no] >>> yes 
```

```
You can undo this by running `conda init --reverse $SHELL`? [yes|no]
[no] >>> yes
no change   /home/sysadmin/anaconda3/condabin/conda
no change   /home/sysadmin/anaconda3/bin/conda
no change   /home/sysadmin/anaconda3/bin/conda-env
no change   /home/sysadmin/anaconda3/bin/activate
no change   /home/sysadmin/anaconda3/bin/deactivate
no change   /home/sysadmin/anaconda3/etc/profile.d/conda.sh
no change   /home/sysadmin/anaconda3/etc/fish/conf.d/conda.fish
no change   /home/sysadmin/anaconda3/shell/condabin/Conda.ps1
no change   /home/sysadmin/anaconda3/shell/condabin/conda-hook.ps1
no change   /home/sysadmin/anaconda3/lib/python3.11/site-packages/xontrib/condaxsh
no change   /home/sysadmin/anaconda3/etc/profile.d/conda.csh
modified    /home/sysadmin/.bashrc

==> For changes to take effect, close and re-open your current shell. <==

Thank you for installing Anaconda3!
sysadmin@vm-u24projpython:~$ 
```

9. Now disable the **auto\_update & channel\_priority** in conda, for getting **UpToDate** and avoid downgrade of package. To disable use below command:

```
$ conda config --set auto_update_conda false
(base) sysadmin@vm-u24projpython:~$ conda config --set auto_update_conda false
$ conda config --set channel_priority false
(base) sysadmin@vm-u24projpython:~$ conda config --set channel_priority false 
```

## RESULT

```
(base) sysadmin@vm-u24projpython:~$ cat .condarc
auto_update_conda: false
channel_priority: disabled
(base) sysadmin@vm-u24projpython:~$ 
```

**10.** Reboot the machine

```
$ sudo reboot
```

**11.** Now update the Anconda packages with below command:

```
$ conda update --all
```

```
(base) sysadmin@vm-u24projpython:~$ conda update --all
Collecting package metadata (current_repodata.json): done
Solving environment: /
```

```
The following packages will be DOWNGRADED:
```

```
scikit-learn          1.3.0-py311ha02d727_0 --> 1.2.2-py311h6a678d5_1
```

```
Proceed ([y]/n)? y
```

```
astropy-5.3.4          | 9.9 MB    | #####| 100%
rpy3-py-0.10.6         | 1007 KB   | #####| 100%
sip-6.7.12              | 603 KB    | #####| 100%
conda-24.1.0             | 1.2 MB    | #####| 100%
intel-openmp-2023.1.0     | 17.2 MB   | #####| 99%
pyqt-5.15.10             | 5.7 MB    | #####| 98%
... (more hidden) ...
```

```
Preparing transaction: done
Verifying transaction: done
Executing transaction: -
```

```
Installed package of scikit-learn can be accelerated using scikit-learn-intelex.
More details are available here: https://intel.github.io/scikit-learn-intelex
```

```
For example:
```

```
$ conda install scikit-learn-intelex
$ python -m sklearnex my_application.py
```

```
done
```

```
(base) sysadmin@vm-u24projpython:~$
```

**12.** Verify your installation with below command:

- **\$ conda list**

*Note : Show Anaconda is installed and working, this will display a list of installed packages and their versions.*

```
(base) sysadmin@vm-u24projpython:~$ conda list
# packages in environment at /home/sysadmin/anaconda3:
#
# Name           Version      Build  Channel
_anaconda_depends  2023.09      py311_mkl_1
_libgcc_mutex       0.1          main
```

- **\$ python**

*Note : This command runs the Python shell, also known as the REPL. If Anaconda is installed and working, the version information it displays when it starts up will include "Anaconda". To exit the Python shell, enter the command \$ quit().*

```
(base) sysadmin@vm-u24projpython:~$ python
Python 3.11.5 (main, Sep 11 2023, 13:54:46) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> quit()
(base) sysadmin@vm-u24projpython:~$
```

- **\$ anaconda-navigator !! Need Linux Graphical Display Manager or Linux Desktop Environment \*\* SKIP FOR SERVER\*\* !!**

*Note : Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® Distribution that allows you to launch applications and manage conda packages, environments, and channels without using command line interface (CLI) commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository. It is available for Windows, macOS, and Linux.*



**ANACONDA.NAVIGATOR**

The Anaconda Navigator interface displays a grid of installed applications:

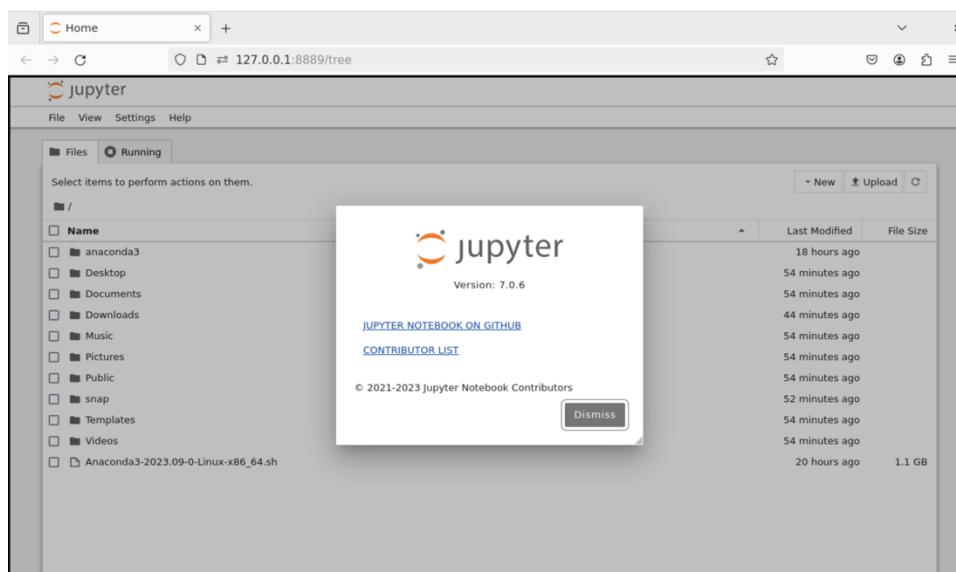
- Anaconda Cloud Notebooks**: Cloud-hosted notebook service from Anaconda.
- JupyterLab**: An extensible environment for interactive and reproducible computing, based on Jupyter Notebook and Architecture.
- jupyter Notebook**: Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
- IPyConsole**: PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
- Spyder**: Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features.
- AWS Graviton**: Running your Anaconda workloads on AWS Graviton-based processors could provide up to 40% better price performance.
- Datalore**: Kick-start your data science projects in seconds in a pre-configured environment. Enjoy coding assistance for Python, SQL, and BigQuery notebooks, and AI model management and code customizations. Use Datalore online for free.
- IBM WatsonX**: IBM WatsonX is an enterprise-ready AI platform including a data store, model builder, and AI model management and monitoring.
- ORACLE Cloud Infrastructure**: OCI Data Science offers a machine learning platform to build, train, manage, and deploy your machine learning models on the cloud with your favorite open-source tools.

**\$ jupyter-notebook !! Need Linux Graphical Display Manager , Linux Desktop Environment or Ubuntu Server Nginx Reverse Proxy !!**

*Note : A notebook is a shareable document that combines computer code, plain language descriptions, data, rich visualizations like 3D models, charts, graphs and figures, and interactive controls. A notebook, along with an editor (like JupyterLab), provides a fast interactive environment for prototyping and explaining code, exploring and visualizing data, and sharing ideas with others.*

```
sysadmin@vm-u24projpython: ~
(base) sysadmin@vm-u24projpython:~$ jupyter-notebook
[I 2024-01-25 20:58:10.555 ServerApp] Package notebook took 0.0000s to import
[I 2024-01-25 20:58:10.653 ServerApp] Package jupyter_lsp took 0.0971s to import
[W 2024-01-25 20:58:10.653 ServerApp] A `jupyter_server_extension_points` funct
[+]
sysadmin@vm-u24projpython: ~
[I 2024-01-25 20:58:18.284 ServerApp] panel.io.jupyter_server_extension | extens
ion was successfully loaded.
[I 2024-01-25 20:58:18.284 ServerApp] The port 8888 is already in use, trying an
other port.
[I 2024-01-25 20:58:18.286 ServerApp] Serving notebooks from local directory: /h
ome/sysadmin
[I 2024-01-25 20:58:18.287 ServerApp] Jupyter Server 2.10.0 is running at:
[I 2024-01-25 20:58:18.287 ServerApp] http://localhost:8889/tree?token=4a62f3cc0
ddda622f47b220cb16c5eedad0fafcea27a8e45aa
[I 2024-01-25 20:58:18.287 ServerApp] http://127.0.0.1:8889/tree?token=4a62f
3cc0ddda622f47b220cb16c5eedad0fafcea27a8e45aa
[I 2024-01-25 20:58:18.287 ServerApp] Use Control-C to stop this server and shut
down all kernels (twice to skip confirmation).
[C 2024-01-25 20:58:18.328 ServerApp]

To access the server, open this file in a browser:
file:///home/sysadmin/.local/share/jupyter/runtime/jpserver-6750-open.ht
ml
Or copy and paste one of these URLs:
http://localhost:8889/tree?token=4a62f3cc0ddda622f47b220cb16c5eedad0fafcea
27a8e45aa
http://127.0.0.1:8889/tree?token=4a62f3cc0ddda622f47b220cb16c5eedad0fafcea
27a8e45aa
[I 2024-01-25 20:58:18.508 ServerApp] Skipped non-installed server(s): bash-lang
```



**13.** Reboot the machine

```
$ sudo reboot
```

## GUIDE : CONFIGURE JUPYTER NOTEBOOK for UBUNTU SERVER using NGINX REVERSE PROXY.

*Note: Below step help access Jupyter Notebook using different computer over browser, as no GUI in ubuntu server.*

- Create the Jupyter Notebook config file by using below command:

```
$ jupyter notebook --generate-config
(base) sysadmin@vm-u24projpython:~$ jupyter notebook --generate-config
Writing default config to: /home/sysadmin/.jupyter/jupyter_notebook_config.py
(base) sysadmin@vm-u24projpython:~$
```

- Create Jupyter notebook password by using below command:

```
$ jupyter notebook password
Password : Jn2024!
(base) sysadmin@vm-u24projpython:~$ jupyter notebook password
Enter password:
Verify password:
[JupyterPasswordApp] Wrote hashed password to /home/sysadmin/.jupyter/jupyter_server_config.json
(base) sysadmin@vm-u24projpython:~$
```

- Copy the HASH password saved in `jupyter_server_config.json`, by opening the file using below command:

```
$ sudo nano .jupyter/jupyter_server_config.json
(base) sysadmin@vm-u24projpython:~$ sudo nano .jupyter/jupyter_server_config.json
[sudo] password for sysadmin:
```

**Hashed\_password :**

```
"argon2:$argon2id$v=19$m=10240,t=10,p=8$784kpGBIyPYvmDMnsyT9VQ$3HvTvYi78RCzrSDSvo/N4ygkbtF1ktnjKT5fbywBR7o"
GNU nano 6.2
.jupyter/jupyter_server_config.json

"IdentityProvider": {
  "hashed_password": "argon2:$argon2id$v=19$m=10240,t=10,p=8$784kpGBIyPYvmDMnsyT9VQ$3HvTvYi78RCzrSDSvo/N4ygkbtF1ktnjKT5fbywBR7o"
}
```

- Open and Paste the password with below information in the "`jupyter_notebook_config.py`" file, by using below command:

```
$ sudo nano /home/sysadmin/.jupyter/jupyter_notebook_config.py
```

### For https SSL certificate

*Note : Use only if Nginx + OpenSSL is setup. In this manual did setup the Nginx+OpenSSL*

```
c.NotebookApp.certfile = 'Your Cert'
  c.NotebookApp.certfile = 'Your cert'
c.NotebookApp.keyfile = 'Your key '
  c.NotebookApp.keyfile = 'Your key'
```

*Example of Certfile : /etc/ssl/localhost/nginx.crt  
Example of Keyfile: /etc/ssl/localhost/nginx.key*

```
c.NotebookApp.password = u'YOUR_HASHED_PASSWORD_HERE'
  c.NotebookApp.password = u'YOUR_HASHED_PASSWORD_HERE'
```

*Example of Hashed\_password :
"argon2:\$argon2id\$v=19\$m=10240,t=10,p=8\$784kpGBIyPYvmDMnsyT9VQ\$3HvTvYi78RCzrSDSvo/N4ygkbtF1ktnjKT5fbywBR7o"*

```
c.NotebookApp.allow_origin = '*'
c.NotebookApp.ip = '*'
c.NotebookApp.allow_remote_access = True
c.NotebookApp.open_browser = False
c.NotebookApp.port = 8888
c.NotebookApp.trust_xheaders = True
c.NotebookApp.quit_button = False
```

**c.NotebookApp.base\_url = 'JN' <--\*Must Add this entry for Nginx to Work\***

**DEFAULT**

```
GNU nano 6.2                               /home/sysadmin/.jupyter/jupyter_notebook_config.py
# Configuration file for notebook.

c = get_config() #noqa

#-----
# Application(SingletonConfigurable) configuration
#-----
## This is an application.
```

**MY LIST**

```
# Configuration file for notebook.

c = get_config() #noqa

#-----
# My Lists
#-----
c.NotebookApp.certfile = '/etc/ssl/localhost/nginx.crt'
c.NotebookApp.keyfile = '/etc/ssl/localhost/nginx.key'

c.NotebookApp.password = u'argon2:$argon2id$v=19$m=10240,t=10,p=8$784kpGBIyPYvmDmnsyT9VQ$3HvTVYi78RCzrSDSvo/V4ygkbtF1ktnjKT5fbyw8R7o'

c.NotebookApp.allow_origin = '*'
c.NotebookApp.ip = '*'
c.NotebookApp.allow_remote_access = True
c.NotebookApp.open_browser = False
c.NotebookApp.port = 8888
c.NotebookApp.trust_xheaders = True
c.NotebookApp.quit_button = False

c.NotebookApp.base_url = 'ON'

#-----
# Application(SingletonConfigurable) configuration
#-----
```

The above configuration does the following things:

- The **password** enables login via a password. Note that this is the hashed password, not plain text.
- **allow\_origin** and **allow\_remote\_access** enable that other hosts than localhost access a notebook.
- **port** sets the port our server now listens to.
- **open\_browser** prevents Jupyter from trying to open a browser window when starting it in the terminal.
- **trust\_xheaders** is necessary since we want our proxy server to handle SSL.
- **quit\_button** you disable the button with which one can shut down Jupyter Notebook. Since we run Jupyter on a server and every time we stop it have to first connect to the server, change user and change directory I prefer to not accidentally shut down Jupyter.

Next **CMD+O** or **CTRL+O** & **CMD+X** or **CTRL+X** to save and close

**5. Create and give full access to "ProjPython" folder using below command:**

```
$ mkdir -m 777 ProjPython
```

```
(base) sysadmin@vm-u24projpython:~$ mkdir -m 777 ProjPython
(base) sysadmin@vm-u24projpython:~$ ls -all
total 1126488
drwxr-x-- 14 sysadmin sysadmin 4096 Feb  7 22:33 .
drwxr-xr-x  4 root   root    4096 Jan 24 01:54 ..
drwxrwxr-x  2 sysadmin sysadmin 4096 Feb  5 15:12 777
drwxrwxr-x  3 sysadmin sysadmin 4096 Jan 25 02:12 .anaconda
drwxrwxr-x 31 sysadmin sysadmin 4096 Feb  6 22:41 anaconda3
-rw-rw-r--  1 sysadmin sysadmin 1153404910 Jan 25 00:43 Anaconda3-2023.09-0-Linux-x86_64.sh
-rw-r--r--  1 sysadmin sysadmin 22827 Feb  6 23:02 .bash_history
-rw-r--r--  1 sysadmin sysadmin 220 Jan  6 2022 .bash_logout
-rw-r--r--  1 sysadmin sysadmin 4258 Jan 25 00:56 .bashrc
drwx----- 4 sysadmin sysadmin 4096 Feb  6 23:59 .cache
drwxrwxr-x  2 sysadmin sysadmin 4096 Feb  5 15:12 chmod
drwxrwxr-x  2 sysadmin sysadmin 4096 Jan 25 02:12 .conda
drwx----- 2 sysadmin sysadmin 4096 Jan 29 22:59 .gnupg
drwxrwxr-x  2 sysadmin sysadmin 4096 Jan 25 16:00 .ipython
drwx----- 2 sysadmin sysadmin 4096 Feb  7 20:43 .jupyter
-rw-----  1 sysadmin sysadmin 20 Feb  6 17:42 .lessht
drwxrwxr-x  3 sysadmin sysadmin 4096 Jan 25 16:00 .local
-rw-r--r--  1 sysadmin sysadmin 807 Jan  6 2022 .profile
drwxrwxrwx  2 sysadmin sysadmin 4096 Feb  7 22:33 ProjPython
-rw-----  1 sysadmin sysadmin 8 Jan 25 01:12 .python_history
drwx----- 2 sysadmin sysadmin 4096 Jan 24 01:54 .ssh
-rw-r--r--  1 sysadmin sysadmin 0 Jan 24 19:27 .sudo_as_admin_successful
-rw-----  1 sysadmin sysadmin 62 Jan 25 02:41 .Xauthority
(base) sysadmin@vm-u24projpython:~$
```

**6. To create the Jupyter Notebook service file, type the below command:**

```
$ sudo nano /etc/systemd/system/jupyter_notebook.service
```

```
(base) sysadmin@vm-u24projpython:~$ sudo nano /etc/systemd/system/jupyter_notebook.service
```

**7. Next past the below text, then save and close.**

**Note: "sysadmin" is user home directory.**

```
[Unit]
```

```
Description=Jupyter notebook
```

```
After=network.target
```

```
StartLimitIntervalSec=0
```

```
[Service]
```

```
Type=simple
```

```
Restart=always
```

```
RestartSec=1
```

```
User=sysadmin
```

```
ExecStart=/home/sysadmin/anaconda3/bin/jupyter-notebook --no-browser --
```

```
config=/home/jupyter/.jupyter/jupyter_notebook_config.py
```

```
Environment="PATH=/home/sysadmin/anaconda3/bin:/home/sysadmin/anaconda3"
```

```
WorkingDirectory=/home/sysadmin/ProjPython/
```

[Install]

WantedBy=multi-user.target

```
GNU nano 6.2                               /etc/systemd/system/jupyter_notebook.service *
[Unit]
Description=Jupyter notebook
After=network.target
StartLimitIntervalSec=0

[Service]
Type=simple
Restart=always
RestartSec=1
User=sysadmin
ExecStart=/home/sysadmin/anaconda3/bin/jupyter-notebook --no-browser --config=/home/jupyter/.jupyter/jupyter_notebook_config.py
Environment="PATH=/home/sysadmin/anaconda3/bin:/home/sysadmin/anaconda3"
WorkingDirectory=/home/sysadmin/ProjPython/

[Install]
WantedBy=multi-user.target

AG Help      A0 Write Out   AW Where Is   AC Cut       AT Execute   AC Location   M-U Undo   M-A Set Mark
AX Exit      AR Read File  AR Replace   AL Paste     AJ Justify   AG Go To Line  M-E Redo   M-G Copy
```

Next **CMD+O** or **CTRL+O** & **CMD+X** or **CTRL+X** to save and close

8. Restart the service monitor to start the service by using below commands:

```
$ sudo systemctl daemon-reload
$ sudo systemctl enable jupyter_notebook.service
$ sudo systemctl start jupyter_notebook.service
$ sudo systemctl status jupyter_notebook.service
(base) sysadmin@vm-u24projpython:~$ sudo systemctl daemon-reload
(base) sysadmin@vm-u24projpython:~$ sudo systemctl enable jupyter_notebook.service
(base) sysadmin@vm-u24projpython:~$ sudo systemctl start jupyter_notebook.service
(base) sysadmin@vm-u24projpython:~$ sudo systemctl status jupyter_notebook.service
● jupyter_notebook.service - Jupyter notebook
  Loaded: loaded (/etc/systemd/system/jupyter_notebook.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2024-02-07 22:53:36 UTC; 21s ago
```

9. Reboot the machine by using below command

```
$ sudo reboot
```

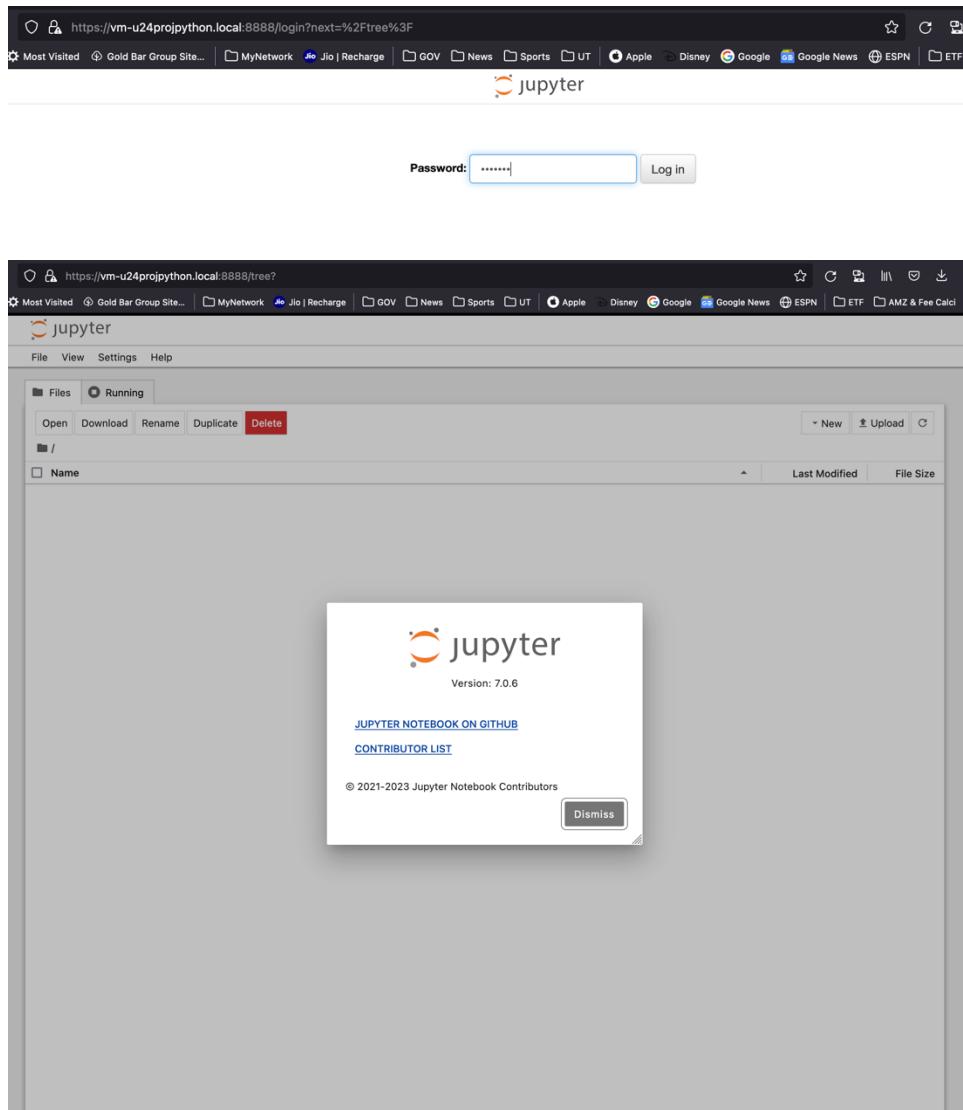
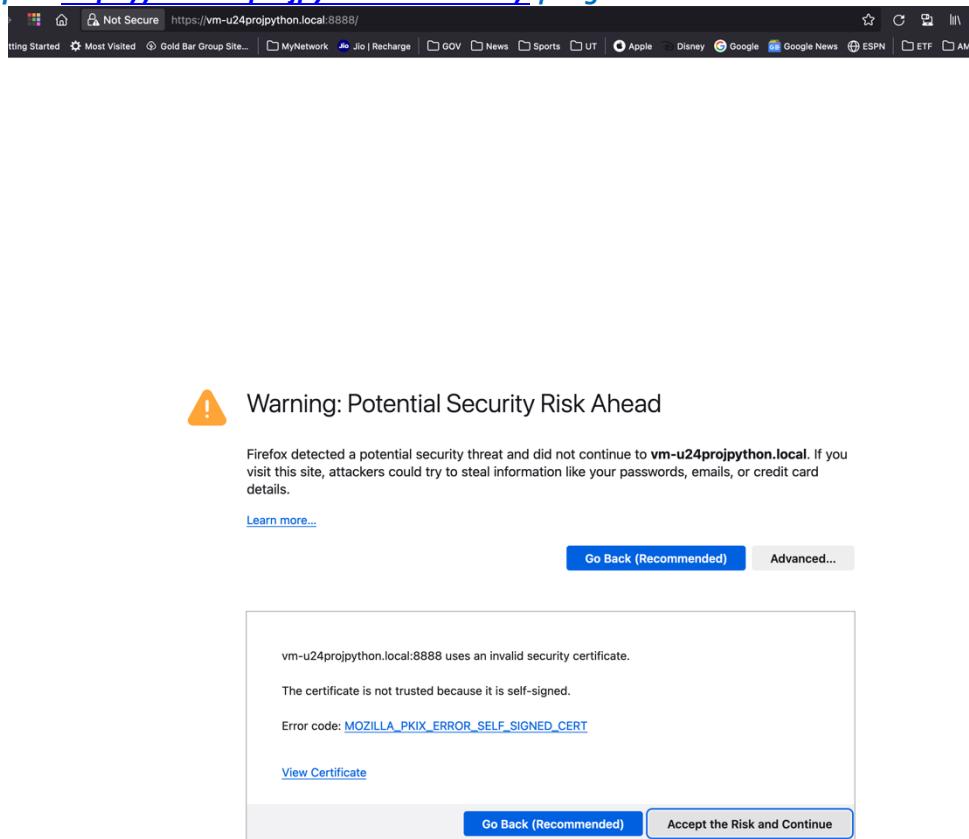
10. Check to service is running successfully after restart the machine and the version of Jupyter Notebook install, use below command:

```
$ sudo systemctl status jupyter_notebook.service
(base) sysadmin@vm-u24projpython:~$ sudo systemctl daemon-reload
(base) sysadmin@vm-u24projpython:~$ sudo systemctl enable jupyter_notebook.service
(base) sysadmin@vm-u24projpython:~$ sudo systemctl start jupyter_notebook.service
(base) sysadmin@vm-u24projpython:~$ sudo systemctl status jupyter_notebook.service
● jupyter_notebook.service - Jupyter notebook
  Loaded: loaded (/etc/systemd/system/jupyter_notebook.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2024-02-07 22:53:36 UTC; 21s ago
```

```
$ jupyter --version
(base) sysadmin@vm-u24projpython:~$ jupyter --version
Selected Jupyter core packages...
IPython          : 8.21.0
ipykernel        : 6.29.1
ipywidgets       : 8.1.1
jupyter_client   : 8.6.0
jupyter_core     : 5.7.1
jupyter_server   : 2.12.5
jupyterlab       : 4.1.0
nbclient         : 0.9.0
nbconvert        : 7.15.0
nbformat         : 5.9.2
notebook         : 7.0.7
qtconsole        : 5.5.1
traitlets        : 5.14.1
(base) sysadmin@vm-u24projpython:~$
```

**11.** Go the different computer and type the localhost:port number and login to see if jupyter is working sucessfully.

**Example : <https://vm-u24projpython.local:8888/> | Login Pass = Jn2024!**



**12.** Open the nginx.conf file using below command:

```
$ sudo nano /etc/nginx/nginx.conf
(base) sysadmin@vm-u24projpython:~$ sudo nano /etc/nginx/nginx.conf
[sudo] password for sysadmin:
(base) sysadmin@vm-u24projpython:~$
```

**13.** Disable compression gzip feature in the Nginx server "nginx.conf" file, to speed up the jupyter or any web application, by updating from on to off below command:

**gzip off**

<u>BEFORE</u>	<u>AFTER</u>
<pre>keepalive_timeout 65; #gzip on; include /etc/nginx/conf.d/*.conf; }</pre>	<pre>keepalive_timeout 65; gzip off; include /etc/nginx/conf.d/*.conf; }</pre>

Next **CMD+O or CTRL+O & CMD+X or CTRL+X** to save and close

14. Create **jupyter.conf** config file in Nginx, by using below command;

```
$ sudo nano /etc/nginx/conf.d/jupyter.conf
```

Copy and paste the below information:

```
upstream Keepalive-jupyter {
    server vm-u24projpython.local:8888;
    keepalive 86400;
}

server {
    #listen      80;
    # ..NGINX - Configuring HTTPS servers module...
    listen      443 ssl;
    listen      [::]:443 ssl;

    server_name vm-u24projpython.local;

    # ..NGINX - Configuring HTTPS servers module...
    ssl_certificate /etc/ssl/localhost/nginx.crt;
    ssl_certificate_key /etc/ssl/localhost/nginx.key;
    ssl_protocols TLSv1.2 TLSv1.1 TLSv1;
    ssl_ciphers      HIGH:!aNULL:!MD5;

    location /JN {
        # ..NGINX - ngx_http_proxy_module (proxy_pass)
        proxy_pass https://vm-u24projpython.local:8888;
        # ..NGINX - ngx_http_proxy_module (proxy_set_header)
        proxy_set_header Host $host;
        # ..NGINX - ngx_http_realip_module
        proxy_set_header X-Real-IP $remote_addr;
        # ..NGINX - ngx_http_proxy_module (Embedded Variables)
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        # ..NGINX - Accepting the PROXY Protocol
        proxy_set_header X-Forwarded-Proto $scheme;

        # ..NGINX - WebSocket Proxying
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }
}
```

```
GNU nano 6.2          /etc/nginx/conf.d/jupyter.conf *
upstream Keepalive-jupyter {
    server vm-u24projpython.local:8888;
    keepalive 86400;
}

server {
    #listen      80;
    # ..NGINX - Configuring HTTPS servers module...
    listen      443 ssl;
    listen      [::]:443 ssl;

    server_name vm-u24projpython.local;

    # ..NGINX - Configuring HTTPS servers module...
    ssl_certificate /etc/ssl/localhost/nginx.crt;
    ssl_certificate_key /etc/ssl/localhost/nginx.key;
    ssl_protocols TLSv1.2 TLSv1.1 TLSv1;
    ssl_ciphers      HIGH:!aNULL:!MD5;

    location /JN {
        # ..NGINX - ngx_http_proxy_module (proxy_pass)
        proxy_pass https://vm-u24projpython.local:8888;
        # ..NGINX - ngx_http_proxy_module (proxy_set_header)
        proxy_set_header Host $host;
        # ..NGINX - ngx_http_realip_module
        proxy_set_header X-Real-IP $remote_addr;
        # ..NGINX - ngx_http_proxy_module (Embedded Variables)
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        # ..NGINX - Accepting the PROXY Protocol
        proxy_set_header X-Forwarded-Proto $scheme;

        # ..NGINX - WebSocket Proxying
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }
}

^G Help      ^O Write Out   ^W Where Is   ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File   ^R Replace    ^U Paste     ^J Justify   ^L Go To Line
```

Next **CMD+O** or **CTRL+O** & **CMD+X** or **CTRL+X** to save and close

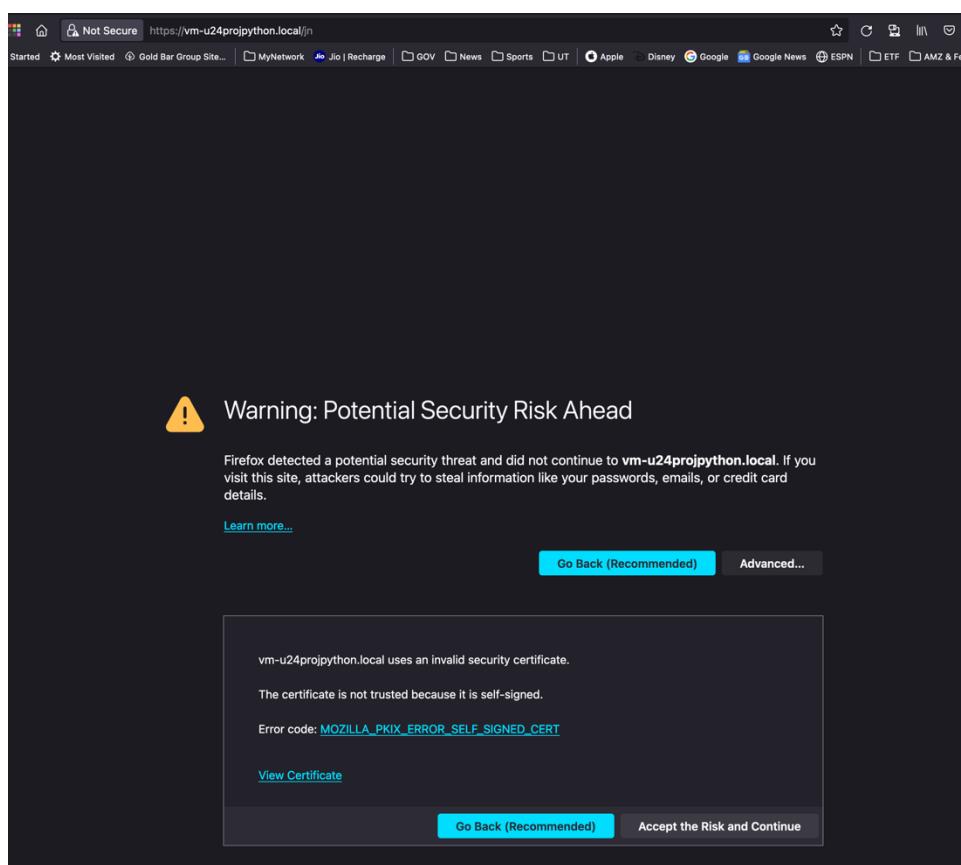
15. To take effect created config file in Nginx, restart the nginx server by using below command:

```
$ sudo systemctl restart nginx
```

```
(base) sysadmin@vm-u24projpython:~$ sudo systemctl restart nginx  
(base) sysadmin@vm-u24projpython:~$
```

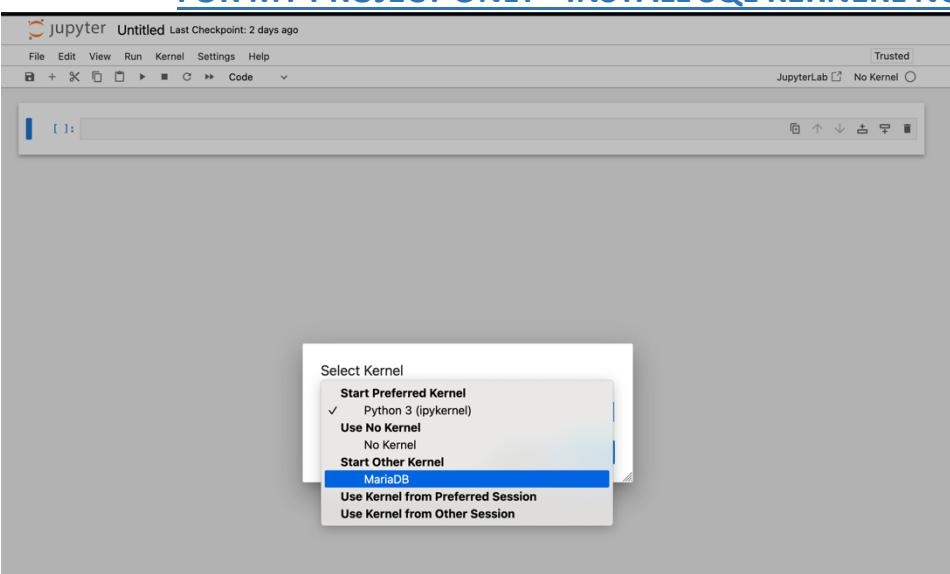
*Now the new url through ngnix will be <https://vm-u24projpython.local/JN>*

16. Verify that new url <https://vm-u24projpython.local/JN> is working by checking in different computer, it should show like below screenshot.



Note : The password is **Jn2024!**, as we created on the above steps.

#### **FOR MY PROJECT ONLY - INSTALL SQL KERNERL NOTEBOOK**



Use below command to install and enable, then login to jupyter to create notebook under MariaDB SQL.

```
$ python3 -m pip install mariadb_kernel
```

```
$ python3 -m mariadb_kernel.install
```

```
(base) sysadmin@vm-u24projpython:~$ python3 -m pip install mariadb_kernel  
Collecting mariadb_kernel  
  Successfully installed PyMySQL-1.1.0 cli-helpers-2.3.1 configobj-5.0.8 cryptography-36.0.2 json2html-1.3.0 mariadb_kernel-0.2.0 mycli-1.27.0 pyaes-1.6.1 pyperclip-1.8.2 setuptools-scm-8.0.4 sqlglot-21.2.1 sqlparse-0.4.4  
(base) sysadmin@vm-u24projpython:~$ python3 -m mariadb_kernel.install  
Installing Jupyter kernel spec  
(base) sysadmin@vm-u24projpython:~$
```

#### **Reference URL**

- Main : Anaconda
  - <https://docs.anaconda.com/free/anaconda/install/linux/>

- <https://docs.anaconda.com/free/anaconda/install/verify-install/>
- <https://www.rosehosting.com/blog/how-to-install-anaconda-on-ubuntu-20-04/>
- Troubleshoot : <https://www.danfry.net/technology/development/troubleshooting-anaconda-installation-issues-on-debian-12/>
- Troubleshoot (SUPERSEDED + DOWNGRADE) : <https://github.com/conda/conda/issues/2898> + (AutoUpdateConda) <https://conda.io/projects/conda/en/latest/user-guide/configuration/use-condarc.html> + (ChannelPriority) <https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-channels.html>
- Anaconda Packages :
  - <https://docs.anaconda.com/free/anaconda/packages/install-packages/>
  - <https://docs.anaconda.com/free/anaconda/reference/packages/pkg-docs/>
- Anaconda Commands :
  - All Commands List : <https://docs.conda.io/projects/conda/en/latest/commands/index.html#>
  - Update Package Command = <https://docs.anaconda.com/free/anaconda/install/update-version/>
  - Package search and install = <https://docs.conda.io/projects/conda/en/latest/user-guide/concepts/pkg-search.html>
  - Cheat Sheet = <https://conda.io/projects/conda/en/latest/user-guide/cheatsheet.html>
- Anaconda Managing :
  - Config File = <https://conda.io/projects/conda/en/latest/user-guide/configuration/use-condarc.html>
  - Environments = <https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html>
  - Channels = <https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-channels.html>
  - Packages = <https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-pkgs.html>
- Main : MongoDB
  - <https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-ubuntu/#std-label-install-mdb-community-ubuntu>
  - <https://www.mongodb.com/community/forums/t/installing-mongodb-over-ubuntu-22-04/159931/86?page=5>
- Python Ubuntu Install : <https://phoenixnap.com/kb/how-to-install-python-3-ubuntu>
- Ubuntu Version Check: <https://askubuntu.com/questions/1353788/cannot-install-anaconda-in-ubuntu-20-04>
- Jupyter Notebook Official Install Guide : <https://docs.jupyter.org/en/latest/install/notebook-classic.html>
- Jupyter Notebook Official Syntax Guide : <https://jupyter-notebook.readthedocs.io/en/master/config.html>
- Jupyter Notebook System Service Guide: <https://medium.com/@desjoerdhaan/manage-your-jupyter-server-with-systemd-9914ff39049d>
- Jupyter Notebook Config file Guide : [https://medium.com/@nicklas\\_bocksberger/setting-up-jupyter-notebook-on-a-server-with-nginx-as-proxy-d579d4075bb](https://medium.com/@nicklas_bocksberger/setting-up-jupyter-notebook-on-a-server-with-nginx-as-proxy-d579d4075bb)
- Jupyter Official Config Guide : <https://jupyter-notebook.readthedocs.io/en/master/config.html>
- Jupyter 3rd Party Password & Config Guide : <https://gist.github.com/OsamaMahmood/2cf83a4d056707843f443baa38483f52>
- Jupyter Service Creation Guide 1 : <https://decovar.dev/blog/2020/05/27/jupyter-notebook-nginx/>
- Jupyter Service Creation Guide 2 : <https://gist.github.com/mkraemerx/b16298087fa4cee95a1d4be0dd52fef2>
- Jupyter SSL Certificate Addon : <https://guruatwork.medium.com/jupyter-notebook-adding-certificates-for-ease-of-use-447f476b9112>
- Jupyter Version Check : <https://stackoverflow.com/questions/47662692/how-can-i-display-the-version-of-my-jupyter-notebook-and-run-cells-in-jupyter-no>
- NGINX - ngx\_http\_proxy\_module (proxy\_pass) : [http://nginx.org/en/docs/http/ngx\\_http\\_proxy\\_module.html#proxy\\_pass](http://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_pass)
- NGINX - ngx\_http\_proxy\_module (proxy\_set\_header) :
  - [http://nginx.org/en/docs/http/ngx\\_http\\_proxy\\_module.html#proxy\\_set\\_header](http://nginx.org/en/docs/http/ngx_http_proxy_module.html#proxy_set_header)
- NGINX - ngx\_http\_realip\_module : [http://nginx.org/en/docs/http/ngx\\_http\\_realip\\_module.html](http://nginx.org/en/docs/http/ngx_http_realip_module.html)
- NGINX - ngx\_http\_proxy\_module (Embedded Variables) : [http://nginx.org/en/docs/http/ngx\\_http\\_proxy\\_module.html#variables](http://nginx.org/en/docs/http/ngx_http_proxy_module.html#variables)
- NGINX - Accepting the PROXY Protocol : <https://docs.nginx.com/nginx/admin-guide/load-balancer/using-proxy-protocol/>
- NGINX - WebSocket Proxying : <http://nginx.org/en/docs/http/websocket.html>
- Nginx - Jupyter \*.Conf Working Example : <https://discourse.jupyter.org/t/trouble-reverse-proxying-to-jupyterhub-on-a-subpath/13146/12>
- Nginx - Jupyter \*.Conf (c.Notebook.default\_url) : <https://github.com/jupyterlab/jupyterlab/issues/8656>
- Nginx - Jupyter \*.Conf Example 1 : <https://github.com/jupyter/notebook/issues/625>
- Nginx - Jupyter \*.Conf Example 2 : <https://stackoverflow.com/questions/48229548/cant-serve-jupyter-notebook-with-nginx-reverse-proxy>
- Nginx - Jupyter \*.Conf Example 3 : <https://gist.github.com/dgacitua/0fd03cb63b34ffa9c28bcb0551fdb4a7>
- Nginx - Jupyter \*.Conf Example 4 : [https://gitlab.com/mikbuch/nginx-jupyter-ubuntu/-/blob/main/jupyter-notebook.conf?ref\\_type=heads](https://gitlab.com/mikbuch/nginx-jupyter-ubuntu/-/blob/main/jupyter-notebook.conf?ref_type=heads)
- Kernel - MariamDB SQL Example : [https://github.com/MariaDB/mariadb\\_kernel](https://github.com/MariaDB/mariadb_kernel)
- Kernel - Additional you can install : <https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>
- Kernel - Jupyter Main : <https://docs.jupyter.org/en/latest/install/kernels.html>

## OFFLINE RECOMMEND

### OPTIONS COVERING

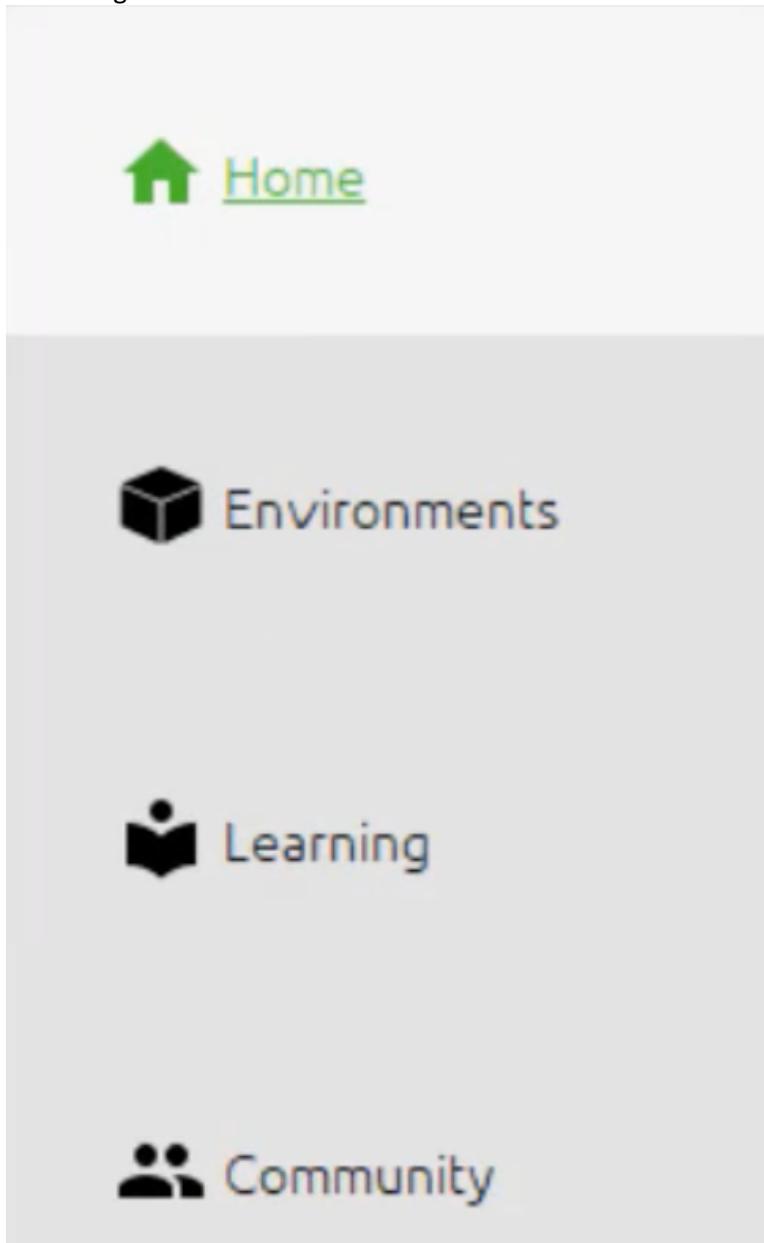
1. **Anaconda Navigator**
2. **Jupyter Notebook**
3. **GitHub Jupyter Notebook Course resources glance, see below url :**
  - a. Python 3 Notebooks can be found here:  
<https://github.com/Pierian-Data/Complete-Python-3-Bootcamp>
  - b. The older Python 2 Notebooks can be found here:  
<https://github.com/jimportilla/Complete-Python-Bootcamp>

### ANACONDA NAVIGATOR TOUR

#### TOUR

Note: Understanding left panel as per below information and image.

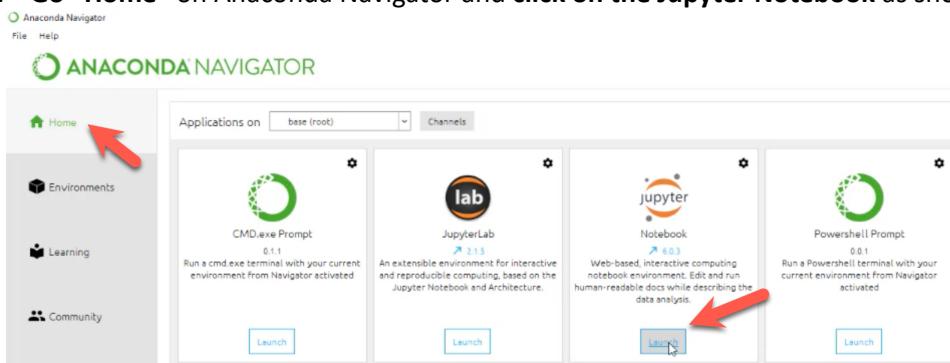
1. **Home** = It is the applications launch location, where various application you be able to launch. For example **Jupyter Notebook** for lab.
2. **Environments** = It is the virtual environment, where you can install particular version of libraries. This will be also explain later in course, but for now **base (root)** is permanent install in computer and other environment you create is temporary for projects.
3. **Learning** = It is knowledge area, where you will find links to books, documents and training videos for learning. Also it is heavily leaning towards data science and machine learning.
4. **Community** = It is area where you will find bunch of links for events and forums, which is skewed towards data science and machine learning.



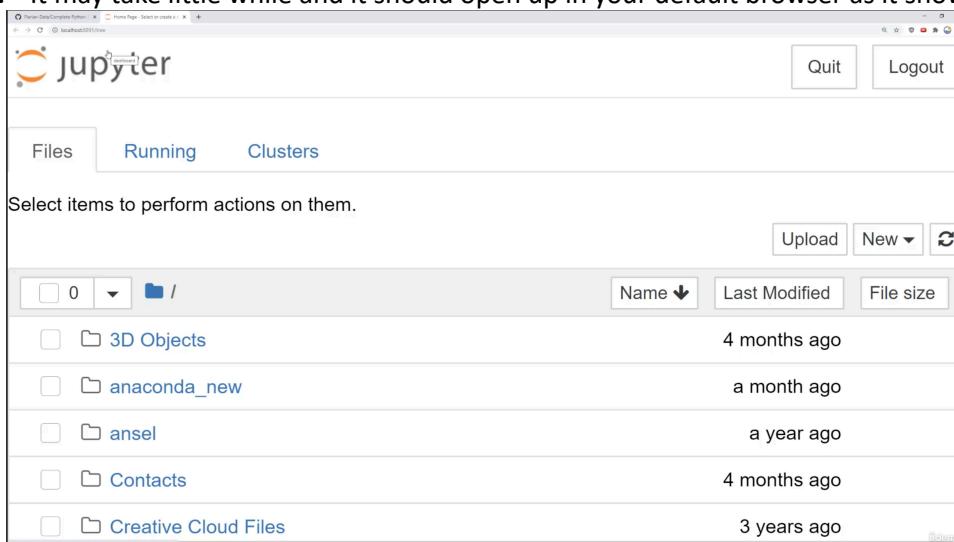
\*\*\*\*\*END OF ANCONDA NAVIGATOR SECTIONS\*\*\*\*\*

## JUYPTER NOTEBOOK TOUR

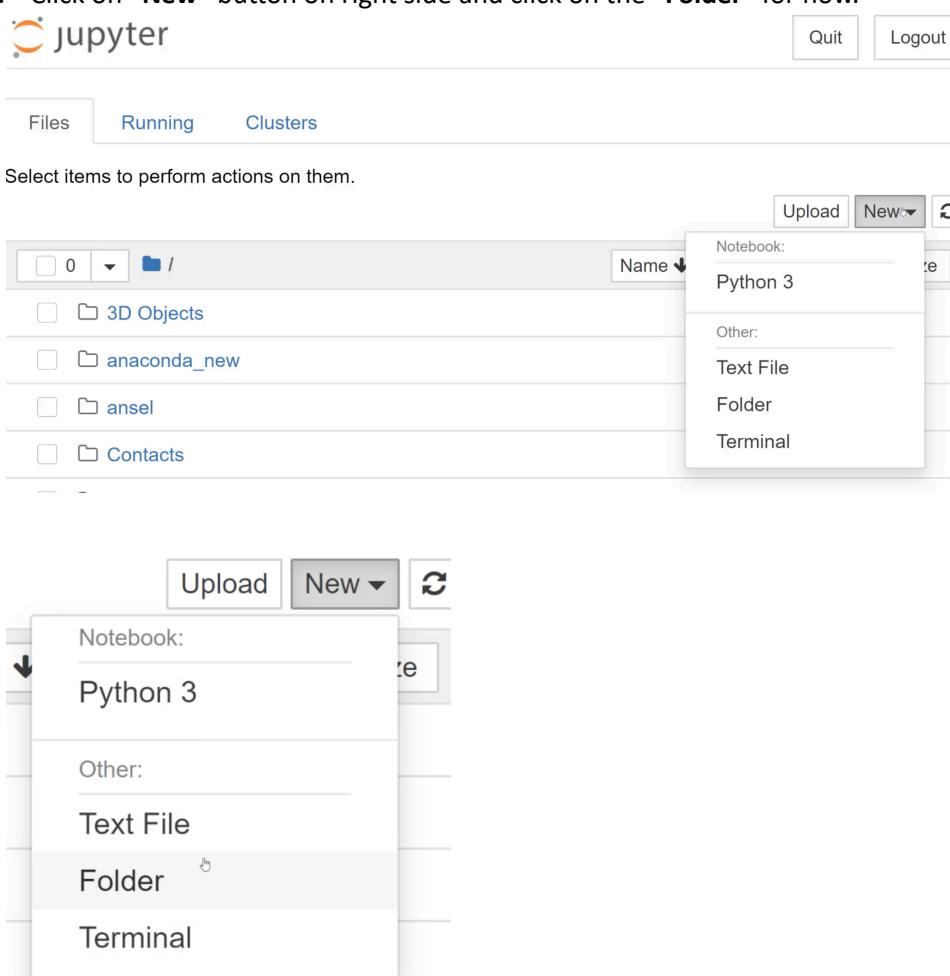
1. Go "Home" on Anaconda Navigator and click on the Jupyter Notebook as shown in below image.



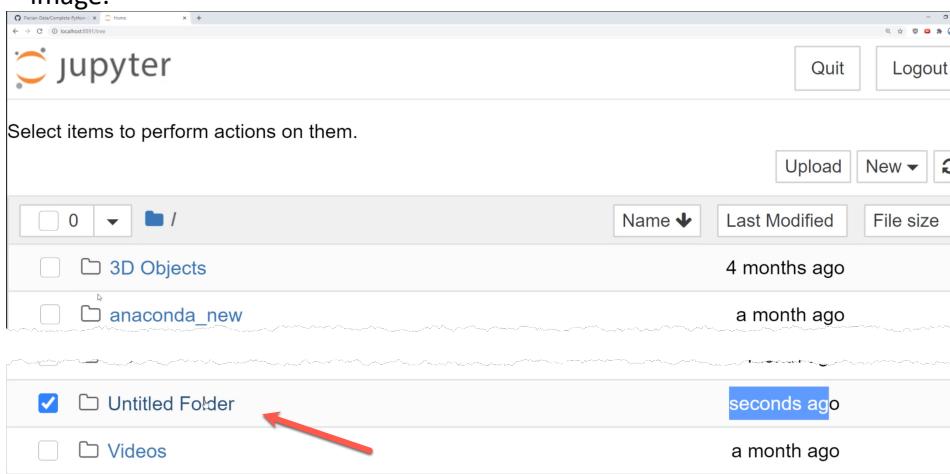
2. It may take little while and it should open up in your default browser as it shown in below image.



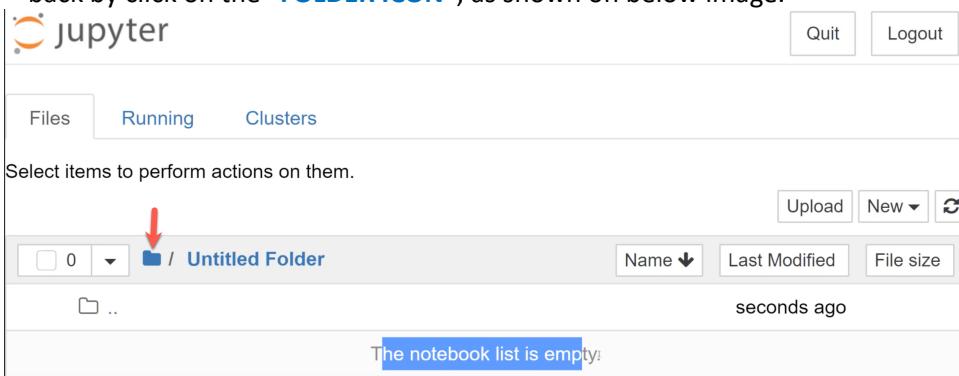
3. Click on "New" button on right side and click on the "Folder" for now.



4. Next you will see "Untitled folder" when you scroll down on list of folder and click on the "Untitled folder" folder as shown in below image.



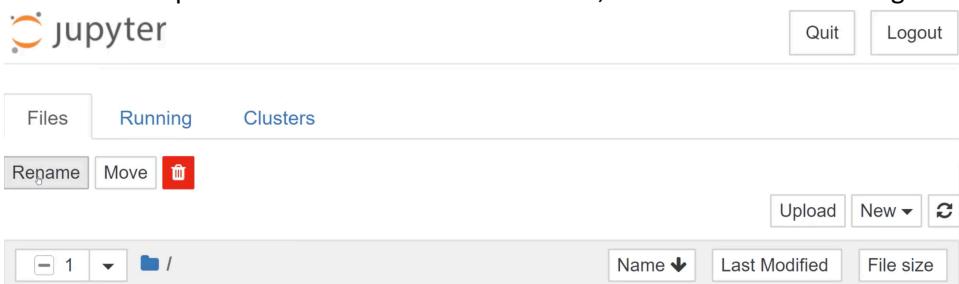
5. Next in the "Untitled folder" folder and you will see it is empty. This where you where allow to create notebook inside. For now go back by click on the "**FOLDER ICON**", as shown on below image.



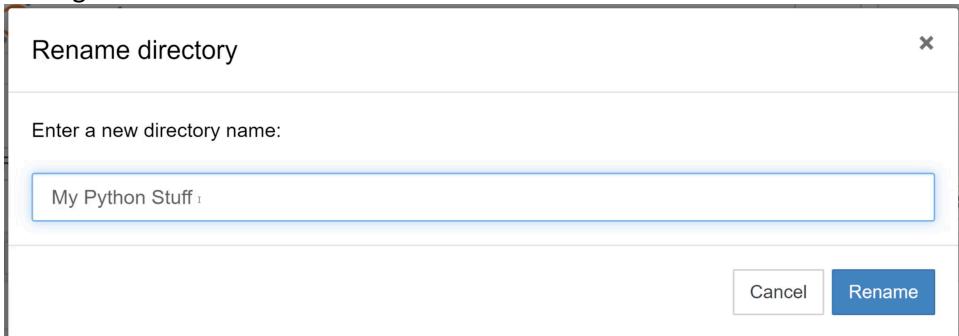
6. Next it will bring you back to previous location, scroll down to "Untitled folder" folder and **place check mark** in box for rename, as shown on below image.



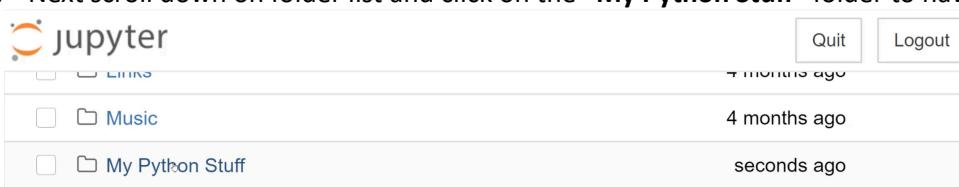
7. Next scroll up and click on the "Rename" button, as shown on below image.



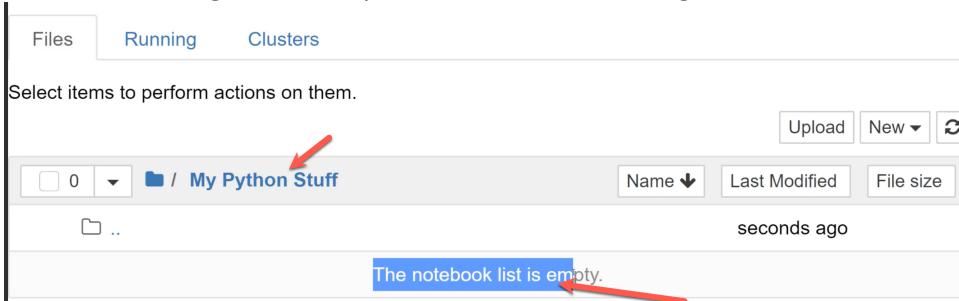
8. Then **box open up** to rename. So just rename to "**My Python Stuff**" for now and click on the "**Rename**" button, as shown below in image.



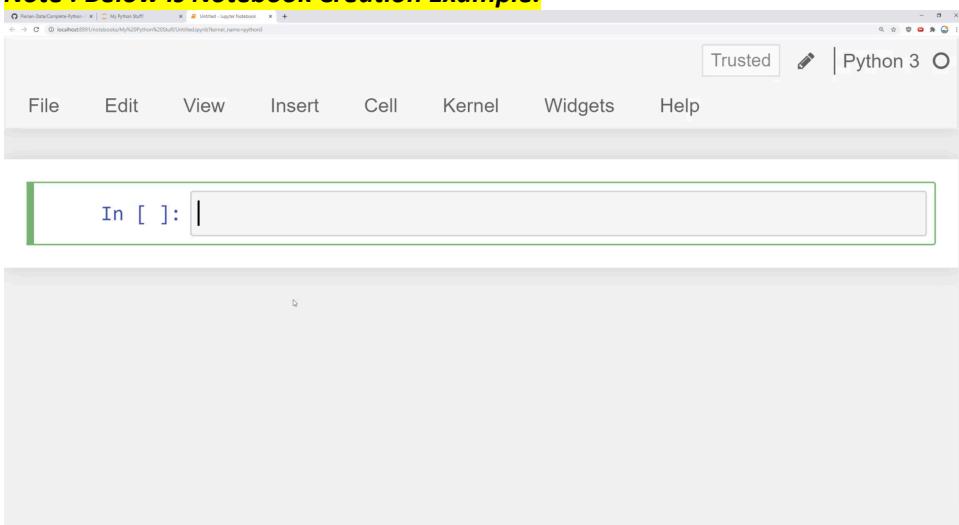
9. Next scroll down on folder list and click on the "**My Python Stuff**" folder to navigate inside, as shown below:



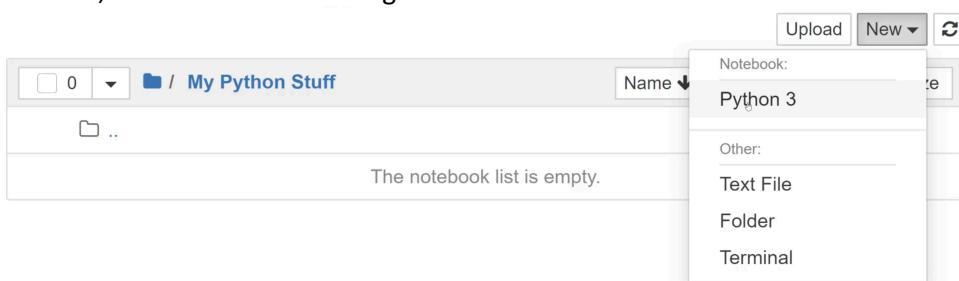
Note : Below image will show you that folder name change from "**Untitled folder**" the to "**My Python Stuff**".



**Note : Below is Notebook Creation Example.**



- 10.** Next click on the button "New" -> "Python 3" under "Notebook:" sections, at top right corner to create your first notebook in this folder, as shown in below image.

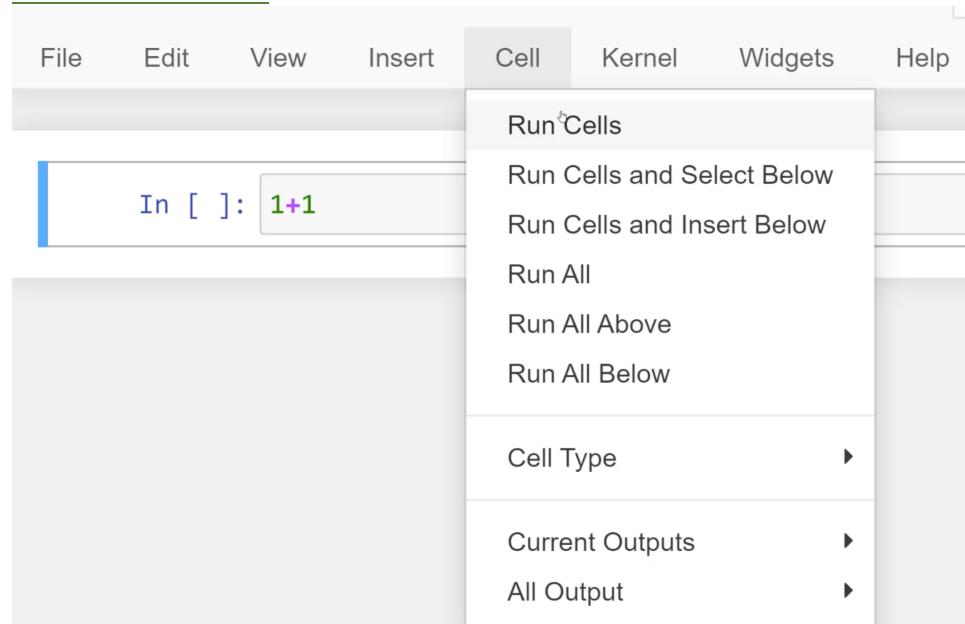


- 11.** Next quick tour on usage of python notebook to run python code. First python code going to use is to **type "1+1"** math equation and go on menu "Cell" -> "Run Cells" or keyboard shortcut "Shift + Enter" to do calculation.

#### PYTHON CODE:

```
In [ ]: 1+1
```

#### RUNNING THE CODE:



#### RESULT:

```
In [1]: 1+1
Out[1]: 2
```

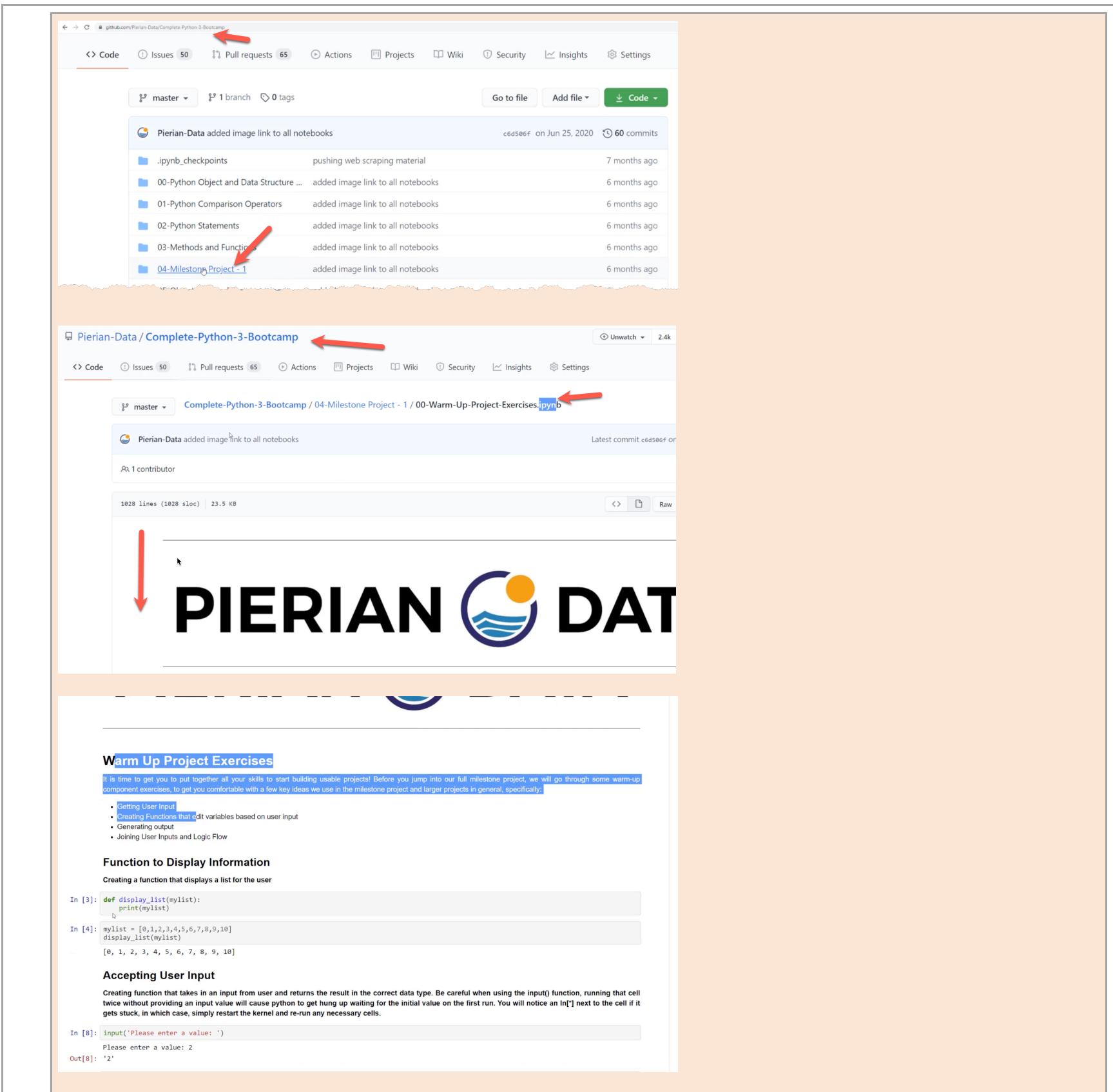
**SHIFT + ENTER:** To run the code, produce result and create new empty line for next code. Also # mean just note not code, so in this case # Shift + Enter is Shift + Enter is a text only not a code to run.

```
In [2]: 1+1 # Shift + Enter
Out[2]: 2
In [ ]:
```

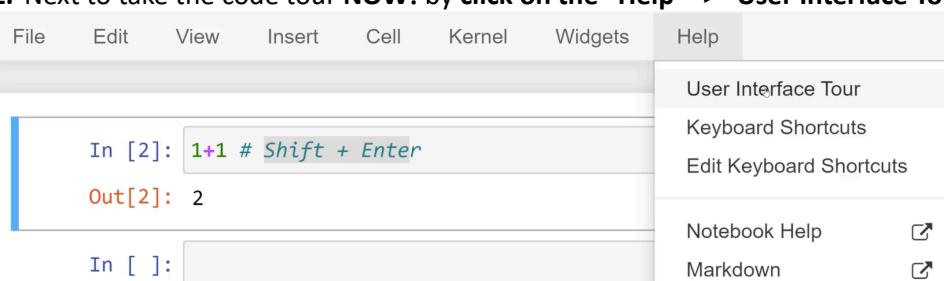
#### PYTHON CODE JUPYTER NOTEBOOK LOCATION

Later on course will show how to download the **Python Code Jupyter Notebooks**, which is located in following URL = <https://github.com/Pierian-Data/Complete-Python-3-Bootcamp>

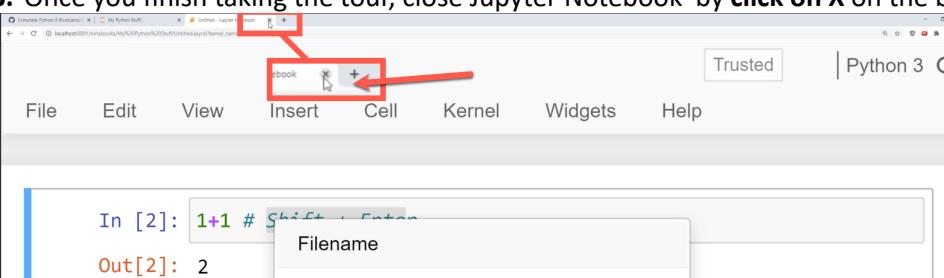
**Note :** Below is the example of .ipynb file located in above url that you can open up on Jupyter Notebook.



**12.** Next to take the code tour **NOW!** by click on the "Help" -> "User Interface Tour" , which is super useful.



**13.** Once you finish taking the tour, close Jupyter Notebook by click on X on the browsers, as shown in image below.



**14.** Next stop the running Jupyter Notebook by placing check on the box and click on the "Shutdown" button, as shown in below image.

**jupyter**

Files    Running    Clusters

Duplicate Shutdown View Edit 

Upload New

1 / My Python Stuff Name ↓ Last Modified File

seconds ago  
Running seconds ago

**Untitled.ipynb**

**RESULT:**

0 / My Python Stuff Name ↓ Last Modified File

seconds ago  
a minute ago

**Untitled.ipynb**

**15.** If you want to delete permanently the Jupyter Notebook, check on the box -> click on the "Trash" button -> "Delete" button on popup window, as shown on below image.

**Note : Same process for folder permanently delete, but only empty one that Jupyter allow.**

Files    Running    Clusters

Duplicate Rename Move Download View Edit 

Upload New 

1 / My Python Stuff Name ↓ Last Modified File size

seconds ago  
a minute ago 975 B

**Untitled.ipynb**

**Delete**  
Are you sure you want to permanently delete: "Untitled.ipynb"?  
Cancel Delete

**RESULT:**

Files    Running    Clusters

Select items to perform actions on them.

Upload New 

0 / My Python Stuff Name ↓ Last Modified File size

seconds ago

**\*\*\*\*\*END OF JUPYTOR NOTEBOOK SECTIONS\*\*\*\*\***

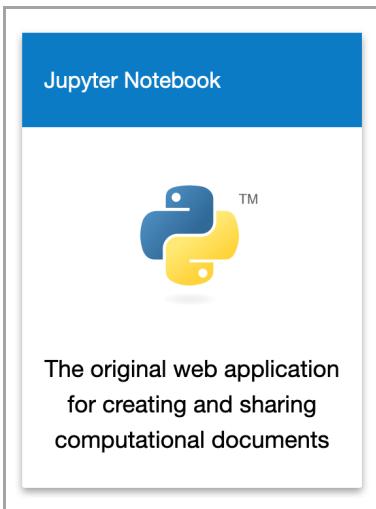
## ONLINE (NO INSTALL) NOT RECOMMEND

### OPTIONS COVERING

1. **Jupyter Lab** = <https://jupyter.org/try>
2. **Google Collab online** = <https://colab.research.google.com/>
3. **Rep.it "Python Interpreter Online"** = <https://replit.com/languages/online-python-compiler>

**Note : Any above option may not save code, may hard to upload your own code and it is not part of this course.**

1. **Jupyter Lab** =
  - a. Go the <https://jupyter.org/try> and click on the below icon for now, as to follow course.

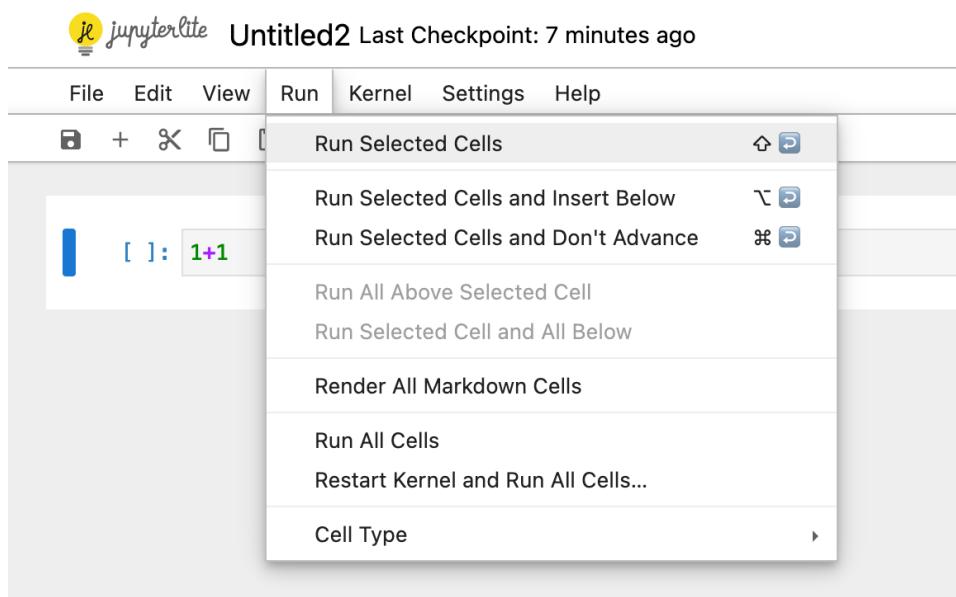
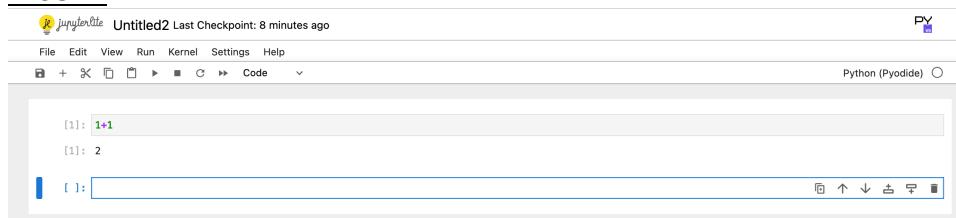


b. Next you will see window like below :

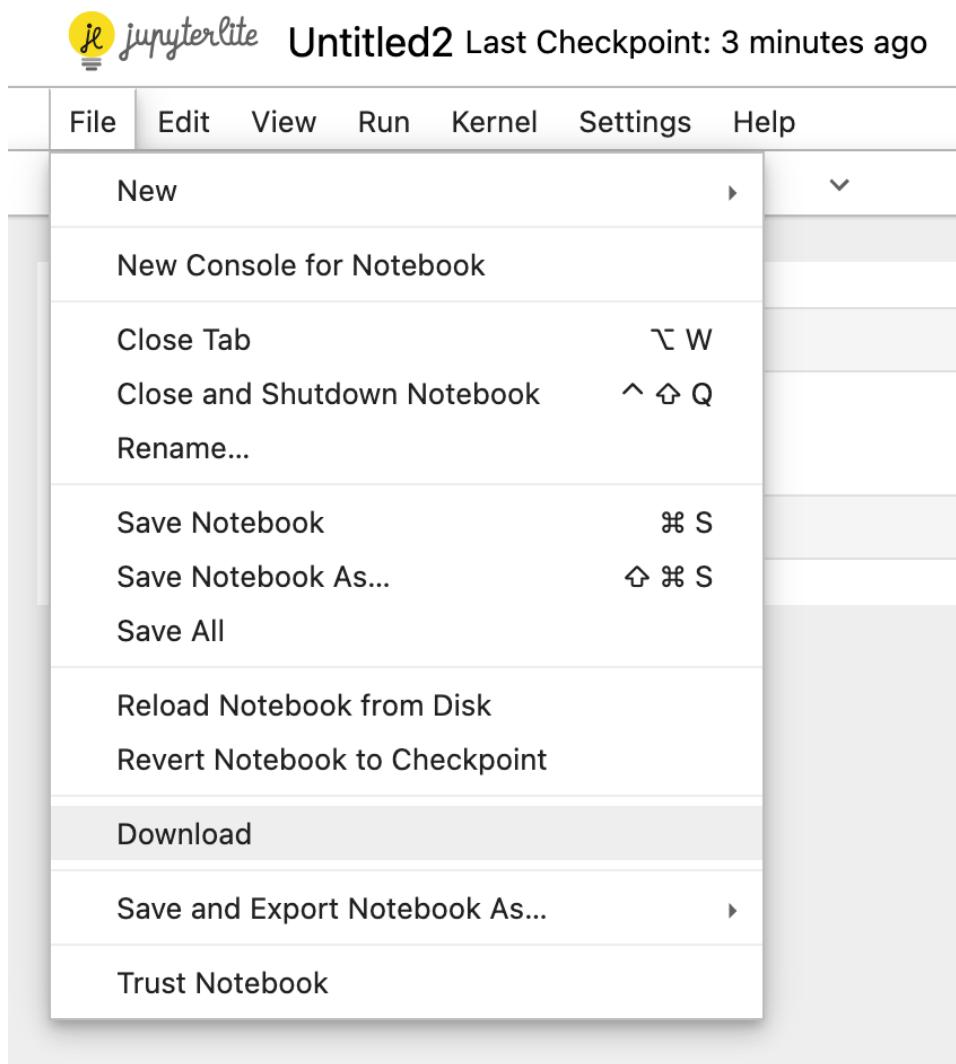
c. Next on the top menu Click on "File" -> "New" -> "Notebook" and on new tab, click on the "Select" button to open new notebook, as shown in below image.

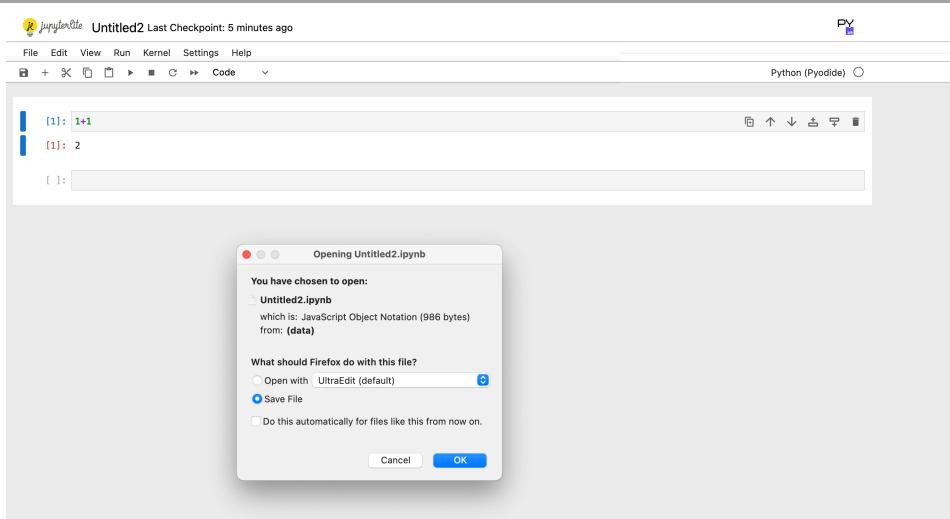
**NOTE : Memory part in video no longer in use as of today.**

d. Then type **1+1** in line and on menu, click on the "Run" -> "Run Selected Cells" to see result, as shown below:

**TYPE:****RUN:****RESULT:**

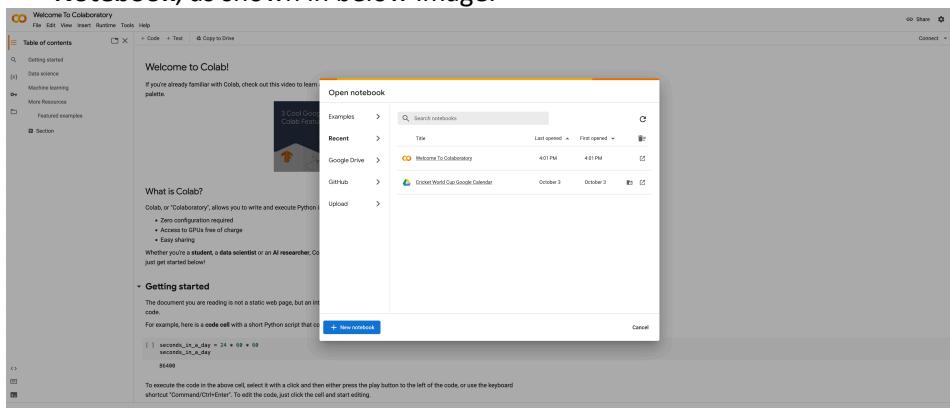
- e. As it doesn't save the notebook, you need to download. So to download on menu **Click on "File" -> "Download"** and **browser will ask saving location**, therefore choose your location and save, as per shown in image below:



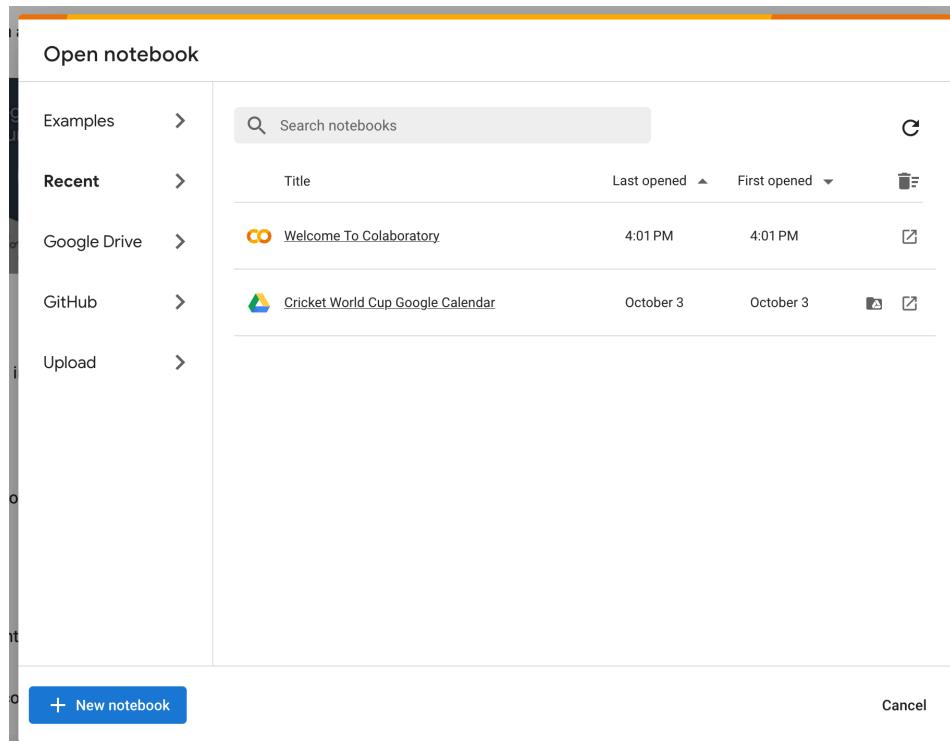


## 2. Google Collab online

- a. Go the url <https://colab.research.google.com/> and login with your gmail account. Which will show you **Google Collab Notebook**, as shown in below image:



- b. Next Click on the "New notebook" button on bottom to create notebook



- c. It will show new notebook, where **type 1+1 and click on "Runtime" -> "Run all"** as shown on image below:

### TYPE:

Untitled0.ipynb

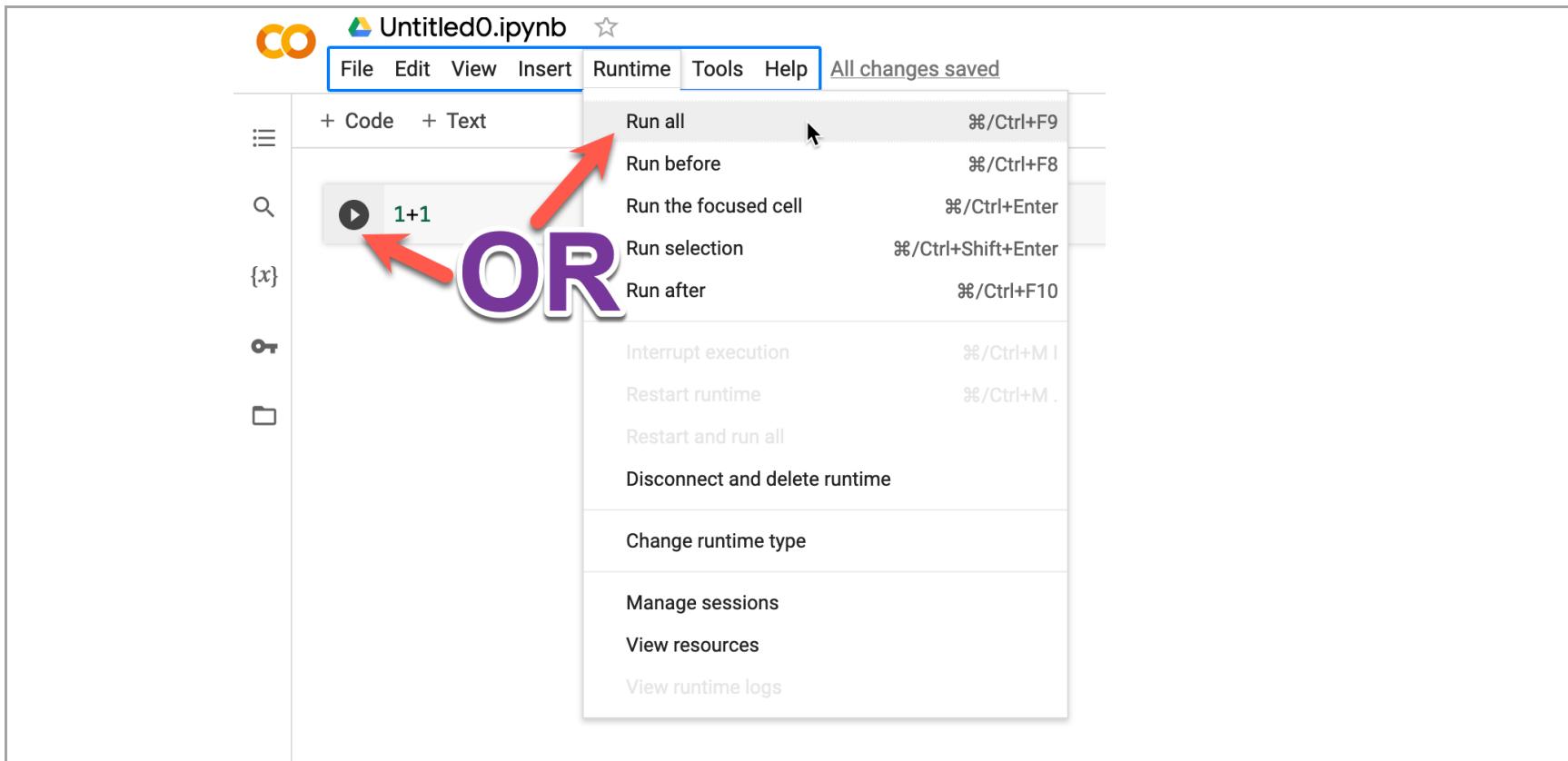
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

1+1

{x}

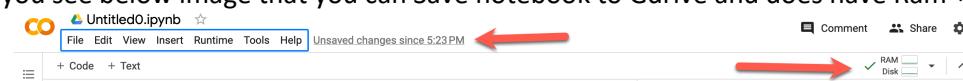
RUN:



**RESULT:**

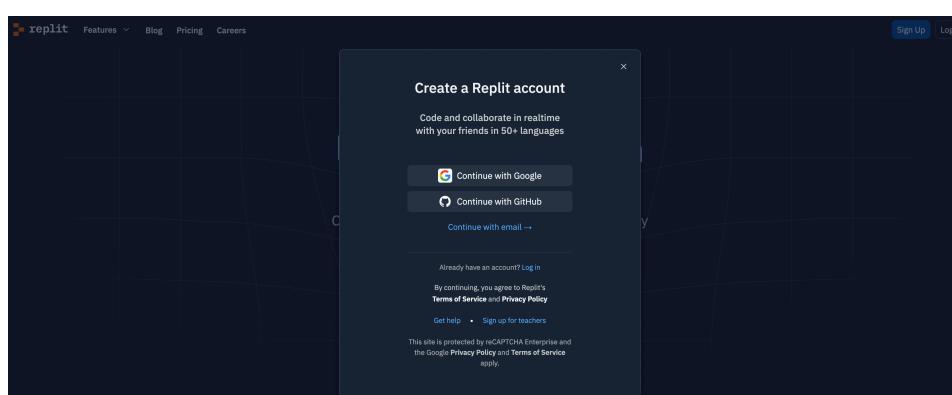
```
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
0s
{x}
2
```

- d. As you see below image that you can Save notebook to Gdrive and does have Ram + Disk limit.



**3. Rep.it "Python Interpreter Online"**

- Note this is for collaborating coding and it have cost to it.
- For price see URL = <https://replit.com/pricing>
- To follow video demo go following URL = <https://replit.com/languages/python3> and signup.



The screenshot shows the Replit workspace interface. A central modal window titled "Welcome to the Workspace" provides introductory information about the workspace. Below the modal, the main workspace area displays a file list on the left containing "main.py", "pyproject.toml", and "poetry.lock". On the right, there is a code editor with the following Python code:

```
1 print(1+1)
```

Below the code editor is a "Run" button. To the right of the code editor, a "Console" tab is open, showing the message: "Results of your code will appear here when you Run the project." At the top of the workspace, there are tabs for "Search", "Files", "Packager files", "Python", and "Run".

**TYPE:**

The code editor shows the input state of the "main.py" file, which contains the line "1 print(1+1)". The "Run" button is visible above the code.

**RUN:**

The workspace interface remains largely the same, but the "Run" button has been clicked, indicated by a small "Return" button appearing next to it. The "Console" tab now displays the output of the executed code: "2".

**RESULT:**

The workspace interface remains largely the same, but the "Run" button has been clicked, indicated by a small "Return" button appearing next to it. The "Console" tab now displays the output of the executed code: "2".