

Systems & Toolchains for AI Engineers: Windows Installation Guide

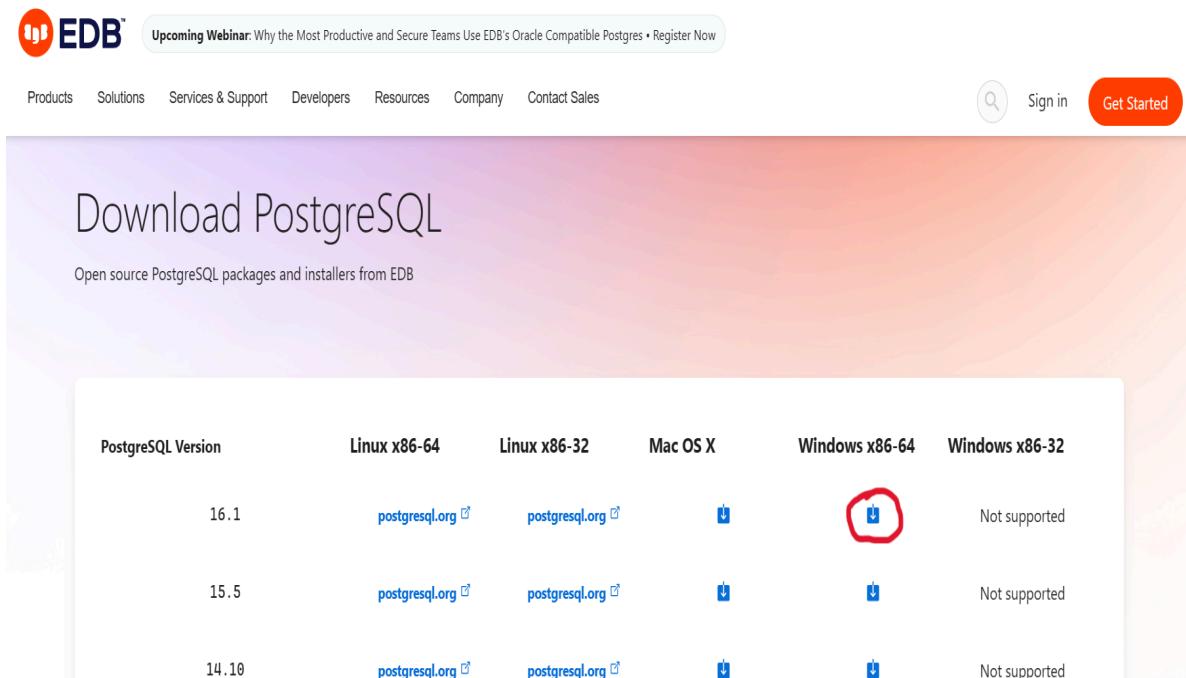
Disclaimer: this guide aims to help you get a high-level idea of required software installations. The provided steps in this guide may get outdated over time and/or may not match your exact operating system file hierarchy. Please use this document as a guide and not a strict document to follow.

PostgreSQL, PgAdmin

PostgreSQL is an open-source relational database management system, developed from the Ingres project and merged with SQL in 1996.

The PostgreSQL installation is as follows.

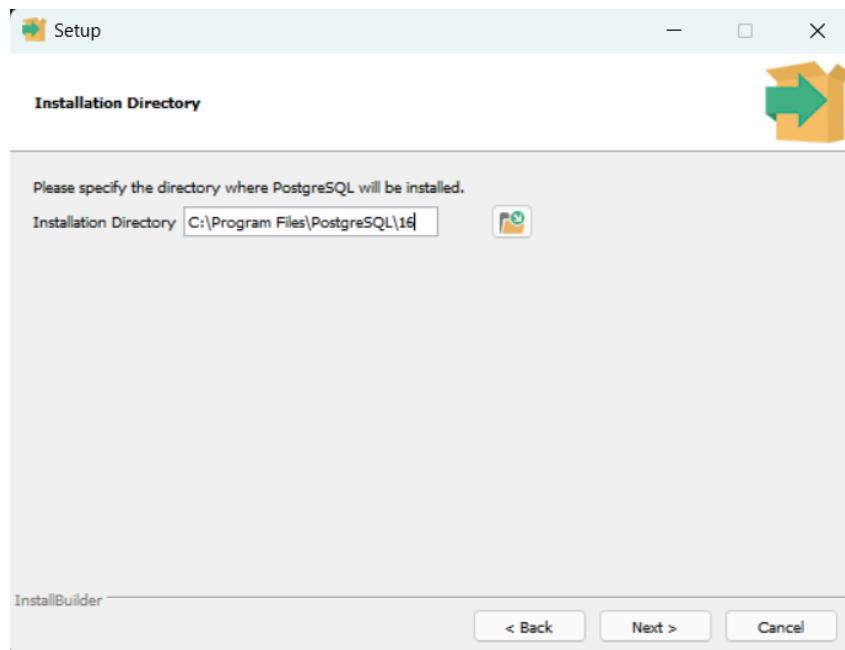
1. Download the latest installation package from
[Community DL Page \(enterprisedb.com\)](#)



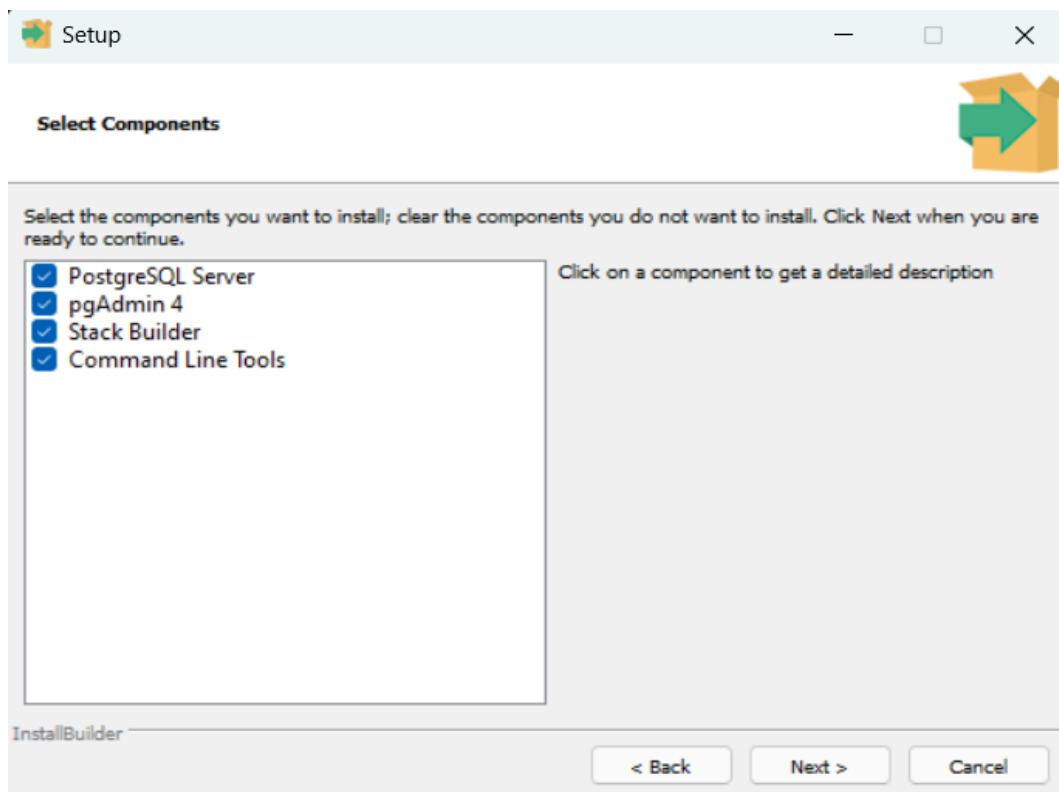
The screenshot shows the EDB PostgreSQL download page. At the top, there's a navigation bar with the EDB logo, a webinar announcement, and links for Products, Solutions, Services & Support, Developers, Resources, Company, Contact Sales, Sign in, and Get Started. The main heading is "Download PostgreSQL". Below it, a sub-header says "Open source PostgreSQL packages and installers from EDB". A table lists PostgreSQL versions (16.1, 15.5, 14.10) against supported platforms: Linux x86-64, Linux x86-32, Mac OS X, Windows x86-64, and Windows x86-32. The Windows x86-64 column contains download icons. The Windows x86-32 column for version 16.1 has a red circle around the download icon, with the note "Not supported" below it. Similar notes appear for other unsupported versions in the same column.

PostgreSQL Version	Linux x86-64	Linux x86-32	Mac OS X	Windows x86-64	Windows x86-32
16.1	postgresql.org	postgresql.org	postgresql.org	Download	Download Not supported
15.5	postgresql.org	postgresql.org	postgresql.org	Download	Download Not supported
14.10	postgresql.org	postgresql.org	postgresql.org	Download	Download Not supported

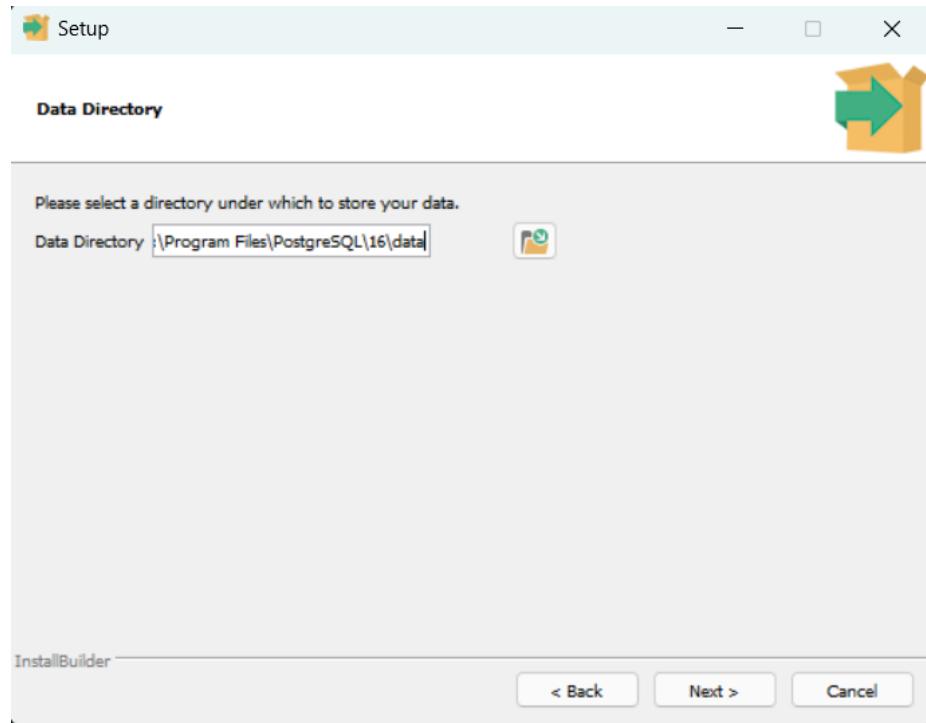
2. Open the setup wizard and follow the instructions.
 - a. Leave the directory in the default location.



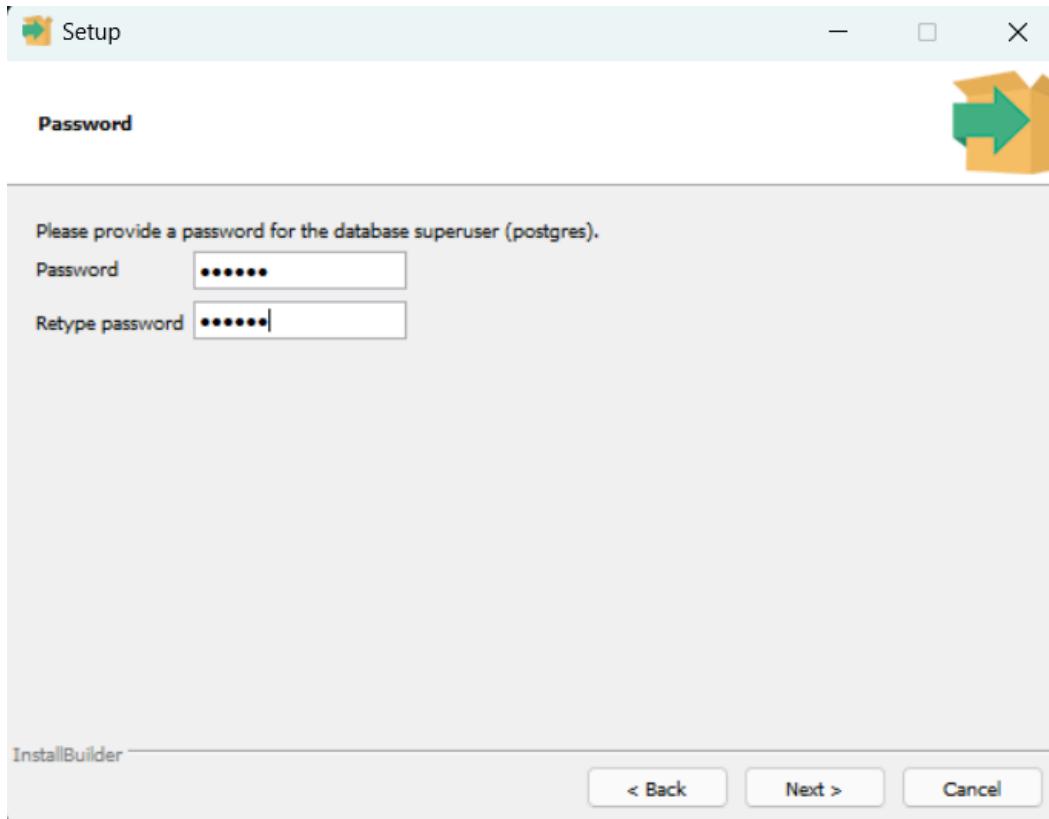
- b. Select the required components for installation. Leave everything checked, but ensure that PostgreSQL server and pgAdmin 4 are selected!



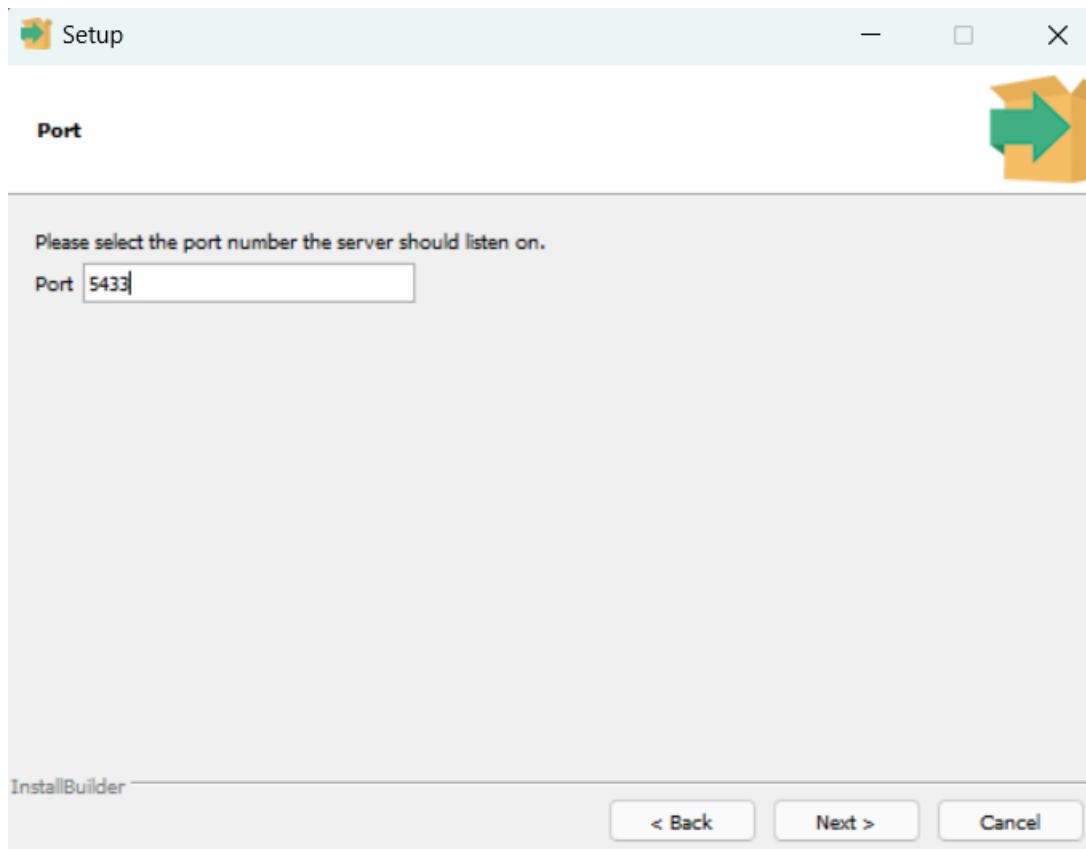
c. Leave the data in the default directory



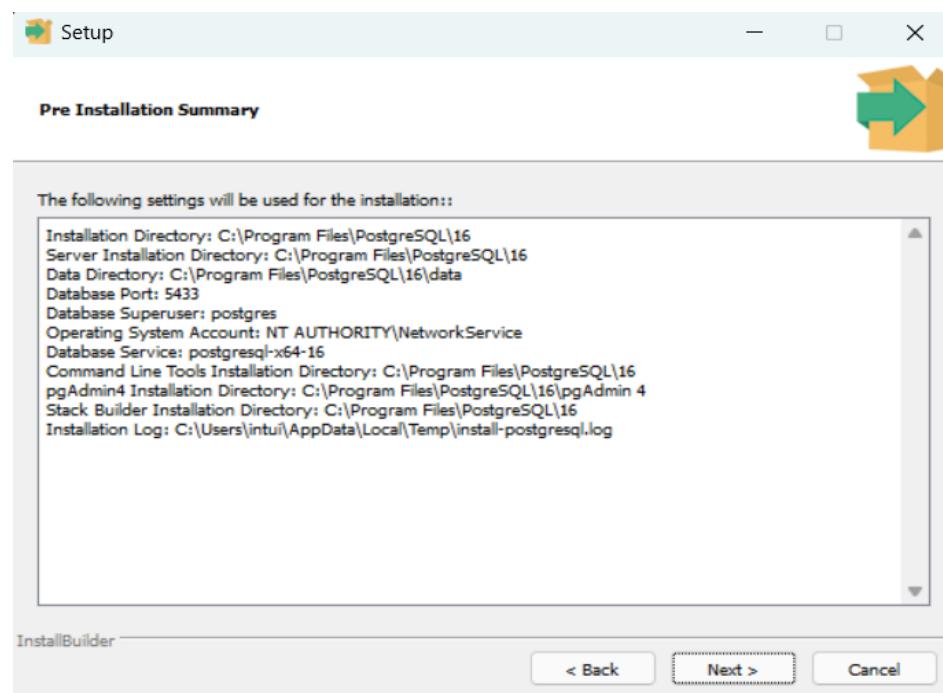
d. Set your superuser password



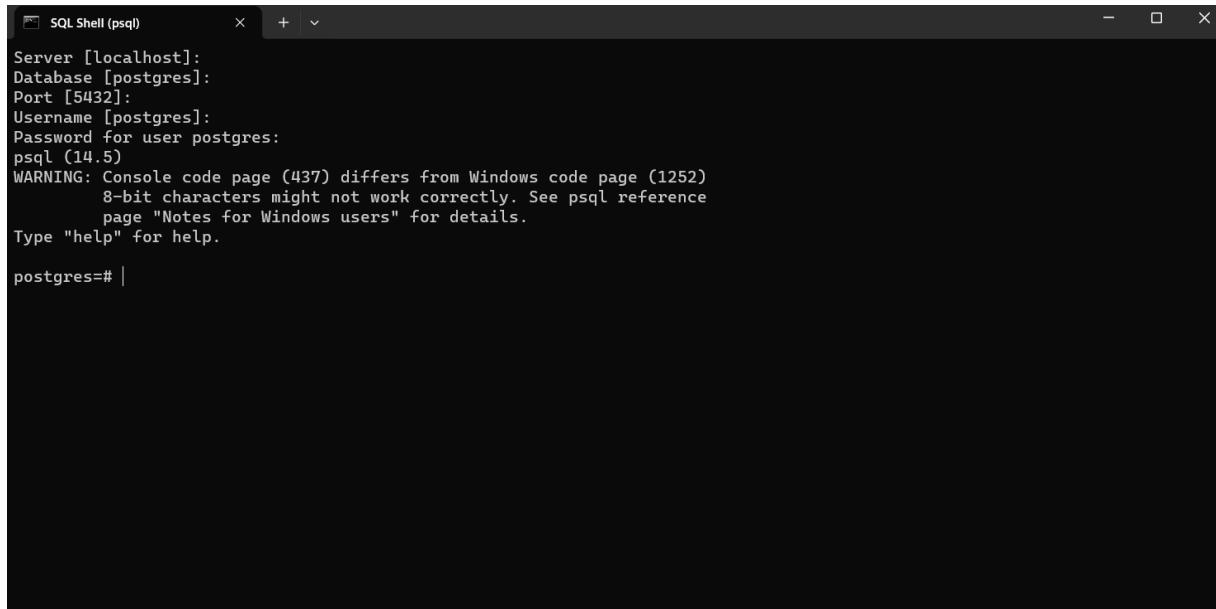
- e. Set the port to the default one suggested by the installation wizard.



- f. After verifying the pre installation summary, install the software



3. Open a psql shell and key in your superuser password to connect

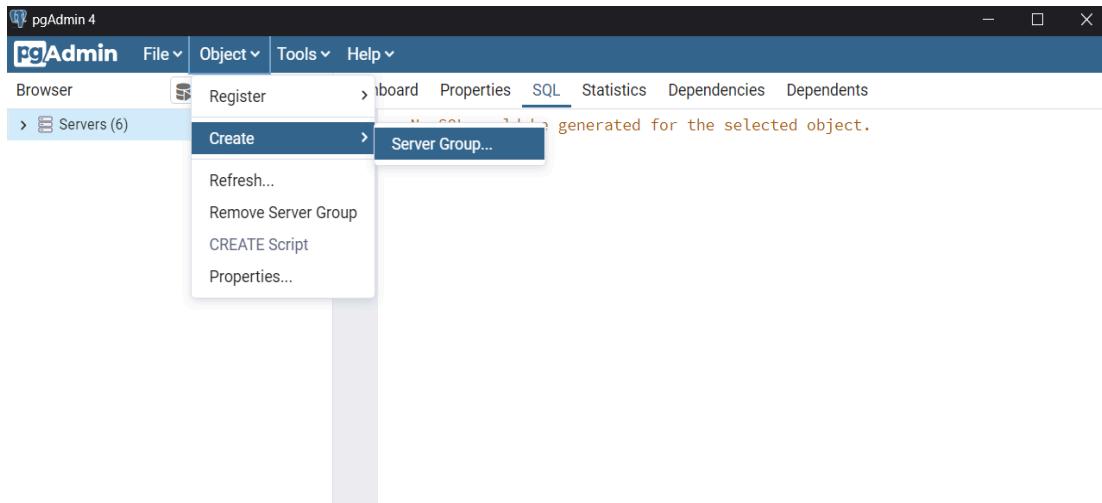


```
SQL Shell (psql)      X + v
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:
pgsql (14.5)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

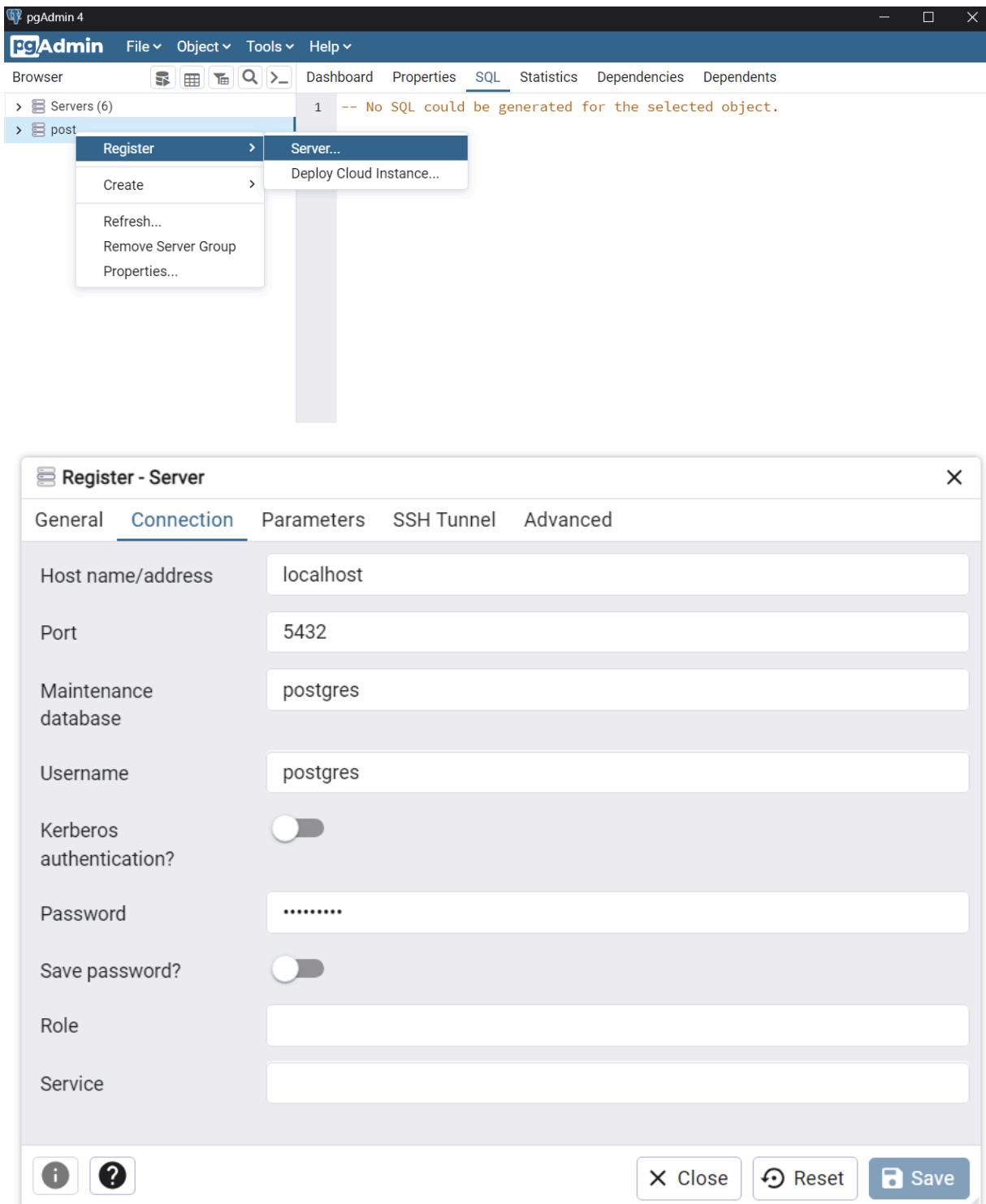
postgres=# |
```

4. If connecting through pgAdmin, there are a few steps to follow.

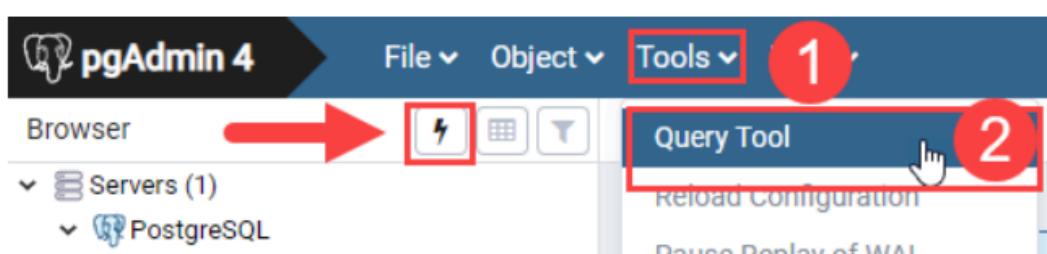
a. Create a new server group

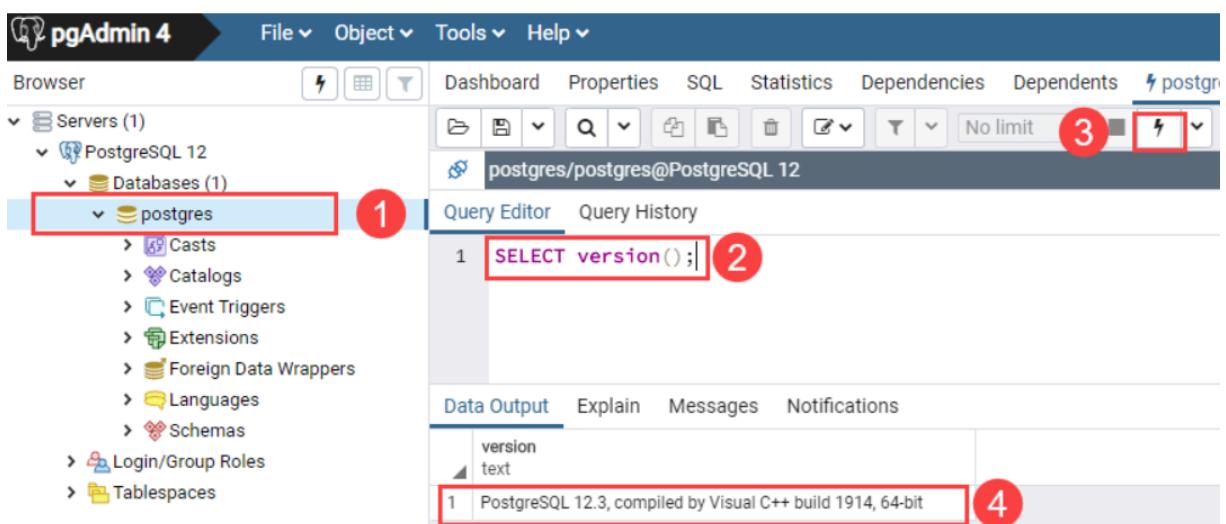


b. Create a new server



c. Open and execute query tool to conduct queries





Apache Spark

Apache Spark is an open-source unified analytics engine for large-scale data processing developed by AMPLab at UC Berkeley.

Install Java:

1. Download the JDK 11 or 17 from:[Oracle Website](#) and choose the installer file, where the arrow in the screenshot can be seen.

JDK 19 will receive updates under these terms, until March 2023 when it will be superseded by JDK 20.
JDK 17 will receive updates under these terms, until at least September 2024.

Java 19 Java 17

Java SE Development Kit 19.0.2 downloads

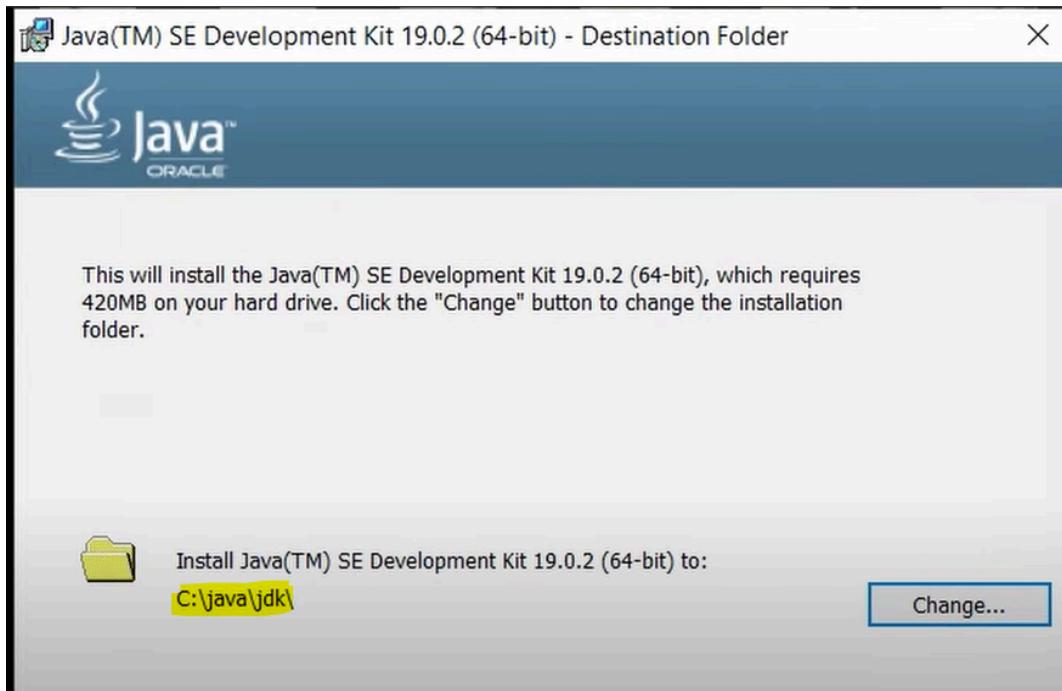
Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications and components using the Java programming language.

The JDK includes tools for developing and testing programs written in the Java programming language and running on the Java platform.

Linux macOS Windows

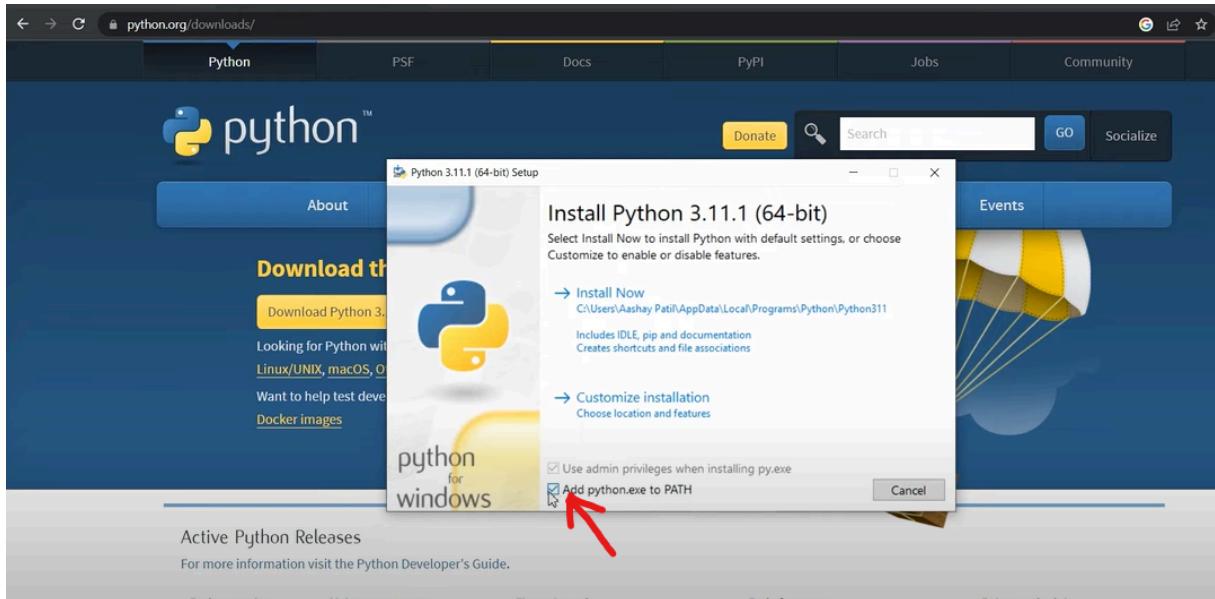
Product/file description	File size	Download
x64 Compressed Archive	179.15 MB	https://download.oracle.com/java/19/latest/jdk-19_windows-x64_bin.zip (sha256)
x64 Installer	158.91 MB	https://download.oracle.com/java/19/latest/jdk-19_windows-x64_bin.exe (sha256)
x64 MSI Installer	157.76 MB	https://download.oracle.com/java/19/latest/jdk-19_windows-x64_bin.msi (sha256)

2. After downloading the jdk file, it is recommended to create a new directory for storing the java file.



Install Python:

Install the latest version of Python and check the box as shown below. If the box is checked then there is no need to create a separate environment for python.



Install Apache Spark:

1. Download files from <https://spark.apache.org/>



Unified engine for large-scale data analytics

GET STARTED

What is Apache Spark™?

Apache Spark™ is a multi-language engine for executing data engineering, data science, and machine learning on single-node machines or clusters.

Key features

Waiting for adservice.google.com...

APACHE Spark™ Download Libraries Documentation Examples Community Developers Apache Software Foundation

Download Apache Spark™

1. Choose a Spark release:
2. Choose a package type:
3. **Download Spark: spark-3.5.0-bin-hadoop3.tgz**

4. Verify this release using the 3.5.0 signatures, checksums and project release KEYS by following these procedures.

Note that Spark 3 is pre-built with Scala 2.12 in general and Spark 3.2+ provides additional pre-built distribution with Scala 2.13.

Link with Spark

Spark artifacts are hosted in [Maven Central](#). You can add a Maven dependency with the following coordinates:

```
groupId: org.apache.spark  
artifactId: spark-core_2.12  
version: 3.5.0
```

Installing with PyPi

[PySpark](#) is now available in pypi. To install just run `pip install pyspark`.

Convenience Docker Container Images

COMMUNITY CODE

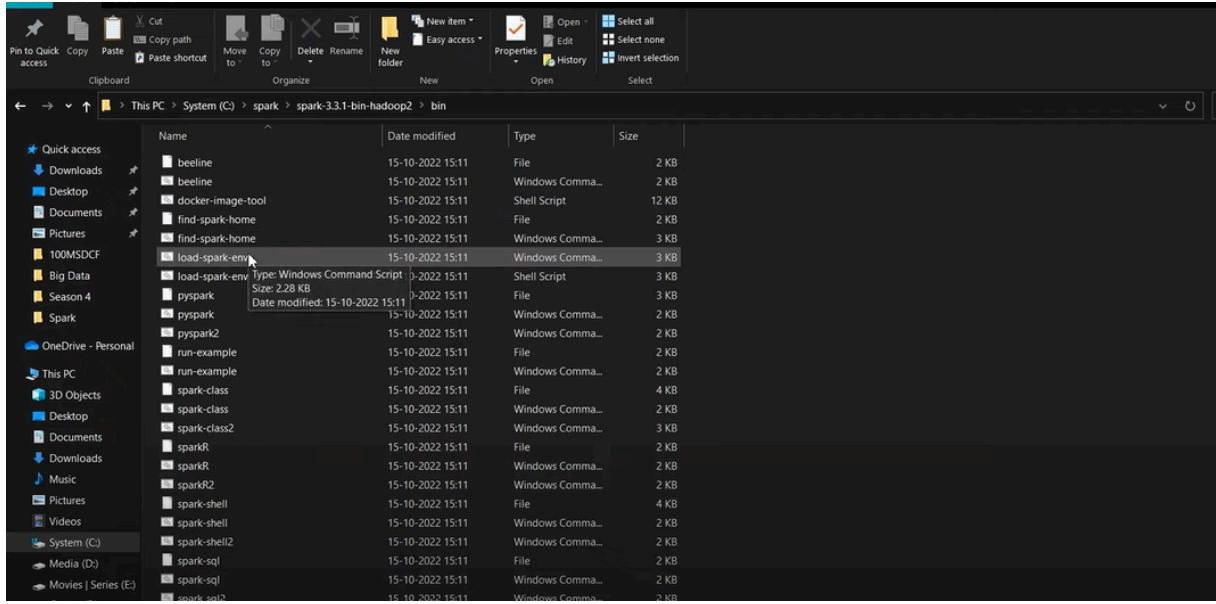
DOWNLOAD SPARK

Built-in Libraries:

- SQL and DataFrames
- Spark Streaming
- MLlib (machine learning)
- GraphX (graph)

[Third-Party Projects](#)

2. Once the download is complete, navigate to the downloads folder, create a new folder in **C drive** named “spark” and paste the file.
3. Now extract the .tar file at the above location in the C drive.
4. All the required files will be present in the bin directory as shown below:



5. But before starting the Spark, winutils needs to be installed. The winutils file should be of the same version as the Spark version.

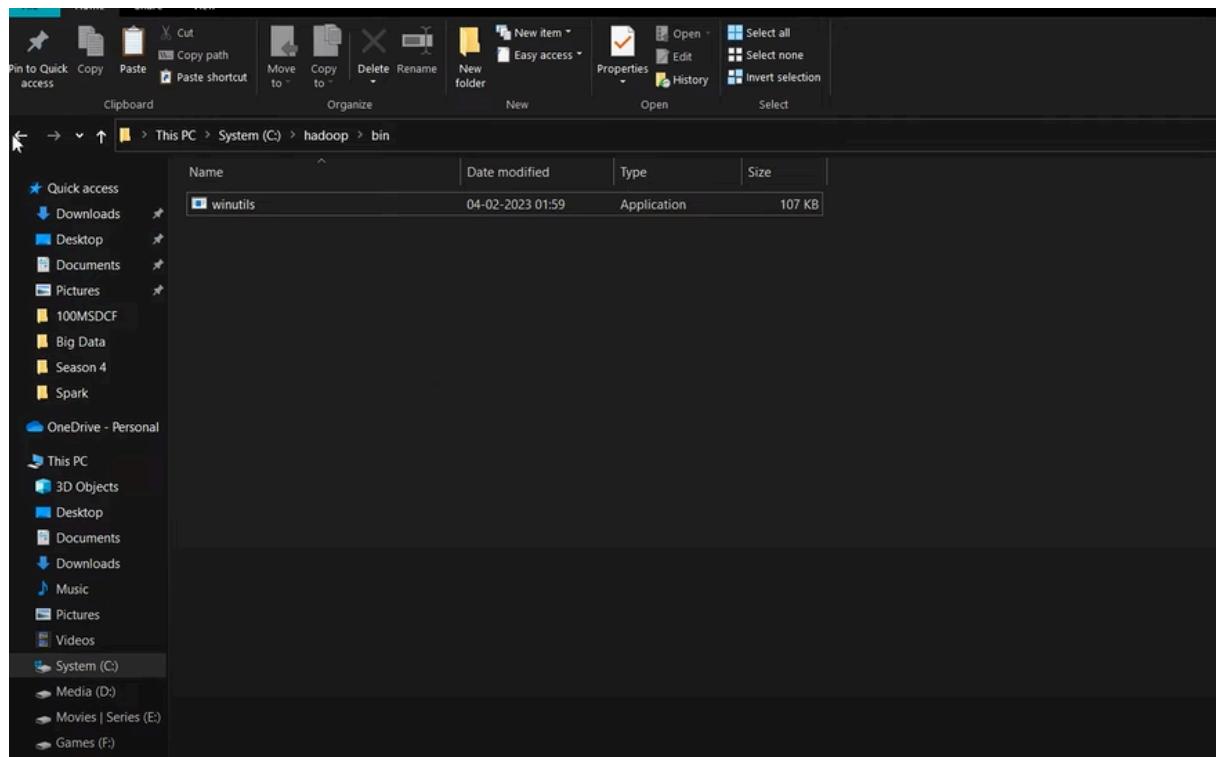
Install winutils:

1. The files with specific versions are present at the following location:<https://github.com/kontext-tech/winutils>

The screenshot shows a GitHub repository page for 'winutils'. At the top, it says 'https://github.com/steveloughran/winutils'. Below that, there's a navigation bar with 'master' (selected), '1 Branch', '3 Tags', a search bar 'Go to file', and buttons for 'Add file' and 'Code'. The main area shows a list of commits. One commit, 'hadoop-3.0.0/bin' (commit e8089ec, last year), is highlighted with a red box. Other commits include 'hadoop-2.6.0/bin', 'hadoop-2.6.3/bin', 'hadoop-2.6.4', 'hadoop-2.7.1', 'hadoop-2.8.0-RC3/bin', 'hadoop-2.8.1', 'hadoop-2.8.3/bin', '.gitattributes', '.gitignore', 'KEYS', and 'LICENSE'. The URL at the bottom is 'https://github.com/steveloughran/winutils/tree/master/hadoop-3.0.0/bin'.

Commit	Description	Date
hadoop-3.0.0/bin	Hadoop 3.0.0 windows binaries; off the release 3.0 tag, pat...	6 years ago
.gitattributes	add gitattributes to try and keep line endings on the BAT fi...	6 years ago
.gitignore	add 2.6.4 and 2.7.1 windows binaries	7 years ago
KEYS	add my new key to KEYS	6 years ago
LICENSE	Initial commit	8 years ago

2. Create a new folder “hadoop” in the **C drive**. In the hadoop folder create another folder named “bin” and place the winutils file at this location.



Now you should be having three folders

- java,
- hadoop and
- spark

Check the java and python versions through command prompt as highlighted in the below screenshot:

```
Microsoft Windows [Version 10.0.19045.2486]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Aashay Patil>java -version
java version "19.0.2" 2023-01-17
Java(TM) SE Runtime Environment (build 19.0.2+7-44)
Java HotSpot(TM) 64-Bit Server VM (build 19.0.2+7-44, mixed mode, sharing)

C:\Users\Aashay Patil>python --version
Python 3.10.1

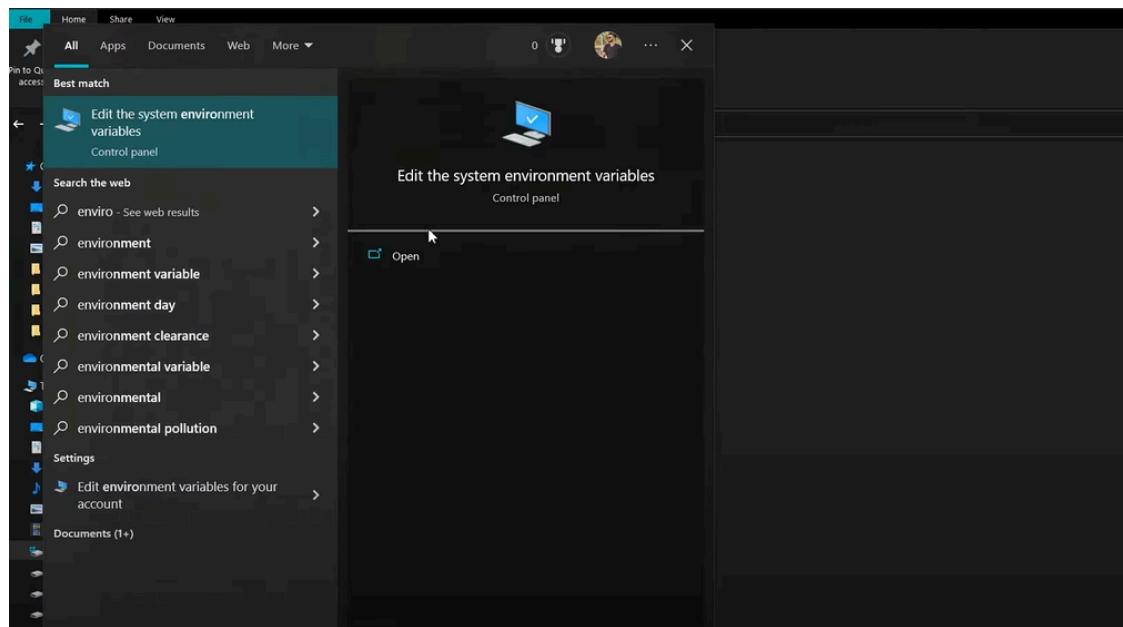
C:\Users\Aashay Patil>
```

But now for spark and hadoop, the path needs to be set

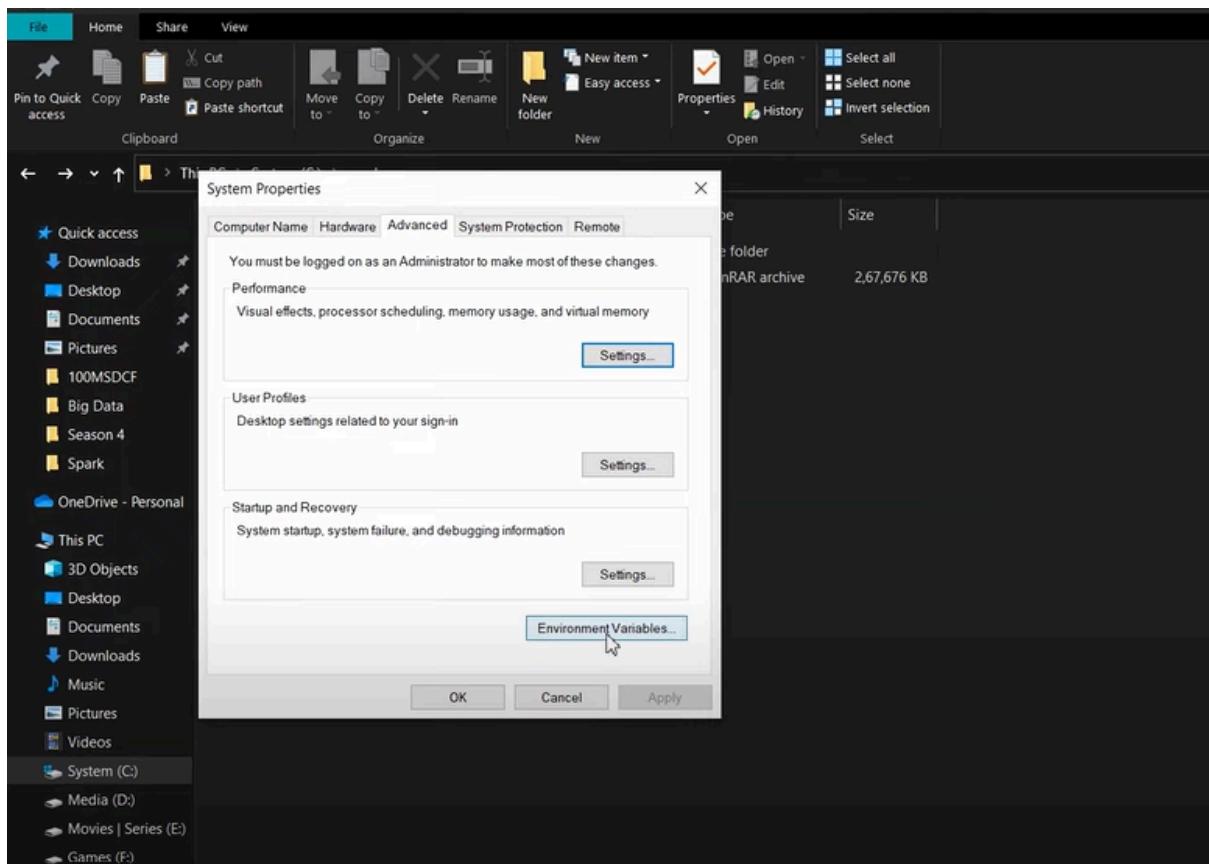
Setting Environment Variables*:

* Please note that the location of the environment variables and path to the bin folder should be very accurate, if not then you might face errors later on.

1. Search for Environment variables in windows.

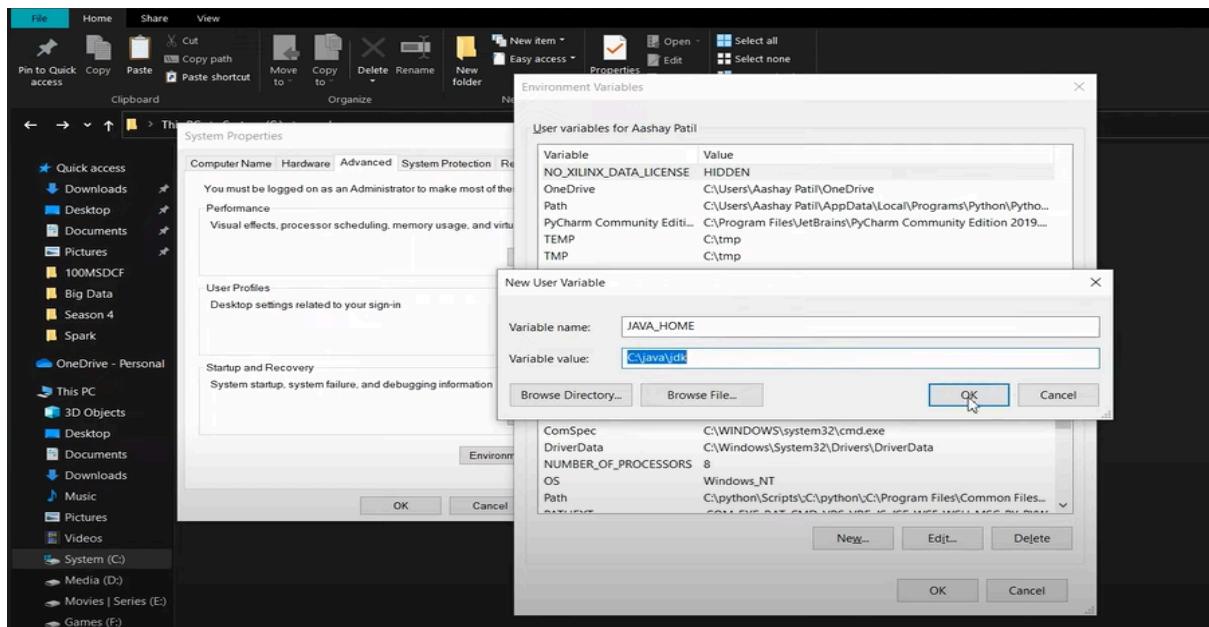


2. Go to “Edit the system environment variables” and select “Environment Variables”

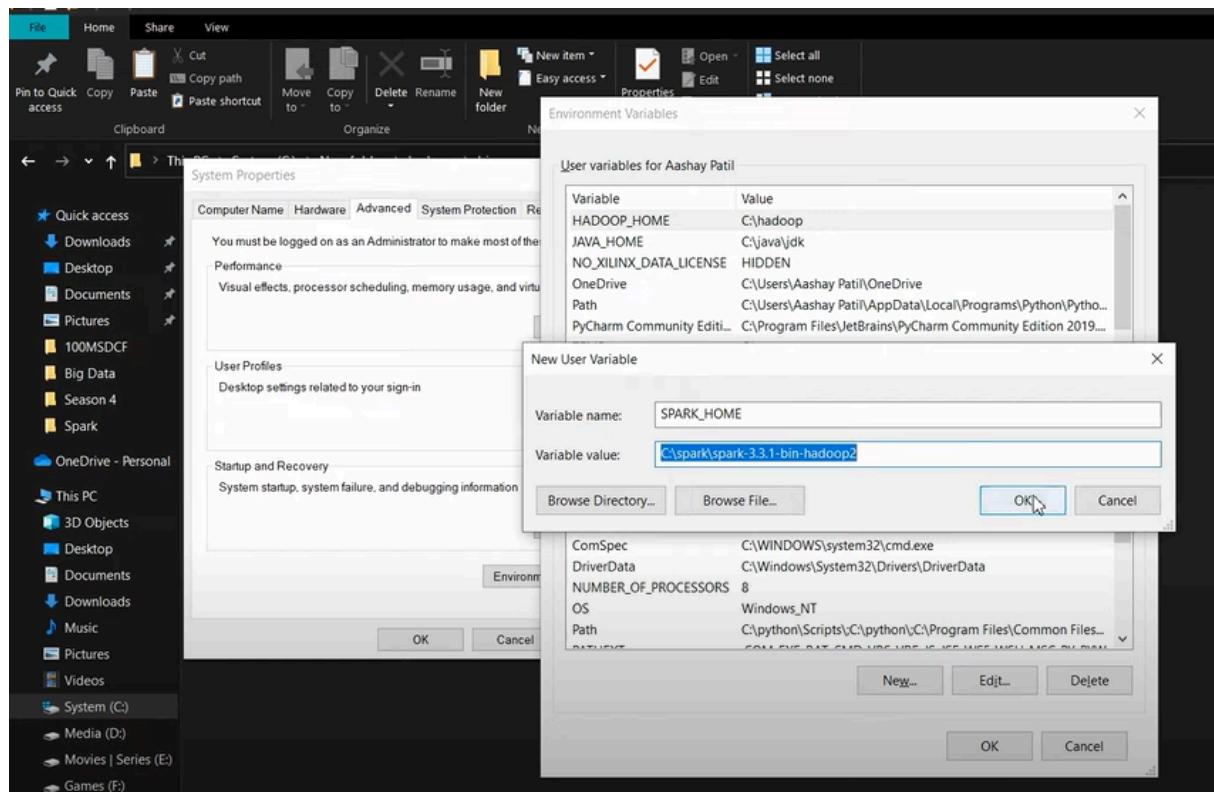
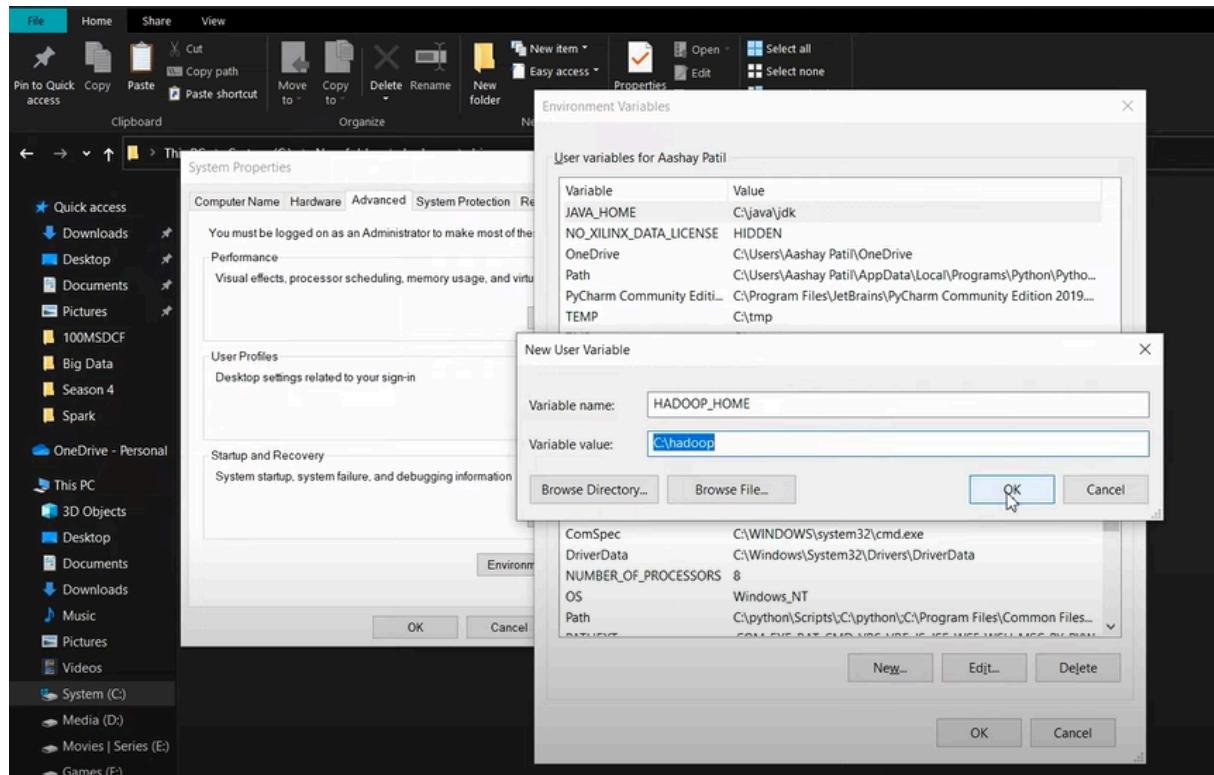


3. Java Path environment variable:

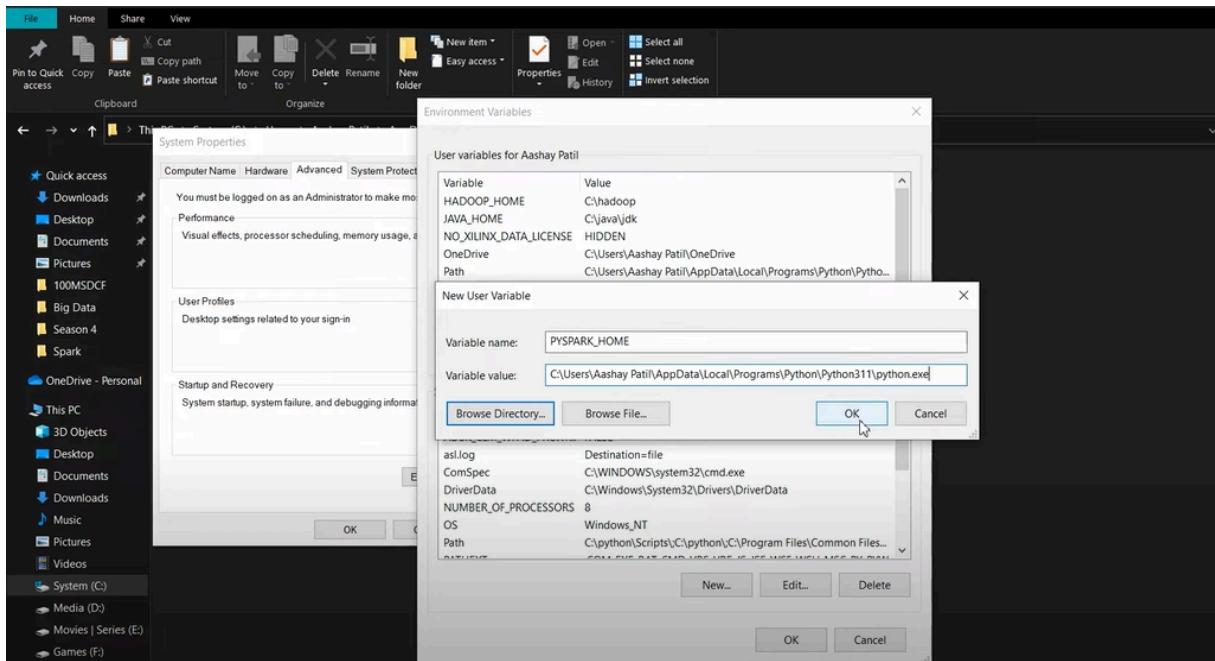
After installing Java, set the “JAVA_HOME” environment variable to the path where jdk file is present



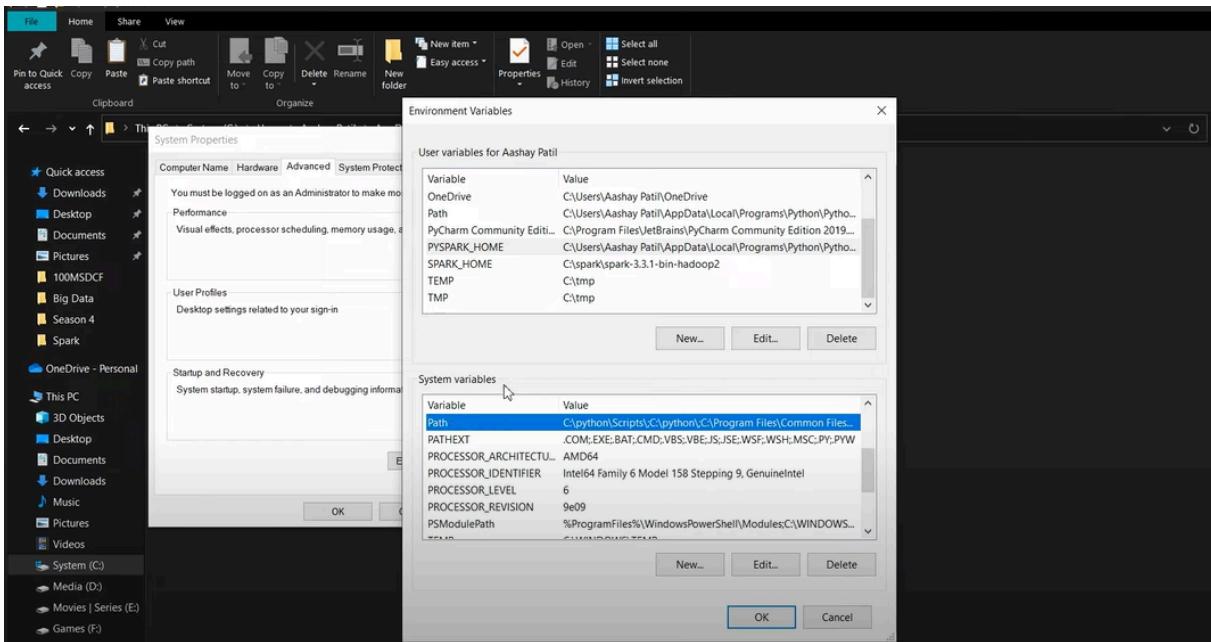
Similarly, follow the same to Hadoop and Spark



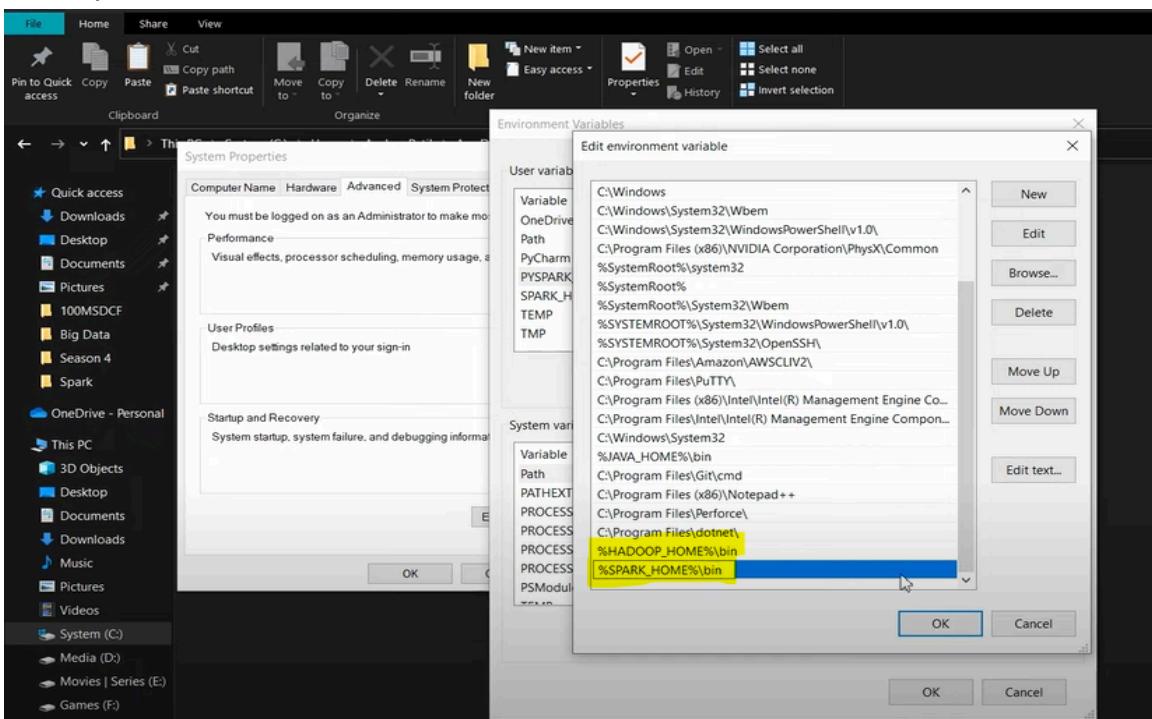
Additionally, to be on the safer side you can also add the path to the Python file. Please note that the location of the .exe file for the python version should be mentioned.



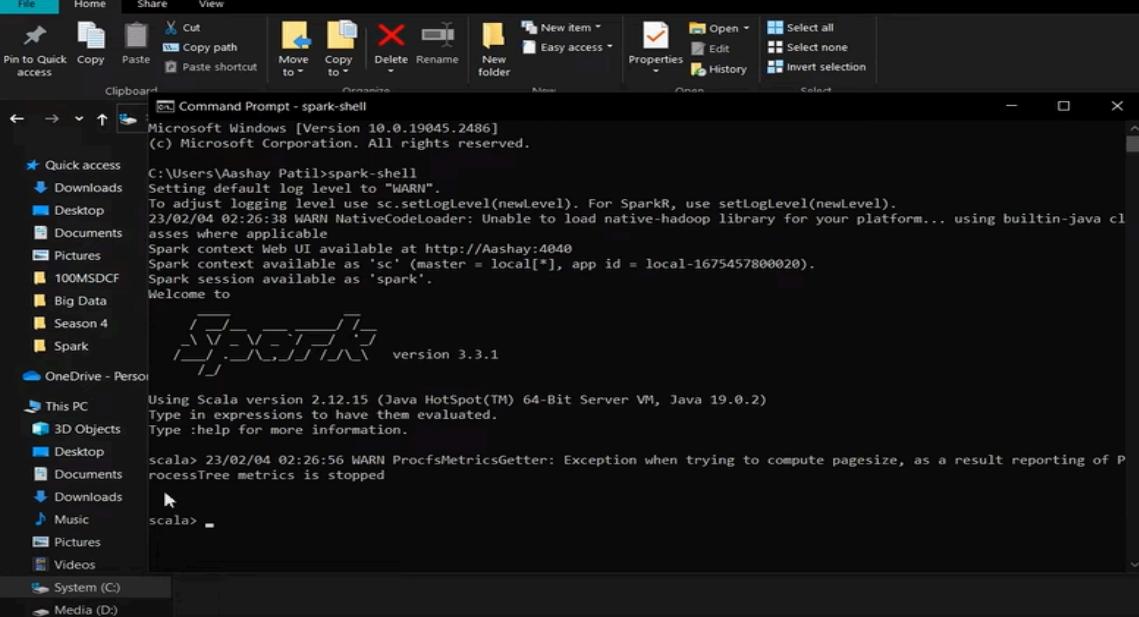
4. Provide path and the bin directories for the environments we just created:
Go to “System variables” and select path, then click on the new option.



Add `%JAVA_HOME%\bin` to the “Path” environment variable, similarly for Spark and Hadoop as shown below:



To verify your installation, go to the command prompt and enter “spark-shell”. If the installation is correct then you should be getting similar output to that shown in the screenshot below:



The screenshot shows a Microsoft Windows Command Prompt window titled "Command Prompt - spark-shell". The window displays the following text:

```
C:\Users\Aashay Patil>spark-shell
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/02/04 02:26:38 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Spark context Web UI available at http://Aashay:4040
Spark context available as 'sc' (master = local[*], app id = local-1675457800020).
Spark session available as 'spark'.
Welcome to
    / \ \
   /   \ \
  /     \ \
 /       \ \
/         \ \
/           \ \
version 3.3.1

Using Scala version 2.12.15 (Java HotSpot(TM) 64-Bit Server VM, Java 19.0.2)
Type in expressions to have them evaluated.
Type :help for more information.

scala> 23/02/04 02:26:56 WARN ProcfsMetricsGetter: Exception when trying to compute pagesize, as a result reporting of ProcessTree metrics is stopped
scala> 
```

TensorFlow

TensorFlow is an open source machine learning framework developed by Google Brain for machine learning operations.

Python can be downloaded from: <https://www.python.org/downloads/>

Anaconda Installation

The recommended installation procedure, as it will install Python among all other relevant dependencies. Anaconda installation for Windows can be found in the following link: [Installing on Windows — Anaconda documentation](#)

TensorFlow can be installed through Anaconda, by creating specific environments. It requires a minimum Windows distribution of Windows 10 and Python version of 3.8 for installation.

CUDA Toolkit Installation

If your computer is a CUDA-capable system, you can install TensorFlow together with the CUDA drivers, allowing for GPU acceleration with your TensorFlow operations.

Windows distributions supported:

- Microsoft Windows 11 21H2
- Microsoft Windows 11 22H2-SV2
- Microsoft Windows 10
- Microsoft Windows Server 2022
- Microsoft Windows Server 2019

CUDA installation is as follows.

a. Verify if system is CUDA-capable

- i. Open a run window from the Start Menu
- ii. Run the following command

```
Control /name Microsoft.DeviceManager
```

- iii. Go to Display Adapters section of the Windows Device Managers

If the NVIDIA card is listed in <https://developer.nvidia.com/cuda-gpus>, the system is CUDA-capable.

b. Install CUDA - CUDA 11.8

Run the following command to install CUDA.

```
conda install -c conda-forge cudatoolkit==11.8  
cudnn==8.9.7
```

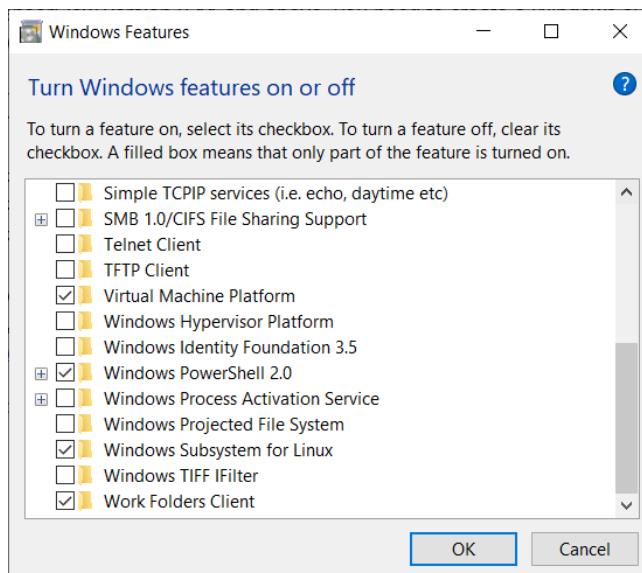
Note: Do not install TensorFlow with conda, it will not be the latest version. TensorFlow is officially released only in PyPi, so use pip for installation procedure.

Pip Installation (TensorFlow version 2.11 and above)

For TensorFlow version 2.11 and above, you need to create a Windows subsystem in Linux (WSL) for installation. The details are shown below.

Open the Windows Command Prompt/PowerShell and run the following command.

1. Turn Windows features on or off – Check the box that reads: Windows Subsystem for Linux



2. Restart your device as instructed
3. Run this command: `wsl --list -online`
4. Run the following command: `wsl --install -d Ubuntu-22.04`

Following installation, open WSL and set a username and password. If there are any errors, please troubleshoot through:

[WSL/WSL/troubleshooting.md at main · MicrosoftDocs/WSL · GitHub](#)

Steps:

1. Open WSL Shell from the Start menu.
2. pip installation: Ensure that pip(Python's package installer) is up-to-date by running:

```
pip install --upgrade pip
```

If pip not installed, install it:

```
sudo apt install -y python3-pip
```

3. TensorFlow: Install TensorFlow by running

- a. For GPU installation:

```
pip install tensorflow[and-cuda]
```

- b. For CPU installation:

```
pip install tensorflow
```

4. Verify Installation: To check if TensorFlow was installed correctly, run the following code in python to verify:

```
import tensorflow as tf  
print(tf.__version__)
```

Conda Installation (TensorFlow version 2.10)

TensorFlow 2.10 can be installed through the Anaconda distribution itself. Conda can be used for this installation procedure but this will not be the latest TensorFlow version.

1. For GPU Installation

```
conda create -n <env_name> tensorflow-gpu  
  
conda activate <env_name>
```

2. For CPU Installation

```
conda create -n <env_name> tensorflow
```

```
conda activate <env_name>
```

This will create an anaconda environment with TensorFlow 2.10 installed.

PyTorch

PyTorch is a machine learning framework based on the torch library originally developed by MetaAI and currently part of Linux Foundation.

It requires a minimum Windows distribution of Windows 10 and Python version of 3.8 for PyTorch 2.1.

Anaconda Installation

The recommended installation procedure, as it will install Python among all other relevant dependencies. Anaconda installation for Windows can be found in the following link: [Installing on Windows — Anaconda documentation](#)

1. Create and activate a new environment in Anaconda through the following code.

```
conda create -n <env_name> anaconda
```

```
conda activate <env_name>
```

Notes:

It is always recommended to create a new environment whenever installing any software. In case of any failure, the environment can always be deleted to prevent any problems at the root level.

Setting the environment to anaconda enables it to install the standard packages such as numpy, pandas, scikit-learn etc.

2. Install the relevant PyTorch configuration files based on the computer's specifications.

You can proceed with the PyTorch installation now.

- [Conda Installation - CUDA 11.8](#)

```
conda install pytorch torchvision torchaudio  
pytorch-cuda=11.8 -c pytorch -c nvidia
```

- Pip Installation - CUDA 11.8

```
pip install --upgrade pip
```

```
pip install torch torchvision torchaudio --index-url  
https://download.pytorch.org/whl/cu118
```

If your system is not CUDA-capable, you can install the CPU version using the following code below.

- Conda Installation - CPU Installation

```
conda install pytorch torchvision torchaudio cpuonly -c  
pytorch
```

- Pip Installation - CPU Installation

```
pip install --upgrade pip
```

```
pip install torch torchvision torchaudio
```

Notes:

Torchvision and torchaudio are common dependencies associated with PyTorch and can be installed concurrently.

If using CUDA, install version 11.8. Version 12.1 has stability issues and may cause errors.

3. Verify your installation.

You can verify your installation by running the following code.

```
import torch  
  
print(torch.__version__)
```

For CPU installation, it will return the following.

```
2.1.1+cpu where 2.1.1 is the Torch version installed.
```

For GPU installation, the code will return the following.

```
2.1.1+cu118 for PyTorch 2.1.1 and CUDA 11.8.
```

Additionally, you can verify GPU installation using the following code.

```
import torch  
  
print(torch.cuda.is_available())
```

This should return True if the GPU installation is correct.