Spark

https://spark.apache.org/downloads.html

Introduction
Apache Spark is an open-source framework that processes large volumes of stream data from multiple sources. Spark is used in distributed computing with machine learning applications, data analytics, and graph-parallel processing.
This guide will show you **how to install Apache Spark on Windows 10** and test the installation.

phoenixNAP GLOBAL IT SERVICES
How to Install Spark on Windows
APACHE Spark

Prerequisites

- A system running Windows 10

- A user account with administrator privileges (required to install software, modify file permissions, and modify system PATH)
- Command Prompt or Powershell
- A tool to extract .tar files, such as 7-Zip

Install Apache Spark on Windows

Installing Apache Spark on Windows 10 may seem complicated to novice users, but this simple tutorial will have you up and running. If you already have Java 8 and Python 3 installed, you can skip the first two steps.

Step 1: Install Java 8

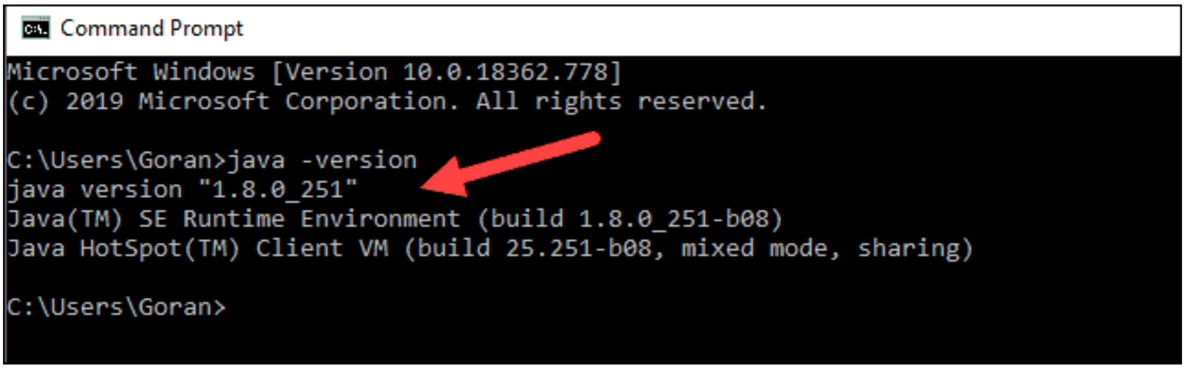
Apache Spark requires Java 8. You can check to see if Java is installed using the command prompt.

Open the command line by clicking **Start** > type *cmd* > click **Command Prompt**.

Type the following command in the command prompt:

```
java -version
```

If Java is installed, it will respond with the following output:

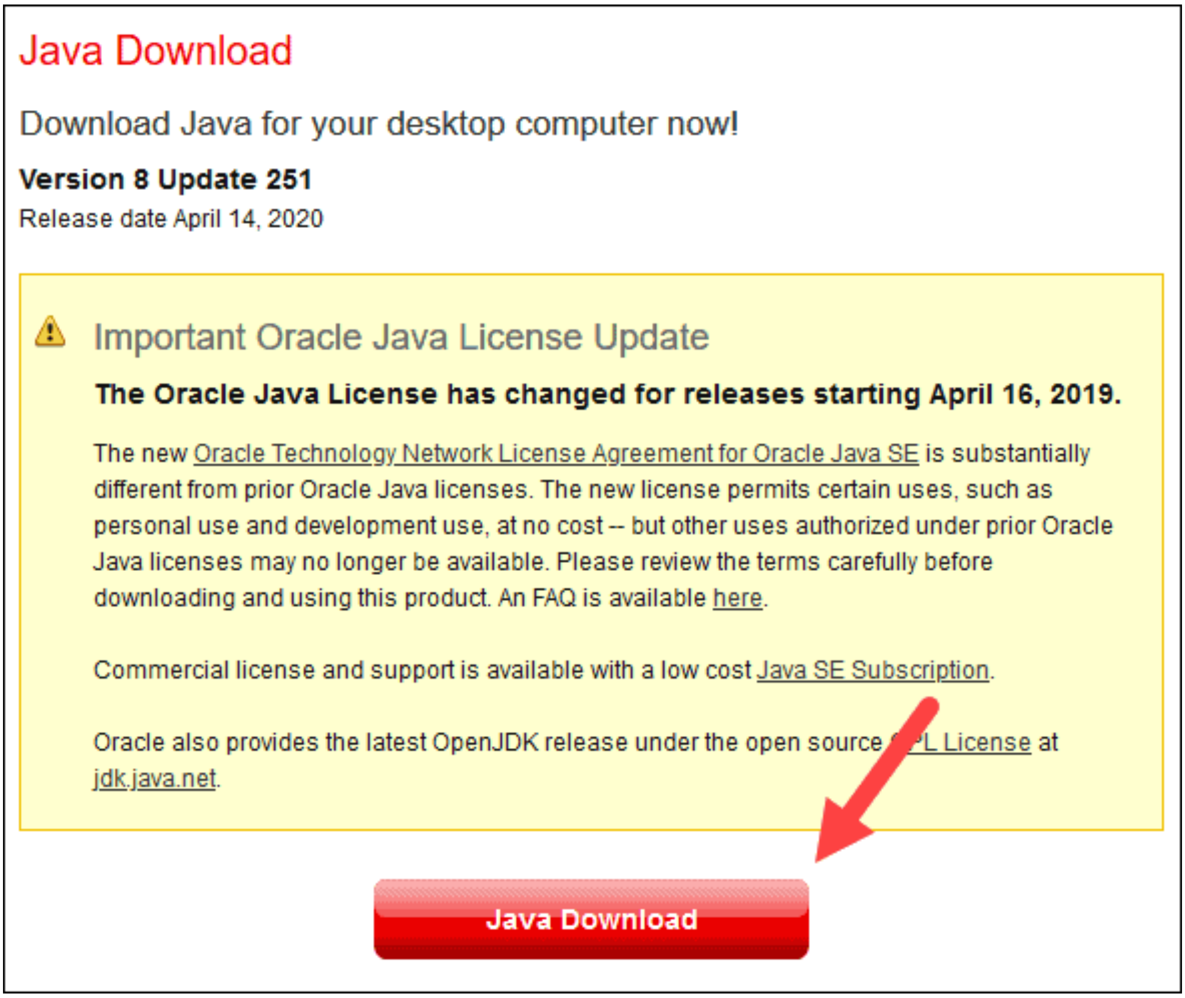


```
Microsoft Windows [Version 10.0.18362.778]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Goran>java -version
java version "1.8.0_251"
Java(TM) SE Runtime Environment (build 1.8.0_251-b08)
Java HotSpot(TM) Client VM (build 25.251-b08, mixed mode, sharing)

C:\Users\Goran>
```

Your version may be different. The second digit is the Java version – in this case, Java 8.

If you don’t have Java installed:

1. Open a browser window, and navigate to https://java.com/en/download/.



Java Download

Download Java for your desktop computer now!

Version 8 Update 251

Release date April 14, 2020

⚠ Important Oracle Java License Update

The Oracle Java License has changed for releases starting April 16, 2019.

The new Oracle Technology Network License Agreement for Oracle Java SE is substantially different from prior Oracle Java licenses. The new license permits certain uses, such as personal use and development use, at no cost -- but other uses authorized under prior Oracle Java licenses may no longer be available. Please review the terms carefully before downloading and using this product. An FAQ is available here.

Commercial license and support is available with a low cost Java SE Subscription.

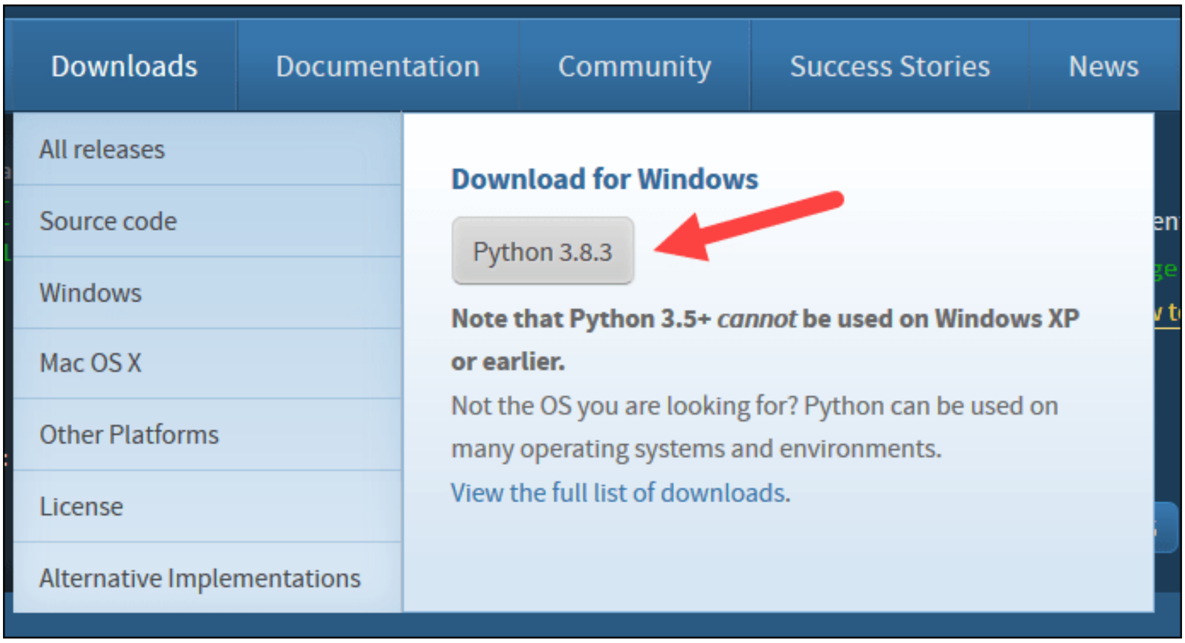
Oracle also provides the latest OpenJDK release under the open source GPL License at jdk.java.net.

2. Click the **Java Download** button and save the file to a location of your choice.
3. Once the download finishes double-click the file to install Java.

Note: At the time this article was written, the latest Java version is 1.8.0_251. Installing a later version will still work. This process only needs the Java Runtime Environment (JRE) – the full Development Kit (JDK) is not required. The download link to JDK is https://www.oracle.com/java/technologies/javase-downloads.html.

Step 2: Install Python

1. To install the Python package manager, navigate to https://www.python.org/ in your web browser.
2. Mouse over the **Download** menu option and click **Python 3.8.3**. 3.8.3 is the latest version at the time of writing the article.
3. Once the download finishes, run the file.



Download for Windows

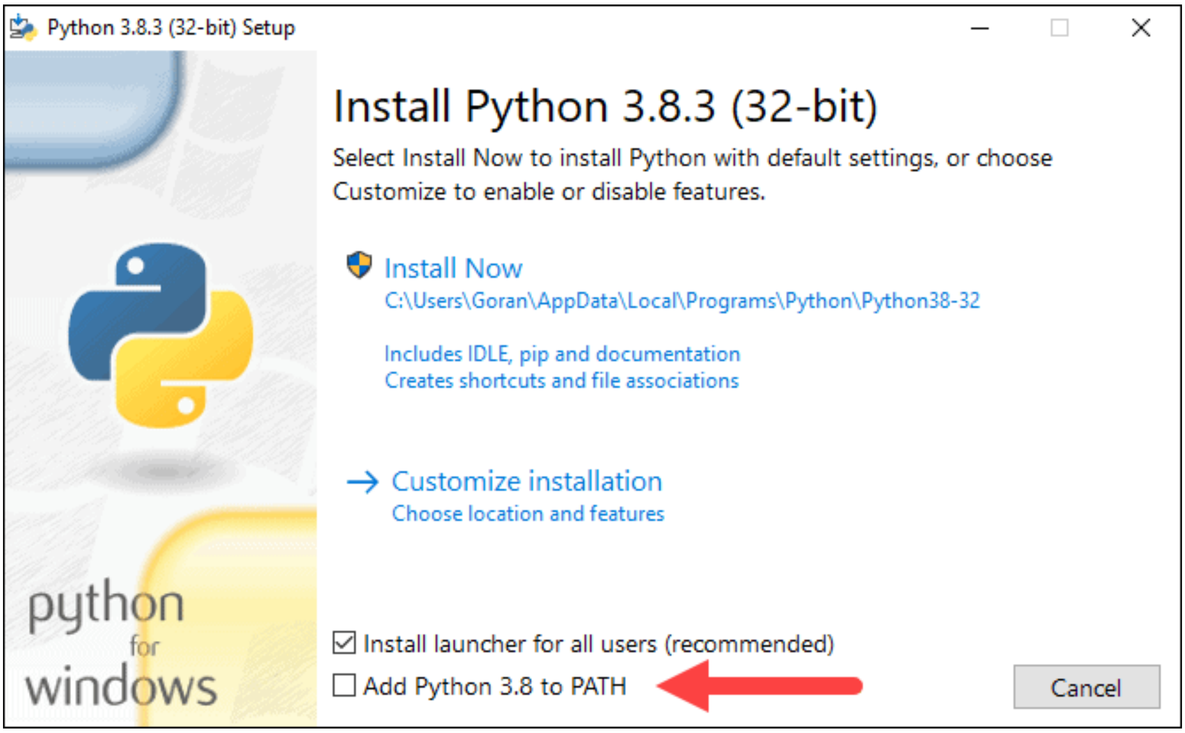
Note that Python 3.5+ *cannot* be used on Windows XP or earlier.

Not the OS you are looking for? Python can be used on many operating systems and environments.

View the full list of downloads.

4. Near the bottom of the first setup dialog box, check off *Add Python 3.8 to PATH*. Leave the other box checked.

5. Next, click **Customize installation**.



Install Python 3.8.3 (32-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\Goran\AppData\Local\Programs\Python\Python38-32

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)

□ Add Python 3.8 to PATH

6. You can leave all boxes checked at this step, or you can uncheck the options you

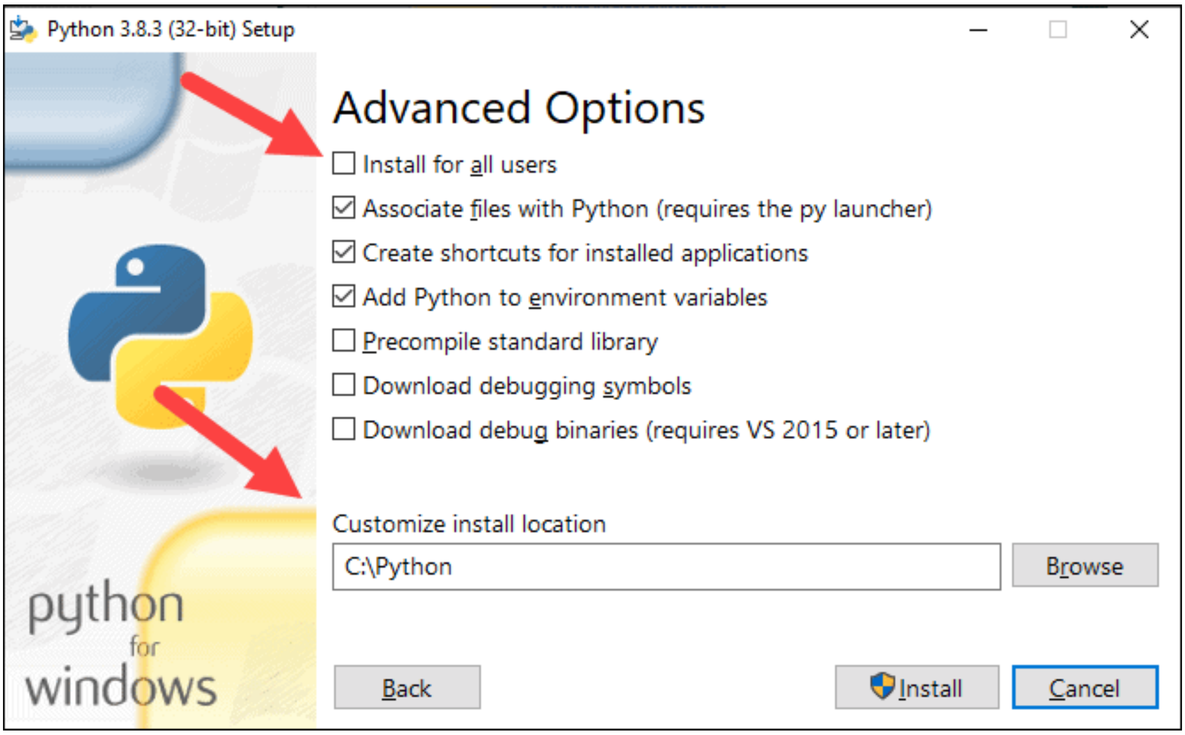
do not want.

7. Click **Next**.

8. Select the box **Install for all users** and leave other boxes as they are.

9. Under *Customize install location*, click **Browse** and navigate to the C drive. Add a new folder and name it *Python*.

10. Select that folder and click **OK**.



Python 3.8.3 (32-bit) Setup

Advanced Options

□ Install for all users
☑ Associate files with Python (requires the py launcher)
☑ Create shortcuts for installed applications
☑ Add Python to environment variables
□ Precompile standard library
□ Download debugging symbols
□ Download debug binaries (requires VS 2015 or later)

Customize install location

C:\Python

11. Click **Install**, and let the installation complete.

12. When the installation completes, click the *Disable path length limit* option at the bottom and then click **Close**.

13. If you have a command prompt open,

restart it. Verify the installation by checking the version of Python:

```
python --version
```

The output should print **Python 3.8.3**.

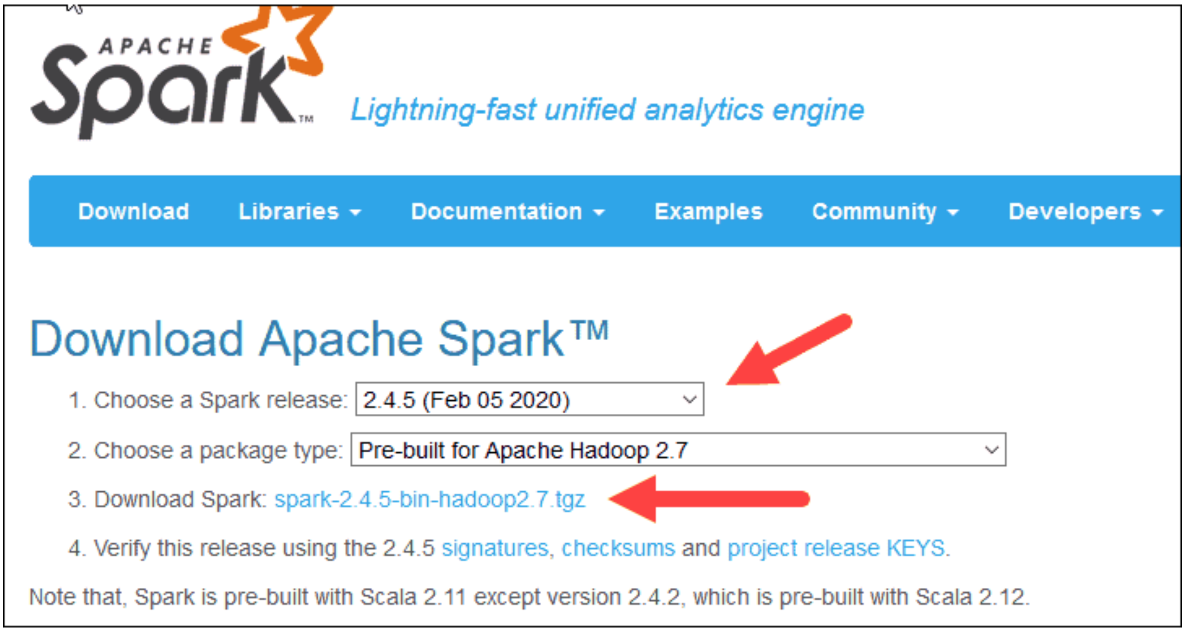
Note: For detailed instructions on how to install Python 3 on Windows or how to troubleshoot potential issues, refer to our Install Python 3 on Windows guide.

Step 3: Download Apache Spark

1. Open a browser and navigate to https://spark.apache.org/downloads.html.
2. Under the *Download Apache Spark* heading, there are two drop-down menus. Use the current non-preview version.

- In our case, in ***Choose a Spark release*** drop-down menu select **2.4.5 (Feb 05 2020)**.
- In the second drop-down ***Choose a package type,*** leave the selection **Pre-built for Apache Hadoop 2.7**.

3. Click the ***spark-2.4.5-bin-hadoop2.7.tgz*** link.



Download Apache Spark™

1. Choose a Spark release: 2.4.5 (Feb 05 2020)
2. Choose a package type: Pre-built for Apache Hadoop 2.7
3. Download Spark: spark-2.4.5-bin-hadoop2.7.tgz
4. Verify this release using the 2.4.5 signatures, checksums and project release KEYS.

Note that, Spark is pre-built with Scala 2.11 except version 2.4.2, which is pre-built with Scala 2.12.

4. A page with a list of mirrors loads where you can see different servers to download from. Pick any from the list and save the file to your Downloads folder.

Step 4: Verify Spark Software File

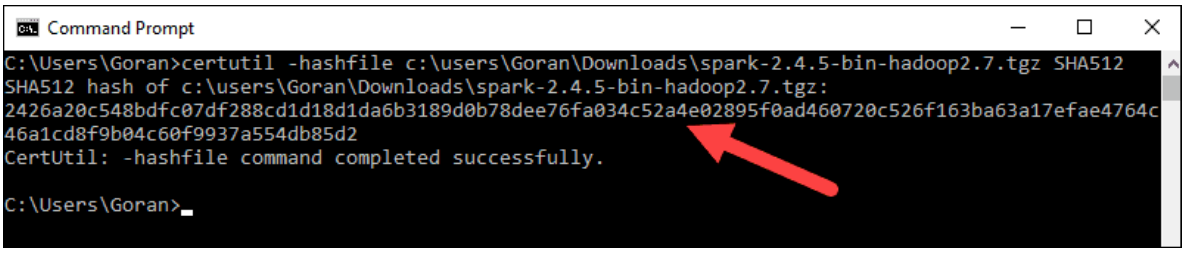
1. Verify the integrity of your download by checking the **checksum** of the file. This ensures you are working with unaltered, uncorrupted software.

2. Navigate back to the *Spark Download* page and open the **Checksum** link, preferably in a new tab.

3. Next, open a command line and enter the following command:

```
certutil -hashfile c:
\users\username\Downloads\spark-2.4.5-
bin-hadoop2.7.tgz SHA512
```

4. Change the username to your username. The system displays a long alphanumeric code, along with the message **Certutil: -hashfile completed successfully**.



```
C:\Users\Goran>certutil -hashfile c:\users\Goran\Downloads\spark-2.4.5-bin-hadoop2.7.tgz SHA512
SHA512 hash of c:\users\Goran\Downloads\spark-2.4.5-bin-hadoop2.7.tgz:
2426a20c548bdfc07df288cd1d18d1da6b3189d0b78dee76fa034c52a4e02895f0ad460720c526f163ba63a17efae4764c
46a1cd8f9b04c60f9937a554db85d2
CertUtil: -hashfile command completed successfully.

C:\Users\Goran>
```

5. Compare the code to the one you opened in a new browser tab. If they match, your download file is uncorrupted.

Step 5: Install Apache Spark

Installing Apache Spark
involves **extracting the downloaded file** to the desired location.

1. Create a new folder named *Spark* in the root of your C: drive. From a command line, enter the following:

```
cd \

mkdir Spark
```

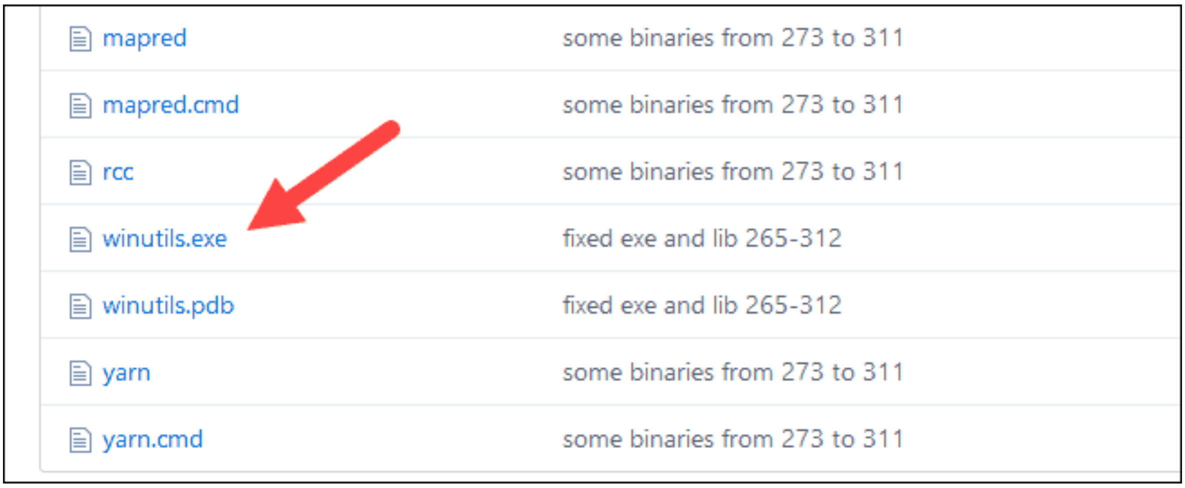
2. In Explorer, locate the Spark file you downloaded.
3. Right-click the file and extract it to *C:\|Spark* using the tool you have on your system (e.g., 7-Zip).

4. Now, your *C:|Spark* folder has a new folder *spark-2.4.5-bin-hadoop2.7* with the necessary files inside.

Step 6: Add winutils.exe File

Download the **winutils.exe** file for the underlying Hadoop version for the Spark installation you downloaded.

1. Navigate to this URL https://github.com/cdarlint/winutils and inside the **bin** folder, locate **winutils.exe**, and click it.



mapred	some binaries from 273 to 311
mapred.cmd	some binaries from 273 to 311
rcc	some binaries from 273 to 311
winutils.exe	fixed exe and lib 265-312
winutils.pdb	fixed exe and lib 265-312
yarn	some binaries from 273 to 311
yarn.cmd	some binaries from 273 to 311

2. Find the **Download** button on the right side to download the file.

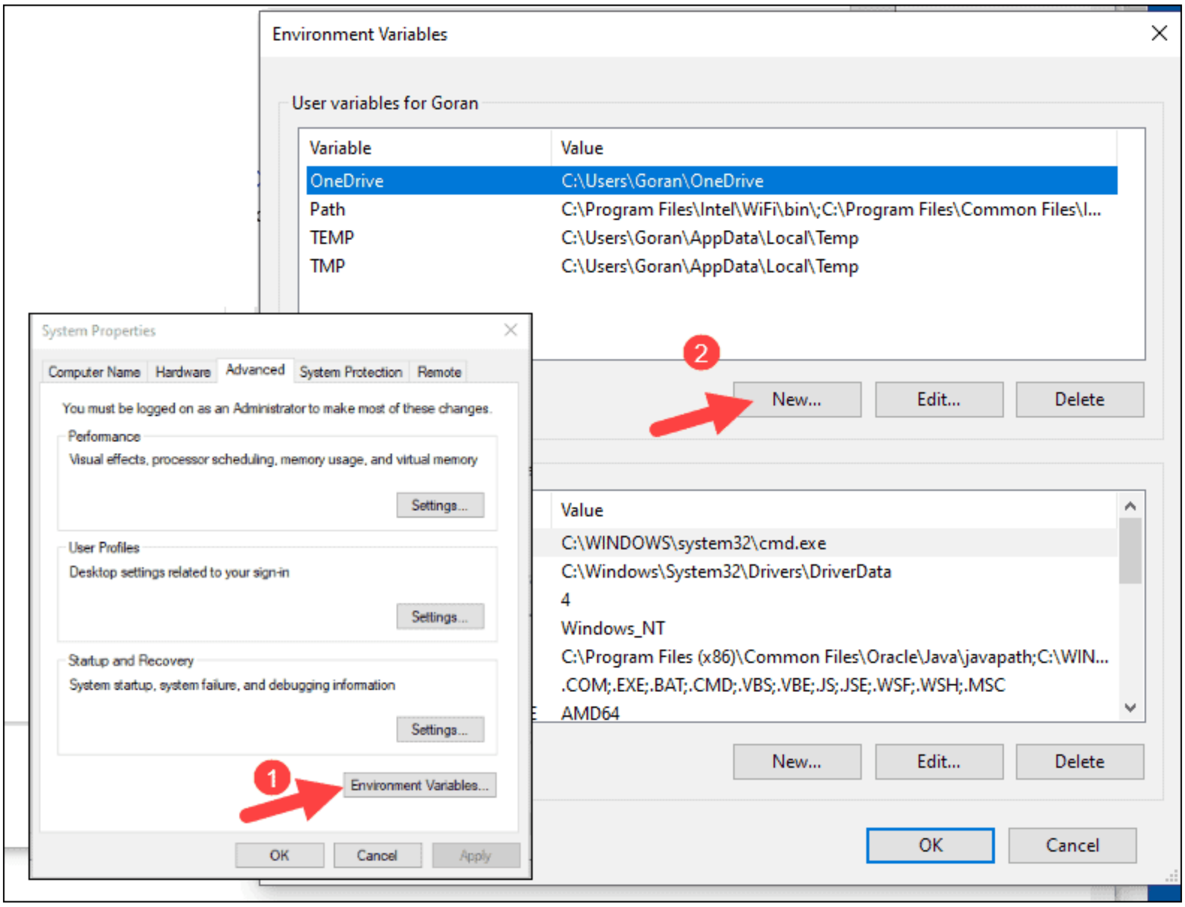
3. Now, create new folders ***Hadoop*** and **bin** on C: using Windows Explorer or the Command Prompt.

4. Copy the winutils.exe file from the Downloads folder to **C:\hadoop\bin**.

Step 7: Configure Environment Variables

Configuring environment variables in Windows adds the Spark and Hadoop locations to your system PATH. It allows you to run the Spark shell directly from a command prompt window.

1. Click **Start** and type *environment*.
2. Select the result labeled ***Edit the system environment variables***.
3. A System Properties dialog box appears. In the lower-right corner, click **Environment Variables** and then click **New** in the next window.



Variable	Value
OneDrive	C:\Users\Goran\OneDrive
Path	C:\Program Files\Intel\WiFi\bin\;C:\Program Files\Common Files\I...
TEMP	C:\Users\Goran\AppData\Local\Temp
TMP	C:\Users\Goran\AppData\Local\Temp

Value
C:\WINDOWS\system32\cmd.exe
C:\Windows\System32\Drivers\DriverData
4
Windows_NT
C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\WIN...
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
AMD64

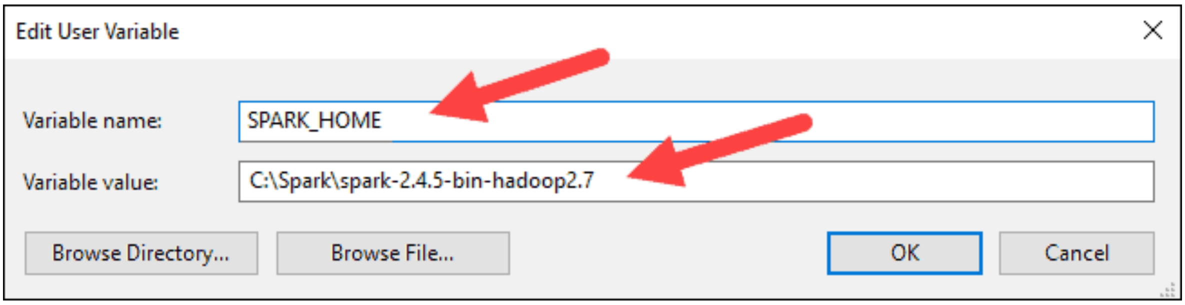
You must be logged on as an Administrator to make most of these changes.

Visual effects, processor scheduling, memory usage, and virtual memory

Desktop settings related to your sign-in

System startup, system failure, and debugging information

4. For *Variable Name* type ***SPARK_HOME***.
5. For *Variable Value* type **C:\Spark\spark-2.4.5-bin-hadoop2.7** and click OK. If you changed the folder path, use that one instead.

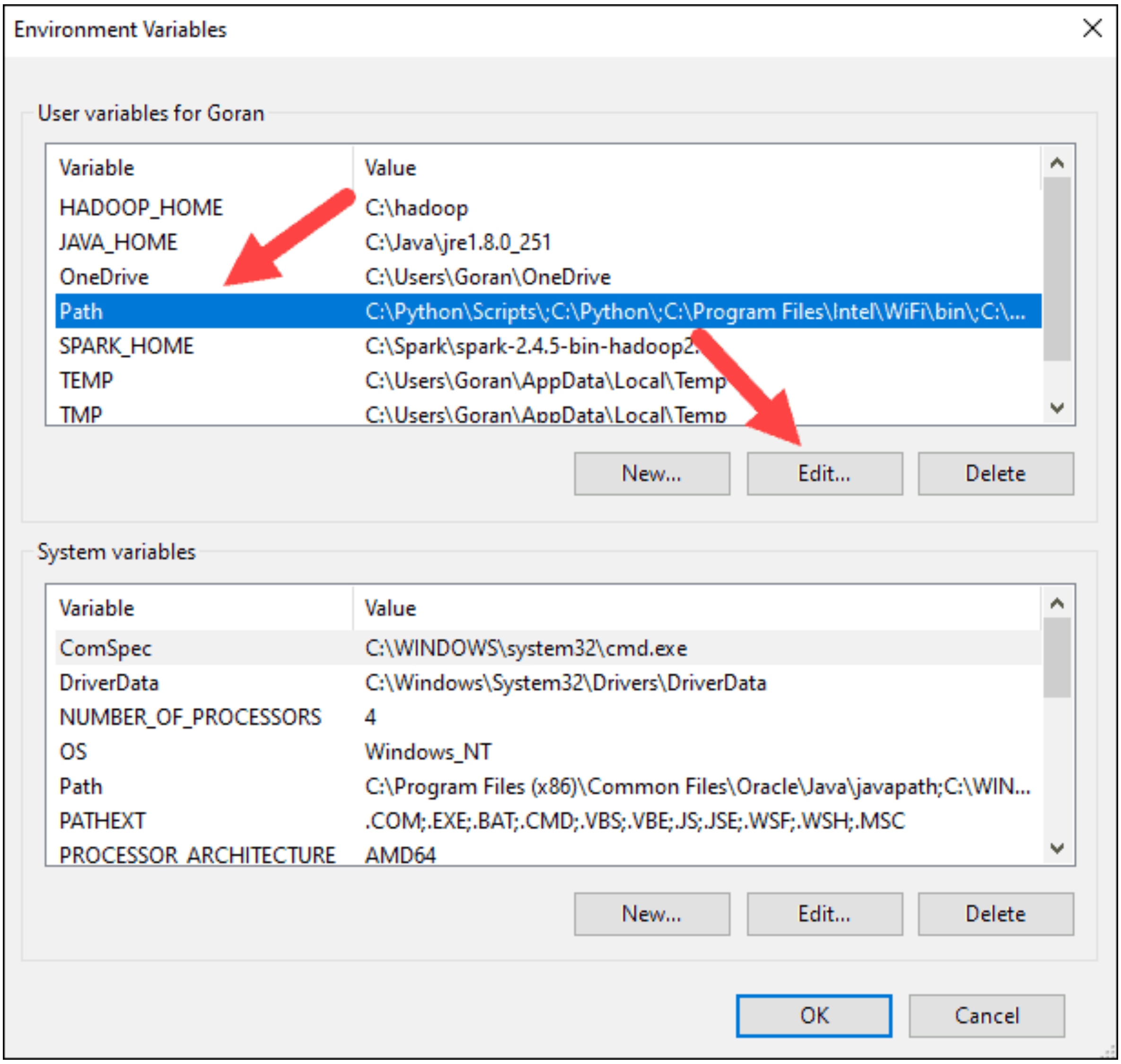


Variable name: SPARK_HOME

Variable value: C:\Spark\spark-2.4.5-bin-hadoop2.7

6. In the top box, click the **Path** entry, then click **Edit**. Be careful with editing the system path. Avoid deleting any entries

already on the list.



Environment Variables

User variables for Goran

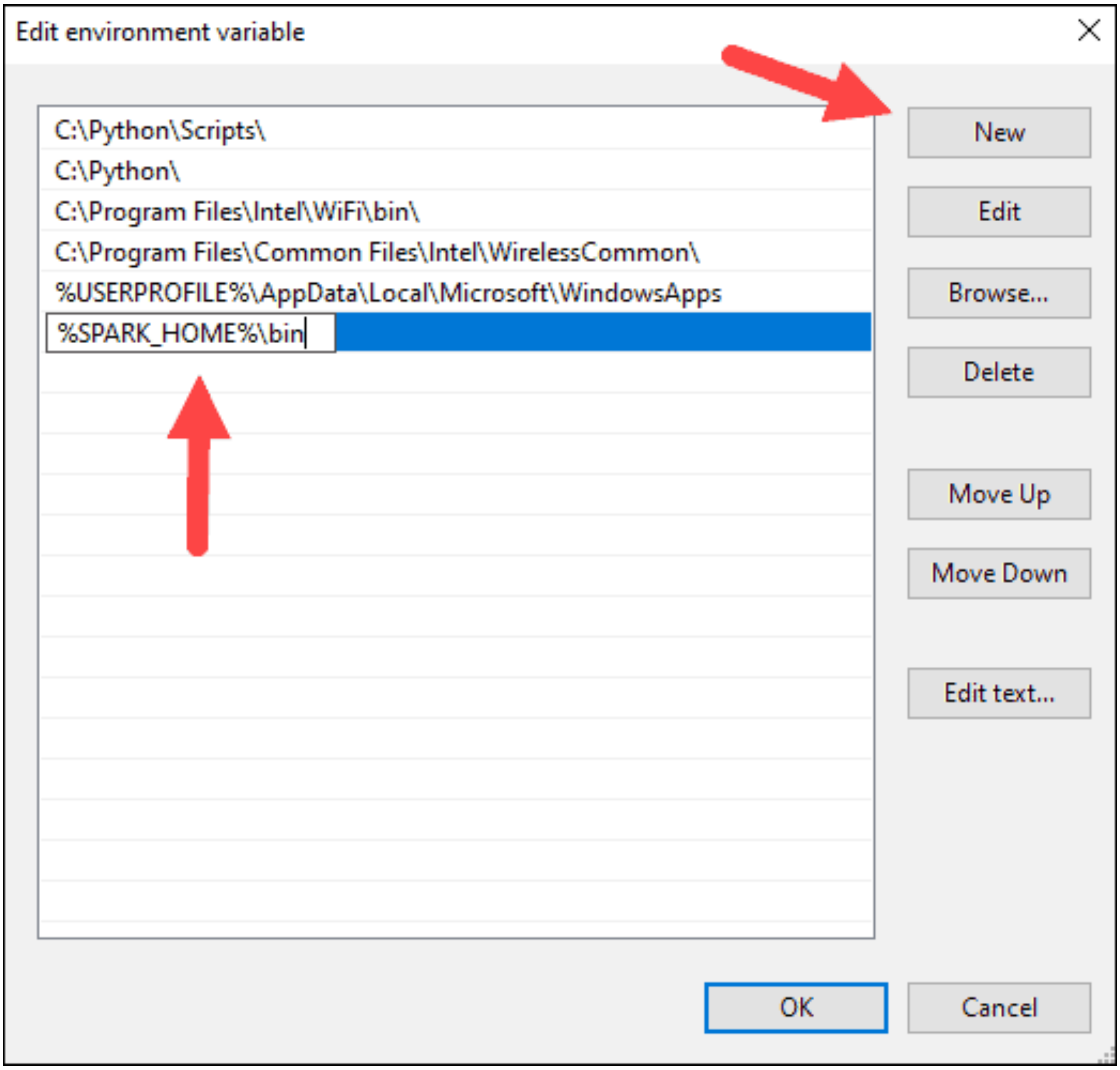
Variable	Value
HADOOP_HOME	C:\hadoop
JAVA_HOME	C:\Java\jre1.8.0_251
OneDrive	C:\Users\Goran\OneDrive
Path	C:\Python\Scripts\;C:\Python\;C:\Program Files\Intel\WiFi\bin\;C:\...
SPARK_HOME	C:\Spark\spark-2.4.5-bin-hadoop2.
TEMP	C:\Users\Goran\AppData\Local\Temp
TMP	C:\Users\Goran\AppData\Local\Temp

System variables

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\WIN...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

7. You should see a box with entries on the left. On the right, click **New**.

8. The system highlights a new line. Enter the path to the Spark folder **C:\Spark\spark-2.4.5-bin-hadoop2.7\bin**. We recommend using **%SPARK_HOME%\bin** to avoid possible issues with the path.



```
C:\Python\Scripts\
C:\Python\
C:\Program Files\Intel\WiFi\bin\
C:\Program Files\Common Files\Intel\WirelessCommon\
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
%SPARK_HOME%\bin
```

9. Repeat this process for Hadoop and Java.

- For Hadoop, the variable name is **HADOOP_HOME** and for the value use the path of the folder you created earlier: **C:\hadoop.** Add **C:\hadoop\bin** to the **Path variable** field, but we recommend using **%HADOOP_HOME%\bin**.
- For Java, the variable name

is **JAVA_HOME** and for the value use the path to your Java JDK directory (in our case it's **C:\Program Files\Java\jdk1.8.0_251**).

10. Click **OK** to close all open windows.

Note: Star by restarting the Command Prompt to apply changes. If that doesn't work, you will need to reboot the system.

Step 8: Launch Spark

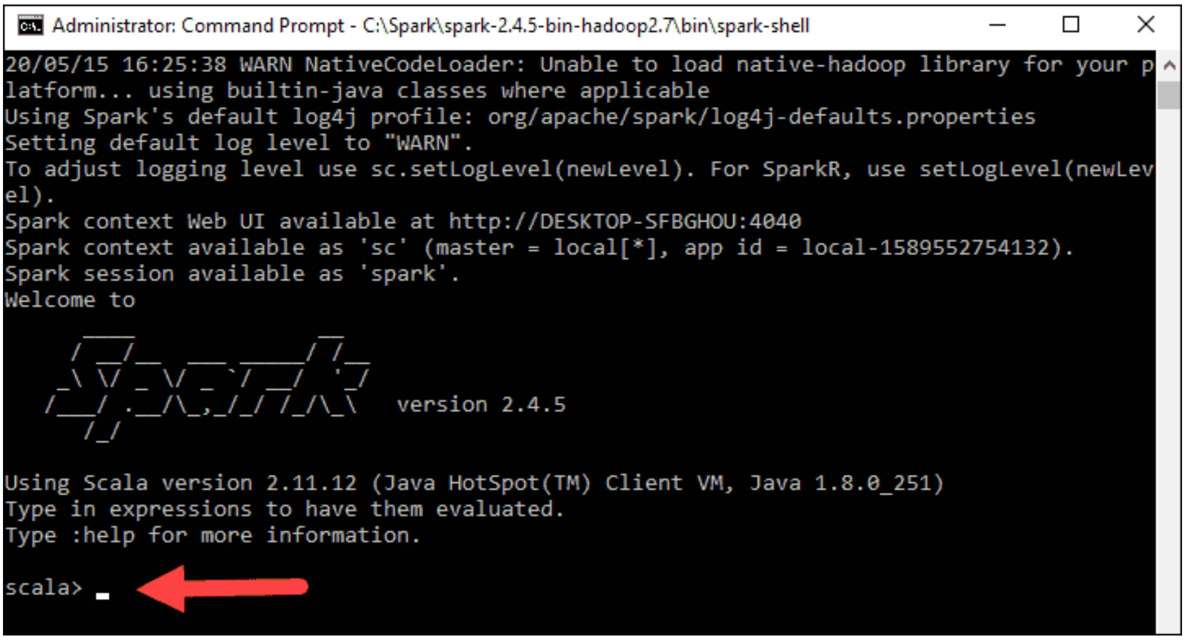
1. Open a new command-prompt window using the right-click and **Run as administrator**:
2. To start Spark, enter:

C:\Spark\spark-2.4.5-bin-hadoop2.7\bin\spark-shell

If you set the **environment path** correctly, you can type **spark-shell** to launch Spark.

3. The system should display several lines indicating the status of the application. You may get a Java pop-up. Select **Allow access** to continue.

Finally, the Spark logo appears, and the prompt displays the **Scala shell**.

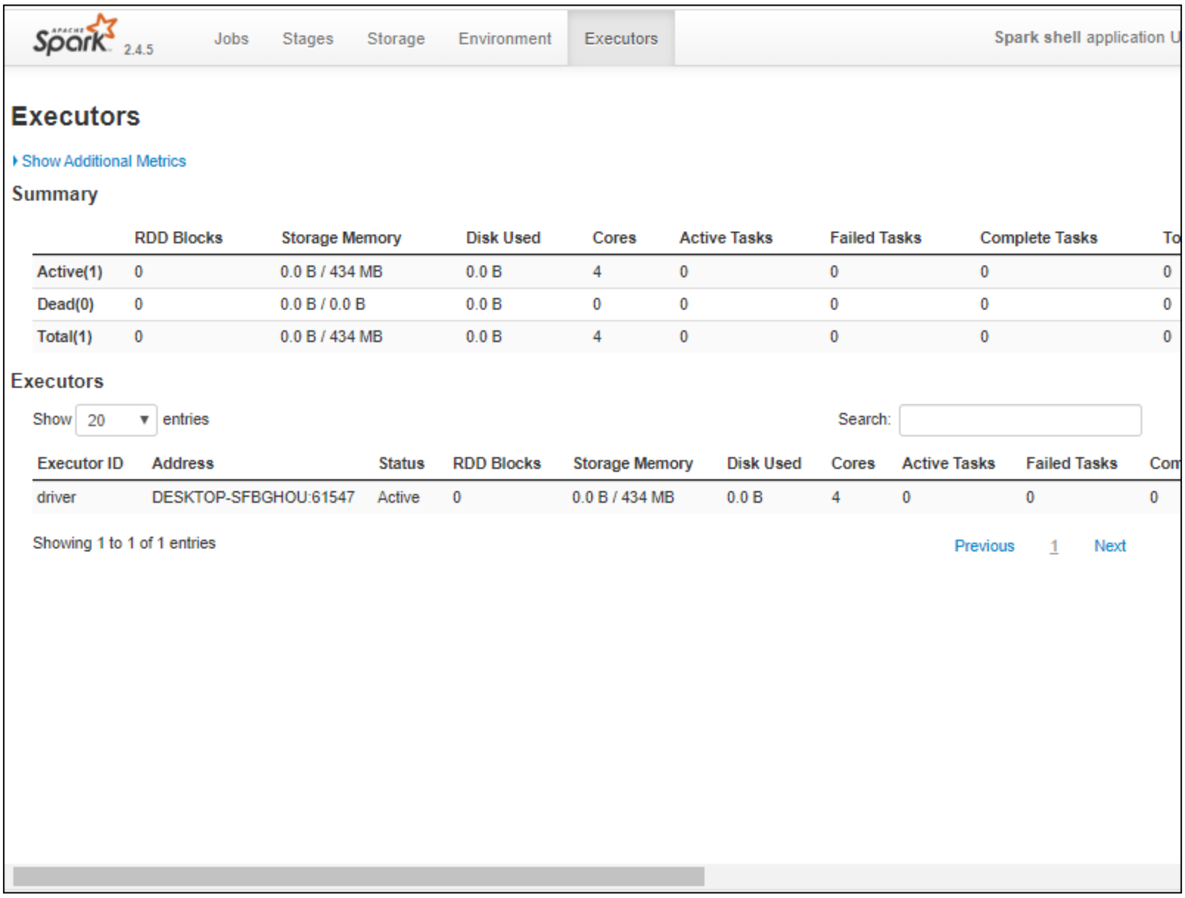

Administrator: Command Prompt - C:\Spark\spark-2.4.5-bin-hadoop2.7\bin\spark-shell

```
20/05/15 16:25:38 WARN NativeCodeLoader: Unable to load native-hadoop library for your p
latform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLev
el).
Spark context Web UI available at http://DESKTOP-SFBGHOU:4040
Spark context available as 'sc' (master = local[*], app id = local-1589552754132).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.5
      /_/

Using Scala version 2.11.12 (Java HotSpot(TM) Client VM, Java 1.8.0_251)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

4., Open a web browser and navigate to **http://localhost:4040/**.

5. You can replace **localhost** with the name of your system.

6. You should see an Apache Spark shell Web UI. The example below shows the *Executors* page.



Executors

Summary

	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	To
Active(1)	0	0.0 B / 434 MB	0.0 B	4	0	0	0	0
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0
Total(1)	0	0.0 B / 434 MB	0.0 B	4	0	0	0	0

Executors

Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Com
driver	DESKTOP-SFBGHOU:61547	Active	0	0.0 B / 434 MB	0.0 B	4	0	0	0

Showing 1 to 1 of 1 entries

7. To exit Spark and close the Scala shell, press **ctrl-d** in the command-prompt window.

Note: If you installed Python, you can run Spark using Python with this command:

pyspark

Exit using quit().

Test Spark

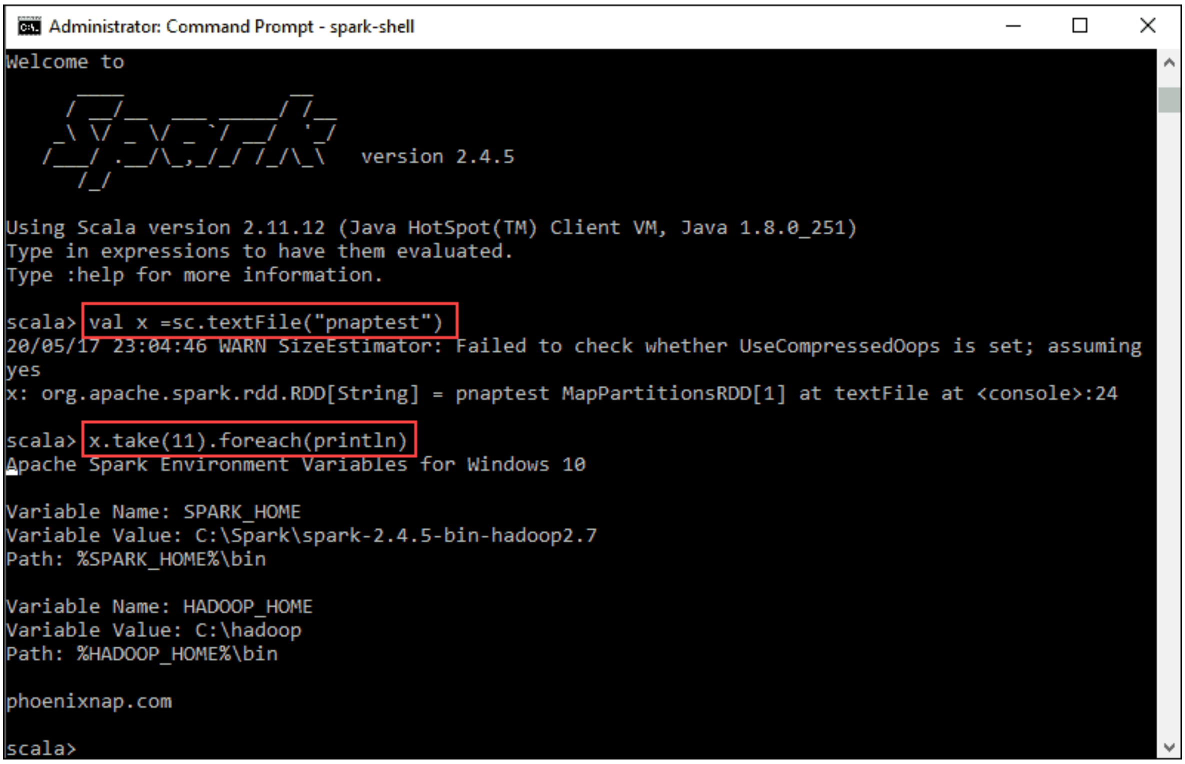
In this example, we will launch the Spark shell and use Scala to read the contents of a file. You can use an existing file, such as the *README* file in the Spark directory, or you can create your own. We created *pnaptest* with some text.

1. Open a command-prompt window and navigate to the folder with the file you want to use and launch the Spark shell.
2. First, state a variable to use in the Spark context with the name of the file. Remember to add the file extension if there is any.

```
val x =sc.textFile("pnaptest")
```

3. The output shows an RDD is created. Then, we can view the file contents by using this command to call an action:

```
x.take(11).foreach(println)
```



```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.5
      /_/

Using Scala version 2.11.12 (Java HotSpot(TM) Client VM, Java 1.8.0_251)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val x =sc.textFile("pnaptest")
20/05/17 23:04:46 WARN SizeEstimator: Failed to check whether UseCompressedOops is set; assuming
yes
x: org.apache.spark.rdd.RDD[String] = pnaptest MapPartitionsRDD[1] at textFile at <console>:24

scala> x.take(11).foreach(println)
Apache Spark Environment Variables for Windows 10

Variable Name: SPARK_HOME
Variable Value: C:\Spark\spark-2.4.5-bin-hadoop2.7
Path: %SPARK_HOME%\bin

Variable Name: HADOOP_HOME
Variable Value: C:\hadoop
Path: %HADOOP_HOME%\bin

phoenixnap.com

scala>
```

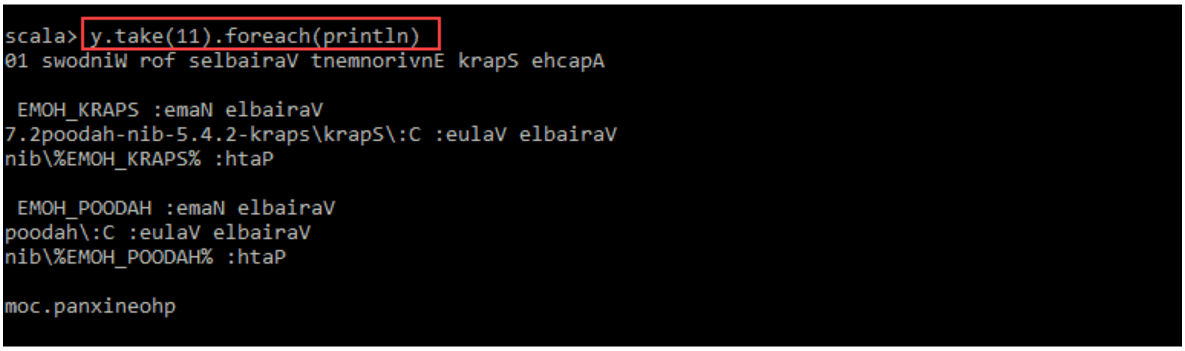
This command instructs Spark to print 11 lines from the file you specified. To perform an action on this file (**value x**), add another value **y**, and do a map transformation.

4. For example, you can print the characters in reverse with this command:

val y = x.map(_.reverse)

5. The system creates a child RDD in relation to the first one. Then, specify how many lines you want to print from the value **y**:

y.take(11).foreach(println)



```
scala> y.take(11).foreach(println)
01 swodniW rof selbairaV tnemnorivnE krapS ehcapA

 EMOH_KRAPS :emaN elbairaV
7.2poodah-nib-5.4.2-kraps\krapS\:C :eulaV elbairaV
nib\%EMOH_KRAPS% :htaP

 EMOH_POODAH :emaN elbairaV
poodah\:C :eulaV elbairaV
nib\%EMOH_POODAH% :htaP

moc.panxineohp
```

The output prints 11 lines of the *pnaptest* file in the reverse order. When done, exit the shell using **ctrl-d**.