Installing the required software

We will go through one by one the installation of required software.

Step 1. Install JAVA

We first need to update the local apt package index and then download and install the packages:

```
$ sudo apt update
```

Apache Spark needs JAVA to run. We can install JAVA by typing:

```
$ sudo apt install openjdk-8-jdk
```

Test your JAVA installation by typing:

```
$ java -version
```

You should see the following output:
OpenJDK version "1.8.0_191"
OpenJDK Runtime Environment (build 1.8.0_191-8u191-b12-2ubuntu0.18.04.1-b12)
OpenJDK 64-Bit Server VM (build 25.191-b12, mixed mode)

Step 2: Set Up Python

To begin the process, we'll install the dependencies we need for our Python programming environment from the Ubuntu repositories. Ubuntu 18.04 comes preinstalled with Python 3.6. We will use the Python package manager pip to install additional components a bit later.

Now, install pip and the Python header files, which are used by some of Jupyter's dependencies by typing:

```
$ sudo apt install python3-pip python3-dev
```

We can now move on to set up a Python virtual environment into which we'll install Jupyter.

Step 3: Create a Virtual Environment for Jupyter

Now that we have Python 3, its header files, and pip ready to go, we can create a Python virtual environment to manage our projects. We will install Jupyter into this virtual environment.

To do this, we first need access to the virtualenv command which we can install with pip. Install the package by typing:

```
$ sudo -H pip3 install virtualenv
```

The -H flag ensures that the security policy sets the home environment variable to the home directory of the target user.
With virtualenv installed, we can start forming our environment. Create and move into a directory where we can keep our project files. You can call this FIT.

```
$ mkdir ~/ FIT
```

```
$ cd ~/ FIT
```

Within the project directory, we'll create a Python virtual environment. For the purpose of
this tutorial, we'll call it jupyter.

```
$ virtualenv jupyter
```

This will create a directory called jupyter within your jupyter directory. Inside, it will install
a local version of Python and a local version of pip. We can use this to install and configure an isolated Python environment for Jupyter.

Before we install Jupyter, we need to activate the virtual environment. You can do that by typing:

```
$ source jupyter /bin/activate
```

Your prompt should change to indicate that you are now operating within a Python virtual environment. It will look something like this:

```
( jupyter ) X@X-VM:~/ FIT $
```

You're now ready to install Jupyter into this virtual environment.

Step 4: Install Jyputer

With your virtual environment active, install Jupyter with the local instance of pip.

Note: When the virtual environment is activated (when your prompt has (jupyter) preceding it), use pip instead of pip3 , even if you are using Python 3. The virtual environment's copy of the tool is always named pip , regardless of the Python version.

```
$ pip install jupyter
```

At this point, you've successfully installed all the software needed to run Jupyter. We can now start the Notebook server.

Step 5: Run Jupyter Notebook

You now have everything you need to run Jupyter Notebook! To run it, execute the following command:

```
( jupyter ) X@X-VM:~/ FIT $ jupyter notebook
```

A log of the activities of the Jupyter Notebook will be printed to the terminal. When you run Jupyter Notebook, it runs on a specific port number. The first Notebook you run will usually use port 8888 . To check the specific port number Jupyter Notebook is running on, refer to the output of the command used to start it:

Output:

```
[I 21:23:21.198 NotebookApp] Writing notebook server cookie secret to
/run/user/1001/jupyter/notebook_cookie_secret
[I 21:23:21.361 NotebookApp] Serving notebooks from local directory:
/home/ student / FIT
[I 21:23:21.361 NotebookApp] The Jupyter Notebook is running at:
[I 21:23:21.361 NotebookApp]
http://localhost: 8888 /?token= 1fefa6ab49a498a3f37c959404f7baf16b9a2eda3eaa6d7
2
[I 21:23:21.361 NotebookApp] Use Control-C to stop this server and shut down all
kernels (twice to skip confirmation).
```

[W 21:23:21.361 NotebookApp] No web browser found: could not locate runnable browser.
[C 21:23:21.361 NotebookApp]
Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost: 8888 /?token= 1fefa6ab49a498a3f37c959404f7baf16b9a2eda3eaa6d72

Step 6: Using Jupyter Notebook

You should now be connected to the jupyter notebook using a web browser. Jupyter Notebook is a very powerful tool with many features. This section will outline a few of the basic features to get you started using the Notebook. Jupyter Notebook will show all of the files and folders in the directory it is run from, so when you're working on a project make sure to start it from the project directory.

To create a new Notebook file, select **New > Python 3** from the top right pull-down menu:

Step 7: Install pyspark

Enter the following installation command in the new cell.

```
!pip install pyspark
```

To run the code, press CTRL+ENTER . You'll receive the following results:
You have now successfully installed pyspark .

Step 8: Test Spark Installation

Enter the following code in the new cell.

```
import pyspark
sc = pyspark.SparkContext(master="local", appName="Spark Test")
print (sc)
```

This script will import pyspark library, and create an instance of Sparkcontext with your master as local and app name (up to you) as parameters. To run the code, press CTRL+ENTER . You'll receive the following results:

Step 9: Install MongoDB

Ubuntu's official package repositories include an up-to-date version of MongoDB, which means we can install the necessary packages using apt . You can install the MongoDB package by typing:

```
$ sudo apt install -y mongodb
```

This command installs several packages containing the latest stable version of MongoDB, along with helpful management tools for the MongoDB server. The database server is automatically started after installation. You can verify it by running the command as below.

```
$ mongo
```

You should receive the following output:
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017

MongoDB server version: 3.6.3
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
http://docs.mongodb.org/
Questions? Try the support group
http://groups.google.com/group/mongodb-user
Server has startup warnings:
2019-03-11T16:09:08.520+1100 I STORAGE [initandlisten]
2019-03-11T16:09:08.520+1100 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2019-03-11T16:09:08.520+1100 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem
2019-03-11T16:09:09.984+1100 I CONTROL [initandlisten]
2019-03-11T16:09:09.984+1100 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2019-03-11T16:09:09.984+1100 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2019-03-11T16:09:09.984+1100 I CONTROL [initandlisten]
>
Now you are inside Mongo Shell which verifies that the MongoDB server is running. Type 'exit' to get out of the Mongo Shell.

How to run Jupyter Notebook?

You can start Jupyter Notebook by typing:

$ cd ~/ FIT

FIT $ source jupyter /bin/activate

(jupyter) X@X-VM:~/ FIT $

(jupyter) X@XVM:~/ FIT $ jupyter notebook

A log of the activities of the Jupyter Notebook will be printed to the terminal. When you run Jupyter Notebook, it runs on a specific port number. The first Notebook you run will usually use port 8888 . To check the specific port number Jupyter Notebook is running on, refer to the output of the command used to start it:

Output:
[I 21:23:21.198 NotebookApp] Writing notebook server cookie secret to /run/user/1001/jupyter/notebook_cookie_secret
[I 21:23:21.361 NotebookApp] Serving notebooks from local directory: /home/ student / FIT
[I 21:23:21.361 NotebookApp] The Jupyter Notebook is running at:
[I 21:23:21.361 NotebookApp]
http://localhost: 8888 /?token= 1fefa6ab49a498a3f37c959404f7baf16b9a2eda3eaa6d72
[I 21:23:21.361 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[W 21:23:21.361 NotebookApp] No web browser found: could not locate runnable browser.
[C 21:23:21.361 NotebookApp]
Copy/paste this URL into your browser when you connect for the first time,
to login with a token:

http://localhost: 8888 /?token= 1fefa6ab49a498a3f37c959404f7baf16b9a2eda3eaa6d72

That's it for this activity. Hope you have enjoyed setting up your own machine for processing big data.