Configuting Jupyter Notebook with Apache Spark.

Jupyter Notebook is a popular application that enables you to edit, run and share Python code into a web view. It allows you to modify and re-execute parts of your code in a very flexible way. That's why Jupyter is a great tool to test and prototype programs.

Latest Version of spark with pyspark API is alredy installed in my system. Next step is to install the jupitor notebook and connecting it with spark. Doing so, when we open spark using the terminal, we can open and access the spark programming platform in jupitor GUI directly.

We can install jupyter notebook as stand alone or as a part of complete conda distribution environment.
In my system jupytor notebook is available as a part of conda environment.

Instaling jupytor with conda:

Find the latest version of conda distribution environment and copy the installer bash script.
Website: https://www.anaconda.com/distribution/

Download the Anaconda Bash Script

```
$ cd /tmp
$ curl -O https://repo.anaconda.com/archive/Anaconda3-2019.03-Linux-x86_64.sh
```

Verify the Data Integrity of the Installer

```
$ sha256sum Anaconda3-2019.03-Linux-x86_64.sh
```

Run the Anaconda Script

```
$ bash Anaconda3-2019.03-Linux-x86_64.sh
```

Review the license agreement by pressing ENTER until you reach the end.
Once you agree to the license, you will be prompted to choose the location of the installation.
You can press ENTER to accept the default location, or specify a different location.

Give yes to the following output
Do you wish the installer to prepend the Anaconda3 install location to PATH in your /home/shyam/.bashrc ? [yes|no]

Activate Installation

```
$ source ~/.bashrc
```

Test Installation

```
$ conda list
```

Set Up Anaconda Environments

```
$ conda create --name my_env python=3
$ conda activate my_env
```

Install Jupyter Notebook – Standalone Instalation

```
$ pip install jupyter  # to install the jupyter notebook.
$ jupyter notebook     # to start the notebook
```

ctrl+c to quit the jupytor note book from terminal.

Configuring PySpark with Jupyter

There are two ways to get PySpark available in a Jupyter Notebook:

Configure PySpark driver to use Jupyter Notebook: running pyspark will automatically open a Jupyter Notebook

Load a regular Jupyter Notebook and load PySpark using findSpark package

First option is quicker but specific to Jupyter Notebook, second option is a broader approach to get PySpark available in your favorite IDE.

a. Configure PySpark driver

Update PySpark driver environment variables: add these lines to your ~/.bashrc (or ~/.zshrc) file.

```
export PYSPARK_DRIVER_PYTHON=jupyter
export PYSPARK_DRIVER_PYTHON_OPTS='notebook'
```

.bashrc is a hidden file in home directory to list it use:

```
$ ls -la ~/ | more
```

There should be a .bashrc on the first page. If not just create it with:

```
$ vi ~/.bashrc
```

Or edit it with:

```
$ gedit ~/.bashrc  # save it after saving the file and restart the terminal.
          or
$ vim ~/.bashrc   # source it using : source ~/.bashrc
```

Restart your terminal and launch PySpark again:

```
$ pyspark
```

b. Using FindSpark package

There is another and more generalized way to use PySpark in a Jupyter Notebook: use findSpark package to make a Spark Context available in your code.

findSpark package is not specific to Jupyter Notebook, you can use this trick in your favorite IDE too.

```
$ pip install findspark
$ jupyter notebook
```

Inside the jupyter notebook, initialize the pyspark with following code.

```
import findspark
findspark.init()
import pyspark
```

* this document is only created for project documentation purpose and referencing only. All the steps followed in this document is complety based on blog written by Charles Bochet . Link to the blog: https://www.sicara.ai/blog/2017-05-02-get-started-pyspark-jupyter-notebook-3-minutes