Installing Spark

Master M2 – Université Grenoble Alpes & Grenoble INP

2021

The document provides the basic instructions for installing and configuring Spark before the first lab session of the course "Data Management in Large Scale Distributed Systems".

The recommended method is to install Spark directly on your machine if you have a Linux system (or a Linux virtual machine). This method will also work on the computers of the lab rooms at Ensimag.

An alternative method that should work on any recent OS is to use a docker image.

The last method is using Google Colab, an online solution to execute python notebooks. This solution can be a temporary alternative if you are experiencing issues with all the other methods. However, it can only be a temporary solution in our opinion.

The three methods are described below.

1 Native installation of Spark on a Linux machine

Please find below the instructions to install Spark on a recent Linux machine. These instructions are valid both for your laptop and for the machines of the lab rooms.

The instructions are for Spark 3.0.1. This is not the most recent version of Spark but it includes all the features you need for this course. Installing newer version of Spark has not been tested in the lab rooms.

1. Download the latest already compiled version of Spark here: `https://www.apache.org/dyn/closer.lua/spark/spark-3.0.1/spark-3.0.1-bin-hadoop2.7.tgz`

2. Extract the downloaded archive:

   ```
   tar zxvf spark-3.0.1-bin-hadoop2.7.tgz
   ```

3. Configure the required environment variables in the file `$HOME/.bashrc` by adding the following lines at the beginning of the file[1]:

   ```
   export SPARK_HOME=PATH_TO_DIR/spark-3.0.1-bin-hadoop2.7
   ```

[1] To open this file, simply run `nano ~/.bashrc` in a terminal

```
export PYTHONPATH="${SPARK_HOME}/python/:$PYTHONPATH"
export PYTHONPATH="${SPARK_HOME}/python/lib/py4j-0.10.9-src.zip:$PYTHONPATH"

export PATH=${SPARK_HOME}/bin:$PATH
```

where `PATH_TO_DIR` correspond to the directory where your stored Spark.

4. Start a new terminal to make your changes active
5. In the new terminal, launch `pyspark` to check that everything works correctly

Troubleshooting

Spark only works with Java 8 to 11. If after executing the previous commands, you experience some problems, it might be that the default Java version in your system is newer than this.

In the lab rooms, you can select the correct Java version to be used by adding the following line to the file `$HOME/.bashrc`:

```
export JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64/"
```

About Scala

To use Scala on your laptop, in addition to installing Spark, you will need to install `sbt`, which is a project builder for Scala projects.

To this end, simply follow the instructions here: `https://www.scala-sbt.org/1.x/docs/Installing-sbt-on-Linux.html`. We recommend using the DEB or RPM package if possible.

2 Installing Spark using a Docker container

If for any reason, you are not able to install Spark directly on your machine, an alternative is to use a Docker container.

2.1 Installing Docker

Docker is a container technology. It allows running images of already installed and configured software, in an isolated environment on top of the operating system.

Here are the instructions to install Spark on Ubuntu. Docker can be installed on other OSes, even MacOS or Windows (we let you look for the corresponding documentation in these cases). The main steps of the installation process are:

1. Installing and configuring Docker

2. Downloading the Spark image
3. Testing that everything works well

2.1.1 Installing and configuring Docker

We will use Docker CE (Community Edition). To install Docker CE on Ubuntu, please follow the instructions here: `https://docs.docker.com/install/linux/docker-ce/ubuntu/#install-docker-ce`

On the same page you can find instructions for other Linux distributions.

By default, Docker containers can be launched only by the *root* user. This can be inconvenient. To allow a normal user to run a docker container, follow these instructions: `https://docs.docker.com/install/linux/linux-postinstall/`

Once you are done with installing Docker, you should test that it works by running the *hello world* of docker in a terminal:

```
docker run hello-world
```

If everything works, congrats, you are almost done !! If you want to know more about Docker, you may start from here: `https://docs.docker.com/engine/docker-overview/`

2.1.2 Downloading the Spark container image

Once Docker is installed, you should pull the Docker image containing Spark that we will use. To do so, run in a terminal:

```
docker pull jupyter/pyspark-notebook
```

WARNING: The image is big. It will take time.

2.1.3 Testing that everything works well

To test that you are able to run Spark on your machine, run the following command in a terminal:

```
docker run -it --rm -p 4040:4040 jupyter/pyspark-notebook \
       /usr/local/spark-3.0.1-bin-hadoop3.2/bin/pyspark
```

At this point, Spark is running on your machine and you have started the Python interactive console (pyspark).

You can run your first Spark command by checking the default level of parallelism used by Spark on your machine, by typing in pyspark:

```
print(sc.defaultParallelism)
```

You can probably even connect to the Spark Web UI here: `http://localhost:4040/`

To exit pyspark and terminate Spark, simply type `Ctrl-D`

3 Using Spark with Google Colab

A last solution to work with Spark is to use Google Colab. Please refer to this short introduction to start using Spark with Google Colab: `https://colab.research.google.com/drive/1-co8gEHx_EJLURFWfw0WZq1uik0uRfqC?usp=sharing`.