Setting Up *LOCAL* Apache Spark Environment in the SNIC Cloud

These are not instructions for setting up a Spark Cluster - although they will be helpful in doing so.

Be careful with quotes and dashes if copy-pasting from the PDF! (you are recommended best to retype).

There are many ways to setup Spark - I've designed these instructions so we can learn about Linux administration a little, and install only what we need for sections A and B.

Login to the SNIC dashboard.

Create a new Instance:

Source:
Select Boot Source: Image
Create New Volume: **No** (note: otherwise we only have 3GB for the root FS)
Ubuntu 16.04 LTS

Flavor:
ssc.small (10GB disk)

Associate a floating IP.

SSH into your instance from your local machine, but forward port 8888 when you do:

```
$ ssh -L 8888:localhost:8888 ubuntu@<your-floating-ip>
```

(This way, when we start the jupyter notebook on our instance later on, we can access it at http://localhost:8888)

Preliminaries:

Fix host resolution issue:
```
$ sudo nano /etc/hosts
```
Add this line (replacing ben-lab-spark) with your hostname to the file if it is not there:
```
127.0.0.1 ben-lab-spark
```

Save and close.

(Explanation: https://en.wikipedia.org/wiki/Hosts_(file))

Update package lists:

```
$ sudo apt-get update
```

Now we follow the instructions on the slides:

```
1.  Install Dependencies:
     a.  Java
          i.  $ sudo apt-get install default-jdk
         ii.  ...faff about with JAVA_HOME, etc. :-)
```

The “faff” is to set the JAVA_HOME variable so that PySpark knows where to find Java.

This is one way to do it:
First, find where Java was installed:

```
$ which java
/usr/bin/java
```

That’s likely a symlink, follow the chain of symlinks back to the installation path:

```
$ readlink -f /usr/bin/java
/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
```

Now set JAVA_HOME to the folder that contains the ‘bin’ folder, and ‘export’ it to all child processes:

```
$ export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre/
```

(One way to set it automatically each time we SSH in, is to add that command to the ~/.bash_profile file inside the instance - a list of commands to execute each time we start the shell).

Back to the slides:

~~b. Python (v3)~~
~~i. $ sudo apt-get install python3.6~~
(skip this step - python3 is already installed by default, we can check like this:

```
$ python3 --version
```

~~c. Scala (if you are using it)~~
~~i. $ sudo apt-get install scala~~
(skip this step, we don’t need to use Scala)

d. PySpark (for the driver node)
(first, we need to install the package manager for Python3 - ‘pip’)
i. $ sudo apt-get install python3-pip

Let’s check we’re using the latest version of pip:

```
$ pip3 --version
pip 8.1.1 from /usr/lib/python3/dist-packages (python 3.5)
```

That's old - and I had issues installing our packages with it, furthermore there are issues updating pip itself with Ubuntu's package manager (don't be tempted!) - https://github.com/pypa/pip/issues/5221 - so lets just invoke pip via python.

First, update pip using pip!

```
$ python3 -m pip install --upgrade pip
```

It will say `'Successfully installed pip-8.1.1'` - but don't be fooled..

```
$ python3 -m pip --version
pip 10.0.1 from /home/ubuntu/.local/lib/python3.5/site-packages/pip (python
3.5)
```

OK that's an up to date version, install pyspark:

```
$ sudo python3 -m pip install pyspark
```

And install jupyter too:

```
$ sudo python3 -m pip install jupyter
```

(we used 'sudo' because both installations seem to need root access).

(The remaining steps on the slide are not necessary for running Spark in local mode.)

Lets try it - start a Jupyter notebook:

```
$ jupyter notebook
```

Copy and paste the link into your web browser (on your local machine) as instructed:

http://localhost:8888/?token=xxxxxxxxx

(Remember we forwarded the port when we ssh'd into our server)
You should see your Jupyter in your browser!

Create a new notebook, and run the example from the first lecture:

https://github.com/benblamey/jupyters/blob/master/ben-spark-master/jupyter/Teaching/Lecture1_Example1_Simple.ipynb

We want to run our application in local mode (not a cluster) for sections A and B, so we need to configure our application:

```
.master("local")
```

When we run the example, we get an error (we can see in the console):

```
2018-04-27 09:34:03 WARN  TaskSetManager:66 - Lost task 0.0 in stage
0.0 (TID 0, localhost, executor driver): java.io.IOException: Cannot
run program "python": error=2, No such file or directory
     at java.lang.ProcessBuilder.start(ProcessBuilder.java:1048)
```

Sounds like Spark can't find python - we need to set PYSPARK_PYTHON.

Kill jupyter with Ctrl-C from the shell.

```
$ export PYSPARK_PYTHON=python3
```

(As before, we can add that line to `~/.bash_profile` so it runs each time).

Restart Jupyter and re-run the example to check that it works!
(You might need to restart the Python kernel.)