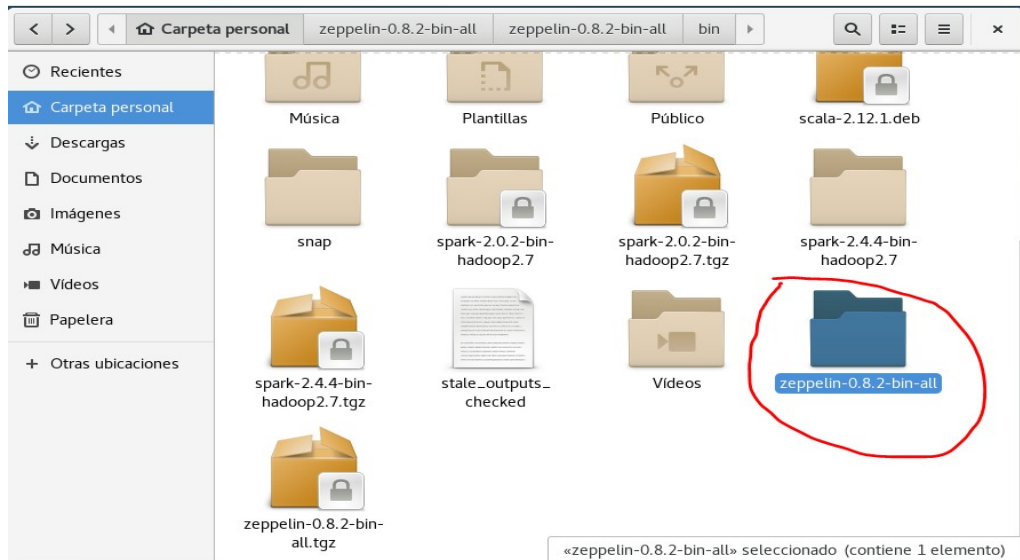


Pasos para configurar nuestro spark con el notebook de zeppelin

Descomprimos el zeppelin que tenemos en la maquina virtual.

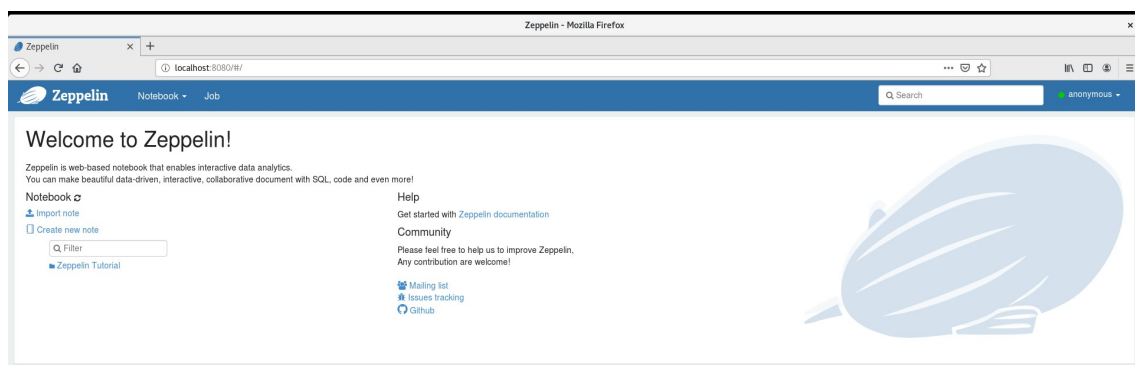


Una vez descomprimido vamos al terminal y desde el directorio donde zeppelin ejecutamos el siguiente comando: `bin/zeppelin-daemon.sh start`

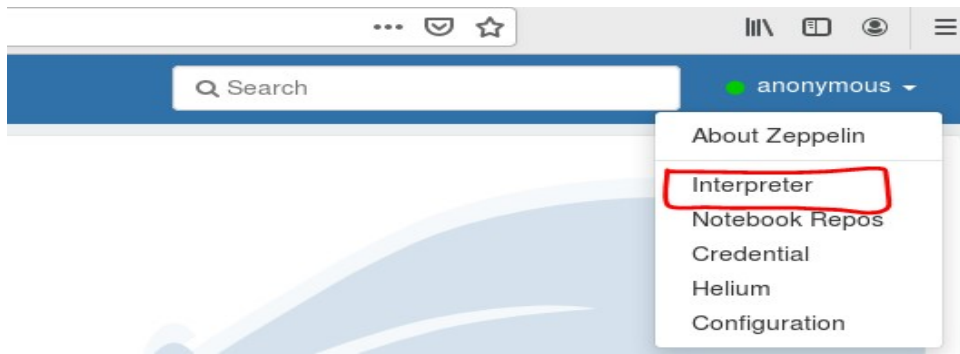
```
keepcoding@debian:~/zeppelin-0.8.2-bin-all/zeppelin-0.8.2-bin-all$ bin/zeppelin-daemon.sh start
Log dir doesn't exist, create /home/keepcoding/zeppelin-0.8.2-bin-all/zeppelin-0.8.2-bin-all/logs
Pid dir doesn't exist, create /home/keepcoding/zeppelin-0.8.2-bin-all/zeppelin-0.8.2-bin-all/run
Zeppelin start [ OK ]
keepcoding@debian:~/zeppelin-0.8.2-bin-all/zeppelin-0.8.2-bin-all$
```

Una vez hecho esto vamos al navegador y en la barra de búsqueda ponemos localhost:8080.

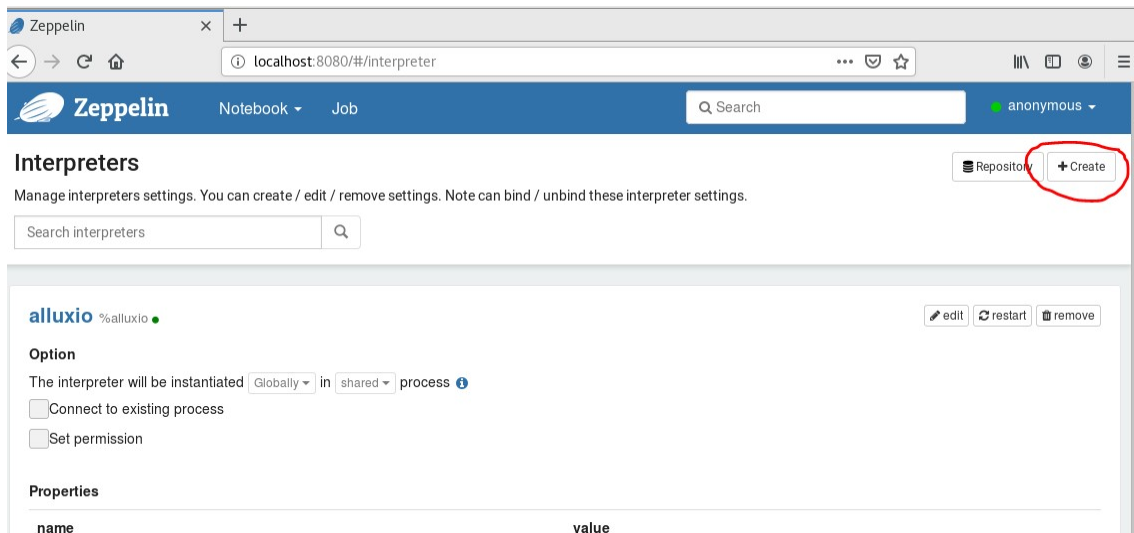
Ya podemos usar zeppelin, ahora vamos a configurarlo.



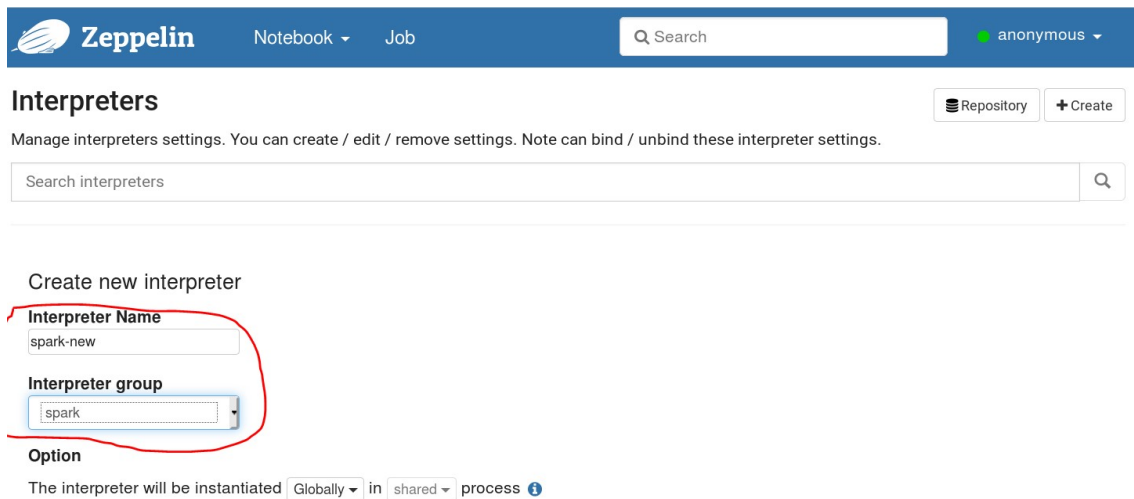
Vamos al desplegable de anonymous y pinchamos en interpreter.



Vamos a crear uno nuevo, para ello pulsamos créate



Le ponemos un nombre al intérprete que vamos a crear y lo metemos en el grupo spark.



En la parte de abajo antes de guardar tenemos que poner SPARK_HOME y la ruta a nuestra carpeta spark y una vez hecho pulsamos en Save y ya hemos creado el interprete.

Zeppelin

localhost:8080/#/interpreter

zeppelin.pyspark.usePython	<input checked="" type="checkbox"/>	<input type="checkbox"/>
zeppelin.spark.concurrentSQL	<input type="checkbox"/>	<input type="checkbox"/>
zeppelin.spark.enableSupportedVersionCheck	<input checked="" type="checkbox"/>	<input type="checkbox"/>
zeppelin.spark.importImplicit	<input checked="" type="checkbox"/>	<input type="checkbox"/>
zeppelin.spark.maxResult	<input type="text" value="1000"/>	<input type="checkbox"/>
zeppelin.spark.printREPLOutput	<input checked="" type="checkbox"/>	<input type="checkbox"/>
zeppelin.spark.sql.interpolation	<input type="checkbox"/>	<input type="checkbox"/>
zeppelin.spark.sql.stacktrace	<input type="checkbox"/>	<input type="checkbox"/>
zeppelin.spark.ui.hidden	<input type="checkbox"/>	<input type="checkbox"/>
zeppelin.spark.ui.WebUrl	<input type="text"/>	<input type="checkbox"/>
zeppelin.spark.useHiveContext	<input checked="" type="checkbox"/>	<input type="checkbox"/>
zeppelin.spark.useNew	<input checked="" type="checkbox"/>	<input type="checkbox"/>
SPARK_HOME	<input type="text" value="/opt/spark/spark-2.4.4-bin-hadoop2.7"/>	

Dependencies

artifact	exclude	action
<input type="text" value="groupid:artifactid:version or local file path"/>	<input type="text" value="(Optional) comma separated groupid:artifactid list"/>	<input type="button" value="+"/>

Si queremos ver que está bien creado en Iterpreters buscamos el que acabamos de crear y vemos que está bien.

Zeppelin - Mozilla Firefox

Zeppelin

Notebook Job

Search

anonymous

Interpreters

Repository + Create

Manage interpreters settings. You can create / edit / remove settings. Note can bind / unbind these interpreter settings.

spark-new

spark-new %spark-new, %sql, %dep, %pyspark, %ipySpark, %r

Option

The interpreter will be instantiated in process

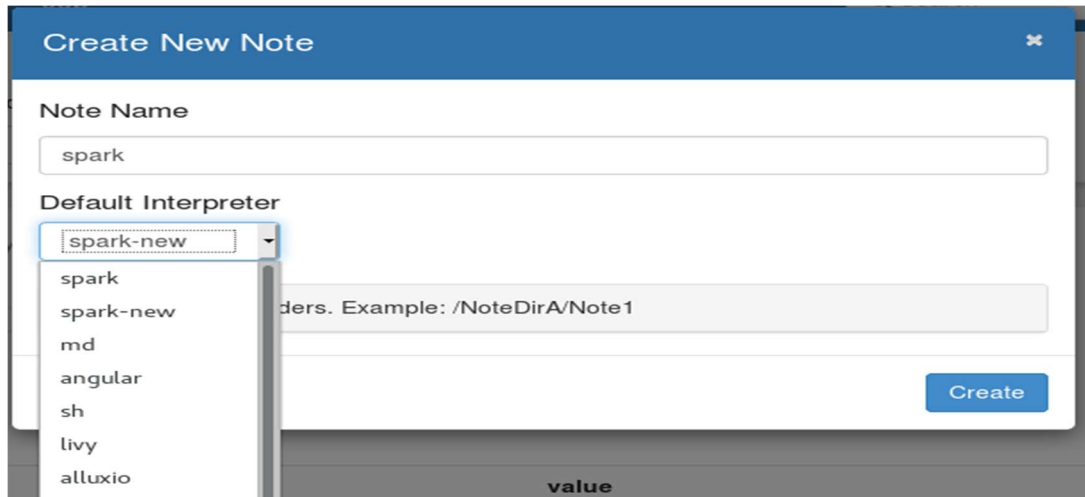
☐ Connect to existing process

☐ Set permission

Properties

name	value
SPARK_HOME	/opt/spark/spark-2.4.4-bin-hadoop2.7
args	
master	local[*]
spark.app.name	Zeppelin
spark.cores.max	

Ahora vamos a crear un nuevo notebook para realizar la práctica, para ello pinchamos en el desplegable que tiene notebook y pinchamos en create new note, le damos un nombre y buscamos el interprete que acabamos de crear.



Create New Note

Note Name

spark

Default Interpreter

spark-new

spark

spark-new

md

angular

sh

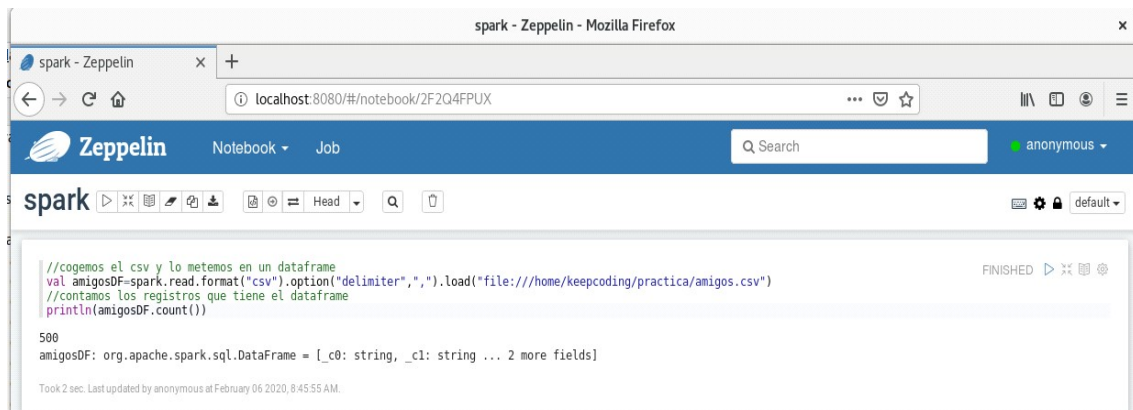
livy

alluxio

Example: /NoteDirA/Note1

Create

Una vez creado vamos a realizar la practica contando los registros de amigos.csv



spark - Zeppelin - Mozilla Firefox

spark - Zeppelin

localhost:8080/#/notebook/2F2Q4FPUX

Zeppelin

Notebook Job

Search

anonymous

spark

Head

```
//cogemos el csv y lo metemos en un dataframe
val amigosDF=spark.read.format("csv").option("delimiter",",").load("file:///home/keepcoding/practica/amigos.csv")
//contamos los registros que tiene el dataframe
println(amigosDF.count())
```

500

amigosDF: org.apache.spark.sql.DataFrame = [_c0: string, _c1: string ... 2 more fields]

Took 2 sec. Last updated by anonymous at February 06 2020, 8:45:55 AM.

FINISHED

como podemos ver ha salido todo correcto.