

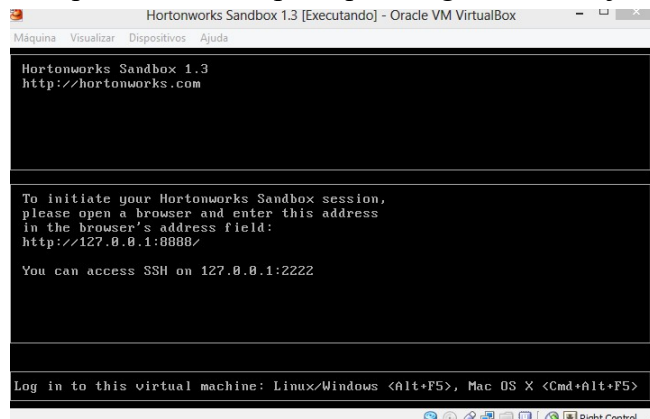
Professor: Cláudio Lúcio

Atividade para acesso ao Spark Shell (Python e Scala)

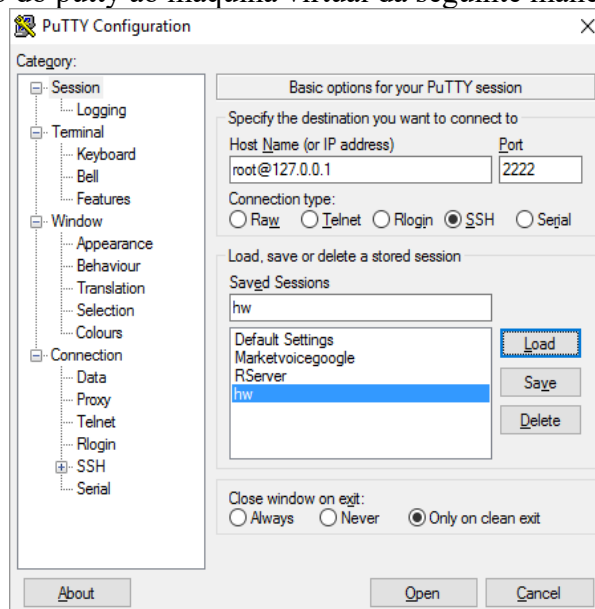
- 1) Para acesso ao Shell do Spark vamos usar a versão do Spark que vem na máquina virtual do HortonWorks:

Esta é a versão 2.3.2 do produto e pode ser obtida em:  
<http://hortonworks.com/products/ Hortonworks-Sandbox/>

- 2) Inicialize a máquina virtual. Espere que a seguinte tela seja exibida:



- 3) A recomendação é usar um cliente para acesso SSH ao servidor do spark. Baixe a ferramenta putty.exe;
- 4) Configure o acesso do putty ao máquina virtual da seguinte maneira:



- 5) Clique em “Open” e então digite a senha do Root: hadoop

```
root@sandbox:~  
Using username "root".  
root@127.0.0.1's password:  
Last login: Fri Dec 23 16:56:09 2016 from 10.0.2.2  
[root@sandbox ~]#
```

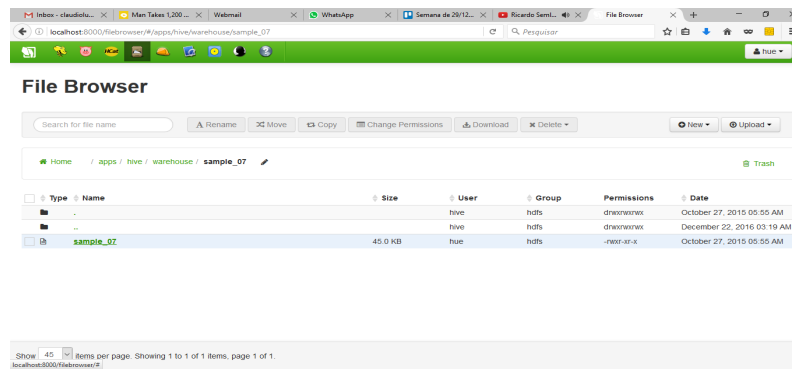
- 6) Primeiramente vamos interagir com a interface python para o Spark, para tal vamos acessar o bin/pyspark:
- 7) Digite os seguintes comandos:
  - `cd /usr/hdp/2.3.2.0-2950/spark/bin`
  - `pyspark`

```
root@sandbox:/usr/hdp/2.3.2.0-2950/spark/bin  
16/12/26 17:25:39 WARN SparkConf: The configuration key 'spark.yarn.applicationMaster.waitTries' has been deprecated as of Spark 1.3 and and may be removed in the future. Please use the new key 'spark.yarn.am.waitTime' instead.  
16/12/26 17:25:39 WARN SparkConf: The configuration key 'spark.yarn.applicationMaster.waitTries' has been deprecated as of Spark 1.3 and and may be removed in the future. Please use the new key 'spark.yarn.am.waitTime' instead.  
16/12/26 17:25:39 INFO Executor: Starting executor ID driver on host localhost  
16/12/26 17:25:40 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 53908.  
16/12/26 17:25:40 INFO NettyBlockTransferService: Server created on 53908  
16/12/26 17:25:40 INFO BlockManagerMaster: Trying to register BlockManager  
16/12/26 17:25:40 INFO BlockManagerMasterEndpoint: Registering block manager localhost:53908 with 265.4 MB RAM, BlockManagerId(driver, localhost, 53908)  
16/12/26 17:25:40 INFO BlockManagerMaster: Registered BlockManager  
Welcome to  
  
██████████ version 1.4.1  
  
Using Python version 2.6.6 (r266:84292, Jul 23 2015 15:22:56)  
SparkContext available as sc, HiveContext available as sqlContext.  
>>>
```

- 8) Veja que estamos com a versão 1.4.1 do Spark;
- 9) Por padrão as mensagens de log apresentadas no spark são muito detalhadas e podem dificultar as tarefas de manipulação de dados; Vamos ajustar os logs que serão apresentados. Vamos editar o arquivo `../conf/log4j.properties` :

- `cd ..`
- `cd conf/`
- `vi log4j.properties`
- Pressione a tecla "insert"
- Altere:





11) O seguinte programa Spark acessa o arquivo listado anteriormente:

- `lines=sc.textFile("/apps/hive/warehouse/sample_07/sample_07")`
- `lines.count()`
- `lines.first()`

```

root@sandbox:/usr/hdp/2.3.2.0-2950/spark
>>>
KeyboardInterrupt
>>>
[2]+  Stopped                  bin/pyspark
[root@sandbox spark]# vi conf/log4j.properties
[root@sandbox spark]# bin/pyspark
Python 2.6.6 (r266:84292, Jul 23 2015, 15:22:56)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-11)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Welcome to

Spark version 1.4.1

Using Python version 2.6.6 (r266:84292, Jul 23 2015 15:22:56)
SparkContext available as sc, HiveContext available as sqlContext.
>>> lines=sc.textFile("/apps/hive/warehouse/sample_07/sample_07")
>>> lines.count()
823
>>> lines.first()
u'00-0000\tAll Occupations\t134354250\t40690'
>>>

```

- Digite CTRL + Z

12) Vamos agora acessar a interface usando o Scala e executar a mesma tarefa:

- Digite `/bin/spark-shell`

```

spark-class sparkk spark-shell spark-sql spark-submit
[root@sandbox spark]# bin/spark-shell
Welcome to

Spark version 1.4.1

Using Scala version 2.10.4 (OpenJDK 64-Bit Server VM, Java 1.7.0_91)
Type in expressions to have them evaluated.
Type :help for more information.
Spark context available as sc.
SQL context available as sqlContext.

scala>

```

- Digite:

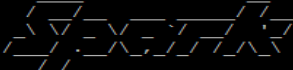
```

val lines = sc.textFile("/apps/hive/warehouse/sample_07/sample_07")
lines.count()
lines.first()

```

```
root@sandbox:/usr/hdp/2.3.2.0-2950/spark
```

Welcome to



version 1.4.1

Using Scala version 2.10.4 (OpenJDK 64-Bit Server VM, Java 1.7.0\_91)  
Type in expressions to have them evaluated.  
Type :help for more information.  
Spark context available as sc.  
SQL context available as sqlContext.

```
scala> val lines = sc.textFile("/apps/hive/warehouse/sample_07/sample_07")  
lines: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[1] at textFile at <console>:21  
  
scala> lines.count()  
res0: Long = 823  
  
scala> lines.first()  
res1: String = 00-0000 All Occupations 134354250 40690  
  
scala>
```