1_initialize_spark

February 21, 2022

0.0.1 Initilize spark - How to start a Spark program?

[1]:
```
# Import SparkContext and SparkConf

# a) SparkConf(loadDefaults=True, _jvm=None, _jconf=None)
#
# Configuration for a Spark application. Used to set various Spark
# parameters as key-value pairs.

# b) SparkContext(master=None, appName=None, sparkHome=None, pyFiles=None,␣
↪environment=None, batchSize=0, serializer=PickleSerializer(), conf=None,␣
↪gateway=None, jsc=None, profiler_cls=<class 'pyspark.profiler.
↪BasicProfiler'>)

# Main entry point for Spark functionality. A SparkContext represents the
# connection to a Spark cluster, and can be used to create L{RDD} and
# broadcast variables on that cluster.

from pyspark import SparkContext, SparkConf
```

[2]:
```
# Get the classname of SparkContext
print(SparkContext)
```

```
<class 'pyspark.context.SparkContext'>
```

[3]:
```
# Get the classname of SparkConf
print(SparkConf)
```

```
<class 'pyspark.conf.SparkConf'>
```

[4]:
```
# Initialize spark
conf = SparkConf().setAppName("MyFirstExample").setMaster("local[4]")
sc = SparkContext(conf=conf)
```

```
22/02/21 11:38:46 WARN Utils: Your hostname, ThinkCentre resolves to a loopback
address: 127.0.1.1; using 10.180.5.223 instead (on interface eno1)
22/02/21 11:38:46 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
address
22/02/21 11:38:47 WARN NativeCodeLoader: Unable to load native-hadoop library
```

```
for your platform… using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use
setLogLevel(newLevel).
```

[5]:
```
conf
```

[5]:
```
<pyspark.conf.SparkConf at 0x7f5127a49a60>
```

[6]:
```
sc
```

[6]:
```
<SparkContext master=local[4] appName=MyFirstExample>
```

0.0.2 So, what are the starting lines of your Spark Program?

[7]:
```
from pyspark import SparkContext, SparkConf
conf = SparkConf().setAppName("MyFirstApp").setMaster("local[4]")
sc = SparkContext(conf=conf)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/tmp/ipykernel_15677/1307338990.py in <module>
      1 from pyspark import SparkContext, SparkConf
      2 conf = SparkConf().setAppName("MyFirstApp").setMaster("local[4]")
----> 3 sc = SparkContext(conf=conf)

~/softwares/spark-3.0.2-bin-hadoop2.7/python/pyspark/context.py in␣
↪__init__(self, master, appName, sparkHome, pyFiles, environment, batchSize,␣
↪serializer, conf, gateway, jsc, profiler_cls)
    131                 " is not allowed as it is a security risk.")
    132
--> 133         SparkContext._ensure_initialized(self, gateway=gateway,␣
↪conf=conf)
    134         try:
    135             self._do_init(master, appName, sparkHome, pyFiles,␣
↪environment, batchSize, serializer,

~/softwares/spark-3.0.2-bin-hadoop2.7/python/pyspark/context.py in␣
↪_ensure_initialized(cls, instance, gateway, conf)
    336
    337                     # Raise error if there is already a running Spark␣
↪context
--> 338                     raise ValueError(
    339                         "Cannot run multiple SparkContexts at once; "
    340                         "existing SparkContext(app=%s, master=%s)"
```

```
ValueError: Cannot run multiple SparkContexts at once; existing␣
 ↪SparkContext(app=MyFirstExample, master=local[4]) created by __init__ at /tmp
 ↪ipykernel_15677/1572797919.py:3
```

[ ]: