

<https://zeppelin.apache.org/docs/0.8.1/interpreter/spark.html>

1 启动的时候 可以不在用户下 做 kinit 认证。。

```
[zeppelin@cdh19-247 ~]$ cat initkey.sh
kinit -kt /home/zeppelin/zeppelin_cdh19-247.keytab zeppelin/cdh19-247@master
[zeppelin@cdh19-247 ~]$ klist
Ticket cache: FILE:/tmp/krb5cc_1005
Default principal: zeppelin/cdh19-247@master

Valid starting    Expires          Service principal
2019-09-09T13:46:00  2019-09-10T13:46:00  krbtgt/master@master
renew until 2019-09-16T13:46:00
```

2

Configuration Setup

1. On the server that Zeppelin is installed, install Kerberos client modules and configuration, krb5.conf. This is to make the server communicate with KDC.
2. Set SPARK_HOME in [ZEPELIN_HOME]/conf/zeppelin-env.sh to use spark-submit (Additionally, you might have to set export HADOOP_CONF_DIR=/etc/hadoop/conf)
3. Add the two properties below to Spark configuration ([SPARK_HOME]/conf/spark-defaults.conf):

```
spark.yarn.principal
spark.yarn.keytab
```

NOTE: If you do not have permission to access for the above spark-defaults.conf file, optionally, you can add the above lines to the Spark Interpreter setting through the Interpreter tab in the Zeppelin UI.

4. That's it. Play with Zeppelin!

spark.submit.deployMode	client
spark.yarn.keytab	/home/zeppelin/zeppelin_cdh19-247.keytab
spark.yarn.principal	zeppelin/cdh19-247@master
spark.yarn.queue	bf_yarn_pool.development

3

```
Charset.forName("utf8")).split("\n"))
case class Bank(age: Integer, job: String, marital: String, education: String, balance: Integer)
val bank = bankText.map(s => s.split(";")).filter(s => s(0) != "\"age\"").map(
  s => Bank(s(0).toInt,
    s(1).replaceAll("\"", ""),
    s(2).replaceAll("\"", ""),
    s(3).replaceAll("\"", ""),
    s(5).replaceAll("\"", "").toInt
  )
)
bank.foreach(println)
org.apache.hadoop.ipc.RemoteException(java.io.IOException): Delegation Token can be issued only with Kerberos or web authentication
at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.getDelegationToken(FSNamesystem.java:7582)
at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.getDelegationToken(NameNodeRpcServer.java:551)
at org.apache.hadoop.hdfs.server.namenode.AuthorizationProviderProxyClientProtocol.getDelegationToken(AuthorizationProviderProxyClientProtocol.java:
at org.apache.hadoop.hdfs.protocolPB.ClientNameNodeProtocolServerSideTranslatorPB.getDelegationToken(ClientNameNodeProtocolServerSideTranslatorPB.ja
at org.apache.hadoop.hdfs.protocol.proto.ClientNameNodeProtocolProtos$ClientNameNodeProtocol$2.callBlockingMethod(ClientNameNodeProtocolProtos.java)
at org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:617)
at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:1073)
at org.apache.hadoop.ipc.Server$Handler$1.run(Server.java:2226)
```

hadoop.proxyuser.zepplin.groups

hadoop.proxyuser.zepplin.hosts

```
文件: core-site.xml
... .. @@ -137,6 +137,14 @@
137 137 <property>
138 138 <name>hadoop.proxyuser.livy.hosts</name>
139 139 <value>10.0.221.69</value>
140 140 </property>
141 + <property>
142 + <name>hadoop.proxyuser.zepplin.groups</name>
143 + <value>*</value>
144 + </property>
145 + <property>
146 + <name>hadoop.proxyuser.zepplin.hosts</name>
147 + <value>*</value>
148 + </property>
141 149 </configuration>
142 150
```

✓	执行服务 HDFS 上的命令 启动 已成功启动 HDFS 服务	HDFS	9月 9, 2:21:37 下午	56.45s
	▶ 启动 HDFS 服务	HDFS	9月 9, 2:21:37 下午	38.58s
	▶ Wait for NameNode(s) to begin responding to RPCs		9月 9, 2:22:16 下午	17.84s
✓	在 3 个服务上同时执行命令 启动 已成功完成 3 个步骤。		9月 9, 2:22:33 下午	39.75s
	▶ 执行服务 YARN (MR2 Included) 上的命令 启动	YARN (MR2 Included)	9月 9, 2:22:34 下午	39.67s
	▶ 执行服务 Sentry 上的命令 启动	Sentry	9月 9, 2:22:34 下午	35.85s
	▶ 执行服务 HBase 上的命令 启动	HBase	9月 9, 2:22:35 下午	35.23s
✓	在 4 个服务上同时执行命令 启动 已成功完成 4 个步骤。		9月 9, 2:23:13 下午	39.35s
	▶ 执行服务 Kafka 上的命令 启动	Kafka	9月 9, 2:23:13 下午	39.26s
	▶ 执行服务 Solr-2 上的命令 启动	Solr-2	9月 9, 2:23:14 下午	35.94s
	▶ 执行服务 Spark 上的命令 启动	Spark	9月 9, 2:23:14 下午	35.73s
	▶ 执行服务 Hive 上的命令 启动	Hive	9月 9, 2:23:14 下午	38.49s
▼	在 3 个服务上同时执行命令 启动 已完成 1/3 个步骤。		9月 9, 2:23:53 下午	
	▶ 执行服务 Oozie-2 上的命令 启动	Oozie-2	9月 9, 2:23:53 下午	<input type="button" value="中止"/>
	▶ 执行服务 Impala 上的命令 启动	Impala	9月 9, 2:23:53 下午	<input type="button" value="中止"/>
	▶ 执行服务 LIVY 上的命令 启动	LIVY	9月 9, 2:23:54 下午	33.85s
▼	在 2 个服务上同时执行命令 启动			
	○ 执行服务 Hue-2 上的命令 启动			
	○ 执行服务 ZEPPELIN 上的命令 启动			

4 ?????

```
org.apache.hadoop.yarn.exceptions.YarnException: Failed to submit application_1568018764289_0001 to YARN : User zeppelin cannot submit applications to queue root.zeppelin
at org.apache.hadoop.yarn.client.api.impl.YarnClientImpl.submitApplication(YarnClientImpl.java:271)
at org.apache.spark.deploy.yarn.Client.submitApplication(Client.scala:174)
at org.apache.spark.scheduler.cluster.YarnClientSchedulerBackend.start(YarnClientSchedulerBackend.scala:56)
at org.apache.spark.scheduler.TaskSchedulerImpl.start(TaskSchedulerImpl.scala:173)
at org.apache.spark.SparkContext.<init>(SparkContext.scala:190)
```

zeppelin-env.sh add ..

```
export ZEPPELIN_JAVA_OPTS="-Dspark.yarn.queue=bf_yarn_pool.development"
```

```
# export ZEPPELIN_JAVA_OPTS # Additional jvm options. for example
export ZEPPELIN_JAVA_OPTS="-Dspark.yarn.queue=bf_yarn_pool.development"
cores.max=16"
# export ZEPPELIN_MEM # Zeppelin JVM mem options Default
```

5

```
ERROR [2019-09-09 15:53:12, 191] ({rpc-server-3-1}
TransportRequestHandler.java[lambda$respond$3]:196) - Error sending result
StreamResponse{streamId=/jars/spark-interpreter-0.8.1.jar, byteCount=20918558,
body=FileSegmentManagedBuffer{file=/home/zeppelin/zeppelin/interpreter/spark/spark-
interpreter-0.8.1.jar, offset=0, length=20918558}} to /10.0.19.48:41664; closing
connection
```

```
io.netty.handler.codec.EncoderException: java.lang.NoSuchMethodError:
io.netty.channel.DefaultFileRegion.<init>(Ljava/io/File;JJ)V
    at
io.netty.handler.codec.MessageToMessageEncoder.write(MessageToMessageEncoder.java:107)
```

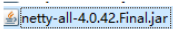
主要的异常信息如下:

```
java.lang.NoSuchMethodError: io.netty.channel.DefaultFileRegion.<init>(Ljava/io/File;JJ)V
```

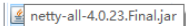
异常分析:

出现NoSuchMethodError这种问题, 一般是因为jar包冲突, 这个异常是因为netty包冲突导致的

spark 使用的netty jar包为



zeppelin使用的jar包为:



jar包版本不一致

解决办法:

将zeppelin/lib目录下netty包, 替换为spark的netty-all-4.0.42.Final.jar包, 重启zeppelin后, 再次执行notebook, 异常解决。

异常处理参考地址: <https://stackoverflow.com/questions/44801089/why-does-spark-fail-with-java-lang-nosuchmethoderror-io-netty-channel-default>

```
-rw-r--r-- 1 zeppelin zeppelin 30034345 9月  9 11:14 zeppelin-web-0.8.1.war
[zeppelin@cdh19-247 zeppelin]$ ll lib/netty-all-4.0.23.Final.jar
-rw-r--r-- 1 zeppelin zeppelin 1779991 9月  9 11:14 lib/netty-all-4.0.23.Final.jar
```

```
ll /opt/cloudera/parcels/SPARK2/lib/spark2
```

```
[zeppelin@cdh19-247 zeppelin]$ ll
/opt/cloudera/parcels/SPARK2/lib/spark2/jars/netty-*
-rw-r--r-- 1 root root 1330219 7月  13 2017
/opt/cloudera/parcels/SPARK2/lib/spark2/jars/netty-3.9.9.Final.jar
-rw-r--r-- 1 root root 2326492 7月  13 2017
/opt/cloudera/parcels/SPARK2/lib/spark2/jars/netty-all-4.0.43.Final.jar
```

解决办法:

```
rm /home/zeppelin/zeppelin/lib/netty-all-4.0.23.Final.jar
cp /opt/cloudera/parcels/SPARK2/lib/spark2/jars/netty-all-4.0.43.Final.jar
/home/zeppelin/zeppelin/lib/
```

<http://www.louisvv.com/archives/1744.html>

```
zeppelin@cdh19-247 zeppelin]$ rm /home/zeppelin/zeppelin/lib/netty-all-4.0.23.Final.jar
zeppelin@cdh19-247 zeppelin]$ cp /opt/cloudera/parcels/SPARK2/lib/spark2/jars/netty-all-4.0.43.Final.jar /home/zeppelin/zeppelin/lib/
zeppelin@cdh19-247 zeppelin]$
```

重启:

```
[zeppelin@cdh19-247 zeppelin]$ ./bin/zeppelin-daemon.sh restart
```

6

```
ERROR [2019-09-09 16:04:15,943] ({pool-2-thread-4} Job.java[run]:190) - Job failed
java.lang.OutOfMemoryError: GC overhead limit exceeded
```

Interpreters 调大 spark driver端内存

spark.cores.max	
spark.driver.memory	2g
spark.executor.memory	2g

7

```
Caused by: java.io.InvalidClassException:
org.apache.commons.lang3.time.FastDateParser; local class incompatible: stream
classdesc serialVersionUID = 2, local class serialVersionUID = 3
```

csv 不能读?

先换Textfile吧。。

```
val file = spark.read.textFile("hdfs://ns1/tmp/a.csv")
file.show(5)
```

```
import org.apache.commons.io.IOUtils
import java.net.URL
import java.nio.charset.Charset
```

```
// Zeppelin creates and injects sc (SparkContext) and sqlContext (HiveContext or
SqlContext)
// So you don't need create them manually

// load bank data
val bankText = sc.parallelize(
  IOUtils.toString(
    new URL("https://s3.amazonaws.com/apache-zeppelin/tutorial/bank/bank.csv"),
    Charset.forName("utf8")).split("\n"))

case class Bank(age: Integer, job: String, marital: String, education: String, balance:
Integer)

val bank = bankText.map(s => s.split(";")).filter(s => s(0) != "\"age\"").map(
  s => Bank(s(0).toInt,
    s(1).replaceAll("\\", ""),
    s(2).replaceAll("\\", ""),
    s(3).replaceAll("\\", ""),
    s(5).replaceAll("\\", ").toInt
  )
)
bank.foreach(println)
```