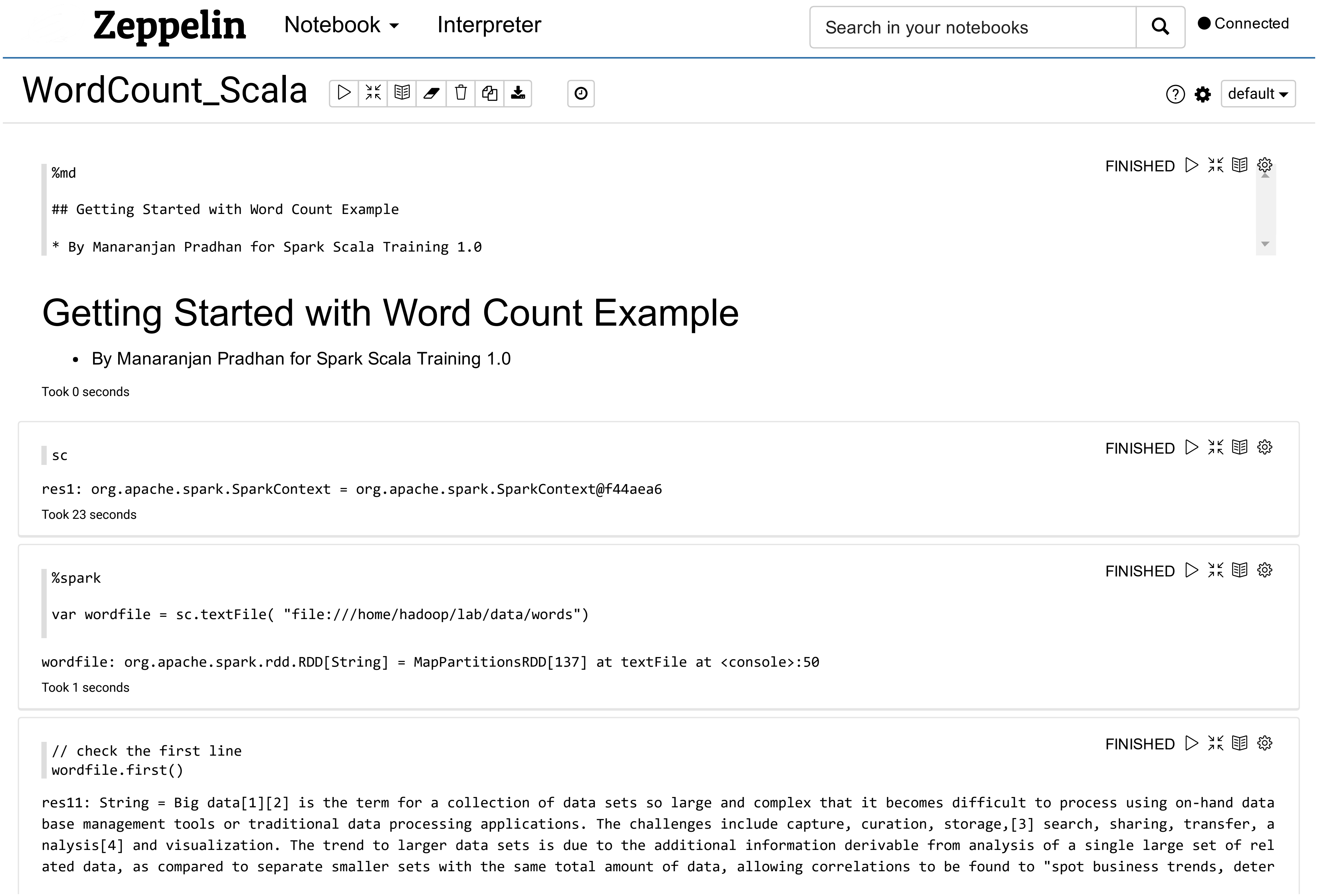

WordCount_Scala

```
%md

## Getting Started with Word Count Example

* By Manaranjan Pradhan for Spark Scala Training 1.0
```

Getting Started with Word Count Example

- By Manaranjan Pradhan for Spark Scala Training 1.0

Took 0 seconds

```
sc
```

```
res1: org.apache.spark.SparkContext = org.apache.spark.SparkContext@f44aea6
```

Took 23 seconds

```
%spark

var wordfile = sc.textFile( "file:///home/hadoop/lab/data/words")
```

```
wordfile: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[137] at textFile at <console>:50
```

Took 1 seconds

```
// check the first line
wordfile.first()
```

```
res11: String = Big data[1][2] is the term for a collection of data sets so large and complex that it becomes difficult to process using on-hand data
base management tools or traditional data processing applications. The challenges include capture, curation, storage,[3] search, sharing, transfer, a
nalysis[4] and visualization. The trend to larger data sets is due to the additional information derivable from analysis of a single large set of rel
ated data, as compared to separate smaller sets with the same total amount of data, allowing correlations to be found to "spot business trends, deter
```

```
mine quality of research, prevent diseases, link legal citations, combat crime, and determine real-time roadway traffic conditions."[5][6][7]A visual
ion created by IBM of Wikipedia edits. At multiple
```

Took 1 seconds (outdated)

READY

READY

FINISHED

```
// split the whole file into words
var words = wordfile.flatMap( line => line.split( " " ) )
```

```
words: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[10] at flatMap at <console>:32
```

Took 0 seconds (outdated)

FINISHED

```
// Print first 10  words
words.take( 10 ).foreach( println )
```

```
Big
data[1][2]
is
the
term
for
a
collection
of
data
```

Took 1 seconds (outdated)

FINISHED

```
// For each word let's split out ( word, 1 )
var word_one = words.map( word => ( word, 1 ) )
```

```
word_one: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[13] at map at <console>:34
```

Took 0 seconds (outdated)

FINISHED ▷

```
// Print first 10  words and counts
word_one.take( 10 ).foreach( println )
```

```
(Big,1)
(data[1][2],1)
(is,1)
(the,1)
(term,1)
(for,1)
(a,1)
(collection,1)
(of,1)
(data,1)
```

Took 1 seconds (outdated)

FINISHED ▷

```
// Now lets reduce to sum up and find total count against each word

var word_counts = word_one.reduceByKey( _+_ )
```

```
word_counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[16] at reduceByKey at <console>:37
```

Took 1 seconds (outdated)

FINISHED ▷

```
// print total counts for first few words
word_counts.take( 10 ).foreach( println )
```

```
((remote,1)
(created,1)
(consideration."[19],1)
(meteorology,,1)
(term,1)
(its,1)
(citations,,1)
(include,1)
(order,1)
(big,2)
```

Took 1 seconds (outdated)

FINISHED ▷

```
// Save the final output to a local file
word_counts.saveAsTextFile("file:///home/hadoop/lab/programs/results/wordcount")
```

Took 0 seconds (outdated)

READY