

## Spark to Neo4j

This blog post is mostly for my benefit and for the ability to go back and remember work that I have done. With that being said, this blog post talks about the Neo4j Spark Connector that allows a user to write from a Spark Data Frame to a Neo4j database. I will show the configuration settings within the Databricks Spark cluster, build some DataFrames within the Spark notebook and then write nodes to Neo4j and merge nodes to Neo4j.

For this blog post, I am using the [community Databricks site](#) to run a simple Spark cluster.

### Configuration:

For the cluster, I chose the Databricks 5.4 Runtime version which includes Apache Spark 2.4.3 and Scala 2.11. There are some Spark configuration options that configure the Neo4j user, Neo4j password and the Neo4j url for the bolt protocol.

Once the configurations were updated and the library added, we can restart the cluster.

*As a side note, the Notebook capability is a great feature.*

### **Creating Nodes:**

First I loaded up the Chicago crime data into a dataframe. Second was to convert the dataframe into a table. From the table, we'll run some SQL and load the results into Neo4j. These steps are shown below.

## Creating a Table

This next screen shows how we can get a spark context and connect to Neo4j. Once we have the context, we can execute several Cypher queries. For our example, we are going to delete existing data and creating a constraint and an index.

## Connecting to Neo4j and Running Queries

In this step, we can select data from the dataframe and load that data into Neo4j using the

Neo4jDataFrame.createNodes option.

Loading Crime Nodes into Neo4j

Merging Crime Beat Nodes

Loading Data into Neo4j Graph Algorithm

## Executing a Graph Algorithm

As you see, we can move data from Spark to Neo4j in a pretty seamless manner. The entire Spark notebook can be found [here](#).

In the next post, we will connect Spark to Kafka and then allow Neo4j to read from the Kafka topic.