Spark execution on EMR using Data Pipeline

Author:	Sri Govind Gutala
Environment:	Amazon Web Services
Date:	September 5, 2016
Document:	DataPipeline_Spark_On_EMR.docx

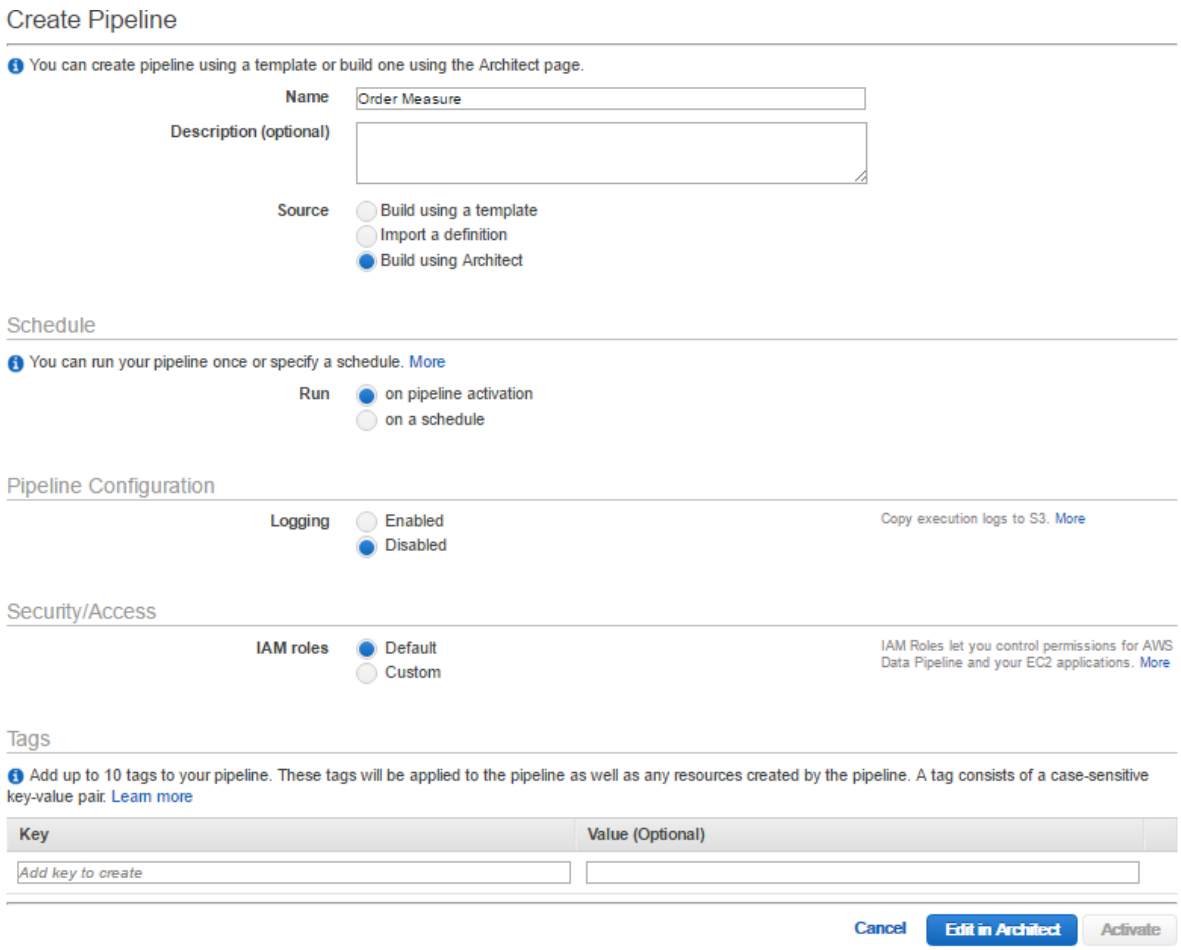
Table of Contents:

1. Introduction:

The main aim of this documentation is to execute a spark application on EMR, which loads data from S3, calculates the total sale and total number of orders placed for every product and stores the data back in s3. Once the data is on S3, it's loaded into Redshift table using RedshiftCopyActivity. Lets implement this using AWS Data Pipeline.

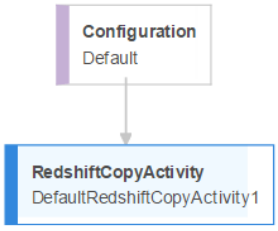
1. Create Data Pipeline

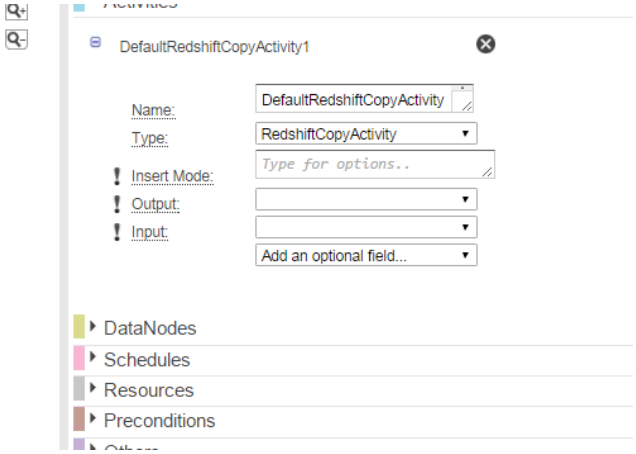
Click on Data Pipeline under Analytics section of AWS Console. Enter the details on the page as show below and click on "**Edit in Architect**".



2. Add RedshiftCopyActivity

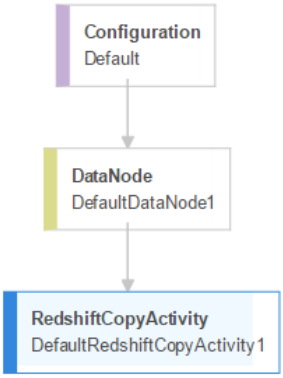
Click on “Add” button and select RedshiftCopyActivity. This activity will copy the data from source to Redshift table. Select Insert Mode as “**APPEND**”. This would change based on your use case.

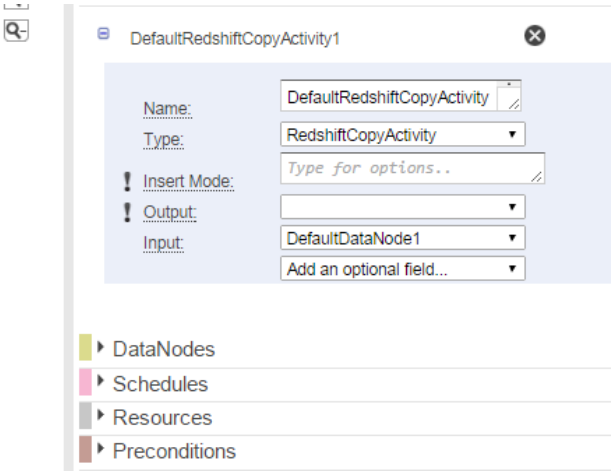




3. Add Redshift Properties

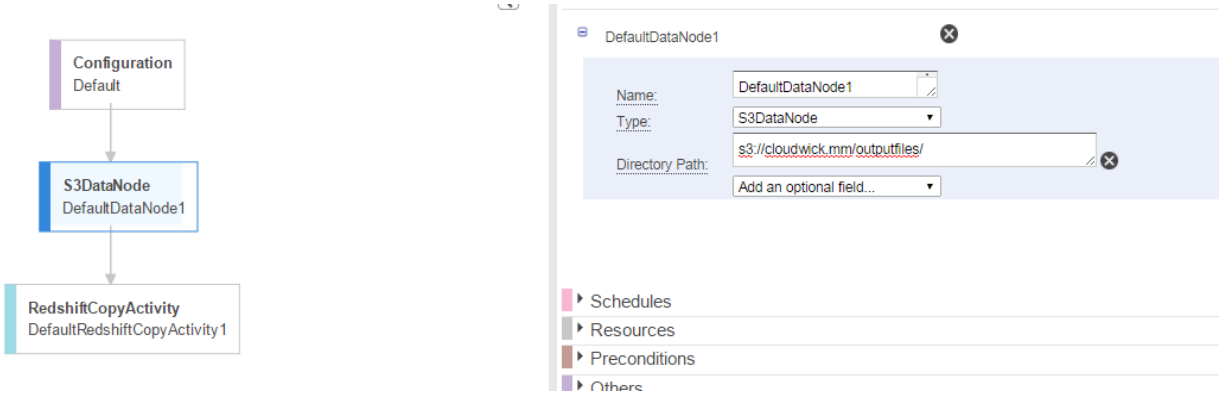
Here our source of data is S3. So click on “**input**” and select “Create New: DataNode”.





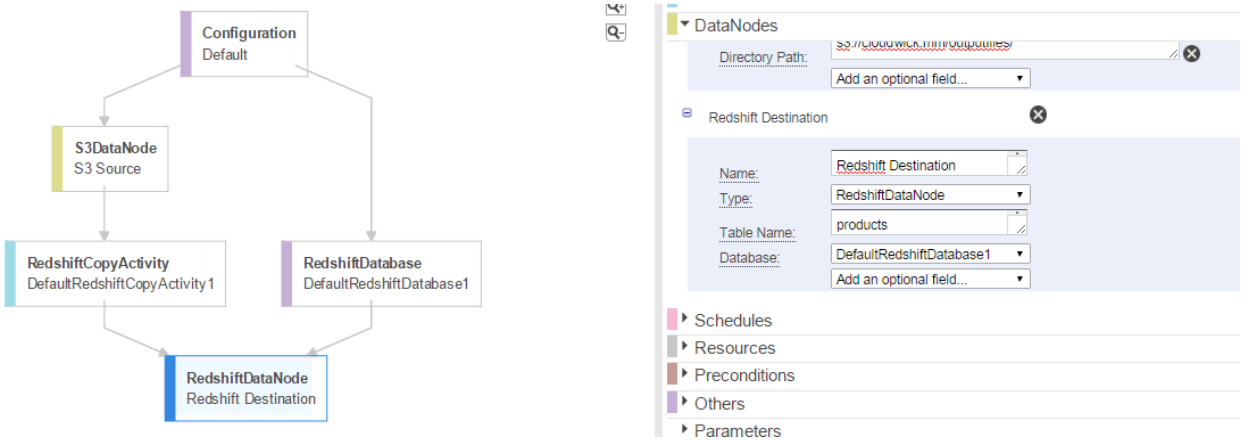
4. Add S3 DataNode properties

Click on "**DataNode**" that was created in above step and select properties as below



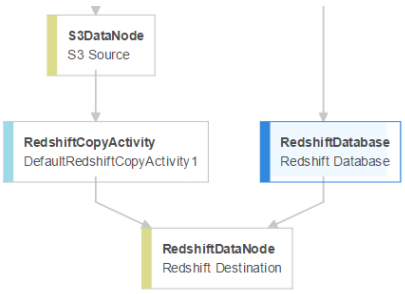
5. Add Redshift DataNode

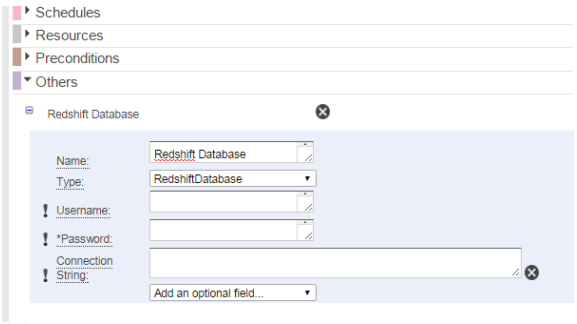
Click on RedshiftCopyActivity and select "**output**" then select "**Create New: DataNode**"
A new DataNode would be created. Click on that and provide a table name to load the data.
Now click on "**Database**" and select "**Create New: RedshiftDatabase**". Now the pipeline looks as below



6. Add Redshift Database Details

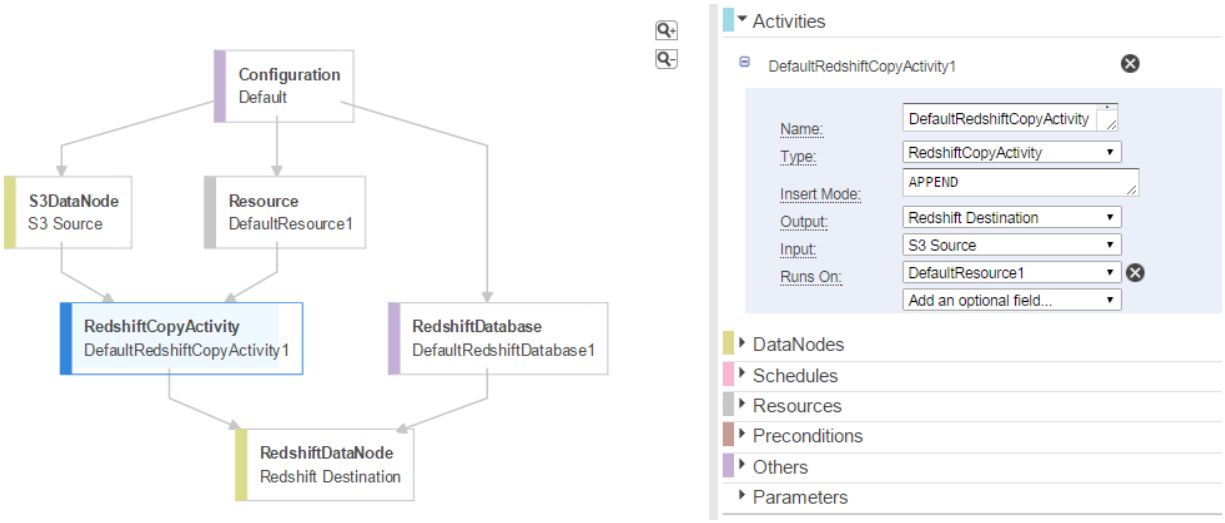
Click on **RedShiftDatabase** and provide the JDBC connections details. Connection string can be copied from the Redshift Cluster information in AWS Console.



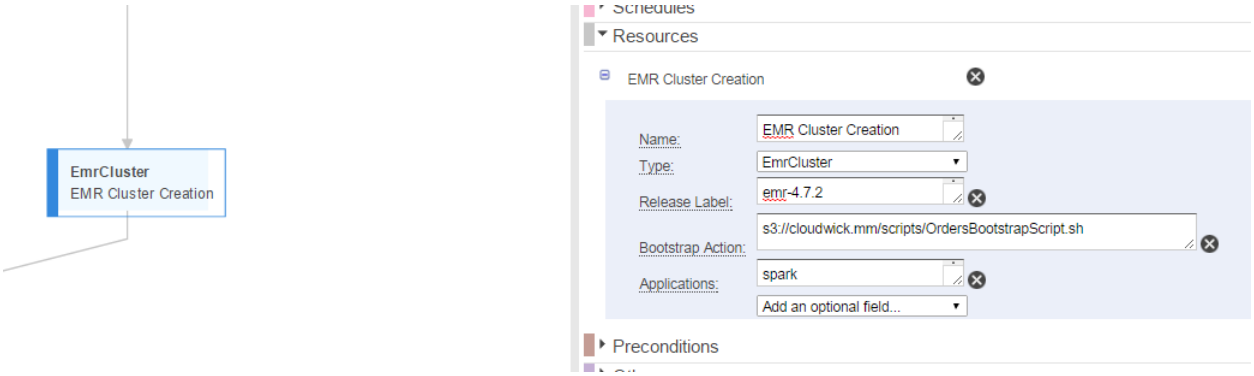


7. Add EMR Resource for computation

Now click on **RedshiftCopyActivity,** select **Runs On** from **Add an Optional field**. select **Create New: Resource** for Runs On field.



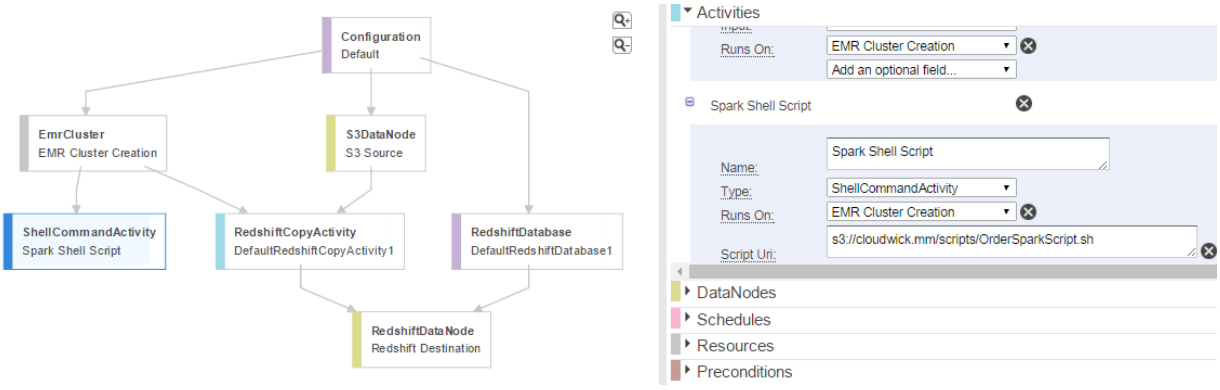
Click on **Resource** that was created in above step and select properties as shown below.



8. Add ShellCommandActivity

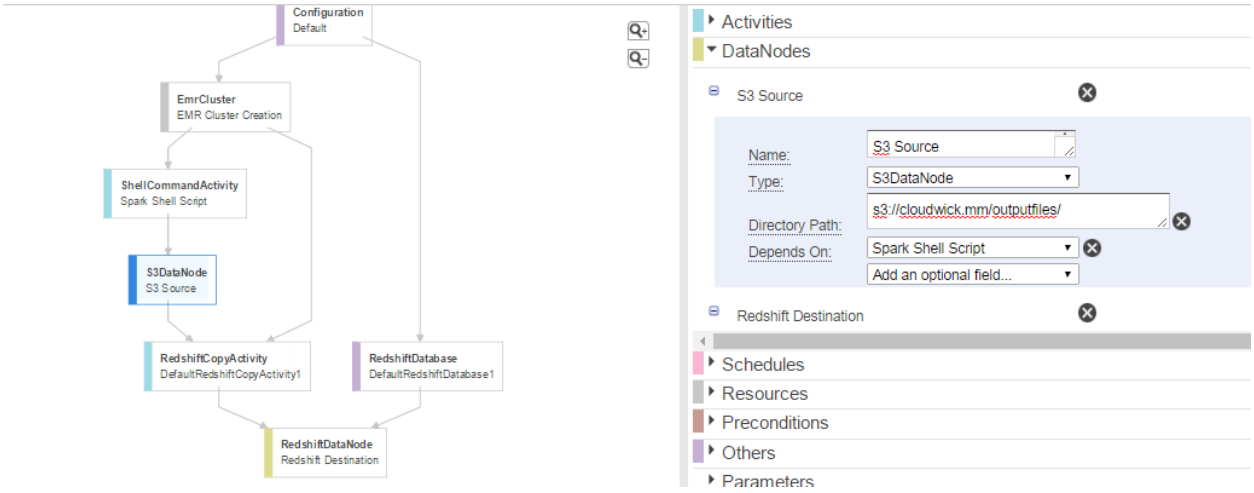
Now the cluster creation is completed. Now select "**ShellCommandActivity**" from Add button which will add the ShellCommandActivity on canvas and click on ShellCommandActivity and provide few properties as shown below. Then the data pipeline looks as below.

Here provide the spark script location which is on S3 in **Script Uri** property.



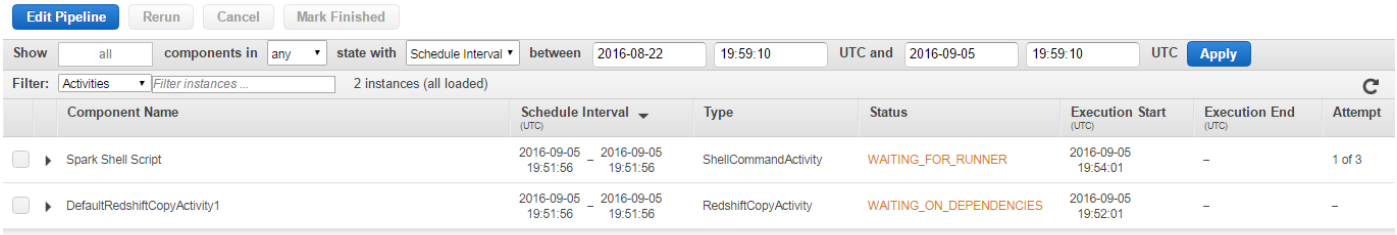
9. Add dependencies

Click on S3DataNode and select "Depends On" property as **Spark Shell Script** which is ShellCommandActivity name**.** Then the pipeline looks as below

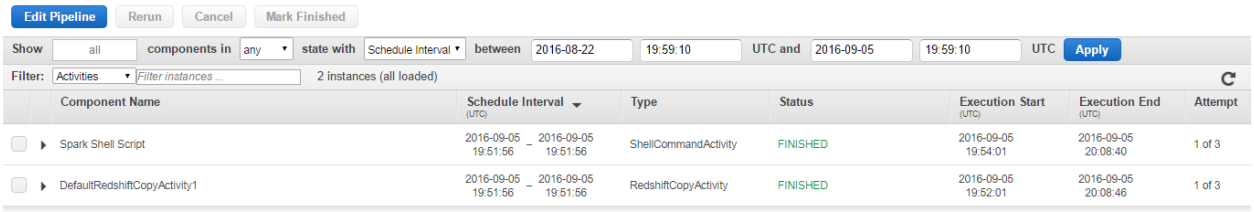


10. Activate Data Pipeline

Click Save and Activate the Data Pipeline. This would create 2 components executing with dependencies as shown below.



Once the EMR cluster is created, the shell script which has the spark submit command will be executed on EMR. Once the execution is successfully completed the pipeline status changes to "**FINISHED**".



2. Verify Redshift Table

We can verify the data in Redshift table by writing a query on SQL client.