

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:	3
1. Create Data Pipeline	3
2. Add RedshiftCopyActivity	4
3. Add Redshift Properties	4
4. Add S3 DataNode properties	5
5. Add Redshift DataNode	5
6. Add Redshift Database Details	6
7. Add EMR Resource for computation	6
8. Add ShellCommandActivity	7
9. Add dependencies	8
10. Activate Data Pipeline	8
2. Verify Redshift Table	9

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**”.

Create Pipeline

i You can create pipeline using a template or build one using the Architect page.

Name	<input type="text" value="Order Measure"/>
Description (optional)	<input type="text"/>
Source	<input type="radio"/> Build using a template <input type="radio"/> Import a definition <input checked="" type="radio"/> Build using Architect

Schedule

i You can run your pipeline once or specify a schedule. [More](#)

Run	<input checked="" type="radio"/> on pipeline activation <input type="radio"/> on a schedule
-----	--

Pipeline Configuration

Logging	<input type="radio"/> Enabled <input checked="" type="radio"/> Disabled	Copy execution logs to S3. More
---------	--	---

Security/Access

IAM roles	<input checked="" type="radio"/> Default <input type="radio"/> Custom	IAM Roles let you control permissions for AWS Data Pipeline and your EC2 applications. More
-----------	--	---

Tags

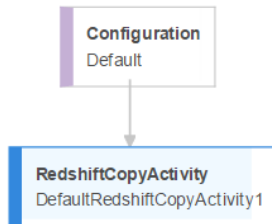
i Add up to 10 tags to your pipeline. These tags will be applied to the pipeline as well as any resources created by the pipeline. A tag consists of a case-sensitive key-value pair. [Learn more](#)

Key	Value (Optional)
<input type="text" value="Add key to create"/>	<input type="text"/>

Cancel [Edit in Architect](#) [Activate](#)

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.



DefaultRedshiftCopyActivity1

Name: DefaultRedshiftCopyActivity

Type: RedshiftCopyActivity

! Insert Mode: Type for options..

! Output:

! Input: Add an optional field...

▶ DataNodes

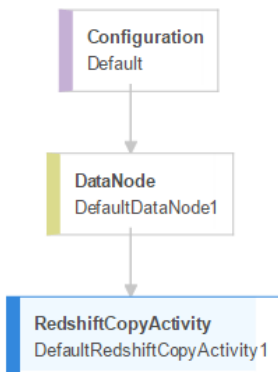
▶ Schedules

▶ Resources

▶ Preconditions

3. Add Redshift Properties

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



DefaultRedshiftCopyActivity1

Name: DefaultRedshiftCopyActivity

Type: RedshiftCopyActivity

! Insert Mode: Type for options..

! Output:

Input: DefaultDataNode1

Add an optional field...

▶ DataNodes

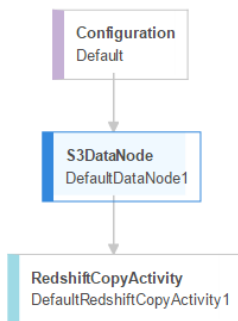
▶ Schedules

▶ Resources

▶ Preconditions

4. Add S3 DataNode properties

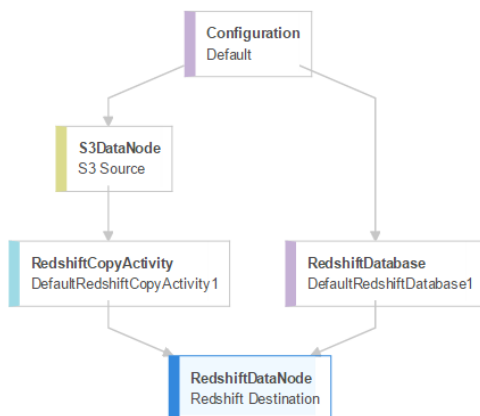
Click on “**DataNode**” that was created in above step and select properties as below



The screenshot shows the configuration panel for 'DefaultDataNode1'. The 'Name' field is 'DefaultDataNode1'. The 'Type' is 'S3DataNode'. The 'Directory Path' is 's3://cloudwick-mm/outputfiles/'. Below the main configuration area, there are expandable sections for 'Schedules', 'Resources', 'Preconditions', and 'Others'.

5. Add Redshift DataNode

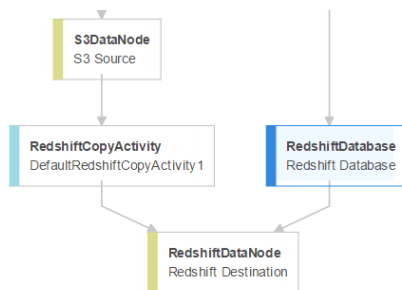
Click on RedshiftCopyActivity and select “**output**” then select “**Create New: DataNode**”
A new DataNode would be created. Click on that and provide a table name to load the data.
Now click on “**Database**” and select “**Create New: RedshiftDatabase**”. Now the pipeline looks as below



The screenshot shows the configuration panel for 'Redshift Destination'. The 'Directory Path' is 's3://cloudwick-mm/outputfiles/'. The 'Name' is 'Redshift Destination'. The 'Type' is 'RedshiftDataNode'. The 'Table Name' is 'products'. The 'Database' is 'DefaultRedshiftDatabase1'. Below the main configuration area, there are expandable sections for 'Schedules', 'Resources', 'Preconditions', 'Others', and 'Parameters'.

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.

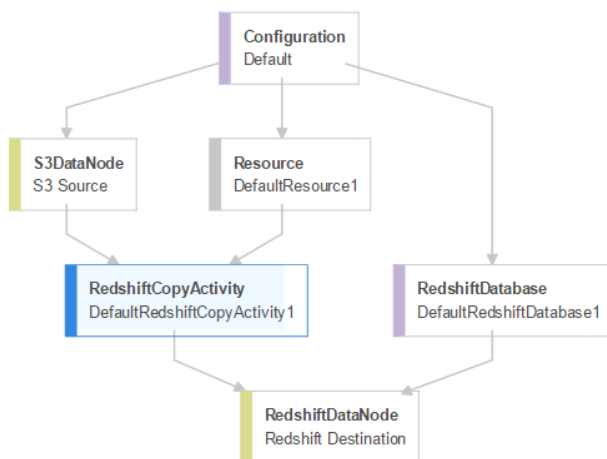


The screenshot shows a configuration window for a Redshift Database. The left sidebar lists categories: Schedules, Resources, Preconditions, and Others. The main area is titled 'Redshift Database' and contains the following fields:

- Name: Redshift Database
- Type: RedshiftDatabase
- Username: (empty field)
- *Password: (empty field)
- Connection String: (empty field)
- Bottom button: Add an optional field...

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.



Activities

DefaultRedshiftCopyActivity1

Name: DefaultRedshiftCopyActivity

Type: RedshiftCopyActivity

Insert Mode: APPEND

Output: Redshift Destination

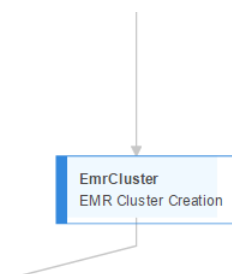
Input: S3 Source

Runs On: DefaultResource1

Add an optional field...

- DataNodes
- Schedules
- Resources
- Preconditions
- Others
- Parameters

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



Schedules

Resources

EMR Cluster Creation

Name: EMR Cluster Creation

Type: EmrCluster

Release Label: emr-4.7.2

Bootstrap Action: s3://cloudwick.mm/scripts/OrdersBootstrapScript.sh

Applications: spark

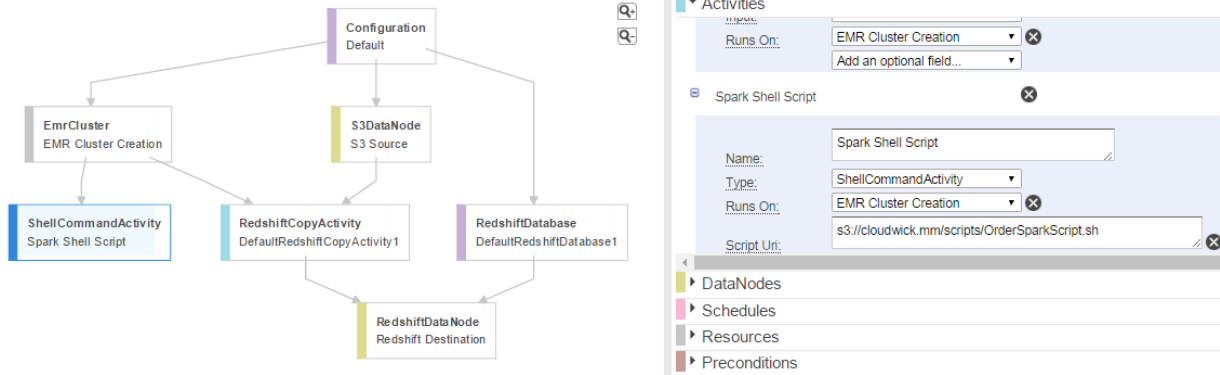
Add an optional field...

Preconditions

8. Add ShellCommandActivity

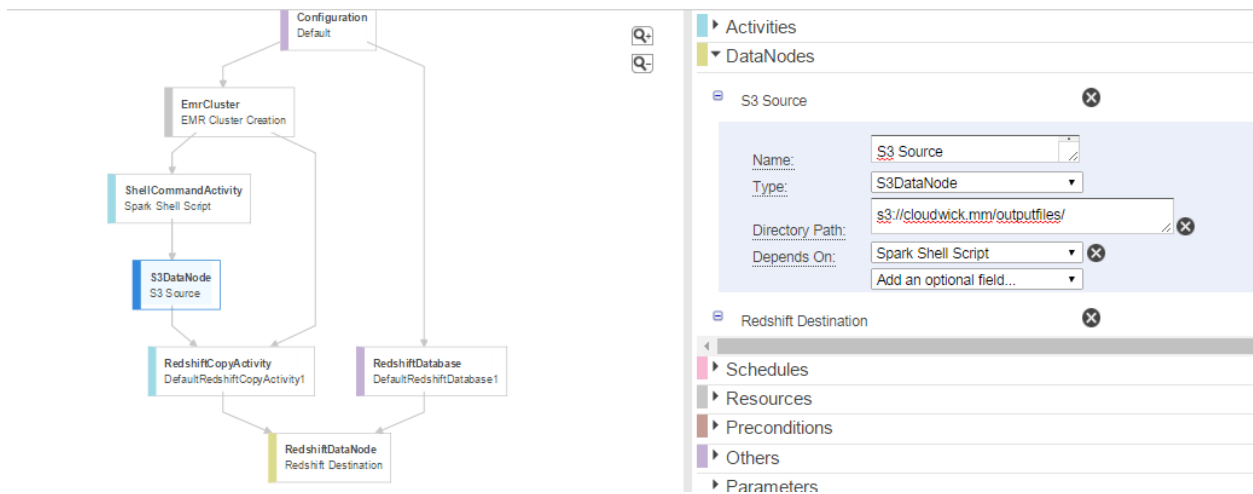
Now the cluster creation is completed. Now select “**ShellCommandActivity**” from Add button which will add the ShellCommandActivity on canvas and click on ShellCommandActivity and provide few properties as shown below. Then the data pipeline looks as below.

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.

Edit Pipeline

Rerun

Cancel

Mark Finished

Show

all

components in

any

state with

Schedule Interval

between

2016-08-22

19:59:10

UTC and

2016-09-05

19:59:10

UTC

Apply

Filter:

Activities

Filter instances ...

2 instances (all loaded)

	Component Name	Schedule Interval (UTC)	Type	Status	Execution Start (UTC)	Execution End (UTC)	Attempt
<input type="checkbox"/>	▶ Spark Shell Script	2016-09-05 19:51:56 - 2016-09-05 19:51:56	ShellCommandActivity	WAITING_FOR_RUNNER	2016-09-05 19:54:01	-	1 of 3
<input type="checkbox"/>	▶ DefaultRedshiftCopyActivity1	2016-09-05 19:51:56 - 2016-09-05 19:51:56	RedshiftCopyActivity	WAITING_ON_DEPENDENCIES	2016-09-05 19:52:01	-	-

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**”.

Edit Pipeline

Rerun

Cancel

Mark Finished

Show

all

components in

any

state with

Schedule Interval

between

2016-08-22

19:59:10

UTC and

2016-09-05

19:59:10

UTC

Apply

Filter:

Activities

Filter instances ...

2 instances (all loaded)

	Component Name	Schedule Interval (UTC)	Type	Status	Execution Start (UTC)	Execution End (UTC)	Attempt
<input type="checkbox"/>	▶ Spark Shell Script	2016-09-05 19:51:56 - 2016-09-05 19:51:56	ShellCommandActivity	FINISHED	2016-09-05 19:54:01	2016-09-05 20:08:40	1 of 3
<input type="checkbox"/>	▶ DefaultRedshiftCopyActivity1	2016-09-05 19:51:56 - 2016-09-05 19:51:56	RedshiftCopyActivity	FINISHED	2016-09-05 19:52:01	2016-09-05 20:08:46	1 of 3

2. Verify Redshift Table

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

SQL Workbench/J MMConnection - Default.wksp		
File Edit View Data SQL Macros Workspace Tools Help		
User=redshiftuser, Schema=public, URL=jdbc:redshift://datalakeredshiftcluster.cu3jvsgimzvz.us-we		
Statement 1		
1		
2 select * from products;		
3		
Result 1		
product_id	sales	quantity
16429400	42754	110
16619294	97548	165
16615933	12464	136
16471902	1781	8
16497025	2596	144
16753024	699	56
16346727	182	13
16704002	9427	35
16395266	2386	20
16425918	71	1
16616835	6067	14
16554562	186	11
16769121	11731	432
16487421	2422	14
16564305	7760	144

Cloudwick Confidential

39899 Balentine Drive, Suite 345, Newark, CA 94560 | sales@cloudwick.com | (855) 839-5650