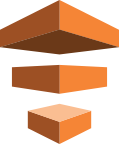
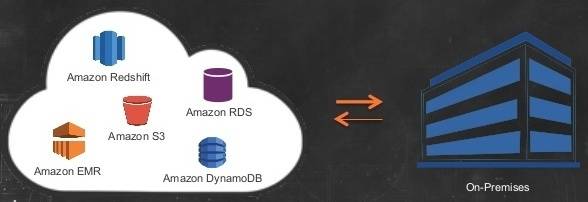
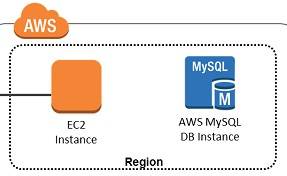
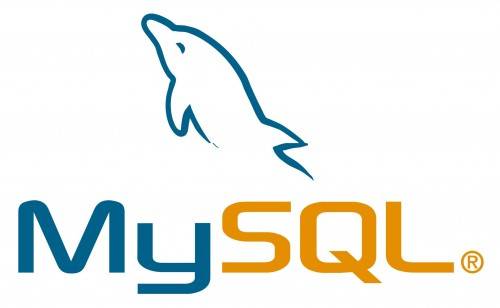
**Leveraging AWS Data Pipeline to Easily Move Relational Data between AWS, On-Premise or Other Services**

In today’s time of data-oriented applications, there is a constant change and challenge in requirements of data onboarding from and to varied sources. You may find yourself stuck between traditional and time-consuming methodologies of data migration. These may incur unnecessary costs and other overheads, eventually affecting the delivery. Amazon launched its Data Pipeline service to embrace this challenge to a considerable extent.

**What does AWS Data Pipeline do?**

Data Pipeline can be used to migrate data between varied data sources in a hustle free manner. In AWS’ own words, ‘*Data Pipeline is a web service that helps you reliably process and move data between different AWS compute and storage services, as well as on-premise data sources, at specified intervals.*‘ Not only this but, Data Pipeline extends its usability to create simple to complex ETL (extract, transform, load) workflows. You can schedule the execution of your workflows and move data loads between varied sources such as [AWS services](http://www.tothenew.com/devops-aws), on-premise servers, and other cloud providers.



**The Challenge**

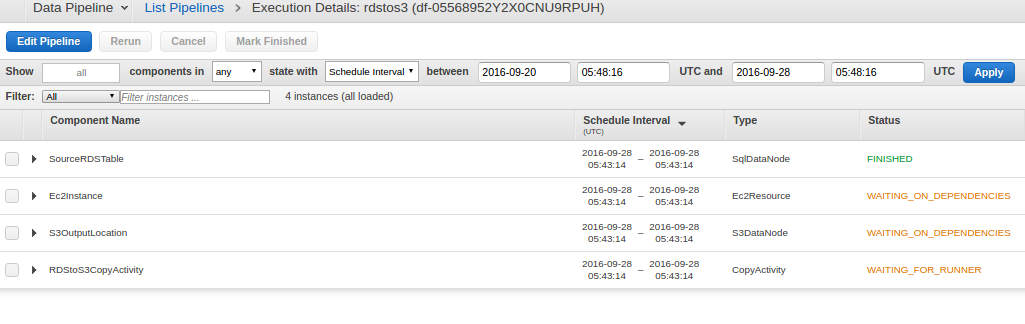
It may be that you need to move a table/s between a production MySQL database running on RDS and a non-production MySQL database running in EC2 or your on-premise server or any other cloud provider, for let’s say, analytical or BI purposes. I have an RDS MySQL instance and an EC2 instance with MySQL server running on it. For the sake of simplicity and understanding, I have created an example table ‘Persons’ in RDS instance with just one record in it. I need to move the record in MySQL database table on an EC2 instance. Going forward, we will see how we can leverage AWS Data Pipeline service to achieve this.

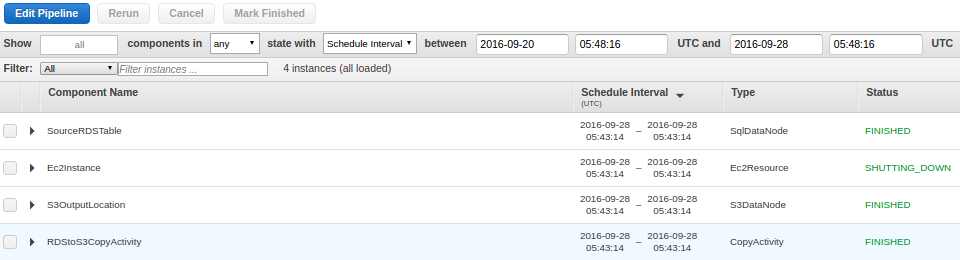
**The Solution**

We will start off by creating our first pipeline that will copy the records in a specified table to S3 bucket in a CSV file.

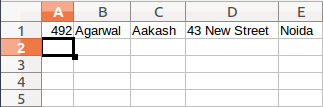
* *Moving records from RDS MySQL table to S3 bucket*

1. Go to the AWS console and open Data Pipeline service. Click on Create new pipeline.
2. Give a desired name and description for your pipeline and select ‘Build using a template.’ Under this, there will be different pre-built templates that AWS offers. Select ‘Full copy of RDS MySQL table to S3′.
3. We see that a new ‘Parameters’ block has appeared with fields that require inputs about the related RDS instance, S3 bucket, and EC2 instance. The EC2 instance, here, is launched at the time of pipeline activation and task runner runs on it. It is the task runner that performs activities that you mention in your pipeline. You may configure the pipeline to tear down the EC2 instance once the activities are finished.
4. Under ‘Schedule’ block, you may choose whether to run the pipeline as soon as it is activated or schedule it. It is an important feature and makes Data pipeline suitable for periodic or frequent executions of activities. Select ‘on pipeline activation.’
5. You may want to enable logging in case you need to troubleshoot pipeline failures.
6. Select ‘Default’ IAM roles and the service will take care of permissions.
7. Finally, you may activate the pipeline once you have verified all the inputs.

In case, there are errors or conflicts; we will be notified. Else, pipeline execution begins, and we see a screen similar to this:  


This takes a while, and you can monitor various components. It shows the dependencies of components as well. Once, the pipeline is successfully executed; all the components will start going to ‘FINISHED’ state. Something like this:  


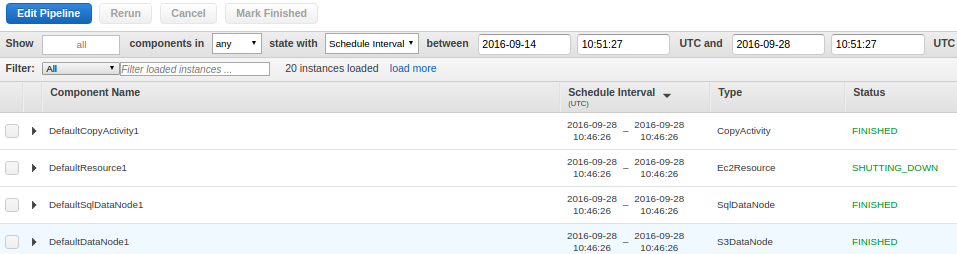
You may check the bucket for the CSV file that would have been created with complete table records. This file was created during my test:



Now, we will move to the second step that is to bring records from the CSV file to the MySQL database in EC2.

* *Moving records from CSV file in S3 bucket to MySQL table in EC2 instance*

1. Start by creating a new pipeline again.
2. Data pipeline does not have any pre-built template for our second scenario, hence we will use the option ‘Build using Architect’ under source.
3. Click on ‘Edit in Architect’ at the bottom to begin adding components to your pipeline.
4. A new split screen appears having one-half as the visualizer and the other half where you can add activities, data nodes, etc.
5. Add a CopyActivity.
6. You will see a new activity under activities. Add new data nodes both under Output and Input. Add Runs on under optional field.
7. You will see two new nodes under DataNodes. Select S3DataNode and SqlDataNode respectively as types in both data nodes. Give the path of the CSV file that got created previously. Add database, table and insert query in SqlDataNode. Example of insert query: INSERT INTO Persons (PersonID, LastName, FirstName, Address, City) VALUES (?,?,?,?,?)
8. Go to ‘Others.’ You will see a Database section. Select JdbcDatabase in type. Connection string format will be jdbc:mysql://EC2-public-dns/databasename. Give com.mysql.jdbc.Driver for JdbcDriverClass. Additionally, give the username and password that has relevant privileges to the database and table.
9. Under resources, you may define the EC2 configuration that you want the task runner to run upon. We are nearly done now.
10. Schedule the pipeline as per your requirement and go on to save it.

If there are no conflicts, then it will ask to activate. Once activated, all the components will start executing according to the pipeline definition and dependencies. Once, the pipeline is successfully executed; you may check the database and verify:

If everything goes fine, you will see that records have been successfully copied to MySQL table.

This was a simple use case with the intent to give a walk-through of the service. But, it clearly opens up the path for DevOps engineers, data scientists, analysts, etc. to create workflows for even production grade scenarios. There are so many other features that AWS Data Pipeline has to offer. One needs to identify the correct approach and measure how the service can be put to use. So, you can start playing right away and do share your experience.