TrackIt

Whitepaper

Data Pipeline Architecture Optimization & Apache Airflow Implementation

aws partner network

Advanced Consulting Partner

Data Pipelines

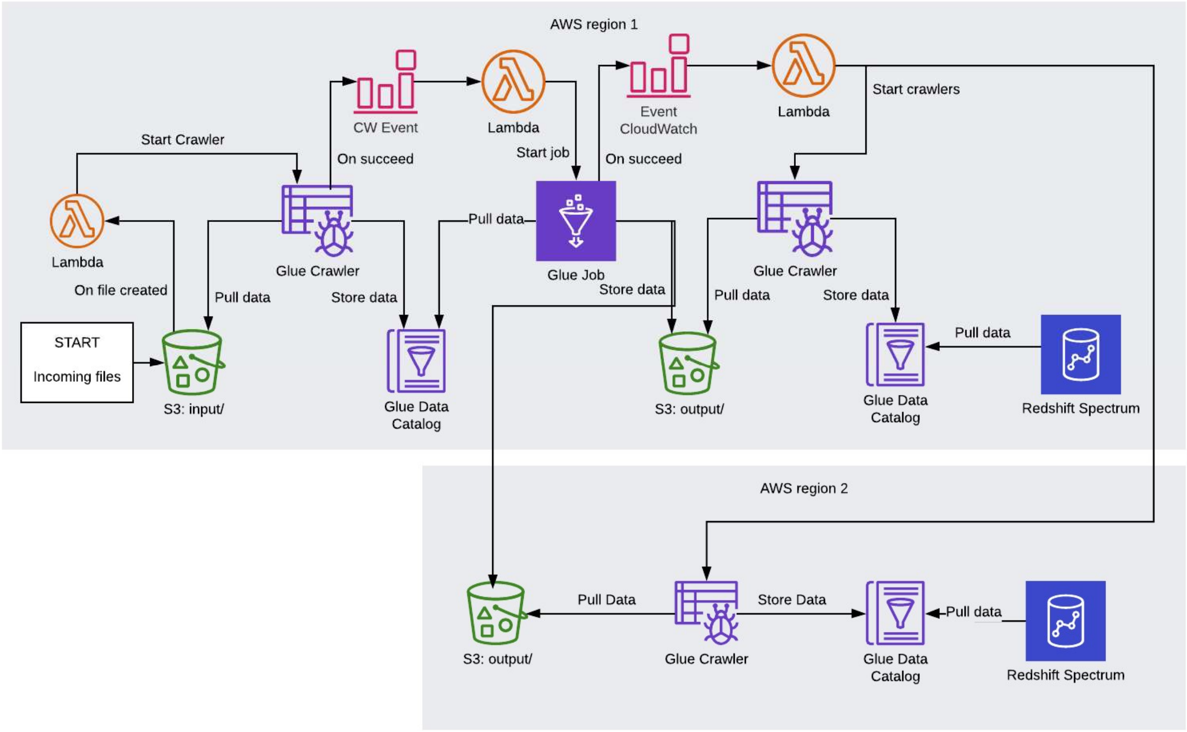
Data pipelines are essential for companies looking to leverage their data to gather reliable business insights. Pipelines allow companies to consolidate, combine, and modify data originating from various sources and make it available for analysis and visualization. However, the numerous benefits that data pipelines provide depends on a company's ability to extract and aggregate its data coming from different sources and thus on the quality of its pipeline architecture choices.

One of TrackIt's clients had implemented a big data pipeline running on AWS that needed to be optimized. The client was leveraging the big data pipeline to enable its data scientists to gain additional insights by exploiting data that originated from CSV files.

However, the company was running into certain architecture-related problems with its pipeline that needed to be fixed and sought our expertise to address these issues.

This article details the TrackIt team's approach to optimizing the architecture of the data pipeline.

Initial Pipeline



How the initial pipeline worked:

- The company's CSV files were first added to an S3 bucket
- Once the files were added to the S3 bucket, an AWS Glue job was automatically triggered to fetch the data from the CSV files and make it available for a Python Spark script, the next step in the pipeline
- A Python Spark script modified the data to make it more suitable for use in Redshift Spectrum
- The modified data was then stored in a new S3 bucket (now in the Parquet file format)
- Once files were added to the new S3 bucket, another Glue job was triggered that made the data available for use by Redshift, an SQL database
- Files from this S3 bucket were also replicated into another S3 bucket hosted on a different AWS region using AWS S3's cross-region replication feature. They needed separate S3 buckets in each region because Redshift Spectrum, which was being used in both regions separately, requires the S3 bucket to be located in the same region.
- An AWS Glue job then fetched data from the latter S3 bucket and made it available to Redshift
- Data scientists could then use a Python script to query the data on Redshift

Problems:

The pipeline implemented by the company had certain issues that were hindering its ability to make the most of its data.

Problem #1 - Inability to Individually Test Jobs

The initial pipeline did not provide the company with the ability to isolate and test individual components of the pipeline. Manually triggering one of the steps of the pipeline launched all the other events that followed it.

Problem #2: Too many steps in the pipeline

The initial pipeline included quite a few additional steps - such as the Lambda functions and CloudWatch events before and after the Glue job - that made the pipeline harder to test and manage. These extra steps could have been avoided with different architectural choices.

Problem #3: Cross-region data replication

There was also an issue arising due to the cross-region data replication feature on S3. The data replication between AWS region 1 and AWS region 2 was not instantaneous and took a few minutes. However, the completion of the AWS Glue job in region 1 was immediately triggering the Glue job in region 2 before the data had finished replicating between the S3 buckets.

Problem #4 - No error notifications

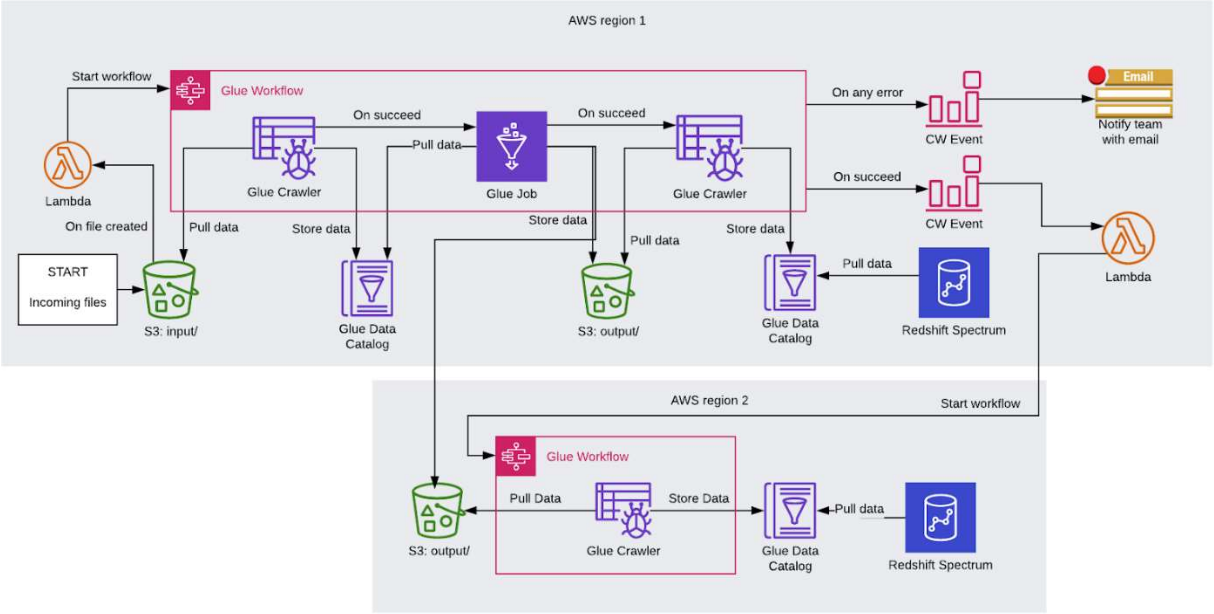
The initial pipeline provided the company with no error notifications. The company often discovered the occurrence of errors weeks after an event, and then only because the data scientist realized that they were missing data. When an error did occur, the next job would simply not be launched. The pipeline did not include an error notification component that would allow the company to immediately become aware of errors happening within the pipeline. The company's engineers had to go onto the console and investigate the history of executions to try to identify and pinpoint errors in the pipeline.

Optimized Pipeline

The following modifications were first proposed by the TrackIt team to the client.

The first modification to the pipeline proposed by the TrackIt team was to use Glue Workflow, a feature of AWS Glue to create a workflow that automatically launches the AWS Glue jobs in sequence. The Glue Workflow would also allow the company to launch and test jobs individually without triggering the whole workflow.

The implementation of the Glue Workflow would also enable the company to simplify the pipeline by getting rid of extraneous Lambda functions and CloudWatch events that had been implemented in the initial pipeline. Instead of having multiple Lambda functions, the new pipeline would have just one Lambda function that triggers the Glue Workflow when files are uploaded into the S3 bucket.



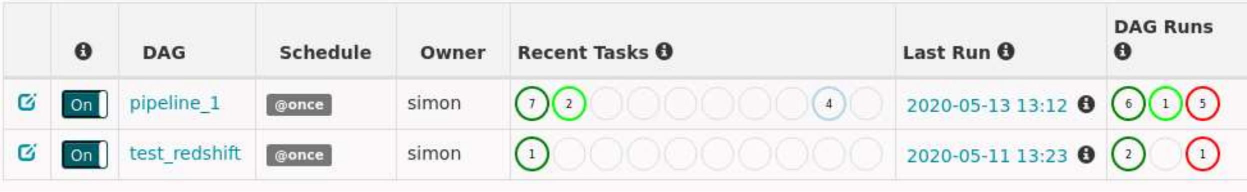
The second modification proposed by the TrackIt team was the addition of an error notification component using Amazon CloudWatch. CloudWatch events would be triggered immediately when an error occurred in the Glue Workflow and would then send either an HTTP request or an email to the team, or could trigger a Lambda function that would execute additional tasks if there was an error.

The third modification to the pipeline proposed by the TrackIt team was to eliminate the use of S3 cross-region replication. Instead, the files are directly added to both S3 buckets (each located in a different region) so that when the Glue job is triggered in region 2, all the files are already up to date in both S3 buckets.

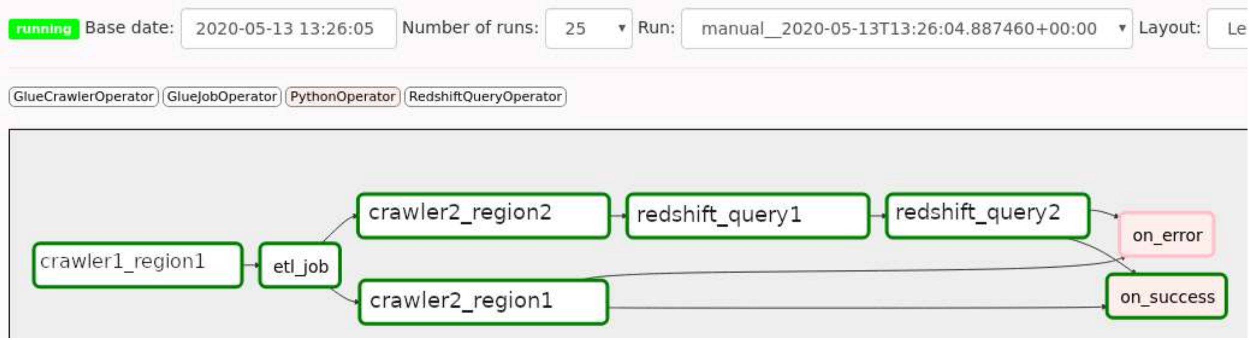
The client was quite pleased with this proposition and wanted to incorporate these new changes to the pipeline using Apache Airflow, a tool used to create and manage complex workflows.

Apache Airflow Implementation

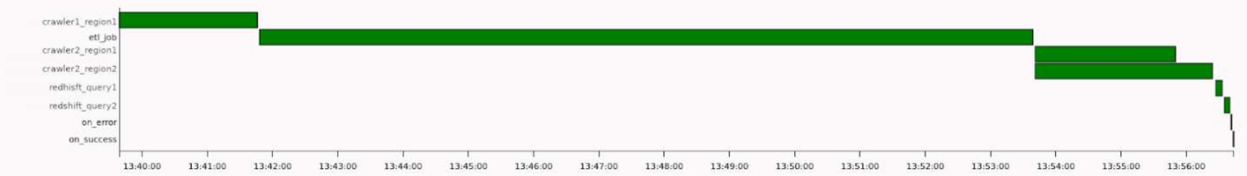
The TrackIt team assisted the client in incorporating the suggested modifications to the pipeline and implementing it on Apache Airflow. The different parts of the pipeline were coded in Python as modules that the client could reuse in the future to build similar pipelines or to further modify the existing one.



	ℹ	DAG	Schedule	Owner	Recent Tasks ℹ	Last Run ℹ	DAG Runs ℹ
	On	pipeline_1	@once	simon	7 2 4	2020-05-13 13:12 ℹ	6 1 5
	On	test_redshift	@once	simon	1	2020-05-11 13:23 ℹ	2 1



running Base date: 2020-05-13 13:26:05 Number of runs: 25 Run: manual__2020-05-13T13:26:04.887460+00:00 Layout: Le



How the final Apache Airflow pipeline works:

- The company's CSV files are first added to an S3 bucket
- The AWS Glue crawler fetches the data from S3
- A Python Spark script is executed that modifies the data and makes it more suitable for use in Redshift Spectrum
- The modified data is then stored in a new S3 bucket (now in the Parquet file format)
- Then in one region, a Glue crawler fetches data
- In the other region, the Glue crawler fetches data and a Redshift script is used to modify data and then update changes
- Data scientists can use a Python script to query the data on Redshift
- If any error occurs within the pipeline, a CloudWatch event is immediately triggered and sends an email to notify the team

About TrackIt

TrackIt is an Amazon Web Services Advanced Consulting Partner specializing in cloud management, consulting, and software development solutions based in Venice, CA.

TrackIt specializes in Modern Software Development, DevOps, Infrastructure-As-Code, Serverless, CI/CD, and Containerization with specialized expertise in Media & Entertainment workflows, High-Performance Computing environments, and data storage.

TrackIt's forté is cutting-edge software design with deep expertise in containerization, serverless architectures, and innovative pipeline development. The TrackIt team can help you architect, design, build and deploy a customized solution tailored to your exact requirements.

In addition to providing cloud management, consulting, and modern software development services, TrackIt also provides an open-source AWS cost management tool that allows users to optimize their costs and resources on AWS.