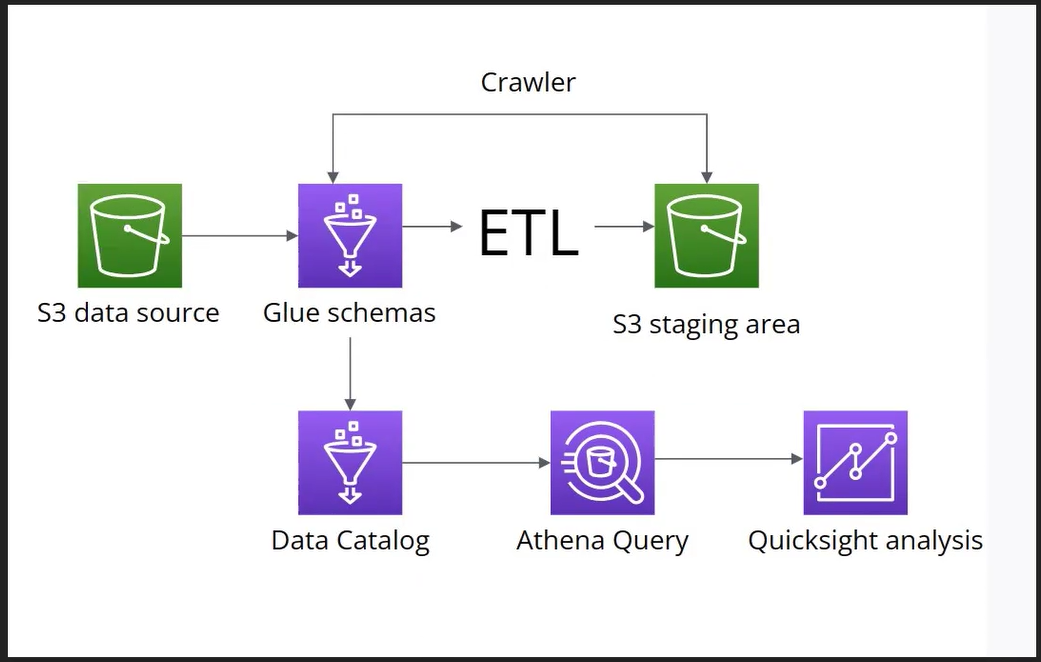
Pokémon Meta Analysis End to End Automated Pipeline



Data Ingestion:

The data is originally uploaded as .csv files to Amazon S3 with each table as a separate file. For the demo we have a total of 13 tables each with 1000+ entries except the few where it was not applicable.

A screenshot of a computer

Description automatically generated

Data Crawler:

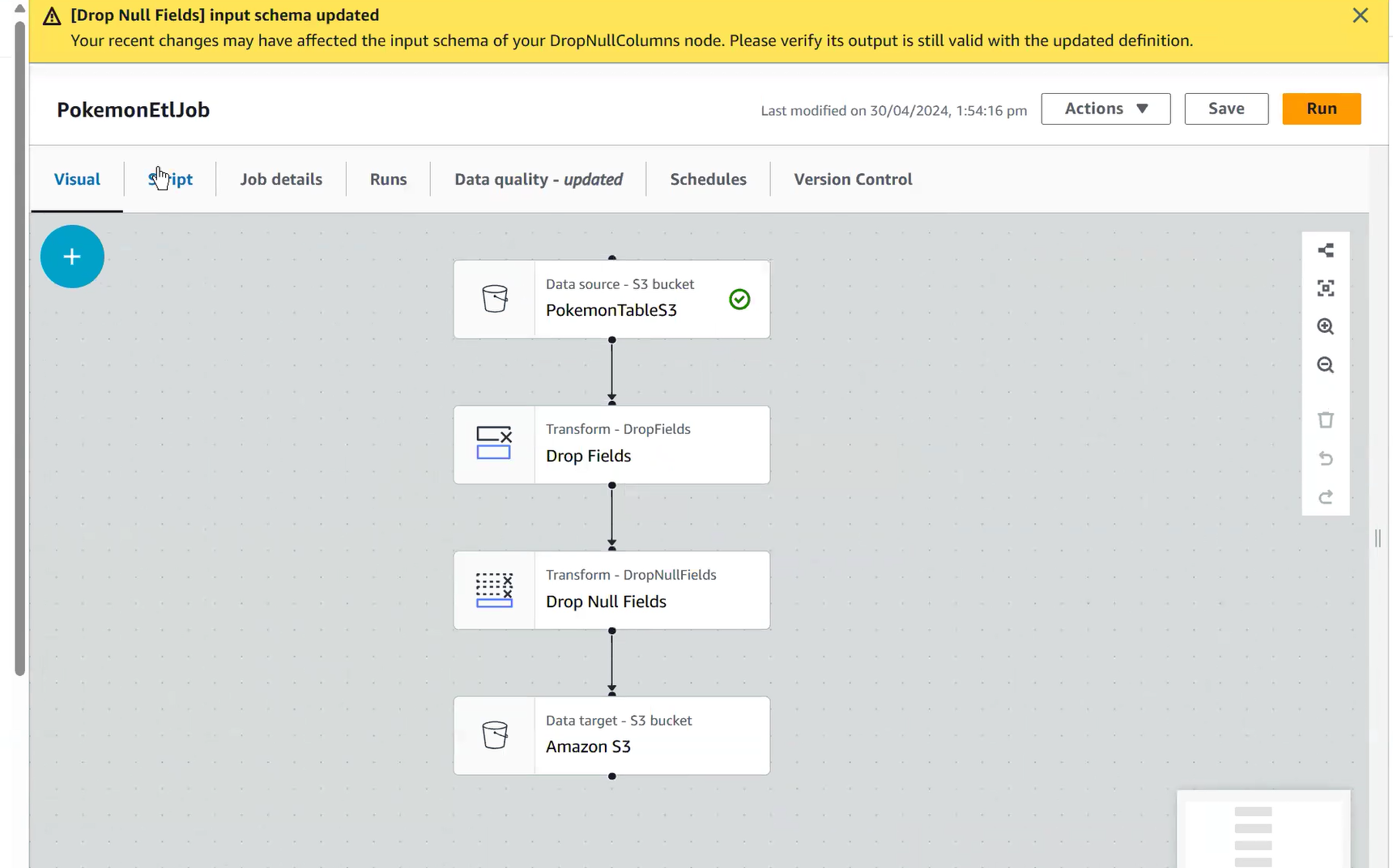
The data is then crawled in order for the schema to be inferred. These crawlers run at a daily basis and so this pipeline caters to any schema changes in the data as well.

A screenshot of a computer

Description automatically generated

ETL Glue Jobs:

Jobs are configured in AWS Glue to perform ETL on a daily basis as well. The data is retrieved from the appropriate bucket, cleaned, transformed and loaded into a new S3 bucket specifically for that dimension



Dimensional Crawlers:

3 new crawlers are then used on a daily basis automatically to infer the schema of these newly created dimension tables stored in the respective S3 buckets. This will allow us to store the dimensions in glue itself in the next step

A screenshot of a computer

Description automatically generated

Transport Job:

The cleaned data in the new S3 dimensional buckets is then transported to AWS Glue Data Catalog in order for the dimensional queries to be run in order to create a fact table.

A screenshot of a computer

Description automatically generated

Dimensional Queries:

The dimensions in the data catalog are then queries using AWS Athena and the result is stored as a custom SQL query. This is the fact table snapshot and is used to then analyse the data as a dashboard.

A screenshot of a computer

Description automatically generated

Analysis and Conclusion:

The analysis represents the Pokémon that have been performing well in terms of win-ratio as individuals as well as according to the win-ratios of the teams that they were used in. It also gives a glimpse of what the win-ratios look like in each of the skill tier brackets.

The Business use case is of course simple as of right now and is only one because of the data limitations but gives a glimpse of what the pipeline is capable of giving proper real-time data.

A screenshot of a computer

Description automatically generated