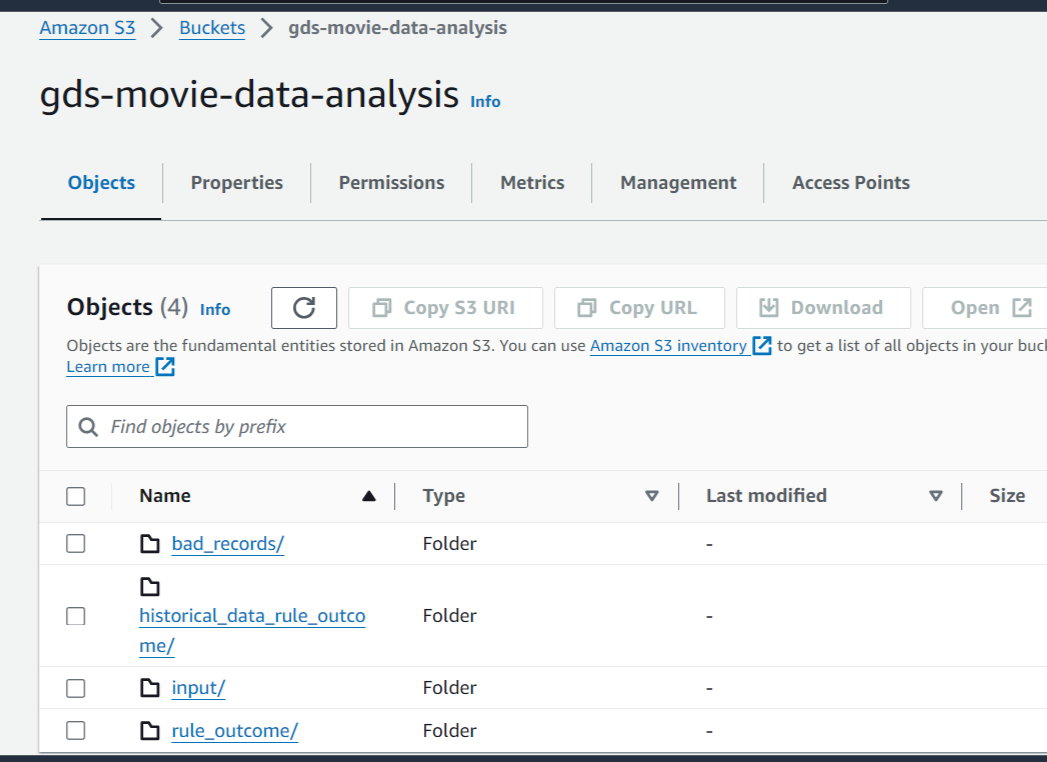
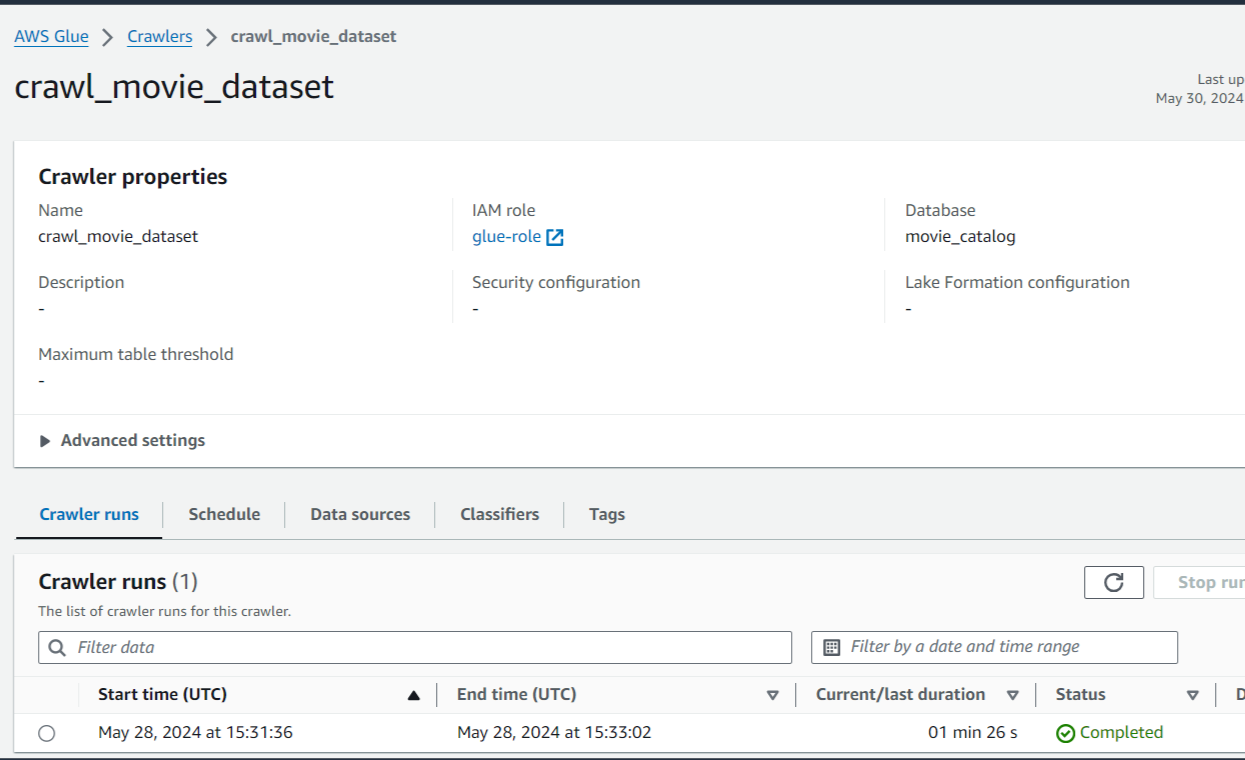
Quality Movie Data Analysis

1. create s3 bucket with bad records, historical\_data\_rule\_outcome, input, rule\_outcome

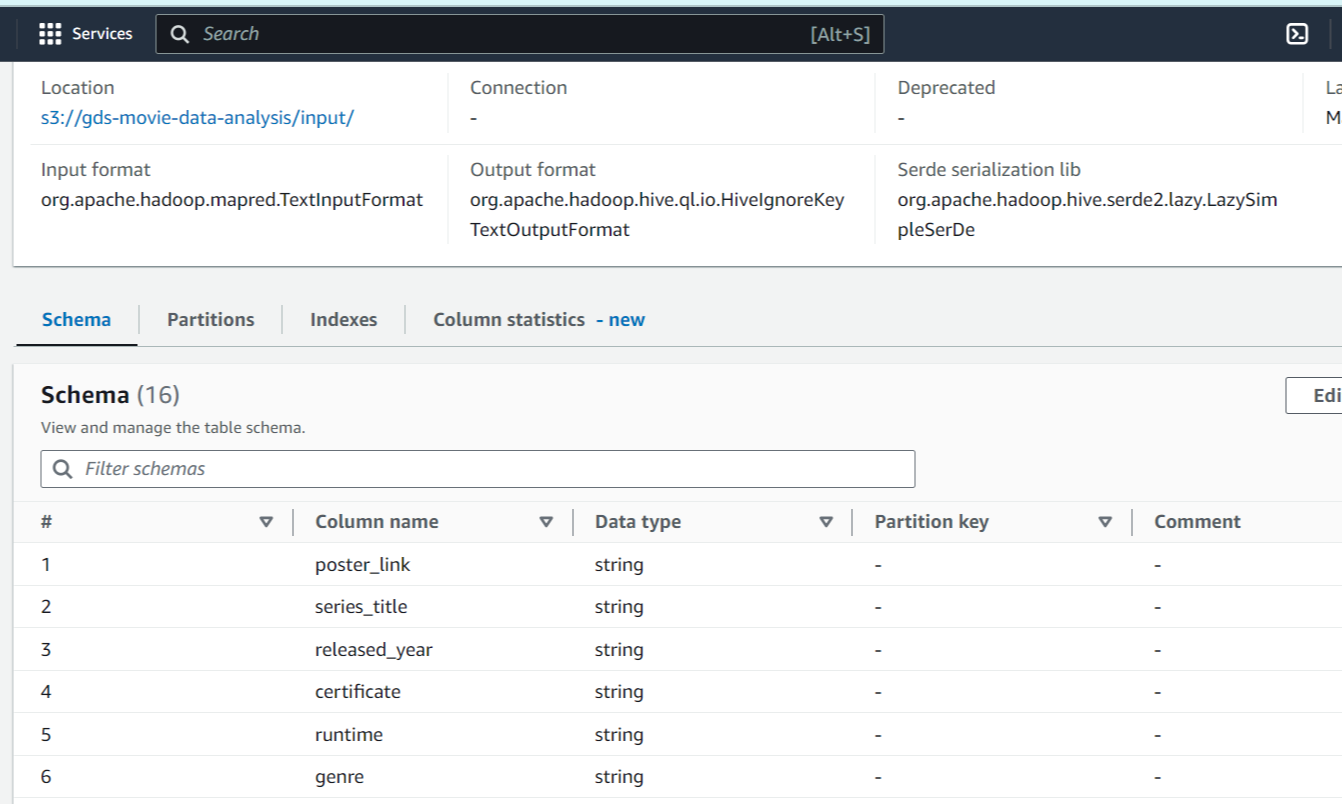


1. created crawler on s3 input source where we also set a data quality rule to monitor the quality of your data assets

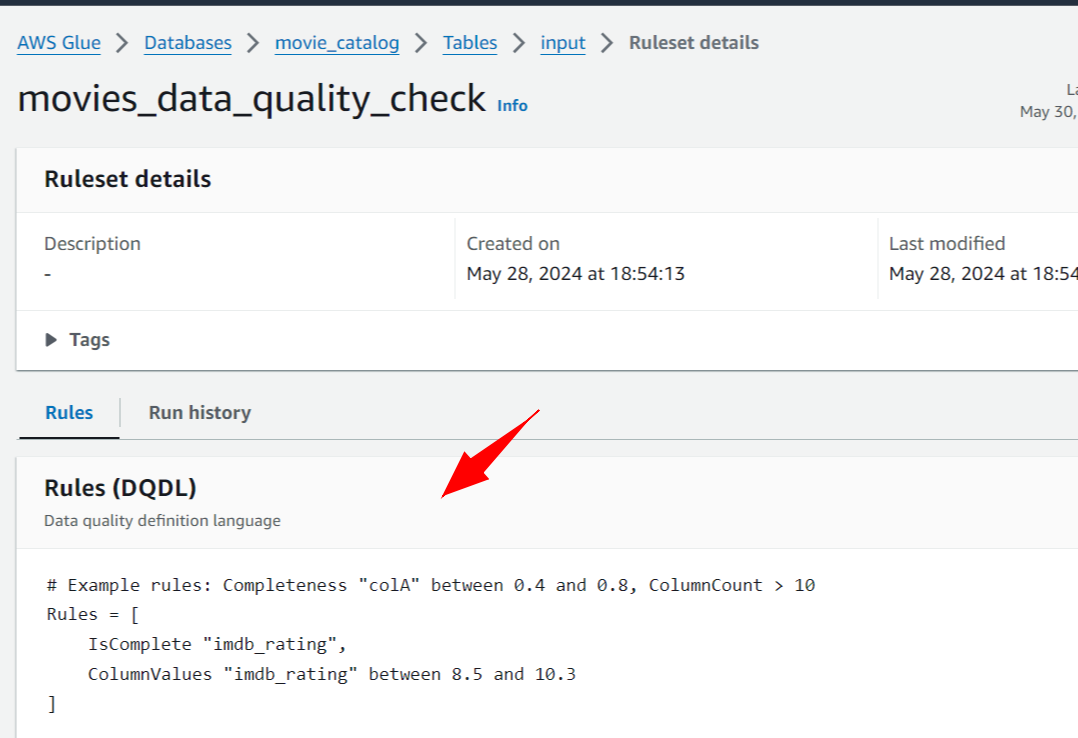
crawler:



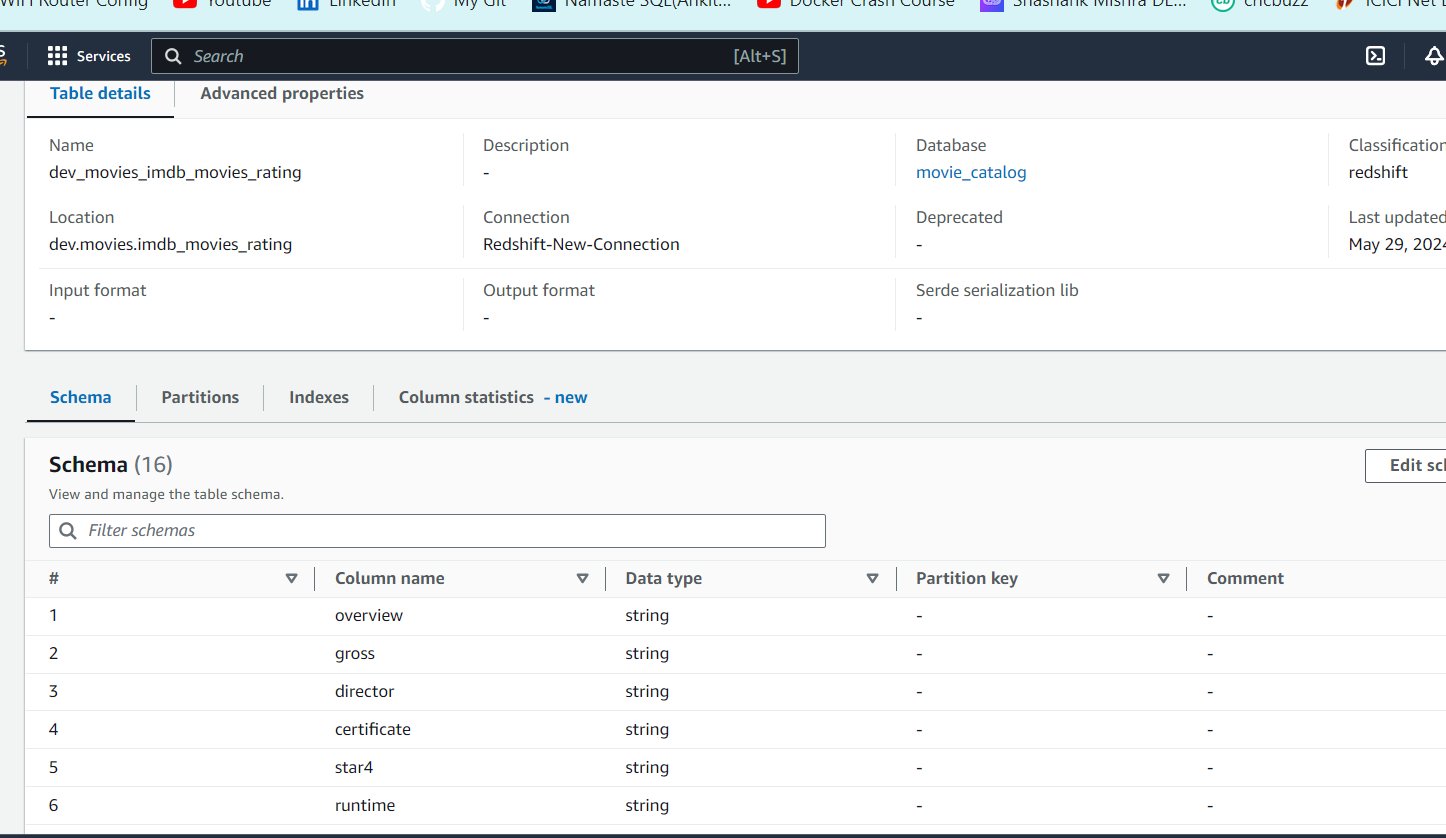
input source catalog table:



1. set a data quality rule to check their must be an imdb rating(not blank). Another rule is rating should consist between 8.5 and 10.3. On running we get passed and failed rule outcomes which was getting stored in s3 bucket another folder 'historical\_data\_rule\_outcome'

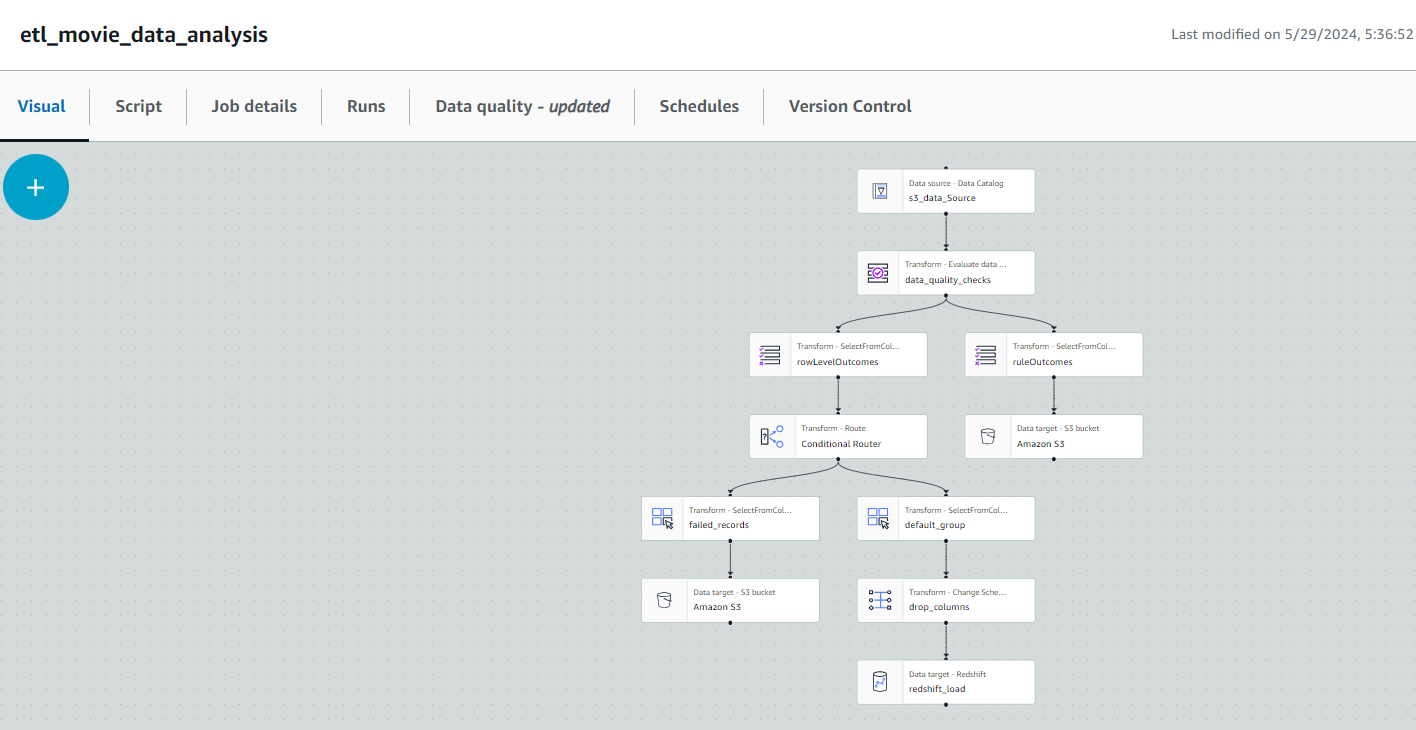


1. Created a jdbc connection for connecting crawler to redshift target table. That crawler will be crawling the target redshift table where output results will be loaded

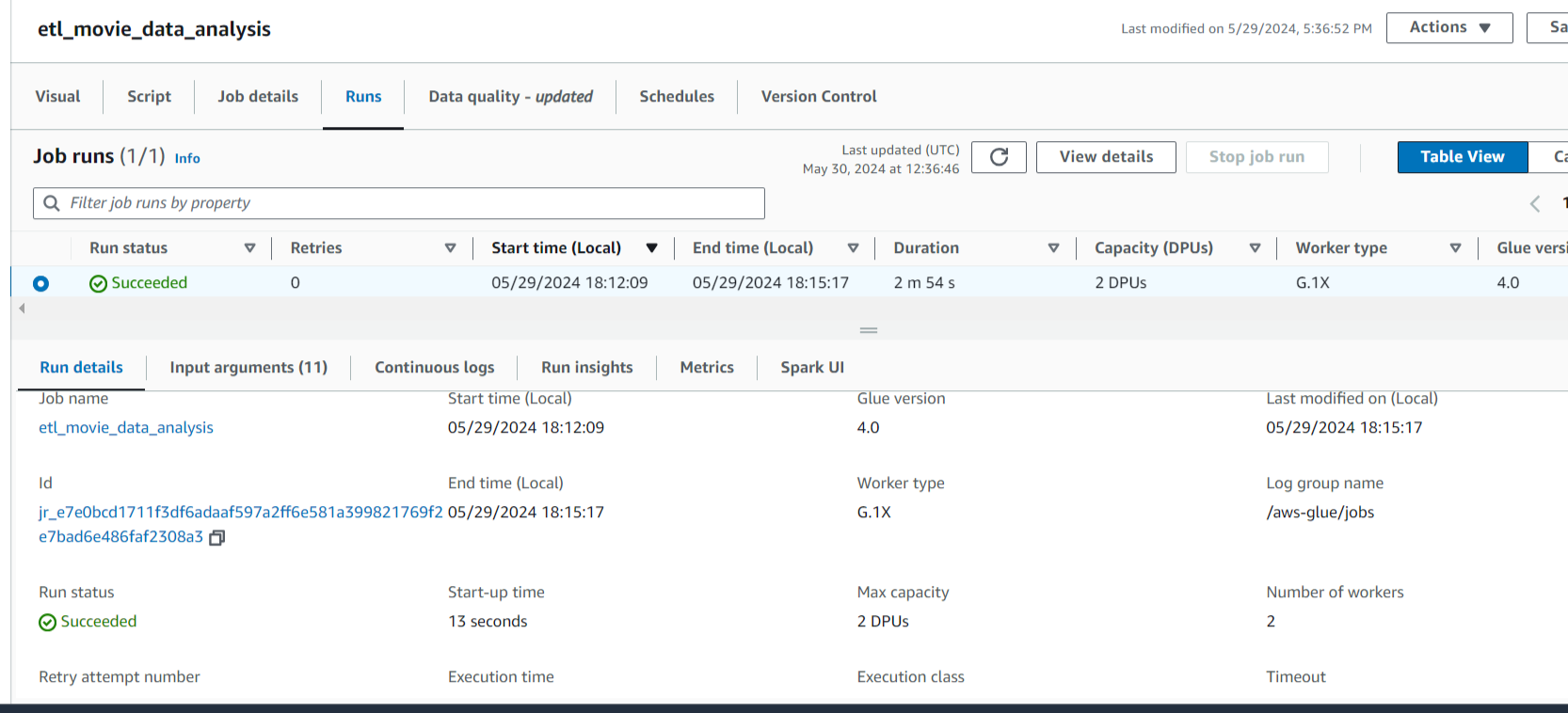


1. build an etl job where s3 input source being read from data catalog. Then we add our first transformation to include our data quality rule set where I also enabled 'add new columns on data quality errors'. We further applied two more transformations i.e., rowLevelOutcomes and ruleOutcomes

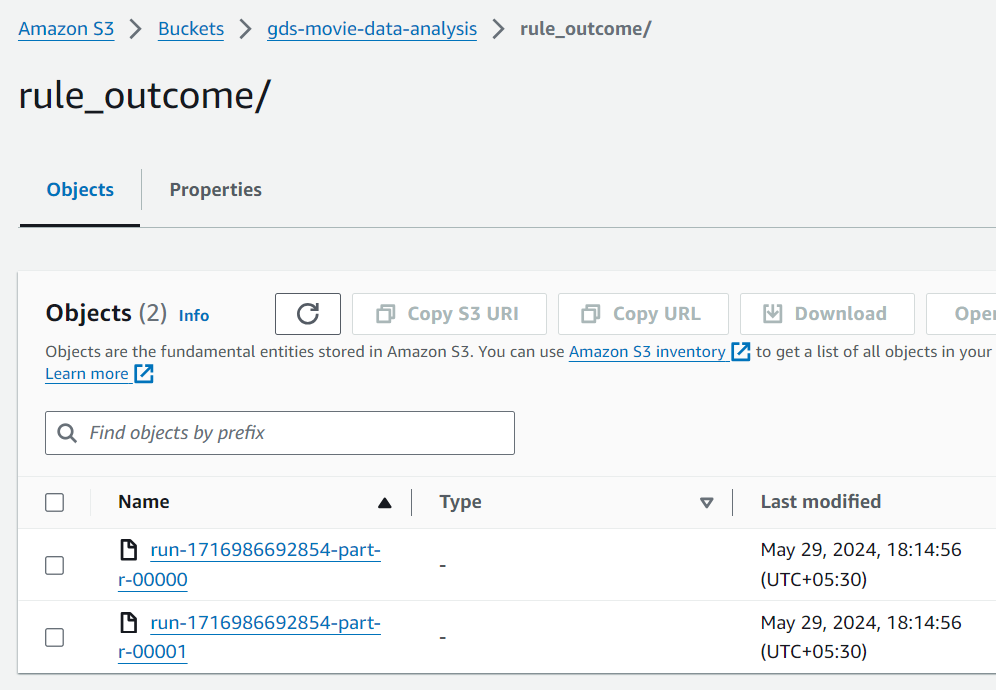
etl job visual representation:



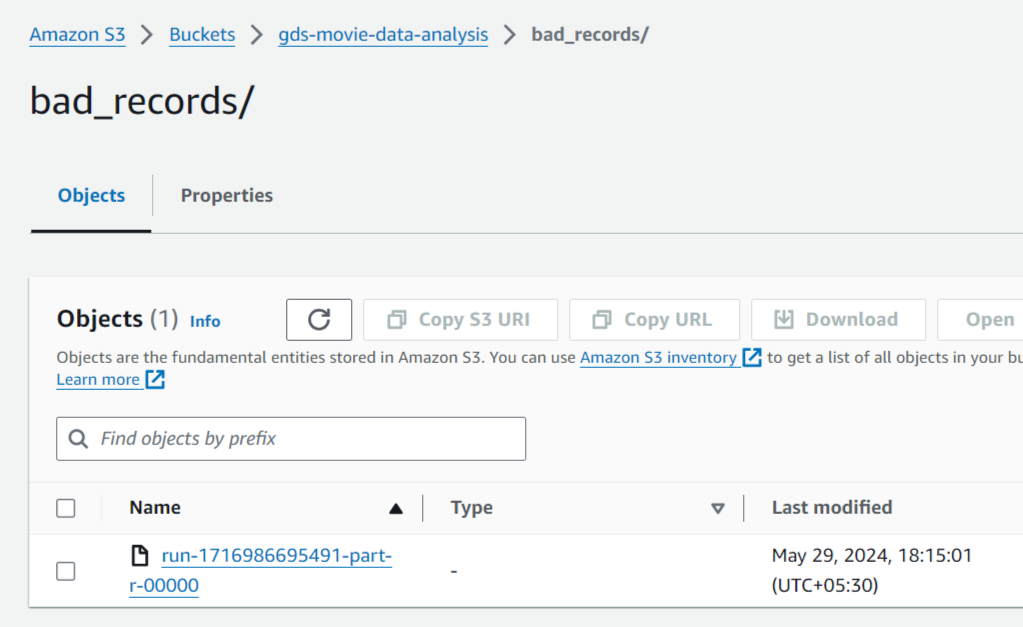
success etl job runs:



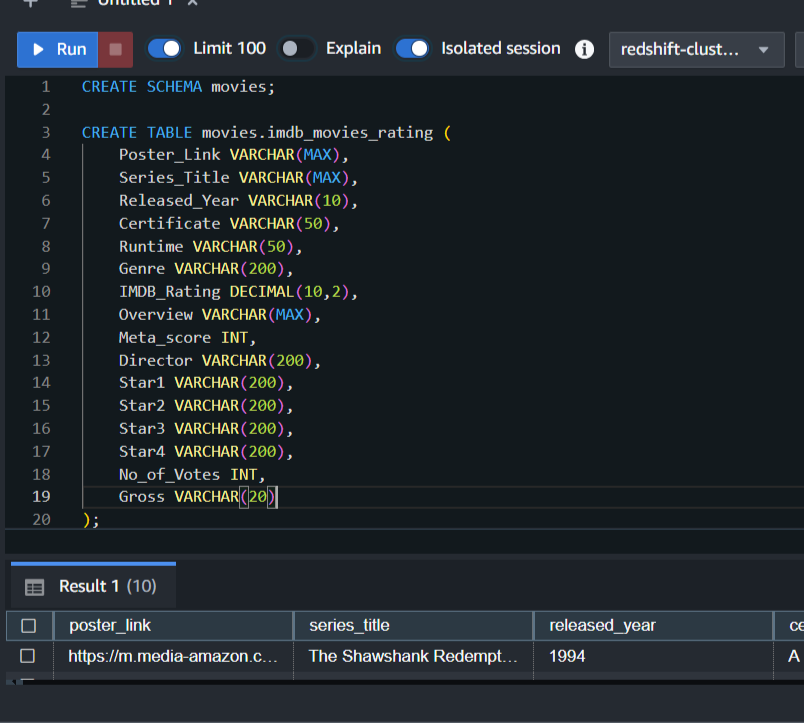
1. ruleOutcomes will be having rule wise outcome with pass or fail reason. And we were storing those in json format at s3 bucket another folder 'rule\_outcome'.



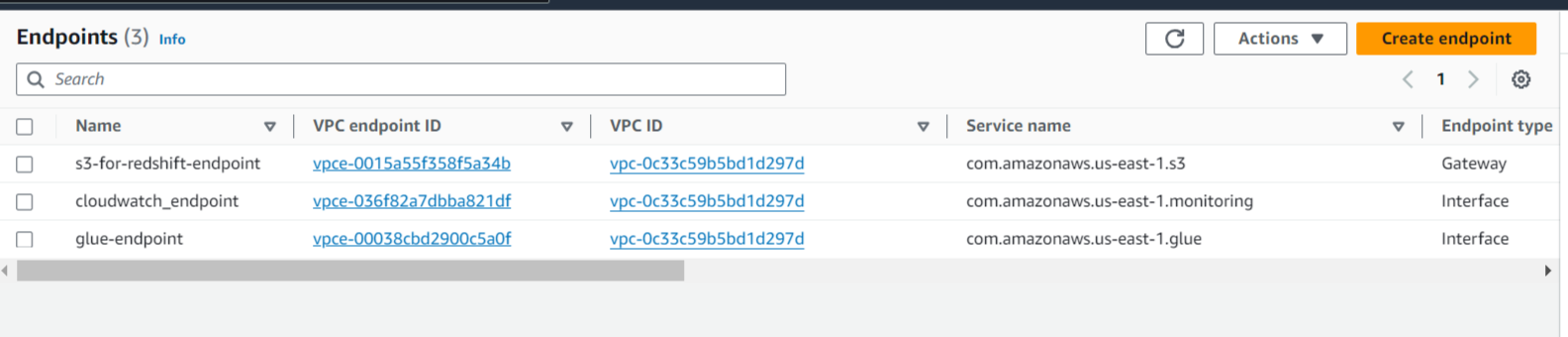
1. On the other hand, rowLevelOutcomes will be having row wise outcome with pass or fail reason. Following rowLevelOutcomes, we further next added a conditional router where we checked with filter condition whether 'DataQualityEvaluationResult' matches 'Passed' value or not.
2. once we have groups on failed and passed records group. For failed records, we were storing data in json format at s3 bucket 'bad\_records' folder.



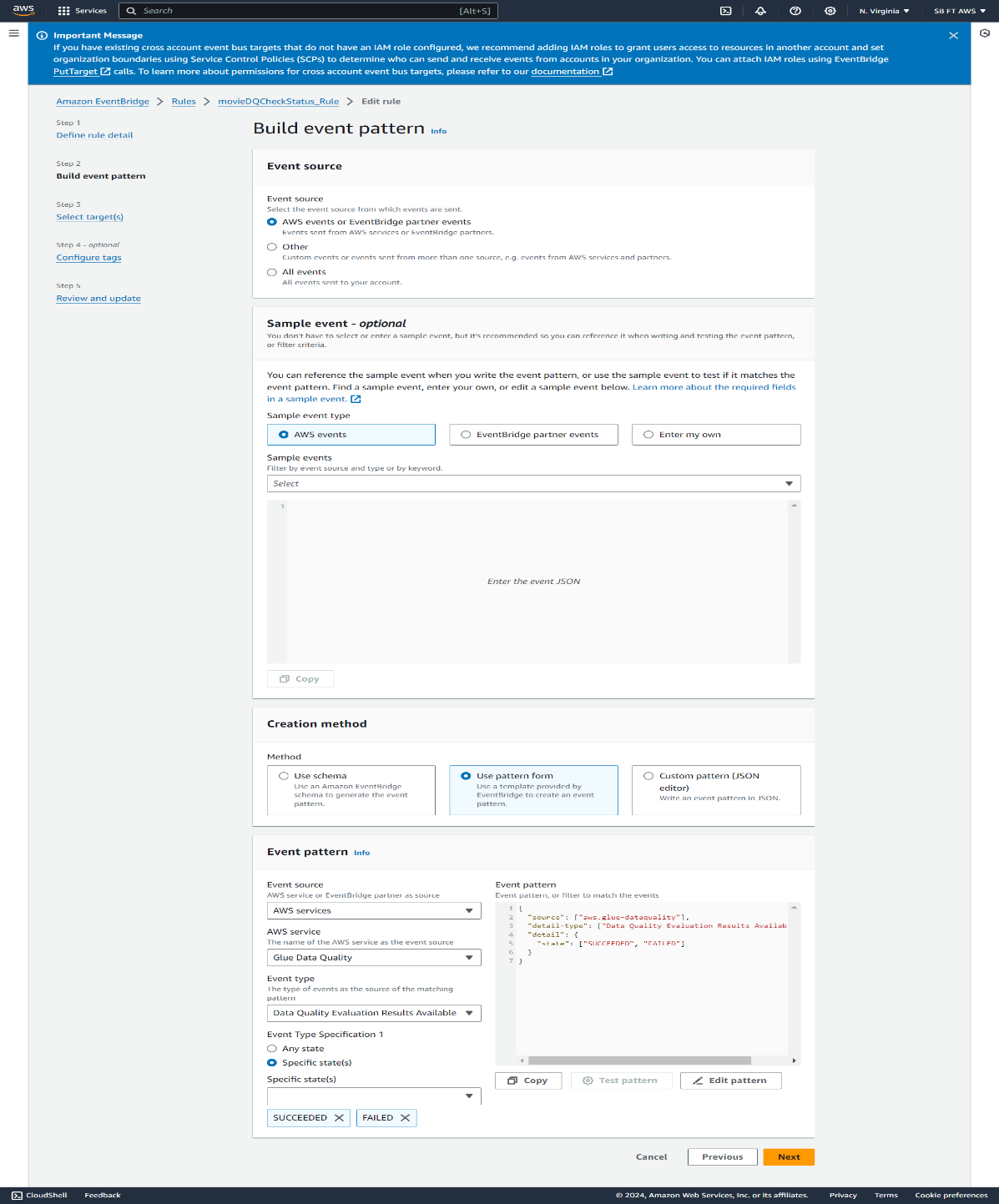
1. For passed records, first we were applying 'change schema' transform where we matched both the source key and target key(redshift target table) column's data types and also dropping the 'data quality' additional columns.
2. As we already created a target redshift table 'imdb\_movies\_rating', so next we uploaded passed records to the target redshift table where assigned 'IAM' access role with the redshift cluster

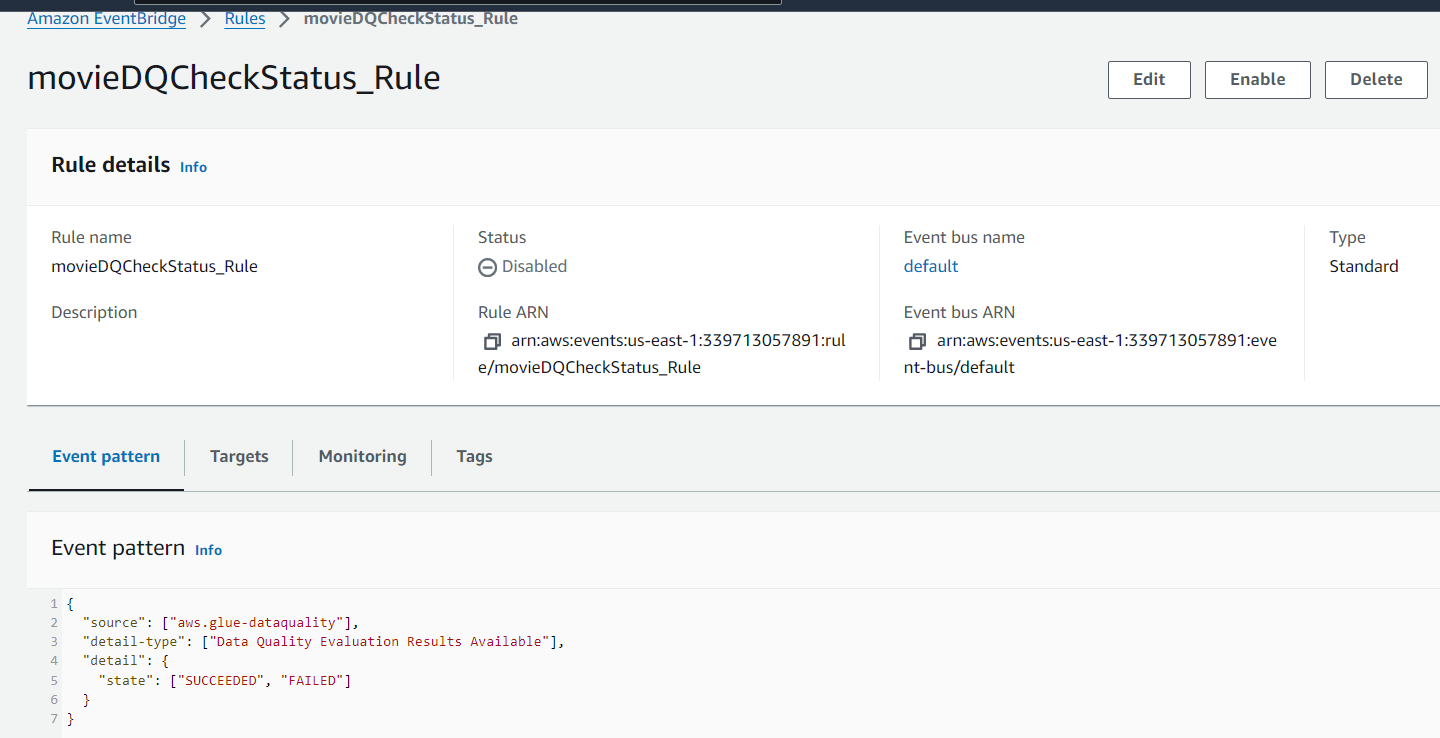


1. Also, added vpc endpoints for glue and cloudwatch under same vpc id(allows for private, secure communication between your Glue jobs and CloudWatch services with no expose to public internet) included in redshift cluster.



1. We next made an eventbridge rule where a pattern is defined to check receiving event's data quality status using 'AWS Glue Data Quality' service with event type 'Data Quality Evaluation Results Available'.





1. We then invoke a target SNS topic service. We were sending success/fail notification using this service.

