Case Study: NetSuite Data Pipeline

Background: The CFO and Accounting Team needed transactional data from Stripe loaded into NetSuite.

Challenges: NetSuite has a latency of .75s/Journal Entry. After close, data can no longer be changed. Stripe can change transactional data anytime within a 45 day window.

Solution: Build a data pipeline using AWS Lambda, Batch, S3, and Snowflake for staging into NetSuite and Tableau.

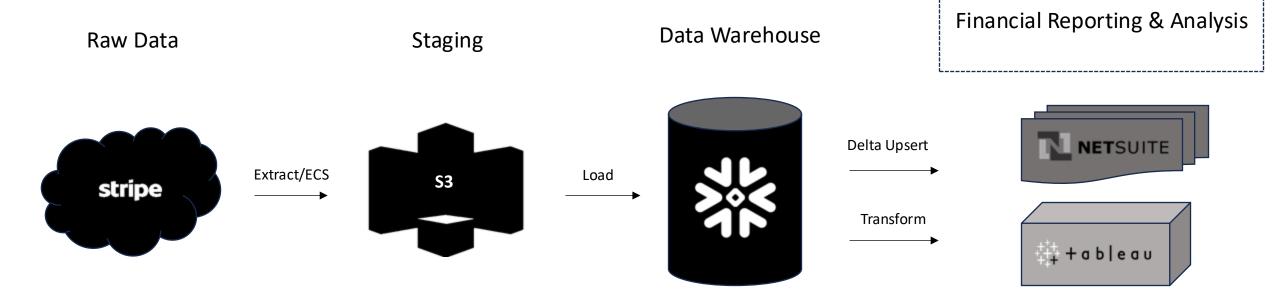
Coding Stack: Python, SQL, and Spark.

Tradeoffs: Batch vs. streaming – Stripe transactional data changes intermittently, so batch was more appropriate. Boomi vs. in-house Python SDK – the accounting team needed an ETL tool that they could monitor, so we built both solutions.

Security and dev ops: ACL for each stage of the process with one VPC for the AWS pipeline.

End result: Successful first close.

Architectural Diagram



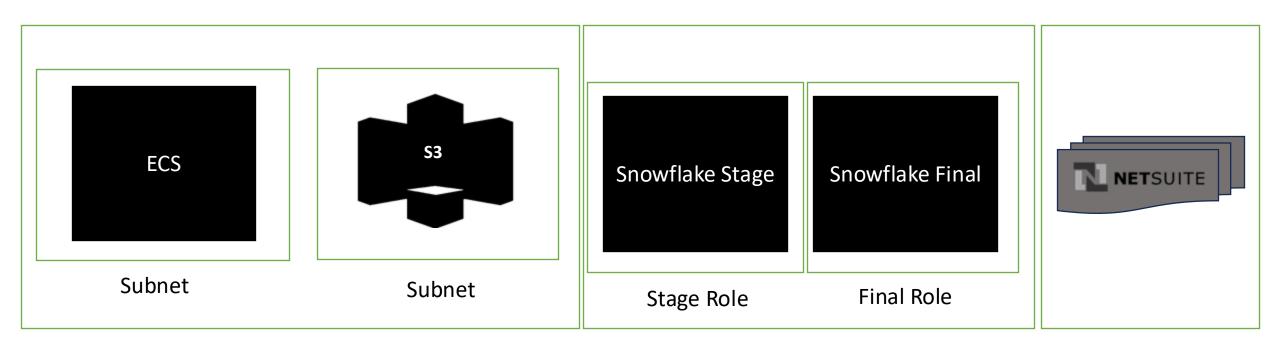
Stripe API, Python and AWS Lambda to extract transactional data (payments, refunds, disputes) Stripe data is loaded into Parquet files and staged in S3 before loading into Snowflake.

Snowflake for landing data via SnowPipe and transforming for ingestion into Tableau and NetSuite

Data from Snowflake is Upserted into Journal Entries for Accounting. Data is also staged in relevant data marts for Tableau Analytics.

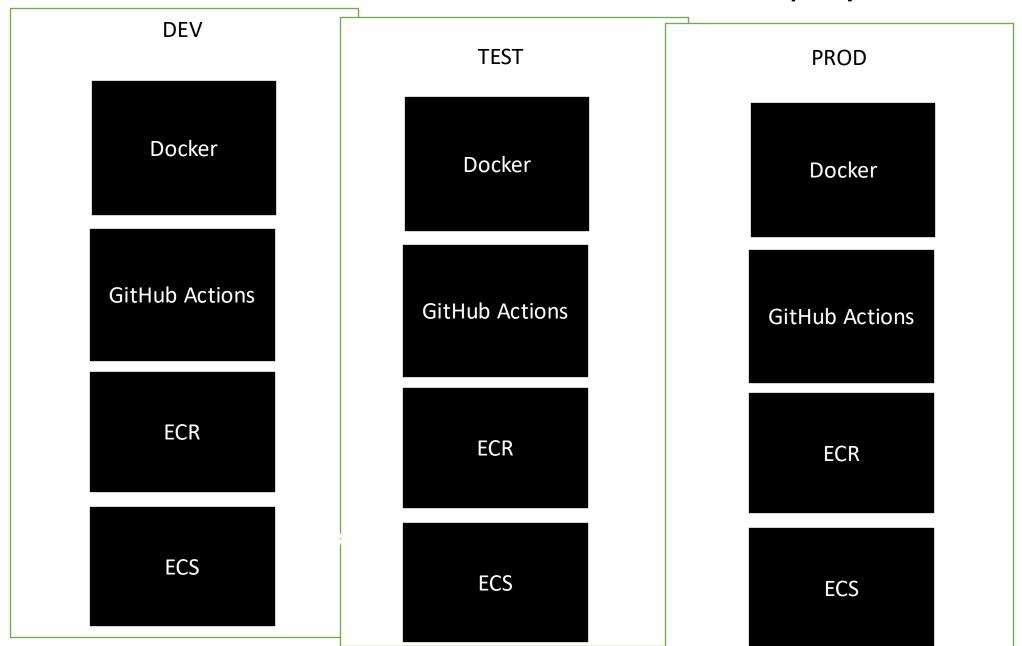
Security

VPC

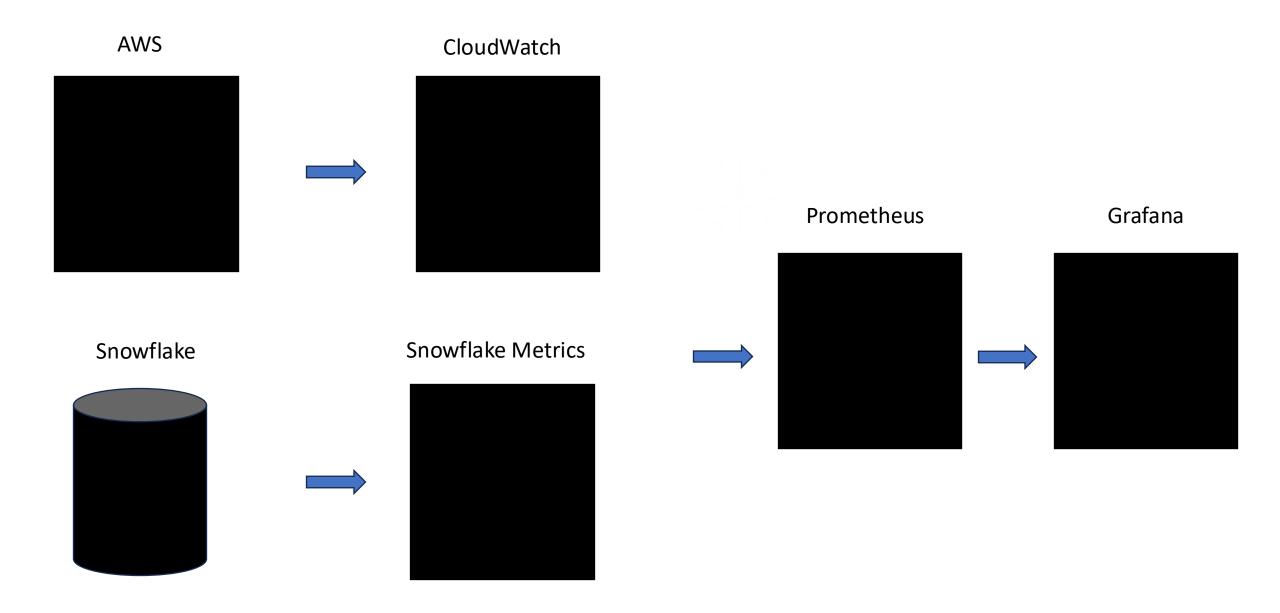


ETL Role Table Level Permissions

Deployment CI/CD



Monitoring



Backup/Logging

