From Chaos to Clarity: The Revamp Style Data Engineering Journey

Final Project Report – Naya Data Engineering Course

# Contributors

**- Harry**

**- Avital**

# Table of Contents

[1. Executive Summary 3](#_Toc207709426)

[2. Core Philosophy & Approach 3](#_Toc207709427)

[3. Architecture Overview 3](#_Toc207709428)

[4. Phase 1 – Building the Foundation with Terraform (IaC) 4](#_Toc207709429)

[5. Phase 2 – Ingestion (Bronze Layer) 5](#_Toc207709430)

[6. Phase 3 – Transformation (Silver Layer) 7](#_Toc207709431)

[7. Phase 4 – Warehousing & Analytics (Gold Layer) 8](#_Toc207709432)

[8. Phase 5 – Orchestration with Step Functions 8](#_Toc207709433)

[9. Phase 6 – Visualization with QuickSight 11](#_Toc207709434)

[10. Security & IAM 11](#_Toc207709435)

[11. Operational Guide (Runbook) 11](#_Toc207709436)

[12. Challenges & Solutions 12](#_Toc207709437)

[13. Conclusion 12](#_Toc207709438)

[14. Appendix – Running the Project 12](#_Toc207709439)

# 1. Executive Summary

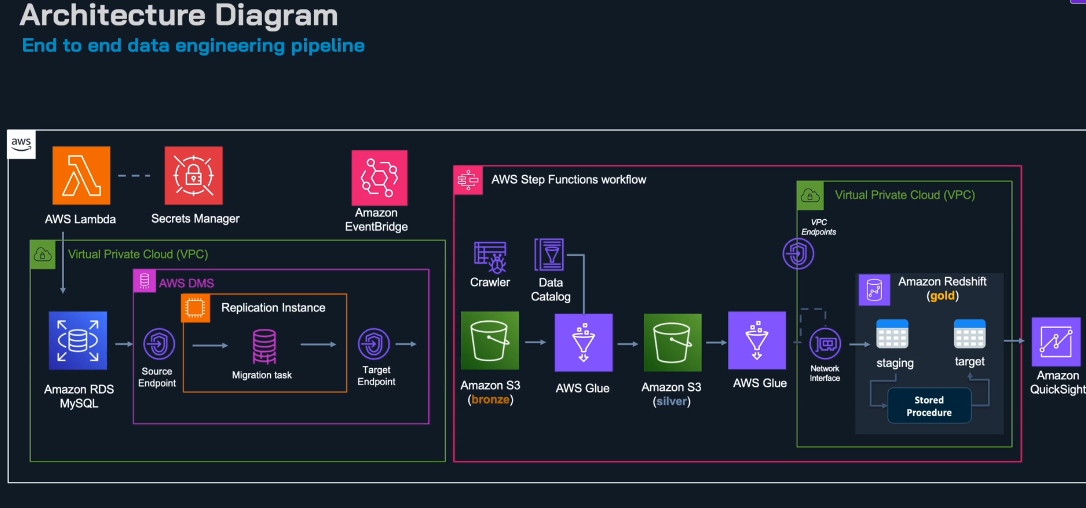
Revamp Style, a growing e-commerce startup, struggled with fragmented, siloed data locked in a monolithic MySQL database. Analysts exported CSVs into Excel, but reports were slow, stale, and error-prone. The business demanded real-time insights, which the old system could not provide. Our mission was to build a modern, automated AWS data pipeline that ingests, cleans, transforms, and delivers business-ready data daily. The result is a scalable, serverless, cost-effective platform orchestrated with Terraform, AWS Glue, Step Functions, Redshift, and QuickSight.

# 2. Core Philosophy & Approach

Every engineering decision was guided by three principles: scalability, automation, and cost-effectiveness. We avoided manual processes and built a fully serverless architecture leveraging AWS managed services. The Medallion Architecture (Bronze, Silver, Gold) structured our data into raw, cleaned, and analytics-ready layers, ensuring clarity, quality, and performance.

# 3. Architecture Overview

Our architecture is built around the Medallion pattern:  
- Bronze: Raw, unfiltered data from RDS via DMS into S3.  
- Silver: Cleaned, transformed, and validated data using AWS Glue.  
- Gold: Aggregated, business-ready data optimized for analytics in Redshift.  
  
Step Functions orchestrate ETL pipelines, and QuickSight dashboards deliver insights.



# 4. Phase 1 – Building the Foundation with Terraform (IaC)

We provisioned all AWS resources with Terraform, ensuring reproducibility, modularity, and version control. The setup.sh script created a modular project structure for RDS, S3, IAM, Glue, DMS, Redshift, and Step Functions. The deploy.sh script allowed selective or full-stack deployments. This IaC approach avoids console errors and enables rapid disaster recovery with a single command.

*Excerpt from setup.sh:*

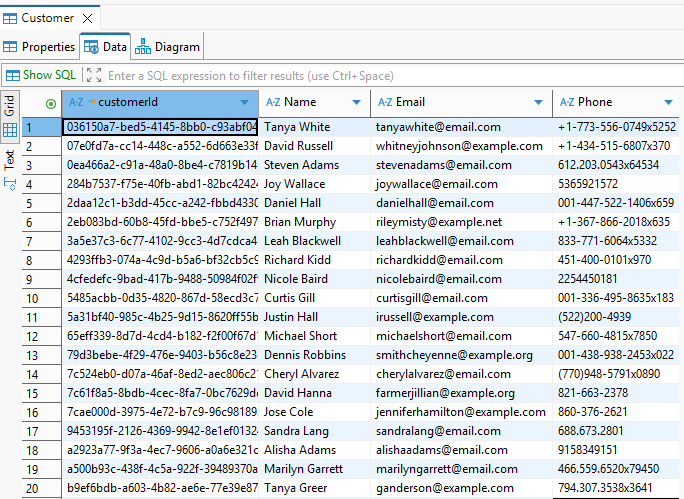
MODULES=(rds s3 iam dms secrets lambda glue redshift stepfunctions)  
for module in "${MODULES[@]}"; do  
 mkdir -p "modules/$module"  
 touch "modules/$module/main.tf"  
done

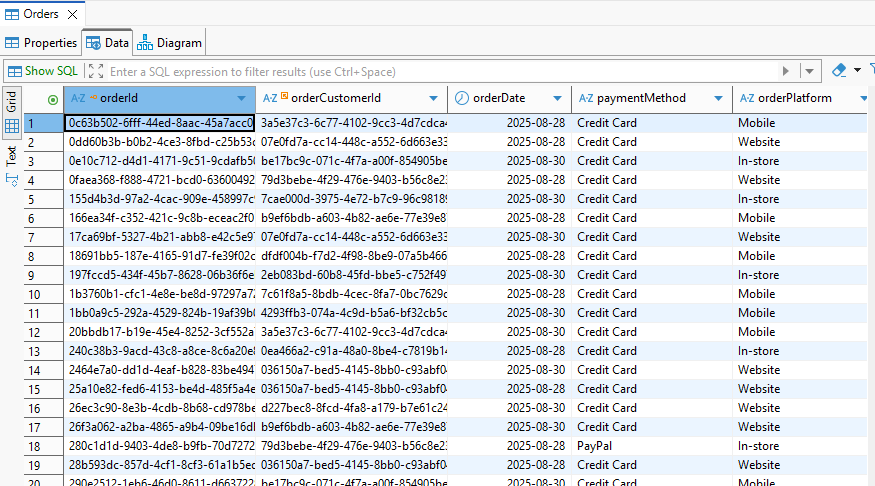
Excerpt from deploy.sh:

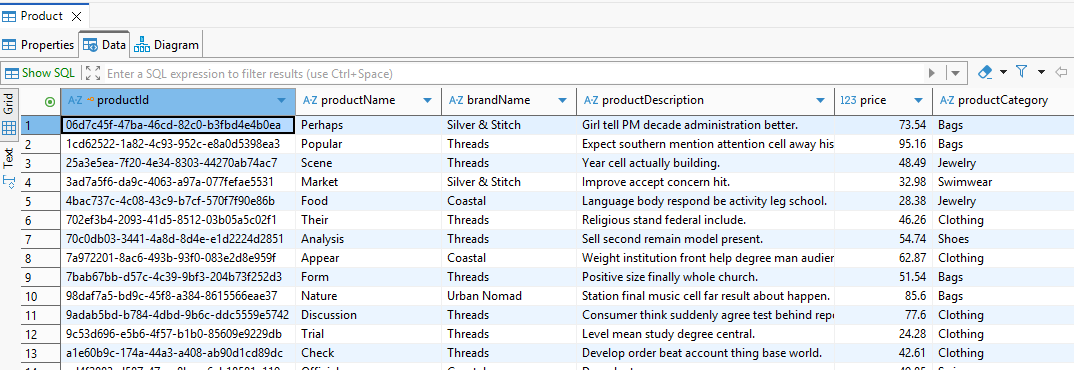
case $PHASE in  
 "vpc") terraform apply -target=module.vpc ;;  
 "rds") terraform apply -target=module.rds ;;  
 "s3") terraform apply -target=module.s3 ;;  
 "iam") terraform apply -target=module.iam ;;  
 "all") terraform apply ;;  
esac

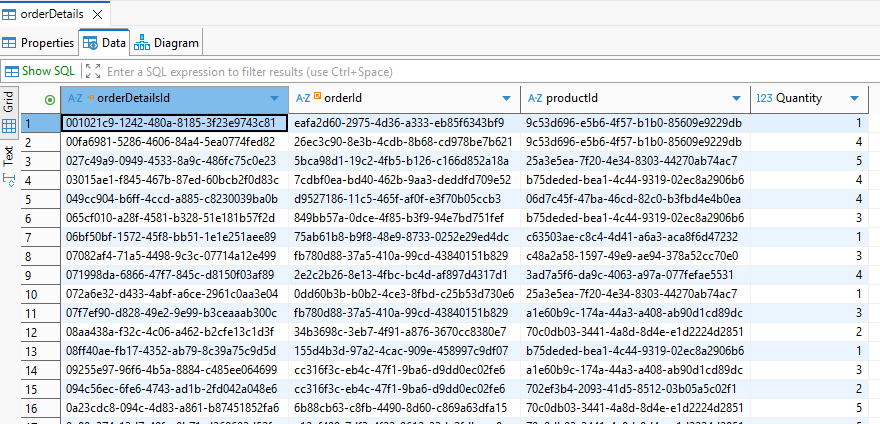
# 5. Phase 2 – Ingestion (Bronze Layer)

Amazon RDS MySQL acted as the source database. Binary logging was enabled for Change Data Capture (CDC). AWS DMS replicated changes continuously into Amazon S3 as raw Parquet files. A Lambda generator populated RDS with synthetic data (Customers, Orders, Products, OrderDetails) using Faker. Credentials were stored securely in Secrets Manager, never hardcoded in Lambda.









# 6. Phase 3 – Transformation (Silver Layer)

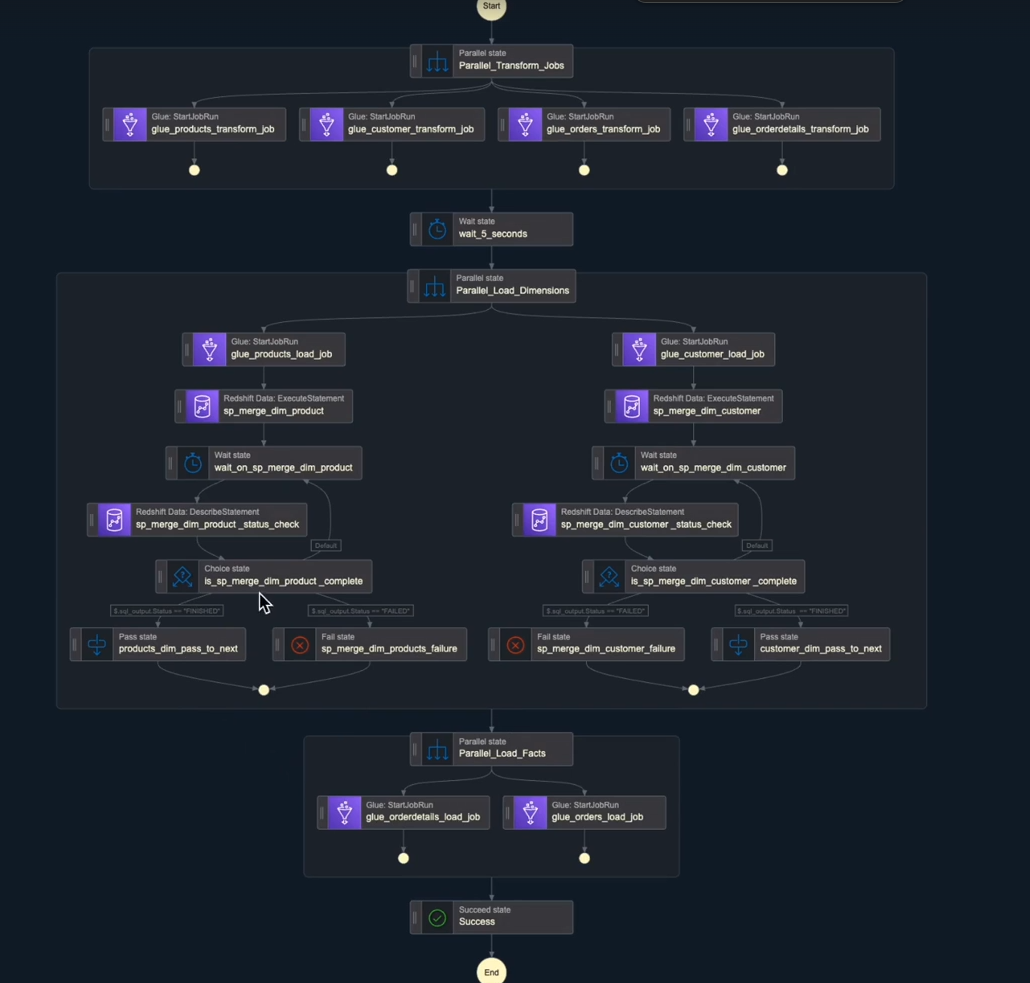
AWS Glue transformed Bronze-layer CSV files into clean, structured Silver-layer datasets. Jobs parsed and cleaned values, enforced schema, renamed columns to snake\_case, and partitioned by ingestion date. The Glue Data Catalog tracked metadata, making data easily queryable. This step ensured analysts worked with consistent, high-quality datasets.

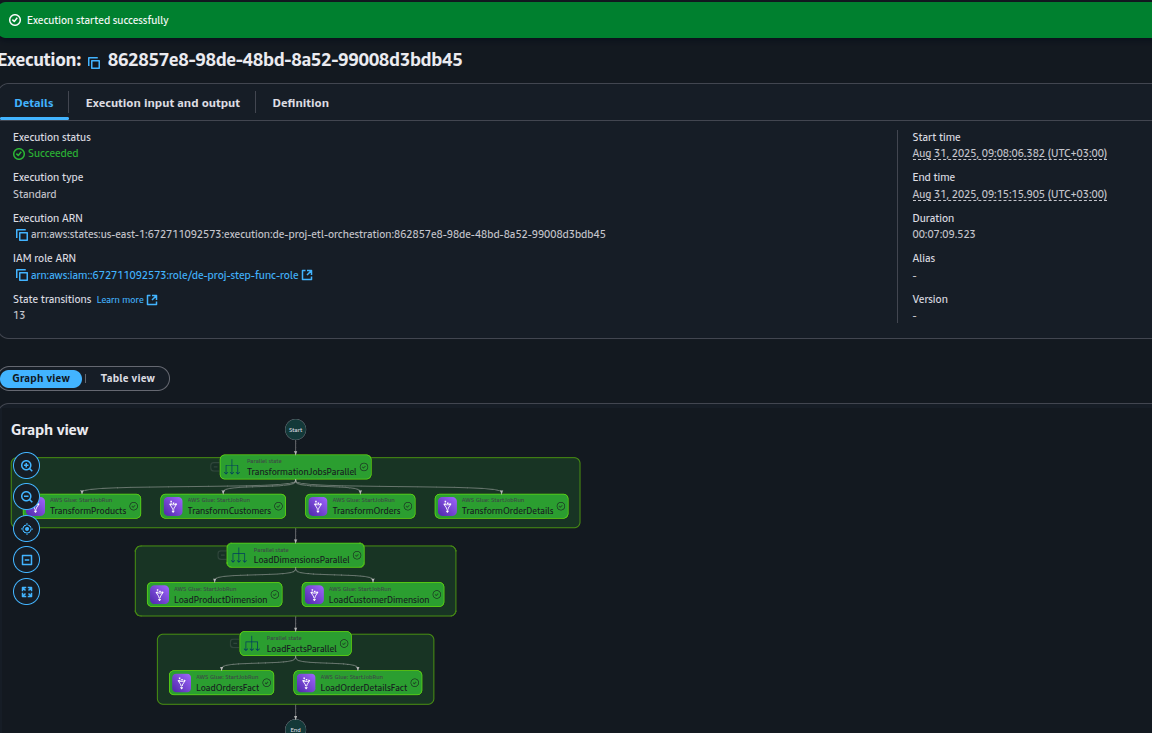
# 7. Phase 4 – Warehousing & Analytics (Gold Layer)

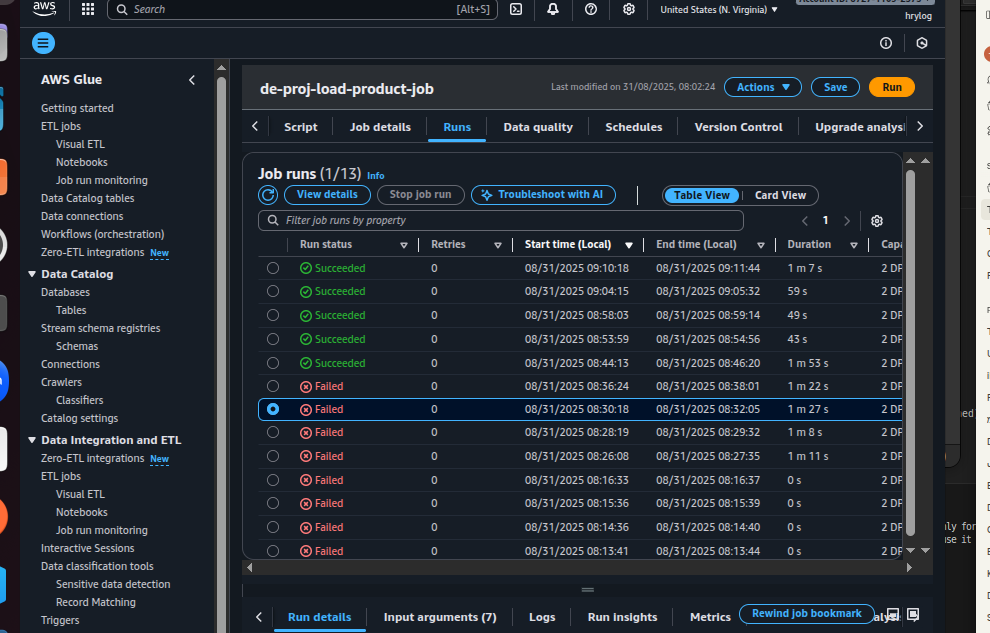
Amazon Redshift served as the data warehouse, optimized for analytical queries. Incremental merges were performed into dimension tables via stored procedures, while fact tables were append-only. This shift from MySQL to Redshift enabled efficient, large-scale analytics.

# 8. Phase 5 – Orchestration with Step Functions

AWS Step Functions orchestrated Glue ETL jobs and Redshift stored procedures. Our state machine executed transformations in parallel, then loaded dimensions and facts. This declarative JSON-driven orchestration enabled retries, error handling, and monitoring.







# 9. Phase 6 – Visualization with QuickSight

Amazon QuickSight dashboards transformed data into insights. KPIs tracked revenue, orders, customer growth, and top-selling products. Interactive visuals allowed filtering and drill-down, democratizing analytics.

# 10. Security & IAM

Security best practices were followed:  
- IAM least-privilege policies.  
- KMS encryption for data at rest in S3 and Redshift.  
- SSL/TLS for data in transit.  
- Secrets Manager for credentials.  
- Avoided 0.0.0.0/0 exposure in production environments.

# 11. Operational Guide (Runbook)

Daily pipeline flow:  
1. Lambda generates new data in RDS.  
2. DMS captures changes and writes to Bronze S3.  
3. Glue Crawlers update schema.  
4. Step Functions trigger ETL jobs to Silver and Gold.  
5. Redshift updated with new facts/dimensions.  
6. QuickSight refreshes dashboards.  
  
This process runs automatically with EventBridge scheduling.

# 12. Challenges & Solutions

- Networking: fixed with VPC endpoints for S3.  
- CDC reliability: resolved with binlog\_format = ROW.  
- Data quality: Glue handled type casting, nulls, naming consistency.  
- Error handling: Step Functions retries & notifications.  
- Cost: daily destroy/recreate pattern with Terraform.

# 13. Conclusion

The Revamp Style Data Engineering Project demonstrates how AWS can transform raw operational data into business-ready insights. With IaC (Terraform), managed ETL (Glue), orchestration (Step Functions), warehousing (Redshift), and visualization (QuickSight), the pipeline is reproducible, resilient, and scalable. From chaos to clarity, Revamp Style now operates with real-time, data-driven decision-making.

# 14. Appendix – Running the Project

To run this project:  
  
- Install Terraform and AWS CLI.  
- Clone the repository and run setup.sh to create structure.  
- Configure variables in terraform.tfvars (region, RDS creds, S3 bucket).  
- Deploy infrastructure: ./deploy.sh all  
- Use Faker-based Lambda generator to populate RDS.  
- Ensure DMS tasks are running for CDC.  
- Trigger Step Functions or EventBridge schedule.  
- Open QuickSight dashboard for insights.  
  
Dependencies (requirements.txt):  
mysql-connector-python==8.0.33  
faker==19.12.0