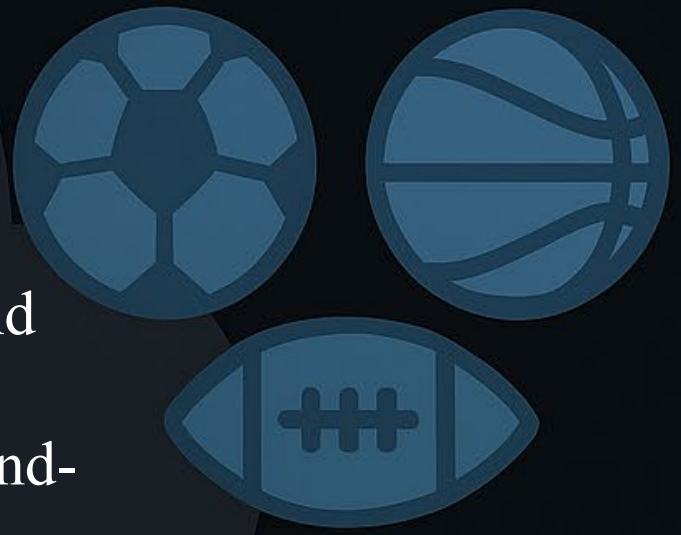
ShieldScrape

Secure & Scalable Cloud-Based Web Data Collection

Project Goal

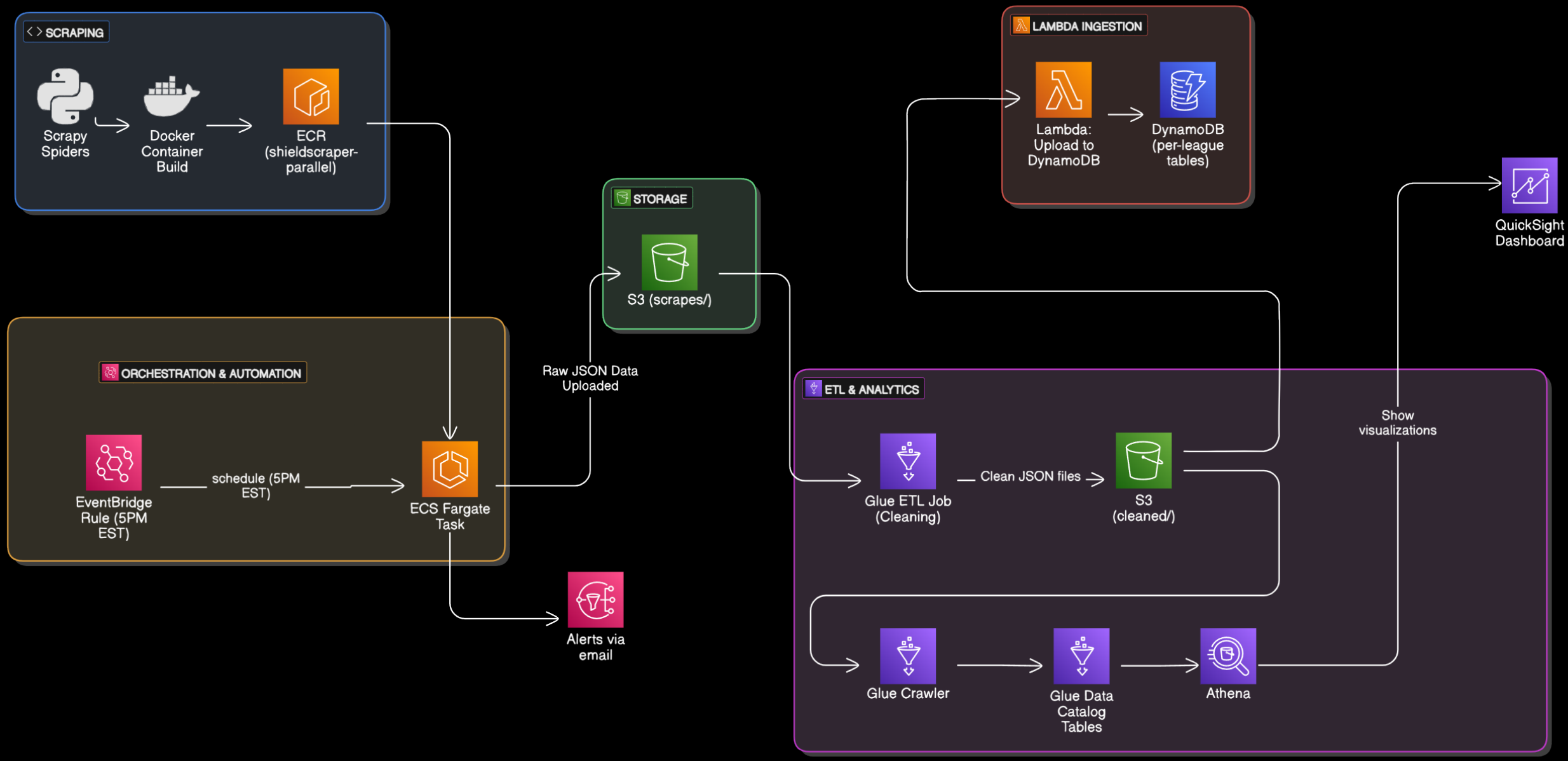
- Collect live ESPN sports data.
- Automate cleaning, storage, querying, and visualization.
- Achieve serverless end-to-end cloud architecture.



Team Contributions

- **Shivali Mate** - Developed the job scrapers and implemented containerization using Docker for scalable deployments.
- **Varun Sonawane** - Designed and built the ETL pipeline integrating AWS services such as Glue, S3, DynamoDB, and Athena.
- **Aryan Dhuru** - Implemented end-to-end automation, including job scheduling, orchestration with EventBridge, and scripting of cloud workflows.
- **Suhana Ambol** – Created impactful data visualizations using Amazon QuickSight and led project documentation efforts.

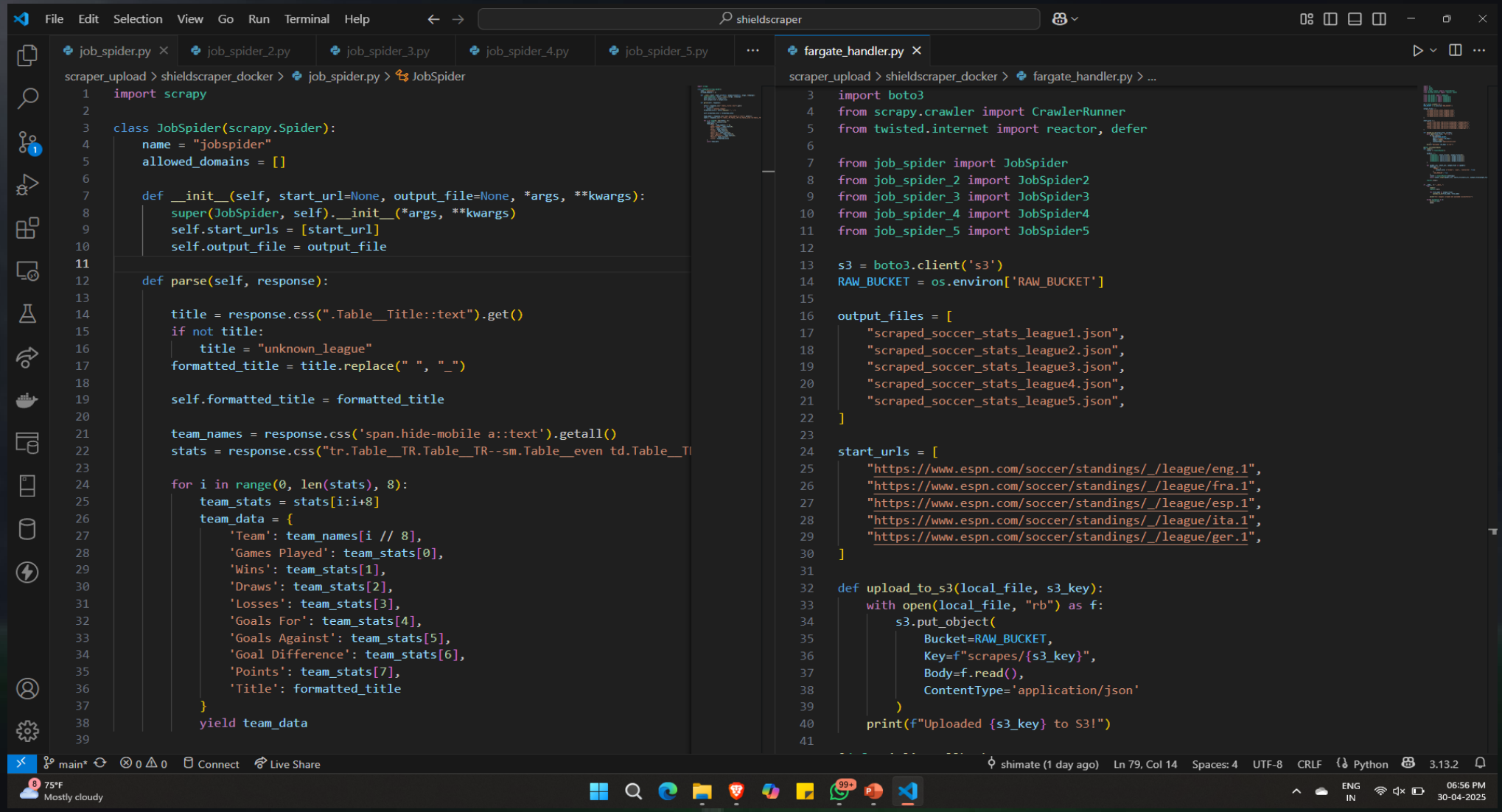
ShieldScraper Cloud Architecture: End-to-End Data Flow



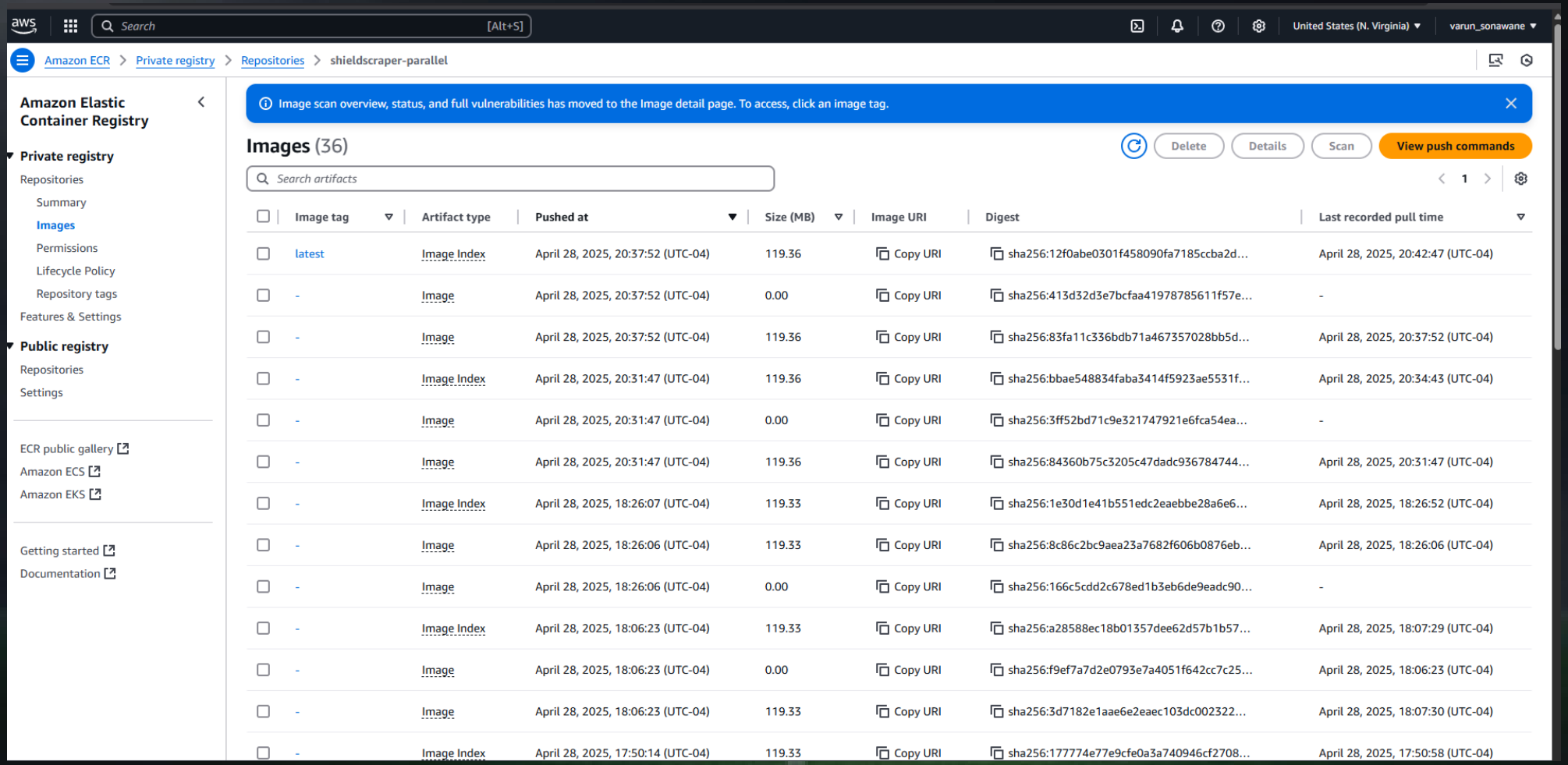
Phase 1: Scraping and Containerization

- Developed multiple Scrapy spiders for different leagues.
- Fields: Team Name, matches played, wins, draws, losses, goals, and points.
- Created a Fargate handler to run spiders in parallel.
- Dockerized the entire scraping setup.
- Pushed Docker image to AWS ECR (**shieldscraper-parallel)** repo.

Phase 1: Scraping and Containerization



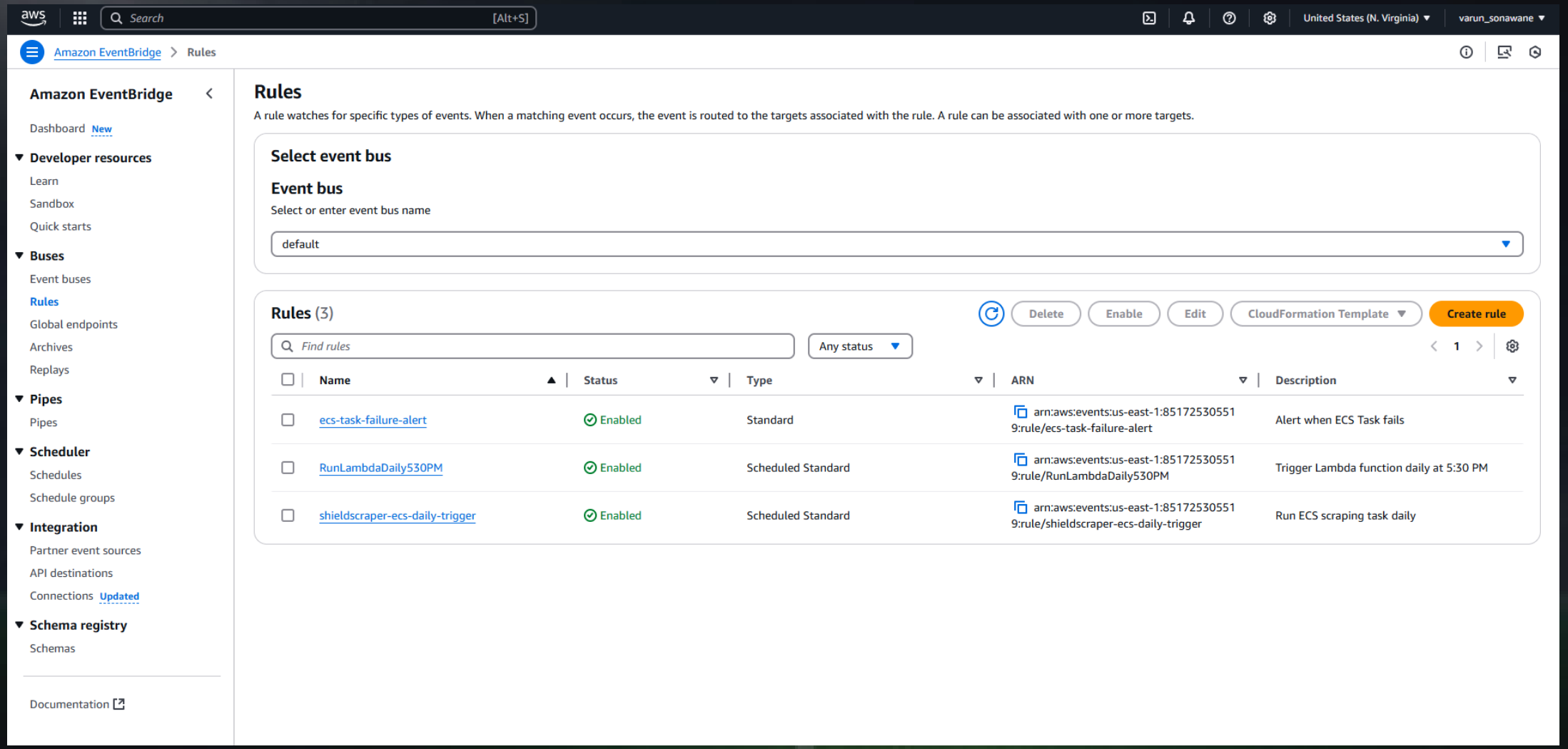
Phase 1: Scraping and Containerization



Phase 2: Scheduled Trigger using EventBridge

- Created AWS EventBridge rule with daily trigger at 05:00 pm EST.
- Event triggers ECS task in shieldscraper-cluster1.
- Task definition: **shieldscraper-parallel-task**

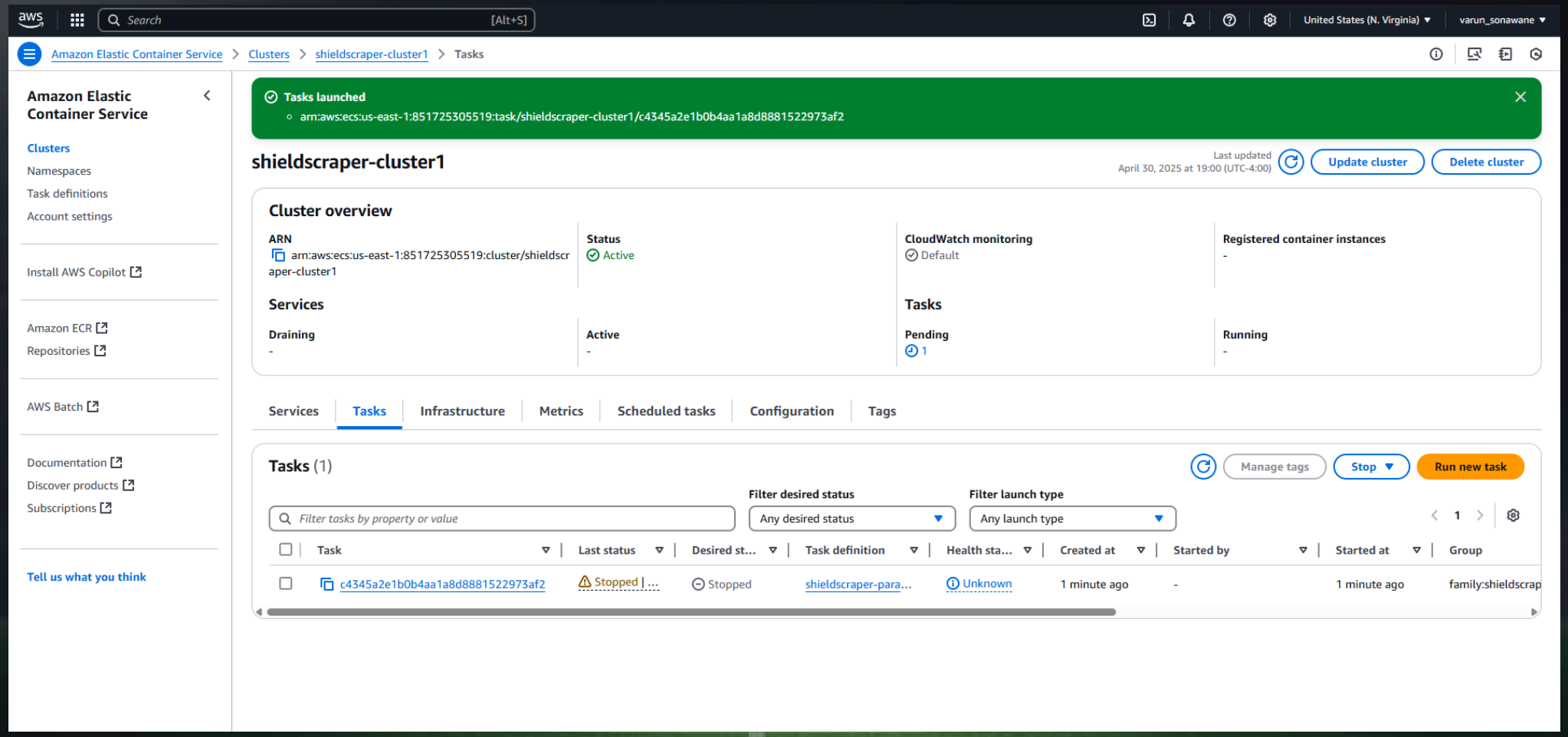
Phase 2: Scheduled Trigger using EventBridge



Phase 3: Data Collection (Scraping)

- Fargate ECS Task launches Scrapy spiders.
- Fresh soccer standings are scraped.
- Output stored to Amazon S3 Bucket.

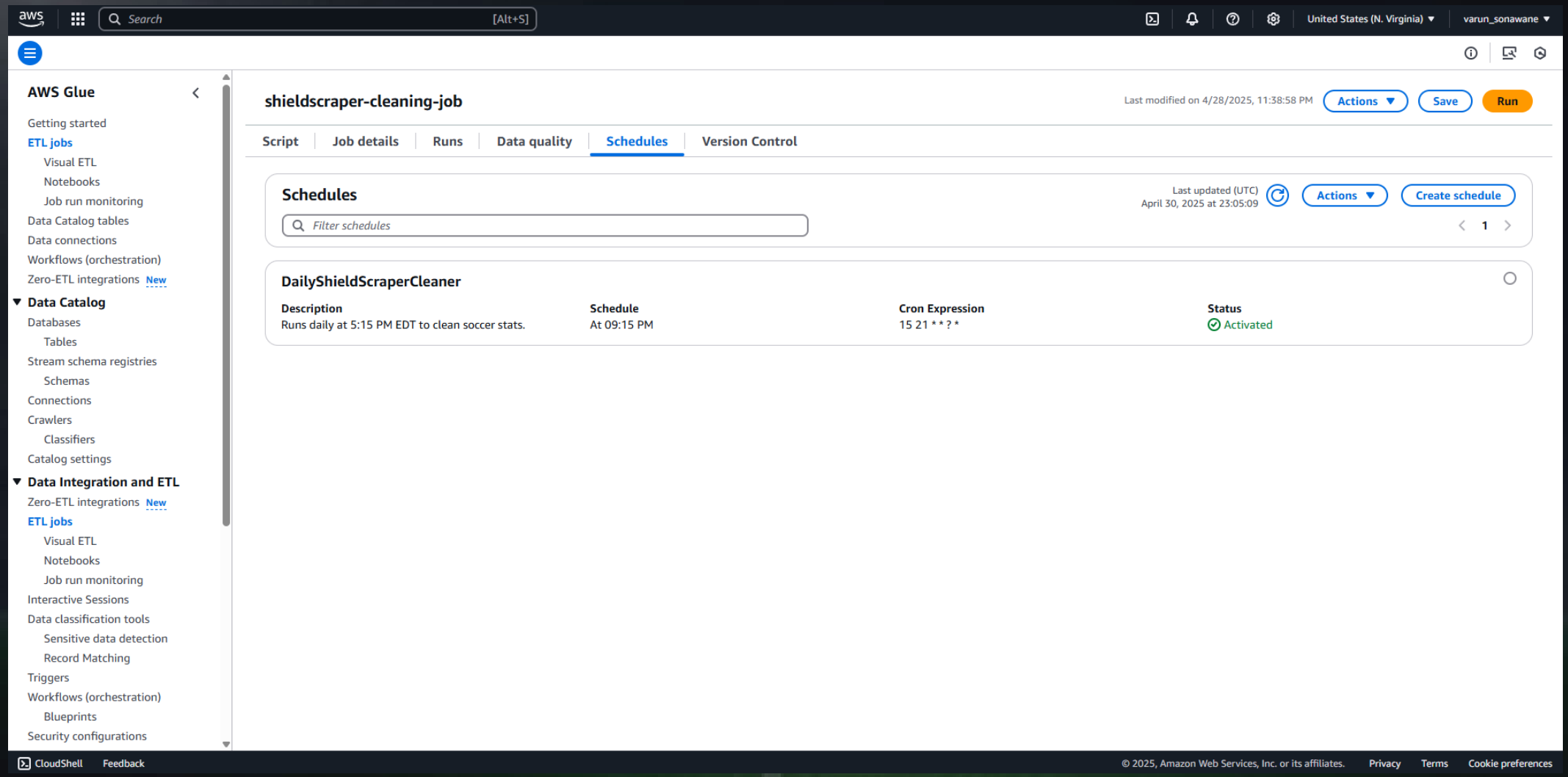
Phase 3: Data Collection (Scraping)



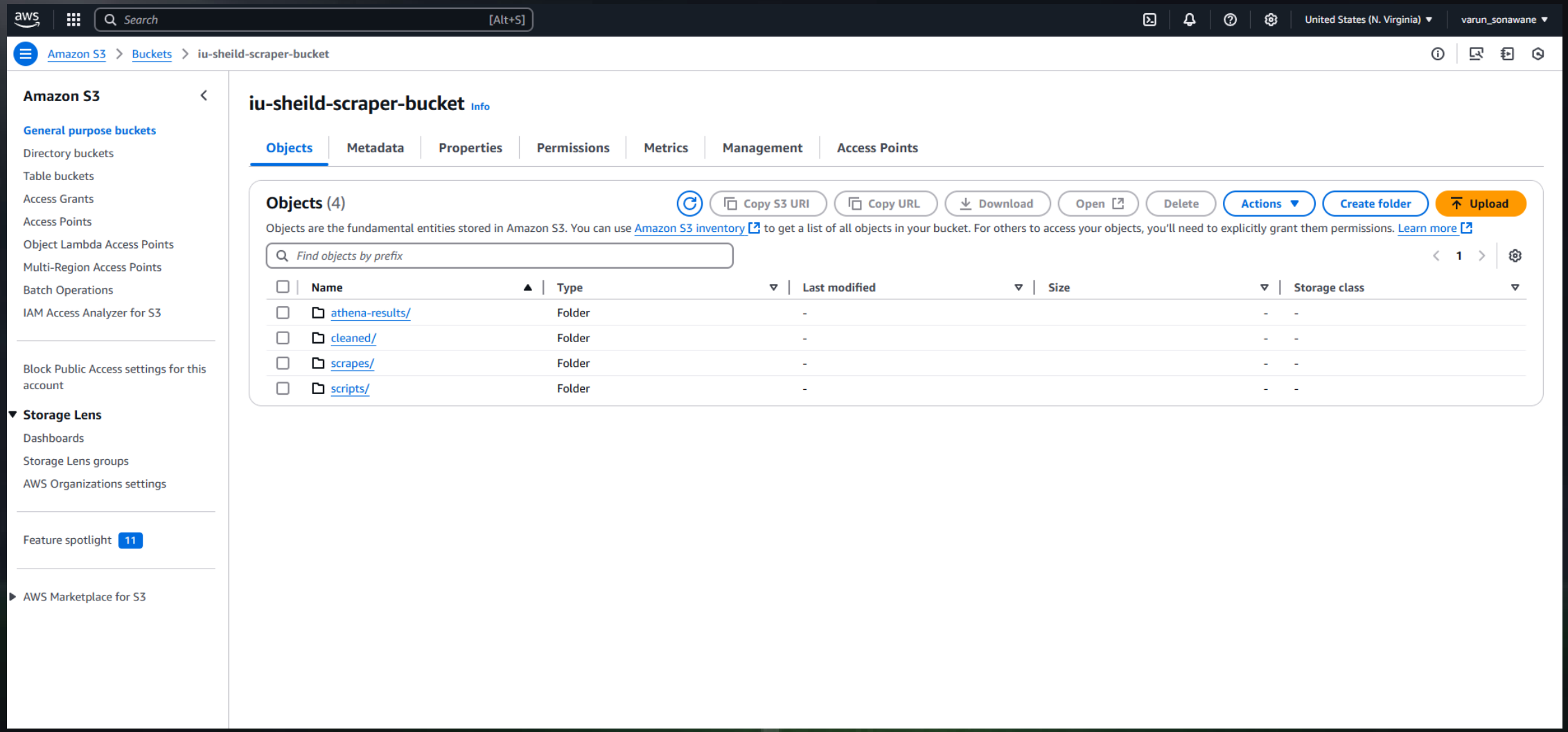
Phase 4: ETL with AWS Glue

- AWS Glue job (shieldscraper-cleaning-job) starts.
- Reads raw files from S3.
- Cleans and formats the data.
- Stores the clean JSON into: s3://iu-sheild-scraper-bucket/cleaned/

Phase 4: ETL with AWS Glue



Phase 4: ETL with AWS Glue



Raw Data VS Cleaned Data

Raw data


```
{"Team": "Real Madrid", "Games Played": "33", "Wins":
"22", "Draws": "6", "Losses": "5", "Goals For": "66",
"Goals Against": "31", "Goal Difference": "+35", "Points":
"72", "Title": "Spanish_LALIGA"},
```

Raw Data VS Cleaned Data

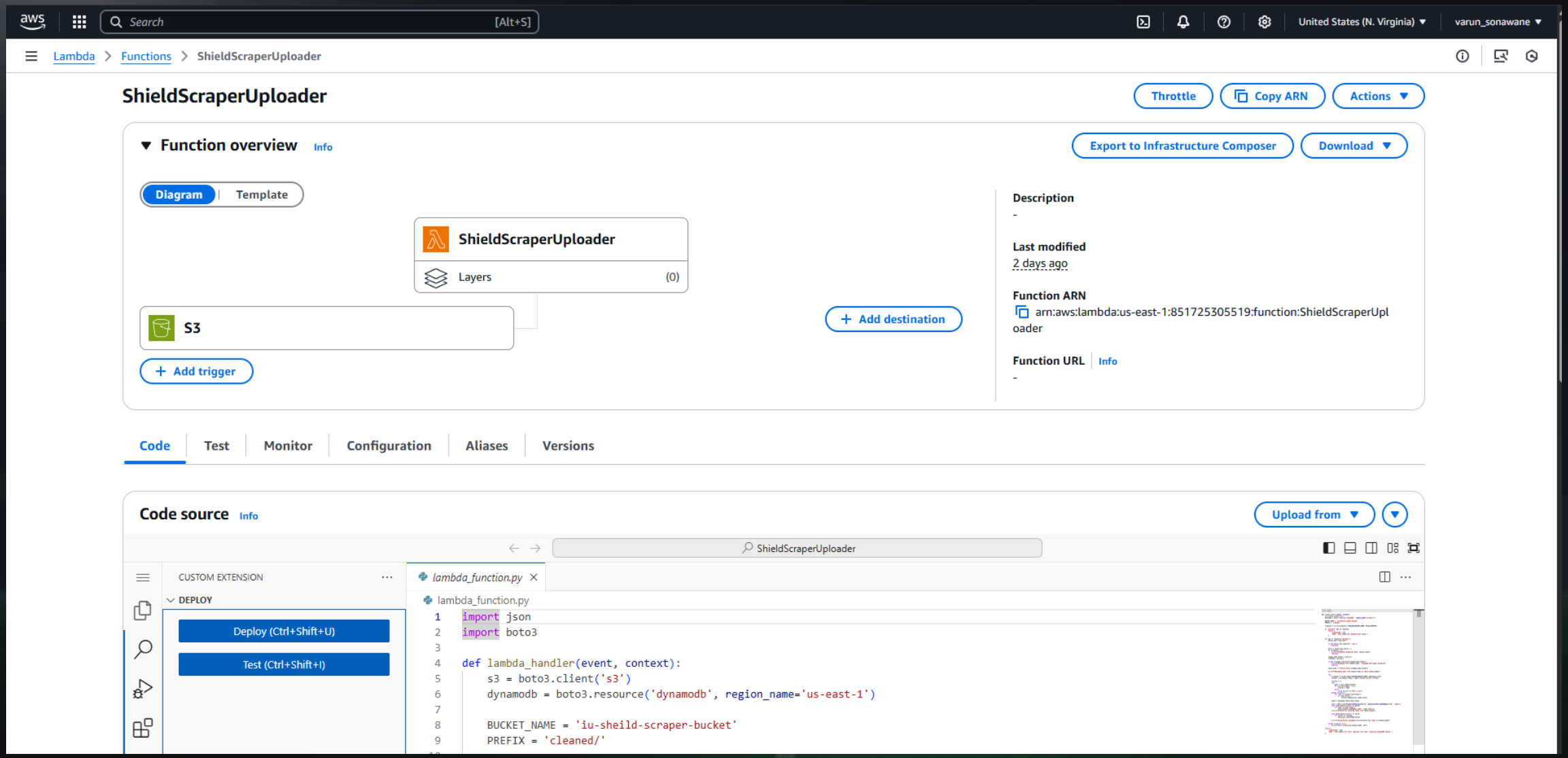
Cleaned data

{"Team": "Real Madrid", "Games Played": 33, "Wins": 22, "Draws": 6, "Losses": 5, "Goals For": 66, "Goals Against": 31, "Goal Difference": 35, "Points": 72}

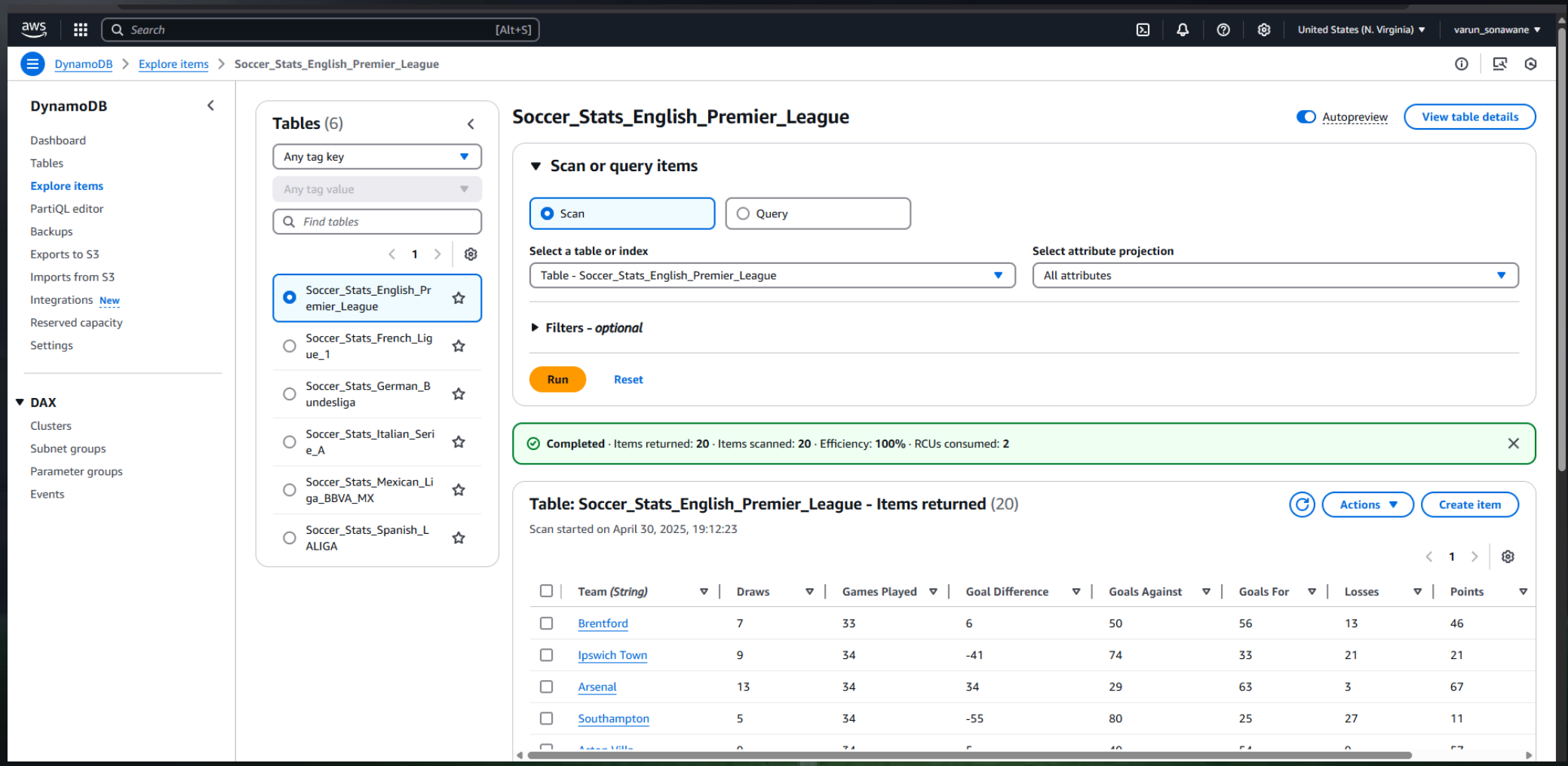
Phase 5: DynamoDB update via Lambda

- AWS Lambda function triggers after ETL completes.
- Pulls cleaned data from /cleaned/.
- Uploads records into separate DynamoDB tables (per league).

Phase 5: DynamoDB update via Lambda



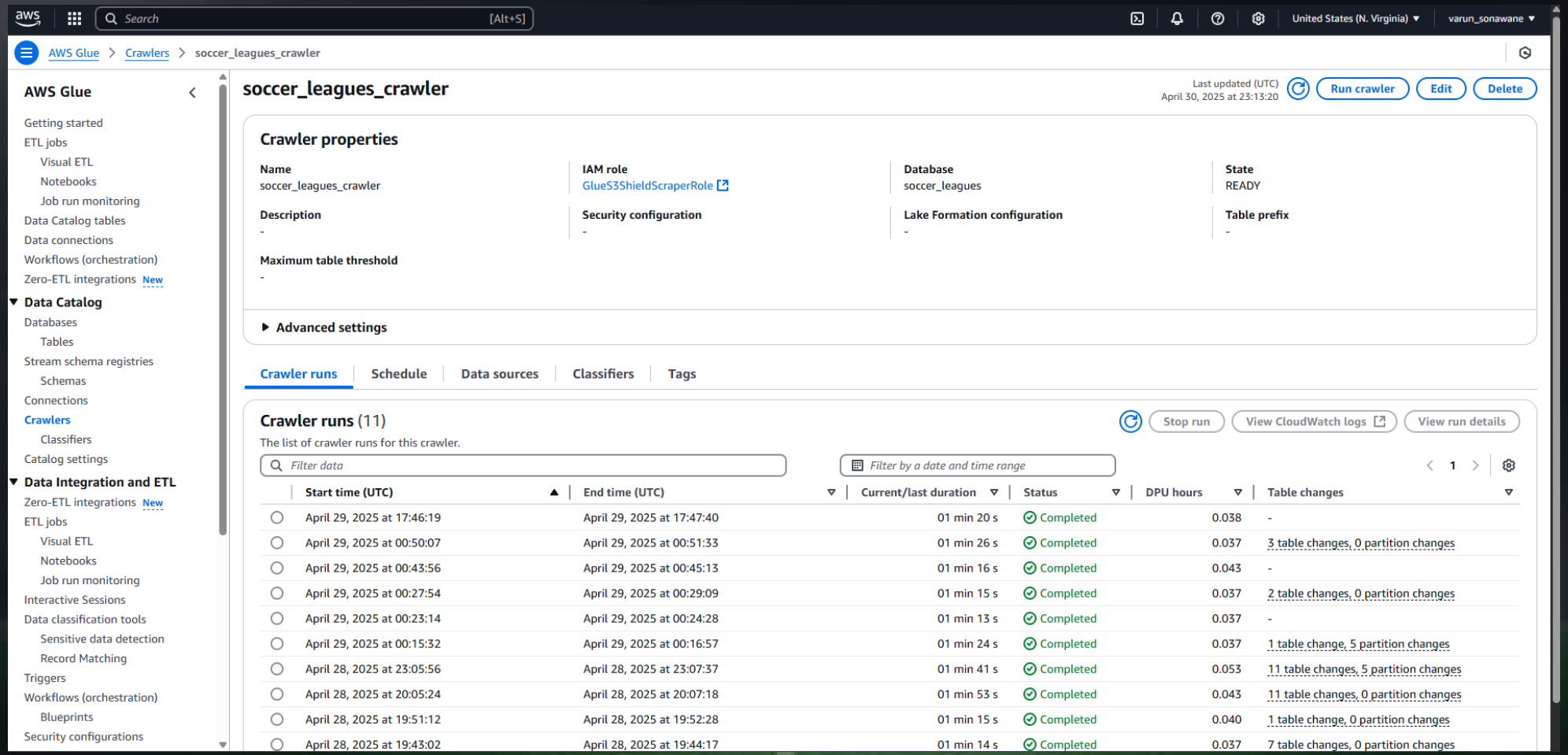
Phase 5: DynamoDB update via Lambda



Phase 6: Glue Crawler Refresh

- Lambda function triggers AWS Glue Crawler.
- Updates Glue Catalog with latest S3 data structure.

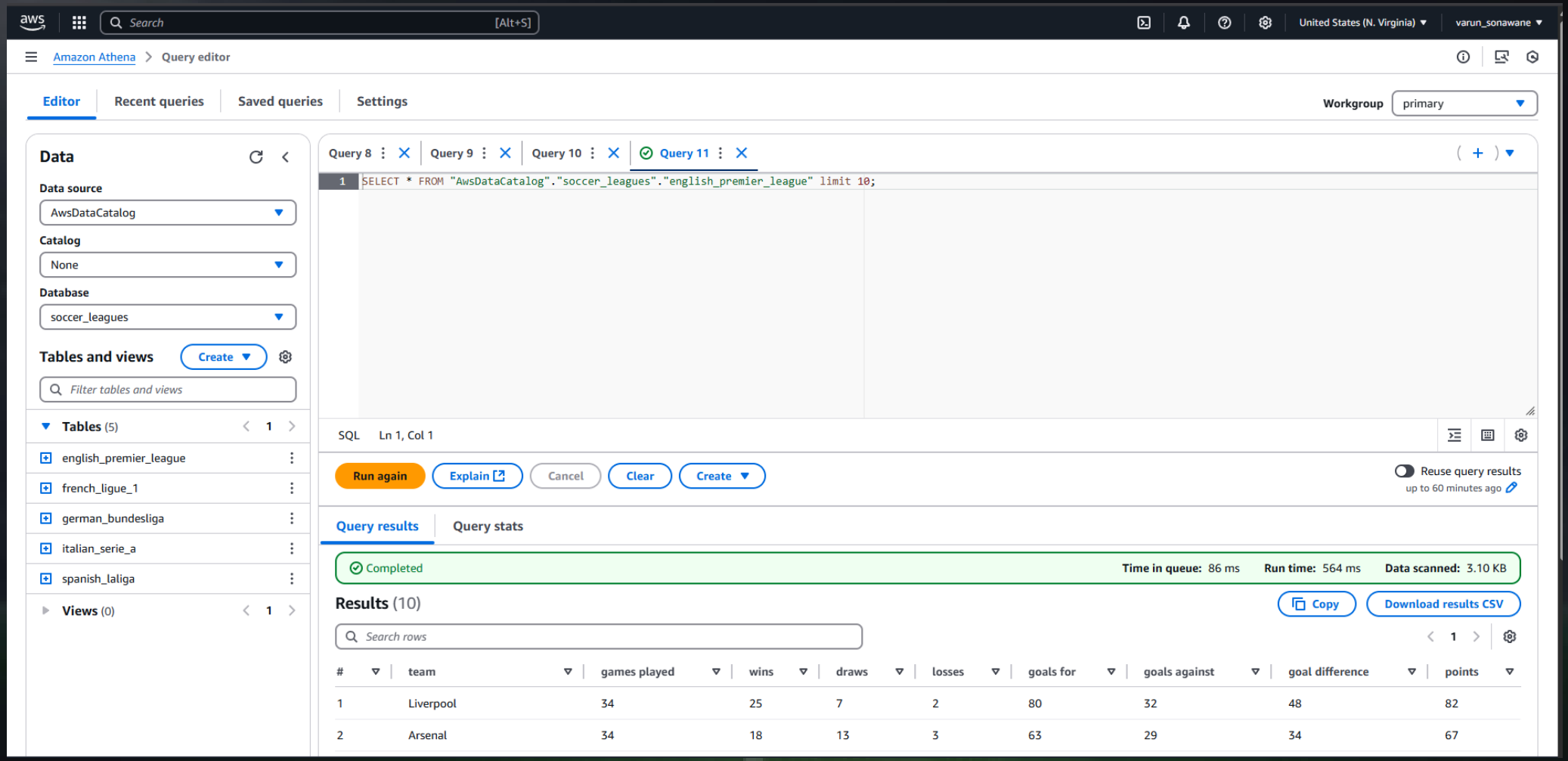
Phase 6: Glue Crawler Refresh



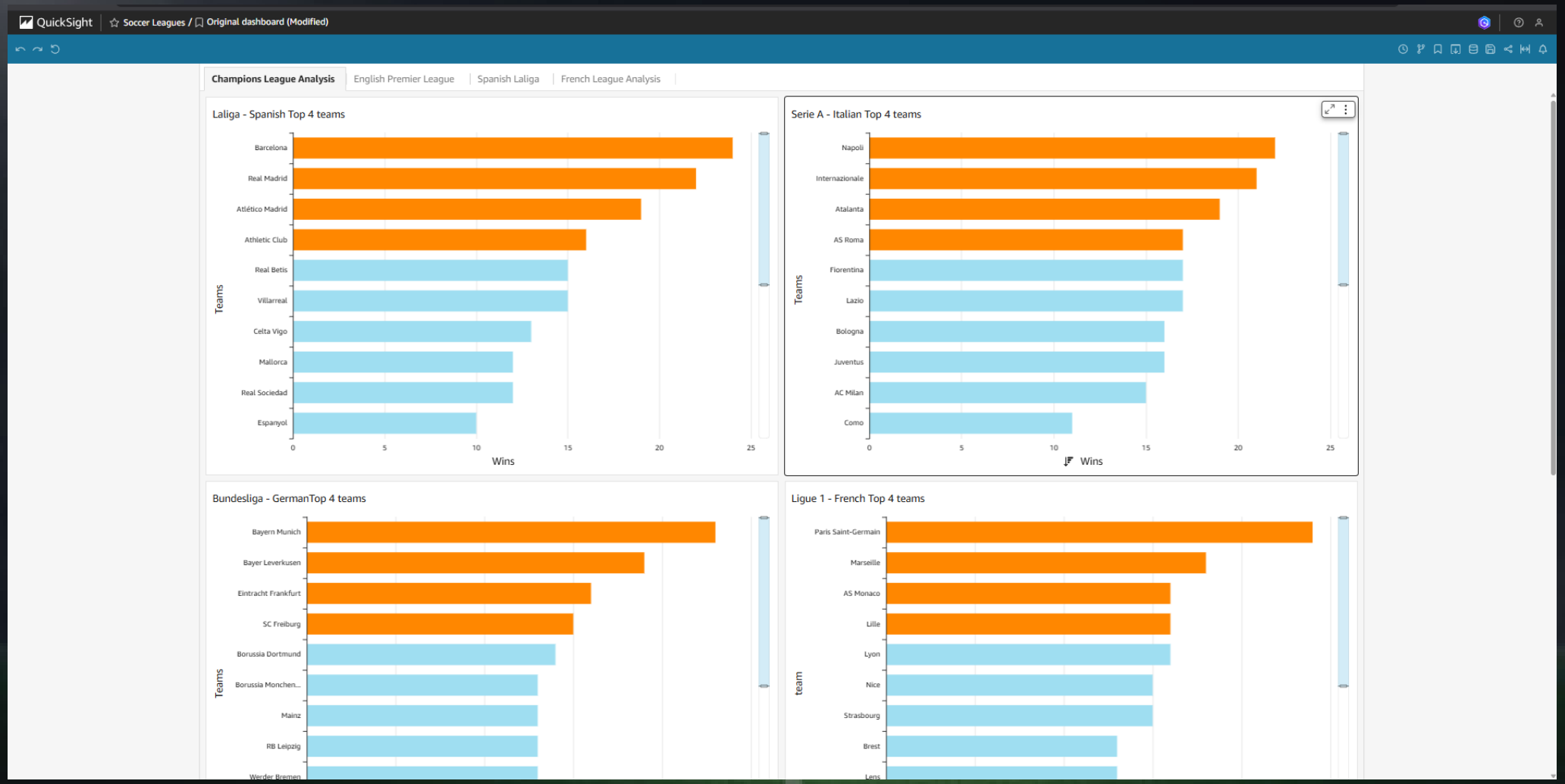
Phase 7: Query via Amazon Athena and Visualization with QuickSight

- Glue Tables auto-created from Crawler metadata.
- Data instantly queryable using Amazon Athena SQL.
- Build QuickSight dashboards linked to Athena tables.
- Dashboard automatically refreshes with latest soccer data.

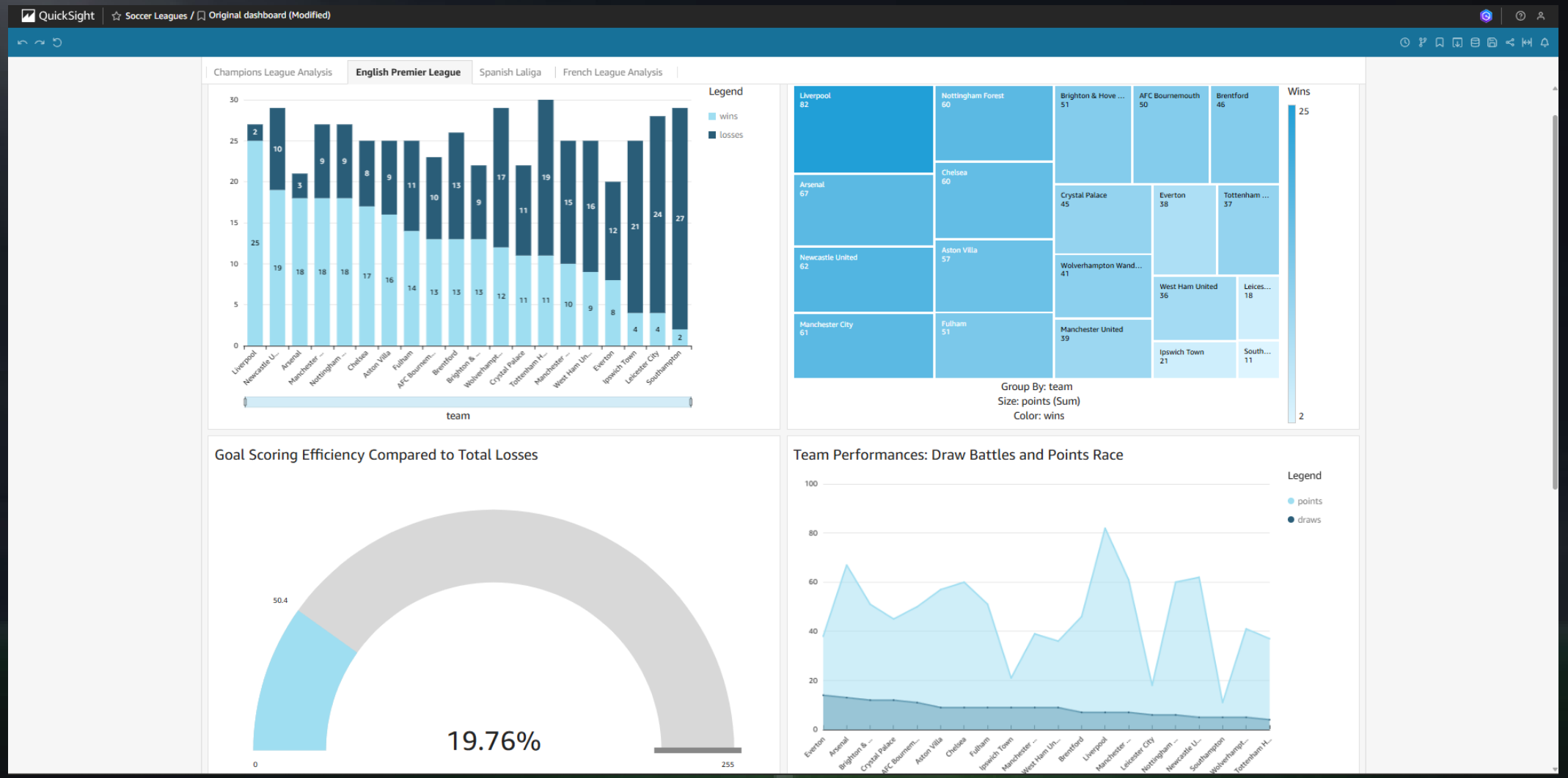
Phase 7: Query via Amazon Athena and Visualization with QuickSight



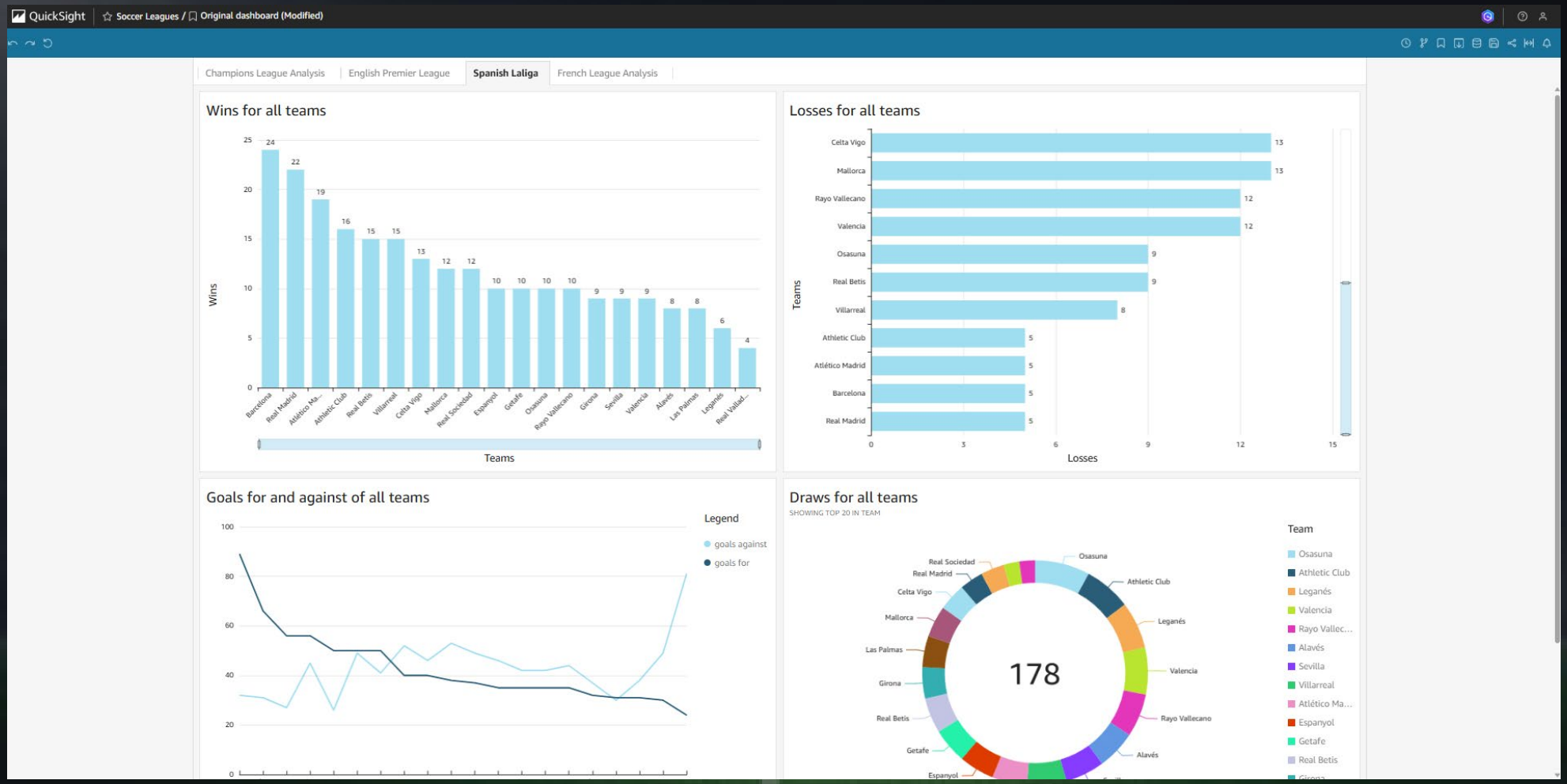
Phase 7: Query via Amazon Athena and Visualization with QuickSight



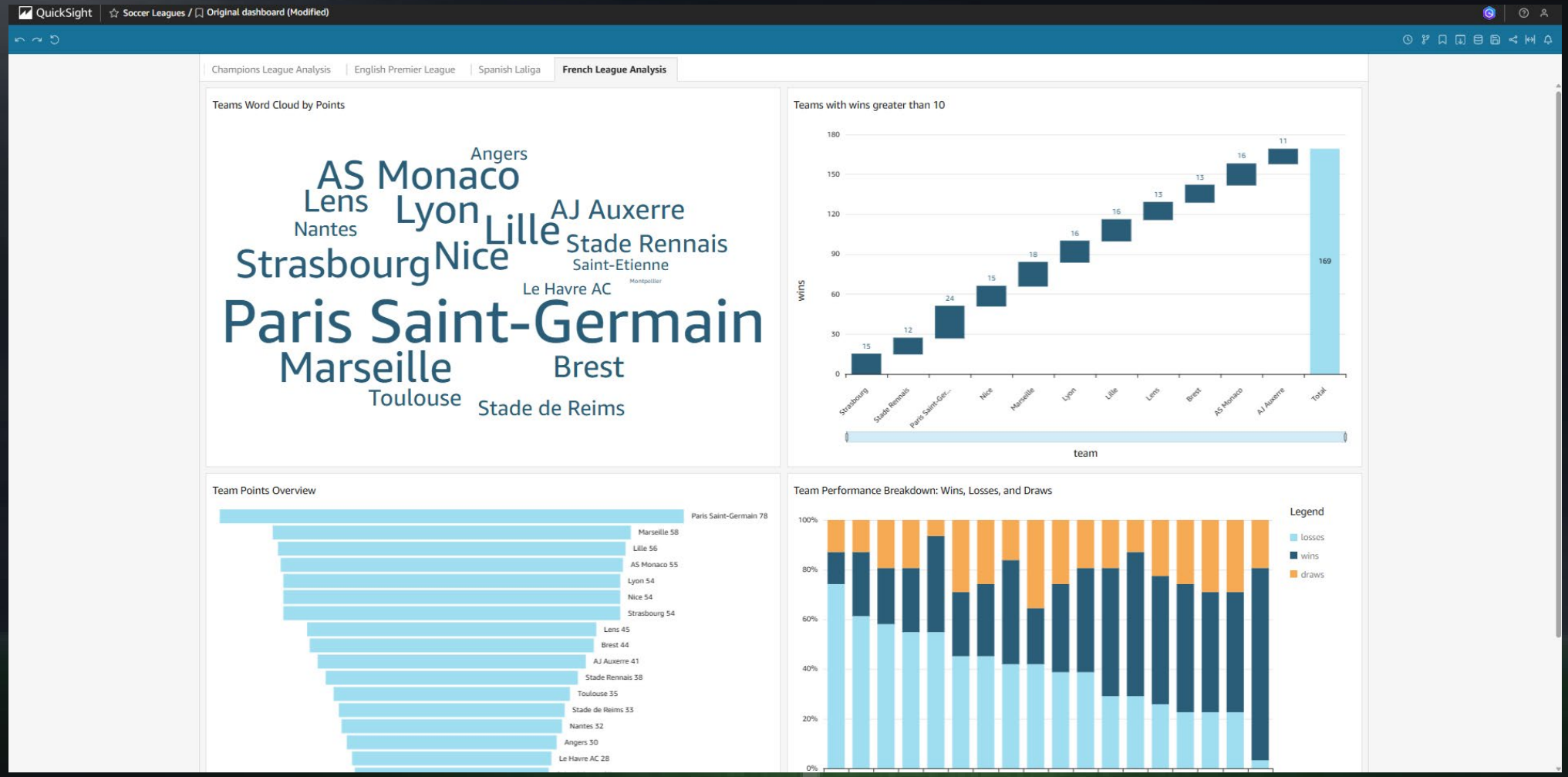
Phase 7: Query via Amazon Athena and Visualization with QuickSight



Phase 7: Query via Amazon Athena and Visualization with QuickSight



Phase 7: Query via Amazon Athena and Visualization with QuickSight



Execution Timeline
Timeline of a ShieldScraper Run

Time (EST)	Actions
05:00 PM	EventBridge triggers ECS Fargate
05:10 PM	Scraping completes, saved to S3
05:15 PM	Glue ETL cleans and stores to /cleaned/
05:25 PM	Lambda uploads to DynamoDB
05:30 PM	Glue Crawler refreshes catalog
06:00 PM	Dashboards show updated data

Conclusion

- ShieldScraper achieves fully automated, scalable sports data collection and visualization using serverless cloud technologies, delivering fresh insights daily with minimal operational effort.
- The project demonstrates strong system design, cloud automation, and scalable data engineering practices.

THANK YOU