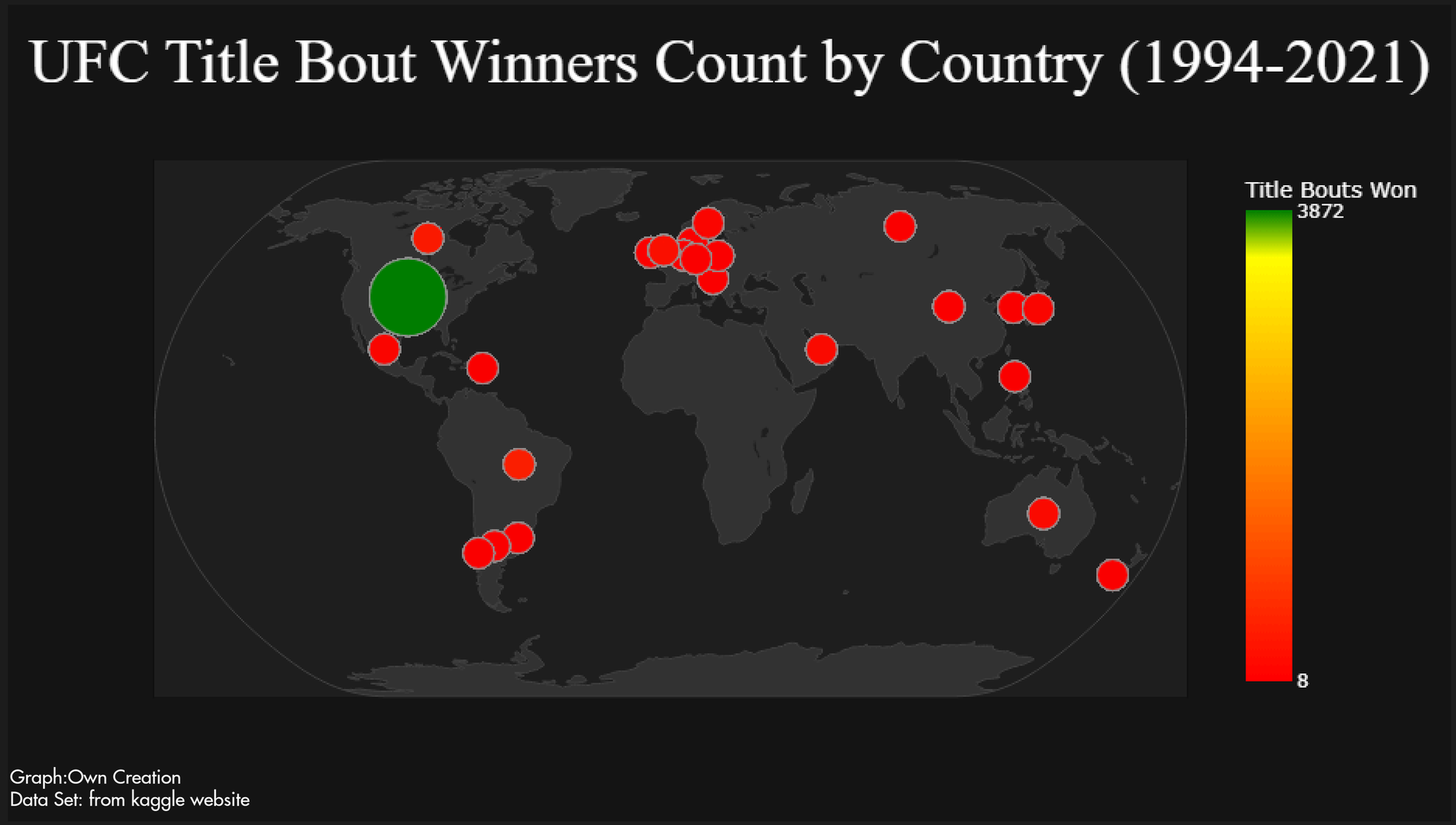
Octagon Insights: Unleashing UFC Data with AWS Data Pipeline



Graph:Own Creation
Data Set: from kaggle website

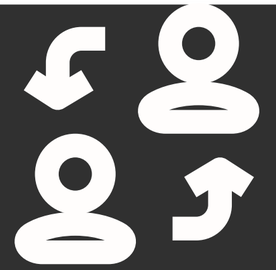
A short guide on data engineering and analysis

Agenda

0) Extract: Sourcing Your Data

- Get your data from databases or data lakes.
- Examples: Power BI dataset, AWS S3 Bucket, Redshift, SQL databases.
- Why This Matters: Quality data extraction sets the foundation for accurate analysis.

3) Load: HTTP API Gateway Endpoint



- Processes data using Pandas
- Converts data types and formats

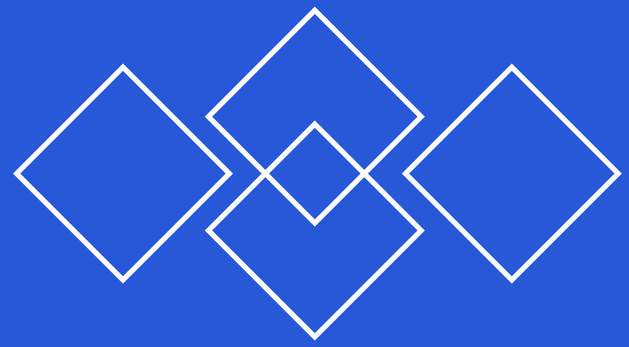
2) Load: Using an HTTP API Gateway endpoint

- Trigger the Lambda function to process data on-demand.
- Place requests from a Jupyter notebook for flexibility.
- Create additional visualizations based on transformed data.

AWS ETL Pipeline

Extract

Unleash UFC fight data from AWS Data Lakes and Databases. Craft SQL queries with Python magic to pinpoint specific datasets, like title bouts or knockout wins. This targeted approach turbocharges efficiency. Use AWS SDK for swift data retrieval. For massive datasets, embrace incremental extraction to keep performance in fighting shape.



Transform

Leverage AWS Lambda functions for fast data transformation. Set up API Gateway as your data's bouncer, complete with bearer token for access control. Configure timeout (max 15 min) to avoid data KOs. Use Pandas to filter UFC winners and standardize data faster than a ref's stoppage. Implement error handling to make your pipeline as resilient as a champion's mindset.

Load

Lambda supercharges the load process, preparing UFC fight data for analysis. It transforms raw data into analysis-ready formats, enabling near-real-time insights. Leverage Pandas for quick statistical computations on fight outcomes. Create visualizations with Pyplot, showcasing summarized data. Lambda's speed allows for flexible, on-demand data preparation, empowering deeper dives into fighter performance and trend analysis.

Summary

Why HTTP API above REST API?

HTTP API offers lower latency and cost, crucial for real-time fight insights. It provides simpler setup, built-in CORS support, and faster deployments. Better performance for high-volume requests makes it ideal for serving fight stats quickly. Lacks some advanced features of REST API, but matches our UFC data serving needs perfectly.

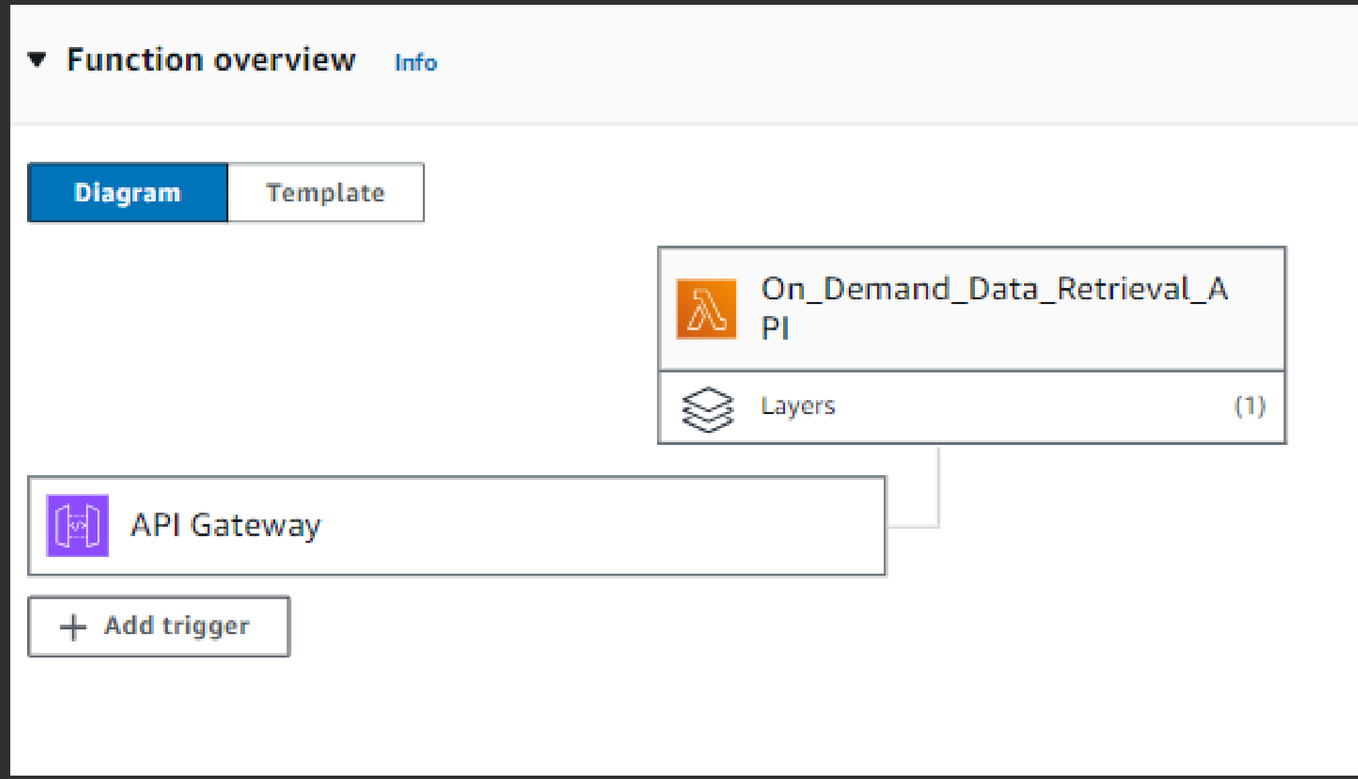
When not to use Lambda functions?

Avoid Lambda for long-running tasks (>15 min), consistent high-volume processing, or applications needing persistent state. Not ideal for resource-intensive computations or when cold start latency is critical. Consider alternatives for predictable, steady workloads or complex, stateful applications.

Why use an AWS lambda function?

Lambda enables rapid, cost-effective data analysis. It scales automatically for fight night spikes, processes data in seconds, and minimizes operational overhead. This serverless approach delivers quick insights for real-time decision-making in the dynamic world of MMA.

AWS Lambda Function Schema



Get Full Access to the project code;

Portafolio:
https://github.com/tomasdevelopment/Automation-Flows-with-Python-/tree/main
Linkedin: https://www.linkedin.com/in/tomas-suarez/