DATA ENGINEERING PROJECT

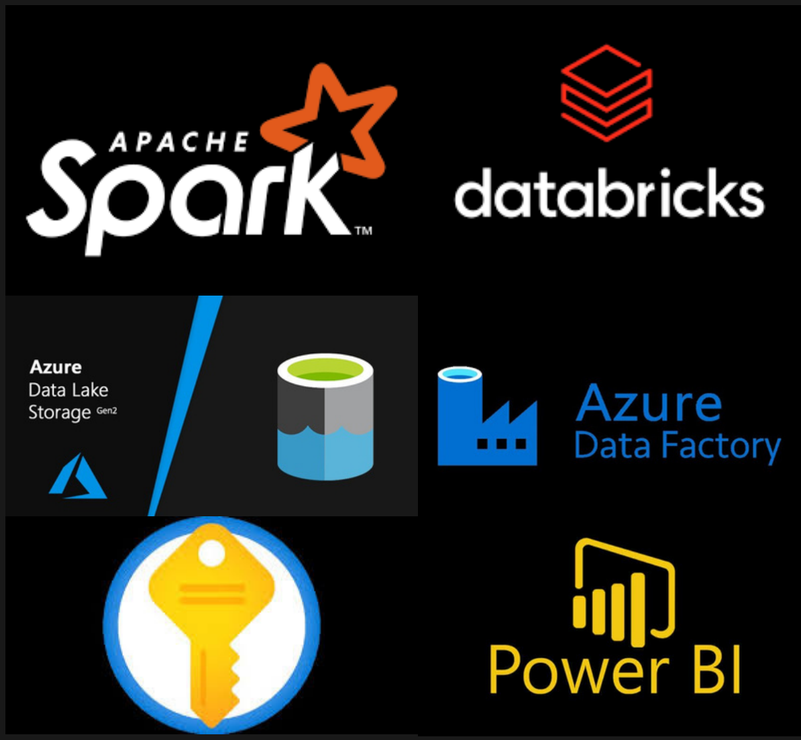
Azure Databricks | Delta Lake | Azure Data Factory | Power BI

Building a cloud-based Lakehouse architecture using real-world Formula 1 racing data.

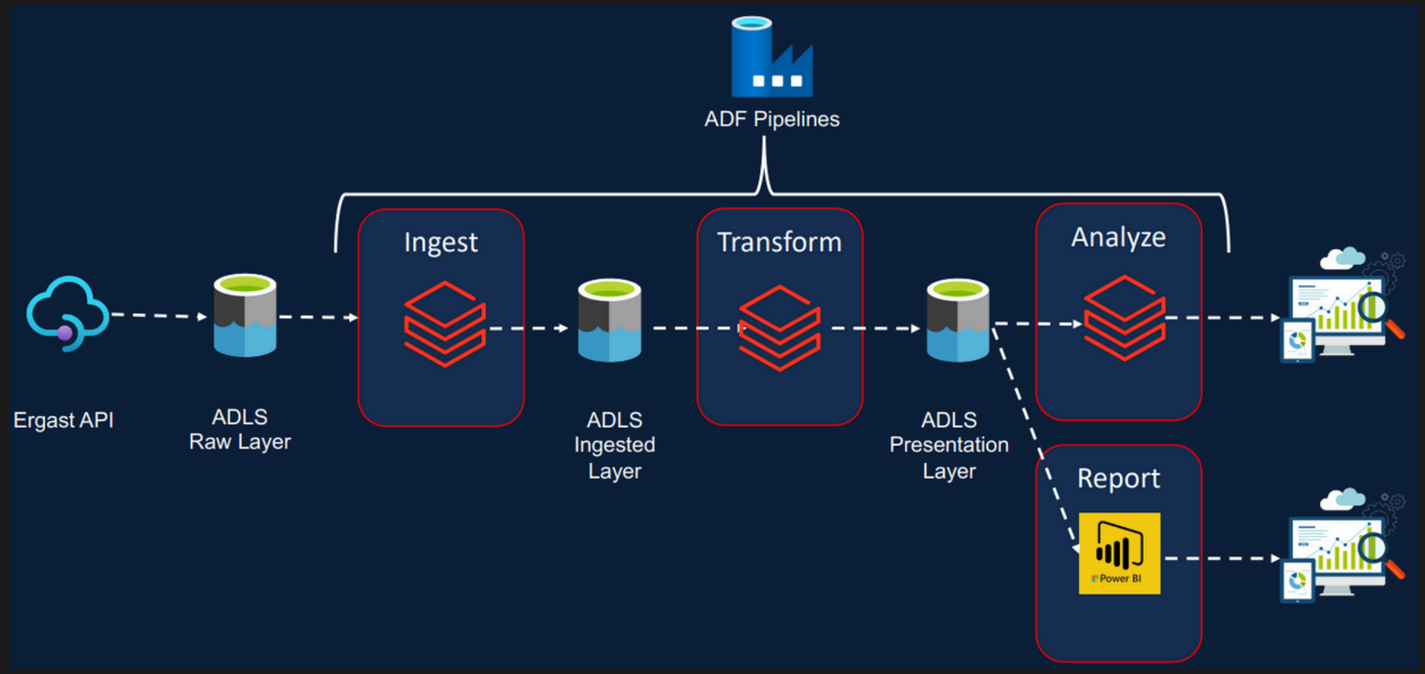
TECH STACK

Focus: Data ingestion, transformation, orchestration, visualization, and governance.

- Azure Databricks
- Azure Data Lake Gen2
- Azure Data Factory (ADF)
- Azure Key Vault
- Power BI
- PySpark & Spark SQL



SOLUTION ARCHITECTURE



Extract

1. Ingest raw data from Ergast API Formula 1 datasets

Load

2. Store raw data in Azure Data Lake Gen2 in the Bronze Storage Layer

Transform

3. Transform and enrich data in Azure Databricks using PySpark & SQL. Store processed data in Delta Lake:
Bronze → Silver → Gold

Pipeline Orchestration

4. Orchestrate notebooks using Azure Data Factory

VISUALIZE

Visualize insights using Power BI

INGESTING & TRANSFORMING RAW DATA

- Defined explicit schema for raw circuits.csv
- Ingested using .read.csv with schema enforcement
- Applied transformations: column renaming, derived fields
- Added ingestion date and environment metadata

```python
# Define the schema for circuits
from pyspark.sql.types import StructType, StructField, IntegerType, StringType, DoubleType

circuits_schemas = StructType(fields=[StructField("circuitId", IntegerType(), False),
                                      StructField("circuitRef", StringType(), True),
                                      StructField("name", StringType(), True),
                                      StructField("location", StringType(), True),
                                      StructField("country", StringType(), True),
                                      StructField("lat", DoubleType(), True),
                                      StructField("lng", DoubleType(), True),
                                      StructField("alt", DoubleType(), True),
                                      StructField("url", StringType(), True),
])

# Read the CSV file
circuits_df = spark.read \
        .option("header",True) \
        .schema(circuits_schemas) \
        .csv(f'{raw_folder_path}/{var_filedate}/circuits.csv')
```

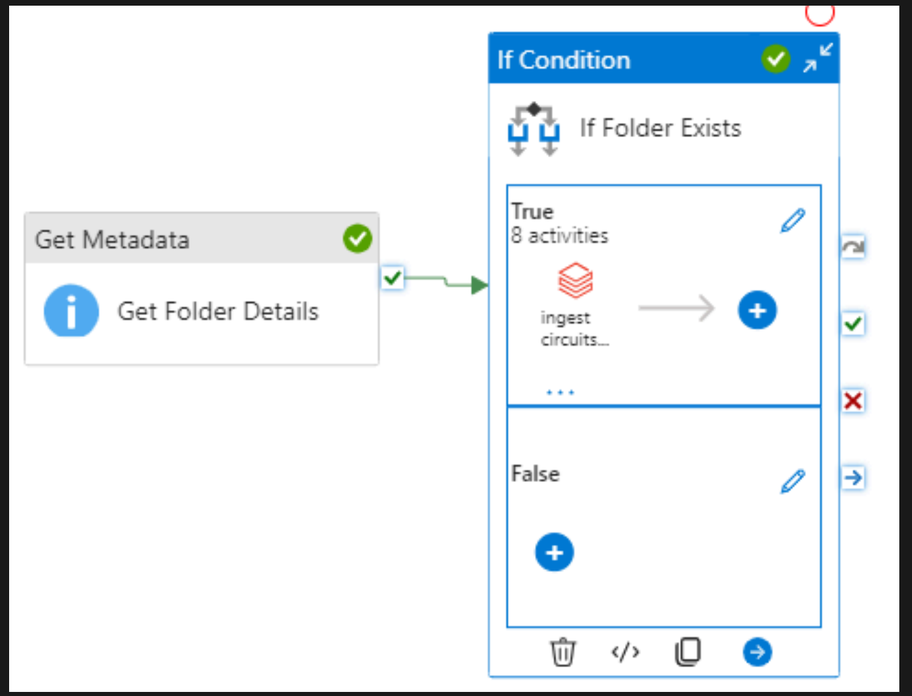
WRITING TRANSFORMED DATA TO DELTA LAKE

- Stored the cleaned circuits dataset in Delta format
- Created managed Delta tables in Databricks
- Enabled faster queries, history tracking & schema evolution

```
circuits_final_df.write \
    .mode("overwrite") \
    .format("delta") \
    .saveAsTable("f1_gold.circuits")
```

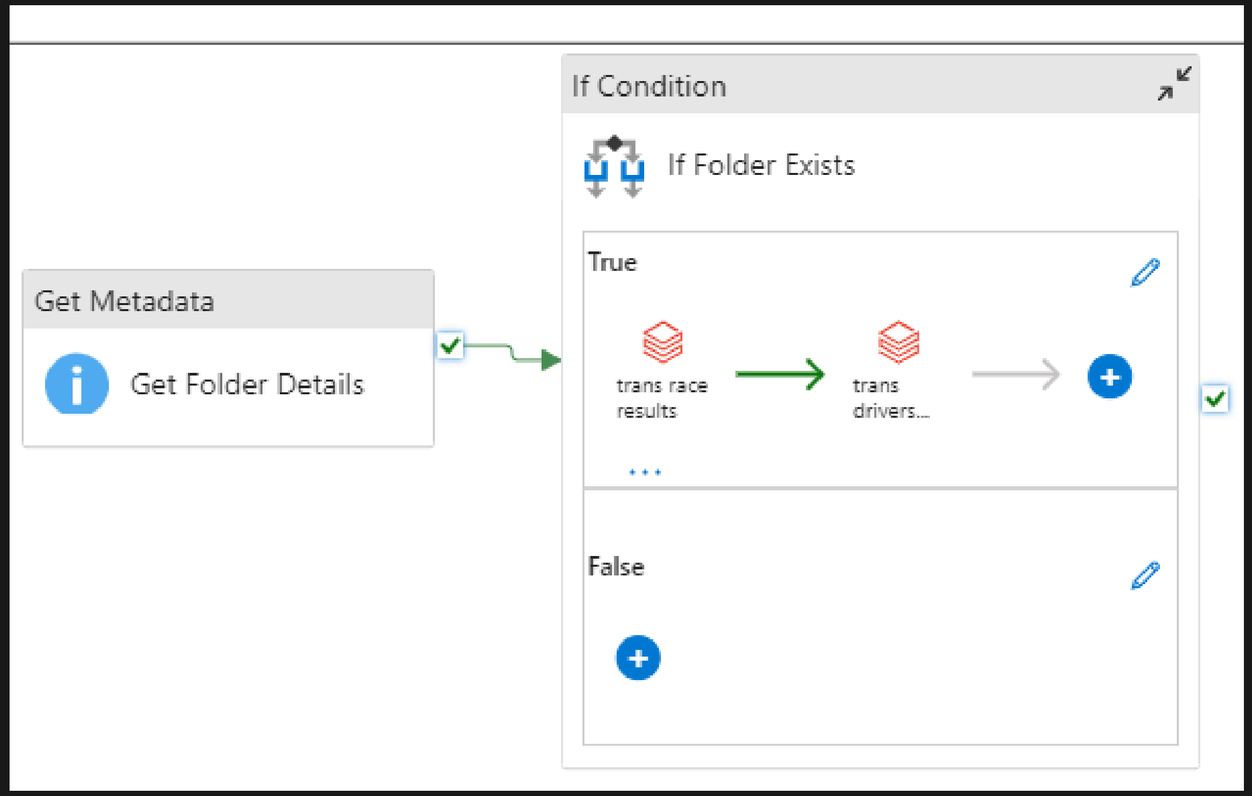
INGESTING RAW DATA WITH AZURE DATA FACTORY

- Used Get Metadata activity to verify folder existence in the raw zone
- Applied If Condition logic to trigger ingestion only when new data is present
- Ingested multiple raw files (CSV, JSON) into Delta tables using Databricks notebooks
- Captured ingestion logic for datasets like circuits, races, constructors, and drivers



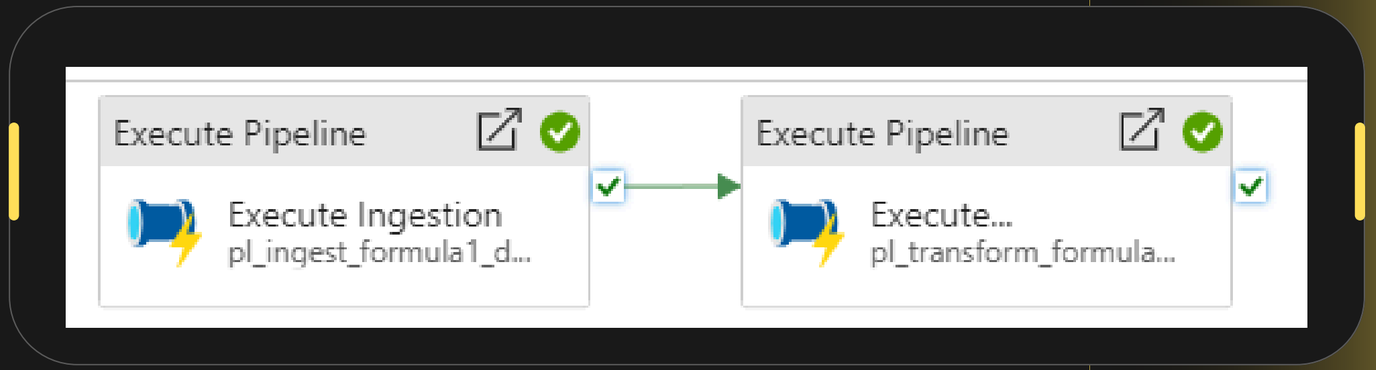
DATA TRANSFORMATION PIPELINE

- Joins across circuits, drivers, races, constructors, and results
- Renamed columns for clarity and standardization
- Added derived fields such as created_date and file_date
- Loaded final results into Delta Lake partitioned by race_year

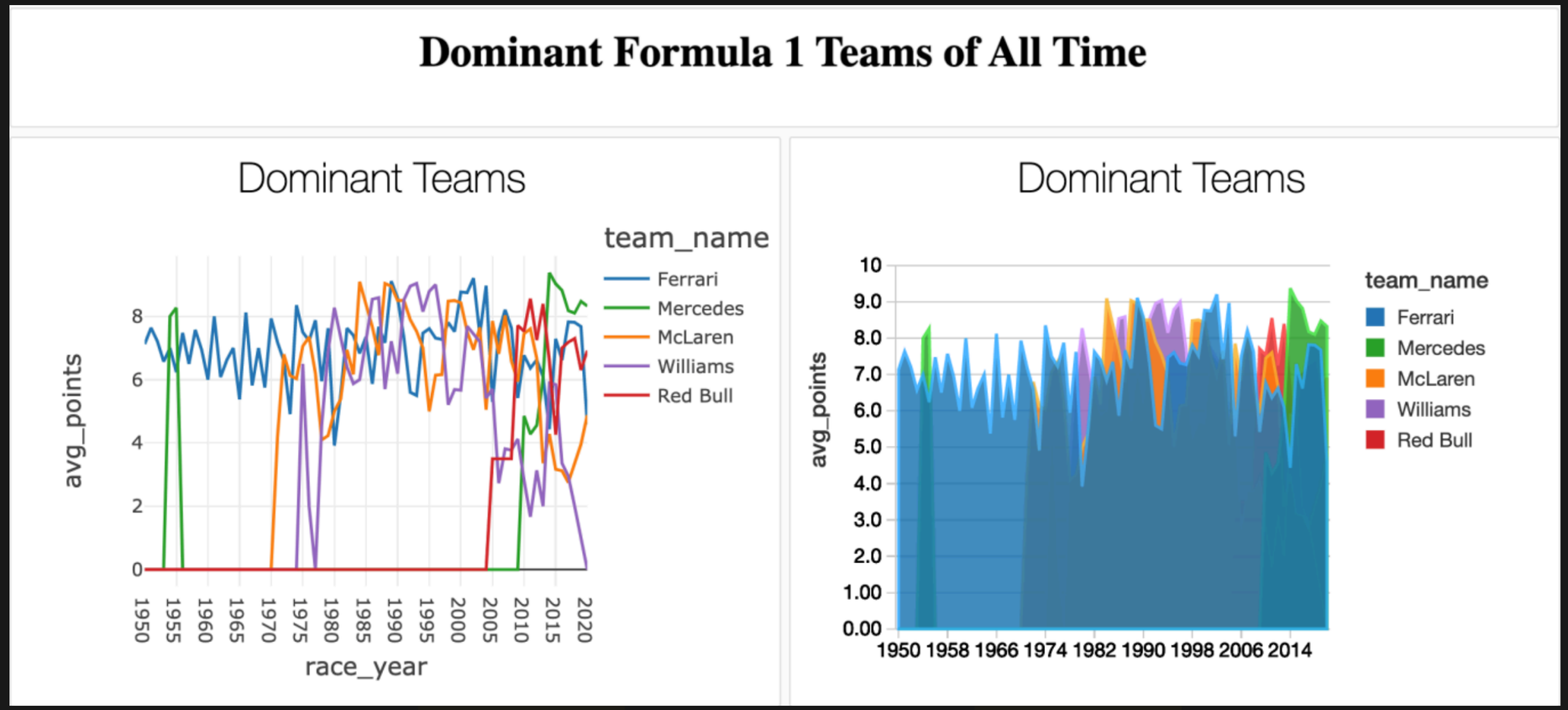


MASTER PIPELINE ORCHESTRATION

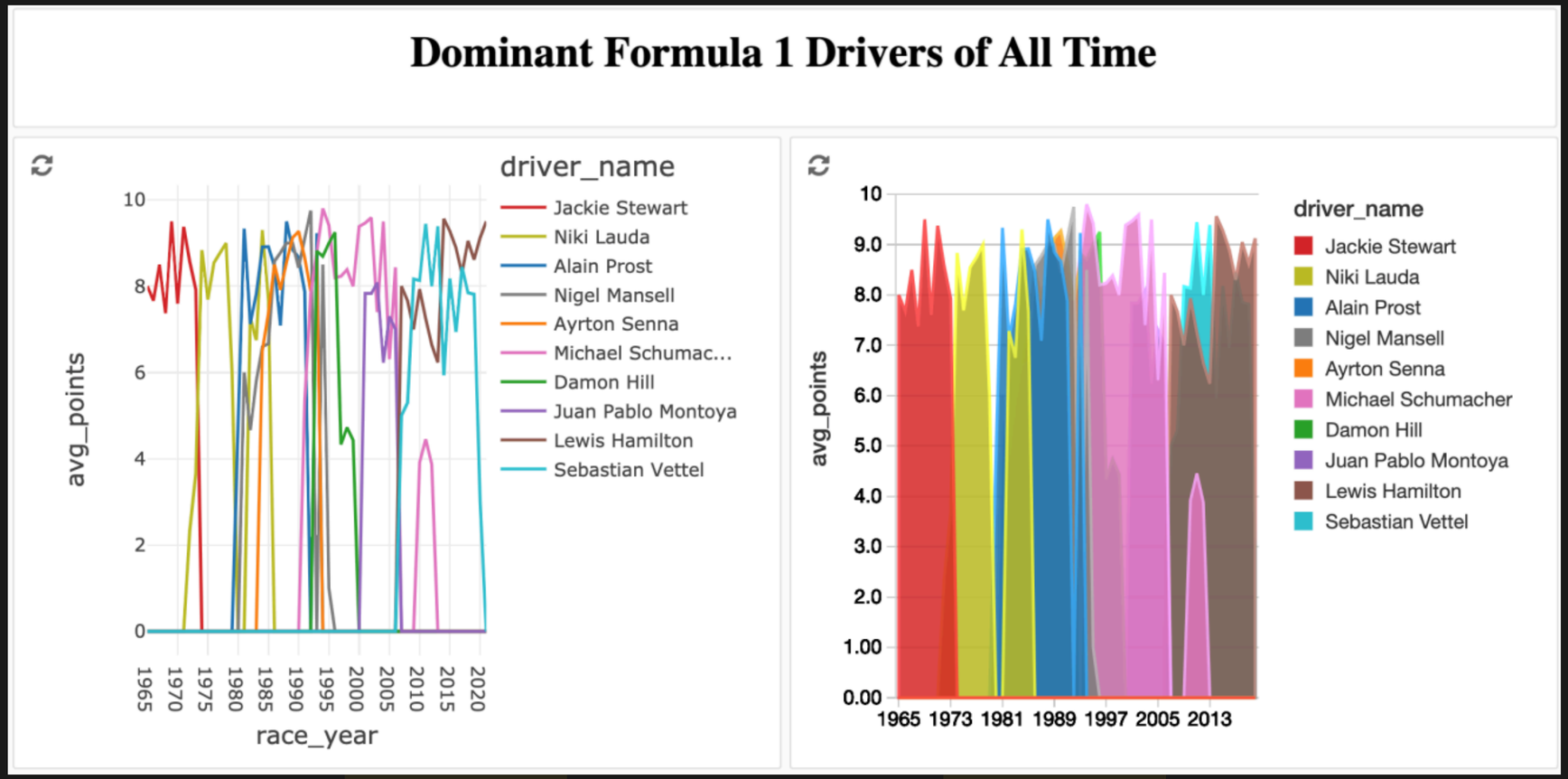
- Sequential execution using Execute Pipeline activities
- Ensures ingestion completes before transformation starts
- Includes pipeline-level error tracking and logging
- Easily extendable for future dependencies (e.g., validation, reporting)



VISUALIZE DATA WITH POWER BI



VISUALIZE DATA WITH POWER BI



Home About Us Solutions Industries Contact

THANK YOU

SITHSABA ZANTSI

Github

LinkedIn