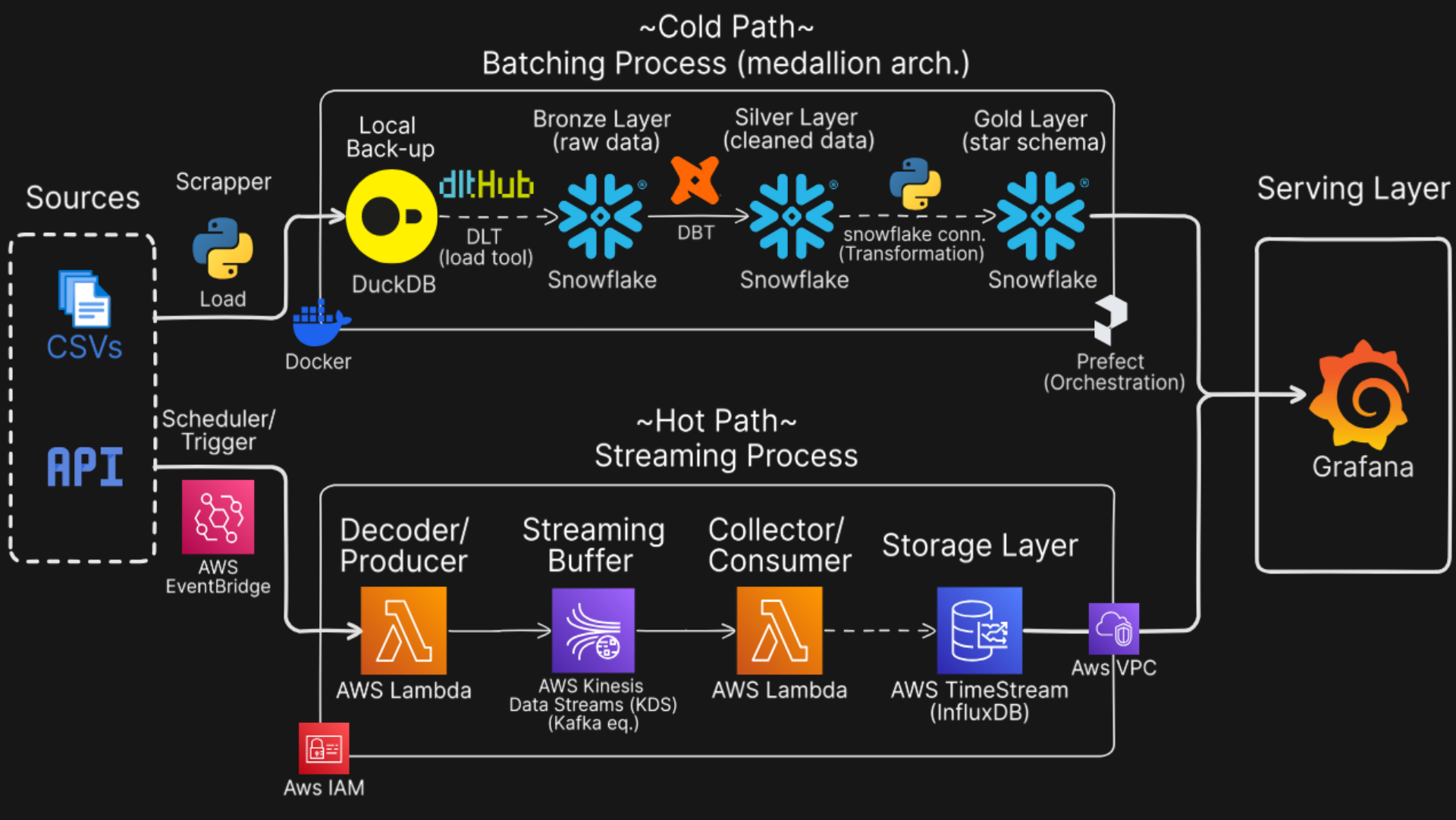

~Cold Path~
Batching Process (medallion arch.)
Local Back-up
Bronze Layer (raw data)
Silver Layer (cleaned data)
Gold Layer (star schema)
Sources
Scrapper
dltHub
DLT (load tool)
DBT
snowflake conn. (Transformation)
Serving Layer
DuckDB
Snowflake
Snowflake
Snowflake
CSVs
Load
Docker
Prefect (Orchestration)
Scheduler/ Trigger
~Hot Path~
Streaming Process
API
AWS EventBridge
Decoder/ Producer
Streaming Buffer
Collector/ Consumer
Storage Layer
Grafana
AWS Lambda
AWS Kinesis Data Streams (KDS) (Kafka eq.)
AWS Lambda
AWS TimeStream (InfluxDB)
Aws VPC
Aws IAM

Real-Time NYC Transit Monitoring & Analytics Platform

An End-to-End Data Pipeline using MTA Public API & Data

Presented on
December 18, 2025

AGENDA

01 Problem Statement & Motivation

02 Target Audience

03 Project Goals & Objectives

04 Data sources

05 Technical Architecture & Tech Stack

06 Implementation & Results

07 Insights & Recommendation

08 Q&A

OUR Team

AbdAelrahman Mostafa

Hussein Mohamed

Maher Mahmoud

Yousef Mahmoud

Moaaz omar Abdefattah

01
Problem Statement & Motivation

Challenge:

MTA data exists in two disconnected formats

- Static schedules: Text files (batch processing)
- Real-time updates: GTFS-Realtime Protobuf (streaming)

Impact:

- Cannot efficiently correlate 'planned' vs 'actual' service
- Protocol Buffers Decoding is not human-readable
- Limited operational visibility

Solution Needed:

data pipeline capable of seamlessly ingesting, decoding, and joining
These two fundamentally different data sources into a **unified**,
Low-latency record and **store** it for operational analytics

02

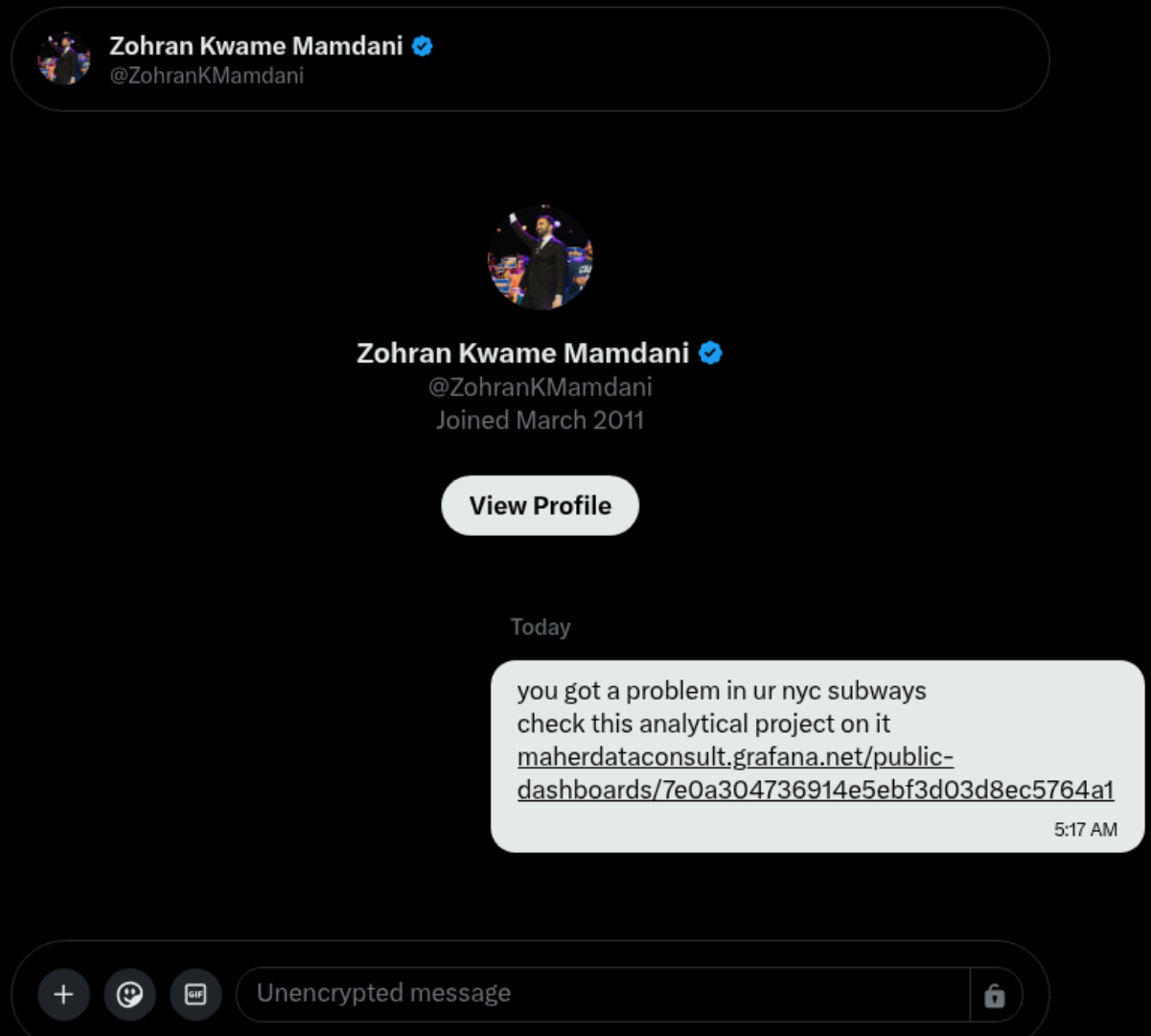
Target Audience

Teams within the Metropolitan Transportation Authority (MTA)

- **Transit Operations Staff:**
  - **Need real-time service health data**
  - **Requires immediate data to make proactive decisions**

- **Performance Analytics Teams:**
  - **Conduct retrospective analysis**
  - **identify bottlenecks**
  - **inform long-term planning**

NYC Mayor


Zohran Kwame Mamdani
@ZohranKMamdani
Zohran Kwame Mamdani
@ZohranKMamdani
Joined March 2011
View Profile
Today
you got a problem in ur nyc subways
check this analytical project on it
maherdataconsult.grafana.net/public-dashboards/7e0a304736914e5ebf3d03d8ec5764a1
5:17 AM
Unencrypted message

03

Project Goals

01 Operational Readiness

02 Provide a live dashboard

03 Data Reliability

Objectives

01 Batch Pipeline

02 Streaming Pipeline

03 Storage Layer

04 unified dashboard

04

Data sources

Batch Data

9 flat txt files
(csv format)

Realtime Data

Restful api
(Protobuf encoded)

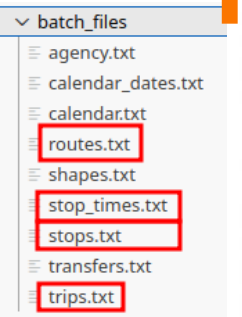
Sectors Covered

Subways Railroads Buses

Focused angle

- Routes
- Trips
- Stops
- Stop_times

- Delay measure



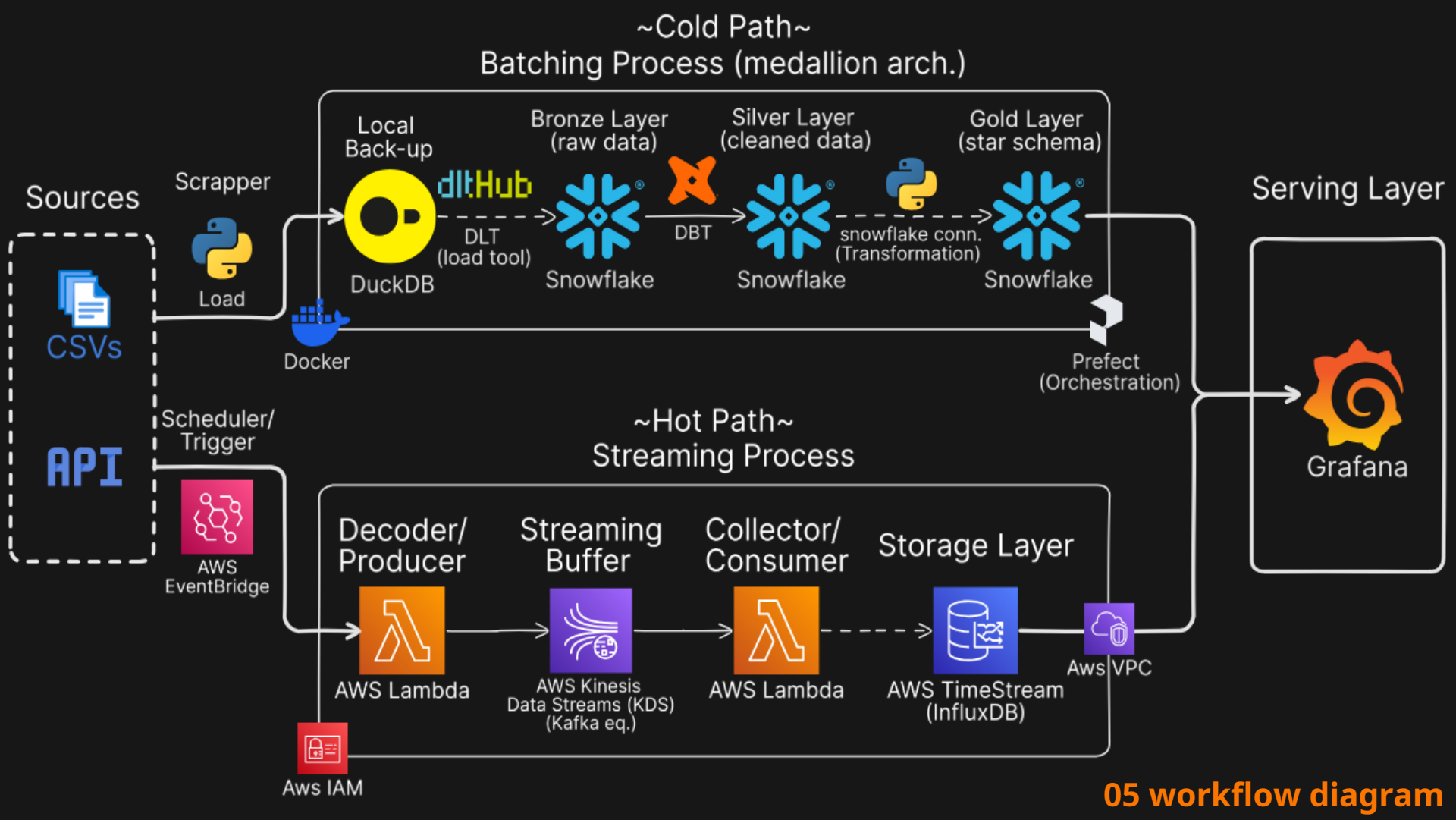
Examples of Analytical Qs

Static Analysis

- Total Routes / Stops / Trips
- Trips per Route comparison
- Identifies longest routes
- Stop frequency analysis
- Top Stations by Trip Volume

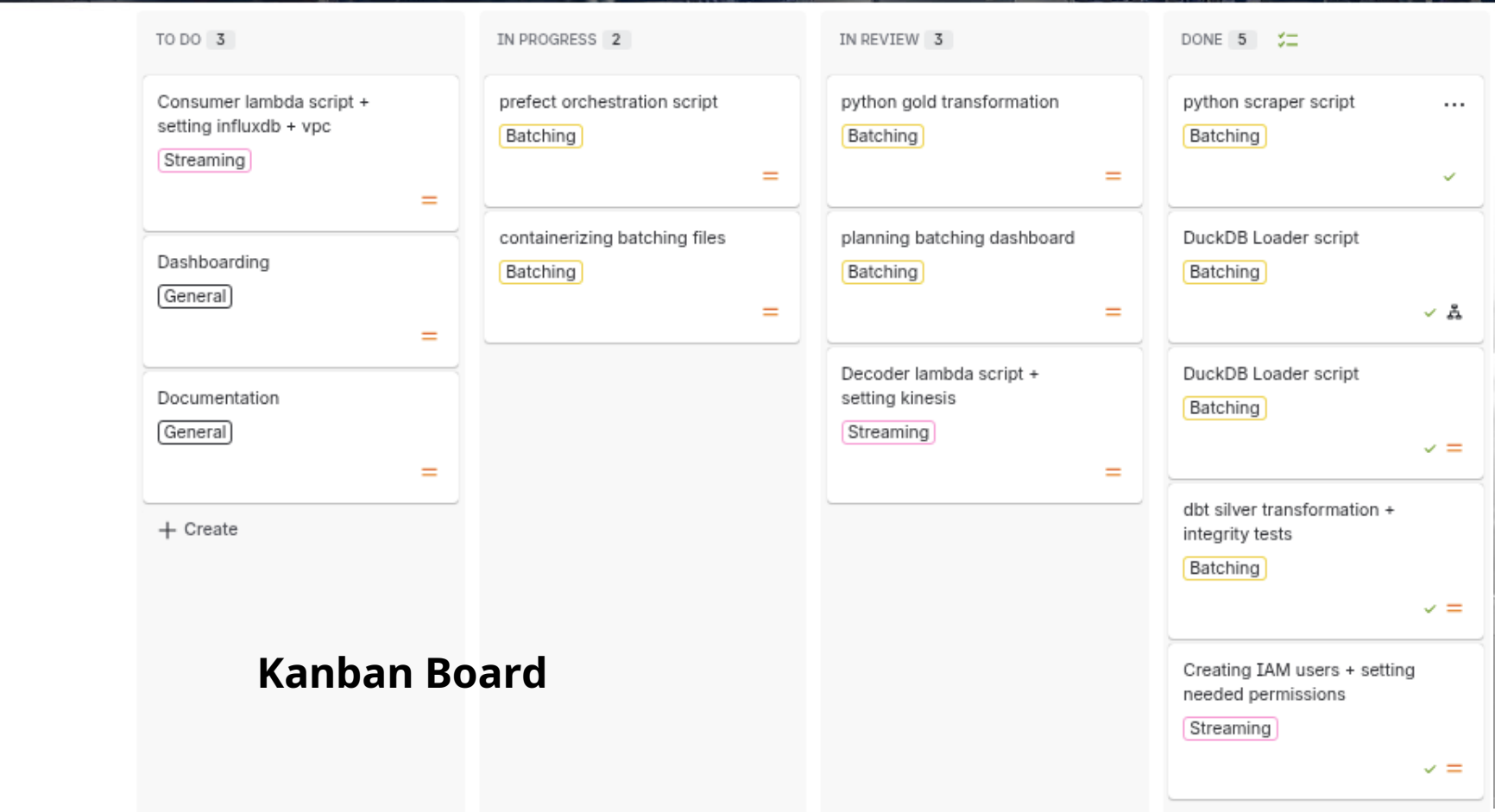
Real-Time Monitoring

- Active Data Points (Last 1 Hour)
- Avg Delay (Last 1 Hour)
- Active Alerts
- Delays Over Time by Route
- Top 10 Delayed Stations
- Route Performance (Delay/Route)



05 workflow diagram

Jira (Project Managment Tool)



Kanban Board

DBT Transformations (Silver layer)

1- Standardizing tables names into snake_case

2- setting the correct datatypes to columns like from bigint to timestamp

3- adding a calculated column(Stop Duration)

4- Integrity tests on all tables identifiers (pk)

5- Framework tests:

- relationships test for referential integrity

- setting composite pk to stop times table

6- output is in new schema named “staging”

```yaml
version: 2

models:
  - name: stg_agency
    columns:
      - name: agency_id
        tests: [unique, not_null]

  - name: stg_calendar
    columns:
      - name: service_id
        tests: [unique, not_null]
      - name: start_date
        tests: [not_null]

  - name: stg_routes
    columns:
      - name: route_id
        tests: [unique, not_null]
      - name: agency_id
        tests:
          - relationships:
              to: ref('stg_agency')
              field: agency_id
```

Python Transformations (Gold layer)

1- Removed tables that is unnecessary to the analysis

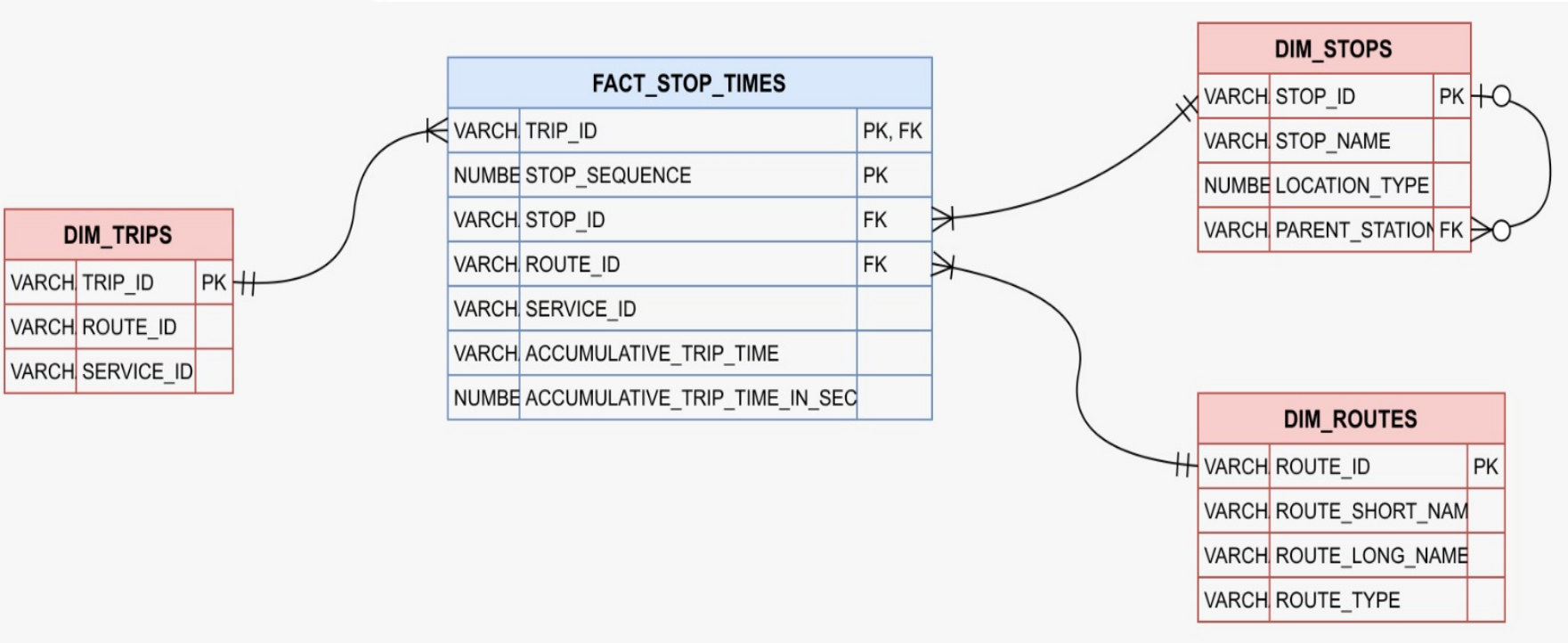
2- designing the star schema and implement it

3- adding a new calculated column “trip_duration”

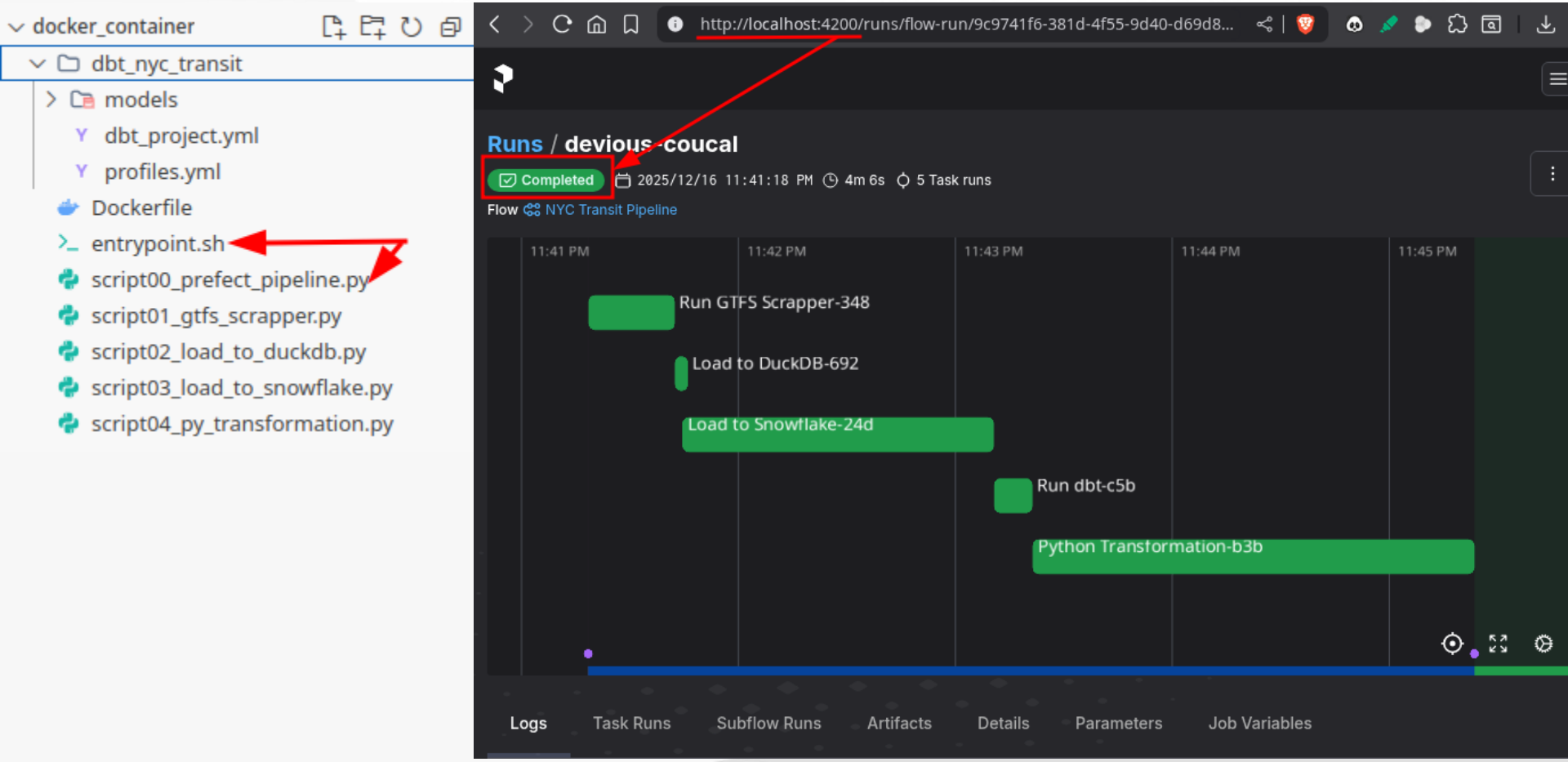
4- output is in new schema named “NYC_transit”

```python
import snowflake.connector
from snowflake.connector.pandas_tools import write_pandas
import pandas as pd

# Create connection
conn = snowflake.connector.connect(
    user="KILLSPORT13",
    password="@|EHSmK13DELAW",
    account="FYHTBNY-UZ80283",
    warehouse="COMPUTE_WH",
    database="NYC_TRANSIT",
    schema="STAGING"
)

# extract data
query = "SELECT * FROM STG_AGENCY"
df1=pd.read_sql(query, conn)
STG_AGENCY=df1.copy()

query = "SELECT * FROM STG_CALENDAR"
df2=pd.read_sql(query, conn)
STG_CALENDAR=df2.copy()

query = "SELECT * FROM STG_CALENDAR_DATES"
df3=pd.read_sql(query, conn)
STG_CALENDAR_DATES=df3.copy()
```

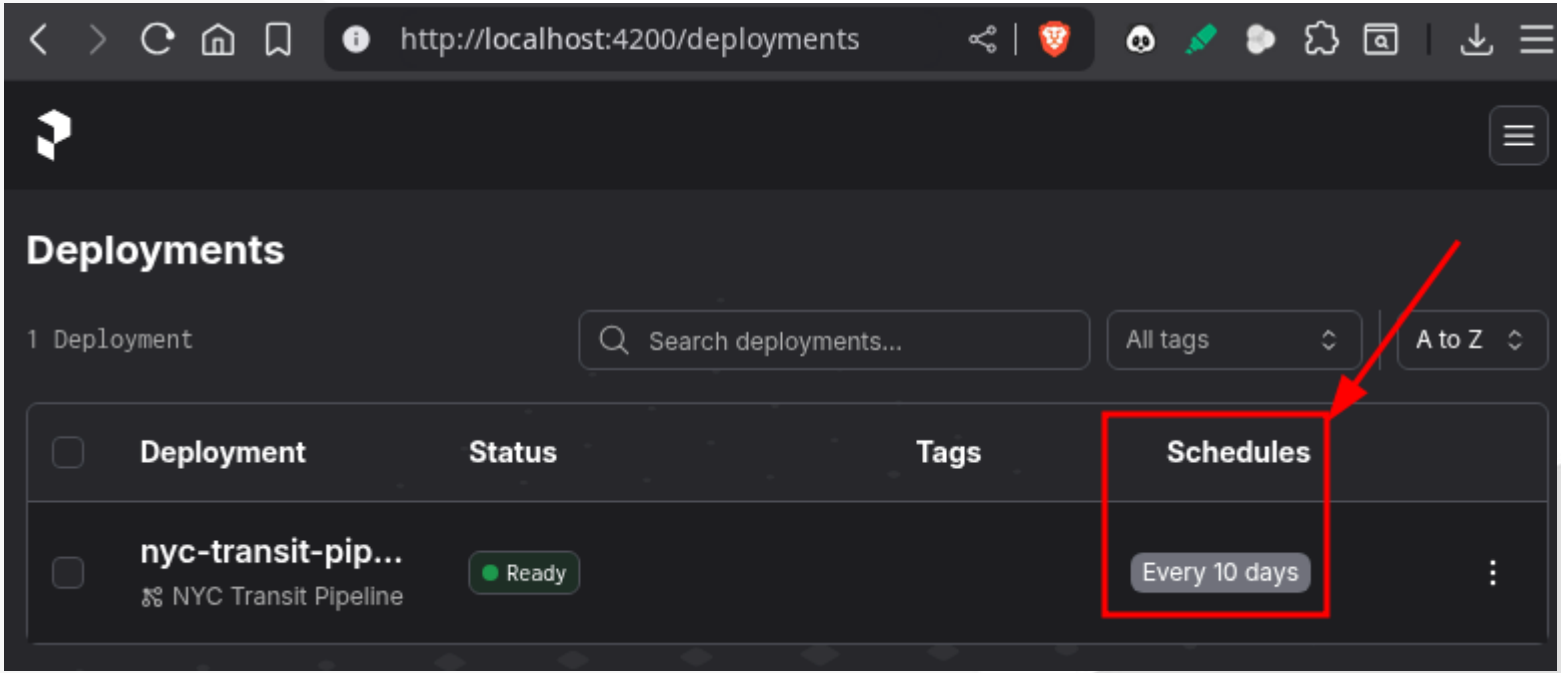
Star Schema (gold layer)



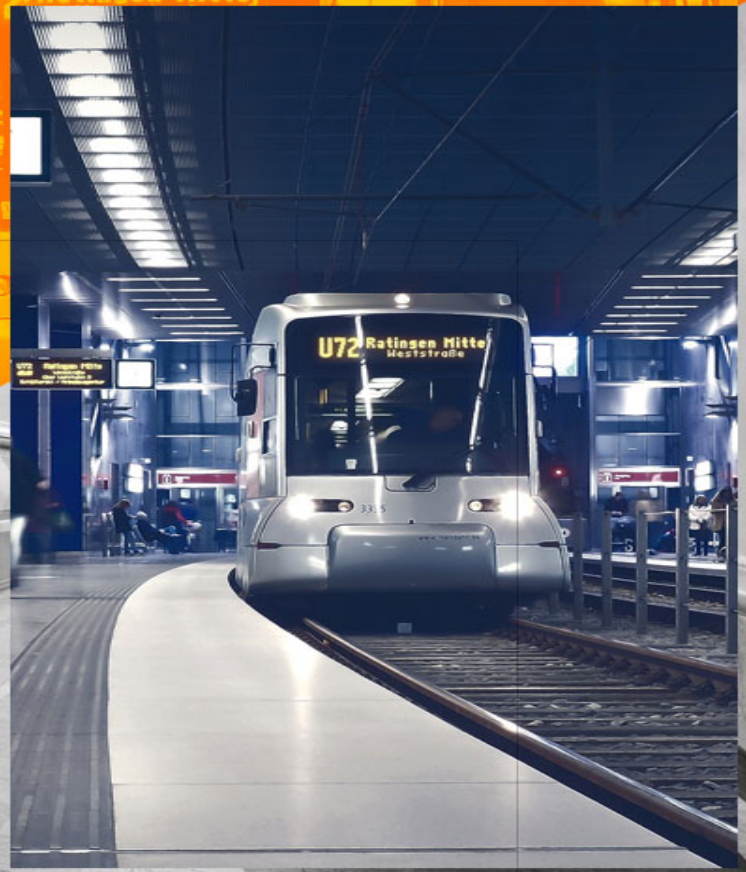
Containerization & Orchestration



Scheduler


http://localhost:4200/deployments
Deployments
1 Deployment
Search deployments...
All tags
A to Z
Deployment
Status
Tags
Schedules
nyc-transit-pip...
NYC Transit Pipeline
Ready
Every 10 days

Monitoring & Analysis (using Grafana)



Total Routes

29

Total Stops

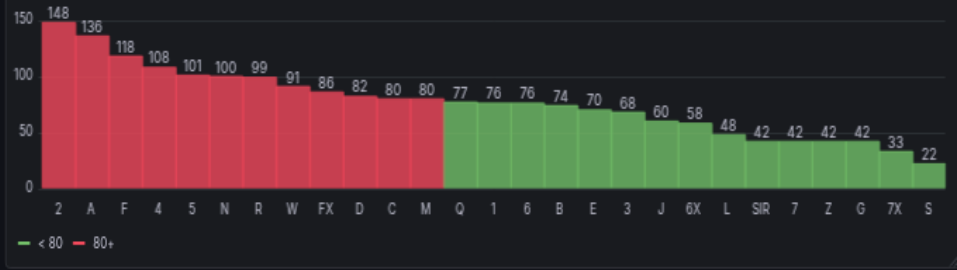
1488

Total Trips (per Week)

84384

How many trips per route?

How many stops per route?



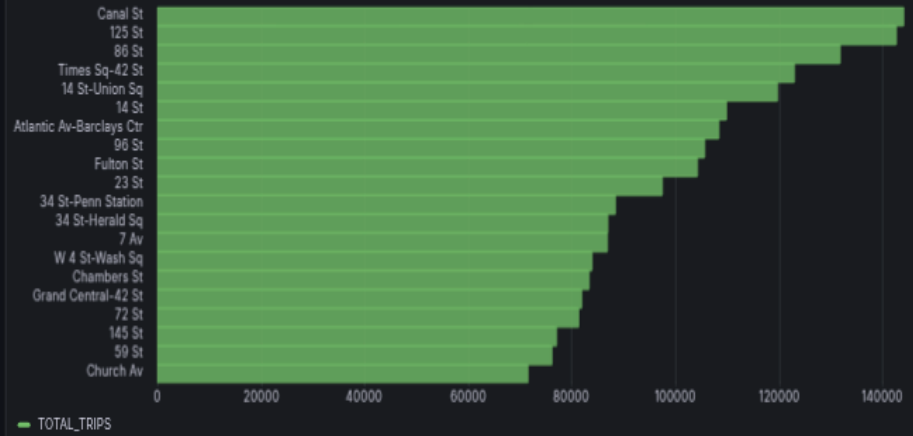
What is the average trip duration per route? (Longest Route)

ROUTE	AVG_TRIP_MINUTES ↓
F	104
D	103
2	99.6
FX	96.3

What is the time between stops for each route?

ROUTE	AVG_SECONDS_PER_STOP ↓
D	146
B	131
N	130
5	128

Which stations have the most trips passing through?



Last 1 hour

Active Data Points (Last 1 Hour)

1301

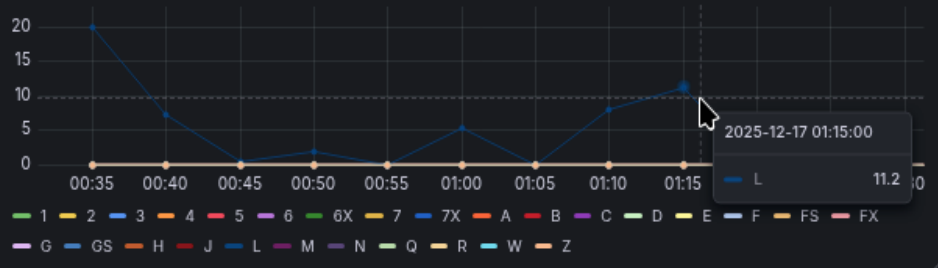
Avg Delay (Last 1 Hour)

0.139

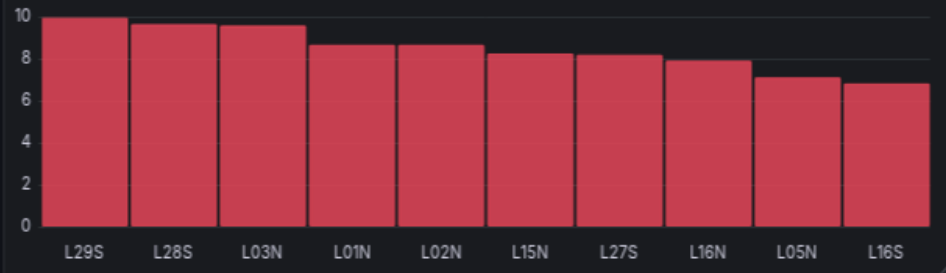
Active Alerts (Last 1 Hour)

322

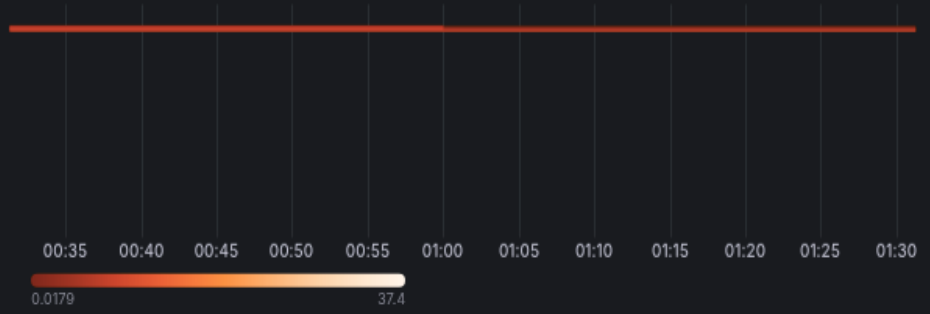
Delays Over Time by Route (Last 1 Hour)



Top 10 Delayed Stations (Last 1 Hour in Seconds)



Delay Heatmap over minutes (last 1 Hour)



Route Performance (Delay per Route last 1 Hout in seconds)

Route	Total_Delay_sec
L	11164
Z	0
W	0
R	0
Q	0
N	0

07
Insights

📊 Batch Analysis Insights:

- Network covers 29 routes with 1,488 stops
- 84,384 weekly trips indicate high utilization
- Route L handles 6,000+ trips (highest volume)
- Routes F, D, Z have longest durations (100+ minutes)
- Major hubs: Canal St, 125 St, Times Sq-42 St (140k+ trips)

⚡ Real-Time Monitoring Insights:

- Low average delay: 0.139 seconds (excellent performance)
- Route L shows 11,164 seconds total delay (needs attention)
- L29S, L28S, L03N stations are delay hotspots
- 322 active alerts indicate high alert volume

07
Recommendation

For Transit Operations:

- Focus resources on Route L (highest delays & volume)
- Investigate L29S, L28S, L03N stations for infrastructure issues
- Implement alert prioritization (322 alerts may cause fatigue)
- Monitor routes with 80+ stops for schedule optimization

For Performance Analytics:

- Correlate long trip durations (F, D, Z) with delay patterns
- Analyze peak hour patterns at top stations
- Study relationship between trip volume and delays

THANK YOU

Any Questions?

GitHub repo: [link]