Niall Langley

Data Engineer | Architect | Consultant

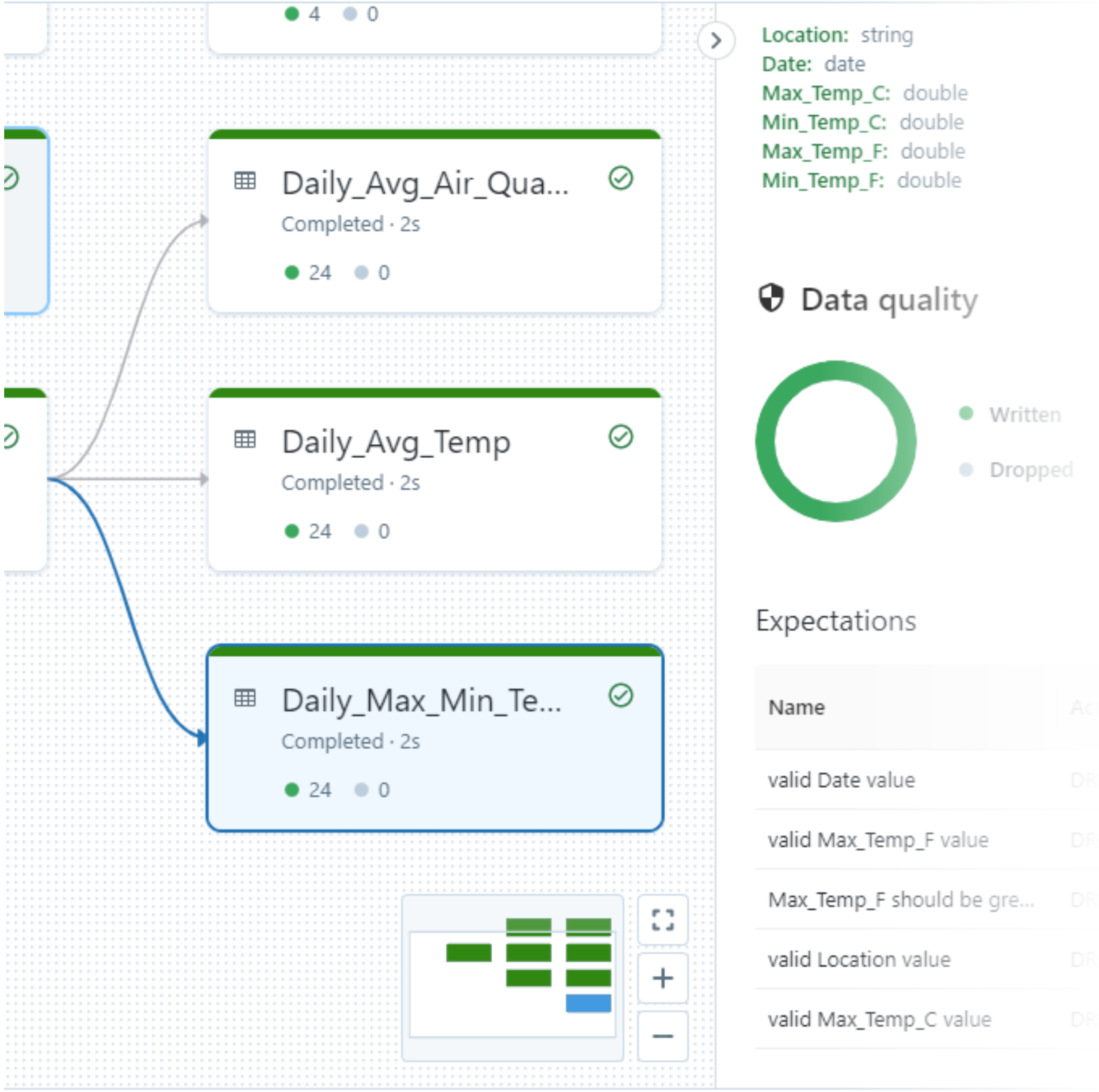
12 years experience working with the Microsoft Data Platform, now spends lots of time playing with Databricks, Spark and ADF in Azure

Blog:	www.sqlsmarts.com
LinkedIn:	uk.linkedin.com/in/niall-langley
Twitter:	@NiallLangley

Introduction to Databricks Delta Live Tables

Introduction

- What are Delta Live Tables
- Why would we use it
- What can it do
- How does it work
- What can catch you out



Location: string
Date: date
Max_Temp_C: double
Min_Temp_C: double
Max_Temp_F: double
Min_Temp_F: double

Data quality

Expectations

Name	Ac
valid Date value	DR
valid Max_Temp_F value	DR
Max_Temp_F should be gre...	DR
valid Location value	DR
valid Max_Temp_C value	DR

What are Delta Live Tables?

- An ETL framework available on Databricks
- Declarative, not procedural
- Stores files in Delta Lake format
- We can use Python or SQL Notebooks to build data pipelines
- Works out dependencies between ETL queries populating tables
- Built in data quality metrics
- Monitoring

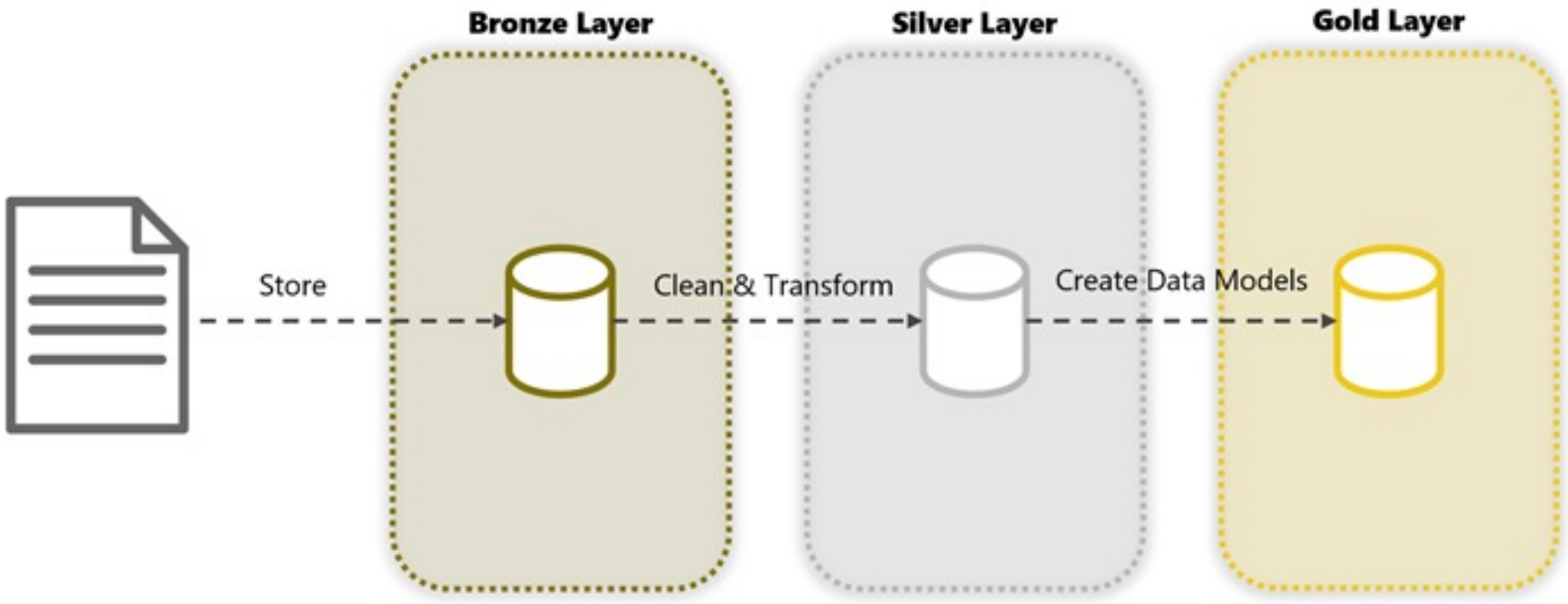
What are Delta Live Tables

- Databricks only – not on Synapse or vanilla spark
- Built on Spark Structured Streaming
- Uses Auto Loader to keep track of source files
- Logic is reusable across multiple pipelines
- Visualise dependencies between data entities

What is Delta Lake

- It’s Parquet
    - Columnar
    - Compressed
    - Open Source
- With a transaction log
- Supports time travel
- Allows for upserts to update and delete data
- Can delete historical records for performance and GDPR compliance

Delta Lake Architcture

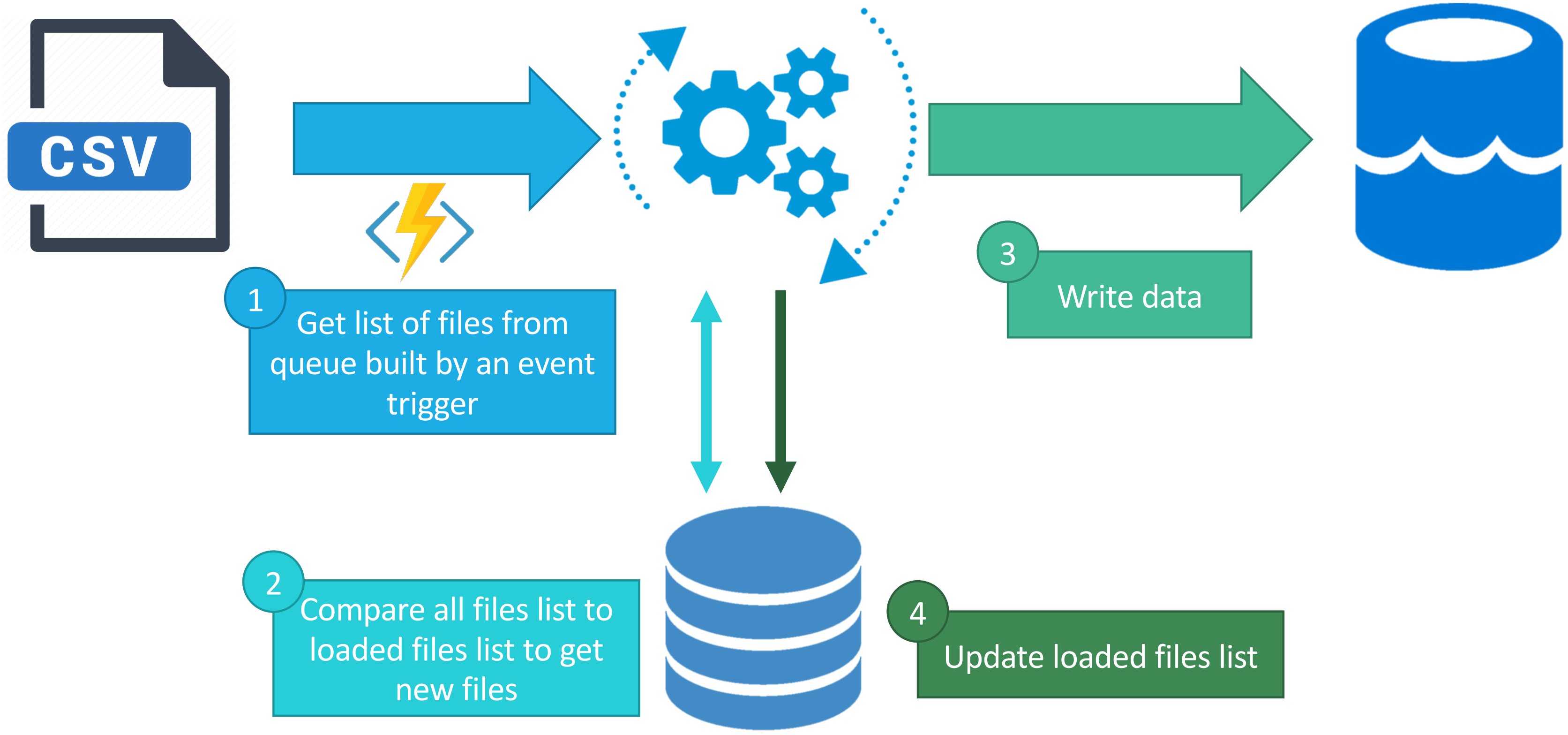


Incremental Loading

- Discovering which files to load is hard
  - Load all files exactly once
  - When/how to trigger
  - Discovery speed
  - Reusable Pattern
- DIY Options
  - Store last loaded file
  - Compare a list of previously loaded files to the list of files
  - Event subscriptions

Auto Loader

- Auto Loader is a framework to allow incremental loading of files



1 Get list of files from queue built by an event trigger

2 Compare all files list to loaded files list to get new files

3 Write data

4 Update loaded files list

Spark Structured Streaming

- Scalable & fault tolerant stream processing engine
- Allows you to write a streaming job that looks like a batch job
- Spark takes care of
  - Continuous incremental running
  - Loading data exactly once
  - Tracking state with checkpointing and logs
- Processes data using micro batches
- Keeps clusters alive!
- Has a trigger option called **Trigger.Once**

Building Pipelines

We build notebooks defining the queries to load tables

The notebook cannot be run on a normal cluster

A query defines the source, the transforms, and the target

If a live table is defined using the **streaming** keyword it is updated incrementally, otherwise it is fully recomputed each run

Views created in a DLT notebook are only available within the pipeline

Running Pipelines

- We create a DLT pipeline on the workflows tab in Databricks
- There are 3 editions to choose from
  - Core – Simple ingestion of data
  - Pro – Adds change data capture support (CDC) and updating target tables
  - Advanced – Adds the data profiling toolset
- The pipeline setting notebook libraries is where you select notebooks
- There is a debug and production mode
- You can run an incremental or full update, for either all tables or a selection of tables

Demo Time!

Conclusions

👍

- Good tool for ingestion and data quality checking
- Dependency resolution works well
- Less time building ELT frameworks
- Can switch to near real time ingestion

👎

- Hard to test and debug
- SQL is not flexible
- No source control for job definitions
- Python works best with… a framework!
- Need to know a bit about streaming

Tips & Tricks

- Can only set some workflow config using JSON, not the UI
  - Clusters
    - vCore type set using `driver_node_type_id` and `node_type_id`
    - Spark config, eg credentials for cloud storage from secrets
    - If using a policy to force a pool, you MUST explicitly set `driver_instance_pool_id` and `instance_pool_id`
- Policies can be used to set allowed cluster configs
  - Some settings can be inherited from a policy, eg spark config
- Joining datasets has some restrictions inherent from Spark Structured Streaming

Slides & Demos

https://github.com/NJLangley/Azure-Demos

Other Resources

https://learn.microsoft.com/en-us/azure/databricks/workflows/delta-live-tables/delta-live-tables-cookbook

https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html

Simon Whiteley on YouTube