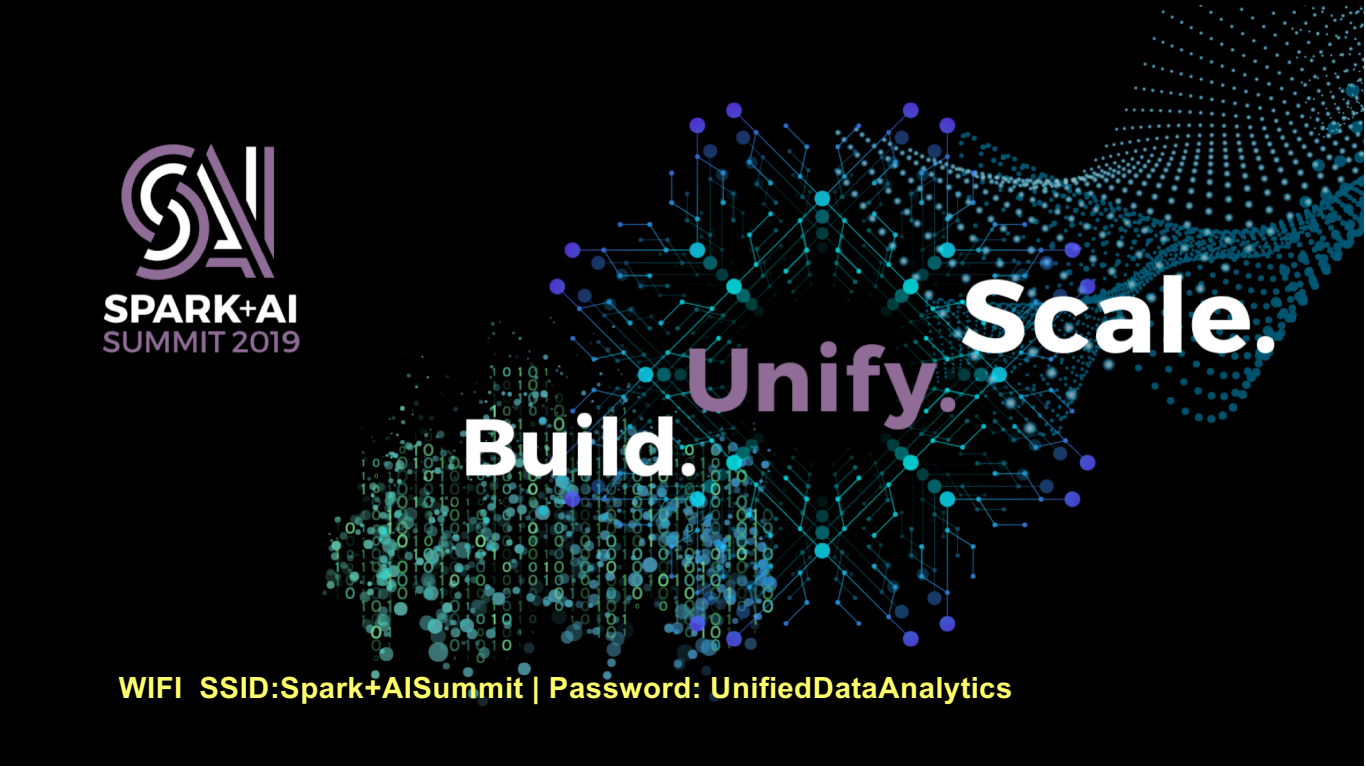

SPARK+AI
SUMMIT 2019
Build.
Unify.
Scale.
WIFI SSID:Spark+AISummit | Password: UnifiedDataAnalytics

Databricks Delta Lake And Its Benefits

Nagaraj Sengodan, HCL Technologies
Nitin Raj Soundararajan, Cognizant Worldwide Limited

#UnifiedDataAnalytics #SparkAISummit

Agenda

1. *The Brief*
   What is Delta lake offering from Databricks and overview
2. *Open Source*
   Good news is Delta Lake is open source now!
3. *Benefits*
   Why I care about Delta lake?
4. *Modern Warehouse*
   We do the necessary steps to deliver the result.
5. *Conclusion*
   Are we ready for building unified data platform?

Who are we ?

NAGARAJ SENGODAN

Senior Manager – Data and Analytics

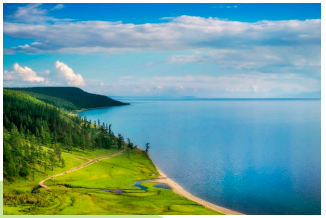
HCL Technologies

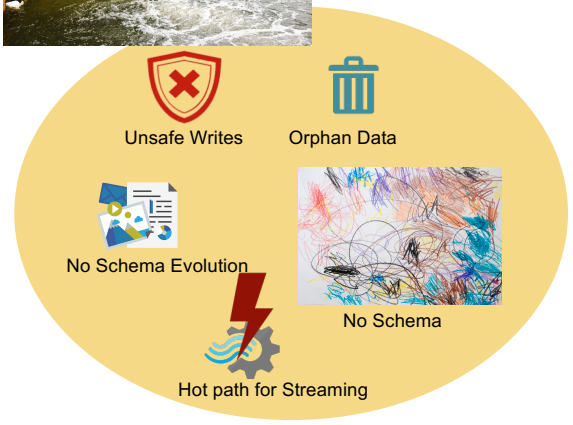
NITIN RAJ SOUNDARARAJAN

Senior Consultant – Data, AI and Analytics

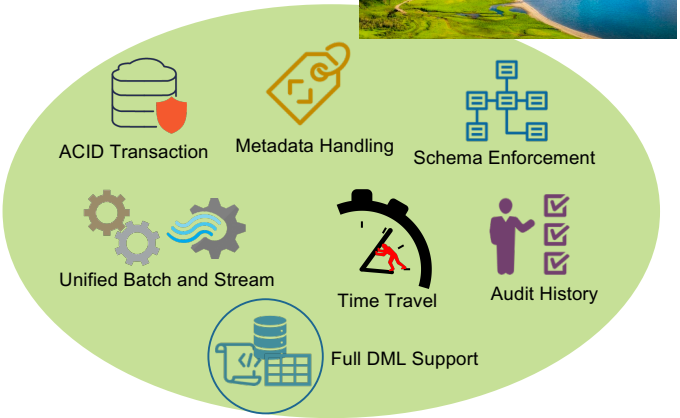
Cognizant Worldwide Limited

Common Challenges with Data Lakes



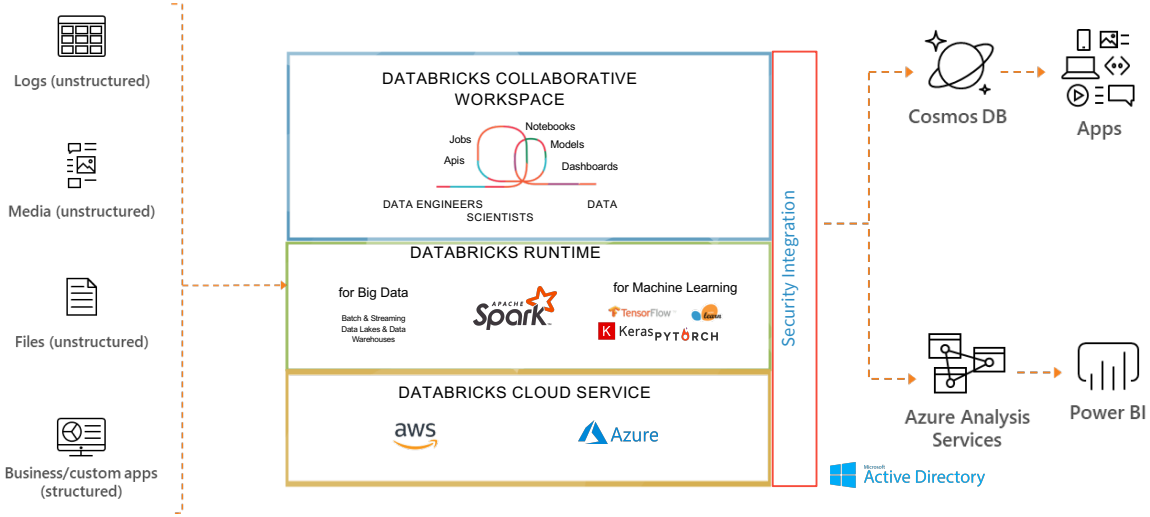


Data Lake getting Polluted

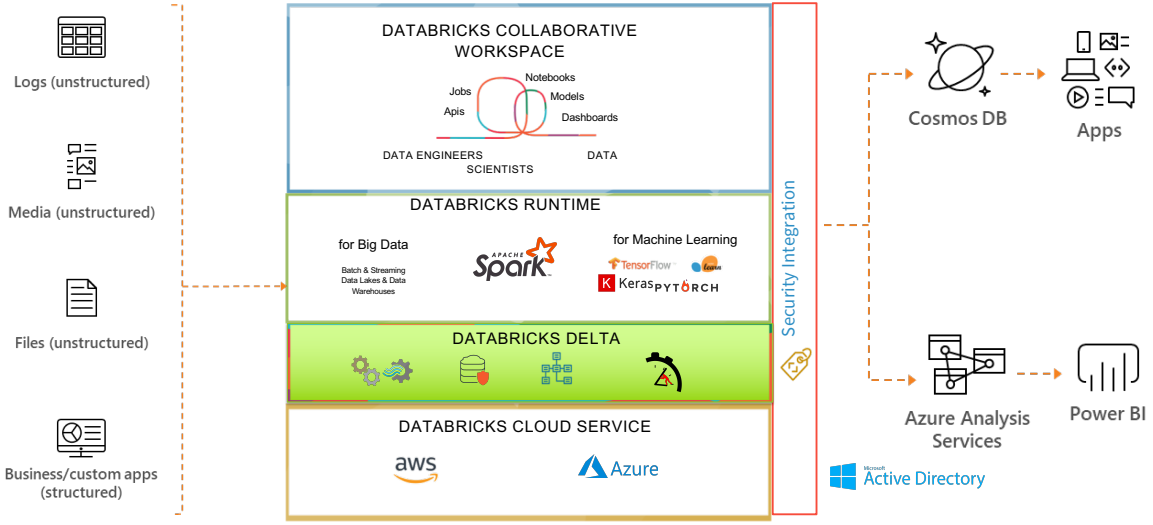


DELTA LAKE

Typical Databricks Architecture



Databricks - Delta Lake Architecture



The Brief

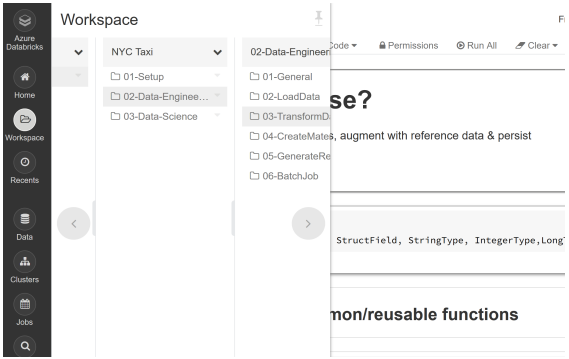
Delta.io – OPEN SOURCE.
APR. 2019
Announcing Delta Lake Open Source Project | Ali Ghodsi

Delta 0.2 – Cloud storage
JUN. 2019
Support for cloud storage (Amazon S3, Azure Blob Storage) and Improved Concurrency (Append-only writes ensuring serializability)

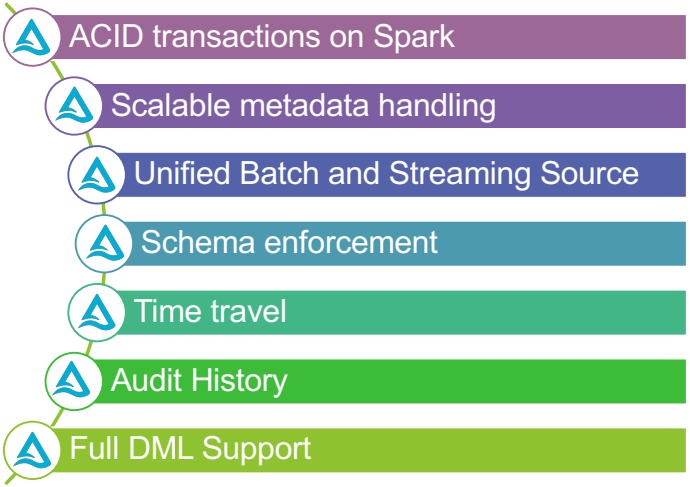
Delta 0.3 – Scala/Java API
AUG. 2019
Scala Java APIs and DML Commands, Query Commit History and vacuuming old files

Delta 0.4 – Python APIs and Convert to Delta
OCT. 2019
Python APIs for DML and utility operations, Convert-to-Delta, SQL for Utility Operations

Demo


Workspace
Azure Databricks
Home
Workspace
Recents
Data
Clusters
Jobs
NYC Taxi
01-Setup
02-Data-Enginee…
03-Data-Science
02-Data-Engineer
01-General
02-LoadData
03-TransformD
04-CreateMate
05-GenerateRe
06-BatchJob
Permissions
Run All
Clear
se?
, augment with reference data & persist
StructField, StringType, IntegerType,Long
mon/reusable functions

Benefits


ACID transactions on Spark
Scalable metadata handling
Unified Batch and Streaming Source
Schema enforcement
Time travel
Audit History
Full DML Support

ACID Transaction on Spark

01	*Multiple Writes*	Every write is a transaction Serial order for writes recorded in a transaction log
02	*Optimistic concurrency*	Multiple writes trying to modify the same files don't happen that often
03	*Serializable isolation level*	Continuously keep writing to a directory or table and consumers to keep reading from the same directory or table

Scalable Metadata Handling

01	*Metadata in Transaction Log*	Metadata information of a table or directory in the transaction log instead of the metastore
02	*Efficient Data Read*	Delta Lake can list files in large directories in constant time

Unified Batch and Streaming Sink

01	*Streaming Link*	Efficient streaming sink with Apache Spark's structured streaming
02	*Near Real Time Analytics*	with ACID transactions and scalable metadata handling, the efficient streaming sink now enables lot of near real-time analytics use cases without having to maintain a complicated streaming and batch pipeline

Schema enforcement

01	*Automatic Schema Validation*	Automatically validates the DataFrame's schema with schema of the table
02	*Column Validation*	Columns that are present in the table but not in the DataFrame are set to null Exception is thrown when when extra column present in the DataFrame but not in Table
03	*Serializable isolation level*	Delta Lake has DDL to explicitly add new columns Ability to update the schema automatically

Time Travel and Data Versioning

01	*Snapshots*	Allows users to read a previous snapshot of the table or directory
02	*Versioning*	Newer version of the files are created when the files are modified during writes and older versions are preserved
03	*Timestamp and Transaction Log*	Provide a timestamp or a version number to Apache Spark's read APIs to read the older version of the table or directory Delta Lake constructs the full snapshot as of that timestamp or version based on the information in the transaction log User can reproduce experiments and reports and also can revert a table to its older versions

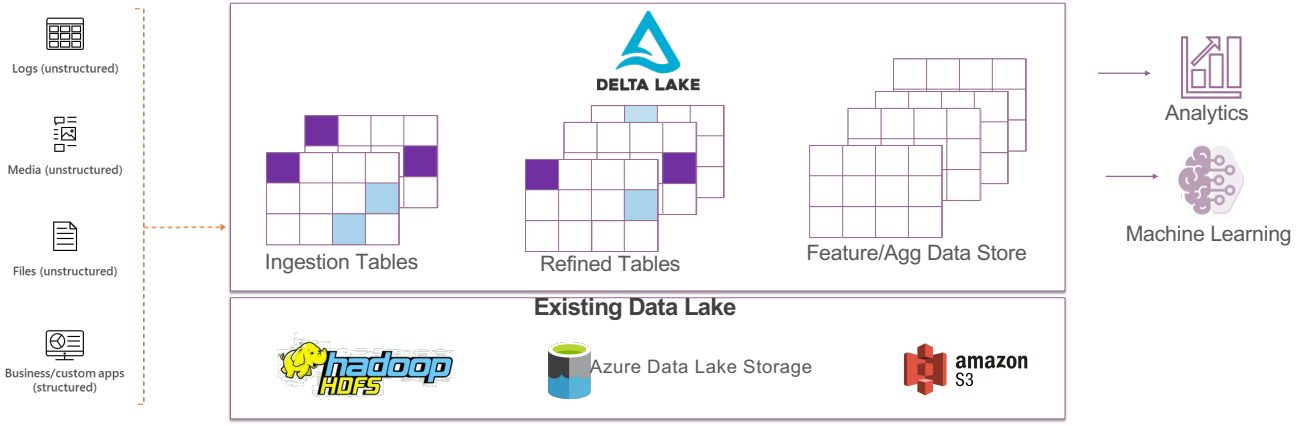
Record Update and Deletion (Coming Soon)

01	*Merge* *Update* *Delete*	Will support Merge, Update and Delete Easily upsert and delete records in data lakes simplify their change data capture and GDPR use cases
02	*File-level* *granularity*	More efficient than reading and overwriting entire partitions or tables

Data Expectations (Coming Soon)

01	*API to set data expectations*	Will support an API to set expectations on tables or directories
02	*Severity to handle expectations*	Engineers will be able to specify a boolean condition and tune the severity to handle data expectations

Modern Data warehouse



Conclusion

	Delta Lake	Data Lake
Reliable	**High** As it enforce schema and ACID operations helps data lake more reliable	**Less** Accept all data and late binding leads lot of orphan data
Unification	**High** Batch and Stream data set can be processed in same pipeline	**Medium** Stream process require hot pipeline
Performance	**High** Z-Order skipping files for efficient read	**Medium** Sequence read
Ease of Use	**Medium** Delta require DBA operations like Vacuum and Optimize	**High** No write on schema and accept any data

Like to know more?

https://github.com/KRSNagaraj/**SparkSummit2019**

https://www.linkedin.com/in/**NagarajSengodan**

https://www.linkedin.com/in/**NitinRajS**/

SPARK+AI SUMMIT 2019

DON'T FORGET TO RATE AND REVIEW THE SESSIONS

SEARCH SPARK + AI SUMMIT

Download on the App Store

GET IT ON Google Play