Building an open source data lake at scale in the cloud

Adrian Woodhead, Principal Engineer
Agenda

Background
Data Lake foundation: data + metadata
High Availability and Disaster Recovery
Data federation
Event-based data processing
Data Lake journey

- “traditional” RDBMS Data Warehouse
- Introduced on-premise Hadoop + Hive cluster
- RDBMS SQL replaced by SQL from Hive
- Slow at busy times
- Painful upgrade path (software and hardware)
- Migration to “Cloud” as primary data lake
Cloud Data Lake High Availability
Cloud Data Lake Redundancy
Redundancy by replication

- Data and Metadata
- Co-ordinated
- Data consistency during replication
- No partial reads
- Completeness more important than latency
Circus Train – Hive dataset replicator

- [https://github.com/HotelsDotCom/circus-train/](https://github.com/HotelsDotCom/circus-train/)
- Metadata only available after data
- Supports HDFS, S3, GCS etc.
- Standard “distcp” and optimised copiers
- Plugin architecture – Notifications, Copiers, Metadata transformations
- Selective data replication – custom filters, “Hive Diff”
- [https://github.com/HotelsDotCom/shunting-yard](https://github.com/HotelsDotCom/shunting-yard)
  - Event-driven Circus Train
Data Lake Silos
Data Lake Silo Solutions

• Move back to a single data lake
  • Scalability issues
  • Increased “blast radius”
• Replicate shared data sets between data lakes
  • Cost of maintaining replication jobs
  • Increased file storage costs
  • Increased network transfer costs
Federated Cloud Data Lake

- [https://github.com/HotelsDotCom/waggle-dance/](https://github.com/HotelsDotCom/waggle-dance/)
- Waggle Dance – a Hive Thrift metastore proxy
- Configure it with “downstream” Hive metastores
- Configure S3 bucket access permissions
- Set “hive.metastore.uris” to Waggle Dance server
- Use as you would Hive metastore in any client app
Waggle Dance Overview

- Federated Hive instance
- Metastore queries
- Virtual database
- Proxy database
- Concrete database
- Mappings
- Read/Write
- Read only

- Waggle-Dance Service
- Primary MS
- External MS X
- External MS Y
Multi-Region Federated Cloud Data Lake

Federate

Replicate

US_WEST_2

US_EAST_1

Replicate

US_WEST_2

US_EAST_1
Federated Cloud Data Lake Best Practices

- Expose read-only endpoints to “external” users
- Separate critical path infrastructure
- Federate data for access within a region
- Replicate data for access in a different region
Federated Cloud Data Lake Alternative

• Presto – distributed SQL query engine for big data
• Federate Hive, MySQL, PostgreSQL and many others

• [https://github.com/prestodb/presto](https://github.com/prestodb/presto)
  OR
• [https://github.com/prestosql/presto](https://github.com/prestosql/presto)
Apiary - Cloud Data Lake Components

- https://github.com/ExpediaGroup/apiary
- Various components for a federated cloud data lake
- Docker images for all services
- Terraform deployment scripts
- Ranger for authorization
- Various optional extensions
Apiary – Metadata Events

- Events for tables/partitions CRUD operations
- Hive MetaStoreEventListener implementations
  - Kafka
  - AWS SNS
- Enable downstream data processing use cases
  - ETL, Governance, Lineage etc
Problem – rewriting data at scale

• Changes to existing data
• Read isolation for long running queries
• Always create new folders for updates
• Repoint Hive data locations
• How to expire “orphaned data”? 
Beekeeper – orphaned data cleanup

- [https://github.com/ExpediaGroup/beekeeper/](https://github.com/ExpediaGroup/beekeeper/)
- Hive table parameter: `beekeeper.remove.unreferenced.data=true`
- Apiary event listener
- Detects data re-writes
- Schedules old data for deletion in future
- Periodically performs the data deletions
Consistent CRUD alternatives

- [https://iceberg.incubator.apache.org/](https://iceberg.incubator.apache.org/) - Iceberg
- [https://delta.io/](https://delta.io/) - Delta Lake
- [https://hudi.apache.org/](https://hudi.apache.org/) - Hudi
Don’t forget to test

- [https://github.com/klarna/HiveRunner/](https://github.com/klarna/HiveRunner/) - Hive SQL unit tests
- [https://github.com/HotelsDotCom/mutant-swarm/](https://github.com/HotelsDotCom/mutant-swarm/) - Code coverage for HiveRunner
- [https://github.com/HotelsDotCom/beeju](https://github.com/HotelsDotCom/beeju) - Unit tests for Thrift Hive metastore service and HiveServer2
Where to next?

• Hybrid cloud
  • best of both worlds but increased complexity
• Multi-cloud
  • best of breed but increased complexity
• Docker + Kubernetes
  • Reduce vendor lock-in
  • Massive scale without too much effort
  • Minimal changes for on-prem/EKS/GKE/AKS etc
Open Source Data Lake Components

Hive Replication
https://github.com/HotelsDotCom/circus-train
https://github.com/ExpediaGroup/shunting-yard

Hive Federation
https://github.com/HotelsDotCom/waggle-dance

Hive Cleanup
https://github.com/ExpediaGroup/beekeeper

Cloud Data Lake
https://github.com/ExpediaGroup/apiary