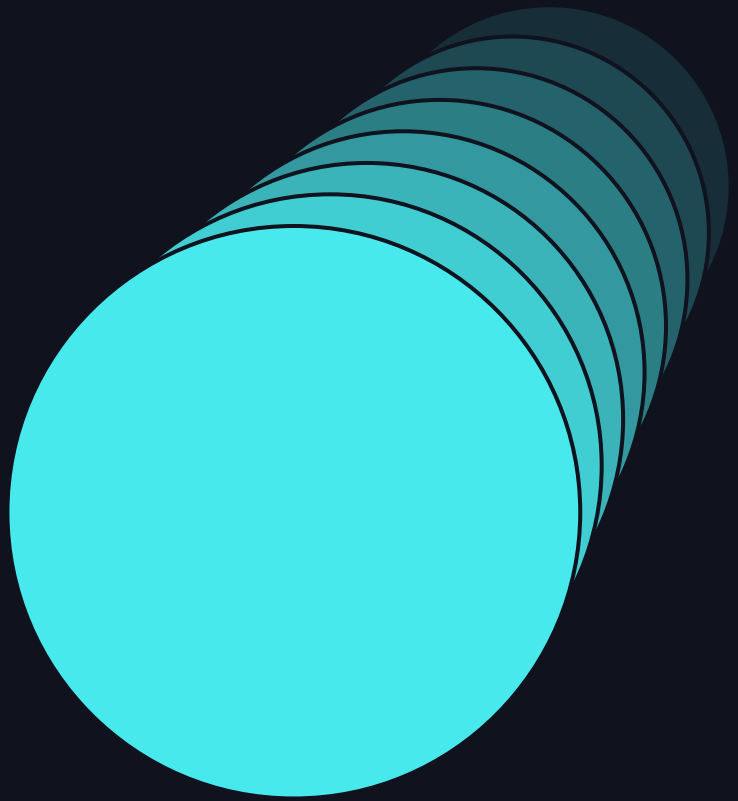
DATA+AI SUMMIT BY databricks

Migrating Udemy's Data Platform to Databricks



Nathan Sullins, *Sr. Data Engineering Manager,* ***Udemy***
Pravin Todkar, *Sr. Solutions Architect,* ***Databricks***
June 11, 2024

Udemy

Udemy has the world's largest professional skills marketplace: the engine behind our model.

75k

Instructors

210k

Marketplace courses

69M

Global learners

180+

Countries

75+

Languages

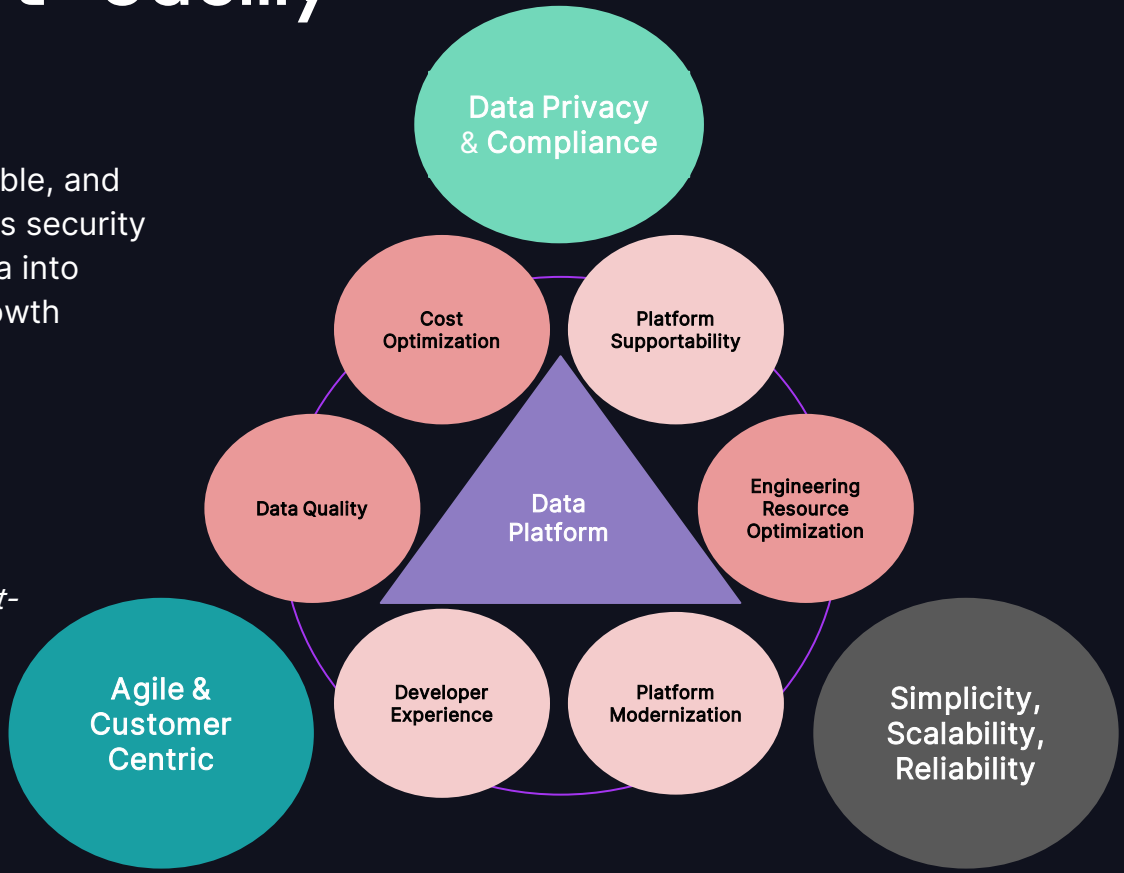
Data Platform at Udemy

The Vision

To empower business with a self-serve, scalable, and performance driven data platform that ensures security and leverages AI capabilities to transform data into actionable insights, driving innovation and growth

The Mission

- Increase *productivity and efficiency* of our data engineering, ML engineering, analytic engineering and data platform teams with a scalable, reliable, elastic, performant but *cost-optimized* platform with better supportability
- To make Udemy Data Lake and data warehouse *secure and compliant*
- Collaborate with multiple teams to have *trustworthy and reliable data quality*



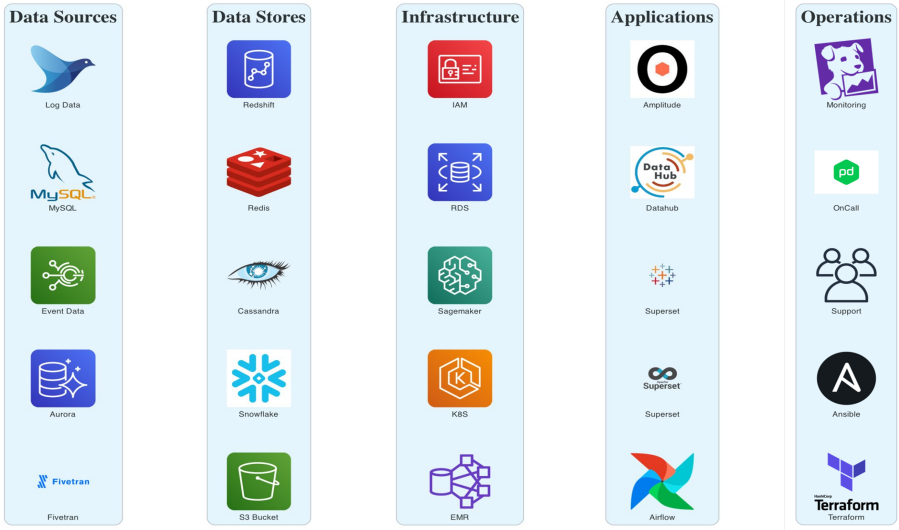
The Data Platform at Udemy before Databricks

Evolution of Udemy's Data Platform

Early focus was on costs (minimizing resources, colocation) with a shift to stability, scalability and functionality provided by managed services.

2022 data platform:

- Disjointed
- Overly complex
- Difficult to use
- Competing toolsets
- Difficult to maintain

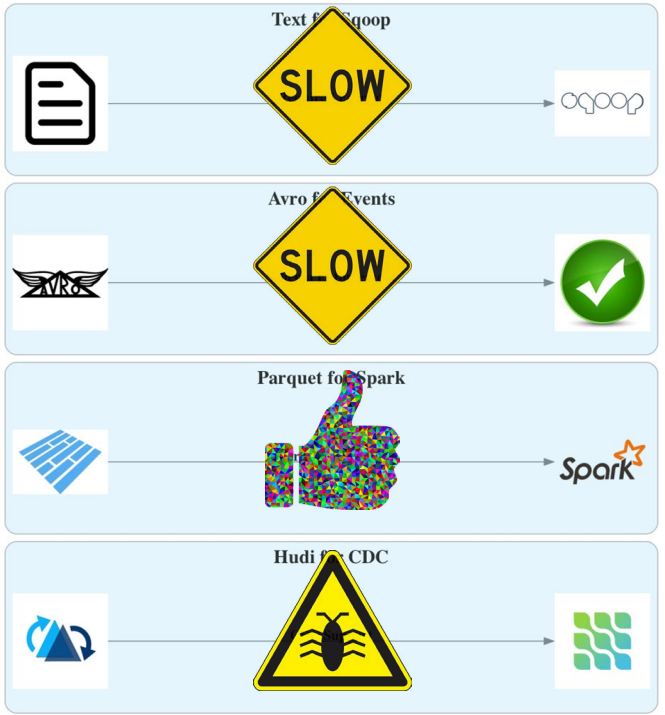


Multiple data formats

Lacking a one size fits all format resulted in slow queries and incomplete data.

Formats old and new

- Text required for Sqoop
- Avro for schema enforcement
- Parquet
- Hudi for CDC



Stuck in the past with Hadoop

From on-prem Hadoop to EMR 5 was great but we lacked ephemeral compute.

EMR 5.31 Pre-serverless

- Running M/R Services
- Noisy Neighbors
- Blue/Green Deployments
- No granular tracking of costs
- EMR upgrades were painful

Data access: Too much, not enough

Managing multiple metastores is a challenge and confusing to users.

Too much complexity

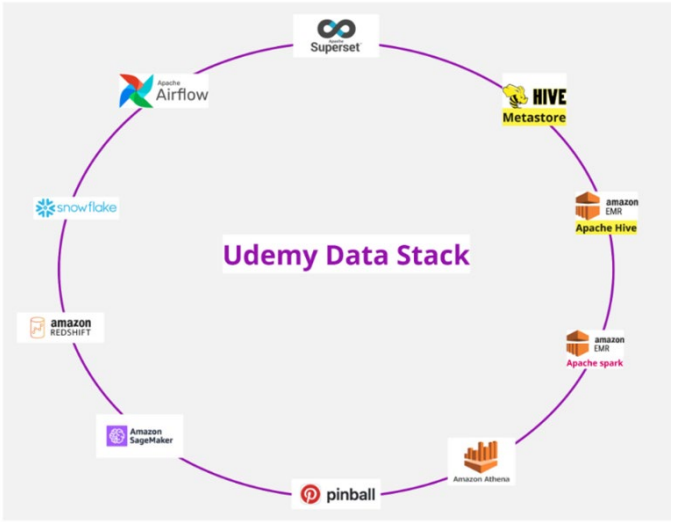
- Multiple Hive Metastores
- Syncing Metastores overhead
- Data users confusion

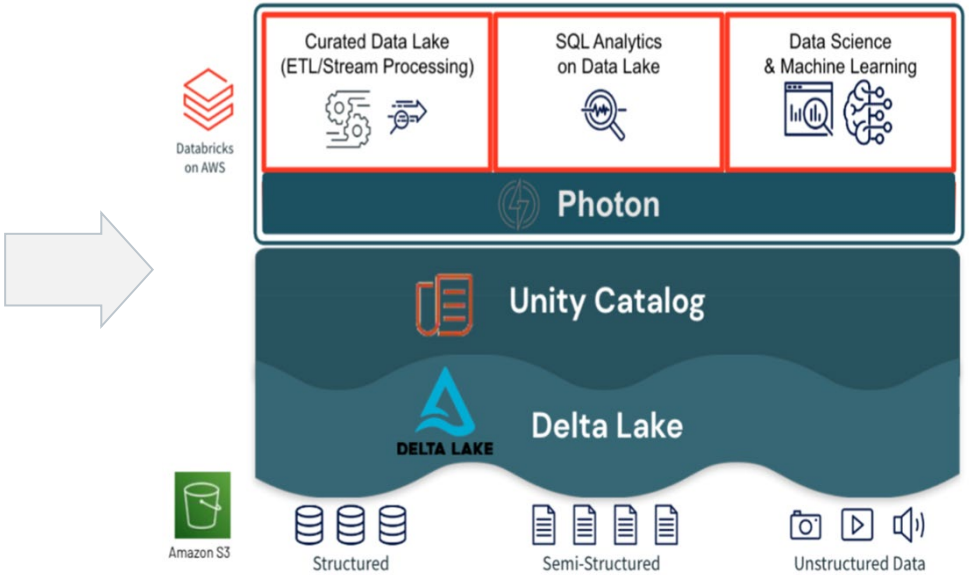
Not enough functionality

- Incomplete RBAC
  - Schema/Table/Column
- Over-permissioning

Unified Data Platform w/ Databricks

A vision for the future emerges.





The Experiment

Is Databricks right for us

What questions should we be asking? What are our concerns?

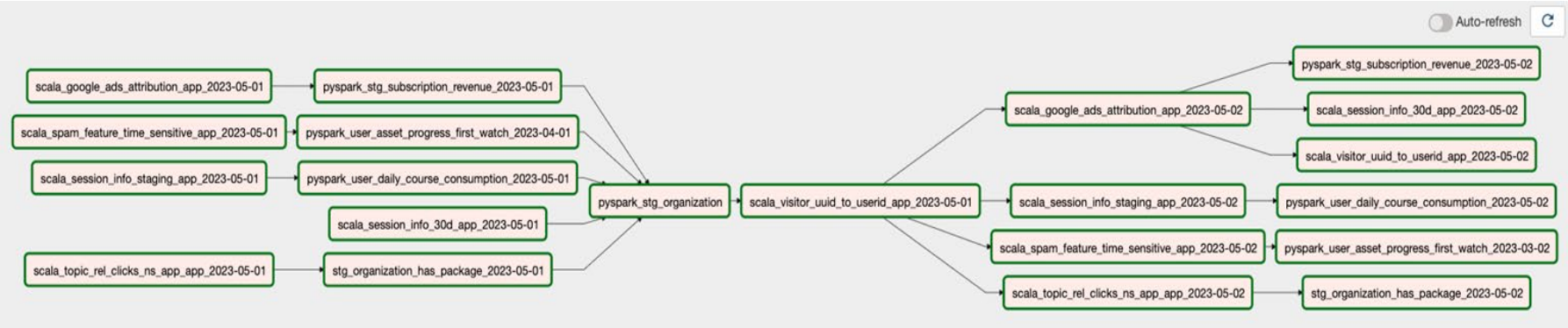
Primary Concerns

- Skepticism on cost per compute unit with DBU + EC2 cost
- Challenges around integration with airflow
  - Overall compatibility
  - With API's and Airflow-Operator
  - Need for task level granularity
  - Cluster spin up times possibly impacting runtimes and cost
- Getting finance and everyone onboard is not trivial.

How we designed the experiment

We invited the competition to work with us to design and execute the experiment.

- EMR6 Serverless vs Databricks offerings
- Used real production workloads with airflow as orchestrator
- Dedicated engineer working with each vendor's SMEs
- Transparently sharing results of each experiment with vendors

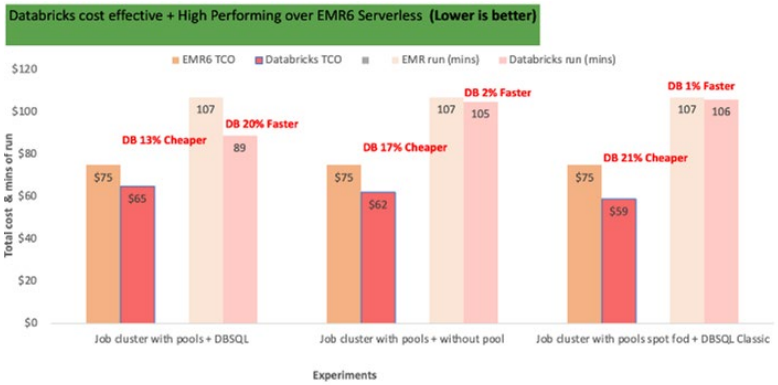


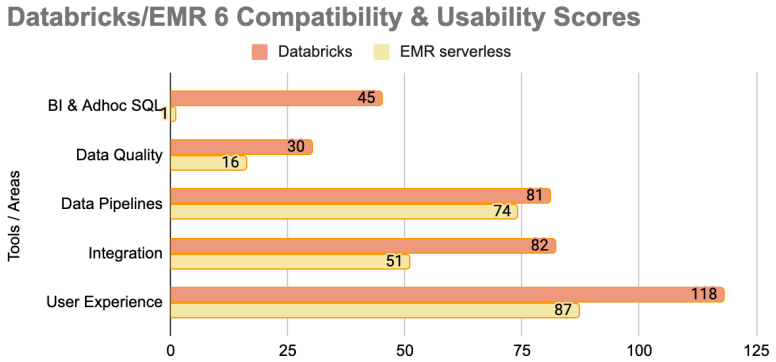
Evaluating the experiment

How did we compare and decide?

- Thorough experimentation with several variations and iterations
- Constant feedback loop with vendor
- Benchmark on cost, performance
- Unbiased by discounts, strive for best performance

- Overall platform potential - lakehouse vision
- Compatibility, scalability and support
- Developer productivity gains
- Ultimately choose best fit for requirements



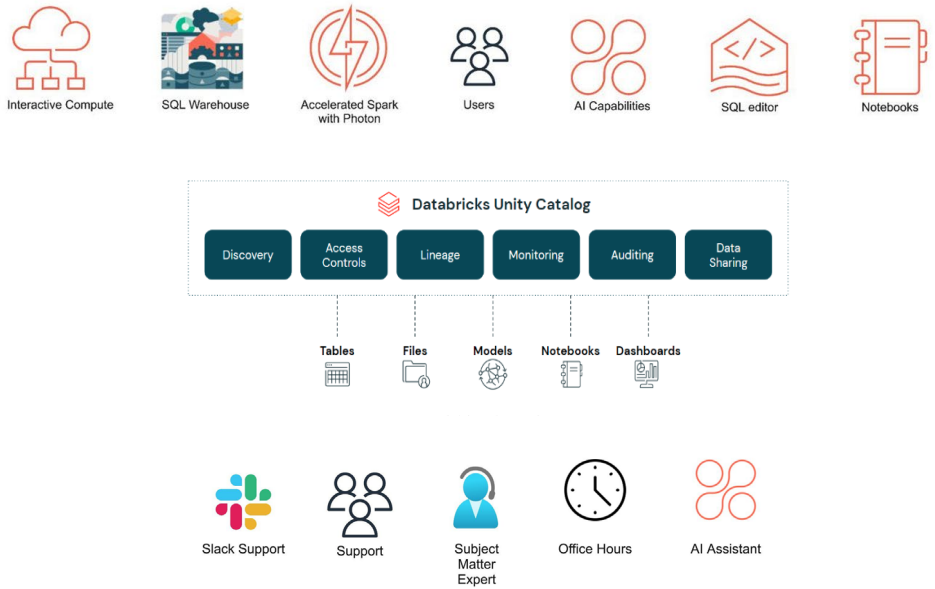


Why Udemy Chose Databricks?

What Databricks provides Udemy

Databricks provides solutions to some of our largest challenges.

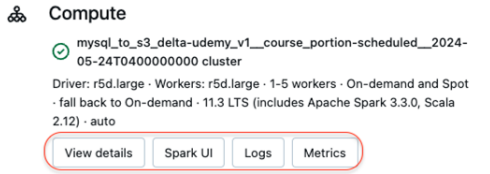
- Single platform and complete ecosystem.
- Notebooks and visualization
- Advanced and efficient Spark engine
- Facilitate collaboration & governance across various data portfolios
- Support & expertise

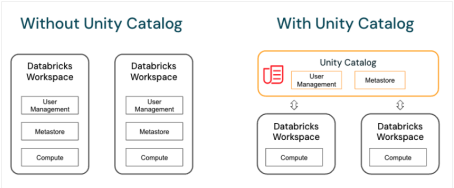


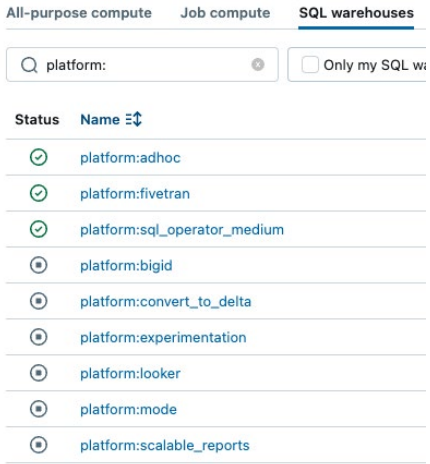
What Databricks provides Udemy

Zooming in we see more benefits to areas such as data access, operational support and security updates.

- Easy access to logs and Spark UI.
- Cluster resource isolation with tagging for audit & cost tracking
- Automated security updates
- Centralized user/groups and access policy management

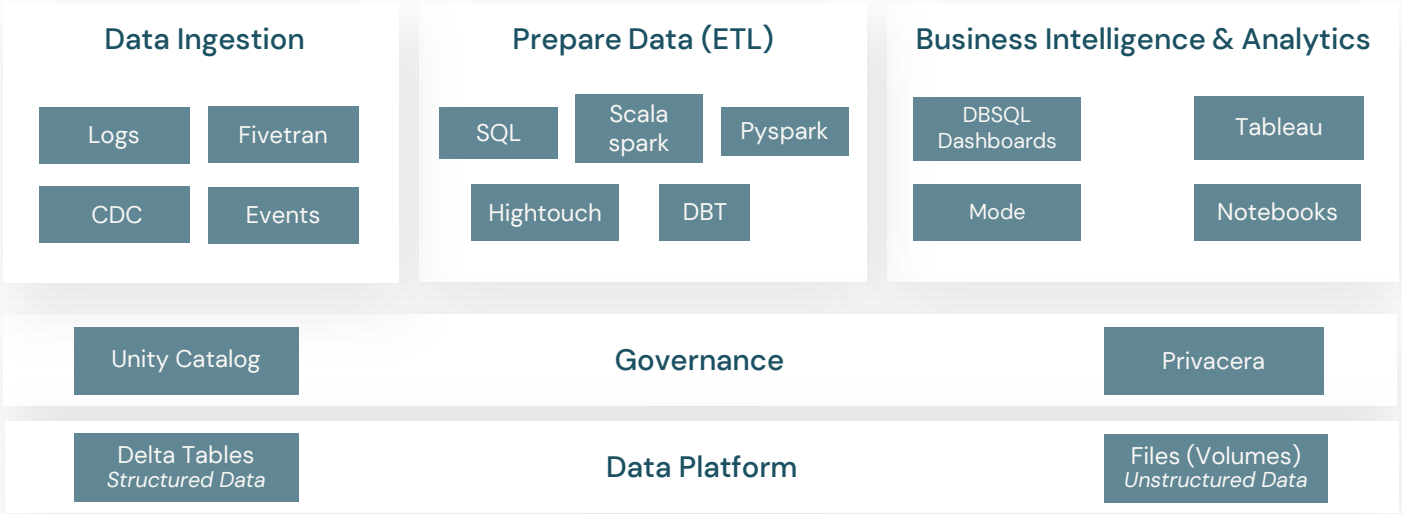






Udemy Data Platform w/ Databricks

All our tools integrated seamlessly with Databricks, providing the essential functionality needed to support a data mesh architecture.



How Udemy Planned for Migration

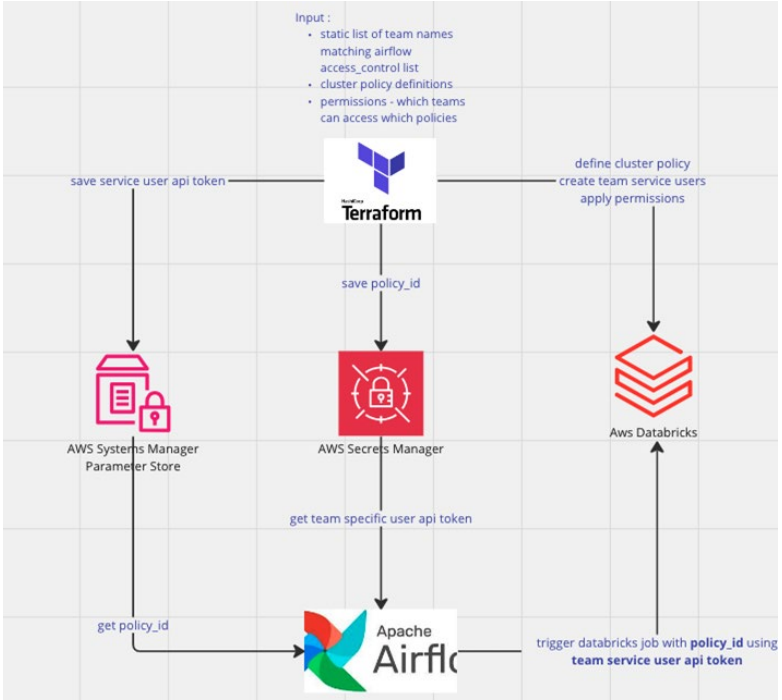
Udemy's migration overview

Comprehensive migration: from data format, metadata management with RBAC to enhanced data processing and visualization.

Control resource utilization

We controlled resource utilization using T-shirt size compute policies and team-based access tokens.

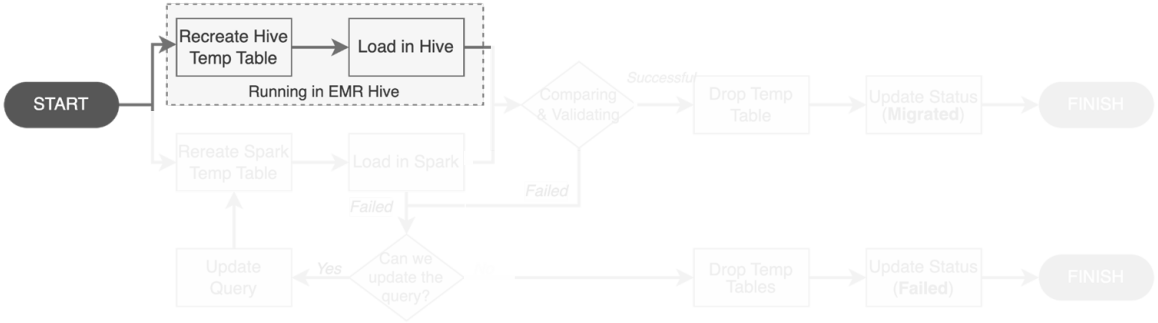
- Terraform to create & manage clusters, compute policies and permissions
- All Purpose cluster & DBSQL - application specific, for resource isolation
- Job cluster - Define T-Shirt (S, M, L) size compute policies with guardrails
  - Policy id stored in parameter store
  - Service principal API token stored in secrets manager
  - Limit access to very large clusters for cost controls
  - Limit high number of concurrent clusters for infrastructure stability
- Airflow uses team specific service principal api token & required T-shirt size policy id to trigger databricks job



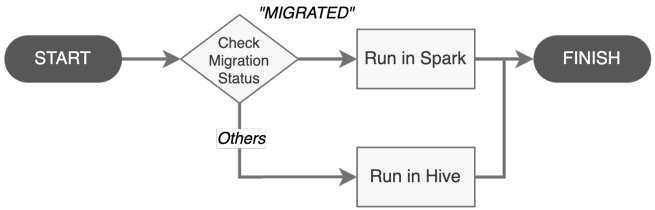
Moving from Hive to Spark

Migrating hundreds of ETLs to Spark with no stakeholder impact is hard.

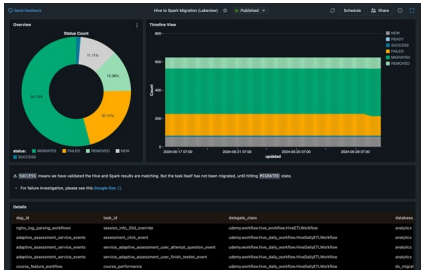
- **We have gained 3X performance in Spark compared to Hive on average*.**
- Migrate in order from downstream to upstream to ensure compatibility.
- Strive for changes to be transparent to user.
- Classify failure cases and resolve them in batches, starting with the low-hanging fruit and progressing to tricky edge cases.



In Airflow



Dashboard for tracking migration progress

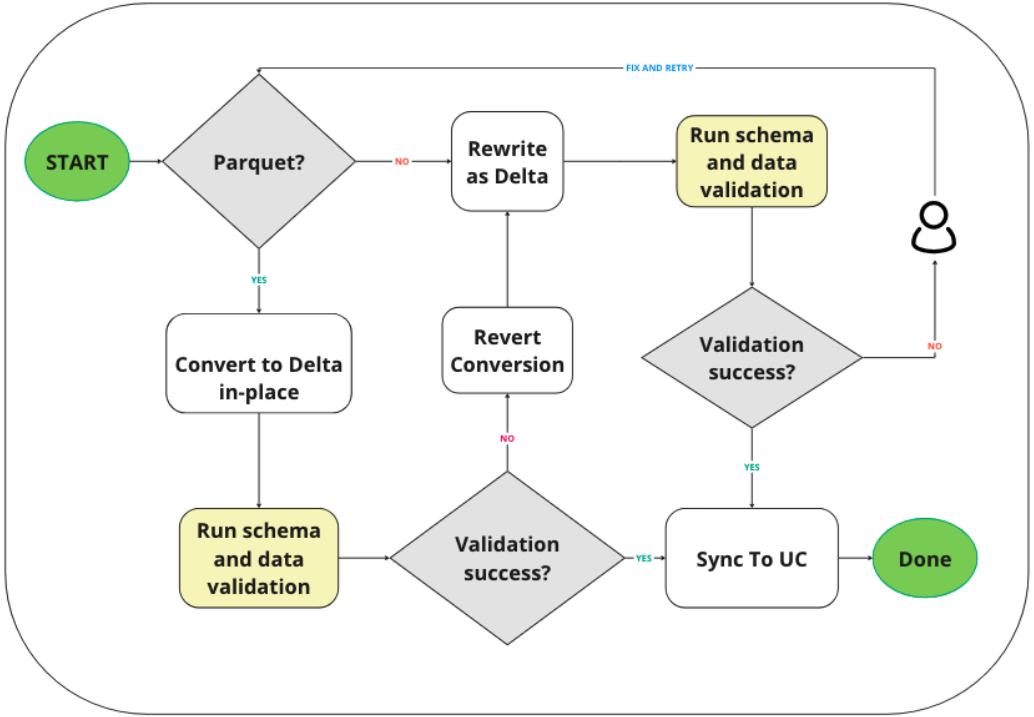


**Not strict comparison, based on Hive running in EMR 5.31 and Databricks SQL warehouse Medium size. Your mileage may vary.*

Unity Catalogs works best w/ Delta

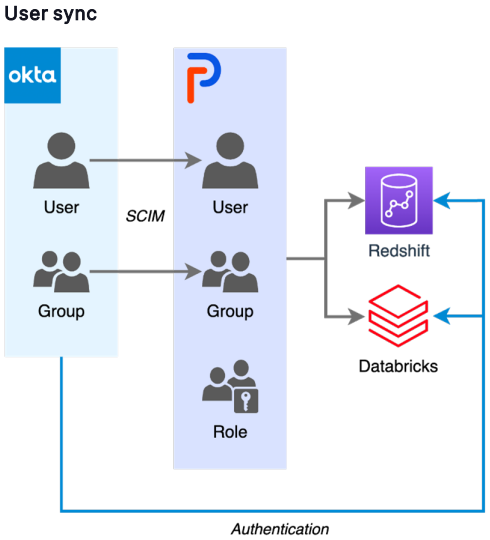
Converting your datasets to Delta will expedite UC enablement.

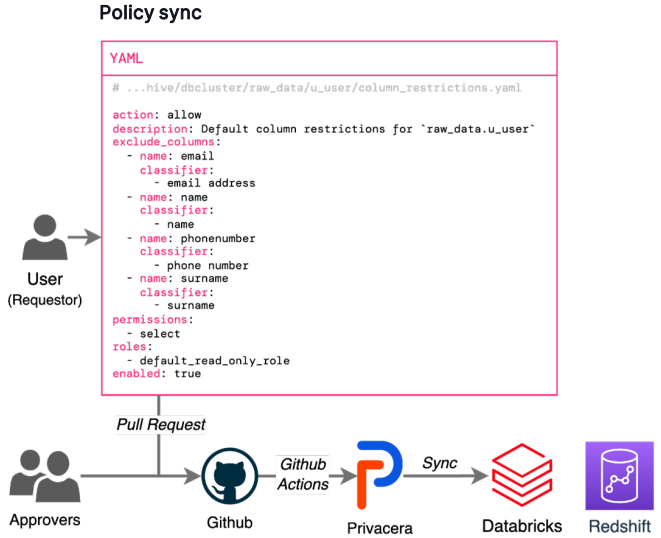
- Classify inventory of datasets
- Determine datasets with non compatible dependencies to skip
- Migration tracker
- Use in-place conversion with parquet format and re-write for non-parquet
- Perform schema and data validation
- Automatic rollback if validation fails



Plan for Data Governance

We integrated Privacera to help manage data access.





Users and groups are synced from Okta to Privacera which are in turn synced to Redshift and Databricks.

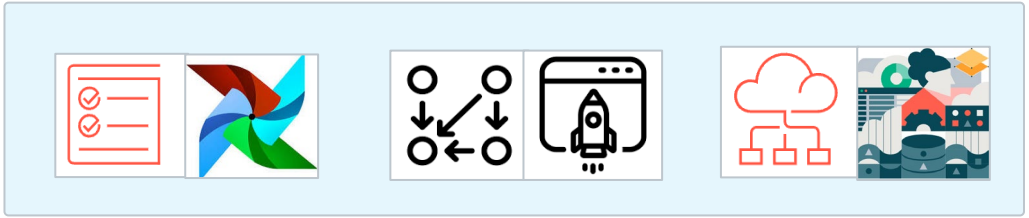
We use Github as a single source of truth for policy permissions. Changes can be authored by data producers and consumers but must be approved

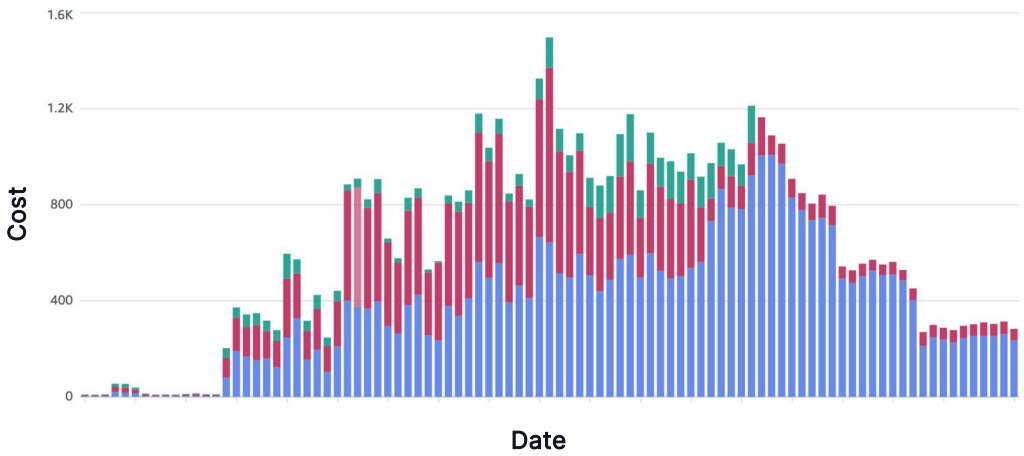
Lessons Learned

Migrations can be complex

Start with your scheduler, it will largely dictate what to migrate and when.

- **Determine your scheduler:** What features do you need like sensors, task groups, UI?. Be aware of cluster start costs and time.
- **Assess dependencies and rollout strategy:** Hive. Sqoop, Sagemaker workflows cannot read delta format.
- **Use cluster policies:** Abstracts complexity from users. Easier to swap cluster specifications
- **Photon:** Included in DBSQL pricing but incurs additional DBU with AP/Jobs cluster, benefit of enabling photon is dependent on workflow/queries.
- **Use tagging:** Spikes may occur when you start migrations but with the right tags and tracking you can control the spend.

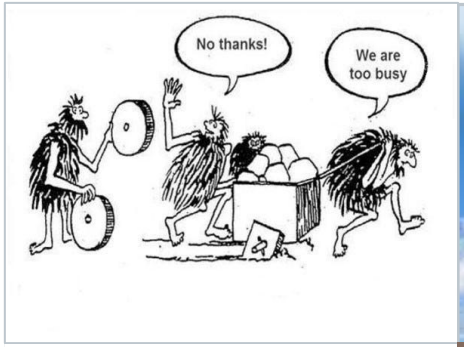


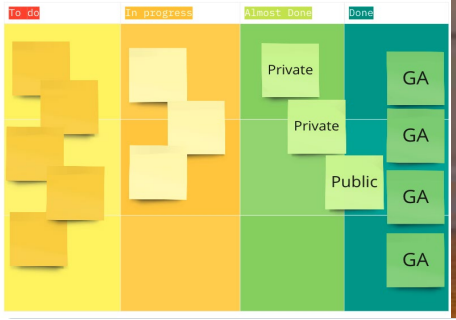


Protect yourself against pitfalls

Inherently things will break, plans will change.

- If you've a small team supporting large org, do not reinvent the wheel
    - Access to logs, spark UI, cluster events and metrics
    - Lineage, tagging and RBAC
- CI-CD process may need to change for ephemeral clusters - init scripts for datadog-agent, managing configs & secrets
- Private Preview - understand the gaps & timeline
    - CLM in preview did not work with delta merge/clones
    - External HMS as federated catalog timeline did not work for us, Rather conversion to delta will get us further on path to UC adoption.
- Include schema and data validation with every change
- Automate and include quick rollbacks



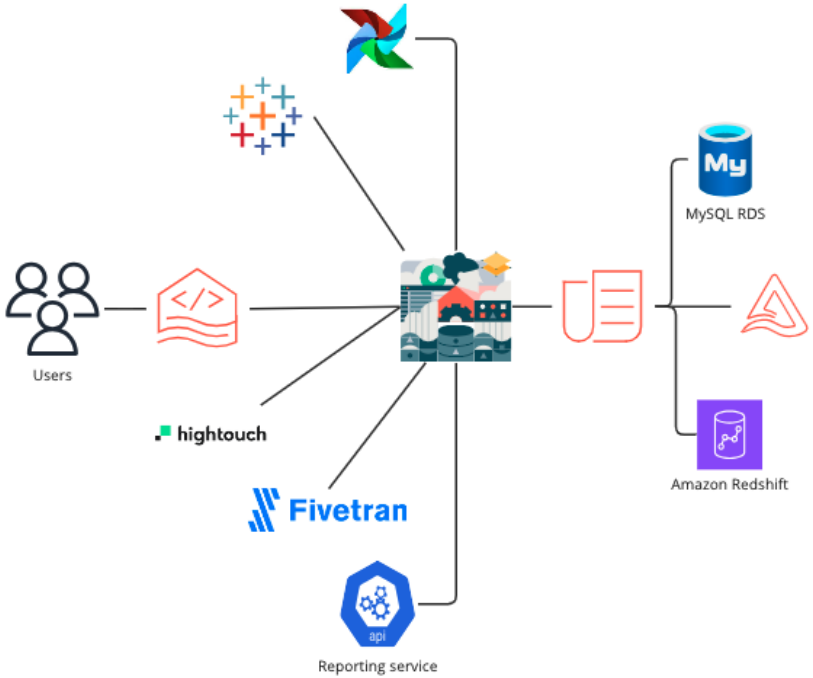


Where to Next

Migration is still underway

We will look to optimize usage of Databricks.

- Lakehouse federation - SQL editor as one stop shop for all querying needs
- Use DBSQL more
- DLT, Databricks scheduler with workflows, Serverless offerings
- Lakehouse IQ and Databricks AI features, DBRX
- Lakehouse monitoring to provide insights based on statistical measures on key datasets
- Refine security story with automated PII detection, symmetric encryption, attribute based access policies
- Assess Udemy's ML stack with Databricks



QA