

Lessons Learned while Migrating Data Pipelines from Enterprise Schedulers to Airflow

Shivnath Babu

CTO/Cofounder @ Unravel
Adjunct Professor @ Duke University

Hari Nair

Senior Software Engineer @ Unravel

TRUSTED BY



About the speakers



Shivnath Babu

Cofounder/CTO at Unravel
Adjunct Professor of Computer Science at Duke University
Focusing on manageability of data pipelines & modern data stack
Recipient of US National Science Foundation CAREER Award,
IBM Faculty Award, HP Labs Innovation Research Award



Hari Nair

Senior Software Engineer @ Unravel
Team Lead, Customer Success and Innovation
Focusing on Data Science and Insights

Unravel radically simplifies DataOps & has strong adoption across platforms & industries

uncover

- Brings together information about all your apps, clusters, resource utilization, users, & datasets in a single place

understand

- Creates end-to-end view of data pipelines to easily track & understand issues
- Tracks & reports on usage across environments
- Checks for & alerts on anomalous behavior

unravel

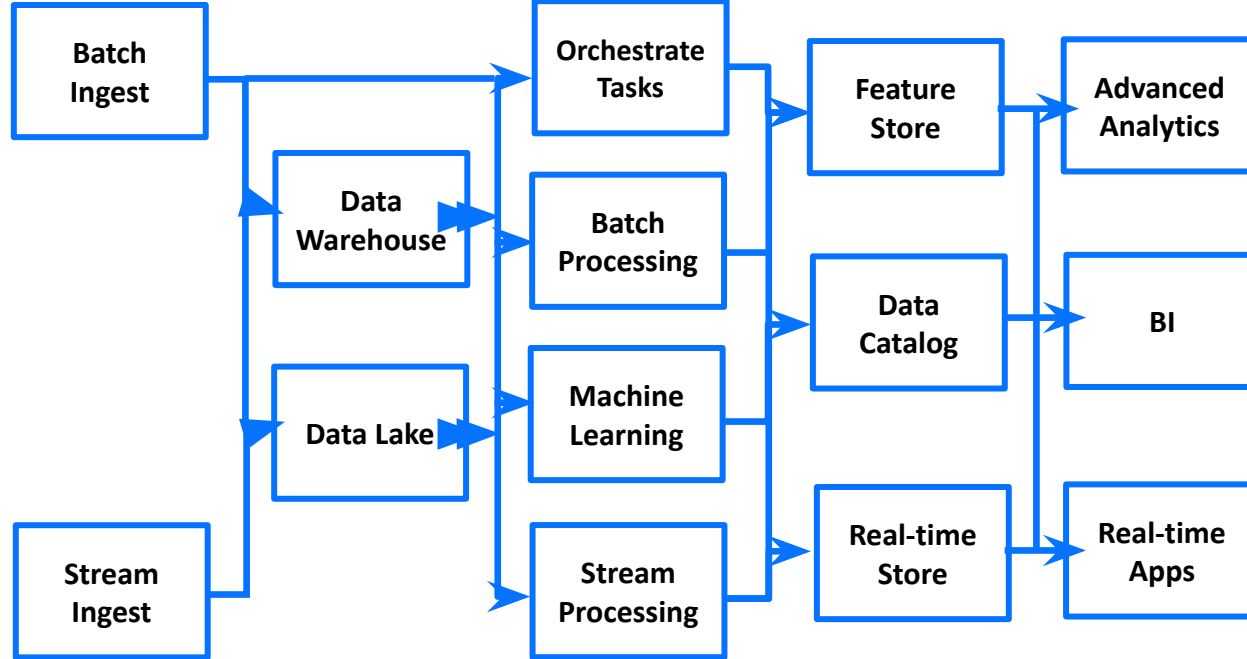
- Uses AI/ML to troubleshoot & optimize apps to meet desired performance & cost needs
- Spots & fixes inefficient usage
- Ensures efficiency, quality, & performance of all apps in development & production



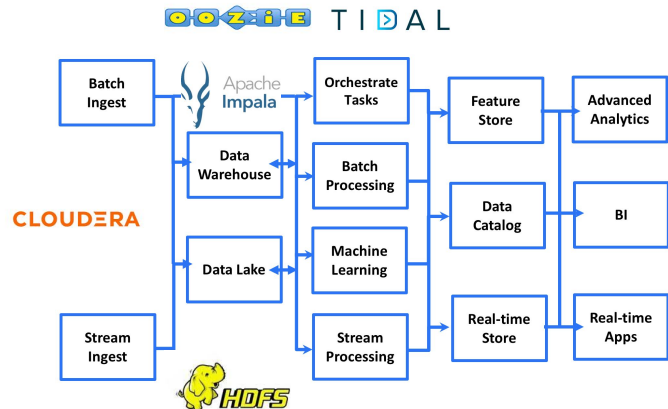
**Many enterprises are modernizing
their data stack and pipelines**

DATA PIPELINE

DATA SOURCES	CAPTURE	STORE	TRANSFORM	PUBLISH	CONSUME	DATA PRODUCTS
--------------	---------	-------	-----------	---------	---------	---------------



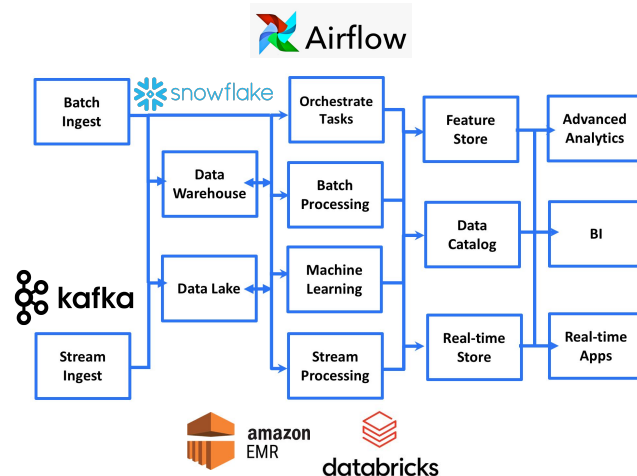
Many enterprises are modernizing their data stacks and pipelines



Large clusters supporting multiple apps and tenants

Less agile

Harder to scale



Smaller, decentralized, app-level clusters

Very agile

Easier to scale

Goals of modernization

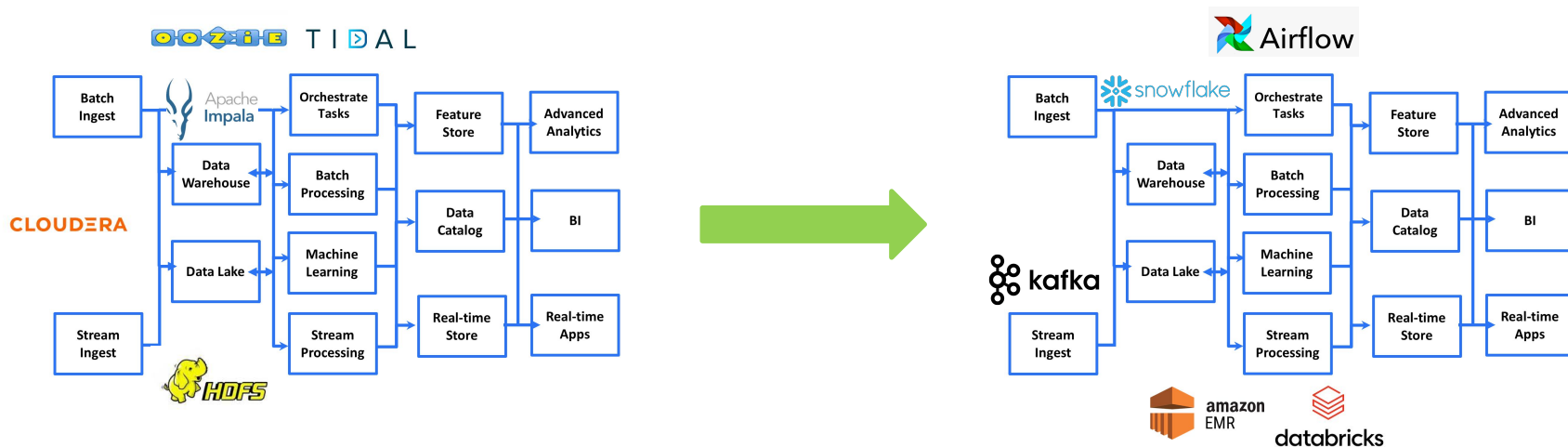
- Improve agility
- Resources no longer become the constraint
- Reduce cost

Why Airflow gets picked as part of modernization:

- Well suited for agile development
- Better suited for cloud-native architectures than traditional schedulers
- Available as a service

Two main phases of modernization

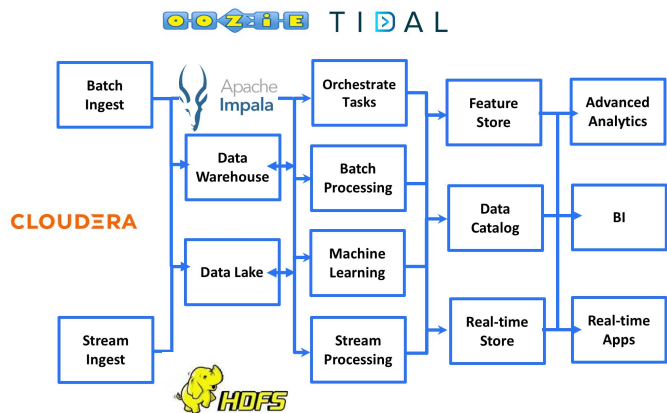
Phase 1: Assess and Plan



Phase 2: Migrate, Validate, and Optimize

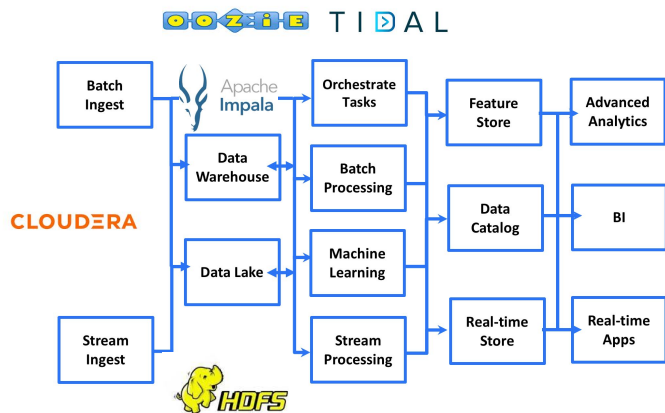
Assess and Plan: Lessons Learned

Assess & plan phase of modernization



- Pipeline discovery
- Resource usage analysis
- Dependency analysis
- Complexity analysis
- Mapping to target environment
- Cost estimation for target environment
- Migration effort estimation

Assess & plan phase: Lessons learned



Pipeline discovery itself can be challenging

- Multiple enterprise schedulers may be in use, e.g., Autosys, Informatica, Oozie, Pentaho, Tidal, etc.
- No common pattern may exist

Fine-grained tracking is needed for accurate resource usage and dependency analysis

Fine-grained tracking for accurate planning

Metadata

Input Data

KPIs

Output Data

HIVE hive_20210402002323_ca97ec23-78a2-4c15-9c65-d41315ce0da2 Workflow

Unravel has 4 recommendations to improve app efficiency.

HIVE Hive Query
hive auser default
cluster 04/01/21 17:23:08 04/01/21 17:33:56

Navigation Execution Graph Gantt Chart Tags

TYPE	STATUS	ID	START TIME	DURATION	I/O	
MR	SUCCESS2576110	04/01/21 17:25:55	3m 12s	101.20 GB	2
MR	SUCCESS2576655	04/01/21 17:31:36	2m 5s	413.22 MB	2

Query Table Task Attempts Attempts Copy Query

```
1 INSERT OVERWRITE TABLE USER_ACTIVITY PARTITION (  
2 CATEGORY  
3 ,PARTITION_DATE  
4 )  
5 select member_guid , activity_date , product , count(*) as activity  
6 from  
7 (  
8 select upper(split(userguid , '[@]')[0]) as member_guid , case when  
9 when (upper(params['event.type']) = 'DOWNLOADHIGHRES' OR upper(par  
10 when upper(params['event.type']) = 'SEARCH' then 'SEARCH'  
11 end as product, event_date as activity_date  
12 from  
13 stock_events  
14 where  
15 upper(params['event.type']) IN ('DOWNLOADWATERMARKED' , 'DOWNLOADHI  
16 AND event_date <= date_sub('2021-03-31', 3) AND event_date >= '2021-03-31')
```

Fine-grained tracking for accurate planning

Metadata

Input Data

KPIs

Output Data

The screenshot displays the Unravel interface for a Hive query. At the top, a navigation bar includes a 'Workflow' button. Below it, a status bar shows 'HIVE Hive Query' with user 'auser' and cluster 'default'. A table lists execution tasks with columns: TYPE, STATUS, ID, START TIME, DURATION, I/O, and # OF YARN APPS. A SQL query snippet is visible on the right, showing an INSERT OVERWRITE statement for 'USER_ACTIVITY' partitioned by 'PARTITION_DATE'. Red arrows and boxes highlight specific elements: 'Workflow', 'HIVE Hive Query', 'DURATION 10m 47s', 'DATA I/O 101.60 GB', 'USER_ACTIVITY', and 'stock_events'.

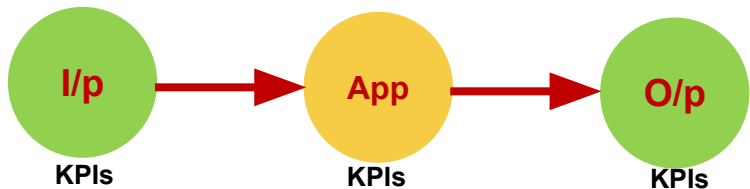
TYPE	STATUS	ID	START TIME	DURATION	I/O	# OF YARN APPS
MR	SUCCESS2576110	04/01/21 17:25:55	3m 12s	101.20 GB	2
MR	SUCCESS2576655	04/01/21 17:31:36	2m 5s	413.22 MB	2

```
1 INSERT OVERWRITE TABLE USER_ACTIVITY PARTITION (
2 CATEGORY
3 ,PARTITION_DATE
4 )
5 select member_guid , activity_date , product , count(*) as activity
6 from
7 (
8 select upper(split(userguid , '[@]')[0]) as member_guid , case when
9 when (upper(params['event.type']) = 'DOWNLOADHIGHRES' OR upper(par
10 when upper(params['event.type']) = 'SEARCH' then 'SEARCH'
11 end as product , event_date as activity_date
12 from
13 stock_events
14 where
15 upper(params['event.type']) IN ('DOWNLOADWATERMARKED' , 'DOWNLOADHI
16 AND event_date <= date_sub('2021-03-31', 3) AND event_date <= '2021-03-31')
```

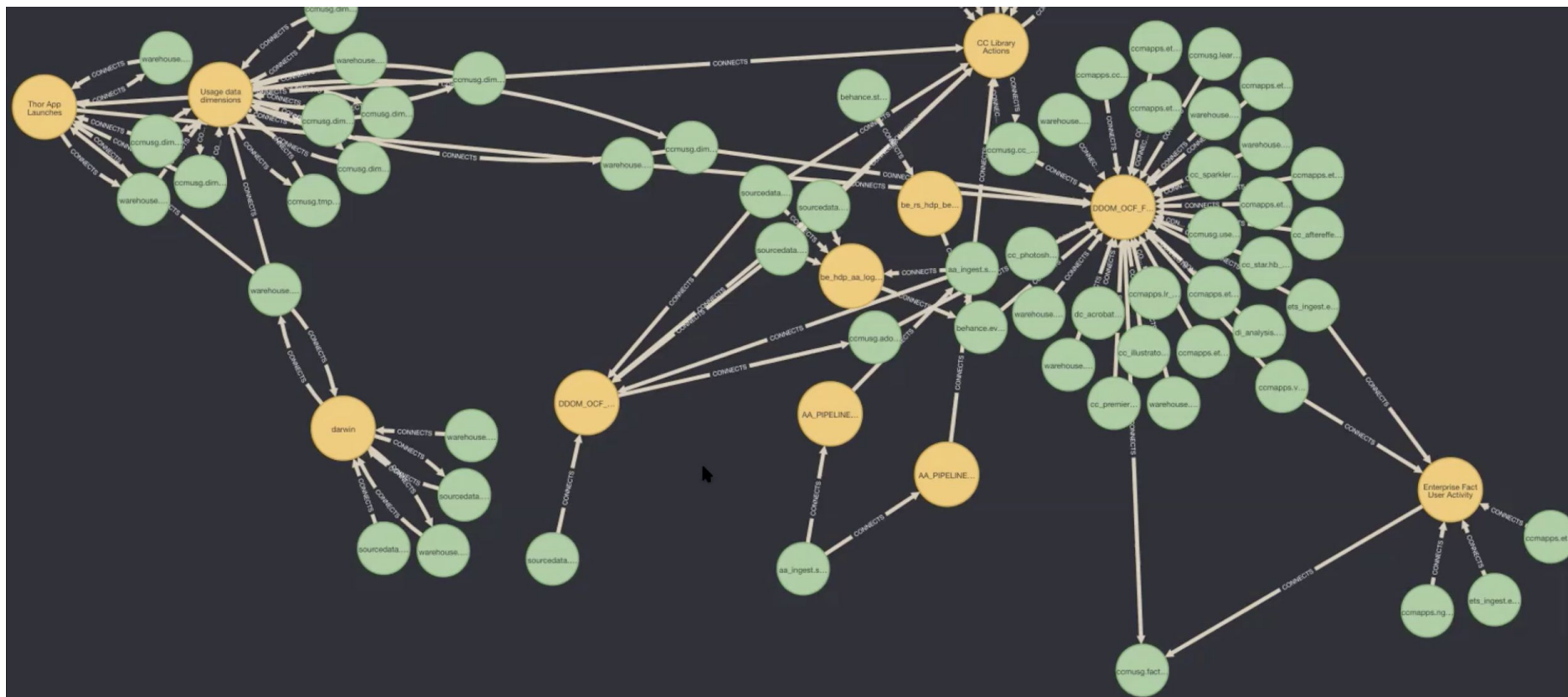
Metadata

Metadata

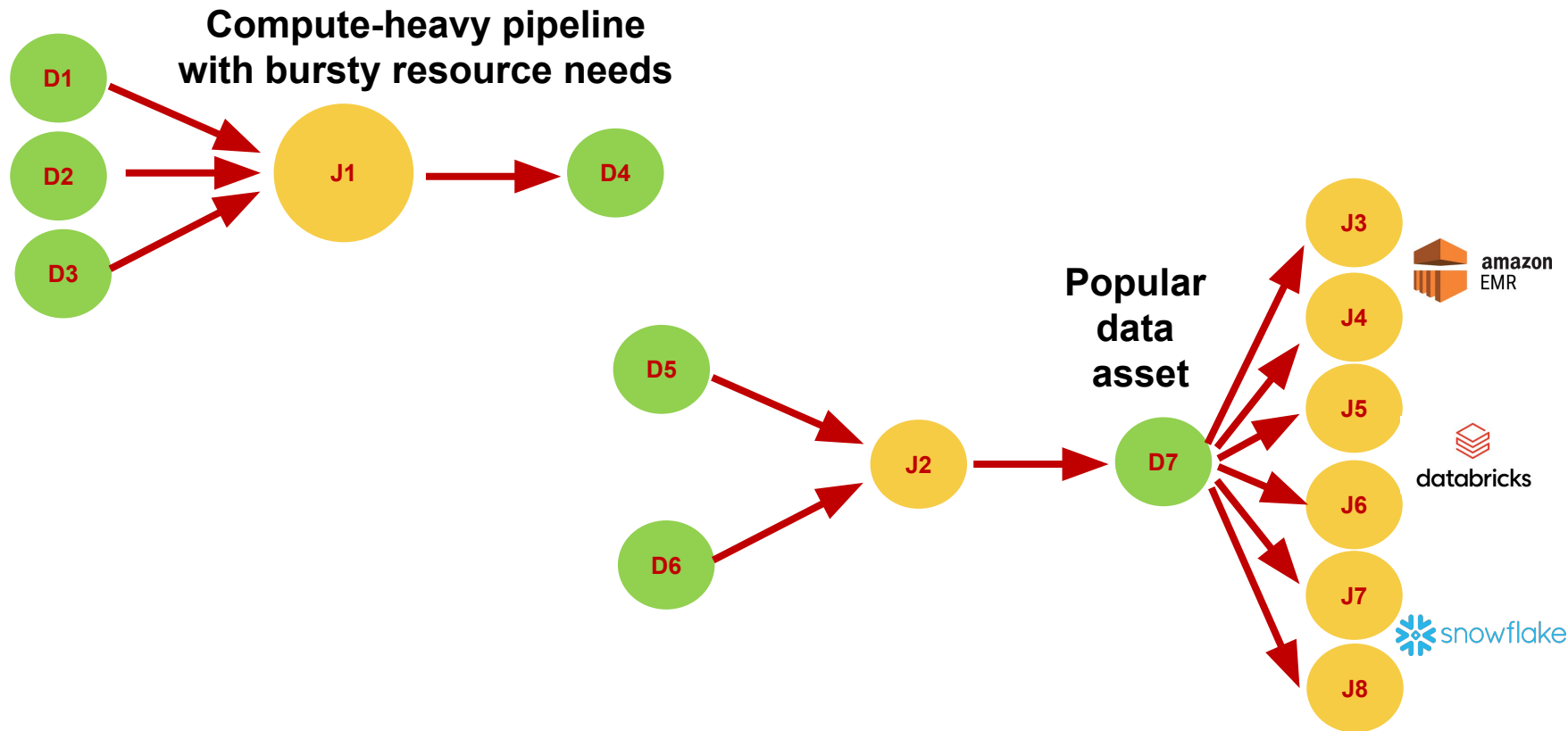
Metadata



The annotated lineage graph

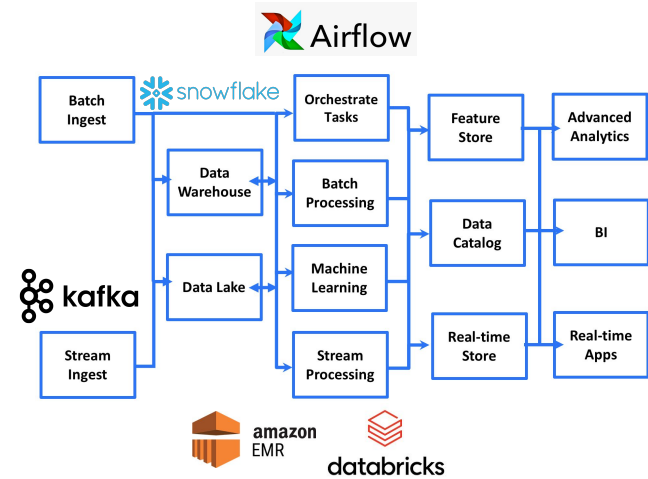
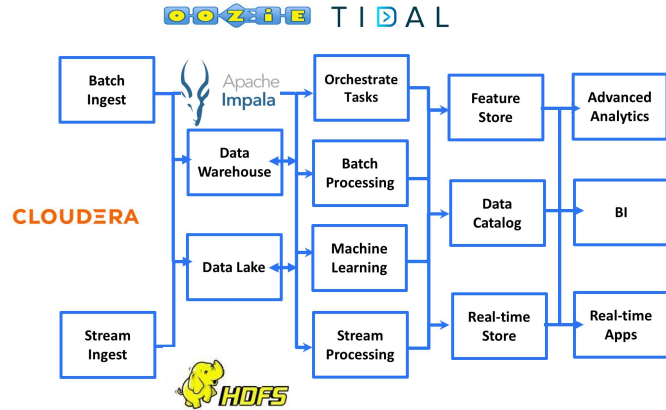


Picking the best migration execution strategy



Migrate, Validate, and Optimize: Lessons Learned

Migrate, Validate, & Optimize phase of modernization



Criteria to Consider

Correctness

Performance

Cost

Undesired Behavior

Wrong results, Failing pipelines

Missed SLAs, Growing lag/backlog

Cost overruns, Going over budget

Guaranteeing pipeline correctness after migration

Ensure that the right checks are in place to validate correctness after the migration

Example checks:

- Daily partitions of Table “SignupsAndSubs” should have at least 1000 records
- “customerPinNumber” should not be NULL

Tools like ***Great Expectations*** make it easy to define checks

Guaranteeing pipeline performance after migration

Ensure that baselining is done and SLAs are defined to ensure performance needs are met after the migration

Example SLAs:

- Pipeline should finish by 6:00 AM PST
- Data in dashboard generated by the pipeline should not be older than 10 mins

SLAs can be defined in ***Airflow***

Tools like ***Unravel*** help pinpoint bottlenecks and suggest performance fixes

Controlling pipeline costs after migration

Ensure that cost budget estimation & planning are done before the migration

Example budget specifications:

- Cost of any one run of the “BI-report” pipeline should not exceed \$100
- Budget for the pipelines generating the “probable_churn” table is \$1M/month

Tools like ***Unravel*** help with cost projection and also recommend fixes for cost inefficiencies

Demo

Sign up for a free trial!

<https://unraveldata.com/saas-free-trial>

shivnath@unraveldata.com

hari@unraveldata.com

Check out our next talk:

**Data Pipeline HealthCheck for
Correctness, Performance, and Cost Efficiency**