May 23–27, 2022
AIRFLOW SUMMIT
Kyte

Scalable and Isolated Dag Development Experience at Lyft

Max Payton & Paul Dittamo
What is our scale

**Users**
- Kyte - 60 WAU
- Airflow - > 350 Active users
- Over 1000 lifetime users
  (including attrition)

**DAGs**
- 4000 DAGs on airflow
- 2500 Active DAGs
- Some DAGs have >1000 tasks
Problem Statement
What Problems were we solving?

Lots of requirements for our Airflow Instance
- Python packages
- Environment Variables
- Airflow Configurations

Pipeline development requires production data
- Mocking data is hard
- Staging data is too random
- Statically defined data becomes out of sync with upstream

Wide Variety of Users
- Some technical (SWE)
- Some not (Legal Compliance Teams)
- Need them all able to test DAGs
Implementation
What is Kyte?

Built on existing Lyft platform
- [Lyft internal container orchestration service](#) (ML Model Training + Batch Prediction)
- Kyte container mirrors Airflow production containers

Additional Details
- Sync DAG changes with git
- Wrapper commands to test DAGs
- No scheduler
Key Features

Users

1. Isolation
   - Metastore Isolation
   - File System Isolation
   - Performance Isolation

2. Ease of Use
   - Utilities Pre-installed
   - Persistent State
   - Jupyter Notebook Support

3. Read access to production resources
   - Users write to personal schemas
   - IAM roles + Envoy routing
Key Features

Platform/Support

4. Ease of Support
Remote Development Environment = 100% Visibility
+ Commoditized Instances

5. Auditable
Given the access to production data, it's important to have a trace of the work

6. Historical use case support
Build the new while supporting the old


Inspired by Google Collab and Polynote
Demo
Learnings
What did we learn?

Users don't love Github as a syncing mechanism
- Users might be unfamiliar with Github syntax
- Uncomfortable committing incomplete code
- Some context switching between environments

Varied preferences for developing locally versus remotely
- Jupyter incomplete as IDE
- Remote environment introduces lag
- Setting up IDE can be a cost

Testing failure cases can be difficult
- SLA misses
- Upstream failures
- BranchPythonOperator cases
Next Steps

Future Work

Testing Support
- Failure cases
- Integration testing of DAGs

Connect the Notebook
Kernel locally
- Improve local development

Linting/Static Analysis
- Already some work for this open source
- Errors introduced when SQL is copied over and parameterized
- Blessed DAG structure
The Dream

Northstar

As Jupyter Notebooks became to Data Analysis, Kyte will become to Airflow dag development