Optimizing Critical Operations: Enhancing Robinhood's Workflow Journey with Airflow

Palanieppan Muthiah
Kevin Wang
Peiqiu Tian

Agenda

- Airflow use cases at Robinhood
- Robinhood’s airflow architecture
- High Availability & Reliability
- Multi Environment platform
- Migration to Airflow 2
- Future plans

Robinhood’s Airflow Use Cases

Trading

- Margin management
- Options exercise
- Checks equity orders
- Equity cost adjustment

Clearing

- Confirm the executions
- Equity settlement
- Track all equities and cryptos we own

Money Movement

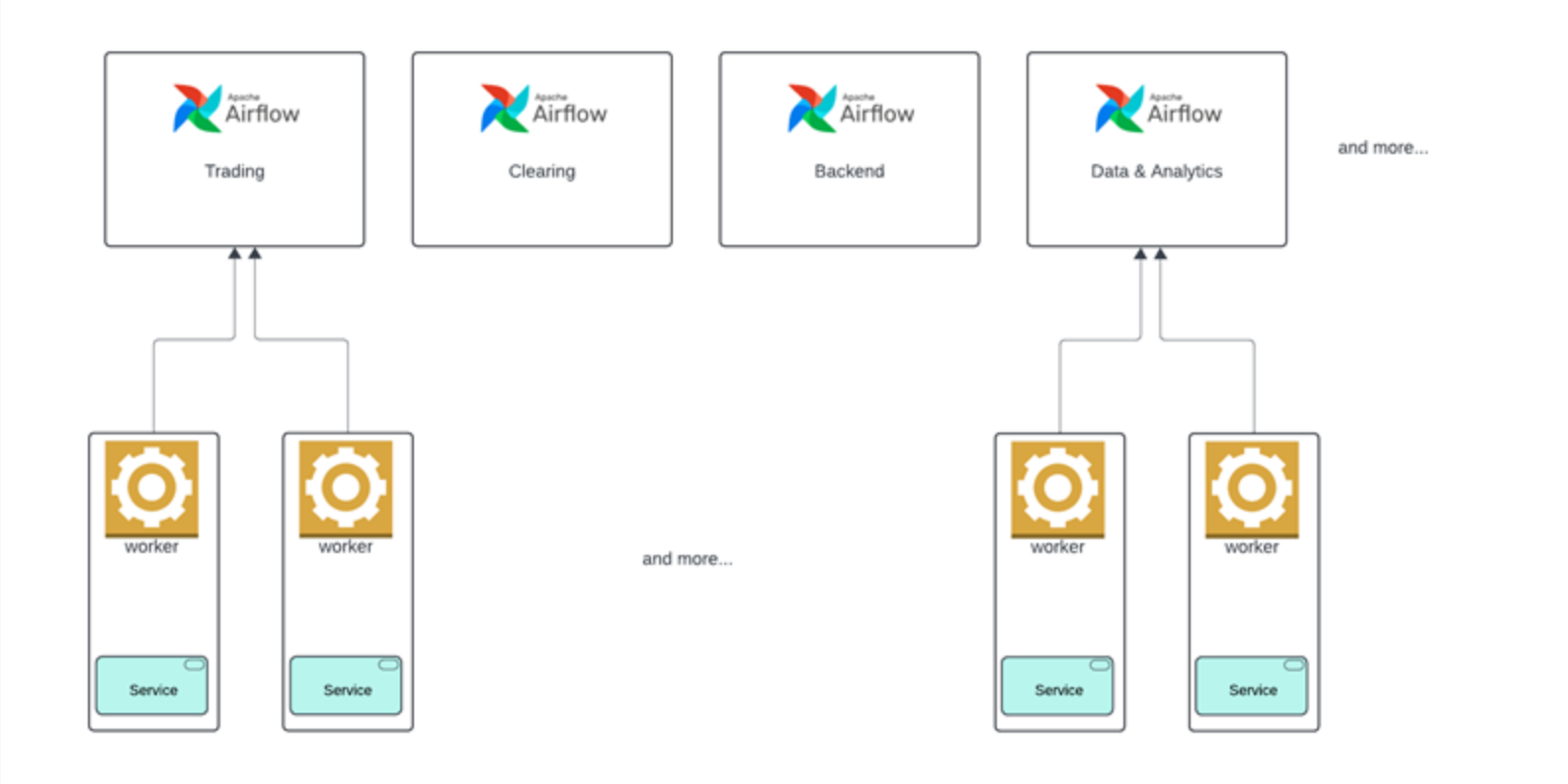
- Batch transactions
- Report Generation
- Feeding data for clearing and accounting

Other Backend Services

- Fetching news sources
- Generating audience of the notifications
- Data ingestion for search

Architecture

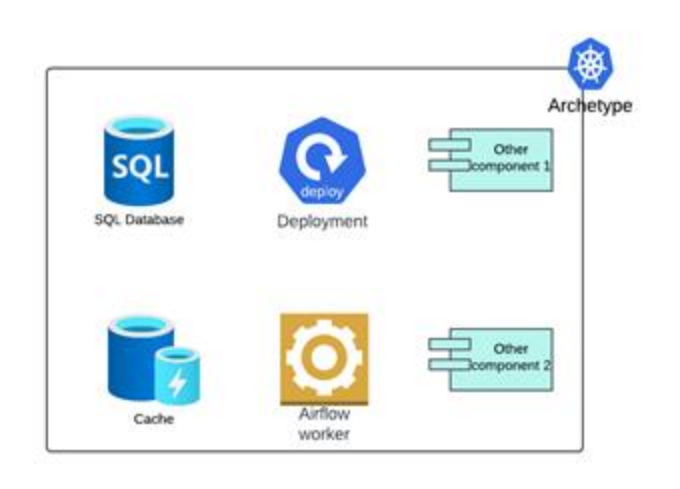
Robinhood’s Airflow architecture



Cluster architecture

Robinhood application integration

- All services are deployed on kubernetes clusters
- Robinhood has Kubernetes service definition template - [Archetype]
- Archetype
  - Shortform definition of various service components
  - Think K8s deployments, databases, caches etc
  - Airflow worker is a component defined in the template
    - Associate the worker to an airflow cluster via config
    - Define DAGs & publish DAG repo
  - Along with it, we provide binary build artifact (Bazel) that contains all Airflow related code



High availability

- Multiple Kubernetes clusters
  - Survive Kubernetes cluster outages
- Multi AZ configuration when possible
- Consider availability at every layer of the stack
  - Scheduler
  - Worker
  - Queue
  - Database
- Airflow 2 High Availability features are critical

Reliability

- Monitoring - track and maintain SLA for key metrics
  - DAG parsing time
  - DAG deployment/sync delay
  - Scheduler delay
  - Queue length
  - Task instance failure rate
- Custom metrics plugin to fill gaps in metrics
- Change management is critical
  - Multiple environments to test changes
- Backup/restore setup for airflow metadata

Benefits & Tradeoffs

- Isolation of workloads ✅
  - Critical workloads can be isolated into dedicated cluster, avoiding noisy neighbor
- Smaller clusters ✅
  - Reduce scalability issues for highly critical workload
- Change management ✅
  - Roll out all changes to lower priority clusters first
  - Avoid critical time windows for deployment - e.g: market open hours
- Upgrades are harder & require more work 🚨
  - Mitigate via lots of automated tools (codegen, migration daemon)
  - Ongoing minor upgrades are still a challenge

Multi-Env Platform

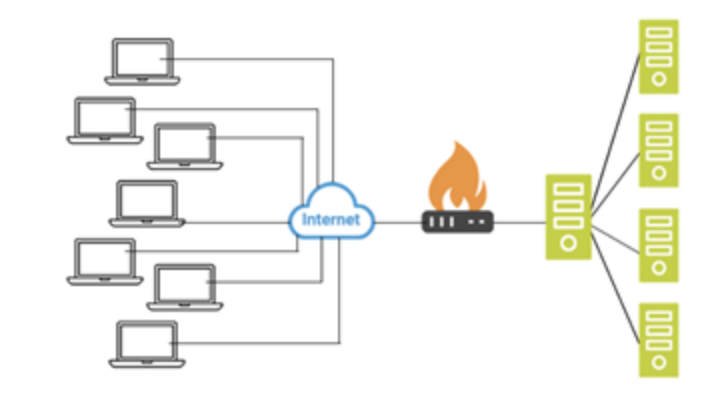
Apollo

- Initial stage of development environment
- personal development on developer's own K8S namespace
- pre-commit and post-commit integration test on isolated environment
- Robinhood Apollo [Blog]

Apollo

Within dynamic K8S namespaces

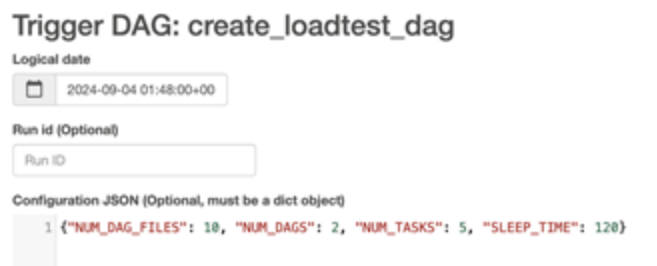
Load & Fault

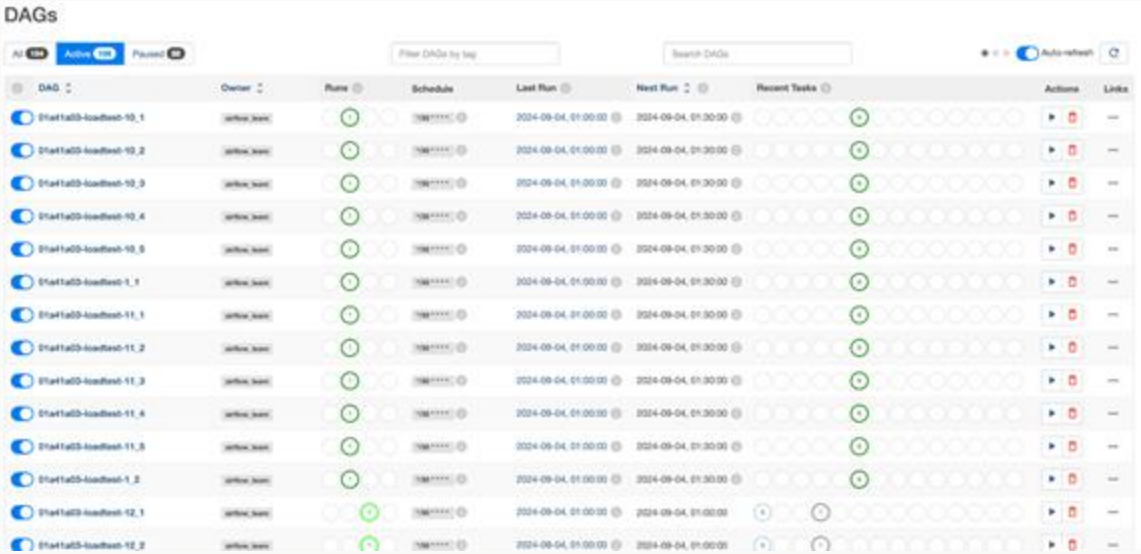


- Generic pre-prod environment
- End to end feature test
- Stress & chaos test

Stress Test

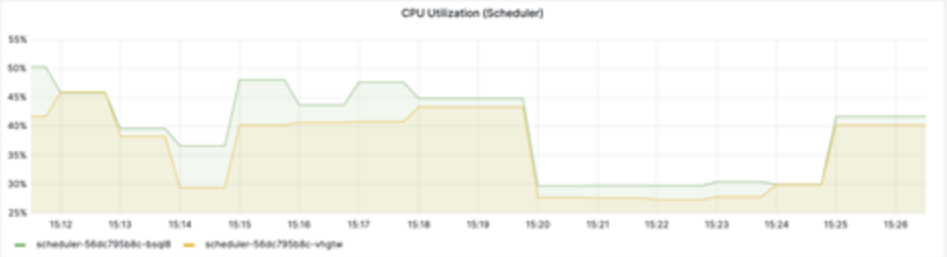
- Stress Generation





Stress Test

- Test Analysis



Production

- the final stage of development where a product is deployed for end-users to use

Migration

Robinhood’s Airflow Migration

Apache Airflow 1.10.1

amazon EC2

SALTSTACK

RabbitMQ

Old

Apache Airflow AIRFLOW 2.

Bazel

kubernetes

amazon SQS

New

Robinhood’s Airflow Migration

Challenges

- 15 Airflow Clusters
- 75 App Workers
- 4,354 DAGs
- 87408 Tasks

Cluster Name	# Workers	# DAGs	# Tasks
Money Movement	14	2082	25875
Clearing	8	295	10644
Backend Service	17	261	1345
Trading	36	1716	49524

Robinhood’s Airflow Migration

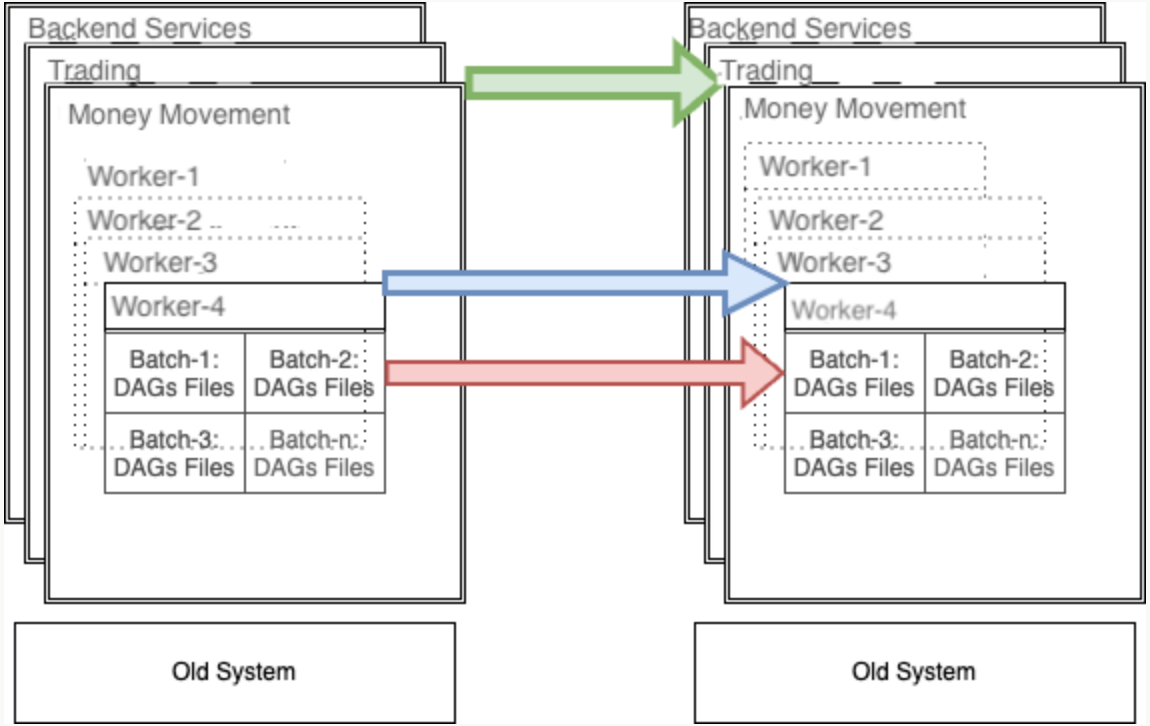
Challenges
15 Airflow Clusters
75 App Workers
4,354 DAGs
87408 Tasks
Critical/P0 DAGs
SLA Tasks
1st-Party and 3rd-Party Deps Compatibility

```
dag_default_args = {
    "owner": OWNER,
    "start_date": datetime(2024, 3, 5),
    "email": [redacted]
    [redacted]
    "queue": "cashier_worker",
    "retries": 0,
    "depends_on_past": False,
    "responders": [redacted],
    "sla": timedelta(hours=8, minutes=5),  # To finish by 8:5PM ET.
    "on_failure_callback": generate_alerts_on_failure(
        slack_channel=slack_channels.CASHIER_ALERTS,
        should_page=True,
    ),
}

dag = PinnacleDAG(
    "[redacted]",
    default_args=dag_default_args,
    concurrency=14,
    schedule_interval="{} {} * * *".format(EXECUTION_MINS, EXECUTION_HOURS),
    sla_miss_callback=get_sla_miss_callback(
        opsgenie_responders=[redacted],
    ),
)
```

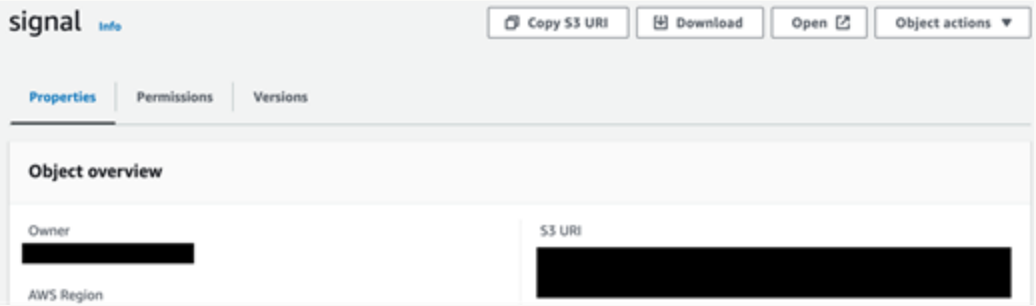
Robinhood’s Airflow Migration

Strategies
Cluster by Cluster
Work by Worker
DAGs Batch by Batch



Robinhood’s Airflow Migration

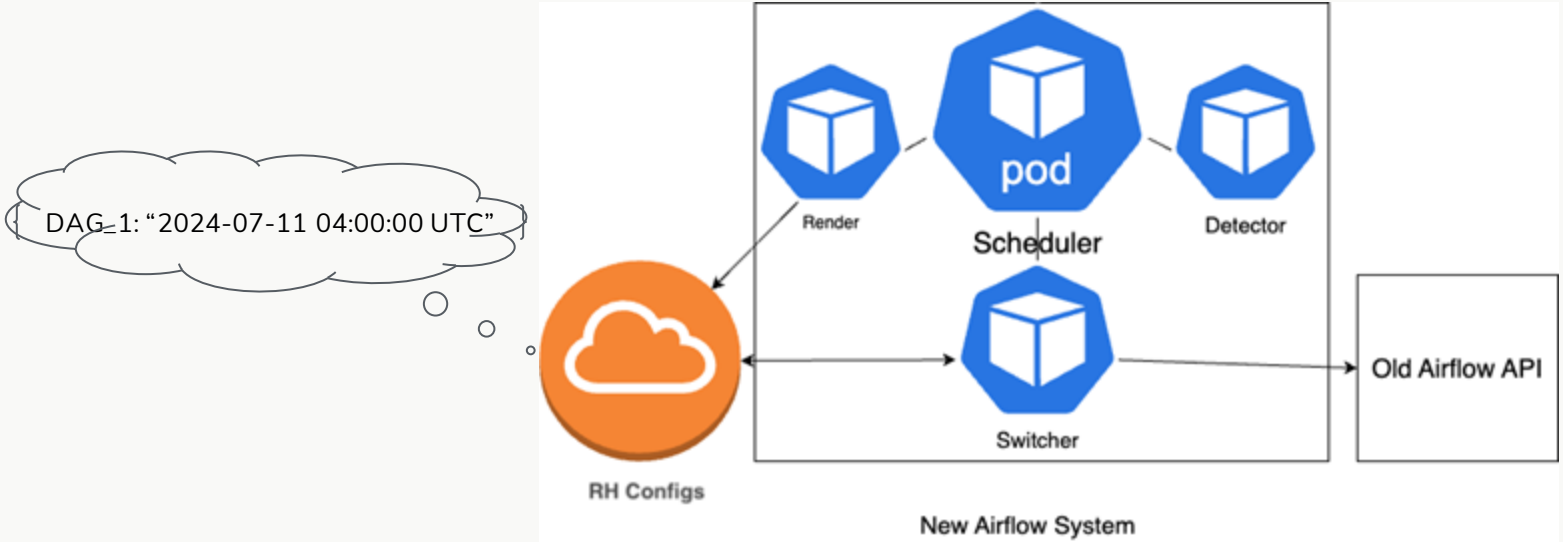
- Customized S3-based Airflow Operators and Sensors
- Handle Daylight Saving Time (DST) changes




```
INFO - Catch 0 keys.   Handle DST changes
INFO - Correct external execution date might be 2023-11-05T04:00:00+00:00 in EasternTimeZone.EDTBehind4H timezone for [redacted]
INFO - Poking 1 keys...
```

Robinhood’s Airflow Migration

Automation Tools



Future Plans

Robinhood’s Future Plans

- New DAG Authentication
- Standalone Backfill Server and Website
- In-Place Airflow Version Upgrade
- AI Integrated Devx Enhancement Tool

Highlights

Highlights

- Critical use cases specifically for fintech
- Architecture for deeper integration with applications
- Multi environments for release
- Personal Airflow cluster for development and testing
- Automation on Airflow 2 Migration

Questions?