

Behind the Scenes: Testing Airflow 3



Introduction



Phani Kumar

Airflow Committer &
Senior Engineering Manager
@ Astronomer

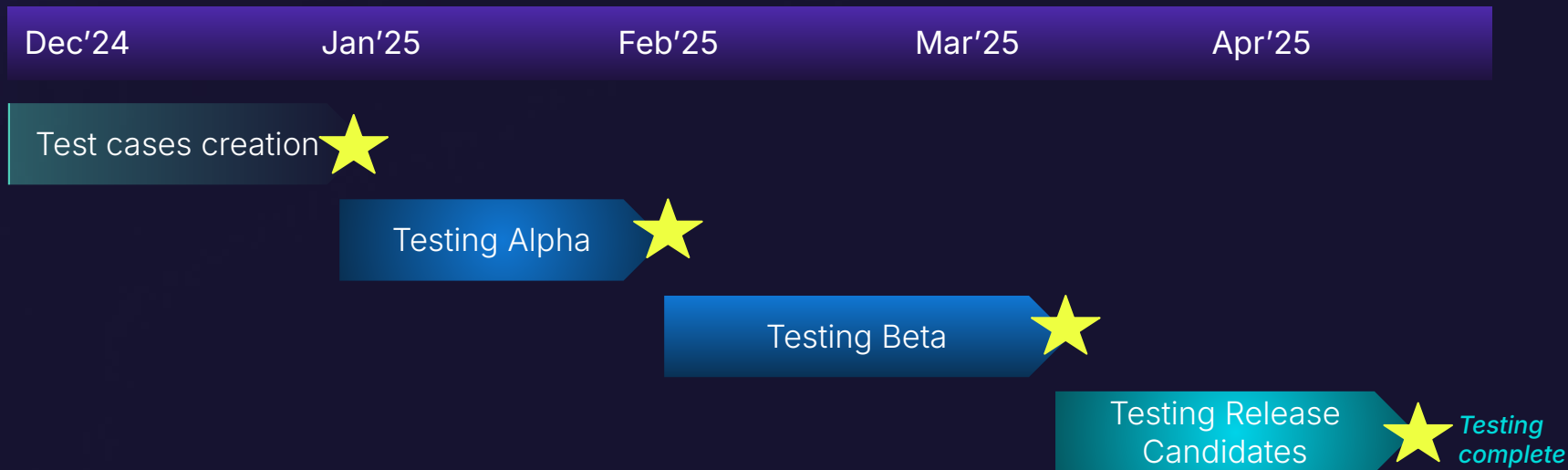


Rahul Vats

Airflow Committer &
Engineering Manager
@ Astronomer



Airflow 3.0 Testing Timeline





Why Airflow 3 Was Different



Architectural overhaul: Task SDK fundamentally changed database communication



Complete UI redesign: Flask-AppBuilder → Modern React application



Migration complexity: Ensuring smooth 2.x → 3.0 transitions



12+ major AIPs: Each requiring comprehensive validation



Production readiness: Millions of DAGs depend on stability



Airflow 3 Test Impact

271 Bugs



- Alpha build: 40
- Beta build: 129
- RC build: 102



Our Testing Philosophy

Testing Pyramid





Our Comprehensive Testing Strategy

Five-Pillar Approach



Task SDK Architecture Testing



Automated Regression Testing



UI Validation



Migration Testing



Comprehensive Validation for 12+ AIPs



Testing Task SDK



Compatibility

- Backward compatibility
- New SDK imports

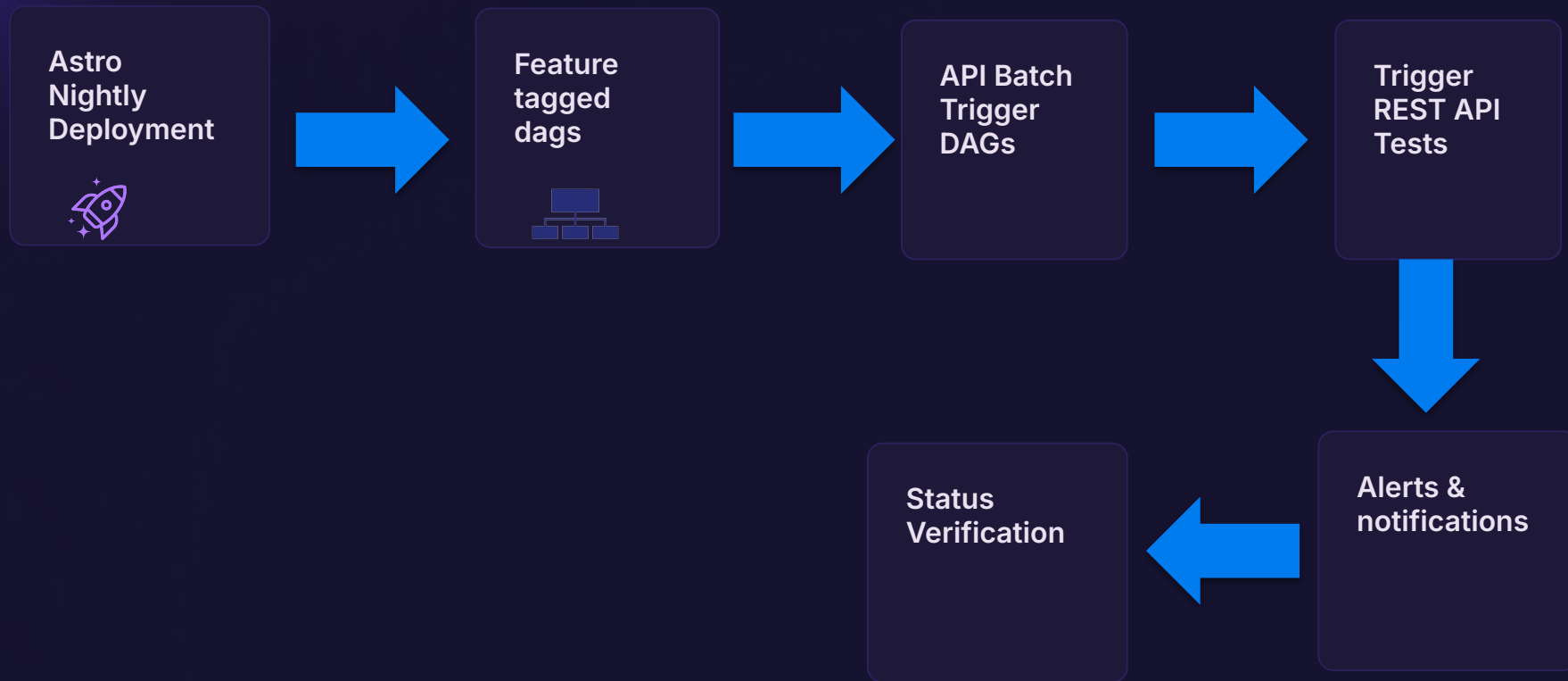


Security Boundary Enforcement

- Direct metadatabase access

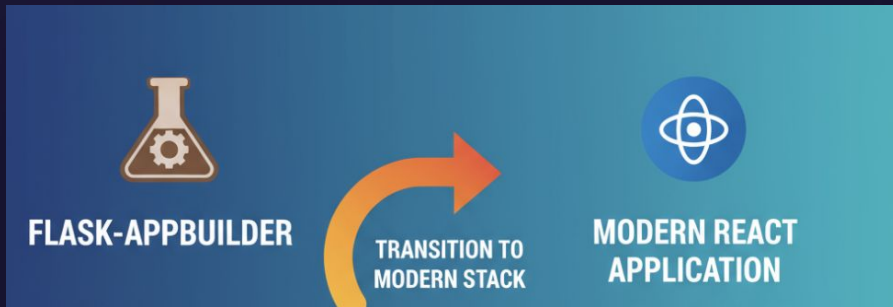


Daily Automated Regression Testing





Complete UI Overhaul Testing



- Every Screen Manually Validated
- Core views
- Admin functionality
- Critical user journeys



Airflow Migration Tools

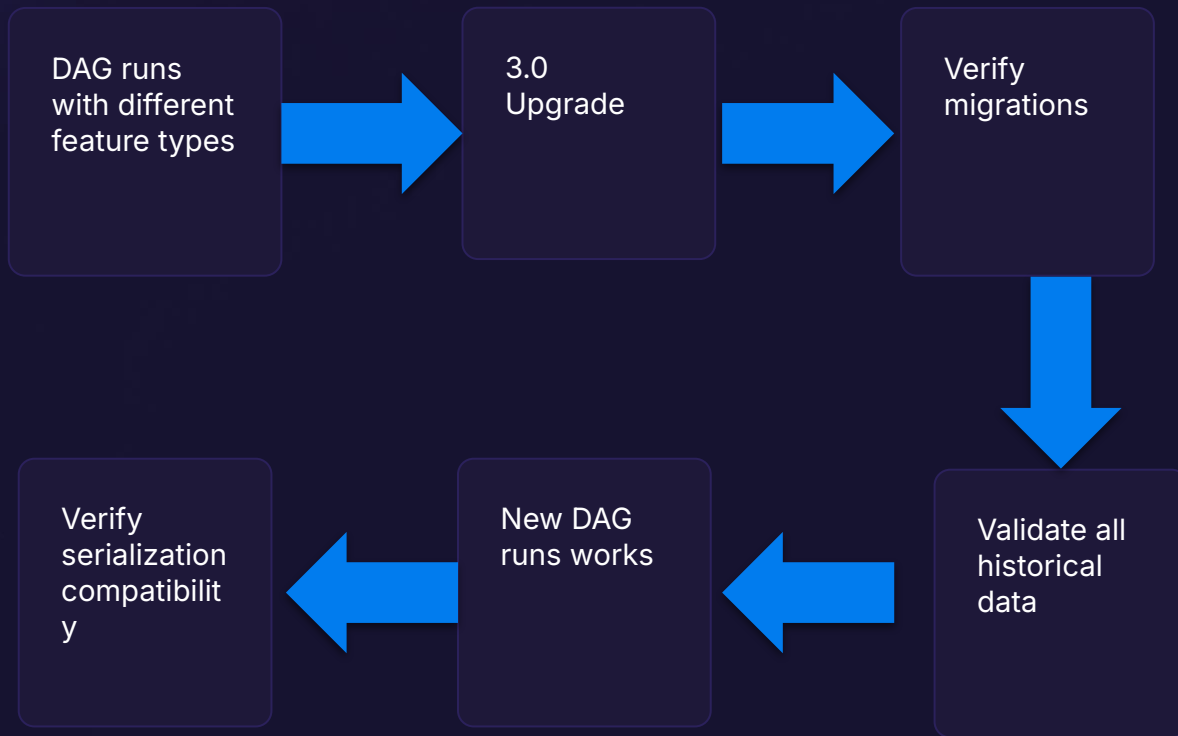
- *airflow db migrate* - To update airflow metadata database
- *airflow config lint* - lint options for the config changes
- *airflow config update* - update options for the config changes in `airflow.cfg`

How to update your DAG Code ?

[Ruff](#) rules to rescue



Migration Testing - The Critical Path





Comprehensive AIP Validation

Testing All Major Changes from Our Test Plan

[Test Plan](#)

- AIP-65/66: DAG Versioning & Bundles
- AIP-74/75: Data Assets & Asset-Centric Syntax
- AIP-78: Scheduler-managed backfill
- AIP-38 Modern Web Application
- AIP-83 Remove Execution Date Unique Constraint from DAG Run
- AIP-84 UI REST API



Testing Infrastructure & Community

- Consistent testing environment powered by Breeze
- Astronomer Nightly deployments run tests daily with different executors
- Unit testing with Airflow CI
- AWS, Google and Astronomer CI pipelines run Providers System tests



Community Collaboration



Regular Dev Calls

- Global participation
- Unified approach
- Knowledge sharing



Systematic Confidence Tracking

- AIP-specific confidence levels
- Regression confidence monitoring
- [Test Plan](#)



Results of Community Collaboration

- Faster issue resolution
- Better test coverage
- Shared ownership
- Higher quality



Key Learnings - The Testing Pyramid Works

- Unit + Integration + Manual + End-to-End testing
- Architectural Changes Need Special Attention
- Migration Testing is Make-or-Break
- Community Collaboration Multiplies Impact
- Automation + Manual = Complete Coverage: Neither approach alone is sufficient



Next Steps...

- Airflow E2E Testing framework
- Task SDK Integration Testing framework
- Improve UI Test Coverage



Questions?



2025 Apache Airflow Survey

Fill it out for a free Airflow 3 fundamentals
certification
(or)
DAG Authoring in Airflow 3 certification
code

