Powering Astro Observe with Clickhouse

Julian LaNeve, Christine Shen
julian@astronomer.io, christine.shen@astronomer.io
2024.12.09

ASTRONOMER

Agenda

About Astronomer & Airflow

Observe: Airflow Observability

How Clickhouse Fits In

Apache Airflow 101

Apache Airflow is an open source tool for **programmatically** authoring, scheduling and monitoring your data pipelines.

With over **30M downloads per month**, Airflow is the most popular open source choice for workflow orchestration around the world.

- Pipelines as code, written in Python
- Highly extensible
- Infinitely scalable
- Large, vibrant OSS community

Apache Airflow Use Cases

Data-Powered Applications

Expose dynamic datasets in a user-facing product

Critical Operational Processes

Generate & file critical regulatory reports

Analytics & Reporting

Power dashboards that influence decision-making

ML Ops

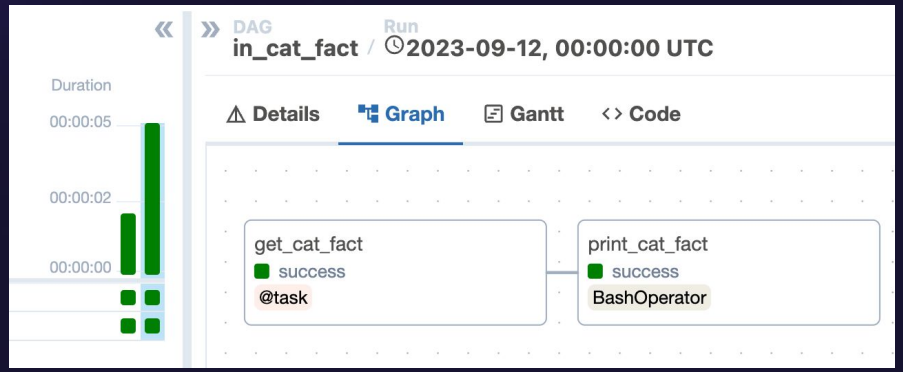
Operationalize homegrown models for differentiated product experiences

Gen AI

Leverage foundational models to drive domain-specific outcomes

Basically... any batch process with data!

Writing a DAG



Pipeline in Airflow = DAG

A DAG contains tasks.

Tasks are defined in Python using operator classes and/or decorators.

dags > in_cat_fact.py > ...

```python
from airflow.decorators import dag, task
from airflow.operators.bash import BashOperator
from pendulum import datetime
import requests


@dag(
    start_date=datetime(2023, 8, 1),
    schedule="@daily",
    catchup=False,
)
def in_cat_fact():
    @task
    def get_cat_fact():
        r = requests.get("https://catfact.ninja/fact")
        return r.json()["fact"]

    get_cat_fact_obj = get_cat_fact()

    print_cat_fact = BashOperator(
        task_id="print_cat_fact",
        bash_command=f'echo "{get_cat_fact_obj}"',
    )

    get_cat_fact_obj >> print_cat_fact


in_cat_fact()
```

About Astronomer

Our Company

Astronomer empowers data teams to bring mission-critical software, analytics, and AI to life and is the company behind Astro, the industry-leading data orchestration and observability platform powered by Apache Airflow®. Astro accelerates building reliable data products that unlock insights, unleash AI value, and powers data-driven applications.

600+ customers | 250 employees | 5 Global Offices

Powering businesses from startups to Fortune 100s

The standard for data pipelines in a cloud-native world

30M
Monthly Downloads

3K
Contributors

35K
GitHub Stars

53K
Slack Community

ASTRONOMER

The driving force behind Apache Airflow

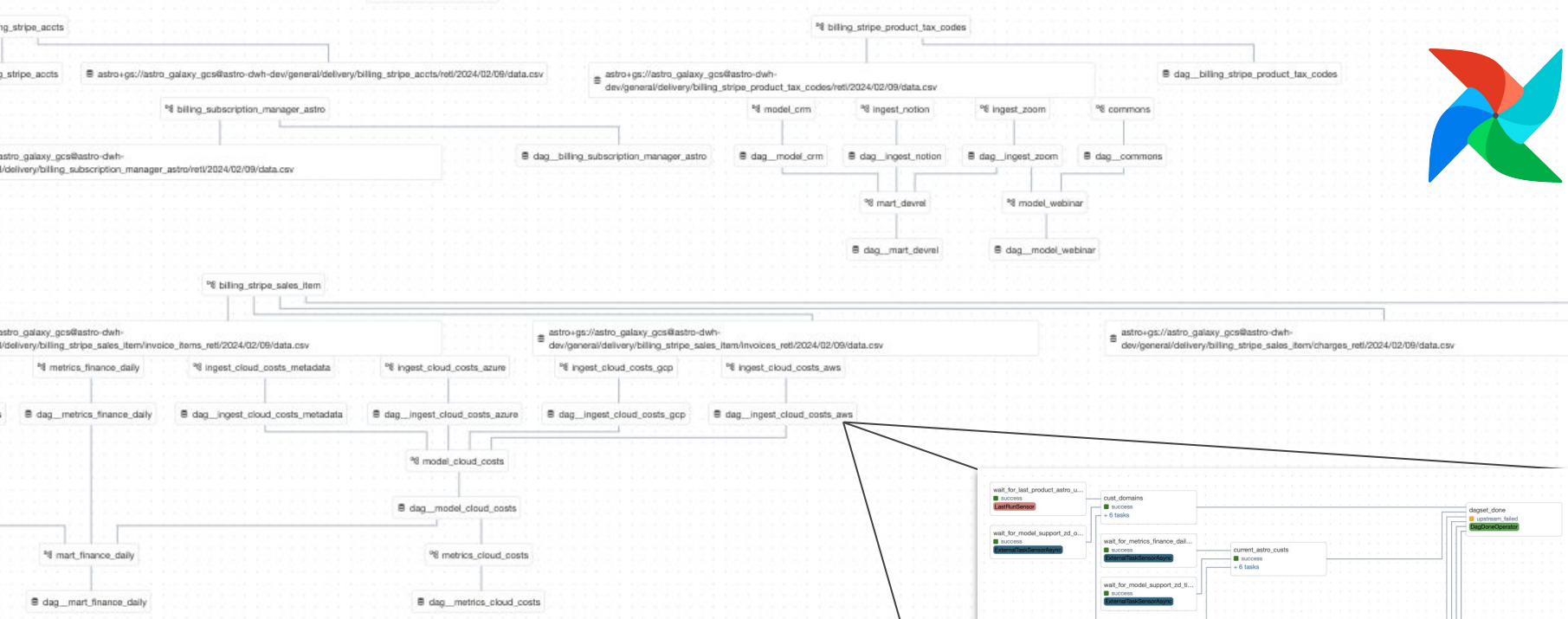
5 offices | 250 employees | 24×7 worldwide support

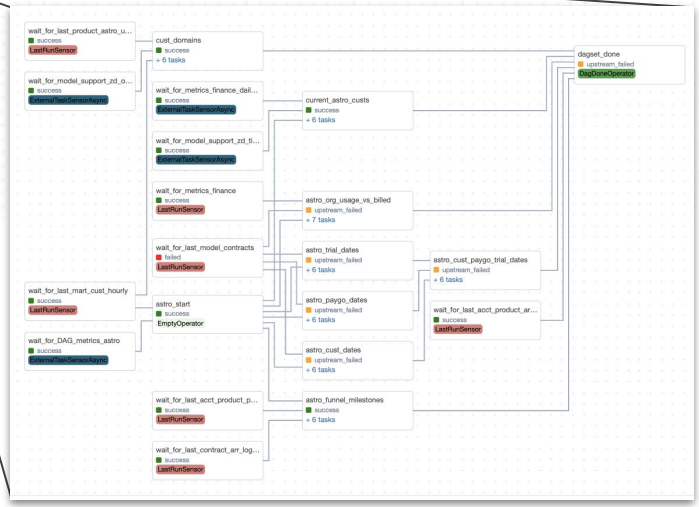
100%
Drives 100% of Airflow releases

55%
Of Airflow code contributed

18 of 25
18 of the top 25 committers on board, 8 PMC members

30K+
30K+ Airflow students in Academy ecosystem





For a typical organization, Airflow runs complex networks of thousands of pipelines (DAGs)

Thousands of teams
rely on us to run
millions of DAGs
and **billions** of tasks

OOM error

Database Error in
model marketing_funnel

invalid
identifier

But sometimes DAGs **fail**

INFO - Marking
task as FAILED

The command returned
a non-zero exit code

Role does not
exist

OOM error

Database Error in
model marketing_funnel

On one of our platforms, in a single month, we ran 18,000,000 DAGs and 100,000,000 tasks.

Of those, **1.5%** had errors.

INFO - Marking
task as FAILED

invalid
identifier

On average, it takes **1.24 hours** for users to resolve errors.

And each user spends **26 hours a month** fixing DAG failures

The command returned
a non-zero exit code

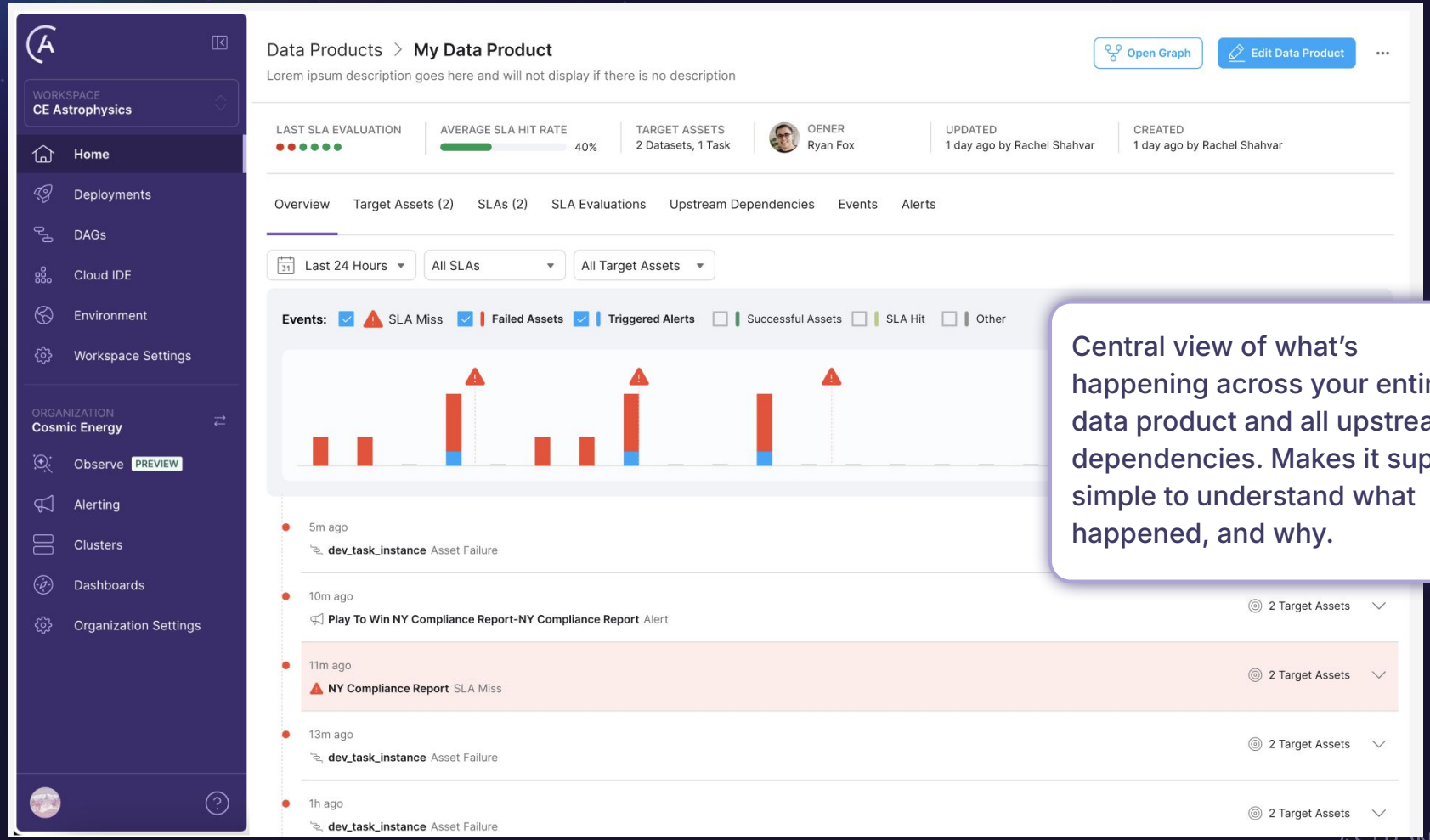
Role does not
exist

There’s always a set of questions you want answered...

- What failed?
- When did it fail?
- Why did it fail?
- What’s the impact of the failure?
- Did anything else fail?

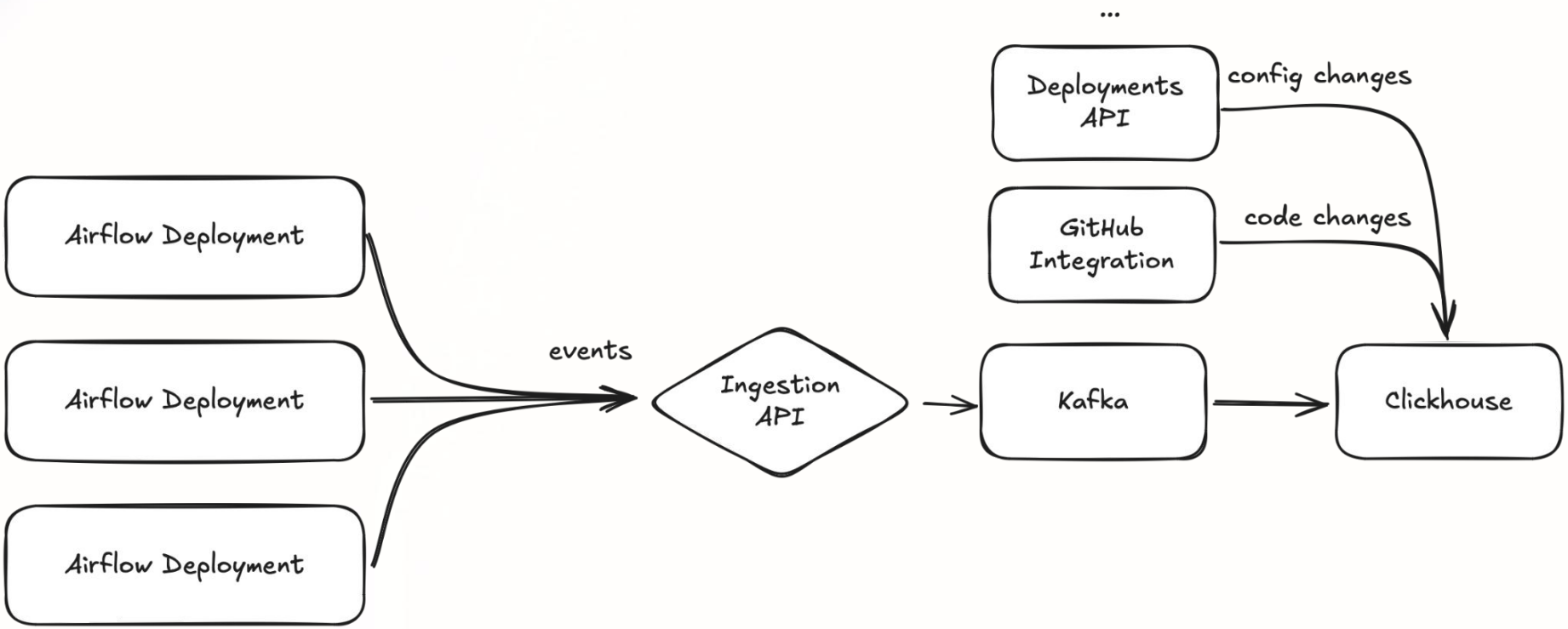
Astro Observe

ASTRONOMER



Central view of what's happening across your entire data product and all upstream dependencies. Makes it super simple to understand what happened, and why.

Architecture



Clickhouse Schema

An “events” table for each event type:

run_start_events
run_complete_events
dataset_events
deploy_events
alert_events
…

run_events RMT

Engine: ReplacingMergeTree

timestamp	DateTime
org_id	String
namespace	String
workspace_id	Nullable(String)
deployment_id	Nullable(String)
dag_id	Nullable(String)
task_id	Nullable(String)
asset_id	String
run_id	String
system_run_id	String
parent_run_id	Nullable(String)
duration	Nullable(Int32)
state	String
type	String
metadata	String

Common fields

Event-specific fields

run_events

Create query | Insert row

#	timestamp	org_id	namespace	workspace_id	deployment_id	dag_id	task_id	asset_id	run_id	system_run_id	parent_run_id	duration	state	type	metadata
1	2024-12-09 0…	cllbl63cv000…	combusting-f…	cm01fgftu004…	clk25k3j5413…	annual-tax-r…	extract_fina…	combusting-f…	01927310-7b6…	scheduled__2…	0192ddc3-0ac…	12	success	airflow_task	{ "conn_id…

Querying the Data

```sql
SELECT
  workspace_id,
  deployment_id,
  ...,
  map('state', state, 'type', type, ...) AS metadata
FROM run_events

UNION ALL

SELECT ...
FROM dataset_events

WHERE org_id = 'cuid...'
AND timestamp BETWEEN '2024-12-01' AND '2024-12-09'
AND ...
```

Benefits to Clickhouse

Things that made our life much easier compared to other solutions.

Scalability

Easily scales by adding new tables without needing to update existing schemas.

Automated Data Retention

Automatically manages data expiration with Time-to-Live (TTL), ensuring efficient storage usage.

Data Deduplication

Table engines automatically eliminate duplicate records, optimizing storage and query performance.

Materialized Views

Precomputed results accelerates complex queries and faster access to transformed data.

Powering Astro Observe with Clickhouse

Julian LaNeve, Christine Shen
julian@astronomer.io, christine.shen@astronomer.io
2024.12.09

ASTRONOMER

ASTRONOMER