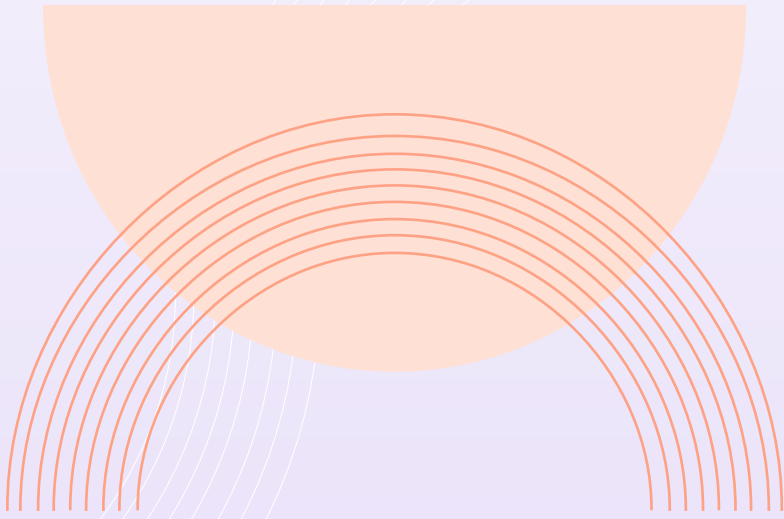




Migrating from Oozie to Astro

Astro is the managed Airflow service engineered for the cloud and backed by technical support and services from the leading committers to Apache Airflow.

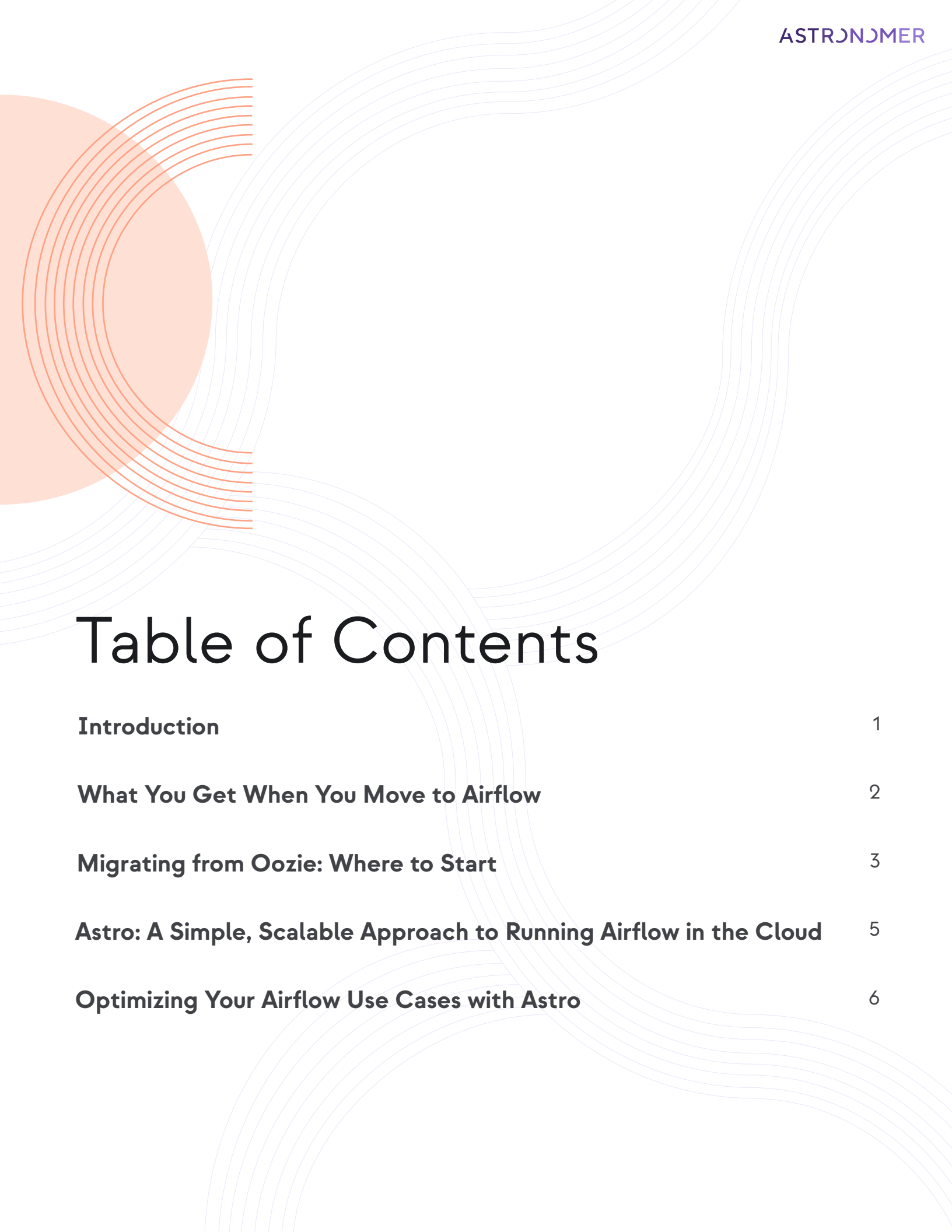


Table of Contents

Introduction	1
What You Get When You Move to Airflow	2
Migrating from Oozie: Where to Start	3
Astro: A Simple, Scalable Approach to Running Airflow in the Cloud	5
Optimizing Your Airflow Use Cases with Astro	6

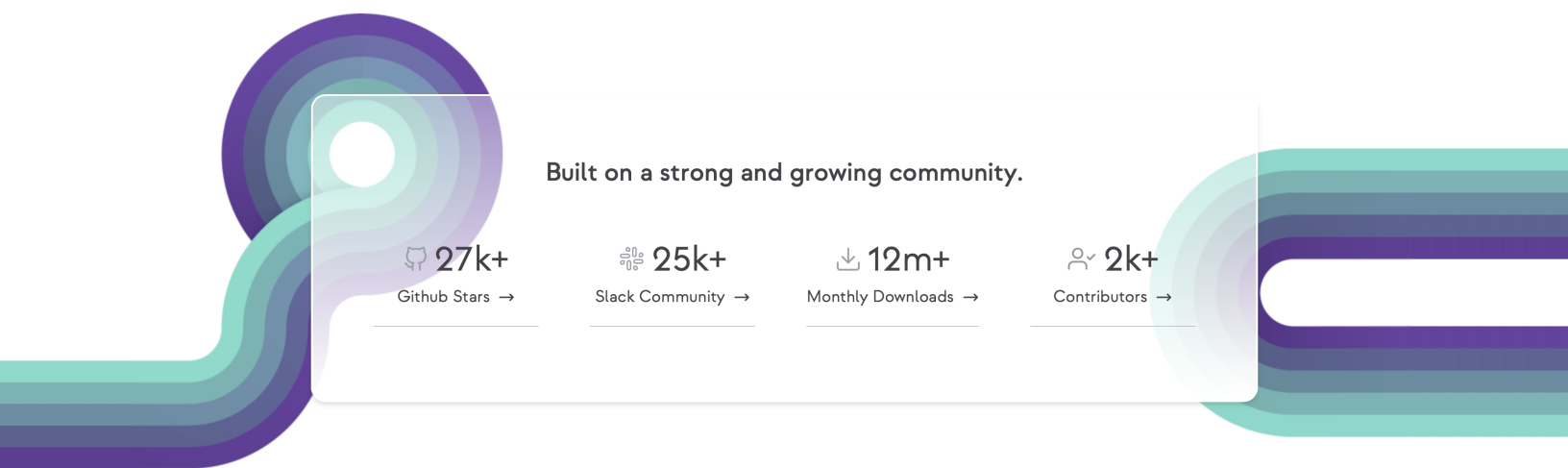
Airflow is the de facto standard for expressing data flows as code – and the right tool for orchestrating data in the cloud.

In many cases, when organizations migrate from Oozie it's because they realize Oozie is no longer a sustainable or optimal way for their team to run their workloads. The cost of maintaining legacy infrastructure has led to reduced developer productivity, or Oozie failures are resulting in data outages that impact the business, or teams' lack of currency with the technology means they're working at a disadvantage.

That's where Apache Airflow comes in. Airflow is the open source standard for workflow orchestration, offering a flexible and scalable way to programmatically author, schedule, and monitor your data pipelines using Python and SQL.

Created at Airbnb in 2014, Airflow was brought into the Apache Software Foundation's Incubator Program in 2016 and announced as a Top-Level Apache Project in 2019. Now, it's recognized as the industry's leading data orchestration solution.

Airflow is downloaded more than 12 million times every month and is used by hundreds of thousands of teams around the world. With more than 140 integrations, and more added by its vibrant community each month, Airflow offers the most comprehensive coverage of data sources and other providers, and is well suited to virtually any use case.



Built on a strong and growing community.

 27k+

[Github Stars →](#)

 25k+

[Slack Community →](#)

 12m+

[Monthly Downloads →](#)

 2k+

[Contributors →](#)

Workflow Scheduling and So Much More: What You Get When You Move to Airflow

Airflow supports any use case you're currently fulfilling with Oozie, and it also dramatically improves your experience building, running, and observing data pipelines. The following Airflow features and functionality will transform the developer experience and set your organization up to take advantage of modern data orchestration.

Integrate the Modern Data Stack

Organizations use Airflow to run and manage legacy ETL and ELT workflows, along with the new ephemeral ETL/ELT patterns enabled by cloud-native computing and the modern data stack. These organizations count on Airflow to orchestrate the batch ETL or ELT dataflows that populate their cloud data warehouses, data lakes, and/or data lakehouses, as well as feed batches of data to their fraud-detection alerts, clickstream analytics, and other time-critical services.

Understand and Manage the State of Your Pipelines with the Airflow UI

The [Airflow UI](#) offers extensive functionality to assist your data teams in understanding the state of their pipelines. The grid view was recently developed by Airflow committers at Astronomer as a compact, intuitive way to visualize complex representations in Airflow's UI — and to quickly surface useful metrics and information about what's happening inside of Airflow. The grid view is the centerpiece of Astronomer's ongoing effort to give users a single place to see what's happening with their tasks and directed acyclic graphs (DAGs).

Improve Efficiency and Timeliness with Data-Driven Scheduling

Data-driven scheduling equips Airflow with the built-in logic it needs to run and manage different kinds of data-driven dependencies, through a new object type called a [Dataset](#). A Dataset is any output from one or more upstream ("producer") tasks that's used as a source of data for one or more downstream DAGs, called "consumers." Airflow now "knows" to run consumer DAGs automatically once the Datasets they depend on are updated, making it much easier to design for common types of data engineering patterns.

Improve Reliability and Timeliness by Breaking Monolithic Pipelines into Micropipelines

Large monolithic pipelines are difficult to maintain and enhance: even the smallest change in format of an incoming data feed, for example, causes the entire pipeline to be updated and redeployed. Moreover, monolithic pipelines often pose a critical challenge to enterprises that need business insights to be available in a timely manner. Now, with the release of [Apache Airflow 2.4](#), monolithic pipelines can be decomposed into smaller, loosely coupled components – or micropipelines – which can then be orchestrated together. It's a transformative development, made possible by the introduction of Datasets as a top-level concept.

Enable More Data Practitioners with Python Pipeline Authoring

The [Astro Python SDK](#) is an open source Python framework for writing data pipelines that run on Airflow. To use it, you need only learn seven Python functions and two Python classes. All data practitioners will find DAG authoring with the SDK easier than with standard Airflow. For example, data scientists can write not only data transformations in Airflow, but also their own extract-and-load operations. In other words, they don't need to wait for spare cycles from a data engineer to load data into their warehouse and get started on writing transformations.

Migrating from Oozie: Where to Start

The idea of a migration might seem daunting, but rest assured, Astronomer has helped companies of all sizes and levels of complexity to migrate off of Oozie to our managed Airflow service, Astro. In this section, we share our migration playbook and key considerations.

First, it's important to plan for your migration. Astronomer recommends taking an iterative approach. A few things to keep in mind include workflow count, workflow complexity, business priority, and team resourcing (availability and capacity to execute the migration).

Once you've made your plan, work iteratively to perform the migration through three main steps:

1. Code Translation and Migration from XML to Airflow

Code migration necessitates turning your Oozie workflows into Airflow workflows. This means trading in XML for Python or SQL. First, we recommend building Oozie-like DAGs so there's less to learn for ongoing operation of your data pipelines.

Start with translating your workflow's Oozie control and action node elements to their corresponding Airflow elements. Airflow doesn't require the majority of Oozie's control node functions because they're baked in or already handled by Airflow. The `start`, `fork`, `join`, and `kill` nodes required in Oozie are handled innately by Airflow. To enable your workflow to make a selection on the execution path to follow, as handled by a `decision` node in Oozie, use the `PythonBranchOperator` in Airflow. The end node does not need to be translated in most cases; though if there is a decision node upstream, you can use the `DummyOperator` to signify the end of the workflow.

Most Oozie action nodes can be mapped directly to Airflow Operators, meaning you don't necessarily need to rewrite the core logic of your workflow. However, you may want to rewrite your logic in Python for ease of continuing to build and iterate in Airflow. An operator defines a unit of work for Airflow to complete. For example, the `PigOperator` can be used to execute a Pig script and `BashOperator` can be used where a Pig job is submitted to run a shell script.

Identify the first Oozie workflows to migrate by considering workflow complexity and team resourcing. Workflows that have no inter-workflow dependencies will be easiest to start with. Resourcing is particularly important because a team will need to have cycles to spend on performing the migration and testing the new workflows. If any of your teams have Airflow expertise, start with those groups.

As you're executing the code migration, be sure to document technical debt along the way and add it to your team's backlog.

2. Migrating Property Files

Oozie property files can be translated into Airflow variables and Jinja macros. Once property file functionality is defined in Airflow, for example in a variable, simply go back to the code and reference the new Airflow variable in your workflow. This process is fairly straightforward and can be performed recursively.

3. User Acceptance Testing

A migration is not complete until the workflows are operating as expected and business continuity is ensured. User Acceptance Testing (UAT) should demonstrate that the workflows being tested are operating in a manner suited to real-world circumstances and usage. An ideal scenario for UAT would include a full production mirror for development with an isolated compute infrastructure.

A Simple, Scalable Approach to Running Airflow in the Cloud: What You Get on Astro

Astro is a fully managed data orchestration platform that's powered by Airflow. Here at Astronomer, we've baked our team's experience with the world's largest organizations running Airflow into every corner of the Astro experience. That means if you're just starting out with Airflow, you'll move faster on Astro. And when you're ready to expand Airflow to many teams, Astro removes the complexity of scaling.

With Astro, you're backed by technical support and services from the leading committers to Airflow. You'll get in-place upgrades and same-day support for new versions of Airflow; runtime optimized for cost and performance; enhanced integrations with hundreds of third-party data services and infrastructure tools that remain 100% upstream compliant; and more. Job scheduling through Airflow is a core component of orchestration, but it's far from sufficient. Astro is a data orchestration platform, with Airflow at the center, that makes everything more manageable, accessible, and reliable. At Astronomer, we see the power of an orchestration platform to transform Airflow into an asset that your entire company uses to publish and consume data that drives the critical decisions your business makes. And as more teams are able to join in, the network effects result in a discoverable, well-managed network of data that binds a business together

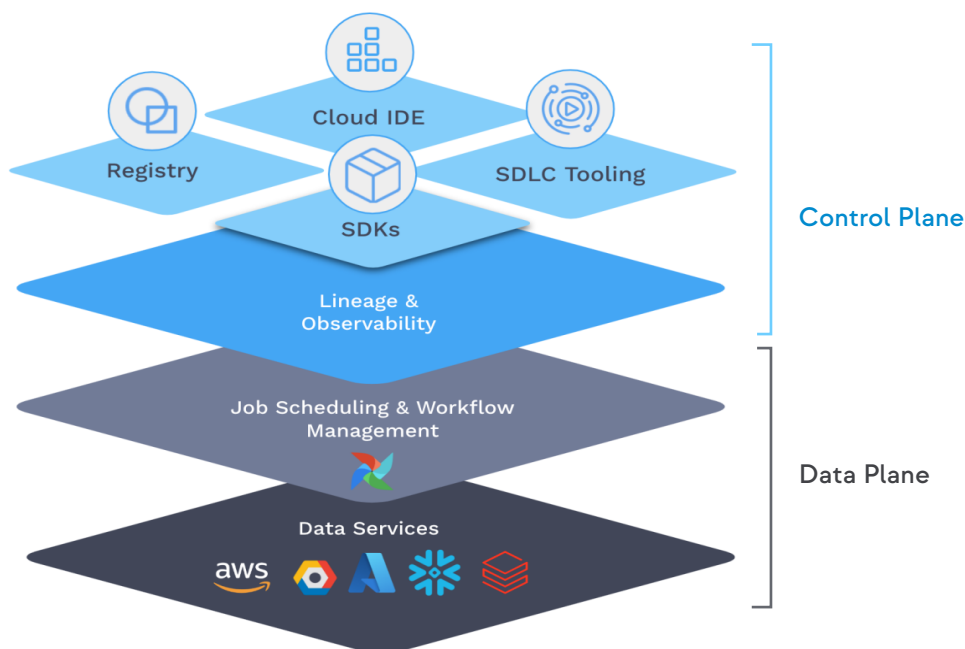


Figure 1: The layers of Astronomer's Airflow-powered data orchestration platform, Astro.

Astro is built with a multi-plane architecture consisting of the Data Plane running Airflow, and a Control Plane. The Control Plane centralizes and dramatically simplifies the operations of managing Airflow deployments: with a UI to bulk-manage all data pipelines, oversee the health of every data project through one dashboard, spin up new deployments at the click of a button, and supervise role-based access for all users.

Optimizing Your Airflow Use Cases with Astro

The Astro Control Plane delivers on the promise of an orchestration platform with its extensive functionality to optimize and augment Airflow use cases.

Minimize Your DevOps Burden to Increase Team Productivity

Fully-managed and SLA-driven support from the Astronomer team minimizes your operational burden. Ongoing support, such as in-place Airflow upgrades, maximize your time spent on innovative work, and not on DevOps.

Optimize Your Workflows

Astro Runtime is tuned for cost and performance by the world's Airflow experts. Smart default settings make it easy for your teams to focus on their workflows, knowing that Astronomer optimizes their projects.

Reduce Data Downtime

Astro comes with built-in features to optimize your Airflow environment, and it offers tunable features to optimize your Worker configurations and avoid failure scenarios from over-utilization of resources

Implement Best Practices with Software Development Lifecycle Tooling

Whether you're checking code into GitHub or committing pipelines developed in the Cloud IDE, CI/CD scripts effectively handle the mapping between feature branches and Airflow deployments (including their associated connections). That means the process of promoting pipelines from development into test or production is automated.

Secure Your Data Workflows in the Cloud

Astronomer offers the highest levels of data security and protection with enterprise-ready compliance. Astro is certified GDPR, CCPA, HIPAA, and PCI-DSS compliant, and it's available on AWS, Google Cloud, and Azure.

Quickly Pinpoint and Remediate Issues

Astro automatically extracts and analyzes lineage metadata, giving organizations a complete view of data lineage from source to target. This lets support personnel quickly pinpoint and resolve data outages, and allows data stewards to monitor and improve data quality. Lineage metadata also gives your organization the information it needs to effectively govern your data, enabling CDOs, CAOs, and others to identify silos, protect sensitive data, and comply with regulatory requirements, as well as promote the responsible reuse of data.

Explore Data Lineage and Maintain Observability

Astronomer has developed a rich set of capabilities around data lineage and observability, targeting both DataOps users who want to find issues and patterns en masse to individual data practitioners who need to troubleshoot pipeline issues.

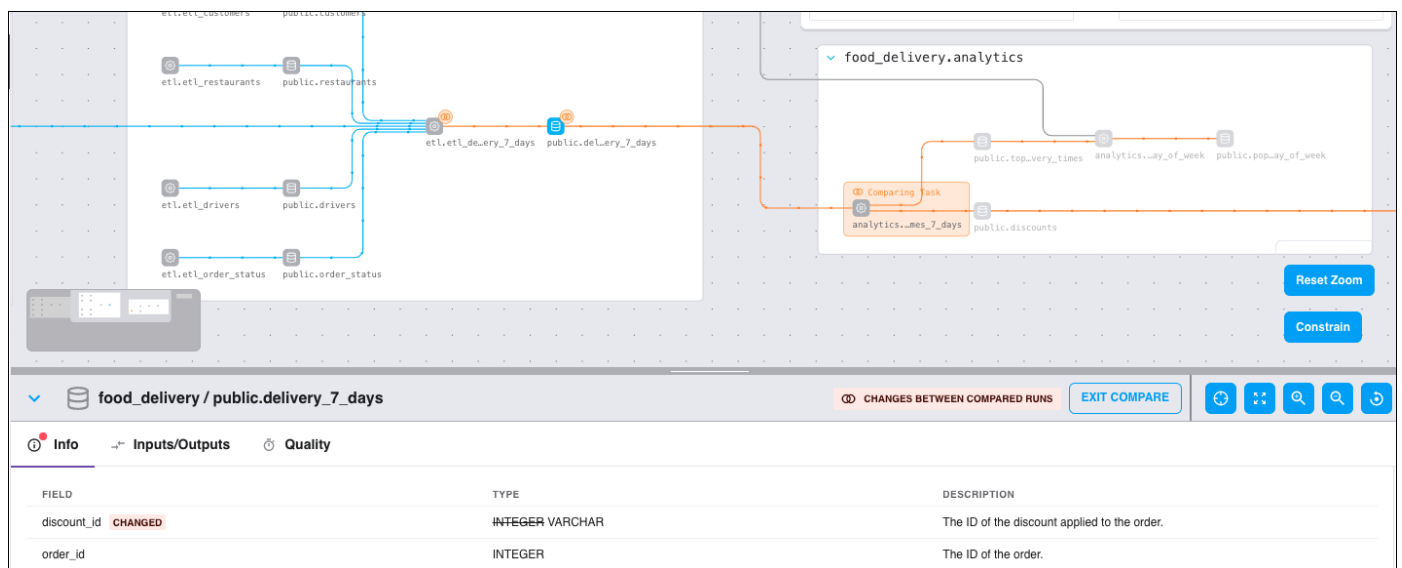


Figure 2: A Lineage view in Astro, comparing lineage graphs from previous runs.

Pipeline Authoring Made Easier with the Cloud Integrated Development Environment (IDE)

Astronomer's Cloud IDE provides an intuitive interface in the Astro UI that treats vanilla Python, SQL, and Bash as first-class citizens. The [Cloud IDE](#) lets you easily implement pre-configured connections, and abstracts virtual environment management from the pipeline authoring process. The Cloud IDE makes it dead-simple for anyone new to Airflow and Astronomer to write their first pipeline.

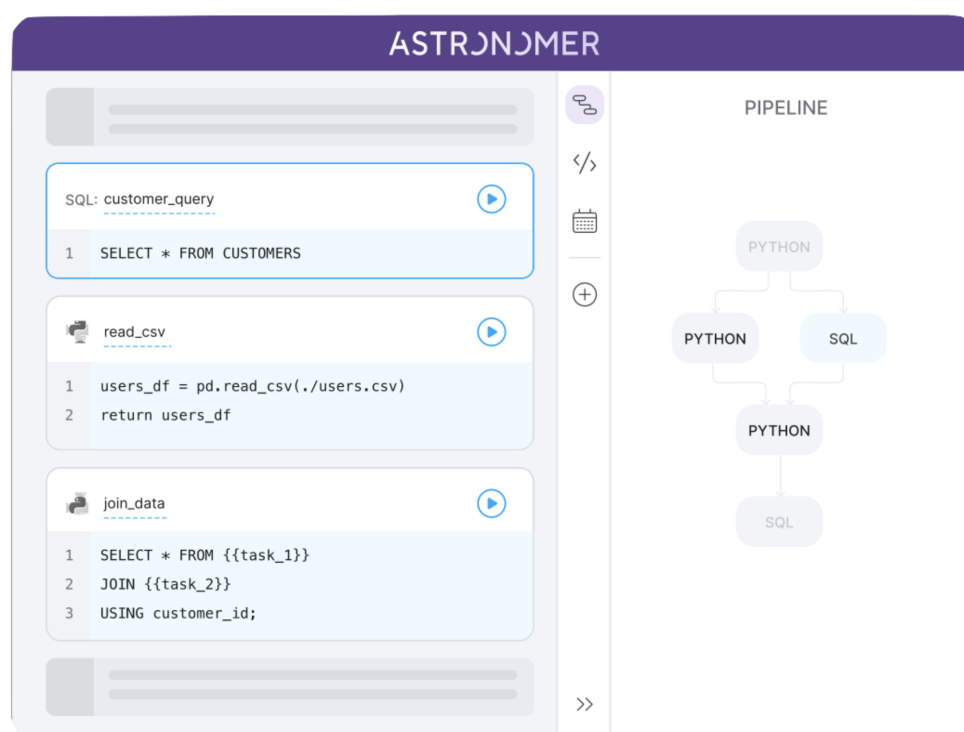


Figure 3: A visual representation of the Astro Cloud IDE.

Support and Guidance from Astronomer Professional Services

Astronomer Professional Services offers technical and strategic support to make your Airflow migration easy, built on experience migrating the largest enterprises off of Oozie. Wherever you are in your journey to Airflow, Astronomer Professional Services provides technical and strategic expertise to accelerate your progress.

Migrate from Oozie to Airflow in weeks, not years.

Astro removes the boundaries of a distributed data ecosystem — and it's all backed by technical support and services from the leading committers to Airflow.

With Astro you can focus on building your business with the power of data, not managing infrastructure.

Get Started with Astro today with a [demo](#).