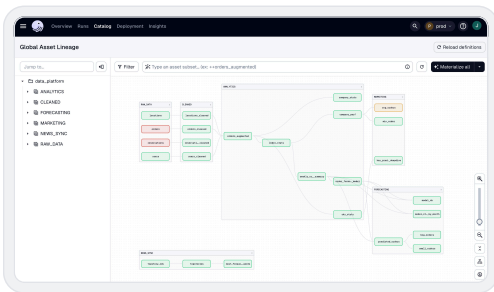


Comparing the open-source solutions

Dagster and Apache Airflow are the two most popular generalized orchestrators used in data engineering today. Both are open-source solutions. For organizations looking to outsource the hosting and infrastructure concerns, the two leading solutions are Dagster+ by Dagster Labs and Astro by Astronomer.

Dagster vs. Airflow

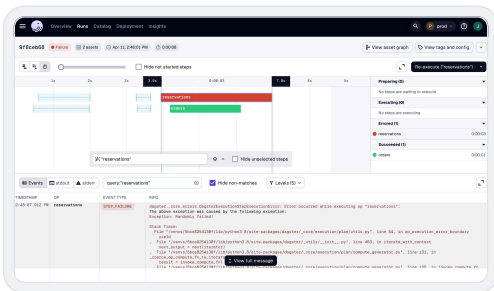
Before we consider the commercial hosting options, what truly separates these solutions are the core principles behind the designs.



Asset-oriented vs. workflow-oriented development

Dagster's asset-orientation makes it a central system of record for how your assets are defined and produced. Your team can collaborate on delivering critical data assets, not on the tasks of pipelines.

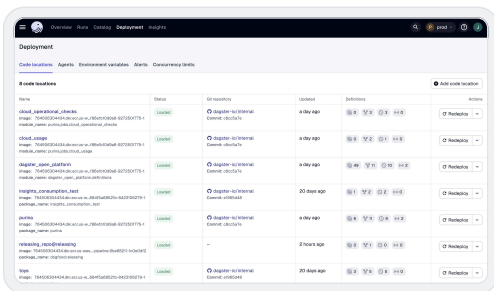
Airflow is workflow-oriented and task-centric. It does not provide asset-aware features or a coherent Python API. It is typically implemented after pipelines have been designed to trigger the required tasks.



Testing and debugging

Dagster is designed for use at every stage of the data development lifecycle. The open-source solution alone facilitates local development, unit testing, CI, code review, staging, and debugging. Dagster+ further expands these capabilities.

Airflow pipelines are harder to test and review outside of production deployments. Many teams working on Airflow end up doing their final testing in production. Here Astronomer helps provide options missing in the open-source solution.



Cloud-native infrastructure

Dagster is cloud- and container-native, and designed for today's data infrastructure (ECS, K8s, Docker). Dependencies are easy to manage and upgrades are smooth. Dagster+ provides a turnkey hosting solution.

Isolating dependencies and provisioning infrastructure with Airflow is more complex and time-consuming than Dagster. Astronomer is a popular option for running Airflow as it helps mitigate these challenges.

ASTRONOMER



Core focus	Workflow orchestration	Data orchestration
Primary building block	Tasks	Assets
Local development support	Yes	Yes+
CI/CD support and dev branching	Yes	Yes+
Environment management	Yes	Yes
dbt support	Yes	Yes+
Alerting	Yes	Yes+
Partitioned data support	Basic	Yes+
Backfills	Basic	Yes+
Embedded ELT support	Yes	Yes+
Sensors isolated from runtime	No	Yes
Cross-team collaboration	Basic	Yes
Data catalog	No	Yes
Column-level lineage	No	Yes
Data quality (assets check)	No	Yes
Automatic updates on freshness checks	Basic	Yes+
Estimating credit usage	Yes+	Yes
Operational observability (including costs)	Yes	Yes+
Community size	Very large	Large