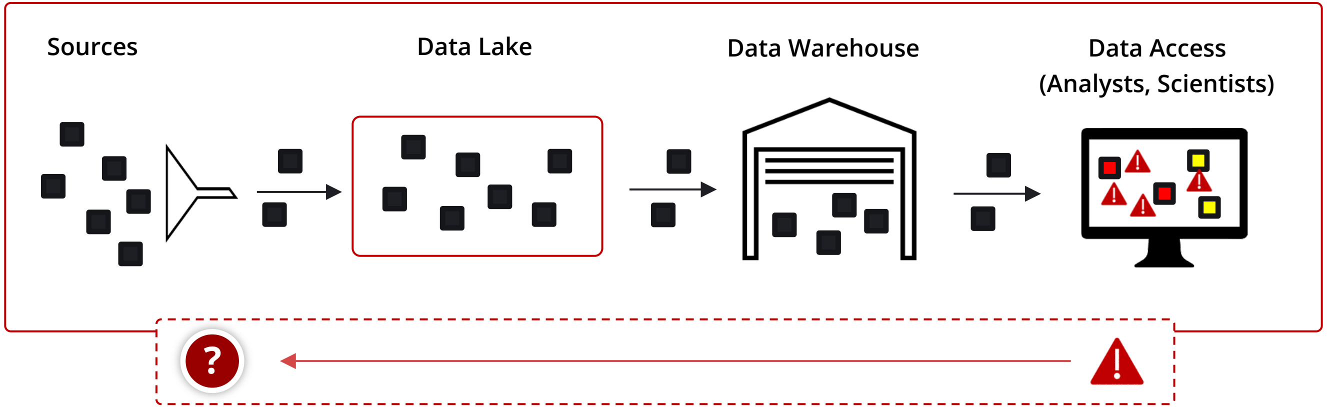
Data Observability

Technical Overview

What is Data Observability?

Problem Statement

Data engineers are **reactive** to data issues



Sources

Data Lake

Data Warehouse

Data Access
(Analysts, Scientists)

Many data quality issues are **overlooked**

Platform only learns about issues when **reported by data consumers**

After issues are reported, they **are not resolved quickly**

Databand

The root causes

Fragmented toolchain

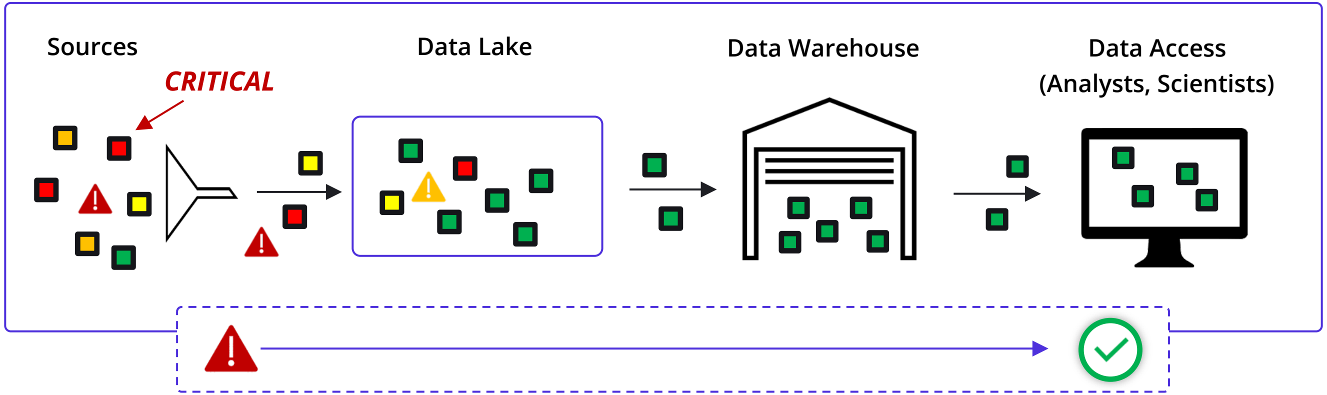
Volume of data

Flooding of noise

Our solution focuses on observing **data in motion**

Solution

Proactive Data Observability: ***Shift left*** *and solve problems at the source*



Improve **MTTD**

Discover issues in real time, early as ingest

Improve **MTTR**

Identify the cause of issues instantly

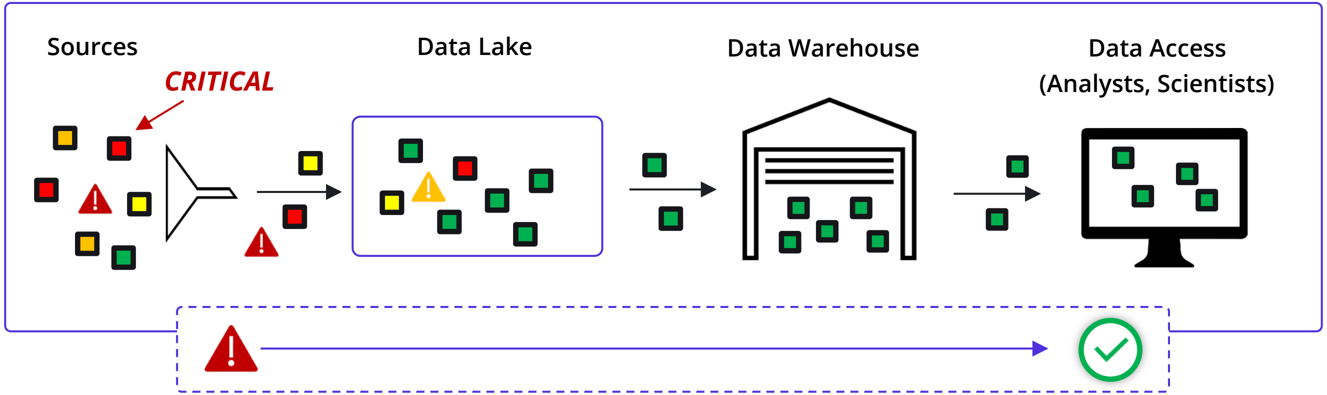
Improve data product **quality**

Enhance trust and consumer satisfaction

Databand focuses on observing **data in motion**

Solution

Proactive Data Observability: ***Shift left*** *and solve problems at the source*



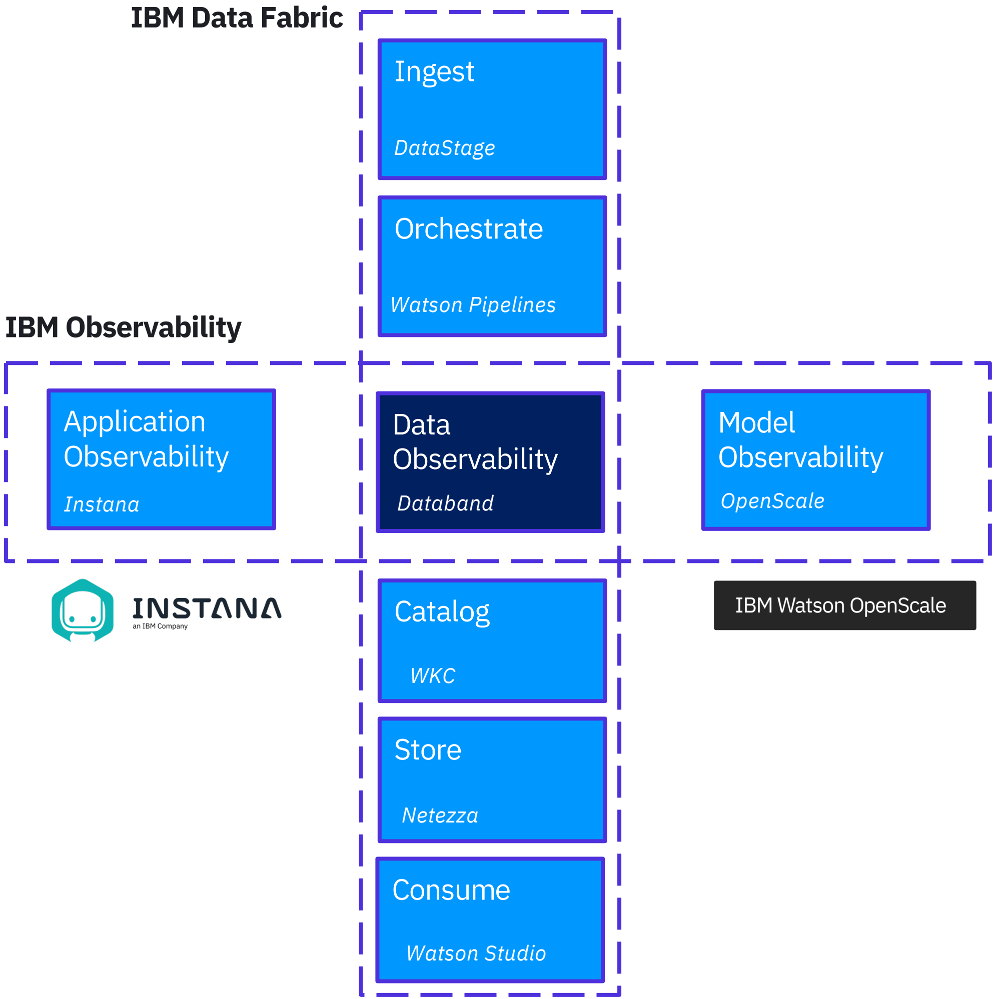
Observe data pipeline **process quality**
- Status
- Performance
- Latency

Observe **data quality and reliability**
- *Schema changes*
- *Data shape*
- *Data freshness*

Supported **data pipelines** and **workflow managers**
- *Airflow*: Python, Spark, dbt, SQL and other operators

The Databand bridges **two** of IBM’s strategic directions, **IBM Data Fabric** and **IBM Observability**.

Bridging these strategies unlocks powerful new use cases for customers and growth for IBM.



IBM Data Fabric

Ingest
DataStage

Orchestrate
Watson Pipelines

IBM Observability

Application Observability
Instana

Data Observability
Databand

Model Observability
OpenScale

INSTANA
an IBM Company

IBM Watson OpenScale

Catalog
WKC

Store
Netezza

Consume
Watson Studio

Databand integration

Pipeline implementation and deployment	Databand integration	Additional details
Spark	No code, optional SDK	Configuration in a Spark cluster – Databand provides a listener
Airflow (all pipelines)	No code for pipeline status, SDK for dataset monitoring	
Python, PySpark, Java, Scala without an workflow engine	SDK for pipeline status and dataset monitoring	
dbt	No code, optional SDK	Syncer for dbt Cloud for a no-code integration. Python SDK can be used to retrieve information about a specific job run
DataStage * (Q4)	No code	Supported for *DataStage Next Gen*

Data Pipelines and Workflow managers

Internal enablement

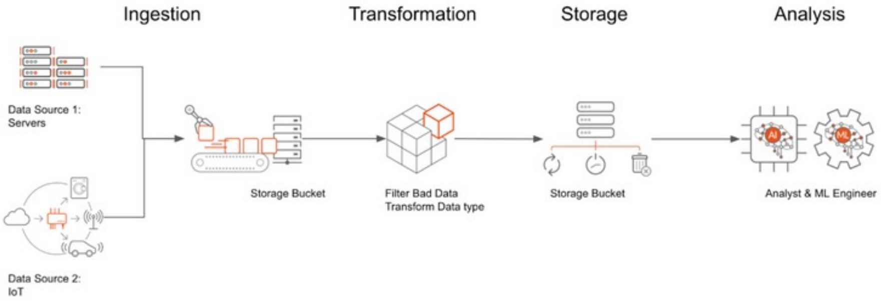
Data pipelines

- *Data pipeline* is a generic term that describes the process of moving data between data sources

    - While in most cases data pipelines performing ETL tasks, a data pipelines can move data without transformations

- Data pipelines can be implemented in a variety of programing languages, technologies, and tools

    1. Languages: *Python, Java, Scala, SQL*
    2. Technologies: *Spark*
    3. Tools: *dbt, DataStage, Azure Data Factory, AWS Data Pipeline, and others*



Workflow engines

- *Workflow* engines are used to orchestrate execution of tasks
    - Many types of workflows are supported by generic *operators*, not just ETL
        - Examples of workflows: ETL, MLOps, DevOps
- Examples of workflow engines
    - *Airflow, Azkaban, MLFlow, Kubeflow, Luigi,* and others
    - *Airflow* is one of the first and one of the most generic workflow engines

Apache Airflow

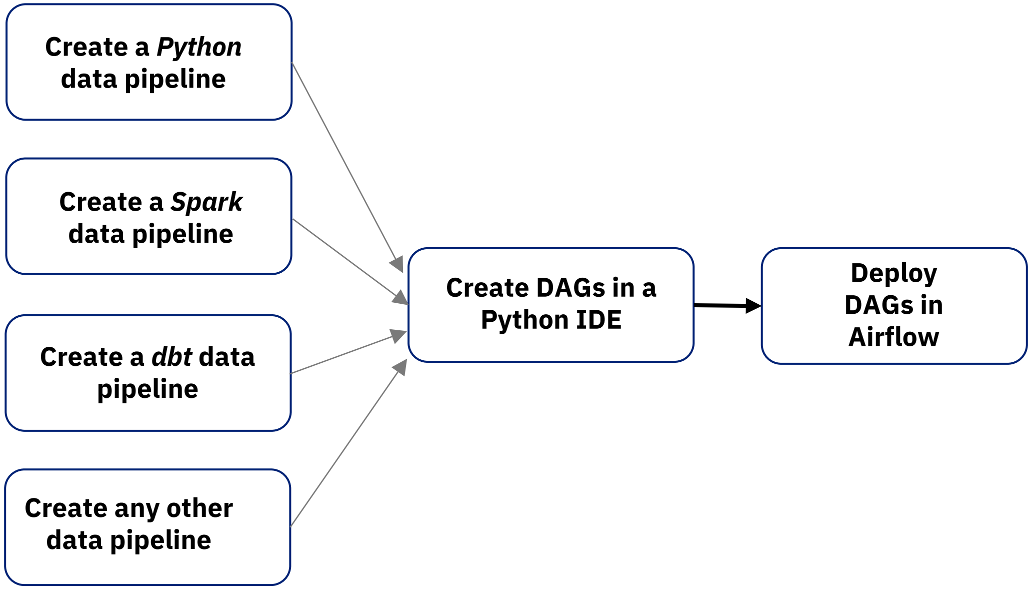
- *Airflow* is one of the first (open source) and one of the most generic workflow engines

Apache Airflow

- Important concepts
  - *DAGs (*Directed Acyclic Graph)
    - A collection of the tasks in a job with relationships and dependencies
    - A DAG is defined in a Python script
  - *Operators*
    - Pre-built functions for frequently used tasks: Python, bash, SQL, and others
  - *Admin console*
    - Understand the features of the admin console

For Databand integration, all DAGS that we are discussing contain are **data pipelines** *(and not other types of pipelines)*

Development and deployment of data pipelines



Create a *Python* data pipeline

Create a *Spark* data pipeline

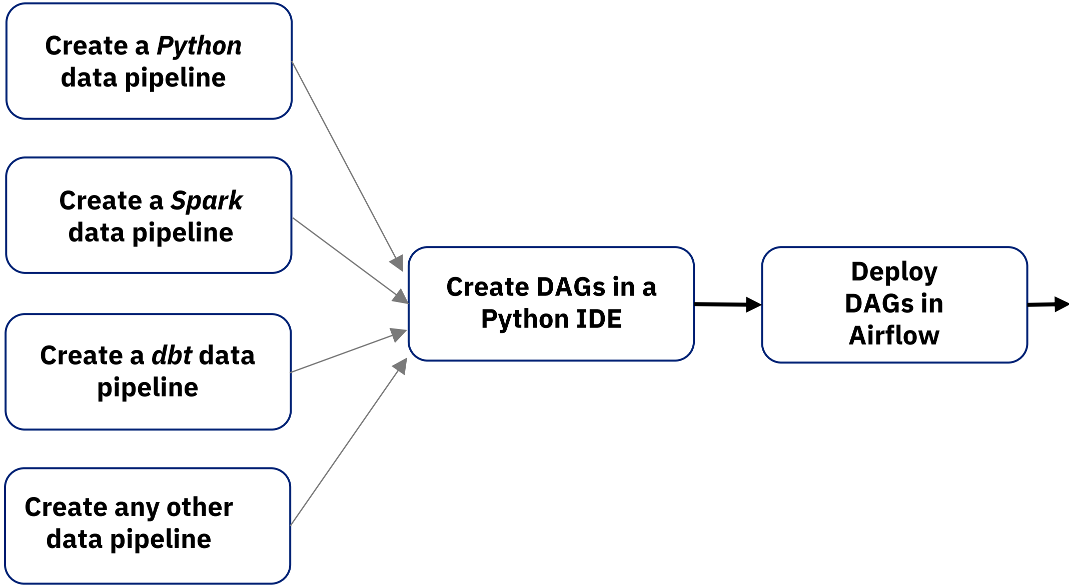
Create a *dbt* data pipeline

Create any other data pipeline

Create DAGs in a Python IDE

Deploy DAGs in Airflow

Development and deployment of data pipelines



Create a *Python* data pipeline

Create a *Spark* data pipeline

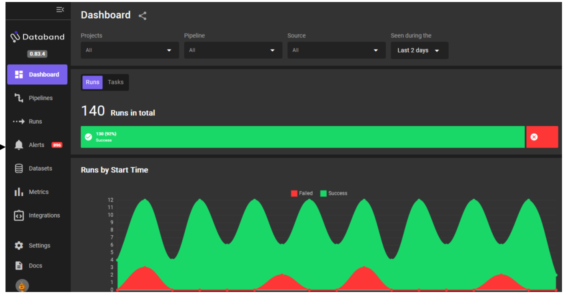
Create a *dbt* data pipeline

Create any other data pipeline

Create DAGs in a Python IDE

Deploy DAGs in Airflow

Databand



Dashboard

Projects
All

Pipeline
All

Source
All

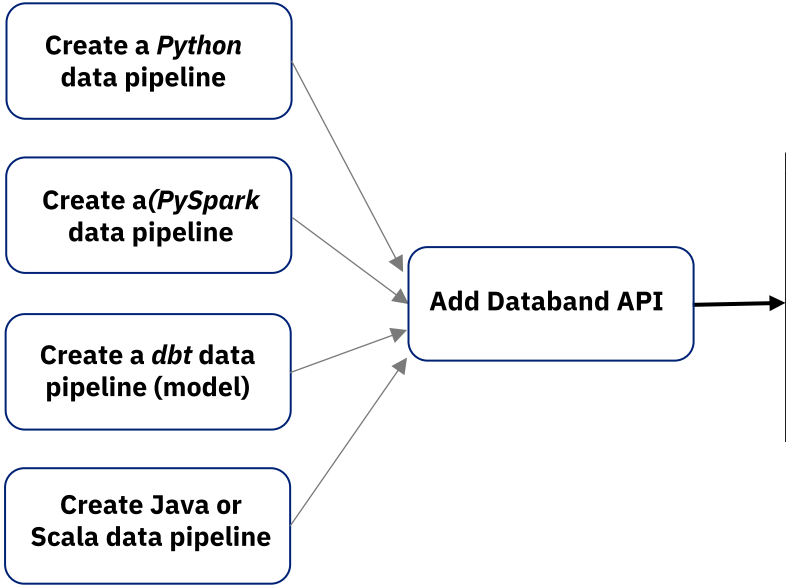
Seen during the
Last 2 days

140 Runs in total

130 (92%)
Success

Runs by Start Time

Development and deployment of data pipelines



Create a *Python* data pipeline

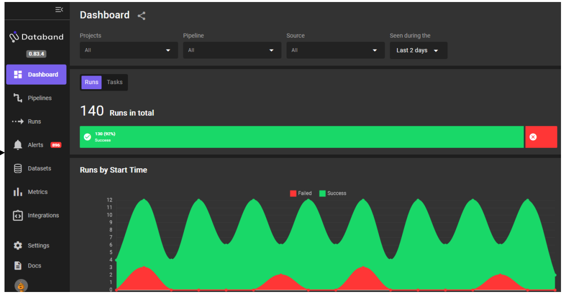
Create a*(PySpark* data pipeline

Create a *dbt* data pipeline (model)

Create Java or Scala data pipeline

Add Databand API

Databand



Dashboard

Projects: All

Pipeline: All

Source: All

Seen during the: Last 2 days

140 Runs in total

130 (92%) Success

Runs by Start Time

Development and deployment of data pipelines

1

Create a *Python* data pipeline

Create a *Spark* data pipeline

Create a *dbt* data pipeline

Create any other data pipeline

Develop

2

Create DAGs in a Python IDE → **Deploy DAGs in Airflow**

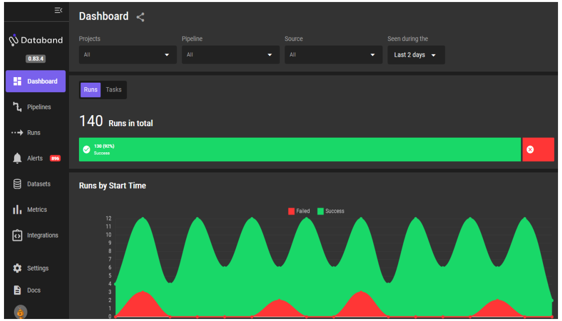
and/or

Add Databand API

Develop

3

Databand



Dashboard

Projects: All

Pipeline: All

Source: All

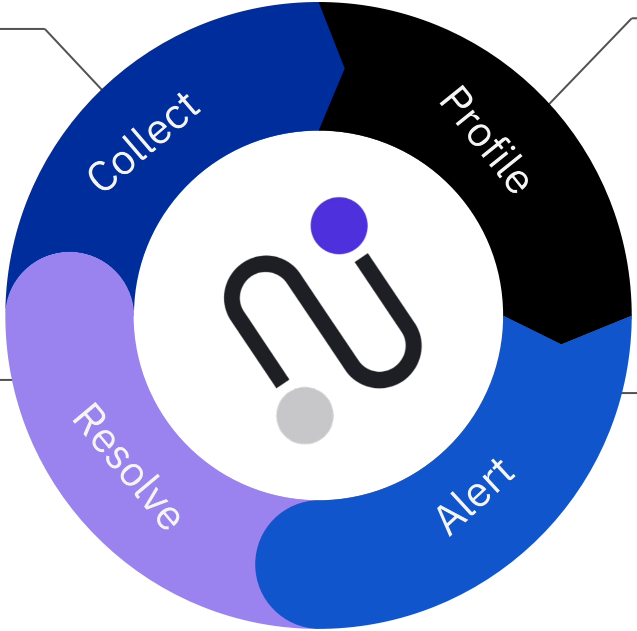
Seen during the: Last 2 days

140 Runs in total

130 (92%) Success

Monitor

The Databand.ai solution

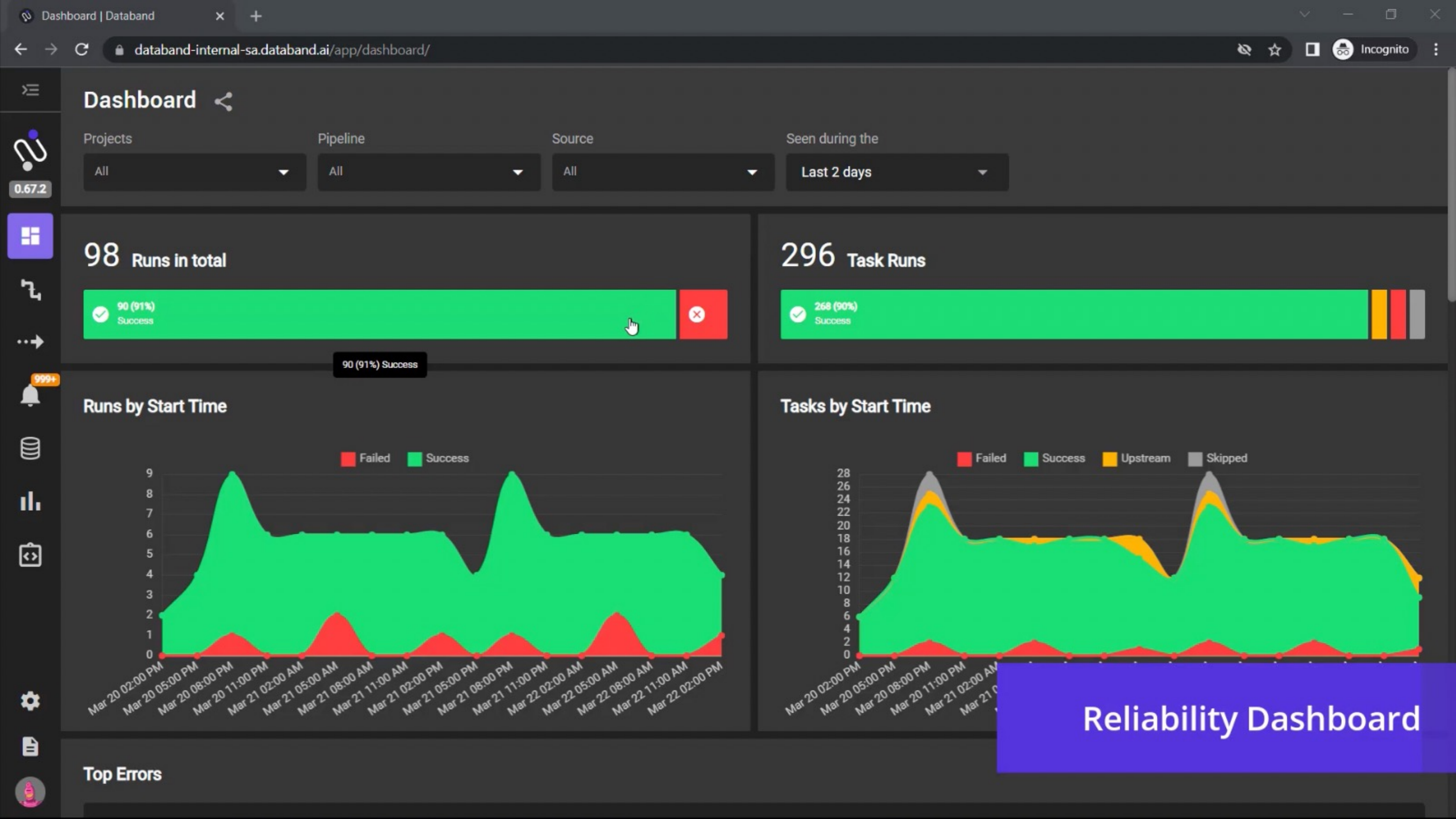


1
Automatically collect metadata
From key solutions in the modern data stack.

2
Build historical baseline
Based on common data pipeline behavior.

3
Alert on anomalies and rules
Based on deviations or breaches.

4
Resolve through automation
Create smart workflows to remediate data quality issues and keep SLAs on track.



Dashboard

Projects: All

Pipeline: All

Source: All

Seen during the: Last 2 days

98 Runs in total

90 (91%) Success

90 (91%) Success

296 Task Runs

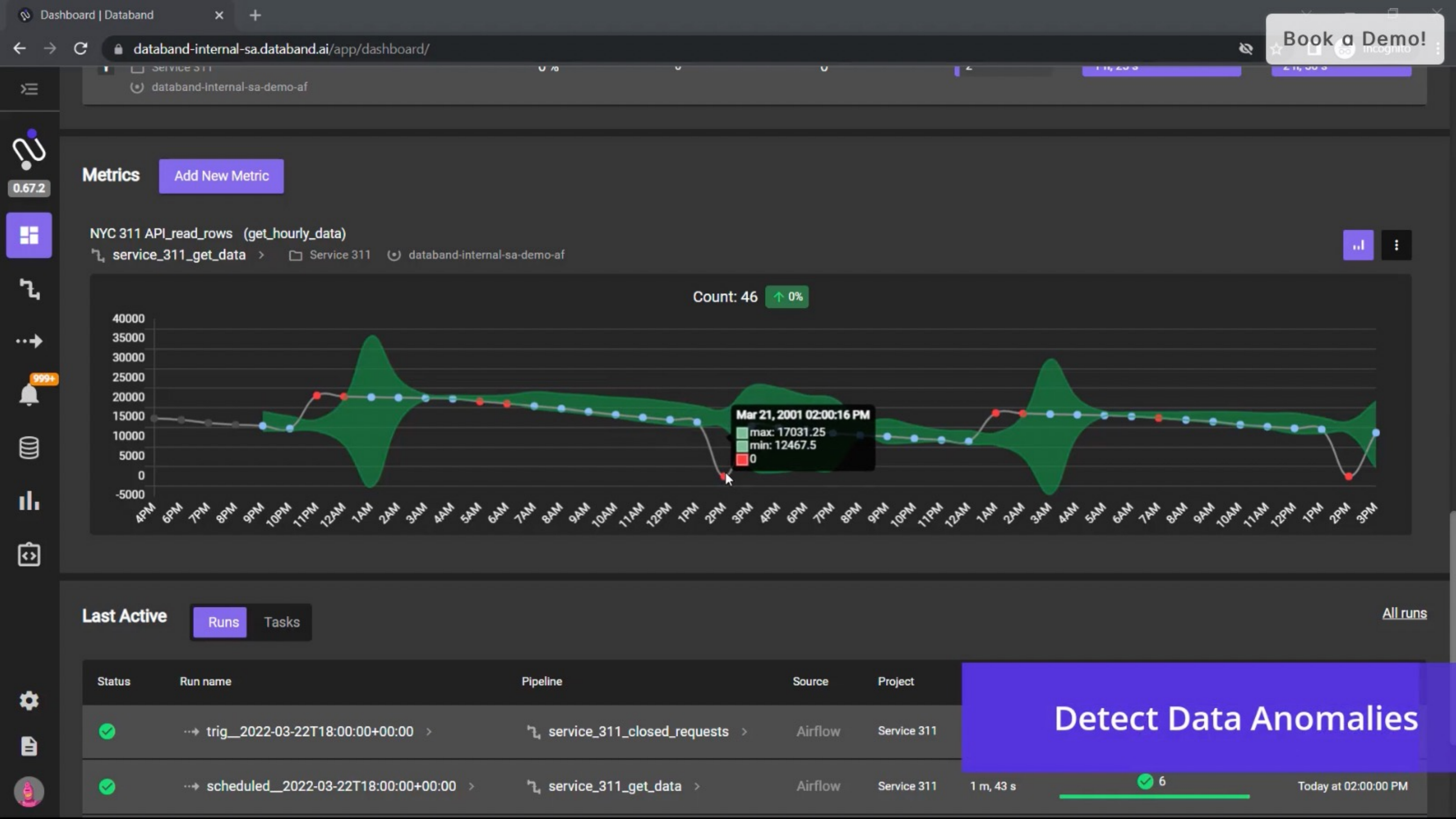
268 (90%) Success

Runs by Start Time

Tasks by Start Time

Reliability Dashboard

Top Errors



Metrics

Add New Metric

NYC 311 API_read_rows (get_hourly_data)

service_311_get_data › Service 311 databand-internal-sa-demo-af

Mar 21, 2001 02:00:16 PM
🟩 max: 17031.25
🟩 min: 12467.5
🟥 0

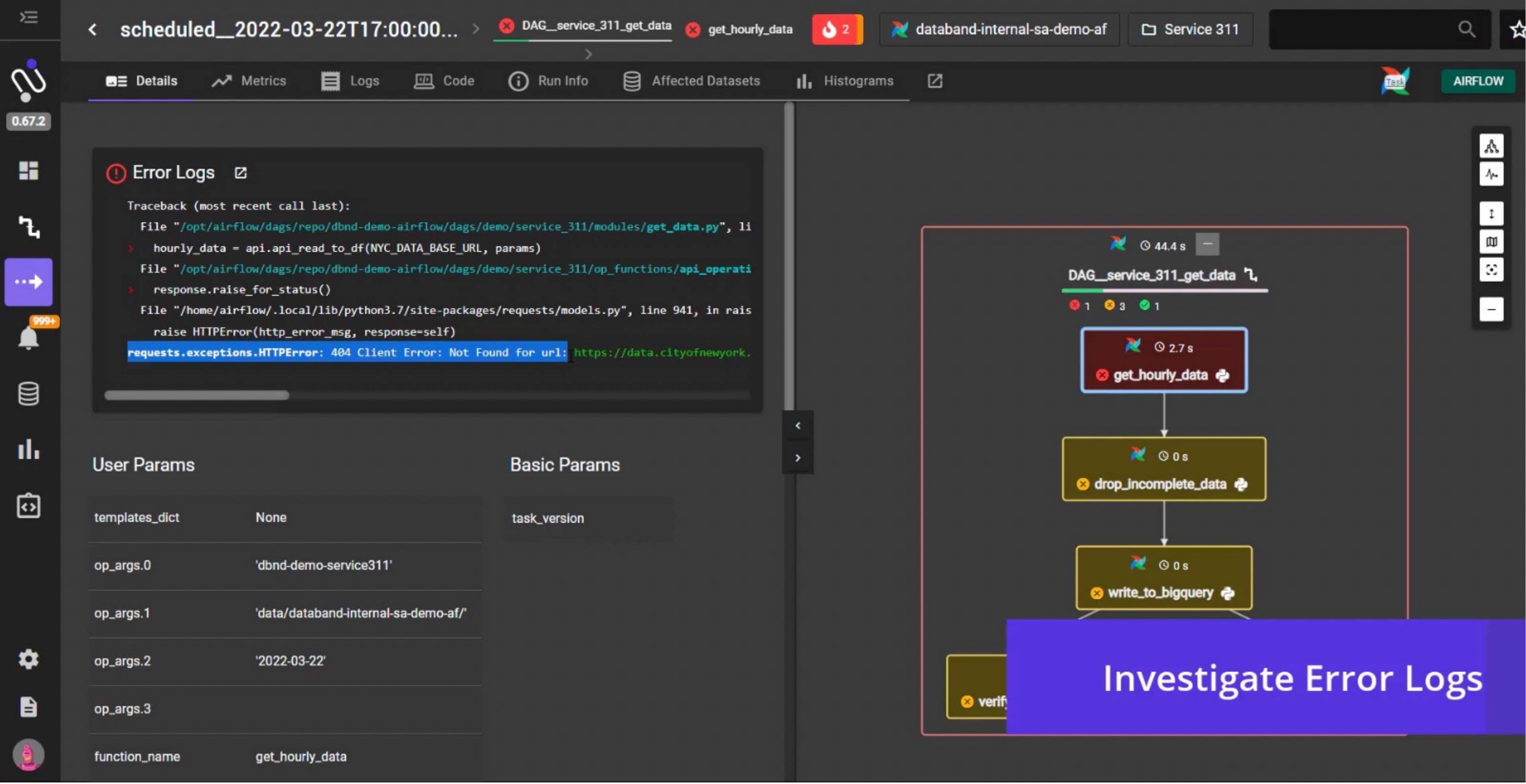
Last Active

Runs Tasks

All runs

Status	Run name	Pipeline	Source	Project			
✓	trig__2022-03-22T18:00:00+00:00 ›	service_311_closed_requests ›	Airflow	Service 311			
✓	scheduled__2022-03-22T18:00:00+00:00 ›	service_311_get_data ›	Airflow	Service 311	1 m, 43 s	✓ 6	Today at 02:00:00 PM

Detect Data Anomalies



Error Logs

```
Traceback (most recent call last):
  File "/opt/airflow/dags/repo/dbnd-demo-airflow/dags/demo/service_311/modules/get_data.py", li
    hourly_data = api.api_read_to_df(NYC_DATA_BASE_URL, params)
  File "/opt/airflow/dags/repo/dbnd-demo-airflow/dags/demo/service_311/op_functions/api_operati
    response.raise_for_status()
  File "/home/airflow/.local/lib/python3.7/site-packages/requests/models.py", line 941, in rais
    raise HTTPError(http_error_msg, response=self)
requests.exceptions.HTTPError: 404 Client Error: Not Found for url: https://data.cityofnewyork.
```

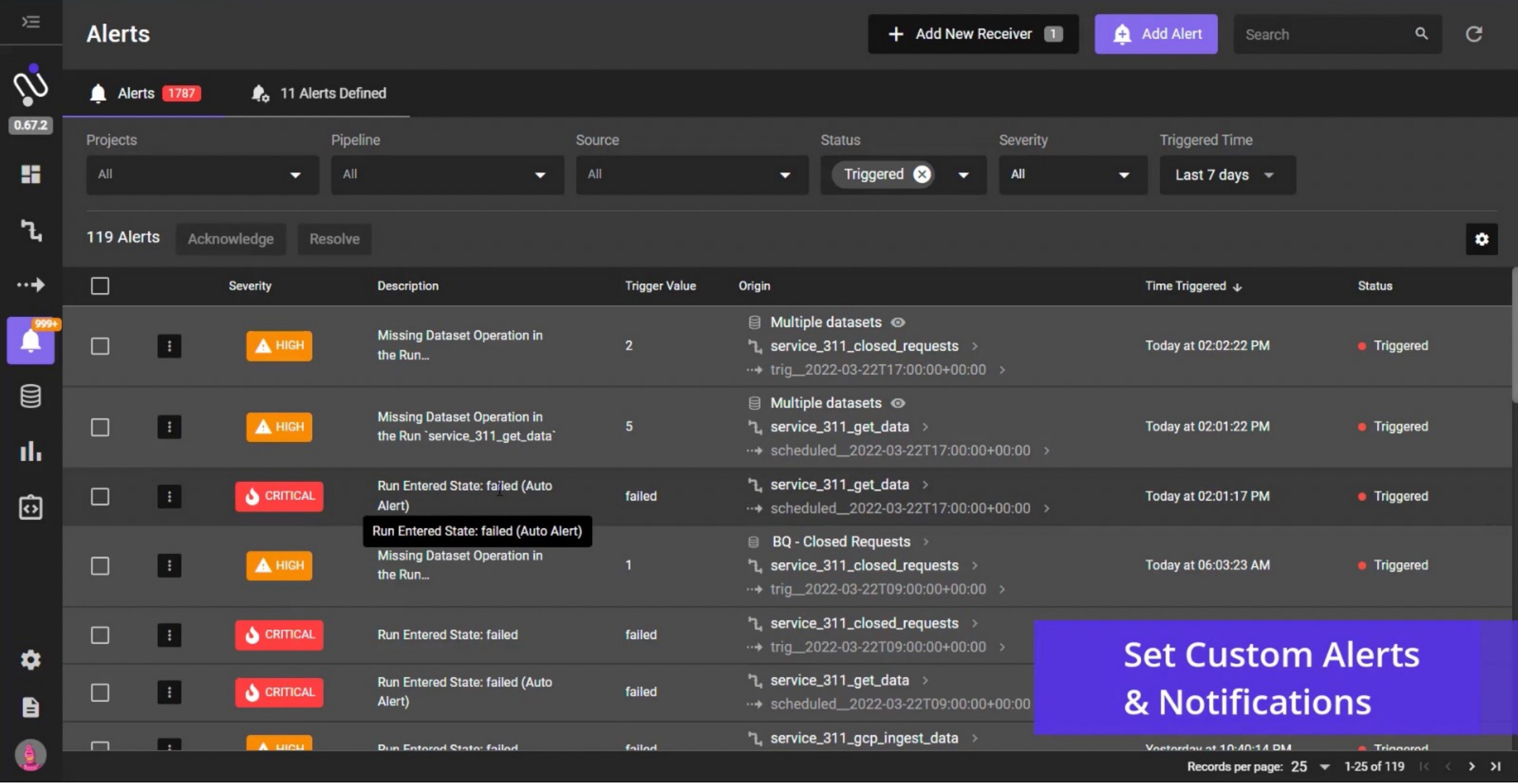
User Params

templates_dict	None
op_args.0	'dbnd-demo-service311'
op_args.1	'data/databand-internal-sa-demo-af/'
op_args.2	'2022-03-22'
op_args.3	
function_name	get_hourly_data

Basic Params

task_version

Investigate Error Logs



Alerts 1787 | 11 Alerts Defined

Projects: All
Pipeline: All
Source: All
Status: Triggered
Severity: All
Triggered Time: Last 7 days

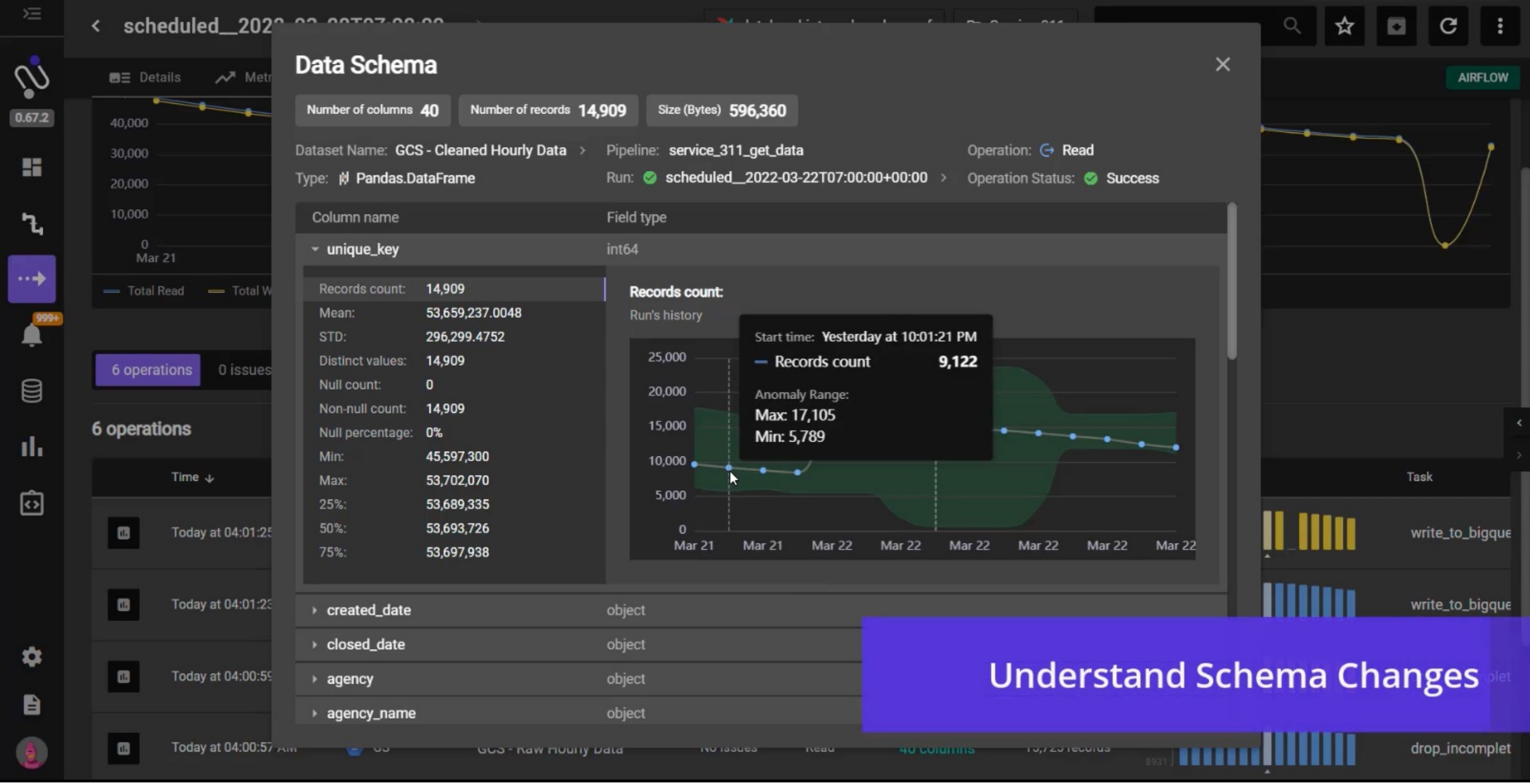
119 Alerts Acknowledge Resolve

□		Severity	Description	Trigger Value	Origin	Time Triggered ↓	Status
□	⋮	HIGH	Missing Dataset Operation in the Run...	2	Multiple datasets service_311_closed_requests › trig__2022-03-22T17:00:00+00:00 ›	Today at 02:02:22 PM	● Triggered
□	⋮	HIGH	Missing Dataset Operation in the Run `service_311_get_data`	5	Multiple datasets service_311_get_data › scheduled__2022-03-22T17:00:00+00:00 ›	Today at 02:01:22 PM	● Triggered
□	⋮	CRITICAL	Run Entered State: failed (Auto Alert)	failed	service_311_get_data › scheduled__2022-03-22T17:00:00+00:00 ›	Today at 02:01:17 PM	● Triggered
□	⋮	HIGH	Missing Dataset Operation in the Run...	1	BQ - Closed Requests › service_311_closed_requests › trig__2022-03-22T09:00:00+00:00 ›	Today at 06:03:23 AM	● Triggered
□	⋮	CRITICAL	Run Entered State: failed	failed	service_311_closed_requests › trig__2022-03-22T09:00:00+00:00 ›		
□	⋮	CRITICAL	Run Entered State: failed (Auto Alert)	failed	service_311_get_data › scheduled__2022-03-22T09:00:00+00:00		
□	⋮	HIGH	Run Entered State: failed	failed	service_311_gcp_ingest_data ›	Yesterday at 10:40:14 PM	● Triggered

Run Entered State: failed (Auto Alert)

Records per page: 25 1-25 of 119

Set Custom Alerts & Notifications



Data Schema

Number of columns 40 | Number of records 14,909 | Size (Bytes) 596,360

Dataset Name: GCS - Cleaned Hourly Data

Type: Pandas.DataFrame

Pipeline: service_311_get_data

Run: scheduled__2022-03-22T07:00:00+00:00

Operation: Read

Operation Status: Success

Column name	Field type
unique_key	int64
created_date	object
closed_date	object
agency	object
agency_name	object

Records count:	14,909
Mean:	53,659,237.0048
STD:	296,299.4752
Distinct values:	14,909
Null count:	0
Non-null count:	14,909
Null percentage:	0%
Min:	45,597,300
Max:	53,702,070
25%:	53,689,335
50%:	53,693,726
75%:	53,697,938

Records count:

Run's history

Start time: Yesterday at 10:01:21 PM

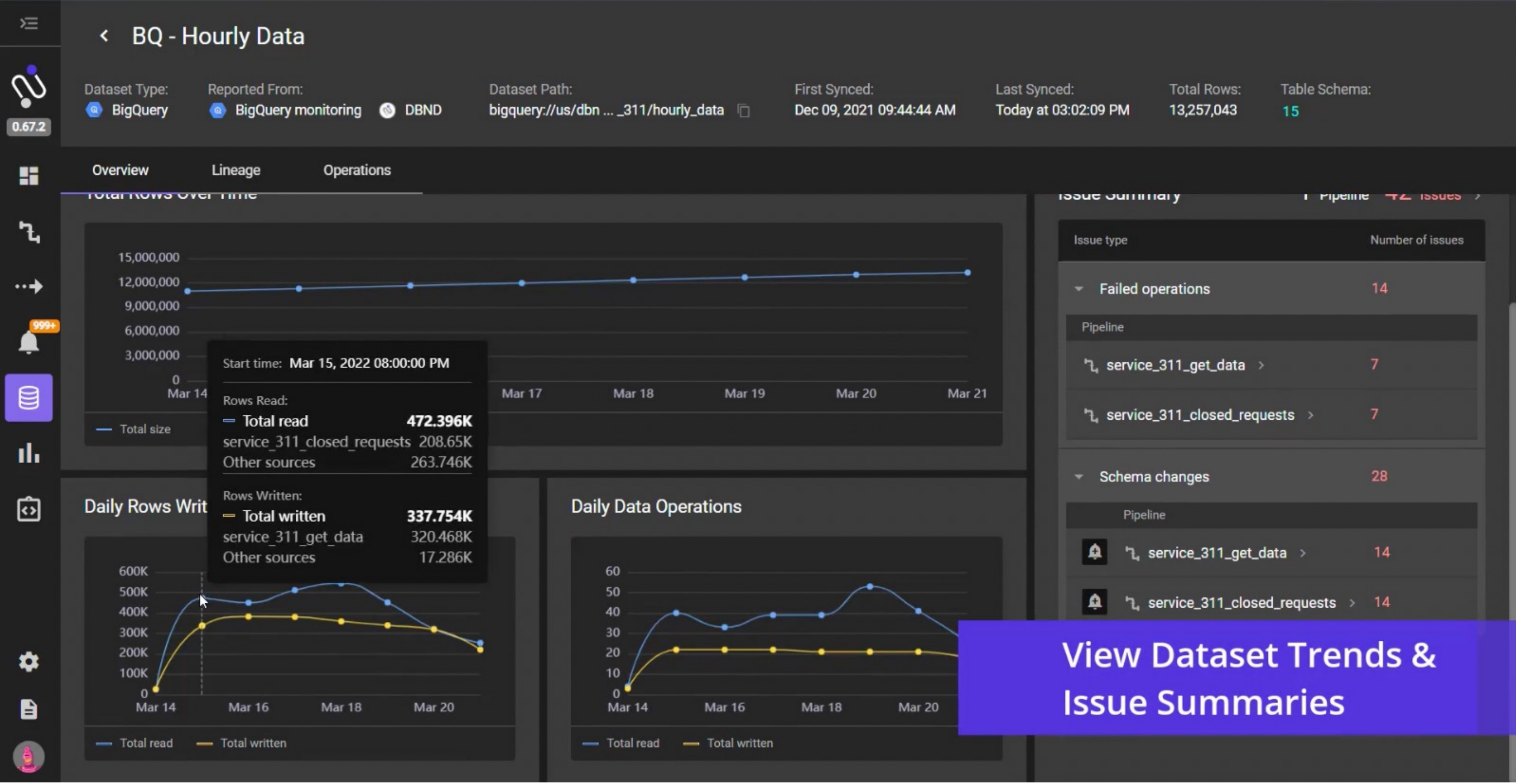
Records count 9,122

Anomaly Range:

Max: 17,105

Min: 5,789

Understand Schema Changes



BQ - Hourly Data

Dataset Type: BigQuery

Reported From: BigQuery monitoring, DBND

Dataset Path: bigquery://us/dbn ... _311/hourly_data

First Synced: Dec 09, 2021 09:44:44 AM

Last Synced: Today at 03:02:09 PM

Total Rows: 13,257,043

Table Schema: 15

Total Rows Over Time

Start time: Mar 15, 2022 08:00:00 PM

Rows Read:	
Total read	472.396K
service_311_closed_requests	208.65K
Other sources	263.746K

Rows Written:	
Total written	337.754K
service_311_get_data	320.468K
Other sources	17.286K

Daily Rows Writ

Daily Data Operations

Issue Summary 1 Pipeline 42 Issues

Issue type	Number of issues
Failed operations	14
Pipeline	
service_311_get_data	7
service_311_closed_requests	7
Schema changes	28
Pipeline	
service_311_get_data	14
service_311_closed_requests	14

View Dataset Trends & Issue Summaries