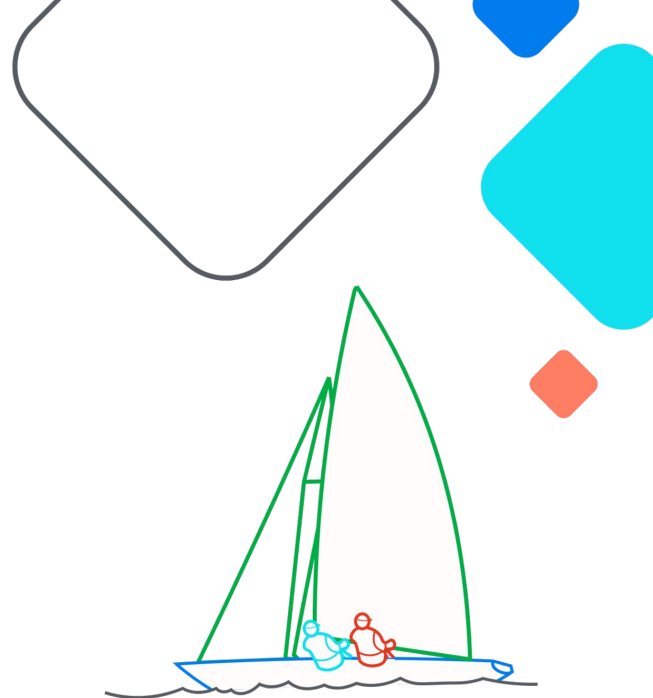


# Data At Rest

## Granular Quality in Flowing Pipelines

Mauricio De Diana  
C.J. Jameson  
2023-09-20



 **Airflow Summit**

Let's flow together

September 19-21, 2023,  
Toronto, Canada

# Data Quality: The Data itself, and how it flows

- Data itself
  - Field-level metrics: null rate, uniqueness, etc
  - Freshness and volume
  - Schema
- Data flow / processing
  - DAGs and tasks
  - dbt pipelines
  - Databricks jobs

# Phases of a data incident

- Prevention
- Detection
- Resolution

# Airflow operators

*Detection/Prevention - field level*

- PythonOperator with an error
- SQLColumnCheckOperator: evaluates fields in isolation
- SQLTableCheckOperator: allows evaluation of multiple fields
- SQLCheckOperator: evaluates single row returned
- ShortCircuitOperator: skips but do not stop/alert

Skips silently, email on failure

Show code example / DAG image, talk about detection vs prevention

# Great Expectations / dbt tests







*Detection/Prevention, field-level*

- More descriptive
- Integrates with Slack, Email
- TODO: Can be used as CB?

Show GE snippet example

Show dbt DAG or snippet or output

# Monte Carlo monitors *Detection/Resolution*

>	Field Health		[Snowflake] demo_env:staging.zuora_invoice <b>Field:</b> invoice_amount <b>Segmented by:</b> status	epost+demo@montecarlodata.com	✔	20 hours ago / in a day	0		📄 ⋮ ⌂
>	Volume Rule	Ensure that the number of new offers does not dip below the rolling 7 day average	[Snowflake] demo_env:raw.offer	mcdemo@montecarlodata.com	✔	a day ago / in an hour	3		📄 ⋮ ⌂
>	SQL Rule	SQL rule to ensure all records from SFDC Opportunity get loaded downstream	[Snowflake] demo_env:raw.salesforce_opportunity, [Snowflake] demo_env:reporting_d_opportunity, [Snowflake] demo_env:staging.salesforce_opportunity <b>Variables:</b> agg (2 values) field (1 values) table_1 (2 values) table_2 (2 values) category (3 values)	mcdemo@montecarlodata.com	sql variables ✔	6 days ago / in 19 hours	0		📄 ⋮ ⌂
>	Field Health		[Snowflake] demo_env:raw.salesforce_account	mcdemo@montecarlodata.com	✔	Triggered dynamically	0		📄 ⋮ ⌂
>	Freshness Rule	Test to ensure that plan data is being updated throughout the day	[Snowflake] demo_env:raw.plan	mcdemo@montecarlodata.com	✔	2 days ago / in 7 hours	3 ⚠		📄 ⋮ ⌂
>	Dimension Tracking		[Snowflake] demo_env:raw.subscription <b>Field:</b> status	mcdemo@montecarlodata.com	✔	10 hours ago / in 2 hours	0		📄 ⋮ ⌂

# Monte Carlo monitors *Detection/Resolution*

◀ Viewing field quality rule: % zero > 0% for invoice\_quantity field in [Snowflake] demo\_env:raw.zuora\_invoice table

- Monitor Details
- Results
- Pass Rate
- Recent Incidents
- Run History
- Change Log

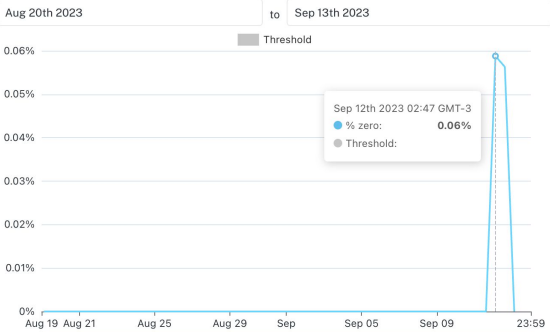
Last run	Sep 13th 2023 02:47 GMT-3 (7 hours ago)
Schedule type	Scheduled
Test interval	This monitor runs every 12 hour(s) starting at May 12th 2023 14:47 GMT-3
Run history:	✔ Monitor ran successfully at Sep 13th 2023 02:47 GMT-3
Status	Model training status: ✔ Fully trained
	Misconfigurations: ✔ None
Created at	May 12th 2023 14:35 GMT-3
Created by	epost+demo@montecarlodata.com
Last modified at	May 12th 2023 14:35 GMT-3
Last modified by	epost+demo@montecarlodata.com
Incidents (7d)	2

Incident Trend (30d)

Date	Incidents
Aug 15th	0
Aug 16th	0
Aug 23rd	0
Aug 27th	0
Aug 31st	0
Sep 4th	0
Sep 6th	0
Sep 12th	2

# Monte Carlo monitors *Detection/Resolution*

## Field Quality Rule run results



Runtime	Result	Rule Breach
Sep 13th 2023 02:47 GMT-3	0%	No breach
Sep 12th 2023 14:48 GMT-3	0.06%	<a href="#">View breach</a>
Sep 12th 2023 02:47 GMT-3	0.06%	<a href="#">View breach</a>



## [Snowflake] SQL rule breached: Circuit Breaker Condition

Opened: Wednesday Sep 13th 2023 10:02 GMT-3 (Now)



NO STATUS

### Summary

#### Breached rows

#### Investigation rows

#### SQL Run History

#### Table Lineage

#### Github

### Breached Rows Returned

[Download CSV](#)

#### BREAKER\_RECORD

2023-02-09 17:41:17.433 Z

Breached rows are stored in the data collector. If you would like to disable such collection, please contact your customer success manager.




# Monte Carlo circuit breaker

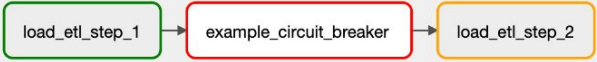
*Detection/Prevention*

2023-08-02T14:15:52Z   Runs 25   Run manual\_2023-08-02T14:15:51.925959+00:00   Layout Left > Right   Update   Find Task...

SimpleCircuitBreakerOperator   SnowflakeOperator

deferred   failed   queued   removed   restarting   running   scheduled   shutdown   skipped   success   up\_for\_reschedule   up\_for\_retry   upstream\_failed   no\_status

Auto-refresh 



```
graph LR; A[load_etl_step_1] --> B[example_circuit_breaker]; B --> C[load_etl_step_2];
```

<https://pypi.org/project/airflow-mcd/>

# Lineage

*Resolution - Data at rest + Data flows*

- Which tables are affected by a data quality problem upstream?
- Is there something in common among apparently unrelated data issues?
- Are there reports affected downstream by this data issue? Which ones?
- Which job(s) populates this table?

# OpenLineage

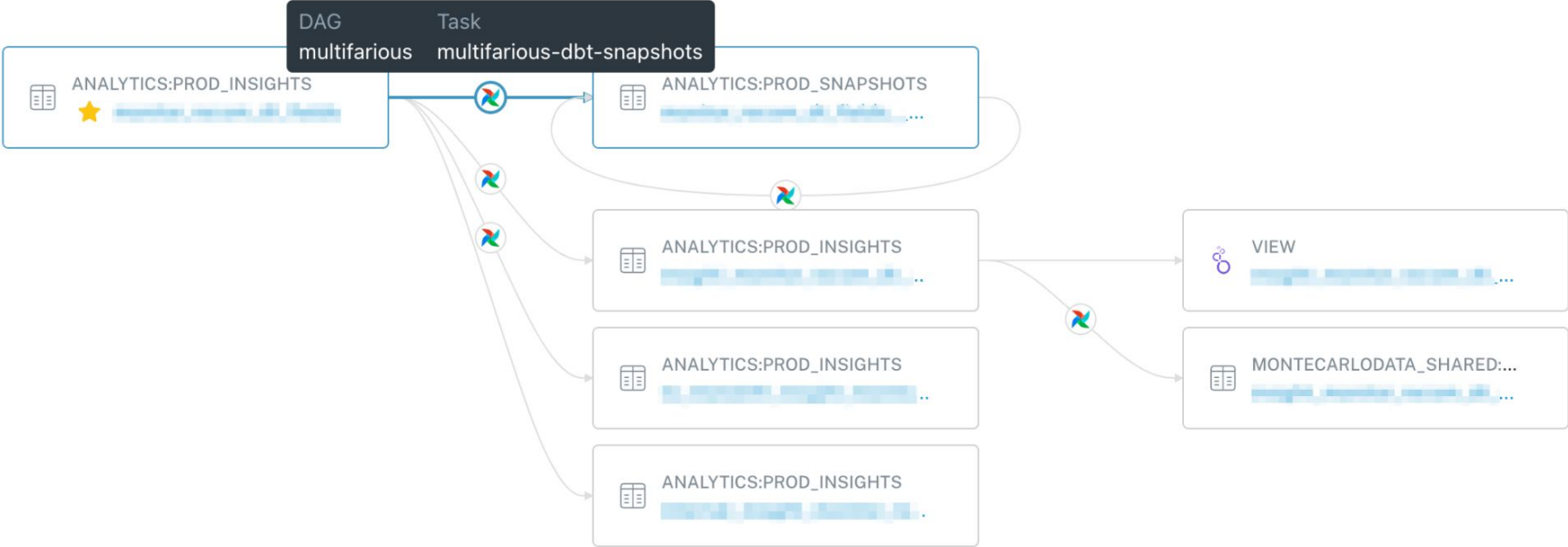
- Marquez
- Astronomer

Show Marquez screenshot

# Monte Carlo lineage Resolution

Direction: Downstream | Depth: 1 | Types: 3 selected | Incident Overlay: 1 day

Show full table names |  Show normalized incidents



# Airflow failures *Resolution*



**Airflow DAG failure: subscription\_load, Task: load\_incremental, Exception: 002003 (02000): 01ae060a-0603-...**

Opened: Tuesday Aug 1st 2023 16:28 GMT-3 (1 month, 12 days ago)



**INVESTIGATING** ▾

Summary

Table Lineage

GitHub

dbt

Past incidents

Debug

Internal

## Summary

DAG

[subscription\\_load](#)

Task

[load\\_incremental](#)

Logs

[View logs](#)

Owner

[mcdemo@montecarlo...](#)

Severity

No severity

Jira issues

[CDI-23](#) [To Do](#)

Notification Channels

[montecarlo.alationcatalog.com](#)

[https://montecarloatainc.webhook.office.co...](#)

[OpsGenie](#)

[private](#)

[private](#)

[private](#)

[https://hooks.zapier.com/\\*\\*\\*\\*fnk/](#)

**002003 (02000): 01ae05d0-0603-a63c-0010-a8830cbcc2b2: SQL compilation error: Task 'DEMO\_ENV.RAW.SUBSCRIPTION\_LOAD' does not exist or not authorized.**

Tuesday Aug 1st 2023 16:28 GMT-3

Start	Execution	End	Duration	Skipped Tasks
<a href="#">Aug 01, 2023 16:27:55 GMT-3</a>	Aug 01, 2023 16:27:54 GMT-3	Aug 01, 2023 16:28:06 GMT-3	00:00:11	-

**002003 (02000): 01ae060a-0603-a63b-0010-a8830cbd726e: SQL compilation error: Task 'DEMO\_ENV.RAW.SUBSCRIPTION\_LOAD' does not exist or not authorized.**

Tuesday Aug 1st 2023 17:26 GMT-3

Start	Execution	End	Duration	Skipped Tasks
<a href="#">Aug 01, 2023 17:26:18 GMT-3</a>	Aug 01, 2023 17:26:18 GMT-3	Aug 01, 2023 17:26:23 GMT-3	00:00:04	-

# Airflow failures Resolution

The screenshot shows the Airflow monitoring interface. At the top, there's a navigation bar with 'Monitors', 'Dashboards', 'Incidents', 'Assets', and 'Settings'. The current date and time are 'Jun 6th 2023 14:15 PDT'. The user is logged in as 'Bryce Heltzel'. The main heading is 'Freshness anomalies in [snowflake] analytics:'. Below this, there's a sidebar with navigation options: Summary, Table Lineage, Query Logs, Delta History (marked as Beta), Github, dbt, Fivetran, Airflow (highlighted), Reports Affected, Past incidents, and Active Monitors. The main content area is titled 'Airflow' and shows a 'Table' and 'Lookback' section. The 'Table' section has a search bar and a 'Lookback' dropdown set to '24 hours'. Below this is a 'Runs' table with the following data:

DAG Run ID	DAG	Task	State	Start	End	Duration
scheduled__2023-06-06T16:58:09.519221+00:00	internal_bi	internal-bi-dbt-models	Success	2023-06-06 13:12:10 PDT	2023-06-06 13:40:59 PDT	00:28:49
scheduled__2023-06-06T13:58:09.519221+00:00	internal_bi	internal-bi-dbt-models	Success	2023-06-06 10:14:12 PDT	2023-06-06 10:42:58 PDT	00:28:46
scheduled__2023-06-06T10:58:09.519221+00:00	internal_bi	internal-bi-dbt-models	Success	2023-06-06 07:08:58 PDT	2023-06-06 07:38:08 PDT	00:29:10
scheduled__2023-06-06T07:58:09.519221+00:00	internal_bi	internal-bi-dbt-models	Success	2023-06-06 04:09:00 PDT	2023-06-06 04:39:13 PDT	00:30:13
scheduled__2023-06-06T04:58:09.519221+00:00	internal_bi	internal-bi-dbt-models	Success	2023-06-06 01:08:45 PDT	2023-06-06 01:39:00 PDT	00:30:14
scheduled__2023-06-06T01:58:09.519221+00:00	internal_bi	internal-bi-dbt-models	Success	2023-06-05 22:10:57 PDT	2023-06-05 22:40:55 PDT	00:29:58
scheduled__2023-06-05T22:58:09.519221+00:00	internal_bi	internal-bi-dbt-models	Success	2023-06-05 19:09:05 PDT	2023-06-05 19:37:34 PDT	00:28:29

At the bottom right, there's a pagination control: 'Rows per page: 10' and '1-7 of 7'.

# Airflow failures *Resolution*

Summary

Runs

Past Incidents (1)

## General Information

DAG

subscription\_load

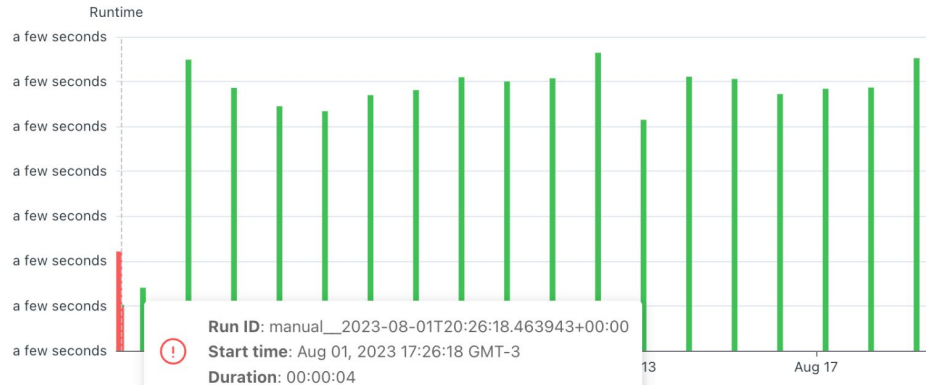
Last run

Sep 12, 2023 05:00:00 GMT-3

## Runtime

Status

All



# Airflow failures *Resolution*

```
# explicit, per callback type
dag = DAG(
    'dag_name',
    on_success_callback=mcd_callbacks.mcd_dag_success_callback,
    on_failure_callback=mcd_callbacks.mcd_dag_failure_callback,
    sla_miss_callback=mcd_callbacks.mcd_sla_miss_callback,
)
```

```
# explicit, per callback type
task = BashOperator(
    task_id='task_name',
    bash_command='command',
    dag=dag,
    on_execute_callback=mcd_callbacks.mcd_task_execute_callback,
    on_success_callback=mcd_callbacks.mcd_task_success_callback,
    on_failure_callback=mcd_callbacks.mcd_task_failure_callback,
    on_retry_callback=mcd_callbacks.mcd_task_retry_callback,
)
```

```
# broad, all callbacks
dag = DAG(
    'dag_name',
    **mcd_callbacks.dag_callbacks,
)
```

```
# broad, all callbacks
task = BashOperator(
    task_id='task_name',
    bash_command='command',
    dag=dag,
    **mcd_callbacks.task_callbacks,
)
```



# Conclusion

# Questions?

[mdediana@montecarlodata.com](mailto:mdediana@montecarlodata.com)  
[@cjcjameson](#) on Twitter  
[www.montecarlodata.com](http://www.montecarlodata.com)



# What do you do with DAG/Task failures?

- do you get an email? A slack? Some other notification?
- is it one data engineer? The whole team? An on-call rotation?
- which DAGs are configured to alert to which channels? Some are more important than others; some are owned by different teams; who will respond?

# Monte Carlo: Airflow failures with tons of context

# Managing a legacy, failing Airflow setup

**You can review Airflow executions with Databand**

# Airflow-over-Databricks: review runtime duration