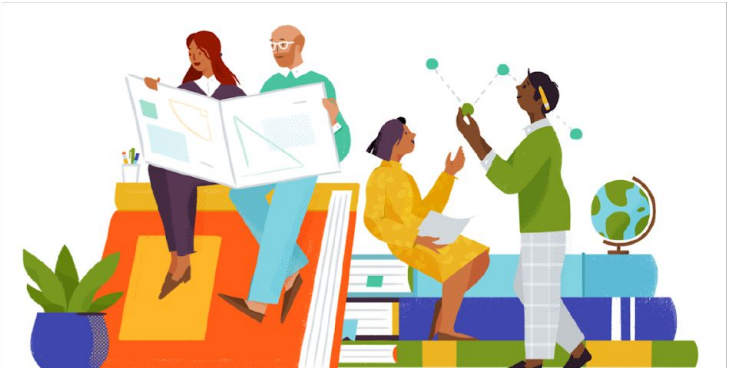
Operating Data Pipeline with Airflow

Ananth Packkildurai



About Slack

2014

Public launch

65

Fortune 100 companies are paid customers

500K

Organizations using slack

slack

About Slack

1000+

Slack employees

12

Data engineers

8M

Active users

Data usage

1 in 2 per week

access data warehouse

500+

Tables

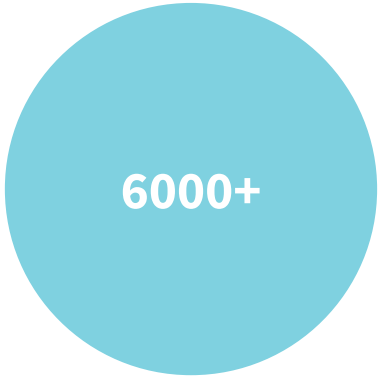
70 million

Events per minute

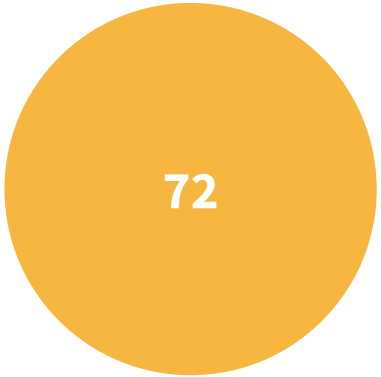
slack

Airflow stats

Active Dags



Tasks Per Day



Contributors

Agenda

1. Airflow Infrastructure
2. Scale Airflow Executor
3. Pipeline Operations
4. Alerting and monitoring

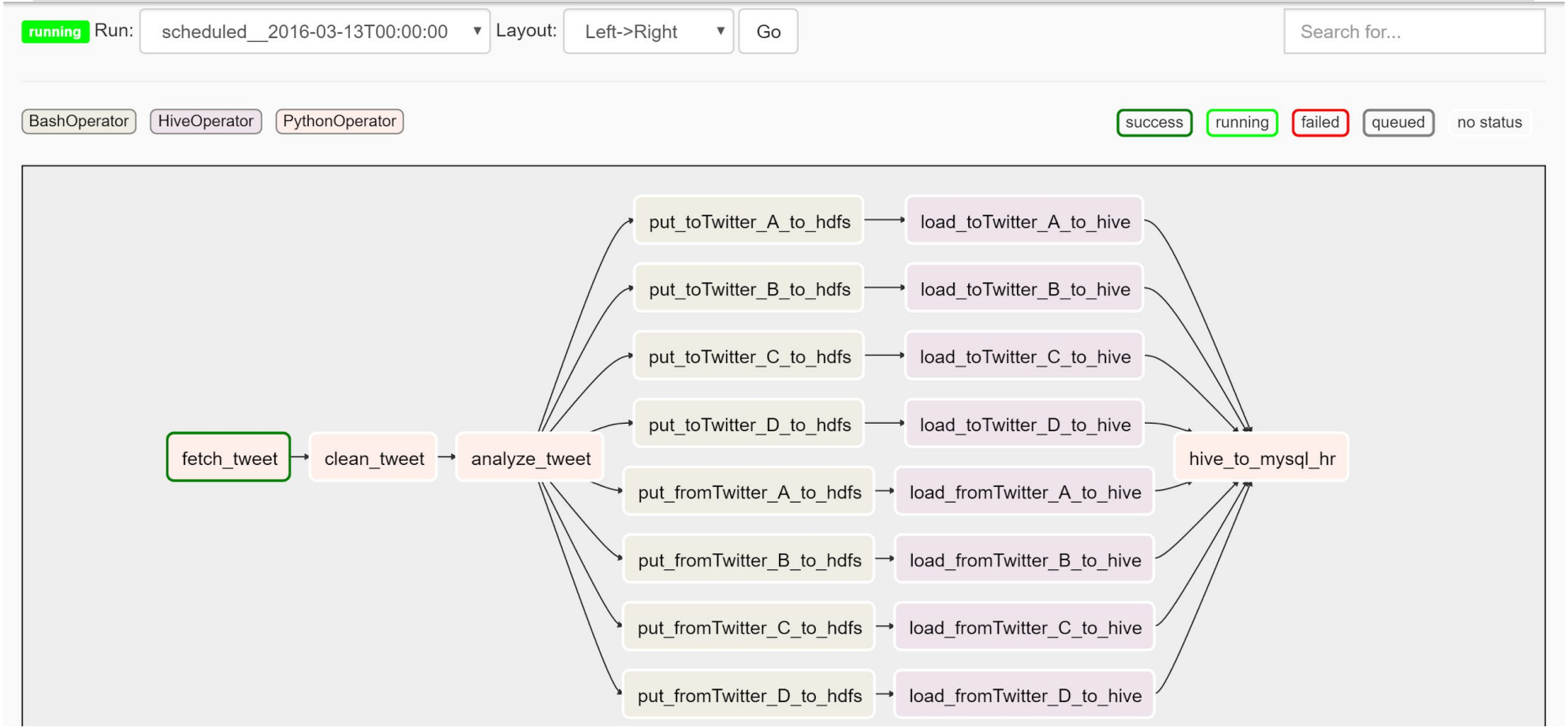
slack

Airflow infrastructure

Airflow overview

- DAG => Graph of airflow tasks (operators)
- Operators =>
  - Sensor Operator (e.g) S3KeySensors
  - Action Operator (e.g) BashOperator, HiveOperator, SparkOperator
  - Transfer Operator (e.g) SalesforceToS3Operator
- Executors:
  - Local Executor
  - Celery Executor
  - Mesos Executor
  - Kubernetes Executor (in progress)

Example DAG


running
Run:
scheduled__2016-03-13T00:00:00
Layout:
Left->Right
Go
Search for...
BashOperator
HiveOperator
PythonOperator
success
running
failed
queued
no status
fetch_tweet
clean_tweet
analyze_tweet
put_toTwitter_A_to_hdfs
load_toTwitter_A_to_hive
put_toTwitter_B_to_hdfs
load_toTwitter_B_to_hive
put_toTwitter_C_to_hdfs
load_toTwitter_C_to_hive
put_toTwitter_D_to_hdfs
load_toTwitter_D_to_hive
put_fromTwitter_A_to_hdfs
load_fromTwitter_A_to_hive
put_fromTwitter_B_to_hdfs
load_fromTwitter_B_to_hive
put_fromTwitter_C_to_hdfs
load_fromTwitter_C_to_hive
put_fromTwitter_D_to_hdfs
load_fromTwitter_D_to_hive
hive_to_mysql_hr

Airflow infrastructure

- Local Executor
- Tarball code deployment
- Continuous deployment with Jenkins
- Flake8, yapf & pytest
- `airflow.sh` shell utility to ensure consistent development environment for all the users.

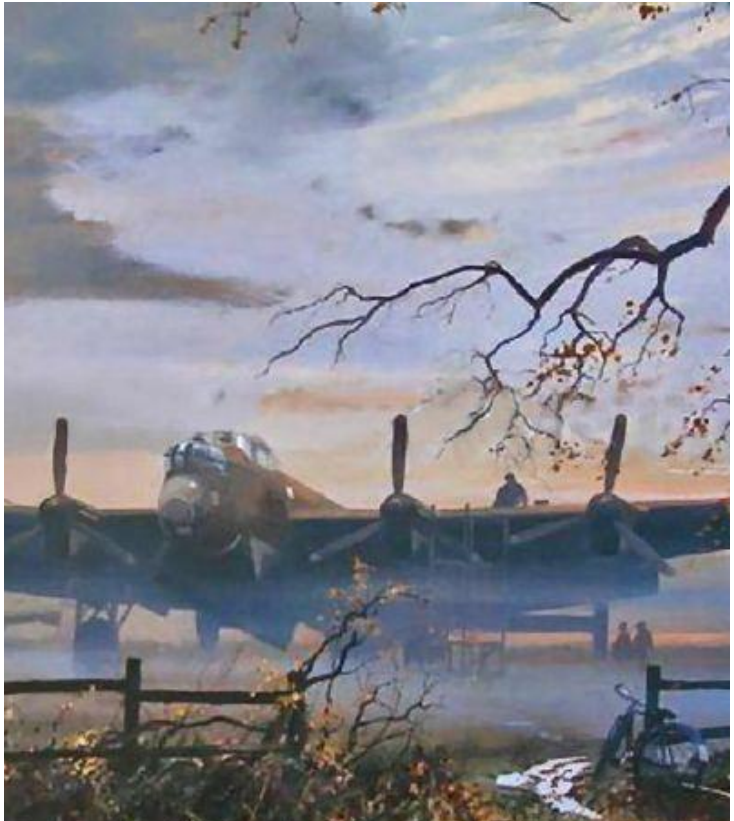
slack

Scale Airflow Executor

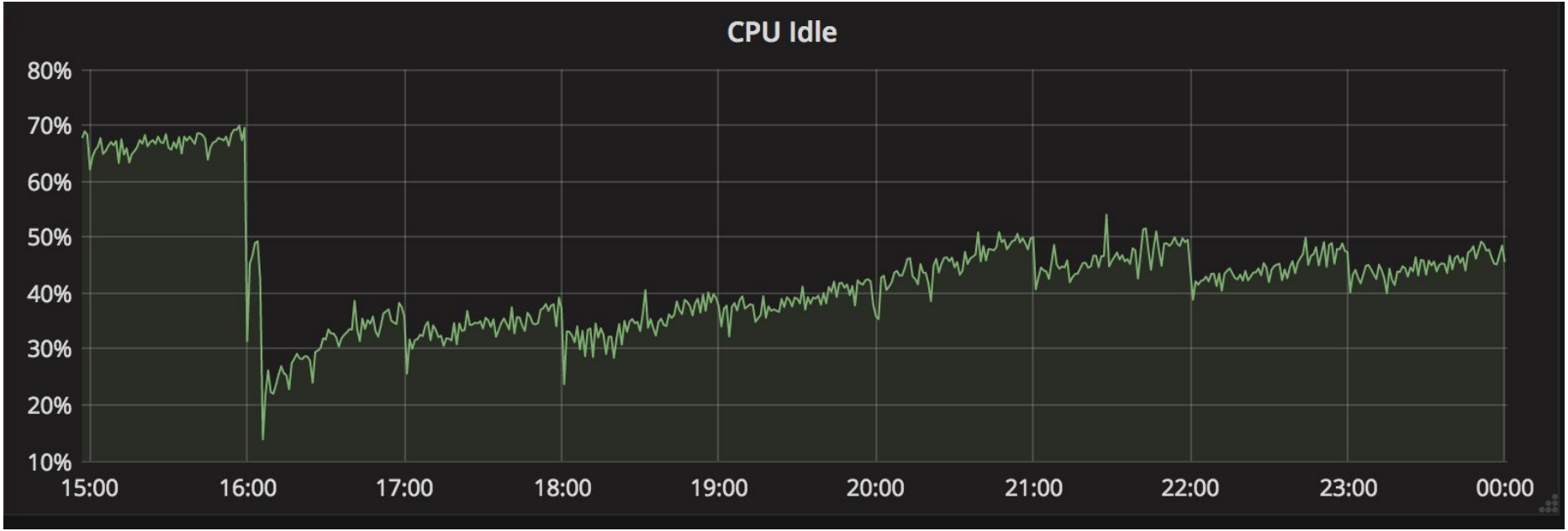
Scale Airflow Executor

It’s just Airflow being Airflow

- Why my task is not running
- Airflow deadlock again
- Airflow not scheduling any tasks



Airflow CPU usage



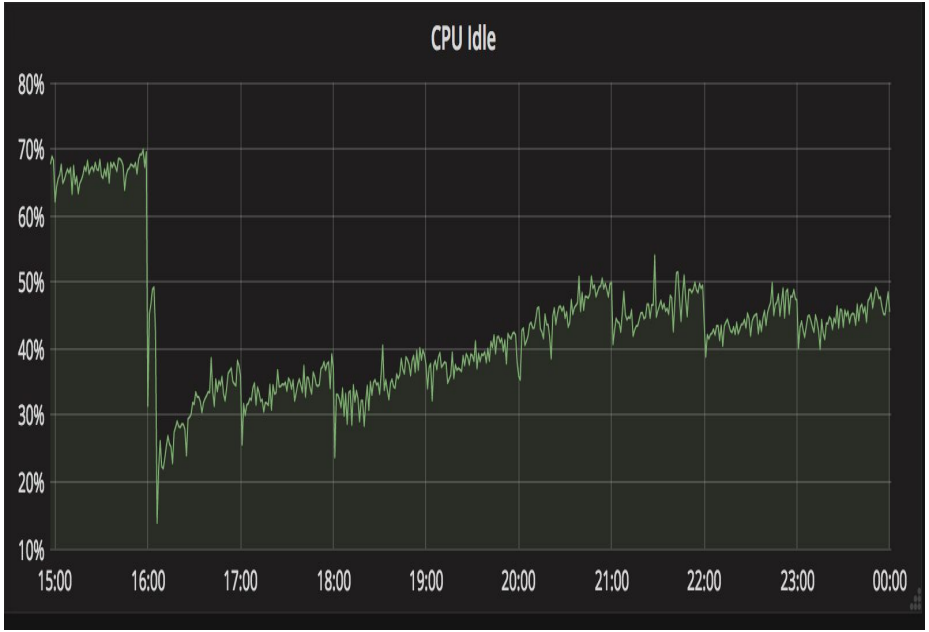
Airflow Multi Retryable Sensors

Airflow local executor launches a new python interpreter per-task, which has been observed to use significant system resources. To minimize the local machine cost of checking external tasks, here we check multiple external tasks in a single task.

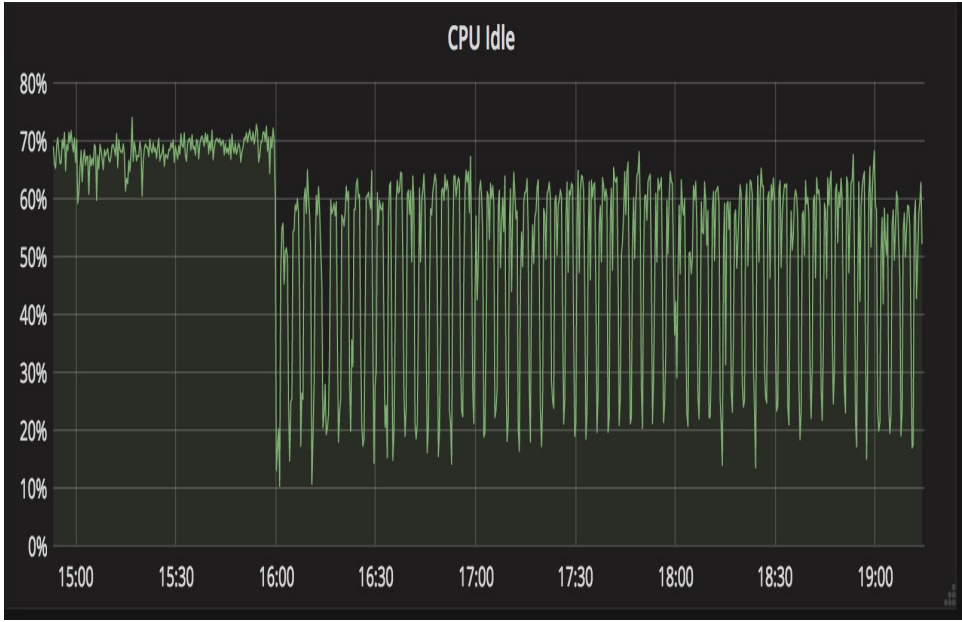
To simplify depending on multiple external tasks, it's recommended using the builder to create new instances:

```
wf = (
    MultiRetryableExternalTaskSensor.Builder(task_id='my_task', dag=dag)
        .waitfor('dag1', 'task1', ...)
        .waitfor('dag2', 'task2', ...)
        .build()
)
```

Retryable Sensors CPU usage



Non-Retryable Sensors Load



Retryable Sensors Load

slack

Pipeline Operations

Airflow fallacies

- The upstream task success is reliable.
- The task remain static after the success state.
- The DAG structure is static.
- The data quality not part of a task life cycle.

Mario: Global DAG operator

```
~/w/d/bin ››› ./mario --help
Usage: mario [OPTIONS] COMMAND1 [ARGS]... [COMMAND2 [ARGS]...]...

Options:
  --help  Show this message and exit.

Commands:
  clear-downstream  Print Airflow commands to clear all...
  dependencies      Export a .graphml representation of airflow...
  downstream        Find all tasks (across DAGs) that are...
  export_graphml    Export a .graphml representation of airflow...
  render            Render Airflow tasks with local params.
  upstream          Find all tasks (across DAGs) that are...
  waitfor-counts    Prints the number of waitfors that are...
  why               Por que?
```

Airflow operations

Hive Partition Sensor Operator

1. Check task success state
2. Check Hive metastore for partition
3. Check S3 path for the `_SUCCESS` file

DQ Check

```
from slack.airflow.data_quality import DQCheck

MyTaskOperator(
 task_id='my_task',
  dag=dag,
  dq_sql=DQCheck('dw.my_table', unique='my_primary_key').is_positive("count(1)", "row_count", over=1).build()
)
```

DAG cleanup

delete_dag <dag name>

slack

DAG Policy Validator

test_external_tasks	Check if external tasks point to valid DAGs and tasks.
test_circular_dependencies	Check if tasks have circular dependencies *across* DAGs.
test_priority_weight	Check that production tasks do not depend on a lower priority task.
test_on_failure	Require that high-priority DAGs have an on-failure alert.

DAG Policy Validator

test_sla	Require that high-priority DAGs have an SLA.
test_sla_timing	SLAs timing should make sense. No job should depend on a task that has an equal or longer SLA than it does.
test_has_retry_and_success_callbacks	Require an on_success_callback for tasks with an on_retry_callback.
test_require_dq_for_prod	Require SQ check for all the high priority tasks.

slack

Alerting and Monitoring

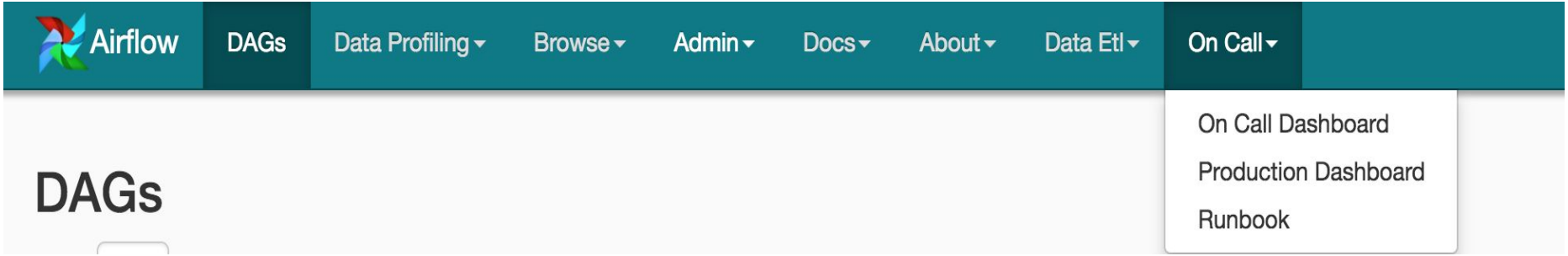
Alerting and Monitoring

- Alerting should be reliable
- Alerts should be actionable.
- Alert when it really matters.
- Suppress repeatable alerts.

OnCall Alert callback

```
from slack.airflow.webhook import Channels, OncallAlertCallback
dag = SlackDAG(
    alerts=OncallAlertCallback(
        channels=Channels.MY_CHANNEL,
        priority='high',
        escalation_chain=['Foo', 'Bar'],
        notes='This task upstream of billing pipeline'
        resources={
            'My runbook': 'https://slack-github.com/path/to/runbook',
        },
        pagerduty_email='data@slack.com',
    ),
)
```

OnCall Alert callback



Sample Alerts

Airflow Alerts APP 5:00 PM

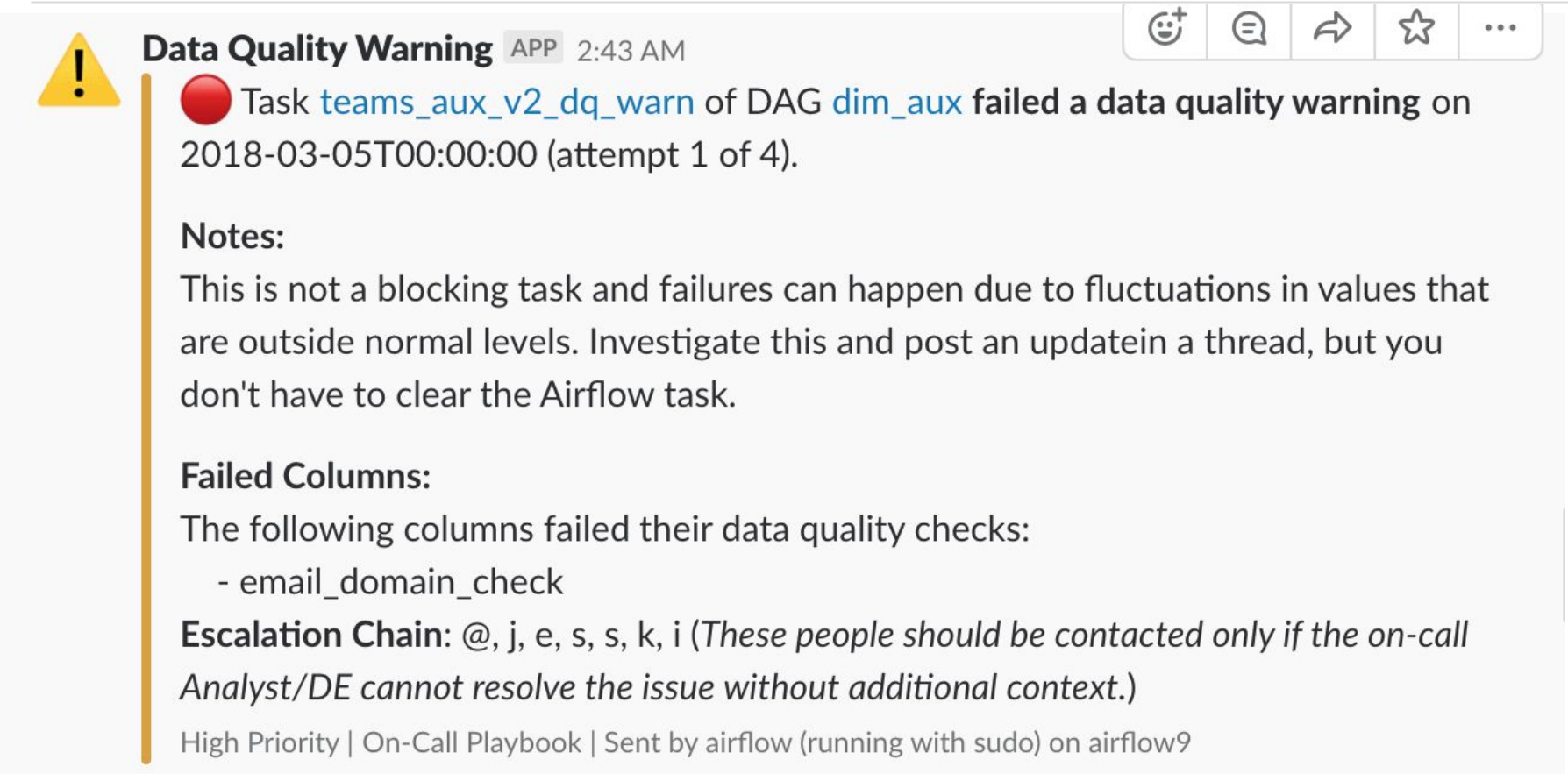
🔴 SLA miss for DAG team_company_segment task **team_company_segment_all_done** on 2018-04-04T00:00:00.

Notes:
This is a core pipeline. No downstream tasks will run until this task succeeds.

Escalation Chain: atl, lj (*These people should be contacted only if the on-call Analyst/DE cannot resolve the issue without additional context.*)

High Priority | On-Call Playbook | Sent by airflow on airflow9

Sample Alerts



Data Quality Warning APP 2:43 AM

🔴 Task teams_aux_v2_dq_warn of DAG dim_aux **failed a data quality warning** on 2018-03-05T00:00:00 (attempt 1 of 4).

Notes:
This is not a blocking task and failures can happen due to fluctuations in values that are outside normal levels. Investigate this and post an updatein a thread, but you don't have to clear the Airflow task.

Failed Columns:
The following columns failed their data quality checks:
- email_domain_check

Escalation Chain: @, j, e, s, s, k, i (*These people should be contacted only if the on-call Analyst/DE cannot resolve the issue without additional context.*)

High Priority | On-Call Playbook | Sent by airflow (running with sudo) on airflow9

Sample Alerts

Data Quality Failure APP 10:43 AM

🔴 Task enterprise_stats of DAG customer_stats **failed a data quality check** on 2018-03-05T00:00:00 (attempt 1 of 4).

Notes:

This is a core pipeline. No downstream tasks will run until this task succeeds.

Failed Columns:

The following columns failed their data quality checks:

- row_count

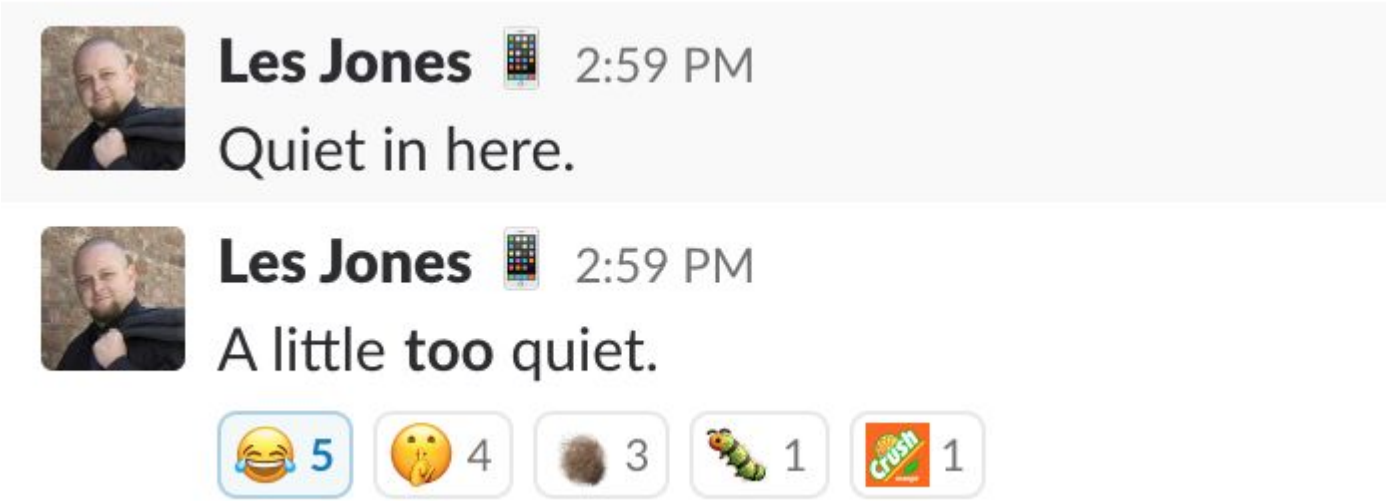
Triage Resources:

- Customer Stats runbook

Show less

High Priority | On-Call Playbook | Sent by airflow (running with sudo) on airflow9

A little too quiet



slack

Summary

Summary

- Keep your infrastructure simple
- You don’t own the platform, the users own it
- Automation is God

slack

Thank You!