



Clearing Airflow obstructions

[@tati_alchueyr](https://twitter.com/tati_alchueyr)

Principal Data Engineer @ BBC Datalab

Airflow 
Summit 2021

Online
15 July 2021

Heads up!

- During this **session** there will be some **quizzes**
- Be prepared to either:
 - **Scan** the **QR code**
 - **Access** using the **URL**



@tati_alchueyr. **__doc__**

- **Brazilian** living in London since 2014
- Principal Data **Engineer** at the **BBC Datalab** team
- Graduated in **Computer Engineering** at Unicamp
- **Passionate** software developer for **18 years**
- Experience in the **private** and **public** sectors
- Developed software for **Medicine**, **Media** and **Education**

- Loves **Open Source**
- Loves **Brazilian Jiu Jitsu**
- Proud mother of **Amanda** (v4.0)



I ♥ Airflow Community & Summit



Tomek
Urbaszek



Jarek
Potiuk



Kaxil Naik



Ash
Berlin-Taylor



Leah
Cole

BBC.Datalab.Hummingbirds

The work presented here is the result of lots of **teamwork** within **one squad** of a much **larger team** and **organisation**



Darren
Mundy



David
Hollands



Richard
Bownes



Tatiana
Al-Chueyr



active squad team members

directly contributed in the past



Bettina
Hermant



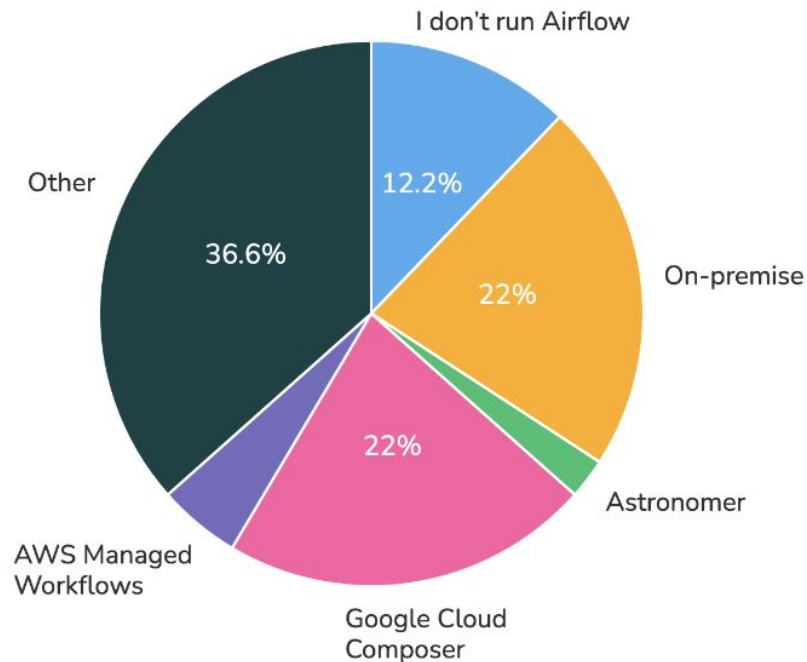
Marc
Oppenheimer

Quiz time how do you run Airflow?

- A. I don't run Airflow
- B. On-premise
- C. Astronomer.io
- D. Google Cloud Composer
- E. AWS Managed Workflows
- F. Other



Quiz time how do you run Airflow?



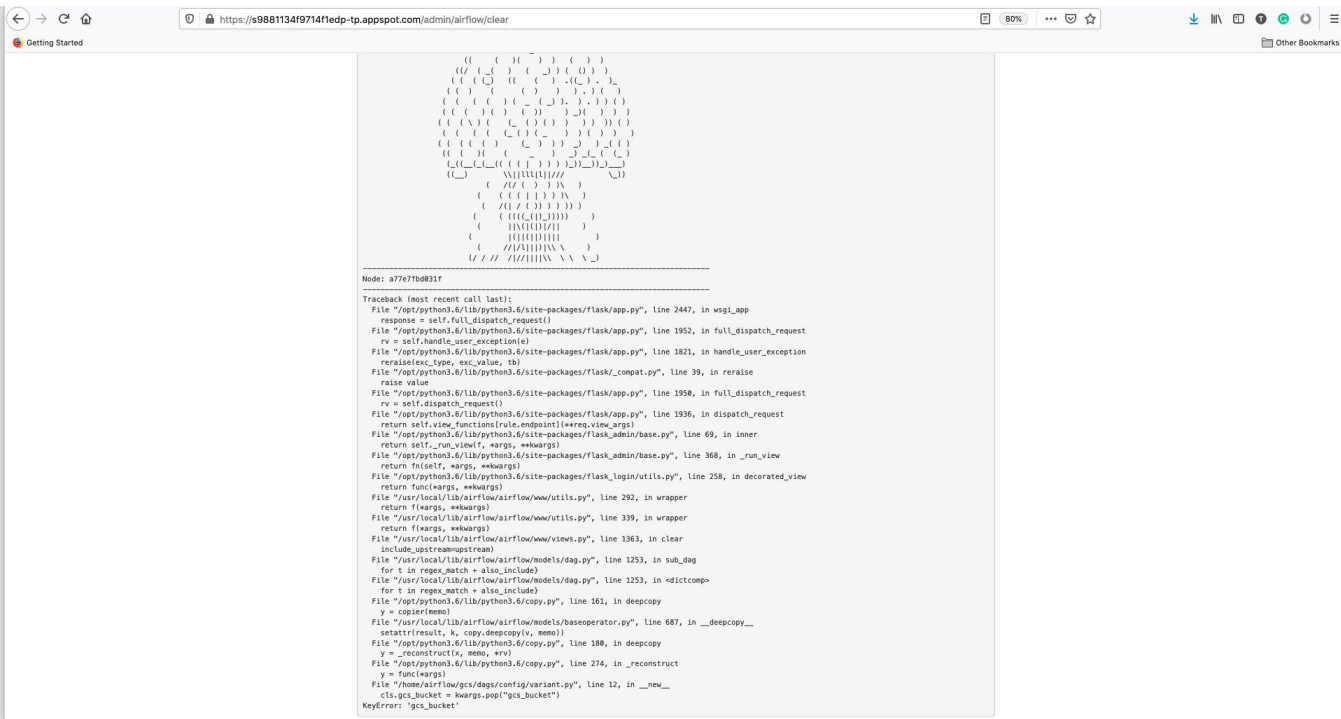
* responses from Airflow Summit 2021 participants, during the presentation



How we use Airflow

when things went wrong

How we use Airflow **when things go wrong**



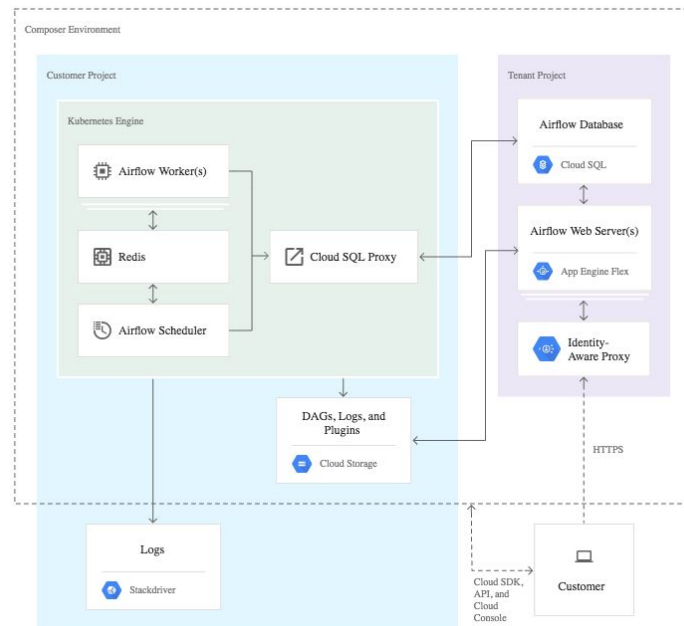
The screenshot shows the Airflow web interface in a browser. The address bar displays the URL `https://s9861134f9714fedp-tp.appspot.com/admin/airflow/clear`. The page title is "Getting Started". The main content area shows a traceback for a `KeyError` with the message `'gcs_bucket'`. The traceback starts from the bottom and goes up, showing the following sequence of calls:

```
File "/opt/python3.6/lib/python3.6/site-packages/flask/app.py", line 2447, in wsgi_app
    response = self.full_dispatch_request()
File "/opt/python3.6/lib/python3.6/site-packages/flask/app.py", line 1952, in full_dispatch_request
    rv = self.handle_user_exception(e)
File "/opt/python3.6/lib/python3.6/site-packages/flask/app.py", line 1821, in handle_user_exception
    reraise(exc_type, exc_value, tb)
File "/opt/python3.6/lib/python3.6/site-packages/flask/_compat.py", line 39, in reraise
    raise value
File "/opt/python3.6/lib/python3.6/site-packages/flask/app.py", line 1958, in full_dispatch_request
    rv = self.dispatch_request()
File "/opt/python3.6/lib/python3.6/site-packages/flask/app.py", line 1936, in dispatch_request
    return self.view_functions[rule.endpoint](**req.view_args)
File "/opt/python3.6/lib/python3.6/site-packages/flask_admin/base.py", line 69, in inner
    return self._run_view(f, *args, **kwargs)
File "/opt/python3.6/lib/python3.6/site-packages/flask_admin/base.py", line 368, in _run_view
    return f(self, *args, **kwargs)
File "/opt/python3.6/lib/python3.6/site-packages/flask_login/utils.py", line 258, in decorated_view
    return func(*args, **kwargs)
File "/usr/local/lib/airflow/airflow/www/utils.py", line 262, in wrapper
    return f(*args, **kwargs)
File "/usr/local/lib/airflow/airflow/www/utils.py", line 339, in wrapper
    return f(*args, **kwargs)
File "/usr/local/lib/airflow/airflow/www/views.py", line 1363, in clear
    include_upstream=upstream)
File "/usr/local/lib/airflow/airflow/models/dag.py", line 1253, in sub_dag
    for t in rege.match = also_include)
File "/usr/local/lib/airflow/airflow/models/dag.py", line 1253, in <dictcomp>
    for t in rege.match = also_include)
File "/opt/python3.6/lib/python3.6/copy.py", line 161, in deepcopy
    y = copier(memo)
File "/usr/local/lib/airflow/airflow/models/baseoperator.py", line 687, in __deepcopy__
    setattr(result, k, copy.deepcopy(v, memo))
File "/opt/python3.6/lib/python3.6/copy.py", line 180, in deepcopy
    y = _reconstruct(memo, rv)
File "/opt/python3.6/lib/python3.6/copy.py", line 274, in _reconstruct
    y = func(*args)
File "/home/airflow/gcs/dags/config/variant.py", line 12, in __new__
    cls.gcs_bucket = kwargs.pop("gcs_bucket")
KeyError: 'gcs_bucket'
```

The traceback ends with the message `KeyError: 'gcs_bucket'`.

How we use Airflow **infrastructure**

- **Managed** Airflow
 - Cloud Composer (GCP)
 - Terraform
- **Celery Executors** GCP constraint
 - Running within Kubernetes (GKE)
- **Outdated** version **1.10.12**
 - Upgrades have been time consuming
 - Last: 1.10.4 => 1.10.12 @ Dec '20
 - GCP supports newer releases
 - 1.10.5 since April '21 (March '21)
 - 2.0.1 since May '21 (Feb '21)





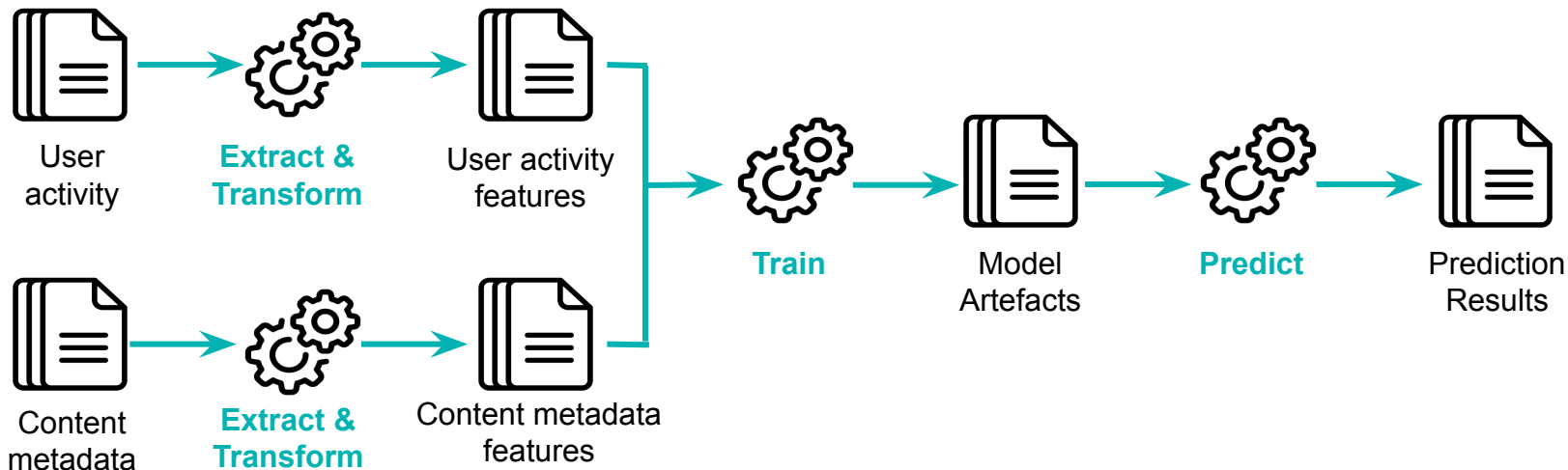
How we use Airflow **operators**

- Execute within **Airflow executors**
 - BaseOperator
 - BashOperator
 - DummyOperator
 - **PythonOperator**
 - ShortCircuitOperator
 - **TriggerDagRunOperator**
 - GCSDDeleteObjectsOperator
- Delegate to **Kubernetes** in a dedicated GKE cluster
 - **GKEPodOperator**
- Delegate to **Apache Beam** (Dataflow)
 - **DataflowPythonOperator**
 - DataflowCreatePythonJobOperator

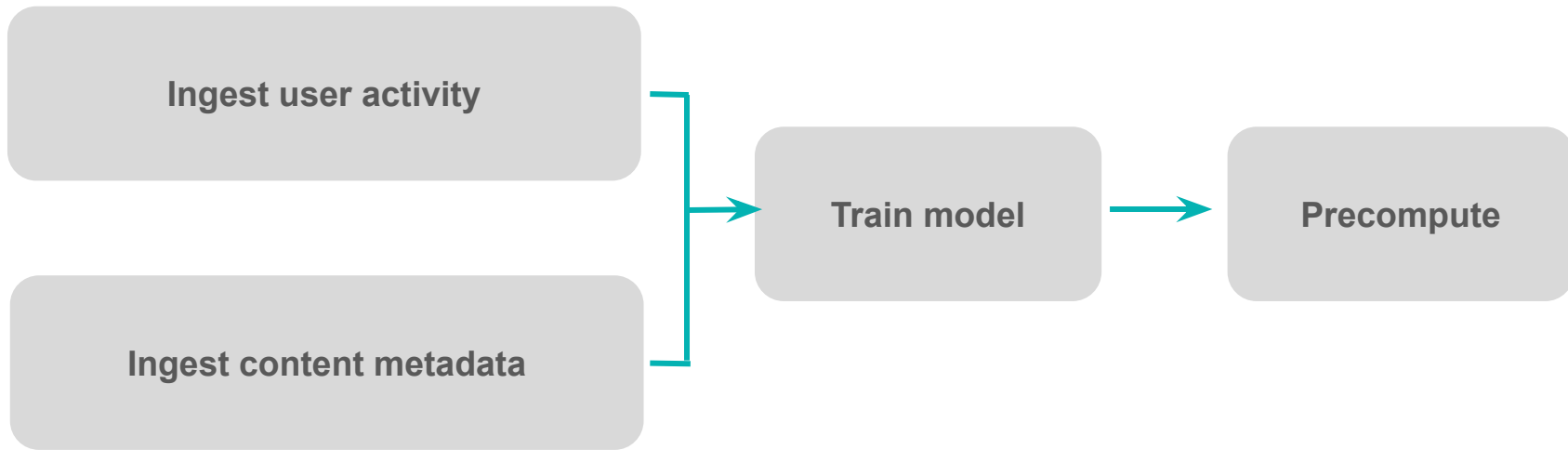
DAGs

How we use Airflow **application**

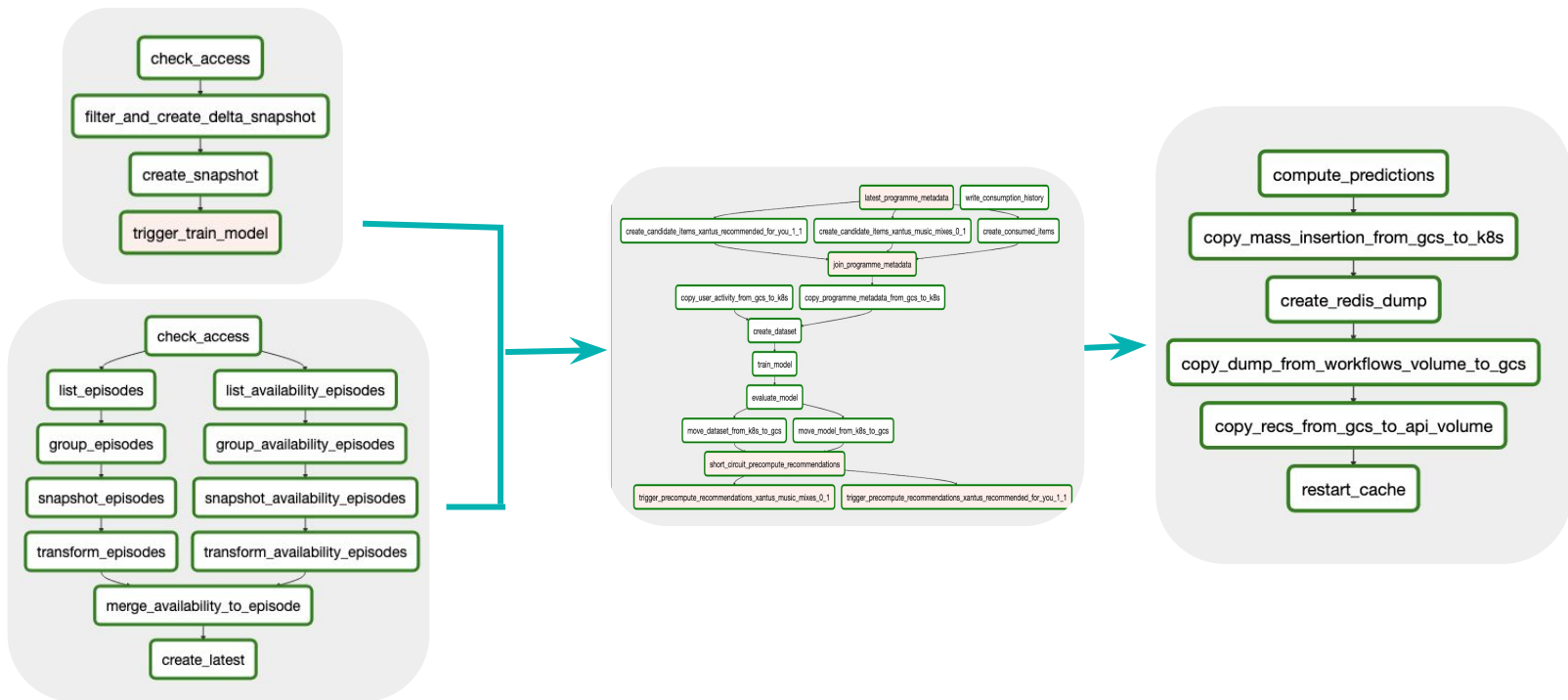
- Data integration: **ETL** (extract, transform, load)
- **Machine learning**: training and precomputation



How we use Airflow **application**



How we use Airflow application



Quiz time which is our most stable DAG?

- A. Ingest & transform Content Metadata
- B. Ingest & transform User Activity
- C. Train model
- D. Precompute recommendations

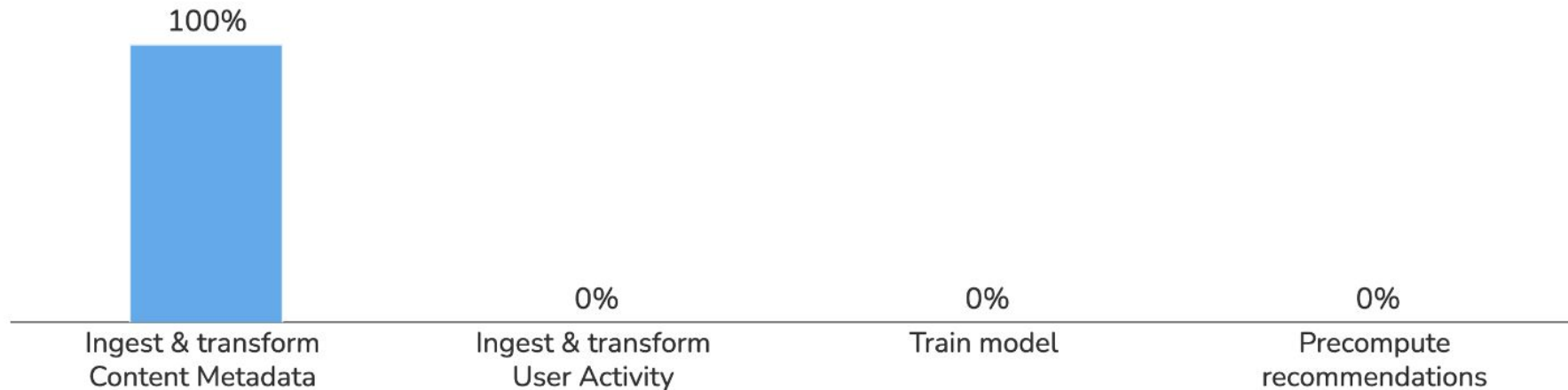


Quiz time which is our most stable DAG? tips

- A. Ingest & transform Content Metadata **Python Operator**
 - ~ 225k records
 - Transforms ~12 GB => 57 MB
- B. Ingest & transform User Activity **Dataflow Operator**
 - ~ 3.5 million records
 - Output: ~ 2 GB
- C. Train model **Kubernetes Operator**
 - Output: ~ 8 GB model & artefacts
- D. Precompute recommendations **Dataflow Operator**
 - ~ 3.5 million records
 - Output: ~ 2.5 GB



Quiz time which is our most stable DAG? attendees



* responses from Airflow Summit 2021 participants, during the presentation

Quiz time which is our most stable DAG? **answer**

A. Ingest & transform Content Metadata

- 2 live incidents

B. Ingest & transform User Activity

- **1 live incident**



C. Train model

- 2 live incidents

D. Precompute recommendations

- 2 live incidents

* from October 2020 until April 2021

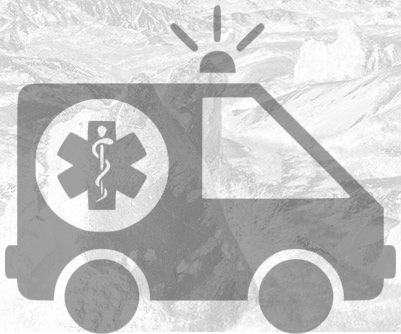


Quiz time which is our most stable DAG? details

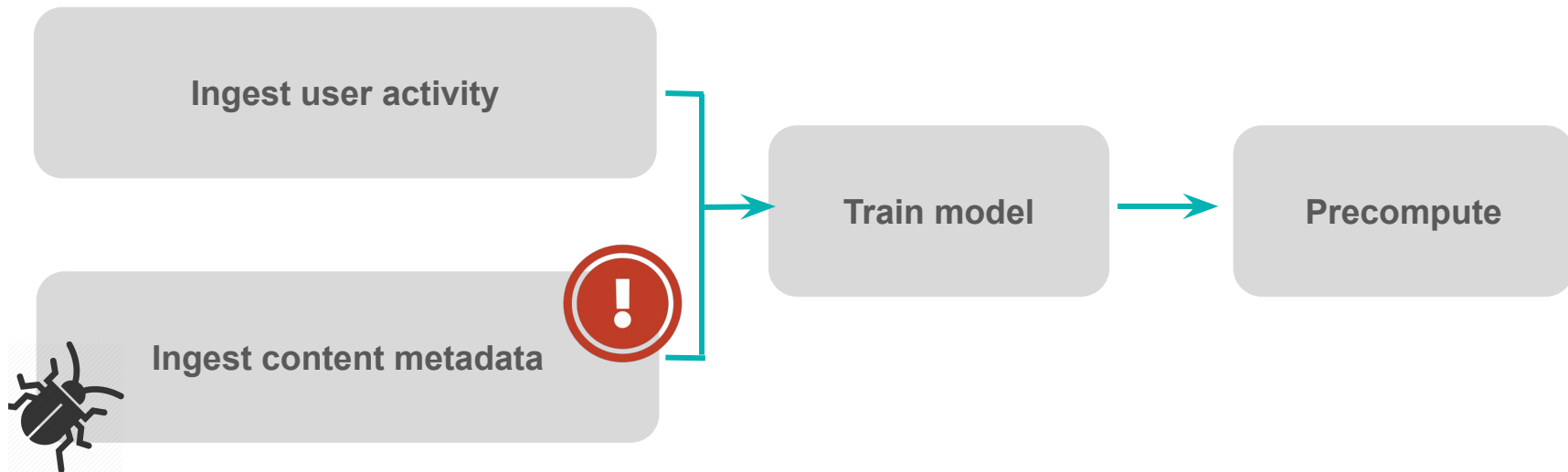
- A. Ingest & transform Content Metadata
 - Insufficient CPU
 - Spikes -> Timeouts (during higher volumes) / Continuing from where it stopped
- B. Ingest & transform User Activity
 - Idempotency issue
- C. Train model
 - K8s Pod reattach
 - Scheduling leading to two tasks running concurrently
- D. Precompute recommendations
 - Change to default job settings in Dataflow
 - GCS access limit
 - Non-stop Dataflow job

Removal of workflows obstructions

when things went wrong



Obstruction 1 The programme metadata chronic issue



Obstruction 1 The programme metadata chronic issue

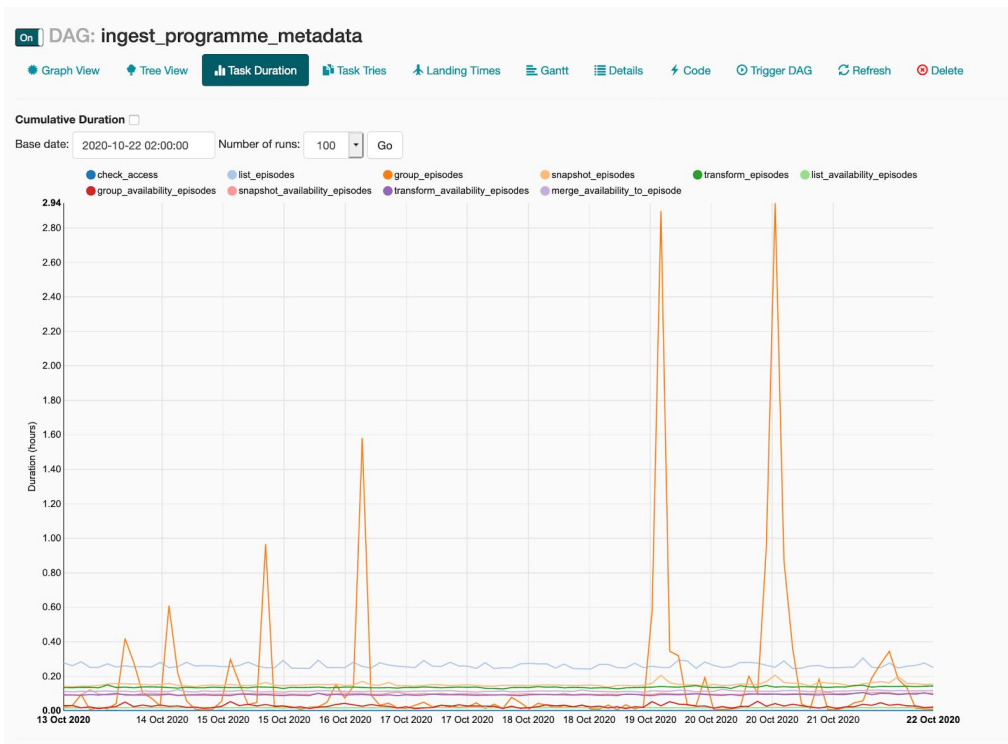
DAG's goals

- Import objects from AWS S3 (protected by STS) into Google Cloud Storage
- Requirements: between dozens and thousands KB-sized objects
- Filter and enrich the metadata
- Merge multiple streams of data and create an up-to-date snapshot

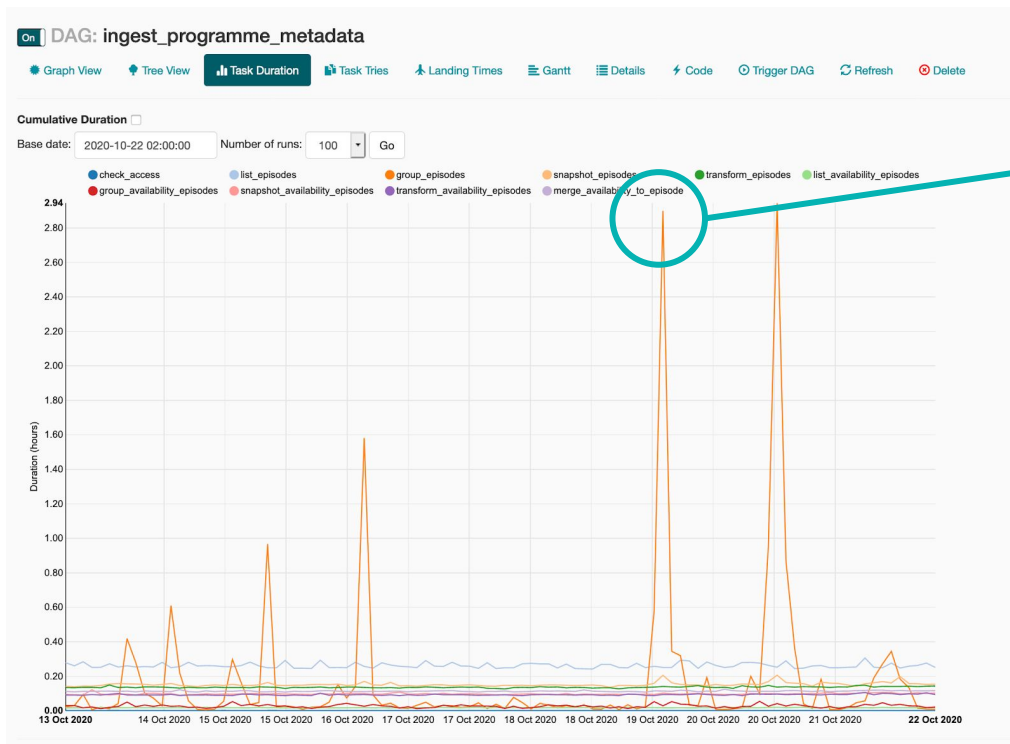


Mostly implemented using subclasses of the **Python Operator** class

Obstruction 1 The programme metadata chronic issue



Obstruction 1 The programme metadata chronic issue



Issue

Depending on the volumes of data, a single **PythonOperator** task which usually takes **10 min** could take almost **3h!**

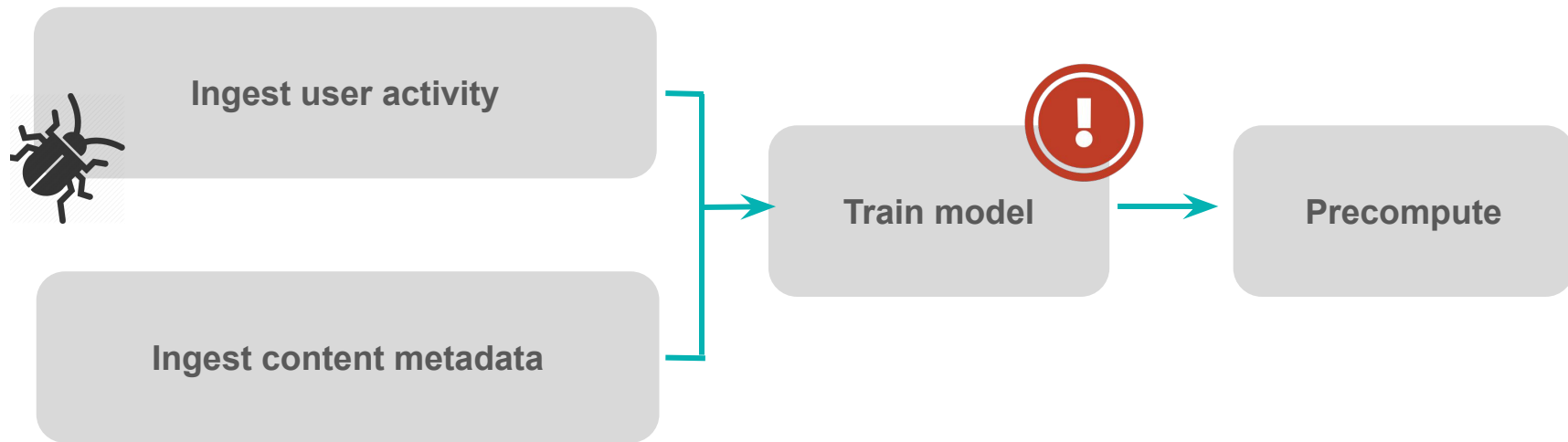
Consequences

Delay
Blocked Airflow executor

Solutions

Increase timeouts
Improve machine type
Delegate processing to another service

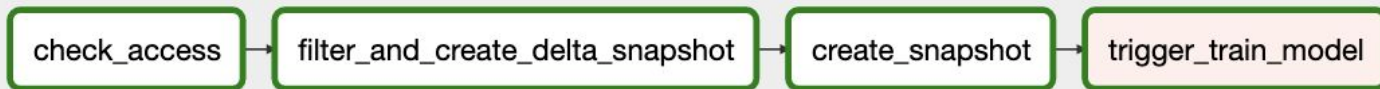
Obstruction 2 When the user activity workflow failed



Obstruction 2 When the user activity workflow failed

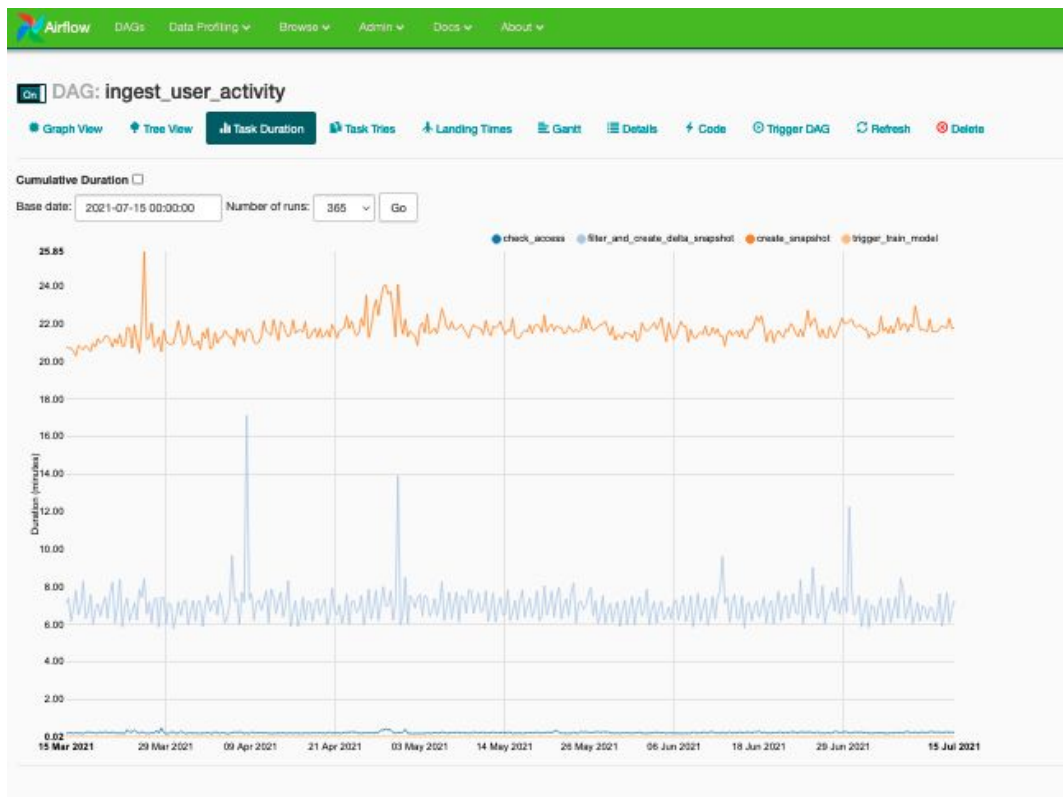
DAG's goals

- Read from user activity Parquet files in Google Cloud Storage
- Filter relevant activity and metadata
- Export a snapshot for the relevant interval of time
- Requirements: millions of records in MB-sized files

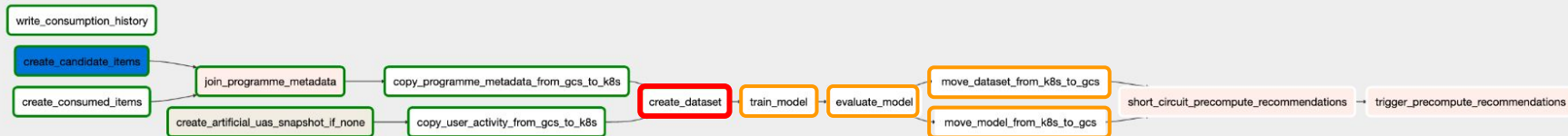
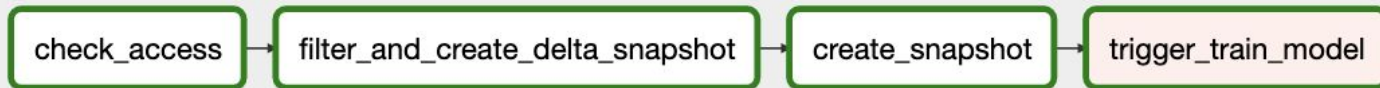


Mostly implemented using subclasses of the **Dataflow Operator** class

Obstruction 2 When the user activity workflow failed



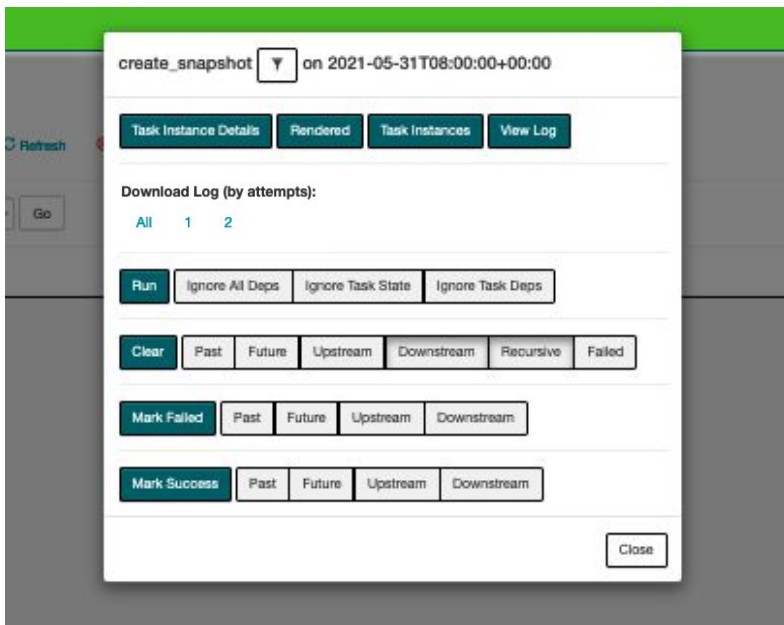
Obstruction 2 When the user activity workflow failed



Obstruction 2 When the user activity workflow failed

Troubleshooting

- The volume of user activity meant to train the model had **doubled!**



A dark, atmospheric background image featuring several hot air balloons floating against a cloudy sky. The balloons are silhouetted, and the overall tone is moody and dramatic.

Obstruction 2 When the user activity workflow failed

What happened

- Dataflow took longer than expected to run a job triggered by Airflow
- Airflow retried
- Both jobs completed successfully - and output the data in the same directory!
- The setup to train the model didn't expect to handle such spike in the volume of data and failed

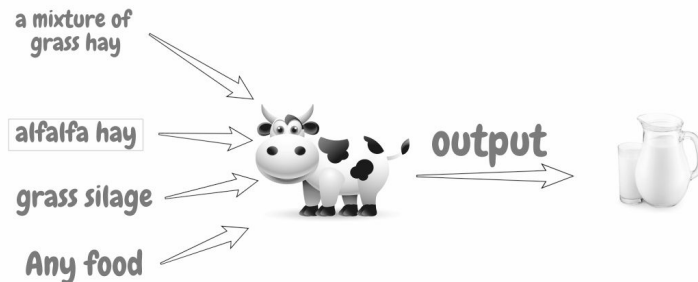
Obstruction 2 When the user activity workflow failed

What happened

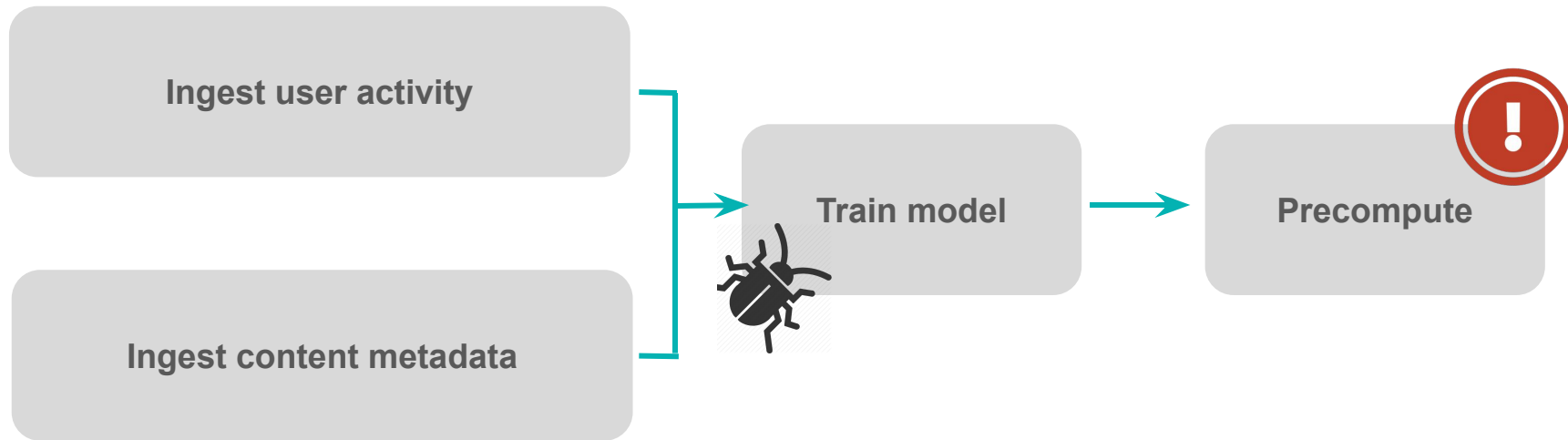
- Dataflow took longer than expected to run a job triggered by Airflow
- Airflow retried
- Both jobs completed successfully - and output the data in the same directory!
- The setup to train the model didn't expect to handle such spike in the volume of data and failed

Solution

- Have idempotent tasks
- Clear the target path before processing a task



Obstruction 3 When precompute failed due to training



Obstruction 3 When precompute failed due to training

```
graph LR; A[compute_predictions] --> B[copy_mass_insertion_from_gcs_to_k8s]; B --> C[create_redis_dump]; C --> D[copy_dump_from_workflows_volume_to_gcs]; D --> E[copy_recs_from_gcs_to_api_volume]; E --> F[restart_cache];
```

compute_predictions → copy_mass_insertion_from_gcs_to_k8s → create_redis_dump → copy_dump_from_workflows_volume_to_gcs → copy_recs_from_gcs_to_api_volume → restart_cache

Obstruction 3 When precompute failed due to training

On DAG: precompute_recommendations_xantus_music_mixes_1_1 schedule: None

Graph View Tree View Task Duration Task Times Landing Times Gantt Details Code Trigger DAG Refresh Delete

Task Instance: compute_predictions 2021-07-08 00:00:00

Task Instance Details Rendered Template Log XCom

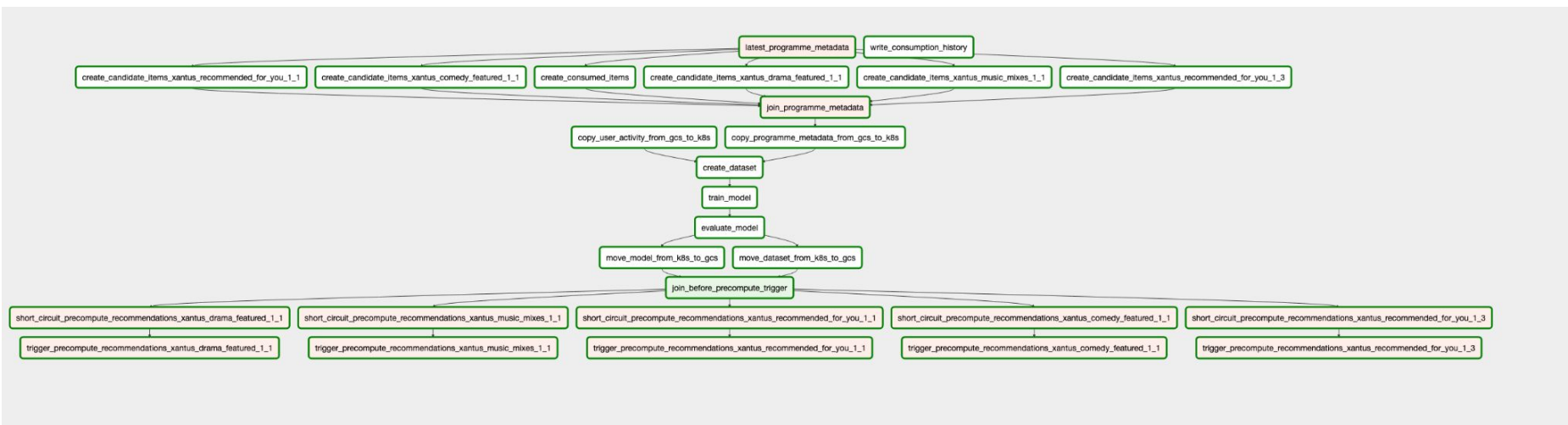
Log by attempts 2 Toggle wrap Jump to end

```
*** Reading remote log from gs://europe-west1-datalab-sounds-a20e55f2-bucket/logs/precompute_recommendations_xantus_music_mixes_1_1/compute_predictions/2021-07-08T00:00:00+00:00/
[2021-07-08 12:20:23,098] [taskinstance.py:671] INFO - Dependencies all met for <TaskInstance: precompute_recommendations_xantus_music_mixes_1_1,compute_predictions 2021-07-08T00:00:00+00:00>
[2021-07-08 12:20:23,175] [taskinstance.py:671] INFO - Dependencies all met for <TaskInstance: precompute_recommendations_xantus_music_mixes_1_1,compute_predictions 2021-07-08T00:00:00+00:00>
[2021-07-08 12:20:23,176] [taskinstance.py:681] INFO -
=====
[2021-07-08 12:20:23,176] [taskinstance.py:882] INFO - Starting attempt 1 of 1
[2021-07-08 12:20:23,176] [taskinstance.py:883] INFO -
=====
[2021-07-08 12:20:23,231] [taskinstance.py:962] INFO - Executing <Task(CustomDataflowPythonOperator): compute_predictions> on 2021-07-08T00:00:00+00:00
[2021-07-08 12:20:23,235] [standard_task_runner.py:54] INFO - Started process 257717 to run task
[2021-07-08 12:20:23,258] [standard_task_runner.py:77] INFO - Running: ['airflow', 'run', 'precompute_recommendations_xantus_music_mixes_1_1', 'compute_predictions', '2021-07-08T00:00:00+00:00']
[2021-07-08 12:20:23,260] [standard_task_runner.py:78] INFO - Job 136629: Subtask compute_predictions
[2021-07-08 12:20:23,792] [logging_mixin.py:112] INFO - Running <TaskInstance: precompute_recommendations_xantus_music_mixes_1_1,compute_predictions 2021-07-08T00:00:00+00:00 [run]
[2021-07-08 12:20:23,938] [gcs_utils.py:49] INFO - attempting to delete from file url list ['gs://datalab-sounds-prod-6c75-data/recommenders/xantus/variants/music-mixes/1.1/preco
[2021-07-08 12:20:24,093] [operators.py:880] INFO - Using Python interpreter: /tmp/dataflow_venv/bin/python
[2021-07-08 12:20:24,093] [operators.py:813] INFO - Running job: compute_predictions_xantus_music_mixes_1_1, formatted_options: {'labels': {'service': 'recommender', 'component': '
[2021-07-08 12:20:24,095] [gcp_dataflow_hook.py:121] INFO - Running command: /tmp/dataflow_venv/bin/python /home/airflow/gcs/dags/predictions/compute_predictions.py --labels=serve
[2021-07-08 12:20:24,233] [gcp_dataflow_hook.py:158] INFO - Start waiting for Dataflow process to complete.
[2021-07-08 12:31:00,625] [gcp_dataflow_hook.py:133] WARNING - b'warning: sdsl: standard file not found: should have one of README, README.rst, README.txt, README.md\n\nwarning:
[2021-07-08 12:31:00,676] [taskinstance.py:1152] ERROR - Dataflow failed with return code 1
Traceback (most recent call last):
  File "/usr/local/lib/airflow/airflow/models/taskinstance.py", line 980, in _run_raw_task
    result = task_copy.execute(context=context)
  File "/home/airflow/gcs/dags/dataflow/operators.py", line 87, in execute
    py_interpreter=self.py_interpreter
  File "/home/airflow/gcs/dags/dataflow/hooks.py", line 27, in start_python_dataflo
    self.start_dataflow(variables, name, [py_interpreter] + py_options + [dataflow], label=formatter
  File "/usr/local/lib/airflow/airflow/contrib/hooks/gcp_api_base_hook.py", line 363, in wrapper
    return func(self, *args, **kwargs)
  File "/home/airflow/gcs/dags/dataflow/hooks.py", line 33, in _start_dataflo
    job_id = _dataflow(cmd).wait_for_done()
  File "/usr/local/lib/airflow/airflow/contrib/hooks/gcp_dataflow_hook.py", line 179, in wait_for_done
    self._proc.returncode)
Exception: Dataflow failed with return code
[2021-07-08 12:31:00,681] [taskinstance.py:1196] INFO - Marking task as FAILED. dag_id=precompute_recommendations_xantus_music_mixes_1_1, task_id=compute_predictions, execution_d
[2021-07-08 12:31:00,681] [google.cloud_logging.py:129] ERROR - {'success': False, 'task': 'compute_predictions', 'dag_id': 'precompute_recommendations_xantus_music_mixes_1_1', 't
[2021-07-08 12:31:02,179] [local_task_job.py:182] INFO - Task exited with return code 1
```


"message": "No such object:

datalab-sounds-prod-6c75-data/recommenders/xan
ntus/model/2021-07-08T00:00:00+00:00/xantus.p
kl"

Obstruction 3 When precompute failed due to training



Obstruction 3 When precompute failed due to training

move_model_from_k8s_to_gcs  on 2021-07-08T00:00:00+00:00

[Task Instance Details](#) [Rendered](#) [Task Instances](#) [View Log](#)

Download Log (by attempts):

[All](#) [1](#) [2](#) [3](#)

[Run](#) [Ignore All Deps](#) [Ignore Task State](#) [Ignore Task Deps](#)

[Clear](#) [Past](#) [Future](#) [Upstream](#) [Downstream](#) [Recursive](#) [Failed](#)

[Mark Failed](#) [Past](#) [Future](#) [Upstream](#) [Downstream](#)

[Mark Success](#) [Past](#) [Future](#) [Upstream](#) [Downstream](#)

[Close](#)

Obstruction 3 When precompute failed due to training

```
[2021-07-08 12:15:55,278] {logging_mixin.py:112}
INFO - Running <TaskInstance:
train_model.move_model_from_k8s_to_gcs
2021-07-08T00:00:00+00:00 [running]> on host
airflow-worker-867b96c854-jzqw7
```

```
[2021-07-08 12:15:55,422]
{gcp_container_operator.py:299} INFO - Using gcloud
with application default credentials.
[2021-07-08 12:15:57,286] {pod_launcher.py:173}
INFO - Event:
move-model-to-gcs-3deac821b57047619c1c9505ddc
5db18 had an event of type Pending
```

```
[2021-07-08 12:15:57,286] {pod_launcher.py:139}
WARNING - Pod not yet started:
move-model-to-gcs-3deac821b57047619c1c9505ddc
5db18
```

```
[2021-07-08 12:17:57,499] {taskinstance.py:1152}
ERROR - Pod Launching failed: Pod Launching failed:
Pod took too long to start
```

```
[2021-07-08 12:18:59,584] {pod_launcher.py:156} INFO
- b'gsutil -m rm
gs://datalab-sounds-prod-6c75-data/recommenders/xant
us/model/2021-07-08T00:00:00+00:00/xantus.pkl ||
true\n'
```

```
[2021-07-08 12:18:59,911] {pod_launcher.py:156} INFO
- b'gsutil -m mv
/data/recommenders/xantus/model/2021-07-08T00:00:0
0+00:00/xantus.pkl (...)
```

```
[2021-07-08 12:19:35,295] {pod_launcher.py:156} INFO
- b'Operation completed over 1 objects/4.7 GiB.
\n'
```

```
[2021-07-08 12:19:35,536] {pod_launcher.py:156} INFO
- b'rm -rf
/data/recommenders/xantus/model/2021-07-08T00:00:0
0+00:00/xantus.pkl\n'
```

```
[2021-07-08 12:19:36,687] {taskinstance.py:1071} INFO
- Marking task as SUCCESS.dag_id=train_model,
```

Obstruction 3 When precompute failed due to training

```
$ kubectl logs move-model-to-gcs-a0b5193a42e040aaa37b3ad82953ee29 -n xantus-training
gsutil -m rm gs://datalab-sounds-prod-6c75-data/recommenders/xantus/model/2021-07-08T00:00:00+00:00/xantus.pkl || true
CommandException: 1 files/objects could not be removed.

gsutil -m mv /data/recommenders/xantus/model/2021-07-08T00:00:00+00:00/xantus.pkl
gs://datalab-sounds-prod-6c75-data/recommenders/xantus/model/2021-07-08T00:00:00+00:00/xantus.pkl
Copying file:///data/recommenders/xantus/model/2021-07-08T00:00:00+00:00/xantus.pkl [Content-Type=application/octet-stream]...

Removing file:///data/recommenders/xantus/model/2021-07-08T00:00:00+00:00/xantus.pkl...
| [1/1 files][ 4.7 GiB/ 4.7 GiB] 100% Done 111.3 MiB/s ETA 00:00:00
Operation completed over 1 objects/4.7 GiB.

rm -rf /data/recommenders/xantus/model/2021-07-08T00:00:00+00:00/xantus.pkl
```

A dark, atmospheric background image featuring several hot air balloons floating against a cloudy sky. The balloons are silhouetted against the lighter clouds, creating a serene yet mysterious scene. The overall tone is dark, with the balloons providing points of interest in the upper half of the frame.

Obstruction 3 When precompute failed due to training

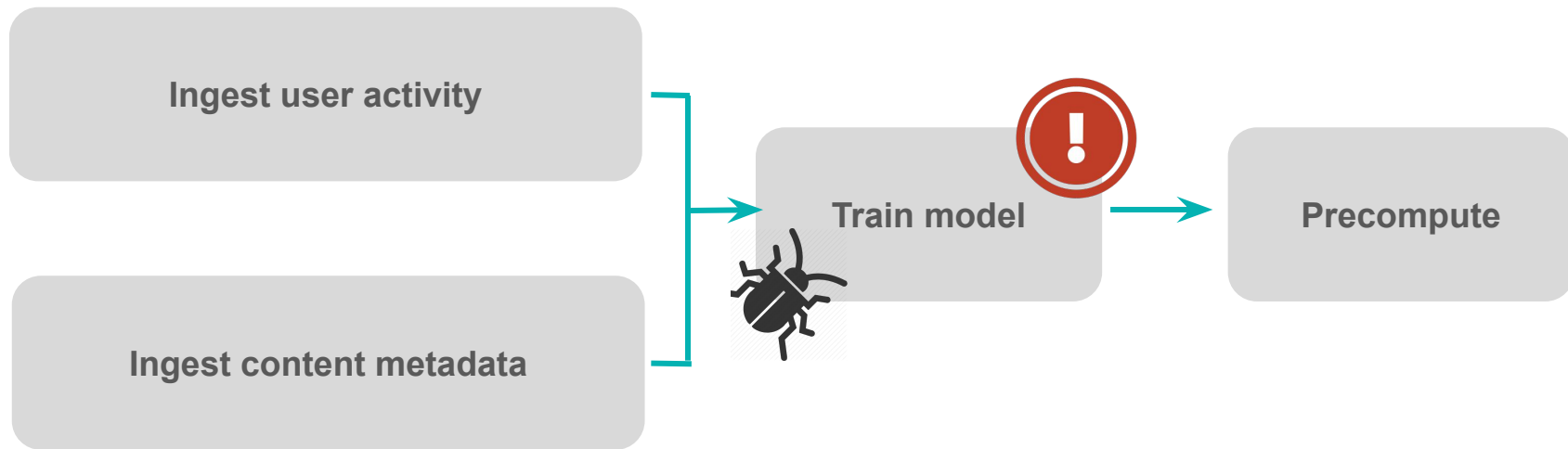
What happened

- The GKE node pool, where jobs were executed, was struggling
- Airflow Kubernetes Operator timed out after a long time waiting (Kubernetes didn't know)
- A new task retry was triggered
- Both pods run concurrently and due to how we implemented idempotency, the data was deleted - but the last task retry was successful

Solution

- Use newer version of the KubernetesPodOperator
- Confirm by the end of the task that the desired artefact exists

Obstruction 4 Intermittent DAG after an Airflow upgrade



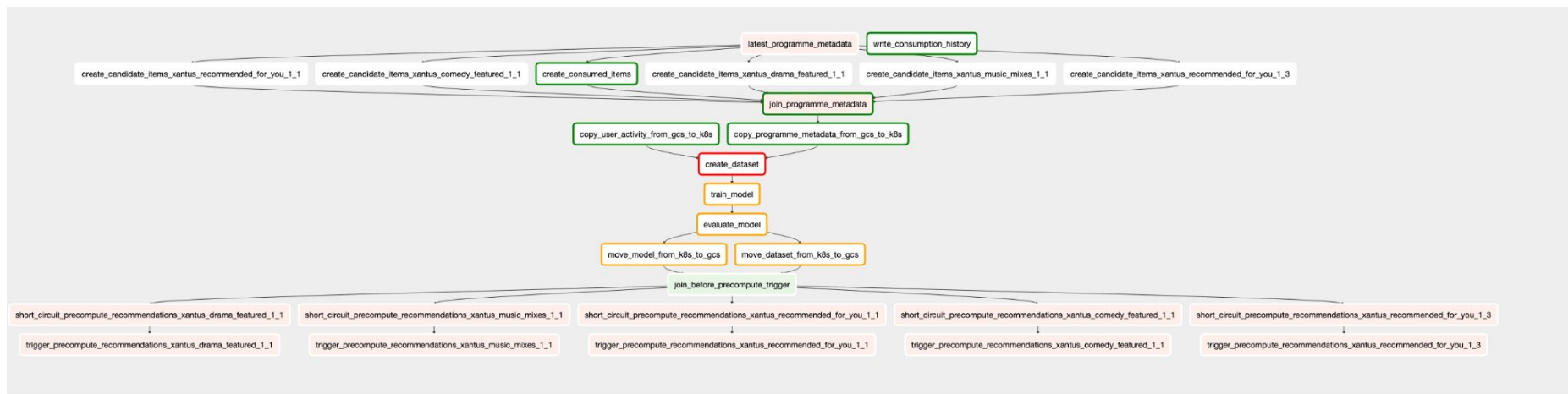
A dark, atmospheric background image featuring several hot air balloons floating against a cloudy sky. The balloons are silhouetted against the lighter clouds, creating a dramatic effect. The overall tone is dark and moody.

Obstruction 4 Intermittent DAG after an Airflow upgrade

What happened

- After upgrading from Airflow 1.10.4 to 1.10.12 some KubernetesPodOperator tasks became intermittent
- Legit running pods failed
- The logs seemed to show that new jobs were trying to reattach to previously existing Pods and failed

Obstruction 4 Intermittent DAG after an Airflow upgrade



Obstruction 4 Intermittent DAG after an Airflow upgrade

create_dataset on 2021-01-30T00:00:00+00:00

Task Instance Details **Rendered** **Task Instances** **View Log**

Download Log (by attempts):

[All](#) [1](#) [2](#) [3](#) [4](#)

Run **Ignore All Dps** **Ignore Task State** **Ignore Task Dps**

Clear **Past** **Future** **Upstream** **Downstream** **Recursive** **Failed**

Mark Failed **Past** **Future** **Upstream** **Downstream**

Mark Success **Past** **Future** **Upstream** **Downstream**

Close

HTTP response headers: HTTPHeaderDict({'Audit-Id': '133bc1f2-388b-490c-bdc6-34053685d5ee', 'Content-Type': 'application/json', 'Date': 'Sat, 30 Jan 2021 09:11:33 GMT', 'Content-Length': '231'})

HTTP response body:

```
b'{"kind": "Status", "apiVersion": "v1", "metadata": {}, "status": "Failure", "message": "container \"base\" in pod \"create-dataset-17a3b6f132e44544a836550be367c670\" is waiting to start: ContainerCreating", "reason": "BadRequest", "code": 400}'
```

(...)

```
"/opt/python3.6/lib/python3.6/site-packages/kubernetes/client/rest.py", line 231, in request
    raise ApiException(http_resp=r)
kubernetes.client.rest.ApiException: (400 Reason: Bad Request
```

Obstruction 4 Intermittent DAG after an Airflow upgrade

Solution

airflow.contrib.operators.kubernetes_pod_operator

Executes task in a Kubernetes POD

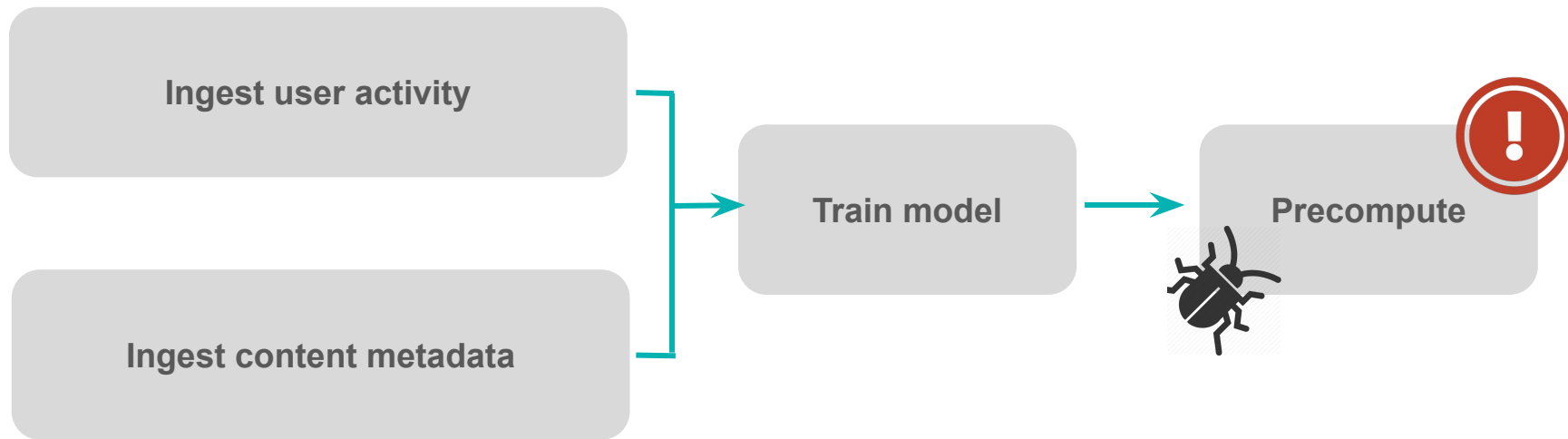
Module Contents

```
class airflow.contrib.operators.kubernetes_pod_operator.KubernetesPodOperator(namespace=None, image=None, name=None,
cmds=None, arguments=None, ports=None, volume_mounts=None, volumes=None, env_vars=None, secrets=None, in_cluster=None,
cluster_context=None, labels=None, reattach_on_restart=True, startup_timeout_seconds=120, get_logs=True,
image_pull_policy='IfNotPresent', annotations=None, resources=None, affinity=None, config_file=None, node_selectors=None,
image_pull_secrets=None, service_account_name='default', is_delete_operator_pod=False, hostnetwork=False,
tolerations=None, configmaps=None, security_context=None, pod_runtime_info_envs=None, dnspolicy=None, schedulername=None,
full_pod_spec=None, init_containers=None, log_events_on_failure=False, do_xcom_push=False, pod_template_file=None,
priority_class_name=None, *args, **kwargs)[source]
```

Bases: airflow.models.BaseOperator

https://airflow.apache.org/docs/apache-airflow/1.10.12/_api/airflow/contrib/operators/kubernetes_pod_operator/index.html

Obstruction 5 When the Dataflow job said no



Obstruction 5 When the Dataflow job said no

On DAG: precompute_recommendations

Graph View

Tree View

Task Duration

Task Tries

Landing Times

Gantt

Details

Code

Trigger DAG

Refresh

Delete

failed

Base date: 2021-02-28 16:00:01

Number of runs: 25

Run: trig__2021-02-28T16:00:00+00:00

Layout: Left->Right

Go

CustomDataFlowPythonOperator

GKEPodOperator

KubectiOperator

compute_predictions

copy_mass_insertion_from_gcs_to_k8s

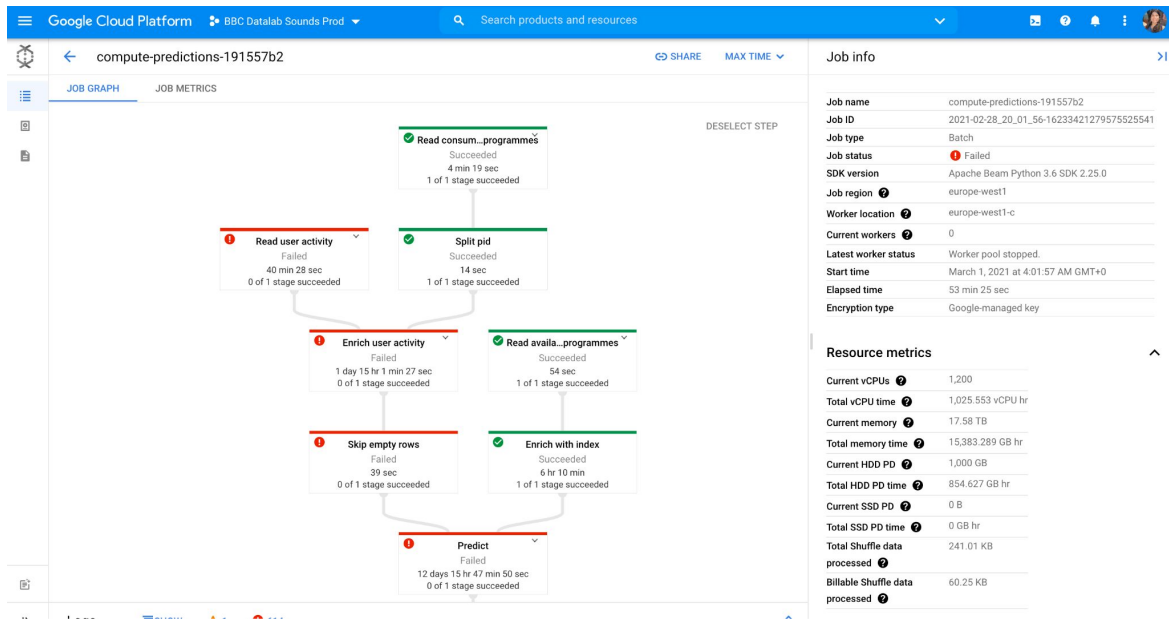
create_redis_dump

copy_dump_from_workflows_volume_to_gcs

copy_recs_from_gcs_to_api_volume

restart_cache

Obstruction 5 When the Dataflow job said no



OSError: [Errno 28] No space left on device During handling

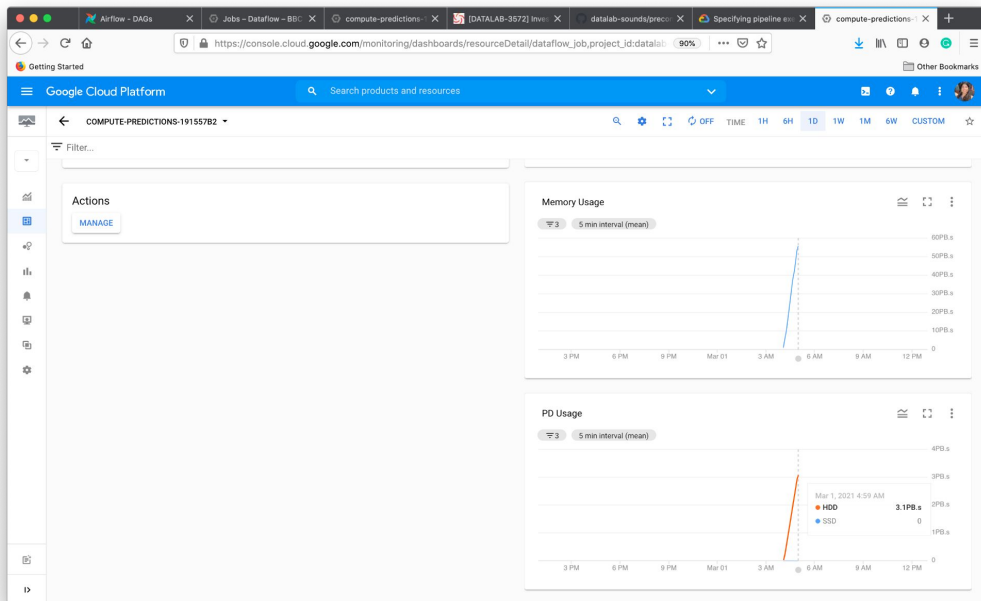
Obstruction 5 When the Dataflow job said no

	successful compute-predictions-21628eba 2021-02-24_20_06_12-17726452853028084617 25 February 2021	unsuccessful compute-predictions-191557b2 2021-02-28_20_01_56-16233421279575525541 1st March 2021
consumption-history/	1.87 GiB	1.88 GiB
consumed-items/	55.63 MiB	55.68 MiB
candidate-items/rfy/	25.06 MiB	25.09 MiB
xantus.pkl	4.67 GiB	4.69 GiB
item_features.npz	1.73 MiB	1.73 MiB
mapping.json	473.84 MiB	476.01 MiB

If a batch job uses Dataflow Shuffle, then the default is 25 GB; otherwise, the default is 250 GB.

<https://cloud.google.com/dataflow/docs/guides/specifying-exec-params#python>

Obstruction 5 When the Dataflow job said no



Job info

Job name	compute-predictions-21628eba
Job ID	2021-02-24_20_06_12-17726452853028084617
Job type	Batch
Job status	✓ Succeeded
SDK version	Apache Beam Python 3.6 SDK 2.25.0
Job region	eu-west-1
Worker location	eu-west-1-b
Current workers	0
Latest worker status	Worker pool stopped.
Start time	February 25, 2021 at 4:06:14 AM GMT+0
Elapsed time	2 hr 13 min
Encryption type	Google-managed key






Resource metrics

Current vCPUs	1,200
Total vCPU time	2,633.094 vCPU hr
Current memory	17.58 TB
Total memory time	39,496.416 GB hr
Current HDD PD	9.77 TB
Total HDD PD time	21,942.453 GB hr
Current SSD PD	0 B
Total SSD PD time	0 GB hr

Labels

airflow-version	v1-10-12-cnmnosr
-----------------	------------------

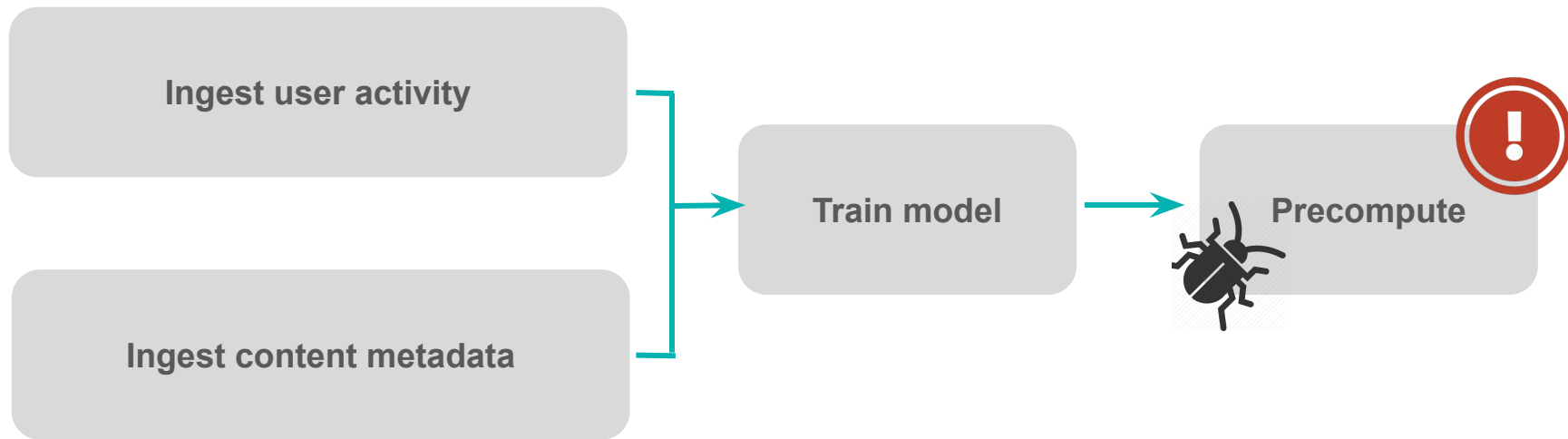
Obstruction 5 When the Dataflow job said no

Google Cloud Platform   Search products and resources   	
← Case 27052933 ✎ REOPEN	Attributes
<p>Dataflow job started using the shuffle service unexpectedly</p> <p>Tatiana Al-Chueyr tatiana.alchueyr@bbc.co.uk March 1, 2021 at 11:37:43 AM GMT+0</p> <p>At the moment, our end-users are being offered stale recommendations in production. The reason for this is because a Dataflow job failed (ID: 2021-02-28_20_01_56-16233421279575525541).</p> <p>It seems the Dataflow job run out of disk, since the the job logs contain the error:</p> <pre>... RuntimeError: OSError: [Errno 28] No space left on device [while running 'Predict/Predict'] ...</pre> <p>When comparing this unsuccessful (ID: 2021-02-28_20_01_56-16233421279575525541) run with the previous successful run (ID: 2021-02-24_20_06_12-17726452853028084617), we noticed the data shuffle service was *automatically* enabled in the failed job. We realised this by looking at the Job resource metrics and seeing the following unexpected metrics: "Total Shuffle data processed" and "Billable Shuffle data processed".</p> <p>We were surprised to see the shuffling charges because we did not make any changes to the code itself - nor did we update the SDK or environment. We are aware that, by default, the shuffle service reduces the</p>	<p>Project datalab-sounds-prod-6c75</p> <p>Priority P2</p> <p>Status Closed</p> <p>Category Big Data</p> <p>Component Cloud Dataflow</p> <p>CC</p>

Solution

```
--experiments=shuffle_mode=appliance
```

Obstruction 6 The never ending Dataflow job



Obstruction 6 The never ending Dataflow job

datalab_devops_int - Jun 14th



google cloud monitoring APP 6:46 PM

Incident #0.m3oacbpboewe is ongoing

[Composer - datalab-sounds-int-bleu] Workflow failure

Workflow Runs for datalab-monitoring-int-f09d Cloud Composer Workflow labels {project_id=datalab-monitoring-int-f09d, workflow_name=datalab-sounds-int-bleu.precompute_recommendations_xantus_recommended_for_you_1_1} is above the threshold of 0.000 ...

compute_predictions

copy_mass_insertion_from_gcs_to_k8s

create_redis_dump

copy_dump_from_workflows_volume_to_gcs

copy_recs_from_gcs_to_api_volume

restart_cache

Obstruction 6 The never ending Dataflow job

```
[2021-05-13 11:52:34,884] {gcp_dataflow_hook.py:1 (...)} Traceback (most recent call last):\n  File\n  "/home/airflow/gcs/dags/predictions/compute_predictions.py", line 337, in run\n    return pipeline\n  File\n  "/tmp/dataflow_venv/lib/python3.6/site-packages/apache_beam/pipeline.py", line 569, in __exit__\n    self.result.wait_until_finish()\n  File\n  "/tmp/dataflow_venv/lib/python3.6/site-packages/apache_beam/runners/dataflow/dataflow_runner.py",\n  line 1650, in wait_until_finish\n    self)\napache_beam.runners.dataflow.dataflow_runner.DataflowRuntimeException: Dataflow pipeline\nfailed. State: FAILED,\n(..)
```

The job failed because a work item has failed 4 times. Look in previous log entries for the cause of each one of the 4 failures. For more information, see <https://cloud.google.com/dataflow/docs/guides/common-errors>. The work item was attempted on these workers:

- compute-predictions-xantu-05130255-opno-harness-mtj1\n Root cause: The worker lost contact with the service.,
- compute-predictions-xantu-05130255-opno-harness-2x4v\n Root cause: The worker lost contact with the service.,
- compute-predictions-xantu-05130255-opno-harness-5gkd\n Root cause: The worker lost contact with the service.,
- compute-predictions-xantu-05130255-opno-harness-2t1n\n Root cause: The worker lost contact with the service.'

Obstruction 6 The never ending Dataflow job

● Name	Type	End time	↓ Elapsed time	Start time
○ compute-predictions-xantus-music-mixes-1-1-87d98f40	Batch	May 13, 2021, 10:44:25 AM	3 days 5 hr	May 10, 2021, 5:21:29 AM
○ compute-predictions-xantus-music-mixes-1-1-	Batch	May 13, 2021,	5 hr 12 min	May 13, 2021,

A dark, atmospheric background image featuring several hot air balloons floating against a cloudy sky. The balloons are silhouetted against the lighter clouds, creating a sense of depth and movement. The overall tone is moody and mysterious.

Obstruction 6 The never ending Dataflow job

Solution

- Backport the latest Google Cloud operators in Apache Airflow
- Particularly:
 - DataflowCreatePythonJobOperator
 - DataflowJobStatusSensor

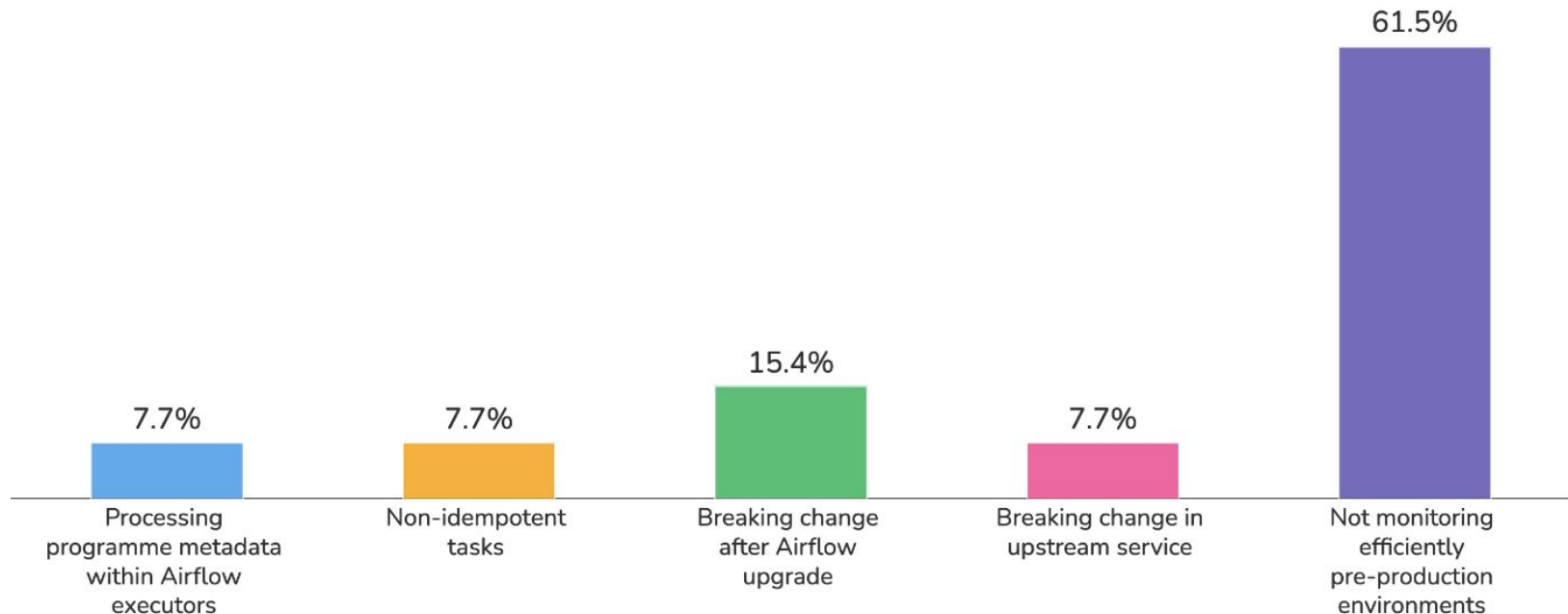
<https://medium.com/google-cloud/backporting-google-cloud-operators-in-apache-airflow-34b6c9efffc8>

Quiz time which do you reckon was the most costly issue?

- A. Processing programme metadata within Airflow executors
- B. Non-idempotent tasks
- C. Breaking change after Airflow upgrade
- D. Breaking change in upstream service
- E. Not monitoring efficiently pre-production environments



Quiz time which do you reckon was the most costly issue?



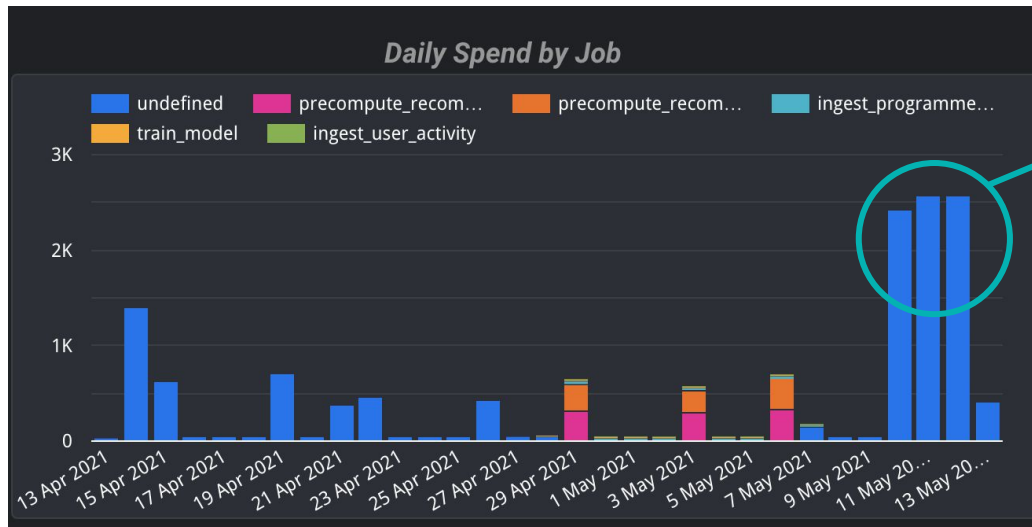
* responses from Airflow Summit 2021 participants, during the presentation

Quiz time which do you reckon was the most costly issue?

- A. Processing programme metadata within Airflow executors
- B. Non-idempotent tasks
- C. Breaking change after Airflow upgrade
- D. Breaking change in upstream service
- E. Not monitoring efficiently pre-production environments



Quiz time which do you reckon was the most costly issue?



The never ending Dataflow Job, triggered with **DataflowOperator** in our int environment, run for over 3 days and costed over £12k



Hygienic workflows throughout development

Hygienic pipelines

A dark, atmospheric background image featuring several hot air balloons floating against a cloudy sky. The balloons are silhouetted and have various patterns, including stripes and solid colors. The overall tone is moody and cinematic.

Smell 1 To plugin or not to plugin

- **Requirement:**
 - Have common packages across multiple DAGs without using PyPI or similar
- **Attempt:**
 - Use plugins to expose those
 - Deploy using
 - `gcloud composer environments storage dags import .`
 - `gcloud composer environments storage dags import .`
- **Problems**
 - Lots of broken deployments
 - Unsynchronised upload of **plugins** and **DAGs** to **Composer** web server & workers
 - Issues in enabling [DAG serialisation](#) with **plugins**

A decorative header image featuring several hot air balloons of various patterns and colors floating against a dark, cloudy sky. The balloons are scattered across the top of the slide, with some appearing larger and more detailed than others.

Smell 1 To plugin or not to plugin

- **Solution:**
 - Stop using plugins
 - Use standard Python packages
 - Upload them using Google Cloud to the Bucket / path
 - `gsutil rm (...)`
 - `gsutil cp (...)`

A dark, atmospheric background image featuring several hot air balloons floating against a cloudy sky. The balloons are silhouetted, with some showing distinct patterns like stripes or checks. The overall tone is moody and professional.

Smell 2 Configuration (mis)patterns

- **Requirement:**
 - Have a strategy for handling environment-specific and common configuration
- **Attempt:**
 - To use Environment variables to declare each env-specific variable
 - To deploy using Terraform
 - To declare common configuration in the DAGs
- **Problems**
 - Each variant updated represented a Cloud Composer deployment
 - Redundant configuration definition across DAGs and multiple utilities modules
 - Hard to identify the data sources / targets paths

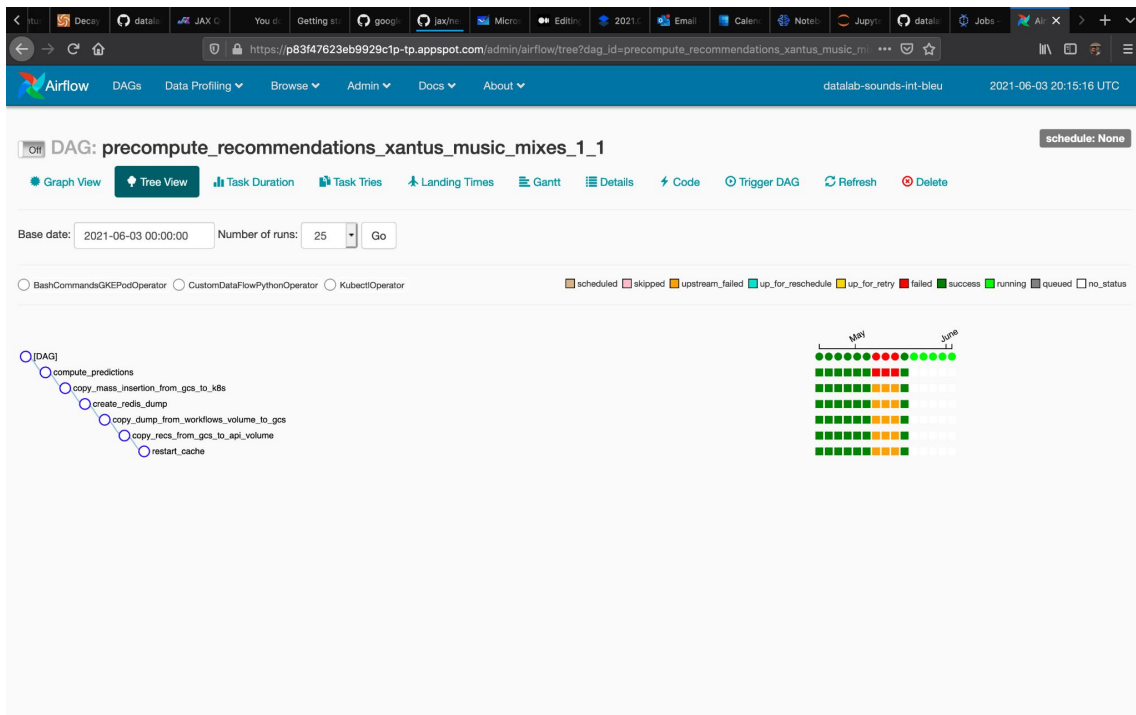
Smell 2 Configuration (mis)patterns

- **Solution:** have configuration files loaded into Airflow variables with paths and variables

```
70 "precompute_recommendations_dag": {
71   "start_date": "2020-12-07T06:00:00.00",
72   "redis_image": "marketplace.gcr.io/google/redis5",
73   "dataflow": {
74     "num_workers": "40",
75     "notused_comments": "30 vCPU with 15 GB RAM per vCPU",
76     "machine_type": "custom-30-460800-ext"
77   },
78   "paths": {
79     "candidate_items_gcs_prefix": "%(gcs_base_path)s/programme-metadata/candidate-items/{{ ts }}/",
80     "workflows_redis_mass_path": "%(gcs_base_path)s/precomputed-recommendations/redis-mass-insertion/{{ ts }}/",
81     "workflows_redis_dump_path": "%(gcs_base_path)s/precomputed-recommendations/redis-dump/{{ ts }}/",
82     "api_redis_dump_path": "%(volume_base_path)s/precomputed_recommendations/",
83     "precompute_recommendations_dag_id": "precompute_recommendations_{{identifier}}s"
84   }
85 },
86 "ingest_user_activity_dag": {
87   "start_date": "2020-12-07T06:00:00.00",
88   "ingest_interval_hours": "8",
89   "snapshot_interval_days": "120",
90   "max_active_dag_runs": "1",
91   "ingest_catchup": "1",
92   "user_activity_bucket": "datalab-sounds-data-dev-ff72_uas",
93   "user_activity_prefix": "dev_test_key",
94   "paths": {
95     "deltas_prefix": "user-activity/uas/deltas/",
96     "delta_snapshot_path": "user-activity/uas/deltas/{{ ts }}/{{formatted_ingest_interval}}s/",
97     "list_original_paths": "workflows/user-activity-to-process/original/{{ ts }}/{{formatted_ingest_interval}}s/",
98     "user_activity_snapshots_path": "user-activity/uas/snapshots/{{ ts }}/{{formatted_snapshot_interval}}s/",
99     "list_deltas_path": "workflows/user-activity-to-process/deltas/{{ ts }}/{{formatted_snapshot_interval}}s/"
100   }
101 },
```

<https://www.astronomer.io/guides/dynamically-generating-dags>

Smell 2 Configuration (mis)patterns



Takeaways **Avoiding live incidents**



- ❑ Keep processing **out** of **Airflow executors**
- ❑ **Idempotency** matters - and it can be hard!
- ❑ **Backporting** is better than sticking to the past
- ❑ Reviewing **release notes** can help avoid live incidents
- ❑ **Monitoring pre-production** environments can **save money**

Takeaways **Keeping the house clean**



- ❑ Avoid **plugins**
- ❑ A **delete-deploy** approach can avoid problems
- ❑ Early **configuration-driven approach** saves time

A dark, atmospheric background image featuring several hot air balloons floating against a cloudy sky. The balloons are silhouetted against the lighter clouds, creating a sense of depth and movement. The overall tone is moody and professional.

Much more than **obstructions**

With the help of **Apache Airflow**, Datalab:

- Was able to end a contract of the BBC, with an **external** recommendation service, by increasing in **59% the audience engagement**
- Serves **daily millions** of **personalised** recommendations to the **BBC audiences**
- Built a **configurable** Machine Learning pipeline **agnostic of the model**
 - Constantly **adds new variants** and **extends workflows**

Thank you!

Tatiana Al-Chueyr

@tati_alchueyr