

Welcome

Berlin dbt November Meetup

Hosts: Kumar Aman & Sowmia Naraynan Siva Subramanian / *Tier Mobility SE*

Join dbt community Slack

getdbt.com/community

#local-berlin



Kumar Aman

*Team Lead Analytics Platform,
Tier Mobility SE*

(he/him)

*I let my mind wander and it never came
back so I like going wandering myself to try
finding where it went.*



kumar.aman@tier.app



Linkedin.com/in/kumar-aman-1011

Sowmia Naraynan Sivasubramanian

*Head of Analytics Platform,
Tier Mobility SE*

(he/him)

*I believe in 80/20 rule. Yes, it took
80% of my time to fill up the fun fact
here and you know the rest..*



 sowmia.subramanian@tier.app

 linkedin.com/in/sowmianarayan

Agenda

- 6 - 6:50 pm - Networking, Pizza and Drinks
- 6:50 - 7 pm - Welcome Remarks by Kumar & Sowmi
- 7 - 7:30 pm - From Metadata to Metrics: Utilizing dbt API to create actionable metrics and SLOs for your data org with Marian & Jerry
- 7:30 - 8 pm - Minimum Viable Stacks - Archetyping the Modern Data Stack for Startups with Saman
- 8 - 9 pm - Follow-up Q&A, drinks and networking



Icebreaker

Two truths and a lie

- Tier was founded in 2019 by Lawrence Leuschner, Matthias Laug & Julian Blessin
- Tier is operating in more than 33 countries and 550 cities
- Tier has been climate neutral since 2020

TIER IS THE LEADING MICRO-MOBILITY PROVIDER GLOBALLY



founded in **2018** by Lawrence Leuschner, Matthias Laug & Julian Blessin



active in **>550 cities**
in 33+ countries, HQ in Berlin



300,000+ e-scooters,
e-mopeds, bikes, cargo bikes
& e-bikes



More than **48 million car**
trips saved since 2018



100% climate neutral
through sustainable ops &
long vehicle lifetime



Our focus:

Providing the
safest, most
equitable & most
sustainable
mobility solution.



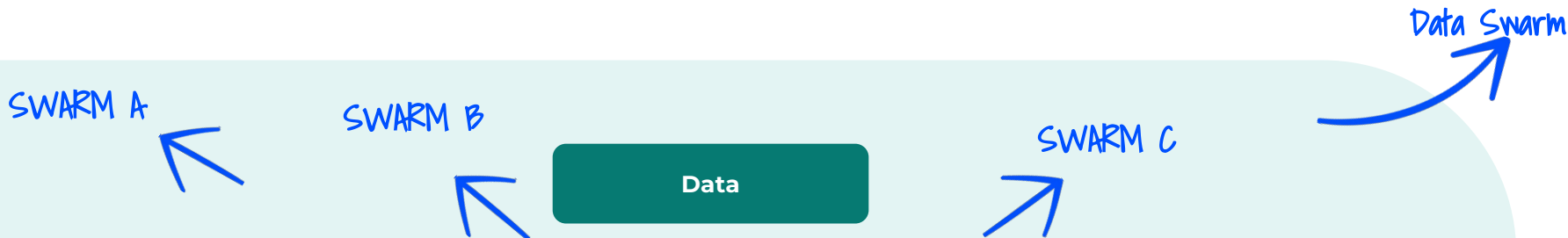
TIER

OUR VEHICLES



TIER

DATA TEAM SETUP



Consumer Analytics

Acquisition

Growth

Ride Experience

City Analytics

Public Policy

Sustainability

Data Sharing

City Product

Company Analytics

Operations

MRE

Supply Chain

Finance & Legal

Markets

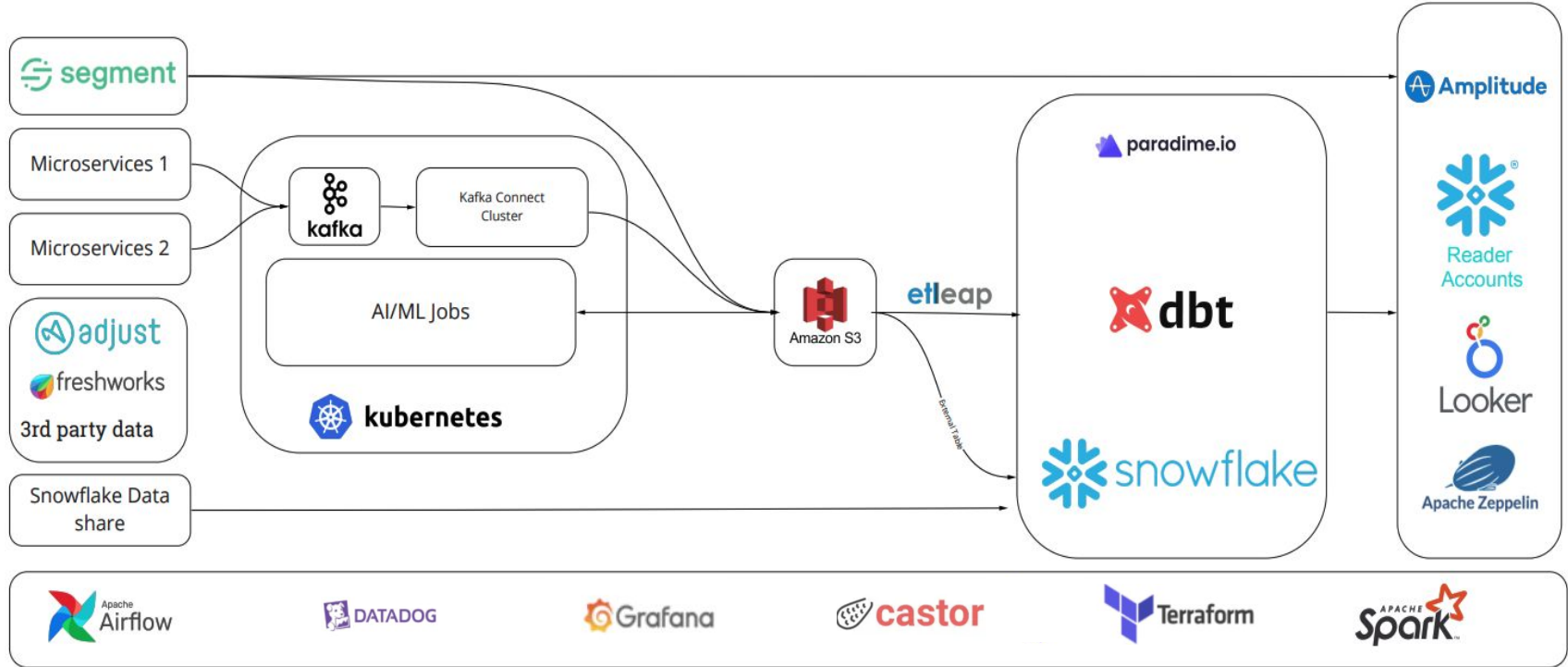
People & IT

Data Platform

Analytics Platform

Data Engineering

DATA WORKFLOW



TIER

WE ARE HIRING, COME JOIN THE RIDE!



- We are hiring and offer remote work from the many countries such as Germany, Spain, Netherlands, United Kingdom, Poland, Hungary, Greece, Austria.
- Being part of the team means you're part of its success.
 - ◆ Our ESOP program gives you an opportunity to have a share in the company
 - ◆ We support your growth with an annual development budget
- Diversity and inclusivity is super important to us, join initiatives & programs such as #WomenofTIER
- Profit from competitive compensation combined with perks like memberships with Urban Sports Club, Blinklist and Headspace.
- Surf your city with free rides on our e-scooters and e-bikes

TIER

WE ARE HIRING !!

Link to job board!

TIER



Join dbt Community Slack

getdbt.com/community

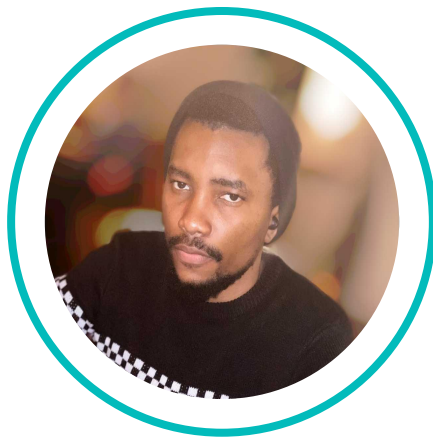
#local-berlin



Speaking Today



Marian Moreno Corral
Tier Mobility SE



Jerry Nwabulo
Tier Mobility SE



Saman Arefi
Project A Ventures

Maria de los Angeles Moreno Corral

*Analytics Engineer,
Tier Mobility SE*

(she/her)

*"I spent more time thinking about a fun
fact than preparing the presentation"*



maria.moreno@tier.app



linkedin.com/in/maria-morenocorral/

Jerry Nwabulo

*Senior Analytics Engineer,
Tier Mobility SE*

(he/him)

*“Working for a mobility company but
can’t ride a scooter.”*



✉ jerry.nwabulo@tier.app

in [Linkedin.com/in/jerry-nwabulo](https://www.linkedin.com/in/jerry-nwabulo)

From Metadata to Metrics:

Utilizing dbt API to create actionable metrics and SLOs for your data org






dbt November Meetup 2022

TIER



Agenda



-  Overview
-  Business Objectives
-  Data Collection Workflow
-  dbt Performance Dashboard
-  SLOs Overview
-  Data Pipeline For SLO Reporting



Overview



TIER



Overview



Are there tangible ways to measure the reliability of your data pipeline and the efficiency of your analytics engineers?

At Tier Mobility, we use the dbt API to fetch metadata of our models and take it all the way to our visualisation tool Looker conveying the state of our infrastructure to Data Management.

Metadata analysis helps our data teams to optimize their models, track the impact of their initiatives, and reduce costs.

Business Objectives

Business Objectives

Data Ownership

- ★ foster data ownership by assigning models to domain and individual
- ★ decentralize cost based on usage per domain

Quality Benchmarks

- ★ % data assets documented
- ★ % models incremental
- ★ % models with tests

How does getting
this information
**generate value for
the team?**

Performance Management

- ★ identify when WH needs to be scaled up
- ★ identify when perf. improvements are required

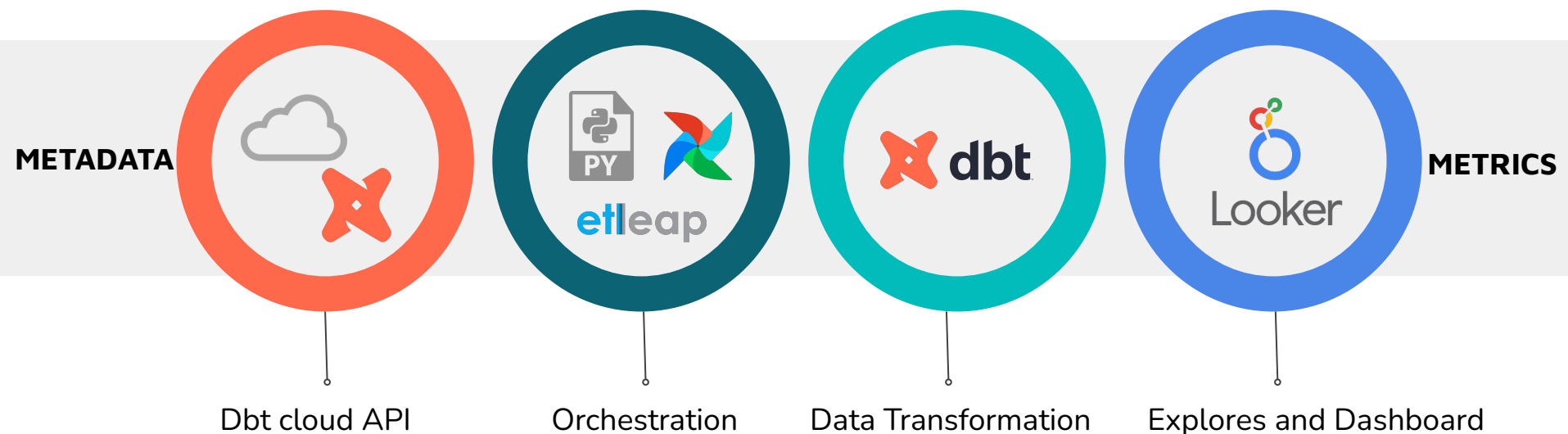
SLOs Tracking

- ★ monitor models' downtime > 3h to ensure the team SLO is being met

Data Collection Workflow

TIER

Data Collection Workflow





Data Collection Workflow

dbt Cloud API

Orchestration

Data Transformation

Looker Explores

The [dbt Cloud API](#) allows us to fetch **metadata from our dbt Cloud account**

Each dbt run generates **Artifacts** which can be downloaded as json files in the Artifacts section of dbt cloud jobs.

We use 2 Artifacts files:



manifest.json

Metadata about all the **resources available** in the dbt project (models, tests, macros, seeds,..) along with **their properties and configurations**

run_results.json

Information about the **resources that are executed** in a complete dbt invocation, and **the timing and status for each**

Models metadata



TIER

```
{
  "metadata": {
    "dbt_schema_version": "https://schemas.getdbt.com/dbt/run-results/v4.json",
    "dbt_version": "1.2.1",
    "generated_at": "2022-09-21T09:05:24.036464Z",
    "invocation_id": "fa6831a7-63e3-416d-b726-1c1deb971916",
    "env": {
      "DBT_CLOUD_PROJECT_ID": "17785",
      "DBT_CLOUD_RUN_ID": "83487188",
      "DBT_CLOUD_JOB_ID": "25559",
      "DBT_CLOUD_RUN_REASON": "Kicked off from UI by maria.moreno@tier.app",
      "DBT_CLOUD_RUN_REASON_CATEGORY": "other"
    }
  },
  "results": [
    {
      "status": "success",
      "timing": [
        {
          "name": "compile",
          "started_at": "2022-09-21T09:04:50.527043Z",
          "completed_at": "2022-09-21T09:04:50.538561Z"
        },
        {
          "name": "execute",
          "started_at": "2022-09-21T09:04:50.549353Z",
          "completed_at": "2022-09-21T09:04:52.035156Z"
        }
      ],
      "thread_id": "Thread-24",
      "execution_time": 1.7457904815673829,
      "adapter_response": {
        "message": "SUCCESS 1",
        "code": "SUCCESS",
        "rows_affected": 1,
        "query_id": "01a71d20-3201-5299-0000-9315207a130a"
      },
      "message": "SUCCESS 1",
      "failures": null,
      "unique_id": "model.bi.raw_dbt_model_run"
    }
  ],
}
```

Results for
each model that
ran

Timing of the first
executed model

```
{
  "status": "success",
  "timing": [
    {
      "name": "compile",
      "started_at": "2022-09-21T09:04:50.526657Z",
      "completed_at": "2022-09-21T09:04:50.538323Z"
    },
    {
      "name": "execute",
      "started_at": "2022-09-21T09:04:50.539370Z",
      "completed_at": "2022-09-21T09:04:52.105920Z"
    }
  ],
  "thread_id": "Thread-23",
  "execution_time": 1.845268726348877,
  "adapter_response": {
    "message": "SUCCESS 1",
    "failures": null,
    "unique_id": "model.bi.raw_dbt_model"
  },
  "message": "SUCCESS 1",
  "failures": null,
  "unique_id": "model.bi.raw_dbt_model"
},
{
  "status": "success",
  "timing": [
    {
      "name": "compile",
      "started_at": "2022-09-21T09:04:50.539370Z",
      "completed_at": "2022-09-21T09:04:52.105920Z"
    },
    {
      "name": "execute",
      "started_at": "2022-09-21T09:04:52.105920Z",
      "completed_at": "2022-09-21T09:04:53.646024Z"
    }
  ],
  "thread_id": "Thread-24",
  "execution_time": 1.5401040104010401,
  "adapter_response": {
    "message": "SUCCESS 1",
    "code": "SUCCESS",
    "rows_affected": 1,
    "query_id": "01a71d20-3201-5299-0000-9315207a130a"
  },
  "message": "SUCCESS 1",
  "failures": null,
  "unique_id": "model.bi.raw_dbt_model_run"
}
],
"elapsed_time": 36.20460247993469,
"args": {
  "write_json": true,
  "use_colors": true,
  "printer_width": 80,
  "version_check": true,
  "partial_parse": true,
  "static_parser": true,
  "profiles_dir": "/tmp/jobs/83487188/.dbt",
  "send_anonymous_usage_stats": true,
  "event_buffer_size": 100000,
  "quiet": false,
  "no_print": false,
  "profile": "user",
  "target": "default",
  "select": [
    "+fact_dbt_model_run"
  ],
  "full_refresh": true,
  "which": "run",
  "rpc_method": "run",
  "indirect_selection": "eager"
}
}
```

Second model that ran

Execution time in
seconds of the whole
run, all the models in
total

Run command



Data Collection Workflow

dbt Cloud API

Orchestration

Data Transformation

Looker Explores

- One **Python script** for pulling each Artifact file
- ***Run_results*** script runs **hourly**, and ***manifest* daily** in Airflow
- Both scripts push data to **ETLeap pipelines**



Scripts



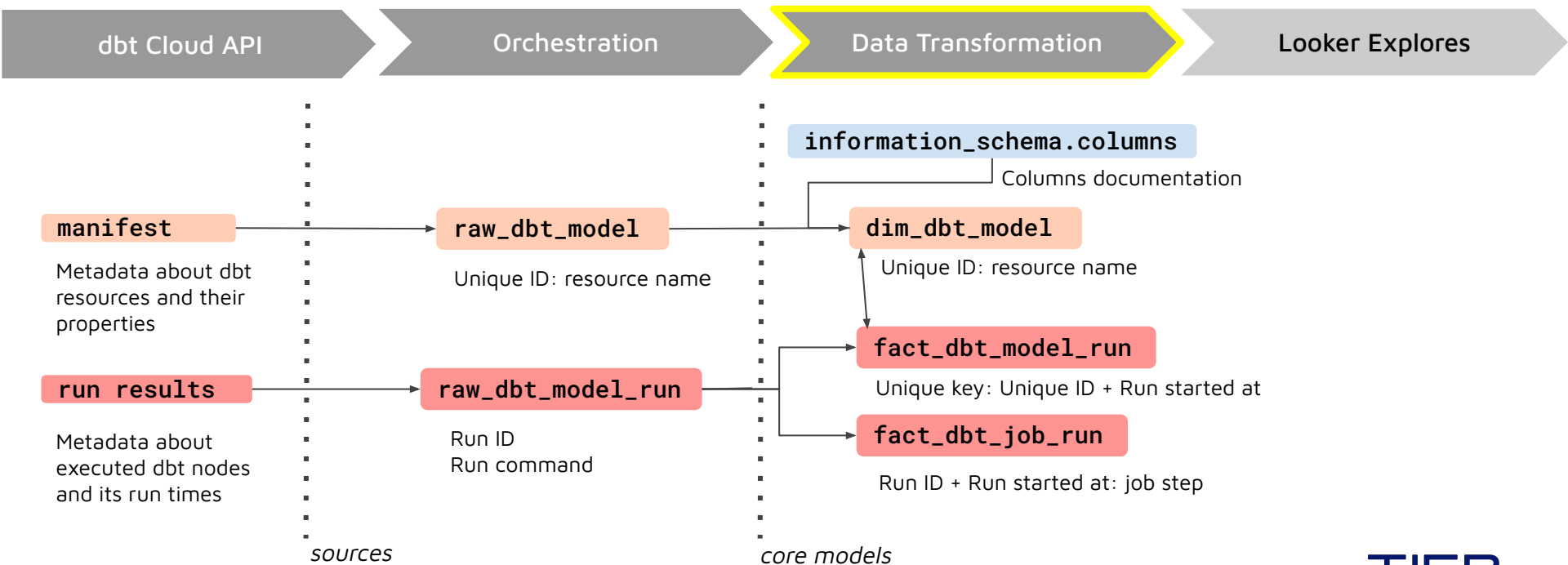
DAGs



Pipelines

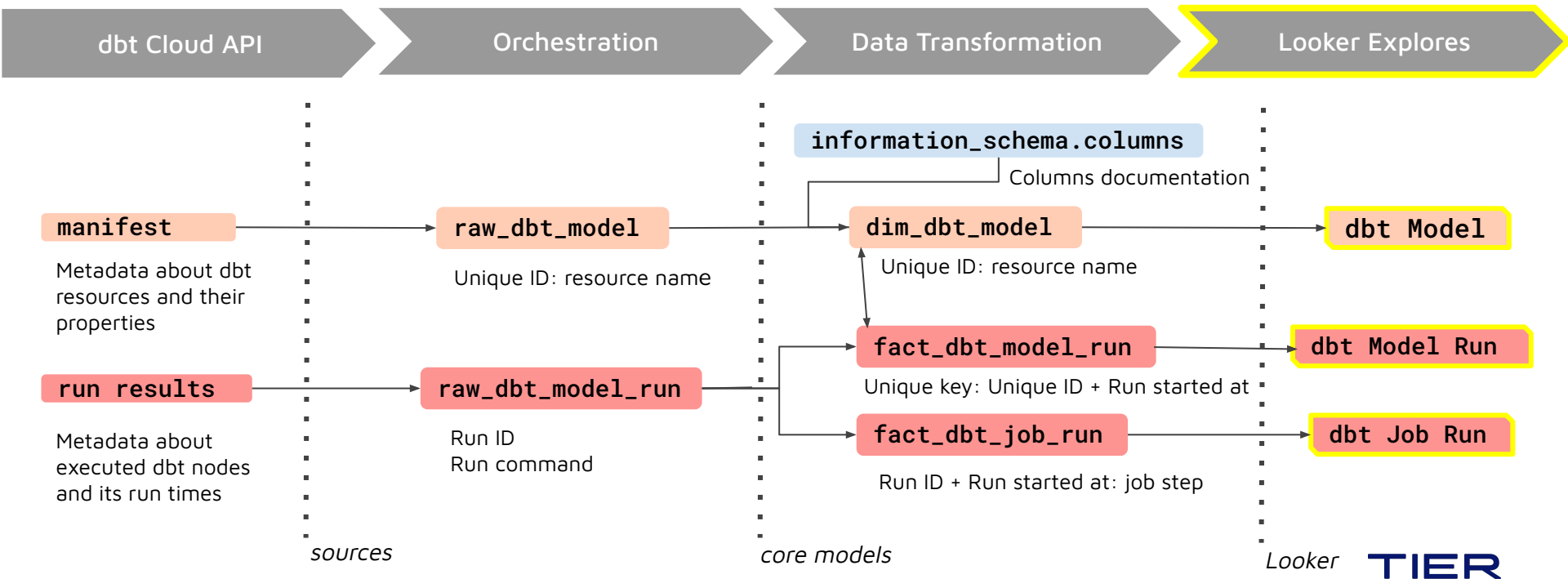


Data Collection Workflow





Data Collection Workflow



Data Collection Workflow - Summary

manifest.json



Metadata about the dbt resources (models, tests and seeds) and their properties - same data available in dbt docs.

Available fields:

- Resource type: model, test or seed
- Name and alias
- Materialization: view, table or incremental model
- Tags: nightly, weekly,...
- Description
- Customised metadata: owner and analytics group
- Unique key
- Pre- and post-hook

dim_dbt_model



Each row represents an available resource in our repository: model, test or seed, with its respective information.



dbt Model explore

Data Collection Workflow - Summary



run_results.json

Metadata about the execution times and results of executed models and scheduled jobs.

fact_dbt_model_run

Each row represents a different run of a dbt resource (model, test, seed), its timing and status

- Run id, timestamp and command
- Run reason: scheduled or kicked off by user
- Result status and execution times
- Downtime duration

dbt Model Run explore

fact_dbt_job_run

Each row represents an executed step within an job, with its timing and total status (success, error, warn,...)

- Job id, job started at, job name, run command
- Job execution times
- Number of models or tests which got an error, success, pass or warn within each job

dbt Job Run explore

dbt Performance Dashboard

TIER

dbt Performance Dashboard

This dashboard allows us to **monitor our dbt project's performance** and, therefore, to:

- have a better overview of data **ownership**
- monitor **documentation and test coverage**
- track **downtimes and errors**
- identify **worst-performing** dbt resources to prioritise their refactoring or fix

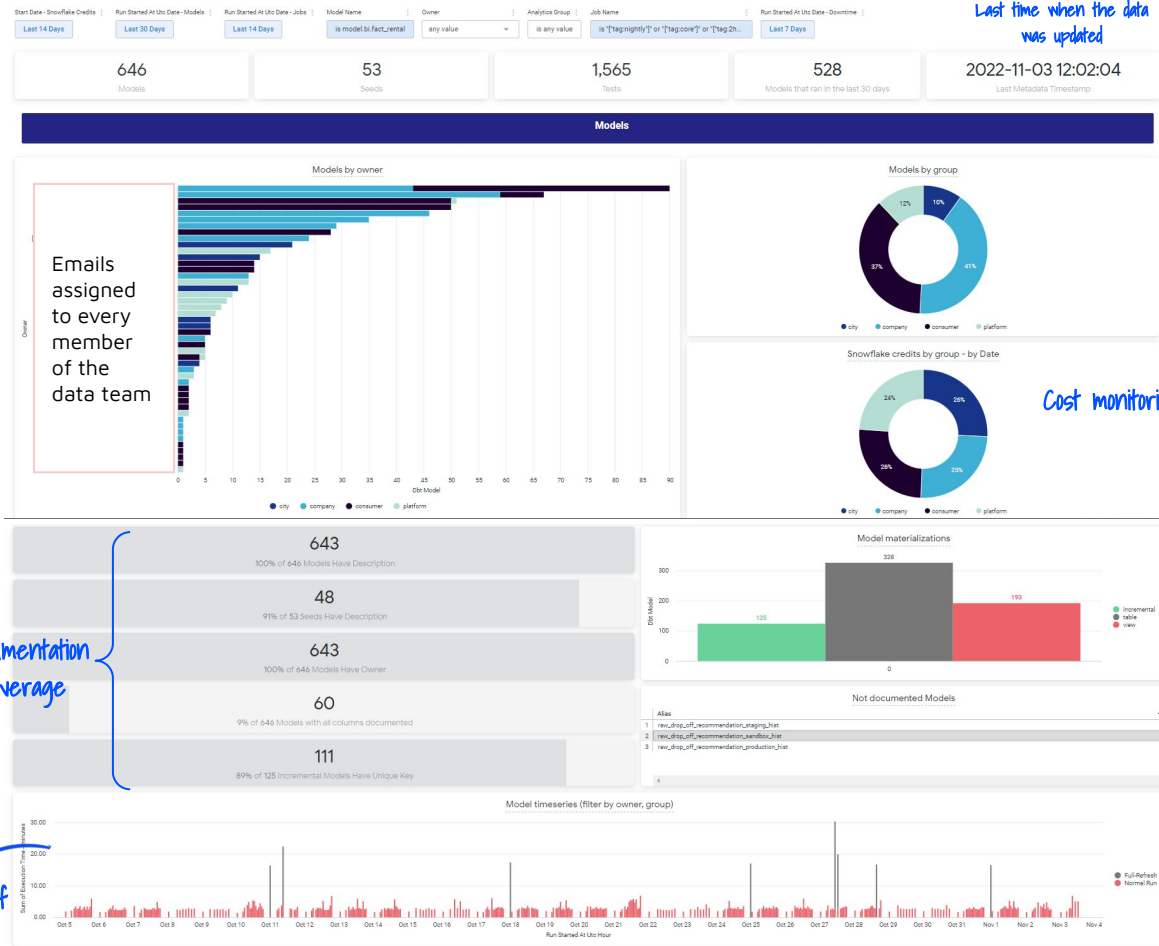


TIER

dbt Performance Dashboard

Models:

- Number of models by owner and group
- Costs by group
- Materializations
- Resources with description, owner, unique key,...
- Model run times



dbt Performance Dashboard



Scheduled Jobs:

- Errors or warnings in the jobs running on schedule
- Run times by scheduled job
- Job timeseries

dbt Performance Dashboard

Downtime:

Models with downtime, minimum, maximum and average downtime, and downtime duration by model.

Filtered by last 7 days.

Start Date - Snowflake Credits | Run Started At Utc Date - Models | Run Started At Utc Date - Jobs | Model Name | Owner | Analytics Group | Job Name | Run Started At Utc Date - Downtime

Last 14 Days | Last 30 Days | Last 14 Days | is model.bi.fact_rental | any value | is any value | is ["tag:nightly"] or ["tag:core"] or ["tag:2h..."] | Last 7 Days

Downtime

Downtime is calculated as the date difference in hours between a model last unsuccessful run and the next successful run of the same model. It's only calculated for the core models, to track SLOs.
Use 'Run Started At Utc Date - Downtime' filter to choose the time period.

0

0% of 524 Models Have Downtime >= 3h

0

Avg Downtime

0

Max Downtime

0

Min Downtime

Downtime duration by model - only Core models

Model Name

Downtime Duration

No Results

Core models that have a downtime > 3h, breaking our SLOs

TIER

dbt Performance Dashboard

Filter by your user to check how
your models are performing

Start Date - Snowflake Credits | Run Started At Utc Date - Models | Run Started At Utc Date - Jobs | Model Name | Owner | Analytics Group | Job Name | Run Started At Utc Date - Downtime |

Last 14 Days | Last 30 Days | Last 14 Days | is model.bi.fact_rental | any value | is any value | is ["tag:nightly"] or ["tag:core"] or ["tag:2h... | Last 7 Days

Performance Ranking

Execution times by model - Top 15 Worst-Performing Models (filter by owner or group)

Model Name	Materialization	Median Execution Times - minutes	Avg Execution Times - minutes	Min Execution Times - minutes	Max Execution Times - minutes
1 model.bi.segments_new_inactive_reactivated_retained_rollup	table	115.61	110.50	99.82	116.94
2 model.bi.dtm_dfx_time_snapshot	table	84.02	82.47	57.91	112.68
3 model.bi.fact_consumer_app_event	incremental	64.07	61.93	37.43	89.04
4 model.bi.fact_customer_acq_channel_and_revenue	table	40.59	41.34	36.53	90.21
5 model.bi.fact_vehicle_state_snapshot	table	40.24	35.83	15.19	69.02
6 model.bi.dtm_zone_herbin_mapping	table	30.59	30.48	14.70	46.92
7 model.bi.fact_customer_acq_cohorts	incremental	29.19	29.44	23.90	34.71
8 model.bi.raw_consumer_app_preprocess	incremental	23.73	24.48	13.52	35.48
9 model.bi.dtm_corr_segmentation_weekly	incremental	22.06	19.32	1.03	27.37
10 model.bi.oba_contact_and_churn	table	13.29	12.95	5.90	22.94
11 model.bi.dtm_revenue	table	11.38	12.47	0.00	41.43

Execution times by test - Top 15 Worst-Performing Tests

Model Name	Median Execution Times - minutes	Min Execution Times - minutes	Max Execution Times - minutes
1 test.bi.unique_segment_tracks_raw_event_id_d5eda4c89d	8.61	5.62	9.79
2 test.bi.unique_fact_consumer_app_event_event_id_#923673bd	8.44	2.85	12.87
3 test.bi.unique_raw_consumer_app_preprocess_event_id_b6c24f0d56	7.60	2.92	16.52
4 test.bi.dtm_unique_combination_of_columns_fact_consumer_app_event_session_id_event_order_9272263b69	6.34	4.44	12.59
5 test.bi.relationships_fact_consumer_app_session_session_id_ref_fact_consumer_app_event_3b7cae9903	3.42	2.16	5.66
6 test.bi.not_null_raw_rental_services_relocation_confirmed_rental_id_43b7633083	1.32	0.02	3.08
7 test.bi.unique_raw_rental_services_relocation_confirmed_rental_id_624c59696e	1.08	0.02	8.31
8 test.bi.not_null_raw_rental_services_pre_auth_rental_id_1928f01929	0.81	0.02	2.11
9 test.bi.assert_frontend_event_session_referential_integrity	0.67	0.44	1.31
10 test.bi.not_null_dtm_rental_geo_route_geo_linestring_7714724220	0.53	0.40	0.96
11 test.bi.unique_adjust_events_raw_event_id_cfe9b2859	0.52	0.25	1.12

Execution times by seed - Top 15 Worst-Performing Seeds

Model Name	Median Execution Times - seconds	Min Execution Times - seconds	Max Execution Times - seconds
1 seed.bi.erp_packages	12.1738	12.1738	12.1738
2 seed.bi.erp_currencies	7.1341	7.1341	7.1341
3 seed.bi.markets_base	3.8105	3.2839	4.1429
4 seed.bi.dtm_test_customer_segmentation_crm	2.8486	2.4397	3.2614

Ranking performing:

Top 15 worst-performing
models, tests and seeds.

It can be filtered by owner and
group, so we can identify our
own models.



SLOs



TIER

SLOs: Overview

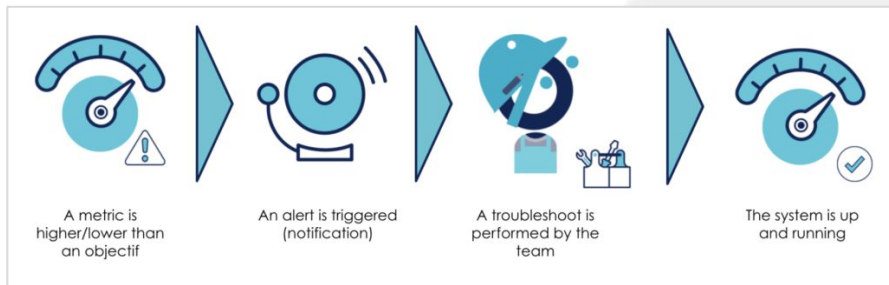


A **metric** is basically a core value/service being measured.

An **SLO** (Service Level Objective) is essentially a level that we want to hit with a particular metric. It is usually expressed as a percentage over a period of time

Customers place food orders in a restaurant.

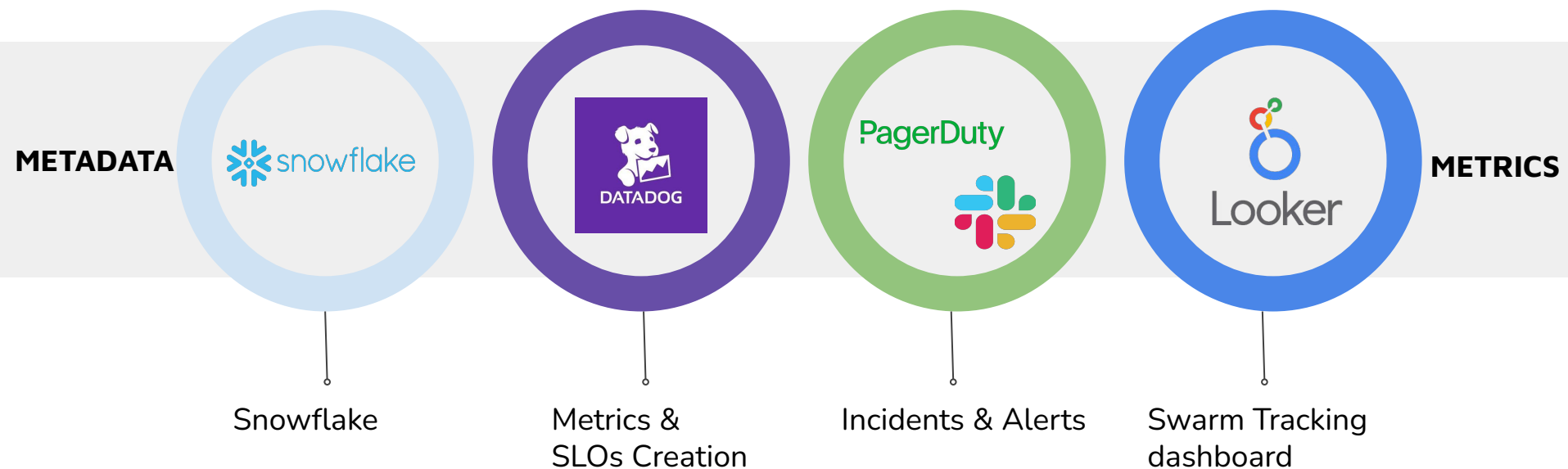
- Did the customer receive the order they requested? This is seen as the metric
- The percentage of orders serviced in 10 minutes (Ideally, you would like all orders for food to be serviced/responded to within 10 minutes) - SLO



Source: <https://blog.octo.com/slo-the-unexpected-power-of-metrics/>

Data Pipeline For SLO Reporting

Data Pipeline For SLO Reporting



Data Pipeline For SLO Reporting

Snowflake

Metrics and SLOs

Incidents and alerts

Swarm tracking dashboard

Persisted Model on Snowflake:

The metrics that are being used to track SLOs are then configured in yml files that feed data from Snowflake to Datadog.

```
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
- account:
  username:
  password: $(data.vault_generic_secret.datadog_snowflake_secrets.data["snowflake_dd_password"])
  role:
  schema:
  database:
  min_collection_interval: 3600
  custom_queries:
    - query: SELECT MAX(METADATA__TIMESTAMP) dbt_failure_rate_core_models_METADATA__TIMESTAMP, SUM(CASE WHEN UPPER(RESULTS_STATUS) = 'SUCCESS' THEN
      columns:
        - name: dbt_failure_rate_core_models_METADATA__TIMESTAMP
          type: tag
        - name: dbt_failure_rate_core_models_SUCCESS
          type: gauge
        - name: dbt_failure_rate_core_models_ERROR
          type: gauge
        - name: dbt_failure_rate_core_models_SKIPPED
          type: gauge
        - name: dbt_failure_rate_core_models_TOTAL_CORE_MODELS
          type: gauge
        - name: dbt_failure_rate_core_models_PERC_ERROR
          type: gauge
        - name: dbt_failure_rate_core_models_PERC_SKIPPED
          type: gauge
        - name: dbt_failure_rate_core_models_PERC_SUCCESS
          type: gauge
        - name: dbt_failure_rate_core_models_SLO
          type: gauge
      tags:
        - prod:dbt_failure_rate_core_models
- account:
  username:
  password: $(data.vault_generic_secret.datadog_snowflake_secrets.data["snowflake_dd_password"])
  role:
  schema:
  database:
  min_collection_interval: 3600
  custom_queries:
    - query: SELECT MAX(METADATA__TIMESTAMP) dbt_delay_core_models_METADATA__TIMESTAMP, SUM(CASE WHEN DAILY_DELAY_TIME > 180 THEN 1 ELSE 0 END) db
      columns:
        - name: dbt_delay_core_models_METADATA__TIMESTAMP
          type: tag
        - name: dbt_delay_core_models_NUMBER_OF_DELAYS
          type: gauge
        - name: dbt_delay_core_models_TOTAL_CORE_MODELS
          type: gauge
        - name: dbt_delay_core_models_PERC
          type: gauge
        - name: dbt_delay_core_models_SLO
          type: gauge
      tags:
        - prod:dbt_delay_core_models
```

Data Pipeline For SLO Reporting

Snowflake

Metrics and SLOs

Incidents and alerts

Swarm tracking dashboard

Definition, creation and notification of monitors and SLO thresholds:

Once in DataDog click Monitors / New Monitor

Views Monitors + Save + New Monitor

Filter monitors

Hide Controls Showing 1-50 of 1,120 results Mute Resolve Delete Edit Tags

STATUS	MUTED LEFT	NAME 1	TAGS
OK		A lot of async lock actions are fail...	team:ravens fl... +1
OK		AWS Health Event (eu-central-1) ...	
OK		AWS Health Event (global) - {{eve...	
OK		AWS IoT Rule Lambda Errors	iot ring-0

Then select Metric

Custom Recommended

Host

Metric →

Anomaly

APM

CI ALPHA

Composite

Error Tracking

In the area we can search for the metric as it was defined on the YAML file

Source Edit

a dbt

+

snowflake.dbt_delay_core_models_SLO

snowflake.dbt_failure_rate_core_models_SLO

snowflake.dbt_failure_rate_core_models_SUCCESS

snowflake.dbt_delay_core_models_NUMBER_OF_DELAYS

snowflake.dbt_delay_core_models_PERC

snowflake.dbt_delay_core_models_TOTAL_CORE_MODELS

snowflake.dbt_failure_rate_core_models_ERROR

snowflake.dbt_failure_rate_core_models_PERC_ERROR

snowflake.dbt_failure_rate_core_models_PERC_SKIPPED

snowflake.dbt_delay_core_models_SLO

type: gauge

aws (14) autoscaling_group (11) iam_profile (11)

name (11) instance-type (7)

dbt_delay_core_models_metadata_timestamp (5)

availability-zone (3) image (2)

k8s.io/cluster-autoscaler/node-template/taint/dedicated...

Click a tag to group by

Edit in Metrics Summary

Data Pipeline For SLO Reporting

Snowflake

Metrics and SLOs

Incidents and alerts

Swarm tracking dashboard

Next we can define the alert conditions.

- Thresholds for alerts and warnings
- Recovery thresholds
- Missing data alerts

3 Set alert conditions

Trigger when the evaluated value is **below** the threshold

Alert threshold: < 50

Warning threshold: < Optional

Notify If data is missing for more than 60 minutes.

We recommend the missing data window be at least 2x the evaluation period above

Advanced options

Recovery thresholds

Alert recovery threshold: >= 50.01

Warning recovery threshold: >= Optional

Missing data options

After 1 hour automatically resolve this event from a no data state.

Evaluation options

Delay monitor evaluation by 0 seconds

Do not require a full window of data for evaluation.

Finally we can define notification options. Who should be notified once there is an alert or warning and also what is the priority (P1 to P5) and what service on PagerDuty should be triggered (for example @pagerduty-analytics-tier-dbt)

4 Notify your team

Edit Preview

DBT Core Models are Delayed for more than 180 minutes

This is the alert for DBT Core Models are Delayed for more than 180 minutes

@pagerduty-analytics-tier-dbt @vitor.cavalanti@tier.app @kumar.aman@tier.app @slack-tier-bi-data-alerts

@pagerduty-analytics-tier-dbt X
@vitor.cavalanti@tier.app X @kumar.aman@tier.app X
@slack-tier-bi-data-alerts X

Renotification

If this monitor stays in Select status renotify every Select time frame

Tags: Select value

Priority: P4 (Low)

TIER

Data Pipeline For SLO Reporting

Snowflake

Metrics and SLOs

Incidents and alerts

Swarm tracking dashboard

Metrics and SLOs failures/warnings created as incidents on Pagerduty and notifications sent to slack

PagerDuty Incidents Services People Automation Analytics Integrations Status

Incidents on analytics-platform

Your open incidents: 0 triggered, 0 acknowledged
All open incidents: 0 triggered, 0 acknowledged

Open Triggered Acknowledged **Resolved** Any Status

<input type="checkbox"/>	Status	Priority	Urgency	Title	Created	Service	Assigned To
<input type="checkbox"/>	Resolved	--	High	Looker Availability Monitor SHOW DETAILS (1 resolved alert)	#17244 on Oct 31, 2022 at 11:59 PM	analytics-tier-dbt	--
<input type="checkbox"/>	Resolved	--	High	Looker Availability Monitor SHOW DETAILS (1 resolved alert)	#16911 on Oct 20, 2022 at 2:08 AM	analytics-tier-dbt	--
<input type="checkbox"/>	Resolved	--	High	Looker Availability Monitor SHOW DETAILS (1 resolved alert)	#16844 on Oct 18, 2022 at 12:20 AM	analytics-tier-dbt	--
<input type="checkbox"/>	Resolved	--	High	No Data: DBT Core Models are Delayed for more than 180 minutes SHOW DETAILS (1 resolved alert)	#16231 on Oct 4, 2022 at 4:37 PM	analytics-tier-dbt	--

Datadog APP 3:24 AM

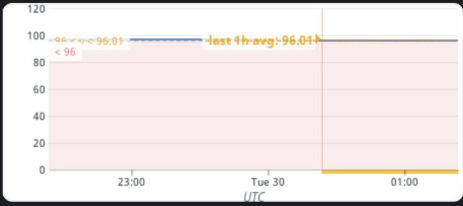
Warn: DBT Core Models are Delayed for more than 180 minutes
This is the alert for DBT Core Models are Delayed for more than 180 minutes

@pagerduty-analytics-tier-dbt @vitor.cavalanti@tier.app @kumar.aman@tier.app @slack-tier-bi-data-alerts

snowflake.dbt_delay_core_models_SLO over * was < 98.0 on average during the last 1h.

Metric value: 96.012 (9 kB) ▾

Notified
@kumar.aman@tier.app,
@vitor.cavalanti@tier.app, ... and 2 more



TIER

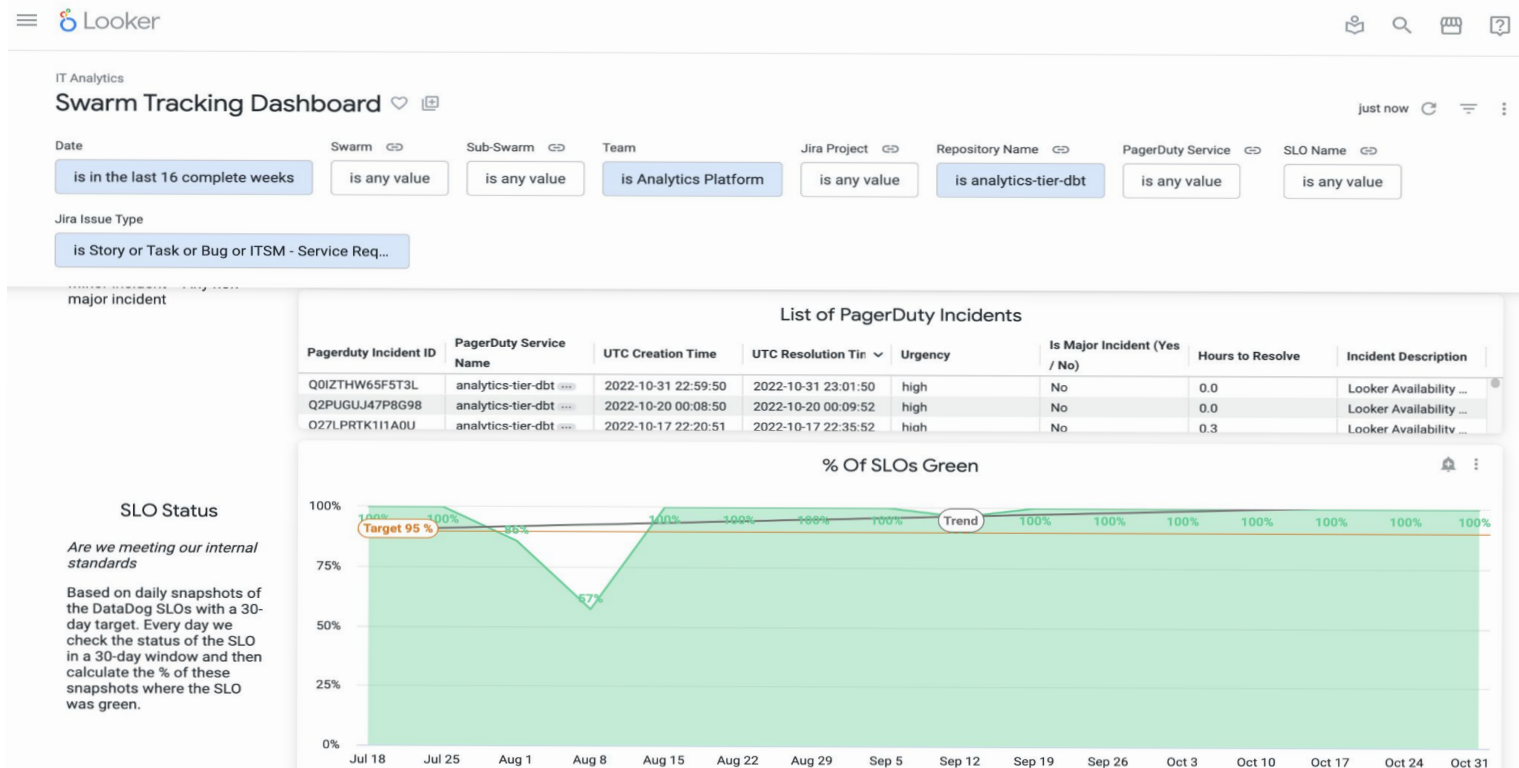
Data Pipeline For SLO Reporting

Snowflake

Metrics and SLOs

Incidents and alerts

Swarm Tracking Dashboard



Data Pipeline For SLO Reporting - Summary

Snowflake

Metrics and SLOs

Incidents and alerts

Swarm Tracking Dashboard

The metrics are configured in **yaml files** that feed data from Snowflake to Datadog



Definition, creation and notification of **SLO thresholds** on DataDog



Metrics and SLOs failures/warnings created as **incidents** on Pagerduty



Notifications are send to Slack



The **Swarm Tracking Dashboard** combines all SLOs for teams in the data organization



Thank you for your attention!
Time for questions 🧠

Thank you!

Stay tuned for upcoming events and other content

Berlin dbt Meetup Group
#local-berlin Slack Channel



Before you leave...

Please share your feedback with us!

