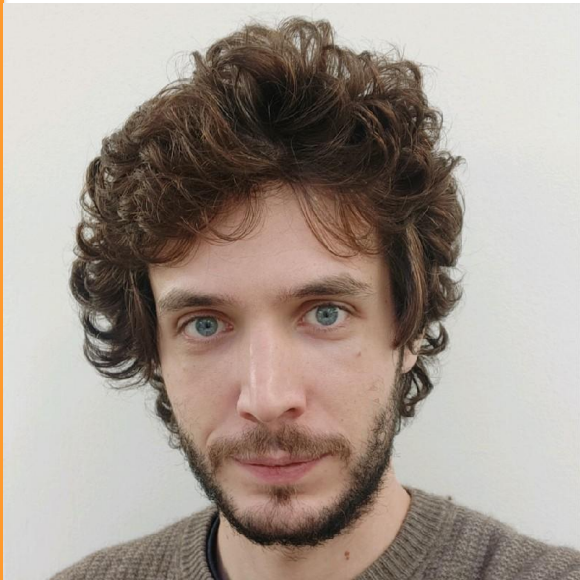




Ensorceler vos processus ETL avec Mage.ai
Marc Chevallier 29/06/2023



Travaillant chez Synaltic depuis 2018, j'ai effectué ma thèse CIFRE entre 2019 et 2022. Celle-ci portait sur le profilage de données à l'aide de l'apprentissage automatique. Actuellement, j'occupe le poste de Responsable R&D chez Synaltic. Dans le but d'accélérer la création de pipelines de données, je me suis intéressé à Mage.ai.



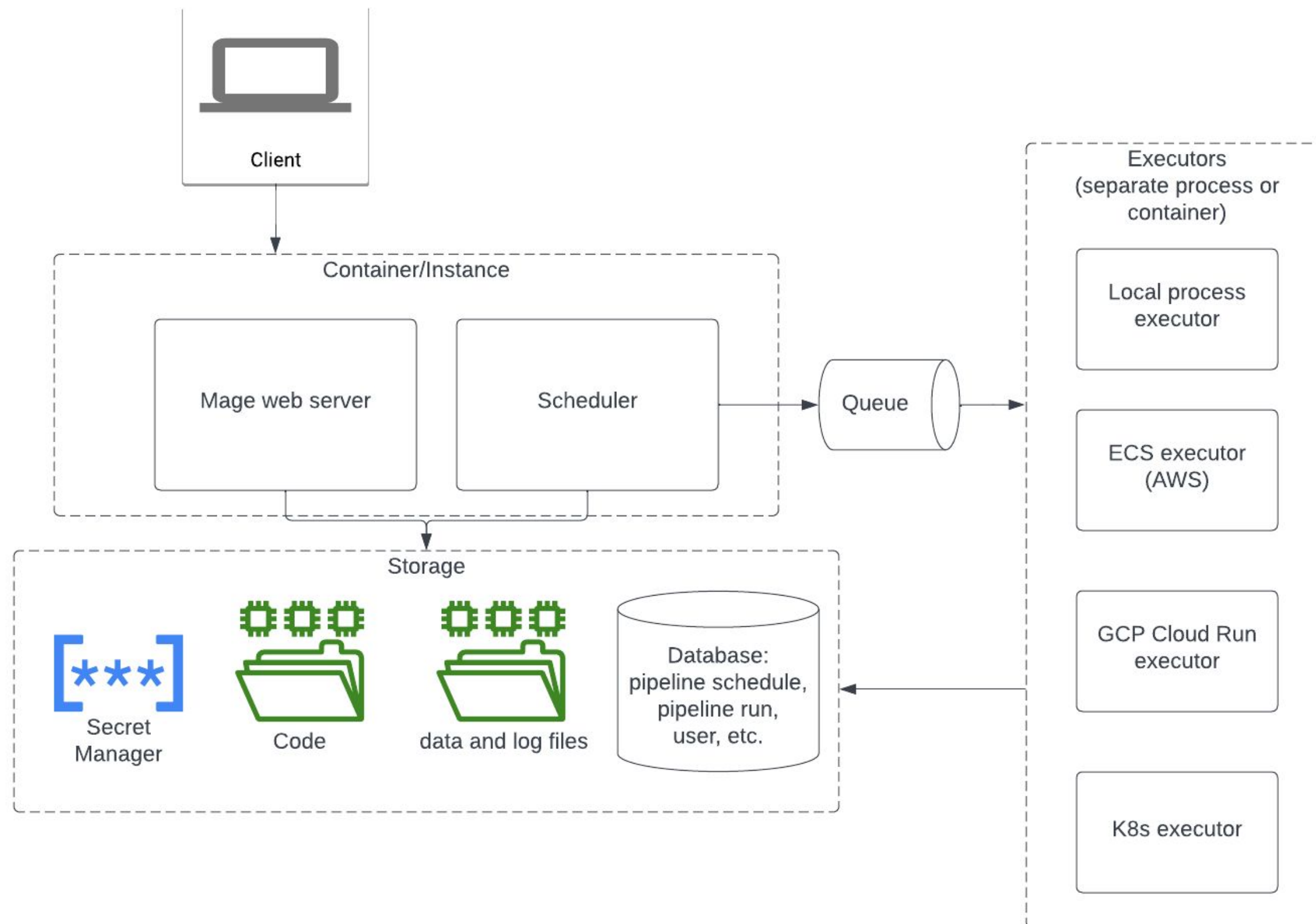
L'histoire de la solution

Mage.ai est une solution qui existe depuis 2021. Initialement, le projet avait pour but de proposer une plateforme de data science, mais la solution s'est avérée ne pas trouver de public. Ils ont ensuite pivoté leur modèle pour proposer une plateforme ETL open source (1er version été 2022) basée sur trois grands principes.

- L'interface utilisateur simplifiée permet de construire visuellement, rapidement et intuitivement des pipelines de données.
- Il propose 3 cas d'utilisation : les pipelines de traitement par lots, les pipelines d'intégration de données et les pipelines de traitement en continu.
- Les bonnes pratiques d'ingénierie sont intégrées. La conception est intrinsèquement modulaire et testable.



Fonctionnement de la solution



Git repository settings

You can enable the Git integration by supplying the url for your remote repository. You will need to [set up your SSH key](#) if you have not done so already.

Defaults to Python's `os.getcwd()` if omitted. Mage will create this local directory if it doesn't already exist.

(OPTIONAL) These fields are recommended if your Git and SSH settings can be reset unexpectedly. Filling out these fields will allow Mage to continue to connect to the remote Git repository.

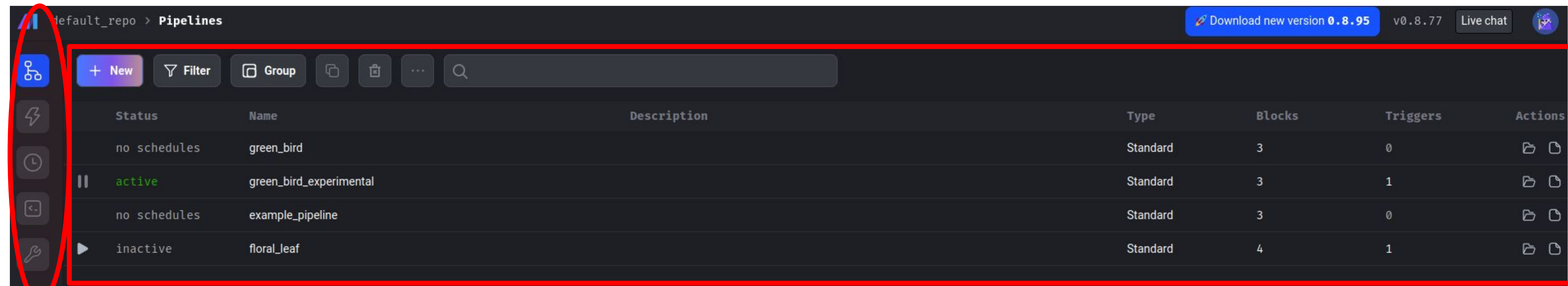
Run `"cat ~/.ssh/id_rsa.pub | base64 | tr -d '\n' && echo"` in terminal to get base64 encoded public key and paste the result here. The key will be stored as a Mage secret.

Follow same steps as the public key, but run `"cat ~/.ssh/id_rsa | base64 | tr -d '\n' && echo"` instead. The key will be stored as a Mage secret.

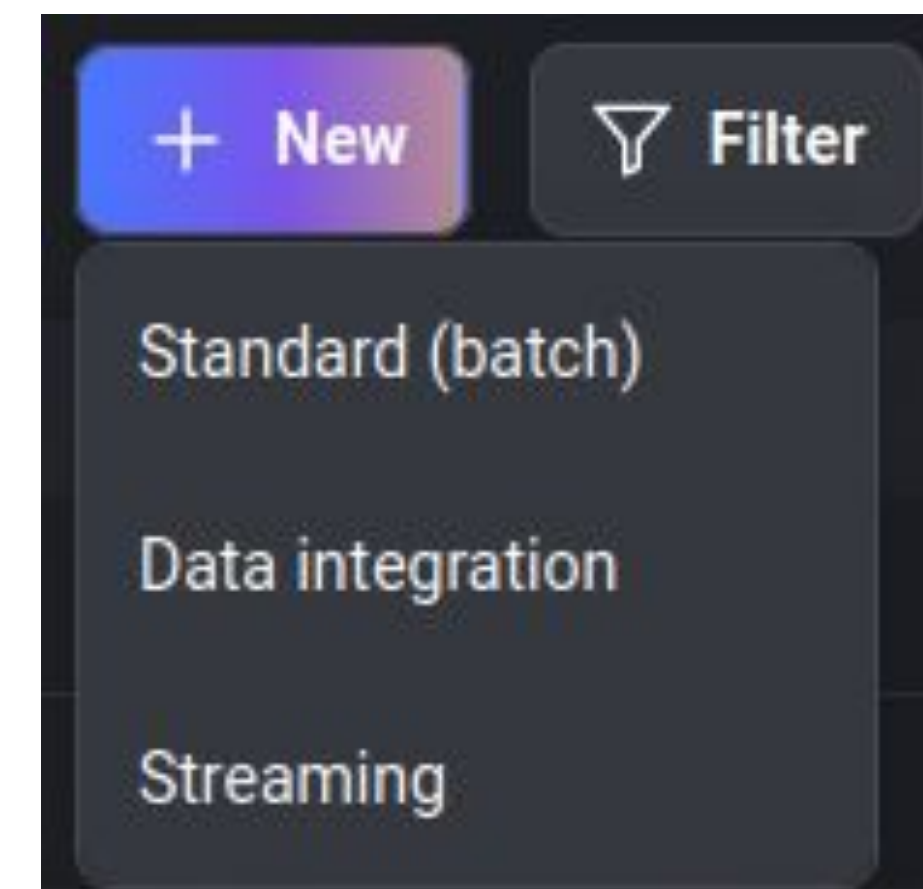
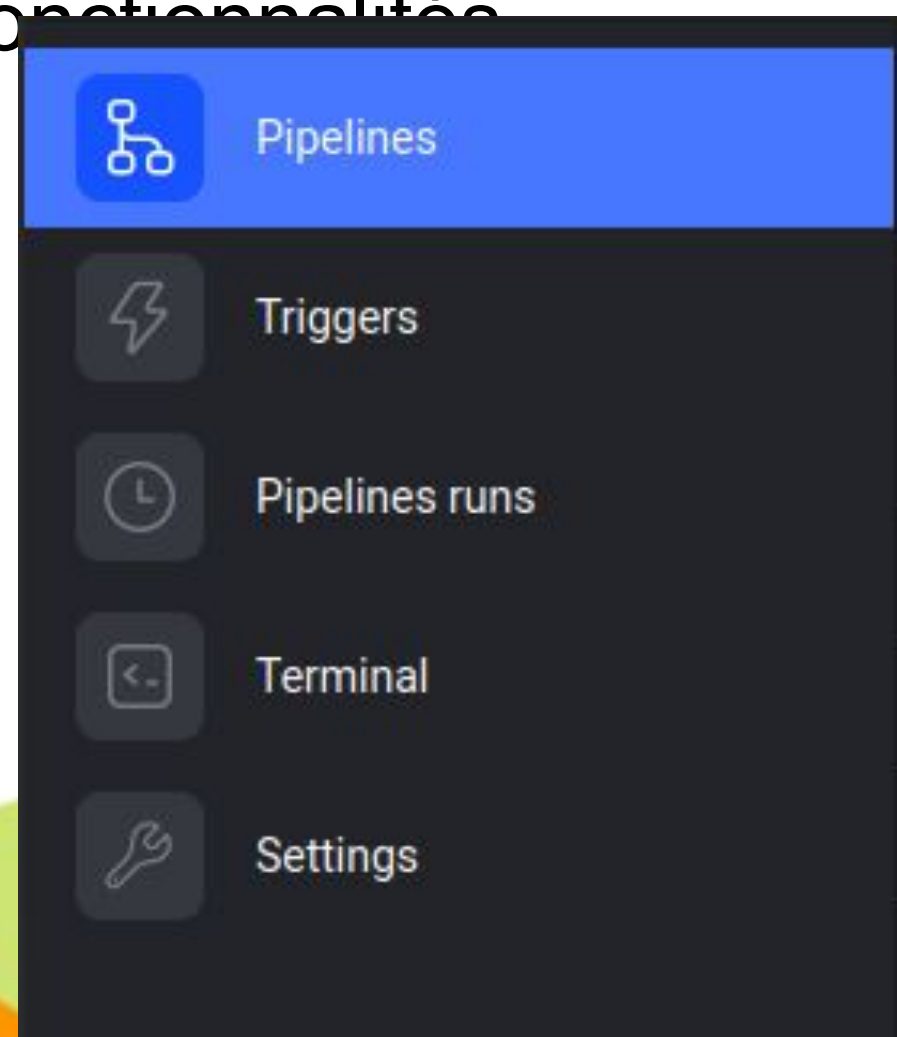
Use Git Sync

[Save repository settings](#)

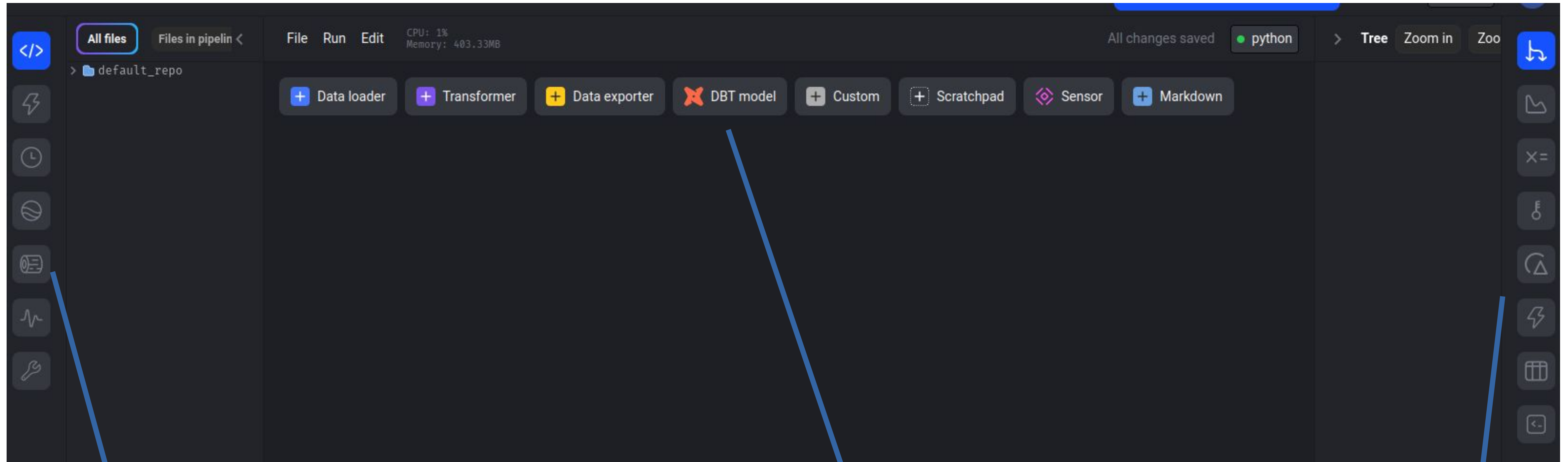
Une interface esthétique et simplifiée



Un menu latéral qui offre un accès direct aux principales fonctionnalités



Création d'un pipeline



Panneau latéral de suivi du pipeline.

Liste des blocs qu'on peut créer

Variable, Secret, Visualisation et Extensions

Une partie du code est pré-générée



```
PY DATA LOADER wild_leaf Edit parent blocks ←×
1 |from mage_ai.data_preparation.repo_manager import get_repo_path
2 |from mage_ai.io.config import ConfigFileLoader
3 |from mage_ai.io.postgres import Postgres
4 |from os import path
5 |if 'data_loader' not in globals():
6 |    |from mage_ai.data_preparation.decorators import data_loader
7 |if 'test' not in globals():
8 |    |from mage_ai.data_preparation.decorators import test
9
10
11 @data_loader
12 def load_data_from_postgres(*args, **kwargs):
13     """
14     Template for loading data from a PostgreSQL database.
15     Specify your configuration settings in 'io_config.yaml'.
16
17     Docs: https://docs.mage.ai/design/data-loading#postgres
18     """
19     query = 'your PostgreSQL query' # Specify your SQL query here
20     config_path = path.join(get_repo_path(), 'io_config.yaml')
21     config_profile = 'default'
22
23     with Postgres.with_config(ConfigFileLoader(config_path, config_profile)) as loader:
24         |return loader.load(query)
25
26
27 @test
28 def test_output(output, *args) → None:
29     """
30     Template code for testing the output of the block.
31     """
32     assert output is not None, 'The output is undefined'
33
```

Les librairies nécessaires sont déjà importées. (De nouvelles peuvent être installées puis importées.)

La fonction qui compose le corps du bloc est déjà pré-générée, il ne reste plus qu'à la compléter.

Un bloc de test est proposé par défaut.

Puis les blocs s'enchaînent



The screenshot shows the Synaltic interface. On the left, three code blocks are listed vertically, each with a play button and a dropdown menu. The first block is a 'DATA LOADER' named 'load_asso' with '(46 lines collapsed)'. The second is a 'TRANSFORMER' named 'json_jo_pandas_jo' with '(174 lines collapsed)'. The third is a 'DATA EXPORTER' named 'to_s3_custom' with '(34 lines collapsed)'. On the right, a flow graph visualizes the execution order: a blue box 'load_asso' at the top, connected by a vertical line to a purple box 'json_jo_pandas_jo', which is connected to a yellow box 'to_s3_custom' at the bottom. Each box has a green checkmark in the top-left corner.

Les blocs s'enchaînent les uns derrière les autres en se passant l'information sous forme de DataFrame ou de dictionnaire.

Un graphe permet de visualiser l'enchaînement des blocs et de choisir les blocs parents.

Test de chaque bloc

```
PY DATA LOADER load_asso Edit parent blocks ←X

@test
def test_output(output, *args) → None:
    """
    Template code for testing the output of the block.
    """
    assert output is not None, 'The output is undefined'

1/1 tests passed.

Cible de l'extraction : ASS20230025.taz
Sampled output is provided here for preview.
{'parution': {'@xmlns:exsl': 'http://exslt.org/common',
'dateParution': '2023-06-20',
'numParution': '20230025',
'listeAnnonces': {'annonce': [{'metadonnees': {'numAnnonce': '1',
'identifiant': '202300250001',
'dept': '01',
'type': {'@code': '1'},
'idAssoc': 'W011006179',
'page': '1',
'themes': {'theme': {'@code': '007030',
```

Chaque bloc peut être testé indépendamment. On peut obtenir une visualisation directe du résultat retourné par le bloc.

Vérification visuelle rapide

PY TRANSFORMER json_jo_pandas_jo

(174 lines collapsed)

1/1 tests passed.

	rna	action	
0	W011006179	creation	2
1	W012015815	creation	2
2	W012015816	creation	2

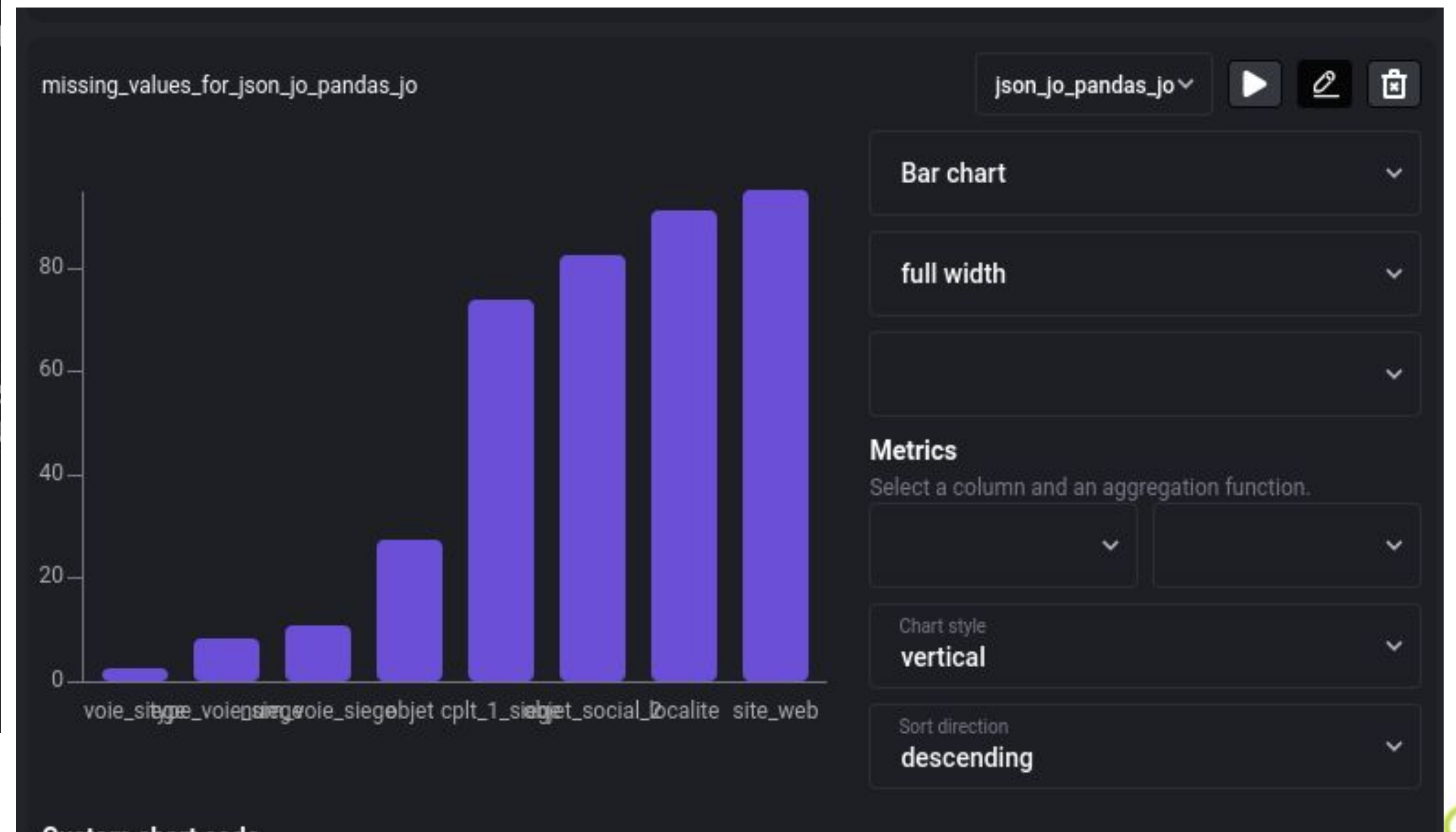
Custom charts Add chart

- Bar chart
- Histogram
- Line chart
- Pie chart
- Table
- Time series bar chart
- Time series line chart

Templates

- % of missing values
- Unique values
- Most frequent values
- Summary overview
- Feature profiles

Un grand nombre de graphiques permettant d'évaluer directement la qualité des données sont utilisables.



Contrôle de la qualité



	mode value	frequency	% of values
rna	W011006179	0.04%	1
action	creation	64.02%	1500
dateDeclaration	2023-06-12	13.53%	317
nom	LA 4L DE MICHEL	0.04%	1

Il est possible de visualiser directement certaines valeurs clés, mais aussi d'intégrer des vérifications en utilisant Great Expectations.

```
PY TRANSFORMER json_jo_pandas_jo
(174 lines collapsed)

1/1 tests passed.

INFO:great_expectations.data_context.types.base:Created temporary directory '/tmp/tmp7rcacyee' for ephemeral docs site
WARNING:py.warnings:/usr/local/lib/python3.10/site-packages/great_expectations/expectations/expectation.py:1477: UserWarning: `result_format` configured at the Validator-level will not be persisted. Please add the configuration to your Checkpoint config or checkpoint_run() method instead.
  warnings.warn(
Calculating Metrics:  0%|          | 0/1 [00:00<?, ?it/s]
Calculating Metrics:  0%|          | 0/1 [00:00<?, ?it/s]

Exception                                 Traceback (most recent call last)
/tmp/ipykernel_1844/3399365736.py in <cell line: 236>()
    234     return find_lambda_val: val is not None, output)
    235
-> 236 df = execute_custom_code()
```

```
PY EXTENSION sparkling_violet

xtension' not in globals():
from mage_ai.data_preparation.decorators import extension

nsion('great_expectations')
alidate(validator, *args, **kwargs):
alidator.expect_table_row_count_to_be_between(min_value=200,max_value=500)

Select blocks to run expectations on
Click a block name to run expectations on it.
json_jo_pandas_jo

INFO:great_expectations.data_context.types.base:Created temporary directory '/tmp/tmpt8a_9zeh' for ephemeral docs site
WARNING:py.warnings:/usr/local/lib/python3.10/site-packages/great_expectations/expectations/expectation.py:1477: UserWarning: `result_format` configured at the Validator-level will not be persisted. Please add the configuration to your Checkpoint config or checkpoint_run() method instead.
  warnings.warn(
```


Orchestrer un pipeline



Trigger type

How would you like this pipeline to be triggered?

Schedule

- This pipeline will run continuously on an interval or just once.

Event

- This pipeline will run when a specific event occurs.

API

- Run this pipeline when you make an API call.

L'orchestration devient très simple, en deux clics on a un pipeline qui s'exécute à la fréquence que l'on désire.

Orchestrer un pipeline

Il est possible de choisir une fréquence personnalisée ou d'utiliser un outil externe qui envoie les informations nécessaires au pipeline pour son démarrage.

Trigger type

How would you like this pipeline to be triggered?

Schedule
This pipeline will run continuously on an interval or just once.

Settings

Trigger name	long snow
Frequency	custom
Cron expression	
Start date and time	2023-06-26 09:14

Endpoint

Make a **POST** request to the following endpoint:

`https://ma[REDACTED]/api/pipeline_schedules/3/p`

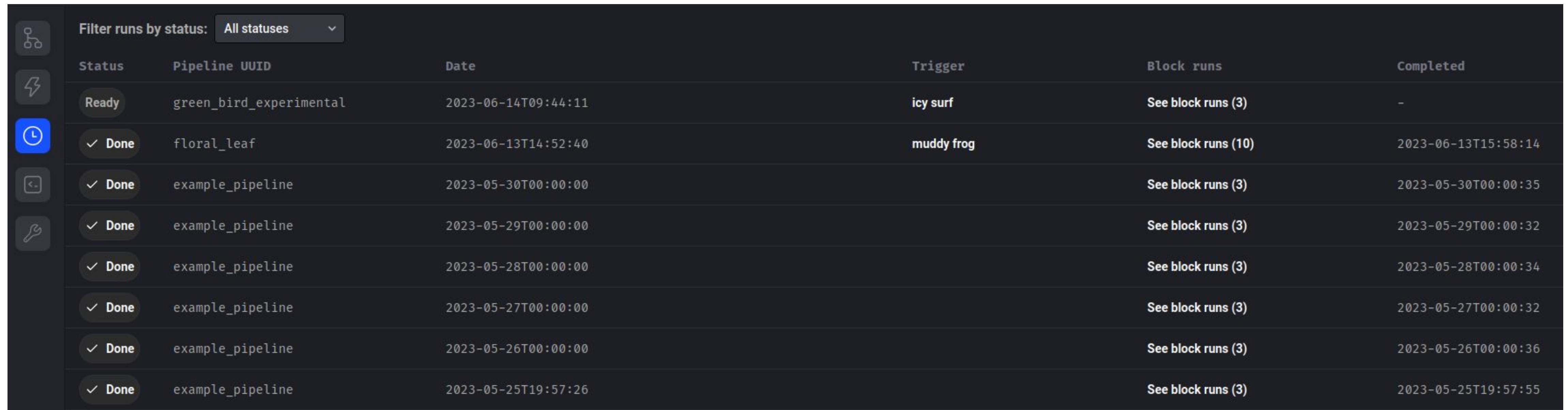
Payload

You can optionally include runtime variables in your request payload. These runtime

```
{
  "pipeline_run": {
    "variables": {
      "key1": "value1",
      "key2": "value2"
    }
  }
}
```


Suivi des pipelines

L'onglet "Pipelines Run" permet de suivre l'exécution de chaque pipeline, tandis que l'onglet "RUN" permet de suivre les exécutions d'un pipeline spécifique.

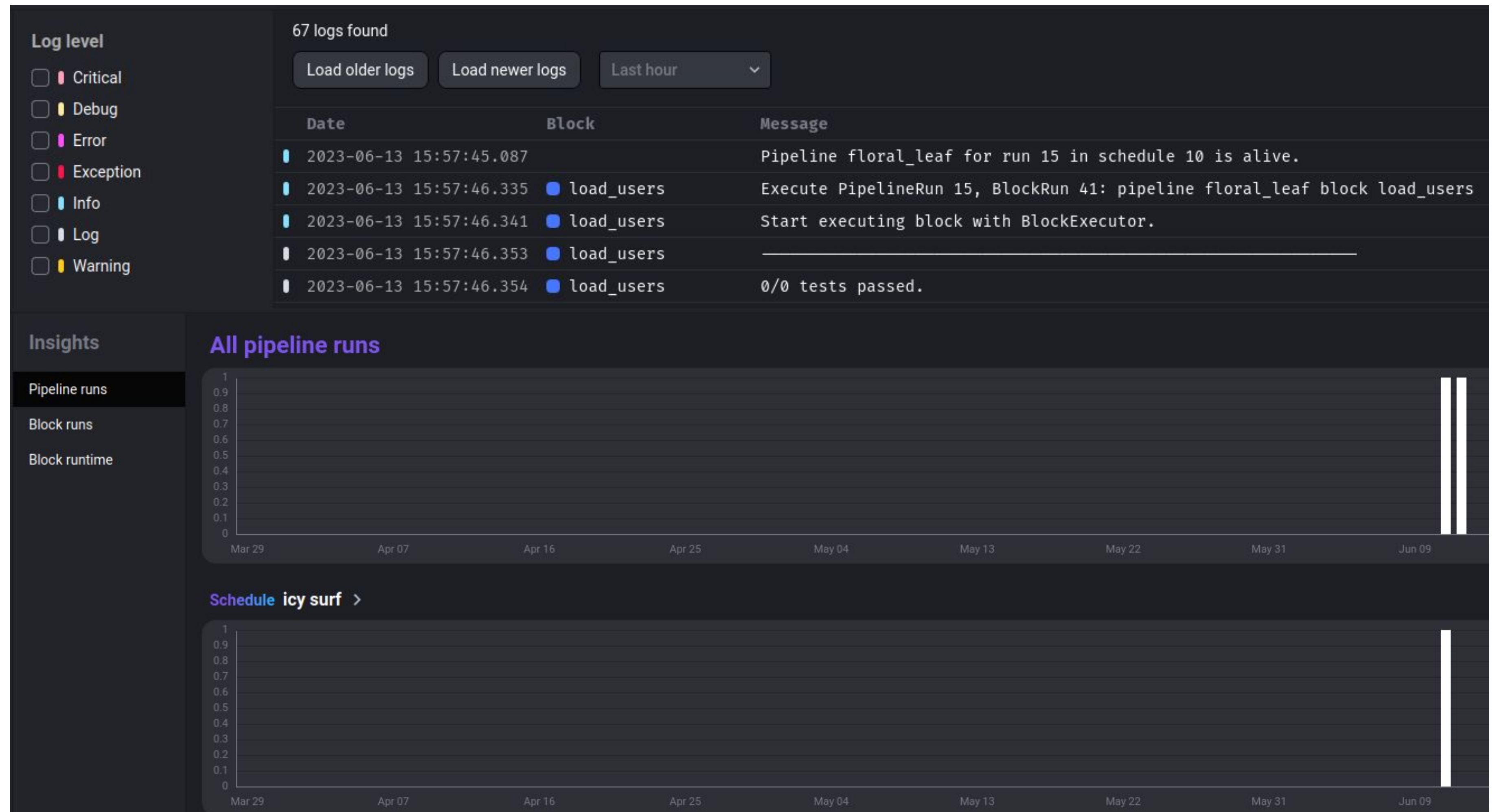


Status	Pipeline UUID	Date	Trigger	Block runs	Completed
Ready	green_bird_experimental	2023-06-14T09:44:11	icy surf	See block runs (3)	-
✓ Done	floral_leaf	2023-06-13T14:52:40	muddy frog	See block runs (10)	2023-06-13T15:58:14
✓ Done	example_pipeline	2023-05-30T00:00:00		See block runs (3)	2023-05-30T00:00:35
✓ Done	example_pipeline	2023-05-29T00:00:00		See block runs (3)	2023-05-29T00:00:32
✓ Done	example_pipeline	2023-05-28T00:00:00		See block runs (3)	2023-05-28T00:00:34
✓ Done	example_pipeline	2023-05-27T00:00:00		See block runs (3)	2023-05-27T00:00:32
✓ Done	example_pipeline	2023-05-26T00:00:00		See block runs (3)	2023-05-26T00:00:36
✓ Done	example_pipeline	2023-05-25T19:57:26		See block runs (3)	2023-05-25T19:57:55

Suivi des pipelines

Pour chaque pipeline, l'onglet "Logs" permet de suivre les journaux (logs).

L'onglet "Monitor" permet quant à lui de suivre quand les pipelines sont déclenchés et par quel déclencheur (trigger).



The screenshot displays the Synaltic monitoring interface. On the left, there is a sidebar with navigation options: "Log level", "Insights", "Pipeline runs", "Block runs", and "Block runtime". The "Log level" section includes checkboxes for Critical, Debug, Error, Exception, Info, Log, and Warning. The "Insights" section is currently expanded to show "Pipeline runs".

The main content area shows "67 logs found" with buttons for "Load older logs", "Load newer logs", and a "Last hour" filter. Below this is a table of logs:

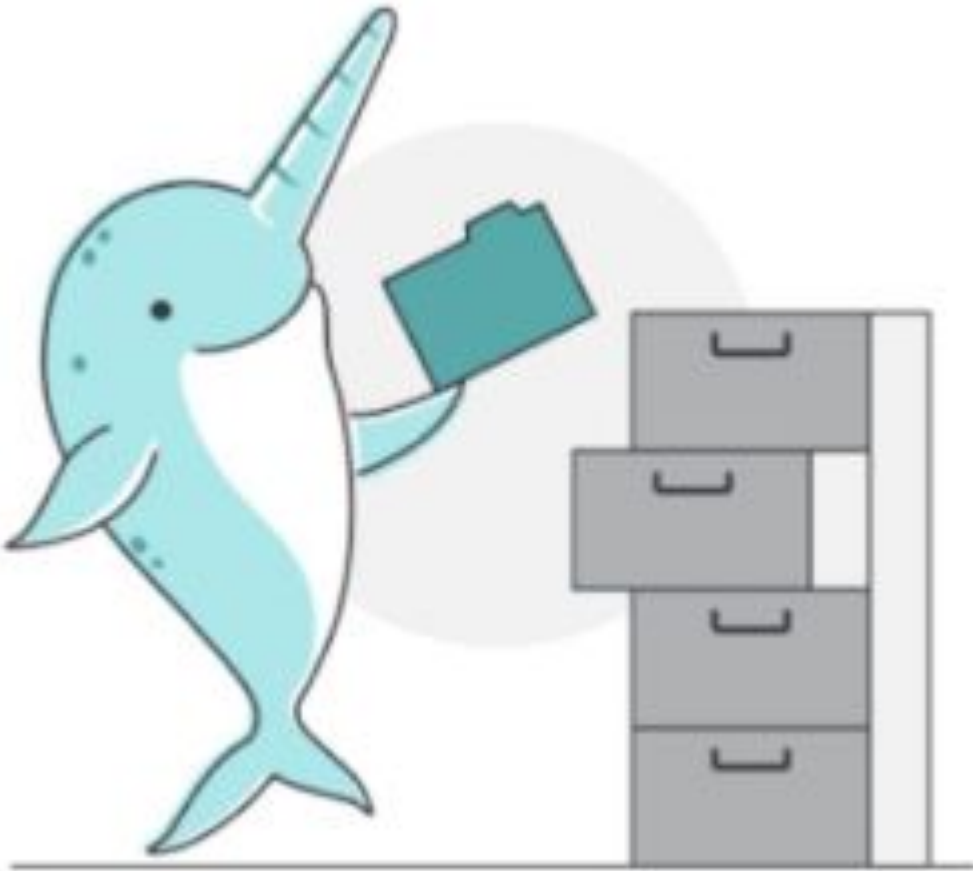
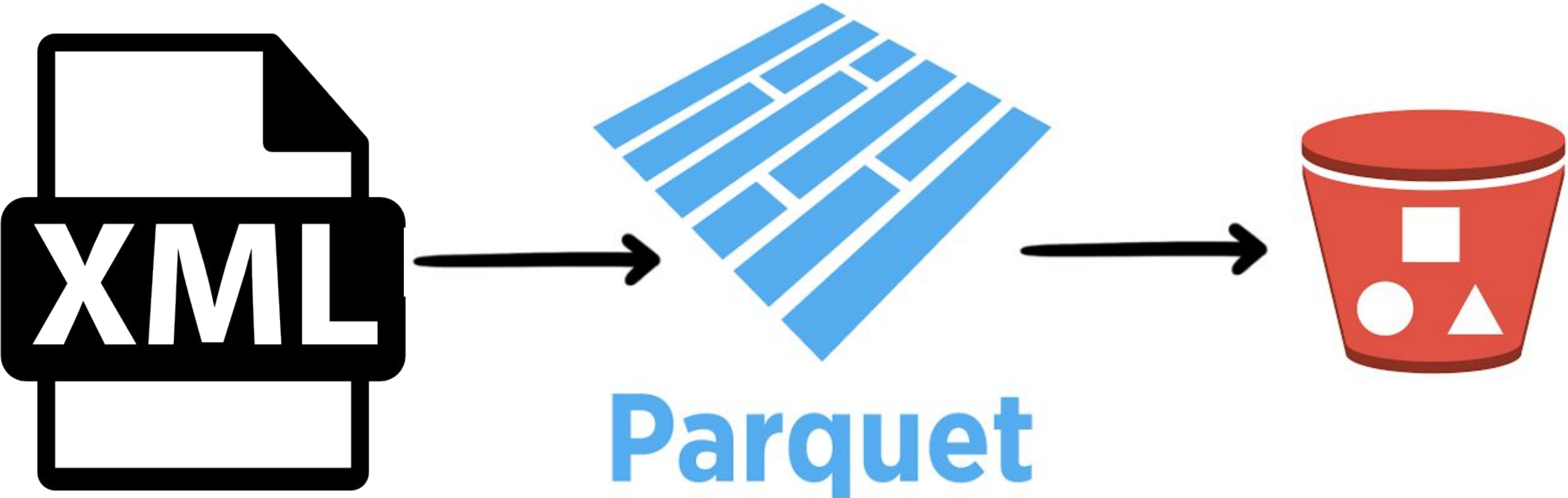
Date	Block	Message
2023-06-13 15:57:45.087		Pipeline floral_leaf for run 15 in schedule 10 is alive.
2023-06-13 15:57:46.335	load_users	Execute PipelineRun 15, BlockRun 41: pipeline floral_leaf block load_users
2023-06-13 15:57:46.341	load_users	Start executing block with BlockExecutor.
2023-06-13 15:57:46.353	load_users	_____
2023-06-13 15:57:46.354	load_users	0/0 tests passed.

Below the logs, there are two line charts. The top chart is titled "All pipeline runs" and the bottom chart is titled "Schedule icy surf >". Both charts have a y-axis from 0 to 1 and an x-axis showing dates from Mar 29 to Jun 09. The charts show a series of data points connected by lines, with some points highlighted by circles.

Exemple d'utilisation

.tar.gz

Journal-officiel.gouv.fr
Associations, fondations et fonds de dotation
Organisations syndicales et professionnelles
Bulletin des annonces légales obligatoires



Exemple d'utilisation

