From 0 to MLOps with ❄ Snowflake Data Cloud in 3 steps with the Kedro-Snowflake plugin

Tags	MLOps, Kedro, Snowflake, **Kedro-Snowflake plugin**, open source, **Snowpark**, Machine Learning
Meta Description	MLOps with the Snowflake Data Cloud: How to build ML pipelines in Kedro and execute them using Snowflake
Intro KB	Announcing the Kedro-Snowflake plugin which allows you to build ML pipelines in Kedro and execute them in a scalable Snowflake environment in three simple steps.
Autor	Marek Wiewiórka Michał BryśMarcin Zabłocki
LINK do grafik	

MLOps on Snowflake Data Cloud

MLOps is an ever-evolving field, and with the selection of managed and cloud-native machine learning services expanding by the day, **it can be challenging to navigate the options available**. With a plethora of managed and cloud-native machine learning services available, it's crucial to choose the right platform for running machine learning pipelines and deploying trained models. **However, three significant pain points persist in the MLOps landscape**:

- Lack of easy access to company's valuable data,
- the need for quick local iteration on ML pipelines
- Lack of a seamless transition to the cloud environment.

With Snowflake being a powerful data warehouse and Snowpark's ease of use, together they make a strong candidate for building complex ML pipelines. If you are not familiar with Snowpark yet, there are a lot of great articles introducing its core concepts and how you can use it for writing data science and machine learning (ML) code, e.g. here, here or here.

There are however at least a few shortcomings of the currently proposed approaches that have not yet been addressed:

- **ML pipelines orchestration** - in the current state, two strategies can be pursued:
    - using an external orchestrator service or tool, such as AzureML Pipelines or Apache Airflow for invoking Snowpark code directly
    - manually wrapping Snowpark code into Python UDFs and using them for building a directed acyclic graph (DAG) of steps of the Snowflake native tasks mechanism

  Unfortunately, neither of these methods seem to be free from flaws - the former requires additional scheduling components to be included in the architecture that makes it more complex and less platform-independent. The latter one is less user-friendly as it requires not only developing training code, but also defining Snowflake DAGs of tasks by means of plain SQL or Terraform programming language.
- **ML model lifecycle management** - there isn't any automation in place that makes it easy to promote/deploy training pipelines between stage/runtime environments - i.e. Development - Test - Production. This requires preparation of Continuous Integration/Continuous Training (CI/CT) processes on your own
- **Code standardization and project templates** - in its current state, Snowpark does not come with any built-in mechanism for code structuring, unit testing or automated documentation generation.

The above list of challenges **clearly indicates missing the integration of the Snowflake environment with an MLOps framework**, such as Kedro.

Today we are proudly announcing a solution that will fill this gap - the kedro-snowflake plugin. In the next post we will also guide you through the whole MLOps platform and ML model deployment on Snowflake. However, let's first take a look at what Kedro is and then let's build an ML pipeline in Kedro and execute it in the Snowflake environment in 3 simple steps.

Kedro - the MLOps Framework

Kedro is a widely-adopted, open-source Python framework that has claimed to bring engineering back to the data science world. The rationale behind using Kedro as a framework for creating maintainable and modular training code is in many aspects, similar to preferring Terraform technology over cloud-vendor native SDK for infrastructure provisioning and can be summarized in the following points:

- standardization of ML project layout,
- portability of ML pipelines,
- reusability code base, modules or even whole pipelines,
- a faster development loop thanks to the possibility of running/testing pipelines locally,
- clear and maintainable codebase with no dependencies on Cloud specific APIs (as an analogy to Terraform providers) and separation of runtime configurations
- multi cloud readiness
- hooks support for further automation,
- seamless integration with plugins mechanism with 3rd party tools like MLflow, pandas-profiling or Docker,
- suitable for easy integration with CI/CD tools for a true MLOps experience.

We at GetInData|Part of Xebia are strong advocates of the Kedro framework as our technology of choice for deploying robust and user-friendly MLOps platforms on many cloud platforms. With our open-source Kedro plugins, you can write your pipeline code and focus on the target model. Then, with the Kedro plugins, you deploy it to any supported platform (see: Running Kedro… everywhere? Machine Learning Pipelines on Kubeflow, Vertex AI, Azure and Airflow - GetInData) without changing the code, making local iterations fast and moving to cloud - seamless.

As of May 2023 we support:

- Google Cloud Platform (http://github.com/getindata/kedro-vertexai),
- Microsoft Azure (https://github.com/getindata/kedro-azureml),
- Amazon Web Services (https://github.com/getindata/kedro-sagemaker),
- Airflow (https://github.com/getindata/kedro-airflow-k8s),
- Kubeflow (https://github.com/getindata/kedro-kubeflow).

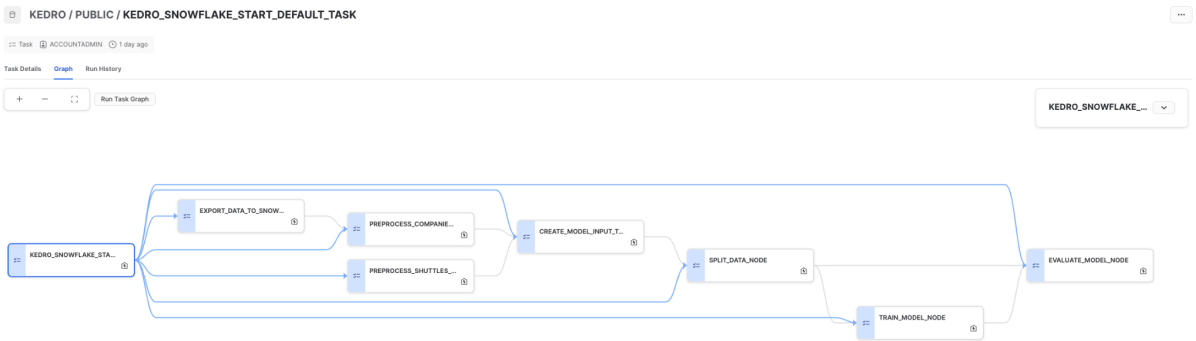
Now the time has come for Snowflake…

Kedro-Snowflake plugin behind the scenes

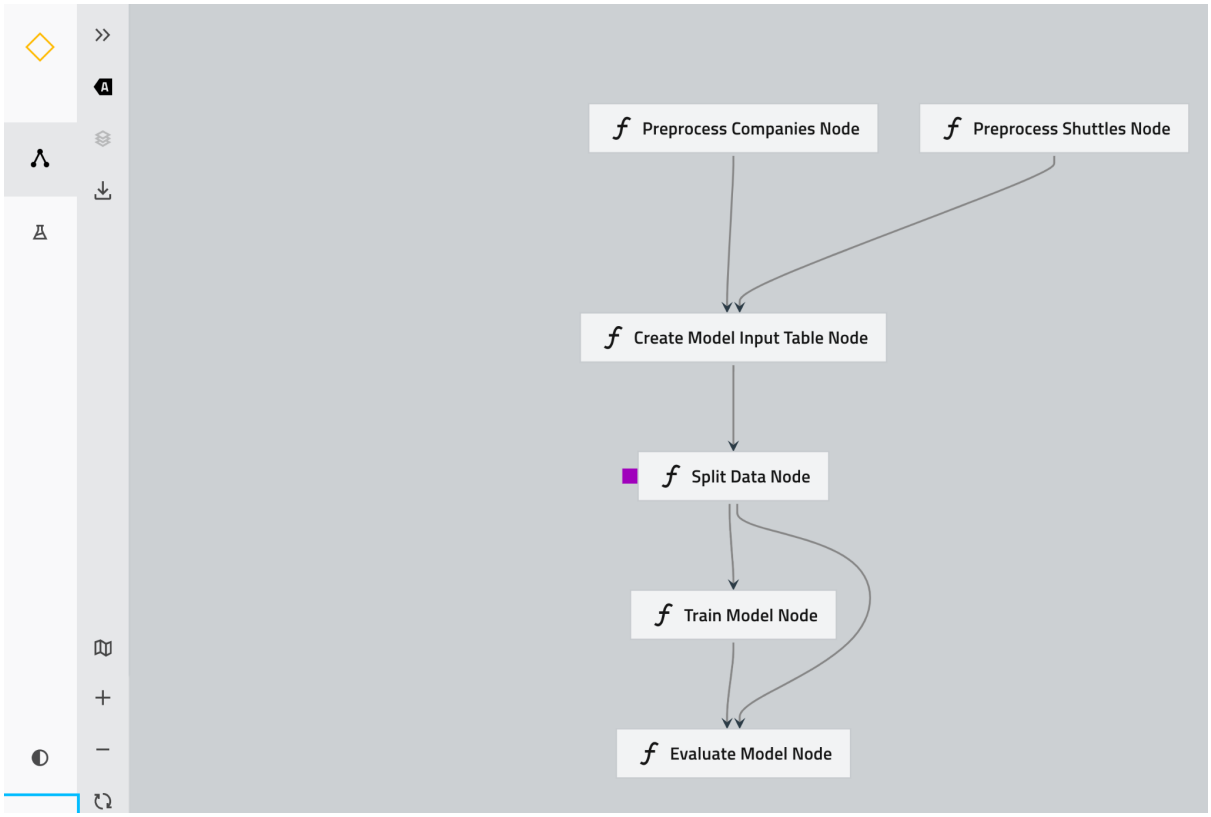
kedro-snowflake is our newest plugin that allows you to run full Kedro pipelines in Snowflake. Right now it supports:

- Kedro starter, to get you up to speed fast
- automatically creating Snowflake Stored Procedures from Kedro nodes (using Snowpark SDK)
- translating the Kedro pipeline into Snowflake task DAGs
- running the Kedro pipeline fully within Snowflake, without an external system
- using Kedro's official SnowparkTableDataSet
- automatically storing intermediate data results as Transient Tables (if Snowpark's DataFrames are used)

The core idea of this plugin is to **programmatically traverse a Kedro pipeline and translate its nodes into corresponding Stored Procedures and at the same time wrap them into Snowflake tasks, while preserving the inter-node dependencies to form exactly the same pipeline DAG on the Snowflake side**. The end result is a Snowflake DAG of tasks like this:



that correspond to the Kedro pipeline:



It also comes with a built-in **snowflights** (port of the official **spaceflights**, extended with Snowflake-related features) starter that will help to bootstrap your Snowflake-based ML projects in seconds.

Quick start - your ML pipeline in 3 steps with Kedro-Snowflake plugin

Let's start with the **snowflights** Kedro starter. First, prepare your environment (i.e. your preferred Python virtual environment). First, just install our kedro-snowlake plugin:

```
pip install "kedro-snowflake>=0.1.2"
```

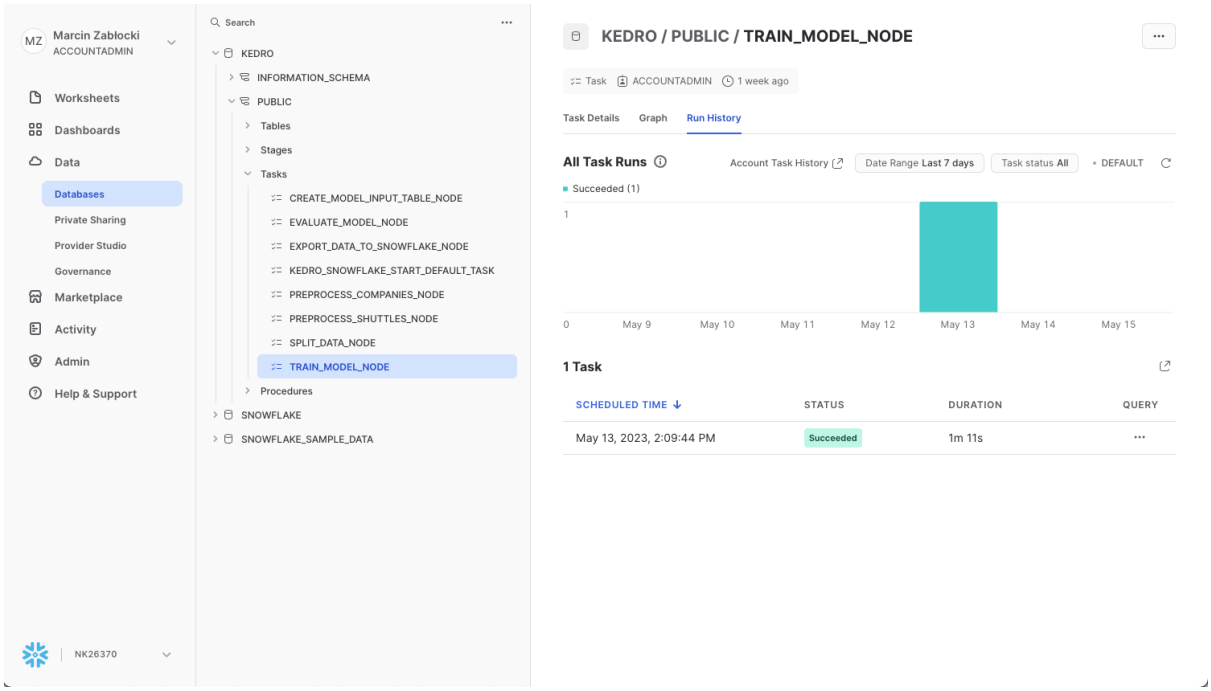
Next, create your first ML pipeline using Kedro and Snowlake. The starter will guide you through the Snowflake connection configuration, including the Snowlake account and warehouse details:

```
kedro new --starter=snowflights --checkout=0.1.2
```

Then run the starter pipeline:

```
kedro snowflake run --wait-for-completion
```

That's it! You can see the ML pipeline execution in the Snowflake UI:



and in the terminal:

```
(kedro-snow-tests-py3.8) (base) ➜  snowflights kedro snowflake run --wait-for-completion
```

This starter will showcase the Kedro-Snowflake integration, including the connection with Snowflake, transforming an ML Pipeline in Kedro to a Snowflake compatible format, and execution of the pipeline in the Snowflake environment. Feel free to build your own pipeline based on this starter or from scratch with our plugin. See more in the following plugin documentation: Kedro Snowflake plugin documentation!

Summary

In this short blog post we presented our newest kedro-snowflake plugin. Thanks to this plugin, you can build your ML pipelines in Kedro and execute them in a scalable Snowflake environment in three simple steps. Stay tuned for the second part of this blogpost in which **we are going to present the whole MLOps platform and ML model deployment with the kedro-snowflake plugin being the core component of it.**