Tags	MLOps, MLOps platform, snowflake, snowflake data cloud, kedro
Meta Description	How to implement the end-to-end MLOps platform on top of Snowflake powered by Kedro and MLflow.
Intro KB	How to implement the end-to-end MLOps platform on top of Snowflake powered by Kedro and MLflow. Dive into the cloud-agnostic MLOps Snowflake Platform
Autor	Marek Wiewiórka Marcin ZabłockiMichał Bryś
LINK do grafik	

From 0 to MLOps with ❄️ Part 2: Architecting the cloud-agnostic MLOps Platform for Snowflake Data Cloud

From 0 to MLOps with Snowflake ❄️

In the first part of the blogpost, we presented our kedro-snowflake plugin that enables you to run your Kedro pipelines on the Snowflake Data Cloud in 3 simple steps. This time we are going to **demonstrate how you can implement the end-to-end MLOps platform on top of Snowflake powered by Kedro and MLflow**. Inspired by the blog posts: 1 and 2 - in this article we present a **novel approach that tries to address the problem of missing External Internet Access in Snowpark.**

This feature would be an absolute game changer for implementing any integration with third party tools not natively available in the Snowflake ecosystem and according to the best of knowledge is on the Snowflake roadmap. So let's dive into the MLOps Platform for Snowflake Data Cloud.

MLOps Platform pillars - recap

What is Kedro? The MLOps Framework

Kedro is a widely-adopted MLOps framework in Python, that brings engineering back to the data science world to help productionize machine learning code seamlessly. Kedro lets you build machine learning pipelines that can work on cloud, edge or on-premises platforms. It's open source and offers tools for data scientists and engineers to create, share and collaborate on machine learning workflows. Additionally, **Kedro allows you to track the entire machine learning lifecycle from data preparation to model deploymen**t. (see also: Run machine learning pipelines on ❄️ Snowflake using Kedro 🔧 MLOPS TUTORIAL)

Z komentarzem [1]: zróbmy embed filmu

Kedro is a tool that can make your Machine Learning projects more scalable and flexible, while keeping things simple. It can run on any platform, whether cloud or edge computing and is designed to be easily scalable. With Kedro, data scientists and engineers can build machine learning workflows compatible with different platforms without worrying about scalability. In addition, Kedro provides flexibility in building machine learning workflows, supporting various data sources, models, and deployment targets. It makes it easier for teams to experiment with new technologies and techniques while maintaining a consistent pipeline.

What is MLflow? The platform for machine learning lifecycle management

MLflow is an open source platform that manages the entire lifecycle of machine learning models. It offers tools to track, manage and visualize workflows, from data preparation to model deployment. Additionally, MLflow promotes collaboration between data scientists and engineers by providing a shared language and understanding of the machine learning process.

Why should you consider MLflow in your MLOps toolbox? Here are the three main benefits MLFlow introduces:

1. **Efficient Machine Learning Development** - MLflow provides tools for tracking, managing and visualizing the entire machine learning workflow from data preparation to model deployment, which helps in the efficient development of machine learning models.
2. **Collaboration** - MLflow enables collaboration among data scientists and engineers by providing a common language and shared understanding of the machine learning process. MLflow makes it easier to work with other team members and share knowledge.
3. **Continuous Improvement** - MLflow provides tools for tracking model performance, which helps improve models over time. By continuously monitoring and analyzing model performance, data scientists can identify the areas where they need to improve their models or processes.

Both Kedro and MLflow are projects supported by Linux Foundation.

What is Terraform?

Terraform is an open-source software tool that enables the infrastructure as code (IaaC) approach in cloud computing, network automation and security. It provides tools to manage your infrastructure using simple, declarative configuration files instead of complex, error-prone manual configurations or scripts. Terraform helps you **automate provisioning, updating and deleting resources across multiple cloud providers** such as AWS, Azure and Google Cloud Platform (GCP).

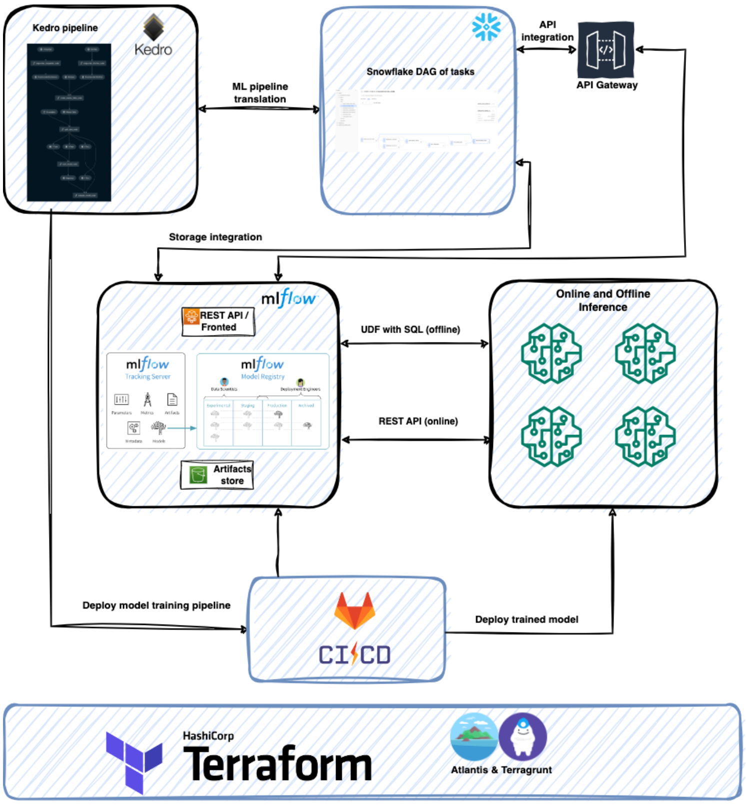
GetInData is also an active contributor to the official Snowflake Terraform Provider, in particular we have recently added support for external function translators that was required for the presented MLflow integration

MLOps Snowflake Platform - high level architecture

In the recent release of our Kedro-snowflake plugin we added beta support for MLflow integration.

Z komentarzem [2]: Dodalem info o 0.2.0 release with MLflow support @sylwia.kolpuc@getindata.com

The diagram below presents proposed MLOps platform architecture in the case of **AWS cloud**.



GCP and Azure deployment scenarios are very much the same except for:

- API Gateway (GCP) or API Management (Azure) for exposing MLflow API
- MLflow hosting - e.g. App Engine, Cloud Run (GCP) or Azure Container Apps, Azure Container Instances (Azure)
- Online model serving - Vertex Endpoints (GCP) or Azure ML Endpoints

Technical deep-dive into kedro-snowflake <-> MLflow integration

Kedro-snowflake <-> MLflow integration is based on the following concepts:

- Snowflake external functions that are used for wrapping POST requests to the MLflow instance. In the minimal setup the following wrapping external functions for MLflow REST API calls must be created:
    - Create run

    - Update run
    - Log param
    - Log metric
    - Search experiment
- Snowflake external function translators for changing the format of the data sent/received from the MLflow instance.
- Snowflake API integration for setting up a communication channel from the Snowflake instance to the cloud HTTPS proxy/gateway service where your MLflow instance is hosted (e.g. Amazon API Gateway, Google Cloud API Gateway or Azure API Management).
- Snowflake storage integration to enable your Snowflake instance to upload artifacts (e.g. serialized models) to the cloud storage (Amazon S3, Azure Blob Storage, Google Cloud Storage) used by the MLflow instance.
- Offline model deployment within Snowflake is handled by the MLflow-Snowflake plugin, which allows you to deploy models trained in popular frameworks (such as PyTorch, Scikit-Learn, TensorFlow, LightGBM, XGBoost, ONNX and others) natively in Snowflake as User-Defined Functions, which then can be called directly from SQL to efficiently (using vectorization) perform inference and obtain the models' predictions. The plugin is backed by the Snowflake team and is actively developed.
- Online model deployment with SageMaker Endpoints with the MLflow official plugin.

YOU® MLOps Snowflake Platform: Pros, cons and alternative approaches

Our MLOps platform assumes a **fully native integration with the Snowflake ecosystem and leverages it for data access, pipeline orchestration, model training as well as model deployment and inference**. Such an architecture has a number of advantages, just to name a few:

- simpler security setup (also thanks to limited data egress)
- fewer dependencies on external services
- substantially less data transfers

- lower pipeline nodes startup overheads when compared to SageMaker/AzureML/Vertex AI
- a unified data and machine learning platform

This is of course not a one-size-fits-all architecture and there are shortcomings that have not yet been addressed in the Snowflake Data Cloud, i.e.:

- GPU support
- preemptible data warehouses
- access to the external services
- multiple Python versions (already available in Public Preview)
- Docker support

Alternatively, one can only use Snowflake in a data-centric way and offload the machine learning pipeline orchestration and training to external systems such as Azure ML, Vertex AI or SageMaker. By doing so, a trade-off will be introduced - some of the ML/AI-native capabilities of the external services could potentially be leveraged, while the data transfer and cross-cloud connection/setup costs may appear. Such an approach may be, however, desirable in some cases when there is a need to reuse an external pipeline orchestrator service that is already used in organizations, such as Apache Airflow.

By building your core ML project and delivering the business value on top of the Kedro framework, later migration and **switching between those two approaches is possible with minimal effort**. So far, we've open sourced 6 major plugins for Kedro: Kedro-Snowflake, Kedro-AzureML, Kedro-VertexAI, Kedro-SageMaker, Kedro-Airflow and Kedro-Kubeflow (check our GitHub repos).

Z komentarzem [3]: dajmy bitly na to https://github.com/getindata

The below table summarizes the pros and cons of different approaches:

	Airflow	**SageMaker/Azure ML/Vertex AI**	**Snowflake/Snowpark**
Orchestration	🟩	🟩	⬜
Native data processing	🟨	🟨	🟩
Docker support	🟩	🟩	🟨
Native ML capabilities	🟨	🟩	⬜
Model deployment support (serving)	🟨	🟩	⬜
Maintenance	High	Low (serverless)	Low / Medium
Extensibility / Customizability	High	Low	Medium
Performance	Depends on setup	Varies	Low to very high
Experiment tracking	External	Built-in	External
GPU support	🟩	🟩	🟨
Language support	Python + Any (Docker)	Python + Any (Docker)	Python / Java / Scala + SQL

Learning curve	Medium	Medium to High	Low
Unstructured data support	🟩	🟩	🟩
Dataset versioning	🟨	🟨 (Azure - yes)	🟩
Open source	🟩	🟨	🟨
Dependency management	Docker	Docker	Anaconda / Package upload (Python / Java)
Kedro support	🟩	🟩	🟩
Distributed training support	▨	🟩	▨

Legend

🟩 Supported	🟨 Not supported	▨ Partially supported

Summary and future improvements

In this condensed blog post we presented our approach to architecting a cloud-agnostic MLOps platform on top of Snowflake Data Cloud, based on three building blocks: **Kedro, MLflow and Terraform**.The proposed solution solves the issue of missing external access to third party services in the Snowflake ecosystem - **this feature once implemented, would simplify the overall architecture by removing the need of using external function wrappers and API gateways.**

But it's not the end of the story - **stay tuned for the third part of this blogpost** in which **we are going to present how to extend this platform even further to support Large Language Models (LLMs)**.

If this topic caught your interest, it's the best moment because **on June 27 at the Snowflake Warsaw Meetup I will be giving a talk on From 0 to MLOps with ❄️ Snowflake Data Cloud**. I encourage you to join!

Z komentarzem [4]: na dole dajmy baner z ebookiem jak w tym poście https://getindata.com/blog/deploy-your-own-databricks-feature-store-azure-using-terraform/ a w buttonie napis SIGN UP AND GET EBOOK, WORKSHOP ACCES AND MLOPS MATERIALS