

WORKSHOP DAY ONSITE
OCTOBER 4, 2023

CONFERENCE DAY ONSITE
OCTOBER 5, 2023

WORKSHOP DAY ONLINE
OCTOBER 6, 2023

DATAMASS GDANSK SUMMIT

CLOUD AGAINST DATA

**The MLOps platform that makes you
productive, everywhere!**

Marek Wiewiórka
GetInData | Part of Xebia

About me



Marek Wiewiórka
PhD | Chief Data Architect at
GetInData | Cloud and Big Data A...



- Chief Data Architect @**GetInData | Part of Xebia**
- Research Assistant at the Warsaw University of Technology
- An Open source contributor to [Snowflake](#) [Terraform Provider](#), [SeQuiLa](#) and [Kedro](#) plugins
- Personally a keen long distance runner and gravel bikes enthusiast

GetInData - At a Glance

- Experts in **Big Data, Cloud, Analytics and ML/AI solutions**
- Team of 120+ consultants, ~60% senior level
- Experience in: **media, e-commerce, retail, fintech, banking, and telco**
- We work with **digital natives where data is core business** (Spotify, Truecaller, Acast, Volt), as well as with traditional enterprises where data is used for improvements
- **A go-to partner** for companies that need tailored and highly scalable data processing and analytics platforms that give competitive advantage and **unlock the full business potential of data**.

SOLUTION AREAS



MLOps & Modern Data Platforms



Data & ML engineering project accelerators

Read: [Kedro plugins](#), [DP Framework](#)



Stream processing & real-time analytics

Technologies



databricks



Looker



Airbyte



Kedro



jupyter



Flink



Apache Airflow



Selected USE CASES

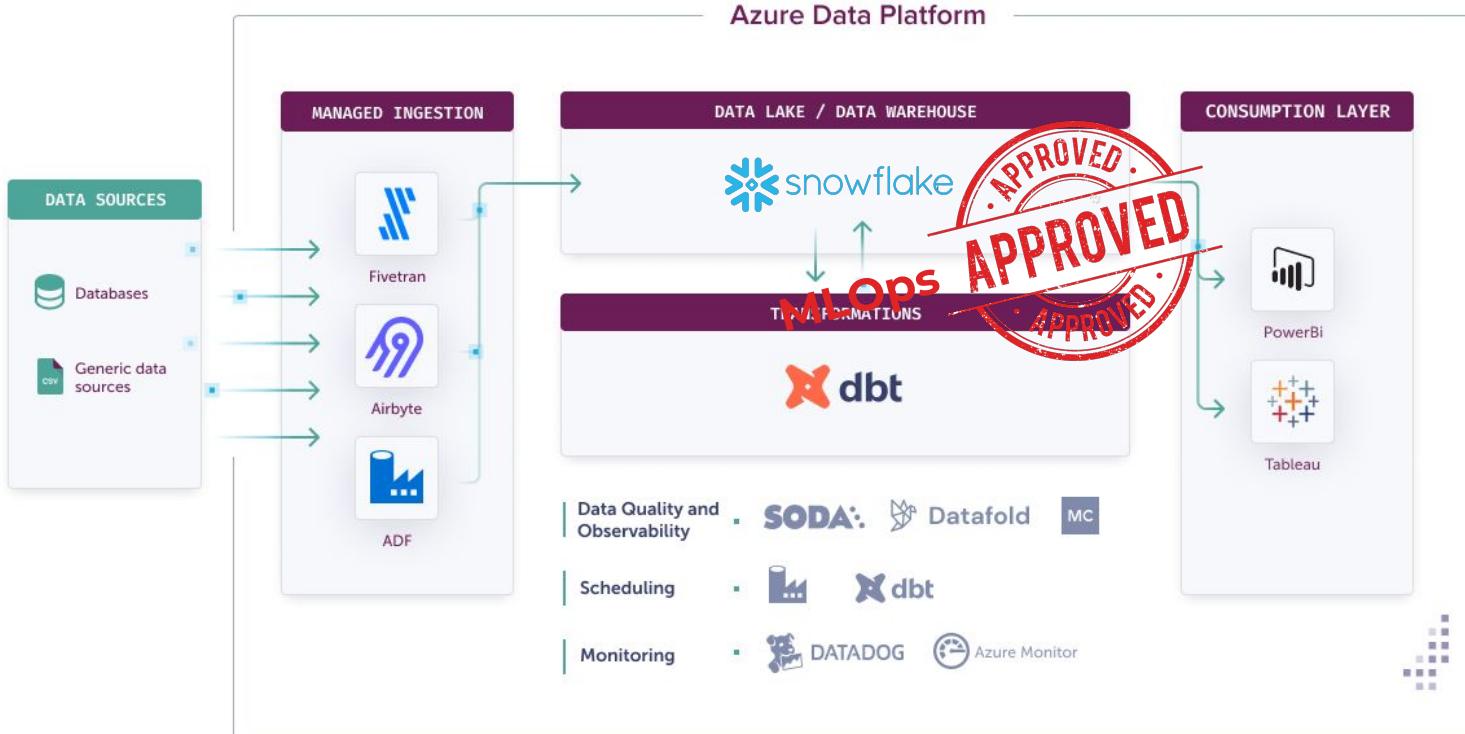
1. Volt.io (Fintech)

- Snowflake-based Modern Data Platform
- Just **4 months** to build from scratch to insights
- Strong focus on platform security
- The right mix of open-source and cloud-managed technologies

2. Other customer references



From MDP¹ to MDP (MLOps-enabled Data Platform)

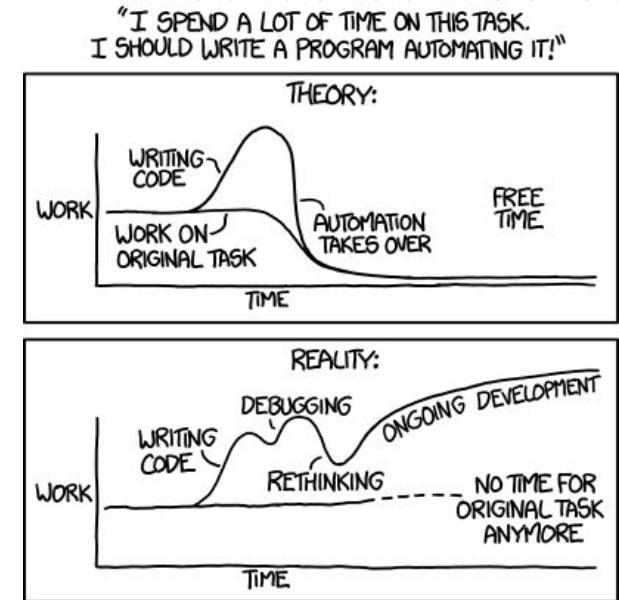


¹MDP - Modern Data Platform

What MLOps is (*not only*) about ?

- Application of the DevOps principles to ML world
- Managing ML model lifecycle
- Tools and platforms
- *Automation and processes*
- Infrastructure as Code

The ultimate goal is: **PRODUCTIVITY**



GID MLOps “Productivity Manifesto”

- Machine Learning and data science should be ***first-class*** citizens of Data Platforms
- Lightweight and KISS
- **Open** standards and cloud ***agnosticism***
- Short development ***feedback loop*** (incl. local dev)
- **Fast** new ML projects bootstrapping and ***standardization***
- Execution environment ***independent*** training pipelines
- Easy ***collaboration***
- ... MLOps capabilities provisioned ***in days not months***

ML projects in layers



Data
Scientist

Experimentation + EDA

Machine Learning frameworks

Example technologies:



XGBoost

ML projects in layers



Data
Scientist

Experimentation + EDA

Machine Learning frameworks

?

Execution environment

Data



MLOps / ML
Engineer

Example technologies:



XGBoost



Building blocks of the GID MLOps



Data
Scientist

Experimentation + EDA

Machine Learning frameworks

Portable
MLOps
framework

Experiment
tracking and
collaboration

IaC and
automation



Cloud Integrations (incl. GID Kedro plugins)

Execution environment

Data



MLOps / ML
Engineer

Example technologies:



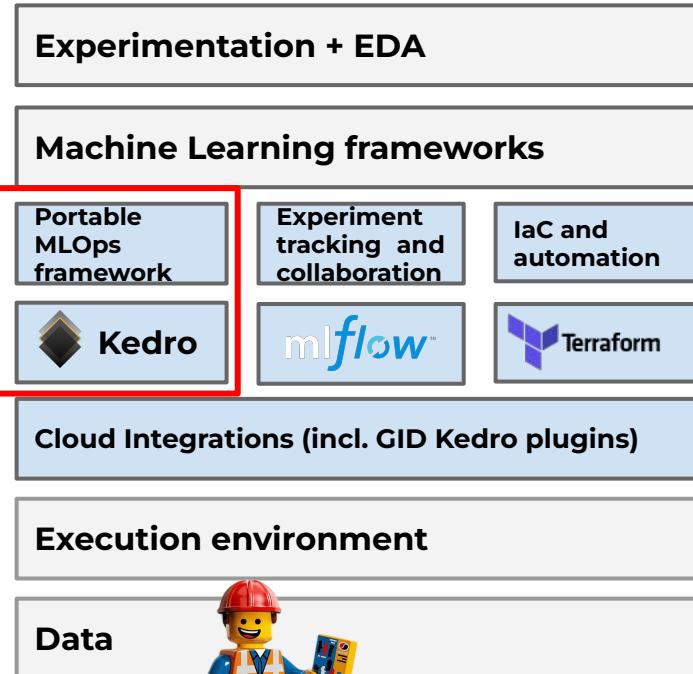
XGBoost



Building blocks of the GID MLOps



Data
Scientist



MLOps / ML
Engineer

Example technologies:



XGBoost



What is Kedro?



Kedro

= Software
Engineering
Principles + Data Science

Kedro is an open-source Python framework
for creating reproducible, maintainable and modular data science code.



McKinsey donates machine
learning pipeline tool Kedro to the
Linux Foundation

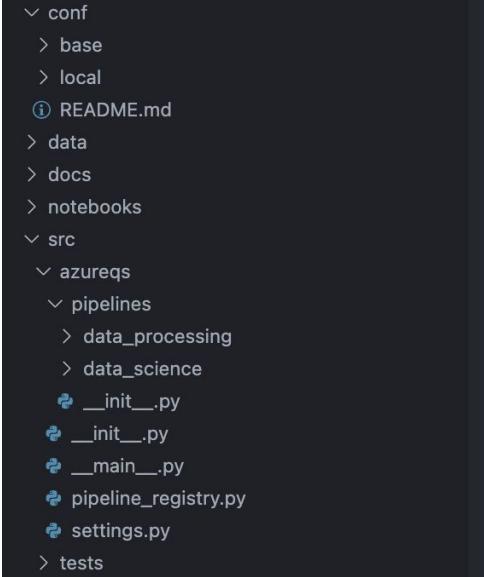


What features does Kedro have?

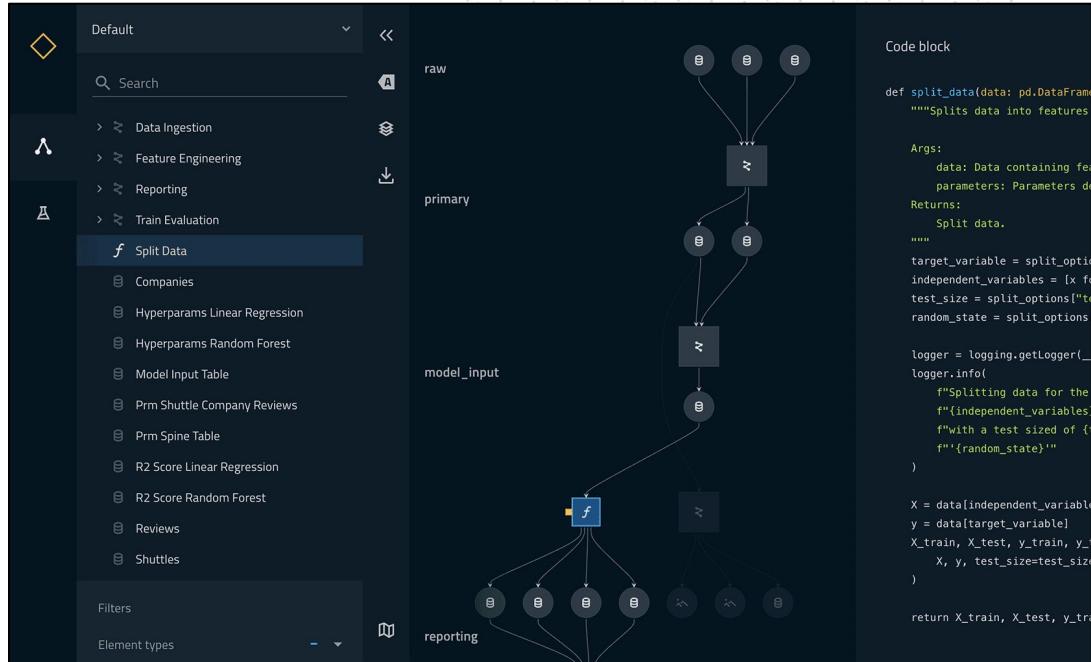
```
✓ conf
  > base
  > local
  ⓘ README.md
  > data
  > docs
  > notebooks
  ✓ src
    ✓ azureqs
      ✓ pipelines
        > data_processing
        > data_science
        ⚡ __init__.py
      ⚡ __init__.py
      ⚡ __main__.py
      ⚡ pipeline_registry.py
      ⚡ settings.py
    > tests
```

**Well defined
project structure
+ project starters**

What features does Kedro have?



**Well defined
project structure
+ project starters**



**Nodes & pipelines
abstractions**

Kedro pipeline - data science & engineering



Kedro node

```
6 def create_pipeline(**kwargs) -> Pipeline:  
7     return pipeline(  
8         [  
9             node(  
10                 func=preprocess_companies,  
11                 inputs="companies",  
12                 outputs="preprocessed_companies",  
13                 name="preprocess_companies_node",  
14             ),  
15             node(  
16                 func=preprocess_reviews,  
17                 inputs="reviews",  
18                 outputs="preprocessed_reviews",  
19                 name="preprocess_reviews_node",  
20             ),  
21             node(  
22                 func=create_model_input_table,  
23                 inputs=["preprocessed_reviews", "  
24                     outputs="model_input_table",  
25                     name="create_model_input_table_node",  
26                 ),  
27             ],  
28         )
```

```
49     def create_model_input_table(  
50         reviews: pd.DataFrame, companies: pd.DataFrame, ratings: pd.DataFrame  
51     ) -> pd.DataFrame:  
52         """Combines all data to create a model input table.  
53  
54         Args:  
55             reviews: Preprocessed data for reviews.  
56             companies: Preprocessed data for companies.  
57             ratings: Raw data for ratings.  
58         Returns:  
59             Model input table.  
60  
61         """  
62         reviews_with_ratings = reviews.merge(ratings, left_on="id", right_on="rating_id")  
63         model_input_table = reviews_with_ratings.merge(  
64             companies, left_on="company_id", right_on="id"  
65         )  
66         model_input_table = model_input_table.dropna()  
67         return model_input_table
```

What about parameters?

```
6 def create_pipeline(**kwargs) -> Pipeline:
7     return pipeline(
8         [
9             node(
10                 func=split_data,
11                 inputs=["model_input_table", "params:model_options"], # Parameters
12                 outputs=["X_train", "X_test", "y_train", "y_test"],
13                 name="split_data_node",
14             ),
15             node(
16                 func=train_model,
17                 inputs=["X_train", "y_train"],
18                 outputs="regressor",
19                 name="train_model_node",
20             ),
21             node(
22                 func=evaluate_model,
23                 inputs=["regressor", "X_test", "y_test"],
24                 outputs=None,
25                 name="evaluate_model_node",
26             ),
27         ],
28     )
```

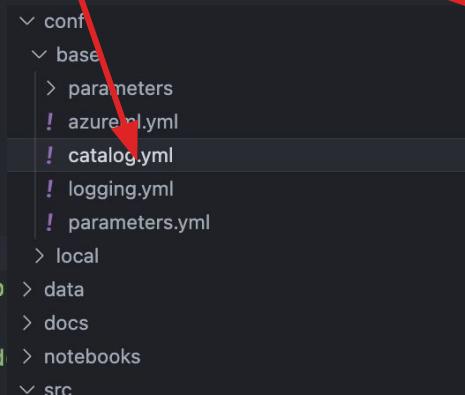
What about parameters?

```
6 def create_pipeline(**kwargs) -> Pipeline:  
7     return pipeline(  
8         [  
9             node(  
10                func=split_data,  
11                inputs=["model_input_table", "  
12                    outputs=["X_train", "X_test",  
13                        name="split_data_node",  
14                    ),  
15                    node(  
16                        func=train_model,  
17                        inputs=["X_train", "y_train"],  
18                        outputs="regressor",  
19                        name="train_model_node",  
20                    ),  
21                    node(  
22                        func=evaluate_model,  
23                        inputs=["regressor", "X_test",  
24                            outputs=None,  
25                            name="evaluate_model_node",  
26                        ),  
27                    ]  
28            )  
29        ]
```

```
    ↘ conf  
    ↘ base  
    ↘ parameters  
    ! data_processing.yml  
    ! data_science.yml  
    ! azureml.yml  
    ! catalog.yml  
    ! logging.yml  
    ! parameters.yml  
    > local  
    > data  
    > docs  
    2  
    3     model_options:  
    4         test_size: 0.2  
    5         random_state: 3  
    6         features:  
    7             - engines  
    8             - passenger_capacity  
    9             - crew  
   10            - d_check_complete  
   11            - moon_clearance_complete  
   12            - iata_approved  
   13            - company_rating  
   14            - review_scores_rating  
   15
```

What about data - Kedro data catalog!

```
6 def create_pipeline(**kwargs) -> Pipeline:
7     return pipeline(
8         [
9             node(
10                 func=preprocess_companies,
11                 inputs="companies",
12                 outputs="preprocessed_companies",
13                 name="preprocess_companies_node",
14             ),
15             node(
16                 func=preprocess_reviews,
17                 inputs="reviews",
18                 outputs="preprocessed_reviews",
19                 name="preprocess_reviews_node",
20             ),
21             node(
22                 func=create_model_input_table,
23                 inputs=["preprocessed_reviews", "p",
24                 outputs="model_input_table",
25                 name="create_model_input_table_nod
26             ),
27         ]
28     )
```

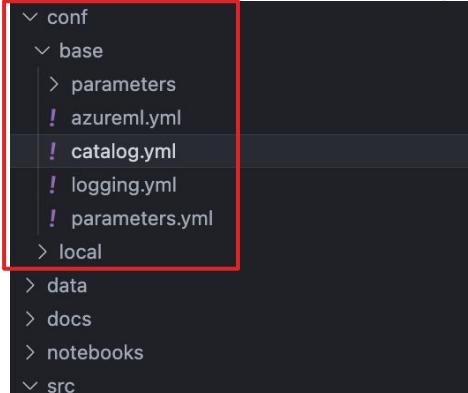


```
42 companies:
43     type: pandas.CSVDataSet
44     filepath: data/01_raw/companies.csv
45
46 reviews:
47     type: pandas.ParquetDataSet
48     filepath: data/01_raw/reviews.parquet
49
50 pictures:
51     type: pillow.ImageDataSet
52     filepath: data/01_raw/images/*.jpg
53
```

Configuration and code separation with envs!

```
companies:  
  type: pandas.CSVDataSet  
  filepath: data/01_raw/companies.csv  
  
reviews:  
  type: pandas.ParquetDataSet  
  filepath: data/01_raw/reviews.parquet  
  
pictures:  
  type: pillow.ImageDataSet  
  filepath: data/01_raw/images/*.jpg  
  
companies:  
  type: pandas.CSVDataSet      Cloud catalog.yml  
  filepath: abfs://my_blob_container/data/01_raw/companies.csv  
  
reviews:  
  type: pandas.SQLQueryDataSet  
  sql: "select * from reviews;"  
  credentials: db_credentials  
  
pictures:  
  type: kedro_azureml.AzureMLFileDataSet  
  dataset: my_dataset_from_azureml  
  filepath: data/01_raw/images/*.jpg
```

Data Catalog

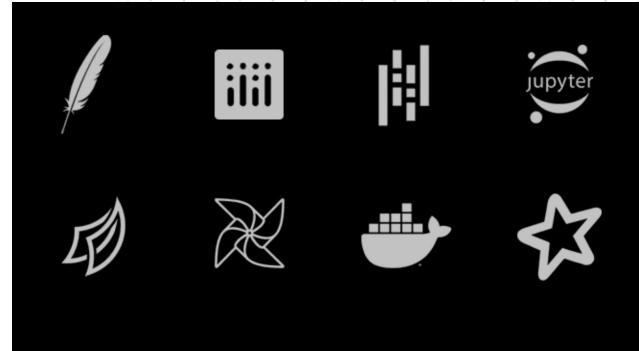
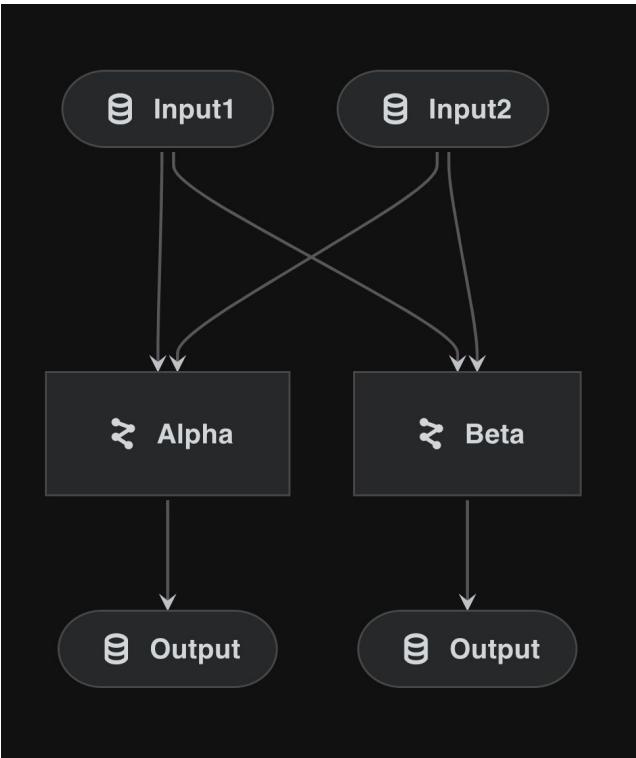


```
42 companies:  
43   type: pandas.CSVDataSet  
44   filepath: data/01_raw/companies.csv  
45  
46 reviews:  
47   type: pandas.ParquetDataSet  
48   filepath: data/01_raw/reviews.parquet  
49  
50 pictures:  
51   type: pillow.ImageDataSet  
52   filepath: data/01_raw/images/*.jpg  
53
```

```
    <!-- conf/base -->  
    <!-- conf/base/local -->  
    <!-- conf/base/logging.yml -->  
    <!-- conf/base/parameters.yml -->  
    <!-- conf/base/parameters -->  
    <!-- conf/base/base -->
```

What other features does Kedro have?

Modular and dynamic pipelines!



Extensibility & Integrations

Kedro can be integrated with multiple industry leading solutions, including:
Apache Spark, Pandas, Dask, Matplotlib, Plotly,
fsspec, Apache Airflow, Jupyter Notebook
and Docker.

Stay tuned for our blog post soon!

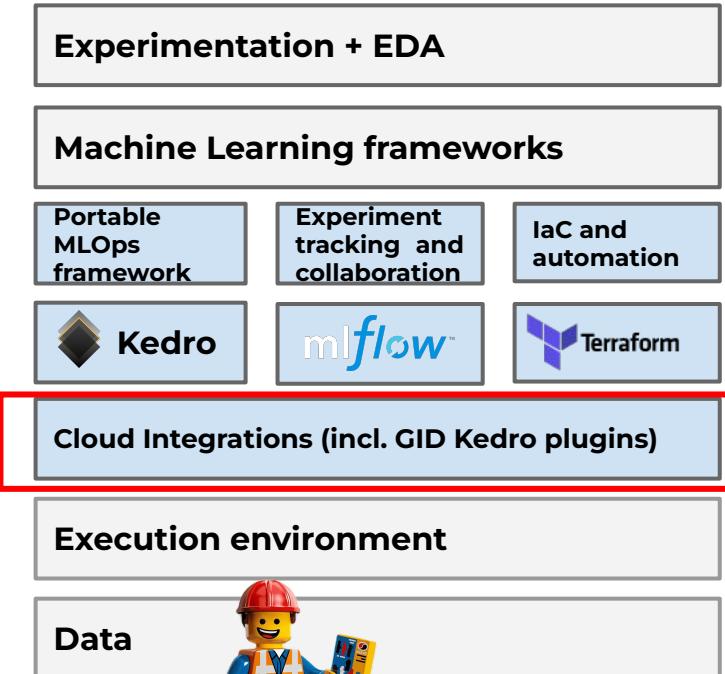
Are we done yet?



Building blocks of the GID MLOps



Data
Scientist



MLOps / ML
Engineer

Example technologies:



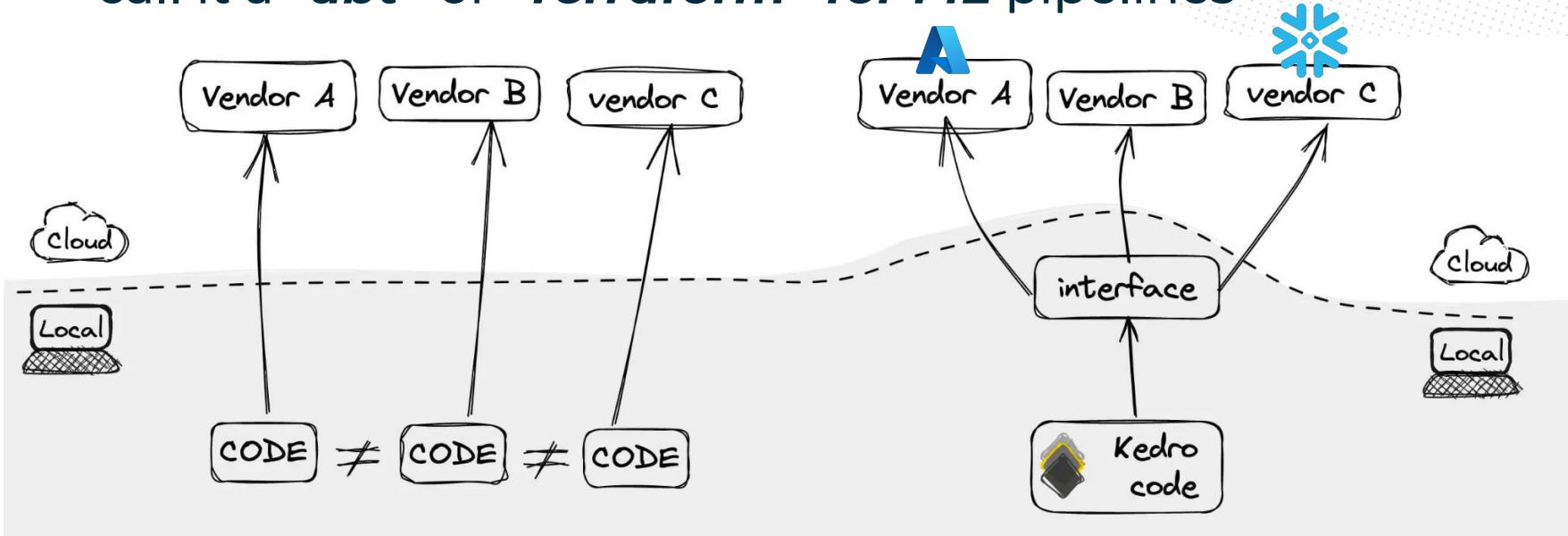
XGBoost



GID MLOps Platform

Kedro plugins to run your ML pipelines, everywhere!

- Kedro is claimed to be a “React” for ML ... but we prefer to call it a “**dbt**” or “**Terraform**” for ML pipelines



Source: [Xebia blog](#)

Write once - run (almost) everywhere



Kedro



Kedro Vertex AI (GCP)

github.com/getindata/kedro-vertexai



Kedro Sagemaker (AWS)

github.com/getindata/kedro-sagemaker



Kedro Kubeflow (Kubernetes)

github.com/getindata/kedro-kubeflow



Kedro AzureML (Azure)

github.com/getindata/kedro-azureml



Kedro Snowflake (all clouds)

github.com/getindata/kedro-snowflake



In progress: **Kedro DBX**

Read more on our blog: [Running Kedro... everywhere? Machine Learning Pipelines on Kubeflow, Vertex AI, Azure and Airflow](#)

Write once - run (almost) everywhere



Kedro



In progress: **Kedro DBX**



Kedro Vertex AI (GCP)

github.com/getindata/kedro-vertexai



Kedro Sagemaker (AWS)

github.com/getindata/kedro-sagemaker



Kedro Kubeflow (Kubernetes)

github.com/getindata/kedro-kubeflow



Kedro AzureML (Azure)

github.com/getindata/kedro-azureml



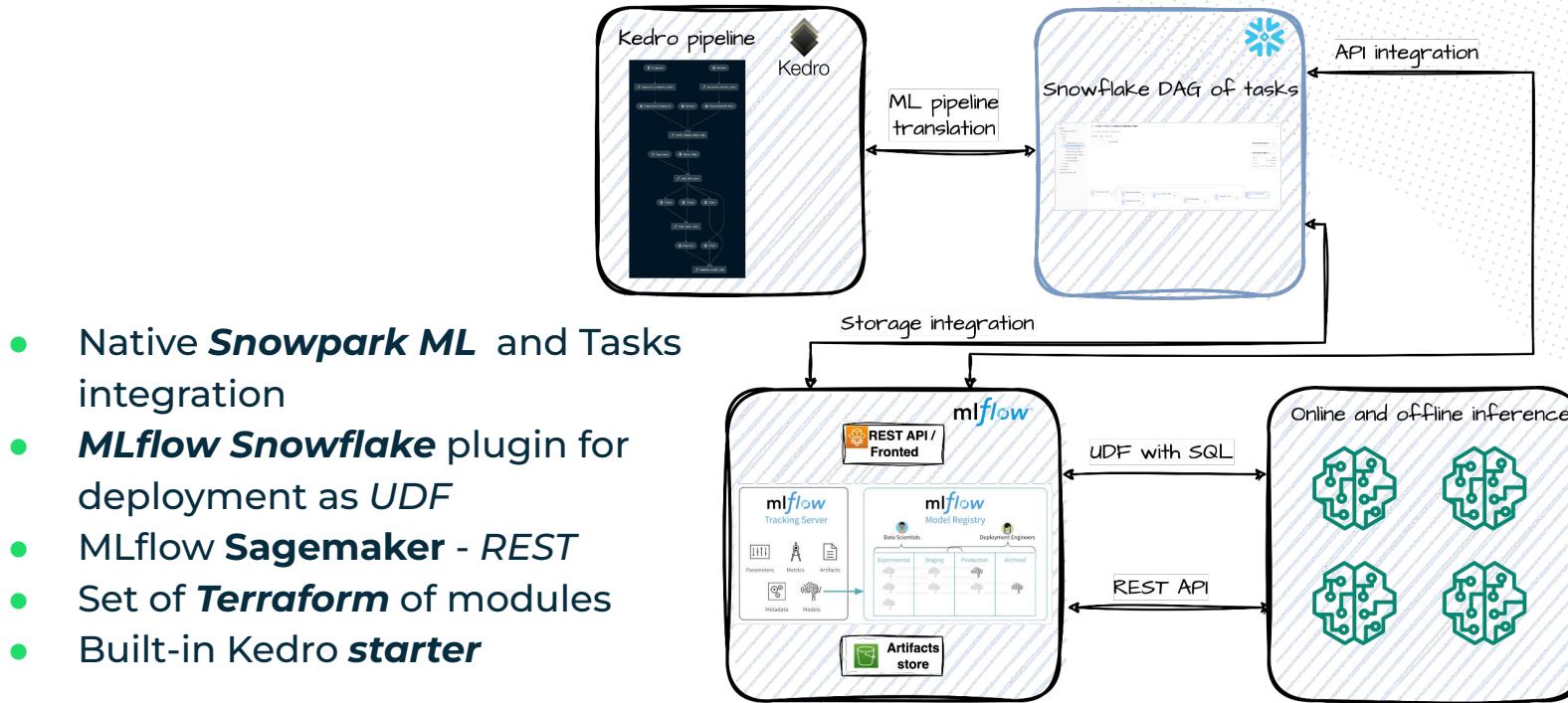
Kedro Snowflake (all clouds)

github.com/getindata/kedro-snowflake



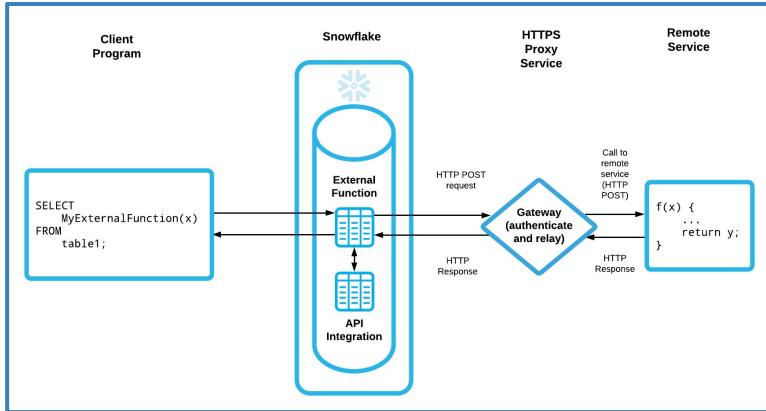
Read more on our blog: [Running Kedro... everywhere? Machine Learning Pipelines on Kubeflow, Vertex AI, Azure and Airflow](#)

Kedro-Snowflake - putting it all together...

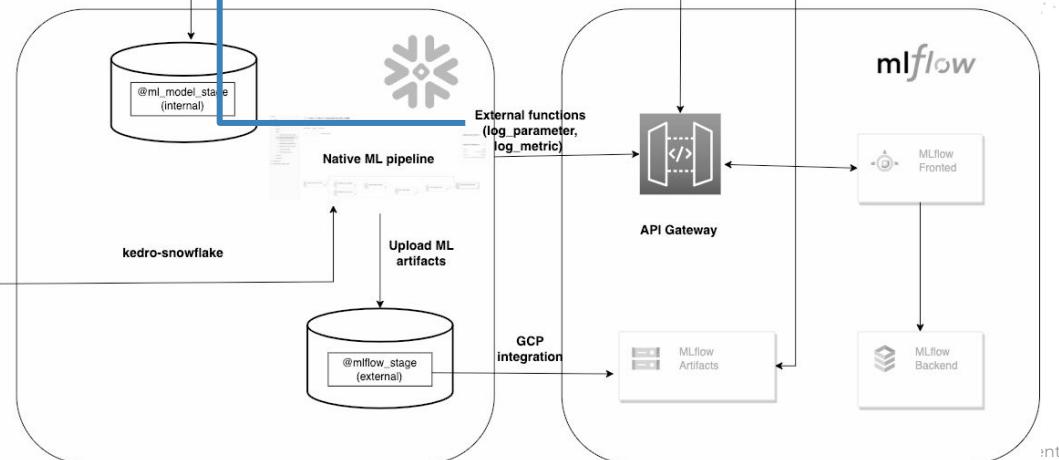




MLOps Platform for Snowflake - GCP



- **Snowflake external functions and API Gateway**





MLOps Platform for Snowflake- GCP

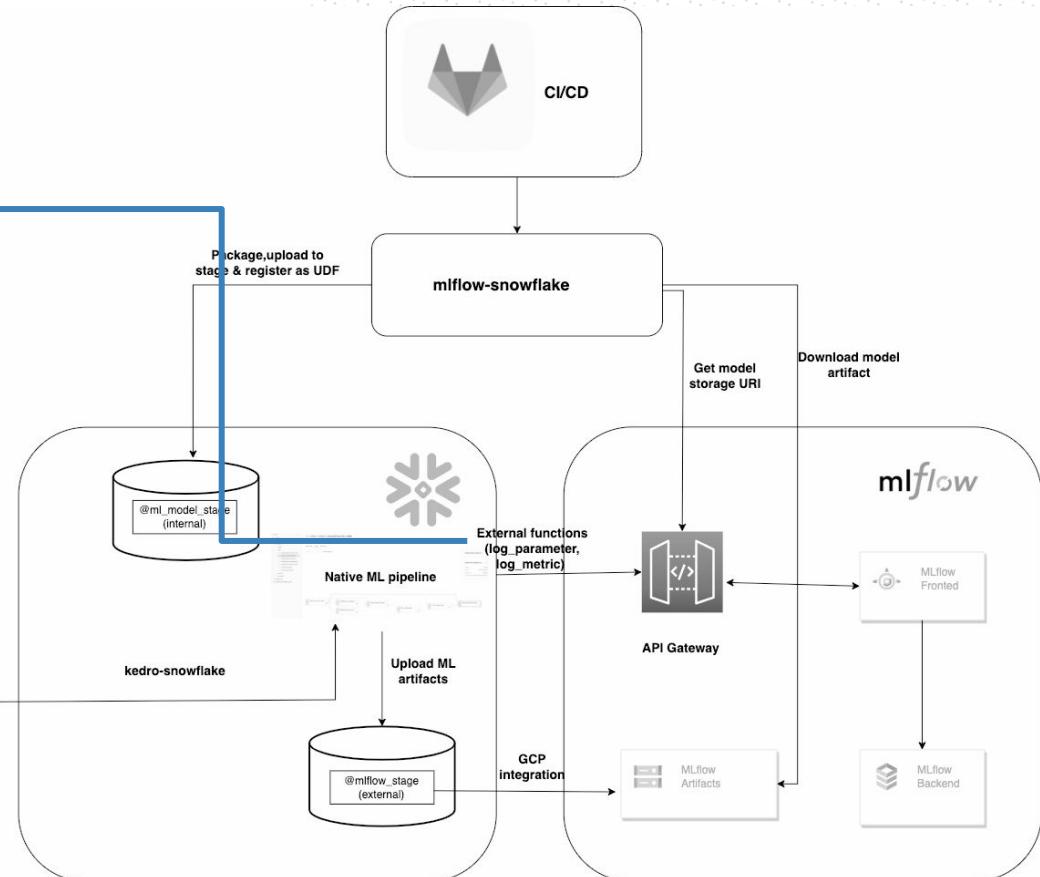
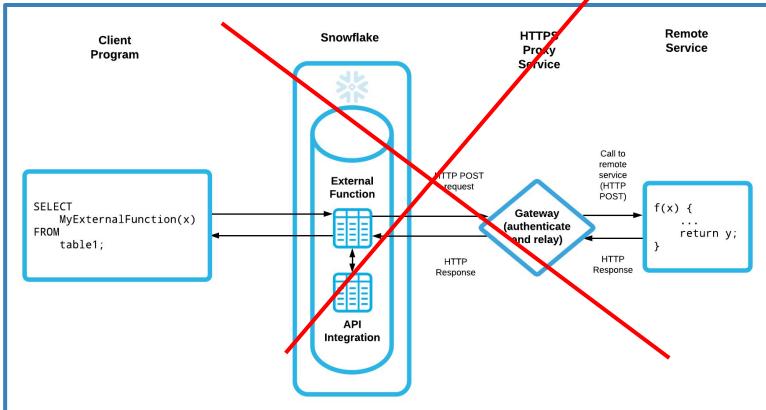
- Glue code for requests/responses to MLflow API
- PR to the Snowflake provider

```
34
35 resource "snowflake_external_function" "mlflow_run_create" {
36     name      = upper("mlflow_run_create")
37     database  = var.database_name
38     schema    = var.schema_name
39     arg {
40         name = "experiment_id"
41         type = "varchar"
42     }
43     return_type          = "OBJECT"
44     return_behavior       = "VOLATILE"
45     api_integration      = snowflake api integration.mlflow qcp.name
46     request_translator   = "${var.database_name}.${var.schema_name}.${snowflake_function.mlflow_run_create_req.name}"
47     response_translator  = "${var.database_name}.${var.schema_name}.${snowflake_function.mlflow_generic_res.name}"
48     url_of_proxy_and_resource = "${var.api_gateway_url}/api/2.0/mlflow/runs/create"
49 }
```

```
36
37     resource "snowflake_function" "mlflow_run_create_req" {
38         name      = upper("mlflow_run_create_req")
39         database  = snowflake_database.db.name
40         schema    = snowflake_schema.schema.name
41         arguments {
42             name = "event"
43             type = "OBJECT"
44         }
45         comment           = "Request translator for MLflow create run"
46         return_type        = "OBJECT"
47         language          = "javascript"
48         statement          = <<EOH
49             let experimentId = EVENT.body.data[0][1]
50             let timestamp = new Date().getTime();
51             return { "body": { "experiment_id": experimentId, start_time: timestamp } }
52         EOH
53     }
```



MLOps Platform for Snowflake - AWS/Azure

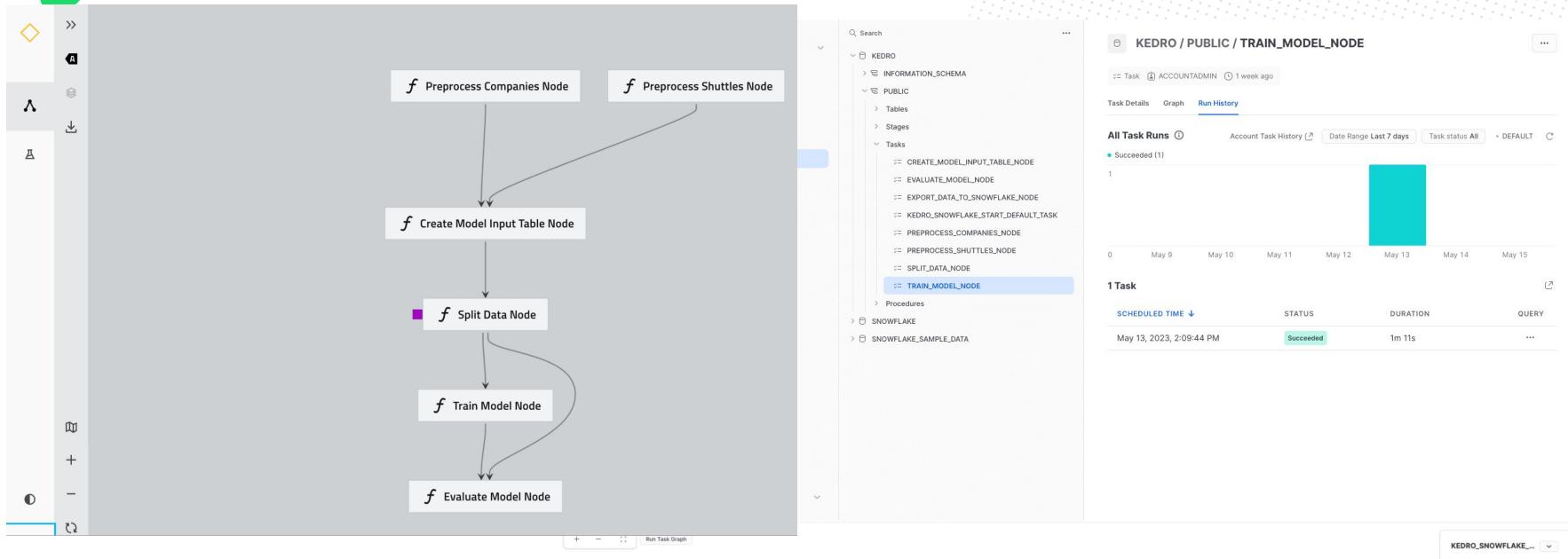


- Snowflake external **integrations** and **network rules**
- **No need for API Gateway**





MLOps Platform for Snowflake



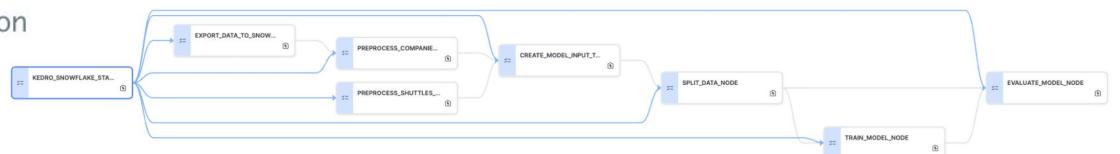
The screenshot displays the MLOps Platform interface for managing a KEDRO pipeline on a Snowflake database.

Task Graph View: On the left, a vertical sidebar contains icons for navigation and search. The main area shows a task graph with nodes: "Preprocess Companies Node", "Preprocess Shuttles Node", "Create Model Input Table Node", "Split Data Node" (highlighted with a purple square), "Train Model Node", and "Evaluate Model Node". Arrows indicate the flow from Preprocess nodes to Create Model Input, then to Split Data, Train Model, and finally Evaluate Model.

File Explorer View: On the right, a file explorer sidebar shows the project structure under "KEDRO". It includes sections for INFORMATION_SCHEMA, PUBLIC (Tables, Stages, Tasks, Procedures), SNOWFLAKE, and SNOWFLAKE_SAMPLE_DATA. The "TRAIN_MODEL_NODE" task is currently selected.

Task Run History View: The main panel shows the "KEDRO / PUBLIC / TRAIN_MODEL_NODE" task run details. It includes tabs for Task Details, Graph, and Run History. The Run History tab displays a single successful run on May 13, 2023, at 2:09:44 PM, which completed in 1m 11s.

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





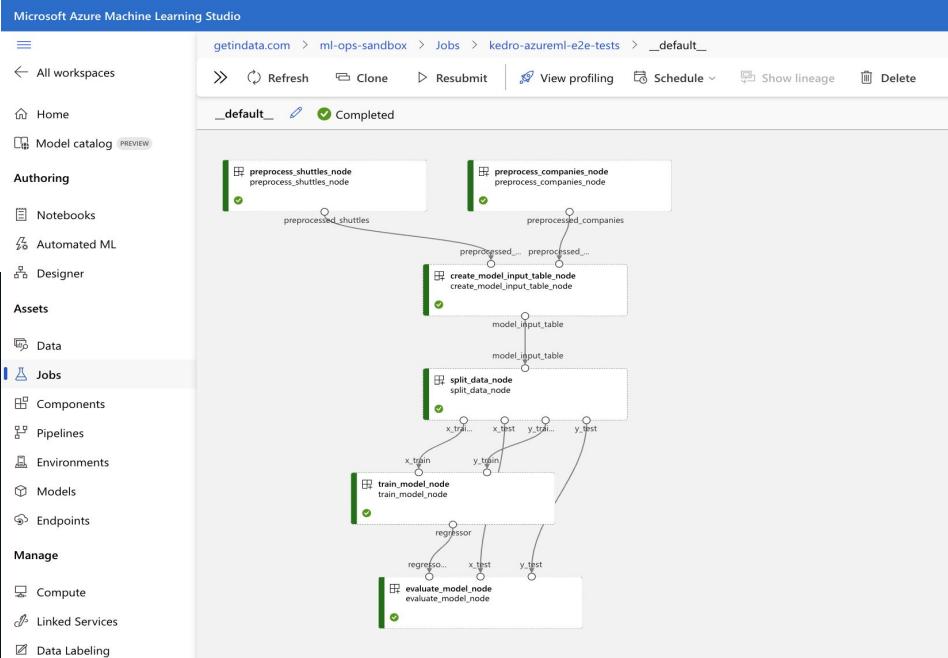
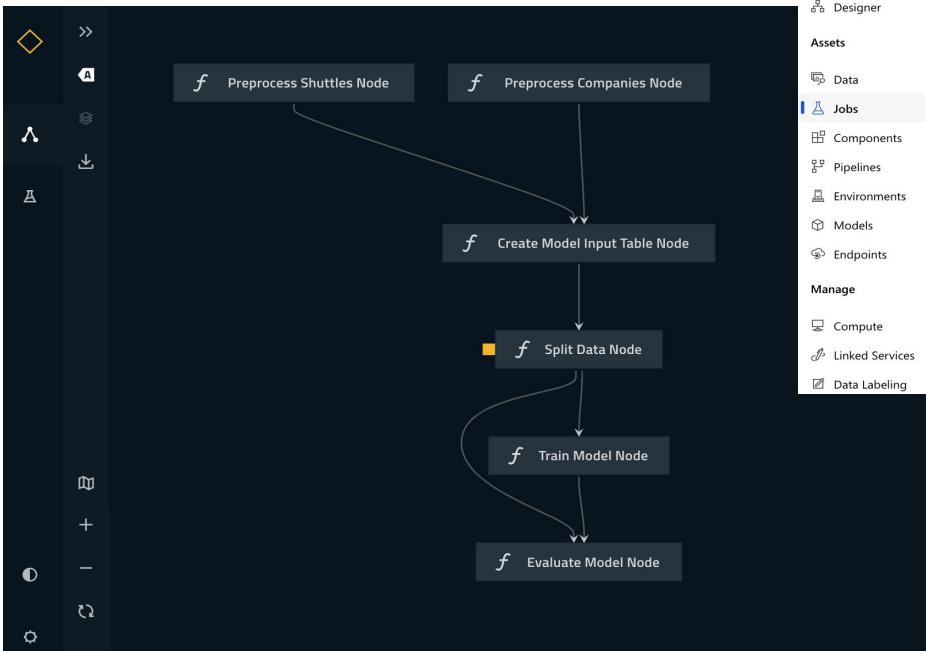
- Support for native Snowflake Tables and Stages in Kedro Data catalog

```
46 companies_snowflake:
47     type: kedro_datasets.snowflake.SnowparkTableDataSet
48     table_name: companies_snowflights_starter
49     credentials: snowflake
50     save_args:
51         mode: overwrite
52
53
54 preprocessed_shuttles:
55     type: kedro_snowflake.datasets.native.SnowflakeStageFileDataSet
56     stage: "@KEDRO_SNOWFLAKE_TEMP_DATA_STAGE" # <-- Snowflake stage to store data in
57     filepath: data/02_intermediate/preprocessed_shuttles.csv # <-- file path within the stage
58     credentials: snowflake # <-- credentials to connect to Snowflake (the same as for SnowparkTableDataSet)
59     dataset: # <-- dataset key defines the dataset type to use
60         type: pandas.CSVDataSet # <-- specify any params for the nested dataset here
```



MLOps Platform for Azure

- **Kedro** AzureML plugin
- Azure ML builtin “**MLflow**”
- Set of **Terraform** of modules



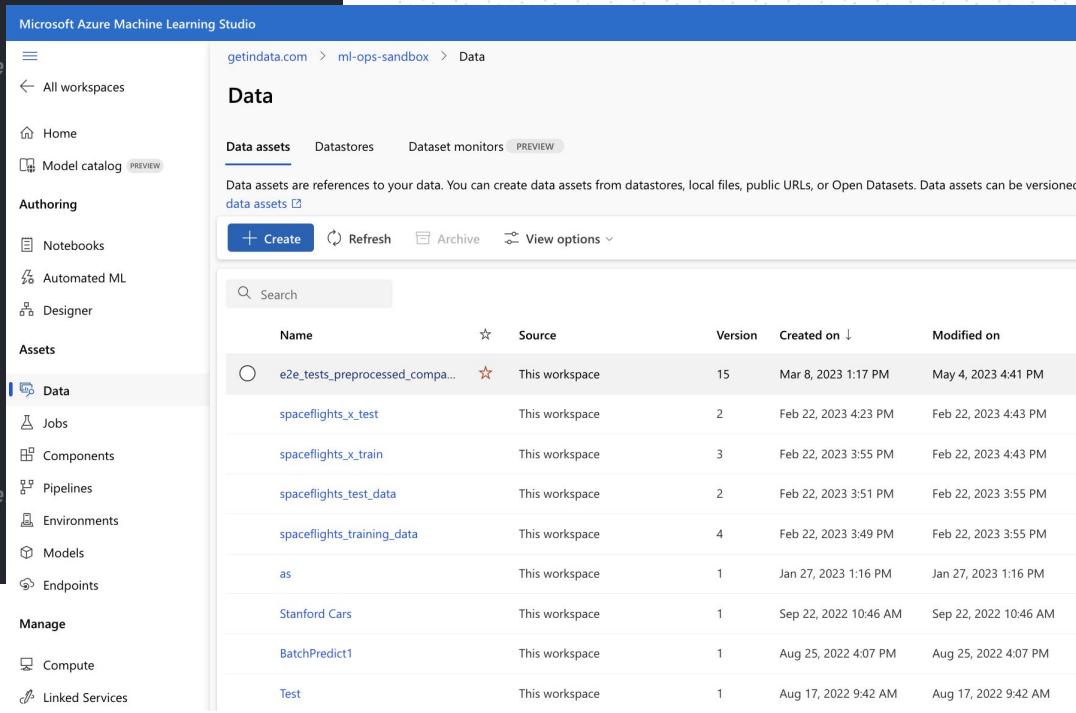
Kedro-AzureML - datasets

```

54 my_pandas_dataframe_dataset:
55   type: kedro_azureml.datasets.AzureMLPandasDataSet
56   azureml_dataset: my_new_azureml_dataset
57
58 # if version is not provided, the latest dataset version will be used
59 azureml_dataset_load_args:
60   version: 1
61
62
63 processed_images:
64   type: kedro_azureml.datasets.AzureMLFileDataSet
65   dataset: pillow.ImageDataSet
66   filename_suffix: '.png'
67   azureml_dataset: processed_images
68   azureml_dataset_save_args:
69     create_new_version: true
70
71 # if version is not provided, the latest dataset version will be used
72 azureml_dataset_load_args:
73   version: 1

```

Microsoft Azure Machine Learning Studio



The screenshot shows the Microsoft Azure Machine Learning Studio interface. On the left, there is a sidebar with navigation links: All workspaces, Home, Model catalog (PREVIEW), Authoring (Notebooks, Automated ML, Designer), Assets (Data, Jobs, Components, Pipelines, Environments, Models, Endpoints), and Manage (Compute, Linked Services). The main area is titled "Data" and shows the "Data assets" tab selected. The URL in the browser bar is getindata.com/ml-ops-sandbox>Data. The page displays a table of data assets:

Name	Source	Version	Created on	Modified on
e2e_tests_preprocessed_compa...	This workspace	15	Mar 8, 2023 1:17 PM	May 4, 2023 4:41 PM
spaceflights_x_test	This workspace	2	Feb 22, 2023 4:23 PM	Feb 22, 2023 4:43 PM
spaceflights_x_train	This workspace	3	Feb 22, 2023 3:55 PM	Feb 22, 2023 4:43 PM
spaceflights_test_data	This workspace	2	Feb 22, 2023 3:51 PM	Feb 22, 2023 3:55 PM
spaceflights_training_data	This workspace	4	Feb 22, 2023 3:49 PM	Feb 22, 2023 3:55 PM
as	This workspace	1	Jan 27, 2023 1:16 PM	Jan 27, 2023 1:16 PM
Stanford Cars	This workspace	1	Sep 22, 2022 10:46 AM	Sep 22, 2022 10:46 AM
BatchPredict1	This workspace	1	Aug 25, 2022 4:07 PM	Aug 25, 2022 4:07 PM
Test	This workspace	1	Aug 17, 2022 9:42 AM	Aug 17, 2022 9:42 AM

MLOps Azure DevOps pipeline

Azure DevOps marcin-getindata / gidmlopsv7 / Pipelines / MLOps demo pipeline / 20230510.1

G gidmlopsv7 +

- Overview
- Boards
- Repos
- Pipelines
- Pipelines**
- Environments
- Releases
- Library
- Task groups
- Deployment groups

← Jobs in run #20230510.1
MLOps demo pipeline

Build & run

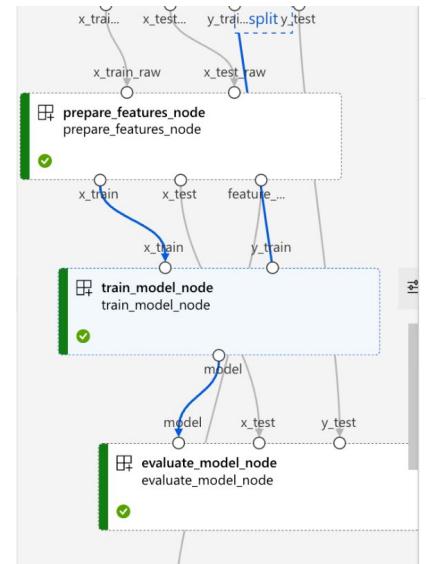
- ▼ MLOps Pipeline 42s
 - Initialize job 1s
 - Checkout MLOps by GetInData ... 1s
 - Login to ACR <1s
 - Pull latest image 1s
 - Build 39s
 - Push
 - Run Kedro pipeline in Azure ML
 - Post-job: Checkout MLOps by Get...
...

Build

Line Number	Progress Bar	Details
488	<div style="width: 78.2%;"></div>	78.2/78.2 KB 238.3
489	<div style="width: 100%;"></div>	Collecting pytoolconfig[global]>=1.2.2
490	<div style="width: 100%;"></div>	Downloading pytoolconfig-1.2.4-py3-none-any.whl (16 kB)
491	<div style="width: 100%;"></div>	Downloading pytoolconfig-1.2.2-py3-none-any.whl (16 kB)
492	<div style="width: 100%;"></div>	Collecting appdirs>=1.4.4
493	<div style="width: 100%;"></div>	Downloading appdirs-1.4.4-py2.py3-none-any.whl (9.6 kB)
494	<div style="width: 100%;"></div>	Collecting idna<4,>=2.5
495	<div style="width: 100%;"></div>	Downloading idna-3.4-py3-none-any.whl (61 kB)
496	<div style="width: 61.5%;"></div>	61.5/61.5 KB 224.5
497	<div style="width: 100%;"></div>	Collecting PySocks!=1.5.7,>=1.5.6
498	<div style="width: 100%;"></div>	Downloading PySocks-1.7.1-py3-none-any.whl (16 kB)
499	<div style="width: 100%;"></div>	Collecting jeepney>=0.6
500	<div style="width: 100%;"></div>	Downloading jeepney-0.8.0-py3-none-any.whl (48 kB)
501	<div style="width: 100%;"></div>	48.4/48.4 KB 203.5
502	<div style="width: 100%;"></div>	Collecting greenlet!=0.4.17
503	<div style="width: 100%;"></div>	Downloading greenlet-2.0.2-cp39-cp39-manylinux_2_17_x86_64.ma
504	<div style="width: 100%;"></div>	610.9/610.9 KB 293.6
505	<div style="width: 100%;"></div>	Collecting backports.weakref
506	<div style="width: 100%;"></div>	Downloading backports.weakref-1.0.post1-py2.py3-none-any.whl
507	<div style="width: 100%;"></div>	Collecting pyproject_hooks

© Copyright. All rights reserved. Not to be reproduced without prior written consent.

Azure ML “MLflow”



train_model_node

Overview Parameters Outputs + logs Metrics Child jobs Logs Authoring

Refresh Create custom chart View as... Current view

Select metrics

Select to view as visualization or table of the data

Search

Select all

Metrics (5)

- training_mean_absolute_error
- training_mean_squared_error
- training_r2_score
- training_root_mean_squared_error
- training_score

1.7061e+4

training_score

0.9190283

```

def train_model(
    X_train: pd.DataFrame, y_train: pd.Series, random_state: int, model_params: dict
):
    """
    Train the model on the training data.
    """
    mlflow.sklearn.autolog(
        log_input_examples=True, log_model_signatures=True, log_models=True
    )
    model = RandomForestRegressor(random_state=random_state, **model_params)
    model.fit(X_train, y_train)
    return model
  
```

Microsoft Azure Machine Learning Studio

getindata.com > ml-ops-sandbox > Models

Model List

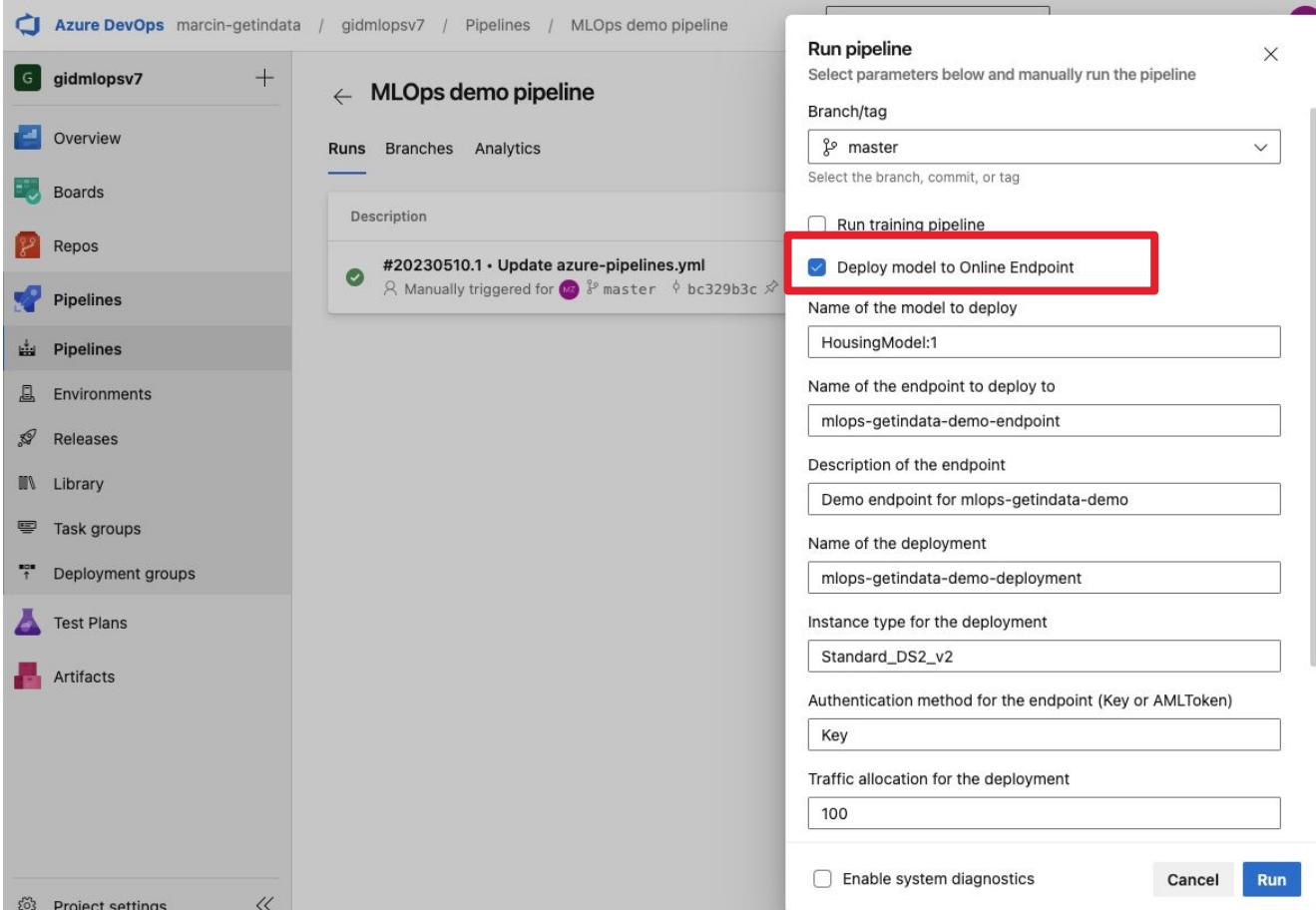
Register Refresh Delete Archive Deploy Compare (preview) View options

Show latest versions only

Search

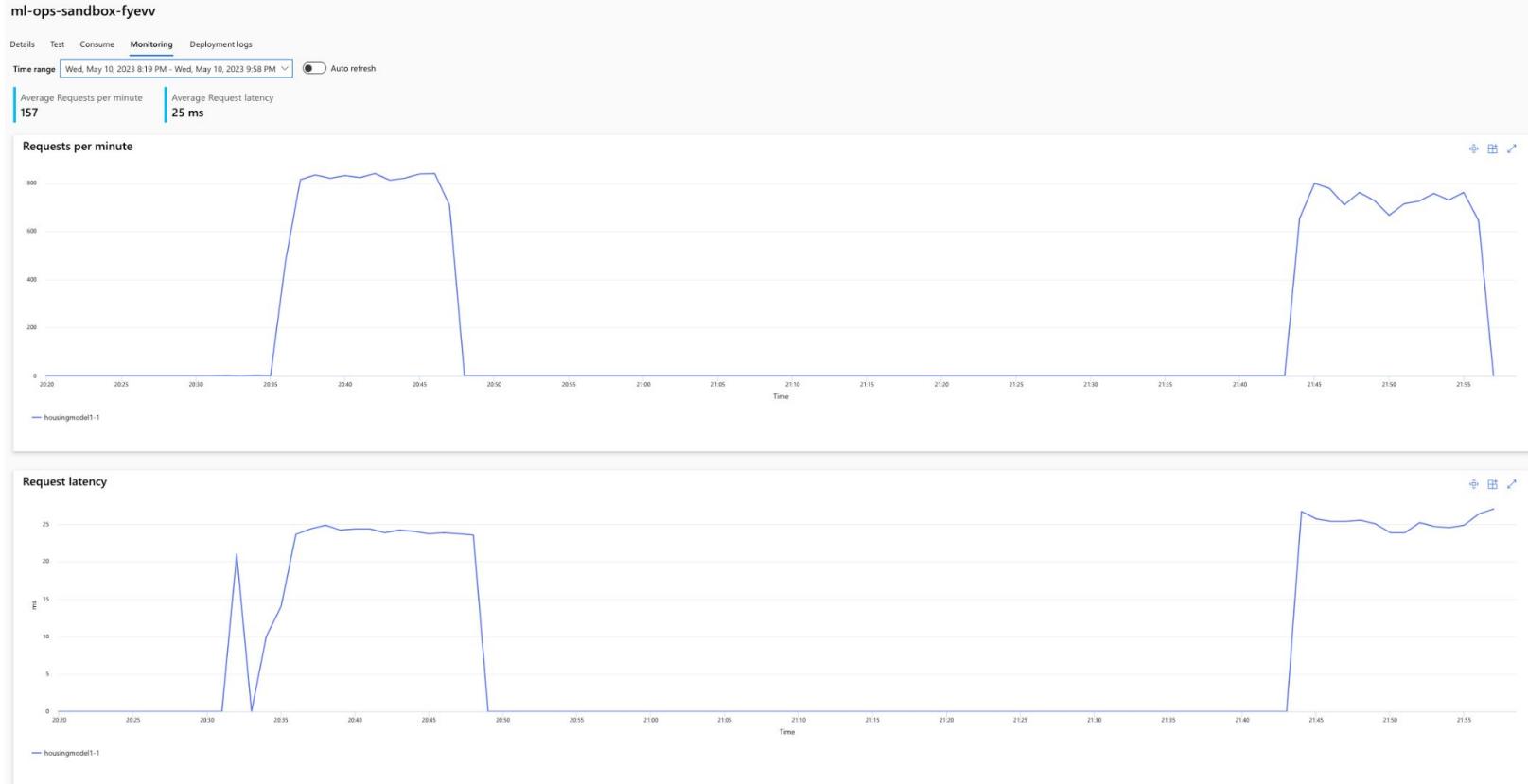
Name	Version	Type	Source	Experiment	Job (Run ID)
HousingModel1	1	MLFLOW	This workspace	xenia-demo	20577d3-a6ad-48bc-a631-869...
HbModel1	1	MLFLOW	This workspace		b8c7f290-b03c-4dfe-be4f-663d...
xgboost1	1	MLFLOW	This workspace		9e8119ff-9c18-4b66-9f71-48424...
TorchScriptModel1	2	MLFLOW	This workspace		2d216592-0400-40ab-9c27-bb...
TorchScriptModel1	1	MLFLOW	This workspace		fe3c369-66e9-4495-88bf-b87b...
PyTorch-Model1	1	MLFLOW	This workspace		d7ce69c-93cc-450d-a1d5-7a11...
asd1	1	MLFLOW	This workspace		01b942bf-adef-4234-adaa-3e7...
active_model	1	MLFLOW	This workspace		13f02d6b-c7f3-44d6-9904-0e7...

Model deployment - Azure DevOps



The screenshot shows the Azure DevOps Pipelines interface for the 'MLOps demo pipeline'. On the left, the sidebar lists various sections: Overview, Boards, Repos, Pipelines (selected), Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Artifacts. The main area displays the 'MLOps demo pipeline' run history, showing a single run triggered manually on the 'master' branch. A modal window titled 'Run pipeline' is open, prompting the user to select parameters for running the pipeline. The 'Branch/tag' dropdown is set to 'master'. Under the 'Run pipeline' section, there are two options: 'Run training pipeline' (unchecked) and 'Deploy model to Online Endpoint' (checked). The checked option is highlighted with a red border. The modal also contains fields for 'Name of the model to deploy' (set to 'HousingModel:1'), 'Name of the endpoint to deploy to' (set to 'mllops-getindata-demo-endpoint'), 'Description of the endpoint' (set to 'Demo endpoint for mllops-getindata-demo'), 'Name of the deployment' (set to 'mllops-getindata-demo-deployment'), 'Instance type for the deployment' (set to 'Standard_DS2_v2'), 'Authentication method for the endpoint (Key or AMLToken)' (set to 'Key'), 'Traffic allocation for the deployment' (set to '100'), and 'Enable system diagnostics' (unchecked). At the bottom right of the modal are 'Cancel' and 'Run' buttons.

Model Monitoring



3 Take-home messages

- Kedro is one of the best MLOps frameworks to make data scientists more **productive** out-of-the-box
- GetinData contributions to Kedro enable users to extend their Data Platforms with MLOps capabilities **seamlessly**
- **Kedro** together with **MLflow** and **Terraform** are the main building blocks of **our MLOps platform**



Part of Xebia

MLOps Evolution

Core TEAM

Michał Bryś

Machine Learning Architect



Marek Wiewiórka

Chief Data Architect

Marcin Zabłocki

MLOps Architect

Artur Dobrogowski

MLOps Engineer



Take DATA Pill and join DATA Matrix

- Best content from **Big Data, Cloud, AI/ML** and more
- simple, **condensed formula**
- sent **weekly** every Friday morning

Subscribe now:

