getindata
Part of Xebia

0 to ML❄ps platform with Snowflake

Snowflake Poznań Meetup, July 20th 2023

About us

getindata
Part of Xebia

Marek Wiewiórka

Chief Data Architect, GetInData | Part of Xebia | marek@getindata.com
Research Assistant at Warsaw University of Technology

- Soon to be PhD in bioinformatics
- An open source contributor to Snowflake Terraform Provider, SeQuiLa and Kedro
- Personally a keen long distance runner and gravel bike enthusiast

Marcin Zabłocki

MLOPS ARCHITECT at GetInData | Part of Xebia | marcin.zablocki@getindata.com

- DE → DS → MLE → MLOps
- member of Kedro TSC
- LEGO fan
- ML-Workout.pl

GetInData - At a Glance

getindata
Part of Xebia

- 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

dbt, mlflow, Fivetran, 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. **(Retail & consumer goods)**
   - **Snowflake migration from AWS to Azure**
   - Strong governance capabilities and **DataOps focus**
3. **Other**

ING, PLAY, iZettle, truecaller, Kcell, VEOLIA, Acast

Partner Certifications: 3 Core Sales Accreditations: 5 Sales Pro / 2 Tech Sales Pro

GetInData joins Xebia

getindata
Part of Xebia

Xebia
Xebia Academy
Xpirit
binx.io
GO DATA DRIVEN
Qxperts
OBLIVION CLOUD CONSULTANCY
g company
XCCELERATED
APPCINO
coMakeIT
PGS SOFTWARE
swiss Q
47 DEGREES
netlink digital solutions

getindata
Part of Xebia

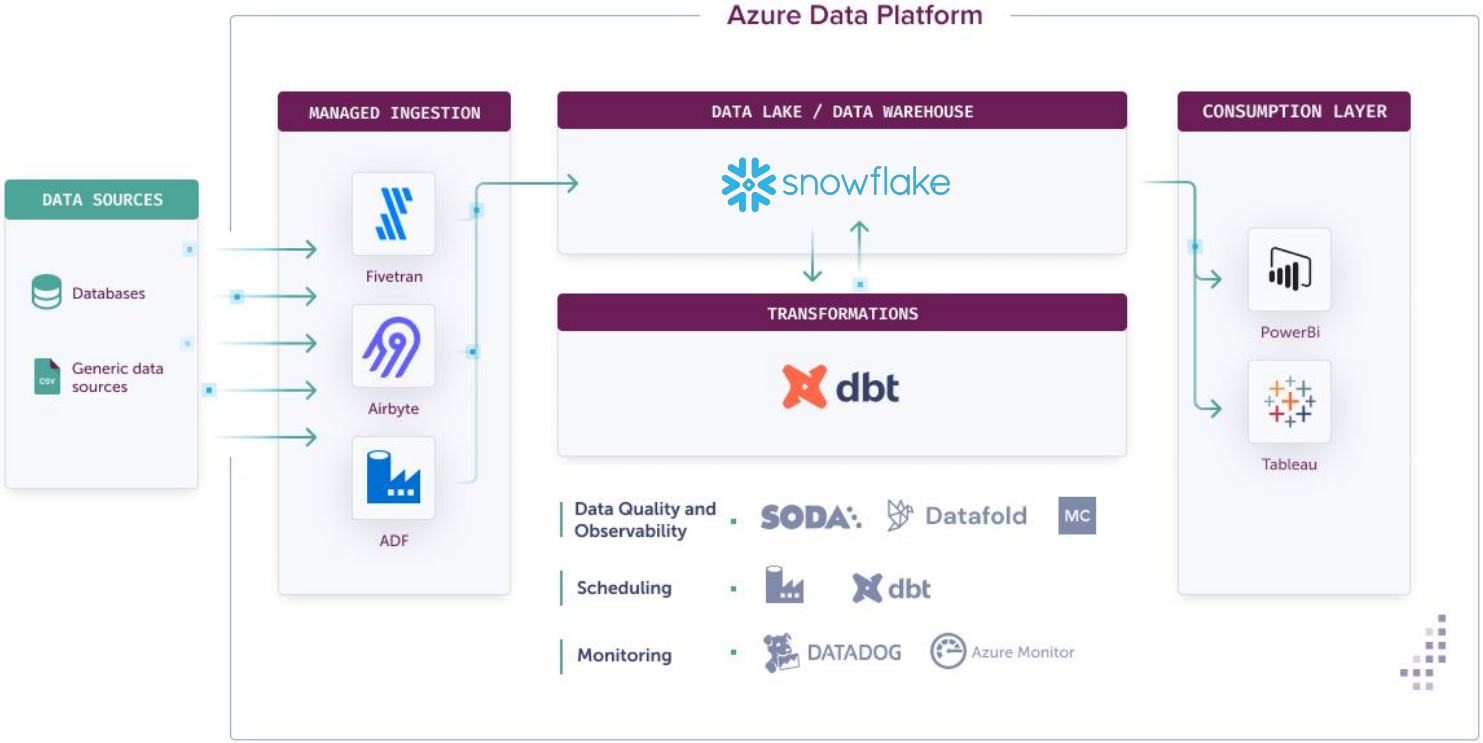
Our Customers

getindata
Part of Xebia

acast
Spotify
ING
SEAT
IAS Integral Ad Science
Kcell
PLAY
truecaller
iZettle
BNP PARIBAS
SYNERISE
FRESHMAIL
allegro
omio formerly go'euro
databricks
Msales
Nsure
PKO Bank Polski
LINK4
Kambi
CSG
KITOPI
NetWorkS
mamava
miinto
VEOLIA
NetSpring
JIMDO
sqooba
tvn
T-Mobile
datakin
ververica

From MDP[1] to MDP 2.0 (MLOps-enabled Data Platform)

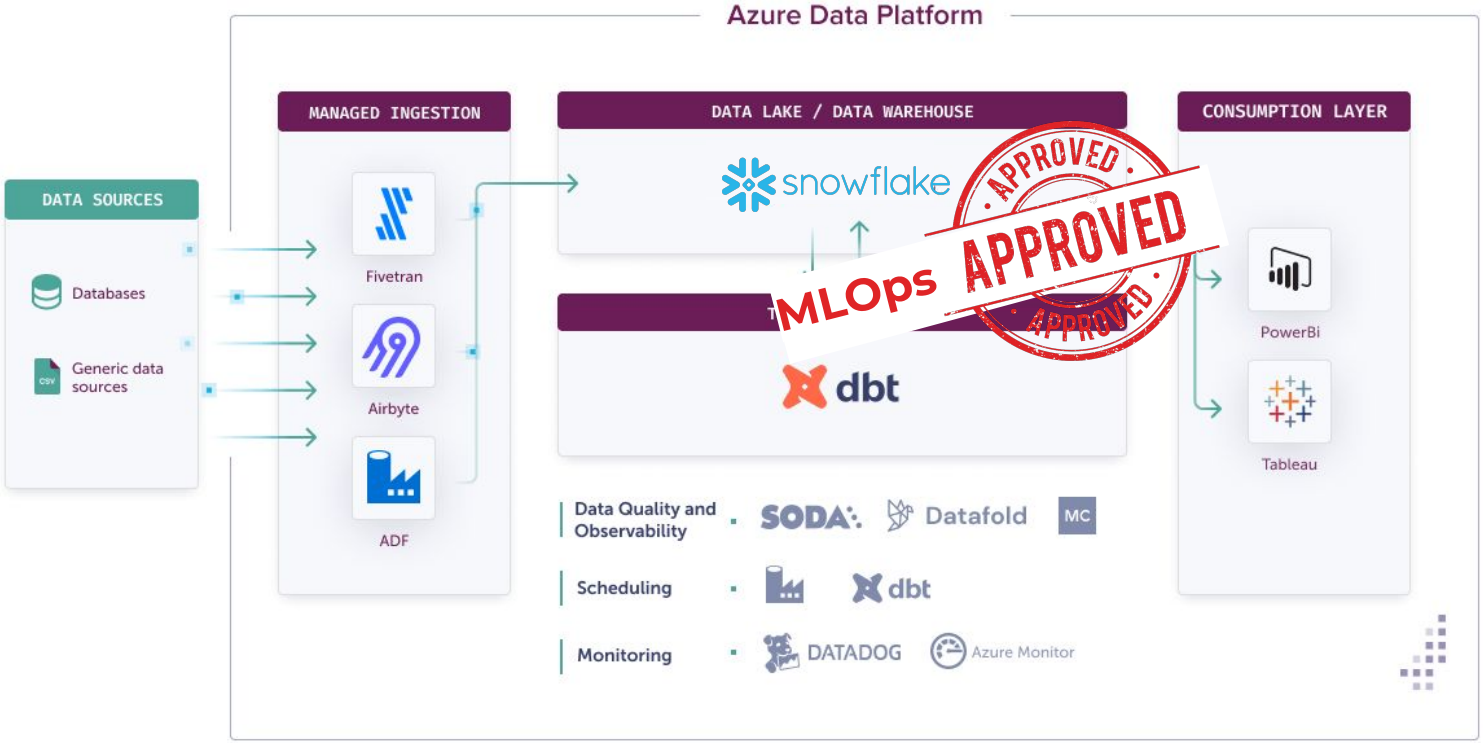
getindata
Part of Xebia



[1]MDP - Modern Data Platform

From MDP[1] to MDP 2.0 (MLOps-enabled Data Platform)

getindata
Part of Xebia



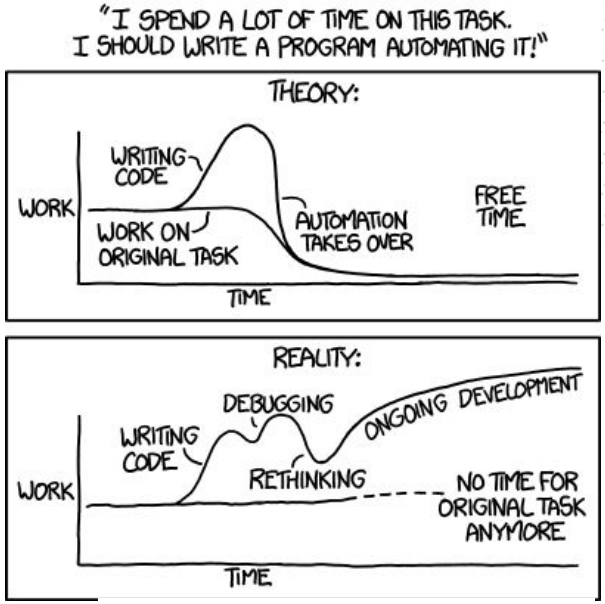
[1]MDP - Modern Data Platform

What MLOps is (*not only*) about ?

getindata
Part of Xebia

- 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**

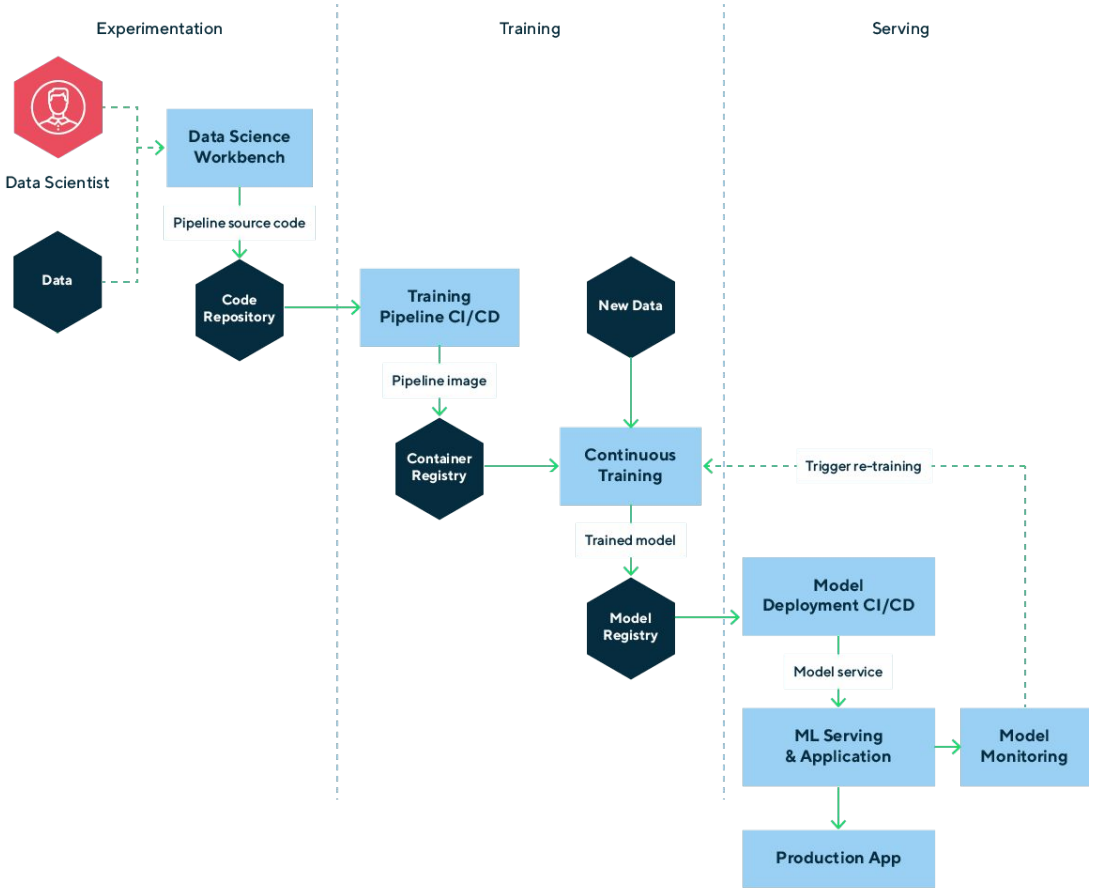


Source: xkcd by Randall Munroe. Automation takes a life of its own.

GID MLOps “Productivity Manifesto”

getindata
Part of Xebia

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



getindata
Part of Xebia

MLOps Platform

A process overview

ML projects in layers

Data Scientist

Experimentation + EDA

Machine Learning frameworks

Example technologies:

ML projects in layers

getindata
Part of Xebia

Data Scientist

Example technologies:

Experimentation + EDA

Machine Learning frameworks

Execution environment

Data

MLOps / ML Engineer

ML projects in layers

Data Scientist

Example technologies:

Experimentation + EDA

Machine Learning frameworks

?

Execution environment

Data

MLOps / ML Engineer

Building blocks of the GID MLOps

getindata
Part of Xebia

Data Scientist

Experimentation + EDA

Machine Learning frameworks

Portable MLOps framework

Experiment tracking and collaboration

IaC and automation

Kedro

mlflow

Terraform

Cloud Integrations (incl. GID Kedro plugins)

Execution environment

Data

MLOps / ML Engineer

Example technologies:

jupyter

learn

XGBoost

GID MLOps Platform

Building blocks of the GID MLOps

getindata
Part of Xebia

Data Scientist

Experimentation + EDA

Machine Learning frameworks

Portable MLOps framework

Experiment tracking and collaboration

IaC and automation

Kedro

mlflow

Terraform

Cloud Integrations (incl. GID Kedro plugins)

Execution environment

Data

MLOps / ML Engineer

Example technologies:

jupyter

learn

XGBoost

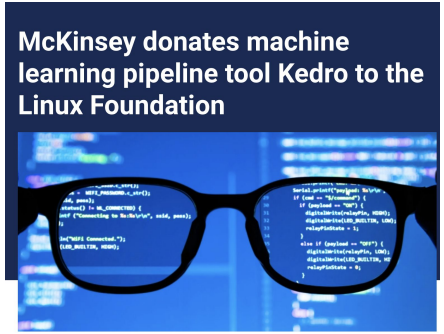
GID MLOps Platform

What is Kedro?

getindata
Part of Xebia

Kedro

= Software Engineering Principles + Data Science



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

What features does Kedro have? (Part 1)

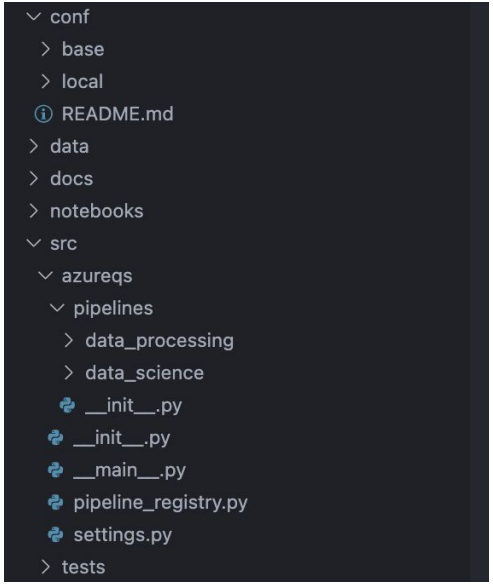
getindata
Part of Xebia

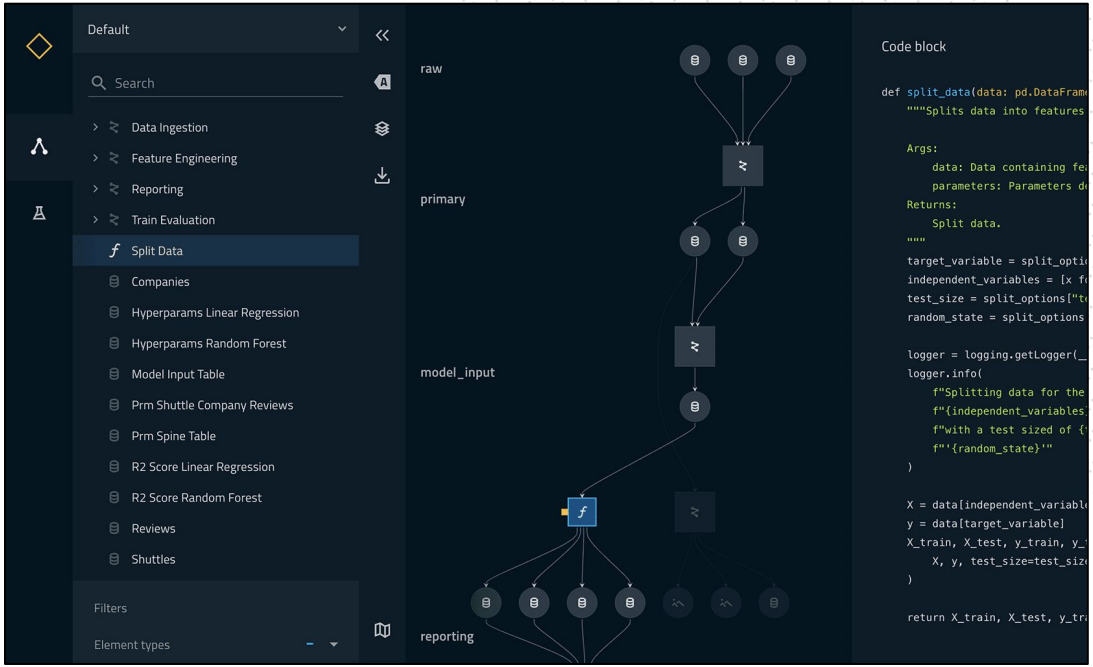
```
conf
  base
  local
  README.md
data
docs
notebooks
src
  azureqs
    pipelines
      data_processing
      data_science
      __init__.py
    __init__.py
    __main__.py
    pipeline_registry.py
    settings.py
  tests
```

Well defined project structure + project starters

What features does Kedro have? (Part 1)

getindata
Part of Xebia





Well defined project structure + project starters

Nodes & pipelines abstractions

Kedro pipeline - data engineering

getindata
Part of Xebia

```python
def create_pipeline(**kwargs) -> Pipeline:
    return pipeline(
        [
            node(
                func=preprocess_companies,
                inputs="companies",
                outputs="preprocessed_companies",
                name="preprocess_companies_node",
            ),
            node(
                func=preprocess_reviews,
                inputs="reviews",
                outputs="preprocessed_reviews",
                name="preprocess_reviews_node",
            ),
            node(
                func=create_model_input_table,
                inputs=["preprocessed_reviews", "preprocessed_companies", "ratings"],
                outputs="model_input_table",
                name="create_model_input_table_node",
            ),
        ]
    )
```

Kedro pipeline - data science

getindata
Part of Xebia

```python
def create_pipeline(**kwargs) -> Pipeline:
    return pipeline(
        [
            node(
                func=split_data,
                inputs=["model_input_table", "params:model_options"],
                outputs=["X_train", "X_test", "y_train", "y_test"],
                name="split_data_node",
            ),
            node(
                func=train_model,
                inputs=["X_train", "y_train"],
                outputs="regressor",
                name="train_model_node",
            ),
            node(
                func=evaluate_model,
                inputs=["regressor", "X_test", "y_test"],
                outputs=None,
                name="evaluate_model_node",
            ),
        ]
    )
```

Kedro node

getindata
Part of Xebia

```python
def create_pipeline(**kwargs) -> Pipeline:
    return pipeline(
        [
            node(
                func=preprocess_companies,
                inputs="companies",
                outputs="preprocessed_companies",
                name="preprocess_companies_node",
            ),
            node(
                func=preprocess_reviews,
                inputs="reviews",
                outputs="preprocessed_reviews",
                name="preprocess_reviews_node",
            ),
            node(
                func=create_model_input_table,
                inputs=["preprocessed_reviews", "
                outputs="model_input_table",
                name="create_model_input_table_node",
            ),
        ]
    )
```

```python
def create_model_input_table(
    reviews: pd.DataFrame, companies: pd.DataFrame, ratings: pd.DataFrame
) -> pd.DataFrame:
    """Combines all data to create a model input table.

    Args:
        reviews: Preprocessed data for reviews.
        companies: Preprocessed data for companies.
        ratings: Raw data for ratings.
    Returns:
        Model input table.

    """
    reviews_with_ratings = reviews.merge(ratings, left_on="id", right_on="rating_id")
    model_input_table = reviews_with_ratings.merge(
        companies, left_on="company_id", right_on="id"
    )
    model_input_table = model_input_table.dropna()
    return model_input_table
```

What about parameters?

```python
def create_pipeline(**kwargs) -> Pipeline:
    return pipeline(
        [
            node(
                func=split_data,
                inputs=["model_input_table", "params:model_options"],
                outputs=["X_train", "X_test", "y_train", "y_test"],
                name="split_data_node",
            ),
            node(
                func=train_model,
                inputs=["X_train", "y_train" "params:model_options"],
                outputs="regressor",
                name="train_model_node",
            ),
            node(
                func=evaluate_model,
                inputs=["regressor", "X_test", "y_test"],
                outputs=None,
                name="evaluate_model_node",
            ),
        ]
    )
```

What about parameters?

getindata
Part of Xebia

```python
def create_pipeline(**kwargs) -> Pipeline:
    return pipeline(
        [
            node(
                func=split_data,
                inputs=["model_input_table", "
                outputs=["X_train", "X_test",
                name="split_data_node",
            ),
            node(
                func=train_model,
                inputs=["X_train", "y_train"],
                outputs="regressor",
                name="train_model_node",
            ),
            node(
                func=evaluate_model,
                inputs=["regressor", "X_test",
                outputs=None,
                name="evaluate_model_node",
            ),
        ]
    )
```

- conf
  - base
    - parameters
      - data_processing.yml
      - data_science.yml
    - azureml.yml
    - catalog.yml
    - logging.yml
    - parameters.yml
  - local
- data
- docs

```yaml
model_options:
  test_size: 0.2
  random_state: 3
  features:
    - engines
    - passenger_capacity
    - crew
    - d_check_complete
    - moon_clearance_complete
    - iata_approved
    - company_rating
    - review_scores_rating
```

What about data?

getindata
Part of Xebia

```python
def create_pipeline(**kwargs) -> Pipeline:
    return pipeline(
        [
            node(
                func=preprocess_companies,
                inputs="companies",
                outputs="preprocessed_companies",
                name="preprocess_companies_node",
            ),
            node(
                func=preprocess_reviews,
                inputs="reviews",
                outputs="preprocessed_reviews",
                name="preprocess_reviews_node",
            ),
            node(
                func=create_model_input_table,
                inputs=["preprocessed_reviews", "preprocessed_companies", "ratings"],
                outputs="model_input_table",
                name="create_model_input_table_node",
            ),
        ]
    )
```

Kedro Data Catalog

```python
def create_pipeline(**kwargs) -> Pipeline:
    return pipeline(
        [
            node(
                func=preprocess_companies,
                inputs="companies",
                outputs="preprocessed_co
                name="preprocess_compani
            ),
            node(
                func=preprocess_reviews,
                inputs="reviews",
                outputs="preprocessed_re
                name="preprocess_reviews
            ),
            node(
                func=create_model_input_
                inputs=["preprocessed_reviews", "preprocessed_companies", "ratings"],
                outputs="model_input_table",
                name="create_model_input_table_node",
            ),
        ]
    )
```

- conf
  - base
    - parameters
    - azureml.yml
    - catalog.yml
    - logging.yml
    - parameters.yml
  - local
- data
- docs
- notebooks
- src

```yaml
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
```

What features does Kedro have? (Part 2)

getindata
Part of Xebia

Local catalog.yml

```yaml
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
```

Cloud catalog.yml

```yaml
companies:
  type: pandas.CSVDataSet
  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 / Environments

What features does Kedro have? (Part 2)

getindata
Part of Xebia

Local catalog.yml

```yaml
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
```

Cloud catalog.yml

```yaml
companies:
  type: pandas.CSVDataSet
  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 / Environments

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.

ML model?

getindata
Part of Xebia

```python
def create_pipeline(**kwargs) -> Pipeline:
    return pipeline(
        [
            node(
                func=split_data,
                inputs=["model_input_table", "params:model_options"],
                outputs=["X_train", "X_test", "y_train", "y_test"],
                name="split_data_node",
            ),
            node(
                func=train_model,
                inputs=["X_train", "y_train","params:model_options"],
                outputs="regressor",
                name="train_model_node",
            ),
            node(
                func=evaluate_model,
                inputs=["regressor", "X_test", "y_test"],
                outputs=None,
                name="evaluate_model_node",
            ),
        ]
    )
```

?

MLflow from Kedro

getindata
Part of Xebia

```python
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
```

Building blocks of the GID MLOps

getindata
Part of Xebia

Data Scientist

Experimentation + EDA

Machine Learning frameworks

Portable MLOps framework

Experiment tracking and collaboration

IaC and automation

Kedro

mlflow

Terraform

Cloud Integrations (incl. GID Kedro plugins)

Execution environment

Data

MLOps / ML Engineer

Example technologies:

jupyter

learn

XGBoost

GID MLOps Platform

MLflow UI

getindata
Part of Xebia

mlflow 2.2.0 Experiments Models

Experiments

Search Experiments

- Default
- demo-mlops
- snowflights

Default Provide Feedback

Experiment ID: 0 Artifact Location: mlflow-artifacts:/0

Description Edit

Table view | Chart view | metrics.rmse < 1 and params.model = "tree"

Time created: All time | State: Active

Run Name	Created	Duration
peaceful-bird-663	2 days ago	3.0min
entertaining-fowl-59	10 days ago	2.3min
whimsical-steed-397	11 days ago	2.2min
powerful-gull-761	11 days ago	2.2min
thoughtful-mink-405	11 days ago	2.7min

mlflow 2.2.0 Experiments Models

Default ›

peaceful-bird-663

Run ID: cfd3a4d6d3a94f7b886a1649898ca62d Date: 2023-06-23 17:08:15

Duration: 3.0min Status: FINISHED

Description Edit

Parameters (16)

Metrics (1)

Name	Value
r2	0.797

Tags

Artifacts

- model
  - MLmodel
  - conda.yaml
  - model.pkl
  - python_env.yaml
  - requirements.txt

mlflow 2.2.0 Experiments Models

Version 2

Registered At: 2023-06-13 10:53:12 Stage: Staging

Description Edit

Tags

Schema

Name	Type
Inputs (8)	
engines	double
passenger_capacity	long
crew	double
d_check_complete	boolean
moon_clearance_complete	boolean

```
conda: conda.yaml
virtualenv: python_env.yaml
loader_module: mlflow.sklearn
```

- Experiment tracking
- Model registry
- Model deployments (online and offline) with service plugins

Building blocks of the GID MLOps

getindata
Part of Xebia

Data Scientist

Experimentation + EDA

Machine Learning frameworks

Portable MLOps framework

Experiment tracking and collaboration

IaC and automation

Kedro

mlflow

Terraform

Cloud Integrations (incl. GID Kedro plugins)

Execution environment

Data

MLOps / ML Engineer

Example technologies:

jupyter

learn

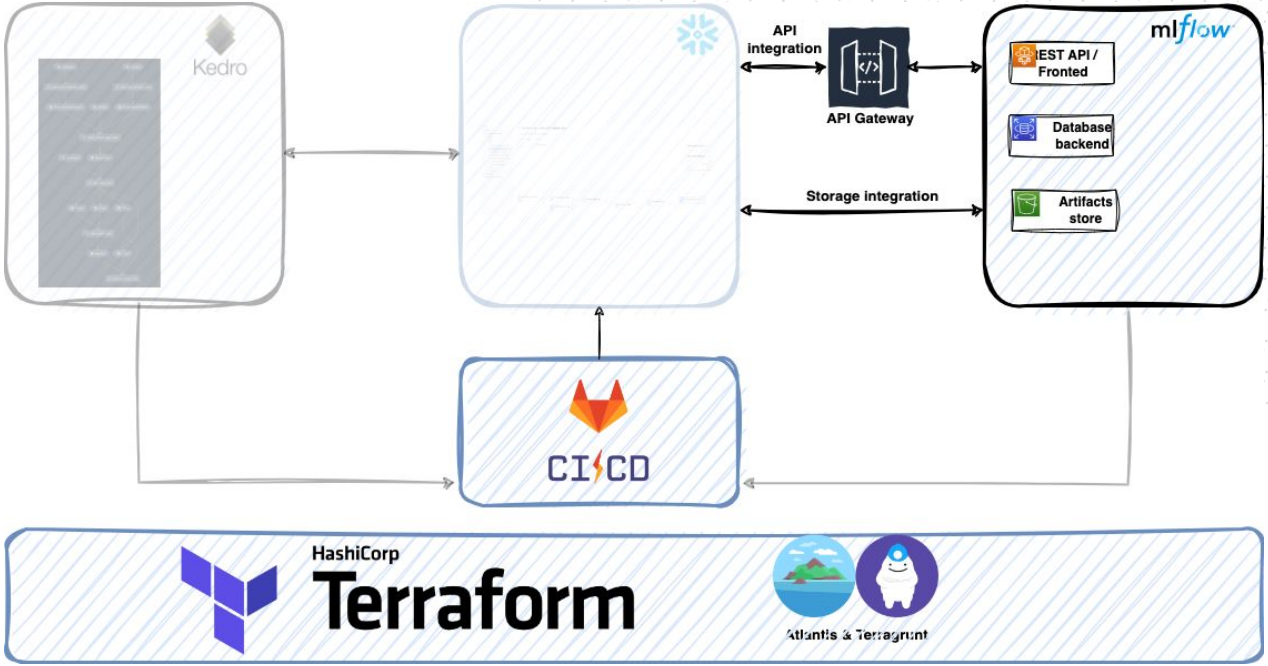
XGBoost

GID MLOps Platform

MLOps Platform provisioned

getindata
Part of Xebia

- Set of ***Terraform*** modules managed by ***Terragrunt***
- Both for Snowflake and specific cloud provider
- ***CI/CD*** templates
- Available for AWS, Azure and GCP clouds



Are we done yet?

getindata
Part of Xebia

Building blocks of the GID MLOps

getindata
Part of Xebia

Data Scientist

Experimentation + EDA

Machine Learning frameworks

Portable MLOps framework

Experiment tracking and collaboration

IaC and automation

Kedro

mlflow

Terraform

Cloud Integrations (incl. GID Kedro plugins)

Execution environment

Data

MLOps / ML Engineer

Example technologies:

jupyter

learn

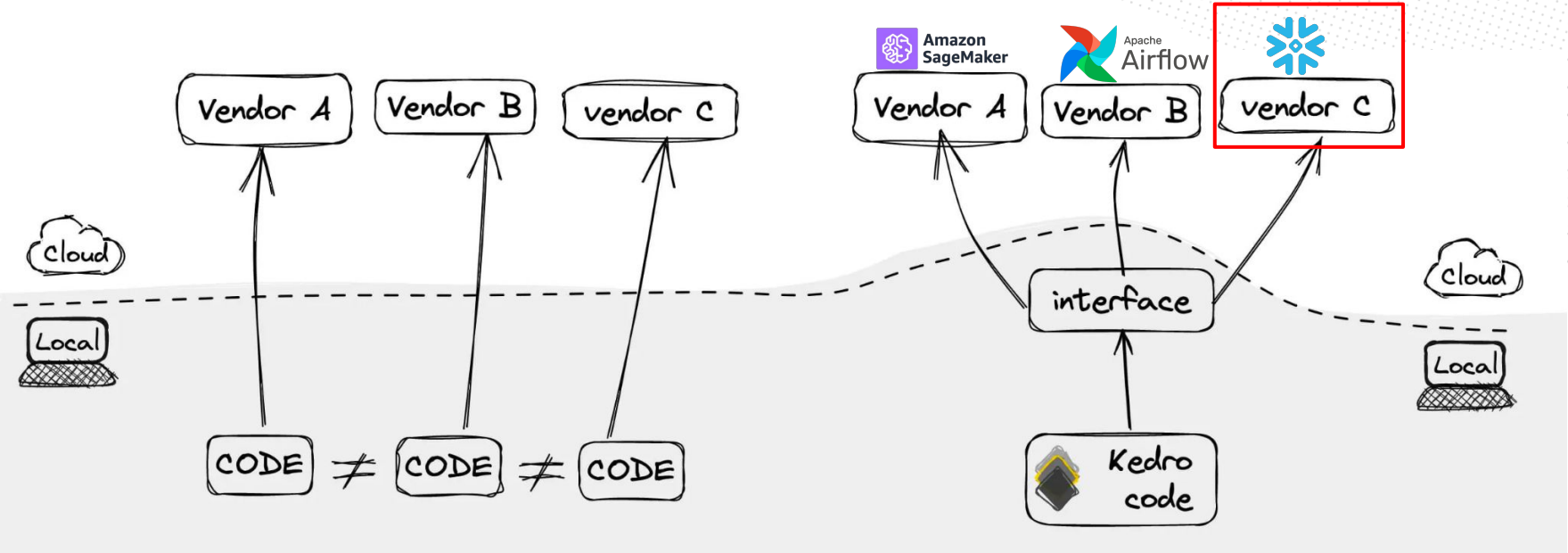
XGBoost

GID MLOps Platform

Why Kedro, again ?!

getindata
Part of Xebia

- 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

getindata
Part of Xebia

Kedro

Kedro Vertex AI (GCP)
github.com/getindata/kedro-vertexai

Kedro Sagemaker (AWS)
github.com/getindata/kedro-sagemaker

Kedro Airflow (Kubernetes)
github.com/getindata/kedro-airflow-k8s

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

Write once - run (almost) everywhere

getindata
Part of Xebia

Kedro

Kedro Vertex AI (GCP)
github.com/getindata/kedro-vertexai

Kedro Sagemaker (AWS)
github.com/getindata/kedro-sagemaker

Kedro Airflow (Kubernetes)
github.com/getindata/kedro-airflow-k8s

Kedro Kubeflow (Kubernetes)
github.com/getindata/kedro-kubeflow

Kedro AzureML (Azure)
github.com/getindata/kedro-azureml

Kedro Snowflake
github.com/getindata/kedro-snowflake

Read more about Snowflake on our blog: From 0 to MLOps with ❄ Snowflake Data Cloud in 3 steps with the Kedro-Snowflake plugin

Putting it all together...

getindata
Part of Xebia

Snowflake

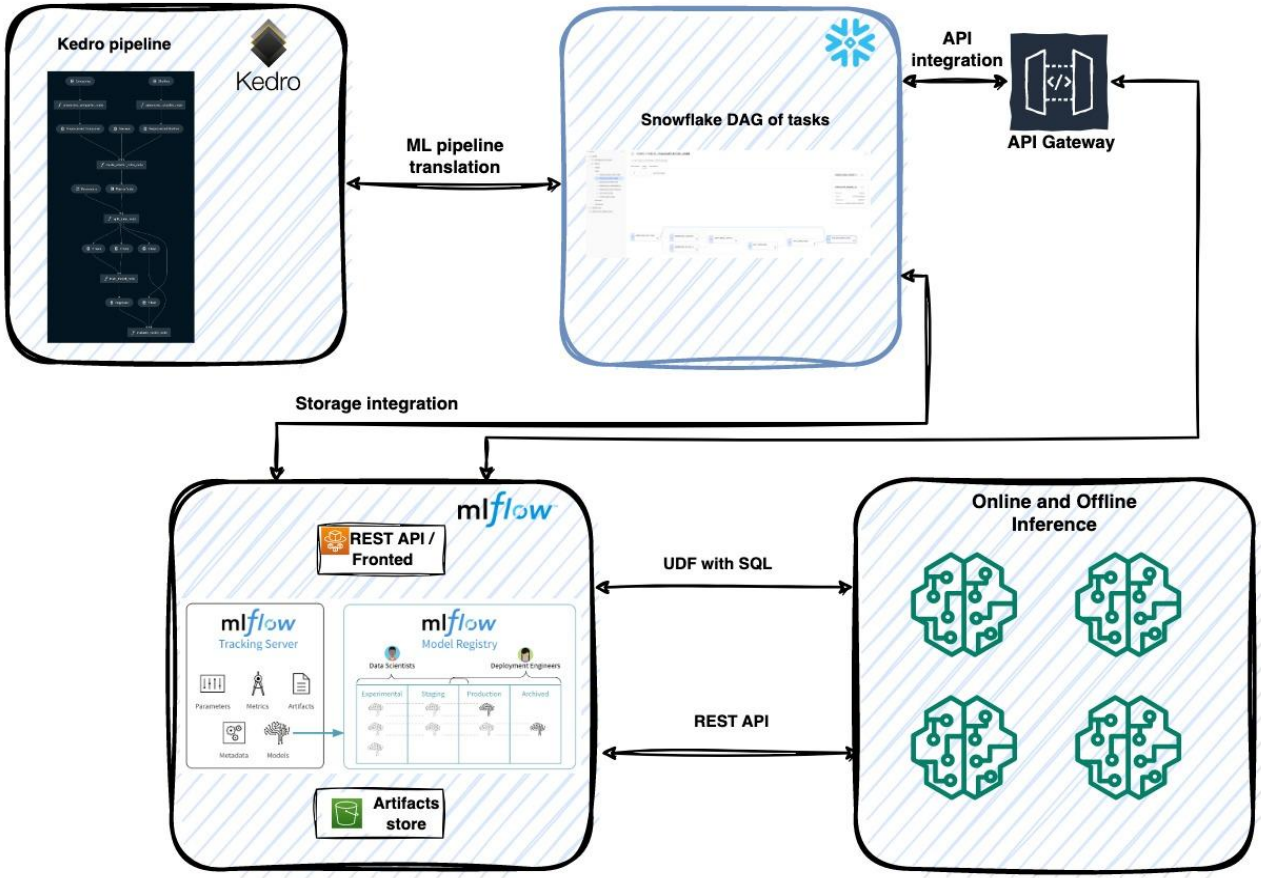
MLflow

Kedro

Putting it all together...

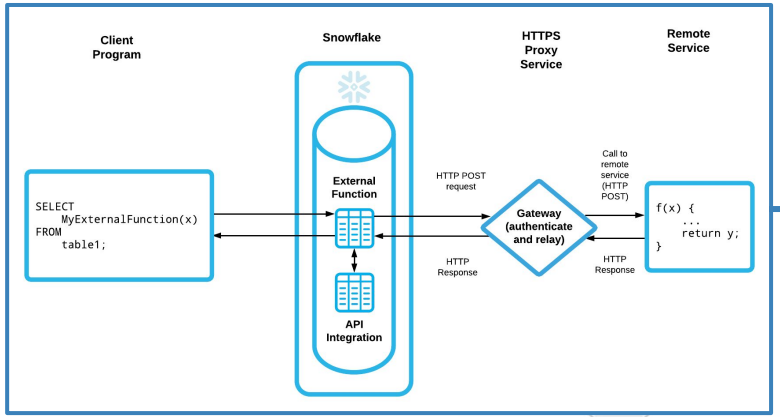
getindata
Part of Xebia

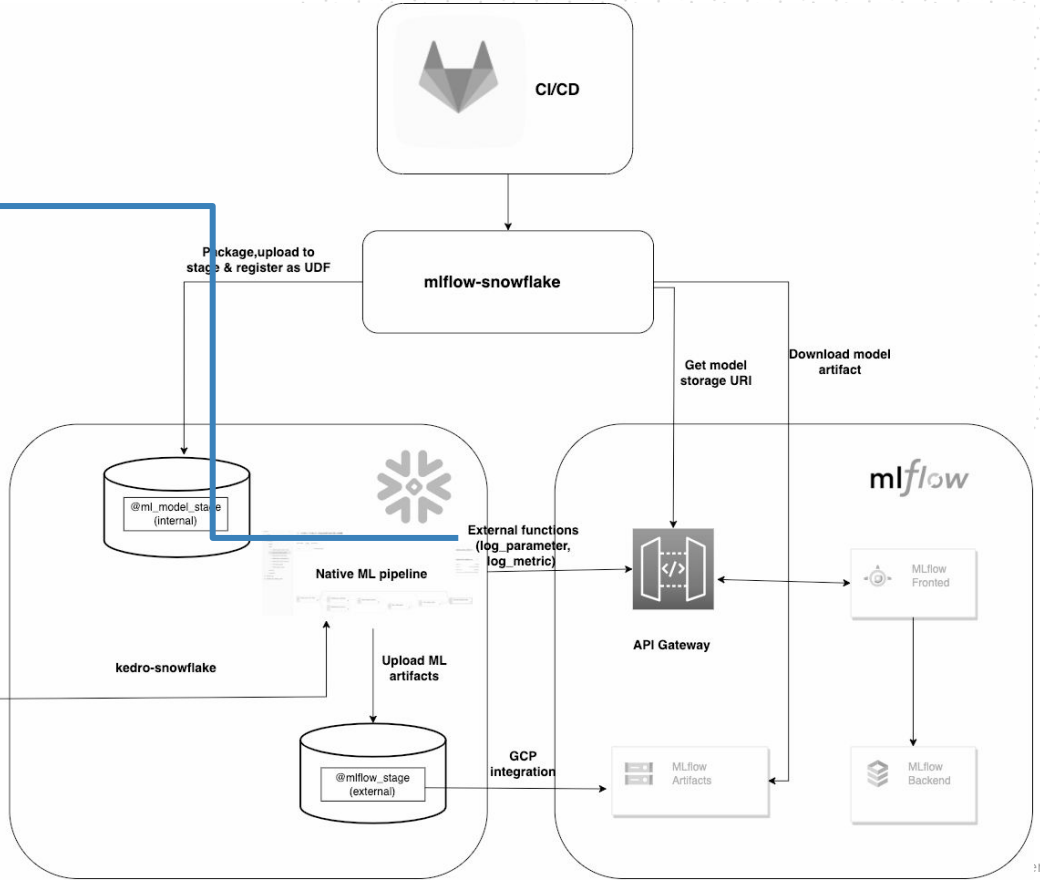
- ***Kedro***-Snowflake plugin
- Native ***Snowpark*** and Tasks integration
- ***MLflow*** with Cloud API Gateway
- ***MLflow Snowflake*** plugin for deployment as *UDF*
- MLflow **Sagemaker** - *REST*
- Set of ***Terraform*** of modules
- Built-in Kedro ***starter***



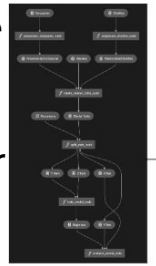
MLOps Platform - MLflow integration

getindata
Part of Xebia





Kedro



- Snowflake external functions
- Support for AWS, Azure and GCP Gateways
- Snowpark External Access in PuPr this year

External functions wrappers

getindata
Part of Xebia

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

```
resource "snowflake_external_function" "mlflow_run_create" {
  name     = upper("mlflow_run_create")
  database = var.database_name
  schema   = var.schema_name
  arg {
    name = "experiment_id"
    type = "varchar"
  }
  return_type               = "OBJECT"
  return_behavior           = "VOLATILE"
  api_integration           = snowflake_api_integration.mlflow_gcp.name
  request_translator = "${var.database_name}.${var.schema_name}.${snowflake_function.mlflow_run_create_req.name}"
  response_translator = "${var.database_name}.${var.schema_name}.${snowflake_function.mlflow_generic_res.name}"
  url_of_proxy_and_resource = "${var.api_gateway_url}/api/2.0/mlflow/runs/create"
}
```

```
resource "snowflake_function" "mlflow_run_create_req" {
  name     = upper("mlflow_run_create_req")
  database = snowflake_database.db.name
  schema   = snowflake_schema.schema.name
  arguments {
    name = "event"
    type = "OBJECT"
  }
  comment             = "Request translator for MLflow create run"
  return_type         = "OBJECT"
  language            = "javascript"
  statement           = <<EOH
      let exeprimentId = EVENT.body.data[0][1]
      let timestamp = new Date().getTime();
      return { "body": { "experiment_id": exeprimentId, start_time: timestamp }}
  EOH
}
```

External functions mappings

getindata
Part of Xebia

- Functions mappings
- Kedro hooks

```
{% if cookiecutter.enable_mlflow_integration|lower != "false" %}
    @hook_impl
    def before_node_run(self, node: Node) -> None:
        if node.name == "export_data_to_snowflake_node":
            mlflow_helpers.log_params(self.run_params)

    @hook_impl
    def after_node_run(self, node: Node) -> None:
        if node.name == "evaluate_model_node":
            mlflow_helpers.run_update("FINISHED")
{% endif %}
```

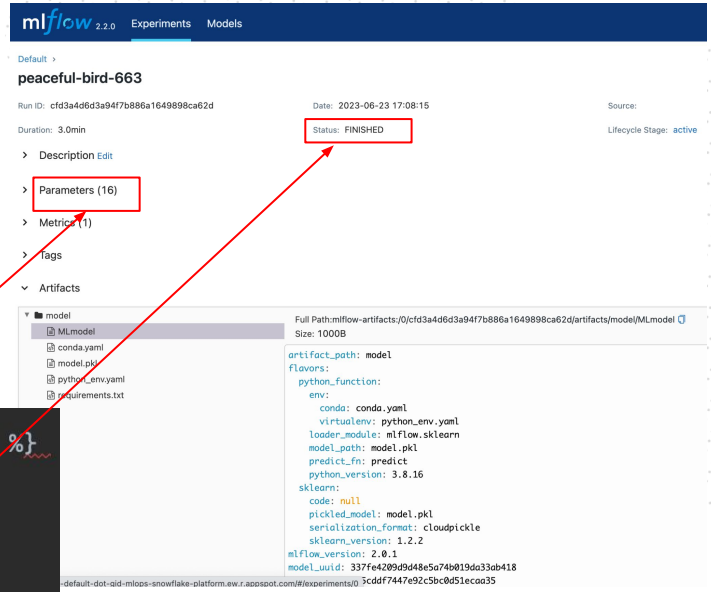
```
{%- if cookiecutter.enable_mlflow_integration|lower != "fa
# EXPERIMENTAL: Either MLflow experiment name to enable ML
# or leave empty
mlflow:
    experiment_name: Default
    stage: "@MLFLOW_STAGE"
    # Snowflake external functions needed for calling MLflow instance
    functions:
      experiment_get_by_name: {{ cookiecutter.snowflake_database | lower }}.{{ cookiecutter.snowflake_schema | lower }}.mlflow_experiment_get_by_name
      run_create: {{ cookiecutter.snowflake_database | lower }}.{{ cookiecutter.snowflake_schema | lower }}.mlflow_run_create
      run_update: {{ cookiecutter.snowflake_database | lower }}.{{ cookiecutter.snowflake_schema | lower }}.mlflow_run_update
      run_log_metric: {{ cookiecutter.snowflake_database | lower }}.{{ cookiecutter.snowflake_schema | lower }}.mlflow_run_log_metric
      run_log_parameter: {{ cookiecutter.snowflake_database | lower }}.{{ cookiecutter.snowflake_schema | lower }}.mlflow_run_log_parameter
{% else %}
```

External functions mappings

getindata
Part of Xebia

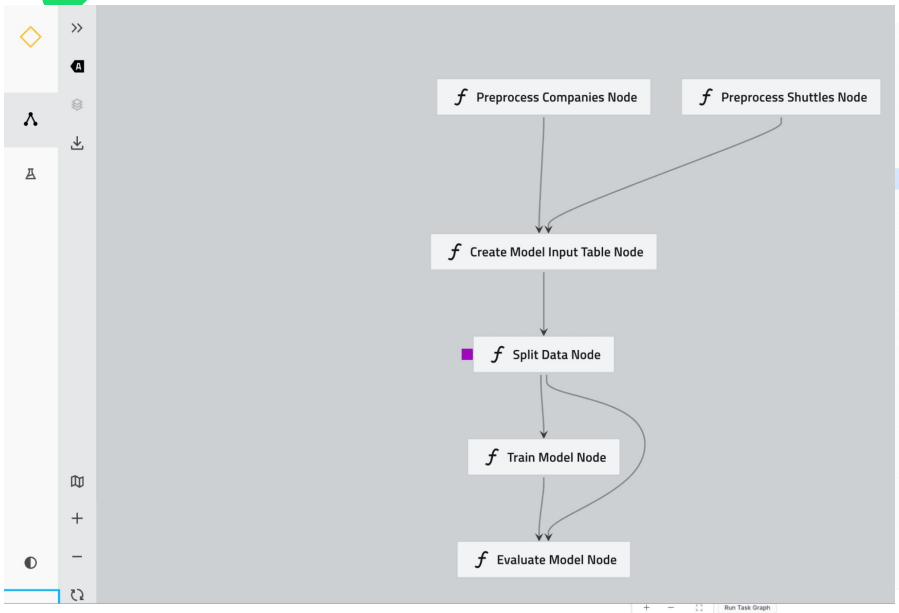
- Functions mappings
- Kedro hooks

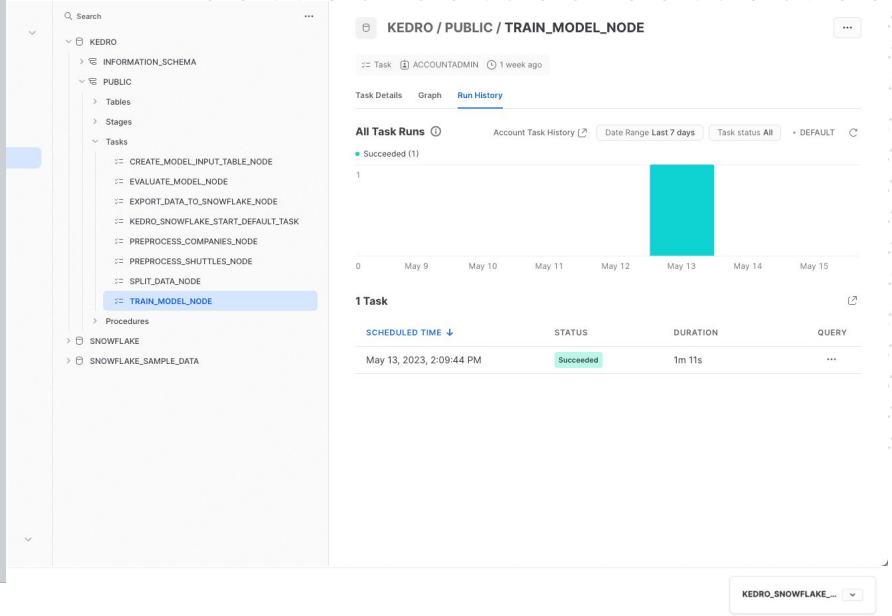
```
21  {% if cookiecutter.enable_mlflow_integration|lower != "false" %}
22      @hook_impl
23      def before_node_run(self, node: Node) -> None:
24          if node.name == "export_data_to_snowflake_node":
25              mlflow_helpers.log_params(self.run_params)
26
27      @hook_impl
28      def after_node_run(self, node: Node) -> None:
29          if node.name == "evaluate_model_node":
30              mlflow_helpers.run_update("FINISHED")
31  {% endif %}
```



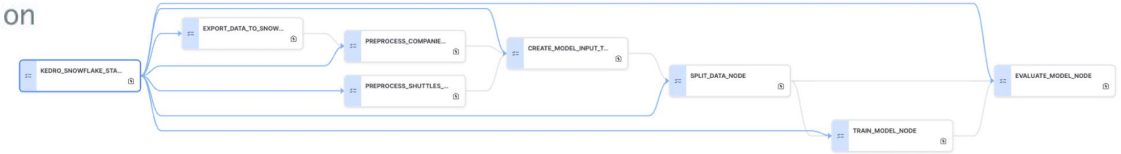
MLOps Platform for Snowflake

getindata
Part of Xebia






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



MLOps Platform for Snowflake

getindata
Part of Xebia

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

```yaml
companies_snowflake:
  type: kedro_datasets.snowflake.SnowparkTableDataSet
  table_name: companies_snowflights_starter
  credentials: snowflake
  save_args:
    mode: overwrite


preprocessed_shuttles:
  type: kedro_snowflake.datasets.native.SnowflakeStageFileDataSet
  stage: "@KEDRO_SNOWFLAKE_TEMP_DATA_STAGE" # <-- Snowflake stage to store data in
  filepath: data/02_intermediate/preprocessed_shuttles.csv # <-- file path within the stage
  credentials: snowflake # <-- credentials to connect to Snowflake (the same as for SnowparkTableDataSet)
  dataset: # <-- dataset key defines the dataset type to use
    type: pandas.CSVDataSet # <-- specify any params for the nested dataset here
```

Alternative approaches

Write once - run (almost) everywhere

getindata
Part of Xebia

Kedro

Kedro Vertex AI (GCP)
github.com/getindata/kedro-vertexai

Kedro Sagemaker (AWS)
github.com/getindata/kedro-sagemaker

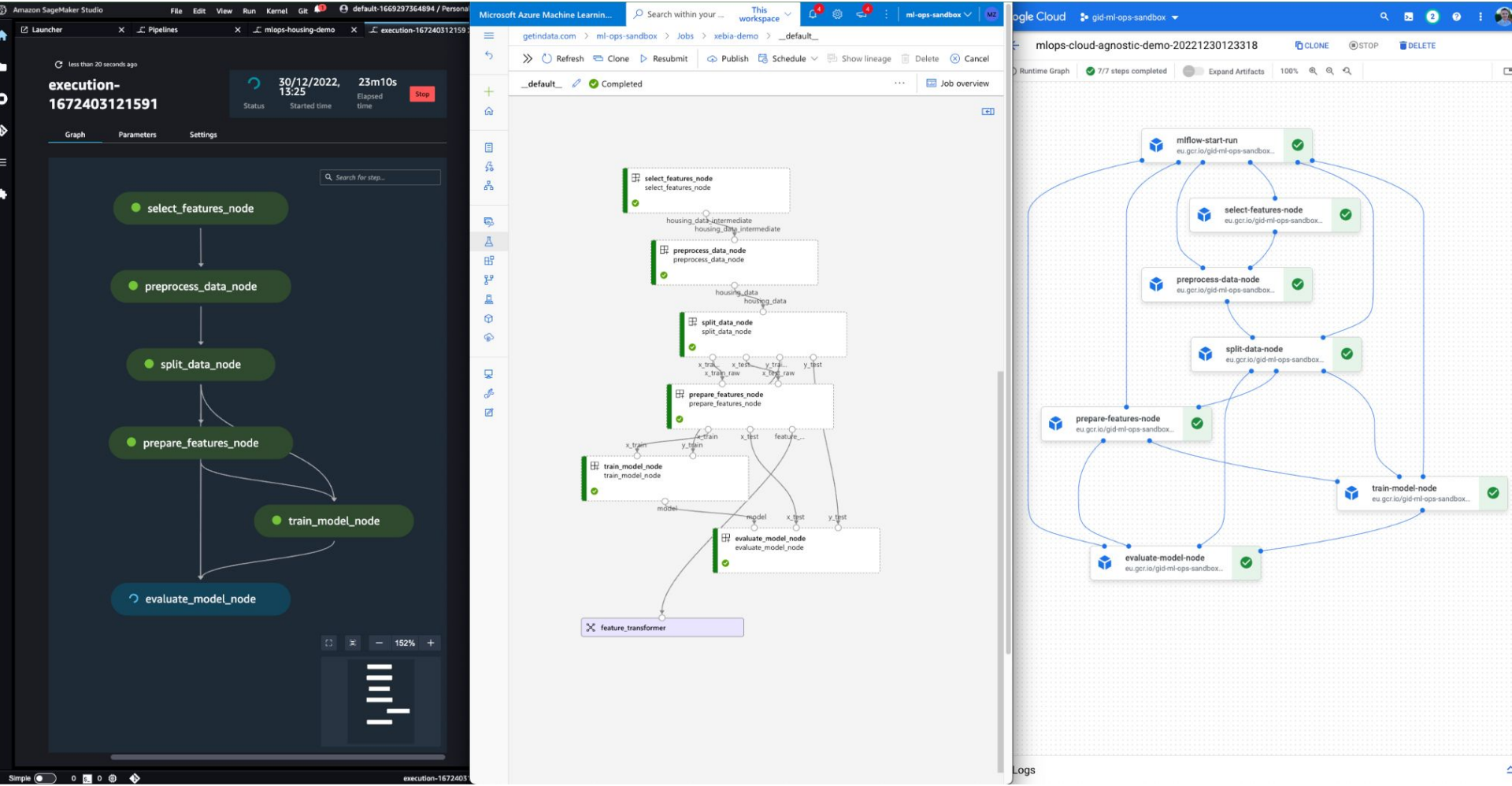
Kedro Airflow (Kubernetes)
github.com/getindata/kedro-airflow-k8s

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


Amazon SageMaker Studio
File Edit View Run Kernel Git
Launcher
Pipelines
mlops-housing-demo
execution-1672403121591
less than 20 seconds ago
execution-1672403121591
30/12/2022, 13:25
Status
Started time
23m10s
Elapsed time
Stop
Graph
Parameters
Settings
Search for step...
select_features_node
preprocess_data_node
split_data_node
prepare_features_node
train_model_node
evaluate_model_node
152%
Simple
Microsoft Azure Machine Learnin...
Search within your ...
This workspace
ml-ops-sandbox
getindata.com > ml-ops-sandbox > Jobs > xebia-demo > __default__
Refresh
Clone
Resubmit
Publish
Schedule
Show lineage
Delete
Cancel
__default__
Completed
Job overview
select_features_node
select_features_node
housing_data_intermediate
housing_data_intermediate
preprocess_data_node
preprocess_data_node
housing_data
housing_data
split_data_node
split_data_node
x_train_raw
x_test_raw
y_test
prepare_features_node
prepare_features_node
x_train
y_train
x_test
feature_...
train_model_node
train_model_node
model
model
x_test
y_test
evaluate_model_node
evaluate_model_node
feature_transformer
Google Cloud
gid-ml-ops-sandbox
mlops-cloud-agnostic-demo-20221230123318
CLONE
STOP
DELETE
Runtime Graph
7/7 steps completed
Expand Artifacts
100%
mlflow-start-run
eu.gcr.io/gid-ml-ops-sandbox...
select-features-node
eu.gcr.io/gid-ml-ops-sandbox...
preprocess-data-node
eu.gcr.io/gid-ml-ops-sandbox...
split-data-node
eu.gcr.io/gid-ml-ops-sandbox...
prepare-features-node
eu.gcr.io/gid-ml-ops-sandbox...
train-model-node
eu.gcr.io/gid-ml-ops-sandbox...
evaluate-model-node
eu.gcr.io/gid-ml-ops-sandbox...
Logs

MLOps orchestration tools in perspective

getindata
Part of Xebia

Kedro-Snowflake vs.

- simpler security setup
- fewer dependencies on external services
- substantially less data transfers
- a unified data and machine learning platform

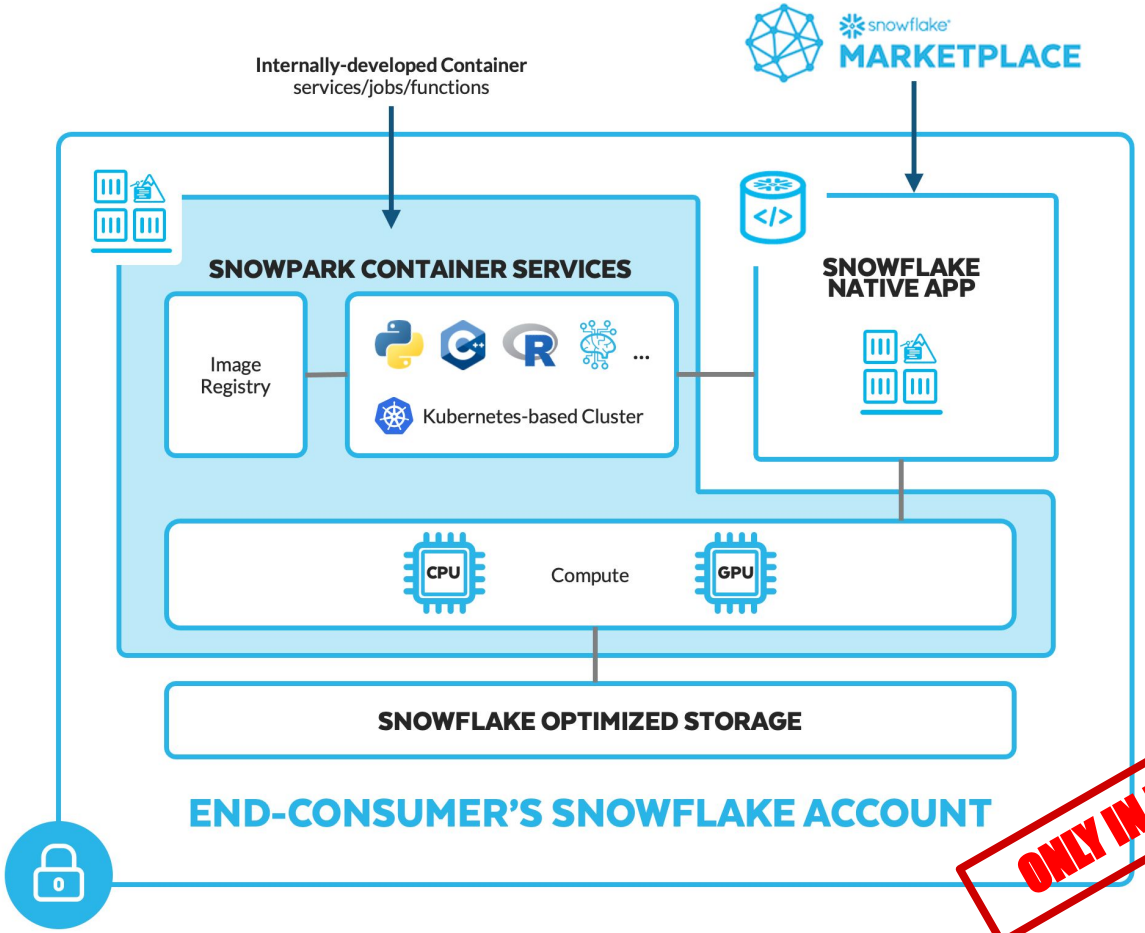
	Airflow	SageMaker/AzureML/VertexAI	Snowflake/Snowpark
Orchestration			
Native data processing			
Docker support			
Native ML capabilities			
Model deployment support (serving)			
Maintenance	High	Low (serverless)	Low / Medium
Extensibility / Customizability	High	Low	Medium
Performance	Depends on setup	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

Container services

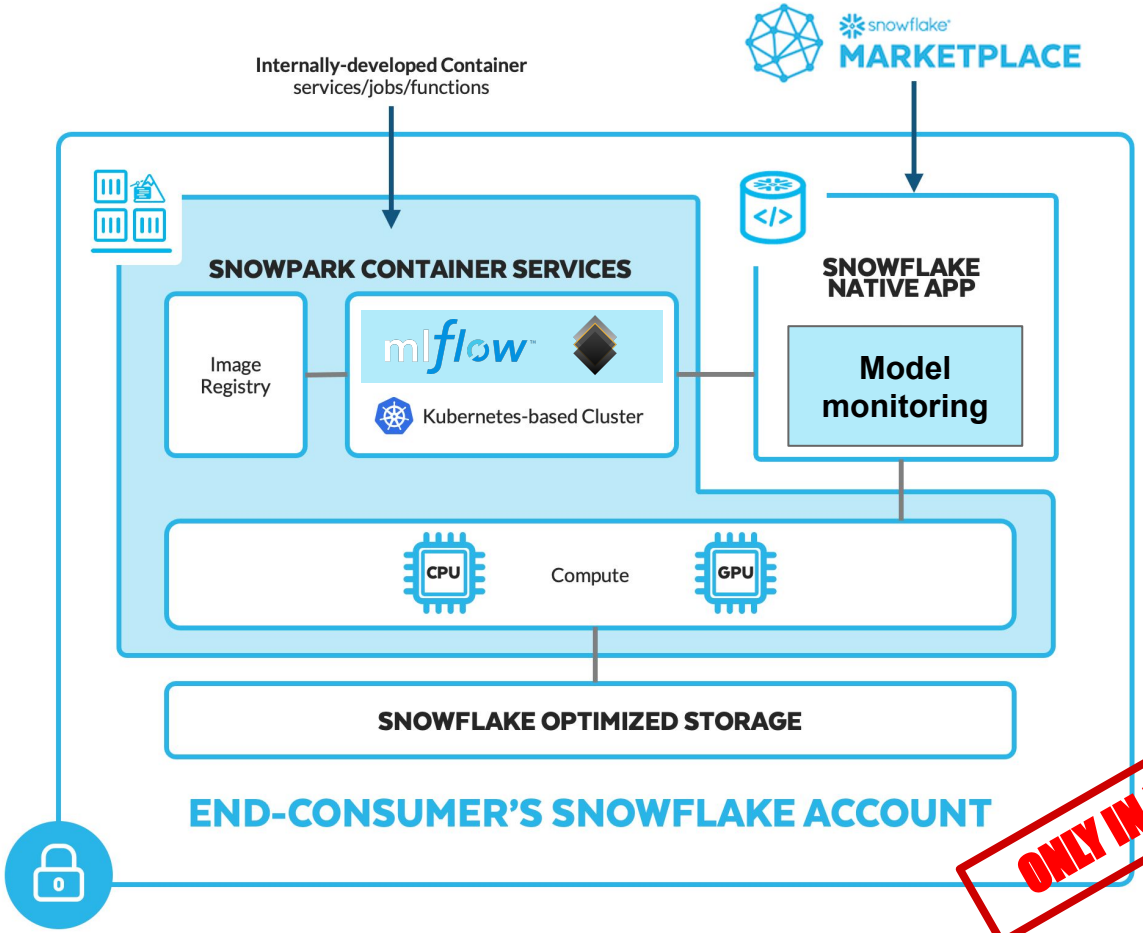
getindata
Part of Xebia



ONLY IN PRIVATE PREVIEW!

Container services

getindata
Part of Xebia



ONLY IN PRIVATE PREVIEW!

getindata

Demo

3 Take-home messages

getindata
Part of Xebia

- 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 Snowflake Data Cloud with MLOps capabilities ***seamlessly***

- Kedro together with MLflow and Terraform are the main building blocks of ***our Snowflake MLOps platform***

References

getindata
Part of Xebia

- github.com/getindata/kedro-snowflake
- github.com/Snowflake-Labs/mlflow-snowflake

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

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

- Running Kedro... everywhere? Machine Learning Pipelines on Kubeflow, Vertex AI, Azure and Airflow

Try it yourself!

1. Install the plugin

```
pip install "kedro-snowflake>=0.1.0"
```

2. Create new project with our Kedro starter ❄️ *Snowflights* 🚀:

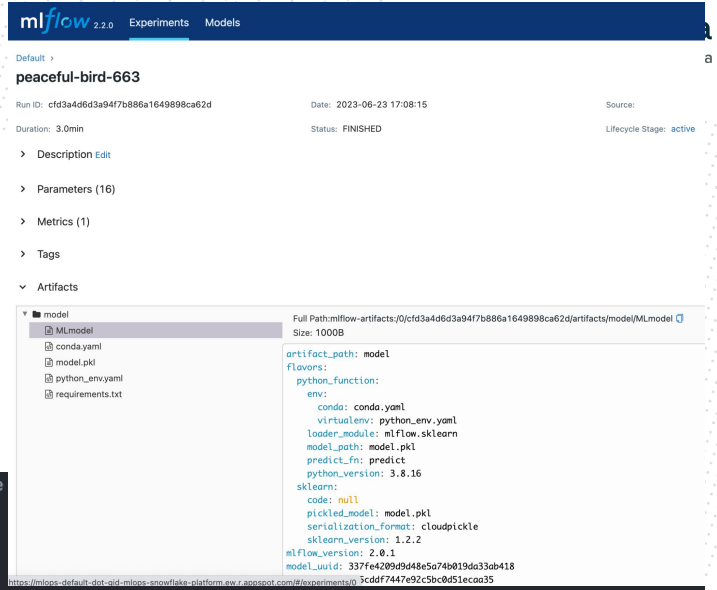
```
kedro new --starter=snowflights --checkout=master
```

▶ And answer the interactive prompts ⬇️ (click to expand)

3. Run the project

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

```
(kedro-snow-tests-py3.8) (base) ➜  snowflights kedro snowflake
```



getindata
Part of Xebia

MLOpsLab

Core TEAM

Marek Wiewiórka
Chief Data Architect at Getindata

marek@getindata.com

Marcin Zabłocki
MLOps Architect at Getindata

marcin.zablocki@getindata.com

Michał Bryś
ML Architect at Getindata

michal.brys@getindata.com

Take DATA Pill and join DATA Matrix

getindata
Part of Xebia

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

Subscribe now: