MLOps: Data Science End-to-End

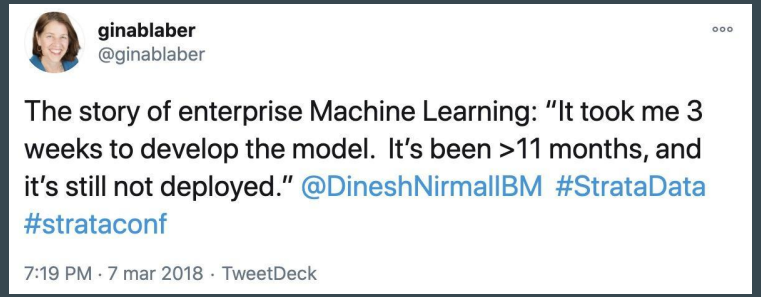
Amadeusz Lisiecki, MLOps Engineer
Roche Contractor
Hackerspace Pomorze - pomorze.hackerspace.pl

Content

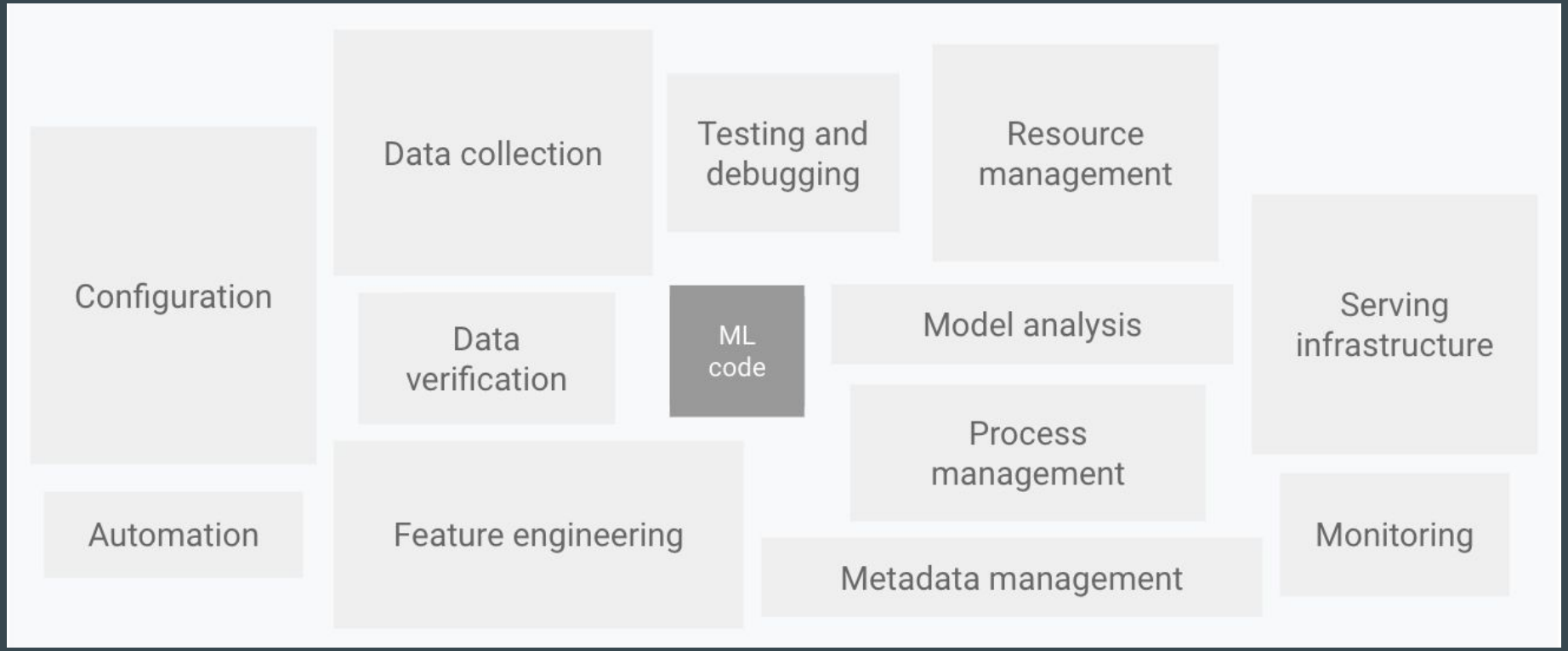
1. Problems around ML
2. MLOps
3. ML Pipelines
4. ML Pipeline + CI/CD
5. Kedro
6. Kubeflow
7. Links

Problems around ML

1. Reproducibility of models and predictions
2. Experiments logging
3. Collaboration
4. Automated deployment with scalability
5. Model monitoring
6. Retraining and improvements
7. Good practices, guidelines, frameworks, ...



Problems around ML

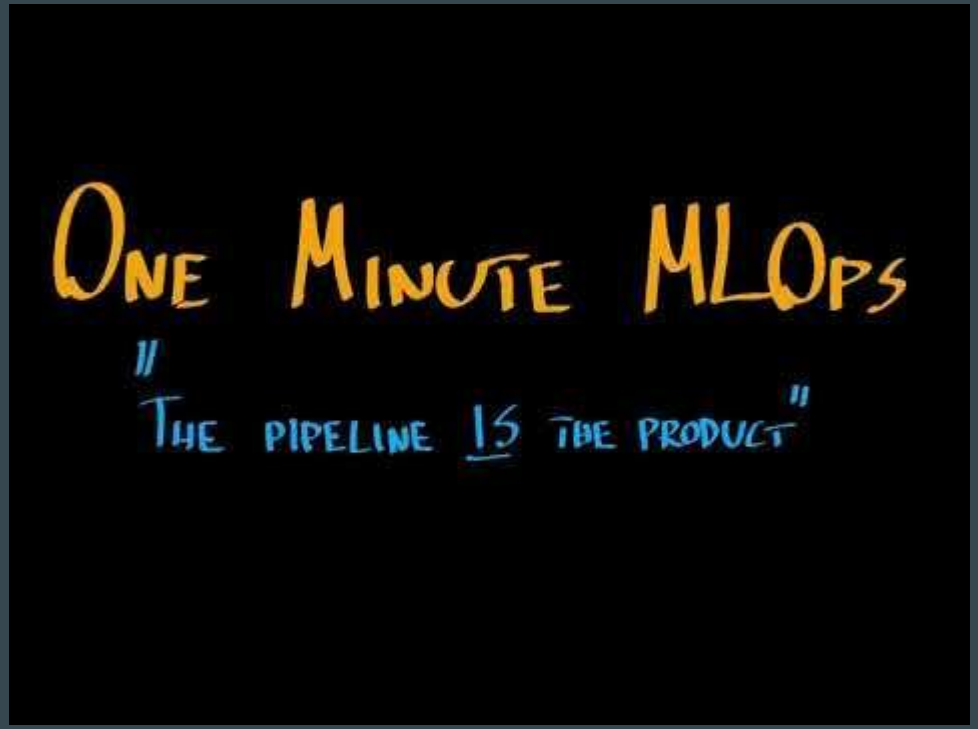


1. *https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf*
2. *https://cloud.google.com/solutions/machine-learning/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning*

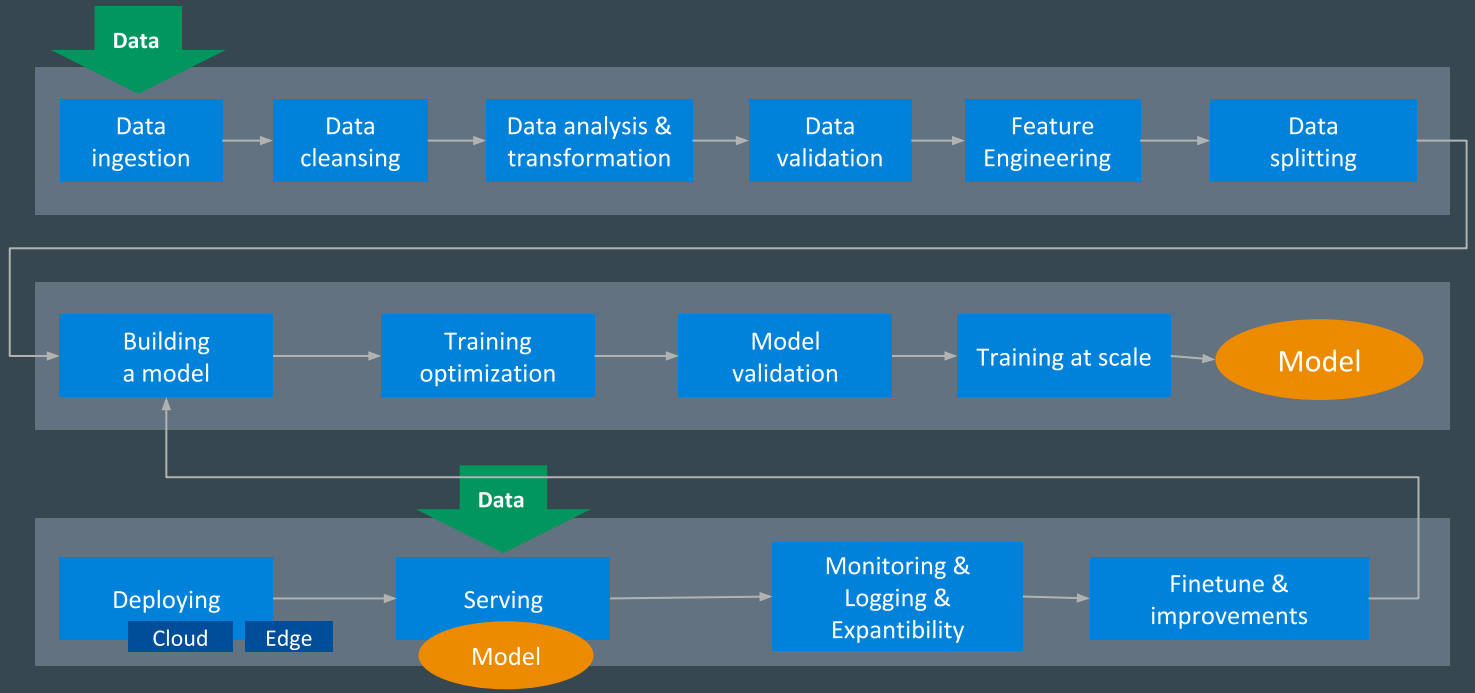
MLOps

1. Closing the gap between Data Scientists and IT to improve the quality and speed of ML development lifecycle.
2. Set of tools and practices around ML development and operations.

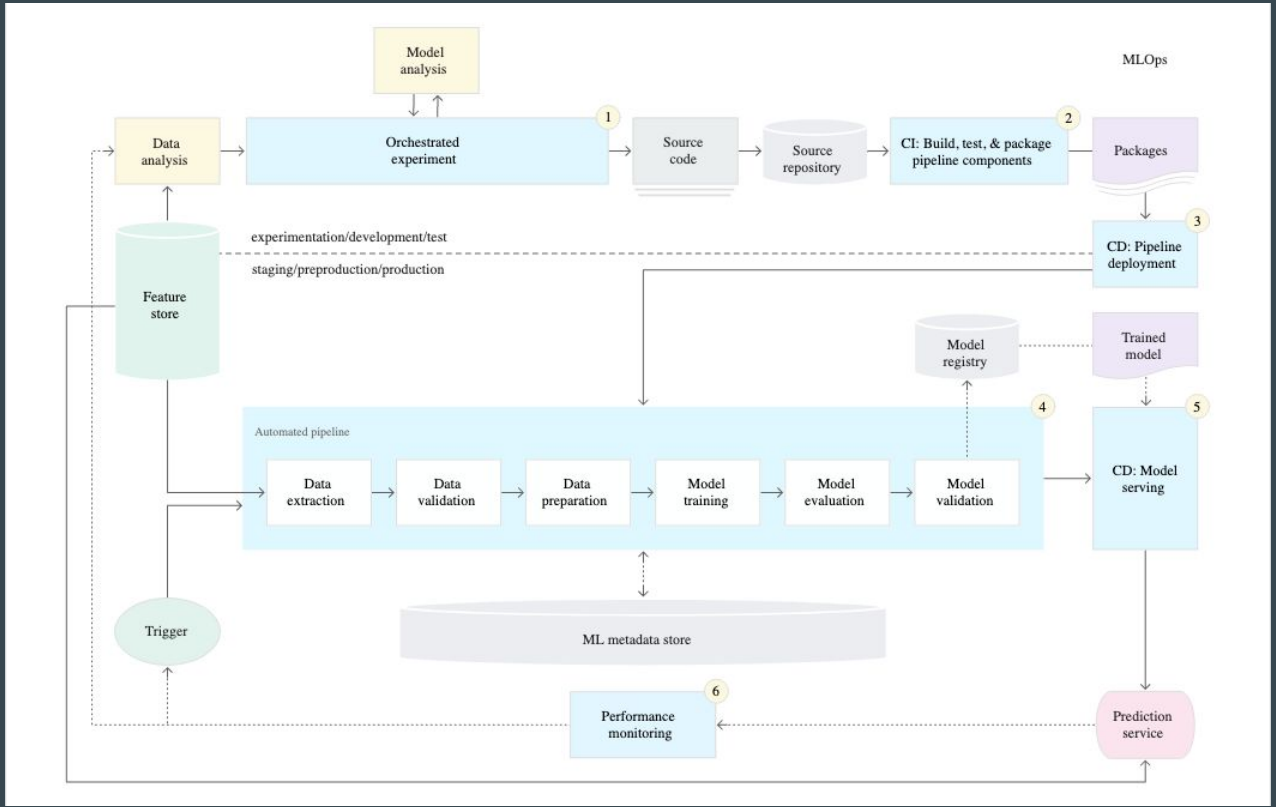
ML Pipeline



ML Pipeline


Data prep
Data
Data ingestion
Data cleansing
Data analysis & transformation
Data validation
Feature Engineering
Data splitting
Model creation
Building a model
Training optimization
Model validation
Training at scale
Model
Data
Rollout
Deploying
Cloud
Edge
Serving
Model
Monitoring & Logging & Expantibility
Finetune & improvements

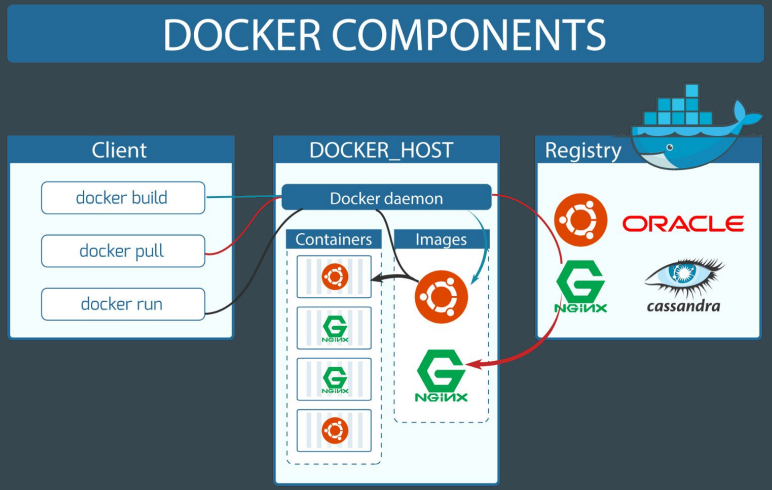
ML Pipeline + CI/CD



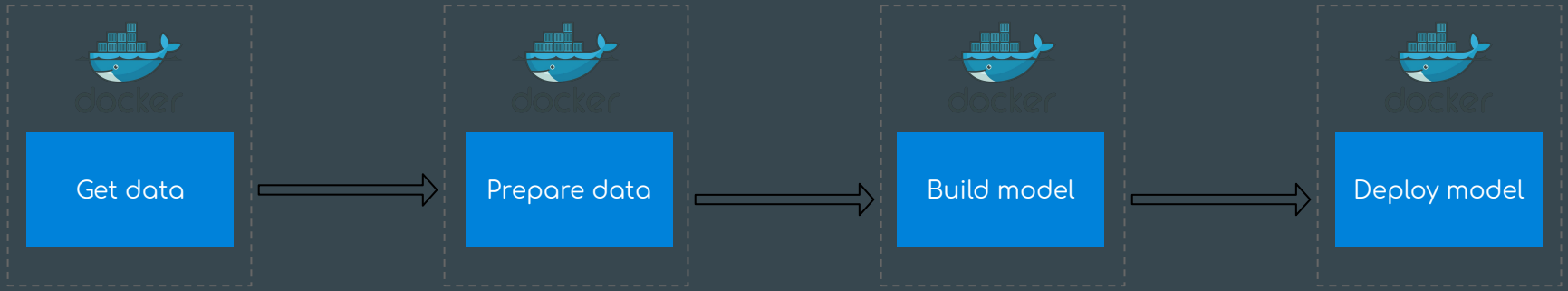
https://cloud.google.com/solutions/machine-learning/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

ML Pipeline + CI/CD | Containerization

```
1  FROM python:3.6-stretch
2  MAINTAINER Tina Bu <tina.hongbu@gmail.com>
3
4  # install build utilities
5  RUN apt-get update && \
6          apt-get install -y gcc make apt-transport-https ca-certificates build-essential
7
8  # check our python environment
9  RUN python3 --version
10 RUN pip3 --version
11
12 # set the working directory for containers
13 WORKDIR  /usr/src/<app-name>
14
15 # Installing python dependencies
16 COPY requirements.txt .
17 RUN pip install --no-cache-dir -r requirements.txt
18
19 # Copy all the files from the project's root to the working directory
20 COPY src/ /src/
21 RUN ls -la /src/*
22
23 # Running Python Application
24 CMD ["python3", "/src/main.py"]
```



ML Pipeline + CI/CD | Environment



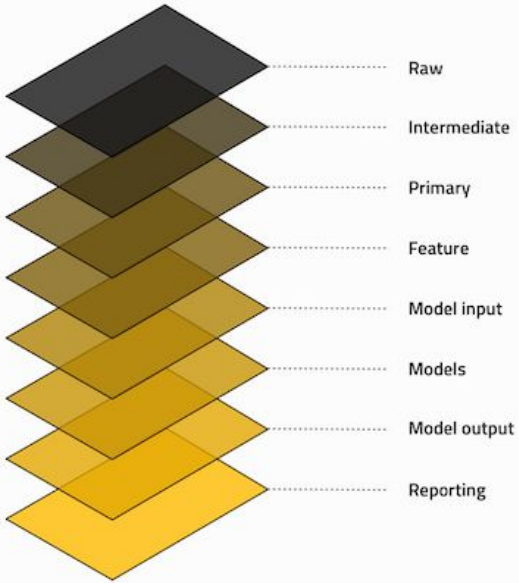
Kedro

Python framework for data and machine learning pipelines.

Applies software engineering best practices to optimize the process of switching from “Proof of Concept” type of projects to Data Science End-to-End, from concept to production environment.

https://kedro.readthedocs.io/
https://github.com/quantumblacklabs/kedro

Kedro | Data Engineering Convention

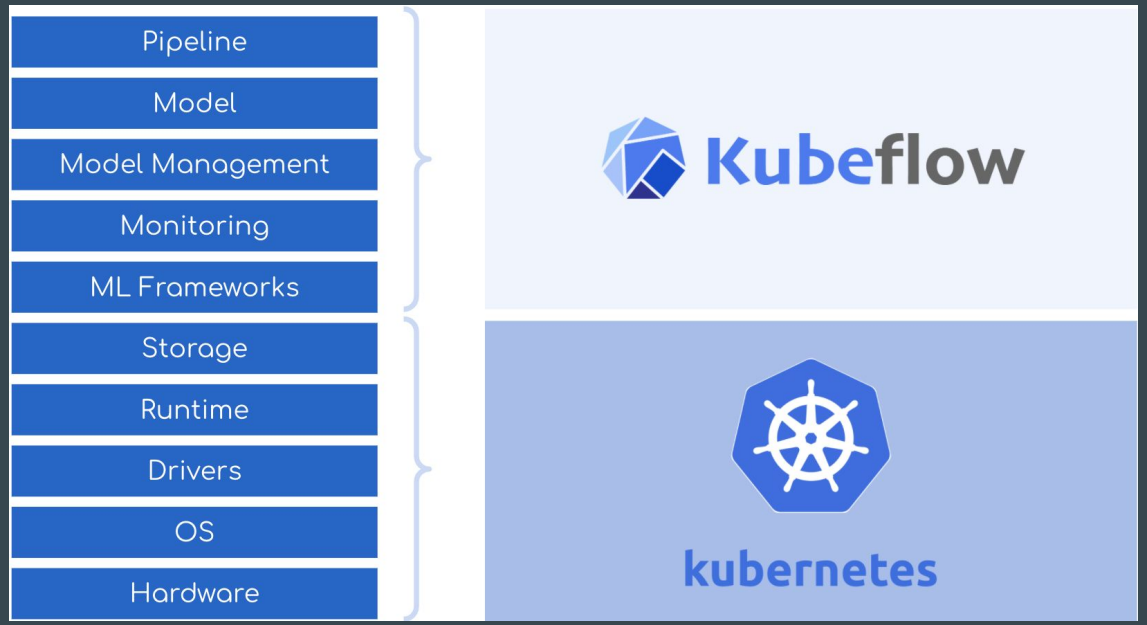


Folder in data	Description
Raw	Initial start of the pipeline, containing the sourced data model(s) that should never be changed, it forms your single source of truth to work from. These data models are typically un-typed in most cases e.g. csv, but this will vary from case to case.
Intermediate	Optional data model(s), which are introduced to type your `raw` data model(s), e.g. converting string based values into their current typed representation.
Primary	Domain specific data model(s) containing cleansed, transformed and wrangled data from either `raw` or `intermediate`, which forms your layer that you input into your feature engineering.
Feature	Analytics specific data model(s) containing a set of features defined against the `primary` data, which are grouped by feature area of analysis and stored against a common dimension.
Model input	Analytics specific data model(s) containing all `feature` data against a common dimension and in the case of live projects against an analytics run date to ensure that you track the historical changes of the features over time.
Models	Stored, serialised pre-trained machine learning models.
Model output	Analytics specific data model(s) containing the results generated by the model based on the `model input` data.
Reporting	Reporting data model(s) that are used to combine a set of `primary`, `feature`, `model input` and `model output` data used to drive the dashboard and the views constructed. It encapsulates and removes the need to define any blending or joining of data, improve performance and replacement of presentation layer without having to redefine the data models.

https://kedro.readthedocs.io/en/stable/12_faq/01_faq.html#what-is-data-engineering-convention

Kubeflow

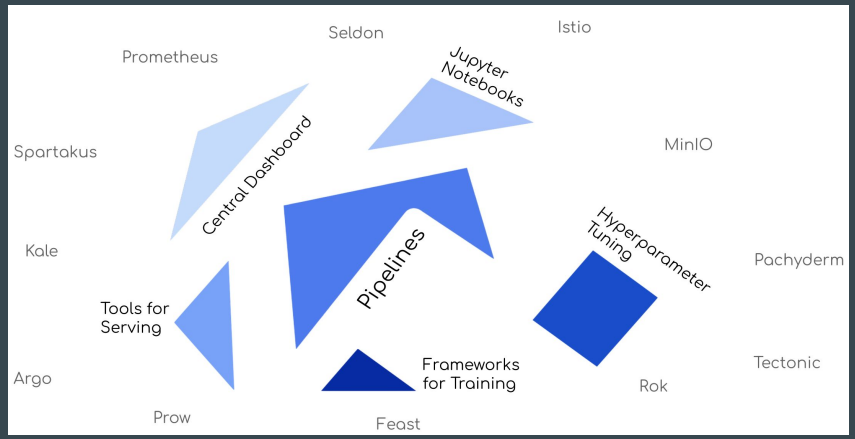
Kubeflow is a cloud-native platform designed to make deployments of machine learning workflows on Kubernetes simple, portable and scalable.



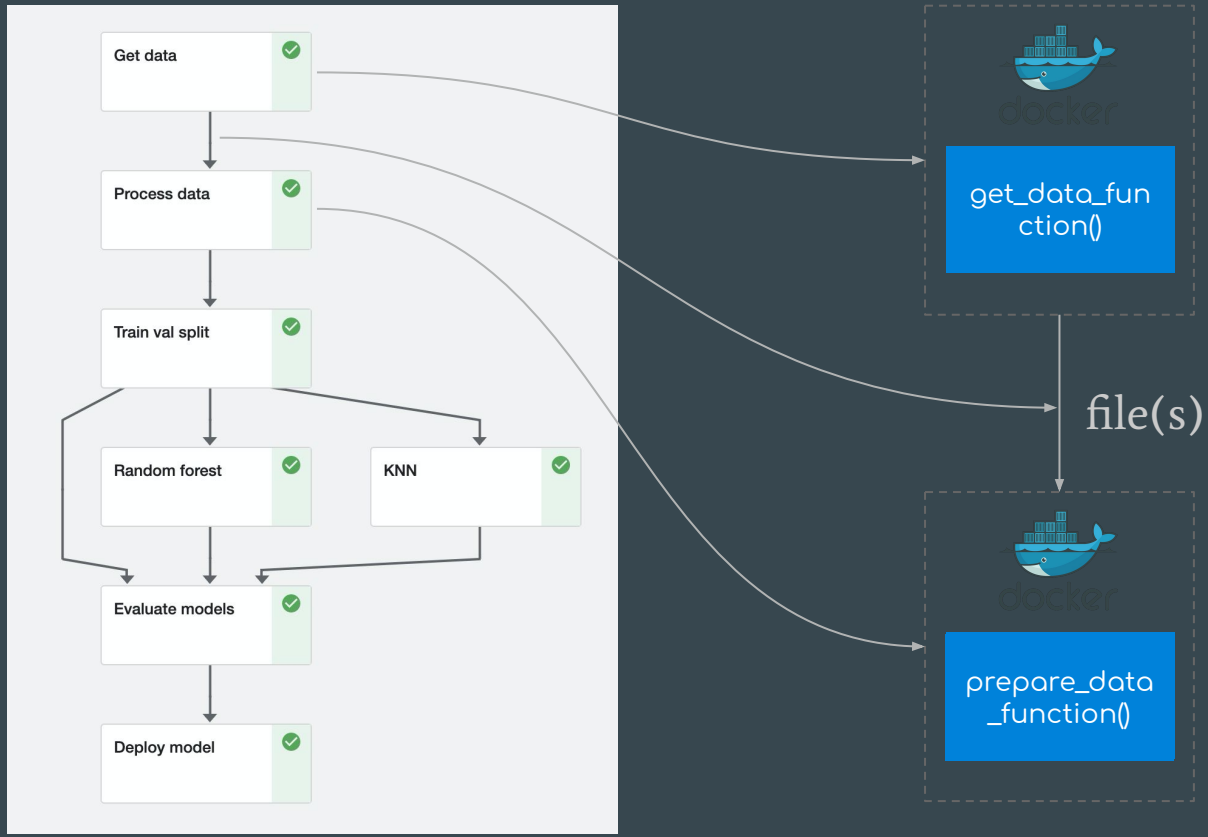
https://www.kubeflow.org/

Kubeflow | Mosaic of components

- Central Dashboard with multi-tenancy
- Metadata tracking
- Jupyter Notebooks on K8s
- Feature store
- Hyperparameter Tuning (Katib)
- Kubeflow Pipelines + SDK
- Kubeflow Fairing
- Serving - KFServing, Seldon, BentoML
- Nuclio



Kubeflow | Pipelines



Links

- Kedro video-tutorial by DataEngineerOne: https://www.youtube.com/watch?v=rf8yBHsDOj4&list=PLTU89LAWKRwEdiDKeMOU2ye6yU9Qd4MRo
- Data Science on AWS, with Kubeflow, by Antje Barth and Chris Fregly, 10h workshop: https://www.youtube.com/watch?v=9_SWaKdZhEM
- Machine Learning Engineering by Andriy Burkov, book: http://www.mlebook.com/
- MiniKF on AWS: https://aws.amazon.com/marketplace/pp/B08MBGH311
- Dive into Deep Learning, book: https://d2l.ai/index.html
- Deep Learning Drizzle, learning materials: https://deep-learning-drizzle.github.io/