Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

Autodeploy

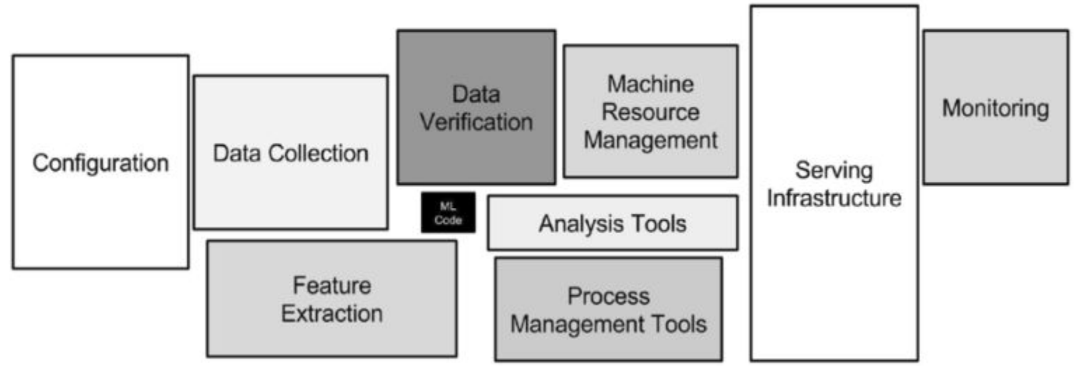
Scalable library for model management

Lenovo-BSC collaboration

January 2020

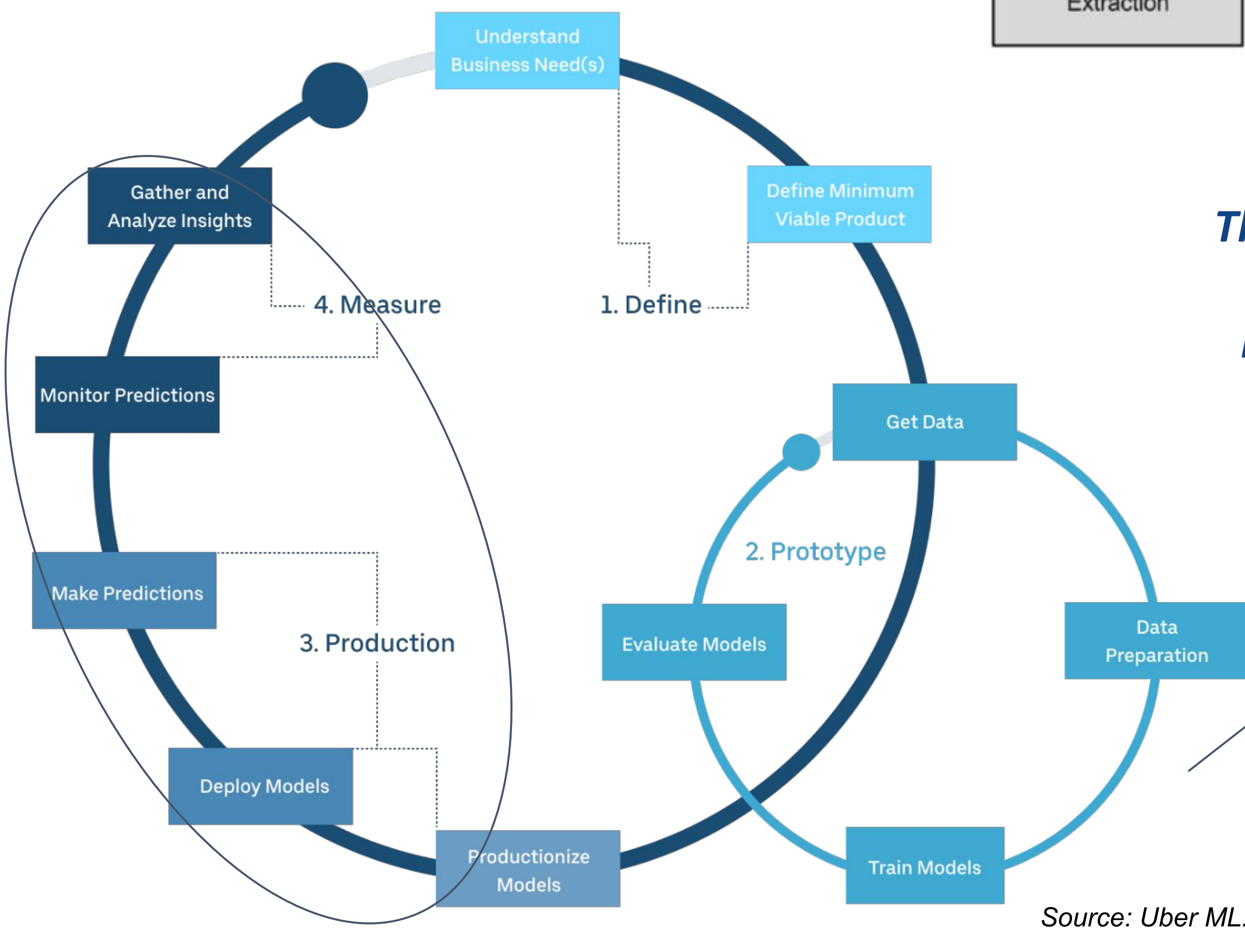
Applied Machine Learning: Workflow

Deployment: The ML code is a small part of the complete pipeline. More steps are needed to get it working.



Source: Sculley D., Holt. G.

The modeling phase is not the end of a workflow, more steps are needed.



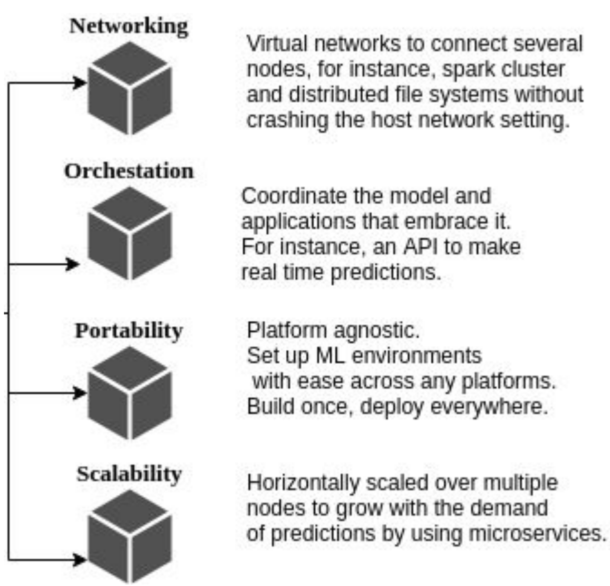
Source: Uber ML.

Applied ML workflow: Having ready predefined workflows for later use will decrease the prototyping and deployment time .

Applied Machine Learning: Issues

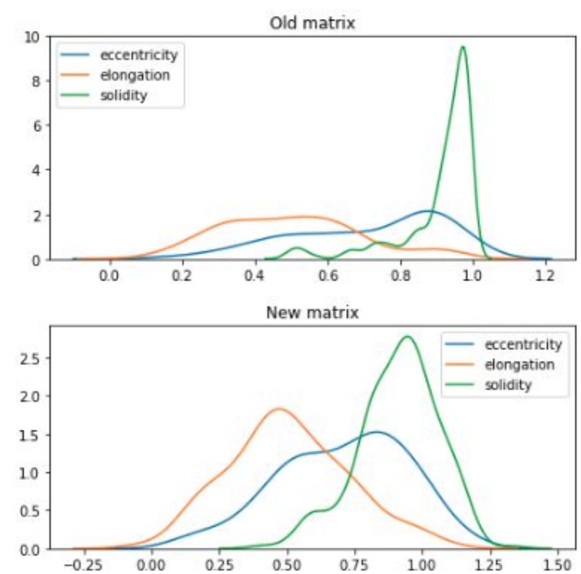
- Moving a complete workflow from development platform to another new platform can break things, e.g, operating system, libraries, dependencies, etc.
- Controlling a myriad of pipelines manually might be hard.
- Some steps in a workflow need different amount and type of computational resources, e.g, RAM, Storage, CPU, GPU.
- The complete workflow might scale from a single node to a cluster.
- The dataset distribution might change (normal, poison, etc, or different patterns). It is called distribution drift. An anomaly detector might help this.
- Some feature levels and balance of classes might change (categories, e.g, before {red, blue}, after {red, blue, black}. Classes, e.g, before {30% men, 70%women}. after {60% men, 40%women}).

Flexibility, portability, scalability



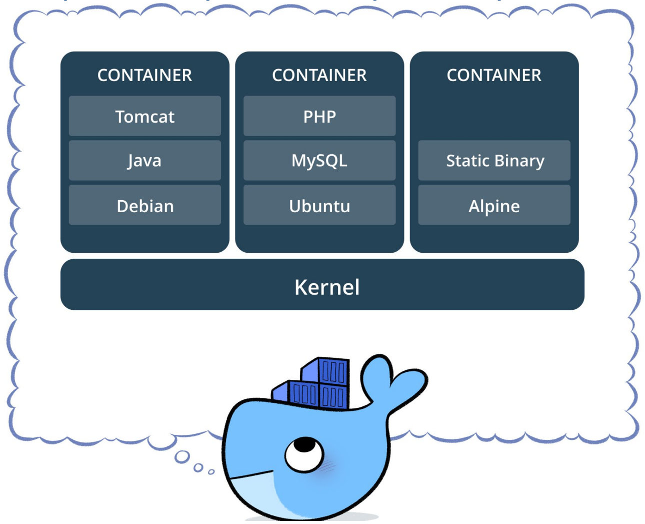
In production, many unexpected situations might arise.

Early drift distribution detector



Applied Machine Learning: Recipes

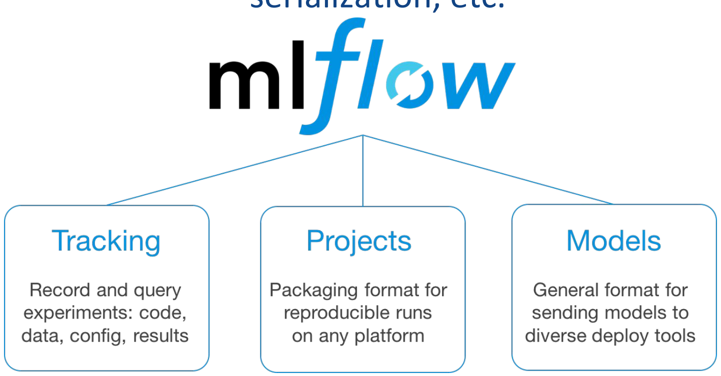
Docker: Isolate environments, portability, scalability, affinity, etc.



Source: arquitectoit.com

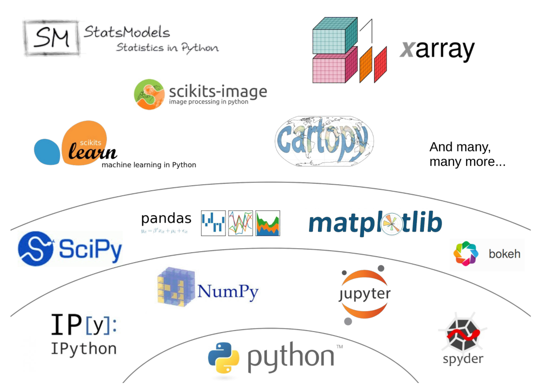
Autodeploy is a high-level library that is built on top of these tools to manage and supervise workflows efficiently.

MLflow: Track metrics, organize projects, model versioning and serialization, etc.

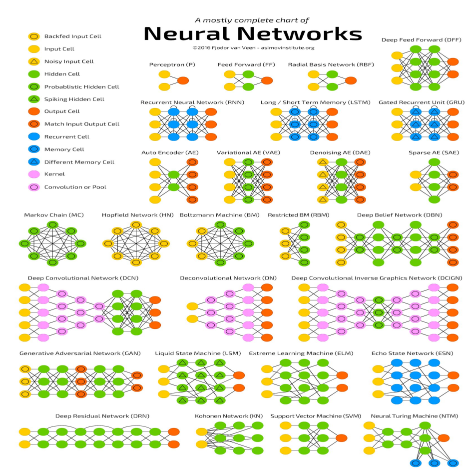


Source: mlflow.org

Python: Fast prototyping and expressiveness. Robust AI ecosystem.



Source: leblancfg.com



Source: fjodor van veen - asimovinstitute.org

Comparison ML tools

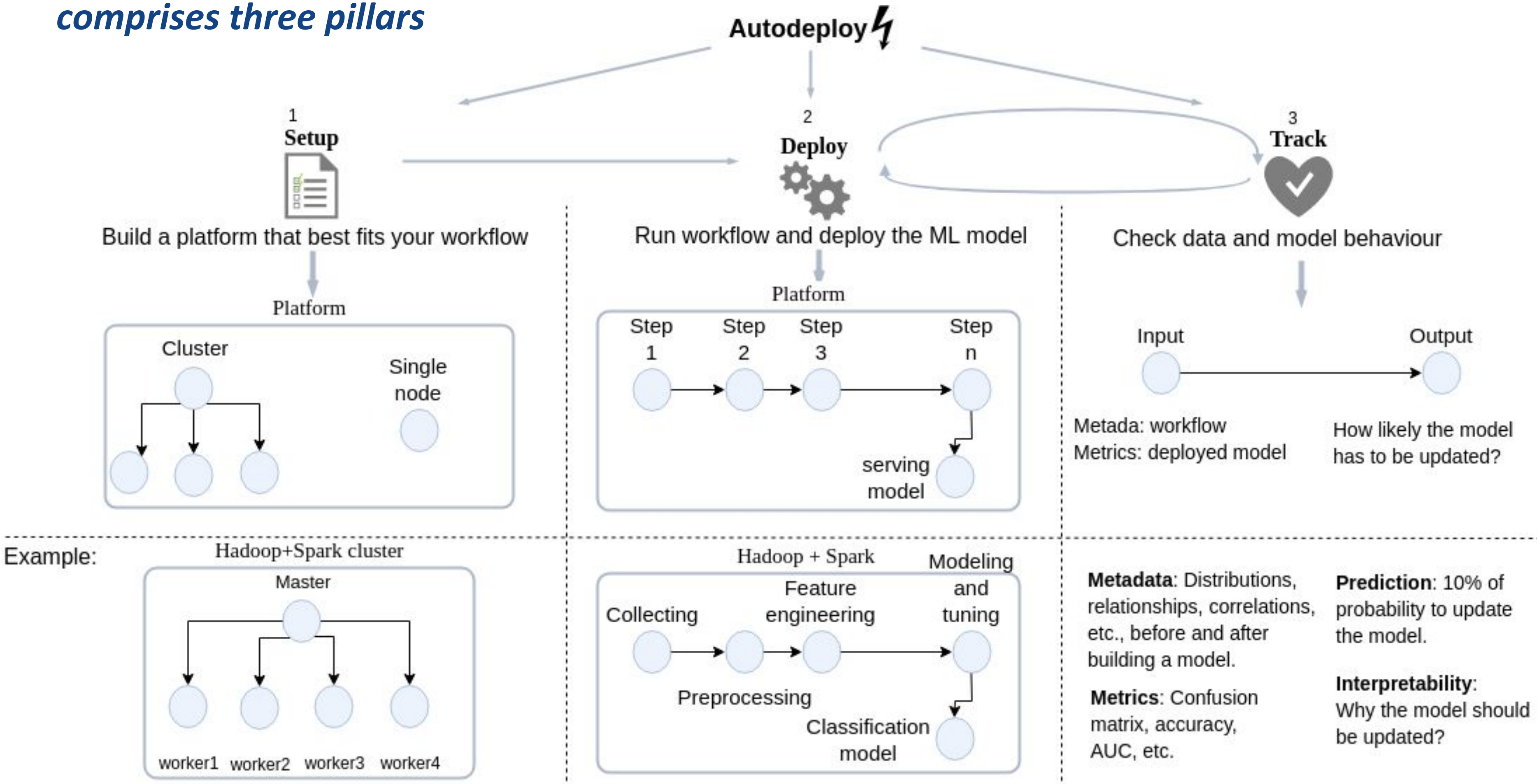
	MLflow	Autodeploy	Kubeflow/Airflow
Learnability (ease of use)	Medium	High	Low
Scheduling capability	No	No	Yes
Dynamic execution	No	Yes	Yes
Experiments tracking	Yes	Yes	Yes
Model versioning	Yes	Yes	Yes
Model checker	No	Yes	No
Orchestration-agnostic	Yes	Yes	No

- Pre-deployment (steps required for getting a model): MLflow, Autodeploy.
- Deployment (put a model into production): MLflow, Autodeploy, Kubeflow/Airflow.
- Post-Deployment (check the model's health): Autodeploy.
- Ease of use: Autodeploy > MLflow > Kubeflow/Airflow.

Autodeploy's main goal is flexibility and real time checking

Autodeploy: High-level overview

The overall idea comprises three pillars

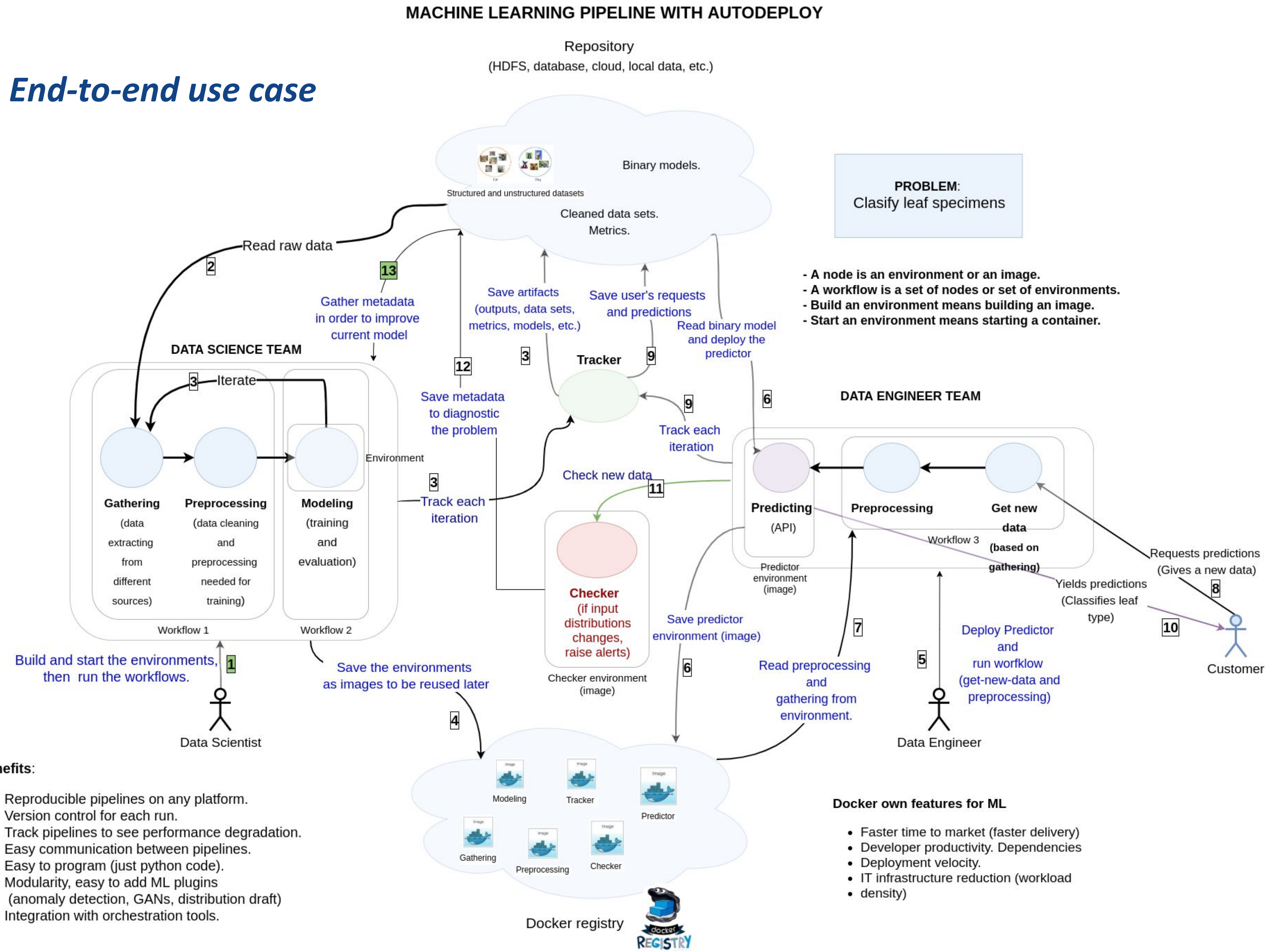


Define what and how your workflows will be used for.

Start the environments, run the workflows and publish models.

Each workflow can be surrounded by a checker. This checker can be seen as a supervisor of future behaviours.

End-to-end use case



Benefits:

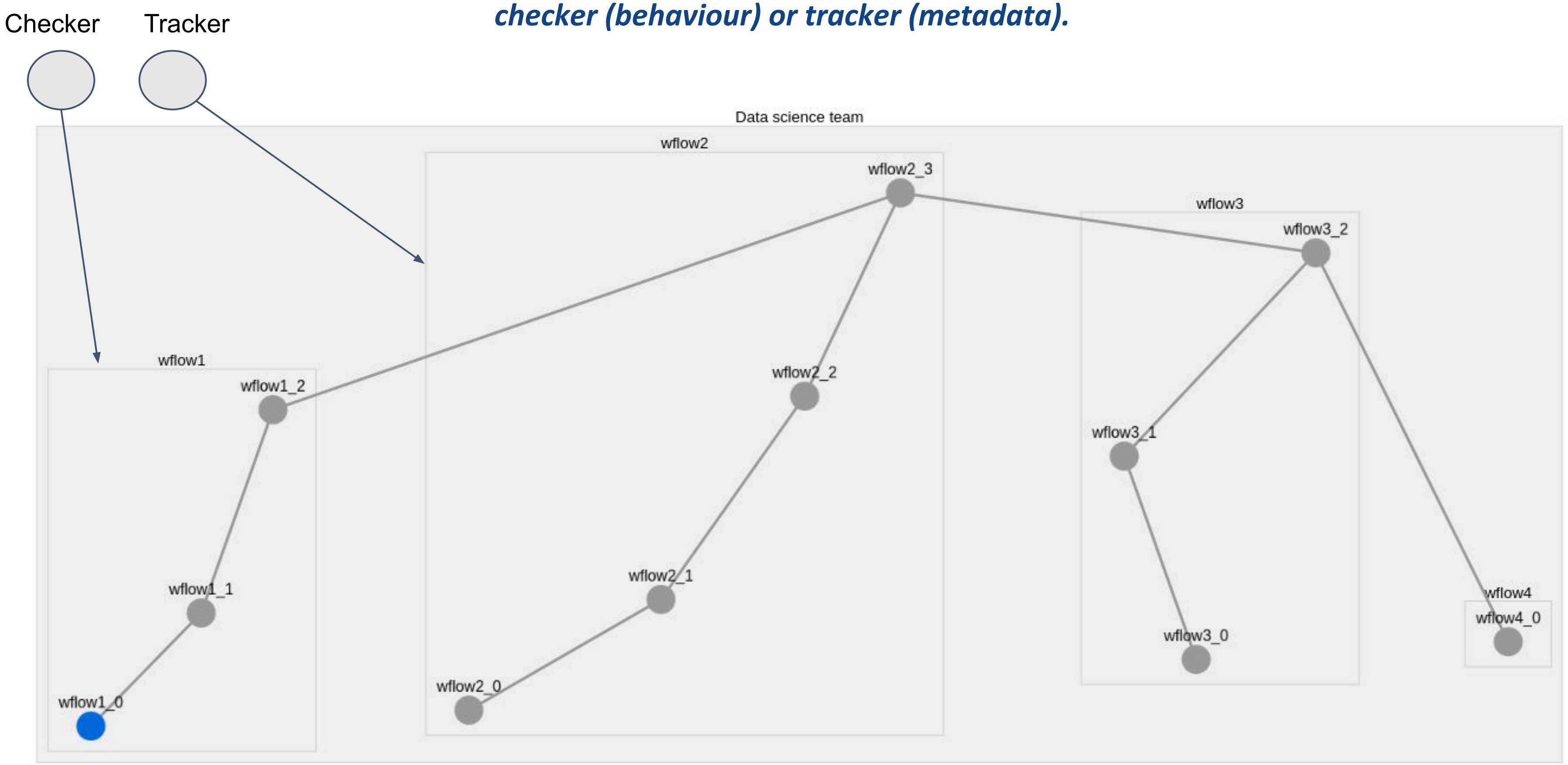
- Reproducible pipelines on any platform.
- Version control for each run.
- Track pipelines to see performance degradation.
- Easy communication between pipelines.
- Easy to program (just python code).
- Modularity, easy to add ML plugins
- (anomaly detection, GANs, distribution draft)
- Integration with orchestration tools.

Docker own features for ML

- Faster time to market (faster delivery)
- Developer productivity. Dependencies
- Deployment velocity.
- IT infrastructure reduction (workload
- density)

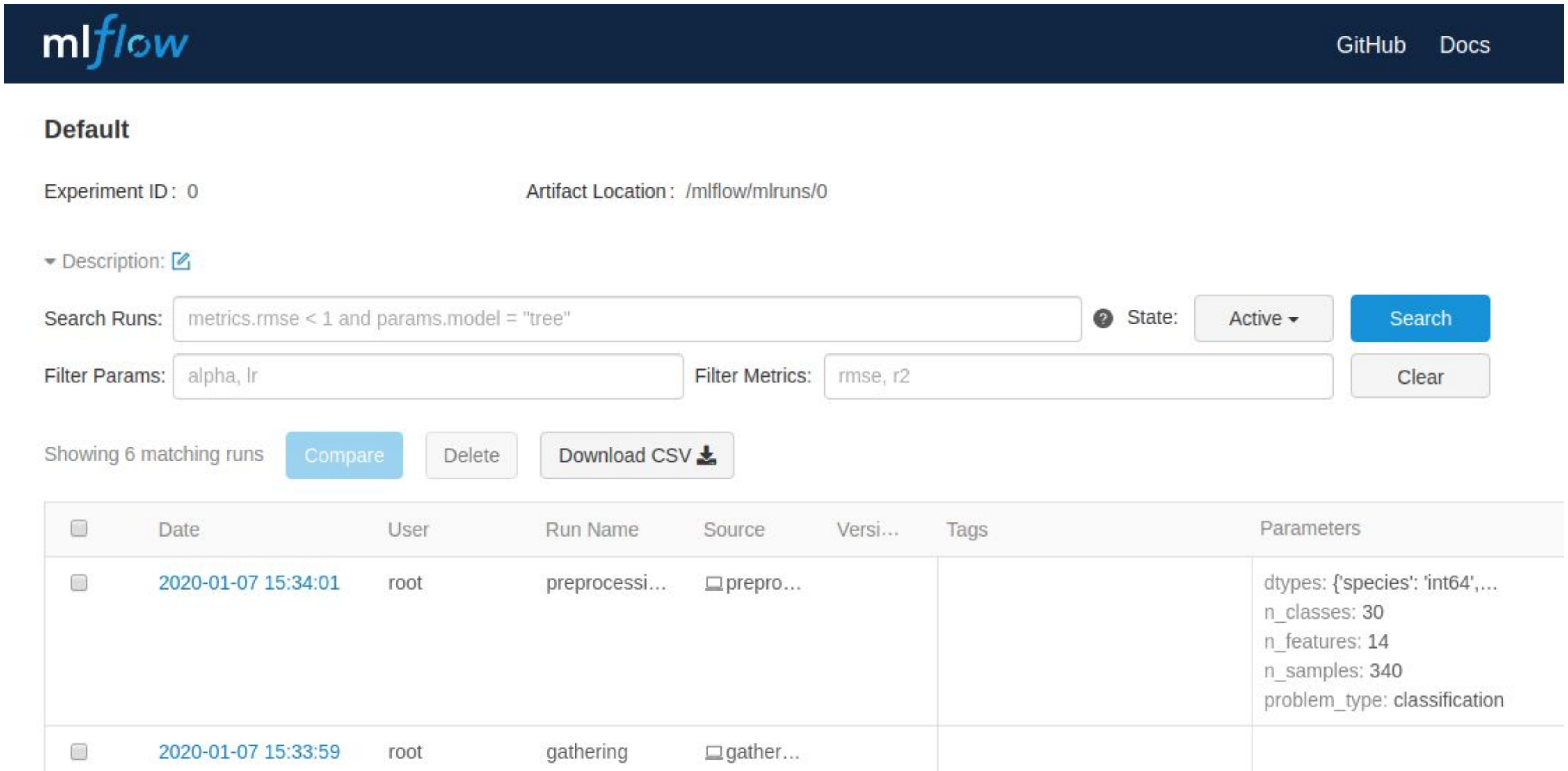
Setup and deployment

Design nested and parallel workflows: *Each node is a computation function, e.g preprocessing or modeling. They can be run following the user ordering design. Besides, each node is a docker container (environment) inside a workflow (rectangle in the following picture). Finally, each workflow has its own checker (behaviour) or tracker (metadata).*



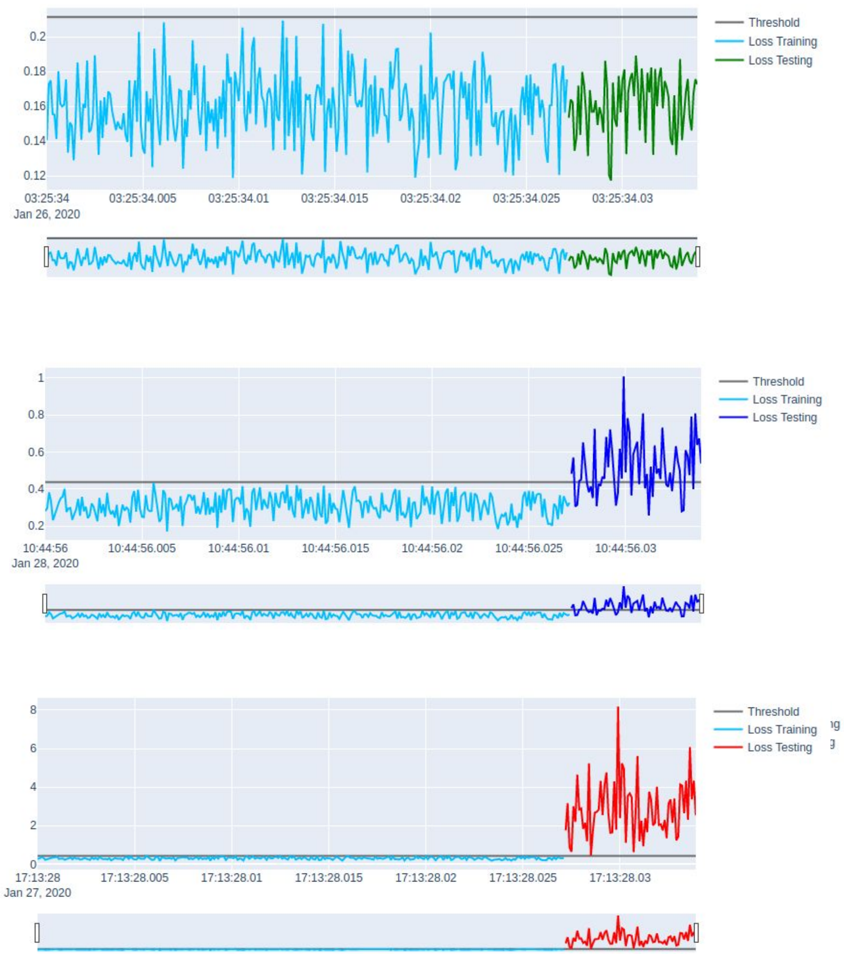
Tracking: Save any workflow metadata for future analysis

This module logs intermediate results belonging to a workflow such as, metrics, statistics, scores, etc.



Checking: supervise how a workflow might behave

Early drift distribution detector (built-in function)



Custom plugins can be added to check the behaviour of any workflow: for sanity, integrity, anomaly, interpretability, etc.

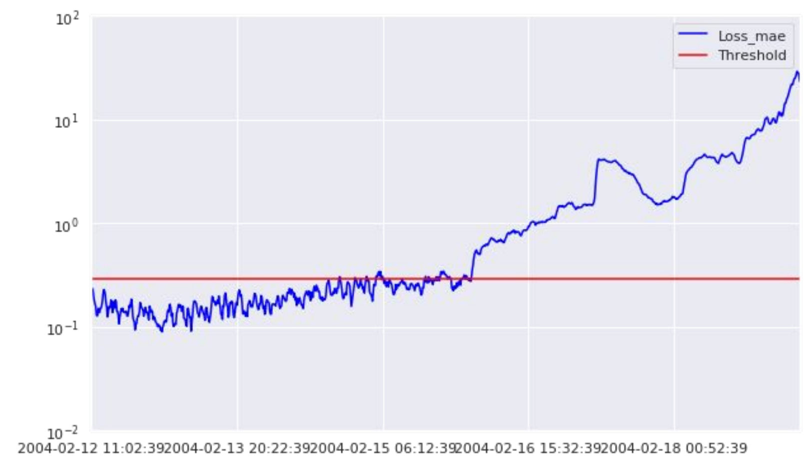
Any plugin is supposed to be run in real time

Early drift distribution detector: architecture

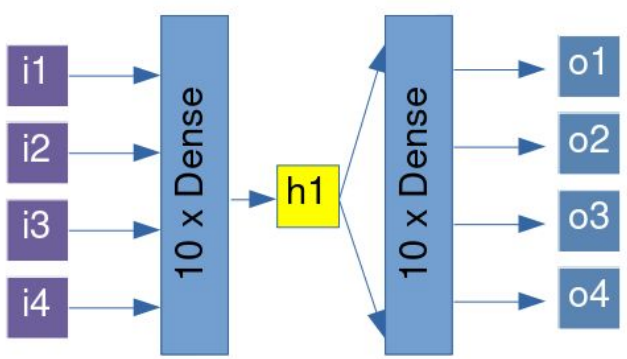
Reconstruction error is calculated to measure how different is a new distribution from the original one

Input

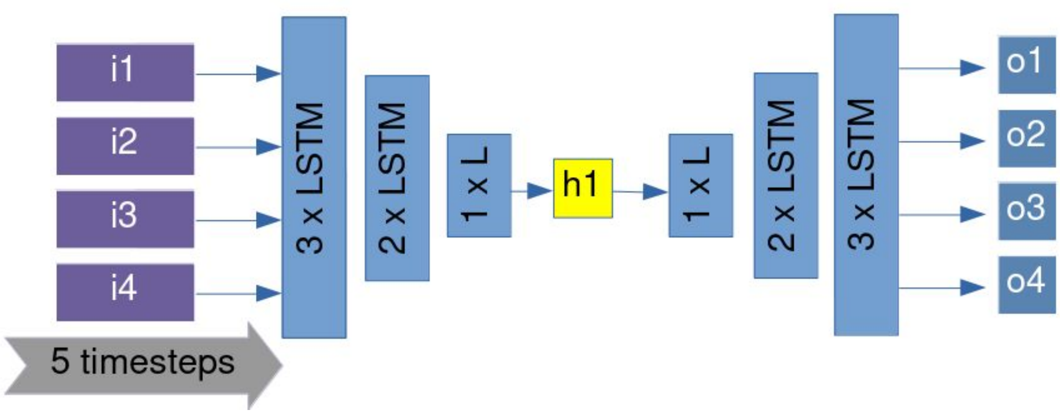
Output



Naive Autoencoder

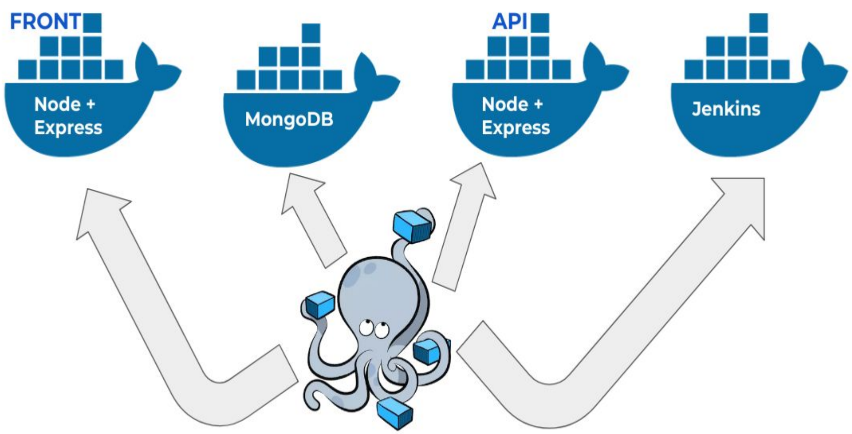


LSTM Autoencoder



Integration: Docker compose

Docker compose behaviour



Source: medium.com

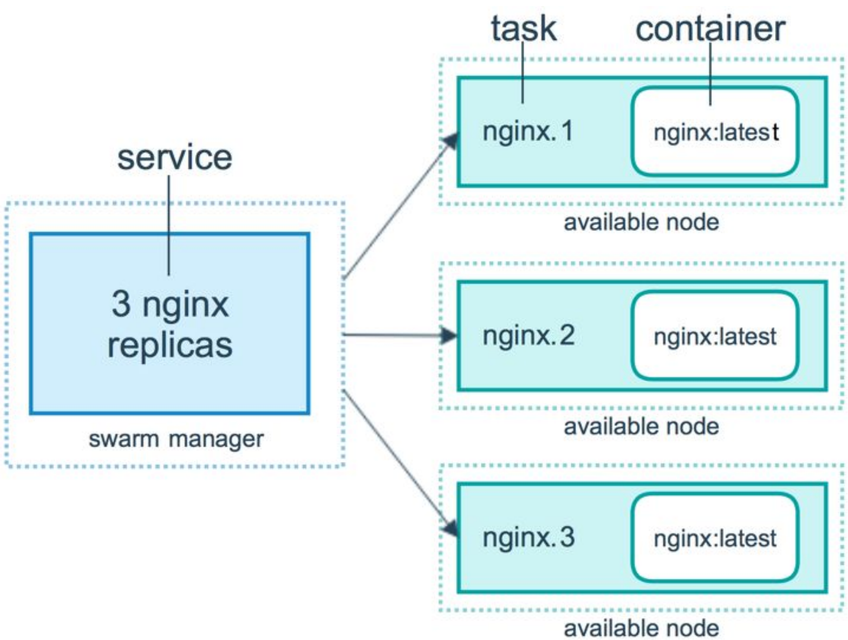
docker-compose.yml

```yaml
version: '3'
services:
  get_new_data:
    image: get_new_data
    container_name: get_new_data-20200126033621
    networks:
    - network-workflow3
    depends_on:
    - tracker-workflow3
    environment:
      MLFLOW_TRACKING_URI: http://tracker-workflow3:8003
    volumes:
    - /home/guess/Desktop/autodeploy/examples/demo2/data-
eng/:/app
    - /home/guess/Desktop/autodeploy/examples/demo2/data-
eng/ad-stuff/ad-tracker/tracker-workflow3:/mlflow
    tty: 'true'
  preprocessing_new_data:
    image: preprocessing_new_data
    container_name: preprocessing_new_data-20200126033621
    networks:
    - network-workflow3
    depends_on:
    - tracker-workflow3
    environment:
      MLFLOW_TRACKING_URI: http://tracker-workflow3:8003
    volumes:
    - /home/guess/Desktop/autodeploy/examples/demo2/data-
eng/:/app
    - /home/guess/Desktop/autodeploy/examples/demo2/data-
eng/ad-stuff/ad-tracker/tracker-workflow3:/mlflow
    tty: 'true'
  tracker-workflow3:
    image: tracker-workflow3
    container_name: tracker-workflow3-20200126033621
    networks:
    - network-workflow3
    volumes:
    - /home/guess/Desktop/autodeploy/examples/demo2/data-
eng/ad-stuff/ad-tracker/tracker-workflow3:/mlflow
    ports:
    - 8008:8003
networks:
  network_workflow3: null
```

Once finished defining the workflows, they can be saved as a docker-compose file

Integration: Docker Swarm

Docker Swarm behaviour



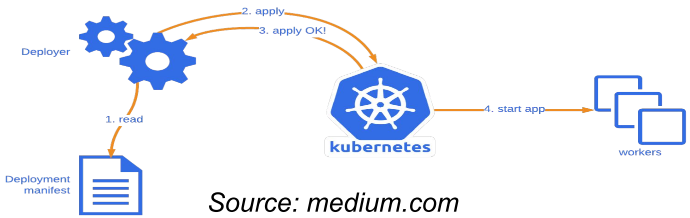
Similar to docker-compose file, some changes are done to deploy a swarm cluster

Source: filepicker.io

Docker Swarm console

```
[xgbravo@nxt2027 compose_repo]$ docker service ls
ID            NAME                              MODE        REPLICAS  IMAGE                             PORTS
5r1l6yx27pfk  my_swarm_gathering                replicated  1/1       gathering:latest
xke5uf9aqdh4  my_swarm_modeling                 replicated  1/1       modeling:latest
ro9haibzt9ma  my_swarm_preprocessing            replicated  1/1       preprocessing:latest
pvdud6whi4pg  my_swarm_tracker_workflow1        replicated  1/1       tracker_workflow1:latest          *:8006->8001/tcp
alnxiq8y3tfs  my_swarm_tracker_workflow1_scale  replicated  0/5       my_swarm:latest
2dh4zpwu48od  my_swarm_tracker_workflow1_scale2 replicated  0/5       my_swarm_tracker_workflow1:latest
ilzhcp546xft  my_swarm_tracker_workflow1_scale3 replicated  5/5       tracker_workflow1:latest
ygwulqaah8a8  my_swarm_tracker_workflow2        replicated  1/1       tracker_workflow2:latest          *:8007->8002/tcp
```

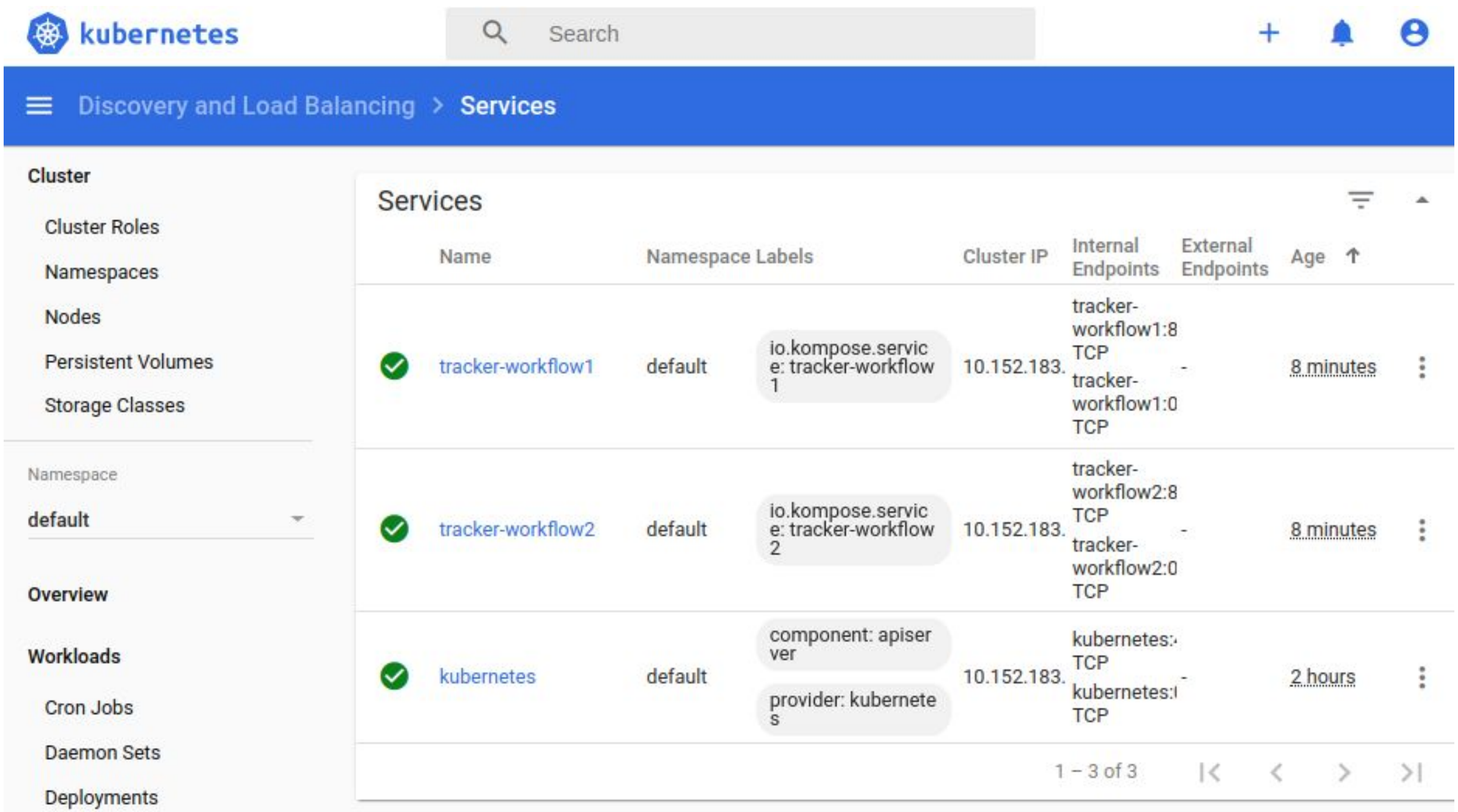
Integration: Kubernetes



Source: medium.com

On going integration with kubernetes

Kubernetes dashboard



Is it relevant?. AI predictions for 2020.

Creator of pytorch: … " place more value on AI model performance beyond accuracy. "

Celeste Kidd, psychologist at the University of California, Berkeley: … " increased awareness of the real-life implications of tech tools … "

Jeff Dean, Google AI chief: … " he wants to see less of an emphasis on slight state-of-the-art advances in favor of creating more robust models. "

Anima, Anandkumar, NVIDIA: … " self-supervision, and self-training methods of training models, which are the kinds of models that can improve through self-training with unlabeled data. "

Dario gil, IBM: …" focus on metrics beyond accuracy to consider the value of models deployed in production. Shifting the field toward building trusted systems instead of prioritizing accuracy above all else will be a central pillar to the continued adoption of AI. "

Keywords: **robust models, interpretable models, trusted models, self-supervision (automatic)**.

Source: https://venturebeat.com/2020/01/02/top-minds-in-machine-learning-predict-where-ai-is-going-in-2020/

Current state

- Creation of nested workflows where each node can be an executor, tracker or checker.
- Isolation of each workflow using docker, it comprises auto-creation of networks, volumes, docker files, images, containers and registries.
- Tested integration with docker-compose.
- Deployment of ML models and function for consumption.
- Tracking module for saving any metadata.
- Checking module (for now one plugin) for post-deployment.

Future work

- Enhance interface module for checking.
- Dashboard for checking module.
- Improve compatibility with docker-compose, swarm, kubernetes.
- Test more use cases.
- Start formalizing autodeploy for writing a paper.
- Start writing documentation.
- Set up autodeploy as a python package.
- Improve drift distribution checker.
- Add option for wipe out any metadata (tracking and checking).
- Add a new plugin for integrity checking.
- Add option to compress all the settings needed to transfer an application.
- Add interface to write on databases.
- Add option to plot nested workflows in jupyter notebooks.