



**Barcelona
Supercomputing
Center**
Centro Nacional de Supercomputación



Autodeploy

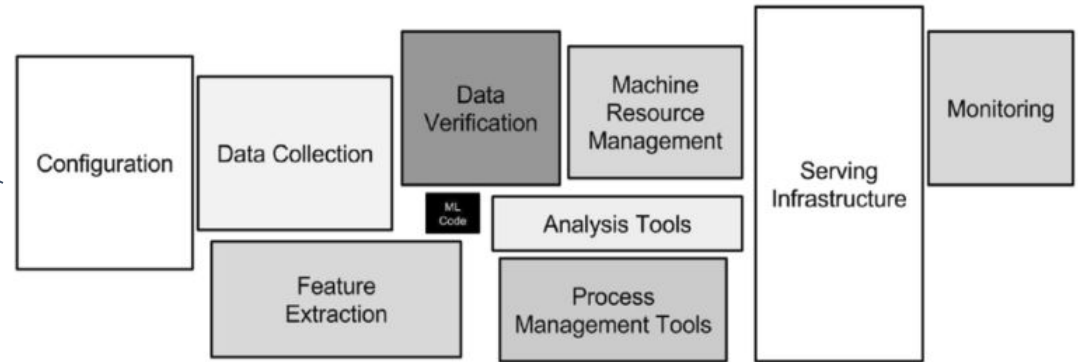
**Scalable library for end-to-end
ML workflow management**

Lenovo-BSC collaboration

January 2020

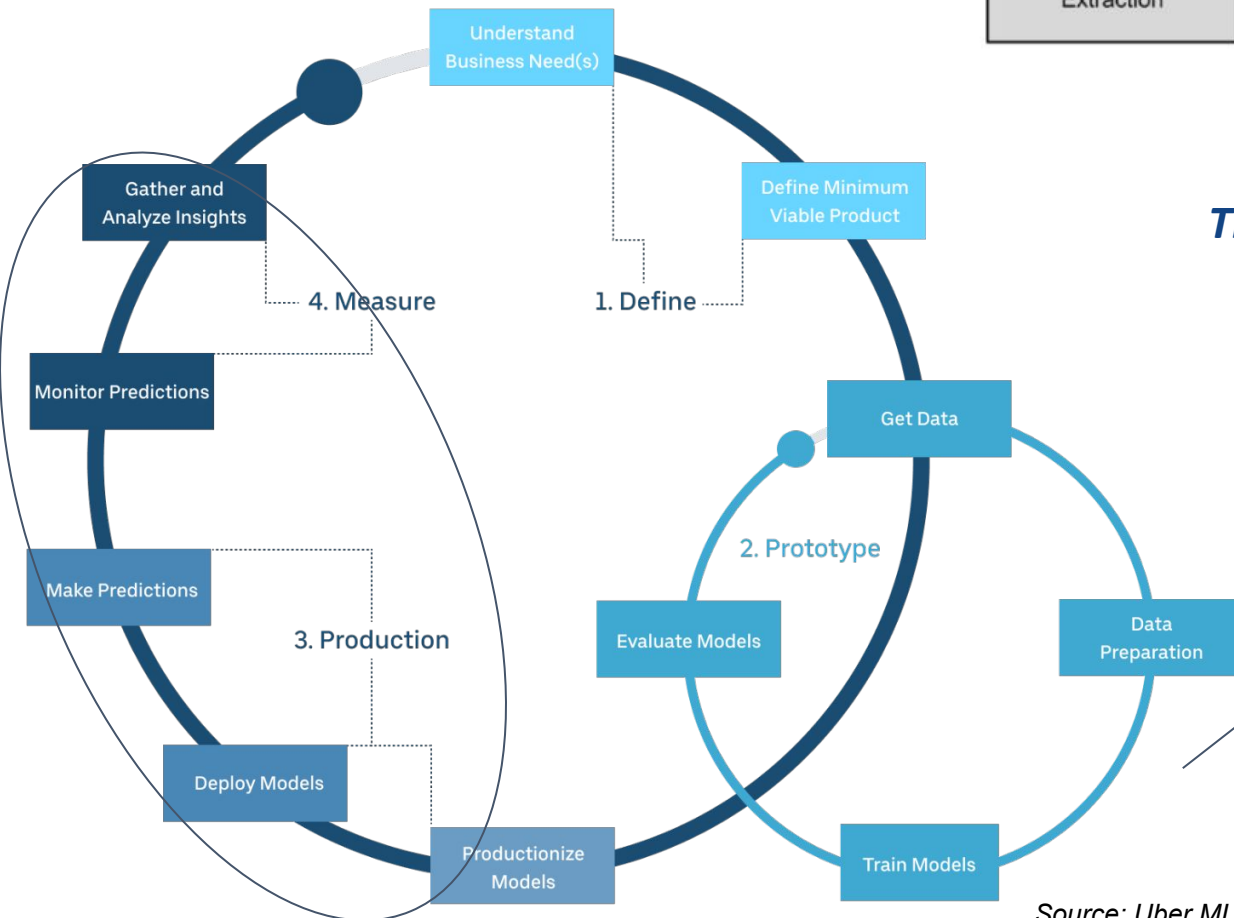
Applied Machine Learning: Workflow

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



Source: Sculley D., Holt. G.

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



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

Source: Uber ML.

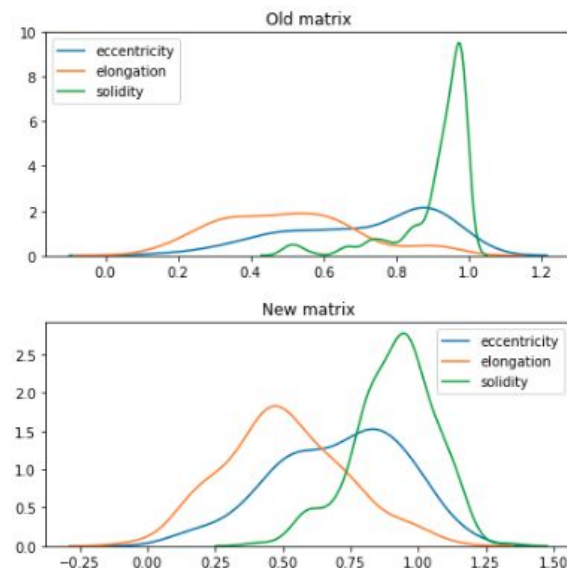
Applied Machine Learning: Issues

- 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}).

Data scientists require tools for ease of deployment, tracking and reproducing experiments.

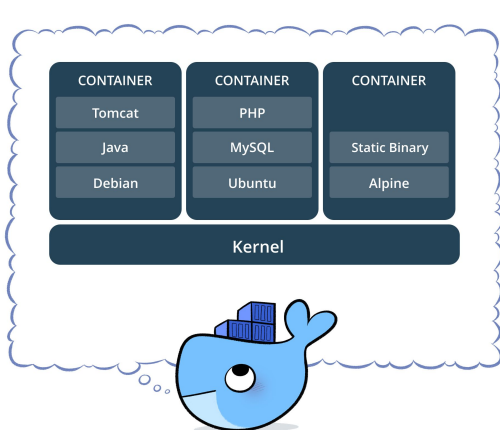
In production, many unexpected situations might arise such as drift distributions, unformatted data, etc.

Early drift distribution detector



Applied Machine Learning: Recipes

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



Source: arquitectoit.com

Networking



Virtual networks to connect several nodes, for instance, spark cluster and distributed file systems without crashing the host network setting.

Orchestration



Coordinate the model and applications that embrace it. For instance, an API to make real time predictions.

Portability



Platform agnostic. Set up ML environments with ease across any platforms. Build once, deploy everywhere.

Scalability



Horizontally scaled over multiple nodes to grow with the demand of predictions by using microservices.

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

Python: Fast prototyping and expressiveness. Robust AI ecosystem.

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

mlflow

Tracking

Record and query experiments: code, data, config, results

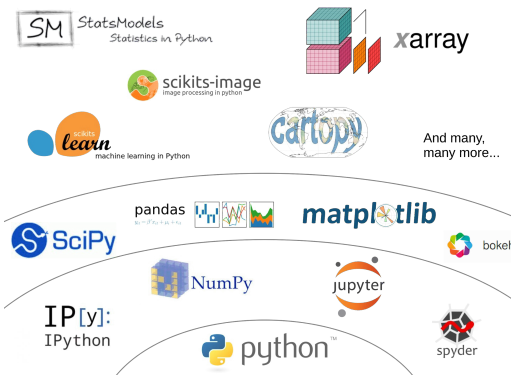
Projects

Packaging format for reproducible runs on any platform

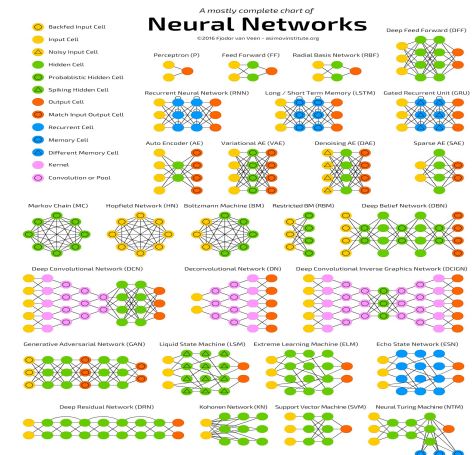
Models

General format for sending models to diverse deploy tools

Source: mlflow.org



Source: leblancfg.com



Source: fjodor.van.veen-asimovinstitute.org

Comparison ML workflow tools

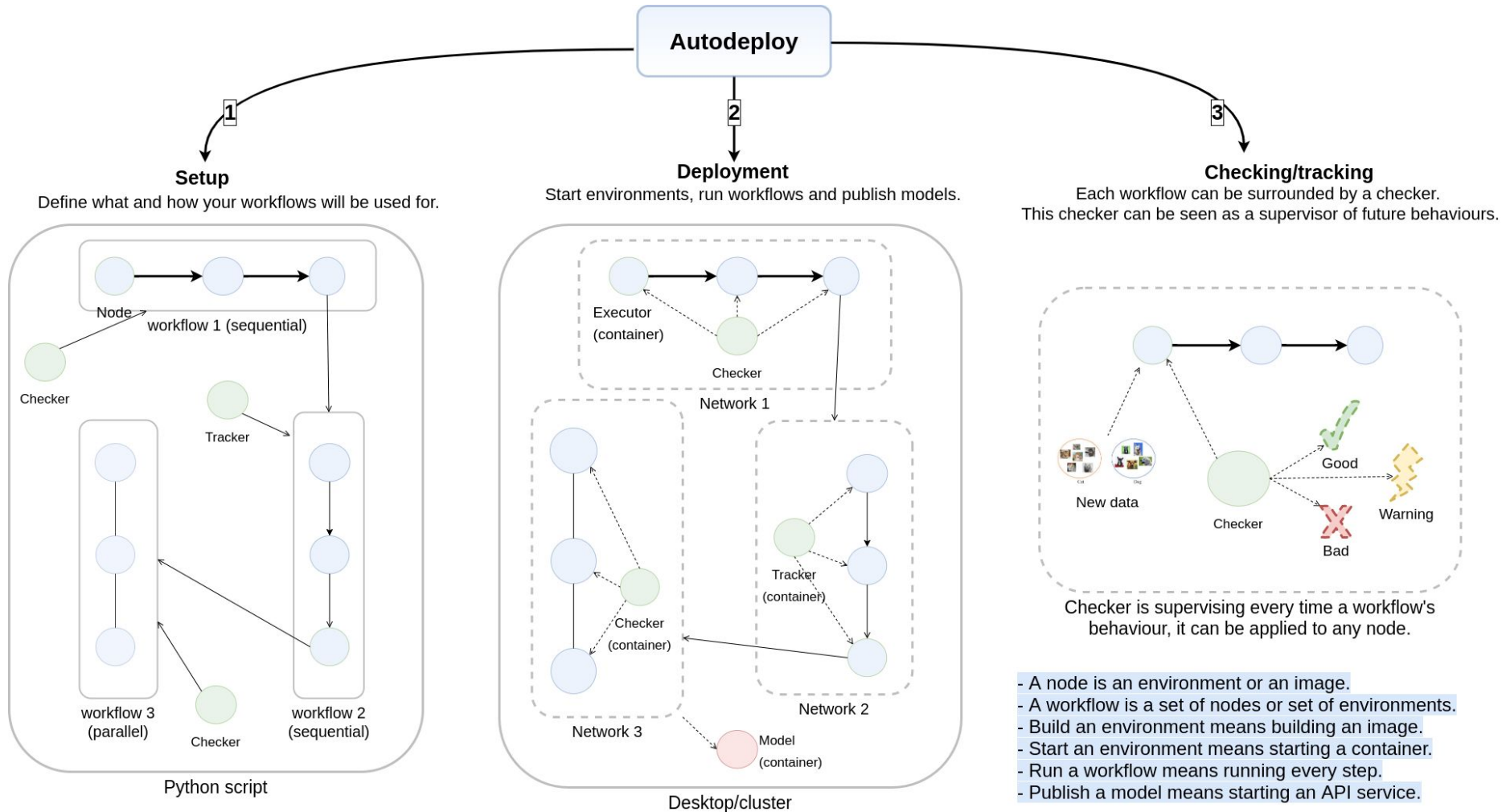
	MLflow	Autodeploy	Kubeflow/Airflow
Usability	Medium	High	Low
Built-in scheduling	No	No	Yes
Dynamic execution	No	Yes	Yes
Experiment tracking	Yes	Yes	Yes
Model versioning	Yes	Yes	Yes
Model checking	No	Yes	No
Orchestration-agnostic	Yes	Yes	No

Most appropriate tools for:

- 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 goals are usability, integration for deployment and real-time checking

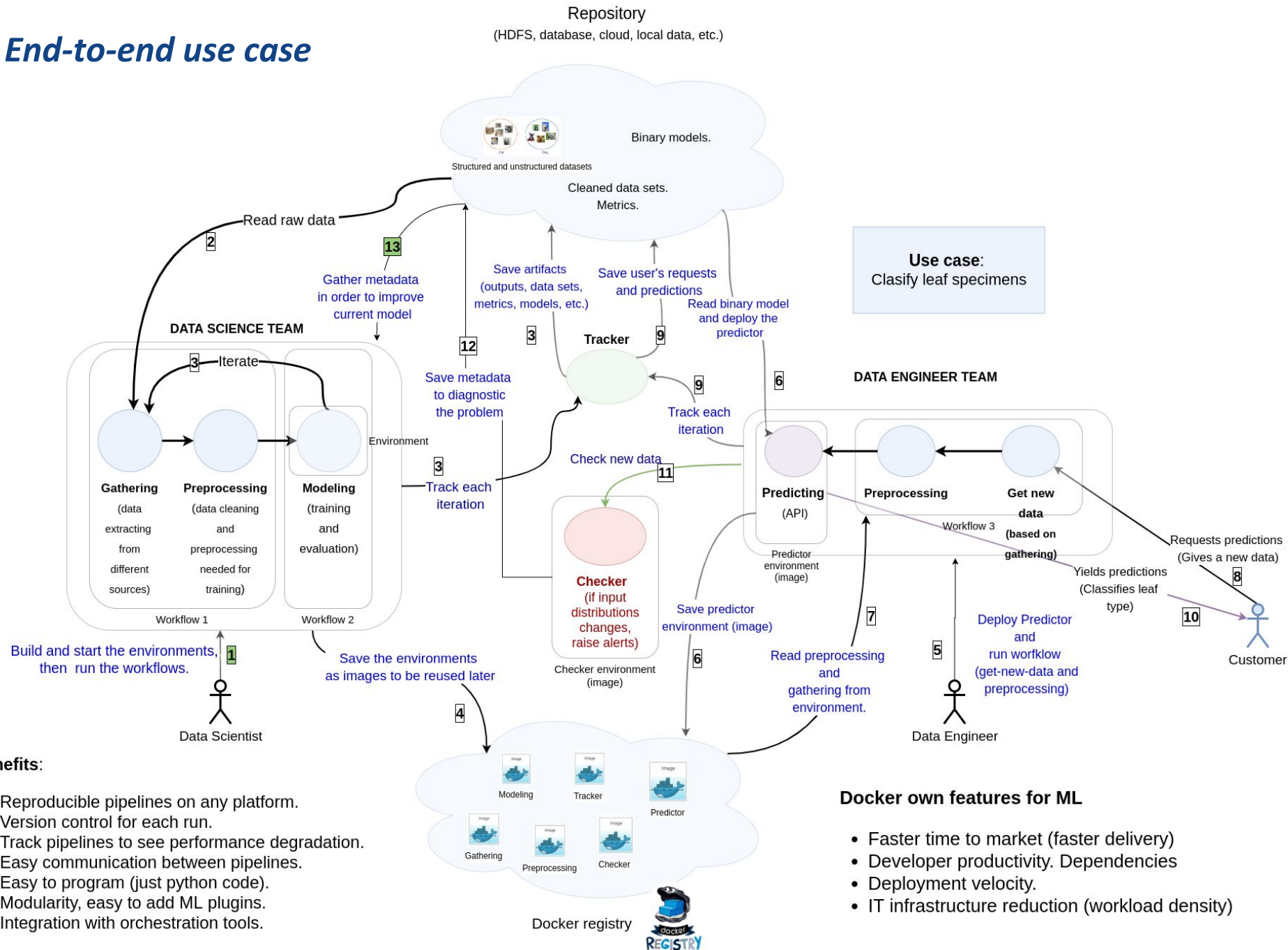
Autodeploy: High-level overview



Autodeploy comprises three pillars

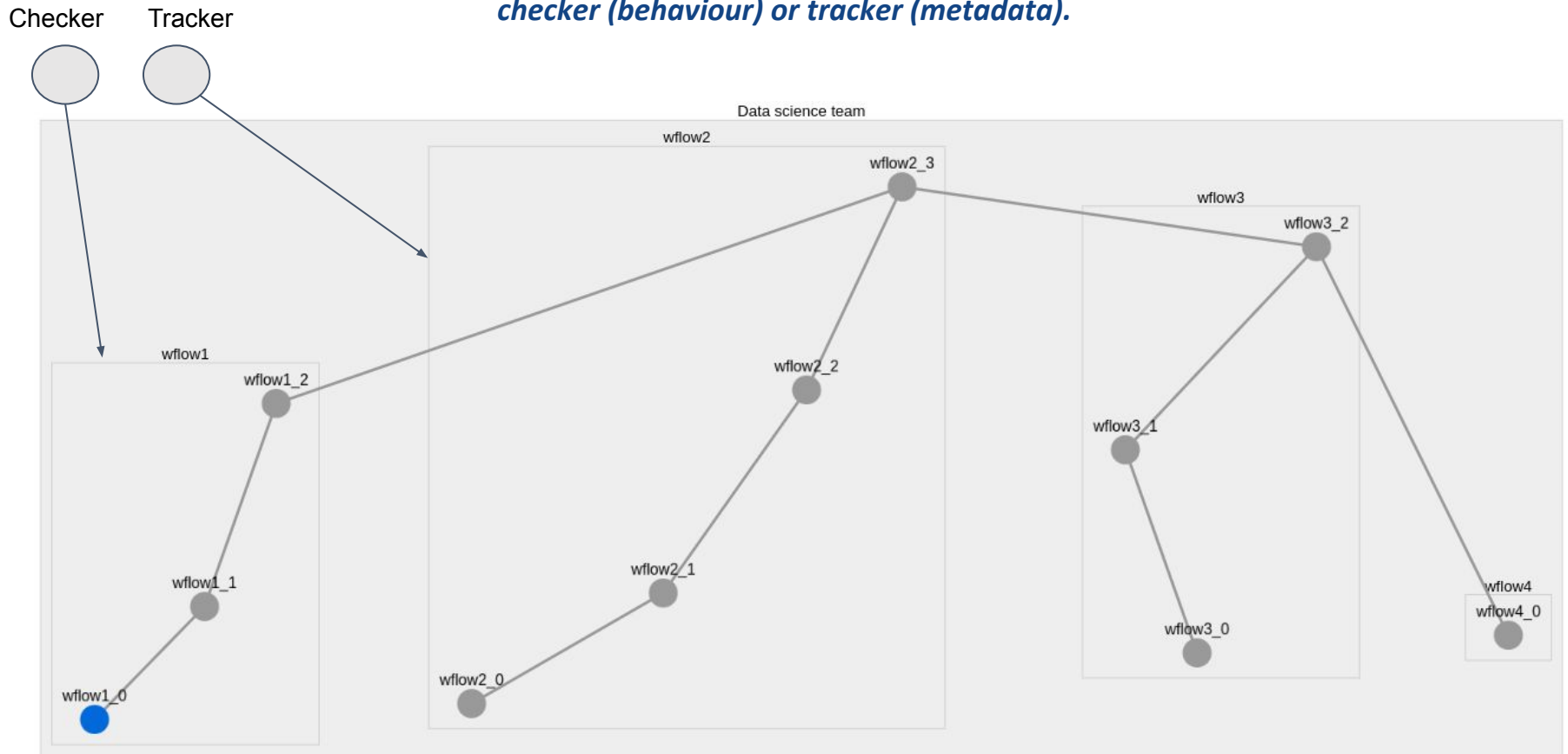
MACHINE LEARNING PIPELINE WITH AUTODEPLOY

End-to-end use case



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,
settings, metrics, statistics, scores, etc.

mlflow

[GitHub](#) [Docs](#)

Default

Experiment ID: 0

Artifact Location: /mlflow/mlruns/0

▼ Description: [✎](#)

Search Runs: metrics.rmse < 1 and params.model = "tree"



State:

Active ▼

Search

Filter Params: alpha, lr

Filter Metrics: rmse, r2

Clear

Showing 6 matching runs

Compare

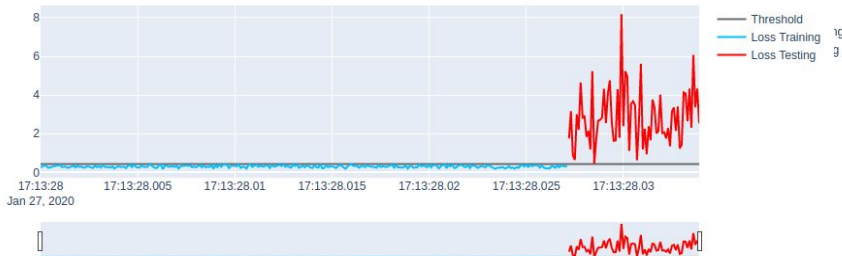
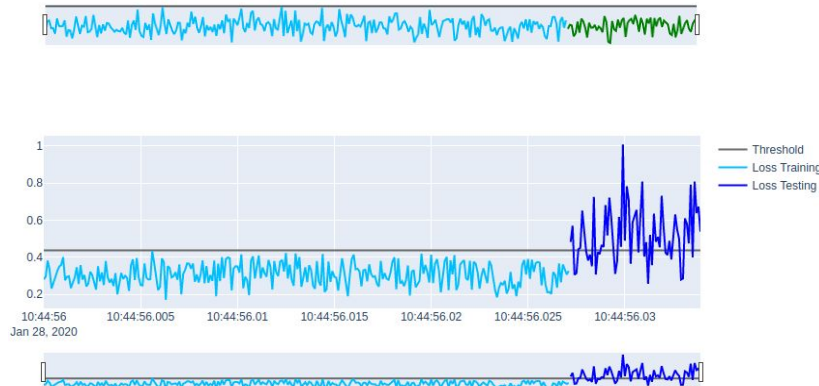
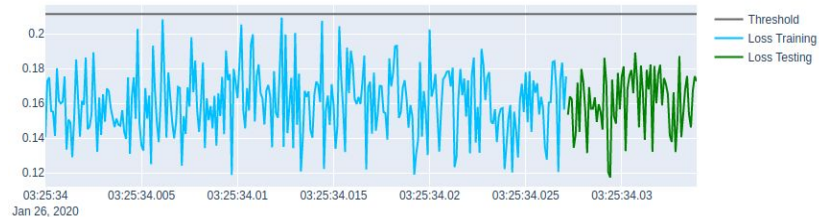
Delete

Download CSV

<input type="checkbox"/>	Date	User	Run Name	Source	Versi...	Tags	Parameters
<input type="checkbox"/>	2020-01-07 15:34:01	root	preprocessi...	prepro...			dtypes: {'species': 'int64', ... n_classes: 30 n_features: 14 n_samples: 340 problem_type: classification
<input type="checkbox"/>	2020-01-07 15:33:59	root	gathering	gather...			

Checking: supervise workflow's behaviour in production

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



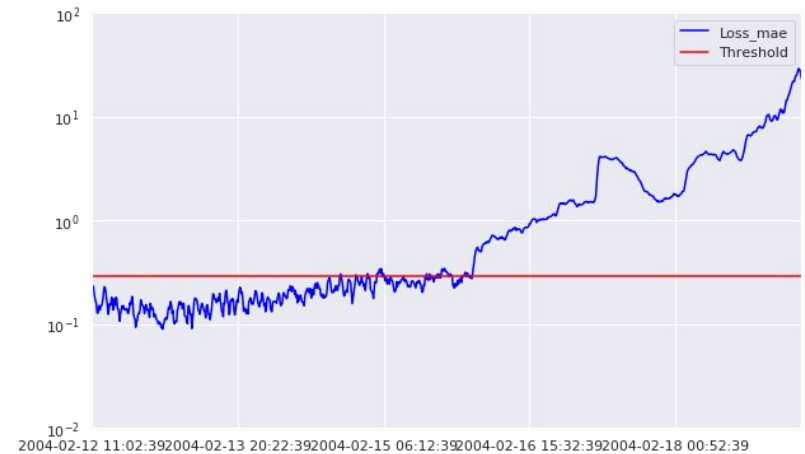
Early drift distribution detector: architecture

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

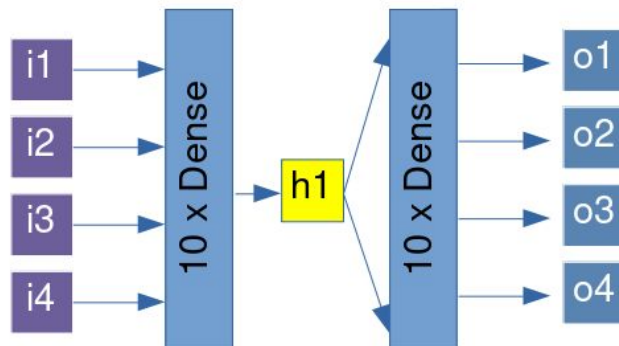
Input



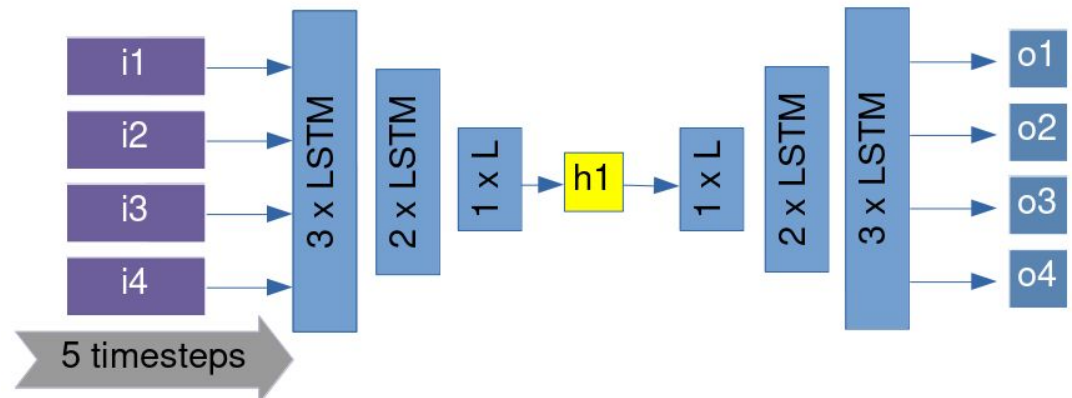
Output



Naive Autoencoder



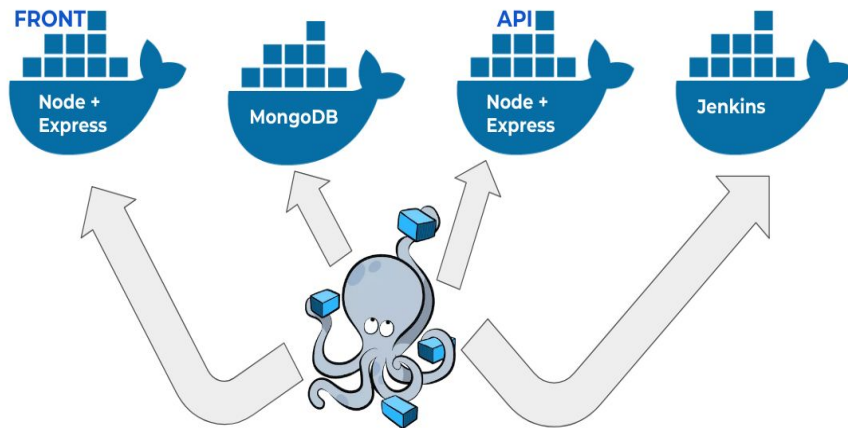
LSTM Autoencoder



Deployment integration: Docker compose

`docker-compose.yml`

Docker compose behaviour



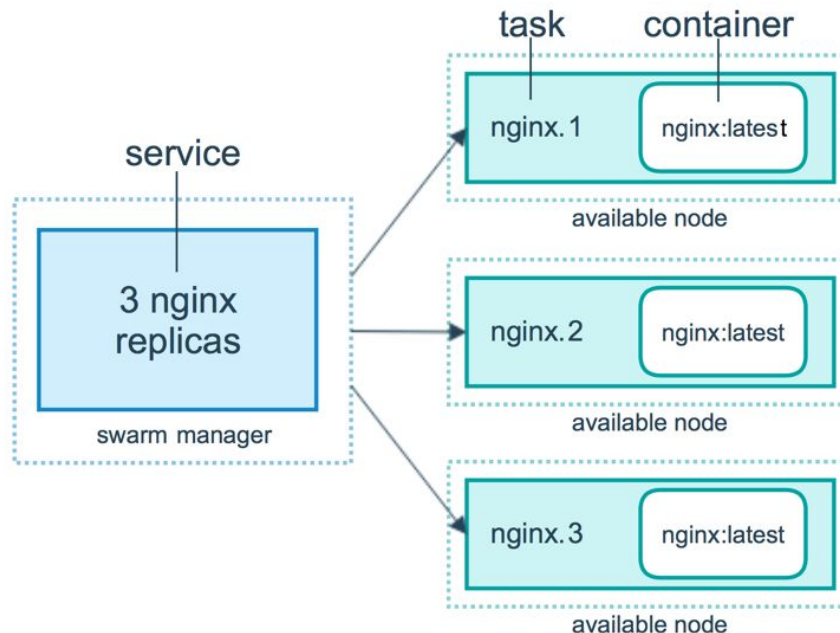
Source: medium.com

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

```
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
```

Deployment integration: Docker Swarm

Docker Swarm behaviour



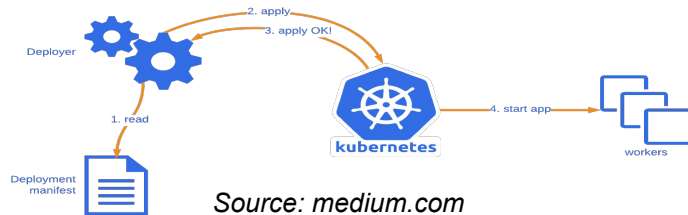
Autodeploy provides a docker-stack file to deploy a swarm cluster to schedule the workflows.

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
alnxig8y3tfs	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

Deployment integration: Kubernetes



Source: medium.com

Autodeploy provides a kube.config file to deploy a kubernetes cluster to schedule the workflows.

Kubernetes dashboard

The screenshot shows the Kubernetes dashboard with the 'Services' page selected. The left sidebar contains navigation links for Cluster, Namespaces, Nodes, Persistent Volumes, Storage Classes, Overview, and Workloads. The main area displays a table of services.

Name	Namespace	Labels	Cluster IP	Internal Endpoints	External Endpoints	Age
tracker-workflow1	default	io.kompose.service: tracker-workflow1	10.152.183.	tracker-workflow1:8 TCP tracker-workflow1:0 TCP	-	8 minutes
tracker-workflow2	default	io.kompose.service: tracker-workflow2	10.152.183.	tracker-workflow2:8 TCP tracker-workflow2:0 TCP	-	8 minutes
kubernetes	default	component: apiserver provider: kubernetes	10.152.183.	kubernetes: TCP kubernetes: TCP	-	2 hours

1 - 3 of 3

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 status

- 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.
- Ongoing development on swarm and kubernetes integration.
- Deploy ML models into services for prediction.
- Tracking module for saving any metadata.
- Checking module (for now, drift distribution plugin) for post-deployment.

Future work

- Lightweight integration with additional third-party tools like LICO platform.
- 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.