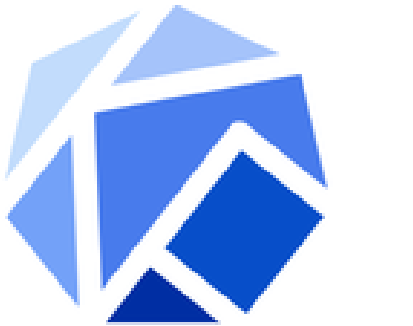
SophiaConf



Kubeflow

Jean Armel Luce

orange™

About me …

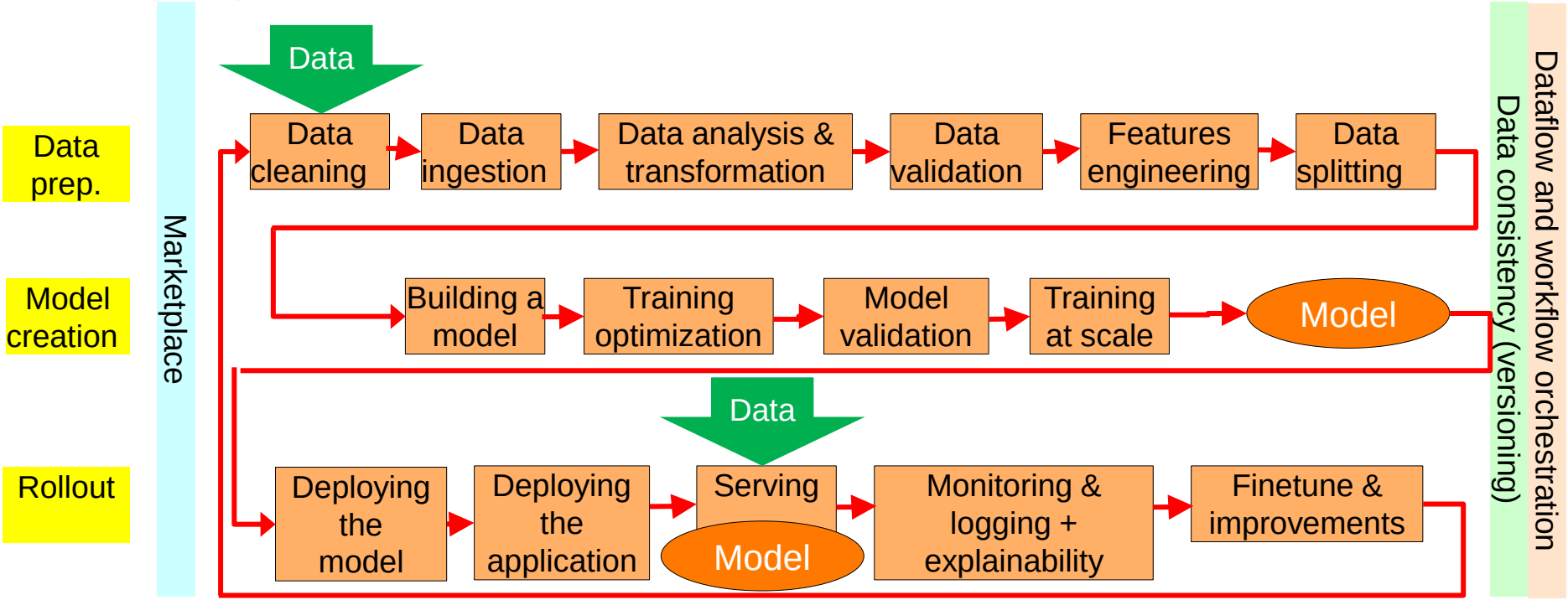
I work at Orange

I like to try, test, experiment …. and push to production

I like to share

- Meetups and local events
- C* Summit EU 2013 (London) : migration from RDBMS to C*
- C* Summit 2014 (San Francisco) : graphs and Hive with C*
- C* Summit 2016 (San Jose) : ML with Spark and C*
- ApacheCon N.A. 2019 (Las Vegas) : C* deployment on top of K8S

ML workflow



Several different actors

… using several different tools
… and some hardware resources

Kubeflow’s mission

Make it Easy for Everyone to Develop, Deploy and Manage Portable, Distributed ML on Kubernetes

Cloud Architecture

- Kubernetes
- Microservices
- Serverless

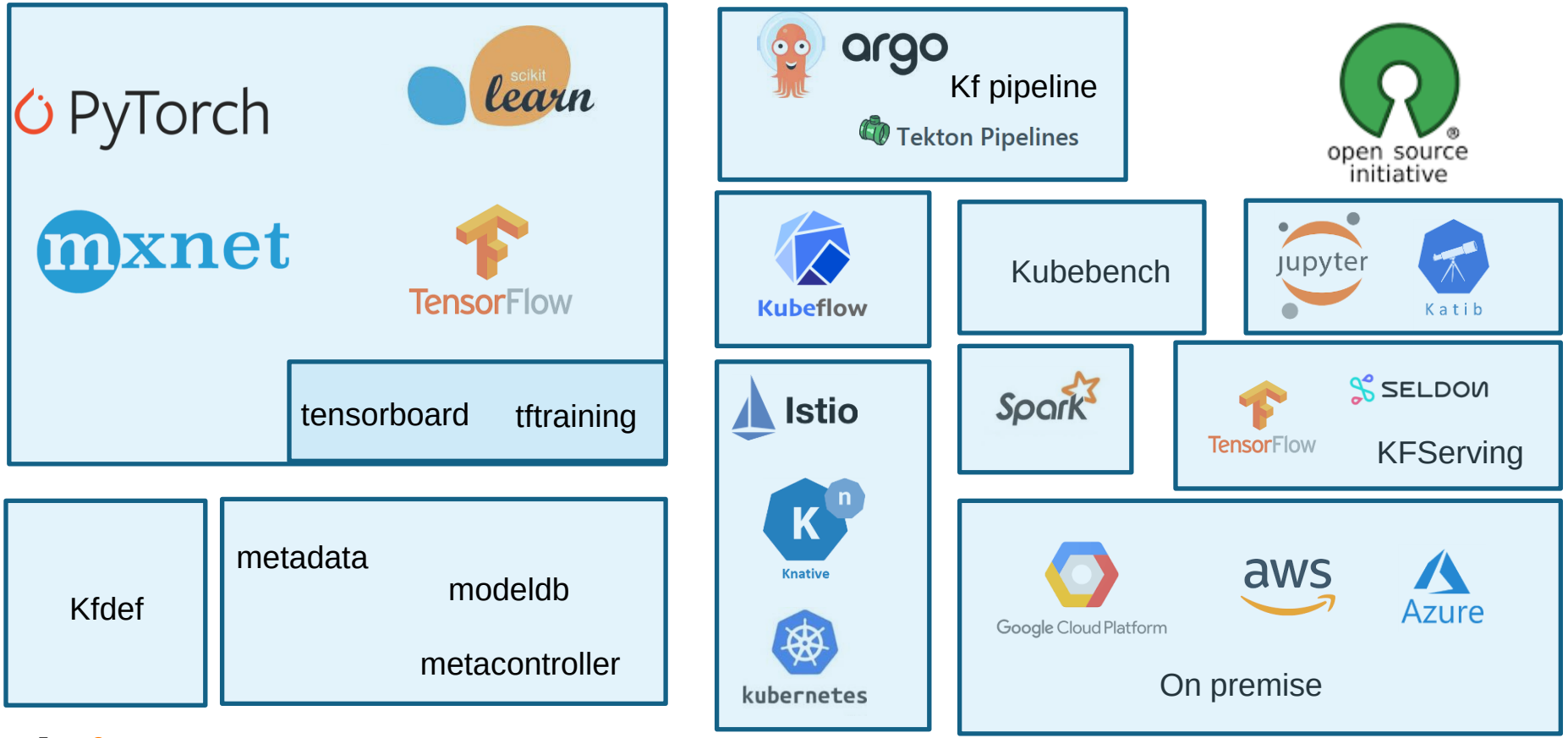
Kubernetes : what ? where ?

- Tasks scheduling
- Automated deployments and upgrades
- Self healing
- Metrics & logs collection
- Security
- …

Anywhere you are running Kubernetes, you should be able to run Kubeflow :

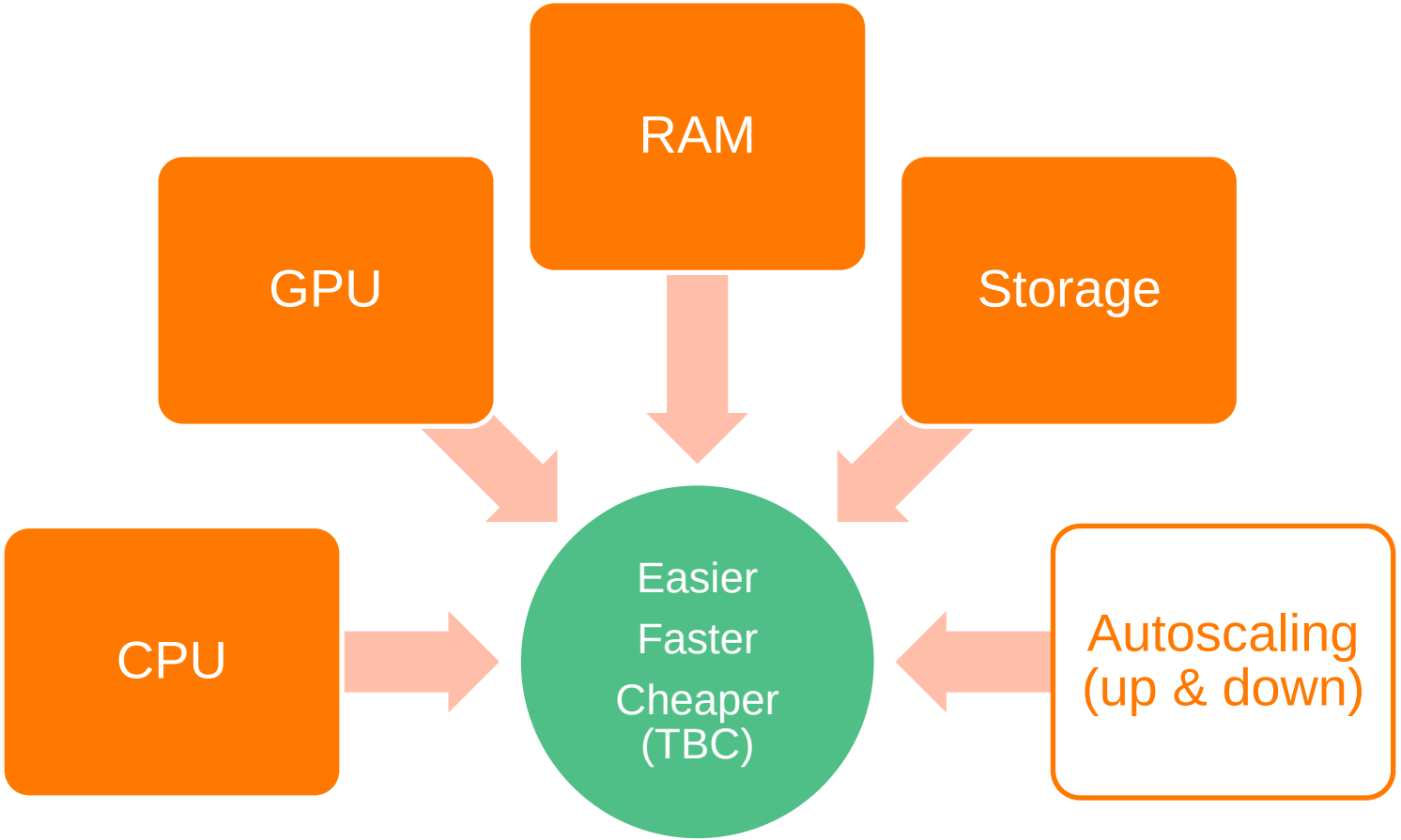
- Public clouds     
- Private clouds 

Microservices : main components Kubeflow in 01/2020

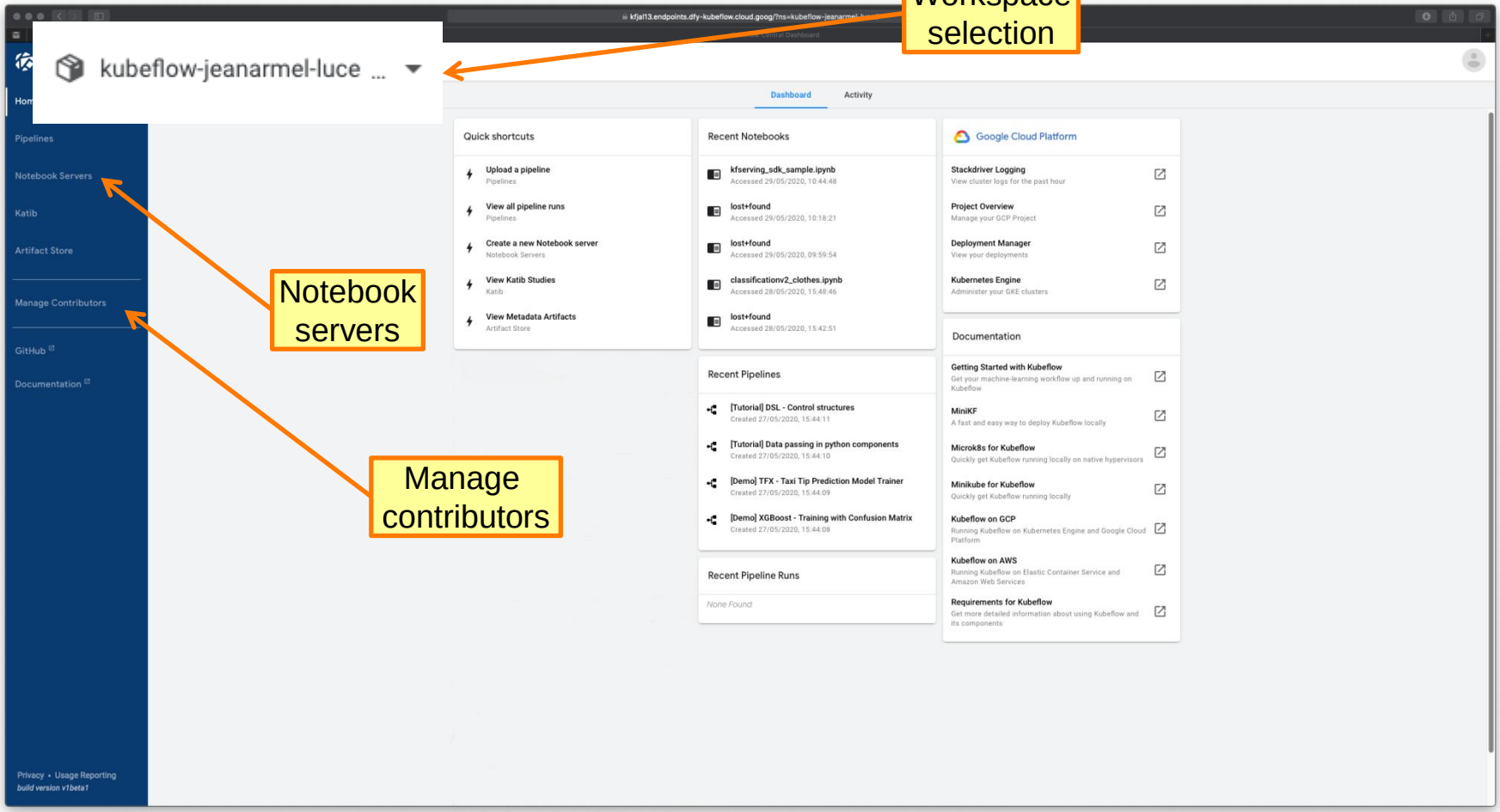


And a few others …

Serverless : access to performant hardware

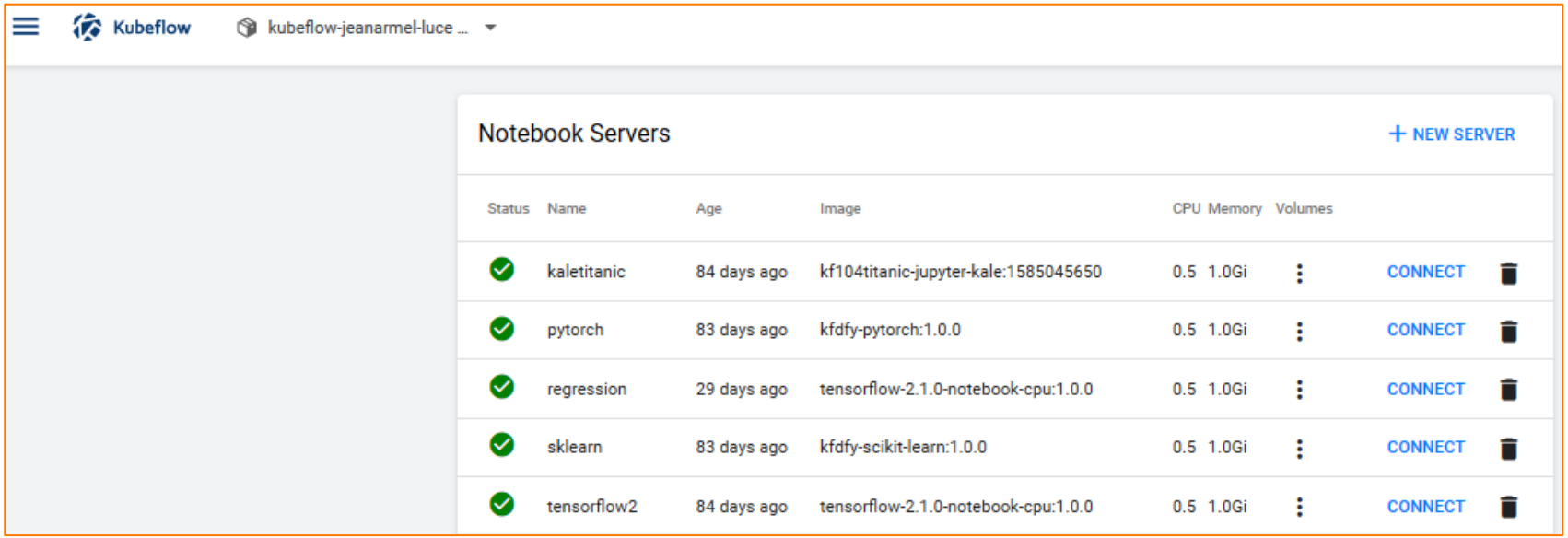


Kubeflow home + working storage



Jupyter notebooks list

A user may have multiple notebooks in a workspace …

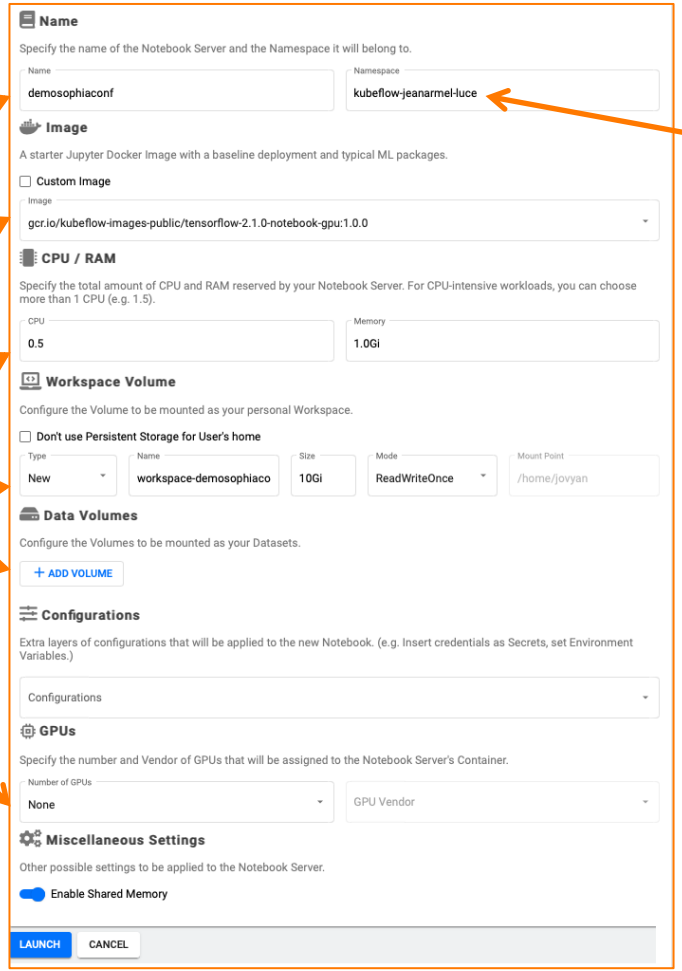


Jupyter notebook creation

A user may have several notebooks.
A name is required to distinguish each of them

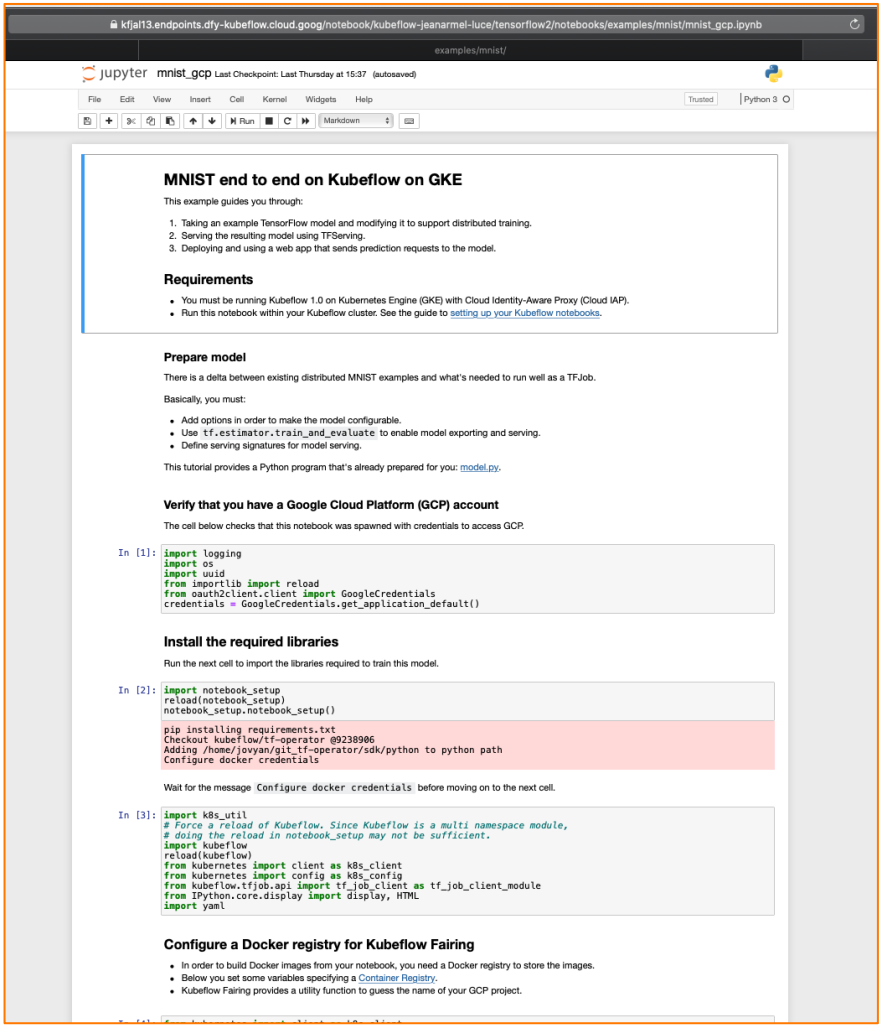
Image « Docker »

Resources (CPU, GPU, RAM, storage)



Workspace

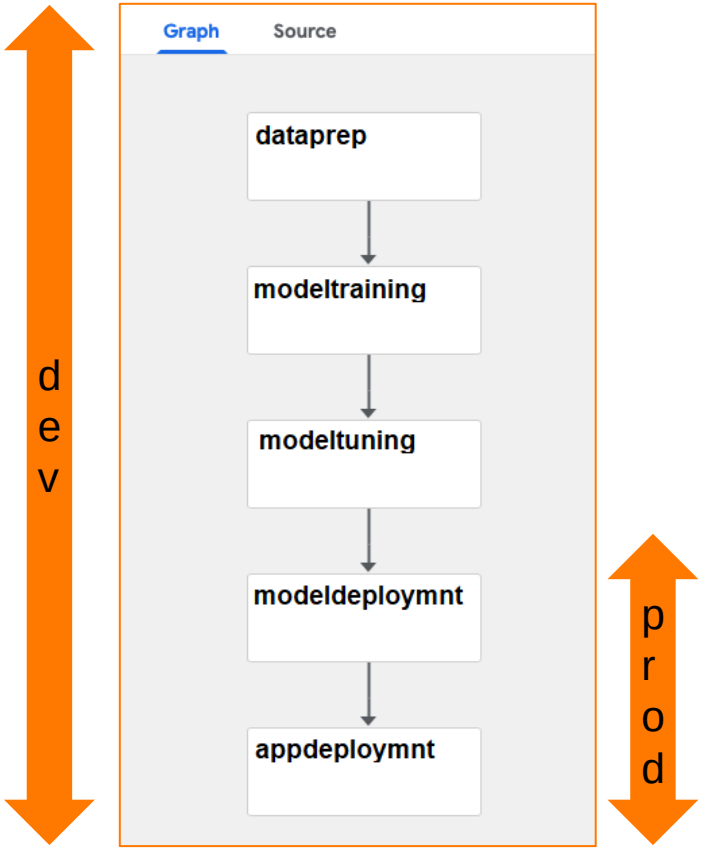
Jupyter notebook



Pipelines

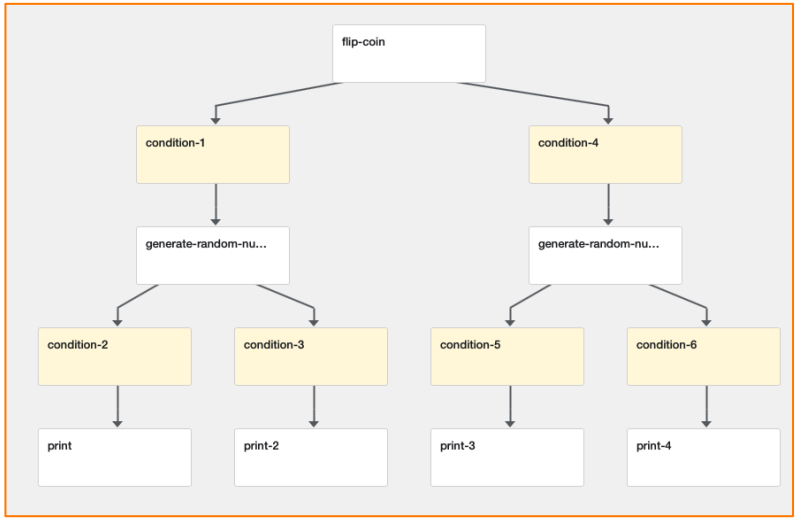
A *pipeline* is a description of an ML workflow, including all of the *components* in the workflow and how they combine in the form of a graph. (See the screenshot below showing an example of a pipeline graph). The pipeline includes the definition of the inputs (parameters) required to run the pipeline and the inputs and outputs of each component.

A *pipeline component* is a self-contained set of user code, packaged as a Docker image, that performs one step in the pipeline. For example, a component can be responsible for data preprocessing, data transformation, model training, and so on.

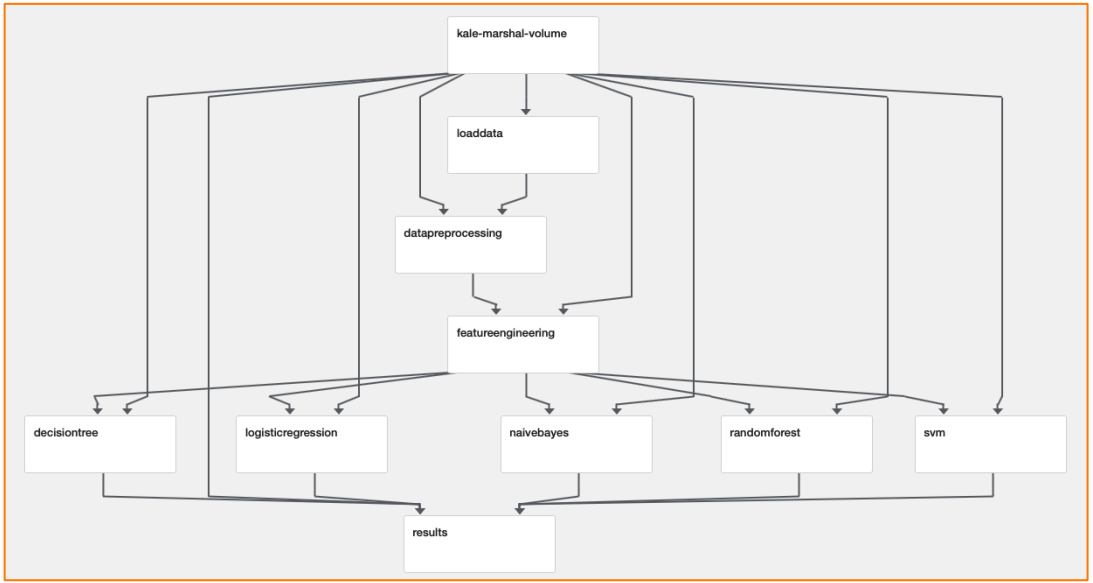


Pipelines – Conditions & parallelization

Conditions



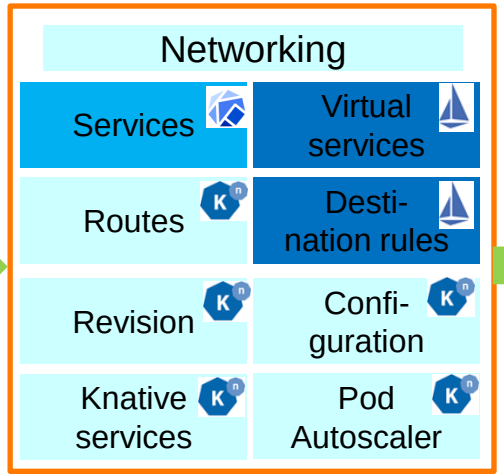
Parallelization



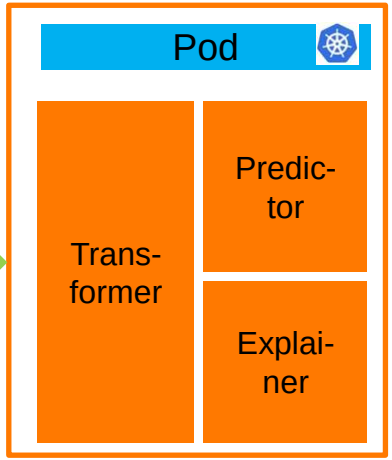
KFServing

- Inference service

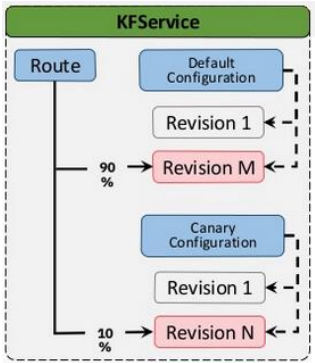
```
apiVersion: "serving.kubeflow.org/v1alpha2"
kind: "InferenceService"
metadata:
  name: "flowers-sample"
spec:
  default:
    predictor:
      tensorflow:
        storageUri: "gs://kfserving-samples/models/tensorflow/flowers"
   canaryTrafficPercent: 10
    canary:
      predictor:
        # 10% of traffic is sent to this model
        tensorflow:
          storageUri: "gs://kfserving-samples/models/tensorflow/flowers-2"
```



- Transformer + Explainer

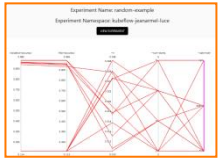


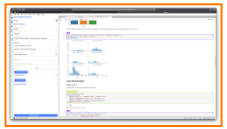
- Canary deployment

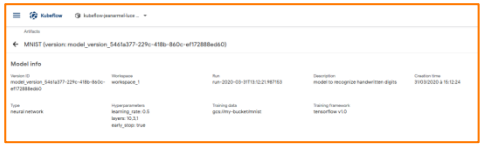


Other microservices or libraries

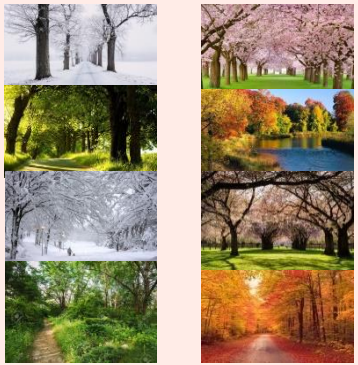
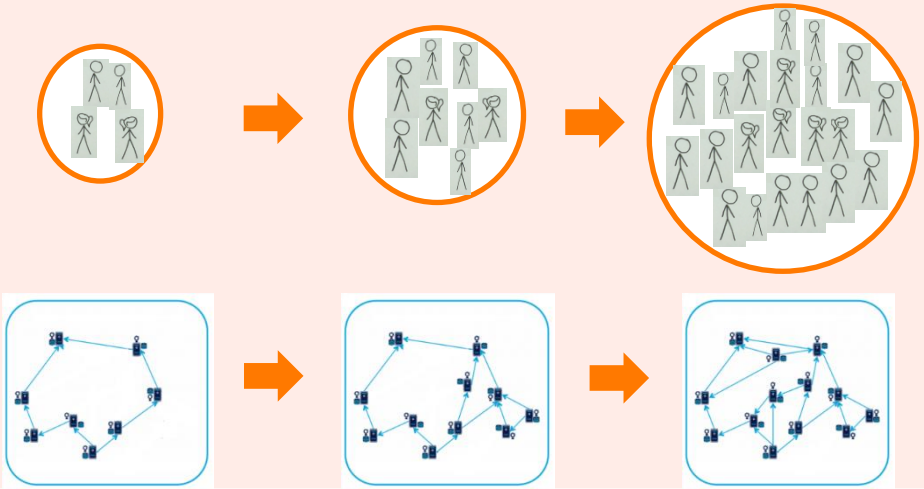
- Katib : hyperparameters tuning
- Kale : generation of pipelines
- Metadata : tracking and managing metadata of ML workflows
- etc …







Kubeflow – Main dates

When	What
December 2017	Kubeflow was open sourced at Kubecon USA
	growth 
February 2020	Kubeflow 1.0 was announced to the public

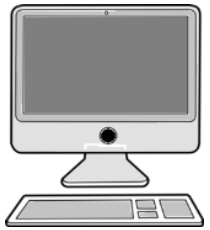
Kubeflow versioning

- Kubeflow 1.0 published in Q1 2020
- Kubeflow embeds several applications with different levels of maturity
  - Each application has its own version id
- Starting from the release of Kubeflow v1.0, the Kubeflow community attributes *stable status* to those applications and components that meet a defined level of stability, supportability, and upgradability.

Give it a try !!! The easiest ways :

- Installation in a cloud. 3 ways :
  1. kfctl : is the control plane for deploying and managing Kubeflow ➔ deployment in ~30 minutes (AWS or GCP)
  2. operator is currently in incubation phase
  3. MiniKF (see below)

- Installation in your laptop : MiniKF
  - Installation via vagrant + virtualbox
  - Comes with : Minikube + Rok + Kubeflow
  - ⚠ System requirements : 12GB RAM, 2CPUs, 50GB disk space mini



Kubeflow main actors / users

THE APACHE SOFTWARE FOUNDATION

Conclusions

- Faster, easier, cheaper, secure ➔ better TTM
- Provides applications that users love for training phase (notebooks, katib, …) or serving (A/B testing, explainer, transformers, …)
- Automation
- Very active open source community
- Still young
- Requires some skills (K8S) for maintenance

Q & A


obrigado
Dank U
Merci
mahalo
Köszi
спасибо
Grazie
Thank you
mauruuru
Takk
Gracias
Dziękuję
Děkuju
danke
Kiitos