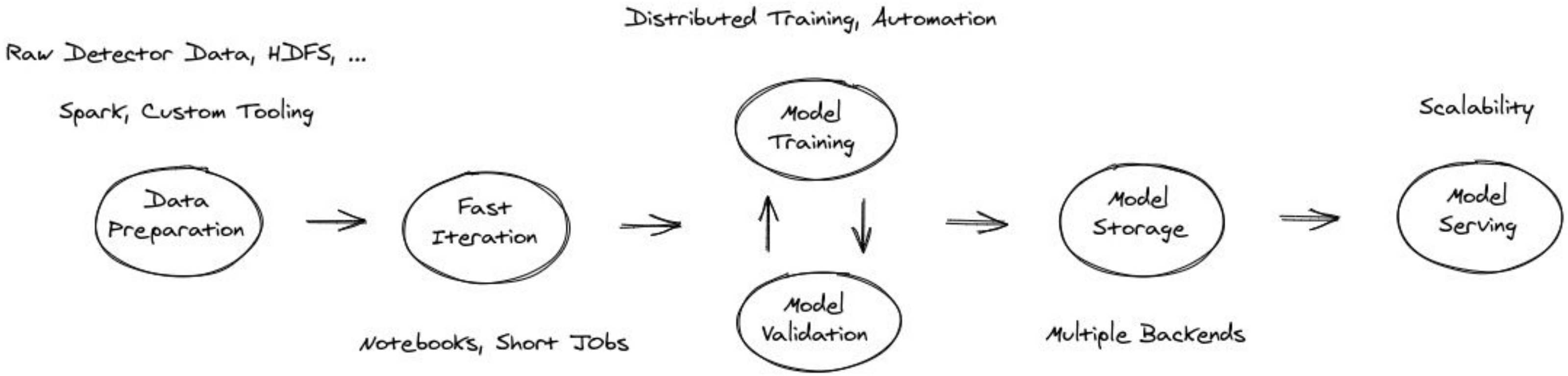
# Scalable Machine Learning at CERN with Kubeflow

## Preparation, Training and Model Serving

Dejan Golubovic, Ricardo Rocha
CERN IT-PW-PI

# Motivation

Offer a platform to manage the full machine learning lifecycle



# Motivation

Ensure **efficient usage** of our

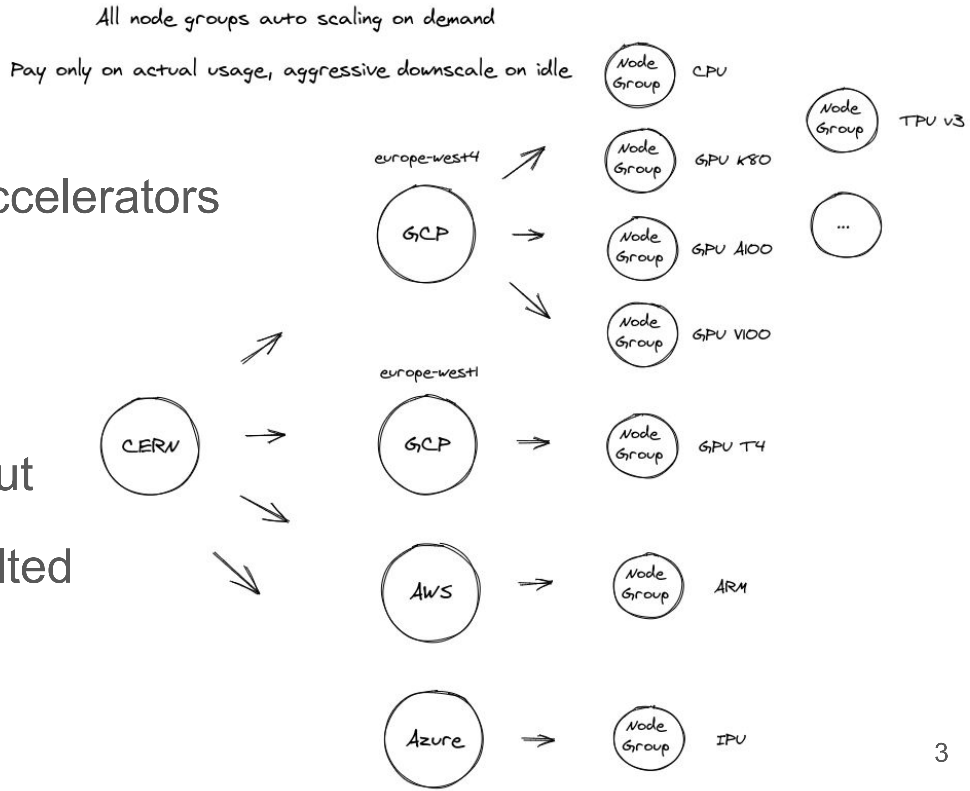
on premises GPU resources

Provide easy access to **public cloud** accelerators

(**GPUs**, **TPUs**, IPUs, FPGAs, …)

Bursting setup already demonstrated, but

access to resources temporarily halted



# Infrastructure

Based on **Kubeflow**, the machine learning toolkit for Kubernetes

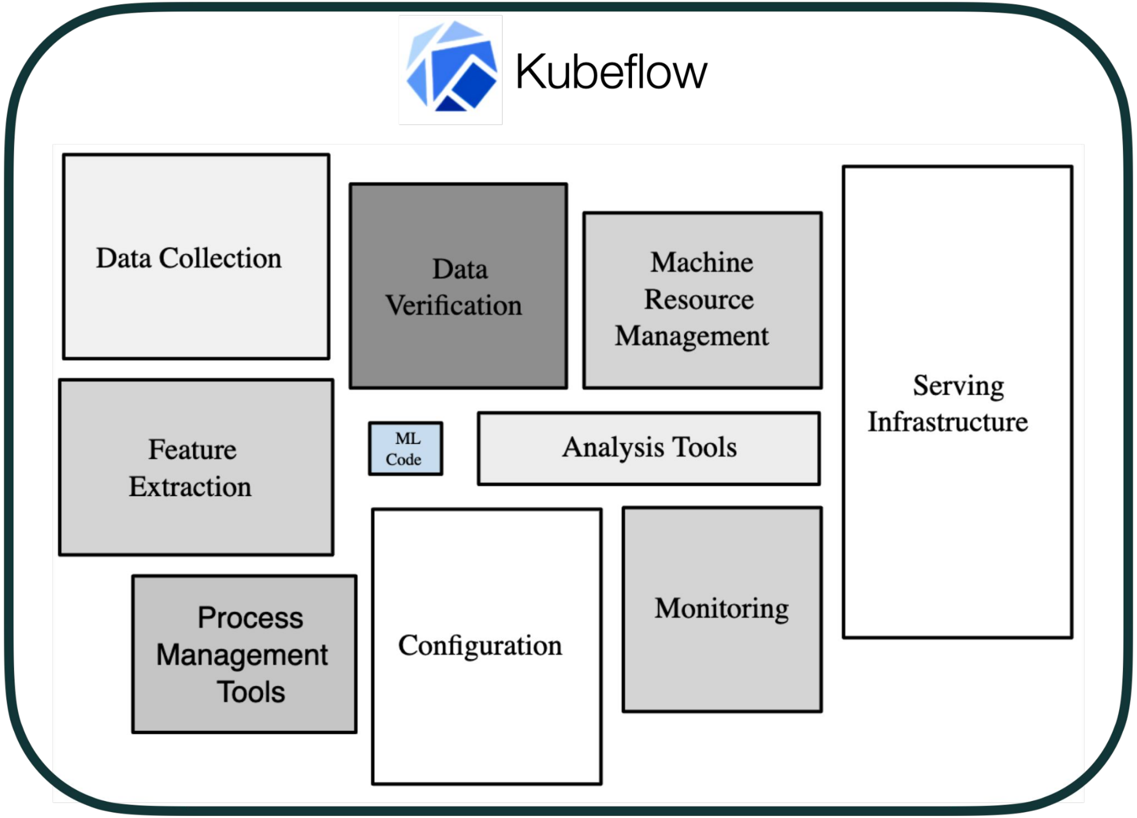
Open source project started by Google in 2017

Declarative API, Operators, Auto healing, Application and Cluster auto scaling

Support for most common frameworks (**TensorFlow**, **PyTorch**, MXNET, …)

In production at multiple companies

Google, Spotify, Bloomberg, Zillow, Arrikto...


Kubeflow
Data Collection
Data Verification
Machine Resource Management
Serving Infrastructure
Feature Extraction
ML Code
Analysis Tools
Process Management Tools
Configuration
Monitoring

# Kubeflow Components and Features

Notebooks

Machine Learning Pipelines

AutoML - Hyperparameter Optimization

Distributed Training

Tensorboards

Model Serving

# Notebooks

Easiest way to **start experimenting** with Kubeflow

Integration with other Kubeflow components

Pipelines, distributed training, inference, AutoML

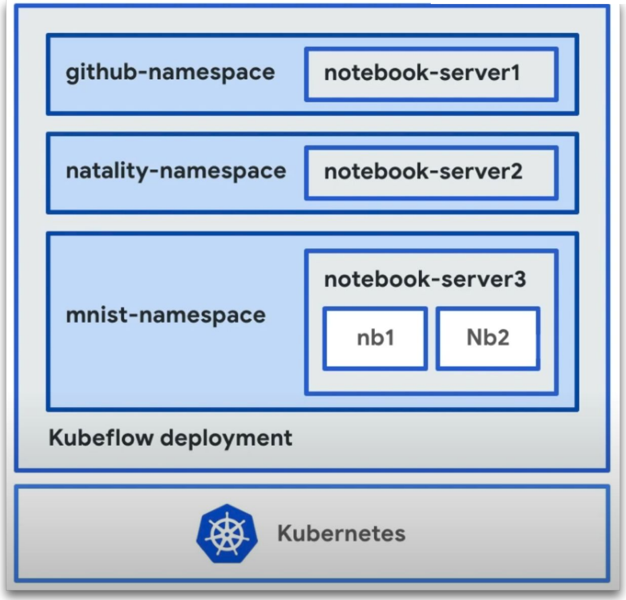
Ability to **customize Python environment**

Or use prebuilt images (Tensorflow, Pytorch)

Select resources (CPU, MEM, GPU)

Pause notebook servers if idle for too long

Good for **experimentation and prototyping phase**



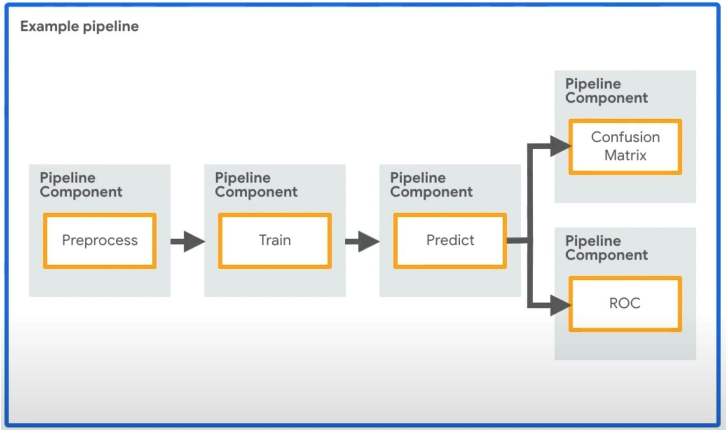
# Machine Learning Pipelines

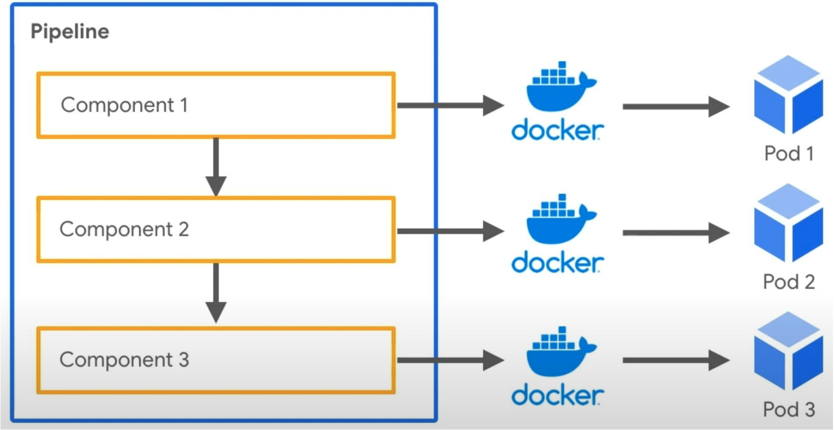
Automated ML workflows

A **user interface** (UI) for managing and tracking experiments, jobs, and runs

An **engine** for scheduling multi-step ML workflows

An **SDK/API** for defining and compiling pipelines and components



# Benefits of Machine Learning Pipelines

Clear isolation between components

Can be scheduled to run periodically

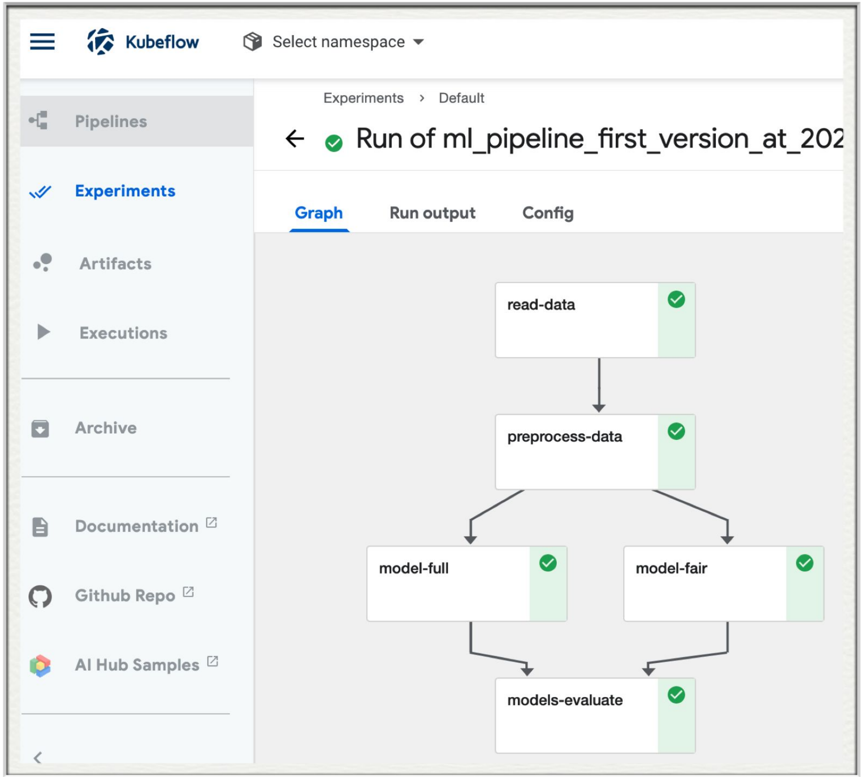
Can run with different input parameters

Versioning

Parallelisation

**Non-blocking GPU access**

**Remote submissions** with a client SDK



# AutoML (Katib) - Hyperparameter Optimization

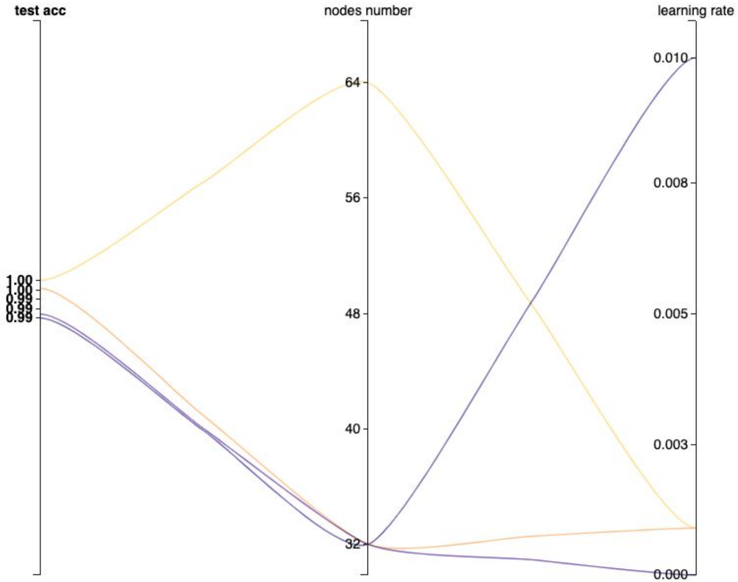
Standardized **development process**

- Create a training script
- Build a Docker image
- Run with various sets of inputs

Improved **hardware efficiency**

- Run each trial on a separate GPU

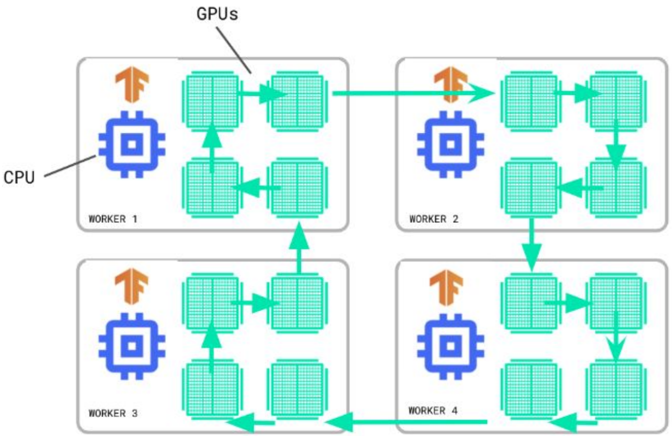
Visualization of **results and metrics**




OVERVIEW TRIALS DETAILS YAML

| Trial name | Status | Test acc | Nodes number | Learning rate |
|---|---|---|---|---|
| test-wze6q-brnmwpbc | Succeeded | 0.99593 | 32 | 0.001 |
| test-wze6q-kqvz9zcp | Succeeded | 0.98831 | 32 | 0.01 |
| test-wze6q-lnbhttzs | Succeeded | 0.98932 | 32 | 0.0001 |
| test-wze6q-qvhz6pm8 | Succeeded | 0.99796 | 64 | 0.001 |

# Distributed Training



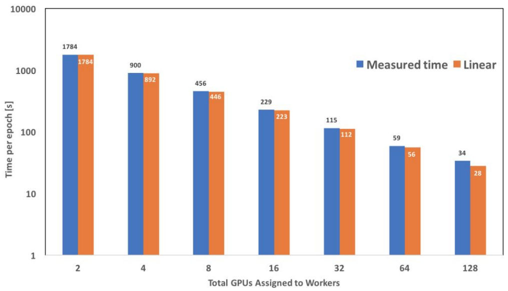

Major ML frameworks support distributed training

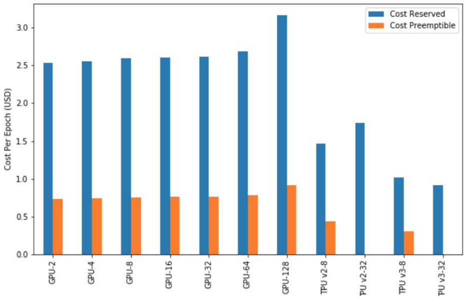
Training jobs split across multiple **local GPUs**

Kubeflow offers distributed training in Kubernetes

TFJob, PytorchJob, MXNetJob, MPIJob, XGBoostJob

Jobs split across multiple **cluster GPUs**




**Accelerating GAN training using highly parallel hardware on public cloud, CHEP 2021**
https://doi.org/10.1051/epjconf/202125102073

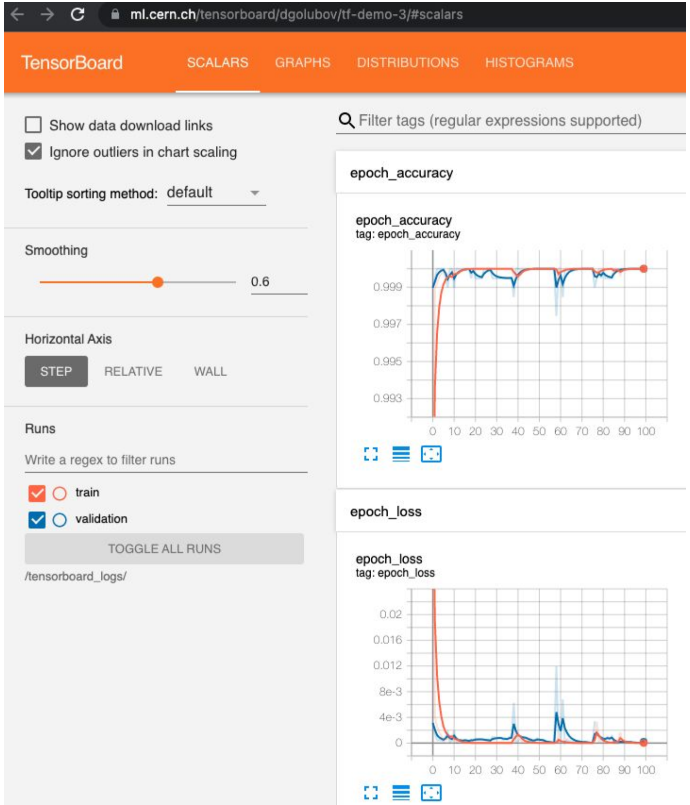
# Tensorboards

Measurements, visualizations for ML workloads

- Track **loss and accuracy**
- Visualize **model graph**
- View custom metrics

Kubeflow allows creation of **Tensorboard servers**

- Monitor model training real-time
- Training from any Kubeflow component



# Model Serving

Deploy a server to **run inference via http requests**

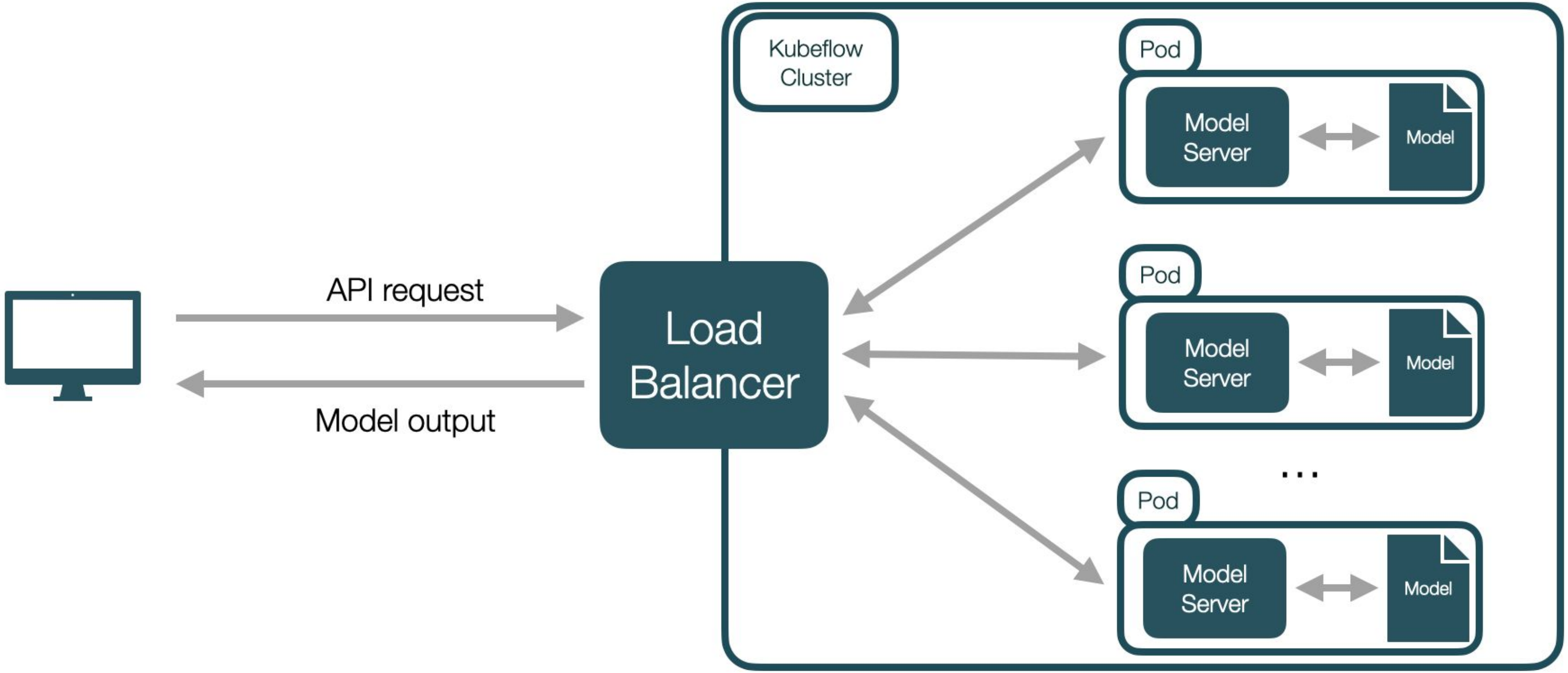
*curl -v -H “Host: host” “http://host_ip/v1/models/mnist:predict" -d @./input.json*

**Serverless architecture**

Automatic scaling per number of requests

Provided via **KServe** component

# Model Serving



# Resource Management

Resources assigned **per profile**

- Memory, CPU, GPU, Kubernetes resources…
- Kubernetes ***ResourceQuota*** for each profile

Personal profiles have a quota of **1 GPU** by default

Quotas for **group profiles** can be increased

- For now by contacting us directly
- Soon via dedicated ServiceNow form

```
apiVersion: v1
kind: ResourceQuota
metadata:
  name: kf-resource-quota
  namespace: dgolubov
status:
  hard:
    limits.cpu: "5"
    limits.memory: 10Gi
    limits.nvidia.com/gpu: "1"
    (soon) limits.nvidia.com/gpu.shared: "1"
    requests.cpu: "5"
    requests.memory: 10Gi
    requests.nvidia.com/gpu: "1"
    (soon) requests.nvidia.com/gpu.shared: "1"
```

# Resource Management - Upcoming

**Time-sliced** NVIDIA GPUs

**Multiple pods on a single GPU**, time sharing

Smaller workloads, ex. notebooks with infrequent GPU utilization

**Physically sliced** NVIDIA GPUs

**GPU memory physically split, full isolation**

Medium to large workloads that require constant GPU access

A ServiceNow form for requesting resources for group profiles

# Storage Integration

**EOS** supported with Kerberos authentication (OAuth2 soon)

**CVMFS** via CSI

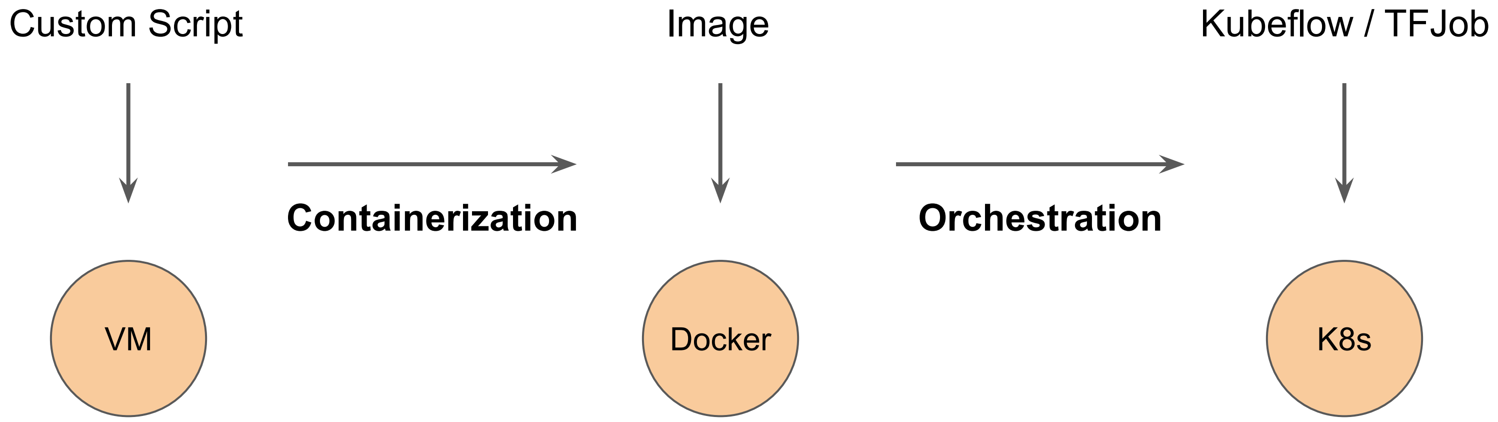
**S3** object storage supported via S3 clients authentication

s3.cern.ch or public cloud providers (Amazon S3, Google Object Storage…)

**registry.cern.ch** - registry for the built images

# Showcase

From a custom script to a large distributed training…



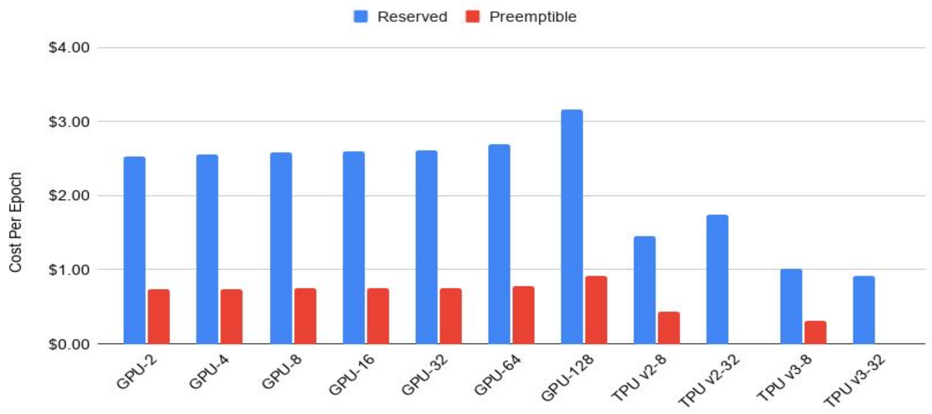
# Showcase

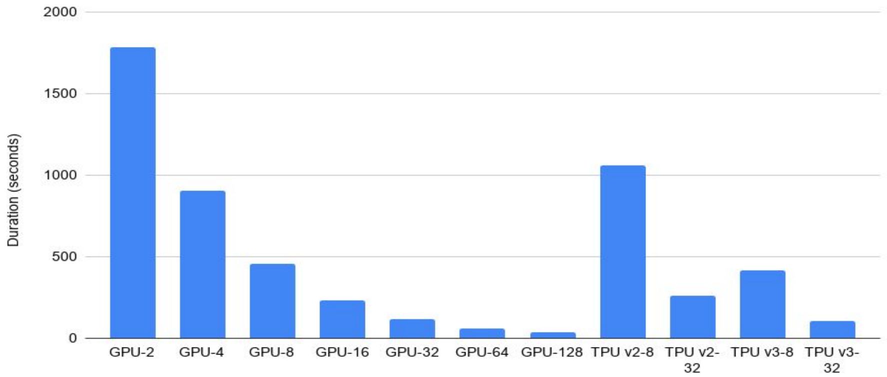
1 to 128 GPUs

3550 to 35 seconds per epoch

x100 speedup

Almost the same total cost

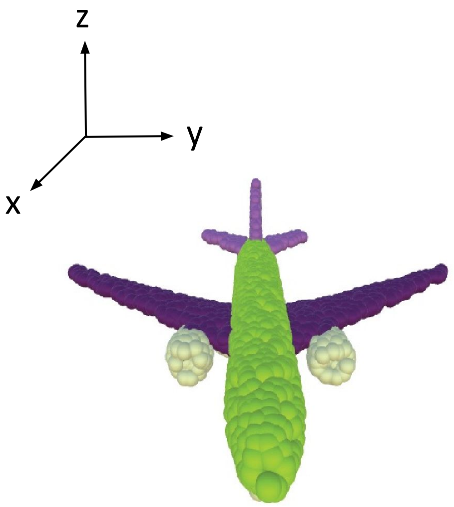


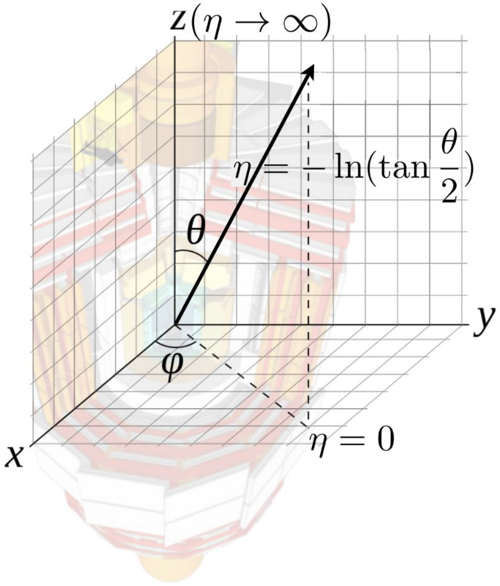



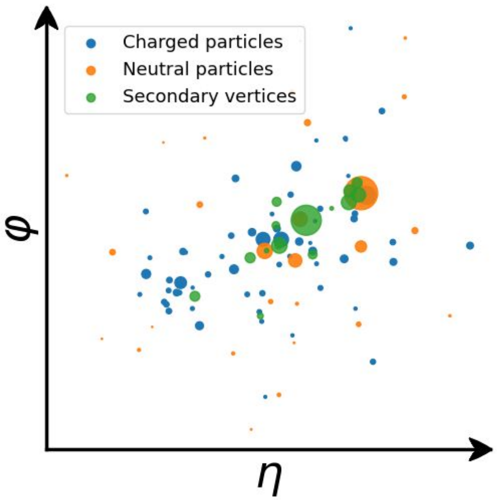
# Use Case - CMS Jet Tagging

Use detector coordinates to represent jets as **particle clouds**

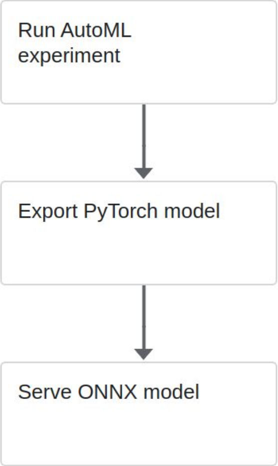
Analogous to **point clouds** in computer vision problems

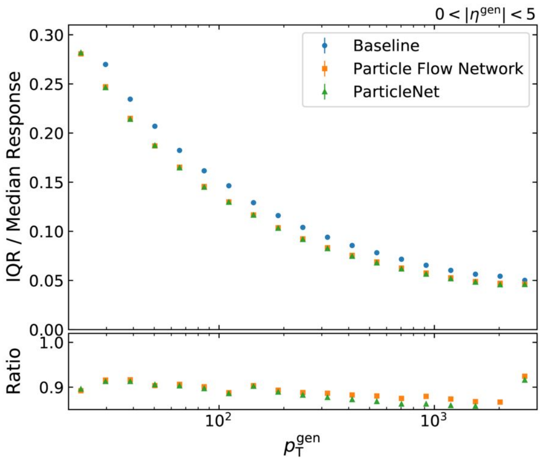


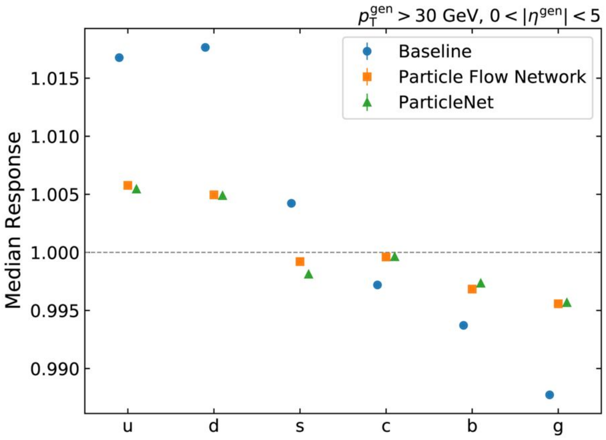



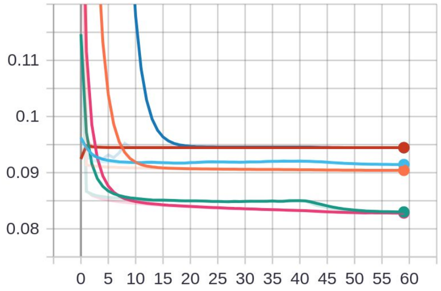



# Use Case - CMS Jet Tagging



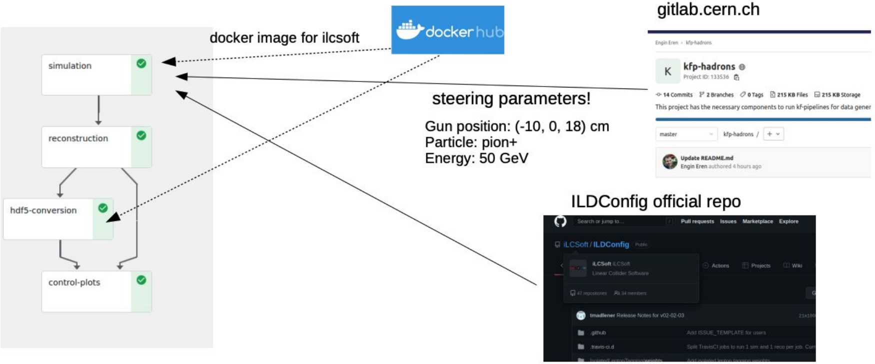

## Model Servers

+ NEW MODEL SERVER

| Status | Name | Age | Predictor | Runtime | Protocol | Storage URI |
|---|---|---|---|---|---|---|
| ✓ | particle-net-regressor-... | 1 month ago | Triton | 21.09-py3 | | s3://jec-data/particle-net-regressor-25a03c |
| ✓ | pfn-regressor-ea37f4 | 2 months ago | Triton | 21.09-py3 | | s3://jec-data/pfn-regressor-ea37f4 |

# Use Case - 3DGAN Desy

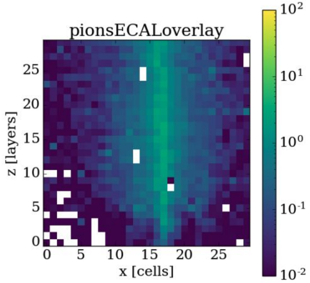
Data pipeline and experiment

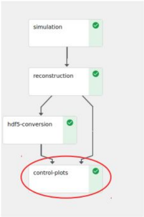




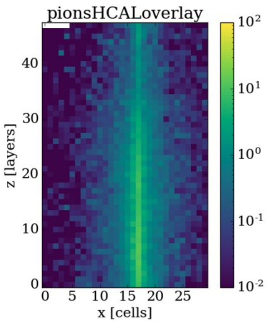
Writing all hdf5 files to CERN-EOS

**ECAL** (z-x)

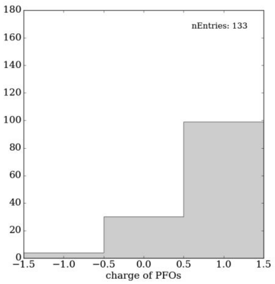


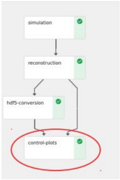




**HCAL** (z-x)

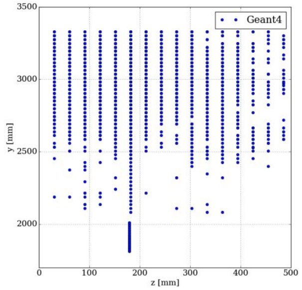



Pandora PFO:




Simulated hits:



# Other Use Cases

OpenLab 3DGAN

AutoML, Distributed Tensorflow Training

UNOSAT

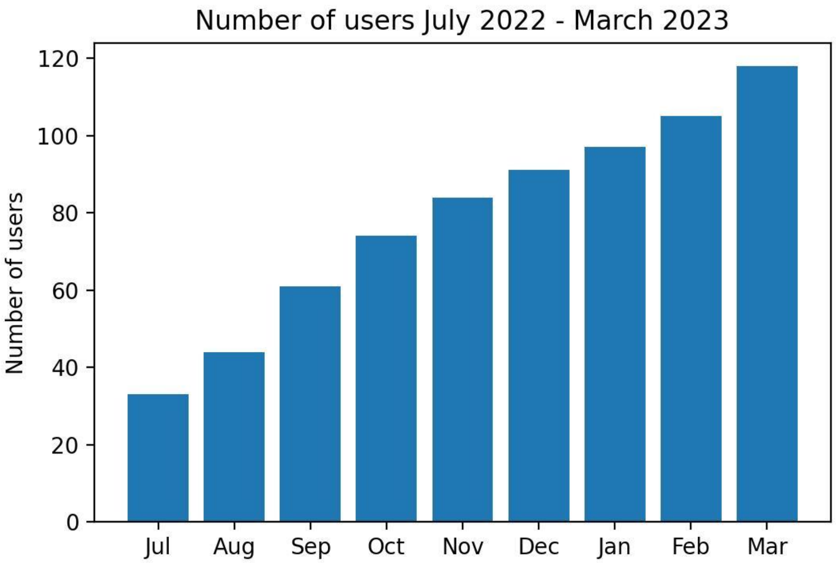
Distributed Pytorch Training, Pipelines

ATLAS Spanet

Pipelines, AutoML, Inference

ADMON Anomaly Detection

Inference Services



# Conclusions

**Community** effort to improve machine learning infrastructure

Kubeflow ongoing active development

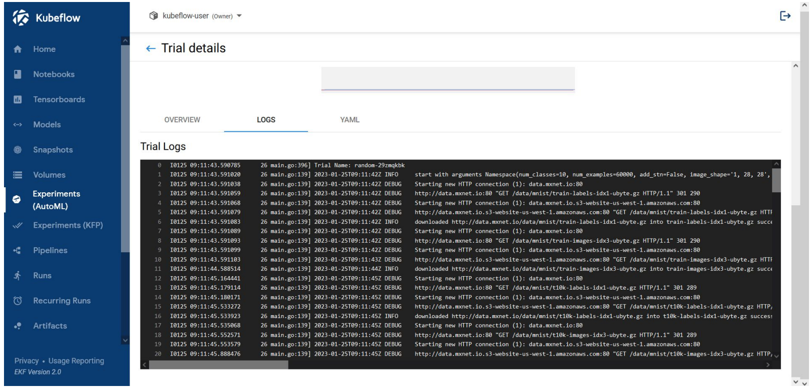
CERN users can influence future developments

High interest in our feedback

Anyone can contribute to open source

[https://github.com/kubeflow](https://github.com/kubeflow)

Everyone is invited to **provide feedback**!



# Previous Talks

KubeCon Europe 2022, May 17 2022

Jet Energy Corrections with GNN Regression using Kubeflow @ CERN

**IT Technical Forum CERN, November 19 2021**

Centralized Management of Your Machine Learning Lifecycle

KubeCon North America 2021, October 12 2021

A Better and More Efficient ML Experience for CERN Users

KubeCon Europe 2021, May 6 2021

Building and Managing a Centralized ML Platform with Kubeflow at CERN

25th International Conference on Computing in High-Energy and Nuclear Physics, May 20 2021

Training and Serving ML workloads with Kubeflow at CERN

Fast Machine Learning for Science Workshop, Dec 01 2020

Making ML Easier with Kubeflow

# Useful Links

https://ml.cern.ch - the service landing page

ml.docs.cern.ch - documentation pages

https://gitlab.cern.ch/ai-ml/examples - examples repository

https://mattermost.web.cern.ch/it-dep/channels/ml - Mattermost channel

For any questions, please write here

Others may benefit from your questions!

# Motivations of users when choosing your service?

Fully integrated platform for the ML lifecycle

- Centralized portal for all functionality
- From interactive development to distributed training to model service

Flexibility in the development / analysis environments

- Users / teams can build and use their own custom images

Availability of GPUs

# Common requests and pain points from end users

Improved transition from notebook to distributed training (Kale, kfp)

- Coming with the new version of kubeflow

Better management of credentials (krb5, oauth2) - coming soon

- Especially for access to storage systems
- Automated renewal critical for long running pipelines

More GPUs!

Improved tooling to debug the jobs themselves

# Positive impact from potential infrastructure changes

Non overcommitted nodes in the underlying openstack layer

- Reliable performance of underlying nodes
- Move to baremetal or no-overcommit virtual machines

Access of public cloud resources (GPUs, TPUs, …)

# What are the key items you are currently working on?

Improvements in debugging tools

Coming with the next service upgrade, PR contribution upstream

Improvements in storage access

Access to EOS with OAuth2 credentials

GPU sharing

Available in the latest kubernetes release, next ml service upgrade

# Current GPU Usage

Overall utilization ~30%

Largely limited by idle notebooks *holding* GPUs

Solution out soon

- Notebooks: time sliced
- Pipelines: full cards/partitions