

Centralized Management of Your Machine Learning Lifecycle

Preparation, Training and Model Serving

Dejan Golubovic, Daniel Holmberg, Ricardo Rocha
CERN

Contents

Motivation

Infrastructure

Kubeflow Features

Demo

Early Adopters Experience

Best Practices for Scalable ML

Conclusion and Next Steps

A bit of history...

First kubeflow trials at CERN started ~2.5 years ago

Helping a few users scale out machine learning use cases

- From a single GPU to distributed training with 10s GPUs

Gradual expansion to a preview service (ongoing)

- Slowly onboarding new users

- Exploring the remaining bits of the ML lifecycle

A bit of history...

First kubeflow trials at CERN started ~2.5 years ago

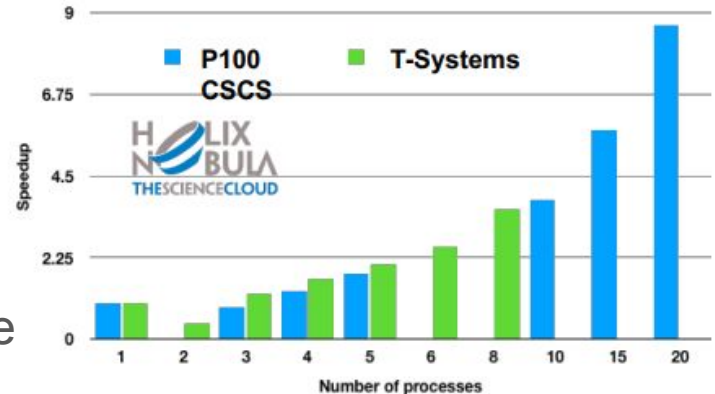
Helping a few users scale out machine learning use cases

From a single GPU to distributed training with 10s GPUs

Gradual expansion to a preview service (ongoing)

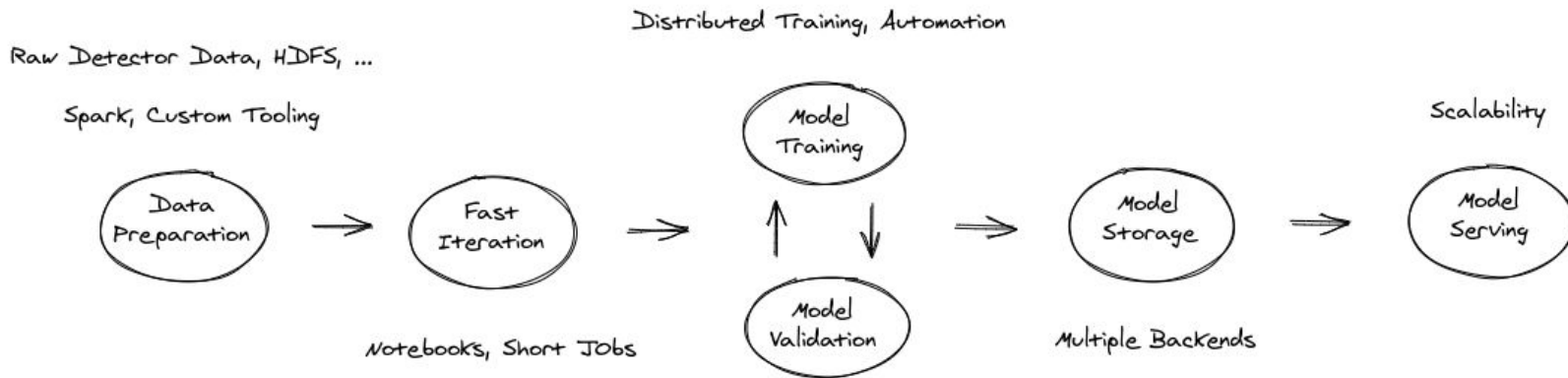
Slowly onboarding new users

Exploring the remaining bits of the ML lifecycle



Motivation

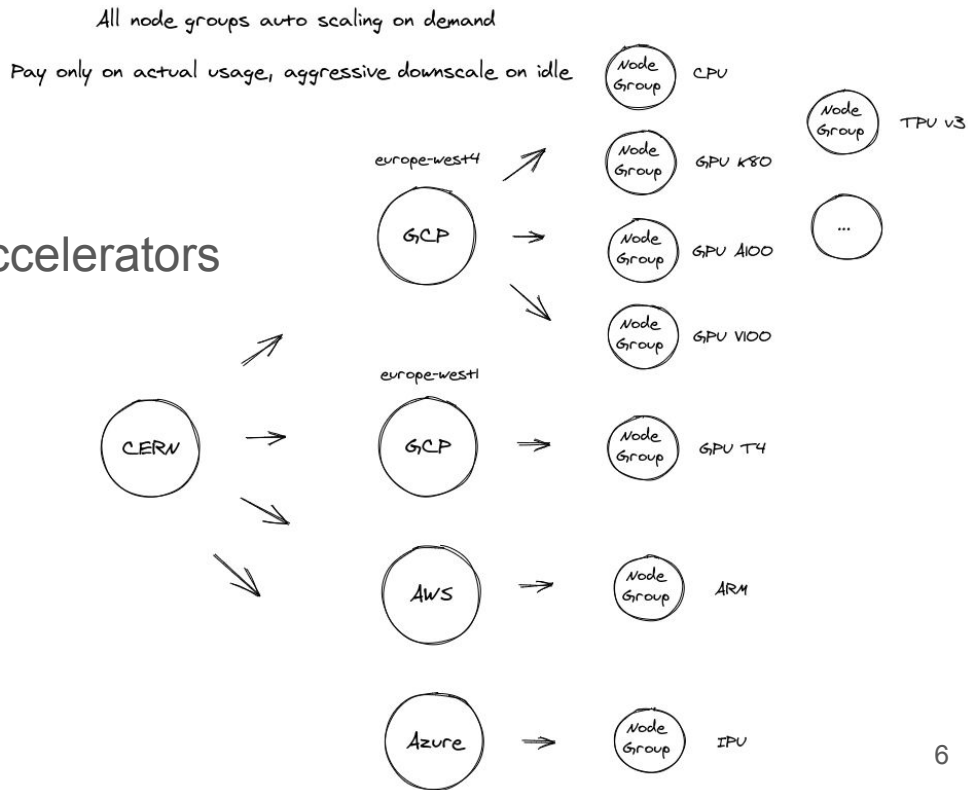
Offer a platform to manage the full machine learning lifecycle



Motivation

Ensure **efficient usage** of our
on premises GPU resources

Provide easy access to **public cloud** accelerators
(**GPUs, TPUs, IPU**s, FPGAs, ...)





Infrastructure

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

Open source project started by Google in 2018

<https://github.com/kubeflow/kubeflow>

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...

Infrastructure

On Premises GPUs: Nvidia V100s and T4s

Full GPU card assignment

Distributed training, hyper parameter optimization, model serving

On Premises vGPUs (soon)

Time sharing of a GPU, up to 4x

Notebooks, code validation, quick iteration

Public Cloud GPUs, TPUs

Available on demand

Not yet fully integrated into the centralized service

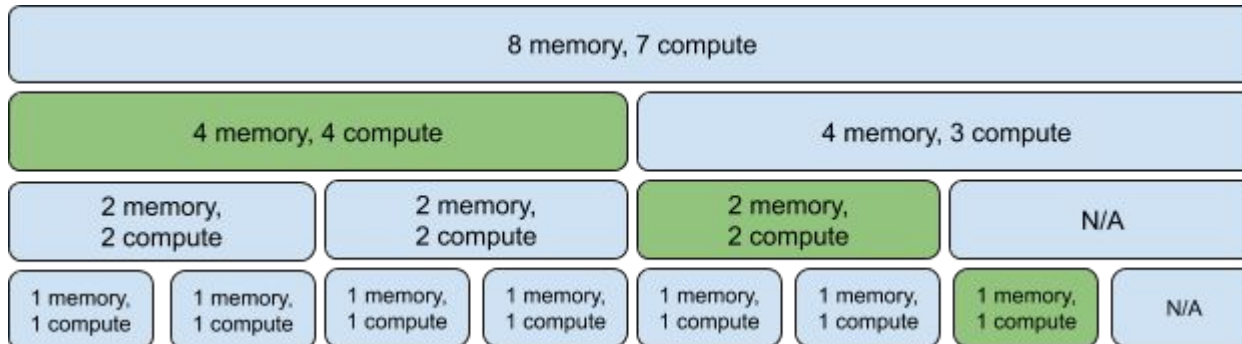
Infrastructure

Expanding soon: Nvidia A100s

Coming to CERN IT beginning of next year

Physical partition of each card up to 7x - Multi Instance GPU (MIG)

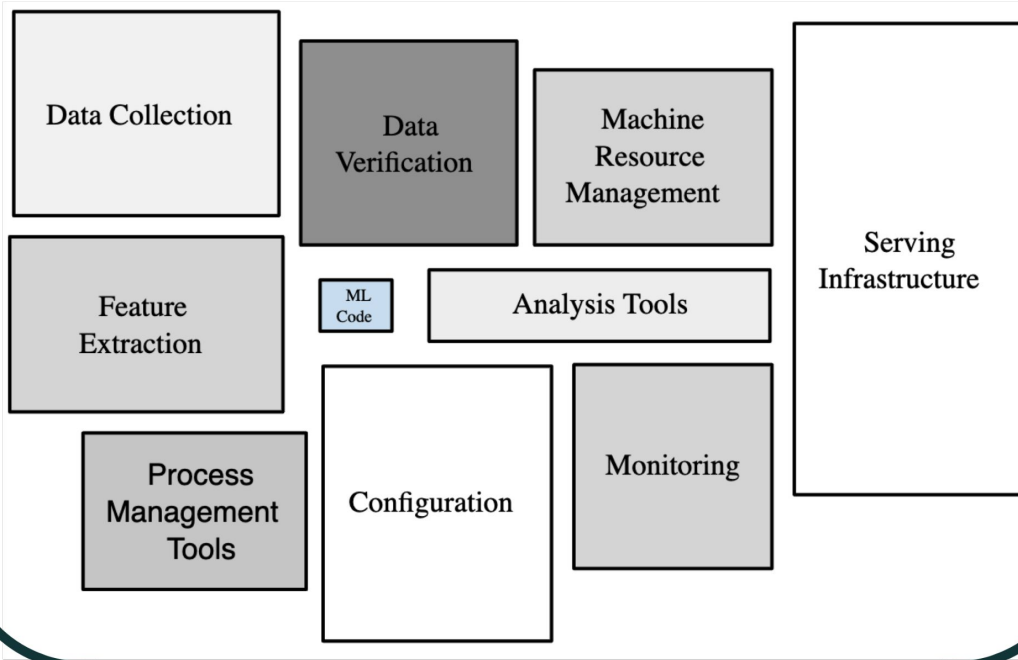
Many other layouts possible



Kubeflow Features



Kubeflow



Kubeflow Components and Features

Jupyter Notebooks



ML Pipelines



Katib - Hyperparameter Optimization

KALE - Notebooks to Pipelines

Distributed Training

Model Serving





Jupyter Notebooks

Easiest way to **start** experimenting with Kubeflow

Integration with other Kubeflow components

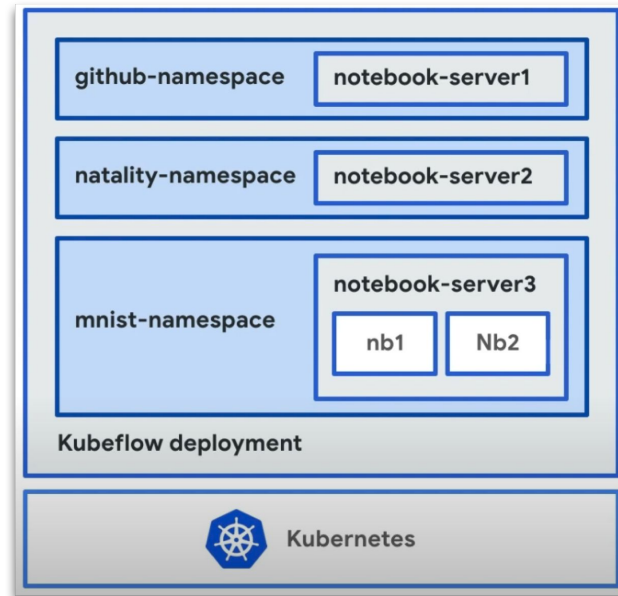
Distributed training, inference services, Katib jobs

Ability to **customize Python environment**

Or use one of prebuilt images (Tensorflow, Pytorch)

Select resources (CPU, MEM, GPU)

Good for **experimentation and prototyping phase**

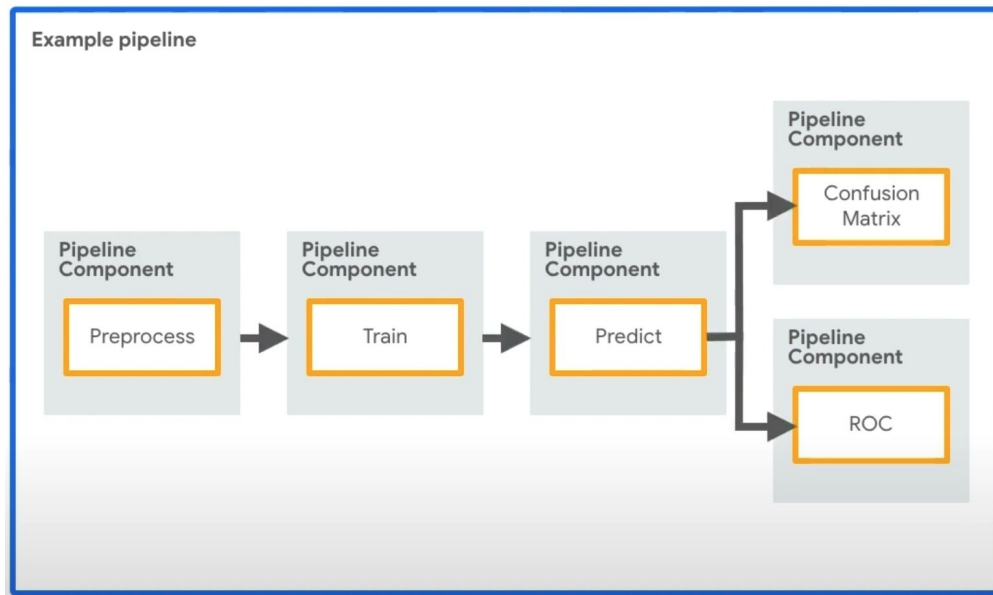




Machine Learning Pipelines

Automated machine learning workflows

Sequence of self-contained and reusable components

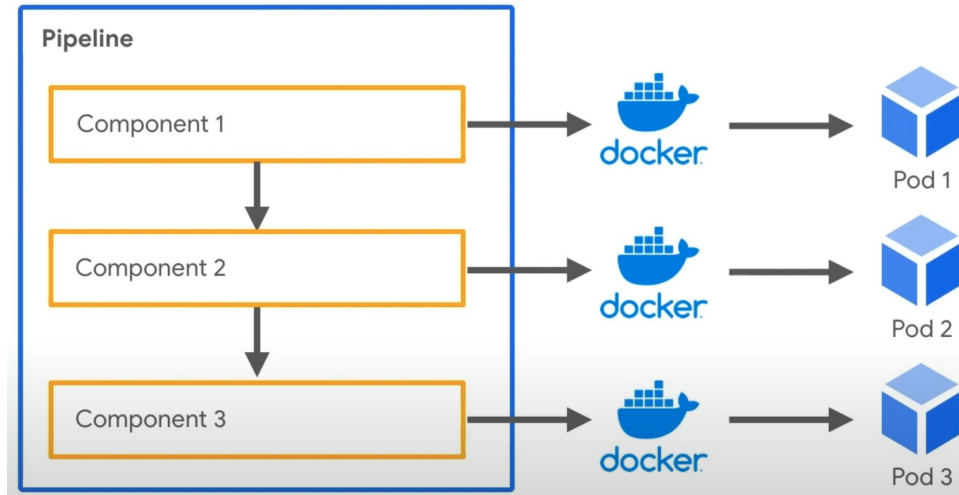


Machine Learning Pipelines

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

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

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



Benefits of Machine Learning Pipelines

Clear isolation between components

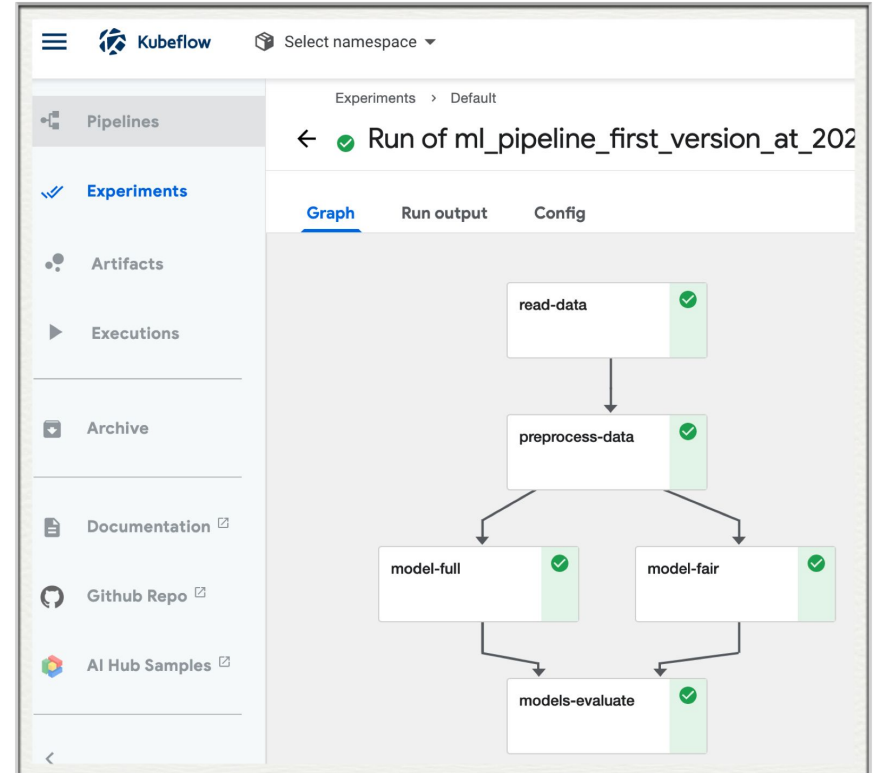
Can be scheduled to run periodically

Can run with different input parameters

Versioning

Parallelisation

Non-blocking GPU access





Katib - Hyperparameter Optimization

Standardized **development process**

Create a training script

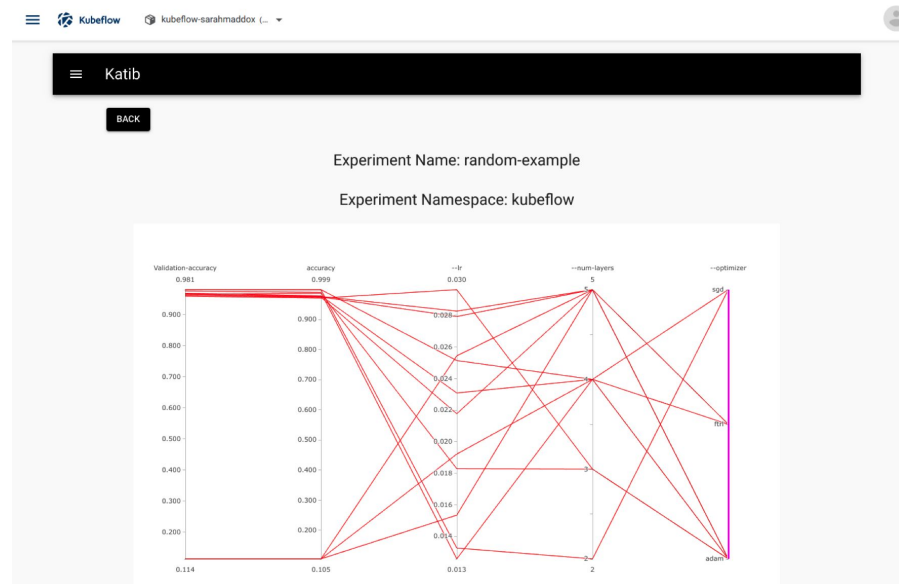
Build a Docker image

Run with various sets of inputs

Improved **hardware efficiency**

Run each trial on a separate GPU

Visualization of **results and metrics**



Distributed Training

Split training jobs across multiple GPUs

TensorFlow and **Pytorch** support distributed training

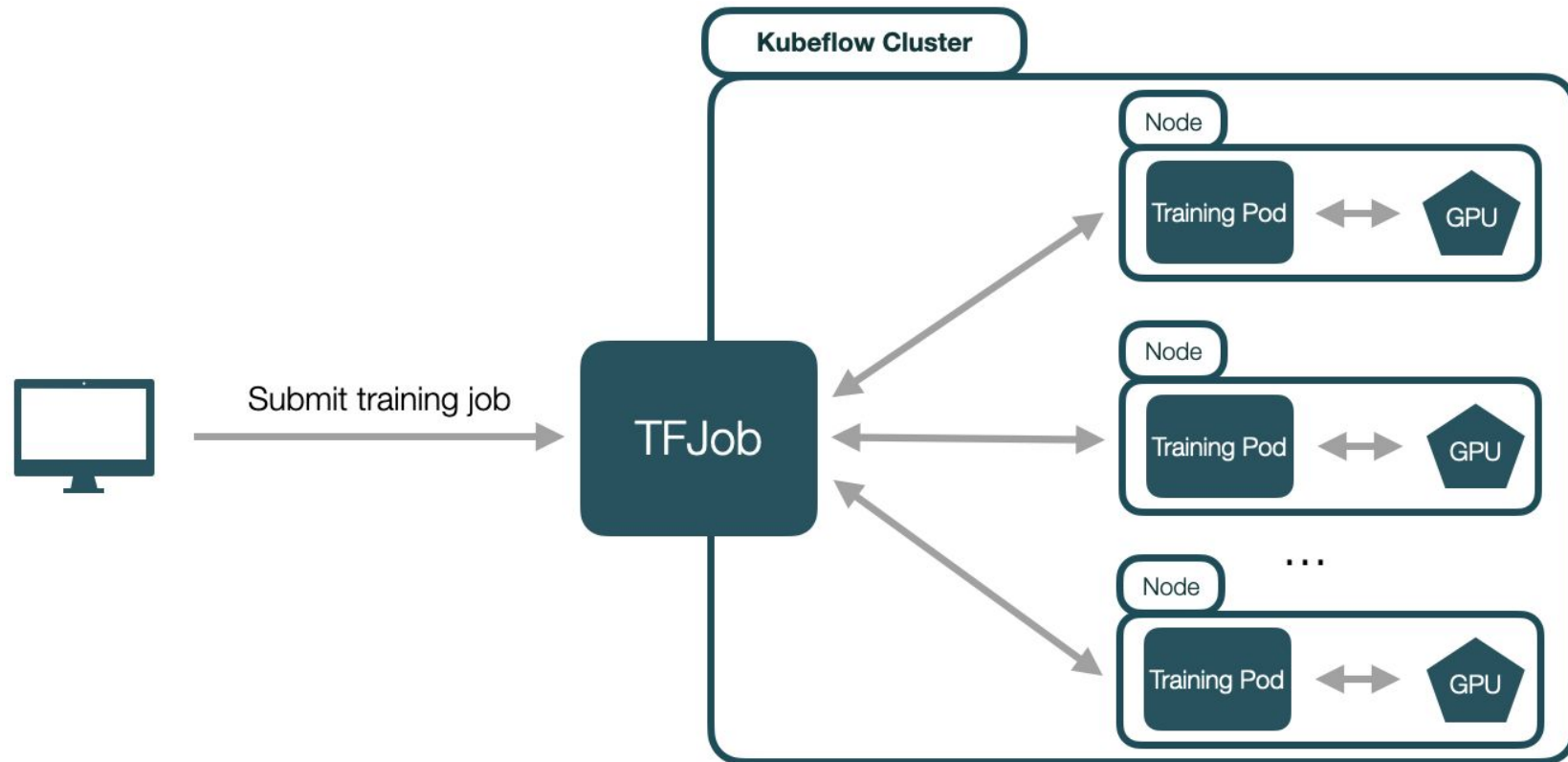
Jobs are split across multiple **local GPUs**

TFJob, PytorchJob - Kubernetes custom resources for distributed training

Jobs are split across multiple **cluster GPUs**

Operators for Tensorflow and Pytorch to support containerised distribution

Distributed GPU Training



TFJob Example

```
apiVersion: kubeflow.org/v1
kind: TFJob
metadata:
  generateName: tfjob
  namespace: your-user-namespace
spec:
  tfReplicaSpecs:
    PS:
      replicas: 1
      restartPolicy: OnFailure
      template:
        metadata:
          annotations:
            sidecar.istio.io/inject: "false"
        spec:
          containers:
            - name: tensorflow
              image: gcr.io/your-project/your-image
              command:
                - python
                - -m
                - trainer.task
                - --batch_size=32
                - --training_steps=1000
```

```
Worker:
  replicas: 3
  restartPolicy: OnFailure
  template:
    metadata:
      annotations:
        sidecar.istio.io/inject: "false"
    spec:
      containers:
        - name: tensorflow
          image: gcr.io/your-project/your-image
          command:
            - python
            - -m
            - trainer.task
            - --batch_size=32
            - ---training_steps=1000
```



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
```

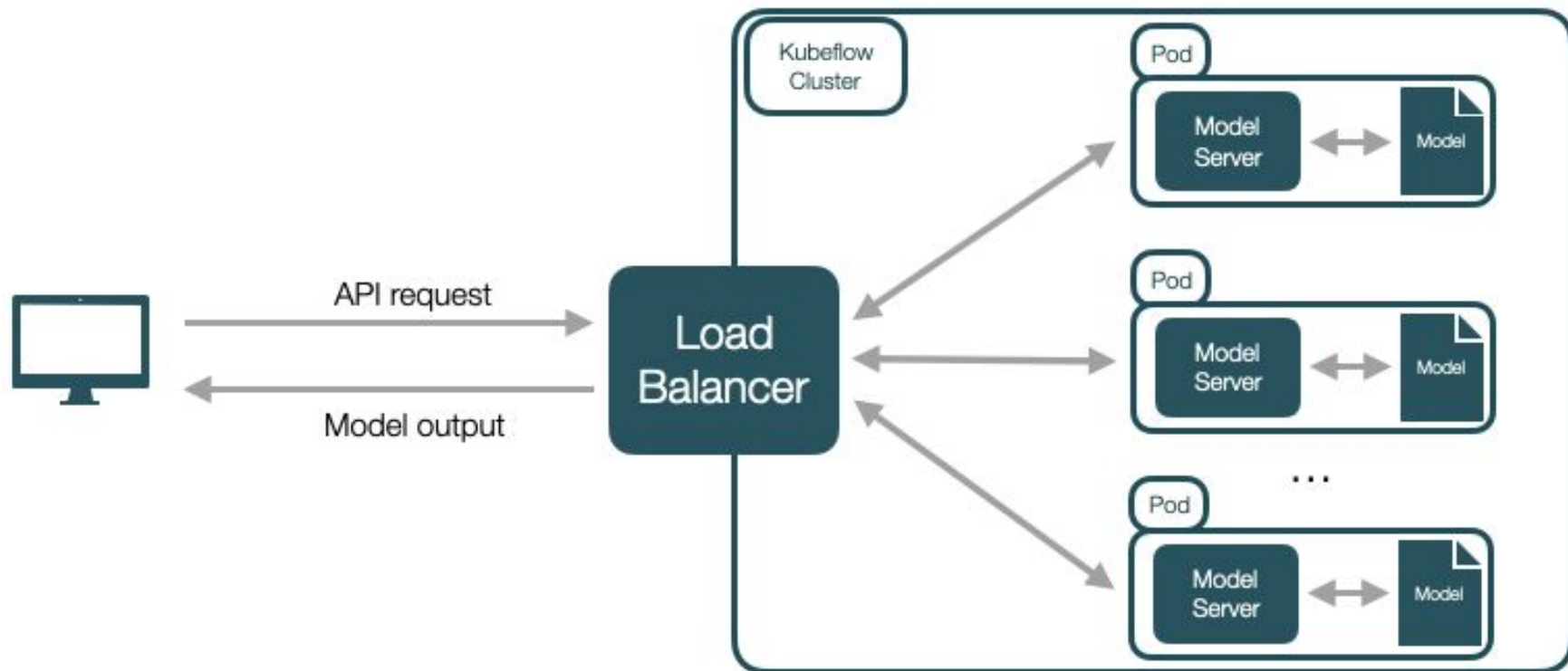
Serverless architecture

Automatic scaling per number of requests

Provided via **KServe** component



Model Serving



Demo

Current Limitations I

Pipelines and pipeline artifacts isolation

Experiments and runs are isolated, but pipelines are not

The fix is coming up, ~3 months

Katib jobs logging

Not there yet, Katib jobs can be monitored using `kubectl`

Automatic credential renewal - coming up in ~1 month

Accessing **cookies for model serving**, currently done manually

Current Limitations II

Running a **Tensorboard server**

Notebook culling not in place yet

Remove unused notebooks periodically

GPU / vGPU availability numbers for notebook creation

No built-in support yet for model versioning, model catalog

Early Adopters Experiences

Early Adopters

OpenLab 3DGAN

UNOSAT

ATLAS Spanet

CMS ParticleNet

ADMON - anomaly detection

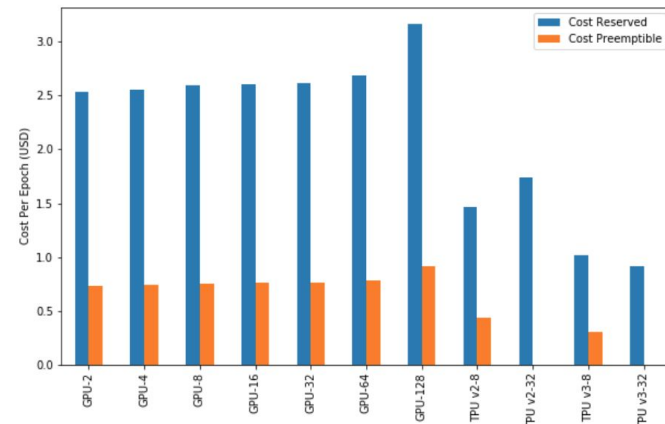
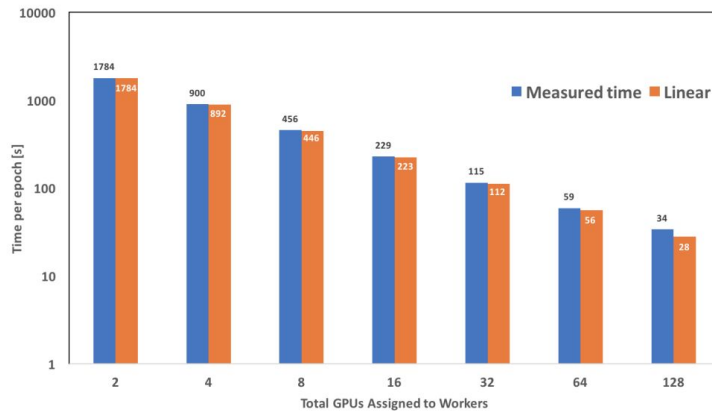
3DGAN OpenLab

Run distributed training with **TFJob** on a **local cluster** and on a **public cloud**

Using up to 128 preemptible **Google Cloud GPUs** and **TPUs**

Resources from the Cloudbank EU project

Particularly well behaved, performance improvement close to linear



Accelerating GAN training using highly parallel hardware on public cloud, CHEP 2021

<https://doi.org/10.1051/epjconf/202125102073>

My experience with Kubeflow | Jona Bossio (CERN)

Context:

- ML project within an ATLAS SUSY search
- **Goal:** assign (uniquely) every resonance particle to a set of jets while preserving symmetries and handling a variable-size set of input jets
- **Plan:** Use [SPANet](#) (Symmetry Preserving Attention Networks) to resolve such assignment problem

How I used ml.cern.ch so far?

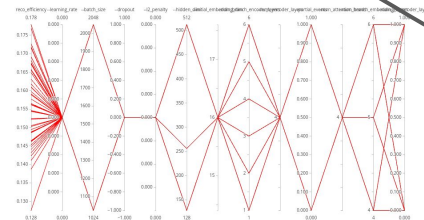
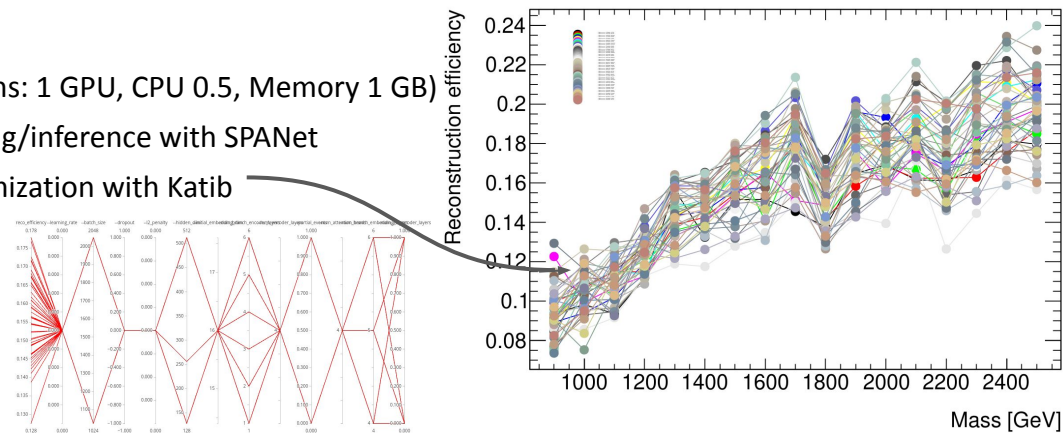
- Jupyter notebook using a custom image (specifications: 1 GPU, CPU 0.5, Memory 1 GB)
 - I used such notebook for training (~1hr)/testing/inference with SPANet
- Kubeflow pipelines for doing a hyperparameter optimization with Katib

Notes:

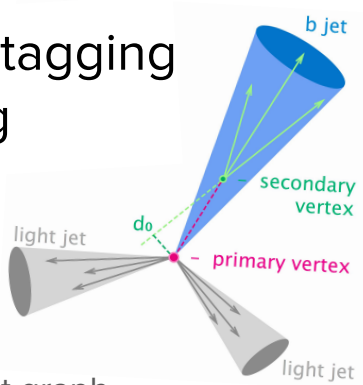
- In both cases, inputs are outputs are located on EOS

Current issues/limitations:

- Not enough available machines with a GPU (limiting possible # of simultaneous trials, sometimes no GPU is available)
- Not possible to run over 24hs (limiting the size of the scan in the hyperparameter space) [kerberos token expires after 24hs]



Hackathon for Jet tagging with CMS Scouting



Infrastructure:

- ParticleNet [[arXiv:1902.08570](https://arxiv.org/abs/1902.08570)] (permutation-invariant graph neural network)

Goal:

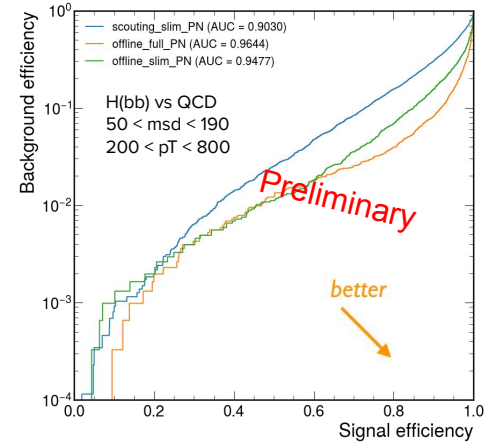
- Study the effects of the CMS Run3 Scouting reconstruction on the jet tagger performance

Team:

- Some experienced and some new to ML/K8s
- Everybody excited to have access to GPU

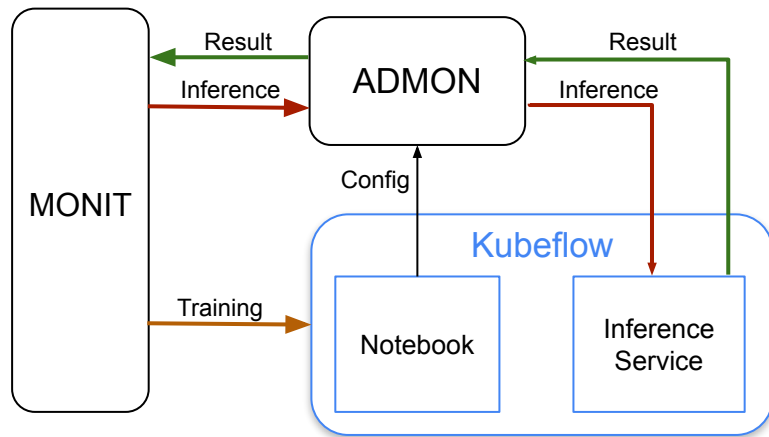
Our initial experience of Kubeflow:

- Very helpful team
 - Daniel Holmberg provided custom Docker image and example
 - Tutorial by Dejan Golubovic
- Limited to max 24h training (Krb5 authentication)
- Some cases more RAM needed
- Would be helpful to have access to Tensorboard for real-time monitoring
- We will continue to use Kubeflow to complete project



Anomaly Detection for MONIT data | ADMON

- **Infrastructure for continuous AD analysis of data stored in MONIT** (IT Monitoring Service)
[under development] - full details @ [ASDF 9th Dec 2021](#)
- **Kubeflow role in ADMON:**
 - Provides complete platform for building and deploying models
 - *KServe* brings important inference service features:
 - Supports HTTP endpoint
 - Request based auto-scaling on CPU/GPU
 - Integrates *Transformer* for pre/post-processing
- **Very positive initial Kubeflow experience:**
 - All components are well integrated
 - K8S cluster available under the hood
 - Deployed *InferenceService* running both *Transformer* and *Predictor*
 - Very grateful to Dejan and Ricardo for the collaboration !
- **Looking forward to some extra features:**
 - Spark support in the Notebook Server
 - e-groups for shared namespace ACLs



*Nikolay Tsvetkov
on behalf of the ADMON Team*

Best Practices for Scalable ML

Code and Datasets Storage

Containers are **ephemeral**, notebook servers can crash

Keep code and datasets on **persistent storage** that can be easily accessed

Code - **GitHub** or **GitLab**, commit regularly

Datasets - **EOS** or **S3** object storage



Model Training

Develop models with scalability in mind

Develop model training to be **distributed** across multiple GPUs

Only **prototype** using a single GPU

[TF Distributed](#) - [TFJob](#)

[Pytorch Distributed](#) - [PyTorchJob](#)

Tensorflow Distributed Example

```
strategy = tf.distribute.MultiWorkerMirroredStrategy()

with strategy.scope():
    model = tf.keras.Sequential([
        tf.keras.layers.Dense(2, input_shape=(5,)),
    ])
    optimizer = tf.keras.optimizers.SGD(learning_rate=0.1)

def dataset_fn(ctx):
    x = np.random.random((2, 5)).astype(np.float32)
    y = np.random.randint(2, size=(2, 1))
    dataset = tf.data.Dataset.from_tensor_slices((x, y))
    return dataset.repeat().batch(1, drop_remainder=True)
dist_dataset = strategy.distribute_datasets_from_function(dataset_fn)

model.compile()
model.fit(dist_dataset)
```

Containerised Workloads

Build Docker images for your ML workloads

Allows for **reproducibility**

Mobility - can run anywhere

Fast deployment

Integration with multiple Kubeflow components

- Pipelines

- Distributed training

- Hyperparameter optimization



Building Docker Images

Automate builds with **Gitlab CI**

Trigger image build with every push

Store images on CERN registries

<https://gitlab.cern.ch/ci-tools/docker-image-builder>

<https://clouddocs.web.cern.ch/containers/registry/gitlab.html>

Building Docker Images

master partienet-images / +

History Find file Web IDE Clone

Merge branch 'kerberos' into 'master' Daniel Rickard Holmberg authored 1 week ago 20d764f3

README CI/CD configuration No license. All rights reserved

Name	Last commit	Last update
notebook	Image naming and pip cache	1 week ago
torchjob	Include kerberos for eos authentication	1 week ago
.gitlab-ci.yml	Image naming and pip cache	1 week ago
README.md	Initial commit	2 weeks ago

master partienet-images / notebook / Lock History Find

Image naming and pip cache Daniel Rickard Holmberg authored 1 week ago

Name	Last commit
..	
Dockerfile	Image naming and pip cache
requirements.txt	Create images

master partienet-images / torchjob / Lock History Find file Web IDE Clone

Include kerberos for eos authentication Daniel Rickard Holmberg authored 1 week ago 6f5a8c80

Name	Last commit	Last update
..		
Dockerfile	Include kerberos for eos authentication	1 week ago
krb5.conf	Include kerberos for eos authentication	1 week ago
requirements.txt	Create images	1 week ago

Conclusion and Next Steps

Where we are today

Already **offering a lot** of what users ask for

- Code validation on notebooks / small jobs

- Distributed Training, Hyper Parameter Optimization

- Model Serving

Direct kubernetes access still required in some cases

- Launching distributed training jobs, checking katib logs

Plus the limitations described earlier

Coming Soon

VSCode development environments

Multi model serving on a single GPU (Nvidia Triton)

Additional dashboard integrations

- Tensorboards

- Katib jobs (Auto ML)

- Model / inference servers

Coming Next

Enforced Quotas

Shared profiles based on ldap / egroups

Improved integration with **other services** - continuous integration, notebooks/swan, etc

Security improvements

Prevent vulnerable workloads, runtime verification, etc











(ASDF) <https://indico.cern.ch/event/1054454/>

Artifact navigation

Experiment with a dataset catalog and automated feature discovery

Model Servers

[+ NEW MODEL SERVER](#)

Status	Name	Age	Predictor	Runtime	Protocol	Storage URI		
	flowers	4 minutes ago	Tensorflow	1.14.0		gs://kf-serving-samples/models/tensorflow/flo...		
	pmml-demo	4 minutes ago	PMML	v0.5.1		https://raw.githubusercontent.com/openscorin...		
	sklearn-iris	4 minutes ago	SKLearn	0.2.1	v2	gs://seldon-models/sklearn/iris		
	torchserve	4 minutes ago	PyTorch	0.3.0	v1	gs://kf-serving-examples/models/torchserve/i...		
	xgboost-iris	4 minutes ago	XGBoost	0.2.1	v2	gs://kf-serving-samples/models/xgboost/iris		

Community

Community effort to improve machine learning infrastructure

Kubeflow still not fully complete

CERN users can influence future developments

High interest in our feedback

Anyone can contribute to open source - <https://github.com/kubeflow>

Everyone is invited to **provide feedback!**

Previous Talks

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](#)

Important 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!

Q&A

Backup Slides

Hyperparameter Jobs

Make sure the script runs on a **single GPU in a notebook server**

Make sure it uses a GPU + it completes

Make sure the script runs on a **single GPU as a Katib job with 1 trial**

Then carefully expand to 2, 4, 10, 20 trials, and monitor closely

Once sure it works, run a complete search

Be aware that **some combinations of HP might crash the script**

Prepare the script for these edge cases (exception handling etc)

Store metrics in a preferred format in two places

In the container storage to be accessed by the UI

On EOS or S3 for persistent storage