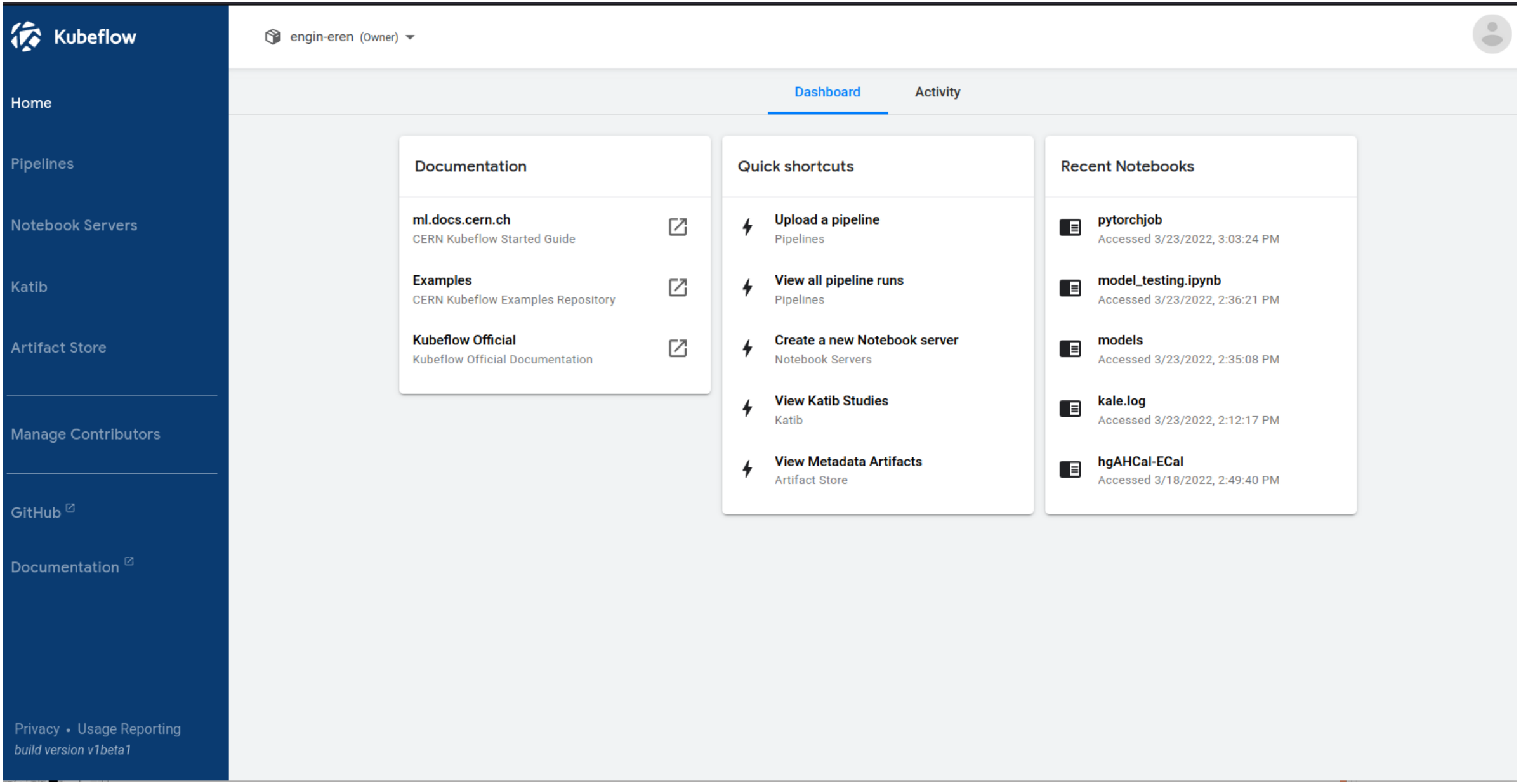
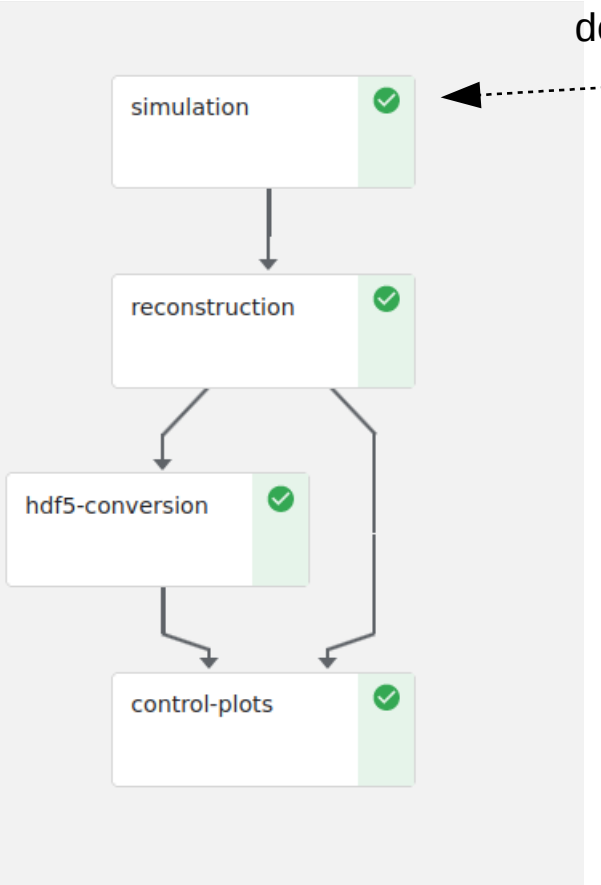
Kubeflow at *ml.cern.ch*



Kubeflow at *ml.cern.ch*

Data pipeline and experiment

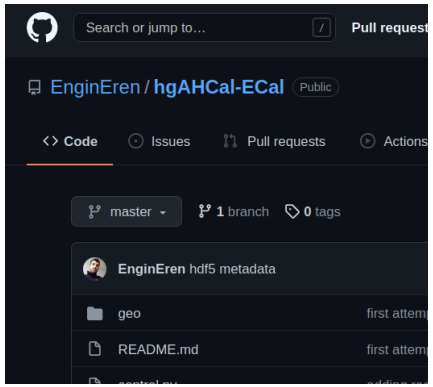


docker image for ilcsoft

steering parameters!

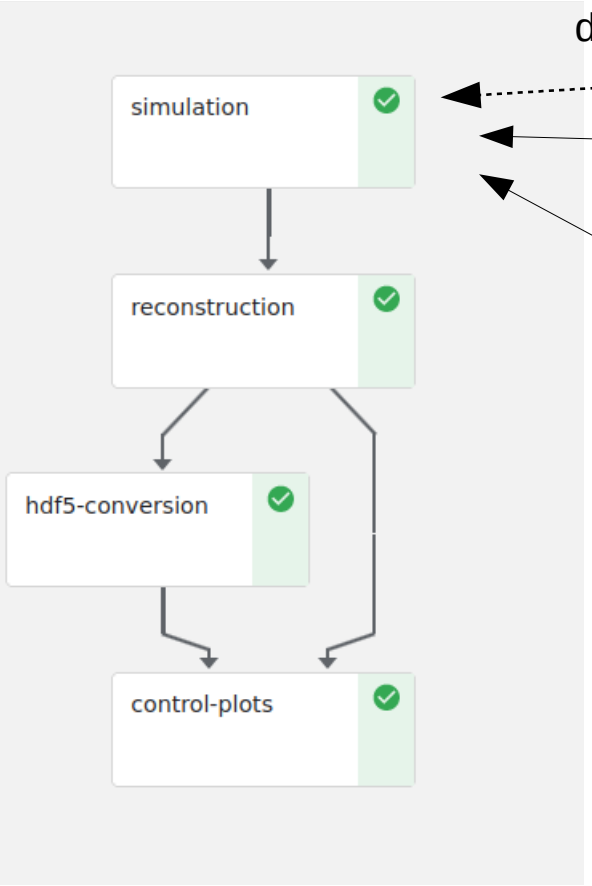
Gun position: (-10, 0, 18) cm
Particle: pion+
Energy: 50 GeV

main github repo



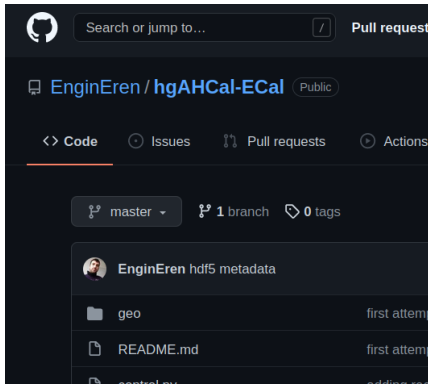
Kubeflow at *ml.cern.ch*

Data pipeline and experiment



docker image for ilcsoft

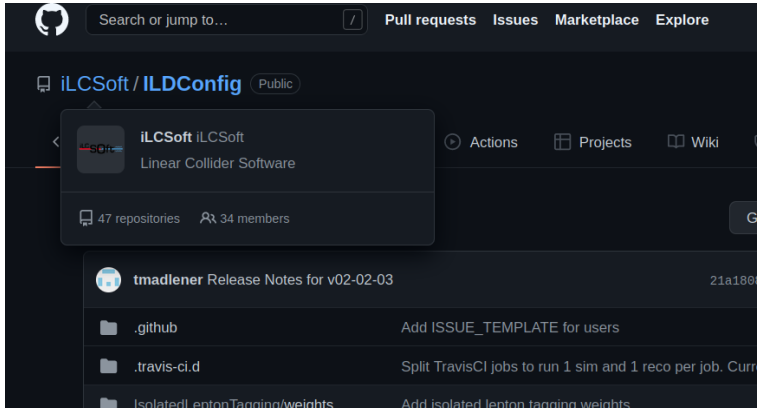
main github repo



steering parameters!

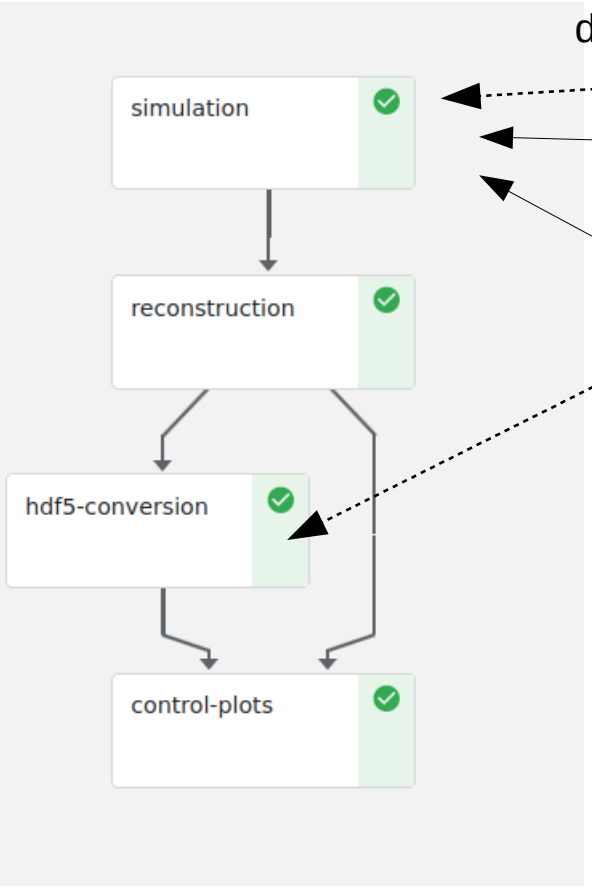
Gun position: (-10, 0, 18) cm
Particle: pion+
Energy: 50 GeV

ILDConfig official repo



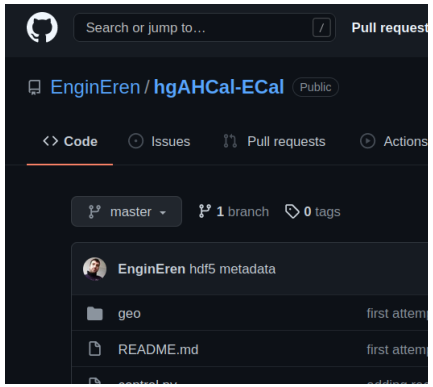
Kubeflow at *ml.cern.ch*

Data pipeline and experiment



docker image for ilcsoft

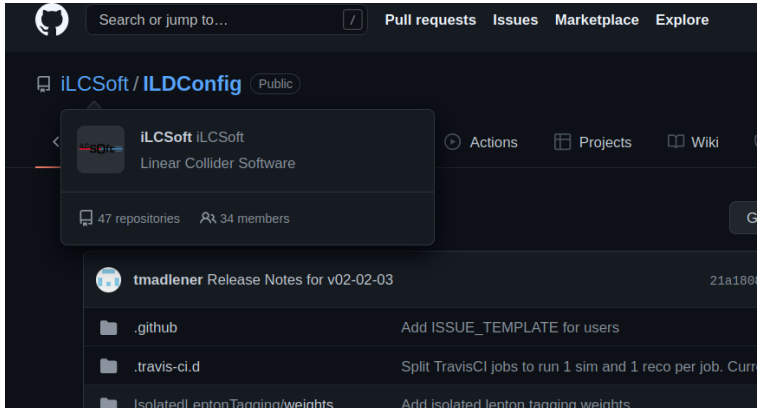
main github repo



steering parameters!

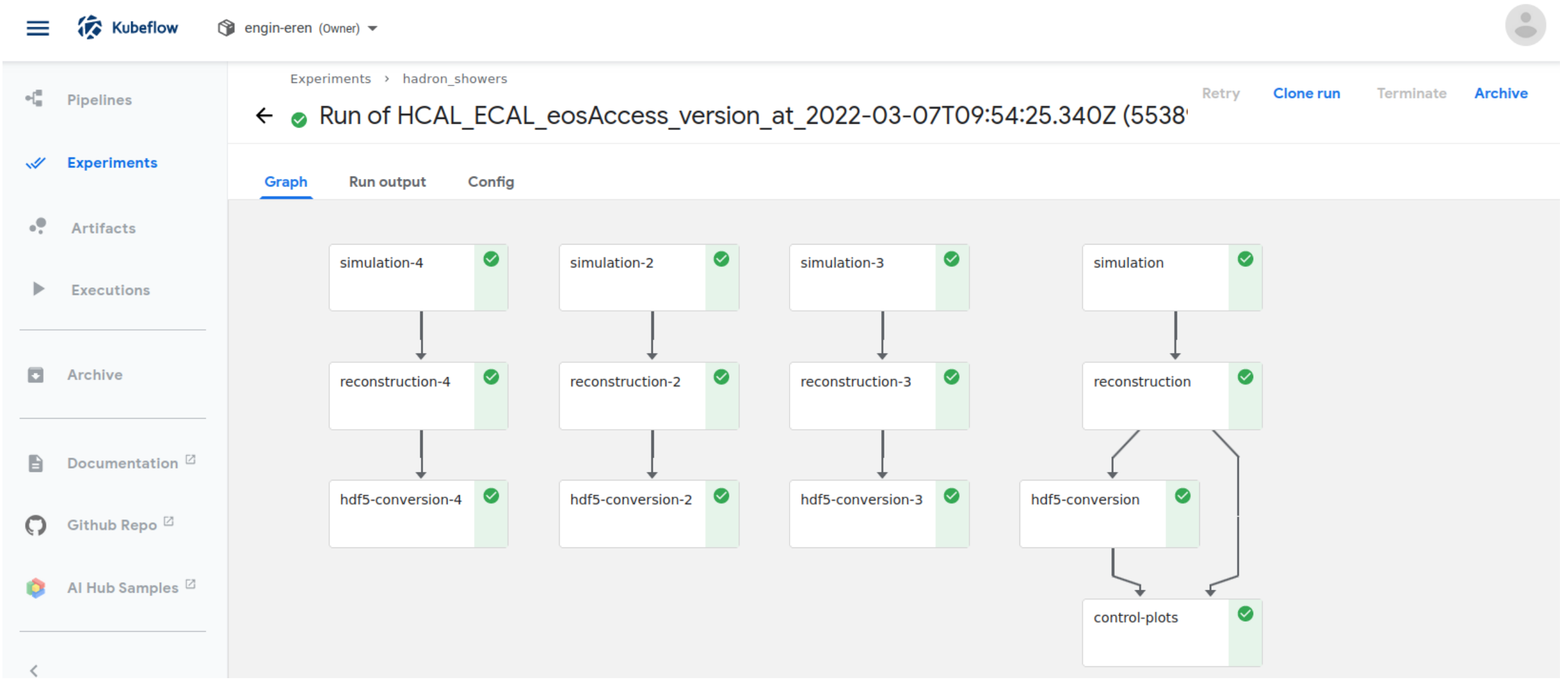
Gun position: (-10, 0, 18) cm
Particle: pion+
Energy: 50 GeV

ILDConfig official repo



Writing all hdf5 files to **CERN-EOS**

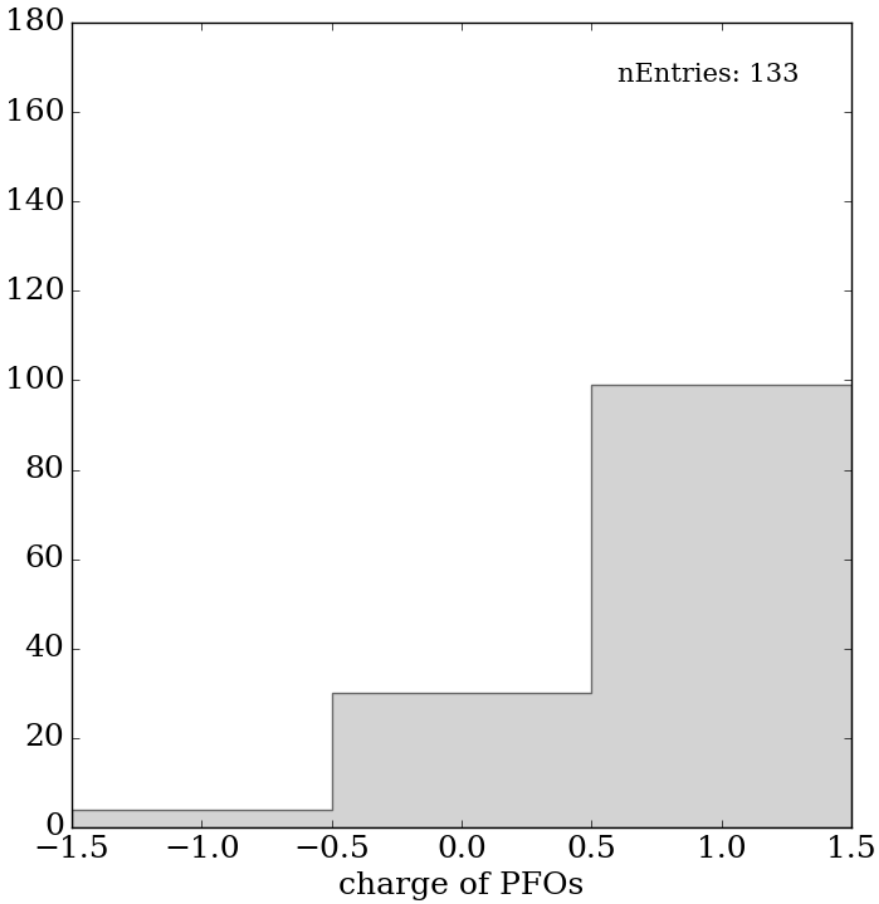
More jobs

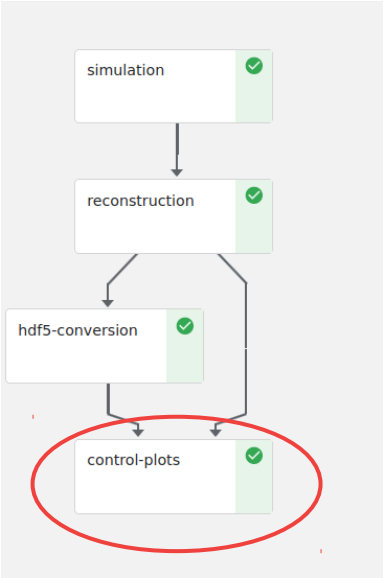


Writing all hdf5 files to CERN-EOS

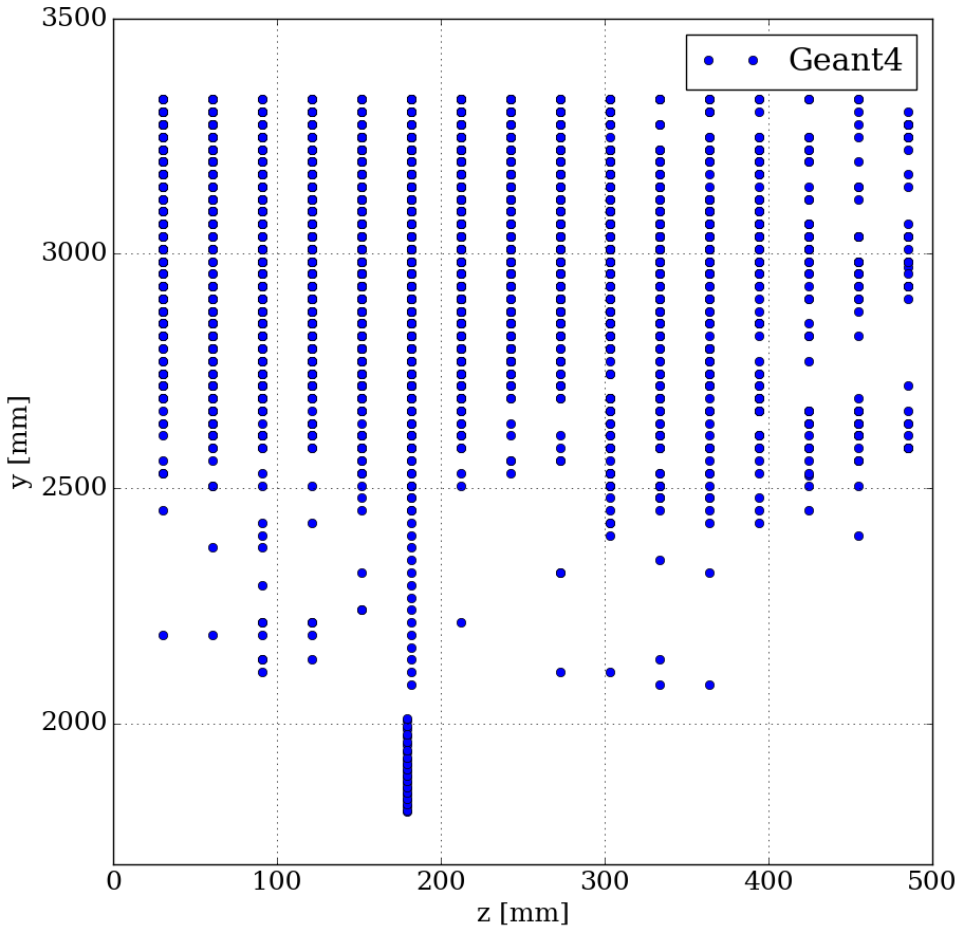
Control Plots (some examples)

Pandora PFO:



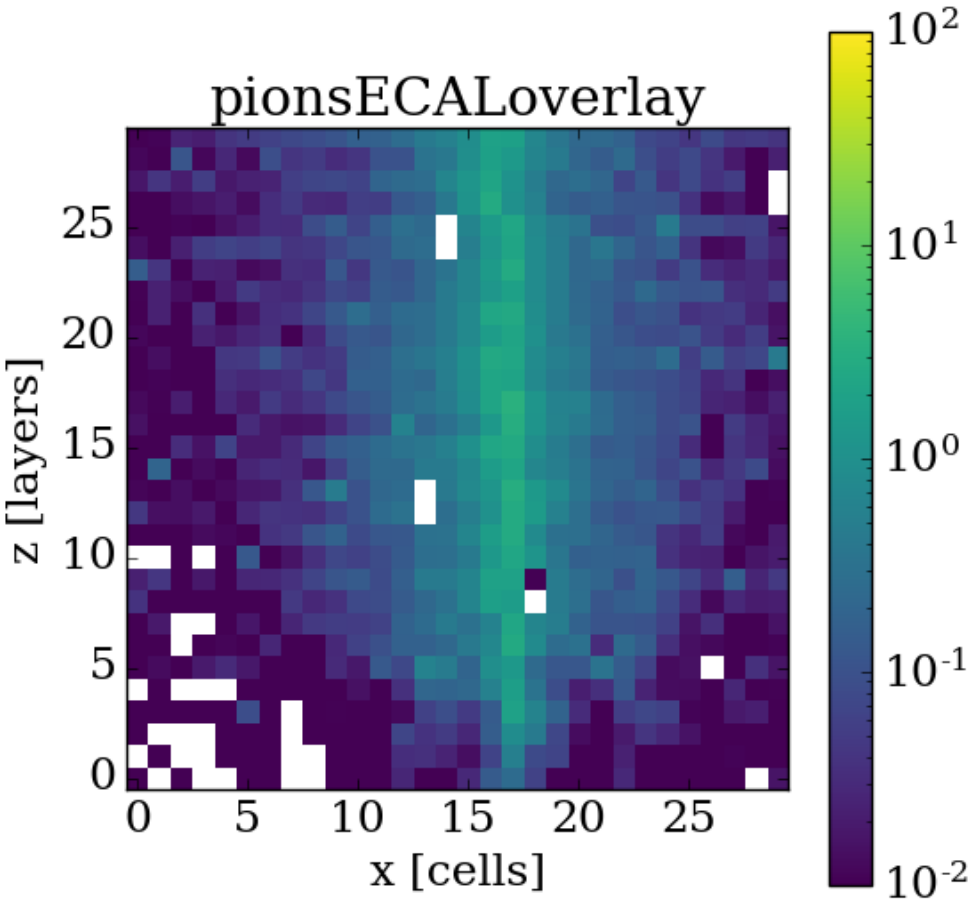


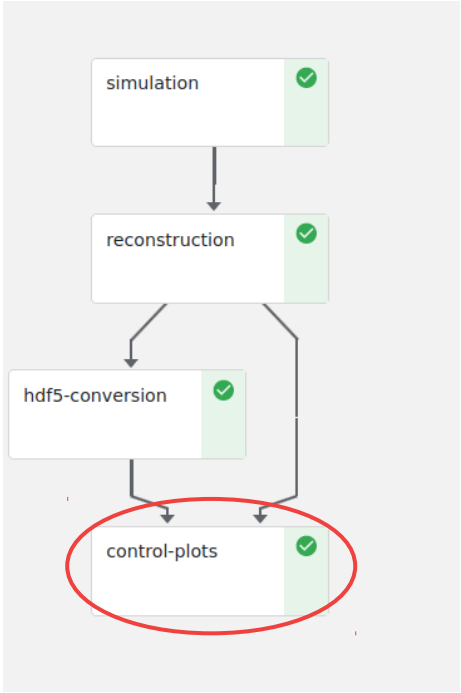
Simulated hits:



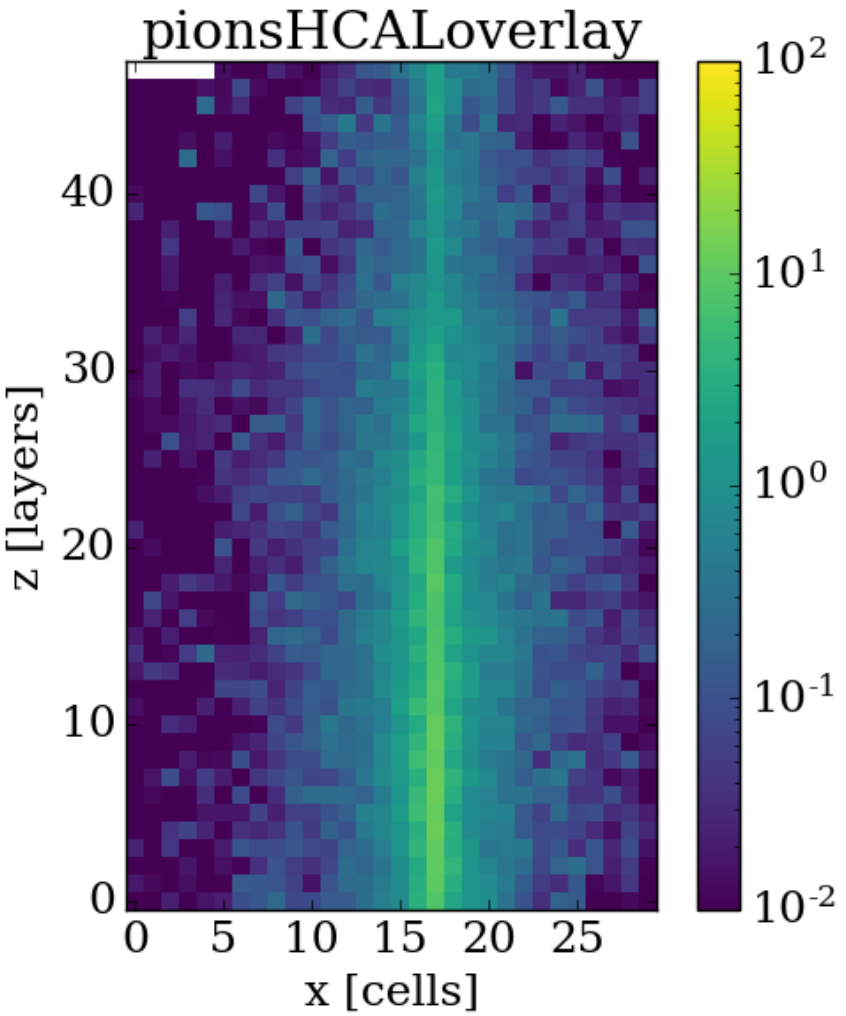
Control Plots (some examples)

ECAL (z-x)





HCAL (z-x)



Training: First prototype in the notebook server (Juypter Lab)

File Edit View Run Kernel Tabs Settings Help

Terminal 1 | constrainer.py | model_testing.ipynb

Markdown | Python 3

/

Name

- hgAHCal-ECal
- models
- pytorchjob
- criticRes.py
- kale.log
- model_testing.ipynb

Generator

[1]:

```
import torch
import torch.nn as nn
import torch.nn.parallel
import torch.nn.functional as F
from torch.autograd import Variable
from torch.utils import data


class DCGAN_G(nn.Module):
    """
        generator component of WGAN
    """
    def __init__(self, ngf, nz):
        super(DCGAN_G, self).__init__()

        self.ngf = ngf
        self.nz = nz

        kernel = 4

        # input energy shape [batch x 1 x 1 x 1 ] going into convolutional
        self.conv1_1 = nn.ConvTranspose3d(1, ngf*4, kernel, 1, 0, bias=False)
        # state size [ ngf*4 x 4 x 4 x 4]

        # input noise shape [batch x nz x 1 x 1] going into convolutional
        self.conv1_100 = nn.ConvTranspose3d(nz, ngf*4, kernel, 1, 0, bias=False)
        # state size [ ngf*4 x 4 x 4 x 4]


        # outs from first convolutions concatenate state size [ ngf*8 x 4 x 4]
        # and going into main convolutional part of Generator
        self.main_conv = nn.Sequential(

            nn.ConvTranspose3d(ngf*8, ngf*4, kernel, 2, 1, bias=False),
            nn.LayerNorm([8, 8, 8]),
```

Notebook Servers + NEW SERVER

Status	Name	Age	Image	CPU	Memory	Volumes	
✓	model-testing-gpu-pytorchv2	5 days ago	pytorch-notebook-gpu-1.8.1:v0.6.1-30	1	4.0Gi		CONNECT

Kubeflow

Home

Pipelines

Notebook Servers

Katib

Artifact Store

Manage Contributors

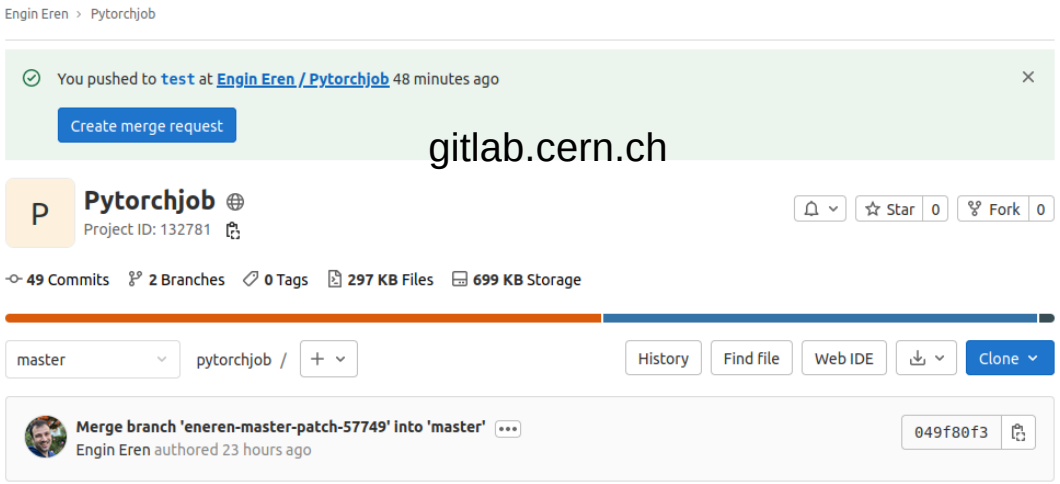
1 1 Python 3 | Idle Mode: Command Ln 1, Col 1 model_testing.ipynb

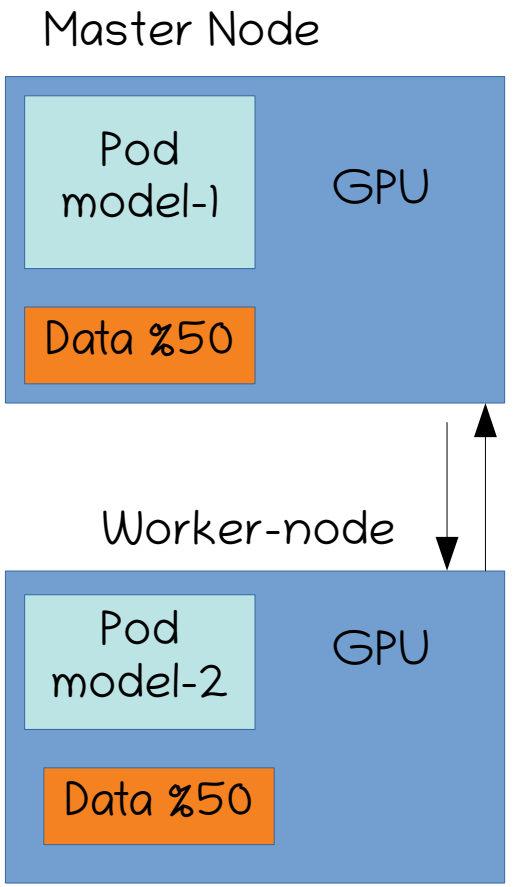
Distributed training (DDP) with pytorch job

Master :
- volumes : EOS config/secret
- docker image: gitlab-registry.cern.ch/eneren/../
- python command: regressor.py --epochs 2
- nvidia-driver

Worker
- same..
- replicas: **1**

YAML

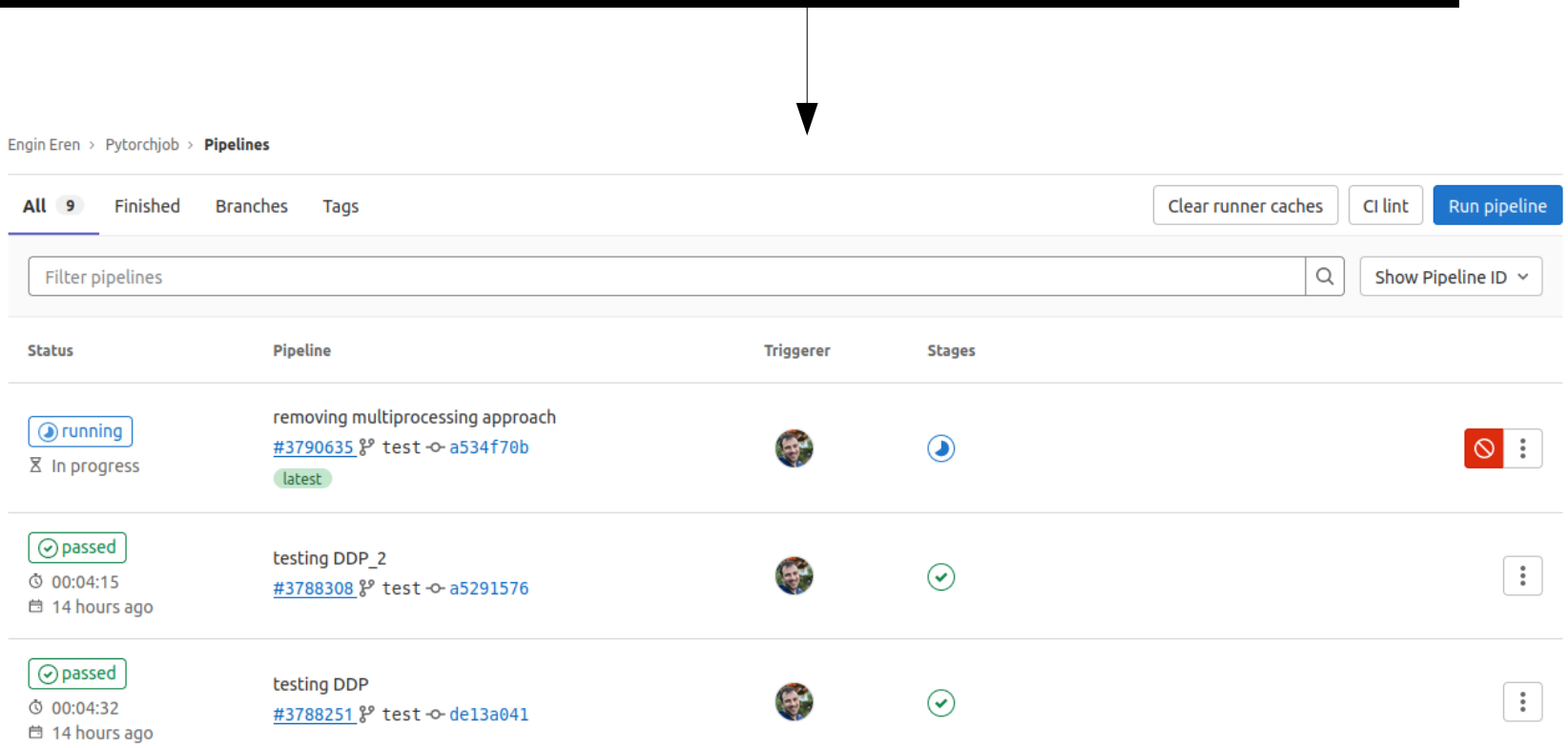


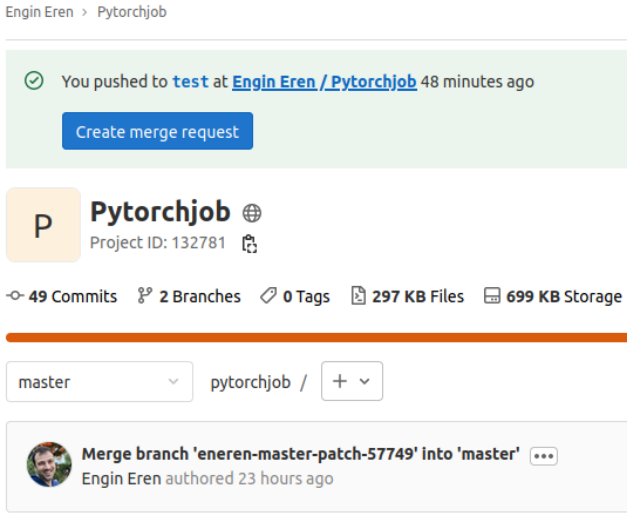


Distributed training: Building the docker image with CI

Make local changes and commit

```
(base) engin@engin-Latitude-7300:~/pytorchjob$ git push origin test
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 519 bytes | 519.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote:
remote: To create a merge request for test, visit:
remote:   https://gitlab.cern.ch/eneren/pytorchjob/-/merge_requests/new?merge_request%5Bsource_branch%5D=test
remote:
To ssh://gitlab.cern.ch:7999/eneren/pytorchjob.git
   a529157..a534f70  test -> test
(base) engin@engin-Latitude-7300:~/pytorchjob$
```





Distributed training: Submit training job in terminal

root@pytorch-dist-regressor- | Terminal 5

```
@model-testing-cpu-0:~/pytorchjob$ kubectl apply -f pytorch_job_regressor_nccl.yaml
pytorchjob.kubeflow.org/pytorch-dist-regressor-samplerv2-testnccl created
@model-testing-cpu-0:~/pytorchjob$ kubectl get pods
```

```
pytorch-dist-regressor-samplerv2-nccl-master-0       1/1     Running       0          55m
pytorch-dist-regressor-samplerv2-nccl-worker-0       1/1     Running       0          55m
```

```
@model-testing-cpu-0:~/pytorchjob$ kubectl logs pytorch-dist-regressor-samplerv2-nccl-master-0
whoami:  0
beginning of python script
True
Using CUDA
Using distributed PyTorch with nccl backend
[init] == local rank: 0, global rank: 0, world size: 2 ==
loading data
init models
starting training...
Energy Regressor --> Train Epoch: 1 [0/74999 (0%)]      loss=2513.1494
Energy Regressor --> Train Epoch: 1 [5000/74999 (13%)]  loss=1959.6224
Energy Regressor --> Train Epoch: 1 [10000/74999 (27%)] loss=1403.6118
Energy Regressor --> Train Epoch: 1 [15000/74999 (40%)] loss=864.0803
Energy Regressor --> Train Epoch: 1 [20000/74999 (53%)] loss=663.8013
Energy Regressor --> Train Epoch: 1 [25000/74999 (67%)] loss=431.6797
Energy Regressor --> Train Epoch: 1 [30000/74999 (80%)] loss=327.7393
Energy Regressor --> Train Epoch: 1 [35000/74999 (93%)] loss=14.7760
Energy Regressor --> Test Epoch: 1 [0/2999 (0%)]        loss=16.1639
[W pthreadpool-cpp.cc:90] Warning: Leaking Caffe2 thread-pool after fork. (function pthreadpool)
Energy Regressor --> Train Epoch: 2 [0/74999 (0%)]      loss=32.3057
@model-testing-cpu-0:~/pytorchjob$
```

Thank you