



A workflow tool for the development of AI on HPC

SOS26 - 12.03.2024 - Kristian Ehlert, Sara Grau

MAELSTROM

www.maelstrom-eurohpc.eu



EuroHPC
Joint Undertaking

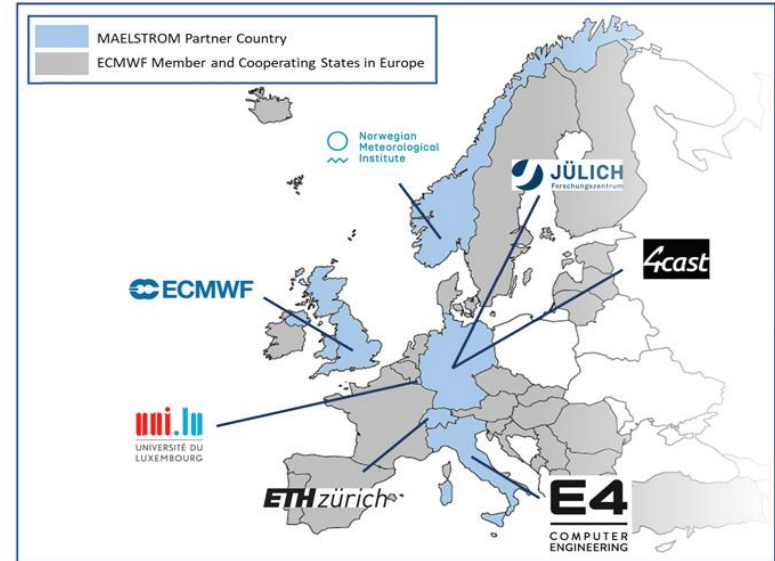
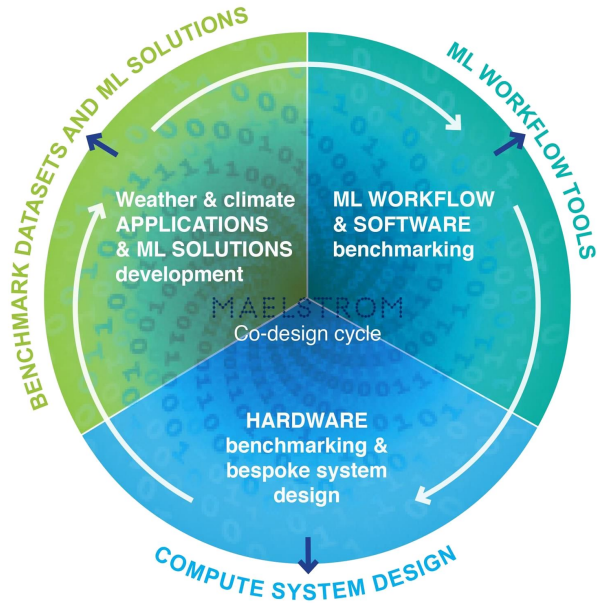


"The MAELSTROM project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No. 955513. The JU receives support from the European Union's Horizon 2020 research and innovation programme and United Kingdom, Germany, Italy, Luxembourg, Switzerland, Norway."

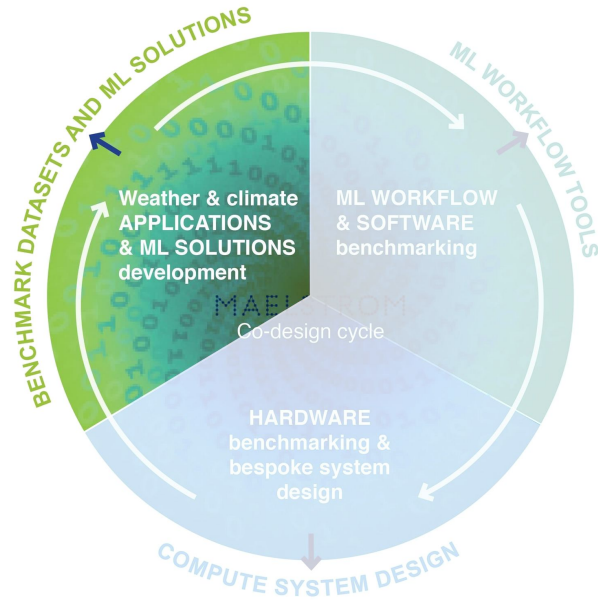
Coordinated by



The Maelstrom project



ML applications in Maelstrom



ML applications in Maelstrom

- 6 different ML applications are developed within Maelstrom
 - Open source code and data (~TB)
- General aim: improving weather forecasts

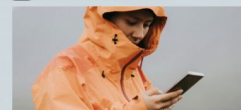
1 Citizen observations for better local forecasts



There is strong demand for accurate local weather forecasts - but NWP models may be unable to forecast local (extreme) weather. We want to use observations of "regular people" to improve high-resolution analyses. NWP post-processing can significantly improve operational weather forecasts on weather apps like yr.no

[See details](#)

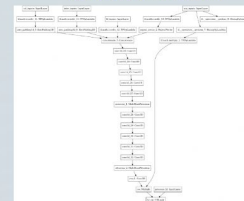
2 Social media data for better local forecasts



We believe that weather-related information from social networks could enhance local weather predictions for most dominant infrastructures in Europe (e.g. airports) in near real-time. So we bring tweets and other social media output to the table as a new "weather sensor".

[See details](#)

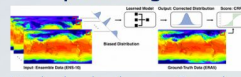
3 Neural network emulators for faster forecast models & data assimilation



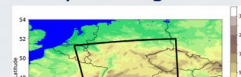
Accelerated weather and climate computing will consume fewer resources and deliver faster predictions on emerging extreme events. This app uses machine learning to accelerate a key and expensive component of weather and climate models, radiative transfer, modeling heating from the sun and cooling from the earth surface.

[See details](#)

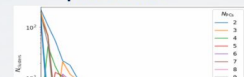
4 Improved ensemble predictions in forecast post-processing



5 Improved local weather predictions in forecast post-processing



6 Predict large-scale weather patterns to support energy production



Two examples

- Improve temperature predictions with citizen weather stations
 - ML model augments forecasts with temperature measurements

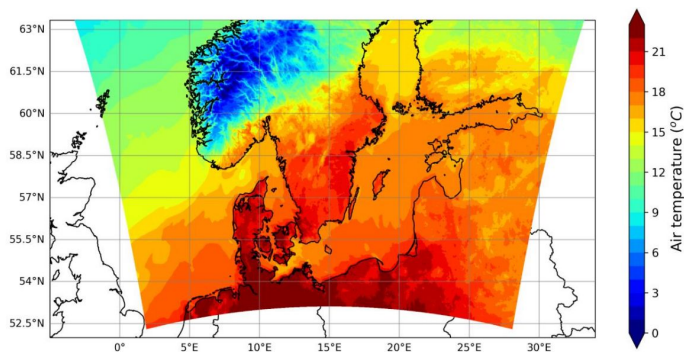


Fig. 4: An example forecast for 13:00Z on September 16, 2023 made using the final U-Net model. The map shows the 50th percentile forecast for the southern half of the domain.

- Use citizen observations on social media to improve precipitation forecasts
 - Fine-tune NLP classifier to decide if “raining” or “not raining” from Tweet

Well that last rumble of thunder made the house shake, I wasn't scared for a couple of seconds

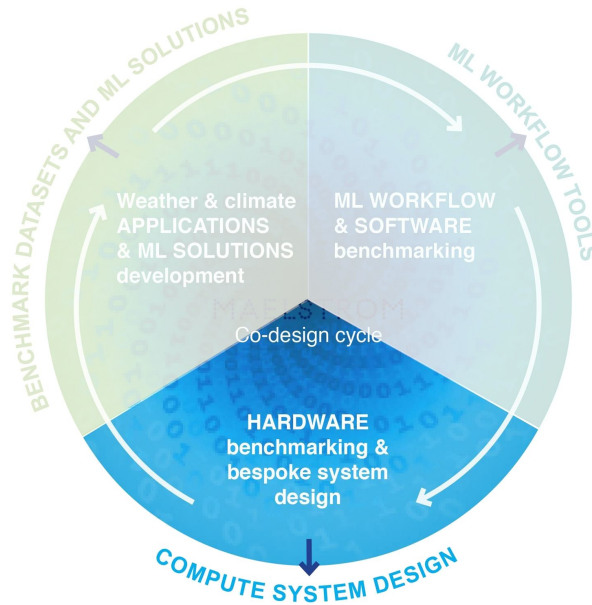
Wandered down the hill in the nice weather

☀️🌊
#university#beach
#sea #seaside #unilife
#sun @ Beach

Pimms and pizza outside BBC television centre!
What a lovely day to wait around for a famous actor. ☀️

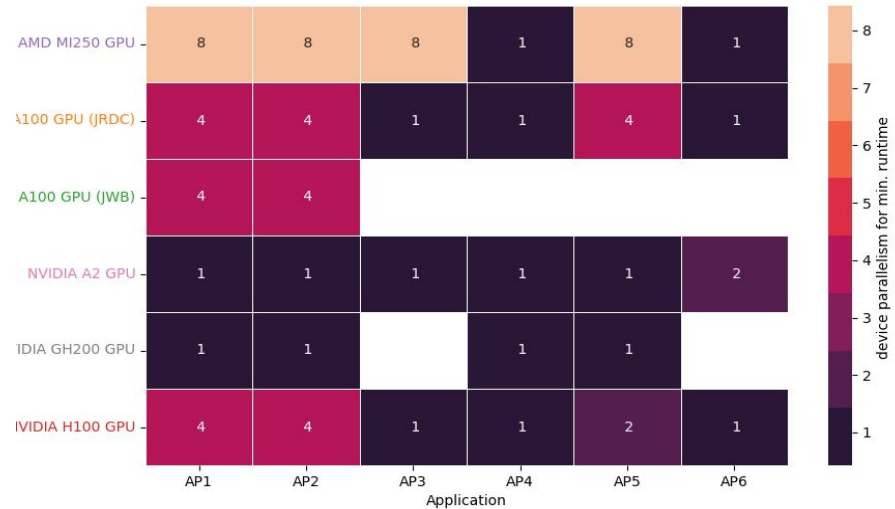
Any danger of having that sun back for today?

Hardware benchmarking in Maelstrom

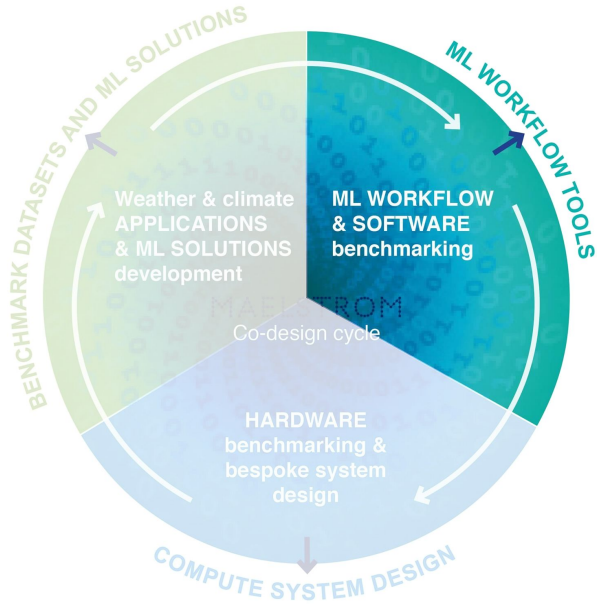


Hardware benchmarking in Maelstrom

- Access required to at least two HPC clusters
- Benchmarking on more than ten different machines with various configurations



ML workflow tool: Mantik





ML tool requirements

Interface to Compute Resources

- Abstract away infrastructure
- Unified access to compute resources (HPC, Cloud)

Reproducibility

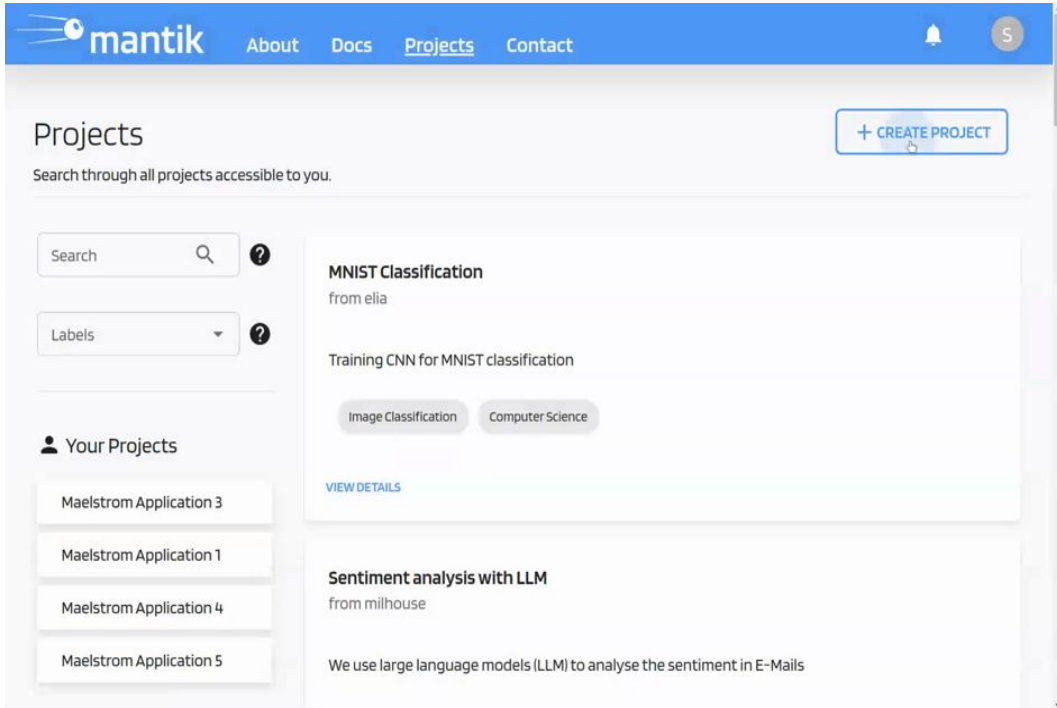
- Recording of input parameters, metrics, models
- Data versioning

Collaboration

- Sharing ML solutions
- Exchange of knowledge
- Improvement of ML solutions

Projects in Mantik

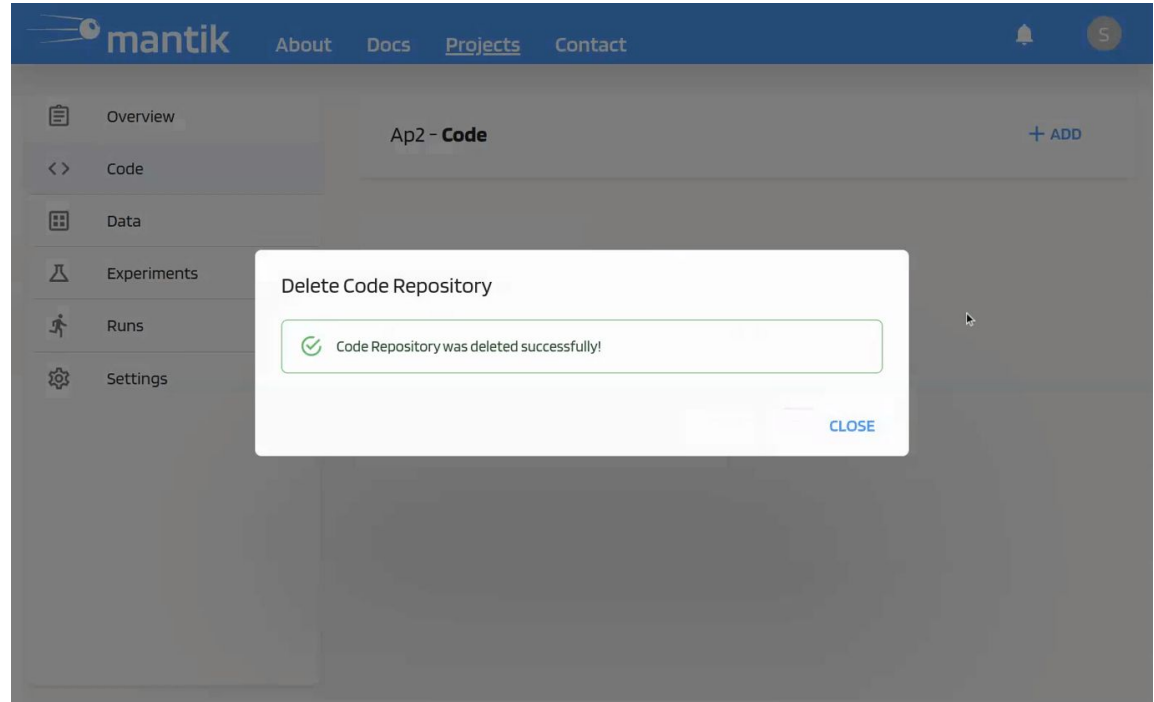
- Basis of working with Mantik
- Holds major assets
 - Code
 - Experiments
 - Runs
- Allow for collaborations on projects



The screenshot shows the Mantik web interface. At the top is a blue navigation bar with the Mantik logo and links for 'About', 'Docs', 'Projects', and 'Contact'. A notification bell and a user profile icon are on the right. Below the navigation bar, the main content area is titled 'Projects' and includes a '+ CREATE PROJECT' button. A search bar and a 'Labels' dropdown menu are provided for filtering. The page displays a list of projects under the heading 'Your Projects'. The first project is 'MNIST Classification' by 'elia', described as 'Training CNN for MNIST classification', with tags for 'Image Classification' and 'Computer Science'. The second project is 'Sentiment analysis with LLM' by 'milhouse', described as 'We use large language models (LLM) to analyse the sentiment in E-Mails'.

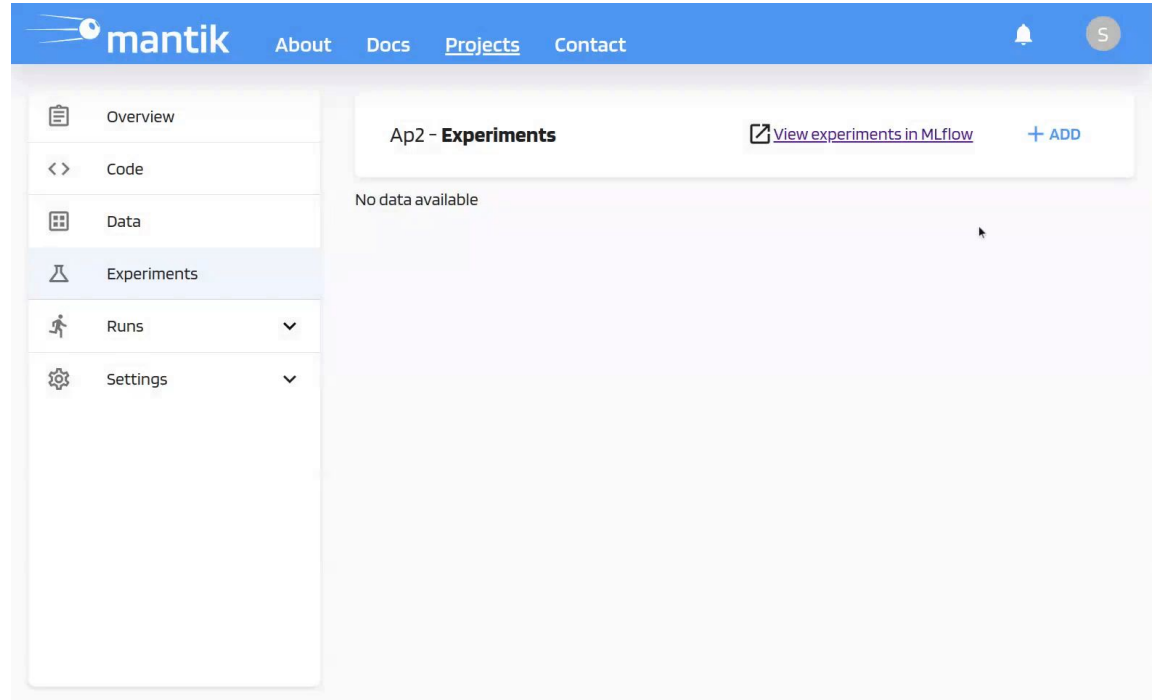
Code in Mantik

- Code linked to git repositories
 - Simplifies versioning
 - Supports major public repository providers



Experiments in Mantik

- Holds tracked parameters and metrics
- Container that attaches metrics/parameters of runs





The screenshot displays the Mantik web application interface. The top navigation bar is blue and contains the Mantik logo, links for 'About', 'Docs', 'Projects', and 'Contact', a notification bell, and a user profile icon labeled 'S'. A left sidebar menu is visible with the following items: 'Overview', 'Code', 'Data', 'Experiments' (highlighted), 'Runs', and 'Settings'. The main content area is titled 'Ap2 - Experiments' and includes a link to 'View experiments in MLflow' and a '+ ADD' button. Below the title, the text 'No data available' is displayed.



Configuration of Runs

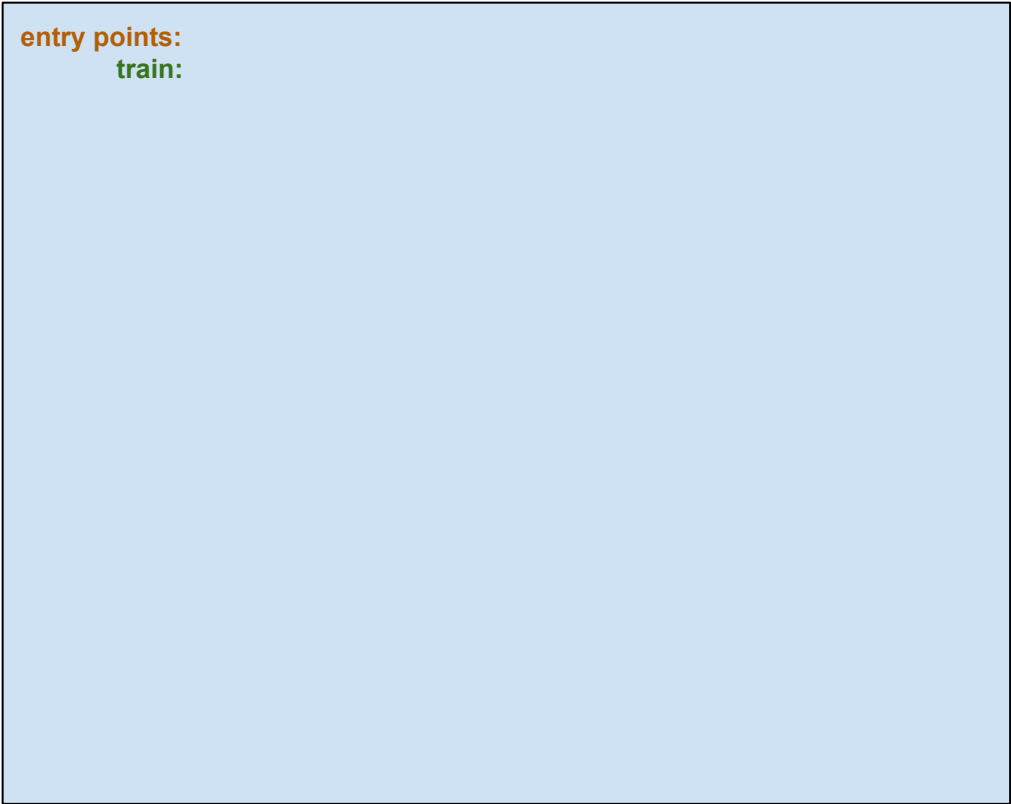
- Two files define full execution on HPC
 - *MLproject* file defines Run to be executed on HPC
 - Compute backend file configures HPC environment

Name
..
 MLproject
 compute-backend-config.yaml



Configuration of execution command

- *MLproject* file defines Run to be executed on HPC
- Entry points define various tasks to be executed as Run



entry points:
train:

Configuration of execution command

- *MLproject* file defines Run to be executed on HPC
- Entry points define various tasks to be executed as Run
- Modifiable parameters may have default values

```
entry points:  
  train:  
    parameters:  
      learning_rate:  
        type: float  
        default: 3e-5  
        ...  
      validation_size:  
        ...  
      ...
```

Configuration of execution command

- *MLproject* file defines Run to be executed on HPC
- Entry points define various tasks to be executed as Run
- Modifiable parameters may have default values
- Command defines execution

```
entry points:  
  train:  
    parameters:  
      learning_rate:  
        type: float  
        default: 3e-5  
        ...  
      validation_size:  
        ...  
        ...  
    command: >  
      python train.py  
        -- validation_size (validation_size)  
        -- ...
```

Configuration of execution command

- *MLproject* file defines Run to be executed on HPC
- Entry points define various tasks to be executed as Run
- Modifiable parameters may have default values
- Command defines execution
- Various tasks can be defined in the same file

```
entry points:
  train:
    parameters:
      learning_rate:
        type: float
        default: 3e-5
      ...
      validation_size:
        ...
      ...
    command: >
      python train.py
      -- validation_size (validation_size)
      -- ...

  split-dataset:
    parameters: ...
    command: >
      python build.py
      -- ..

  benchmark: ...
```



Configuration of HPC environment

- Compute backend file configures runtime environment
- Specify HPC access via API

Firecrest:

ApiUrl: <https://firecrest.cscs.ch>

TokenUrl: <https://auth.cscs.ch/auth/realms/firecrest-clients/protocol/openid-connect/token>

Machine: daint



Configuration of HPC environment

- Compute backend file configures runtime environment
- Specify HPC access via API
- Define runtime environment on compute/login node

Firecrest:

ApiUrl: <https://firecrest.cscs.ch>

TokenUrl: <https://auth.cscs.ch/auth/realms/firecrest-clients/protocol/openid-connect/token>

Machine: daint

Environment:

PreRunCommand:

Command: |

module load Stages/2023 GCCcore/.11.3.0 Python/...

source /p/scratch/deepacf/maelstrom/.../bin/activate

ExecuteOnLoginNode: false

Variables:

GIT_PYTHON_REFRESH: quiet



Configuration of HPC environment

- Compute backend file configures runtime environment
- Specify HPC access via API
- Define runtime environment on compute/login node
- Specify job resources

Firecrest:

ApiUrl: <https://firecrest.cscs.ch>

TokenUrl: <https://auth.cscs.ch/auth/realms/firecrest-clients/protocol/openid-connect/token>

Machine: daint

Environment:

PreRunCommand:

Command: |

module load Stages/2023 GCCcore/.11.3.0 Python/...

source /p/scratch/deepacf/maelstrom/.../bin/activate

ExecuteOnLoginNode: false

Variables:

GIT_PYTHON_REFRESH: quiet

Resources:

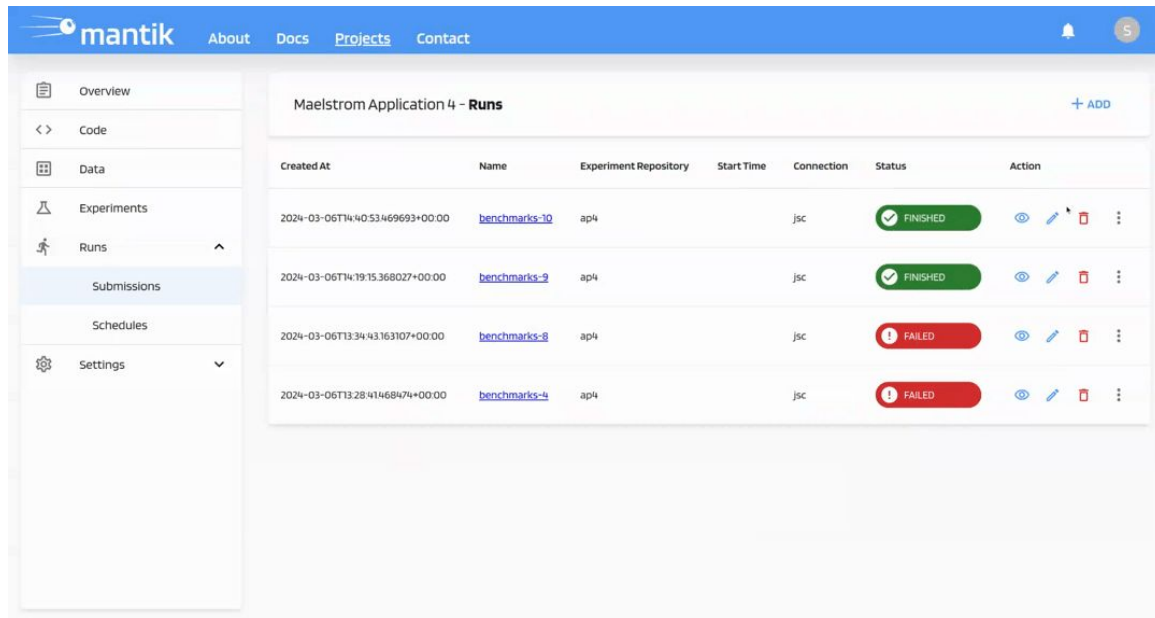
Queue: normal

Nodes: 1

















NodeConstraints: gpu

Submitting a Run

- Execution of Code on HPC
- Modify parameters on-the-fly
- Re-run functionality

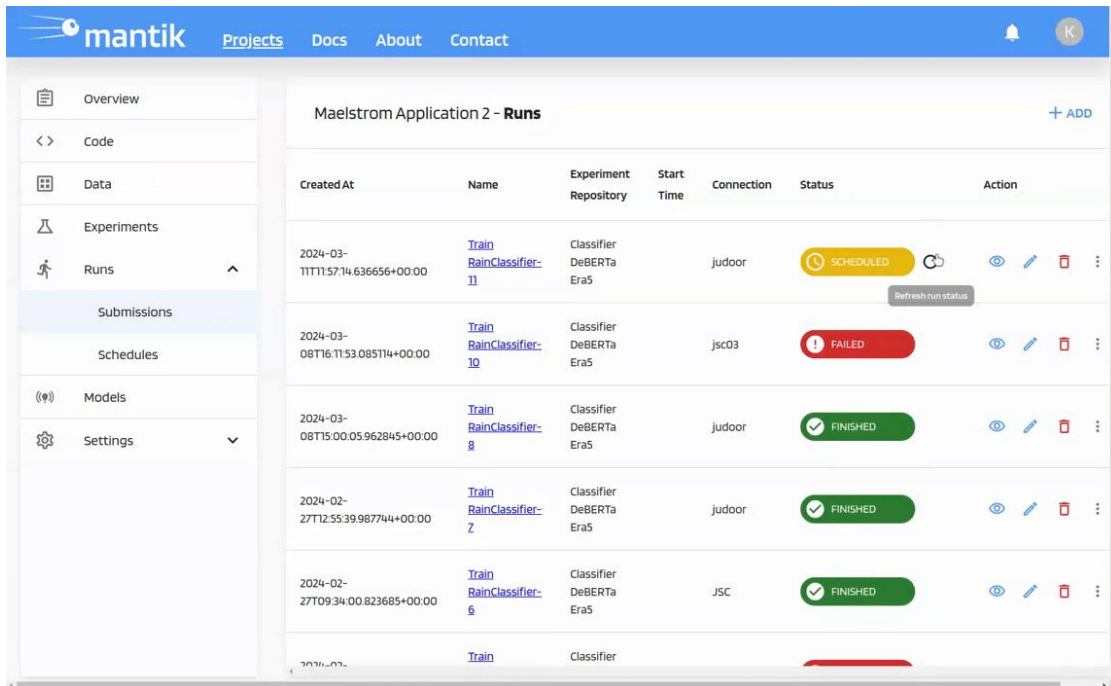


The screenshot displays the Mantik web interface. The top navigation bar includes the Mantik logo and links for 'About', 'Docs', 'Projects', and 'Contact'. A left sidebar contains a navigation menu with options: Overview, Code, Data, Experiments, Runs (selected), Submissions, Schedules, and Settings. The main content area is titled 'Maelstrom Application 4 - Runs' and features a table of run records. Each record includes a 'Created At' timestamp, a 'Name' (e.g., 'benchmarks-10'), an 'Experiment Repository' ('ap4'), a 'Start Time', a 'Connection' ('jsc'), a 'Status' (either 'FINISHED' in green or 'FAILED' in red), and an 'Action' column with icons for viewing, editing, deleting, and a dropdown menu. A '+ ADD' button is located in the top right corner of the table area.

Created At	Name	Experiment Repository	Start Time	Connection	Status	Action
2024-03-06T14:40:53.469693+00:00	benchmarks-10	ap4		jsc	FINISHED	   
2024-03-06T14:19:15.368027+00:00	benchmarks-9	ap4		jsc	FINISHED	   
2024-03-06T13:34:43.163107+00:00	benchmarks-8	ap4		jsc	FAILED	   
2024-03-06T13:28:41.468474+00:00	benchmarks-4	ap4		jsc	FAILED	   

Fetching Status and Logs of a Run

- View status of job
- Check HPC and application logs
- Download run artifacts to local machine

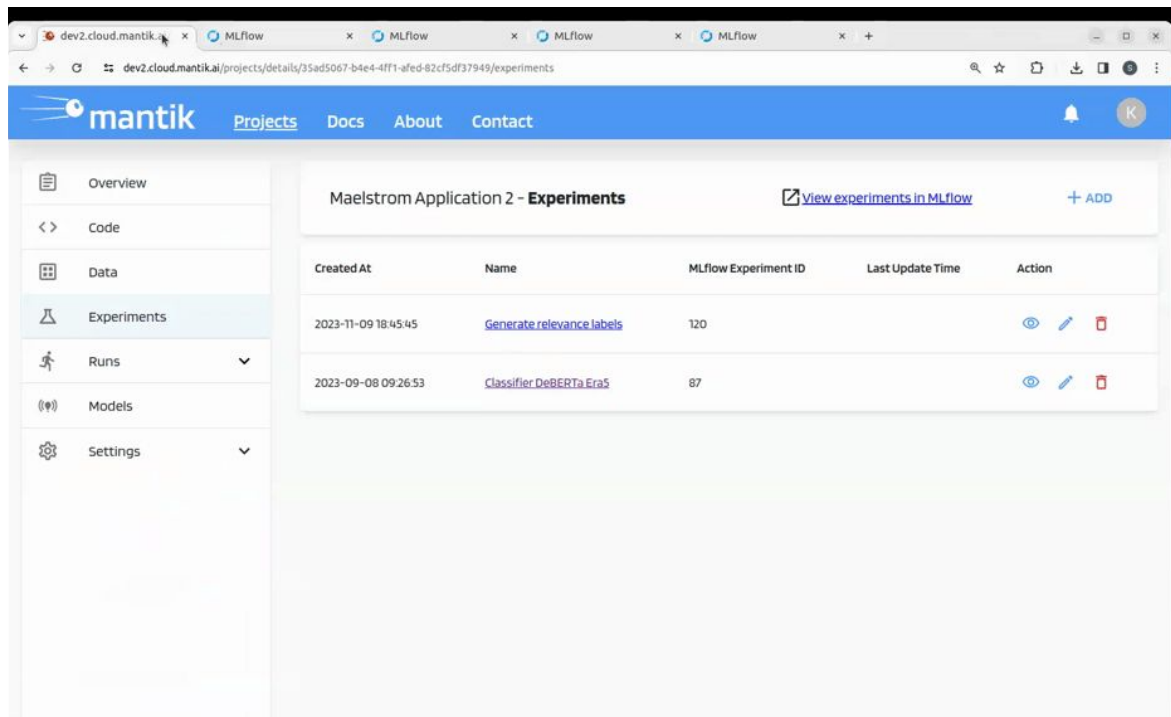


The screenshot displays the Mantik web interface for 'Maelstrom Application 2 - Runs'. The interface includes a navigation sidebar on the left with options like Overview, Code, Data, Experiments, Runs, Submissions, Schedules, Models, and Settings. The main content area shows a table of runs with the following data:







Created At	Name	Experiment Repository	Start Time	Connection	Status	Action
2024-03-11T11:57:14.636656+00:00	Train RainClassifier-11	Classifier DeBERTa Era5		judoor	SCHEDULED	Refresh run status , View , Edit , Delete
2024-03-08T16:11:53.085114+00:00	Train RainClassifier-10	Classifier DeBERTa Era5		jsc03	FAILED	View , Edit , Delete
2024-03-08T15:00:05.962845+00:00	Train RainClassifier-8	Classifier DeBERTa Era5		judoor	FINISHED	View , Edit , Delete
2024-02-27T12:55:39.987744+00:00	Train RainClassifier-Z	Classifier DeBERTa Era5		judoor	FINISHED	View , Edit , Delete
2024-02-27T09:34:00.823685+00:00	Train RainClassifier-6	Classifier DeBERTa Era5		JSC	FINISHED	View , Edit , Delete

Tracking of results

- Real-life tracking of metrics, e.g. loss, accuracy
- Interactively compare model performance

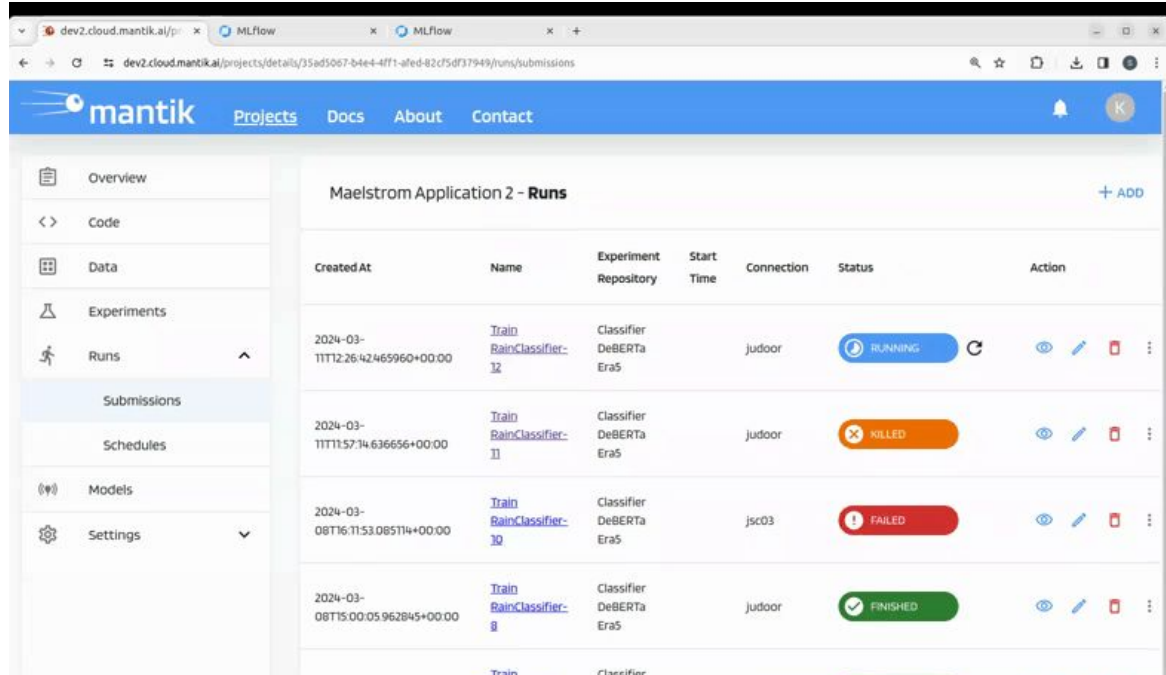


The screenshot displays the Mantik MLflow web interface. The browser address bar shows the URL: `dev2.cloud.mantik.ai/projects/details/35ad5067-b4e4-4ff1-afed-82cf5df37949/experiments`. The page title is "Mantik" with navigation links for "Projects", "Docs", "About", and "Contact". A sidebar on the left contains a menu with "Overview", "Code", "Data", "Experiments" (selected), "Runs", "Models", and "Settings". The main content area is titled "Maelstrom Application 2 - Experiments" and includes a link to "View experiments in MLflow" and a "+ ADD" button. Below this is a table with the following data:

Created At	Name	MLflow Experiment ID	Last Update Time	Action
2023-11-09 18:45:45	Generate relevance labels	120		  
2023-09-08 09:26:53	Classifier DeBERTa Era5	87		  

Tracking of results

- Real-life tracking of metrics, e.g. loss, accuracy
- Interactively compare model performance



The screenshot displays the MLflow web interface for a project named 'Maelstrom Application 2'. The interface includes a navigation sidebar on the left with options like Overview, Code, Data, Experiments, Runs, Submissions, Schedules, Models, and Settings. The main content area shows a table of runs with columns for Created At, Name, Experiment Repository, Start Time, Connection, Status, and Action. The table lists four runs with various statuses: RUNNING, KILLED, FAILED, and FINISHED.

Created At	Name	Experiment Repository	Start Time	Connection	Status	Action
2024-03-11T12:26:42.465960+00:00	Train RainClassifier-32	Classifier DeBERTa Era5		judoor	RUNNING	👁️ ✎️ 🗑️ ⋮
2024-03-11T15:7:14.636656+00:00	Train RainClassifier-11	Classifier DeBERTa Era5		judoor	KILLED	👁️ ✎️ 🗑️ ⋮
2024-03-08T16:11:53.085114+00:00	Train RainClassifier-30	Classifier DeBERTa Era5		jsc03	FAILED	👁️ ✎️ 🗑️ ⋮
2024-03-08T15:00:05.962845+00:00	Train RainClassifier-8	Classifier DeBERTa Era5		judoor	FINISHED	👁️ ✎️ 🗑️ ⋮



Conclusions

- Platform built as GUI for ease of use
 - General **access** and interaction with multiple **HPC** clusters
 - Allows developers to code applications **locally** on their laptop
 - Built in **reproducibility** of Runs
 - Real-life **tracking** of results via same interface
 - Enables **collaborative** development of projects
- Built on top of feature rich libraries
 - HPC APIs: FirecREST, UNICORE
 - Tracking: MLflow
- (Current) Limitations
 - Data not as flexible
 - Manual runtime setup required
- Planned features
 - Model deployment for inference with “one click”



<https://cloud.mantik.ai/>

Contacts:

kristian.ehlert@4-cast.de

markus.abel@ambrosys.de

MAELSTROM

www.maelstrom-eurohpc.eu



EuroHPC
Joint Undertaking



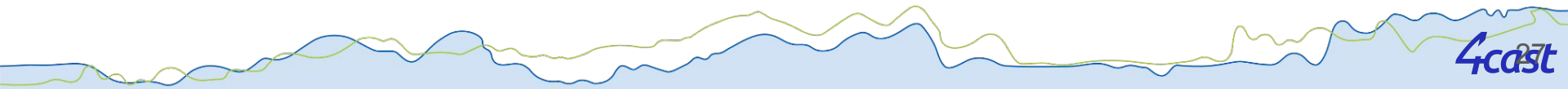
"The MAELSTROM project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No. 955515. The JU receives support from the European Union's Horizon 2020 research and innovation programme and United Kingdom, Germany, Italy, Luxembourg, Switzerland, Norway."

Coordinated by





Backup



Projects Page

mantik About Docs Projects Contact

Projects

+ CREATE PROJECT

Search
Maelstrom

LABELS
Add...

Datasets: Size
< 1 kB
< 1 MB
1 MB - 1 GB
1-10 GB
10-100 GB
100 GB - 1 TB
1 TB - 100 TB
100 TB - 1 PB

Maelstrom Application 2
from kristian.ehlert
Incorporate social media data into the prediction framework
[VIEW DETAILS](#)

Maelstrom Application 4
from saragrau
Improved ensemble predictions in forecast post-processing
Feature Extraction Image-to-Image Meteorology
[VIEW DETAILS](#)

Maelstrom Application 3
from saragrau
Neural network emulators for faster forecast models & data assimilation

Search *Projects*
by *Labels*

Scopes of Labels
are predefined

Labels are
grouped in
different classes

New Labels can
be suggested via
a Service Desk

Projects Page

Create a Project

PROJECT NAME *

SHORT DESCRIPTION *

LONG DESCRIPTION

B U I [List Icons] [Link Icon] [Table Icon]

Word Count: 0

Make this a Public Project

LABELS

add...

CANCEL CREATE

Description will show up in the *Overview of the Project*

Supports Markdown format

Project setup

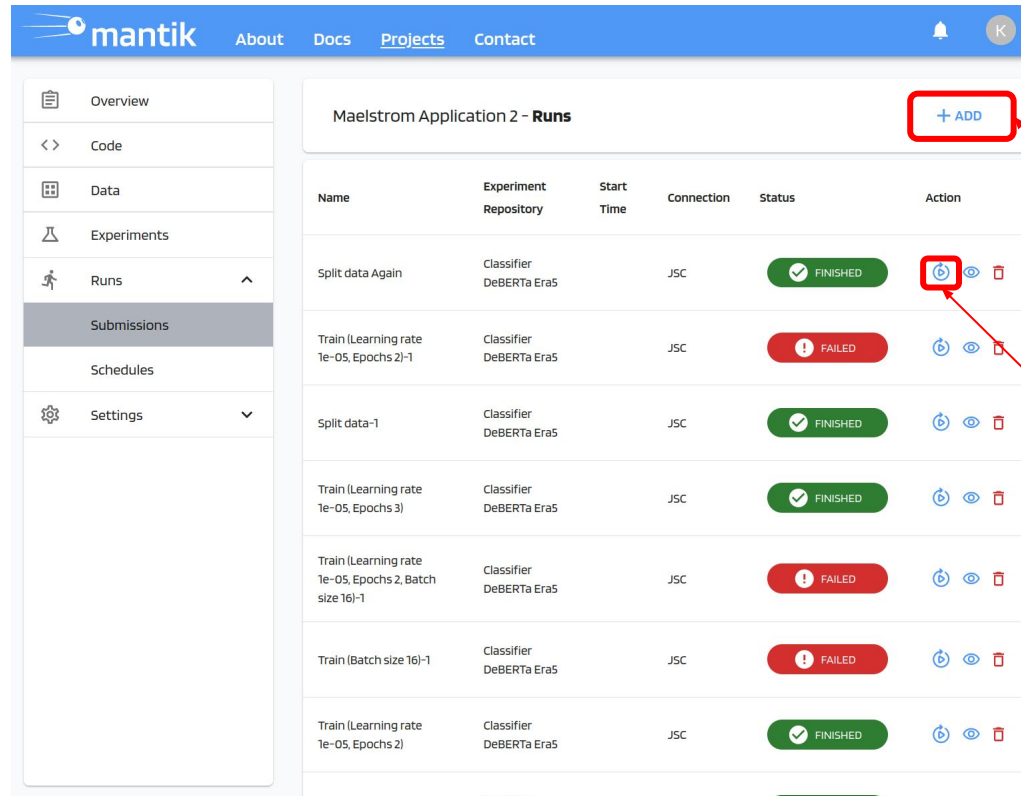
The screenshot shows the Mantik web interface. The top navigation bar includes 'mantik', 'About', 'Docs', 'Projects', and 'Contact'. The left sidebar has a menu with 'Overview', 'Code', 'Data', 'Experiments', 'Runs', and 'Settings'. The 'Data' and 'Runs' items are highlighted with red boxes. The main content area shows 'Maelstrom Application 2 - Code' with a '+ ADD' button. Below this, there is a section for 'Maelstrom A2 Github' with a link to 'https://github.com/4castRenewables/maelstrom-a2' and a red box around the link icon. The 'A2 - Social Media Data Analysis' section includes tags for 'Text Classification', 'Python', and 'MIT'.

Data management
Tab

Tracks history when
you use a Mantik CLI
command to move
data from your
machine to HPC

Link to your Git
code repository

Submitting a *Run*



Mantik About Docs Projects Contact

Maelstrom Application 2 - **Runs** + ADD

Name	Experiment Repository	Start Time	Connection	Status	Action
Split data Again	Classifier DeBERTa Era5		JSC	FINISHED	Re-Run View Delete
Train (Learning rate 1e-05, Epochs 2)-1	Classifier DeBERTa Era5		JSC	FAILED	Re-Run View Delete
Split data-1	Classifier DeBERTa Era5		JSC	FINISHED	Re-Run View Delete
Train (Learning rate 1e-05, Epochs 3)	Classifier DeBERTa Era5		JSC	FINISHED	Re-Run View Delete
Train (Learning rate 1e-05, Epochs 2, Batch size 16)-1	Classifier DeBERTa Era5		JSC	FAILED	Re-Run View Delete
Train (Batch size 16)-1	Classifier DeBERTa Era5		JSC	FAILED	Re-Run View Delete
Train (Learning rate 1e-05, Epochs 2)	Classifier DeBERTa Era5		JSC	FINISHED	Re-Run View Delete

Every Submission of a compute job to a HPC cluster is called a *Run*

Runs are grouped in *Experiments* (later more)

Re-Run Button

Change parameters and repeat an earlier *Run*

Submitting a Run

Every *Run* is part of a **mlflow** Experiment

Choose *Connection* of the cluster and the respective budget account

Create New Run

Select the parameters of your run

Name*

Experiment Description

Classifier DeBERTa Era5 (MLflow Experiment ID: 87)

Code Repository*

Maelstrom A2 Github (<https://github.com/4castRenewables/maelstrom-a2>)

Branch Name

main

Relative Path to MLflow Project File*

Relative path to Backend Config*

Connection

JSC

UNICORE Compute Budget Account*

deepacf

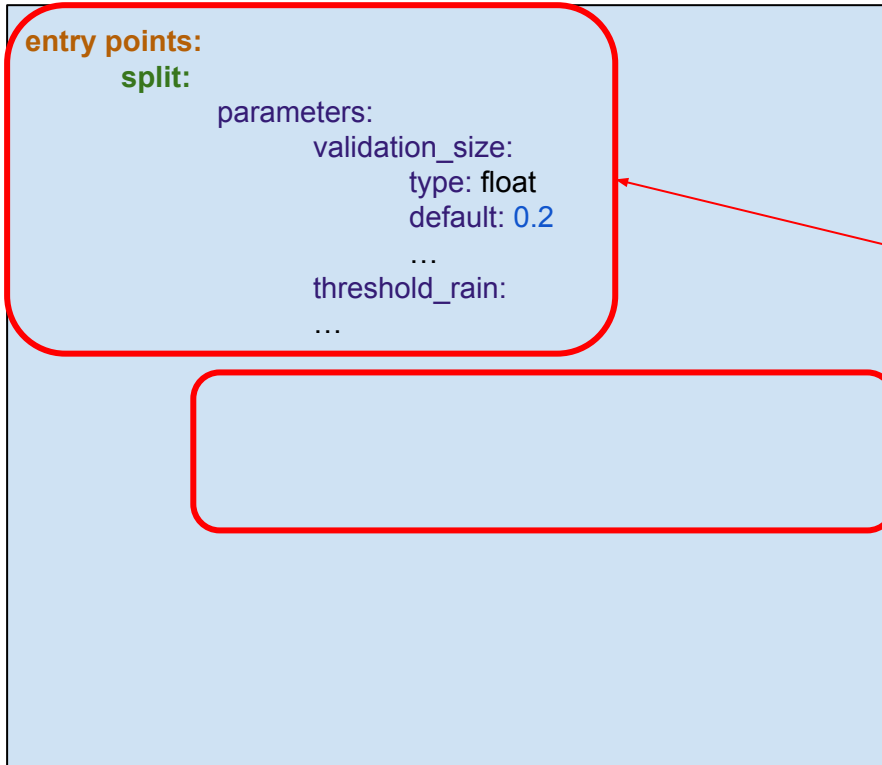
Please ensure that all fields with * are filled out.

CANCEL RUN!

Mantik needs to find the code for submission

Most important:
Path to the **2 necessary configuration files** in the Code repository



Configuration File I - MLproject



Entry points specify a pre-defined scenario through parameters that will be handed over to the respective script

Actual python command that executes the script

Configuration File I - MLproject

 The file has to be named 'MLproject' 

```
entry points:  
  split:  
    parameters:  
      validation_size:  
        type: float  
        default: 0.2  
        ...  
      threshold_rain:  
        ...  
    Command: >  
      python build.py  
        -- validation_size (validation_size)  
        -- ...  
  train:  
    parameters: ...  
    Command: >  
    ...  
  benchmark: ...
```

Entry points specify a pre-defined scenario through parameters that will be handed over to the respective script

Actual python command that executes the script

Additional entry points addressing other scripts (for example to Benchmark a *Run*)

Updated a *Run* form

Create New Run

Select the parameters of your run

Name *

Experiment Repository *

Classifier DeBERTa Era5 (MLflow Experiment ID: 87)

Code Repository *

Maelstrom A2 Github (<https://github.com/4castRenewables/maelstrom-a2>)

Branch Name

main

Relative Path to MLflow Project File *

Relative path to Backend Config *

Connection

JSC

UNICORE Compute Budget Account *

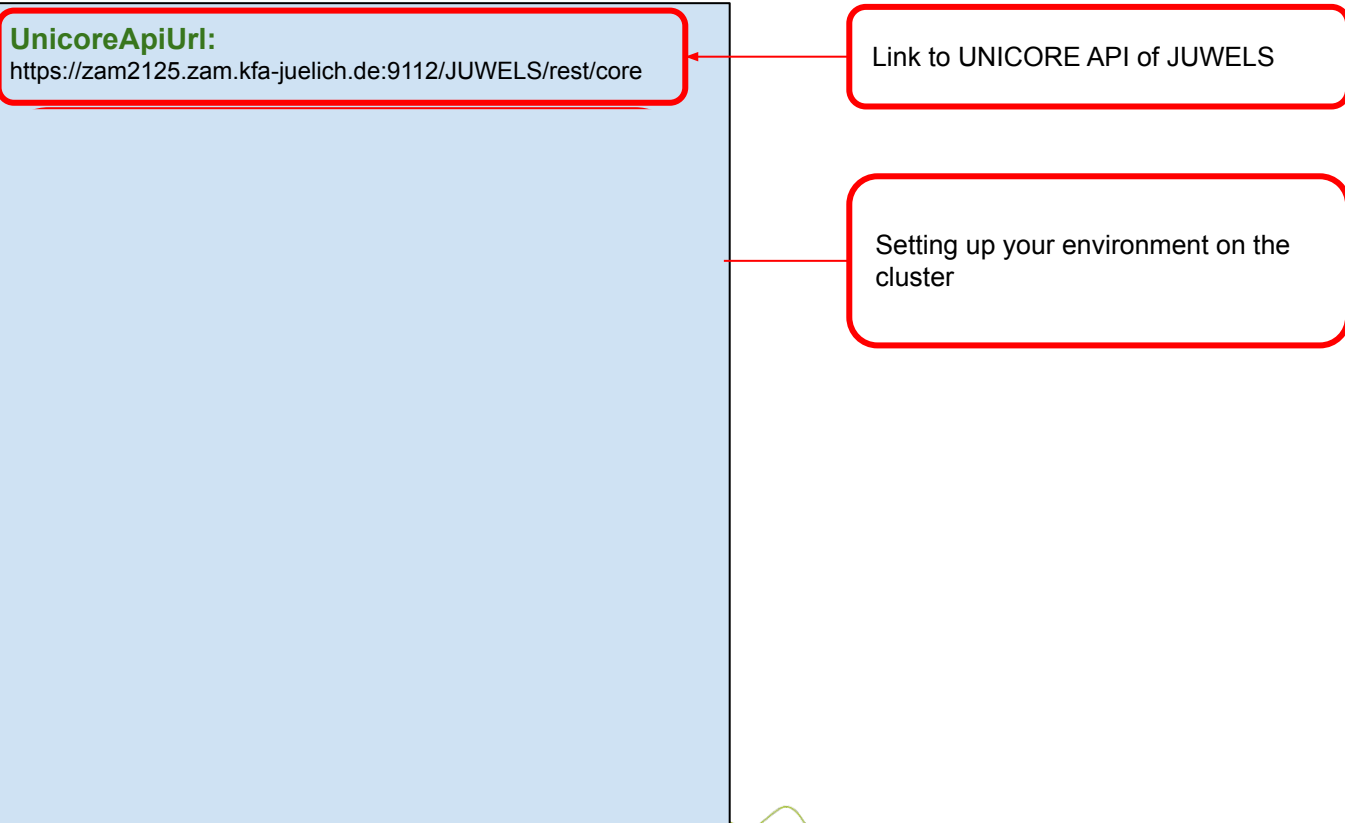
deepacf

Please ensure that all fields with * are filled out.

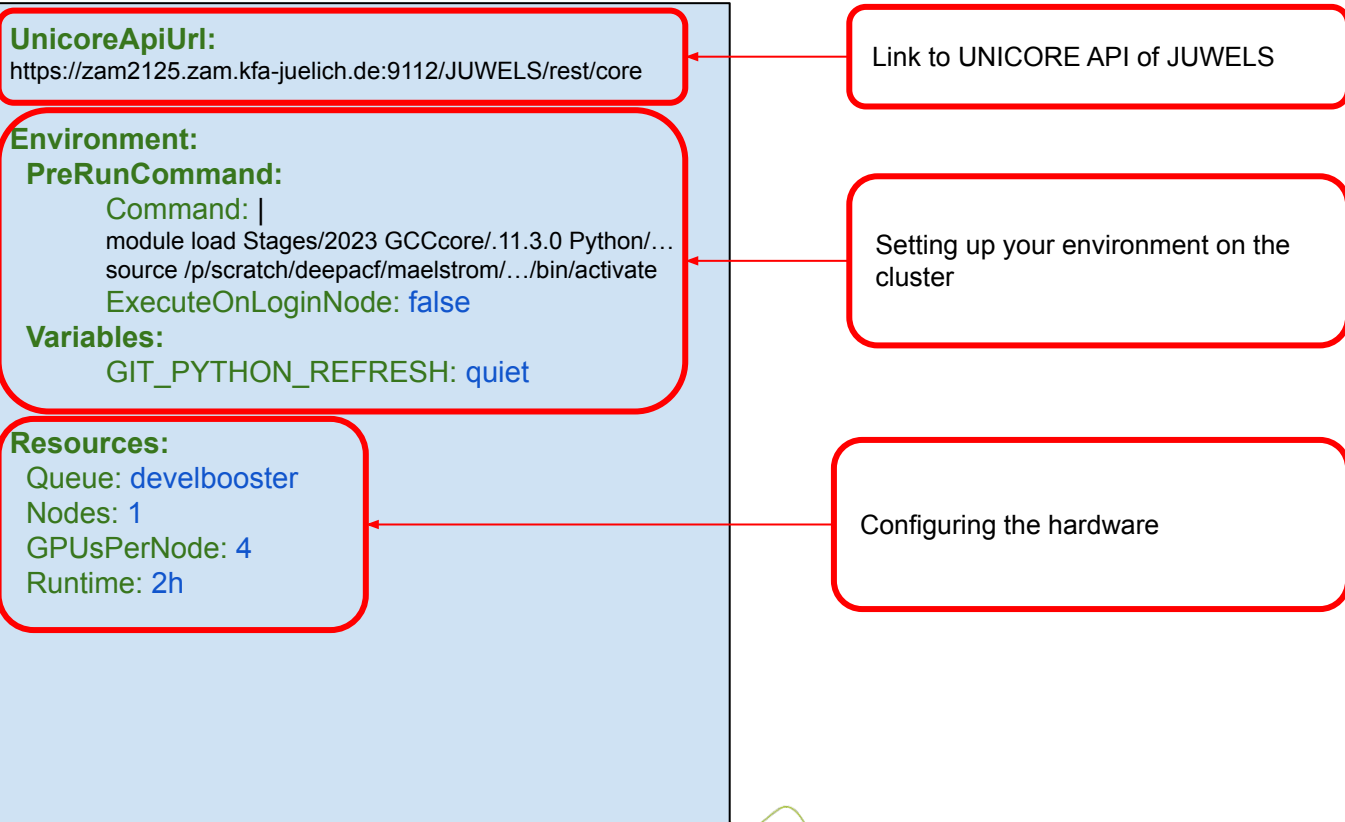
CANCEL RUN!

Mantik identifies the parameters specified in the MLproject config file and updates the form accordingly

Configuration File II - Backend



Configuration File II - Backend



Updated a *Run* form

Create New Run

Select the parameters of your run

Name *

Experiment Repository *
Classifier DeBERTa Era5 (MLflow Experiment ID: 87)

Code Repository *
Maelstrom A2 Github (<https://github.com/4castRenewables/maelstrom-a2>)

Branch Name
main

Relative Path to MLflow Project File *

Relative path to Backend Config *

Connection
JSC

UNICORE Compute Budget Account *
deepacf

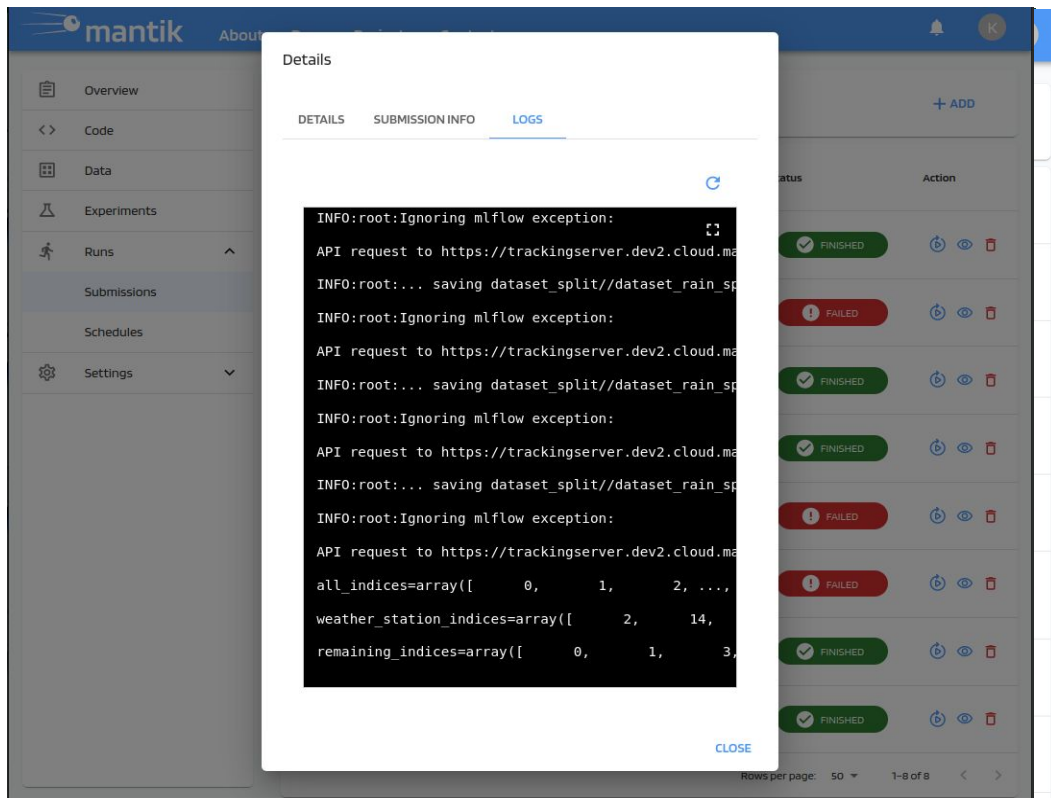
Please ensure that all fields with * are filled out.

CANCEL RUN!

Mantik adds content of backend config to the *Run* form

Enables adjustments to the HPC configuration on the fly

Run Details & Creating Run Schedules

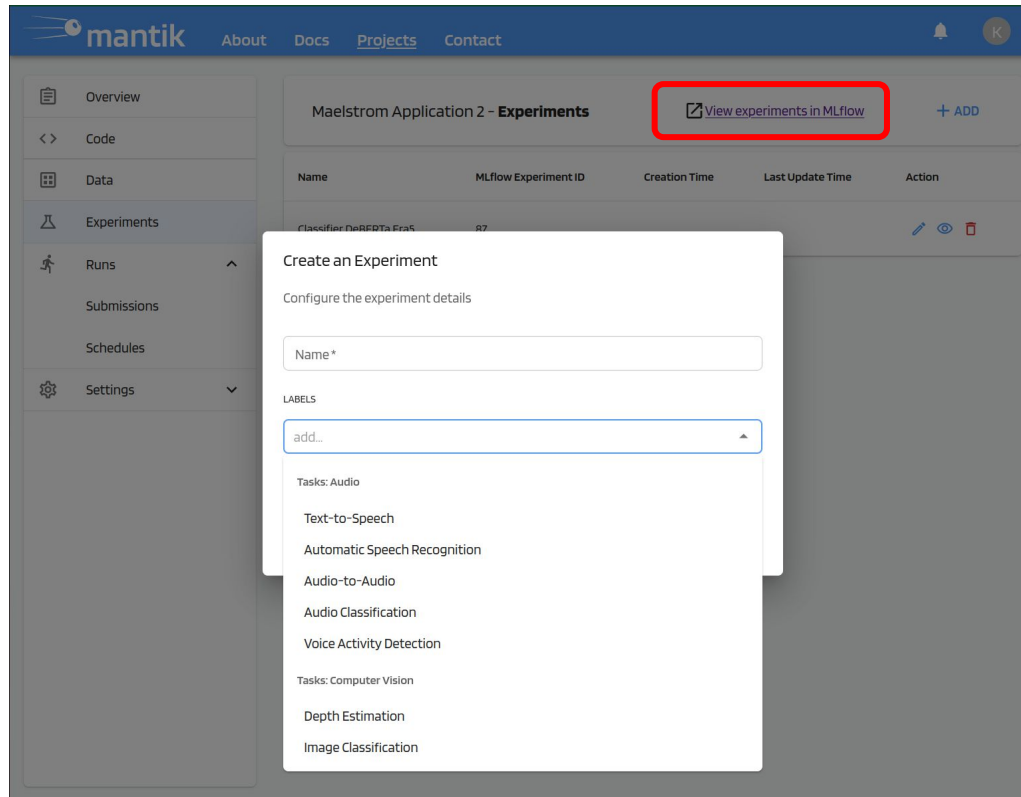


The screenshot displays the Mantik web interface. A 'Details' modal window is open, showing a log viewer with the following content:

```
INFO:root:Ignoring mlflow exception:  
API request to https://trackingserver.dev2.cloud.ma  
INFO:root:... saving dataset_split//dataset_rain_sp  
INFO:root:Ignoring mlflow exception:  
API request to https://trackingserver.dev2.cloud.ma  
INFO:root:... saving dataset_split//dataset_rain_sp  
INFO:root:Ignoring mlflow exception:  
API request to https://trackingserver.dev2.cloud.ma  
INFO:root:... saving dataset_split//dataset_rain_sp  
INFO:root:Ignoring mlflow exception:  
API request to https://trackingserver.dev2.cloud.ma  
INFO:root:... saving dataset_split//dataset_rain_sp  
INFO:root:Ignoring mlflow exception:  
API request to https://trackingserver.dev2.cloud.ma  
all_indices=array([ 0, 1, 2, ...,  
weather_station_indices=array([ 2, 14,  
remaining_indices=array([ 0, 1, 3,
```

The background interface shows a sidebar with navigation options: Overview, Code, Data, Experiments, Runs, Submissions, Schedules, and Settings. The main area displays a table of runs with columns for status and action. The status column shows a mix of 'FINISHED' (green) and 'FAILED' (red) entries. The action column contains icons for refresh, view, and delete.

Embedding *mlflow* in Mantik



The screenshot displays the Mantik web application interface. The top navigation bar includes 'mantik', 'About', 'Docs', 'Projects', and 'Contact'. The main content area is titled 'Maelstrom Application 2 - Experiments' and features a '+ ADD' button and a button labeled 'View experiments in MLflow' which is highlighted with a red box. Below this is a table with columns for 'Name', 'MLflow Experiment ID', 'Creation Time', 'Last Update Time', and 'Action'. A modal window titled 'Create an Experiment' is open, prompting the user to 'Configure the experiment details'. It contains a 'Name*' input field, a 'LABELS' dropdown menu with 'add...' selected, and a list of task categories: 'Tasks: Audio' (Text-to-Speech, Automatic Speech Recognition, Audio-to-Audio, Audio Classification, Voice Activity Detection) and 'Tasks: Computer Vision' (Depth Estimation, Image Classification).

Embedding **mlflow** in Mantik



Tracking of experiments including

- Input parameters
- Metrics
- Artifacts (e.g. Figures)
- Models



in real-time on
cloud.mantik.ai/mlflow/



Visualization in MLflow GUI



Model versioning



Inference with versioned models

Classifier DeBERTa Era5

Provide Feedback

Share

Experiment ID: 87 Artifact Location: mlflow-artifacts:/87

> Description Edit

Table view

Chart view

metrics.rmse < 1 and params.model = "tree"



Sort: Created



Refresh

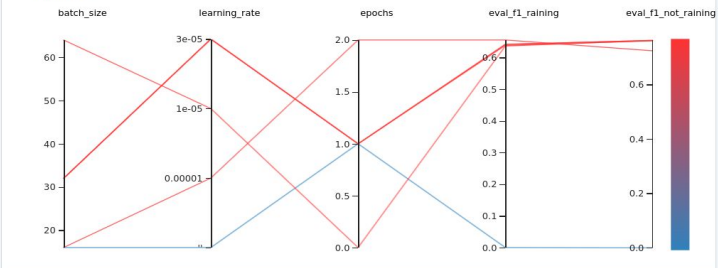
Time created: All time

State: Active

Run Name
Train (Batch size 16)
Train (Learning rate 1e-05, Epochs ...)
Train (Learning rate 1e-05, Epochs ...)
Train (Learning rate 1e-05, Epochs 2)
Train 2h
train
split-data
data split-1
data split
split data-2
split data-1

12 matching runs

Comparing 6 runs



epoch

Comparing first 6 runs