

Corso: Big Data Architecture

Machine Learning OPerationS with ClearML @SNAP4City



Roadmap

1. Main advantages of MLOPS
2. Adapting ClearML to Snap4City Architecture
3. Basic concepts of ClearML
4. Getting started demo for ClearML at Snap4City

Why MLOPS?

Main advantages in **orchestrating a cluster of GPUs**:

- Workload **distribution**
- **Parallel** execution
- Avoiding bottlenecks guarantees execution **speed**



UNIVERSITÀ
DEGLI STUDI
FIRENZE

DINFO
DIPARTIMENTO DI
INGEGNERIA
DELL'INFORMAZIONE

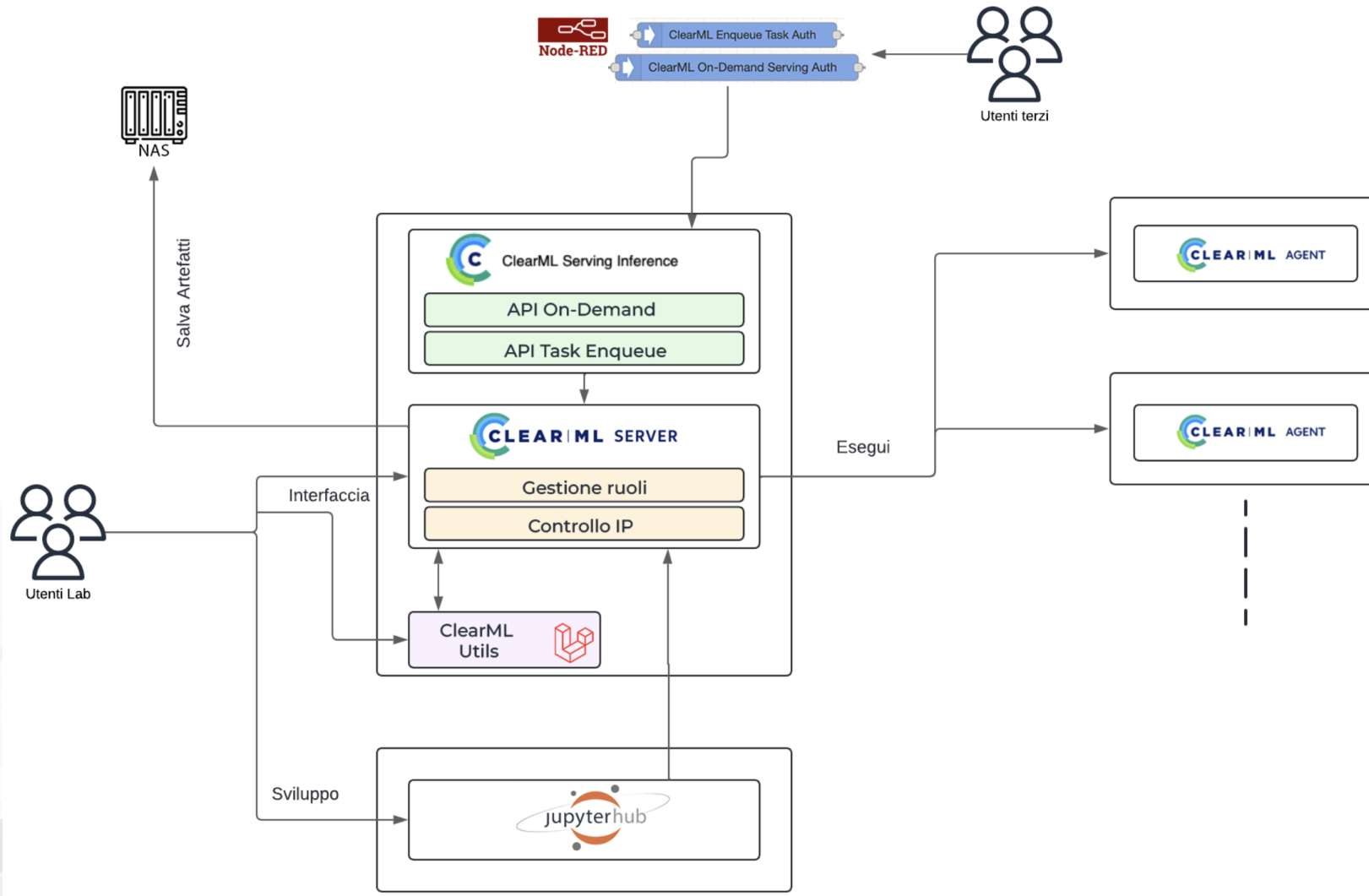
DISIT
DISTRIBUTED SYSTEMS AND
INTERNET TECHNOLOGIES LAB
DISTRIBUTED DATA INTELLIGENCE
AND TECHNOLOGIES LAB



Section I

CLEARML ARCHITECTURE AT SNAP4CITY

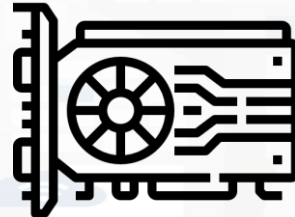
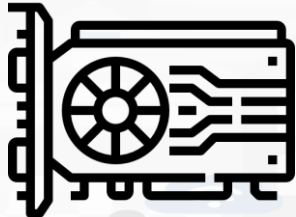
ClearML - Snap4City Architecture



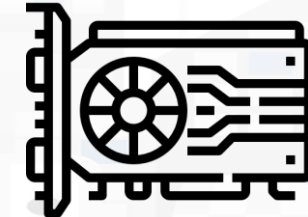
ClearML Agent

ClearML Server is used in order to orchestrate the execution of tasks in the **Snap4City GPU cluster**.

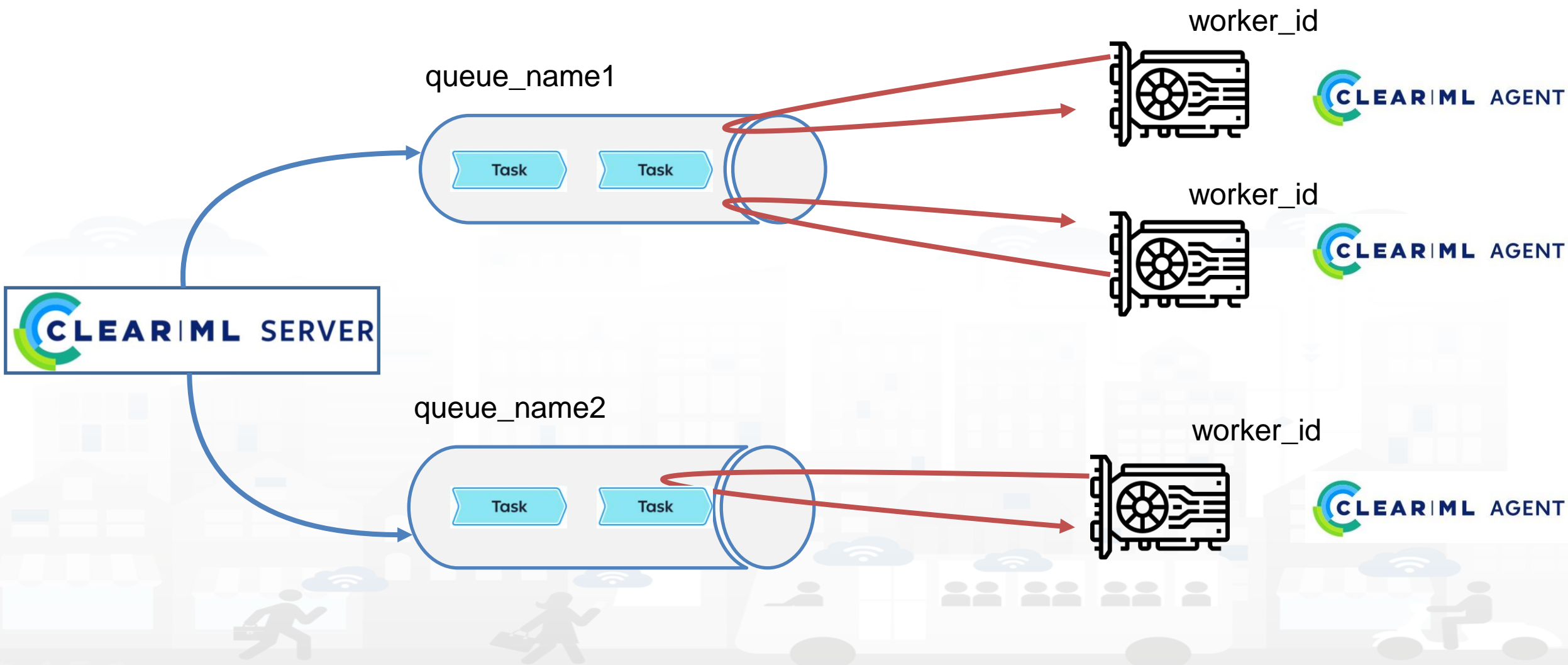
On each of those machines an instance of **ClearML Agent** is installed. This allows the execution of tasks and ensures workload distribution.



...



Workers & Queues

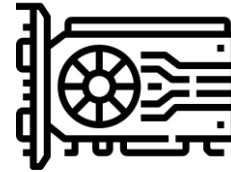


Agent execution

queue_name



worker_id



CLEAR|ML AGENT



CLEAR|ML SERVER



Roles definition and security management

By default **ClearML** allows every user with credentials to create tasks from all devices.



ClearML for **Snap4City** has been designed in order to guarantee that only verified users can access to the GPU cluster. IP verification via ClearML Server implemented.

ClearML allows every user to visualize every project and visualize queues.



Defined the **Admin** role in order to differentiate users with privileges for monitoring purpose and guarantee privacy of common users.



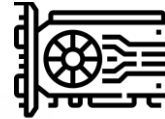
A live **notification system** has been designed to ensure the monitoring of tasks execution and the **resources availability**.

What about inference?

Can we use
your
models?



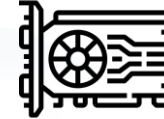
CLEAR|ML AGENT



CLEAR|ML SERVER



CLEAR|ML AGENT

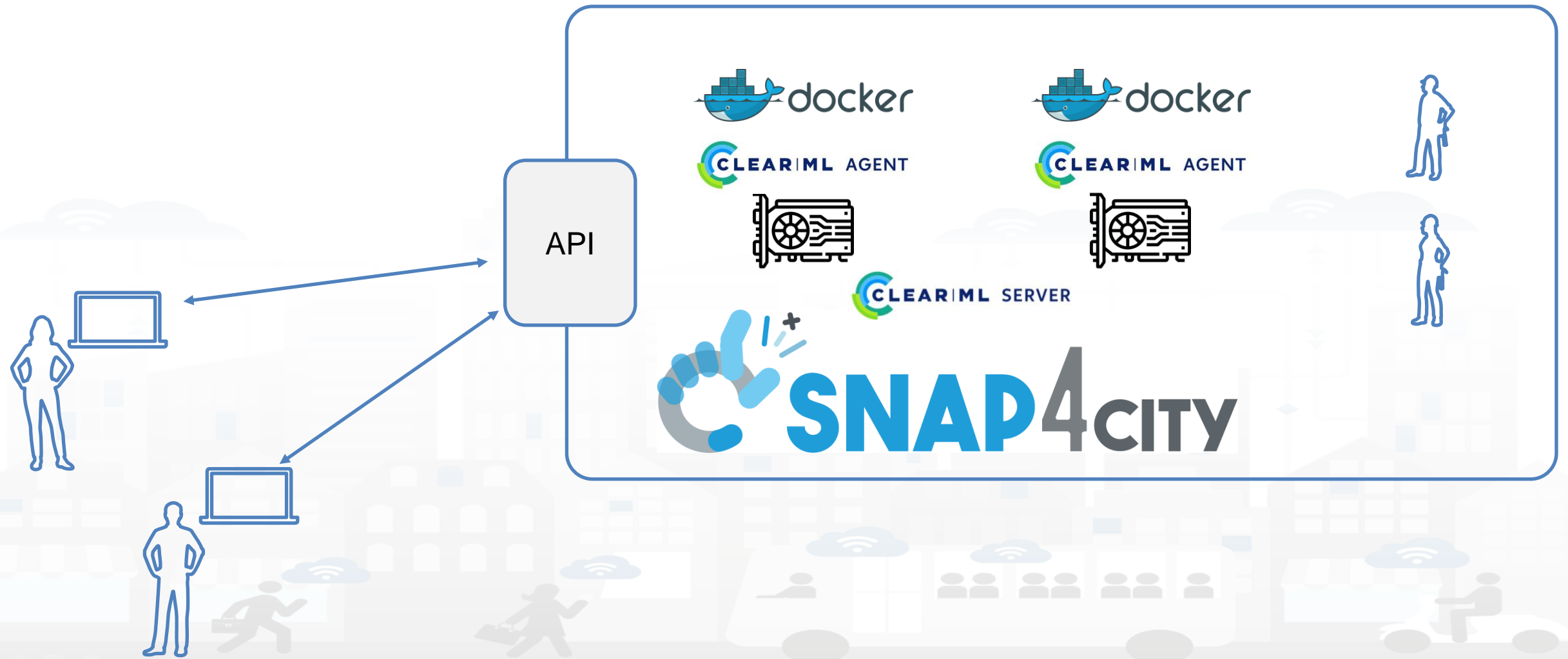


What about inference?

Can we use
your
models?

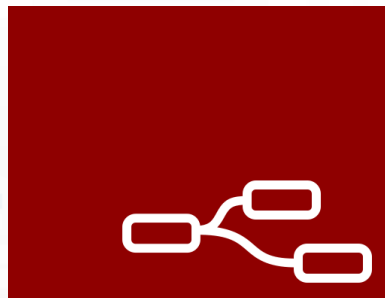


What about inference?



Providing inference services

- 1.API Serving On Demand:** real time inference on **pre-trained models**.
- 2.API Task Enqueue:** allowing user to send **task to queues** without knowing the architecture of the system.



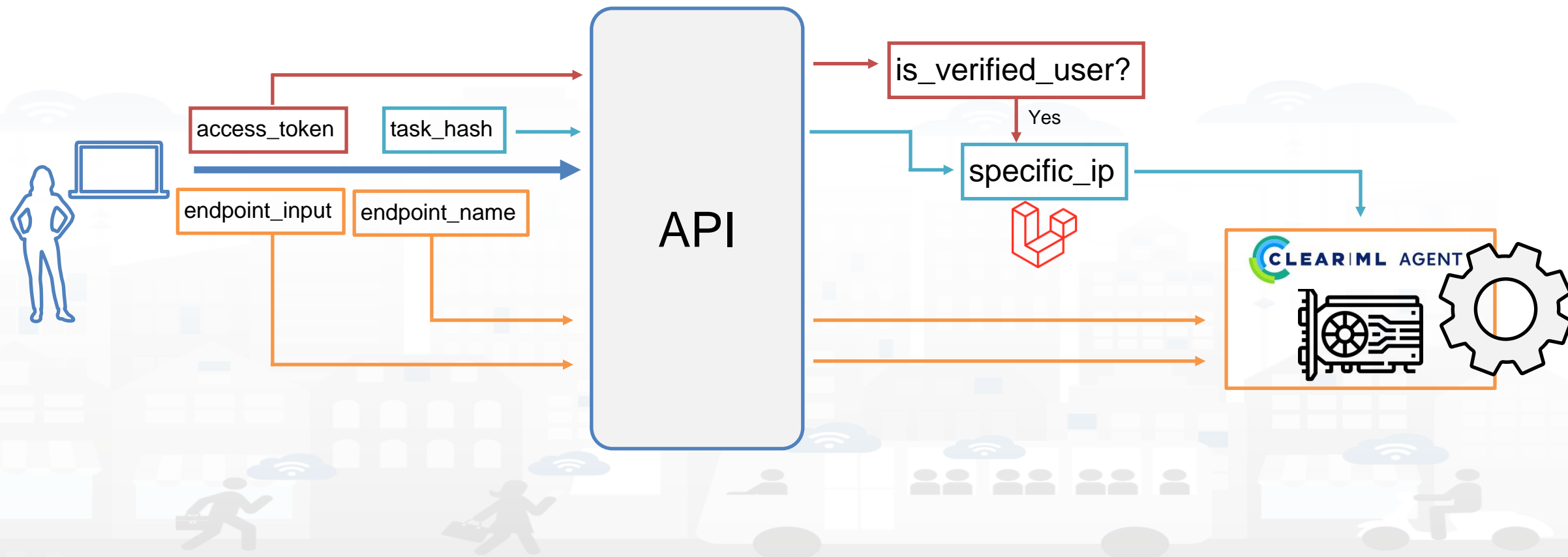
Node-RED

ClearML
Utils



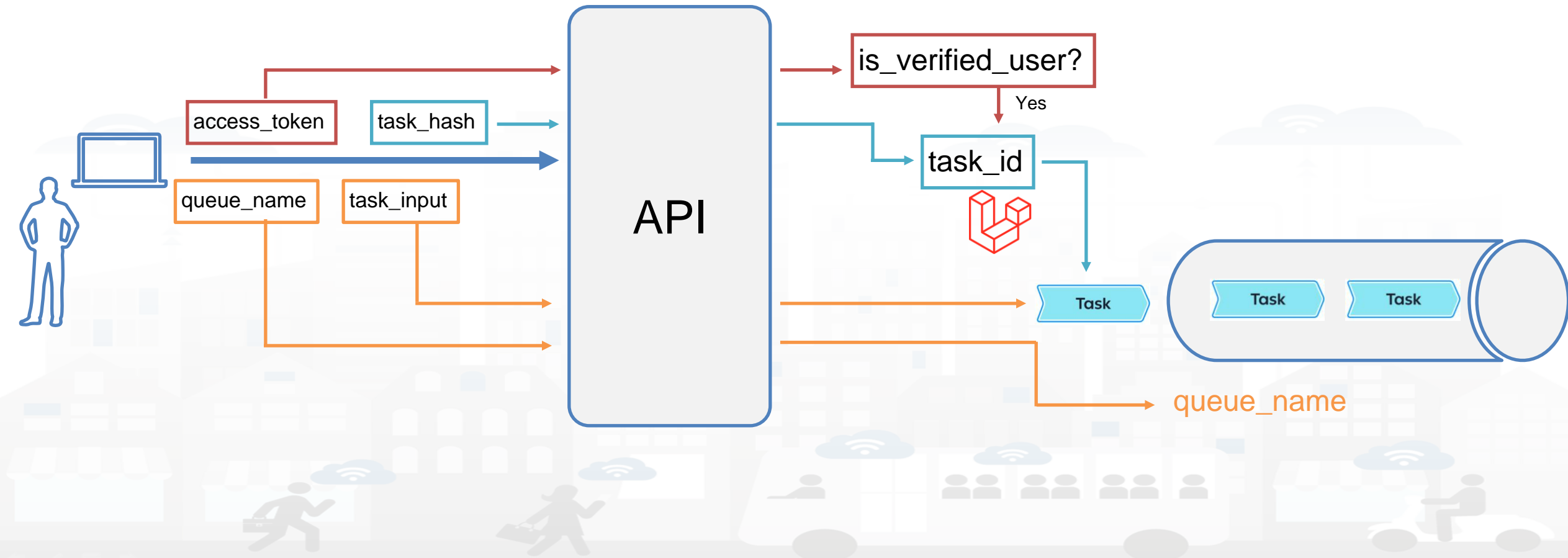
API On Demand

Real time inference on **pre-trained** models



API Task Enqueue

Send task to **queues**



ClearML Utils: What and How

Developed for internal use only:

- **Calls routing**
- **resources handling** for API serving and task enqueue
- **monitoring** calls to services and log tracking/visualization

ClearML
Utils



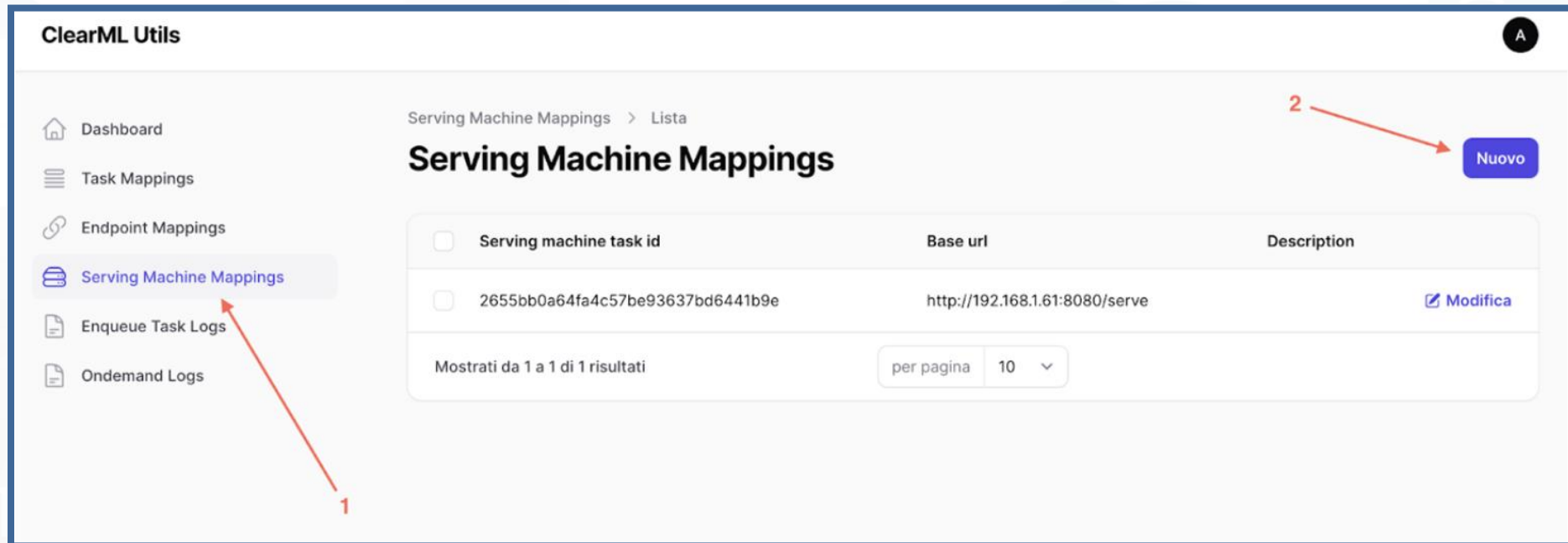
ClearML Utils: Dashboard

Shows **available resources** and **current usages** of services:

- **Number** of **machines** configured for **on-demand serving**
- **Number** of **endpoints** for **on-demand serving**
- **Number** of **available tasks** for **task enqueue**
- Visualization tool for **monitoring the requests trend** on both on-demand and task enqueue

ClearML Utils: Serving

Machine task id is used in order to **map** the specific **machine ip** with the **serving task**. This is done in order to ensure that **the right model** is called during serving.



ClearML Utils

Serving Machine Mappings > Lista

Serving Machine Mappings

☐ Serving machine task id Base url Description

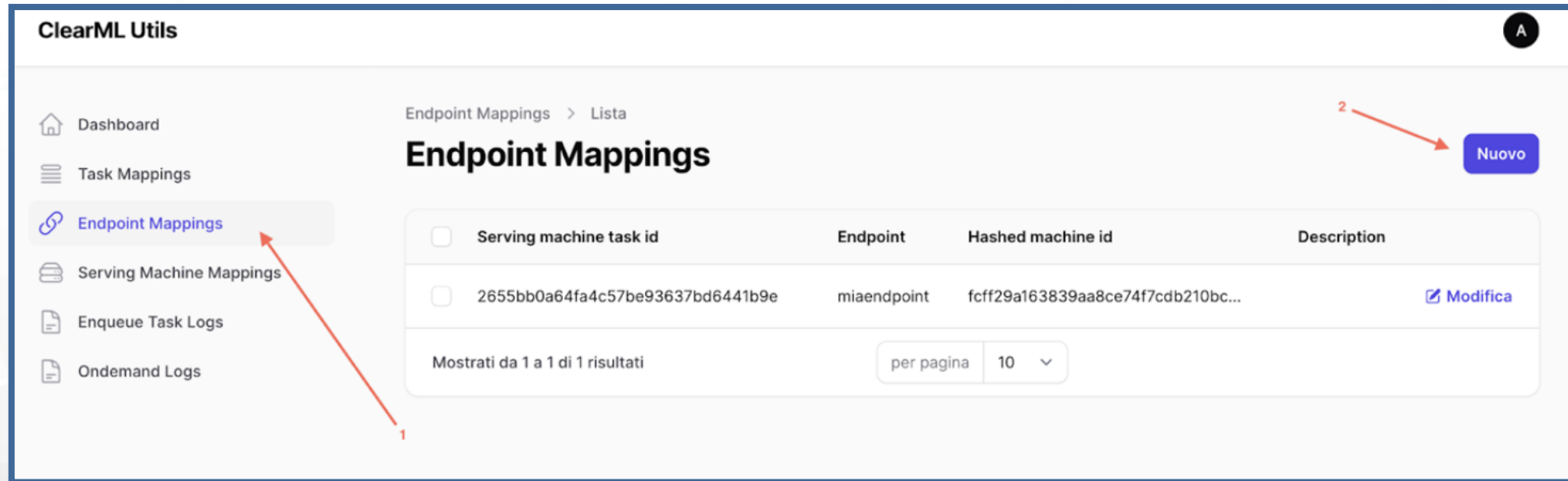
☐ 2655bb0a64fa4c57be93637bd6441b9e http://192.168.1.61:8080/serve [Modifica](#)

Mostrati da 1 a 1 di 1 risultati per pagina 10

ClearML Utils: On demand endpoints

Developers create **endpoints** using ClearML Serving then:

1. Define the **endpoint name** and the associated **task_id**
2. An **hashing** of the **serving task** is **generated**, external users don't need to know the real task_id



ClearML Utils

Endpoint Mappings > Lista

Endpoint Mappings

<input type="checkbox"/> Serving machine task id	Endpoint	Hashed machine id	Description
<input type="checkbox"/> 2655bb0a64fa4c57be93637bd6441b9e	miaendpoint	fcff29a163839aa8ce74f7cdb210bc...	Modifica

Mostrati da 1 a 1 di 1 risultati

per pagina 10

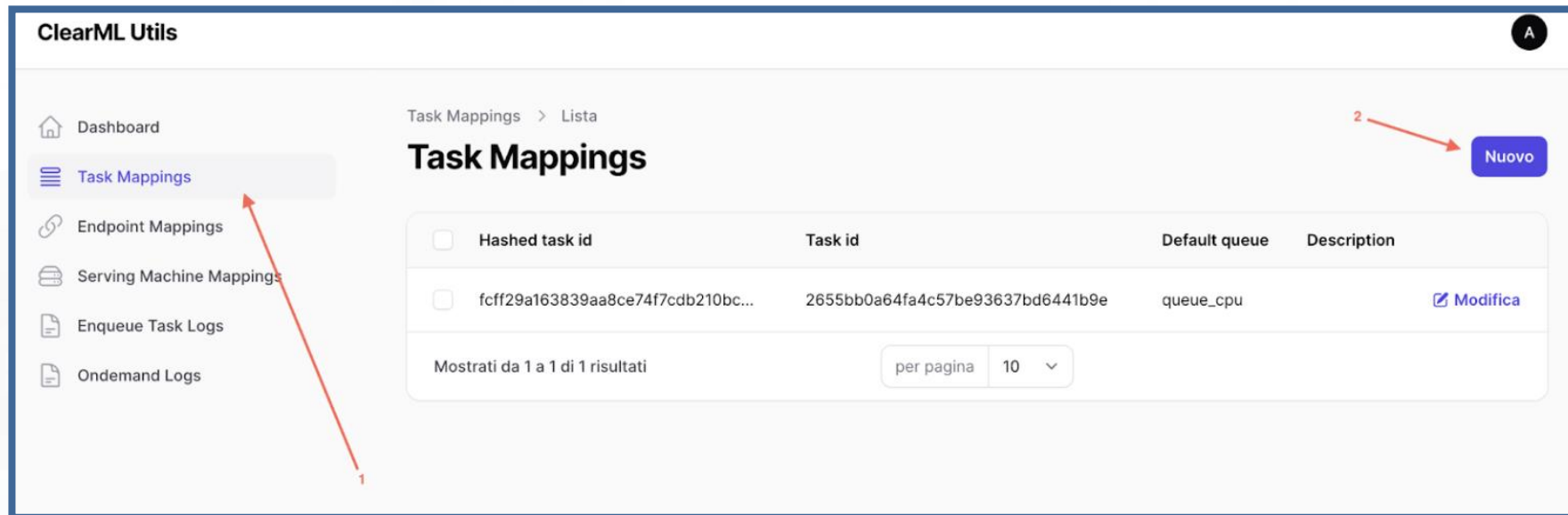
[Nuovo](#)

ClearML Utils: Task enqueue

Task enqueue need to clone a task and send it to the right queue.

Developers:

1. Insert **task id** and the **queue name**
2. App generates a **task hash id** that allows the user to **enqueue a task**



ClearML Utils

Task Mappings > Lista

Task Mappings

<input type="checkbox"/> Hashed task id	Task id	Default queue	Description
<input type="checkbox"/> fcff29a163839aa8ce74f7cdb210bc...	2655bb0a64fa4c57be93637bd6441b9e	queue_cpu	Modifica

Mostrati da 1 a 1 di 1 risultati

per pagina 10

ClearML Utils: On Demand Logging

ClearML Utils

Dashboard

Task Mappings

Endpoint Mappings

Serving Machine Mappings

Enqueue Task Logs

Ondemand Logs

Ondemand Logs > Lista

Serving machine task id

b746714516a7409bbaa68a8402a5c1a2

Endpoint

nome_endpoint

Status

✓

Created at

10/07/24 12:31:45

Vedi

Mostrati da 1 a 1 di 1 risultati

per pagina

10

1

2

ClearML Utils

Dashboard

Task Mappings

Endpoint Mappings

Serving Machine Mappings

Enqueue Task Logs

Ondemand Logs

Ondemand Logs > Guarda

Guarda Ondemand Log

Data Creazione

10/07/2024, 12:31:45

Nome Endpoint

nome_endpoint

Dettaglio

Serving Machine Task ID

b746714516a7409bbaa68a8402a5c1a2

Successo

✓ Yes

✗ No

ClearML Utils: Task Enqueue Logging

ClearML Utils

Enqueue Task Logs > Lista

Enqueue Task Logs

<input type="checkbox"/> Task id	Status	Created at	
<input type="checkbox"/> 2655bb0a64fa4c57be93637bd6441b9e	✓	10/07/24 10:20:02	👁 Vedi

Mostrati da 1 a 1 di 1 risultati

per pagina 10

1

2

ClearML Utils

Enqueue Task Logs > Guarda

Guarda Enqueue Task Log

Data Creazione: 10/07/2024, 10:20:02

Task ID: 2655bb0a64fa4c57be93637bd6441b9e

Dettaglio:

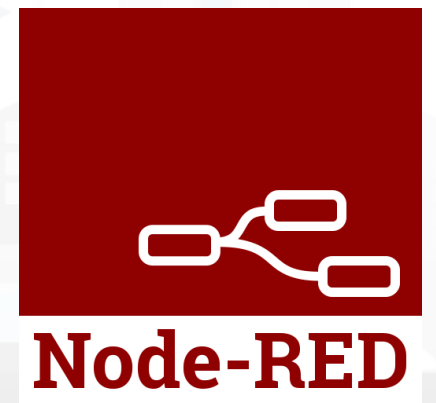
Successo

✓ Yes ✕ No

Node-RED Microservices

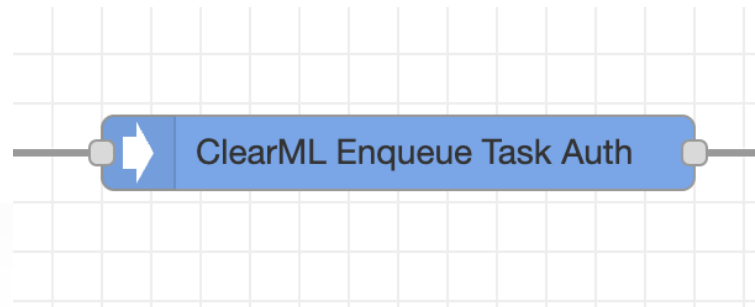
Programming **microservices** using **visual programming** with node.js

- **Task Enqueue block:** sending a task to the queue using json
- **On Demand serving block:** call immediate inference using json





Node-RED: Task Enqueue block

```
{
  "task_id": "hash_task_id",
  "queue_name": "queue",
  "input": {
    "param1": "value_1",
    "param2": "value_2"
  }
}
```



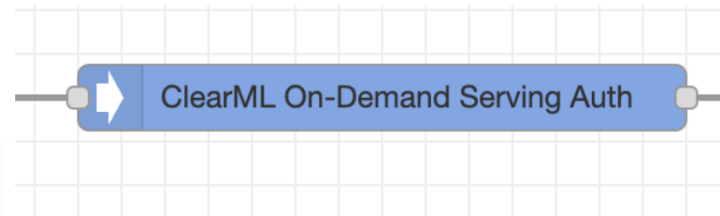
```
{
  "status": "ok"
}
```

Authentication	none		
task_id	my_hashed_task_id		
queue_name	my_queue		
input	{ }		

Robust Authentication implemented using authentication_token requested by the block and provided by Snap4City

Node-RED: On Demand block

```
{
  "machine_id": "serving_hash",
  "endpoint": "endpoint_name",
  "input": {
    "param1": "value_1",
    "param2": "value_2"
  }
}
```



```
{
  "status": "ok"
}
```

Authentication
none

machine_id
serving machine task id

endpoint
endpoint service id

input
{}

Robust Authentication implemented using authentication_token requested by the block and provided by Snap4City



UNIVERSITÀ
DEGLI STUDI
FIRENZE

DINFO
DIPARTIMENTO DI
INGEGNERIA
DELL'INFORMAZIONE

DISIT
DISTRIBUTED SYSTEMS AND
INTERNET TECHNOLOGIES LAB
DISTRIBUTED DATA INTELLIGENCE
AND TECHNOLOGIES LAB



Section II

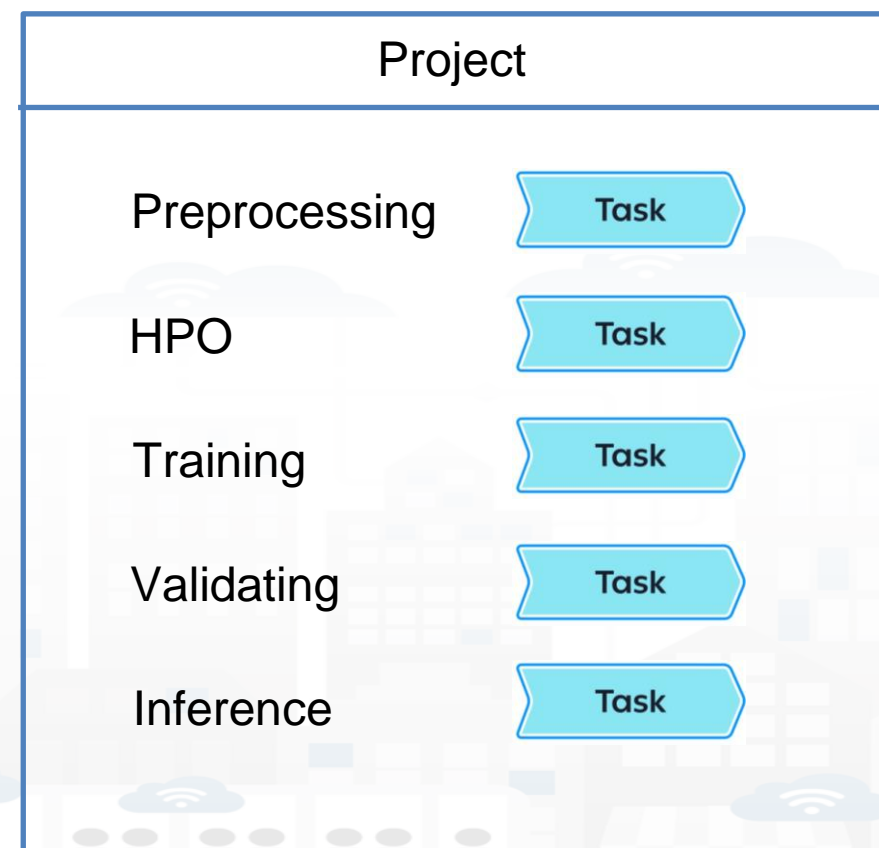
CLEARML FUNDAMENTAL CONCEPTS



Tasks and Projects

We refer to **Task** as the core ClearML component. Can be viewed as an **experiment instance**.

It is composed of all that code that we are interested in tracking (a training phase for example). Must be **uniquely identified** and all its code must be recoverable and reproducible.



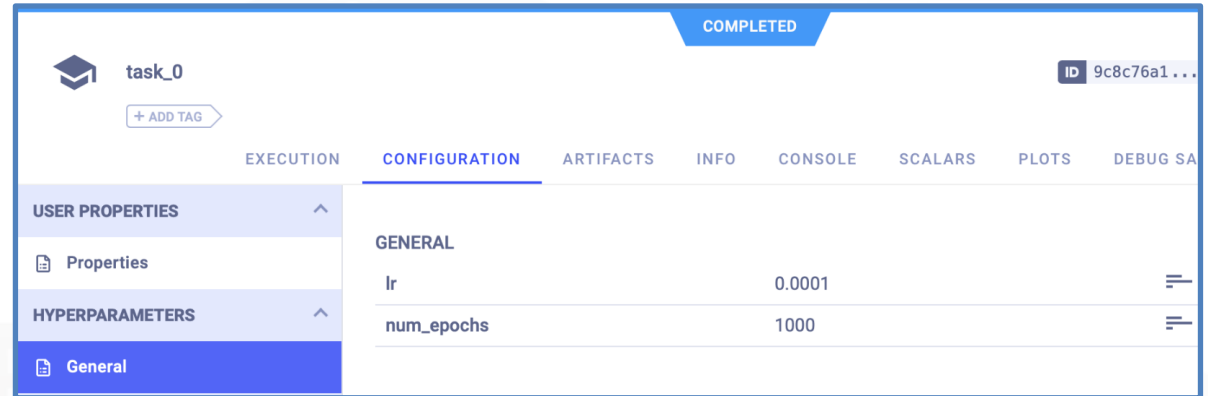
Hyperparameters

ClearML provides **Hyperparameters** handling.

Hyperparams are viewed as experiment (so **Task**) **configurations**.

The user must be able to run a task on a starting set of hyperparameters values, check all the results (so **Artifacts**) associated with those values.

```
hyperparams = {'num_epochs': 1000, 'lr': 0.0001}  
task.connect(hyperparams)
```



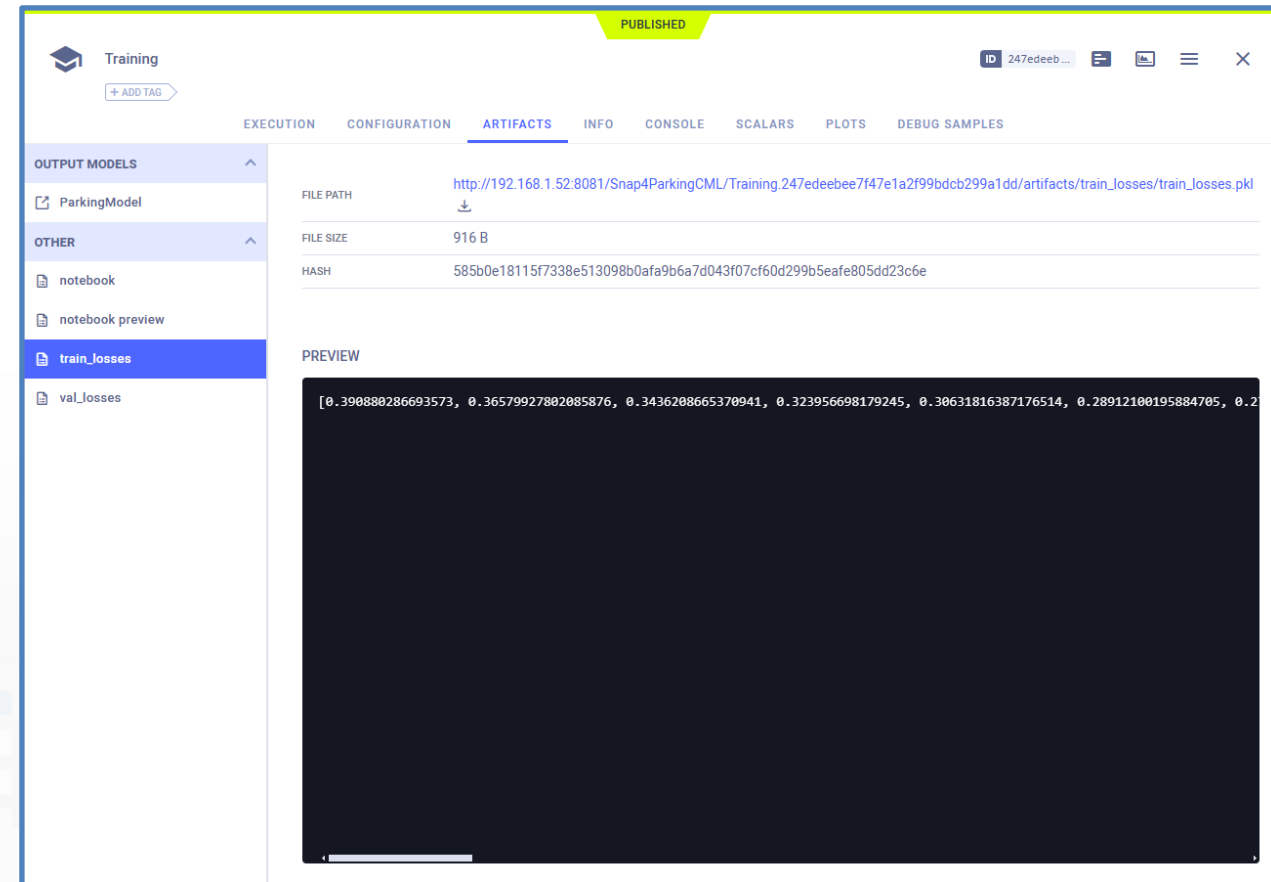
The image shows a screenshot of the ClearML web interface for a task named 'task_0'. The task status is 'COMPLETED' and its ID is '9c8c76a1...'. The 'CONFIGURATION' tab is selected, showing a table of hyperparameters. The table has columns for the parameter name, its value, and a menu icon. The parameters listed are 'lr' with a value of 0.0001 and 'num_epochs' with a value of 1000. On the left sidebar, there are sections for 'USER PROPERTIES' and 'HYPERPARAMETERS', with 'General' selected under the latter.

task_0		ID
		9c8c76a1...
EXECUTION		
CONFIGURATION		
ARTIFACTS		
INFO		
CONSOLE		
SCALARS		
PLOTS		
DEBUG SA		
USER PROPERTIES		
Properties		
HYPERPARAMETERS		
General		
GENERAL		
lr	0.0001	
num_epochs	1000	

Artifacts

Code reproducibility doesn't relies only on recoverable lines of code (it's not code versioning). Reproducibility means that ClearML must guarantee to the user the ability to **store, visualizing, recovering** and eventually **manipulate** everything the **code produces.**

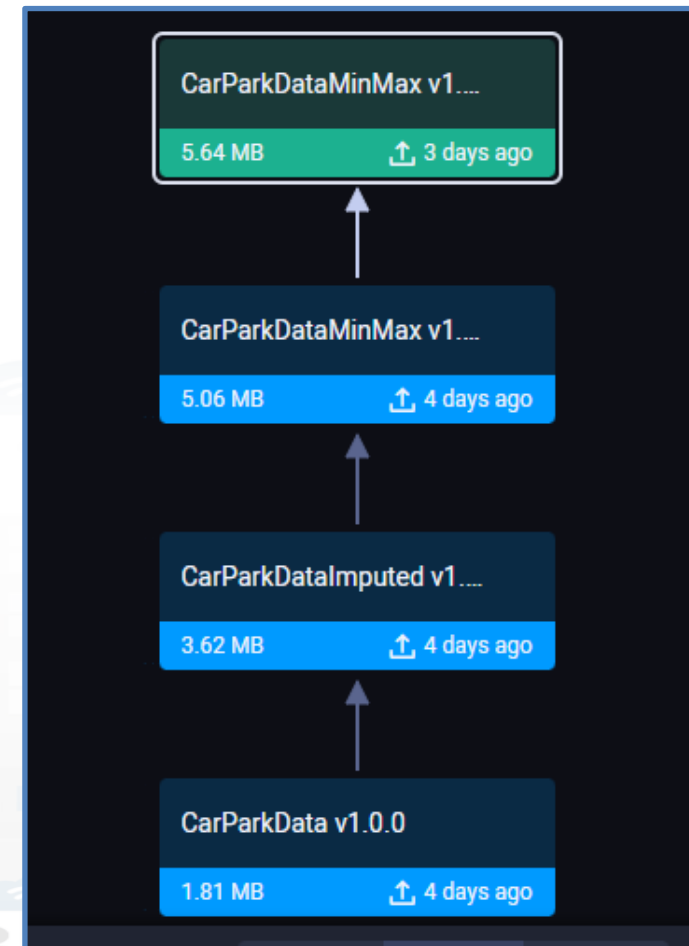
Artifacts are basically the outputs of a Task.



Dataset versioning

ClearML allows developers to **create, update, upload** and **retrieve datasets**.

Developers can create **different versions** of the same dataset with incremental approach while **keeping track** of every update.






Model deployment

OVERVIEWEXPERIMENTSMODELS

PUBLISHED

 Training - ParkingModel

ID 0ed5847a ...

+ ADD TAG


GENERALNETWORKLABELSMETADATALINEAGESCALARSPLOTS

CREATED AT:Nov 29 2024 15:21

UPDATED AT:Nov 29 2024 15:21

FRAMEWORK:PyTorch

STATUS:Published

MODEL URL:'Snap4ParkingCML/Training.247edeebee7f47e1a2f99bdc299a1dd/models/ParkingModel.pth' 

USER:ClearUser14

ARCHIVED:No

PROJECT:Snap4ParkingCML

DESCRIPTION:Created by task id: 247edeebee7f47e1a2f99bdc299a1dd



UNIVERSITÀ
DEGLI STUDI
FIRENZE

DINFO
DIPARTIMENTO DI
INGEGNERIA
DELL'INFORMAZIONE

DISIT
DISTRIBUTED SYSTEMS AND
INTERNET TECHNOLOGIES LAB
DISTRIBUTED DATA INTELLIGENCE
AND TECHNOLOGIES LAB



Section III

A GETTING STARTED DEMO





UNIVERSITÀ
DEGLI STUDI
FIRENZE

DINFO
DIPARTIMENTO DI
INGEGNERIA
DELL'INFORMAZIONE

DISIT
DISTRIBUTED SYSTEMS AND
INTERNET TECHNOLOGIES LAB
DISTRIBUTED DATA INTELLIGENCE
AND TECHNOLOGIES LAB

SNAP4CITY



Authentication with Snap4City

Snap4City

Switch To New Layout (Beta)

User: clearmldemo, Org: DISIT
Role: AreaManager, Level: 0
Logout

My Snap4City.org

Tour Again

www.snap4solutions.org

Dashboards (Public)

Dashboards of My Organization

My Dashboards in My Organization

My Data Dash Dev OpenSrcDash

Extra Dashboard Widgets

Data Management, HLT

Knowledge and Maps

Processing Logics / IOT App

Entity Directory and Devices

Resource Manager

Development Tools

DISIT Cluster MLOps ClearML

Jupyter Hub - Python

Jupyter3-[3090] Hub - Python

Jupyter4-[3090] Hub - Python

Jupyter5-[4090] Hub - Python

Access to: Web Scraping Tool

Access to: R Studio Development

Access to: ETL Development

Knowledge Base Graphs

Knowledge Base Queries

Smart City API Docs: Swagger

Internal API Docs: Swagger

Testing API by Postman

Custom Code Access

Snap4City

Home / How and Why To Use it / Tools / Tutorials and Videos

Home / Tutorials and Videos / Welcome to Snap4City

Welcome to Snap4City

You can't delete this newsletter because it has not been sent to all its subscribers.

SNAP4CITY
Control and Plan an
Horizontal AI Platform
Digital Twin for all

SNAP4CITY
Mobility and Transport
Operation and Plan
Digital Twin

SNAP4CITY
Smart Energy and
Smart Buildings
Operation and Plan
Digital Twin

SNAP4CITY
Environment and
Waste Management
Digital Twin

SNAP4CITY
City Users' Services,
Tourism Management
and Safety, Digital Twin

LATEST NEWS

SELECT
1st Place award to
SNAP4CITY

Snap4City
Training on Tools
and Platform

Digital Twin

Scenarios

Organizations

**THE EVENT
FOR BETTER
CITIES**

Innovations

Interoperability

Installations

What People say

Mobile Apps

IOT Devices

IOT Applications

Data Analytics

Dashboards

Living Lab

Smart City API

**Smart City
Ontology**

Work with Us

Articles

**SNAP4CITY on
EUROPEAN OPEN
SCIENCE CLOUD
MARKETPLACE**

SNAP4CITY HACKATHON
ONLY 2019 AND FOR A CONNECTED CITY

INDUSTRY 4.0

Snap4Industry

Snap4Home

Training: [Training Course Snap4City](#)

TECHNICAL OVERVIEW: <https://www.snap4city.org/download/video/Snap4City-PlatformOverview.pdf>

Development Life Cycle: <https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf>

Client-Side Business Logic Widget Manual: <https://www.snap4city.org/download/video/ClientSideBusinessLogic-WidgetManual.pdf>

MicroX troubleshooting ENG: <https://www.snap4city.org/download/video/TroubleShooting-MicroX-on-premise-en.pdf>

MicroX troubleshooting ITA: <https://www.snap4city.org/download/video/TroubleShooting-MicroX-on-premise.pdf>

Booklet Data Analytics, Snap4Solutions: https://www.snap4city.org/download/video/DPL_SNAP4SOLU.pdf

Sintesi in Italiano di Snap4City, Novembre 2022: <https://youtu.be/0bf1eMXHE>

We suggest you to start from [TRAINING 2023 Training Course Snap4City - 2023 Edition](#)

Some news:

eShare in a Snap - The innovative car sharing and car pooling service

Digital Twin Cityverse FAQ to Snap4City

AMMIRARE: make the beach system more resilient to climate change risks through the implementation of natural based solutions

TOURISMO: TOURISM Innovative and Sustainable Management of fIOWs

CAI4DSA: Collaborative explainable neuro-symbolic AI for Decision Support Assistant

SASUAM: Solutions for Safe, Sustainable and Accessible Urban Mobility

Username: clearmldemo

Search

Search

-Any-

**Training on Tools
and Platform**

Powered by
www.km4city.org

Node-RED

MOST
CENTRO NAZIONALE PER LA MOBILITÀ SOSTENIBILE

**Organization
Groups**

DISIT

Developer

Operativo

**Updates on
Tools**

News from Snap4City &
slides, Where to Meet
Snap4City experts [new](#)
roottooladmin1

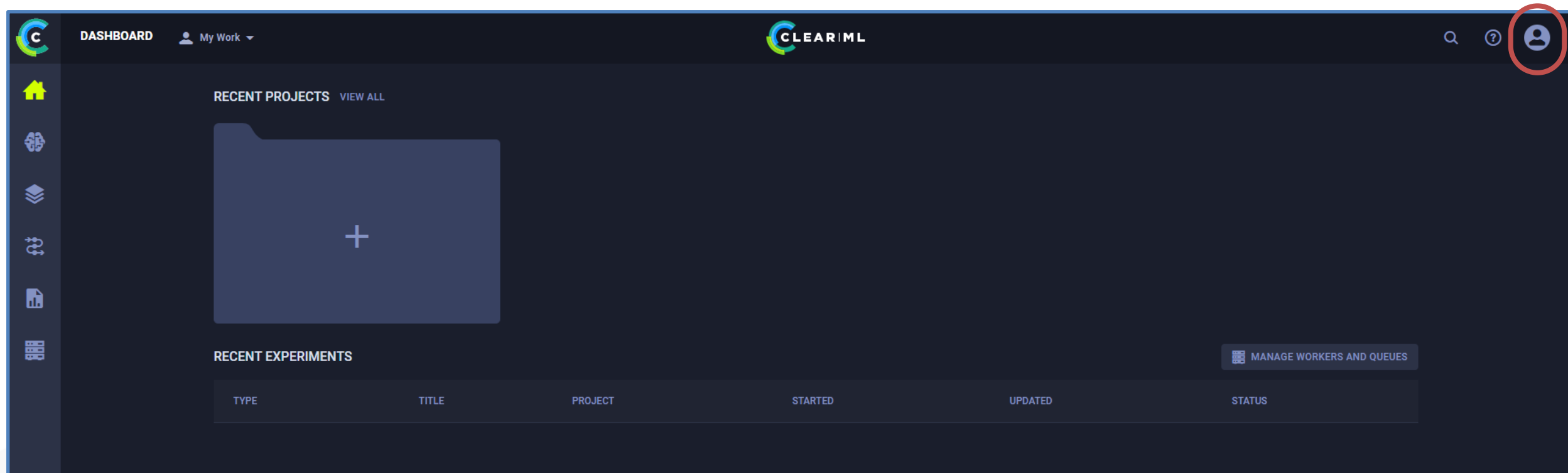
Snap4City at the Smart
City Expo World Congress
2024 [new](#)
roottooladmin1

Development Tools

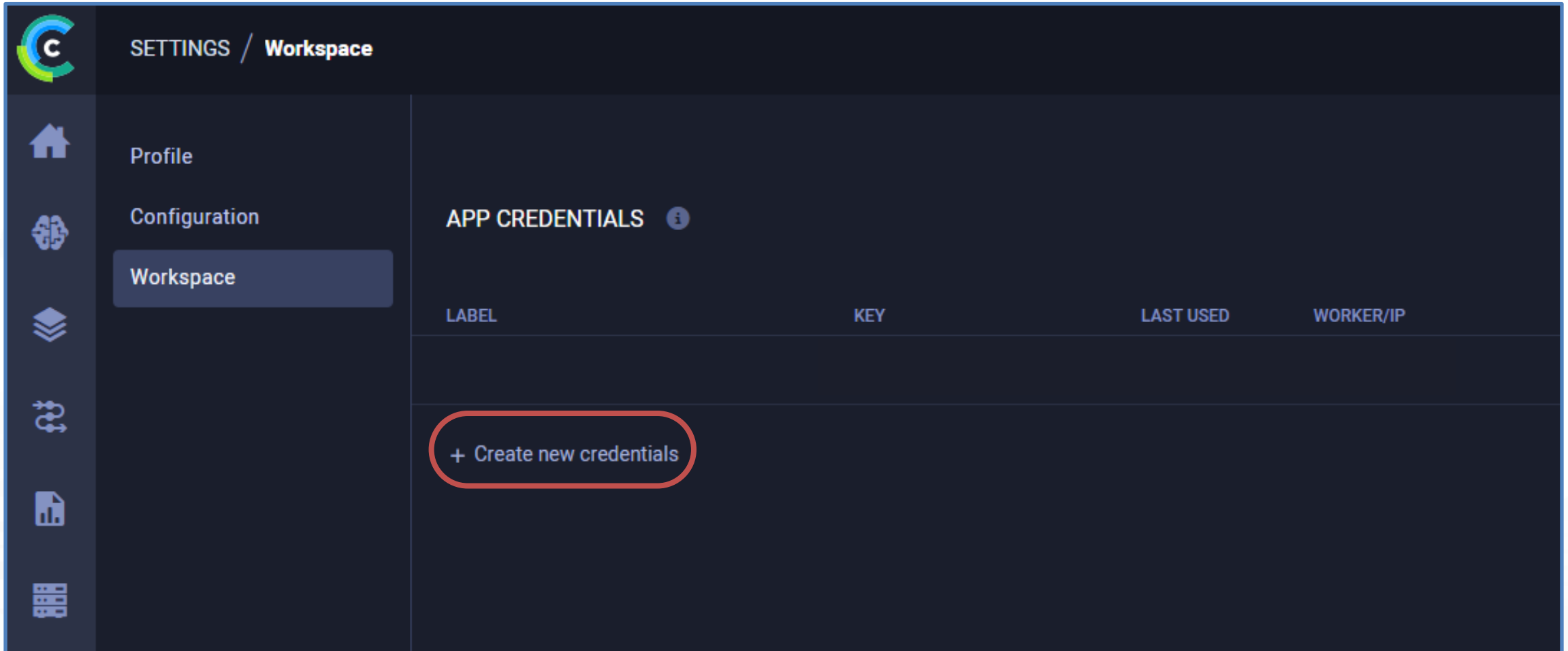
DISIT Cluster MLOps ClearML

Jupyter Hub - Python

ClearML Server account



Getting API keys



The screenshot shows the SNAP4CITY settings interface. The left sidebar contains a navigation menu with icons for Home, Configuration, Workspace, and other settings. The main content area is titled 'SETTINGS / Workspace'. Under the 'APP CREDENTIALS' section, there is a table with columns: LABEL, KEY, LAST USED, and WORKER/IP. A red rounded rectangle highlights the '+ Create new credentials' button located below the table.

LABEL	KEY	LAST USED	WORKER/IP
+ Create new credentials			

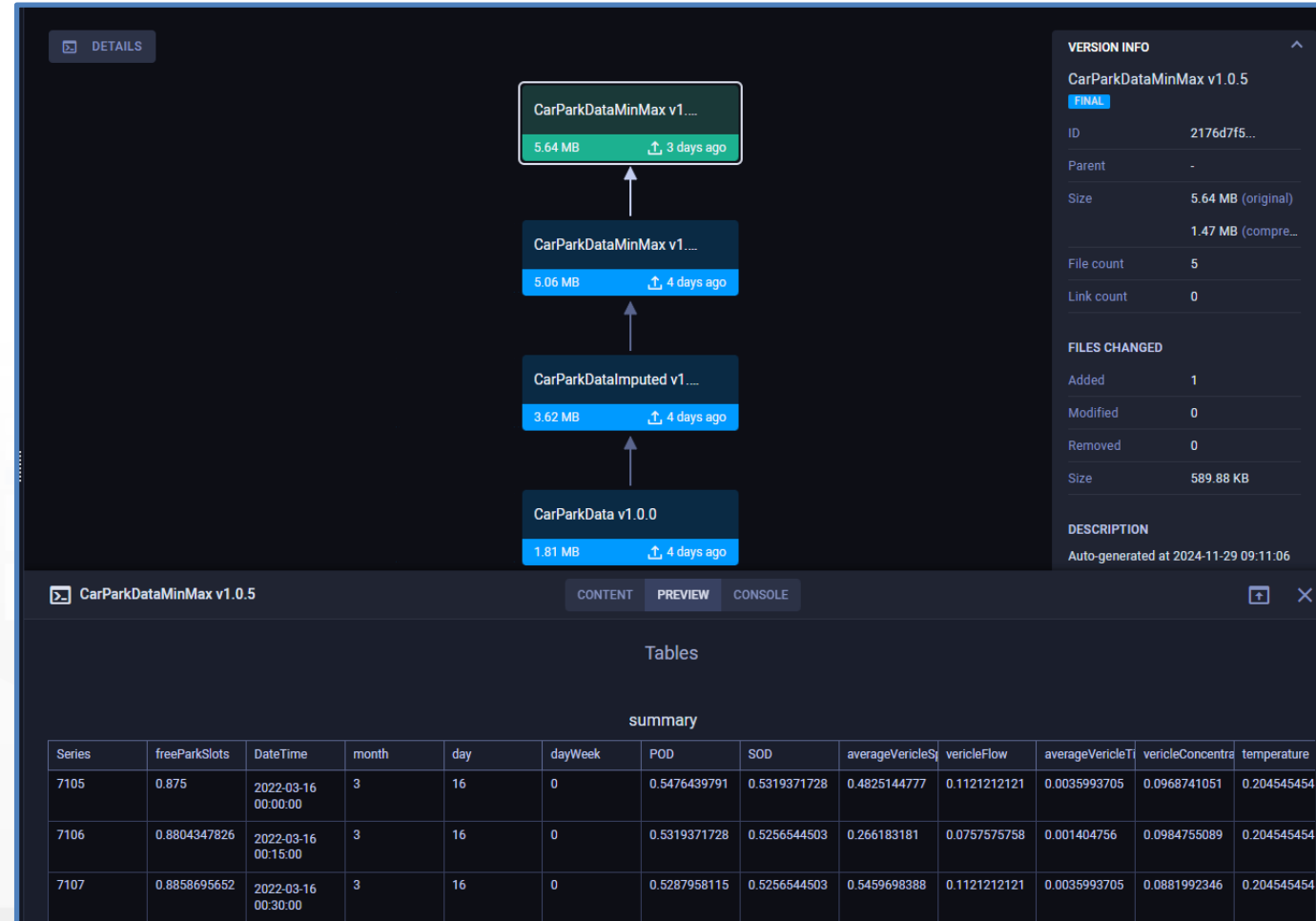
Data uploading

```
dataset = Dataset.create(dataset_project =  
<YOUR_PROJECT>, dataset_name =  
<DATASET_NAME>)
```

```
dataset.add_files(path = <DATA_PATH>)
```

```
dataset.upload()
```

```
dataset.finalize()
```



The screenshot shows the Google Cloud Datastore console interface. At the top, a 'DETAILS' tab is selected. The main area displays a vertical hierarchy of dataset versions for 'CarParkDataMinMax v1.0.5'. The versions are: 'CarParkData v1.0.0' (1.81 MB, 4 days ago), 'CarParkDataImputed v1.0.0' (3.62 MB, 4 days ago), 'CarParkDataMinMax v1.0.0' (5.06 MB, 4 days ago), and 'CarParkDataMinMax v1.0.5' (5.64 MB, 3 days ago). The 'CarParkDataMinMax v1.0.5' version is highlighted with a green border. To the right, a 'VERSION INFO' panel shows details for 'CarParkDataMinMax v1.0.5', including ID, Parent, Size (5.64 MB original, 1.47 MB compressed), File count (5), Link count (0), and a 'FILES CHANGED' section. Below the version hierarchy, a 'CONTENT' tab is selected, showing a table of data. The table has a 'summary' header and columns for Series, freeParkSlots, DateTime, month, day, dayWeek, POD, SOD, averageVericleSt, vericleFlow, averageVericleTi, vericleConcentra, and temperature. The table contains three rows of data for Series 7105, 7106, and 7107.

Series	freeParkSlots	DateTime	month	day	dayWeek	POD	SOD	averageVericleSt	vericleFlow	averageVericleTi	vericleConcentra	temperature
7105	0.875	2022-03-16 00:00:00	3	16	0	0.5476439791	0.5319371728	0.4825144777	0.1121212121	0.0035993705	0.0968741051	0.204545454
7106	0.8804347826	2022-03-16 00:15:00	3	16	0	0.5319371728	0.5256544503	0.266183181	0.0757575758	0.001404756	0.0984755089	0.204545454
7107	0.8858695652	2022-03-16 00:30:00	3	16	0	0.5287958115	0.5256544503	0.5459698388	0.1121212121	0.0035993705	0.0881992346	0.204545454

Jupyter notebook: setup

```
import torch
import os
import pandas as pd
import numpy as np
import torch.nn as nn
import torch.optim as optim
import json
import matplotlib.pyplot as plt
from clearml import Task, Logger, Dataset, OutputModel, StorageManager
from enum import Enum
from torch.utils.data import DataLoader, TensorDataset
from typing import Tuple

os.environ['CLEARML_WEB_HOST'] = "http://"
os.environ['CLEARML_API_HOST'] = "http://"
os.environ['CLEARML_FILES_HOST'] = "http://"
os.environ['CLEARML_API_ACCESS_KEY'] =
os.environ['CLEARML_API_SECRET_KEY'] =

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

Task.add_requirements("numpy", "1.23.5")
Task.add_requirements("pandas", "1.3.3")
Task.add_requirements("accelerate", "0.31.0")
Task.add_requirements("bitsandbytes", "0.43.1")

task = Task.init(project_name = "Snap4ParkingCML", task_name="Training", output_uri = True)
output_model = OutputModel(task = task, framework = "PyTorch")
#output_model.set_upload_destination("models/ParkingModel/")
task.set_base_docker("")
task.execute_remotely(queue_name=
)
#StorageManager.upload_folder("./models", '/models/')
hyperparams = dict()
criterion = nn.MSELoss()
hyperparams['num_epochs'] = 100
task.connect(hyperparams)
```

Jupyter notebook: retrieving data


```
dataset = Dataset.get(dataset_id=  
dataset_folder = dataset.get_local_copy()  
  
train_data = pd.read_csv(dataset_folder+"/train_data.csv")  
val_data = pd.read_csv(dataset_folder+"/validation_data.csv")  
test_data = pd.read_csv(dataset_folder+"/test_data.csv")  
print(train_data.head())  
print(test_data.head())
```


Jupyter notebook: uploading artifacts and saving model

```
torch.save(model, "ParkingModel.pth")
task.upload_artifact(name = "train_losses", artifact_object = train_losses)
task.upload_artifact(name = "val_losses", artifact_object = val_losses)
plt.figure(figsize=(10, 6))
plt.title("Training vs Validation Loss")
plt.xlabel("epoch")
plt.ylabel("loss")
plt.plot(range(len(train_losses)), train_losses, label="training loss", color="red")
plt.plot(range(len(val_losses)), val_losses, label="validation loss", color="blue")
plt.legend()
plt.show()

task.mark_completed()
task.close()
task.publish()
```

ClearML: monitoring execution


Training

+ ADD TAG

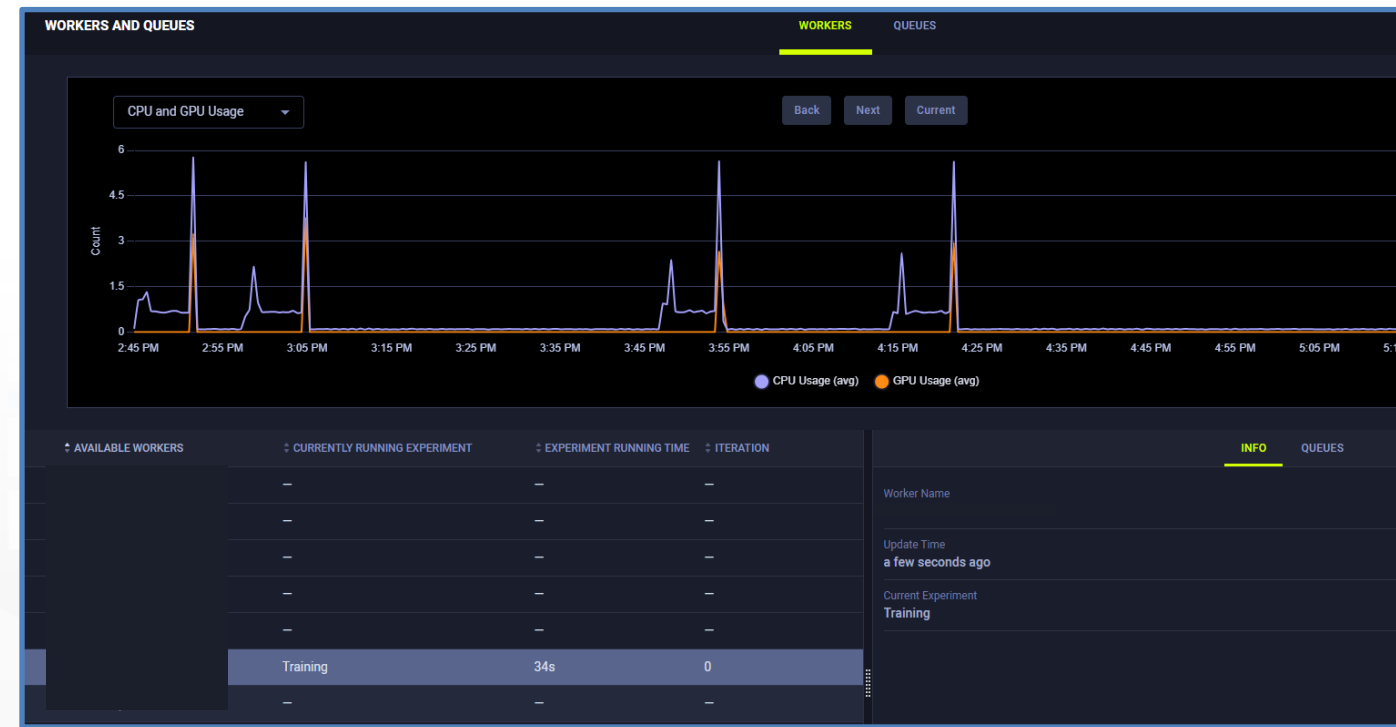
EXECUTION
CONFIGURATION
ARTIFACTS
INFO
CONSOLE

Hostname: 42-4090

```

Series ... Series.1
0 10081 ... 10081
1 10082 ... 10082
2 10083 ... 10083
3 10084 ... 10084
4 10085 ... 10085
[5 rows x 15 columns]
Measures of Training:
-/X_train length:5088
-/y_train length:5088
-/X_train instance shape:(672, 15)
Measures of Validation:
-/X_val length:2304
-/y_val length:2304
-/X_val instance shape:(672, 15)
Measures of Test:
-/X_test length:960
-/y_test length:960
-/X_test instance shape:(672, 15)

```





ClearML: task ending savings 1

Training

ADD TAG

PUBLISHED

ID e66ce5bb...

EXECUTION

CONFIGURATION

ARTIFACTS

INFO

CONSOLE

SCALARS

PLOTS

DEBUG SAMPLES

COMPLETED AT:	Dec 2 2024 18:02
RUN TIME:	01:37m
QUEUE:	queue
WORKER:	
CREATED BY:	ClearUser14
PARENT TASK:	N/A
PROJECT:	Snap4ParkingCML
ID:	
CLEARML VERSION	clearml-1.16.5
CLI	/root/.clearml/venvs-builds/3.10/code/Snap4ParkingCML.py
OS	Linux-5.15.0-102-generic-x86_64-with-glibc2.35
cpu_cores	32
datasets	
gpu_count	1
gpu_driver_cuda_version	12.4
gpu_driver_version	550.76
gpu_memory	24GB
gpu_type	
hostname	
ide	Jupyter
memory_gb	62.5

ClearML: task ending savings 2

