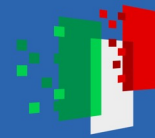




Finanziato
dall'Unione europea
NextGenerationEU



Ministero
dell'Università
e della Ricerca



Italiadomani
PIANO NAZIONALE
DI RIPRESA E RESILIENZA



terabit

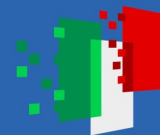
AI_INFNO report

Luca Giommi – INFNO CNAF



AI_INFNO

Artificial Intelligence technologies
for INFNO research



Scope and objectives

The provisioning of a **common, stable, and reliable** ground for researchers involved in ML to develop, review and share their applications, **crossing the borders between different communities**, INFN units, experiments and research domains

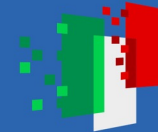
Provide a **centrally maintained cloud-based infrastructure** for interactive and batch ML fast prototyping, with access to modern hardware accelerators (GPU, FPGA...) and systems tuned for ML performance



Finanziato
dall'Unione europea
NextGenerationEU



Ministero
dell'Università
e della Ricerca



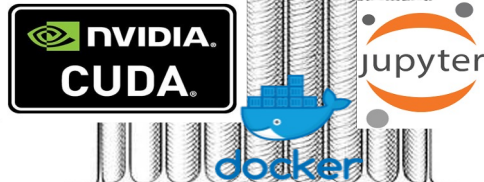
Italiadomani
PIANO NAZIONALE
DI RIPRESA E RESILIENZA



From ML_INFEN to AI_INFEN: the structure of the project

Applications of Machine Learning
HEP, MedPhys, GW detection, Theory...

Infrastructure



WP 1

Stewardship



WP 2

Knowledge Base



WP 3



Virtualization and orchestration layer
developed and maintained by INFN Cloud



INFN Cloud Resources: Infrastructure

ML_INFN has been among the first and most enthusiastic users of INFN Cloud.

Computing resources available to AI_INFN are located in Room Tier-1 of CNAF and managed through a virtualization layer (**OpenStack of Cloud@CNAF**) in INFN Cloud.

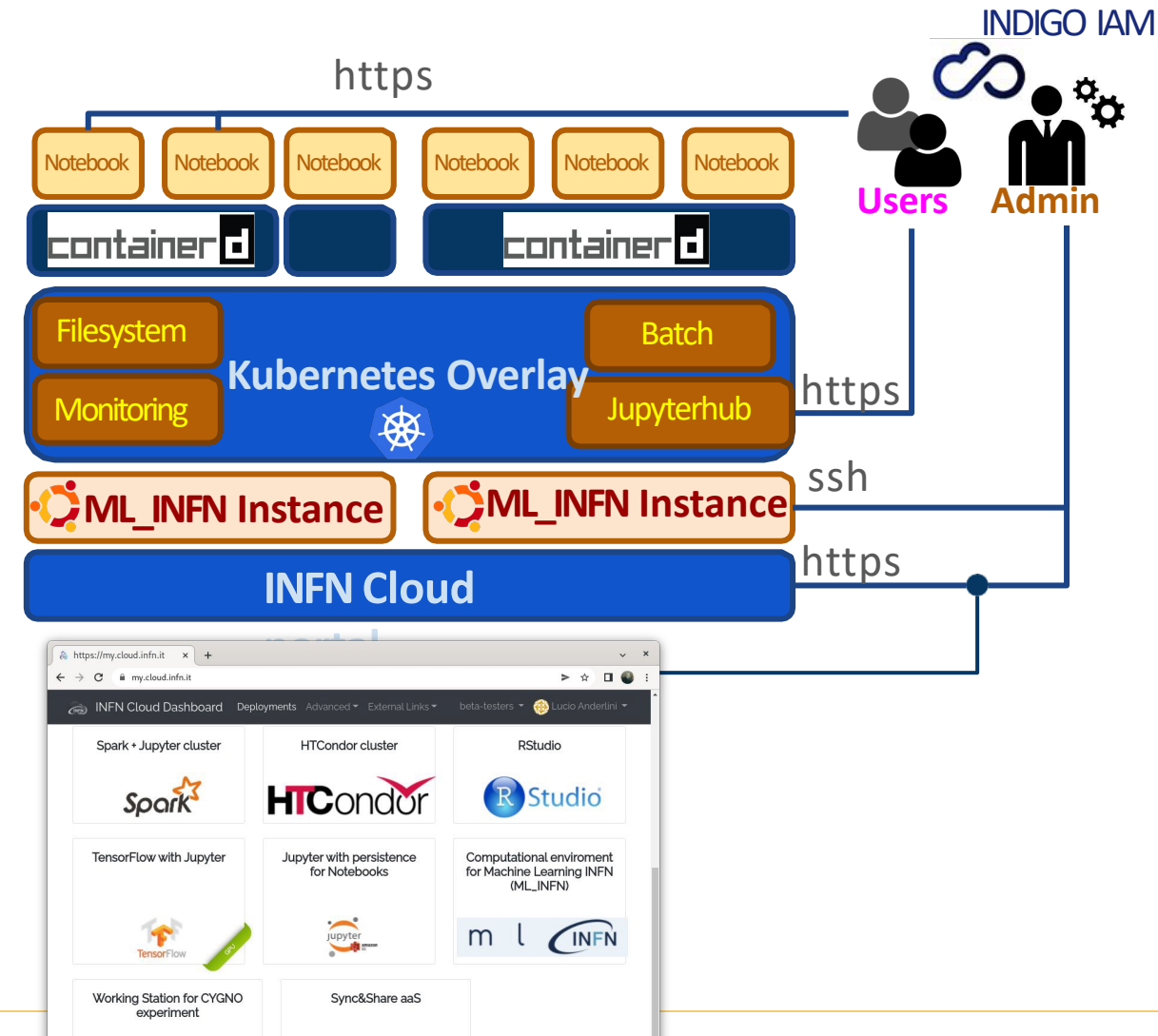
- **Server 1:** 8 nVidia *Tesla T4* (CSN5) + 5 nVidia *RTX 5000* (ML_CLOUD, Firenze)
- **Server 2:** 1 *A100* (CSN5) + 1 A30 (Dip. di Fisica, UniFi)
- **Server 3:** 3 *A100* (CNAF)

Partitioning A100 GPUs with MIG (*Multi Instance GPU*) technology, we manage to serve up to 42 GPU for interactive development.



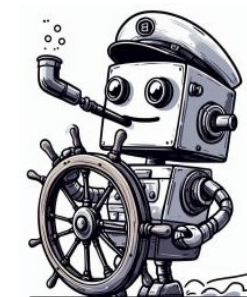
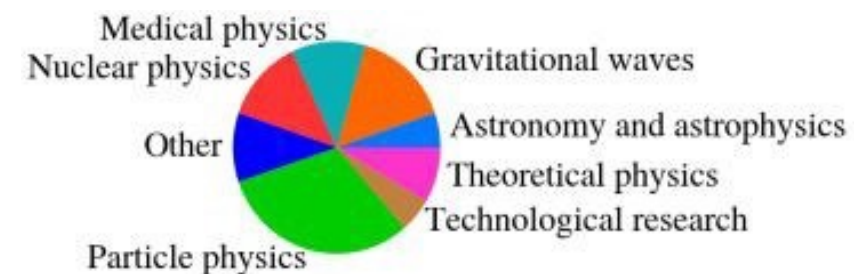
INFN Cloud Resources: Architecture

- The ML_INFN outcome: **“sharing precious GPUs through the Cloud is feasible and effective!”**
- With AI_INFN, we improved on sharing by **decoupling data from computing resources**, with a **filesystem shared across the VMs**
- An additional abstract, elastic overlay is added on top of multiple VMs **Kubernetes Overlay**:
 - login via AAI → **INDIGO IAM**
 - Monitoring & Accounting
 - Managed software environments for ML
- Adding and removing VMs enables manual horizontal scaling



A stress test: the ML_INFN Hackathon

- ML_INFN organized training events (“*hackathons*”), targeting **entry level** (june 2021, december 2021, june 2023) and **advanced** (Bari in november 2022, Pisa in november 2023) audience.
- In the latest event, the AI_INFN’s new platform was stress-tested:
 - at **Cloud@CNAF** (using 2 ×A100 GPUs for up to 14 participants)
 - at **ReCaS-Bari** (using 4 ×A100 GPUs for up to 28 participants)
- Independent networks and file-systems
- Shared IAM authentication
- Synchronized software environments
- Intensive use of the GPUs





Managed software environments: docker

JupyterHub

hub.ai.cloud.infn.it/hub/spawn

Home Token Admin

Server Options

Welcome to the AI_INFN Platform!

You are logged as: anderlini

You are member of the following projects: lamarr, pinne, sosc, hackathon, supermem, pcc

The platform is currently under active development, and not compliant with GDPR rules, yet. **Do not upload personal data. Uploaded data might be lost.**

You might want to read the [Frequently Asked Questions](#) and contribute to our [Knowledge Base](#).

Docker image:

Number of cores:
 1 2 3 4 8 cores

Memory size:
 2 4 8 31 32 GB

Hardware accelerator:
 None
 nVidia A100 1g MIG partition (10 GB) 10/14 available
 nVidia RTX5000 (32 GB) 1/4 available
 nVidia A100 (80 GB) 0/1 available

Start

The customizable docker image defines the user interface.
Default: VS Code, Dask, MinIO (soon Rados)

JupyterLab (auto-B)

hub.ai.cloud.infn.it/user/anderlini/lab/w...

File Edit View Run Kernel Tabs Settings Help

Filter files by name

Name	Last Modified
/	
jfs	1 minute ago
minio	1 minute ago
private	yesterday
shared	1 minute ago
system	1 minute ago

Notebook

Python 3 (ipykernel) keras3 VS Code IDE [↗]

Console

Python 3 (ipykernel) keras3

Simple 0 0 Mem: 205.40 / 2048.00 MB Launcher 1

Managed software environments: conda

Configuring the Python software stack to properly control the GPU is sometimes challenging and requires time and expertise.

Sometimes, projects require multiple environments in the same JupyterLab session: picking the right docker image is not a viable option.

A cross-platform and language agnostic package and environment manager, which solves portability between collaborators and is adopted particularly when python external tools are used.

Conda utilization on JupyterLab:

- Allows to manage dependencies of Python projects efficiently.
- Provides isolated environments to execute Python code and Jupyter notebooks, independent of the underlying docker image.
- Users are encouraged to clone and customize the managed conda environments to add their project's dependencies.

The logo for Conda, featuring a green snake head icon on the left, followed by the word "CONDA" in a bold, green, sans-serif font.



Managed software

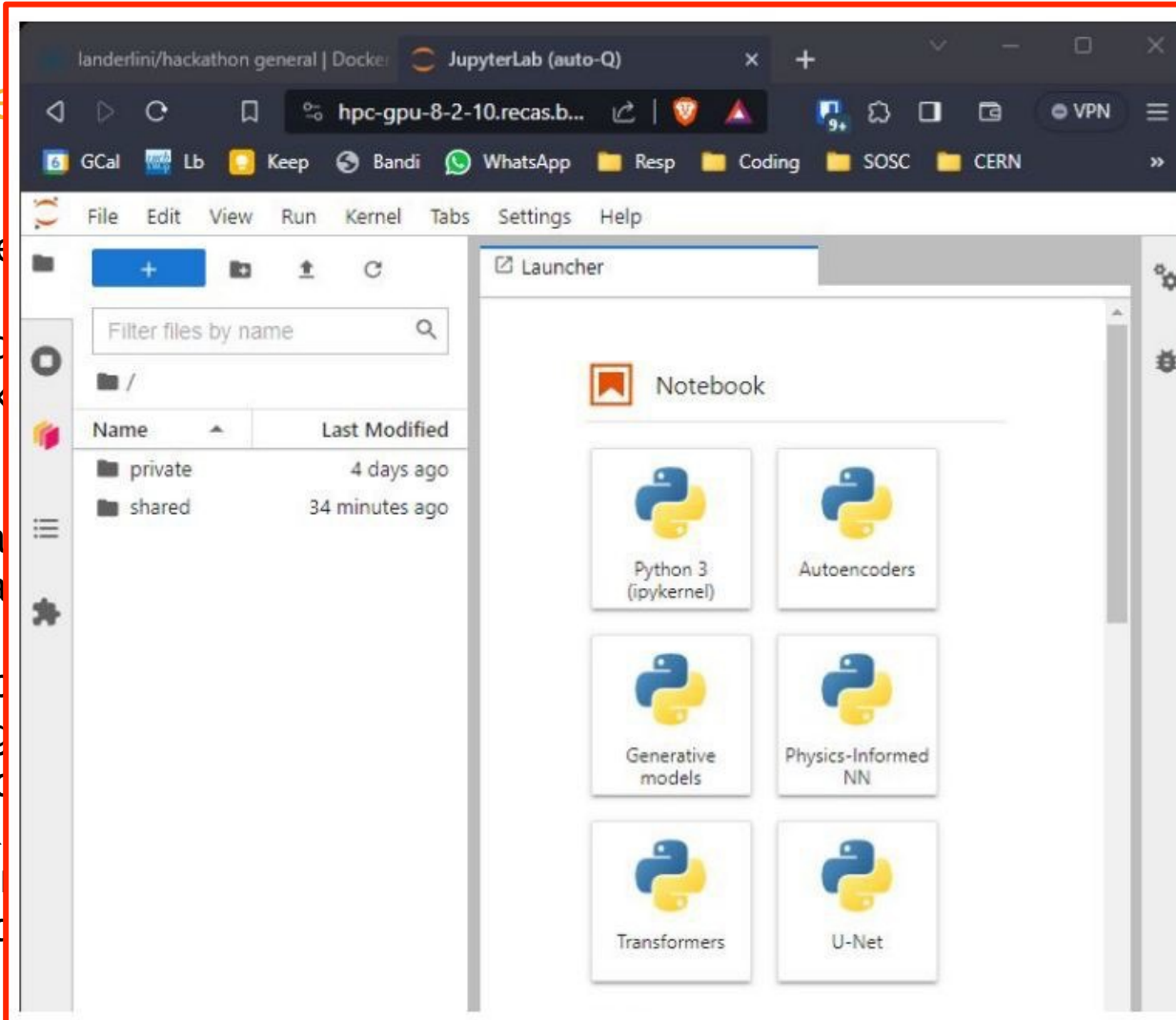
Configuring the Python GPU is sometimes challenging

Sometimes, projects require a JupyterLab session: pick the right option.

A cross-platform and language-independent interface between collaborators and the underlying infrastructure.

Conda utilization on JupyterLab

- Allows to manage dependencies
- Provides isolated environments independent of the underlying docker
- Users are encouraged to create their own project's dependencies



CONDA

solves portability issues.

works, independent of the

environments to add their

Managed software: apptainer

Main problem with conda: it generates environments with 10000+ files, bad for any file system.

A nightmare when distributed.

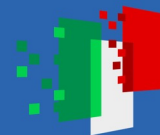


APPTAINER

- Apptainer is a containerization platform offering an isolated, reproducible environment for application execution.
- Allows to pack an application and all its dependencies in a container, granting portability and consistency of the execution environment.

Advantages of Conda + Apptainer:

- **Conda** is what developers expect, **Apptainer** (squashfs) delivers envs as a single file.
- **Reproducibility:** By using Conda for development and Apptainer for execution, it's possible to ensure complete reproducibility of the environment both during development and distribution.



Monitoring & Accounting with GPU

Three levels of monitoring & accounting:

- **Resource provisioning accounting:** report on resource usage
- **Resource provisioning monitoring:** check if allocated resources are in use or idle
- **Service accounting:** to have vision of the balance and distribution of the resources among projects and, in case of high load, to enforce/guarantee fair access to resources between users.
 - This is to have control over who is using the AI_INFN platform and to do what. In this way we can estimate how much we could shrink the CPU and RAM resources allocated to a single-accelerator task without an evident penalty in performance

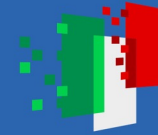




Finanziato dall'Unione europea
NextGenerationEU



Ministero dell'Università e della Ricerca



Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA



NVIDIA GPU OPERATOR
NVIDIA Driver
NVIDIA Container Runtime
NVIDIA Kubernetes Device Plugin
NVIDIA GPU Monitoring

KUBERNETES
CONTAINER ENGINE
LINUX DISTRIBUTION

SOFTWARE
HARDWARE

EGX SYSTEMS
From Jetson Nano to NGC-Ready for Edge Servers



Prometheus



Accounting:
Configuration of a PostgreSQL server through Ansible
(Nadir Marcelli & Stefano Stalio)

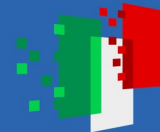
- Allows synchronous replication on one or more secondary servers
- Configuration of an SSL connection to ensure a secure communication channel for replication
- Includes configuration of pgbackrest for periodic backup
- Installation of repmgr for automatic failover management.



Finanziato dall'Unione europea
NextGenerationEU



Ministero dell'Università e della Ricerca



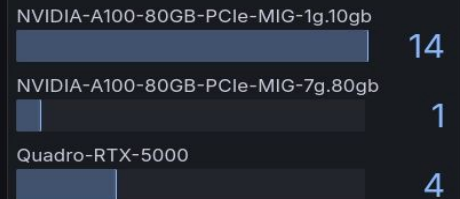
Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA



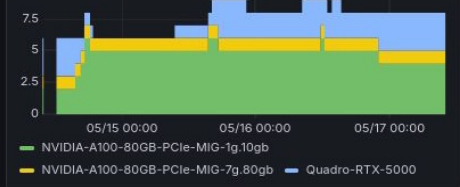
24h 30d Registered

9 19 57

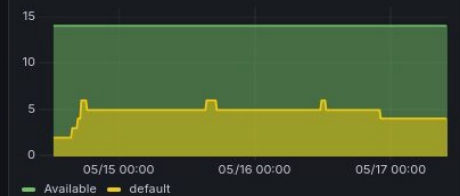
Available



Allocated



A100 1g.10gb



RTX 5000



Local storage



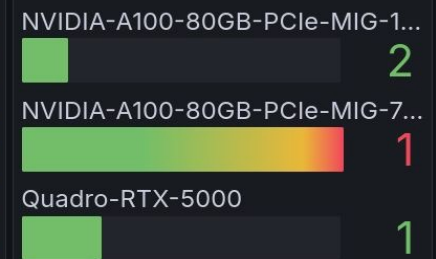
CPU



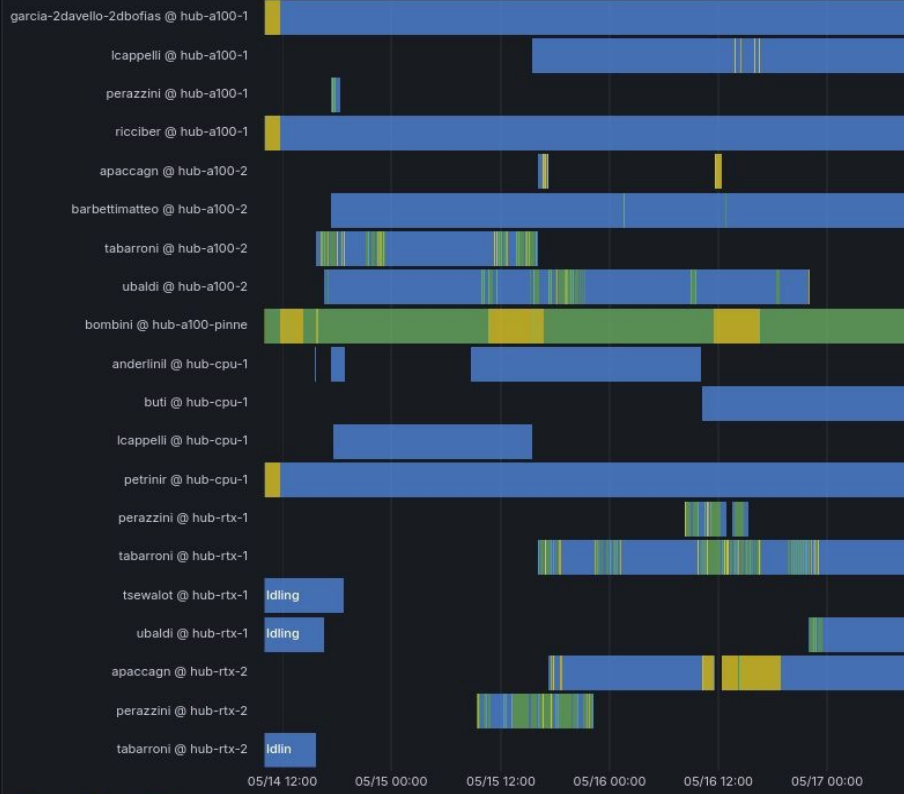
Memory



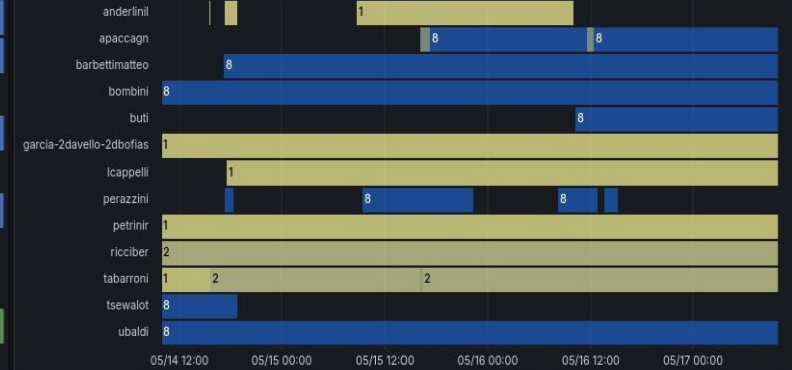
GPUs



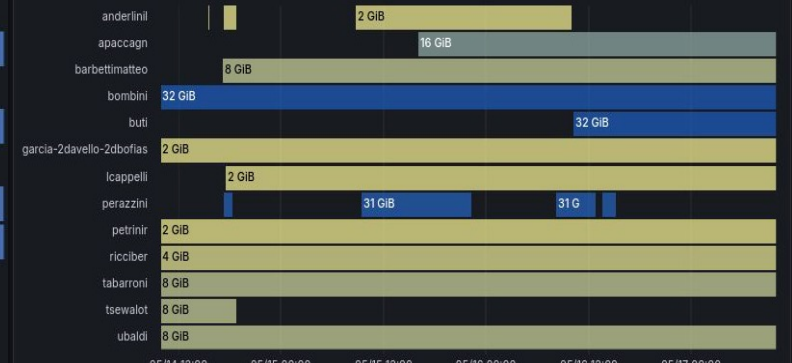
Summary of user's activity



Requested CPU



Requested Memory





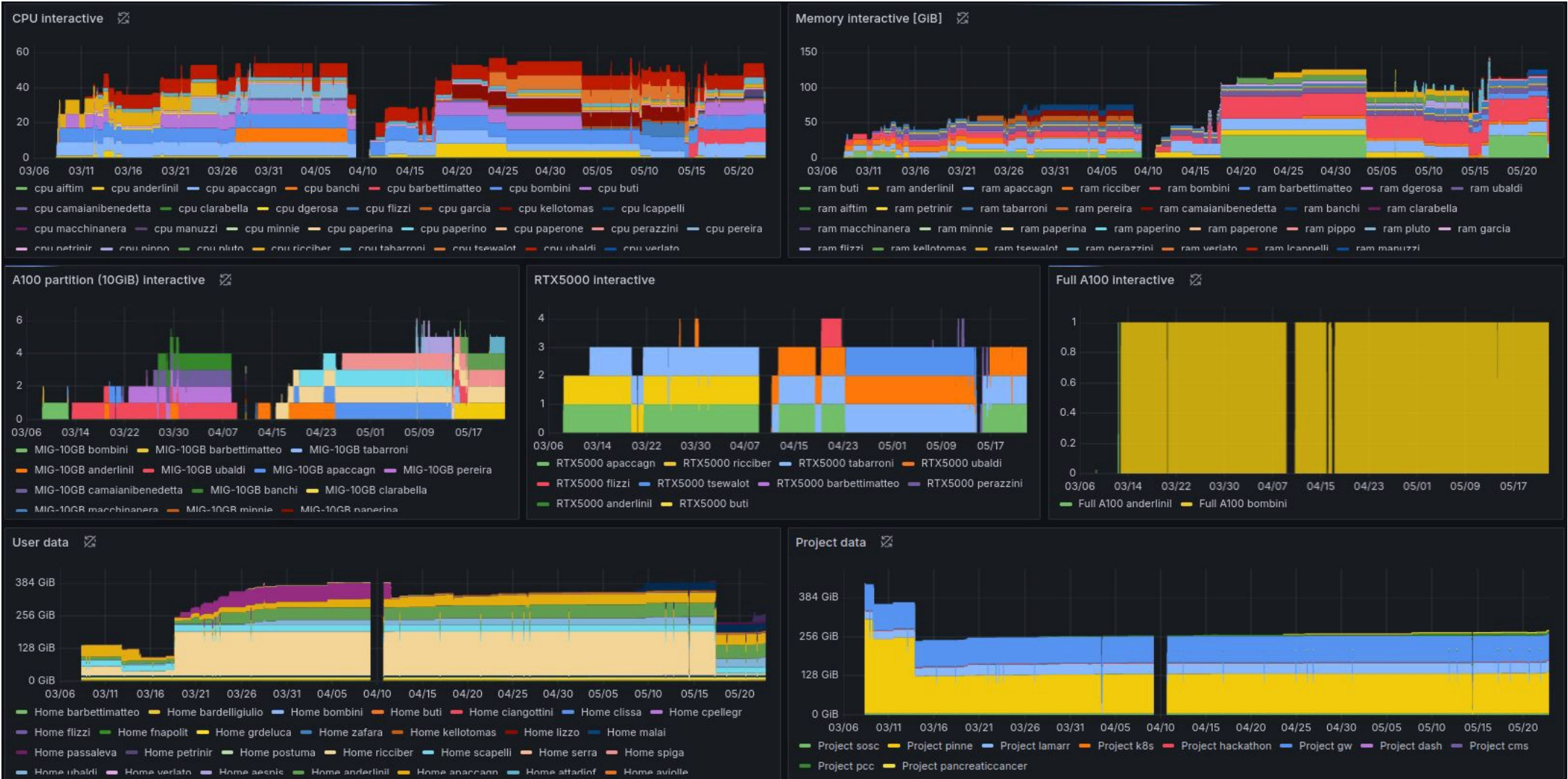
Finanziato dall'Unione europea
NextGenerationEU



Ministero dell'Università e della Ricerca



Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

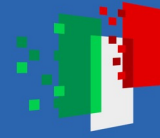




Finanziato
dall'Unione europea
NextGenerationEU



Ministero
dell'Università
e della Ricerca



Italiadomani
PIANO NAZIONALE
DI RIPRESA E RESILIENZA



Ongoing developments



From interactive to batch jobs

- Once an analysis or the development of a model is mature, analysts want to scale it on more resources:
 - longer training time than available interactively;
 - freeing interactive resources for development;
 - parallel execution of multiple trials...



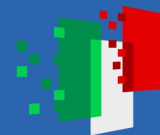
- We are developing a microservice ([vk-dispatcher](#)) translating an interactive session into a [Kubernetes Job](#), executed on the cluster resources.



Development is our priority!
Batch workloads must not
affect the interactive use of the
platform.



Need for a batch
management system,
instantaneously evicting
opportunistic batch jobs.

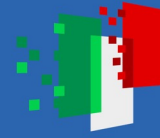


Kueue is a set of APIs and a controller meant to simplify and improve job queue management in Kubernetes.

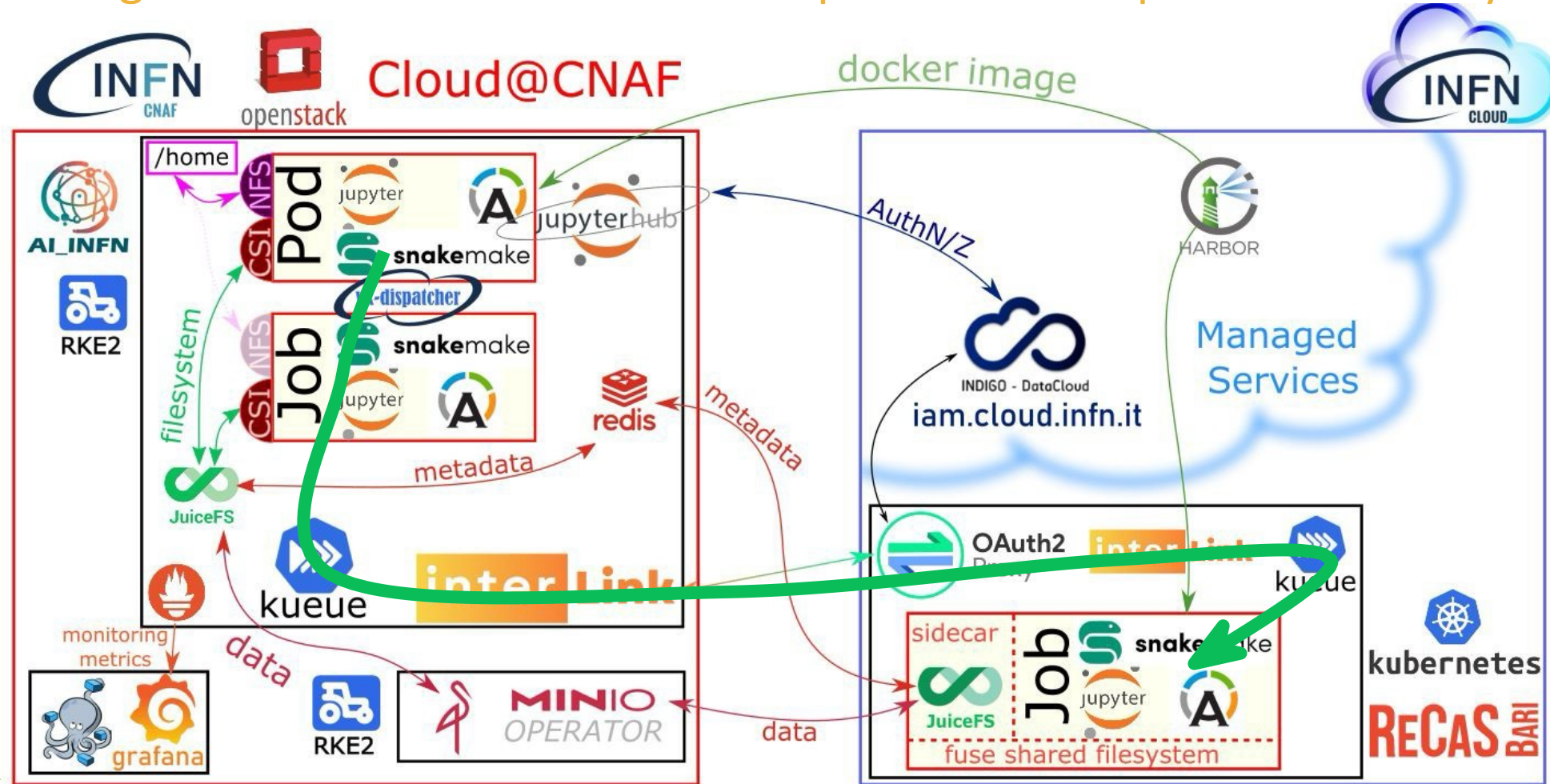
- **Queue management:** Provides a solid infrastructure for job queue management, allowing reliable and scalable execution of jobs inside the Kubernetes cluster.
- **Integration with Kubernetes resources:** Kueue integrates natively with Kubernetes' resources and functionality, making use of orchestration and management features of the cluster.
- **Monitoring and Scalability:** Thanks to dedicated controllers, Kueue simplifies monitoring of job state and allows to scale resources automatically based on workload.


















vk-dispatcher + kueue were alpha-tested with three different applications.
Effective for analysis workflows combining CPU-only and GPU-powered steps.



Offloading the workload with interLink: a proof-of-concept tested on May 16



Conclusion

Feature	Proof of concept	Beta-tested in hub.ai	Available for all users	Ready for DataCloud
Interactive development (GPU)	2023-05-18	2023-12-13	2024-03-08	
Interactive develop. (QC/FPGA)	<i>QC coming soon</i>			
Monitoring	2024-03-18	2024-04-22	2024-05-13	
Accounting	2024-03-18	<i>coming soon</i>		
Batch job submission	2023-12-19	2024-04-18		
Offloading towards Kueue	2024-05-16			
Offloading to Docker (GPU)	<i>coming soon</i>			

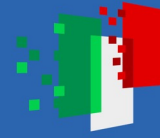
Stay tuned by joining our mailing list: ai-infn-csn5@lists.infn.it



Finanziato
dall'Unione europea
NextGenerationEU



Ministero
dell'Università
e della Ricerca



Italiadomani
PIANO NAZIONALE
DI RIPRESA E RESILIENZA



Grazie per l'attenzione

luca.giommi@cnaif.infn.it