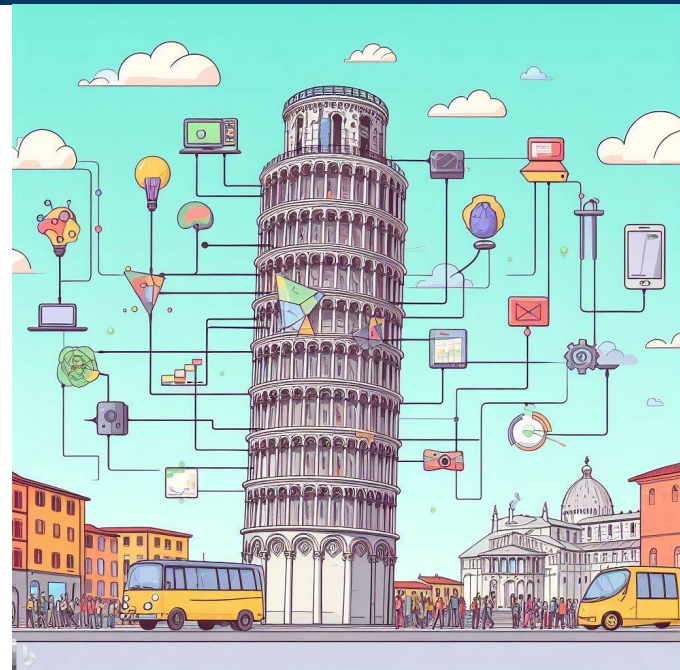


*Fifth Hackathon of Machine Learning
Advanced Level
November 13–16, Pisa*

Advanced Hackathon Infrastructure



Lucio Anderlini
Istituto Nazionale di Fisica Nucleare
SEZIONE DI FIRENZE



Artificial Intelligence technologies
for INFN research

Introduction

To practice with the machine learning exercises we want you to access to GPUs in the Cloud.

We have access to resources at:

- CNAF in Bologna
- ReCaS Bari

Thanks to the **Cloud approach**, the two sites are almost indistinguishable, though the flavor of the GPUs is slightly different.

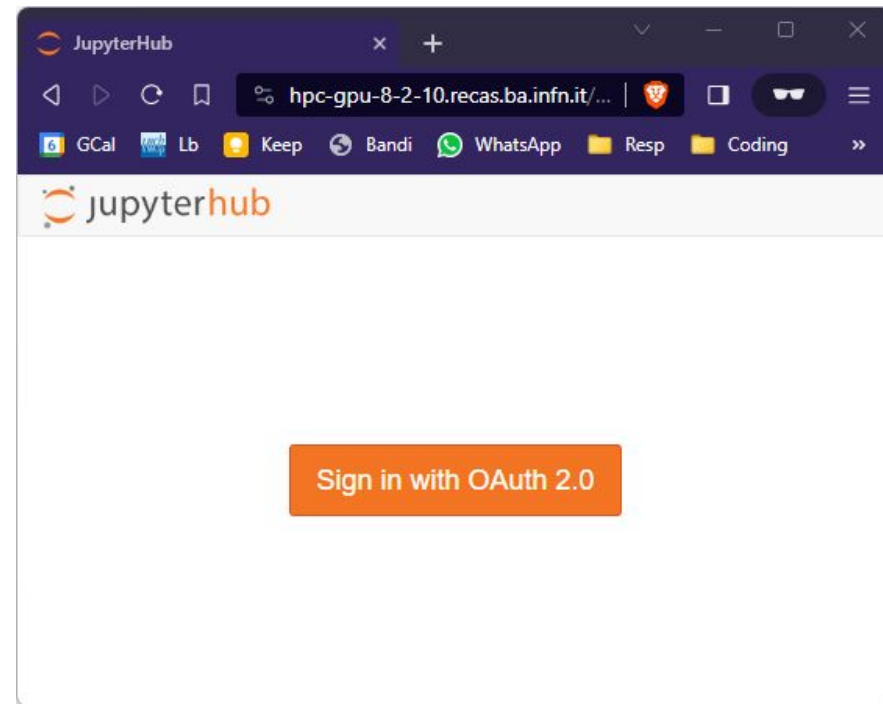
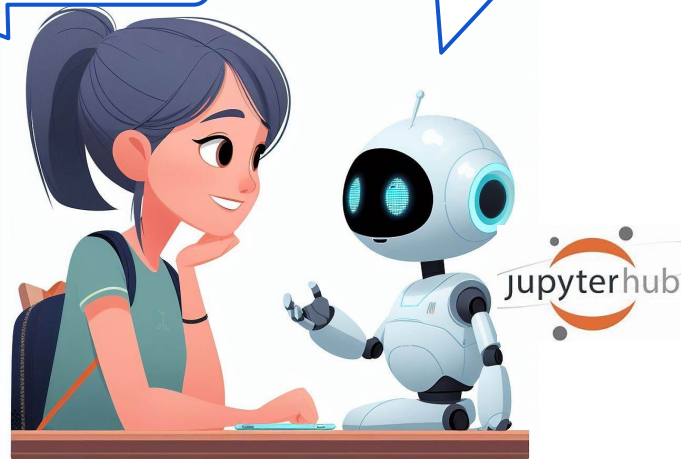




Behind the scenes: *Landing page*

Hey, I am Jupyter Service
running at ReCaS in Bari.
I am not sure we met before...

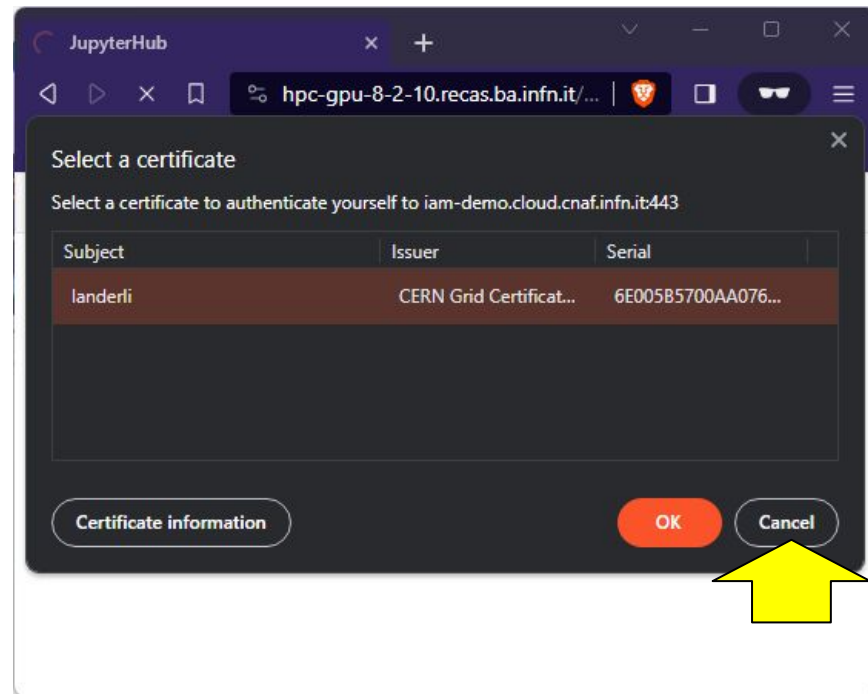
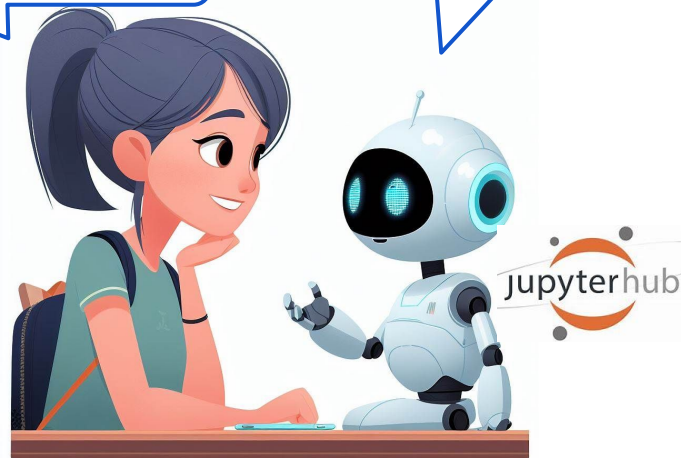
It's me!



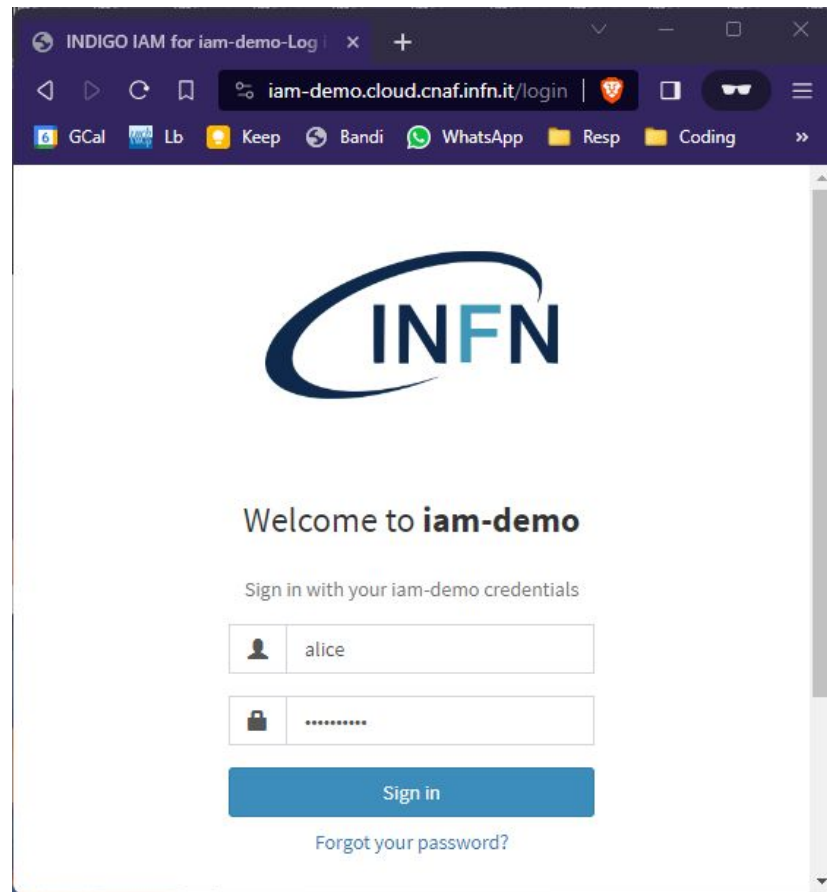
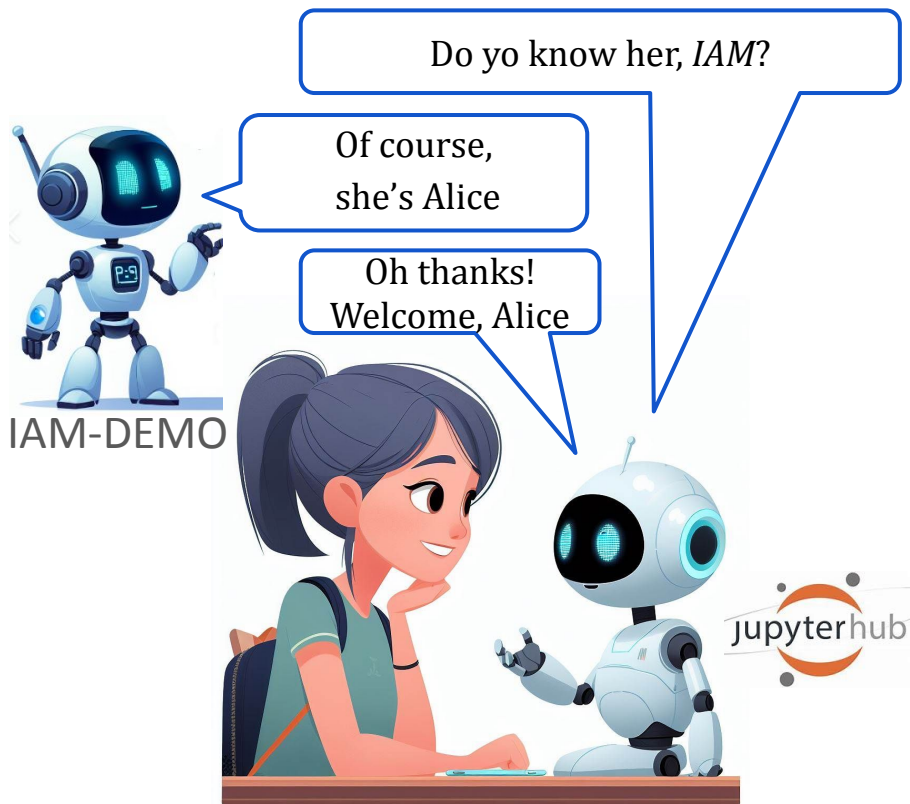
Authentication procedure

Oh.. maybe you are a WLCG user?

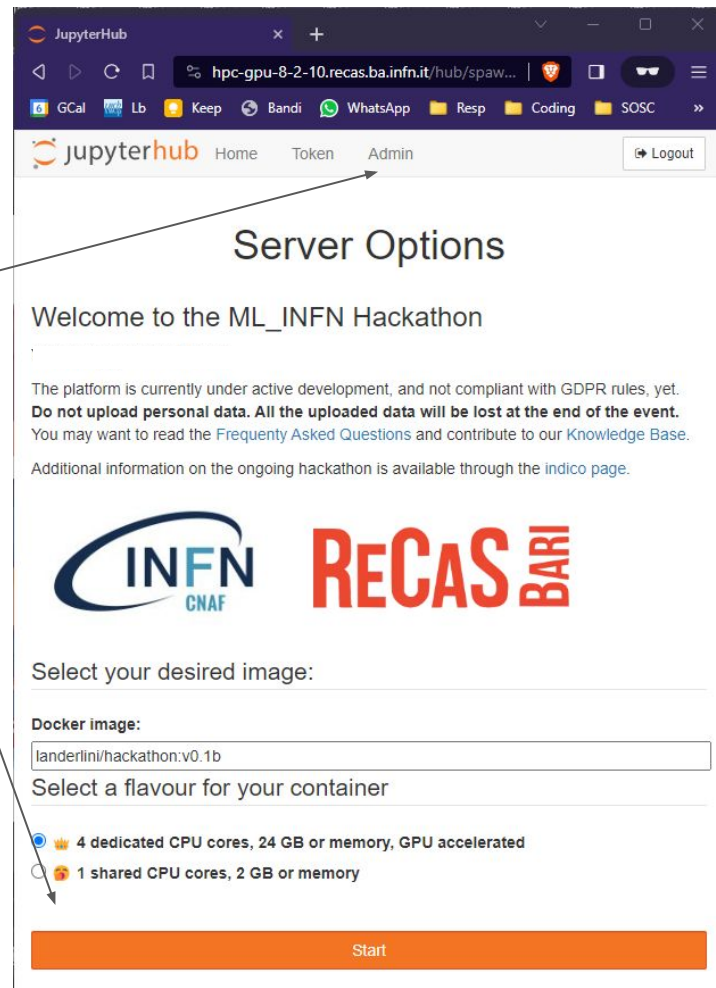
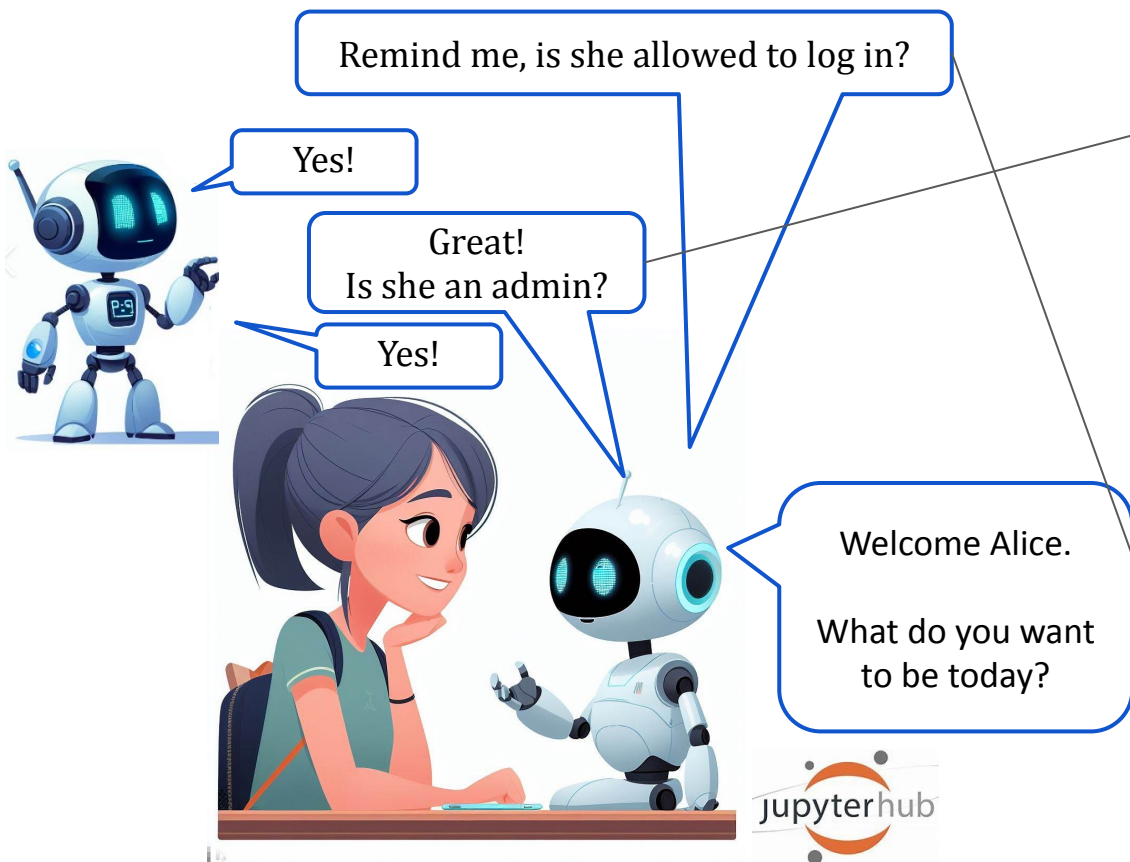
Not here for that



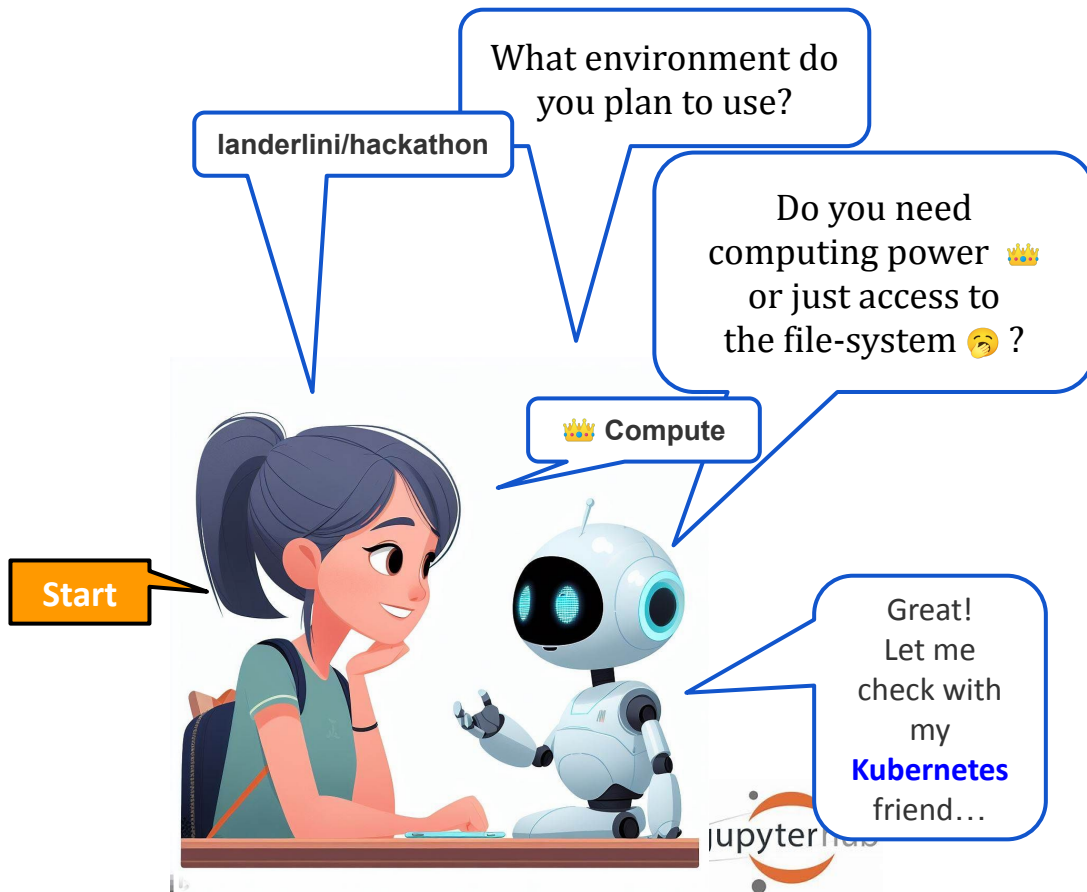
Authentication



Authorization



Authorization



JupyterHub

hpc-gpu-8-2-10.recas.ba.infn.it/hub/spaw...

GCal Lb Keep Bandi WhatsApp Resp Coding SOSC

Home Token Admin Logout

Server Options

Welcome to the ML_INFN Hackathon

The platform is currently under active development, and not compliant with GDPR rules, yet. **Do not upload personal data. All the uploaded data will be lost at the end of the event.** You may want to read the [Frequently Asked Questions](#) and contribute to our [Knowledge Base](#). Additional information on the ongoing hackathon is available through the [indico page](#).

Select your desired image:

Docker image:
landerlini/hackathon:v0.1b

Select a flavour for your container

- 🏰 4 dedicated CPU cores, 24 GB or memory, GPU accelerated
- 🏠 1 shared CPU cores, 2 GB or memory

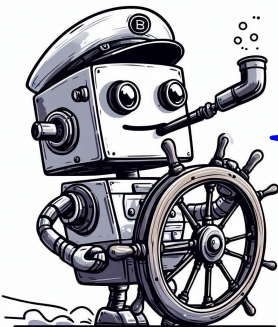
Start

Spawning: Allocating resources

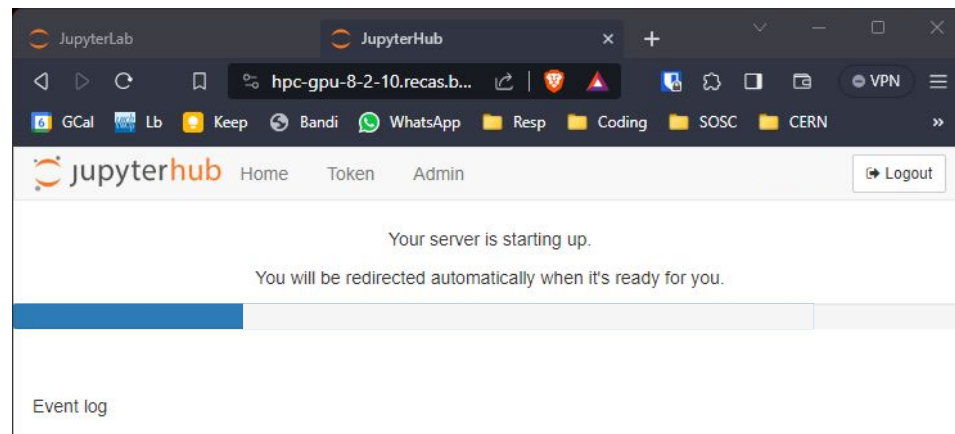
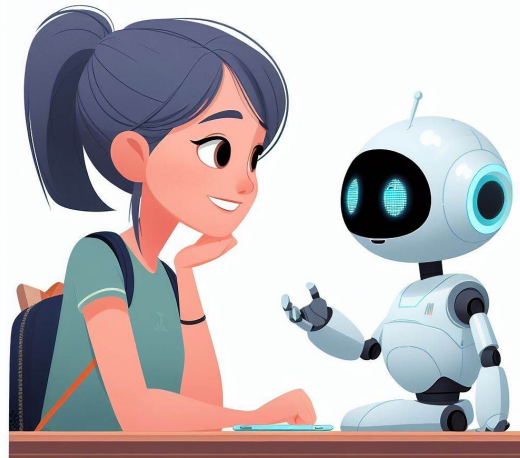


Aye! I have nVidia GPUs, today, and there is an available GPU slot in one of the computing nodes. It's yours!

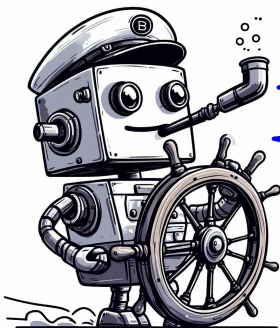
5GB	5GB	5GB	5GB	5GB	5GB	5GB	5GB
	1g.5gb	GPU Instance					
		<ul style="list-style-type: none"> • Fixed partition of memory and compute • Fixed amount of "other" GPU Engines (depending on size) 					
	1 compute	1 compute	1 compute	1 compute	1 compute	1 compute	1 compute



CPU and memory are available as well. I will start a **docker** for you.

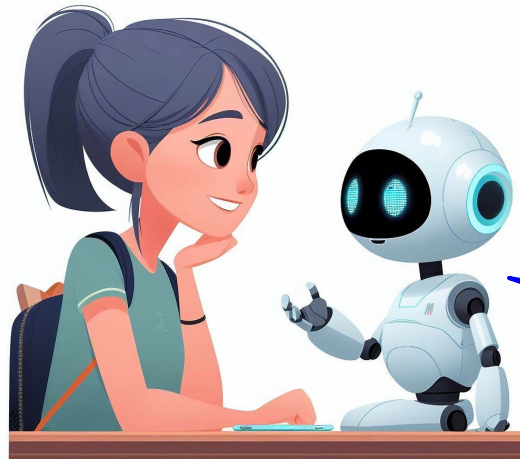


Spawning: *Docker image*



I found the image you requested, **landerlini/hackathon**, ON hub.docker.com I will download it for you, if needed.

CPU and memory are available as well. I will start a **docker** for you.



landerlini/hackathon general | x JupyterLab (auto-Q)

hub.docker.com/reposit...

landerlini / hackathon

Description
This repository does not have a description

Last pushed: 11 days ago

Docker commands
To push a new tag to this repository:
`docker push landerlini/hackathon:tagname`

Tags
This repository contains 4 tag(s).

Tag	OS	Type	Pulled	Pushed
v0.1b		Image	8 days ago	11 days ago
v0.1a		Image	11 days ago	11 days ago
v0.1		Image	11 days ago	11 days ago

Automated Builds
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.
Available with Pro, Team and Business subscriptions. [Read more about automated builds](#)

JupyterLab

GCal Lb Keep

jupyterhub Home Token Admin Logout

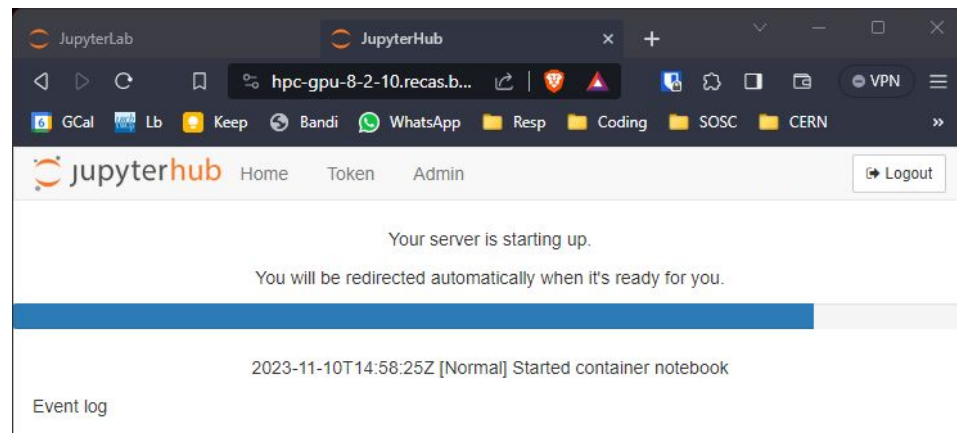
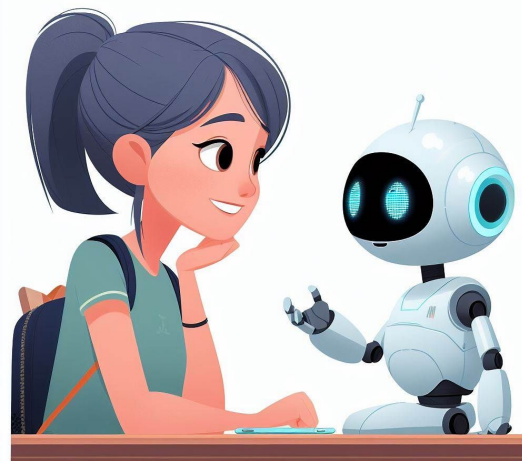
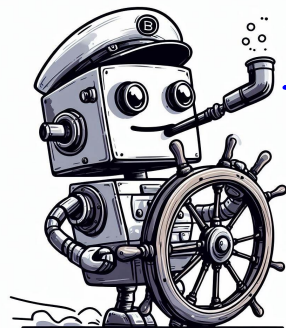
Your server is starting up.
You will be redirected automatically when it's ready for you.

Kubernetes, when you are done, please install the **conda** environments, as well.

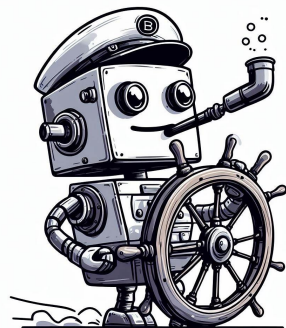


Spawning: *PostStart* hook and conda environment

Aye!
Indeed conda is installed in this image.
Just one last second...

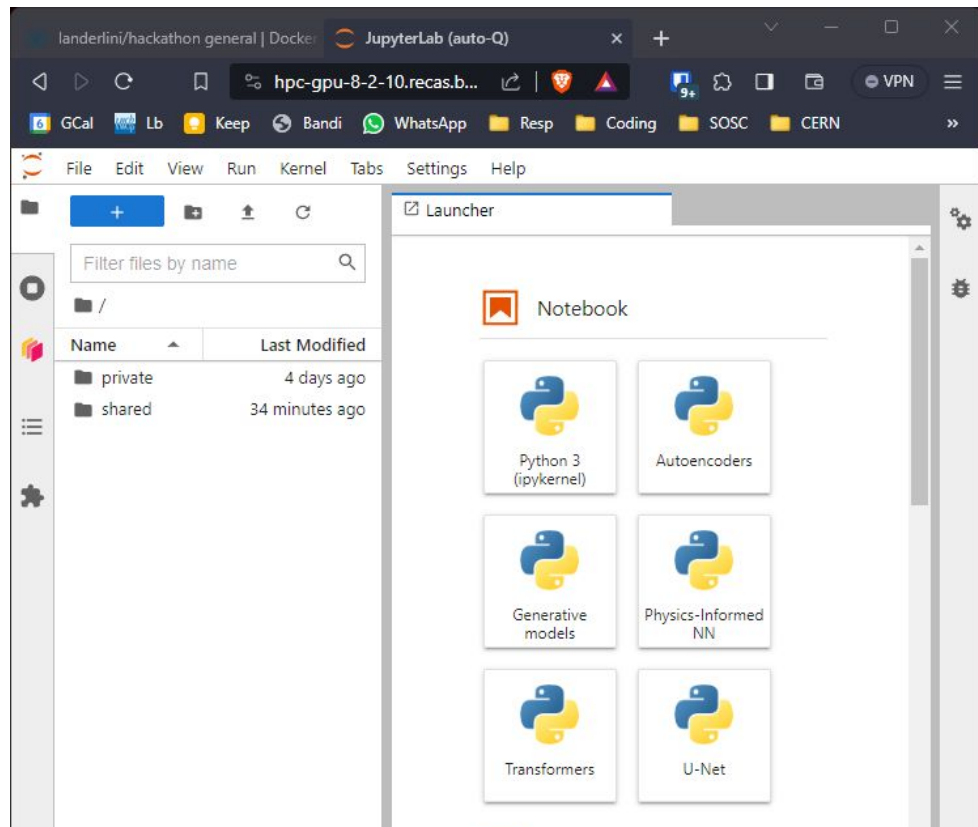
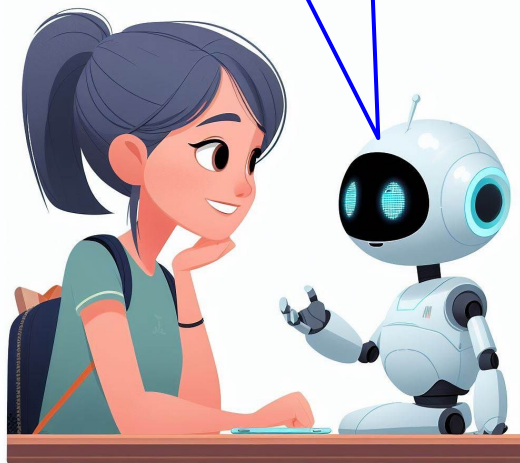


Spawning: Starting the user interface



Here you go...

That's it for me...
Have fun with
jupyterlab



User interface

Available conda environments installed at cluster level and distributed via NFS to the various nodes.
Will use `cvmfs` one day, eventually.

Dask dashboard
A door towards scalability (not for this hackathon)



The screenshot shows a Jupyter Notebook interface. On the left is a file browser with a search bar and a table of files. On the right is a 'Notebook' section with a grid of conda environments.

Name	Last Modified
private	4 days ago
shared	34 minutes ago

Environment Name
Python 3 (ipykernel)
Autoencoders
Generative models
Physics-Informed NN
Transformers
U-Net

Cluster volumes mounted here

Everywhere else is “Ephemeral Filesystem”
Any data will be lost when the container is turned off

The screenshot shows the JupyterLab 'File' menu with the following options: New, New Launcher (Ctrl+Shift+L), Open from Path..., New View for, New Console for Activity, Close Tab (Alt+W), Shutdown Terminal (Ctrl+Shift+Q), Close All Tabs, Save (Ctrl+S), Save As... (Ctrl+Shift+S), Save All, Reload from Disk, Revert to Checkpoint, Rename..., Download, Save and Export Notebook As..., Save Current Workspace As..., Save Current Workspace, Launch Dask Dashboard Layout, Print... (Ctrl+P), Hub Control Panel, and Log Out.

The terminal window in the background displays the output of the 'nvidia-smi' command:

GPU	GI ID	CI ID	MIG Dev	Memory-Usage	Vol	Shared
				BAR1-Usage	SM	CE ENC DEC OFA JPG
2	7	0	0	3MiB / 4864MiB 0MiB / 8191MiB	14	0 1 0 0 0 0

The terminal window shows the following commands and output:

```

root@jupyter-anderlini:/mlinf# source private/.bashrc
(base) root@jupyter-anderlini:/mlinf# conda env list
# conda environments:
#
      /envs/autoencoders
      /envs/pinn
      /envs/transformers
      /envs/unets
      /mlinf/private/myenv
base   * /usr/local/miniconda3
    
```

The 'nvidia-smi' command output is as follows:

GPU	Name	Perf	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr.	ECC
Fan	Temp	Pwr	Usage/Cap	Memory-Usage	GPU-Util	Compute M	MIG	En
0	NVIDIA A100-PCI...	28C	P0	57W / 250W	00000000:01:00:0	Off	N/A	Def
1	NVIDIA A100-PCI...	24C	P0	34W / 250W	00000000:25:00:0	Off	N/A	D
2	NVIDIA A100-PCI...	23C	P0	32W / 250W	00000000:C1:00:0	Off	N/A	Defa
3	NVIDIA A100-PCI...	23C	P0	33W / 250W	00000000:E1:00:0	Off	N/A	Defa

MIG devices:

GPU	GI ID	CI ID	MIG Dev	Memory-Usage	Vol	Shared
				BAR1-Usage	SM	CE ENC DEC OFA
2	7	0	0	3MiB / 4864MiB 0MiB / 8191MiB	14	0 1 0 0 0 0

4 nVidia A100 GPUs in the bare metal (no virtualization at ReCaS)

Alice's MIG device on GPU 2

But I need MY VERY SPECIAL PACKAGES!

No TensorBoard, no machine learning!!!!!!

I cannot work without VS Code!!!!!!!!!!!!!!!!!!!!!!



You are in the Cloud, man.

Just customize things.



Customization layers *(in the hands-on in a minute)*

Custom conda environments:

If you just need custom software, but the user interface is fine →

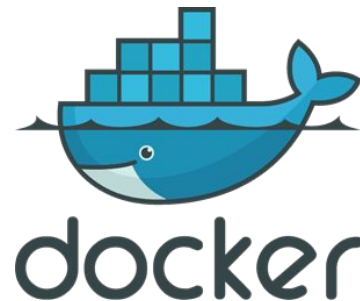
For example, installing TensorBoard



Custom Docker image:

If you need a custom user interface or anything does not fit in conda →

For example, installing VS code

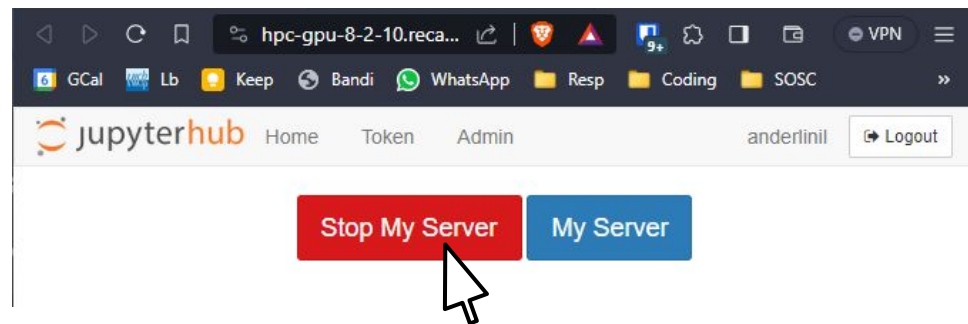
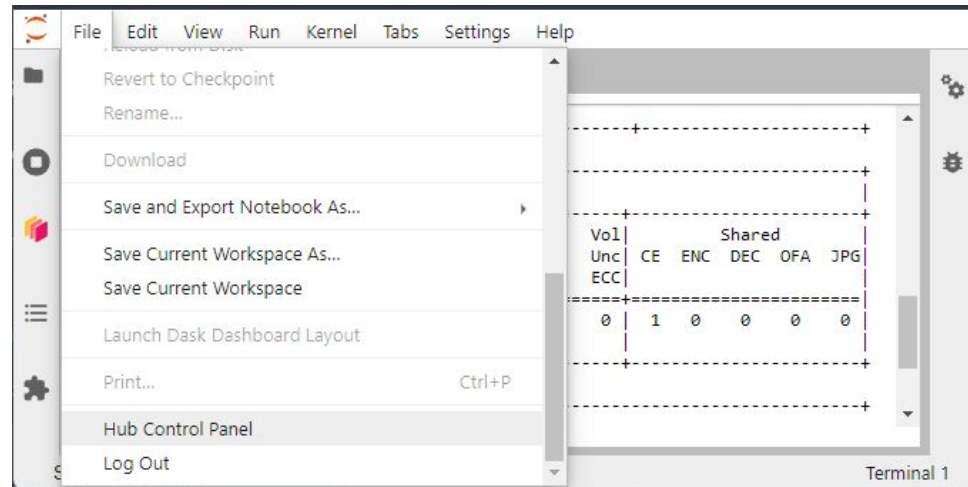
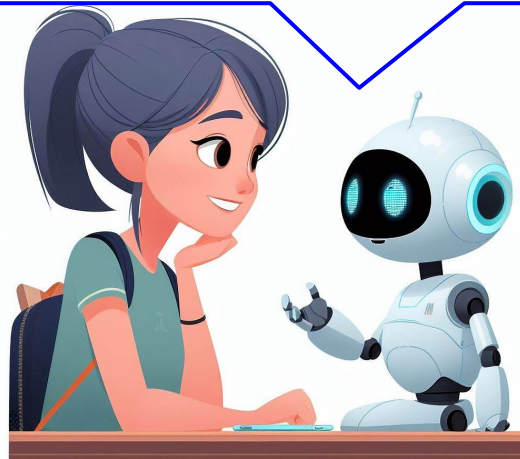


Culling: Starting the user interface

Ah, one last thing...

If I don't see you around for a while (few hours),
I will destroy your container to free the
 resources for others. Be prepared.

But please, consider **freeing resources yourself**
 when you are done



And what about *scaling*?

Once you have your custom environment, running your custom software in your custom Docker container, **it is time for you to *scale up*.**

You will want to run the same code, in a non-interactive way, multiple times, possibly in parallel.

This comes with two complications:

- **economic**: a single GPU to enable development has a negligible cost *wrt.* your salary. The cost of hundreds of GPUs processing buggy workloads while you sleep is unacceptable.
- **competition and arbitration**: policies for a **fair share** of the resources must be formulated and implemented.



The future...



The Fifth Scientific Committee of INFN (CSN5) has recently funded a *three-year project* to evolve ML_INFN facing these complications:

- Enhance our ability of using ALL (or at least most) resources in the Cloud if not employed otherwise (opportunistic usage), including FPGAs and Quantum Processors.
- Develop an infrastructure (*monitoring, batch system and user-support*) to enable implementing **extensive usage and fair share of resources** accessible to **multiple scientific communities** for hardware-accelerated machine-learning workloads.
- The European projects ICSC and Terabit are providing resources to develop and commission the whole thing.



Summary

- **JupyterHub** on **Kubernetes** is the door for **ML** and **Data Analysis** tasks in the Cloud.
- Resources available in **different servers in the same site** can be transparently combined and accessed as a single cluster (e.g. CNAF).
- The **same platform can be deployed in very different scenarios** (e.g. CNAF on VMs and ReCaS on bare metal) providing very similar user experience.
- Users can **customize their development environment** with the tools they are used to (e.g. **conda environments**) or Cloud-native solutions (e.g. **Docker containers**).
- In the coming months it will become possible to **scale up** to multiple instances accessing resources otherwise unused possibly from remote computing centers.

DISCLAIMER

*Please, consider yourselves as **beta testers of this novel setup** and be patient if something will need hot fixes during the hackathon.*