Third edition of the
Machine Learning @ INFN (ML_INFN)
advanced level hackathon

Open Science Cloud

Daniele Spiga INFN-Pg

Istituto Nazionale di Fisica Nucleare

Bari 21-24 Novembre 2022

Agenda

Introduction to Open Science Cloud… a disclaimer

- My personal view

OSC at INFN (My personal view)

- Few examples
- A focus on ML_INFN toolkits @ INFN-Cloud

Summary

spiga@pg.infn.it

Disclaimer

What is Open Science Cloud? a buzzword ?!?

- **EOSC:** European Open Science Cloud
- **SOSC:** School on Open Science Cloud
- **GOSC:** Global Open Science Cloud
- ...

I’d read it as the sum of two “concepts”:

- **Open Science:** An approach to the scientific process that focuses on spreading knowledge as soon as it is available using digital and collaborative technology (made of Expert groups, publications, news and events)
- **Cloud** computing: as a mean to use digital and collaborative technology making use of hosted services, such as data storage, servers, networking as well high level services and software over the internet

From a research e-infrastructure perspective (my view)

Allow researcher to exploit “free” and open services to manage workflow, build pipeline, data processing and analysis and, of course, to share/to reuse technical solutions

- Allow researchers to focus on science

Technical drivers:

- to enable users **to create and provision infrastructure deployments, automatically and repeatedly**, **with almost zero effort**.
- To Implement the ***Infrastructure as Code*** **paradigm** based on declarative approach: **allows to describe “What” instead of “How”**
    - Let the underlying system to deal with technicalities
- To promote (and support) **container-based solutions**
- To grant data sharing among users/infrastructures

spiga@pg.infn.it

…and from user perspective: few pillars

At first order I think end users should handle just few pillars

- What the user should/might see out of all of the underlying system?

Software management: a central role is played by container. A standard unit of software suitable to create **user tailored environment**, (share and port everywhere).

- Docker is an open source platform for building, deploying, and managing containerized applications:a **handy application encapsulation.**
- a de facto standard to manage runtime environments, and we use dockers everywhere [see later]
- Tip Docker store

Infrastructure management: in principle user might chose to know “nothing” about infrastructure (SaaS model and above).

- If a researcher need/want to customize its infrastructure, the system (the Cloud) should offer handles… **through templates** [see later]

INFN-Cloud

An **internal effort** at the INFN level in order to manage a (large) fraction of the INFN resources, in order to decouple user needs from the availability of local and dedicated hardware: this applies both to data and compute

Aims at providing solutions for a wide rage of user/community needs:

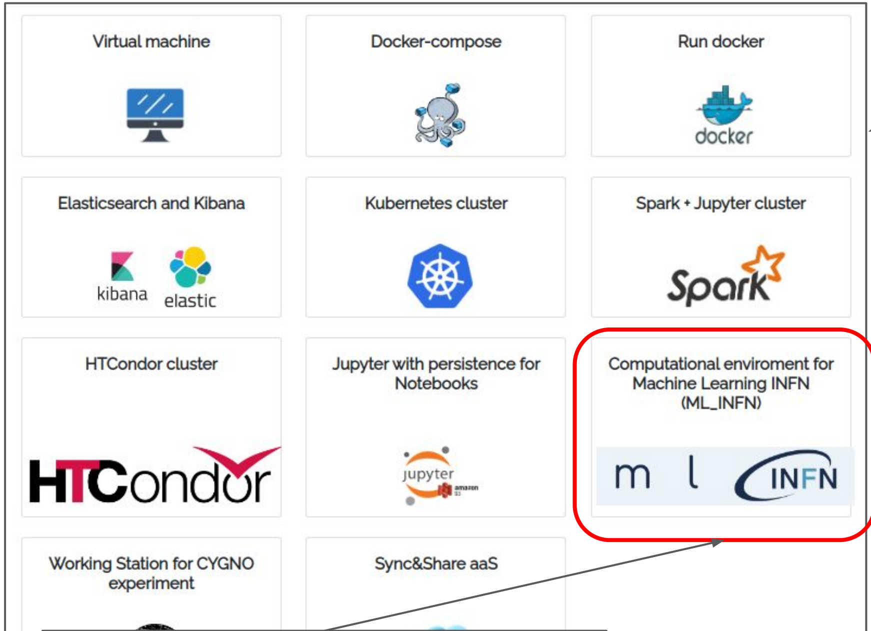
- Computing **Resources optimization**
- **Reuse** of solutions
- Support R&D: **design your computing model**
- A platform for **training**
- Ease (democratize) the access to the computing capacity:
    - Think to **access specialized hw such as: accelerators** (GPU, FPGA…)

Few highlights

spiga@pg.infn.it

A bit more in concrete

Services ready for use. Just instantiate your own



Example of a production ready use case [more lather]

Two major highlights

A ecosystem providing **a platform**, **toolkits**, **support and experience** to develop and prototype ad hoc solutions based on open source and industry standard (even cloud-native) technologies

- co-design and develop **customized solution for specific needs**

There are additional features and functionalities check here

So if you'd like to raise the bar

Examples…

Automation: Building your pipeline

- Exploiting cloud-native services to build a "event based" system/workflow

Workflow management: interactive processing

- Exploiting parallelism, Implement Interactive analysis workflow

Specialized hardware:

- "pioneered" by INFN-Cloud & ML_INFN joint venture (GPU access)

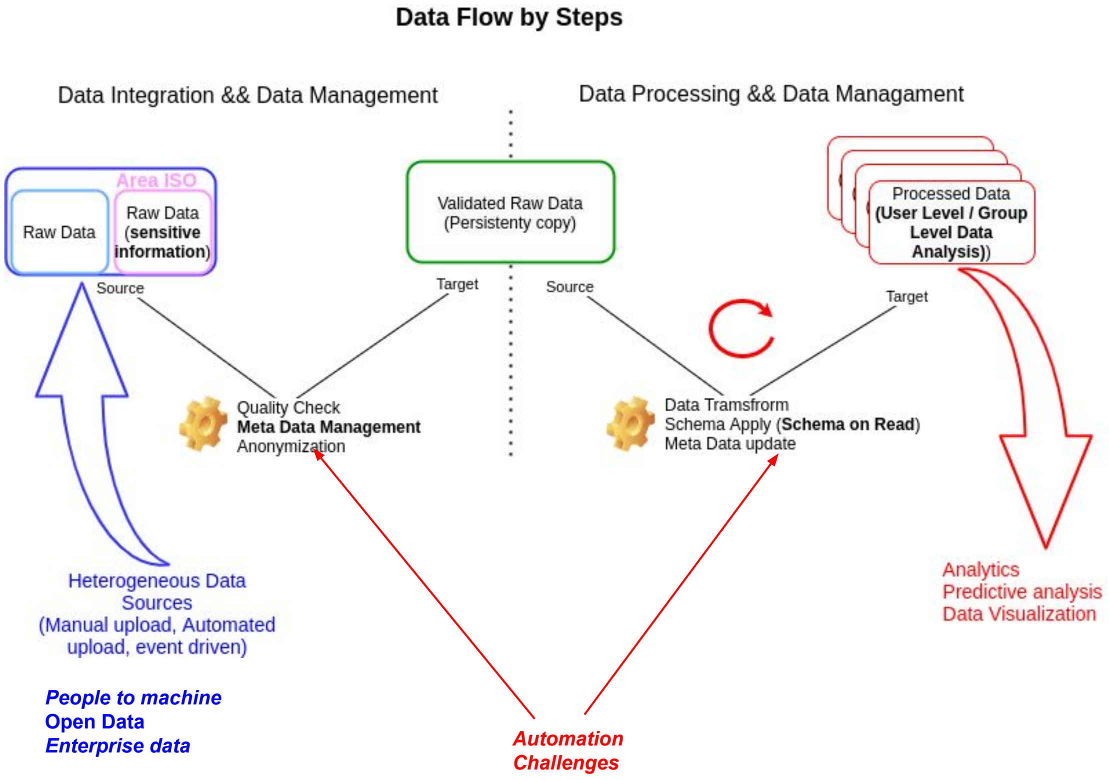
Several initiatives on going, two quite advanced, where this co-design is happening:

- CYGNO experiment, HERD Experiment (CSN2)

Workflow automation (a real case at INFN)

Some high level requirements:

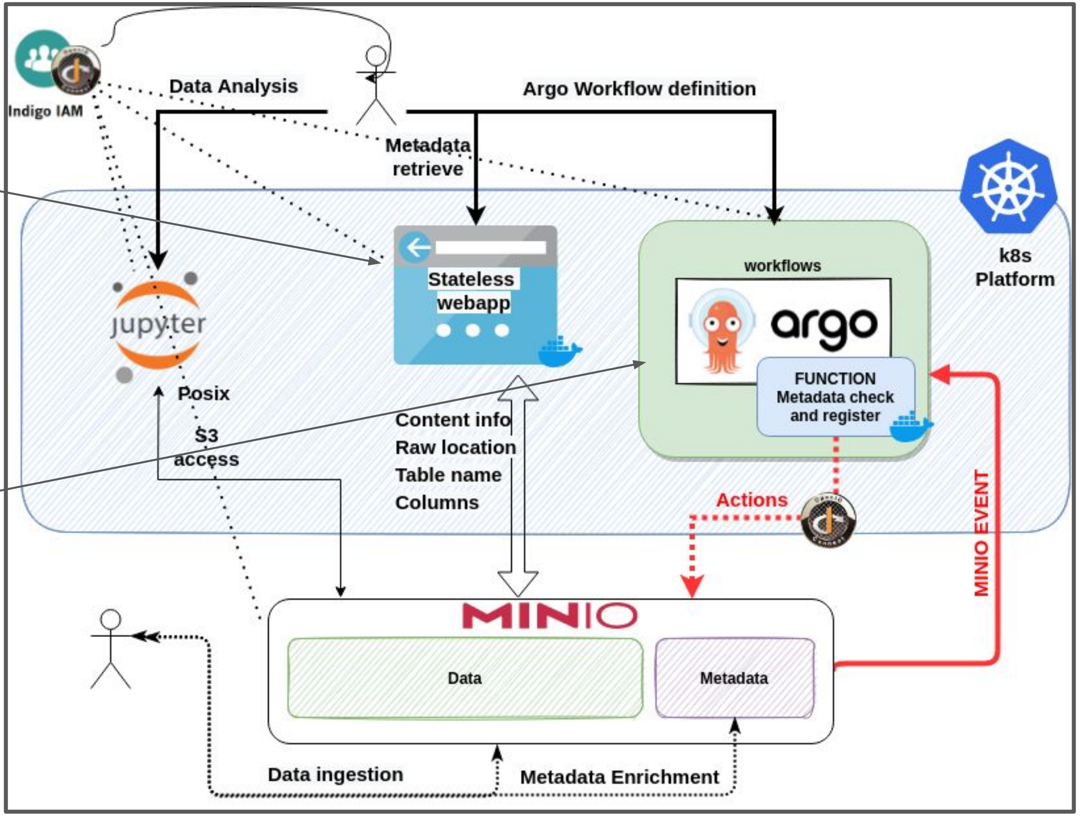
- **Structured and unstructured** data archival
- Preserve data in its **original format**
- **Enable automated data validation (and organization)**
- **Enable automated data pre-processing, transformation…**
- **Easily find your data**



A cloud-native platform would look like

The only custom service (containerized)

Argo Workflows is an open source container-native workflow engine for orchestrating parallel jobs on Kubernetes.



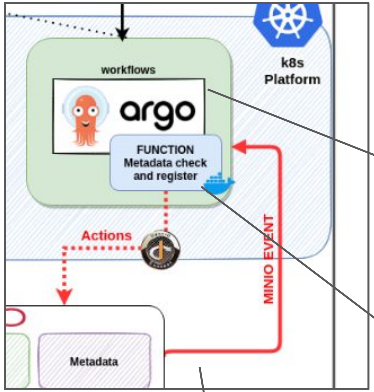
Almost everything is implemented with Industry Standard solutions

Container (docker) everywhere

Container Orchestration

- Self healing
- Automate
- Easy service deployment

Architecture of The PLANET experiment@CSN5



… under the hood (declarative, container…)

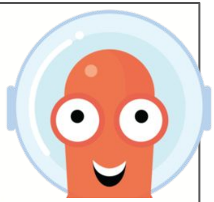
Argo Sensor detect the event and **trigger the validation**

The EventSource: Each upload generate a Minio EVENT

```
spec:
  minio:
    example:
      endpoint: 'planet-store.cloud.cnaf.infn.it:9000'
      bucket:
        name: demo-raw
      accessKey:
        name: artifacts-minio
        key: accesskey
      secretKey:
        name: artifacts-minio
        key: secretkey
      events:
        - 's3:ObjectCreated:Put'
```

MINIO

```
triggers:
  - template:
      name: minio-workflow-trigger
      k8s:
        source:
          resource:
            apiVersion: argoproj.io/v1alpha1
            kind: Workflow
            metadata:
              generateName: artifact-workflow-2-
              namespace: argo-events
            spec:
              entrypoint: hook
              templates:
                - container:
                    args:
                      - THIS_WILL_BE_REPLACED
                    command:
                      - hook
                    env:
                      - name: ACCESSKEYID
                        value: admin-creds
                      - name: ENDPOINT
                        value: 'planet-store.cloud.cnaf.infn.it:9000'
                      - name: SECRETACCESSKEY
                        value: '223^sU#0!ksss'
                    image: 'dodasts/planet-demo-hook:v0'
                    imagePullPolicy: Always
                  name: hook
```



A customizable validation function is automatically executed:

If metadata OK then:

tell Minio to move data to the validated bucket

else

tell MinIO to move data to triage && notify

fi

Custom

Put here your code

spiga@pg.infn.it

Managing workload: Transparent offloading

Distributing workload to parallelize data processing can be a complex task.

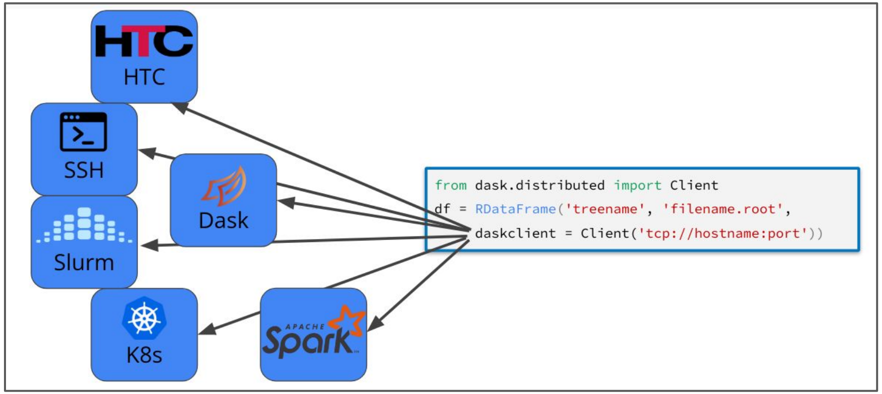
A dream would be to be able to access huge amount of computing capacity quickly and easily

- to process (huge amount of) data → **Interactive** or **Quasi interactive**
- reduce the time to insight: going interactive over huge amount of data

Simplifying: to being able to scale up to a full workstation and transparently scale out to a cloud

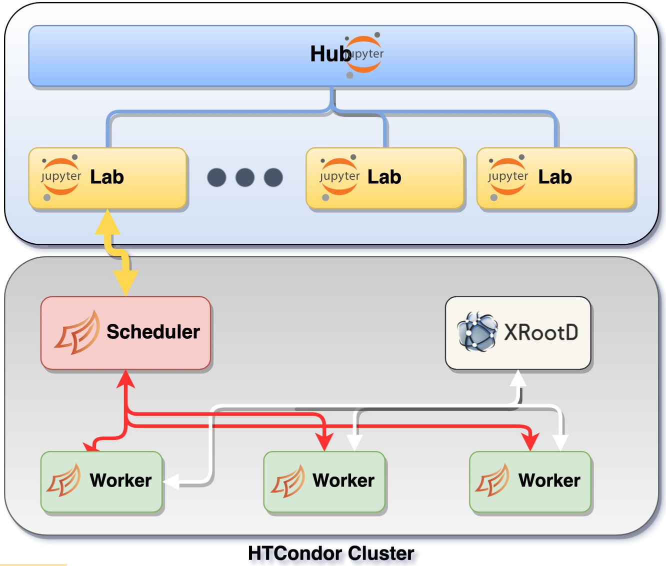
Several communities are exploiting the use high level frameworks capable of leverage distributed computing engine

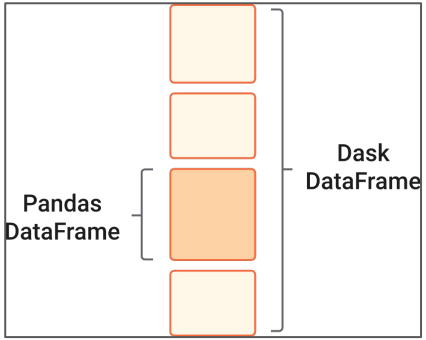
- I.e. RDataFrame is getting traction @HEP



An R&D @ INFN for HEP (CMS)

- **JupyterHub** (JHub) and **JupyterLab** (JLab) to manage the **user-facing part of the infrastructure**
  - **Comple abstraction**
- **DASK** to introduce the **scaling over a (highly) distributed system**
  - **Huge amount of resources, quickly and easily**
- **[XRootD is a bit HEP specific.. See it as a way to access any data anywhere]**





Dask is not HEP. It is a library that allow to scale the existing **Python and PyData ecosystem**.
- Looks and feels like the pandas API, but for parallel and distributed workflows.

R&D on interactive data analysis More details [here]

Early results at CMS

Measured two different workflow distribution approaches

- Using VBS SSWW with a light lepton and an hadronic tau in final state
    - ported from legacy approach (nanoAOD-tools/plain PyROOT-based) to RDataFrame.
- Data processed about 2TB (Data + Monte Carlo)
- The comparison tests are performed
    - on the same nodes of the cluster
    - very same HTCondor infrastructure
    - A fair benchmark.

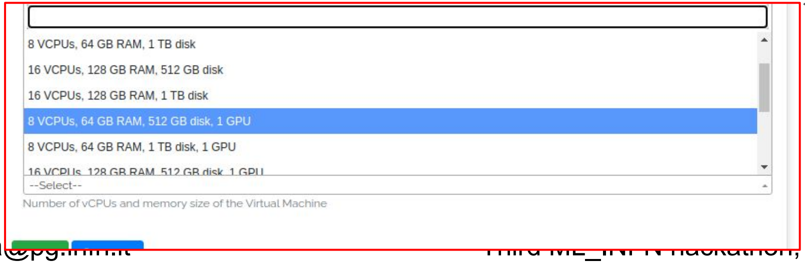
Preselection		
	Legacy	RDF (O2)
Overall time	3h 40min	25min
Overall rate	693 Hz	7306 Hz
Event-loop rate	721 Hz	8473 Hz
Overall network read	488 GB	371 GB
Average RSS per-node	Ca. 13 GB	Ca. 17 GB

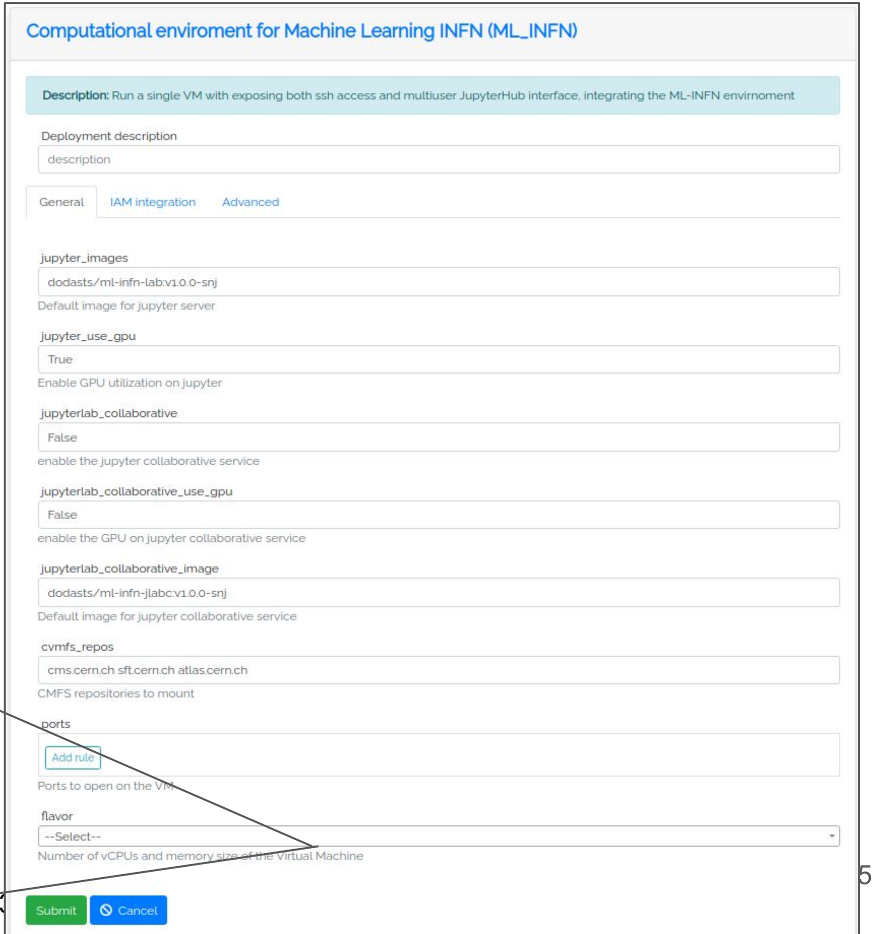
Postselection		
	Legacy	RDF
Overall time	0.25h	0.08h
Overall rate	306 Hz	855 Hz
Event-loop rate	412 Hz	1976 Hz
Overall network read	11 GB	10 GB
Average RSS per-node	Ca. 1 GB	Ca. 15 GB

ML-INFN @ INFN-Cloud

Build and promote the adoption of performant and tailored technological platforms because a effective platform for Machine Learning prototyping and developing might represent a technical challenge

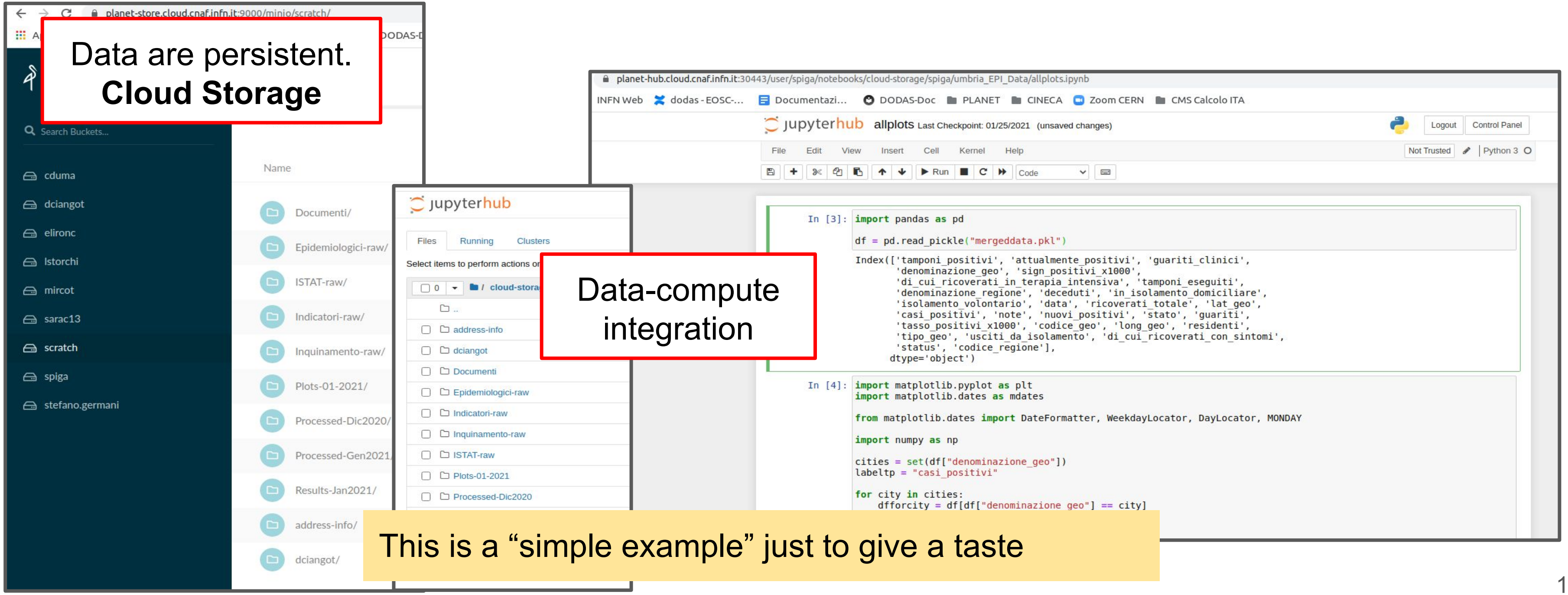
- **access to hardware accelerators** (GPUs) and possibly a non-limiting **access to data** (i.e. training data).
- Handles to **create user tailored environment** is a key
- and finally groups need to **collaborate and share resources, data and code**.





Data and compute: connecting (a few) dots

The ultimate goal is to seamlessly integrating data and compute (**The INFN DataLake**).



Data are persistent.
Cloud Storage

Data-compute integration

This is a “simple example” just to give a taste

Summary

Open Science Cloud: presented a personal point of view and INFN perspectives

- At INFN we are build and growing an ecosystem for these technical matters

An overview of tools and solutions in the scope of the INFN-Cloud Portfolio

- Where to start and where to possibly contribute (idea and solutions)

Few R&D examples made with the aim to stimulate questions, curiosity and… possibly sinergies

Mentioned R&D developments are there thanks to the work of many people

- Credits: INFN-Cloud Team; CMS Italy computing team; PLANET Team; ML_INFN Team