Minio and Jupyter with persistence for notebooks

Uso e sviluppo di applicazioni e servizi su INFN Cloud (CLueApp)
13-16 Sept 2022

Federica Fanzago – INFN Padova
federica.fanzago@pd.infn.it

Outline

- Jupyter notebook and JupyterHub: a short introduction
- Persistence: the INFN Cloud object storage
- How to deploy "Jupyter with persistence" in INFN Cloud
- Example of usage (live)

Jupyter notebook

- Project started in 2014 by the Jupyter project organization (https://jupyter.org)
- Open-source web application that provides an interactive environment to create documents called notebooks (.ipynb)
  - A notebook can contain formatted text, equations, images and code
  - The Jupyter notebook provides interpreters for various programming languages as python, R, Julia (kernels, default ipython)
  - The interactive code is executed via browser
  - Single user, notebooks can be exported and shared via git, email…
- Very useful for data science and scientific computing

Jupyter notebook installation and interface

- Prerequisite: python (3.3 or greater)
  - Using the python's package manager, pip, or installing anaconda (python distributed platform) (https://www.anaconda.com) or container.

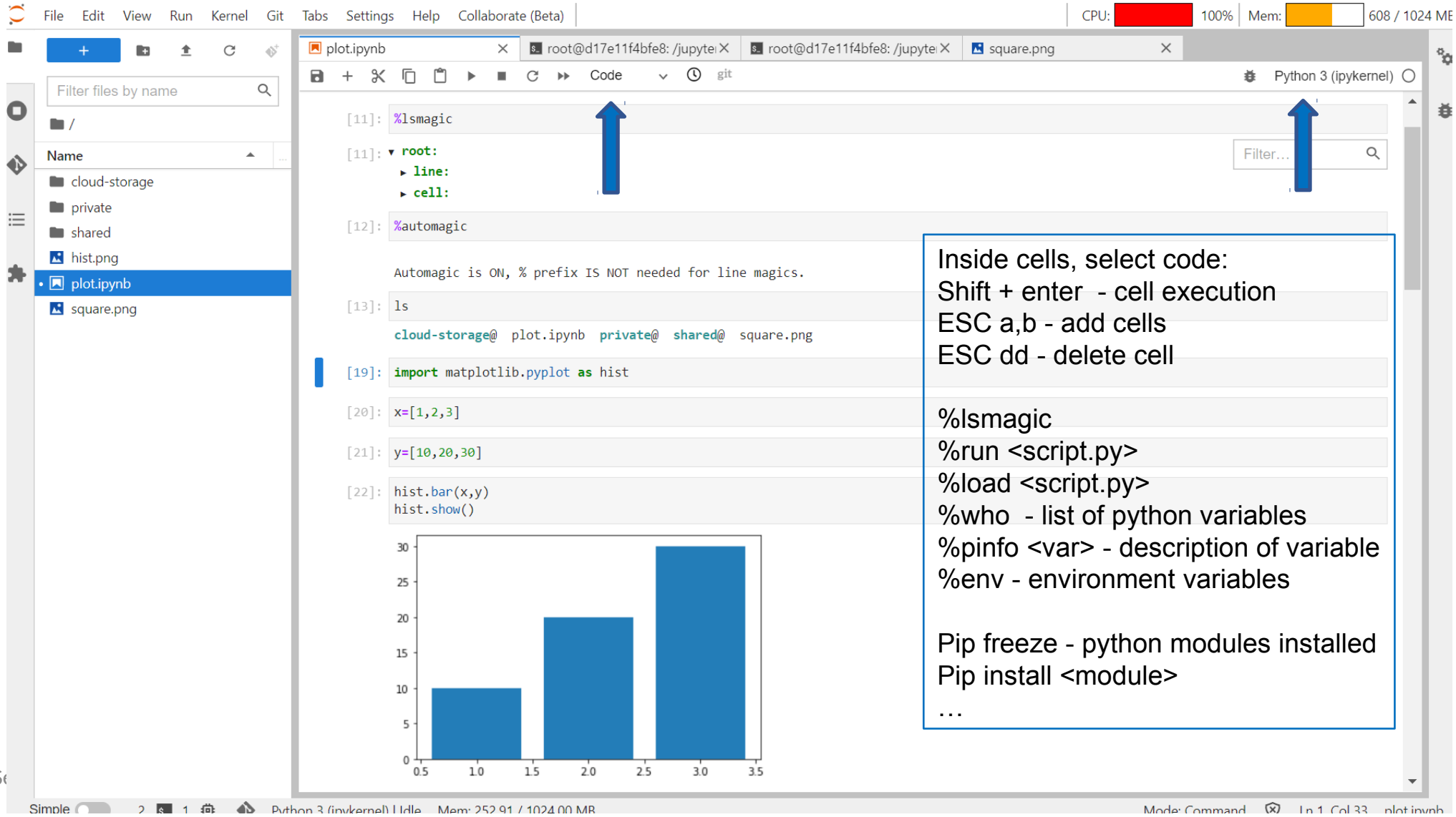
> pip3 install jupyter

- The Jupyter Notebook interface allows the management of kernels and notebooks

> jupyter notebook
> It starts the server and returns web url

- The interactive work is done inside "cells" that can contain and execute code, commands and formatted text
- Each cell is executed in sequential way. During the execution, an [*] is shown. After the execution, its sequential number [x].

Cells and commands in the notebook

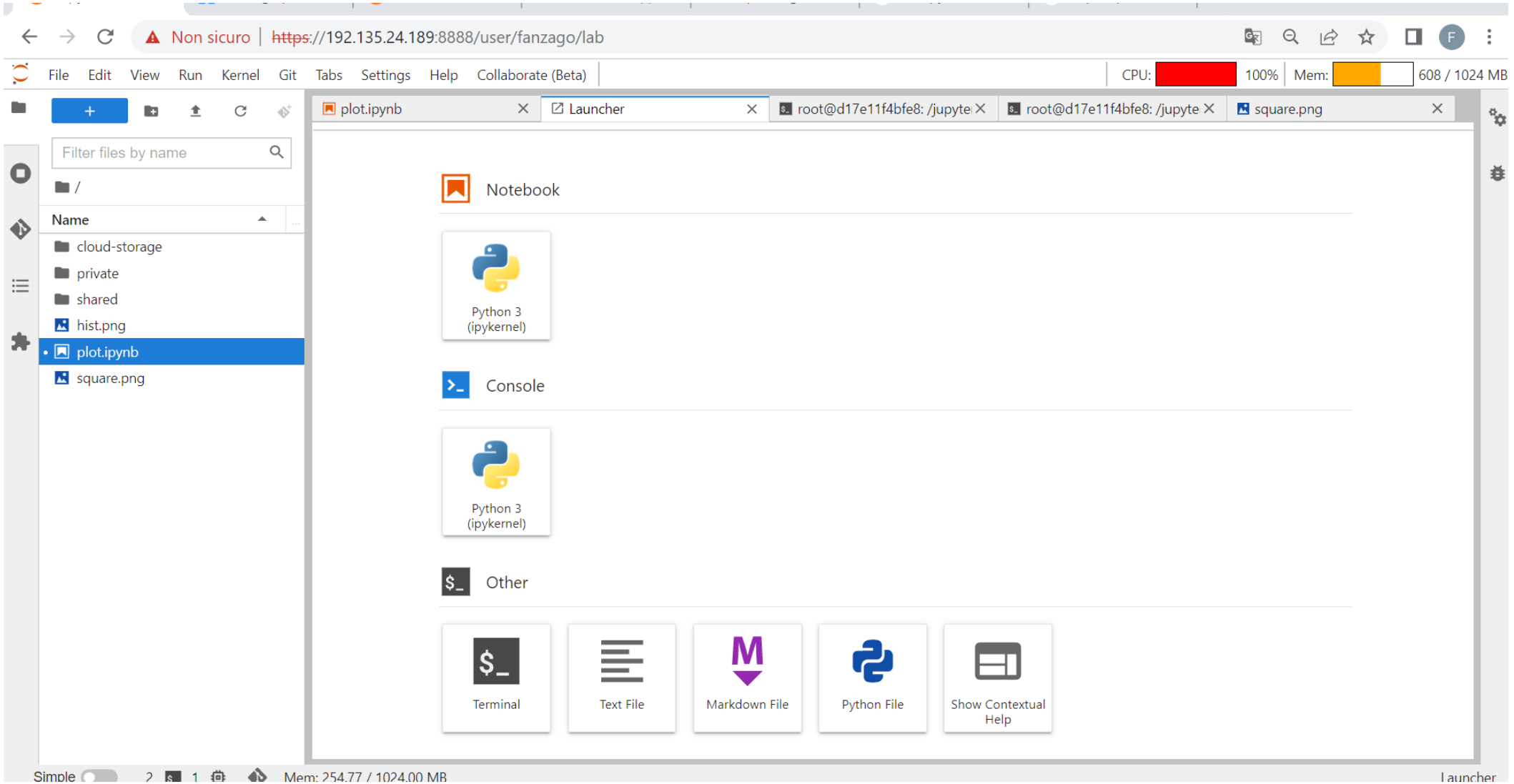


JupyterLab and JupyterHub

- The JupyterLab is a more extensible and composable interactive computing interface for more complex workflows.
  - It is an integrated environment that aggregates in a single interface more tools, included Notebook.
- The JupyterHub allows group of users to use “private” Jupyter notebook server in collaborative way on shared resources.
- Some INFN Cloud services are implementing Jupyter (hub + lab)

JupyterLab is currently the default in Jupyter docker images

JupyterLab dashboard



INFN Cloud services based on Jupyter

https://www.cloud.infn.it/service-catalogue

Compute Services

A list of services that enable a specific cloud technology

Analytics

A collection of ad-hoc solutions for analytic purpose

Machine Learning

List of ready-to-use Machine Learning services

Data Services

Data management and storage services

Scientific Community Customizations

Customized environments

Elasticsearch and Kibana

Deploy a virtual machine pre-configured with the Elasticsearch search and analytics engine and with Kibana for simple visualization

Spark + Jupyter cluster

Deploy a complete Spark 3.0.1 + Jupyter Notebook on top of a Kubernetes (K8s) computing cluster

Jupyter with persistence for Notebooks

Run Jupyter on a single VM enabling Notebooks persistence

Centralized Notebooks as a Service

Use the INFN Cloud centrally managed Jupyter Notebooks as a Service Solution

Endpoint: https://hub.cloud.infn.it

Hands-on

No sys-admin nomination

Scientific Community Customizations

Customized environments

Working Station for CYGNO experiment

Run a single VM with all the CYGNO envirnoment exposing both ssh access and Jupyter

A collection of ad-hoc solutions for analytic purpose

Machine Learning

List of ready-to-use Machine Learning services

Working Station for Machine Learning INFN (ML_INFN)

Run a single VM with all the ML-INFN environment exposing both ssh access and Jupyter

Where to store notebook documents?

- Saving document: the default path is the local “jupyter-workspace” directory of machine where the notebook server is running
  - It can run also in a VM or container
    - Data are lost if the VM is deleted, or the container removed
- Need to have a “permanent” space where to store them and guarantee persistence of data.
  - In INFN Cloud the object storage responds to this request

Object storage in INFN Cloud

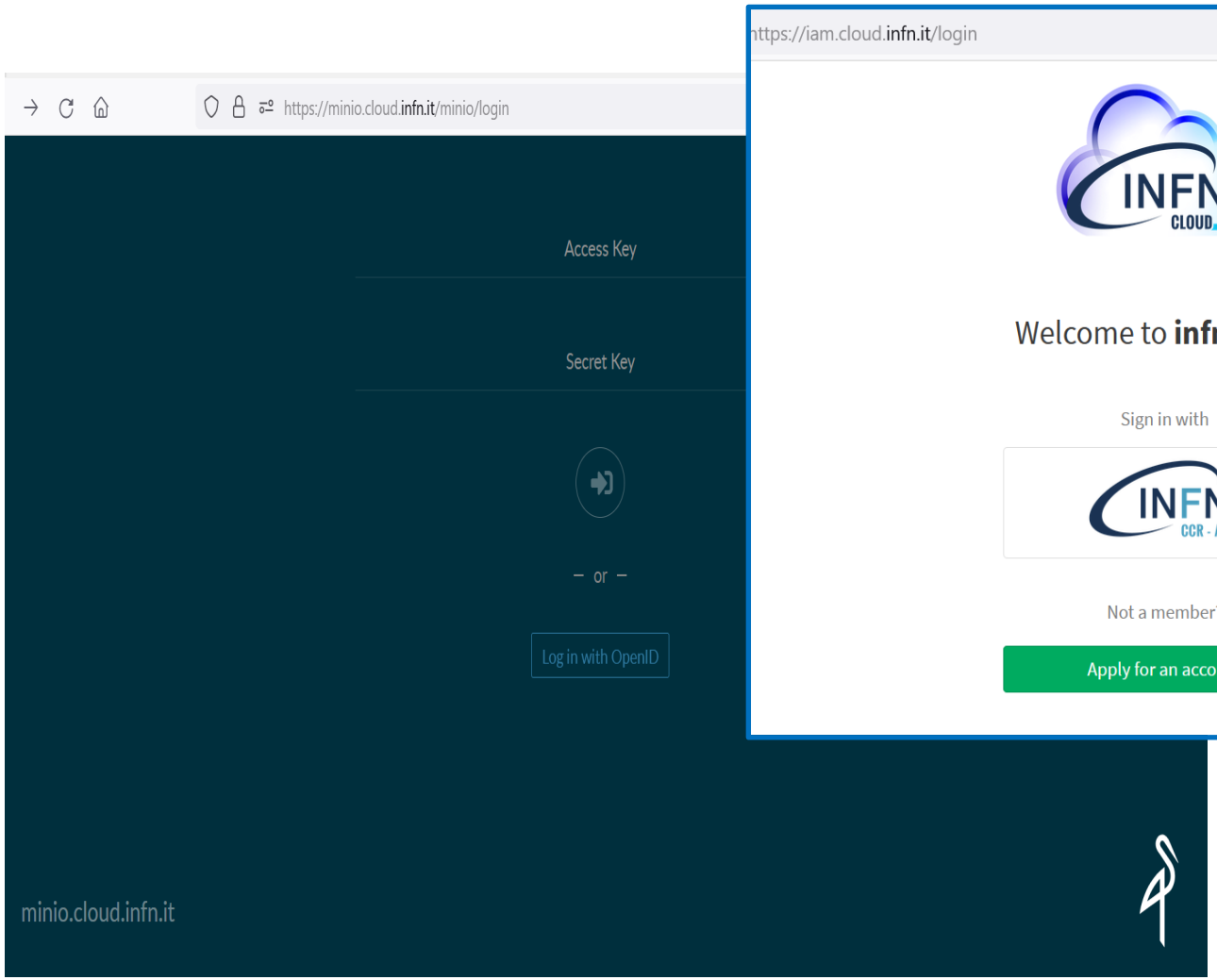
- INFN Cloud provides object storage based on Openstack Swift, as centrally managed service.
- The object storage is replicated in the two sites of backbone, Bari and CNAF
  - It guarantees the redundancy of data
  - There is not backup of data
- INFN Cloud is using the Minio-gateway software on Swift
  - It provides a simple web interface to access the storage, create buckets and manage files.
  - It provide also S3 api for the access to object storage via code
- Service URL https://minio.cloud.infn.it

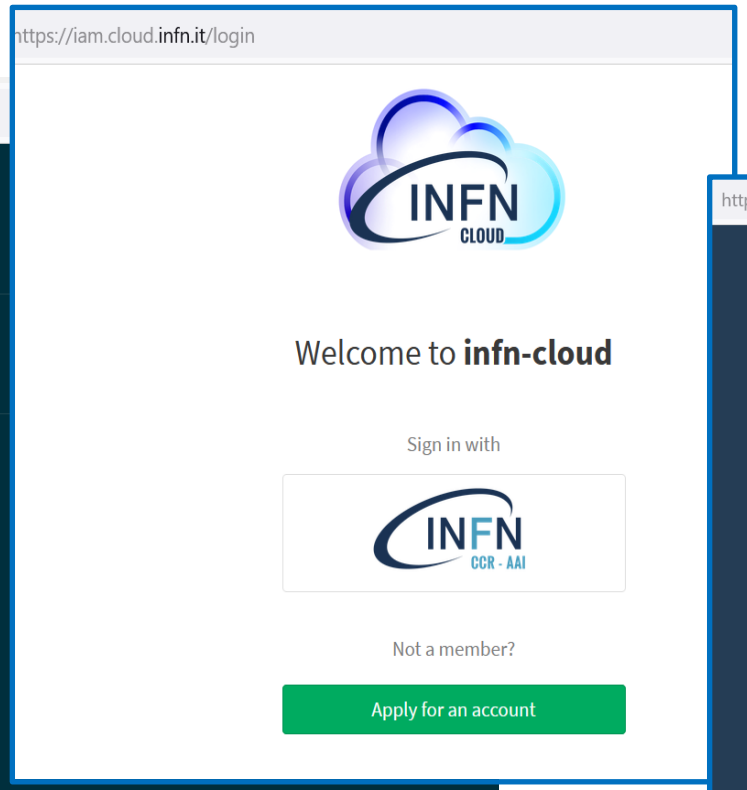
Persistent data with object storage

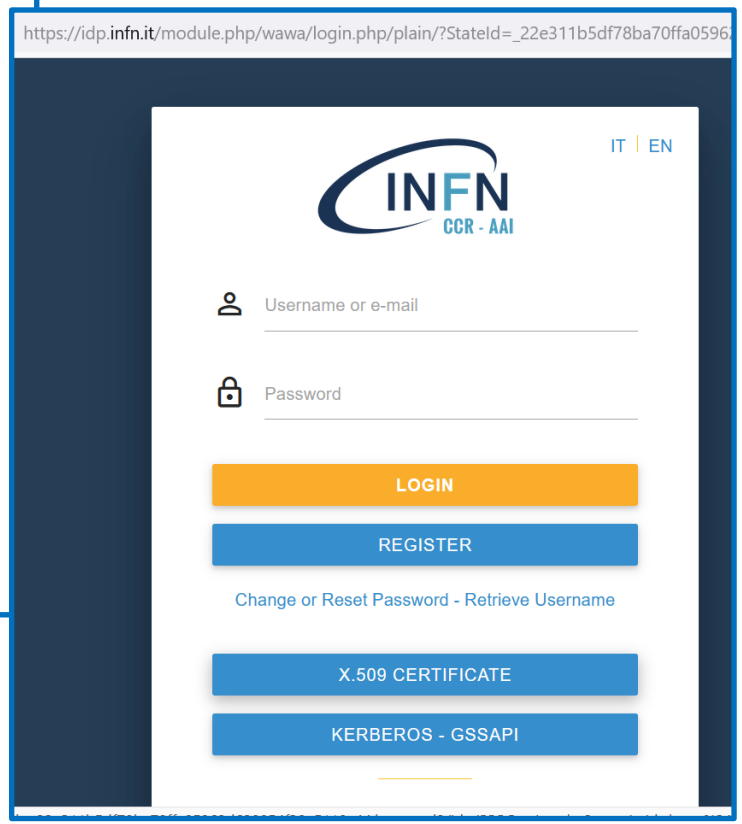
- Only authorized users can access the storage service (authentication and authorization via IAM)
- Its configuration allows users to have a personal bucket (directory) labeled as the username and an area under /scratch.
- The quota for each user is 200GB.
- Data stored in the personal bucket are private, data under /scratch are visible and downloadable by all.

Login to webui via OpenID

https://minio.cloud.infn.it





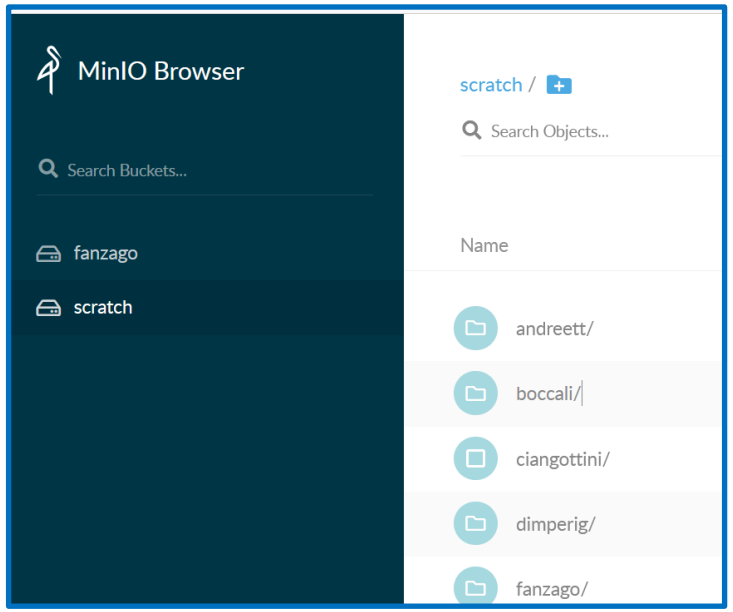


Persistent data with object storage

- All the services running notebooks mount these areas, so they are visible in the file system as posix directories.
  - cloud-storage dir

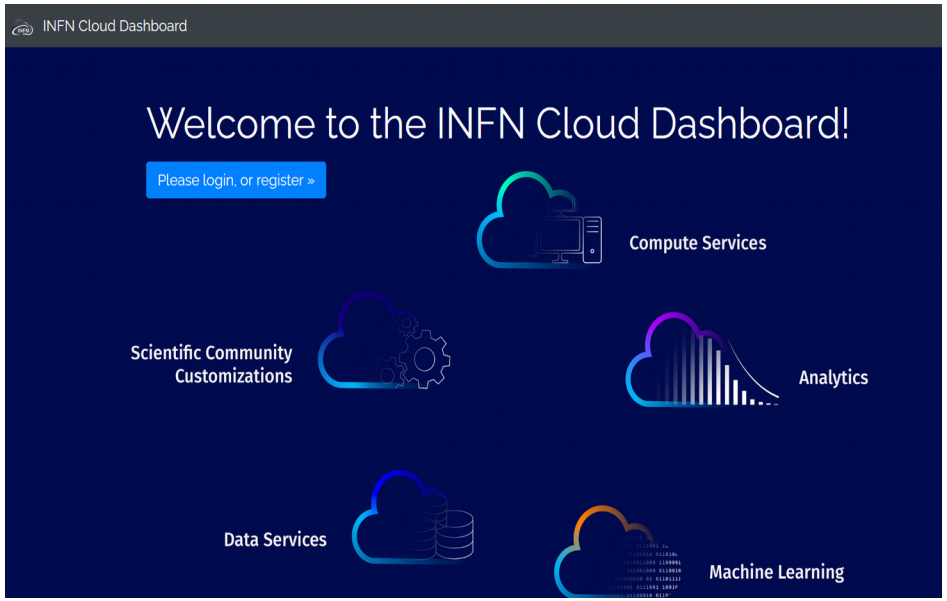
Using sts-wire, fuse and RClone

- The personal bucket can be created by user accessing the Minio webui otherwise it is automatically created the first time user instantiates a notebook.

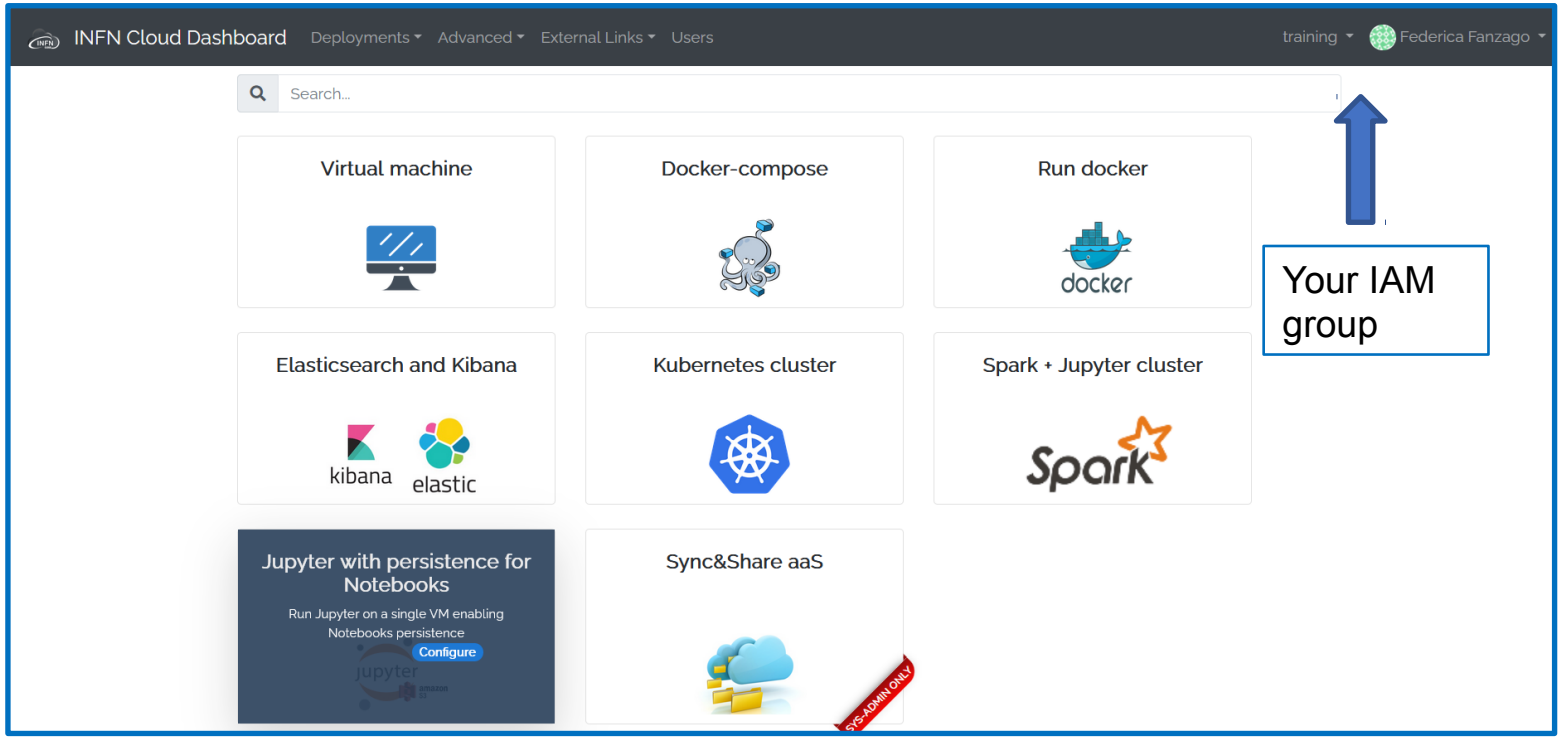


“Jupyter with persistence” deployment

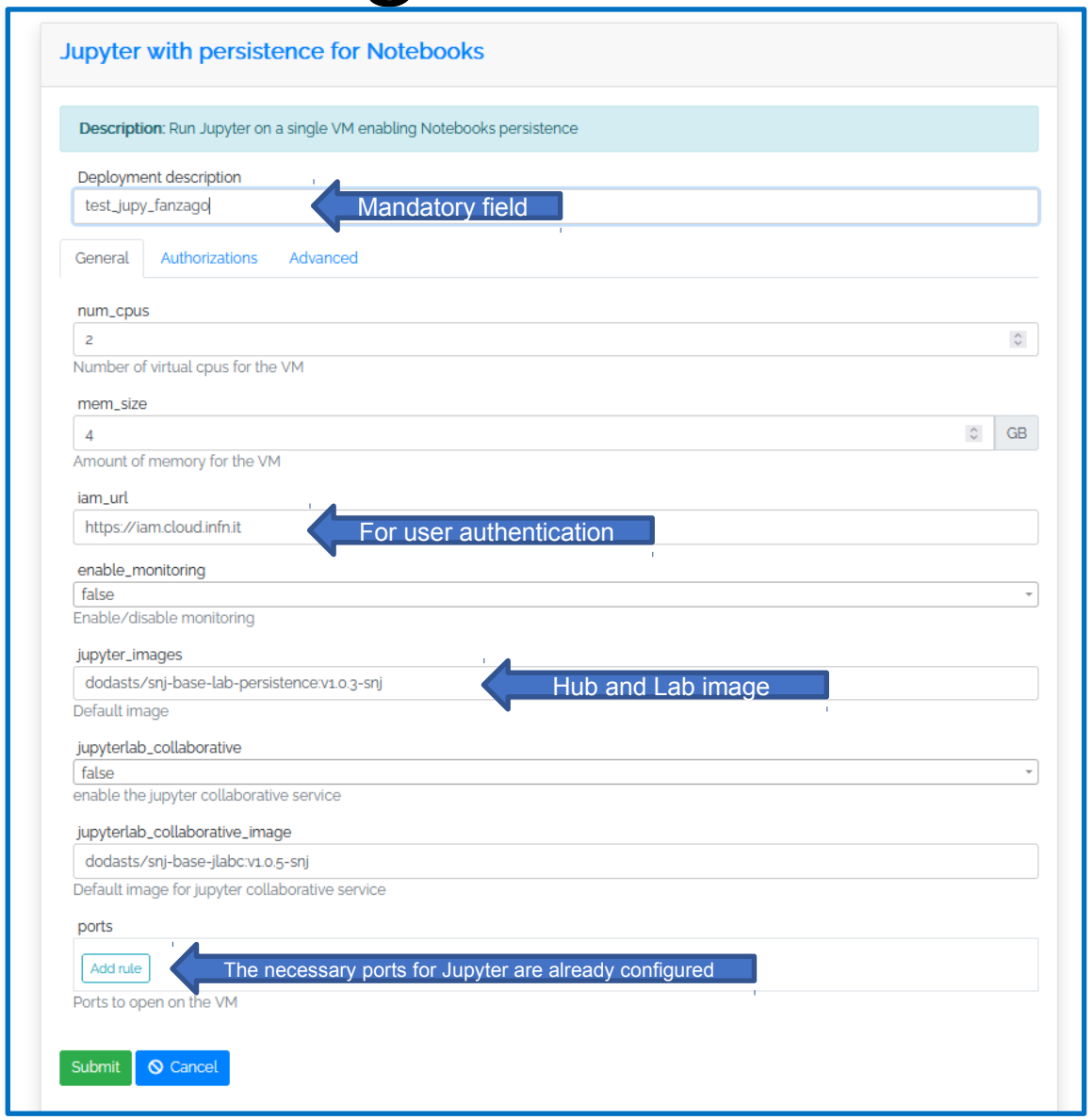
- Login in the INFN dashboard https://my.cloud.infn.it and select the “Jupyter with persistence” button. Then configure the service filling the form.

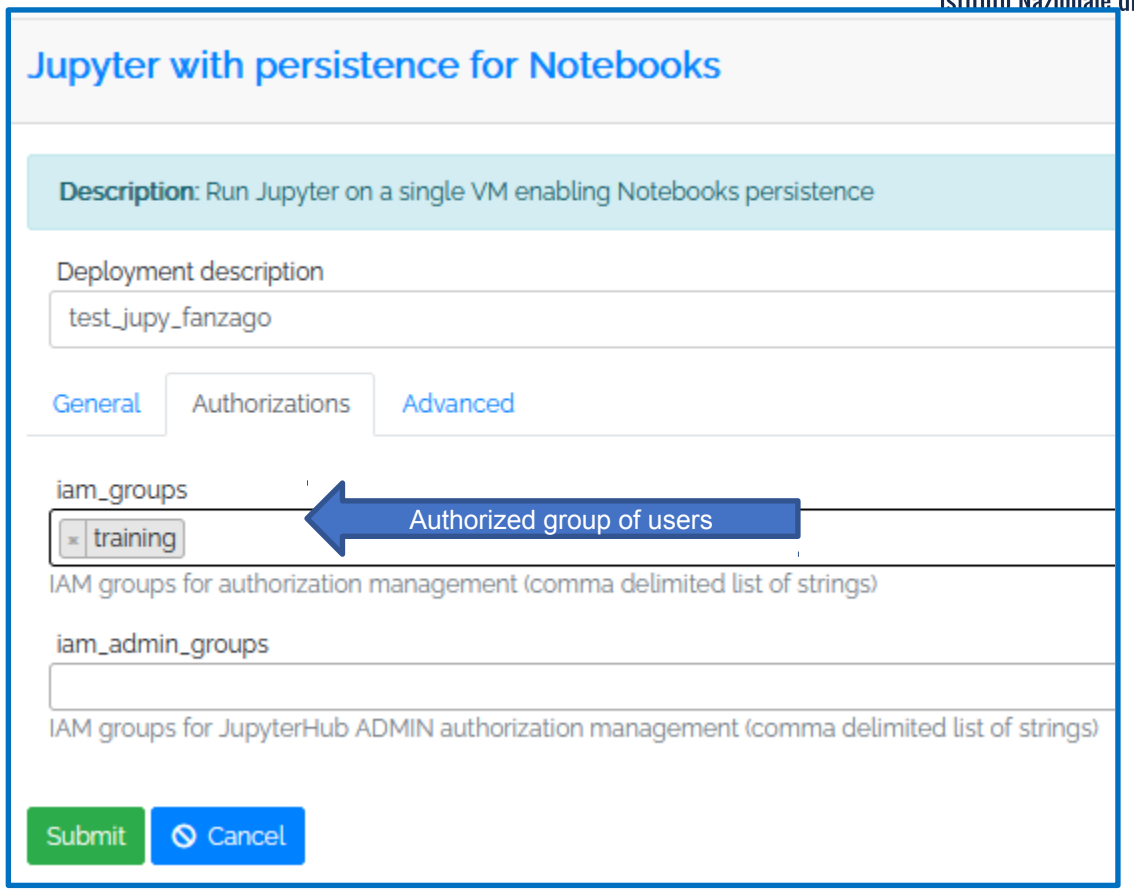


Authentication and authorization via IAM



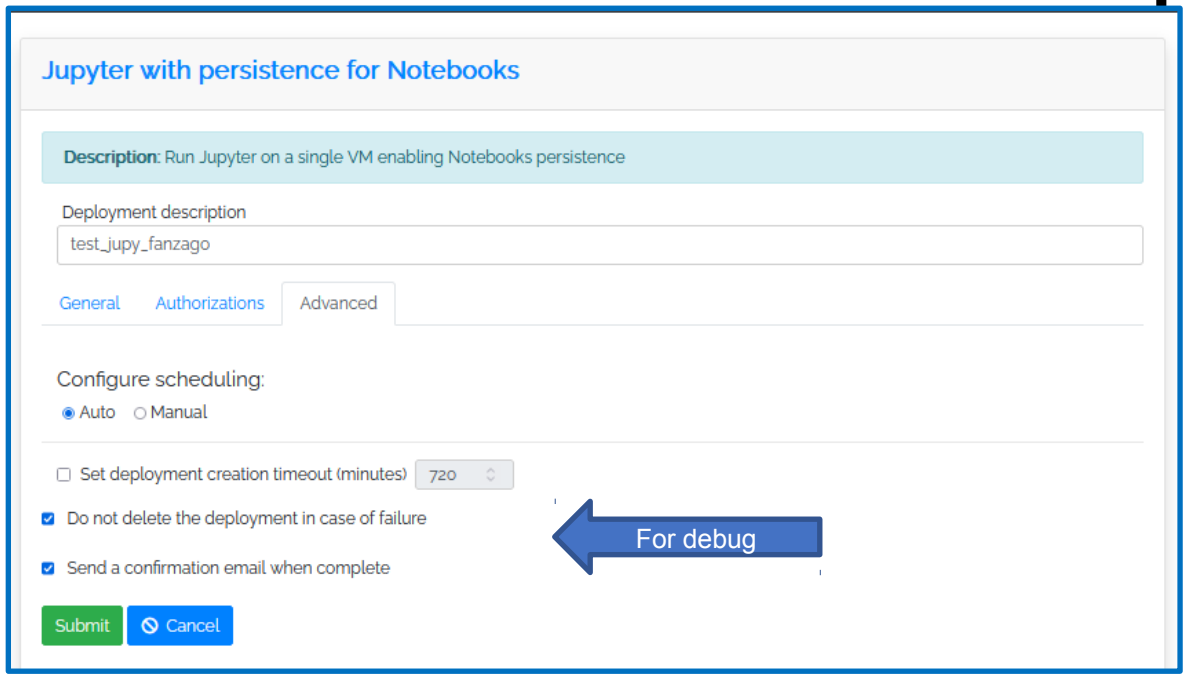
Filling the form…

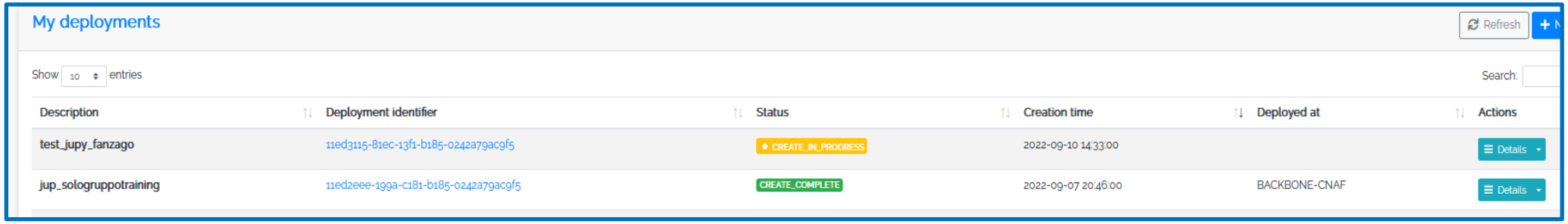




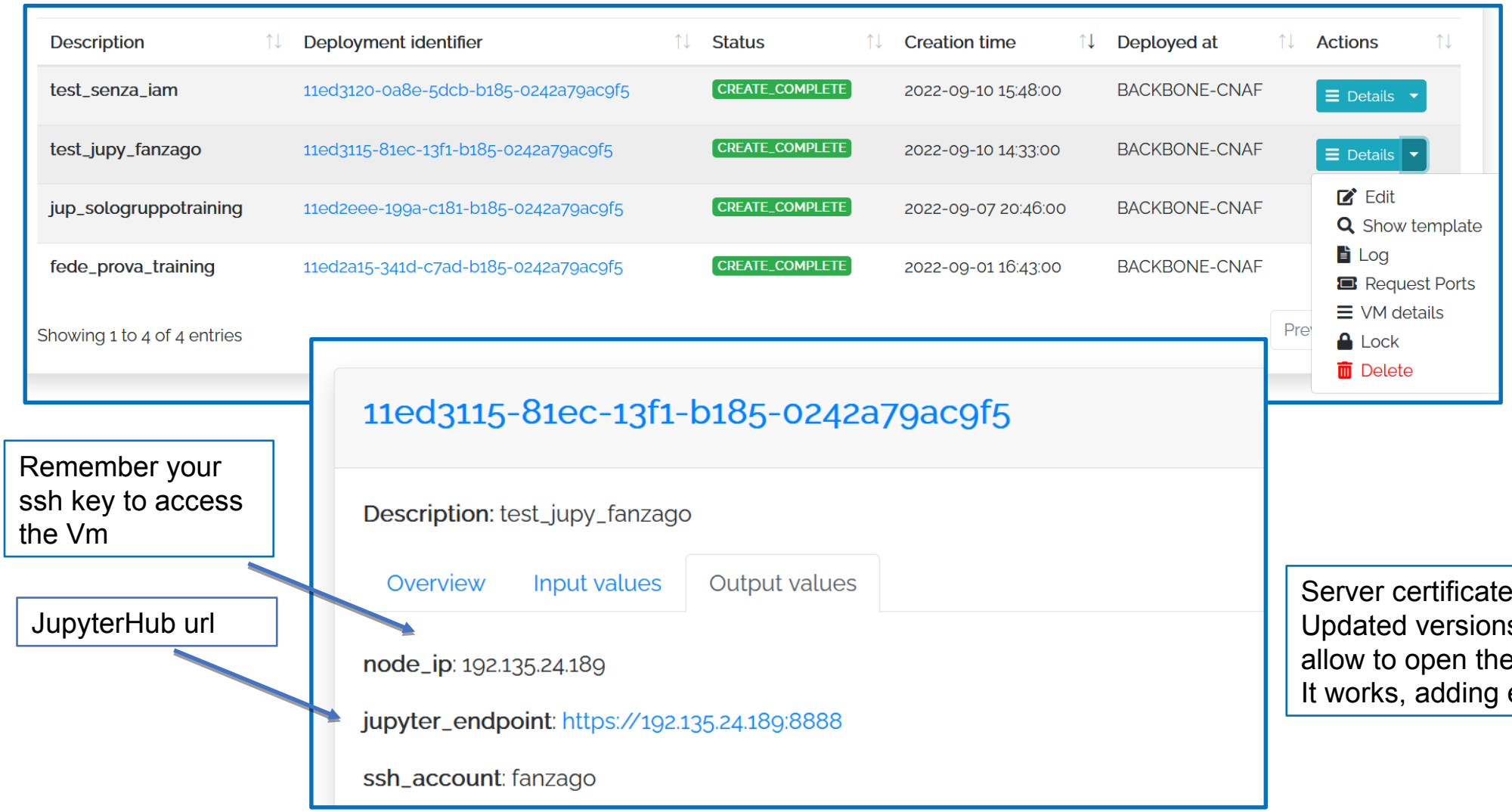
User guide:
https://guides.cloud.infn.it/docs/users-guides/en/latest/users_guides/howto13_jh_with_persistence.html

…then submit the deployment





Deployment details

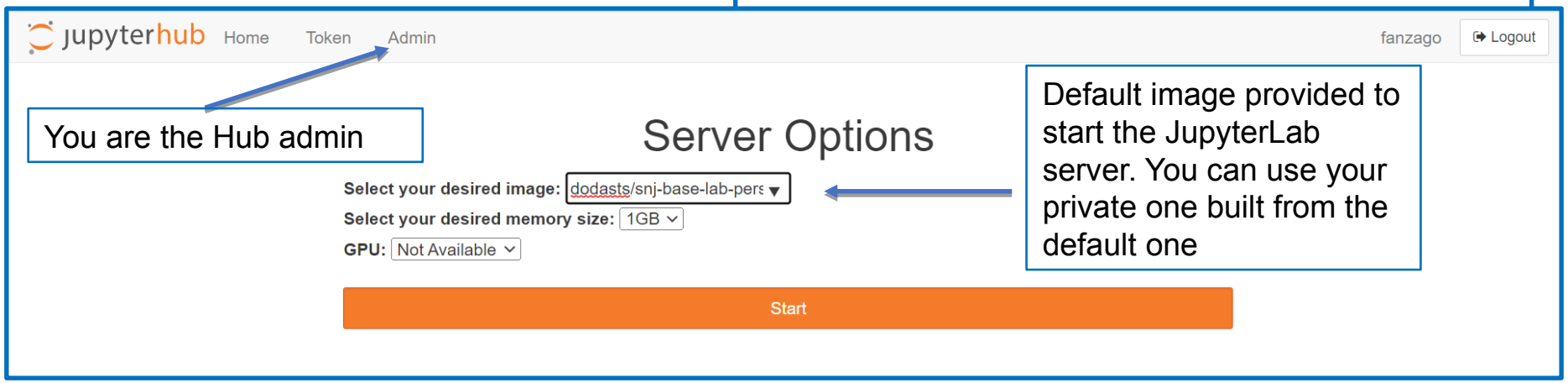


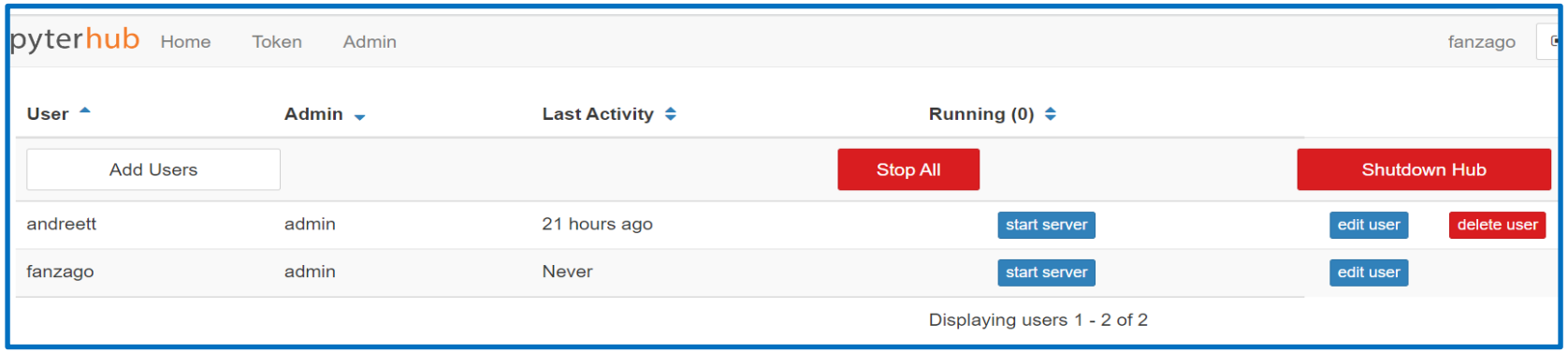
Remember your ssh key to access the Vm

JupyterHub url

Server certificates are self-signed. Updated versions of Firefox in Linux don't allow to open the link for security reasons. It works, adding exception, in Chrome

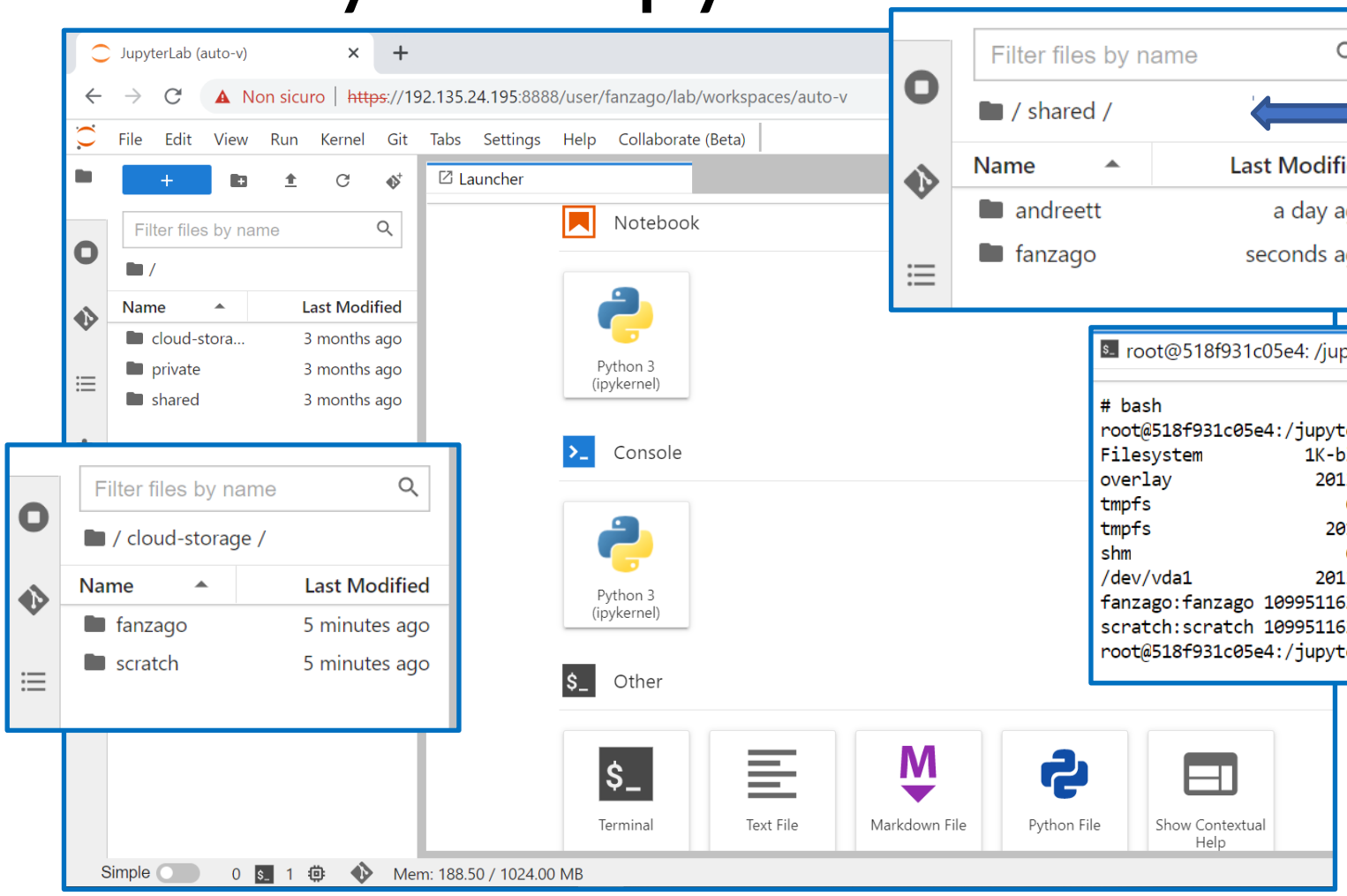
Your Jupyterhub

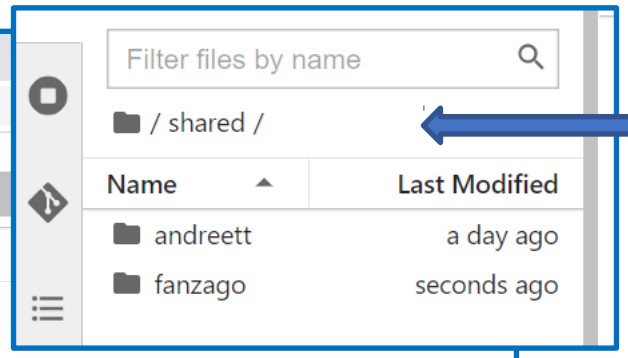




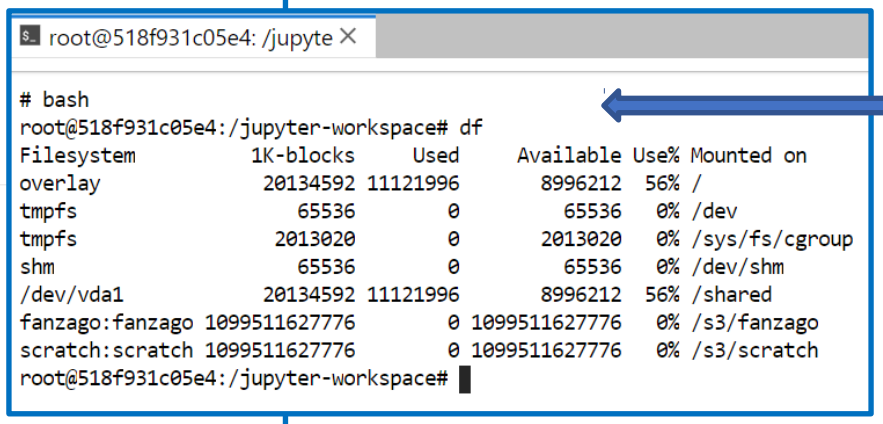
Hub's admin can control users and notebooks

And your JupyterLab





Shared area among users of authorized group



Inside Lab container

Best practice: don't execute notebooks in the cloud-storage area but run them in the "local" one. Use the object storage only as archive.

Live example

- Access to https://minio.cloud.infn.it
- Create your bucket
- Instantiate your “Jupyter with persistence for Notebook”
- Upload a file
- Install python libraries and run some code

References

- INFN Cloud homepage: https://www.cloud.infn.it/
- User guides: https://guides.cloud.infn.it/docs/users-guides/en/latest/
- Service catalogue: https://www.cloud.infn.it/service-catalogue/
- INFN Cloud dashboard: https://my.cloud.infn.it/
- INFN Cloud Object Storage service: https://minio.cloud.infn.it
- INFN Notebook as a Service: https://hub.cloud.infn.it
- INFN Cloud support: https://servicedesk.cloud.infn/it

Containers in the VM

```
fanzago@vnode-0:~$ sudo docker ps
CONTAINER ID   IMAGE                                          COMMAND                  CREATED          STATUS                  PORTS
                                                                                                   NAMES
614198cc94df   dodasts/snj-base-lab-persistence:v1.0.3-snj    "tini -s -- jupyterh…"   36 seconds ago   Up 32 seconds           8889/
tcp                                                                                                jupyter-fanzago
fd1fd7782f92   dodasts/monitoring-grafana:v1.0.5-monitoring   "/run.sh -config /op…"   3 days ago       Up 3 days               0.0.0
.0:3000->3000/tcp, :::3000->3000/tcp                                                               monitoring_grafana
cac3853d61d1   prom/prometheus:v2.30.3                        "/bin/prometheus --c…"   3 days ago       Up 3 days               0.0.0
.0:9090->9090/tcp, :::9090->9090/tcp                                                               monitoring_prometheus
c1053aabc34c   gcr.io/cadvisor/cadvisor:v0.37.5               "/usr/bin/cadvisor -…"   3 days ago       Up 3 days (healthy)     8080/
tcp                                                                                                monitoring_cadvisor
21c06e24da05   prom/node-exporter:v1.2.2                      "/bin/node_exporter"     3 days ago       Up 3 days               9100/
tcp                                                                                                monitoring_node_exporter
093e4de1f276   dodasts/snj-base-jhub:v1.0.2p1-snj             "/usr/bin/python3 /u…"   3 days ago       Up 3 days               8000/
tcp                                                                                                jupyterhub_jupyterhub_1
e0af2178ec3f   jupyterhub/configurable-http-proxy             "/srv/configurable-h…"   3 days ago       Up 3 days               0.0.0
.0:8001->8001/tcp, :::8001->8001/tcp, 8000/tcp, 0.0.0.0:8888->8888/tcp, :::8888->8888/tcp          jupyterhub_http_proxy_1
fanzago@vnode-0:~$
```