INFN Cloud centralized services

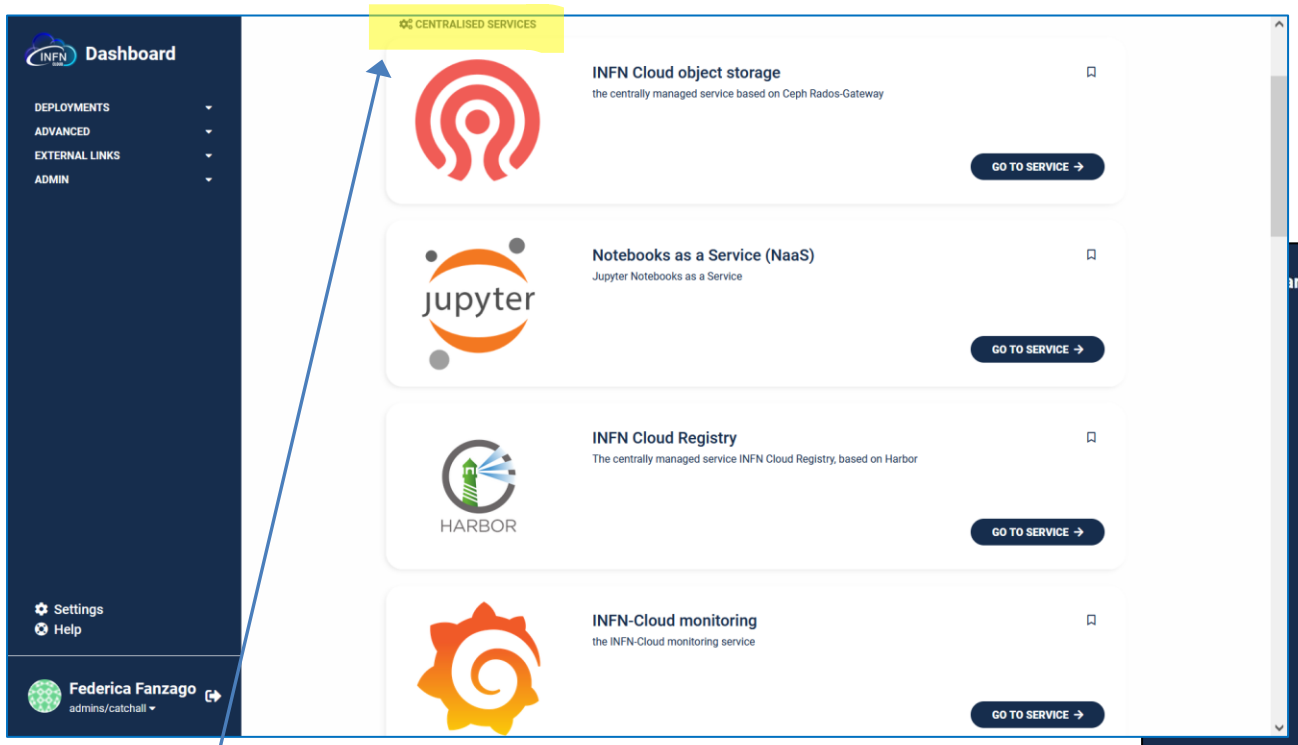
Corso di formazione per neoassunti nelle attività di computing – INFN DATACloud
17 Luglio 2024

Federica Fanzago – INFN Padova

Outline

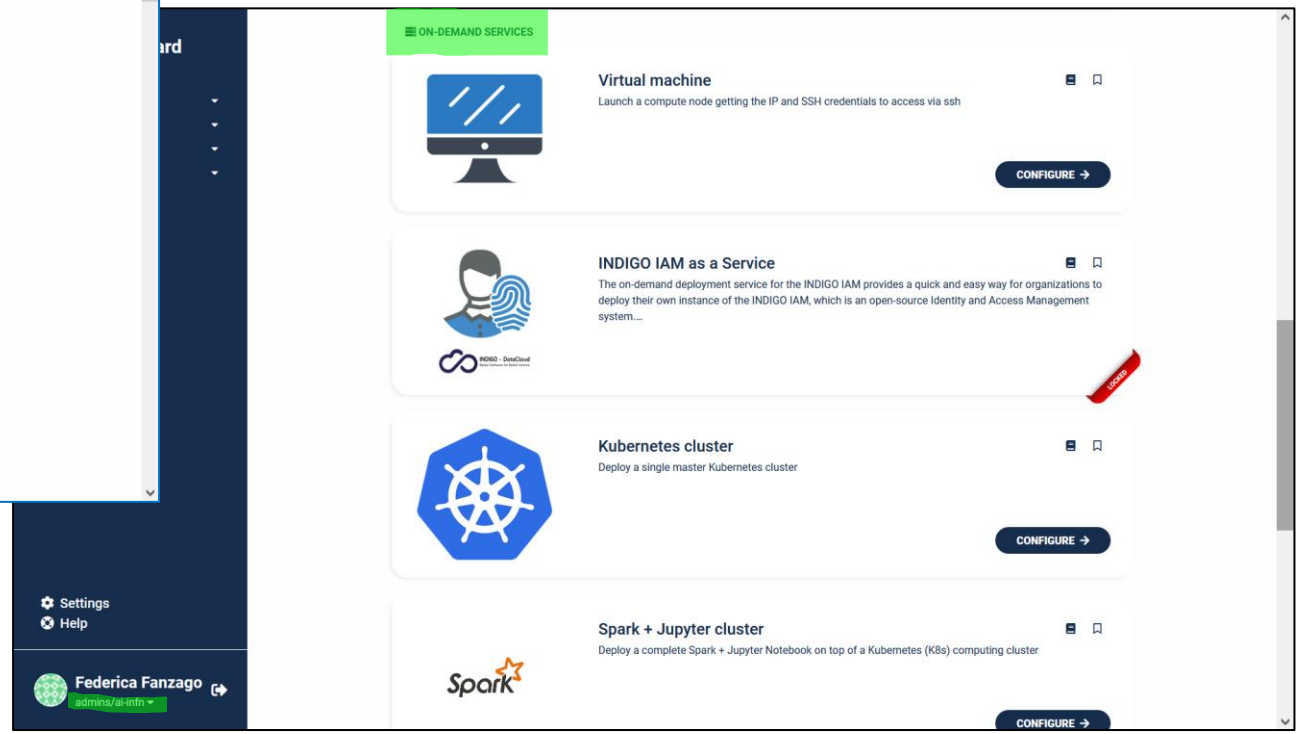
- Short description
  - INFN Cloud object storage (Ceph-RGW)
  - INFN Cloud Registry (Harbor)
  - Notebook as a Service (NaaS)

Available services



Same view for all the registered INFN Cloud users

View depends on user's iam group



Centrally managed services

- Fully-managed
  - their installation, configuration, update and security are under «INFN Cloud» responsability (dedicated team).
- Installed on the INFN Cloud backbone, CNAF and Bari.
- Available and ready to use for all the registered users in INFN Cloud
  - Iam groups added by default: users/catchall, users/s3 and users/naas
  - the system admin nomination «nomina ad amministratore INFN Cloud (quale utente amministratore)» isn't required

INFN Cloud object storage

- https://s3webui.cloud.infn.it
- https://guides.cloud.infn.it/docs/users-guides/en/latest/users_guides/centralised/objectstorage.html
- INFN Cloud provides object storage as centrally managed service. Ceph Storage Cluster is the backend
  - Migrated from Openstack Swift with Minio-gw in June 2024
- This storage is the persistent one for INFN Cloud.
- The object storage is replicated in the two sites of backbone, CNAF and Bari (no distributed cluster)
  - It guarantees the redundancy of data and disaster recovery (replica via sync)
  - No data backup

RADOS Gateway (RGW)

- INFN Cloud is using the Ceph Rados Gateway (RESTful gateway for object storage)
  - It provides access via S3 compatible api
- There are three instances of RGW in each site (High availability)
- Only authorized users can access the storage service (authentication and authorization via IAM) .
- A web interface to access the storage, create buckets and manage files has been developed using S3 protocol and OAuth2/OpenID Connect

Login to webui via OpenID

Persistent data with object storage

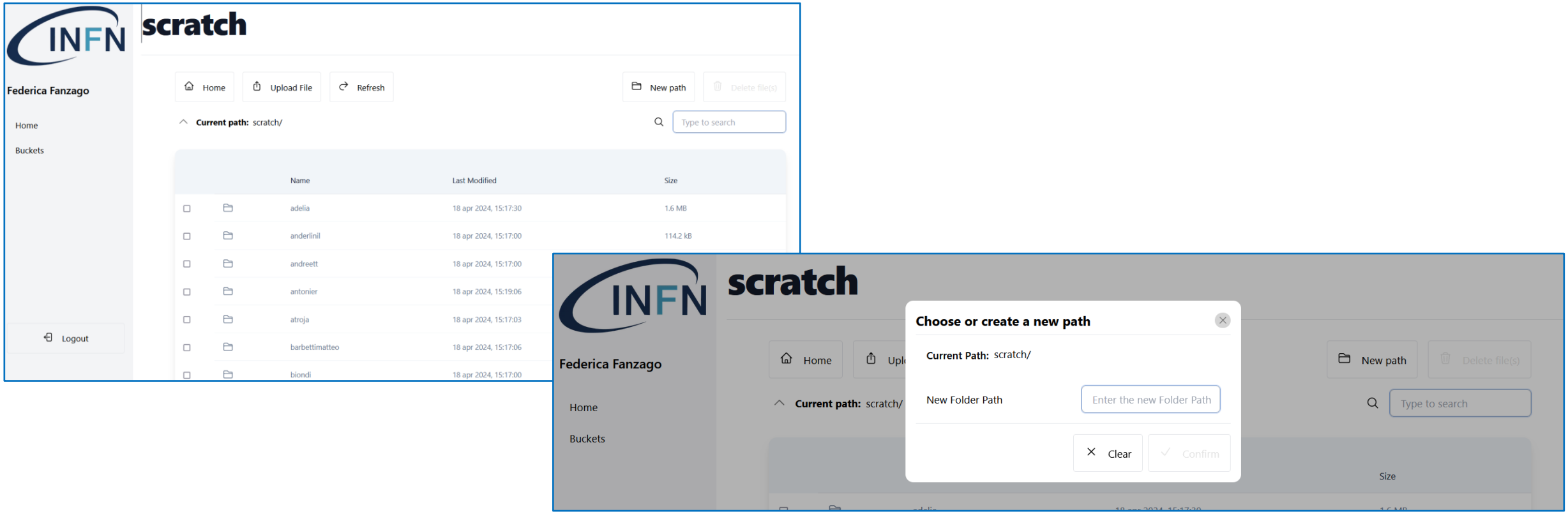
- The storage configuration allows users to have a personal bucket (directory) and an area under /scratch labeled as their INFN AAI username.
- The quota for each user is 200GB.
- Data stored in the personal bucket are private, data under /scratch are visible and downloadable by all.
- The storage is integrated with the Open Policy Agent (OPA) that enforces bucket policies rules.

Personal bucket

- The personal bucket can by create by user accessing the Ceph RGW webui otherwise it is automatically created the first time user instantiates a Jupyter notebook via INFN Cloud.

Personal area under scratch

- The area under /scratch has to be created by the user



S3 api and user configuration backup

- Prerequisites to get the IAM token with audience «object»:
  - Oidc-agent installed
  - Oidc profile configured
- S3 api to access data via code (boto3 api + Secure Token Service)
- A system for the backup offline for users to save files/configurations via Rclone is work in progress…
- Via Rclone the object storage is accesible as local file system
  - rclone ls profile:/<bucket>

```
s3client = boto3.client('s3',
    aws_access_key_id = response['Credentials']['AccessKeyId'],
    aws_secret_access_key = response['Credentials']['SecretAccessKey'],
    aws_session_token = response['Credentials']['SessionToken'],
    endpoint_url="https://rgw.cloud.infn.it",
    region_name='default')
```

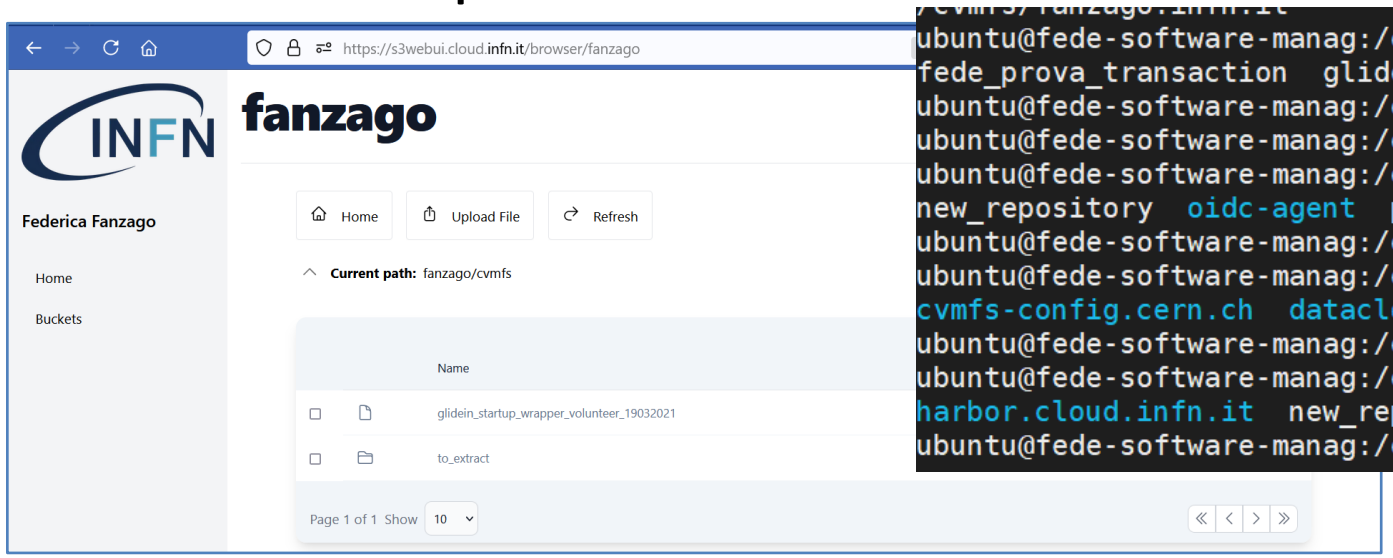
```
# Show files in a bucket

for key in s3client.list_objects(Bucket='scratch')['Contents']:
        print(key['Key'])
```

Object storage also for

- Cvmfs users repo i.e /cvmfs/username.infn.it
- Cvmfs software repo «datacloud.infn.it»
- Cvmfs contenerized software repo «unpacked.infn.it»
- Harbor repos

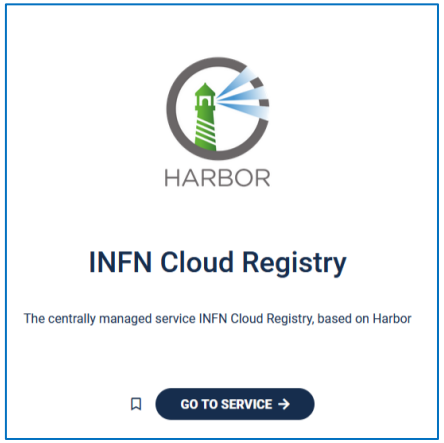
CernVM-File system (cvmfs) is a a read-only file system

Harbor container registry




```
ubuntu@fede-software-manag:/cvmfs/fanzago.infn.it$ ls
fede_prova_transaction  glidein_startup_wrapper_volunteer_19032021  mysoftware  new_
ubuntu@fede-software-manag:/cvmfs/fanzago.infn.it$ cd ..
ubuntu@fede-software-manag:/cvmfs$ cd datacloud.infn.it
ubuntu@fede-software-manag:/cvmfs/datacloud.infn.it$ ls
new_repository  oidc-agent  prova-new-cvmfs-keysetup.txt  repo  sts-wire  test
ubuntu@fede-software-manag:/cvmfs/datacloud.infn.it$ cd ..
ubuntu@fede-software-manag:/cvmfs$ ls
cvmfs-config.cern.ch  datacloud.infn.it  fanzago.infn.it
ubuntu@fede-software-manag:/cvmfs$ cd unpacked.infn.it
ubuntu@fede-software-manag:/cvmfs/unpacked.infn.it$ ls
harbor.cloud.infn.it  new_repository
ubuntu@fede-software-manag:/cvmfs/unpacked.infn.it$
```

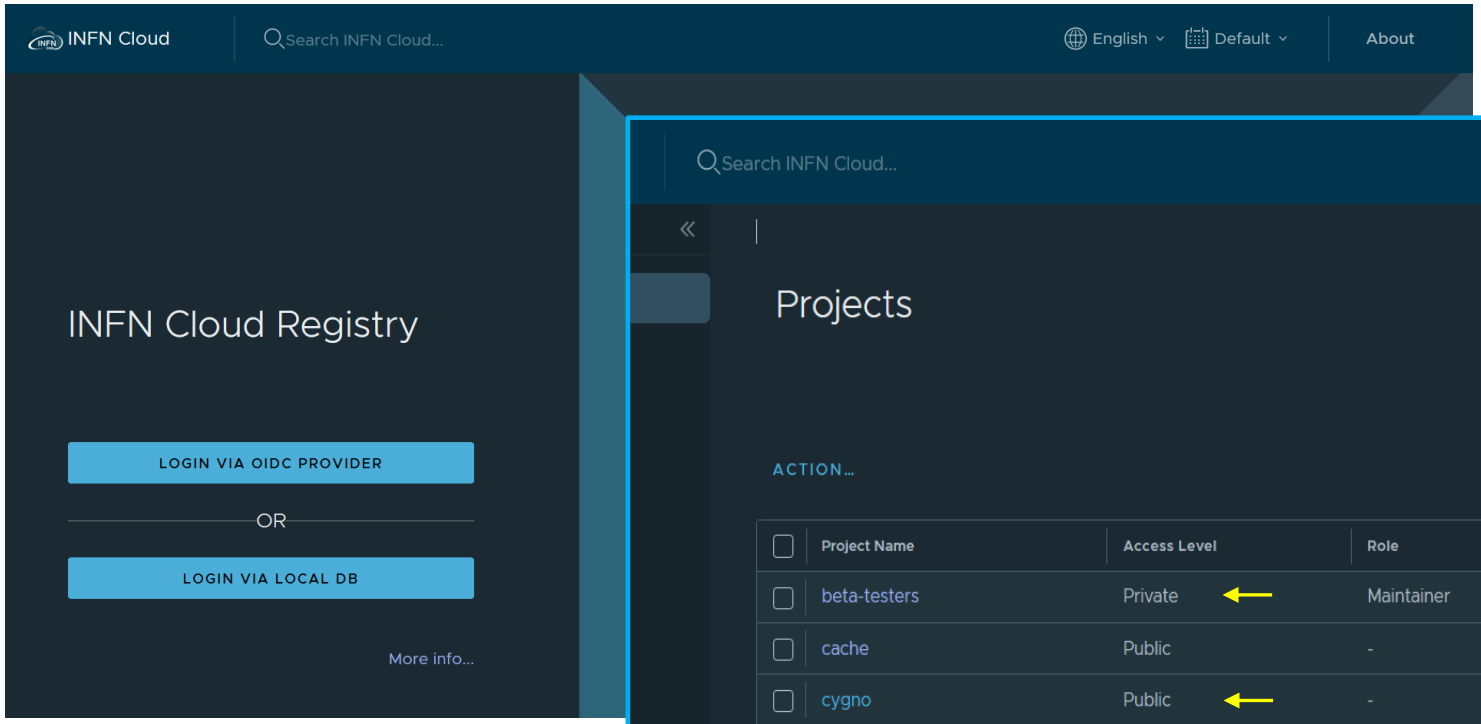
INFN Cloud Registry

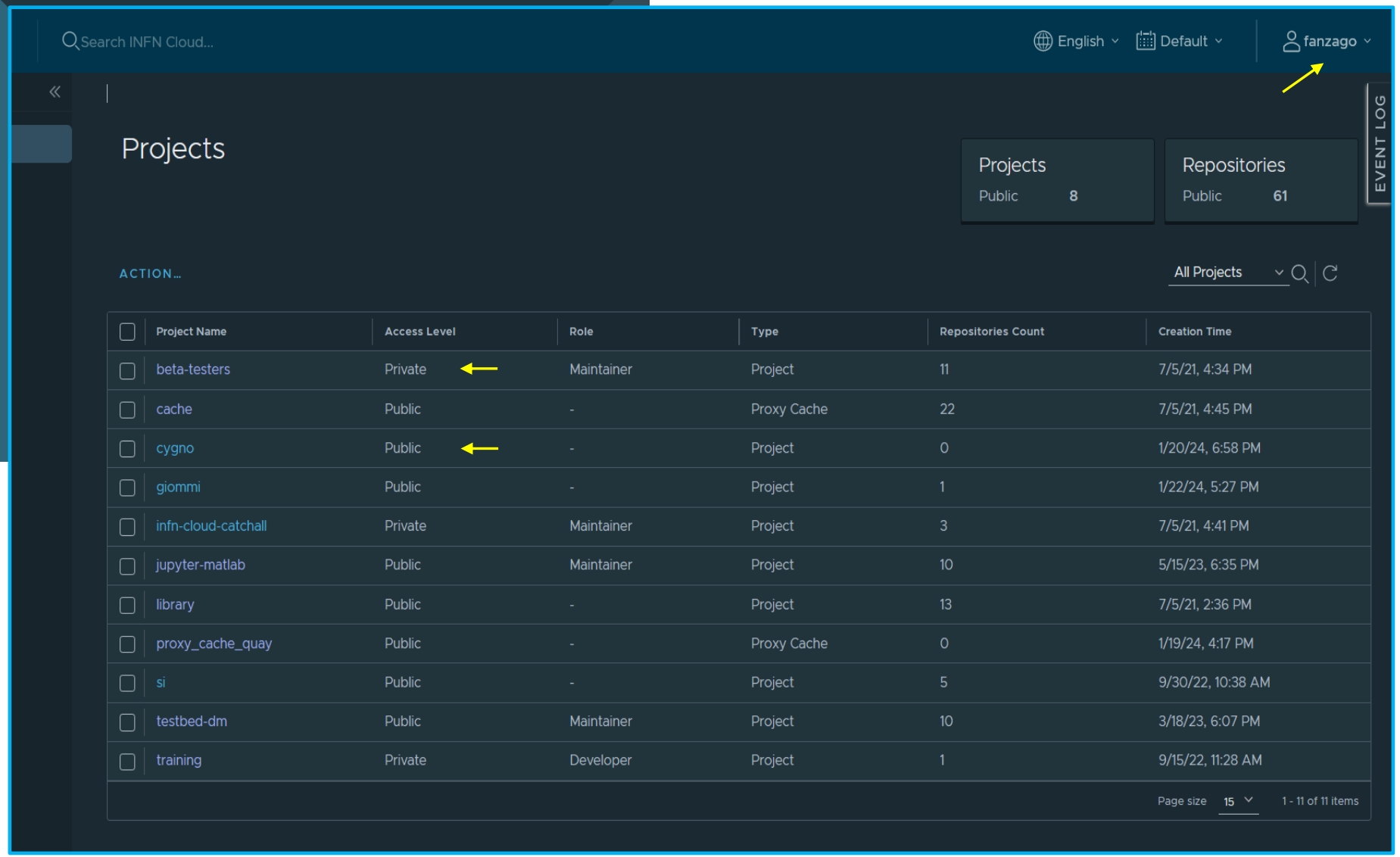


- https://harbor.cloud.infn.it
- https://guides.cloud.infn.it/docs/users-guides/en/latest/users_guides/centralised/harbor.html
- The INFN Cloud Registry for container images and helm charts, based on Harbor software.
  - «Project Harbor is an an open-source trusted cloud native registry project that stores, signs, and scans content»
- Harbor extends some functionalities of Docker Hub (i.e checks vulnerabilities) and overcomes its pull rate limit (i.e replica, proxy-cache).

Harbor: access

- Installed on the backbone (CNAF and Bari), high availability
- Users can access the service via oidc provider, using INFN-AAI credentials.
  - Users can see projects associated with their iam groups (if not visible, projects have to be created by Harbor admin).
- Two types of project supported:
  - Public: any user can pull images from this project
  - Private: only users who are members of the project can pull images

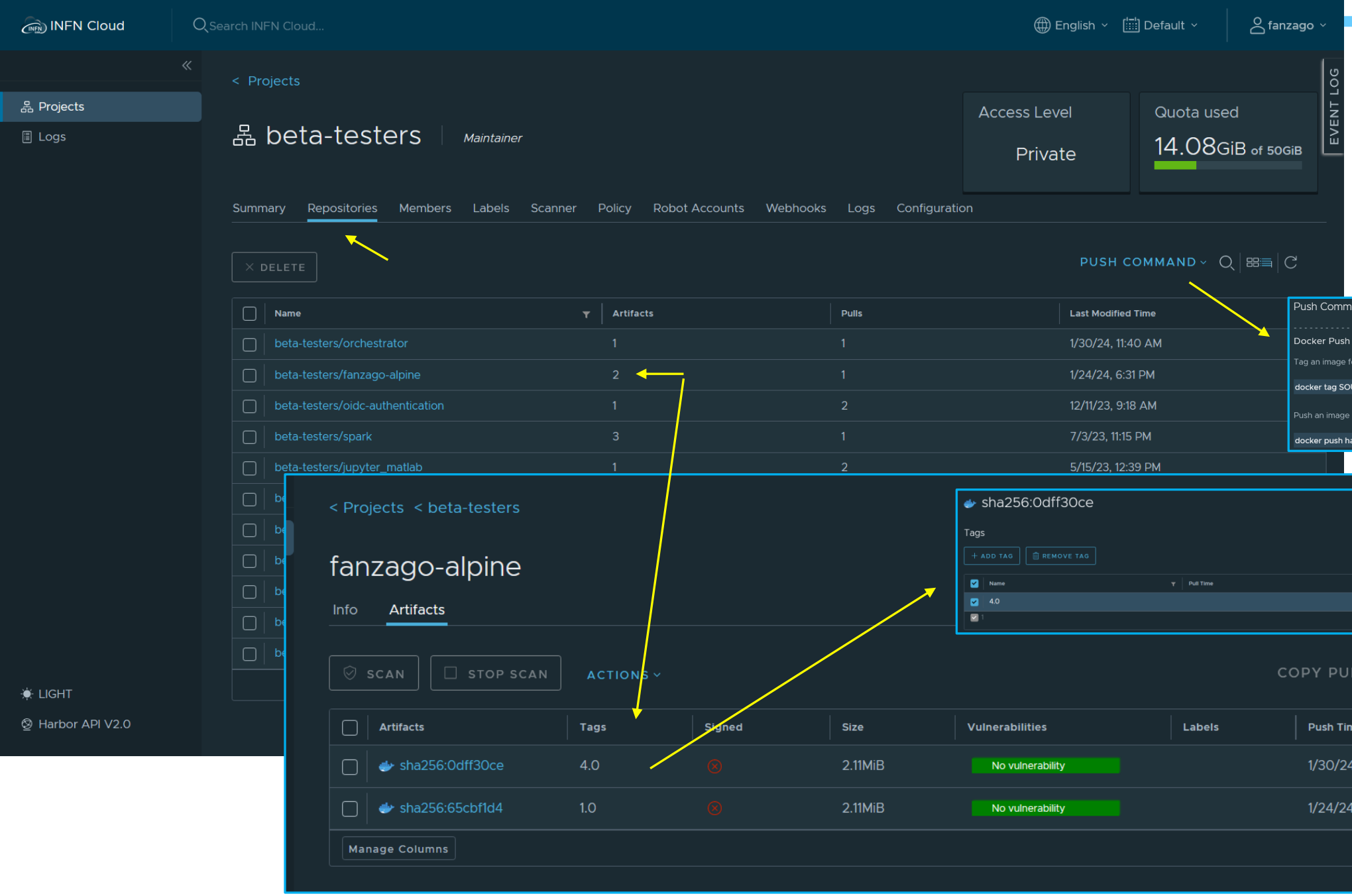




Users have different abilities depending on the role they have in a project.

Limited guest, Guest, Developer, Master, Project Admin

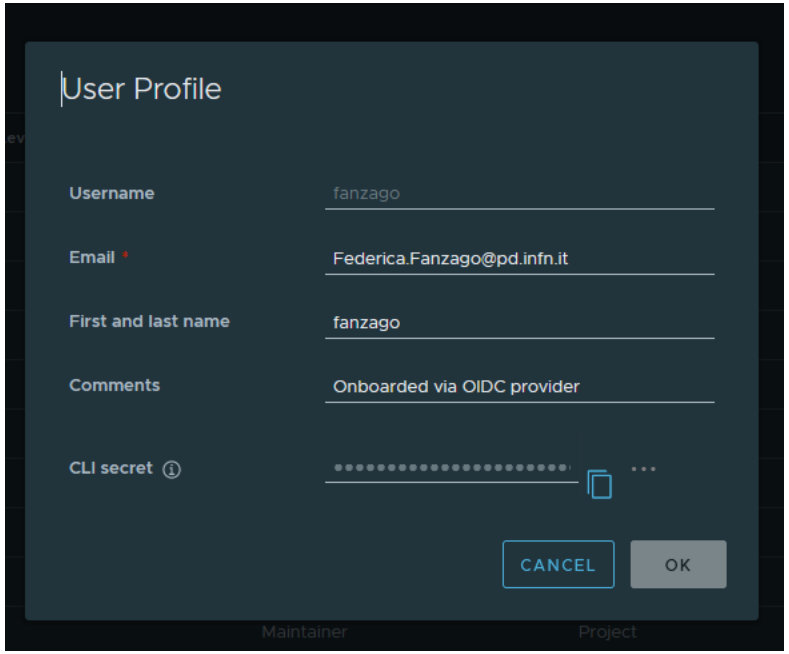
Federica Fanzago INFN Padova



INFN CLOUD
Istituto Nazionale di Fisica Nucleare

Harbor: push and pull

- How to push and pull of files to/from repos
  - Needed a terminal with docker up and running
  - Login into the service via docker command using username and password (CLI secret) shown clicking on username and then “user profile”

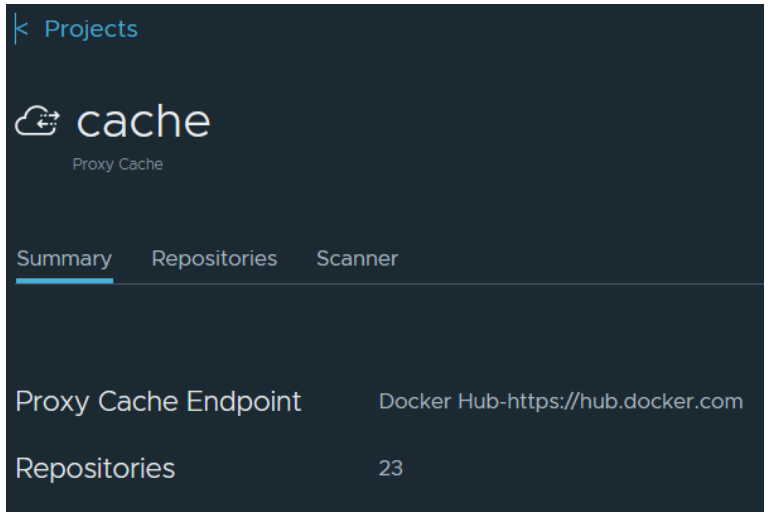



```
## login
$ docker login harbor.cloud.infn.it
## tag the image
$ docker tag SOURCE_IMAGE[:TAG] harbor.cloud.infn.it/project/REPOSITORY[:TAG]
## push the image
$ docker push harbor.cloud.infn.it/project/REPOSITORY[:TAG]
## pull the image
$ docker pull harbor.cloud.infn.it/project/REPOSITORY[:TAG]
```

Harbor: other details

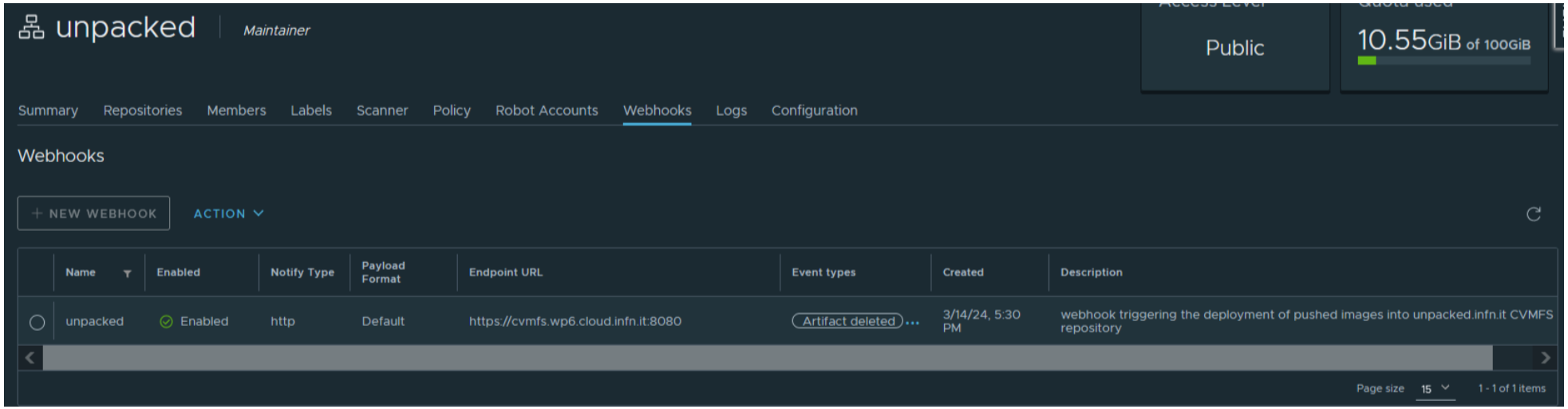
- Images are stored in the object storage of INFN Cloud, a quota can be guarantee for projects. The bucket is unique.
- Proxy cache configured:
  - when a pull request comes to a proxy cache project, if the image is not cached, Harbor pulls the image from the target registry and serves the pull command as if it is a local image from the proxy cache project

$ harbor.cloud.infn.it/cache/<docker_repo_name>/<image>:<tag>



Harbor and cvmfs

- In pre-production:
  - CernVM-File system (cvmfs) is a a read-only file system designed to deliver scientific software onto virtual machines and physical worker nodes in a fast, scalable, and reliable way (LHC experiments)
  - Can be used to distribute container images pushed in Harbor (cvmfs webhook) under /cmvfs/unpacked.infn.it/harbor.cloud.infn.it/unpacked area

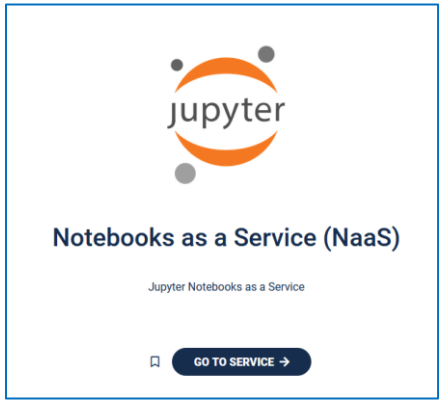


Harbor and apptainer

- Images are readily available to run with apptainer (formerly singularity)
  - Singularity containers are preferred when running applications in HPC systems. Singularity containers can be run without sudo.
- Push in “cvmfs harbor” area

```
## in the terminal where docker is installed and running
$ docker login harbor.cloud.infn.it
## tag the image
$ docker tag my-image:1.0  harbor.cloud.infn.it/unpacked/my-image:1.0
## push the image
$ docker push  harbor.cloud.infn.it/unpacked/my-image:1.0
## In the cvmfs client where apptainer is installed
$ apptainer exec '/cvmfs/unpacked.infn.it/harbor.cloud.infn.it/unpacked/my-image:1.0' /bin/sh
```

Notebook as a service (NaaS)



- https://hub.cloud.infn.it
- https://guides.cloud.infn.it/docs/users-guides/en/latest/users_guides/centralised/naas.html
- It is installed on the backbone (CNAF and Bari), high availability
  - JupyterHub server
- After login through IAM, user starts the “personal” notebook using the docker image made available for the Jupyter Notebook
  - INFN-Cloud base image: harbor.cloud.infn.it/datacloud-templates/jaas_user_containers:1.2.0
- Automatic mount of S3 RGW object storage user area (cloud-storage)

Jupyter notebook

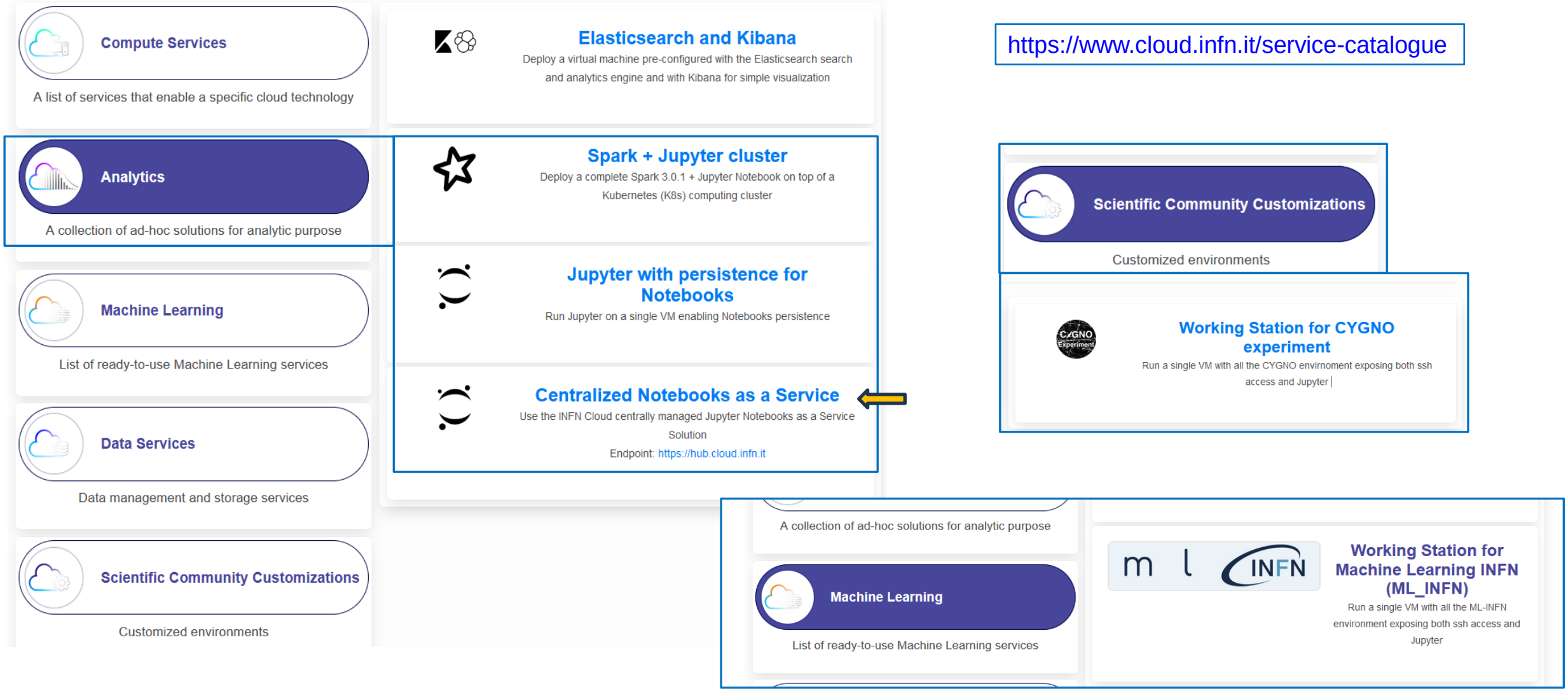
- Project started in 2014 by the Jupyter project organization https://jupyter.org
- Open-source web application that provides an interactive environment to create documents called notebooks (.ipynb)
  - A notebook can contain formatted text, equations, images and code
  - The Jupyter notebook provides 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

- Analysis facility
- AI_INFN
- CYGNO

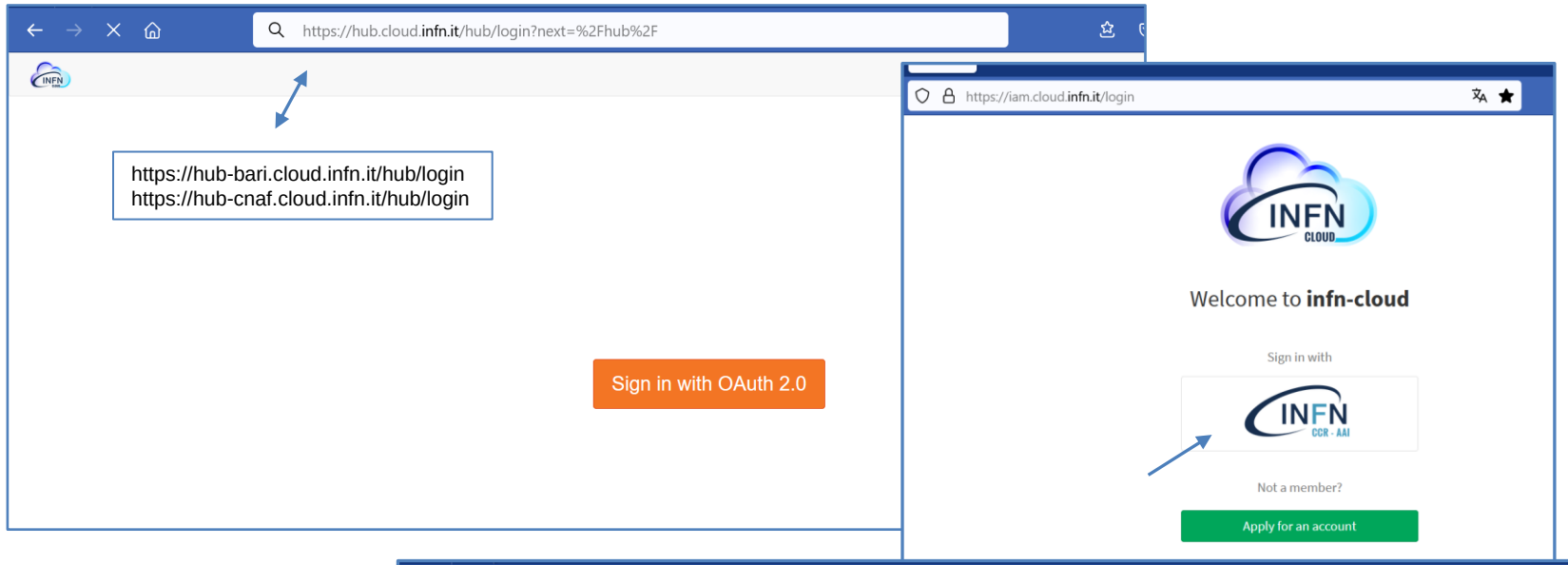
JupyterLab and JupyterHub

- 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 on shared resources.
- Some INFN Cloud services are implementing Jupyter (hub + notebook (lab))

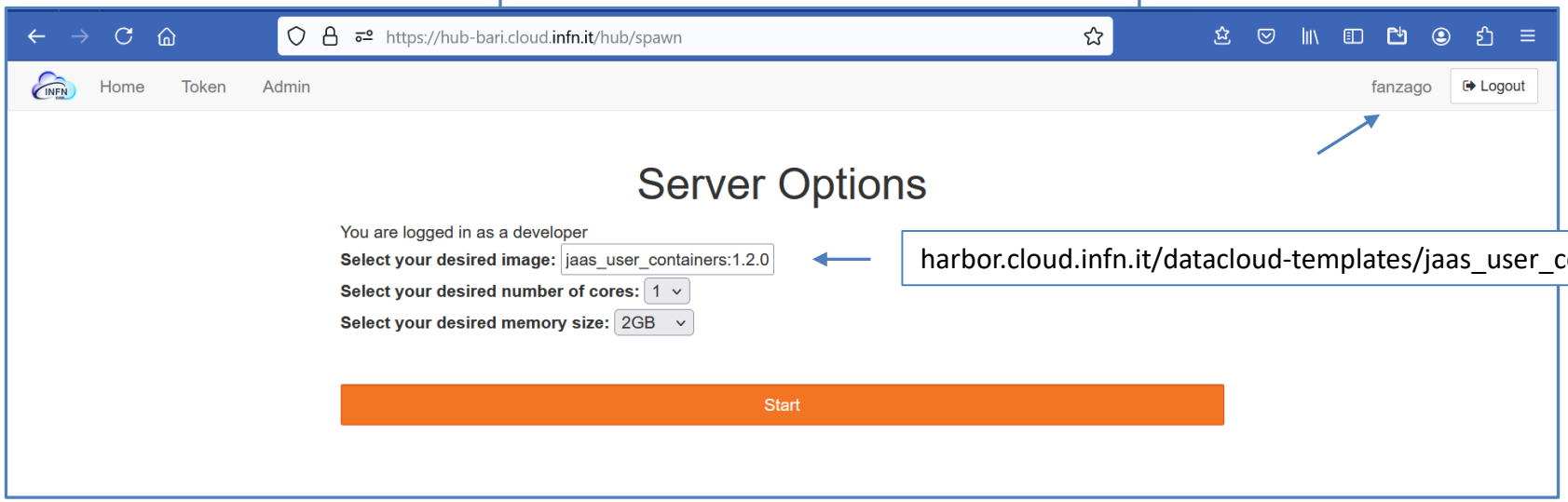
INFN Cloud services based on Jupyter



NaaS Hub Interface



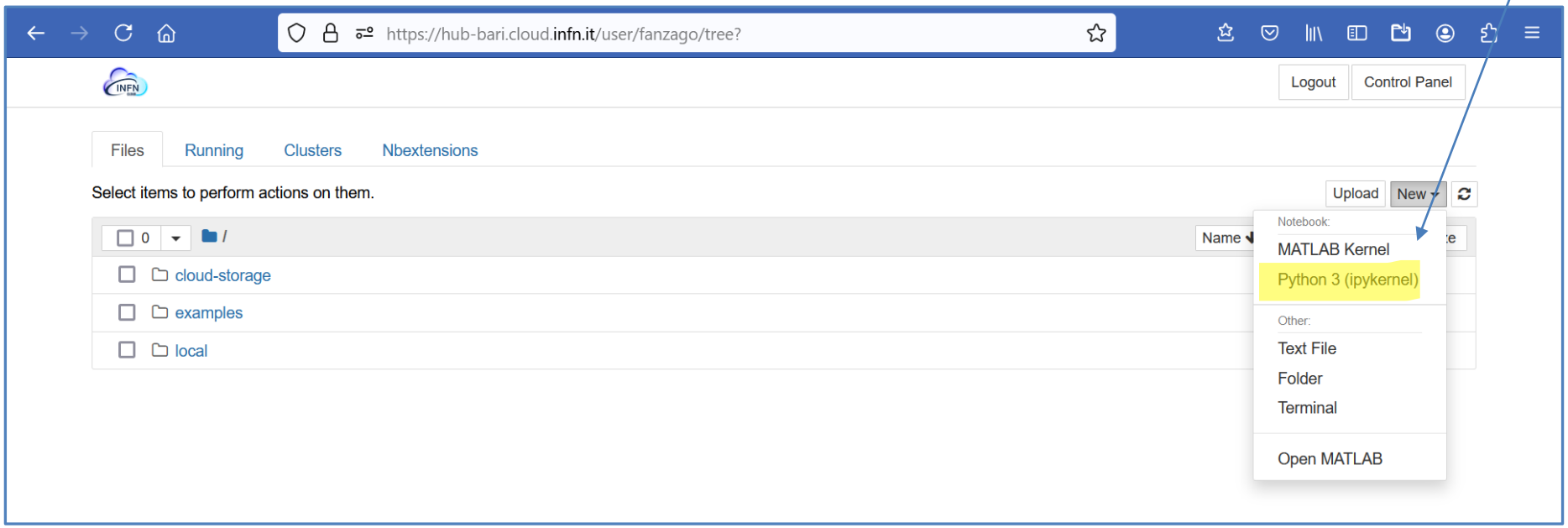
https://hub.cloud.infn.it



harbor.cloud.infn.it/datacloud-templates/jaas_user_containers:1.2.0

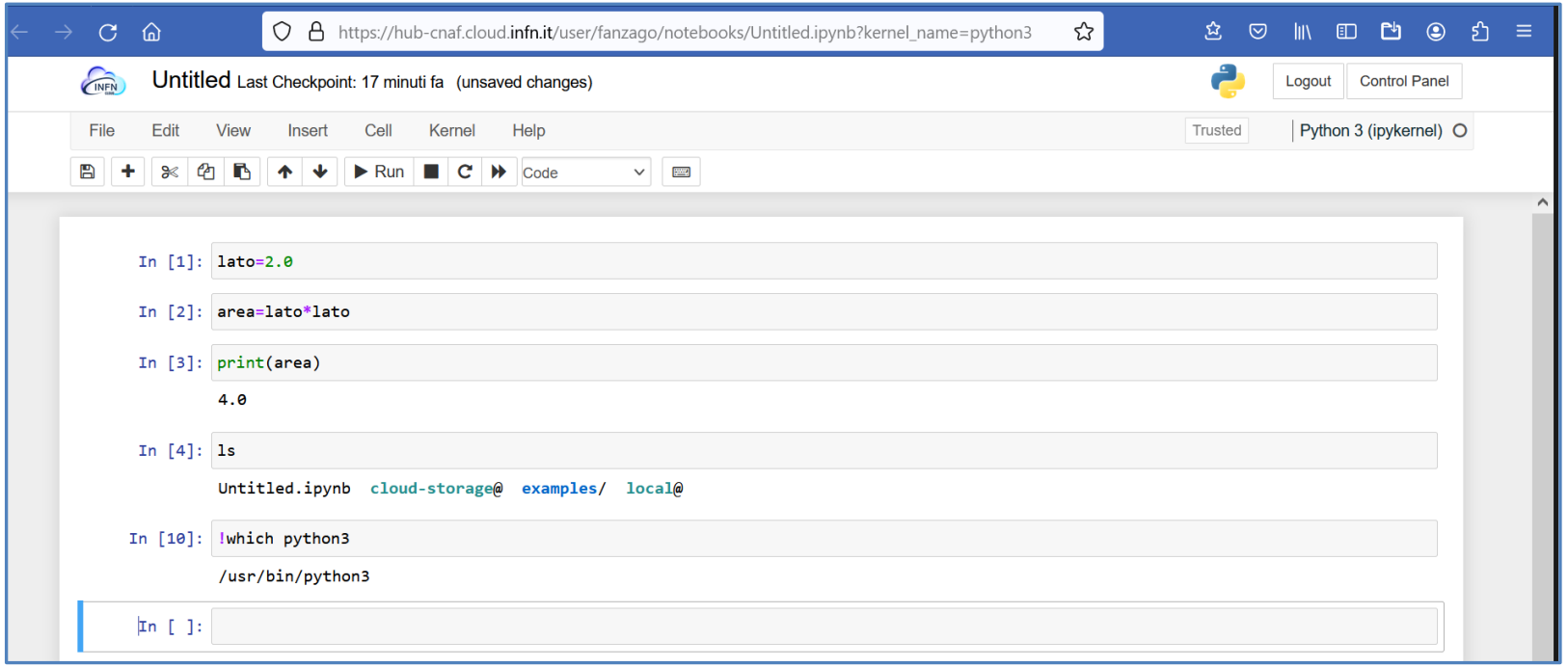
Jupyter notebook interface

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



Jupyter notebook interface

- 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].



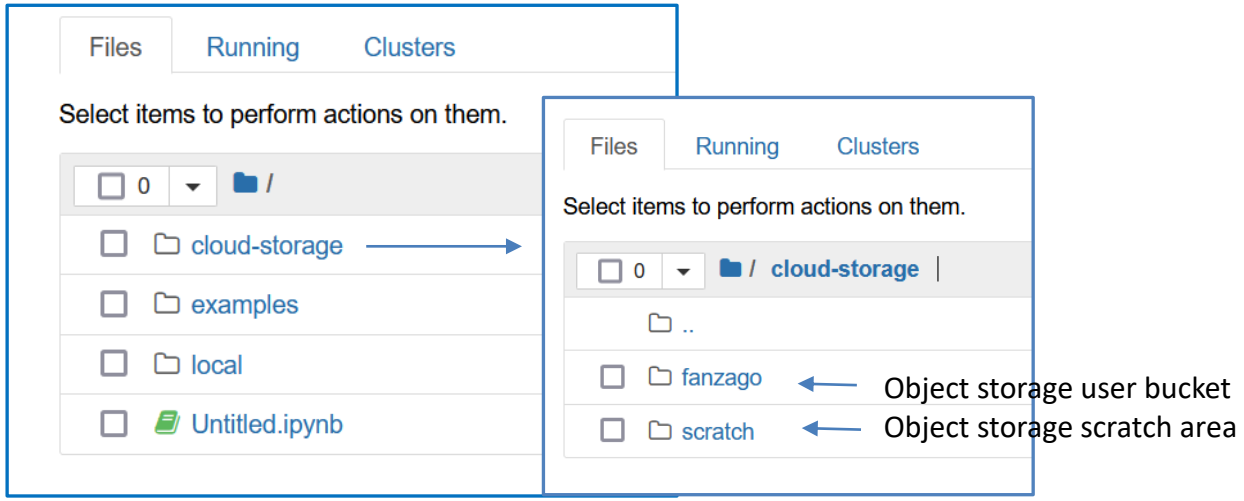
Inside cells, select code:
Shift + enter - cell execution
ESC a,b - add cells
ESC dd - delete cell

%lsmagic
%run <script.py>
%load <script.py>
%who - list of python variables
%pinfo <var> - description of variable
%env - environment variables

Pip freeze - python modules installed
Pip install <module>

Where to store notebook documents?

- Saving document: the default path is the local “jupyter-workspace” directory of the system where the notebook server is running
  - It runs in a container: data are lost if the container is removed
- Need to have a “permanent” space where to store them and guarantee persistence of data.
  - In INFN Cloud the object storage meets this request



Local = directory «local» in the server (bari or cnaf)

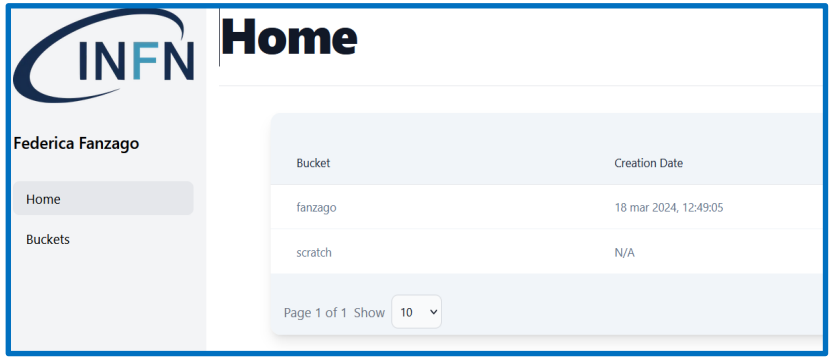
```
# /bin/bash
13
root@jupyter-fanzago:/workarea# ls -ltr
total 0
drwxr-xr-x. 2 root root 58 Mar 28  2023 examples
lrwxrwxrwx. 1 root root 14 Mar 28  2023 local -> /opt/user_data
lrwxrwxrwx. 1 root root  3 Mar 28  2023 cloud-storage -> /s3
root@jupyter-fanzago:/workarea# ls
Untitled.ipynb  cloud-storage  examples  local
root@jupyter-fanzago:/workarea# l
```

Persistent directory for data

- All the services running notebooks mount these areas, so they are visible in the file system as posix directories.
  - cloud-storage dir
- The personal bucket can by create by user accessing the Ceph-RGW webui otherwise it is automatically created the first time user instantiates a notebook.

Using sts-wire, fuse and Rclone

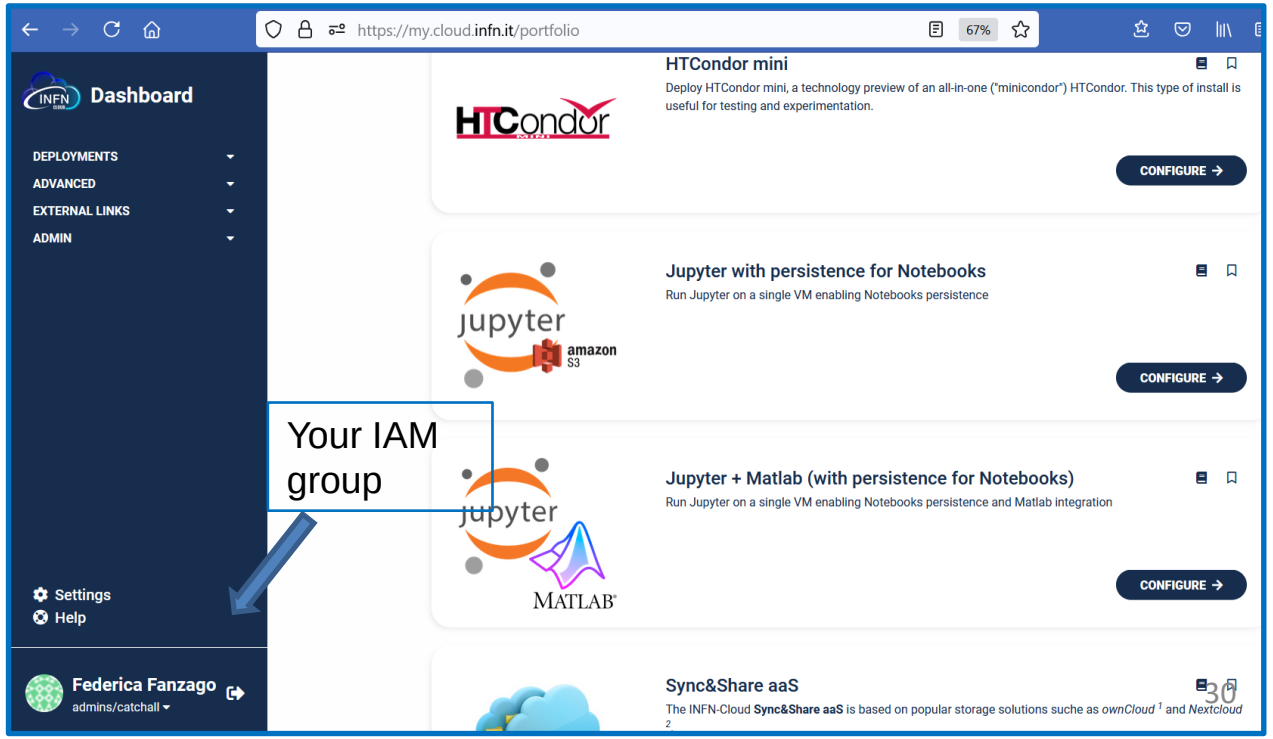
`



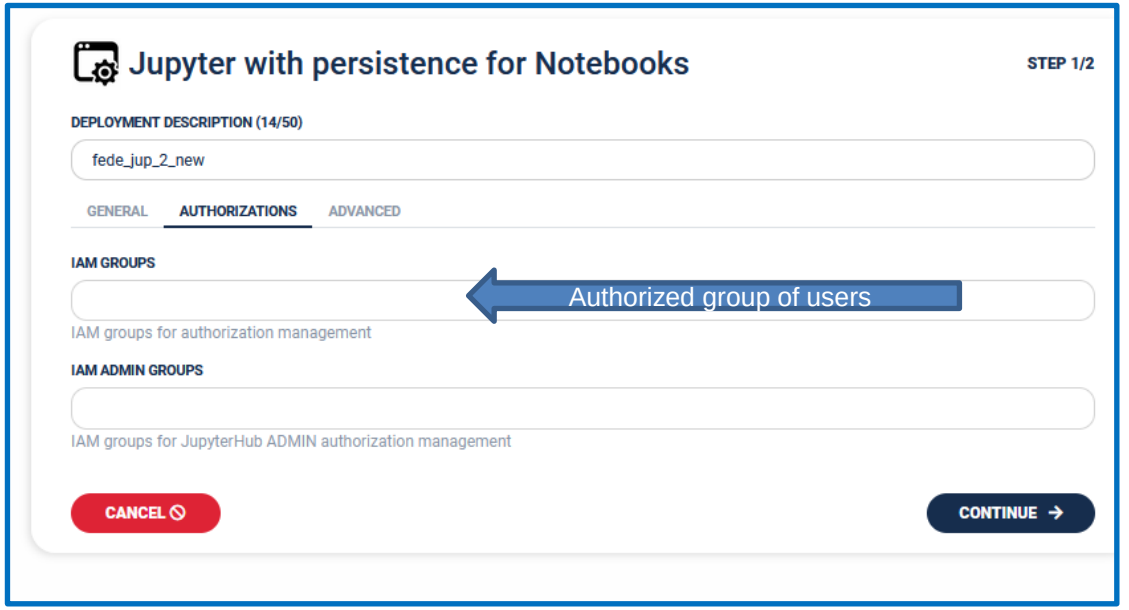
Comparing with “Jupyter with persistence” deployment (self-managed service)

- 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

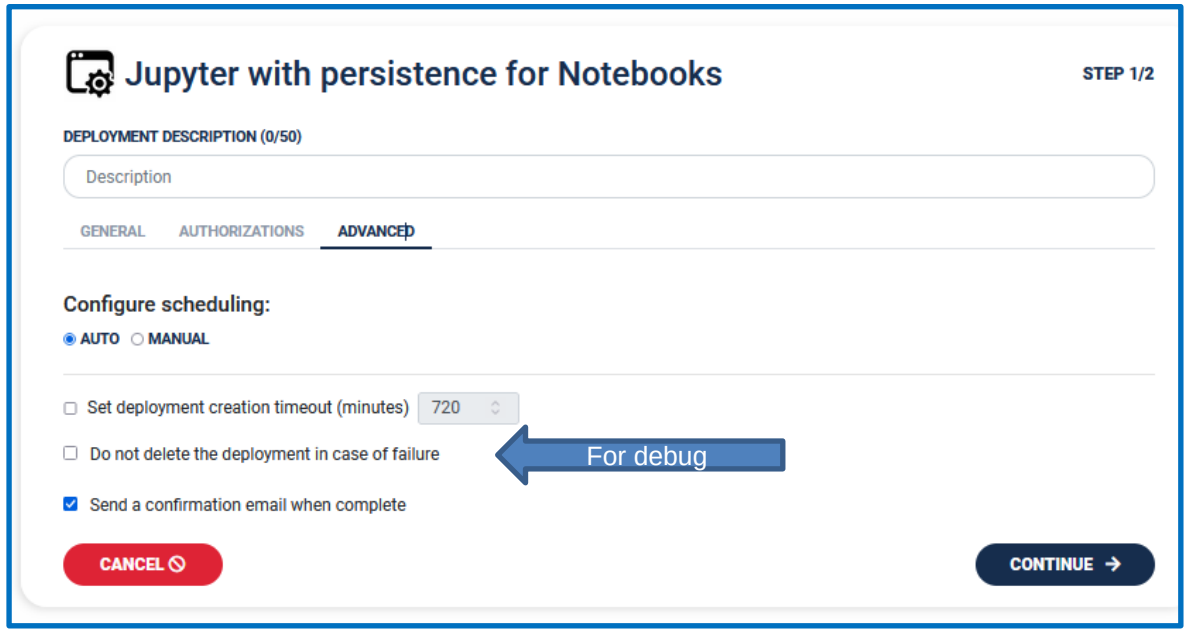


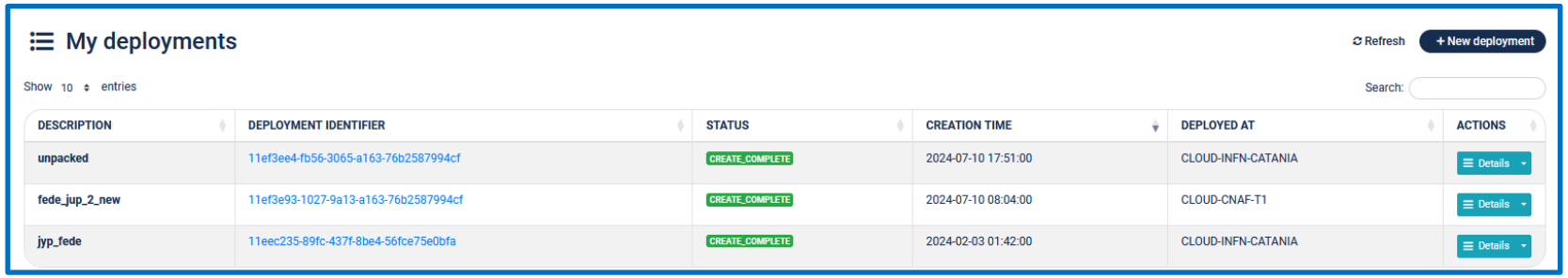
Filling the form…



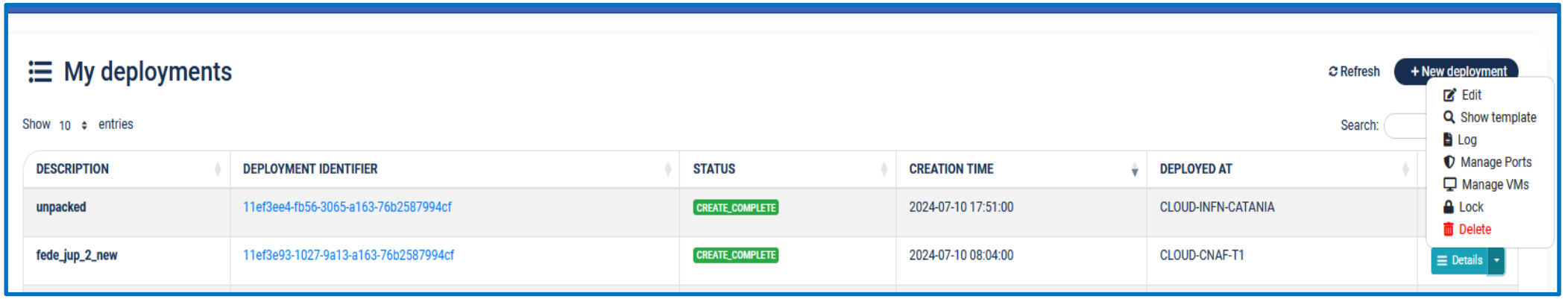
User guide:
https://guides.cloud.infn.it/docs/users-guides/en/latest/users_guides/sysadmin/compute/jh_with_persistence.html

…then submit the deployment





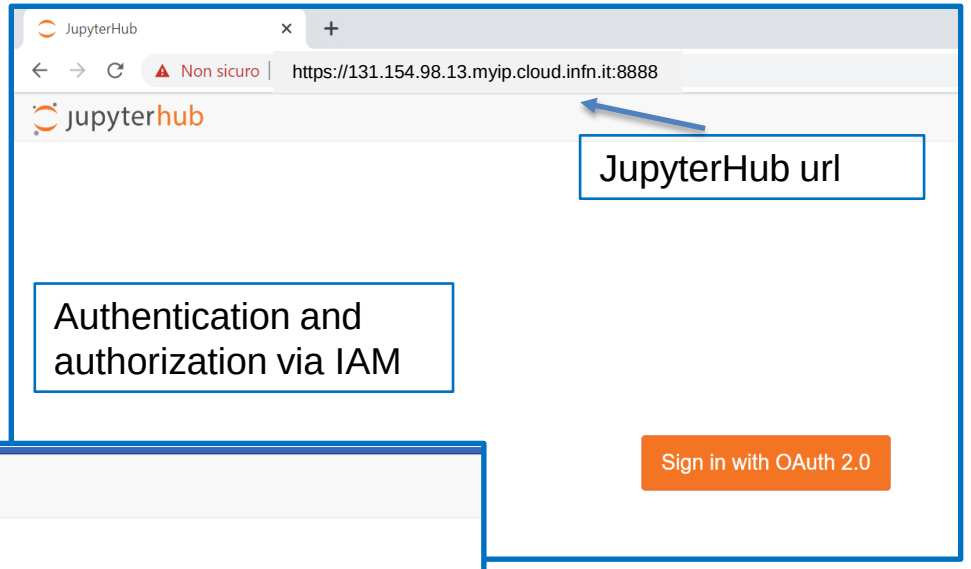
Deployment details

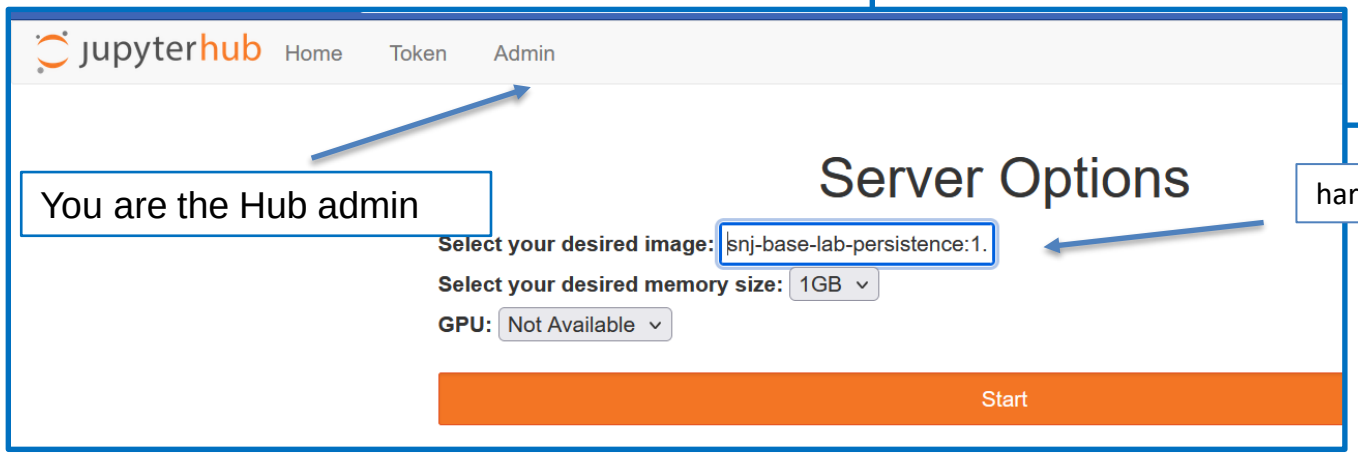


Remember your ssh key to access the Vm

JupyterHub url

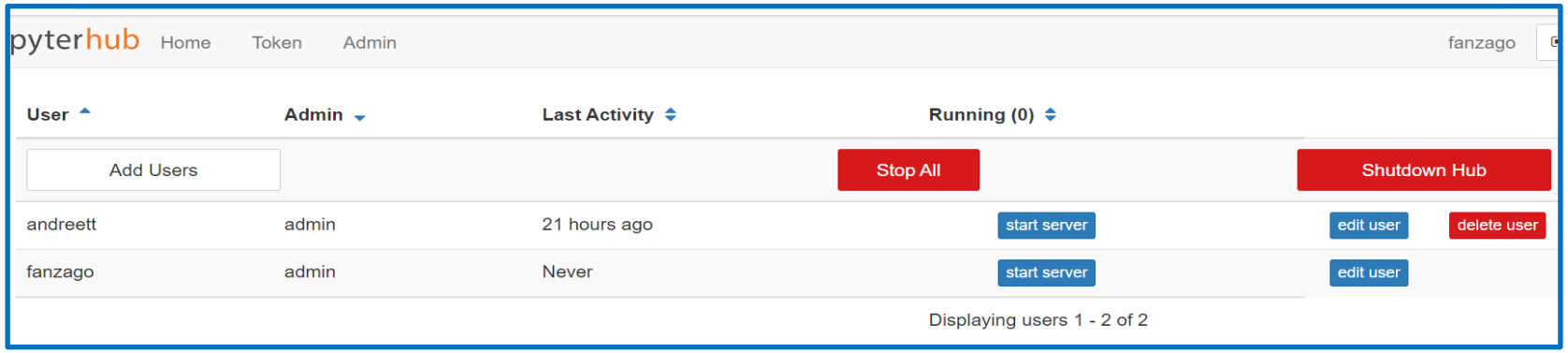
Your JupyterHub





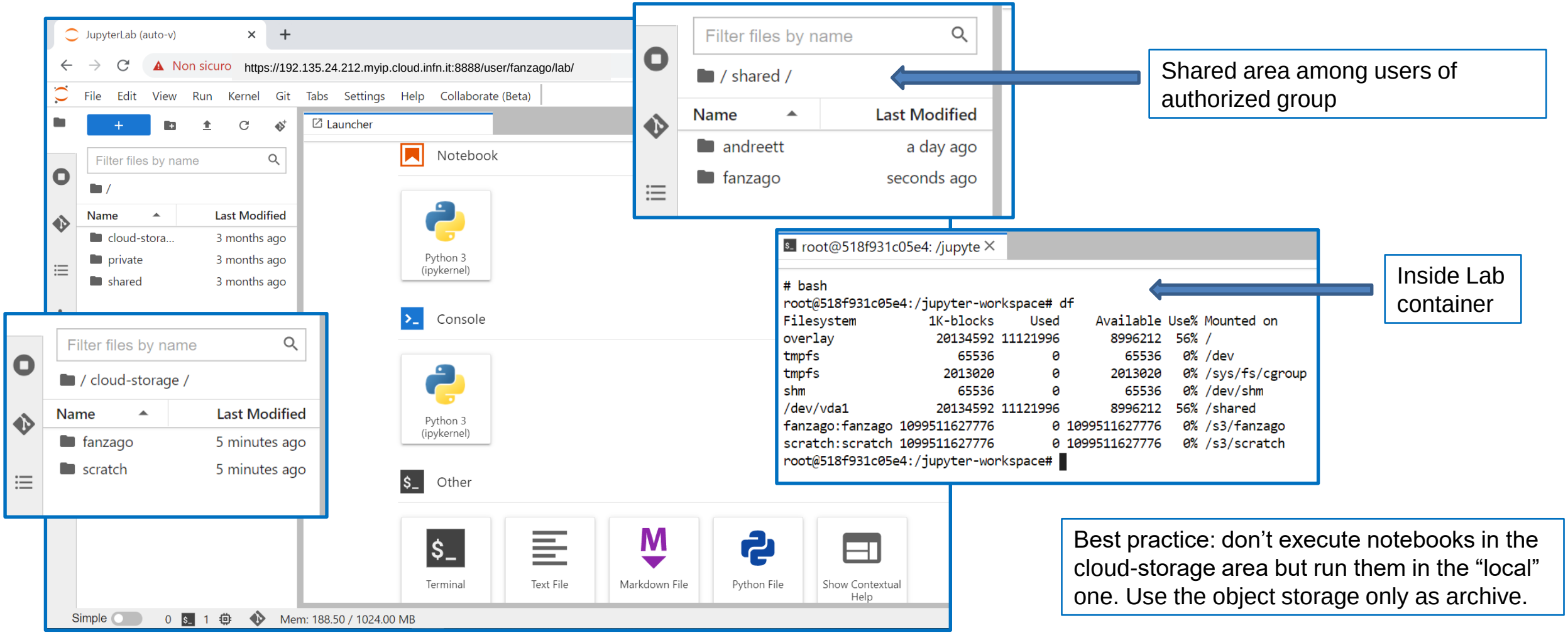
harbor.cloud.infn.it/datacloud-templates/snj-base-lab-persistence:1.2.0

Image provided to start the JupyterLab server. You can use your private one built from the default one



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.

Containers in the VM

>> ssh -i <key> <ssh_account>@node_ip

Container Lab

```
root@vnode-0:~# docker ps
CONTAINER ID   IMAGE                                                                  COMMAND                  CREATED      STATUS      PORTS
                                                                                       NAMES
a3c03a94c48c   harbor.cloud.infn.it/datacloud-templates/snj-base-lab-persistence:1.2.0   "tini -s -- jupyterh…"   2 days ago   Up 2 days   8889/tc
p                                                                                      jupyter-fanzago
4c0121547797   harbor.cloud.infn.it/datacloud-templates/snj-base-jhub:1.2.0             "/usr/bin/python3 /u…"   2 days ago   Up 2 days   8000/tc
p                                                                                      jupyterhub-jupyterhub-1
930e5a55145b   jupyterhub/configurable-http-proxy                                      "/bin/sh -c ' while …"   2 days ago   Up 2 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
40e1cd49b0b7   harbor.cloud.infn.it/cache/library/traefik:latest                       "/entrypoint.sh --lo…"   2 days ago   Up 2 days   0.0.0.0
:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp                       proxy
root@vnode-0:~#
```

Container Hub

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 support: https://servicedesk.cloud.infn/it