



**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)

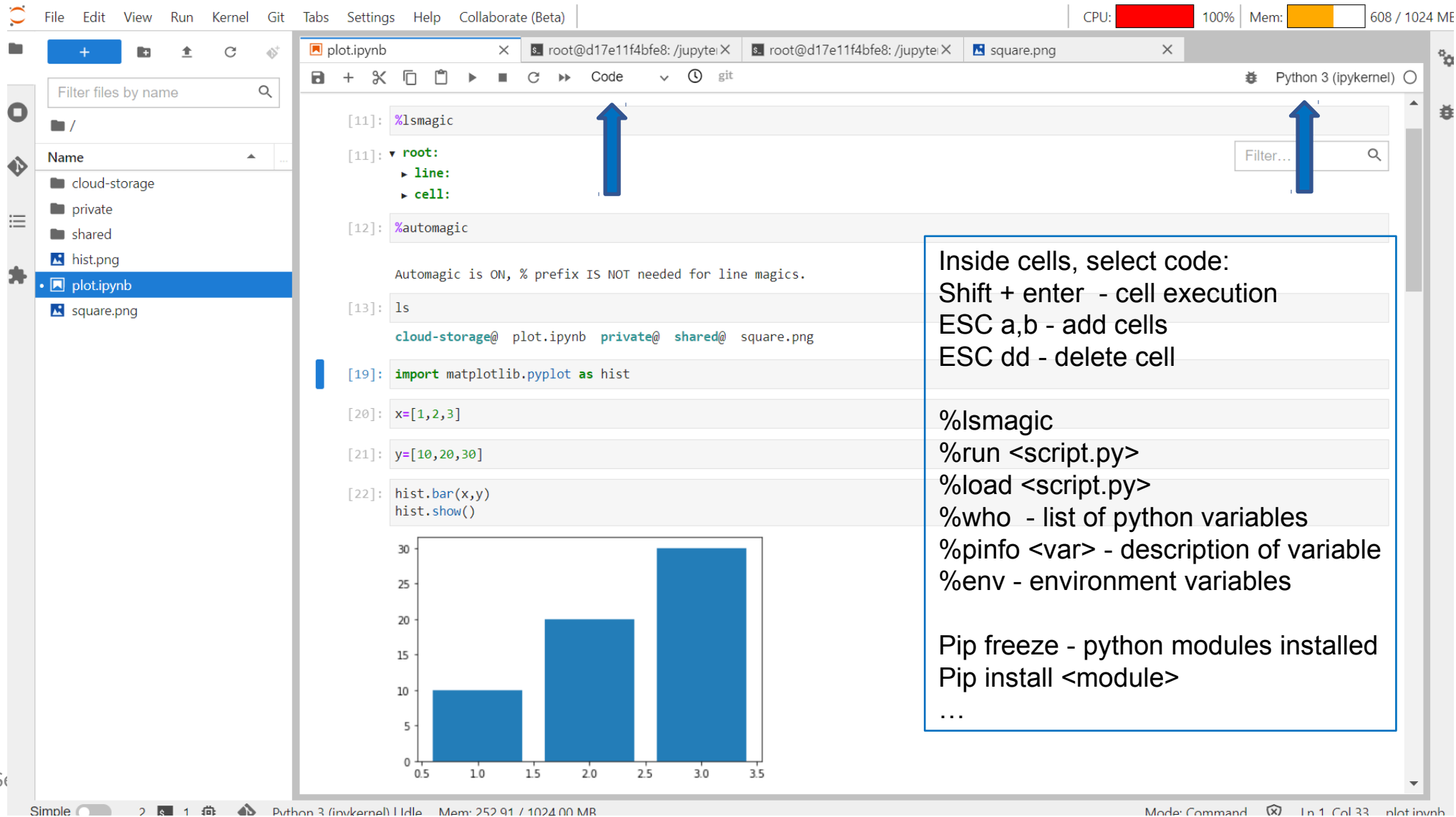
Jupyter notebook

- 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.
- The Jupyter Notebook interface allows the management of kernels and notebooks
 - `> pip3 install jupyter`
- The interactive work is done inside “cells” that can contain and execute code, commands and formatted text
 - `> jupyter notebook`
It starts the server and returns web url
- 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



The screenshot shows a Jupyter Notebook interface with the following elements:

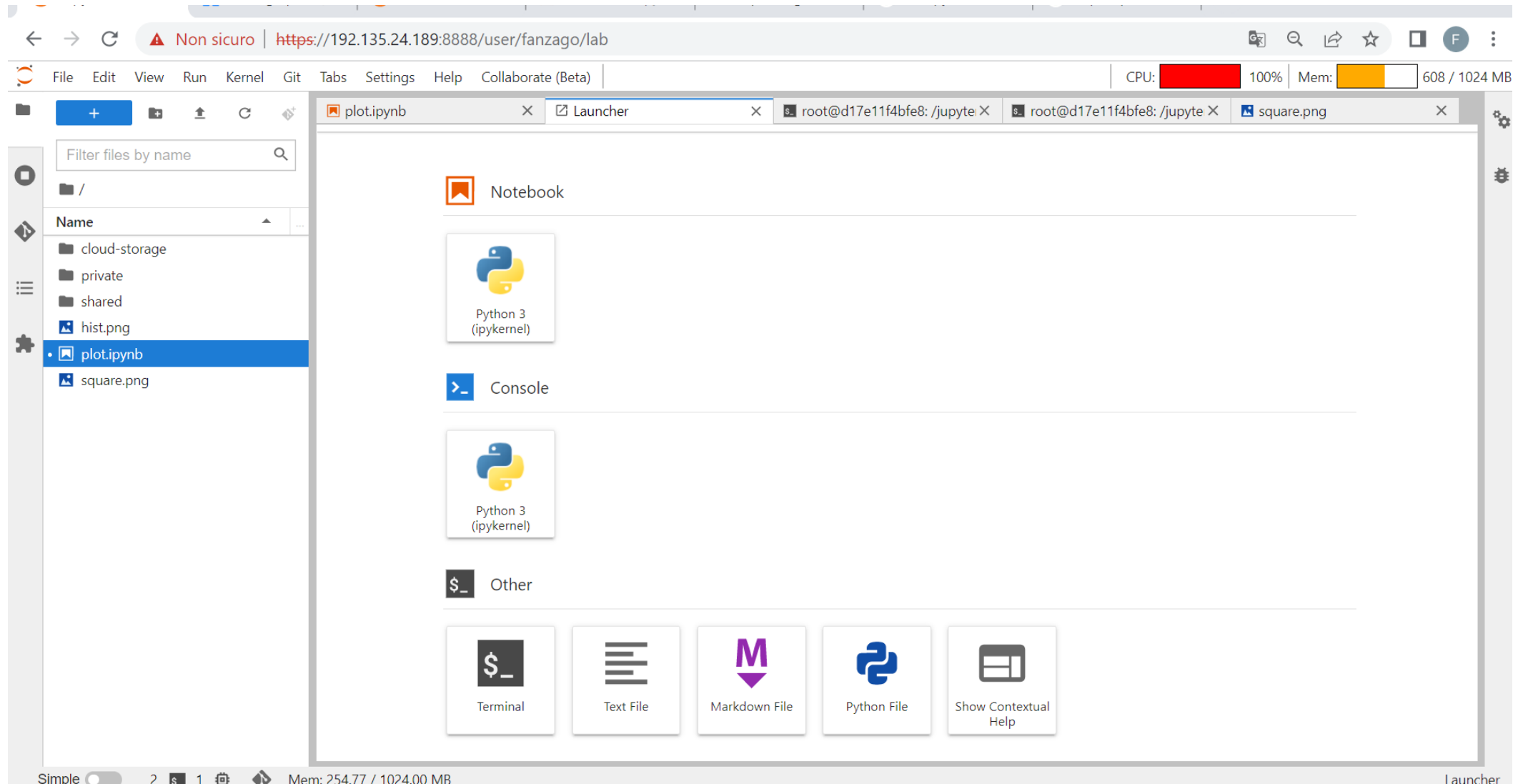
- File Explorer (Left):** Shows a file tree with folders like 'cloud-storage', 'private', 'shared' and files like 'hist.png', 'plot.ipynb', and 'square.png'.
- Code Editor (Center):** Contains several code cells:
 - Cell [11]: `%lsmagic` (highlighted with a blue arrow pointing to the cell number).
 - Cell [12]: `%automagic` (highlighted with a blue arrow pointing to the cell number).
 - Cell [13]: `ls` (output: `cloud-storage@ plot.ipynb private@ shared@ square.png`).
 - Cell [19]: `import matplotlib.pyplot as hist` (highlighted with a blue vertical bar on the left).
 - Cell [20]: `x=[1,2,3]`
 - Cell [21]: `y=[10,20,30]`
 - Cell [22]: `hist.bar(x,y)` and `hist.show()` (output: a bar chart).
- Bar Chart (Bottom):** A bar chart with x-axis values [1, 2, 3] and y-axis values [10, 20, 30].
- Text Box (Right):** A box containing the following text:
 - 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>
 - ...

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

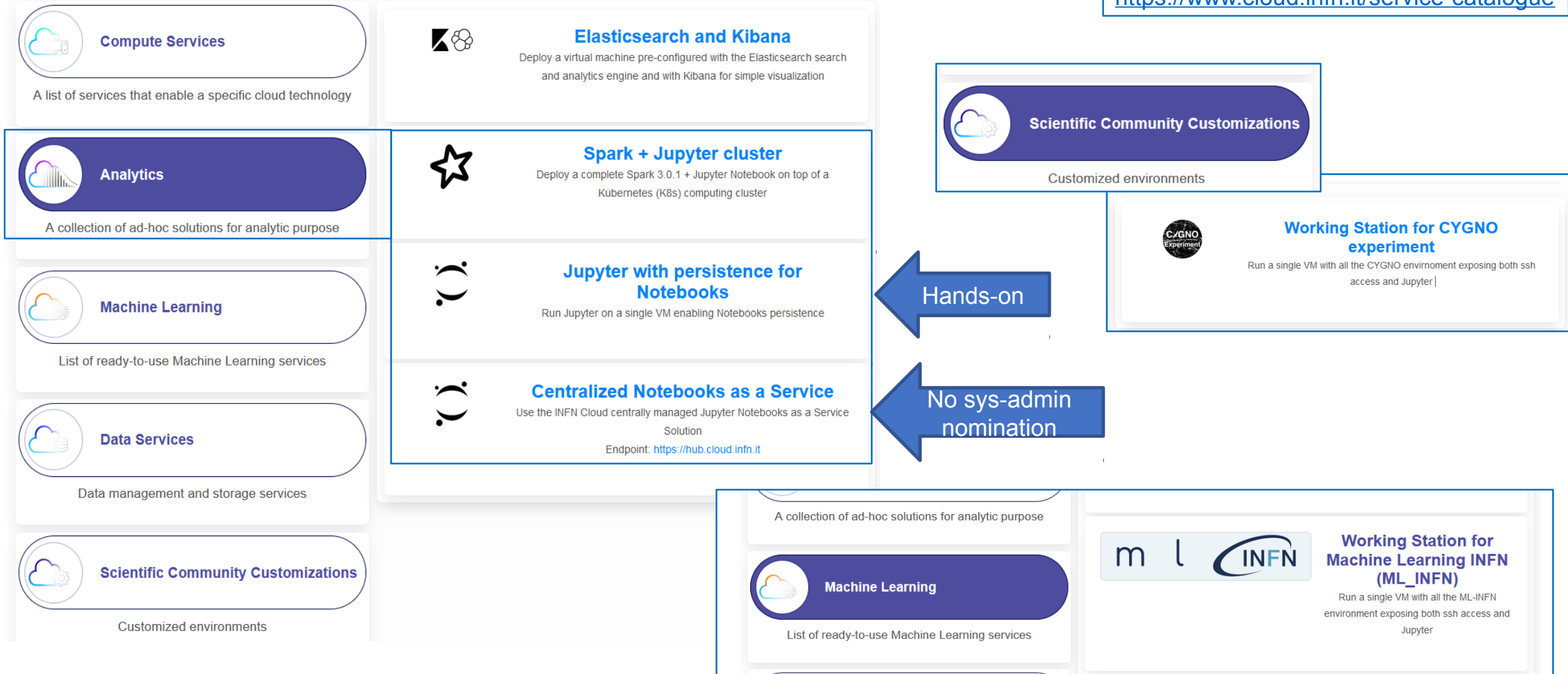
JupyterLab dashboard



The screenshot displays the JupyterLab dashboard in a web browser. The address bar shows the URL `https://192.135.24.189:8888/user/fanzago/lab`. The browser's status bar indicates CPU usage at 100% and memory usage at 608 / 1024 MB. The JupyterLab interface includes a top menu bar with options like File, Edit, View, Run, Kernel, Git, Tabs, Settings, Help, and Collaborate (Beta). A left sidebar shows a file explorer with a search bar and a list of files and folders: `/`, `cloud-storage`, `private`, `shared`, `hist.png`, `plot.ipynb` (selected), and `square.png`. The main workspace is divided into three sections: **Notebook**, **Console**, and **Other**. The Notebook section contains a `Python 3 (ipykernel)` kernel icon. The Console section also contains a `Python 3 (ipykernel)` kernel icon. The Other section contains icons for `Terminal`, `Text File`, `Markdown File`, `Python File`, and `Show Contextual Help`. The bottom status bar shows `Simple` mode, a `Mem: 254.77 / 1024.00 MB` indicator, and a `launcher` button.

INFN Cloud services based on Jupyter

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



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

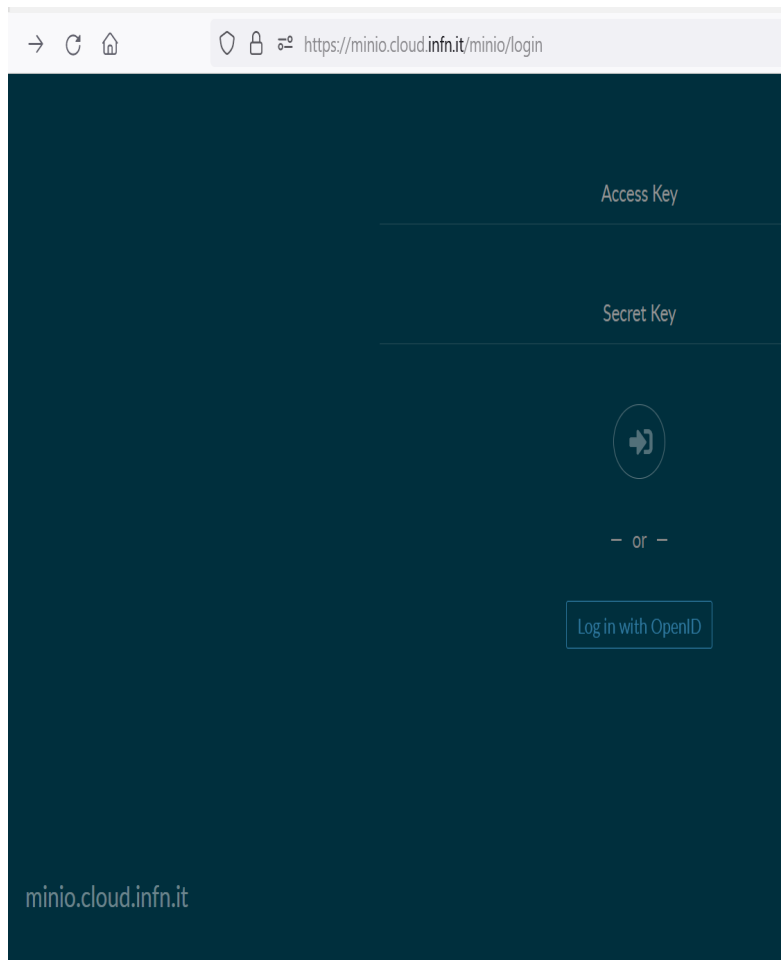
- INFN Cloud provides object storage based on Openstack Swift, as centrally managed service.
- The object storage is replicated in the two sites of backbone, Bari and CNAF
 - It guarantees the redundancy of data
 - There is not backup of data
- INFN Cloud is using the Minio-gateway software on Swift
 - 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>



→ ↻ 🏠 <https://minio.cloud.infn.it/minio/login>

Access Key

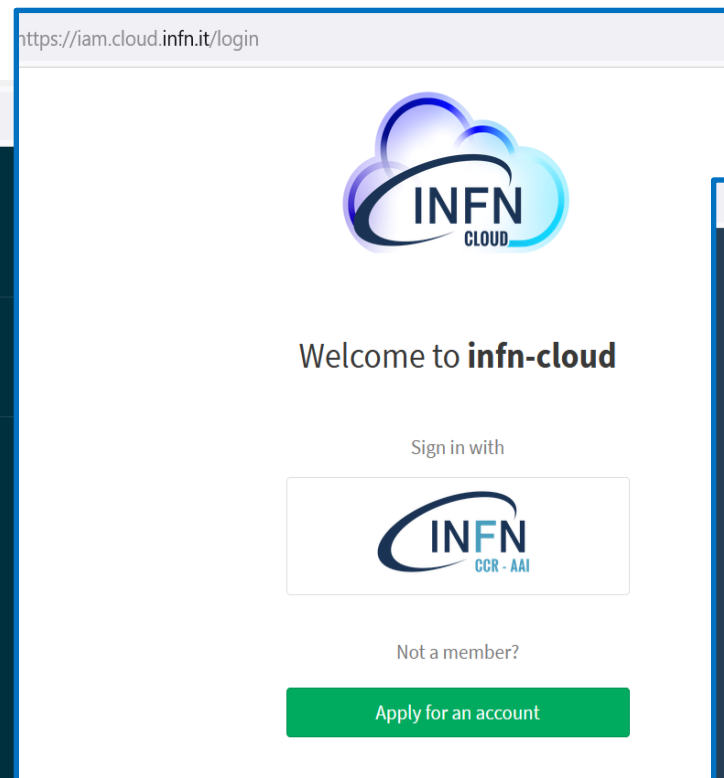
Secret Key

→


— or —

[Log in with OpenID](#)

minio.cloud.infn.it




<https://iam.cloud.infn.it/login>



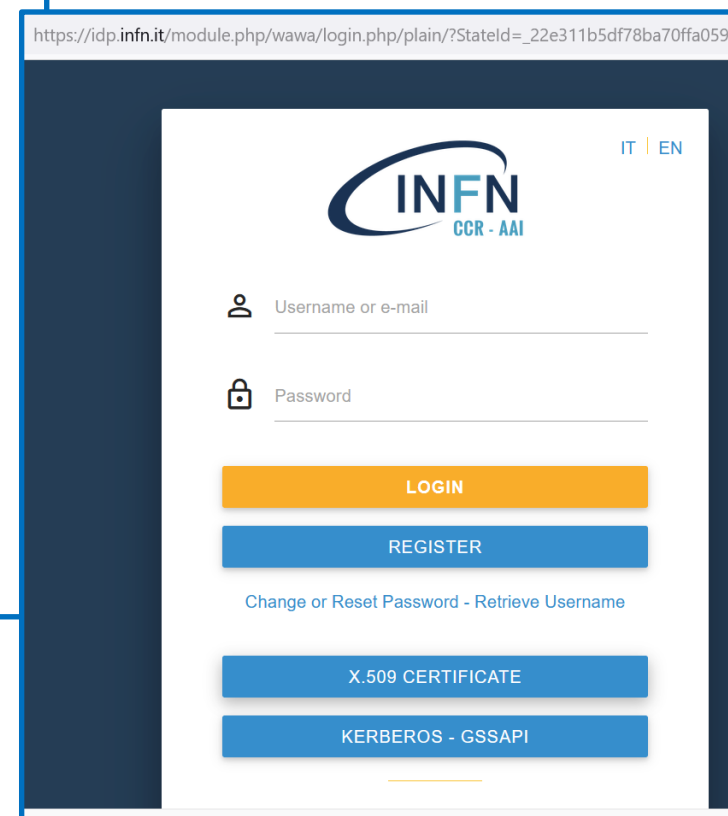
Welcome to **infn-cloud**

Sign in with




Not a member?

[Apply for an account](#)



https://idp.infn.it/module.php/wawa/login.php/plain?StateId=_22e311b5df78ba70ffa0596



IT | EN

👤 Username or e-mail

🔒 Password

[LOGIN](#)

[REGISTER](#)

[Change or Reset Password - Retrieve Username](#)

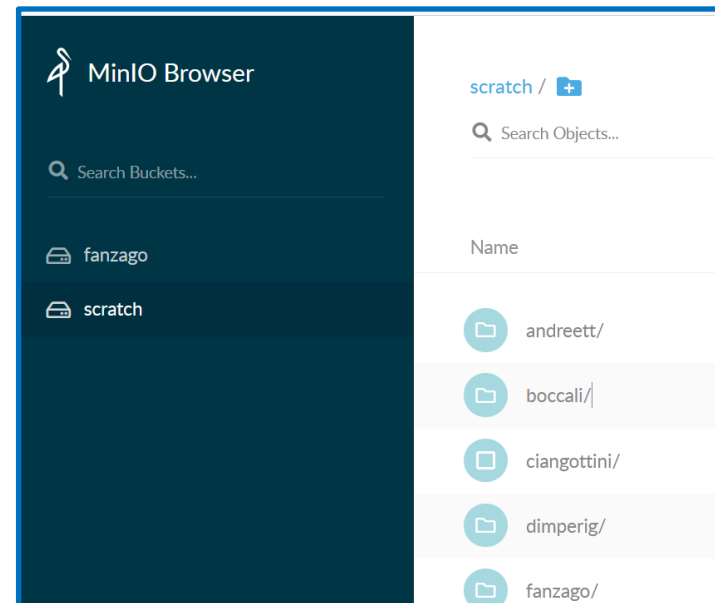
[X.509 CERTIFICATE](#)

[KERBEROS - GSSAPI](#)

Persistent data with object storage

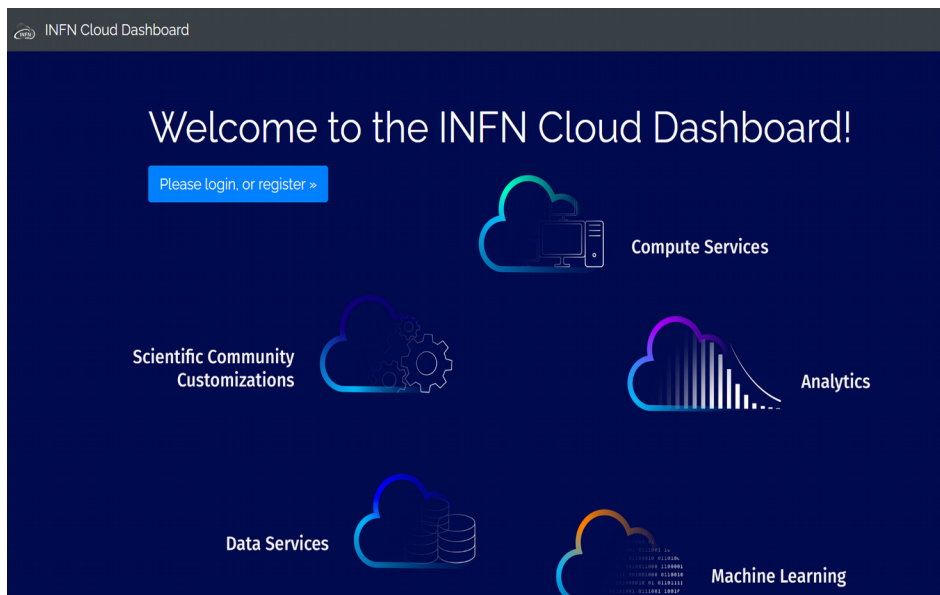
- All the services running notebooks mount these areas, so they are visible in the file system as posix directories.
 - cloud-storage dir
- The personal bucket can be created by user accessing the Minio webui otherwise it is automatically created the first time user instantiates a notebook.

Using sts-wire, fuse and RClone

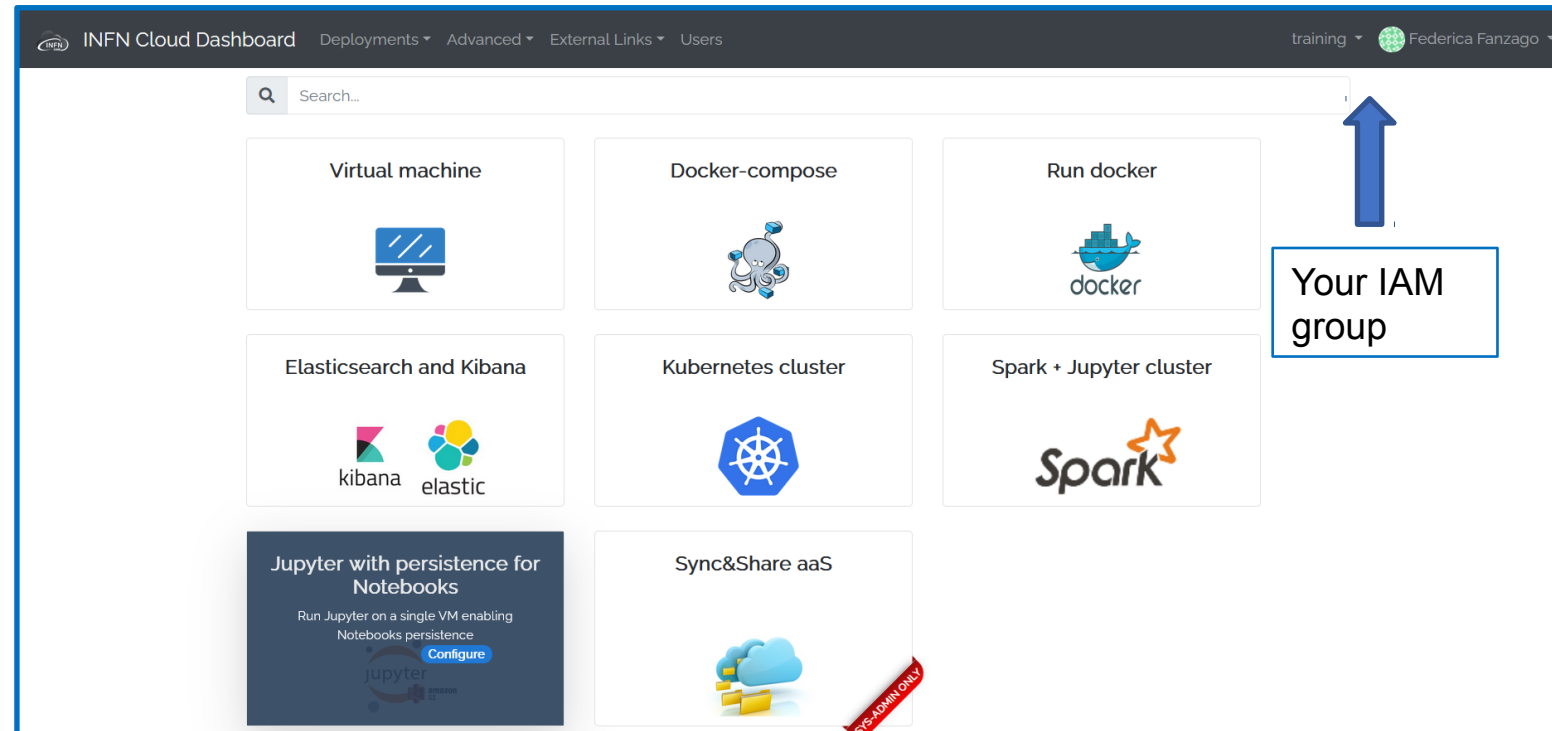


“Jupyter with persistence” deployment

- 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...

Jupyter with persistence for Notebooks

Description: Run Jupyter on a single VM enabling Notebooks persistence

Deployment description
test_jupy_fanzago ← **Mandatory field**

General Authorizations Advanced

num_cpus
2
Number of virtual cpus for the VM

mem_size
4 GB
Amount of memory for the VM

iam_url
https://iam.cloud.infn.it ← **For user authentication**

enable_monitoring
false
Enable/disable monitoring

jupyter_images
dodasts/snj-base-lab-persistencev1.0.3-snj ← **Hub and Lab image**
Default image

jupyterlab_collaborative
false
enable the jupyter collaborative service

jupyterlab_collaborative_image
dodasts/snj-base-jlabcv1.0.5-snj
Default image for jupyter collaborative service

ports
Add rule ← **The necessary ports for Jupyter are already configured**
Ports to open on the VM

Submit Cancel

Jupyter with persistence for Notebooks

Description: Run Jupyter on a single VM enabling Notebooks persistence

Deployment description
test_jupy_fanzago

General Authorizations Advanced

iam_groups
training ← **Authorized group of users**
IAM groups for authorization management (comma delimited list of strings)

iam_admin_groups
IAM groups for JupyterHub ADMIN authorization management (comma delimited list of strings)

Submit Cancel

User guide:

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

...then submit the deployment

Jupyter with persistence for Notebooks

Description: Run Jupyter on a single VM enabling Notebooks persistence

Deployment description
test_jupy_fanzago

General Authorizations Advanced


Configure scheduling:
 Auto Manual

Set deployment creation timeout (minutes) 720

Do not delete the deployment in case of failure

Send a confirmation email when complete

Submit Cancel



My deployments Refresh +

Show 10 entries Search:

Description	Deployment identifier	Status	Creation time	Deployed at	Actions
test_jupy_fanzago	11ed3115-81ec-13f1-b185-0242a79ac9f5	CREATE_IN_PROGRESS	2022-09-10 14:33:00		Details
jup_sologruppotraining	11ed2eee-199a-c181-b185-0242a79ac9f5	CREATE_COMPLETE	2022-09-07 20:46:00	BACKBONE-CNAF	Details

Deployment details

Description	Deployment identifier	Status	Creation time	Deployed at	Actions
test_senza_iam	11ed3120-0a8e-5dcb-b185-0242a79ac9f5	CREATE_COMPLETE	2022-09-10 15:48:00	BACKBONE-CNAF	Details
test_jupy_fanzago	11ed3115-81ec-13f1-b185-0242a79ac9f5	CREATE_COMPLETE	2022-09-10 14:33:00	BACKBONE-CNAF	Details
jup_sologruppotraining	11ed2eee-199a-c181-b185-0242a79ac9f5	CREATE_COMPLETE	2022-09-07 20:46:00	BACKBONE-CNAF	<ul style="list-style-type: none"> Edit Show template Log Request Ports VM details Lock Delete
fede_prova_training	11ed2a15-341d-c7ad-b185-0242a79ac9f5	CREATE_COMPLETE	2022-09-01 16:43:00	BACKBONE-CNAF	

Showing 1 to 4 of 4 entries

[11ed3115-81ec-13f1-b185-0242a79ac9f5](#)

Description: test_jupy_fanzago

Overview Input values Output values

node_ip: 192.135.24.189

jupyter_endpoint: <https://192.135.24.189:8888>

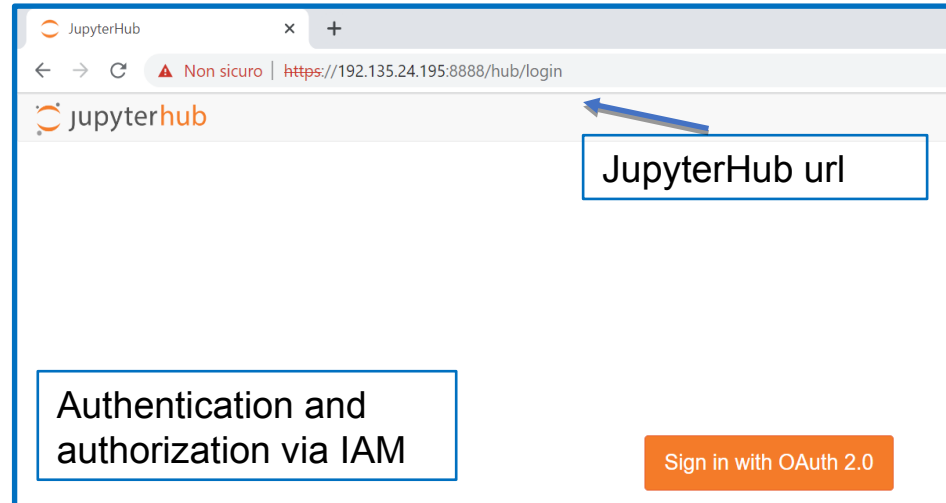
ssh_account: fanzago

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

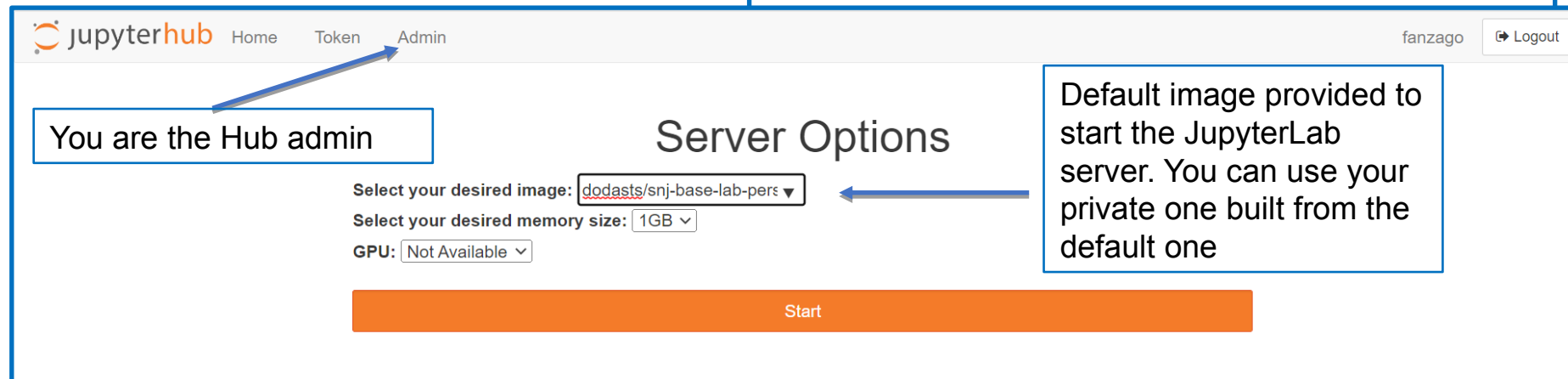
Your Jupyterhub



JupyterHub url

Authentication and authorization via IAM

Sign in with OAuth 2.0



You are the Hub admin

Server Options

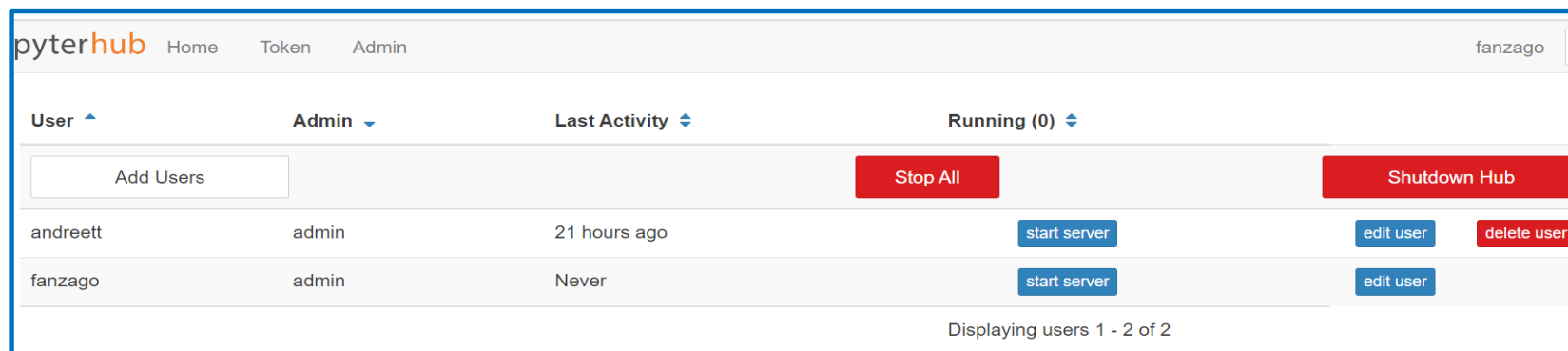
Select your desired image:

Select your desired memory size:

GPU:

Start

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

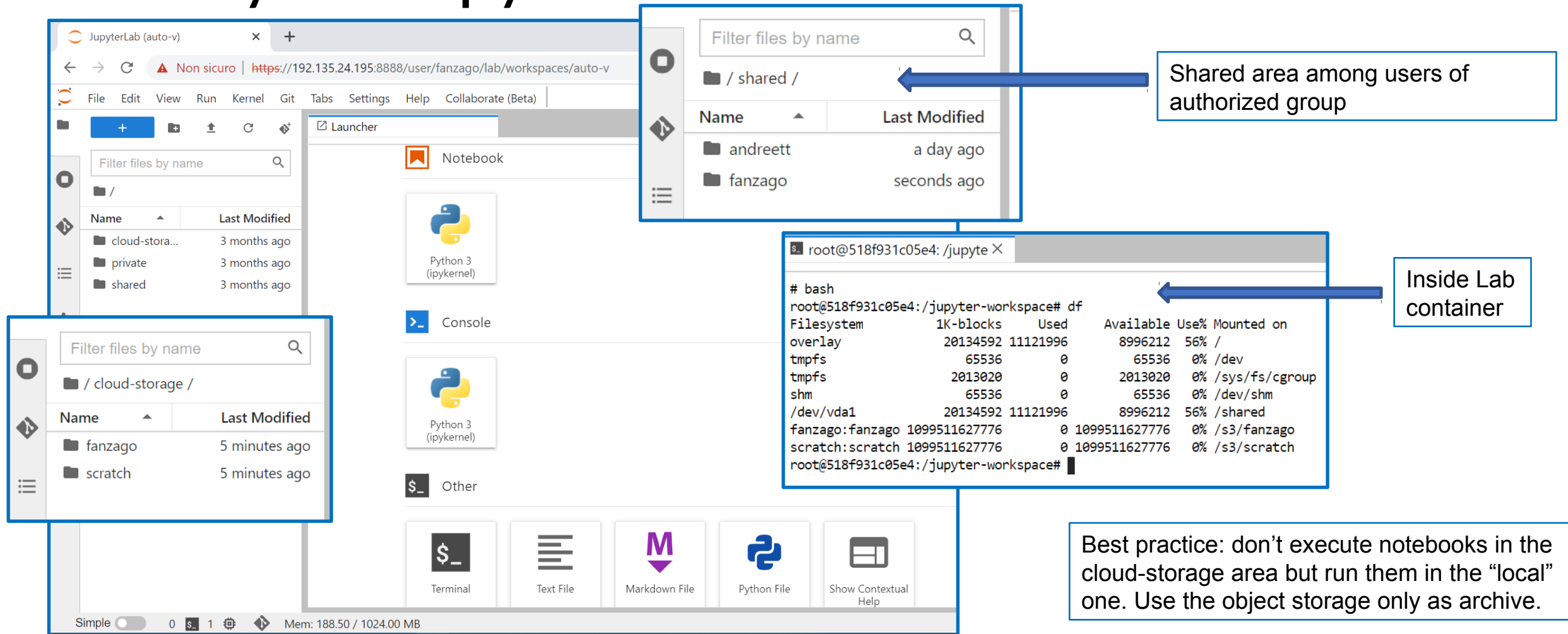


User	Admin	Last Activity	Running (0)
Add Users			
Stop All			
Shutdown Hub			
andreett	admin	21 hours ago	start server edit user delete user
fanzago	admin	Never	start server edit user

Displaying users 1 - 2 of 2

Hub's admin can control users and notebooks

And your JupyterLab



The image shows a JupyterLab interface with several callouts:

- Shared area among users of authorized group:** Points to the file browser showing a shared directory with subdirectories 'andrett' and 'fanzago'.
- Inside Lab container:** Points to a terminal window showing the output of the 'df' command, which lists mounted filesystems and their usage.
- Best practice:** A text box stating: "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."

Terminal Output (df command):

```

root@518f931c05e4: /jupyter X
# bash
root@518f931c05e4: /jupyter-workspace# df
Filesystem      1K-blocks    Used   Available Use% Mounted on
overlay          20134592 11121996    8996212  56% /
tmpfs             65536         0         65536   0% /dev
tmpfs            2013020         0         2013020  0% /sys/fs/cgroup
shm              65536         0         65536   0% /dev/shm
/dev/vda1        20134592 11121996    8996212  56% /shared
fanzago:fanzago 1099511627776 0 1099511627776  0% /s3/fanzago
scratch:scratch 1099511627776 0 1099511627776  0% /s3/scratch
root@518f931c05e4: /jupyter-workspace#

```

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
614198cc94df	dodasts/snj-base-lab-persistence:v1.0.3-snj	"tini -s -- jupyterh..."	36 seconds ago	Up 32 seconds	8889/tcp
fd1fd7782f92	dodasts/monitoring-grafana:v1.0.5-monitoring	"/run.sh -config /op..."	3 days ago	Up 3 days	0.0.0
cac3853d61d1	prom/prometheus:v2.30.3	"/bin/prometheus --c..."	3 days ago	Up 3 days	0.0.0
c1053aabc34c	gcr.io/cadvisor/cadvisor:v0.37.5	"/usr/bin/cadvisor -..."	3 days ago	Up 3 days (healthy)	8080/tcp
21c06e24da05	prom/node-exporter:v1.2.2	"/bin/node_exporter"	3 days ago	Up 3 days	9100/tcp
093e4de1f276	dodasts/snj-base-jhub:v1.0.2p1-snj	"/usr/bin/python3 /u..."	3 days ago	Up 3 days	8000/tcp
e0af2178ec3f	jupyterhub/configurable-http-proxy	"/srv/configurable-h..."	3 days ago	Up 3 days	0.0.0

```
fanzago@vnode-0:~$
```