Quantum-Notebook: a Docker stack for quantum computing

Presenter:

Presenter: Antonio Falabella (INFN-CNAF)

Co-author

Dr. Silvio Pardi (INFN-Napoli)

Quantum Computing @ INFN Bologna 2022

Tools for Quantum Computing

- Today we have an increasing set of software for quantum computing that allows to
  - Design and describe quantum circuits
  - Interact with real quantum machines.

- Scientists who want to play with quantum circuits may profit of na enviroment properly configured on which develop their ideas.

- System administrators are called to support the final users by simplifying as much as possible the software installation and distribution.

JupyterLab

JupyterLab is a web-based interactive development environment for notebooks, code, and data, largely used by the scientific community.

Jupyter Docker Stacks provides a set of ready-to-run Docker images containing Jupyter applications and interactive computing tools.

- minimal-notebook
- base-notebook
- datascience-notebook
- pyspark-notebook
- r-notebook
- scipy-notebook
- tensorflow-notebook

https://jupyterlab.readthedocs.io/en/stable/

https://github.com/jupyter/docker-stacks

Quantum Notebook

The idea is to extend the Jupyter Docker Stacks with a new r**eady to use image** (the Quantum Notebook) containing the most common tools for quantum computing.

The docker image can be deployed quickly over a standalone machine or a Cloud infrastructure.

Quantum Notebook is published on the INFN GIT repository:

https://baltig.infn.it/quantum_computing/quantum_notebook

List of selected libraries

- OPENQASM https://github.com/openqasm/openqasm
- QISKIT IBM https://qiskit.org/
- CIRQ https://github.com/quantumlib/cirq
- PyQUIL from Rigetti https://github.com/rigetti/pyquil
- ProkjectQ https://projectq.readthedocs.io/en/latest/index.html
- myQLM https://myqlm.github.io/myqlm_specific/install.html
- QSHARP Microsoft https://docs.microsoft.com/it-it/dotnet/core/install/linux-ubuntu

Two approaches for Quantum Notebook

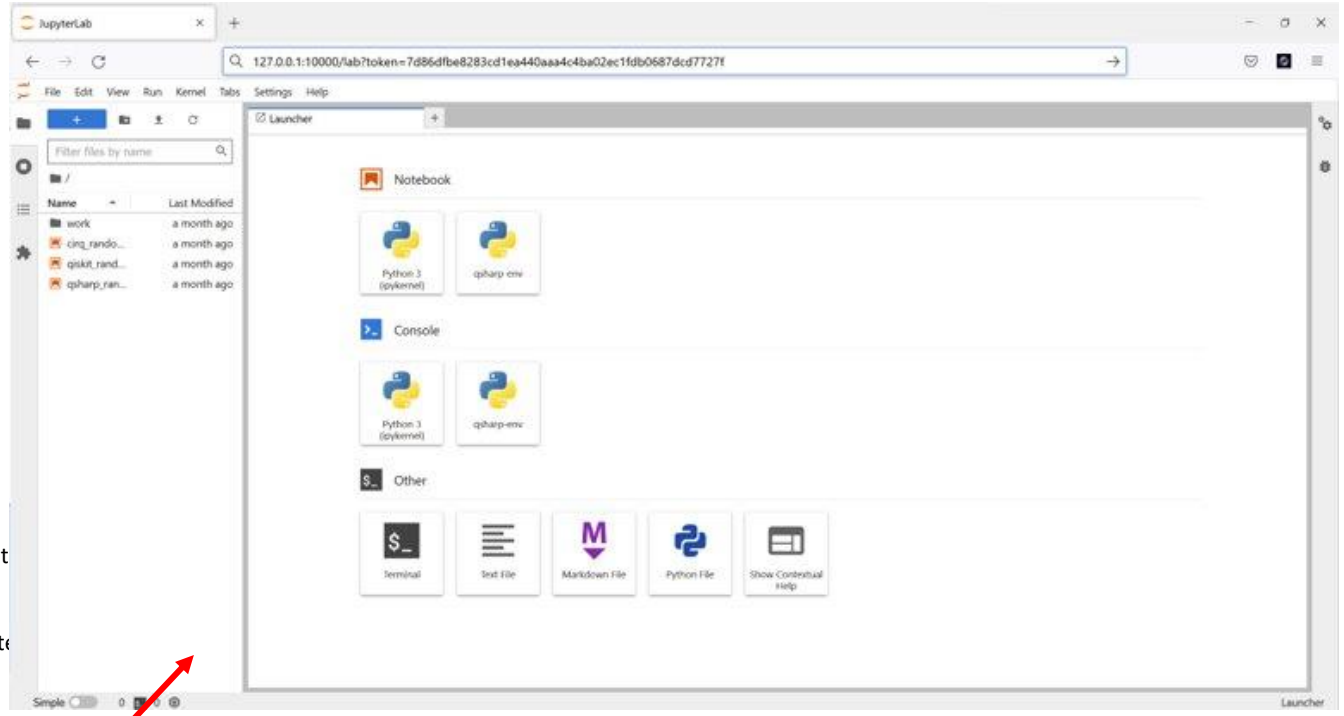
- **Standalone Docker Image**
  - Following the Jupyter Docker Stacks project, this solution provides a ready to deploy docker image that can be used over any private resources.

- **Quantum Notebook as a service over INFN Cloud**
  - This approach allow to deploy a Quantum Notebook as a service over the INFN Cloud taking advantage of the other Cloud services: remote network access, authentication and data persistency .

```
# Copyright (c) Jupyter Development Team.
# Distributed under the terms of the Modified BSD License.
# Added Quantum Library support
FROM jupyter/scipy-notebook

# Installation Quantum libs
RUN pip install --quiet --no-cache-dir \
# Needed library
#    qat \
# OPENQASM
    openqasm3  \
# QISKIT IBM https://qiskit.org/
    qiskit \
# CIRQ https://github.com/quantumlib/cirq
    cirq \
# PyQUIL from righetti  https://github.com/rigetti/pyquil
    pyquil \
# ProkjectQ https://projectq.readthedocs.io/en/latest/index.html
    projectq \
    ipykernel \
    pylatexenc \
    pylatex &&\
#    python -m qat.magics.install && \
    fix-permissions "${CONDA_DIR}" && \
    fix-permissions "/home/${NB_USER}"
```

Standalone Docker image

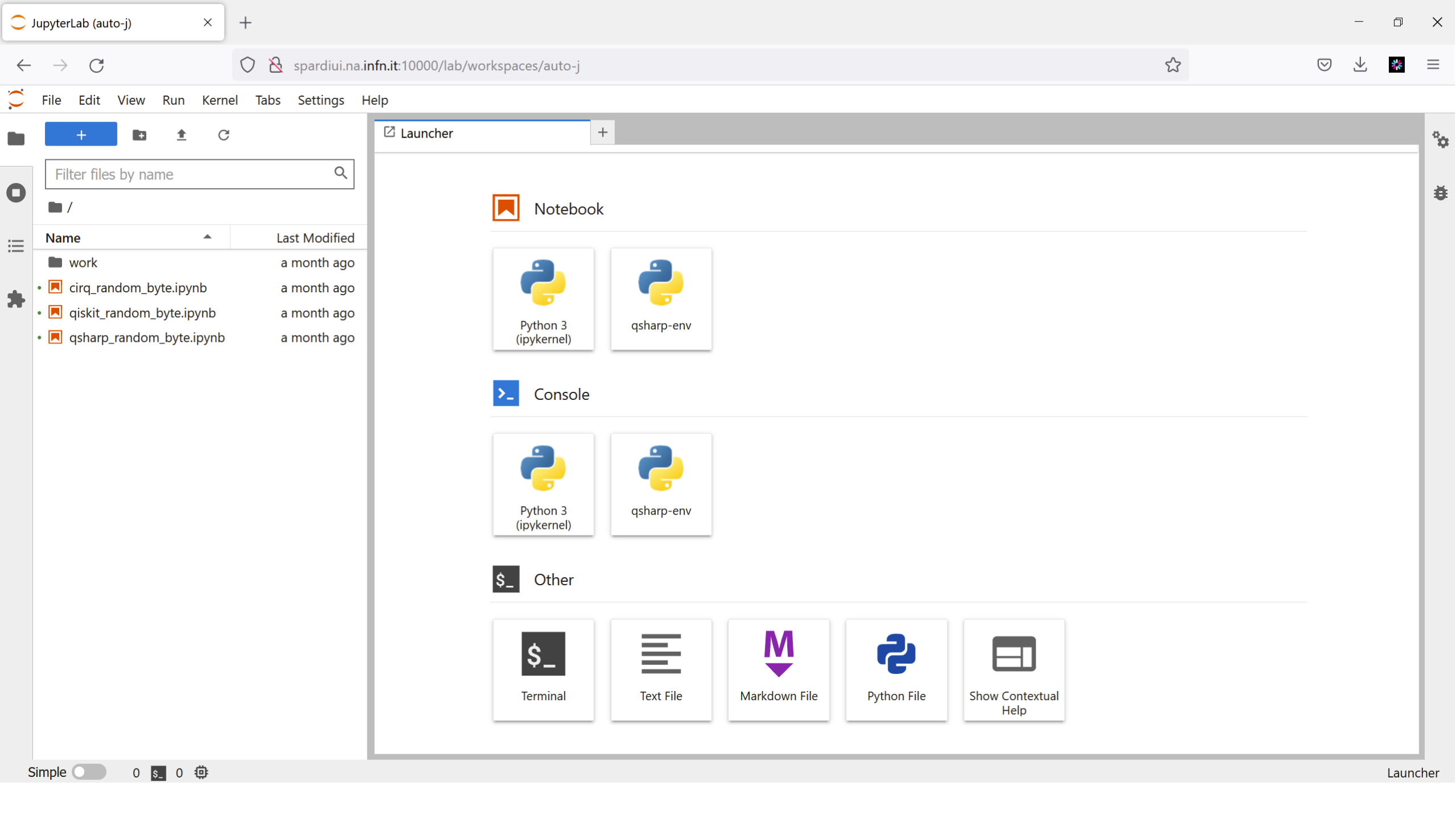
The standalone version of Quantum Notebook can be run over every server, PC, laptop with docker server running.

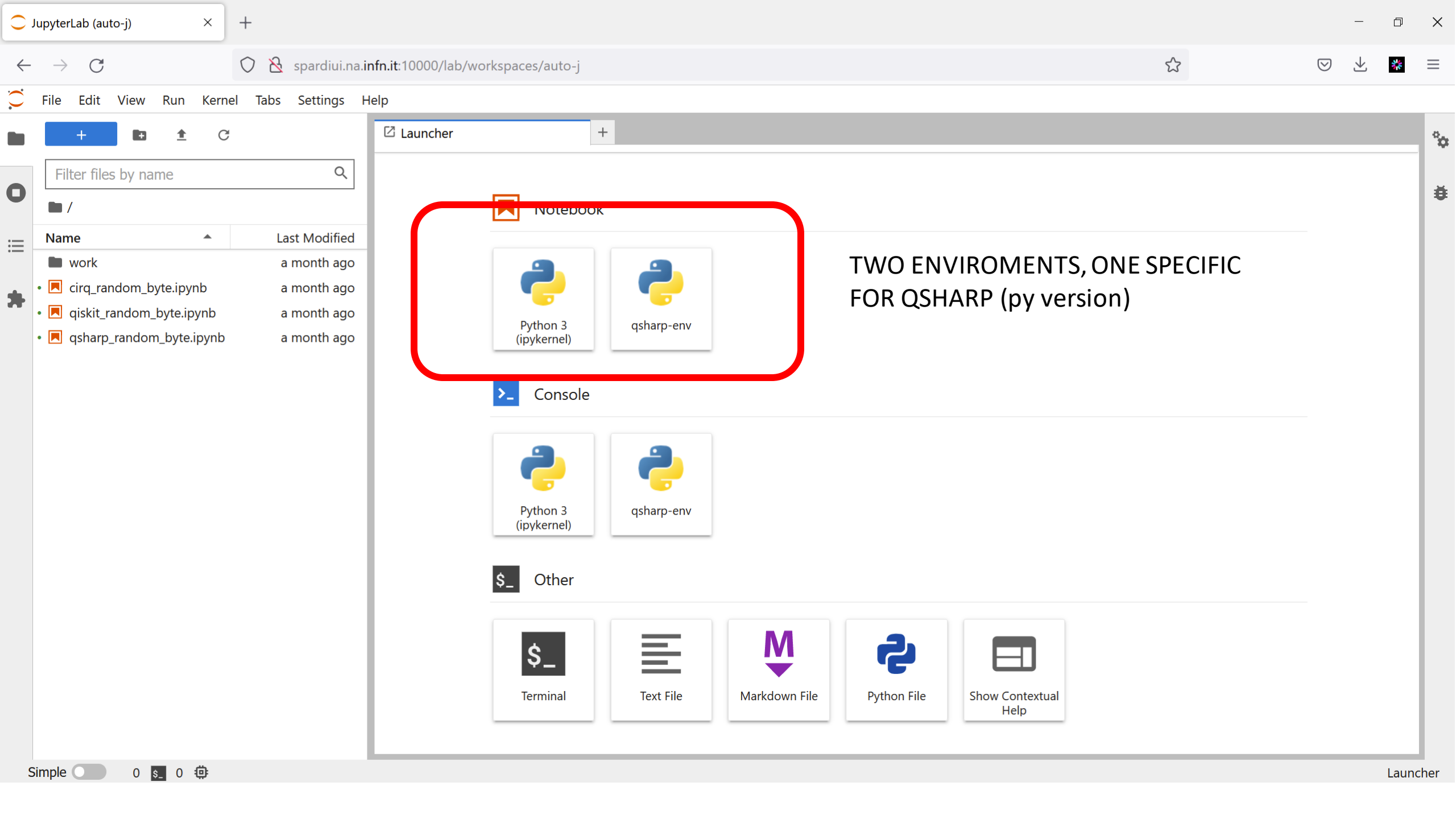


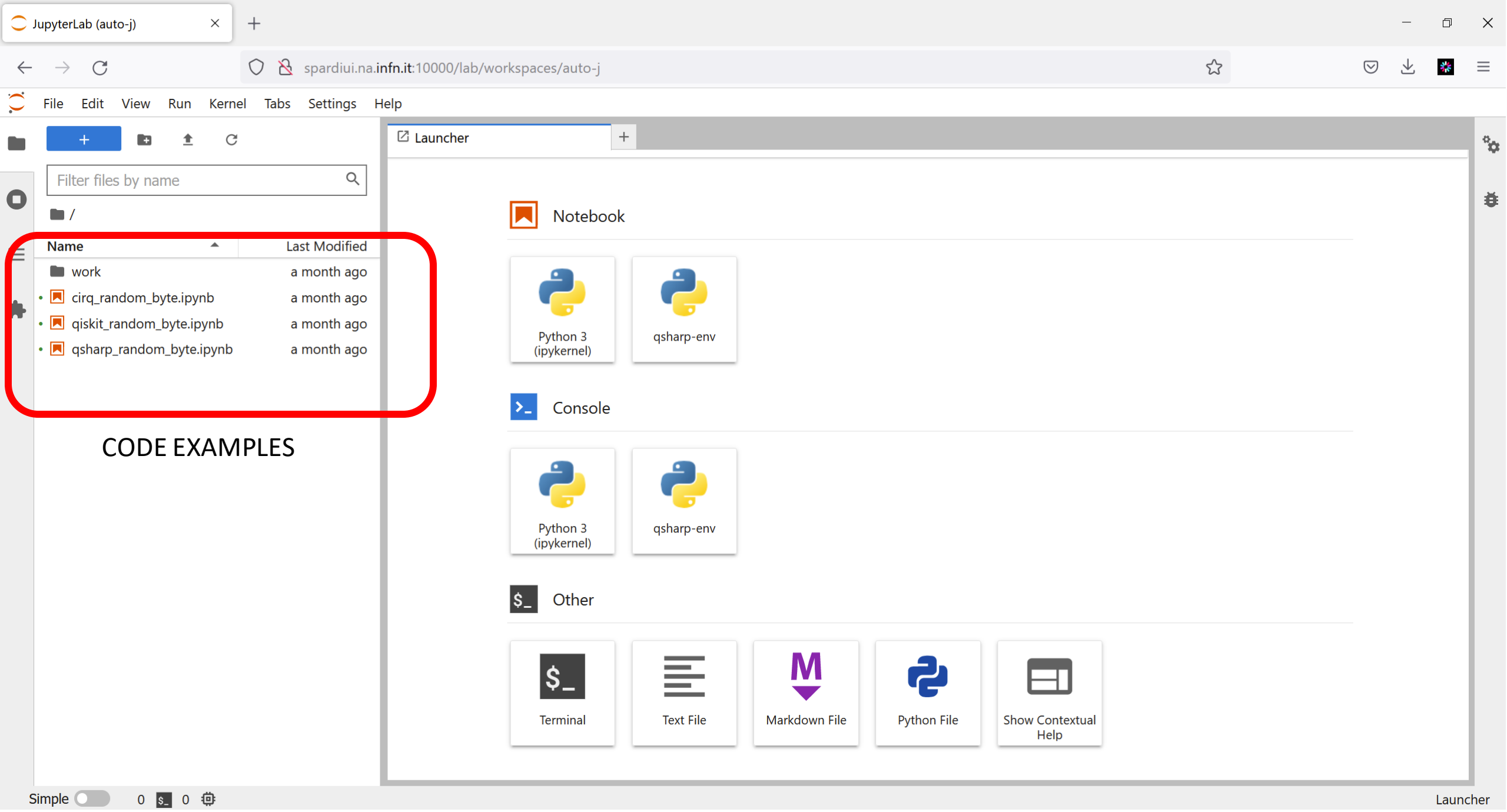
[spardi@spardiui ~]$ **docker run -p 10000:8888 d9cbfb7acf59**

Entered start.sh with args: jupyter lab
Executing the command: jupyter lab
[I 2022-09-08 13:37:11.091 ServerApp] jupyterlab | extension was successfully linked.
[I 2022-09-08 13:37:11.104 ServerApp] nbclassic | extension was successfully linked.
[I 2022-09-08 13:37:11.106 ServerApp] Writing Jupyter server cookie secret to /home/jovyan/.local/share/jupyter/runt
[I 2022-09-08 13:37:12.006 ServerApp] notebook_shim | extension was successfully linked.
[I 2022-09-08 13:37:12.053 ServerApp] notebook_shim | extension was successfully loaded.
[I 2022-09-08 13:37:12.055 LabApp] JupyterLab extension loaded from /opt/conda/lib/python3.10/site-packages/jupyte
[I 2022-09-08 13:37:12.055 LabApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
[I 2022-09-08 13:37:12.059 ServerApp] jupyterlab | extension was successfully loaded.
[I 2022-09-08 13:37:12.082 ServerApp] nbclassic | extension was successfully loaded.
[I 2022-09-08 13:37:12.082 ServerApp] Serving notebooks from local directory: /home/jovyan
[I 2022-09-08 13:37:12.083 ServerApp] Jupyter Server 1.18.1 is running at:
[I 2022-09-08 13:37:12.083 ServerApp] http://67ec3f1082cf:8888/lab?token=27afd5e384f4f76203963d542709c8c7fb0bf09a21a36348
[I 2022-09-08 13:37:12.083 ServerApp] or http://127.0.0.1:8888/lab?token=27afd5e384f4f76203963d542709c8c7fb0bf09a21a36348
[I 2022-09-08 13:37:12.083 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 2022-09-08 13:37:12.093 ServerApp]

To access the server, open this file in a browser:
file:///home/jovyan/.local/share/jupyter/runtime/jpserver-7-open.html
Or copy and paste one of these URLs:
http://67ec3f1082cf:8888/lab?token=27afd5e384f4f76203963d542709c8c7fb0bf09a21a36348
or http://127.0.0.1:8888/lab?token=27afd5e384f4f76203963d542709c8c7fb0bf09a21a36348


JupyterLab (auto-j)
spardiui.na.infn.it:10000/lab/workspaces/auto-j
File
Edit
View
Run
Kernel
Tabs
Settings
Help
Filter files by name
/
Name
Last Modified
work
a month ago
cirq_random_byte.ipynb
a month ago
qiskit_random_byte.ipynb
a month ago
qsharp_random_byte.ipynb
a month ago
Launcher
Notebook
Python 3 (ipykernel)
qsharp-env
Console
Python 3 (ipykernel)
qsharp-env
Other
Terminal
Text File
Markdown File
Python File
Show Contextual Help
Simple
0
0
Launcher


JupyterLab (auto-j)
spardiui.na.infn.it:10000/lab/workspaces/auto-j
File Edit View Run Kernel Tabs Settings Help
Launcher
Filter files by name
/
Name
Last Modified
work a month ago
cirq_random_byte.ipynb a month ago
qiskit_random_byte.ipynb a month ago
qsharp_random_byte.ipynb a month ago
Notebook
Python 3 (ipykernel)
qsharp-env
TWO ENVIROMENTS, ONE SPECIFIC FOR QSHARP (py version)
Console
Python 3 (ipykernel)
qsharp-env
Other
Terminal
Text File
Markdown File
Python File
Show Contextual Help
Simple
0
0
Launcher


JupyterLab (auto-j)
spardiui.na.infn.it:10000/lab/workspaces/auto-j
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/
Name
Last Modified
work a month ago
cirq_random_byte.ipynb a month ago
qiskit_random_byte.ipynb a month ago
qsharp_random_byte.ipynb a month ago
CODE EXAMPLES
Launcher
Notebook
Python 3 (ipykernel)
qsharp-env
Console
Python 3 (ipykernel)
qsharp-env
Other
Terminal
Text File
Markdown File
Python File
Show Contextual Help
Simple
0
0
Launcher

qiskit_rando... (3) - JupyterLab

spardiui.na.infn.it:10000/lab/tree/qiskit_random_byte.ipynb

File Edit View Run Kernel Tabs Settings Help

Launcher | qiskit_random_byte.ipynb | cirq_random_byte.ipynb | qsharp_random_byte.ipynb

Code

Python 3 (ipykernel)

Filter files by name

/

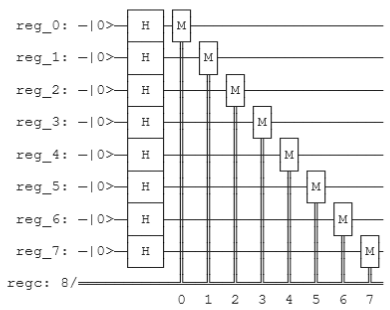
Name	Last Modified
work	a month ago
cirq_random_byte.ipynb	a month ago
qiskit_random_byte.ipynb	a month ago
qsharp_random_byte.ipy...	a month ago

[5]:
```
## Programming Quantum Computers
##   by Eric Johnston, Nic Harrigan and Mercedes Gimeno-Segovia
##   O'Reilly Media
##
## More samples like this can be found at http://oreilly-qc.github.io
#
from qiskit import QuantumCircuit, QuantumRegister, ClassicalRegister, execute, Aer, IBMQ, BasicAer
import math
## Uncomment the next line to see diagrams when running in a notebook
%matplotlib inline

## Example 2-2: Random byte
# Set up the program
reg = QuantumRegister(8, name='reg')
reg_c = ClassicalRegister(8, name='regc')
qc = QuantumCircuit(reg, reg_c)

qc.reset(reg)          # write the value 0
qc.h(reg)              # put it into a superposition of 0 and 1
qc.measure(reg, reg_c) # read the result as a digital bit

backend = BasicAer.get_backend('statevector_simulator')
job = execute(qc, backend)
result = job.result()

# Convert the result into a random number
counts = result.get_counts(qc)
print('counts:',counts)
for key,val in counts.items():
    n = sum([(int(x) << i) for i,x in enumerate(key)])
    print('Random number:', n)
#outputstate = result.get_statevector(qc, decimals=3)
#print(outputstate)
qc.draw()        # draw the circuit
```

QISKIT EXAMPLE

```
counts: {'10110110': 1}
Random number: 109
```

[5]:

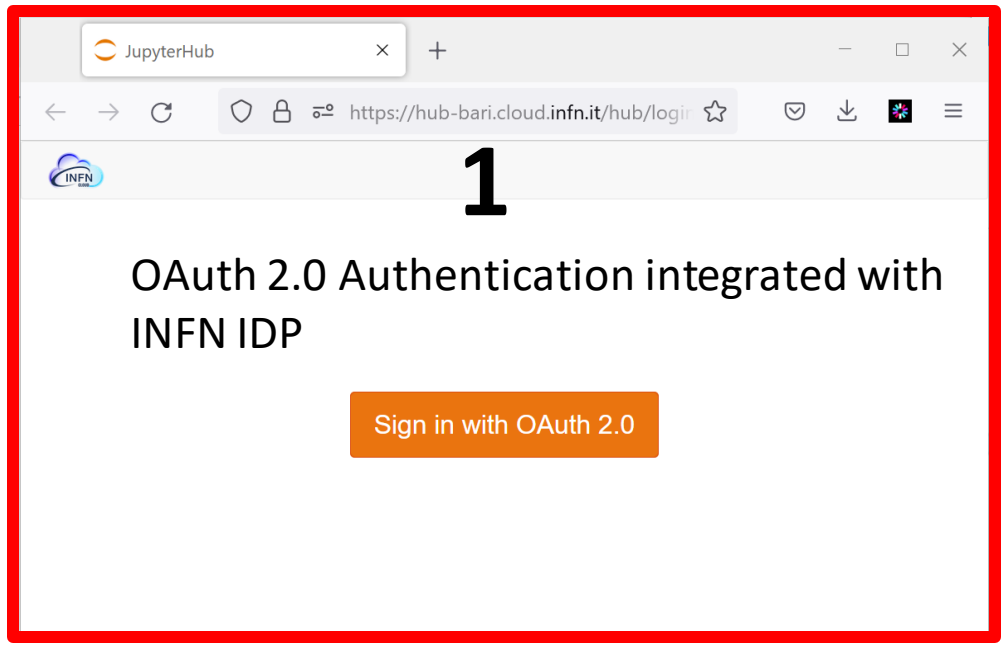


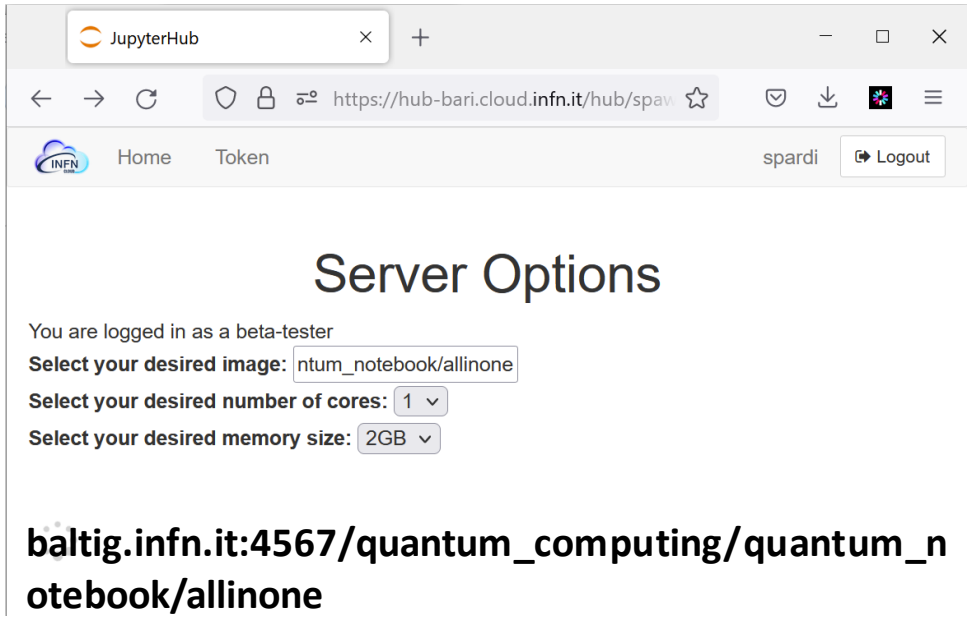
[ ]:

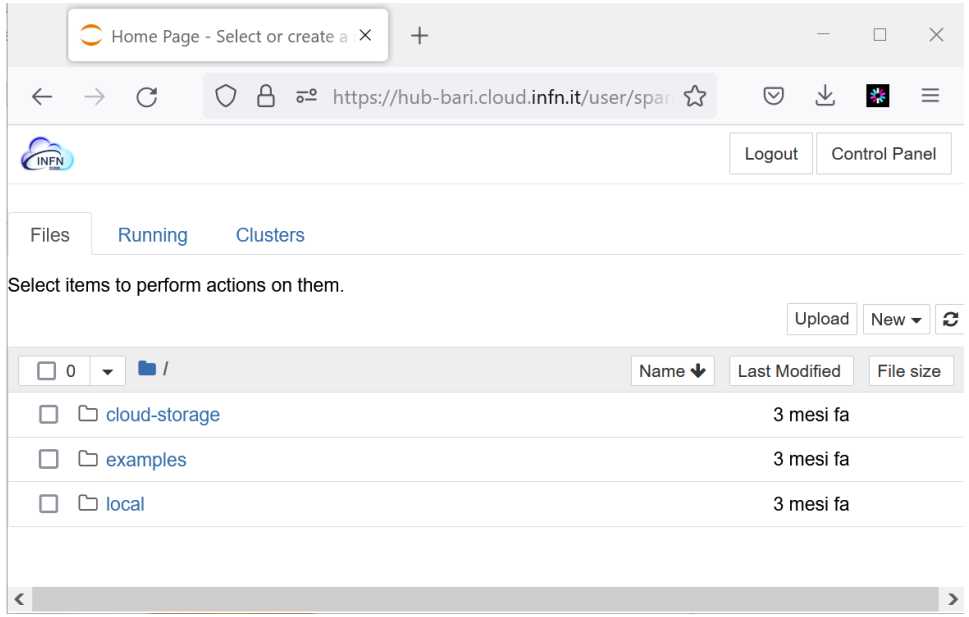
Simple 0 3 Python 3 (ipykernel) | Idle

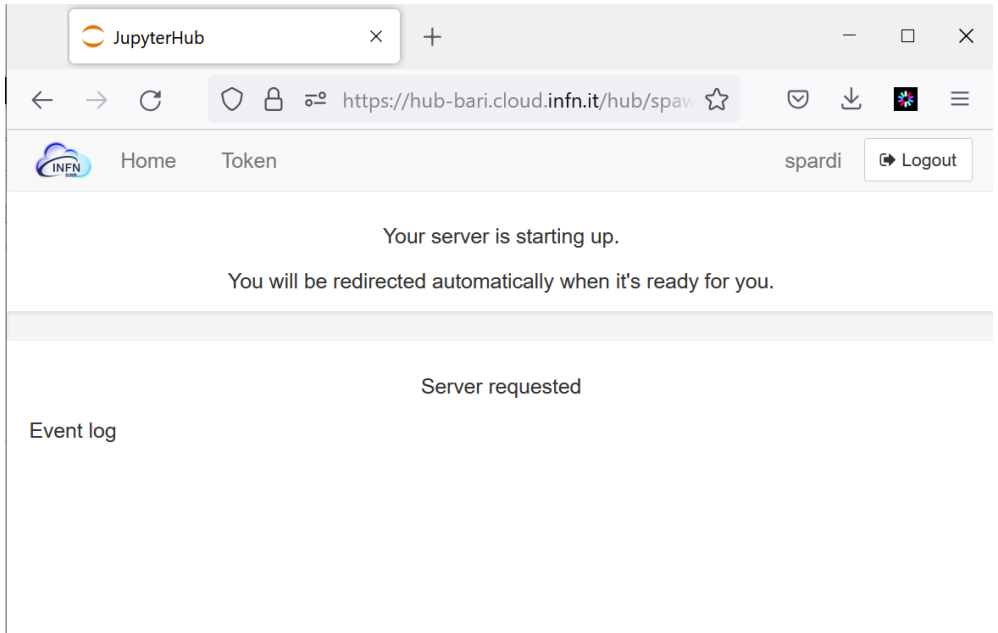
Mode: Edit Ln 4, Col 3 qiskit_random_byte.ipynb

INFN Cloud Image

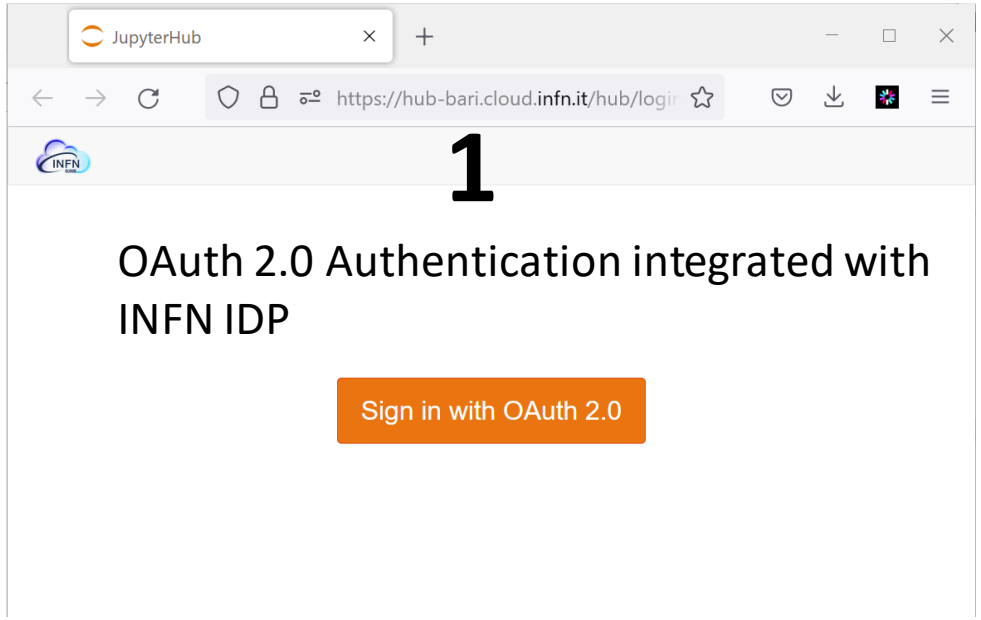


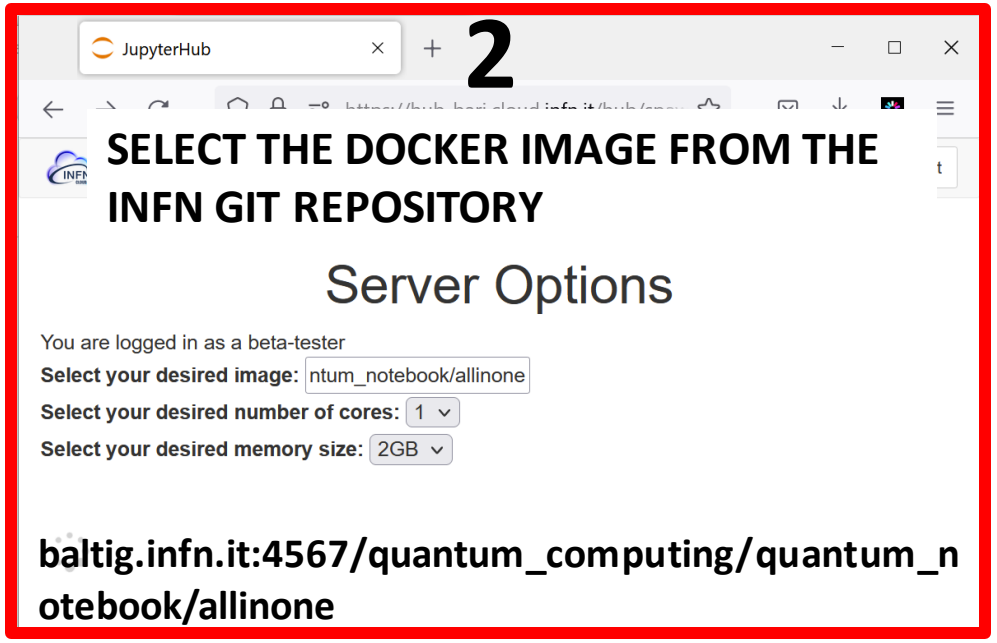


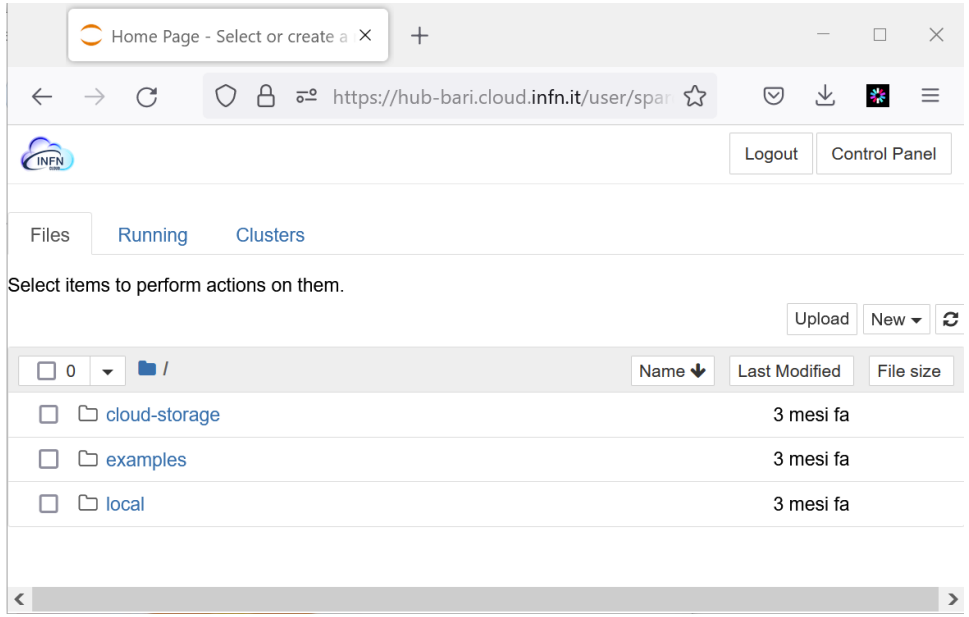


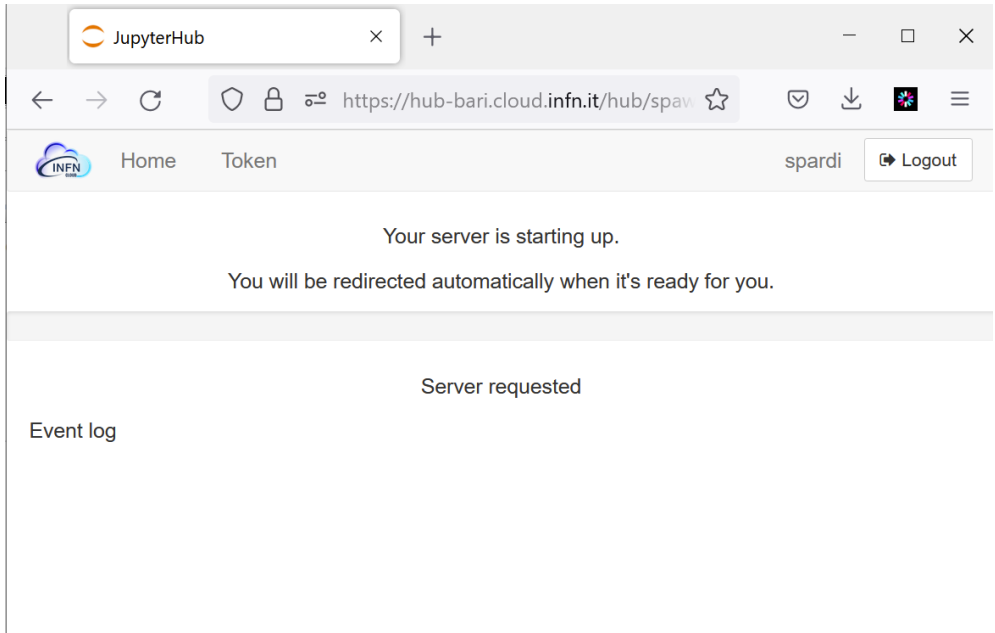


INFN Cloud Image

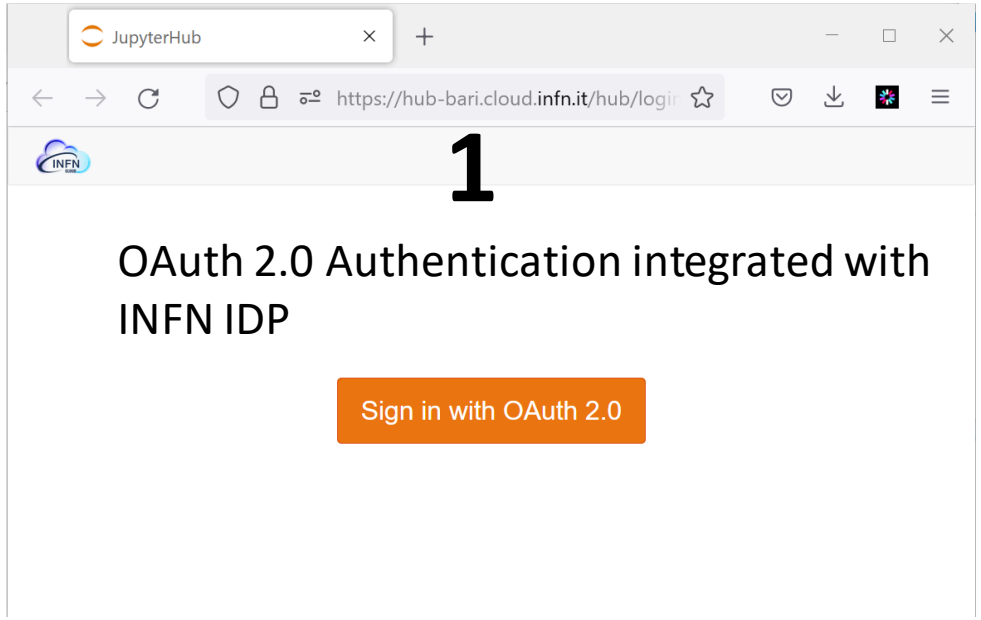


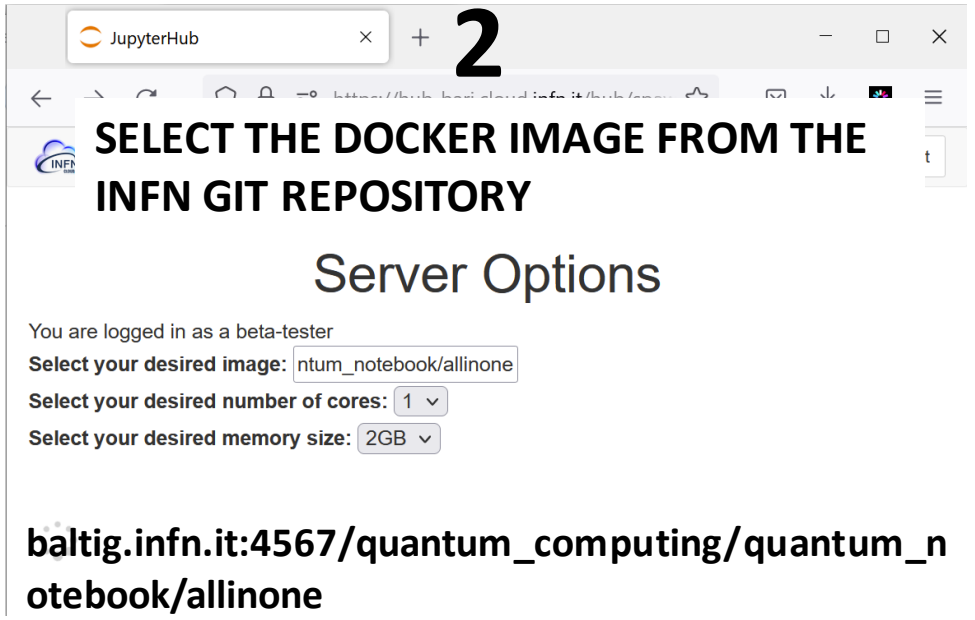


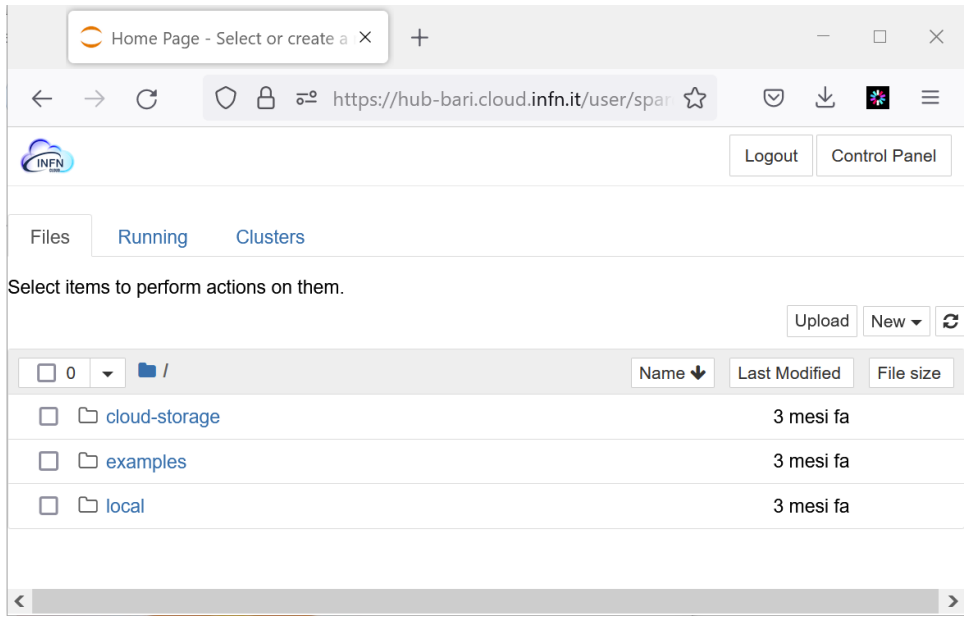


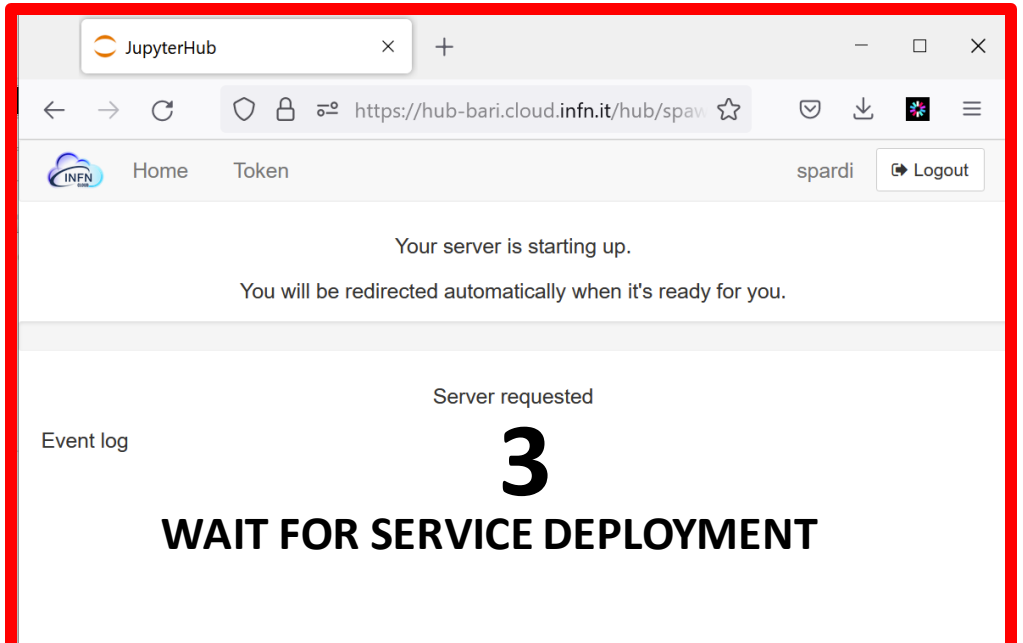


INFN Cloud Image

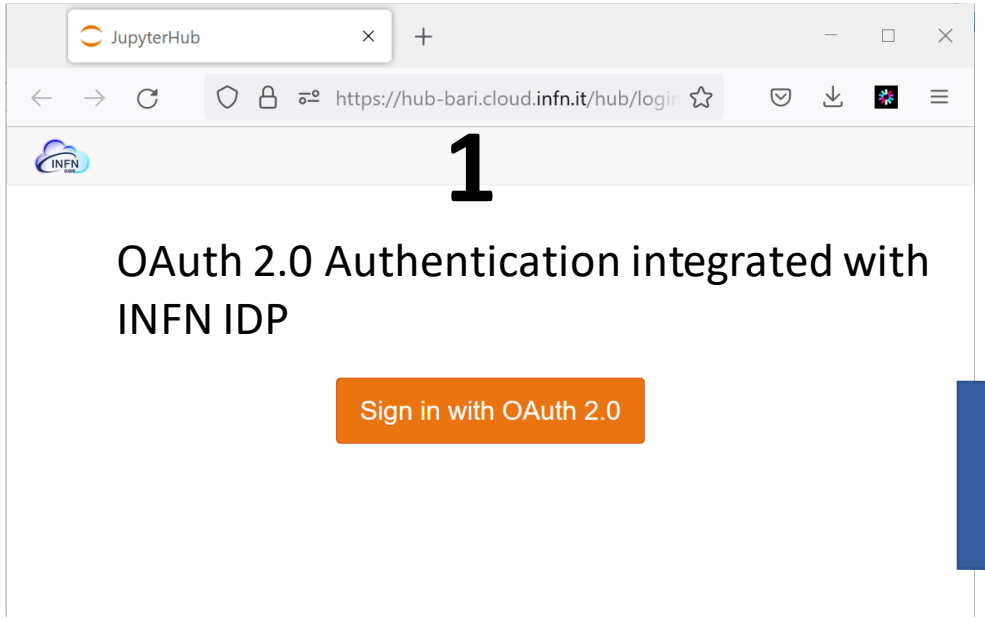


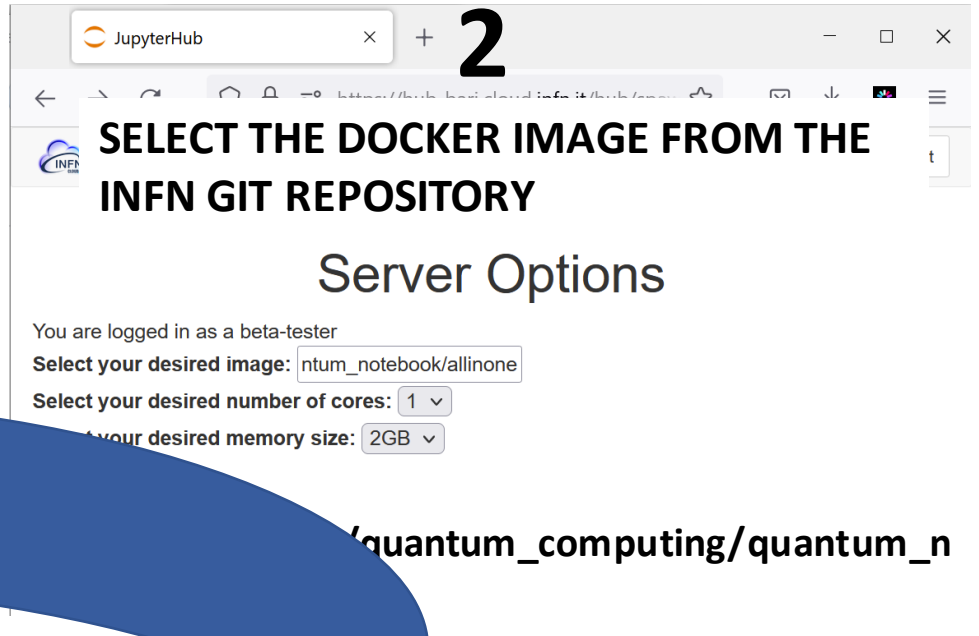


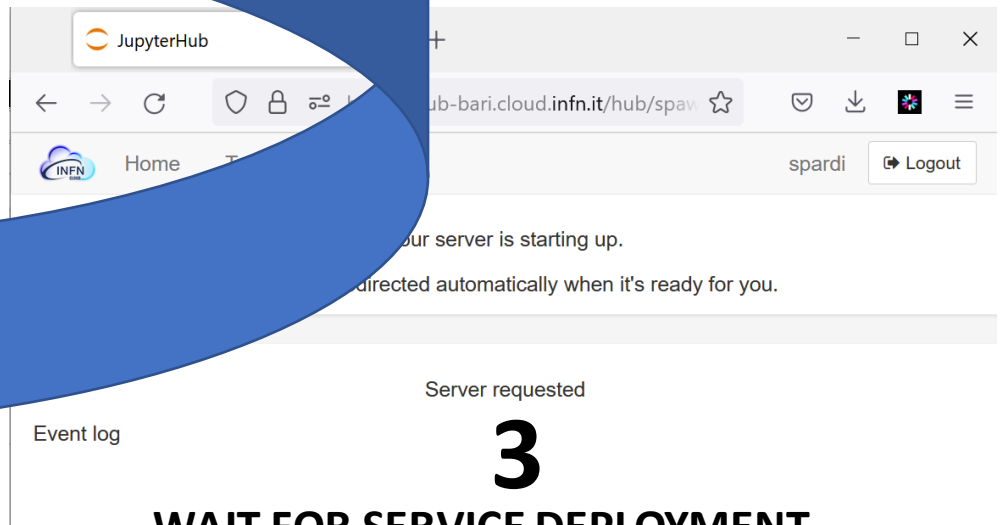


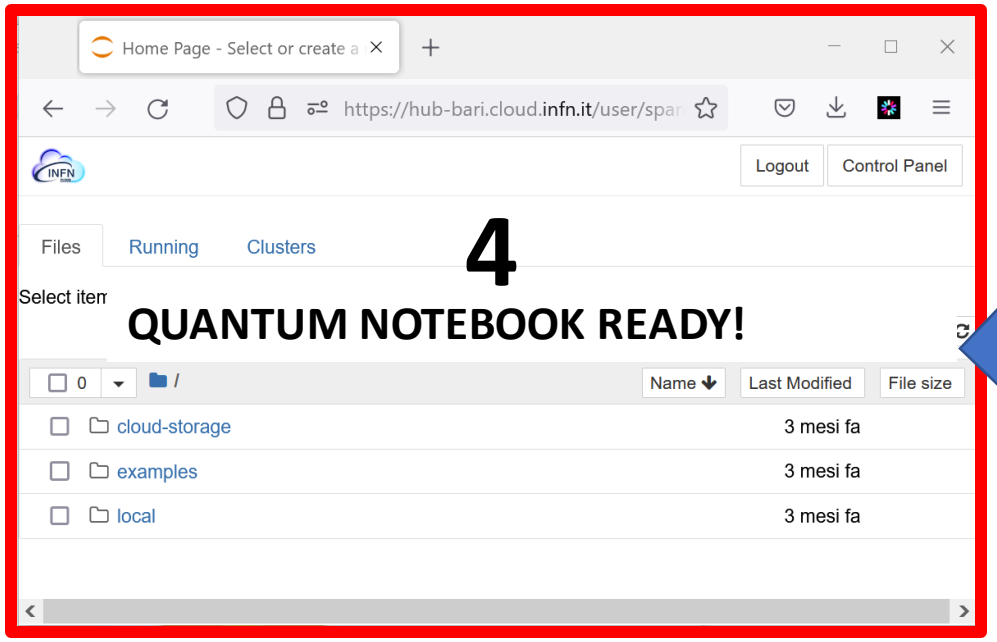


INFN Cloud Image









Conclusions

- A Quantum Notebook as been created with two possibile deployment:
  - Standalone Docker Image
  - Quantum Notebook as a service over INFN Cloud
- The docker image contains the latest versions of the some of the most used Quantum libraries however it is very easy to exented for additional needs .
- Possibile usage: Researchers, Students, preparation of tutorials, summer schools etc.