



A.D. 1308

unipg

UNIVERSITÀ DEGLI STUDI
DI PERUGIA

Uninuvola: il portale di calcolo dell'Università di Perugia

Casi d'uso

Marco Pezzella (marco.pezzella@unipg.it)

Indici degli argomenti

- I. Introduzione (3-5)
- II. Le funzioni base (6-9)
- III. Casi d'uso: immagini presenti (10-20)
- IV. Casi d'uso: immagini personalizzate (21-24)
- V. Conclusioni (25-27)

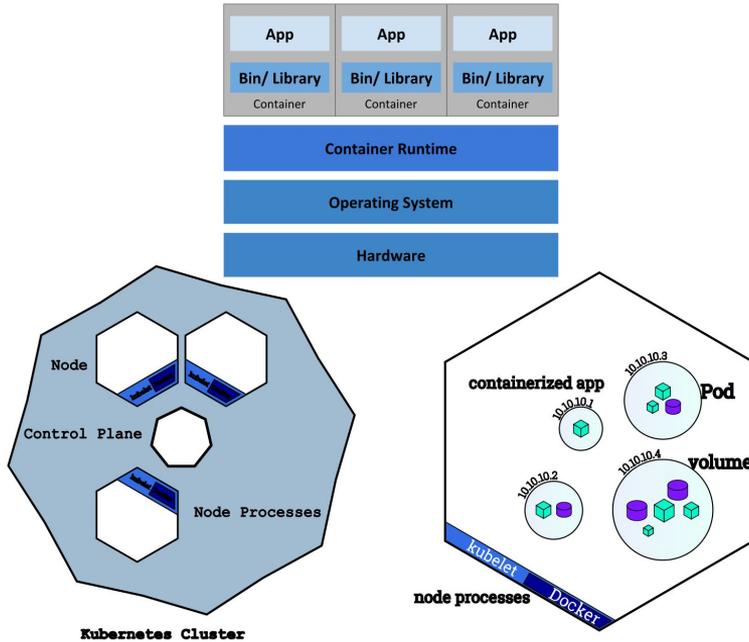
Lo scopo del progetto

Lo scopo è creare il prototipo d'infrastruttura federata basata su soluzioni tecnologiche all'avanguardia e open source.

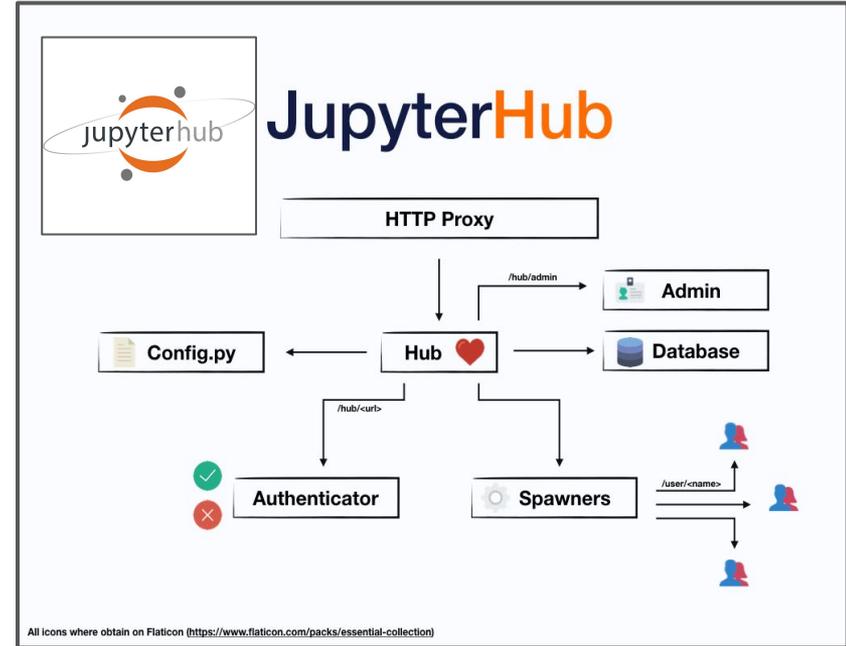
L'esperienza acquisita sarà utilizzata per il futuro **C-Lab**.



L'infrastruttura

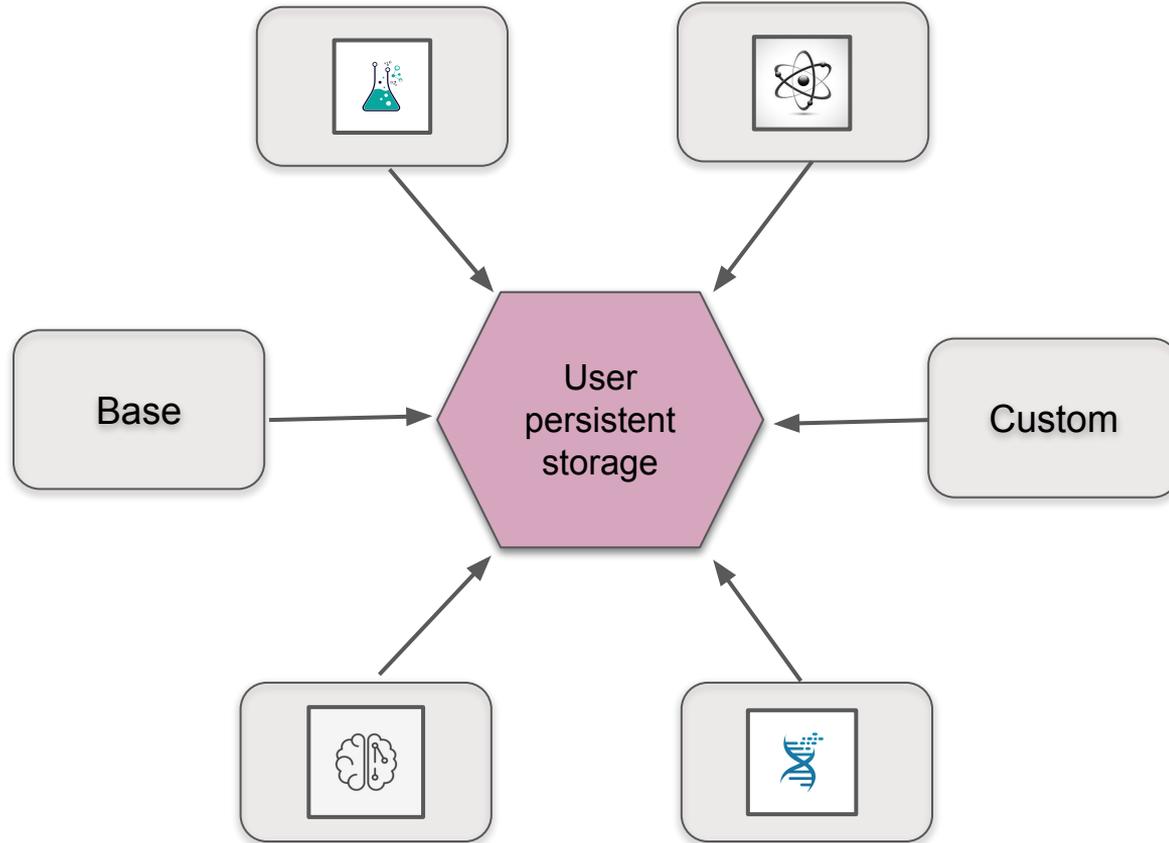


Kubernetes è usato per il deployment e la gestione delle applicazioni containerizzate.



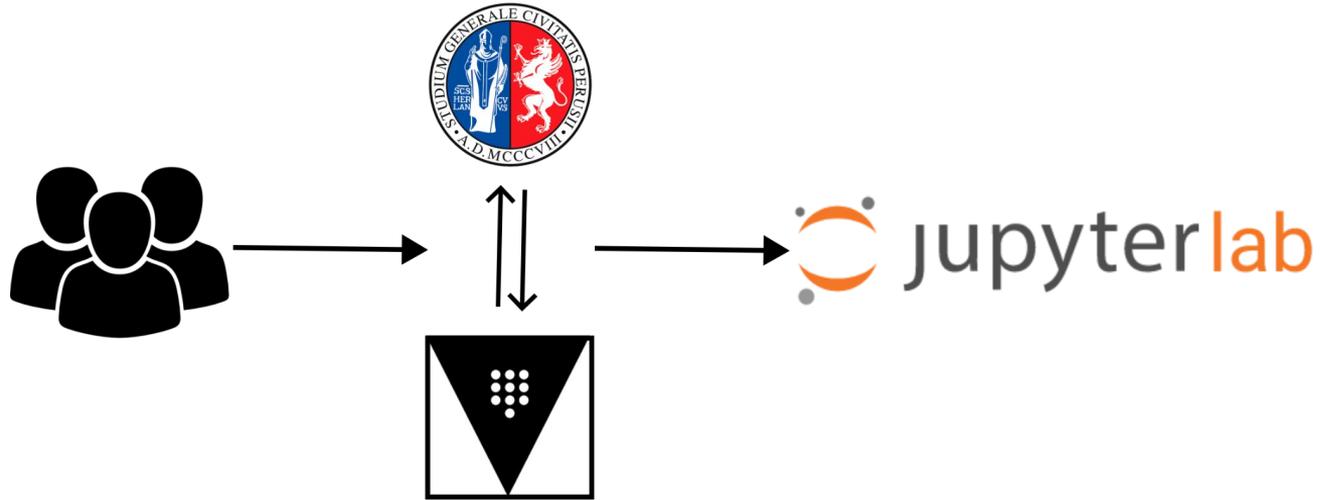
JupyterHub consente di gestire ambienti di notebook Jupyter multi-utente.

Storage



Le funzioni base

Login e scelta delle immagini

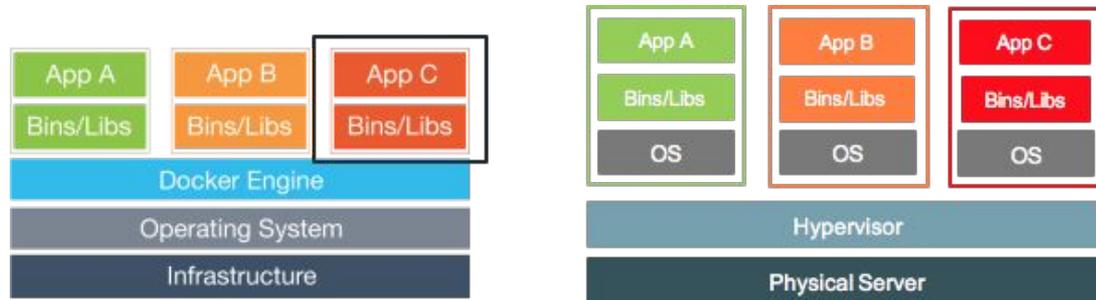


<https://uninuvola.fisgeo.unipg.it>

Immagini e container

Un **immagine** è un pacchetto binario che viene istanziato in un **container**, un ambiente isolato dove le applicazioni possono essere eseguite.

Un **container** è più leggero di una VM e condivide lo stesso kernel del SO che lo ospita.
Risolve problemi di **riproducibilità** e **retrocompatibilità**.

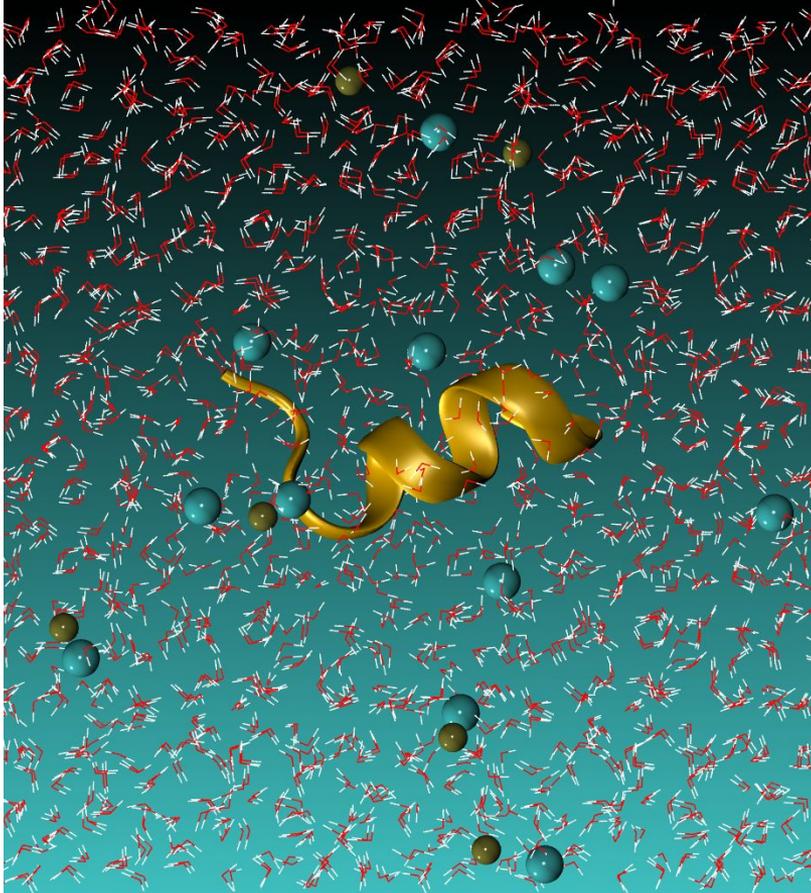


Ricapitolando

- I. JupyterHub può essere usato come terminale e come notebook.
- II. GUI possono essere usate con Xpra
- III. Possibilità di includere cloud storage (in progress)
- IV. Potete passare da Colab a Jupyter facilmente, [per esempio...](#)

Casi d'uso I: immagini presenti

Computational Chemistry



Allo stato attuale, l'immagine offre una serie di tools per simulazioni di **dinamica molecolare**.

Useremo una [protetina antimicrobiotica della rana cinese](#) come nostro modello.

Next steps: introduzione di codici più pesanti chiamati da dentro l'immagine

Un'immagine, vari codici

NAMD3

Eseguitibile compilato con OpenMP, ottimizzato per simulazioni.

Compatibile con GPU.

AmberTools 23

Utile per analisi e piccole simulazioni.
Installato tramite conda, single core.

Installabile localmente in parallelo

DL_POLY

Programma per MD compilato con MPI.

Alto livello di customizzazione.

OpenMM

Serie di librerie per simulazioni con linguaggi differenti.

Implementazione in conda

Analisi dei risultati

Possiamo installare strumenti di analisi all'interno dell'immagine, come **MDTRAJ**. Vediamo come funziona con codici differenti.

Misura del **$g(r)$** : quanto è probabile trovare la molecola A vicino a B?

Misura del **RMSD**: quanto varia la struttura durante la dinamica?

Misura della **lunghezza della proteina**: quanto distano le due estremità?

```
Code
import matplotlib.pyplot as plt
import mdtraj as md
import numpy as np

def measure_head_tail_distance(traj, topology):
    # Identify the head (N-terminus) and tail (C-terminus) atoms
    n_terminus = topology.select('name N and resid 0')
    c_terminus = topology.select('name C and resid 12')

    if len(n_terminus) == 0 or len(c_terminus) == 0:
        raise ValueError("Couldn't identify N-terminus or C-terminus atoms.")

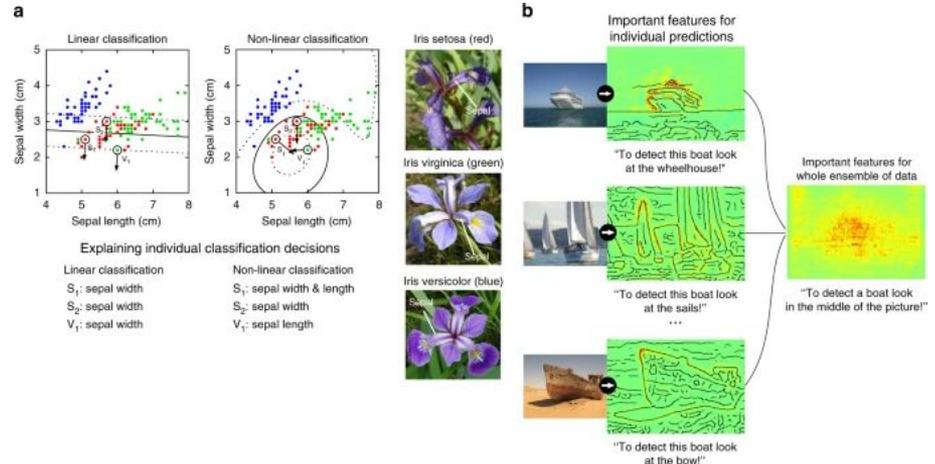
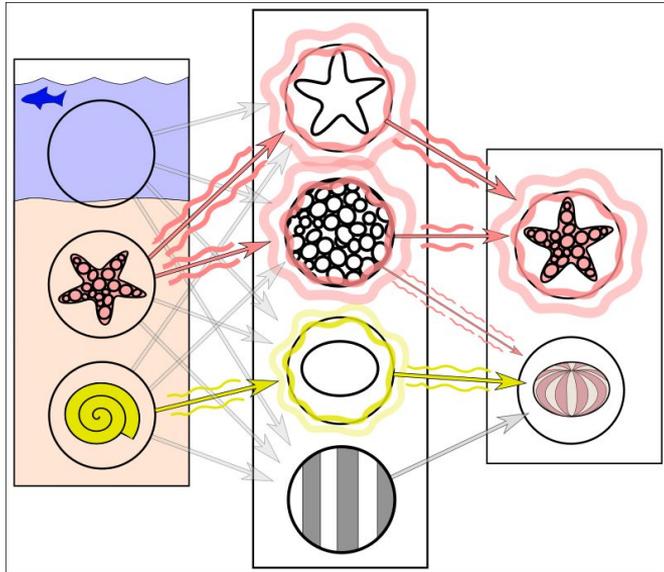
    # Calculate distances between the head and tail atoms for each frame
    distances = np.empty(traj.n_frames)
    for i in range(traj.n_frames):
        distance = md.compute_distances(traj[i], np.array([[n_terminus[0], c_terminus[0]]]))
        distances[i] = distance[0]

    return distances
```

Machine Learning

Il **machine learning** è lo studio di algoritmi statistici che apprendono dai dati disponibili e generalizzano per esplorare nuovi dati. Vengono utilizzati per problemi di classificazione e predizione.

Le **reti neurali** sono ispirate alla struttura del cervello e possono eseguire azioni complesse. Il loro vantaggio risiede nell'apprendimento da analisi precedenti e dall'esplorazione di variabili nascoste.



Sklearn: classificazione vini



Input variables:

- 1 - fixed acidity
- 2 - volatile acidity
- 3 - citric acid
- 4 - residual sugar
- 5 - chlorides
- 6 - free sulfur dioxide
- 7 - total sulfur dioxide
- 8 - density
- 9 - pH
- 10 - sulphates
- 11 - alcohol

Output variable:

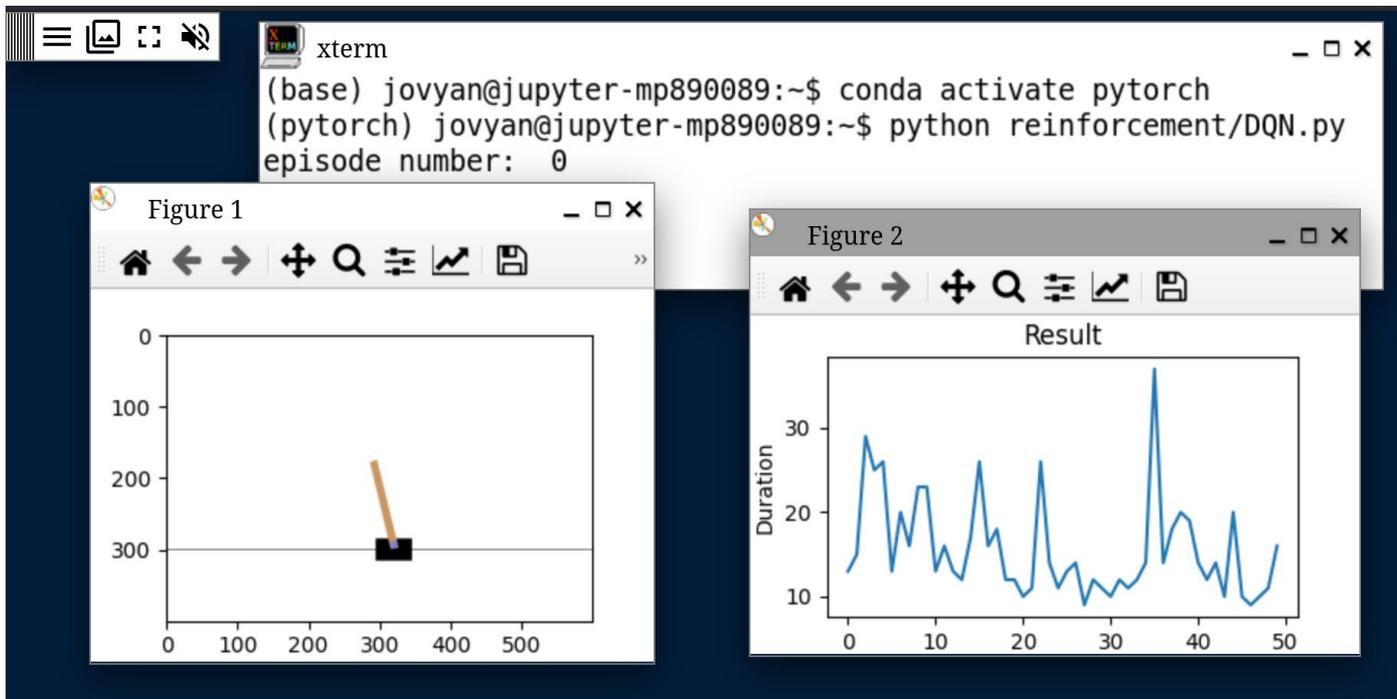
- 12 - quality (score between 0 and 10)



PyTorch: Apprendimento Rinforzato

Abbiamo un carrello, legato ad un palo, che può muoversi lateralmente.

L'algoritmo DQN salva le transizioni osservate e le riutilizza negli step successivi.



Computational Physics

OpenFOAM è un software open-source per la fluidodinamica. Le nuove versioni sono rilasciate ogni sei mesi.

ParaView è un software open-source per processare e visualizzare dati scientifici, utilizzabile con OpenMP e MPI.

Paraview

The image shows the Paraview software interface. The main window displays a 3D visualization of a sphere with streamlines. The streamlines are colored according to a velocity magnitude scale, ranging from 0.0e+00 (blue) to 2.2e+01 (red). The sphere is also colored according to a 'cst3' scale, ranging from 1.0e-01 (blue) to 1.3e-01 (red). The interface includes a Pipeline Browser on the left, a Properties panel at the bottom left, and a RenderView window on the right. The Pipeline Browser shows a sequence of filters: Wavelet1, disk_out_ref.ex2, Contour5, disk_out_ref.ex2, Contour4, Glyph1, StreamTracer1, and WarpByVector1. The Properties panel shows the StreamTracer1 filter settings, including the Integrator Type (Runge-Kutta 4), Streamline Parameters (Maximum Streamline Length: 4.2336), Seed Type (Line), and Line Parameters (Length: 25.9022, Show Line checked). The Line Parameters section includes a table for Point1 and Point2 coordinates:

Point	X	Y	Z
Point1	-5.75	-5.75	-10
Point2	5.75	5.75	10.16

A note below the table reads: "Note: Use 'P' to place alternating points on mesh or 'Ctrl+P' to snap to the closest mesh point. Use '1'/'Ctrl+1' for point 1 and '2'/'Ctrl+2' for point 2." The interface also features a menu bar (File, Edit, View, Sources, Filters, Extractors, Tools, Catalyst, Macros, Help), a toolbar, and a Threshold Table window.

Quantum computing

Ambiente pensato per poter svolgere calcoli con il computer quantistico e con vari simulatori

ocean

Contiene le librerie necessarie per utilizzare gli strumenti sviluppati da D-Wave.

qiskit

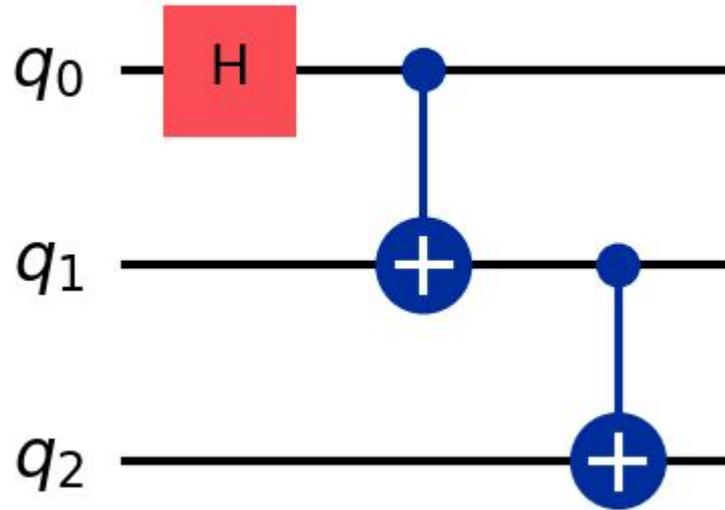
Contiene le librerie necessarie per utilizzare i vari strumenti sviluppati da IBM.

spinqit

La libreria contiene i pacchetti necessari per connettersi al computer quantistico Triangulum e per eseguire simulazioni da remoto.

Esempio: lo stato GHZ

$$|\text{GHZ}\rangle = \frac{|000\rangle + |111\rangle}{\sqrt{2}}$$



Casi d'uso II: immagini personalizzate

Accedere a immagini personalizzate

Può essere utile condividere un'immagine fra ricercatori, oppure testare immagini prima di pubblicarle.

Nuove immagini JupyterHub possono essere caricate da [Docker Hub](#).

Server Options

- UniNuvola Base
- Computational Chemistry
- Machine Learning
- Computational Physics
- Quantum Computing

● Other images

Select a custom image

Image

Other...

Custom image

Start

marcpezz / genomics2

Contains:
Image

Last pushed: 1 day
ago

☆ 0

↓ 2

Public

Scout inactive

marcpezz / genomics1

Contains:
Image

Last pushed: 2
days ago

☆ 0

↓ 3

Public

Scout inactive

marcpezz / exomol_compiled

Contains:
Image

Last pushed: 8
days ago

☆ 0

↓ 6

Public

Scout inactive

Come scrivere un'immagine

```
FROM jupyterhub/k8s-singleuser-sample:3.2.2-0.dev.git.6456.hf11f8ea8
MAINTAINER Mirko Mariotti <mirko.mariotti@unipg.it>
```

```
USER root
```

```
RUN apt-get update; apt-get install -y --no-install-recommends iproute2 xpra git curl vim
RUN apt-get install -y --no-install-recommends curl
RUN apt-get install -y --no-install-recommends xterm
```

```
COPY Duo.tar.gz exocross.tar.gz dvr3d.tar.gz /app/.
RUN tar -zxvf Duo.tar.gz && tar -zxvf exocross.tar.gz && tar -zxvf dvr3d.tar.gz && rm *.tar.gz
RUN cp Duo/j-duo-v218.v1.x exocross/xcross.exe /usr/bin/.
RUN git clone https://github.com/Beryl-Jingxin/PyExoCross.git #&& pip install -r requirements.txt
```

```
RUN conda update -n base -c defaults conda && \
conda create -n pyexocross python=3.11.5 -y && \
conda run -n pyexocross pip install ipykernel && \
conda run -n pyexocross python -m ipykernel install --name pyexocross --display-name pyexocross
```

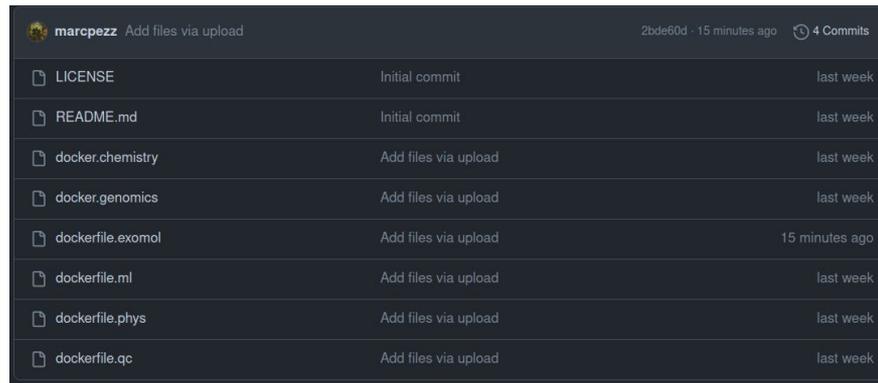
Genera la tua immagine!

Ingredienti:

1. Account [Docker Hub](#)
2. docker su macchina locale
3. Un dockerfile

Il **dockerfile** può essere scritto con un qualsiasi text editor e conterrà le istruzioni per generare l'immagine con **docker**.

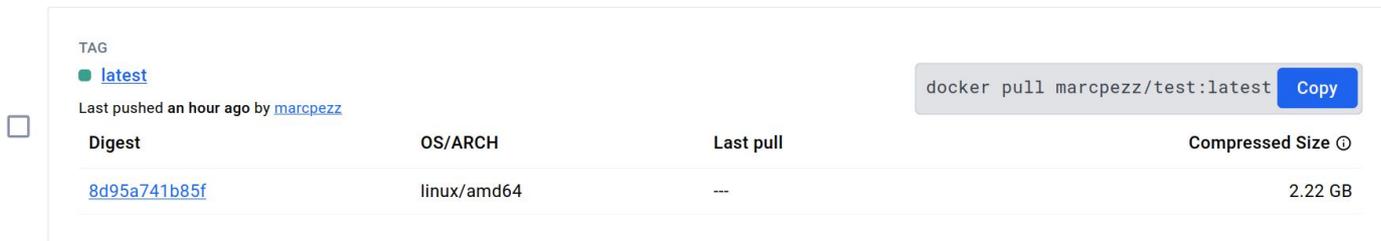
L'immagine deve essere accessibile su **Docker Hub**.



A screenshot of a Docker Hub repository page for the user 'marcpezz'. The page shows a list of files added via upload. The files listed are: LICENSE, README.md, docker.chemistry, docker.genomics, dockerfile.exomol, dockerfile.ml, dockerfile.phys, and dockerfile.qc. Each file entry includes the file name, the commit message (e.g., 'Initial commit' or 'Add files via upload'), and the time since the last commit (e.g., 'last week' or '15 minutes ago').

File Name	Commit Message	Last Commit
LICENSE	Initial commit	last week
README.md	Initial commit	last week
docker.chemistry	Add files via upload	last week
docker.genomics	Add files via upload	last week
dockerfile.exomol	Add files via upload	15 minutes ago
dockerfile.ml	Add files via upload	last week
dockerfile.phys	Add files via upload	last week
dockerfile.qc	Add files via upload	last week

```
~$ docker build -t marcpezz/test:latest . ; docker push marcpezz/test:latest
```

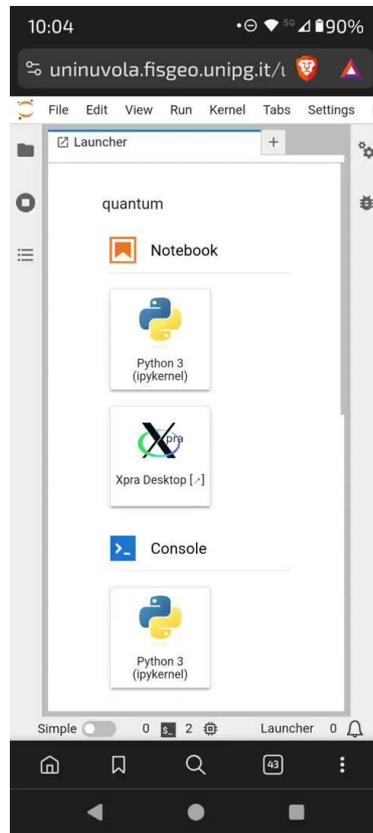


A screenshot of the Docker Hub image page for the repository 'marcpezz/test:latest'. The page shows the image tag 'latest' and a button to 'docker pull marcpezz/test:latest'. Below this, there is a table with columns for Digest, OS/ARCH, Last pull, and Compressed Size. The table shows one entry with Digest '8d95a741b85f', OS/ARCH 'linux/amd64', Last pull '---', and Compressed Size '2.22 GB'.

Digest	OS/ARCH	Last pull	Compressed Size
8d95a741b85f	linux/amd64	---	2.22 GB

Conclusioni

Anche da cellulare!



Per concludere

- I. Il prototipo è operativo e pronto per essere testato.
- II. Abbiamo un manuale (aggiornato al 10/06/2024).
- III. Funzionalità da completare: programmi pesanti & storage remoto.

Cosa potete fare: Utilizzate UniNuvola, fateci sapere se ci sono bug, librerie da installare, immagini particolari, *etc...*

Environment 1: sklearn

L'environment contiene le librerie necessarie per svolgere calcoli con Tensorflow and Scikit learn.
L'esempio scelto è quello della classificazione dei fiori.

Non siamo interessati alle prestazioni (al momento).



roses



dandelion



tulips



sunflowers



roses



sunflowers



dandelion



sunflowers



roses

