Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

INFN

ICSC
Centro Nazionale di Ricerca in HPC,
Big Data and Quantum Computing

Instructions for offloading to RAC resources

Tommaso Tedeschi (INFN Perugia)

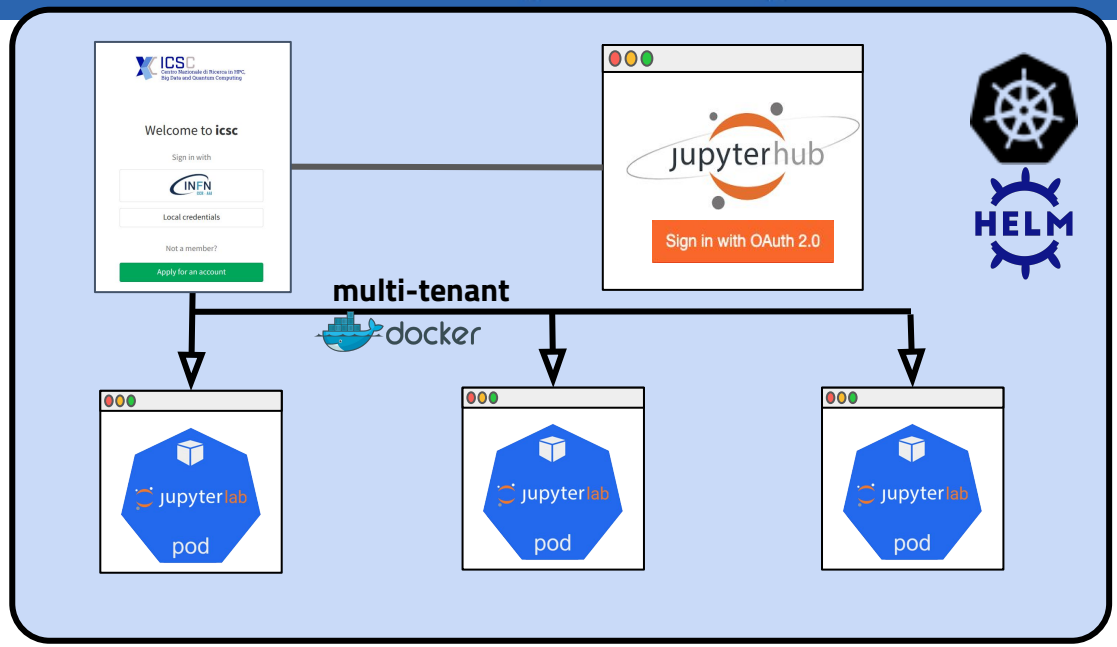
WP2 UC2.2.2 Meeting - 17 July 2025 - ZOOM

The high-rate analysis hub

A jupyterhub is deployed on a k8s cluster (**128 vCPUs and 258 GB**) via the official Spoke2 JHub Helm repo

- endpoint is here and Indigo-IAM is used for authentication

Users choose JupyterLab image to deploy and after deployment completion, they get access to a full IDE (persistent storage, terminal, notebooks, editors)



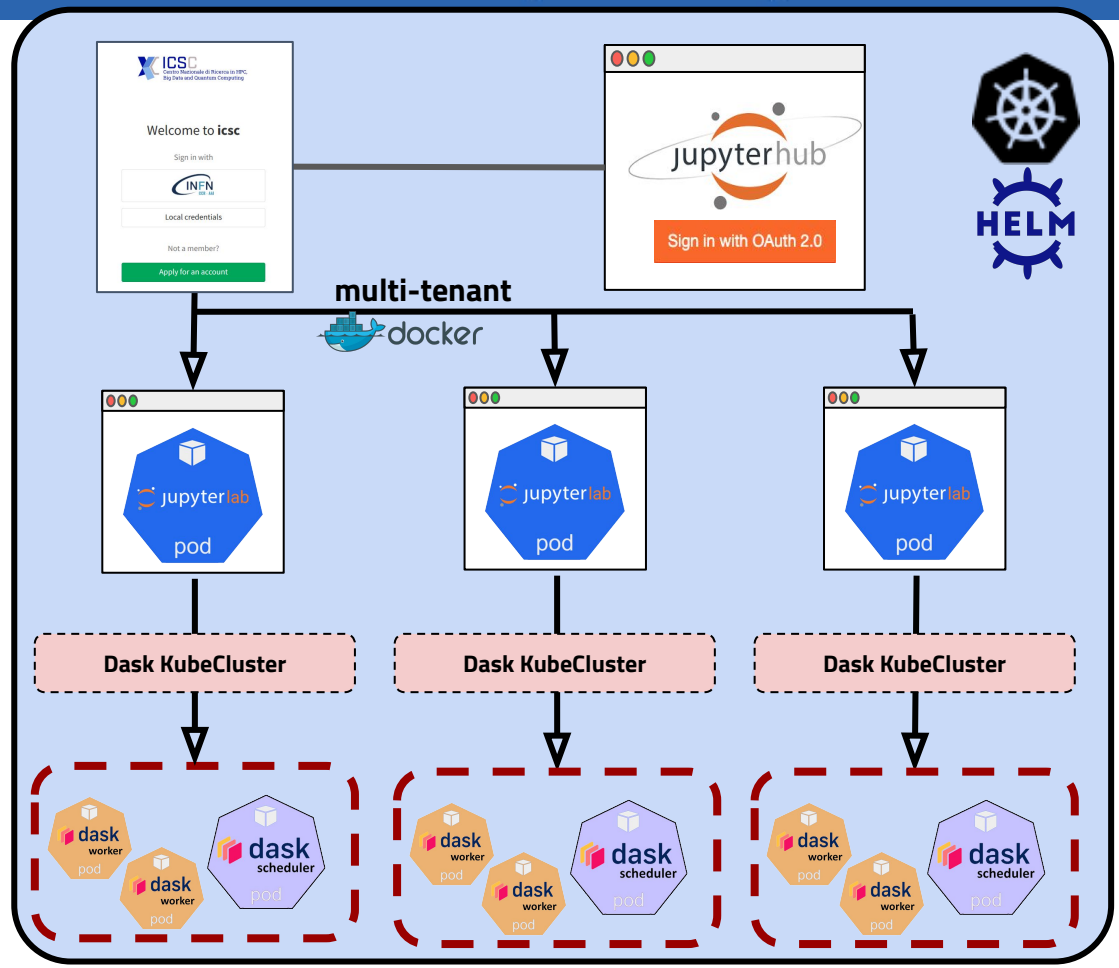
How we used to scale

A jupyterhub is deployed on a k8s cluster (**128 vCPUs and 258 GB**) via the official Spoke2 JHub Helm repo

- endpoint is here and Indigo-IAM is used for authentication

Users choose JupyterLab image to deploy and after deployment completion, they get access to a full IDE (persistent storage, terminal, notebooks, editors)

Users used to scale up their python-based computation, using Dask library, within the Kubernetes cluster



Where we are now

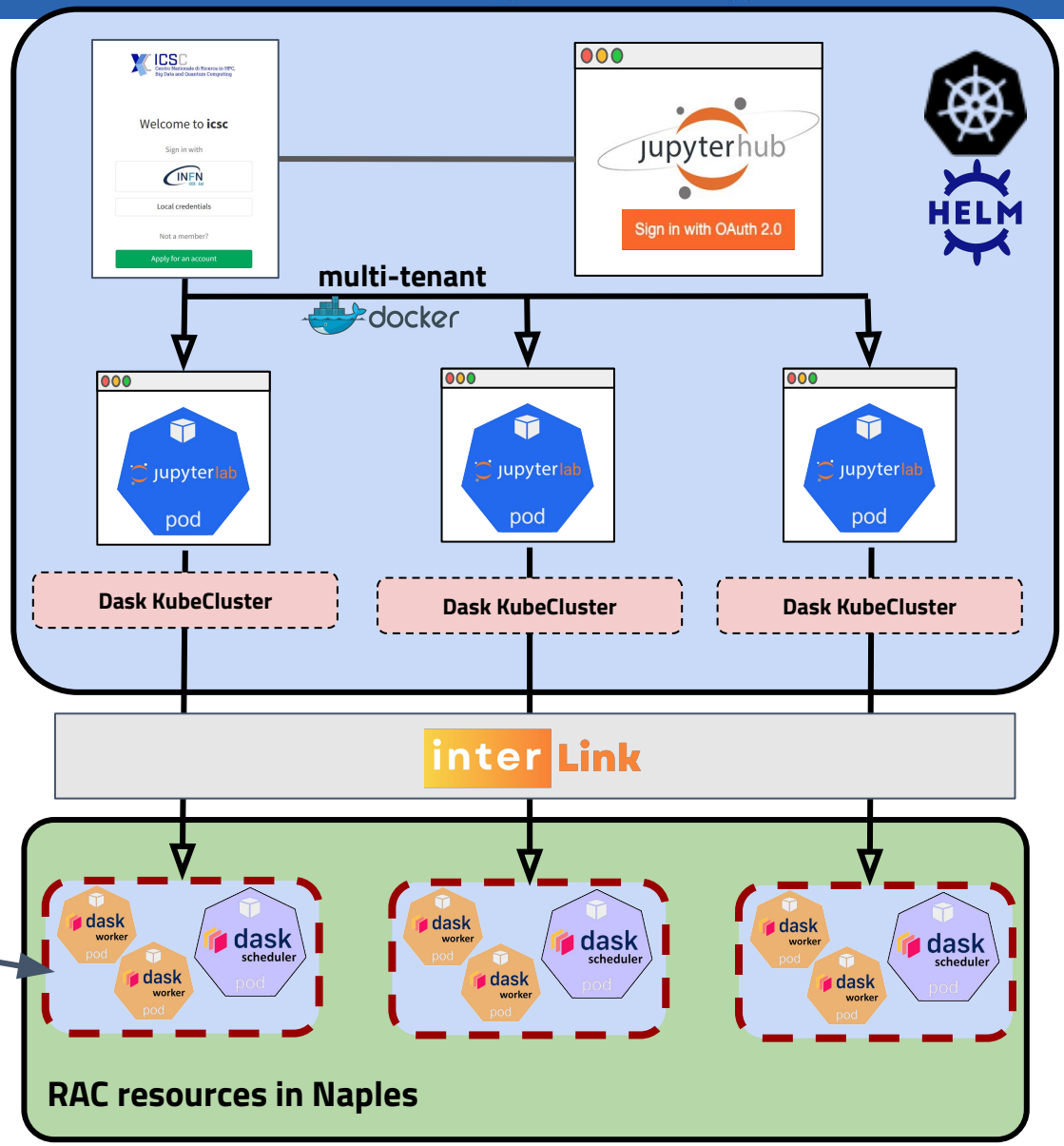
A jupyterhub is deployed on a k8s cluster (**128 vCPUs and 258 GB**) via the official Spoke2 JHub Helm repo

- endpoint is here and Indigo-IAM is used for authentication

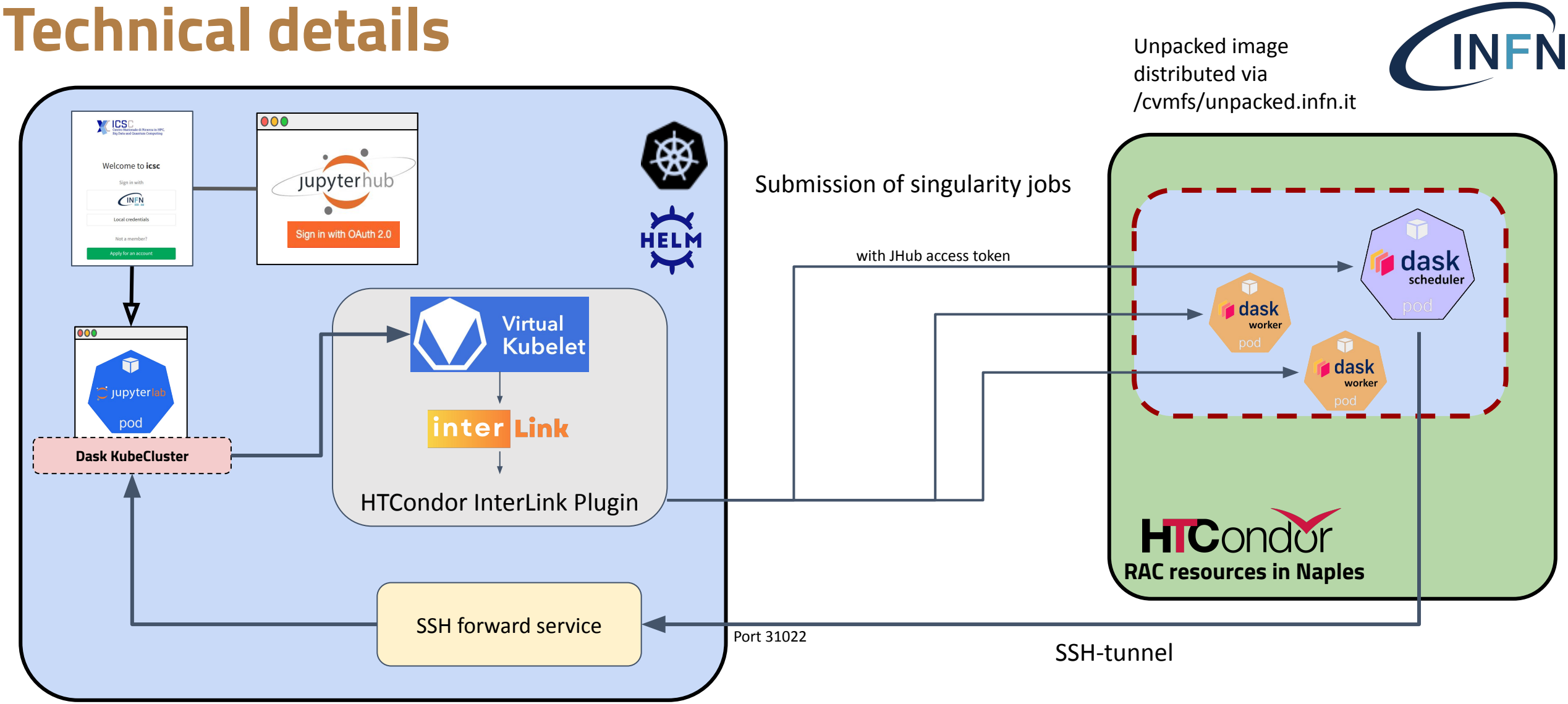
Users choose JupyterLab image to deploy and after deployment completion, they get access to a full IDE (persistent storage, terminal, notebooks, editors)

Users can now scale up their python-based computation, using Dask library, offloading to external RAC resources hosted in Naples:

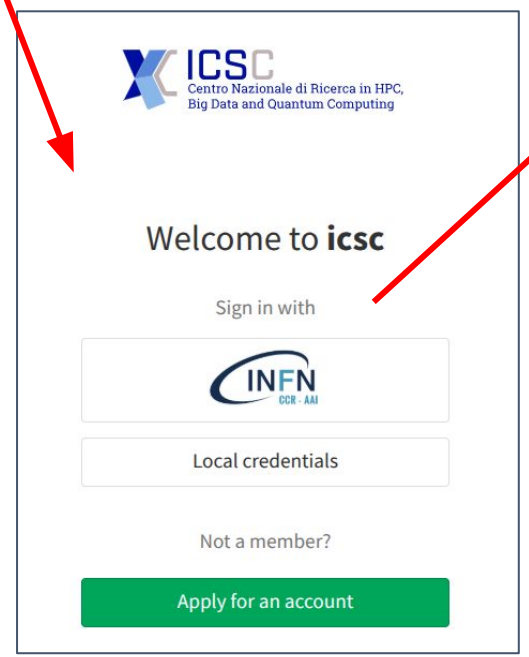
- 70 nodes (96 cores each)
- accessible via an HTCondor-CE

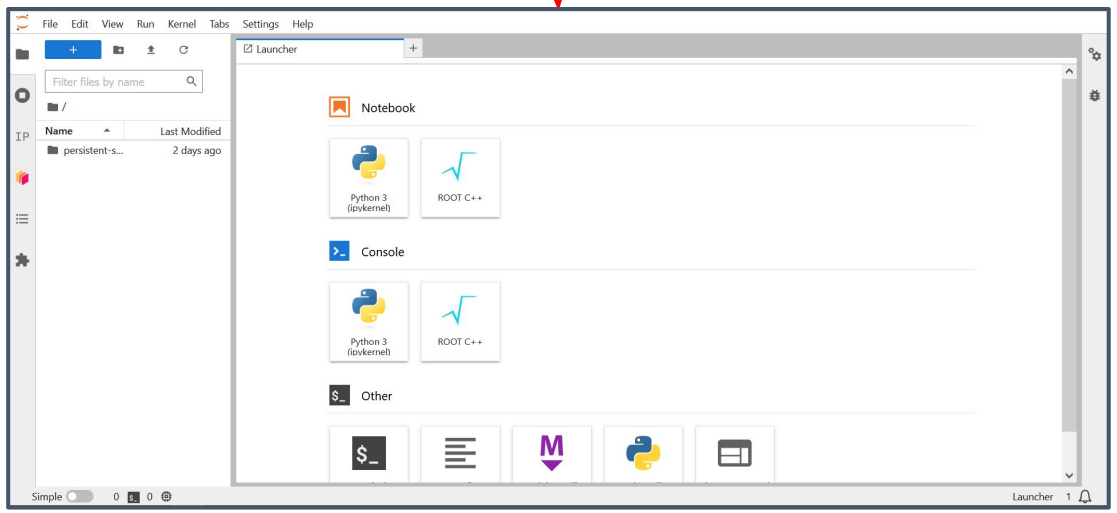


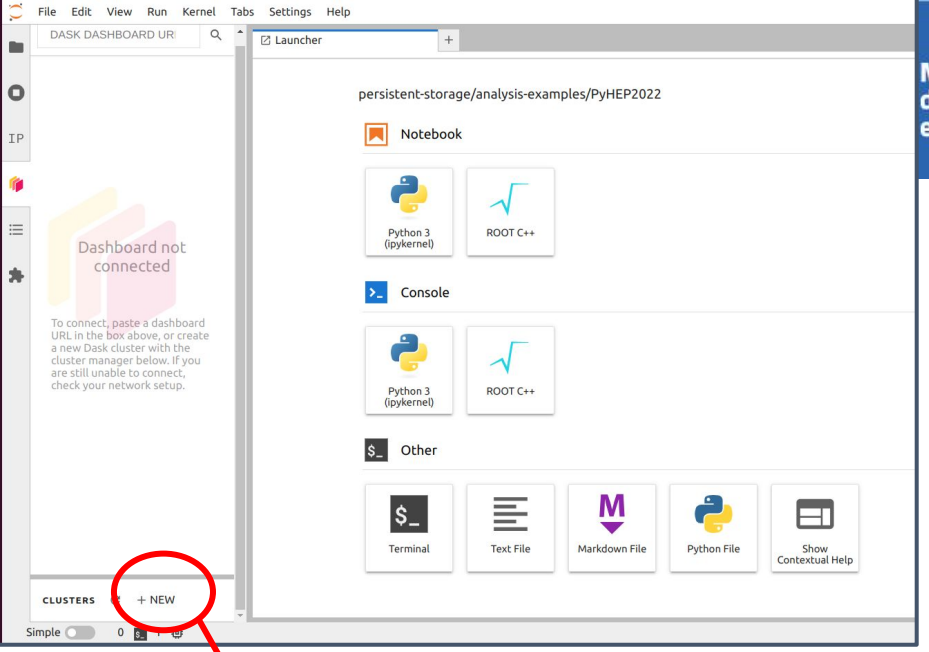
Technical details

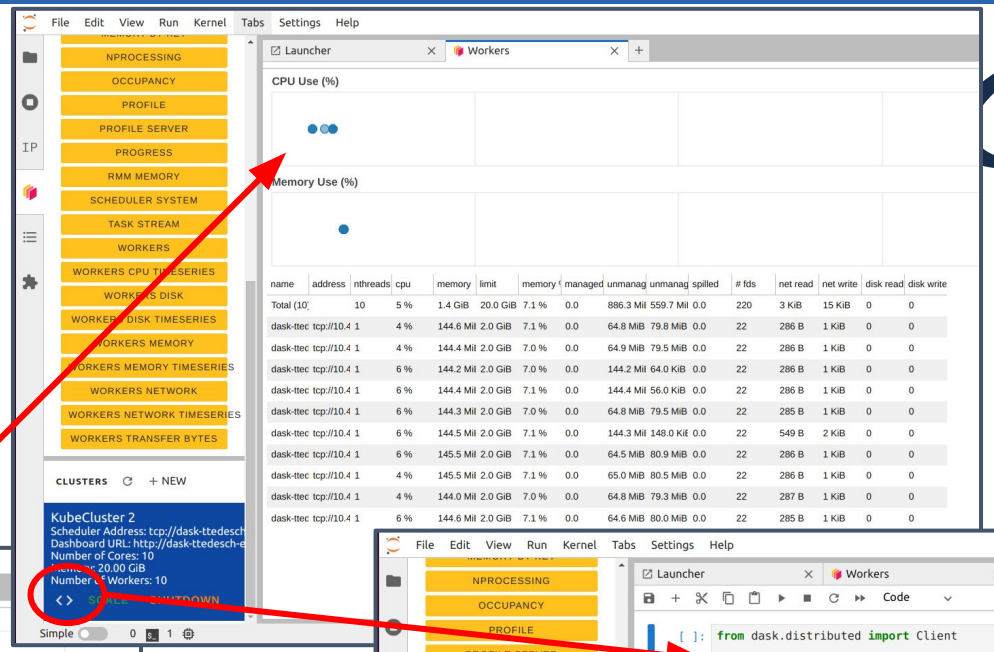


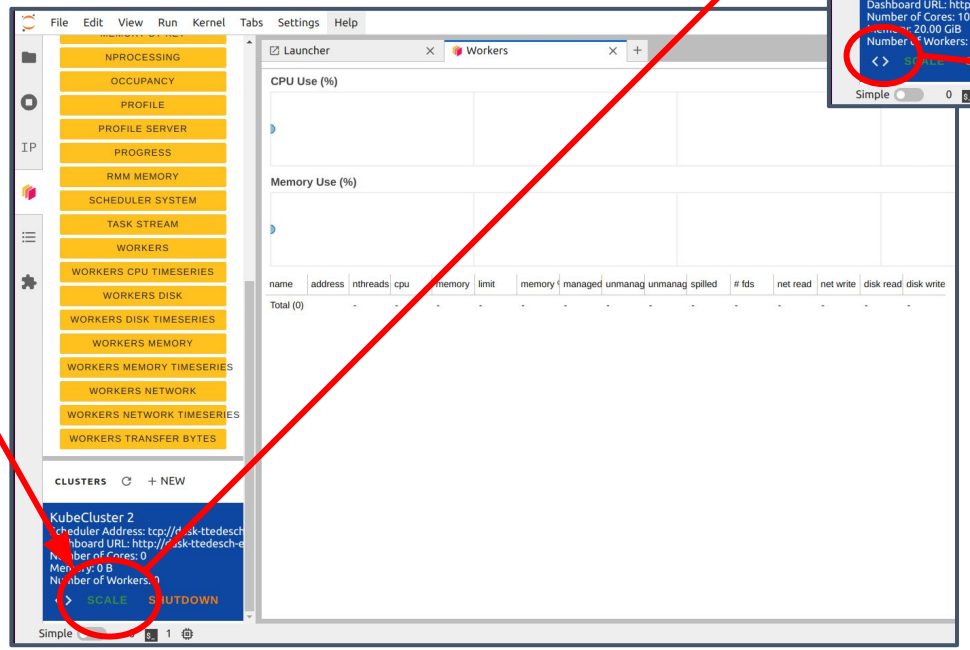
Access steps

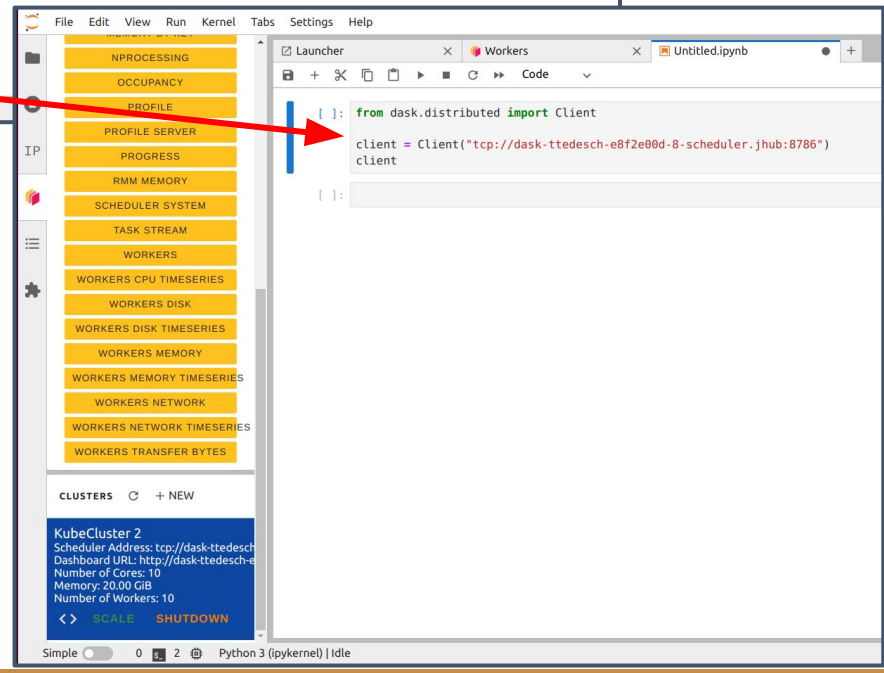












IMPORTANT:
some delay may occur due to the job queueing system. Once connected, please wait ~10s before scaling the cluster

Some additional info

INFN

- cvmfs is mounted both in jlab session and in the nodes (unpacked, sft, cms, grid for now)
- Repo where images are developed: https://github.com/ICSC-Spoke2-repo/wp5-custom-images/tree/highrate-offloading
- As before, **each user is assigned 10 GB of persistent storage**:
  - Anything that is stored outside the `persistent-storage` will be lost when the JLab session ends
  - No redundancy nor any production backup is in place for persistent storage: **BACKUP EVERYTHING ELSEWHERE!** since data will be lost in case of a node failure

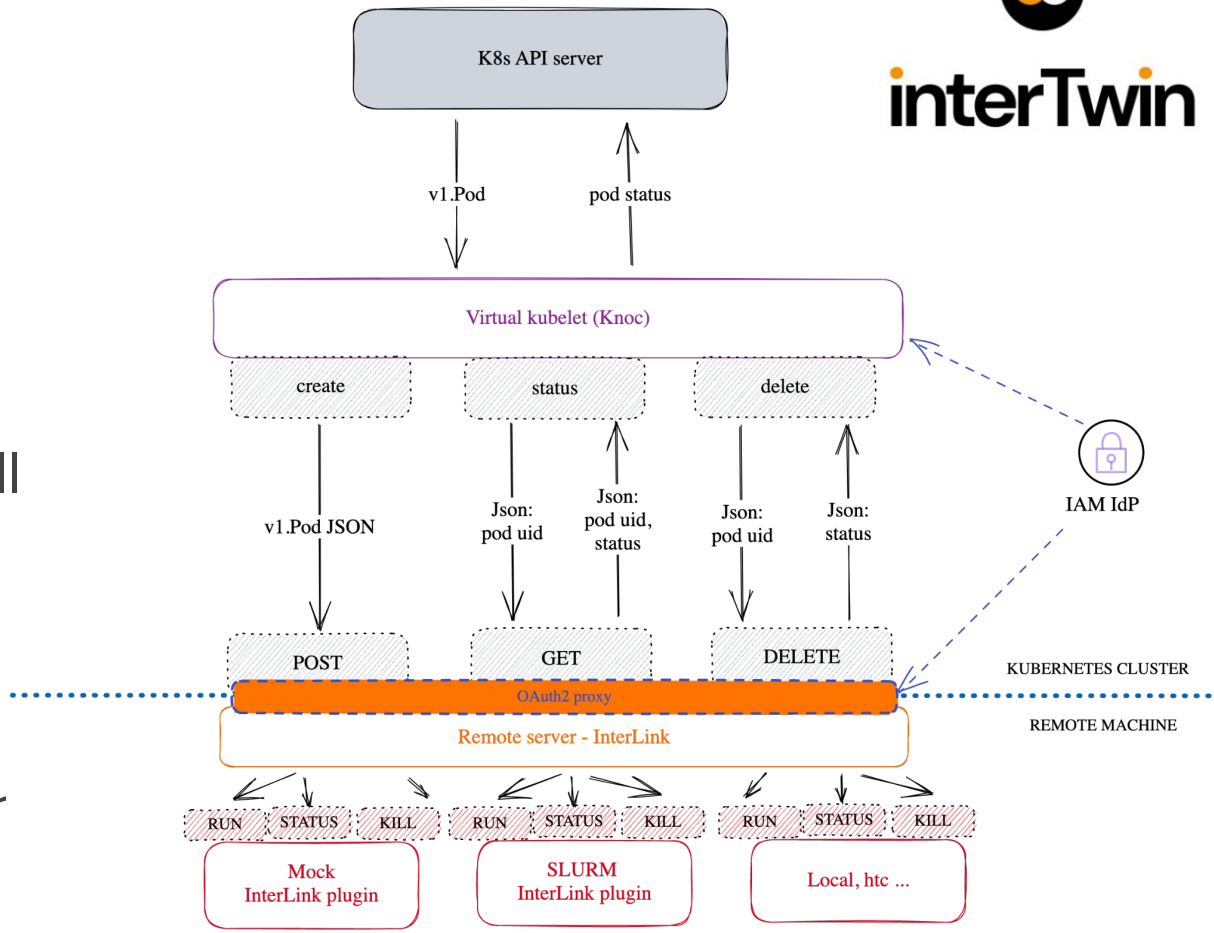
Backup

The technical solution: InterLink

InterLink aims to provide an abstraction for the execution of a Kubernetes pod on any remote resource capable of managing a container execution lifecycle.

The project consists of two main components:

- **A Kubernetes Virtual Node**: based on the VirtualKubelet technology. Translating request for a kubernetes pod execution into a remote call to the interLink API server.
- **The interLink API server**: a modular and pluggable REST server where you can create your own container manager plugin (called sidecar), or use the existing ones: remote docker execution on a remote host, singularity Container on a remote SLURM or **HTCondor batch system**, etc...

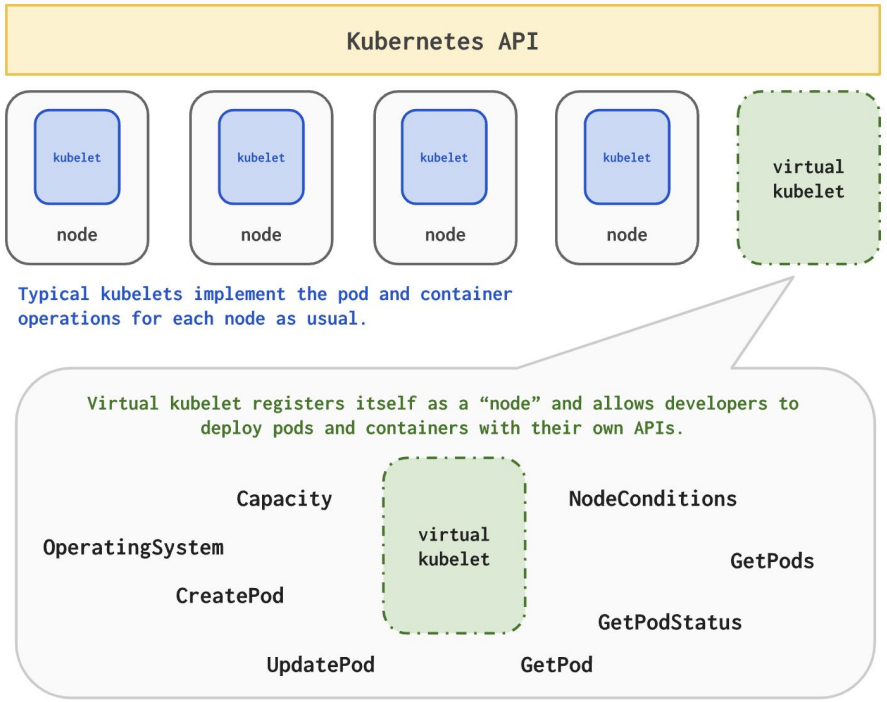


https://github.com/interTwin-eu/interLink

Components: VK

https://virtual-kubelet.io/

- **Virtual kubelet (VK)**:
  - "Open-source Kubernetes kubelet implementation that masquerades as a kubelet. This allows Kubernetes nodes to be backed by Virtual Kubelet providers"
- Can be imagined as a translation layer:
  - "I take your pod and run your container wherever I want"
- Registers virtual node and pulls work to run
- The pod lifecycle is managed via interlink rest calls
- Oauth2 via service token kept "refreshed"



Components: Interlink + Oauth2 proxy

INFN

- Oauth2 proxy: authN with IAM and authZ configurable on aud and groups
- "Digests" and manipulates calls from VK to the sidecar
- Self contained binary, distributable on all OS without dependencies