# INTERACTIVE HPC WITH JUPYTERLAB

## JSC HPC-Support-Corner

2025-05-28 | JENS HENRIK GÖBBERT (J.GOEBBERT@FZ-JUELICH.DE)
TIM KREUZER (T.KREUZER@FZ-JUELICH.DE)

## WebUIs

JupyterLab

VSCode

R Studio

Xpra Desktop

MATLAB Web

NEST Desktop

more to come

## Topics-specific

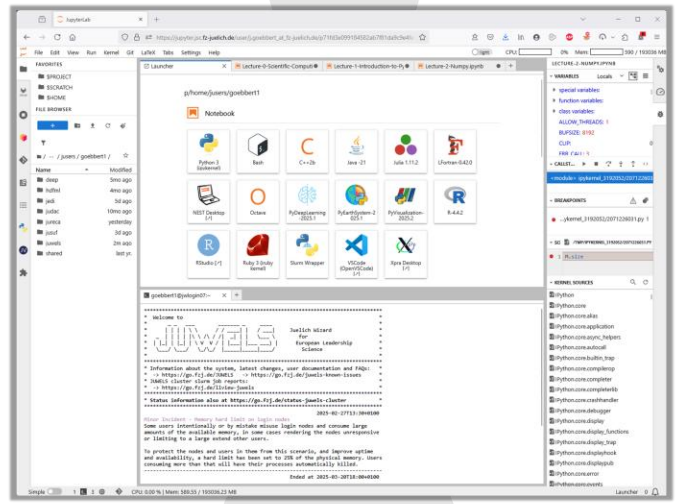
DeepLearning

EarthSystems

Visualization

JUNIQ

# Increase Productivity



## Languages

Python

C++ (Cling)

LFortran

Julia

Java

Ruby

R

Octave

Bash

MATLAB

LaTeX

more to come

| JUNIQ | JSC Cloud | JUST Storage | JUWELS Cluster | JUWELS Booster | JURECA-DC | JUPITER |
|---|---|---|---|---|---|---|
|  |  |  |  |  |  |  |
| **Quantum Computing Platform** | 12,384 vCPUs | 14750 HDD + 48 NVMe | 122,768 cores Intel Skylake | 44,928 cores AMD EPYC Rome | 98,304 cores AMD EPYC Rome | **1st European Exascale Supercomputer** |
| JUQCS, Qaptiva 800 | NVIDIA 16x V100, 3x A100 | **= 211 PByte** | 224 NVIDIA V100 GPUs | 3,744 NVIDIA A100 GPUs | 768 NVIDIA A100 GPUs | **> 1 EFlops** |
| D-Wave, PASCAL, QSolid, | 14 TiB memory, 1 TiB NVMe | + | 275 TByte memory | 629 TByte memory | 443 TByte memory | |
| OpenSuperQplus | **OpenStack Cluster** | 43.000 Tapes | **12.27 PFlops** | **73.02 PFlops** | **18.51 PFlops** | |
| | | **= 478 PByte** | | | | |

# JUPYTER NOTEBOOK

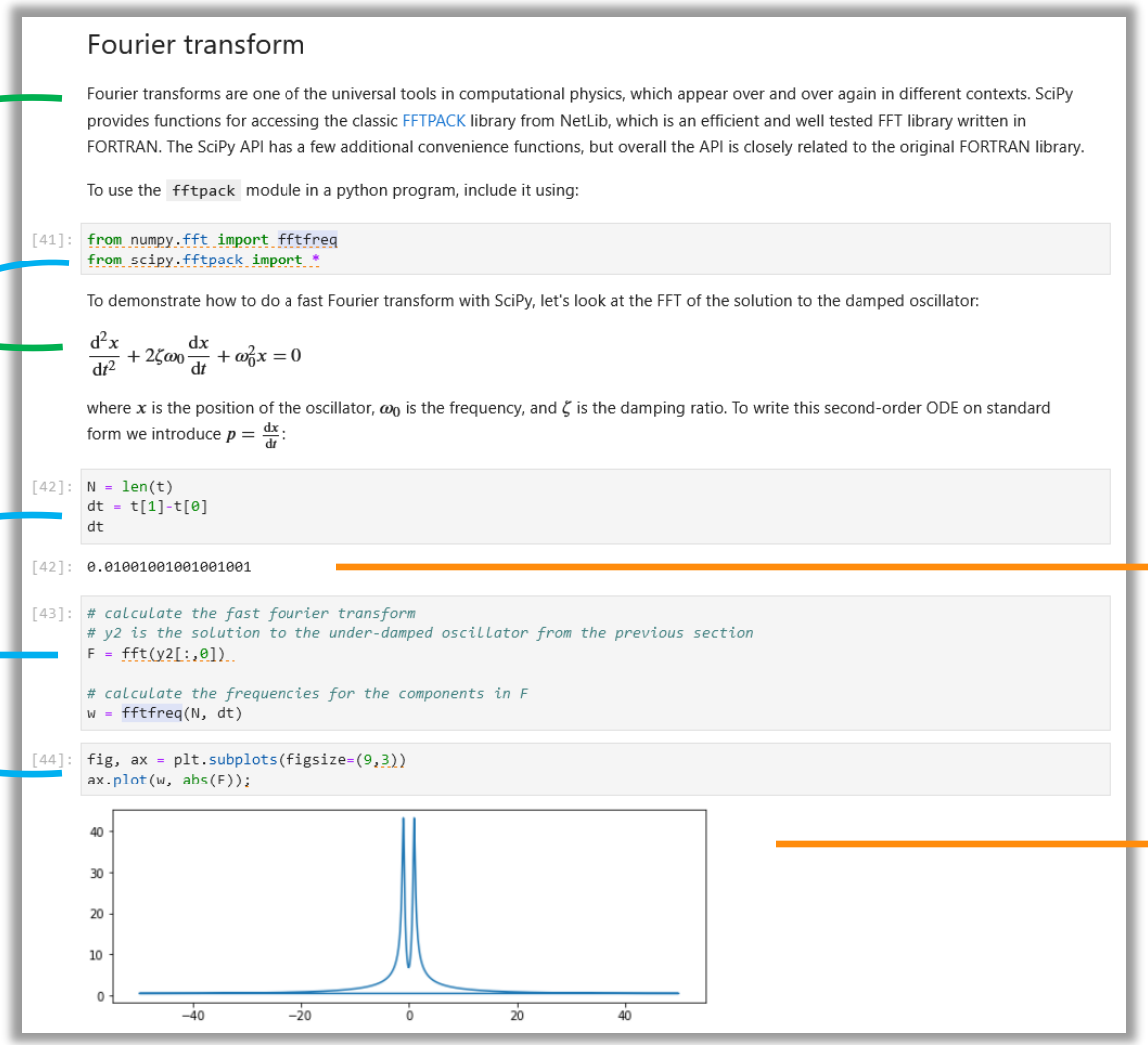
## creating reproducible computational narratives

Markdown Cells

Code Cells

Output

Output

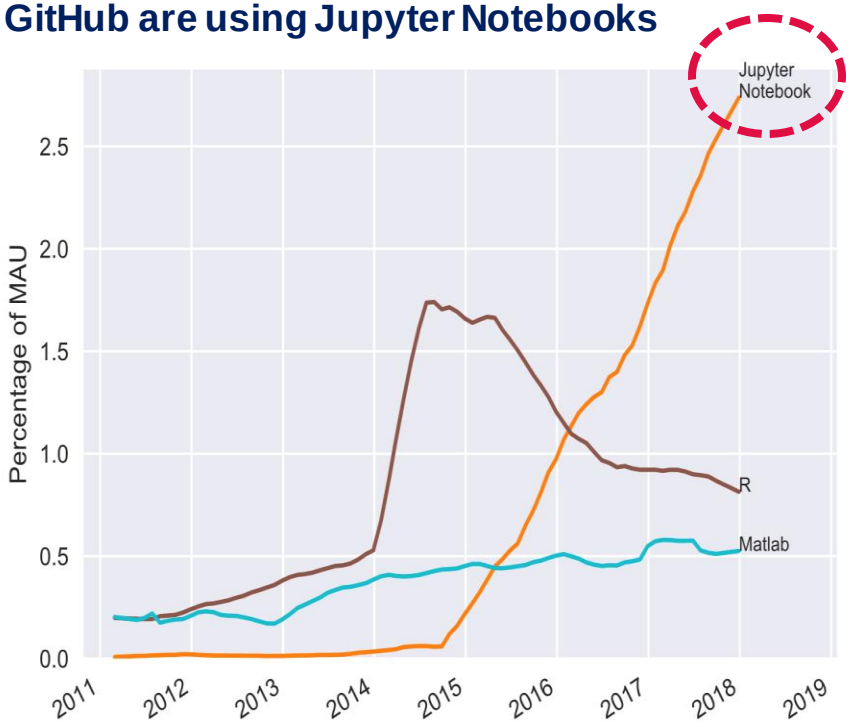


# MOTIVATION

## Rise of Jupyter´s popularity

- In 2007, Fernando Pérez and Brian Granger announced
  „**Ipython**: a system for interactive scientific computing“ [1]
- In 2014, Fernando Pérez announced
  a spin-off project from IPython called **Project Jupyter**.
  - IPython continued to exist as a Python shell and a kernel for Jupyter, while the Jupyter notebook moved under the Jupyter name.
- In 2015, GitHub and the Jupyter Project announced
  native rendering of Jupyter notebooks file format (.ipynb files) on the **GitHub**
- In 2017, the **first JupyterCon** was organized by O'Reilly in New York City.
  Fernando Pérez opened the conference with an inspiring talk. [2]
- In 2018, **JupyterLab** was announced
  as the next-generation web-based interface for Project Jupyter.
- In 2019, JupyterLab 1.0 …
  In 2020, JupyterLab 2.0 …
  In 2021, JupyterLab 3.0 …
  In 2023, JupyterLab 4.0 …

**Counting how many Monthly Active Users (MAU) on GitHub are using Jupyter Notebooks**




https://www.benfrederickson.com/ranking-programming-languages-by-github-users/
https://github.com/benfred/github-analysis


[1] Pérez F, Granger BE (2007) Ipython: a system for interactive scientific computing. Comput Sci Eng 9(3):21–29
[2] Pérez F, Project Jupyter: From interactive Python to open science -> https://www.youtube.com/watch?v=xuNj5paMuow




JÜLICH Forschungszentrum

# HISTORY OF JUPYTERLAB AT JSC


## Initial Basis

JupyterLab modules
**Authentication** via Unity/IdM
**Authorization** via UNICORE
Orchestration Docker Swarm
Synchronization of User-DBs
Basic Data Protection Regulation
Fulfill Safety Requirements

## Usage

Inplace Dokumentation
R, Julia, C++, Octave, Ruby
JupyterLabs on OpenStack
Dashboard Development
**JupyterLab Usability**
Kernel for Vis, DL
Testing & Benchmarking

## Features

**Remote Desktop** Integration
Optional 2-Factor Auth.
**Use for Workshops**
Specialized Functionalities
Enhanced Data Access
Extended Logging
Cross-Side Demonstration

## Redesign

Switch to **Kubernetes**
Redesign Management
Switch to **JupyterLab 3**
GPFS through UFTP
Support for User Extensions
Easybuild Modularization

## Customization

**Project/Community JHubs**
Upgrade JHub Entrance-UI
Comp. Resource Permissions
Maintenance Improvements
Upgrade of Load Balancer
**Modularization of Backend**
**External Clouds & HPC**




JÜLICH Forschungszentrum

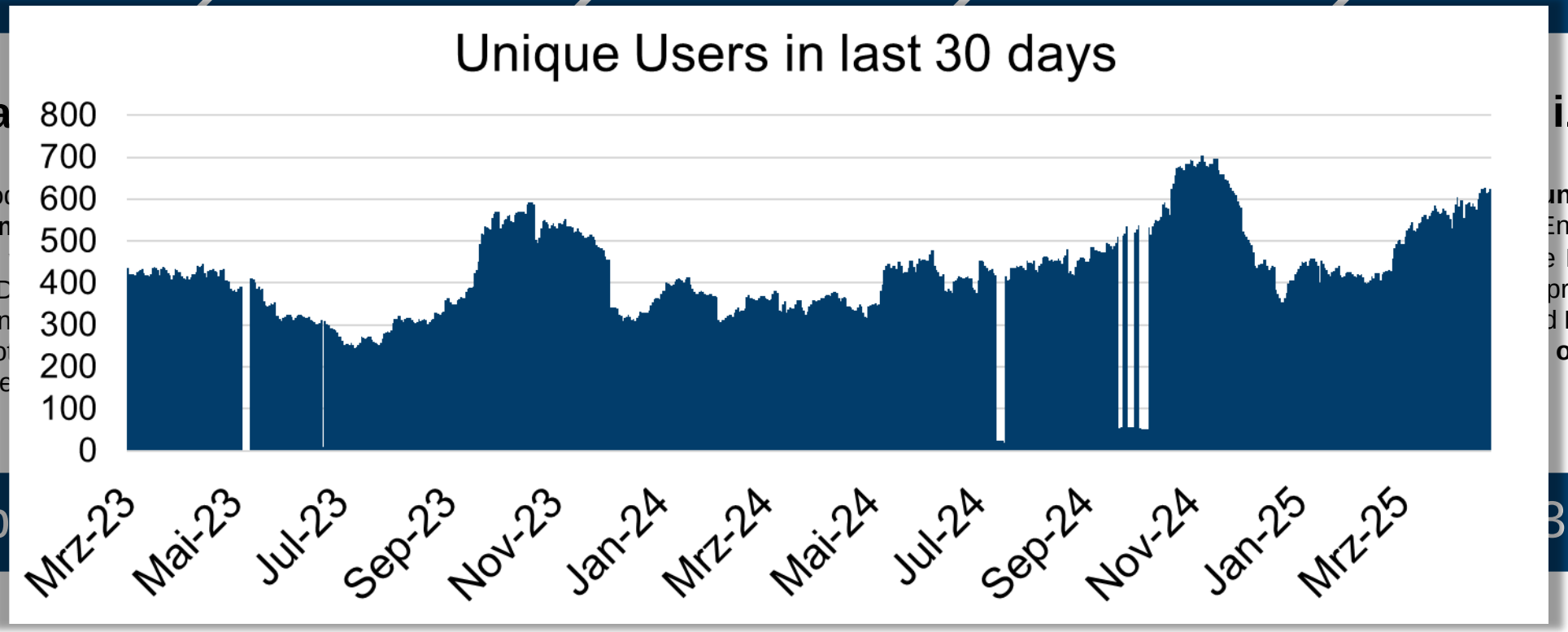
# HISTORY OF JUPYTERLAB AT JSC

2018 2019 2020 2021 2022

**Initia** **ization**

JupyterLab mod **unity JHubs**
**Authentication** Entrance-UI
**Authorization** Permissions
Orchestration D provements
Synchronization d Balancer
Basic Data Pro **of Backend**
Fulfill Safety Re

JLab B+X






JÜLICH Forschungszentrum

# TERMINOLOGY

## What is JupyterLab

**JupyterLab**

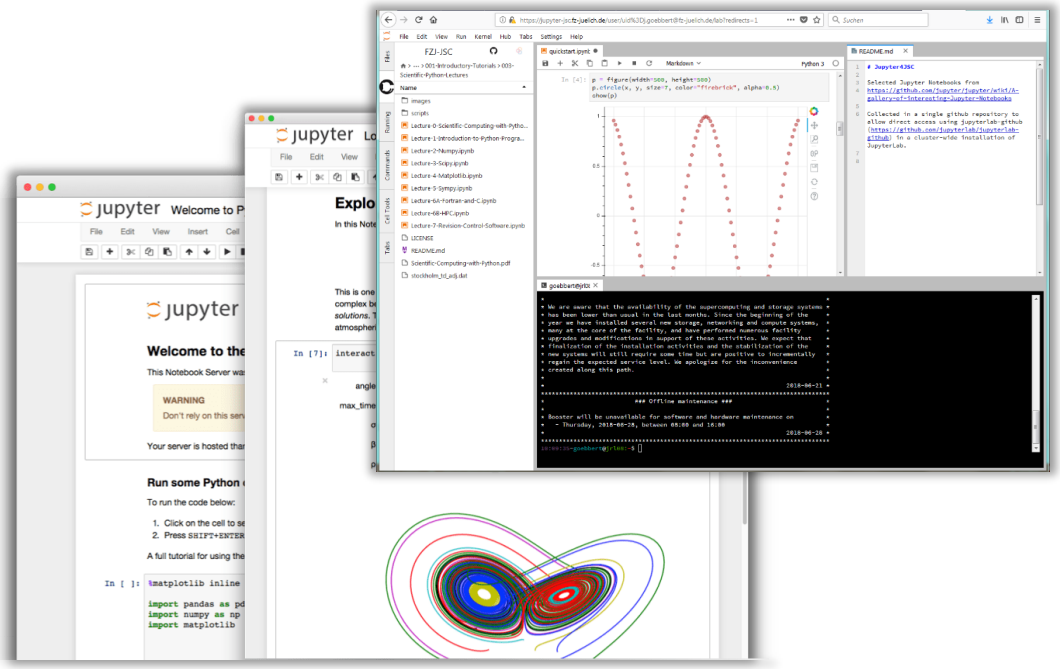
- **Interactive** working environment in the web browser
- For the creation of **reproducible** computer-aided narratives
- Very **popular** with researchers from all fields
- Jupyter = Julia + Python + R

Multi-purpose working environment

- Language agnostic
- Supports execution environments (“*kernels*”)
  - For dozens of languages: Python, R, Julia, C++, ...
- Extensible software design („*extensions*“)
  - many server/client plug-ins available
  - Eg. in-browser-terminal and file-browsing

Document-Centered Computing (“*notebooks*”)

- Combines code execution,
  rich text, math, plots and rich media.
- All-in-one document called Jupyter Notebook



https://jupyterlab.readthedocs.io

JÜLICH
Forschungszentrum

# TERMINOLOGY

## What is a Jupyter Notebook?

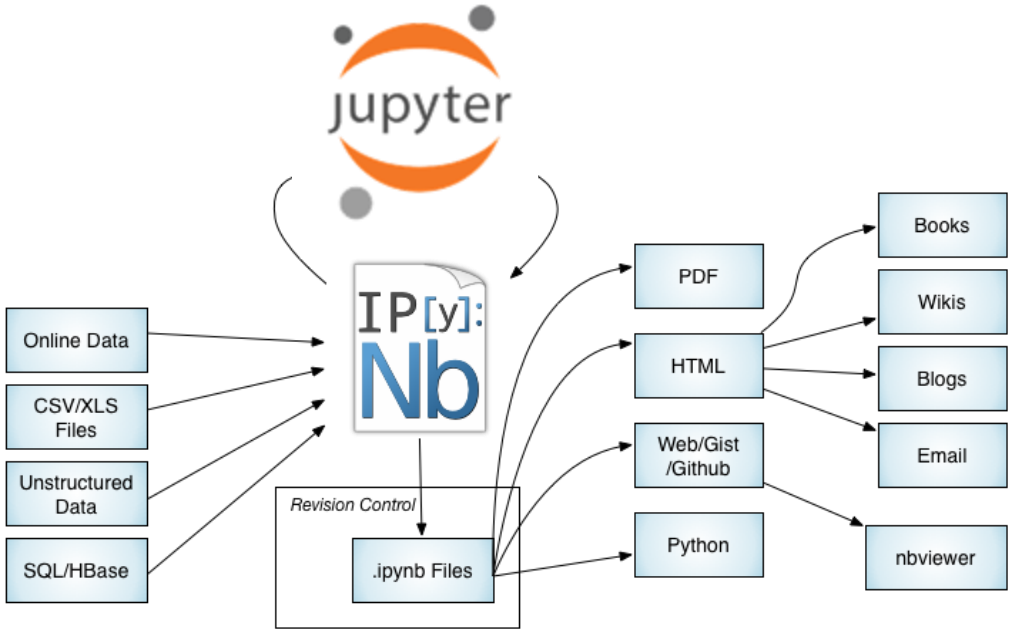
**Jupyter Notebook**

A notebook document (file extension .ipynb)
is a document that can be rendered in a web browser

- It is a file, which stores your work in JSON format
- Based on a set of open standards for interactive computing

- Allows development of custom applications with embedded interactive computing.
- Can be extended by third parties

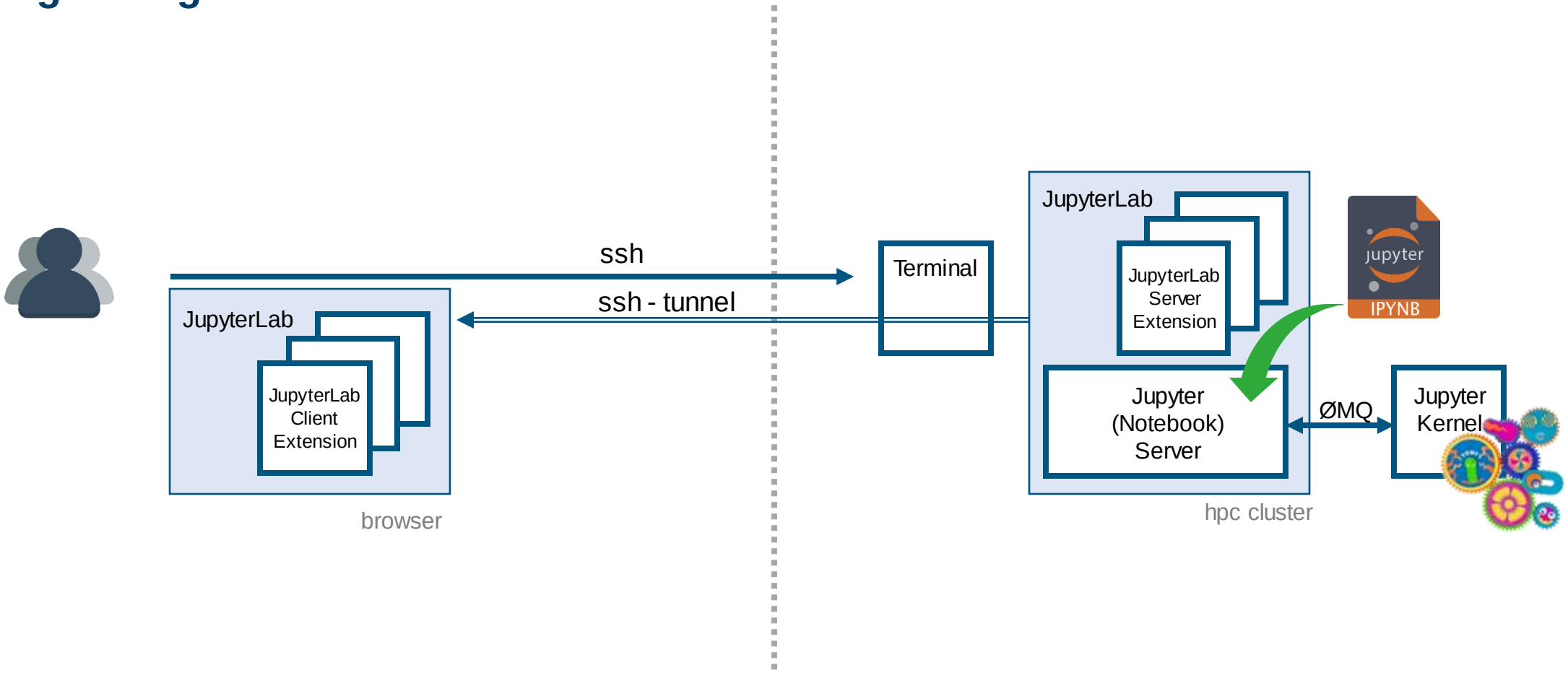
- Directly convertible to PDF, HTML, LateX ...
- Supported by many applications
  such as GitHub, GitLab, etc..




https://jupyter-notebook.readthedocs.io/
https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks

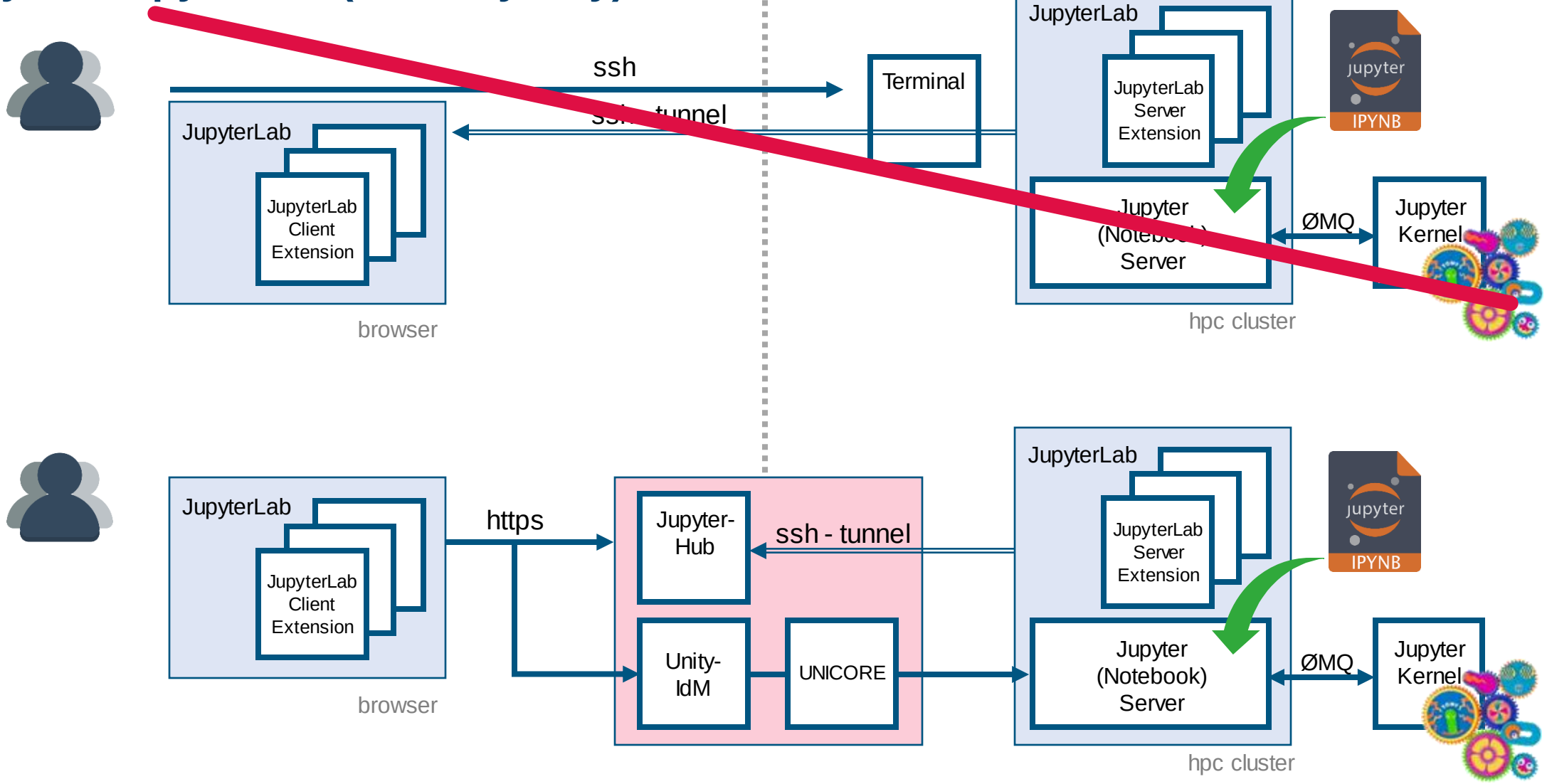
# TERMINOLOGY

## Bringing all together

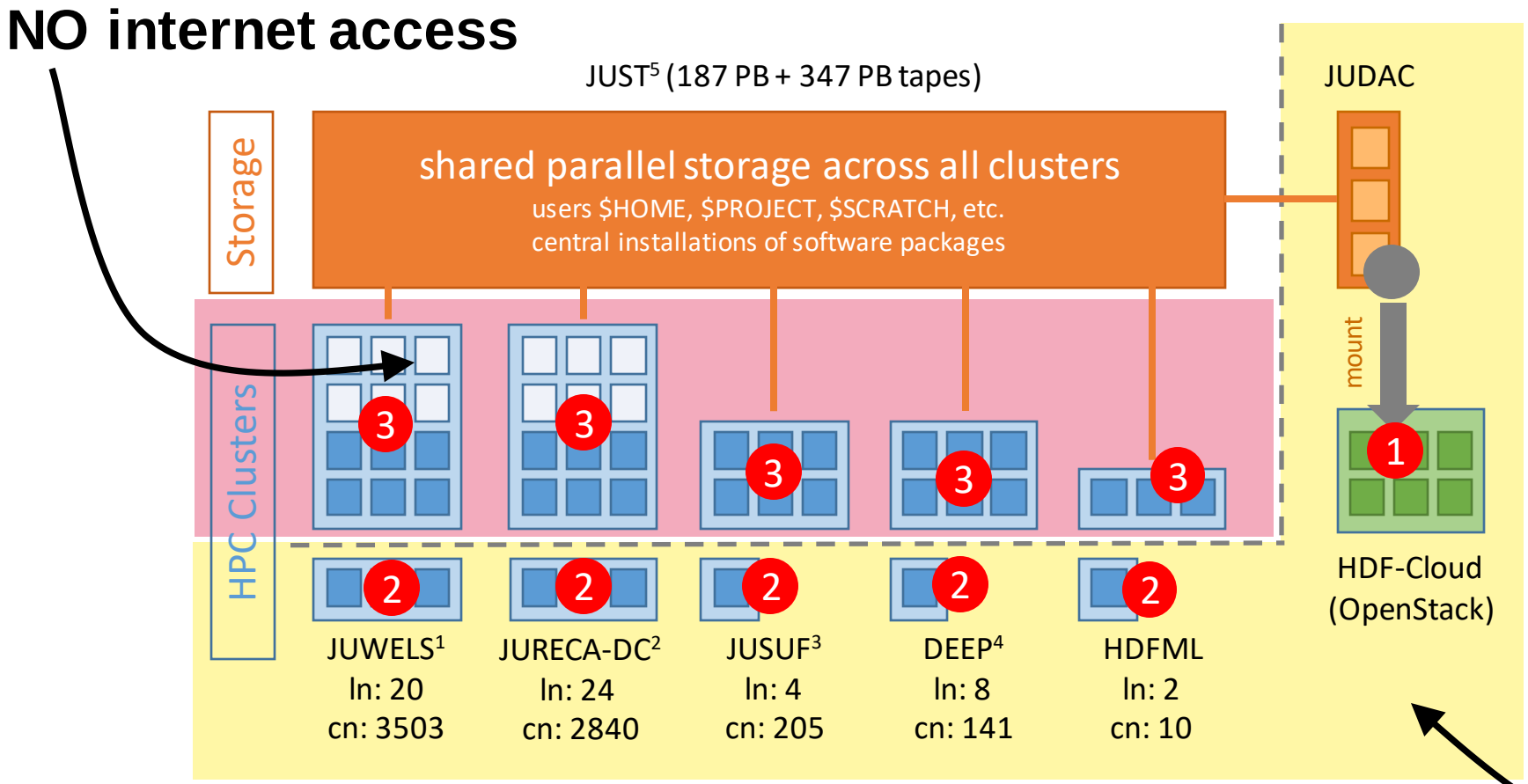


# JUPYTER-JSC WEBSERVICE

## Start your JupyterLab (the easy way)



# JUPYTERLAB EVERYWHERE




**JupyterLab everywhere**

1. JupyterLab on cloud
2. JupyterLab on login nodes
3. JupyterLab on compute nodes

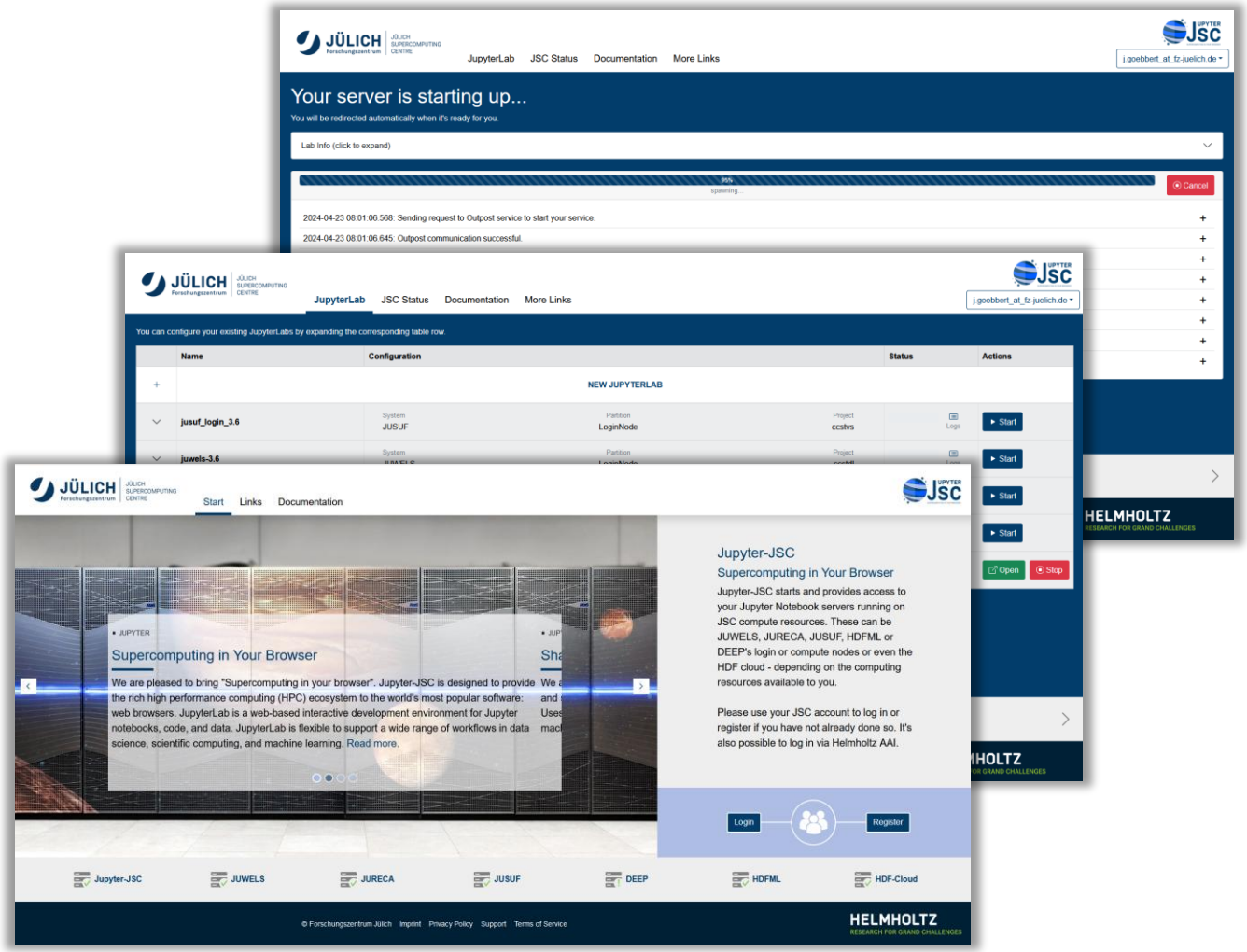
no. login nodes = ln
no. compute nodes = cn

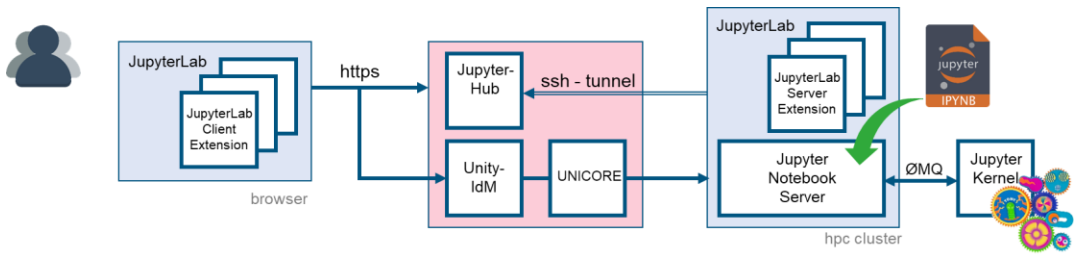

[1] https://apps.fz-juelich.de/jsc/hps/juwels/configuration.html
[2] https://apps.fz-juelich.de/jsc/hps/jureca/configuration.html
[3] https://apps.fz-juelich.de/jsc/hps/jusuf/configuration.html
[4] https://www.fz-juelich.de/en/ias/jsc/systems/prototype-systems/deep_system
[5] https://apps.fz-juelich.de/jsc/hps/just/configuration.html

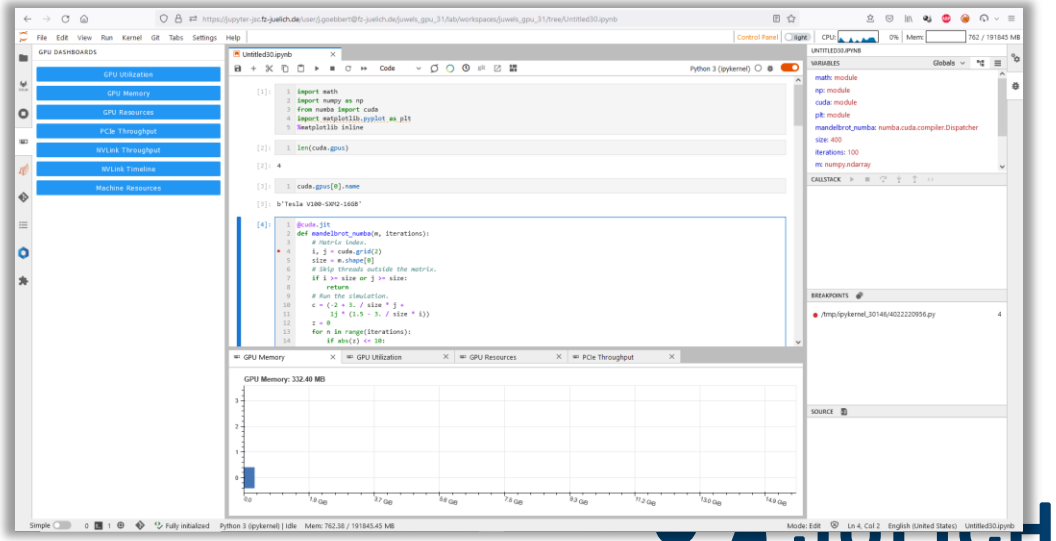

JÜLICH Forschungszentrum

# JUPYTER-JSC WEBSERVICE

## Start your JupyterLab




JÜLICH Forschungszentrum

# TERMINOLOGY

## What is a JupyterLab Extension?

**JupyterLab Extension**

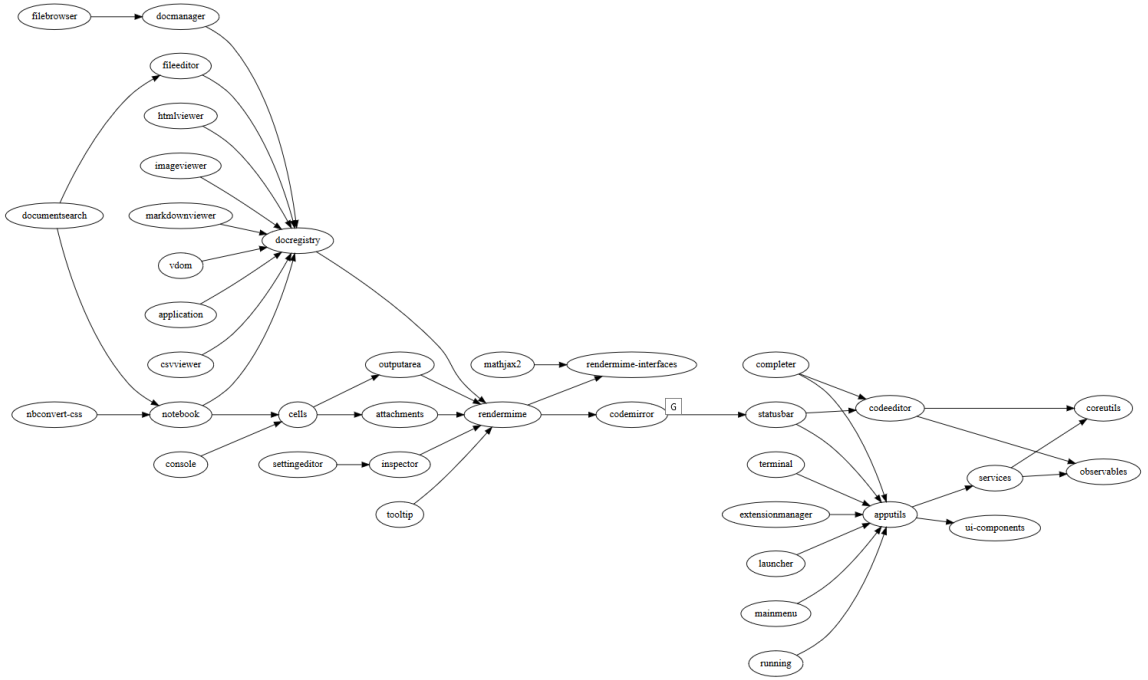
JupyterLab extensions can customize or enhance any part of JupyterLab.

JupyterLab Extensions

- provide new file viewers, editors, themes
- provide renderers for rich outputs in notebooks
- add items to the menu or command palette
- add keyboard shortcuts
- add settings in the settings system.

- Extensions can even provide an API for other extensions to use and can depend on other extensions.

The whole JupyterLab itself is simply a **collection of extensions** that are no more powerful or privileged than any custom extension.

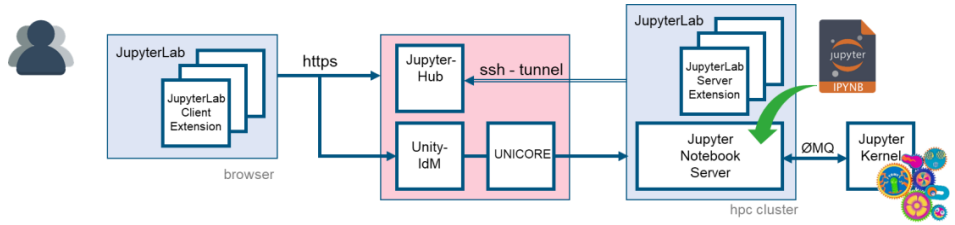



https://jupyterlab.readthedocs.io/en/stable/user/extensions.html
https://github.com/topics/jupyterlab-extension

JÜLICH Forschungszentrum

# JUPYTERLAB EXTENSIONS

## Some general information

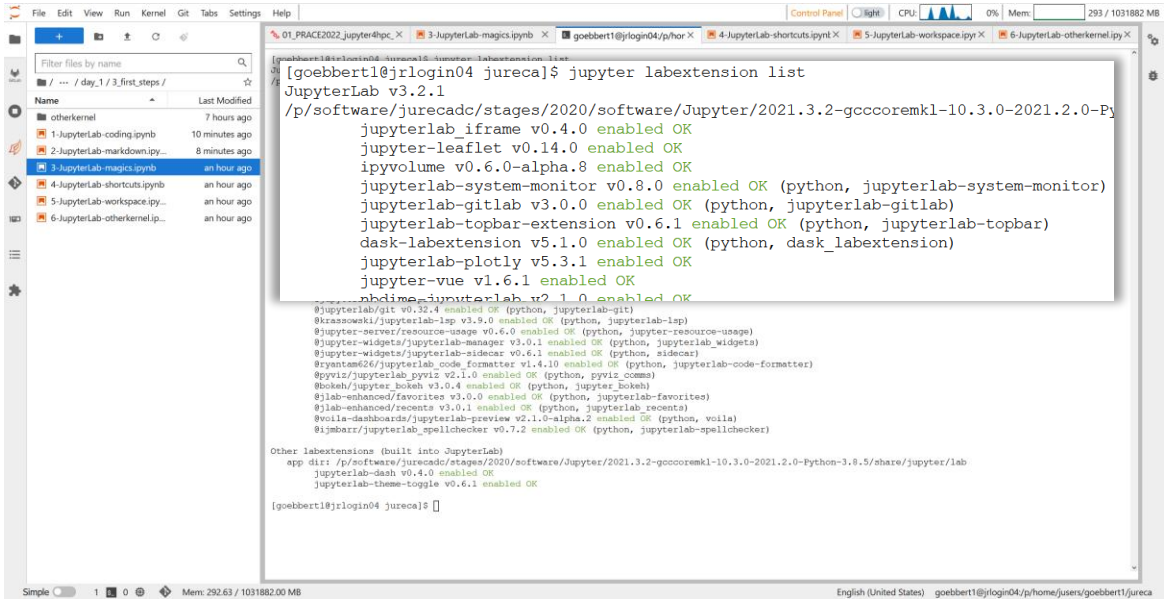



**List the installed JupyterLab extensions**

- Open the Launcher
- Start a Terminal
- Run command `jupyter labextension list`

**Extensions are installed in JupyterLab´s Application Directory, which**

- stores any information that JupyterLab persists
  - including settings and built assets of extensions
- default location is `<sys-prefix>/share/jupyter/lab`
- can be relocated by setting **$JUPYTERLAB_DIR**
  - contains the JupyterLab static assets
    - (e.g. `static/index.html`)




https://jupyterlab.readthedocs.io/en/stable/user/extensions.html

JÜLICH Forschungszentrum

# JUPYTERLAB EXTENSIONS

## Installed by default at Jupyter-JSC

| Core packages | Version | Link | Description |
|---|---|---|---|
| jupyterlab | 4.2.1 | https://jupyterlab.readthedocs.io | - |
| notebook | 7.2.0 | https://jupyter-notebook.readthedocs.io | Jupyter Notebook as a Jupyter Server extension |
| jupyterlab_server | 2.27.1 | https://jupyterlab-server.readthedocs.io | Server components for JupyterLab applications |
| jupyterhub | 4.1.5 | https://jupyterhub.readthedocs.io | Multi-user server for Jupyter notebooks |

| Optional extension | Version | Link | Description |
|---|---|---|---|
| jupyterlab-nvdashboard | 0.10.0 | https://github.com/rapidsai/jupyterlab-nvdashboard , eb-file | A JupyterLab extension for displaying dashboards of GPU usage. |
| jupyter-slurm-provisioner | 0.6.0 | https://github.com/FZJ-JSC/jupyter-slurm-provisioner , eb-file | Allows to start Jupyter kernels as a SLURM job remote from the Jupyter server |
| nglview | 3.1.2 | http://nglviewer.org/nglview/latest/ , eb-file | Jupyter widget to interactively view molecular structures and trajectories |
| jupyter-ai | 2.15.0 | https://jupyter-ai.readthedocs.io/ , eb-file | A generative AI extension for JupyterLab |

| Core Kernels | Version | Link | Description |
|---|---|---|---|
| Bash | 0.9.3 | https://github.com/takluyver/bash_kernel , eb-file | A bash kernel for IPython |
| Cling (C++) | 20231018 | https://github.com/root-project/cling/ , eb-file | Jupyter kernel for the C++ programming language |
| Julia | 1.9.5 | https://github.com/JuliaPy/pyjulia , eb-file | python interface to julia |
| LFortran | 0.30.0 | https://lfortran.org/ , eb-file | Modern interactive LLVM-based Fortran compiler |
| Octave | 8.4.0 | https://www.octave.org/ , eb-file | Scientific Programming Language - Powerful mathematics-oriented syntax with built-in 2D/3D plotting and visualization tools |
| R | 4.3.2 | https://irkernel.github.io , eb-file | R kernel for Jupyter |
| Ruby | 3.2.2 | https://github.com/SciRuby/iruby , eb-file | Ruby kernel for Jupyter |

| Community Kernels | Version | Link | Description |
|---|---|---|---|
| DeepLearning | 2024.3 | eb-file | Python kernel incl. a collection of extra modules/packages for Deep Learning |
| PyEarthSystem | 2024.3 | eb-file | Python kernel incl. a collection of extra modules/packages for the Earth System community |
| QuantumComputing | 2024.5 | eb-file | Python kernel incl. a collection of extra modules/packages for the quantum computing community |
| Visualization | 2024.3 | eb-file | Python kernel incl. a collection of extra modules/packages for visualization |

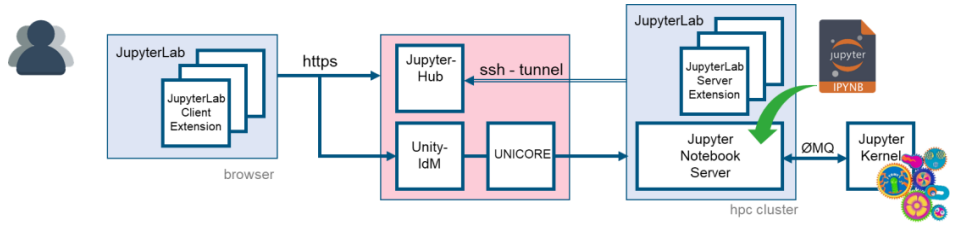
| JupyterLab Applications | Version | Link | Description |
|---|---|---|---|
| Xpra | 5.0.8 | https://xpra.org eb-file | Remote desktop for X11 applications in the browser |

| Core Extensions | Version | Link | Description |
|---|---|---|---|
| jupyter-server-proxy | 4.1.2 | https://jupyter-server-proxy.readthedocs.io | Jupyter notebook server extension to proxy web services. |
| jupyterlab-lsp | 5.1.0 | https://github.com/jupyter-lsp/jupyterlab-lsp | Coding assistance for JupyterLab using Language Server Protocol |
| ipympl | 0.9.4 | https://matplotlib.org/ipympl/ | Interactive features of matplotlib in Jupyter |
| ipyleaflet | 0.19.1 | https://ipyleaflet.readthedocs.io | Interactive maps in the Jupyter notebook |
| bqplot | 0.13.0rc0 | https://bqplot.github.io/bqplot/ | Plotting library for IPython/Jupyter notebooks |
| jupyterlab_gitlab | 4.0.0 | https://github.com/jupyterlab-contrib/jupyterlab-gitlab | A JupyterLab extension for browsing GitLab repositories |
| jupyterlab_git | 0.50.1 | https://github.com/jupyterlab/jupyterlab-git | A Git extension for JupyterLab |
| nbdime | 4.0.1 | https://nbdime.readthedocs.io/ | Tools for diffing and merging of Jupyter notebooks. |
| jupyterlab_latex | 4.0.0 | https://github.com/jupyterlab/jupyterlab-latex | JupyterLab extension for live editing of LaTeX documents |
| plotly | 5.22.0 | https://plotly.com/python/ | Python graphing library for interactive, publication-quality graphs. |
| jupyter_bokeh | 4.0.4 | https://github.com/bokeh/jupyter_bokeh | An extension for rendering Bokeh content in JupyterLab notebooks |
| panel | 1.3.8 | https://panel.holoviz.org/ | The powerful data exploration & web app framework for Python |
| holoviews | 1.18.3 | https://holoviews.org/ | With Holoviews, your data visualizes itself. |
| jupyterlab_h5web | 12.1.0 | https://github.com/silx-kit/jupyterlab-h5web | Open and explore HDF5 files in JupyterLab. Can handle very large (TB) sized files, and datasets of any dimensionality |
| ipyparallel | 8.8.0 | https://ipyparallel.readthedocs.io | IPython Parallel: Interactive Parallel Computing in Python |
| dask_labextension | 7.0.0 | https://github.com/dask/dask-labextension | JupyterLab extension for Dask |
| voila | 0.5.7 | https://voila.readthedocs.io | Voilà turns Jupyter notebooks into standalone web applications |
| nbdev | 2.3.25 | https://nbdev.fast.ai/ | Create delightful software with Jupyter Notebooks |
| sidecar | 0.7.0 | https://github.com/jupyter-widgets/jupyterlab-sidecar | A sidecar output widget for JupyterLab |
| dash | 2.17.0 | https://plotly.com/dash | Data Apps & Dashboards for Python. No JavaScript Required. |
| jupyterlab-spellchecker | 0.8.4 | https://github.com/jupyterlab-contrib/spellchecker | Spellchecker for JupyterLab notebook markdown cells and file editor. |
| jupyterlab-favorites | 3.2.1 | https://github.com/NERSC/jupyterlab-favorites | Add the ability to save favorite folders to JupyterLab for quicker browsing |
| jupyterlab-resource-usage | 1.0.2 | https://github.com/jupyter-server/jupyter-resource-usage | JupyterLab extension to for monitoring your own resource Usage |
| jupyterlab-tour | 4.0.1 | https://github.com/jupyterlab-contrib/jupyterlab-tour | A JupyterLab UI tour built on jupyterlab-tutorial and react-joyride. |
| papermill | 2.5.0 | https://papermill.readthedocs.io | Parameterize, execute, and analyze notebooks |
| pyunicore | 1.0.0 | https://github.com/HumanBrainProject/pyunicore | UNICORE REST bindings for python |

https://gitlab.jsc.fz-juelich.de/jupyter4jsc/j4j_notebooks/-/blob/documentation/05-News&Updates/Announcement-2024-06_JupyterLab4.2-Upgrade.ipynb

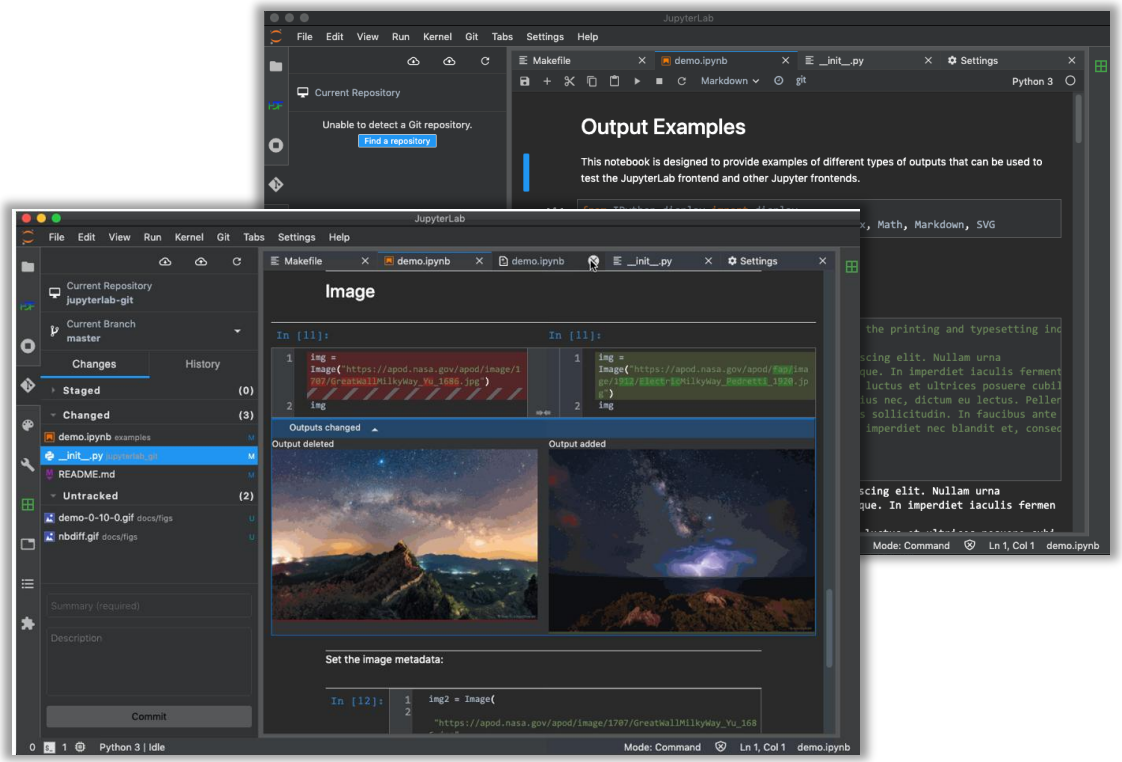
# JUPYTERLAB EXTENSIONS

## Installed by default at Jupyter-JSC




**JupyterLab-Git**

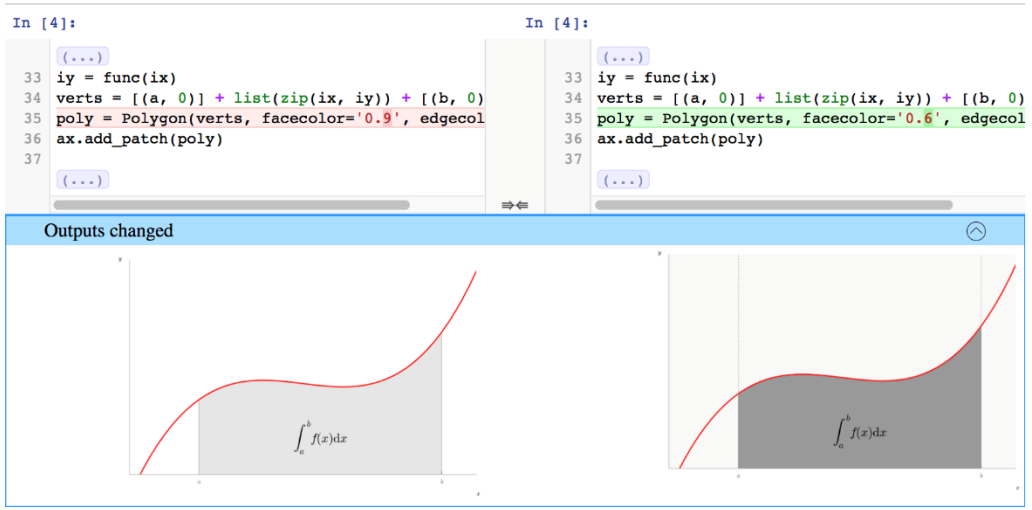
JupyterLab extension for version control using Git




https://github.com/jupyterlab/jupyterlab-git

**NBDime**

Tools for diffing and merging of Jupyter notebooks.

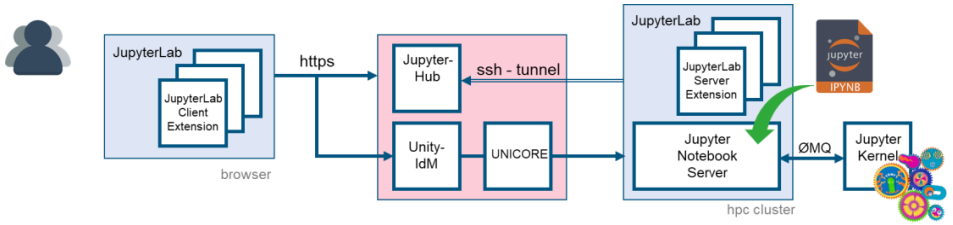



https://github.com/jupyter/nbdime

JÜLICH Forschungszentrum

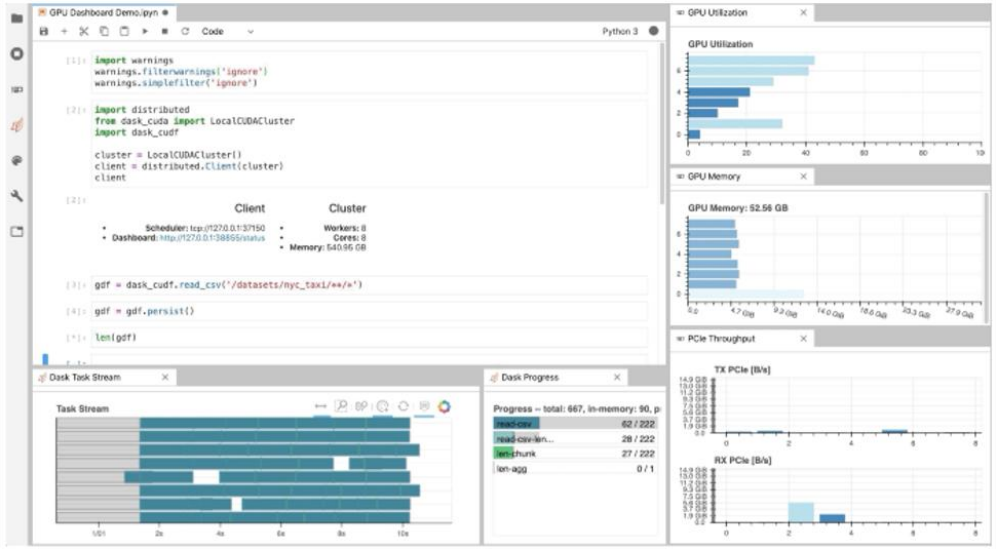
# JUPYTERLAB EXTENSIONS

## Installed by default at Jupyter-JSC




**NVDashboard**

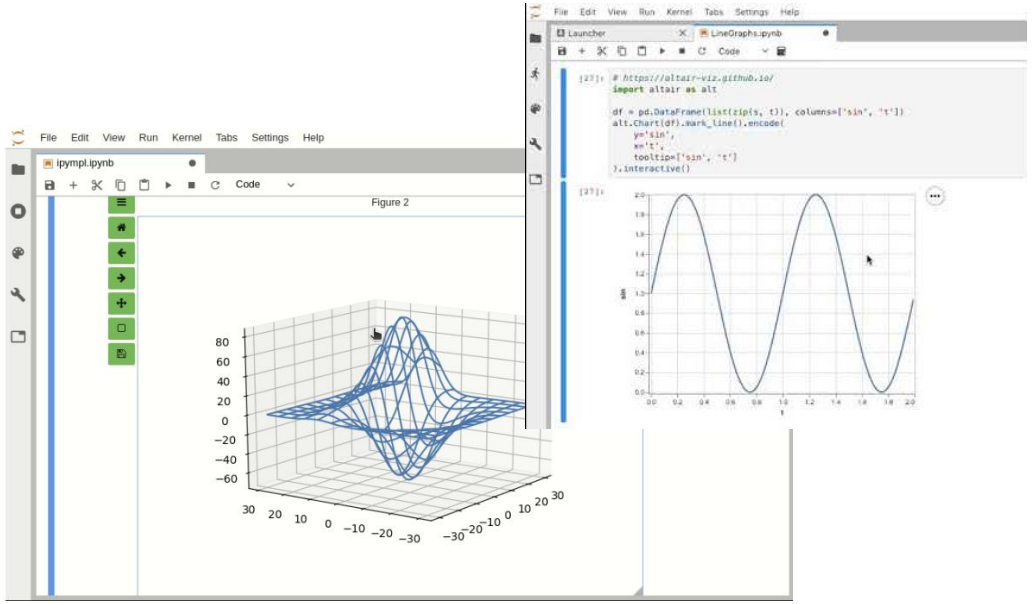
NVDashboard is an open-source package for the real-time visualization of NVIDIA GPU metrics in interactive Jupyter Lab environments.




https://github.com/rapidsai/jupyterlab-nvdashboard
https://developer.nvidia.com/blog/gpu-dashboards-in-jupyter-lab/

**IPyMPL - matplotlib**

Leveraging the Jupyter interactive widgets framework, ipympl enables the interactive features of matplotlib in the Jupyter notebook and in JupyterLab.




https://github.com/matplotlib/ipympl

JÜLICH Forschungszentrum

# TERMINOLOGY

## What is a Jupyter Kernel?

**Jupyter Kernel**
A “kernel” refers to the separate process which executes code cells within a Jupyter notebook.

Jupyter Kernel

- **run code** in different programming languages and environments.
- can be **connected to** a notebook (one at a time).
- **communicates** via ZeroMQ with the JupyterLab.

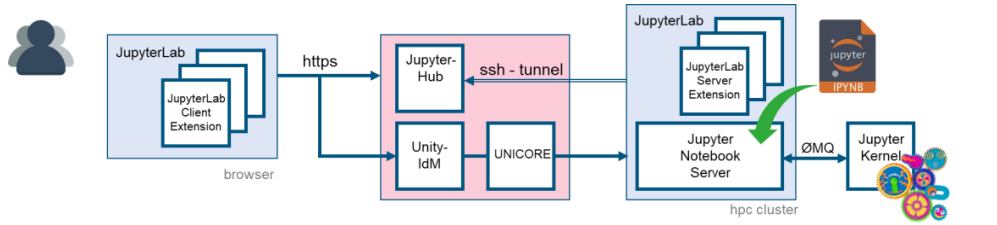
- Multiple **preinstalled** Jupyter Kernels can be found on our clusters
  - Python, R, Julia, Bash, C++, Ruby, JavaScript
  - Specialized kernels for visualization, quantum-computing

- You can easily **create your own kernel** which for example runs your specialized virtual Python environment.


https://jupyter-notebook.readthedocs.io/
https://github.com/jupyter/jupyter/wiki/Jupyter-kernels
https://zeromq.org

# JUPYTER KERNEL

## How to create your own Juypter Kernel



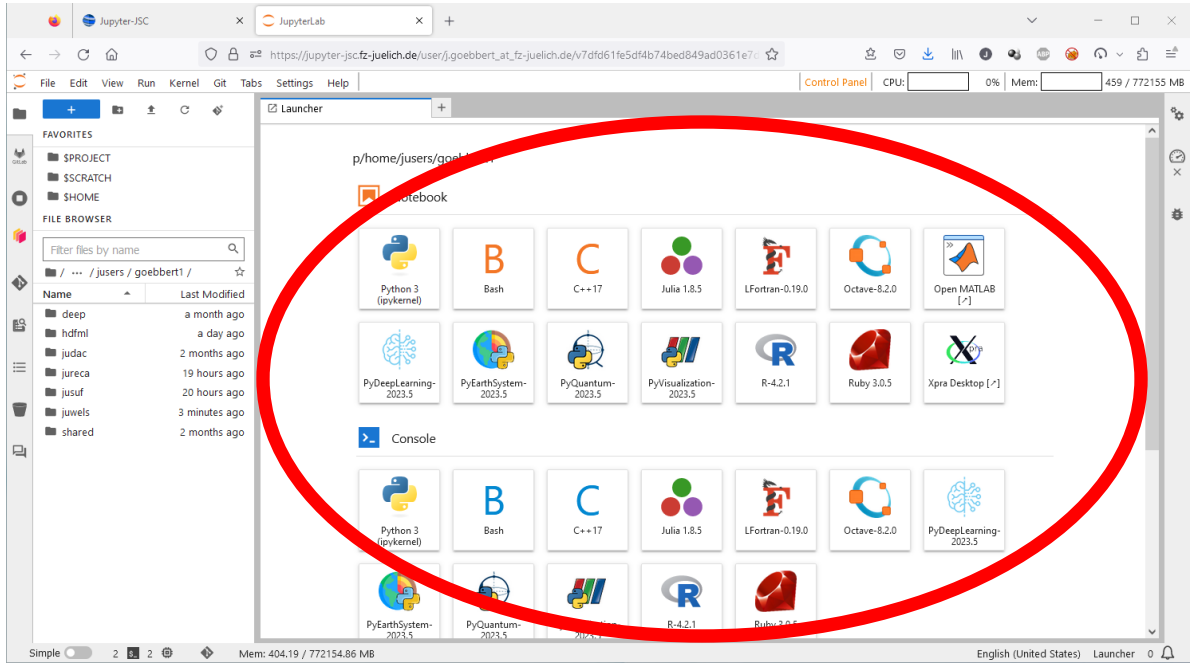

**Jupyter Kernel**

A “kernel” refers to the separate process which executes code cells within a Jupyter notebook.

Jupyter Kernel

- run code in different programming languages **and environments**.
- can be connected to a notebook (one at a time).
- communicates via ZeroMQ with the JupyterLab.

- Multiple **preinstalled** Jupyter Kernels can be found on our clusters
  - Python, R, Julia, Bash, C++, Ruby, JavaScript
  - Specialized kernels for visualization, quantum computing



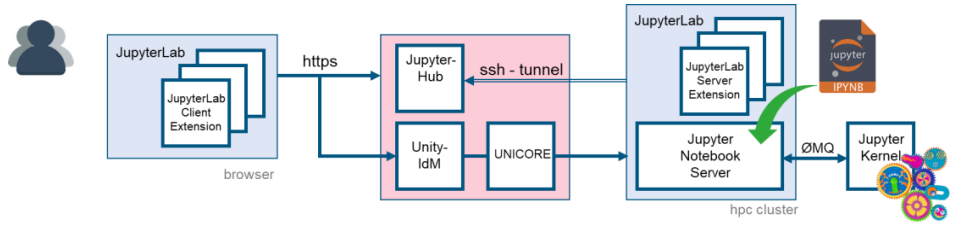

You can easily **create your own kernel** which for example runs your specialized virtual Python environment.

https://github.com/jupyter/jupyter/wiki/Jupyter-kernels

JÜLICH Forschungszentrum

# JUPYTER KERNEL

## How to create your own Juypter Kernel

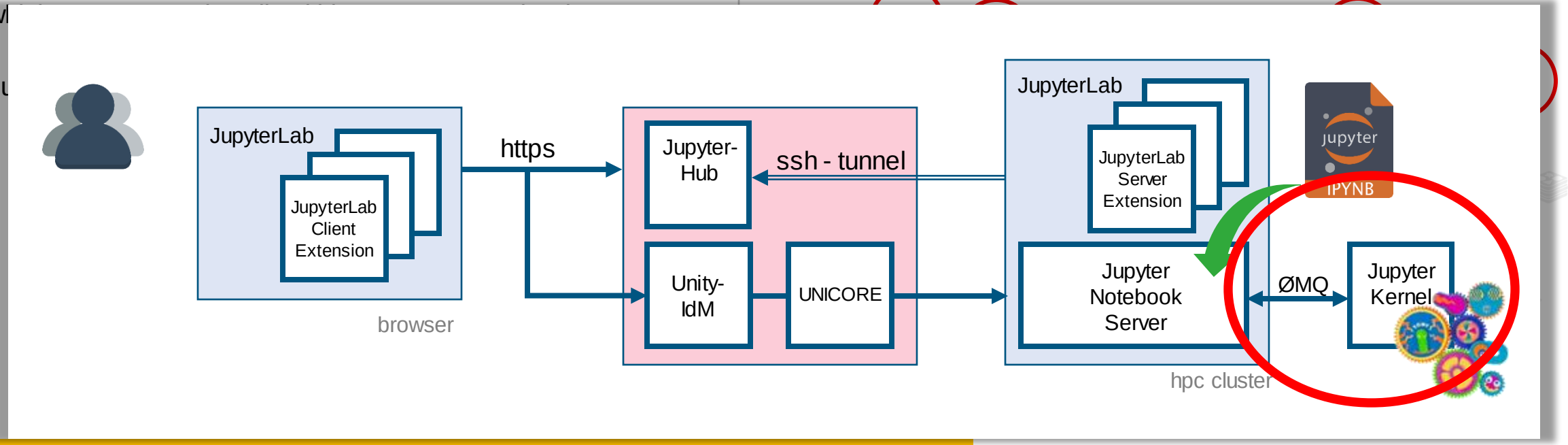



**Jupyter Kernel**

A “kernel” refers to the separate process w

Ju

- 
- 
- 
- 

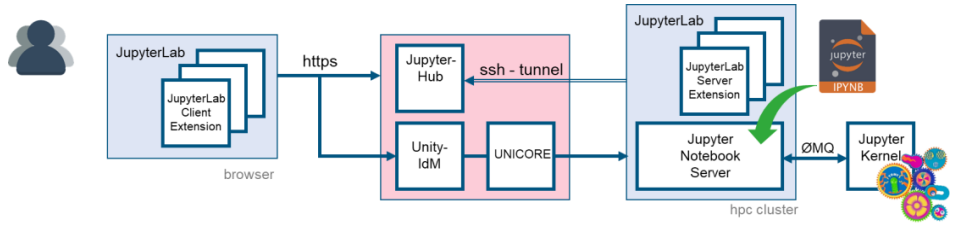



You can easily **create your own kernel** which for example runs your specialized **virtual Python environment** including **modules of the system.**

https://github.com/jupyter/jupyter/wiki/Jupyter-kernels

# JUPYTER KERNEL

## How to create your own Juypter Kernel




**Jupyter Kernel**

A “kernel” refers to the separate process which executes code cells within a Jupyter notebook.

Jupyter Kernel

- run code in different programming languages **and environments**.
- can be connected to a notebook (one at a time).
- communicates via ZeroMQ with the JupyterLab.

- Multiple **preinstalled** Jupyter Kernels can be found on our clusters
  - Python, R, Julia, Bash, C++, Ruby, JavaScript
  - Specialized kernels for visualization, quantumcomputing

**Building your own Jupyter kernel is a three step process**

1. Create/Pimp new **virtual Python environment**
   `venv`
2. Create/Edit **launch script** for the Jupyter kernel
   `kernel.sh`
3. Create/Edit Jupyter **kernel configuration**
   `kernel.json`

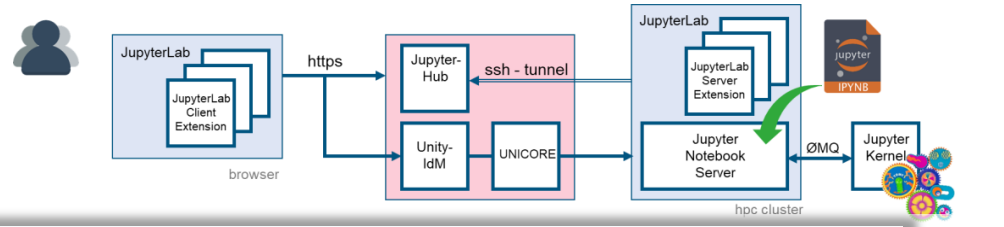
You can easily **create your own kernel** which for example runs your specialized **virtual Python environment** including **modules of the system.**

https://github.com/jupyter/jupyter/wiki/Jupyter-kernels

JÜLICH Forschungszentrum

# JUPYTER KERNEL

## How to create your own Juypter Kernel

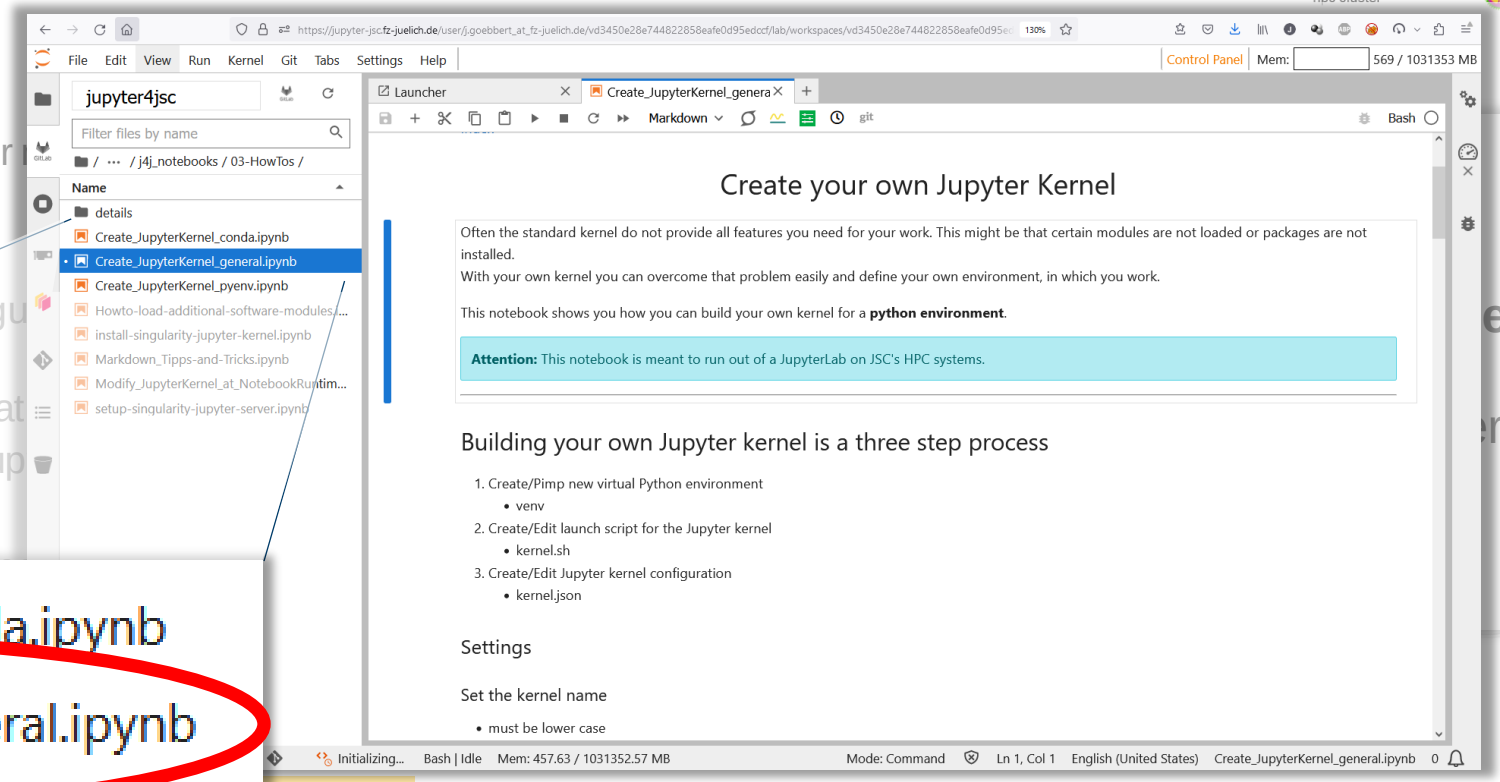


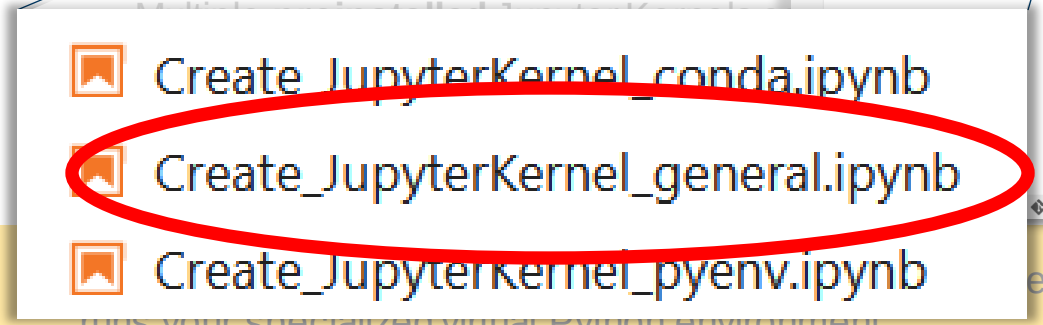

### Jupyter Kernel

A “kernel” refers to the separate process which executes code cells within a Jupyter

Jupyter Kernel

- run code in different programming langu **and environments**.
- can be connected to a notebook (one at
- communicates via ZeroMQ with the Jup




runs your specialized virtual Python environment.

https://github.com/jupyter/jupyter/wiki/Jupyter-kernels

https://gitlab.version.fz-juelich.de/jupyter4jsc/j4j_notebooks/-/blob/master/001-Jupyter/Create_JupyterKernel_general.ipynb

JÜLICH Forschungszentrum

# QUESTIONS?


Training course: https://gitlab.jsc.fz-juelich.de/jupyter4jsc/training-2025.04-jupyter4hpc