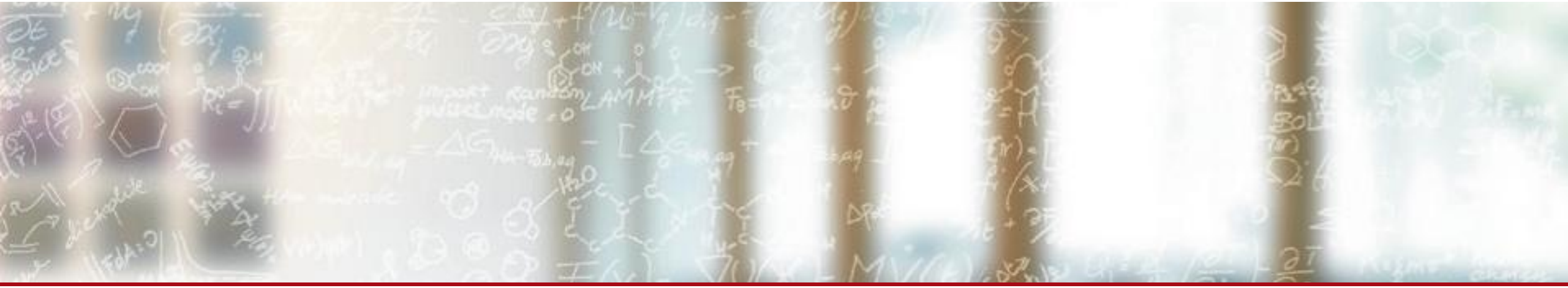
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Interactive computing on Alps

- **Jupyter** notebooks
- Setting up **VSCode**
- **FirecREST** web portal and recent updates

CSCS User Lab Day, 29 August 2025

This session’s presentations

1. Introduction to using **Jupyter Notebooks** via **JupyterHub**
2. Best practices for setting up **VSCode** on Alps vClusters
3. **FirecREST web portal** and recent improvements

Tim Robinson

HPC Platform Service Manager

Prashanth Kanduri

Research Software Engineer

Juan Dorsch

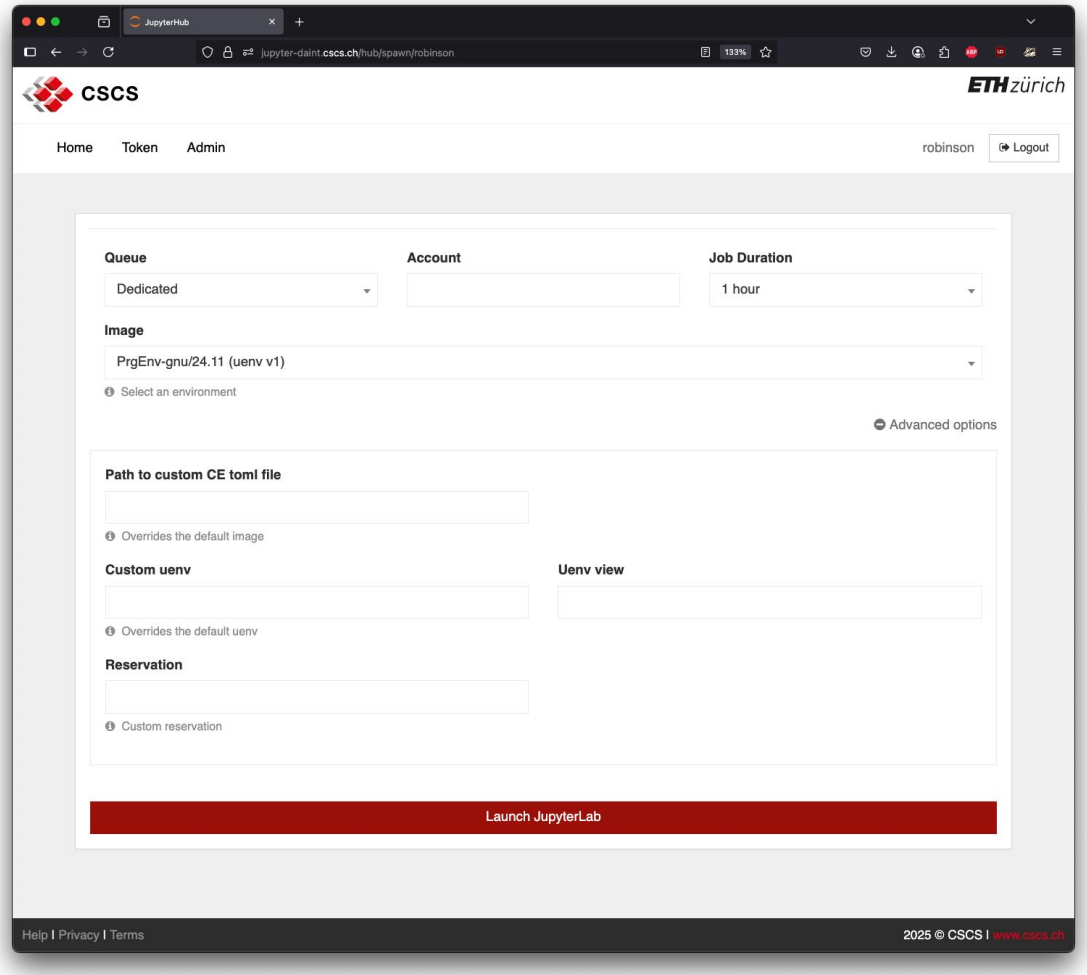
Senior Software Engineer

JupyterHub on Alps

- Runs JupyterLab on compute nodes on Alps
- Separate deployments per vCluster
  - https://jupyter-daint.cscs.ch
  - https://jupyter-eiger.cscs.ch
  - https://jupyter-santis.cscs.ch
  - https://jupyter-clariden.cscs.ch
- Documentation is available in CSCS Docs
  - Log file is saved in `slurm-<jobid>.out` in `$HOME` or `$SCRATCH`
- Software environment is provided by uenv or containers
  - You can use your own virtual environment by creating a kernel
  - You can load your own uenv (⊕Advanced options)
  - You can use your own container TOML (⊕Advanced options)

Spawner options

- Queue
  - Dedicated (if available)
  - If job does not start, try Standard
- Account
- Job Duration
- Image
  - Preinstalled uenv, or
  - Preinstalled container image
- Advanced options
  - Path to custom container TOML file, or
  - Custom uenv (with optional view)
  - Reservation



User stories

Uenv and creating a kernel

***User story**: I use the **PrgEnv-gnu uenv**. I use a Python virtual environment with the Python modules I need. How can I use this as a kernel in JupyterHub?*

- Select “Image” -> “PrgEnv-gnu/24.11 (uenv v1)”
- Launch JupyterLab
- It will pull the uenv if needed (first launch can take a few min)
- Install packages in a virtual environment on top of the Python in the base uenv

```
python -m venv --system-site-packages venv-<base-image-version>
. venv-<base-image-version>/bin/activate
pip install ipykernel
```

- Create a kernel for the virtual environment

```
python -m ipykernel install ${VIRTUAL_ENV:+--env PATH $PATH --env VIRTUAL_ENV $VIRTUAL_ENV}
--user --name="<kernel-name>"
```

Demo

- https://jupyter-daint.cscs.ch

Pre-installed container

***User story**: I want to use a Jupyter notebook with **PyTorch** to test algorithms and workflows quickly on Grace Hopper architecture.*

- Select “Image” -> “nvcr.io/nvidia/pytorch:25.04-py3 (PyTorch 2.7.0)”
- Launch JupyterLab
- Under the hood, this creates a TOML file and launches the container with the Container Engine

```
$ cat $SCRATCH/jhub.toml
image = "/capstor/store/cscs/cscs/jupyter/pytorch/pt-25.04-py3-jlab.sqsh"
mounts = ["/iopsstor/scratch/cscs/robinson:/iopsstor/scratch/cscs/robinson",
"/capstor/scratch/cscs/robinson:/capstor/scratch/cscs/robinson"]
workdir = "/capstor/scratch/cscs/robinson"
```

- It will pull the container image if needed (first launch can take a few min)

Custom container environment (1)

***User story**: The default base images don't meet my needs. I want to use JupyterHub with my **custom container image** that I use for my other workflows.*

- Add a few additional packages to your Dockerfile

```
FROM nvcr.io/nvidia/pytorch:25.05-py3
RUN pip install --no-cache \
    jupyterlab \
    jupyterhub==4.1.6 \
    pyfirecrest==1.2.0 \
    SQLAlchemy==1.4.52 \
    oauthenticator==16.3.1 \
    notebook==7.3.3 \
    jupyterlab_nvdashboard==0.13.0 \
    git+https://github.com/eth-cscs/firecrestspawner.git
```

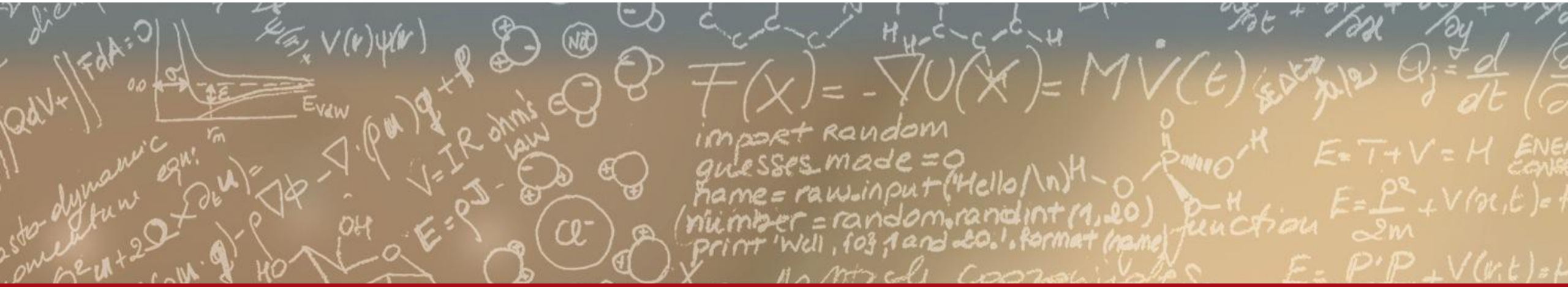
- Rebuild your container image

Custom container environment (2)

- Create a corresponding TOML file

```
image = "/capstor/scratch/cscs/${USER}/ce-images/ngc-pytorch+25.05.sqsh"

mounts = [
    "/capstor",
    "/iopsstor",
    "/users/${USER}/.local/share/jupyter",
    "/etc/slurm",
    "/usr/lib64/libslurm-uenv-mount.so",
    "/etc/container_engine_pyxis.conf"
]

workdir = "/capstor/scratch/cscs/${USER}"

writable = true

[annotations]
com.hooks.aws_ofi_nccl.enabled = "true"
com.hooks.aws_ofi_nccl.variant = "cuda12"

[env]
CUDA_CACHE_DISABLE = "1"
TORCH_NCCL_ASYNC_ERROR_HANDLING = "1"
MPICH_GPU_SUPPORT_ENABLED = "0"
```

- "Advanced Options" -> "Path to custom CE toml file"
  - Insert the full path to the TOML
  - This overrides the default image
- Launch JupyterLab



Let us know your use cases!

- Read the documentation at https://docs.cscs.ch/access/jupyterlab/
- Give your feedback / requests for enhancement at https://support.cscs.ch

Thanks!