# Deploying infrastructure-as-a-service at GSI with JupyterHub

Jeremy Wilkinson (GSI Helmholtzzentrum für Schwerionenforschung GmbH, Darmstadt)

Jupyter notebooks allow interactive Python-based graphical analysis to be run remotely from the user's laptop over a web socket. However, running such sessions using cluster resources is complex to set up and not always possible due to firewall policies. In addition, specialised resources such as GPUs for machine learning may require dedicated support such as up-to-date drivers and framework libraries. The JupyterHub [1] service offers a central solution for spawning, running, and managing Jupyter instances via the institute's job scheduler, in a way that is transparent to the user and with authentication managed securely by central services.



#### JupyterHub VirgoSpawner:

- Custom spawner plugin allows deployment of jobs to GSI's Virgo computing cluster [2]



- Spawns and manages Jupyter sessions via web interface, simplifying user workflow

- Job allocation based on user's standard quota, with job submission transparent to user

- User can customise job parameters according to resource needs (configurable by service admins)

- Stable environment provided by centrally managed Apptainer [3] containers, including support for GPU drivers where needed

- Error handling in case of failure in job submission

- Planned for deployment within this year

#### Key security considerations for planning:

- Web service does not run with elevated rights or

## **External packages:**

- Authentication using KeyCloak [4] (with 2FA) as SSO provider and OIDC token broker

- KeyCloakAuthenticator plugin [5] from CERN's SWAN service handles auth state within JupyterHub
- Mapping between GSI Web login and Linux cluster account handled in LDAP

- Validation of OIDC token using Motley Cue [6], authorised by ssh-oidc PAM plugin

## directly access Slurm credentials

- No access to cluster resources or files via Web without 2-factor authentication
- Job submission performed as the user with secured credentials handled by trusted service no unauthenticated impersonation of user
- Spawner plugin validates + sanitises user option inputs to prevent invalid or malicious spawn parameters
- Extension to AsyncSSH [7] Python package allows passing of OIDC token as SSH credential



#### **References:**

- [1] JupyterHub: https://jupyterhub.readthedocs.io
- [2] GSI Virgo Cluster: https://hpc.gsi.de/virgo/
- [3] Apptainer: https://apptainer.org
- [4] KeyCloak: https://www.keycloak.org/
- [5] KeyCloakAuthenticator: https://github.com/swan-cern/jupyterhub-extensions
- [6] Motley Cue: https://dianagudu.github.io/motley\_cue/
- [7] AsyncSSH: https://asyncssh.readthedocs.io/en/latest/