



Contribution ID: 129

Type: **Presentation**

## **Jupyterhub on Kubernetes as a platform for developing secure shared environment for data analysis at MAX IV**

*Tuesday 12 March 2024 10:00 (20 minutes)*

MAX IV Laboratory has operated as a user facility since 2016 and continuously evolving the IT infrastructure to facilitate data collection and enable end-user data analysis possibilities. Jupyterhub running on the bare-metal Kubernetes cluster is one of the primary environments at MAX IV premises aimed to address the challenge of providing secure and shared service, while optimizing access to compute and GPU resources for scientific data analysis.

An initial key objective was the development of a fully unprivileged container environment that operates seamlessly with existing user credentials. This approach aims to enhance security without compromising accessibility of the scientific data. The goal of achieving this without direct modification of the notebook container image is challenging but solved via the introduction of helper services that sync data from the user database (LDAP) to the Kubernetes objects and mounting the overlay inside the container.

In order to achieve enhanced resource visibility within the containers, the project integrates LXCFS [1] into the platform. This integration provides a comprehensive view of available resources, a pivotal feature for efficient data analysis and management when it comes to running parallel tasks e.g. via OpenMP.

The project also emphasizes robust GPU sharing support, covering both V100 and A100 GPUs, including dedicated and shared MIG partitions on the A100 GPUs. Management of GPU memory within Kubernetes is facilitated through MortalGPU [2] - an in-house developed fork of MetaGPU. This solution allows for overcoming commitment and limitation of GPU memory, akin to RAM, providing fine-grained control and container-scoped visibility of GPU resources. Prometheus exporter of MortalGPU shows GPU resource utilization, offering usage statistics in Grafana and insights into resource availability during instance spawning.

To further extend capabilities, JupyterHub hooks are utilized to introduce “Compute Instance profiles” support to offer end-users to choose between shared GPU partition or dedicated resource options. Moreover, hooks are used for enabling precise control over profiles and images access restrictions through RBACs defined in the IdP.

Additionally, extra containers within the Pods with Jupyter Notebooks are conditionally deployed to enforce features like walltime restrictions for notebooks or utilize JupyterHub as an OIDC client, providing JWT tokens to CLI tools (such as in the Nordugrid ARC [3] client case).

Running the setup on Kubernetes platform brings benefits of re-using the infrastructure to deploy several environments. Currently we are running production deployment, test deployment (for minor updates testing before production) and the “next” deployment for major further developments. We are working towards establishing another deployment exclusively for providing EOSC service as Open Data analysis platform. Moreover the same infrastructure is used for Jupyter notebooks CI testing as well [4].

This project exemplifies successful usage of JupyterHub on Kubernetes as a base platform for developing a secure, shared environment for data analysis at MAX IV. By addressing resource management, security, and extensibility, it delivers collaborative scientific data analysis platform.

The contribution will provide implementation details of the platform and example use-cases running at MAX IV.

[1] LXCFS: <https://linuxcontainers.org/lxcfs>

[2] MortalGPU - Kubernetes device plugin implementing the sharing of Nvidia GPUs between workloads: <https://gitlab.com/MaxIV/kubernetes/mortalgpu>

[3] “Advanced Resource Connector middleware for lightweight computational Grids”. M.Ellert et al., Future Generation Computer Systems 23 (2007) 219-240.

[4] Brudvik, J., Schoen, S., Matej, Z., & Barty, A. (2021). ExPaNDS Testing and Validation Framework (1.0). Zenodo. <https://doi.org/10.5281/zenodo.5718671>

**Primary author:** SALNIKOV, Andrii

**Co-authors:** MATEJ, Zdenek; ERMAKOV, Dmitrii; BRUDVIK, Jason

**Presenter:** SALNIKOV, Andrii

**Session Classification:** Collaborative Data Science and Visualisation

**Track Classification:** Main sessions: User Voice: Innovative Applications, Data Science Environments & Open Data