1. Conference on Research Data Infrastructure

Poster Session I: September, 12th 2023, 5:30-7:00 pm, Audimax Foyer (Building 30.95)

https://doi.org/10.5281/zenodo.8333419

Published: September, 12th 2023

JupyterHub as a Service

On-Demand Course-Related JupyterHubs for Research and Teaching

Nils Mittler[1 2], Frank Förster[1 2], Alexander Goesmann[2], and Burkhard Linke[1 2]

[1]Bioinformatics Core Facility, Justus Liebig University Giessen, Giessen, Germany [2]Bioinformatics and Systems Biology, Justus Liebig University Giessen, Giessen, Germany

Abstract: Jupyter Notebook is a popular tool for writing, documenting, and sharing code. It is a server program that can be run on a local computer and that provides a web-based environment for working with code and data. A JupyterHub forms an organizational layer for Jupyter Notebooks on a server and allows authentication and authorization of users. Using a JupyterHub, it is possible to provide a predefined and uniform software stack for all users of this JupyterHub, which is especially attractive for courses. However, running your own, self-configurable JupyterHub brings challenges in the form of the required resources and knowledge to run a web service. JupyterHub as a Service (JHaaS) addresses this issue by enabling fully automated on-demand deployments of JupyterHub instances in arbitrary Kubernetes clusters. To do this, a course leader requests a JupyterHub from JHaaS and defines a configuration appropriate for that course. A governance can agree to this request, whereupon JHaaS takes care of the deployment of the JupyterHub before the course starts and the deletion after the course ends. Course participants can apply to join the JupyterHub and can be accepted by the course leader. With JHaaS, it is easy to deploy JupyterHub instances for many use cases while giving control over the definition of and the access to the JupyterHub to the course leader. JHaaS is currently in a very early stage of development and is accordingly subject to active change. A trial run in regular courses is planned for the upcoming winter semester of 2023–2024 and our source code is available on GitHub: https://github.com/orgs/JLU-BCF/repositories.

Keywords: Jupyter Notebook, Jupyter, JupyterHub, cloud-native

1 Introduction

Jupyter Notebook is an established tool in the fields of data science and scientific computing, e.g., for writing, documenting, and sharing code. It is a server program that provides a web-based and thus largely device-independent environment with a predefined software stack. This server program can be executed and used on a local computer, but it does not provide authentication and authorization mechanisms. Therefore, it is not suitable for remote access.

This is where JupyterHub comes in and provides a central web service that manages Jupyter Notebooks and allows authentication and authorization of users. The server program for the Jupyter Notebook doesn't need to be installed on the local machine, but can be deployed elsewhere and accessed via the web, managed by JupyterHub. This creates a largely device-independent environment, which is particularly attractive in the context of teaching courses, workshops, and other events (hereafter together referred to as ”courses”).

For courses that require a specific software stack, setting up this environment on the respective end user devices can take a lot of time. This problem can be addressed by using a JupyterHub that spawns predefined Jupyter Notebooks. Public, internal, or private JupyterHub instances can be used for this purpose. For the course leader, public and internal instances, on the one hand, lack the necessary control over the software stack for the spawned Jupyter Notebooks and the resources to be used. On the other hand, deploying a private JupyterHub for a specific course brings other challenges, in particular the need for resources and knowledge to run a web service.

Thus, we developed JupyterHub as a Service (*JHaaS*). It allows course leaders without the appropriate knowledge or resources to create their own JupyterHub that spawns Jupyter Notebooks predefined by the course leaders themselves. *JHaaS* differentiates itself from existing solutions in that it does not provide a single large JupyterHub instance for all courses, but actually rolls out a full-fledged JupyterHub instance for each single course. This approach allows the course leader to define the optimal software stack for his course in a Jupyter Notebook, which includes software used within the notebook, the kernel itself, any extensions, and web proxy applications.

2 Goals

JHaaS is intended to be a cloud-based solution for automated lifecycle management of JupyterHub instances on arbitrary Kubernetes clusters. It allows course leaders to submit a request for a JupyterHub for their course, which will be reviewed by a governance. In the case of an accepted request, *JHaaS* will automatically take care of deploying a corresponding JupyterHub instance on a Kubernetes cluster before the course starts. Moreover, course participants can apply themselves to participate in the corresponding JupyterHub and can then be accepted by the course leader. Thus, the course leader has full control over the participants on the JupyterHub. At the end of the course, *JHaaS* will arrange for the complete deletion of the JupyterHub instance and all associated data.

Due to the possibility for participants to execute arbitrary code in a Jupyter Notebook, security is one of the top design goals: proven designs such as multi-factor authentication are to be used. Moreover, we try to ensure compliance with applicable data protection regulations. In addition, the application should be designed to be interoperable with many identity providers, to be user-friendly and intuitive, and to minimize administrative overhead.

3 Relation to Research Data Infrastructures

In the context of NFDI, data is aggregated and must subsequently be processed. Jupyter Notebook has proven to be an easy-to-use tool for this purpose. With *JHaaS*, it is possible to seed initial data from various backends, such as Aruna Object Storage [1], into a Jupyter Notebook. In the medium term, a “data browser” will be implemented within the *JHaaS* user interface. Thus, users will be enabled to load data from existing backends or upload custom data in a simple manner.

Given the flexible design of *JHaaS*, selecting a specific Kubernetes cluster to deploy the JupyterHub instance on, will be available in a later release. In a federated setup, for example, a Kubernetes cluster could be selected with geographic proximity to the research data to be used, which could significantly improve data processing performance.

Some NFDI consortia already operate their own JupyterHub instances for research purposes. Currently, such instances are collected and listed centrally to give researchers an overview of the available JupyterHubs together with their specifications in Germany [2]. This already shows the relevance of the Jupyter products for the Research Data Infrastructure.

4 Results and Outlook

JHaaS is being developed with the set goals in mind and is currently at a very early stage of development. The core technical functionalities, such as the deployment and deletion of JupyterHub instances, authentication and authorization, management of participants, and participation in JupyterHubs have already been implemented. Nevertheless, organizational and legal requirements still need to be developed and evaluated. Investigating these requirements is a high priority for the future.

At this stage of development, a lot can still change on the technical side, and there is a lot of potential to expand the functionality of *JHaaS*. The connection to various storage backends, such as Aruna Object Storage [1], as well as an applicable quota implementation, is currently being investigated. Conceivable extensions would be, for example, support for GPUs, in order to work with GPU-aware software in Jupyter Notebooks, or the connection to learning management systems, such as Stud.IP or Moodle, in order to integrate the use of *JHaaS* more closely into teaching.

In addition, the core of JHaaS can be adapted to deploy other software, such as electronic lab books. Extensions or spin-offs in this direction are also conceivable.

JHaaS is currently available on Github [3] and can be tested on our test setup. Testing by means of an evaluation operation is planned for use in regular courses in the coming winter semester 2023–2024. The further development of *JHaaS* will be largely influenced by the feedback collected from the evaluation operation. Therefore, feedback from course leaders and participants is warmly welcome.

Author contributions

Conceptualization: NM, BL; **Data curation:** n/a; **Formal analysis:** n/a; **Funding acquisition:** AG; **Investigation:** n/a; **Methodology:** NM, BL; **Project administration:** BL; **Resources:** NM, FF, BL; **Software:** NM, BL; **Supervision:** BL; **Validation:** NM, FF, BL; **Visualization:** NM, FF; **Writing – original draft:** NM, FF; **Writing – review & editing:** NM, FF, AG, BL

Competing interests

The authors declare that they have no competing interests.

Funding

FF is funded by the German Research Foundation DFG under the National Research Data Infrastructure–NFDI4Microbiota(NFDI 28/1)–460129525.

References

[1] Aruna Storage Team. "Aruna Object Storage." (2023), [Online]. Available: https://aruna-storage.org/.

[2] WG RSE, NFDI Section Common Infrastructures. "NFDI Jupyter Services." (2023), [Online]. Available: https://nfdi-jupyter.github.io/services/.

[3] JHaaS Development Team. "JHaas Repositories." (2023), [Online]. Available: https://github.com/orgs/JLU-BCF/repositories.