EPiC Series in Computing

Volume XXX, 2022, Pages 50–55

Proceedings of EUNIS 2022 – The 28th International Congress of European University Information Systems

Hybrid Cloud Scaleout: Orchestrating Workloads with GitLab

Marius Politze[1*]

[1] RWTH Aachen University, Germany

politze@itc.rwth-aachen.de

Abstract

The project The FAIR Data Spaces project aims to create a common cloud-based data space for industry and research by connecting services already created in Gaia-X, IDS, NFDI and EOSC, and demonstrates this in its own demonstrators. The demonstrator FAIR Data Quality Analysis and Workflows is giving researchers a platform to define and run workflows for FAIR data, the demonstrator aims to serve as a showcase for a hybrid cloud scale-out scenario. While running user defined workflows on research data stored in git repositories, the essence of the demonstrator is hiding the technical complexity of the hybrid cloud scale-out that is needed to supply the computational power for running the workflow steps. In order to achieve this the demonstrator uses state-of-the-art cloud technologies combined with the most recent developments from the Gaia-X frameworks.

1 Introduction

The FAIR Data Spaces project aims to create a common cloud-based data space for industry and research. It builds on the groundwork of the federated secure data infrastructure Gaia-X, the International Data Spaces (IDS), the National Research Data Infrastructure (NFDI) and the European Open Science Cloud (EOSC). In the created data space, research data in different scientific disciplines and industries will be made available and used according to the FAIR principles (findable, accessible, interoperable, reusable) (Wilkinson, et al., 2016). The project uses its own resources to establish connections between services already created in Gaia-X, IDS, NFDI and EOSC, and demonstrates this in its own demonstrators for a data space on biodiversity, for quality assurance of research data and for cross-platform data analysis.

One of the mentioned demonstrators aims to provide *FAIR Research Data Quality Assurance and Workflows* to researchers based on GitLab and the automation features originally coming from the area of software development namely continuous integration (CI) and continuous deployment (CD). It is

* https://orcid.org/0000-0003-3175-0659

based on the GitLab installation already established at RWTH Aachen University, which is offered as a community Software as a Service (Saas). Among giving researchers a platform to define and run workflows for FAIR data, the demonstrator aims to serve as a showcase for a hybrid (private, community, public) cloud scale-out scenario. Apart from research data management process also teaching activities like the ones designed by Küppers (Küppers, Politze, & Schroeder, 2017) can greatly profit from the possibility of scaling infrastructures as needed.

In the existing environment, users are authenticated via DFN-AAI, the German branch of the eduGAIN federation, and are granted access to resources managed directly by GitLab, in particular Git repositories and user-defined workflows. Decentralized "task runners" use the workflow steps to combine research data management tasks such as loading referenced datasets, quality control, analysis, or publications and automatically apply them to the data. In this scenario, GitLab handles the orchestration of user-defined workflows on the decentralized infrastructure currently provided by the users themselves.

The demonstrator explores two private-public cloud scale-out scenarios on this basis:

- Scaling of compute resources via "Task Runner".
- Scaling of storage resources via referenced datasets.

In the demonstrator, exemplary data from areas of fluid systems engineering and transportation systems are processed.

Figure 1 gives a broad overview of the components that interact within the demonstrator: Technically, individual workflow steps are referenced in the form of readily available Docker images. Orchestration is performed on the basis of user-defined workflows by "GitLab Runner". In the planned scale-out scenario, the GitLab Runner uses a public-cloud-based Kubernetes instance in which both the GitLab Runner and the individual workflow steps are instantiated as containers. The "GitLab Runner" automatically transfers smaller data volumes from the Git repository. Larger data sets must be stored in an object store, referenced and then processed.

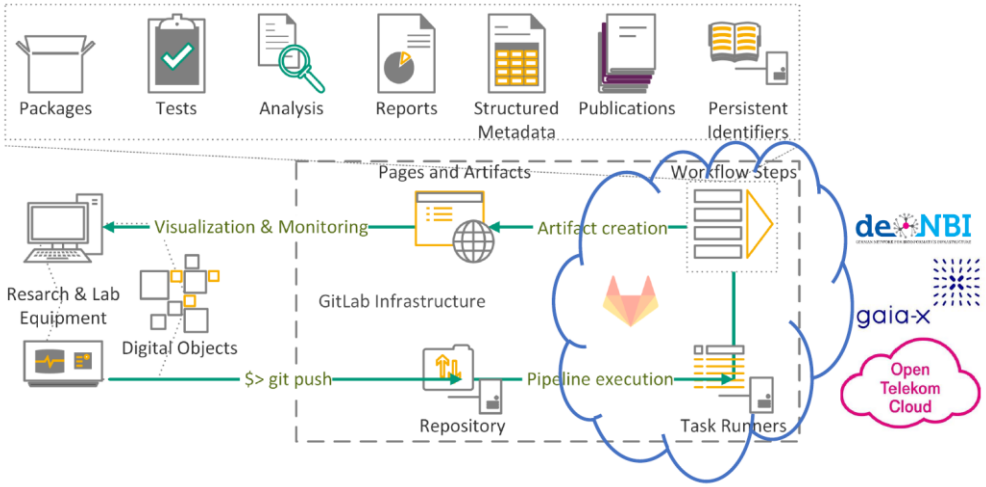


Figure 1: Envisioned Workflow Supported by the Demonstrator (Politze, 2021a)

In particular, the public cloud infrastructure should enable automatic scaling of compute for workflow steps, provisioning and access to object stores. In addition to scalability, the individual workflow steps must be separated from one another to prevent data access by third parties. Role-based access control (RBAC) is also required for the object stores in order to protect data from unauthorized access.

2 Gitlab and GitLab Runner Architecture

Technically the demonstrator relies on readily available cloud technologies and interfaces. Figure 2 gives an overview of the individual architectural components and how they are triggered within the git-CI-based data analysis workflow: The workflow assumes that the research data is stored within the git repository. Adding or changing of files within the repository (git push) then triggers the workflow. The workflow itself is defined within the repository and is then enacted using the GitLab Runner component. Workflow steps for the GitLab Runner are defined using a reference to a Docker image. Technically the GitLab Runner will hence pull the docker image and schedule it within the container runtime. It will then pull the data from the git repository and start the workflow. Intermediate results are stored as artifacts and can be passed on to the next workflow step.

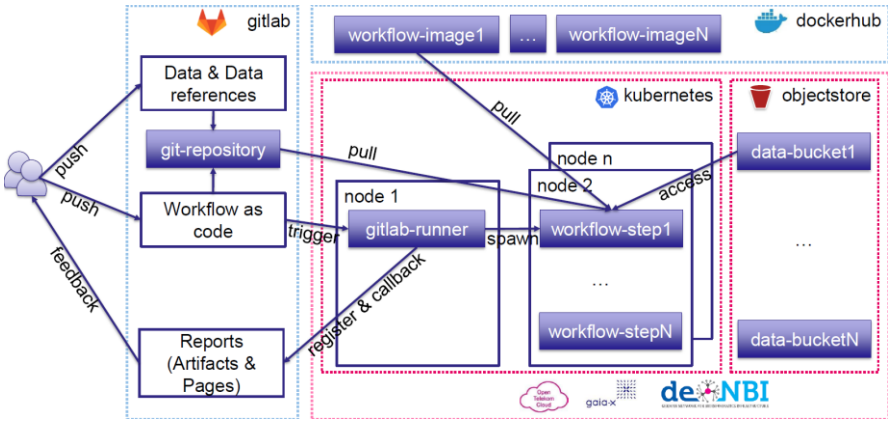


Figure 2: Architectural overview of the Demonstrator (Politze, 2021b)

2.1 Deployment in a Hybrid Cloud Environment

To implement the described process of a private-public cloud scale-out, the components of the demonstrator were instantiated in the different environments. Thus, the existing SaaS solution continues to be used for the operation of GitLab in the private cloud at RWTH Aachen University. For the execution of workflows and the individual workflow steps, the GitLab Runner component is then used to orchestrate the scale-out into public cloud infrastructure. For this purpose, the implementation uses the “Cloud Container Engine” (CCE) offered by “Open Telekom Cloud”.

The SaaS offering provided in the demonstrator consists of several software components that are instantiated in a hybrid cloud scenario. The IaaS (Infrastructure as a Service) and PaaS (Platform as a Service) layers are hidden from the user, as the user only interacts with the instances of the software. Figure 3 shows a corresponding schematic diagram with the separation between the layers. The different color coding shows the different responsibilities for the provision or operation of the individual components in the hybrid cloud.

2.2 Scaling through Gaia-X Compatibility

In addition to investigating the scaling options of the individual components, conformant self-descriptions in JSON-LD format were also generated for the components of the demonstrator. Special attention was paid to the components that are essential for the demonstrator and visible to the user. In addition to the visibility by the users, the technical responsibility for the components in the architecture also plays an essential role here: The components required in the SaaS offering are technically outside

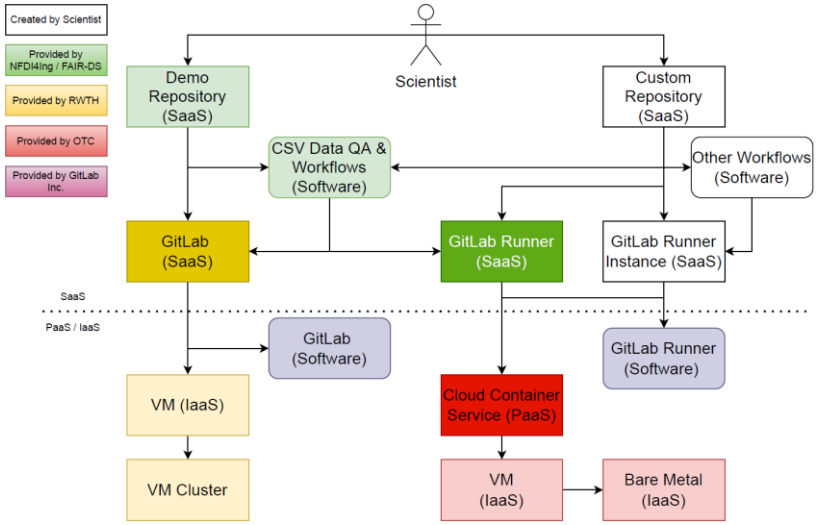


Figure 3: Deployment of Components of the Demonstrator as SaaS, PaaS and IaaS Services

the responsibility of the SaaS operator, so detailed modeling should be carried out by the provider of the IaaS and PaaS solutions.

However, for demonstration purposes, especially of the dependencies between different service offerings, some placeholders have been added for self-descriptions of the very components that lie at the interface between the offerings.

These created self-descriptions could now be used by Federation Services to enable users to access the SaaS and PaaS offerings according to the gaia-x specifications.

The corresponding self-descriptions are available in the project's Git repository under open source license[1]

3 Presentation of FAIR Workflows to the Users

The essence of the demonstrator is hiding the technical complexity of the hybrid cloud scale-out from the users and allowing them to concentrate on the implemented workflow. As such, GitLab remains the primary interface for researchers to interact with.

Having both, the research data and the workflow within the source control mechanisms of git makes the workflow results fully reproducible and not only allows thorough validation of the data but also sharing and reusing of community defined workflows making the methods reproducible (Gundersen, Gil, & Aha, 2018).

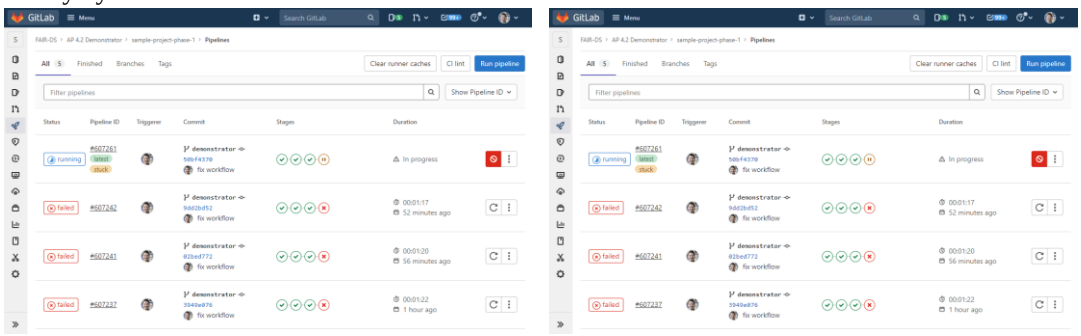
In the first phase of the demonstrator, exemplary data from the area of traffic systems has been used for demonstration. In order not to violate intellectual property rights, the demonstrator uses a data set that is artificially generated but mimics a real data set. In the implemented scenario, the FAIR quality assurance mechanisms and workflows are used to save and process contributions to a traffic count, which are collected by many people at the same time and typically manually. The demonstrator repository is available under open source license[2].

Technically, individual workflow steps are initially referenced in the form of Docker images. Since no dedicated images are currently available for many workflow steps, generic images are usually used,

[1] https://git.rwth-aachen.de/fair-ds/GAIAXServiceSelfDescriptions/-/merge_requests/2
[2] https://git.rwth-aachen.de/fair-ds/ap-4-2-demonstrator/sample-project-phase-1

which in some cases require a great deal of effort to configure. The orchestration is performed by the GitLab Runner in the cloud-based Kubernetes instance, in which both the GitLab Runner itself and the individual workflow steps are instantiated as containers.

The results of individual quality assurance and workflow steps are presented to the users in an overview, so that, for example, errors during input can be identified and improved directly. Figure 4 shows corresponding overviews for quality assurance of manually curated files and workflow steps based on them. The results are visualized using a simple traffic light system and can thus be interpreted directly by the users.



a) Execution status of workflow steps b) Data quality validation results

Figure 4: Workflow results as presented to the user in GitLab

In the demonstrator repository, a simple 4-step workflow is implemented to ensure FAIR principles:

- Validation of manually curated data (Reusable)
- Transformation of the manually curated data into a standardized format (Interoperable)
- Visualization of the data in graphical form (Reusable)
- Making the data available to third parties (Accessible)

The step of publishing the data (Findable) was initially waived for the demonstrator in Phase 1 but is going to be implemented to automatically build data releases and publish them to data repositories like Zenodo.

4 Outlook

The demonstrator shows how a private-public cloud scale-out can be achieved on the basis of an existing community cloud SaaS offering of the GitLab software. In the first phase, the focus was essentially on scaling computing power in the homogeneous environment of Kubernetes. The second phase will specifically consider cloud-based data stores, as well as heterogeneous scaling options. To this end, the demonstrator must also be ported to other public and private clouds such as the DeNBI[3]Cloud or for HPC centers.

While a scale-out can be technically hidden from the user this does not release the users or service providers from distributing resources in an economic manner. In future scenarios resource capacities should be included in a project management system that allows quotation of such shared resources as it has been demonstrated for example in the area of storage systems (Politze, et al., 2020).

Since the Gaia-x specifications are being further developed in parallel with this project, the created self-descriptions will continue to be successively adapted to the innovations. In the process, additional infrastructure components that were not yet considered in phase 1 will also be described.

[3] https://www.denbi.de/

The Demonstrator Repository shown is to be extended by further workflow steps, in particular for publication but also for quality assurance but also for access to external data sources. In the course of the project, a collection of preconfigured workflow steps will be created, which can be used by researchers to define their own workflows based on the FAIR principles.

5 Acknowledgements

The work was supported with resources granted by NFDI4Ing, funded by Deutsche Forschungsgemeinschaft (DFG) under project number 442146713, and FAIR Data Spaces, funded by the German Federal Ministry of Education and Research (BMBF) under funding reference FAIRDS11.

6 References

Gundersen, O. E., Gil, Y., & Aha, D. W. (2018). On Reproducible AI: Towards Reproducible Research, Open Science, and Digital Scholarship in AI Publications. *AI Magazine, 39*(3), pp. 56–68. doi:10.1609/aimag.v39i3.2816

Küppers, B., Politze, M., & Schroeder, U. (2017). Reliable e-Assessment with GIT – Practical Considerations and Implementation. In J. Bergström, *European Journal of Higher Education IT 2017-1*. Umeå, Sweden.

Politze, M. (2021a). *Running FAIR Data Workflows With git and GitLab.* doi:10.6084/m9.figshare.16565976

Politze, M. (2021b). *FAIR Data Spaces AP 4.2 Architecture Overview.* doi:10.6084/m9.figshare.16566138.v2

Politze, M., Claus, F., Brenger, B., Yazdi, M. A., Heinrichs, B., & Schwarz, A. (2020). How to Manage IT Resources in Research Projects? Towards a Collaborative Scientific Integration Environment. In Y. Epelboin, M. Mennielli, A. Pacholak, P. Kähkipuro, G. Ferell, C. Diaz, . . . O. Tasala, *European Journal of Higher Education IT 2020-1*. Paris, France.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., . . . Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data, 3*. doi:10.1038/sdata.2016.18

7 Authors' Biography

Dr. Marius Politze is head of the department "Research Process and Data Management" at the IT Center of RWTH Aachen University. Before that he held various posts at the IT Center as software developer, software architect and as a teacher for scripting and programming languages. His research focuses on Semantic Web and Linked Data architectures for distributed and service-oriented systems in the area of research data management.