Managing FHIR servers with Docker containers for practical courses in university teaching

Johann Frei, Dominik Müller, Frank Kramer, Florian Auer

Angaben zur Veröffentlichung / Publication details:

Frei, Johann, Dominik Müller, Frank Kramer, and Florian Auer. 2025. "Managing FHIR servers with Docker containers for practical courses in university teaching." In *Global healthcare transformation in the era of artificial intelligence and informatics*, edited by John Mantas, Arie Hasman, Parisis Gallos, Emmanouil Zoulias, and Konstantinos Karitis, 491–95. Amsterdam: IOS Press. https://doi.org/10.3233/shti250768.

Global Healthcare Transformation in the Era of Artificial Intelligence and Informatics
J. Mantas et al. (Eds.)

doi:10.3233/SHTI250768

Managing FHIR Servers with Docker Containers for Practical Courses in University Teaching

Johann FREI[a,1], Dominik MÜLLER[a], Frank KRAMER[a] and Florian AUER[a]
[a] *IT-Infrastructure for Translational Medical Research, University of Augsburg, Germany*
ORCiD ID: Johann Frei https://orcid.org/0000-0003-0323-0904,
Dominik Müller https://orcid.org/0000-0003-0838-9885,
Frank Kramer https://orcid.org/0000-0002-2857-7122,
Florian Auer https://orcid.org/0000-0002-5320-8900

Abstract. Resource interference and frustrating experiences for students and instructors often arise from traditional teaching methods involving shared FHIR® server instances. We present a web-based application to automate the deployment and management of isolated FHIR server instances within Docker containers, accessible through an intuitive web interface. This system allows for the easy creation, management, and resetting of individual servers, each pre-populated with sample FHIR data, facilitating hands-on learning. By providing a user-friendly platform with direct access to FHIR resources, our application enhances the educational experience, enabling students to focus on mastering FHIR concepts and practical implementation. This tool effectively addresses the challenges of teaching practical FHIR skills, contributing to a more efficient and effective learning environment.

Keywords. Interoperability, Fast Healthcare Interoperability Resources, Containerization, Automation, Teaching

1. Introduction

The Health Level Seven (HL7®) Fast Healthcare Interoperability Resources (FHIR®) [1] standard has emerged as a pivotal framework for facilitating the seamless exchange of medical data, driving innovation in healthcare information systems [2]. While the theoretical understanding of FHIR is readily accessible, practical experience is indispensable for the effective implementation of FHIR-based solutions. In university settings, in-person sessions allow teachers to react to the individual issues of students and thus enable better progressions. Therefore, guided practical courses on basic and advanced topics play a major role in teaching at universities.

A crucial bottleneck in practical courses is the submission and evaluation of programming code of the course participants. Submissions of solutions for tasks and sub-tasks are usually the basis for deriving grades, and therefore need to be documented. Existing platforms for automated code submission and evaluation offer valuable insights

[1] Corresponding Author: Johann Frei; E-mail: johann.frei@informatik.uni-augsburg.de.

into streamlining this process [3]. However, a particular challenge arises when teaching FHIR, as students often work on a shared instance of a FHIR server. Simple installations typically lack robust user management and resource isolation, leaving students vulnerable to interfering with each other's work. Moreover, manually setting up a FHIR server instance involves steps that may exceed the expertise of novice students in a teaching environment [4]. Consequently, this can lead to frustrating experiences for both students and instructors, hindering effective learning. Conversely, manually setting up and managing numerous server instances is a tedious and time-consuming task. Therefore, there is a need for a simplified and automated process for managing FHIR server instances in educational settings.

2. Methods

To address the challenges of managing FHIR servers in practical courses, we developed a Linux-based web application designed for the automated management of FHIR server instances. This application comprises three primary components: a Python-based web application for managing FHIR server instances through standard web browsers, Linux Bash scripts for performing initiation and management tasks, and a Docker stack for running isolated FHIR server instances. Figure 1 illustrates the architecture of the application, showcasing its three distinct layers. This architectural design is primarily driven by software requirements, but as a consequence, allows for clear separation and isolation of the different application components.

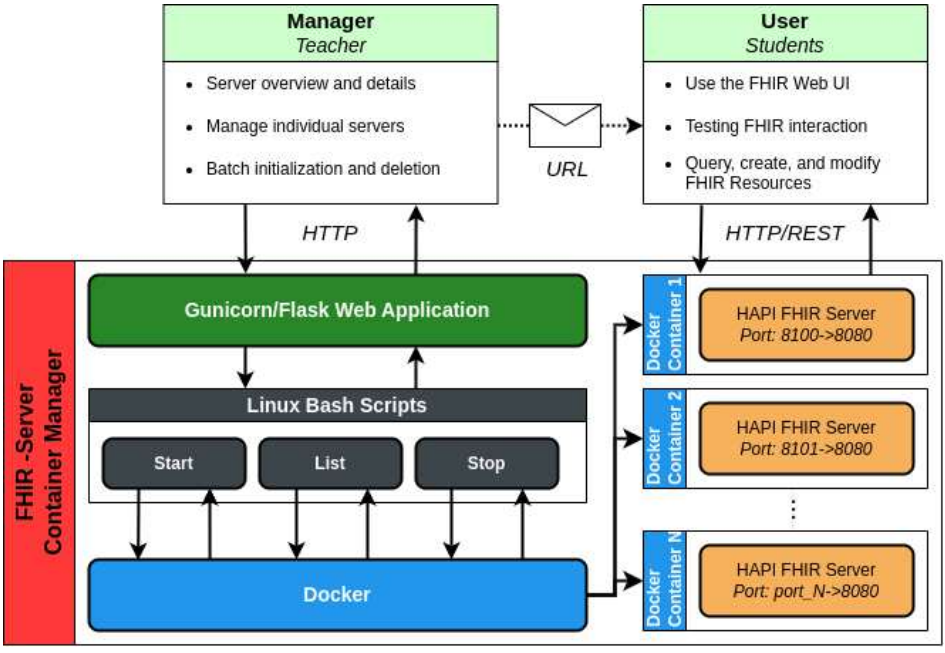

Figure 1. Architecture of the FHIR-Server Container Manager with User and Manager interaction.

The Python web application is built using Gunicorn, a Python HTTP server, and Flask, a lightweight web framework. Gunicorn serves as a robust and scalable HTTP server, efficiently handling concurrent requests, while Flask provides a flexible and minimalist framework for developing web applications. The Bootstrap front-end framework, version 5, is employed to create a consistent and responsive layout throughout the application's user interface.

A Python-based Docker API client and Linux Bash scripts are utilized for automating server setup and initialization. Bash is a powerful command-line interpreter that enables the scripting of complex tasks, making it ideal for automating repetitive processes. These scripts are responsible for setting up and initializing the FHIR server instances. Environment variables are strategically employed to dynamically control the initiation of FHIR server instances, allowing for flexible configuration and customization.

Single FHIR servers are deployed within individual Docker containers, with each server hosted on a distinct port to ensure isolation and differentiation. Docker is a containerization platform that allows applications and their dependencies to be packaged into isolated containers. This approach provides several advantages, including consistent environments, simplified deployment, and efficient resource utilization. The user can define a specific port range for the FHIR servers during the startup of the FHIR container management server to avoid conflicts with additional running services.

3. Results

The web interface of our FHIR server container manager provides a user-friendly and intuitive experience for managing individual FHIR server. New FHIR instances can be easily set up, and existing ones can be efficiently managed. The application facilitates the following actions:

- **Overview of Running Servers:** Provides a comprehensive view of all currently active FHIR server instances servers, including their status and essential access links (Fig. 2A).
- **Start and Stop of Individual FHIR Servers:** Enables the initiation of individual FHIR servers with a single button click. This action triggers the creation and startup of a new, isolated Docker container hosting the HAPI-FHIR server, pre-populated with predefined exemplary data.
- **Comprehensive Information about FHIR Server Instances:** Displays detailed information about each FHIR server instance, including its status, direct links to the FHIR server's REST API, its FHIR web UI, and an exemplary query for Patient resources (Fig. 2B).
- **Server Management:** Enables users to directly reset, stop, and remove individual server instances via the web interface, providing them with full control over their server environment.
- **Batch Deletion:** Provides the capability to delete all running FHIR server instances in a single operation.

Upon initiation, each FHIR server is pre-populated with sample data for three interconnected FHIR resources: Patient, Organization, and Encounter. This ensures consistency for each student, providing a standardized starting point for practical

exercises. This data illustrates the basic use of FHIR, serving as a foundation for advancing in learning FHIR. Furthermore, the ability to easily reset the server instances through the web application allows students to quickly recover from mistakes, fostering a more forgiving and effective learning environment.

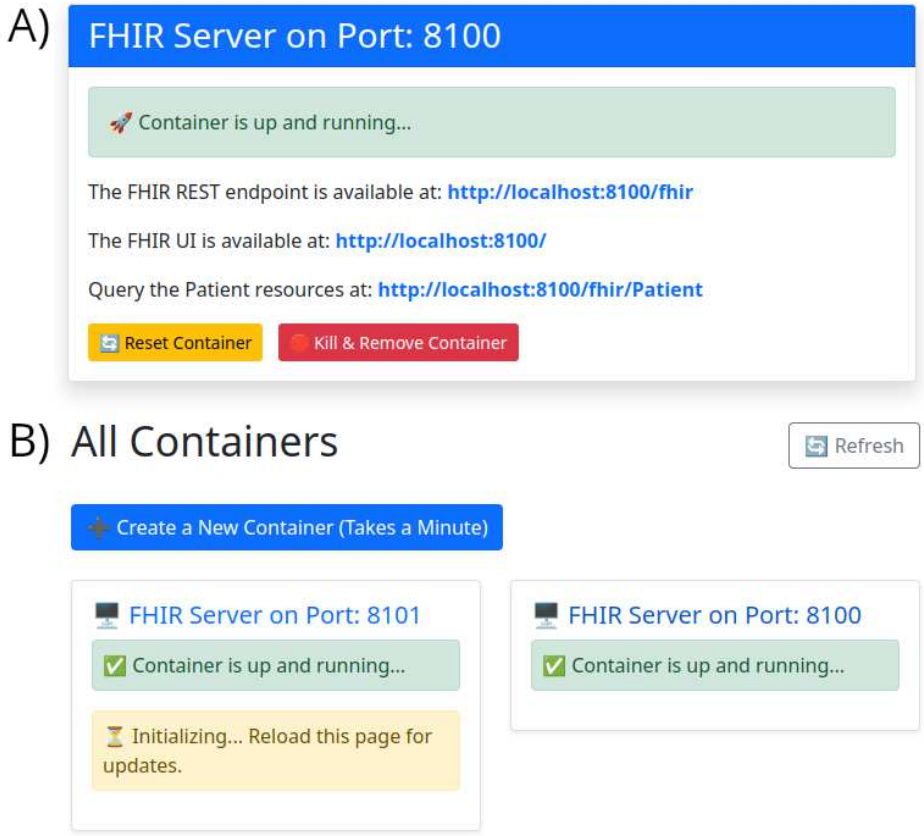

Figure 2. User interface of the FHIR-Server Container Manager. Option for the interaction with single containers are provided (A), as well as details about all running containers and corresponding states (B).

4. Discussion

The development of our FHIR-Server Container Manager significantly improves practical FHIR education by automating the deployment and management of isolated server instances. This approach allows students to engage with real-world FHIR servers in a controlled environment.

However, integrating user authentication and authorization would allow for more precise control over server access and resource management, enhancing security and traceability. Expanding the sample data to include a broader range of FHIR resources and use cases would further enrich the practical learning experience. Supporting other FHIR server implementations would increase the application's versatility and applicability across different educational and infrastructural contexts.

While the integration of containerized FHIR server management may appear to be an incremental improvement—given that similar approaches exist within DevOps and educational contexts—we argue that its value lies in its educational orientation. Traditional DevOps tools such as Ansible AWX or Kubernetes-based workflows focus primarily on scalability and reliability in production environments. Our solution repurposes these container orchestration principles specifically for FHIR education, emphasizing student-centric features such as per-user isolation, simplified deployment, and integration with learning tools.

Future work includes extending the REST API to enable deeper integration with educational platforms such as misitCMS[3]. Preliminary teaching sessions demonstrated successful integration for provisioning individualized FHIR validation servers during code submissions. While full evaluation is pending, initial feedback is promising, and further validation in both grading systems and real teaching environments is planned.

Finally, we note that existing literature and research on practical FHIR server education, particularly in terms of automated deployment, evaluation, and student integration, remain limited. This underscores the need for more studies and contributions in this niche intersection of health informatics education and educational technology.

5. Conclusions

Our FHIR-Server Container Manager provides a practical and efficient solution for teaching FHIR in university settings. By automating the deployment and management of isolated FHIR server instances, we address critical challenges related to resource interference and user isolation. The intuitive web interface, coupled with pre-populated sample data and direct access to FHIR resources, facilitates hands-on learning and empowers students to explore FHIR concepts effectively. The ability to quickly reset server instances supports iterative learning and experimentation. This tool contributes to a more streamlined and effective learning experience, ultimately fostering a deeper understanding of FHIR and its practical applications in healthcare data interoperability.

The source code for the FHIR server container manager described in this paper is available on GitHub at https://github.com/frankkramer-lab/FHIR-Container-Manager/. The repository includes the Python web application, Linux Bash scripts, and Docker configuration files necessary to deploy and manage the system.

References

[1] HL7 FHIR [Internet]. [cited 2025 Mar 10]. Available from: http://hl7.org/fhir/
[2] Duda SN, Kennedy N, Conway D, Cheng AC, Nguyen V, Zayas-Cabán T, et al. HL7 FHIR-based tools and initiatives to support clinical research: a scoping review. Journal of the American Medical Informatics Association. 2022 Sep 1;29(9):1642–53.
[3] Auer F, Frei J, Müller D, Kramer F. Perspective on Code Submission and Automated Evaluation Platforms for University Teaching. Stud Health Technol Inform. 2022 Jun 6;290:912–6.
[4] Hussain MA, Langer SG, Kohli M. Learning HL7 FHIR Using the HAPI FHIR Server and Its Use in Medical Imaging with the SIIM Dataset. J Digit Imaging. 2018 Jun 1;31(3):334–40.