Gottfried Wilhelm Leibniz Universität Hannover
Faculty of Business Administration and Economics
Institute of Information Systems

Developing and Evaluating LLM Applications for Analyzing Containerized Software Log Data Using the RAG Framework: A Case Study Approach Examining Opportunities, Risks and, Potential

Master Thesis

For the award of the academic degree “Master of Science (M.Sc.)” in the Master’s degree program in Economics of the Faculty of Economics and Faculty of the Leibniz University of Hannover

Submitted by:

Name: Graw, Philipp Alexander
[redacted]
[redacted]

Examiner: Prof. Dr. Michael H. Breitner
Supervisor: Fenja Schulte

Hannover, 06.03.2025

Table of Contents

1. Introduction

1.1 Problem Statement and Motivation

The adoption of container technologies such as Docker and the increasing popularity of Kubernetes as an orchestration platform have significantly changed modern software development and deployment. Containers enable a unified and portable environment, while Kubernetes has established itself as the leading solution for managing containerized applications and therefore as a central component of modern cloud infrastructures. Regardless of whether private clouds, public clouds, hybrid cloud solutions, multi-cloud environments or even bare-metal applications are involved, Kubernetes serves as the central framework for container orchestration in cloud infrastructures (Kubernetes a, retrieved 31.10.2024).

Despite the increasing spread of Kubernetes, companies face considerable challenges when using this technology. Around 76% of companies state that the high complexity of Kubernetes significantly delays the adaptation of their cloud infrastructures (SpectroCloud, 2024). Application migrations come to a standstill and newly developed applications are often used incorrectly. This can lead to considerable costs. In addition, retraining employees is time-consuming and costly. Since cloud infrastructures are often highly customized to the individual needs of customers, general training programs often reach their limits and offer only limited added value. These challenges highlight the need for innovative solutions to improve the efficiency and effectiveness of Kubernetes in cloud environments. In this thesis, we investigate whether GenAI can provide a promising solution to these challenges. To investigate this, we develop an application prototype based on RAG and GenAI. The prototype is designed to analyze relevant data streams from Kubernetes clusters and makes it available to developers in a searchable, natural language format, which simplifies error analysis and problem solving. The concepts of RAG and agent-based AI frameworks are given special consideration in this study. The case study aims to evaluate the potential, risks and opportunities of this approach. Finally, further fields of research and development perspectives are identified.

1.2 Objective of the Thesis

This master thesis aims to develop an approach to solve practical problems related to the complexity of modern software systems using RAG and LLMs. To achieve this, we have designed a RAG-based system architecture that serves as a basis for future developments.

The architecture developed in this thesis serves as a proof of concept for a system that leverages RAG technology to provide detailed insights into underlying systems.

In our specific use case, we design the framework to process Kubernetes cluster data. Through literature research and expert interviews, we identified relevant data streams

and components within a modular RAG framework. The system aggregates this data and seamlessly integrates it into an LLM using various RAG-based concepts. This framework enables natural language search capabilities, allowing users to access and understand data streams, even in complex architectures.

This thesis evaluates the potential, risks and opportunities of using RAG to process data from complex software systems like Kubernetes. The architecture we developed serves as a foundation for future implementations and evaluations, with applications that could extend beyond Kubernetes to other complex software systems. For Kubernetes specifically, our solution aims to lower entry barriers and reduce complexity by making this backbone of modern cloud infrastructure more accessible, ultimately helping bridge the DevOps cultural gap and accelerate cloud adoption.

Finally, the thesis presents three artefacts.

1. The risks, opportunities and potential of combining complex software systems with GenAI-RAG applications - specifically for the use case of Kubernetes.
2. The overall problems faced when working with Kubernetes.
3. An architecture for an application dedicated to simplifying and automate the work with complex software systems such as Kubernetes.

The architectural artifact incorporates the most relevant RAG components and advanced enhancement strategies, including model fine-tuning and AI agents. It particularly focuses on all relevant Kubernetes cluster data sources, which were identified through expert interviews conducted for this research.

1.3 Research Question

The three central research questions addressed in this paper are as follows:

1. What issues do companies face when dealing with Kubernetes?
2. How can an architectural framework for a RAG-GenAI application be designed to simplify or solve these identified problems?

The second research question aims to identify relevant data sources, RAG components, fine-tuning methods and AI agent enhancements. The primary focus is on identifying the relevant data and modular RAG components needed, along with their specific implementation for the Kubernetes case study.

3. What are the risks, opportunities and potential benefits of using RAG-GenAI applications to analyze Kubernetes systems?

1.4 Structure of the Thesis

The paper begins with foundational concepts, covering DevOps, cloud computing, containers, Kubernetes, RAG and Agentic AI. Next, we outline our methodology, which

combines the Design Science Research (DSR) Framework with the BAUSTEIN Framework. Our knowledge base is built through comprehensive literature research and expert interviews. We conducted interviews using two distinct guidelines: one for Kubernetes experts and another for AI specialists. This approach allowed us to gather comprehensive insights about problems, required data sources, and implementation and design considerations. Based on these insights, we develop an end-to-end architecture for a RAG-GenAI application with a Chat Interface designed to address the priorly identified problems. Additionally, we aim to examine the associated risks, opportunities and potential benefits. The main section presents our identification of relevant data sources, exploration of RAG components, fine-tuning approaches and agentic AI applications. We examine the challenges of working with Kubernetes, detail the architectural design and derive the design knowledge artifact. The thesis concludes with an analysis of identified risks, opportunities and potential benefits, along with a synthesis of key findings and recommendations for future research.

2. Theoretical Framework

2.1 DevOps

Development & Operation (DevOps) is a concept described as a cultural change within software development and IT operations teams (Khattak et al., 2023). It promotes collaboration, integration and communication between teams to increase development speed, frequency and quality (Khattak et al., 2023). DevOps is seen as a combination of practices and cultural movements aimed at breaking down the barriers between development and operations teams to build better software faster and more reliably (Jayakody et al., 2023). DevOps is characterized by the promotion of collaboration between development and operations teams to minimize conflicts in software delivery. It is seen as a cultural change within organizations that emphasizes open communication and shared responsibility. A central role is played by the automation of processes, which increases efficiency and reduces human error. DevOps also enables real-time feedback from customers so that products can be better customized. Continuous integration and deployment improve the quality and speed of software development. Clear responsibilities and incentives promote team motivation and performance, while transparency, trust and respect strengthen collaboration and form the basis for DevOps success (Jayakody et al., 2023).

The DevOps culture is more than just a collection of tools and practices. It embodies a comprehensive cultural transformation that fundamentally reshapes collaboration and communication between development and operations teams. At its core is open communication that keeps all team members informed of progress and challenges and shared responsibility for project success (Jayakody et al., 2023).

The culture is built on the promotion of trust, respect and transparency, creating a positive working atmosphere. Clear incentives and accountability motivate teams, while acceptance of cultural change and permanent training ensures adaptability to

Cultural Adoption Barriers: Finally, experts warned about resistance to change and training needs, stating that organizational resistance could hinder AI adoption (EK1, EK2, EK3, EK4, EK5, EA1, EA2). Ensuring effective training and fostering a culture of AI acceptance were seen as essential for successful implementation.

Experts recommend several key measures to mitigate AI-related risks in Kubernetes. AI should serve as an assistant rather than a replacement for human expertise, with human oversight remaining crucial for safety-critical decisions (EK4). Data security risks could be tackled by applying strict RBAC policies to ensure that AI systems only access authorized data, preventing security breaches (EK2, EK6). Furthermore, systems must be transparent, providing clear explanations of their decision-making process to build trust (EK4). Monitoring and feedback mechanisms should validate AI recommendations through human review (EK4). Organizations must also focus on cultural adaptation, emphasizing AI as a supportive tool and providing proper training to ensure smooth adoption (EK6). Another crucial point is smart data processing through pre-filtering. This approach optimizes system efficiency and prevents scalability issues by processing only essential data (EK6).

8.4 Conclusion

The integration of AI into Kubernetes presents significant potential for enhancing error detection, automation and management of Kubernetes Clusters. Experts highlight key benefits, including the acceleration of error diagnosis through automated root cause analysis, proactive detection and prevention of incidents to reduce mean time to resolution (MTTR) and improved accessibility of Kubernetes for non-experts through natural language interfaces. Additionally, AI can facilitate the automation of repetitive tasks, such as manifest creation and security analysis and has the potential to transform DevOps workflows through AI agents and GitOps integration as well as the integration into ticketing platforms for an end-to-end process coverage.

However, the adoption of AI-driven solutions in Kubernetes requires a well-defined risk mitigation strategy. Key challenges include hallucinations or unreliable results in general, security vulnerabilities and problems on how to handle RBAC, scalability concerns, costs, over-reliance on AI, biases in decision-making and implementation complexity. Ensuring transparent, access-controlled and explainable AI models with human oversight is critical to harnessing these technologies effectively without compromising infrastructure security and stability.

An effective communication of the results should consider the following stakeholders.

Stakeholder	Role	Level of Engagement	Ideation Contribution
Project Team	Developers, researchers, designers	High	Lead design and development of the MVP
Kubernetes Administrators	End-Users	High	Provide Insights - operational challenges and needs
CNCF	Industry standards	Low	Guidelines - best practice and trends
IT Managers/ Executives	Decision Makers	Medium	Influence Budget allocation and project organisation
Investors/Sponsors	Financial Support	Medium	Evaluate project viability and ROI
AI/ML Experts	Technical Advisors	High	Contribute expertise on RAG and LLM Integration
Industry Partners	Collaborators and users	Medium	Share additional use cases and co-development solutions
Regulatory Bodies	Compliance and governance	Medium	Ensure adherence to industry regulations and standard

Table 23 - Stakeholder Analysis - Own Illustration

9. Limitations

The artifact developed here is merely an entry point and must be continuously developed further. As the application has not been implemented, no tests or evaluations could be carried out. This was also a limiting factor in implementing the feedback of the experts. It is important to test the application and the RAG implementation against various benchmarks regarding common metrics such as accuracy and to continuously develop them. Furthermore, data must be continuously collected for the further development and above all, the fine-tuning of the application and used to train the individual LLM components. To guarantee a strong system in the long term, a foundation model specifically tailored to Kubernetes should be trained as a generator.

One limitation of the approach is the purely theoretical foundation of the artifact. Although the AI experts bring practical know-how to the development of the artefact and are familiar with the implementation and processing of complex data, there are only three AI experts, which results in a relatively low input of GenAI expertise.

The semi-structured interview methodology inherently introduces potential framing bias, where the structure and wording of questions may inadvertently influence participant responses and shape the direction of the conversation in ways that could