National University of Computer and Emerging Sciences, Lahore

MLOps-Driven HPC Infrastructure

Fatima Siddiqui 20L-0970 BS(SE)

Hassan Rehman 20L-1280 BS(SE)

Usman Faisal 20L-1385 BS(SE)

Supervisor: Mr. Usama Hassan Alvi

Final Year Project

June 9, 2024

Anti-Plagiarism Declaration

This is to declare that the above publication was produced under the:

Title: MLOps-Driven HPC Infrastructure

is the sole contribution of the author(s), and no part hereof has been reproduced as it is the basis (cut and paste) that can be considered Plagiarism. All referenced parts have been used to argue the idea and cited properly. I/We will be responsible and liable for any consequence if a violation of this declaration is determined.

Date: June 9, 2024

Name: Fatima Siddiqui

Signature: .....

Name: Hassan Rehman

Signature: .....

Name: Usman Faisal

Signature: .....

Author's Declaration

This states Authors' declaration that the work presented in the report is their own, and has not been submitted/presented previously to any other institution or organization.

Abstract

The primary objective of this project is to provide a tailored, on-premises, High-Performance Computing infrastructure for implementing and utilizing AI/ML technologies for organizations, particularly SMEs. This project also recognizes barriers such as intricate infrastructure setup, security concerns, and the dynamics of adapting AI models, thus also aims to simplify AI/ML infrastructure deployment and management. The key deliverables includes an automated configuration of GPU servers, event-driven operations, pre configuration of tools for AI/ML Development, a user-friendly management platform, and finally the integration of reporting tools like Prometheus. The solution assures that organizations, regardless of their technical skills or resources, so they can use AI/ML technologies for effective decision-making and innovation while maintaining data security and lowering operating costs.

Executive Summary

In today's advanced world of Artificial Intelligence and Machine learning, the need of a robust and efficient infrastructure has never been more crucial. Especially for the Small and Medium Enterprises (SMEs), therefore our project "MLOps-Driven HPC Infrastructure" investigates and proposes a solution to the complexity of establishing an on-premises high-performance computer infrastructure specialised for AI/ML applications.

The discipline of AI and ML has made great development recently, with a huge growth in the demand for effective and scalable solutions. Nonetheless, because to the continuously changing nature of data and the challenges associated with maintaining extensive infrastructure settings, SMEs have difficulty efficiently adopting AI models into their operations. Using external cloud platforms to distribute sensitive data raises security concerns and incurs continuous costs, necessitating the knowledge of specialists.

Consequently, the main objective of our project is to tackle the diverse problems that many SMEs are facing. These issues include security vulnerabilities, adherence to regulatory requirements, financial implications related to expenses connected with cloud services and the intricacies involved in establishing and overseeing AI/ML infrastructure. The key objective is to create a sophisticated HPC infrastructure solution that is automated, adaptive, secure, and also user-friendly.

The literature review section offers a thorough and all-encompassing examination of previous related works on the idea and also currently available related software about the subject matter. The discourse highlights the prevalent knowledge, ideas, and thoughts regarding the topic, emphasising their strengths and faults. Moreover, the research also highlights the significance of using on-premises HPC technologies, particularly in scenarios involving sensitive data problem, due to their ability to provide heightened security and data management.

Among the key deliverables of the project is to provide an automated setup process for GPU servers inside the Virtual Machine because these servers serve as the fundamental infrastructure for any AI/ML development. The project provides the devs with pre-configured tools which are specifically designed for the creation of artificial intelligence and machine learning, in order to assist them in their work and save time.

The report's Risk Analysis section explains potential risk that might occur in this project. One of the primary concerns is the challenge of sourcing hardware particularly in acquiring high-performance infrastructure components. Therefore, such challenges could lead to delays in the project setup and deployment. Another significant risk we identified is software compatibility issues, emphasizing the importance of ensuring that all software components function harmoniously. Furthermore, data security

and privacy are top priorities, with the requirement to protect the safety and security of sensitive data in order to avoid any legal and reputational challenges.

Moreover, the high-level and low-level design chapter outlines the advanced system architecture comprised of four primary components: Frontend, Backend, Kubernetes, and Helm Charts & Operators. These components are meticulously engineered to assure efficiency, scalability, and adaptability, making them especially beneficial for SMEs handling complex AI/ML systems. Also the system utilizes a range of technologies such as Next.js, MongoDB, Go, Python, Kubernetes, Helm Charts, and Ansible which are all integrated within a microservices architecture. Additionally, the document include a detailed class and a sequence diagrams which provides a more clearer understanding of the system's operational framework.

Next, the implementation and test cases chapter explains the details on how the system actually works and outlines the test cases on which the system was tested. This begins with the installation of a standalone Ubuntu system and the activation of virtualization in BIOS. Following that, the chapter walks through the installation of requirements such as Vagrant for VM setup, Ansible for automation, and VirtualBox for virtualization. Moving on, details of backend and frontend servers are provided. Finally, the implementation details of authentication and queue management system is also provided. In the next part, the test metric tables of both functional and non functional are also giving which provides an accuracy estimate for the system.

Finally, the user manual section provides detailed guidance for new users to install, configure, and effectively utilize the system. It covers setup requirements like Ubuntu OS and essential tools such as Docker and Kubernetes. The manual also provides a detailed step-by-step installation instructions. It also guides through initial system setup, including root account configuration and service account creation for Kubernetes. Moreover, it also describes the system's interface and navigation for the users.

In conclusion, this report offers a comprehensive perspective on the requirements of our project, the challenges faced, and architecture and solutions for crafting a robust HPC infrastructure specifically tailored for AI/ML applications. The main focus on on-premises solutions ensures data security and provides cost-effectiveness and adaptability, establishing it as a viable option for SMEs looking to enter the AI-driven arena.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

In today's digital age, organizations from various sectors face the challenge of maintaining huge and constantly changing datasets. This increase has been accompanied by a compelling need to utilize these datasets to derive actionable insights, especially for advancements in Artificial Intelligence and Machine Learning. Over the past decade, many of these companies have made substantial investments in High Performance Computing. Essential activities like data preprocessing, model training, and hyperparameter tuning have become complex due to the vast data of computation nodes, therefore requiring sophisticated hardware accelerations such as GPUs.

While HPC has traditionally been the mainstay for specialized engineering and scientific applications, AI and deep learning have opened up new horizons. These AI-driven methodologies, popularized by tech titans, are now being used across sectors for purposes ranging from autonomous driving to drug development. Therefore, given the similarities in infrastructure requirements, such as GPU-accelerated compute nodes and expansive storage systems there is a compelling case for integrating HPC and AI/ML workflows. A study titled "Big enterprise registration data imputation: Supporting spatiotemporal analysis of industries in China" by [1]. emphasizes on the importance of HPC frameworks which addresses big data computational issues, particularly in the context of enterprise registration data, which can be seen as a testament to the importance and advantages of on-prem HPC for SMEs. The on-prem HPC offers substantial cost savings over the long term, especially when considering data transfer costs associated with cloud solutions. Moreover, on prem HPC provides more enhanced security measures, ensuring that sensitive data remains within the organization's control and is not exposed to potential external threats. Furthermore, the performance of on-premises HPC systems may be adjusted to the organization's unique demands, providing maximum computing efficiency.

Among the complexity created by large datasets and the sophisticated processes of AI and ML, our project strives to deliver a solution customized to provide such organisations with a simplified infrastructure configuration. Automating the deployment of essential tools and libraries on GPU bare-metal servers and providing pre-configured tools for AI/ML development, data storage, and management eliminates the task of building from scratch and utilizing extra resources. In essence, our project not only bridges the gap between HPC and modern AI/ML methodologies but also democratizes access to advanced AI capabilities, enabling organizations to harness the full potential of their data without being encumbered by infrastructural limitations.

1.1 Purpose of this Document

The purpose of this document is to comprehensively outline our Final Year Project (FYP), which focuses on alleviating the challenges organizations face in effectively utilizing their expansive datasets for advancements in AI and ML. This project is crucial in the current digital transformation era since efficient data management and utilization are critical.

Our project's main aim is to provide streamlined solution by integrating on-premises High-Performance Computing (HPC) to tailor the infrastructure for AI and ML needs. We recognize the need for a web interface that allows organizations to customize their deployments according to their unique data and computational requirements.

The primary research question guiding our project is: "How can we design an on-premises solution that simplifies leveraging extensive datasets for AI and ML innovations?" This report will thoroughly elucidate our project's vision, define the problem domain, and comprehensively analyze the challenges that warrant this solution. Furthermore, it will outline our specific objectives and goals, shed light on the employed methodology, and delve into the technical aspects of our solution.

The report will culminate by presenting our project's design, implementation, testing, and evaluation processes. Additionally, it will address any limitations encountered during this project and propose potential future developments to enhance further the integration of on-premises HPC in aiding SMEs and organizations to unlock the true potential of their data for AI and ML applications.

1.2 Intended Audience

The primary intended audience for this FYP report on AI and ML infrastructure development and management is a diverse group of stakeholders with a vested interest in AI and ML technologies' advancements and practical applications. Given the project's technical orientation, the primary readership is expected to be:

1. **FYP Evaluators:** A panel of professors from FAST, especially of Computer Science, AI, and ML domains. These individuals will critically evaluate the project's technical depth, methodologies, and contributions to the field.
2. **Industry Professionals:** Given the project's emphasis on practical solutions for AI/ML infrastructure challenges, professionals from the tech industry, especially those involved in AI/ML infrastructure setup, management, and automation, such as our external advisory's company, Questra Digital will find the report's findings and solutions valuable.

3. **Organizational Decision-Makers:** Leaders and decision-makers from SMEs looking to integrate or enhance AI/ML capabilities within their operations. They would be keen on understanding the feasibility, benefits, and potential ROI of implementing the proposed solutions.

4. **Research Scholars and Students:** Those pursuing research or academic interests in HPC, AI, ML, and related infrastructure might find the report valuable for understanding current challenges and solutions.

5. **Open-Source Community:** Developers, contributors, and enthusiasts from the open-source community interested in AI/ML infrastructure. Given the project's open-source nature, this audience can benefit from the report's findings, potentially contributing to, adapting, or extending the project's solutions for broader applications and use cases.

1.3 Definitions, Acronyms, and Abbreviations

The list of abbreviations used in this document is as follows:

SDG: Sustainable Development Goal

FYP: Final Year Project

UI: User Interface

UX: User Experience

API: Application Programming Interface

AI: Artificial intelligence

ML: Machine Learning

SME: Small Medium Enterprise

AWS: Amazon Web Services

GPU: Graphics processing unit

MLOps: Machine Learning Operations

DevOps: Development and Operations

HPC: High Performance Computing

RBAC: Role-Based Access Control

PV: Physical Volume

PVC: Persistent Volume Claim

1.4 At the End of Introduction Chapter

The report begins with an **Introduction** that sets the context and defines the scope of the document. The **Project Vision** chapter delves into the problem domain, elaborating on the goals, objectives, and broader scope of the project. A comprehensive **Literature Review** follows, discussing the current state of *AI/ML infrastructure*, *high-performance computing in AI/ML*, and the debate between *Cloud vs. on-premises solutions*. The **Software Requirement Specifications** chapter provides a detailed breakdown of the software's features, requirements, use cases, and user interface, along with a thorough risk analysis.

Chapter 2 Project Vision

Our project main goal is to help businesses and SMEs of any size or financial capabilities may harness the power of developing and managing AI/ML for the purpose of improving operational efficiency and decision-making processes.

Furthermore, this vision involves the demand for a flexible infrastructure capable of supporting a diverse variety of data requirements while also assuring precise resource control and optimization. Automation and streamlined procedures are critical for this approach since they inspire innovative techniques and accelerate the production and deployment of AI/ML models.

In short, to make the AI/ML infrastructure environment more user friendly and more accessible also with an emphasis on using SME-specific on-premises HPC. The importance of on-premises HPC in this project cannot be overstated as it provides SMEs with a solid foundation for their needs and also working it into the AI and ML domains and catering to their unique requirements and constraints.

2.1 Problem Domain Overview

The main concern of this project is with HPC infrastructure for deploying clusters in order to managing AI/ML models. Moreover, employees lacking extensive technical knowledge can get benefits of these advanced methodologies. Numerous organizations including SMEs, government agencies, hospitals and institutions which are confronted with the growing complexities of managing and processing massive datasets. Often, they contain sensitive and critical information; these datasets are integral for driving advancements in areas such as AI/ML. However, setting up, configuring, and managing the AI/ML operations infrastructure can be daunting. The lack of streamlined processes and the intricacies involved in hardware and software setups often act as barriers which prevents many entities from fully harnessing the potential of AI and ML technologies.

Key components of the project domain includes the development of AI and ML infrastructure which comprises both hardware e.g. GPU servers and software e.g. Kubernetes clusters, JupyterHub, Kubeflow and MLflow. Providing the option for customization and recognizing the diverse requirements of organizations, and the project aims to create adaptable solutions tailored to specific data and operational needs. Automation is critical for simplifying complicated operations such as GPU server setup, Kubernetes deployment, and ML project management, eventually improving efficiency and lowering entry barriers to AI/ML adoption.

Effective data storage and management are critical aspects addressed by exploring various solutions, such as LakeFS and Scaladb. User accessibility is another priority, emphasizing user-friendly interfaces

like JupyterHub. Furthermore, the domain includes MLOps, which focuses on holistic management, including training, deployment, and monitoring. To extract relevant insights, business intelligence and reporting technologies such as Metabase are connected. Scalability is critical, and the project is looking at effective approaches to scale AI/ML applications.

While we are primarily focused on on-premises infrastructure, we are aware of the security challenges associated with cloud-based solutions. A study [2] which reported on the security challenges and practicalities within a cloud HPC project that used very sensitive and confidential data. The study emphasized the security concerns when moving services from on-premise into a public cloud HPC. This underscores the importance of on-premises HPC solutions especially when dealing with sensitive data, as they offer enhanced security and control over data.

2.2 Problem Statement

Developing a customised AI/ML infrastructure is critical for businesses, but conventional solutions frequently fall short of unique requirements. SME's struggle to adjust AI models to changing data and deal with complicated infrastructure settings. Deploying sensitive data on external cloud platforms raises security risks as well as ongoing expenditures, necessitating the services of specialised specialists.

This project addresses the above challenges by developing an automated, adaptable, secure, and user-friendly AI/ML infrastructure solution for SMEs. The solution is based on on premises HPC which grants SMEs control over their hardware and reducing operational expenses. It ensures data security by keeping sensitive data on its servers and streamlining AI/ML tool management and deployment. This integration will improve the productivity, agility and data utilisation, allowing SMEs to thrive in an AI-driven world.

2.3 Problem Elaboration

There are multiple challenges faced by the SMEs. First of all, security and compliance risks occur when deploying sensitive data on external cloud platforms, so the financial burden associated with recurring cloud service costs and the need for specialized expertise, the complexity of setting up and managing AI/ML infrastructure, lack of customization in generic cloud-based AI/ML solutions, difficulties in adapting AI models to evolving data dynamics, and intricacies in configuring and optimizing the AI/ML infrastructure. Addressing these issues is crucial for democratising AI/ML, making these technologies more accessible and efficient for organisations, particularly SMEs, and encouraging innovation across several industries.

Complexity in Infrastructure Setup and Optimization: Creating an infrastructure for AI/ML project development necessitates substantial knowledge and adequate resources. Configuring GPU bare-metal servers, installing Kubernetes clusters, and optimising AI/ML technologies such as JupyterHub, Kubeflow, and MLflow can be complex and time-consuming tasks. Simplifying and automating these tasks is vital for reducing the barrier in entry for AI/ML adoption.

Data Dynamics and Model Adaptability: The changing nature of the data adds the difficulty to update AI models to the most recent data. As a result, keeping AI models up to date with constantly changing data may be a difficult and time-consuming task. It is critical to have a system that can handle the shifting nature of data while also ensuring that AI models stay relevant and correct.

Customization and Flexibility: Generic cloud-based AI/ML solutions may not meet the particular and growing demands of SMEs. These systems frequently lack the adaptability to unique data types, operational needs, and industry peculiarities. As a result, we must design AI/ML models and infrastructure to correspond with the organization's particular objectives, and the data landscape is critical for obtaining relevant insights and achieving optimal performance.

Professional Expertise and Resource Dependency: Setting up the AI/ML infrastructure in traditional cloud platforms often require specialized knowledge and also expertise. But, SMEs in particular may lack the internal resources and the expertise staff to configure and manage complex cloud based AI/ML environments. This causes the need for hiring professionals with the requisite skills involves additional costs so making setting up and managing AI/ML infrastructure an expensive affair.

Security Concerns: Deploying sensitive data on external cloud platforms like AWS poses significant security risks. Confidentiality, integrity and data availability become paramount concerns especially when dealing with critical organizational information. Therefore, it is of utmost importance to handle this concern with high priority. Also, to ensure data security and compliance with multiple regulatory frameworks, data deployment alternatives must be reevaluated.

Cost Implications: Using well-known cloud services such as AWS incurs regular charges, potentially burdening SMEs with significant financial responsibilities. These costs might include data storage fees, compute fees, and fees for specialized AI/ML services. Furthermore, hiring people with expertise in cloud architecture and machine learning to manage and optimize these cloud resources would increase the cost of the organization.

Addressing the multiple challenges above is really fundamental to realizing the project's vision of democratizing AI/ML, enabling organizations, especially SMEs to seamlessly utilize AI/ML technologies. By focusing on enhancing security, reducing costs, simplifying infrastructure setup ensuring flexibility and customization, accommodating data dynamics and easing infrastructure complexity and optimiza-

tion. This project aims to revolutionize how organizations approach and benefit from AI/ML, ultimately driving innovation and growth across diverse sectors.

2.4 Goals and Objectives

The primary goal of this project is to develop a comprehensive and ready-to-use solution that empowers organizations, including businesses of all sizes, to eliminate the barriers that hinder organizations from leveraging their data for AI/ML advancements due to challenges in setting up dedicated infrastructure, tools, data, and platforms.

- **Efficient Infrastructure Setup:** Develop an automated solution using Ansible to streamline the installation of AI/ML tools on GPU servers and enable organizations to create on-prem environments by deploying and managing Kubernetes clusters.
- **Enhanced GPU Utilization and User-Friendly Access:** Integrate JupyterHub to provide a user-friendly platform for researchers and developers to utilize GPUs efficiently for their AI/ML tasks.
- **End-to-End ML Lifecycle Management:** Incorporate MLFlow and Kubeflow to enable comprehensive management of ML projects, covering training, deployment, and lifecycle monitoring. Implement Helm charts and operators to simplify the deployment and management of ML components.
- **Responsive Scalability and Data Storage:** Utilize KEDA to enhance AI/ML application responsiveness through event-triggered operations. To effectively manage data types, offer diverse storage options, including Minio, Trino, MongoDB, and Postgres.
- **Insightful Reporting:** Integrate Metabase for generating business intelligence reports, aiding data-driven decision-making and implementing a micro-service architecture for scalable and adaptable AI/ML service deployment.
- **User-Friendly Web Management Portal:** Develop an intuitive web interface for customizable AI/ML configuration to enhance accessibility and ease of use.

These goals collectively aim to empower organizations with a simplified and accessible solution for effectively leveraging AI/ML technologies within their unique contexts.

2.5 Project Scope

The project's primary scope is to develop an integrated solution that simplifies and automates AI/ML infrastructure setup, tailored specifically for SMEs through on-premises HPC. This solution will benefit

a broad spectrum of organizations, from government entities to mobile companies, aiming to optimize AI/ML resource deployment and management. The key components include automated configuration of GPU servers, dynamic grid computing using Kubernetes, and user-friendly GPU accessibility via JupyterHub. Additionally, pre-configured ML lifecycle tools will ensure accurate and efficient output monitoring, facilitating effective MLOps. Integration of event-driven operations using KEDA and various data storage solutions will cater to diverse data requirements, aligning with the MLOps framework. A notable enhancement will be the incorporation of Metabase to bolster business intelligence and reporting capabilities, enabling actionable insights from AI/ML models and data for informed, data-driven decision-making.

In the web component, the project will be adopting a micro-service architecture, utilizing Next JS for the front-end and a combination of Python and Go for the back-end. MongoDB will serve as the primary database solution. Dockers, Redis, Nginx, Github Actions, and ArgoCD will be used to provide smooth integration and deployment. So, effective project management and communication will be facilitated through platforms such as Slack, Notion, and Github. The deployment strategy will encompass both on-premises setups focusing on integrating HPC to enhance performance and security, and cloud-based solutions on the Google Cloud Platform (GCP), providing flexibility and scalability. Through this comprehensive approach our project aims to provide SMEs and various organizations with a robust, scalable, and user-friendly solution for their AI/ML infrastructure needs, promoting efficient MLOps and leveraging the power of on-premises HPC.

2.6 Sustainable Development Goal (SDG)

Our software development project is intrinsically aligned with the ideals of “Industry, Innovation, and Infrastructure,” which underscores the importance of resilient infrastructure, sustainable industrialization, and fostering innovation. So, our project aims to streamline technology infrastructure and promote innovation by making AI and ML accessible to organizations across sectors. In doing so, we lay the foundation for sustainable industrial growth and the development of cutting-edge solutions, directly advancing the objectives of this SDG.

Figure 2.1: Sustainable Development Goal

This figure show the goal of Industry, Innovation, and Infrastructure

2.7 Constraints

The following section identifies some constraints that are contained in our project:

- Budget Restraints: Because SMEs usually have limited budgets and the on premise HPC system's hardware, software, and maintenance expenses should be affordable and cost-effective.
- Scalability: Ensure that the system is designed to adapt to the changing demands of the organization. And the firm should be able to manage additional data, consumers and also processing needs as it grows.
- Customization and Flexibility: Offer a system that can be tailored for the specific needs and use cases of different SMEs such as hospitals, government agencies etc. The system should have a flexiblility to adapt to the varying requirements of different companies and use cases.
- Technical Knowledge: SMEs may lack the technical skills required to administer and maintain an HPC system. So, to guarantee that they can manage and use the system efficiently we have to provide documentation, tutorials, training and also support.
- Security and Compliance: Addressing the security issues while also ensuring data privacy laws and industry specific requirements are followed. This is especially important when we working with private information like medical or governmental data.
- Hardware and Infrastructure Constraints: We have to consider the SMEs available hardware and infrastructure. Creation of the system, including recommendations for improvements if necessary while making effective use of existing resources.
- Ease of Use: Creating a user-friendly and intuitive interface which makes deploying and managing machine learning models simple and also enables non-technical users to use the system easily.
- Resource Management Optimization: System development with efficient resource management (such as GPUs) to decrease downtime and improve use.

2.8 Business Opportunity

The project of on premises HPC system for MLOps gives tremendous economic potential for SMEs by providing them with an accessible and cost-effective solution in their various industries. This platform can attract SMEs looking to improve their operations by expediting access to high-performance computing resources and simplifying machine learning model deployment.

2.9 Stakeholders Description/ User Characteristics

The users of the system will be SMEs i.e., root user, admins, team leads and team members of the organization.

2.9.1 Stakeholders Summary

In this project, various stakeholders play an essential role in managing and utilization of infrastructure. Thus, the main target of this project is SMEs which include the root user, admins, team leads and team members. The root user has administrative rights as he will manage system configurations, grant permissions to other users, and manage the security of HPC infrastructure.

Then, admins will handle system maintenance software-related tasks, ensure smooth operations, and support the team. Team Leads are basically technical leads responsible for overseeing specific teams. So, they have authority over project-related activities, resource allocation, and coordination of tasks. Team leads work closely with administrators to ensure their team's requirements are met within the HPC environment.

Team members use the HPC system to conduct simulations, complete tasks, analyse data, and contribute to the achievement of goals and objectives.

Furthermore, because this is an open-source initiative, developers will be involved as stakeholders.

2.9.2 Key High-Level Goals and Problems of Stakeholders

In our project, the on-premises HPC environment, stakeholders, including the Organization Root User, Admin Users, Team Leads and Team Members, have distinct high-level goals and challenges. The Organization's Root User, who is the sole user of an organization, is primarily focused on ensuring robust security measures to protect the HPC system from potential vulnerabilities and optimizing resource allocation for peak performance. However, they face challenges related to security vulnerabilities and the management of system complexities. However, they still grapple with resource scalability and the integration of evolving technologies into the existing infrastructure. Team Leads, who own one or more teams within an organization, are dedicated to project success and optimal resource allocation for projects. They encounter resource bottlenecks and the need for effective team coordination. Lastly, Team Members, who can be part of one or more teams in an organization are aiming for efficient workflows and thorough data analysis. They must deal with resource availability issues and the ongoing requirement to optimise algorithms for optimal efficiency. Collectively, all these stakeholders contribute to the effective functioning and success of HPC operations while navigating their respective goals and challenges.

Chapter 3 Literature Review / Related Work

This section provides a detailed overview of prior studies or projects related to our FYP. It lists, summarizes, and objectively assesses past research in this domain. The main aim of our study is to elucidate the existing knowledge, theories, and notions about our idea, highlighting their merits and limitations. Subsequent sections delve deeper into the comprehensive review of past research.

3.1 Definitions, Acronyms, and Abbreviations

AI: Artificial intelligence

ML: Machine Learning

SME: Small Medium Enterprise

AWS: Amazon Web Services

GPU: Graphics processing unit

MLOps: Machine Learning Operations

DevOps: Development and Operations

HPC: High Performance Computing

Iot: Internet of Things

AIoT: Artificial Intelligence of Things

CI: Continuous Integration

CD: Continuous Delivery

TFX: TensorFlow Extended

HPCC: High-Performance Computing Cluster

High-performance computing: The use of supercomputers or computer clusters to solve advanced computation problems.

HPC infrastructure: refers to the integrated system of hardware, software, and networking components specifically designed and optimized to support High-Performance Computing tasks.

3.2 Detailed Literature Review

This section will include a detailed literature review of the challenges and advancements in HPC or AI/ML infrastructure setup, management, and deployment. As the organizations recognize the need of AI and machine learning for promoting innovation and productivity, then the requirement for strong and simplified infrastructure becomes critical. However, several barriers ranging from security concerns to

the complexities of tool integration are hindering their seamless adoption. This literature review will delve into the depths of these challenges and also exploring studies with projects that have addressed various areas in the problem. Furthermore, we will highlight the solutions and best practices proposed by researchers and industry experts shedding light on the current state of the art.

3.2.1 AI/ML Development Infrastructure Automation

AI/ML have emerged as transformational powers in this new era of technology, transforming industries and rethinking operating paradigms. However, fully realising the capabilities requires a need for a resilient and agile infrastructure which is capable of adapting to the ever-changing needs of AI/ML workloads. AI/ML Infrastructure Automation aims to streamline this process, eliminating manual bottlenecks and ensuring optimal resource utilization. By automating the setup, deployment and management of AI/ML infrastructures, organizations can focus on innovation and problem-solving rather than getting entangled in the intricacies of infrastructure management. This accelerates the AI/ML development cycle and democratizes access, allowing even entities with limited resources to tap into the transformative power of AI and ML. As AI/ML models become more intricate and data-intensive, a seamless automated infrastructure becomes paramount. Our FYP project proposal dives further into this study topic, imagining a future in which AI/ML infrastructure automation is not a luxury but a need for organisations seeking to remain at the forefront of technological innovation. The following studies shed light on the advancements and challenges in the role of automation in the AI/ML lifecycle.

3.2.1.1 Summary of Studies

The study [3] focuses on integrating AIoT with edge computing to enhance IoT operations and decision-making. The paper introduces an Edge MLOps framework infrastructure designed to automate Machine Learning at the edge, facilitating continuous model training, deployment, delivery, and monitoring. Moreover, there is also an increasing adoption of ”AutoML” to bridge the AI/ML skills gap in software engineering. This is explored in a study that benchmarks various AutoML tools and surveys user perceptions. While AutoML solutions have shown promise in producing superior models, they don’t fully automate every stage of the ML development workflow[4]. Meanwhile, the paper [5] discusses the unique challenges in developing and deploying ML applications compared to traditional software. It underscores the importance of managing ML artifacts and the need of a cloud computing infrastructure for the development and deployment of ML applications. The concept of MLOps, blending DevOps with ML, is central to their discourse. They assess platforms like TFX and Kubeflow, noting performance ambiguities. Furthermore, the paper [6] discusses the use of automation tools provided by Azure DevOps to integrate MLOps and DevOps automated pipelines. It presents a unique DevOps architecture

for developing intelligent Tiny ML dairy agricultural sensors. The paper also discusses the challenges faced in the agricultural field, especially the erratic nature of sensor-generated data and security concerns due to cloud infrastructure.

3.2.1.2 Critical analysis of the studies

Study 1: Edge MLOps: An Automation Framework for AIoT Applications [3] The following paper delves into integrating AIot using edge computing, a topic of significant contemporary relevance. The authors present a comprehensive Edge MLOps framework that seamlessly merges cloud and edge environments, facilitating continuous model training, deployment, delivery, and monitoring. This framework's practicality is underscored by its experimental validation using a real-world application, specifically in forecasting air quality. Furthermore, the paper commendably acknowledges and addresses the challenges associated with traditional ML at the edge, such as manual deployment and security vulnerabilities. It also thoughtfully touches upon enhancing data privacy through methods like Federated Learning. However, while the detailed breakdown of the Edge MLOps framework is a strength, the paper's deep technical dive might pose comprehension challenges for those unfamiliar with the intricacies of AIoT, edge computing, and MLOps. Additionally, a more in-depth comparison with existing solutions and thorough exploration of the framework's scalability and security aspects would have further enriched the paper. Overall, it is a valuable contribution to the field, albeit with areas that could benefit from added depth.

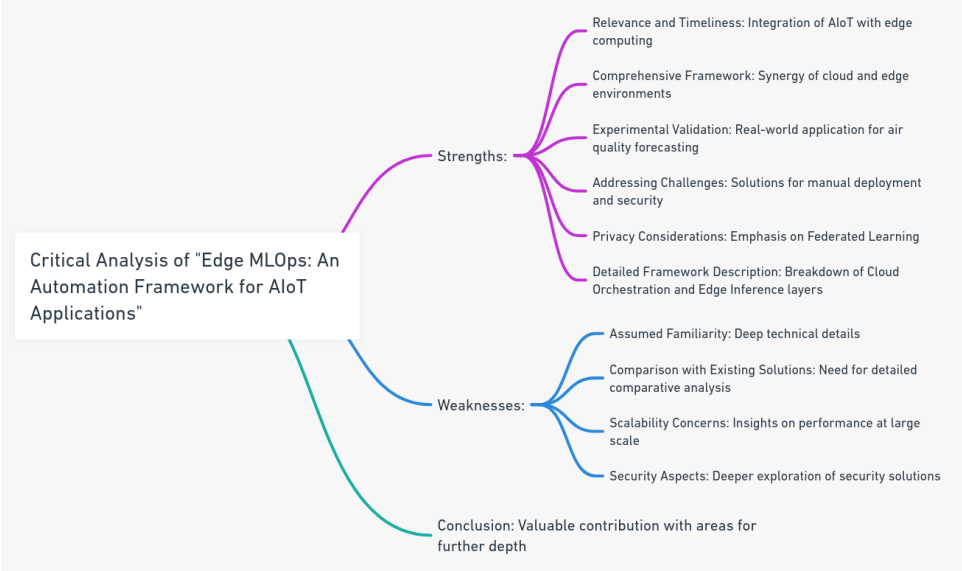

Figure 3.1: Critical Analysis of Edge MLOps

This figure illustrates the strengths & weaknesses of Edge MLOps

Study 2: Assessing the Use of AutoML for Data-Driven Software Engineering[4] This paper delves into the burgeoning adoption of AI and ML in software applications. It underscores the challenges companies grapple with in hiring experts in these domains and positions AutoML as a pivotal solution to bridge this skills gap. Adopting a comprehensive approach, the authors present a dual-faceted study that encompasses a benchmark of 12 AutoML tools and a user-centric survey complemented by follow-up interviews. This methodology offers quantitative data and sheds light on the qualitative aspects of AutoML's use and perception. One of the standout revelations of the study is the capability of AutoML solutions to craft models that potentially surpass those meticulously developed by seasoned researchers, especially in the Software Engineering arena. This insight holds profound implications for the SE community, hinting at AutoML's invaluable utility for specific tasks. Furthermore, the paper bridges a knowledge void by meticulously exploring the depth of AutoML's integration in teams at the forefront of crafting AI/ML-centric systems and by gauging its resonance among industry practitioners. However, the paper's scope seems slightly myopic, concentrating predominantly on the core tasks of the Analysis stage in the ML workflow and offering limited insights into the preparation and dissemination phases. This focus, while valuable, might restrict the study's holistic view of the entire ML process. Additionally, the study's lens on sentiment analysis from technical text as the primary SE task for evaluating AutoML tools raises questions about the generalizability of the findings to a broader spectrum of SE tasks. While the paper offers a deep dive into the perceptions of software engineers vis-à-vis AutoML, a more expansive perspective encapsulating other key stakeholders in the AI/ML development trajectory could have enriched the discourse. In essence, the paper is a significant contribution to the discourse on AutoML in Software Engineering, offering a blend of its potentialities and areas ripe for enhancement.

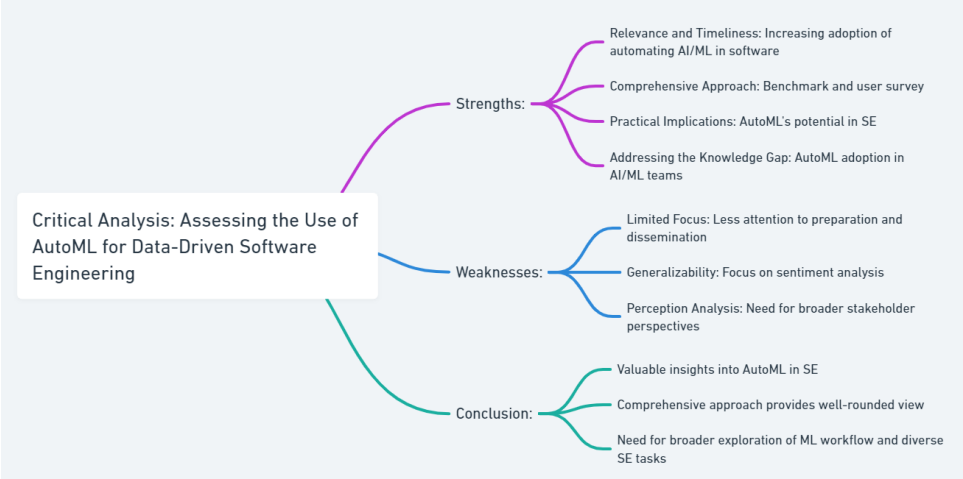

Figure 3.2: Critical Analysis of AutoML

This figure illustrates the strengths & weaknesses of AutoML

Study 3: Towards MLOps: A Case Study of ML Pipeline Platform [5] The following study offers an in-depth exploration of the rapidly evolving domain of MLOps. It underscores the distinct challenges of developing and deploying ML applications, emphasizing the stark differences from traditional software development, especially concerning lifecycle management, feedback mechanisms, and data handling intricacies. One of the paper's notable strengths is its timeliness and relevance, addressing the pressing need for streamlined and reliable ML application production and the requisite infrastructural support. The authors' comprehensive approach is evident in their detailed examination of platforms such as TFX, ModelOps, and Kubeflow, all of which promise end-to-end lifecycle management for ML endeavors. Furthermore, the paper's practical orientation is showcased by constructing a functional ML platform, integrating existing CI/CD tools with Kubeflow. This platform serves as a foundation for running ML pipelines, offering valuable insights into resource consumption metrics, including GPU utilization. However, the paper is not without its limitations. Its scope, while detailed, could benefit from an expanded exploration of other platforms and a comparative analysis with traditional software practices. Though substantial, the depth of discussion on MLOps might be enriched by delving into the challenges and potential integrative solutions with traditional DevOps practices. Additionally, including real-world case studies or industry-specific applications of the platform could have provided a more holistic perspective. In summary, the paper is a significant contribution to the MLOps discourse, blending its complexities with practical solutions, but there remains room for a broader perspective and real-world application insights.

Figure 3.3: Critical Analysis of Towards MLOps Study

This figure illustrates the strengths & weaknesses of Towards MLOps Study

3.2.1.3 Relationship to the proposed idea

The primary objective of our project is to create a holistic infrastructure with dedicated tools, data, and platforms that enable organizations, spanning from small businesses to larger entities, to overcome chal-

lenges in harnessing their AI/ML advancements. In relation to our project, the three studies presented above have the following connections:

- **Study 1[3]:** This study focuses on the integration of AIoT using edge computing, presenting an Edge MLOps framework that merges cloud and edge environments. This study aligns with our project's goal of streamlining AI/ML advancements by emphasizing automation, continuous model training, deployment, and monitoring. The study's practical application in forecasting air quality and its emphasis on data privacy through Federated Learning also resonates with the proposed project's objectives. However, its deep technical focus might pose comprehension challenges for a broader audience, and a more extensive comparison with existing solutions could have been beneficial.
- **Study 2[4]:** This paper delves into the challenges companies face in adopting AI and ML due to the expertise gap and positions AutoML as a solution. This study's comprehensive approach, which includes a benchmark of AutoML tools and user-centric surveys, aligns with our project's aim to simplify and automate AI/ML processes. The revelation that AutoML can produce superior models compared to traditional methods is particularly relevant.
- **Study 3[5]:** This study offers insights into the domain of MLOps, highlighting the challenges in ML application development and deployment. The study's emphasis on platforms like TFX, ModelOps, and Kubeflow and its practical approach to constructing an ML platform aligns with our project's objectives of comprehensive ML lifecycle management and efficient resource utilization. However, its scope could benefit from a broader exploration of platforms and a more in-depth discussion of MLOps challenges.

Collectively, all these studies highlight the growing importance and complexities of AI/ML advancements and the need for solutions for an infrastructure like our project that aims to bridge the gap between AI/ML's technological potential and its practical implementation using automation, as illustrated in Figure 3.4.

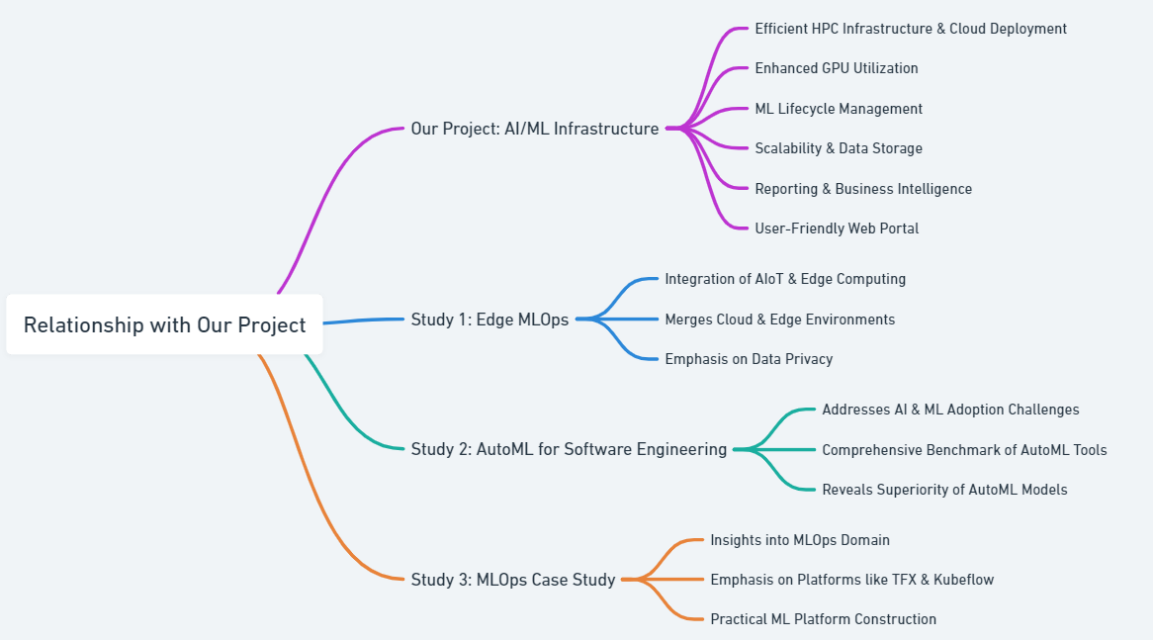

Figure 3.4: Comparative Analysis of Our Project and Related AI/ML Automation Studies

This figure provides a visual representation of the relationship between the primary components of our project idea and the key findings from related automation studies.

3.2.2 High-Performance Computing in AI/ML

3.2.2.1 Summary of the Prior HPC Systems

Singularity is a prominent containerization tool designed explicitly for HPC environments. Unlike other container solutions, Singularity is tailored to meet the unique demands of HPC, offering compatibility, reproducibility, and performance. Containers, in essence, encapsulate an application and its dependencies into a single, portable unit, ensuring that it runs consistently across various computing environments.

In the context of AI and ML, Singularity offers several advantages:

1. **Reproducibility**: Artificial Intelligence and machine learning models need very specific versions of different libraries and other dependencies. Moreover, the singularity makes sure that development and deployment environment remains same that eliminates the ”it works on my machine” problem.
2. **Performance**: The singularity containers can achieve near-native performance which ensures that AI/ML models run properly on HPC clusters.
3. **Compatibility**: The singularity allows AI/ML practitioners to run containers that are built using other container tools which ensures a broad compatibility range.

4. **Security**: The singularity does not need root privileges to run and addressing a security concern in shared HPC environments.

5. **Portability**: AI/ML researchers can run models on their own local machines inside Singularity containers.Then, seamlessly move them to HPC clusters for the purpose of training or inference that will ensure consistent results.

As AI/ML computational requirements are increasing, so singularity is important in making bridge between development and deployment in HPC contexts. A study [7] emphasizes HPC's important role in AI and ML through the use of Singularity, mainly for data-heavy tasks like computer vision, natural language processing and many others. As, Cloud computing is important for the handling of vast datasets, here HPC offers superior computational power, storage, and security. The deployment of AI on HPC shows multiple challenges, especially with the intricate AI environment needs and dependencies on external systems for some particular software packages. Singularity was initially designed for HPC and now it has become the go-to container runtime for such systems. Now it is offering a solution to these challenges and difficulties by ensuring application portability, compatibility, and security.

HPCC Systems is the open-source, data-intensive computing system platform which is designed for the big data analytics landscape. Moreover, it integrates the storage and processing capabilities that allow efficient and scalable data processing. HPCC Systems provides with a conducive environment used for handling huge datasets which is basically an important aspect of machine learning. It has distributed architecture which ensures rapid data processing and model training that helps it as an invaluable tool for AI/ML practitioners. The platform's inherent parallel processing capabilities accelerate the training of complex models, therefore facilitating faster insights and predictions from large-scale data sets.

3.2.2.2 Critical analysis of HPC in AI/ML

In the below section, we will analyze strengths and weaknesses of AI/ML fields of both software defined above.

1. Singularity HPC Singularity HPC is a prominent platform with multiple strengths tailored for AI/ML tasks. The important benefit is containerization. Singularity HPC facilitates containerized applications that ensures AI/ML development occur in the consistent environments. The portability of this platform complements this feature as containers crafted with Singularity can be effortlessly transferred and executed across diverse HPC environments. Moreover the Singularity HPC boasts compatibility, having been designed to align with numerous existing workflows and tools prevalent in the HPC community. By keeping in mind the security point, Singularity containers can operate without necessitating elevated privileges that guarantees a secure milieu for AI/ML endeavors.

However, Singularity HPC is not without its challenges. An important issue lies in complexity mainly for those who are new to containerization. Setting up and administering Singularity containers can be intricate, potentially acting as a deterrent for some users. Another limitation pertains to performance. While containers present many benefits, they might introduce some performance overhead when combined with native installations. From the study [7], we illustrated the architecture diagram Figure3.6, which was used in the paper for Singularity in AI/ML development.

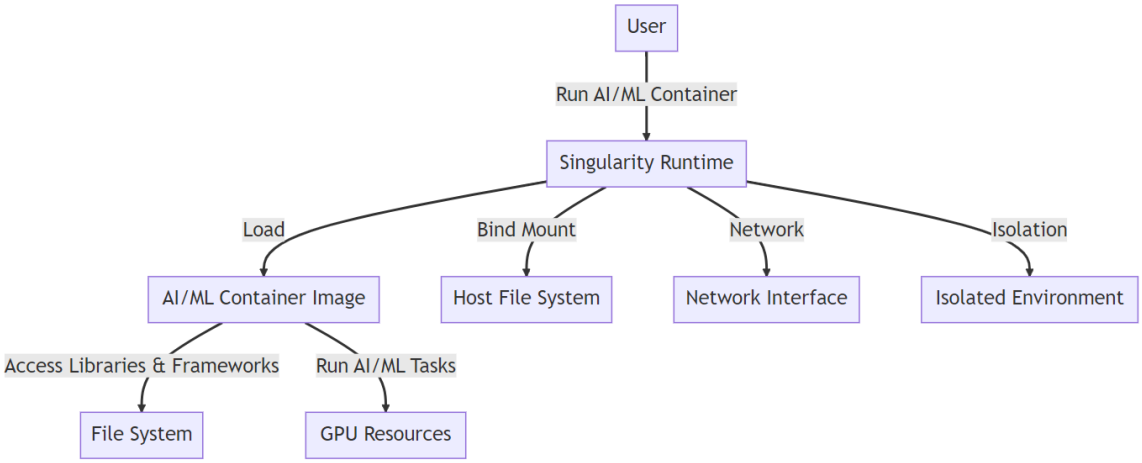

Figure 3.5: Architecture of Singularity Systems

The figure depicts the architecture of Singularity system in the AI/ML Context

2. HPCC Systems HPCC Systems is a platform which has different strengths, making it suitable for AI/ML tasks. Scalibility is one of its primary advantages which is designed to scale out seamlessly. This scalibility enables the processing of extensive datasets and handles the vast amounts of data typically associated with AI/ML tasks. Moreover HPCC Systems offers an Integrated Development Environment (IDE) which is known as ECL IDE, streamlines the writing, debugging, and deploying of ECL code. Additionally. one more significant strength is the refinery capabilities of platform's data, mainly the Thor component. Thor facilitates data cleaning, transformation and linking all vital steps in preparing datasets for AI/ML applications. HPCC Systems is designed for parallel data processing. This is a feature which significantly expedites AI/ML computations. The platform also boasts flexibility, supporting batch and real-time data processing and catering to various AI/ML requirements.

Like any other system, HPCC has also its challenges. One of the weaknesses is its learning curve that is associated with the ECL language which is used for data processing. This language might be unfamiliar terrain for many AI/ML practitioners, that can create a barrier for entrance. One more limitation is the platform's integration capabilities with popular AI/ML libraries. The integration with popular libraries such as TensorFlow or PyTorch may not be easy but could pose challenges for some users. The below figure depicts the architecture diagram for the use of HPCC Systems in AI/ML development.

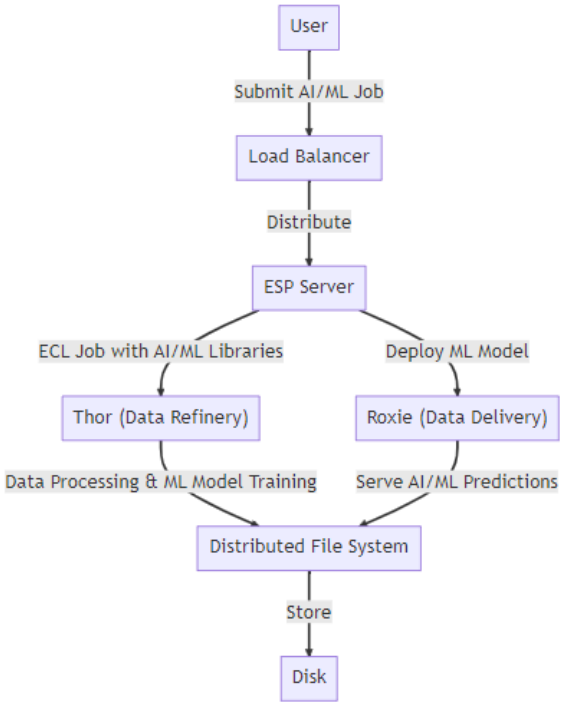

Figure 3.6: Architecture of HPCC Systems

The figure depicts the architecture of HPCC Systems in the AI/ML Context

3.2.2.3 Relationship with the proposed idea

The main idea is to harness the power of on-premises HPC built for SMEs that aims to overcome the multiple complexities of AI/ML. This project seeks a complete solution that integrates different tools and different platforms to streamline AI/ML operations. It covers the complete infrastructure setup to end-to-end ML lifecycle management.

We studied for two systems which are Singularity HPC and HPCC Systems, reveal both parallels and areas of potential research expansion.

Singularity HPC is highlighted for its containerization capabilities, which ensure consistent AI/ML development environments. The system's strengths, such as reproducibility, performance, compatibility, security, and portability, align with our project's vision of providing a consistent and secure environment for AI/ML tasks. However, the challenges of Singularity HPC, like its complexity and potential performance overhead, resonate with the barriers our project aims to address.

Conversely, HPCC Systems [8] is recognized as an open-source platform primarily crafted for big data analytics. Its notable strengths, such as scalability, integrated development environment, data refinery capabilities, parallel processing, and flexibility, align with your project's goal of handling vast datasets efficiently. The platform's ability to process data in parallel can accelerate AI/ML computations, which aligns with your vision of speeding up AI/ML model development and deployment. However, the chal-

lenges associated with HPCC Systems, like the learning curve of the ECL language and potential integration issues with popular AI/ML libraries and applications, are areas our project will address.

In summary, both Singularity HPC and HPCC Systems offer capabilities that align with our project's vision of simplifying and automating AI/ML infrastructure setup for SMEs. The strengths and challenges associated with these systems offer valuable insights into the need to provide an HPC Infrastructure with MLOps, which can guide the development and refinement of our project's solution.

3.2.3 Cloud vs On-Premises AI/ML HPC Infrastructure

3.2.3.1 Summary of Available Cloud and On-Premises Infrastructures for AI/ML Development

Cloud vs. On-Premises AI/ML HPC Infrastructure presents a pivotal decision for organizations in the modern digital landscape. On the cloud front, platforms like Amazon SageMaker from AWS and Azure Machine Learning from Microsoft Azure offer robust machine learning capabilities, allowing users to build, train, and deploy models with ease. These platforms provide scalability and flexibility. While on-premises solutions like NVIDIA DGX Systems and HPE Apollo Systems offer us dedicated high-performance computing infrastructure that are tailored for AI and ML workloads. Such systems provide control, security, and data protection. The study by Golec [9] underscores this distinction, emphasizing that on-premises hardware and applications are managed on-site without third-party involvement, granting organizations full control over their data centers. This becomes especially crucial for healthcare organizations and government bodies that handle sensitive data. The paper further notes that while cloud solutions offer flexibility and adaptability, many organizations remain skeptical about cloud security, making on-premises infrastructure a preferred choice for those prioritizing data protection and compliance.

3.2.3.2 Amazon SageMaker

Amazon SageMaker, as highlighted in the paper [10], is an end-to-end machine learning platform offered by Amazon Web Services. Amazon Sagemaker is especially designed to streamline the process of building, training and deploying AI/ML models. SageMaker being cloud-centric offers tools and services which simplify the machine learning lifecycle(data labeling to deployment and inference).

Strengths:

1. *End-to-End AI/ML Platform*: SageMaker is a complete suite of tools which cover the complete machine learning lifecycle covering from data labeling to model training and deployment.
2. *Performance*: This platform is used for achieving high throughput, ensuring efficient neural net-

work training and inference.

3. *Flexibility*: SageMaker supports both GPU acceleration and auto-scaling that makes it adaptable to various workloads.

Weaknesses:

1. *Cost*: SageMaker comes at a higher price point which could be a barrier for SMEs or those organizations with has limited budgets.
2. *Complexity*: SageMaker might arise challenges for the users who are unfamiliar with AWS or those without the technical expertise to navigate its offerings.
3. *Specific Use Case*: The platform is tailored for deep learning. This point might limit the usability and applicability for broader or non-ML tasks.

Amazon SageMaker is well known for its capability to provide low latency as compared to other services like AWS Lambda. However, the pay-per-use model of serverless platforms like Lambda can result in cost savings, making them more suitable for burst traffic. The paper also highlights the trade-offs between performance and cost when using cloud AI/ML infrastructure, with SageMaker being a prominent player in this space due to its comprehensive offerings and performance capabilities.

3.2.3.3 Azure Machine Learning

Azure Machine Learning by Microsoft is a cloud-based service that facilitates the entire machine learning lifecycle. It provides tools and services that streamline the process of building, training, and deploying AI/ML models. While primarily a cloud-centric platform, Azure ML also offers on-premises solutions through Azure Stack, catering to organizations requiring on-site HPC infrastructure. This flexibility allows businesses to harness the power of AI/ML while maintaining control over their data and infrastructure, making it easier to develop and deploy machine learning models in a secure and efficient manner[11].

Strengths:

1. *Comprehensive Platform*: Azure ML provides us with end-to-end platform for the building, training, and deployment of machine learning models.
2. *Integration with Other Azure Services*: Azure ML integrates with other Azure services(Azure Data Factory, Azure Databricks,Azure Kubernetes Service), which allows for a more cohesive and streamlined workflow.
3. *Support for Open Source Frameworks*: Azure ML supports many open-source machine learning

frameworks such as TensorFlow, PyTorch, and scikit-learn which ensures the flexibility. This caters to a broad audience of developers.

4. *Automated Machine Learning*: Azure ML has a feature of AutoML which automatically selects the best algorithm and hyperparameters for a given dataset. This reduces the time and effort required for model selection and tuning.

Weaknesses:

1. *Complexity*: Azure ML can be overwhelming for the starters because of its vast array of features and services. Moreover, the steep learning curve might be required for those new to the Azure ecosystem.
2. *Cost*: Azure ML operates on a pay-as-you-go model. While this can be cost-effective for sporadic usage, it can become expensive for large-scale, continuous operations. Organizations need to monitor their usage closely to avoid unexpected costs.
3. *Dependency on Azure Ecosystem*: Organizations that are heavily invested in other cloud platforms might find it challenging to migrate or integrate their existing workflows with Azure ML.
4. *Limited On-Premises Support*: Azure ML is primarily a cloud-based service. While there is support for on-premises scenarios through Azure Stack, it might not cater to all the requirements of organizations that prefer or need to keep their operations entirely on-premises, as identified on page 10 in [11].

3.2.3.4 NVIDIA DGX Systems

The NVIDIA DGX-1 System, architecture provided white paper[12], is a specialized on-premises HPC system for deep learning and AI/ML. It integrates eight Tesla P100 GPUs in a hybrid NVLink topology, supplemented by Intel Xeon CPUs and InfiniBand network cards, aiming for top-tier deep learning performance.

Strengths:

1. *High-Performance Architecture*: Designed for high throughput, it features a unique NVLink topology with eight Tesla P100 GPUs.
2. *Deep Learning Focus*: Tailored specifically for deep learning, it offers a scalable platform for both research and production.
3. *Advanced Technologies*: Incorporates the Tesla P100 with HBM2 memory, ensuring high bandwidth and compact server design.

Weaknesses:

1. *Complexity*: Its advanced design might be challenging for those without deep technical knowledge.
2. *Cost*: Its high-end features are premium, potentially deterring smaller entities.
3. *Niche Focus*: Primarily optimized for deep learning, which might limit its broader applicability.

Incorporating insights from the paper, the NVIDIA DGX-1 System is undeniably a powerhouse in deep learning. Its architecture, combined with the latest technologies, positions it as a leader in the field. However, its intricate design and cost could restrict its use to well-resourced organizations.

3.2.3.5 HPE Apollo Systems

The HPE Apollo Systems are a family of high-density server solutions designed by Hewlett Packard Enterprise to deliver exceptional performance, scalability, and efficiency for HPC and AI/ML workloads. These systems are optimized for specific tasks and are known for their ability to handle large-scale data analytics and complex computational tasks, making them ideal for research and enterprise environments. According to [13], the HPE Apollo Systems play a significant role in the on-premises HPC landscape, particularly for AI/ML development.

Strengths:

1. *Diverse Hardware Resources*: The Apollo Systems, as part of the Bridges-2 platform, offer a range of computing nodes, from Regular Memory and Large Memory nodes to Extreme Memory nodes and GPU nodes, catering to various computational needs in AI/ML training and development.
2. *High-Performance Storage*: The hierarchical HPE ClusterStor filesystem consists of fast SSD, capacity, and archival storage. This ensures the rapid and reliable data access which is crucial for AI/ML applications.
3. *Flexible Service Nodes*: Apollo Systems provides 32 Service Nodes supports complex research platforms and multiple management tasks.

Weaknesses:

1. *Complexity*: Multiple ranges of nodes and configurations might cause difficulty for users or organizations without the expertise to navigate, handle and optimize their usage.
2. *Potential Overhead*: Management and maintenance of such a diverse system might create system administration and monitoring overhead.
3. *Niche Specialization*: While the Apollo Systems are optimized for specific tasks like AI/ML, their

specialization might limit their broader applicability in other computational domains.

In short, the HPE Apollo Systems, as part of the Bridges-2 platform, stands out as a robust on-premises solution for AI/ML development. Their diverse hardware resources, advanced storage solutions, and flexible service nodes make them a valuable asset. However, their complexity and potential administrative overhead might challenge certain users or organizations.

3.2.3.6 Relationship to the proposed idea

Our proposed project's idea is to establish a holistic and user-friendly framework for organizations, enabling them to harness the potential of AI/ML by addressing challenges in infrastructure, tools, data, and platforms. Central to this initiative is the emphasis on HPC for developing and managing AI/ML models. Key tenets of the project include automation, tailored solutions, efficient data storage mechanisms, user-centric accessibility, MLOps, and scalability. Additionally, the project underscores the security implications of cloud-based solutions, advocating for the primacy of on-premises HPC solutions.

In correlation with existing software solutions, notable parallels and dissimilarities are emerging. Amazon SageMaker, for instance, parallels the project's comprehensive ML management aspirations but is predominantly cloud-centric and potentially cost-prohibitive for SMEs. Azure Machine Learning, while offering a similar end-to-end platform and on-premises solutions, is deeply entrenched in the Azure ecosystem, which could pose integration challenges. NVIDIA DGX Systems, tailored for deep learning, align with the project's high-performance on-premises orientation but may be restrictive in cost and scope for broader AI/ML applications. Lastly, HPE Apollo Systems resonate with the project's HPC emphasis, but their specialized nature might introduce complexities for certain users.

3.3 Software Tools Summary Table

The following table, provides a concise overview of the key findings and insights gleaned from the tools and software provided in the sections above.

Table 3.1: Software Tools Summary Table

A summarization of software tools identified in the related work, focusing on their features, relevance to the application, and potential limitations.

Application	Features	Relevance to Application	Limitations
Edge MLOps framework [3]	Automation for AIoT applications, Edge delivery using containers, Automated ML monitoring	Edge MLOps for SMEs, On-Premises HPC, Focus on AI/ML infrastructure	Manual deployment, HPC security challenges
AutoML [4]	Solution to AI/ML skills gap, Automates end-to-end AI/ML pipelines	Democratizes AI/ML for SMEs, Automates AI/ML pipelines	Not fully automated ML Flow
TFX, ModelOps, and Kubeflow [5]	End-to-End Lifecycle management of ML Applications, Pipeline configuration/SDK for ML workflow	Vision of democratizing AI/ML for SMEs, Lifecycle of ML Apps	No explicit listing of integrations, Challenges in data management
Singularity HPC [7]	GPU Support for AI/ML Development, Resource Limit Control, Container Scheduling	AI/ ML development on HPC, Usage of HPC Clusters to enhance performance	Extensive User Knowledge Required, HPC is not Ready and configured for AI/ML Development
HPCC Systems [8]	Parallelized creation & training of ML models, Large set of evaluation metrics, High-performance processing for big data	Aligning with AI/ML deployment for SMEs, Easing integration with on-premises HPC	Lack evaluation methods to monitor
Amazon SageMaker [10]	Provides GPU acceleration, Offers auto-scaling capabilities, End-to-end ML platform	Idea for on-premises architectures to balance workloads	Instances can be underutilized, leading to higher costs
Azure Machine Learning [11]	Cloud-based ML as a service, Built-in regression modules	Drag and drop canvas interface, Integrated development environment	Limited metrics in the Evaluation, Assumes data preparation is complete

Application	Features	Relevance to Application	Limitations
NVIDIA DGX Systems [12]	AI supercomputer for deep learning training, System software & libraries tuned for scaling deep learning	On-premises HPC Infra, Seamless integration with AI/ML frameworks	Power consumption, Not Cost-effective Solution
HPE Apollo Systems [13]	Hierarchical HPE ClusterStor filesystem, Unified data access	On-premises HPC Infra, Supports AI/ML data analytics	Complex Configuration. Costly

3.4 Conclusion

Through a thorough analysis above, we identify several choices available in choosing the AI/ML development infrastructure, each with its unique strengths and challenges. The chapter comprehensively reviewed several essential tools and platforms, including Singularity HPC, HPCC Systems, Amazon SageMaker, Azure Machine Learning, NVIDIA DGX Systems, and HPE Apollo Systems. Each system offers distinct capabilities tailored to specific AI/ML needs, ranging from containerization and parallel data processing to cloud-based and on-premises solutions.

Singularity HPC and HPCC Systems, in particular, highlight the significance of on-premises HPC solutions, emphasizing the importance of consistent, secure, and scalable environments for AI/ML tasks. Meanwhile, platforms like Amazon SageMaker and Azure Machine Learning, being powerful and comprehensive cloud solutions, bring forth considerations of cost, complexity, and ecosystem dependencies. On the other hand, specialized on-premises solutions like NVIDIA DGX Systems and HPE Apollo Systems underscore the significance of high-performance, tailored infrastructure for AI/ML despite the potential challenges in terms of cost, scope, and complexity.

Our proposed project basically aims to create a user-friendly HPC framework that will address the multiple challenges and difficulties of AI/ML infrastructure. From the insights of reviewed systems, it is shown that a balanced approach that emphasize automation, tailor solutions, efficient data storage, user-centricity, MLOps, scalability, and security, is paramount. By the evolvement of AI/ML field, we can take multiple lessons from the systems which can guide us in the development and refinement of our project, ensuring it remains responsive to the ever-changing needs of the AI/ML demands.

Chapter 4 Software Requirement Specifications

This chapter covers the software and hardware requirements, quality attributes, and functional/non-functional requirements of the system. The graphical user interface, use cases that the system will address, and database design are also discussed.

4.1 List of Features

The following are the some important features of the system:

1. Scalable Infrastructure
2. Customization
3. Resource Allocation
4. Version Control
5. Data Source Integration
6. Workflow Automation

4.2 Functional Requirements

Below are mentioned the functional requirements of the system:

4.2.1 User Account Management

- **Root User Registration:** The system shall allow root user to enter personal and organizations information for account creation.
- **Login:** The system shall allow the proper login to users using their credentials.
- **Logout:** The system shall allow the log out by users from their accounts.
- **Edit Profile:** The system shall allow users to edit their profile successfully.

4.2.2 User Permission & Roles

- **Root User:** The system shall allow the root user to have complete access to all the related features and functions.
- **Admin User:** The system shall allow multiple admin users for each organization. Admin users shall have the following functional requirements:

  - Create, edit and delete user accounts within their functions of their organization.
  - Assign and revoke roles and permissions to different team leads and different team members within their organization.
  - Access and update different organization-level settings and other configurations.
  - View and manage the teams and other team-related data within their organization.

- **Team Leads:** The system shall have different team lead of teams in organization. Team leads shall have the following functional requirements:
  - Create and manage multiple teams within their organization.
  - Add, update, remove team members from their teams.

- **Team Members:**The system shall allow multiple team members within a team. Team members shall have the following functional requirements:
  - Team member shall view projects that are within their teams.

4.2.3 Cluster Management

- **Cluster Deployment:** The system shall allow the users to deploy new clusters of their organization.
- **Cluster Configuration:** The system shall allow users to configure clusters for resource allocation of their organization.
- **Cluster Resource Allocation:** The system shall allow users to allocate hardware resources like GPUs and software packages to clusters of their organization.
- **Real-Time Monitoring:** The system shall provide real-time resource utilization monitoring for clusters.

4.2.4 Project Management

- **Project Creation:** The system shall allow users to create AI/ML projects by using different tools and configurations.
- **Project Configuration:** Users shall be able to configure the project according to their needs.
- **Project Tools Configuration:** The system shall allow users to drag and drop tools in the project space.
- **Project Deployment:**The system shall allow users to deploy projects in their allocated clusters.

4.2.5 System Logs

- **Log Viewing:** The system shall allow users to view different types of project logs such as error logs.
- **Categorize Logs:** The system shall users to categorize logs such as project activity logs, error logs, or system event logs.
- **Log Export:**Users shall be able to export log data in various formats, such as CSV or JSON.

4.2.6 Other Requirements

- **Versioning:** The system shall allow user to track different versions of the project.
- **Notification:** The system shall be able to generate different notifications based on the resource and project monitoring.
- **Tutorials of Deployment:** The system shall provide tutorials and guides on deployment best practices, application setup, and resource optimization.

4.3 Quality Attributes

The quality attributes of the system defines the measures on which the performance of software is judged. It is a checklist of the things that is needed which is going to help in assessing the quality of software. The following are the essential quality attributes of the system:

4.3.1 Reliability

- The system shall give correct information consistently and in any working environment.

4.3.2 Maintainability

- The system shall be easy to maintain
  - It shall be easy to improve the system by adding more code.
  - It shall be easy to integrate new features and functionalities into the system in the future.

4.3.3 Integrity

- The system shall provide only authorized access to the critical features.

4.3.4 Interoperability

- The system shall be designed to work on multiple operating systems and different hardware

4.3.5 Security

- The system shall be secure and protect sensitive user data from unauthorized access.

4.3.6 Usability

- The system shall have an easy and user-friendly interface, which should be easy to navigate.

4.4 Non-Functional Requirements

This section contains the non-functional requirements to measure the system's quality, which includes terms such as time of availability, reliability, usability, serviceability, performance, and security of the system.

4.4.1 Availability

- The system shall ensure measures for maximum possible uptime. The goal is to achieve and maintain at least 95% uptime.

4.4.2 Reliability

- The system shall be designed with fault tolerance mechanisms to minimize service disruptions in case of hardware or software failures.
- The system shall continuously monitor and respond to system anomalies, errors, or performance degradation. System admins should be notified within 5 minutes of any critical issue.

4.4.3 Usability

- The system's user interface should be designed such that users can start using the application within 5-10 minutes, and it should be self-explanatory.
- The average time it takes for a user to complete common tasks within the application should not exceed a specific time limit. For example, ”95% of users can complete a common task within 5 minutes.”
- The system should aim for a low error rate in user interactions. For instance, ”The error rate for user interactions should not exceed 2%.”

4.4.4 Security Requirements

- Each user of the system must be uniquely identifiable through their email address.
- Each user shall have a distinct password to ensure secure access to their individual accounts.
- Duplicate credentials for multiple users shall be strictly prohibited.

4.4.5 Performance requirements

- The system shall be able to serve up to 100 organization users at a time.
- The system shall have low response times ($\leq$ 10 seconds) in sending queries to the database.
- The system shall deploy the clusters within a maximum time of 10 minutes whenever new clusters are requested.
- The system shall have an error rate of near 0%.

4.5 Assumptions

The following aspects have been assumed for the specification of the system:

- All Users of the system have access to web browsers and devices.
- User has the availability of suitable hardware to deploy the infrastructure.
- Users will provide the correct and appropriate configurations of the hardware.
- User must have a suitable connection to access the online web portal.

4.6 Use Cases

This section outlines the use cases that showcase crucial core features of the final system:

4.6.1 Deploy Cluster

Table 4.1: Deploy Cluster

Allows System Admins to create a new on-premises cluster with user-friendly templates and validation

Name	Deploy a new cluster for AI/ML project		
Actors	System Admin		
Summary	This use case allows user to create a cluster on the on-premises environment.		
Pre-Conditions	The user must be logged in. Required resources must exist The user must have the necessary permissions		
Post-Conditions	A cluster with the required information is created		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user selects “Cluster Management”	2	The system will display a list of all Cluster with a button “Add New Cluster” on the top
3	The user clicks on the “Add new Cluster”	4	The system provides templates and prompts the user to select their cluster.
5	The user chooses the cluster and clicks deploy cluster	6	The system prompts the user to provide the necessary information in order to deploy the cluster
7	The user specifies the cluster details and clicks “Deploy cluster”	8	The system validates the provided cluster details and deploys it
Alternative Flow			
5-A	The user added some incompatible cluster configuration details.	6-A	The system will show an error message “Compatibility issues with selected resource Resource Name”

4.6.2 Remove Cluster Resources

Table 4.2: Remove Cluster Resources

Allows System Admins to remove cluster resources

Name	Remove Cluster Resources		
Actors	System Admin		
Summary	This use case descibes that user can remove resources from cluster and customize resources withing the cluster.		
Pre-Conditions	• The user must be logged in. • The user must have the necessary permissions.		
Post-Conditions	The cluster's resources are successfully customized.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The users selects "Cluster Management"	2	The system will display a list of all Cluster with a button "Add New Cluster" on the top
3	The user selects the cluster which he wants to customize	4	The system provides details of the selected cluster
5	The user selects to "customize cluster" the resources.	6	The system shows options for adding or removing resources
7	The user will click on the Remove button in the front of that resource	8	The system will verify that the resource is attached to any project. The system will show a confirmation dialogue
9	The user will click on Yes	10	The system will remove the resource from that cluster
Alternative Flow			
7-A	The user selects a resource that is assigned to any project within the cluster	8-A	The system will show an error message "Cannot remove this resource"
9-A	The user will click on No	10-A	The system will not remove the resource from the cluster

4.6.3 Add a New Project

Table 4.3: Add a New Project

Enables users to add a new project with given application types

Name	Add a New Project		
Actors	Team Leads, Admin		
Summary	The user can create a new application of the selected type.		
Pre-Conditions	The user must be logged in and have permission to create a new application.		
Post-Conditions	A new project is successfully created with the selected application type, and cluster resources are allocated for the project.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user clicks on the New Project button on the top of application	2	The system will display a list of multiple types of selected applications available
3	The user will select a specific application type from the list	4	The system will provide a template for that application
5	The user will enter the application name, description, and configuration settings in the input fields	6	The system will verify the information and ask for the cluster where the user wants to create the project
7	The user will select a cluster	8	The system shows a confirmation box
9	The user will select yes	10	A new project will be created within that cluster
Alternative Flow			
5-A	The user will leave some fields empty	6-A	The system will display an error message “Please fill all the fields”
7-A	The user will click on No	8-A	The system will redirect to the dashboard without creating any application

4.6.4 Drag and Drop Tools to the Projects

Table 4.4: Drag and Drop Tools to the Projects

Enables Team Members to manage tools by dragging and dropping them from a tool management section

Name	Drag and Drop Tools to the Projects		
Actors	Team Member		
Summary	This use case enables users to drag and drop tools to the projects from the centralized tool management section.		
Pre-Conditions	• The actor must be logged in and have access to the dashboard. • The actor must have access to that project.		
Post-Conditions	The selected tools are successfully added to the specified project.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user selects a project from the list of available projects on the dashboard	2	The system will display a list of all projects
3	The user clicks on the “Tools Management” section	4	The system presents the list of available tools that can be added to the project
5	The user drags the selected tool icon to the project’s workspace	6	The tool is successfully added to the project’s workspace

4.6.5 Login

Table 4.5: Login

Allows registered users to access accounts with validation and error handling

Name	User Login		
Actors	User		
Summary	This use case allows a registered user to log in to the system using their credentials.		
Pre-Conditions	User must be registered in the system. The system is operational.		
Post-Conditions	The user is successfully logged in and gains access to their account’s features and data.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user enters their username and password and clicks the ”Login” button.	2	The system validates the user’s credentials and grants access if they are correct.
Alternative Flow			
1	The user enters incorrect credentials.	2-A	The system displays an error message, indicating that the login failed. The user can retry.

4.6.6 Customization of Project Configuration with GPU Resources and Software Packages

Table 4.6: Customization of Project Configuration with GPU Resources and Software Packages

Allows team leads of organization to update configurations according to needs by selecting GPU resources and other packages.

Name	Customize Project Configuration with GPU Resources		
Actors	Team Lead		
Summary	This use case enables project administrators or users to modify the configurations of project according to their AI/ML needs by selecting GPU resources and other software packages.		
Pre-Conditions	• The user must be logged in. • GPU resources must be available and recognized by the system. • Software that is compatible with the project must also exist.		
Post-Conditions	The project's configuration is successfully customized with selected GPU resources and software packages.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user navigates to the project dashboard	2	The system will display a list of all projects
3	The user selects the project from the list	4	The system will display the dashboard of that project
5	The user clicks on the "Projects Configuration" section	6	The system displays project configuration options, such as GPU and software package selection
7	The user will click on "Edit project configuration"	8	The system presents a list of available GPU resources and software resources for selection
9	The user selects the required software packages and GPUs for the project	10	The selected GPUs and Software packages are added to the project's configuration
Alternative Flow			
8	The user attempts to select an incompatible or unsupported resource	9-A	The system will show an error message "Compatibility issues with selected resource ResourceName"

4.6.7 Register as the Root User for an Organization

Table 4.7: Register as the Root User for an Organization

Enables the Organization's Root User to successfully register, providing personal and company information, with validation and confirmation steps

Name	Register as the Root User for an Organization		
Actors	Organization Root User		
Summary	This use case enables the root user to register.		
Pre-Conditions	• The user must be on the registration page of the website.		
Post-Conditions	The organization is successfully registered on the website, and the Root User can now access the organization's account using the provided credentials.		
Special Requirements	Each organization will have only one root user.		
Basic Flow			
Actor Action		**System Response**	
1	The user enters the required personal information such as name, email address, password, and any additional details.	2	The system validates the entered information, ensuring all required fields are filled and the email address is in a valid format.
3	The user enters the required company information such as name, type of organization, size, geography, and any additional details.	4	The system validates the entered information, ensuring all required fields are filled and this root user for this organization doesn't exist before.
5	The user reviews all the provided information.	6	The website presents a summary of the entered information for the Root User to review and confirm.
7	The Root User confirms the accuracy of the information.	8	The website processes the registration request and creates an organization account
Alternative Flow			
1	The user enters incomplete or wrong personal details.	2-A	Appropriate error messages are displayed to the user, prompting them to correct the provided information.
3	The user enters the details of the organization that already exists in the record.	4-A	Appropriate error messages are displayed to the user, prompting them to correct the provided information.

4.6.8 Add Admin to the Organization

Table 4.8: Add Admin to the Organization

Allows the Organization's Root User to add new admins, providing necessary details, with validation and confirmation steps

Name	Add Admin to the Organization		
Actors	Organization Root User		
Summary	The root user can add admins to the organization.		
Pre-Conditions	• The Root User is logged into the organization's account on the website. • The organization has an existing registration.		
Post-Conditions	The new admin is successfully added to the organization, and their details are reflected in the organization information.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The Root User navigates to the admin management section within the organization's account	2	The website displays the admin management interface, showing a list of current admins and an option to add a new admin.
3	The Root User selects the option to add a new admin and fills in the required details for the new admin, including name, email, and position.	4	The website validates the entered details for correctness and completeness.
5	The Root User confirms the addition of the new admin.	6	The website processes the request, adds the new admin to the organization
Alternative Flow			
3	The Root User selects the option to add a new admin and fills in the required information either wrong or incomplete.	4-A	Appropriate error messages are displayed to the user, prompting them to correct the provided information.
5	The Root User enters the admin that is already in the list of admins.	6-A	Error message is shown that admin already exists.

4.6.9 Remove Admin from the Organization

Table 4.9: Remove Admin to the Organization

Allows the Organization's Root User to remove admins from the organization

Name	Remove Admin from the Organization		
Actors	Organization Root User		
Summary	The root user can remove admins from the organization.		
Pre-Conditions	• The Root User is logged into the organization's account on the website.		
Post-Conditions	The existing admin is successfully removed from the organization		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The Root User navigates to the admin management section within the organization's account	2	The website displays the admin management interface, showing a list of current admins and an option to remove an existing admin.
3	The Root User selects an existing admin to remove from the organization	4	The website asks for the deletion of admin.
5	The Root User confirms the removal of the existing admin.	6	The website removes the existing admin from the organization

4.6.10 Removing User from Tool's Queue

Table 4.10: Removing User from Tool's Queue

Describes the process of an Admin removing a user from the queue of a specific tool within the infrastructure

Name	Removing User from Tool's Queue		
Actors	Admin, System		
Summary	**Summary:** This use case describes the process of an Admin removing a user from the queue		
Pre-Conditions	**Pre-Conditions:** • The Admin has access to the infrastructure system.		
Post-Conditions	**Post-Conditions:** The specified user is successfully removed from the queue of the selected tool.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The Admin selects the desired tool.	2	The System opens the tool's details.
3	The Admin selects the user from the tool's queue.	4	The System marks the user for removal.
5	The Admin clicks on "Remove from Queue".	6	The System removes the user from the tool's queue.

4.6.11 Add User to Organization

Table 4.11: Add User to Organization

Allows Root User or Admin to add new user to organization, with validation and confirmation steps

Name	Add User to Organization
Actors	Root User, Admin
Summary	A Root User or Admin can add a new user to their organization
Pre-Conditions	• The organization exists. • The Actor is logged into the organization's account on the website. • The Actor has root or admin level access.
Post-Conditions	The new user account is successfully added to the organization, and their details are reflected in the user management interface.
Special Requirements	None

Basic Flow			
Actor Action		**System Response**	
1	The Actor navigates to the user management page within the organization's account.	2	The website displays the list of users.
3	The Actor clicks the add button.	4	The website opens a modal to add a new user details.
5	The Actor enters the necessary details such as name, email, and user type.	6	The website validates the provided information and prompts for confirmation.
7	The Admin confirms the addition of the new user.	8	The website processes the request, adds the new user to the organization, and updates the user list. The website displays a confirmation message indicating successful addition of the new user.
Alternative Flow			
3	The Actor fills incorrect or incomplete details	4-A	The system shows an error message to fill correct requirements.

4.6.12 Remove User from the Organization

Table 4.12: Remove User from the Organization

Allows Root User or Admin to remove user from the organization, with confirmation steps

Name	Remove User from Organization		
Actors	Root User, Admin		
Summary	A Root User or Admin can remove a user from their organization		
Pre-Conditions	.The Actor has root or admin level access.		
Post-Conditions	The user account is successfully removed from the organization, and their details are reflected in the user management interface.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The Actor navigates to the user management page within the organization's account.	2	The website displays the list of users.
3	The Actor clicks the edit button on the desired user.	4	The website opens a modal which has desired user details.
5	The Actor clicks the remove button.	6	The website processes the request, removes the new user from the organization

4.6.13 Manage Versions

Table 4.13: Manage Versions

Allows Team Leads and Admins to manage project versions, with options to select and restore specific versions

Name	Manage Versions		
Actors	Team Leads/Admin		
Summary	A user is able to manage the versions of a project.		
Pre-Conditions	The User is logged into the organization's account on the system.		
Post-Conditions	The user is successfully able to manage the versions of a project.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The User navigates to the project dashboard of the system.	2	The system shows the project dashboard, and the version control option is at the top.
3	The user selects the versions option.	4	The system displays all the versions of that project.
5	The user selects one of the versions from the versions list.	6	The system asks for the confirmation of changing versions.
7	The user confirms the change.	8	The system restores the version successfully.
Alternative Flow			
7	The user cancels the version change.	8-A	System returns to the versions list and changes nothing.

4.6.14 Monitor Resource

Table 4.14: Monitor Resource

Enables Team Leads and Admins to view and monitor hardware resource consumption for selected projects in both tabular and graphical formats

Name	Monitor Resource
Actors	Team Leads, Admin
Summary	A user is able to view the resources and their performances.
Pre-Conditions	User is logged in to the system and has access to organization clusters.
Post-Conditions	The User can see and monitor the hardware resource consumption for the selected project in both tabular and graphical formats.
Special Requirements	None

Basic Flow			
Actor Action		**System Response**	
1	The User navigates to the cluster management section within the organization's account.	2	The system displays the cluster management interface and shows multiple clusters.
3	User selects one of the clusters from the cluster list to view its resources.	4	The system displays options for managing the cluster and viewing its resource consumption.
5	The User selects the option to view hardware resource consumption.	6	The system navigates to a new page that shows the resource consumption of that cluster along with the multiple projects and each project's resource consumption.
7	The User chooses to view the hardware resource consumption in a graphical format.	8	The system opens a graphical representation (e.g., line chart, bar chart) displaying the hardware resource consumption over time for the selected project. Time intervals (e.g., hourly, daily) may be selected by the User.

4.6.15 View Tutorials

Table 4.15: View Tutorials

Allows Team Leads, Admins, and Team Members to view tutorials, with step-by-step explanations for various tasks

Name	View Tutorials		
Actors	Team Leads, Admin, Team Members		
Summary	A user is able to view the tutorials.		
Pre-Conditions	User is logged in to the system.		
Post-Conditions	The user is able to view tutorial.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The User navigates to the main dashboard of the system.	2	The system displays the main dashboard having the tutorial button on the navigation bar.
3	User selects the view tutorial button from the dashboard.	4	The system displays all tutorials such as create project/ assign resource to project.
5	The User selects one of the tutorials from the list.	6	The system plays the tutorial that explains in a step-by-step manner for the user.

4.6.16 View and Export Project Logs

Table 4.16: View and Export Project Logs

Allows Team Leads, Team Members, and Admins to view and export project logs, with options for selecting log types and exporting log data

Name	View and Export Project Logs		
Actors	Team Lead, Team Member, Admin		
Summary	This use case will enable the users to see the specific project logs such as errors or console logs.		
Pre-Conditions	• The user must be logged in. • An AI/ML project must be created and deployed within the AI/ML infrastructure.		
Post-Conditions	The user can see the project logs.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user navigates to the project dashboard.	2	The system will display a list of all projects.
3	The user selects the project from the list.	4	The system will display the dashboard of that project.
5	The user will click on “View Logs”.	6	The system opens a log viewer or log display interface.
7	The user specifies the log type or category such as ”Project Activity Logs”, ”Error Logs”.	8	The system filters and displays the requested log type.
9	The user clicks on ”Export Project Logs”.	10	The system offers options for exporting or downloading log data.

4.6.17 Utilizing Infrastructure Tool Queue

Table 4.17: Utilizing Infrastructure Tool Queue

Describes the process by which a User interacts with the infrastructure system to access and utilize a specific tool's queue

Name	Utilizing Infrastructure Tool Queue		
Actors	User		
Summary	This use case describes the process by which a User interacts with the infrastructure system to access and utilize a specific tool's queue.		
Pre-Conditions	• The User has access to the infrastructure system. • The infrastructure system is operational and accessible.		
Post-Conditions	The User is successfully added to the queue of the desired tool.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The User navigates to the infrastructure system.	2	The System displays the details of available infrastructure resources and tools.
3	The User selects the desired tool from the list.	4	The System opens the detailed information page for the selected tool, including its current queue status.
5	The User clicks on ”Get in Queue” button.	6	The System processes the request and adds the User to the queue of the selected tool.
Alternative Flow			
5	The tool's queue is full.	5-A	The System displays a notification indicating that the tool's queue is full.

4.6.18 Un-installation of the Tools

Table 4.18: Un-installation of the Tools

Allows the user to uninstall the tool from installed tools list

Name	Un-installation of the Tools
Actors	Admin
Summary	This use case describes how a User (Admin) can uninstall the tool from the given list of already installed tools
Pre-Conditions	**Pre-Conditions:** • The user is already stored in the database and registered as a valid user and has admin access • The tool is already installed in the infrastructure.
Post-Conditions	**Post-Conditions:** Tool is uninstalled from infrastructure successfully.
Special Requirements	None

Basic Flow			
Actor Action		**System Response**	
1	User clicks on install tool from navbar	2	System shows drag and drop page
3	User drags the tool from the space to the inventory	4	System shows the tool dragged into the inventory.
5	User clicks on uninstall tool.	6	System shows message: ”Tool is uninstalled successfully”

4.6.19 Manage Notifications

Table 4.19: Manage Notifications

Allows Team Leads and Admins to manage project and cluster notifications, with options to enable specific notification types

Name	Manage Notifications
Actors	Team Leads/Admin
Summary	A user is able to manage the notifications regarding the project or overall cluster.
Pre-Conditions	The User is logged into the organization’s account on the system.
Post-Conditions	The user is successfully able to manage the notifications.
Special Requirements	None

Basic Flow			
Actor Action		**System Response**	
1	The User navigates to the main dashboard of the system.	2	The system shows the main dashboard and setting button on the navigation bar.
3	The user selects the settings button.	4	The system displays all the settings including notification settings.
5	The user turns the radio button for notifications on.	6	The system asks for multiple notifications.
7	The user accepts to turn on notifications.	8	The system sends email messages for turning on notifications.

4.6.20 Adjusting Tool Queue Limit

Table 4.20: Adjusting Tool Queue Limit

Describes the process of a User adjusting the queue limit for a specific tool within the cluster settings

Name	Adjusting Tool Queue Limit		
Actors	Admin		
Summary	This use case describes the process of a User adjusting the queue limit for a specific tool within the cluster settings.		
Pre-Conditions	• The User has access to the cluster settings.		
Post-Conditions	The User successfully adjusts the queue limit for the selected tool.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The User goes to cluster settings.	2	The System opens cluster settings page.
3	The User clicks on tool settings.	4	The System shows all the tools.
5	The User selects the tool.	6	The System opens the tool settings.
7	The User clicks on ”Set Queue Limit”.	8	The System opens a dialogue box.
9	The User enters the desired number and clicks on ”Set”.	10	The System adds the User into the tool’s queue.

4.6.21 Run Command

Table 4.21: Run Command

Allows users to run command and fetch responses from the terminal

Name	Run Command		
Actors	Team Leads, Team Member, Admin		
Summary	This use case describes how a User can run command and fetch results.		
Pre-Conditions	**Pre-Conditions:** • The User is logged into the organization’s account on the system. • The User must have the necessary permissions for the project.		
Post-Conditions	**Post-Conditions:** User is able to view output results of that command run.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user navigates to the Run command Dashboard of the system.	2	The system shows the input and output boxes.
3	The user types the command that want to enter.	4	The system takes command and shows the desired output or the error.

4.6.22 Integrate AI/ML Project with External Data Source

Table 4.22: Integrate AI/ML Project with External Data Source

Allows Team Leads and Team Members to interface an AI/ML project with an external data source, with options for selecting the data source type and connection details

Name	Integrate AI/ML Project with External Data Source
Actors	Team Leads, Team Member
Summary	This use case describes how a User (Project Manager or Administrator) can interface an AI/ML project with an external Data Source.
Pre-Conditions	**Pre-Conditions:** • The User is logged into the organization's account on the system. • The User must have the necessary permissions for the project.
Post-Conditions	**Post-Conditions:** Data Source is successfully integrated with the project.
Special Requirements	None

Basic Flow			
Actor Action		**System Response**	
1	The user navigates to the All Project Dashboard of the system.	2	The system shows the list of projects.
3	The user selects the AI/ML project they want to integrate with a data source.	4	The system loads the project details and displays project-specific settings and options.
5	The user clicks on "Data Source Management".	6	The System opens the page showing all the necessary information about the data source.
7	The user selects the "Add Data Source" option.	8	The system provides a list of available data source types, including databases, data warehouses, and external APIs.
9	The user selects the type of data source they want to integrate with.	10	The system asks the User to provide the necessary connection details for the selected data source.
11	The user saves the data source configuration.	12	The system validates the provided connection details and checks for compatibility with the selected data source.
12	The user navigates to this specific data source icon.	13	The system displays all the necessary data tables and files.
Alternative Flow			
9	The user selects a data source that is not compatible with the project.	10-A	The system displays an error "Compatibility Issues with this Data Source".
11	The user provides some wrong information about the data source.	12-A	The system shows an error "Credential Issues".

4.7 Hardware and Software Requirements

Following are the hardware and software requirements for the system.

4.7.1 Hardware Requirements

The hardware requirements for the development and deployment of the system are as follows.

Compute Infrastructure:

- We need high-performance servers that will have powerful CPUs and enough RAM that will handle workloads of AI and ML efficiently.
- GPUs (Graphics Processing Units) that are optimized for AI-related computations.

Networking:

- We need High-speed and low-latency networking equipment that will facilitate seamless communication between infrastructure components.

Storage:

- Large datasets and model artifacts can be handled effectively using high-capacity and high-speed storage options.
- data storage systems tailored for particular data types include Minio for object storage, MongoDB for NoSQL storage, and Postgres for relational storage.

Security Measures:

- Firewalls provide security protocols to guarantee data integrity and guard against illegal access.

4.7.2 Software Requirements

- Linux-based operating system for the servers, such as CentOS or Ubuntu Server.
- For managing and orchestrating containers, use Kubernetes.
- Ansible is used for configuration management and automation.
- Helm for Kubernetes package administration.
- For providing a browser-based interface for GPU utilization and group creation of AI models, see JupyterHub.
- For handling the entire machine learning lifecycle, including experimentation, reproducibility, and deployment, use MLFlow or Kubeflow.

- For effective and automated scaling based on events and metrics, use KEDA (Kubernetes-based Event-Driven Autoscaling).
- Grafana is used to create dashboards and visualize Prometheus data.
- solutions for centralized logging (like the ELK stack) to gather logs and extract information for troubleshooting.
- Trino (formerly Presto) provides federated data access and querying across different storage platforms.
- Minio for storing objects.
- For NoSQL storage, use MongoDB.
- Relational storage using Postgres.

4.8 Graphical User Interface

The tentative GUI dumps of screens that will be implemented in the system have been illustrated as follows:

4.8.1 Login

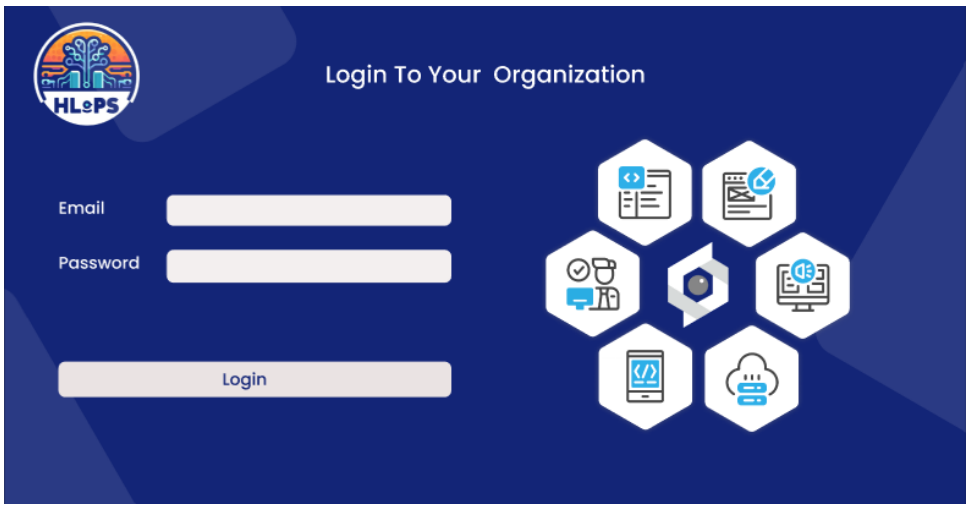

Figure 4.1: Login Screen

The figure depicts the login screen of the system used by all users.

4.8.2 Signup

Figure 4.2: Signup Screen

The figure depicts the Signup screen of the system used by root user.

4.8.3 Home Page

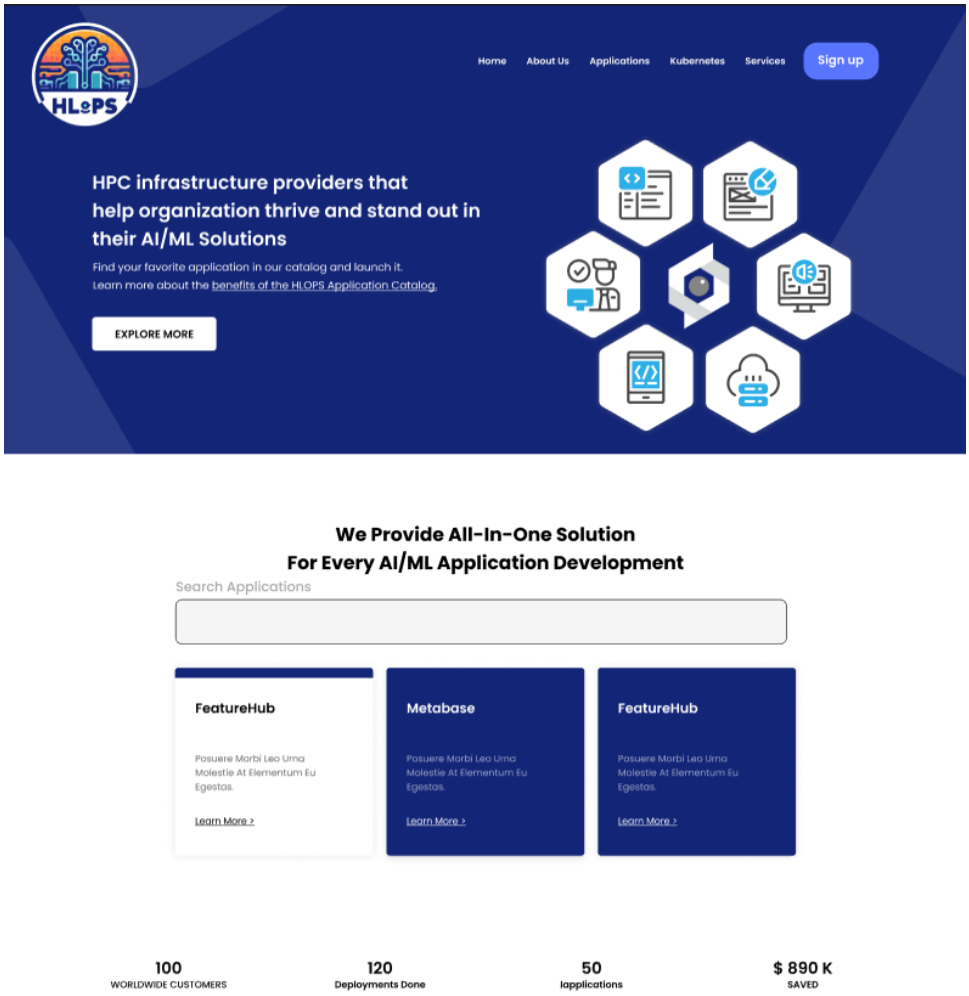

Figure 4.3: Home Screen

The figure depicts the home page of the system that will be shown to all users as a landing page. The data is sample just for the sake of GUI.

4.8.4 Manage Users

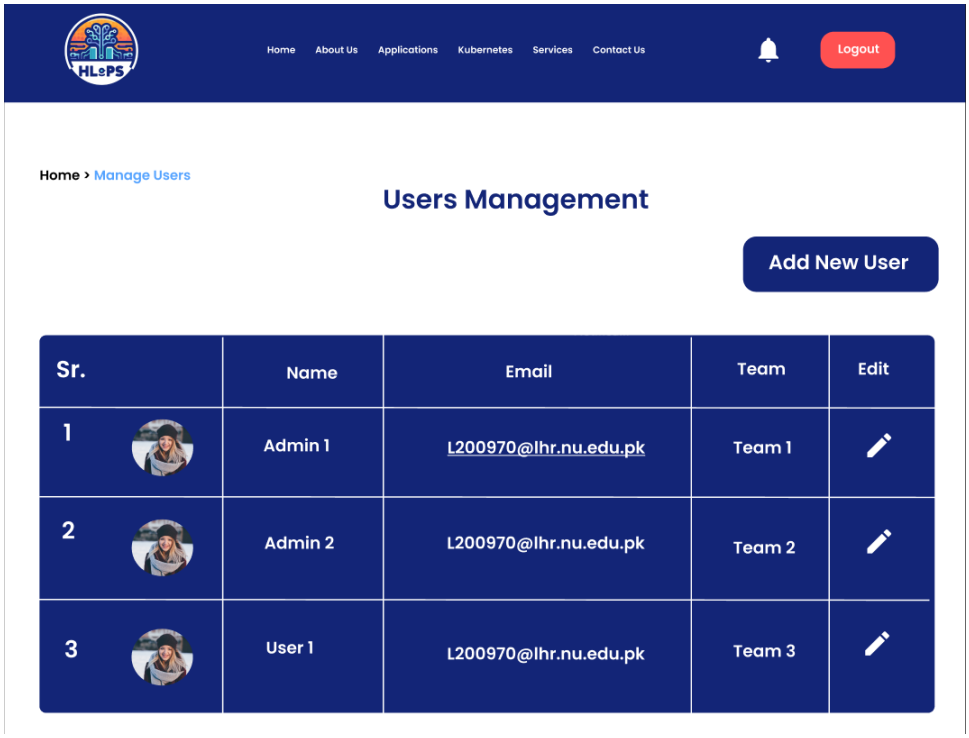

Figure 4.4: Manage Users Screen

The figure depicts the Manage Users screen of the system used by root or admin user

4.8.4.1 Add New User Modal

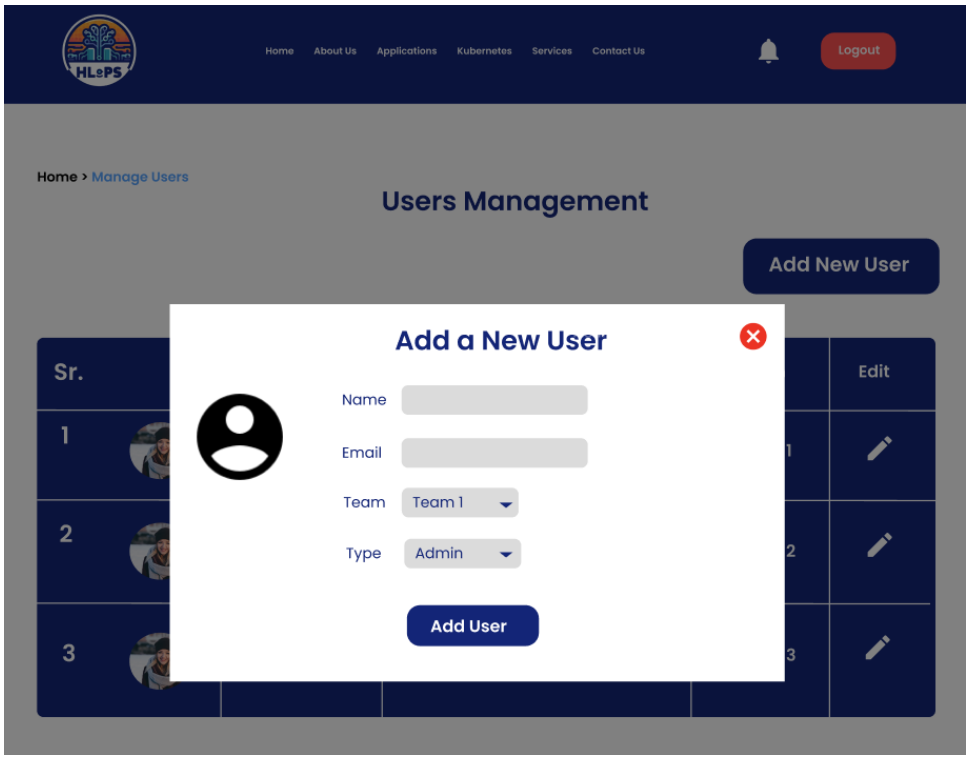

Figure 4.5: Add User Screen Modal

The figure depicts the Add User Modal Screen of the system used by root or admin user

4.8.5 Manage Clusters

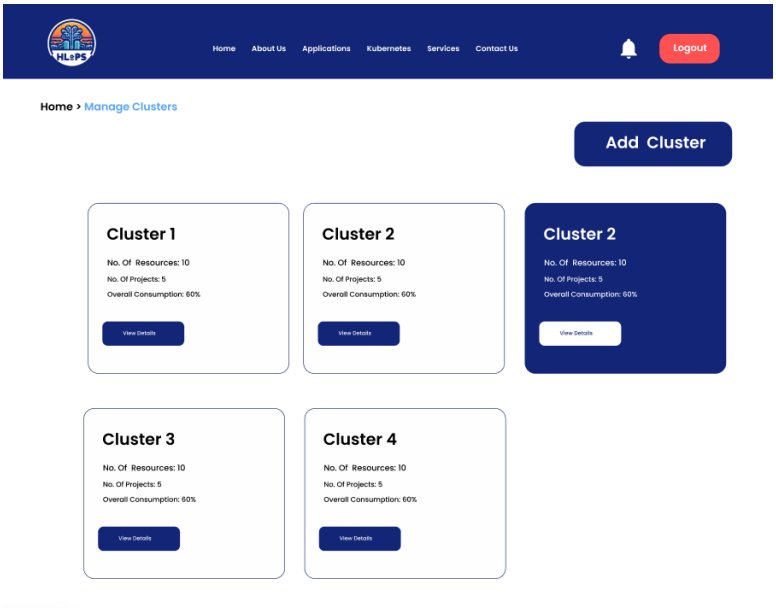

Figure 4.6: Manage Clusters Screen

The figure depicts the Cluster Management Screen. The clusters list shown is just a sample for the sake of GUI.

4.8.6 Cluster Deployment

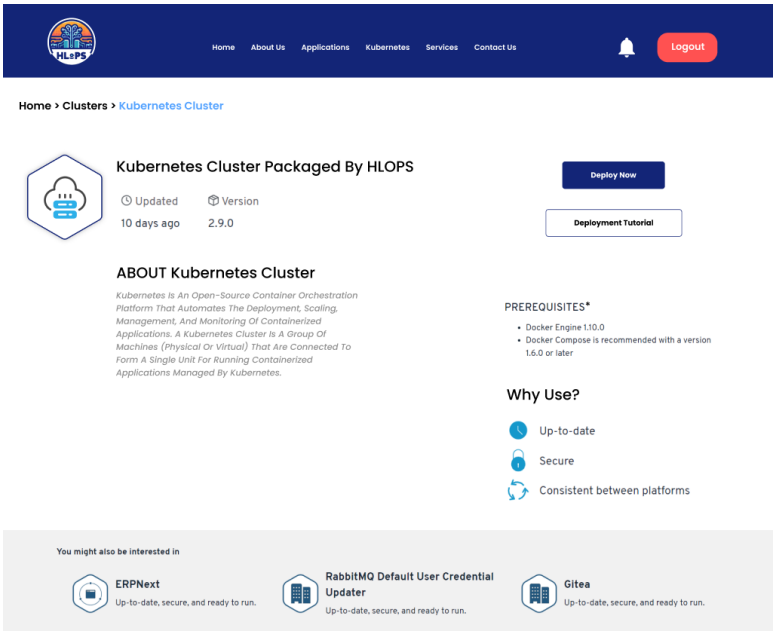

Figure 4.7: Cluster Deployment Screen

The figure depicts the Cluster Deployment. The cluster shown is just a sample for the sake of GUI.

4.8.6.1 Cluster Deployment Modal

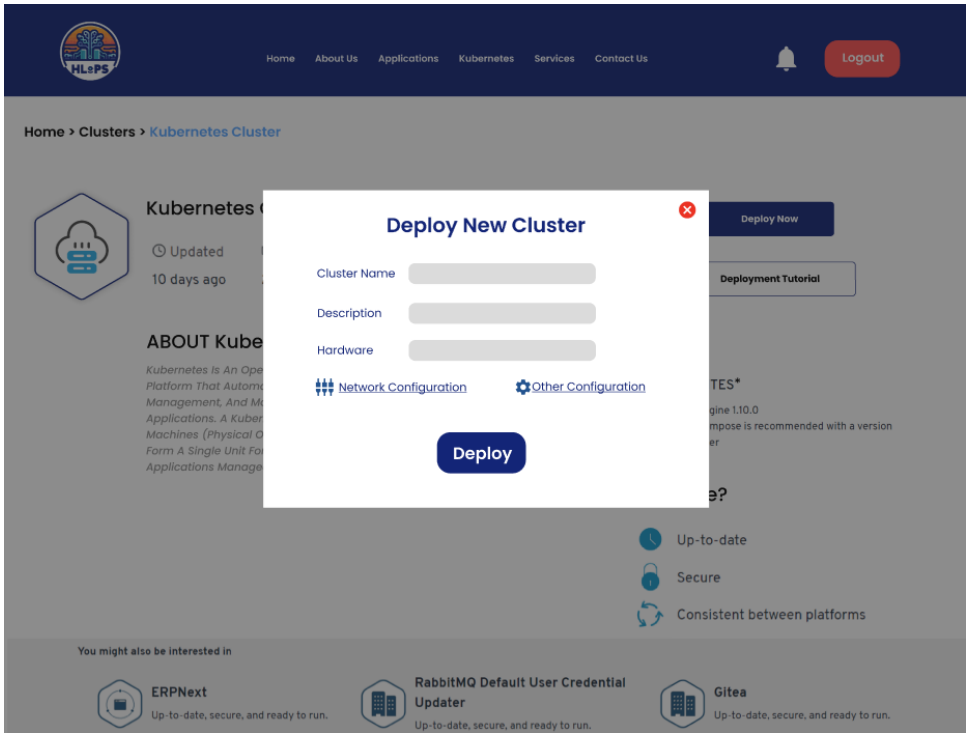

Figure 4.8: Cluster Deployment Modal Screen

The figure depicts the Cluster Deployment Modal where information will be entered by user

4.8.7 Cluster Customization

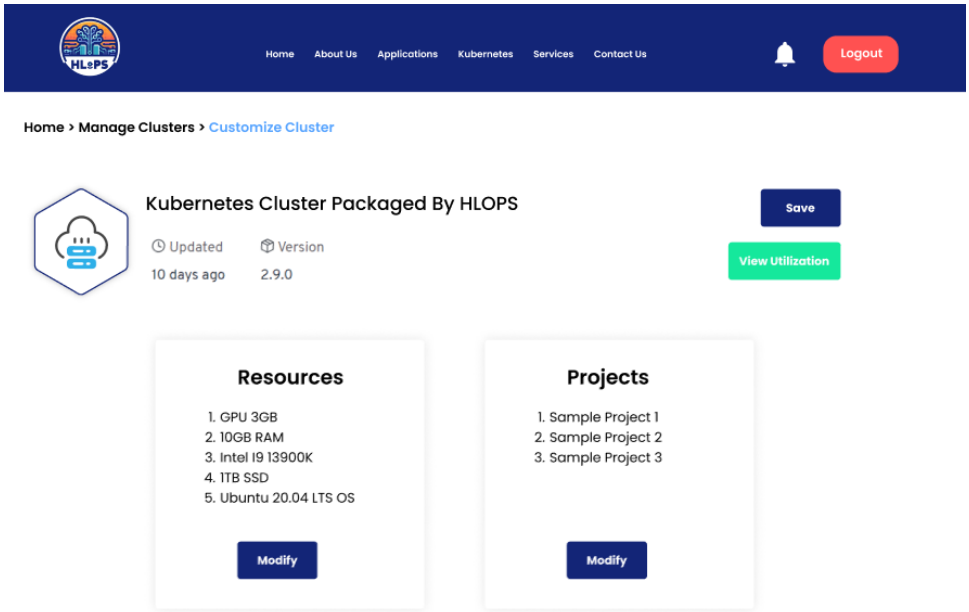

Figure 4.9: Cluster Customization Screen

The figure depicts the Cluster Customization Screen. The user can assign new or edit resources here

4.8.8 Manage Teams

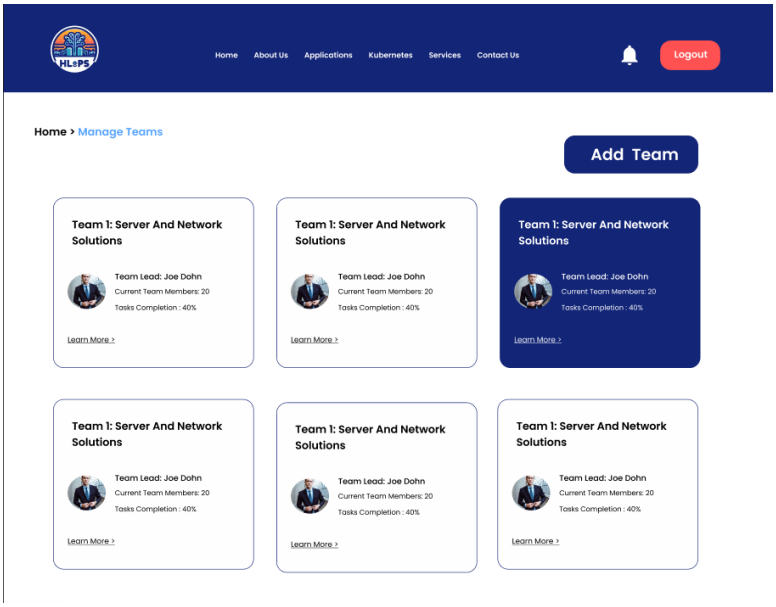

Figure 4.10: Manage Teams Screen

The figure depicts the Manage Teams screen of the system used by root or admin user

4.8.9 Resource Monitoring of Cluster

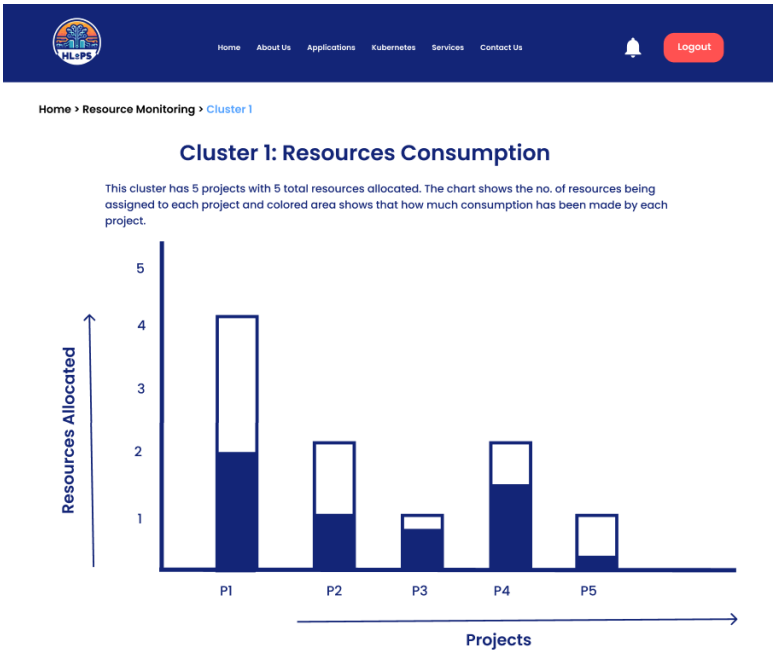

Figure 4.11: Resource Monitoring of Cluster

The figure depicts the Resource Monitoring of Cluster of the system. The graph is tentative in the figure, used just as a sample

4.8.10 Drag and Drop Tools

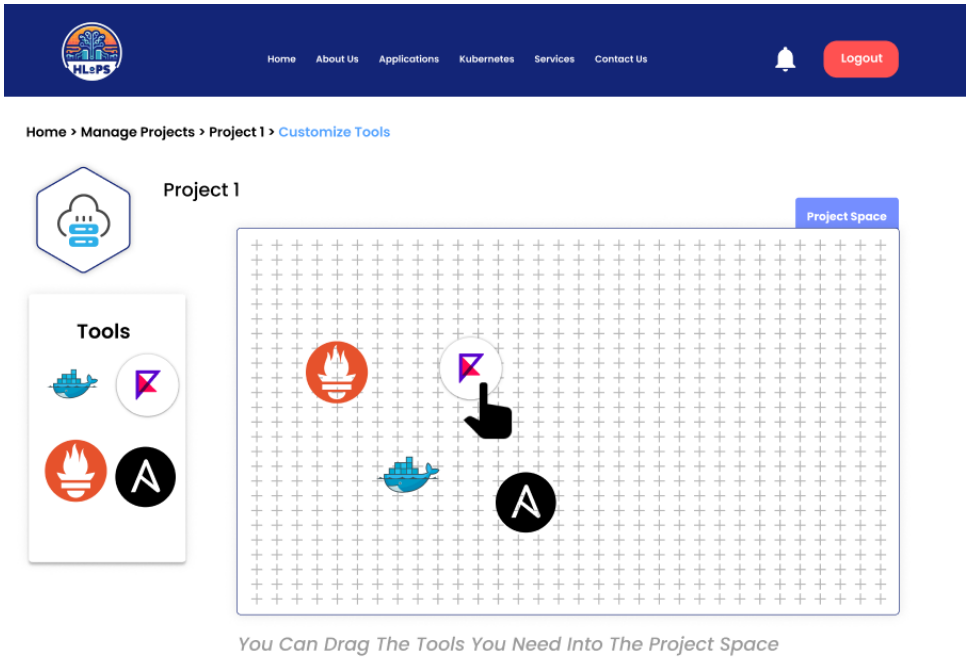

Figure 4.12: Drag and Drop Tools in Project

The figure depicts how the user can drag and drop there desired tools in the project space. The tools are tentative in the figure, used just as a sample

4.8.11 Project Logs

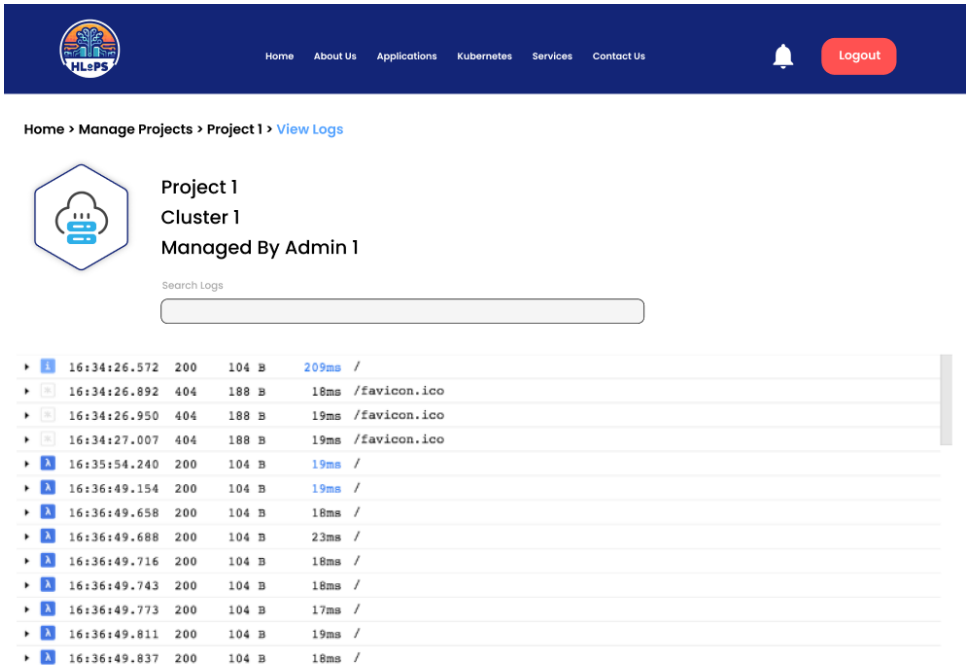

Figure 4.13: View Project Logs

The figure depicts how the project logs will be shown. The data is just a sample

4.9 Database Design

4.9.1 ER Diagram

The ER diagram of the system is divided into three main parts:

4.9.1.1 User & Organization

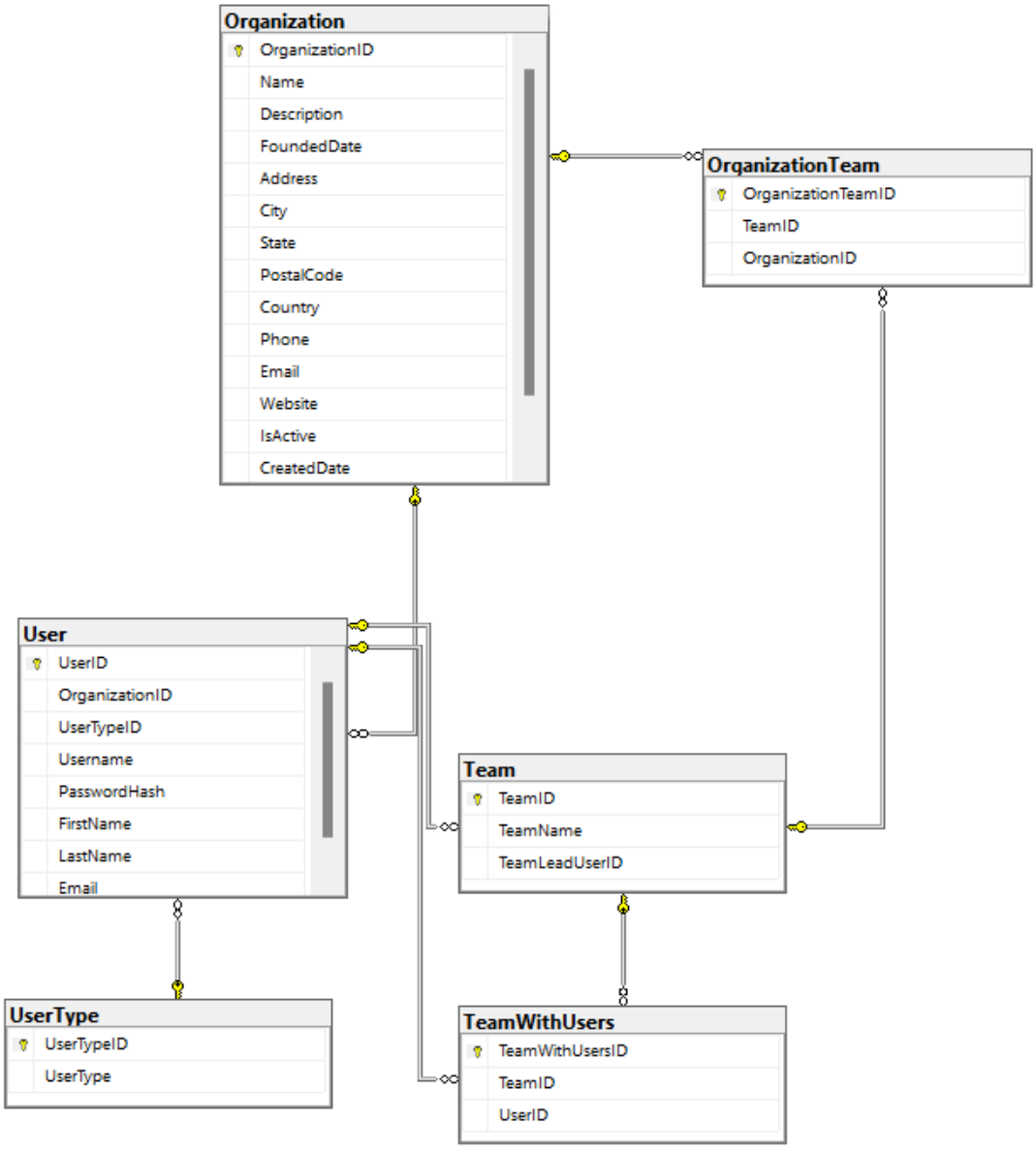

Figure 4.14: ER Diagram of the User & Organization

The figure depicts the ER Diagram of the user and organization management

4.9.1.2 Project Management

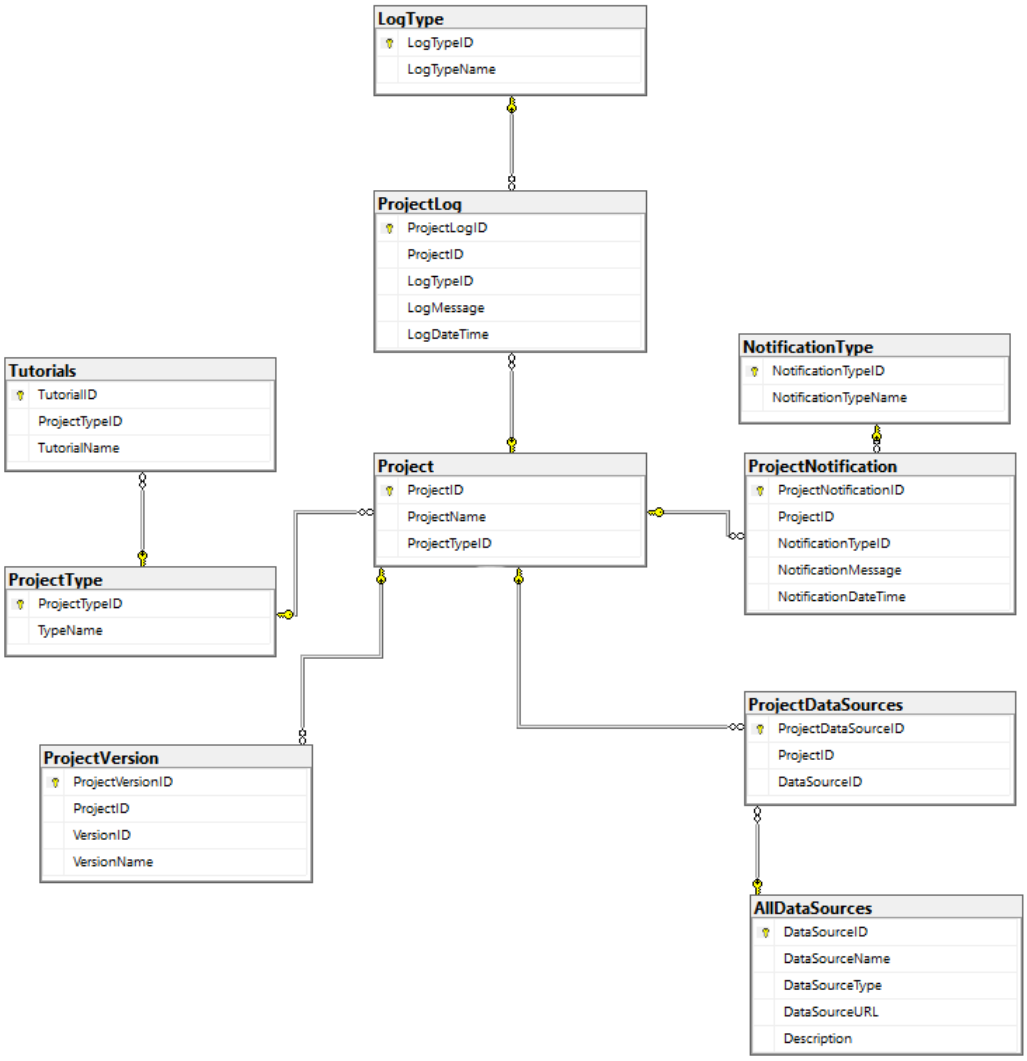

Figure 4.15: ER Diagram of the Project Management

The figure depicts the ER Diagram of the project management with other related tables

4.9.1.3 Cluster Management

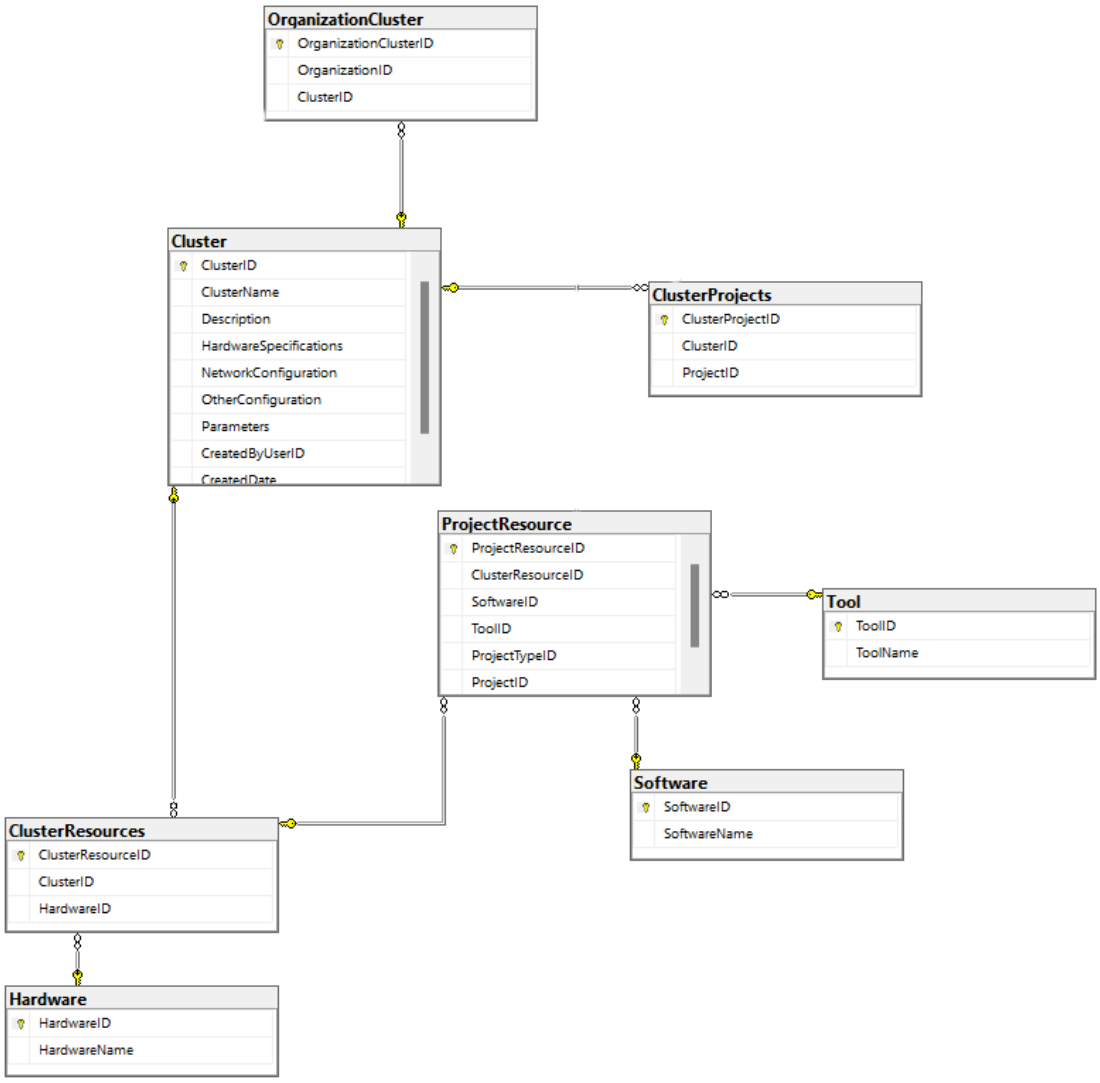

Figure 4.16: ER Diagram of the Cluster Management

The figure shows the ER Diagram of the cluster management

Each of these components are represented by their own ER diagram, which provides a clear visual overview of the database schema and its relationships. These diagrams can serve as important guidelines for implementing and managing the database of the system effectively, ensuring seamless data organization and retrieval in our system.

4.9.2 Data Dictionary

The following table shows the data dictionary for our database which gives a detail of the key data elements and their respective definitions, data types, and constraints. This data dictionary will be a useful resource for learning the structure and properties of database tables.

Table 4.23: Data Dictionary

This table shows the data members and their types also other info about the members

Entity	Attribute	DataType	Null	Def.Value	R.To	Rtype	Description
Organization	OrganizationID	INT	No				Unique identifier for each organization record.
	Name	String	No				Organization name.
	Description	String	Yes				Longer textual organization information.
	FoundedDate	Date	Yes				Founded Date
	Address	String	Yes				Physical address of the organization.
	City	String	Yes				City where the organization is located.
	State	String	Yes				State or province where the organization is situated.
	PostalCode	String	Yes				Postal or ZIP code
	Country	String	Yes				Country of organization
	Phone	String	Yes				Contact phone number
	Email	String	NO				Contact email address
	Website	String	Yes				URL of the website.
	IsActive	String	Yes				Boolean value showing active or not
	CreatedDate	Date	Yes	Current Date			Record creation time.
	ModifiedDate	Date	Yes	Current Date			Record Modification time.
UserType	UserTypeID	INT	No				Identifier
	UserType	String					Type like Admin

Entity	Attribute	DataType	Null	Def.Value	R.To	Description
User	UserID	INT	No			Unique identifier for a user.
	OrganizationID	INT	No		organization Table	Identifier for the user's organization.
	UserTypeID	INT	No		usertype table	Type of user within the organization.
	Username	String	No			User's login username.
	PasswordHash	String	No			Hashed password for security.
	FirstName	String	No			User's first name.
	LastName	String	Yes			User's last name.
	Email	String	Yes			User's email address.
Cluster	ClusterID	INT	No			Unique identifier for a cluster.
	ClusterName	String	No			Name of the cluster, which is required.
	Description	String	No			Description of the cluster.
	Hardware Specifications	String	No			Hardware specifications of the cluster.
	Network Configuration	String	No			Network configuration details for the cluster.
	Other Configuration	String	Yes			Other configuration information about the cluster.
	Parameters	String	No			Parameters (possibly stored as JSON).
	Created ByUserID	INT	No		User Table	Identifier of the user who created the cluster.
	Created Date	Date	No		Current Date	Date and time when the cluster was created, with a default of the current date and time.

Entity	Attribute	DataType	Null	Def.Value	R.To	Description
ProjectType	ProjectTypeID	INT	No			Unique identifier for a project type.
	TypeName	String	No			Name of the project type, which is required.
ToolType	ToolID	INT	No			Unique identifier for a tool.
	ToolName	String	No			Name of the tool, which is required.
Hardware	HardwareID	INT	No			Unique identifier for hardware.
	HardwareName	String	No			Name of the hardware, which is required.
Software	SoftwareID	INT	No			Unique identifier for software.
	SoftwareName	String	No			Name of the software, which is required
Cluster-Resource table	ClusterResourceID	INT	No			Unique identifier for a cluster resource.
	ClusterID	INT	No		Cluster-Table	Identifier of the cluster to which the resource belongs, which is not nullable.
	HardwareID	INT	No		Hardware Table	Identifier of the hardware resource, which is not nullable.
organization Cluster	Organization ClusterID	INT	No			Unique identifier for an organization-cluster relationship.
	Organization ID	INT	No		Organization Table	Identifier of the organization, which is not nullable.
	ClusterID	INT	No		Cluster Table	Identifier of the cluster, which is not nullable.
Project Resource	Project ResourceID	INT	No			Unique identifier for a project resource.
	Cluster ResourceID	INT	No		Cluster Resource Table	Identifier of the cluster resource from the ”ClusterResources” table.
	SoftwareID		No		Software Table	Identifier of the software from the ”Software” table.
	ToolID	INT	No		Tool Table	Identifier of the tool from the ”Tool” table.
	Project TypeID	INT	No		Project Type	Identifier of the project type from the ”ProjectType” table.

Entity	Attribute	DataType	Null	Def.Value	R.To	Description
	ProjectID	INT	No		Project Table	Identifier of the project to which this resource is related.
Project	ProjectID	INT	No			Unique identifier for a project.
	Project Name	String	No			Name of the project, which is required.
	Project TypeID	INT	No		Project Type Table	Type of the Project
Cluster Projects	Cluster ProjectID	INT	No			Unique identifier for a cluster-project relationship.
	ClusterID	INT	No		cluster table	Identifier of the cluster, which is not nullable.
	ProjectID	INT	No		project table	Identifier of the project, which is not nullable.
Team Table	TeamID	INT	No			Unique identifier for a team.
	TeamName	String	No			Name of the team , which is required.
	TeamLead UserID	INT	No		user table	Identifier of the user who leads the team, which is not nullable.
TeamWithUsers	TeamWith UsersID	INT	No			Unique identifier for a team-user relationship.
	TeamID	INT	No		team table	Identifier of the team, which is not nullable.
	UserID	INT	No		user table	Identifier of the user, which is not nullable.
Organization Team	OrganizationTeamID	INT	No			Unique identifier for an organization -team relationship.
	TeamID	INT	No		Team Table	Identifier of the team, which is not nullable.
	OrganizationID	INT	No		Organiza- tion Table	ID of the organ ization.
Tutorial Table	TutorialID	INT	No			id for a tutorial.
	ProjectTypeID	INT	No		Project Table	Id of Project type
	TutorialName	String	No			Name of the tutorial

Entity	Attribute	DataType	Null	Def.Value	R.To	Description
Log Type	LogTypeID	INT	NO			ID of the log
	LogTypeName	String	NO			Name of the log
Project Log	ProjectLogID	INT	NO			ID of the project log.
	ProjectID	INT	NO		Project Table	ID of the associated project.
	LogTypeID	INT	NO		Log Table	ID of the log type.
	LogMessage	String	Yes			Message in the log.
	LogDateTime	String	NO			Date and time of the log.
Notification Type	Notification TypeID	INT	NO			ID of the notification type
	Notification TypeName	String	NO			Name of the notification type.
Project Notification	Project NotificationID	INT	NO			ID of the project notification.
	ProjectID	INT	NO		Project Table	ID of the associated project.
	Notification TypeID	INT	NO			ID of the notification type.
	Notification Message	String	Yes			Message in the notification.
	Notification DateTime	Date	NO	Current Date		Date and time of the notification

This data dictionary provides a comprehensive overview of the key fields within our database tables, helping to ensure data integrity and consistency throughout the system.

4.10 Risk Analysis

This section outlines the potential risks that can occur in this project. They are as follows:

1. **Sourcing Hardware:**We may encounter difficulties procuring high-performance infrastructure components like as servers, GPUs, and VMWare resources. So, it may create delays in project setup and deployment.

2. **Software Compatibility Issues:**It may be challenging to ensure that all software components, such as operating systems, AI/ML frameworks, and orchestration tools, perform properly together due to versioning problems.

3. **Maintaining Open Source Project:**It will be an open-source project, and it is always challenging to maintain open-source projects. Therefore, finding community support for this project could be challenging.

4. **Data Security and Privacy:** Keeping sensitive data safe and secure is really important. If there's

a data breach or privacy issue, it could cause legal and reputation problems

It is important to be aware of these risks and take steps to prevent them. By doing so, we can increase our chances of a successful project and avoid any unnecessary setbacks.

4.11 Conclusion

This chapter presents a detailed and comprehensive set of Software Requirement Specifications for the system. The chapter meticulously outlines various use cases, delineating the roles of actors like Team Leads, Admins, and Team Members in managing different aspects of the system. Key functionalities such as managing admins and teams, creating and assigning team leads, adding and removing team members, and handling project logs are thoroughly described, each with a clear set of pre-conditions, post-conditions, and step-by-step flows. The chapter also discusses critical hardware and software requirements needed to effectively build and maintain this infrastructure, emphasizing the need for high-performance servers, GPUs, and specific software like Kubernetes and MLFlow. Additionally, it addresses potential risks associated with the project, including hardware sourcing challenges, software compatibility issues, open-source maintenance, and data security concerns. The systematic approach in specifying requirements, coupled with a keen focus on user interaction and system functionality, demonstrates a profound understanding of the complexities involved in managing AI/ML projects and the infrastructure that supports them. This chapter lays a solid foundation for the successful development, deployment, and maintenance of a robust and efficient AI/ML HPC infrastructure system.

Chapter 5 High-Level and Low-Level Design

This chapter explores the project's design aspects, both from a high-level and low-level perspective. It encompasses an overview of the system, design factors, the system's architecture, strategies for architecture, as well as class diagrams and sequence diagrams. Additionally, it covers the policies and tactics associated with the project.

5.1 System Overview

This software system is divided into four integral components, each tailored to streamline the AI/ML infrastructure from the ground up. The system is designed with a clear distinction between the client-side experience and server-side operations, ensuring not only a user-centric interface but also a robust, scalable backend that can adapt to the rapidly changing data landscape. With this structure, SMEs can leverage a state-of-the-art platform for developing and deploying AI/ML models, enhancing their operational efficiency without the complexities traditionally associated with such endeavors. A detailed description of the system architecture is provided in Section 5.3.

5.2 Design Considerations

This section describes many issues that need to be addressed or resolved before attempting to devise a complete design solution.

5.2.1 Assumptions and Dependencies

The following are some assumptions that we are making while designing this system:

- The user will have the hardware components necessary to support the resource-intensive nature of ML/AI processes.
- The Kubernetes cluster and the local infrastructure are set up to accept and run deployment scripts from the web portal.
- User projects are assumed to be compatible with the provided configuration tools.
- End user has the basic knowledge about the deployment of the cluster.
- End user is aware of resource management to avoid conflicts between different projects.
- User local infrastructure such as GPUs have support to configure the MLOPs environment.

Following dependencies will govern the working of this system:

- ML/OPS assumes Kubernetes as the orchestration platform. Therefore, the system needs Kubernetes for infrastructure management and orchestration.
- ML/OPS uses Helm for Kubernetes package management and Ansible for configuration management. Without these tools, automated setup and deployment, essential for system consistency and Kubernetes cluster application deployment, would be difficult.
- Without compatible GPU drivers and software libraries (e.g., CUDA for NVIDIA GPUs), the system cannot use GPU resources.
- There must be reliable network connectivity which is very crucial for communication between components and potential interactions with external data sources.

5.2.2 General Constraints

The global limitations or constraints that have a significant impact on the design of the system's software are provided in the sub-sections below.

5.2.2.1 Hardware and software environment

- Different hardware like GPUs and software versions may not operate well together, producing operational and performance problems.
- Some tools might not easily work with different kinds of projects because they aren't configured to do so, causing potential compatibility issues.
- Internet not connected.

5.2.2.2 Availability or volatility of resource

- Limited availability or capacity of GPUs will impact the smooth operation of resource-intensive ML operations which will lead to reduce the performance of the system.

5.2.2.3 End-user environment

- Insufficient end-user technical knowledge.
- The local infrastructure does not support the HPC on-premises infrastructure.
- The end-user platform is not correctly connected to the web platform.

5.2.2.4 Network communications

- The project operates in an environment with limited network bandwidth.

- Firewall policies restrict certain types of network communication.
- High Latency Across Distributed Nodes.

5.2.2.5 Data Repository and Distribution Constraints

- Project data sources are not connected.
- Authentication Issues in Data Access.
- Incompatibility in data distribution protocols across components.

5.2.2.6 Interoperability Constraints

- There are different types of projects so interoperability issues between tools and projects can occur.
- The project relies on external APIs and communication protocols that may change independently.

5.2.2.7 Performance Requirement Constraints

- The processing capacity of the GPUs is insufficient for intensive computing jobs.
- The local infrastructure does not fully meet the compatibility requirements for high-performance computing.
- Insufficient RAM and processing capacity on local machines or servers.

5.2.3 Goals and Guidelines

The following outlines the key goals and guidelines that have shaped the design and development of this project:

- User Experience and Usability: Designing a system that is intuitive and easy to use, especially for SMEs with limited technical expertise. Therefore, prioritizing user interface simplicity with tutorials and clear documentation.
- Security and Data Protection: A key concern for SMEs is the security vulnerabilities and regulatory compliance when dealing with AI/ML infrastructure.This would be a tough challenge for us.The main purpose of this project is to create a secure and user-friendly AI/ML infrastructure solution.It also guarantees data protection by storing sensitive information in an encrypted form on dedicated servers.
- Scalability and Flexibility: Creating a system that can easily scale and adapt to varying workloads

and evolving AI/ML technologies. As a result establishing a modular system architecture employing scalable technologies like Kubernetes which ensures compatibility with AI/ML inventions.

- Automation and Simplification: One main purpose is less the difficulty of setting up and managing AI/ML infrastructure for SMEs by automating repetitive operations which simplifies deployment using tools such as Ansible.

In conclusion, these goals and guidelines are important for the system and ensures that it is an advanced, efficient, accessible, secure, and future-proof system.

5.2.4 Development Methods

In this project we will be using scrum approach. Scrum is an agile development methodology that helps to enhance software that depends on cycles that are repeatable and iterative. There will be multiple sprints for the work and each with a deadline. This time frame will not be longer than a week. After every sprint we will have a 15-20 minute meeting to discuss progress and make any necessary modifications to the overall schedule. Scrum is very flexible development strategy that makes it simpler to add new features or change projects that is why employing it makes sense.

5.3 System Architecture

The architecture of the system is a cornerstone of its functionality offering a comprehensive infrastructure that supports a wide range of operations essential in high-performance computing environments especially in AI/ML contexts. We divides it into four key components each serving a distinct but interconnected role. These components include the Frontend, Backend, Kubernetes, and Helm Charts & Operators. Together they form a cohesive unit that ensures efficiency, scalability and flexibility, crucial for modern AI/ML applications. There is a detailed diagram below which shows this

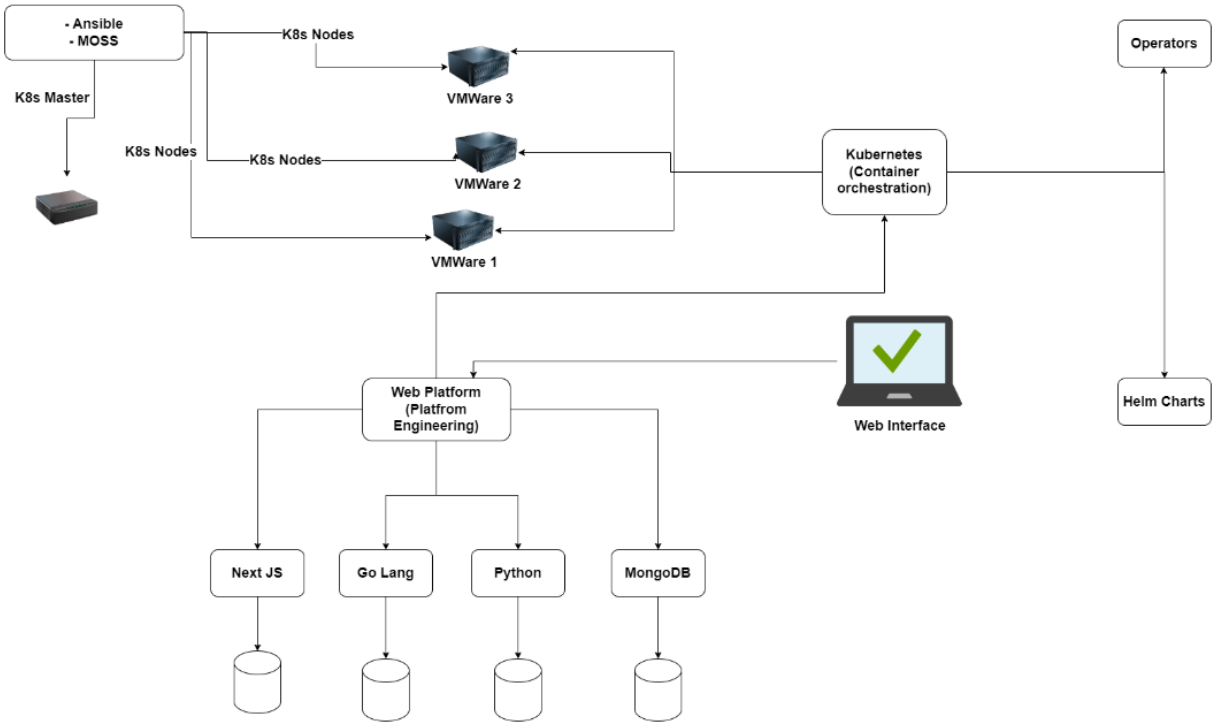

Figure 5.1: System Architecture Diagram

The figure represents system's infrastructure, highlighting container orchestration and diverse programming frameworks

5.3.1 Frontend

The front end of the system is created with Next.js to provide a dynamic user experience with server-side rendering for speed. The frontend is critical because it not only provides a dynamic user experience but also integrates element like responsive design and accessibility.

5.3.2 Backend

Kubernetes is used in the construction of the backend to manage a group of microservices. It employs a two-language strategy with Python handling AI/ML and data processing and Go handling high-efficiency tasks. Although MongoDB is used for data storage in this structure, other databases may be used as well.

5.3.3 Kubernetes

The architecture employ Kubernetes for different types of microservices orchestration, with a language-agnostic approach also using Go for performance-centric services and Python for AI/ML and data tasks, supported by MongoDB for data storage.

5.3.4 Helm Charts And Operators

A comprehensive DevOps approach is shown by the usage of Helm Charts and Operators which make it possible to manage the application lifecycle on Kubernetes smoothly. This ensures SMEs don't need to have in-depth knowledge of the underlying technologies in order to maintain a continuous delivery pipeline for their AI/ML apps.

Finally, the system architecture described here is a sophisticated and unified framework developed for AI/ML high-performance computing. It is made up of four major components, each of which plays an important role in assuring efficiency, scalability, and adaptability. So, the main purpose of helm chart in our project is to install and configure different types of tools.This framework that combines a dynamic user interface with reliable backend processing and advanced MLOps/DevOps approaches in a Kubernetes environment is very useful for SMEs operating complicated AI/ML applications. The integrated design is demonstrated by the whole system diagram which also highlights the system's effective operation.

5.4 Architectural Strategies

The architectural strategies described in the following section provide a clear view of the key abstractions and mechanisms that includes the system's architecture. Architectural Strategies are very important for a project. These strategies are thoughtfully selected to fulfill the system's goals and objectives, ensuring an optimal balance between robust performance and ease of use for end-users.

5.4.1 Use of the Frontend Stack: Next.js and MongoDB

With its server-side rendering capabilities Next.js offers an engaging and dynamic frontend user experience that is essential for high-performance apps. That is why we are using Next js in our project. Moreover, because of its adaptable document structure, MongoDB is used to store and retrieve data in a way that is compatible with the object-oriented programming style, which improves application efficiency and developer productivity.

5.4.2 Backend Technologies: Go, Python, and Kubernetes

The backend architecture leverages the Go language for its lightweight nature and efficiency in handling concurrent operations, making it ideal for performance-critical backend services. Python is used for its strong support in AI/ML and data processing tasks. In python we will try to use python flask. Kubernetes stands as the backbone for container orchestration, ensuring that services are scalable and resilient.

5.4.3 DevOps Tools: Helm Charts and Operators

Helm Charts and Kubernetes Operators represents a strategic choice for managing complex deployments and maintaining a consistent development, delivery, and maintenance cycle for AI/ML applications. They also enable developers to automate the deployment and manage the application lifecycle efficiently.

5.4.4 Infrastructure Automation: Ansible and MOSS

Ansible is employed for automating infrastructure provisioning, configuration management, and application deployment, facilitating a repeatable and reliable environment setup. MOSS provides a platform for container application deployment which enhances the system's operatability and security.

5.4.5 Error Detection and Recovery

A strong error detection and recovery system are necessary to preserve system integrity. This include advanced technologies for recording, monitoring and alerting that enable early problem identification and resolution. Furthermore it guarantees great availability and dependability of the services.

5.4.6 Communication Mechanism

REST APIs enables loosely coupled service to exchange data in JSON format between the frontend and backend in a consistent and stateless manner. This ensures that the architectures multiple components communicate well, enabling integration and modular development.

5.4.7 Scalability Strategie

The system is intended to grow horizontally which allows us for the addition of resources (such as Kubernetes cluster nodes) to handle rising loads. This scalability is extended further by the use of load balancing and service discovery algorithms.It ensures optimal resource use and performance.

5.4.8 Architectural Pattern

The microservices architectural pattern is a pivotal aspect of the system. It enables independent deployment and scaling of individual service components.It is fundamental to achieving a robust and scalable AI/ML HPC infrastructure.

To finalize these architectural techniques and design patterns comprise a complete approach to developing a scalable, efficient and dependable system. The use of Next.js and MongoDB on the front end, Go and Python on the back end, Kubernetes for orchestration, Helm Charts and Operators for deployment

and Ansible for infrastructure automation all within a microservices architecture embodies a modern system designed for flexibility and growth.

5.5 Domain Model/Class Diagram

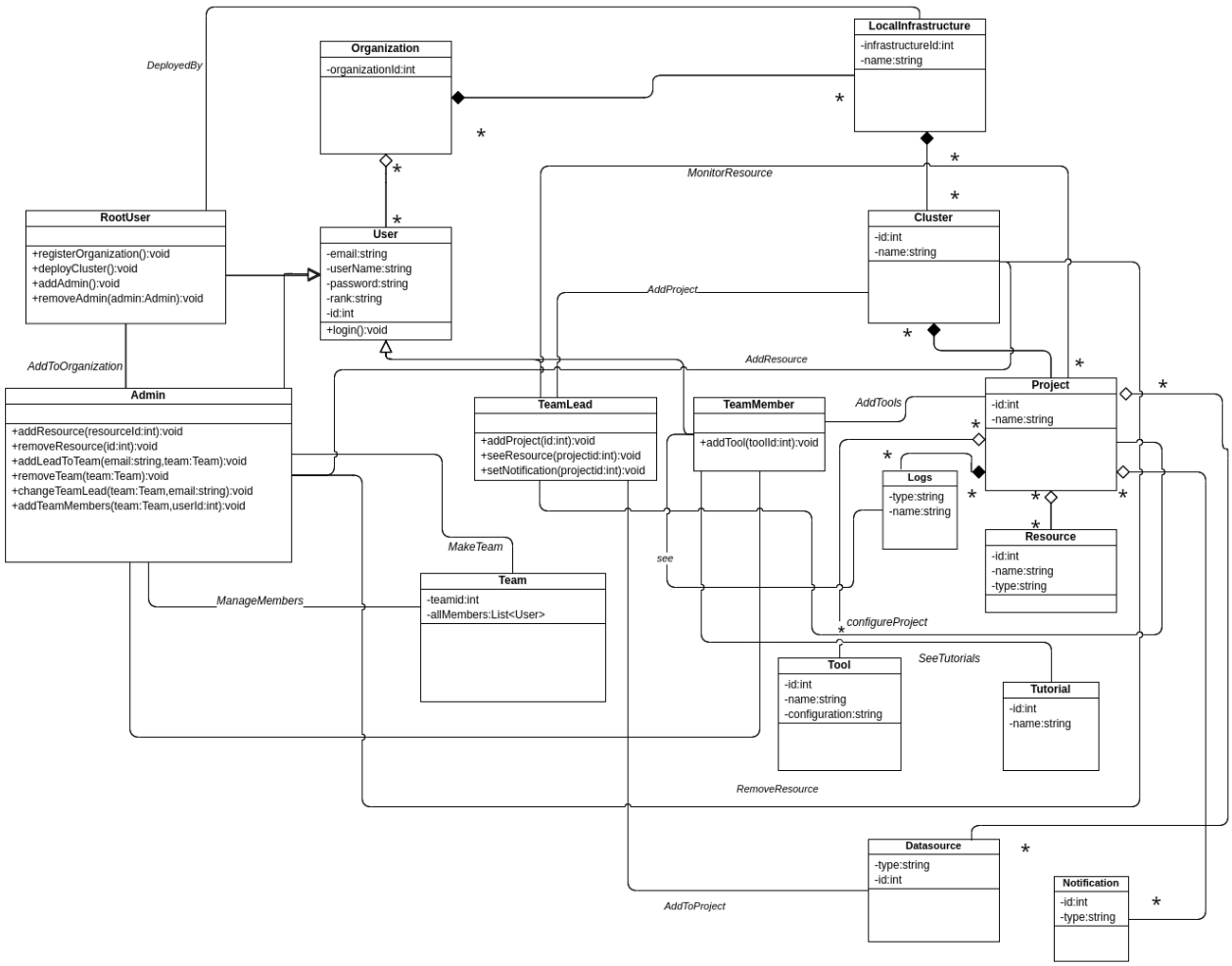

Figure 5.2: Class Diagram

Figure shows classes and the relations

5.6 Sequence Diagrams

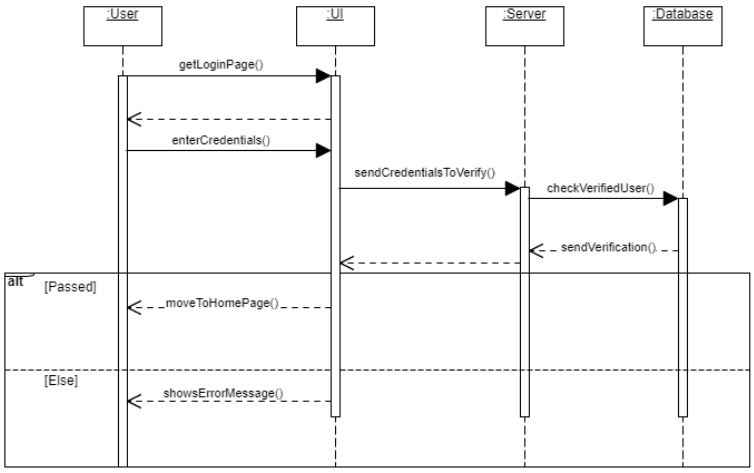

Figure 5.3: Login Sequence Diagram

The sequence diagram of login a user is given above

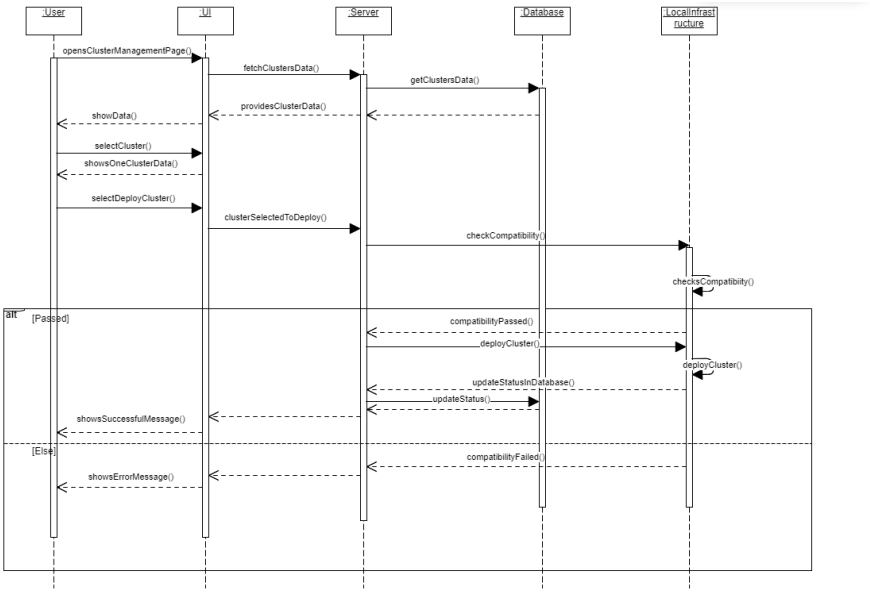

Figure 5.4: Deploy a Cluster

The sequence diagram of deploy cluster is given above

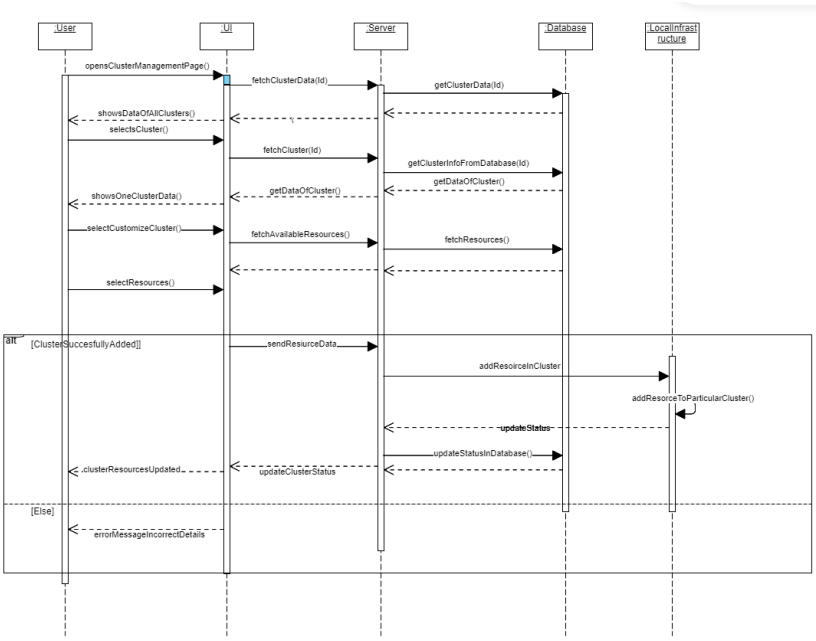

Figure 5.5: Add Cluster Resources

The sequence diagram of add cluster resources is given above

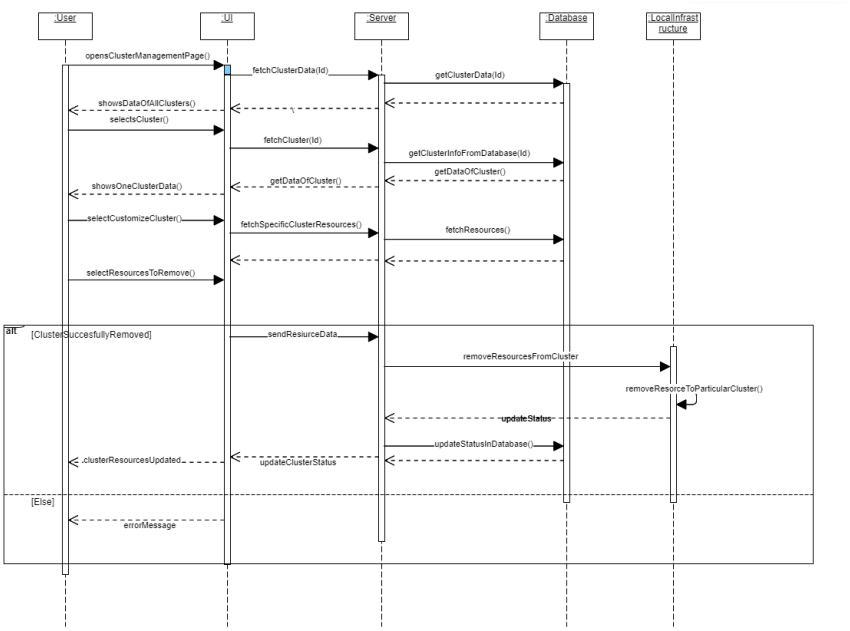

Figure 5.6: Remove Cluster Resources

The sequence diagram of remove cluster resources is given above

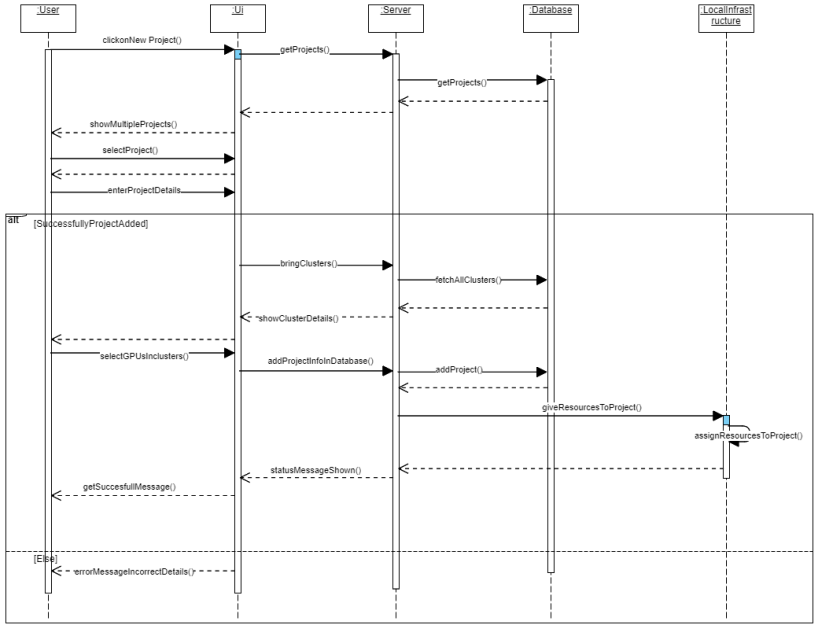

Figure 5.7: Add a New Project

The sequence diagram of add new project is given above

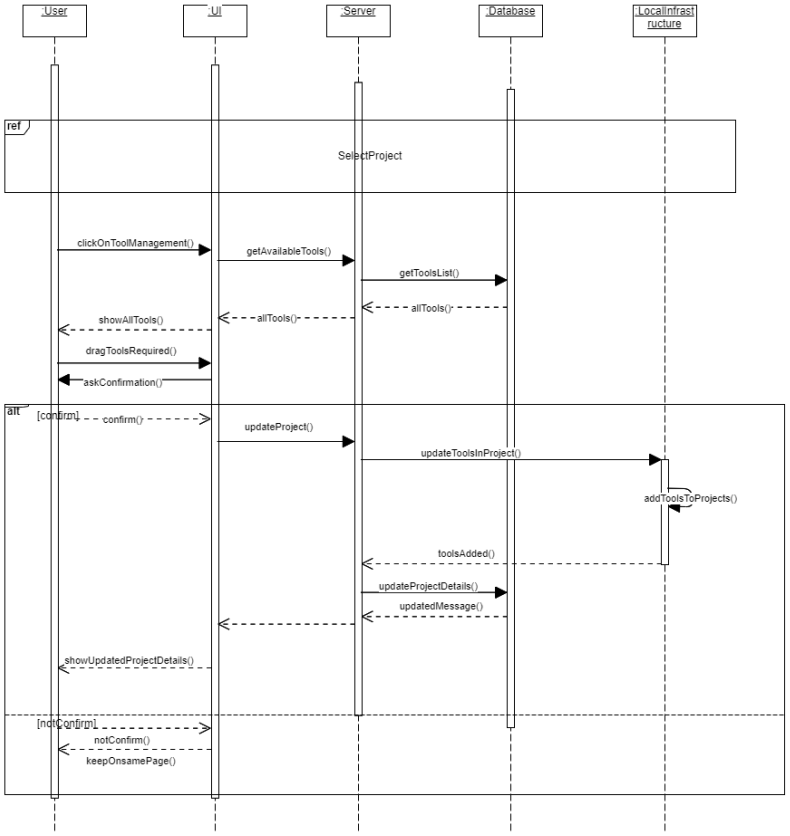

Figure 5.8: Drag and Drop Tools to Project

The sequence diagram of Drag and Drop Tools to Project is given above

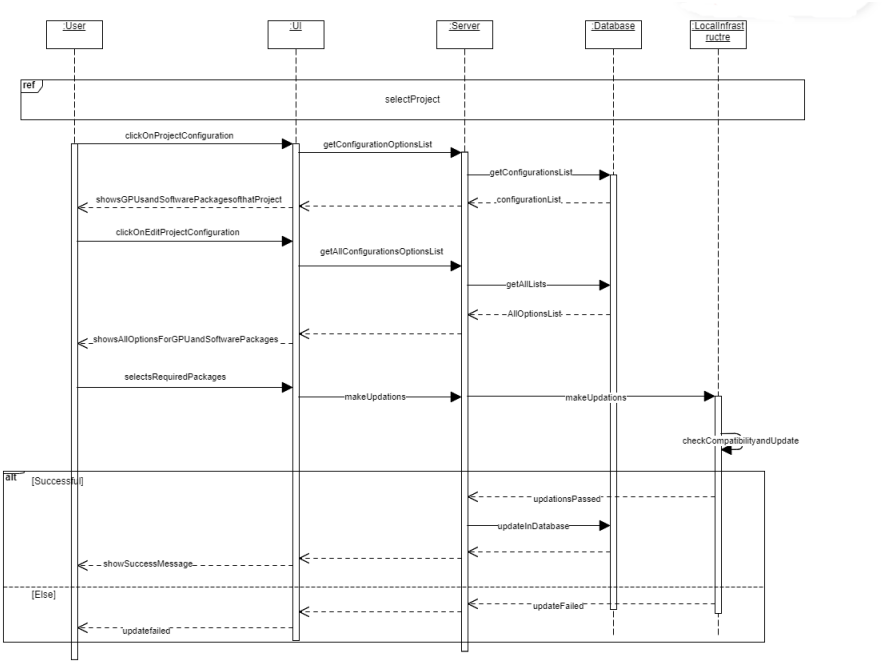

Figure 5.9: Customize Project Configuration with GPU Resources and Software Packages

The sequence diagram of customizing Project Configuration with GPU Resources and Software Packages is given above

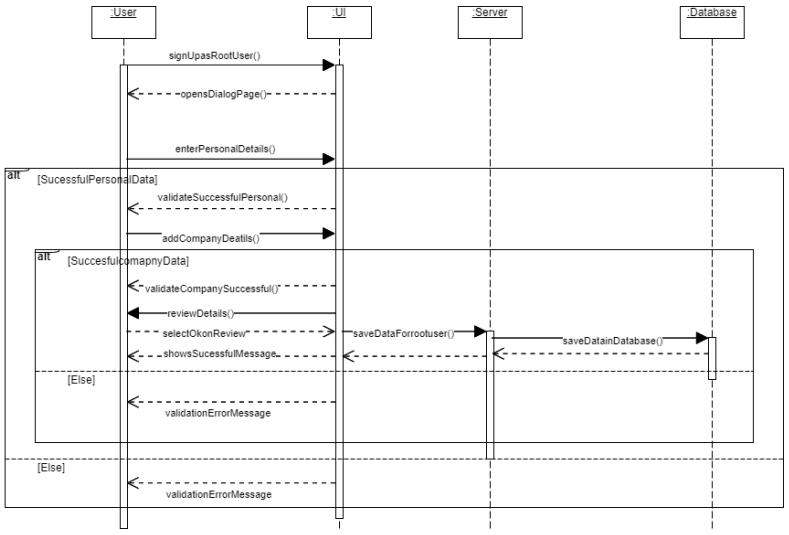

Figure 5.10: Resgister as Root User

The sequence diagram of register as a root user is given above

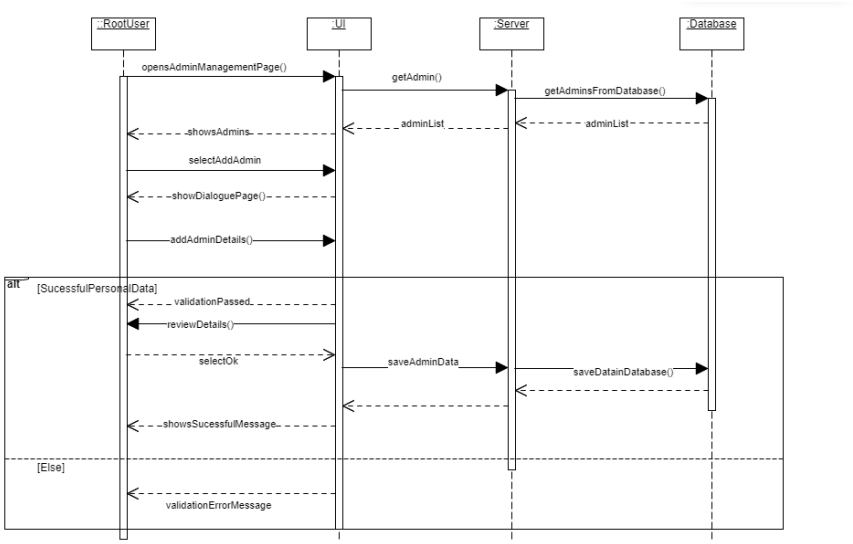

Figure 5.11: Add an admin

The sequence diagram of add an admin is given above

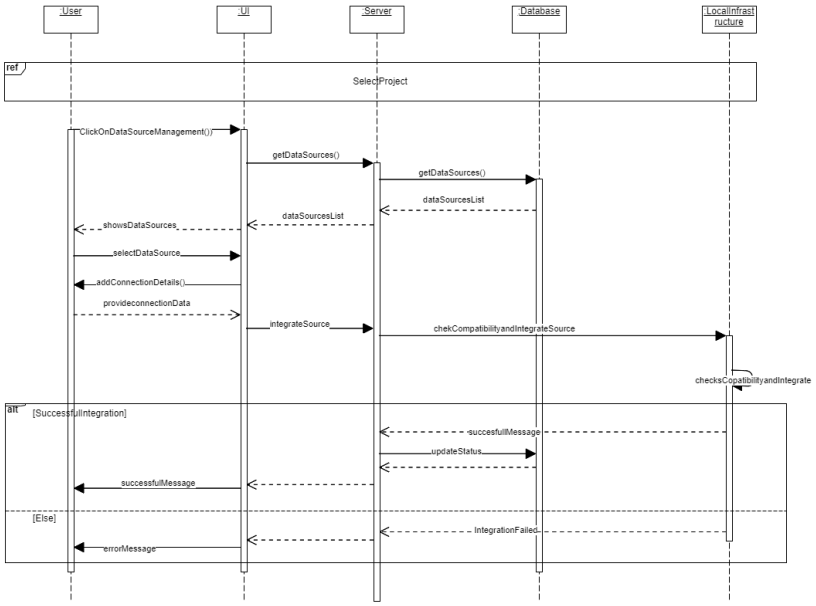

Figure 5.12: Integrate AI/ML Project with External Data Source

The sequence diagram of integrate AI/ML Project with External Data Source is given above

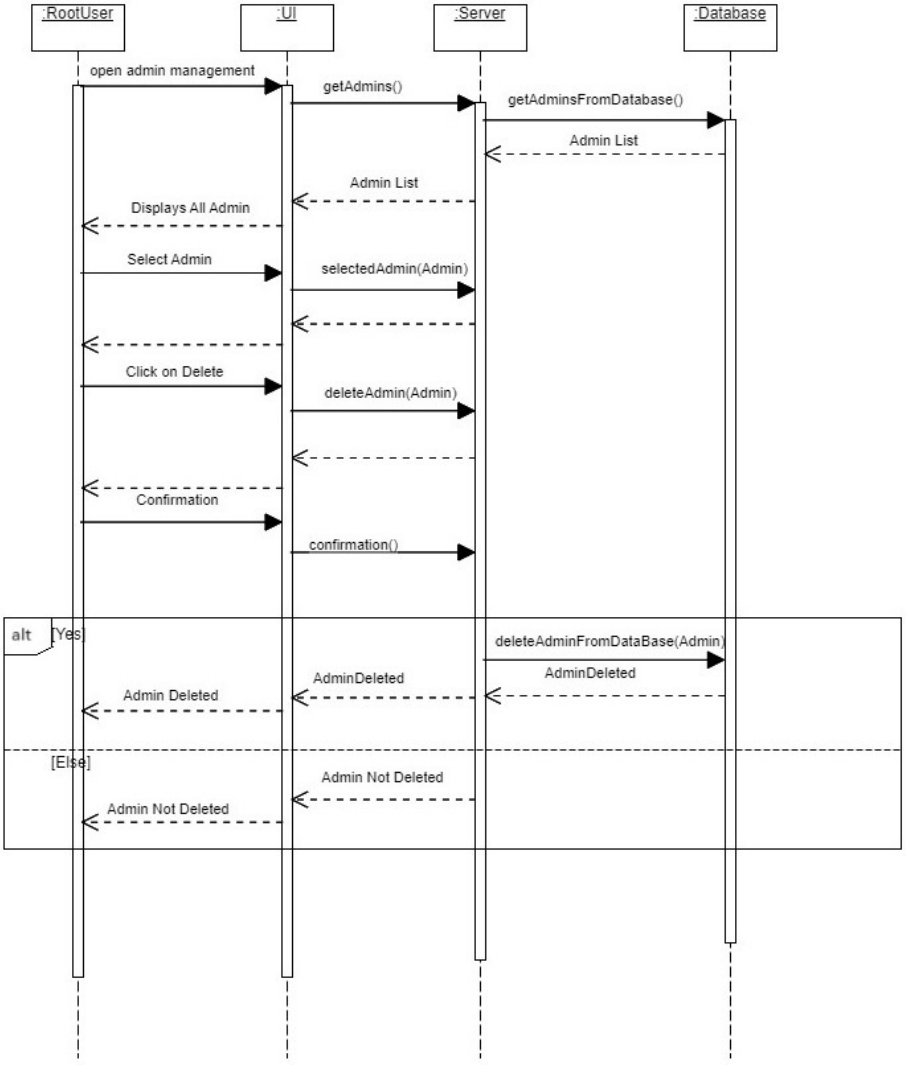

Figure 5.13: Remove an Admin

The sequence diagram of remove an admin is given above

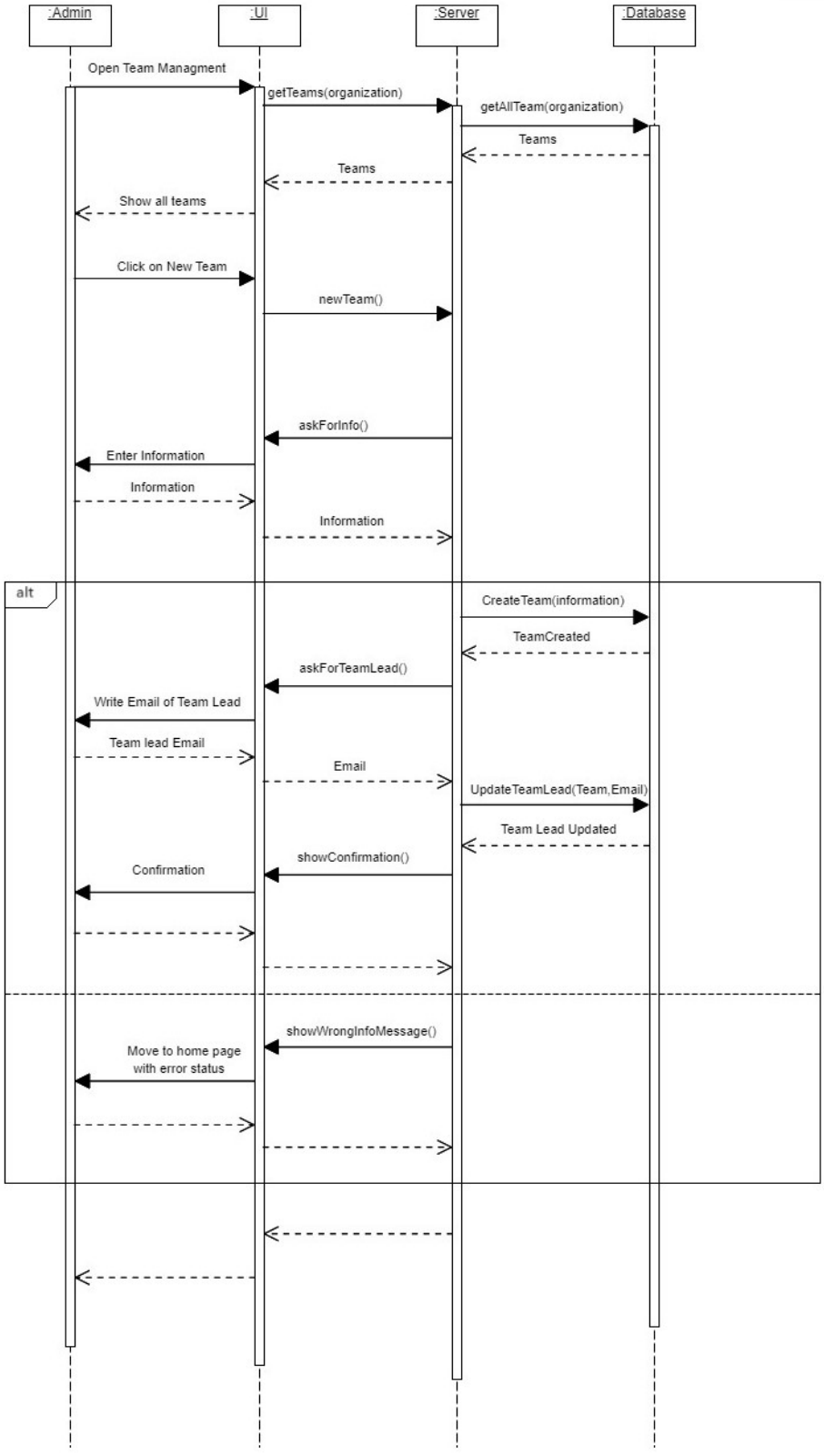

Figure 5.14: Make a team and assign Team Leads

The sequence diagram of make a team and assign team leads is given above

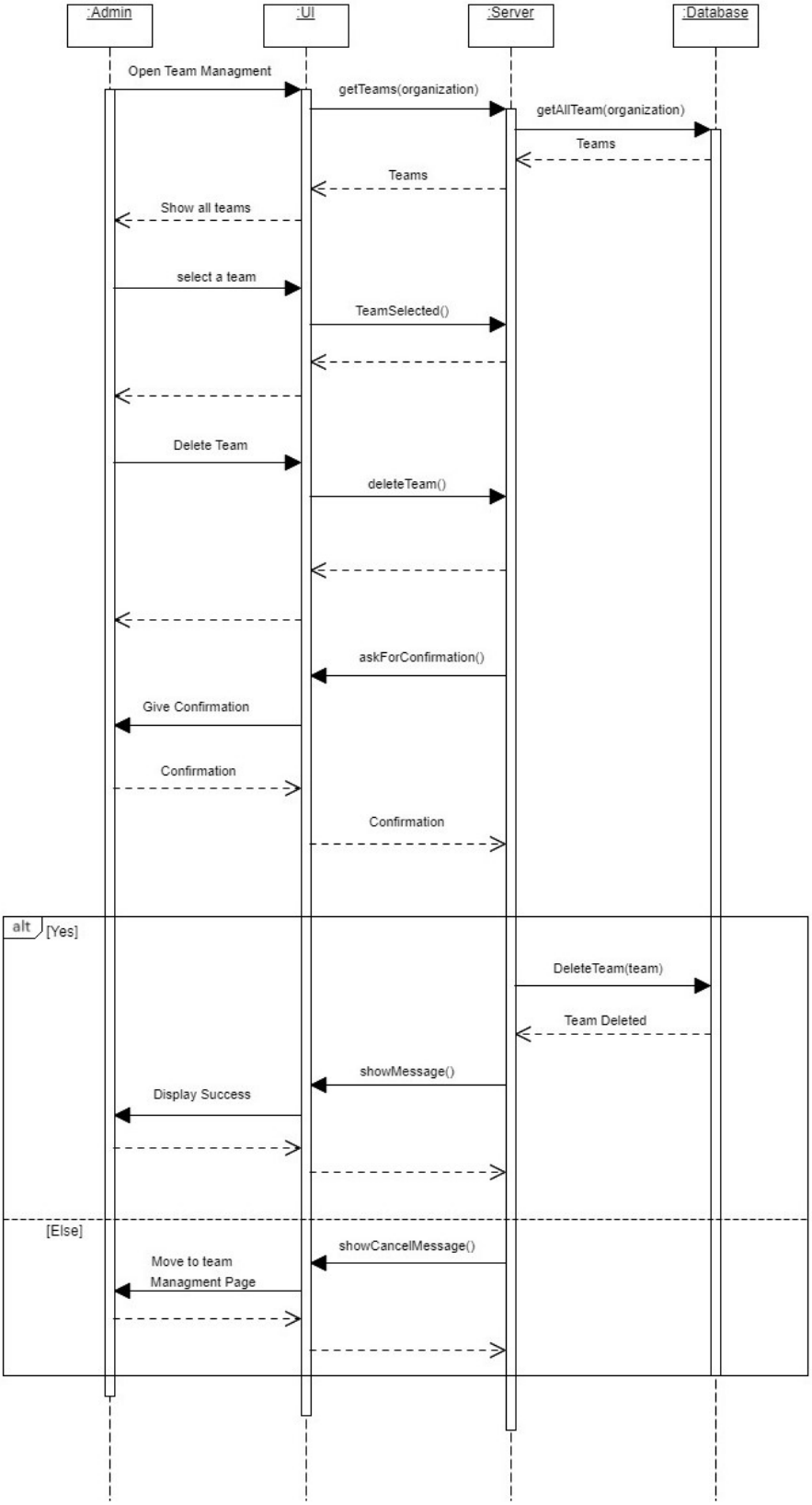

Figure 5.15: Remove a Team

The sequence diagram of remove a team is given above

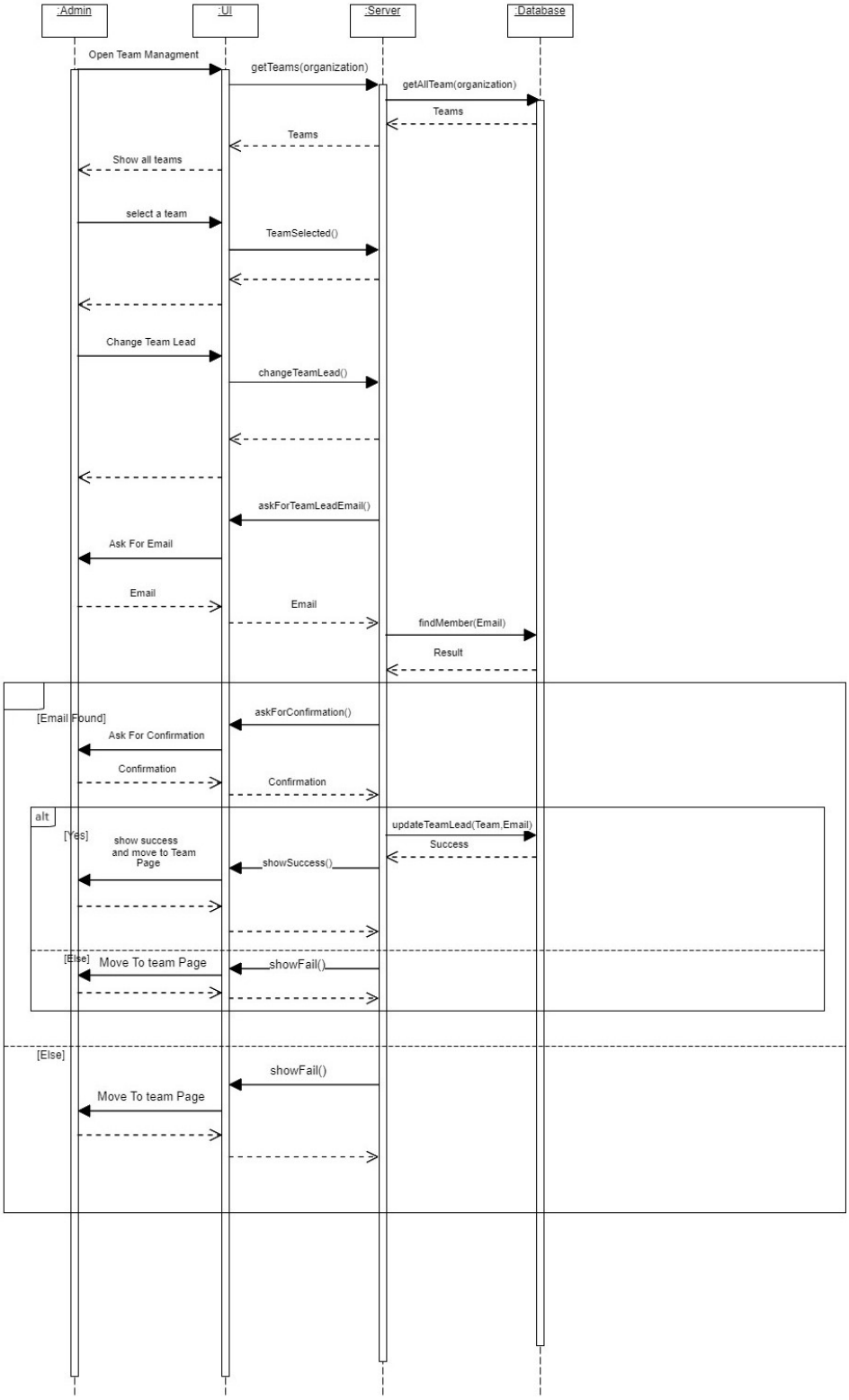

Figure 5.16: Change Team Lead

The sequence diagram of change a team lead is given above

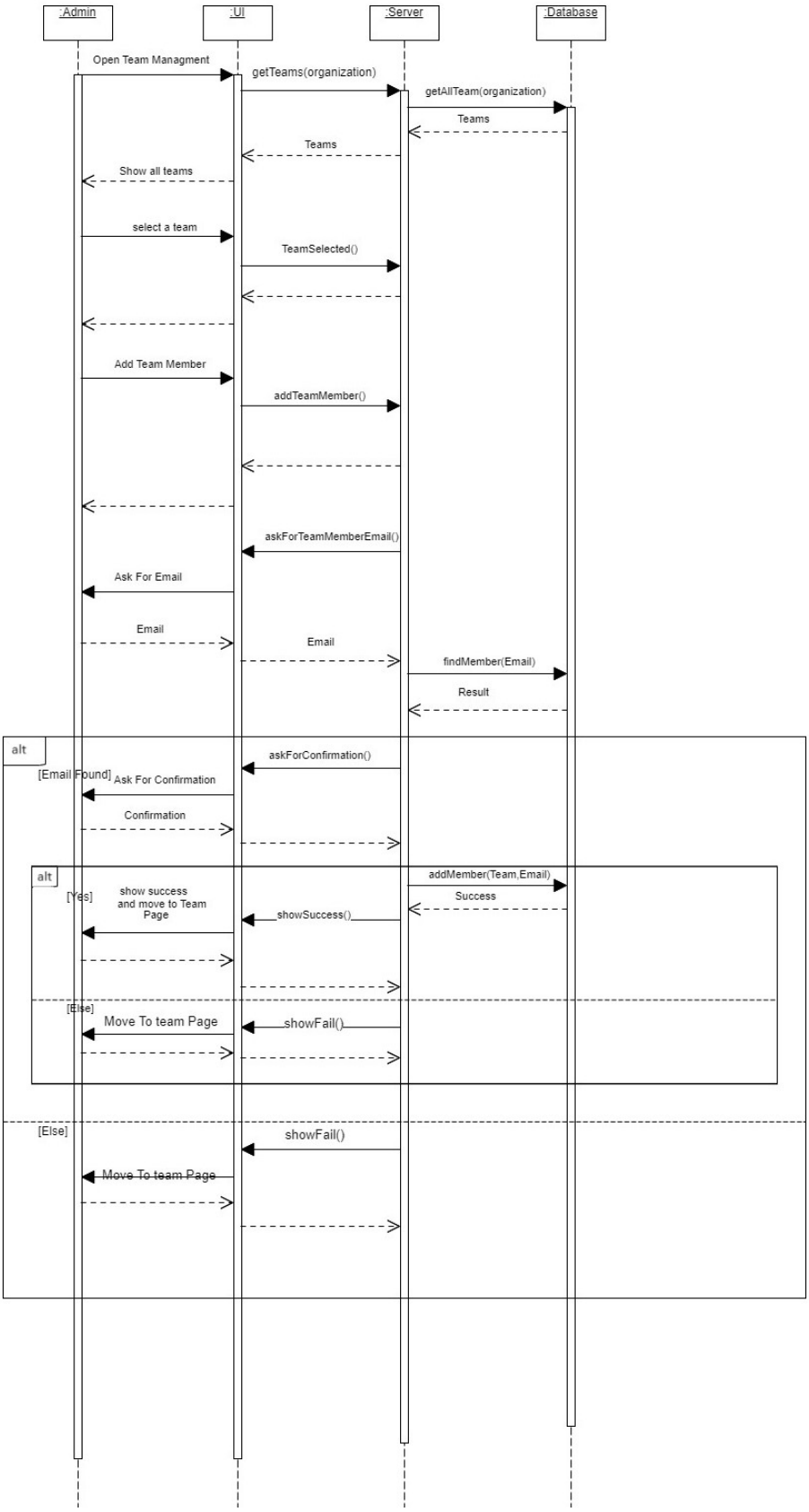

Figure 5.17: Add a Team Member

The sequence diagram of add a team member is given above

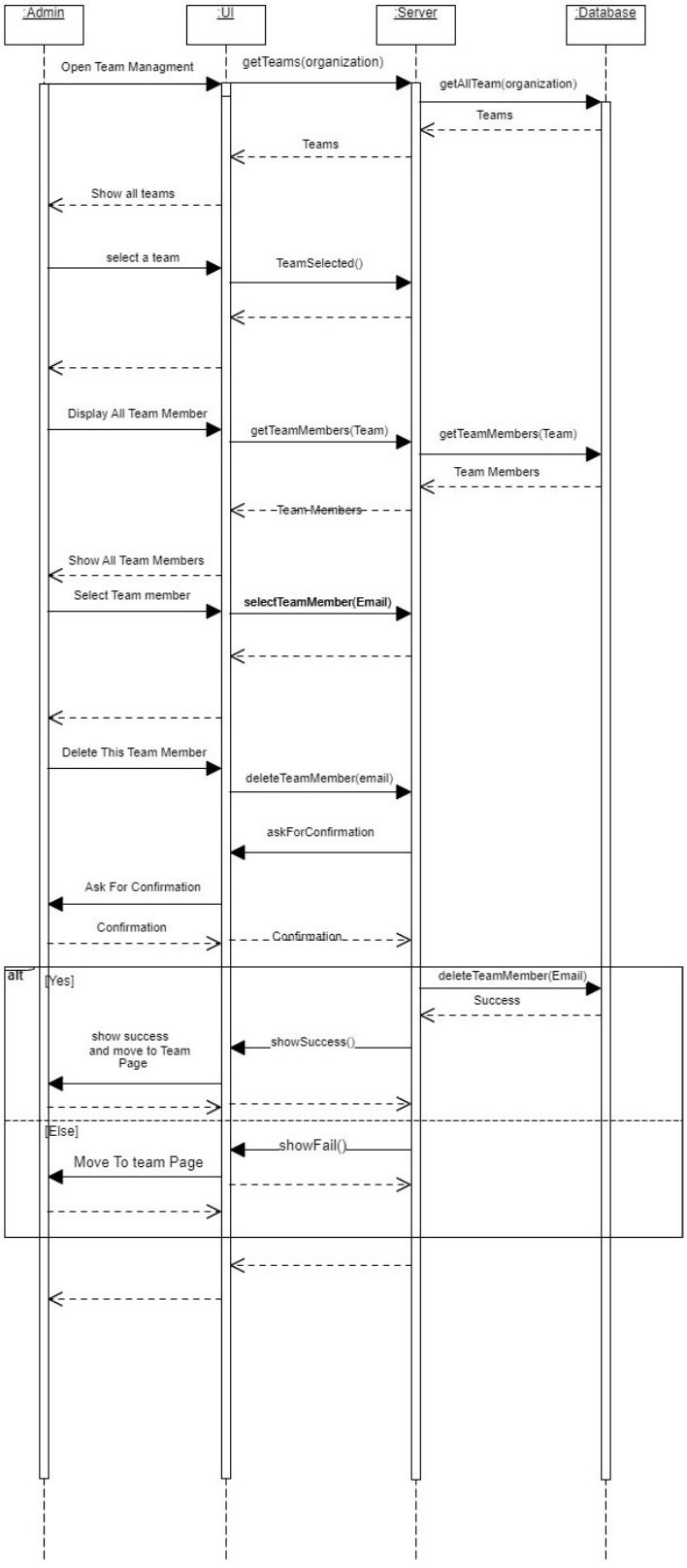

Figure 5.18: Remove Team Member

The sequence diagram of remove a team member is given above

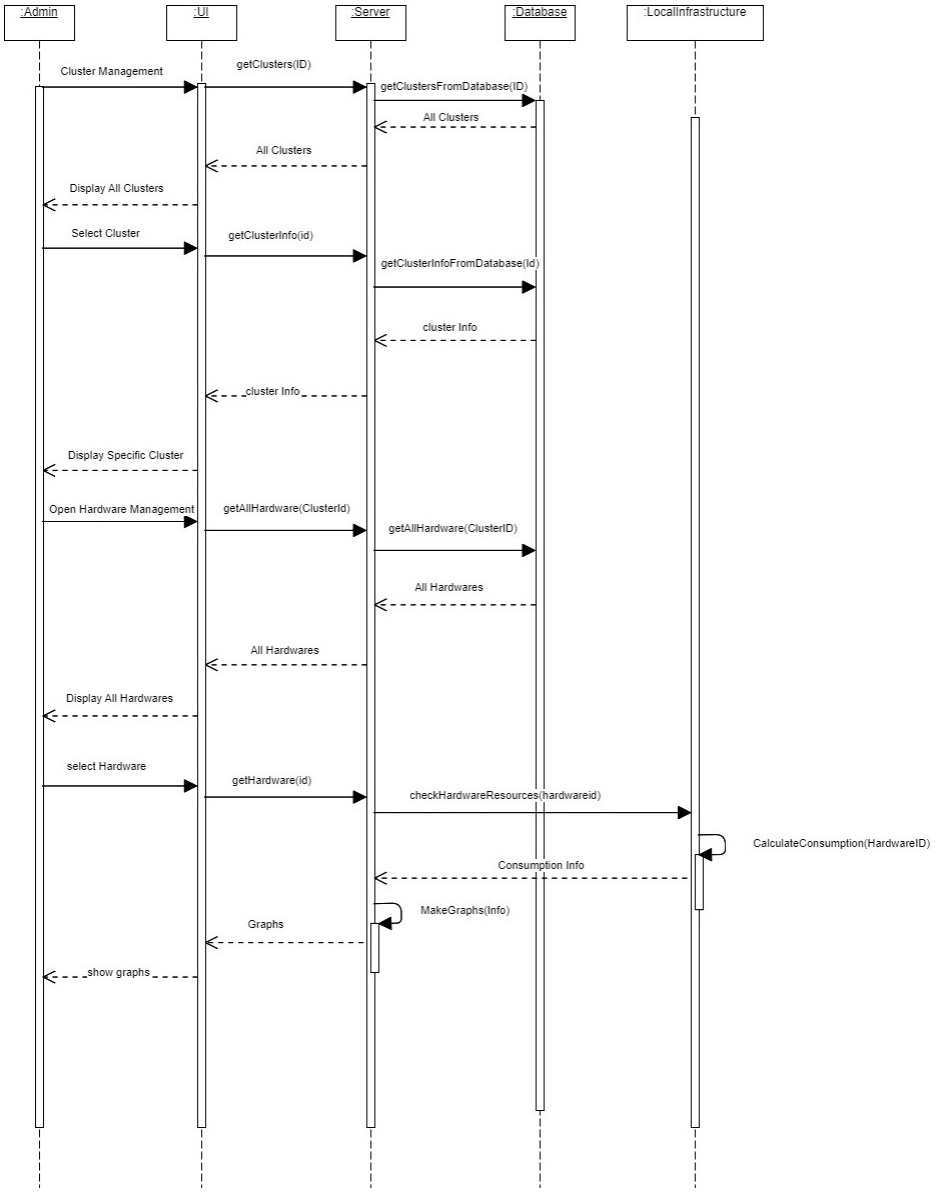

Figure 5.19: Monitor Resource

The sequence diagram of monitor Resource is given above

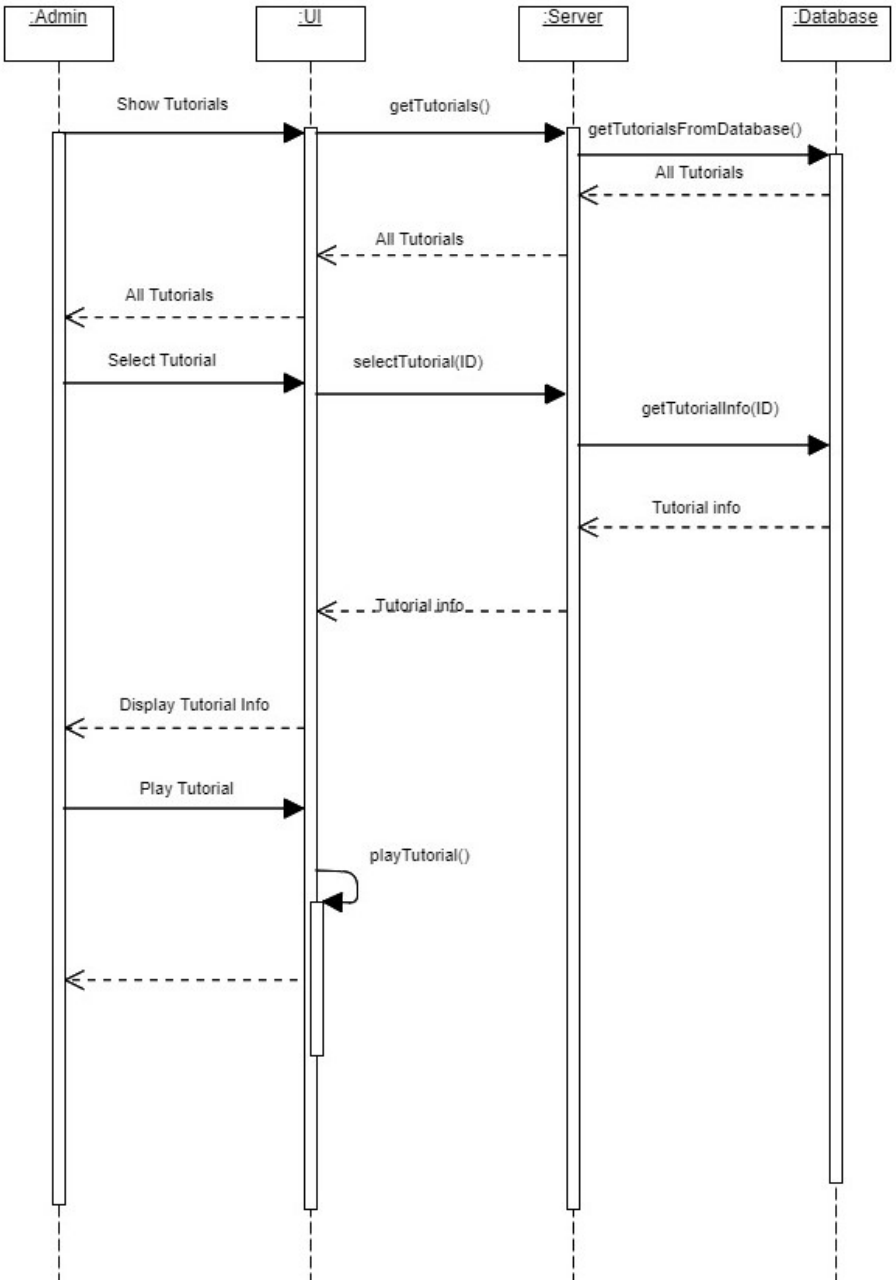

Figure 5.20: View Tutorial

The sequence diagram of view Tutorial is given above

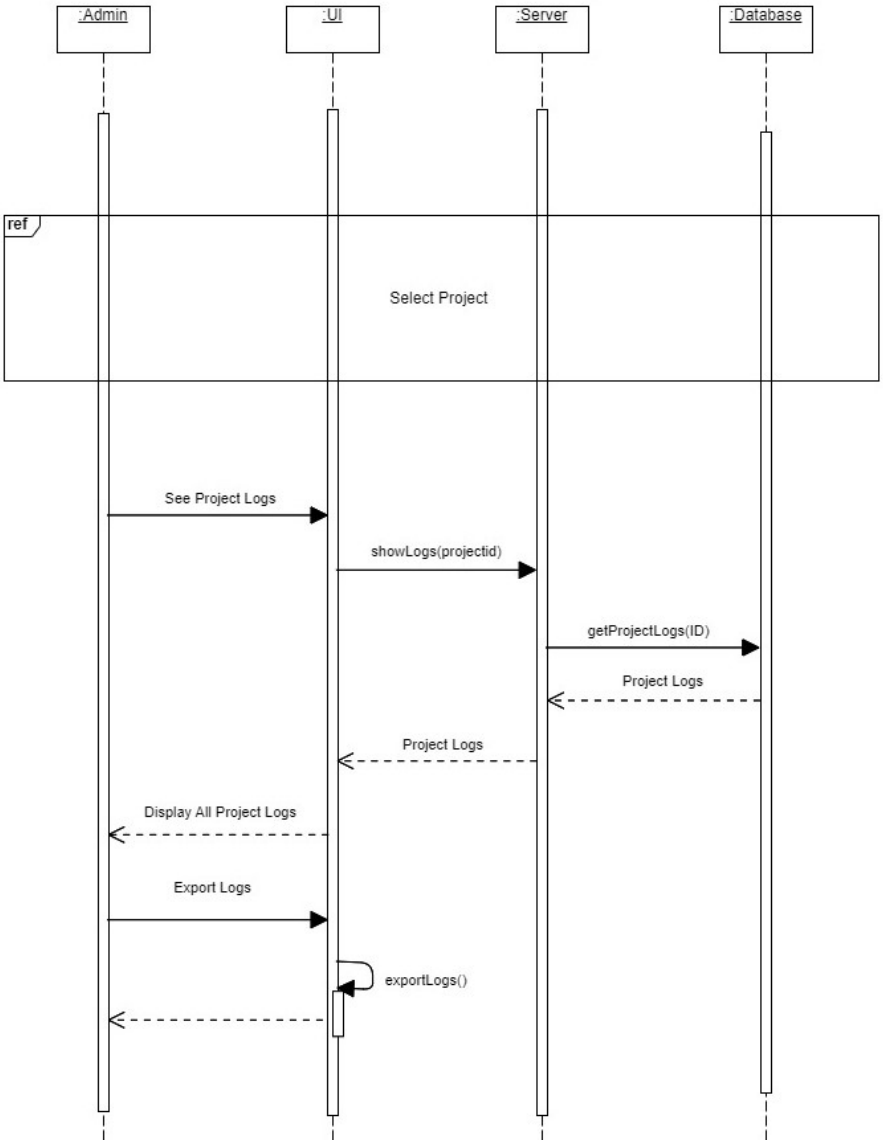

Figure 5.21: View and Export Project Logs

The sequence diagram of view and export project logs is given above

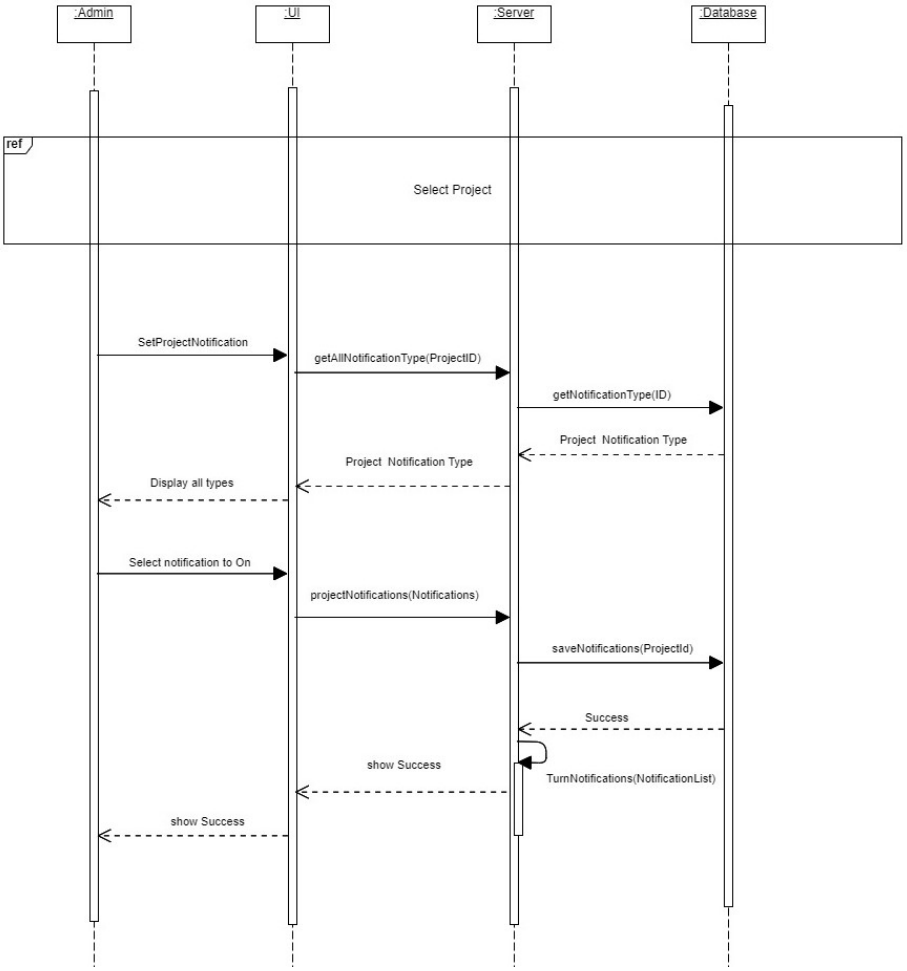

Figure 5.22: Manage Notifications

The sequence diagram of manage notifications is given above

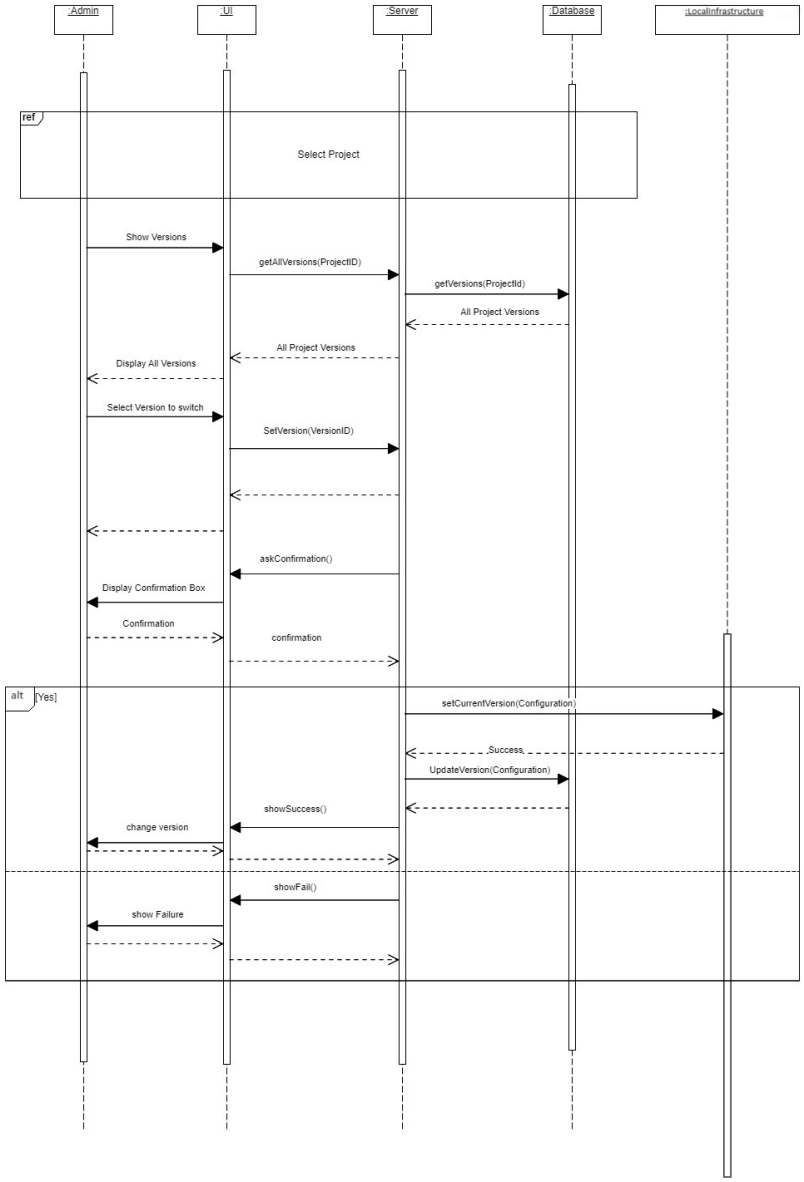

Figure 5.23: Manage Versions

The sequence diagram of manage versions is given above

5.7 Policies and Tactics

This section outlines the various non-architectural policies and tactics that will be employed in the project. These are aimed at refining the interface and implementation details of the system without altering its high-level structure.

5.7.1 Tools and Technologies

In line with our system's diverse functionalities, an array of specialized tools and technologies is employed to optimize each aspect of the project. From cluster automation to AI/ML development and data management, each tool has been carefully selected to enhance the system's efficiency and effectiveness, ensuring robust performance across different environments.

- Ansible & Vagrant: For automating GPU server configuration.
- Kubernetes: To create adaptable grids for computing.
- JupyterHub, Kubeflow, MLflow: Integration for AI/ML development.
- LakeFS and Scaladb: For data storage and management.
- Metabase: For business intelligence reporting.

5.7.2 Coding Guidelines and Conventions

As demonstraded in the architecture diagram in Figure 5.1 the system deals with several separate components, i.e. backend, frontend and Kubernetes, therefore the programming languages shall differ depending on the requirements.

- Languages: Primarily Python and Go language for backend and JavaScript for frontend.
- Style Guides: Adherence to PEP 8 standards for Python and Airbnb style guide for JavaScript.
- Version Control: Utilization of Git for source code management, with clear commit message conventions and pull request practices.

5.7.3 User Interface Design

The user interface is a critical component of our system, representing the gateway through which users interact with our services. Emphasizing user-centric design principles, the interface is crafted to be intuitive, adaptable, and aesthetically pleasing, thereby enhancing user experience and system accessibility.

- Web Management Portal: Development of an intuitive and customizable web interface for AI/ML configuration.
- Design Principles: Emphasis on simplicity, modern aesthetics, and usability.

5.7.4 Testing and Quality Assurance

Quality assurance is pivotal in ensuring the reliability and stability of our system. By adopting a comprehensive testing approach that encompasses various testing methodologies, we aim to identify and rectify issues early in the development cycle, thus maintaining high standards of quality and performance.

- Approach: Combination of unit, integration, and system testing using whitebox and blackbox testing techniques.
- Tools: Use of PyTest for Python and Jest for JavaScript.
- Continuous Integration/Deployment: Integration of testing within the CI/CD pipeline for automated quality checks using Github Actions.

These policies and strategies are essential for the system's development and success, while also ensuring that it adheres to industry standards and best practices.

5.8 Conclusion

Lastly, this section focused on the system architecture, a sophisticated and comprehensive framework that was created for the development of AI/ML using high-performance computation. The four primary components of this design are the Frontend, Backend, Kubernetes, and Helm Charts/Operators. The efficiency, scalability, and adaptability of each component are essential, and they are particularly beneficial for SMEs that operate intricate AI/ML systems. A dynamic user interface, robust backend processing, and intelligent MLOps/DevOps processes are integrated into the design within a Kubernetes context. The integrated system is illustrated in the comprehensive diagram, which underscores the system's efficacy and seamless interaction. Furthermore, the chapter discusses numerous architectural techniques, domain models, sequence diagrams, and policies and tactics relating to system development, coding rules, user interface design, and testing and quality assurance.

Chapter 6 Implementation and Test Cases

This chapter describes the methods and technologies required to build up and deploy a Kubernetes virtual cluster locally using Docker, Ansible, and Vagrant. Installing a standalone Ubuntu operating system is the first step in setting up a local cluster on a personal computer or laptop. Because of its simplicity of use, substantial community support, and interoperability with numerous development tools, Ubuntu is a popular choice. To ensure the system operates efficiently and works as expected, therefore multiple test cases are also developed and executed which are included in next sections.

6.1 Implementation

This section details the practical steps required to implement the Kubernetes cluster by leveraging tools like Docker, Ansible, and Vagrant within a standalone Ubuntu environment, and also implementation details of other use cases discussed in the previous chapters.

6.1.1 Installing Standalone Ubuntu

We can begin by creating a bootable USB drive by downloading the latest Ubuntu ISO from the official website and using a tool like Rufus or Etcher to make the USB stick bootable. Once the installation completes, remove the installation media and reboot the system. After logging in, you must update your system to the most recent packages. Additionally, you must enable virtualization, such as Intel VT or AMD-V, in the BIOS settings. This is crucial for the efficient operation of virtual devices or containers.

6.1.2 Installing Prerequisites

Before deploying the cluster locally, a robust foundation comprising several critical tools is required. The cluster's seamless setup and operation are guaranteed by the distinctive roles of each tool. This section delineates the prerequisites for the deployment of a Kubernetes cluster, with a particular emphasis on Vagrant, Ansible, VirtualBox, and Docker:

- Vagrant: is an indispensable utility for automating the creation and provisioning of virtual machines. In this phase, Vagrant will rapidly establish and configure the virtual machines that will serve as the Kubernetes nodes. This encompasses the configuration of the network, the allocation of resources such as CPU and memory, and the preparation of the environment for Kubernetes installation.
- Ansible: is a sophisticated automation utility that is used for intra-service orchestration, application deployment, and cloud provisioning. In this instance, Ansible is essential for automating

cluster configuration and administration. This involves automating Kubernetes component installation, defining network settings, and maintaining consistency across all cluster nodes.

- VirtualBox: is a freely available, open-source hosted hypervisor designed for x86 virtualizing. It enables users to run different operating systems on the same computer, which makes it excellent for testing and development. In our project, VirtualBox provides the virtualization layer where the VMs, managed by Vagrant, will run.

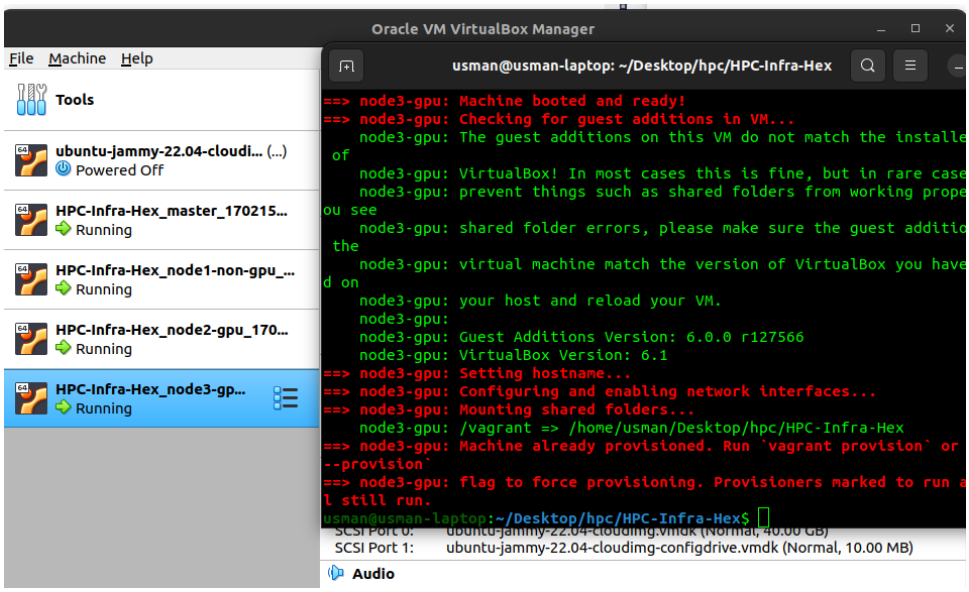

Figure 6.1: Setting up Clusters through Vagrant

The screenshot depicts the automation of installation of VMs through ansible config and vagrant

6.1.3 Installation of Docker

Installation of the docker is the second last phase of our implementation. These are some steps to install docker in each VM ware using ansible and vagrant

- Open your ansible playbook folder and find your main.yml file.
- Write commands to install dockers, you can also specify in which machine you want to install docker.
- After editing your main.yml, you need to restart the provision.
- You can use command Vagrant Provision to start installation of the docker in each of the vm ware.
- After successfull installation you can check if the docker is installed in each vm ware successfully.

The docker will now be installed in each vmware successfully

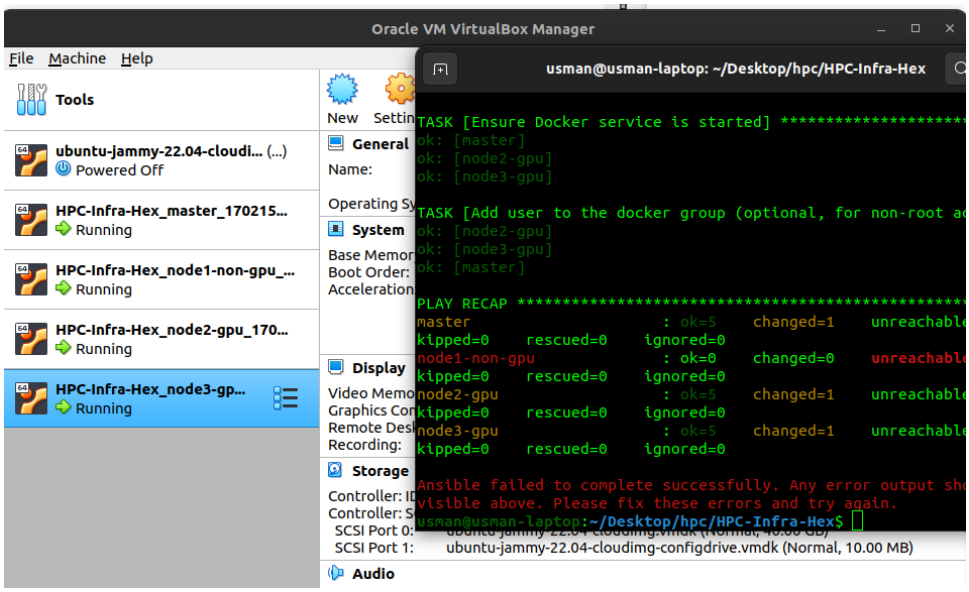

Figure 6.2: Installing Docker using Ansible

The screenshot depicts the installation of docker in the VMs using Ansible Notebook

6.1.4 Extending Ansible to Kubernetes Controller and Worker Nodes

As part of setting up a Kubernetes environment, extending Ansible to manage Kubernetes controller and worker nodes is a critical step. This approach streamlines the configuration and deployment of the Kubernetes cluster, ensuring a consistent and automated setup. This section will guide you through integrating Ansible with Kubernetes, focusing on defining groups, creating roles, and executing playbooks for the successful setup of controller and worker nodes.

- Define groups for controller and worker nodes, specifying IP addresses or hostnames.
- Create Ansible roles for Kubernetes components to organize the playbook.
- Update `main.yml` playbook for Kubernetes components.
- Install control plane components (kubeadm, kubelet, kubectl) on controller nodes.
- Install worker components (kubelet) on worker nodes.
- Use `kubeadm init` on a controller node to initialize the Kubernetes control plane.
- Retrieve join token from `kubeadm init` and use it to join worker nodes.
- Copy generated `kubeconfig` file to your local machine for Kubernetes configuration.
- Add tasks to verify successful Kubernetes installation and configuration.
- Update documentation to reflect changes for Kubernetes deployment.
- Execute playbook using `ansible-playbook` command.

6.1.5 Assigning Cluster Roles

The project aims to assign different roles to different users which will deal with multiple clusters and thus will have ”clusterrolebindings”. In Kubernetes, cluster roles are used to define a set of permissions for accessing and interacting with various resources within the cluster. These roles can be assigned to users, groups, or service accounts. Moreover, for dealing with security issues that which user should do which command and have what permissions, cluster roles are important. Like in our use cases, we had root user, admin users, team leads, we need RBAC permissions for this which is basically assigning multiple roles to different clusters and namespaces within them.

6.1.6 Installation and Configuration of Tools

One of the primary objectives of this project encompasses the automation of tool installation and configuration processes. To facilitate the seamless installation of tools, we utilize Helm charts, a widely recognized package manager for Kubernetes. This approach streamlines the deployment and management of applications within our Kubernetes infrastructure. To further enhance our application’s usability, we have introduced a dedicated API endpoint, ‘install-tool/¡toolname¿‘, designed to expedite the installation of various tools. Below, we detail the process and specifics of the tools currently supported by our application.

6.1.6.1 Prometheus Installation

Prometheus is a freely available toolbox for monitoring and alerting. The system is specifically designed to prioritize dependability, expandability, and ease of use. In order to install Prometheus, with just one click, can be deployed and configured, streamlining the monitoring setup process.

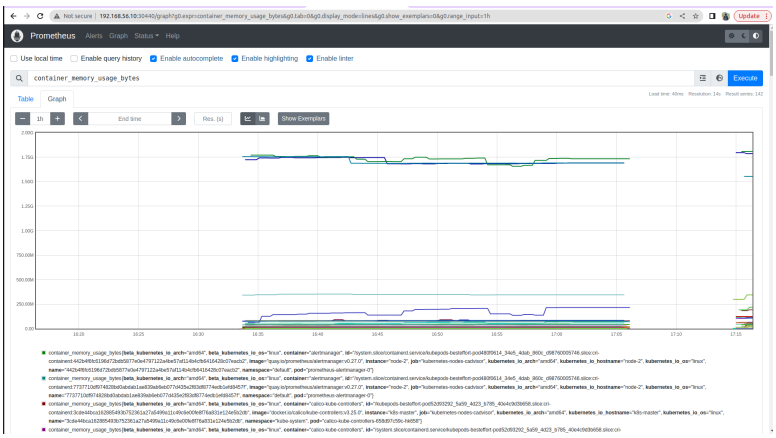

Figure 6.3: Installing Prometheus

The screenshot depicts the dashboard of Prometheus running in the cluster

6.1.6.2 Grafana Installation

Grafana is an open-source platform that provides potent and refined visualizations of metrics data, making it ideal for monitoring and observability. It supports a wide range of data sources, including Prometheus, making it a popular choice for visualizing the data collected by Prometheus in a more accessible and meaningful way. Our app also streamlines and automates the process to install Grafana. However, to access the dashboard, the user has to manually login using the username and password, which are stored in Kubernetes secrets. Retrieve and decrypt this information with the following command:

```
kubectl get secret --namespace default grafana -o yaml
```

The following image shows a demo dashboard which can be used to monitor CPU usage and other information:

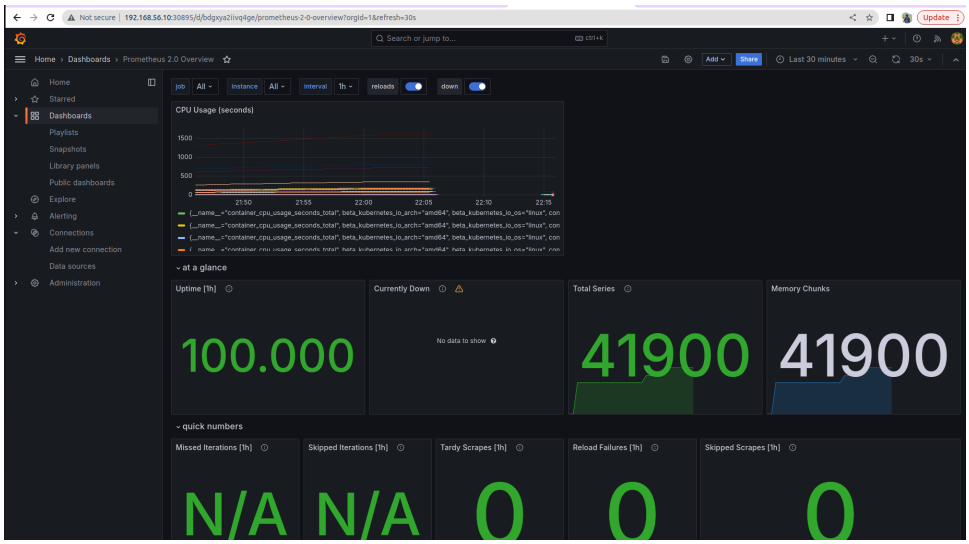

Figure 6.4: Installing Grafana

The screenshot depicts the dashboard of Grafana running in the cluster

6.1.7 Adding Prometheus as a Data Source in Grafana

With access to the Grafana dashboard, Prometheus can now be added as a data source by following steps:

Log into the Grafana dashboard using the admin credentials obtained in the previous step. Navigate to the ”Add data source” section and select Prometheus. Provide the service URL for the Prometheus server (e.g., prometheus-server-ext) that you wish to connect to Grafana. Save the new data source configuration. By following these steps, you have successfully integrated Grafana with Prometheus, enabling you to create detailed and informative dashboards that leverage the comprehensive monitoring

data collected by Prometheus. This integration is crucial for gaining insights into the performance and health of kubernetes infrastructure.

6.1.7.1 JupyterHub Installation

The JupyterHub Helm chart is obtained from a designated repository to commence the installation procedure. This phase is essential because it guarantees that the most recent version of JupyterHub is retrieved and primed for deployment. The Helm chart is installed directly onto a Kubernetes node within our environment after the installation. This installation procedure is essential for the provisioning of JupyterHub, which enables it to function as a web-based, multi-user environment for interactive computation. Additionally, authentication is configured in JupyterHub to ensure that specific users or administrators can preserve their sessions.

6.1.7.2 BinderHub Installation

BinderHub installation commences with the procurement of the BinderHub Helm chart. This chart is indispensable for the configuration of BinderHub, a software that converts Git repositories into interactive Jupyter environments. We proceed to generate a config.yaml file after retrieving the Helm chart. This file contains essential configuration information, such as the Docker repository URL, which BinderHub will utilize to retrieve images. In addition, a secret.yaml file is generated, which will contain the Docker credentials necessary for the deployment of containers within our infrastructure. Once these preparatory steps have been completed, BinderHub is deployed via our application's API, and a Persistent Volume is established to guarantee data persistence across sessions.

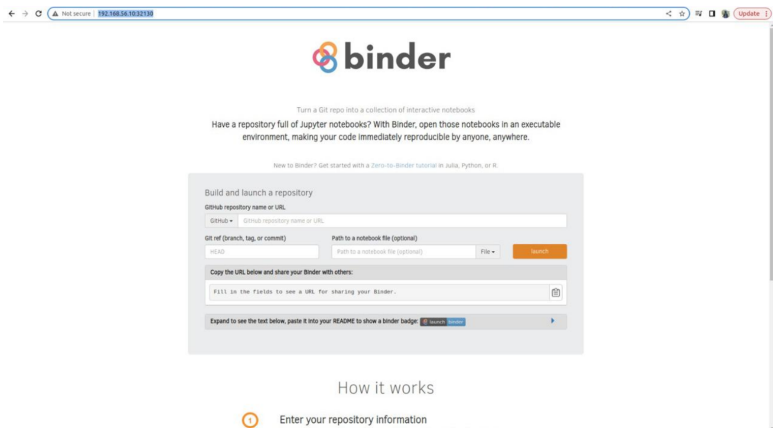

Figure 6.5: Installing BinderHub

The screenshot depicts the installation of BinderHub in the cluster

6.1.7.3 Linking JupyterHub & BinderHub

In order to enhance the integration between JupyterHub and BinderHub within our application, we have implemented a function known as ”bind-binderhub.” This function is indispensable for the establishment of a connection between the two platforms. It accomplishes this by connecting the service port of JupyterHub to the deployed BinderHub instance. This integration allows for the automatic activation of repositories in JupyterHub that contain Jupyter notebooks, thereby allowing for a seamless user experience. This link ensures that users may smoothly switch between creating and engaging with interactive notebooks, hence promoting a streamlined and cohesive workflow inside our program.

6.1.8 Server Deployment

Server deployment is an important part of the development process that involves setting up and installing the backend and user parts of an application on a collection of servers. This process makes sure that the service is launched correctly and that end users can access it. To make the release process run smoothly, management tools, network setup, and containerization are all used. The steps for deploying both the backend and user parts of a server application are broken down below.

6.1.8.1 Backend

The deployment of the backend, specifically a Flask server, on a cluster involves several key steps: This initial step requires creating a Docker container for the Flask application. The commands used are as follows:

- To build the Docker image: `sudo docker build -t <username>/backend:latest .`
- To push the Docker image to a registry: `sudo docker push <username>/backend:latest`

Helm charts are used to manage Kubernetes applications. The Flask application is deployed using the Helm chart with the command:

```
helm install flask-chart ./flask-chart
```

The deployment status can be monitored using Kubernetes commands like `kubectl get pods` or `kubectl get deployments`.

After ensuring the backend pod is in a running state, port-forwarding is configured to allow access to the service. The command used is:

```
kubectl port-forward service/flask-chart 5000:5000
```

6.1.8.2 Frontend

For deploying the frontend, which utilizes NextJS, the steps are similar yet tailored to the frontend specifics:

The NextJS application is containerized by building and pushing its Docker image using the following commands:

- To build the Docker image: `sudo docker build -t <username>/nextjs-cont:latest .`
- To push the Docker image to a registry: `sudo docker push <username>/nextjs-cont:latest`

The NextJS application is deployed on the cluster using its respective Helm chart with:

```
helm install nextjs-chart ./nextjs-chart
```

The deployment's progress can be checked with `kubectl get pods` or `kubectl get deployments`.

Once the frontend pod is operational, port-forwarding is set up to facilitate access to the NextJS application. The command executed is:

```
kubectl port-forward service/nextjs-chart 3000:80
```

Consequently, the application becomes accessible at http://localhost:3000/.

This structured approach to deploying both backend and frontend components ensures that the application is not only deployed efficiently but also remains scalable and manageable throughout its lifecycle.

6.1.9 Authentication

In our project, authentication is a critical component ensuring secure access and data integrity. The implementation of authentication involves several key technologies and practices to safeguard user credentials and maintain a secure application environment.

6.1.9.1 Backend Security

For authentication management, we implemented JSON Web Tokens. JWT is an open standard that defines a self-contained and compact method for securely transmitting information between parties in the form of a JSON object. This information is trustworthy and verifiable as a result of its digital signature. We implemented JWT to assure secure communication between the client and server and to manage user sessions. The operation of our system is as follows:

- **Token Generation**: A JWT is generated upon successful authentication, which includes a payload containing user-specific information and a signature to confirm its authenticity. The client is

subsequently sent this token, which is subsequently stored locally.

- **Token Validation**: The JWT is incorporated into the Authorization payload by the client for subsequent queries. The server validates the token to ensure that it has not expired or been tampered with. This procedure facilitates user authentication without necessitating the transmission of sensitive information.
- **Token Expiry and Refresh**: To enhance security, tokens have an expiration time after which they are no longer valid. We also implemented token refresh mechanisms to issue new tokens before the current ones expire, ensuring continuous access without frequent logins.

Moreover, storing plain-text passwords in the database is a significant security risk. To mitigate this, we used hashing algorithms to store hashed versions of user passwords. This way, even if the database is compromised, the actual passwords remain secure.

6.1.9.2 Frontend Security

On the frontend, as we used Next.js, a React-based framework that provides several security features and practices to enhance the overall security of our application. Due to this choice, Next.js already offers Server-Side Rendering, which helps protect sensitive data by keeping it on the server and only sending the necessary data to the client, thereby reducing the risk of exposing sensitive data through client-side scripts. To further enhance security, we implemented middleware functions in Next.js to protect API routes by checking for valid JWT tokens before processing requests, ensuring that only authenticated users can access protected endpoints. Additionally, to mitigate Cross-Site Request Forgery attacks, we utilized Next.js's built-in support for handling CSRF tokens, protecting our forms and API endpoints. We also configured Content Security Policy headers to prevent various types of attacks such as Cross-Site Scripting and data injection attacks, specifying which sources of content are trusted to mitigate these risks.

6.1.10 Cluster Queue Mechanism

Due to limited storage space, the creation of an on-premises cluster also needs a strong queue system to handle the distribution of permanent volume claims. This cluster has a front-end site that lets people set up and use different tools. Because storage and PVCs are limited, it's important to set up a system that controls user access based on the availability of resources.

6.1.10.1 Implementation of Queue Mechanism

The first step in the application is to make Persistent Volumes that change based on how many people are connected. When a user asks for a tool to be deployed, the current supply of PVs is checked. If a PV is available, a PVC is attached to the user's distribution right away, giving them access to the tool they asked for. The amount of PVs is controlled so that the storage is used most efficiently and all users can get to the resources they need. A database stores information about each PV so that the state and access of them can be managed well. This database has information like PV numbers, their current state (whether they are open or in use), and user information that goes with them. By keeping this information up to date, the system can quickly check on the state of each PV, making sure that reports on resource sharing are correct and up to date in real time. When all of the PVCs are taken, the system instantly adds new requests to a list. The queue management system keeps track of ongoing requests in the order they were received so that they can be handled quickly by users. People waiting in line are let in as soon as a PVC opens up. The system keeps an eye on the state of each PVC at all times, changing the list and letting users know when the resources they've been waiting for become available.

To improve freedom and management control, the system has features that let managers control the line. Administrators can see what users are waiting in line and change the order of the users if needed to give them priority. This feature gives administrators more freedom in situations where certain users need entry right away. Administrators can move a person ahead in the queue so that they can get in right away when a PVC opens up.

The whole process is illustrated in the figure below:

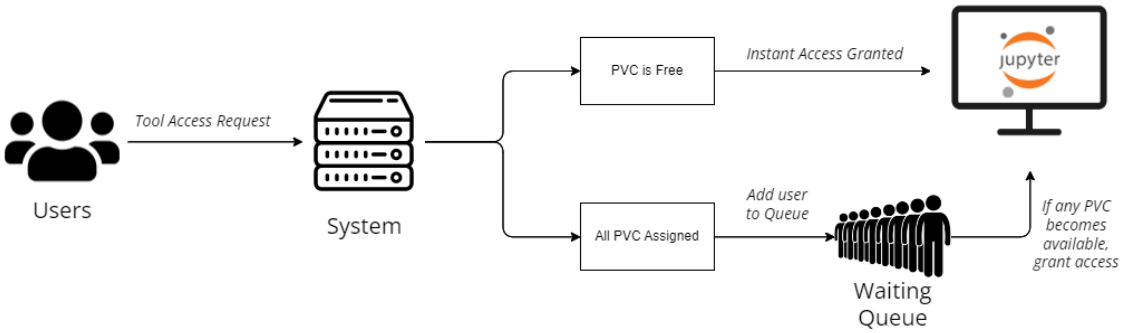

Figure 6.6: Queue Mechanism Process Flow

The diagram illustrates the process flow for managing user requests, checking for PV availability, adding users to the queue

The on-premises cluster makes sure that everyone has fair and efficient access to limited storage resources by incorporating the queue system. Users can quickly set up and use the tools they want, and

the queue management system makes sure that things stay in order and resources are distributed, even when resources are limited.

6.2 Test Case Design and Description

This section provides a comprehensive design and description of the test cases implemented for the system. The aim of these test cases is to systematically assess the functionality, reliability, and performance of the developed system under various scenarios. The standard template is followed throughout all the test cases, outlining the criteria and followed by a detailed description of each test scenario, including inputs, expected outcomes, and the rationale behind their selection. This organized method makes sure that all important parts of the system are carefully checked, which makes it easier to find any problems and confirm that the system works well as a whole.

6.2.1 Functional Test cases

The following are the functional test cases of the system:

6.2.1.1 Login

Table 6.1: Login

Test case to check if the user is able to login to the system

Login			
Test Case ID:	*1*	**QA Test Engineer:**	*Fatima Siddiqui*
Test case Version:	*1*	**Reviewed By:**	*Hassan Rehman*
Test Date:	*25-04-2024*	**Use Case Reference(s):**	*Login*
Revision History:	*None*		
Objective:	*To check if user is able to successfully login to the system*		
Product/Ver/ Module:	*Login module of system*		
Environment:	*Both web and backend server is deployed inside master node and running successfully.*		
Assumptions:	*Login screen and button available for login to the user.*		
Pre-Requisite:	*The user is already stored in the database and registered as a valid user*		
Step No.	**Execution description**	**Procedure result**	
1	*User enters their credentials, i.e., their username and password, and clicks login.*	*System validates the credentials and takes the user to the home screen.*	
Comments: The test case is passed. The user is successfully logged in if provided correct credentials.			
✓ **Passed** **Failed** **Not Executed**			

6.2.1.2 Run Command

Table 6.2: Run Command

Test case to check if the user is able to run command

Run Command			
Test Case ID:	*2*	**QA Test Engineer:**	*Fatima Siddiqui*
Test Case Version:	*1*	**Reviewed By:**	*Usman Faisal*
Test Date:	*26-04-2024*	**Use Case Reference(s):**	Run Command
Revision History:	*None*		
Objective:	*To check if user is able to execute command and fetch data instead through terminal.*		
Product/Ver/Module:	*Cluster Module*		
Environment:	*Both web and backend server is deployed inside master node and running successfully.*		
Assumptions:	*The text box is available to write the command and output box is visible to the user.*		
Pre-Requisite:	*The logged in user has access to the run command.*		
Step No.	**Execution description**		**Procedure result**
1	*User types the command in the text box instead on terminal.*		*System fetches the response and displays that on the output box.*
Comments: The test case is passed. The user is able to run a command inside the node.			
✓ **Passed** **Failed** **Not Executed**			

6.2.1.3 Remove Cluster Resources

Table 6.3: Remove Cluster Resources

Test case to check if the user is able to remove resources from the cluster

Remove Cluster Resources			
Test Case ID:	*3*	**QA Test Engineer:**	*Fatima Siddiqui*
Test Case Version:	*1*	**Reviewed By:**	*Hassan Rehman*
Test Date:	*29-04-2024*	**Use Case Reference(s):**	Remove Cluster Resources
Revision History:	*None*		
Objective:	*To check if the user can remove resources and customize resources within the cluster.*		
Product/Ver/Module:	*Remove Cluster Resources Module*		
Environment:	*Both web and backend server is deployed inside master node and running successfully.*		
Assumptions:	*Cluster management screen and button is visible to the user.*		
Pre-Requisite:	*User is logged on and has the necessary permissions for cluster customization.*		
Step No.	**Execution description**		**Procedure result**
1	*The user selects the cluster which he wants to customize and checks for resources to remove.*		*The system verifies that resource within the cluster and detaches it.*
2	*The user confirms the detaching of that resource from the cluster and clicks to remove the resource from the cluster.*		*The system updates data and removes the resource from the cluster.*
Comments: The test case is passed. The resources were modified in the cluster.			
✓ **Passed** **Failed** **Not Executed**			

6.2.1.4 Installation of Tools

Table 6.4: Installation of Tools

Test case to check if the user is able to install the tools in infrastructure

Installation of Tools			
Test Case ID:	*4*	**QA Test Engineer:**	*Hassan Rehman*
Test Case Version:	*1*	**Reviewed By:**	*Usman Faisal*
Test Date:	*29-04-2024*	**Use Case Reference(s):**	Drag and Drop Tools to the Projects
Revision History:	*None*		
Objective:	*To check if User is able to install tool*		
Product/Ver/Module:	*Tool Installation Module*		
Environment:	*Set up has been done on local machine and running successfully.*		
Assumptions:	*The user is logged in their account.*		
Pre-Requisite:	*The user is a valid user and has admin access.*		
Step No.	**Execution description**		**Procedure result**
1	*User clicks on install tool*		*System shows drag & drop page*
2	*User drag the tools into space*		*System shows tools in the space*
3	*User click on deploy tool*		*System shows success message*
Comments: The test case is passed. New tool is installed in the infrastructure.			
✓ **Passed** **Failed** **Not Executed**			

6.2.1.5 Uninstallation of Tools

Table 6.5: Uninstallation of Tools

Test case to check if the user is able to uninstall the tools from infrastructure

Uninstallation of Tools			
Test Case ID:	*7*	**QA Test Engineer:**	*Hassan Rehman*
Test Case Version:	*1*	**Reviewed By:**	*Usman Faisal*
Test Date:	*29-04-2024*	**Use Case Reference(s):**	Uninstall Tool
Revision History:	*None*		
Objective:	*To check if User is able to uninstall tool or not*		
Product/Ver/Module:	*Tool Uninstallation Module*		
Environment:	*Set up has been done on local machine and running successfully.*		
Assumptions:	*The user is logged in their account.*		
Pre-Requisite:	*The user is already stored in the database and registered as a valid user and has admin access. The tool is already installed in the infrastructure*		
Step No.	**Execution description**		**Procedure result**
1	*User clicks on install tool from navbar*		*System shows drag and drop page*
2	*User drags the tool from the space to the inventory*		*System shows the tool dragged into the inventory*
3	*User clicks on uninstall tool*		*System shows message: "Tool is uninstalled successfully"*
Comments: The test case is passed. The tool is uninstalled from infrastructure			
✓ **Passed** **Failed** **Not Executed**			

6.2.1.6 Create a New Organization

Table 6.6: Create a New Organization

Test case to check if the system allows root user to create a new organization

Create Organization			
Test Case ID:	*5*	**QA Test Engineer:**	*Usman Faisal*
Test Case Version:	*1*	**Reviewed By:**	*Hassan Rehman*
Test Date:	*26-04-2024*	**Use Case Reference(s):**	Register as the Root User for an Organization
Revision History:	*None*		
Objective:	*To check if root user is able to create a new organization*		
Product/Ver/Module:	*Organization Module*		
Environment:	*Both web and backend server is deployed inside master node and running successfully.*		
Assumptions:	*The root user is logged in their account.*		
Pre-Requisite:	*The logged in user has root access and there is no organization currently exsiting in the system.*		
Step No.	**Execution description**		**Procedure result**
1	*User clicks on create organization*		*System opens up a new modal screen which has input fields.*
2	*User enters their organization information, i.e., name, type, size and clicks create.*		*System initializes a new organization and takes user to management users page.*
Comments: The test case is passed. A new organization is created.			
✓ **Passed** **Failed** **Not Executed**			

6.2.1.7 Add Admin to Organization

Table 6.7: Add Admin to Organization

Test case to check if the system allows root user to add a new admin to organization

Add Admin			
Test Case ID:	*6*	**QA Test Engineer:**	*Usman Faisal*
Test Case Version:	*1*	**Reviewed By:**	*Fatima Siddiqui*
Test Date:	*26-04-2024*	**Use Case Reference(s):**	Add Admin to the Organization
Revision History:	*None*		
Objective:	*To check if root user is able to add a admin to organization*		
Product/Ver/Module:	*Organization Module*		
Environment:	*Both web and backend server is deployed inside master node and running successfully.*		
Assumptions:	*The logged in user has root access.*		
Pre-Requisite:	*The logged in user has root access and has an existing organization.*		
Step No.	**Execution description**		**Procedure result**
1	*User clicks on add a user on user management page.*		*System opens a modal box which has input fields.*
2	*User enters information e.g full name, email, selects user type as admin*		*System creates the new user successfully and redirects to user management page.*
Comments: The test case is passed. A new admin user is created in the database.			
✓ **Passed** **Failed** **Not Executed**			

6.2.1.8 Delete Admin in Organization

Table 6.8: Delete Admin in Organization

Test case to check if the system allows root user to delete an admin in organization

Delete Admin			
Test Case ID:	*8*	**QA Test Engineer:**	*Usman Faisal*
Test Case Version:	*1*	**Reviewed By:**	*Hassan Rehman*
Test Date:	*26-04-2024*	**Use Case Reference(s):**	Remove Admin from the Organization
Revision History:	*None*		
Objective:	*To check if root user is able to delete a admin in organization*		
Product/Ver/Module:	*Organization Module*		
Environment:	*Both web and backend server is deployed inside master node and running successfully.*		
Assumptions:	*The logged in user has root access.*		
Pre-Requisite:	*The logged in user has root access and has an existing organization.*		
Step No.	**Execution description**		**Procedure result**
1	*User clicks on edit button on the admin they want to delete on user management page.*		*System opens a modal box which has details of admin and a delete button.*
2	*User clicks the delete button*		*System prompts user for a confirmation message.*
3	*User clicks the Yes button*		*System successfully removes the admin, updates the list and redirects to user management page.*
Comments: The test case is passed. The admin user is deleted in the database.			
✓ **Passed** **Failed** **Not Executed**			

6.2.1.9 Resource Monitoring

Table 6.9: Resource Monitoring

Test case to check if the user with admin access can monitor resources

Resource Monitoring			
Test Case ID:	*9*	**QA Test Engineer:**	*Hassan Rehman*
Test Case Version:	*1*	**Reviewed By:**	*Usman Faisal*
Test Date:	*29-04-2024*	**Use Case Reference(s):**	Resource Monitoring
Revision History:	*None*		
Objective:	*To check if User with admin access can monitor resources*		
Product/Ver/Module:	*Resource Monitoring Module*		
Environment:	*Set up has been done on local machine and running successfully.*		
Assumptions:	*The user is logged in their account with admin access.*		
Pre-Requisite:	*The user is already stored in the database and registered as a valid user with admin access. Grafana monitoring tool is already installed in the tool management*		
Step No.	**Execution description**		**Procedure result**
1	*User clicks on cluster settings*		*System opens cluster settings page*
2	*User clicks on monitor resource*		*System redirects to Grafana tool*
Comments: The test case is passed. The System opened the grafana tool.			
✓ **Passed** **Failed** **Not Executed**			

6.2.1.10 Add User to Organization

Table 6.10: Add User to Organization

Test case to check if the system allows root user to add a new user to organization

Add Admin			
Test Case ID:	*10*	**QA Test Engineer:**	*Usman Faisal*
Test Case Version:	*1*	**Reviewed By:**	*Fatima Siddiqui*
Test Date:	*26-04-2024*	**Use Case Reference(s):**	Add User to the Organization
Revision History:	*None*		
Objective:	*To check if root or admin user is able to add a new user to organization*		
Product/Ver/Module:	*Organization Module*		
Environment:	*Both web and backend server is deployed inside master node and running successfully.*		
Assumptions:	*The logged in user has root or admin access.*		
Pre-Requisite:	*The logged in user has root access and has an existing organization.*		
Step No.	**Execution description**		**Procedure result**
1	*User clicks on add a user on user management page.*		*System opens a modal box which has input fields.*
2	*User enters information e.g full name, email, selects user type as user*		*System creates the new user successfully and redirects to user management page.*
Comments: The test case is passed. A new user is created in the database.			
✓ **Passed** **Failed** **Not Executed**			

6.2.1.11 View Tool Tutorial

Table 6.11: View Tool Tutorial

Test case to check if the system allows the user to view tool deployment tutorial

View Tool Tutorial			
Test Case ID:	*11*	**QA Test Engineer:**	*Usman Faisal*
Test Case Version:	*1*	**Reviewed By:**	*Hassan Rehman*
Test Date:	*27-04-2024*	**Use Case Reference(s):**	View Tutorials
Revision History:	*None*		
Objective:	*To check if the user is able to view tutorial of tool deployment*		
Product/Ver/Module:	*Tool Installation Module*		
Environment:	*Both web and backend server is deployed inside master node and running successfully.*		
Assumptions:	*At least one cluster exists.*		
Pre-Requisite:	*The user is logged in with a valid account and is on tool deployment page.*		
Step No.	**Execution description**		**Procedure result**
1	*User clicks on "i" information icon on the tool installation page.*		*System opens a modal box which has picture and steps in order to deploy the tool.*
2	*User clicks the next button*		*System opens the next page in the tutorial.*
3	*User clicks the close button*		*System successfully closes the modal box.*
Comments: The test case is passed. The user is able to view tutorial of tool deployment.			
✓ **Passed** **Failed** **Not Executed**			

6.2.1.12 Setting Queue Limit to Tool

Table 6.12: Setting Queue Limit to Tool

Test case to check if the user is able to set queue to a tool

Setting Queue Limit to Tool			
Test Case ID:	*12*	**QA Test Engineer:**	*Hassan Rehman*
Test Case Version:	*1*	**Reviewed By:**	*Usman Faisal*
Test Date:	*29-04-2024*	**Use Case Reference(s):**	Adjusting Tool Queue Limi
Revision History:	*None*		
Objective:	*To set queue limit to a tool*		
Product/Ver/Module:	*Tool Settings Module*		
Environment:	*Set up has been done on local machine and running successfully.*		
Assumptions:	*The user is logged in their account with admin access.*		
Pre-Requisite:	*The user is already stored in the database and registered as a valid user with admin access, and the tool is already installed in the infrastructure*		
Step No.	**Execution description**		**Procedure result**
1	*User goes to cluster settings*		*System opens cluster settings page*
2	*User clicks on tool settings*		*System shows all the tools*
3	*User selects the tool*		*System opens the tool settings*
4	*User enters value and click on set queue limit*		*System shows success message*
Comments: The test case is passed. The new Queue limit is set.			
✓ **Passed** **Failed** **Not Executed**			

6.2.1.13 Adding User into Queue to Tool

Table 6.13: Adding User into Queue to Tool

Test case to check if the user is able to set queue limit to a tool

Adding User into Queue to Tool			
Test Case ID:	*13*	**QA Test Engineer:**	*Hassan Rehman*
Test Case Version:	*1*	**Reviewed By:**	*Usman Faisal*
Test Date:	*29-04-2024*	**Use Case Reference(s):**	Utilizing Infrastructure Tool Queue
Revision History:	*None*		
Objective:	*To add a user into the queue of a tool*		
Product/Ver/Module:	*Tool Queue Module*		
Environment:	*Set up has been done on local machine and running successfully.*		
Assumptions:	*The user is logged in their account with appropriate access.*		
Pre-Requisite:	*The tool is installed and accessible to the user.*		
Step No.	**Execution description**		**Procedure result**
1	*User goes to the infrastructure*		*System displays infrastructure details*
2	*User selects the desired tool*		*System opens the tool's details*
3	*User clicks on "Get in Queue"*		*System adds the user into the tool's queue*
Comments: The test case is passed. The user has been successfully added into the tool's queue.			
✓ **Passed** **Failed** **Not Executed**			

6.2.1.15 Removing User from Queue of Tool

Table 6.15: Removing User from Queue of Tool

Test case to check if the user with admin access is able to remove users from queue of tool

Removing User from Queue of Tool			
Test Case ID:	*15*	**QA Test Engineer:**	*Hassan Rehman*
Test Case Version:	*1*	**Reviewed By:**	*Usman Faisal*
Test Date:	*29-04-2024*	**Use Case Reference(s):**	Removing User from Tool's Queue
Revision History:	*None*		
Objective:	*To remove a user from the queue of a tool*		
Product/Ver/Module:	*Tool Queue Module*		
Environment:	*Set up has been done on local machine and running successfully.*		
Assumptions:	*The user is logged in their account with admin access.*		
Pre-Requisite:	*The tool is installed and accessible to the user, and the user is already in the queue of the tool.*		
Step No.	**Execution description**		**Procedure result**
1	*Admin goes to the infrastructure*		*System displays infrastructure details*
2	*Admin selects the desired tool*		*System opens the tool's details*
3	*Admin clicks on "Remove from Queue"*		*System removes the user from the tool's queue*
Comments: The test case is passed. The user has been successfully removed from the tool's queue.			
✓ **Passed** **Failed** **Not Executed**			

6.2.1.14 Delete User from Organization

Table 6.14: Delete User from Organization

Test case to check if the system allows root user to delete an admin in organization

Delete Admin			
Test Case ID:	*16*	**QA Test Engineer:**	*Usman Faisal*
Test Case Version:	*1*	**Reviewed By:**	*Hassan Rehman*
Test Date:	*26-04-2024*	**Use Case Reference(s):**	Remove Admin from the Organization
Revision History:	*None*		
Objective:	*To check if root user is able to delete a admin in organization*		
Product/Ver/Module:	*Organization Module*		
Environment:	*Both web and backend server is deployed inside master node and running successfully.*		
Assumptions:	*The logged in user has root access.*		
Pre-Requisite:	*The logged in user has root access and has an existing organization.*		
Step No.	**Execution description**		**Procedure result**
1	*User clicks on edit button on the admin they want to delete on user management page.*		*System opens a modal box which has details of admin and a delete button.*
2	*User clicks the delete button*		*System successfully removes the admin, updates the list and redirects to user management page.*
Comments: The test case is passed. The admin user is deleted in the database.			
✓ **Passed** **Failed** **Not Executed**			

6.2.1.16 Adding Resources to Cluster

Table 6.16: Adding Resources to Cluster

Test case to check if the user can add resources to the cluster.

Adding Resources to Cluster			
Test Case ID:	*17*	**QA Test Engineer:**	*Fatima Siddiqui*
Test Case Version:	*1*	**Reviewed By:**	*Usman Faisal*
Test Date:	*25-04-2024*	**Use Case Reference(s):**	Add Cluster Resources
Revision History:	*None*		
Objective:	*To check if the user can add and customize resources within the cluster.*		
Product/Ver/Module:	*Adding Resources to Cluster Module*		
Environment:	*Both web and backend server is deployed inside master node and running successfully.*		
Assumptions:	*Cluster management screen and button to add cluster is visible to the user.*		
Pre-Requisite:	*User is logged in and has the necessary permissions for cluster creation.*		
Step No.	**Execution description**		**Procedure result**
1	*The user selects the cluster which he wants to customize and checks for resources to add.*		*The system verifies compatibility of that resource with the cluster.*
2	*The user fills in the details about the resources and clicks to add cluster.*		*The system stores data and adds resources to the cluster.*
Comments: The test case is passed. The resources were modified in the cluster.			
✓ **Passed** **Failed** **Not Executed**			

6.2.1.17 User Joining Queue (Added to Waiting List)

Table 6.17: User Joining Queue (Added to Waiting List)

Test case to check if the user is able to join a queue for a tool

User Joining Queue (Added to Waiting List)			
Test Case ID:	*14*	**QA Test Engineer:**	*Hassan Rehman*
Test Case Version:	*1*	**Reviewed By:**	*Usman Faisal*
Test Date:	*29-04-2024*	**Use Case Reference(s):**	Adjusting Tool Queue Limit
Revision History:	*None*		
Objective:	*To join a queue for a tool, but being added to the waiting list due to a filled queue*		
Product/Ver/Module:	*Tool Queue Module*		
Environment:	*Set up has been done on local machine and running successfully.*		
Assumptions:	*The user is logged into their account. The desired tool's queue is already filled to its maximum capacity.*		
Pre-Requisite:	*The tool is installed and accessible to the user, and the desired tool's queue is filled to its maximum capacity.*		
Step No.	**Execution description**		**Procedure result**
1	*User goes to the infrastructure*		*System displays infrastructure details*
2	*User selects the desired tool*		*System opens the tool's details*
3	*User clicks on "Get in Queue"*		*System attempts to add the user to the queue*
4	*System checks the queue status*		*Queue is found to be filled to its maximum capacity*
5	*User is added to the waiting list*		*System confirms the user is added to the waiting list*
Comments: The test case is passed. The user has been successfully added to the waiting list.			
✓ **Passed** **Failed** **Not Executed**			

6.2.2 Non Functional Test case

The following are the non-functional test cases of the system:

6.2.2.1 Correctness Test Case: Verifying Correctness of Tool Functionality

Table 6.18: Correctness Test Case: Verifying Correctness of Tool Functionality

Test case to verify the Correctness of tool installation

Non Functional Use case: Verifying Correctness of Tool Functionality			
Test Case ID:	*14*	**QA Test Engineer:**	*Hassan Rehman*
Test Case Version:	*1*	**Reviewed By:**	*Usman Faisal*
Test Date:	*01-05-2024*	**Use Case Reference(s):**	-
Revision History:	*None*		
Objective:	*To verify the correctness of the tool's functionality*		
Product/Ver/Module:	*Tool Functionality Module*		
Environment:	*Stable environment with required dependencies*		
Assumptions:	*The tool is properly configured and accessible*		
Pre-Requisite:	*User has appropriate permissions to access the tool*		
Step No.	**Execution description**		**Procedure result**
1	*User accesses the tool's interface*		*Tool's interface is successfully accessed*
2	*User performs predefined actions and inputs*		*Actions and inputs are executed accurately*
3	*User observes tool's behavior and output*		*Behavior and output are as expected*
Comments: The test case is passed. The tool's functionality has been verified to be correct.			
✓ **Passed** **Failed** **Not Executed**			

6.2.2.2 Reliability Test Case: Installing Tool from UI

Table 6.19: Reliability Test Case: Installing Tool from UI

Test case to verify the Reliability of tool installation

Reliability Test Case: Installing Tool from Infrastructure			
Test Case ID:	*14*	**QA Test Engineer:**	*Hassan Rehman*
Test Case Version:	*1*	**Reviewed By:**	*Usman Faisal*
Test Date:	*01-05-2024*	**Use Case Reference(s):**	-
Revision History:	*None*		
Objective:	*To verify that the tool can be successfully installed from the infrastructure*		
Product/Ver/Module:	*Tool Installation Module*		
Environment:	*Infrastructure environment with necessary dependencies*		
Assumptions:	*The infrastructure is accessible and stable*		
Pre-Requisite:	*User has appropriate permissions to install the tool*		
Step No.	**Execution description**		**Procedure result**
1	*User accesses the infrastructure*		*Infrastructure is successfully accessed*
2	*User navigates to the tool installation section*		*Tool installation section is successfully reached*
3	*User initiates the tool installation process*		*Tool installation process starts*
4	*User completes the installation steps*		*Tool is successfully installed*
Comments: The test case is passed. The tool has been successfully installed from the infrastructure.			
✓ **Passed** **Failed** **Not Executed**			

6.3 Test Metrics

This section illustrates the metrics used to evaluate the performance and effectiveness of the system, including both functional and non-functional requirements. We focus on two key metrics:

- **Test Case Defect Density**:

$$\text{Test Case Defect Density} = \left(\frac{\text{Number of Test Cases Failed}}{\text{Number of Test Cases Executed}} \right) \times 100\%$$

- **Test Case Effectiveness**:

$$\text{Test Case Effectiveness} = \left(\frac{\text{Number of Defects Detected Using Test Cases}}{\text{Total Number of Defects Detected}} \right) \times 100\%$$

6.3.1 Functional Test Metrics

Table 6.20: Functional Test Metrics

This table outlines the metrics for the functional test cases of the system.

Metric	Purpose
Number of Test Cases	17
Number of Test Cases Passed	17
Number of Test Cases Failed	0
Test Case Defect Density	0
Test Case Effectiveness	75

6.3.2 Non-Functional Test Metrics

Table 6.21: Non-Functional Test Metrics

This table outlines the metrics for the non-functional test cases of the system.

Metric	Purpose
Number of Test Cases	2
Number of Test Cases Passed	2
Number of Test Cases Failed	0
Test Case Defect Density	0
Test Case Effectiveness	66

6.4 Conclusion

This chapter details the processes and technologies essential for setting up and deploying a Kubernetes virtual cluster locally using Docker, Ansible, and Vagrant. The chapter commences with a walkthrough for establishing a local cluster, starting with the installation of Ubuntu as the operating system, praised for its user-friendliness and broad community support. It describes how to make an Ubuntu USB stick that can boot up, how to update the system, and how to turn on virtualization in the BIOS. Then, it concludes by demonstrating how to extend the Ansible in order to managing Kubernetes controller and worker nodes. This emphasizes a streamlined configuration and deployment. This process involves defining groups, creating roles, initializing the Kubernetes control plane, joining worker nodes, and executing the ansible-playbook command, culminating with an update of the deployment documentation. Moreover, it includes about the cluster roles assignment, installation of tools needed for ML , the automation of BinderHub, JupyterHub and their inter-linkage. Then it describes about the key steps used for server deployment of Backend and Frontend on the Kubernetes cluster which were initially running at local hosts. Additional information about authentication and queue mechanisms is also provided. Furthermore, the testing framework is also used to verify the reliability and efficiency of the deployed infrastructure. A suite of test cases assessed both functional and non-functional system aspects, with metrics like Test Case Defect Density and Test Case Effectiveness highlighting our robust testing techniques.

Chapter 7 User Manual

This chapter will provide a thorough documentation for the new users in order to setup, install and configure the system to meet their needs, ensuring they can fully leverage the capabilities of the system to streamline the deployment and management of AI/ML technologies.

7.1 Introduction

The MLOps-Driven HPC Infrastructure is a robust, scalable solution crafted to meet the needs of SMEs seeking to integrate advanced AI and ML technologies into their operations. This manual serves as a guide to help users navigate the system, from initial setup to daily operations to maintenance.

7.1.1 Overview of the System

The system is built upon the virtualization which means that users can utilize the high-performance GPU servers by integrating them, configured for optimal AI/ML operations. Key features include:

- Automated setup and configuration processes
- Pre-configured tools for AI/ML development
- A user-friendly management platform
- Organization's cluster user access management
- Integration of resource monitoring tools

The architecture supports both on-premises and cloud deployments using the IP configuration of virtual or real machines, utilizing technologies like Kubernetes, Helm Charts, and Ansible for management and orchestration.

7.1.2 Purpose & Benefits

The infrastructure's main purpose is to eliminate common barriers such as complex setup processes, security vulnerabilities, and the high costs associated with AI/ML implementations. It is designed to be accessible to organizations regardless of their size or technical expertise. If utilized our system, the organizations can expect the following benefits:

- **Reduced Operational Costs**: By automating numerous aspects of the setup and management, the system decreases the need for extensive technical staff and lowers operational costs.
- **Enhanced Data Security**: With on-premises deployment, sensitive data is kept secure within the

organization's control, mitigating risks associated with external cloud storage.

- **Scalability and Flexibility**: The system adapts to changing data and computational demands without user intervention, making it highly scalable and flexible.
- **User-Friendly Interface**: The management platform is designed to be intuitive, allowing users to easily configure and monitor their AI/ML environments.

7.2 Getting Started

This section guides new users through the initial steps necessary for setting up the system. It includes the prerequisites for installation, software requirements, and a brief overview to prepare your environment for deployment.

7.2.1 System Prerequisites

To ensure a smooth installation and operation of the application, the following system requirements must be met. Please ensure that your system is compatible with these specifications:

- **Operating System**: Ubuntu (latest stable version recommended)
- **Version Control**: Git (latest version) for cloning repositories
- **JavaScript Runtime**: Node.js with npm (latest stable version) for managing frontend dependencies
- **Programming Language**: Python (latest stable version) for running backend services
- **Containerization**: Docker (latest version) for building and running containers
- **Orchestration**: Kubernetes (v1.28+) for managing containerized applications
- **Automation Scripting Tool**: Ansible (latest version) for automating software provisioning, configuration management, and application deployment
- **Virtualization**: Vagrant (2.4+) for building and managing virtual machine environments
- **Virtual Box**: Virtual Box (latest version) for multiple VM nodes

Ensure that your machine has adequate memory and processing power to handle the computational demands, especially when working with AI/ML workloads in a virtualized environment. For optimal performance, a multi-core processor, a minimum of 16GB RAM and a powerful latest GPU is recommended.

7.2.2 Setup & Installation

Setting up the MLOps-Driven HPC Infrastructure involves several steps, from cloning the repository to deploying the components on Kubernetes. Follow these instructions to ensure a proper setup.

7.2.2.1 Cloning the Repository

Start by cloning the project repository from GitHub to get the necessary files and scripts. Open your terminal and run the following command:

```
git clone https://github.com/Questra-Digital/HPC-Infra-Hex
```

This will download the repository to your local machine in a new directory called “HPC-Infra-Hex”.

7.2.2.2 Launching Virtual Machines

Navigate to the directory where the repository has been cloned. Initialize the virtual environment using Vagrant, which will configure and start all required virtual machines (VMs). In your terminal, execute:

```
cd HPC-Infra-Hex
vagrant up
```

This command will start the setup of the VMs based on the configuration specified in the “Vagrantfile”. Note that this process can take some time, depending on your system’s performance and internet speed.

7.2.2.3 Deploying the Backend and Frontend

Once the VMs are up and running, you need to deploy the backend and frontend components within the master node of your Kubernetes cluster. Follow the detailed steps provided in Section 6.1.8 of this report to proceed with the deployment. This section includes commands and configurations needed to ensure the services are correctly set up and port forward them to access on your host machine.

7.2.2.4 Copying the Kubernetes Configuration File

To manage your Kubernetes cluster nodes from the host machine, you need to copy the Kubernetes configuration file from the virtual machine to your local system. This allows you to use “kubectl” commands directly from your host machine. Use the following command, replacing “Virtual Machine IP” with the actual IP address of your virtual machine:

```
scp vagrant@{Virtual Machine IP}:~/.kube/config ~/.kube/config
```

This command utilizes scp (secure copy protocol) to transfer the “.kube/config” file from the virtual machine to the “.kube” directory on your host machine, ensuring you have the necessary permissions to

interact with the Kubernetes cluster.

7.2.3 Initial Setup

7.2.3.1 Service Account & Initializing Cluster Access Roles

To manage the resources and operations within your Kubernetes cluster securely, you need to create a specific service account named “hpc”. Use the following command:

```
kubectl create serviceaccount hpc
```

After creating the service account, apply the necessary cluster role bindings to set permissions. First, ensure you are in the directory containing the “clusterrolehpc.yaml” file, which defines the access roles for the hpc service account. Run the following command to create the service account and apply the cluster role configurations:

```
kubectl apply -f clusterrolehpc.yaml
```

This step initializes the cluster access roles, granting the hpc service account the required permissions to operate within the cluster effectively.

7.2.3.2 Root Account Setup

Once the system’s frontend is up and running, you can begin setting up the root account. This account will have owner privileges and allow you to configure settings across the entire system. After setting up port forwarding, open a web browser and navigate to “http://localhost:3000”. If you are accessing the page for the first time, you will be prompted to set up the root account. This setup process involves entering details about your organization and creating initial administrative credentials. Follow the on-screen instructions to complete this setup. You will need to provide:

- Organization Details
- Root Account’s email
- A strong password for the root account

Once you have filled in all the required fields, submit the form to finalize the root account setup. Upon successful submission and verification of the details, the system will redirect you to the homepage of the MLOps-Driven HPC Infrastructure. From here, you can start configuring and managing the system according to your organization’s needs.

7.3 System Interface

In this section, we'll introduce new users to our system portal that provides a simplified overview of its pages and the overall flow of the interface.

7.3.1 System Navigation Overview

Our streamlined navigation system boasts a user-friendly navbar that offers an instant access to crucial pages. This important feature enhances user experience, ensuring seamless navigation to desired destinations.

7.3.2 User Management Page

For the user management page, the root user can add admins, team members , team leads and similarly can remove them which goes into the hierarchy, thus allowing the users to have privileged access to specific functionalities.

7.3.3 Tools Management Page

Within our tools management page the users can enjoy a seamless experience of adding and removing tools according to their infrastructure precisely to their needs. The interface provides a straightforward process: users can select from a list of available tools to install those that serve their requirements using the drag and drop box and ensuring a customized setup.

On the administrative side the admins possess the authority to remove unnecessary tools from the infrastructure. This ensures optimal efficiency and resource management. Moreover through the queue management system, admins can prioritize tool installation and removal requests, maintaining an orderly workflow.

Furthermore, users have visibility into the status of tool installation or removal, empowering them with real-time updates on progress.

7.3.4 Run Command Page

This innovative feature allows the users to execute commands directly from the system interface, eliminating the need to switch to a separate terminal. By providing a command box within the interface the users can easily input commands and execute them within the system nodes with ease and efficiency. This seamless integration enhances user workflow that enables swift and direct interaction with the system without the hassle of navigating to a terminal window.

7.3.5 Resource Monitoring Page

In our system, we've leveraged Grafana to provide users with comprehensive insights into resource utilization. Grafana offers a dynamic dashboard showcasing vital metrics such as CPU usage, memory utilization, GPU performance, error rates, and more. With this tool at their disposal, users can effortlessly monitor the health and performance of their resources.

7.4 Maintenance & Troubleshooting

7.4.1 Introduction

In this section, we outline the procedures for maintaining and troubleshooting the HPC-Infra-Hex project.

7.4.2 Maintainers

As an open-source project, different students will take on the role of maintaining the codebase. Below are the current maintainers:

1. Hassan Rehman (hassan210302@gmail.com)
2. Usman Faisal (usmanfaisal49@gmail.com)
3. Fatima siddiqui (l200970@lhr.nu.edu.pk)

7.4.3 Common Issues Troubleshooting

This section provides guidance on troubleshooting common issues encountered while working on the project, particularly related to Kubernetes.

7.4.3.1 Issue 1: Missing Docker Installation

Symptoms: Docker is not installed on Kubernetes worker nodes.
Solution: Install Docker on the affected nodes using the appropriate installation method for your operating system. Ensure Docker is properly configured and running.

7.4.3.2 Issue 2: Slow Network Performance

Symptoms: Kubernetes pods experience slow network connectivity or high latency.
Solution: Check network configurations, including network policies, ingress, and egress rules. Analyze network traffic using tools like tcpdump or Wireshark. Optimize network settings and consider using network accelerators if necessary.

7.4.3.3 Issue 3: Kubernetes Cluster Unavailability

Symptoms: The Kubernetes cluster becomes unresponsive or inaccessible.

Solution: Check the status of Kubernetes control plane components (apiserver, controller-manager, scheduler). Verify the health of worker nodes and underlying infrastructure. Restart Kubernetes components if necessary and investigate logs for errors.

7.4.3.4 Issue 4: Insufficient Node Resources

Symptoms: Pods fail to schedule due to insufficient node resources (CPU, memory, disk).

Solution: Monitor node resource usage using Kubernetes dashboard or command-line tools. Adjust resource requests and limits for pods as needed. Consider scaling the cluster or upgrading hardware resources to meet demand.

7.4.3.5 Issue 5: DNS Resolution Problems

Symptoms: Pods are unable to resolve DNS names or experience DNS lookup failures.

Solution: Verify DNS configuration in Kubernetes cluster settings. Ensure DNS service is running and accessible to all pods. Troubleshoot DNS resolution using tools like nslookup or dig.

7.4.3.6 Issue 6: Persistent Volume Mount Failures

Symptoms: Pods fail to mount persistent volumes or experience errors related to storage.

Solution: Check the status of persistent volumes (PVs) and persistent volume claims (PVCs). Verify storage class configuration and availability of underlying storage resources. Troubleshoot storage connectivity and permissions issues.

7.4.3.7 Issue 7: Missing PersistentVolumes (PVs)

Symptoms: PersistentVolumes required by the application are not created.

Solution: Apply the PersistentVolume configuration using the following command:

```
kubectl apply -f pv.yaml
```

Ensure that the PersistentVolume configuration is correct and matches the requirements of the application.

7.4.4 Reporting Issues

If you encounter any issues not covered in this documentation, please report them on the GitHub repository's issue tracker.

7.5 Conclusion

The user manual section aims to provide comprehensive guidance to new users from installation to everyday operations. Our goal is to provide users a simple processes through a user-friendly interface and detailed, step-by-step instructions. Throughout this manual, it outlines how to set up and maintain the system, navigate through its interfaces, and address potential issues. The integration of tools like Kubernetes, Docker, and Grafana is aimed at enhancing operational efficiency and providing real-time analytics that support informed decision-making. As technology evolves, so too will the challenges of managing the system. We encourage users to provide feedback and participate in the ongoing development of this project by contributing to our GitHub repository. Your insights are invaluable to improving and adapting the system to meet future needs.

Chapter 8 Conclusion and Future Work

The project ”MLOps-Driven HPC Infrastructure” has laid the groundwork for an innovative MLOps-driven HPC infrastructure designed to streamline AI/ML workflows in an efficient and scalable manner. This project aims to solve the problems of private SMEs such as keeping their data on premises as well as secure and confidential and others that were highlighted in the sections above. The combination of cutting-edge computing resources and contemporary operating procedures has produced a stable platform that can manage challenging AI/ML jobs. In the development phase, we have successfully implemented on of the key features of the project which was deploying clusters through Ansible and Vagrant configuration. Throughout this report, we have provided a detailed architecture of the system through comprehensive class, ER diagrams and descriptions. These articulate the system's core functionalities, such as resource allocation, process orchestration, data handling, and user interaction mechanisms, and testing. A detailed user manual is also provided in order to help the users setup the system on their systems.

As for the next stage of development, there are several areas where this project will be implemented in phases:

- Integration with Metabase and Feature Hub: Integrate our system with Metabase for business intelligence and analytics, as well as Feature Hub for feature flagging in ML models, to expand its possibilities so that users will benefit from data analysis and model maintenance as a result of this.
- Customizable AI/ML Pipelines: Create more adaptable AI/ML pipelines that can be quickly tailored to a wide range of applications and requirements.
- System Deployment: Setting up and deploying the whole system on organization's GPUs and infrastructure.

As we move forward with our project, our key objective will be to continually expand, update, and improve our MLOps-driven HPC infrastructure to fulfil the increasing demands of AI/ML development for SMEs while being at the cutting edge of technical growth.

Bibliography

[1] F. Li, Z. Gui, H. Wu, J. Gong, Y. Wang, S. Tian, and J. Zhang, "Big enterprise registration data imputation: Supporting spatiotemporal analysis of industries in china," *Computers, Environment and Urban Systems*, vol. 70, pp. 9–23, 2018.

[2] M. Eldred, A. Good, and C. Adams, "A case study on data protection and security decisions in cloud hpc," in *2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 564–568, IEEE, 2015.

[3] E. Raj, D. Buffoni, M. Westerlund, and K. Ahola, "Edge mlops: An automation framework for aiot applications," pp. 191–200, 10 2021.

[4] F. Calefato, L. Quaranta, F. Lanubile, and M. Kalinowski, "Assessing the use of automl for data-driven software engineering," *arXiv preprint arXiv:2307.10774*, 2023.

[5] Y. Zhou, Y. Yu, and B. Ding, "Towards mlops: A case study of ml pipeline platform," *2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, pp. 494–500, 2020.

[6] A. Ganne, "Applying azure to automate dev ops for small ml smart sensors," *International Research Journal of Modernization in Engineering Technology*, vol. 4, no. 12, 2022.

[7] N. Zhou, Y. Georgiou, M. Pospieszny, L. Zhong, H. Zhou, C. Niethammer, B. Pejak, O. Marko, and D. Hoppe, "Container orchestration on hpc systems through kubernetes," *Journal of Cloud Computing*, vol. 10, no. 1, pp. 1–14, 2021.

[8] A. Suryanarayanan, A. Chala, L. Xu, G. Shobha, J. Shetty, and R. Dev, "Design and implementation of machine learning evaluation metrics on hpcc systems," in *2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, pp. 1–7, IEEE, 2019.

[9] D. Golec, I. Strugar, and D. Belak, "The benefits of enterprise data warehouse implementation in

cloud vs. on-premises," *ENTRENOVA-ENTerprise REsearch InNOVAtion*, vol. 7, no. 1, pp. 66–74, 2021.

[10] D. Chahal, M. Mishra, S. Palepu, and R. Singhal, "Performance and cost comparison of cloud services for deep learning workload," in *Companion of the ACM/SPEC International Conference on Performance Engineering*, pp. 49–55, 2021.

[11] A. Botchkarev, "Evaluating performance of regression machine learning models using multiple error metrics in azure machine learning studio," *Available at SSRN 3177507*, 2018.

[12] N. A. Gawande, J. A. Daily, C. Siegel, N. R. Tallent, and A. Vishnu, "Scaling deep learning workloads: Nvidia dgx-1/pascal and intel knights landing," *Future Generation Computer Systems*, vol. 108, pp. 1162–1172, 2020.

[13] S. T. Brown, P. Buitrago, E. Hanna, S. Sanielevici, R. Scibek, and N. A. Nystrom, "Bridges-2: A platform for rapidly-evolving and data intensive research," in *Practice and Experience in Advanced Research Computing*, pp. 1–4, 2021.