Kadmelia implementation

Mobile and distributed systems, D7024E

Group 12

Authors:

Adrian Fingal	adrfin-1@student.ltu.se
Oscar Johansson	osajoh-9@student.ltu.se
Hugo Hedlund	hughed-0@student.ltu.se

Github:

https://github.com/Hedlund01/d7024e

Department of Computer Science, Electrical and Space Engineering

September 5, 2025

Contents

1 Frameworks

In this section, we will discuss the various frameworks and technologies that are being utilized in the project. This includes both the backend and frontend technologies, as well as any other tools that are essential for the development process.

Programming language: Go is used as the primary programming language for the backend services, leveraging its concurrency features and performance.

Logging: Logrus is used for structured logging in the application, providing a simple API for logging that allows to pass the container ip address for each log entry to easily identify which node the log entry originated from.

Containerization: Docker in swarm mode is used for container orchestration, enabling the management of multiple containers.

Automation: Bash scripts are employed to automate common tasks such as building and deploying the application, making it easier to manage the deployment and development process.

2 Sprint 0

During sprint 0 we setup the project repository from the Github template given in the course. We also configured a containerization strategy using Docker in swarm mode. We established bash scripts to automate common tasks, such as building and deploying the application. This lets us with a simple script deploy up to n instances, where n is a argument for the deploy script, with ease. Utilizing these scripts, we can easily scale up and down the number of instances in our swarm and Kademlia network in the upcoming sprints.
After setting up the containerization tools, we started on the implementation of finding nodes in the same docker network and establishing communication between them. Our approach involved DNS lookups via the docker DNS service, to get the docker ip addresses of the other containers in the same docker swarm stack. The drawback of this approach is that the containers that start up early won't be able to find the containers that start up later, since they don't exist yet. We didn't solve this problem yet, as we will implement Kademlia protocol for peer discovery to address these issues in the upcoming sprint.

3 Sprint 1

3.1 Planned work

As illustrated in Figure 1 1, the plan is to complete all underlying functionality (M1–M7) during the first sprint. Subsequently, focus will shift to the qualification goals (U1–U5). This structure was chosen because establishing and verifying the core functionality first ensures a stable foundation, which in turn facilitates the successful implementation of the qualification objectives.

3.2 Organizing Sprints

To structure and organize the work efficiently, GitHub Projects is used to create a storyboard for both sprints. The storyboard is available under the Projects tab in the repository, providing an overview of the workflow throughout the course.

Appendices

Appendix A

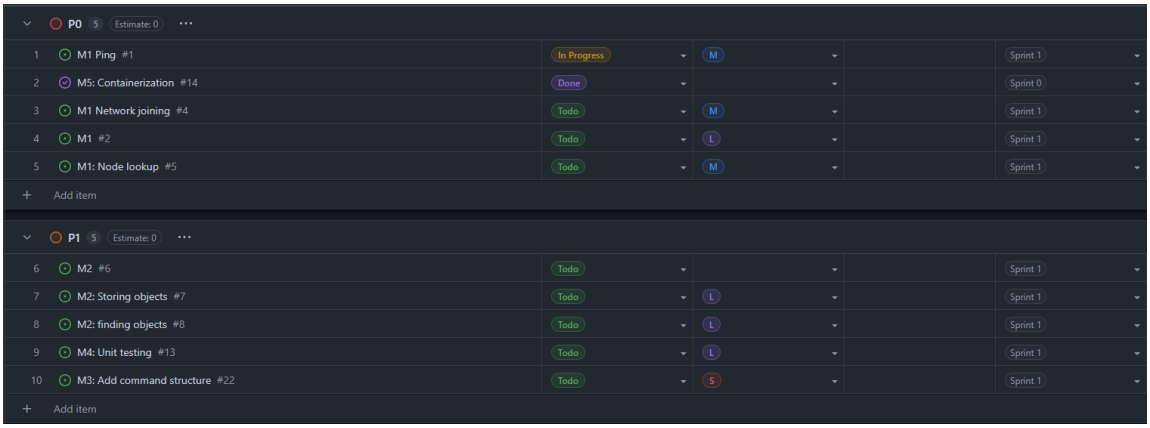

Figure 1: Todos for Sprints at Sprint 0

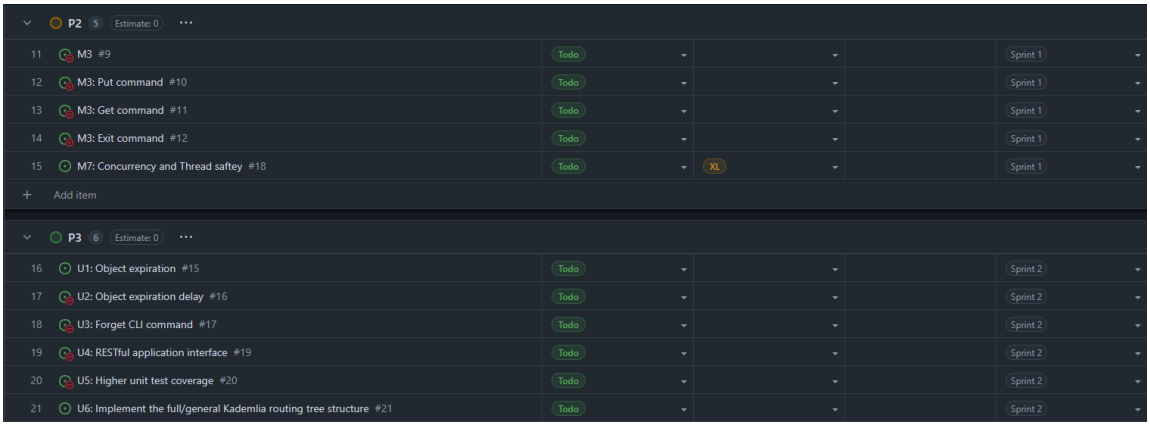

Figure 2: Todos for Sprints at Sprint 0, continued