Project Management

Configuration management

Author: **Pavel Šesták** (**xsesta07**)

April 26, 2023

Contents

1 Introduction

The aim of this work is to summarize the development process including the technologies, software and versioning used.

2 Development platform

This project is designed as a web application that is split into a reactive javascript client that communicates with an api gateway. The Api gateway exposes the frontend to a network of microservices, where each microservice manages part of the domain. The web application, due to the absence of a public domain, is made available on the network using zerotier. The frontend client is written in the Svelte framework. Individual microservices are implemented in dotnet. The minimal api framework is used to make the REST interface available. Asynchronous communication between individual services is ensured using the RabbitMQ service. Data is stored in NoSQL MongoDB databases due to higher speed and the lack of use of standard relational database checks.Since the management of more than ten services is very demanding, CI/CD is used, which performs automatic tests and builds for us. If the build in the master branch passes, then it is automatically deployed to Docker on the production server. This technique enables the independent deployment of versions of individual services even several times a day without the entire product having to be unavailable.

3 Software tools used for development

Due to the extensive stack used during development, the list of tools used is varied. For the development of the reactive frontend, the standard package tool npm is used, and as a development environment, the lightweight visual studio code is used. For writing in dotnet we use the standard Visual Studio 2022 Community Edition from Microsoft, which together with Rider is the only reasonable choice for dotnet. The MongoDbCompass desktop application is used for direct access to the database. The Docker Desktop program is used to run individual services in the local environment. For testing was used Postman and jMeter for performance testing.

4 Versioning

The Git product was used for source code versioning. The Git repository was hosted remotely on Gitlab. In the git settings, we restricted access to the main branch and branch merging took place in scrum meetings with the participation of all members of the development team.

5 Requests for changes

Changes are always discussed in team meetings, where domain experts always find some improvement to the current design. The design is immediately edited in a shared repository, so all programmers have access to it. UML diagrams are enriched with many notes for this reason.

6 Source code formatting

The source code formatting was determined to be the default dotnet formatting defined in Visual Studio. These standards are met thanks to the IDE, which can immediately reformat the entire project according to the specified criteria.

7 Monitoring

For monitoring microservices is used Prometheus scraping and Grafana visualization tools.

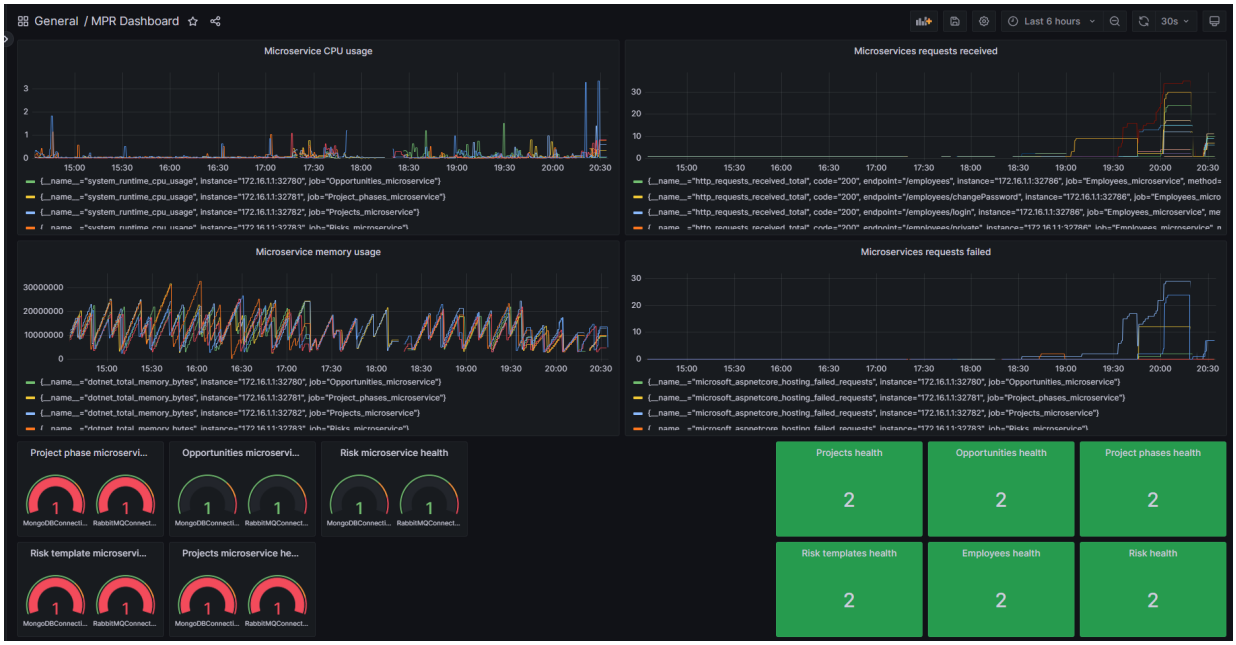

Figure 1: Grafana administration dashboard