

DevOps autumn 2020 – Assignment 5

By Johannes Martikkala – 253026

In this assignment we formed simple docker-compose of four different containers and rabbitmq in between those. We had original container which delivered the initial messages, intermediate container which took the messages and after a second pushed them back to rabbitmq, observer which received all the messages and lastly httpserver which read the file observer wrote. In between all of those we had rabbitmq which was the main topic of this assignment.

Topic based communications has its strengths. For starters, it is clearly more scalable than some other message patterns as it removes completely the need to track listeners. You only need to know what to listen or where to publish the messages while the software providing the rabbitmq does the rest. Also, it offers way to the data handler to publish messages whenever needed without considering if the receiver is yet to receive that message.

However, it comes with limitations as well. For starters, there is not clear way to operate if the message queues are in different servers without having them speak through HTTP-requests. Also, it does not work well with asynchronous services in a way that if the listener is not available just when the message is delivered it is lost completely. These things can cause harm and since original publisher cannot know if the message is read without external messaging, the whole system can be left in unknown state.

All in all, this assignment was fine. It was already something that I have been done in previous courses and while it is always good to refresh your memory on doing things again, I did not feel like I learned a lot. The main thing I was missing in this assignment was motivation to learn these things. For starters, I have no clue how this is related to our course. Of course, these are tools I might use when developing products, but the goal of this assignment was missing. It felt like I was doing something for sake of doing things rather than learning something meaningful.

Extra info when running this project:

There is a very rare bug whenever you are running this which may cause it to not function properly. As the containers start in undefined order there is scenario where messages can be sent before any reader is ready. However, due to this bug there is also more rarer bug where when trying to initialize containers if for some reason the previous processes (in previous runtime) haven't been terminated correctly, the socket it tries to initialize isn't free yet causing the whole compose to halt. If this happens, please run these containers again.