

# Docker – bigger case example from real life

**Developing, continuous integration (CI) and continuous delivery (CD) of team developed software and other assets to running environment**

12.2.2026

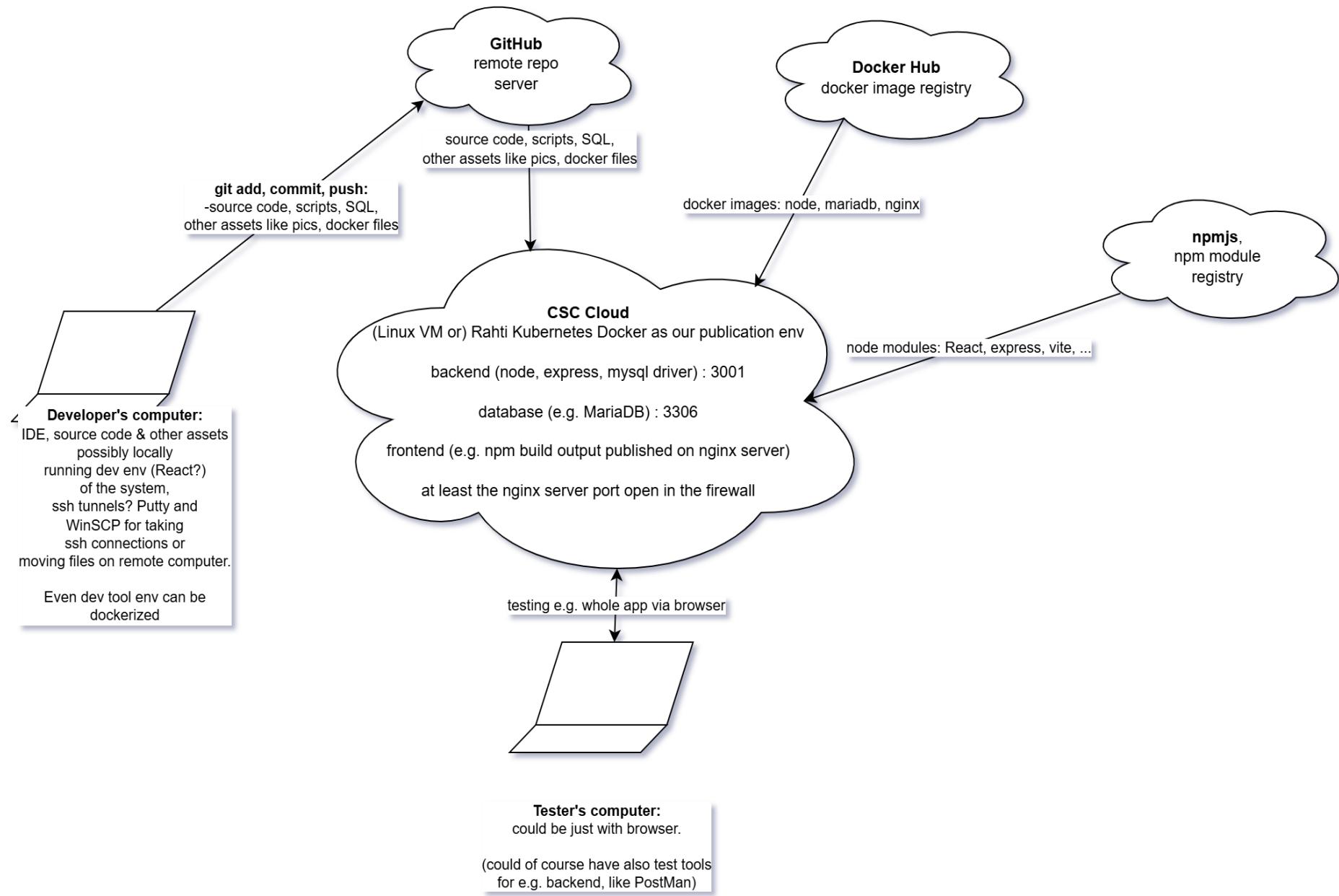


# DevOps principle image

- The following image depicts one kind of DevOps situation and needs. A normal project where we have been developing a full-stack application (frontend, backend, database) for a customer.
- Without a good DevOps process there would be a lot of 'manual' tool and server installation, code/assets/binaries updating, configuration and deployment.
- And potentially also problems with software versions etc.

# DevOps example

- What benefits you see with this kind of approach?
- What skills you'll need to learn?



# Task – Consider how DevOps scripting could help

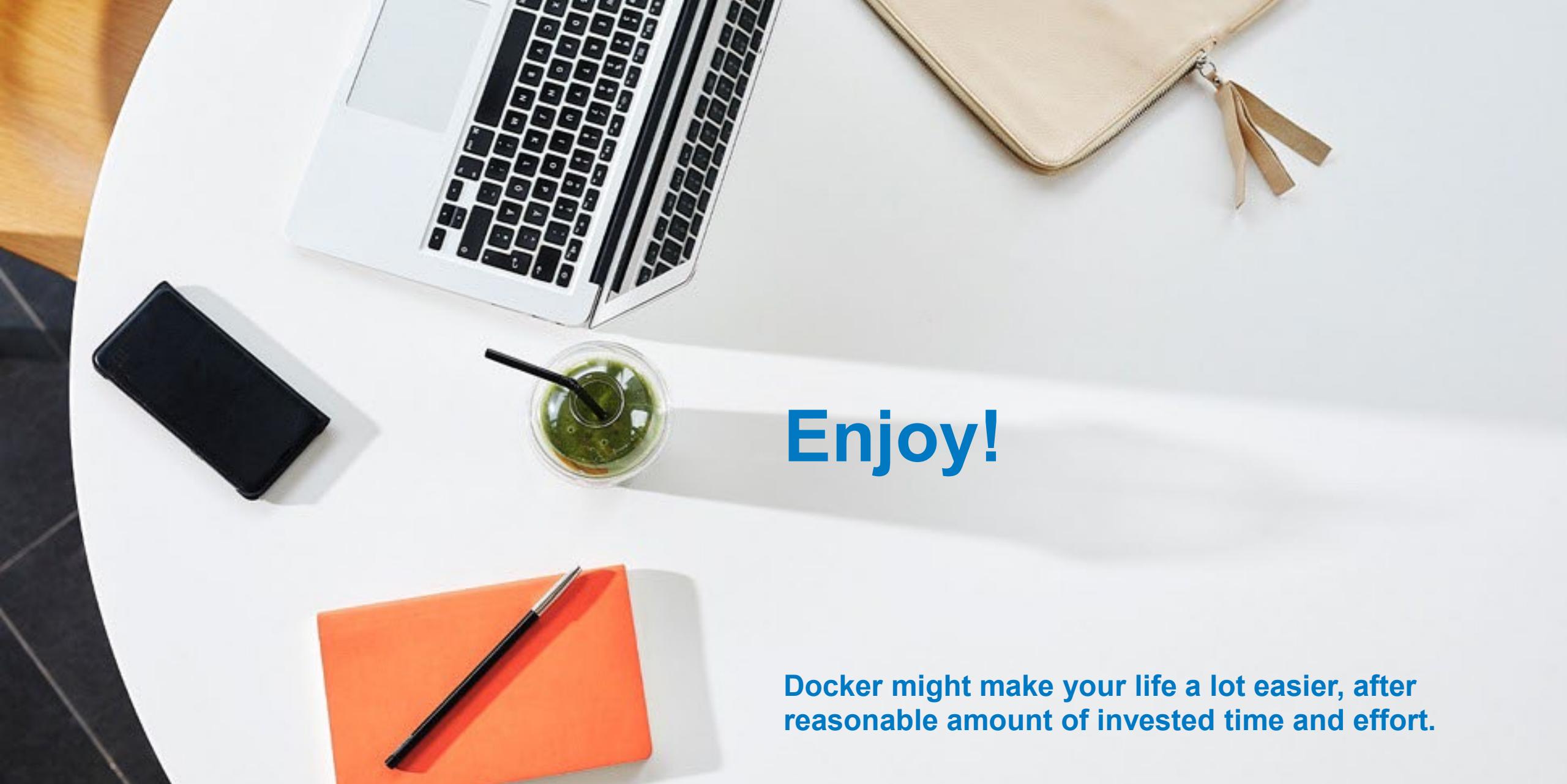
- Think how much we could improve our work processes, products and speed, if building of all the items in the picture could be written **in scripts**?

# Answers 1/2 – The DevOps scripting could ...

- save a lot of repetitive 'manual' work that is stressing, boring after Nth time doing same installation, and error-prone
- be triggered automatically when a new commit is made to main branch in GitHub. Or checked periodically e.g. with Linux *cron* timing command
- fetch always the latest versions of: 1. developers' work, 2. ready-made code libraries and 3. ready-made images for docker containers. (Well, if we want the latest version, e.g. while experimenting maybe)
- build the runnable versions for production environment (not just the development environment versions like e.g. with React web app development on laptop) and deliver them

# Answers 2/2 – The DevOps scripting could

- Be used in building automated testing (and whole CI), or even for continuous delivery (CD) of the product to production environment
- Be improved continuously so that they would gradually automatize more and more with the DevOps
- Be used as basis for the DevOps in the next project for a new customer! Knowledge and many scripts can be reapplied to new projects. The core tools are open source and used by majority of the IT organizations.
- Allow moving the whole system easily to different OS or different version of the OS, as containers run on any machine with Docker installed, regardless of the host computer's software or libraries.



# Enjoy!

Docker might make your life a lot easier, after reasonable amount of invested time and effort.