{() => fs}

Fullstack Part 11

So you have a fresh feature ready to be shipped. What happens next? Do you upload files to a server manually? How do you manage the version of your product running in the wild? How do you make sure it works, and roll back to a safe version if it doesn’t?

Doing all the above manually is a pain and doesn’t scale well for a larger team. That’s why we have Continuous Integration / Continuous Delivery systems, in short CI/CD systems. In this part, you will gain an understanding of why you should use a CI/CD system, what can one do for you, and how to get started with GitHub Actions which is available to all GitHub users by default.

This module was crafted by the Engineering Team at Smartly.io. At Smartly.io, we automate every step of social advertising to unlock greater performance and creativity. We make every day of advertising easy, effective, and enjoyable for more than 650 brands worldwide, including eBay, Uber, and Zalando. We are one of the early adopters of GitHub Actions in wide-scale production use. Contributors: Anna Osipova, Anton Rautio, Mircea Halmagiu, Tomi Hiltunen.

Part updated 16th January 2024

- *Example project dependencies updated*

Part 10 Part 12

Previous part

Next part

About course

Course contents

FAQ

Partners

Challenge