Pipeline as Code

CONTINUOUS DELIVERY WITH JENKINS, KUBERNETES, AND TERRAFORM

MOHAMED LABOUARDY

MANNING
SHELTER ISLAND

For online information and ordering of this and other Manning books, please visit www.manning.com. The publisher offers discounts on this book when ordered in quantity. For more information, please contact

Special Sales Department
Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964
Email: orders@manning.com


Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964

Development editor: Karen Miller
Technical development editor: Christopher Haupt
Review editor: Mihaela Batinić
Production editor: Deirdre S. Hiam
Copy editor: Sharon WIlkey
Proofreader: Keri Hales
Technical proofreader: Werner Dijkerman
Typesetter and cover designer: Marija Tudor

ISBN 9781617297540
Printed in the United States of America

brief contents

contents

preface

Ten years ago, I wrote my first makefile to automate the testing, building, and deployment of a C++ application. Three years later, while working as a consultant, I came across Jenkins and Docker and discovered how to take my automation skills to the next level with CI/CD principles.

The beauty of CI/CD is that it's simply a rigorous way of recording what you're already doing. It doesn't fundamentally change how you do something, but it encourages you to record each step in the development process, enabling you and your team to reproduce the entire workflow later at scale. Over the next few months, I started writing blog posts, doing talks, and contributing to CI/CD-related tools.

However, setting up a CI/CD workflow has always been a very manual process for me. It was done via defining a series of individual jobs for the various pipeline tasks through a graphical interface. Each job was configured via web forms—filling in text boxes, selecting entries from drop-down lists, and so forth. And then the series of jobs were strung together, each triggering the next, into a pipeline. This made the troubleshooting experience a nightmare and reverting to the last known configuration in case of failure a tedious operation.

A few years later, the *pipeline-as-code* practice emerged as part of a larger "as code" movement that includes *infrastructure as code*. I could finally configure builds, tests, and deployment in code that is trackable and stored in a centralized Git repository. All the previous pains were alleviated.

I became a fan and believer of pipeline as code, as I transitioned from being a software engineer, tech leader, and senior DevOps manager to now co-leading my first startup as CTO. Pipeline as code became an important part of each project I was part of.

I had the chance to work on different types of architecture—from monolithic, to microservices, to serverless applications—having built and maintained CI/CD pipelines for large-scale applications. Along the way, I accumulated tips and best practices to follow while going through the journey of continuous everything.

The idea of sharing that experience is what triggered this book. Implementing pipeline as code is challenging for many teams, as they require the use of many tools and processes that all work together. The learning curve takes a lot of time and effort, leading people to wonder whether it's worth it. This book is a handbook experience on how to build a CI/CD pipeline from scratch, using the most widely adopted CI solution: Jenkins. I hope the result will help you embrace the new paradigm of building CI/CD pipelines.