Continuous Integration, Delivery, And Deployment: Reliable And Faster Software Releases With Automating Builds, Tests, And Deployment

Winner of the 2011 Jolt Excellence Award! Getting software released to users is often a painful, risky, and time-consuming process. This groundbreaking new book sets out the principles and technical practices that enable rapid, incremental delivery of high quality, valuable new functionality to users. Through automation of the build, deployment, and testing process, and improved collaboration between developers, testers, and operations, delivery teams can get changes released in a matter of hours— sometimes even minutes–no matter what the size of a project or the complexity of its code base. Jez Humble and David Farley begin by presenting the foundations of a rapid, reliable, low-risk delivery process. Next, they introduce the "deployment pipeline," an automated process for managing all changes, from check-in to release. Finally, they discuss the "ecosystem" needed to support continuous delivery, from infrastructure, data and configuration management to

governance. The authors introduce state-of-the-art techniques, including automated infrastructure management and data migration, and the use of virtualization. For each, they review key issues, identify best practices, and demonstrate how to mitigate risks. Coverage includes • Automating all facets of building, integrating, testing, and deploying software • Implementing deployment pipelines at team and organizational levels • Improving collaboration between developers, testers, and operations • Developing features incrementally on large and distributed teams • Implementing an effective configuration management strategy • Automating acceptance testing, from analysis to implementation • Testing capacity and other non-functional requirements • Implementing continuous deployment and zero-downtime releases • Managing infrastructure, data, components and dependencies • Navigating risk management, compliance, and auditing Whether you're a developer, systems administrator, tester, or manager, this book will help your organization move from idea to release faster than ever—so you can deliver value to your business rapidly and reliably.
Continuous integration, continuous delivery, and continuous

deployment are key software delivery processes in a DevOps environment. But what does each one do for your product development and release cycles? Brent Laster explains what these terms really boil down to, and how they work separately and together to help your team release software. This powerful set of disciplines, best practices, and technologies automates the process of integrating and delivering source code changes from inception through production. Although their implementation may vary, these processes are necessary to ensure that software is released frequently, reliably, and with high quality. You'll learn how: Continuous integration ensures that individual code changes are suitable for inclusion in the code base and merged in successfully Continuous delivery assembles your product, automatically testing the quality and functionality along the way, and produces deliverables that are proven to be deployable Continuous deployment simplifies releasing the product to customers, whether it's in the cloud, via download, or in some other format, while also allowing for limited deployments or rolling deployments back Getting started with the processes and the tools to continuously deliver high-quality software About This Book Incorporate popular

development practices to prevent messy code Automate your build, integration, release, and deployment processes with Jenkins, Git, and Gulp?and learn how continuous integration (CI) can save you time and money Gain an end-to-end overview of Continuous Integration using different languages (JavaScript and C#) and tools (Gulp and Jenkins) Who This Book Is For This book is for developers who want to understand and implement Continuous Integration and Delivery in their daily work. A basic knowledge of at least JavaScript and HTML/CSS is required. Knowing C# and SQL will come in handy. Most programmers who have programmed in a (compiled) C-like language will be able to follow along. What You Will Learn Get to know all the aspects of Continuous Integration, Deployment, and Delivery Find out how Git can be used in a CI environment Set up browser tests using Karma and Selenium and unit tests using Jasmine Use Node.js, npm, and Gulp to automate tasks such as linting, testing, and minification Explore different Jenkins jobs to integrate with Node.js and C# projects Perform Continuous Delivery and Deployment using Jenkins Test and deliver a web API In Detail The challenge faced by many teams while implementing Continuous Deployment is that it requires the use of many tools and processes

that all work together. Learning and implementing all these tools (correctly) takes a lot of time and effort, leading people to wonder whether it's really worth it. This book sets up a project to show you the different steps, processes, and tools in Continuous Deployment and the actual problems they solve. We start by introducing Continuous Integration (CI), deployment, and delivery as well as providing an overview of the tools used in CI. You'll then create a web app and see how Git can be used in a CI environment. Moving on, you'll explore unit testing using Jasmine and browser testing using Karma and Selenium for your app. You'll also find out how to automate tasks using Gulp and Jenkins. Next, you'll get acquainted with database integration for different platforms, such as MongoDB and PostgreSQL. Finally, you'll set up different Jenkins jobs to integrate with Node.js and C# projects, and Jenkins pipelines to make branching easier. By the end of the book, you'll have implemented Continuous Delivery and deployment from scratch. Style and approach This practical book takes a step-by-step approach to explaining all the concepts of Continuous Integration and delivery, and how it can help you deliver a high-quality product. Provides recommendations and case studies to help with the

implementation of Scrum.

Continuous delivery adds enormous value to the business and the entire software delivery lifecycle, but adopting this practice means mastering new skills typically outside of a developer's comfort zone. In this practical book, Daniel Bryant and Abraham Marín-Pérez provide guidance to help experienced Java developers master skills such as architectural design, automated quality assurance, and application packaging and deployment on a variety of platforms. Not only will you learn how to create a comprehensive build pipeline for continually delivering effective software, but you'll also explore how Java application architecture and deployment platforms have affected the way we rapidly and safely deliver new software to production environments. Get advice for beginning or completing your migration to continuous delivery Design architecture to enable the continuous delivery of Java applications Build application artifacts including fat JARs, virtual machine images, and operating system container (Docker) images Use continuous integration tooling like Jenkins, PMD, and find-sec-bugs to automate code quality checks Create a comprehensive build pipeline and design software to separate the deploy and release

processes Explore why functional and system quality attribute testing is vital from development to delivery Learn how to effectively build and test applications locally and observe your system while it runs in production
This course teaches concepts by deep-dive on-hand exercises. Throughout the course, you will learn the required toolset by using both on-premise, open-source, and hosted cloud solutions. You'll find checklists, best practices, and critical points mentioned throughout the lessons, making things more interesting. Key Features Explains in detail cloud-native continuous integration and delivery Demonstrates how to run a build in a CI/CD system Shows continuous delivery to Docker Registry and continuous deployment to Kubernetes Book Description Cloud-native software development is based on developing distributed applications focusing on speed, stability, and high availability. With this paradigm shift, software development has changed substantially and converted into a more agile environment where distributed teams develop distributed applications. In addition, the environment where the software is built, tested and deployed has changed from bare-metal servers to cloud systems. In this course, the new concepts of cloud-native

Continuous Integration and Delivery are discussed in depth. Cloud-native tooling and services such as cloud providers (AWS, Google Cloud) containerization with Docker, container-orchestrators such as Kubernetes will be a part of this course to teach how to analyze and design modern software delivery pipelines. What you will learn Learn the basics of DevOps patterns for cloud-native architecture Learn the cloud-native way of designing CI/CD systems Create multi-stage builds and tests for Docker. Apply the best practices for Docker container images Experiment using GitLab CI/CD pipelines for continuous integration Build and test their applications on cloud Learn how to continuously deliver to Docker registry Learn how to continuously deploy to Kubernetes Experiment using GitLab CI/CD pipelines for Continuous Delivery Configure and deploy software to Kubernetes using Helm Who this book is for This book is ideal for professionals interested in cloud-native software development. To benefit the most from this book, you must be familiar with developing, building, testing, integrating, and deploying containerized microservices into cloud systems.
A beginner's guide to implementing Continuous Integration and Continuous Delivery using Jenkins About This Book Speed up and

increase software productivity and software delivery using Jenkins Automate your build, integration, release, and deployment processes with Jenkins—and learn how continuous integration (CI) can save you time and money Explore the power of continuous delivery using Jenkins through powerful real-life examples Who This Book Is For This book is for anyone who wants to exploit the power of Jenkins. This book servers a great starting point for those who are in the field DevOps and would like to leverage the benefits of CI and continuous delivery in order to increase productivity and reduce delivery time. What You Will Learn Take advantage of a continuous delivery solution to achieve faster software delivery Speed up productivity using a continuous Integration solution through Jenkins Understand the concepts of CI and continuous delivery Orchestrate many DevOps tools using Jenkins to automate builds, releases, deployment, and testing Explore the various features of Jenkins that make DevOps activities a piece of cake Configure multiple build machines in Jenkins to maintain load balancing Manage users, projects, and permissions in Jenkins to ensure better security Leverage the power of plugins in Jenkins In Detail In past few years, Agile software development has seen

tremendous growth across the world. There is huge demand for software delivery solutions that are fast yet flexible to frequent amendments. As a result, CI and continuous delivery methodologies are gaining popularity. Jenkins' core functionality and flexibility allows it to fit in a variety of environments and can help streamline the development process for all stakeholders. This book starts off by explaining the concepts of CI and its significance in the Agile world with a whole chapter dedicated to it. Next, you'll learn to configure and set up Jenkins. You'll gain a foothold in implementing CI and continuous delivery methods. We dive into the various features offered by Jenkins one by one exploiting them for CI. After that, you'll find out how to use the built-in pipeline feature of Jenkins. You'll see how to integrate Jenkins with code analysis tools and test automation tools in order to achieve continuous delivery. Next, you'll be introduced to continuous deployment and learn to achieve it using Jenkins. Through this book's wealth of best practices and real-world tips, you'll discover how easy it is to implement a CI service with Jenkins. Style and approach This is a step-by-step guide to setting up a CI and continuous delivery system loaded with hands-on examples

The DevOps 2. 0 Toolkit
DevOps: Continuous Delivery, Integration, and Deployment with DevOps
Continuous Integration and Continuous Deployment using AWS services
Camel Design Patterns
Continuous Integration Vs. Continuous Delivery Vs. Continuous Deployment, 2nd Edition
The complete guide to accelerate collaboration with Jenkins, Kubernetes, Terraform and Azure DevOps
Build and release quality software at scale with Jenkins, Travis CI, and CircleCI
Improving Software Quality and Reducing Risk
Continuous Practices: A Strategic Approach to Accelerating the Software Production System
iOS 8 Swift Programming Cookbook
Build, test, and deploy cloud-native applications in the cloud-native way
Reusable Object Models

For any software developer who has spent days in "integration hell," cobbling

together myriad software components, Continuous Integration: Improving Software Quality and Reducing Risk illustrates how to transform integration from a necessary evil into an everyday part of the development process. The key, as the authors show, is to integrate regularly and often using continuous integration (CI) practices and techniques. The authors first examine the concept of CI and its practices from the ground up and then move on to explore other effective processes performed by CI systems, such as database integration, testing, inspection, deployment, and feedback. Through more than forty CI-related practices using application examples in different languages, readers learn that CI leads to more rapid software development, produces deployable software at every step in the development lifecycle, and reduces the time between defect introduction and detection, saving time and lowering costs. With successful implementation of CI, developers reduce risks and repetitive manual processes, and teams receive better project visibility. The book covers How to make integration a "non-event" on your software development projects How to reduce the amount of repetitive processes you perform when building your software Practices and techniques for using CI effectively with your teams Reducing the risks of late defect discovery, low-quality software, lack of visibility, and lack of deployable software Assessments of different CI servers and related tools on the market The book's companion Web site, www.integratebutton.com, provides updates and code examples.

Automate release processes, deployment, and continuous integration of your

application as well as infrastructure automation with the powerful services offered by AWS About This Book Accelerate your infrastructure's productivity by implementing a continuous delivery pipeline within your environment Leverage AWS services and Jenkins 2.0 to perform complete application deployments on Linux servers This recipe-based guide that will help you minimize application deployment downtime Who This Book Is For This book is for developers and system administrators who are responsible for hosting their application and managing instances in AWS. It's also ideal for DevOps engineers looking to provide continuous integration, deployment, and delivery. A basic understanding of AWS, Jenkins, and some scripting knowledge is needed. What You Will Learn Build a sample Maven and NodeJS Application using CodeBuild Deploy the application in EC2/Auto Scaling and see how CodePipeline helps you integrate AWS services Build a highly scalable and fault tolerant CI/CD pipeline Achieve the CI/CD of a microservice architecture application in AWS ECS using CodePipeline, CodeBuild, ECR, and CloudFormation Automate the provisioning of your infrastructure using CloudFormation and Ansible Automate daily tasks and audit compliance using AWS Lambda Deploy microservices applications on Kubernetes using Jenkins Pipeline 2.0 In Detail AWS CodeDeploy, AWS CodeBuild, and CodePipeline are scalable services offered by AWS that automate an application's build and deployment pipeline. In order to deliver tremendous speed and agility, every organization is moving toward automating an entire application pipeline. This book will cover all the AWS services required to automate your deployment

to your instances. You'll begin by setting up and using one of the AWS services for automation – CodeCommit. Next, you'll learn how to build a sample Maven and NodeJS Application using CodeBuild. After you've built the application, you'll see how to use CodeDeploy to deploy the application in EC2/Autoscaling. You'll also build a highly scalable and fault tolerant continuous integration (CI)/continuous deployment (CD) pipeline using some easy-to-follow recipes. Following this, you'll achieve CI/CD for Microservices application and reduce the risk within your software development lifecycle. You'll also learn to set up an infrastructure using CloudFormation Template and Ansible, and see how to automate AWS resources using AWS Lambda. Finally, you'll learn to automate instances in AWS and automate the deployment lifecycle of applications.By the end of this book, you'll be able to minimize application downtime and implement CI/CD, gaining total control over your software development lifecycle. Style and approach This book takes a "How to do it" approach, providing with easy solutions to automate common maintenance and deployment tasks.

Configure and extend Jenkins to architect, build, and automate efficient software delivery pipelines About This Book Configure and horizontally scale a Jenkins installation to support a development organization of any size Implement Continuous Integration, Continuous Delivery, and Continuous Deployment solutions in Jenkins A step-by-step guide to help you get the most out of the powerful automation orchestration platform that is Jenkins Who This Book Is For If you are a novice or intermediate-level Jenkins user who has used Jenkins before

but are not familiar with architecting solutions and implementing it in your organization, then this is the book for you. A basic understanding of the core elements of Jenkins is required to make the best use of this book. What You Will Learn Create and manage various types of build jobs, and implement automation tasks to support a software project of any kind Get to grips with the automated testing architecture, and scalable automated testing techniques Facilitate the delivery of software across the SDLC by creating scalable automated deployment solutions Manage scalable automation pipelines in Jenkins using the latest build, test, and deployment strategies Implement a scalable master / slave build automation platform, which can support Windows, Mac OSX, and Linux software solutions Cover troubleshooting and advanced configuration techniques for Jenkins slave nodes Support a robust build and delivery system by implementing basic infrastructure as code solutions in configuration management tools such as Ansible In Detail With the software industry becoming more and more competitive, organizations are now integrating delivery automation and automated quality assurance practices into their business model. Jenkins represents a complete automation orchestration system, and can help converge once segregated groups into a cohesive product development and delivery team. By mastering the Jenkins platform and learning to architect and implement Continuous Integration, Continuous Delivery, and Continuous Deployment solutions, your organization can learn to outmanoeuvre and outpace the competition. This book will equip you with the best practices to implement

advanced continuous delivery and deployment systems in Jenkins. The book begins with giving you high-level architectural fundamentals surrounding Jenkins and Continuous Integration. You will cover the different installation scenarios for Jenkins, and see how to install it as a service, as well as the advanced XML configurations. Then, you will proceed to learn more about the architecture and implementation of the Jenkins Master/Save node system, followed by creating and managing Jenkins build jobs effectively. Furthermore, you'll explore Jenkins as an automation orchestration system, followed by implementing advanced automated testing techniques. The final chapters describe in depth the common integrations to Jenkins from third-party tools such as Jira, Artifactory, Amazon EC2, and getting the most out of the Jenkins REST-based API. By the end of this book, you will have all the knowledge necessary to be the definitive resource for managing and implementing advanced Jenkins automation solutions for your organization. Style and approach This book is a step-by-step guide to architecting and implementing automated build solutions, automated testing practices, and automated delivery methodologies. The topics covered are based on industry-proven techniques, and are explained in a simple and easy to understand manner. Get past the myths of testing in agile environments - and implement agile testing the RIGHT way. * * For everyone concerned with agile testing: developers, testers, managers, customers, and other stakeholders. * Covers every key issue: Values, practices, organizational and cultural challenges, collaboration, metrics, infrastructure, documentation, tools, and more. * By two of the world's most

experienced agile testing practitioners and consultants. Software testing has always been crucial, but it may be even more crucial in agile environments that rely heavily on repeated iterations of software capable of passing tests. There are, however, many myths associated with testing in agile environments. This book helps agile team members overcome those myths -- and implement testing that truly maximizes software quality and value. Long-time agile testers Lisa Crispin and Janet Gregory offer powerful insights for three large, diverse groups of readers: experienced testers who are new to agile; members of newly-created agile teams who aren't sure how to perform testing or work with testers; and test/QA managers whose development teams are implementing agile. Readers will learn specific agile testing practices and techniques that can mean the difference between success and failure; discover how to transition 'traditional' test teams to agile; and learn how to integrate testers smoothly into agile teams. Drawing on extensive experience, the authors illuminate topics ranging from culture to test planning to automated tools. They cover every form of testing: business-facing tests, technology-facing tests, exploratory tests, context-driven and scenario tests, load, stability, and endurance tests, and more. Using this book's techniques, readers can improve the effectiveness and reduce the risks of any agile project or initiative.

This innovative book recognizes the need within the object-oriented community for a book that goes beyond the tools and techniques of the typical methodology book. In Analysis Patterns: Reusable Object Models, Martin Fowler focuses on the

end result of object-oriented analysis and design—the models themselves. He shares with you his wealth of object modeling experience and his keen eye for identifying repeating problems and transforming them into reusable models. Analysis Patterns provides a catalogue of patterns that have emerged in a wide range of domains including trading, measurement, accounting and organizational relationships. Recognizing that conceptual patterns cannot exist in isolation, the author also presents a series of "support patterns" that discuss how to turn conceptual models into software that in turn fits into an architecture for a large information system. Included in each pattern is the reasoning behind their design, rules for when they should and should not be used, and tips for implementation. The examples presented in this book comprise a cookbook of useful models and insight into the skill of reuse that will improve analysis, modeling and implementation.

Continuous integration, continuous delivery, and continuous deployment are key software delivery processes in a DevOps environment. But what does each one do for your product development and release cycles? In this updated report, Brent Laster explains what these terms really boil down to and how they work separately and together to help your team release software. This powerful set of disciplines, best practices, and technologies automates the integration and delivery of source code changes from inception through production. Although their implementation may vary, continuous integration, continuous delivery, and continuous deployment are necessary to ensure that software is released

frequently, reliably, and with high quality. You'll learn how: Continuous integration makes certain that individual code changes are suitable for inclusion in the code base and merged in successfully Continuous delivery assembles your product, automatically tests quality and functionality, and produces deliverables that are proven to be deployable Continuous deployment simplifies releasing the product to customers-whether it's in the cloud, via download, or in some other format-while also allowing for limited deployments or rolling deployments back This valuable resource for business professionals, software engineering managers, senior developers, and architects will also explore how containers and Kubernetes interact in this environment.

Learn continuous deployment and automation with code-signing, continuous testing, building, deploying, and releasing of your app. Key Features A practical guide on automating your mobile development pipeline with Fastlane, Jenkins, and Slack. Build, test, run and deploy your mobile application release with this end to end guide. Implement Continuous Integration, delivery, and deployment practices to optimize your application development workflow for faster and efficient release builds. Book Description Competitive mobile apps depend strongly on the development team's ability to deliver successful releases, consistently and often. Although continuous integration took a more mainstream priority among the development industry, companies are starting to realize the importance of continuity beyond integration and testing. This book starts off with a brief introduction to fastlane—a robust command-line tool that enables iOS and

Android developers to automate their releasing workflow. The book then explores and guides you through all of its features and utilities; it provides the reader a comprehensive understanding of the tool and how to implement them. Themes include setting up and managing your certificates and provisioning and push notification profiles; automating the creation of apps and managing the app metadata on iTunes Connect and the Apple Developer Portal; and building, distributing and publishing your apps to the App Store. You will also learn how to automate the generation of localized screenshots and mesh your continuous delivery workflow into a continuous integration workflow for a more robust setup. By the end of the book, you will gain substantial knowledge on delivering bug free, developer-independent, and stable application release cycle. What you will learn Harness the fastlane tools for the Continuous Deployment strategy Integrate Continuous Deployment with existing Continuous Integration. Automate upload of screenshots across all device screen-sizes Manage push notifications, provisioning profiles, and code-signing certificates Orchestrate automated build and deployments of new versions of your app Regulate your TestFlight users and on-board new testers Who this book is for This book is intended for mobile developers who are keen on incorporating Continuous integration and deployment practices in their workflow.

Continuous Integration
A Concise Guide with Examples
Reliable and faster software releases with automating builds, tests, and

Understand various tools and practices for building a continuous integration and delivery pipeline effectively Key Features Get up and running with the patterns of continuous integration Learn Jenkins UI for developing plugins and build an effective Jenkins pipeline Automate CI/CD with command-line tools and scripts Book Description Hands-On Continuous Integration and Delivery starts with the fundamentals of continuous integration (CI) and continuous delivery (CD) and where it fits in the DevOps ecosystem. You will explore the importance of stakeholder collaboration as part of CI/CD. As you make your way through the chapters, you will get to grips with Jenkins UI, and learn to install Jenkins on different platforms, add plugins, and write freestyle scripts. Next, you

will gain hands-on experience of developing plugins with Jenkins UI, building the Jenkins 2.0 pipeline, and performing Docker integration. In the concluding chapters, you will install Travis CI and Circle CI and carry out scripting, logging, and debugging, helping you to acquire a broad knowledge of CI/CD with Travis CI and CircleCI. By the end of this book, you will have a detailed understanding of best practices for CI/CD systems and be able to implement them with confidence. What you will learn Install Jenkins on multiple operating systems Work with Jenkins freestyle scripts, pipeline syntax, and methodology Explore Travis CI build life cycle events and multiple build languages Master the Travis CI CLI (command-line interface) and automate tasks with the CLI Use CircleCI CLI jobs and work with pipelines Automate tasks using CircleCI CLI and learn to debug and troubleshoot Learn open source tooling such as Git and GitHub Install Docker and learn concepts in shell scripting Who this book is for Hands-On Continuous Integration and Delivery is for system administrators, DevOps engineers, and build and release engineers who want to understand the concept of CI and gain hands-on experience working with prominent tools in the CI ecosystem. Basic knowledge of software delivery is an added advantage.

Automating the Continuous Deployment Pipeline with Containerized MicroservicesAbout This Book* First principles of devops, Ansible, Docker, Kubernetes, microservices* Architect your software in a better and more efficient way with

microservices packed as immutable containers* Practical guide describing an extremely modern and advanced devops toolchain that can be improved continuouslyWho This Book Is ForIf you are an intermediate-level developer who wants to master the whole microservices development and deployment lifecycle using some of the latest and greatest practices and tools, this is the book for you. Familiarity with the basics of Devops and Continuous Deployment will be useful.What You Will Learn * Get to grips with the fundamentals of Devops* Architect efficient software in a better and more efficient way with the help of microservices* Use Docker, Kubernetes, Ansible, Ubuntu, Docker Swarm and more* Implement fast, reliable and continuous deployments with zero-downtime and ability to roll-back* Learn about centralized logging and monitoring of your cluster* Design self-healing systems capable of recovery from both hardware and software failuresIn DetailBuilding a complete modern devops toolchain requires not only the whole microservices development and a complete deployment lifecycle, but also the latest and greatest practices and tools. Victor Farcic argues from first principles how to build a devops toolchain. This book shows you how to chain together Docker, Kubernetes, Ansible, Ubuntu, and other tools to build the complete devops toolkit.Style and approach This book follows a unique, hands-on approach familiarizing you to the Devops 2.0 toolkit in a very practical manner. Although there will be a lot of theory, you won't be able to complete this book by reading it in a metro on a way to work. You'll

need to be in front of your computer and get your hands dirty.

Gain the techniques and tools that enable a smooth and efficient software development process in this quick and practical guide on Python continuous integration (CI) and continuous delivery (CD). Based on example applications, this book introduces various kinds of testing and shows you how to set up automated systems that run these tests, and install applications in different environments in controlled ways. Python Continuous Integration and Delivery tackles the technical problems related to software development that are typically glossed over in pure programming texts. After reading this book, you[]ll see that in today's fast-moving world, no software project can afford to go through development, then an integration phase of unpredictable length and complexity, and finally be shipped to the customer -- just to find out that the resulting application didn't quite fill their need. Instead, you[]ll discover that practicing continuous integration and continuous delivery reduces the risks by keeping changes small and automating otherwise painful processes. What You Will Learn Carry out various kinds of testing, including unit testing and continuous integration testing, of your Python code using Jenkins Build packages and manage repositories Incorporate Ansible and Go for automated packaging and other deployments Manage more complex and robust deployments Who This Book Is For Python programmers and operating staff that work with Python applications. Simplify your DevOps roles with DevOps tools and techniques Key Features Learn to

utilize business resources effectively to increase productivity and collaboration Leverage the ultimate open source DevOps tools to achieve continuous integration and continuous delivery (CI/CD) Ensure faster time-to-market by reducing overall lead time and deployment downtime Book Description The implementation of DevOps processes requires the efficient use of various tools, and the choice of these tools is crucial for the sustainability of projects and collaboration between development (Dev) and operations (Ops). This book presents the different patterns and tools that you can use to provision and configure an infrastructure in the cloud. You'll begin by understanding DevOps culture, the application of DevOps in cloud infrastructure, provisioning with Terraform, configuration with Ansible, and image building with Packer. You'll then be taken through source code versioning with Git and the construction of a DevOps CI/CD pipeline using Jenkins, GitLab CI, and Azure Pipelines. This DevOps handbook will also guide you in containerizing and deploying your applications with Docker and Kubernetes. You'll learn how to reduce deployment downtime with blue-green deployment and the feature flags technique, and study DevOps practices for open source projects. Finally, you'll grasp some best practices for reducing the overall application lead time to ensure faster time to market. By the end of this book, you'll have built a solid foundation in DevOps, and developed the skills necessary to enhance a traditional software delivery process using modern software delivery tools and techniques What you will learn Become well versed

with DevOps culture and its practices Use Terraform and Packer for cloud infrastructure provisioning Implement Ansible for infrastructure configuration Use basic Git commands and understand the Git flow process Build a DevOps pipeline with Jenkins, Azure Pipelines, and GitLab CI Containerize your applications with Docker and Kubernetes Check application quality with SonarQube and Postman Protect DevOps processes and applications using DevSecOps tools Who this book is for If you are a developer or a system administrator interested in understanding continuous integration, continuous delivery, and containerization with DevOps tools and techniques, this book is for you. Speed up the software delivery process and software productivity using the latest features of Jenkins Key Features Take advantage of a Continuous Integration and Continuous Delivery solution to speed up productivity and achieve faster software delivery See all the new features introduced in Jenkins 2.x, such as Pipeline as code, Multibranch pipeline, Docker Plugin, and more Learn to implement Continuous Integration and Continuous Delivery by orchestrating multiple DevOps tools using Jenkins Book Description In past few years, agile software development has seen tremendous growth. There is a huge demand for software delivery solutions that are fast yet flexible to numerous amendments. As a result, Continuous Integration (CI) and Continuous Delivery (CD) methodologies are gaining popularity. This book starts off by explaining the concepts of CI and its significance in the Agile. Next, you'll learn how to configure and set up Jenkins

in many different ways. The book exploits the concept of "pipeline as code" and various other features introduced in the Jenkins 2.x release to their full potential. We also talk in detail about the new Jenkins Blue Ocean interface and the features that help to quickly and easily create a CI pipeline. Then we dive into the various features offered by Jenkins one by one, exploiting them for CI and CD. Jenkins' core functionality and flexibility allows it to fit in a variety of environments and can help streamline the development process for all stakeholders. Next, you'll be introduced to CD and will learn how to achieve it using Jenkins. Through this book's wealth of best practices and real-world tips, you'll discover how easy it is to implement CI and CD using Jenkins. What you will learn Get to know some of the most popular ways to set up Jenkins See all the new features introduced in the latest Jenkins, such as pipeline as code, Multibranch pipeline, and more Manage users, projects, and permissions in Jenkins to ensure better security Leverage the power of plugins in Jenkins Learn how to create a CI pipeline using Jenkins Blue Ocean Create a distributed build farm using Docker and use it with Jenkins Implement CI and CD using Jenkins See the difference between CD and Continuous Deployment Understand the concepts of CI Who this book is for The book is for those with little or no previous experience with Agile or CI and CD. It's a good starting point for anyone new to this field who wants to leverage the benefits of CI and CD to increase productivity and reduce delivery time. It's ideal for Build and Release engineers, DevOps engineers, SCM

(Software Configuration Management) engineers, developers, testers, and project managers. If you‖re already using Jenkins for CI, you can take your project to the next level‖CD.

Explore the high-in demand core DevOps strategies with powerful DevOps tools such as Ansible, Jenkins, and Chef Key Features ‖Get acquainted with methodologies and tools of the DevOps framework ‖Perform continuous integration, delivery, deployment, and monitoring using DevOps tools ‖Explore popular tools such as Git, Jenkins, Maven, Gerrit, Nexus, Selenium, and so on ‖Embedded with assessments that will help you revise the concepts you have learned in this book Book Description DevOps is the most widely used software engineering culture and practice that aim sat software development and operation. Continuous integration is a cornerstone technique of DevOps that merges software code updates from developers into a shared central mainline. This book takes a practical approach and covers the tools and strategies of DevOps. It starts with familiarizing you with DevOps framework and then shows how toper form continuous delivery, integration, and deployment with DevOps. You will explore DevOps process maturity frameworks and progression models with checklist templates for each phase of DevOps. You will also be familiar with agile terminology, methodology, and the benefits accrued by an organization by adopting it. You will also get acquainted with popular tools such as Git, Jenkins ,Maven, Gerrit, Nexus, Selenium, and so on.You will learn

configuration, automation, and the implementation of infrastructure automation (Infrastructure as Code) with tools such as Chef and Ansible. This book is ideal for engineers, architects, and developers, who wish to learn the core strategies of DevOps. What you will learn ⬛Get familiar with life cycle models, maturity states, progression and best practices of DevOps frameworks ⬛Learn to set up Jenkins and integrate it with Git ⬛Know how to build jobs and perform testing with Jenkins ⬛Implement infrastructure automation (Infrastructure as Code) with tools such as Chef and Ansible ⬛Understand continuous monitoring process with tools such as Splunk and Nagios ⬛Learn how Splunk improves the code quality Who this book is for This book is for engineers, architects, and developers, who wish to learn the core strategies of DevOps.

A single dramatic software failure can cost a company millions of dollars - but can be avoided with simple changes to design and architecture. This new edition of the best-selling industry standard shows you how to create systems that run longer, with fewer failures, and recover better when bad things happen. New coverage includes DevOps, microservices, and cloud-native architecture. Stability antipatterns have grown to include systemic problems in large-scale systems. This is a must-have pragmatic guide to engineering for production systems. If you're a software developer, and you don't want to get alerts every night for the rest of your life, help is here. With a combination of case studies about huge losses - lost revenue, lost reputation, lost time, lost opportunity - and

practical, down-to-earth advice that was all gained through painful experience, this book helps you avoid the pitfalls that cost companies millions of dollars in downtime and reputation. Eighty percent of project life-cycle cost is in production, yet few books address this topic. This updated edition deals with the production of today's systems - larger, more complex, and heavily virtualized - and includes information on chaos engineering, the discipline of applying randomness and deliberate stress to reveal systematic problems. Build systems that survive the real world, avoid downtime, implement zero-downtime upgrades and continuous delivery, and make cloud-native applications resilient. Examine ways to architect, design, and build software - particularly distributed systems - that stands up to the typhoon winds of a flash mob, a Slashdotting, or a link on Reddit. Take a hard look at software that failed the test and find ways to make sure your software survives. To skip the pain and get the experience...get this book.

Analysis Patterns
Continuous Integration Vs. Continuous Delivery Vs. Continuous Deployment
Continuous Delivery
Continuous Deployment to Kubernetes: Continuously deploying applications with Jenkins to a Kubernetes cluster
Continuous Delivery in Java
Continuous Software Engineering

Continuous Delivery Pipeline - Where Does It Choke?

Distinguishing Terms and Understanding how Their Implementation Methods and Tools Differ

Continuous Integration, Delivery, and Deployment

Solutions & Examples for iOS Apps

A Practical Guide to Continuous Delivery

Learning Continuous Integration with Jenkins

Using Continuous Delivery, you can bring software into production more rapidly, with greater reliability. A Practical Guide to Continuous Delivery is a 100% practical guide to building Continuous Delivery pipelines that automate rollouts, improve reproducibility, and dramatically reduce risk. Eberhard Wolff introduces a proven Continuous Delivery technology stack, including Docker, Chef, Vagrant, Jenkins, Graphite, the ELK stack, JBehave, and Gatling. He guides you through applying these technologies throughout build, continuous integration, load testing, acceptance testing, and monitoring. Wolff's start-to-finish example projects offer the basis for your own experimentation, pilot

programs, and full-fledged deployments. A Practical Guide to Continuous Delivery is for everyone who wants to introduce Continuous Delivery, with or without DevOps. For managers, it introduces core processes, requirements, benefits, and technical consequences. Developers, administrators, and architects will gain essential skills for implementing and managing pipelines, and for integrating Continuous Delivery smoothly into software architectures and IT organizations. Understand the problems that Continuous Delivery solves, and how it solves them Establish an infrastructure for maximum software automation Leverage virtualization and Platform as a Service (PAAS) cloud solutions Implement build automation and continuous integration with Gradle, Maven, and Jenkins Perform static code reviews with SonarQube and repositories to store build artifacts Establish automated GUI and textual acceptance testing with behavior-driven design Ensure appropriate performance via capacity testing Check new features and problems with exploratory testing Minimize risk throughout automated production software rollouts Gather and

analyze metrics and logs with Elasticsearch, Logstash, Kibana (ELK), and Graphite Manage the introduction of Continuous Delivery into your enterprise Architect software to facilitate Continuous Delivery of new capabilities If you are a developer, tester, or a person in operations or Devops who wants to start practising CI, start using TeamCity or both, then this book is for you. Moreover, if you have thought about bringing CI into your team, if you are already using a CI tool and want to move to TeamCity, or if you are looking for ideal practises and techniques while implementing CI with TeamCity, this book will be useful. Entirely rewritten for Apple's Swift programming language, this updated cookbook helps you overcome the vexing issues you're likely to face when creating apps for iOS devices. You'll find hundreds of new and revised recipes for using the iOS 8 SDK, including techniques for working with Health data and HomeKit accessories, enhancing and animating graphics, storing and protecting data, sending and receiving notifications, and managing files and folders among them.

Each recipe includes sample code on GitHub that you can use right away. Use CloudKit APIs to store information in the cloud with ease Create custom keyboards and extensions Access users' health-related information with HealthKit Interact with accessories inside the user's home with HomeKit Create vibrant and lifelike user interfaces with UIKit Dynamics Use the Keychain to protect your app's data Develop location-aware and multitasking-aware apps Work with iOS 8's audio and video APIs Use Event Kit UI to manage calendars, dates, and events Take advantage of the accelerometer and the gyroscope Get working examples for implementing gesture recognizers Retrieve and manipulate contacts and groups from the Address Book Determine a camera's availability and access the Photo Library Continuous Architecture provides a broad architectural perspective for continuous delivery, and describes a new architectural approach that supports and enables it. As the pace of innovation and software releases increases, IT departments are tasked to deliver value quickly and

inexpensively to their business partners. With a focus on getting software into end-users hands faster, the ultimate goal of daily software updates is in sight to allow teams to ensure that they can release every change to the system simply and efficiently. This book presents an architectural approach to support modern application delivery methods and provide a broader architectural perspective, taking architectural concerns into account when deploying agile or continuous delivery approaches. The authors explain how to solve the challenges of implementing continuous delivery at the project and enterprise level, and the impact on IT processes including application testing, software deployment and software architecture. Covering the application of enterprise and software architecture concepts to the Agile and Continuous Delivery models Explains how to create an architecture that can evolve with applications Incorporates techniques including refactoring, architectural analysis, testing, and feedback-driven development Provides insight into incorporating modern software development when

structuring teams and organizations
This book provides essential insights on the adoption of modern software engineering practices at large companies producing software-intensive systems, where hundreds or even thousands of engineers collaborate to deliver on new systems and new versions of already deployed ones. It is based on the findings collected and lessons learned at the Software Center (SC), a unique collaboration between research and industry, with Chalmers University of Technology, Gothenburg University and Malmö University as academic partners and Ericsson, AB Volvo, Volvo Car Corporation, Saab Electronic Defense Systems, Grundfos, Axis Communications, Jeppesen (Boeing) and Sony Mobile as industrial partners. The 17 chapters present the "Stairway to Heaven" model, which represents the typical evolution path companies move through as they develop and mature their software engineering capabilities. The chapters describe theoretical frameworks, conceptual models and, most importantly, the industrial experiences gained by the partner companies in applying

novel software engineering techniques. The book's structure consists of six parts. Part I describes the model in detail and presents an overview of lessons learned in the collaboration between industry and academia. Part II deals with the first step of the Stairway to Heaven, in which R&D adopts agile work practices. Part III of the book combines the next two phases, i.e., continuous integration (CI) and continuous delivery (CD), as they are closely intertwined. Part IV is concerned with the highest level, referred to as "R&D as an innovation system," while Part V addresses a topic that is separate from the Stairway to Heaven and yet critically important in large organizations: organizational performance metrics that capture data, and visualizations of the status of software assets, defects and teams. Lastly, Part VI presents the perspectives of two of the SC partner companies. The book is intended for practitioners and professionals in the software-intensive systems industry, providing concrete models, frameworks and case studies that show the specific challenges that the partner companies

encountered, their approaches to overcoming them, and the results. Researchers will gain valuable insights on the problems faced by large software companies, and on how to effectively tackle them in the context of successful cooperation projects.

Achieve the Continuous Integration and Continuous Delivery of your web applications with ease About This Book Overcome the challenges of implementing DevOps for web applications, familiarize yourself with diverse third-party modules, and learn how to integrate them with bespoke code to efficiently complete tasks Understand how to deploy web applications for a variety of Cloud platforms such as Amazon EC2, AWS Elastic Beanstalk, Microsoft Azure, Azure Web Apps, and Docker Container Understand how to monitor applications deployed in Amazon EC2, AWS Elastic Beanstalk, Microsoft Azure, Azure Web Apps using Nagios, New Relic, Microsoft Azure, and AWS default monitoring features Who This Book Is For If you are a system admin or application and web application developer with a basic knowledge of programming and want to get hands-

on with tools such as Jenkins 2 and Chef, and Cloud platforms such as AWS and Microsoft Azure, Docker, New Relic, Nagios, and their modules to host, deploy, monitor, and manage their web applications, then this book is for you. What You Will Learn Grasp Continuous Integration for a JEE application—create and configure a build job for a Java application with Maven and with Jenkins 2.0 Create built-in delivery pipelines of Jenkins 2 and build a pipeline configuration for end-to-end automation to manage the lifecycle of Continuous Integration Get to know all about configuration management using Chef to create a runtime environment Perform instance provisioning in AWS and Microsoft Azure and manage virtual machines on different cloud platforms—install Knife plugins for Amazon EC2 and Microsoft Azure Deploy an application in Amazon EC2, AWS Elastic Beanstalk, Microsoft Azure Web Apps, and a Docker container Monitor infrastructure, application servers, web servers, and applications with the use of open source monitoring solutions and New Relic Orchestrate multiple

build jobs to achieve application deployment automation—create parameterized build jobs for end-to-end automation In Detail The DevOps culture is growing at a massive rate, as many organizations are adopting it. However, implementing it for web applications is one of the biggest challenges experienced by many developers and admins, which this book will help you overcome using various tools, such as Chef, Docker, and Jenkins. On the basis of the functionality of these tools, the book is divided into three parts. The first part shows you how to use Jenkins 2.0 for Continuous Integration of a sample JEE application. The second part explains the Chef configuration management tool, and provides an overview of Docker containers, resource provisioning in cloud environments using Chef, and Configuration Management in a cloud environment. The third part explores Continuous Delivery and Continuous Deployment in AWS, Microsoft Azure, and Docker, all using Jenkins 2.0. This book combines the skills of both web application deployment and system configuration as each chapter contains

one or more practical hands-on projects. You will be exposed to real-world project scenarios that are progressively presented from easy to complex solutions. We will teach you concepts such as hosting web applications, configuring a runtime environment, monitoring and hosting on various cloud platforms, and managing them. This book will show you how to essentially host and manage web applications along with Continuous Integration, Cloud Computing, Configuration Management, Continuous Monitoring, Continuous Delivery, and Deployment. Style and approach This is a learning guide for those who have a basic knowledge of application deployment, configuration management tools, and Cloud computing, and are eager to leverage it to implement DevOps for web applications using end-to-end automation and orchestration. Build, package, and deploy software projects, developed with any language targeting any platform, using Azure pipelines. The book starts with an overview of CI/CD and the need for software delivery automation. It further delves into the basic concepts of Azure pipelines followed by a hands-on

guide to setting up agents on all platforms enabling software development in any language. Moving forward, you will learn to set up a pipeline using the classic Visual Editor using PowerShell scripts, a REST API, building edit history, retention, and much more. You'll work with artifact feeds to store deployment packages and consume them in a build. As part of the discussion you'll see the implementation and usage of YAML (Yet Another Markup Language) build pipelines. You will then create Azure release pipelines in DevOps and develop extensions for Azure pipelines. Finally, you will learn various strategies and patterns for developing pipelines and go through some sample lessons on building and deploying pipelines. After reading Hands-on Azure Pipelines, you will be able to combine CI and CD to constantly and consistently test and build your code and ship it to any target. What You Will Learn Work with Azure build-and-release pipelines Extend the capabilities and features of Azure pipelines Understand build, package, and deployment strategies, and versioning and patterns with

Azure pipelines Create infrastructure and deployment that targets commonly used Azure platform services Build and deploy mobile applications Use quick-start Azure DevOps projects Who This Book Is For Software developers and test automation engineers who are involved in the software delivery process.

Deployment Automation
DevOps for Web Development
Release It!

Streamline software development with Jenkins, the popular Java-based open source tool that has revolutionized the way teams think about Continuous Integration (CI). This complete guide shows you how to automate your build, integration, release, and deployment processes with Jenkins—and demonstrates how CI can save you time, money, and many headaches. Ideal for developers, software architects, and project managers, Jenkins: The Definitive Guide is both a CI tutorial and a comprehensive Jenkins reference. Through its wealth of best practices and real-world tips, you'll discover how easy it is to set up a CI service with Jenkins. Learn how to install, configure, and secure your Jenkins server Organize and monitor general-purpose build jobs Integrate automated tests to verify builds, and set up code quality reporting Establish effective team notification strategies and techniques Configure build pipelines, parameterized jobs, matrix builds, and other advanced jobs Manage a farm of Jenkins servers to run distributed builds Implement automated deployment and continuous delivery Develop faster with DevOps DevOps embraces a culture of unifying the creation and distribution of technology in a way that allows for faster release cycles and more resource-efficient product updating. DevOps For Dummies provides a guidebook for those on the development or operations side in need of a primer on this way of working. Inside, DevOps evangelist Emily Freeman provides a roadmap for adopting the management and technology tools, as well as the culture changes, needed to dive head-first into DevOps. Identify your

organization's needs Create a DevOps framework Change your organizational structure Manage projects in the DevOps world DevOps For Dummies is essential reading for developers and operations professionals in the early stages of DevOps adoption. Getting started with the processes and the tools to continuously deliver high-quality softwareAbout This Book* Incorporate popular development practices to prevent messy code* Automate your build, integration, release, and deployment processes with Jenkins, Git, and Gulp?and learn how continuous integration (CI) can save you time and money* Gain an end-to-end overview of Continuous Integration using different languages (JavaScript and C#) and tools (Gulp and Jenkins)Who This Book Is ForThis book is for developers who want to understand and implement Continuous Integration and Delivery in their daily work. A basic knowledge of at least JavaScript and HTML/CSS is required. Knowing C# and SQL will come in handy. Most programmers who have programmed in a (compiled) C-like language will be able to follow along.What You Will Learn* Get to know all the aspects of Continuous Integration, Deployment, and Delivery* Find out how Git can be used in a CI environment* Set up browser tests using Karma and Selenium and unit tests using Jasmine* Use Node.js, npm, and Gulp to automate tasks such as linting, testing, and minification* Explore different Jenkins jobs to integrate with Node.js and C# projects* Perform Continuous Delivery and Deployment using Jenkins* Test and deliver a web APIn DetailThe challenge faced by many teams while implementing Continuous Deployment is that it requires the use of many tools and processes that all work together. Learning and implementing all these tools (correctly) takes a lot of time and effort, leading people to wonder whether it's really worth it. This book sets up a project to show you the different steps, processes, and tools in Continuous Deployment and the actual

problems they solve.We start by introducing Continuous Integration (CI), deployment, and delivery as well as providing an overview of the tools used in CI. You'll then create a web app and see how Git can be used in a CI environment. Moving on, you'll explore unit testing using Jasmine and browser testing using Karma and Selenium for your app. You'll also find out how to automate tasks using Gulp and Jenkins. Next, you'll get acquainted with database integration for different platforms, such as MongoDB and PostgreSQL. Finally, you'll set up different Jenkins jobs to integrate with Node.js and C# projects, and Jenkins pipelines to make branching easier.By the end of the book, you'll have implemented Continuous Delivery and deployment from scratch.Style and approachThis practical book takes a step-by-step approach to explaining all the concepts of Continuous Integration and delivery, and how it can help you deliver a high-quality product.

Over a half-million sold! The sequel, The Unicorn Project, is coming Nov 26 "Every person involved in a failed IT project should be forced to read this book."—TIM O'REILLY, Founder & CEO of O'Reilly Media "The Phoenix Project is a must read for business and IT executives who are struggling with the growing complexity of IT."—JIM WHITEHURST, President and CEO, Red Hat, Inc. Five years after this sleeper hit took on the world of IT and flipped it on it's head, the 5th Anniversary Edition of The Phoenix Project continues to guide IT in the DevOps revolution. In this newly updated and expanded edition of the bestselling The Phoenix Project, co-author Gene Kim includes a new afterword and a deeper delve into the Three Ways as described in The DevOps Handbook. Bill, an IT manager at Parts Unlimited, has been tasked with taking on a project critical to the future of the business, code named Phoenix Project. But the project is massively over budget and behind schedule. The CEO

demands Bill must fix the mess in ninety days or else Bill's entire department will be outsourced. With the help of a prospective board member and his mysterious philosophy of The Three Ways, Bill starts to see that IT work has more in common with a manufacturing plant work than he ever imagined. With the clock ticking, Bill must organize work flow streamline interdepartmental communications, and effectively serve the other business functions at Parts Unlimited. In a fast-paced and entertaining style, three luminaries of the DevOps movement deliver a story that anyone who works in IT will recognize. Readers will not only learn how to improve their own IT organizations, they'll never view IT the same way again. "This book is a gripping read that captures brilliantly the dilemmas that face companies which depend on IT, and offers real-world solutions."—JEZ HUMBLE, Co-author of Continuous Delivery, Lean Enterprise, Accelerate, and The DevOps Handbook ———— "I'm delighted at how The Phoenix Project has reshaped so many conversations in technology. My goal in writing The Unicorn Project was to explore and reveal the necessary but invisible structures required to make developers (and all engineers) productive, and reveal the devastating effects of technical debt and complexity. I hope this book can create common ground for technology and business leaders to leave the past behind, and co-create a better future together."—Gene Kim, November 2019

Unleash the combination of Docker and Jenkins in order to enhance the DevOps workflow About This Book Build reliable and secure applications using Docker containers. Create a complete Continuous Delivery pipeline using Docker, Jenkins, and Ansible. Deliver your applications directly on the Docker Swarm cluster. Create more complex solutions using multi-containers and database migrations. Who This Book Is For This book is indented to provide a

full overview of deep learning. From the beginner in deep learning and artificial intelligence to the data scientist who wants to become familiar with Theano and its supporting libraries, or have an extended understanding of deep neural nets. Some basic skills in Python programming and computer science will help, as well as skills in elementary algebra and calculus. What You Will Learn Get to grips with docker fundamentals and how to dockerize an application for the Continuous Delivery process Configure Jenkins and scale it using Docker-based agents Understand the principles and the technical aspects of a successful Continuous Delivery pipeline Create a complete Continuous Delivery process using modern tools: Docker, Jenkins, and Ansible Write acceptance tests using Cucumber and run them in the Docker ecosystem using Jenkins Create multi-container applications using Docker Compose Managing database changes inside the Continuous Delivery process and understand effective frameworks such as Cucumber and Flyweight Build clustering applications with Jenkins using Docker Swarm Publish a built Docker image to a Docker Registry and deploy cycles of Jenkins pipelines using community best practices In Detail The combination of Docker and Jenkins improves your Continuous Delivery pipeline using fewer resources. It also helps you scale up your builds, automate tasks and speed up Jenkins performance with the benefits of Docker containerization. This book will explain the advantages of combining Jenkins and Docker to improve the continuous integration and delivery process of app development. It will start with setting up a Docker server and configuring Jenkins on it. It will then provide steps to build applications on Docker files and integrate them with Jenkins using continuous delivery processes such as continuous integration, automated acceptance testing, and configuration management. Moving on you will learn how to ensure quick application deployment with

Docker containers along with scaling Jenkins using Docker Swarm. Next, you will get to know how to deploy applications using Docker images and testing them with Jenkins. By the end of the book, you will be enhancing the DevOps workflow by integrating the functionalities of Docker and Jenkins. Style and approach The book is aimed at DevOps Engineers, developers and IT Operations who want to enhance the DevOps culture using Docker and Jenkins. An exploration of continuous deployment to a Kubernetes cluster, using a wide range of Kubernetes platforms with instructions on how to develop a pipeline on a few of the most commonly used CI/CD platforms. Key Features The fifth book of DevOps expert Viktor Farcic's bestselling DevOps Toolkit series, with a discussion of the difference between continuous delivery vs. continuous deployment, and which is best for the user Guides readers through the continuous deployment process using Jenkins in a Kubernetes cluster Provides an overview of the best practices for building, testing, and deploying applications through fully automated pipelines. Book Description Building on The DevOps 2.3 Toolkit: Kubernetes, Viktor Farcic brings his latest exploration of the Docker technology as he records his journey to continuously deploying applications with Jenkins into a Kubernetes cluster. The DevOps 2.4 Toolkit: Continuously Deploying Applications with Jenkins to a Kubernetes Cluster is the latest book in Viktor Farcic's series that helps you build a full DevOps Toolkit. This book guides readers through the process of building, testing, and deploying applications through fully automated pipelines. Within this book, Viktor will cover a wide-range of emerging topics, including an exploration of continuous delivery and deployment in Kubernetes using Jenkins. It also shows readers how to perform continuous integration inside these clusters, and discusses the distribution of Kubernetes applications, as well as installing and setting up Jenkins. Work with

Viktor and dive into the creation of self-adaptive and self-healing systems within Docker. What you will learn Gain an understanding of continuous deployment Learn how to build, test, and deploy applications into Kubernetes Execute continuous integration inside containers Who this book is for Readers with an intermediate level understanding of Kubernetes and hands-on experience.

The step-by-step guide to going live with new software releases faster - reducing risk and delivering more value sooner! * *Fast, simple, repeatable techniques for deploying working code to production in hours or days, not months! *Crafting custom processes that get developers from idea to value faster than ever. *Best practices for everything from source code control to dependency management and in-production tracing. *Common obstacles to rapid release - and pragmatic solutions. In too many organizations, build, testing, and deployment processes can take six months or more. That's simply far too long for today's businesses. But it doesn't have to be that way. It's possible to deploy working code to production in hours or days after development work is complete - and Go Live presents comprehensive processes and techniques for doing so. Written by two of the world's most experienced software project leaders, this book demonstrates how to dramatically increase speed while reducing risk and improving code quality at the same time. The authors cover all facets of build, testing, and deployment, including: configuration management, source code control, release planning, auditing, compliance, integration, build automation, and more. They introduce a wide range of advanced techniques, including inproduction monitoring and tracing, dependency management, and the effective use of virtualization. For each area, they explain the issues, show how to mitigate the risks, and present best practices. Throughout, Go Live focuses on

powerful opportunities for individual improvement, clearly and simply explaining skills and techniques so they can be used every day on real projects. With this book's help, any development organization can move from idea to release faster -- and deliver far more value, far more rapidly.

Continuous Integration for the Masses
Python Continuous Integration and Delivery
Learning DevOps
Jenkins: The Definitive Guide
A Novel about IT, DevOps, and Helping Your Business Win
Software Development Using Scrum
Hands-On Continuous Integration and Delivery
Automating mobile application development and deployment for iOS and Android
Mastering Jenkins
Agile Testing
Succeeding with Agile