

Release It Design And Deploy Production Ready Software Pragmatic Programmers

Whether your company is considering serverless computing or has already made the decision to adopt this model, this practical book is for you. Author Jason Katzer shows early- and mid-career developers what's required to build and ship maintainable and scalable services using this model. With this book, you'll learn how to build a modern production system in the cloud, viewed through the lens of serverless computing. You'll discover how serverless can free you from the tedious task of setting up and maintaining systems in production. You'll also explore new ways to level up your career and design, develop, and deploy with confidence. In three parts, this book includes: The Path to Production: Examine the ins and outs of distributed systems, microservices, interfaces, and serverless architecture and patterns The Tools: Dive into monitoring, observability and alerting, logging, pipelines, automation, and deployment Concepts: Learn how to design security and privacy, how to manage quality through testing and staging, and how to plan for failure Provides information on ways to effectively design and release an application.

A lot of work is required to release an API, but the effort doesn't always pay off. Overplanning before an API matures is a wasted investment, while underplanning can lead to disaster. This practical guide provides maturity models for individual APIs and multi-API landscapes to help you invest the right human and company resources for the right maturity level at the right time. How do you balance the desire for agility and speed with the need for robust and scalable operations? Four experts from the API Academy show software architects, program directors, and product owners how to maximize the value of their APIs by managing them as products through a continuous life cycle. Learn which API decisions you need to govern and how and where to do so Design, deploy, and manage APIs using an API-as-a-product (AaaP) approach Examine ten pillars that form the foundation of API product work Learn how the continuous improvement model governs changes throughout an API's lifetime Explore the five stages of a complete API product life cycle Delve into team roles needed to design, build, and maintain your APIs Learn how to manage your API landscape—the set of APIs published by your organization

Give users the real-time experience they expect, by using Elixir and Phoenix Channels to build applications that instantly react to changes and reflect the application's true state. Learn how Elixir and Phoenix make it easy and enjoyable to create real-time applications that scale to a large number of users. Apply system design and development best practices to create applications that are easy to maintain. Gain confidence by learning how to break your applications before your users do. Deploy applications with minimized resource use and maximized performance. Real-time applications come with real challenges - persistent connections, multi-server deployment, and strict performance requirements are just a few. Don't try to solve these challenges by yourself - use a framework that handles them for you. Elixir and Phoenix Channels provide a solid foundation on which to build stable and scalable real-time applications. Build applications that thrive for years to come with the best-practices found in this book. Understand the magic of real-time communication by inspecting the WebSocket protocol in action. Avoid performance pitfalls early in the development lifecycle with a catalog of common problems and their solutions. Leverage GenStage to build a data pipeline that improves scalability. Break your application before your users do and confidently deploy them. Build a real-world project using solid application design and testing practices that help make future changes a breeze. Create distributed apps that can scale to many users with tools like Phoenix Tracker. Deploy and monitor your application with confidence and reduce outages. Deliver an exceptional real-time experience to your users, with easy maintenance, reduced operational costs, and maximized performance, using Elixir and Phoenix Channels. What You Need: You'll need Elixir 1.9+ and Erlang/OTP 22+ installed on a Mac OS X, Linux, or Windows machine. The overwhelming majority of a software system's lifespan is spent in use, not in design or implementation. So, why does conventional wisdom insist that software engineers focus primarily on the design and development of large-scale computing systems? In this collection of essays and articles, key members of Google's Site Reliability Team explain how and why their commitment to the entire lifecycle has enabled the company to successfully build, deploy, monitor, and maintain some of the largest software systems in the world. You'll learn the principles and practices that enable Google engineers to make systems more scalable, reliable, and efficient—lessons directly applicable to your organization. This book is divided into four sections: Introduction—Learn what site reliability engineering is and why it differs from conventional IT industry practices Principles—Examine the patterns, behaviors, and areas of concern that influence the work of a site reliability engineer (SRE) Practices—Understand the theory and practice of an SRE's day-to-day work: building and operating large distributed computing systems Management—Explore Google's best practices for training, communication, and meetings that your organization can use React is an open-source JavaScript library that is used for building user interfaces or UI components. This React book is designed to take you through the most valuable design patterns in React, helping you learn how to apply design patterns and best practices in real-world scenarios.

This volume constitutes the proceedings of the 8th IFIP WG 8.1 Conference on the Practice of Enterprise Modeling held in November 2015 in Valencia, Spain. The PoEM conference series started in 2008 and aims to provide a forum sharing knowledge and experiences between the academic community and practitioners from industry and the public sector. The 23 short papers accepted were carefully reviewed and selected from 72 submissions and are organized in eight sections on Evolving Enterprises, Securing Enterprises, Making Empirical Studies, Investigating Enterprise Methods, Acquiring User Information, Managing Risks and Threats, Engineering Methods, and Making Decisions in Enterprises.

Terraform has become a key player in the DevOps world for defining, launching, and managing infrastructure as code (IaC) across a variety of cloud and virtualization platforms, including AWS, Google Cloud, Azure, and more. This hands-on second edition, expanded and thoroughly updated for Terraform version 0.12 and beyond, shows you the fastest way to get up and running. Gruntwork cofounder Yevgeniy (Jim) Brikman walks you through code examples that demonstrate Terraform's simple, declarative programming language for deploying and managing infrastructure with a few commands. Veteran sysadmins, DevOps engineers, and novice developers will quickly go from Terraform basics to running a full stack that can support a massive amount of traffic and a large team of developers. Explore changes from Terraform 0.9 through 0.12, including backends, workspaces, and first-class expressions Learn how to write production-grade Terraform modules Dive into manual and automated testing for Terraform code Compare Terraform to Chef, Puppet, Ansible, CloudFormation, and Salt Stack Deploy server clusters, load balancers, and databases Use Terraform to manage the state of your infrastructure Create reusable infrastructure with Terraform modules Use advanced Terraform syntax to achieve zero-downtime deployment

Set up complete CI and CD pipelines for your serverless applications using DevOps principles Key Features Understand various services for designing serverless architecture Build CD pipelines using various cloud providers for your serverless applications Implement DevOps best practices when building serverless applications Book Description Serverless applications are becoming very popular among developers and are generating a buzz in the tech market. Many organizations struggle with the effective implementation of DevOps with serverless applications. DevOps for Serverless Applications takes you through different DevOps-related scenarios to give you a solid foundation in serverless deployment. You will start by understanding the concepts of serverless architecture and development, and why they are important. Then, you will get to grips with the DevOps ideology and gain an understanding of how it fits into the Serverless Framework. You'll cover deployment framework building and deployment with CI and CD pipelines for serverless applications. You will also explore log management and issue reporting in the serverless environment. In the concluding chapters, you will learn important security tips and best practices for secure pipeline management. By the end of this book, you will be in a position to effectively build a complete CI and CD delivery pipeline with log management for serverless applications. What you will learn Explore serverless fundamentals and effectively combine them with DevOps

Set up CI and CD with AWS Lambda and other popular Serverless service providers with the help of the Serverless Framework Perform monitoring and logging with serverless applications Set up a dynamic dashboard for different service providers Discover best practices for applying DevOps to serverless architecture Understand use cases for different serverless architectures Who this book is for DevOps for Serverless Applications is for DevOps engineers, architects, or anyone interested in understanding the DevOps ideology in the serverless world. You will learn to use DevOps with serverless and apply continuous integration, continuous delivery, testing, logging, and monitoring with serverless.

APIs are transforming the business world at an increasing pace. Gain the essential skills needed to quickly design, build, and deploy quality web APIs that are robust, reliable, and resilient. Go from initial design through prototyping and implementation to deployment of mission-critical APIs for your organization. Test, secure, and deploy your API with confidence and avoid the "release into production" panic. Tackle just about any API challenge with more than a dozen open-source utilities and common programming patterns you can apply right away. Good API design means starting with the API-First principle - understanding who is using the API and what they want to do with it - and applying basic design skills to match customers' needs while solving business-critical problems. Use the Sketch-Design-Build method to create reliable and scalable web APIs quickly and easily without a lot of risk to the day-to-day business operations. Create clear sequence diagrams, accurate specifications, and machine-readable API descriptions all reviewed, tested, and ready to turn into fully-functional NodeJS code. Create reliable test collections with Postman and implement proper identity and access control security with AuthO-without added cost or risk to the company. Deploy all of this to Heroku using a continuous delivery approach that pushes secure, well-tested code to your public servers ready for use by both internal and external developers. From design to code to test to deployment, unlock hidden business value and release stable and scalable web APIs that meet customer needs and solve important business problems in a consistent and reliable manner. 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.

Get started building custom Fiori applications for your enterprise. This book teaches you how to design, build, and deploy enterprise-ready, custom Fiori applications in SAP HANA. Tips and tricks collected from projects using Fiori applications (built consuming OData models and REST APIs) and integrating third-party JS libraries are presented. Also included are examples using Fiori templates from different tools such as the SAP Web IDE and the new Visual Studio Code extensions. This book explains the 5 design principles that all Fiori applications are built upon: Role-based, Responsive, Coherent, Simple, and Delightful. The book expands on consuming OData services and REST APIs internal and external to SAP HANA. The Fiori application exercise demonstrates the use of the MVC pattern, JavaScript modularization, reuse of SAP UI5 controls, debugging, and the tools required for a complete scenario. The book closes with an exercise showcasing a finished single page application with multiple views and layouts, navigation between the views, and deployment of the application to AWS. This book is simple enough for entry-level developers getting started in web frameworks but also highlights integration points from the data models being consumed from the application, and shows how the application communicates with back-end services, resulting in a complete front-end custom Fiori application. What You Will Learn Know the 5 Fiori design principles Understand how to consume OData and REST API models Apply the MVC pattern using XML views and the SAP UI5 controls along with controller behavior in JavaScript Debug and deploy the application Who This Book is For Web developers and application leads who have some experience in JavaScript frameworks and web development and understand web protocol communication

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

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.

What others in the trenches say about The Pragmatic Programmer... "The cool thing about this book is that it's great for keeping the

programming process fresh. The book helps you to continue to grow and clearly comes from people who have been there.” —Kent Beck, author of *Extreme Programming Explained: Embrace Change* “I found this book to be a great mix of solid advice and wonderful analogies!” —Martin Fowler, author of *Refactoring and UML Distilled* “I would buy a copy, read it twice, then tell all my colleagues to run out and grab a copy. This is a book I would never loan because I would worry about it being lost.” —Kevin Ruland, Management Science, MSG-Logistics “The wisdom and practical experience of the authors is obvious. The topics presented are relevant and useful.... By far its greatest strength for me has been the outstanding analogies—tracer bullets, broken windows, and the fabulous helicopter-based explanation of the need for orthogonality, especially in a crisis situation. I have little doubt that this book will eventually become an excellent source of useful information for journeymen programmers and expert mentors alike.” —John Lakos, author of *Large-Scale C++ Software Design* “This is the sort of book I will buy a dozen copies of when it comes out so I can give it to my clients.” —Eric Vought, Software Engineer “Most modern books on software development fail to cover the basics of what makes a great software developer, instead spending their time on syntax or technology where in reality the greatest leverage possible for any software team is in having talented developers who really know their craft well. An excellent book.” —Pete McBreen, Independent Consultant “Since reading this book, I have implemented many of the practical suggestions and tips it contains. Across the board, they have saved my company time and money while helping me get my job done quicker! This should be a desktop reference for everyone who works with code for a living.” —Jared Richardson, Senior Software Developer, iRenaissance, Inc. “I would like to see this issued to every new employee at my company...” —Chris Cleeland, Senior Software Engineer, Object Computing, Inc. “If I’m putting together a project, it’s the authors of this book that I want. . . . And failing that I’d settle for people who’ve read their book.” —Ward Cunningham

Straight from the programming trenches, *The Pragmatic Programmer* cuts through the increasing specialization and technicalities of modern software development to examine the core process—taking a requirement and producing working, maintainable code that delights its users. It covers topics ranging from personal responsibility and career development to architectural techniques for keeping your code flexible and easy to adapt and reuse. Read this book, and you’ll learn how to Fight software rot; Avoid the trap of duplicating knowledge; Write flexible, dynamic, and adaptable code; Avoid programming by coincidence; Bullet-proof your code with contracts, assertions, and exceptions; Capture real requirements; Test ruthlessly and effectively; Delight your users; Build teams of pragmatic programmers; and Make your developments more precise with automation. Written as a series of self-contained sections and filled with entertaining anecdotes, thoughtful examples, and interesting analogies, *The Pragmatic Programmer* illustrates the best practices and major pitfalls of many different aspects of software development. Whether you’re a new coder, an experienced programmer, or a manager responsible for software projects, use these lessons daily, and you’ll quickly see improvements in personal productivity, accuracy, and job satisfaction. You’ll learn skills and develop habits and attitudes that form the foundation for long-term success in your career. You’ll become a Pragmatic Programmer.

Augmented reality (AR) and virtual reality (VR) provide flexibility in education and have become widely used for the promotion of multimedia learning. This use coincides with mobile devices becoming prevalent, VR devices becoming more affordable, and the creation of user-friendly software that allows the development of AR/VR applications by non-experts. However, because the integration of AR and VR into education is a fairly new practice that is only in its initial stage, these processes and outcomes need to be improved. *Designing, Deploying, and Evaluating Virtual and Augmented Reality in Education* is an essential research book that presents current practices and procedures from different technology-implementation stages (design, deployment, and evaluation) to help educators use AR/VR applications in their own teaching practices. The book provides comprehensive information on AR and VR applications in different educational settings from various perspectives including but not limited to mobile learning, formal/informal learning, and integration strategies with practical and/or theoretical implications. Barriers and challenges to their implementation that are currently faced by educators are also addressed. This book is ideal for academicians, instructors, curriculum designers, policymakers, instructional designers, researchers, education professionals, practitioners, and students.

Summary Microservices in Action is a practical book about building and deploying microservice-based applications. Written for developers and architects with a solid grasp of service-oriented development, it tackles the challenge of putting microservices into production. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Invest your time in designing great applications, improving infrastructure, and making the most out of your dev teams. Microservices are easier to write, scale, and maintain than traditional enterprise applications because they’re built as a system of independent components. Master a few important new patterns and processes, and you’ll be ready to develop, deploy, and run production-quality microservices. About the Book *Microservices in Action* teaches you how to write and maintain microservice-based applications. Created with day-to-day development in mind, this informative guide immerses you in real-world use cases from design to deployment. You’ll discover how microservices enable an efficient continuous delivery pipeline, and explore examples using Kubernetes, Docker, and Google Container Engine. What’s inside An overview of microservice architecture Building a delivery pipeline Best practices for designing multi-service transactions and queries Deploying with containers Monitoring your microservices About the Reader Written for intermediate developers familiar with enterprise architecture and cloud platforms like AWS and GCP. About the Author Morgan Bruce and Paulo A. Pereira are experienced engineering leaders. They work daily with microservices in a production environment, using the techniques detailed in this book. Table of Contents PART 1 - The lay of the land Designing and running microservices Microservices at SimpleBank PART 2 - Design Architecture of a microservice application Designing new features Transactions and queries in microservices Designing reliable services Building a reusable microservice framework PART 3 - Deployment Deploying microservices Deployment with containers and schedulers Building a delivery pipeline for microservices PART 4 - Observability and ownership Building a monitoring system Using logs and traces to understand behavior Building microservice teams A comprehensive guide with basic to advanced SRE practices and hands-on examples. KEY FEATURES ? Demonstrates how to execute site reliability engineering along with fundamental concepts. ? Illustrates real-world examples and successful techniques to put SRE into production. ? Introduces you to DevOps, advanced techniques of SRE, and popular tools in use. DESCRIPTION Hands-on Site Reliability Engineering (SRE) brings you a tailor-made guide to learn and practice the essential activities for the smooth functioning of enterprise systems, right from designing to the deployment of enterprise software programs and extending to scalable use with complete efficiency and reliability. The book explores the fundamentals around SRE and related terms, concepts, and techniques that are used by SRE teams and experts. It discusses the essential elements of an IT system, including microservices, application architectures, types of software deployment, and concepts like load balancing. It explains the best techniques in delivering timely software releases using containerization and CI/CD pipeline. This book covers how to track and monitor application performance using Grafana, Prometheus, and Kibana along with how to extend monitoring more effectively by building full-stack observability into the system. The book also talks about chaos engineering, types of system failures, design for high-availability, DevSecOps and AIOps. WHAT YOU WILL LEARN ? Learn the best techniques and practices for building and running reliable software. ? Explore observability and popular methods for effective monitoring of applications. ? Workaround SLIs, SLOs, Error Budgets, and Error Budget Policies to manage failures. ? Learn to practice continuous software delivery using blue/green and canary deployments. ? Explore chaos engineering, SRE best practices, DevSecOps and AIOps. WHO THIS BOOK IS FOR This book caters to experienced IT professionals, application developers, software engineers, and all those who are looking to develop SRE capabilities at the individual or team level. TABLE OF CONTENTS 1. Understand the World of IT 2. Introduction to DevOps 3. Introduction to SRE 4. Identify and Eliminate Toil 5. Release Engineering 6. Incident Management 7. IT Monitoring 8. Observability 9. Key SRE KPIs: SLAs, SLOs,

SLIs, and Error Budgets 10. Chaos Engineering 11. DevSecOps and AIOps 12. Culture of Site Reliability Engineering

One of the biggest challenges for organizations that have adopted microservice architecture is the lack of architectural, operational, and organizational standardization. After splitting a monolithic application or building a microservice ecosystem from scratch, many engineers are left wondering what's next. In this practical book, author Susan Fowler presents a set of microservice standards in depth, drawing from her experience standardizing over a thousand microservices at Uber. You'll learn how to design microservices that are stable, reliable, scalable, fault tolerant, performant, monitored, documented, and prepared for any catastrophe. Explore production-readiness standards, including: Stability and Reliability: develop, deploy, introduce, and deprecate microservices; protect against dependency failures Scalability and Performance: learn essential components for achieving greater microservice efficiency Fault Tolerance and Catastrophe Preparedness: ensure availability by actively pushing microservices to fail in real time Monitoring: learn how to monitor, log, and display key metrics; establish alerting and on-call procedures Documentation and Understanding: mitigate tradeoffs that come with microservice adoption, including organizational sprawl and technical debt

Get the knowledge you need to deploy a top-quality Exchange service The latest release of Microsoft's messaging system allows for easier access to e-mail, voicemail, and calendars from a variety of devices and any location while also giving users more control and freeing up administrators to perform more critical tasks. This innovative new field guide starts with key concepts of Microsoft Exchange Server 2013 and then moves through the recommended practices and processes that are necessary to deploy a top-quality Exchange service. Focuses on the Exchange ecosystem rather than just the features and functions of the Exchange product Focuses on scenarios facing real customers and explains how problems can be solved and requirements met Zooms in on both on-premises deployments as well as Exchange Online cloud deployments with Office 365 Helps you thoroughly master the new version with step-by-step instruction on how to install, configure, and manage this multifaceted collaboration system Whether you're upgrading from Exchange Server 2010 or earlier, installing for the first time, or migrating from another system, this step-by-step guide provides the hands-on instruction, practical application, and real-world advice you need.

Describes ways to incorporate domain modeling into software development.

Praise for the first edition: "This excellent text will be useful to every system engineer (SE) regardless of the domain. It covers ALL relevant SE material and does so in a very clear, methodical fashion. The breadth and depth of the author's presentation of SE principles and practices is outstanding." –Philip Allen This textbook presents a comprehensive, step-by-step guide to System Engineering analysis, design, and development via an integrated set of concepts, principles, practices, and methodologies. The methods presented in this text apply to any type of human system -- small, medium, and large organizational systems and system development projects delivering engineered systems or services across multiple business sectors such as medical, transportation, financial, educational, governmental, aerospace and defense, utilities, political, and charity, among others. Provides a common focal point for "bridging the gap" between and unifying System Users, System Acquirers, multi-discipline System Engineering, and Project, Functional, and Executive Management education, knowledge, and decision-making for developing systems, products, or services Each chapter provides definitions of key terms, guiding principles, examples, author's notes, real-world examples, and exercises, which highlight and reinforce key SE&D concepts and practices Addresses concepts employed in Model-Based Systems Engineering (MBSE), Model-Driven Design (MDD), Unified Modeling Language (UMLTM) / Systems Modeling Language (SysMLTM), and Agile/Spiral/V-Model Development such as user needs, stories, and use cases analysis; specification development; system architecture development; User-Centric System Design (UCSD); interface definition & control; system integration & test; and Verification & Validation (V&V) Highlights/introduces a new 21st Century Systems Engineering & Development (SE&D) paradigm that is easy to understand and implement. Provides practices that are critical staging points for technical decision making such as Technical Strategy Development; Life Cycle requirements; Phases, Modes, & States; SE Process; Requirements Derivation; System Architecture Development, User-Centric System Design (UCSD); Engineering Standards, Coordinate Systems, and Conventions; et al. Thoroughly illustrated, with end-of-chapter exercises and numerous case studies and examples, Systems Engineering Analysis, Design, and Development, Second Edition is a primary textbook for multi-discipline, engineering, system analysis, and project management undergraduate/graduate level students and a valuable reference for professionals.

Winner of the Shingo Publication Award Accelerate your organization to win in the marketplace. How can we apply technology to drive business value? For years, we've been told that the performance of software delivery teams doesn't matter? that it can't provide a competitive advantage to our companies. Through four years of groundbreaking research to include data collected from the State of DevOps reports conducted with Puppet, Dr. Nicole Forsgren, Jez Humble, and Gene Kim set out to find a way to measure software delivery performance? and what drives it? using rigorous statistical methods. This book presents both the findings and the science behind that research, making the information accessible for readers to apply in their own organizations. Readers will discover how to measure the performance of their teams, and what capabilities they should invest in to drive higher performance. This book is ideal for management at every level. Bring the best out of DevOps and build, deploy, and maintain applications on AWS About This Book Work through practical examples and gain DevOps best practices to successfully deploy applications on AWS Successfully provision and operate distributed application systems and your AWS infrastructure using DevOps Perform Continuous Integration and deployment and fine-tune the way you deliver on AWS Who This Book Is For This book is for system administrators and developers who manage AWS infrastructure and environments and are planning to implement DevOps in their organizations. Those aiming for the AWS Certified DevOps Engineer certification will also find this book useful. Prior experience of operating and managing AWS environments is expected. What You Will Learn Design and deploy infrastructure as code within your AWS Virtual Private Cloud Implement Continuous Integration using AWS Services Configure EC2 instances using SaltStack Implement Continuous Deployment using Jenkins and the AWS CLI Collect important metrics and log data to gain more insight into infrastructure and applications Troubleshooting popular issues with some less known techniques using the AWS platform In Detail Knowing how to adopt DevOps in your organization is becoming an increasingly important skill for developers, whether you work for a start-up, an SMB, or an enterprise. This book will help you to drastically reduce the amount of time spent on development and increase the reliability of your software deployments on AWS using popular DevOps methods of automation. To start, you will get familiar with the concept of IaC and will learn to design, deploy, and maintain AWS infrastructure. Further on, you'll see how to design and deploy a Continuous Integration platform on AWS using either open source or AWS provided tools/services. Following on from the delivery part of the process, you will learn how to deploy a newly created, tested, and verified artefact to the AWS infrastructure without manual intervention. You will then find out what to consider in order to make the implementation of Configuration Management easier and more effective. Toward the end of the book, you will learn some tricks and tips to optimize and secure your AWS environment. By the end of the book, you will have mastered the art of implementing DevOps practices onto AWS. Style and approach This book is packed full of real-world examples demonstrating use cases that help you deploy DevOps best practices on AWS.

Annotation Over the past 10 years, distributed systems have become more fine-grained. From the large multi-million line long monolithic applications, we are now seeing the benefits of smaller self-contained services. Rather than heavy-weight, hard to change Service Oriented Architectures, we are now seeing systems consisting of collaborating microservices. Easier to change, deploy, and if required retire, organizations which are in the right position to take advantage of them are yielding significant benefits. This book takes an holistic view of the things you need to be cognizant of in order to pull this off. It covers just enough

understanding of technology, architecture, operations and organization to show you how to move towards finer-grained systems. 44 reusable patterns to develop and deploy reliable production-quality microservices-based applications, with worked examples in Java Key Features 44 design patterns for building and deploying microservices applications Drawing on decades of unique experience from author and microservice architecture pioneer Chris Richardson A pragmatic approach to the benefits and the drawbacks of microservices architecture Solve service decomposition, transaction management, and inter-service communication Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About The Book Microservices Patterns teaches you 44 reusable patterns to reliably develop and deploy production-quality microservices-based applications. This invaluable set of design patterns builds on decades of distributed system experience, adding new patterns for composing services into systems that scale and perform under real-world conditions. More than just a patterns catalog, this practical guide with worked examples offers industry-tested advice to help you design, implement, test, and deploy your microservices-based application. What You Will Learn How (and why!) to use microservices architecture Service decomposition strategies Transaction management and querying patterns Effective testing strategies Deployment patterns This Book Is Written For Written for enterprise developers familiar with standard enterprise application architecture. Examples are in Java. About The Author Chris Richardson is a Java Champion, a JavaOne rock star, author of Manning's POJOs in Action, and creator of the original CloudFoundry.com. Table of Contents Escaping monolithic hell Decomposition strategies Interprocess communication in a microservice architecture Managing transactions with sagas Designing business logic in a microservice architecture Developing business logic with event sourcing Implementing queries in a microservice architecture External API patterns Testing microservices: part 1 Testing microservices: part 2 Developing production-ready services Deploying microservices Refactoring to microservices

The First Complete Guide to DevOps for Software Architects DevOps promises to accelerate the release of new software features and improve monitoring of systems in production, but its crucial implications for software architects and architecture are often ignored. In DevOps: A Software Architect's Perspective, three leading architects address these issues head-on. The authors review decisions software architects must make in order to achieve DevOps' goals and clarify how other DevOps participants are likely to impact the architect's work. They also provide the organizational, technical, and operational context needed to deploy DevOps more efficiently, and review DevOps' impact on each development phase. The authors address cross-cutting concerns that link multiple functions, offering practical insights into compliance, performance, reliability, repeatability, and security. This guide demonstrates the authors' ideas in action with three real-world case studies: datacenter replication for business continuity, management of a continuous deployment pipeline, and migration to a microservice architecture. Comprehensive coverage includes • Why DevOps can require major changes in both system architecture and IT roles • How virtualization and the cloud can enable DevOps practices • Integrating operations and its service lifecycle into DevOps • Designing new systems to work well with DevOps practices • Integrating DevOps with agile methods and TDD • Handling failure detection, upgrade planning, and other key issues • Managing consistency issues arising from DevOps' independent deployment models • Integrating security controls, roles, and audits into DevOps • Preparing a business plan for DevOps adoption, rollout, and measurement

Learn the skills necessary to design, build, and deploy applications powered by machine learning (ML). Through the course of this hands-on book, you'll build an example ML-driven application from initial idea to deployed product. Data scientists, software engineers, and product managers—including experienced practitioners and novices alike—will learn the tools, best practices, and challenges involved in building a real-world ML application step by step. Author Emmanuel Ameisen, an experienced data scientist who led an AI education program, demonstrates practical ML concepts using code snippets, illustrations, screenshots, and interviews with industry leaders. Part I teaches you how to plan an ML application and measure success. Part II explains how to build a working ML model. Part III demonstrates ways to improve the model until it fulfills your original vision. Part IV covers deployment and monitoring strategies. This book will help you: Define your product goal and set up a machine learning problem Build your first end-to-end pipeline quickly and acquire an initial dataset Train and evaluate your ML models and address performance bottlenecks Deploy and monitor your models in a production environment

Don't engineer by coincidence-design it like you mean it! Filled with practical techniques, Design It! is the perfect introduction to software architecture for programmers who are ready to grow their design skills. Lead your team as a software architect, ask the right stakeholders the right questions, explore design options, and help your team implement a system that promotes the right -ilities. Share your design decisions, facilitate collaborative design workshops that are fast, effective, and fun-and develop more awesome software! With dozens of design methods, examples, and practical know-how, Design It! shows you how to become a software architect. Walk through the core concepts every architect must know, discover how to apply them, and learn a variety of skills that will make you a better programmer, leader, and designer. Uncover the big ideas behind software architecture and gain confidence working on projects big and small. Plan, design, implement, and evaluate software architectures and collaborate with your team, stakeholders, and other architects. Identify the right stakeholders and understand their needs, dig for architecturally significant requirements, write amazing quality attribute scenarios, and make confident decisions. Choose technologies based on their architectural impact, facilitate architecture-centric design workshops, and evaluate architectures using lightweight, effective methods. Write lean architecture descriptions people love to read. Run an architecture design studio, implement the architecture you've designed, and grow your team's architectural knowledge. Good design requires good communication. Talk about your software architecture with stakeholders using whiteboards, documents, and code, and apply architecture-focused design methods in your day-to-day practice. Hands-on exercises, real-world scenarios, and practical team-based decision-making tools will get everyone on board and give you the experience you need to become a confident software architect.

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

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.

Master the Shiny web framework—and take your R skills to a whole new level. By letting you move beyond static reports, Shiny helps you create fully interactive web apps for data analyses. Users will be able to jump between datasets, explore different subsets or facets of the data, run models with parameter values of their choosing, customize visualizations, and much more. Hadley Wickham from RStudio shows data scientists, data analysts, statisticians, and scientific researchers with no knowledge of HTML, CSS, or JavaScript how to create rich web apps from R. This in-depth guide provides a learning path that you can follow with confidence, as you go from a Shiny beginner to an expert developer who can write large, complex apps that are maintainable and performant. Get started: Discover how the major pieces of a Shiny app fit together Put Shiny in action: Explore Shiny functionality with a focus on code samples, example apps, and useful techniques Master reactivity: Go deep into the theory and practice of reactive programming and examine reactive graph components Apply best practices: Examine useful techniques for making your Shiny apps work well in production

Ship It! is a collection of tips that show the tools and techniques a successful project team has to use, and how to use them well. You'll get quick, easy-to-follow advice on modern practices: which to use, and when they should be applied. This book avoids current fashion trends and marketing hype; instead, readers find page after page of solid advice, all tried and tested in the real world. Aimed at beginning to intermediate programmers, Ship It! will show you: Which tools help, and which don't How to keep a project moving Approaches to scheduling that work How to build developers as well as product What's normal on a project, and what's not How to manage managers, end-users and sponsors Danger signs and how to fix them Few of the ideas presented here are controversial or extreme; most experienced programmers will agree that this stuff works. Yet 50 to 70 percent of all project teams in the U.S. aren't able to use even these simple, well-accepted practices effectively. This book will help you get started. Ship It! begins by introducing the common technical infrastructure that every project needs to get the job done. Readers can choose from a variety of recommended technologies according to their skills and budgets. The next sections outline the necessary steps to get software out the door reliably, using well-accepted, easy-to-adopt, best-of-breed practices that really work. Finally, and most importantly, Ship It! presents common problems that teams face, then offers real-world advice on how to solve them.

Microservices can have a positive impact on your enterprise—just ask Amazon and Netflix—but you can fall into many traps if you don't approach them in the right way. This practical guide covers the entire microservices landscape, including the principles, technologies, and methodologies of this unique, modular style of system building. You'll learn about the experiences of organizations around the globe that have successfully adopted microservices. In three parts, this book explains how these services work and what it means to build an application the Microservices Way. You'll explore a design-based approach to microservice architecture with guidance for implementing various elements. And you'll get a set of recipes and practices for meeting practical, organizational, and cultural challenges to microservice adoption. Learn how microservices can help you drive business objectives Examine the principles, practices, and culture that define microservice architectures Explore a model for creating complex systems and a design process for building a microservice architecture Learn the fundamental design concepts for individual microservices Delve into the operational elements of a microservices architecture, including containers and service discovery Discover how to handle the challenges of introducing microservice architecture in your organization

This book provides an insight into the key practical aspects and best practice of 4G-LTE network design, performance, and deployment Design, Deployment and Performance of 4G-LTE Networks addresses the key practical aspects and best practice of 4G networks design, performance, and deployment. In addition, the book focuses on the end-to-end aspects of the LTE network architecture and different deployment scenarios of commercial LTE networks. It describes the air interface of LTE focusing on the access stratum protocol layers: PDCP, RLC, MAC, and Physical Layer. The air interface described in this book covers the concepts of LTE frame structure, downlink and uplink scheduling, and detailed illustrations of the data flow across the protocol layers. It describes the details of the optimization process including performance measurements and troubleshooting mechanisms in addition to demonstrating common issues and case studies based on actual field results. The book provides detailed performance analysis of key features/enhancements such as C-DRX for Smartphones battery saving, CSFB solution to support voice calls with LTE, and MIMO techniques. The book presents analysis of LTE coverage and link budgets alongside a detailed comparative analysis with HSPA+. Practical link budget examples are provided for data and VoLTE scenarios. Furthermore, the reader is provided with a detailed explanation of capacity dimensioning of the LTE systems. The LTE capacity analysis in this book is presented in a comparative manner with reference to the HSPA+ network to benchmark the LTE network capacity. The book describes the voice options for LTE including VoIP protocol stack, IMS Single Radio Voice Call Continuity (SRVCC). In addition, key VoLTE features are presented: Semi-persistent scheduling (SPS), TTI bundling, Quality of Service (QoS), VoIP with C-DRX, Robust Header Compression (RoHC), and VoLTE Vocoders and De-Jitter buffer. The book describes several LTE and LTE-A advanced features in the evolution from Release 8 to 10 including SON, eICIC, CA, CoMP, HetNet, Enhanced MIMO, Relays, and

LBS. This book can be used as a reference for best practices in LTE networks design and deployment, performance analysis, and evolution strategy. Conveys the theoretical background of 4G-LTE networks Presents key aspects and best practice of 4G-LTE networks design and deployment Includes a realistic roadmap for evolution of deployed 3G/4G networks Addresses the practical aspects for designing and deploying commercial LTE networks. Analyzes LTE coverage and link budgets, including a detailed comparative analysis with HSPA+. References the best practices in LTE networks design and deployment, performance analysis, and evolution strategy Covers infrastructure-sharing scenarios for CAPEX and OPEX saving. Provides key practical aspects for supporting voice services over LTE, Written for all 4G engineers/designers working in networks design for operators, network deployment engineers, R&D engineers, telecom consulting firms, measurement/performance tools firms, deployment subcontractors, senior undergraduate students and graduate students interested in understanding the practical aspects of 4G-LTE networks as part of their classes, research, or projects.

Deploy serverless and scalable cloud-native applications with Jakarta EE KEY FEATURES ? Example-driven approach crafted specially for developers and architects. ? Covers all core areas for cloud-native development. ? Step-by-step implementation of core concepts, including application scalability and security, serverless, and containerization. DESCRIPTION The book helps readers to get a basic understanding of features provided by the cloud and core concepts of cloud native development. A hands-on approach makes sure that after reading the book, one can straight away implement the concepts in their daily design and development activities. The book starts with the basics of cloud computing and moves on to understanding the core concepts to create a production-ready cloud-native application. The book helps readers to develop a code that is testable and maintainable to support Agile cloud native development. This book also talks about the security and scalability aspects of applications which are the backbone of any large-scale application. The book covers advanced cloud-native application development approaches using containers and serverless approaches. The book will help readers to get ready for a cloud-native development journey. Whether one is creating a small application or a large-scale application, core concepts explained in this book remain relevant and will work as a guiding light for developers and architects. WHAT YOU WILL LEARN ? Explains the core features that are part of cloud computing. ? Build applications that are fast to market due to testability and maintainability. ? Build applications that are secured against vulnerabilities. ? Build applications that are easy to scale. WHO THIS BOOK IS FOR The book is meant for software developers, architects, and technical readers who want to learn about Cloud-based application development. Basic knowledge of the Java programming language or Jakarta EE platform is expected to understand code examples used in the book. TABLE OF CONTENTS 1. Introduction to Cloud Computing 2. Design for Cloud 3. Major Players in Cloud Computing 4. Sample Application Using Jakarta EE 5. Testing Cloud-Native Applications 6. Continuous Integration and Continuous Delivery 7. Securing Cloud-Based Applications 8. Scalability 9. Monitoring, Alerting, and Reporting 10. Containers 11. Serverless Computing 12. Best Practices for Developing Cloud-Native Applications

Increase profitability, elevate work culture, and exceed productivity goals through DevOps practices. More than ever, the effective management of technology is critical for business competitiveness. For decades, technology leaders have struggled to balance agility, reliability, and security. The consequences of failure have never been greater?whether it's the healthcare.gov debacle, cardholder data breaches, or missing the boat with Big Data in the cloud. And yet, high performers using DevOps principles, such as Google, Amazon, Facebook, Etsy, and Netflix, are routinely and reliably deploying code into production hundreds, or even thousands, of times per day. Following in the footsteps of The Phoenix Project, The DevOps Handbook shows leaders how to replicate these incredible outcomes, by showing how to integrate Product Management, Development, QA, IT Operations, and Information Security to elevate your company and win in the marketplace.

Learn to implement DevOps using Docker & Kubernetes. About This Book Learning DevOps, container, and Kubernetes within one book. Leverage Kubernetes as a platform to deploy, scale, and run containers efficiently. A practical guide towards container management and orchestration Who This Book Is For This book is targeted for anyone, who wants to learn containerization and clustering in a practical way using Kubernetes. No prerequisite skills required, however, essential DevOps skill and public/private Cloud knowledge will accelerate the reading speed. If you're advanced readers, you can also get a deeper understanding of all the tools and technique described in the book. What You Will Learn Learn fundamental and advanced DevOps skills and tools Get a comprehensive understanding for container Learn how to move your application to container world Learn how to manipulate your application by Kubernetes Learn how to work with Kubernetes in popular public cloud Improve time to market with Kubernetes and Continuous Delivery Learn how to monitor, log, and troubleshoot your application with Kubernetes In Detail Containerization is said to be the best way to implement DevOps. Google developed Kubernetes, which orchestrates containers efficiently and is considered the frontrunner in container orchestration. Kubernetes is an orchestrator that creates and manages your containers on clusters of servers. This book will guide you from simply deploying a container to administrate a Kubernetes cluster, and then you will learn how to do monitoring, logging, and continuous deployment in DevOps. The initial stages of the book will introduce the fundamental DevOps and the concept of containers. It will move on to how to containerize applications and deploy them into. The book will then introduce networks in Kubernetes. We then move on to advanced DevOps skills such as monitoring, logging, and continuous deployment in Kubernetes. It will proceed to introduce permission control for Kubernetes resources via attribute-based access control and role-based access control. The final stage of the book will cover deploying and managing your container clusters on the popular public cloud Amazon Web Services and Google Cloud Platform. At the end of the book, other orchestration frameworks, such as Docker Swarm mode, Amazon ECS, and Apache Mesos will be discussed. Style and approach Readers will be taken through fundamental DevOps skills and Kubernetes concept and administration with detailed examples. It introduces comprehensive DevOps topics, including microservices, automation tools, containers, monitoring, logging, continuous delivery, and popular public cloud environments. At each step readers will learn how to leverage Kubernetes in their everyday lives and transform their original delivery pipeline for fast and efficient delivery.

With the immense cost savings and scalability the cloud provides, the rationale for building cloud native applications is no longer in question. The real issue is how. With this practical guide, developers will learn about the most commonly used design patterns for building cloud native applications using APIs, data, events, and streams in both greenfield and brownfield development. You'll learn how to incrementally design, develop, and deploy large and effective cloud native applications that you can manage and maintain at scale with minimal cost, time, and effort. Authors Kasun Indrasiri and Sriskandarajah Suhothayan highlight use cases that effectively demonstrate the challenges you might encounter at each step. Learn the fundamentals of cloud native applications Explore key cloud native communication, connectivity, and composition patterns Learn decentralized data management

techniques Use event-driven architecture to build distributed and scalable cloud native applications Explore the most commonly used patterns for API management and consumption Examine some of the tools and technologies you'll need for building cloud native systems

[Copyright: 0c70640c0e1ab51c349411f331f3d5ce](#)