Mastering GitHub Actions : Advance Your Automation Skills with the Latest Techniques for Software Integration and Deployment /

Chapman, Eric (
1932-),
author

Monografía

Explore the full spectrum of GitHub Actions to unlock your team's potential and become a pro in no time Key Features Master GitHub events to foster a self-service mindset Elevate your GitHub Actions knowledge to a whole new level through real-world examples Learn how to integrate with popular cloud-based products within your workflows Purchase of the print or Kindle book includes a free PDF eBook Book Description Navigating GitHub Actions often leaves developers grappling with inefficiencies and collaboration bottlenecks. Mastering GitHub Actions offers solutions to these challenges, ensuring smoother software development. With 16 extensive chapters, this book simplifies GitHub Actions, walking you through its vast capabilities, from team and enterprise features to organization defaults, self-hosted runners, and monitoring tools. You'll learn how to craft reusable workflows, design bespoke templates, publish actions, incorporate external services, and introduce enhanced security measures. Through hands-on examples, you'll gain best-practice insights for team-based GitHub Actions workflows and discover strategies for maximizing organization accounts. Whether you're a software engineer or a DevOps guru, by the end of this book, you'll be adept at amplifying productivity and leveraging automation's might to refine your development process. What you will learn Explore GitHub Actions' features for team and business settings Create reusable workflows, templates, and standardized processes to reduce overhead Get to grips with CI/CD integrations, code quality tools, and communication Understand self-hosted runners for greater control of resources and settings Discover tools to optimize GitHub Actions and manage resources efficiently Work through examples to enhance projects, teamwork, and productivity Who this book is for This book is for developers with a foundation in CI/CD, code quality tools, and team communication keen on exploring GitHub Actions. It's ideal for DevOps engineers, system administrators, software developers, IT specialists, automation aficionados, and university students focused on software integration and deployment. Those familiar with GitHub's ecosystem will find this content insightful

https://rebiunoda.pro.baratznet.cloud:28443/OpacDiscovery/public/catalog/detail/b2FpOmNlbGVicmF0aW9uOmVzLmJhcmF0ei5yZW4vMzU4MTkyNjg

Título: Mastering GitHub Actions Advance Your Automation Skills with the Latest Techniques for Software Integration and Deployment Eric Chapman

Edición: First edition

Editorial: Birmingham Packt Publishing [2024] 2024

Descripción física: 1 online resource (490 pages)

Nota general: Description based upon print version of record What is HashiCorp Cloud Vault?

Contenido: Cover -- Title Page -- Copyright and Credits -- Dedication -- Contributors -- Table of Contents -- Preface -- Part 1: Centralized Workflows to Assist with Governance -- Chapter 1: An Overview of GitHub and GitHub Actions -- Technical requirements -- Exploring the GitHub platform -- Walk-through of the overview interface and features -- The different types of accounts on GitHub -- Creating a personal GitHub account -- The different types of plans on GitHub -- GitHub Free -- GitHub Team -- GitHub Enterprise -- Creating an organization account -- A brief introduction to GitHub Actions -- An overview of organization accounts -- Managing teams -- Organization defaults -- GitHub Actions -- The .github repository -- Templates and reusable workflow repositories -- Reusable workflows -- Workflow templates -- Summary -- Chapter 1: Exploring Workflows -- Technical requirements -- Exploring workflow capabilities -- Events or triggers -- Jobs -- Steps -- Features of a workflow -- Using events to trigger workflows -- Workflow structuring and good habits -- Exploring workflow jobs -- Understanding how jobs work -- Running jobs in a container -- Running services -- Summary -- Chapter 3: Deep Dive into Reusable Workflows and Composite Actions -- Technical requirements -- Setting up the CLI -- Introducing reusable workflows -- Reusable workflows versus normal workflows -- Use cases for reusable workflows -- Limitations of reusable workflows -- Understanding composite actions -- Key concepts of composite actions -- Use cases for composite actions -- Limitations of composite actions -- Creating a reusable build pipeline -- Marking a workflow as a reusable workflow -- Creating a local composite action -- Debugging techniques for workflows -- act workflow debugging tool -- Visual Studio Code GitHub Actions plugin -- Debugging event data -- Runner diagnostic logging Step debug logging -- Workflow monitoring and alerting -- GitHub notification capabilities -- Integrating GitHub Actions with Slack -- Datadog -- Summary -- Chapter 4: Workflow Personalization Using GitHub Apps -- Technical requirements -- GitHub token options -- GitHub App tokens -- GitHub personal access tokens -- Workflow tokens -- Creating a GitHub App -- Installing and managing the App's credentials -- Leveraging a GitHub App in various use cases -- Summary -- Chapter 5: Utilizing Starter Workflows in Your Team -- Technical requirements -- What are starter workflows? -- Accessing starter workflows -- Starter workflows in a private repository -- Creating our starter workflow -- Utilizing our starter workflow -- Applications of self-service reusable workflows -- Summary -- Part 2: Implementing Advanced Patterns within Actions -- Chapter 6: Using HashiCorp Vault in GitHub -- Technical requirements -- Understanding what OIDC is -- How to get an Identity token in a workflow -- OIDC in reusable workflows -- Setting up a HashiCorp Cloud Vault instance -- What is HashiCorp Cloud Vault? -- Creating a HashiCorp Cloud Platform account -- Accessing your Vault cluster -- Understanding secret engines and where secrets are stored -- Creating a secret in Vault that GitHub Actions can access -- Adding our Slack API token as a secret -- Enabling JWT authentication in HashiCorp -- Enabling JWT for GitHub-produced tokens -- What is JWKS? -- Creating a HashiCorp secret policy -- Creating a HashiCorp JWT role -- Setting up a workflow to use HashiCorp -- Exploring other security hardening techniques -- Implementing CODEOWNERS -- OIDC action recommendations -- HashiCorp actions recommendations -- Azure Key Vault -- Summary -- Chapter 7: Deploying to Azure Using OpenID Connect -- Technical requirements -- Exploring our infrastructure using Bicep -- Why Bicep? Understanding the components of Bicep -- Azure infrastructure requirements -- Deploying locally with the Azure CLI -- Understanding the folder structure -- Deploying our resource group with the Azure CLI -- Deploying our container registry with the Azure CLI -- Deploying our container instance with the Azure CLI -- Deploying infrastructure alongside our application code -- The teardown -- Adding infrastructure to our repository -- Best practices and areas for improvement -- Reusable workflows for applications and infrastructure -- Authorizing our deployments with Azure and OIDC -- Subject patterns and their limitations -- Using OIDC for Azure in the reusable workflow -- Correcting the trust -- Summary -- Chapter 8: Working with Checks -- Technical requirements -- Exploring check suites and checks -- What are checks? -- Introducing check suites -- Interacting with the Checks API -- Understanding commit statuses -- What are commit statuses? -- Interacting with commit statuses -- When should we use commit statuses over checks? -- Creating checks and check suites -- Understanding how check suites work -- Working with the GitHub API -- Creating a check -- Failing and passing checks -- Conclusions -- Playing with check outcomes -- Creating custom actions -- What makes an action an action? -- Types of actions -- Defining outputs in actions -- Let's create an action -- Putting our action to use -- Exploring further action opportunities -- Branding -- Creating/updating check runs -- Controlling check failures -- Creating checks with different tokens -- Summary -- Chapter 9: Annotating Code with Actions -- Technical requirements -- Exploring annotations within checks -- The structure of an annotation -- Creating annotations on a check run -- Understanding the validation rules for annotations -- Annotations in action Introducing annotation support for our RichChecks action -- Creating annotations from build output -- Infrastructure linting -- Displaying our results -- Creating annotations from a GitHub App using Probot -- Exploring Probot-powered apps -- The importance of a spell checker -- Summary -- Chapter 10: Advancing with Event-Driven Workflows -- Technical requirements -- Understanding GitHub events more deeply -- The core of GitHub event payloads -- Tailored automation with events -- Creating an issue from a pull request -- Subscribing to the event -- Introducing the GitHub Issues API -- Creating an issue -- Linking pull requests to issues -- Promoting your new releases -- What are GitHub releases? -- Creating a release manually -- Subscribing to the event -- Creating the communication -- Targeting published releases -- Designing a chatbot using ChatGPT -- Subscribing to the event and collecting the data -- Creating a conversation with OpenAIs APIs -- Helpful AI actions and apps -- Code Autopilot - AI coder -- OpenCommit -- Summary -- Chapter 11: Setting Up Self-Hosted Runners -- Technical requirements -- Exploring self-hosted runners -- Action runner variants -- Action runner groups -- Deploying self-hosted runners -- A local runner in action -- Exploring ARC -- How does it work? -- Scaling options -- Monitoring and troubleshooting -- Running ARC locally -- A brief overview of Kubernetes -- Understanding the importance of containers -- Setting up Minikube -- Installing Helm -- Deploying ARC on Minikube -- Using the cloud for your runs -- Setting up Kubernetes using Bicep -- Setting up a GitHub App -- Deploying the new ARC -- Advanced techniques with ARC -- Scaling in ARC -- Running within a proxy -- Runner labels -- Customizing the runner specification -- Observing our self-hosted infrastructure -- Housekeeping -- Summary Part 3: Best Practices, Patterns, Tricks, and Tips Toolkit -- Chapter 12: The Crawler Pattern -- Technical requirements -- Introducing the crawler pattern -- How does this pattern work? -- Methods of feeding the matrix -- Making bulk repository changes -- Adding permissions to all repositories -- Adding branch protection -- Making bulk content changes -- Understanding the workflow -- Rolling out content in bulk -- Summary -- Chapter 13: The Configuration Centralization Pattern -- Technical requirements -- Understanding the central configuration pattern -- Creating a repository indexer -- Schedules in GitHub workflows -- Setting up the indexer -- Hosting the results using GitHub Pages -- Configuring the repository for GitHub Pages -- Creating the action for the deployment -- Where to go next -- Summary -- Chapter 14: Using Remote Workflows to Kickstart Your Products -- Technical requirements -- Introducing repository dispatch events -- Advantages of repository dispatch events -- Requirements for creating a repository dispatch event -- Setting up a repository dispatch workflow -- Understanding product kickstarters -- A brief overview of product kickstarters -- Creating template repositories -- Building a kickstarter template -- Internal developer portal and other use cases -- Adding portal visibility for developers -- Exploring other use cases -- Summary -- Chapter 15: Housekeeping Tips for Your Organization -- Technical requirements -- Managing GitHub costs -- GitHub spending limits -- Alerting on usage -- Usage monitoring -- Workflow usage across the organization -- Optimizing uploads and downloads -- Compressible artifact action -- Methods for eliminating GitHub storage -- Useful reporting techniques for your organization -- Creating a dormant user report -- Creating a user member contribution report -- Managing your action updates with Dependabot Summary

ISBN: 1-80512-330-0

Materia Título preferido: Git (Computer file)

Materia: Computer software- Development Open source software

Enlace a formato físico adicional: 1-80512-862-0