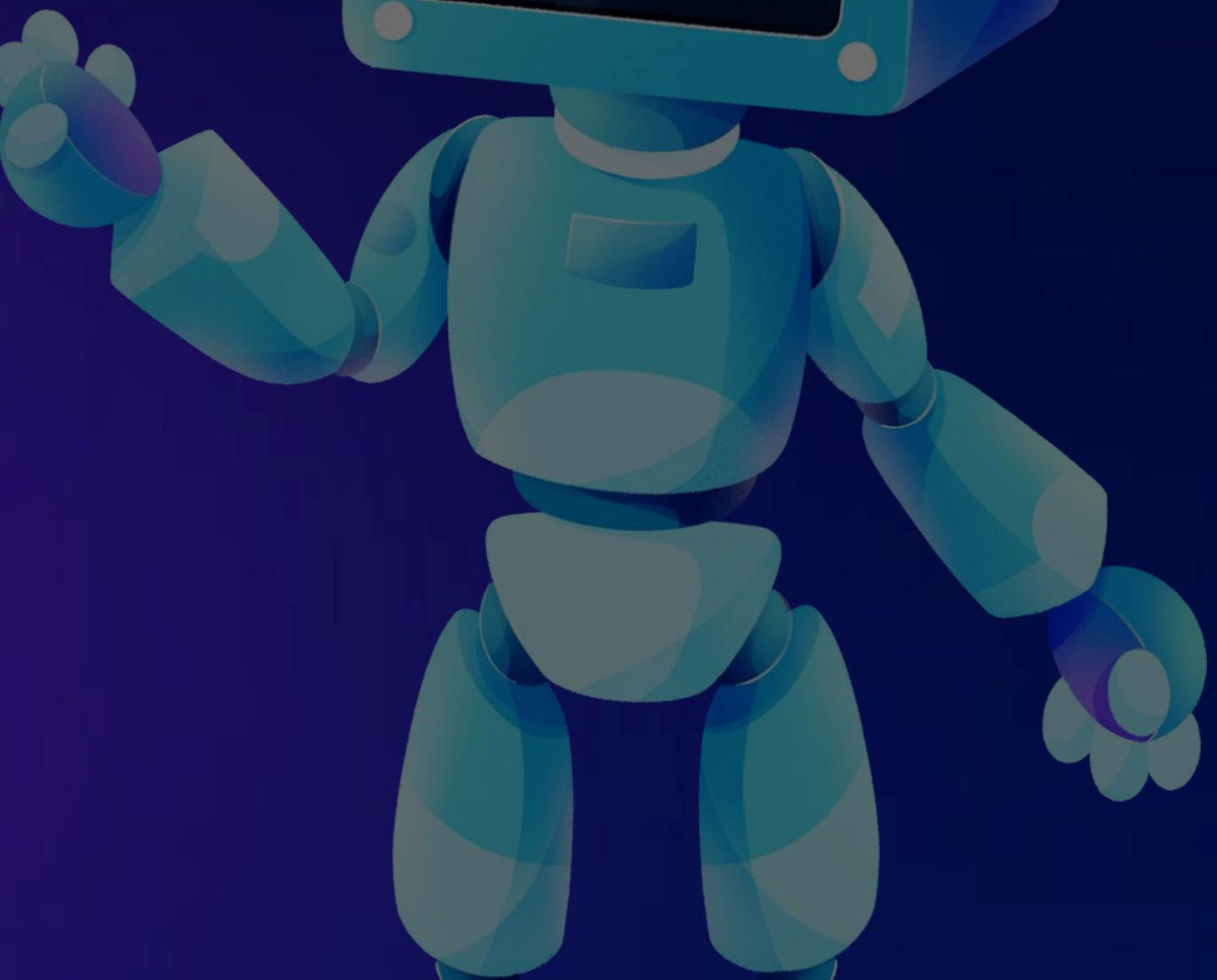
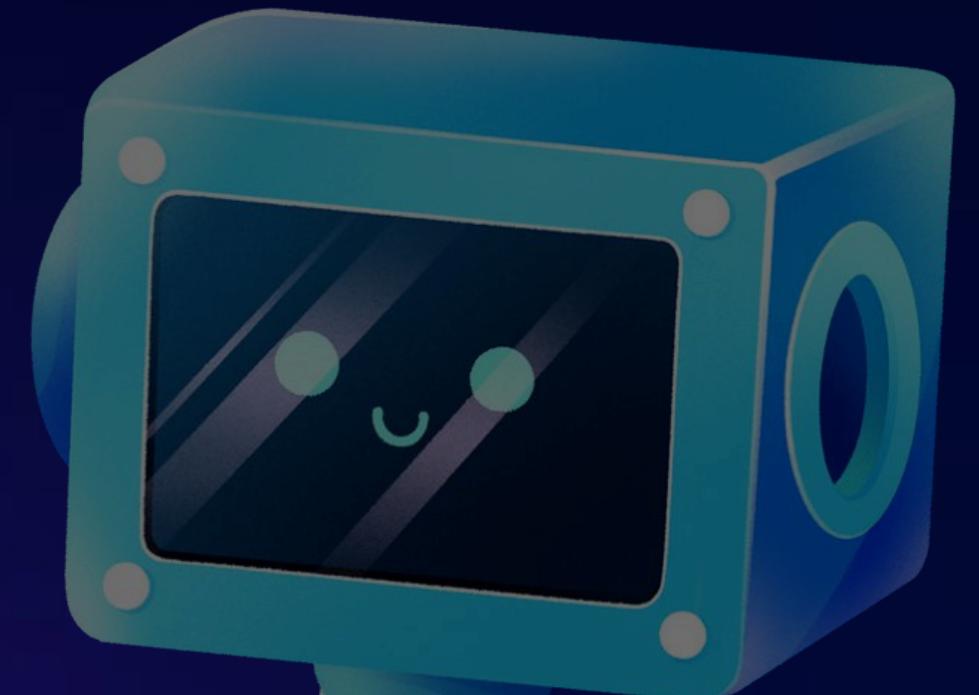


GITHUB ACTIONS

CI/CD



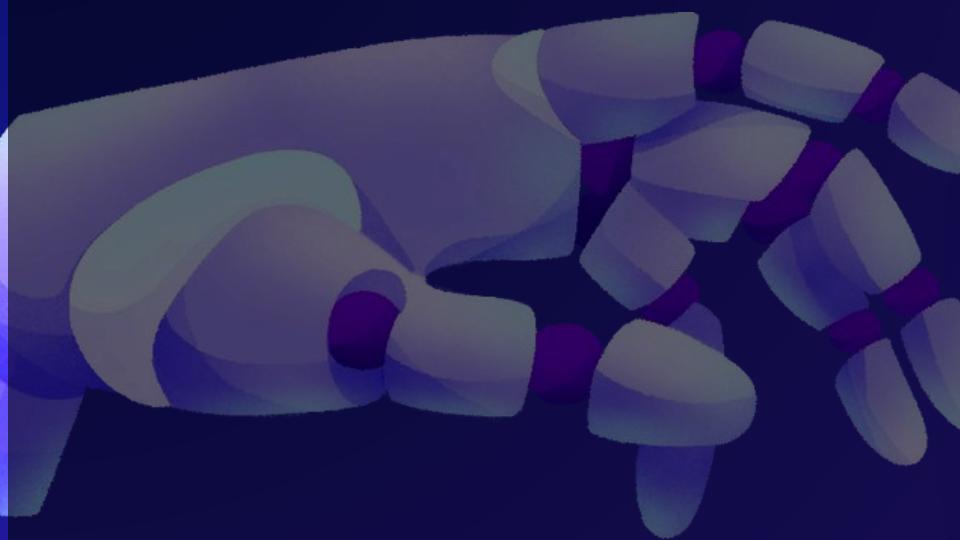


TABLE OF CONTENTS

- Introduction to CI/CD
 - Overview of GitHub Actions
 - Setting up GitHub Actions for CI/CD
 - Best practices for CI/CD with GitHub
 - Real-World Examples
 - Q&A
 - Conclusion
- 

ABOUT ME

Hillary Nyakundi



Curriculum Dev - freeCodeCamp,
Technical Writer - fCC, OpenMRS...
Community Lead - WTD-Kenya

X @larymak1

in /hillary-nyakundi

@larymak



INTRO TO CI/CD

Continuous Integration (CI) - The practice of automatically integrating code changes from multiple contributors into a shared repository several times a day.

Continuous Deployment (CD): The practice of automatically deploying every code change that passes the CI tests to a production environment.

Why CI/CD?

- **Improved Code Quality** - Regular testing and integration reduce bugs and enhance code quality.
- **Faster Time to Market** - Automated processes speed up the release cycles.
- **Enhanced Collaboration** - Teams can work more cohesively with continuous feedback.



GITHUB ACTIONS



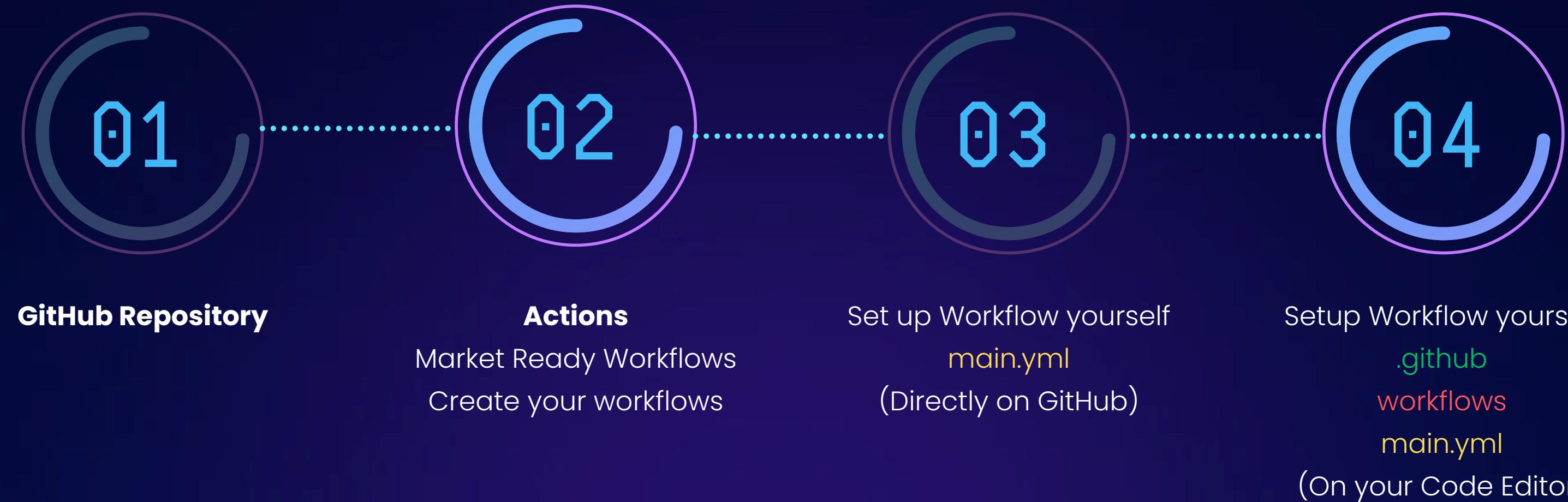
What are GitHub Actions

- GitHub Actions is a powerful CI/CD tool that allows you to automate your software workflows directly within GitHub.
- It uses YAML syntax to define workflows that can build, test, and deploy your code.

Benefits

- Integrated with GitHub
- Customizable
- Community-Driven

SETTING UP GITHUB ACTIONS FOR CI/CD



DEMO & EXAMPLES

KEEP WORKFLOWS SIMPLE AND MODULAR

Break down complex workflows into smaller, reusable steps.
Use pre-built actions from the GitHub Marketplace to avoid
reinventing the wheel.

SECURE YOUR SECRETS

Store sensitive information such as API keys and
credentials using GitHub Secrets.

BEST CODING PRACTICES

Make sure you adhere to the best coding practices so
others can easily understand and read your code.

MONITOR AND MAINTAIN

Regularly monitor your workflows and update
dependencies.
Use notifications to stay informed about build and
deployment statuses.



Q&A

RESOURCE PAGE

