

CI/CD

By: Carlos Murillo, Nicholas Dobard, and
Norman Cattles (Pod 6)



GitHub Actions

What is Continuous Integration?

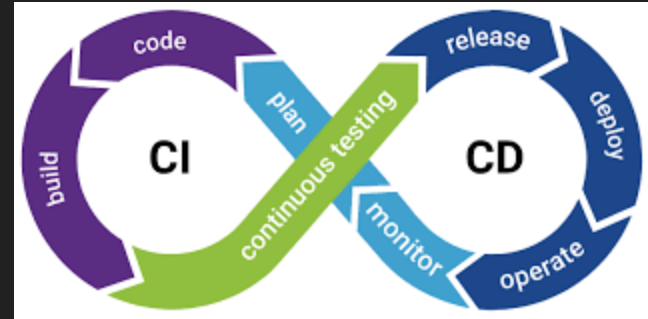
- CI is an approach to software development that automates repetitive and error-prone tasks that also promotes consistently integrated environments
- In simpler terms, it is the automated building and testing of your application on every push to a shared repository
- Continuous integrations consists of 4 main steps:
 - Commit
 - Build
 - Test
 - Inform

What is Continuous Deployment?

- The automated deployment of successful code changes from a pre-production environment into production
- If all tests pass, every new commit deploys directly to production without the need for manual intervention
- Continuous Deployment consists of 3 main steps:
 - Deployment
 - Monitoring
 - Feedback

What is a CI/CD Pipeline?

- A sequence of automated steps to deliver software from development to production
- Key Stages
 - Source: Triggered by code changes (code push or pull request)
 - Build: Compiling and packaging the application
 - Test: Running unit, integration, and system tests
 - Deploy: Releasing the application to staging or production



What are the Benefits?



Version control



Faster release
cycles

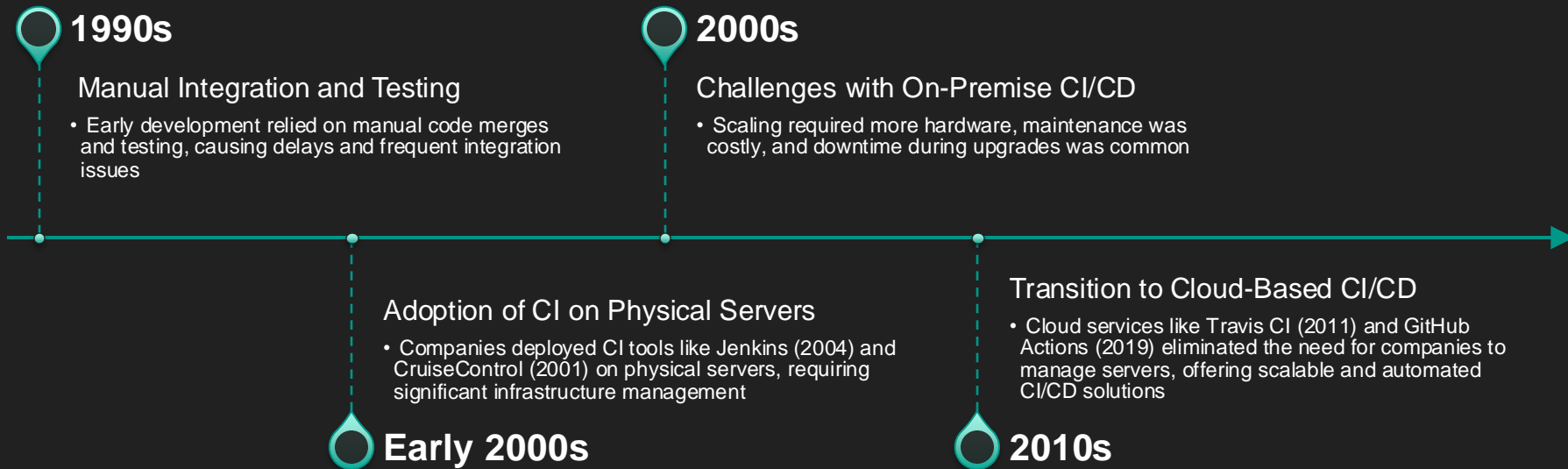


Code quality
improvements



Enhanced team
collaboration

History of CI/CD



GitHub Actions

- What is GitHub Actions?
 - Introduce Workflows
 - Events/Jobs/Runners
 - Integration with GitHub



GitHub Actions

What is GitHub Actions?

GitHub's Native CI/CD Tool

Automates workflows directly within GitHub repositories

Workflow Automation

Uses **YAML** files to define automated processes like builds, tests, and deployments

Event-Driven

Triggers workflows based on GitHub events (e.g., push, pull_request, or schedule)

Highly Customizable

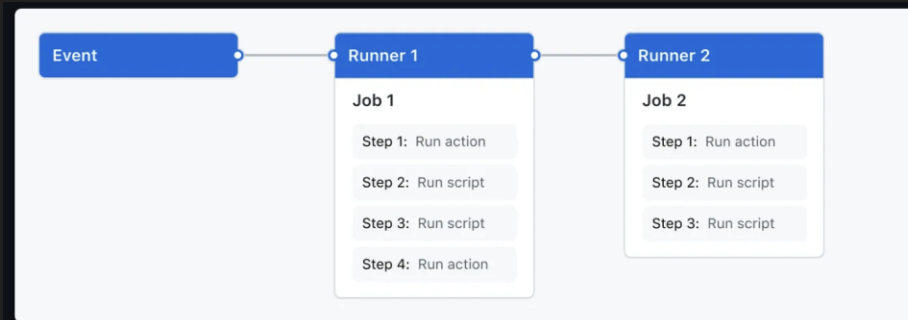
Supports custom actions, parallel jobs, and matrix builds for flexible automation

Seamless Integration

Built into GitHub, enabling easy integration with existing projects and third-party services

Workflows

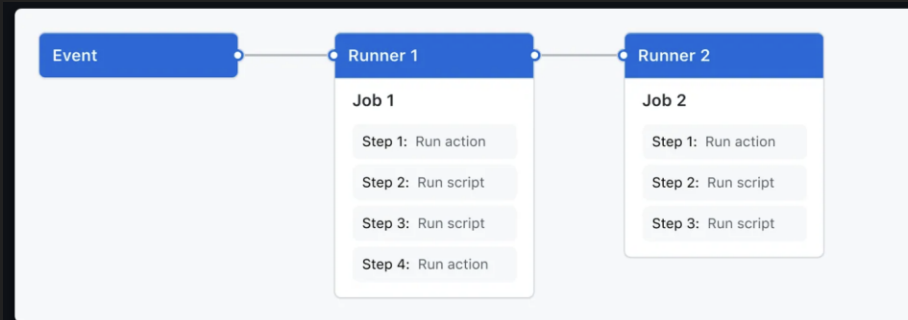
- **Definition:** Automated processes defined in **YAML files** that run jobs in response to GitHub events (e.g., push, pull_request)
- **Location:** Workflows are stored in the `.github/workflows/` directory of your repository



Workflows

- **Components**

- **Triggers:** Conditions that initiate a workflow in response to specific events, like a code push
- **Jobs:** A collection of steps executed on virtual machines (runners), which can run in parallel or sequentially
- **Steps:** Individual commands or actions that are part of a job



Jobs

- **Definition**

- A set of steps that are executed sequentially within a workflow

- **Parallel Execution**

- Multiple jobs can run simultaneously by default, improving workflow efficiency

- **Runs on Virtual Machines**

- Each job runs in its own isolated environment (runner), such as Ubuntu, macOS, or Windows

- **Dependencies Between Jobs**

- Jobs can be set to run sequentially or dependent on the success of other jobs

```
jobs:  
  my_first_job:  
    name: My first job  
  my_second_job:  
    name: My second job
```

Actions

- **Definition**

- Predefined, reusable tasks that perform specific functions within a job

- **Reusable Across Projects**

- Actions can be reused in multiple workflows or repositories, saving time and effort

- **Types of Actions**

- **Built-In Actions:** Provided by GitHub, like actions/checkout to clone a repository
- **Custom Actions:** Created by developers to perform specific tasks

- **Marketplace**

- GitHub Actions Marketplace offers thousands of ready-made actions for common tasks (e.g., building, testing, deploying)

- **Used in Steps:** Actions are invoked within steps to automate tasks

Actions

Automate your workflow from idea to production

Filter: All

By: All creators

Sort: Popularity



TruffleHog OSS

Scan GitHub Actions with TruffleHog

Action



Metrics embed

An infographics generator with 40+ plugins and 300+ options to display stats about your GitHub account

Action



yq - portable yaml processor

create, read, update, delete, merge, validate and do more with yaml

Action



Super-Linter

Super-linter is a ready-to-run collection of linters and code analyzers, to help validate your source code

Action



Gosec Security Checker

Runs the gosec security checker

Action



OpenCommit — improve commits with A...

Replaces lame commit messages with meaningful AI-generated messages when you push to remote

Action



Rebuild Armbian and Kernel

Support Amlogic, Rockchip and Allwinner boxes

Action



SSH Remote Commands

Executing remote ssh commands

Action



GitHub Pages action

GitHub Actions for GitHub Pages. Deploy static files and publish your site easily. Static-Site-Generators-friendly

Action



Cache

Cache artifacts like dependencies and build outputs to improve workflow execution time

Action



Build and push Docker images

Build and push Docker images with Buildx

Action



generate-snake-game-from-github...

Generates a snake game from a github user contributions grid. Output the animation as gif or svg

Action



Deploy to GitHub Pages

This action will handle the deployment process of your project to GitHub Pages

Action



Rebuild Armbian

Build Armbian Linux

Action



GitHub Script

Run simple scripts using the GitHub client

Action



GH Release

GitHub Action for creating GitHub Releases

Action



ChatGPT CodeReviewer

A Code Review Action Powered By ChatGPT

Action



Setup Node.js environment

Setup a Node.js environment by adding problem matchers and optionally downloading and adding it to the PATH

Action



FTP Deploy

Automate deploying websites and more with this GitHub action via FTP and FTPS

Action

Jobs

Event/Trigger

```
1  # Trigger the workflow on development branch
2
3  on:
4    push:
5      branches:
6        - development
7
8  jobs:
9    test:
10     runs-on: ubuntu-latest
11     steps:
12       - uses: actions/checkout@v2
13       - name: Set up Python
14         uses: actions/setup-python@v2
15         with:
16           python-version: '3.9'
17       - run: pip install -r requirements.txt
18       - run: pip install pytest
19       - name: Run Tests
20         run: pytest
21
22    build:
23     runs-on: ubuntu-latest
24     needs: test
25     steps:
26       - uses: actions/checkout@v2
27       - name: Build Docker image
28         run: docker build -t task-api .
```

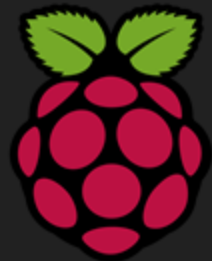
Jobs

Workflow Trivia

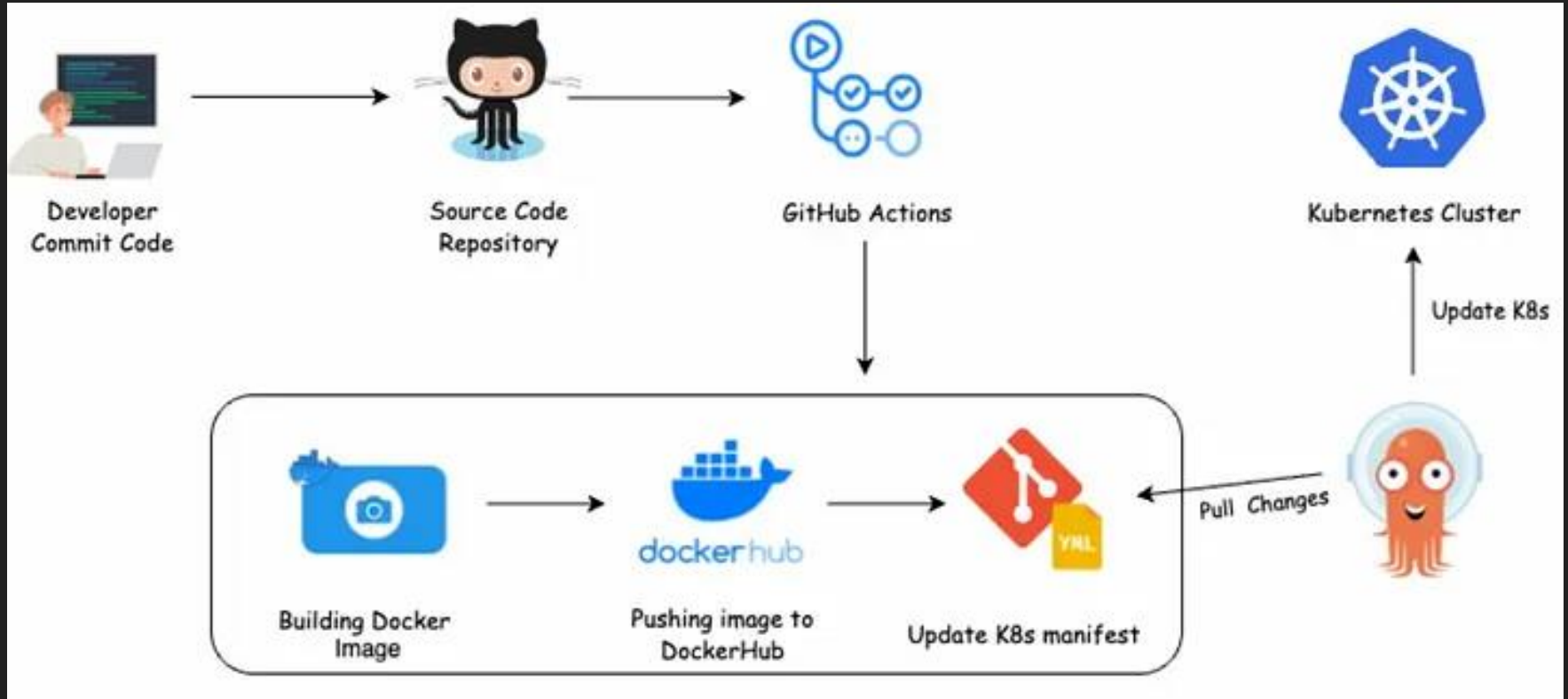
Building a CI/CD Pipeline with GitHub Actions

Touchless Kiosk Integration

- **GitHub Actions:**
 - Builds a new Docker image when code is pushed to the main branch
 - Pushes the image to a container registry
 - Updates the Kubernetes deployment config in the repo
- **Kubernetes (K3s or MicroK8s):**
 - Runs containers on Raspberry Pis
 - Manages deployments
- **ArgoCD:**
 - Monitors the repo for config changes
 - Automatically updates Kubernetes with the new image version
 - Ensures seamless deployment across all Raspberry Pis



Touchless Kiosk Integration



Deliverable

Create a GitHub Actions Workflow and upload it to GitHub!

Questions? / Happy Halloween

