The Problem & The Promise of CI/CD

Welcome! Today we'll explore CI/CD with GitHub Actions and learn how to automate our workflow.

by Benyam Seifu

Made with Gamma

The Problem & The Promise of CI/CD

Daily Code Pushes

Our team pushes code daily. Without automation, delays and errors creep in

Why CI/CD?

1 **Early Bug Detection**

Find problems before they reach production.

2 **Faster Feedback**

Instant results on every code change.

3 **Reliable Releases**

Consistent and predictable deployments.

Introducing GitHub Actions

GitHub Actions is a platform integrated into GitHub that automates tasks like building, testing, and deploying code based on events (e.g., a push or pull request).

```
name: CI
on:
  push:
    branches: [ main ]
jobs:
  build-and-test:
  ...
```

Q&A: What step do you think we need first to build our pipeline?

Building Our First Workflow

1

Checkout

Fetch the latest code from the repository.

2

Setup Java

Install the necessary Java environment.

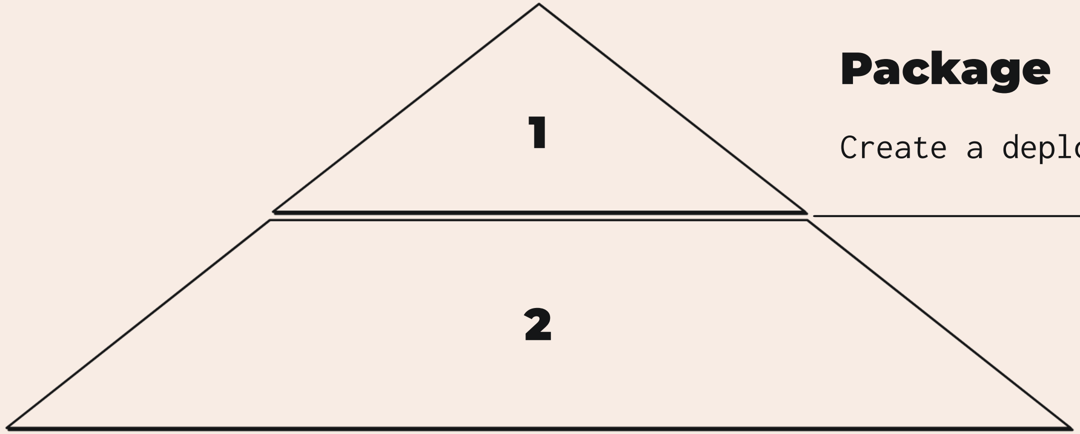
3

Run Tests

Execute tests to verify code quality.

Reflection: If tests fail, do we deploy? Why or why not?

Extending to Build & Deploy



Package

Create a deployable artifact (e.g., JAR file).

Deploy

Push the artifact to your target environment.

Common Pitfalls & Troubleshooting

Missing Actions

Ensure you've installed the required tools.

Incorrect File Names

Check for typos in paths and file names.

Permissions

Verify that your workflow has the necessary access.

If something doesn’t work, what’s the first step you’d take to debug?

Wrap-Up & Review

summarize CI/CD in your own words