CI/CD overview notes

CI/CD through the lens of github actions

What is CI/CD

What does this translate to when using github

Examples

Azure develops, same but different

Continuous integration: the practice of automatically integrate changes from contributors.

Pipeline : Plan – code – build – test

Continuous Delivery: the practice of automatically delivering an artifact when the codebase changes.

Continuous Deployment. The practice of automatically deploying a project when the code changes.

Don’t commit to main use pull requests

When a pr is made: build the code, test the code , only on success.

Define the target.

Restrict deletions

Require pull request before merging

Require status checks to pass

Require branches to be up to date before merging

Block force pushes.

Synercoder / cicd example plain java

Using workflows to automate:

YAML file that define the steps the automation.

Workflow files are located in the folder .github/workflow.

Github- hosted runners

Name:

On :

Workflow\_dispatch:

Pull\_request:

Jobs:

Build:

Name: “ build and test” (important)

Runs-on: ubuntu-latest

Steps:

Uses:actions/checkout@v4

Name: set up JDK17

Uses: actions/setup-java@v4

With:

Java-version: 17

Distribution: ‘oracle’

Name: build classes

Environment varables: define environment variable globally, per job or per step.

Most specific variable is used.

Expressions

Secrets

Dependencies. (needs keyword)

Conditions

Matrix strategies

Actions ( pre packaged scripts)

Integration tests

Docker example: https:// github.com/synercoder/CICD-example-docker-testing

Docker-test.yml file (check its important for workflow)

Anything you can do in the command line you can automate in a pipeline

<https://github.com/synercoder/CICD-example-java-maven>

azure pipeline

machine settings

conditional expressions

dependencies

actions vs tasks

come work/have your internship @ fotofabriek