CI-CD

Continuous Integration

Continuous Delivery

Build

2 Backend → 2yrs

Impl

tests

Unit E2E Integration

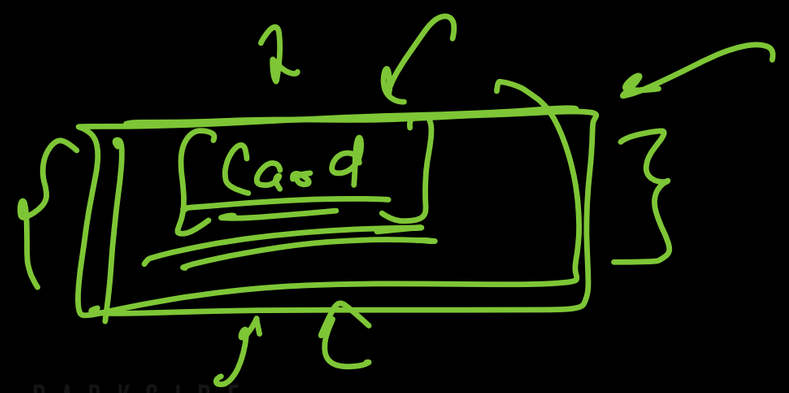
and more ...

1) Code the feature

Some other tests are failing

2) Write some tests ✓

3) Commit



JOIN THE DARKSIDE

Code → unit → commit

automatically

1) Build
2) Linting
3) Unit.
4) Integrati.
5) E2E
⋮

CI Pipeline

JOIN THE DARKSIDE

→ CI (continuous integration) aims to make it easier for developers to integrate their code changes easily.

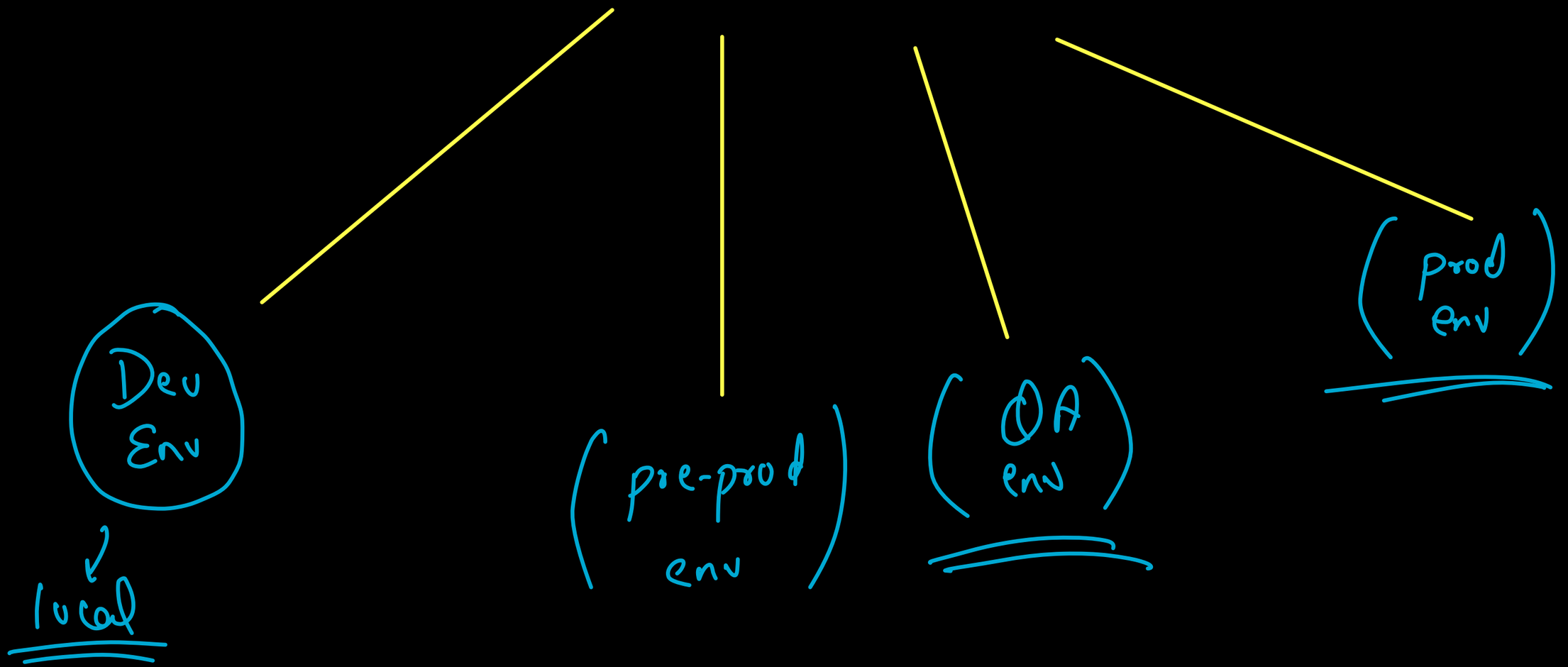
→ It helps in improving the efficiency. Every time we submit the code, the CI system automatically build & test & checks everything to ensure that no errors occur or anything fails.

JOIN THE DARKSIDE

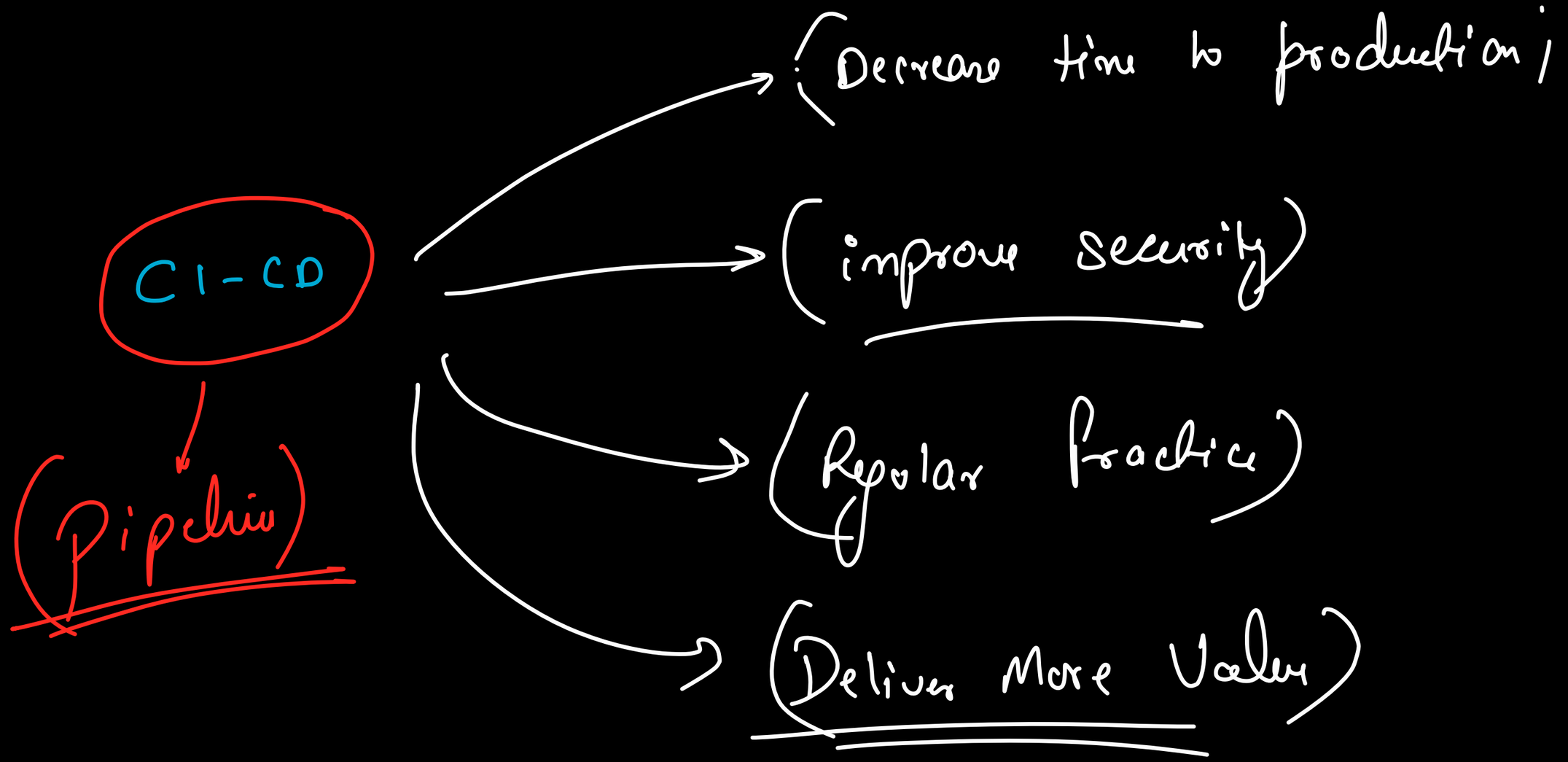
→ if there is anything breaking the CI will help us
to identify it early.

JOIN THE DARKSIDE

Continuous Delivery (CD)

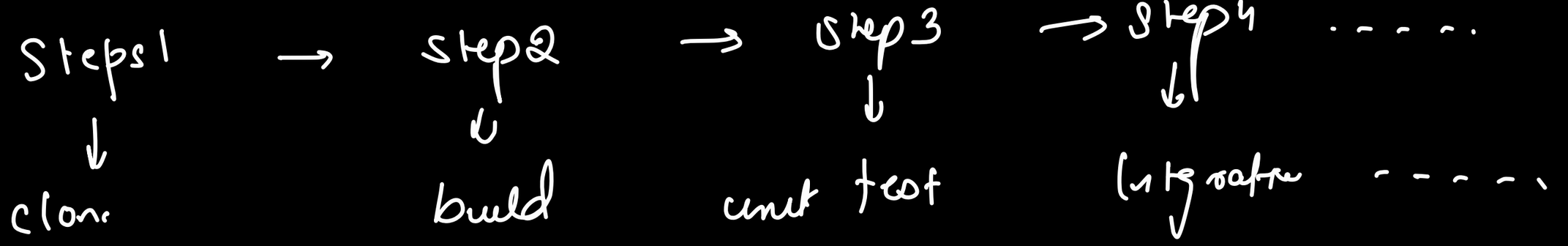


JOIN THE DARKSIDE



JOIN THE DARKSIDE

→ A build pipeline is a sequence of operations & tools from Source code to the deployed system.



JOIN THE DARKSIDE

Tools for implementing CI-CD

- azure devops
- aws codepipeline
- Github actions
- Gitlab CI-CD
- Jenkins
- Travis CI
- Circle CI

JOIN THE DARKSIDE

JSON

easy

YAML

file format

{ key : value }

(YAML Ain't Markup Language)

Strict indentation

Config , data exchange

JOIN THE DARKSIDE

Github Actions

powerful automation tool integrated in github.com.

you can setup automated workflows.

- workflow → automated process created using one or more jobs
- jobs → one operation
- step → individual task
- actions

JOIN THE DARKSIDE

Jobs → it is a set of steps to be executed

↓

setup a pipeline that builds & tests a project.

JOIN THE DARKSIDE

pipeline

workflow

YAML

events

push on master

Job1

Step1

Step2

Job2

Step1

Step2

github

.github/workflows

yaml

JOIN THE DARKSIDE

```yaml
name: "Hello world workflow"

# Controls when the action will run. Triggers
on:
  push:
    branches:
      - master
      - production

jobs:
  hello_word:
    runs-on: ubuntu-latest
  steps:
    - name: "Print hello world"
      run: echo "Hello world"

    - name: "Print current date"
      run: date
```

name of the workflow / Pipeline

whenever we push code on master / production branch

defines the jobs

whenever the event happens we will trigger the pipeline

event to trigger the pipeline

name of the job

defines the env of runner on which job runs

step 1

step 2

steps of the job

JOIN THE DARKSIDE

click here to see all the pipeline/workflow

singhsanket143 / **Sample-Github-Actions-Workflow**

Type / to search

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

Sample-Github-Actions-Workflow

Public

Pin | Unwatch 1 | Fork 0 | Star 0

master | 2 Branches | 0 Tags | Go to file t | Add file | Code

singhsanket143 Added workflow ✓ 3836b3b · 2 hours ago | 4 Commits

.github/workflows	Added workflow	2 hours ago
README.md	Added workflow	2 hours ago

README

Welcome to the sample github actions workflow setup.

Adedd first workflow

About

No description, website, or topics provided.

Readme
Activity
0 stars
1 watching
0 forks

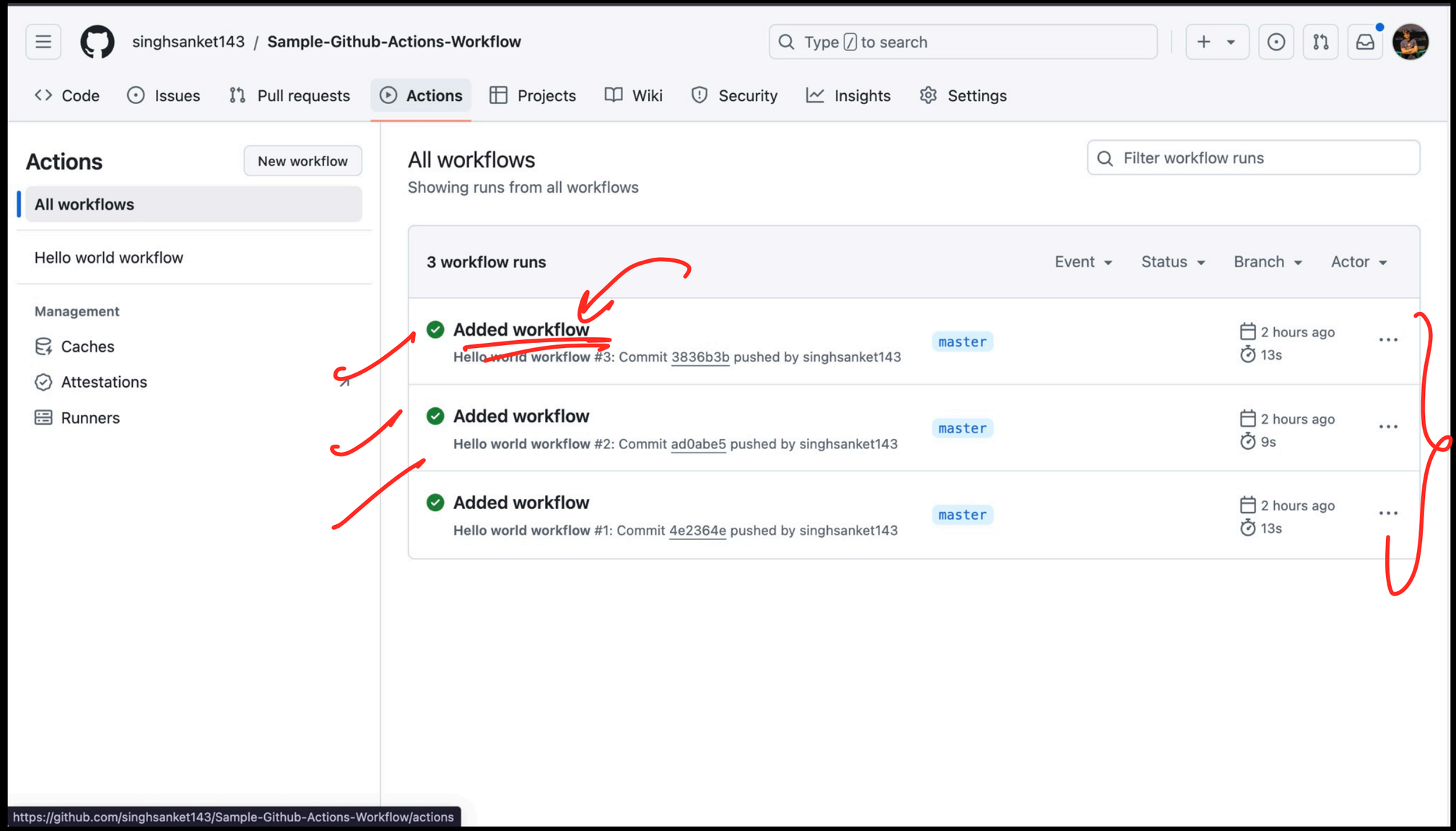
Releases

No releases published
Create a new release

Packages

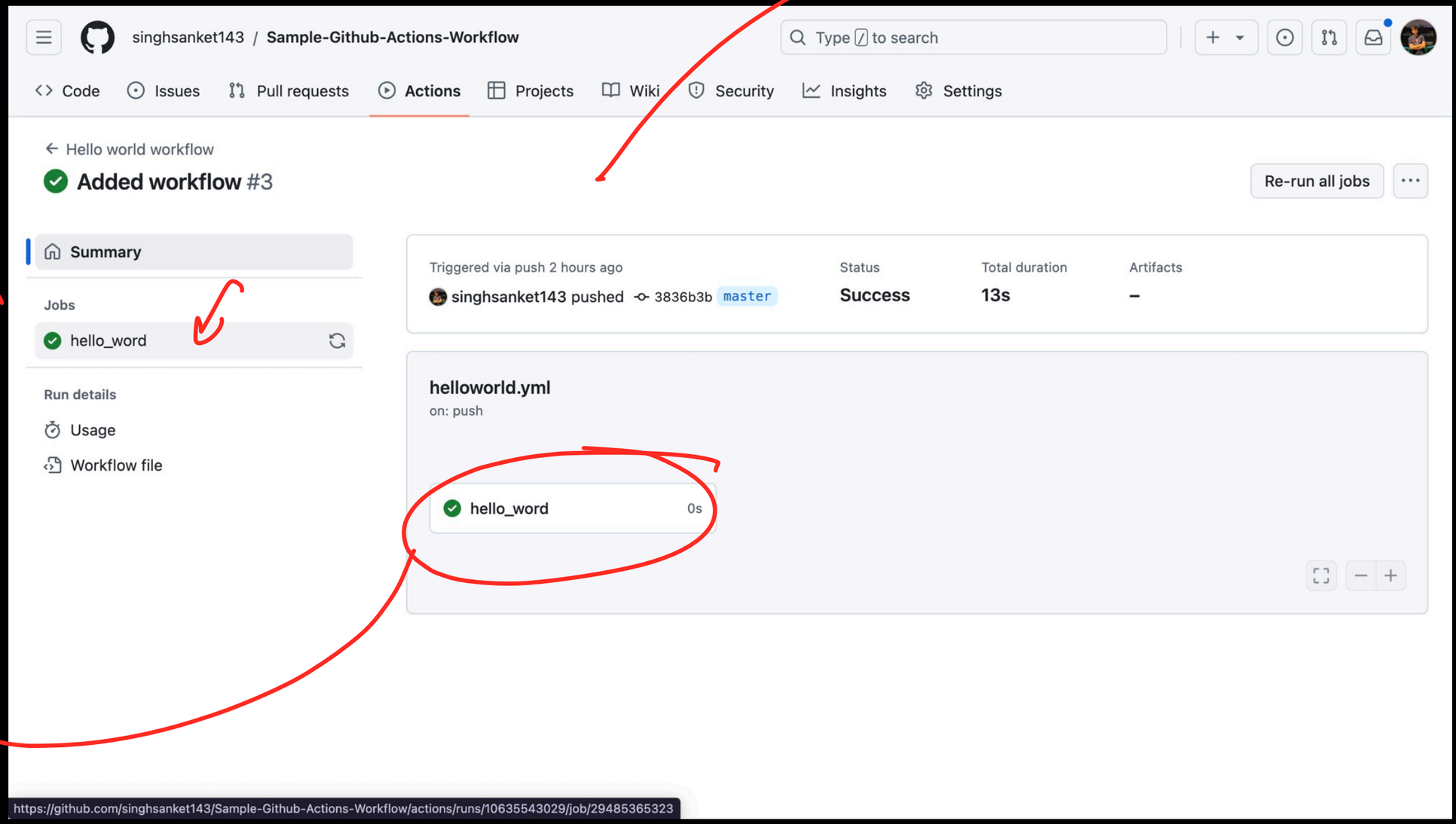
No packages published
Publish your first package

JOIN THE DARKSIDE



JOIN THE DARKSIDE

details of a single exec



list of jobs

JOIN THE DARKSIDE

singhsanket143 / **Sample-Github-Actions-Workflow**

Type / to search

Code | Issues | Pull requests | **Actions** | Projects | Wiki | Security | Insights | Settings

← Hello world workflow

Added workflow #3

Re-run all jobs ···

Summary

Jobs

- hello_word

Run details

- Usage
- Workflow file

hello_word
succeeded 2 hours ago in 0s

Search logs

- Set up job 0s
- Print hello world 1s
- Print current date 0s
- Complete job 0s

Steps of the job

JOIN THE DARKSIDE

name of the pipeline

any push of a commit to master / test branch

```yaml
name: "Multi event pipeline / workflow" # Name of the pipeline
on:
    push: # Event: Whenever a push is made to the repository on the master / test branch
        branches:
            - master
            - test
    pull_request: # Event: Whenever a pull request is made to the repository
        branches:
            - master
        types:
            - closed
            - opened
    workflow_dispatch:
```

events

to run jobs if a pull req is opened or closed on branch master.

to manually trigger jobs from UI

if is to add conditions when to run the job.

```yaml
jobs:
    on-push: # Job 1
        if: github.event_name == 'push'
        runs-on: ubuntu-latest
        steps: # Steps to be executed in the job
            - name: "Running on master/test push" # Step 1
              run: echo "Running on master/test push"

    on-pull-request: # Job 2 ⋯

    on-pull-request-opened:
        if: github.event_name == 'pull_request' && github.event.action == 'opened'
        runs-on: ubuntu-latest
        steps:
            - name: "Running on pull request opened"
              run: echo "Running on pull request opened"

    on-pull-request-closed:
        if: github.event_name == 'pull_request' && github.event.action == 'closed'
        runs-on: ubuntu-latest
        steps:
            - name: "Running on pull request closed"
              run: echo "Running on pull request closed"

    on-workflow-dispatch: # Job 3
        if: github.event_name == 'workflow_dispatch'
        runs-on: ubuntu-latest
        steps: # Steps to be executed in the job
            - name: "Running on workflow dispatch" # Step 1
```

JOIN THE DARKSIDE