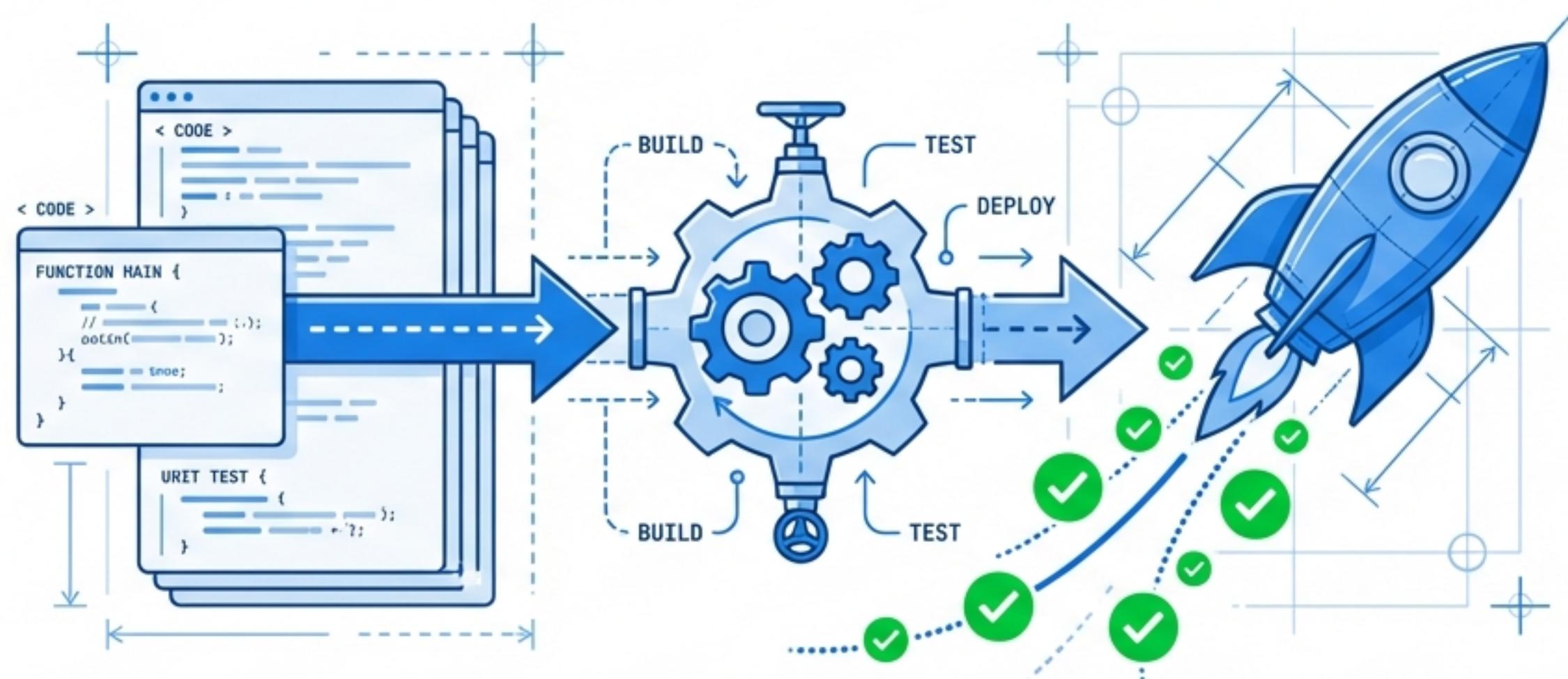


# GitHub Actions: Автоматизований CI/CD

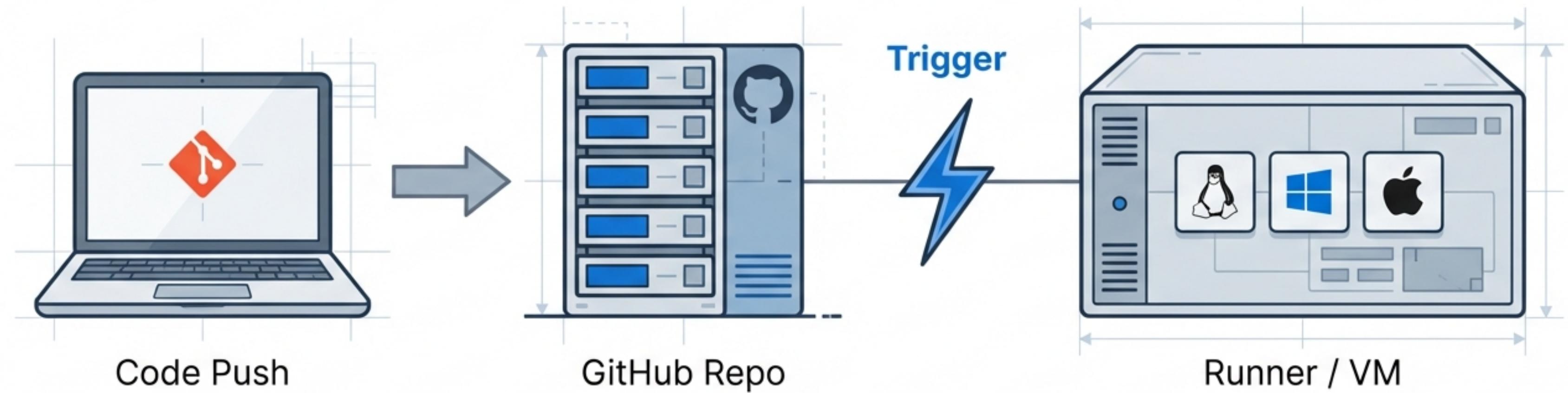
## Від "Zero" до E2E тестування



Ваш код заслуговує на більше, ніж ручне тестування.

[Read Time: 10 min] | [Level: Engineering]

# Екосистема: Що відбувається "під капотом"?



## CI/CD в репозиторії

Автоматизація без зовнішніх  
сервісів.

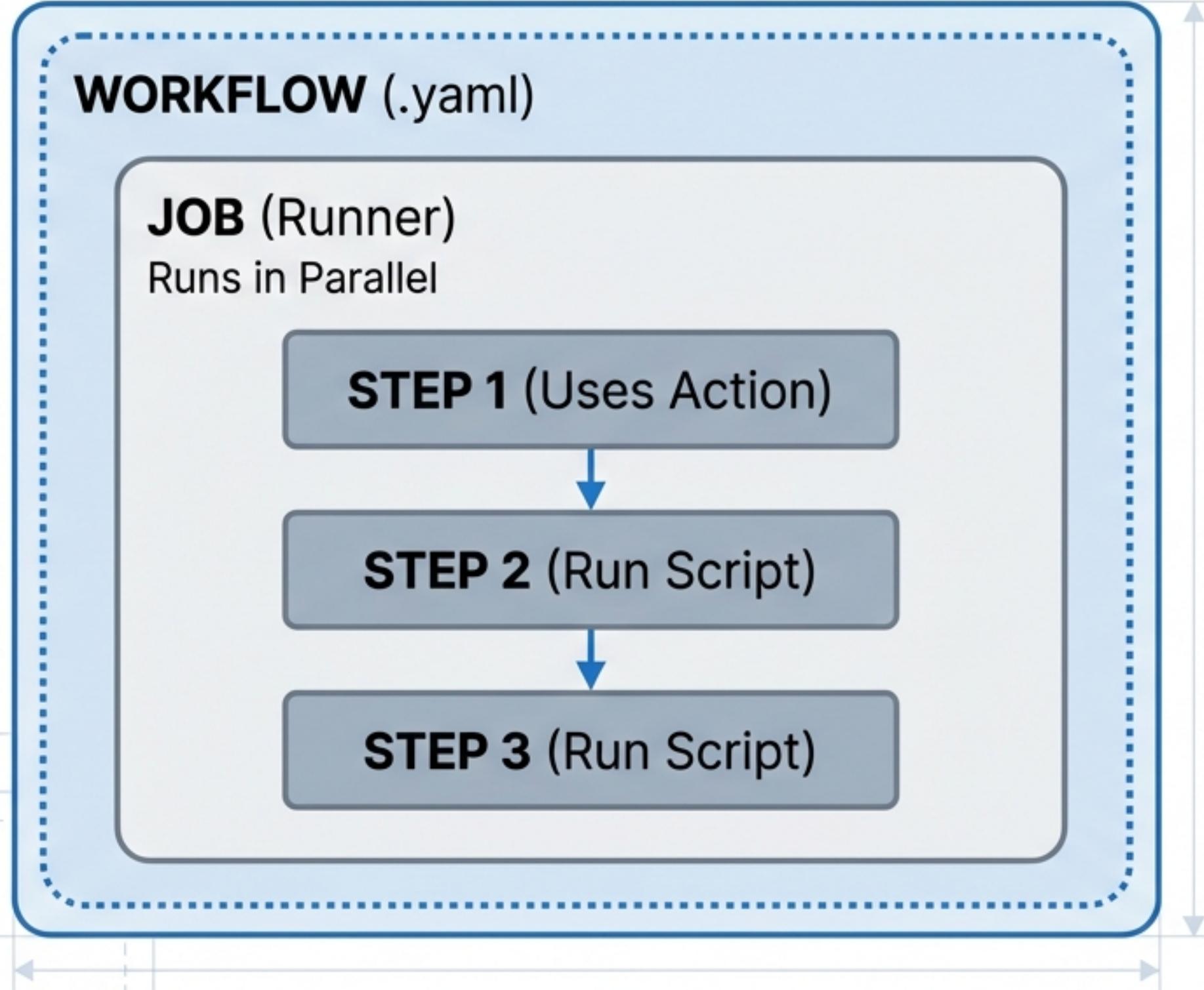
## Віртуальні машини

GitHub виділяє чисте  
середовище (Linux, Windows,  
macOS) для кожного запуску.

## Інтеграція

Cypress та Playwright  
запускаються прямо тут через  
YAML конфігурацію.

# Анатомія Workflow



- **Workflow:** Автоматизований процес (YAML файл).
- **Job:** Набір кроків на **\*одному\*** runner.
- **Step:** Скрипт (`run`) або готова дія (`uses`), що виконуються **\*послідовно\***.

**Pro-Tip:**  
Steps у рамках одного Job мають спільний простір і можуть обмінюватися даними. Jobs — ізольовані.

# Інфраструктура та Білінг

## RUNNERS

- **GitHub-hosted:**

Ubuntu (найдешевше), Windows, macOS. Кожен run – це нова VM.



- **Self-hosted:**

Ваші власні сервери або контейнери.  
Повний контроль, економія на хвилинах.



## ЛІМІТИ

- **Публічні репо:**

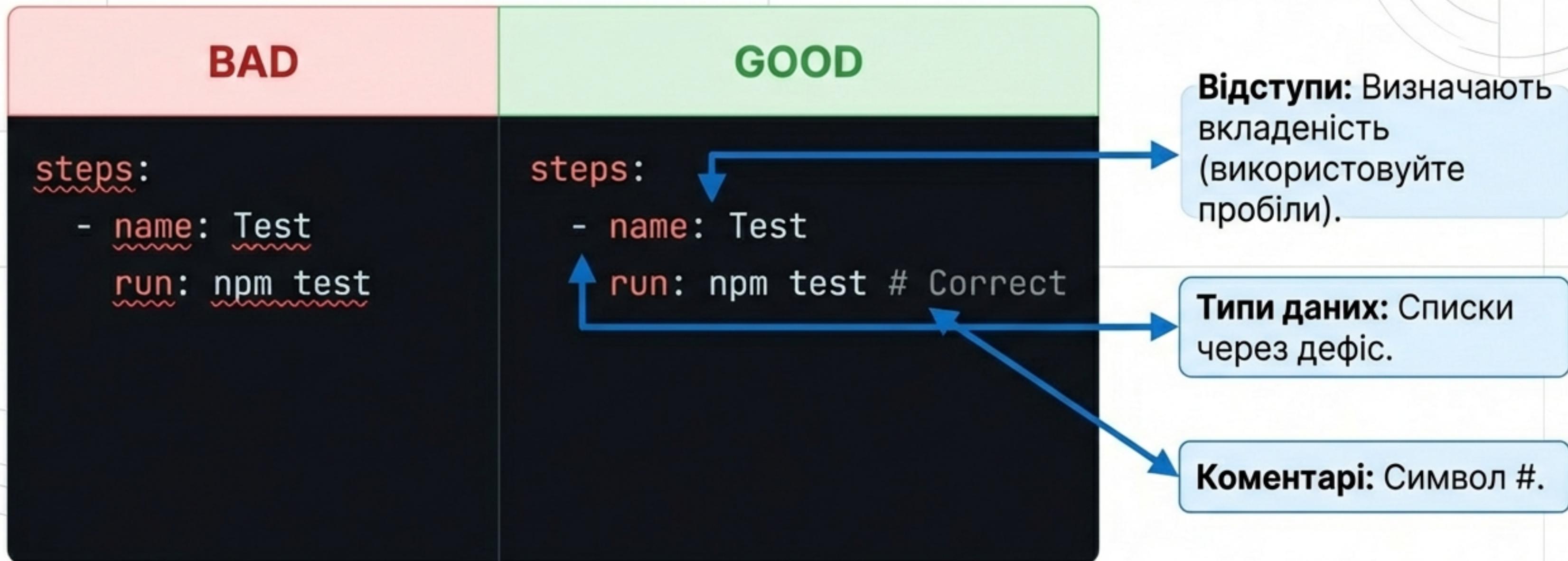
Безкоштовно (стандартні runners).

- **Приватні репо:**

Ліміт хвилин залежно від плану.

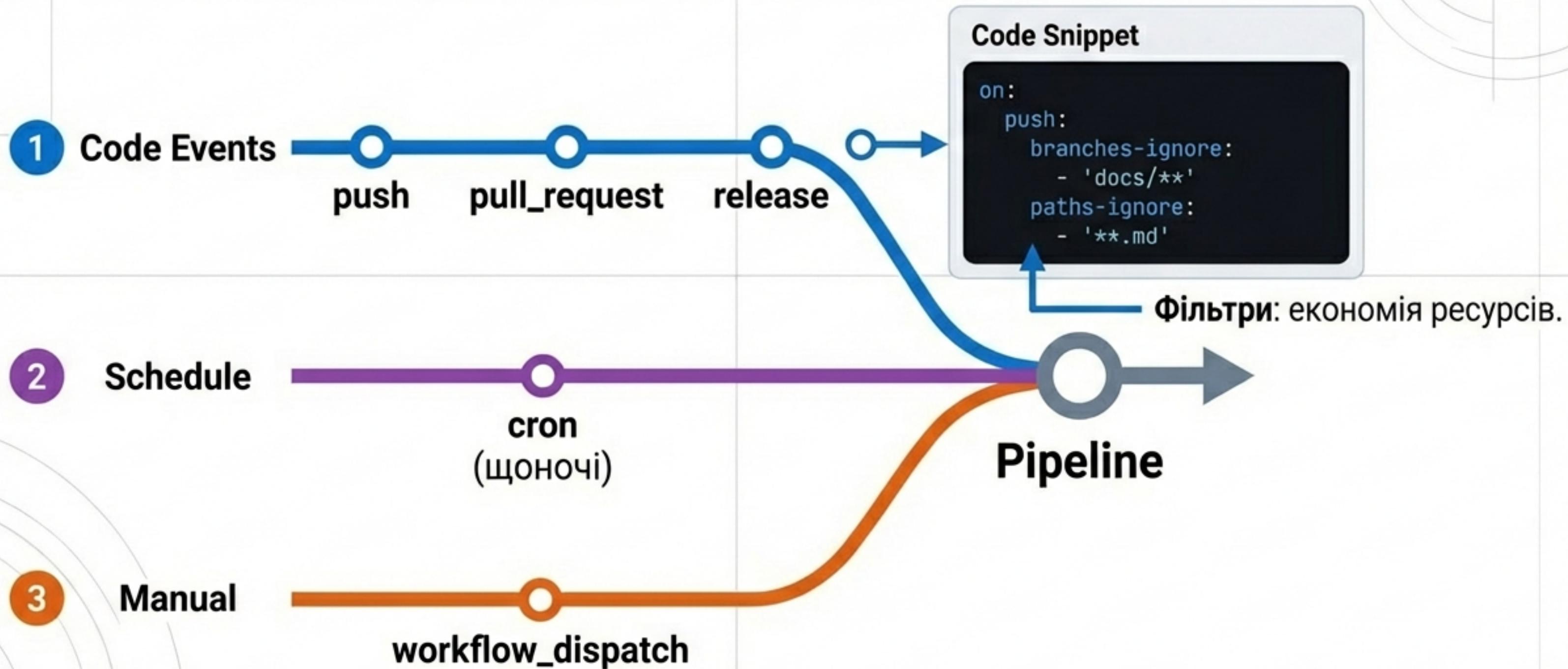
**Warning:** Контролюйте витрати через Billing налаштування.

# Чому YAML? Синтаксис має значення



YAML — це мінімалізм. Немає зайвих дужок, лише дані.

# Тригери: Керування Часом та Подіями



# Cypress: Базова Інтеграція

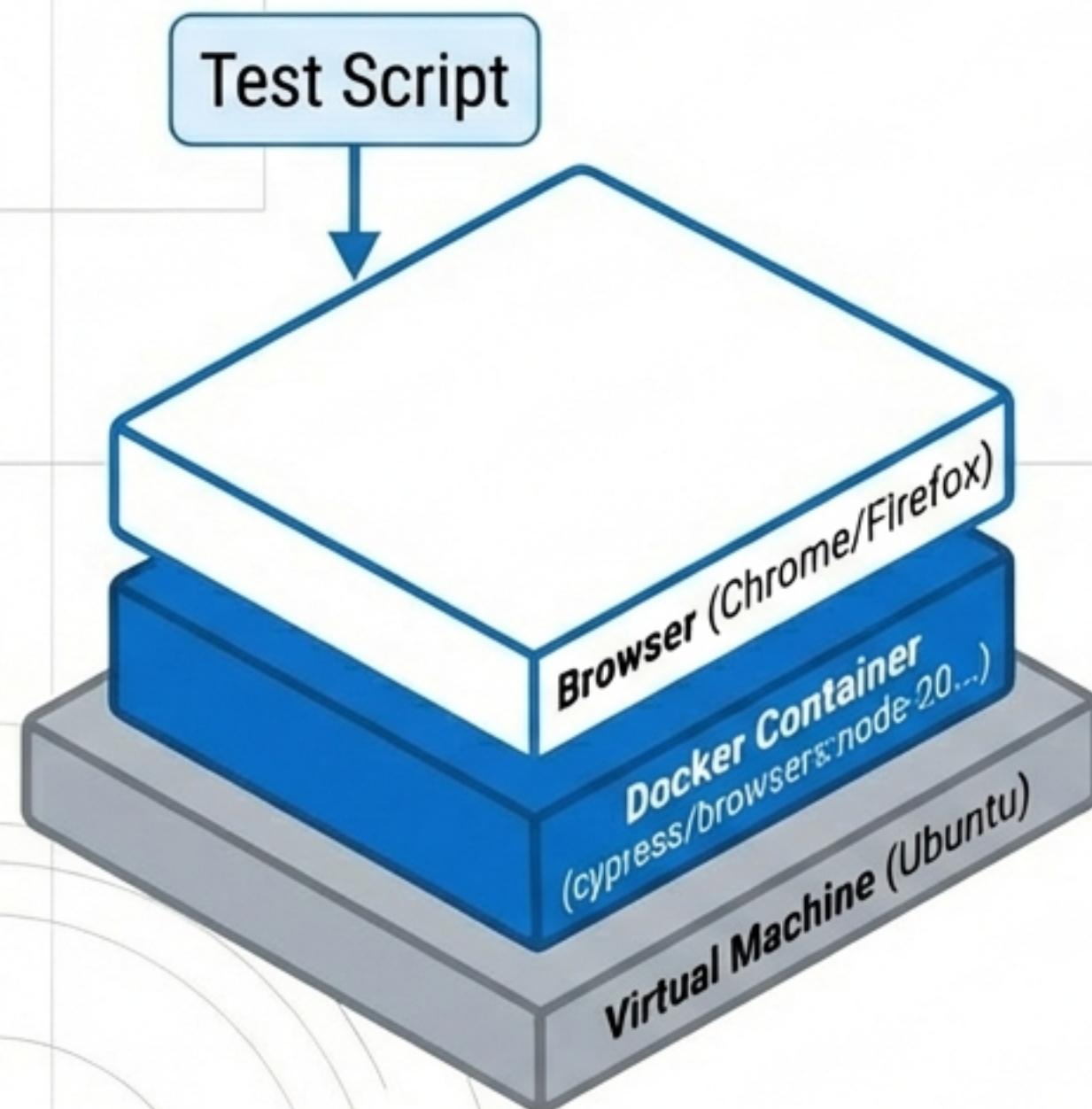
```
1 name: E2E Tests
2 on: [push]
3 jobs:
4   cypress-run:
5     runs-on: ubuntu-latest
6     steps:
7       - name: Checkout
8         uses: actions/checkout@v4
9       - name: Cypress Run
10        uses: cypress-io/github-action@v6
```

**Electron Browser**  
(Default)

**Automated Setup:**  
npm install + test run

**Headless mode** за  
замовчуванням.

# Cypress: Браузери та Контейнери



```
container: cypress/browsers:node-20.9.0-chrome-118  
...  
with:  
  browser: chrome  
  env: host=api.dev, port=3000
```

Змінні env передаються одним рядком через кому.

# Безпека: Secrets та Keys



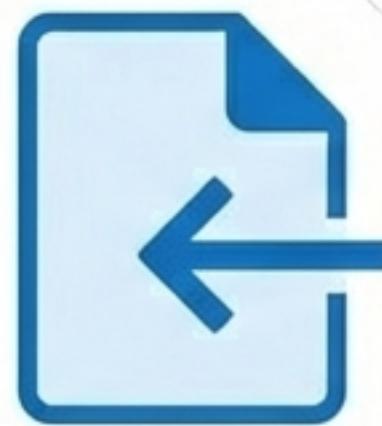
**Generate Key**

Cypress Cloud



**Store Secret**

GitHub Settings -> Secrets

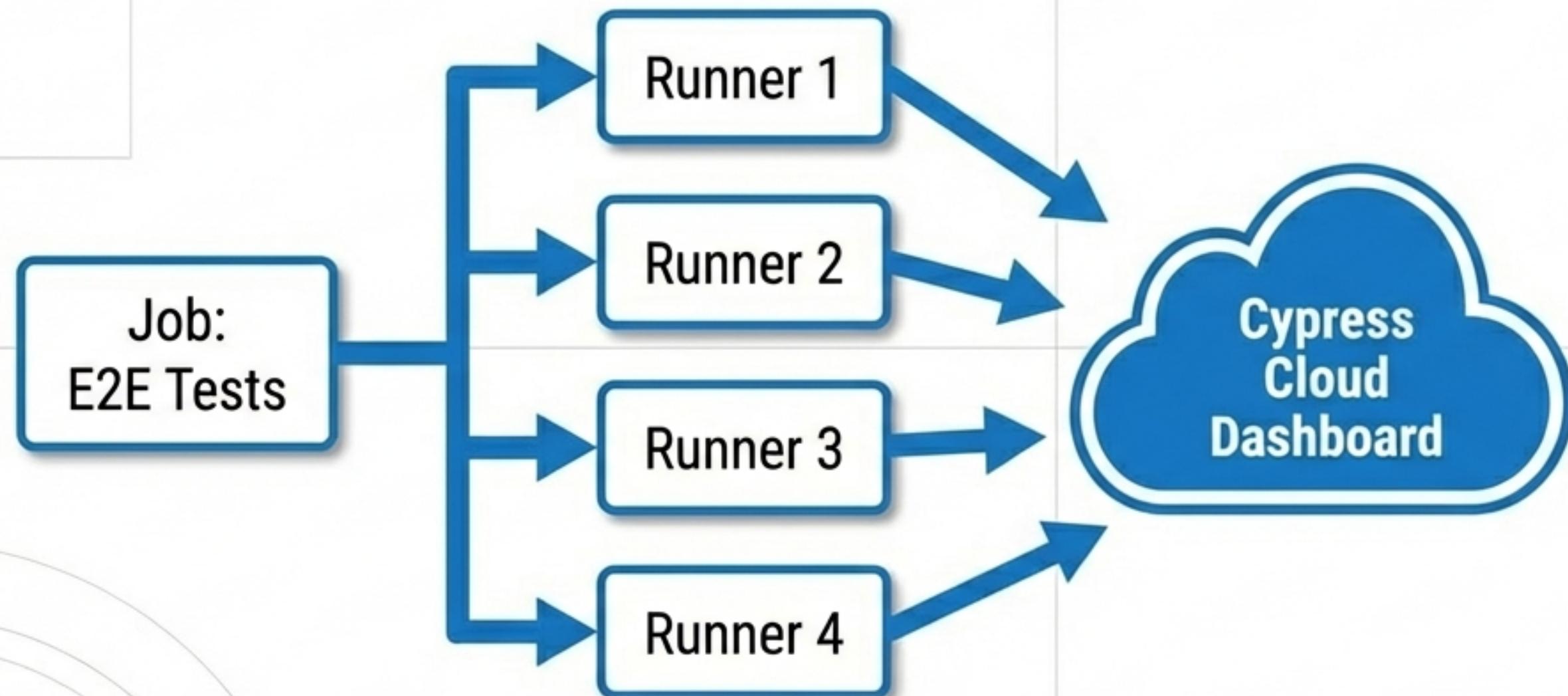


**Inject into YAML**

```
env:  
  CYPRESS_RECORD_KEY: ${{ secrets.CYPRESS_RECORD_KEY }}  
  GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
```

**Warning:** Ніколи не комітьте ключі прямо в код!

# Cypress Cloud: Паралелізація



**Matrix Strategy:**  
**containers: [1, 2, 3, 4]**  
**record: true**  
**parallel: true**

**Pro-Tip**

**Problem:** Re-runs blocked?

**Solution:** Генеруйте унікальний `ci-build-id` (UUID).

# Playwright: Командний рядок

```
Terminal
user@github-runner:~$ npm ci
(installing dependencies...)
user@github-runner:~$ npx playwright install --with-deps
(downloading browsers...)
user@github-runner:~$ npx playwright test
(running tests...)
```

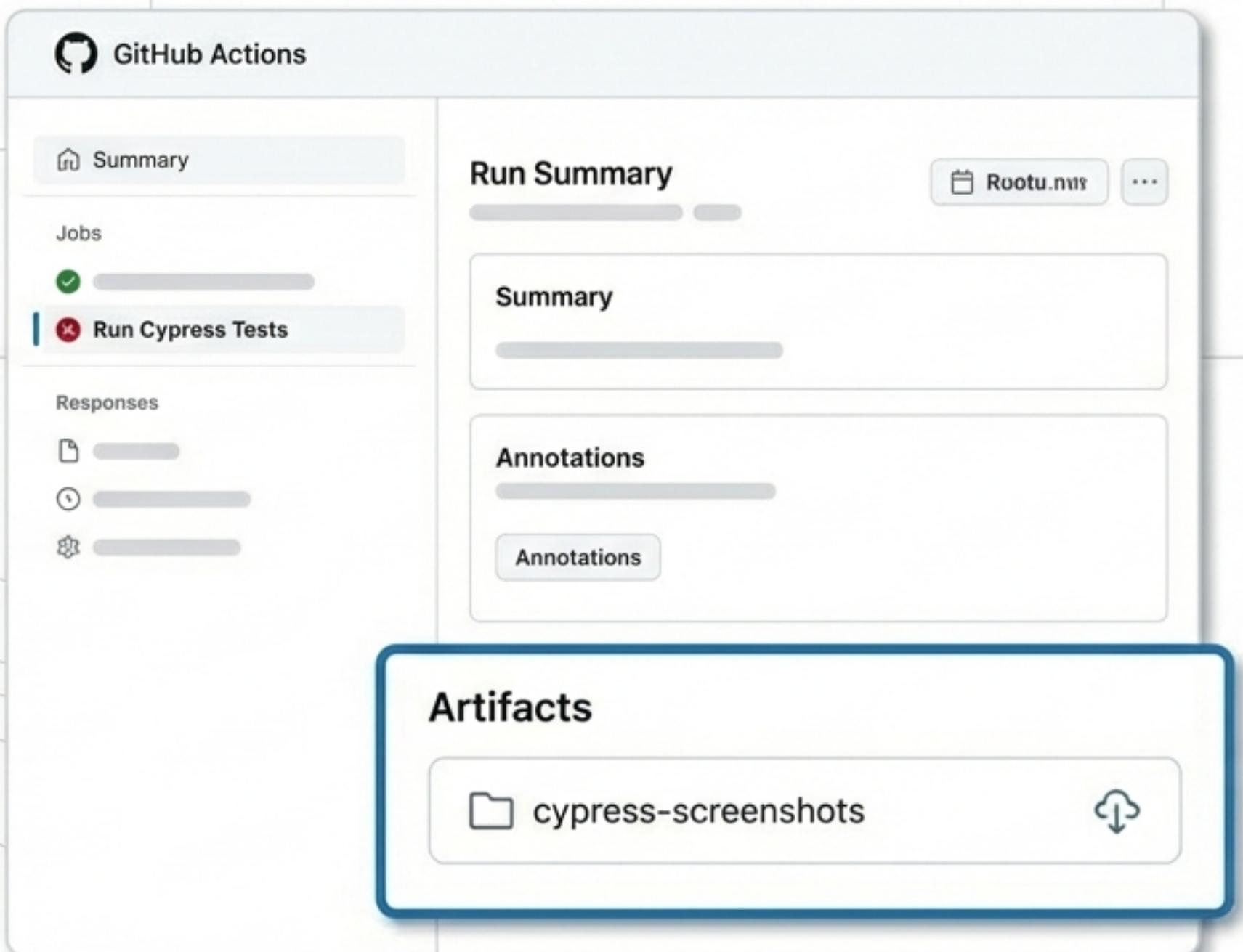
Playwright  
використовує  
стандартні `run`  
кроки замість  
готового action.

# Playwright: Docker та Звіти



- Docker Image:**  
v1.39.0-jammy  
(Match package.json!)
- Snippet:**  
`if: always()` –  
зберігати звіт навіть  
якщо тести впали.
- HTML звіти не**  
**інтерактивні** в GitHub  
UI – завантажуйте їх  
локально.

# Артефакти: Коли щось пішло не так



```
- name: Upload Screenshots  
if: failure()  
uses: actions/upload-artifact@v4  
with:  
path: cypress/screenshots
```

Використовуйте `'if: failure()'`,  
щоб ловити докази помилок.

# Master Cheatsheet

## Keywords

- `on`
- `jobs`
- `steps`
- `needs`
- `runs-on`
- `uses`

## Events

`push`  
`pull_request`  
`schedule (cron)`  
`paths-ignore`

## Variables

`${{ secrets.NAME }}`  
`${{ github.run_id }}`  
`${{ env.VAR }}`

## Best Practices

- `fail-fast: false`
- Cache `node_modules`
- Avoid hard-coded versions

# Автоматизація — це Інвестиція

- ✓ GitHub Actions дає CI/CD "з коробки".
- ✓ Почніть з простого `push` тригера.
- ✓ Додайте Docker для стабільності.
- ✓ Масштабуйте через паралелізацію.



**Зробіть свій перший workflow сьогодні.**

[docs.github.com/actions](https://docs.github.com/actions)