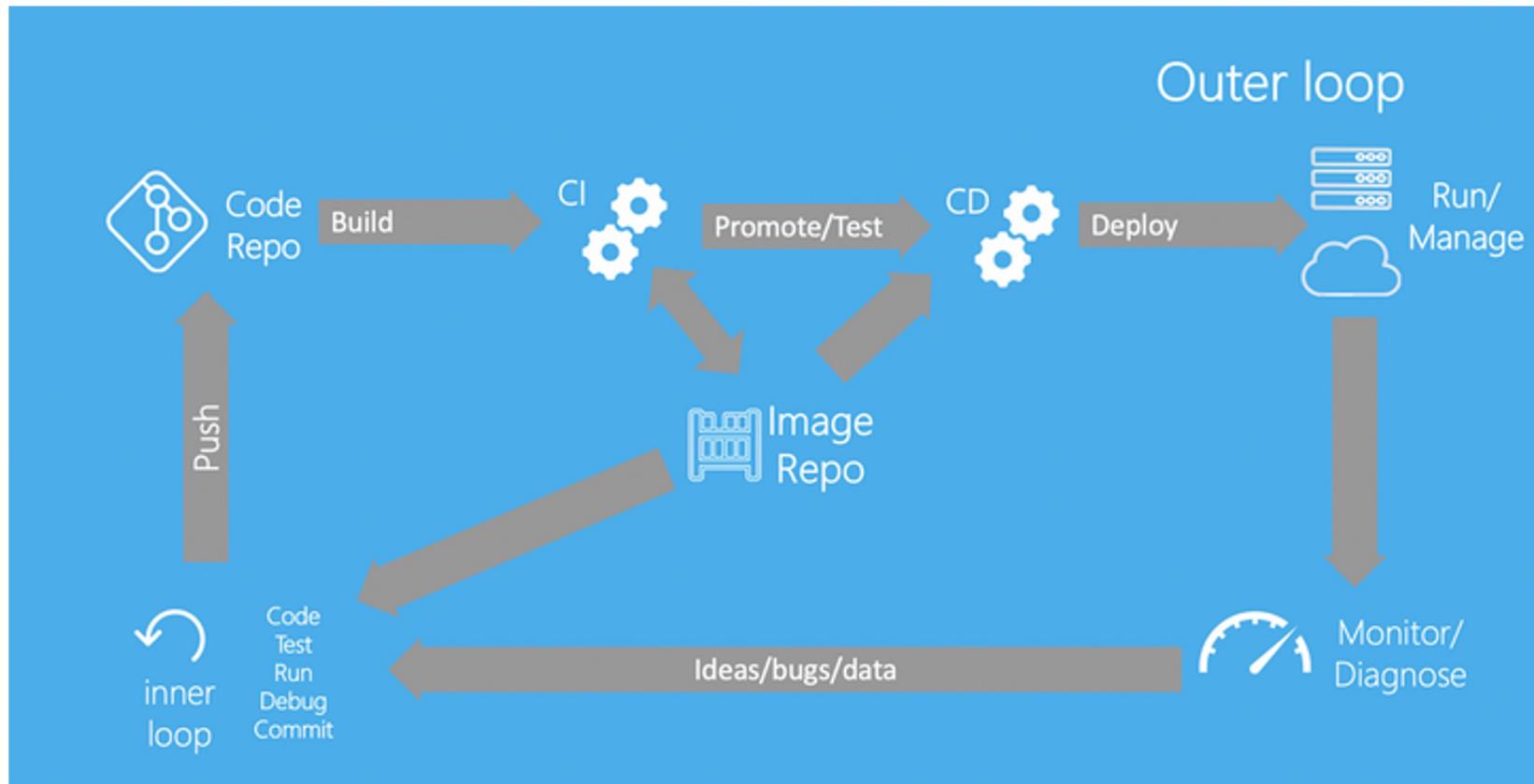


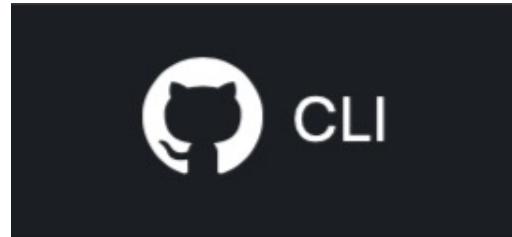
# PFSwChO – Laboratorium6

Obszar "działan" na to i kolejne laboratoria



# PFSwChO – Laboratorium 6

GitHub – przydatne dodatki



<https://cli.github.com>

Instalacja w środowisku Linux (różne dystrybucje)

[https://github.com/cli/cli/blob/trunk/docs/install\\_linux.md](https://github.com/cli/cli/blob/trunk/docs/install_linux.md)

Dokumentacja i tutoriale

<https://docs.github.com/en/github-cli>

Referencyjny podręcznik Github CLI

<https://cli.github.com/manual/>

```
[~] └─ gh --version
gh version 2.20.0 (2022-11-08)
https://github.com/cli/cli/releases/tag/v2.20.0
[~]
└─ gh help
Work seamlessly with GitHub from the command line.

USAGE
  gh <command> <subcommand> [flags]

CORE COMMANDS
  auth:      Authenticate gh and git with GitHub
  browse:    Open the repository in the browser
  codespace: Connect to and manage codespaces
  gist:      Manage gists
  issue:    Manage issues
  pr:        Manage pull requests
  release:  Manage releases
  repo:     Manage repositories
```

#### ACTIONS COMMANDS

run:	View details about workflow runs
workflow:	View details about GitHub Actions workflows

# PFSwChO – Laboratorium 6

## GitHub CLI – podstawowa konfiguracja, cz. I

```
apple ~% gh auth --help
Authenticate gh and git with GitHub

USAGE
  gh auth <command> [flags]

CORE COMMANDS
  login:      Authenticate with a GitHub host
  logout:     Log out of a GitHub host
  refresh:    Refresh stored authentication credentials
  setup-git:   Configure git to use GitHub CLI as a credential helper
  status:     View authentication status
  token:      Print the auth token gh is configured to use
```

Należy wygenerować parę kluczy dla realizacji dostępu do konta na GitHub za pomocą GitHub CLI

```
[~]
└─ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? SSH
? Upload your SSH public key to your GitHub account? /home/slawek/.ssh/gh_cli_q16ub_ed25519.pub
? Title for your SSH key: GitHub CLI
? How would you like to authenticate GitHub CLI? Paste an authentication token
Tip: you can generate a Personal Access Token here https://github.com/settings/tokens
The minimum required scopes are 'repo', 'read:org', 'admin:public_key'.
? Paste your authentication token: *****
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
✓ Uploaded the SSH key to your GitHub account: /home/slawek/.ssh/gh_cli_q16ub_ed25519.pub
✓ Logged in as SenatorP51
```

# PFSwChO – Laboratorium6

## GitHub CLI – podstawowa konfiguracja, cz. II

```
[~] └─ gh auth login --with-token < ~/.ssh/gh_p.txt  
[~] └─ gh auth status  
github.com  
  ✓ Logged in to github.com as SenatorP51 (oauth_token)  
  ✓ Git operations for github.com configured to use https protocol.  
  ✓ Token: *****
```

Przykład

```
[~] └─ gh help actions  
Welcome to GitHub Actions on the command line.  
  
GitHub CLI integrates with Actions to help you manage runs and workflows.  
  
Interacting with workflow runs  
gh run list:      List recent workflow runs  
gh run view:     View details for a workflow run or one of its jobs  
gh run watch:    Watch a workflow run while it executes  
gh run rerun:    Rerun a failed workflow run  
gh run download: Download artifacts generated by runs
```

UWAGA: W dalszej części instrukcji jak i laboratorium wykorzystywana będzie metoda autentykacji oparta na **personal-access-token**



<https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token>

Sprawdzenie poprawności logowania

[https://cli.github.com/manual/gh\\_auth\\_login](https://cli.github.com/manual/gh_auth_login)

Integracja z GitHub Actions

[https://cli.github.com/manual/gh\\_run](https://cli.github.com/manual/gh_run)

[https://cli.github.com/manual/gh\\_workflow](https://cli.github.com/manual/gh_workflow)

# PFSwChO – Laboratorium6

## GitHub – przydatne dodatki



<https://desktop.github.com>

<https://docs.github.com/en/desktop>

GitHub – przydatne dodatki **REPOZYTORIUM ghcr.io**

You can store and manage **Docker and OCI images** in the Container registry, which uses the package namespace <https://ghcr.io>.

1. Należy utworzyć (zweryfikować ustawienia) personal-access-token (**UWAGA: wersj "classic"**)

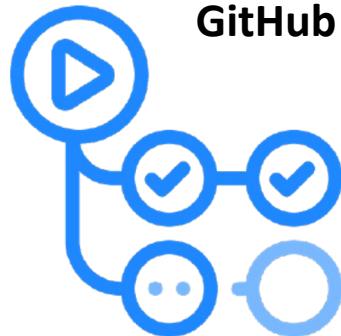
<https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token>

2. Następnie można zalogować się na repozytorium (**UWAGA: należy uruchomi Docker Desktop**)

<https://docs.github.com/en/packages/working-with-a-github-packages-registry/working-with-the-container-registry>

```
[apple ~] export CR_LAB=ghcr.io/ SenatorP51
[apple ~] echo $CR_LAB | docker login ghcr.io -u SenatorP51 --password-stdin
Login Succeeded
[apple ~]
```

Linki, o których warto pamiętać:



<https://github.com/pricing>

<https://docs.github.com/en/actions/learn-github-actions/usage-limits-billing-and-administration#about-billing-for-github-actions>

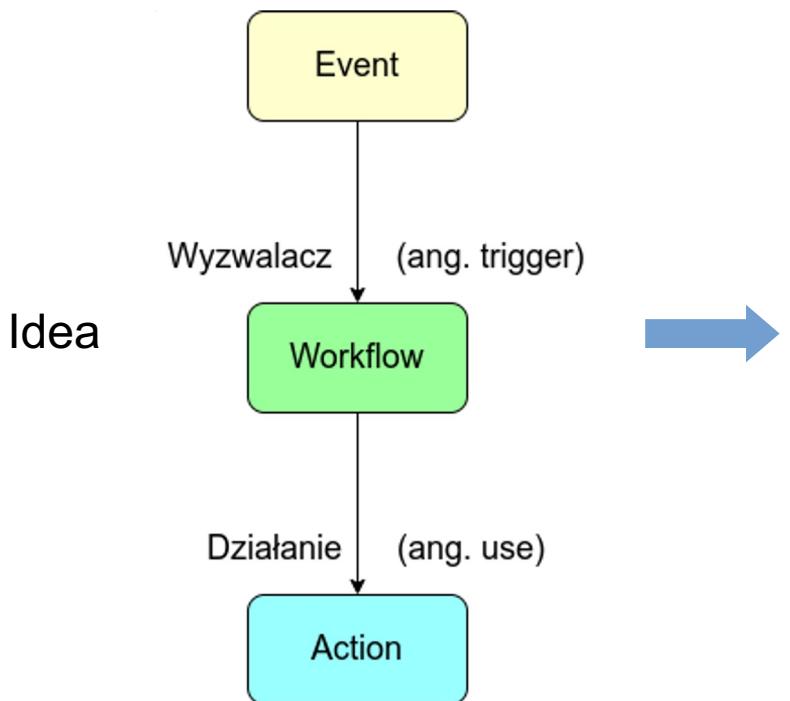
- całkowicie zintegrowany z GitHub,
- Działanie może być powiązane z dowolnym zdarzeniem w ramach posiadanego kontanta GitHub,
- zbiór definicji “Workflow” rozwijana przez społeczność,
- bogaty (i wciąż rosnący zbiór opisów “Actions”,
- brak ograniczeń na wymagany typ platformy (realizacji działań w oparciu o VM Linux, Windows lub MacOS), język programowania czy też docelowe środowisko chmurowe.

## PFSwChO – Laboratorium 6

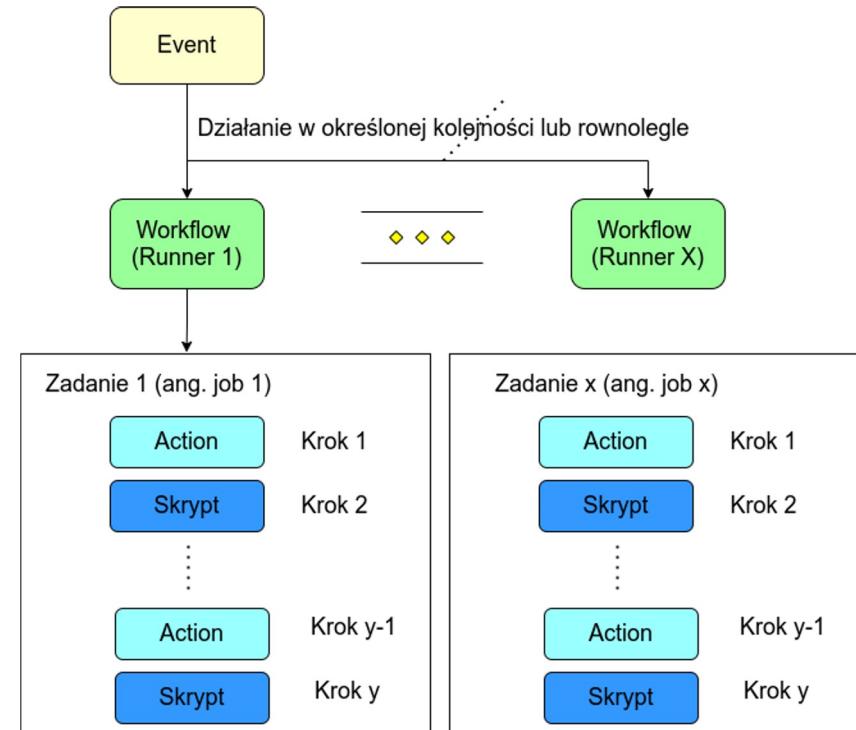
### GitHub Actions – podstawy, cz. I

<https://docs.github.com/en/actions>

<https://github.com/features/actions>



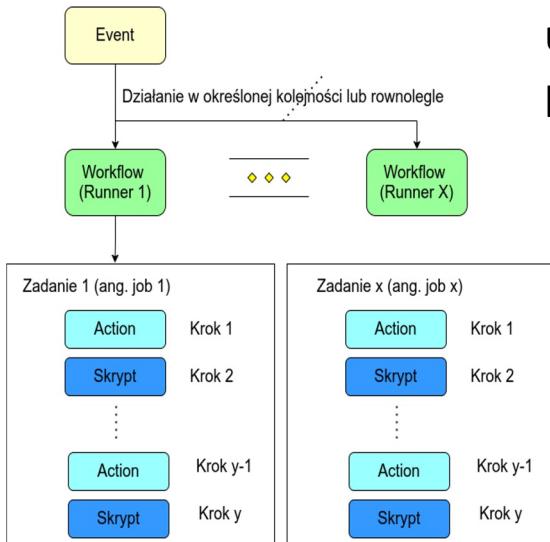
### Realizacja



# PFSwChO – Laboratorium 6

## GitHub Actions – podstawy, cz. II

**Event** – określone zdarzenie w ramach danego repozytorium, które zostało zdefiniowane jako wyzwalacz w danym workflow.



**Workflow** – zautomatyzowany ciąg działań, realizowany w wyniku wystąpienia określonego zdarzenia (ang. event) w danym repozytorium, uruchamiany manualnie bądź wywołany przez zewnętrzne działania planowe (ang. scheduler).

**Job** – zbiór kroków (ang. Steps) do wykonania w ramach danego workflow. Każdy krok może mieć formę skryptu (ang. script execution) lub predefiniowanej akcji (ang. action run). Poszczególne kroki (ang. steps) są wykonywane w kolejności wystąpienia i są wzajemnie zależne. Dane w obrębie jednego runnera mogą być współdzielone przez poszczególne kroki.

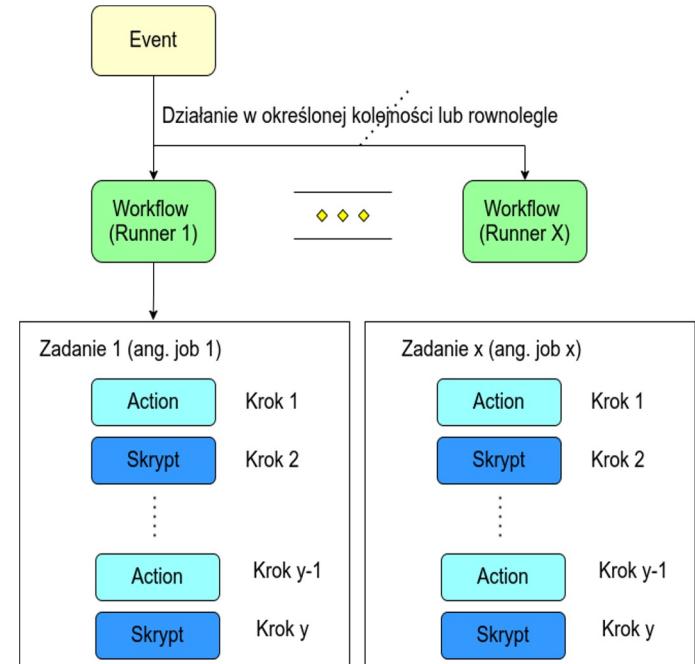
# PFSwChO – Laboratorium 6

## GitHub Actions – podstawy, cz. III

**Runner** – serwer, na którym wykonywane są zadania opisane w danym workflow (po pojawienniu się właściwego event-u). Każdy runner wykonuje jedno zadanie (job). Obecnie dostępne są: Ubuntu Linux, Microsoft Windows oraz macOS.

<https://docs.github.com/en/actions/using-github-hosted-runners/about-github-hosted-runners>

Action – aplikacja dedykowana dla platformy GitHub Actions, która wykonuje zestaw często powtarzanych zadań (np. zdobudowanie i przesłanie obrazów do wskazanego repozytorium). Można opracować własne Action lub (zdecydowanie częstsze rozwiązanie), wykorzystać Actions zgromadzone na **GitHub Marketplace**.



# PFSwChO – Laboratorium 6

## GitHub Actions Marketplace

The screenshot shows the GitHub Actions Marketplace interface. At the top, there is a search bar with the query "docker". A red arrow points from the text "Marketplace / Search results" to the search bar. Below the search bar, there are tabs for "Types" (highlighted), "Actions" (selected), and "Apps". On the left, there is a sidebar with categories: API management, Chat, Code quality, Code review, Continuous integration, Dependency management, Deployment, IDEs, Learning, and Localization. The main area is titled "Actions" and contains the sub-section "Actions". It says "An entirely new way to automate your development workflow." and shows "555 results for 'docker' filtered by Actions". There are five GitHub Actions listed, each with a ship icon, the name, the author, a brief description, and the number of stars:

Action Name	Author	Description	Stars
Docker Buildx Bake	By docker	GitHub Action to use Docker Buildx Bake as a high-level build command	72 stars
Build and push Docker images	By docker	Build and push Docker images with Buildx	2.8k stars
Docker Metadata action	By docker	GitHub Action to extract metadata (tags, labels) for Docker	454 stars
Docker Login	By docker	GitHub Action to login against a Docker registry	535 stars
Docker Setup QEMU	By docker	Install QEMU static binaries	235 stars
Docker Setup Buildx	By docker	Set up Docker Buildx	513 stars

### UWAGA:

Można wykorzystywać (poza klasycznymi skryptami) samodzielnie przygotowane receptury „Actions”

<https://docs.github.com/en/actions/creating-actions/about-custom-actions>

# PFSwChO – Laboratorium 6

## GitHub Actions – PRZYKŁAD, cz. I

<https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions#understanding-the-workflow-file>

SenatorP51 / **cloudlab** Public

Actions

Choose a workflow

Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a workflow to get started.

Skip this and [set up a workflow yourself](#) →

Search workflows

Suggested for this repository

- Docker image**  
By GitHub Actions
- SLSA Generic generator**  
By Open Source Security Foundation (OpenSSF)
- Jekyll using Docker image**  
By GitHub Actions

# PFSwChO – Laboratorium 6

## GitHub Actions – PRZYKŁAD, cz. II

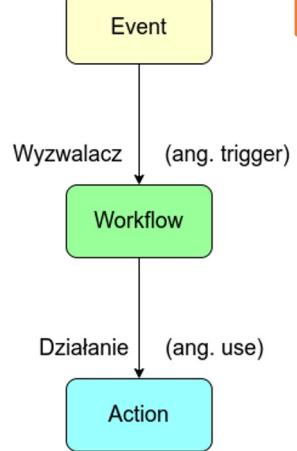
Przebieg działania w ramach GitHub Action definiowany jest w pliku yaml

The screenshot shows a GitHub repository's Actions page. At the top, there are navigation links: Code, Issues, Pull requests, Actions (which is highlighted), Projects, Wiki, Security, Insights, and Settings. Below the header, the word "Actions" is displayed, followed by a "New workflow" button. On the left, there's a sidebar with "All workflows" and a search bar containing the text "CI". Under the search bar, there are links for "Management" and "Caches". The main content area is titled "CI" and contains a link to "main1.yml". Below this, it says "0 workflow runs" and provides a note: "This workflow has a `workflow_dispatch` event trigger." In the bottom right corner of the main content area, there is a blue icon consisting of four circles connected by lines, forming a play button shape. Below the icon, the text "This workflow has no runs yet." is displayed.

# PFSwChO – Laboratorium 6

## GitHub Actions – PRZYKŁAD, cz. III

- GitHub triggered events:  
push, pull\_request,  
public
- Scheduled events: `schedule`
- Manually triggered:  
`workflow_dispatch`  
(external systems)

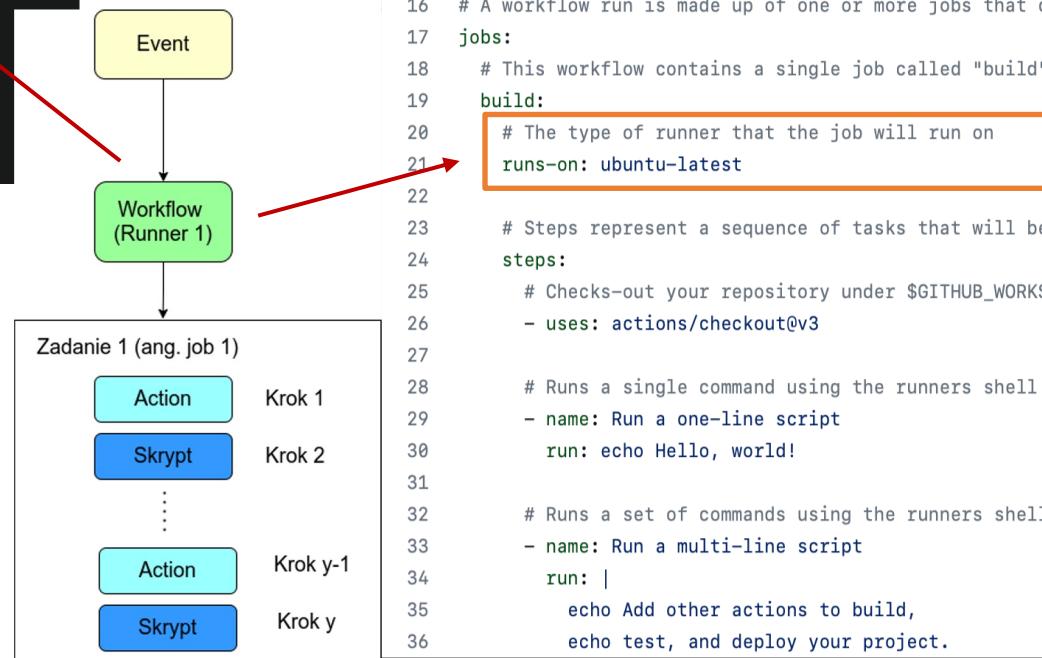


```
1 # This is a basic workflow to help you get started with Actions
2
3 name: CI
4
5 # Controls when the workflow will run
6 on:
7   # Triggers the workflow on push or pull request events but only for the main branch
8   push:
9     branches: [ main ]
10  pull_request:
11    branches: [ main ]
12
13 # Allows you to run this workflow manually from the Actions tab
14 workflow_dispatch:
15
16 # A workflow run is made up of one or more jobs that can run sequentially or in parallel
17 jobs:
18   # This workflow contains a single job called "build"
19   build:
20     # The type of runner that the job will run on
21     runs-on: ubuntu-latest
22
23   # Steps represent a sequence of tasks that will be executed as part of the job
24   steps:
25     # Checks-out your repository under $GITHUB_WORKSPACE, so your job can access it
26     - uses: actions/checkout@v3
27
28     # Runs a single command using the runners shell
29     - name: Run a one-line script
30       run: echo Hello, world!
31
32     # Runs a set of commands using the runners shell
33     - name: Run a multi-line script
34       run: |
35         echo Add other actions to build,
36         echo test, and deploy your project.
```

# PFSwChO – Laboratorium 6

## GitHub Actions – PRZYKŁAD, cz. IV

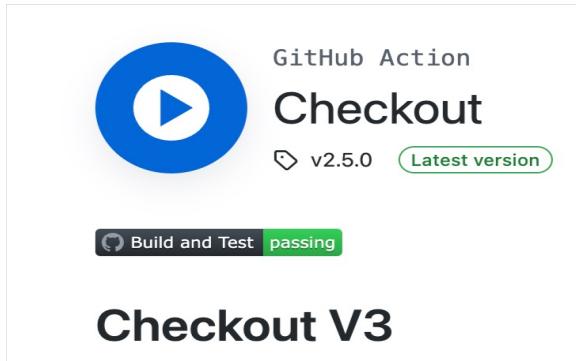
```
runs-on: ${{ matrix.os }}  
  
strategy:  
  matrix:  
    node-version: [8.x, 10.x, 12.x]  
    os: [macos-latest, windows-latest, ubuntu-18.04]  
  
steps:  
- uses: actions/checkout@v1  
- name: Use Node.js ${{ matrix.node-version }}  
  uses: actions/setup-node@v1  
  with:  
    node-version: ${{ matrix.node-version }}  
- name: npm install, build, and test  
  run: |  
    npm ci  
    npm run build --if-present  
    npm test
```



Przykład tzw. Workflow  
“macierzowego” (ang.  
matrix workflow)

# PFSwChO – Laboratorium 6

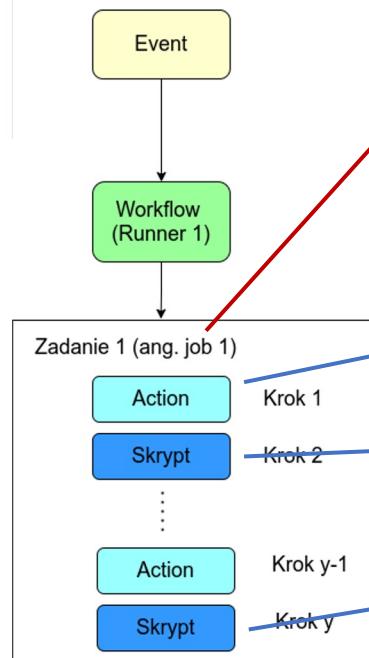
## GitHub Actions – PRZYKŁAD, cz. V



### Checkout V3

<https://github.com/marketplace/actions/checkout>

**action – uses:  
skrypt – run:**



```
1 # This is a basic workflow to help you get started with Actions
2
3 name: CI
4
5 # Controls when the workflow will run
6 on:
7   # Triggers the workflow on push or pull request events but only for the main branch
8   push:
9     branches: [ main ]
10  pull_request:
11    branches: [ main ]
12
13 # Allows you to run this workflow manually from the Actions tab
14 workflow_dispatch:
15
16 # A workflow run is made up of one or more jobs that can run sequentially or in parallel
17 jobs:
18   # This workflow contains a single job called "build"
19   build:
20     # The type of runner that the job will run on
21     runs-on: ubuntu-latest
22
23   # Steps represent a sequence of tasks that will be executed as part of the job
24   steps:
25     # Checks-out your repository under $GITHUB_WORKSPACE, so your job can access it
26     - uses: actions/checkout@v3
27
28     # Runs a single command using the runners shell
29     - name: Run a one-line script
30       run: echo Hello, world!
31
32     # Runs a set of commands using the runners shell
33     - name: Run a multi-line script
34       run: |
35         echo Add other actions to build,
36         echo test, and deploy your project.
```

# PFSwChO – Laboratorium6

## ZADANIE

```
❯ ~/ gh repo create
? What would you like to do? Push an existing local repository to GitHub
? Path to local repository .
current directory is not a git repository. Run `git init` to initialize it
❯ ~/ gh repo create
? What would you like to do? Create a new repository on GitHub from scratch
? Repository name GALab6
? Description GitHub Actions tests
? Visibility Public
? Would you like to add a README file? Yes
? Would you like to add a .gitignore? Yes
? Choose a .gitignore template Actionscript
? Would you like to add a license? No
? This will create "GALab6" as a public repository on GitHub. Continue? Yes
✓ Created repository SenatorP51/GALab6 on GitHub
? Clone the new repository locally? Yes
Klonowanie do „GALab6”...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Pobieranie obiektów: 100% (4/4), gotowe.
```

1. Należy utworzyć publiczne repozytorium o nazwie GALab6 wykorzystując GitHub CLI

Potwierdzenie poprawności

```
❯ ~/ gh repo list | grep GALab6
SenatorP51/GALab6 GitHub Actions tests public 2022-11-1
❯ ~/ cd ~/GALab6
❯ ~/GALab6/ [main] ls -al
total 16
drwxr-xr-x 5 slawek staff 160 14 lis 12:22 .
drwxr-x---+ 51 slawek staff 1632 14 lis 12:25 ..
drwxr-xr-x 12 slawek staff 384 14 lis 12:22 .git
-rw-r--r-- 1 slawek staff 350 14 lis 12:22 .gitignore
-rw-r--r-- 1 slawek staff 30 14 lis 12:22 README.md
❯ ~/GALab6/ [main] █
```

# PFSwChO – Laboratorium6

## ZADANIE

2. Należy do utworzonego katalogu lokalnego skopiować zawartość przykładowej aplikacji frontend-owej. Kody dostępne na moodle jako plik **Lab6.zip**.

**UWAGA: na wykorzystywany systemie MUSI być zainstalowane narzędzie git**

```
 MacBook-Pro:~/GALab6/ [main] git status
Na gałęzi main
Twoja gałąź jest na bieżąco z „origin/main”.

Nieśledzone pliki:
  (użyj „git add <plik>...”, żeby uwzględnić, co zostanie złożone)
    .github/
    Dockerfile
    build/
    dockerfile.v1
    package-lock.json
    package.json
    public/
    src/

nie dodano nic do złożenia, ale są nieśledzone pliki (użyj „git add”, aby śledzić)
```

# PFSwChO – Laboratorium6

## ZADANIE

### 3. Należy utworzyć nowy „commit” dla nowej zawartości repozytorium

```
[apple ~ /GALab6/ [main] git add *
[apple ~ /GALab6/ [main+] git status
Na gałęzi main
Twoja gałąź jest na bieżąco z „origin/main”.
```

Zmiany do złożenia:

(użyj „git restore --staged <plik>...”, aby wycofać)

```
nowy plik: Dockerfile
nowy plik: build/asset-manifest.json
nowy plik: build/favicon.ico
nowy plik: build/index.html
... 100
```

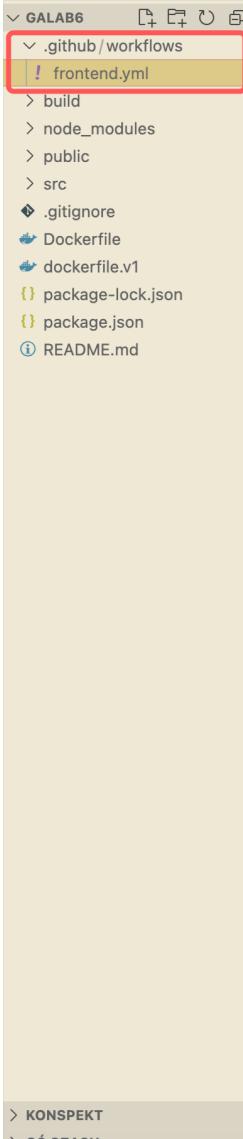


```
[apple ~ /GALab6/ [main+] git commit -a -m "source files added"
[main 2bc0984] source files added
 37 files changed, 16493 insertions(+)
  create mode 100755 Dockerfile
  create mode 100755 build/asset-manifest.json
  create mode 100755 build/favicon.ico
  create mode 100755 build/index.html
```

# PFSwChO – Laboratorium6

## ZADANIE

4. Proszę otworzyć zawartość repozytorium w edytorze i przeanalizować przedstawiony niżej plik definiujący Github Actions workflow.



```
.github > workflows > ! frontend.yml
1 name: GitHub Action flow to Docker hub
2
3 on:
4   push:
5     branches: [main]
6
7 jobs:
8   build-push-images:
9     name: Budowa i publikacja obrazu na repozytorium
10    runs-on: ubuntu-latest
11
12 steps:
13
14   - name: Checkout code
15     uses: actions/checkout@v2
16
17   - name: Buildx set-up
18     id: buildx
19     uses: docker/setup-buildx-action@v1
20
21   - name: Login to DockerHub
22     uses: docker/login-action@v1
23     with:
24       username: ${{ secrets.DOCKER_HUB_USERNAME }}
25       password: ${{ secrets.DOCKER_HUB_ACCESS_TOKEN }}
26
27
28   - name: Build and push
29     id: docker_build
30     uses: docker/build-push-action@v2
31     with:
32       context: .
33       file: ./dockerfile.v1
34       push: true
35       tags: |
36         spg51/lab:ghfront1
37
```

# PFSwChO – Laboratorium6

## ZADANIE

UWAGA: ponownie użyto personal-access-token

### 4. Na koniec należy wykonać polecenie push

```
apple ~ /GALab6/ [main] git push
Username for 'https://github.com': SenatorP51
Password for 'https://SenatorP51@github.com':
Wymienianie obiektów: 41, gotowe.
Zliczanie obiektów: 100% (41/41), gotowe.
Kompresja delt z użyciem do 10 wątków
Kompresowanie obiektów: 100% (39/39), gotowe.
Zapisywanie obiektów: 100% (40/40), 339.30 KiB | 9.69 MiB/s, gotowe.
Razem 40 (delta 2), użyte ponownie 0 (delta 0), paczki użyte ponownie 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/SenatorP51/GALab6.git
  dbdeec7..2bc0984 main -> main
```

### 5. Na koniec należy wykonać polecenie push dla deklaracji GitHub Actions

```
apple ~ /GALab6/ [main+] git add .github
apple ~ /GALab6/ [main+] git commit -a -m "GA workflow added"
[main 58f3d07] GA workflow added
  1 file changed, 37 insertions(+)
   create mode 100755 .github/workflows/frontend.yml
apple ~ /GALab6/ [main] git push
Wymienianie obiektów: 6, gotowe.
Zliczanie obiektów: 100% (6/6), gotowe.
Kompresja delt z użyciem do 10 wątków
Kompresowanie obiektów: 100% (3/3), gotowe.
Zapisywanie obiektów: 100% (5/5), 766 bajtów | 766.00 KiB/s, gotowe.
Razem 5 (delta 1), użyte ponownie 0 (delta 0), paczki użyte ponownie 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/SenatorP51/GALab6.git
  2bc0984..58f3d07 main -> main
```

## PFSwChO – Laboratorium6

### Do zapoznania się do kolejnego laboratorium

1. Proszę zapoznać się z efektami wykonania polecenia push na GitHub. Powinien uruchomić się proces budowania obrazów zgodnie z przesłaną definicją. Proces ten zakończył się niepowodzeniem, dlaczego ??? Co należy poprawić ???
2. Należy samodzielnie przyjrzeć się przykładom za poniższego adresu internetowego:

<https://docs.github.com/en/actions/quickstart>

<https://github.com/actions/starter-workflows>