

CI/CD aplikacji WEB

Artur Pietrzkiewicz

Michał Sarota

Adam Plotnik

Szymon Rewilak

Informatyka Techniczna WiMiP

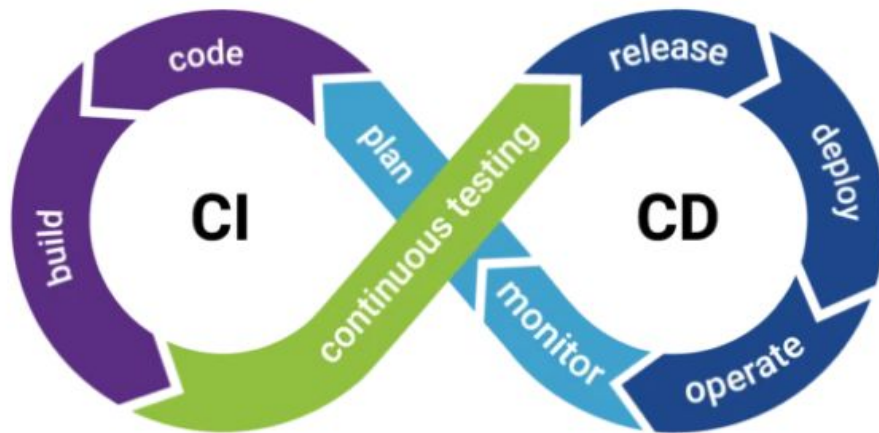
II stopień I semestr

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie
AGH University of Science and Technology

18.05.2023

CI/CD - podstawowe informacje

Continuous integration (CI) i Continuous Delivery (CD) to zbiór zasad, wytycznych, kultura pracy i kolekcja dobrych praktyk dotyczących pracy nad projektami informatycznymi.



CI/CD - podstawowe informacje

Dzięki nim zespół developerów ma możliwość częstszego dostarczania pewnych, przetestowanych i sprawdzonych zmian w kodzie.

Implementacja tych praktyk często nazywana jest **CI/CD pipeline** i jest uważana za jeden z najlepszych i najefektywniejszych sposobów pracy nad projektami informatycznymi i ich rozwojem.

Continuous integration – ciągła integracja

Jest to filozofia wprowadzania zmian w kodzie aplikacji, która polega na tym, że zespół developerów wprowadza szereg drobnych zmian w kodzie, które następnie okresowo sprawdzane są w repozytorium wersji kodu. Celem stosowania CI jest osiągnięcie spójnego i zautomatyzowanego sposobu budowania i testowania aplikacji. Dzięki konsekwentnemu procesowi integracji zespoły częściej wprowadzają zmiany w kodzie częściej, co prowadzi do lepszej współpracy i lepszej jakości oprogramowania.

Continuous delivery – ciągłe dostarczanie

Continuous delivery rozpoczyna się tam, gdzie continuous integration się kończy. CD automatyzuje proces wdrażania aplikacji i wprowadzanych zmian w kodzie do przygotowanej infrastruktury serwerowej. Kluczową cechą CD jest zadbanie o automatyzację procesu dostarczania zmian dla wszystkich wykorzystywanych środowisk i wykonanie niezbędnych dodatkowych mechanizmów takich jak np. wysłanie żądania do serwera WWW, wykonanie zapytań SQL, wysłanie powiadomienia czy nawet restart / przeładowanie ustawień lub uruchomionych usług.

Zalety CI/CD

- **Automatyzacja procesu wdrażania:** CI/CD umożliwia automatyzację procesu wdrażania aplikacji, co przyspiesza tempo dostarczania nowych funkcji i poprawek.
- **Częstsze dostarczanie wartości:** Dzięki CI/CD zespoły deweloperskie mogą częściej dostarczać wartość biznesową. Szybkie iteracje i dostarczanie nowych funkcji lub poprawek często prowadzi do większej satysfakcji klientów i większej konkurencyjności na rynku.
- **Szybka wykrywalność błędów:** CI/CD umożliwia wykrycie błędów w oprogramowaniu na wczesnym etapie.
- **Lepsza współpraca zespołów:** CI/CD promuje lepszą współpracę między członkami zespołu deweloperskiego.
- **Ułatwione skalowanie:** Dzięki CI/CD proces skalowania aplikacji może być bardziej zautomatyzowany.

Wady CI/CD

- **Skomplikowana konfiguracja początkowa:** Implementacja CI/CD wymaga pewnego wysiłku w zakresie konfiguracji początkowej. Narzędzia, skrypty i infrastruktura muszą być dostosowane do specyficznych potrzeb projektu.
- **Wymaga odpowiednich testów:** Aby CI/CD było skuteczne, konieczne jest posiadanie odpowiednich testów automatycznych. Bez solidnego zestawu testów jednostkowych, integracyjnych i wydajnościowych, automatyczne wdrażanie może prowadzić do pojawienia się błędów w produkcyjnym środowisku.
- **Potrzeba stabilnej infrastruktury:** CI/CD wymaga stabilnej infrastruktury i narzędzi, które mogą obsłużyć regularne budowanie, testowanie i wdrażanie aplikacji. Dlatego wymaga to odpowiedniego nakładu czas

Najpopularniejsze technologie CI/CD

Jenkins jest jednym z najpopularniejszych narzędzi CI/CD. Jest to oprogramowanie open-source, które umożliwia automatyzację procesu integracji, testowania i wdrażania aplikacji.



Jenkins

Wady i zalety Jenkins

Zalety

1. Open-source i rozwijana społeczność
2. Rozległe wsparcie dla integracji
3. Skalowalność i elastyczność
4. Automatyzacja i ciągła dostawa

Wady


1. Skomplikowana konfiguracja
2. Zużycie zasobów
3. Brak wbudowanej obsługi kontenerów
4. Nieintuicyjny interfejs użytkownika

Tworzenie pipeline w Jenkins

Dodanie nowego joba i wybranie pipeline


Enter an item name

» Required field




Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used to build other Jenkins projects.




Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) a type.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

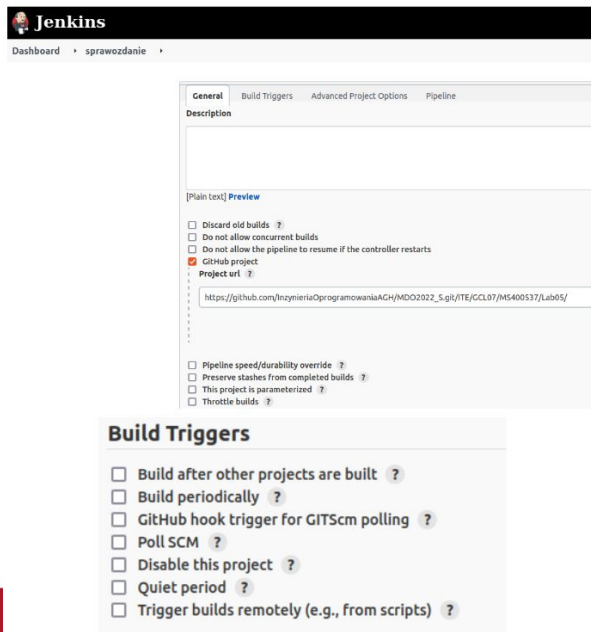


Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates items as long as they are in different folders.

Tworzenie pipeline w Jenkins

Ustawienie URL projektu, który będzie się budował oraz triggerów, które będą włączać budowę



Jenkins

Dashboard » sprawozdanie »

General Build Triggers Advanced Project Options Pipeline

Description

[Plain text] [Preview](#)

☐ Discard old builds ?

☐ Do not allow concurrent builds

☐ Do not allow the pipeline to resume if the controller restarts

☒ GitHub project

Project url ?

`https://github.com/InzynieriaOprogramowaniaAGH/MDO2022_S_git/ITE/CCL07/MS400537/Lab05/`

☐ Pipeline speed/durability override ?

☐ Preserve stashes from completed builds ?

☐ This project is parameterized ?

☐ Throttle builds ?

Build Triggers

☐ Build after other projects are built ?

☐ Build periodically ?

☐ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

☐ Disable this project ?

☐ Quiet period ?

☐ Trigger builds remotely (e.g., from scripts) ?

Tworzenie pipeline w Jenkins

Ustawienie potrzebnych informacji odnośnie repozytorium, z którego projekt będzie się budował

Pipeline

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

`https://github.com/InzynieriaOprogramowaniaAGH/MDO2022_S_git`

Credentials ?

- none - Add

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

`*/MS400537_lab5`

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add

Script Path ?

`ITE/GCL07/MS400537/Lab05/Jenkinsfile`

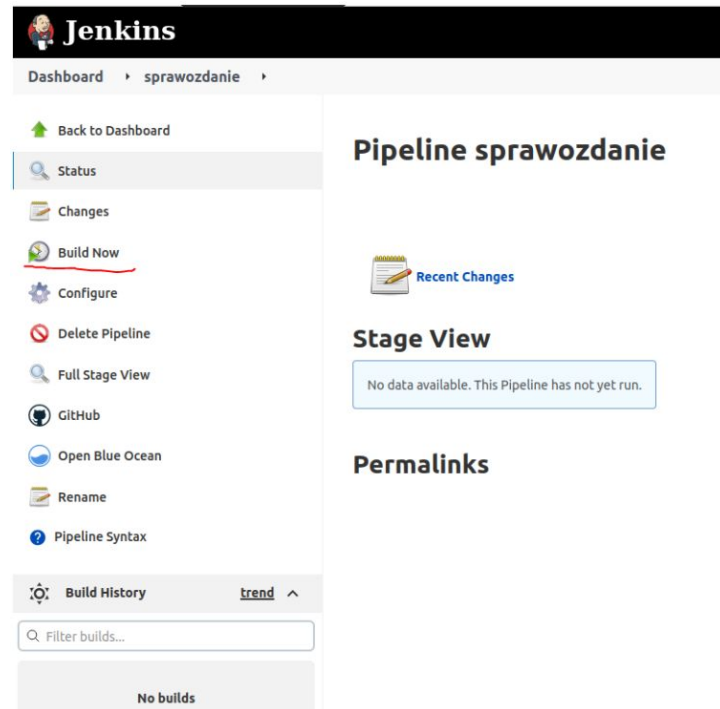
☒ Lightweight checkout ?

Pipeline Syntax

Save Apply

Tworzenie pipeline w Jenkins

Tworzenie buildu



The screenshot shows the Jenkins web interface for a pipeline report. The left sidebar contains a menu with the following items: Back to Dashboard, Status, Changes, Build Now (highlighted with a red underline), Configure, Delete Pipeline, Full Stage View, GitHub, Open Blue Ocean, Rename, and Pipeline Syntax. The main content area is titled 'Pipeline sprawozdanie' and includes a 'Recent Changes' section with a pencil icon. Below this is a 'Stage View' section with a message: 'No data available. This Pipeline has not yet run.' The 'Permalinks' section is also visible. At the bottom, there is a 'Build History' section with a search bar and a message: 'No builds'.

Tworzenie pipeline w Jenkins

Wynik buildu, który zawierał 5 stage'ów. Po jego wykonaniu, dostępne do pobrania są artefakty

Last Successful Artifacts

-  `esm.js` 1.66 KB [view](#)
-  `index.js` 2.13 KB [view](#)
-  `rollup.config.js` 532 B [view](#)

 **Recent Changes**

Stage View



Azure DevOps

Azure DevOps to platforma usług dostarczana przez Microsoft. Wspiera zarządzanie kodem źródłowym, śledzenie zadań, wdrażania, monitorowanie.



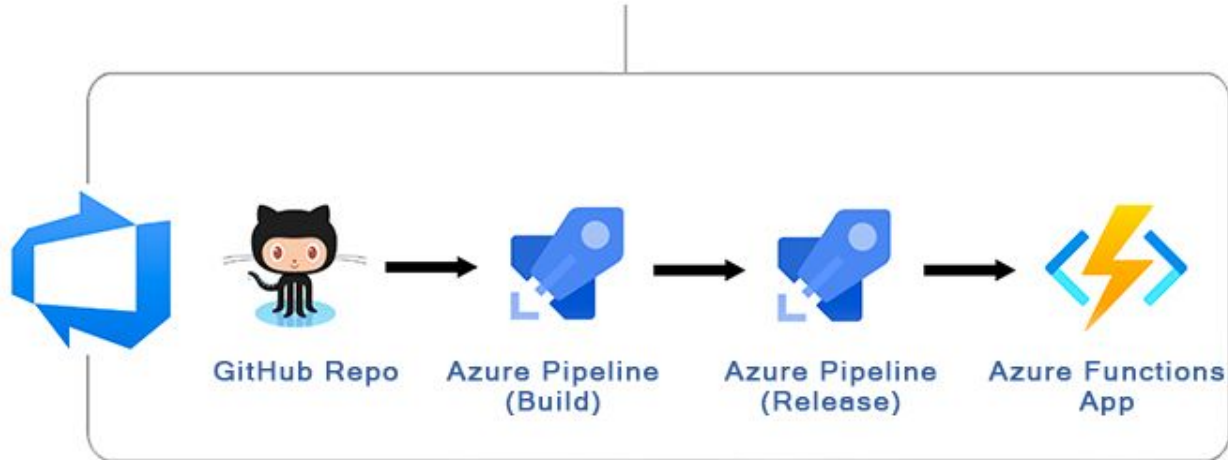
Azure DevOps

Mocne strony Azure DevOps

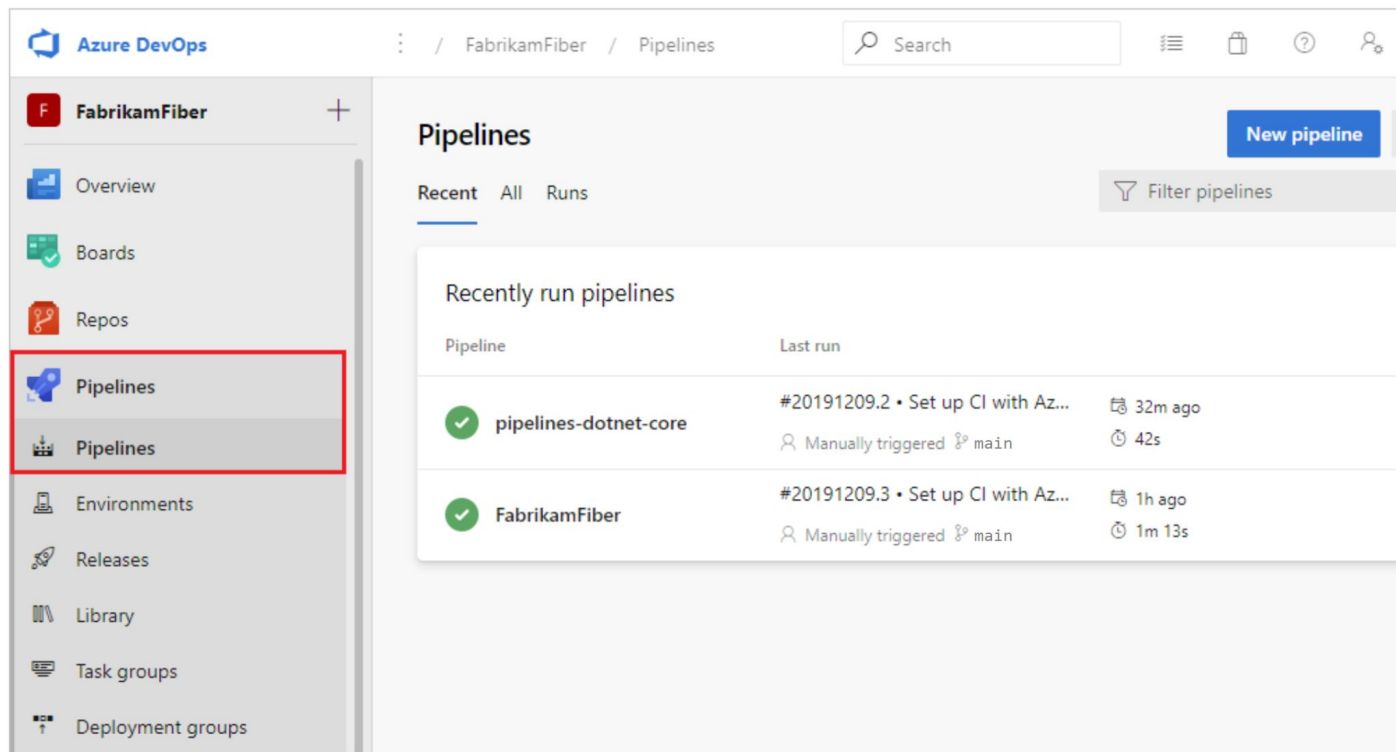
- Wsparcie dla wielu technologii
- Progressive Deployment
- Integracja z platformą chmurową Azure
- Funkcje raportujące i analizy
- Integracja z wieloma platformami

Azure DevOps Pipeline

Azure DevOps Pipeline.



Zarządzanie pipeline'ami



Azure DevOps / FabrikamFiber / Pipelines

Search





FabrikamFiber

- Overview
- Boards
- Repos
- Pipelines**
- Pipelines
- Environments
- Releases
- Library
- Task groups
- Deployment groups

Pipelines New pipeline

Recent All Runs Filter pipelines

Recently run pipelines

Pipeline	Last run
 pipelines-dotnet-core	#20191209.2 • Set up CI with Az... Manually triggered  main 32m ago 42s
 FabrikamFiber	#20191209.3 • Set up CI with Az... Manually triggered  main 1h ago 1m 13s

Uruchamianie pipeline

Pipelines

Recent

All

Runs

New pipeline

Filter pipelines

Recently run pipelines

Pipeline	Last run
<div>✓</div> <div>pipelines-dotnet-core</div>	<div>#20191209.2 • Set up CI with Az...</div> <div>36m ago</div> <div>Manually triggered</div> <div>main</div> <div>42s</div>
<div>✓</div> <div>FabrikamFiber</div>	<div>#20191209.3 • Set up CI with Az...</div> <div>1h ago</div> <div>Manually triggered</div> <div>main</div> <div>1m 13s</div>

Archiwizacja pipeline

Pipelines New pipeline ⋮

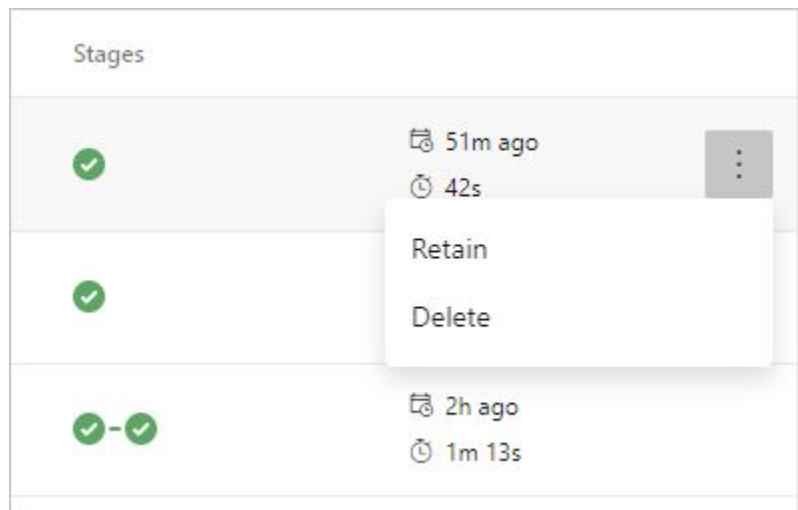
Recent All **Runs** ⌵

⌵ Filter by keywords State Repository Requested for Tags X





All pipeline runs

Description	Stages	
Set up CI with Azure Pipelines #20191209.2 on pipelines-dotnet-core main d4964...	✓	44m ago 42s
Set up CI with Azure Pipelines #20191209.1 on pipelines-dotnet-core main d4964...	✓	53m ago 46s
Set up CI with Azure Pipelines #20191209.3 on FabrikamFiber main 2b4b23c	✓ - ✓	1h ago 1m 13s
Set up CI with Azure Pipelines #20191209.2 on FabrikamFiber main 2b4b23c	✓ - ✓	1h ago 1m 55s
Set up CI with Azure Pipelines #20191209.1 on FabrikamFiber main 2b4b23c	✓ - ✓	1h ago 1m 6s

Archiwizowane pipeline mogą zostać powtórzone



Stage w Azure DevOPS

<div>Stages</div> <div><u>Jobs</u></div>			
Name	Status	Stage	Duration
 Build	Success	Build	 40s
 DeployWeb	Success	Deploy	 10s

Artefakty

All definitions > Fabrikam

Pipeline Tasks Variables Retr...

Artifacts **+ Add**

SampleWebAppWithTestsBuild

Schedule not set

Add artifact

Source type

✓ Build Azure Repos... GitHub TFVC

3 more artifact types ▾

Project * ⓘ

Fabrikam

Source (Build definition) * ⓘ

Fabrikam.CI

Default version * ⓘ

Latest

Latest from build definition default branch with tags

Latest from specific branch with tags

Specific version

Specify at the time of release creation

Add

Artefakty cz. 2

Artifact

Build - TestsWebAppBuild

Project

Fabrikam

Source (Build definition)

TestsWebAppBuild

Delete

...

✓ Mark primary

Inne popularne technologie CI/CD

1. GitLab CI/CD
2. Travis CI
3. CircleCI
4. Bamboo
5. TeamCity



Travis CI



Bamboo



TeamCity

CI/CD w największych firmach



CI/CD w dużych firmach

- Pełna automatyzacja na każdym etapie CI/CD
 - Statystyczna analiza kodu
 - W pełni zautomatyzowane testowanie
 - Automatyczne wykrywanie błędów
- Obsługa ogromnych projektów
- Infrastruktura jako kod (*infrastructure as code, IaC*)
- Progressive Deployment
- Percentage Rollout: tylko określona liczba użytkowników otrzymuje nowe wersje aplikacji, po czym następuje monitoring
- Setki lub tysiące deploymentów dziennie
- Zespoły odpowiadają za wprowadzanie zmian na wielu frontach równocześnie

Dziękujemy za uwagę!

