

Jenkins, CI



Tomasz Woliński

StormIT.pl
fb.com/stormitPL

Poznajmy się

Agenda

- Continuous Integration (**CI**)
- Jenkins
- System zadań w Jenkins

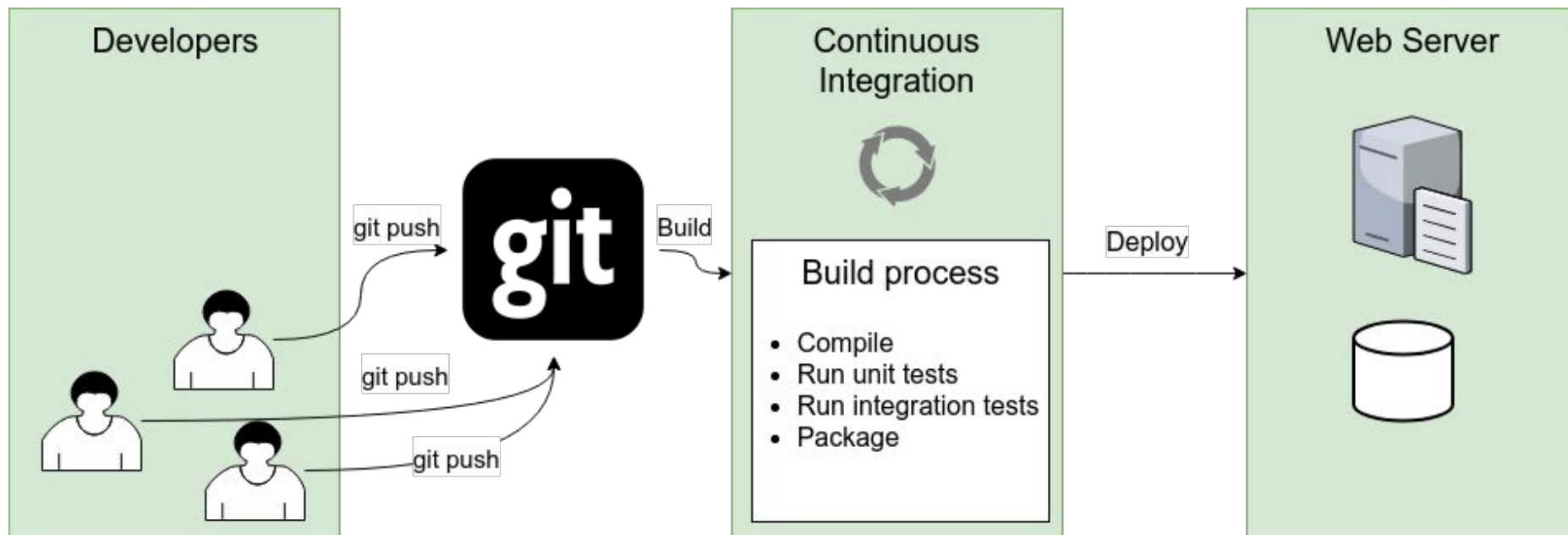
Continuous Integration

Wprowadzenie do CI

Ciągła Integracja

- ✓ Praktyka programistyczna, w której zmiany wprowadzane przez różnych programistów są łączone możliwie jak najczęściej
- ✓ Poszczególne zmiany są weryfikowane przez automatyczny proces budowania i testowania aplikacji

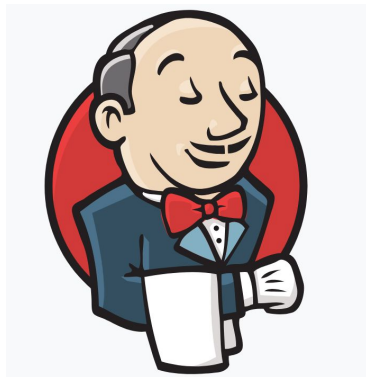
Proces ciągłej integracji



Korzyści z ciągłej integracji

- ✓ Częstsze i szybsze integrowanie małych zmian
 - łatwiej wykryć błędy
 - szybciej się o nich dowiemy
- ✓ Szybsze testowanie zmian w środowisku zbliżonym do produkcyjnego
- ✓ Automatyzacja
 - szybciej, taniej, bardziej przewidywalnie
- ✓ Łatwiejsza refaktoryzacja
- ✓ Możliwość dostarczenia aplikacji z dowolnego punktu w historii
 - np. deploy aplikacji sprzed miesiąca / rollback

Jenkins



- <https://jenkins.io/>
- Java
- <https://github.com/jenkinsci/jenkins>

- Open Source, MIT
- dawniej Hudson
- alternatywy: Bamboo, GitLab, TeamCity ...



Jenkins instalacja

- `docker run -p 8080:8080 -p 50000:50000
jenkins/jenkins:lts`
- `docker run
-p 8080:8080
-p 50000:50000
-v jenkins_home:/var/jenkins_home
jenkins/jenkins:lts`

Jenkins instalacja

- hasło z logów dockera
- instalujemy **BEZ** wtyczek

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

```
/var/jenkins_home/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

Continue

Jenkins – standardowe możliwości

- ✓ Budowanie aplikacji
- ✓ Uruchomienie testów jednostkowych, integracyjnych ...
- ✓ Migracja bazy danych
- ✓ Wygenerowanie dokumentacji z kodu
- ✓ Weryfikacja metryk kodu
(code coverage, statyczna analiza kodu)
- ✓ Wysłanie powiadomienia ze statusem (mail, slack ...)
- ✓ Uruchomienie DOWOLNEGO skryptu

Jenkins konfiguracja

Manage Jenkins



Configure System

Configure global settings and paths.



Configure Global Security

Secure Jenkins; define who is allowed to access/use the system.



Configure Credentials

Configure the credential providers and types



Global Tool Configuration

Configure tools, their locations and automatic installers.



Reload Configuration from Disk

Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.



Manage Plugins

Add, remove, disable or enable plugins that can extend the functionality of Jenkins.



System Information

Displays various environmental information to assist trouble-shooting.



System Log

System log captures output from `java.util.logging` output related to Jenkins.



Load Statistics

Check your resource utilization and see if you need more computers for your builds.



Jenkins CLI

Access/manage Jenkins from your shell, or from your script.



Script Console

Executes arbitrary script for administration/trouble-shooting/diagnostics.

Jenkins zadania

Welcome to Jenkins!

Please **create new jobs** to get started.

Enter an item name

» This field cannot be empty, please enter a valid name



Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system.

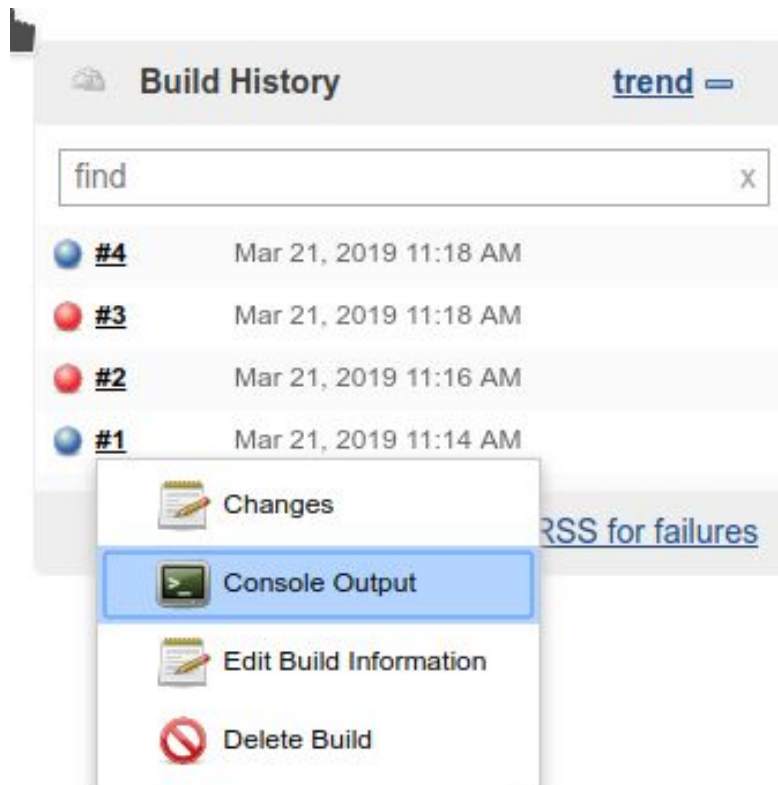
OK

Jenkins – pierwsze zadanie

- ✓ Dodaj zadanie typu *Freestyle project*, które co minutę wyświetli aktualną datę
- ✓ Zweryfikuj działanie zadania w konsoli
- ✓ W ramach testu wykonaj błędny skrypt

All	+				
S	W	Name ↓	Last Success	Last Failure	
		Task - date	2 min 24 sec - #1	11 sec - #2	
Icon: S M L		W Description	%		
		 Build stability: 1 out of the last 2 builds failed.	50		

Jenkins – historia



The screenshot displays the Jenkins 'Build History' interface. At the top, there's a header with a cloud icon, the text 'Build History', and a 'trend' link with a minus sign. Below this is a search bar containing the text 'find' and a close button 'x'. The main content area lists four builds, each with a colored status icon, a build number, and a timestamp:

Status	Build Number	Timestamp
Success (Blue)	#4	Mar 21, 2019 11:18 AM
Failure (Red)	#3	Mar 21, 2019 11:18 AM
Failure (Red)	#2	Mar 21, 2019 11:16 AM
Success (Blue)	#1	Mar 21, 2019 11:14 AM

A context menu is open over the build list, showing four options with corresponding icons:

- Changes (Icon: Notepad and pencil)
- Console Output (Icon: Terminal, highlighted with a blue background)
- Edit Build Information (Icon: Notepad and pencil)
- Delete Build (Icon: Prohibited sign)

At the bottom right of the build history area, there is a link for 'RSS for failures'.

Build triggers







- ✓ Cyklicznie
- ✓ Po zakończeniu innego zadania
- ✓ Po zmianie zewnętrznego zasobu
 - np. mechanizm PULL sprawdzający zmiany w repozytorium GIT
- ✓ Przez ręczne wywołanie adresu URL

Jenkins – zadania zewnętrzny trigger

- ✓ Do istniejącego zadania dodaj nowy wyzwalacz przez adres URL
- ✓ W logach zobacz przez kogo zadanie zostało uruchomione

Jenkins – łańcuch wywołań zadań

- ✓ trigger: *Build after other projects are built*
- ✓ *Post-build Actions: Build other projects*
- ✓ Zadanie: przygotuj łańcuch wywołań składający się z przynajmniej trzech zadań

		<u>Task – chain – 1</u>	1 min 4 sec - #1
		<u>Task – chain – 2</u>	55 sec - #1
		<u>Task – chain – 3</u>	45 sec - #1

Jenkins – łańcuch wywołań zadań

- ✓ poprzedni / następny
zależny projekt

The screenshot shows the Jenkins web interface for a job named "Task - chain - 2". The interface is divided into several sections:

- Navigation Bar:** At the top, it says "Jenkins" and "Task - chain - 2".
- Left Sidebar:** Contains several icons and links for managing the job:
 - Powrót do tablicy (Return to dashboard)
 - Status (Status)
 - Rejestr zmian (Change log)
 - Przestrzeń robocza (Workspace)
 - Uruchom (Build)
 - Usuń Projekt (Delete Project)
 - Konfiguruj (Configure)
 - Rename
- Main Content Area:**
 - Project Task – chain – 2:** The title of the job.
 - Przestrzeń robocza (Workspace):** A link to view the workspace.
 - Rejestr zmian (Change log):** A link to view the change log.
 - Poprzednie zależne projekty (Previous dependent projects):** A link to "Task - chain - 1".
 - Następne zależne projekty (Next dependent projects):** A link to "Task - chain - 3".
 - Odnośniki niezmiennie (Unchanging links):** A section with links to build history:
 - Ostatnie zadanie (#1), 30 sek temu (Last build (#1), 30 seconds ago)
 - Ostatnie stabilne zadanie (#1), 30 sek temu (Last stable build (#1), 30 seconds ago)
 - Ostatnie pomyślne zadanie (#1), 30 sek temu (Last successful build (#1), 30 seconds ago)
 - Ostatnie zakończone zadanie (#1), 30 sek temu (Last finished build (#1), 30 seconds ago)
- Bottom Section:** A "Historia zadań" (Build History) section with a search bar and a list of builds. The first build is labeled "#1" and dated "2019-03-29 09:11". There are also RSS links for "Dla wszystkich" (For all) and "Dla nieudanych" (For failed).

Jenkins – stałe linki

Ostatnie zadanie:

- ostatnie
- stabilne
- pomyślne
- niepomyślne
- zakończone

Odnośniki niezmiennie

- [Ostatnie zadanie \(#3\), 40 sek temu](#)
- [Ostatnie stabilne zadanie \(#2\), 1 min 6 sek temu](#)
- [Ostatnie pomyślne zadanie \(#2\), 1 min 6 sek temu](#)
- [Ostatnie niepomyślne zadanie \(#3\), 40 sek temu](#)
- [Ostatnie zakończone zadanie \(#3\), 40 sek temu](#)

<https://wiki.jenkins.io/display/JENKINS/Terminology>

Jenkins + Git

- ✓ Plugin
- ✓ <https://plugins.jenkins.io/git>



Jenkins + Git, source code management

Source Code Management

☐ None
☒ Git

Repositories

Repository URL

Credentials [Add](#)

[Advanced...](#)

[Add Repository](#)

Branches to build


Branch Specifier (blank for 'any')

[Add Branch](#)

Repository browser

Additional Behaviours [Add](#)

Jenkins + Git, workspace





Jenkins







Jenkins > Git - example >


[Back to Dashboard](#)
[Status](#)
[Changes](#)
Workspace
[Wipe Out Current Workspace](#)
[Build Now](#)
[Delete Project](#)
[Configure](#)
[Rename](#)

Workspace of Git - example on master





 .git		
 src/test/java/pl/stormit		
 .gitignore	Mar 21, 2019 3:17:09 PM	12 B  view
 pom.xml	Mar 21, 2019 3:17:09 PM	887 B  view

 [\(all files in zip\)](#)

Jenkins, Git – Zadanie

- ✓ Zainstaluj plugin dla Git
- ✓ Dodaj nowy projekt, który ściągnie pliki z repozytorium Git
- ✓ Projekt powinien być automatycznie pobierany po każdej zmianie w repozytorium
- ✓ Podejrzyj zmiany z repozytorium z poziomu Jenkins

 [Back to Dashboard](#) [Status](#) [Changes](#) [Workspace](#) [Build Now](#)

Changes

[#2 \(Mar 21, 2019 3:27:08 PM\)](#)

1. n/a — [t.wolinski](#) / [githubweb](#)

Jenkins + Git + Maven

- ✓ Plugin
- ✓ <https://plugins.jenkins.io/maven-plugin>



Jenkins, Maven – instalacja

Add Git ▼

Maven

Maven installations

Add Maven

Maven

Name

☒ Install automatically [?](#)

Jenkins, Maven – testy

The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a red box with the number '2', a search bar, and the user 'Tomek' with a 'log out' link. The breadcrumb trail is 'Jenkins > Maven - example > #4 > Test Result'. On the right of the breadcrumb is a link 'ENABLE AUTO REFRESH'. The left sidebar contains a list of links: 'Back to Project', 'Status', 'Changes', 'Console Output', 'Edit Build Information', 'Delete Build', 'Polling Log', 'Git Build Data', 'No Tags', 'Test Result' (which is highlighted), and 'Redeploy Artifacts'. The main content area is titled 'Test Result' and shows '1 failures (+1)' with a progress bar that is 1/5 red. Below this is a table for the test results:

Module	Fail	(diff)	Total	(diff)
pl.stormit:jenkinsexample	1	+1	5	+1

Below the table is a section titled 'Failed Tests' with a link to [pl.stormit:jenkinsexample](#). Under this link is another table showing the details of the failed test:

Test Name	Duration	Age
pl.stormit:Example1Test.test5	13 ms	1

Jenkins, Maven – zadanie

- ✓ Zainstaluj plugin Maven
- ✓ Skonfiguruj Maven w ustawieniach
- ✓ Dodaj nowy projekt typu *Maven*
- ✓ Projekt powinien pobrać kod źródłowy z repozytorium GIT, zbudować projekt oraz wykonać wszystkie testy
- ✓ Projekt powinien być automatycznie budowany po zmianach w repozytorium
- ✓ Zweryfikuj wyniki testów

CI – Dobre praktyki

- ✓ Pełna automatyzacja (budowanie aplikacji; wszystkiego)
- ✓ Automatyczna weryfikacja aplikacji (testy)
- ✓ Testowanie aplikacji po każdej możliwie małej zmianie (*commit*)
- ✓ Powiadomienia o niepowodzeniu oraz reagowanie na nie
 - integracja ze slackiem
- ✓ Optymalizacja szybkości całego procesu
 - możliwie szybka informacja zwrotna
- ✓ Środowisko testowe zbliżone do produkcyjnego



Thanks!!

Pytania?

Znajdziesz mnie na:
StormIT.pl