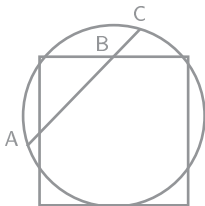


Inżynieria oprogramowania

Radosław Klimek

2015-23



<http://home.agh.edu.pl/rklimek>

1 Potok CI/CD & DevOps

1 Potok CI/CD & DevOps

Potok CI/CD & DevOps



Gustave CAILLEBOTTE: *Cykliniarze*

Ciągła integracja

Definicja

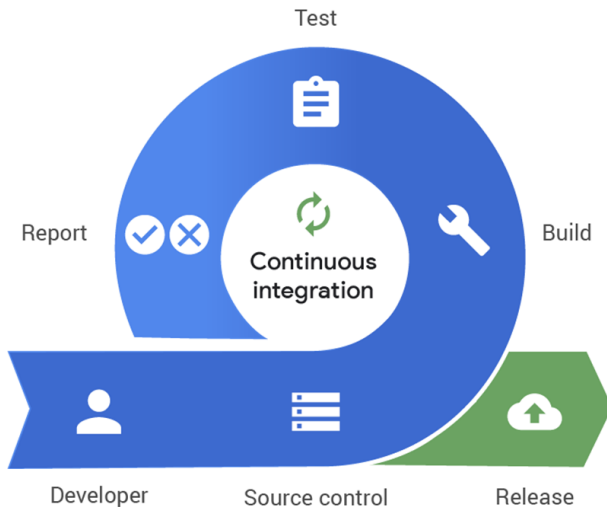
Ciągła integracja CI (ang. *Continuous Integration*), praktyka polegająca na jak najczęstszej integracji zmian z główną gałęzią oprogramowania. Za każdą zmianą w kodzie uruchamiane są automatyczne testy, a aplikacja jest budowana. Pozwala to na szybszą identyfikację błędów, problemów z bezpieczeństwem.

- Często integracja zmian zmniejsza ryzyko konfliktów w kodzie, pomimo że nad aplikacją może pracować wiele osób.
- Szybciej znalezione błędy są łatwiejsze do poprawy, gdy mamy do czynienia z małą zmianą zrobioną niedawno.

Ciągła integracja (cd.)

- Standardową praktyką jest połączenie systemu kontroli wersji z uruchomieniem testów automatycznych.
- Po każdej wrzuconej zmianie automatycznie mogą zostać uruchomione narzędzia do statycznej analizy kodu, oceny jakości kodu.
- Następnie aplikacja zostaje zbudowana, a dalej wykonane testy jednostkowe i integracyjne.
- Automatyzacja tego procesu pozwala zaoszczędzić czas programistów, zwiększyć jakość kodu i zmniejszyć ilość błędów.

Ciągła integracja – przykład



Ciągłe dostarczanie

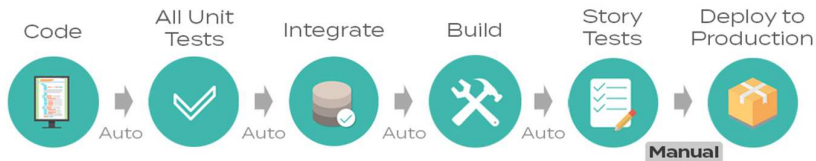
Definicja

Ciągłe dostarczanie CD (ang. *Continuous Delivery*), praktyka w której po poprawnej walidacji oprogramowania za pomocą testów automatycznych, następuje wypuszczenie nowej wersji programu. Dzieje się to automatycznie przy każdej zmianie w programie.

- Dzięki temu, że program jest obłożony testami, mamy większą pewność że aktualizacja pójdzie zgodnie z planem.
- Automatyzacja procesu pozwala na szybsze dostarczenie zbudowanej aplikacji na środowisko testowe albo produkcyjne.

Ciągłe dostarczanie – przykład

Continuous Delivery



Po co stosujemy CI/CD?

- Cel tych technik jest taki sam, a jest nim uproszczenie procesu tworzenia oprogramowania i wypuszczania aktualizacji.
- Różnica między nimi to jakiej automatyzacji używamy – czy automatyzacji budowania aplikacji i testów, czy automatyzacji wypuszczania aktualizacji.
- Narzędzia CI/CD:
 - systemy kontroli wersji,
 - serwery automatyzacji,
 - narzędzia do analizy kodu,
 - narzędzia do wirtualizacji i zarządzania środowiskami.

Narzędzia kontroli wersji

System kontroli wersji czyli oprogramowanie służące do śledzenia zmian głównie w kodzie źródłowym oraz pomocy programistom w łączeniu zmian dokonanych w plikach przez wiele osób w różnym czasie.

Przykłady:

- Git,
- Mercurial,
- Subversion.

Narzędzia kontroli wersji – przykład

The screenshot shows the GitHub interface for the 'torvalds/linux' repository. At the top, there's a search bar and navigation links for Pull requests, Issues, Codespaces, Marketplace, and Explore. Below that, the repository name 'torvalds/linux' is displayed with a 'Public' label and a 'Watch 8.1k' button. The main navigation bar includes 'Code', 'Pull requests (312)', 'Actions', 'Projects', 'Security', and 'Insights'. The repository details show the current branch as 'master', 1 branch, and 775 tags. There are buttons for 'Go to file', 'Add file', and 'Code'. The repository description is 'Linux kernel source tree'. On the right, statistics show 145k stars, 8.1k watching, and 46.3k forks. A 'Releases' section shows 775 tags. The main content area lists files and folders with their commit history:

File/Folder	Commit Message	Time Ago
Documentation	Merge tag 'for-linus' of git://git.kernel.org/pub/scm/virt/kvm/kvm	2 days ago
LICENSES	LICENSES: Add the copyleft-next-0.3.1 license	2 months ago
arch	Merge tag 'x86_urgent_for_v6.2_rc4' of git://git.kernel.org/pub/scm/...	8 hours ago
block	Merge tag 'block-6.2-2023-01-13' of git://git.kernel.dk/linux	2 days ago
certs	certs: make system keyring depend on built-in x509 parser	3 months ago
crypto	Merge tag 'v6.2-p1' of git://git.kernel.org/pub/scm/linux/kernel/git/...	last month
drivers	Merge tag 'edac_urgent_for_v6.2_rc4' of git://git.kernel.org/pub/scm/...	8 hours ago

Serwery automatyzacji

Serwer automatyzacji czyli oprogramowanie służące do automatyzacji związanej z powstawaniem oprogramowania. Pozwala zdefiniować zestaw instrukcji tzw. pipeline, w którym można określić dokładnie jak aplikacja ma zostać zbudowana, testowana i wdrażana na środowisko. Najczęściej łączy się to z systemem kontroli wersji i automatycznie uruchamia potrzebne kroki po dodaniu zmian do kodu.

Serwer automatyzacji – przykład

The screenshot shows the GitHub Actions interface for a pipeline named 'env-var-test'. The pipeline is currently running, and the dashboard displays a list of pipeline runs with their status, workflow, branch, and start time. The pipeline is configured to run on the 'context-env-var' branch.

Dashboard **Project** **Branch**
 All Pipelines > env-var-test > context-env-var

env-var-test [Add team members](#)

[Edit Config](#) [Trigger Pipeline](#) [Project Settings](#)

Filters: Everyone's Pipelines env-var-test context-env-var Auto-expand

Pipeline	Status	Workflow	Branch / Commit	Start	Duration	Actions
env-var-test 85	Running	test-env-var-2	context-env-var 94af749	3s ago	5s	🔄 🔄 🗑️ ⋮
Jobs	Running	test-env-var-1	context-env-var 94af749	3s ago	6s	🔄 🔄 🗑️ ⋮
Jobs	Running	env-var-1 137			4s	
env-var-test 85	Success	test-env-var-2	context-env-var 94af749	26s ago	6s	🔄 🔄 🗑️ ⋮
Jobs	Success	env-var-2 136			3s	
Jobs	Success	test-env-var-1	context-env-var 94af749	26s ago	5s	🔄 🔄 🗑️ ⋮
Jobs	Success	env-var-1 135			3s	
env-var-test 84	Success	test-env-var-2	context-env-var 94af749	53s ago	8s	🔄 🔄 🗑️ ⋮
Jobs	Success	env-var-2 134			4s	
Jobs	Success	test-env-var-1	context-env-var 94af749	53s ago	6s	🔄 🔄 🗑️ ⋮

Can't find an organization?
[Check permissions](#) and update access to the ones you want.

Notifications Status **OPERATIONAL** Docs Orbs Support

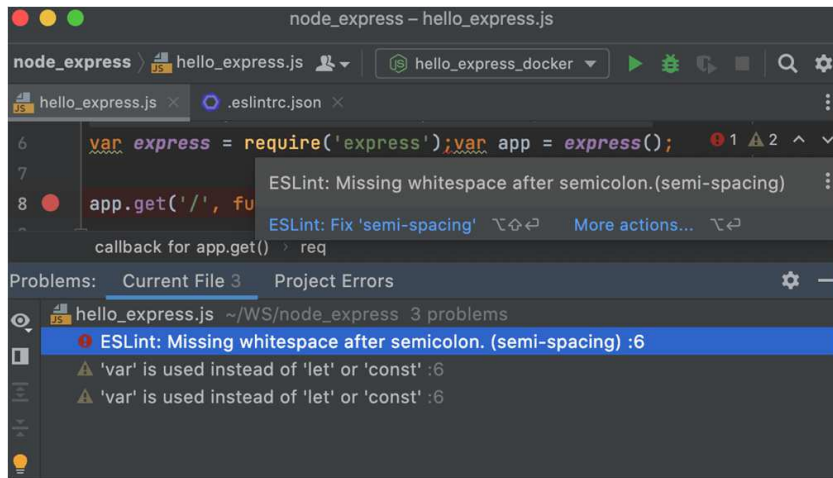
Narzędzia analizy kodu

Narzędzia analizy kodu pozwalają na ciągłe sprawdzanie jakości kodu, wykonują statyczną analizę w kodzie w poszukiwaniu błędów. Szukają duplikatów, sprawdzają zgodność z zadanymi standardami. Mogą zostać automatycznie uruchomione przez serwery automatyzacji, przez co każde zmiany w kodzie mogą zostać automatycznie sprawdzone.

Przykłady:

- ESLint,
- SonarQube.

Narzędzia analizy kodu – przykład



Narzędzia testowania oprogramowania

Narzędzia do testowania oprogramowania pozwalają na automatyzację procesów testowania oraz umożliwiają dokładniejsze i skuteczniejsze testowanie oprogramowania. Narzędzia dzielą się zależnie od rodzaju testów, które chcemy przeprowadzić.

Przykłady:

- Selenium,
- Cypress,
- Jest,
- JUnit.

Jednym z popularniejszych narzędzi do testowania automatycznego jest Selenium, które pozwala na testowanie aplikacji webowych za pomocą interfejsu użytkownika.

Narzędzia testowania oprogramowania – przykład

The screenshot displays the Cypress test runner interface. On the left, the test results for `LoginForm.cy.tsx` are shown, indicating a successful run with 10 passing tests. The test suite includes several assertions and a form test. The code snippet for the form test is as follows:

```
1 mount <LoginForm ... />
2 contains Username
3 -find input
4 contains Password
```

On the right, a preview of the login form is shown, rendered in Chrome 101 at 500x500 (81%) resolution. The form contains the following elements:

- Log In** (Section Header)
- Username:** (Label) followed by an input field.
- Password:** (Label) followed by an input field.
- Login** (Button)

Narzędzia wirtualizacji i zarządzania środowiskami

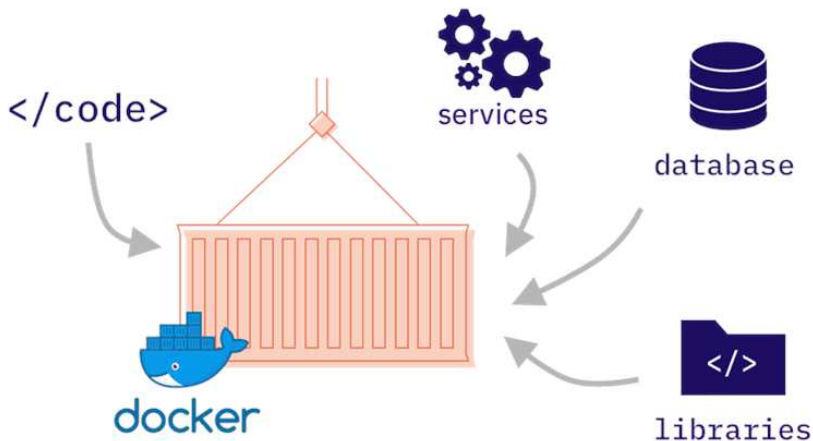
Narzędzia do wirtualizacji pozwalają na emulowanie różnych systemów operacyjnych i sprzętowych na jednej fizycznej maszynie, co pozwala na uruchamianie różnych aplikacji i środowisk bez konieczności posiadania kilku fizycznych maszyn.

Przykłady:

- Docker,
- Vagrant.

Popularnym rozwiązaniem stosowanym w CI/CD jest Docker, w którym budujemy środowiska oparte o kontenery, czyli małe samodzielne jednostki oprogramowania, które mogą być uruchamiane w dowolnym systemie operacyjnym.

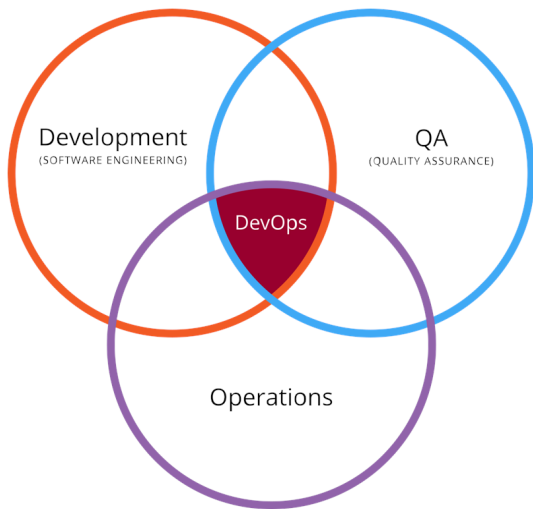
Narzędzia wirtualizacji – przykład



Czym jest DevOps?

DevOps jest to zbiór podejść, dobrych praktyk oraz narzędzi, które usprawniają proces tworzenia oprogramowania. Osoba zajmująca się DevOps, jest to ktoś kto szuka, używa i konfiguruje powyższe elementy.

DevOps – ilustracja



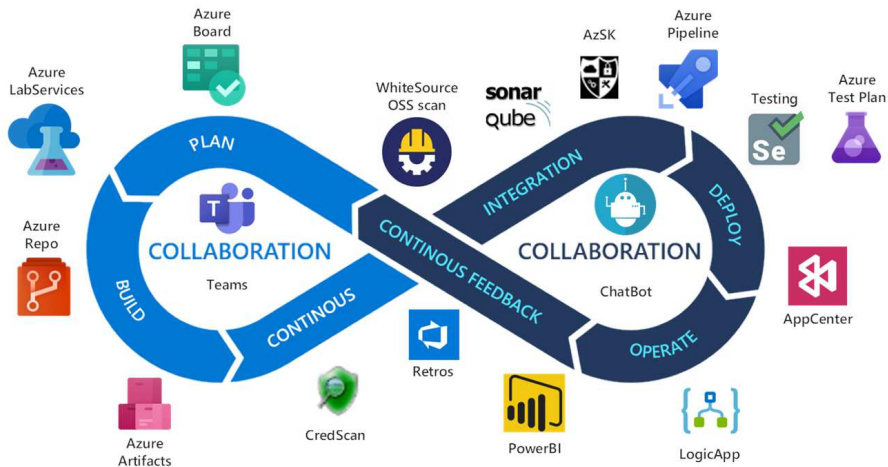
Podstawy DevOps

- Continuous Integration (CI) – polega na ciągłym integrowaniu i testowaniu kodu, co pozwala na szybkie wykrywanie i rozwiązywanie błędów.
- Continuous Delivery (CD) – polega na ciągłym wprowadzaniu nowych funkcjonalności do produktu, co pozwala na szybsze dostarczanie nowych funkcjonalności do klientów.
- Infrastructure as Code (IaC) – polega na opisanu infrastruktury (np. serwerów, baz danych) w formie kodu, co pozwala na automatyzację procesów zarządzania infrastrukturą.
- Monitoring and Logging – polega na ciągłym monitorowaniu i rejestrowaniu działania oprogramowania oraz infrastruktury, co pozwala na szybkie wykrywanie i rozwiązywanie problemów.

Dlaczego stosujemy DevOps?

DevOps jest coraz bardziej popularny wśród firm, ponieważ pozwala na szybsze wprowadzanie nowych funkcjonalności i usprawnień do oprogramowania, co przekłada się na lepszą jakość produktów i zwiększenie zadowolenia klientów.

DevOps – ilustracja




Microsoft Azure