


Wzorce Wdrożeniowe

Zespół: M.B, J.M, A.R, T.S, H.N



Wzorce Wdrożeniowe

Czyli rodzice CI/CD

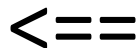


Czym są wzorce?

- Wzorce wdrożeniowe lub strategie wdrożeniowe to zbiór rozwiązań dla typowych problemów związanych z wdrażaniem.
- Sprawiają, że proces wdrożenia jest bezpieczniejszy
- Zmniejszają ryzyko wystąpienia błędów w nowych wersjach
- Przykłady wzorców wdrożeniowych:
 - Canary deployment
 - Blue/green deployment
 - A/B testing

Po co te całe wzorce?

- Co jeśli w wewnętrznym środowisku testowym wszystko działało a na produkcji nowa wersja zaczyna się sypać?
- Co jeżeli nowa wersja nie przypadła klientom do gustu i trzeba robić rollback aplikacji na wszystkich serwerach produkcyjnych?



Stosowanie dobrych wzorców wdrożeniowych pozwala w większości uniknąć tych przypadków.



Canary Server

O jaki ładny kanarek!

Segmentation fault (core dumped)

O nie...

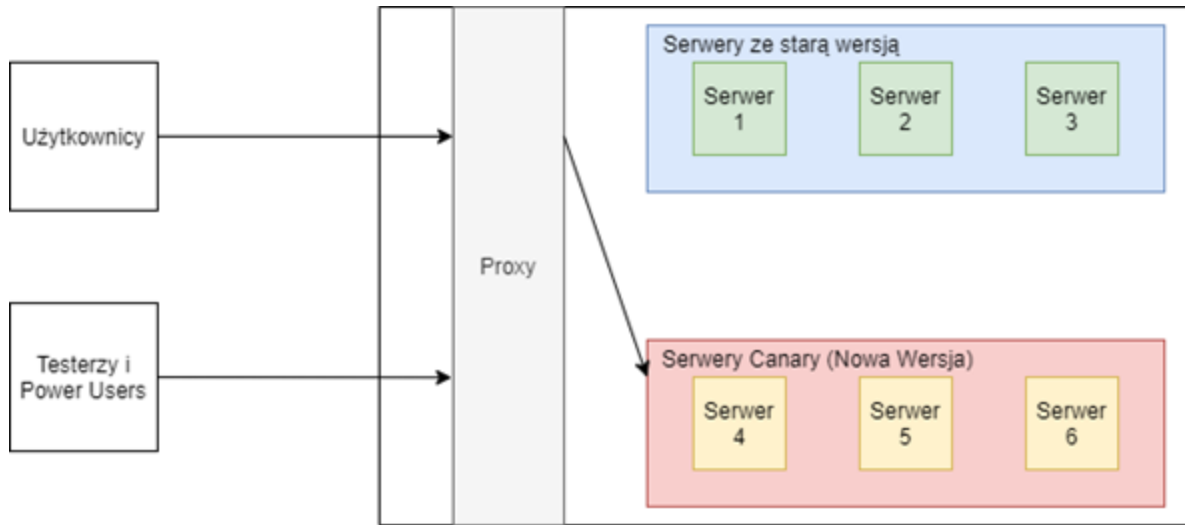
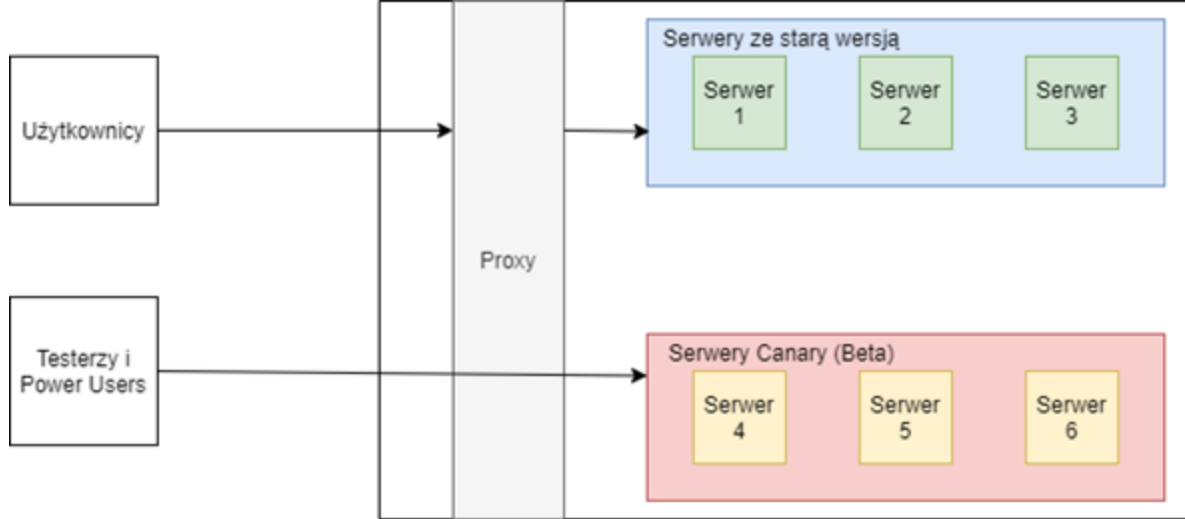


- Nowa wersja aplikacji zostaje uruchomiona na pewnym procencie serwerów produkcyjnych
- Użytkownicy są przekierowywani na serwery ze starą wersją aplikacji
- Nowa wersja jest testowana przez automatyczne testy, testerów i chętnych użytkowników. Aplikacja jest obserwowana pod kątem wydajności, niezawodności i współpracy z istniejącymi systemami
- Nowa wersja jest propagowana na pozostałe serwery lub robiony jest rollback serwerów canary w zależności od wyników testów


Wdrożenie za pomocą serwerów canary pozwala na przetestowanie nowej wersji aplikacji w środowisku produkcyjnym bez zaburzania dostępu do starej wersji.

Testerzy i chętni użytkownicy mogą korzystać serwerów canary (beta) do oceny nowej wersji.

Jeśli nowa wersja się sprawdzi ruch jest zamieniany i wszyscy użytkownicy są przekierowywani na serwery canary aż reszta serwerów zostanie zaktualizowana



Deploy
nowej
wersji



Blue/Green Deployment

Boże/Gdzie idą moje pakiety?... a... tu są

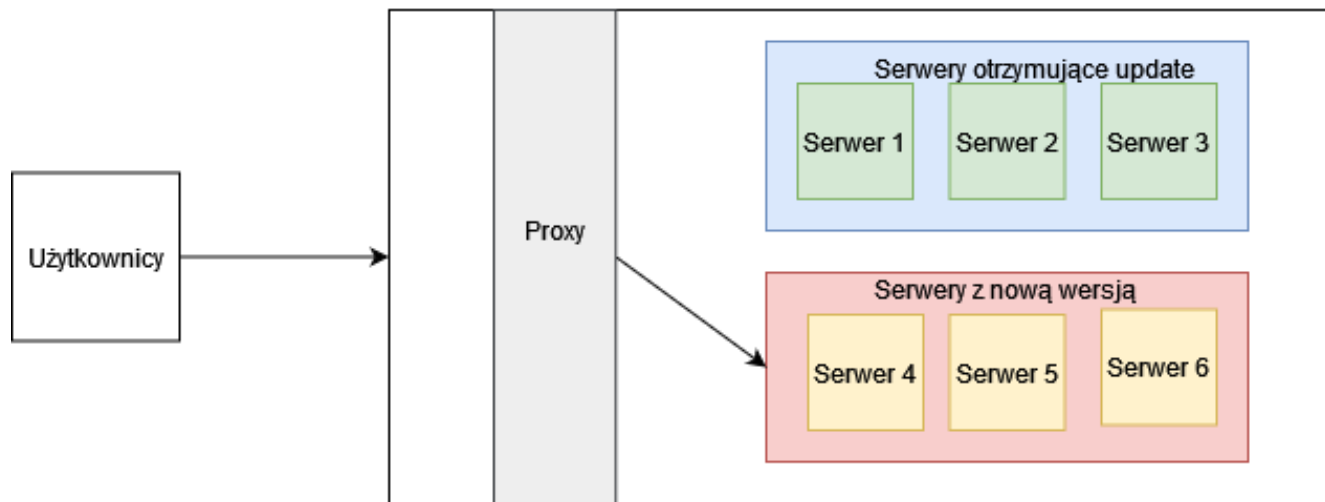
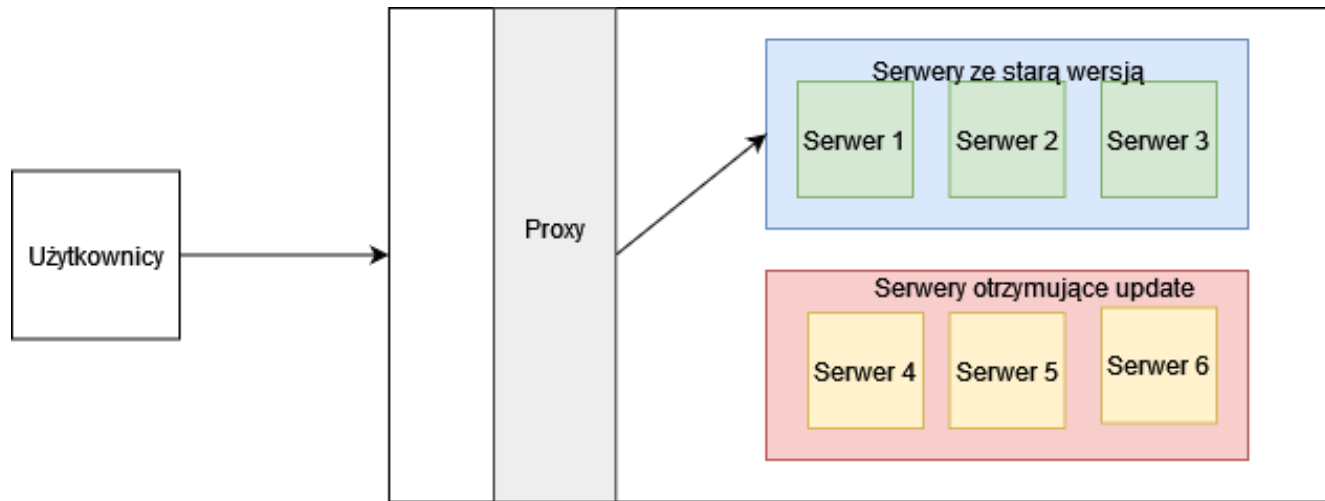


Celem wdrożenia Blue/Green jest minimalizacja czasu gdzie aplikacja jest niedostępna

Podczas takiego deployu nie przeprowadzane są żadne testy ale pozwala to na szybkie wypchnięcie nowej wersji na wszystkie serwery produkcyjne bez zauważalnych luk w dostępności serwisu

Popularną praktyką jest podzielenie środowisk produkcyjnych na dwie grupy, gdzie jedna z nich otrzyma deploy jako pierwsza.

- Część serwerów wybierana jest do otrzymania nowej wersji
- Ruch użytkowników przekierowywany jest na serwery ze starą wersją
- Kiedy deploy nowej wersji na wybrane serwery jest ukończony, ruch użytkowników przekierowywany jest na serwery z nową wersją
- Pozostałe serwery ze starą wersją otrzymują update





A/B Testing

Ale będzie krzyk o ten nowy layout

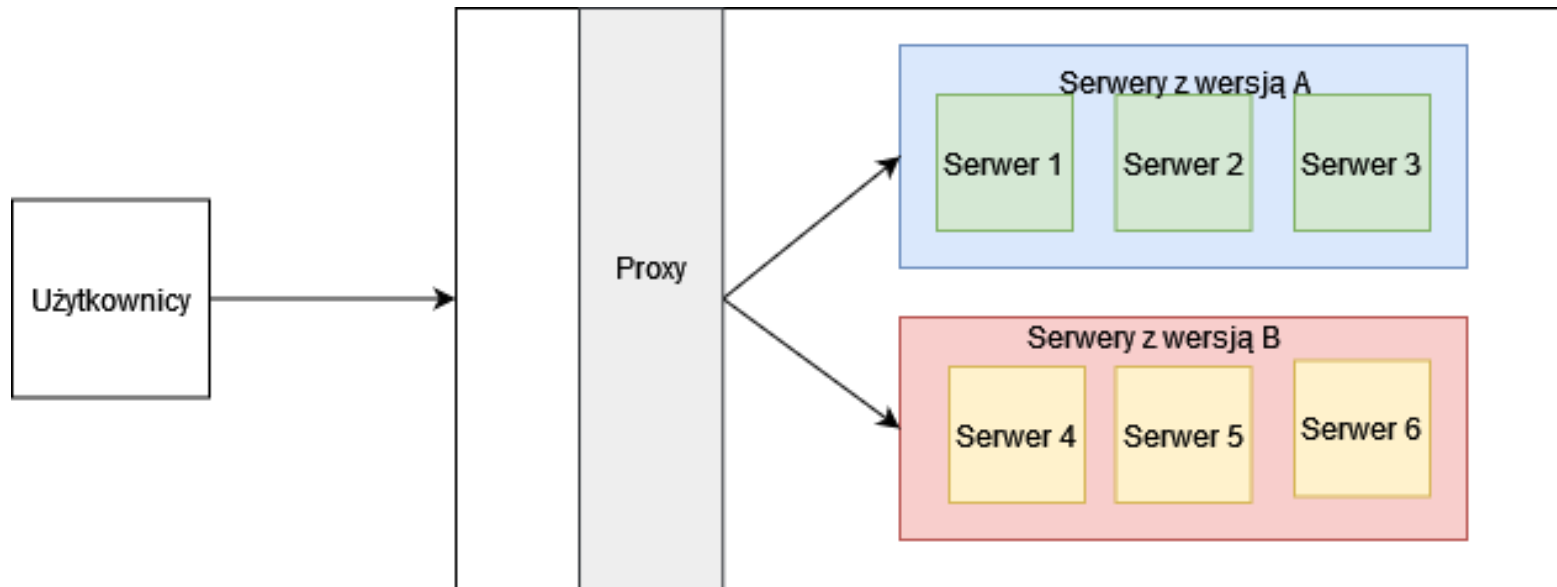



- Serwery produkcyjne są dzielone na dwa segmenty: A, które będą zawierały starą wersję aplikacji i B, które będą miały nową.
- Wybrana jest pewna grupa użytkowników, którzy otrzymają dostęp do wersji B.
- Zbierane są opinie użytkowników o wersji B.
- Jeśli opinie są pozytywne reszta serwerów produkcyjnych otrzymuje update podczas którego wszyscy użytkownicy są kierowani na serwery B.

Testy A/B służą zwykle do sprawdzenia opinii użytkowników o zmianach w aplikacji. Zwykle chodzi o zmiany dotyczące frontendu.

W opisanym modelu A/B, A jest starą wersją aplikacji a B nową.

Inne podejście do testów A/B to stworzenie dwóch nowych wersji aplikacji o różnym wyglądzie i sprawdzenie która wersja zbiera więcej pozytywnych i deploy takiej wersji na resztę produkcji lub integracje jej najlepiej odbieranych elementów do następnej wersji





Continuous deployment

Testerzy mile widziani... gdzie jest aktualna dokumentacja?



- Jest wzorcem wdrażania, który tworzy rurociąg, w którym każdy krok jest wykonywany przez zautomatyzowany proces
- Każdy krok jest wykonany bez ludzkiej interwencji
- Jeśli w rurociągu pojawi się nowa wersja, wszystkie kroki zostaną wykonane automatycznie chyba że pojawi się błąd

Proces ciągłego wdrażania powstał jako ewolucja ciągłego dostarczania które z kolei jest ewolucją ciągłej integracji. Powstał on z potrzeby szybszego procesu wydawania wersji, gdzie zautomatyzowanie procesów stało się rozwiązaniem na skrócenie czasu wydawania wersji aplikacji.

Ciągłe wdrażanie nie jest ciągłym dostarczaniem lub ciągłą integracją - te wzorce wdrażania mają wiele podobieństw, jednak wszystkie są różne.

Różnią się one poziomem zautomatyzowania.

Wykonywać
Automatycznie

Wykonywać
Ręcznie

Integracja ciągła



Dostarczanie ciągłe



Wdrażanie ciągłe



Głównym celem ciągłego wdrażania jest zminimalizowanie czasu pomiędzy opracowaniem wersji a jego dostarczeniem na produkcję.

Automatyzuje on kroki dewelopera, tak aby zminimalizować zakres błędu ludzkiego i uczynić dostawę bezpieczniejszą.

Wdrażanie ciągłe może zostać używane razem z blue/green deployment, canary deployment, and A/B testing

Do implementacji wdrażania ciągłego, konieczne jest posiadanie dobrej kultury testowania.

Jest tak ponieważ jakość testów będzie definiować jakość wersji, jak również sukces jego implementacji.

Ponadto, dokumentacja musi aktualizować wszelkie nowe wersje viceversa wszelkie aktualizacje w aplikacji muszą zostać odzwierciedlone w dokumentacji.