

DevOps i automatyzacja

Odpowiedzialność za usługę, feedback i małe zmiany

mgr inż. Jakub Woźniak

Zarządzanie Systemami Rozproszonymi · Semestr zimowy 2026/27

Forma

- semestr zimowy 2026/27
- 15 spotkań × maks. 45 minut
- wykład i laboratorium

Oś kursu

- zmiana
- wdrożenie
- obserwacja
- reakcja na awarię

Od kontenera i klastra, przez automatyzację oraz obserwowalność, do odporności i odtworzenia usługi.

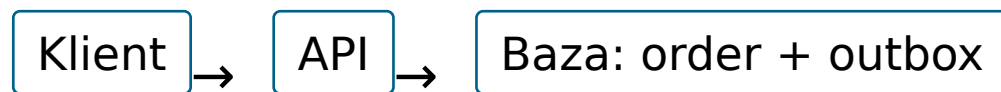
mgr inż. Jakub Woźniak

- Informatyka · specjalność: systemy rozproszone
- Politechnika Poznańska · Wydział Informatyki · 2017
- jakub.wozniak@cs.put.poznan.pl

Praca zawodowa

CTO for Software Services · IBM Consulting

IBM Consulting nie jest organizatorem ani partnerem przedmiotu.



- Zamówienie i outbox: jedna transakcja; 202 po trwałym zapisie.
- Publisher publikuje zapisane zdarzenia; możliwe powtórzenia.
- Worker pobiera wiadomość, zapisuje skutek, potem potwierdza odbiór.

C. Richardson — Transactional outbox.

- Awaria po zmianie konfiguracji
- DevOps jako sposób organizacji odpowiedzialności
- CALMS, małe zmiany i pętla feedbacku
- 🏔 Usługa zarządzana a odpowiedzialność zespołu
- Retry zamówienia i idempotencja
- Decyzja operacyjna

Sygnał

Po wdrożeniu nowej konfiguracji worker nie łączy się z kolejką. API nadal zapisuje zamówienia w bazie, lecz zlecenia nie są realizowane.

Co wiemy

- rośnie liczba wiadomości oczekujących;
- klient może nadal dostać potwierdzenie zamówienia;
- zmiana objęła konfigurację workera.

Czego jeszcze nie wiemy

- czy problem dotyczy wersji workera, uprawnień czy kolejki;
- czy poprzednia wersja jest zgodna z zaległymi wiadomościami;
- czy można bezpiecznie kontynuować wdrożenie.

Przekazanie

Zespół przekazuje artefakt do wdrożenia. Informacja o skutkach zmiany wraca późno i do innego zespołu.

Współodpowiedzialność

Osoby przygotowujące zmianę współpracują przy jej wdrożeniu, obserwacji i odzyskaniu działania usługi.

J. Humble, D. Farley, Continuous Delivery, 2010, rozdz. 1.

Obszar	Pytanie dla systemu zamówień
Culture	Kto razem podejmuje decyzję o zatrzymaniu wdrożenia?
Automation	Które kroki odtworzymy bez ręcznej konfiguracji?
Lean	Jak zmniejszyć zakres następnej zmiany?
Measurement	Po jakich sygnałach poznamy, że zamówienia są realizowane?
Sharing	Gdzie zapisujemy wnioski i procedurę po incydencie?

G. Kim i in., The DevOps Handbook, 2. wyd., 2021; CALMS.

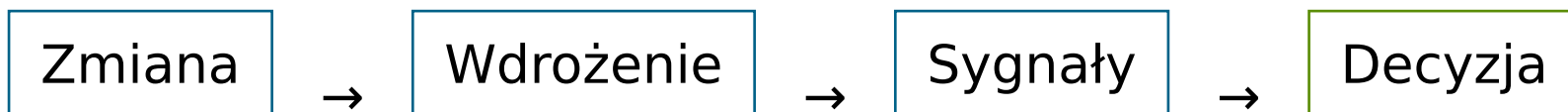
Zmiana ryzykowna

Nowy worker, nowa konfiguracja kolejki i migracja schematu bazy trafiają do produkcji w jednym wdrożeniu.

Zmiana kontrolowana

Najpierw wdrażamy workera zgodnego z poprzednim formatem wiadomości, obserwujemy przepływ, a kolejne zmiany wprowadzamy osobno.

Efekt: prostsza diagnoza, mniejszy zakres rollbacku i krótszy czas między hipotezą a wynikiem.



← Wynik decyzji określa kolejną zmianę i pomiar ←

Hipoteza

„Nowy worker odbiera i przetwarza zamówienia tak jak poprzedni.”

Decyzja

Kontynuować rollout, zatrzymać go albo wykonać kontrolowany rollback.

Perspektywa	Sygnał	Interpretacja
Klient	odsetek udanych zamówień, czas odpowiedzi	czy usługa realizuje cel użytkownika
Przepływ	liczba i wiek wiadomości w kolejce	czy worker nadąża z przetwarzaniem
Komponent	błędy połączenia workera z kolejką	gdzie rozpocząć diagnozę

Pułapka

Działający proces workera nie jest dowodem, że zamówienia są realizowane.

B. Beyer i in., Site Reliability Engineering, 2016, rozdz. 6.

Dostawca usługi

- infrastruktura usługi kolejki;
- dostępność komponentu zgodnie z umową;
- mechanizmy szyfrowania i aktualizacje platformy.

Zespół aplikacji

- tożsamości i uprawnienia workera;
- konfiguracja, alerty i procedura reakcji;
- zgodność formatu wiadomości oraz poprawność przetwarzania.

Microsoft Learn, Shared responsibility in the cloud, 2026.



Po timeoutcie klient może ponowić żądanie. Bez dodatkowego mechanizmu API może utworzyć drugie zamówienie i opublikować drugą wiadomość.

Klucz idempotencji

Klient przekazuje identyfikator operacji. API atomowo zapisuje klucz, treść operacji i wynik wraz z zamówieniem. Ponowienie tej samej operacji zwraca zapisany rezultat; ten sam klucz z inną treścią jest odrzucany.

RFC 9110, sekcja 9.2.2; właściwości metod HTTP nie rozwiązują automatycznie semantyki zamówienia.

Po zmianie workera obserwujesz przez pięć minut (dane dydaktyczne):

- napływ do kolejki: bez zmiany;
- błędy API: 0,2% → 0,3%;
- liczba wiadomości w kolejce: 200 → 4 800;
- wiek najstarszej wiadomości: 10 s → 4 min.

Pytanie

Czy kontynuujesz rollout, zatrzymujesz go czy robisz rollback? Jakiego dwa sprawdzenia wykonasz przed następną decyzją?

Uzasadnienie

Rosnąca kolejka i wiek najstarszej wiadomości wskazują, że przepływ zamówień nie nadąża. To przesłanka wstrzymania promocji, ale jeszcze nie dowód błędu nowej wersji; sprawdzenia wymagają również zależności.

Przed rollbackiem sprawdź:

- błędy uwierzytelnienia i połączeń oraz różnicę konfiguracji między wersjami;
- zgodność poprzedniego workera z zaległymi wiadomościami i tempo ich opróżniania.

Rollback wymaga zgodności poprzedniej wersji z danymi i potwierdzenia, że może ograniczyć wpływ. Samo wstrzymanie promocji nie przywraca przetwarzania.

- DevOps łączy przygotowanie zmiany z odpowiedzialnością za jej skutki.
- Mała zmiana staje się bezpieczna dopiero wraz z sygnałami, które pozwalają ją ocenić.
- Usługa zarządzana zmienia podział obowiązków, lecz nie zastępuje odpowiedzialności za aplikację.

Pytanie na koniec

Jaką metrykę dodałbyś, aby odróżnić wolne przetwarzanie od całkowitego zatrzymania workera?

- C. Richardson, *Transactional outbox*: microservices.io.
- J. Humble, D. Farley, *Continuous Delivery*, Addison-Wesley, 2010, rozdz. 1.
- G. Kim i in., *The DevOps Handbook*, 2. wyd., IT Revolution, 2021, rozdz. 1.
- B. Beyer i in., *Site Reliability Engineering*, O'Reilly, 2016, rozdz. 6: sre.google.
- Microsoft Learn, *Shared responsibility in the cloud*: learn.microsoft.com.
- R. Fielding i in., RFC 9110, sekcja 9.2.2: rfc-editor.org.