



Inteligentna, skalowalna aplikacja dzięki platformie Google Cloud

Mariusz Krzemień

Modernizacja aplikacji do modelu mikroserwisów z monitoringiem wydajności, skalowaniem i automatycznym reagowaniem na awarie z wykorzystaniem predykcji na przykładzie migracji aplikacji Java do chmury

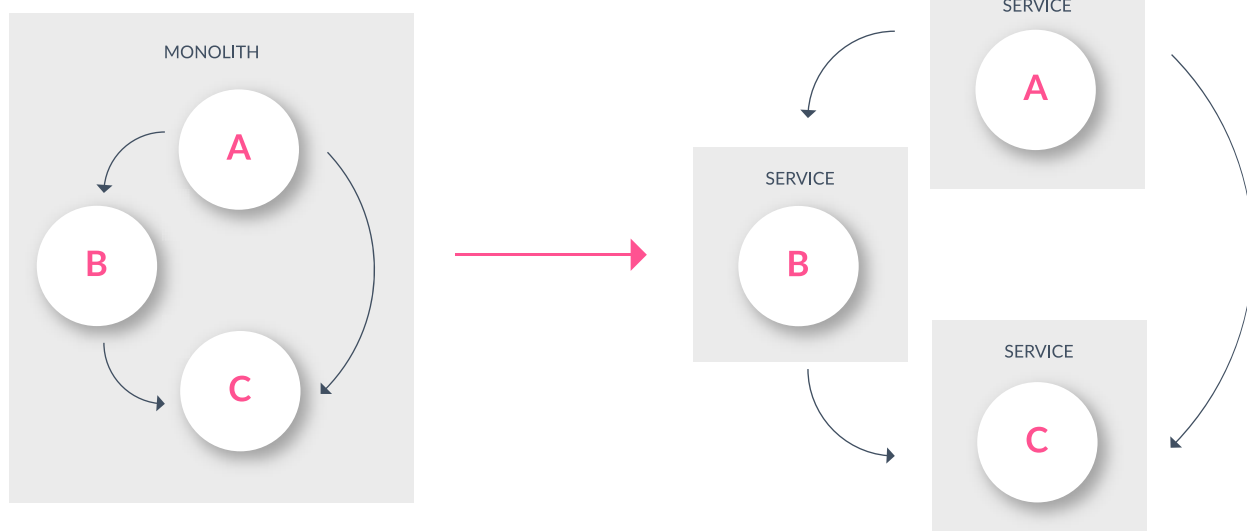
Model mikroserwisów

O mikroserwisach słyszał już chyba każdy. Mikroserwisy są obecnie jednym z wiodących i najchętniej wykorzystywanych modeli architektonicznych przy projektowaniu aplikacji. Praktycznie w każdym projekcie, w którym w grę wchodzi migracja systemów on-premises do chmury, mikroserwisy są od początku jednym z głównych docelowych rozwiązań, branych pod uwagę w analizie wariantowej i kosztowej.

I bardzo często to rozwiązanie wygrywa – jeżeli nie od razu, kiedy na początku decydujemy się na model lift-and-shift, to z pewnością w niedalekiej perspektywie czasowej po migracji.

Dlatego warto tutaj wymienić te najważniejsze spośród zalet tego podejścia, wynikające z samej istoty mikroserwisów, które otrzymujemy de facto z automatu. Są to:

- ✓ szybkość działania,
- ✓ wysoka wydajność,
- ✓ brak zależności od dostawcy usług platformy systemowej,
- ✓ odporność na awarie,
- ✓ zwiększona łatwość modernizacji, a tym samym prostsze zmniejszanie długu technologicznego w przyszłości.



Rys. 5 Migracja monolitycznych systemów on-premises do elastycznych modeli mikroserwisów

Czy trzeba specjalnie przygotować aplikację?

Co jest najważniejsze podczas procesu podejmowania jakiegokolwiek decyzji? Zdecydowanie dla większości z nas jest to możliwość wyboru spośród wielu dostępnych rozwiązań. Im szersze jest dostępne portfolio, tym oczywiście lepiej. W ten sposób możemy optymalnie odpowiedzieć na potrzeby i zrealizować postawione cele biznesowe. Oczywiście wybór docelowego rozwiązania nigdy nie jest łatwy i w większości przypadków zależy od bardzo wielu czynników. W przypadku, kiedy nasza aplikacja nie korzysta z modelu mikroserwisów, wskazane jest dokonanie analizy, która odpowie nam na pytanie, czy warto zachować istniejącą architekturę i zdecydować się na model lift-and-shift, czy może zamienić ją na bardziej nowoczesne podejście. Każdy przypadek należy rozpatrzyć i przeanalizować indywidualnie na etapie przygotowania projektu migracyjnego. Jeśli posiadamy już aplikację napisaną w języku Java, jednej z najpopularniejszych obecnie platform, migracja jest znacznie prostsza.

Czy trzeba specjalnie przygotować taką aplikację? Odpowiadając w najprostszym sposobie: poza standardowymi czynnościami, takimi jak konteneryzacja oraz dostosowaniem do ogólnych wymogów stawianych przez chmurę związanych przede wszystkim z zarządzaniem i monitorowaniem aplikacji – nie trzeba. Aplikacja może pozostać w architekturze monolitu lub może być zmodernizowana do architektury mikroserwisów.

Jeżeli zdecydujemy się na modyfikację aplikacji, platforma taka jak Google Cloud zaoferuje nam środowisko składające się z wielu dodatkowych natywnych mechanizmów wspierających to podejście oraz rozszerzony wybór rodzajów platform dostępnych dla tego języka. Wybór jest w uproszczeniu zależny od tego, czy i w jakim stopniu chcemy minimalizować wysiłek potrzebny na utrzymanie oraz jaki stopień kontroli i zarządzania chcemy zachować.

Mechanizmy automatyzujące utrzymanie i zarządzanie



Jedną z głównych zalet migracji do chmury jest szeroka oferta natywnych narzędzi w obszarze zarządzania i utrzymania aplikacji, takich jak mechanizmy monitorowania, śledzenia błędów lub alterowania. Mechanizmy te zapewniają ciągłe, bazujące na zdarzeniach

identyfikowanych w czasie rzeczywistym, śledzenie stanu i kondycji aplikacji oraz wizualizacje tego stanu przy pomocy dashboardów, tworzenie metryk oraz alertów opartych na zdarzeniach. Wszystkie te mechanizmy ułatwiają zarządzanie i umożliwiają błyskawiczne reagowanie na problemy przez zespół utrzymaniowy. Źródłem danych mogą być logi lub parametry bieżące stanu kondycji aplikacji. Dane te mogą obejmować zarówno parametry systemowe, jak i być tak skonfigurowane, aby monitorować poprawność przetwarzania przypadków użycia z perspektywy biznesowej.

A czy można przewidzieć i zareagować na awarię zanim ona nastąpi?

Migrując aplikację do chmury, często zastanawiamy się, jakie będą wymierne korzyści. Możemy je rozpatrywać nie tylko z perspektywy oszczędności – często jest to jeden z głównych czynników decydujących o opłacalności projektu migracyjnego, lecz także z perspektywy na pierwszy rzut oka nie tak oczywistej, jednak w długofalowej wizji bardzo istotnej i gwarantującej nie tylko zmniejszone koszty utrzymania, ale także zapewnienie ciągłości działania oraz redukcji nakładów technicznych i organizacyjnych na jej utrzymanie – perspektywy potencjalnej awarii. Przewidywać awarie możemy poprzez zastosowanie pewnego rodzaju sztucznej inteligencji, pozwalającej nam utrzymać aplikacje w sposób wysoce zautomatyzowany. Używając rozwiązań typu on-premises, nie możemy 24/7 monitorować i reagować na zwiększone zapotrzebowanie aplikacji na zasoby automatycznie i szybko, co powoduje przerwę w jej działaniu.

W przypadku chmury sytuacja znacząco się zmienia. Mechanizmy autoscalingu oraz predykcji – bo o nich mowa, stanowią jedną z głównych wartości dodanych platform chmurowych. Są to natywne mechanizmy gwarantujące zapewnienie ciągłości i odpowiedniej wydajności w przypadku zmiennych warunków, w jakich pracuje aplikacja. Mowa tu przede wszystkim o zmiennym obciążeniu i zmiennej liczbie transakcji spowodowanej zwiększonym ruchem klienckim. Platforma chmurowa jest w stanie samodzielnie dostosować zasoby podczas zmieniających się warunków, w których aplikacja działa i w czasie rzeczywistym dokonać automatycznej korekcji parametrów kontenerów – przydzielonych zasobów lub wydajności powiązanych usług. Mechanizmy te są również w stanie przewidzieć przyszłe zapotrzebowanie, tym samym z wyprzedzeniem reagować na potencjalne awarie, tj. zwiększać zasoby, redukując tym samym niemal do zera możliwy okres niedostępności bądź obniżonej

wydajności, w którym aplikacja dostosuje się do zmienionych warunków działania, w tym zwiększonego obciążenia ruchem klienckim. Przyczynkami do podjęcia decyzji dotyczącej zapewnienia zasobów w przyszłości są historia użycia aplikacji w przeszłości oraz obecne warunki. W ten sposób tworzone są modele zachowania. W przypadku braku mechanizmów predykcji, autoscaler może jedynie zwiększać zasoby bazując na obserwowanych zmianach w obciążeniu aplikacji w czasie rzeczywistym i na nie reagować. Mechanizmy, czy to autoscalingu bazującego na obecnym obciążeniu, czy wykorzystującego mechanizmy predykcji, są dostępne i automatycznie zarządzane odpowiednio przez wybranie i dopasowane do naszego przypadku środowiska chmurowego na poziomie instancji maszyn wirtualnych (MIG) lub kontenerów (Kubernetes).



PAULINA
CONSULTANT