

Martyna Filipiak,  
Krzysztof Hubert Kołodziejski

## **Zarządzanie i modernizacja platformy e-commerce w chmurze**

### **Wstęp i analiza problemów**

W ostatnim czasie zauważono wiele problemów związanych z działaniem naszej platformy e-commerce. Wielokrotnie dostawaliśmy też zgłoszenia użytkowników dotyczące szybkości odpowiedzi serwerów lub wyskakujących błędów 500. Błędy te zaczęły przekładać się na CTR, dzięki któremu wiemy, że mimo większej klikalności reklam nasze produkty sprzedają się coraz rzadziej. W wyniku analizy sytuacji na spotkaniach zidentyfikowano problem z wydajnością aplikacji. Z tego względu przeprowadzono analizę, aby lepiej nakreślić w czym leży problem i zaproponować sposób modernizacji architektury. Po przeprowadzeniu analizy stworzono plan modernizacji systemu, opisano technologie, które zostaną do tego użyte oraz nakreślono w jaki sposób ma przebiegać proces modernizacji, aby zminimalizować zakłócenia systemu.

W miarę jak liczba użytkowników platformy rosła, zaczęto zauważać znaczące pogorszenie jej wydajności. Strona internetowa stawała się coraz bardziej opóźniona, co objawiało się dłuższym czasem ładowania poszczególnych elementów i stron. Użytkownicy musieli czekać coraz dłużej na załadowanie się treści, co negatywnie wpływało na ich doświadczenie i satysfakcję z korzystania z platformy. Zwiększone obciążenie serwerów spowodowane rosnącą liczbą aktywnych użytkowników prowadziło do częstszego występowania zatorów w przetwarzaniu danych, co z kolei miało bezpośredni wpływ na wydajność systemu i czas realizacji zamówień, który uległ wydłużeniu. Proces, który wcześniej był szybki i sprawny, zaczął się przeciągać, co skutkowało opóźnieniami w dostarczaniu produktów czy usług.

Obecny design platformy nie jest zaprojektowany z myślą o elastycznym dostosowywaniu się do dynamicznie rosnących potrzeb użytkowników. Gdy liczba użytkowników wzrasta, a obciążenie systemu staje się większe, platforma zaczyna napotykać na poważne problemy z wydajnością. System staje się niestabilny, co objawia się częstymi awariami, spowolnieniem działania poszczególnych funkcji oraz dłuższym czasem ładowania się stron i realizacji zadań. Użytkownicy mogą doświadczać opóźnień w przetwarzaniu ich zapytań oraz zamówień, co bezpośrednio wpływa na ich doświadczenie i zadowolenie z korzystania z platformy.

Zauważono także problemy z niezawodnością, co przejawia się w postaci częstych awarii systemu. Te awarie prowadziły do sytuacji, w których platforma stawała się całkowicie niedostępna dla użytkowników. Problemy te miały miejsce zarówno w godzinach szczytu, gdy obciążenie było największe, jak i w losowych momentach, co dodatkowo komplikowało sytuację. Każda awaria oznaczała, że użytkownicy nie mogli korzystać z kluczowych funkcji platformy, takich jak przeglądanie ofert, dokonywanie zakupów czy korzystanie z usług wsparcia. Brak dostępu do platformy mógł prowadzić do poważnych konsekwencji, takich jak utrata możliwości dokonania zakupów, realizacji zamówień czy komunikacji z obsługą klienta. Awarie miały negatywny wpływ na reputację platformy. Każdy incydent niedostępności przyczyniał się do spadku zaufania użytkowników, którzy zaczęli wątpić w niezawodność systemu. Wymagały one od zespołu technicznego intensywnego zaangażowania w diagnozowanie i naprawianie problemów, zamiast koncentrować się na rozwijaniu nowych funkcji i ulepszaniu istniejących.

Ostatnie problemy z platformą podkreślają pilną potrzebę wprowadzenia fundamentalnych zmian w jej infrastrukturze. Konieczne jest zainwestowanie w solidne rozwiązania technologiczne, które zapewnią stabilność systemu i zminimalizują ryzyko przyszłych awarii.

## **Modernizacja architektury**

W odpowiedzi na zidentyfikowane problemy stworzono plan zmian architektury. Modernizacja obejmie wdrożenie platform i narzędzi takich jak:

- Compute Engine,
- Kubernetes Engine (GKE),
- GKE Autopilot (dodatek do Kubernetes Engine),
- Cloud SQL,
- Cloud Functions,
- Cloud Monitoring i Logging.

Powyższe narzędzia pozwolą nie tylko na poprawę wydajności, skalowalności i niezawodności, ale także wdrożenie ich nie będzie zauważalne dla użytkownika, co pozwoli na minimalizację zakłóceń.

Najważniejszym elementem modernizacji będzie wdrożenie Compute Engine. Usługa ta jest dostępna jako część platformy Google Cloud Platform. Ponieważ już korzystamy z tego rozwiązania, to Compute Engine będzie prostym do uruchomienia. Jego główną cechą, która przemawia za zastosowaniem tego rozwiązania to przede wszystkim nieprzerwane serwowanie naszej platformy. Oznacza to, że nawet podczas migracji nasza platforma będzie ciągle dostępna. Ponadto, pozwala na zmniejszanie i zwiększanie dostępnych

zasobów. W chwilach gdy analiza rynku wskazuje, że klienci wydają mniej pieniędzy i ruch sieciowy jest mniejszy, istnieje możliwość zmniejszenia wykorzystywanej pamięci. Z kolei gdy planujemy wyprzedaże lub widzimy zwiększone zainteresowanie naszym sklepem możemy zwiększać dostępne zasoby. Co ważne, Compute Engine pozwala wykorzystywać kontenery dzięki czemu będzie idealnym narzędziem do wykorzystywania wraz z kolejną platformą, którą planujemy wdrożyć, czyli Kubernetes Engine.

Kubernetes jest platformą do zarządzania kontenerami, która umożliwia planowanie, wdrażanie, skalowanie oraz zarządzanie cyklami życia aplikacji. Jest to narzędzie, które znacząco ułatwia tworzenie i zarządzanie wieloma instancjami naszej aplikacji, zapewniając jednocześnie równomierne rozłożenie obciążenia pomiędzy nimi. Dzięki wbudowanemu LoadBalancerowi Kubernetes automatycznie dystrybuje ruch użytkowników między instancjami, co gwarantuje stabilność i efektywność działania aplikacji. Tego typu podejście minimalizuje ryzyko przeciążeń, które mogłyby wystąpić, gdyby pojedyncza instancja aplikacji próbowała obsłużyć cały ruch użytkowników.

Aby uniknąć konieczności manualnego określania, ile pamięci przydzielić dla każdego klastra, wdrożony zostanie GKE Autopilot. Dzięki temu narzędziu automatycznie dostosuje się do potrzeb poszczególnych części naszego sklepu, optymalizując alokację pamięci.

Kolejną usługą wartą wdrożenia, która jest oferowana przez Google Cloud Platform jest Cloud SQL. Jest to usługa zarządzania relacyjnymi bazami danych. Korzystając z Cloud SQL, unikamy konieczności utrzymywania własnej infrastruktury bazodanowej, co pozwala nam skoncentrować się na rozwoju naszej platformy zamiast na obsłudze systemów. Jedną z głównych zalet Cloud SQL jest zapewnienie wysokiej dostępności i skalowalności naszej bazy danych. Dzięki temu możemy być pewni, że nasza baza danych jest zawsze dostępna i odporna na nagłe wzrosty ruchu. Dodatkowo, koszty korzystania z Cloud SQL są elastyczne i dostosowane do rzeczywistych potrzeb naszej aplikacji, co pozwala optymalizować wydatki na infrastrukturę. GCP umożliwia także automatyczne tworzenie kopii zapasowych bazy danych w wybranych przez nas oknach czasowych, co zapewnia bezpieczeństwo danych. Ostatnią ważną zaletą jest elastyczność w obsłudze zapytań do bazy danych. Cloud SQL automatycznie skaluje zasoby obliczeniowe w zależności od obciążenia, co eliminuje ryzyko przeciążenia serwera i zapewnia płynność działania aplikacji nawet w przypadku wzmożonego ruchu.

## **Plan zarządzania zmianami**

Wszystkie powyższe zmiany należy wdrożyć stosując plan procesu modernizacji, wprowadzając je krok po kroku, aby zminimalizować zakłócenia dla użytkowników.

Najważniejszym etapem będzie migracja bazy danych, ponieważ jest ona niezbędna do działania aplikacji. Ze względu na to, właśnie przeniesienie się na to rozwiązanie, będzie pierwszym etapem modernizacji architektury. Dodatkowo należy podkreślić, że nasza platforma już stoi na Google Cloud Platform, a Cloud SQL jest oferowane przez tę platformę, dlatego migracja będzie płynna. Dostosowanie aplikacji do korzystania z danych przechowywanych w chmurze sprowadza się głównie do zmiany źródła pobierania danych. Dodatkowo, kluczowe przeniesienie bazy danych zostało zaplanowane na godziny nocne, co pozwoli uniknąć przestojów i zapewni ciągłość działania systemu bez zakłóceń dla użytkowników.

Kolejnym krokiem będzie implementacja Compute Engine. Przeniesienie naszej platformy na ten system również powinno przebiegać płynnie, ponieważ jest to kolejne narzędzie dostępne w ramach Google Cloud Platform. Dodatkowo, Compute Engine umożliwia serwowanie strony nawet w trakcie migracji, co minimalizuje wpływ na negatywne doświadczenia użytkowników korzystających z platformy w tym czasie.

Kolejnym istotnym krokiem w naszym procesie będzie skalowanie naszej aplikacji przy użyciu Kubernetesa. Jest potężnym narzędziem do zarządzania kontenerami, które pozwala nam efektywnie zarządzać i skalować naszymi zasobami w chmurze. Planujemy także wdrożyć LoadBalancer, który będzie odpowiedzialny za równomierne rozdzielenie ruchu pomiędzy różnymi instancjami naszej platformy. Jest możliwe, że na początku przejścia na Kubernetesa użytkownicy mogą zauważyć lekki spadek wydajności, gdyż proces migracji i dostosowania może wymagać czasu na optymalizację i synchronizację działających klastrów. Jednakże po pełnej implementacji kolejnych klastrów oraz zoptymalizowaniu konfiguracji, oczekujemy znacznego wzrostu wydajności całej platformy. Dzięki skalowalności i elastyczności Kubernetesa, będziemy mogli szybko reagować na rosnące zapotrzebowanie i zapewnić użytkownikom ciągłość oraz wydajność usług.

## **Testowanie i monitorowanie**

W celu sprawdzenia, czy wprowadzone zmiany przynoszą oczekiwane rezultaty, zapewnione zostanie ciągłe testowanie i monitorowanie platformy. Pozwoli to na wykrycie dotychczasowych problemów gdyby powróciły oraz zauważenie nowych, które nie zostały zidentyfikowane podczas tej analizy.

Aby monitorować wydajność i ruch, skorzystamy z funkcji obserwacji Google Kubernetes Engine. Logi naszej aplikacji będą wysyłane do usługi Cloud Logging, gdzie będziemy mieli możliwość analizy, przeszukiwania i monitorowania logów w czasie rzeczywistym. Natomiast metryki dotyczące zdrowia klastra Kubernetes, zużycia zasobów i innych parametrów środowiska będą zbierane i wysyłane do usługi Cloud Monitoring. Dzięki

temu będziemy mieć wgląd w działanie klastra oraz możliwość konfigurowania i wizualizowania niestandardowych wskaźników wydajności.

Używając narzędzia Prometheus w Google Kubernetes Engine, będziemy mogli ustawiać alerty, które zostaną wywołane w sytuacjach krytycznych, takich jak przekroczenie określonych wartości granicznych dla metryk monitorowanych przez nasz klaster Kubernetes. Dzięki temu szybko zareagujemy na potencjalne problemy i zagrożenia dla stabilności i wydajności naszej aplikacji działającej na GKE.

Dodatkową zaletą Cloud Monitoringu jest możliwość konfiguracji powiadomień na podstawie wybranych zdarzeń. Możemy zdefiniować warunki grupowania zdarzeń, co pozwala na otrzymywanie powiadomień tylko wtedy, gdy określone zdarzenie wystąpi wielokrotnie. Na przykład, możemy ustawić powiadomienie tak, że zostaniemy poinformowani dopiero po tym, gdy konkretne zdarzenie wystąpiło dziesięć razy. Dzięki temu unikamy niepotrzebnych interwencji przy jednorazowych incydentach, które nie powtórzą się w przyszłości.

Możemy też wykorzystać metryki do analizowania trendów w naszych danych. Metryki pozwolą nam zidentyfikować, które operacje użytkowników generują największe obciążenie naszej aplikacji. Dzięki temu będziemy mogli śledzić wzorce użytkowania oraz identyfikować momenty, gdy nasza aplikacja jest najbardziej obciążona. Na podstawie tych informacji będziemy mogli podejmować decyzje dotyczące optymalizacji zasobów, skalowania aplikacji oraz planowania długoterminowych strategii rozwoju naszej infrastruktury IT.

Do testowania aplikacji wykorzystamy Cloud Functions, które oferują możliwość testowania dwóch rodzajów funkcji: funkcji powiązanych z żądaniami HTTP oraz funkcji wywoływanych przez eventy zachodzące na naszej platformie. Cloud Functions umożliwiają łatwe testowanie funkcji, które reagują na żądania HTTP. Możemy symulować różne typy żądań (GET, POST, itp.) i testować reakcję funkcji na różne scenariusze, takie jak przekazywanie parametrów, obsługa błędów, czy sprawdzanie wydajności. Cloud Functions pozwalają również na testowanie funkcji, które są wywoływane przez zdarzenia w naszej platformie, na przykład przez zmiany w systemie plików, zdarzenia Firebase, czy wiadomości pub/sub w Google Cloud Pub/Sub. Możemy symulować te zdarzenia i sprawdzać, czy funkcje reagują poprawnie na pojawiające się eventy.

Dodatkowo Cloud Functions może zostać podłączone z mechanizmem CI/CD na platformie GIT, więc możemy mieć pewność, że w przyszłości, rozbudowując naszą platformę, każda z jej wersji poprawnie przejdzie testy napisane przez nas w przeszłości.

Dzięki takim testom możemy upewnić się, że nasze funkcje działają zgodnie z oczekiwaniami. To pozwala na lepsze zrozumienie zachowania aplikacji w różnych scenariuszach oraz zminimalizowanie ryzyka wystąpienia błędów.

## Podsumowanie

Plan modernizacji naszej platformy e-commerce został stworzony jako odpowiedź na dostrzeżone problemy związane z jej wydajnością i niezawodnością. Zidentyfikowano szereg narzędzi, które pozwolą na poprawę tych aspektów. Te narzędzia to Compute Engine, Kubernetes Engine (GKE), GKE Autopilot, Cloud SQL, Cloud Functions, Cloud Monitoring i Logging.

Compute Engine zostanie wykorzystany jako główny silnik serwujący naszą platformę. Jego kluczową cechą jest możliwość nieprzerwanego działania, nawet podczas migracji. Dzięki temu nasza platforma będzie nieustannie dostępna dla użytkowników. Compute Engine oferuje również możliwość skalowania zasobów w zależności od aktualnych potrzeb, co pozwala na optymalizację wykorzystania pamięci.

Kolejnym ważnym narzędziem będzie Kubernetes Engine. Jest to platforma do zarządzania kontenerami, która pozwala na efektywne zarządzanie i skalowanie naszych zasobów w chmurze. Dzięki Kubernetes Engine będziemy mogli równomiernie rozłożyć obciążenie między różnymi instancjami naszej platformy, co zapewni jej stabilność i wydajność.

Aby dodatkowo usprawnić proces zarządzania zasobami, wprowadzimy GKE Autopilot. To narzędzie pozwoli na automatyczne dostosowanie pamięci do potrzeb poszczególnych części naszego sklepu, co przyczyni się do optymalizacji alokacji zasobów. Zarządzanie naszą bazą danych będzie możliwe dzięki wykorzystaniu Cloud SQL. Ta usługa pozwoli nam uniknąć konieczności utrzymania własnej infrastruktury bazodanowej i skupić się na rozwoju naszej platformy. Cloud SQL zapewni wysoką dostępność i skalowalność naszej bazy danych, a także zapewni bezpieczeństwo danych dzięki możliwości automatycznego tworzenia kopii zapasowych.

Do monitorowania wydajności i ruchu na naszej platformie wykorzystamy Cloud Monitoring i Logging. Te narzędzia pozwolą nam na analizę, przeszukiwanie i monitorowanie logów w czasie rzeczywistym oraz na zbieranie i analizowanie metryk dotyczących zdrowia klastra Kubernetes, zużycia zasobów i innych parametrów środowiska.

Wreszcie, do testowania naszej aplikacji wykorzystamy Cloud Functions. Dzięki temu narzędziu będziemy mogli przeprowadzać testy różnych funkcji naszej aplikacji i upewnić się, że działają one zgodnie z oczekiwaniami. To pozwoli na lepsze zrozumienie zachowania aplikacji w różnych scenariuszach oraz zminimalizowanie ryzyka wystąpienia błędów.

Wszystkie te zmiany wprowadzane będą stopniowo, zgodnie z planem procesu modernizacji. Dzięki temu będziemy mogli zminimalizować zakłócenia dla użytkowników i zapewnić płynność przejścia na nową architekturę.