Projektowanie i wdrażanie systemów w chmurze

Lista zadań na pracownię 15.12.17

1. [0 pkt] Na rozgrzewkę wykonaj sześcioetapowy, interaktywny samouczek o podstawach systemu Kubernetes: https://kubernetes.io/docs/tutorials/kubernetes-basics/
2. * [0 pkt extra] Przemyśl, jak mogłaby wyglądać infrastruktura pod platformę oferującą interaktywne samouczki podobne do tych z powyższego zadania. Jakie technologie mogą być przydatne?

3. [12 pkt] Zbudujemy nieduży system udający architekturę mikroserwisów do obsługi prostej strony internetowej. Tematyka strony oraz rodzaj danych, jakie będzie przetwarzać, jest dowolna. Nie interesuje nas treść ani wygląd strony, tylko organizacja infrastruktury.

   Z punktu widzenia użytkownika strona ma być w stanie przyjąć jakieś dane (zwykłym prostym formularzem + HTTP GET/POST), zapisać je do bazy danych, oraz umożliwić użytkownikowi przeglądanie danych w bazie (np. prosta wyszukiwarka, lookup w bazie albo jakieś statystyki z zebranych rekordów).

   Podzielmy taki system na cztery komponenty:
   A) Aplikacja prezentująca stronę do użytkownika
   B) Aplikacja wprowadzająca wpisy do bazy
   C) Aplikacja przeprowadzająca odczyty/statystyki z danych
   D) Baza danych przechowująca dane

   Do części D użyj solidnej bazy zarządzanej przez Google (Cloud SQL).

   Pozostałe części będziemy uruchamiać w klastrze kontenerów. Każda z trzech aplikacji będzie spakowana w osobny obraz kontenera. Wyobrazimy sobie, że development każdego z tych trzech komponentów będzie prowadzony niezależnie (w celach edukacyjnych *możesz* je napisać w różnych językach).

   Aplikacja A nie powinna komunikować się bezpośrednio z bazą. Zamiast tego powinna **wysyłać zapytania HTTP** do aplikacji B oraz C. Te będą oferować wygodny interfejs dostępu do danych w bazie. Najlepiej, jeżeli dodatkowo będą przeprowadzać jakieś proste przekształcenie na danych, które przekazują (np. aplikacja C może obliczać jakieś rozmaite statystyki i zwracać listę wyników).

   Porada: Aplikacje B i C wygodnie jest zrobić w Pythonie za pomocą biblioteki Flask. Aplikację A wygodnie jest zrobić w PHP. Można natomiast używać dowolnych języków i bibliotek.

   Każdą z aplikacji nazwij nieco sensowniej oraz opakuj w osobny obraz Dockera i zamieść wszystkie powstałe obrazy w Google Container Registry (GCR). Zadbaj, aby konfiguracja aplikacji (adres/hasło do bazy, adresy do aplikacji B i C) nie była zawarta w obrazie, tylko pobierana ze zmiennych środowiskowych.

   Uruchom klaster Kubernetes w GCP. Uruchom aplikacje A-C w kontenerach na tym klastrze. Zadbaj, o zwiększoną dostępność usługi: każda aplikacja powinna być uruchomiona w co najmniej trzech kopiach na różnych fizycznych instancjach. Przygotuj konfigurację *deploymentu* (jednego lub kilku), która pozwoli Ci łatwo uruchomić wiele kopii aplikacji oraz łatwo wymieniać ich wersje. Przygotuj *service* (jeden lub kilka), które będą udostępniać usługi na zewnątrz klastra (być może przez zintegrowany z klastrem load-balancer). Przemyśl, co powinno znajdować się w obrębie jednego *pod*a, a co w *service*

(istnieje wiele poprawnych odpowiedzi) oraz jakie są zalety i wady wybranej przez Ciebie organizacji kontenerów.
Upewnij się, że całość jest dostępna dla użytkownika z publicznego Internetu, a dodatkowo, że system funkcjonuje poprawnie z punktu widzenia użytkownika.

Zepsuj coś w kodzie wybranej aplikacji, nazwij to “nową wersją” i zbuduj nowy obraz Dockera. Zasymuluj release aplikacji, umieszczając nowe wersje obrazów w klastrze. Sprawdź, że klaster używa “zepsutego” obrazu. Zaobserwuj, jak zachowuje się serwis z punktu widzenia użytkownika. Za pomocą *deploymentu* wykonaj w klastrze rollback do wcześniejszej wersji.

Szczegółowa punktacja:

- [4 pkt] Przygotowanie apek
- [1 pkt] Zbudowanie obrazów
- [4 pkt] Konfiguracja kubernetesa
- [1 pkt] Całość działa poprawnie
- [2 pkt] Aktualizacja aplikacji i rollback

4. * [2 pkt extra] Skonfiguruj dowolne narzędzie CI/CD (np. CircleCI lub TravisCI) tak, by opublikowanie nowych commitów w Twoim repo skutkowało automatycznym zbudowaniem nowego obrazu oraz umieszczeniem go w Twoim GCR.