

Projektowanie i wdrażanie systemów w chmurze

Lista zadań na pracownię 2021.10.26

W zadaniach na tej liście uruchamiamy strony internetowe przez HTTP, ale kompletnie nie interesuje nas treść na tych stronach - mogą być brzydkie i niedopracowane. Skupiamy się na uruchamianiu usług chmurowych, komunikacji z serwerami i konfigurowaniu ich. W poniższych zadaniach mówiąc "serwer" mamy oczywiście na myśli *instance* w AWS EC2 lub *VM instance* w *Google Compute Engine*.

1. [0 pkt.] Aktywuj swój kupon do Google Cloud. Rozejrzyj się po interfejsie, zajrzyj w dział w Billing i zapamiętaj by go od czasu do czasu sprawdzać. Przygotuj narzędzie CLI (`gcloud`) do pracy.
2. [2 pkt.] Zainstaluj klasyczny serwer HTTP¹ (np. *apache2* lub *nginx*) na serwerze i skonfiguruj go tak, by serwował prostą, ale nie całkiem statyczną, stronę, złożoną np. z prostego pliku *.php* oraz 2 obrazków i jakiegoś szablonu *.css*. Jeżeli nie lubisz PHP, możesz oczywiście wybrać dowolny inny język, byleby było widać, że treść strony nie jest statyczna. Precyzyjniej ujmując: przynajmniej część treści strony, widoczna dla użytkownika, musi być generowana przez serwer przy każdych odwiedzinach. Dla przykładu może np. wyświetlać losowy tekst lub licznik odwiedzin - ale nie skupiamy się na programowaniu ciekawych aplikacji.
Obrazki oraz CSS trzymaj nie na serwerze, tylko np. Google Cloud Storage (lub S3) i wstaw odpowiednie łącza w dokument. Upewnij się, że strona działa poprawnie i jest publicznie dostępna.
Zastanów się, jakie narzędzia lub triki mogłyby ułatwić Ci koordynację i publikację jednoczesnych zmian strony na serwer i danych na Cloud Storage.
3. [2 pkt.] Uruchom nieduży serwer i zainstaluj na nim porządnego CMSa np. *Wordpress*, *Drupal*, *Joomla* lub podobne narzędzie. Możesz też wybrać inny CMS, ale niech przynajmniej używa bazy danych. Nie używaj gotowych obrazów serwerów z zainstalowanym CMSem, użyj serwera z "czystym" distro linuksowym i samodzielnie zainstaluj i skonfiguruj CMS. Do bazy danych użyj właściwej usługi chmurowej, nie instaluj jej samemu na serwerze. Upewnij się, że całość działa poprawnie i da się sterować CMSem z poziomu strony internetowej, ale nie musisz do niego wprowadzać żadnej konkretnej treści.
4. [4 pkt.] Przygotuj (prosty, nie musi być super niezawodny ani elegancko zorganizowany) skrypt/program w dowolnym języku, który uruchomi nową instancję, połączy się z nią, zainstaluje na niej serwer HTTP i przygotuje jakąś minimalną (sam jeden *index.html* wystarczy) statyczną stronę, a następnie utworzy **obraz** takiego serwera, do późniejszego wykorzystania². Zweryfikuj, że nowe serwery, ręcznie uruchamiane z użyciem obrazu przygotowanego przez Twój skrypt, faktycznie od razu serwują tę prostą stronkę. Przygotuj szablon instancji tworzący takie serwery, a następnie użyj *instance group* by automatycznie wymienić tę instancję na nową, gdy wystąpią z nią jakieś problemy (np. gdy serwer HTTP przestanie poprawnie odpowiadać, co łatwo zasymulować). Zastanów się nad wadami i zaletami takiej automatyzacji.

Klucz/token do programowej komunikacji z GCP można uzyskać np. tworząc *service account* w dziale IAM - trzeba tylko zadbać, by takie konto posiadało uprawnienia wymagane przez naszą aplikację. Alternatywnie, można użyć grupy komend `gcloud auth`

¹ Taki tradycyjny program wyspecjalizowany w serwowaniu plików i/lub uruchamianiu interpreterów. Własny serwer (np. *express.js*) nie wystarczy: chodzi o to by zasmakować konfigurowania klasycznego serwera HTTP.

² Istnieją gotowe programy wyspecjalizowane w konfigurowaniu serwerów czy tworzeniu obrazów. Poznamy je już wkrótce, ale na razie z nich nie korzystamy - celem tego zadania jest poćwiczyć samodzielne sterowanie chmurą z poziomu własnego programu i własnoręczną automatyzację operacji na serwerze.

`application-default` aby przygotować w systemie domyślny, uniwersalny żeton dostępu, który będzie automatycznie używany przez wszystkie aplikacje korzystające z bibliotek do GCP, jeżeli nie znajdą one innych kluczy.