Projektowanie i wdrażanie systemów w chmurze

Lista zadań na ćwiczenia 2021.11.30

Przykłady, które pokazaliśmy na wykładzie, znajdują się tutaj: https://github.com/rafalcieslak/cloud2021/tree/master/terraform - wykonaliśmy je Terraformem v1.0.11 (provider google 4.1.0).
Zalecamy ograniczenie wydatków używając komendy `terraform destroy` po zakończonej pracy, która kasuje całą infrastrukturę opisaną w konfiguracji (kolejne użycie `terraform apply` utworzy wszystkie zasoby od nowa).
Systemy, które będziemy tworzyć w poniższych zadaniach, są bardzo podobne do tych z poprzednich pracowni. Częściowe wykorzystanie poprzednich rozwiązań (np. obrazów serwerów) pomoże zaoszczędzić czas i skupić się na sednie tej listy - pracy z Terraformem.

1. [5 pkt] Tworzymy prostą infrastrukturę składającą się z jednego serwera www i chmurowej bazy danych. Opisz przy pomocy Terraforma konfigurację zasobów chmurowych.
   a. Serwer powinien posiadać prywatny i publiczny adres IP. Prywatny adres IP powinien należeć do specjalnie stworzonej podsieci dla serwerów www.
   b. Firewall chmurowy dotyczący serwera powinien być ustawiony tak, aby serwer www akceptował wyłącznie połączenia:
      - na porcie 80 i 443 z całego Internetu oraz
      - na porcie 22 (ssh) z określonej grupy adresów IP (np. tylko z Twojego adresu IP i sieci w Instytucie Informatyki UWr).
   c. Użyj Cloud SQL by przygotować bazę danych. Baza danych musi akceptować wyłącznie połączenia przychodzące z serwera www (wykorzystaj fakt, że tworząc serwer www poznasz jego adres IP i możesz przekazać go jako parametr do konfiguracji bazy danych).

   Nie musisz instalować żadnych ciekawych aplikacji na serwerze, ale przynajmniej wytestuj łączność do serwera i między serwerem a BD.

2. [6 pkt] Opisz Terraformem konfigurację środowiska z poprzedniej pracowni: load-balancer HTTP + kilka serwerów aplikacji. W tym celu przygotuj dwa moduły, które mogły by się przydać w przyszłości: jeden moduł będzie opisywał konfigurację load-balancera, drugi będzie przygotowywał serwery aplikacji. Specyfikacja argumentów (variables) i atrybutów (output) modułów powinna być taka, aby dało się je wygodnie używać w różnych scenariuszach, ale w szczególności:
   a. Moduł serwerów aplikacji powinien:
      - Przyjmować argumenty określające: pożądaną liczbę serwerów, rodzaj instancji na serwery.
      - Przygotować nową podsieć w której będą uruchamiane serwery oraz rozsądne reguły zapory ogniowej. Serwery aplikacji nie powinny używać publicznych IP.
      - Przygotować serwery tak, by od razu serwowały witrynę HTTP, bez konieczności ręcznej konfiguracji.
      - Adresy serwerów aplikacji powinny być udostępnione jako wyjście z tego modułu.
   b. Moduł load-balancera powinien:
      - Przyjmować argument określający adresy IP serwerów, do których load-balancer ma kierować zapytania[1].
      - Automatycznie przygotować własny serwer z load-balancerem oraz skonfigurować go, by kierował ruch do wskazanych serwerów. Najłatwiej będzie to zrobić używając obrazu serwera, oraz składając *startup script* instancji tak, aby serwer na starcie odpowiednio ustawił konfigurację.
      - Zarezerwować publiczny adres IP i przygotować podsieć, w której będzie umieszczona instancja serwera oraz właściwe dla takiego load-balancera reguły zapory ogniowej.
      - Adres IP, pod którym dostępny jest load-balancer, udostępnić jako wyjście modułu.

[1] Dokładny format danych w tym argumencie określ wedle uznania tak, by moduł był wygodny w użyciu.

Zastanów się, czy nie ma więcej argumentów/wyników, które warto, by te moduły obsługiwały.
Aby wypróbować oba moduły, napisz prostą konfigurację chmury, która używa obu modułów, podając im przykładowe wartości argumentów oraz łącząc je w jeden współpracujący system.

3. [*1 pkt extra] Przepisz moduł serwerów aplikacji z poprzedniego zadania tak, by nie zarządzał N sztukami serwerów, tylko aby przygotowywał *instance group* o stałym rozmiarze N.