

# Projektowanie i wdrażanie systemów w chmurze

## Lista zadań na pracownię 05.01.18

Może się okazać, że nowe konta AWS mają strasznie niski limit na liczbę uruchomionych instancji EC2 (np. 1). W takim przypadku wystarczy skontaktować się z supportem AWS używając linku z menu po zalogowaniu się na konsolkę internetową. Prosząc o podniesienie limitu na instancje wyjaśnijcie, że używacie tej usługi w ramach zajęć na uczelni. Nie będą o nic więcej pytać, ale być może trzeba będzie trochę poczekać (kilka godzin, może dzień).

W ramach tej pracowni nie oceniamy sposobu konfiguracji serwerów, nie trzeba używać Ansible.

Kolejne zadania używają podobnych konfiguracji stopniowo zwiększając wymagania; dozwolone i wskazane jest oddawanie rozwiązań prezentując jeden komplet serwerów / usług chmurowych do wszystkich zadań.

Tworzenie oraz usuwanie load-balancerów w AWS zajmuje 10-30 minut. A warto usunąć po pracy, bo miesiąc używania jednego LB kosztuje ok 20\$.

1. [1 pkt] Na rozgrzewkę uruchom jedną instancję EC2 z wybranym systemem operacyjnym, skonfiguruj ją by używała klucza SSH do którego masz dostęp, ustaw związaną z nią *security group* tak, aby dało się połączyć przez SSH oraz HTTP, przydziel tej instancji publiczny adres IP i połącz się z nią przez SSH. Zainstaluj też serwer HTTP i upewnij się, że jest publicznie dostępny.
2. \* [2 pkt extra] Wykonując zadania 2 - 4 użyj Terraforma.  
Właściwie to zadanie powinno być warte ujemne punkty, ponieważ z Terraformem jest dużo łatwiej ☹.
3. [4 pkt] Przygotuj trzy identycznie skonfigurowane instancje z serwerami HTTP w różnych strefach dostępności. Wykorzystaj AWS ELB aby równoważyć ruch między trzema serwerami. Użyj AWS RDS aby przygotować wysokodostępna bazę danych, której będą używać te serwery (nie muszą robić nic pożytecznego, wystarczy, że wykażą, że mają łączność i dostęp do bazy). Nie używaj domyślnych podsieci, skonfiguruj VPC samodzielnie. Zwróć uwagę na konfigurację następujących elementów:
  - a. VPC, kilka podsieci (osobne dla instancji RDS, interfejsów LB, oraz serwerów HTTP).  
Uwaga: w AWS jedna podsieć może być wewnątrz tylko jednej AZ, więc potrzebne będą aż trzy sieci na serwery HTTP.
  - b. *internet gateway*, route tables (określające któredy ma być kierowany ruch w każdej podsieci)
  - c. *security groups* dla serwerów i dla RDS
  - d. ELB (powinien być publicznie dostępny)Upewnij się, że całość działa poprawnie, a serwery są dostępne z publicznego internetu przez load-balancer.
4. [2 pkt] Wykonaj zadanie 3 z dodatkową uwagą: niech serwery HTTP nie posiadają publicznego adresu IP (bo nie jest im potrzebny). To oznacza, że musisz skonfigurować NAT gateway, aby mogły się połączyć z internetem (np. by zaktualizować oprogramowanie). Dodatkowo przygotuj jedną dodatkową instancję w nowej podsieci, która posłuży jako *bastion host* z publicznym adresem IP. Dzięki temu aby połączyć się SSH do serwerów HTTP będzie można połączyć się wpierw do *bastion hosta*, a następnie do właściwego serwera.
5. [1 pkt] Przygotuj własny AMI (obraz maszyny) z gotową konfiguracją serwera HTTP jak w zadaniu 3. Zadbaj, by instancje uruchomione z takiego AMI były od razu gotowe do pracy za load-balancerem (bez dodatkowej konfiguracji po stronie serwera).  
Taki obraz można wykonać ręcznie używając konsoli AWS lub AWS CLI, ale można też skorzystać z gotowego narzędzia, np. Packer.
6. [4 pkt] Utwórz grupę autoskalującą, która będzie uruchamiać serwery HTTP na podstawie obrazu AMI z poprzedniego zadania. Zadbaj, by serwery uruchamiały się w różnych strefach dostępności. Powiąż grupę autoskalującą z load-balancerem (zrezygnuj z "ręcznie uruchomionych" instancji serwerów HTTP z poprzednich zadań). Zmień pożądaną liczbę serwerów w obrębie grupy; upewnij się, że brakujące serwery są uruchamiane poprawnie, a nadmiarowe są wyłączane automatycznie. Skonfiguruj proste health-checki do serwerów w grupie autoskalującej. Następnie zasymuluj awarię na wybranym serwerze: na przykład zaloguj się na niego i wywołaj *kernel panic*, zawieś jakimś procesem system albo wyłącz *nginx*. Zaobserwuj jak zachowa się grupa autoskalująca, spróbuj ocenić jaki ma to efekt z punktu widzenia użytkowników.