

# Projektowanie i wdrażanie systemów w chmurze

Lista zadań na pracownię 2021.11.16

Będziemy przygotowywać środowisko złożone z kilku serwerów HTTP, podłączonych do wspólnego load balancera. Użyjemy trzech rodzajów serwerów w systemie:

- a) Serwery aplikacji (z serwerem HTTP)
- b) Load-balancer (przyjmujący zapytania HTTP i rozdzielający je na serwery aplikacji)
- c) Serwer CMS (prezentujący niezależną "stronę główną", nie związany z load-balancerem)

Planujemy używać jednego load-balancera, ale serwerów aplikacji powinno być więcej, np. trzy. Przygotuj konfigurację serwerów opisanych poniżej za pomocą Ansible. Upewnij się, że konfiguracja jest kompletna, tzn. nie wymaga żadnych ręcznych akcji na serwerach.

1. Przygotuj playbooks dla każdego rodzaju serwera. Najlepiej użyj ról Ansible, aby opisać grupy zadań, a następnie wykorzystaj je w playbookach.
  - a. [1 pkt] Serwery aplikacji powinny mieć uruchomiony serwer HTTP (np. *nginx* lub *apache*) udostępniający jakąś prostą witrynę - może być całkiem banalna, statyczna, ale nie całkiem pusta<sup>1</sup>.
  - b. [1 pkt] Load-balancer, powinien używać np. *haproxy* (lub *nginx* w trybie reverse-proxy) z taką konfiguracją, by przekazywał zapytania HTTP do serwerów aplikacji. W tym celu konfiguracja load-balancera może potrzebować adresów IP serwerów aplikacji, a ponieważ te mogą się zmieniać, przekaż je w konfiguracji jako zmienne (ustawiane w playbooku, inventory albo z linii komend przy wykonywaniu playbooku).
  - c. [1 pkt] Serwer CMS pełniący rolę strony głównej dla naszej aplikacji musi mieć zainstalowany dowolny CMS<sup>2</sup>. Cały proces instalacji powinien być w pełni zautomatyzowany. Podobnie do poprzedniej pracowni, użyjmy bazy zarządzanej w chmurze<sup>3</sup> (wszystkie detale połączenia należy sparametryzować, by wydzielić je od procesu instalacji CMSa).
2. [1 pkt] Wszystkie serwery powinny odmawiać logowania SSH za pomocą hasła (używamy tylko par kluczy). Do tego celu może przydać się opcja *PasswordAuthentication* w pliku */etc/ssh/sshd\_config* oraz moduł Ansible *replace*, który pozwala wygodnie nadpisać drobny fragment pliku. Dodatkowo, ustaw (ten sam plik konfiguracyjny) banner w stylu MOTD, który powita każdego, kto zaloguje się na serwer używając SSH.
3. [1 pkt] Musimy zbierać logi serwerów HTTP w bezpieczne miejsce - poza serwerami aplikacji. Istnieją dedykowane programy, które zajmują się tym zadaniem (można ich użyć), ale na nasze potrzeby wystarczy dużo prostsze rozwiązanie - np. *cron job* który raz na kilka minut będzie wysyłał pliki logów serwera HTTP np. do Cloud Storage, albo w inne bezpieczne miejsce.

Wykonując powyższe ćwiczenia Ansible aktualizuje pracujące serwery, co ma różne wady (jakie?). W następujących zadaniach wypróbujemy *immutable servers*.

4. [1 pkt] Przygotuj konfigurację Packera, która zbuduje obrazy serwerów każdego rodzaju z konfiguracji Ansible przygotowanej w poprzednich zadaniach, oraz przygotuj te obrazy.
5. [1 pkt] Przeprowadź zmianę konfiguracji serwera nie modyfikując żadnej działającej instancji. W tym celu zmień coś w rolach/playbooku (np. treść strony HTTP), zbuduj nowy obraz Packerem, a następnie podmień instancje na nowe, używając świeżego obrazu.
6. Dodatkowo przyznajemy:
  - a. [1pkt] za elegancką organizację kodu Ansible (poprawne użycie ról, inventory, zmienne),
  - b. [1pkt] jeżeli po wykonaniu wszystkich powyższych zadań cały system działa poprawnie.

<sup>1</sup> Jeśli chcesz, sparametryzuj tego playbooka tak, by każdy z serwerów aplikacji wyświetlał odrobinę inną stronę. W ten sposób łatwo sprawdzisz, że zapytania wysłane do load-balancera są faktycznie obsługiwane przez różne serwery aplikacji.

<sup>2</sup> Dla zabawy warto spróbować coś innego niż Wordpress - prawie wszystkie CMSy instaluje się bardzo podobnie.

<sup>3</sup> Istnieją moduły Ansible które potrafią przygotowywać zasoby w chmurze i można by za ich pomocą uruchomić bazę w GCP SQL. Ale odradzamy to podejście - skupmy się na używaniu Ansible do konfigurowania serwerów, a wkrótce poznamy lepsze narzędzia do zarządzania zasobami chmurowymi.

7. [\*1 pkt extra] Tworząc serwery z obrazów, już nie musimy łączyć się do nich by je konfigurować za pomocą Ansible. Wykorzystaj ten fakt, by uruchomić serwery aplikacji bez publicznego adresu IP, odizolowane od publicznego Internetu. Skonfiguruj połączenie pomiędzy load-balancerem a serwerami aplikacji tak, by używało połączeń w sieci wewnętrznej, za pomocą prywatnych adresów IP.

Przykłady, które pokazaliśmy na wykładzie dostępne są tutaj: <https://github.com/rafalcieslak/cloud2021/>

Dokumentacja wbudowanych modułów Ansible:

<https://docs.ansible.com/ansible/latest/collections/ansible/builtin/index.html#plugins-in-ansible-builtin>

Szczegóły jak przygotować *playbook*: [https://docs.ansible.com/ansible/latest/user\\_guide/playbooks.html](https://docs.ansible.com/ansible/latest/user_guide/playbooks.html)

Szczegóły o tworzeniu ról: [https://docs.ansible.com/ansible/latest/user\\_guide/playbooks\\_reuse\\_roles.html](https://docs.ansible.com/ansible/latest/user_guide/playbooks_reuse_roles.html)

Dokumentacja Packera: <https://www.packer.io/docs/>