

# Wirtualne Sieci Obliczeniowe

Bartłomiej Krawczyk, Mateusz Brzozowski

## Automatyzacja: Skalowalność i wysoka dostępność - serwisy bezstanowe

Skalowalność i wysoka dostępność

Funkcje:

- automatyzacja skalowania usługi (dodawanie/usuwanie VM),
- równoważenie obciążenia (np. haproxy)
- zwiększanie niezawodności.

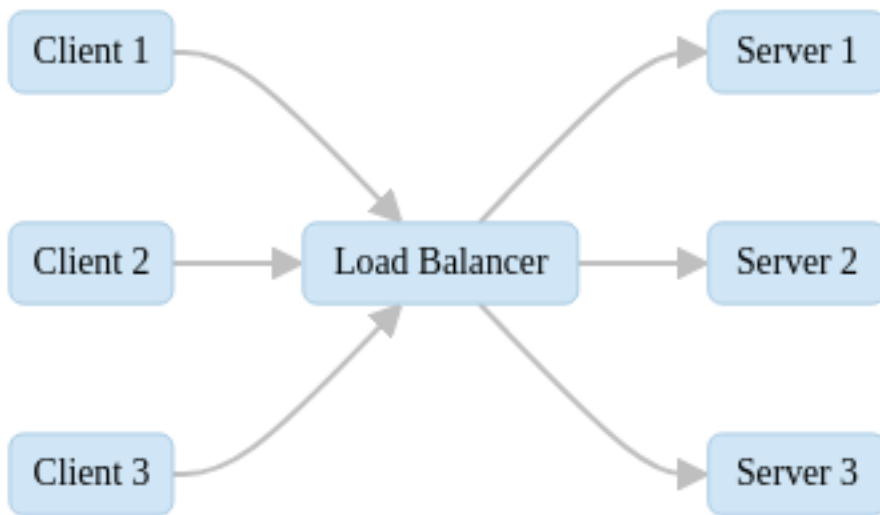
Sprawdzić jaki wpływ na obsługę ma awaria jednej/większej liczby maszyn. Serwis bezstanowy.

Automatyzacja zarządzania maszynami wirtualnymi:

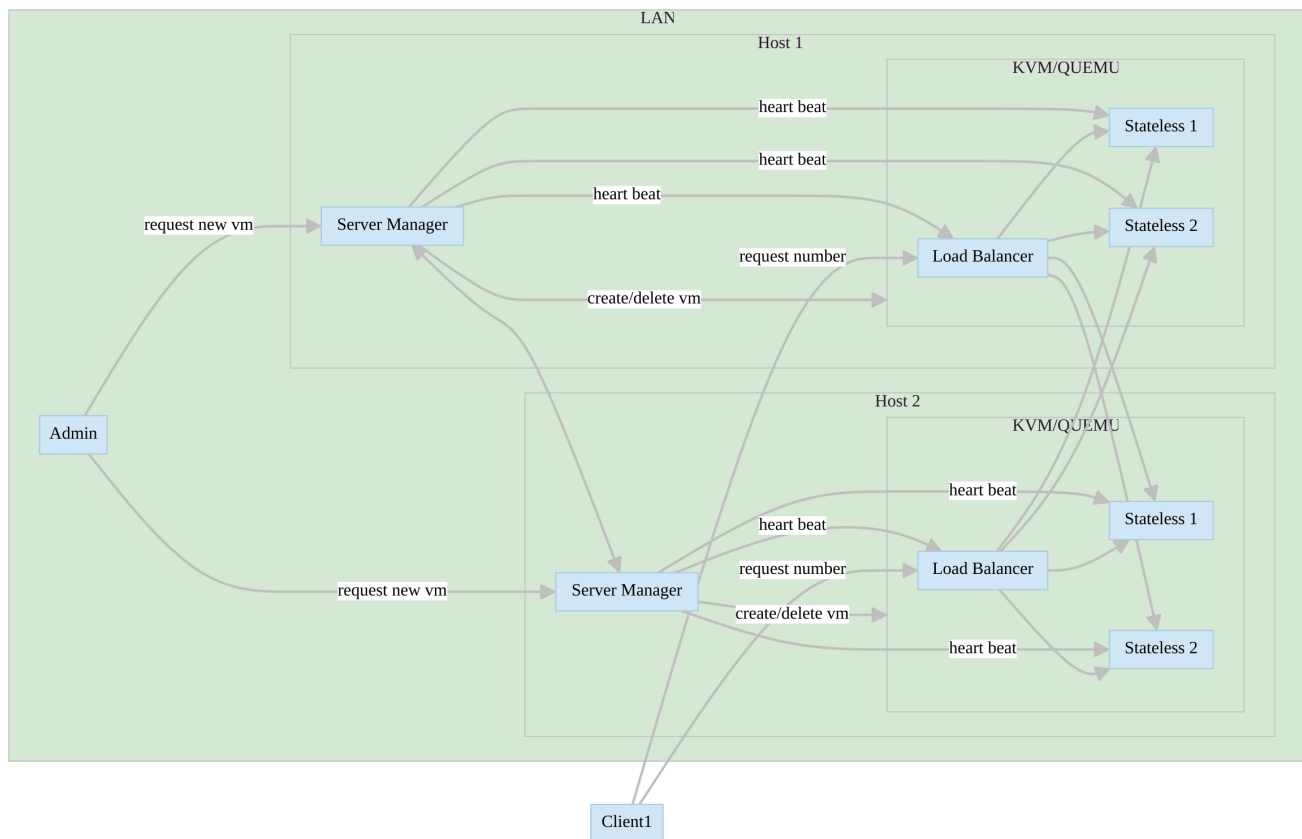
- Zestaw skryptów/program ułatwiający zarządzanie VM/klastrem
- Opracować odpowiedni scenariusz
- Użycie narzędzi!

## Scenariusz

- wiele serwisów bezstanowych, które zwracają losową wartość (bool, int, float, double)
- całych ruch przechodzi przez load-balancer (nginx), postawiony na oddzielnej maszynie wirtualnej, jeden dla każdego serwera w klastrze



- na każdym laptopie w naszym klastrze uruchomiony jest daemon manager, który zarządza maszynami wirtualnymi
- każdy manager posiada endpointy do utworzenia i usuwania maszyny wirtualnej z serwisem bezstanowym, do aktualizacji serwisów z innych menadżerów
- menadżerzy mają skonfigurowane swoje adresy ip, propagują między sobą informacje o uruchomionych maszynach



## Ryzyko

- Maszyna z serwisem bezstanowym umiera:

Menadżer przez cały czas działa serwisu utrzymuje połączenie heartbeat, serwis co jakiś czas wysyła wiadomość zwrótną o treści: `data: {"status": "OK"}` sygnalizującą poprawne działanie serwisu. Jeśli menadżer nie wykryje przez określony czas połączenia, kilkakrotnie próbuje nawiązanie połączenia, jeśli się to nie uda to usuwamy taką maszynę i menadżer stawia nową maszynę w jej miejsce.

Adres ip nowej maszyny jest taki sam jak adres ip starej maszyny. W przypadku błędnej odpowiedzi serwera, odpytuje kolejną maszynę (proxy\_next\_upstream error).

- Maszyna z load balancerem umiera

Heartbeat w ramach load balancera działa podobnie jak w serwisie bezstanowym

Mamy jeden publiczny adres ip, który jest na starcie przypisany do jednego menadżera, jeśli menadżer ma problem ze swoim load balancerem, to mianuje drugiego menadżera głównym i przypisuje do niego publiczny adres ip, a nasz load balancer wyłączamy i próbujemy postawić na nowo z innym adresem ip.

## Sprzet

Dwa laptopy z systemem Ubuntu w tej samej sieci wifi.

## Narzędzia

- Wirtualizator: KVM
- Serwisy: Spring w kotlinie
- Zarządzanie konfiguracją maszyn: Ansible
- Skrypty testowe: Bash

## Plan testów

Skrypt który odpytuje load balancer, działa z pewnym opóźnieniem, cały czas dopóki nie zostanie zatrzymany. W oddzielnej konsoli wyłączymy maszynę bezstanową/ load balancer, podobnie na drugim laptopie i monitorujemy jak zachowuje się serwis, czy wszystkie odpowiedzi zwracane są poprawne.