

Scale services with load balancing

<https://github.com/knehez/isi> - folder example2

HAProxy is an open-source software that provides High Availability services, load balancing, and proxying for **TCP** and **HTTP**-based applications. It is used to distribute incoming network traffic across multiple servers to improve performance, scalability, and reliability of applications. **HAProxy** acts as a reverse proxy, meaning that it receives requests from clients and forwards them to the appropriate server based on various criteria such as load balancing algorithms, server health checks, and session persistence.

docker-compose.yml

```
version: "3.3"
services:
  web:
    build: .
    ports:
      - "5000"
  redis:
    image: "redis:alpine"
  haproxy:
    image: "haproxytech/haproxy-alpine:2.4"
    volumes:
      - ./haproxy.cfg:/usr/local/etc/haproxy/haproxy.cfg:ro
    depends_on:
      - web
    ports:
      - "80:80"
```

This is a sample HAProxy configuration that defines various global settings, default settings, and frontend/backend configurations. Here's an explanation of each section:

1. The global section defines global settings for the entire HAProxy configuration. In this configuration, it specifies the location of the HAProxy stats socket at `/var/run/api.sock` and sets the user and group ownership to `haproxy`. It also specifies the log format and location, where logs are sent to `stdout` and use the raw format with a logging level of `info`.
2. The defaults section defines default settings for all frontends and backends. In this configuration, it sets the mode to `http`, which means that it expects HTTP requests. It also sets various timeout values, including client, connect, server, and HTTP request timeouts. Additionally, it enables global logging.

The frontend stats section defines a frontend for the HAProxy stats interface. It binds to port 8404 and enables the stats interface. It sets the stats URI to `/` and refreshes the stats page every 10 seconds.

1. The frontend `myfrontend` section defines a frontend for incoming HTTP traffic. It binds to port 80 and specifies the mode as `http`. It also specifies the default backend `webserver`, which will receive traffic that does not match any other defined frontend.
2. The backend `webserver` section defines a backend that consists of five servers, all with the name `web` and listening on port 5000. The `check` option specifies that HAProxy should check the health of each server before forwarding traffic to it.

In summary, this configuration sets up HAProxy to listen on port 80 for incoming HTTP traffic and distribute it across five backend servers that are checked for health before traffic is forwarded. It also sets up a stats interface on port 8404 to monitor the HAProxy instance.

From:

<https://edu.iit.uni-miskolc.hu/> - **Institute of Information Science - University of Miskolc**

Permanent link:

https://edu.iit.uni-miskolc.hu/tanszek:oktatas:iss_t:docker2?rev=1681670749

Last update: **2023/04/16 18:45**

