

МИНЦИФРЫ
«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4
«Распределение запросов»
по дисциплине «Архитектура вычислительных систем»

Выполнил: студент гр. ИП-212

Жеребцов Дмитрий Евгеньевич

Новосибирск 2024

Постановка задачи

Задание.

Распределение запросов

- Реализовать схему балансировки нагрузки. Для этого запустить минимум два контейнера или VM.
- DoD на 3
 - Распределение запросов с использованием nginx
- DoD на 4
 - DoD на 3
 - Распределение запросов с использованием haproxy или traefik
- DoD на 5
 - Показать Service discovery Traefik на примере docker-провайдера (Запуск контейнера с определенными labels должен автоматически привести к появлению нового endpoint в traefik и, как следствие, включение его в пул распределения запросов)
 - Распределение запросов с помощью lvs/ipvs - для этого использовать три VM. У каждой свой ip адрес. Одна VM является фронтендом балансировщика, две другие - бэкэндами.

Выполнение работы

```
docker-compose.yml ×
services:
  backend1:
    build:
      context: .
      dockerfile: Dockerfile
      args:
        - HTML_FILE=index1.html
    networks:
      - lab4_acs
    ports:
      - "8081:80"

  backend2:
    build:
      context: .
      dockerfile: Dockerfile
      args:
        - HTML_FILE=index2.html
    networks:
      - lab4_acs
    ports:
      - "8082:80"

  nginx:
    image: nginx:latest
    networks:
      - lab4_acs
    ports:
      - "8080:80"
    depends_on:
      - backend1
      - backend2
    volumes:
      - ./nginx.conf:/etc/nginx/nginx.conf

networks:
  lab4_acs:
    external: true
```

```
nginx.conf x
# nginx.conf
events {worker_connections 1024;}

http {
    upstream backend {
        server backend1:80;
        server backend2:80;
    }

    server {
        listen 80;

        location / {
            proxy_pass http://backend;
        }
    }
}

megadrage .../ACS/lab4 ? main ? 20:42
→ curl http://localhost:8080
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Backend 1</title>
</head>
<body>
    <h1>Hello from Backend 1!</h1>
</body>
</html>

megadrage .../ACS/lab4 ? main ? 20:42
→ curl http://localhost:8080
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Backend 2</title>
</head>
<body>
    <h1>Hello from Backend 2!</h1>
</body>
</html>

megadrage .../ACS/lab4 ? main ? 20:42
```

✖ haproxy.cfg

global

log stdout format raw local0

defaults

log global

option httplog

timeout connect 500ms

timeout client 5000ms

timeout server 5000ms

frontend http_front

bind *:80

default_backend http_back

backend http_back

balance roundrobin

server backend1 backend1:80 check

server backend2 backend2:80 check

✖ traefik.yml

global:

checkNewVersion: true

sendAnonymousUsage: false

api:

dashboard: true

insecure: true

log:

level: INFO

accessLog:

filePath: "/logs/traefik.log"

format: json

filters:

statusCodes:

- "200" # log successful http requests

- "400-599" # log failed http requests

bufferingSize: 0

fields:

headers:

defaultMode: drop # drop all headers per default

names:

User-Agent: keep # log user agent strings

serverTransport:

insecureSkipVerify: true

entryPoints:

http:

address: :80

providers:

File provider for connecting things that are outside of docker / defining middleware

file:

filename: /etc/traefik/fileConfig.yml

watch: true

Docker provider for connecting all apps that are inside of the docker network

docker:

watch: true

network: lab4_acs # Add Your Docker Network Name Here

exposedByDefault: false