

nginx learning

date: 2024.6.13
author: zilin xu
resouce: 狂神说

intro

假设我们有很多client和server，我们需要一个中间人做反向代理,像是一个虚拟的服务器接受和转发我们client的请求并发送给对应的server；我们还需要通过这个中间人去分配服务器的权重，比如说server a可以搭载64G。server b可以搭载6G，我们当然希望更多的请求可以打到server a。所以我们需要中间人去做负载均衡。

This is **nginx**

这世上所有的架构问题：没有什么是加一层解决不了的问题。

正向代理vs反向代理

正向代理：比如VPN，他是代理客户端的请求。

反向代理：代理我们的服务器的，比如baidu在上海深圳北京都有服务器，但你访问永远是www.baidu.com.

负载均衡

轮询：依次循环

加权轮询

iphash：

动静分离

有些静态的请求直接通过nginx返回

install nginx and basic commands

```
→ nginxLearning git:(main) x brew install nginx
==> Downloading https://ghcr.io/v2/homebrew/core/nginx/manifests/1.27.0
#####
##### 100.0%
==> Fetching nginx
==> Downloading
https://ghcr.io/v2/homebrew/core/nginx/blobs/sha256:64223e300749be61e1
#####
##### 100.0%
==> Pouring nginx--1.27.0.arm64_sonoma.bottle.tar.gz
==> Caveats
Docroot is: /opt/homebrew/var/www
```

```
The default port has been set in /opt/homebrew/etc/nginx/nginx.conf to
8080 so that
nginx can run without sudo.
```

```
nginx will load all files in /opt/homebrew/etc/nginx/servers/.
```

```
To start nginx now and restart at login:
```

```
brew services start nginx
```

```
Or, if you don't want/need a background service you can just run:
```

```
/opt/homebrew/opt/nginx/bin/nginx -g daemon\ off\;
```

```
==> Summary
```

```
📦 /opt/homebrew/Cellar/nginx/1.27.0: 27 files, 2.4MB
```

```
==> Running `brew cleanup nginx`...
```

```
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
```

```
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
```

start and stop the nginx service

```
brew services start nginx
## then you can visit http://localhost:8080
brew services stop nginx
```

nginx path on my Mac

```
/opt/homebrew/etc/nginx/
```

```
cd /opt/homebrew/etc/nginx/
code .
```

edit `.conf` file and reload it

change the listen port into `8081`

run `brew services reload nginx`

implement a simple 负载均衡

implement two simple server using `JS`

```
mkdir serverJS
cd serverJS
touch server1.js server2.js
```

edit two `.js` file

```
const http = require('http');

const server = http.createServer((req, res) => {
  res.writeHead(200, { 'Content-Type': 'text/plain' });
  res.end('Response from Server 1\n');
});

server.listen(3001, () => {
  console.log('Server 1 is running on http://localhost:3001');
});

const http = require('http');

const server = http.createServer((req, res) => {
  res.writeHead(200, { 'Content-Type': 'text/plain' });
  res.end('Response from Server 2\n');
});

server.listen(3002, () => {
  console.log('Server 2 is running on http://localhost:3002');
});
```

start two servers

```
node server1.js
node server2.js
```

edit `nginx.conf`

```
upstream backend {
  server localhost:3001;
  server localhost:3002;
}

server {
  listen      8080;
  server_name localhost;

  location / {
    proxy_pass http://backend;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
  }
}
```

reload and start nginx

```
brew services reload nginx  
brew services restart nginx
```