

PRÁCTICA 5

Benchmarking



UNIVERSIDAD DE GRANADA

Titulación: Ingeniería Informática + ADE

FLORIN EMANUEL TODOR GLIGA

ÍNDICE

1. [Parte Básica](#)
 - a. [B1: Configuración del entorno de benchmarking](#)
 - b. [B2: Implementación con Apache Benchmark](#)
 - c. [B3: Implementación con Locust](#)
 - d. [B4: Ejecución de pruebas de carga](#)
 - i. [Locust](#)
 - ii. [Apache Benchmark](#)
 - iii. [Escalado automático](#)
 - e. [B5: Análisis de resultados](#)
2. [Parte Avanzada](#)
 - a. [A1: Desarrollar tareas avanzadas en Locustfile.](#)
 - b. [A2. Crear escenario multicontenedor con algún CMS](#)
 - c. [A3. Ejecución y Análisis de cargas de prueba avanzadas sobre CMS](#)

Parte Básica

Las prácticas están realizadas previamente al cambio que se hizo en prado con el documento, que si no mal recuerdo en vez de utilizar 6 workers en Locust se solicita que se utilice 5.

B1: Configuración del entorno de benchmarking

Como llevo comentando en todas las prácticas previas a estas, por preferencia personal voy a dejar todos los ficheros usados en las prácticas anteriores (aunque con las modificaciones necesarias para usarlo en esta práctica 5) . Por ello, voy a crear los directorios necesarios junto a los demás que tengo.

```
● > tree -d .
├── dashboard_file
├── file_sd
├── logs_apache
├── logs_envoy
├── logs_escalado
├── logs_haproxy
├── logs_nginx
├── logs_traefik
├── P5-ab
├── P5-flotodor-apache
│   ├── apache_config
│   └── P5-flotodor-iptables-web
├── P5-flotodor-certificados
│   ├── CA
│   ├── SubCA
│   └── Webs
├── P5-flotodor-envoy
├── P5-flotodor-haproxy
│   └── config_balanceador
├── P5-flotodor-nginx
│   ├── config_balanceador
│   ├── config_webs
│   └── P5-flotodor-iptables-web
├── P5-flotodor-traefik
├── P5-locust
├── prometheus_file
└── web_flotodor

28 directories
```

B2: Implementación con Apache Benchmark

Creamos el docker file de AB:

```
1  # P5-ab/DockerfileAB
2  # Imagen mínima de Debian + Apache Benchmark (ab)
3  FROM debian:bookworm-slim
4
5
6
7  # Instalamos ab y certificados raíz para poder probar HTTPS
8  RUN apt-get update \
9      && apt-get install -y --no-install-recommends apache2-utils ca-certificates curl \
10     && rm -rf /var/lib/apt/lists/*
11
12 # ab es la entrypoint por defecto; el CMD se puede sobrescribir en docker-compose
13 ENTRYPOINT ["ab"]
14
```

Creamos el docker-compose para hacer las pruebas de carga al balanceador (realmente llevo usando AB desde la práctica 1 y en mi opinión no le veo la utilidad de crear un contenedor para ejecutar estas pruebas de cargas, para las cuales, cuando terminan se eliminan el contenedor. Sin embargo, lo haré como se solicita.

Recordar al profesorado que todas las ejecuciones las hago desde mi script inicial creado en la práctica 1, es decir, desde [init.sh](#).

```

1 # P5-ab/docker-compose.yml
2
3 >Run All Services
4 services:
5   >Run Service
6   apache-benchmark-P5:
7     build:
8       context: .           # apunta al directorio P5-ab
9       dockerfile: DockerfileAB
10    image: flotodor-ab-image:p5
11    container_name: apache_benchmark-P5
12
13    # Comando por defecto:
14    # -k → ignora certificados inválidos (útil si el balanceador usa un cert. autofirmado)
15    # -n → nº total de peticiones
16    # -c → concurrencia
17    # Cambia estas cifras a tu gusto o sobreescríbelas con `docker compose run`
18    command: >
19      -k -n 10000 -c 100 https://192.168.10.50/
20
21    networks:
22      red_web:
23        ipv4_address: 192.168.10.60
24
25 networks:
26   red_web:
27     external: true

```

Para ejecutarlo hago uso de `./init.sh -u ab`

Para poder ejecutarlo, primero debemos de ejecutar por ejemplo `./init.sh -u nginx`, dando lugar a que se ejecute el docker compose del balanceador de carga de nginx. Ya posteriormente usamos `-u ab` para realizar la carga.

Al ejecutar ambas partes nos sale lo siguiente:

```

• > ./init.sh -u nginx
[✓] Red red_servicios ya existe.
[✓] Red red_web ya existe.
[i] Estrategia de balanceo: round-robin (por defecto)
[i] Comprobando puertos 8080 a 8089...
[+] Running 10/10
  ✓ Container apache_benchmark-P5 Removed
  ✓ Container web2 Started
  ✓ Container web3 Started
  ✓ Container web7 Started
  ✓ Container web4 Started
  ✓ Container web8 Started
  ✓ Container web1 Started
  ✓ Container web6 Started
  ✓ Container web5 Started
  ✓ Container nginx_balanceador Started
[+] Servicios iniciados con Nginx.
• > ./init.sh -u ab
[✓] Red red_servicios ya existe.
[✓] Red red_web ya existe.
[i] Comprobando puertos 8080 a 8089...
WARN[0000] Found orphan containers ([nginx_balanceador web7 web3 web1 web5 web8 web2 web6 web4]) for this project. If you removed or renamed this service in your compose file, you can run this command with the --remove-orphans flag to clean it up.
[+] Running 1/1
  ✓ Container apache_benchmark-P5 Started
[+] Servicios iniciados con Apache Benchmark.

```

Podemos ver que nos sale un warning, pero esto se debe a que cuando ejecuto `-u ab`, le he quitado la parte de `--remove-orphans` para que no borre los contenedores de los servidores web y del balanceador. Por eso, el warning no significa ningún “peligro”.

```

[+] Running 1/1
  ✓ Container apache_benchmark-P5 Started
[+] Servicios iniciados con Apache Benchmark.
• > docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
AMES          c44d985cc3d0   flotodor-ab-image:p5     "ab -k -n 10000 -c 1..." About a minute ago Up 1 second
pache_benchmark-P5   flotodor-nginx_balanceador-image:p5 "/docker-entrypoint..." About a minute ago Up About a minute 0.0.0.0:80->80/tcp, [::]:80->80/tcp, 0.0.0.0:443->443/tcp, [::]:443->443/tcp
ginx_balanceador    74c545d7de2c   flotodor-apache-image:p5 "/usr/local/bin/entr..." About a minute ago Up About a minute 443/tcp, 9100/tcp, 0.0.0.0:8087->80/tcp, [::]:8087->80/tcp
eb7            0f9311c8186e   flotodor-apache-image:p5 "/usr/local/bin/entr..." About a minute ago Up About a minute 443/tcp, 9100/tcp, 0.0.0.0:8083->80/tcp, [::]:8083->80/tcp

```

Podemos ver cómo se ejecuta todo correctamente.

B3: Implementación con Locust

Respecto al `docker.compose`:

```

services:
  master-flotodor:
    image: locustio/locust:latest
    hostname: master-flotodor
    command: [
      "-f", "/mnt/locust/locustfile.py",
      "--master",
      "-H", "https://192.168.10.50:443"
    ]
    ports: [ "8089:8089" ]
    volumes:
      - ./P5-locust/locustfile.py:/mnt/locust/locustfile.py:ro
    networks:
      red_web:
        aliases: [ master-flotodor ]

  worker-flotodor:
    image: locustio/locust:latest
    command: [
      "-f", "/mnt/locust/locustfile.py",
      "--worker",
      "--master-host", "master-flotodor"
    ]
    depends_on: [ master-flotodor ]
    volumes:
      - ./P5-locust/locustfile.py:/mnt/locust/locustfile.py:ro
    networks: [ red_web ]

networks:
  red_web:
    external: true

```

Respecto al fichero de python para las tareas definidas y el establecimiento de la configuración necesaria para HTTPS:

```

# P5-locust/locustfile.py
from locust import HttpUser, task, between

# ---- OPCIÓN A : solo HTTPS (lo más rápido) -----
class P5UsuariosHTTPS(HttpUser):
    """
    Simula usuarios que pasan por el balanceador HTTPS
    (-H https://192.168.10.50:443 en docker-compose).
    """
    wait_time = between(1, 5)

    @task
    def index(self):
        # OJO → URL RELATIVA, ¡nunca http://... aquí!
        self.client.get("/index.php", verify=False, name="/index.php")

```

Para esta parte voy a ejecutar las pruebas de carga con Locust, veremos que hay errores debido a la práctica anterior y que comentaré cuál es la solución que propongo.

```

• > ./init.sh -u nginx
[✓] Red red_servicios ya existe.
[✓] Red red_web ya existe.
[i] Estrategia de balanceo: round-robin (por defecto)
[i] Comprobando puertos 8080 a 8089...
[+] Running 9/9
  ✓ Container web3           Started
  ✓ Container web1           Started
  ✓ Container web2           Started
  ✓ Container web6           Started
  ✓ Container web8           Started
  ✓ Container web4           Started
  ✓ Container web5           Started
  ✓ Container web7           Started
  ✓ Container nginx_balanceador Started
[+] Servicios iniciados con Nginx.
• > ./init.sh -u locust
[✓] Red red_servicios ya existe.
[✓] Red red_web ya existe.
[i] Comprobando puertos 8080 a 8089...
WARN[0000] Found orphan containers ([nginx_balanceador web5 web1 web8 web6 web2])
d with the --remove-orphans flag to clean it up.
[+] Running 7/7
  ✓ Container p5-granjaweb-master-flotodor-1 Started
  ✓ Container p5-granjaweb-worker-flotodor-2 Started
  ✓ Container p5-granjaweb-worker-flotodor-5 Started
  ✓ Container p5-granjaweb-worker-flotodor-4 Started
  ✓ Container p5-granjaweb-worker-flotodor-1 Started
  ✓ Container p5-granjaweb-worker-flotodor-3 Started
  ✓ Container p5-granjaweb-worker-flotodor-6 Started
[+] Clúster Locust (1 master, 6 workers) en marcha.

```

~ / Escritorio/SWAP/P5-granjaweb | on main !15 72

Workers
6

10000

Ramp up (users started/second) *

100

Host

https://192.168.10.50:443

Advanced options

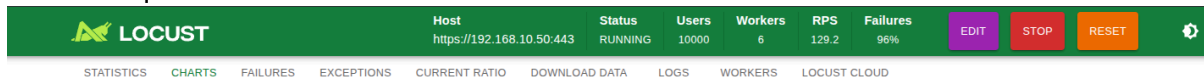
Run time (e.g. 20, 20s, 3m, 2h, 1h20m, 3h30m10s, etc.)

5m

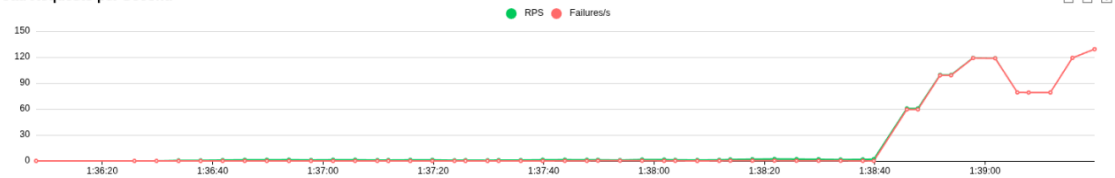
Profile

START

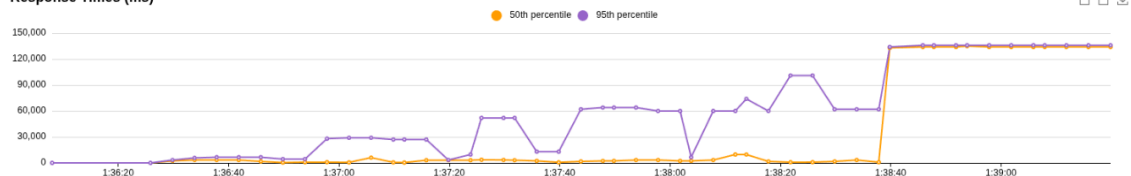
Podemos comprobar una respuesta muy extraña, pero que creo que se debe a las limitaciones que coloqué en la práctica anterior con el uso de iptables para evitar los ataques de DDoS.



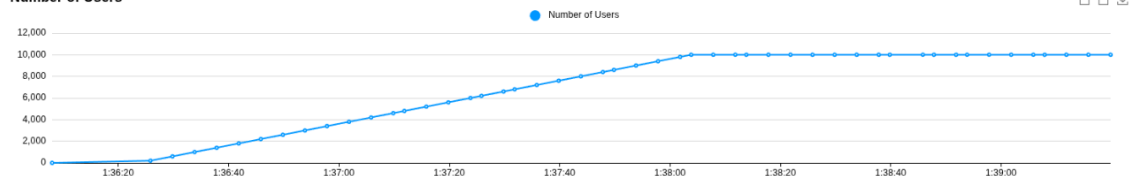
Total Requests per Second



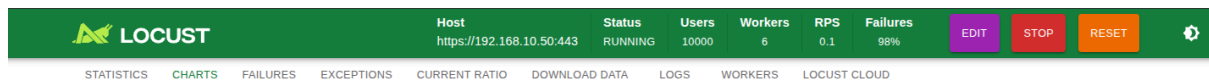
Response Times (ms)



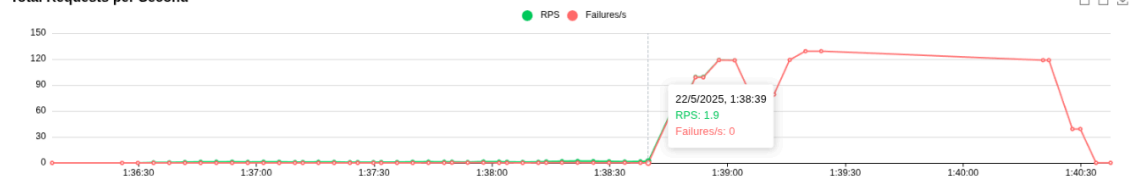
Number of Users



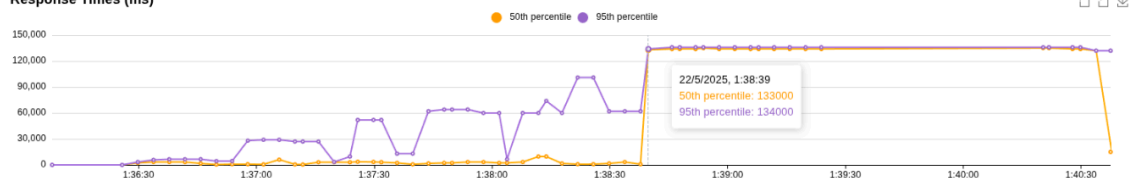
ABOUT



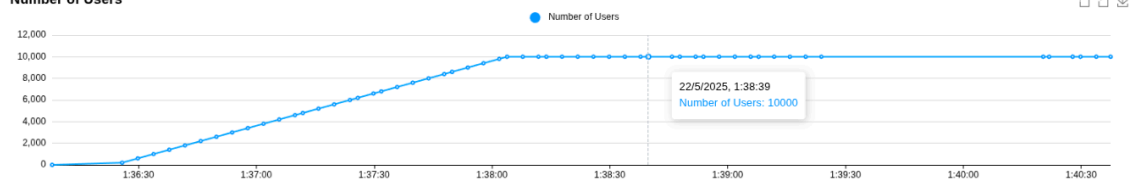
Total Requests per Second



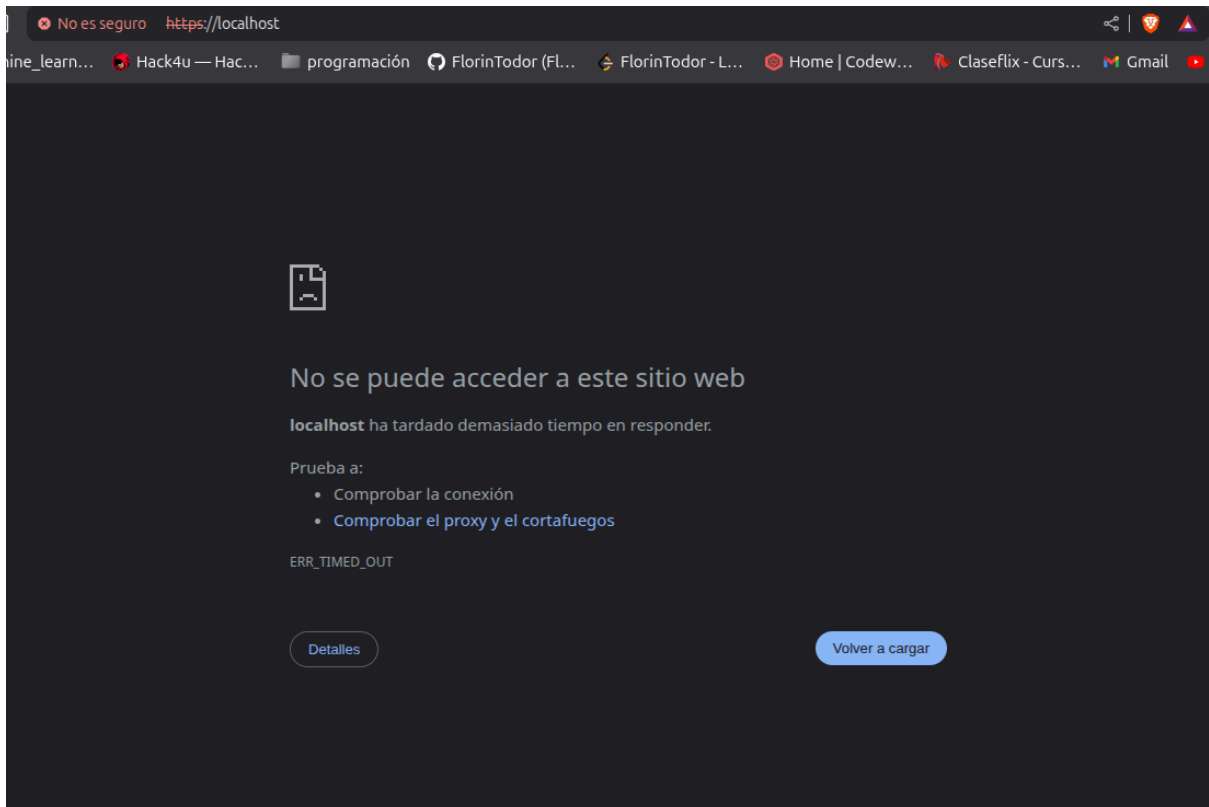
Response Times (ms)



Number of Users

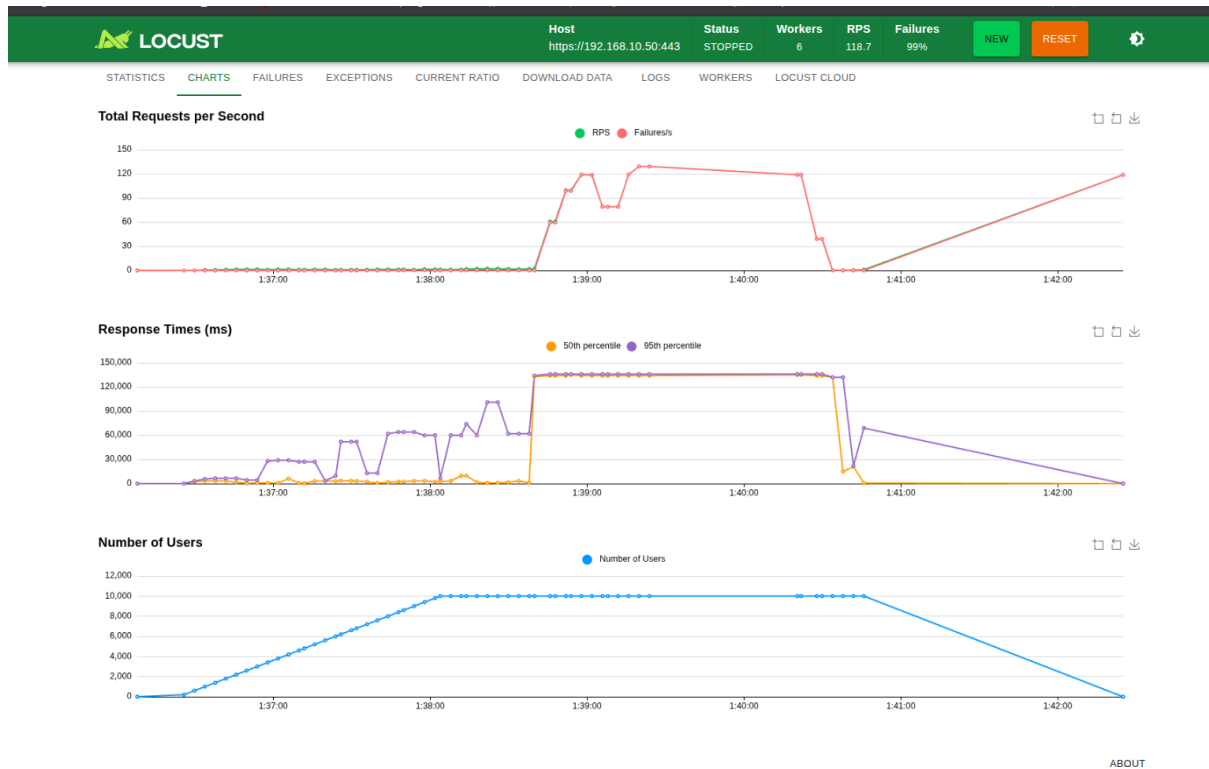


ABOUT



Por ello, aumenta de gran forma y de repente, el gráfico con los errores. Ahora cuando termine la prueba de carga comentaré los resultados y verificaré si se debe a las reglas del iptables.

Al finalizar la prueba de carga tenemos el siguiente resultado:



Los errores son los siguientes:

STATISTICS	CHARTS	FAILURES	EXCEPTIONS	CURRENT RATIO	DOWNLOAD DATA	LOGS	WORKERS	LOCUST CLOUD
# Failures	Method	Name	Message					
15	GET	/index.php	RemoteDisconnected('Remote end closed connection without response')					
12607	GET	/index.php	ConnectTimeoutError(urllib3.connection.HTTPSConnection object at 0x...: 'Connection to 192.168.10.50 timed out. (connect timeout=None)')					
1	GET	/index.php	SSLError(SSLEOFError(8, '[SSL: UNEXPECTED_EOF_WHILE_READING] EOF occurred in violation of protocol (_ssl.c:1010)'))					

STATISTICS															
CHARTS		FAILURES		EXCEPTIONS		CURRENT RATIO		DOWNLOAD DATA		LOGS		WORKERS		LOCUST CLOUD	
Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s			
GET	/index.php	12810	12623	134000	136000	136000	132473.55	2	215766	115.66	118.7	118.7			
	Aggregated	12810	12623	134000	136000	136000	132473.55	2	215766	115.66	118.7	118.7			

Podemos ver que la mayoría de los errores son peticiones que ni siquiera llegan a completar la conexión de TCP/SSL con el balanceador. Además de observar que casi todas las peticiones agotan el timeout global, dando lugar a mayor fallo. Cuando empiezan los 10.000 usuarios, todo se convierte en error o timeout. Dando lugar a que el servidor se caiga como he comentado anteriormente.

Como he comentado antes, como las pruebas las estoy haciendo en local y con la misma dirección ip (que no es una simulación real), estoy “cargandome” el servidor web debido a que tengo algunas reglas muy estrictas, que son:

```
## LIMITAR FRECUENCIA DE NUEVAS CONEXIONES POR IP (hashlimit) ##
iptables -A INPUT -p tcp --dport 80 -m conntrack --ctstate NEW -m hashlimit --hashlimit 3/sec --hashlimit-burst 5 --hashlimit-mode srcip --hashlimit-name http_limit -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -m conntrack --ctstate NEW -j DROP

iptables -A INPUT -p tcp --dport 443 -m conntrack --ctstate NEW -m hashlimit --hashlimit 3/sec --hashlimit-burst 5 --hashlimit-mode srcip --hashlimit-name https_limit -j ACCEPT
iptables -A INPUT -p tcp --dport 443 -m conntrack --ctstate NEW -j DROP
```

```
## CADENA DE RATE LIMITING ##
iptables -F RATE_HTTP 2>/dev/null || iptables -N RATE_HTTP

# Límite general 3 req/s con ráfaga de 3
iptables -A RATE_HTTP -m limit --limit 3/second --limit-burst 3 -j RETURN

# Protección específica por IP
for SRC in $(cat /dev/null); do
```

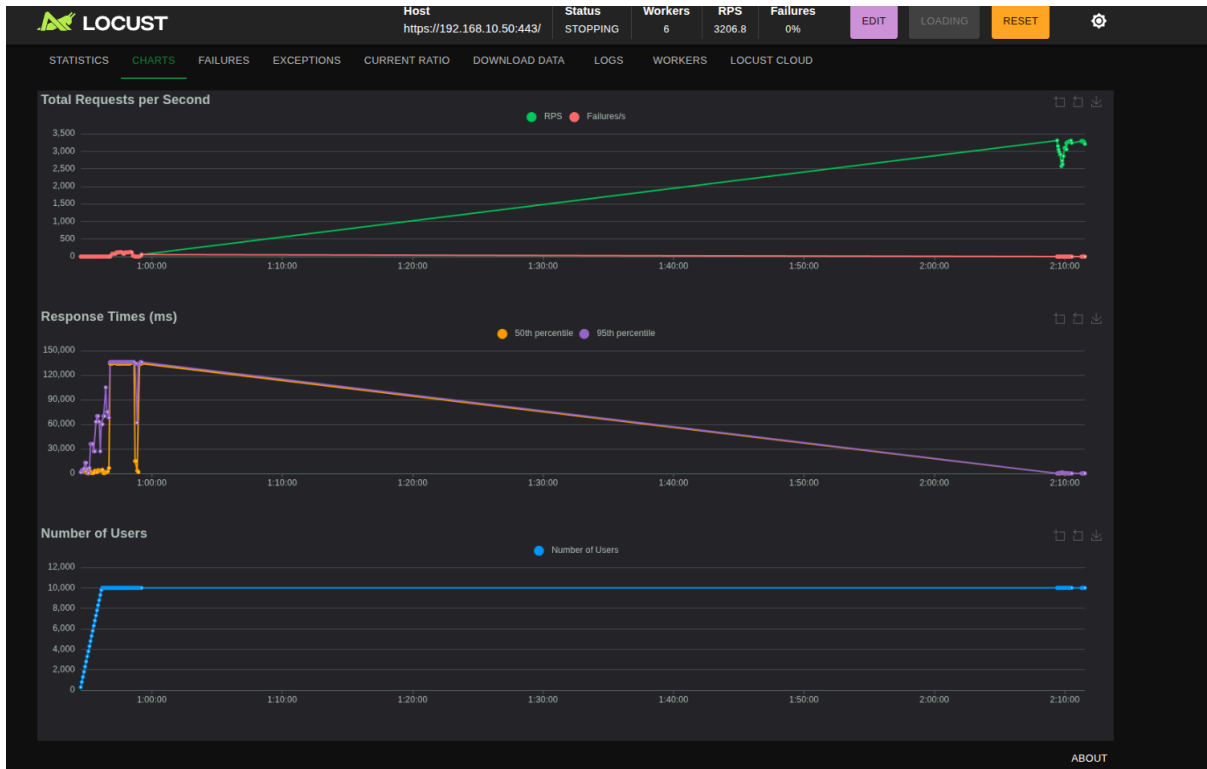
```
58 done
59
60 ### LÍMITE BÁSICO DE SYN FLOOD ###
61 iptables -A INPUT -p tcp --syn -m limit --limit 1/s --limit-burst 4 -j ACCEPT
62 iptables -A INPUT -p tcp --syn -j DROP
63
```

IMPORTANTE:

Es decir, estoy limitando las ráfagas de tráfico, 3 req/s. Todo lo que llegue después termina en DROP. Por otro lado, Máximo de 3 conexiones nuevas por IP y menos de 20 sockets simultáneos por IP. Además, de aceptar un solo SYN por segundo dando lugar a errores de timeout o no reset.

Por ello, supongo que en la prueba de AB básica ocurrirá lo mismo, por lo que voy a desactivar estas reglas específicas y volver a realizar una prueba de carga con Locust y ya después la básica de ab y al comprobar que todo funciona correctamente, probaré a hacer la parte de escalado automático.

Vale, **efectivamente se debía a las reglas de iptables**, ahora he realizado la carga de nuevo, no hay ningún problema en las webs. Sin embargo, ha llegado un momento en el que mi ordenador estaba sin poder reaccionar y me ha cerrado sesión de mi usuario.



EN las gráficas vemos justamente ese bajón en el que justo se me había cerrado sesión porque todo estaba pillado. Por lo demás vemos que ahora no tenemos ningún error.

La IA me comenta que se debe a que pedí más usuarios de lo que soporta mi ordenador (respecto a la carga y el uso de los hilos) ya que utilizan demasiada CPU.

Apache Benchmark

Ahora, voy a realizar la prueba básica de AB

```

● > ./init.sh -u nginx
[✓] Red red_servicios ya existe.
[✓] Red red_web ya existe.
[i] Estrategia de balanceo: round-robin (por defecto)
[!] nginx_balanceador ya está corriendo. Reiniciándolo para aplicar nueva estrategia...
[✓] nginx_balanceador reiniciado con la estrategia rb.
[i] Comprobando puertos 8080 a 8089...
[+] Running 16/16
  ✓ Container p5-granjaweb-worker-flotodor-2   Removed
  ✓ Container p5-granjaweb-master-flotodor-1   Removed
  ✓ Container p5-granjaweb-worker-flotodor-3   Removed
  ✓ Container p5-granjaweb-worker-flotodor-6   Removed
  ✓ Container p5-granjaweb-worker-flotodor-5   Removed
  ✓ Container p5-granjaweb-worker-flotodor-1   Removed
  ✓ Container p5-granjaweb-worker-flotodor-4   Removed
  ✓ Container web1                             Running
  ✓ Container web2                             Running
  ✓ Container web8                             Running
  ✓ Container web5                             Running
  ✓ Container web7                             Running
  ✓ Container web3                             Running
  ✓ Container web6                             Running
  ✓ Container web4                             Running
  ✓ Container nginx_balanceador                Started
[+] Servicios iniciados con Nginx.
● > ./init.sh -u ab
[✓] Red red_servicios ya existe.
[✓] Red red_web ya existe.
[i] Comprobando puertos 8080 a 8089...
WARN[0000] Found orphan containers ([nginx_balanceador web7 web3 web1 web2 web8 web5 web4 w
d with the --remove-orphans flag to clean it up.
[+] Running 1/1
  ✓ Container apache_benchmark-P5   Started
[+] Servicios iniciados con Apache Benchmark.

○ □ | □ ~/Escritorio/SWAP/P5-granjaweb | on □ □ main !16 ?2

```

```

100% 124 (longest request)
> docker logs apache_benchmark-P5
This is ApacheBench, Version 2.3 <$Revision: 1913912 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.10.50 (be patient)
Completed 1000 requests
Completed 2000 requests
Completed 3000 requests
Completed 4000 requests
Completed 5000 requests
Completed 6000 requests
Completed 7000 requests
Completed 8000 requests
Completed 9000 requests
Completed 10000 requests
Finished 10000 requests

Server Software:      nginx/1.27.5
Server Hostname:      192.168.10.50
Server Port:          443
SSL/TLS Protocol:     TLSv1.3,TLS_AES_256_GCM_SHA384,2048,256
Server Temp Key:      X25519 253 bits

Document Path:        /
Document Length:      7928 bytes

Concurrency Level:    100
Time taken for tests:  4.901 seconds
Complete requests:    10000
Failed requests:      4999
  (Connect: 0, Receive: 0, Length: 4999, Exceptions: 0)
Keep-Alive requests:  5001
Total transferred:    80850060 bytes
HTML transferred:     79230010 bytes
Requests per second:  2040.33 [#/sec] (mean)
Time per request:     49.012 [ms] (mean)
Time per request:     0.490 [ms] (mean, across all concurrent requests)
Transfer rate:        16109.49 [Kbytes/sec] received

```

```

Connection Times (ms)
      min  mean[+/-sd] median   max
Connect:    0    25  25.4    27    78
Processing:   3    24  10.8    23    61
Waiting:     1    15   5.4    16    41
Total:       5    49  23.8    50   124

Percentage of the requests served within a certain time (ms)
 50%    50
 66%    62
 75%    71
 80%    73
 90%    80
 95%    84
 98%    91
 99%    97
100%   124 (longest request)

```

```

~ | ~/Escritorio/SWAP/P5-granjaweb | on ~ main !16 ?2

```

Vemos que no hay ningún problema.

Escalado automático

He querido probar la escena de escalado automático para estas pruebas de carga, ya que me parecía interesante.

Para ello primero voy a ejecutar los contenedores con el balanceador del escalado automático de la práctica 2.

Para ello, voy a ejecutar `./init.sh` -u escalado. Posteriormente, veremos como se comporta con las pruebas de carga realizadas (realmente esto ya lo hice con la práctica 2, pero volveré a hacerlo). He decidido utilizar la parte del escalado automático porque siendo la última práctica me parece interesante poder usarlo y hacer un buen benchmark.

Para ello, voy a recordar que hice uso de node exporter, prometheus y grafana (esto sobretodo para hacerlo más visual). Además, uso del script que escala de forma automática la creación y eliminación de servidores webs hasta un máximo de 20 contenedores (si no mal recuerdo). Además, hice uso de haproxy como balanceador (para mostrar gráficamente la creación de las webs y si están funcionando correctamente).

Para ello, también tengo que volver a preparar grafana, pero solamente mostraré la salida de la gráfica.

Pero antes de realizar nada con el escalado automático, voy a realizar las pruebas básicas con nginx como balanceador y sin escalado.

Para realizar las pruebas con escalado automático también debo de activar en los iptables de los servidores webs la entrada para el puerto 9100 tanto para prometheus como para yo usarlo y verlo desde el navegador, esto es debido a que tengo que recopilar la información de node exporter para pasarlo al prometheus de dichos servidores. Esto lo haré nada más que para esta prueba, posteriormente lo reactivaré.

```
iptables -A INPUT -p tcp -s $BALANCER_IP --dport 443 -j ACCEPT  
  
# Permitir tráfico para el puerto 9100 desde la subred  
  
PROMETHEUS_IP="192.168.10.100"  
iptables -A INPUT -p tcp -s $PROMETHEUS_IP --dport 9100 -j ACCEPT  
  
iptables -A INPUT -p tcp -s 192.168.10.0/24 --dport 9100 -j ACCEPT
```



```

[i] Escalador iniciado. Esperando señal de parada...
[i] CPU Promedio: 47%
[i] CPU estable. No se escala ni desescala.
[i] CPU Promedio: 23%
[i] CPU estable. No se escala ni desescala.
[i] CPU Promedio: 5%
[i] CPU baja, pero ya están las 8 webs mínimas. No se desescala más.
[i] CPU Promedio: 4%
[i] CPU baja, pero ya están las 8 webs mínimas. No se desescala más.

=> => Writing image sha256:14eb1245/rfe/2f/2b59ced/b5f164026dd6aa8899dd8celd93779a4025cd65
=> => naming to docker.io/library/flotodor-ab-image:p5

View build details: docker-desktop://dashboard/build/default/default/grwsc12rv57ymjzevtld5npi
•> ./init.sh -u escalado
[✓] Red red servicios ya existe.
[✓] Red red_web ya existe.
[i] Estrategia de balanceo: round-robin (por defecto)
[i] Comprobando puertos 8080 a 8089...
[+] Running 12/12
  ✓ Container web4      Started
  ✓ Container prometheus Started
  ✓ Container node-exporter Started
  ✓ Container web3      Started
  ✓ Container web8      Started
  ✓ Container web7      Started
  ✓ Container web2      Started
  ✓ Container web6      Started
  ✓ Container web5      Started
  ✓ Container web1      Started
  ✓ Container grafana   Started
  ✓ Container haproxy_balanceador Started
[+] Servicios iniciados con monitorización y escalado automático.
[✓] Escalador iniciado en nueva terminal (gnome-terminal)

~ /Escritorio/SWAP/P5-granjaweb | on main !20 72

```

Vamos a realizar las pruebas de AB y Locust

Vale, el primero error que veo es que no tengo el uso de HTTPS, es decir, no lee el certificado SSL, voy a arreglarlo.

Tras solucionar los errores (modificar ficheros de configuración, añadir el certificado ssl y abrir el puerto 443), la prueba de ab:

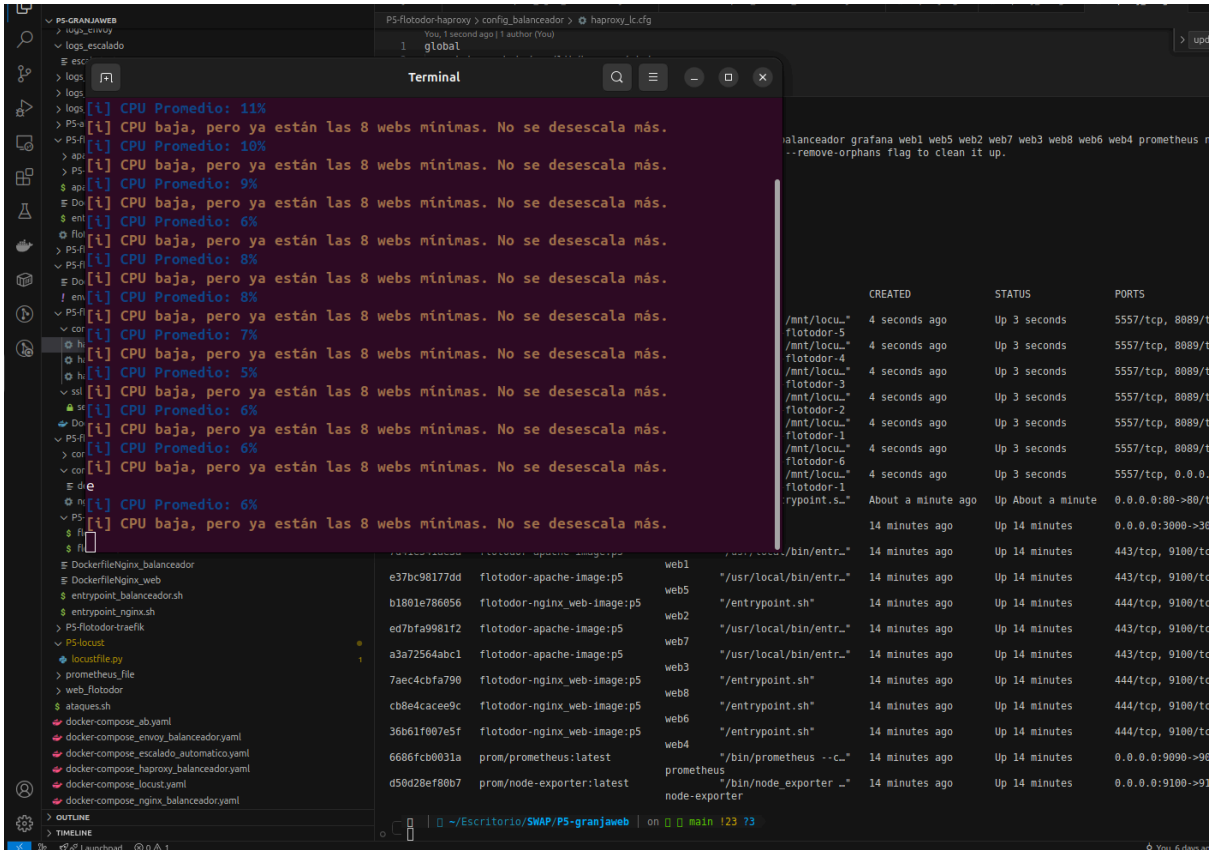
The screenshot shows the VS Code interface with the following components:

- EXPLORER:** A file tree on the left showing the project structure, including files like `haproxy.cfg`, `nginx.conf`, and `locustfile.py`.
- TERMINAL:** The main panel on the right displaying the output of the `haproxy -c` command. It shows that the configuration is valid and provides detailed statistics for 10,000 requests.

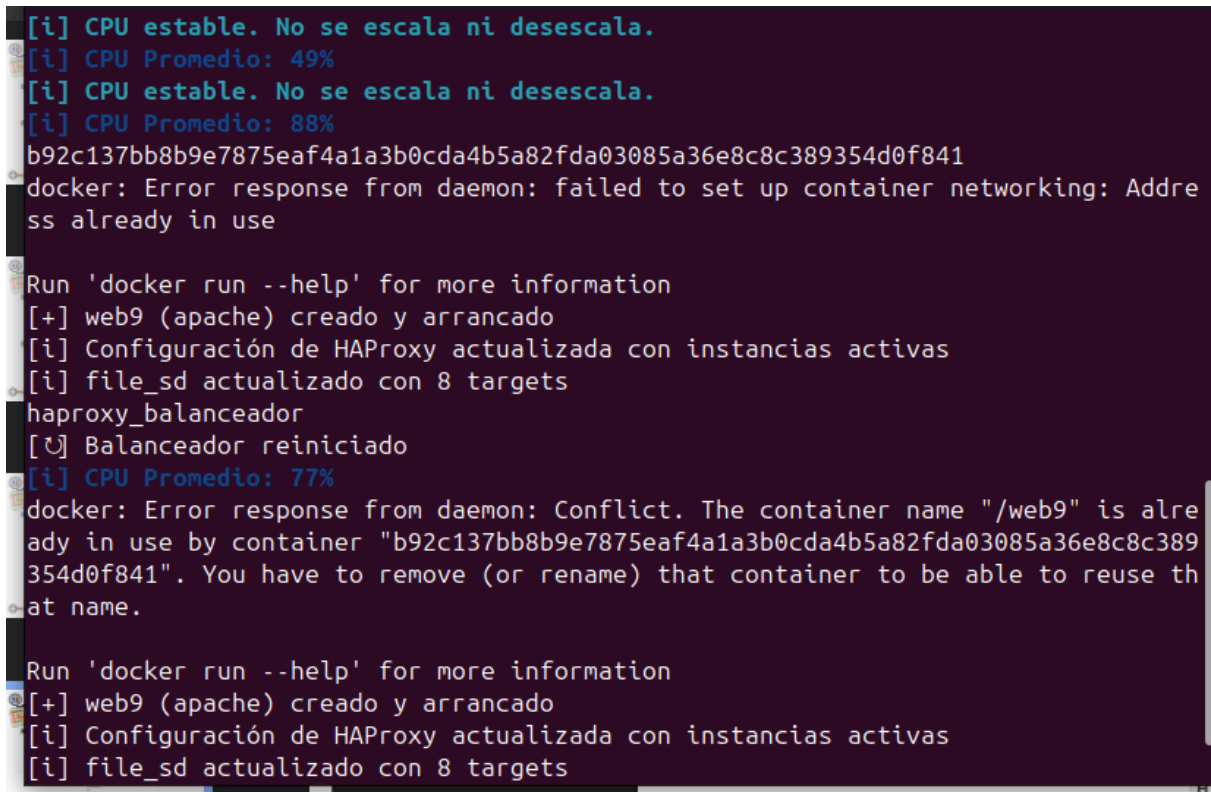
haproxy -c Output:

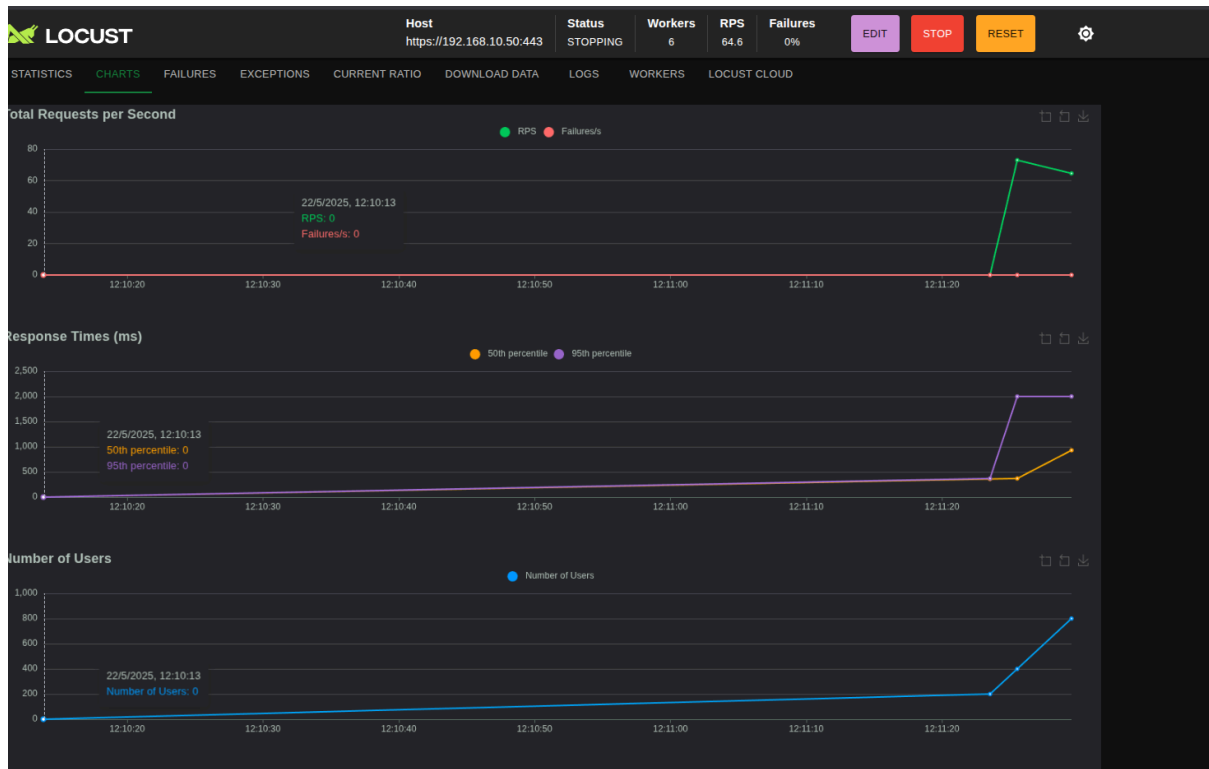
```

1 global
2   stats socket /var/lib/haproxy/stats
3
4 defaults
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
26
```



Vale, pues ha sido una decepción, se ha quedado pillado y se han reiniciado los servidores webs:





Vemos esta salida antes de que se produzca el error, los logs no nos muestran ningún error para poder revisar el por qué ha sucedido esto.

Vale, vemos que aunque sea una buena idea hacer esta prueba, es mejor no seguir con ello, simplemente por falta de tiempo (estamos a finales de mayo y hay alta carga de trabajo) además de que tendría que modificar varias partes de la práctica 2 para que no se solapen ips al crear nuevos contenedores (que realmente me valdría con cambiar el puerto para el locust), pero debo de seguir con esta práctica (igualmente, esto lo he realizado para mi, aunque en verano seguiré revisando).

B5: Análisis de resultados

Podemos analizar los resultados de varios contextos de pruebas realizadas.

Dentro de Locust, podemos ver los resultados con la configuración del firewall para bloquear DDoS y sin el bloqueo, respecto a Apache Benchmark podemos ver los resultados con la configuración sin el bloqueo y luego lo comentado en el Escalado automático.

Todas las pruebas realizadas sobre Locust, son con los siguientes parámetros:

LOCUST

Host: https://192.168.10.50:443 | Status: READY | Workers: 6 | RPS: 0 | Failures: 0%

Start new load test

Number of users (peak concurrency): 10000

Ramp up (users started/second): 100

Host: https://192.168.10.50:443

Advanced options

Run time (e.g. 20, 20s, 3m, 2h, 1h20m, 3h30m10s, etc.): 5m

Profile

START

- 10.000 usuarios en concurrencia
- Uso de 100 para ramp up
- Prueba de carga realizada durante 5 minutos

LOCUST

Host: https://192.168.10.50:443 | Status: STOPPED | Workers: 6 | RPS: 2038.9 | Failures: 0%

NEW RESET

STATISTICS CHARTS FAILURES EXCEPTIONS CURRENT RATIO DOWNLOAD DATA LOGS WORKERS LOCUST CLOUD

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/index.php	705432	2658	53	1900	2700	495.99	1	26899	7893.15	2038.9	34.7
Aggregated		705432	2658	53	1900	2700	495.99	1	26899	7893.15	2038.9	34.7

LOCUST

Host: https://192.168.10.50:443 | Status: STOPPED | Workers: 6 | RPS: 2038.9 | Failures: 0%

NEW RESET

STATISTICS CHARTS FAILURES EXCEPTIONS CURRENT RATIO DOWNLOAD DATA LOGS WORKERS LOCUST CLOUD

# Failures	Method	Name	Message
2658	GET	/index.php	RemoteDisconnected('Remote end closed connection without response')

Podemos observar que se comenta que hay fallos, sin embargo, en el gráfico no se muestra nada. Igualmente, en las pruebas realizadas en el [apartado anterior](#), no se muestra este error, pero igualmente vamos a comentar estos nuevos resultados.

Podemos ver:

- Número de peticiones total más de 700.000
- Fallos 2658
- Mediana de respuesta 53ms, por debajo del segundo
- Errores debidos a REmoteDisconnected, realmente el servidor no ha caído en ningún momento por lo que no comprendo el error.
- Vemos que se ha utilizado 6 workers (aunque se ha actualizado el pdf para usar con 5)
- RPS alcanzado 2038.9

Recordar que esta comprobación es sin el bloqueo de ataques DDoS comentados en el apartado anterior. Debido a que con el bloqueo realmente sale un 99% de error debido a lo que comenté por el uso de la misma ip.

Respecto a **Apache Benchmark**

```

> docker logs apache_benchmark-P5
This is ApacheBench, Version 2.3 <$Revision: 1913912 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.10.50 (be patient)
Completed 1000 requests
Completed 2000 requests
Completed 3000 requests
Completed 4000 requests
Completed 5000 requests
Completed 6000 requests
Completed 7000 requests
Completed 8000 requests
Completed 9000 requests
Completed 10000 requests
Finished 10000 requests


Server Software:      nginx/1.27.5
Server Hostname:      192.168.10.50
Server Port:          443
SSL/TLS Protocol:     TLSv1.3,TLS_AES_256_GCM_SHA384,2048,256
Server Temp Key:      X25519 253 bits


Document Path:        /
Document Length:      7918 bytes


Concurrency Level:     100
Time taken for tests:   4.705 seconds
Complete requests:     10000
Failed requests:        4997
    (Connect: 0, Receive: 0, Length: 4997, Exceptions: 0)
Keep-Alive requests:   4997
Total transferred:     80849820 bytes
HTML transferred:      79229970 bytes
Requests per second:   2125.30 [#/sec] (mean)
Time per request:      47.052 [ms] (mean)
Time per request:      0.471 [ms] (mean, across all concurrent requests)
Transfer rate:         16780.25 [Kbytes/sec] received


Connection Times (ms)
      min   mean[+/-sd] median   max
Connect:    0    24  24.2     24    84
Processing:  2    23  10.4     21    60
Waiting:    2    14   5.2     15    39
Total:      5    46  22.7     48   128


Percentage of the requests served within a certain time (ms)
 50%    48
 66%    59
 75%    66
 80%    69
 90%    77
 95%    81
 98%    85
 99%    94
100%   128 (longest request)

```

- Peticiones fallidas 4997
- Peticiones completadas exitosamente 5003

Igualmente los errores que se muestran es por longitud inesperada o vacía

```

Time taken for tests: 4.703 seconds
Complete requests: 10000
Failed requests: 4997
 (Connect: 0, Receive: 0, Length: 4997, Exceptions: 0)
Keep-Alive requests: 4997

```

Por lo tanto, no es por la propia conexión con el servidor.

Tareas Avanzadas

Para esta parte, al no poder haber asistido a clase he tenido que hacer más uso de la IA, sobretodo para entender la parte de CMS.

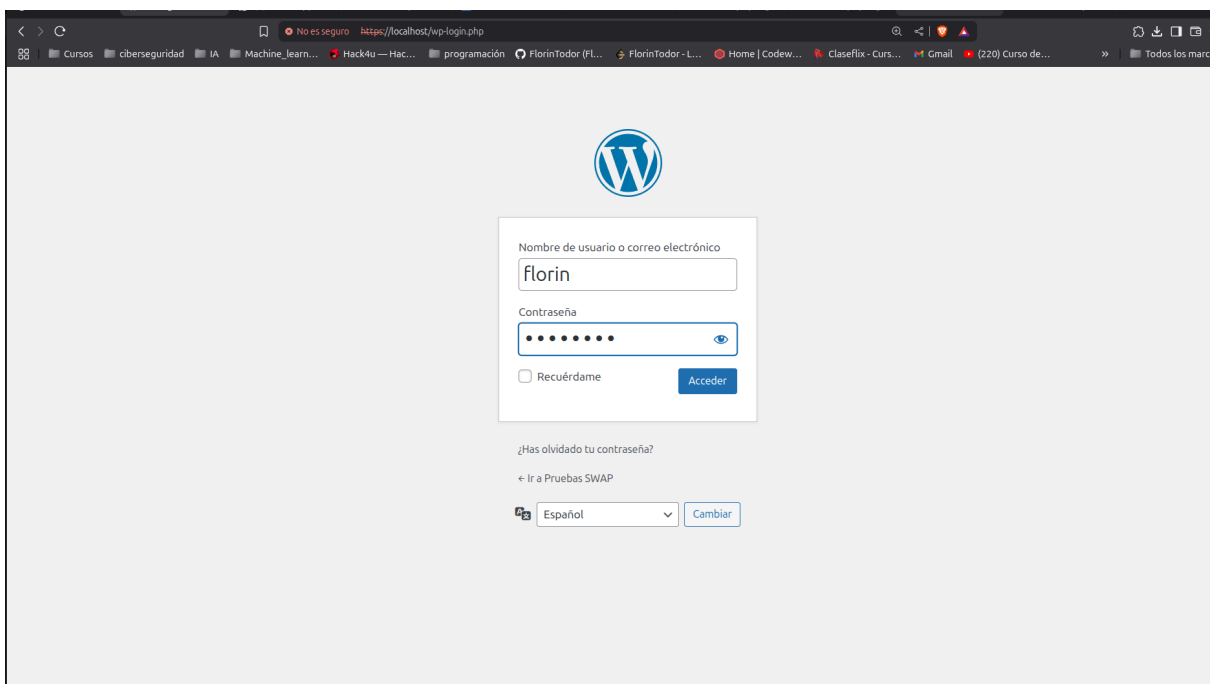
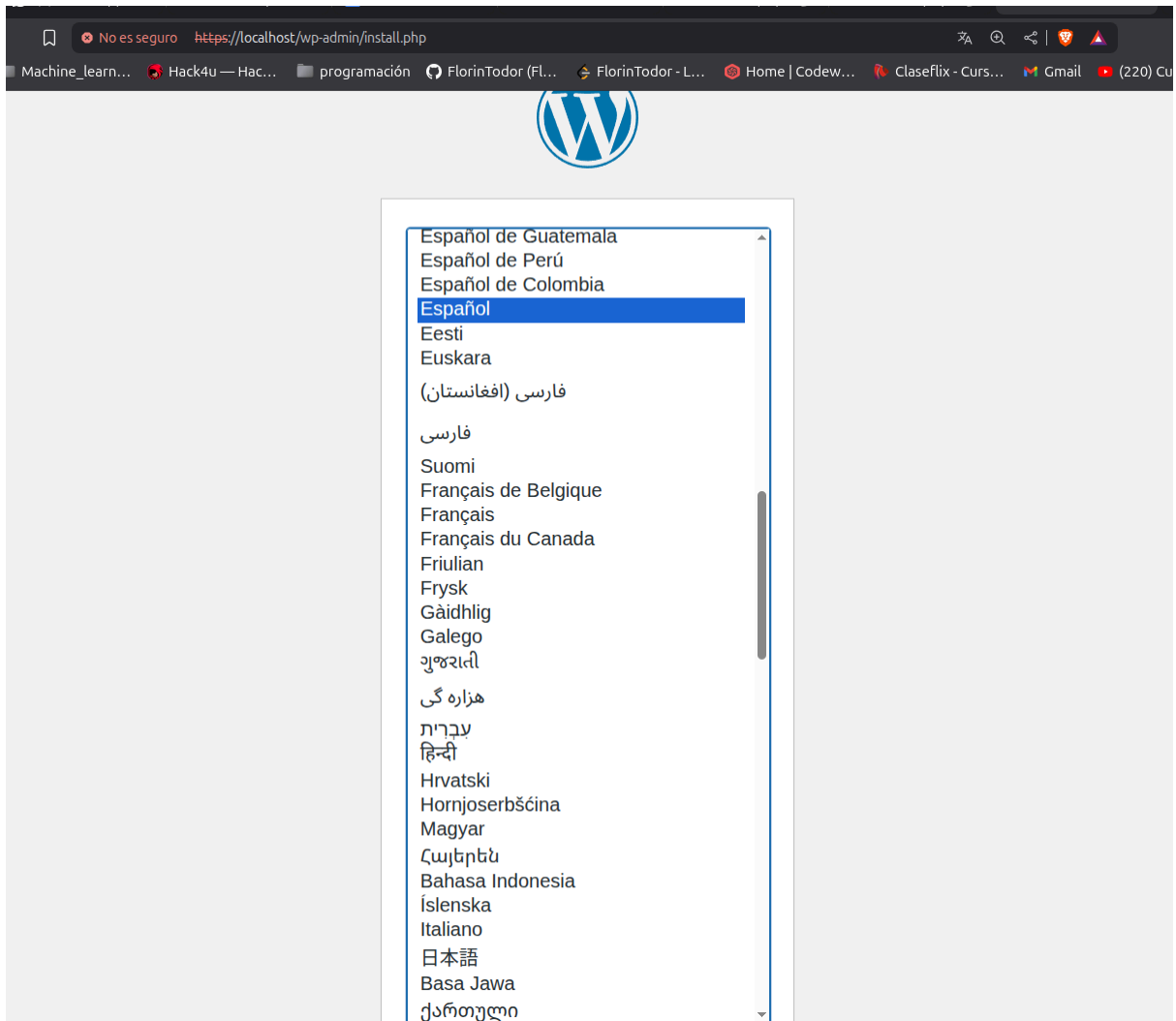
Por otro lado, comentar que voy a utilizar HAproxy como balanceador de carga en vez de Nginx como hasta ahora (sobretodo porque es más visual los logs).

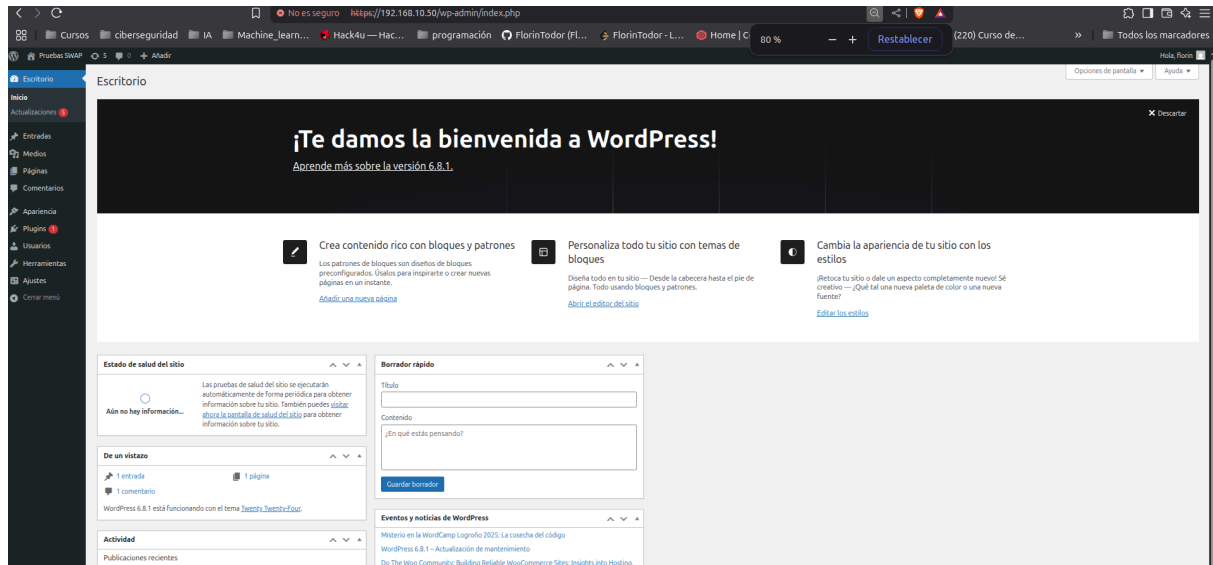
Antes de comenzar a explicar las 3 tareas avanzadas voy a mostrar la creación del CMS de wordpress

The screenshot shows the HAProxy statistics page with the following sections:

- rotator**: A table with columns for Queue, Session rate, Sessions, Bytes, Denied, Errors, Warnings, Status, LastChk, Wght, Act, Bck, Chk, Dwn, Downtime, and Thrft. It shows Frontend and Backend statistics.
- backend_static**: A table with columns for Queue, Session rate, Sessions, Bytes, Denied, Errors, Warnings, Status, LastChk, Wght, Act, Bck, Chk, Dwn, Downtime, and Thrft. It shows statistics for web1 through web8 and Backend.
- backend_cms**: A table with columns for Queue, Session rate, Sessions, Bytes, Denied, Errors, Warnings, Status, LastChk, Wght, Act, Bck, Chk, Dwn, Downtime, and Thrft. It shows statistics for wp1 and Backend.
- stats**: A table with columns for Queue, Session rate, Sessions, Bytes, Denied, Errors, Warnings, Status, LastChk, Wght, Act, Bck, Chk, Dwn, Downtime, and Thrft. It shows Frontend and Backend statistics.

Posteriormente, tenemos la instalación del propio Wordpress, Usuario florin, contraseña SWAP1234





A1: Desarrollar tareas avanzadas en [Locustfile.py](#)

```

You, 2 hours ago | 1 author (You)
1 # PS-locust/locustfile.py - navegación anónima + login admin + login repetido
2 import os, random, time
3 from locust import HttpUser, task, between, SequentialTaskSet
4
5 WP_USER = os.getenv("WP_USER", "florin")
6 WP_PASSWORD = os.getenv("WP_PASSWORD", "SWAP1234")
7
8 # ----- FLUJO A: navegación anónima -----
9 | You, 2 hours ago + Uncommitted changes
10 | You, 2 hours ago | 1 author (You)
11 class PublicBrowsing(SequentialTaskSet):
12     @task(3)
13     def home(self):
14         self.client.get("/", name="Anon - Home")
15
16     @task(2)
17     def post_detail(self):
18         post_id = random.randint(1, 20)
19         self.client.get(f"/?p={post_id}", name="Anon - Post")
20
21     @task(2)
22     def search(self):
23         q = random.choice(["locust", "wordpress", "swap", "flotodor"])
24         self.client.get(f"/?s={q}", name="Anon - Search")
25
26 # ----- FLUJO B: login y navegación por wp-admin -----
27 | You, 2 hours ago | 1 author (You)
28 class AdminNavigation(SequentialTaskSet):
29     def on_start(self):
30         """Login al iniciar el usuario."""
31         payload = {
32             "log": WP_USER,
33             "pwd": WP_PASSWORD,
34             "wp-submit": "Log In",
35             "redirect_to": "/wp-admin/",
36             "testcookie": "1"
37         }
38         with self.client.post("/wp-login.php", data=payload, name="Login", catch_response=True, allow_redirects=False) as r:
39             if r.status_code in (301, 302) or "Dashboard" in r.text:
40                 r.success()
41             else:
42                 r.failure("login-failed")
43
44     @task(2)
45     def dashboard(self):
46         self.client.get("/wp-admin/index.php", name="Admin - Dashboard")
47
48     @task(1)
49     def posts_list(self):
50         self.client.get("/wp-admin/edit.php", name="Admin - Posts List")
51
52     @task(1)
53     def new_post(self):
54         self.client.get("/wp-admin/post-new.php", name="Admin - New Post")
55
56     @task(1)
57     def profile(self):
58         self.client.get("/wp-admin/profile.php", name="Admin - Profile")
59
60

```

```

self.client.get('/wp-admin/profile.php', name='Admin - Profile')

# ----- FLUJO C: login clásico repetido -----

You, 2 hours ago | 1 author (You)
class LoginStorm(SequentialTaskSet):
    @task
    def login_attempt(self):
        payload = {
            "log": WP_USER,
            "pwd": WP_PASSWORD,
            "wp-submit": "Log In",
            "redirect_to": "/wp-admin/",
            "testcookie": "1"
        }
        with self.client.post("/wp-login.php", data=payload, name="Login", catch_response=True, allow_redirects=False) as r:
            if r.status_code in (301, 302) or "Dashboard" in r.text:
                r.success()
                time.sleep(random.uniform(0.5, 1.5)) # más suave
            else:
                r.failure("login-failed")
                time.sleep(random.uniform(1, 3))

# ----- USUARIOS LOCUST -----

You, 2 hours ago | 1 author (You)
class AnonUser(HttpUser):
    tasks = [PublicBrowsing]
    wait_time = between(1, 3)
    host = "https://192.168.10.50"
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.client.verify = False

You, 2 hours ago | 1 author (You)
class LoginUser(HttpUser):
    tasks = [LoginStorm]
    wait_time = between(4, 10)
    host = "https://192.168.10.50"
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.client.verify = False

You, 2 hours ago | 1 author (You)
class AdminUser(HttpUser):
    tasks = [AdminNavigation]
    wait_time = between(3, 6)
    host = "https://192.168.10.50"
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.client.verify = False

```

Como podemos ver, se divide en tres partes.

1. Navegación pública anónima, con uso de la clase de PublicBrowsing, que representa el comportamiento de un visitante no autenticado, como visitar la página oficial, post individuales, realizar búsquedas.
2. Navegación autenticada en el panel de admin, con uso de la clase AdminNavigation, que representa el flujo de trabajo de un usuario autenticado en Wordpress, con tareas como el propio login, acceso al panel, listar entradas, redactar nueva entrada, ver perfil del usuario.
3. Un pequeño ataque de login repetido con fuerza bruta suave, con uso de la clase LoginStorm, que simula accesos repetidos a la página de login.

A2. Crear escenario multicontenedor con algún CMS

Para esta parte, como he comentado anteriormente, he usado Wordpress como CMS. He reutilizado el docker compose del balanceador de carga de HAproxy realizado en las prácticas anteriores.

```

You, 2 hours ago | 1 author (You)

# Datos comunes para todos los servicios de apache
x-common-apache-config: &common-apache-config
image: flotodor-apache-image:p5
restart: always
volumes:
  - ./web_flotodor:/var/www/html
  - ./logs_apache:/var/log/apache2
  - ./P5-flotodor-certificados:/etc/apache2/ssl

# Datos comunes para todos los servicios de nginx
x-common-nginx_web-config: &common-nginx_web-config
image: flotodor-nginx_web-image:p5
restart: always
volumes:
  - ./web_flotodor:/usr/share/nginx/html:ro
  - ./logs_nginx:/var/log/nginx
  - ./P5-flotodor-certificados:/etc/nginx/ssl

x-common-haproxy_balanceador-config: &common-haproxy_balanceador-config
image: flotodor-haproxy_balanceador-image:p5
restart: always
volumes:
  - ./logs_haproxy:/var/log/haproxy

x-common-wordpress-base: &wordpress_base
image: wordpress:6.5-php8.3-apache
restart: always
environment:
  WORDPRESS_DB_HOST: wp_db:3306
  WORDPRESS_DB_NAME: wp
  WORDPRESS_DB_USER: wpuser
  WORDPRESS_DB_PASSWORD: wppw
  WORDPRESS_CONFIG_EXTRA: "define('WP_HOME','https://192.168.10.50');define('WP_SITEURL','https://192.168.10.50');"
volumes:
  - wp_shared:/var/www/html/wp-content

#####

### TENGO QUE BUSCAR ALGUNA FORMA PARA QUE NO SE REPITA TANTO CODIGO ###

```

```

147
148   > Run Service
149   haproxy_balanceador:
150     <<: *common-haproxy_balanceador-config
151     container_name: haproxy_balanceador
152     environment:
153       - SERVER_NAME=haproxy_balanceador
154     networks:
155       red_web:
156         ipv4_address: 192.168.10.50
157     ports:
158       - "80:80"
159       - "443:443"
160       - "9000:9000"
161     depends_on:
162       - web1
163       - web2
164       - web3
165       - web4
166       - web5
167       - web6
168       - web7
169       - web8
170
171   ## --- base de datos -----
172   > Run Service
173   wp_db:
174     image: mariadb:11
175     restart: always
176     environment:
177       MYSQL_ROOT_PASSWORD: rootpw
178       MYSQL_DATABASE: wp
179       MYSQL_USER: wpuser
180       MYSQL_PASSWORD: wppw
181     volumes:
182       - wp_db_data:/var/lib/mysql
183     networks:
184       red_servicios:
185         ipv4_address: 192.168.20.10 # solo back-end
186
187   ## --- WordPress #1 -----
188   > Run Service
189   wordpress1:
190     <<: *wordpress_base
191     container_name: wordpress1
192     networks:
193       red_web:
194         ipv4_address: 192.168.10.60
195       red_servicios:
196         ipv4_address: 192.168.20.11
197
198   volumes:
199     wp_db_data:
200     wp_shared:
201
202   networks:
203     red_web:
204       external: true
205     red_servicios:
206       external: true

```

No he mostrado la parte de las webs estáticas porque no es relevante (es lo mismo que desde la práctica 1 con los servidores webs de nginx y apache).

Como vemos, tanto Wordpress como MariaDB están conectados a la red_servicios.

Para la parte de asegurar la configuración para dirigir el tráfico hacia las instancias del CMS sería la siguiente configuración del CMS en HAproxy:

```
#####
# 1. GLOBAL & DEFAULTS
#####
global
    log stdout format raw daemon
    stats socket /var/lib/haproxy/stats
    tune.ssl.default-dh-param 2048          # evita warnings DHE

defaults
    mode                http
    option               httplog
    option               forwardfor except 127.0.0.0/8  # añade X-Forwarded-For
    option               http-server-close             # 1 conexión por req. al back-end
    timeout connect      5s
    timeout client        50s
    timeout server        50s
    maxconn              2000

#####
# 2. FRONTEND
#####
frontend flotorador
    bind *:80
    bind *:443 ssl crt /etc/ssl/haproxy/servidor.pem alpn h2,http/1.1

    http-request redirect scheme https code 308 unless { ssl_fc }
    http-request set-header X-Forwarded-Proto https if { ssl_fc }
    http-request set-header Host %[req.hdr(Host)]

    acl is_cms path_beg -i /wp-admin /wp-login.php /wp-json /blog /index.php /
    use_backend backend_cms if is_cms
    default_backend backend_static          # resto          → webs estáticas

#####
# 3. BACK-END 'ESTÁTICO' (tal como lo tenías)| You, 6 hours ago • A finished
#####
backend backend_static
    balance roundrobin
    option httpchk GET /index.php
    server web1 192.168.10.2:80 check maxconn 32
    server web2 192.168.10.3:80 check maxconn 32
    server web3 192.168.10.4:80 check maxconn 32
    server web4 192.168.10.5:80 check maxconn 32
    server web5 192.168.10.6:80 check maxconn 32
    server web6 192.168.10.7:80 check maxconn 32
    server web7 192.168.10.8:80 check maxconn 32
    server web8 192.168.10.9:80 check maxconn 32
```



```

64 #####
65 # 3. BACK-END 'ESTÁTICO' (tal como lo tenías)| You, 6 hours ago • A fi
66 #####
67 backend backend_static
68     balance roundrobin
69     option httpchk GET /index.php
70     server web1 192.168.10.2:80 check maxconn 32
71     server web2 192.168.10.3:80 check maxconn 32
72     server web3 192.168.10.4:80 check maxconn 32
73     server web4 192.168.10.5:80 check maxconn 32
74     server web5 192.168.10.6:80 check maxconn 32
75     server web6 192.168.10.7:80 check maxconn 32
76     server web7 192.168.10.8:80 check maxconn 32
77     server web8 192.168.10.9:80 check maxconn 32
78
79 #####
80 # 4. BACK-END 'CMS' (## ➤ NUEVO)
81 #####
82 backend backend_cms
83     cookie WPID insert indirect nocache
84     option httpchk GET /wp-login.php
85     http-check expect status 200 302
86     timeout check 5s
87     server wp1 192.168.10.60:80 cookie s1 check inter 3s rise 3 fall 2
88
89 #####
90 # 5. STATS UI (como ya tenías)
91 #####
92 listen stats
93     bind *:9000
94     mode http
95     stats enable
96     stats uri /estadisticas_flotodor
97     stats realm HAProxy\ Statistics
98     stats auth flotodor:SWAP1234
99

```

A3. Ejecución y Análisis de cargas de prueba avanzadas sobre CMS

Primero, ejecutamos todos los contenedores:

```

4CE226314075
[ ] [ ] ~/Escritorio/SWAP/P5-granjaweb | on [ ] [ ] main !4 ?1
• > ./init.sh -u haproxy
[✓] Red red_servicios ya existe.
[✓] Red red_web ya existe.
[i] Estrategia de balanceo: round-robin (por defecto)
[i] Comprobando puertos ...
[+] Running 11/11
✓ Container p5-granjaweb-wp_db-1 Started
✓ Container wordpress1 Started
✓ Container web4 Started
✓ Container web3 Started
✓ Container web2 Started
✓ Container web6 Started
✓ Container web5 Started
✓ Container web8 Started
✓ Container web7 Started
✓ Container web1 Started
✓ Container haproxy balanceador Started
[+] Servicios iniciados con HAProxy.
• > ./init.sh -u locust
[✓] Red red_servicios ya existe.
[✓] Red red_web ya existe.
[i] Comprobando puertos ...
WARN[0000] Found orphan containers ([haproxy_balanceador web5 web1 web2
up.
[+] Running 7/7
✓ Container p5-granjaweb-master-flotodor-1 Started
✓ Container p5-granjaweb-worker-flotodor-6 Started
✓ Container p5-granjaweb-worker-flotodor-2 Started
✓ Container p5-granjaweb-worker-flotodor-1 Started
✓ Container p5-granjaweb-worker-flotodor-4 Started
✓ Container p5-granjaweb-worker-flotodor-5 Started
✓ Container p5-granjaweb-worker-flotodor-3 Started
[+] Clúster Locust (1 master, 6 workers) en marcha.
[ ] [ ] ~/Escritorio/SWAP/P5-granjaweb | on [ ] [ ] main !4 ?1

```

```

✓ Container p5-granjaweb-worker-flotodor-4 Started
✓ Container p5-granjaweb-worker-flotodor-4 Started
✓ Container p5-granjaweb-worker-flotodor-5 Started
✓ Container p5-granjaweb-worker-flotodor-3 Started
[+] Clúster Locust (1 master, 6 workers) en marcha.
• > docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED        STATUS        PORTS                               NAMES
c51c18e9e8ba        locustio/locust:latest "locust -f /mnt/locu..." 25 seconds ago Up 24 seconds 5557/tcp, 8089/tcp                p5-granjaweb-worker-flotodor-5
654b8ba773de        locustio/locust:latest "locust -f /mnt/locu..." 25 seconds ago Up 24 seconds 5557/tcp, 8089/tcp                p5-granjaweb-worker-flotodor-3
b2947f028fc1        locustio/locust:latest "locust -f /mnt/locu..." 25 seconds ago Up 24 seconds 5557/tcp, 8089/tcp                p5-granjaweb-worker-flotodor-2
3a435a2e48e        locustio/locust:latest "locust -f /mnt/locu..." 25 seconds ago Up 24 seconds 5557/tcp, 8089/tcp                p5-granjaweb-worker-flotodor-1
b34588885177        locustio/locust:latest "locust -f /mnt/locu..." 25 seconds ago Up 23 seconds 5557/tcp, 8089/tcp                p5-granjaweb-worker-flotodor-4
94982291c8b9        locustio/locust:latest "locust -f /mnt/locu..." 25 seconds ago Up 23 seconds 5557/tcp, 8089/tcp                p5-granjaweb-worker-flotodor-6
9f6f526a75d4        locustio/locust:latest "locust -f /mnt/locu..." 25 seconds ago Up 24 seconds 5557/tcp, 0.0.0.0:8089->8089/tcp, [::]:8089->8089/tcp  p5-granjaweb-master-flotodor-1
194d13070a57        flotodor-haproxy-ba... "docker-entrypoint.s..." 14 minutes ago Up 29 seconds 0.0.0.0:80->80/tcp, [::]:80->80/tcp, 0.0.0.0:443->443/tcp, [::]:443->443/tcp, 0.0.0.0:9000->9000/tcp, [::]:9000->9000/tcp  haproxy_balanceador
2892fcd4277c        flotodor-apache-ima... "/usr/local/bin/entr..." 17 minutes ago Up 29 seconds 443/tcp, 9100/tcp, 0.0.0.0:8085->80/tcp, [::]:8085->80/tcp  web5
0892742494e        flotodor-apache-ima... "/usr/local/bin/entr..." 17 minutes ago Up 29 seconds 443/tcp, 9100/tcp, 0.0.0.0:8081->80/tcp, [::]:8081->80/tcp  web1
1112662b2b87        flotodor-nginx-web-... "/entrypoint.sh"        17 minutes ago Up 29 seconds 444/tcp, 9100/tcp, 0.0.0.0:8082->80/tcp, [::]:8082->80/tcp  web2
1f9c6d7f9e9        flotodor-nginx-web-... "/entrypoint.sh"        17 minutes ago Up 29 seconds 444/tcp, 9100/tcp, 0.0.0.0:8088->80/tcp, [::]:8088->80/tcp  web8
a5217631bdc        flotodor-apache-ima... "/usr/local/bin/entr..." 17 minutes ago Up 29 seconds 443/tcp, 9100/tcp, 0.0.0.0:8087->80/tcp, [::]:8087->80/tcp  web7
36e44f9913        flotodor-apache-ima... "/usr/local/bin/entr..." 17 minutes ago Up 29 seconds 443/tcp, 9100/tcp, 0.0.0.0:8083->80/tcp, [::]:8083->80/tcp  web3
ff364c79566        flotodor-nginx-web-... "/entrypoint.sh"        17 minutes ago Up 29 seconds 444/tcp, 9100/tcp, 0.0.0.0:8086->80/tcp, [::]:8086->80/tcp  web6
5858a844d74        flotodor-nginx-web-... "/entrypoint.sh"        17 minutes ago Up 29 seconds 444/tcp, 9100/tcp, 0.0.0.0:8084->80/tcp, [::]:8084->80/tcp  web4
894f92961cb        wordpress:6.5-php8... "docker-entrypoint.s..." 22 minutes ago Up 29 seconds 80/tcp                        wordpress1
5af6d88954d        mariadb:11          "docker-entrypoint.s..." 22 minutes ago Up 29 seconds 3306/tcp
[ ] [ ] ~/Escritorio/SWAP/P5-granjaweb | on [ ] [ ] main !4 ?1

```

Pruebas realizadas:

Start new load test

Number of users (peak concurrency) *

30

Ramp up (users started/second) *

5

Host

https://192.168.10.50:443

Advanced options

Run time (e.g. 20, 20s, 3m, 2h, 1h20m, 3h30m10s, etc.)

2m

Profile

LOCUST

Host

https://192.168.10.50:443

Status

STOPPED

Workers

6

RPS

8.4

Failures

31%

NEW

RESET

STATISTICS

CHARTS

FAILURES

EXCEPTIONS

CURRENT RATIO


DOWNLOAD DATA

LOGS

WORKERS

LOCUST CLOUD

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s	
GET	Admin - Dashboard	60	0	33	47	48	35.32	26	48	1460	1.8	0	<div></div>
GET	Admin - New Post	21	0	33	47	47	35.77	27	47	1463	0	0	
GET	Admin - Posts List	26	0	33	46	48	36.52	28	48	1459	0.2	0	
GET	Admin - Profile	20	0	33	45	45	34.29	28	45	1462	0.1	0	
GET	Anon - Home	130	0	43	63	67	44.11	37	82	84988	1.8	0	
GET	Anon - Post	81	73	33	42	48	33.26	26	48	57998.75	1.9	1.8	
GET	Anon - Search	78	0	35	42	45	35.55	29	45	64546.03	1.5	0	
POST	Login	78	78	19	38	46	22.42	13	46	7165.72	1.1	1.1	
	Aggregated	494	151	35	47	65	35.34	13	82	43573.59	8.4	2.9	

 LOCUST

Host

https://192.168.10.50:443

Status

STOPPED

Workers

6

RPS


8.4

Failures

31%

NEW

RESET



STATISTICS

CHARTS

FAILURES

EXCEPTIONS

CURRENT RATIO

DOWNLOAD DATA

LOGS

WORKERS

LOCUST CLOUD

Failures

Method

Name

Message

73

GET

Anon - Post

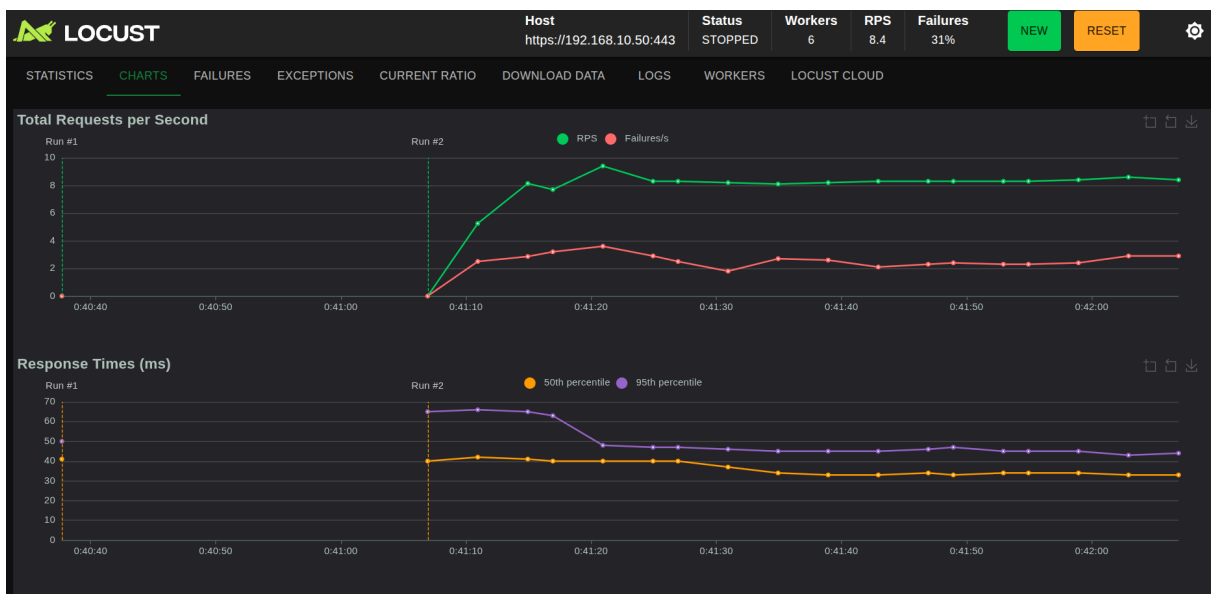
HTTPError(404 Client Error: Not Found for url: Anon - Post)

78

POST

Login

CatchResponseError('login-failed')



Por la parte de las tareas administrativas vemos que no hay ningún problema, sin embargo, vemos problemas en el login, en el post se debe a que se solicitan ids entre 1-20 y muchos de ellos no existen en la base de datos, por lo que devuelve el error de 404. Igualmente, se estabiliza después del “boom” de errores.

Siguiente prueba:
100, 10 y 3m.

Edit running load test

Number of users (peak concurrency) *

100

Ramp up (users started/second) *

10

UPDATE

LOCUST

Host

https://192.168.10.50:443

Status

STOPPED

Workers

6

RPS

26.3

Failures

29%

NEW

RESET

STATISTICS

CHARTS

FAILURES

EXCEPTIONS

CURRENT RATIO

DOWNLOAD DATA

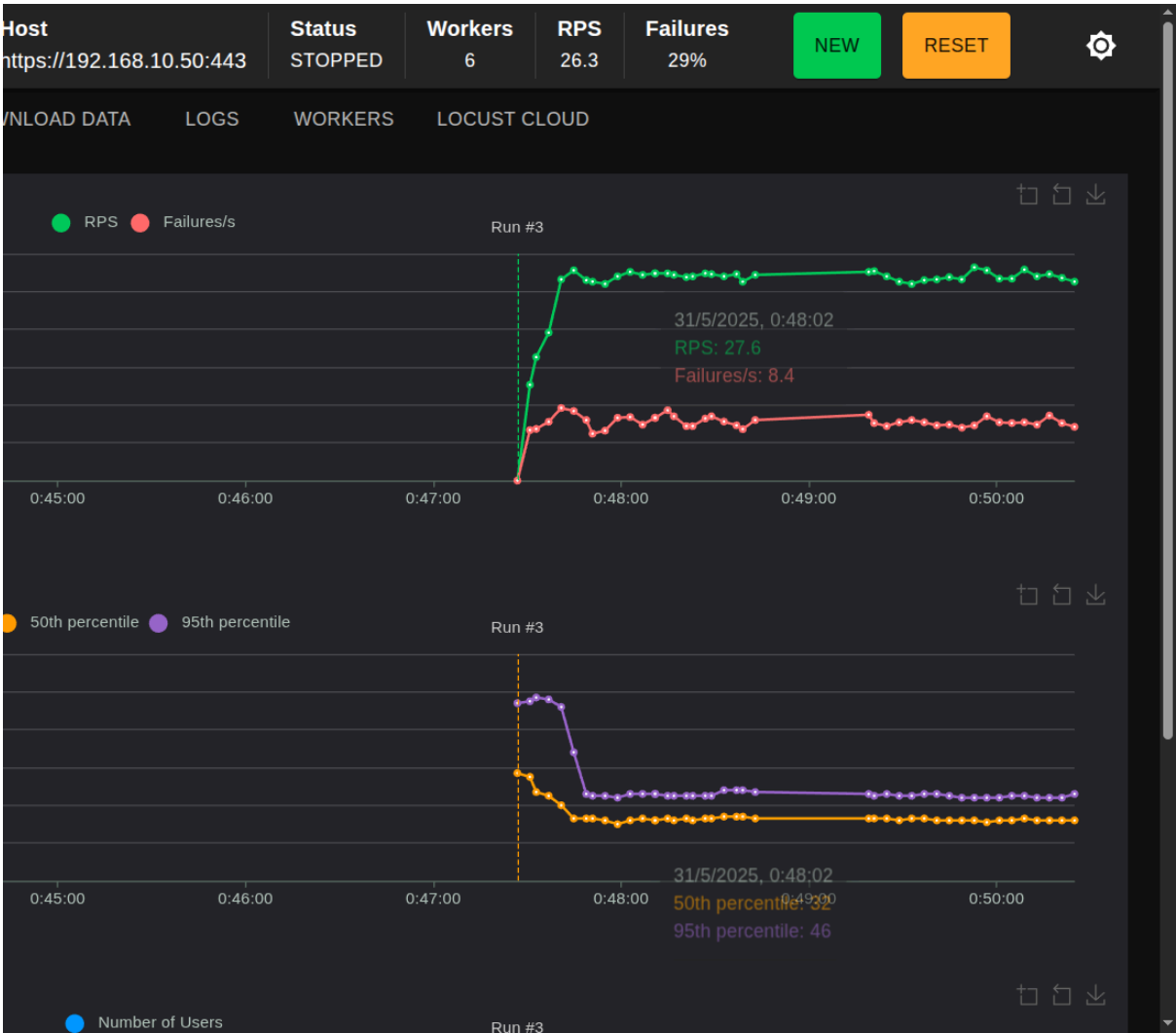
LOGS

WORKERS

LOCUST CLOUD

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s	
GET	Admin - Dashboard	545	0	31	38	57	31.9	26	136	1460	3.1	0	
GET	Admin - New Post	254	0	31	36	42	31.3	27	51	1463	1	0	
GET	Admin - Posts List	265	0	31	36	38	31.53	26	92	1459	1.7	0	
GET	Admin - Profile	248	0	31	35	46	31.12	26	60	1462	1.2	0	
GET	Anon - Home	1231	0	43	52	95	44.17	37	121	84988	7	0	
GET	Anon - Post	794	717	31	43	51	32.12	26	74	58265.13	4	3.5	
GET	Anon - Search	776	0	34	41	45	34.79	28	52	63844.5	4.7	0	
POST	Login	684	684	16	61	78	20.76	13	86	7178.87	3.6	3.6	
	Aggregated	4797	1401	33	47	72	33.87	13	136	43204.7	26.3	7.1	

ABOUT



# Failures	Method	Name	Message
717	GET	Anon - Post	HTTPError(404 Client Error: Not Found for url: Anon - Post')
684	POST	Login	CatchResponseError('login-failed')

Con estos parámetros, vemos un comportamiento muy similar, con los mismos errores comentados anteriormente solo que en mayor cantidad (más usuarios y más tiempo de prueba de carga).

Por lo tanto, vemos que el rendimiento en sí es muy bueno, que los errores comentados es debido a situaciones de prueba de carga pero con zonas que no tenemos en nuestro wordpress pero que había que probarlos.

Vemos que el ratio de errores no es muy grande, sino entorno a un 30% por los errores ya comentados.

Como recomendación, primero sería mejorar el propio test de login y evitar peticiones a post inexistentes , para evitar los errores .

IA

Enlace: <https://chatgpt.com/share/683a36ef-68e0-800d-9d10-b859914e1fd3>

Al comienzo, la IA, me recomendaba utilizar docker warm. Sin embargo, no lo hice y modifiqué mi script de [initial.sh](#) para que al hacer `./init.sh` -u locust se creen los workers (sus replicas).

Por otro lado, he usado esta vez más la IA que nunca debido a que me han surgido muchos errores, al comienzo por el uso de iptables muy restringido que tenía, luego con el desarrollo del CMS y los errores que me daban al hacer pruebas de carga (inventándose la IA algunas partes). Pero al final, decidí no implementar tal cual las cosas propuestas por la IA, como el uso de dos wordpress, ya que me daba error en las comunicaciones con MariaDB.

Por lo demás, lo he utilizado para la parte más educativa y entender/aprender las partes que nunca había escuchado o por ejemplo que no sabía que era en sí un CMS.

Por lo demás, en programación esta vez me ha “fallado” más que en el resto de prácticas, comento esto porque en las anteriores prácticas comenté que casi nunca fallaba en la parte de programación y esta vez sí.

Pero por lo demás todo bien, al final es una herramienta muy útil.

Poco más que añadir, he intentado realizar pruebas con el escalado automático, pero no tengo mucho tiempo ahora con los exámenes finales para seguir probandolo.