

TODOS ESTOS ANÁLISIS FUERON APLICADOS EN LA RUTA/INFO

GZIP

Podemos ver que el tamaño de la ruta /info es de 1.8kb y utilizando gzip para poder comprimir, el tamaño pasa a ser a 1.1kb

RESULTADO ARTILLERY CON FORK

```
1 Running scenarios...
2 Phase started: unnamed (index: 0, duration: 1s) 15:32:29(-0300)
3
4 Phase completed: unnamed (index: 0, duration: 1s) 15:32:30(-0300)
5
6 All VUs finished. Total time: 10 seconds
7
8
9 Summary report @ 15:32:32(-0300)
10
11
12 http.codes.200: ..... 1000
13 http.request_rate: ..... 72/sec
14 http.requests: ..... 1000
15 http.response_time:
16 min: ..... 7
17 max: ..... 286
18 median: ..... 144
19 p95: ..... 175.9
20 p99: ..... 202.4
21 http.responses: ..... 1000
22 users.completed: ..... 50
23 users.created: ..... 50
24 users.created_by_name.0: ..... 50
25 users.failed: ..... 0
26 users.session_length:
27 min: ..... 811.8
28 max: ..... 2009.2
29 median: ..... 2725
30 p95: ..... 2893.5
31 p99: ..... 2893.5
32
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\elnacholi\Desktop\desafios\desafio-backend-3>
History restored

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\elnacholi\Desktop\desafios\desafio-backend-3>
```

RESULTADO ARTILLERY CON CLUSTER

```
1 Running scenarios...
2 Phase started: unnamed (index: 0, duration: 1s) 15:37:05(-0300)
3
4 Phase completed: unnamed (index: 0, duration: 1s) 15:37:06(-0300)
5
6 All VUs finished. Total time: 10 seconds
7
8
9 Summary report @ 15:37:12(-0300)
10
11
12 http.codes.200: ..... 1000
13 http.request_rate: ..... 274/sec
14 http.requests: ..... 1000
15 http.response_time:
16 min: ..... 3
17 max: ..... 143
18 median: ..... 45.2
19 p95: ..... 77.5
20 p99: ..... 92.8
21 http.responses: ..... 1000
22 users.completed: ..... 50
23 users.created: ..... 50
24 users.created_by_name.0: ..... 50
25 users.failed: ..... 0
26 users.session_length:
27 min: ..... 159.8
28 max: ..... 1110.2
29 median: ..... 1022.7
30 p95: ..... 1274.3
31 p99: ..... 1300.1
32
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\elnacholi\Desktop\desafios\desafio-backend-3>
History restored

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\elnacholi\Desktop\desafios\desafio-backend-3>
```

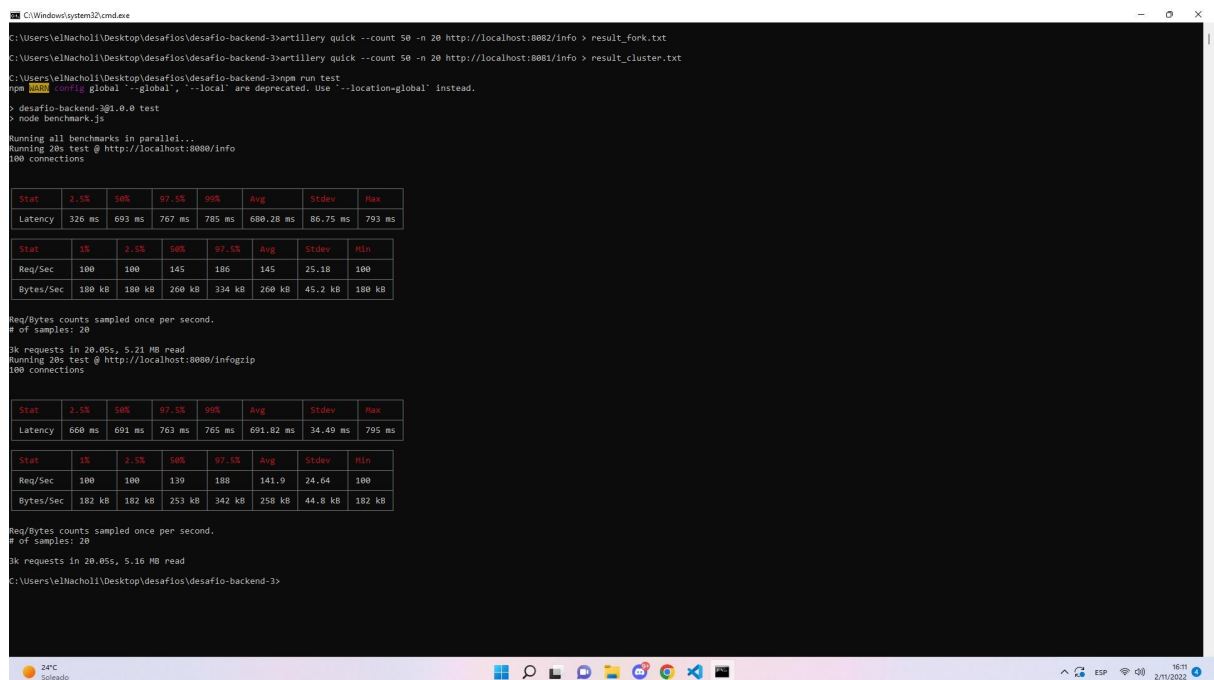
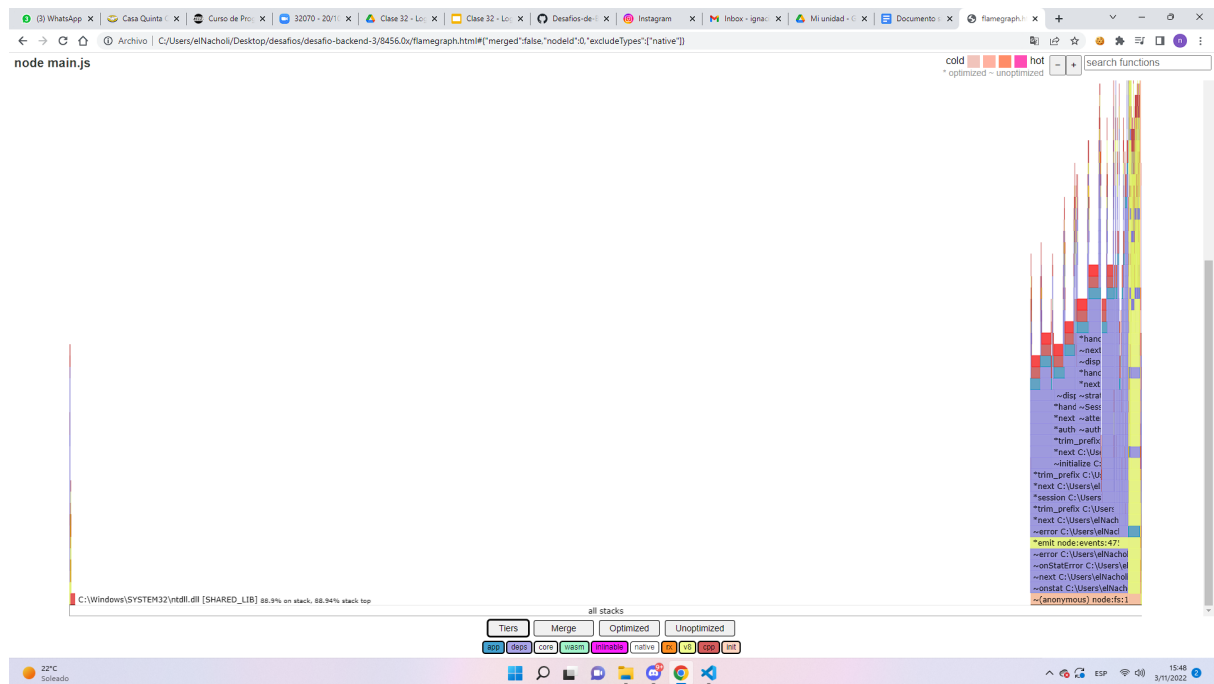
CONCLUSIÓN

Utilizando Artillery en los dos modos Fork y Cluster, realizando 50 conexiones concurrentes con 20 request por cada una, podemos ver que la media de respuestas por segundo es mucho más alta en el cluster, por este motivo es más eficiente utilizar este modo.

Por otro lado, los milisegundos de latencia se ven que también son más altos en modo Fork que en el modo Cluster.

Por lo tanto podemos concluir que es mejor ejecutar en modo cluster

AUTOCANNON



CONCLUSIÓN

Utilizando Autocannon en línea de comandos, emulando 100 conexiones concurrentes realizadas en un tiempo de 20 segundos.

En autocannon llegamos a la misma conclusión que en artillery, que es más conveniente utilizar modo cluster, porque manejamos muchos procesos y por ende este modo es más eficiente y se puede tener más concurrencia

Es más eficiente utilizar modo cluster porque podemos tener más concurrencia

Por lo que podemos decir que sus procesos duran muchísimo menos tiempo en el stack (microseconds) como debe suceder en procesos más eficientes.