

Smoke test

```

      .io
    execution: local
      script: .\smoke_Test.js
      output: -

scenarios: (100.00%) 1 scenario, 5 max VUs, 1m30s max duration (incl. graceful stop):
    * default: 5 looping VUs for 1m0s (gracefulStop: 30s)

✓ response code was 200

✓ checks.....: 100.00% ✓ 361450          ✗ 0
data_received.....: 102 MB   1.7 MB/s
data_sent.....: 32 MB   530 kB/s
http_req_blocked.....: avg=6.46µs   min=0s med=0s max=18.32ms p(90)=0s p(95)=0s
http_req_connecting.....: avg=4.42µs   min=0s med=0s max=4.01ms p(90)=0s p(95)=0s
✓ http_req_duration.....: avg=764.95µs min=0s med=0s max=11.34ms p(90)=4ms p(95)=4ms
  { expected_response:true }...: avg=764.95µs min=0s med=0s max=11.34ms p(90)=4ms p(95)=4ms
✓ http_req_failed.....: 0.00%   ✓ 0              ✗ 361450
http_req_receiving.....: avg=71.08µs   min=0s med=0s max=7.34ms p(90)=0s p(95)=0s
http_req_sending.....: avg=8.12µs    min=0s med=0s max=4.52ms p(90)=0s p(95)=0s
http_req_tls_handshaking.....: avg=0s        min=0s med=0s max=0s     p(90)=0s p(95)=0s
http_req_waiting.....: avg=685.74µs  min=0s med=0s max=4.76ms p(90)=4ms p(95)=4ms
http_reqs.....: 361450  6024.108795/s
iteration_duration.....: avg=822.97µs  min=0s med=0s max=20.73ms p(90)=4ms p(95)=4ms
iterations.....: 361450  6024.108795/s
vus.....: 5         min=5       max=5
vus_max.....: 5         min=5       max=5

running (1m00.0s), 0/5 VUs, 361450 complete and 0 interrupted iterations
default ✓ [=====] 5 VUs  1m0s
```

Como podemos ver el servidor es capaz perfectamente de aguantar la carga que se solicita, al ser esta muy baja ya que estamos en un smoke test en el que simplemente se prueba que todo funcione correctamente y que se pueden hacer peticiones al servidor sin problema.

Breakpoint con executors

HTTP Performance overview



metric	Trends						
	avg	max	med	min	p90	p95	p99
http_req_blocked	175µs	405ms	0ms	0ms	0ms	0ms	110µs
http_req_connecting	160µs	405ms	0ms	0ms	0ms	0ms	0ms
http_req_duration	122ms	4s	110ms	0ms	266ms	292ms	422ms
http_req_receiving	4ms	1s	0ms	0ms	8ms	23ms	84ms
http_req_sending	284µs	463ms	0ms	0ms	0ms	0ms	1ms
http_req_tls_handshaking	0ms	0ms	0ms	0ms	0ms	0ms	0ms
http_req_waiting	118ms	4s	105ms	0ms	260ms	284ms	407ms
iteration_duration	155ms	1m	115ms	0ms	271ms	300ms	450ms

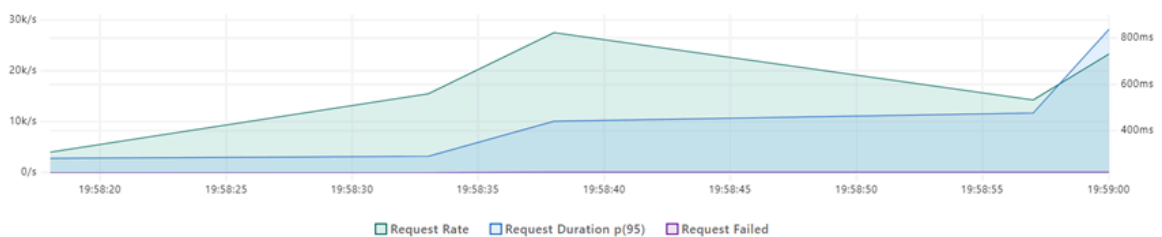
En este caso aunque al servidor le cuesta algo más debido a que la carga ya es muy elevada aún es capaz de aguantar el número de usuarios virtuales (100.000), como podemos ver las peticiones empiezan a subir de manera gradual hasta llegar a los 100.000 usuarios virtuales, donde permanece constante durante unos minutos y como es observable los fallos son mínimos, tanto que no llega a ese máximo del 1%.

Breakpoint sin executors

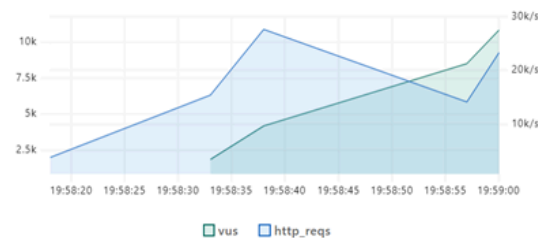
Trends							
metric	avg	max	med	min	p90	p95	p99
http_req_blocked	3ms	1s	0ms	0ms	0ms	0ms	164ms
http_req_connecting	3ms	1s	0ms	0ms	0ms	0ms	162ms
http_req_duration	310ms	11s	283ms	0ms	479ms	577ms	908ms
http_req_receiving	1ms	631ms	0ms	0ms	0ms	1ms	28ms
http_req_sending	191µs	11s	0ms	0ms	0ms	0ms	2ms
http_req_tls_handshaking	0ms	0ms	0ms	0ms	0ms	0ms	0ms
http_req_waiting	308ms	2s	283ms	0ms	477ms	570ms	907ms
iteration_duration	374ms	15s	318ms	0ms	537ms	685ms	1s

Counters			Rates		Gauges	
metric	count	rate	metric	rate	metric	value
data_received	238 MB	4.04 MB/s	http_req_failed	0.01/s	vus	15.9k
data_sent	73.6 MB	1.25 MB/s			vus_max	100k
http_reqs	847.5k	14.35k/s				
iterations	833.6k	14.12k/s				

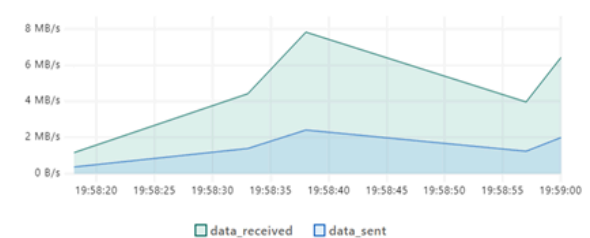
HTTP Performance overview



VUs



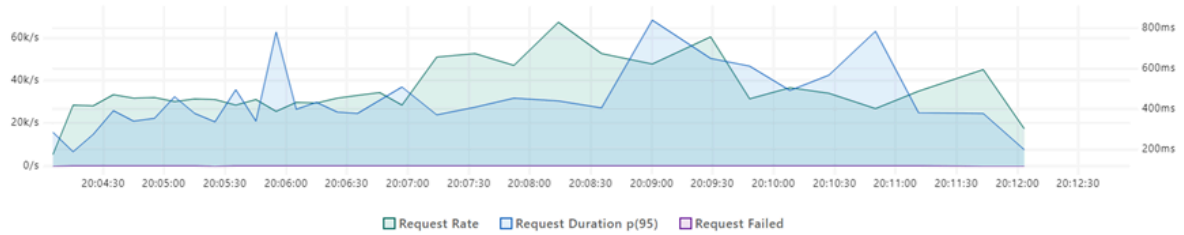
Transfer Rate



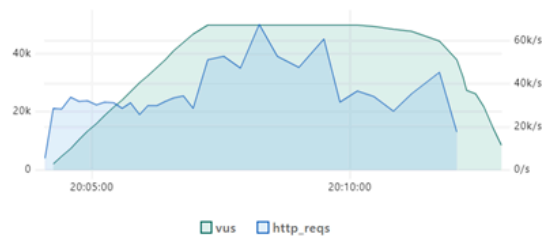
Aquí a diferencia de en el anterior caso, al no haber executors las subidas son más pronunciadas, tanto que el servidor no es capaz de aguantar las peticiones y al minuto se incumple el threshold de que las peticiones fallidas se mantengan por debajo del 1%. Sin embargo podemos ver como se han llegado a los 100.000 vus, considerando así que ese es el máximo que el servidor puede soportar y que usaremos como referencia en los siguientes casos.

Average load test

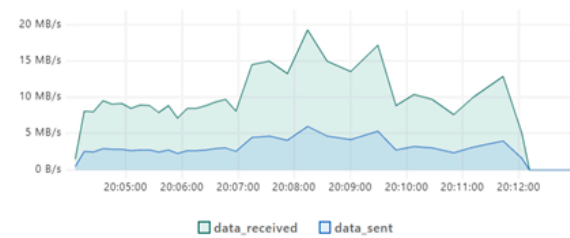
HTTP Performance overview



VUs



Transfer Rate



Trends

metric	avg	max	med	min	p90	p95	p99
http_req_blocked	2ms	7s	0ms	0ms	0ms	0ms	4ms
http_req_connecting	2ms	7s	0ms	0ms	0ms	0ms	2ms
http_req_duration	257ms	11s	236ms	0ms	411ms	525ms	768ms
http_req_receiving	1ms	10s	0ms	0ms	0ms	1ms	34ms
http_req_sending	186µs	10s	0ms	0ms	0ms	0ms	2ms
http_req_tls_handshaking	0ms	0ms	0ms	0ms	0ms	0ms	0ms
http_req_waiting	255ms	9s	235ms	0ms	409ms	523ms	764ms
iteration_duration	856ms	6m	249ms	0ms	446ms	570ms	991ms

Counters

metric	count	rate
data_received	3.22 GB	5.95 MB/s
data_sent	993 MB	1.84 MB/s
http_reqs	11.4m	20.99k/s
iterations	11.3m	20.92k/s

Rates

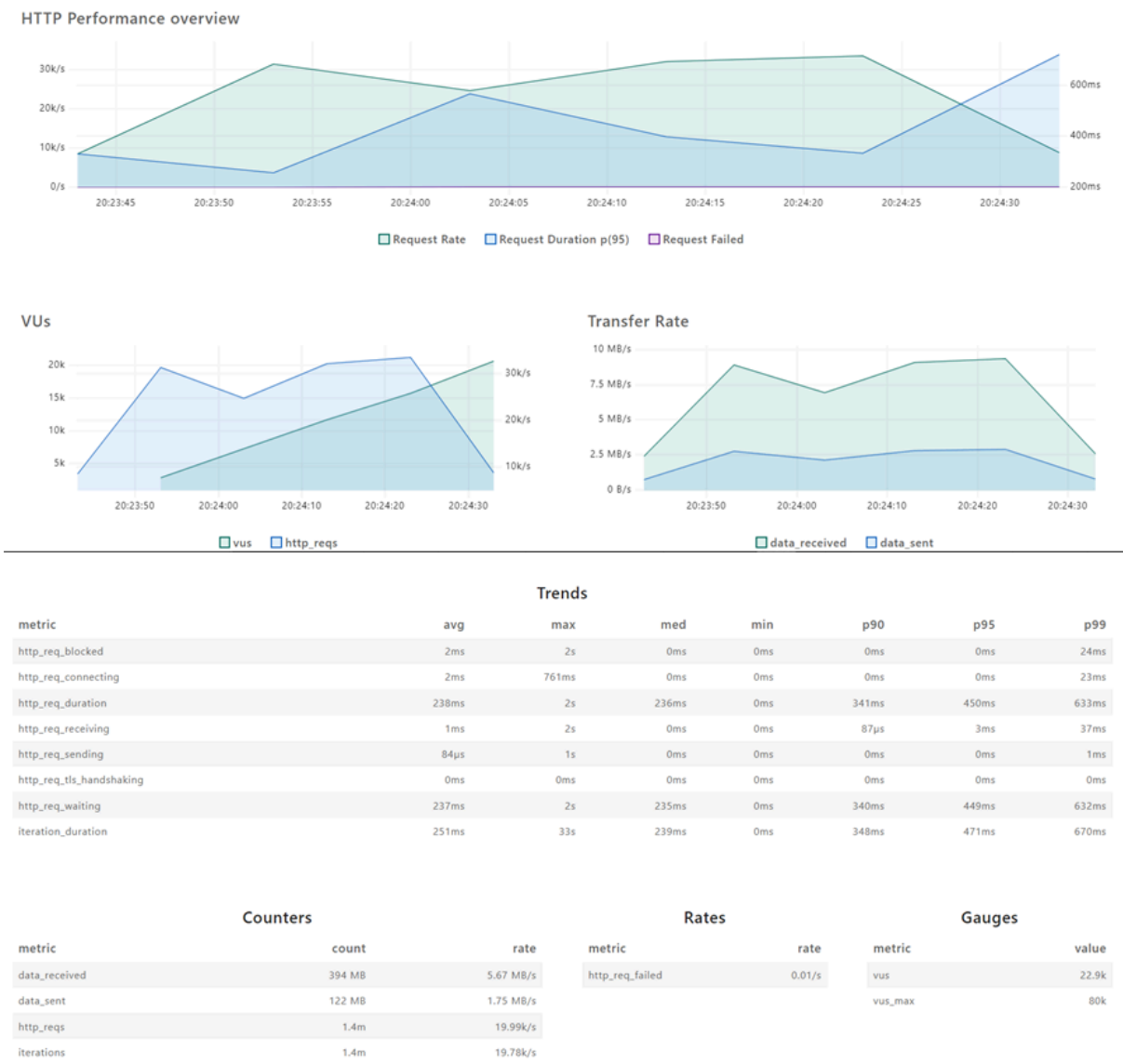
metric	rate
http_req_failed	0.01/s

Gauges

metric	value
vus	344
vus_max	50k

Como se puede observar en la gráfica, los VU's aumentan de forma lineal llegando a ser 50000 (50% de los VU's totales) en 3 minutos, luego se mantienen 3 minutos en esos 50000 para luego disminuir a 0 en 2 minutos, por eso la bajada es más pronunciada que la subida. Sobre el test, se ve como no ha habido ningún inconveniente y no se ha superado ese 1% de peticiones fallidas en ningún momento pudiendo acabar el test sin problemas. Lo que sí se puede apreciar es que la duración de las peticiones aumentan considerablemente cuando llega al 50% de los VU's totales.

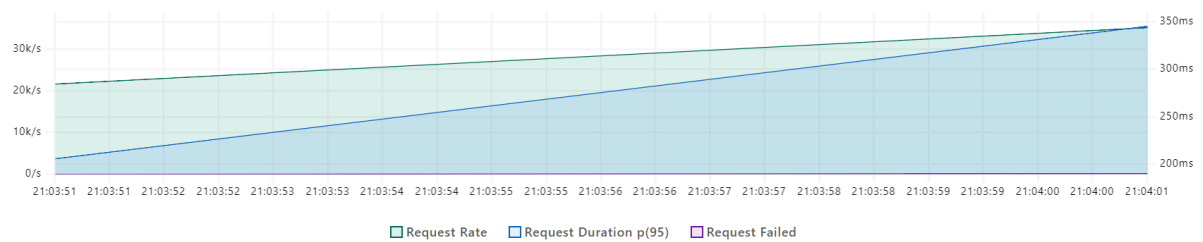
Stress test



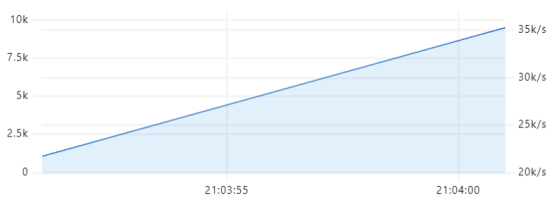
Como se puede observar en la gráfica, los VU's suben de forma lineal llegando a ser 20% de los VU's totales, después de esto podemos ver como caen a consecuencia de que no se han pasado los checks establecidos y por tanto se ha abortado el test, viendo así que no ha sido capaz de superar el test de stress, ya que la carga es mayor de la que puede soportar el servidor web.

Spike test

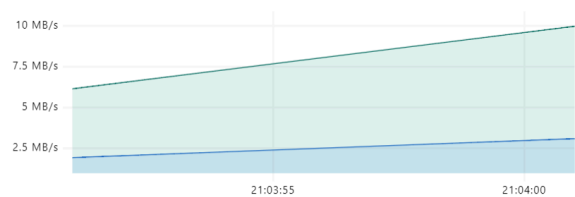
HTTP Performance overview



VUs



Transfer Rate



Trends

metric	avg	max	med	min	p90	p95	p99
http_req_blocked	2ms	591ms	0ms	0ms	0ms	0ms	43ms
http_req_connecting	2ms	591ms	0ms	0ms	0ms	0ms	41ms
http_req_duration	191ms	2s	194ms	0ms	282ms	350ms	507ms
http_req_receiving	1ms	618ms	0ms	0ms	771µs	4ms	43ms
http_req_sending	78µs	803ms	0ms	0ms	0ms	0ms	1ms
http_req_tls_handshaking	0ms	0ms	0ms	0ms	0ms	0ms	0ms
http_req_waiting	189ms	2s	193ms	0ms	281ms	349ms	505ms
iteration_duration	198ms	9s	200ms	0ms	290ms	364ms	526ms

Counters

metric	count	rate
data_received	202 MB	6.82 MB/s
data_sent	62.5 MB	2.11 MB/s
http_reqs	706.3k	23.8k/s
iterations	702.1k	23.65k/s

Rates

metric	rate
http_req_failed	0.01/s

Gauges

metric	value
vus	15.3k
vus_max	40k

Este test consiste en incrementar los VU's hasta un 40% de los totales en 1 minuto, unos 40000, y bajar a 0 en otro minuto. Lo que se observa en la gráfica es como ni llega a durar el tiempo que debería ni se consigue ese volumen de VU's ya que el test falla a los pocos segundos de comenzar. Esto se debe a que supera el 0.5% de peticiones fallidas y por tanto el test acaba prematuramente.