TP 1 de Arquitectura del Software

1C - 2022

Informe

Grupo 3

Marchese Milena	100962
Sebastián Blázquez	99673

Sección 1

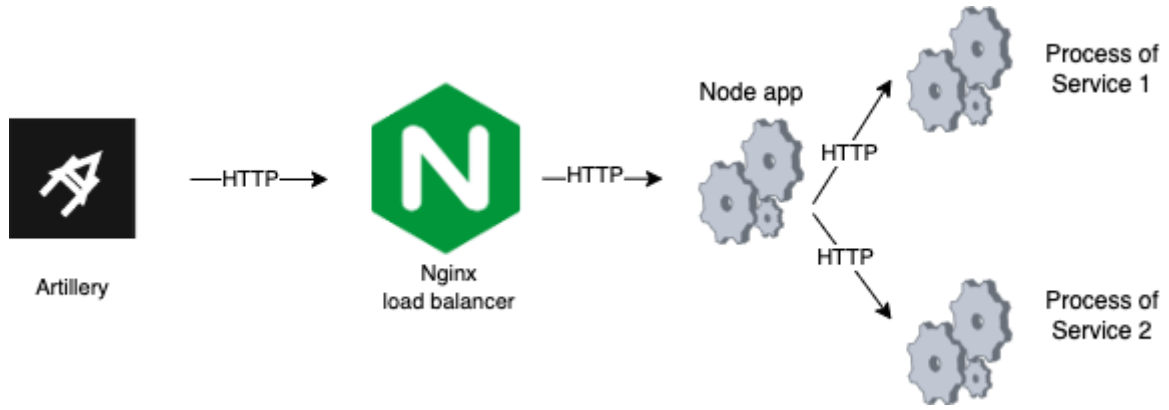
En esta sección detallaremos cómo se comportan las 4 implementaciones:

- Healthcheck: Respuesta de un valor constante.
- Proxy1: Invocación a servicio 1 provisto por la cátedra.
- Proxy2: Invocación a servicio 2 provisto por la cátedra.
- Heavy: Loop de cierto tiempo.

Frente a los diferentes escenarios variando la cantidad de nodos.

Análisis para un solo nodo

Vista components & connectors:



Con Artillery se simulan los diferentes escenarios, de estos escenarios surge el tráfico de requests que pasará a través de Nginx hacia la aplicación en node. Desde la aplicación en node se llama a los servicios que brinda bbox.

Load

- Healthcheck


```
http.codes.200: ..................................................... 4177
http.request_rate: .................................................. 29/sec
http.requests: ...................................................... 4177
http.response_time:
  min: .............................................................. 2
  max: .............................................................. 89
  median: ........................................................... 7
  p95: .............................................................. 15
  p99: .............................................................. 23.8
```

No hay degradación y el tiempo de respuesta se mantiene aproximadamente constante.

- Proxy1


```
http.codes.200: ..................................... 4149
http.request_rate: .................................. 30/sec
```

```
http.requests: ..................................................... 4149
http.response_time:
  min: ............................................................. 1404
  max: ............................................................. 1820
  median: .......................................................... 1408.4
  p95: ............................................................. 1436.8
  p99: ............................................................. 1525.7
```

No hay degradación y el tiempo de respuesta se mantiene constante.

- Proxy2


```
errors.ETIMEDOUT: ................................................. 4054
http.codes.200: ................................................... 107
http.request_rate: ................................................ 23/sec
http.requests: ..................................................... 4161
http.response_time:
  min: ............................................................. 758
  max: ............................................................. 9870
  median: .......................................................... 3678.4
  p95: ............................................................. 9416.8
  p99: ............................................................. 9801.2
```

Rápidamente degrada y aumentan los tiempos de respuesta de 750ms a 7s aproximadamente.

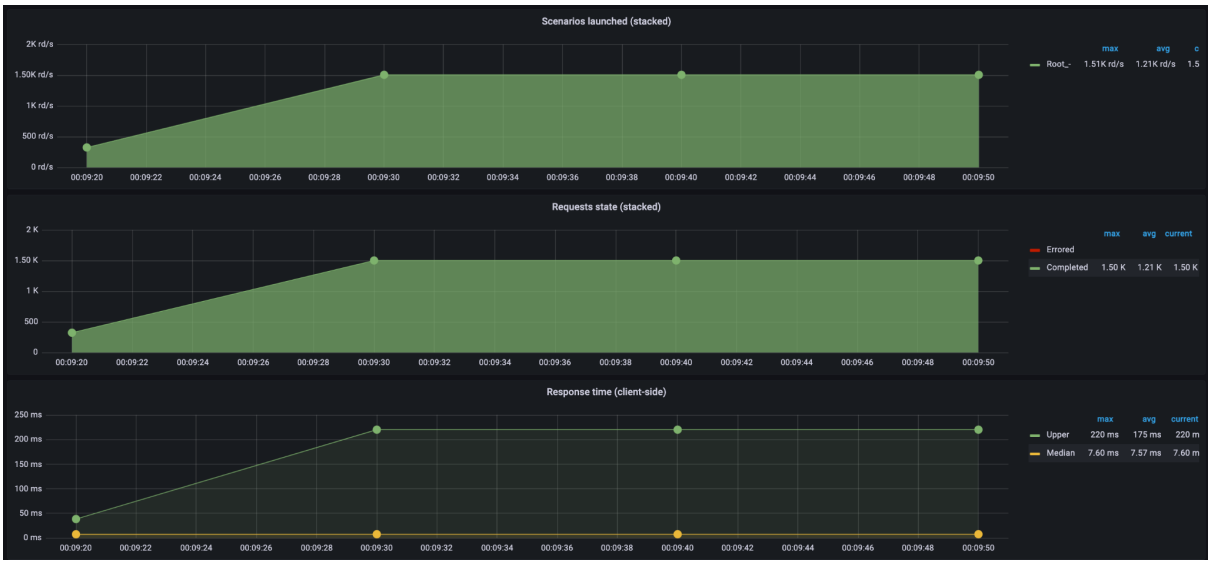
- Heavy


```
errors.ETIMEDOUT: ..................................... 4176
http.codes.200: ........................................ 1
http.request_rate: ..................................... 26/sec
http.requests: ......................................... 4177
http.response_time:
  min: ................................................. 9997
  max: ................................................. 9997
  median: .............................................. 9999.2
  p95: ................................................. 9999.2
  p99: ................................................. 9999.2
```

Degrada completamente y completa un solo request.

Spike

- Healthcheck




```
http.codes.200: ....................................................... 6135
http.request_rate: .................................................... 220/sec
http.requests: ........................................................ 6135
http.response_time:
```

```
    min: ..................................................................... 2
    max: ..................................................................... 1327
    median: .................................................................. 46.1
    p95: ..................................................................... 804.5
    p99: ..................................................................... 1200.1
```

No hay degradación y el tiempo de respuesta es constante en promedio. Sin embargo el tiempo de respuesta máximo se aleja considerablemente del promedio, lo cual no ocurre con otros escenarios de carga para el mismo endpoint.

- Proxy1


```
    errors.ECONNRESET: ....................................................... 1600
    errors.ETIMEDOUT: ........................................................ 2939
    http.codes.200: .......................................................... 1678
    http.request_rate: ....................................................... 192/sec
    http.requests: ........................................................... 6217
    http.response_time:
      min: ................................................................... 1404
      max: ................................................................... 7265
      median: ................................................................ 1495.5
      p95: ................................................................... 7117
      p99: ................................................................... 7260.8
```

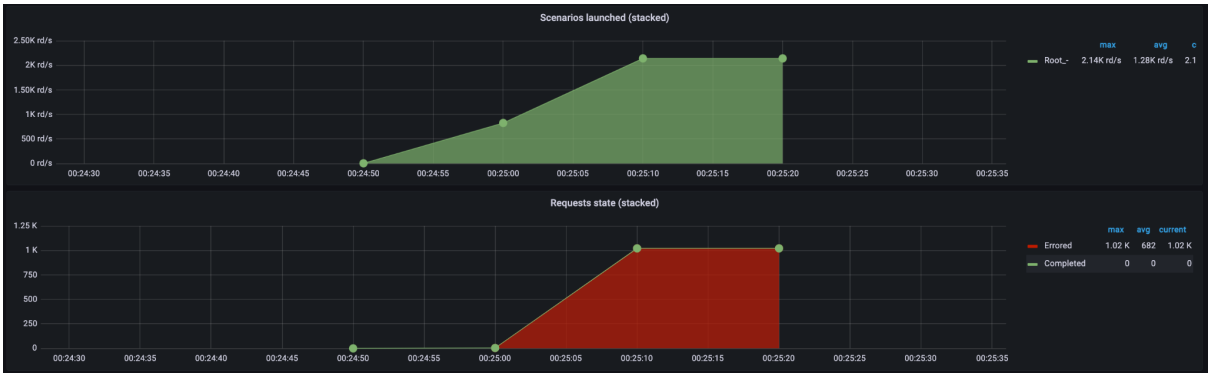
Responde correctamente hasta 2000 requests por segundo pero pasando dicho umbral el servicio se degrada considerablemente aumentando los tiempos de respuesta y los errores.

- Proxy2


```
errors.ECONNRESET: ..................................................... 1697
errors.ETIMEDOUT: ...................................................... 4455
http.codes.200: ........................................................ 64
http.request_rate: ..................................................... 168/sec
http.requests: ......................................................... 6216
http.response_time:
  min: ................................................................. 767
  max: ................................................................. 9726
  median: .............................................................. 4403.8
  p95: ................................................................. 9047.6
  p99: ................................................................. 9607.1
```

Rápidamente degrada y aumentan los tiempos de respuesta a 1.5s aproximadamente.

- Heavy




```
errors.ECONNRESET: ..................................................... 1622
errors.ETIMEDOUT: ...................................................... 4598
http.request_rate: ..................................................... 138/sec
http.requests: ......................................................... 6220
vusers.created: ........................................................ 6220
vusers.created_by_name.Root (/): ....................................... 6220
vusers.failed: ......................................................... 6220
```

Degrada completamente y no completa ninguno de los requests.

Volume

- Healthcheck


```
http.codes.200: ..................................................... 8220
http.request_rate: ................................................... 53/sec
http.requests: ...................................................... 8220
http.response_time:
  min: ............................................................... 2
  max: ............................................................... 1023
  median: ............................................................ 6
  p95: ............................................................... 19.1
  p99: ............................................................... 63.4
```

No hay degradación y tiene un tiempo de respuesta constante en promedio.

- Proxy1


```
http.codes.200: ..................................................... 8235
http.request_rate: ................................................... 53/sec
http.requests: ...................................................... 8235
http.response_time:
```

```
  min: ..... 1404
  max: ..... 2635
  median: ..... 1408.4
  p95: ..... 1525.7
  p99: ..... 2186.8
```

No hay degradación y tiene un tiempo de respuesta constante.

- Proxy2


```
errors.ETIMEDOUT: ..... 8167
http.codes.200: ..... 93
http.request_rate: ..... 53/sec
http.requests: ..... 8260
http.response_time:
  min: ..... 757
  max: ..... 9830
  median: ..... 4492.8
  p95: ..... 9047.6
  p99: ..... 9607.1
```

Rápidamente degrada y aumentan los tiempos de respuesta a 8s aproximadamente.

- Heavy

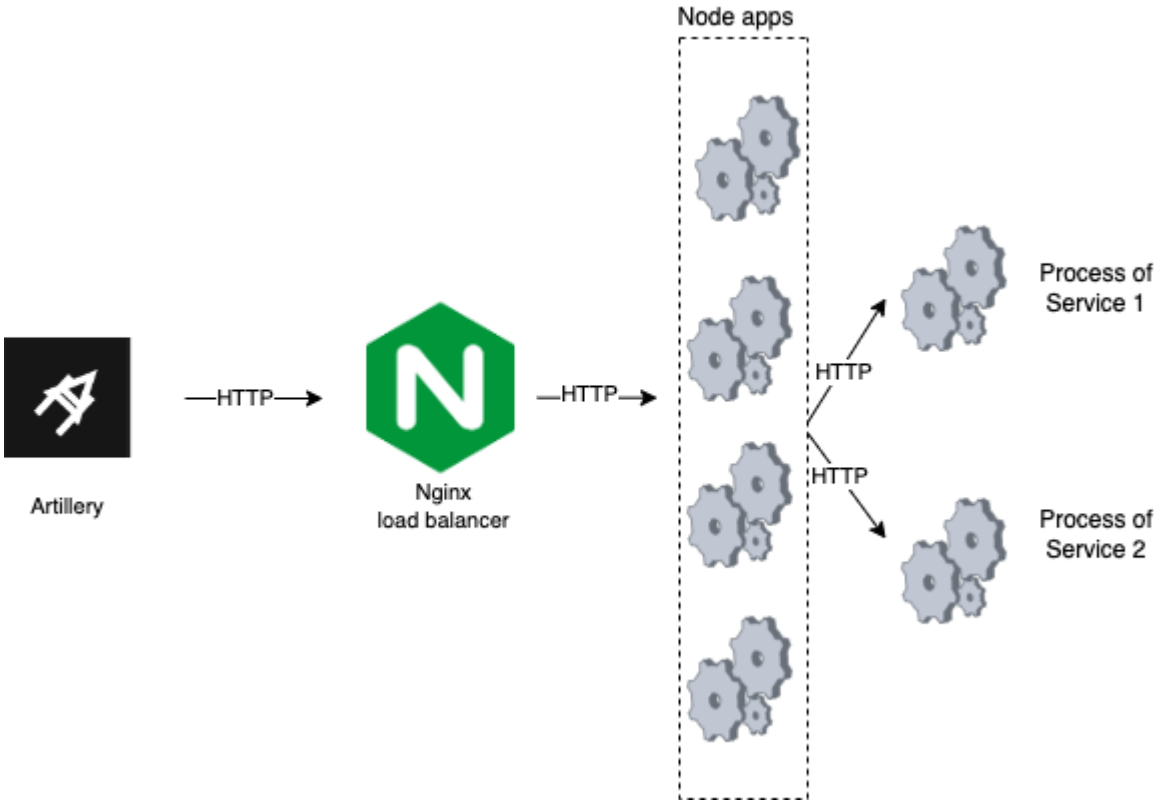

```
errors.ETIMEDOUT: ..... 8254
http.request_rate: ..... 49/sec
```

```
http.requests: ..................................................... 8254
vusers.created: .................................................... 8254
vusers.created_by_name.Root (/): ................................... 8254
vusers.failed: ..................................................... 8254
```

Degrada completamente y no completa ningún request.

Análisis para cinco nodos

Vista de components & connectors:



Mantenemos el mismo flujo que para el caso de un solo nodo pero esta vez escalando horizontalmente teniendo cinco aplicaciones que se comunicacion los dos servicios que nos brinda bbox.

Load

- Healthcheck

```
http.codes.200: ..................................................... 4178
http.request_rate: ................................................. 30/sec
http.requests: ..................................................... 4178
http.response_time:
  min: ............................................................. 2
  max: ............................................................. 105
  median: .......................................................... 6
  p95: ............................................................ 10.9
  p99: ............................................................ 18
```

No hay degradación y tiene un tiempo de respuesta constante en promedio.

- Proxy1


```
http.codes.200: ..................................................... 4182
http.request_rate: ................................................. 30/sec
http.requests: ..................................................... 4182
http.response_time:
  min: ............................................................ 1404
  max: ............................................................ 2351
  median: ......................................................... 1408.4
  p95: ............................................................ 1436.8
  p99: ............................................................ 1720.2
```

No hay degradación y tiene un tiempo de respuesta constante en promedio.

- Proxy2


```
errors.ETIMEDOUT: ..................................................... 4043
http.codes.200: ....................................................... 114
http.request_rate: .................................................... 24/sec
http.requests: ........................................................ 4157
http.response_time:
  min: ................................................................ 757
  max: ................................................................ 9859
  median: ............................................................. 4147.4
  p95: ................................................................ 9230.4
  p99: ................................................................ 9607.1
```

Rápidamente degrada y aumentan los tiempos de respuesta a 8s aproximadamente.

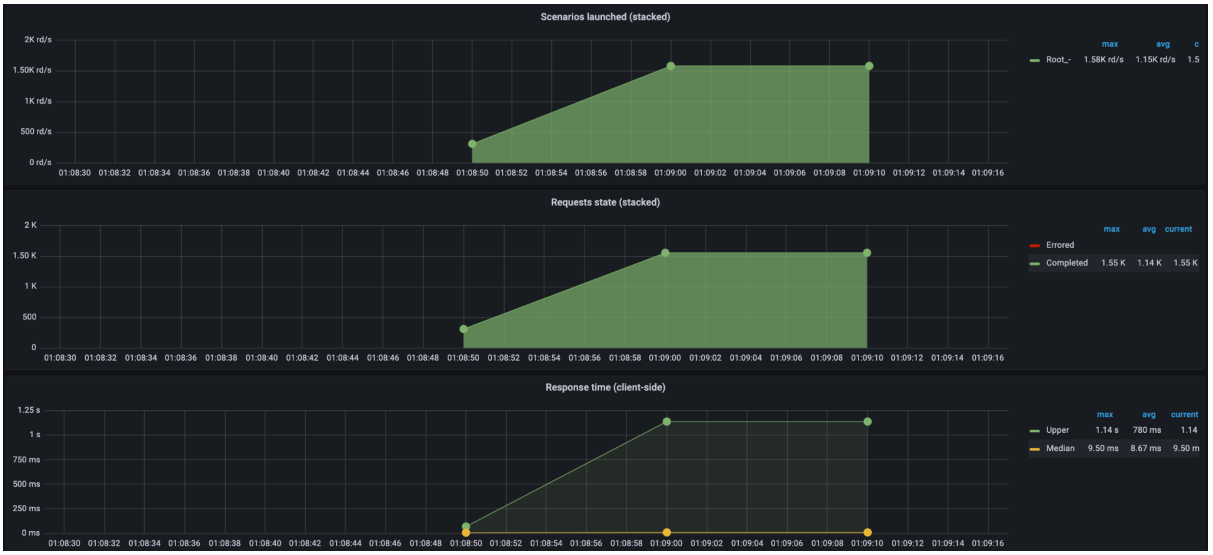
- Heavy


```
errors.ETIMEDOUT: ..................................................... 4187
http.request_rate: .................................................... 23/sec
http.requests: ........................................................ 4187
vusers.created: ....................................................... 4187
vusers.created_by_name.Root (/): ...................................... 4187
vusers.failed: ........................................................ 4187
```

Degrada completamente y no completa ningún request.

Spike

- Healthcheck




```
http.codes.200: ..................................................... 6091
http.request_rate: .................................................. 222/sec
http.requests: ...................................................... 6091
http.response_time:
  min: .............................................................. 2
  max: .............................................................. 3117
  median: ........................................................... 96.6
  p95: .............................................................. 2231
  p99: .............................................................. 2836.2
http.responses: ..................................................... 6091
vusers.completed: ................................................... 6091
vusers.created: ..................................................... 6091
vusers.created_by_name.Root (/): .................................... 6091
vusers.failed: ...................................................... 0
vusers.session_length:
  min: .............................................................. 3.6
  max: .............................................................. 3119.7
  median: ........................................................... 113.3
  p95: .............................................................. 2276.1
  p99: .............................................................. 2893.5
```

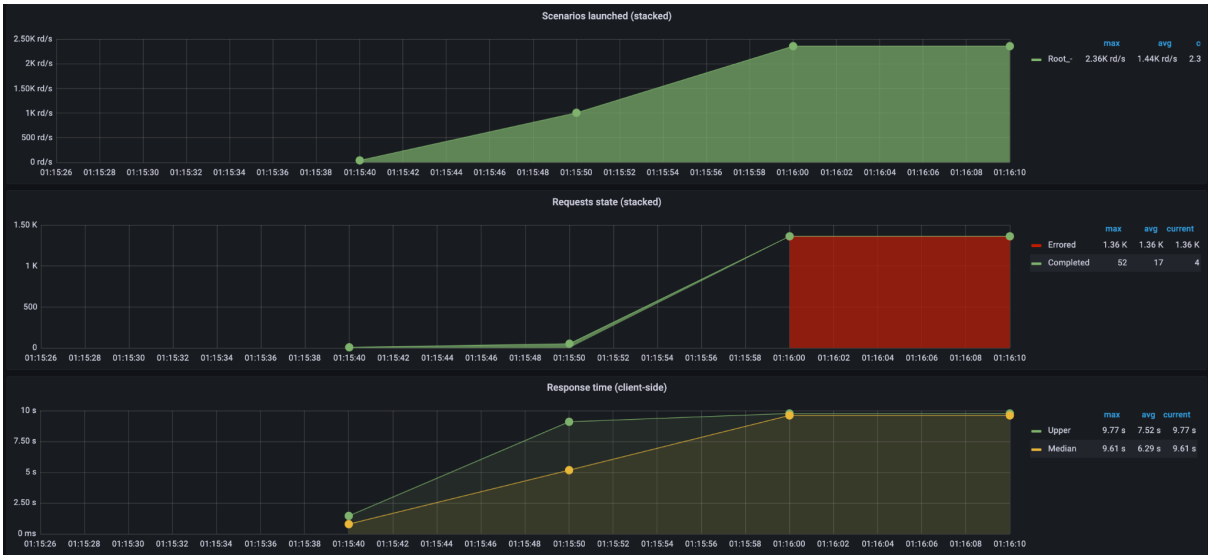
No hay degradación y el tiempo de respuesta es constante en promedio. Sin embargo el tiempo de respuesta máximo se aleja considerablemente del promedio, lo cual no ocurre con otros escenarios de carga para el mismo endpoint (excepto en el spike con un solo nodo).

- Proxy1


```
errors.ECONNRESET: ..................................................... 1271
errors.ETIMEDOUT: ...................................................... 618
http.codes.200: ........................................................ 4299
http.request_rate: ..................................................... 161/sec
http.requests: ......................................................... 6188
http.response_time:
  min: ................................................................. 1404
  max: ................................................................. 9995
  median: .............................................................. 4231.1
  p95: ................................................................. 9801.2
  p99: ................................................................. 9999.2
http.responses: ........................................................ 4299
vusers.completed: ...................................................... 4299
vusers.created: ........................................................ 6188
vusers.created_by_name.Root (/): ....................................... 6188
vusers.failed: ......................................................... 1889
vusers.session_length:
  min: ................................................................. 1406.2
  max: ................................................................. 10005.5
  median: .............................................................. 4231.1
  p95: ................................................................. 9801.2
  p99: ................................................................. 9999.2
```

Completa los requests de forma satisfactoria pero aumenta el tiempo de respuesta promedio de forma considerable.

- Proxy2




```
errors.ECONNRESET: ..................................................... 1546
errors.ETIMEDOUT: ..................................................... 4590
http.codes.200: ....................................................... 64
http.request_rate: .................................................... 144/sec
http.requests: ........................................................ 6200
http.response_time:
  min: ................................................................ 766
  max: ................................................................ 9764
  median: ............................................................. 4676.2
  p95: ................................................................ 9047.6
  p99: ................................................................ 9801.2
http.responses: ....................................................... 64
vusers.completed: ..................................................... 64
vusers.created: ....................................................... 6200
vusers.created_by_name.Root (/): ...................................... 6200
vusers.failed: ........................................................ 6136
vusers.session_length:
  min: ................................................................ 768.9
  max: ................................................................ 9766.6
  median: ............................................................. 4676.2
  p95: ................................................................ 9047.6
  p99: ................................................................ 9801.2
```

Rápidamente degrada y aumentan los tiempos de respuesta a 7s aproximadamente.

- Heavy

Degrada completamente y no completa ninguno de los requests.

Conclusiones

- Healthcheck

  Este endpoint es el que mejor tiempos de respuesta tiene, lo cual es bastante lógico dado que no realiza ninguna operación que tome un tiempo considerable en procesarse. En general tiene un tiempo de respuesta promedio menor a 10ms y su tiempo de respuesta máximo es similar. Sin embargo frente a una carga de tipo spike el tiempo de respuesta máximo se aleja considerablemente del promedio.

- Proxy1

  Bajo los escenarios de carga de tipo Load y Volume responde en 1.4s aproximadamente. Sin embargo cuando la carga es mucho mayor como en el caso del Spike degrada a partir de 2000 requests por segundo aproximadamente aumentando los tiempos de respuesta y la cantidad de errores devueltos. Por otro lado, al agregar varias réplicas de la aplicación node notamos que el servicio logra responder sin errores, aunque el tiempo de respuesta sí aumenta.

- Proxy2

  Como se verá en la siguiente sección, debido a que este servicio es sincrónico y posee únicamente 5 workers degrada rápidamente cuando la cantidad de requests es mayor a dicho valor. Durante un instante el servicio logra completar aproximadamente 50 requests por segundo de forma satisfactoria, pero aumentando el tiempo de respuesta de 750ms a 8s aproximadamente. A su vez este comportamiento no varía al agregar más réplicas de la aplicación node.

- Heavy

  Solo en un escenario este endpoint logró completar un request, en el resto devolvió errores frente a la totalidad de la carga. Debido al comportamiento bloqueante y de único thread del endpoint es el que peor performance logró y agregar réplicas de la aplicación node no ayudó a mejorar su performance.

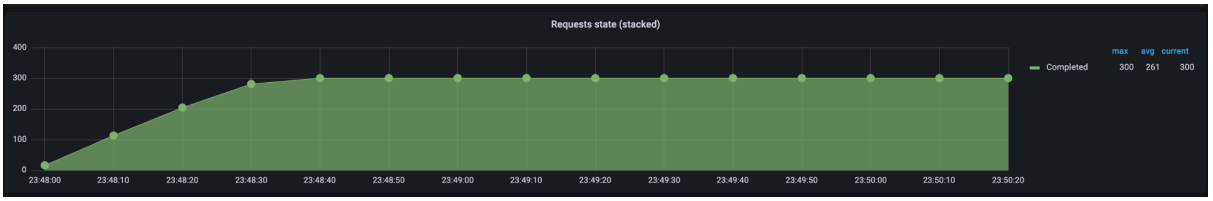
Sección 2

1. Sincrónico / Asincrónico: Uno de los servicios se comportará de manera sincrónica, y el otro de manera asincrónica. Deberán detectar de qué tipo es cada uno.

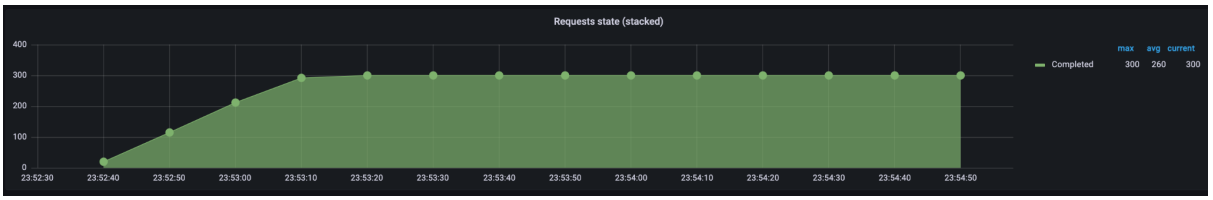
   Podemos observar a lo largo de los escenarios que probamos que el servicio 1 (Proxy1) presenta una mejor performance que el servicio 2 (Proxy2), con esto podemos detectar que el servicio 1 es el asincrónico y el servicio 2 es el sincrónico ya que, por ejemplo, es el que sufre más timeouts y su performance es más parecida al escenario implementado como Heavy o intensivo.

   Por ejemplo, veamos el estado de las requests en el caso de un solo nodo en el escenario Load para cada implementación:

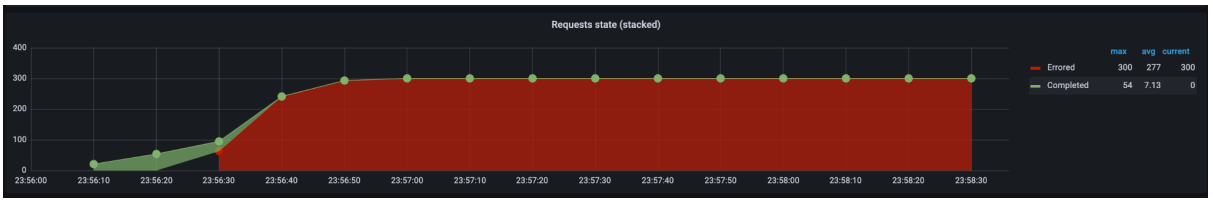
Healthcheck



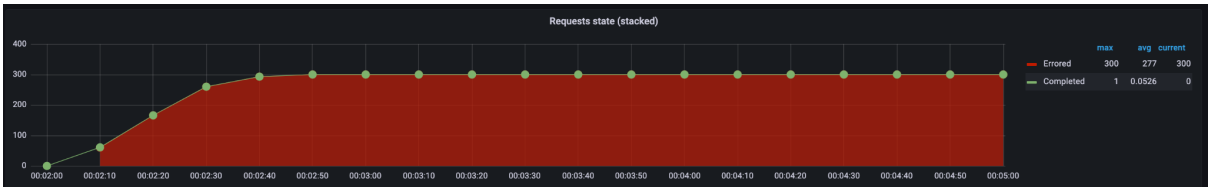
Proxy1



Proxy2



Heavy



2. Cantidad de workers (en el caso sincrónico): El servicio sincrónico está implementado con una cantidad de workers. Deberán buscar algún indicio sobre cuál es esta cantidad. El servicio asincrónico tiene una cantidad de event loops, que

también podrían intentar calcular, aunque esto es bastante más difícil y les recomendamos hacerlo sólo si terminaron con el resto.

Para resolver este punto se utilizo Apache HTTP server benchmarking tool. Realizando de uno a cinco requests de forma concurrente vemos que los tiempos de respuesta se mantienen bajo los 770ms:

```
Server Software:        nginx/1.21.6
Server Hostname:        localhost
Server Port:            5555

Document Path:          /app/proxy2
Document Length:        14 bytes

Concurrency Level:      1
Time taken for tests:   0.761 seconds
Complete requests:      1
Failed requests:        0
Total transferred:      235 bytes
HTML transferred:       14 bytes
Requests per second:    1.31 [#/sec] (mean)
Time per request:       760.880 [ms] (mean)
Time per request:       760.880 [ms] (mean, across all concurrent
requests)
Transfer rate:          0.30 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.0      0       0
Processing:   761  761   0.0    761     761
Waiting:      761  761   0.0    761     761
Total:        761  761   0.0    761     761
```

Sin embargo al realizar 6 requests de forma concurrente vemos que la performance baja de forma considerable y que el tiempo de respuesta máximo se duplica:

```
Server Software:        nginx/1.21.6
Server Hostname:        localhost
Server Port:            5555

Document Path:          /app/proxy2
Document Length:        14 bytes

Concurrency Level:      6
Time taken for tests:   2.275 seconds
```

```
Complete requests:      6
Failed requests:        0
Total transferred:      1410 bytes
HTML transferred:       84 bytes
Requests per second:    2.64 [#/sec] (mean)
Time per request:       2274.808 [ms] (mean)
Time per request:       379.135 [ms] (mean, across all concurrent
requests)
Transfer rate:          0.61 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.1      0       0
Processing:   758  889 307.1    766    1516
Waiting:      758  889 307.2    766    1516
Total:        759  889 307.2    766    1516
```

Consecuentemente podemos concluir que el proxy 2 tiene 5 workers, y por ello al realizar 6 requests concurrentes uno de ellos tomó el doble de tiempo dado que tuvo que esperar a que uno de los workers se encuentre disponible.

3. Demora en responder: Cada servicio demora un tiempo en responder, que puede ser igual o distinto entre ellos. Deberán obtener este valor para cada uno.

   Ejecutamos un escenario normal donde por 60 segundos hay un arrival rate de 5 requests:

Servicio	Min	Media	Max	p99
Proxy1	1408	1408.4	1437	1436.8
Proxy2	760	1022.7	1528	1525.7