# Locust Test Sonuçları

## Test senaryosu - 1

Amaç: Bucket'ı aşacak şekilde requestler atarak kullanıcının rate limit'e takıldığını görme.

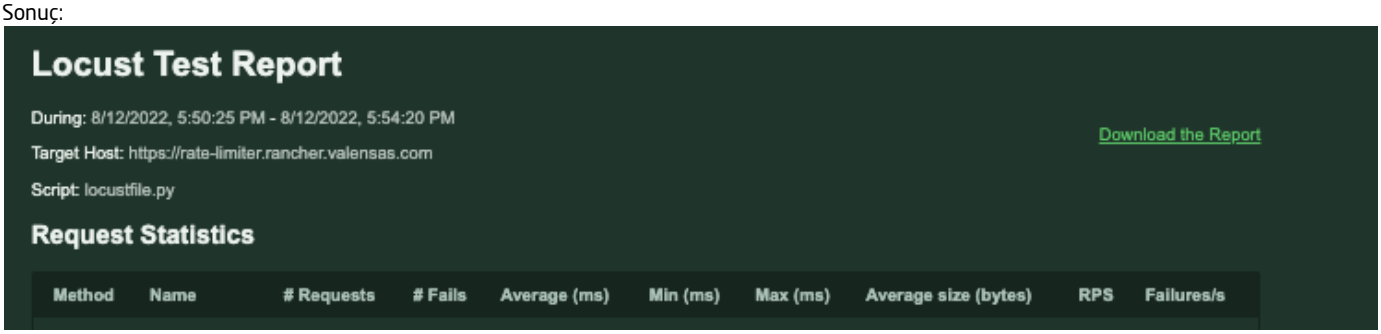
Yöntem:

- Rate limiter dakikada 10 tokena sahiptir.
- Bir get requesti 3, bir post requesti 5 token tüketir.
- Sırasıyla bir get ve bir post request atılır, sonrasında atılan get ve post requestlerin rate limitera takılması beklenir.
- Bucket'ın dolması için bir dakika beklenilir.
- Test metodu her kullanıcı random bir ip'ye sahip olacak şekilde 100 eşzamanlı kullanıcı ile çalıştırılır.

Kod:

```
from locust import task, HttpUser, constant
import logging
import random

class User(HttpUser):
    # Rate limiter have 10 token in 1 minute
    # A get request consumes 3 tokens, a post request consumes 5 tokens
    # 1 get and 1 post will be requested by the user with a random ip,
HTTP 200 will return, 8 tokens will be consumed
    # The next get and post request with the same ip will be responded
with HTTP 429, it is marked as success
    # The process will be repeated after 60 seconds
    # The code will be runned with a number of concurrent users
    wait_time = constant(60)

    @task
    def first(self):
        # self.ip
        ip = random.randint(10000,99999)

        self.client.get('/get', headers={'X-Forwarded-For': 'ip: %s' %
str(ip)})
        self.client.post('/post', headers={'X-Forwarded-For': 'ip: %s'
% str(ip)})

        # COMMENT OUT THE FOLLOWNG CODE FOR TESTING RESPONSE TIME WITH
AND WITHOUT RATE LIMITER

        with self.client.get('/get', catch_response = True, headers={'X-
Forwarded-For': 'ip: %s' % str(ip)}) as resp:
            if resp.status_code == 429:
                resp.success()
                logging.info("Rate limiter is successful")
        with self.client.post('/post', catch_response = True, headers=
{'X-Forwarded-For': 'ip: %s' % str(ip)}) as resp:
            if resp.status_code == 429:
                resp.success()
                logging.info("Rate limiter is successful")
```

Sonuç:

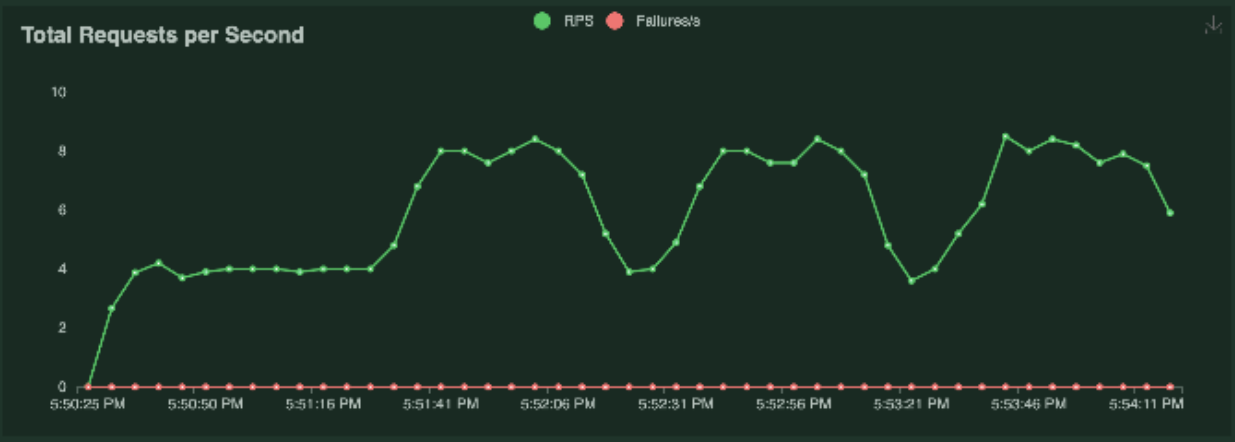


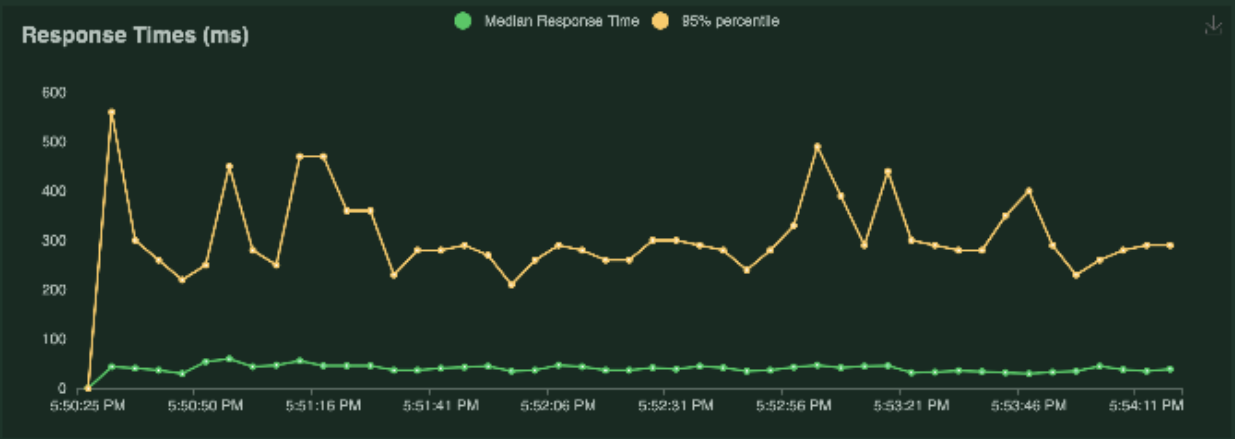
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| GET | /get | 708 | 0 | 131 | 15 | 995 | 7 | 3.0 | 0.0 |
| POST | /post | 708 | 0 | 52 | 17 | 829 | 8 | 3.0 | 0.0 |
| | **Aggregated** | **1416** | **0** | **91** | **15** | **995** | **7** | **6.0** | **0.0** |

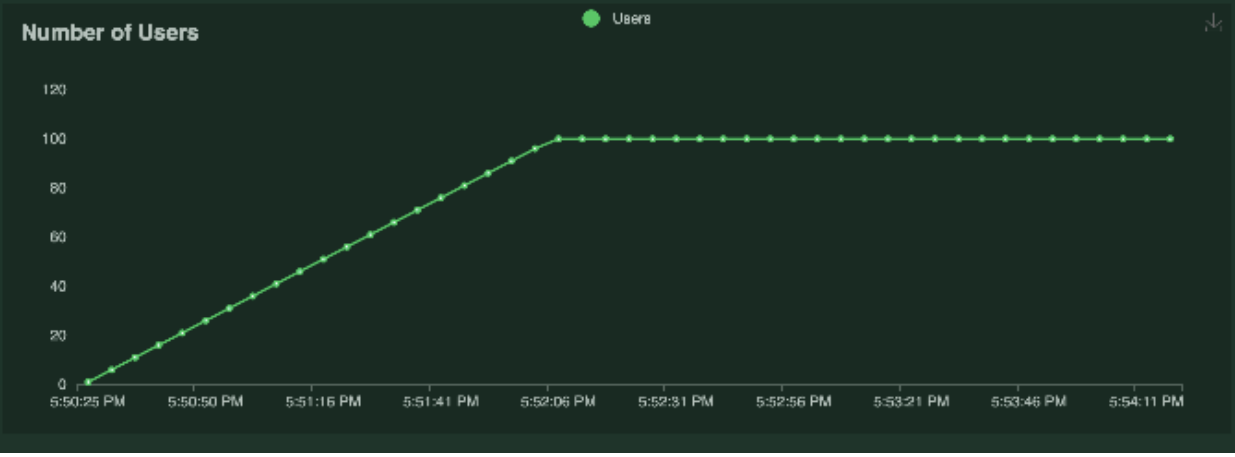
## Response Time Statistics

| Method | Name | 50%ile (ms) | 60%ile (ms) | 70%ile (ms) | 80%ile (ms) | 90%ile (ms) | 95%ile (ms) | 99%ile (ms) | 100%ile (ms) |
|---|---|---|---|---|---|---|---|---|---|
| GET | /get | 110 | 130 | 170 | 220 | 280 | 330 | 600 | 1000 |
| POST | /post | 31 | 35 | 41 | 48 | 85 | 190 | 340 | 830 |
| | **Aggregated** | **39** | **48** | **110** | **150** | **240** | **290** | **520** | **1000** |

## Charts




## Test senaryosu - 2

Amaç: Rate Limiter enabled/disabled olacak şekilde backend servise request atma.

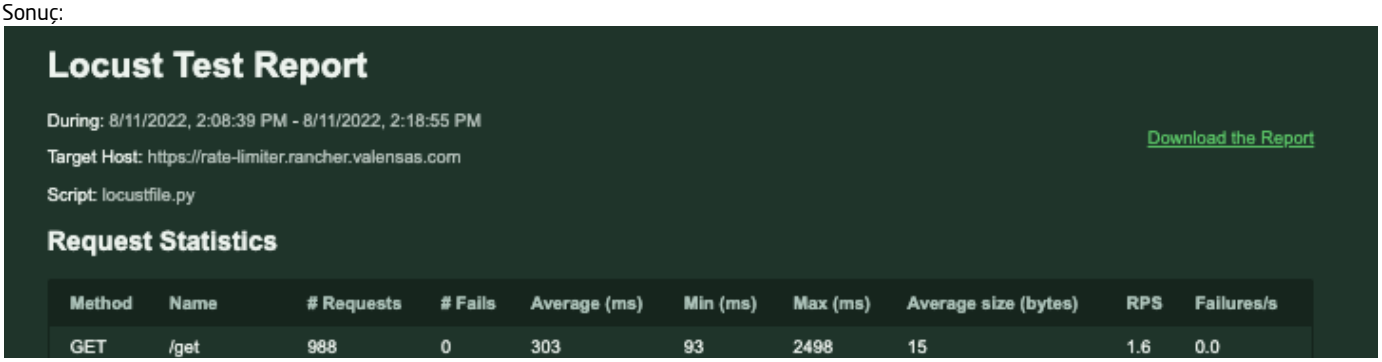
Yöntem:

- 100 eşzamanlı kullanıcı tarafından servise dakikada bir get ve bir post request atılır.
- Test kodu rate limiter varken ve yokken çalıştırılır.
- Response timelar incelenir.

Kod:

```
from locust import task, HttpUser, constant
import logging
import random

class User(HttpUser):
    # Rate limiter have 10 token in 1 minute
    # A get request consumes 3 tokens, a post request consumes 5 tokens
    # The process will be repeated after 60 seconds
    # The code will be runned with a number of concurrent users
    wait_time = constant(60)

    @task
    def first(self):
        # self.ip
        ip = random.randint(100000,999999)

        self.client.get('/get', headers={'X-Forwarded-For': 'ip: %s' %
str(ip)})
        self.client.post('/post', headers={'X-Forwarded-For': 'ip: %s'
% str(ip)})
```

Sonuç:



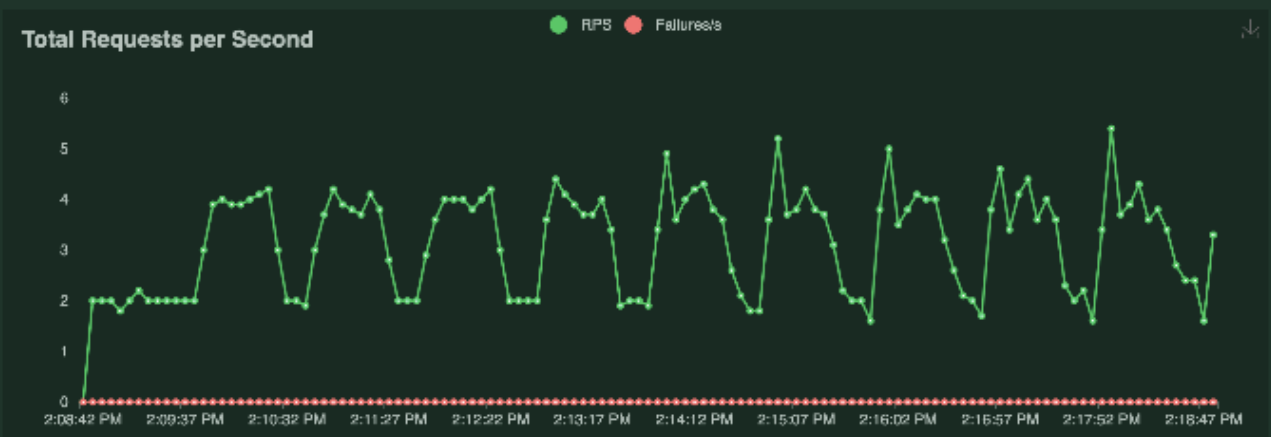
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| POST | /post | 988 | 0 | 118 | 23 | 2866 | 16 | 1.6 | 0.0 |
| | **Aggregated** | **1976** | **0** | **211** | **23** | **2866** | **15** | **3.2** | **0.0** |

## Response Time Statistics

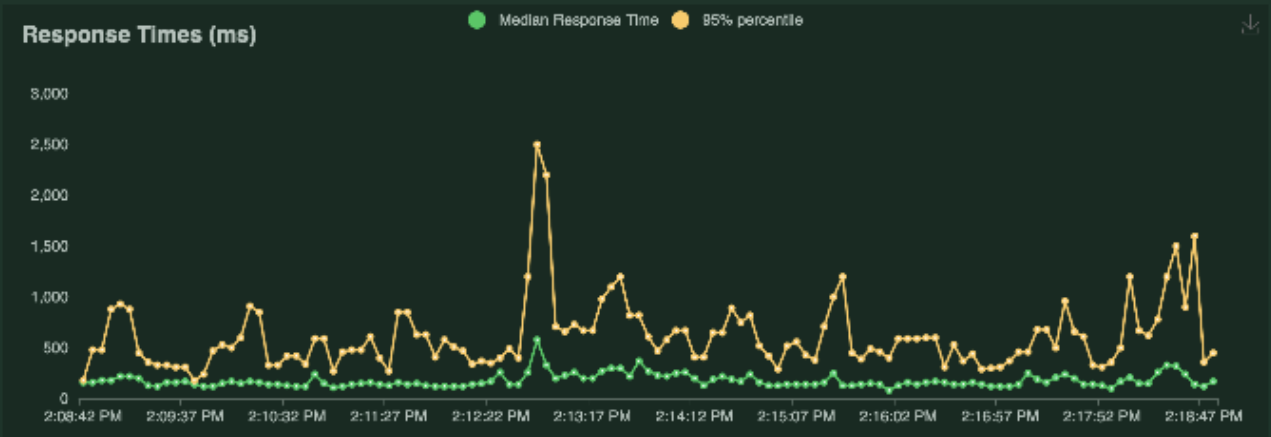
| Method | Name | 50%ile (ms) | 60%ile (ms) | 70%ile (ms) | 80%ile (ms) | 90%ile (ms) | 95%ile (ms) | 99%ile (ms) | 100%ile (ms) |
|---|---|---|---|---|---|---|---|---|---|
| GET | /get | 250 | 280 | 320 | 390 | 560 | 700 | 1200 | 2500 |
| POST | /post | 53 | 63 | 86 | 180 | 290 | 410 | 850 | 2900 |
| | **Aggregated** | **150** | **200** | **260** | **310** | **440** | **600** | **1100** | **2900** |

## Charts

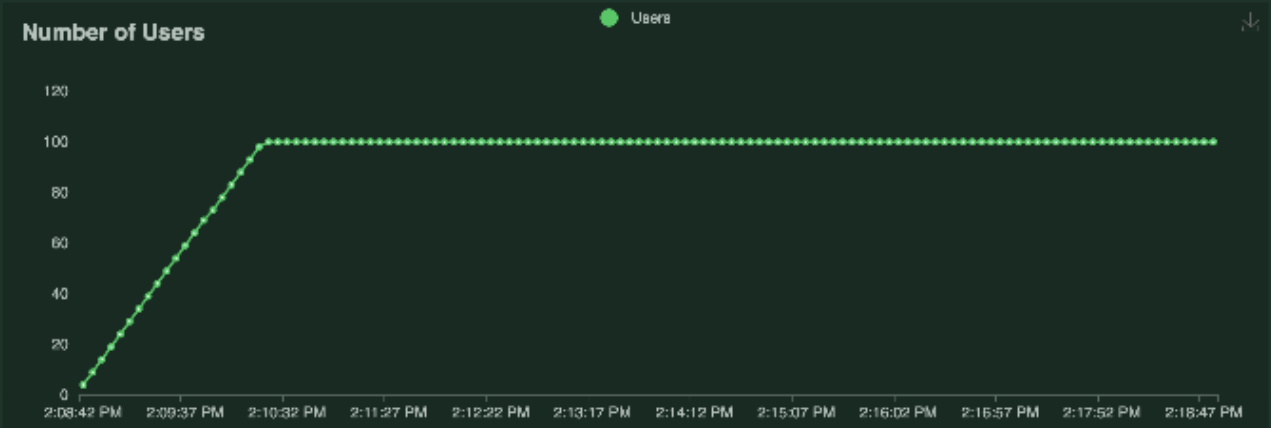
### Total Requests per Second




### Response Times (ms)




### Number of Users



## Final ratio

### Ratio per User class

- 100.0% User
  - 100.0% first

### Total ratio

- 100.0% User
  - 100.0% first

# Locust Test Report

During: 8/11/2022, 2:21:42 PM - 8/11/2022, 2:31:57 PM

Target Host: https://rate-limiter.rancher.valensas.com

Script: locustfile.py

Download the Report

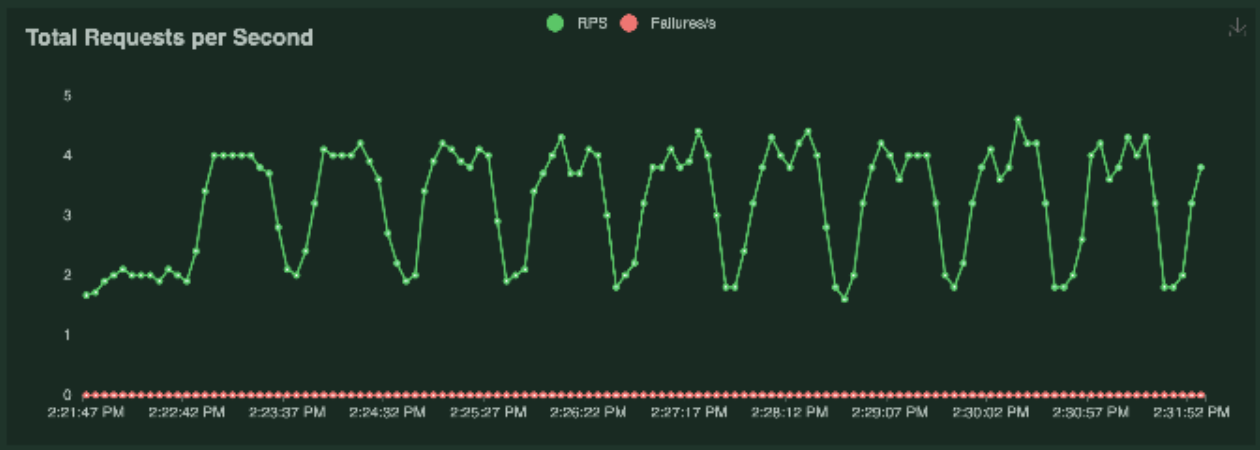
## Request Statistics

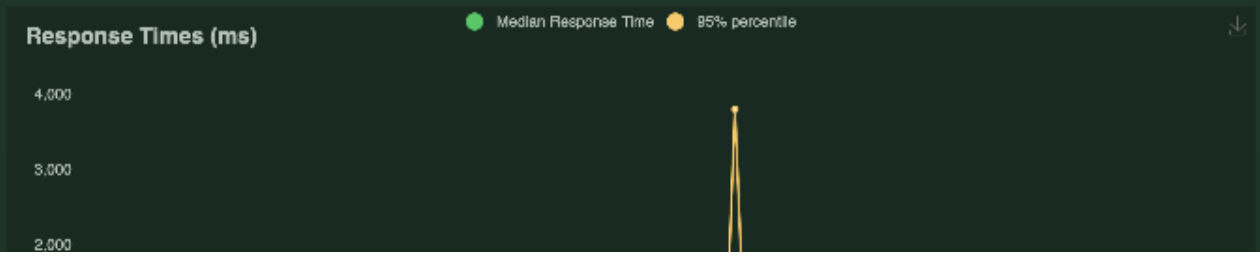
| Method | Name | # Requests | # Fails | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | RPS | Failures/s |
|---|---|---|---|---|---|---|---|---|---|
| GET | /get | 984 | 0 | 254 | 87 | 3814 | 15 | 1.6 | 0.0 |
| POST | /post | 984 | 0 | 75 | 20 | 919 | 16 | 1.6 | 0.0 |
| | **Aggregated** | **1968** | **0** | **164** | **20** | **3814** | **15** | **3.2** | **0.0** |

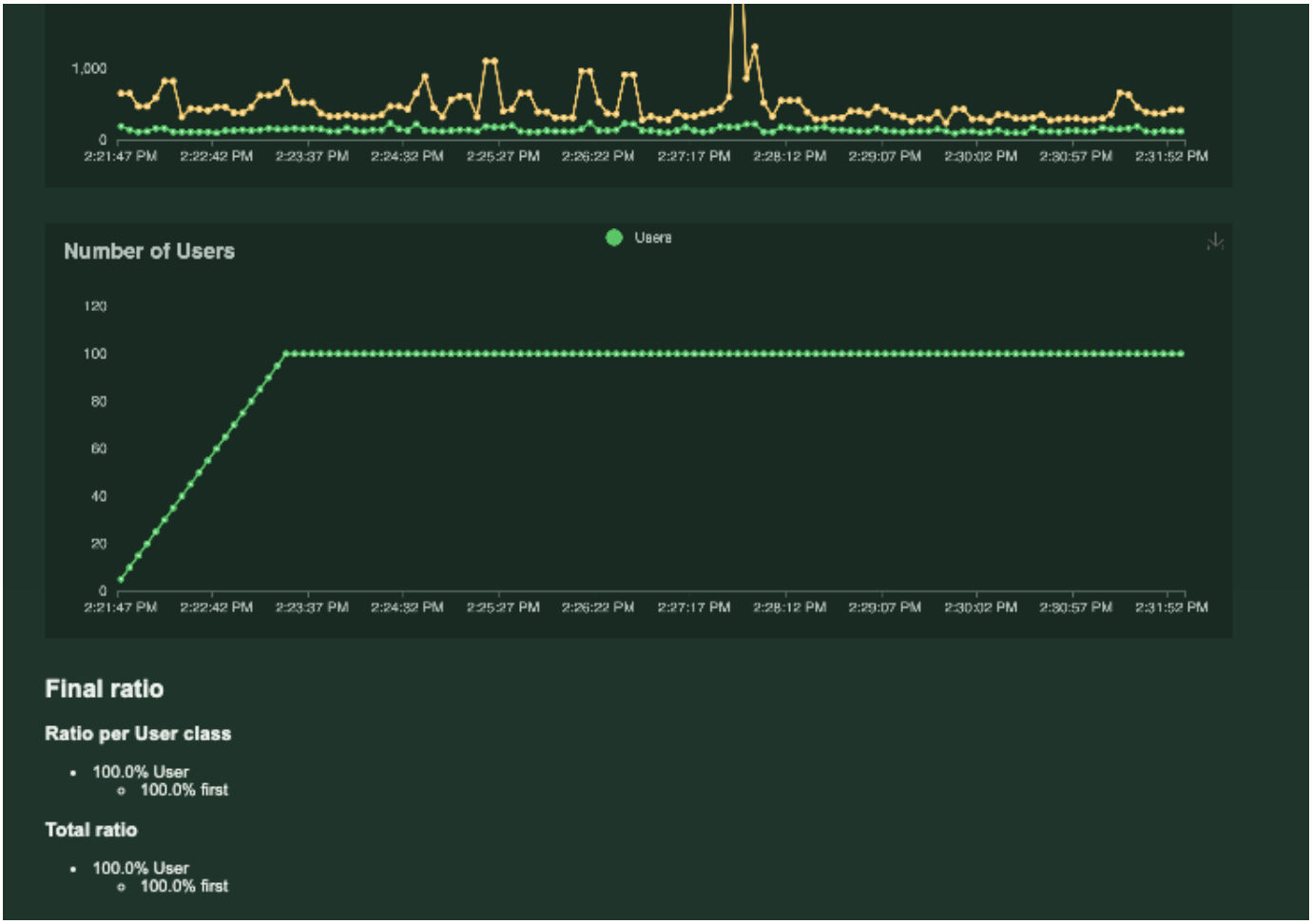
## Response Time Statistics

| Method | Name | 50%ile (ms) | 60%ile (ms) | 70%ile (ms) | 80%ile (ms) | 90%ile (ms) | 95%ile (ms) | 99%ile (ms) | 100%ile (ms) |
|---|---|---|---|---|---|---|---|---|---|
| GET | /get | 220 | 250 | 280 | 300 | 380 | 530 | 1000 | 3800 |
| POST | /post | 43 | 50 | 59 | 82 | 190 | 240 | 470 | 920 |
| | **Aggregated** | **130** | **160** | **210** | **260** | **320** | **410** | **810** | **3800** |

## Charts






## Test senaryosu - 3

Amaç: Response'a delay vererek etkisini gözlemleme.

Yöntem:

- Wiremock ile servis oluşturulur.
- Gateway wiremock'a yönlendirilir.
- Wiremock 10, 100 ve 1000 ms geciktirilerek ikinci test senaryosu ile denenir.

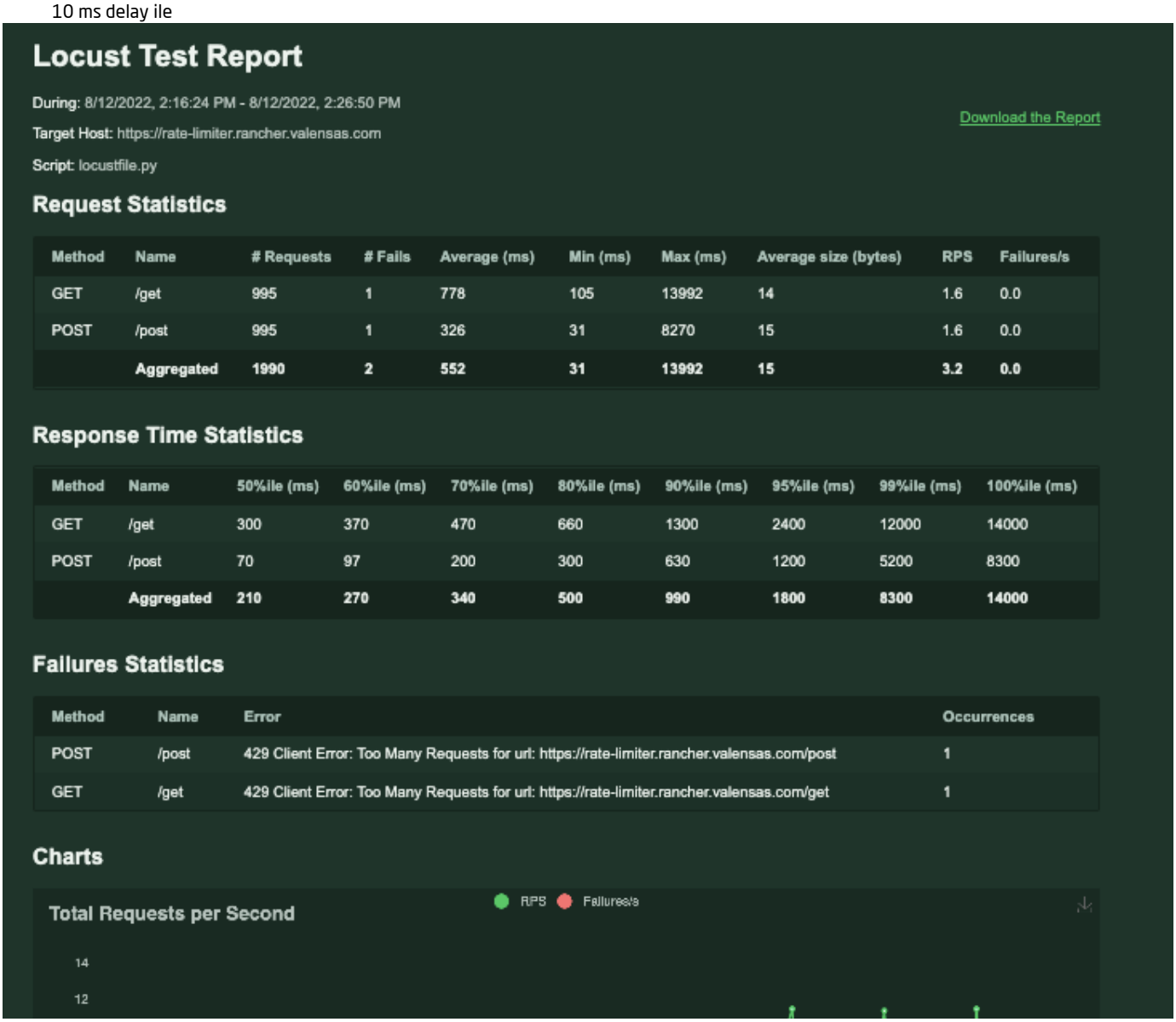
Kod:

```
{
  "id" : "e2a6cc05-7f44-403f-9c0a-63e935ecfa48",
  "name" : "get",
  "request" : {
    "url" : "/get",
    "method" : "GET"
  },
  "response" : {
    "status" : 200,
    "body": "Hello from get!",
    "fixedDelayMilliseconds": 1000
  }
}
```

```
{
  "id" : "53fba64f-45d9-4044-9f65-6712ffd75cb3",
  "name" : "get",
  "request" : {
    "url" : "/post",
    "method" : "POST"
  },
  "response" : {
    "status" : 200,
    "body": "Hello from post!",
    "fixedDelayMilliseconds": 1000
  }
}
```

Sonuç:

10 ms delay ile



# Locust Test Report

During: 8/12/2022, 2:16:24 PM - 8/12/2022, 2:26:50 PM

Target Host: https://rate-limiter.rancher.valensas.com

Script: locustfile.py

Download the Report

## Request Statistics

| Method | Name | # Requests | # Fails | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | RPS | Failures/s |
|---|---|---|---|---|---|---|---|---|---|
| GET | /get | 995 | 1 | 778 | 105 | 13992 | 14 | 1.6 | 0.0 |
| POST | /post | 995 | 1 | 326 | 31 | 8270 | 15 | 1.6 | 0.0 |
| | **Aggregated** | **1990** | **2** | **552** | **31** | **13992** | **15** | **3.2** | **0.0** |

## Response Time Statistics

| Method | Name | 50%ile (ms) | 60%ile (ms) | 70%ile (ms) | 80%ile (ms) | 90%ile (ms) | 95%ile (ms) | 99%ile (ms) | 100%ile (ms) |
|---|---|---|---|---|---|---|---|---|---|
| GET | /get | 300 | 370 | 470 | 660 | 1300 | 2400 | 12000 | 14000 |
| POST | /post | 70 | 97 | 200 | 300 | 630 | 1200 | 5200 | 8300 |
| | **Aggregated** | **210** | **270** | **340** | **500** | **990** | **1800** | **8300** | **14000** |

## Failures Statistics

| Method | Name | Error | Occurrences |
|---|---|---|---|
| POST | /post | 429 Client Error: Too Many Requests for url: https://rate-limiter.rancher.valensas.com/post | 1 |
| GET | /get | 429 Client Error: Too Many Requests for url: https://rate-limiter.rancher.valensas.com/get | 1 |

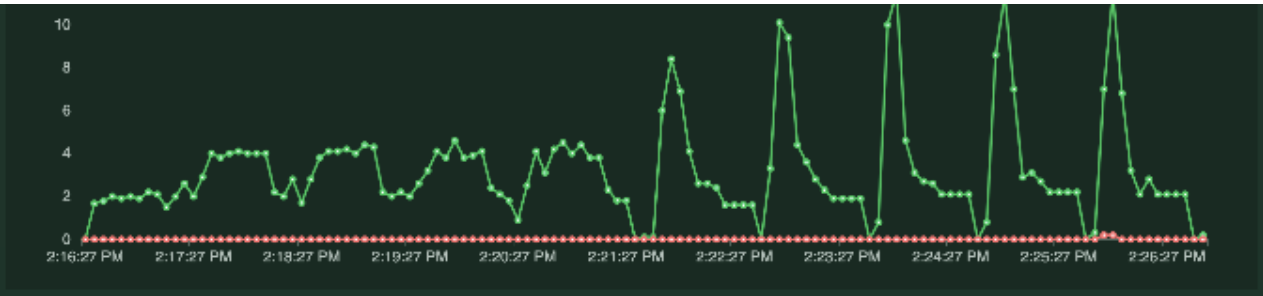
## Charts

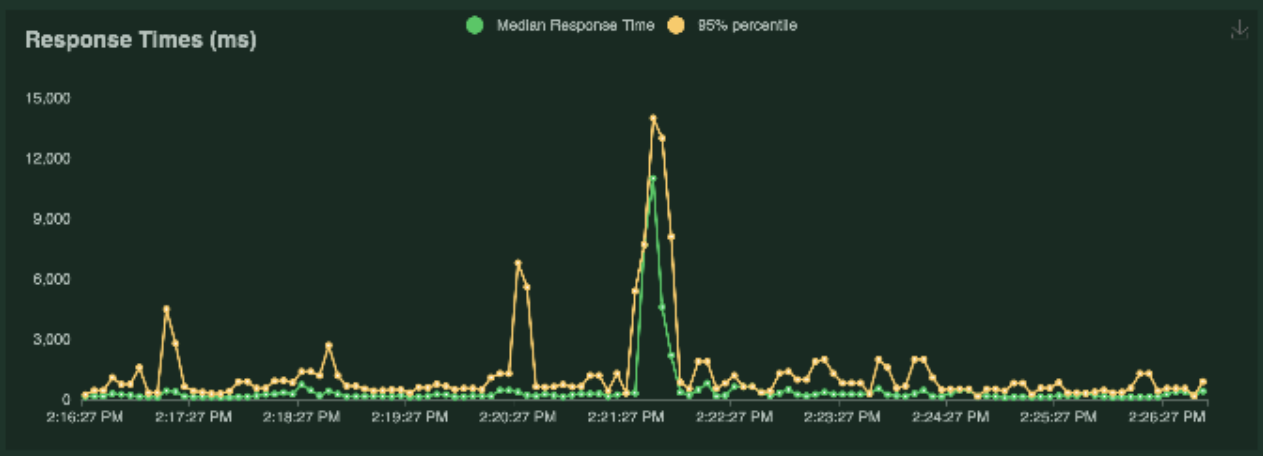
Total Requests per Second

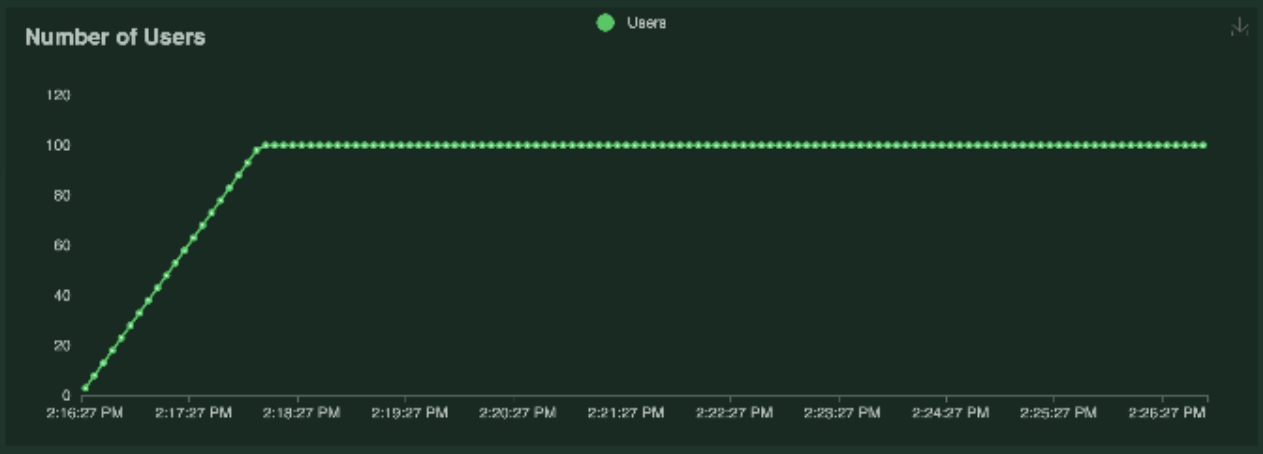
RPS Failures/s

14

12




## Final ratio

### Ratio per User class

- 100.0% User
  - 100.0% first

### Total ratio

- 100.0% User
  - 100.0% first


100 ms delay ile

# Locust Test Report

During: 8/12/2022, 3:49:06 PM - 8/12/2022, 3:55:21 PM

Target Host: https://rate-limiter.rancher.valensas.com

Script: locustfile.py

Download the Report

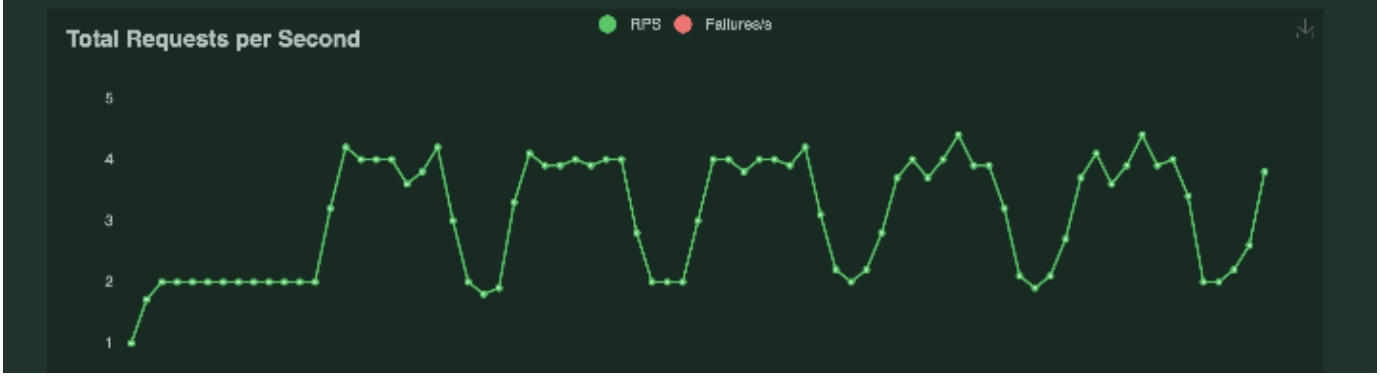
## Request Statistics

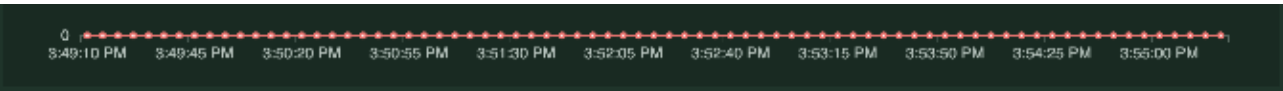
| Method | Name | # Requests | # Fails | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | RPS | Failures/s |
|---|---|---|---|---|---|---|---|---|---|
| GET | /get | 585 | 0 | 338 | 195 | 1737 | 15 | 1.6 | 0.0 |
| POST | /post | 584 | 0 | 178 | 123 | 975 | 16 | 1.6 | 0.0 |
| | **Aggregated** | **1169** | **0** | **258** | **123** | **1737** | **15** | **3.1** | **0.0** |

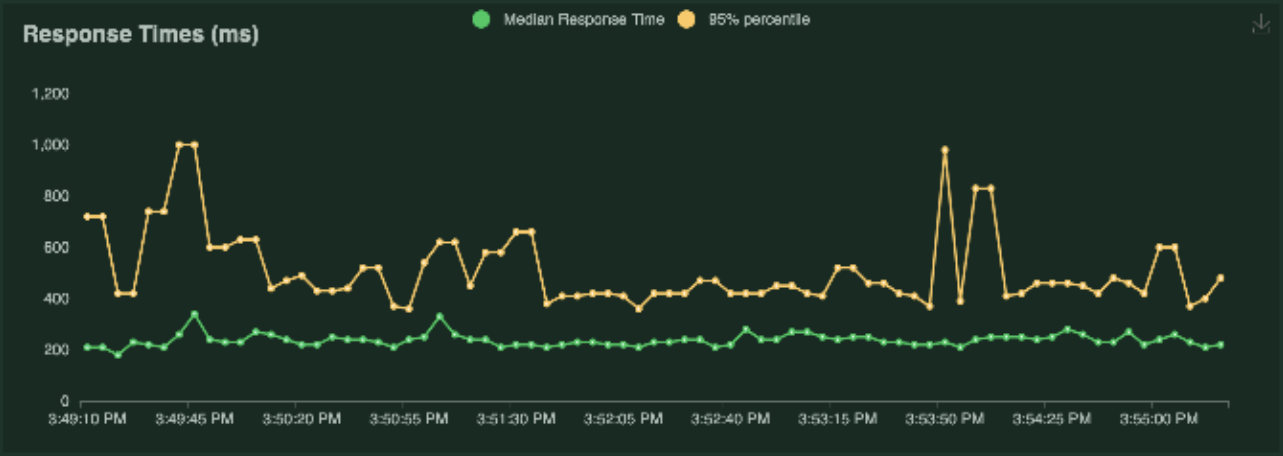
## Response Time Statistics

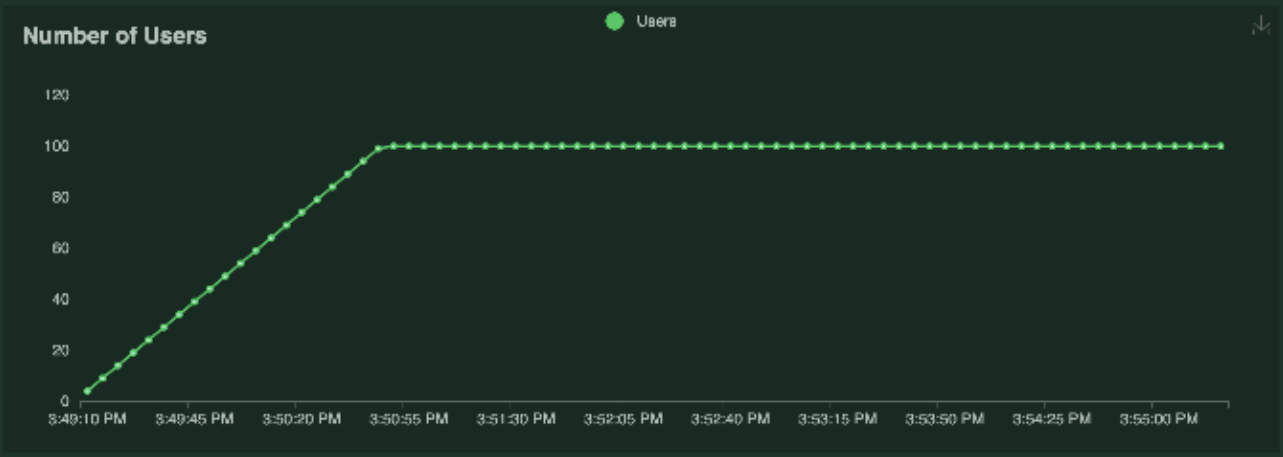
| Method | Name | 50%ile (ms) | 60%ile (ms) | 70%ile (ms) | 80%ile (ms) | 90%ile (ms) | 95%ile (ms) | 99%ile (ms) | 100%ile (ms) |
|---|---|---|---|---|---|---|---|---|---|
| GET | /get | 310 | 330 | 360 | 410 | 450 | 540 | 1000 | 1700 |
| POST | /post | 150 | 160 | 180 | 200 | 270 | 330 | 460 | 980 |
| | **Aggregated** | **230** | **260** | **300** | **340** | **410** | **450** | **740** | **1700** |

## Charts






## Final ratio

### Ratio per User class

- 100.0% User
  - 100.0% first

### Total ratio

- 100.0% User
  - 100.0% first


1000 ms delay ile

# Locust Test Report

During: 8/12/2022, 4:42:22 PM - 8/12/2022, 4:52:32 PM

Target Host: https://rate-limiter.rancher.valensas.com

Script: locustfile.py

Download the Report

## Request Statistics

| Method | Name | # Requests | # Fails | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | RPS | Failures/s |
|---|---|---|---|---|---|---|---|---|---|
| GET | /get | 949 | 1 | 1210 | 290 | 2307 | 14 | 1.6 | 0.0 |
| POST | /post | 948 | 1 | 1053 | 22 | 1650 | 15 | 1.6 | 0.0 |
| | **Aggregated** | **1897** | **2** | **1132** | **22** | **2307** | **15** | **3.1** | **0.0** |

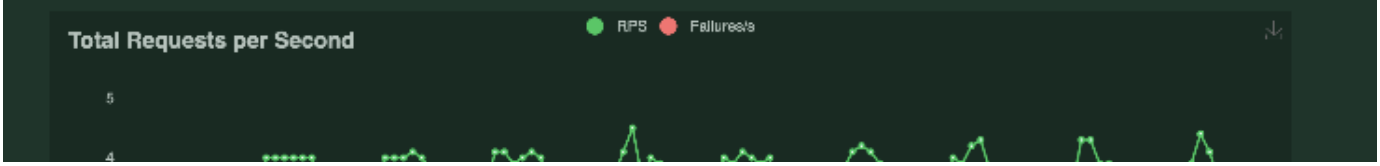
## Response Time Statistics

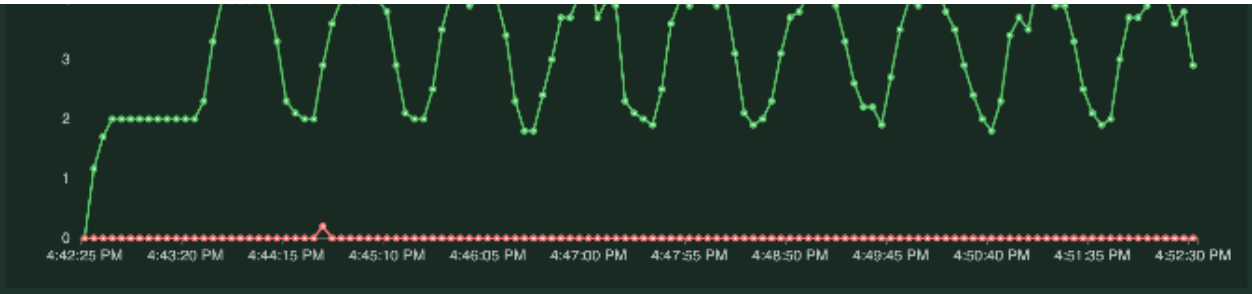
| Method | Name | 50%ile (ms) | 60%ile (ms) | 70%ile (ms) | 80%ile (ms) | 90%ile (ms) | 95%ile (ms) | 99%ile (ms) | 100%ile (ms) |
|---|---|---|---|---|---|---|---|---|---|
| GET | /get | 1200 | 1200 | 1200 | 1300 | 1300 | 1400 | 1600 | 2300 |
| POST | /post | 1000 | 1000 | 1000 | 1100 | 1100 | 1200 | 1300 | 1700 |
| | **Aggregated** | **1100** | **1100** | **1200** | **1200** | **1300** | **1300** | **1600** | **2300** |

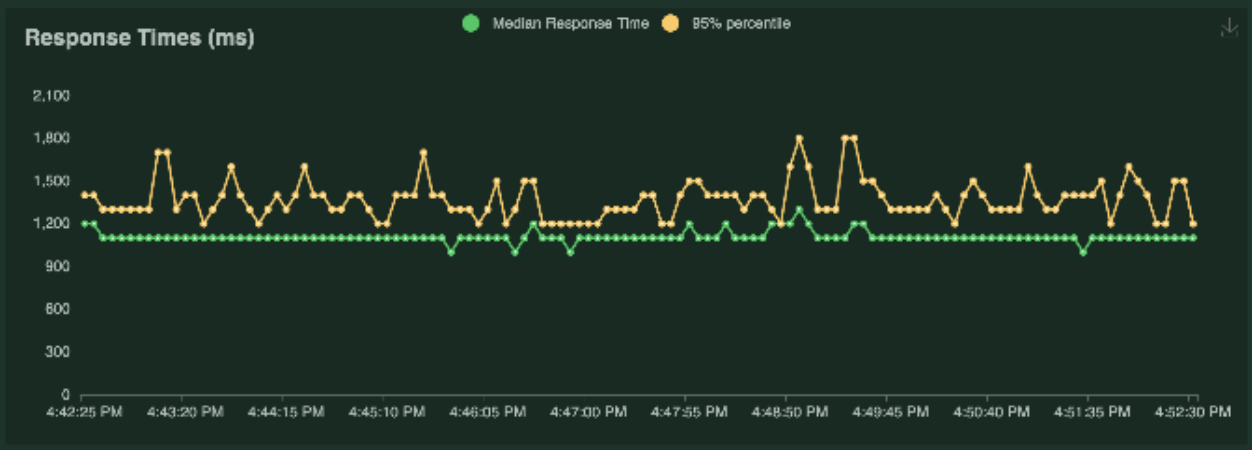
## Failures Statistics

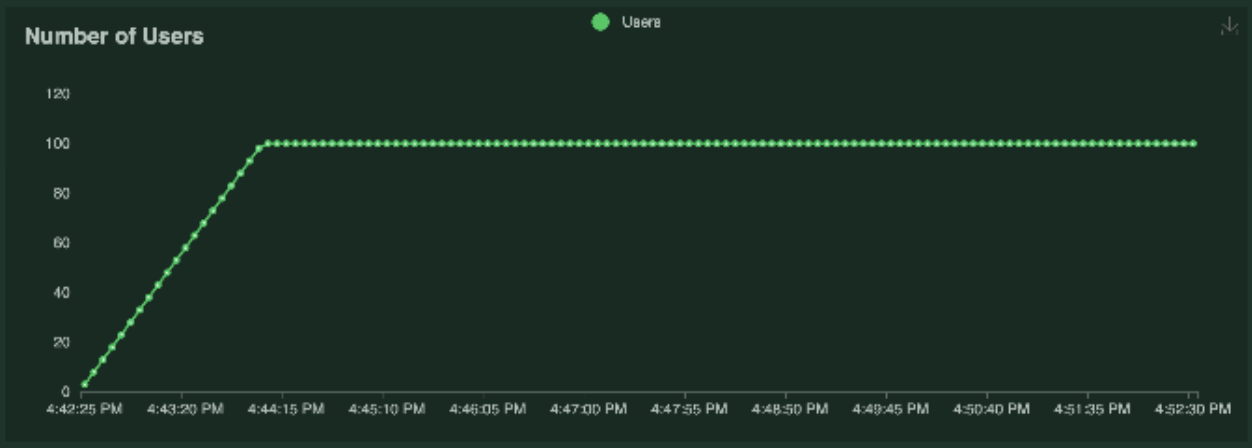
| Method | Name | Error | Occurrences |
|---|---|---|---|
| POST | /post | 429 Client Error: Too Many Requests for url: https://rate-limiter.rancher.valensas.com/post | 1 |
| GET | /get | 429 Client Error: Too Many Requests for url: https://rate-limiter.rancher.valensas.com/get | 1 |

## Charts





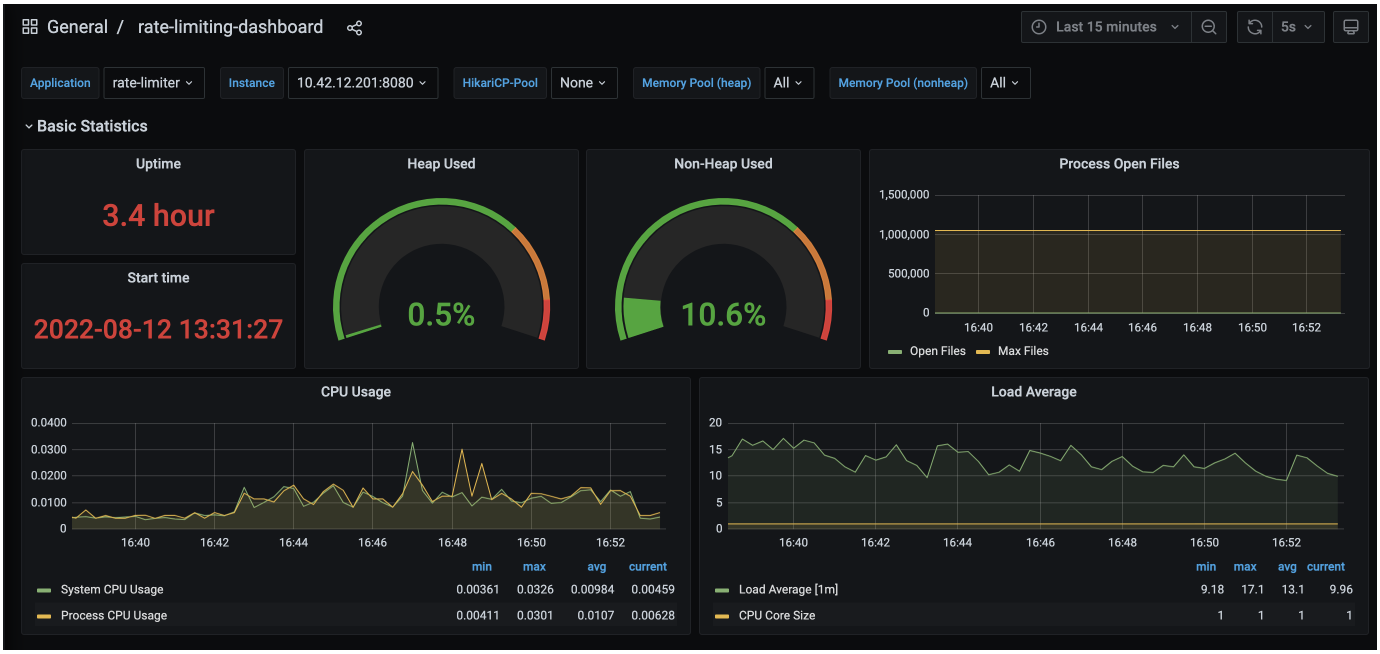

## Final ratio

### Ratio per User class

- 100.0% User
  - 100.0% first

### Total ratio

- 100.0% User
  - 100.0% first


General / rate-limiting-dashboard
Last 15 minutes
5s
Application
rate-limiter
Instance
10.42.12.201:8080
HikariCP-Pool
None
Memory Pool (heap)
All
Memory Pool (nonheap)
All
Basic Statistics
Uptime
3.4 hour
Start time
2022-08-12 13:31:27
Heap Used
0.5%
Non-Heap Used
10.6%
Process Open Files
1,500,000
1,000,000
500,000
0
16:40
16:42
16:44
16:46
16:48
16:50
16:52
Open Files
Max Files
CPU Usage
0.0400
0.0300
0.0200
0.0100
0
16:40
16:42
16:44
16:46
16:48
16:50
16:52
min
max
avg
current
System CPU Usage
0.00361
0.0326
0.00984
0.00459
Process CPU Usage
0.00411
0.0301
0.0107
0.00628
Load Average
20
15
10
5
0
16:40
16:42
16:44
16:46
16:48
16:50
16:52
min
max
avg
current
Load Average [1m]
9.18
17.1
13.1
9.96
CPU Core Size
1
1
1
1