Stress Performance Test Setup and Results

Test Setup:

The stress performance tests were designed to evaluate the response of an API under varying loads using the K6 load testing tool. The test script simulates a typical user interaction with an API that involves creating, retrieving, updating, and deleting a user profile. The test was structured with different stages to simulate increasing and decreasing levels of user load. - User waits are added to simulate

The test configuration specified four stages:

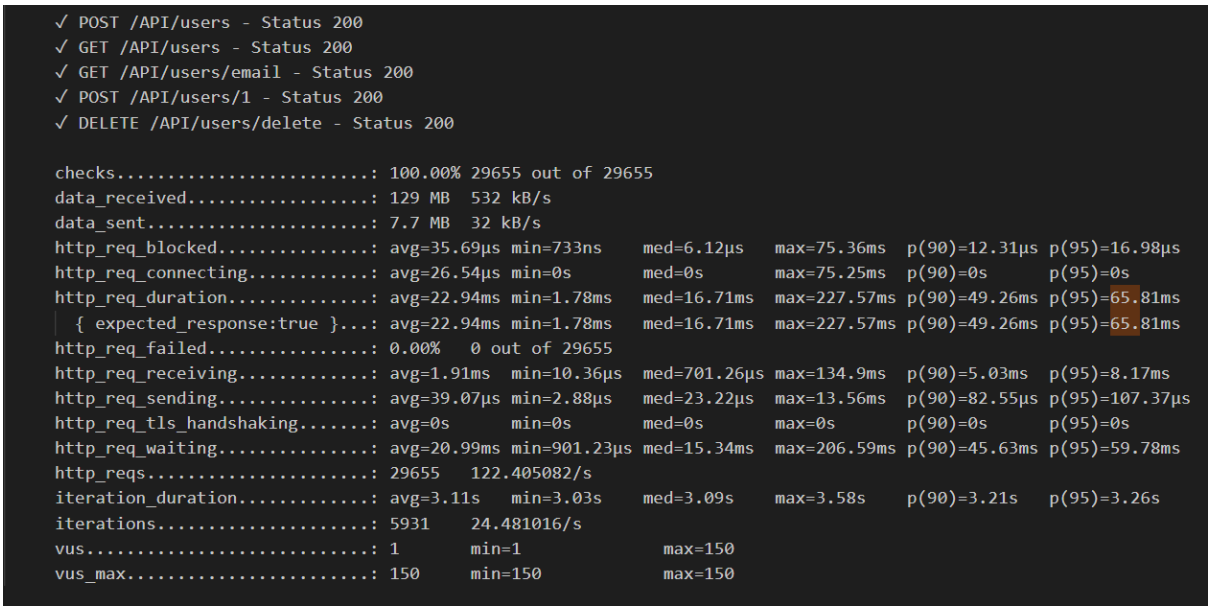
1. **Stage 1**: Ramp-up to 50 users over 1 minute.
2. **Stage 2**: Ramp-up to 100 users over the next 1 minute.
3. **Stage 3**: Ramp-up to 150 users over 1 minute.
4. **Stage 4**: Ramp-down to 10 users over 1 minute.

```
export const options = {
    stages: [
        { duration: '1m', target: 50 },
        { duration: '1m', target: 100 },
        { duration: '1m', target: 150 },
        { duration: '1m', target: 10 },
    ],
};
```

Each virtual user executed a sequence of operations:

- **POST /API/users**: Creating a user profile with a unique email address.
- **GET /API/users**: Retrieving all users.
- **GET /API/users/email**: Querying for a user by their unique email address.
- **POST /API/users/1**: Updating an existing user profile.
- **DELETE /API/users/delete**: Deleting the user by email.

Test Results:



```
✓ POST /API/users - Status 200
✓ GET /API/users - Status 200
✓ GET /API/users/email - Status 200
✓ POST /API/users/1 - Status 200
✓ DELETE /API/users/delete - Status 200

checks.....................: 100.00% 29655 out of 29655
data_received..............: 129 MB  532 kB/s
data_sent..................: 7.7 MB  32 kB/s
http_req_blocked...........: avg=35.69µs min=733ns    med=6.12µs   max=75.36ms  p(90)=12.31µs p(95)=16.98µs
http_req_connecting........: avg=26.54µs min=0s       med=0s       max=75.25ms  p(90)=0s      p(95)=0s
http_req_duration..........: avg=22.94ms min=1.78ms   med=16.71ms  max=227.57ms p(90)=49.26ms p(95)=65.81ms
  { expected_response:true }...: avg=22.94ms min=1.78ms   med=16.71ms  max=227.57ms p(90)=49.26ms p(95)=65.81ms
http_req_failed............: 0.00%   0 out of 29655
http_req_receiving.........: avg=1.91ms  min=10.36µs  med=701.26µs max=134.9ms  p(90)=5.03ms  p(95)=8.17ms
http_req_sending...........: avg=39.07µs min=2.88µs   med=23.22µs  max=13.56ms  p(90)=82.55µs p(95)=107.37µs
http_req_tls_handshaking...: avg=0s      min=0s       med=0s       max=0s       p(90)=0s      p(95)=0s
http_req_waiting...........: avg=20.99ms min=901.23µs med=15.34ms  max=206.59ms p(90)=45.63ms p(95)=59.78ms
http_reqs..................: 29655   122.405082/s
iteration_duration.........: avg=3.11s   min=3.03s    med=3.09s    max=3.58s    p(90)=3.21s   p(95)=3.26s
iterations.................: 5931    24.481016/s
vus........................: 1       min=1              max=150
vus_max....................: 150     min=150            max=150
```

The results were collected from a 4-minute run and include multiple key performance metrics such as response times, throughput, and error rates.

- **Virtual Users**: The test successfully scaled up to 150 users and was then again gradually scaled down.
- **Throughput**: The overall throughput was 122.4 requests per second, with a total of 29,655 HTTP requests made during the test.
- **Response Times**:
    - The average request duration was 22.94 milliseconds.
    - The average duration of each full test iteration was 3.11 seconds.
- **Error Rate**: The test completed without any errors. All checks passed successfully with 100% of the requests returning status 200 (OK).
- **Resource Utilization**:
- **Data Transfer**: The test transferred a total of 129 MB of data, with 7.7 MB sent to the server and 129 MB received.