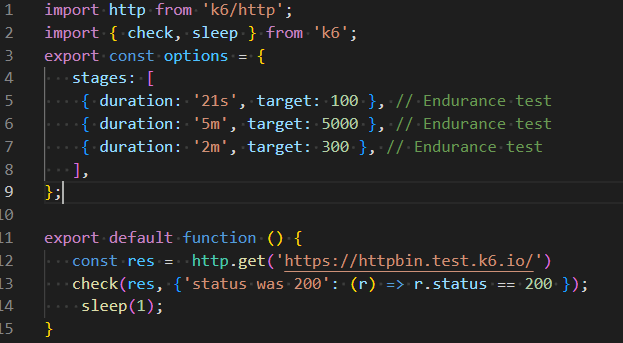
***TERHELÉSTESZTELÉS***

LOAD: Vizsgáljuk, hogy hogyan fut a program egy megadott aktív felhasználószámmal megadott idő alatt.

ENDURANCE: A LOAD-ra alapuló terhelésteszt metódus, vizsgálja, hogy hogyan fut a program hosszabb időn keresztül nagy stressz alatt.

✓ status was 200

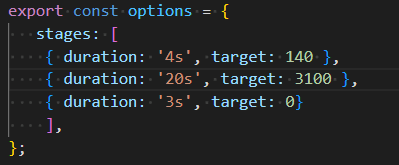
checks.........................: 100.00% ✓ 58804 ✗ 0

data\_received..................: 584 MB 1.6 MB/s

data\_sent......................: 6.5 MB 18 kB/s

running (6m01.2s), 000/500 VUs, 58804 complete and 0 interrupted iterations

default ✓ [ 100% ] 000/500 VUs 6m0s

STRESS: A programot magas kapacitáson futtatjuk, keressük a maximumot, amit el tud bírni.

✗ status was 200

↳ 98% — ✓ 8932 / ✗ 99

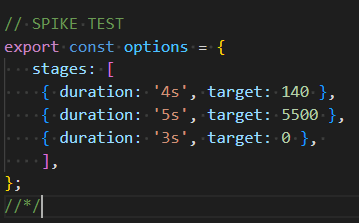
checks.........................: 98.90% ✓ 8932 ✗ 99

data\_received..................: 105 MB 2.8 MB/s

data\_sent......................: 2.1 MB 55 kB/s

running (37.7s), 0000/3100 VUs, 9031 complete and 0 interrupted iterations

default ✓ [ 100% ] 0000/3100 VUs 27s

SPIKE: A programot rövid idő alatt maximum kapacitáson futtatjuk.

✗ status was 200

↳ 99% — ✓ 6918 / ✗ 18

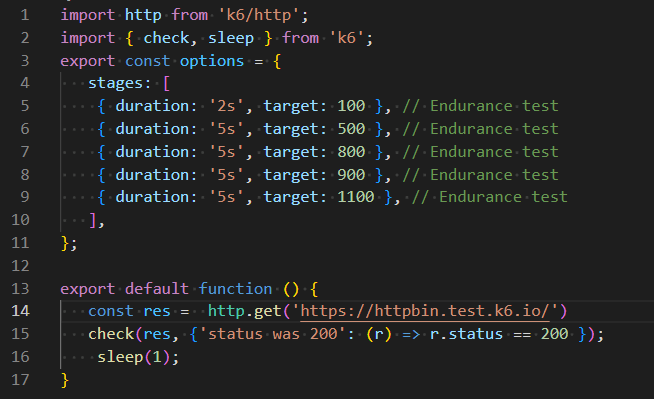
checks.........................: 99.74% ✓ 6918 ✗ 18

data\_received..................: 98 MB 3.0 MB/s

data\_sent......................: 2.6 MB 81 kB/s

running (32.5s), 0000/5500 VUs, 6936 complete and 0 interrupted iterations

default ✓ [ 100% ] 0000/5500 VUs 12s

CAPACITY: Keressük azt, hogy a program milyen felhasználómennyiségnél kezd el problémákba ütközni.

✓ status was 200

checks.........................: 100.00% ✓ 6318 ✗ 0

data\_received..................: 68 MB 2.6 MB/s

data\_sent......................: 1.1 MB 40 kB/s

running (26.3s), 0000/1100 VUs, 6318 complete and 0 interrupted iterations

default ✓ [ 100% ] 0000/1100 VUs 22s

A teszteket a Docker segítségével hajtom végre egy JavaScript programon.

A cat Chase.js | docker run --rm -i grafana/k6 run – parancs futtatásával a terminálban lefuttatom az aktív kódot a JavaScriptben.

import http from 'k6/http';

import { check, sleep } from 'k6';

export const options = {

stages: [

{ duration: '15s', target: 100 },

{ duration: '10s', target: 50000 },

{ duration: '25s', target: 150 },

],

};

export default function () {

const res = http.get('https://httpbin.test.k6.io/')

check(res, { 'status was 200': (r) => r.status == 200 });

http.get('http://test.k6.io');

sleep(1);

}

✓ status was 200

checks.........................: 100.00% ✓ 58804 ✗ 0

data\_received..................: 584 MB 1.6 MB/s

data\_sent......................: 6.5 MB 18 kB/s

running (6m01.2s), 000/500 VUs, 58804 complete and 0 interrupted iterations

default ✓ [ 100% ] 000/500 VUs 6m0s