

Four major approaches in load testing:

- Spike
- Stress
- Load
- Soak

Install and import K6

```
import http from 'k6/http';
```

```
export let options = {  
  insecureSkipTLSVerify:true,  
  noConnectionReuse: false,  
  vus: 10,  
  duration: '30s',  
};  
export default function () {  
  http.get('http://test.k6.io');  
}
```


----- sadrzi sve opcije za same testove

----- virtualni korisnici

----- vremenski period

----- krajnja tacka koju zelimo da testiramo,
svaki virtualni korisnik izvršava svaki put

```
C:\Users\Selma\Desktop\JS>k6 run hello.js
```



```
execution: local  
script: hello.js  
output: -  
  
scenarios: (100.00%) 1 scenario, 10 max VUs, 1m0s max duration (incl. graceful stop):  
  * default: 10 looping VUs for 30s (gracefulStop: 30s)  
  
running (0m30.3s), 00/10 VUs, 778 complete and 0 interrupted iterations  
default [=====] 10 VUs 30s
```

| | | | | | | | |
|-------------------------------|--------------|--------------|--------------|--------------|----------------|----------------|--|
| data_received..... | 9.3 MB | 306 kB/s | | | | | |
| data_sent..... | 161 kB | 5.3 kB/s | | | | | |
| http_req_blocked..... | avg=16.95ms | min=0s | med=0s | max=6.21s | p(90)=0s | p(95)=0s | |
| http_req_connecting..... | avg=8.52ms | min=0s | med=0s | max=1.54s | p(90)=0s | p(95)=0s | |
| http_req_duration..... | avg=175.75ms | min=110.9ms | med=139.43ms | max=4.66s | p(90)=222.36ms | p(95)=304.65ms | |
| { expected_response:true }... | avg=175.75ms | min=110.9ms | med=139.43ms | max=4.66s | p(90)=222.36ms | p(95)=304.65ms | |
| http_req_failed..... | 0.00% | 0 | 0 | 1556 | | | |
| http_req_receiving..... | avg=1.56ms | min=0s | med=0s | max=165.24ms | p(90)=5.29ms | p(95)=9.25ms | |
| http_req_sending..... | avg=55.15µs | min=0s | med=0s | max=13.24ms | p(90)=0s | p(95)=0s | |
| http_req_tls_handshaking..... | avg=5.67ms | min=0s | med=0s | max=4.66s | p(90)=0s | p(95)=0s | |
| http_req_waiting..... | avg=174.14ms | min=108.58ms | med=138.05ms | max=4.66s | p(90)=221.84ms | p(95)=304.2ms | |
| http_reqs..... | 1556 | 51.38168/s | | | | | |
| iteration_duration..... | avg=388.06ms | min=251.56ms | med=286.12ms | max=7.51s | p(90)=459.27ms | p(95)=560.17ms | |
| iterations..... | 778 | 25.69084/s | | | | | |
| vus..... | 10 | min=10 | max=10 | | | | |
| vus_max..... | 10 | min=10 | max=10 | | | | |

```

import http from 'k6/http';
import { sleep } from 'k6';    ----- pauza nakon pozivanje metode za svakog vus
export let options = {
  insecureSkipTLSVerify:true,
  noConnectionReuse: false,
  vus: 10,
  duration: '10s',
};
export default function () {
  http.get('http://test.k6.io');
  sleep(1);    ----- sleep 1s, svaki vus poziva API u jednoj sekundi,
               pogodan za testiranje koliko zahtjeva u sekundi moze podnijeti
}

```

```

C:\Users\Selma\Desktop\JS>k6 run hello.js

  AKG
  .io

execution: local
script: hello.js
output: -

scenarios: (100.00%) 1 scenario, 10 max VUs, 40s max duration (incl. graceful stop):
  * default: 10 looping VUs for 10s (gracefulStop: 30s)

running (10.7s), 00/10 VUs, 80 complete and 0 interrupted iterations
default [=====] 10 VUs 10s

data_received.....: 1.0 MB 94 kB/s
data_sent.....: 20 kB 1.8 kB/s
http_req_blocked.....: avg=27.8ms min=0s med=0s max=312.5ms p(90)=161.74ms p(95)=260.75ms
http_req_connecting.....: avg=16.17ms min=0s med=0s max=137.84ms p(90)=125.82ms p(95)=128.77ms
http_req_duration.....: avg=133.83ms min=122.79ms med=131.12ms max=245.72ms p(90)=144.67ms p(95)=151.34ms
  { expected_response:true }...: avg=133.83ms min=122.79ms med=131.12ms max=245.72ms p(90)=144.67ms p(95)=151.34ms
http_req_failed.....: 0.00% 0 160
http_req_receiving.....: avg=1.03ms min=0s med=0s max=122.37ms p(90)=1ms p(95)=1.08ms
http_req_sending.....: avg=35.99µs min=0s med=0s max=1ms p(90)=0s p(95)=119.07µs
http_req_tls_handshaking.....: avg=9.54ms min=0s med=0s max=174.64ms p(90)=0s p(95)=132.63ms
http_req_waiting.....: avg=132.76ms min=122.79ms med=130.53ms max=167.86ms p(90)=144.12ms p(95)=149.25ms
http_reqs.....: 160 14.919463/s
iteration_duration.....: avg=1.33s min=1.25s med=1.27s max=1.76s p(90)=1.7s p(95)=1.75s
iterations.....: 80 7.459732/s
vus.....: 10 min=10 max=10
vus_max.....: 10 min=10 max=10

```

Stress

- Koristi se za određivanje granica sistema (stability and reliability)
- Svrha – testiranje stabilnosti i pouzdanosti kada je sistem u ekstremnim uvjetima, maksimalan capacity, odrediti the breaking point i failure mode, da li će se sistem oporaviti without manual intervention nakon stress testiranja

```
import http from 'k6/http';
import { sleep } from 'k6';
export let options = {
  insecureSkipTLSVerify:true,
  noConnectionReuse: false,
  stages: [
    { duration: '2m', target: 100 }, // below normal load
    { duration: '5m', target: 100 },
    { duration: '2m', target: 200 }, // normal load
    { duration: '5m', target: 200 },
    { duration: '2m', target: 300 }, // around the breaking point
    { duration: '5m', target: 300 },
    { duration: '2m', target: 400 }, // beyond the breaking point
    { duration: '5m', target: 400 },
    { duration: '10m', target: 0 }, // scale down. Recovery stage
  ],
  --- duration: vremenski period u kojem se raspodjele targeti
  target: cilj n-usera
};
export default function () {
  const BASE_URL = 'https://test-api.k6.io';
  const responses = http.batch([
    ['GET', `${BASE_URL}/public/crocodiles/1/`],
    ['GET', `${BASE_URL}/public/crocodiles/2/`],
    ['GET', `${BASE_URL}/public/crocodiles/3/`],
    ['GET', `${BASE_URL}/public/crocodiles/4/`],
  ]);

  sleep(1);
}
```

Spike

- varijacija stress testinga, umjesto da se postepeno poveca, vrlo brzo dode do velikih brojeva
- 4 kategorije Koliko je sistem bio dobar : Excellent, Good, Poor, Bad.

```
import http from 'k6/http';
import { sleep } from 'k6';

export const options = {
  stages: [
    { duration: '10s', target: 100 }, // below normal load
    { duration: '1m', target: 100 },   --- warming up
    { duration: '10s', target: 1400 }, // spike to 1400 users
    { duration: '3m', target: 1400 }, // stay at 1400 for 3 minutes
    { duration: '10s', target: 100 }, // scale down. Recovery stage
    { duration: '3m', target: 100 },
    { duration: '10s', target: 0 },
  ],
};

export default function () {
  const BASE_URL = 'https://test-api.k6.io';

  const responses = http.batch([
    ['GET', `${BASE_URL}/public/crocodiles/1/`],
    ['GET', `${BASE_URL}/public/crocodiles/2/`],
    ['GET', `${BASE_URL}/public/crocodiles/3/`],
    ['GET', `${BASE_URL}/public/crocodiles/4/`],
  ]);

  sleep(1);
}
```

Load

- Procjena trenunih performansi, u smislu istovremenih korisnika ili zahtjeva u sekundi.
- Procijena trenutnih performansi sistema pod tipičnim i peak opterećenjem.
- Standardi performansi ostaju dok unosimo promjene sistem

```
import http from 'k6/http';
import { sleep } from 'k6';

export const options = {
  stages: [
    { duration: '5m', target: 100 }, // simulate ramp-up of traffic from
    1 to 100 users over 5 minutes.
    { duration: '10m', target: 100 }, // stay at 100 users for 10 minutes
    { duration: '5m', target: 0 }, // ramp-down to 0 users
  ],
  thresholds: {
    'http_req_duration': ['p(99)<1500'], // 99% of requests must complete
    below 1.5s
  },
};

----- thresholds - provjere tokom testa, kako bi utvrdili da li odgovara
nekim kriterijima

export default function () {
  let response = http.get('http://test.k6.io');
  sleep(1);
}
```

Soak

- Svrha – da potvrdi pouzdanost Sistema tokom duzeg vremenskog perioda, provjera curenja memorije, da očekivana ponovna pokretanja aplikacije ne gube zahtjeve
- How to run: pomocu stress odrediti max amount of users, get 75-80% , set vues to that value, run in 3 stages

```
import http from 'k6/http';
import { sleep } from 'k6';

export const options = {
  stages: [
    { duration: '2m', target: 400 }, // ramp up to 400 users
    { duration: '3h56m', target: 400 }, // stay at 400 for ~4 hours
    { duration: '2m', target: 0 }, // scale down. (optional)
  ],
};

const API_BASE_URL = 'https://test-api.k6.io';

export default function () {
  http.batch([
    ['GET', `${API_BASE_URL}/public/crocodiles/1/`],
    ['GET', `${API_BASE_URL}/public/crocodiles/2/`],
    ['GET', `${API_BASE_URL}/public/crocodiles/3/`],
    ['GET', `${API_BASE_URL}/public/crocodiles/4/`],
  ]);

  sleep(1);
}
```