

МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий
Кафедра Информатики и информационных технологий

направление подготовки

09.03.02 «Информационные системы и технологии»

ЛАБОРАТОРНАЯ РАБОТА № 8

Дисциплина: Тестирование программного обеспечения

Тема: Основы нагрузочного тестирования с k6

Выполнил(а): студент(ка) группы: 221-3711

До Дык Зунг

(Фамилия И.О.)

Дата, подпись _____
(Дата) (Подпись)

Проверил: _____
(Фамилия И.О., степень, звание) (Оценка)

Дата, подпись _____
(Дата) (Подпись)

Замечания: _____

Москва

2025

Задание (Автоматическое тестирование):

1. Установите и настройте k6.

```
PS C:\WINDOWS\system32> winget install k6 --source winget
Found k6 [GrafanaLabs.k6] Version 1.4.2
This application is licensed to you by its owner.
Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.
Downloading https://github.com/grafana/k6/releases/download/v1.4.2/k6-v1.4.2-windows-amd64.msi
30.0 MB / 30.0 MB
Successfully verified installer hash
Starting package install...
Successfully installed
PS C:\WINDOWS\system32>
```

```
Command Prompt
Microsoft Windows [Version 10.0.26200.7462]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>k6 version
k6.exe v1.4.2 (commit/5b725e8a6a, go1.25.4, windows/amd64)
```

2. Напишите скрипт, который создает виртуальных пользователей (VUs), отправляющих GET-запрос на выбранный endpoint.

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

const BASE_URL = __ENV.BASE_URL || 'http://localhost:8000/';

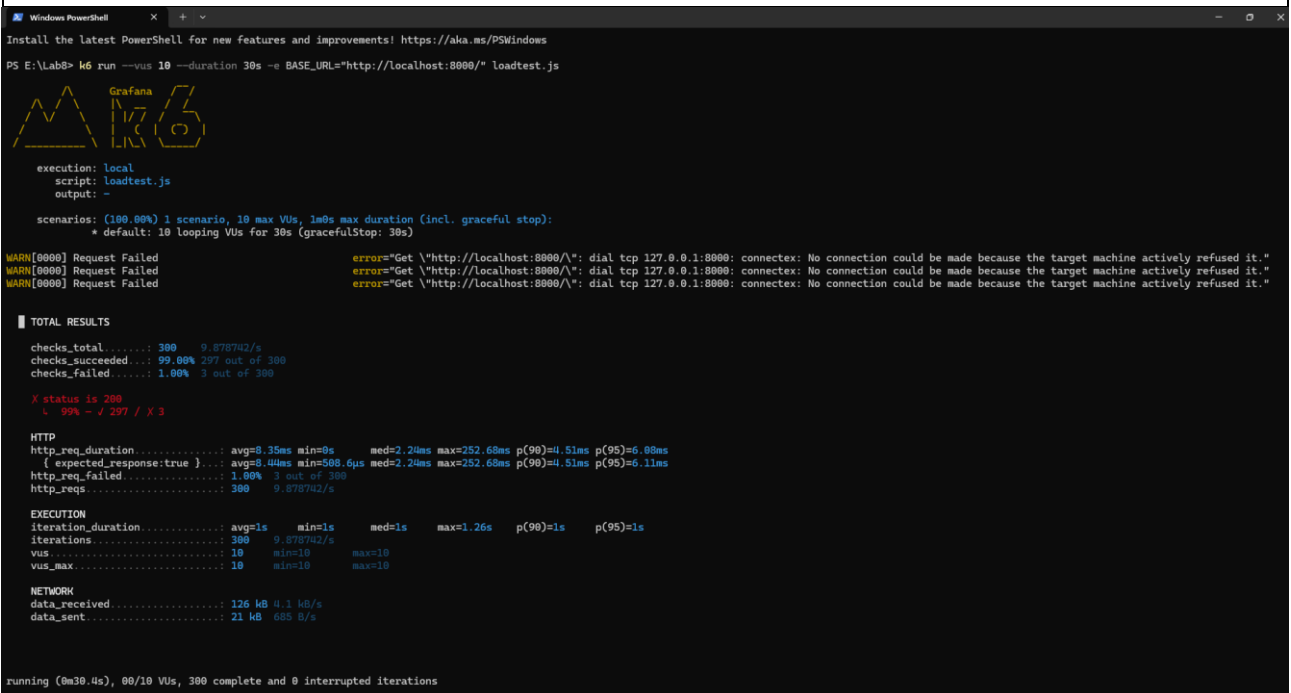
export default function () {
  const res = http.get(BASE_URL);

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

  sleep(1);
}
```

3. Запустите тест с разными параметрами нагрузки (например, 10 VUs на 30s и 50 VUs на 10s).

```
k6 run --vus 10 --duration 30s -e BASE_URL="http://localhost:8000/" loadtest.js
```



```
Windows PowerShell
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS E:\Lab> k6 run --vus 10 --duration 30s -e BASE_URL="http://localhost:8000/" loadtest.js

      M      Grafana
     /  \    /  \
    /    \  /    \
   /      \/      \
  /        \        \
 /          \          \
/            \            \

execution: local
script: loadtest.js
output: -

scenarios: (100.00%) 1 scenario, 10 max VUs, 1m0s max duration (incl. graceful stop):
  * default: 10 looping VUs for 30s (gracefulStop: 30s)

WARN[9000] Request Failed          error="Get \"http://localhost:8000/\": dial tcp 127.0.0.1:8000: connect: No connection could be made because the target machine actively refused it."
WARN[9000] Request Failed          error="Get \"http://localhost:8000/\": dial tcp 127.0.0.1:8000: connect: No connection could be made because the target machine actively refused it."
WARN[9000] Request Failed          error="Get \"http://localhost:8000/\": dial tcp 127.0.0.1:8000: connect: No connection could be made because the target machine actively refused it."

■ TOTAL RESULTS

checks_total.....: 300    9.878742/s
checks_succeeded...: 99.00% 297 out of 300
checks_failed.....: 1.00%  3 out of 300

X status is 200
  99% - 4 297 / X 3

HTTP
http_req_duration.....: avg=8.35ms min=0s    med=2.24ms max=252.68ms p(90)=4.51ms p(95)=6.88ms
{ expected_response:true }...: avg=8.44ms min=508.6µs med=2.24ms max=252.68ms p(90)=4.51ms p(95)=6.11ms
http_req_failed.....: 1.00%  3 out of 300
http_reqs.....: 300    9.878742/s

EXECUTION
iteration_duration.....: avg=1s    min=1s    med=1s    max=1.26s  p(90)=1s  p(95)=1s
iterations.....: 300    9.878742/s
vus.....: 10    min=10    max=10
vus_max.....: 10    min=10    max=10

NETWORK
data_received.....: 126 kB  4.1 kB/s
data_sent.....: 21 kB  685 B/s

running (0m30.4s), 00/10 VUs, 300 complete and 0 interrupted iterations
```

TOTAL RESULTS

checks_total.....: 300 9.96149/s

checks_succeeded...: 100.00% 300 out of 300

checks_failed.....: 0.00% 0 out of 300

“ status is 200

HTTP

http_req_duration.....: avg=2.65ms min=510.2µs med=2.28ms
max=10.34ms p(90)=4.19ms p(95)=5.17ms

{ expected_response:true }...: avg=2.65ms min=510.2µs med=2.28ms
max=10.34ms p(90)=4.19ms p(95)=5.17ms

http_req_failed.....: 0.00% 0 out of 300

http_reqs.....: 300 9.96149/s

EXECUTION

```

iteration_duration.....: avg=1s    min=1s    med=1s    max=1.01s
p(90)=1s    p(95)=1s
iterations.....: 300    9.96149/s
vus.....: 10    min=10    max=10
vus_max.....: 10    min=10    max=10

```

NETWORK

```
data_received.....: 127 kB 4.2 kB/s
data sent.....: 21 kB 697 B/s
```

running (0m30.1s), 00/10 VUs, 300 complete and 0 interrupted iterations
default â€œ [100%] 10 VUs 30s

```
k6 run --vus 50 --duration 10s -e BASE_URL="http://localhost:8000/" loadtest.js
```

```
Windows PowerShell
```

```
MARN[0096] Request Failed      error=Get "http://localhost:8090/\": dial tcp 127.0.0.1:8090: connectex: No connection could be made because the target machine actively refused it."
MARN[0096] Request Failed      error=Get "http://localhost:8090/\": dial tcp 127.0.0.1:8090: connectex: No connection could be made because the target machine actively refused it."
MARN[0096] Request Failed      error=Get "http://localhost:8090/\": dial tcp 127.0.0.1:8090: connectex: No connection could be made because the target machine actively refused it."
MARN[0096] Request Failed      error=Get "http://localhost:8090/\": dial tcp 127.0.0.1:8090: connectex: No connection could be made because the target machine actively refused it."
MARN[0097] Request Failed      error=Get "http://localhost:8090/\": dial tcp 127.0.0.1:8090: connectex: No connection could be made because the target machine actively refused it."
MARN[0097] Request Failed      error=Get "http://localhost:8090/\": dial tcp 127.0.0.1:8090: connectex: No connection could be made because the target machine actively refused it."
MARN[0097] Request Failed      error=Get "http://localhost:8090/\": dial tcp 127.0.0.1:8090: connectex: No connection could be made because the target machine actively refused it."
MARN[0097] Request Failed      error=Get "http://localhost:8090/\": dial tcp 127.0.0.1:8090: connectex: No connection could be made because the target machine actively refused it."
MARN[0098] Request Failed      error=Get "http://localhost:8090/\": dial tcp 127.0.0.1:8090: connectex: No connection could be made because the target machine actively refused it."
MARN[0098] Request Failed      error=Get "http://localhost:8090/\": dial tcp 127.0.0.1:8090: connectex: No connection could be made because the target machine actively refused it."
MARN[0098] Request Failed      error=Get "http://localhost:8090/\": dial tcp 127.0.0.1:8090: connectex: No connection could be made because the target machine actively refused it."
MARN[0099] Request Failed      error=Get "http://localhost:8090/\": dial tcp 127.0.0.1:8090: connectex: No connection could be made because the target machine actively refused it."
MARN[0099] Request Failed      error=Get "http://localhost:8090/\": dial tcp 127.0.0.1:8090: connectex: No connection could be made because the target machine actively refused it."
MARN[0099] Request Failed      error=Get "http://localhost:8090/\": dial tcp 127.0.0.1:8090: connectex: No connection could be made because the target machine actively refused it."
MARN[0099] Request Failed      error=Get "http://localhost:8090/\": dial tcp 127.0.0.1:8090: connectex: No connection could be made because the target machine actively refused it."

■ TOTAL RESULTS

checks_total .....: 500    49.5H4825/s
checks_succeeded....: 64.20% 321 out of 500
checks_failed.....: 35.79% 179 out of 500

X status is 200
   64% = ✓ 321 / X 179

HTTP
http_req_duration...: avg=4.71ms min=0s     med=6.12ms max=13.94ms p(90)=9.16ms p(95)=9.94ms
{ expected_response:true }
avg=7.34ms min=1.89ms med=7.36ms max=13.94ms p(90)=9.63ms p(95)=10.64ms
http_req_failed.....: 35.79% 179 out of 500
http_reqs.....: 500    49.5H4825/s

EXECUTION
iteration_duration...: avg=1s min=1s         med=1s max=1.01s p(90)=1.01s p(95)=1.01s
iterations.....: 500    49.5H4825/s
vus.....: 50 min=50 max=50
vus_max.....: 50 min=50 max=50

NETWORK
data_received.....: 136 kB 14 kb/s
data_sent.....: 23 kb 2.2 kb/s

running (10.1s), 90/50 VUs, 500 complete and 0 interrupted iterations
default ✓ [=====] 50 VUs 1ls
PS E:\Lab8> |
```

TOTAL RESULTS

checks_total.....: 500 49.575466/s

checks_succeeded...: 68.20% 341 out of 500

checks_failed.....: 31.80% 159 out of 500

âœ— status is 200

†³ 68% €” “ 341 / âœ— 159

HTTP

http_req_duration.....: avg=4.41ms min=0s med=3.13ms max=15.84ms
p(90)=9.96ms p(95)=11.63ms

{ expected_response:true }...: avg=6.47ms min=513.29Âµs med=6.75ms
max=15.84ms p(90)=10.8ms p(95)=12.29ms

http_req_failed.....: 31.80% 159 out of 500

http_reqs.....: 500 49.575466/s

EXECUTION

iteration_duration.....: avg=1s min=1s med=1s max=1.02s
p(90)=1.01s p(95)=1.01s

iterations.....: 500 49.575466/s

vus.....: 50 min=50 max=50

vus_max.....: 50 min=50 max=50

NETWORK

data_received.....: 144 kB 14 kB/s

data_sent.....: 24 kB 2.4 kB/s

running (10.1s), 00/50 VUs, 500 complete and 0 interrupted iterations
default “ [100%] 50 VUs 10s

4. Проанализируйте выходные данные: количество запросов, время ответа, процент ошибок.

4.1. Сценарий 1: 10 VUs в течение 30s

Количество запросов

- http_reqs: **300** запросов, примерно **9.96149 req/s**

Время ответа

- http_req_duration: **avg=2.65ms, med=2.28ms, p(90)=4.19ms, p(95)=5.17ms, max=10.34ms**

Процент ошибок

- http_req_failed: **0.00%** (0/300)
- Проверка status is 200: **100%** (300/300)

Короткий вывод: при нагрузке 10 VUs система отвечает быстро и стабильно (ошибок нет).

4.2. Сценарий 2: 50 VUs в течение 10s

Количество запросов

- http_reqs: **500** запросов, примерно **49.575466 req/s**

Время ответа

- http_req_duration: **avg=4.41ms, med=3.13ms, p(90)=9.96ms, p(95)=11.63ms, max=15.84ms**

Процент ошибок

- http_req_failed: **31.80%** (159/500)
- Проверка status is 200: только **68.20%** (341/500) → **31.80% неуспешно** (159/500)

Короткий вывод: при 50 VUs пропускная способность выше, но появляется существенная доля ошибок (31.8%) и растёт задержка.

4.3. Сравнение двух уровней нагрузки (итог по пункту 4)

1) Throughput / число запросов

- 10 VUs/30s: **~9.96 req/s**
- 50 VUs/10s: **~49.58 req/s**

→ Пропускная способность выросла примерно в **~5 раз** при увеличении нагрузки.

2) Время ответа

- Avg: **2.65ms** → **4.41ms** (рост примерно в ~1.7 раза)
- p95: **5.17ms** → **11.63ms** (рост примерно в ~2.25 раза)

→ При увеличении нагрузки задержка растёт, особенно заметно по p95 (tail latency).

3) Error rate / устойчивость

- **0.00%** → **31.80%**

→ При 50 VUs/10s система становится **нестабильной** (почти 1/3 запросов с ошибкой).

Общий вывод:

- **10 VUs/30s:** система **стабильна**, низкая задержка, **0% ошибок**.
- **50 VUs/10s:** система **достигает предела по нагрузке** (ошибки резко растут + увеличивается время ответа).

Задание (Ручное тестирование): Сформулируйте цели теста (что мы хотим проверить? пропускную способность? устойчивость?).

Цели нагрузочного тестирования (что именно проверяем):

1. Проверка пропускной способности (Throughput):

Определить, сколько запросов в секунду система может обработать при низкой (10 VUs) и повышенной (50 VUs) нагрузке по метрикам `http_reqs` и `req/s`.

2. Проверка времени ответа (Latency):

Оценить, как меняется задержка при росте нагрузки, используя

http_req_duration (особенно avg и p95), чтобы понять поведение “типичных” и “самых медленных” запросов.

3. Проверка устойчивости/стабильности (Stability):

Проверить, появляются ли ошибки при увеличении числа пользователей: http_req_failed и процент успешных проверок status is 200.

4. Определение предела нагрузки (порог по производительности):

Найти уровень нагрузки, при котором система начинает заметно деградировать (рост latency или ошибок), чтобы понимать, где требуется оптимизация/масштабирование.