



Тестирование производительности SportsBook 2.0 с использованием k6

Daniil Matafonov

October 2020

www.altenar.com





Об авторе



Daniil Matafonov

QA Automation Engineer

Занимаюсь автоматизацией тестирования SportsBook 2.0 в компании Altenar, которая является поставщиком программного обеспечения, для лицензированных игорных операторов



О компании

Altener — международная B2B компания, с офисами в России, на Мальте и в Греции, которая занимается разработкой софта для лицензированных букмекеров, оперирующих в Европе и Южной Америке.

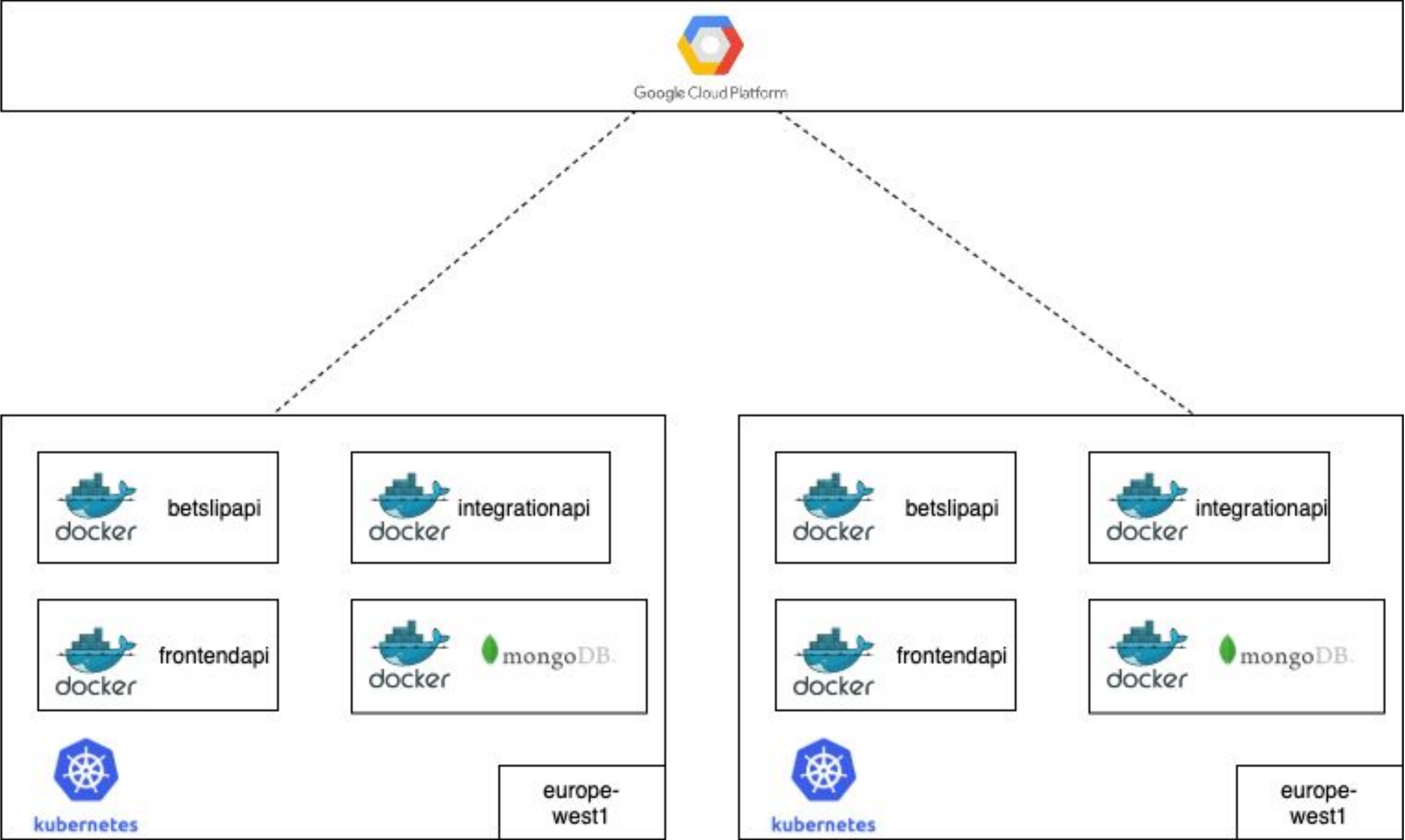


О продукте

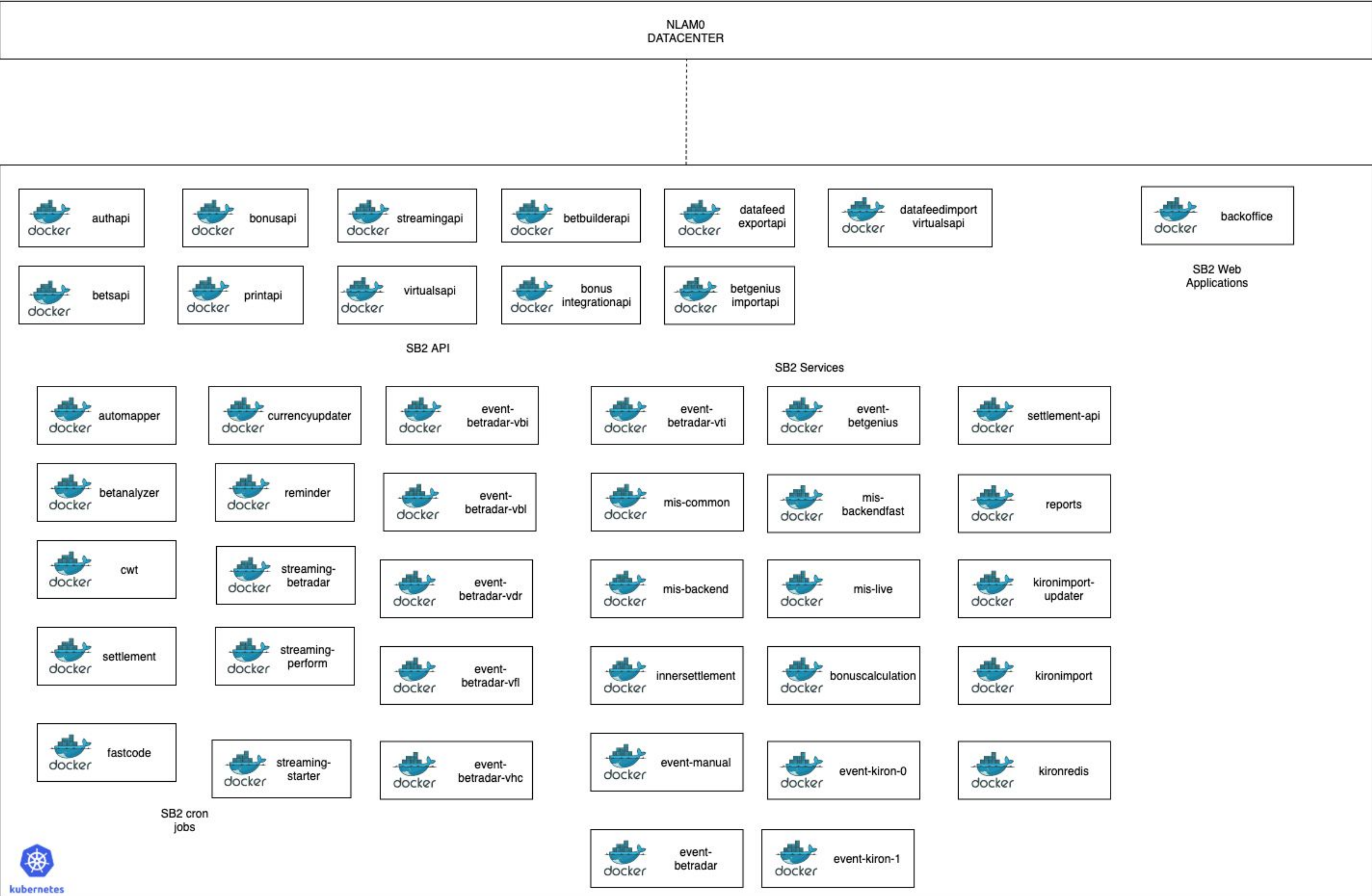
SportsBook - программное обеспечение предназначенное для выполнения ставок на различные спортивные соревнования, включая гольф, футбол, баскетбол, бейсбол, хоккей, скачки, виртуальные виды спорта и другие.



Архитектура SportsBook 2.0 Google Cloud Cluster



SportsBook 2.0 NLAM cluster





Предыстория нагрузочного тестирования



Анализ инструментов

Функциональные особенности	Jmeter	Gatling	k6	Locust
Распределенное выполнение тестов	Master/slave approach [tricky]. The main node for launching slave nodes that simulate traffic.	Manually	Manually. See #1007 below	Master/slave approach. The main node for displaying live statistics and launching slave nodes that simulate traffic
Документация / Поддержка	JMeter is an Apache project. One of the oldest tools due to which it has a large community. Detailed documentation.	The tool supported by the company of the same name, has an open source version. Detailed documentation.	Open Source project supported by LoadImpact . Detailed documentation.	It is developing quite actively, a group of major developers stands out [independent]. Detailed documentation.
Протоколы [gRPC, WebSocket, HTTP]	Out-of-box: <ul style="list-style-type: none"> • HTTP [1.1] Via external plugin: <ul style="list-style-type: none"> • WebSocket • HTTP 2 	Out-of-box: <ul style="list-style-type: none"> • HTTP [1.1/2] • WebSocket Via external plugin: <ul style="list-style-type: none"> • gRPC [only unary calls] 	Out-of-box: <ul style="list-style-type: none"> • HTTP [1.1/2] • WebSocket Note: See #441 below	Out-of-box: <ul style="list-style-type: none"> • HTTP Note: See #17 and #83 below
Отчеты	InfluxDB + Grafana. Built-in GUI [slow]	InfluxDB + Grafana	InfluxDB + Grafana	Built-in web UI.
Особенности использования	Mostly GUI oriented. Test plans are generated in the form of large xml	Tests are written in Scala [functional paradigm]	Tests are written in Javascript run by ECMAScript 5.1(+) implementation written	Protocol-agnostic approach, out of the box only HTTP. Tests are written in Python.



Достоинства и недостатки



K6

- K6 is a modern load testing tool, created by [Load Impact](#).
- It provides a clean, approachable scripting API, [local](#) and [cloud execution](#), and flexible configuration.
- link: <https://k6.io/>



K6 Features

- **Scripting in ES6 JS:** support for [modules](#) to aid code reusability across an organization
- **Everything as code:** test logic and [configuration options](#) are both kept in JS for version control friendliness
- **Automation-friendly:** [checks](#) (like asserts) and [thresholds](#) for easy and flexible CI configuration!
- **HTTP/1.1, HTTP/2** and **WebSocket** protocol support
- **TLS features:** [client certificates](#), [configurable SSL/TLS versions and ciphers](#)
- **Components included:** [Cookies](#), [Crypto](#), [Custom metrics](#), [Encodings](#), [Environment variables](#), JSON, [HTML forms](#), [files](#), [flexible execution control](#), and more.
- **Built-in HAR converter:** record browser sessions as [.har files](#) and [directly convert them to k6 scripts](#)
- **Flexible metrics storage and visualization:** [InfluxDB](#) (+Grafana), [JSON](#) or [k6 Cloud](#)
- **Cloud execution** and distributed tests (currently only on infrastructure managed by [Load Impact](#), with native distributed execution in k6 [planned](#) for the near future!)



K6 Install

Windows

You can manually download and install the [official .msi installation package](#) or, if you use the [chocolatey package manager](#), follow [these instructions](#) to set up the k6 repository.

Docker

```
docker pull loadimpact/k6
```

Linux (deb and rpm packages)

<https://k6.io/docs/getting-started/installation#linux-deb-and-rpm-packages>

Mac (brew)

<https://k6.io/docs/getting-started/installation#mac-brew>



K6 running

In order to run k6 tests you have to run the following command:

Basic run:

```
k6 run test.js
```

Run with InfluxDB:

```
k6 run -o influxdb=http://localhost:8086/Sb2BettingMetrics test.js
```



Сборка, запуск тестов



Metrics

- **vus** - Current number of active virtual users.
- **iteration_duration** - The time it took to complete one full iteration of the default/main function.
- **iterations** - The aggregate number of times the VUs in the test have executed the JS script.
- **http_reqs** - How many HTTP requests has k6 generated, in total.
- **http_req_duration** - Total time for the request. How long did the remote server take to process the request and respond, without the initial DNS lookup/connection times

Config parameters:

stages - property allows to configure ramping behaviour.

duration - A string specifying the total duration a test run should be run for.

target - specify the target number of VUs to ramp up or down to for a specific period.

minIterationDuration - Specifies the minimum duration for every single execution (i.e. iteration) of the default function should be.

httpDebug - used for log all HTTP requests and responses.



Проблемы в k6 (k6 version 0.26)

При запуске теста с количеством VUS > 2000 k6 прерывал выполнение с ошибкой
FATA[0052] stream error: stream ID 5; INTERNAL_ERROR



Сценарии в k6

- shared-iterations
- per-vu-iterations
- constant-vus
- ramping-us
- constant-arrival-rate
- ramping-arrival-rate
- externally-controlled



Интеграция с influxDB



Статистика в k6