Breaking things on purpose: Getting started with k6 Load Testing

Hi,
I am Razvan Vancea

Teaching practical test automation with an engineering mindset

Principal QA Engineer @ Zitec

Founder @ Test Automation Incubator

Certified Trainer / Mentor / Speaker

YouTube Content Creator @ Learn with RV

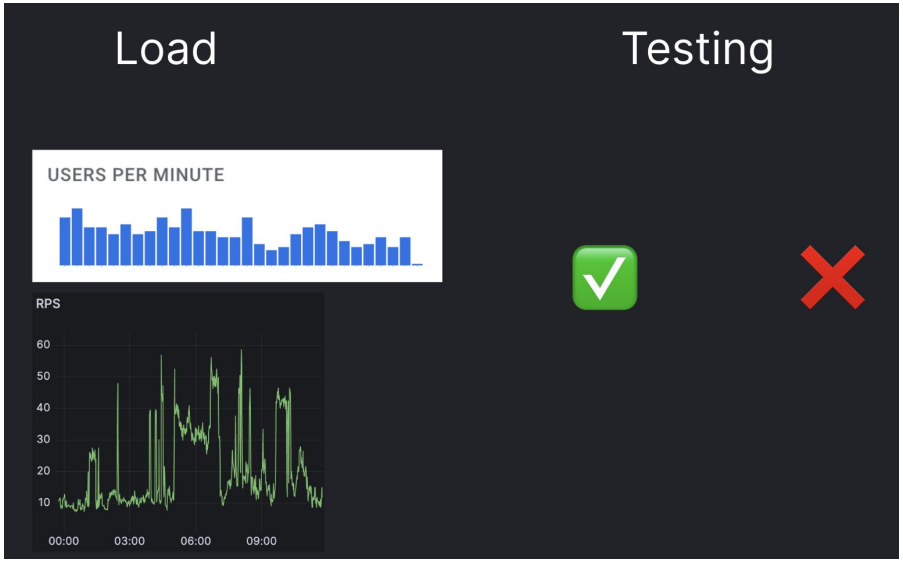
Grafana k6 Champion

Agenda

- Introduction
- Environment setup check
- Foundation & first test
- Building realistic tests
- Advanced scenarios
- Metrics, insights & next steps
- Q & A

Performance Testing

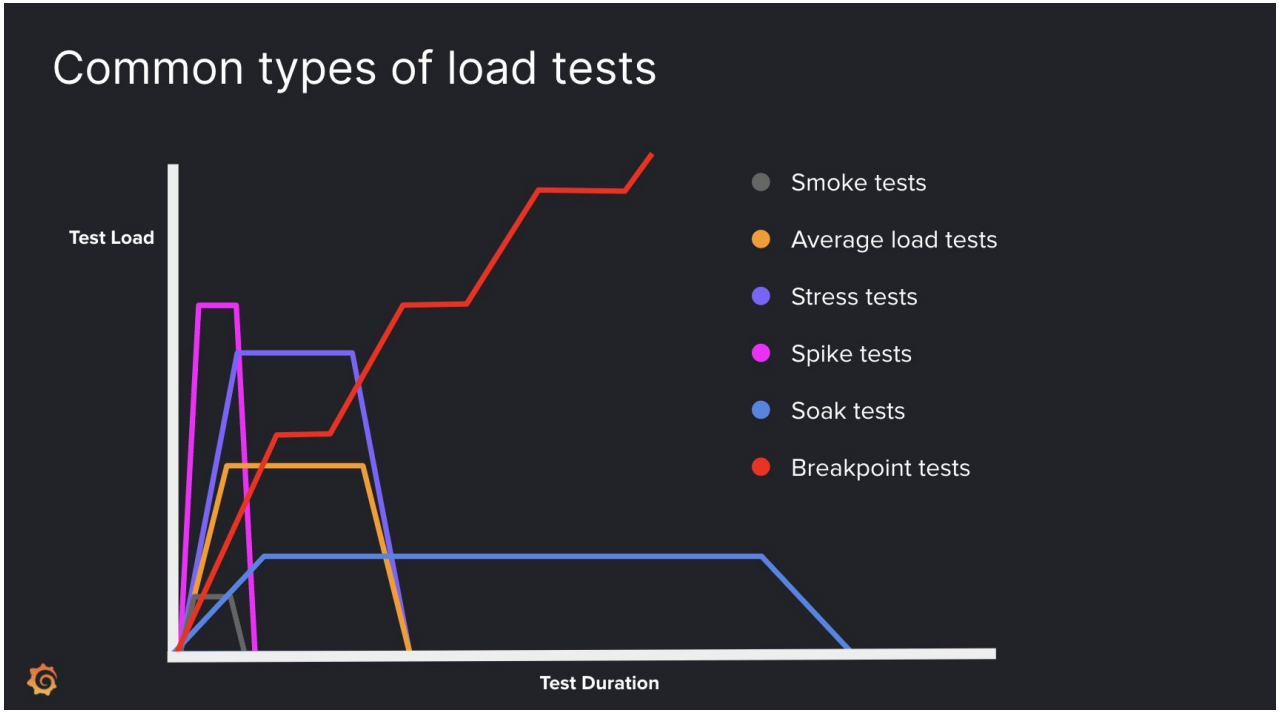
Performance testing is a non functional testing type, aiming to evaluate the speed, responsiveness and scalability of a system. It is an “umbrella” for other testing types.



Load Testing

“Load testing is the process of putting demand on a system and measuring its response”

Load Testing Types



Load Testing Types

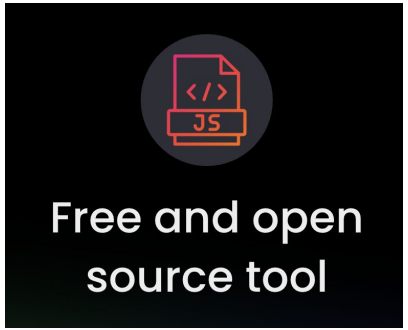
Type	VUs/Throughput	Duration	When?
Smoke	Low	Short (seconds or minutes)	When the relevant system or application code changes. It checks functional logic, baseline metrics, and deviations
Average-load	Average production	Mid (5-60 minutes)	Often to check system maintains performance with average use
Stress	High (above average)	Mid (5-60 minutes)	When system may receive above-average loads to check how it manages
Soak	Average	Long (hours)	After changes to check system under prolonged continuous use
Spike	Very high	Short (a few minutes)	When the system prepares for seasonal events or receives frequent traffic peaks
Breakpoint	Increases until break	As long as necessary	A few times to find the upper limits of the system

Myths about load testing

- Performance testing = load testing
- You need many users to find issues
- We will see performance issues in production anyway

K6 - intro

Grafana k6 is an open-source, developer-friendly, and extensible performance testing tool that helps you catch performance issues early and proactively improve reliability.



K6 - intro

Key concepts to know

Virtual Users (VUs)

How many users are simulated

```
VU1 -> login -> request
VU2 -> login -> request
VU3 -> login -> request
...
```

Duration

How long the test runs

```
VU1 -> login -> request
VU2 -> login -> request
VU3 -> login -> request

Run test for 1 minute
```

Sleep

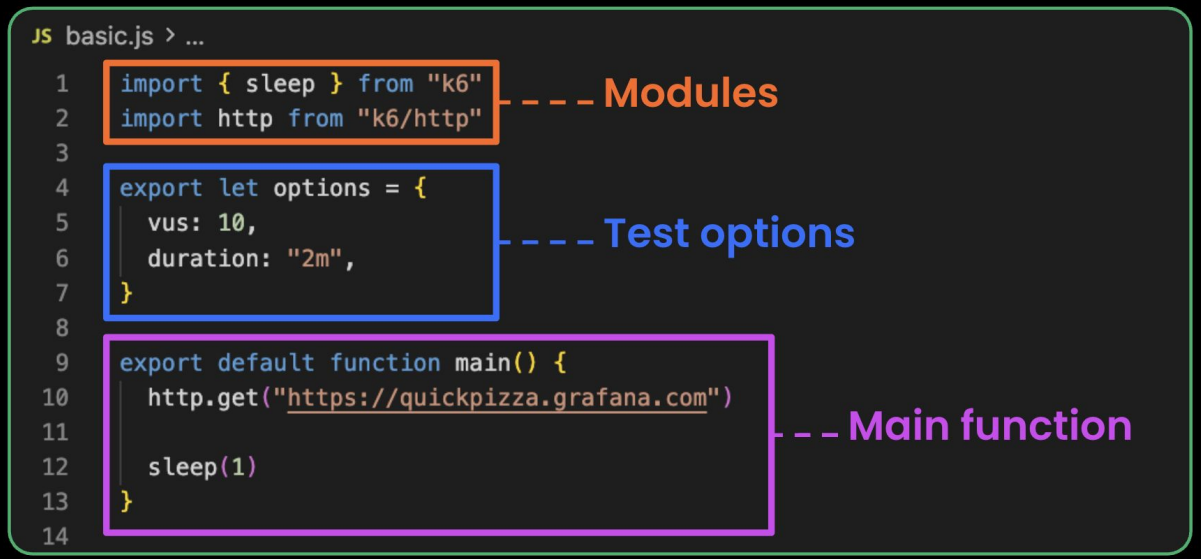
How long users wait before action

```
VU1 -> login -> sleep -> request
VU2 -> login -> sleep -> request
VU3 -> login -> sleep -> request
...
```

K6 - intro

Introduction to k6 OSS

Anatomy of a test script



K6 - intro

Introduction to k6 OSS

Key concepts to know

Thresholds

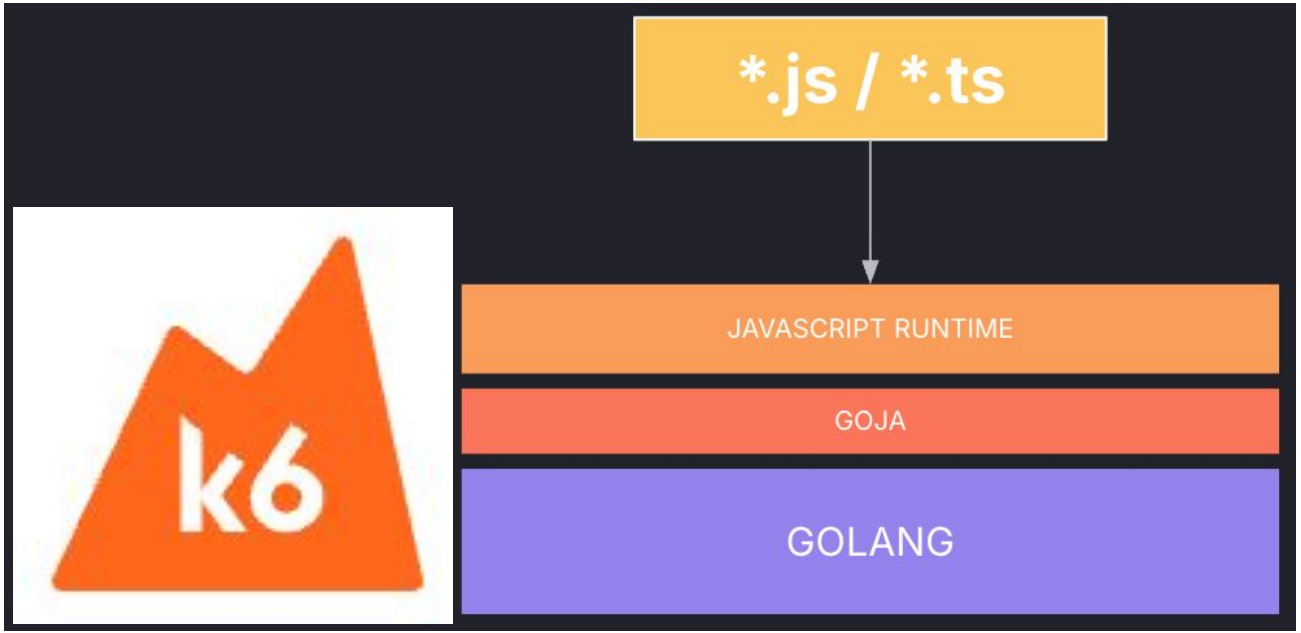
Test pass/fail criteria

Stages

Modelling dynamic load

```
export let options = {
  thresholds: {
    http_req_duration: ["p(95)<=200"],
  },
  stages: [
    { duration: "2m", target: 10 },
    { duration: "1m", target: 30 },
    { duration: "2m", target: 10 },
  ],
}
```

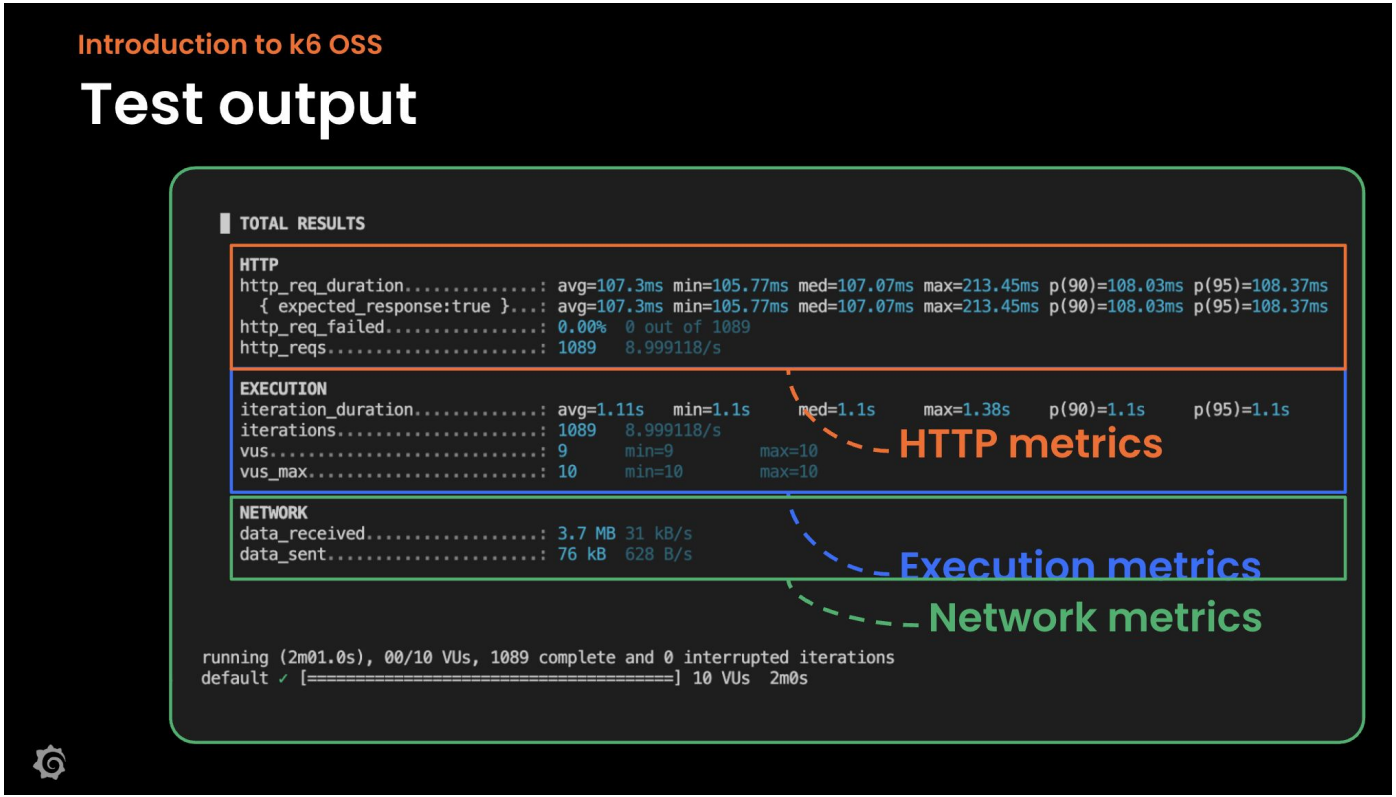
K6 architecture



K6 Results


k6 Results
k6 run script.js
Web dashboard
Terminal
k6 run script.js -o output-xyz
Amazon CloudWatch
Apache Kafka
Datadog
Dynatrace
Elasticsearch
Grafana Cloud Prometheus
InfluxDB
Netdata
New Relic
Prometheus
StatsD
TimescaleDB
Storage
Visualization

K6 Results


Introduction to k6 OSS
Test output
TOTAL RESULTS
HTTP
http_req_duration.....: avg=107.3ms min=105.77ms med=107.07ms max=213.45ms p(90)=108.03ms p(95)=108.37ms
{ expected_response:true }...: avg=107.3ms min=105.77ms med=107.07ms max=213.45ms p(90)=108.03ms p(95)=108.37ms
http_req_failed.....: 0.00% 0 out of 1089
http_reqs.....: 1089 8.999118/s
EXECUTION
iteration_duration.....: avg=1.11s min=1.1s med=1.1s max=1.38s p(90)=1.1s p(95)=1.1s
iterations.....: 1089 8.999118/s
vus.....: 9 min=9 max=10
vus_max.....: 10 min=10 max=10
NETWORK
data_received.....: 3.7 MB 31 kB/s
data_sent.....: 76 kB 628 B/s
HTTP metrics
Execution metrics
Network metrics
running (2m01.0s), 00/10 VUs, 1089 complete and 0 interrupted iterations
default ✓ [==========] 10 VUs 2m0s

Env setup

- NodeJS & NPM
- k6
- VSCode (IDE)
- Git
- Clone workshop repo
  https://gra.fan/ntdk6
- Clone demo app repo
  https://tinyurl.com/books-demo-app

Practice Time

Thank you!