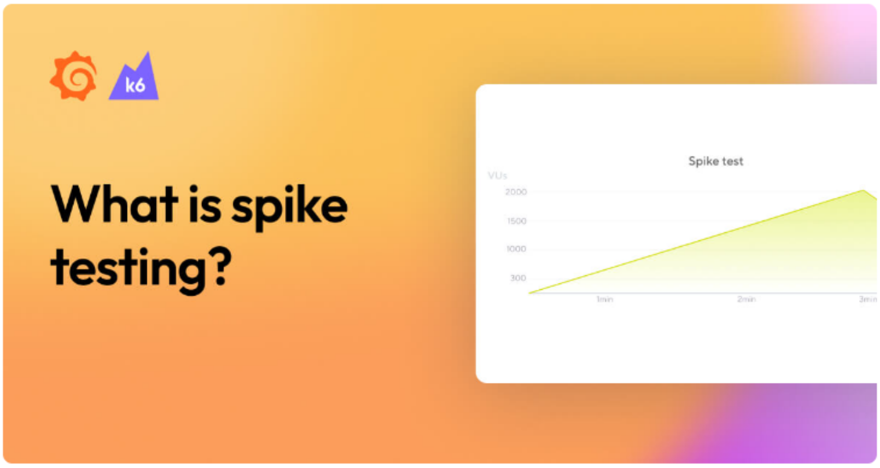


Spike testing: A beginner's guide

Grafana Labs Team • 30 Jan, 2024 • 4 min

This content was originally published on k6.io.
30 Jan 2024

A spike test is a type of load testing that verifies whether the system survives and performs under sudden and massive rushes of utilization.

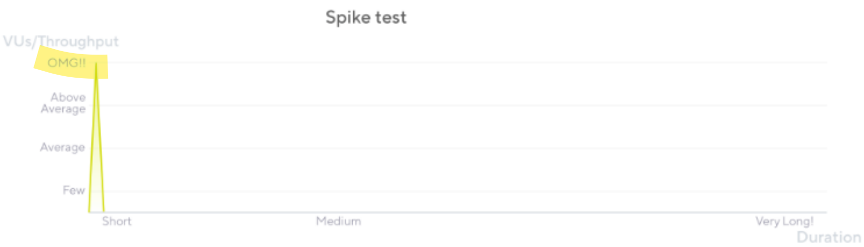
Ex

Spike tests are useful when the system may experience events of sudden and massive traffic. Examples of such events include ticket sales (Taylor Swift), product launches (PS5), broadcast ads (Super Bowl), process deadlines (tax declaration), and seasonal sales (Black Friday). Also, spikes in traffic could be caused by more frequent events such as rush hours, a particular task, or a use case.

Spike testing increases to extremely high loads in a very short or non-existent ramp-up time. Usually, it has no plateau period or is very brief, as real users generally do not stick around doing extra steps in these situations. In the same way, the ramp-down time is very fast or non-existent, letting the process iterate only once.

This test might include different processes than the previous test types, as spikes often aren't part of an average day in production. It may also require adding, removing, or modifying processes on the script that were not in the average-load tests.

Occasionally, teams should revamp the system to allow or prioritize resources for the high-demand processes during the event.



When to perform a spike test

This load test must be executed when the system expects to receive a sudden rush of activity.

When the system expects this type of behavior, the spike test is the type of load testing that can help identify how the system will behave and if it will survive the sudden rush of activity. The load is considerably higher than average and might focus on a different set of processes than the other test types.

Considerations

When preparing for a spike test, consider the following:

- Focus on key processes in this test type

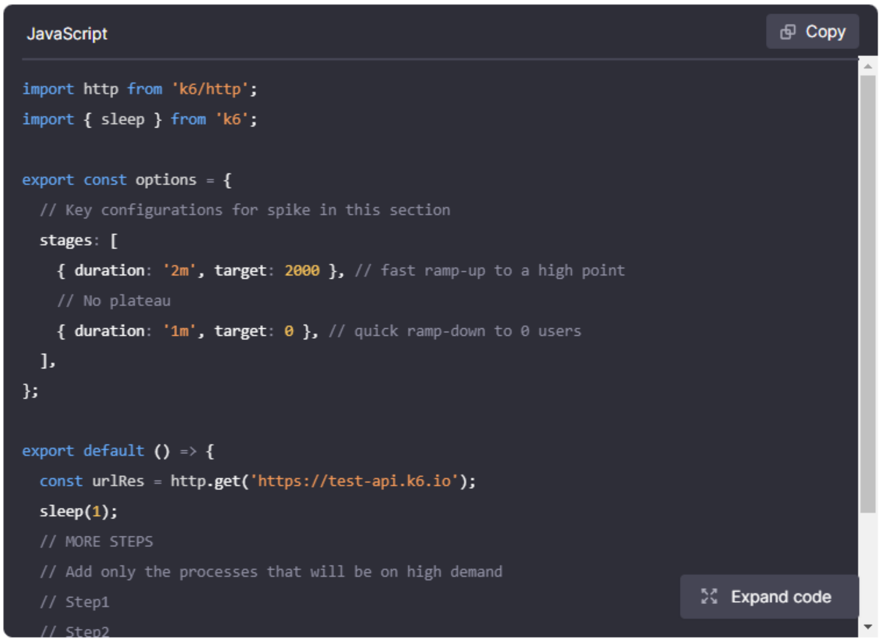
- **Focus on key processes in this test type.**
  Assess whether the spike in traffic triggers the same or different processes from the other test types. Create test logic accordingly.
- ✓ **The test often won't finish.**
  Errors are common under these scenarios.
- ✓ **Run, tune, repeat.**
  When your system is at risk of spike events, the team must run a spikes test and tune the system several times.
- **Monitor.**
  Backend monitoring is a must for successful outcomes of this test.

Spike testing example

The key differentiator of the spike test are the simulation of sudden and very high loads. It lacks a plateau (full load) duration or is usually brief.

Sometimes, the test may require a load plateau for some time. If a plateau is needed, it's generally short. A ramp down can also be quick or unnecessary as the goal is to suddenly increase the system's load.

Below is a load testing example using Grafana k6 to perform a spike test:

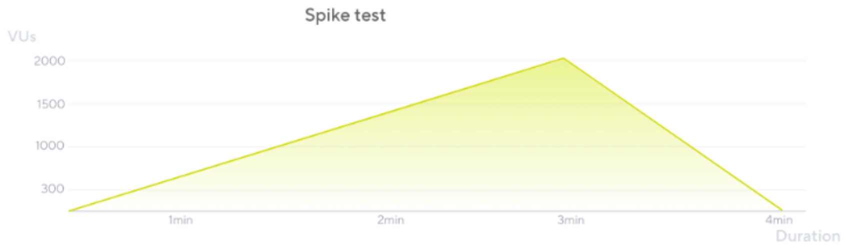


```javascript
import http from 'k6/http';
import { sleep } from 'k6';

export const options = {
  // Key configurations for spike in this section
  stages: [
    { duration: '2m', target: 2000 }, // fast ramp-up to a high point
    // No plateau
    { duration: '1m', target: 0 }, // quick ramp-down to 0 users
  ],
};

export default () => {
  const urlRes = http.get('https://test-api.k6.io');
  sleep(1);
  // MORE STEPS
  // Add only the processes that will be on high demand
  // Step1
  // Step2
```

In a spike test, load quickly increases to an extreme level. The ramp-down period follows when the test reaches the maximum, returning to 0 quickly.



Results analysis

Some performance metrics to assess in spike tests include pod speeds, recovery times after the load rush, time to return to normal, or the behavior on crucial system processes during the overload.

Finding how the system responds to the sudden rush (if it survives) helps to optimize it to guarantee that it can perform during a real event. In some events, the load is so high that the whole system may have to be optimized to deal with the key processes. In these cases, repeat the test until your confidence in the system is high.

Grafana Cloud is the easiest way to get started with Grafana k6 and performance testing. We have a generous forever-free tier and plans for every use case. Sign up for free now!

Tags

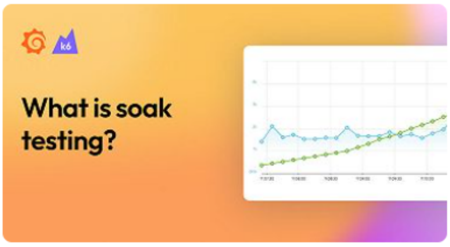
k6 Load testing Performance testing

Up next

Grafana Labs Team · 30 Jan 2024 · 4 min read

Stress testing: A beginner's guide

k6 Load testing Performance testing



Grafana Labs Team · 30 Jan 2024 · 4 min read

Soak testing: A beginner's guide

k6 Load testing Performance testing

Grafana Labs Team · 30 Jan 2024 · 3 min read

Smoke testing: A beginner's guide

k6 Load testing Performance testing

Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

Grafana

- Overview
- Deployment options
- Plugins
- Dashboards

Products

- Grafana Cloud
  - Grafana Cloud Status
- Grafana Enterprise Stack
- Grafana Cloud Application Observability
- Grafana Cloud Frontend Observability
- Grafana Cloud IRM
- Grafana Cloud k6
- Grafana Cloud Logs
- Grafana Cloud Metrics
- Grafana Cloud Profiles
- Grafana Cloud Synthetic Monitoring
- Grafana SLO

Open Source

- Grafana
- Grafana Loki
- Grafana Mimir
- Grafana OnCall
- Grafana Tempo
- Grafana Agent
- Grafana Alloy
- Grafana k6
- Prometheus
- Grafana Faro
- Grafana Pyroscope
- Grafana Beyla
- OpenTelemetry
- Grafana Tanka
- Graphite
- GitHub

Learn

- Grafana Labs blog
- Documentation
- Downloads
- Community
- Community forums
- Community Slack
- Grafana Champions
- Community organizers
- Grafana ObservabilityCON
- GrafanaCON 2024
- The Golden Grot Awards
- Successes
- Workshops
- Videos
- OSS vs Cloud
- Load testing

Company

- The team
- Press
- Careers
- Events
- Partnerships
- Contact
- Getting help
- Merch

Grafana Cloud Status

Legal and Security Terms of Service Privacy Policy Trademark Policy