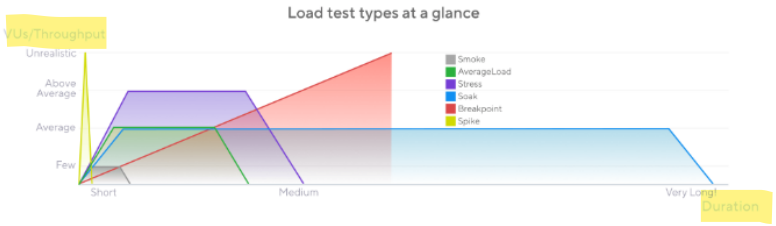
Types of load testing

Many things can go wrong when a system is in use. On an average day, the system must run numerous operations simultaneously and respond to different requests from a range of users, but there could also be a sudden spike in users or a major event that pushes your system to its limits — or beyond. To prepare for these performance risks, teams use load testing to see how an application or system will perform in a variety of use cases.

But a good load-testing strategy goes beyond just executing a single script. Different patterns of traffic create different risk profiles for a given application. For comprehensive preparation, teams must test the system against different types of load testing.

Types of load testing:
- Smoke testing
- Average load testing
- Soak
- Spike
- Breakpoint
- Stress



Here we will review the most common types of load testing and the use cases for each load testing type.

Related content

What type of testing is load testing?

Load testing is a subset of performance testing that generally looks for how a system responds to normal and peak usage. You're looking for slow response times, errors, crashes, and other issues to determine how many users and transactions the system can accommodate before performance suffers.

Testers today typically rely on open source load testing tools or specialized, cloud-based automation tools, like Grafana Cloud k6, to test the system with virtual users and simulated data volumes to see how the extra load impacts performance. They monitor and measure response time, throughput, and resource utilization to detect potential bottlenecks or scaling issues that need to be addressed before the system is deployed in production.

Grafana cloud k6 - cloud based automation tool

When you have a fast software development and delivery process, engineering teams need a robust and reliable testing suite that can keep up with the pace of continuous development and deployment. Such a testing platform helps teams ensure the high quality of every release. To make testing reliable, teams should perform different types of load testing across various environments, including development, canary, QA, pre-production, and production. Teams should also automate testing in continuous delivery pipelines to prevent errors when shipping new features or experiments iteratively to the end-user.

Although automation and frequency vary depending on the load testing type, continuous and automated load testing are now standard practices and the end goal for most types of load testing.

Load testing vs. performance testing

While load testing and performance testing are related, they are distinct types of testing.

Load Testing --> System can handle increased traffic or load

As we've discussed, load testing simulates user activity to determine how well a system can handle increased traffic or load.

Performance Testing --> How the system perform overall -> Speed, scalability, reliability, resource utilization

Performance testing is an umbrella term for measuring how well a system or application performs overall. This could include testing for speed, scalability, reliability, and resource utilization in order to identify areas of improvements. Performance testing includes load testing but also encompasses other types of testing, such as browser performance testing and synthetic monitoring.

How many types of load tests are there?

An application performs differently depending on the volume and duration of traffic it handles at any given moment. You should never assume your app will perform the same when supporting 10 or 100 users vs. 1,000 or 5,000 users and beyond.

There are six common types of load testing that you can execute on your applications to measure performance under different loads.

1. Smoke test

Smoke tests verify the system functions with minimal load, and they are used to gather baseline performance values. Smoke tests are also called shakeout tests.

This test type consists of running tests with a few VUs. For example, more than 5

virtual users (VUs) could be considered a mini-load test. Similarly, the test should be executed for a short period, either a low number of iterations or a duration from seconds to a few minutes maximum.

2. Average-load test

Average-load tests assess how the system performs under a typical load for your system or application. Typical load might be a regular day in production or an average timeframe in your daily traffic. This test also might be called a day-in-life test or volume test.

Day-in-life test or volume test

Average-load tests simulate the number of concurrent users and requests per second that reflect average behaviors in the production environment. This type of test typically increases the throughput or VUs gradually and maintains that average load for some time. Depending on the system's characteristics, the test may stop suddenly or have a short ramp-down period.

3. Stress test

Stress tests help you discover how the system functions with the load at peak traffic. Stress testing might also be called rush-hour testing, surge testing, or scale testing. See the What is stress testing in load testing? section to learn more.

Rush-hour test / Scale test / Surge test

4. Spike test

A spike test verifies whether the system survives and performs under sudden and massive rushes of utilization.

Spike tests are useful when the system may experience events with exceptional traffic volumes. 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.

Spike testing increases to extremely high loads in a very short or non-existent ramp-up time. In the same way, the ramp-down 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 testing script that are not normally incorporated in your average-load tests.

5. Breakpoint test

Breakpoint tests discover your system's limits. Breakpoint testing is also known as capacity, point load, and limit testing.

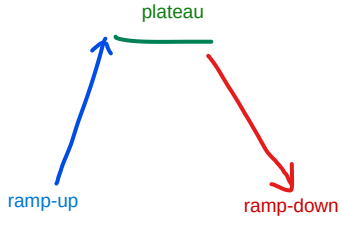
The reasons you might want to conduct a breakpoint test include:

- To tune or care for your system's weak spots to reallocate those higher limits at higher levels.
- To help plan remediation steps in those cases and prepare for when the system nears those limits.

It's not only about knowing at what point your system will fail. It's also a test to help determine where and how a system starts to fail and helps teams prepare for such limits.

A breakpoint test ramps to unrealistically high numbers. This test commonly has to be stopped manually or automatically as thresholds start to fail. When these problems appear, the system has reached its limits.

6. Soak test



Soak tests are a variation of the average-load test. The main difference is the test duration. In a soak test, the peak load is usually an average load, but the peak load duration extends several hours or even days. Though the duration is considerably longer, the ramp-up and ramp-down periods of a soak test are the same as an average-load test.

A soak test might also be called an endurance, constant high load, or stamina test.

Soak tests focus on analyzing the following:

- The system's degradation of performance and resource consumption over extended periods.
- The system's availability and stability during extended periods.

The following table provides some broad comparisons of the six main types of load testing:

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

The overall goal of a load test determines the type of load test(s) you may need. The test types that you choose will then inform how you plan, structure, and execute your load test. But each application, organization, and testing project differs.

Our recommendation for load testing is always to **start simple and test frequently**. For example, start with smoke tests, then progress to higher loads and longer durations. You can always iterate and grow your testing suite, adding more load testing types as you gradually incorporate load testing into your workflows.

What is stress testing in load testing?

Stress testing is a popular type of load testing that assesses how the system performs when the workload is heavier than usual.

Stress tests verify the stability and reliability of the system under heavier than normal usage. Systems may receive higher than usual workloads, such as process deadlines, paydays, rush hours, the end of the workweek, and many other occasions that might cause frequent higher-than-average traffic.

When you run a stress test, consider the following best practices, tips, and tricks:

- **Load should be higher than what the system experiences on average**

Some testers might have default targets for stress tests—say an increase upon average load by 50% or 100% — but there's no fixed percentage.

The load simulated in a stress test depends on the type of situation that the system may be subject to. Sometimes this may be just a few percentage points above average. Other times, it can be 50 to 100% higher, as mentioned. Some stressful situations can be double, triple, or even orders of magnitude higher than an average load.

Regardless, you should define the load in stress tests according to the risk load patterns that the system may receive.

- **Only run stress tests after running average-load tests**

Identify performance issues under average-load tests before trying anything more challenging. This sequence is essential.

- **Re-use the average-load test script**

Modify the parameters to have higher load or VUs.

- **Expect worse performance compared to average load**

This test determines how much the performance degrades with the extra load and whether the system survives it. A well-performing system should respond with consistent response times when handling a constant workload for an extended period.

Load testing types: best practices

When you write and run different load testing types, consider the following best practices.

1. Start with a smoke test

Start with a smoke test. Before beginning larger tests, validate that your load testing scripts work as expected and that your system performs well with a few users.

After you know that the script works and the system responds correctly to minimal load, you can move on to average-load tests. From there, you can progress to more complex load patterns.

2. The specifics depend on your use case

Systems have different architectures and different user bases. As a result, the correct load testing strategy is highly dependent on the risk profile for your organization. Avoid thinking in absolutes.

For example, the open source tool Grafana k6 can model load by either number of VUs or by number of iterations per second (open vs. closed). When you design your test, consider which pattern makes sense for the type.

What's more, no single test type eliminates all risk. To assess different failure modes of your system, incorporate multiple test types. The risk profile of your system determines what test types to emphasize:

- Some systems are more at risk of longer use, in which case soaks should be prioritized.
- Others are more at risk of intensive use, in which case stress tests should take precedence.

No matter the situation, no single test can uncover all issues.

What's more, the categories themselves are relative to use cases. A stress test for one application could be considered an average-load test for another. Indeed, as you have seen, there is not even consensus about the specific names of the various types of load testing.

3. Aim for simple designs and reproducible results

While the specifics are greatly context-dependent, what's constant is that you want to make results that you can compare and interpret.

Stick to simple load patterns. For all test types, directions are the same: ramp-up, plateau, ramp-down.

Avoid "rollercoaster" series where load increases and decreases multiple times. These will waste resources and make it hard to isolate issues.

Why load testing matters

A comprehensive suite of various load tests deployed on your system can ultimately help you deliver an excellent user experience.

When you apply load testing, you are ensuring that the experience for users is:

- **Fast.** Savvy customers expect your website to be fast — whether it's on a typical day or during a major event — and may take their business to competitors if they experience slow-performing pages.
- **Error-free.** If users experience glitches or errors, they may post about their negative experiences on public forums like social media and customer review websites.
- **Positive.** Many factors can lead to a negative user experience — slowness, errors, etc. Load testing helps you ensure users have positive experiences when using your applications.

Users expect a great experience from every site and application they visit. A best-in-class user experience is critical to the success of your business — and leveraging different types of load testing can assure you deliver a premium experience in any circumstance.

> **An easier way to get started**
> Grafana Cloud is the easiest way to get started with metrics, logs, traces, and dashboards. We have a generous forever-free tier and plans for every use case.
> Sign up for a free Grafana Cloud account →
> Read more about Grafana Cloud →