Gatling

2026

WHITEPAPER

The complete guide to SLOs for performance

How to define, validate, and operate
Service Level Objectives for your load testing practice

Most load tests produce data, not decisions

Teams run a test, get a p95 of 340ms, and then debate whether that's good.

The test did its job, but nobody defined what passing looked like. A Service Level Objective (SLO) closes that gap. It's a performance contract: a target your system must meet, evaluated continuously, producing a clear verdict on every run.

The cost of leaving that undefined is no longer abstract.

300k+

Cost of a single hour of downtime for 90%+ of mid-size and large enterprises (ITIC, 2024)

41%

Report losses between $1 million and $5M for every hour of downtime

1,000+

Companies report issues when choosing the wrong SLOs

This guide covers

DEFINE

Why load tests without SLOs leave value on the table, and the anatomy of an SLO: metric, threshold, compliance.

CALIBRATE

Which metrics to track and which to ignore, plus industry benchmarks drawn from FDIC, DORA, Google, and ITIC.

ENACT

How to put SLOs to work: use error budgets to govern releases, and turn every load test into a decision, not a debate.

Who is this guide for: Engineering managers accountable for performance standards, platform and SRE teams standardizing targets across services, and development teams who want an unambiguous answer at the end of every test.

A load test without an SLO is just data

The 2025 O'Reilly/Google survey found that 54% of organizations don't use SLOs at all.

But the direction of travel is clear, and the question is whether you get there before your next incident does.

82%

Cost of a single hour of downtime for 90%+ of mid-size and large enterprises (ITIC, 2024)

95%

Say SLOs help them make better business decisions

54%

Still use no SLOs today, moving from early adopter to standard practice

The cost of operating without them is subtle, but deadly

Unfalsifiable test results

Without a threshold, no run can fail, which means no run can meaningfully pass either.

Inconsistent judgment

Whether a run was "good" depends on who's looking and what they remember from last time.

Invisible regressions

Performance degrades gradually. Without a fixed line, each small regression looks acceptable against the previous one.

Unproductive escalations

When an incident hits production, the question "did we test for this?" has no defensible answer.

The anatomy of an SLO

In Gatling Enterprise Edition, every SLO comes down to three parameters.

METRIC

What you're measuring

A response time percentile (p50, p95, p99, up to p99.9999) or the error ratio across all scenarios.

THRESHOLD

The target value

How long you hold it. Milliseconds for response time, a percentage for error ratio.

COMPLIANCE

How long you held it

Is the percentage of seconds during the run where the condition held.

That third parameter is what separates an SLO from a plain assertion. A pass/fail tells you whether the run ever crossed the line. A compliance score tells you how long you stayed on the right side of it. If your p99 held under 400ms for 94% of the run, you know that number, and you know there's a 6% you need to look at.

Results show up in the run summary as colour-coded gauges:

Results show up in the run summary as colour-coded gauges:



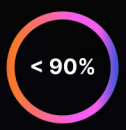
Green

Means the target held 99% of the time or more. The system held the line.



Orange

Means somewhere between 90% and 99%. Close enough to be worth a look



Red

Means under 90%

The SLO was breached

SLOs are configured entirely within Gatling Enterprise Edition, so engineering managers and platform teams can define and adjust thresholds without depending on the developer who wrote the test.

Choosing your metrics

Three metrics cover the vast majority of performance SLO needs.
In trading, a 1 millisecond latency advantage can be worth $100 million annually to a major brokerage.

1 **p95 response time: your primary threshold**
It captures the experience of your slowest 5% of requests: representative of real degradation without being dominated by outliers.

p50 hides tail behavior entirely; half your users could be having a bad experience and the median wouldn't move.

2 **p99 response time: your safety net**
A service can pass p95 while a predictable 1% of sessions time out. For revenue-critical paths, that 1% matters.

One nuance: set p99 at 95% compliance rather than 99% tail latency is inherently noisier, and holding it to 99% produces false failures on healthy runs.

3 **Error ratio: your first signal of systemic failure**
The most overlooked of the three: almost every team tracks availability, fewer track latency percentiles, almost none define an error ratio target.

Yet it catches the "available but failing" state that availability monitoring misses.

COMMON MISTAKES TO AVOID

Using averages instead of percentiles: a 100ms average can hide 1% of requests taking several seconds.

Setting 100% targets: They're expensive, rarely match user perception, and leave no room for safe change.

Tracking p99.9+ on application-layer services: appropriate for carrier-grade infrastructure, noise for everyone else.

The risk: A performance Gap means competitive position lost in milliseconds, reporting windows missed, and regulatory exposure that compounds with every hour the system isn't fully operational.

Measure what users actually feel

For example, Google's Gmail team improved availability from 99.0% to over 99.9% not by changing the service, but by measuring SLIs at the client instead of the server. The server-side numbers had looked fine all along.

Where you measure matters as much as what you measure.

Reduce risk with Gatling.
Welcome to Continuous Performance Intelligence.

AI capabilities
Automated run summaries surface what went wrong without manual analysis. IDE integration and MCP server support bring load testing into the workflows your engineers already use.

SLOs and compliance scoring
Define response time and error rate targets directly in Gatling Enterprise Edition. Every run returns a compliance score, not a pass/fail, a precise percentage. Know exactly how long your system held the line.

Test as code
Write load test scenarios in Java, JavaScript or Scala. Versioned, reviewed, and living in the same repository as your application. Plug directly into your CI/CD pipeline.

Private locations
Load generation runs inside your network perimeter. No traffic leaves, no firewall exceptions. Test what's actually in production.

Setting thresholds for your service

Industry benchmarks are context, not prescription.

Your SLO should reflect three things about your specific service.

1 **Service type sets the base**
Authentication needs to respond in 150ms. It's a gate, and users have zero tolerance for slow logins.

A background pipeline can take 2 seconds, because the latency that matters is time-to-completion, not acknowledgement. Expectations vary by an order of magnitude across service types; a single standard produces thresholds that are either dangerously loose or impossibly tight.

2 **Volume calibrates the consequence**
A 5% p95 breach at 100 requests/minute affects 5 users. At 10,000 requests/minute, it affects 500, every minute.

High-volume services warrant tighter thresholds; low-traffic internal tools can afford looser ones.

3 **Stakes determine the margin**
If you have contractual SLAs, your internal SLO needs a safety buffer below your external commitment.

Operate exactly at the SLA line and you will breach it. If users tolerate latency, give your team headroom instead

Try our SLO advisor

Answer four questions: industry, service type, volume, stakes and get tailored p95, p99, and error ratio targets with the reasoning behind each, ready to configure in Gatling Enterprise Edition.

GATLING.IO/SLO-ADVISOR

Industry benchmarks

Every benchmark below traces to a regulatory document, formal standard, or controlled study.

INDUSTRY	AVAILABILITY	P95	ERROR RATE
Fintech and payments	99.999%	<200ms	<0.005%
E-commerce and retail	99.95%	<300ms	<0.1%
SaaS and enterprise	99.9%	<500ms	<0.1%
Media and streaming	99.95%	<500ms	<0.2%
Healthcare	99.99%	<300ms	<0.01%

90%

Of businesses now require a minimum of 99.99% availability. Just 52.6 minutes of downtime per year

44%

Target five nines, allowing only 5.26 minutes of downtime annually (ITIC, 2024)

4 nines

Is no longer aspirational; it's the baseline expectation for serious production systems

Operating SLOs day to day

Setting targets is one thing. Using them to run an engineering organisation is another. **Use error budgets to govern releases because an error budget isn't an abstraction; it's a spending account.**

An SLO creates an error budget: the unreliability you can absorb before reliability work takes priority. A 99.95% target on 10 million monthly requests allows 5,000 bad events, and every one has a price.

$14,056
Average cost of unplanned downtime per minute

$23,750
Per minute for large enterprises specifically

5,000
Bad events allowed monthly at 99.95% on 10M requests

BUDGET STATE	ACTION
Healthy	Normal releases
50% consumed	Increase monitoring, reduce risky deploys
80% consumed	Require approval for critical-path changes
Exhausted	Freeze non-critical releases

Separate your failures from the provider's

Track a customer-facing SLO alongside an internal SLO. This pinpoints where failures originate and stops "the provider was down" from masking your own gaps.

Test your SLOs; don't just monitor them

Monitoring tells you what happened. Load testing tells you what will happen under pressure. An SLO that has never been validated against peak traffic is a hypothesis, not a commitment.

From definition to verdict

The complete SLO loop turns performance from a debate into a measurement.

1. **Define**
   Choose p95, p99, and error ratio targets suited to your service. Start with the SLO Advisor if you need a defensible first number.
2. **Configure**
   Set them in Gatling Enterprise Edition without using code. Stack as many SLOs per test as you need.
3. **Run**
   Every load test evaluates every SLO continuously, second by second.
4. **Read**
   Compliance gauges give an instant verdict: green held the line, orange needs a look, red breached.
5. **Act**
   Burn-rate informs release decisions; failed compliance points to exactly where to investigate.
6. **Refine**
   Adjust thresholds as your system and your standards evolve.

Try our SLO advisor

Answer four questions: industry, service type, volume, stakes and get tailored p95, p99, and error ratio targets with the reasoning behind each, ready to configure in Gatling Enterprise Edition.

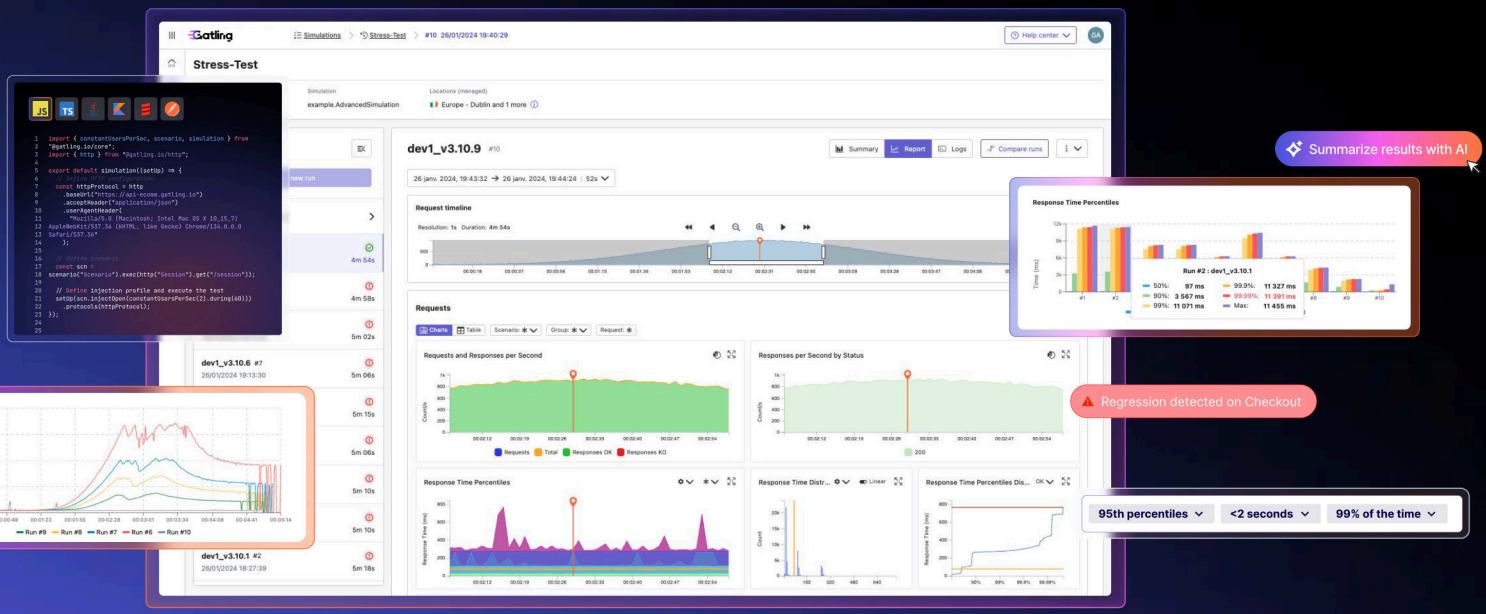
Building performance into the business

Financial institutions that treat performance testing as an engineering checkpoint will keep getting surprised by predictable failures.

The ones that treat it as an operational discipline, with clear ownership, realistic conditions, and decision-driven results, will ship faster, fail less, and operate with the confidence that earns and keeps trust in an industry where trust is everything.

The pattern is the same across every financial services company in this whitepaper: test private, business-critical systems inside your own infrastructure. Scale from open source to enterprise when governance and visibility matter. Embed testing into your release process. Use results to make decisions, not just produce reports.

Your next peak event, month-end close, or major release is already scheduled. The question is whether you'll know your systems are ready before it arrives, or after.



Gatling

Gatling Enterprise Edition is the platform purpose-built to deliver Continuous Performance Intelligence: transforming load testing from a technical activity into an organizational capability that business leaders can govern, product teams can engage with, and engineering teams can scale.

With its powerful open-source and enterprise platforms, Gatling empowers teams to test APIs, microservices, and web apps in real-world conditions.

Trusted by thousands of companies worldwide, Gatling is the performance backbone for development, QA, and DevOps teams building the next generation of software.

Whether you're scaling APIs, migrating to the cloud, or handling flash traffic spikes, Gatling helps you deliver fast, reliable performance.

Ready to implement continuous performance intelligence?

See how AI-assisted performance testing can take you to the next level

Talk to an expert ›