Gatling

2025

DATASHEET

Top load testing metrics to follow

For Performance Engineering, Platform, and QA Teams

TOP LOAD TESTING METRICS TO FOLLOW

Modern teams need more than pass/fail results, they need rich, contextualized metrics to diagnose bottlenecks and validate resilience at scale. Gatling Enterprise Edition gives you the full picture, from request-level behavior to network and infrastructure signals.

1. Requests and Responses per Second

What it is

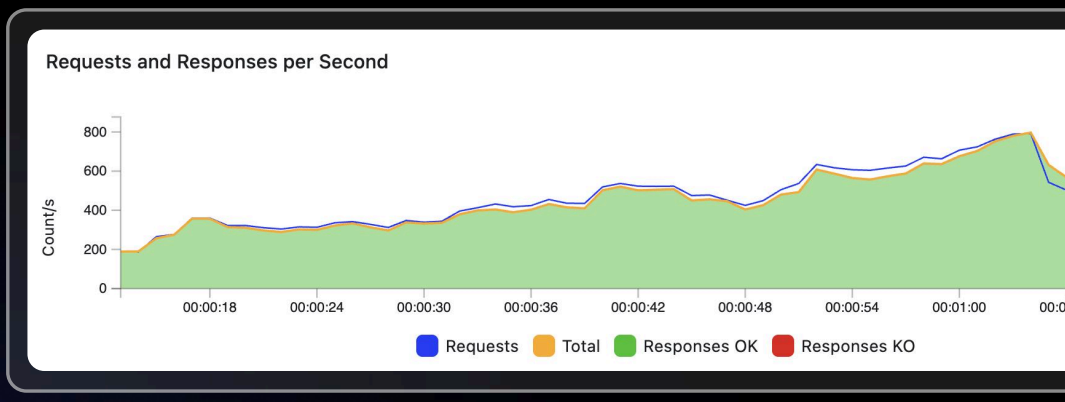
Tracks the volume of requests sent to the system and responses returned every second during a load test.

What it reveals

Throughput trends reveal system capacity under increasing load. If the request curve continues rising while the response curve plateaus or drops, it usually signals thread pool exhaustion, backend saturation, or queue overflows.

Pro tips

Overlay throughput with concurrent users and error rates to identify the exact inflection point where performance starts degrading.



2. Response Time Percentiles (P95 / P99)

What it is

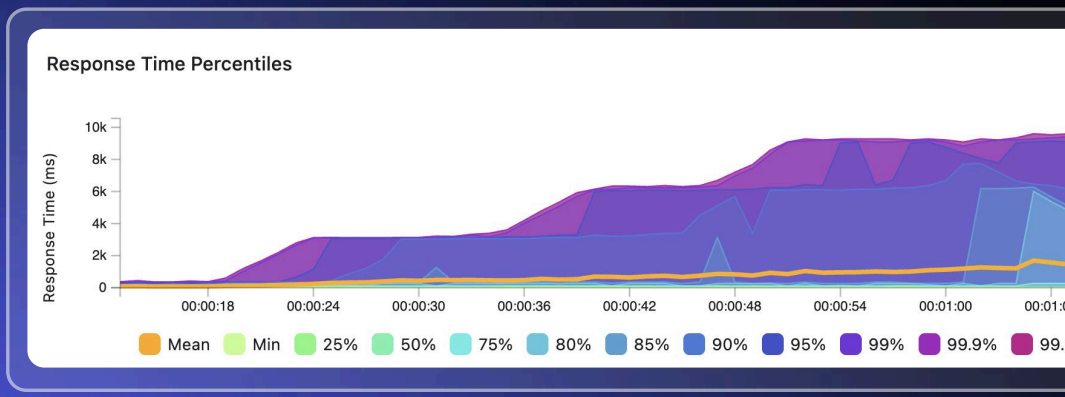
Measures latency for the slowest users (typically the 95th and 99th percentiles) rather than averages.

What it reveals

Percentiles expose tail latency, which is what real users experience in worst-case scenarios. Average response times can mask these spikes entirely. A rising P99 with a stable mean is a classic early warning sign of a regression.

Pro tips

Always track P95 and P99 during ramp-up and peak phases. Tail latency is what breaks SLAs and drives churn, not averages.



3. Errors per Second & Responses by Status Code

What it is

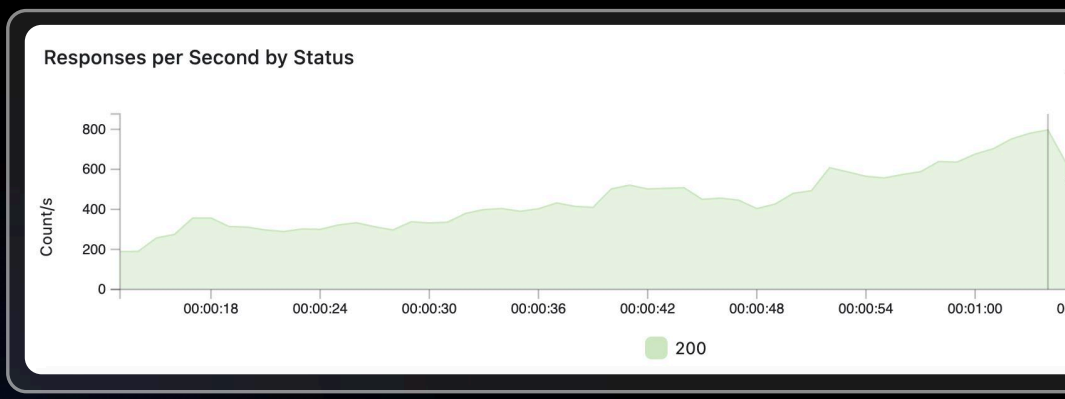
Monitors the volume of errors (e.g. 4xx, 5xx) per second and breaks down responses by HTTP status codes.

What it reveals

Error spikes correlate strongly with system instability. For example, 5xx surges after peak load indicate capacity or dependency failures, while 4xx growth may reflect routing or client issues.

Pro tips

Slice errors by endpoint and scenario to locate the root cause quickly. A sudden 5xx spike combined with flattening throughput often indicates backend collapse.



4. Virtual User Metrics (Arrival, Termination & Concurrency)

What it is

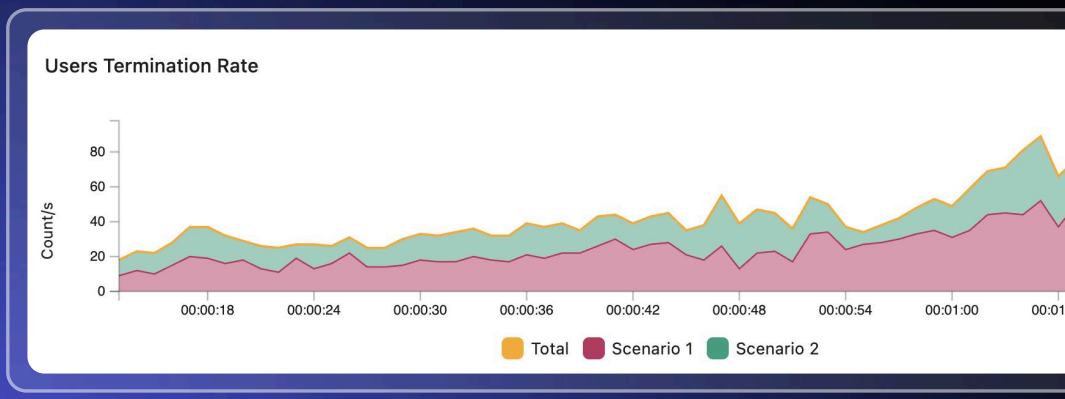
Tracks how many users start (arrival), finish (termination), and are active simultaneously (concurrency) during the test.

What it reveals

These metrics give a clear picture of load patterns and system response. Diverging arrival and termination curves often indicate slow transactions, timeouts, or abandoned sessions.

Pro tips

Use concurrency alongside response percentiles to pinpoint capacity ceilings. Unexpected drops in termination rate usually point to blocking issues.



5. Group Duration Percentiles

What it is

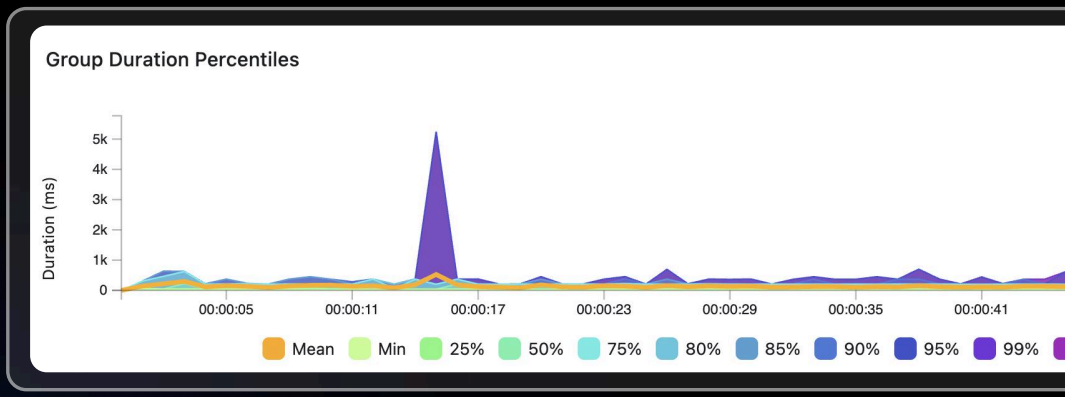
Measures the end-to-end duration of defined groups of requests, typically mapped to user journeys or business transactions (e.g. checkout, search, login).

What it reveals

Group metrics expose compounded latency across multiple services and steps, which single request timings can miss. A single slow API might not break SLAs, but multiple small delays add up quickly.

Pro tips

Use group percentiles to benchmark real business flows. This helps prioritize optimizations that have the most customer impact.



6. TCP Connect Duration Percentiles

What it is

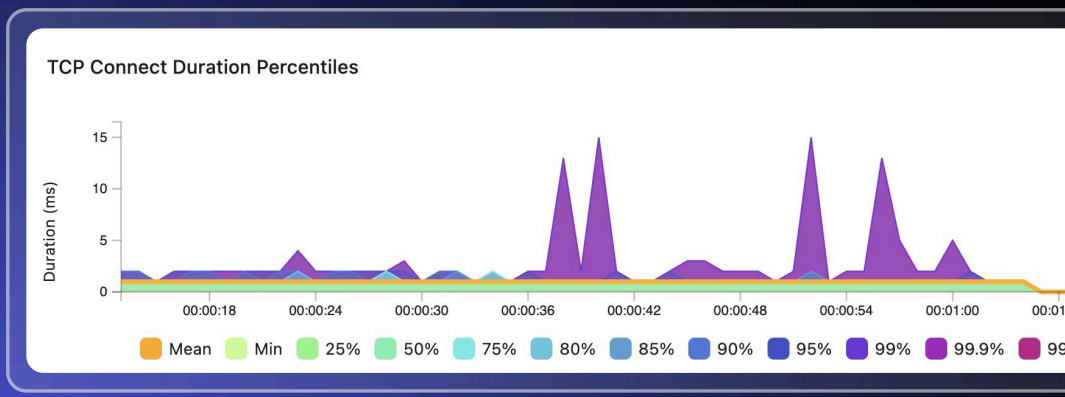
Measures the time taken to establish TCP connections from the load generator to the system under test.

What it reveals

Rising TCP connect times typically point to network or infrastructure bottlenecks (e.g. load generator saturation, exhausted connection pools, firewall limits) before application performance visibly degrades.

Pro tips

Watch for increases in TCP connect percentiles during ramp-up. If they rise before response times do, the issue is likely outside the app layer.



7. TLS Handshake Duration Percentiles

What it is

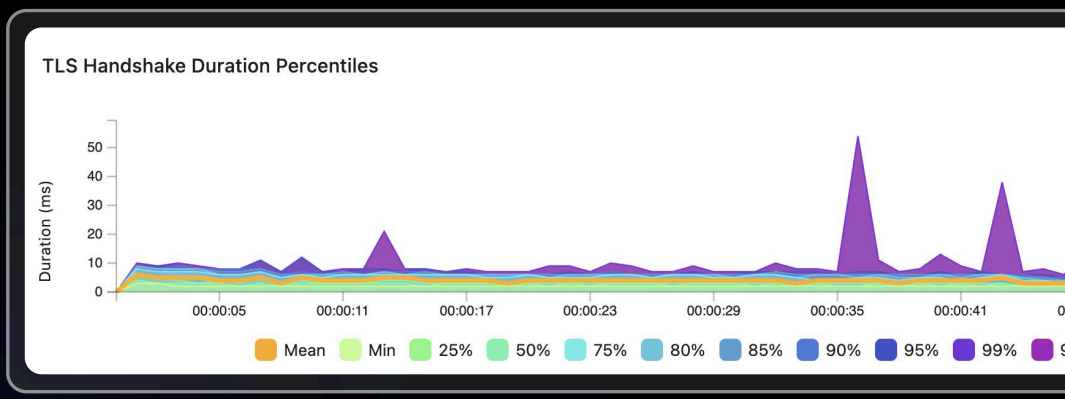
Tracks the duration of SSL/TLS handshakes during secure connection establishment.

What it reveals

Handshake latency often becomes significant at scale, especially with CPU-heavy ciphers or certificate chain issues. Sudden increases during high concurrency tests can add unexpected overhead.

Pro tips

Baseline TLS handshake times early, and watch for spikes under load. These are often invisible in APM tools but noticeable in Gatling's client-side view.



8. Load Generator Infrastructure Metrics (CPU & Heap)

What it is

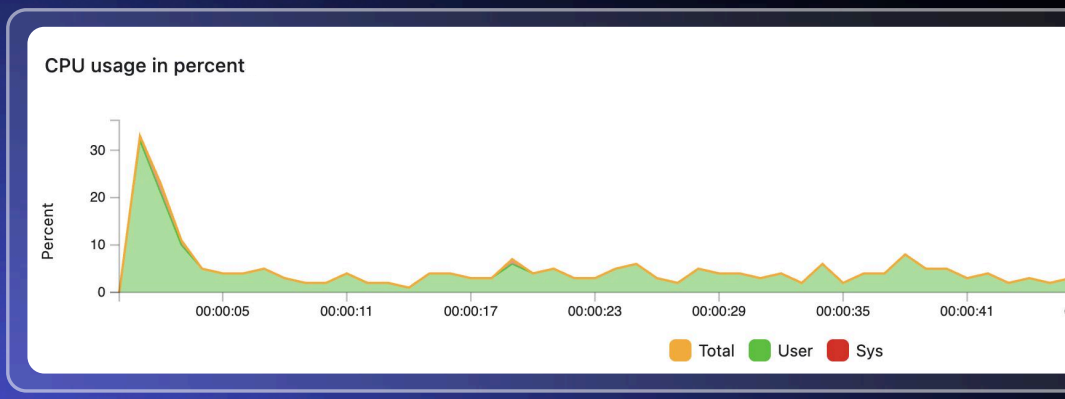
Monitors CPU usage and memory consumption on the load generators themselves to ensure the test infrastructure can sustain the desired traffic.

What it reveals

If generators hit CPU or heap limits, they can become the bottleneck, introducing artificial latency or failing to maintain target injection rates.

Pro tips

Always check generator health before blaming the system under test. Sustained CPU above 80% or heap usage above 85% can skew results dramatically.



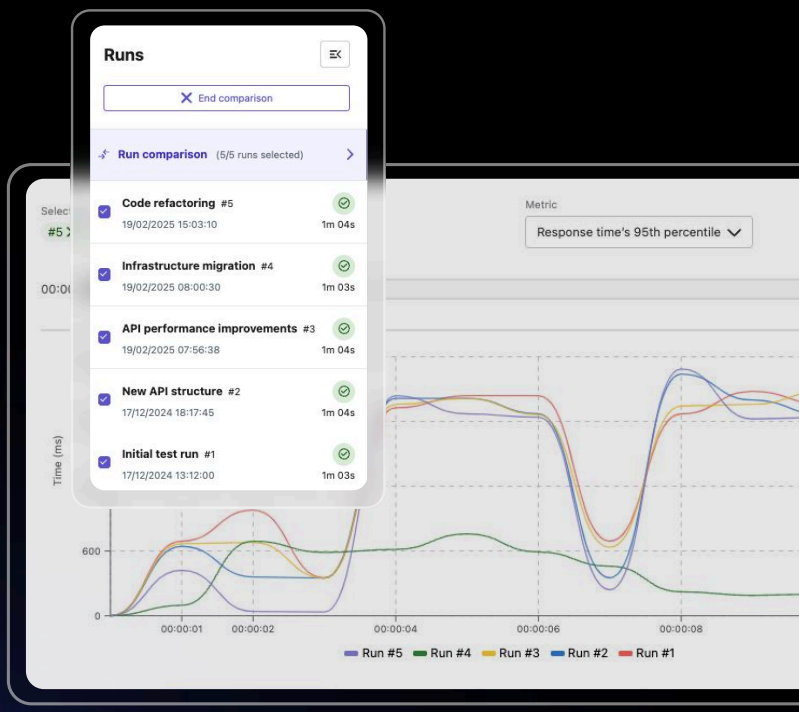
How to go further

Once you've established a strong foundation by tracking key load testing metrics, the next step is to connect the dots between individual test results, long-term trends, and production telemetry. This is where Gatling Enterprise Edition really shines.

Observe trends over time

Load testing isn't just about validating a single release, it's about understanding how your system evolves under pressure. By analyzing performance trends across multiple runs, teams can detect subtle regressions, capacity drifts, or architectural bottlenecks that might otherwise go unnoticed.

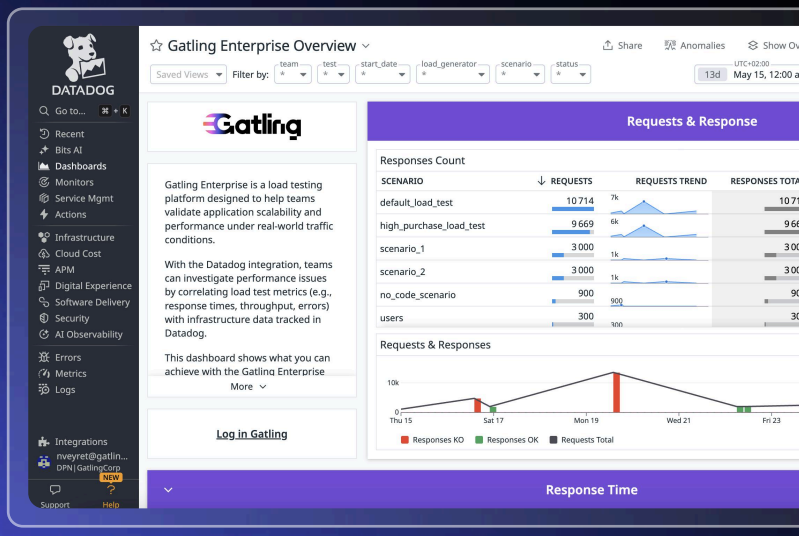
With Gatling Enterprise Edition's built-in dashboards, you can **compare test runs side by side**, track key metrics over weeks or months, and spot performance degradation before it turns into production incidents. This historical view gives engineering and QA teams the data they need to make informed, proactive decisions instead of reacting after failures occur.



Correlate load testing with APM & observability data

Performance metrics gain their full diagnostic power when combined with infrastructure and application telemetry. Gatling's **native integrations with Datadog and Dynatrace** allow you to overlay load testing data with real-time observability signals from your production environment.

This correlation helps teams pinpoint the exact cause of slowdowns or errors: whether it's inefficient code, network saturation, database contention, or infrastructure limits. It also aligns developers, SREs, and QA teams around a single shared view of system behavior, reducing friction and speeding up root-cause analysis.



If load testing matters, your tool should too.

Your users demand speed and reliability, your load testing platform must deliver the same, with power, precision, and scalability. **Gatling Enterprise Edition** offers a complete platform built for modern teams, distributed systems, and real-world performance demands, aligned along five product pillars:

Analyze smarter & act faster

Gain real-time visibility with dashboards, trend comparisons, and actionable insights.

Create tests your way

Build tests via code, low-code, or no-code, import Postman, script in JS/TS or Java, or design visually.

Unlock automations

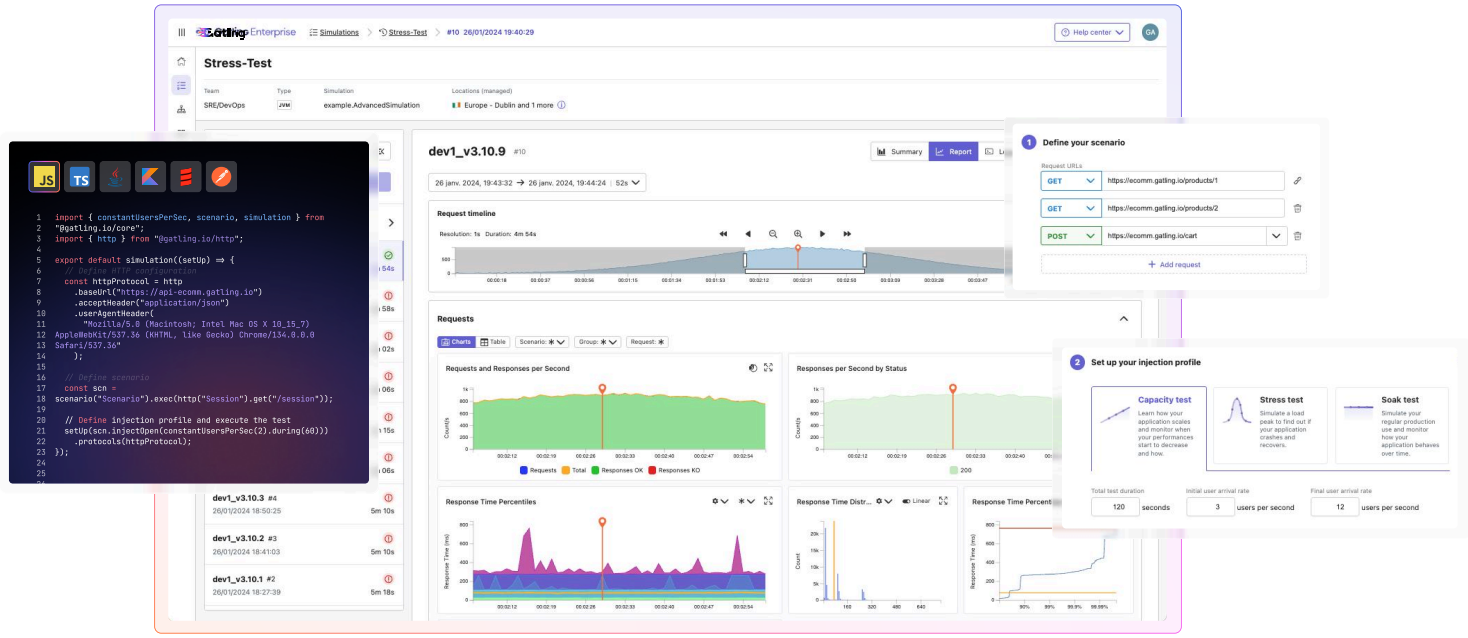
Trigger simulations via CI/CD or API, apply stop criteria, and gate releases with performance thresholds.

Collaborate & share results easily

Use RBAC, SSO, quotas, and shared reports. Share results via Slack, Teams, or Jira.

Deploy load generators anywhere

Run tests from Gatling managed regions, your cloud, or on-prem.



Gatling

Gatling is the leading solution for modern load testing, enabling developers and organizations to deliver fast, reliable applications at 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 evaluate Enterprise Edition?

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

Talk to an expert ›