Gatling

2025

DATASHEET

Leveraging Gatling for GitOps-driven load testing

Automate test environments, enforce performance gates, and deliver reliable applications at scale with Gatling Enterprise Edition.

LEVERAGING GATLING FOR GITOPS-DRIVEN LOAD TESTING

Modern DevOps teams have automated nearly every part of their delivery pipeline: Build systems, deployments, observability, and even infrastructure provisioning.

Yet performance testing often remains the exception: manual, detached, and inconsistent.

As organizations adopt GitOps, it is time to extend the same principles (version control, declarative state, automation, and auditability) to load testing.

Gatling Enterprise Edition brings performance testing into the GitOps era.

By combining test-as-code principles with enterprise orchestration, Gatling Enterprise Edition enables platform, QA, and SRE teams to improve performance testing across clusters, CI/CD pipelines, and global regions.

It transforms tests into first-class citizens of your delivery workflow: written as code, versioned in Git, and executed automatically in CI/CD pipelines. Meanwhile, load-testing infrastructure is managed declaratively through Infrastructure-as-Code.

The result: safer releases, faster feedback loops, and measurable reliability gains across the entire delivery lifecycle.

This datasheet explains how adopting GitOps-compatible load testing enables platform teams to continuously verify reliability and scalability without slowing down delivery velocity or developer feedback loops.

GitOps: From infrastructure to Everything-as-Code

GitOps is not a product. It is a software operations model built around a simple idea: Git as the single source of truth for everything that defines a system.

In practice, every piece of operational state (application configuration, infrastructure resources, policies, and deployment definitions) is stored as code in Git.

Each modification follows the same lifecycle as application code: it is proposed through a pull request, reviewed, and applied automatically to target environments by a reconciler or CI/CD system.

The result is an auditable, version-controlled, and self-healing system that continuously converges toward its declared desired state.

GitOps has already transformed how modern organizations manage infrastructure. The same approach can now be applied to performance validation, ensuring that scalability and responsiveness are verified continuously as part of the same lifecycle that governs build and deploy steps.

Why GitOps for load testing?

Traditional load testing models are incompatible with modern continuous delivery.

They depend on manual execution, isolated tooling, and static staging environments that fail to reproduce real-world conditions. The result is brittle feedback loops, false positives, and performance regressions discovered too late in the cycle.

A GitOps-compatible approach to load testing solves these problems by applying the same rigor that governs deployment automation:

Versioned

Simulations, configurations, and thresholds are stored in Git and reviewed like any other source artifact.

Automated

Each merge, tag, or deployment trigger can start performance tests automatically in CI/CD.

Declarative

Test environments, infrastructure, and datasets are provisioned as code, ensuring reproducibility.

Observable

Results are pushed automatically to APM and observability tools, dashboards, and monitoring systems, closing the feedback loop.

Core capabilities that enable GitOps with Gatling Enterprise Edition

Test-as-Code

In Gatling, simulations are written as code using JavaScript, TypeScript, Scala, Java, or Kotlin.

Each simulation explicitly describes:

- The user journey and sequence of requests,
- Data feeders for parameterization,
- The injection profile defining virtual user load and arrival rate,
- Assertions that define acceptable performance thresholds.

By describing test logic as code, load testing inherits the same collaboration and governance as software engineering.

Key advantages:

- **Traceability:** Every change is linked to a commit, providing full history and rollback capability.
- **Collaboration:** Developers, SREs, and QA engineers can review and test changes through standard Git workflows.
- **Reproducibility:** A simulation behaves identically whether it runs locally, in CI, or in distributed infrastructure.
- **Automation:** Tests can be triggered automatically on pull requests or before deployments.

The result? Your load tests evolve at the same speed as your codebase, remain reliable and reviewable, and stay aligned with your release cadence.

Configuration-as-Code

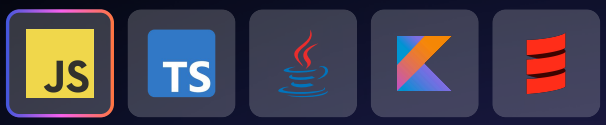
Beyond simulation logic, Gatling Enterprise Edition allows teams to define test configurations declaratively using a descriptor file (typically .conf).

This file acts as a manifest capturing the entire execution context of a test, including:

- The simulations and packages to include,
- Load profiles, durations, and pacing,
- Environment variables such as API keys, base URLs, or credentials,
- Load generator topology, regional distribution, and resource allocations,
- Input datasets or feeder definitions.

When this configuration is stored in Git, every test becomes versioned, reviewable, and reproducible. It guarantees consistent behavior across developer workstations, CI pipelines, and pre-production environments, eliminating the “works on my machine” syndrome.

The result? configuration consistency becomes part of your engineering discipline. Each simulation runs predictably, regardless of where it is executed.




```
import { constantUsersPerSec, scenario, simulation } from
"@gatling.io/core";
import { http } from "@gatling.io/http";

export default simulation((setUp) => {
  // Define HTTP configuration
  const httpProtocol = http
    .baseUrl("https://api-ecomm.gatling.io")
    .acceptHeader("application/json")
```

With Gatling Enterprise Edition you can scale the same simulation logic across regions, teams, or environments while maintaining a single source of truth.

Core capabilities that enable GitOps with Gatling Enterprise Edition

Build from Git

In a GitOps workflow, the repository defines everything necessary to build, test, and deploy. Gatling Enterprise Edition extends that model to performance validation.

With Build from Git, Gatling Enterprise Edition retrieves simulations and configurations directly from your repository (GitHub, GitLab, Bitbucket, or equivalent) without any packaging step.

The process is simple:

1. Reference the repository and branch within Gatling Enterprise Edition.
2. On each commit or merge, the system pulls the source, builds the artifact, and runs the test in the defined environment.
3. Assertions and stop criteria enforce pass or fail logic automatically.

Each run can be tied to a specific Git commit when using Build from Git, ensuring full traceability between performance metrics and the exact source code that produced them.

The result? a continuous feedback loop that aligns code changes, infrastructure updates, and performance outcomes within the same version control lifecycle.

Infrastructure-as-Code

Scalable performance testing depends as much on disciplined infrastructure management as on test logic.

Gatling Enterprise Edition allows you to manage load testing infrastructure using Infrastructure-as-Code (IaC) with the same declarative tools you already use for production systems.

You can provision and manage:

- Private locations for secure, self-hosted load generation,
- Load generators across multiple cloud regions,
- Network and resource specifications such as CPU, memory, and bandwidth allocation.

IaC definitions can be maintained in Terraform, AWS CloudFormation, or Helm charts for Kubernetes.

This model enables ephemeral test environments, short-lived isolated setups created automatically to validate a pull request or pre-production deployment. Once tests complete, the environment is destroyed.

The result? predictable, cost-efficient, and policy-compliant test infrastructure that fits seamlessly within GitOps delivery patterns.

```
module "location" {
  source          = "git::https://
github.com/gatling/gatling-enterprise-
control-plane-deployment//terraform/aws/
location"
  id              = "prl_aws"
  description     = "Private Location on
AWS"
  region          = "<Region>"
  subnets         = ["<SubnetId>"]
  security-groups = ["<SecurityGroupId>"]
}
```

Combined with Gatling Enterprise Edition orchestration, IaC ensures that the same code which provisions environments also executes tests and collects telemetry.

GitOps-ready continuous load testing

Traceable, observable, and auditable results

Every run is linked to a specific commit (when using Build from Git) and visualized in Gatling Enterprise dashboards.

These dashboards include trend and comparison views, enabling teams to track regressions, measure performance evolution across branches, and verify compliance with SLOs.

Gatling also integrates with major APM and observability platforms such as Datadog and Dynatrace.

These integrations make it possible to correlate load test metrics with real infrastructure telemetry, accelerating root-cause analysis and reducing mean time to resolution.

Seamless CI/CD automation

Gatling integrates natively with GitHub Actions, GitLab CI, Jenkins, and other automation systems.

Test execution can be configured declaratively in YAML files or triggered via API.

Typical usage patterns include:

- Post-deployment smoke tests,
- Nightly regression tests,
- Pre-release stress validation before production rollout.

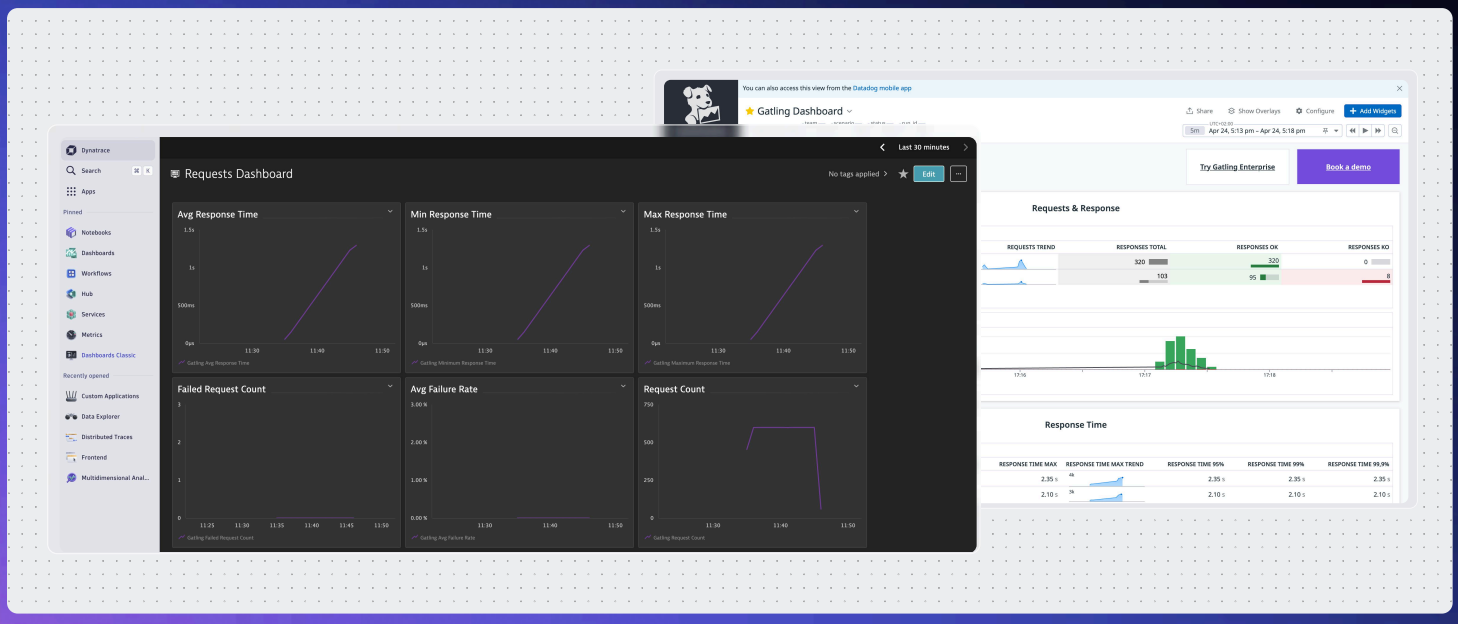
Assertions and stop criteria ensure that only performant builds are promoted.

Declarative performance governance

Tests are controlled by assertions, which are performance thresholds defined as code.

They can target latency percentiles, throughput, error rates, or request success ratios. When thresholds are violated, the pipeline fails automatically, blocking promotion to the next stage.

This transforms performance testing into a governed policy, versioned and enforced like other compliance checks.



With Gatling Enterprise Edition performance testing becomes continuous, declarative, and fully traceable, with feedback loops that reach both developers and observability systems.

Choosing the right platform for modern load testing

Embedding load testing into the GitOps lifecycle transforms performance validation from a manual activity into an automated, measurable discipline. With Gatling Enterprise Edition, simulations, configurations, and infrastructure definitions are all expressed as code, versioned, and governed through the same workflows that manage application delivery.

Results are observable and auditable, providing end-to-end traceability from commit to performance impact. The outcome is a delivery pipeline where reliability grows in parallel with speed. Each release is deployed with confidence, because performance verification is no longer an afterthought but a built-in step of your continuous delivery process.

Analyze smarter and 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 automation

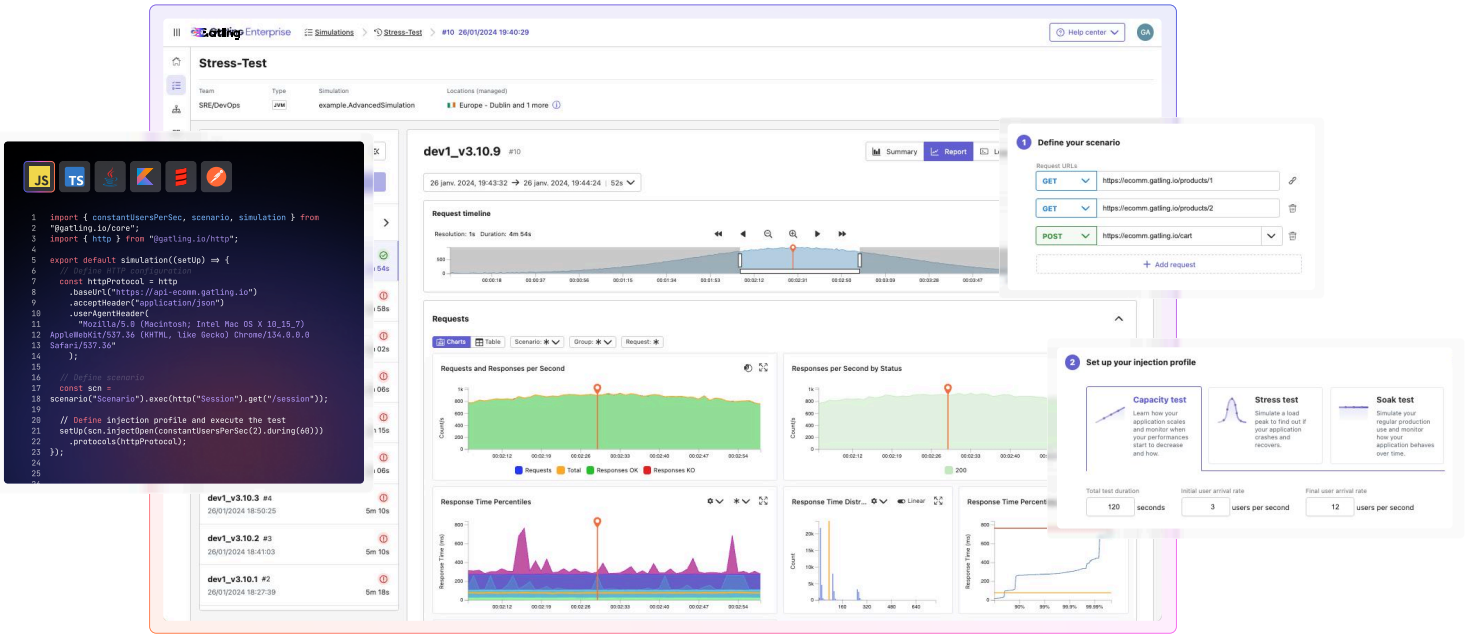
Trigger simulations via CI/CD or API, apply stop criteria, and gate releases with performance thresholds.

Collaborate and 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 ›