WEBSCALE

DATASHEET

Webscale Cloud Application Testing

Proactive Readiness for Critical Shopping Events

The sudden growth in ecommerce has made it difficult for application owners to predict traffic volumes to their online storefronts, and this is likely to continue in 2021 as well. To be ready for the holidays, or any special marketing event, merchants need to understand their application's readiness to handle increased surges of traffic. Research shows that online consumers go elsewhere if they deem a site too slow (taking more than 3 seconds to load) or if they experience availability issues, such as a failing checkout. While the Webscale platform provides multiple caching layers and proactively auto-scales application servers, there are numerous architecture and code dependencies that impact a site's ability to scale under significant load.

It is therefore of critical importance to understand:

- The impact of site changes to functionality, availability and performance.
- Site response times under heavy load.
- For new code deployments, what site functionality is slower than expected?
- What pages within the new build have slowed to the point where they may impact search rankings?
- At what point, if any, do errors start appearing for the users?
- How does the user's web experience change under increasing load?

Integrating application testing into a well defined CI/CD process of build-test-deploy is important for any business-critical site or application. Site owners should conduct a thorough evaluation of their site's performance through multiple checkpoints, such as new code roll outs, platform updates, or additions or changes to plug-ins or functions. These tests should be carried out frequently, and at peak demand, to ensure that any critical issues, particularly around availability and scalability, are avoided. It is especially critical to do all of the above before the high traffic shopping events, which is why Webscale introduced Cloud Application Testing.

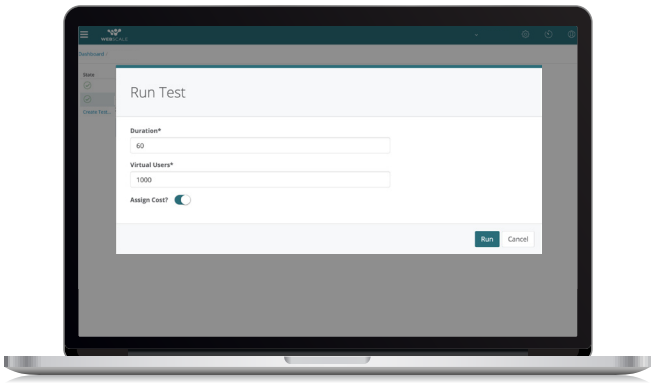
Webscale Cloud Application Testing

Webscale's Cloud Application Testing is run as a synthetic test across any staging or cloned production site, hosted in any cloud provider or on-premise data center. The purpose of the service is to simulate end-user patterns and shopping behavior on the application, especially right after new code deployments, and to stair-step into higher traffic volumes at a higher scale, while measuring the site's behavior, responsiveness, and overall performance.

How it Works

Webscale Cloud Application Testing executes a series of tests against a staging site or, in the case of a Webscale managed application, a Webscale hosted clone of your live production website. User sessions are replicated to make the same API calls that real production users make, simulating site traffic down to the last detail for logged in and guest users, all the way through a successful checkout experience, including third party API calls, if relevant. The service then captures these results using the Webscale Customer Portal, with the traffic logs available to view in the Webscale Traffic Viewer, for as long as the temporary testing application is available. A report is also made available, on-demand, to describe the tests, the patterns, and the results.

Tests can be defined by metrics such as “Duration” and “Number of Users”, depending on the subscription. At Go-Live, the Webscale provisioning team can pre-populate a set number of tests and make them available to our customers to run at specific intervals and ensure site stability.



Benefits

Easy-to-deploy: Initiate a test in just a few clicks and run it for a few minutes or longer.

Repeatable: Replay your best sales event by incorporating the test as a code in your CI/CD pipeline.

Scalable: Seamlessly scale to thousands of requests per second and concurrent users.

Cloud-native: Built for storefronts on any platform and any public cloud.

Enhance user experience: Evaluate performance under load to ensure optimum customer experience at all times.

Mitigate risks related to code release: Real-world testing of code to ensure successful deployments.

Process and Metrics

The system as originally deployed is tested for capacity using anonymous users (non-logged-in users). This is a replay of accelerated traffic with add-to-cart and checkouts. This establishes a baseline capacity from the system, can identify code and infrastructure deficiencies during load, and captures the bulk of load generated by high user traffic. The testing tool also adds the ability to have logged-in users, add-to-cart, and a high volume of checkouts to the baseline traffic.

Several key measurements are recorded during each test, with a goal of identifying the average response time to the requests submitted and determining what issues may be causing any anomalies in the results.

To ensure your storefront can not only handle a surge in traffic but also continue to process large volumes of simultaneous checkouts, Webscale can also carry out testing on your shopping cart.

Some of the tests carried out include:

Increasing the load test volume by enhancing user sessions and executing multiple test runs on the access logs provided.

Monitoring CPU load on the database.

Generating a checkout test by recording an actual browser session where a product is added to the cart, checked out, the payment POST request is successful, and checkout success page* is reached with a 200 response code.

Monitoring for any scaling issues, and ensuring that application response time remains within test parameters.

Performing tests as if all traffic has been served from the origin. Once the site is live, there will be less traffic coming to the origin due to CDN offloading.

Amplifying traffic by a minimum of 10x during the overall test using a sliding scale.

Setting response time goal and monitoring total requests made, and average response time during the test.

Setting checkout rate according to, and beyond, peak checkout rates identified from past Black Friday and Cyber Monday sales events.

Ensuring testing is limited by traffic generation only, and not infrastructure capacity.

Results will include the number of sessions of each type with the total session count, the duration, average response time, number of requests, status codes for each request, and a link to Traffic Viewer corresponding to the requests made by the load test. A breakdown of each type of model executed during the test is also made available.

* In order to facilitate checkout page load test, the application will need to support non-payment gateways or “fake” non-fulfillment product SKUs