Tricentis

How to scale fast performing applications in the cloud without scaling your budget

Tricentis

Table of contents

THE NEED FOR A CLOUD-SPECIFIC PERFORMANCE TESTING STRATEGY

The faster an organization can conceive, build, and ship software, the more likely it is to succeed in today's digital economy. The cloud presents significant opportunities to achieve the required speed, whether you are building new applications or optimizing existing ones. According to a Flexera survey, in 2021 more than 50% of businesses globally reported their cloud usage spend would continue to increase.

But the cloud only operates and scales resources – it won't automatically make your applications fast, stable, and scalable. Performance testing is critical to identify and mitigate risks to your application, architecture design, and cloud hosting environment. Effective cloud performance testing requires tracking key performance indicators (KPIs) such as cost, usage, performance, availability, and security, as well as testing systems availability, reliability, response time, throughput, and capacity. A sound performance testing strategy should ultimately help you verify that your application meets the following requirements:

- **Performance:** ensuring an application is as fast when operated by many concurrent users as it is for a single user
- **Scalability**: the infrastructure's ability to successfully adapt to usage increases
- **Stability:** ensuring that changes in usage, features, or infrastructure won't cause distributed ripple effects that result in severe incidents and unanticipated costs for an organization

Two key dimensions of cloud performance testing

Your testing strategy should help you understand the performance-related characteristics and implications of your application architecture, whether you are testing a distributed or microservices system that was built in the cloud or a monolithic system that is being migrated to the cloud.

Migrate: Performance testing is critical to the success of any cloud migration project, whether you are planning a lift-and-shift migration strategy or a more complex re-platforming or re-architecting approach. Regardless of the approach, it is critical to assess the migration's impact on the application's performance, particularly bandwidth and latency, to ensure an application is prepared for its new cloud home.

Build: While many traditional performance testing principles apply, building a performance strategy for cloud-native applications requires testing teams to consider how performance will be impacted by millions of concurrent users coming from multiple geographical locations, variances in load, and more.
In the following white paper, we'll cover what you need to consider when creating a cloud performance testing strategy – whether you're migrating, building, or both – and share strategies that will help you scale application performance without scaling your budget.

CLOUD PERFORMANCE: CONSIDERATIONS AND IMPLICATIONS

Scaling fast-performing applications in the cloud introduces new considerations and implications that should be factored into the testing process.

Consider cloud architecture-related changes

Though it may seem that moving to the cloud and autoscaling systems will decrease or eliminate the risk for performance and stability, this isn't the case. Rather, it merely shifts this risk elsewhere. When considering the value and goals of in-cloud testing, it's important to keep in mind that your system is only as strong as its slowest component.

It's also important to note that **scaling in the cloud isn't equivalent to improved performance**. While the cloud can scale resources infinitely if you have an unlimited budget, infrastructure alone won't guarantee your apps perform well. An application that performs sluggishly for one user won't be improved by scaling the load.

In distributed architectures, not everything is configured by default appropriately for each service or component. Distributing services across cloud resources and network boundaries introduces new latencies you likely have never experienced in data centers or in single-region, lift-and-shift migrations.

Scalability isn't free

The fact is that scalability comes at a cost. Scale can't replace poorly performing code, and autoscaling merely provides a Band-Aid while hiding performance bottlenecks. This will ultimately cost more in autoscaling infrastructure and defeat the purpose of cloud optimization in the first place, limiting the value of the cloud.

Though you may be able to offload many of your hosting management issues, cloud hosting won't address problems in the underlying code of the app that hinder performance.

Solving the issue at its root cause will help your app run efficiently, diminishing the size of capacity swings and lowering the cost of autoscaling.

Another major consideration is identifying where you don't have strong enough IT controls and response plans for scalability versus cost-acceptable tolerances.

It takes a lot of time and energy to understand how well your cloud applications and systems are performing, using resources, spending, etc., and to harmonize the impact of scalability on revenue versus

the cost of scaling. In fact, there are tools on the market, such as VMWare's CloudHealth, designed to help organizations understand this. Truly understanding cloud performance and the value of your efforts requires monitoring and measurement along multiple dimensions.

Observability is required

You can't measure what you can't see, and a lack of good information prevents informed decisions. Without observability in testing, scaling in the cloud causes KPIs that used to be within arm's length to become out of reach.

Observability is particularly important when it comes to release readiness. For example, how do we know that the new features on a company's homepage won't significantly impact users and infrastructure? It's also critical to consider the impacts of load on distributed systems that can't be measured virtually. Without including sound observability practices pre-production, you run the risk of only identifying issues when they arise in production environments – without the necessary insight to understand what caused the issue in the first place.

Don't confuse the simplicity of using managed services with a lack of complexity. The complexity you had before moving to the cloud is still there. It has merely shifted, now to a black box. When infrastructure shifts, measurement mechanisms shift too, often leaving you out of the loop on existing KPIs that weren't re-mapped to new architectural stress points.

Single points of failure must be addressed

A single point of failure (SPOF) can cause a whole system to go down. Identifying and minimizing this risk should be a major consideration in the testing process. Entire regions, not just zones, can go down, even if you cluster across availability zones.

For example, the Amazon Web Services cloud experienced multiple outages in multiple regions of the United States in December 2021. This in turn impacted many services hosted on AWS including Netflix, Slack, Ring, Hulu, Disney+, Zoom, and DoorDash, among others.

Especially in cloud-native services, over time distributed systems unintentionally introduce architectural assumptions and SPOFs due to usage upticks, cloud downtime, and insufficient multi-region distribution. Unfortunately, organizations often only spot these issues when it's too late.

The question you must ask yourself is: How do you proactively validate where the SPOFs are in your cloud architectures without putting pressure on the system under test or your revenue-generating users?

A multi-cloud approach isn't simple

Implementing a multi-cloud strategy, or the use of more than one cloud vendor, has many advantages in terms of flexibility and optimization. Since 2019, statistics indicate that IT leaders have consistently used a multi-cloud approach for workloads to leverage different application services and guarantee availability. It's likely that a multi-cloud architecture in which application workloads can move freely across clouds will become the dominant approach in the near future.

However, clouds are rarely identical.

Different cloud environments have different infrastructure performance characteristics. Each will have its own unique KPI sources measurement mappings, and dashboards. Thus, different automated deployment semantics and parameters lead to fragmented implementation of performance configurations.
Multi-cloud scaling also poses potential risks to areas such as cost management, security, and data, which highlights the need for software testing to address multi-cloud risks.

App regression introduces unique cloud migration challenges

Directly after initial migration or launch into the cloud, continuous development still goes on. Potential regression is an important component to track during this process, to ensure that changes to existing applications don't break your present functionality.

Cloud–based application development is a different stack and modern approach that comes with its own unique challenges for technology leaders. While a containerization approach helps package deployment motions, it also introduces a new layer of virtualization – which means new dynamics to contend with.
For instance, when others on the cloud use available resources, it can impact your cloud performance. You may experience a failure that isn't due to your own activity. This is known as "noisy neighbors." Noisy neighbors can degrade your cloud performance not just in your own clusters and Virtual Private Clouds (VPCs), but due to public (and sometimes even private) cloud infrastructure.

In addition, without years of prior experience running large-scale, container-based architectures, improper container resource limits and missteps in your autoscale strategy can easily occur.

Using the right testing tools and strategies can help address these concerns and eliminate potential missteps as your organization scales its applications to the cloud.

STRATEGIES FOR APPROACHING PERFORMANCE TESTING IN THE CLOUD

The right cloud performance testing strategy should ensure that your applications run smoothly and as intended within the cloud, and that you can scale successfully without impacting your users or having to scale your budget as a result of missteps. Here are the key elements of a successful cloud performance testing approach.

Step 1: Solidify your cloud migration strategy

Baseline comparisons: The first step of migration testing is to establish your performance testing baselines, and ensure no regressions occur against your Service Level Objectives (SLOs). You can also reuse existing on-premises baselines for new cloud environments.

Resource measurements: Ensure that you have access to and are competent with changes in how KPIs are reflected and measured in cloud systems.

Reuse test assets: It should not be difficult or toilsome to use the same test assets between lower and higher cloud pre-production environments. This should be a key part of your continuous performance and automated pipeline strategy.

Profile network emulation: Determine what impact additional latency and potential bandwidth constraints will have on existing architecture if it was moved to a cloud as-is.

Step 2: Develop a cloud-native strategy

Cloud-based software development: Use the same modern asset management systems (Git and Artifact Repositories) for performance testing assets and resources, as well as the same versioning, branching, and access strategies in application and service development.

In addition, run performance sanity checks locally before incurring CI pipeline-based performance feedback loop infrastructure and time costs. It's critical to trigger the right-sized performance testing at the right time using automated pipelines for each team, product, or service.

Cloud-based deployment: Change only details that matter when transitioning testing from lower to higher environments, such as service or server names, service-level agreements (SLAs), and test data sets. Switch monitoring technologies based on which cloud and resource types while preserving these in a standardized performance platform. Use the same testing semantics between multiple pipeline types or deployment scripting strategies, as not all teams use the same CIs and infrastructure.

Cloud-based performance and measurement: Automate the process of generating and communicating your baseline to current comparisons on every build, deploy, or release. This should include deep and wide telemetry of system and service resources in every test by default. Allow all teams to act on subject-matter summarizations, but provide self-service dashboards so that teams can analyze and drill down when they need to diagnose causes of automatically identified issues.

Step 3: Adopt cloud-enabled performance testing

Use the cloud to make load testing easier and cheaper: Performance has traditionally been the most complex, time consuming, and expensive part of testing. With a cloud-enabled performance testing solution, you can create test environments with your entire infrastructure, provision them, and shut them down all within a few hours at a relatively low cost. Because cloud architecture is ever-evolving and continuously impacted by other services and APIs, it is critical to develop a cloud-ready, automated performance testing strategy to continuously measure and observe the impact of changes so you can keep applications running smoothly.

ADOPTING CLOUD-ENABLED PERFORMANCE AND LOAD TESTING SOLUTIONS

Enabling a cloud-specific, automated approach to performance and load testing provides a myriad of benefits in the delivery of scalable, fast-performing applications running on cloud infrastructure. Adopting cloud-ready, SaaS-based performance testing software such as Tricentis NeoLoad can both streamline operations and help keep spending in check, as well as:

- **Eliminate the IT bottleneck:** With NeoLoad SaaS, infrastructure is available and automatically configurable, and is ready to be turned on or off as needed. Because this is a self-service process, performance testing teams no longer have to wait for IT to procure load testing infrastructure.

- **Decrease infrastructure costs:** With NeoLoad SaaS, you can use only what you need, with no more always-on load testing. Infrastructure is the costliest part of performance testing, not only procuring but also maintaining, upgrading, provisioning, and tying machines up all day and all week for tests that only run for a couple hours here and there. Infrastructure as a Service (IaaS) for cloud computing greatly reduces this cost.

- **Establish earlier and real-time feedback loops:** Automating performance testing in cloud CI pipelines means the impact of changes will always be accounted for, limiting surprises in production. It also means you'll catch bugs early, when they're easier to fix and less complex.

Geo test: Initiate load testing with IaaS from any location with access to a browser.

Test in a realistic environment: Test different requests and loads from different locations across hybrid systems.

Gain multi-endpoint test flexibility: There are many different ways to enter a system or application, including via APIs, mobile, and various other device types. With Tricentis NeoLoad, you can connect to and test across them all.

Many organizations are seeing the benefits of cloud migration. Yet roughly one-third of a company's IT budget commonly goes toward cloud services. Mismanaged IT projects can create huge overages that quickly add up. The good news is that well-managed cloud migration will save a company an average of 15% on their IT budget. With a sound performance testing strategy and the right tools, you can achieve your migration and delivery goals in complex cloud environments and maximize savings at the same time.

ABOUT TRICENTIS

Tricentis is the global leader in enterprise continuous testing, widely credited for reinventing software testing and delivery for DevOps and agile environments. The Tricentis AI-based, continuous testing platform provides automated testing and real-time business risk insight across your DevOps pipeline. This enables enterprises to accelerate their digital transformation by dramatically increasing software release speed, reducing costs, and improving software quality. Tricentis has been widely recognized as the leader by all major industry analysts, including being named the leader in Gartner's Magic Quadrant five years in a row. Tricentis has more than 1,800 customers, including the largest brands in the world, such as Accenture, Coca-Cola, Nationwide Insurance, Allianz, Telstra, Dolby, RBS, and Zappos.

To learn more, visit www.tricentis.com or follow us on LinkedIn, Twitter, and Facebook.

AMERICAS

2570 W El Camino Real,
Suite 540
Mountain View, CA 94040
United States of America
office@tricentis.com
+1-650-383-8329

EMEA

Leonard-Bernstein-Straße 10
1220 Vienna
Austria
office@tricentis.com
+43 1 263 24 09 – 0

APAC

2-12 Foveaux Street
Surry Hills NSW 2010,
Australia
frontdesk.apac@tricentis.com
+61 2 8458 0766

v. 0222