



# Auto Scaling Selenium Tests with KEDA - A New Approach to Driving Testing Efficiency

# Contents

**01**

Introduction

**02**

What is a Selenium Grid?

- The Issue with Selenium Grid Scaling

**03**

What is KEDA?

- Solving the Auto Scaling challenge with KEDA

**04**

The Benefits of Auto Scaling with KEDA

**05**

Achieve testing excellence on Selenium Grid with SLK's approach

# Introduction

Software testing becomes a critical business process in a world where software is the backbone of most products and services. As applications get more complex and use-case scenarios multiply, manual testing is no longer an option – it is tedious, time-consuming, error-prone, and limited in scale and scope. Spending valuable time in manual testing could derail launch timelines, limit releases, and impact quality and customer experience.

Organizations know this and have been actively looking for ways and tools to automate testing. Over 40% of organizations consider test automation an important aspect of their quality assurance process<sup>1</sup>. One of the preferred solutions for automated testing is Selenium. This open-source testing suite allows testers to run error-free tests much faster across a large permutation of browsers and operating systems. These advantages make Selenium Grid (a tool from the Selenium suite) the testers' go-to tool for running Selenium tests across web browsers and operating systems.

## What is a Selenium Grid

A Selenium Grid is a tool within the popular open-source Selenium testing suite. It allows users to run their Selenium tests on various environments, including different versions of web browsers and operating systems. Such testing helps ensure the proper functioning of web applications across a wide range of configurations. Selenium Grid makes it easy to run tests in parallel on multiple machines.

However efficient the Selenium Grid may be at automating tests; it lacks scaling capability. What if businesses wanted to scale the test to, say, 10,000 test cases? At best, Selenium Grid can allow enterprises to run tests in parallel by setting up a small cluster. But they will only ever be able to run tens of thousands of tests simultaneously. Finishing such a considerable workload at this speed can take between 7 and 10 days. That's way too long! In this article, we'll see how this can change.

## The Issue with Selenium Grid Scaling

Scaling with Selenium Grid entails increasing or decreasing the number of nodes (machines executing Selenium tests) in the Selenium Grid in response to fluctuating workloads (the number of pending test executions). Such manipulation is fraught with the risk of sub-optimal use of resources. Test executions are sluggish when too few nodes are available to handle the demand at that instant. Conversely, underutilization of resources occurs when too many nodes operate to take a relatively small workload.

Running tests on many different environments can be resource-intensive, particularly in the cloud, where resources are metered. Therefore, there is a need to use resources optimally. Optimizing the resources is the challenge in scaling with Selenium Grid. The solution lies in auto-scaling the tests. KEDA brings this to the table – balancing the resource and the workload -in real-time.

## What is KEDA?

KEDA is an open-source Kubernetes-based Event-Driven Auto scaler. It allows users to drive the scaling of any containerized application in a container management system (Kubernetes) based on the number of events needing to be processed. It's a tool that helps users automatically adjust the quantum of resources (like computers or servers) used for a task, depending on the amount of pending work in the queue.

## Solving the Auto Scaling challenge with KEDA

KEDA is an open-source Kubernetes-based Event-Driven Auto scaler. In the context of Selenium tests, users can use KEDA to scale the number of Selenium Grid nodes (machines executing Selenium tests) up and down as needed, depending on the number of pending test executions.

With KEDA, users can automatically scale Selenium Grids. The KEDA API or KEDA user interface lets them deploy a Selenium Grid cluster and configure KEDA to scale the number of replicas based on the number of pending test executions.

They can dynamically scale Selenium Grids using triggers, such as the number of open tasks in a message queue. KEDA will summon just the right amount of Selenium Grid resources to handle the event-driven workloads. No less, no more. KEDA changes the game by bringing automation to scaling.

**But how do users harness the power of KEDA to scale a Selenium Grid? They have to define two things:**

- First, a scaling target that specifies the number of replicas (nodes) that they want to maintain, and
- Second, an appropriate trigger to initiate scaling.

For instance, if the trigger is set as the number of pending jobs in a message queue when the number of pending jobs exceeds a certain threshold, KEDA scales the deployment up by creating additional replicas to handle the increased workload. Conversely, KEDA scales the deployment down by removing unnecessary replicas when the job pendency drops below a certain threshold.

## The Benefits of Auto Scaling with KEDA

Selenium Grid enables users to automate their tests. KEDA brings in the capability to automate the much-needed scaling. This combination further extends the benefits of testing with Selenium Grid alone.

How do users benefit from using KEDA to scale their Selenium Grids? There are several benefits, such as:



### Parallel Execution

Selenium Grid allows users to run tests concurrently on different machines. Therefore, users make significant savings in the overall testing time.



### Cross-platform Testing

Users can exploit the inherent cross-platform capabilities of Selenium Grid for testing their applications on a wide range of configurations.



### Distributed Testing

Selenium Grid enables users to spread their tests across multiple machines, making the testing process more efficient.



### Scalability

Users can benefit from KEDA's capability to scale testing infrastructure in alignment with varying workloads. They can ensure that the right resources are available to run tests efficiently.



### High Availability

With KEDA, users can ensure the increased availability of their testing infrastructure for tackling any unforeseen failures and disruptions.



### Optimized Cost

Since users can scale Selenium Grid clusters based on the number of pending test executions, ensuring that their workload utilizes the right amount of resources at a given time becomes possible. Such fine-tuning translates into lower overall cloud costs for running Selenium tests.



### Improved Efficiency

Auto-scaling Selenium Grid clusters in the cloud helps users run their tests as quickly as possible. They can minimize the time and system resources required for testing.



### Enhanced Compatibility Testing

By using KEDA auto scaler-assisted Selenium Grid clusters, users can conveniently test their applications in many environments, further improving the compatibility of their web applications.

## Achieve Testing Excellence on Selenium Grid with SLK's Approach

Although efficient, the KEDA-assisted auto-scaling of Selenium Grid isn't intelligent enough. Consider this. A user has 5000 test cases. They want to test them for two browsers, namely Firefox and Chrome. The KEDA-assisted auto-scaling process doesn't have the intelligence to decide an optimal number of browsers to open automatically. Instead, it runs all the 5000 tests on the two browsers in one go. It can't figure out the number and size of test batches - should batches be equal, should small batches be introduced first, etc. Similar considerations apply to the browsers when there are many to test on.

At SLK, we help businesses overcome this challenge and dramatically enhance the benefits of KEDA-assisted auto-scaling process. They can deploy our solution on their infrastructure. At the heart of our solution is an innovative use of determinants, such as how users scale it, i.e., how they decide to host their automated test grids or the number of browsers.

At a conservative estimate, users can save 35-40% on testing costs with our solution, which employs very lightweight Kubernetes containers. Additionally, users can significantly shorten their go-to-market time.

The cascading effect of this innovation will positively impact the ROI metrics, such as faster feedback cycles, higher developer productivity, higher savings in time, frequent release to production, and lower production incidents.



Want to know more about how we can help you level up testing efficiency on Selenium?

Write to us at [hello@slkgroup.com](mailto:hello@slkgroup.com)

## References

<sup>1</sup> <https://info.kobiton.com/hubfs/Mobile%20Test%20Automation%20Survey/State%20of%20Test%20Automation%202020-2021%20Edition%20-%20An%20Industry%20Survey%20by%20Kobiton.pdf?hsCtaTracking=3090d5a3-6b9a-4e0f-8b5e-099a9e8e4cbe%7Ca0eec9d5-023d-4c00-ab67-ef7b7a5e779c>

## About Us

SLK is a global technology services provider focused on bringing AI, intelligent automation, and analytics together to create leading-edge technology solutions for our customers through a culture of partnership, led by an evolutionary mindset. For over 20 years, we've helped organizations across diverse industries - insurance providers, financial service organizations, investment management companies, and manufacturers - reimagine their business and solve their present and future needs. Being A Great Place To Work Certified, we encourage an approach of constructively challenging the status quo in all that we do to enable peak business performance for our customers and for ourselves, through disruptive technologies, applied innovation, and purposeful automation. Find out how we help leading organizations reimagine their business at <https://www.slksoftware.com/>

© SLK Software Pvt. Ltd. 2023

[www.slksoftware.com](http://www.slksoftware.com)

