deviQA

10+	1M+	100+	300+
YEARS OF EXPERIENCE	HOURS WORKED	ENGINEERS	HAPPY CLIENTS

Case Study: Betting Application

Automated Testing, Performance Testing, Web Testing

Project Overview

What Betting Application had in the testing processes when they came to us and what they got after they started working with us.

Before improvement

- A lot of issues related to the real time update for bets.
- Failed expectations in the site performance between the expected quantity of the online users and real quantity of the online users

After improvement

- Developed a test suite that was able to generate test data based on the number of browsers in Kubernetes cluster.
- Test farm was developed to execute our tests in parallel from 200 machines.
- Major bottlenecks were found on the frontend, backend, and GraphQl server.
- Developed a solution that avoids login bottleneck to check that more than 10000 customers can work with the application in the same time
- Developed an automated test suite that can be easily expanded due to the requirements about the amount of online users
- Provided an exact amount of users that can be online at the same time in the application

5000+	1000+	200	3
THREADS FOR PERFORMANCE	TESTS DEVELOPED	MACHINES USED	FULL TIME QA AUTOMATION ENGINEERS

QA Team:
3 Full-Time QA Engineers

Project length:
1 month

TECHNOLOGIES & TOOLS

- Kubernetes
- Zalenium
- Selenide
- Java
- Junit
- Allure
- AWS
- EKS

The Challenge

One of our clients, which is a betting company, had strong needs to perform web application performance testing using 10000 "real users."

That means that each virtual user must do his activity in a real browser. Unfortunately, in this case, common performance tools like Jmeter, NeoLoader, etc. won't do the job.

Our main goal was to load the most critical actions that real users will heavily use. Users should do their activities stepwise, for example:

All users should log in during 5 minutes and wait. Once all users are successfully logged in, they should do bulk actions in the same period.

During testing, we faced a lot of issues related to the realtime update for bets. That stopped us from running performance tests, and we had to add complicated logic with different conditions to make our tests successful.

As we performed our test runs on production (this was the main requirement from our client), we had to add additional timeouts to avoid DDOS attack protection.

Another problem was to run all the tests from one machine. As a result, all requests were in the queue, and that was not what we need to do the performance testing.

Our clients were not able to stop deployment of new features, and as a result, we got updated for UI and backend two times a week that required to update our automation tests from time to time. Such activities slow down our performance.

Achievements

We decide to use Selenide automation framework for developing user actions and Zalenium to run all these huge numbers of tests. Zalenium is primely selenium grid but in Kubernetes.

Kubernetes was chosen to get the ability to orchestrate selenium nodes and scale them up quick. To generate such amount of users, we required a huge number of resources and lucky us AWS was able to provide it to us.

Our Kubernetes cluster contained 200 c5d.18xlarge nodes, each of them had 72 cores and 144 GB of RAM.

We have developed a test suite that was able to generate test data based on the number of browsers in Kubernetes cluster.

Test farm was developed to execute our tests in parallel from 200 machines.

Major bottlenecks were found on the frontend, backend, and GraphQl server. The client expected to support load for 5000+ users, but after the first test run the maximum number of users without downtime was only 717.

Services Provided

Our engineers developed more than 1000 test cases in addition to the 300 test cases from the client's in-house testing team.

Automated Testing →

3 Full-time automation QA Engineers were involved to the project.

Performance Testing →

Our QA Lead was involved in the process of formation test cases for each sprint.

Web Testing →

Let us know your details
so we can get back to you for discussion!

Telephone:
+1 805 491 9331
+44 1922 214429

Skype:
deviqa_co

Email:
info@deviqa.com

https://www.deviqa.com/

1. **Free proof of concept**
   We will conduct a free proof of concept and prove that you can trust our quality assurance services
2. **12 hours to start**
   Within 12 hours, our team is ready to start your project
3. **Quick team resize**
   The size of the QA team on your project could be resized in no time
4. **Daily progress reports**
   We send an email report with detailed statistics and progress on daily basis