



How GitLab iterates on SQL performance optimization workflow to reduce downtime risks



About GitLab

GitLab is a complete, open source DevOps platform, delivered as a single application. Organizations rely on GitLab's source code management, CI/CD, security, and more to deliver software rapidly.

More info: <https://gitlab.com>

Industry
Software

Headquarters
All-remote team

Employees
1300+¹

Challenge

GitLab is one of the fastest-growing private software companies (by revenue growth and relative number of employees). GitLab also reports that the number of their features has grown exponentially over the past years². GitLab adds value to the product at a predictable pace by making software releases every month. GitLab uses a monolith architecture on top of a 12 TiB PostgreSQL database executing 60k to 80k transactions per second. Such load and database size require a significant amount of time dedicated to the preparation of releases related to database changes to reduce the risk of downtime and performance degradation. Otherwise, the company could run into expensive consequences as service availability problems, reputation damage, release flow disruption, and incident response.

We incorporated Joe Bot into our development processes in mid-2019.

Joe Bot provides our developers with rapid feedback on query statistics and SQL suggestions when testing their queries against production-like data.

Joe Bot helps developers to identify and correct performance bottlenecks before they get to production.

Craig Gomes

Engineering Manager,
Memory and Database

1. Meet the GitLab team – <https://about.gitlab.com/company/team/>

 Database Lab (Postgres.ai)

In order to reduce the risks of production incidents, all database changes must be reviewed and verified in an environment similar to production prior to a production deployment. Data privacy, compliance, and security are very important for GitLab as a SaaS provider.

GitLab has comprehensive Security Requirements for Development and Deployment, including the least privileged database access limitations for more than 150 engineers involved in the development process. Such limitations are challenging for developers working on database changes. From more than 150 engineers there are approximately 10 engineers focused on the database (the ratio is common for the industry) which could create communication bottlenecks. Testing on staging or in development environments is also not sufficient. Differences in test and production environments may lead to bugs not being caught in time. Despite having solid best practices of CI/CD and fully automated staging database provision, replicating a production-like environment, load, and usage pattern is difficult as long as time and resource consuming. Which are common problems for the industry.

The primary goals are:

- to eliminate the risk of production incidents by providing independent yet close to production full-sized database copies to improve review and development process of database migrations while maintaining a high level of data security and low infrastructure costs;
- to remove the bottleneck from the development and review process to unblock backend engineers who have questions like "How will this query behave in production? How will it behave if we add this index?".

Solution

As a solution for the challenge described above, GitLab is working with Database Lab to iteratively build solutions to enable software teams to detect and avoid performance degradation and downtime incidents prior to deploying to production. Database Lab provides rapid, easy, and cost-effective on-demand database cloning of production databases of any size. As opposed to traditional ways of dealing with the data (like data sampling, fixtures, manual copying of the database) Database Lab is capable of provisioning multi-terabyte database clones identical to production literally in seconds. Using Database Lab clones, reliability of database changes could be drastically improved.

Using a copy-on-write filesystem allows provisioning database copies of any size in seconds as opposed to hours and days using traditional manual copying.

Database Lab Virtual DBA for SQL performance optimization (Joe Bot) was deployed in GitLab over a year ago. It allows the execution of any queries, including Data Definition Language (DDL) and Data Manipulation Language (DML) statements on production-like data to estimate query and index performance. For example, it's possible to see a full query execution plan, timings, amount of data that query will read and write (more details). Virtual DBA bot does not affect the production cluster, because it uses database clones matching production databases under the hood. While using the Virtual DBA bot, developers can get performance metrics but not the data itself. That's why it is easy to kick start the bot usage and related workflow process.

 Database Lab (Postgres.ai)

This was our first iteration in automating the performance feedback of PostgreSQL statements. Prior to this, developers either needed to ask database experts to help with verifying index optimization ideas, getting plans for long-running or modifying queries or they would manually export the SQL statements themselves and run in a tool such as <https://explain.depesz.com/>. This was a big (and growing) bottleneck in the development process of queries optimizations, which was successfully removed with Database Lab technology, the Virtual DBA chatbot, and by updating the database review guidelines to the development process. Database Lab drastically optimizes time in the development process and money spent on resources. Also, the bot helps developers to better understand SQL, the behavior of large PostgreSQL databases, and learn more quickly because it is easy to iterate when database clones can be provisioned in seconds.

GitLab is currently working on automating the migration testing process by utilizing the thin-cloning capabilities provided by Database Lab. Local testing isn't always enough to catch migration issues that only occur in production. By using the underlying thin-cloning technology, GitLab is working to provide an automated step to validate the migration steps securely against production-like data thereby catching these issues earlier in the software development lifecycle.

Key benefits



Provide developers the ability to provision dedicated environments for development and testing



The query performance optimization environment is in sync with the production data state



Reduce and eliminate the risk of performance degradation



Reduce time to provision database clones from days to seconds for database experts with access to the production



Developers can securely run queries of any type without the necessity of production DB access, not asking DB experts for help

Postgres.ai

Postgres.ai revolutionizes development and testing processes. Developers, DBAs, SREs and QA engineers get database-related work done much faster and with better quality thanks to thin clones and a high level of automation provided by Database Lab.

Database Lab Platform covers a broad range of applications related to developing and administration of databases, including data provisioning for dev/test and staging environments, SQL optimization, database changes verification in CI/CD, data recovery.