

Modernizing infrastructure effectively and seamlessly:

How to 'Git' to the root of the client's problem before resolving it

Industry

Manufacturing & Automation

Technology

AWS, GitLab

Services

Transformation & Modernization



Client profile

Partnering for infrastructure modernization

Our client is a Swedish conglomerate, offering a range of products and services related to locks, doors, gates, and entrance automation. This includes controlling access and confirming identities through keys, cards, tags, mobile, and biometric identity verification systems.

The client was focused on modernizing their IT infrastructure, tools, products, and services, and Nagarro partnered with them to achieve this goal. As the central R&D team, we provided tools such as GitLab, Atlassian, MIRO, Sonar, Nexus, and Black Duck to different divisions to facilitate and ensure smooth and seamless operations. This success story involves migrating GitLab, a popular version control and CI/CD tool, from AWS Instance to AWS EKS, to derive more benefits than we get by moving to Kubernetes cluster.

The challenge

Struggling with performance issues

The R&D Tools team provides GitLab to different divisions as a VCS and CI/CD solution. Initially, the Omnibus package based GitLab was hosted on an Amazon EC2 instance, aligning with the customer's needs at that time. However, the enrolment of more divisions has seen a significant increase in the number of users. Consequently, the higher usage of GitLab was causing performance issues. Code creation took more time and availability also became a problem. The client was keen to resolve these issues promptly and effectively.

The solution

Delivering top-notch service quality for a smooth launch

Nagarro's journey to successfully overcoming these challenges began with a collaborative effort with our client, where we first meticulously planned the migration strategy. Our dedication to delivering top-notch service quality led us to overcome all the challenges by taking a deliberate and phased approach.

We carefully crafted the migration approach, deciding to move away from Omnibus package-based installation and opting for HELM installation on Amazon EKS. Additionally, we employed S3 buckets for storing Job artifacts, LFS Objects, Uploads, Merge request diffs, packages, dependency proxy, page contents, project level secure files, and utilized PostgreSQL database to maintain primary metadata. The database was configured as an external instance on AWS, utilizing an m5.xLarge instance for hosting.

The true beauty of this solution lies in our adoption of Kubernetes instead of going for an omnibus-based installation. This guarantees high availability, allowing our infrastructure to scale as required, without confining ourselves to just one node. Since we're using Amazon EKS, we're not concerned with managing the control plane node; instead, we have three nodes serving traffic, with the ability to scale up to eight when required. Built-in features like HPA (Horizontal POD autoscaling) and VPA (Vertical POD autoscaling) can manage any large traffic hitting our system.

We used Terraform to manage all the infrastructure and built automated pipelines to set up the infrastructure with one click. We also conducted threat modeling and risk assessment exercises to ensure the system's security and robustness. Using Database Migration Services, we migrated from the old database to PostgreSQL. This took approximately 8 to 10 hours to migrate the data to S3 buckets and PostgreSQL.

To ensure comprehensive and centralized logging, we harnessed SumoLogic, seamlessly forwarding application logs from Cloud Logging using a Pub-Sub model. We used HELM to set up monitoring, relying on SumoLogic for both logging and monitoring solutions. PagerDuty enables us to send alerts to the support team in case of any issues with the infrastructure.

As we completed each step in the DEV environment, we encouraged teams to verify their workloads and integrations. We set aside ample time for testing and all the issues raised during the testing phase were effectively addressed.

The production migration was completed over a span of 2 days during the weekend and it was successfully launched to users on the next working day.

Impact to business

Proof of a successful migration

- The Amazon EKS solution (provided to the client) is highly available, scalable, and has minimal downtime, effectively addressing the challenges they had previously faced.
- 0 downtime: It has significantly improved the DevOps metrics, particularly MTTR (Mean Time to Recover), where no downtime has been observed in the infrastructure.
- With single-click infrastructure provisioning, we can create a new environment on-the-fly in no time.
- Upgrades, downgrades, and feature releases, which were scheduled to occur twice a month when GitLab introduced new releases, have been automated within their defined pipelines. This automation has significantly reduced the time required for these activities and allows them to be performed at any time without disrupting business operations.
- Allows GitLab as trusted entity where permissions will be managed via IAM roles.
- Integrated existing runners with strong build capabilities with the AWS EKS setup.



About Nagarro

Nagarro helps future-proof your business through a forward-thinking, fluidic, and CARING mindset. We excel at digital engineering and help our clients become human-centric, digital-first organizations, augmenting their ability to be responsive, efficient, intimate, creative, and sustainable. Today, we are 18,000 experts across 36 countries, forming a Nation of Nagarrians, ready to help our customers succeed.

For more information, visit www.nagarro.com.