



**Swiss Insurance Agency
Drives **Efficiency** and
Stability with Scalable
GitLab on EKS**



Introduction

In the fast-paced world of insurance, where customer satisfaction hinges on swift response and reliability, a Swiss Insurance Agency faced a critical challenge: their **existing GitLab infrastructure struggled to keep pace with the demands of their agile development workflows.**

Manual scaling of runners led to bottlenecks and delays, hampering collaboration and efficiency.

To unlock continuous delivery and adapt to fluctuating workloads, the agency embarked on a journey to **design a highly available and scalable GitLab implementation on Amazon Elastic Kubernetes Service (EKS).**

Problem

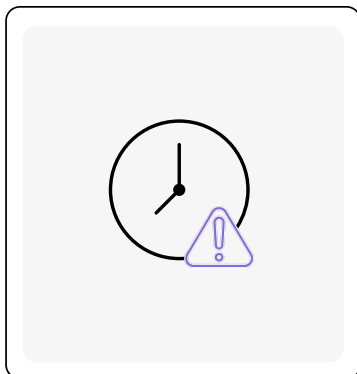


The agency's GitLab environment faced several limitations:



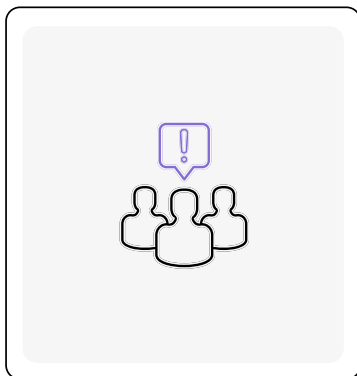
Limited Scalability:

Manual runner provisioning was **time-consuming** and **inflexible**, hindering the ability to handle fluctuations in development workload.



Downtime Risks:

The lack of redundancy and fault tolerance increased the **risk of outages**, disrupting the development process and impacting **customer service**.



Inefficient Collaboration:

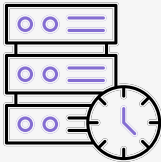
Bottlenecks caused by slow build times and **pipeline execution delays** hampered team **productivity** and collaboration.

To address these challenges, the agency sought a solution that would:



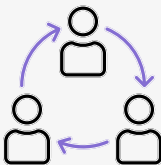
Automate runner scaling:

Dynamically adjust the number of runners based on demand, ensuring **efficient resource utilization** and optimal performance.



Enhance Availability:

Design a resilient architecture with redundancy and failover mechanisms to **minimize downtime** and maintain **service continuity**.



Boost Collaboration:

Streamline workflows and reduce build times to empower developers and **optimize team productivity**.

Solution



The agency implemented the following key solutions:



Automatic Runner Scaling:

- Leveraged the **Horizontal Pod Autoscaler (HPA)** to automatically scale runners based on CPU utilization, ensuring optimal resource allocation.
- Defined **custom metrics** and thresholds to **trigger scaling** based on specific workload requirements.



Resilient Architecture:

- Deployed GitLab on EKS using a **multi-AZ (Availability Zone)** configuration for redundancy and fault tolerance.
- Implemented **PostgreSQL HA cluster** with replication for database high availability.
- Utilized **Kubernetes** readiness and liveness probes to ensure **automated pod restarts** in case of failures.



Continuous Delivery Pipeline:

- Integrated **GitLab CI/CD** with automated testing and deployment processes to streamline workflows and **reduce manual intervention**.
- Optimized **build scripts** and container images to improve build times and **pipeline execution efficiency**.

Results



The implementation of this scalable GitLab infrastructure on EKS yielded significant benefits, including:

Improved Scalability:

Automatic runner scaling seamlessly adjusts to workload **fluctuations**, **eliminating bottlenecks and delays.**

Enhanced Availability:

High availability architecture **minimizes downtime** and ensures **uninterrupted service** even during potential disruptions.

Increased Productivity:

Streamlined workflows and faster build times empower developers and **boost team collaboration.**

Faster Time to Market:

Continuous **delivery pipeline** facilitates rapid deployment of new features and updates, **improving responsiveness** to customer needs.



Reduced Costs:

Efficient resource utilization **minimizes infrastructure costs** and optimizes cloud resource spending.

Conclusion

By embracing a scalable and highly available **GitLab infrastructure on EKS**, the Swiss Insurance Agency has transformed its development process. Automation, resilience, and efficient workflows empower their teams to deliver **exceptional customer service** and drive continuous innovation. This case study is a testament to the power of **cloud-native technologies** in unlocking agility and efficiency in today's competitive insurance landscape.

