Self-Hosted GitLab CI/CD Platform on AWS

Self-Hosted GitLab CI/CD Platform on AWS for Enterprise Production Use

This project focused on designing and deploying a self-hosted GitLab CI/CD platform in a production-ready AWS environment. The objective was to create a secure, scalable, and efficient DevOps infrastructure tailored for enterprise use.

The solution involved setting up GitLab Community Edition (CE) on an Amazon EC2 instance running Red Hat Enterprise Linux (RHEL). The instance was configured with optimized system settings, including 4+ GB RAM, 16+ GB storage, and custom firewall and SELinux policies to ensure production-level performance and security.

To manage CI/CD pipelines, a dedicated GitLab Runner was deployed on a separate EC2 instance. The runner was registered with the GitLab server and configured using the Docker executor, enabling containerized and isolated pipeline job execution. This setup improves job consistency and supports parallel builds.

For external access, the GitLab server was assigned a static Elastic IP, later mapped to a custom domain with HTTPS (SSL) enabled using Let's Encrypt. The GitLab instance was fine-tuned for a low-memory environment with additional swap memory and selective disabling of non-critical services like Prometheus.

Automated daily backups and monitoring mechanisms were integrated to ensure data safety and high availability. AWS services such as S3 were prepared for future offsite backup storage.

The GitLab Runner infrastructure was designed with scalability in mind, allowing horizontal scaling by launching additional runners as the team or workload grows. Runners were tagged and grouped for targeted pipeline execution, ensuring optimized resource use.

This production-ready setup enables a fully functional DevOps pipeline, improving deployment speed, code quality, and release reliability-while maintaining cost-efficiency and full control over infrastructure within AWS.