aembit

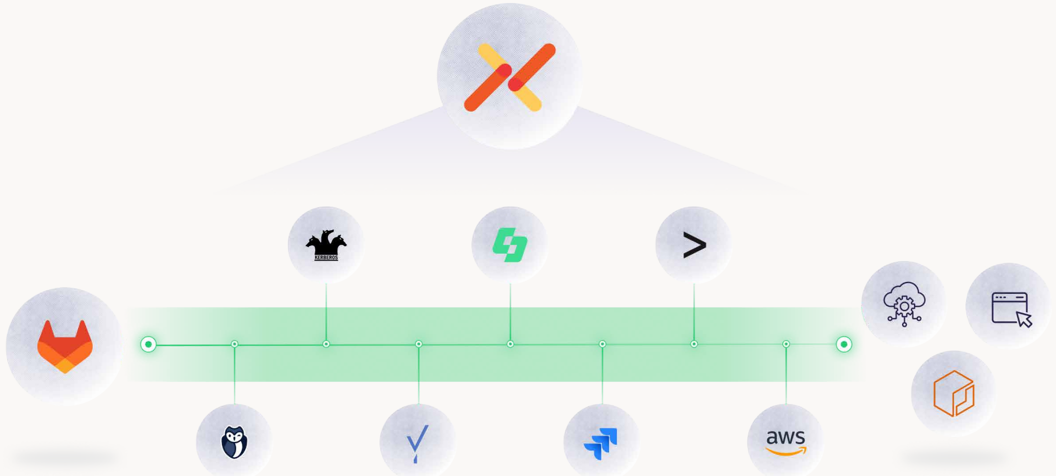
Enhancing Workload Access for *GitLab*

GitLab CI/CD pipelines are one of the primary methods of end-to-end software delivery. This is beyond pushing code to cloud, and includes testing and documentation. The automated process includes dozens of tools working together, which each require authentication and authorization. Securely managing and injecting credentials into these tools and pipelines remains a critical challenge. Hardcoded secrets, fragmented storage, and manual rotation expose organizations to significant security risks, audit complexities, and operational overhead.

The recent breach of Pearson and a repeat breach of the Internet Archive was due to exposed GitLab tokens and over privileged account.

Aembit's Workload IAM automates and enhances GitLab by providing comprehensive credential lifecycle management and secure, policy-driven credential injection.

We empower DevOps and developers to eliminate static secrets from their CI/CD pipelines, automate credential rotation and injection, enforce granular access policies, and gain centralized auditability, all without complex code changes.



Pain Points

GitLab provides built-in mechanisms for managing secrets, but these fall short for enterprise-grade security and scale.

1. **Static Secrets Proliferation:** Teams often resort to storing long-lived API keys and personal access tokens as variables, which are still static and vulnerable to leakage if the GitLab environment is compromised.
2. **Manual Credential Management:** Manual rotation of secrets is error-prone, time-consuming, and frequently neglected, leaving systems exposed.
3. **Lack of Granular Control:** Native GitLab variables offer limited granular access control, making it hard to enforce least privilege for individual jobs or stages.
4. **Fragmented Secret Storage:** Secrets are scattered across GitLab variables, developer machines, and various external vaults.
5. **Limited Auditability:** Tracing who (or what workload) accessed which secret, when, and why is challenging, hindering compliance and incident response.
6. **Complexity of Federation:** Implementing secure workload identity federation for cloud providers (AWS, Azure, GCP) or SaaS services directly within GitLab requires significant boilerplate code and expertise.

Request a demo at www.aembit.io

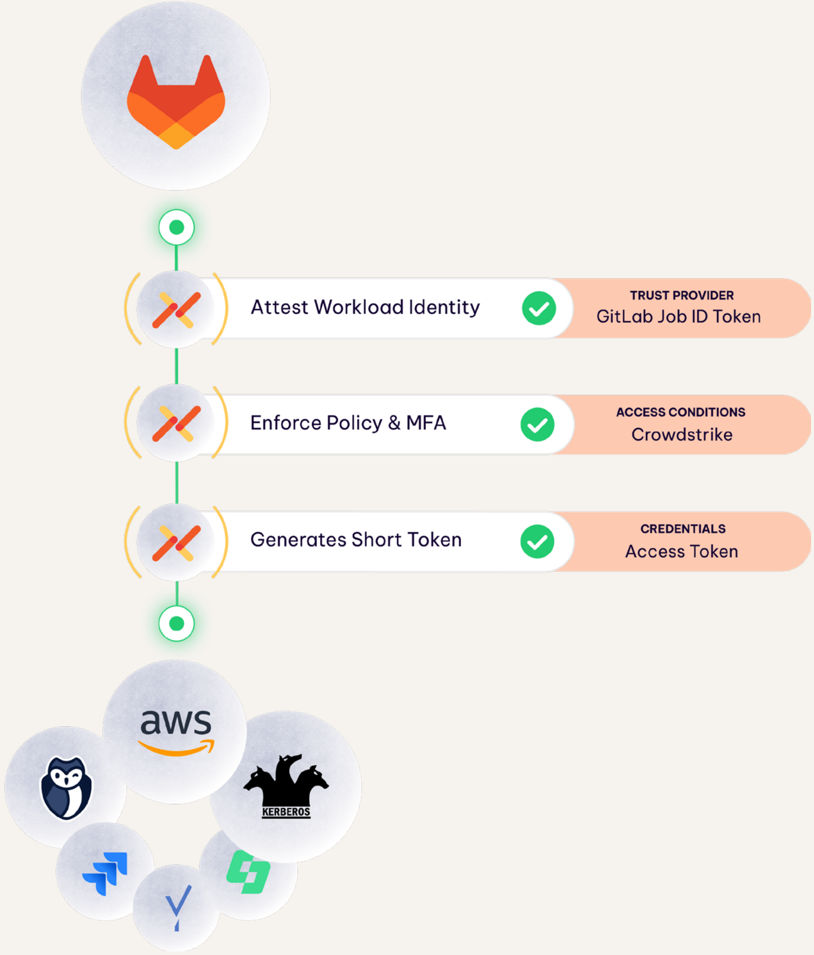
How *Aembit Helps*

If CI/CD is a "software assembly line," the more you can automate the better. Aembit automates aspects that DevOps, developers, and security teams are focused on, streamlining critical security processes within this assembly line.

By providing a comprehensive suite of capabilities, Aembit addresses the challenges of credential management in dynamic CI/CD environments, significantly enhancing security posture and operational efficiency.

Developers can focus on creating awesome applications and no longer need to design and code authentication and authorization. Others in the software delivery lifecycle, such as your technical writers, no longer need static, direct access to cloud storage and Content Delivery Networks (CDNs).

Your build and deployment pipelines can take care of distributing the new content securely.



Attest Workload Identity ✓ — TRUST PROVIDER: GitLab Job ID Token

Enforce Policy & MFA ✓ — ACCESS CONDITIONS: Crowdstrike

Generates Short Token ✓ — CREDENTIALS: Access Token

Aembit Benefits and Capabilities That Improve GitLab

Secure Credential Lifecycle Management

- Automated secrets seed rotation
- Just-in-time credential provisioning
- Centralized credential store

Identity-Driven Credential Injection

- GitLab job identity and verification
- Policy-based access control
- Zero hardcoded secrets
- Seamless credential injection

Enhanced Security and Compliance

- MFA for machines
- Eliminate secret sprawl
- Centralized audit trails