A Day in the Life of *a Developer...* Fighting *Secrets Sprawl* in CI/CD

A visual look at the identity and access challenges developers face in GitLab pipelines, the risks these issues create, and how teams can solve them.

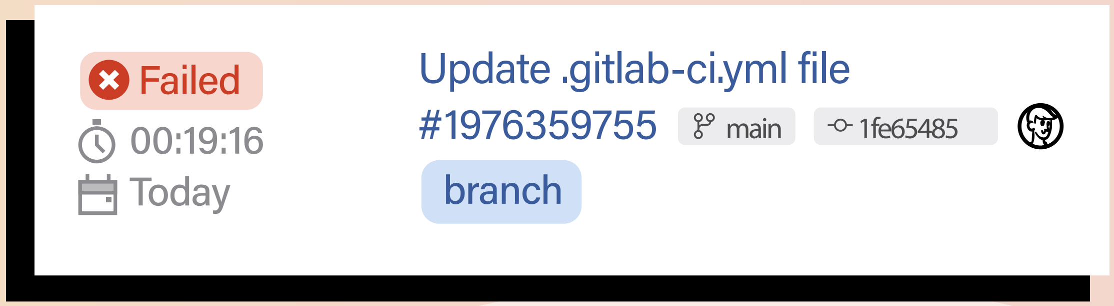
Meet Alex

Alex is a developer working in GitLab CI/CD.

aembit

Most days are spent building, testing, and deploying software. Today, the team is also dealing with credential issues.

Build Failure — 9:14 AM



Failed
00:19:16
Today

Update .gitlab-ci.yml file
#1976359755 main 1fe65485
branch

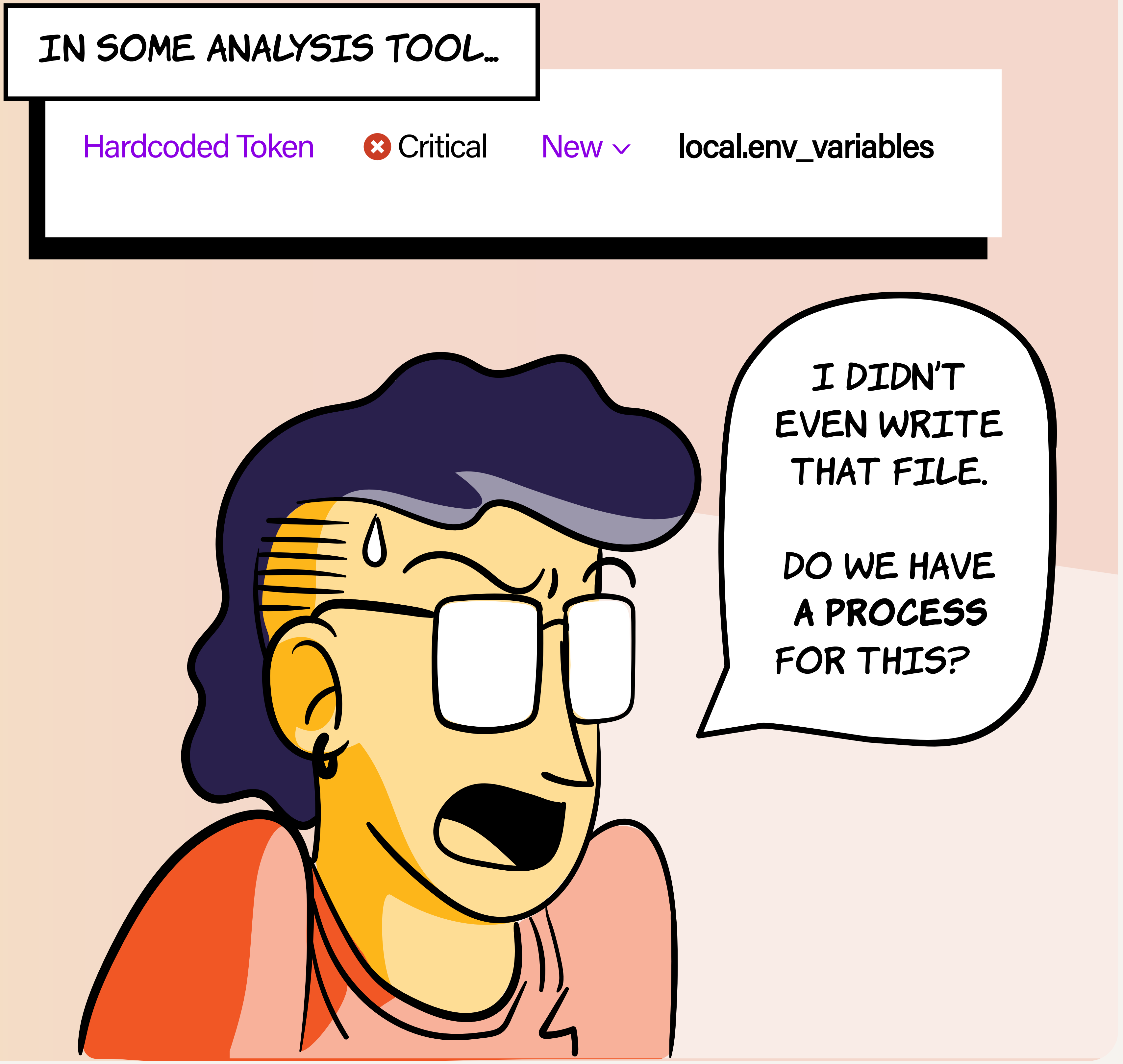
```
$ echo ${AWS_SECRET_ACCESS_KEY}
***********
```

Debugging – 11:20 AM

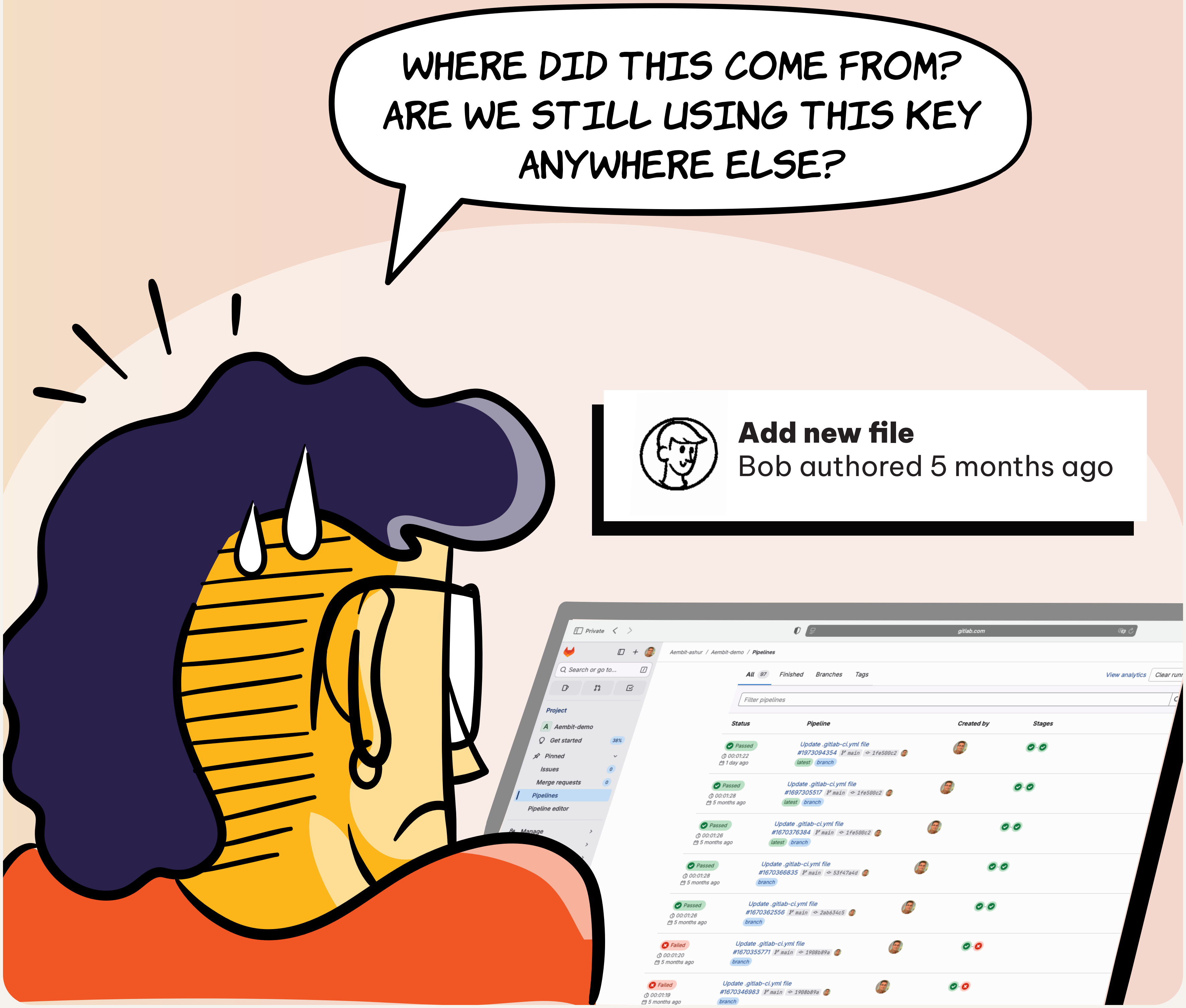
```
$ echo ${TOKEN}
ghp_abcdef1234567890
```

WHY IS THIS SHOWING IN THE LOGS AS PLAINTEXT?
I THOUGHT MASKED MEANT... MASKED?

Code Analysis — 1:00 PM

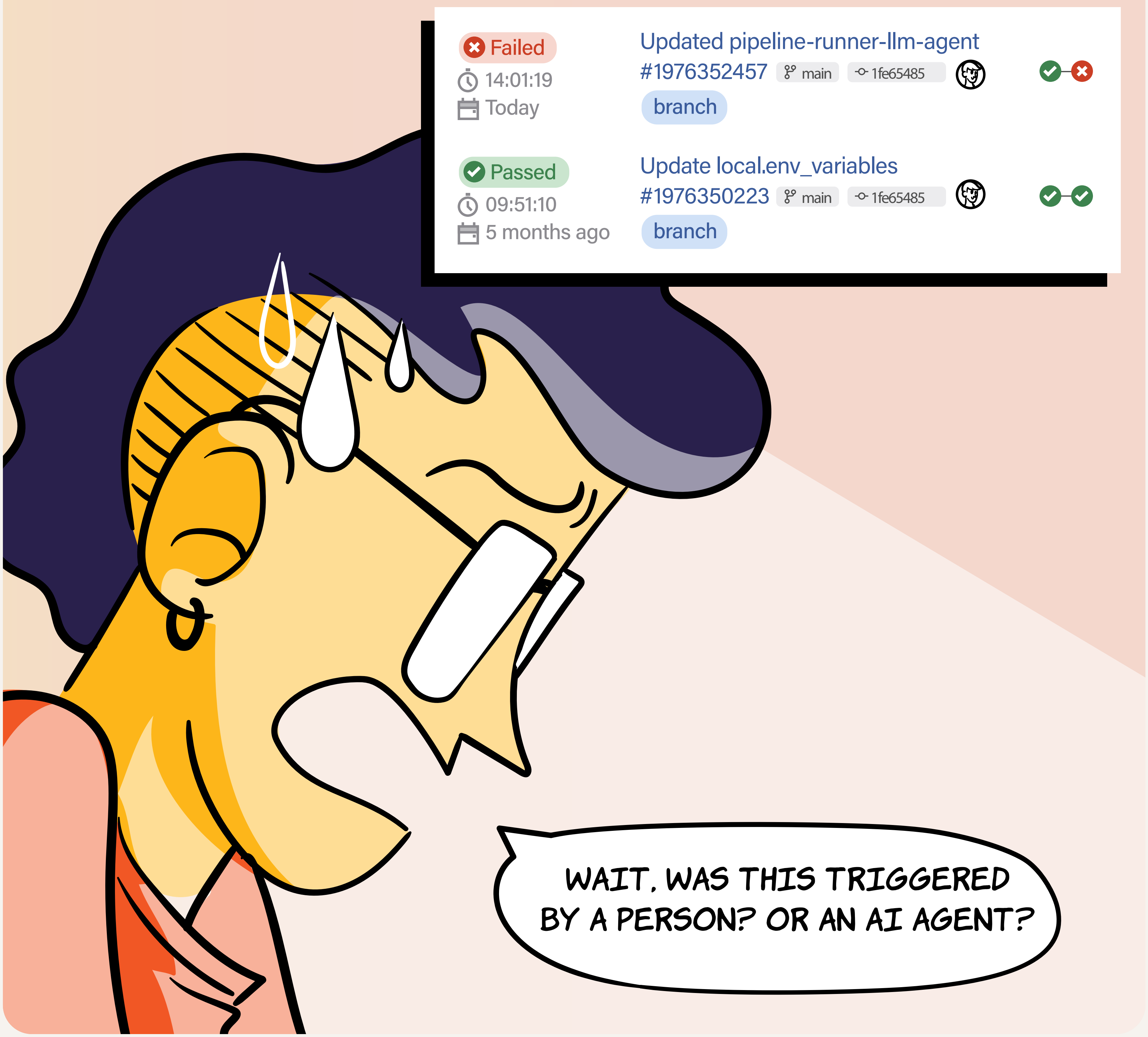


Investigation – 4:43 PM



Status	Pipeline	Created by	Stages
Passed 00:01:22 1 day ago	Update .gitlab-ci.yml file #1973094354 main 1fe500c2 latest branch		✓ ✓
Passed 00:01:28 5 months ago	Update .gitlab-ci.yml file #1697305517 main 1fe500c2 latest branch		✓ ✓
Passed 00:01:26 5 months ago	Update .gitlab-ci.yml file #1670376384 main 1fe500c2 latest branch		✓ ✓
Passed 00:01:28 5 months ago	Update .gitlab-ci.yml file #1670366835 main 53f47a4d branch		✓ ✓
Passed 00:01:26 5 months ago	Update .gitlab-ci.yml file #1670362556 main 2ab634c5 branch		✓ ✓
Failed 00:01:20 5 months ago	Update .gitlab-ci.yml file #1670355771 main 1908b89a branch		✓ ✗
Failed 00:01:19 5 months ago	Update .gitlab-ci.yml file #1670346983 main 1908b89a branch		✓ ✗

Audit – 7:39 PM



✗ Failed 14:01:19 Today	Updated pipeline-runner-llm-agent #1976352457 main 1fe65485 branch	✓–✗
✓ Passed 09:51:10 5 months ago	Update local.env_variables #1976350223 main 1fe65485 branch	✓–✓

How Secrets *Accumulate*

As Alex discovered there's no single place where secrets live. Instead, they show up in different forms across different layers of the GitLab environment. This is usually not done maliciously. Most of it is well-intentioned or done out of necessity. But over time, it adds up – and the risk compounds.

Where *Credentials* Typically Surface

Project-Level CI/CD Variables
Often used to store tokens or keys for build, test, or deploy steps (subtext options).

Group-Level Inherited Variables
Applied across many projects, sometimes without clear visibility or scope control.

.gitlab-ci.yml Files
Credentials occasionally hardcoded or indirectly referenced in scripts.

.env Files Committed to Source
Sensitive values included for local testing but accidentally pushed.

Shell Scripts Passed Between Teammates
Credentials embedded in helper scripts or bootstrap routines.

Tokens Shared in Slack or Pasted Into Issues
Quick workarounds that bypass audit and lifecycle controls.

Developer Machines
Locally stored API keys, often reused in CI/CD jobs or copied into repos.

CI/CD Job Logs
Secrets accidentally printed due to logging, misconfigured scripts, or lack of masking.

More Than an *Inconvenience*

There's broader context for what Alex is experiencing.

Private Repos Leak Secrets *8x* More Often

Internal (private) code repositories — including GitLab — leak secrets at a rate 8x higher than public ones, due to overconfidence in their security and lack of consistent oversight.

1 in 3 Private Repos Contains a Plaintext Secret

35% of private repositories on platforms like GitLab are found to contain at least one plaintext secret — including passwords, API keys, or tokens.

99% of GitLab API Keys Are Over-Permissioned

58% of keys have full access. The other 41% still have read-level access to sensitive data — violating least privilege.

Source: GitGuardian

Real-World Breaches Started With *GitLab Tokens*

In both the **Pearson** and **Internet Archive** incidents, improperly managed GitLab tokens were the initial access point for attackers.

"Threat actors compromised Pearson's developer environment in January 2025 through an exposed GitLab Personal Access Token (PAT) found in a public.git/config file."

Source: BleeingComputer

"...The initial breach of Internet Archive started with them finding an exposed GitLab configuration file on one of the organization's development servers, services-hls.dev.archive.org."

Source: BleeingComputer

What Developers Like Alex *Deserve* Instead

😤 What's Actually Annoying

- Hardcoding secrets just to get unblocked.
- CI/CD jobs silently failing because a token expired.
- Reusing one token across environments because rotation is a mess.
- Rolling back changes not because the code was bad, but because the secret broke.
- No clear way to know who or what used a secret — or when.

😌 What They Actually Want

- Credentials that show up when the job runs and vanish right after.
- Rotation handled by the platform, not by hand or cron.
- Policies that say whether this app gets access or not — and that's it.
- Logs that say this job accessed that resource at this time.
- Auth that's invisible, traceable, and boring, like it should be.

Dev time is *too valuable* for auth plumbing.

Let’s make it... *just work.*

Try Aembit *Free*

Because secrets don’t belong in your pipeline.

www.aembit.io

aembit