

Refactoring Code autonomously



Iain Breen
Senior Engineering Manager
Intercom



Live Demo from the console....



Intercom's code base is pretty decently sized

- 3 Main Apps
 - Intercom (Ruby on Rails) is ~14 Y/O, ~900k LoC, ~1.4m testing LoC
 - Embercom ([Ember.js](#)) is ~11 Y/O, ~300k LoC
 - AI (Python) is ~8 Y/O, ~460k LoC

Common dysfunctions across the repos

01

Dead Code - features that were spiked out, deprecated or shut down without being removed from the codebase

02

Stale Feature Flags - is it really a feature flag if every customer is opted in...?

03

Naturally evolved architecture patterns...

04

Engineer Onboarding cost is HIGH

05

It's hard for the AI to work with

What gives AI the best chance of success?

Code Organization

- ✂️ Break up large files
- ✂️ Organize related functions together
- ✂️ Maintain a clear Directory Structure

Naming

- ✂️ Use clear, meaningful and consistent naming
- ✂️ Avoid acronyms & abbreviations

Documentation

- ✂️ Ensure the codebase is documented heavily - including design decisions.
- ✂️ Use README or [claude.md](#) to provide context
- ✂️ A short description of what a file is meant to do at the top is helpful

Coding Style

- ✂️ Consistent formatting and style will allow agents to learn preferences
- ✂️ Use intuitive abstractions
- ✂️ Follow standard best practices for your lang / framework

**That's a lot of
work...**

Refactorer



AI Platform using Claude Code for
autonomous codebase improvements

What are we using it for?

Key Workflow Use Cases

Code Conversion

i.e converting Javascript to Typescript

Code Organization

Directory Structure updates - moving to a service-based structure

Documentation & Naming

File summarization, readme at service level

PR Review

High context PR Review

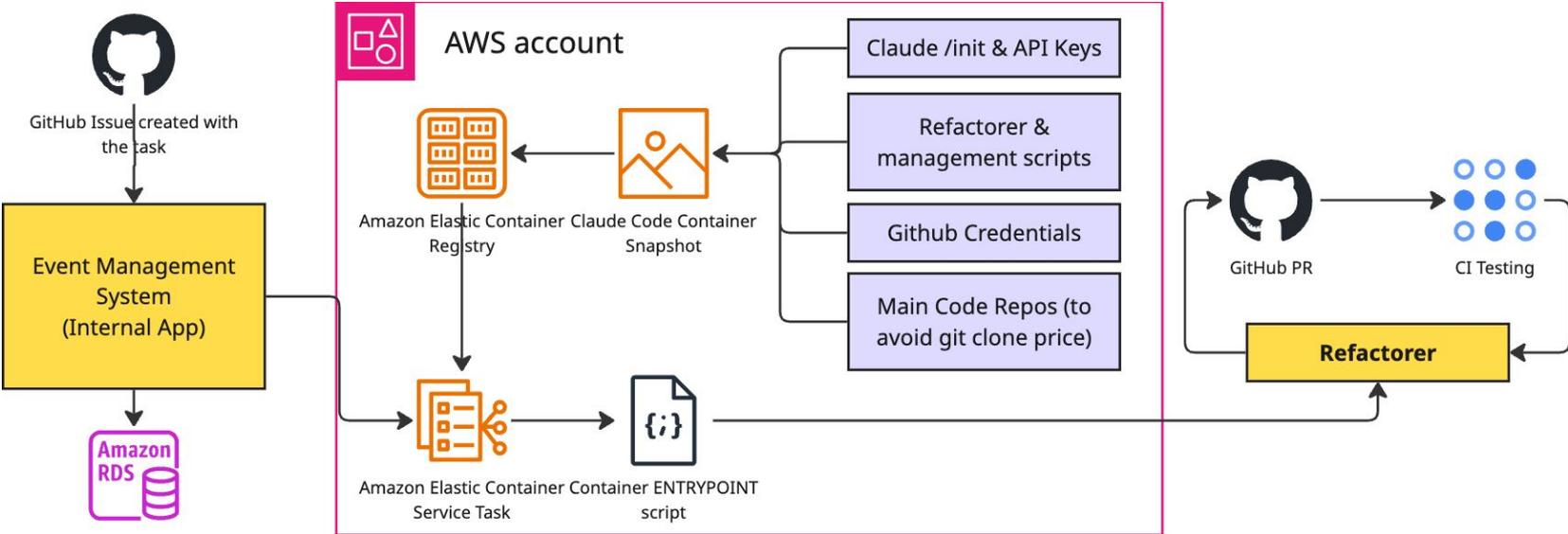
Feature Creation

Async AI platform for new work

How does it work?

Phase	Description
01 Task Definition	User creates a task description in a Github Issue
02 Ingestion	Parse task, analyse codebase, identify files; verify; create Github Issues
03 Execution	For each Issue, build instructions & invoke Claude Code; verify with OpenAI
04 Git Ops	Create feature branch, commit changes & push, create PR
05 Polishing	Run checks (lint, test) & fix automatically until all test pass (or timeout)
06 Review & Merge	Human reviews the generated PR & merges

Infra Architecture



The Task

Simply create a Github issue and tag it with a label



trousev opened yesterday



Allow to retry claude call if it failed with 500 error.

It became a routine when `claude` is failing in the middle by returning this:

```
{"type":"error","error":{"type":"api_error","message":"Internal server error"}}
```



Your goal is to implement restart-ability of claude using it's mechanism with `--resume` flag.

Do read documentation (<https://docs.anthropic.com/en/docs/claude-code/sdk>).

Your goal is:

1. When claude is being started, capture session ID
2. When claude fails, restart it with captured session ID

Do implement some retry policy: if there was more than 3 restarts -- ok, it's fine to fail.

Create sub-issue



The Initial Output

Automatic PR Creation

Add retry mechanism for Claude API calls #123

 Merged trowsev merged 9 commits into [master](#) from [ai-refactorer/add-claude-retry-mechanism-1747831227-3lpeydw](#)  18 ho

 Conversation 12  Commits 9  Checks 4  Files changed 2



trowsev commented 19 hours ago 

Closes: [#118](#)

This pull request introduces a retry mechanism for Claude API calls to improve the robustness and reliability of our application when interacting with the Claude AI service.

Why we are making these changes:

The Claude API occasionally experiences transient internal server errors (500 status code), which can disrupt the user experience and cause our application to fail unexpectedly. By implementing a retry mechanism with session resumption, we can gracefully handle these temporary failures and provide a more seamless experience for our users.

What was changed and how:

We've updated the `ai.py` file to include a retry mechanism for Claude API calls. The main changes include:

1. Capturing the session ID when starting a Claude session.
2. Implementing a retry loop with exponential backoff for API calls.
3. Using the captured session ID to resume sessions when retrying after a failure.
4. Adding a configurable maximum retry limit (default: 3 attempts).
5. Updating and adding tests to verify the new retry functionality.

These changes will significantly improve our application's resilience to temporary Claude API failures, ensuring a more stable and reliable user experience.



Refinement

Working through review



trowsev commented [19 hours ago](#)

[View reviewed changes](#)

refactorer/ai.py

Outdated

```
130 +         return None
131 +
132 +         # Look for session ID in verbose output
133 +         match = re.search(r'session ID: ([a-zA-Z0-9-]+)', output)
```



trowsev [19 hours ago](#)

Author ...

You should assume this is a standard JSON from claude as per docs: <https://docs.anthropic.com/en/docs/claude-code/sdk>

Read this docs and ensure that you're parsing this correctly. Don't make wild searches, just parse JSON properly.



Reply...

[Resolve conversation](#)

Into Production

Engineer merges



trowsev commented [18 hours ago](#)

Author ...

Summary

I've successfully implemented a retry mechanism for Claude when it exits with a non-zero code. Here's what I did:

1. Created a new helper function `_should_retry_claude()` that returns `True` for any `CalledProcessError`, not just internal server errors.
2. Modified the error handling in the `claude()` function to use this new helper function, which allows it to retry whenever Claude exits unexpectedly, including when killed with SIGKILL.
3. Updated the error messages to indicate when Claude exited unexpectedly rather than just for internal server errors.
4. Added appropriate test coverage for the new functionality.

The changes address the feedback from the comment, ensuring that Claude execution will automatically retry even if it's killed with SIGKILL or crashes for any other reason. This makes the entire system more robust when faced with unexpected failures.



[Fix linting issues flagged by flake8 refactorer](#)

✓ [5d32bd4](#)



trowsev merged commit [1214164](#) into [master](#) [18 hours ago](#)

4 checks passed

[View details](#)

[Revert](#)

Let's check back in...



Thank You

