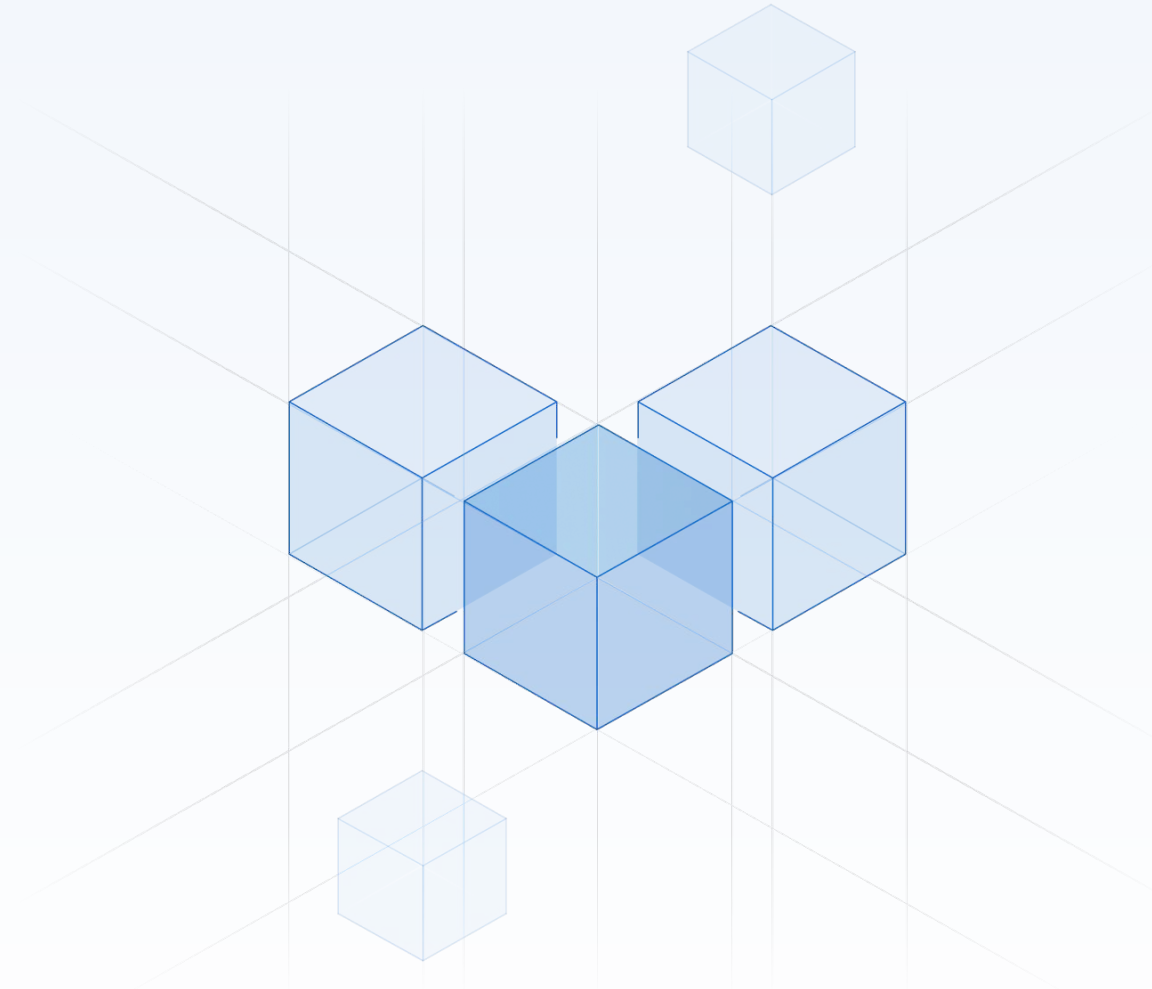
February 20, 2025

AssemblyAI's Success with Aspect Workflows

AssemblyAI provides advanced AI-driven speech-to-text services. They offer tools for audio transcription, sentiment analysis, topic detection, and other leading capabilities. They are known for high accuracy, scalability, and easy API integrations. AssemblyAI enables businesses to rapidly build voice-based applications. Needless to say, keeping their builds fast and their engineers moving is critical for providing world-class products to their customers.

Alex Eagle



Challenges

AssemblyAI's Bazel story began in early 2022, when they were using over 250 source repositories, many of which were branched from the history of others. The resulting forks diverged, preventing improvements or bugfixes from being applied consistently. New and old engineers alike struggled to find the correct source code. Releases often took over a week and rollbacks were commonplace. Even versions of libraries such as ffmpeg would vary between a researcher's machine where model training occurred and the production deployment, resulting in subtle transcoding bugs.

A monorepo existed at this time, but it was not consistent. Top-level folders had their own tooling choices. Dependencies and common approaches were not shared between them. AssemblyAI's team evaluated Bazel, Pants, and Buck2. The Head of Technology chose to start a revived monorepo effort with Bazel because of its high level of community support and his familiarity with the core team.

Solutions

AssemblyAI contacted Aspect in December 2022. They were seeking help with executing py_binary via gunicorn in a Python docker image. During a brief consulting engagement, Aspect quickly resolved AssemblyAI's issue and then migrated the company to a new monorepo. Backed by industry-standard SLAs, we supported AssemblyAI engineers in a shared Slack channel.

AssemblyAI became interested in our self-hosted Workflows product to speed up their Continuous Integration. They were focused on support for Github Actions and custom security groups along with optimizing AI model packaging, training, and inference workflows. AssemblyAI's trial began in April 2023.

The trial was evaluated on the following criteria:

1. Availability and scalability of Aspect's CI runners
2. High action cache hit rate and low analysis cache discard rate (based on provided Grafana dashboards)
3. Cost effectiveness
4. Selectively stamping and delivering only changed artifacts
5. Auditable back-references from artifact to the monorepo state where it was produced
6. Alerting on-call engineers to CI breakage via our "buildcop" service
7. Providing enough information to enable Continuous Deployment

Results

Deployments sped up with fewer rollbacks, new engineers onboarded faster, and AI researchers and engineers improved collaboration by working together in a shared repository.

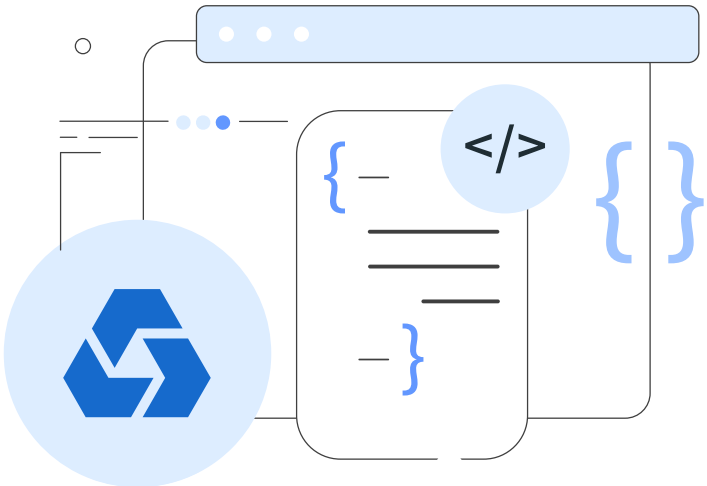
AssemblyAI saw significant improvements in build consistency. Optimized dependencies led to fast and cost-effective AI training and inference cycles. Developer CI wait time **decreased by 60%**. The cost to operate CI plummeted; the team estimates a **10x reduction**. During one week in particular, AssemblyAI's legacy **runners cost $600** while the self-hosted Aspect Workflows **runners cost $5**.

60% Wait time decreased	**10x** Reduced compute costs

Why It Worked

In a machine learning environment, it is critical for dependencies to be consistent between researchers conducting training and production systems performing inference. A Bazel-based Python monorepo with self-hosted Aspect Workflows ensures that AssemblyAI reliably manages dependency versions, leading to better performance and reduced overhead.

However, our proudest achievement was receiving an email from a former AssemblyAI engineer asking us to set up Workflows at his new startup!




Aspect
BUILD
AssemblyAI