J. R. Hill

—

hiljusti@so.dang.cool — 206.948.5692

▷ skills

A polyglot programmer interested in extreme force multiplication and enabling the "best tool for the job" whether it's new support for a cutting-edge build tool, or the ancient and forgotten tools of the past.

Large-Scale Software Maintenance ⇀ For the past couple years I've enjoyed building support for new languages and toolchains at Amazon, supporting a 20+ year old enterprise codebase that serves more than 500k software projects and 80k developers, and providing centralized incident response to critical CVEs.

Programming Languages ⇀ Experience with a variety of build and packaging tools, and more than 100 programming languages spanning many paradigms (Especially OO, Functional, and Logic programming). Created and maintained real-world applications in dozens of languages, mentored students in 28 programming languages, and authored educational material.

▷ goals

I'm currently driven by the concept of "done." I want to continue on a path of developing high-value software that stands on its own. I don't believe that the only path to job security is to consider all software projects as a forever-changing "product" that requires an unending backlog of feature requests and hefty support burdens over time. Instead I am learning to solve very large, hard problems (especially when the result is pointless human toil) and move to the next hard problem.

▷ charity and learning

Exercism ⇀ Contribute to exercism.org, a learning and mentoring platform for programming languages. Mentored students in 28 languages, and helped develop and launch tracks for Fortran and Zig.

Rail ⇀ Crafting Rail, a virtual machine implemented in Rust for concatenative programming languages. It currently has two experimental language frontends to its interpreted mode: The Joy-like *dt* and the Lispy *stap*.

Sigi ⇀ Developed Sigi, an organization tool for terminal lovers who hate organizing. It's a todo-list generator distributed through multiple Linux package managers.

▷ career highlights

amazon, code foundations, 2020–now

Community Leader ⇀ Created community groups within Amazon for various technologies like "Gradle Hackers," a collection of 132 engineers and managers from various business units across Amazon, AWS, and subsidiaries with a strong core of contributors and advisors. Created, consulted, or assisted with other communities for languages and software including Kotlin, Scala, Groovy, Flask, C/C++, Zig, .NET, Apache Maven, Android, Python, R, Dafny, Elixir, Erlang, Ruby, Haskell, and Crystal.

Improved Java Language Experience ⇀ Maintained the Java development experience for Amazon, AWS, and subsidiaries through large-scale software curation, community management, modernizing templates, improving build tool integrations, and patching foundational software used transitively by >500,000 software projects. I've directly contributed to more than 6,000 software projects.

More Tools in Amazon's Toolbox ⇀ In Amazon's older "Brazil" codebase, I codified templates for basic Rust, Go, Kotlin, and NodeJS usage patterns. I also built support for Zig and examples of cross-compiling for different Operating Systems and Architectures.

In Amazon's newer "Peru" codebase, I codified the Java experience, and built support for Zig, Apache Maven, Apache Ant/Ivy, CMake, SBT and Clojure. I also provided the earliest examples of Android and Lambda projects as well as the earliest examples for building C, Fortran, and X86 64-bit Assembly. (GNU Assembler and NASM)

"Builder Tools 102" ⇀ Began a cross-org effort (Learning tech, Tech writers, Builder Tools Product Management, etc) to supplement and extend newhire training for software engineers in areas where a 101 course was sorely lacking. This became a 102 course launched in 2023.

Centralized CVE Mitigation ⇀ I provided large-codebase-compatible patches to critical 3p software.

aws commerce platform, 2018–2020

Replication Validator ⇀ As part of an organization-wide project to build out the first new region of AWS's billing infrastructure, I designed and built a validation framework based on an AWS Kinesis and Step Functions architecture. The framework utilized Infrastructure As Code and allowed multiple teams to perform big-data parity checks before, during, and after a data migration to ensure that only targeted data was migrated, and to detect anomalies like write events during migration.

Cutover Service ⇀ As part of an organization-wide project to build out the first new region of AWS's billing infrastructure, I designed and built a low-latency cutover service and library to categorize live traffic and define valid routes across peer services in multiple regions.

aws support, 2014–2017

Operational Tools ⇀ TODO: TT Kiosk, Link Alchemizer – Still not replaced as of 2023!

Automated Refund Application ⇀ (As a Technical Customer Service Agent) Proposed and implemented automation via JavaScript to very manual web-based refund tool that saved AWS >$1mm in 2014 operating costs.