Ryan Parman

Cloud-native engineering leader, looking to pivot into project, product, and program management roles.

Links: GitHub (personal) • GitHub (side project) • LinkedIn • Stack Overflow • Role-targeted résumés
Format: Web • PDF • Word • OpenDocument • Raw Markdown

Summary

Ryan Parman is a seasoned problem-solver who excels at listening, adapting, and driving continuous improvement. Committed to delivering exceptional work, building impactful solutions, and elevating team performance.

Looking to pivot from a technical leadership role into a product/program management role, Ryan is seeking opportunities to leverage his technical acumen, leadership skills, and passion for shipping impactful projects successfully.

Work Experience

Older roles are truncated for brevity. If interested, details can be found on GitHub.

McGraw Hill — Remote (since COVID), previously Seattle, WA

Principal Engineer, Cloud Center of Excellence (January 2024—October 2024)

- Assumed a role influencing the technical direction of the entire organization. Ensured a focus on real-world, actionable feedback and provided strategic direction aligned with practical needs.
- Continued to be involved in the oversight and direction of our AWS stack, security, guardrails, and more.
- Identified opportunities to extend the security measures and guardrails developed for AWS to other cloud platforms.

Principal Cloud and Platform Engineer (June 2020—January 2024)

- Transitioned from Engineering Manager to a strategic technical leadership role.
- Prolific documentarian. Documentation is worth 50% of your grade.
- Partnered with McGraw Hill Enterprise Architecture and AWS Professional Services to deploy AWS Control Tower and AWS Identity Center. Lowered costs and increased control over account guardrails.
- Managed the program for building and maintaining base AMIs for all of McGraw Hill.
- Rebuilt our Artifactory cluster with a "cattle, not pets" approach. Ran the project from inception to completion, including the majority of development. Worked across dozens of teams and hundreds of services to complete the project.
- Proactively added support for lower-cost ARM64 CPUs, opening the door for ~$450k/year in cost savings.

Engineering Manager, Site Reliability (October 2018—June 2020)

- Managed a team of four, while working to level-up the team's technical skills and leadership capabilities. Conducted regular 1:1s, performance reviews, and career development discussions.
- Led the *Site Reliability Engineering* (SRE) team in addressing macro-oriented problems affecting decentralized, heterogeneous engineering teams across the company. Empowered greater self-service for engineering teams.
- Revamped the Seattle SRE interview process to prioritize a 70/30 focus on software engineering (Dev) and systems operations (Ops). Emphasized leadership qualities, bias for action, and high curiosity.
- Owned and served as the key decision-maker in development of a core platform for company-wide, reliability-focused projects.
- Formed and led a leadership group to establish a process maintaining reusable Terraform modules which could be composed together according to a service's needs.

Staff Software Engineer (October 2016—October 2018)

- Led the development of multiple Tier-1 services within the educational content authoring pipeline, leveraging technologies such as REST, GraphQL, API design, Amazon ECS, Docker, Terraform, ePubs, and security best practices.

- Provided the technical direction of these projects, promoted their adoption across the organization, provided comprehensive documentation, and offered ongoing guidance on adoption.
- Led the development of the authoring component of McGraw Hill's SmartBook 2.0 product, and the internal system which indexes authored content, builds ePubs, and encodes images/video for McGraw Hill's ePub CDN.
- Introduced the adoption of continuous integration (CI), continuous delivery (CD), rapid deployment practices, and Docker containers.
- Introduced a more hands-on monitoring approach, enabling development teams to actively engage in their own operations. Achieved significantly lower *Mean Time to Recovery* (MTTR).
- Served as a core resource in adopting *Infrastructure-as-Code* (IaC) tools such as Terraform and Packer.

WePay — Redwood City, CA

DevOps Engineer (April 2015—September 2016)

- Led a cross-company initiative to upgrade the monolithic application from PHP 5.4 to PHP 5.6 (the latest at the time). Facilitated cross-team collaboration among all major engineering teams and QA departments in order to achieve results.
- Initiated a program to automate the creation of base server images for our cloud servers. They allowed new servers to boot and begin serving traffic ~75% faster.
- Began investigating ways to implement *configuration-as-code* for our cloud infrastructure.

Senior API Engineer (April 2014—April 2015)

- Took the lead on the company's HackerOne program, coordinating across teams to address security issues.
- Built a development environment for engineering teams. Reduced new engineer onboarding time from 2 weeks → 1 day.
- Instrumental in designing WePay's MFA-as-a-Service offering. (U.S. patent filing US15042104)
- Developed new API endpoints to help expand WePay's business and support its partners.

Amazon Web Services — Seattle, WA

Web Development Engineer II (March 2010—April 2014)

- AWS hard-forked my open-source *CloudFusion* project into the AWS SDK for PHP, then hired me to work on it.
- Collaborated with the AWS Elastic Beanstalk team to provide PHP support for the platform, which launched in March 2012.
- Played a key role in the creation and development of the AWS SDK for PHP v2, incorporating significant changes in the PHP language and community since CloudFusion was first written in 2005.
- Collaborated with the AWS Design team on the AWS Management Console, to build a robust and user-friendly console. Led one of the first teams to provide reusable UI building blocks at AWS.
- Focusing on Amazon's *Customer Obsession* leadership principle, I successfully pushed for being better stewards of our community. Included increased transparency, better communication, and improved tooling for developers. [Examples]

Examples of Technical Documentation

Much of my other work is published inside of corporate Confluence/wikis. Here are some examples of my public-facing documentation:

- Setting up macOS for development
- Local AWS Lambda environments (with Go)
- Local development environment (devsec-tools)
- Configuring DataGrip for Valkey (devsec-tools)
- Diagrams of Artifactory infrastructure and software configuration.
- Diagram of a secrets-rotation system.

Recommendations

See a selective list of recommendations from co-workers and peers.

Groups and Accomplishments

- U.S. patent filing, "System and Methods for User Authentication across Multiple Domains" (US15042104) (2016)
- U.S. patent filing, "Hive-based Peer-to-Peer Network" (US8103870B2) (2007)
- Voting representative for AWS, PHP Framework Interoperability Group (2012–2013)

Keywords and Skills

This list is not exhaustive, but is targeted toward the skills most relevant to PM, TPM, and Product roles.

Confluence, Jira, building platforms, coordination with downstream services, cross-collaboration (dozens of teams, hundreds of services), organization of complex projects, product development, product roadmap management, project documentation, project management, risk mitigation, stakeholder management, technical documentation, vendor management.

Education

Obtained a **Bachelor of Arts** degree in *Design and Visualization* from *Silicon Valley College* (now Carrington College) in San Jose, CA.

Graduated in *November 2003* with a **3.84** GPA.