Ryan Parman

Cloud-native engineering leader with a focus on reliability, scalability, and security for the modern web.

Most-Recently: Enterprise Architect, Cloud Center of Excellence at McGraw Hill.

GitHub (personal) • GitHub (side project) • LinkedIn • Stack Overflow • Web résumé • PDF résumé • Word résumé

Summary

Ryan Parman is a cloud-native engineering leader with over 25 years of experience, who specializes in technical leadership, software development, site reliability engineering, and cybersecurity for the modern web. A seasoned problem-solver who excels at listening, learning, adapting, and driving continuous improvement. Committed to delivering exceptional work, building impactful solutions, and elevating team performance. Thrives in environments which empower innovation and becoming a force-multiplier.

Key accomplishments include:

- Founding member of the AWS SDK team.
- Patented multi-factor authentication as a service at WePay.
- Instrumental in defining CI, CD, and SRE disciplines at McGraw Hill.
- Conceived the idea of serverless, event-driven, responsive functions in the cloud at Amazon Web Services in 2010 (AWS Lambda).
- Contributed significantly to numerous other high-impact projects.

Technical Skills and Software

While my experience and personal technical interests are broad, the following list is focused more on my interest in DevTools, DevOps, and SRE roles. I would be happy to share additional experience for other areas upon request.

> 📓 **NOTE**
>
> Each skill listed includes a current proficiency level — Low, Medium, High, or Expert — along with a directional arrow indicating proficiency trends. An upward arrow (↑) signifies that I am actively working with the skill, and my proficiency is likely to increase over time. A downward arrow (↓) indicates that I have not utilized the skill recently, and my proficiency may decrease unless refreshed.

- **Operating Systems:** macOS (Expert: ↑), CentOS Linux (High: ↓), Amazon Linux 2 (High: ↓), Amazon Linux 2023 (High: ↑), Alpine Linux (High: ↑), Windows (Med), Ubuntu Linux (Med: ↑).
- **Standard Software Engineering Toolbox:** Dependency injection, performance profiling, character encodings, Git, Linux, Makefiles, and other fundamentals (High: ↑); memorized algorithms (Low); memorized Big-O notation (Low; I never learned it formally).
- **Programming Languages:** Golang (High: ↑), Python (High: ↑), Bash (High: ↑), *Modern* PHP (High: ↓) (not the bad old PHP that everyone hates), Browser JavaScript (High: ↓), Node.js JavaScript (Medium: ↓), Ruby (Low: ↓). Starting to learn Swift, but am just scratching the surface.
- **Cloud Computing:** AWS (Organizations, EC2, RDS, S3, CloudFront, SQS, SNS, IAM, STS, CloudWatch Monitoring, CloudWatch Logs + Insights, Lambda, ECS-on-EC2, ECR, API Gateway, Auto-scaling, CloudTrail, Elastic Transcoder, ElastiCache, Route 53, ELB/ALB, ACM, SSM, Parameter Store) (mostly High/Expert: ↑), AWS SDKs + CLI (High: ↑), AWS Well-Architected Framework (High: ↑); Google Cloud's core infrastructure services (Low: ↓), Microsoft Azure (None: ↑)
- **Provisioning:** Terraform/OpenTofu (Expert: ↑), Terragrunt (Med/High: ↑), Packer (High), Ansible (Med: ↓), Vagrant (Med: ↓), writing custom Modules (Expert: ↑), writing custom providers with the Plugin Framework (Med: ↑).
- **API and Scalable System Design:** Understanding and designing highly-scalable, distributed systems for running web applications and web services (High: ↑); JSON-over-HTTP web service API design (High); GraphQL with Relay implementations (Med/High: ↑); Understand the difference between micro-service vs a "distributed monolith" (High: ↑); OpenAPI (née Swagger) (Med); JSON Schema (High: ↑); gRPC (Low: ↑); 12-factor design (High: ↑); Ent (Med: ↑).
- **Containers and Orchestration:** Docker (High: ↑), Amazon ECS (High: ↓); Kubernetes (Low: ↑).
- **Enterprise Services:** Artifactory (Expert: ↑), Jira (High: ↑), Confluence (High: ↑), GitHub Enterprise (High: ↑), GitHub (High: ↑), Pingdom (Med: ↓), New Relic (Med: ↑), Datadog (Med: ↓), Papertrail (Med: ↓), Slack (High: ↑), PagerDuty (High: ↑).
- **Databases & Key-Value/Document stores:** MySQL (Med: ↑), Redis (High: ↓), PostgreSQL (Med: ↑), Memcache (Low: ↓), Atlas (Low: ↑).

- **Metrics, Traces, and Logs:** OpenTelemetry (Med: ↑), New Relic (Med: ↑), Datadog (Med: ↓), Jaeger (Low: ↑).
- **Metadata and Config Formats:** RDFa, Dublin Core, FOAF, OpenSearch, JSON-LD, Microformats, RSS, Atom (RFC 4287), JSON, YAML, TOML, XML, HCL, Schema.org, Open Graph.

Work Experience & Notable Projects

Northwood Labs — Side Project

Owner (January 2024—Present)

- Northwood Labs is an incubator for some open-source and passive-income projects.
- **Custom Linux Packages:** Currently building a repository of custom Linux packages for the matrix of Alpine Linux, Amazon Linux, and Ubuntu Linux across `x86_64` and `arm64` architectures, that can be added as a standard repository definition and installed via the system's package manager.
- **CSP Evaluator:** Currently building a parser and evaluator for Content Security Policy (CSP) directives in Go.
- **Terraform Provider:** Built a Go library of functionality which (I believe) should have been built-in to Terraform/OpenTofu, then wrapped it in a custom Provider. Having this functionality exposed as Go code, Data Sources, and Provider-defined Functions, makes it easier to write integration tests with Terratest, ensuring with identical implementations.
- **Multi-Platform Docker:** Built a downloader for GitHub release assets which helps streamline the ability for a Dockerfile to pull the correct build of a release for the right OS/Arch. Simplifies building multi-platform images.
- **AWS Organization Security:** AWS has a pattern for multi-account organizations which they call "hub and spoke." Built a library + CLI tool which simplifies the process of traversing the hub-and-spoke in order to execute a command on the spoke account using the same methodology an any automated process would.
- **AWS Session Manager:** The terminal is the right tool for shell sessions. Built a TUI for simplifying the ability to connect to an SSM-enabled EC2 instance using your Terminal.
- **Configuration for `tflint`:** Built a tool for generating up-to-date configurations for AWS/GCP/Azure for use with tflint. The `tflint` software helps identify security and best-practice errors in Terraform/OpenTofu code.

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

Enterprise Architect, Cloud Center of Excellence (January 2024—October 2024)

- Assumed a role influencing the technical direction of the entire organization.
- Prevented "Ivory Tower Syndrome" by collaborating closely with members of the Cloud Center of Excellence, Reliability Engineering, Cybersecurity, Networking, and Application Development teams.
- 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.
- Collaborated with peers focusing on Microsoft Azure and Oracle Cloud Infrastructure.
- Deepened understanding of our cloud fabric to enable high-performance networking across multiple clouds.
- Identified opportunities to extend the security measures and guardrails developed for AWS to other cloud platforms.
- Trained peers on the effective use of Terraform for cloud management.

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

- Transitioned from Engineering Manager to a strategic technical leadership role, focusing on projects I'd initiated. By stepping away from direct personnel management, I was able to concentrate on providing technical leadership.
- **Documentation:** Prolific documentarian. Documentation is worth 50% of your grade.
- **Reliability Platform:** Either directly or collaboratively designed and maintained AWS Control Tower, Artifactory, GitHub Enterprise, GitHub Actions, Circle CI Enterprise, Jenkins, and more.

- **AWS Control Tower:** 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. Enabled automated provisioning of new accounts, and developed smoke tests as a post-provisioning validation step.
- **Base AMI program:** Leveraged insights from Packer, CIS Benchmarks, security patching, and the specific needs of internal AMI customers to develop a unified build pipeline integrating best practices. Achieved zero downtime across the organization by automating the seamless rotation of hosts to use the new AMIs. Automated rollbacks for failed nodes.
- **Streamlining:** Integrated Terraform, Monitoring-as-Code, Base AMIs, and custom security tooling to empower application teams. Enabled deployment of Docker images with minimal configuration to Amazon ECS clusters, incorporating best practices, infrastructure monitoring, and operational tooling. Reduced overall costs by streamlining deployment processes.
- **Preventative automation:** Conducted comprehensive scans of Route 53 and other DNS providers to obtain a mapping of thousands of active websites owned by McGraw Hill. Enabled faster rotation of expiring datacenter certificates by identifying both the certificates and their installation locations. Verified required DNS records for self-rotating *Amazon Certificate Manager* certificates.
- **Self-hosted GitHub Actions runners:** Implemented Amazon EKS to deploy self-hosted runners for GitHub Actions within our GitHub Enterprise environment. Developed hourly smoke tests to validate the GitHub Actions runner environment and the imported actions for internal developers. Enhanced visibility and provided working examples to effectively leverage actions, improving overall developer efficiency.
- **Automation for Artifactory:** Rebuilt our Artifactory cluster with a “cattle, not pets” approach. Migrated artifacts from NFS to S3, which significantly lowered costs. Rewrote configurations using Terraform to eliminate manual configurations. Moved service-user management into Terraform. This automation reduced human error, improved security posture, and increased consistency, leading to a better developer experience.
- **Token Vending Machine:** Enable continuous token and password rotation for our engineering teams, providing a "push-button, receive-token" solution. Leveraged AWS Secrets Manager, Lambda, KMS, IAM policies, and custom CLI software written in Go. Implemented the initial integration for service accounts in Artifactory.
- **ARM64 Adoption:** Proactively addressed dependencies on Intel x86_64 by updating build pipelines for ARM64 compatibility. Implemented ARM64 runners for GitHub Actions and established ARM64 parity in Artifactory for remote repositories. Authored tutorials and hands-on documentation for utilizing Docker BuildKit to produce multi-platform container images. Prepared the organization for the adoption of AWS Graviton (ARM64) CPUs for cost optimization, ensuring all components were in place for seamless integration.

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

- Owned and served as the key decision-maker in development of a core platform for company-wide, reliability-focused projects.
- As development teams transitioned to Full-Cycle Development, led the Site Reliability Engineering (SRE) team in addressing macro-oriented problems affecting over 75 decentralized, heterogeneous engineering teams across the company. These initiatives empowered greater self-service for engineering teams, enabling them to move faster without reinventing the wheel.
- **Managing humans:** Managed a team of four — one full-time and three contractors. Unfortunately, the team was spread thinly, and I *also* had to pick-up several hands-on engineering tasks to keep up with our workload. Ultimately, I was pulled back into engineering in my subsequent role because there was a greater need for me there.
- **ECS-optimized Amazon Linux Base AMI:** Customized the AWS-provided AMIs to comply with Level-2 CIS Guidelines for both Amazon Linux and Docker. Collaborated closely with cybersecurity, operations, and various business units to ensure compliance. Achieved high levels of opt-in adoption, enhancing confidence among cybersecurity and operations organizations in the product development teams.
- **Prism:** Developed custom security and operational tooling where off-the-shelf tools wouldn't give us what we needed, to understand the current posture of ±200 AWS accounts. Made the data transparent to ALL engineers, enabling teams to be involved in improving their infrastructure stacks.
- **Monitoring-as-Code:** Leveraged Terraform and Go to streamline the generation and ongoing maintenance of dashboards and monitors in Datadog and New Relic across a large, heterogeneous range of applications.
- Formed and led a leadership group to establish a rigorous process for developing, patching, distributing, and maintaining reusable **Terraform modules** utilized by numerous product development teams across the company.
- Assumed engineering management responsibilities for the Site Reliability Engineering (SRE) group in McGraw Hill’s Seattle office. Led initiatives to better integrate our office and practices with the expanding SRE practice across all U.S. offices. Joined the SRE

leadership group to guide and participate in developing improved reliability processes, collaborating with product development teams to adopt and implement these practices.

- Revamped the Seattle SRE **interview process** to prioritize the recruitment of high-quality engineers with a 70/30 focus on software engineering (Dev) and systems operations (Ops), emphasizing strong leadership qualities. Integrated numerous ideas and leadership principles from experience at AWS to enhance the recruitment strategy.
- Implemented a more collaborative SRE-style approach by closely integrating with development teams, effectively minimizing the practice of siloed hand-offs to operations teams. This initiative enhanced cooperation and efficiency within the areas supported by the Seattle SRE team.

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.
- Lead 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.
- Initiated the adoption of continuous integration (CI), continuous delivery (CD), rapid deployment practices, and Docker containers. Championed "dogfooding" of new processes, resulting in deployments that were both more frequent and more reliable.
- Introduced a more hands-on monitoring approach, enabling development teams to actively engage in their own operations rather than relying solely on third-party vendors used by other groups in the company. Achieved significantly lower Mean Time to Recovery (MTTR) during incidents by implementing application-level metrics tracking and introducing Key Performance Indicators (KPIs).
- Served as a core team member in migrating all new infrastructure to Infrastructure-as-Code (IaC) tools such as Terraform and Packer. Identified patterns across applications and initiated efforts to streamline infrastructure maintenance using shared, reusable Terraform modules.

Perimeter of Wisdom, LLC

Co-Owner, CTO, Producer (February 2015—2018)

- Developed the entire website for "The First-Time Offender's Guide to Freedom," managing all technical aspects from inception to deployment. Also performed all production work on the eBook authored by E. M. Baird.
- Utilized then-modern front-end technologies — including Bootstrap, LESS, JavaScript, Gulp.js, npm, and Bower — to build the website's front end.
- Developed the back end using PHP 5.6 with HHVM and Nginx, integrating MySQL, Redis, Slim Framework, Monolog, Pimple, Twig, Guzzle, Doctrine, Phinx, and Symfony components.
- Deployed the application using Ansible and developed within a Vagrant environment running Ubuntu.
- Conducted unit, integration, and functional testing using PHPUnit, Behat, Mink, and Selenium. Leveraged Amazon SES for email delivery, Amazon S3 for static file storage, Stripe for payment processing, Linode for web hosting, and MaxMind for IP-based geolocation. Integrated Google Books and Dropbox to ensure customers always had access to the latest errata fixes.

WePay — Redwood City, CA

DevOps Engineer (April 2015—September 2016)

- Enhanced WePay's cloud infrastructure provisioning by optimizing update deployment processes and managing security patches. Improved application and infrastructure monitoring.
- Streamlined the planning, development, deployment, and maintenance of new microservices throughout the company.
- Led a cross-company initiative to upgrade the monolithic application's software stack from PHP 5.4 to PHP 5.6. Facilitated cross-team collaboration among all major engineering teams and QA departments.
- Managed the replacement of over 200 servers across multiple environments, achieving zero customer-facing downtime.
- Maintainer of multiple tier-1 systems including Artifactory, GitHub Enterprise, and Phabricator.

Senior API Engineer (April 2014—April 2015)

- Developed new API endpoints to help expand WePay's business and support its partners.
- Was instrumental in designing/developing WePay's MFA-as-a-Service offering. (U.S. Patent filing US15042104 "System and Methods for User Authentication across Multiple Domains".)
- Enhanced the security of WePay's products by coordinating fixes with cross-functional teams while managing competing priorities. Personally resolved numerous issues to ensure product integrity and protect customer data.

> **NOTE**
>
> Experience greater than 10 years ago is available upon request: Amazon Web Services, Rearden Commerce (Deem), Yahoo!, WarpShare, Stryker, Digital Impact (Acxiom Digital)

Recommendations

A full list of recommendations can be found on my LinkedIn profile.

Groups & Accomplishments

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

Education

Silicon Valley College (now Carrington College) — San Jose, CA

Bachelor of Arts, Design and Visualization (November 2003)

- GPA: 3.84