Rajath Rajendra Pai

Platform & DevOps Engineer | AWS, CI, automation

Gothenburg, Sweden · rajathpai2000@gmail.com

github.com/rajathpi · rajathpi.github.io · linkedin.com/in/rajath-pai · kaggle.com/rajathrpai



PROFILE

I build and run the systems other engineers rely on, and I like owning them end to end: design, launch, measure, and fix them when they break. At Coretura, an automotive software joint venture, I built and run the service that keeps work tracking in step across six company GitHub organizations, and I work on build speed and pre-release testing for a pilot customer. Before that I was a data scientist at Volvo Group, running HR approval and rating systems. M.Sc. in Data Science and AI from Chalmers.

10,500+	**0 errors**	**26×**	**48 → 24**
tickets kept in step across 6 orgs	in ~70,000 sync runs in 30 days	faster first sync, 23.5 s to 0.89 s	minutes per CI image rebuild

TECHNICAL SKILLS

Cloud & CI	AWS (Lambda, API Gateway, DynamoDB, S3, ECR, IAM, CloudWatch), Terraform, Docker and BuildKit, GitHub Actions, GitHub GraphQL API, Copybara
Automation	Microsoft Graph, Power Apps, Power Automate, SharePoint, Jira, Exalate (Groovy)
Data & ML	pandas, PySpark, Databricks, Power BI, PyTorch, Hugging Face, LoRA
Languages	Python, SQL, Bash, C++

EXPERIENCE

Coretura — Gothenburg, Sweden

DevOps Engineer, Developer Experience — Mar 2026 – Present

- **Cross-company backlog sync.** Designed, built and run a service that keeps **9 project boards in 6 GitHub organizations** in step both ways: status, sprint, dates and every other field, which GitHub cannot do on its own. When a ticket changes, GitHub calls a small Python function on AWS (API Gateway, Lambda, DynamoDB) that updates every other board holding that ticket. Measured in Oct 2026: **10,500+ tickets**, about **70,000 runs in 30 days**, 0 errors.
- **Two production incidents.** Updates were silently lost once a board passed 5,000 tickets, the most one board search would return. Caching, retries and newest-first reads fixed it and cut the first sync of a new ticket from **23.5 s to 0.89 s**. Later, a company-wide planning day used up GitHub's hourly API allowance, because the service searched whole boards only to learn a ticket was *not* on them. Storing those answers ahead of time cut that cost from **352,440 to 477 API points**, and failed updates now wait in a list and are retried.
- **Careful changes.** Each change is backed up, then rolled out to one ticket, a few, then all; 56 unit tests. Now moving the service into Terraform (infrastructure as code) and the production AWS account, with a queue between receiving and processing events.
- **Pre-release testing.** Made pre-release (canary) tests for a pilot customer run on the customer's real codebase with the change merged in, instead of on the change alone, with one test branch per change so several can run at once. Fixed a check that showed passing runs as failed. Rehearsing a proposed pull-request gate, I found it copied the whole branch instead of the change: one rehearsal run put back **14,016 files** that had been deleted.
- **Faster builds.** With a colleague, moved the CI Docker build cache to S3 (managed in Terraform, one cache per branch, 14-day expiry). Rebuilds of the developer image went from about **48 to 24 minutes**, and CI machines that are used once and thrown away became practical.
- **Checking the code scanner.** Tested the static-analysis pipeline (Coverity, which scans C code for bugs before release) in three partner environments by planting **8 known defects**, one per throwaway branch. It caught 4 of 8. I traced the misses to the set of checks that was switched on and wrote down the setting that would widen it.
- **Partner onboarding.** Wrote a Python tool on Microsoft Graph that sends guest invitations in bulk from an access spreadsheet, with a dry run first and a log of every run. Used to give **80+ partner engineers** access.
- **Jira sync between companies.** Set up two-way comment sync between two companies' Jira projects (Exalate, with Groovy scripts): **113 existing comments** backfilled, deletions carried both ways. Added attachment sync; a full re-sync afterwards created no duplicates.
- **Customer issue reporting.** Audited the customer issue-reporting system end to end across four repositories, found why customer tickets were not reaching triage, and turned the findings into the rollout plan I now lead.

Volvo Group — Gothenburg, Sweden

Associate Data Scientist — Mar 2025 – Feb 2026

- **Position approvals.** Maintained and documented the system managers use to open new job positions (manager, VP, SVP, then recruiting), built on Power Apps, SharePoint and Power Automate. When the company reorganized, moved it to the new organization while approvals were still running, with **every pending position carried across**, checked against a colleague's hand-made list.

- **Flow generator.** Wrote a Python notebook that generated dozens of near-identical flows from one template instead of building each by hand.
- **Consultant ratings.** Built the app, the data pipeline and the access system for rating external consultants. Managers rate their own consultants 1 to 5 and see only those in their own reporting line. A daily Python job matches **every active consultant** against HR data, and its count check passed on **every logged run**.
- **Stand-in access.** Expecting the question "my manager is on holiday, who rates their consultants?", I built a request app before anyone asked: another manager gets access for set dates and loses it automatically afterwards, so nobody tracks it by hand. It was used within two days of launch.
- **HR documents.** Automated personal HR documents from a spreadsheet: compensation statements and promotion certificates, each sent to the right manager.

Volvo Group — Gothenburg, Sweden
People Data Analyst — Sep 2024 – Nov 2024
Made data pipelines and Power BI reports faster (Power Query, DAX); automated data workflows in Python.

Volvo Group — Gothenburg, Sweden
Master's Thesis, Vehicle Data — Jan 2024 – Jun 2024
With a co-author, used time-series models on truck sensor, weather and road-type data to predict when drivers switch off a safety warning. Our main finding: about a quarter of the recorded switch-offs could not have been the driver and were likely logging errors, so we advised fixing the logging before building more models.

Volvo Group — Gothenburg, Sweden
Summer roles, logistics and people analytics — Summers 2023, 2024
Matched booking requests to shipping notices that shared no clean ID, with fuzzy text matching and a CatBoost model (PySpark on Azure Databricks). Built a regression in Power BI to forecast workforce make-up at one site.

Chalmers University of Technology — Gothenburg, Sweden
Research Assistant, Vehicle Safety — Feb 2023 – Jun 2023
Analyzed and labelled e-scooter ride data for the e-SAFER safety project.

KAGGLE

Top 10% in three finished competitions: **ARC Prize 2025** (124 of 1,455), **Biohub Cell Tracking** (194 of 3,947) and **Kaggriculture** (786 of 10,148). My public starter notebook for *MAP: Student Math Misconceptions* has **105+ upvotes**.

OPEN SOURCE

vLLM — Aug 2026
#52622, merged
Made the `/detokenize` endpoint answer bad requests with a 4xx error instead of 500, so a user's mistake no longer looks like a server outage. Also merged small fixes in LongHorizon-Harness and ai-agent-book.

EDUCATION

Chalmers University of Technology — Gothenburg, Sweden
M.Sc. Data Science and AI — 2022 – 2024

NMAM Institute of Technology — Karnataka, India
B.E. Computer Science — 2018 – 2022