

# JARED MAHOTIERE

Jared Mahotiere | Bear, DE | (302) 803-7673 | [jmahotie@purdue.edu](mailto:jmahotie@purdue.edu) | LinkedIn: [linkedin.com/in/jared-mahotiere](https://www.linkedin.com/in/jared-mahotiere) | GitHub: [github.com/jmahotiedu](https://github.com/jmahotiedu) | Site: [jmahotiedu.github.io](https://jmahotiedu.github.io)

## SUMMARY

Production operations and reliability engineer focused on incident triage, change-window execution, monitoring/alerting, and SOP-driven service stability across industrial and cloud platforms.

## EDUCATION

**Purdue University** - B.S. Electrical Engineering Technology (Computer Engineering Technology)

Minor: Computer & IT | Certificate: Entrepreneurship & Innovation | Expected May 2026

## LEADERSHIP & ORGANIZATIONS

**Delta Tau Delta (Campus Chapter):** DEI Chair | **National Society of Black Engineers (NSBE):** Member

## SKILLS

Operations: incident triage, escalation, SOP/runbook execution, change/release coordination, SLA-focused delivery | Monitoring: alert investigation, dashboard interpretation, Prometheus, Grafana, log-driven troubleshooting | Automation: Python, C#, TypeScript, Quartz.NET, GitHub Actions | Platforms/Data: SQL Server, Oracle QMOS, PostgreSQL, Redis, AWS (ECS, ALB, RDS, ElastiCache) | Mainframe Ramp: z/OS architecture, JCL/TSO/JES2/SDSF concepts, operations control discipline

## EXPERIENCE

**Nucor Corporation - Software/Automation Engineering Intern | Darlington, SC | May-Aug 2024 and May-Aug 2025**

- Implemented Quartz.NET automation for weekly hold/priority reports to 4 departments (Mill 1, Mill 2, Saw Cut, Scrap), replacing manual report distribution.
- Led cross-functional validation and rollout planning (test cases, sign-offs, punch-list closure) with operations, maintenance, quality, sales, and shipping for on-time delivery.
- Built the Hold Disposition Management system in Blazor Server (Telerik UI), integrating Oracle QMOS with a SQL Server-backed priority system to centralize hold-status decisions.
- Executed controls-focused startup validation, signal-path troubleshooting, and sensor/actuator commissioning to de-risk cutovers before production go-live.

## PROJECTS

**workflow-orchestrator - TypeScript, Redis Streams, Postgres**

**Links:** GitHub: [github.com/jmahotiedu/workflow-orchestrator](https://github.com/jmahotiedu/workflow-orchestrator), Live: [workflow-orc-demo-alb-1577468805.us-east-1.elb.amazonaws.com](https://workflow-orc-demo-alb-1577468805.us-east-1.elb.amazonaws.com)

- Implemented durable run/task state, idempotent retries, and dead-letter handling to improve operational resiliency and reduce repeat incident impact in queued workloads.

**Event Stream Platform - C#, .NET, WebSocket, WAL, Prometheus**

**Links:** GitHub: [github.com/jmahotiedu/event-stream-platform](https://github.com/jmahotiedu/event-stream-platform)

- Built WAL-backed ingest/replay workflows with observability, backfill, and reconciliation tooling to support recovery operations, troubleshooting, and post-incident validation.

**Feature Flag Platform - TypeScript, Node.js, Redis, React**

**Links:** GitHub: [github.com/jmahotiedu/feature-flag-platform](https://github.com/jmahotiedu/feature-flag-platform), Live: [feature-flag-demo-alb-1145770048.us-east-1.elb.amazonaws.com](https://feature-flag-demo-alb-1145770048.us-east-1.elb.amazonaws.com)

- Shipped deployment-safe rollout controls with RBAC, auditability, and deterministic rollouts to reduce production risk during releases and configuration changes.

## OPEN SOURCE CONTRIBUTIONS

- Bloomberg comdb2 (Java): fixed JDBC metadata cursor isolation to prevent getTables() result-set invalidation during version lookup (PR #5731).
- Bloomberg comdb2 (C/C++/SQL): backported targeted SQLite security fixes and validated with a source-build harness matrix (PR #5743).
- Cross-repo PR history (PicoClaw + Bloomberg): <https://github.com/pulls?q=is%3Apr+author%3Ajmahotiedu+org%3Aaspeed+org%3Abloomberg>.