

JARED MAHOTIERE

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

SUMMARY

Backend and systems software engineer focused on distributed services, C systems programming, .NET backend development, SQL performance optimization, AWS ECS deployments, and open-source architecture hardening.

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

Languages: C, C#, TypeScript, Python, Java | Backend: .NET, ASP.NET Core, Node.js, Express, REST, gRPC, concurrency, POSIX networking | Data: PySpark, Kafka, XGBoost, Prophet, FastAPI | Storage/Infra: PostgreSQL, SQL Server, Redis Streams, Docker, GitHub Actions, Prometheus, Grafana, AWS (ECS, ALB, RDS, ElastiCache), Terraform

EXPERIENCE

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

- Developed and maintained Blazor/.NET real-time operator dashboards and robust back-end services, enhancing process transparency and improving steel production workflows.
- Managed and analyzed production data in SQL Server/QMOS databases; developed optimized queries and recommended new tables/columns to support process improvement.
- Built automated reporting and alert systems using Quartz.NET with real-time email notifications for maintenance and quality events, reducing manual monitoring and accelerating response.
- Migrated legacy Visual Basic applications to .NET/Blazor, reducing technical debt; utilized Git for version control, peer code reviews, and codebase integrity.

PROJECTS

workflow-orchestrator - TypeScript, Node.js, Redis Streams, PostgreSQL, AWS ECS | [Project Link](#)

- Built a distributed workflow engine with DAG validation, Redis Streams consumer groups, idempotent retries, and durable Postgres run/task state; benchmarked 25/25 runs in 15.94s and validated AWS ECS Fargate deployment with ALB/RDS/ElastiCache.

IoT Streaming ETL Pipeline - Kafka, PySpark, Airflow, Great Expectations, Terraform | [Project Link](#)

- Built streaming ETL processing 100+ events/sec with Bronze/Silver/Gold architecture, checkpointer recovery, Great Expectations validation, and Prometheus/Grafana observability; provisioned AWS core infrastructure (MSK/S3/VPC/ECR) with Terraform.

Retail Sales Forecasting Dashboard - Python, FastAPI, XGBoost, Streamlit, AWS ECS | [Project Link](#)

- Shipped a live forecasting product on AWS ECS Fargate; achieved XGBoost R2=0.91 and ~11% MAPE with FastAPI inference, Streamlit UI, and 90%+ automated test coverage.

OPEN SOURCE CONTRIBUTIONS

databricks/cli (PR #4504) - Go, Auth/Config Governance | [Project Link](#)

- Fixed bundle-context auth precedence so explicit host/profile inputs win for non-bundle workspace commands.

sipeed/picoclaw (PR #213) - Go, Provider Architecture | [Project Link](#)

- Implemented protocol-family refactor to stabilize provider selection behavior and reduce edge-case failures.

bloomberg/comdb2 (PR #5743) - C/C++, SQLite Security | [Project Link](#)

- Backported targeted SQLite security fixes and validated behavior with source-build and harness verification.