

Rick Stoneback

Full Stack Software Engineer

rwstoneback.github.io

linkedin.com/in/richardstoneback

Essex, Vermont

rwstoneback@gmail.com

WORK EXPERIENCE

Lead Software Engineer & Tech Lead, Cox Automotive - 2019 to present - Burlington, VT

- Developed Kafka Streams applications for collection and transformation of automotive website analytics data, sent by thousands of websites including Autotrader and KBB.
- Developed GraphQL APIs to expose datasets via an Apollo federated supergraph.
- Created a url redirect service utilized by multiple parts of the organization to map consumer information with their website usage. Supported peak traffic of 300k RPM, using AWS Lambda, ECS, Fargate, API Gateway, Dynamo, and more.
- Developed serverless ETL pipelines to enhance and combine massive datasets, using AWS Glue, Lambda, ECS, SNS, SQS, S3, Dynamo, Aurora Postgres, and Step Functions.
- Worked with the Product Owner to groom features, as well as with Architects and Principals to bring new ideas and technologies into sight. Paired often with other developers on the team to work through issues, and held 1-1s to check in on concerns.

Staff Software Engineer, MicroStrain - 2008 to 2019 - Williston, VT

- Lead developer on [MSCL](#), a library used to communicate with MicroStrain's wireless and wired sensors, available in C++, C#, and Python. Directly supported customers using the library to develop their own software.
- Product owner and full-stack developer on [SensorConnect](#), the main PC software for all of MicroStrain's sensors. C++ backend, with HTML/JavaScript/WebGL frontend.
- Designed and developed sensorcloud.com.

EDUCATION

B.S. in Game Programming, Champlain College - 2010 - Burlington, VT

TECHNICAL SKILLS / EXPERIENCE

- Scala, Java, C++, C#, Python
- TypeScript, JavaScript, React
- Kafka, Kubernetes, Docker
- Terraform, CDK, Github Actions
- New Relic, PagerDuty
- SNS, SQS, S3
- Lambda, Glue, EC2, ECS, Fargate
- DynamoDB, Aurora PostgreSQL, Snowflake
- API Gateway, EC2, IAM
- Secrets Manager, Cloudwatch