

Shivu Donmardi Gorva

📍 Boston, MA ✉ donmardigorva.s@northeastern.edu 📞 617-606-8602 [in linkedin.com/in/dgshivu](https://www.linkedin.com/in/dgshivu) [📄 github.com/shivz3232](https://github.com/shivz3232)

Objective

Passionate Computer Science graduate student with 4.0 GPA specializing in ML Systems. Strong foundations in Machine Learning, Distributed Systems and Operating Systems. Experienced in building and operating production backend services in C, Golang, Node.js on AWS using Docker, Kubernetes, Terraform and Vibe Coding. Seeking to deepen hands-on experience with GPUs & Distributed Systems for AI inference infrastructure at scale at Meshy.

Education

Northeastern University | Masters in Computer Science **GPA: 4.00** *Sep 2025 – May 2027*
Coursework: Advanced Algorithms, Distributed Systems, OS Implementation, OO Design Paradigm

PES University | Bachelors in Computer Science *Aug 2018 – May 2022*
Coursework: Data Structures, Algorithms, Linear Algebra, Computer Networks, Compilers, DBMS

Technical Skills

Languages: C, Golang, Node.js, Python, Java, CUDA - OpenMPI - OpenMP (exposure)

Cloud: Git, AWS, GCP, Linux, Terraform, Docker, Kubernetes, Prometheus, Grafana, GitHub Actions, Ansible

Databases: MySQL, PostgreSQL, MongoDB, Redis, BigQuery, S3, DynamoDB, RDS


Distributed Systems: Chandy-Lamport Snapshot, SWIM protocols, DHT, 2PC & 3PC, PAXOS & RAFT (exposure)

Relevant Projects

Inovact – Social Platform for Students - AWS, PostgreSQL, Golang, Node.js [Inovact](#) 

- Designed and deployed microservices written in Golang and Node.js using Terraform on AWS powering Chat, Authentication, Mobile App Server, and Recruitment Dashboard containerized with Docker on ECS.

Systems Research & Implementation

Distributed Systems Snapshot Algorithm - C, Concurrency, TCP/IP, Docker [GitHub](#) 
Based on Lamport & Chandy (1985), "Distributed Snapshots: Determining Global States of Distributed Systems"

- Built low-level networking components from scratch: custom binary serialization/deserialization, linked-list message queues, and full-duplex TCP & UDP channels — similar in nature to IoT device communication stacks.
- Implemented multi-threaded C code using mutexes and condition variables for safe concurrent message processing, reflecting the synchronization demands of real-time embedded and RTOS environments.

Distributed Membership Protocol - C, Multi-Threaded, Docker, Kubernetes [GitHub](#) 
Based on "SWIM: Scalable Weakly-consistent Infection-style Process Group Membership Protocol"

- Designed and implemented a fault-tolerant distributed membership protocol in C, addressing the fundamental challenges of peer discovery, crash detection, rejoin handling, and leader election in an asynchronous network.
- Implemented a 3-Phase Commit-style leader election protocol triggered on leader failure, ensuring the oldest surviving node is elected while preventing split-brain scenarios during concurrent failures.

Professional Experience

BetterCloud | Backend Software Engineer - GraphQL, OOP, TDD | Chicago, IL (Remote) *Mar 2024 – Aug 2025*

- **Translated ambiguous service boundaries** to API contracts through architecture diagrams, resolving cross-team ambiguity and accelerating development readiness by 50%.
- **Led cross-team design** of a license reclamation workflow, achieving 100% adoption across target users — including 38% enterprise accounts — within 60 days.
- **Owned end-to-end design** and release of the Xero integration in Ruby, with self-service configuration and idempotent syncs, doubling adoption and reducing support issues by 90%.

G2 | Backend Software Engineer - AWS, Ruby, PostgreSQL | Bangalore, India *Jul 2022 - Mar 2024*

- Automated AWS integration setup via CloudFormation, cutting onboarding from a week to minutes and eliminating a 46% transaction drop rate.
- Designed PostgreSQL GIN indexes on transactional JSON data, cutting query latency by 50% and improving throughput 83% across 5 microservices.
- Revamped the enterprise SFTP ingestion pipeline using AWS Transfer Family and SNS, enabling real-time processing notifications to customers via Slack.