

Pratik Fandade

716-339-8155 | fandadep6@gmail.com | linkedin.com/in/pratikfandade | github.com/prkbuilds |

SKILLS

- **Programming:** C, C++ (STL), Python, Rust, Go | **Data & Storage:** PostgreSQL, MongoDB, Redis, SQL, BigQuery
- **Cloud & DevOps:** AWS, GCP, Docker, CI/CD, Gardener, RabbitMQ, Kubernetes, Terraform
- **Backend & Systems:** Distributed Systems, REST APIs, gRPC, cron-jobs, Caching, Threading, Synchronization
- **Reliability & Quality:** OpenTelemetry, Grafana, Logging, Metrics, Monitoring

EXPERIENCE

Redprint

Software Engineer

New York, US

Jun 2025 – Present

- Improved API reliability by reducing error rate from 1.2% to **0.2%** by adding **OpenTelemetry** tracing + structured logging, request tracing/correlation IDs, and SLO-based alerting (latency/error rate/saturation) on critical endpoints.
- Cut **MTTR** from 75 min to 20 min by building **Grafana** dashboards latency, queue depth, worker failures, DB health) and writing runbooks for top incident patterns.
- Reduced bulk-write time from 45s to **3s** by converting row-by-row updates to **set-based Postgres** operations, batching, and transaction scoping to minimize lock contention
- Scaled from 3k to **15k DAU** by tuning autoscaling, right-sizing compute, and caching hot reads; maintained stable tail latency during peak traffic.
- Reduced production regressions by **60%** by introducing health checks, graceful shutdown, canary rollouts, and automated smoke tests in the deployment pipeline

Colgate Palmolive

Software Engineer

Mumbai, India

Feb 2022 – Aug 2024

- Shipped a bulk-edit engine that updated **100,000+ rows** in ~1s (previously 30s) by using **set-based postgres updates**, batching, and async recomputation with **RabbitMQ** workers on **Kubernetes**.
- Improved async throughput **3x** and reduced peak backlog **78%** by splitting hot/cold queues, tuning worker concurrency/prefetch, and implementing retries + DLQ with idempotent handlers.
- Reduced p95 API latency from ~**800ms** to ~**150ms** by adding indexes, rewriting expensive queries, and moving recomputation off the request path into async jobs.
- Increased peak worker scalability from ~**30** to ~**100** concurrent workers during planning windows by tuning kubernetes autoscaling and right-sizing cpu/memory requests/limits.

PROJECTS

TacoDB | C++, SQL, Storage Engine, Buffer Manager

- Built a lightweight relational database with a disk-backed storage engine, buffer pool, and write-ahead logging to ensure durability and crash recovery.
- Implemented concurrency + correctness primitives (locking/latches, transactions) and optimized I/O paths via page caching, eviction policy tuning, and structured performance profiling.
- Designed query execution + planning with a cost-based optimizer and operators (scan/join/aggregate), focusing on predictable latency under large datasets.

Distributed Raft-based Redis | Go, gRPC, Raft, Docker

- Engineered a fault-tolerant, replicated KV store using Raft (leader election, log replication, snapshots) to provide consistency and automated failover under node loss.
- Hardened the system for production-like conditions by adding idempotent request handling, timeouts/retries, backpressure, and metrics for p95 latency, replication lag, and leader churn.
- Containerized and orchestrated multi-node clusters with Docker for repeatable deployments; built failure-injection tests (partition/kill/restart) to validate recovery guarantees.

EDUCATION

University at Buffalo, State University of New York (SUNY)

Master of Science in Computer Science and Engineering

Buffalo, NY

Aug 2024 – Dec 2025

Vishwakarma Institute of Technology, Pune

Bachelor of Technology in Information Technology

Pune, Maharashtra

Aug 2018 – Aug 2022