

AASHISH KUMAR

+918084662242 aashishkr.iitkgp@gmail.com linkedin.com/in/kr-aashish github.com/kr-aashish

Education

Indian Institute of Technology Kharagpur

Jun 2018 – May 2023

Dual Degree (B.Tech & M.Tech), CGPA: 8.52/10

Kharagpur

Technical Skills

Programming Languages: C++, Java, Golang, Python, JavaScript, TypeScript

Markup/Query Languages: HTML5, CSS3, T-SQL, pgSQL

Frameworks & ORMs: Spring Boot, Gin, Express.js, Flask, jOOQ, Mongoose, Prisma, Gorm

Databases: Redis, MongoDB, DynamoDB, Elasticsearch, SQLite, SQL Server, PostgreSQL, pgvector

Deployment: Docker, GitHub Actions, Azure DevOps Build & Release Pipelines, Deployment Groups, IIS, Nginx, Last9

Cloud: AWS S3, EC2, ECS, ELB, Lambda, RDS, ElastiCache, CloudWatch, SNS, SQS, SES, IAM, CloudFront, Route53

Experience

FlashMonk

Dec 2024 - Present

Software Development Engineer

Bangalore

- Redesigned **IMAP** service with event-based mail fetch, auto-retry, configurable domains, improved wait times by **16%**
- Enhanced email compose flow, reducing errors by **30%**, and migrated the mail view API to **ES**, cutting latency by **25%**
- Migrated **40 services** from CloudWatch to Last9 with **OTEL**, achieving **30%** cost reduction and better observability
- Optimized infrastructure by decommissioning **300M index**, **2 RDS instances**, 1 EFS, right-sizing ES/RDS clusters
- Migrated email mime data from **RDS to S3**, consolidating email storage architecture and reducing RDS costs by **30%**
- Re-engineered email information table with nested partitions and deleted **3B** promotional records from S3 and database

RootQuotient

Jun 2023 - Nov 2024

Software Development Engineer

Chennai

- Created Data Service APIs to migrate Cancer Registry System, from **PL/SQL** to **TS**, leveraging **MS SQL** & Prisma
- Migrated .NET Web Services with RPC client & SOAP APIs with WSDL in Node.js, consuming DLLs via FFI wrappers
- Scheduled **cron** job consuming HL7 documents with locks and **configurable** interval, scaled to cluster threads via **IPC**
- Wrote unit & integration test suites with a coverage of **85%**, optimized on-premise infra using Grafana's **K6** perf tests
- Configured build pipeline with static code analyzer, and release pipeline with Azure agents, deploying to Windows **IIS**

Amazon

May 2022 - Jul 2022

Software Development Engineer Intern

Bangalore

- Designed **UUID** for CGS Dashboard versioning and improved cadence from **weekly to every second** in the EMR job
- Revamped Audit Log DynamoDB to store metadata for CGS Dashboard, tracking published versions in CGS Version UI
- Reengineered the **CSV View** to render the dashboard directly from **S3**, **eliminating the cost** of maintaining the DDB
- Devised **CSV diff** algorithm to compare dashboard versions by primary index, providing list of changes to stakeholders

Projects

Jira 2.0

Apr 2024 - Jun 2024

Autonomous Project Management Suite

View System Design

- Crafted a **Lambda**, which triggers upon **FRD** upload to **S3**, and creates user stories using **Llama3** running in an **EC2**
- Facilitated **Event Bridge** to consume Slack messages, leveraged LLMs through **Flask** APIs to compile updates on tasks
- Employed **SES** to dispatch task approvals to stakeholders, then invoked issue tracker's APIs to reflect changes in backlog

DeployX

Jan 2024 - Mar 2024

UI Deployment Automation

Design Reference

- Built an API server with **RDS**, queuing build jobs and streaming logs to **Kafka** for **ClickHouse**, accessible via polling
- Provisioned an **ECS** cluster which runs **ECR** images in **Fargate**, to process the build jobs and dump the artifacts to **S3**
- Developed a **reverse proxy** for UI builds, streaming assets from S3, **mapping subdomain** to S3 key with RDS query

Aranpani

Nov 2023 - Dec 2023

Fundraiser Management System

OpenAPI Docs

- Modeled project, donor & payment APIs with **JWT** auth, **Joi** validation, opting S3 **presigned URL** for media uploads
- Penned **Unit** and **Acceptance** test suites, maintaining a code coverage of above **90%**, leveraging **Jest** and **SuperTest**
- Engineered resilience with **MongoDB** Replica Sets, **PM2** cluster & APM, **Nginx** web server and **Route53** name server