

Introduction

Full Name	Sai Srinivas Lakkakula
Email	saisrinivas.lakka@gmail.com
Phone	-
LinkedIn	https://www.linkedin.com/in/saisrinivasl/
GitHub	https://github.com/saisrinivaslakkakula
Portfolio Website	https://saisrinivasl.me

Work Experience

Software Development Engineer, Amazon Web Services – Jan 2023 – present

- Expanded the range of commercial engines available on AWS RDS by adding new engine, IBM DB2, resulting in a 20% increase in customer satisfaction.

Notes:

I currently serve as a software development engineer with the task of integrating a new commercial database engine into AWS Relational Database Service (RDS). This is a novel addition to the RDS lineup. My role involves working on the existing RDS platform's codebase, where I collaborate to create new microservices and modify existing ones. These changes aim to seamlessly incorporate the IBM DB2 database engine alongside other established engines like MS SQL Server, Oracle, MySQL, and Postgres.

Within the broader context of RDS, it's important to understand two main components: the Control Plane and the Data Plane. My primary focus is on the Data Plane. When customers request a new RDS DB2 instance, my team's responsibility is to orchestrate the allocation of necessary hardware and software resources for successful database installation. Moreover, we monitor these databases, enabling essential operations such as Create, Read, Update, and Delete within the created databases. We also design and implement vital microservices that cover a spectrum of functionalities, including database backups, restoration, security patching, log management, and monitoring metrics.

The linchpin of these processes within the Data Plane is the Host Manager Service. This Java-based application functions persistently on the host environment, facilitating continuous and bidirectional communication between the Control Plane and the Data Plane. It's essential to

note that requests, whether in response to failures or customer needs, can only be initiated from the Control Plane. To sum up, envision the Control Plane as the commanding officer of the RDS Instance 'army,' while the Data Plane serves as the indispensable communication conduit for each individual 'soldier.'

- Developed APIs that help Database migration from on-premise to RDS DB2 via S3 buckets in 2 weeks, saving 10 hours of manual work per week.

Notes:

Having explored the internal workings of an RDS, an intriguing aspect emerges in relation to IBM DB2. This well-established database software has existed for decades, but its users have largely confined themselves to on-premise servers due to the absence of cloud technology during the software's initial development. With RDS being a fully managed cloud service, a significant challenge arises: migrating existing on-premise databases to RDS.

As RDS Engineers, we devised a solution to enable the seamless migration of customer databases from on-premise to RDS. This solution entails a semi-automated process. In this process, the on-premise Database Administrator first creates offline backups of their databases and then uploads these backups to AWS S3 buckets, which are our in-house AWS storage service. Subsequently, the Database Administrator utilizes an API provided by our team to transfer data from the S3 buckets to RDS Databases.

I undertook the task of designing and developing the second segment of the API call, which empowers DBAs to initiate the API call and thereby restore data to RDS databases. Completing this task within a span of 2 weeks, our team accomplished a significant milestone. This newly established process effectively reduces manual DBA workload by an estimated 10 hours per week.

- Improved the functionality of RDS DB2 Host Manager Service by uploading local logs stored in Host Manager to AWS CloudWatch periodically, reducing disk space usage by 80%.

Notes: As previously discussed, the Host Manager service holds pivotal importance within the Data Plane. A specific scenario arises wherein the operations team (comprising our engineers) requires access to logs from the Host Manager (HM) Service. These logs become crucial during instances of single point failures or for debugging purposes. Internally, due to constrained disk space, the HM is constrained to retain logs for only 2 days, after which they are automatically deleted. This limitation poses a challenge for engineers attempting to debug issues that occurred beyond this timeframe.

To address this challenge, a solution emerged: the need to store these logs in a cloud-based location with an extended retention period. I spearheaded the design and development of an

end-to-end service to facilitate this. The service operates through a recurring job within the host manager. This job is responsible for uploading logs to the AWS CloudWatch service.

The introduction of this system brings forth several advantages. By uploading logs to AWS CloudWatch, we can significantly extend the retention period of logs, allowing us to maintain them for a prolonged duration (typically set at 30 days). This mechanism grants engineers access to historical data beyond the limited 2-day window, enhancing their ability to diagnose and troubleshoot issues. Furthermore, the accumulated logs in CloudWatch also enable us to derive valuable analytics and insights from the data, contributing to an improved understanding of system behavior and performance.

- Resolved customer issues with a 95%+ success rate, 30-min avg resolution time, boosting customer satisfaction & product quality.

Notes: As part of our role, all engineers, including myself, take turns being on-call for a week every 2–3 months. During this time, we receive customer tickets that highlight issues they're facing. Our job is to quickly address and fix these problems. I'm proud to mention that I've been quite effective, managing to resolve around 95% of the customer tickets that have come my way.

Another accomplishment I didn't include in my resume is the development of Java-based asynchronous stored procedures. In the world of Database Management Software (DBMS), there's a feature that lets you use external stored procedures. These are like special tools that DBMS can use, written in languages like Java, C, or C++, when the regular tools like SQL can't get the job done.

In our application, there was a situation where we needed to do something a bit more complex than what regular SQL could handle. It involved working with files in a special way (I can't provide all the details because of security concerns). To solve this, I created Java-based external stored procedures. These procedures act as a bridge between SQL and Java, allowing us to use Java code to handle these special file operations that SQL couldn't manage on its own.

Software Development Engineer Intern June 2022 - August 2022 **Cisco Systems — WebEx San Jose, CA**

- Developed Java-based APIs in WebEx Platform Shared Services that enhanced client-side error logging and search functionality.
- Assured the code quality by designing test cases with 100% coverage in both Unit and Integration tests.
- Actively Monitored production incidents and ensured the services availability to maintain 99% uptime.

Notes:

During my time as a Software Engineer Intern at Cisco's Webex Collaboration Business Unit, I had the opportunity to contribute to the Webex suite, a product designed for team collaboration and communication. Within this unit, I focused on a service within the Webex backend known as "client-logs." This service is responsible for uploading client logs to an on-premises PostgreSQL database. These logs are crucial for operations personnel to diagnose critical failures and extract valuable insights.

My assignment during the internship was to create an API that would enable efficient log retrieval and pagination. This API needed to accept input parameters such as the log resource name, start and end date-time stamps, and the desired number of results per page. The challenge was to organize the logs based on dates, calculate the number of records according to the provided offset parameter from the API, and then deliver the segmented results back to the client.

I successfully completed this task within a two-week timeframe, including unit and integration testing. I did encounter a setback during the internship due to being ill for a couple of weeks because of COVID-19. Despite this, over the course of the 10-week internship, I also dedicated time to onboarding tasks, studying relevant documents, and familiarizing myself with the team's objectives and the organization's culture.

While the tasks I undertook might not have had a significant impact on the overall scope, I genuinely enjoyed my experience at Cisco. It provided me with insights into Cisco's business offerings, allowed me to become acquainted with various teams, and immerse myself in the organizational culture. The work environment and the inclusive atmosphere that the team provided made my time at Cisco enjoyable and enriching.

In addition to my primary tasks, I also took on the role of mentoring a team of high school students for Cisco's internal hackathon competition. I had the privilege of guiding and supporting this team throughout the competition. I'm proud to share that the team I mentored achieved a remarkable second place in the competition. This experience allowed me to contribute to the development of these young talents and witness their success firsthand.

Software Development Engineer - II Oct 2017 - Dec 2020
Capgemini Technology Services Chennai, India

- Developed PHP based APIs for 'Namaste Capgemini' Android/iOS mobile application and streamlined 70% organization's logistic operations impacted by COVID-19.

Notes:

This accomplishment holds a special place for me. We actively contributed to the development of an internal mobile application named "Namaste Capgemini." This initiative began during the early days of the COVID-19 pandemic when the entire country was under lockdown, including IT companies. However, some exceptions were made for critical resources such as networking and IT support staff, as well as Software Engineering leads engaged in crucial financial services projects.

To facilitate these exceptional cases, specific permissions were granted by local government agencies. Employees in these roles were required to sign a health declaration form indicating their absence of COVID-19 symptoms and their non-residence in any government-designated containment zones. Moreover, the organization needed to implement COVID-19 precautions for these employees, including arranging transportation, providing isolated workspaces, and ensuring proper cleaning protocols were followed.

Recognizing the need to streamline this complex process involving multiple teams, we devised an automation solution. The concept was to create a system where business leaders could input basic employee details using their exclusive accounts. Once this data was entered, employees mandated to work in-office would receive an email a day in advance, containing specific instructions. As part of this process, employees were required to download a mobile app called "Namaste Capgemini."

As the backend architect for the application, I played a pivotal role in its development. Our team crafted PHP-based APIs to support functionalities such as business leader login, automated emails and text messages for employees, generation of unique employee entry IDs for QR codes, and more.

The app's workflow was as follows: after downloading and logging into the app with CORP details, the employee's information was synced. If in-office attendance was needed, the app displayed this requirement prominently. The employee was prompted to sign a health declaration form, generating a unique QR code for office entry upon completion. The app also offered transportation details, workstation numbers, emergency contacts, up-to-date COVID-19 information, and FAQs.

Our efforts didn't go unnoticed. The application earned a nomination for the Aegis Graham Bell Tech Innovation Award in 2020. While we didn't reach the finals, the nomination itself was an accomplishment we're proud of, recognizing the innovation and impact of our work.

- Pioneered ANTLR and Java based automatic code analysis and review tool for legacy target programming languages. Peddled up technical code review process by reducing 78% of manual effort.

Notes:

Here, I had the chance to showcase my innovation and R&D skills within Capgemini. As a member of the Center of Excellence (CoE) team, I was tasked with developing custom tools and

automation solutions for both internal teams and external clients. One particular challenge I undertook was to create a check-style tool for mainframe COBOL applications, a task typically hampered by the limitations of the mainframe environment.

In the realm of mainframe coding, code reviews present a unique challenge. These reviews need to be conducted manually by peer programmers, examining individual code changes. This process can be time-consuming and frustrating due to the lack of integration with modern CI/CD tools, and the fact that peer reviewers work in a basic black and green screen environment, incompatible with sophisticated code IDEs.

To address this, I embarked on a solution-building journey. My research led me to ANTLR (Another Tool For Language Recognition), a framework rooted in Compiler Design principles. ANTLR takes the grammar of a target programming language, generates an abstract syntax tree for COBOL programs, and parses them into the tool. However, ANTLR doesn't offer a graphical interface; it generates a data structure called a tree and stores it in memory.

Building on ANTLR's foundation, I created a Java application. This application traverses the Abstract Syntax Tree, cross-referencing the COBOL program's syntax against pre-set rules. Once the traversal is complete, the application generates a .csv file detailing coding rule violations: the number of violations, line numbers in the source COBOL program where the rules were broken and suggested corrective actions.

Engineers working on COBOL programs found this tool immensely helpful. In one instance, a client required adherence to around 250 coding rules/guidelines during code reviews. My tool automated nearly 195 of these guidelines, achieving a remarkable 78% automation rate. If we consider a reviewer spending an average of 10 minutes for a 100-line COBOL program, the tool's impact is profound. The estimated review time reduction of 78% translates to just 2-2.5 minutes for each 100 lines of COBOL code, revolutionizing the review process.

- Accelerated 50% of organization level talent and resource management operations by designing and developing a cloud-based web application called MY360.

Notes:

I was involved in creating a tool called MY360, designed exclusively for internal HR personnel. This tool tackles a common challenge in service-based IT companies: the concept of a bench. The bench refers to a pool of resources not assigned to specific client projects but kept readily available for immediate deployment based on business needs. This model is prevalent in most service-based or consulting firms. However, the HR leadership recognized an issue with this setup.

Typically, bench resources are associated with a specific skill-set family. For instance, someone tagged as a Java developer would mainly be considered for client interviews requiring Java skills. This approach limits resource utilization and neglects their secondary or newly developed

skills. MY360 addresses this problem by enabling HR personnel to identify resources based on skill shortages rather than predefined skill families.

For example, if an employee has a secondary skill as a Python developer but is primarily tagged as a Java developer, MY360 allows them to be considered for projects needing both Java and Python expertise.

The tool collects employee details, skills, achievements, certifications, and recommendations from supervisors. It employs an advanced algorithm that recommends resources to HR personnel based on skill gaps and requirements. HR can input specific criteria such as skill sets, years of experience, certifications, location preferences, etc., and receive a curated list of recommendations. In essence, MY360 functions like an internal LinkedIn for talent optimization.

The tool's impact has been significant. HR personnel no longer need to manually sift through profiles for hours to find the right matches, as was the case before. With MY360's precise recommendations, the process is streamlined, achieving a 50% automation from the HR side. However, candidates are still required to update their skills periodically, maintaining a manual aspect to their application. Overall, the estimated automation from both sides is roughly 50%.

- Performed Legacy Code Analysis using CAP360 In-house application for multiple Banking, Insurance clients. The tool identified 85% technical debt and optimized MIPS effectively.

Notes:

In contrast to MY360, we developed a Software as a Service (SAAS) application named CAP360, catering to external customers. The primary objective of CAP360 is to conduct Legacy Code analysis within mainframe applications, pinpointing redundant or unused code components. Mainframe technology is often considered dated, and many of the original programmers responsible for writing code, even for major fintech companies globally, may have retired. Consequently, outdated versions of source code might remain within mainframe servers without being actively used. This inadvertently drives up costs for customers, as these applications should ideally be retired but still persist in the systems.

CAP360 addresses this issue by comprehensively scanning the entire codebase of any mainframe application. Its purpose is to detect orphaned components or application subsystems. This information empowers customers to cross-check and decide whether decommissioning is necessary. This tool effectively assists customers in identifying and addressing redundant code within their mainframe applications, thereby optimizing costs and improving overall efficiency.

In addition to my other achievements, I've had the opportunity to create full-stack web applications to enhance various experiences. For internal hackathon competitions, I developed

a platform that streamlines the organization and management of these events, making them more accessible and engaging for participants.

Furthermore, I worked on crafting customized websites tailored to specific client visits. These dedicated websites offer comprehensive details about the visit, including day-by-day agendas, information about hotel and flight bookings, logistical guidance, and more. These applications serve to enhance client experiences, providing them with a user-friendly platform to access and manage all the pertinent information related to their visit.

Write up draft I am using when this question is being asked in the job applications.

Please provide an example or evidence of your exceptional ability in problem solving.

One of my most exceptional abilities was kindled during my freshman year in college, when my affinity for problem-solving and programming began to flourish. Armed with an Electrical Engineering background, I was naturally drawn to the intricacies of logic and computation. Little did I know that this passion would lead me to create a simple yet impactful solution that marked my first significant achievement in the world of programming.

In those early days, my college's grading criteria posed a perplexing challenge. Students were grappling with manual calculations to estimate their internal grades before final exams. Recognizing this pain point, I embarked on a journey that would eventually shape my exceptional problem-solving ability. With determination and a coding mindset, I developed a nifty C program – a modest executable file – that I distributed on flash drives to my fellow students. This unassuming grade calculator application swiftly became the go-to tool for calculating estimated grades for each course. While it lacked the glitz of a polished user interface, it symbolized my initial strides into the realm of programming and innovation.

But that was just the beginning. As I progressed through my college years, my knack for tackling real-world challenges through coding only grew stronger. With an insatiable curiosity, I created a variety of applications that catered to specific needs. One notable example was an online slam book, designed to allow senior students to capture and share cherished memories with friends. Another venture involved crafting a digital transaction app tailored for the classic Monopoly board game, accommodating up to four players. These endeavors were not mere academic projects; they were the fruit of my insatiable thirst for innovative problem-solving.

My drive to conquer new technological frontiers became an intrinsic part of who I am. With each application, I delved into unfamiliar technologies from scratch, fearlessly navigating uncharted territory. This eagerness to push boundaries, coupled with my persistent pursuit of innovative solutions, sets me apart. These examples highlight a core trait of mine – the relentless passion to explore and excel in problem-solving. They showcase my dedication to

learning, my penchant for innovation, and my unwavering commitment to mastering new technologies.

As I reflect on those early days, I'm reminded that even the smallest steps can lead to monumental achievements. My journey, from crafting a simple grade calculator to creating diverse applications, echoes my profound interest in technology, innovation, and the boundless potential of problem-solving. I am excited to bring this exceptional ability to your team, to contribute and thrive in an environment that values creativity, curiosity, and pushing the boundaries of what's possible.

Academic Projects:

MySmart Cal - A cloud-based public calendar

Time is precious. People who value time should always try their best to improve their productivity. Be it a freelancing activity or free community service, it is often difficult for people offering the services to coordinate the right time window and location with the people who need to utilize their services. What if there exists a simple application that solves the problem of finding the correct matches? What if a person likes to volunteer for community service and could add their available free time on a public platform so that the needy can book their appointment? Yes, the Smart Cal application is an innovative solution that uses technology to solve this problem.

Finding and keeping clients is the first and most difficult challenge for freelancers or service providers (users in other words). It is also a never-ending task. For example, a software engineer, who wants to offer career guidance to the students preparing for coding interviews in his available time, has no platform to connect with them. This often becomes a challenge, but it can be overcome by maintaining transparency in time and commitments between the clients and users.

To address the discussed problem, we are implementing a shared calendar application, Smart Cal. Users and the clients who employ them are the two personae of this application. Users can specify their services and available hours. Employers or clients can search for required services and schedule an appointment. Other common functionalities, such as editing existing appointments, searching for a freelancer or service provider, managing availability, etc., will be part of our application. This application will act as a bridge between freelancers and clients to manage their work and time effectively.

Technologies used: React.js, JSX, HTML, CSS, Redux, Java, SpringBoot, SpringData, MongoDB, Apache Kafka, AWS EC2, git, SDLC, Agile.

UberEats Website Clone

Introduction:

The purpose of this project is to design the system of online restaurant order management system (OMS). The order management system chosen for this project is one of the famous food

delivery applications called **Uber Eats**. The end goal of the project is to design the OMS system with 75% of estimated replica functionality of original application.

System Design:

Persona:

In the OMS System, there are 2 personae.

1. User – Who places the order
2. Restaurant – who manages the restaurant, tracks the order and changes it.

Front End:

React – A powerful JavaScript Framework used for building single page applications. Redux – A state management library used by react to manage global state of the application. JSX (Embedded HTML in JS) CSS – Cascaded Style sheets.

Middleware:

Kafka: Kafka is a message queue based distributed system used to decouple the backed architecture into micro services. Kafka works on publisher, subscriber model where the client sends the request messages to the Kafka queue into the certain topic. The backend micro service is subscribed to the same topic, consumes the request, executes the business logic and sends back the response to the response topic queue. Passport JWT: Passport is the strategy used to verify the authenticity of incoming request by using JSON Web Token. node.js has a special package called passport-jwt which provides useful functions to verify the incoming request and check it against the JWT. After checking the authenticity, the particular request maker is authorized to use the requested resource.

Backend:

Node Js – A JavaScript runtime on the server side based on Google Chromes V8 Engine and supports ES6 features of JavaScript Express – Nodejs Framework for handling incoming requests and outgoing responses. Database : MongoDB – An open-Source Schema Free No SQL Database Technology which supports its native query language called MongoDB Query Language (MQL). MongoDB stores each record into its granular entity called Document. The document is of type JSON with data as key-value pairs. NoSQL is flexible as it eliminates features like joins and Sub-Queries. Also, the referential entities can be reduced as much as possible because all the logical data can be stored in a single document and the collection of documents is called as 'collection'

Miscellaneous tools:

Redux Dev Tools Extension – Very useful tool in debugging the Redux related issues in the development phase. PostMan – Web/Native HTTP Client used to test APIs MongoDB Compass – A multi-platform tool provided by mongoDB for to manage the database and it provides GUI for the database. Database Schematics Design:

- Since the application is migrated from MySQL to MongoDB, many dependencies can be removed. especially the entities that require joins as NoSQL's power comes with no joins.
- The design is thus modified by combining orders and order details in the old design into single collection of documents called orders.
- Similarly, In the earlier design, favorites were added into separate table with restaurant ID and CustomerID as foreign keys. In the new design, the favorites is combined with user data as favorites as a field in the user document.

- User and restaurant address is simplified by adding main field as address and the document is subdivided into street, city, state etc.

Airplane Crash Analysis using the methods of Data Mining

You hear the pilot yelling “Mayday Mayday!” over the intercom. What do you do? Brace? Hug your loved ones? Turn off airplane mode and text your family? We hope you never have to experience anything remotely close to this but the recent 20 year memorial of 9/11 got us thinking about what causes a plane to crash. Though we cannot completely eradicate the mishaps of airplanes, we will study the causes of fatalities and use various data mining techniques to determine the risk associated with flying.

Our dataset comes from the National Transportation and Safety Board and consists of over 74 thousand accidents and incidents from 1948 to 2013 (<https://data.nts.gov/avdata>). The problem that we set out was to determine whether aviation is becoming safer as time moves forward. This is quantified by analyzing the number of fatal accidents with respect to relevant crash data as well as quantifying the total number of accidents that have occurred overtime. Not only that but we set out to determine if a crash can be predicted given various factors/features. If a crash can be predicted then we will be able to evaluate the likelihood that a crash will occur and would eventually be able to step in and stop the flight. In our analysis, we have considered specific parameters that are crucial for predicting (or can lead up to) a crash, such as no. of engines, weather, etc. With these pointers the manufacturers and air crew members can take extra caution before taking off.

This was performed with techniques such as dimensionality reduction to cluster whether certain types of crashes are fatal, linear regression to predict the total number of fatalities, logistic regression to determine the strongest flight predictors to lead to a fatal crash, random forest to binary classify if a crash would be fatal or not. More in-depth details of these processes are seen in the Methods section.

We plan to have our project roll out in 3 phases. Phase 1 being the cleaning of data, getting it into a usable format for future EDA and Data Mining Algorithms. Phase 2 is the EDA, where we will investigate the data and draw inferences by plotting graphs and charts. Phase 3 being the portion where we implement various concepts introduced to us in the course and other topics previously learned. As of right now, the methods that we are using are: Machine Learning (Random Forest, Linear and Logistic Regression), Dimensionality Reduction (PCA and possible MCA). We want to use Machine Learning algorithms to determine if there will be severe fatalities (binary 1 or 0) given factors such as plane type, season, time, location, operator, etc. Since the end goal is being able to predict the correct number of fatalities we will use measures of success such as: Accuracy, Precision, Recall, Mean Square Error (MSE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE). We will do multiple train test splits on the datasets and compare them using the measures of success listed as well as using cross validation to ensure that our values are as best as they can be.

Quora insincere Text Classification using the methods of ML

The objective of this project is to identify insincere text within the Quora platform, encompassing both question text, user responses, and comments. Manual inspection of content can be a laborious task for content moderators as it involves scrutinizing text, sentences, or words that deviate from Quora's community guidelines. To streamline this process, we've leveraged Machine Learning techniques such as Bag of Words, TF/IDF, and Sentiment Analysis to create an ML model capable of assessing the insincerity score within parsed Quora text. This innovation aids content moderators in automating the content moderation process effectively.