Sreekanth Reddy Balne
AWS DevOps Engineer | #AWSCertified #TerraformCertified
Mobile: (603) 367-3999
sreekanthreddybalne@gmail.com

I am a Software Engineer with a solid grounding in **Python**, Java, **AWS**, **DevOps**, **SQL**, **NoSQL**, **Terraform** and various **Frontend** related technologies, aiming to leverage a meticulous and balanced approach to developing high-quality, scalable solutions in a structured environment that values precision and practical implementation.

CERTIFICATIONS & PORTFOLIO

- AWS Certified Solutions Architect - Associate (Click to see certification)
- HashiCorp Certified Terraform Associate (Click to see certification)
- Certifications: (click to see all certifications)
- LinkedIn: (click to see LinkedIn profile)
- Github: (click to see Github profile)
- Github OpenSource work: (click to see open source contributions)
- StackOverflow: (click to see contributions on StackOverflow)

KEY ACHIEVEMENTS

- **Strategic Integration Expertise**: At Limit (Apollo Brokers Inc.), spearheaded the development of a unified insurance application submission system, minimizing the applicant's time spent on the application while enhancing the accurate submission of information to various insurance providers.
- **Data Lake Architect**: Implemented a Data Lake solution at Factset, managing large-scale data storage and analysis, optimizing workflows, and accelerating deployment timelines.
- **DevOps Innovator**: At Cognerium Robotic Labs, designed and implemented automated CI/CD pipelines, streamlined deployment processes, and established monitoring solutions to enhance system reliability and performance.

Python

- Strong command of Object-Oriented Programming (OOP) concepts and Object-Oriented Analysis, consistently applying this knowledge to architect scalable and maintainable applications.
- Proficient in building **REST**ful **API**s using Python frameworks such as **Django**, **Flask** and **FastAPI**.
- Proven track record of successfully integrating with third-party applications and libraries, fostering interoperability and enhancing overall system functionality.
- Proficient in using various Python libraries such as **Pydantic, Pandas** and **PySpark** for large-scale data processing.
- Exceptionally skilled in constructing Real-Time applications with WebSocket communication, delivering responsive and dynamic user experiences.
- Follows **Test Driven Development** (TDD) principles by writing comprehensive **unit tests** before any code is developed and performing thorough **integration tests** to ensure all components function seamlessly together, thereby maintaining overall system integrity.
- Superior Debugging and Troubleshooting experience with various programming related issues.
- Adheres strictly to the **DRY** (**Do Not Repeat Yourself**) principle, promoting code efficiency and maintainability.

Data Engineering

- Knowledgeable in **SQL** and **NoSQL** databases such as **MongoDB** and **DynamoDB**.
- In-depth knowledge of **Data Management** - **Data Extraction**, **Data Validation** and **Data Cleaning** of both structured and unstructured data.
- SQL Skills – Proficient in writing SQL queries, including sub-selects, joins, window functions, normalization principles and indexing strategies.
- Contributed to the development of a **Data Lake** system to store and process structured and unstructured data, and provide access to data/metadata to various downstream applications, especially Machine Learning teams.
- Implemented various feed pipelines for the ingestion and categorization of PDF, CSV, XML and JSON files.
- Experience working with **Parquet files** for efficient storage and retrieval of large datasets.
- Proficient in running data transformations using **Pandas** and **PySpark** for large-scale data processing.
- Have used Apache Airflow for orchestrating complex **Data Workflows**.
- Experience in implementing and managing data streaming solutions using AWS Kinesis.
- Skilled in using **Kibana** and Elasticsearch for log analysis and visualization.
- Proficient in using **Retool** to build custom internal tools for data analysis and process automation.
- Developed and maintained **ETL pipelines** using Databricks to process, transform, and analyze large datasets, leveraging Apache Spark for distributed data processing.
- Implemented data governance and security measures within Databricks, ensuring compliance with industry standards and best practices for data privacy and protection.
- Extensive experience with XML technologies including XML, XSL, XSD, XSLT, XQuery, XPath, and DTD.

DevOps

- Proficient in **Python scripting**, focusing on **DevOps**, **CI/CD**, and AWS **Cloud Architecture**.
- Skilled in design, development, and implementation of various stand-alone, client-server, and cloud architecture-based applications in AWS across various domains.
- Superior troubleshooting and technical support abilities with data migrations, network connectivity, security, and database applications.
- **AWS Cloud services**: VPC, Security Groups, Internet Gateway, IAM, Lambda, S3, SQS, SNS, EC2, ECS, ECR, EKS, RDS, DynamoDB, Elastic Cache (Redis), Athena, EMR, Redshift, Glue, SageMaker, Kinesis, Cloudwatch, EventBridge, API Gateway, KMS, Secrets Manager, Config, CloudFront, Route53, ELB, Batch, Auto Scaling, CodeCommit, CodeBuild, CodePipeline, Step Functions, S3-Select, CloudSearch, Amplify, AppStream 2.0, Workspace Thin Client, etc.
- Experience in deploying and managing big data frameworks on AWS EMR and optimizing data pipelines for large-scale data processing.
- Proficient in designing and deploying **public** and **private** facing **network** applications on AWS.
- Highly efficient in configuring cross-account deployment using AWS CodePipeline, CodeBuild, and CodeDeploy by creating cross-account policies and roles on IAM.
- Experienced in using Splunk, **Kibana**, Elasticsearch, and AWS CloudWatch for log management, monitoring, and analysis.
- Experienced in creating Docker containers leveraging existing Linux containers and AMIs, and creating Docker containers from scratch.
- Experience with orchestration tools like AWS Step Functions and Apache Airflow.
- Proficient in designing, developing, and deploying microservices architecture.
- Provided installation, configuration, and maintenance of pre-production and production cloud environments.
- Adept at managing multi-environment deployments, ensuring seamless transitions between development, staging, and production environments.
- Skilled in using tools like **SonarQube**, Pylint, and Black for continuous inspection of code quality and security.
- Certified Terraform Associate, proficient in provisioning and managing cloud resources, with knowledge of AWS CloudFormation.
- Deploy, maintain, support, and troubleshoot resources in a cloud environment.
- Evaluate, test, deploy, and maintain both custom-developed and third-party software upgrades.
- Maintain systems such as test environments, source control, and automated build/test/deploy systems to support SDLC and services.
- Maintain production services to meet SLAs.
- Take ownership of production issues, working closely with development and customer success teams on issue resolution.
- Provide developer support on an ongoing basis, frequently collaborating with development teams.
- Provide 24x7 production support as part of a team rotation, resolving or escalating issues as appropriate.
- Support releases on a regularly scheduled basis, as well as emergency releases as needed.
- Deploy application and data changes to stage and production environments as needed.
- Proficient in troubleshooting and resolving production issues.
- Design and implement new environments, services, infrastructure, and application architecture modifications.
- Research, evaluate, and implement operational improvements, application packages, and architectural modifications.
- Participate in change control, release planning, and other operational planning.
- Remain current on industry-leading solutions for public cloud hosting (primary focus on Azure), including deployment, scaling, monitoring, data storage, and cost management.
- Remain current on industry-leading solutions and best practices for infrastructure and application security tools and technology.

Frontend

- Proficient in building dynamic and responsive web applications using HTML, CSS, and **JavaScript**. Experienced with both JavaScript and **TypeScript** for developing robust and maintainable codebases.
- Skilled in leveraging modern JavaScript frameworks and libraries such as **Angular** and **ReactJS** to create Single Page Applications (**SPAs**) that provide excellent user experiences.
- Extensive experience in integrating REST APIs into frontend applications to ensure seamless data exchange and functionality.
- Adept at implementing secure authentication mechanisms to protect user data and ensure safe user interactions.
- Knowledgeable in using **Figma** for UI/UX design.
- Experienced with CSS frameworks like Bootstrap and CSS-in-JS solutions to expedite development and maintain consistent styling.
- Skilled in using **RxJS** for managing state and handling asynchronous events, ensuring efficient data flow within applications.
- Competent in using **Node.js** for server-side development, enabling the creation of full-stack JavaScript applications.
- Focused on optimizing frontend performance to improve load times and user experience through techniques such as code splitting and lazy loading.

Others

- Around 3 years of experience working with **Java** and **Spring Boot**.
- Proficient in all stages of the Software Development Life Cycle (**SDLC**) including Analysis, Design, Development, **Debugging**, **Support**, **Integration**, Maintenance, and Enhancements of various applications.
- Experienced in both Agile (SCRUM) and Waterfall development methodologies.
- Familiar with project management tools like JIRA, Linear and Trello.
- Strong interpersonal skills with experience coordinating with development, QA, and support teams.
- Proven leadership experience in leading a development team, ensuring project delivery, and mentoring team members.

SKILLS

Languages	Python, Java, Perl, SQL, Terraform
Python Frameworks & Libraries	Django, django-rest-framework, Flask, FastAPI, PyTest, Pandas, PySpark
AWS Services	VPC, Security Groups, Internet Gateway, IAM, Lambda, S3, SQS, SNS, EC2, ECS, ECR, RDS, DynamoDB, Elastic Cache(Redis), Athena, EMR, Redshift, Glue, API Gateway, Kinesis, CloudWatch, EventBridge, KMS, Secrets Manager, Config, CloudFront, Route53, ELB, Batch, Auto Scaling, CodeCommit, CodeBuild, CodePipeline, Step Functions, S3-Select, CloudSearch, Amplify, AppStream 2.0, Workspace Thin Client, etc
GCP Services	Firebase(Authentication, Cloud Firestore, Cloud Functions, Remote Config, Storage)
Databases	MySQL, PostgreSQL, SQL server
Data Engineering	Databricks, Data Lake, Data Pipelines, ETL, ELT, Data Migration
NoSQL Databases	DynamoDB, MongoDB, Cassandra
Data Storage and Analysis	Elasticsearch, Kibana, Redis
CI/CD	Github Actions, Bitbucket Pipelines, AWS CodePipeline, AWS CodeBuild, Circle CI, Jenkins
Operating Systems	Windows, Linux, Ubuntu
Web Services	WSDL, JSON, SOAP UI, REST API
Web technologies	NodeJS, ReactJS, NextJS, Angular, Groovy, JavaScript, TypeScript, Express.JS, jQuery, XML, JSON, CSS3, HTML, XHTML, Bootstrap, AJAX, React Native, NodeJS
Design Tools	Figma, Adobe Photoshop, Adobe After Effect
Build tools	Docker, Maven, Jenkins
Infrastructure as Code (IAC)	Terraform, CloudFormation
Version Controls	SVN, CVS, GitHub, Bitbucket
Others	WebSocket, Webhooks, Web Scraping, Regular Expressions, Postman, Kotlin, Android Development, Apache Zookeeper, RabbitMQ, ActiveMQ, Apache Kafka

PROFESSIONAL EXPERIENCE

Limit (Apollo Brokers Inc), California **Jun 2022 — Till date**
Role: Lead Developer/AWS DevOps Engineer

Limit is a digital wholesale brokerage firm that focuses on providing streamlined insurance solutions, particularly in the areas of cyber, professional liability, and management liability insurance. The platform aims to simplify the insurance quoting process for retail brokers by integrating them with a wide array of carriers. This allows brokers to get instant quotes and access to more than 50 carriers, saving significant time and effort. It has recently launched Limit AI – an AI assistant for insurance brokers and underwriters.

Key Achievements:

- Enabled integration with 18+ insurance carriers, improving data submission accuracy by 40% and reducing processing time by 30%.
- Achieved high system uptime and boosted performance, supporting significant user base growth.

Partner Integrations:

- Spearheaded the development of integration solutions between the company's platform and multiple insurance carriers, enabling efficient data exchange.
- Provided expert technical support and problem-solving for complex integration challenges.
- Acted as the primary point of contact for partners, ensuring clear and effective communication.
- Collaborated with carrier partners to understand ongoing changes, enhancements, and implementations necessary to keep our systems up-to-date, ensuring seamless processing of their API responses without any failures.
- Ensured thorough documentation of integration processes, configurations, and troubleshooting guides.
- Maintained compliance with relevant standards, regulations, and best practices.
- Fostered cross-functional collaboration with product management, sales, and customer support teams to align integration efforts with company initiatives.

Unified Application Submission System:

- Designed and implemented a unified application submission system using Python, Django, ReactJS, and Next.js, simplifying the process for brokers and ensuring accuracy in data collection.
- Developed data segregation algorithms to route answers to the appropriate insurance carriers, thereby improving submission accuracy and efficiency.
- Utilized AWS Lambda and FastAPI to handle webhook events from carrier's web servers, and store them in DynamoDB to ensure that data is not lost. This will later be used by a downstream process, in a polling fashion.
- Wrote unit tests using the Django testing framework to ensure the quality and reliability of the application.
- Wrote integration tests between Limit's platform and multiple carrier's test platforms to ensure that we are aware of any ongoing changes, and addressed them immediately to not disrupt the consuming partners that are using our test environment to test their integration with ours.
- Implemented GitHub Actions to check code quality, run unit tests, perform integration tests, and send release reports.

Admin Dashboard:

- Created an admin dashboard and various utility applications for retail brokers using Django Admin.
- Developed various Django actions integrated with Limit AI to ease the process of drafting emails, reviewing deficiencies, and other tasks for retail brokers.
- Integrated with Front App to generate tickets for client follow-ups.

PR Review Reminder and Scorer:

- Developed a Python script to remind developers to review PRs and encourage timely reviews.
- The script integrates with Github API and Discord API, hosted as a Lambda function on AWS and scheduled using AWS Event Bridge.
- The message posted on Discord tags each engineer and the PRs they need to review. Additionally, assigned a score to each engineer based on the quality of the reviews done the previous day.

PR Environments:

- Implemented a pipeline for creating pull request (PR) environments to enable thorough system testing before merging code into the master branch.
- Utilized GitHub Actions and Terraform to create these environments based on specific patterns in the branch name or the PR description.
- Made the environments configurable directly from the PR description using XML tags.
- Ensured that the environments are destroyed immediately after the PR is merged or one hour after their creation.
- Implemented the ability to use production, staging, or development database replicas as needed.

Retool Components:

- Built various components in Retool to view all the data related to individual database entities at a single location. For example, in a relational database, the data entities are scattered across tables. Given a data entity, the Retool component runs an SQL query to fetch all the related data and displays it in a single place.
- Designed complex SQL queries and optimized them for performance.

Others:

- Integrated with Sentry for error tracking.
- Set up a Bastion Server to manage access to the platform's admin dashboard for brokers and servers for developers.
- Set up a deployment pipeline to pull code, set up the test environment, run unit tests, build Docker images, push to ECR, configure or update the environment, deploy code using blue-green deployment, run integration tests, and send detailed release reports to downstream teams.
- Wrote Terraform scripts to manage infrastructure across all environments: development, staging, and production.
- Deployed and managed applications on Amazon EKS, ensuring high availability and scalability of microservices.
- Created comprehensive documentation for processes, configurations, and codebases.
- Gained exposure to the insurance industry.

Environment: Integration Solutions, Data Exchange, Python, Django, Next.js, Unified Application Submission System, Data Segregation Algorithms, Market Partner Collaboration, REST APIs, Scalable Systems, Admin Dashboard Development, AWS Lambda, SQS, Asynchronous Processing, Front Application Integration, Unit Testing, Django Testing Framework, GitHub Actions, Code Quality, Integration Tests, Release Reports, PR Review Scripts, Bastion Server Setup, Real-Time Communication, Websockets, Terraform, Infrastructure Management, Deployment Pipeline, Docker, ECR, AWS EventBridge, Notifications, Cron Jobs, System Maintenance, Health Checks, Automation Tasks, Retool, Internal Tools, Data Analysis, Documentation, Training, Cloud Infrastructure, Spark Streaming.

Factset, Hyderabad, India **Jul 2020 — Jun 2022**
Role: Software Developer

Factset is a financial data and software company that provides integrated financial data and software solutions to investment professionals. The company's offerings include comprehensive analytics, real-time and historical data, and research management tools, enhancing portfolio management and investment decision-making.

Key Achievements:

- Implemented data ingestion pipelines processing over 50,000+ documents per day, ensuring robust validation and categorization.
- Enhanced the storage and retrieval of over 10 TB of diverse data, ensuring efficient querying capabilities for downstream applications, leading to a 30% improvement in data access speeds.
- Optimized the Data Lake architecture and reduced costs by 30%.

Data Lake:

- I played a pivotal role in managing the storage and retrieval of diverse data types within the Data Lake infrastructure. Our focus is on handling feeds that supply various categories of documents, including PDFs, HTML files, CSVs, and JSONs. The workflow involves processing these documents to extract granular information, storing it in optimized S3 buckets, and providing efficient querying capabilities for downstream applications.
- Implemented data ingestion pipelines to listen for document arrivals from various feeds, running validations before ingesting into Data Lake.
- Developed a Lambda process to categorize incoming documents (HTML, PDFs, CSVs, JSONs, and XMLs) and direct them to specific SQS queues.
- Implemented and deployed ECS processes for efficient document processing, extracting granular information such as words, paragraphs, tables, headers, and titles from HTM and PDF documents.
- Utilized AWS Autoscaling to ensure that enough ECS processes are up and running to cater to the unprecedented spike in documents arrival.
- Designed a storage strategy that prioritized performance, ensuring the efficient storage and retrieval of extracted information in S3 buckets.
- Developed a custom query engine for querying documents for words, tables, paragraphs, titles, headings, and data related to specific categories - metadata.
- Developed a Python tool called as Carbon Query, for reading metadata already stored in the Data Lake and also for ingesting additional metadata into Data Lake -- enabling Machine Learning and Data Collecion teams to interact with the Data Lake.
- Configured and managed ElasticSearch, Kibana, and LogHive stack to establish a resilient logging and log search functionality.
- Implemented monitoring mechanisms to oversee the health and performance of diverse microservices within the Data Lake platform.

Raw Data Classifier:

- Implemented processes to classify raw data extracted from various documents into words, paragraphs, titles, headings, tables, rows and cells.
- Raw data is not just classified here. The processed data will contain all the information related to each entity – the exact location in the document, length of the text, language and it identifier.
- Employed BeautifulSoup for processing HTML documents and pdfkit for PDFs.
- This data is stored in the Data Lake in the respective S3 buckets – which will further be used by Machine Learning teams to generate Data catalogs or run analysis.

Central Logging Platform:

- Implemented a Central Logging Platform (CLP) using Kibana and AWS Kinesis Streams to enable Data Lake developers and downstream teams see all the logs related to processing of various kinds of documents.
- To achieve this, we have implemented two packages – LogHive and LogBee.
- Log”Bee” is the client package used by all the document processing systems, whereas Log”Hive” is a consumer which consumes all the logs coming from Log”Bee”s in the form of streams via Kinesis Stream.
- This has been really helpful in tracing any issues occurring in the pipelines – data ingestion and data processing.

Secret Vault:

- Developed a Python package that enables storing of secret keys alongside source code in an encrypted format.
- Utilized AWS Key Manager Service (KMS) to encrypt and decrypt the keys are required.
- This enabled developers to modify/use not-so-important secrets alongside their code as needed.

Data Collector (Thunder Viewer):

- A UI tool that allows users to collect data from HTML and PDF documents.
- They can collect words, paragraphs, tables, titles or headings from these documents and tag them. Collection is done by highlighting the portion of data they’d like to collect and clicking on a button.
- Each entity in the collected data can be traced back to its source document using position, length and an identifier.
- They can either use the collected data for further processing or simply tag the data and store it back in the Data Lake as metadata.
- I led the end-to-end design, development, and deployment of this real-time web application.
- Employed a mechanism to retrace the user’s steps in the process of data collection, if an error is encountered in the process.
- Developed the application using FastAPI as the backend and server-side rendered Jinja templates for frontend.
- Implemented various middlewares for authentication, logging and notifications in the FastAPI application.
- Utilized DynamoDB for efficient storage of session-related information, encompassing details about user sessions, opened documents, and highlighted source/metadata. Adopted a NoSQL single table design for streamlined data management.

- Enabled users to open sessions, highlight text, and empowered external applications to connect, listen to highlights, and process data collaboratively.
- Employed Redis as a Pub/Sub service, ensuring effective WebSocket communication within the application.
- Orchestrated the deployment of the application on AWS ECS using Terraform as Infrastructure as Code (IaC) and automated the deployment pipeline with Github and AWS.
- Contributed to ongoing performance improvements through enhancements and, when necessary, architectural changes.

Others:

- Experience using S3-Select, Athena and Glue Job to query and process data.
- Employed AWS Batch to process bulk data on request-basis from various clients.
- Deployed containerized applications using Docker, orchestrated with AWS ECR & ECS, integrated with CI/CD pipelines using CodeBuild and Github Actions.
- Utilized AWS Elastic Load Balancer (ELB) along with Elastic Container Service (ECS) to ensure that we are able to handle unprecedented peak loads.
- Deployed the applications in multiple availability zones (AZs) to ensure reliability and disaster proof. Also, deployed few process that are very critical for the company across multiple regions.
- Wrote Terraform modules for reusability across various projects within the Data Lake Team.
- Also, imported existing infrastructure under terraform management.
- Employed SonarQube for continuous inspection of code quality.
- Ran cost analysis between Apache Airflow and AWS Step Functions, and decided that Airflow is much cheaper.
- Utilized Apache Airflow for data workflow automation.

Environment: Python, PySpark, AWS (S3, EC2, Glue, Athena, EMR, ElasticCache, ECS, ELB, EKS, IAM, KMS, CloudWatch, CodeBuild, Kinesis, DynamoDB, StepFunctions), Apache Airflow, Data Ingestion Pipelines, BeautifulSoup, Docker, CI/CD pipelines, CodeBuild, GitHub Actions, Elasticsearch, Kibana, LogHive, Data Modelling, schema design, query engines, performance tuning, optimization, HTML parsing, Terraform, SQL, PostgreSQL, MySQL, NoSQL, ORM tools, Agile Scrum, Flask, FastAPI, middleware, JWT, SQLAlchemy, Terraform.

Cognerium Robotic Labs, Hyderabad, India **Apr 2016 — Jul 2020**
Role: Software Developer

Cognerium is an AI powered wealth growth platform for Advisors and Sales Leaders.

Key Achievements:

- Created test cases achieving a 95% test coverage and significantly reducing defects across various
- applications.
- Utilized ORM techniques and managed transactions, enhancing database operations and reducing query
- execution time by 25%.

Platform as a Service (BankplusOne):

- The platform serves as a master application of Cognerium, designed to efficiently manage client access across multiple Bankplusone applications through Single Sign-On (SSO).
- Implemented a comprehensive user roles and access management system, empowering administrators to create users, assign access, and control permissions. Also ensured users could only view features of applications to which they were granted access.
- Successfully transformed the entire application in response to frequently changing requirements, demonstrating adaptability and agility.
- Deployed a server with Prerender.io to render Angular pages, improving Search Engine Optimization (SEO).
- Handled the sitemap to enhance the platform's visibility on search engines.
- Involved in the implementation of RESTful API services using Django-rest-framework, UI development for the website bankplusone.com, and various enhancements to ensure availability, scalability, and reliability.

Wealth Growth:

- Wealth Growth, an integral component of the Bankplusone application stack, is a specialized module within Wealth Management. This module focuses on grouping prospective leads and evaluating them against multiple wealth products. The workflow involves the Marketing Manager qualifying leads, assigning them to the Advisor Manager, who then scores the leads against available advisors. Advisors, in turn, conduct events, potentially making sales and recording the Assets Under Management (AUM) acquired. The project encompasses user roles and access management, integration with other applications within the Bankplusone Stack, and the incorporation of machine learning models for refining prospect data.

CECL – Current Expected Credit Loss

- CECL represents a new credit loss accounting standard, and my role in the project involved collaborating with Data Scientists to implement interfaces for their machine learning models. This initiative required adept handling of substantial data both on the frontend and backend, utilizing chunk-based processing to mitigate memory limitations on the server.

Environment: Business requirements gathering, System Impact Analysis (SIA), functionality documentation, High-Level Design (HLD), Low-Level Design (LLD), Microsoft Visio, use case development, activity diagrams, sequence diagrams, RESTful APIs, Java, Spring Boot, Jersey, Python, Django, SOAP web services, RESTful web services, Python scripting, automation, structural design patterns, MVC, Façade, Front Controller, Spring Security, Java 8 features, streams, lambda expressions, Hibernate ORM, transaction management, SQL queries, stored procedures, MySQL, JUnit, integration testing, Spring Test, Mockito, AngularJS, Custom Directives, Services, Controllers, real-time communication, user roles, access management, RxJS, reactive programming, machine learning models, SEO optimization, Prerender.io, application configuration, Quality Assurance collaboration, bug fixing, Nginx, build automation scripts, continuous integration, Jenkins, migration strategies, transformation rules, validation processes, ETL/ELT, SAP HANA, data consolidation, data cleaning, deduplication rules, ETL jobs, cutover plans, purchase order management, invoice management, accounts receivable data migration, data migration strategy, Databricks, Delta Lake, Databricks Notebooks, clusters, workspace management, AWS services, S3, PySpark, data pipelines, Infrastructure as Code (IaC), Terraform, AWS Batch, data validation, Retool, documentation, training sessions, cloud infrastructure provisioning, cloud infrastructure management.