Amala B

United States | (678) 674-4437 | amalaber4040@gmail.com | LinkedIn | Portfolio

PROFESSIONAL SUMMARY

Senior Python Developer with 10+ years of experience building scalable web applications, microservices, and data pipelines across industries like finance, healthcare, and telecom. Skilled in Django, Flask, RESTful and GraphQL APIs, and front-end tools like ReactJS and AngularJS. Strong cloud expertise (AWS, Azure, GCP), data engineering with PySpark and Airflow, and CI/CD automation using Jenkins, Terraform, and Docker. Experienced in ML/NLP model deployment, system monitoring with Datadog, and ensuring security compliance (e.g., HIPAA).

TECHNICAL SKILLS

Programming	Python, C/C++, Java, SQL, JavaScript
Python Frameworks	FastAPI, Django, Flask, Pyramid, web2py, SQLAlchemy (ORM)
Web Technologies	HTML5/CSS3, JavaScript, jQuery/AJAX, JSON/XML, XPath, Debugging
Python Libraries	NumPy, Pandas, SciPy, scikit-learn, PyTorch
Version Control	Git, GitHub, Subversion
DevOps & Cloud	Docker, Kubernetes, Terraform; AWS (S3), Azure, GCP
ML/AI	NLP/Chatbots; Hugging Face, MLflow, LangChain (RAG), TensorFlow
Testing Tools	unittest, pytest, RESTful API
Database tools	MySQL, Oracle (PL/SQL), Microsoft SQL Server, PostgreSQL, MongoDB
Data Eng & Streaming	Airflow, Spark/PySpark, Kafka

PROFESSIONAL EXPERIENCE

Sr. Python Developer
Humana (New York, USA) **July 2023 – Present**

- Developed scalable and production-grade RESTful APIs using FastAPI and Python, delivering low-latency services to internal systems.
- Followed Object-Oriented Programming (OOPs) principles to build modular, maintainable, and testable backend services.
- Built microservices architecture with FastAPI and Docker, orchestrated through Kubernetes for better scalability.
- Designed ML pipelines with ML flow Projects to ensure reproducible experiment execution across environments.
- Designed asynchronous endpoints using FastAPI's async/await to handle concurrent I/O operations efficiently.
- Created comprehensive unit, integration, and end-to-end test cases for APIs using Pytest and FastAPI TestClient.
- Utilized BitBucket and GitHub for source control, branching strategy, pull requests, and code reviews.
- Built Airflow DAGs to orchestrate ingestion of clinical, insurance, and patient feedback data into a unified data lake.
- Integrated GraphQL APIs with Python-based backends FastAPI enabling optimized client-server communication and reducing over-fetching and under-fetching of data.
- Developed secure Flask-based web applications for internal hospital use, enabling access to patient summaries, test results, and appointment scheduling.
- Build and deploy scalable, event-driven AWS Lambda functions to support microservices architecture and automate backend processes.
- Designed and normalized PostgreSQL schemas to store patient records, medications, allergies, and care plans with ACID compliance.
- Developed scalable PySpark ETL pipelines to process millions of EMR records from distributed file systems and transform them into analytics-ready datasets.
- Designed end-to-end Retrieval-Augmented Generation (RAG) pipelines using LangChain with vector stores and LLMs.
- Built near real-time search functionality using Elasticsearch integrated with FastAPI services.

- Integrated PySpark with Hive to support large-scale healthcare reporting across hospital networks, improving query performance by 40%.
- Created real-time monitoring dashboards with Kibana and Grafana for service health and performance insights.
- Designed automated health checks, metrics collection, and alerts using Prometheus and Datadog.
- Managed CI/CD workflows for multiple environments, enabling seamless staging and production rollouts.
- Wrote and maintained Terraform scripts for deploying cloud infrastructure supporting FastAPI services.
- Integrated Flask applications with EMR systems using HL7/FHIR APIs, facilitating seamless data exchange between clinical platforms.
- Enabled row-level security in PostgreSQL for patient data, granting fine-grained access to authorized care providers only.
- Collaborated with DevOps to containerize services using Docker and deploy them in AWS ECS/EKS.
- Developed custom resolvers and middleware to handle business logic, authorization checks, and request/response transformations within the GraphQL layer.
- Implemented OAuth2 and JWT-based authentication for secure API access and multi-tenant support.
- Built workflow automation systems for onboarding processes using FastAPI, Celery, and RabbitMQ.
- Orchestrated background tasks and retry mechanisms using Celery workers with Redis as the broker.
- Configured task-level retry policies and SLAs in Airflow to ensure timely execution of sensitive hospital reporting workflows.
- Designed user-friendly interfaces using Flask templates for doctors to filter and download patient analytics reports.
- Enabled data sync automation between MongoDB, Elasticsearch, and cloud data lake using API hooks.
- Used PySpark UDFs and DataFrames to calculate patient comorbidity indexes based on multiple diagnosis codes.
- Maintained developer-friendly API documentation using FastAPI's built-in Swagger UI and Redoc.
- Integrated AI/ML model endpoints exposed through FastAPI, including LLM and BERT model inference APIs.
- Worked closely with ML engineers to create MLOps pipelines for model serving, retraining, and monitoring.
- Optimized the load time and memory consumption of NLP model endpoints exposed via FastAPI.
- Participated in Agile ceremonies, including sprint planning, retrospectives, and backlog grooming with cross-functional teams.
- Used Airflow's XCom to track lineage of patient-level data through multi-step ETL processes.
- Developed robust application logic using Python, JavaScript, and Java to ensure scalable and reliable solutions.
- Created clear and user-friendly API documentation with Swagger/OpenAPI, simplifying interactions with RESTful services.
- Used Postman to thoroughly test and validate RESTful APIs, ensuring they perform reliably in various scenarios.
- Implement RFID solutions to track patient movements within hospitals.
- Build RESTful and gRPC-based microservices using FastAPI to handle modular business logic.
- Built and optimized data pipelines using Golang and Python, enhancing performance and data processing efficiency.
- Designed and implemented RESTful APIs with FastAPI to support diverse application functionalities.
- Integrated GPT-4 into electronic health record (EHR) systems to generate patient care recommendations and support clinical decision-making processes.
- Managed version control for Python projects using Bitbucket Git repositories, maintaining clean and consistent commit history.
- Write and optimize Lambda functions using Python, Node.js, or other supported languages ensuring high performance and low latency.
- Work closely with data governance teams to align data quality implementations with enterprise data standards.
- Develop and fine-tune LLMs using Python frameworks like Hugging Face Transformers, PyTorch, or TensorFlow for domain-specific applications.
- Enable fast patient identification during emergencies using RFID bands.
- Designed and implemented real-time data streaming pipelines using Amazon Kinesis for high-throughput ingestion and processing.
- Designed and automated Grafana dashboards to visualize real-time application metrics, ensuring critical insights for system monitoring and performance optimization.
- Created and maintained feature branches, release branches, and hotfix workflows using Git branching strategies.
- Use RFID to monitor the usage and location of medical assets.

- Designed DynamoDB schemas to optimize GraphQL queries and mutations for speed and scalability.
- Designed and implemented infrastructure as code (IaC) solutions using tools like Terraform, CloudFormation, or Ansible, integrated with Python workflows.
- Designed and implemented responsive and interactive user interfaces using React.js, ensuring seamless user experiences across devices.
- Developing and optimizing Spark applications using PySpark 3.x for large-scale data processing and analytics.
- Customize and integrate Informatica Data Quality tools with existing systems to automate data validation and profiling.
- Develop and integrate Python-based AI/ML models into Palantir Foundry to enable scalable, data-driven decision-making.
- Led the design and implementation of efficient ETL pipelines using Python, ensuring smooth data migration and transformation across various systems.
- Created scalable GraphQL endpoints that supported versioning and modular design principles, improving maintainability of microservices and large applications.
- Implemented and monitored Bitbucket Pipelines to automate CI/CD for Python microservices.
- Developed and deployed end-to-end machine learning models using Python libraries such as scikit-learn, TensorFlow, and PyTorch.
- Designed and maintained relational database schemas and queries in PostgreSQL and MySQL to support scalable application development.

Environment: Python, Java, ETL,IDQ, Graph QL ,Lambda, Grafana, LLM, JavaScript, NoSQL, Golang, FastAPI, Flask, Django, AngularJS, React, Swagger/OpenAPI, Postman, PostgreSQL, MongoDB, Snowflake, Oracle, PySpark, Pandas, NumPy, TensorFlow, PyTorch, Apache Kafka, Spark, Jenkins, Git, Docker, Kubernetes, Datadog, RHEL, Terraform, Ansible, Redis, GPT-4, NLTK, spaCy, Rasa, matplotlib, and Snowflake SQL.

Sr. Python Developer
Deutsche Bank (Wilmington, DL) **Jan 2019- April 2022**

- Designed and developed scalable RESTful APIs using FastAPI, ensuring low-latency and high-throughput architecture for internal applications.
- Implemented robust exception handling and data validation mechanisms using Pydantic in FastAPI-based microservices.
- Integrated MongoDB, Redis, and Elasticsearch for different storage and caching requirements, enabling fast retrieval and full-text search capabilities.
- Developed and maintained CI/CD pipelines using Bitbucket pipelines and GitHub Actions to automate testing and deployments.
- Maintained high code quality standards through automated unit testing (PyTest), code linting (flake8, black), and code coverage monitoring.
- Built real-time dashboards with Kibana to monitor API health, response time, and error rates, reducing mean time to detect (MTTD) issues.
- Utilized GraphQL Subscriptions for real-time data updates using WebSockets, allowing live data synchronization between client and server.
- Led the migration from legacy Flask services to FastAPI, reducing technical debt and improving scalability.
- Participated in code reviews, sprint planning, and backlog grooming, ensuring Agile best practices and delivery of high-quality software.
- Configure secure IAM roles and policies for Lambda functions to ensure proper access control and compliance.
- Contributed to building a workflow automation engine, triggering event-based tasks and asynchronous pipelines with Celery and Redis.
- Designed and implemented API versioning and documentation using Swagger (OpenAPI), improving collaboration across cross-functional teams.
- Collaborated closely with DevOps to implement containerized deployments using Docker and Kubernetes for all services.
- Applied data indexing and sharding strategies in MongoDB and Elasticsearch to improve the scalability of backend services.
- Developed unit and integration test suites covering edge cases and business logic, achieving over 90% code coverage.
- Contributed to internal research on LLM and BERT model usage, exploring opportunities to integrate NLP into backend services (PoC level).
- Mentored junior developers on clean code practices, version control strategies, and API design principles.

- Orchestrated the architecture and development of the backend using Python and Django, while implementing a React, Webpack, Redux, and PostgreSQL stack for the frontend, ensuring a seamless full-stack application.
- Conduct experiments and tests to validate AI models and their performance in real-world scenarios.
- Participated in code reviews and team meetings to ensure code quality and adherence to best practices in Python and C++ development.
- Connected GraphQL APIs to SQL and NoSQL databases (PostgreSQL, MongoDB, etc.) through ORMs such as SQLAlchemy or Django ORM, ensuring data consistency and performance.
- Implemented GraphQL query optimization techniques, including data batching, caching, and lazy loading with DataLoader, reducing API response times and server load.
- Use AWS CloudWatch for real-time logging, metrics collection, and alarms to monitor Lambda function health and performance.
- Developed and maintained Python scripts and applications with PyCharm, utilizing its refactoring tools to optimize code structure and performance.
- Optimize PySpark jobs by fine-tuning Spark configurations, partitioning strategies, and caching.
- Implemented data validation and error handling mechanisms in Python to ensure the integrity and accuracy of data stored in Sybase databases.
- Develop interactive data visualizations and dashboards in Palantir Foundry using Python to present AI outputs.
- Migrated legacy data from Oracle databases to modern NoSQL solutions like MongoDB using custom Python scripts.
- Created Python scripts for device provisioning and firmware updates, ensuring efficient deployment and maintenance of IoT networks.
- Collaborated with distributed teams using pull requests, inline comments, and task tracking in Bitbucket.
- Collaborated with front office traders, finance, and risk teams to gather requirements and deliver scalable solutions aligned with regulatory and compliance objectives.
- Developed custom Lambda functions to handle API authentication and enforce security policies.
- Integrated Python-based monitoring scripts with Grafana to collect, process, and visualize custom metrics tailored to application performance needs.
- Automated deployment of FastAPI/Django apps using Bitbucket Pipelines with Docker and AWS ECS.
- Designed automated backup workflows for IaaS environments, ensuring high availability and disaster recovery using Python.
- Integrate LLMs into business workflows, APIs, or applications for tasks like chatbots, content creation, or decision support.
- Analyze and process real-time market data to generate actionable insights for investment strategies.
- Worked closely with quant teams and the risk engine team to integrate quantitative models into the analytics platform for derivatives pricing and risk evaluation.
- Automate deployment of Lambda functions using AWS Code Pipeline, Code Build, or tools like Serverless Framework and SAM.
- Used Bitbucket Tags and Releases to manage versioned deployments for staging and production environments.
- Designed and developed RESTful APIs in Python using Django Rest Framework, ensuring compatibility with TypeScript-based front-end applications.
- Implemented real-time monitoring and alerting using Datadog to detect and respond to issues in the system promptly.
- Utilized TypeScript to build interactive and responsive user interfaces for web applications, ensuring a seamless user experience.
- Created custom ETL scripts in Python to automate complex data extraction, transformation, and loading tasks, enhancing overall workflow efficiency.
- Collaborated with data scientists and analysts to implement machine learning models and algorithms in Python, leveraging Palantir's data analytics capabilities to derive actionable insights and predictions.
- Designed and implemented RESTful APIs using Python within Palantir Foundry, facilitating seamless data exchange and integration with external systems.

Environment: Python, ETL, Django, Lambda, Pyspark,IDQ, NoSQL, Django Rest Framework, Flask, FastAPI, React, Redux, Webpack, TypeScript, PostgreSQL, Sybase, Palantir Foundry, GCP, Kafka, Redis, Memcached, .NET, PyTorch, Pandas, NumPy, Scikit-learn, pytest, Terraform, Jenkins, GitLab CI, CircleCI, Prometheus, Grafana, Fluentd, Vagrant, Kubernetes, Docker, SQL, Oracle, PL/SQL, GraphQL, HTML, BDD, and Linux systems.

Python Developer
Comcast (Philadelphia, PA) **June 2016- Dec 2018**

- Spearheaded the design and implementation of RESTful APIs, encompassing database design and caching strategies to ensure optimal performance.
- Assisting in the migration and refactoring of Perl code to other languages, such as Python, to modernize legacy systems and align with strategic objectives.
- Develop scripts and processes to identify and resolve data anomalies, ensuring high levels of data accuracy and integrity.
- Conducted research on state-of-the-art deep learning methods and implemented innovative solutions using PyTorch to address complex problems.
- Deployed Airflow on Kubernetes to scale job execution across telecom data centers.
- Engineered scalable and robust distributed systems using Python, JavaScript, and NodeJS, deployed through a Kubernetes controller on the AWS public cloud.
- Contributed to the development of open-source PyTorch libraries and tools, enhancing the capabilities and usability of the framework.
- Engineered PostgreSQL schemas for high-throughput storage of tower metrics, SMS logs, and call records with horizontal scalability.
- Used Spark MLlib in PySpark to build churn prediction models based on service usage patterns and complaints.
- Manage versions and aliases of Lambda functions to support blue-green deployments and safe rollbacks.
- Developed new features for RESTful APIs, implementing OAuth for secure authorization and utilizing Celery and Redis for reliable queue-based workflows in backend systems supporting telecommunication platforms.
- Contributed to web application development using Flask and Django, ensuring scalable and maintainable solutions for telecom services.
- Used Airflow sensors to wait for new call log files before initiating ETL tasks to ensure data consistency.
- Implement monitoring tools to evaluate LLM performance and accuracy, ensuring continuous improvements in outputs.
- Enhanced application security by integrating OKTA for authentication and enabling SSL to secure data in transit.
- Built and maintained Docker container clusters managed by Kubernetes on GCP, ensuring efficient deployment of telecom services.
- Consumed and managed API data in React applications by fetching and processing JSON responses from Python/Django or Flask APIs.
- Integrated PostgreSQL with ETL tools and APIs for real-time telecom data warehousing and downstream processing.
- Integrated FastAPI services with real-time Kafka event streams to notify customers about dropped calls or poor signal coverage.
- Created analytics dashboards in Flask for regional managers to track service disruptions and outage reports.
- Used Celery with Flask to process long-running jobs like billing reconciliation and customer churn notifications asynchronously.
- Refactored batch jobs and migrated legacy data extracts from Informatica to Python-based microservices, achieving minimal downtime during deployment.
- Migrated data warehouses from Redshift to Snowflake to achieve centralized, scalable, and optimized solutions for telecom data analytics.
- Collaborated with frontend teams (React, Angular) to define API contracts via GraphQL, ensuring schema changes were backward-compatible and well-documented using tools like GraphiQL or Apollo Studio.
- Monitored and secured GraphQL APIs by implementing rate limiting, depth limiting, and query complexity analysis to prevent malicious usage and ensure production-grade stability.
- Deployed microservices on AWS ECS and EC2 instances to improve modularity and maintainability in telecommunication systems.
- Created reusable PySpark components to analyze signal degradation trends across cell towers.
- Automated deployment processes with Python scripts using the Boto module to interact with AWS resources, reducing manual intervention in telecom operations.
- Leverage Glue Dynamic Frames for advanced data transformations and data format handling.
- Created ETL pipelines for telecom data management using Apache Airflow in serverless mode and integrated with Redshift DB for scheduling.
- Performed OAuth API testing using Postman, Python, and HTML to validate telecom platform documentation and identify errors.
- Developed Docker containers to manage application lifecycles, ensuring consistency across environments for telecom services.

- Designed backend modules with PL/SQL and PostgreSQL stored procedures for telecom data management and analysis.
- Created SOA-compliant web services, registered them with service registries for streamlined communication between telecom systems.
- Designed multi-tenant FastAPI architecture to support multiple telecom clients on a single backend.
- Developed Single Page Applications (SPA) for telecom dashboards using JavaScript frameworks like ReactJS and Angular2.
- Utilized NumPy and Pandas for data manipulation and analysis in telecom data pipelines.
- Built unit and regression test frameworks using PyUnit and PyTest to ensure the reliability of Python applications in telecom environments.
- Automated server infrastructure setups for DevOps services using Ansible and Python scripts, supporting agile telecom application development.
- Developed rigorous unit tests and integration tests using Pytest and Selenium frameworks, adhering to FDA requirements, thereby guaranteeing high-quality and compliant software applications.
- Introduced GitHub Actions as a Continuous Integration (CI/CD) solution, automating the build, test, and continuous deployment processes for a complex software application.
- Designed and implemented ETL (Extract, Transform, Load) pipelines to consolidate P&C data from various sources into a central data warehouse for comprehensive analysis.
- Maintained documentation for AWS Glue workflows, including data dictionaries, job schedules, and dependencies, to facilitate knowledge sharing and ensure the reproducibility of data pipelines.
- Implemented infrastructure as code by developing Terraform scripts for EC2 instances, Elastic Load Balancers, and S3 buckets to enhance deployment efficiency.
- Optimize Glue jobs for performance and cost by configuring execution parameters and partitioning strategies.
- Utilized SQL to design and maintain complex relational databases, ensuring efficient and scalable data management for a large-scale enterprise application.
- Maintained comprehensive technical documentation and training materials covering REST APIs, SQL, React, Angular, and Docker.

Environment: Python, Django, Airflow, Postgresql, FastAPI, Flask, GraphQL, Pyspark, Lambda, Web Framework, ETL, React, HTML, Bootstrap, MongoDB, JavaScript, JSON, Sublime Text, database access, Django Web Framework, NodeJS

Sr. Python Developer
Home Depot (Birmingham, AL) **Dec 2014 - May 2016**

- Utilized Python libraries like NumPy and Pandas for market analysis, incorporating machine learning techniques to enhance data-driven decision-making.
- Collaborated with development teams to optimize Python application performance on EKS, identifying and resolving bottlenecks through profiling, tuning, and resource allocation adjustments.
- Develop and deploy Python-based microservices using Azure Kubernetes Service (AKS) or Azure App Services for inventory tracking, order processing, and personalized recommendations.
- Integrated Airflow with Apache Kafka for processing real-time transaction streams and syncing with downstream analytics tools.
- Implemented full-text search capabilities in PostgreSQL to improve product lookup experiences on e-commerce websites.
- Used Selenium WebDriver API to automate tests on cross browser and cross platform environment. Sound knowledge on Object Oriented Programming concept and Java Language.
- Built customer-facing Flask portals that supported login, browsing, checkout, and post-purchase tracking functionalities.
- Built PySpark models for identifying high-value customers using RFM analysis and fed output into BI tools.
- Implemented front-end solutions for third-party web services, employing jQuery, HTML, AJAX, JSON, and JavaScript to enhance user interactions.
- Integrated PyCharm with virtual environments and package managers like pip and conda, facilitating efficient dependency management and project isolation.
- Integrated CI/CD pipelines using tools like Cloud Build or Jenkins for automated testing, building, and deployment of Python applications on GCP.
- Worked on cross-platform projects using .NET Core and Python, ensuring compatibility and functionality across different operating systems.

- Created admin dashboards using Flask and Jinja2 templates to allow store managers to monitor inventory and campaign performance.
- Tuned PostgreSQL queries using EXPLAIN ANALYZE to optimize reporting performance across multi-million row tables.
- Developed and integrated microservices for retail operations using Azure Logic Apps and Service Bus, streamlining communication across systems.
- Developed dynamic Airflow DAGs that adapt based on changing retail store configurations and pricing rules.
- Developed backup and recovery engine for VM backup/recovery using VMware vSphere APIs, GoLang programming language and RabbitMQ Message bus (communication interface).
- Developed pipeline logic using PySpark's window and join transformations to support personalized retail campaign analytics.
- Implement event-driven architectures using Azure Event Hubs or Service Bus for handling high-volume retail transactions.
- Automated deployment processes using tools like Docker and Kubernetes, ensuring smooth deployment of Python and TypeScript applications.
- Design and implement RFID-enabled self-checkout systems, allowing customers to scan multiple items simultaneously.
- Integrate RFID systems with point-of-sale (POS) applications for seamless transaction processing.
- Used Microservices Docker with Kubernetes interacting through a combination of REST and Apache Kafka message brokers.
- Designed and developed the UI of websites using HTML, XHTML, AJAX, CSS, and JavaScript, ensuring a visually appealing and responsive user experience.

Environment: Python, NumPy/SciPy, Pandas, Azure, PySpark, machine learning, jQuery, HTML, AJAX, JSON, JavaScript, Cherry Py, Django, SQL Alchemy, XHTML, CSS, Docker, golang, AngularJS, Bootstrap, Single Page Applications (SPAs), MySQL, Python Flask, SOAP, XML, webapp2, GUI, CI/CD pipelines, XML Schema.

Software Developer
XPO Logistics (Brooklyn, NY) **Jan 2019- April 2022**

- Collaborated with senior developers to build and maintain logistics applications using Python and the Django framework.
- Assisted in developing backend modules to handle shipment tracking, order processing, and inventory updates.
- Wrote simple, clean Python scripts to automate routine logistics tasks such as data validation, file conversions, and report generation.
- Supported the integration of logistics systems with third-party APIs and tools using REST and SOAP.
- Helped create user-friendly web pages using Django templates along with HTML, CSS, and Bootstrap for responsive design.
- Connected Python applications to MySQL databases to store and retrieve order and warehouse data.
- Participated in basic data analysis using Pandas and NumPy to support delivery performance and route optimization reports.
- Learned and applied version control using Git and GitHub, contributing to code commits and pull requests.
- Supported unit testing and debugging efforts to ensure reliability of features before deployment.
- Documented code, wrote comments, and updated internal wikis to support knowledge sharing within the development team.
- Participated in Agile team ceremonies like daily stand-ups and sprint reviews to stay aligned with project goals.

Environment: Python, Django, HTML, CSS, Node.js, Flex, SOAP, XML, Python SDK, Cassandra, GitHub, My-SQL, JavaScript, Eclipse, Shell Scripting, jQuery, Git, Jira, Windows and LINUX.

EDUCATION

Bachelor of Technology (B. Tech.) **June 2010 - April 2013**
Chaitanya Deemed to be University (India)

Professional Scrum Master I
Scrum.org (Professional Certification) **August 2025**