**Name:** Ganesh Gouru |**Title:** Senior Python Full Stack Engineer | GenAI | Cloud | Devops

**Mobile:** (682) 549-9695 |**Email:** [ganesh.gouru91@gmail.com](mailto:ganesh.gouru91@gmail.com) | **LinkedIn**: [linkedin.com/in/ganeshgouru91](https://www.linkedin.com/in/ganeshgouru91)

A blue and white logo

AI-generated content may be incorrect.A close up of a logo

AI-generated content may be incorrect.**CERTIFICATIONS**

* **PCPP1™ - Certified Professional Python Programmer**
* **AWS Certified Solutions Architect - Associate**
* **Azure Developer Associate**

**PROFESSIONAL SUMMARY**

Over a multi-year span, this candidate has driven comprehensive testing initiatives across API and UI layers using pytest and Playwright, embedding coverage into CI environments managed by GitHub Actions and Jenkins. They implemented robust QA strategies including parametrized tests, mocking, and integrated browser tests, leading to significant functional coverage across asynchronous workflows and service tiers. Their work ensured early validation of production features through automated pipelines across financial systems, contributing to platform reliability in regulated environments.   
  
In addition to testing, the candidate has designed scalable backend systems using Python frameworks like FastAPI and Django within AWS-native platforms, incorporating Kafka, AWS Lambda, ECS, and Step Functions for processing financial data and enabling event-driven architectures. They led platform operations with Terraform-managed infrastructure and CI/CD reliability, while advancing observability using CloudWatch and structured logging. Across financial domains, their attention to secure design, auditing, and compliance ensured strong alignment with industry standards for sensitive workflow processing.  
  
**TECHNICAL SKILLS**

|  |  |
| --- | --- |
| **Programming & Scripting** | Python, SQL |
| **Frontend Libraries & Frameworks** | React, Playwright, Jest, React Testing Library |
| **Backend Frameworks & Libraries** | FastAPI, Flask, Django, Pydantic, SQLAlchemy, Django REST Framework |
| **Cloud Services** | AWS, Lambda, ECS, EKS, S3, RDS, CloudWatch, Step Functions, API Gateway, IAM |
| **DevOps & Infrastructure** | Terraform, GitHub Actions, Jenkins, Docker, VPC, WAF, parameter store |
| **Databases & Messaging** | PostgreSQL, Redis, Kafka, Snowflake |
| **Testing & QA** | pytest, Playwright, Jest, contract tests, mocking, test parametrization, linting, coverage gates |
| **Monitoring & Security** | CloudWatch, Splunk, structured logging, distributed tracing, KMS encryption, RBAC, OAuth2, audit trails, circuit breakers |
| **Development Tools** | Celery, Swagger, OpenAPI, dbt, pgBouncer, feature flags, outage guards |
| **Version Control** | GitHub |
| **Methodologies** | CI/CD, TDD, infrastructure as code, async workflows, microservices |
| **AI/ML** | Pandas, NumPy, scikit-learn, Spark, PySpark, Databricks |

**PROFESIONAL EXPERIENCE**

**CLIENT: Atlantic Health, Remote Dec 2022 – Present**

**ROLE: Lead Python Developer**

**PROJECT SUMMARY**

Led the engineering effort to build a cloud-native platform for real-time care coordination and secure e-prescribing for Atlantic Health. Focused on scalable Python microservices, layered testing strategies, and full-stack observability across AWS. Automated healthcare workflows spanning prescription routing, insurance validation, and audit tracing using Kafka pipelines, Step Functions orchestration, and Terraform-managed infrastructure. Applied CI/CD rigor and strict access controls to meet compliance, with targeted support for GenAI inference routes in select async services.

**RESPONSIBILITES**

* Led the end-to-end testing strategy using pytest and Playwright, embedding GitHub Actions to automate test runs and browser validation across environments.
* Implemented layered QA pipelines with Pytest fixtures, mocking, and test parametrization to reach 90% functional coverage across REST and async workflows.
* Built resilient E2E test harnesses using Playwright, integrated with feature toggles and outage guards to validate frontend-backend integrity in CI.
* Oversaw platform architecture using Python microservices (FastAPI/Django) and PostgreSQL, enforcing RBAC, OAuth, and encrypted audit trails for compliance.
* Designed and deployed scalable backend services using Flask and FastAPI, leveraging async patterns and Redis for real-time notification and context workflows.
* Refactored backend data pipelines and queues using Python and Kafka, introducing Celery workers with retry safety and dead-letter handling for job orchestration.
* Led Terraform-managed infrastructure provisioning across AWS EKS, Lambda, and ECS for containerized services and event-driven API integrations.
* Maintained CI/CD integrity using GitHub Actions across staging and production, enabling automated tests, rollout checks, and multi-stage deployment flows.
* Coordinated MLOps patterns across feature flag services and async inference routes, including structured logging, streaming inference, and fallback handling.
* Deployed Kafka-based data pipelines with PySpark listeners feeding summaries and telemetry into Databricks via modular ETL orchestrations.
* Managed cloud resources with Terraform across EKS, ECS Fargate, Lambda, API Gateway, and S3, embedding CloudWatch for metric-checks and alerting.
* Instrumented observability using CloudWatch logs and metrics across services, applying structured logs, status spans, and distributed tracers for real-time signaling.
* Performed targeted security reviews across auth flows and token scopes, applying KMS-based encryption, throttling, and circuit breakers in high-risk APIs.
* Directed orchestration workflows via AWS Step Functions to sequence retries, enrichments, and state transitions for multi-step operations.
* Acted as technical lead across internal audits, cost optimization, and deployment handoffs for event-driven finance services with Python and AWS.

**ENVIRONMENT**Python, FastAPI, Flask, Django, SQLAlchemy, PostgreSQL, Kafka, Redis, Spark, Databricks, AWS, EKS, ECS, Lambda, API Gateway, S3, Terraform, Step Functions, GitHub Actions, Jenkins, PyTest, Playwright, CloudWatch, KMS

**CLIENT: Fidelity, Mckinney, Tx Dec 2020 – Nov 2022**

**ROLE: Senior Python Developer**

**PROJECT SUMMARY**

Contributed to Fidelity's analytics modernization by developing Python-based backend services and testing frameworks for self-service dashboards and governed reporting APIs. Improved test coverage through Playwright, pytest, and Jest, while managing CI/CD with GitHub Actions and Jenkins. Built data pipelines using Kafka, Spark, and Databricks to unify investment and compliance reporting. Provisioned AWS infrastructure with Terraform, supporting low-latency analytics and secure financial data operations.

**RESPONSIBILITES**

* Automated API and UI test suites using pytest and Playwright within CI/CD pipelines built around GitHub Actions and Jenkins for financial reporting tools.
* Wrote unit and integration tests for React dashboards and workflow UIs using Jest and RTL to increase coverage across dynamic forms and routing contexts.
* Maintained Python backend services with FastAPI and Flask, implementing data validation, caching layers, and async APIs for self-service analytics tools.
* Built backend-for-frontend APIs using Pydantic and SQLAlchemy, streamlining frontend consumption of investment and tracking data via REST endpoints.
* Provisioned EKS clusters with Terraform for backend services, enforcing IAM roles and network policies across multiple AWS environments.
* Oversaw CI/CD processes using GitHub Actions and Jenkins with automated linting, build pipelines, and coverage gates enforced pre-deployment.
* Contributed to real-time event streaming with Kafka, wiring data flows into Snowflake and downstream analytics using Python enrichment workers.
* Tuned ETL jobs on Databricks with Spark transformations and dbt pipelines, supporting schema evolution and lineage tracking for financial operations.
* Defined Terraform modules and parameter stores for AWS-native services (S3, ECS, RDS, CloudWatch), supporting audit dashboards and alert streams.
* Instrumented logs and metrics via CloudWatch across infrastructure and service tiers, adding structured logging for easier incident review and RCA.
* Deployed caching and session resolution via Redis and pgBouncer to enhance concurrency and minimize latency on high-throughput backend APIs.
* Documented endpoints using Swagger/OpenAPI and embedded contract tests to accelerate QA cycles and ensure integration correctness.
* Rebuilt legacy Java services as FastAPI-based Python microservices, standardizing data syncs, metrics, and authentication workflows across systems.
* Guided junior developers on Python testing strategies and CI/CD discipline, improving code reliability and team velocity during crunch periods.
* Delivered features aligned with compliance and observability needs across AWS banking pipelines under fast-paced stakeholder demands.

**ENVIRONMENT**Python, FastAPI, Flask, SQLAlchemy, PostgreSQL, Kafka, Redis, Snowflake, Spark, Databricks, AWS, EKS, ECS, S3, RDS, Terraform, GitHub Actions, Jenkins, PyTest, Playwright, Jest, React Testing Library, CloudWatch

**CLIENT: State of Michigan, Lansing, MI Jul 2017 – Nov 2020**

**ROLE: Full Stack Engineer**

**PROJECT SUMMARY**

Delivered secure backend services and data pipelines to support state financial workflows including payment validation, ledger reconciliation, and audit processing. Built Django-based microservices with RBAC, encrypted audit trails, and OAuth security, integrating Kafka, Redis, and Spark for high-volume transaction processing. Supported financial data ingestion and SLA monitoring via Terraform-managed AWS resources. Led test automation, infrastructure observability, and remediation cycles for audit-aligned banking programs.

**RESPONSIBILITES**

* Built Python backend microservices using Django and DRF to support payment validation, account mapping, and secure API exchange for state financial workflows.
* Refactored batch data loaders in Django and PostgreSQL to streamline payment reconciliations and ledger sync with third-party finance systems.
* Wrote unit and integration tests using Pytest and Jest across API layers, Kafka consumers, and job workers to ensure consistent test coverage and stability.
* Developed testable Django middleware with OAuth2, RBAC policies, and encrypted audit trails to secure financial workflows across departments.
* Deployed containerized services on AWS ECS with Terraform, managing rollout pipelines and audit payload capture across inter-agency network zones.
* Built CI/CD pipelines using Jenkins and GitHub Actions for automated lints, build verifications, and controlled deployments across QA and production tiers.
* Wrote Spark DAGs with row-level validation and PII filtering, supporting secure financial record ingestion from Kafka into Snowflake-backed data marts.
* Ran Celery Beat and time-triggered jobs to orchestrate remittance extracts, failure reruns, and ledger balancing for finance audit programs.
* Provisioned AWS infrastructure with Terraform for finance pipelines, including ECS, RDS, S3, CloudWatch, and tagging strategies aligned to budgets.
* Instrumented CloudWatch and Splunk dashboards for finance processing pipelines, improving detection of bottlenecks and SLA violations.
* Built API tools and bulk importers for financial document intake and metadata publication to legacy adjudication services.
* Used Kafka and Redis to coordinate async payments and notification queues across vendor endpoints, preserving accurate log trails.
* Integrated IAM roles, VPC endpoints, and WAF policies to standardize security for payment APIs and ledger-access workflows.
* Mentored analysts and junior engineers during audit remediation cycles, reviewing log failures and standardizing observability across legacy layers.
* Supported modernization efforts across payment workflows, regulatory feeds, and internal audit streams for state financial platforms.

**ENVIRONMENT**Python, Django, Django REST, PostgreSQL, Kafka, Redis, Spark, Snowflake, AWS, ECS, RDS, S3, Terraform, Jenkins, GitHub Actions, PyTest, Jest, CloudWatch, Splunk, IAM, WAF

**CLIENT: Airtel, Mumbai, India May 2012 – Jul 2015**

**ROLE: Software Engineer**

**RESPONSIBILITIES**

* Developed **Java 7/8** backend services using **Spring MVC** and early **Spring Boot 1.x**, exposing APIs for telecom billing, usage tracking, and CRM modules.
* Built **Flask** automation utilities for internal data sync, log parsing, and light dashboards.
* Designed and tuned **Oracle 11g** and **MySQL** stored procedures, triggers, and views for high-volume subscriber and usage data.
* Integrated **MongoDB** for customer activity logs and **Redis** for caching subscriber sessions.
* Implemented **REST** and **SOAP** APIs for payment gateways, partner integrations, and internal service orchestration.
* Used **RabbitMQ** and **ActiveMQ** for async message exchange across billing and notification services.
* Configured and deployed Java and Flask apps on **AWS EC2**, using **S3**, **RDS**, and **CloudFront** for storage, databases, and content delivery.
* Managed builds and deployments through **Jenkins**, **Maven**, and **SonarQube**, maintaining code quality and automated pipelines.
* Deployed APIs and UIs on **Tomcat**, and **Apache**, optimizing configurations for reliability and performance.
* Built and maintained **AngularJS 1.x** and **Bootstrap** customer service dashboards integrated with backend APIs.
* Automated frontend builds processes using **Grunt** and **Gulp**, streamlining deployments.
* Applied **Jasmine** and **Karma** for frontend unit testing; **JUnit** and **Mockito** for backend validation.
* Developed **Shell** and **Python** scripts for data cleanup, monitoring, and release automation tasks.
* Implemented **Nagios** for system monitoring and alerting in legacy environments.
* Integrated **ELK Stack** for centralized log analytics and dashboarding.
* Used **New Relic** to monitor application performance and optimize slow-running services.
* Conducted **Selenium WebDriver** testing for UI regression and **JMeter** load testing for stress validation before major releases.
* Collaborated with QA and DevOps teams to troubleshoot production issues, reduce downtime, and streamline release cycles.
* Maintained version control in **Git** and **TFS**, following Agile sprint cycles and continuous delivery practices.

**ENVIRORMENT**

Java 7/8, Spring MVC, Spring Boot 1.x, Flask, AngularJS 1.x, Bootstrap, jQuery, Grunt, Gulp, Oracle 11g, MySQL, MongoDB, Redis, RabbitMQ, ActiveMQ, AWS (EC2, S3, RDS, CloudFront), Jenkins, Maven, SonarQube, Nexus, Tomcat, Nginx, Apache, ELK Stack, New Relic, Nagios, Selenium WebDriver, JMeter, Git, TFS, Agile (Scrum)

**EDUCATION DETAILS**

Master’s in computer science, Aug 2015 - May 2017  
Florida Atlantic University Boca Raton

Bachelor’s in computer science June 2008 - Apr 2012  
Parul Institute of Science and technology, India