AB InBev MLOps Challenge

Deploying a Machine Learning Model as an API

Cristhian Guerrero

MLOps Computer Engineer

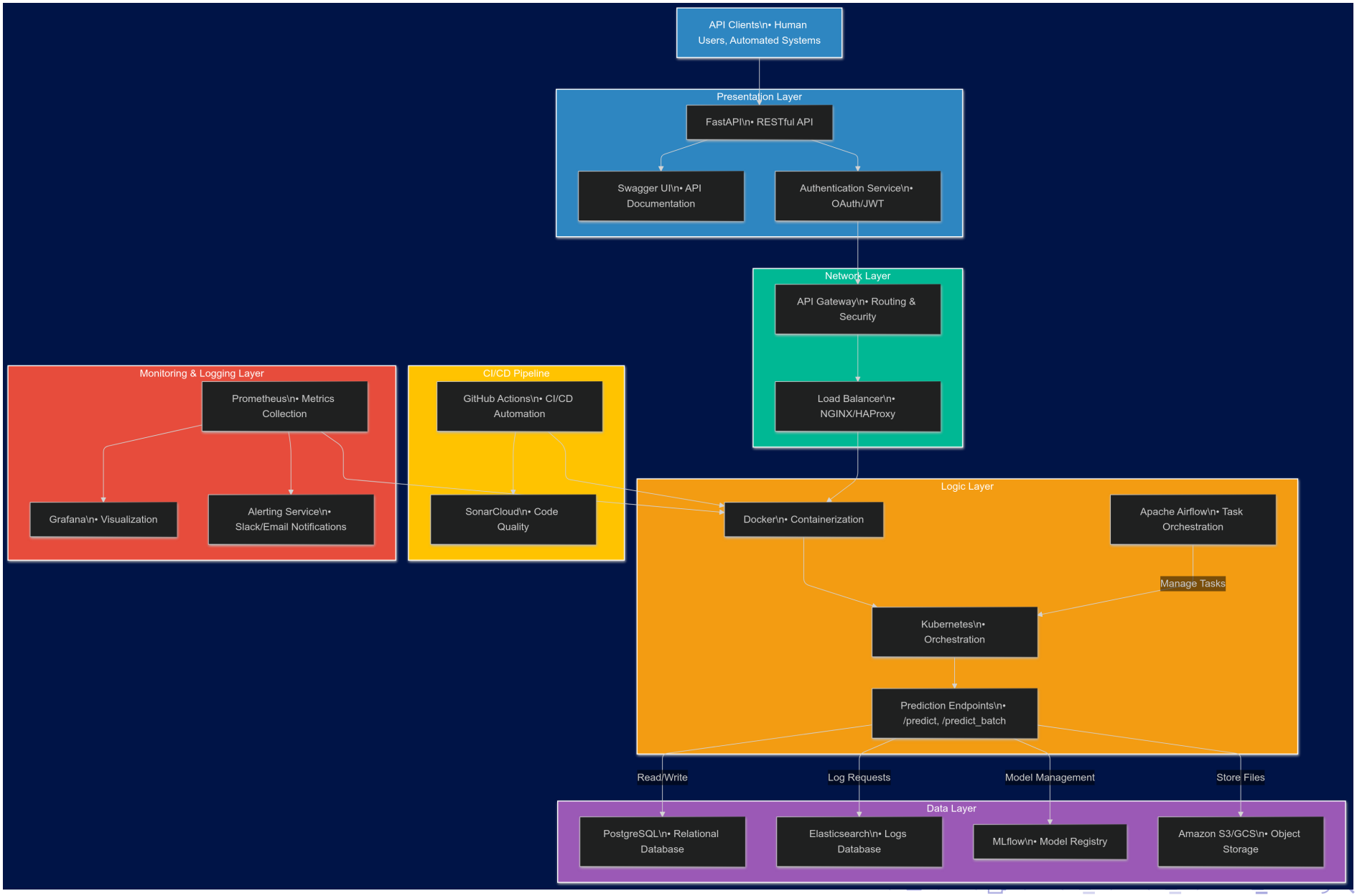
October 17, 2024

Table of Contents

Project Overview

- **Objective:** Deploy a pre-trained machine learning model as a RESTful API.
- **Requirements:**
  - Follow OOP standards in Python.
  - Include inline documentation.
  - Implement batch predictions.
  - Store predictions in a database.
  - Use Docker Compose for local deployment.
  - Run tests and integrate SonarCloud.

System Architecture



Components

- **FastAPI:** Serves the ML model via RESTful API.
- **PostgreSQL:** Stores predictions made by the model.
- **Docker & Docker Compose:** Containerization and orchestration.
- **GitHub Actions & SonarCloud:** CI/CD pipeline and code quality analysis.

Planning

- Analyzed project requirements.
- Designed system architecture.
- Selected appropriate technologies.

Technology Stack

- **Python:** Core programming language.
- **FastAPI:** For building the API.
- **SQLAlchemy:** ORM for database interactions.
- **Docker & Docker Compose:** For containerization.
- **GitHub Actions:** For continuous integration.
- **SonarCloud:** For code quality assurance.

Implementation Steps

1. Set up project structure.
2. Implemented the ML model exposure via FastAPI.
3. Ensured OOP principles by creating classes like `Predictor`.
4. Added inline documentation in English.
5. Enabled batch predictions.
6. Integrated database using SQLAlchemy.
7. Wrote tests and configured SonarCloud.
8. Containerized the application with Docker and Docker Compose.

Challenge 1: Docker Compose Configuration

- **Issue:** Networking and dependency management between services.
- **Solution:**
  - Defined services in `docker-compose.yml` with proper dependencies.
  - Used environment variables for configuration.

Challenge 2: SonarCloud Integration

- **Issue:** Configuring SonarCloud with a private GitHub repository.
- **Solution:**
  - Generated and securely stored SonarCloud tokens.
  - Updated GitHub Actions workflow to include SonarCloud analysis.

Challenge 3: Batch Predictions and Data Integrity

- **Issue:** Handling large volumes of data efficiently.
- **Solution:**
  - Optimized data processing using pandas.
  - Ensured atomic transactions when storing predictions.

Technical Skills Enhanced

- Deepened knowledge of FastAPI and asynchronous programming.
- Improved understanding of Docker networking and orchestration.
- Gained experience with CI/CD pipelines and code quality tools.

Best Practices Adopted

- Implemented clean code principles and OOP standards.
- Used environment variables for configuration management.
- Ensured code readability through inline documentation.

Testing Strategy

- Wrote unit tests for API endpoints using pytest.
- Validated data inputs and outputs.
- Tested database interactions.

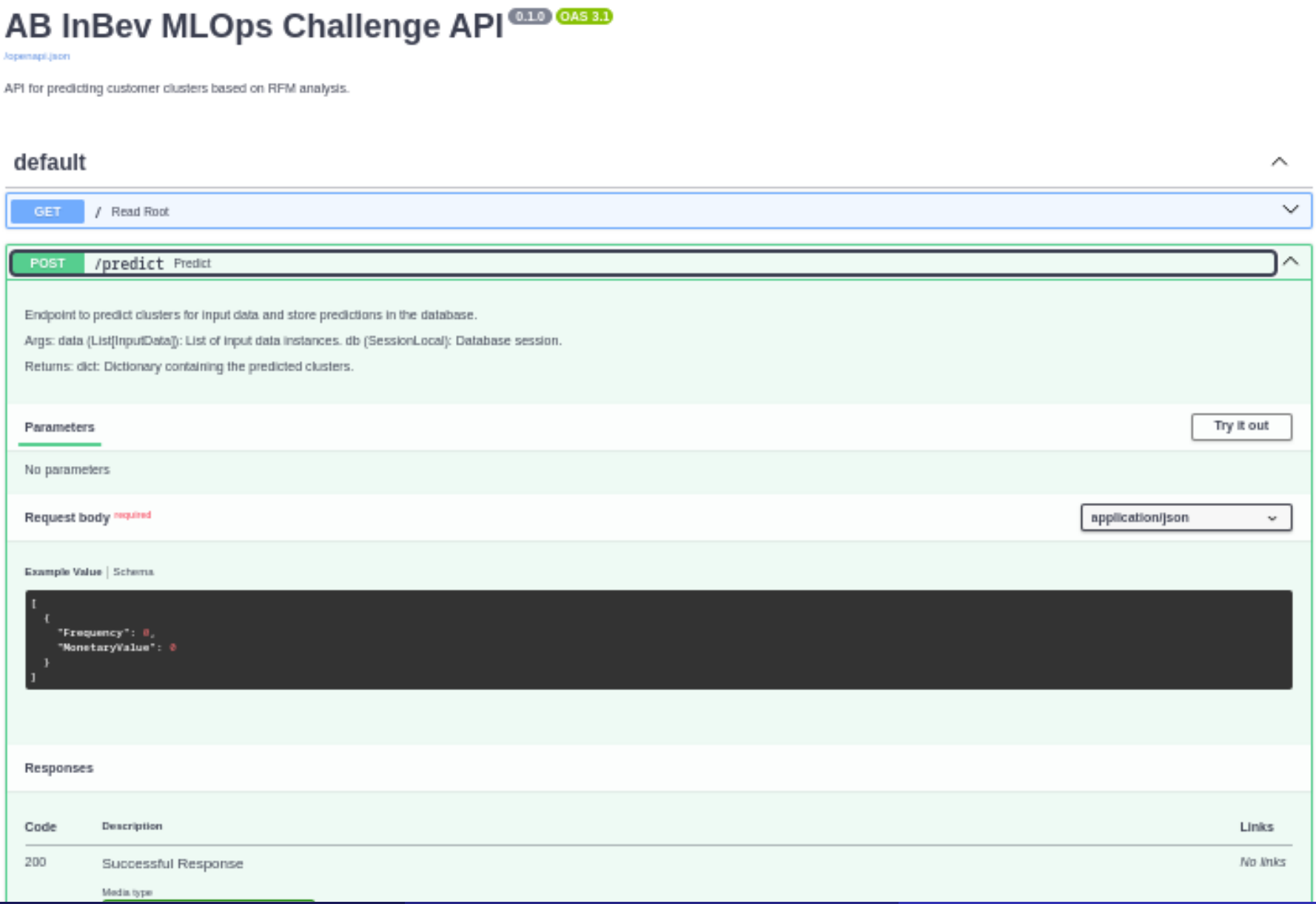
Continuous Integration

- Configured GitHub Actions to run tests on each push.
- Integrated SonarCloud for code analysis.
- Maintained high code coverage and minimal code smells.

Functional Application

- Met all specified requirements.
- Successfully deployed the API locally using Docker Compose.
- Stored predictions in PostgreSQL database.

Demonstration



Summary

- Successfully deployed a machine learning model as an API.
- Implemented best practices in software development.
- Enhanced skills in MLOps and DevOps tools.

Future Improvements

- Deploy the application to a cloud platform.
- Implement authentication and authorization.
- Add monitoring and logging tools.
- Scale the application using Kubernetes.

Get in Touch

- **Author:** Cristhian Guerrero
- **Email:** cristhian3815@gmail.com
- **LinkedIn:** linkedin.com/in/cristhian-guerrero
- **Title:** MLOps Computer Engineer