1. Deployment Process

The Social Media Application is deployed using microservices architecture, with the backend REST API and the frontend static site deployed separately on Google Cloud Run. The PostgreSQL database is hosted on Google Cloud SQL, and an optional API Gateway can be used to route requests between the frontend and backend. This architecture ensures separation of concerns, scalability, and easier maintenance. The frontend communicates with the backend API, which in turn interacts with the Cloud SQL database to process data and user requests.

A diagram of a cloud service

AI-generated content may be incorrect.

The backend is deployed by first building a Docker image from the backend source code and pushing it to Google Container Registry. It is then deployed to Cloud Run with environment variables that configure the database connection and JWT secret.

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

A screenshot of a computer

AI-generated content may be incorrect.

The frontend is built using npm and deployed separately to Cloud Run, allowing it to scale independently.

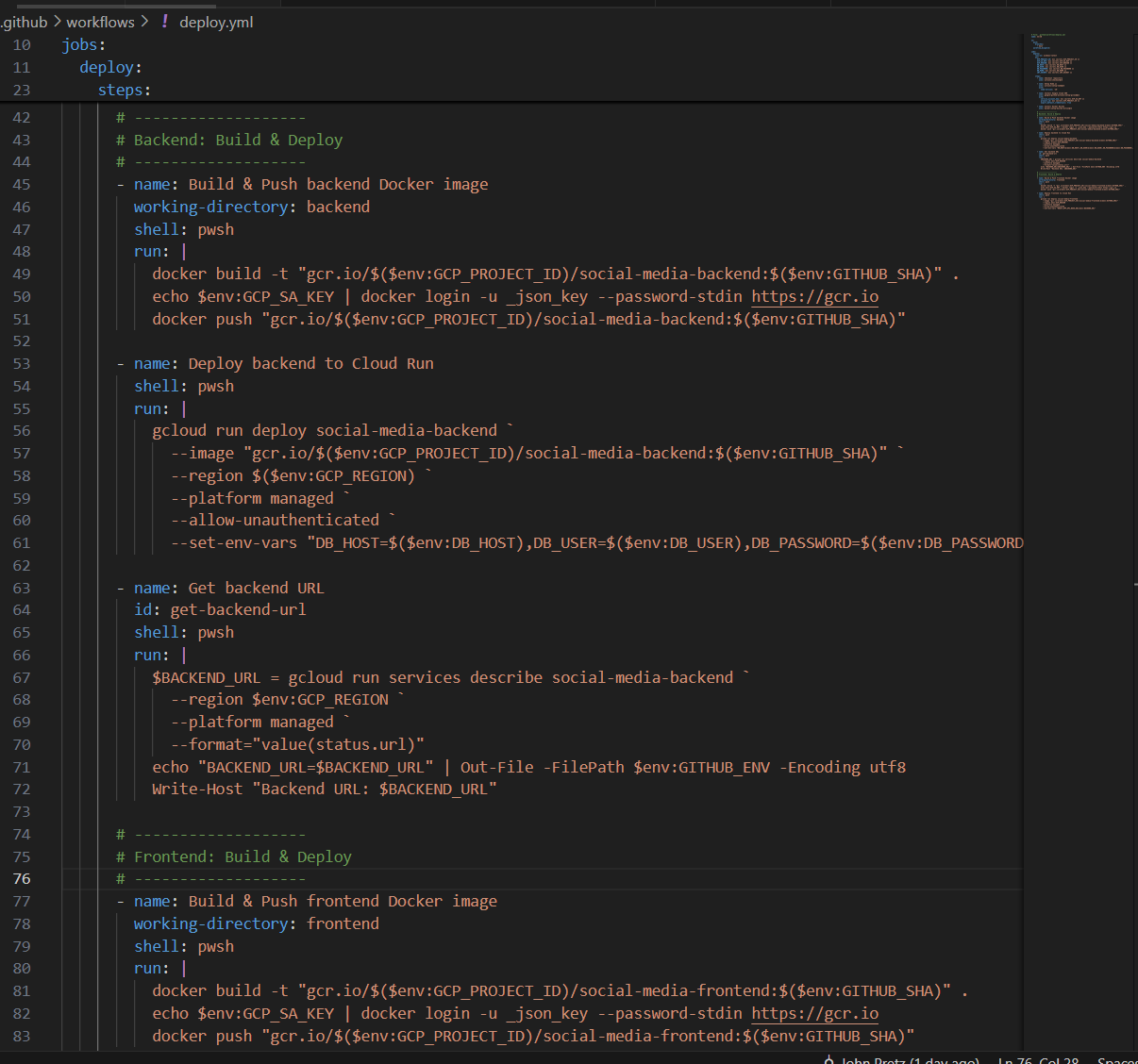
A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A GitHub Actions CI/CD pipeline automates the build, testing, deployment, and monitoring processes, ensuring a streamlined and repeatable deployment process.



1. Security & Scalability Measures

The application implements multiple security measures to protect data and user interactions. Authentication and authorization are handled through JWT tokens, and access to Cloud Run services and Cloud SQL is restricted using Google Cloud IAM roles following the principle of least privilege. Data in transit is encrypted using HTTPS/TLS, and database connections are encrypted as well. These measures ensure that both user credentials and application data remain secure.

A screenshot of a computer

AI-generated content may be incorrect.

A close-up of a message

AI-generated content may be incorrect.Scalability is achieved through Cloud Run's automatic scaling capabilities, allowing backend and frontend services to handle increased traffic. Can be scaled using read replicas and connection pooling to maintain performance under high loads. An API Gateway or load balancer can also be configured to distribute incoming requests efficiently, ensuring consistent performance even during traffic spikes.

1. Monitoring & Logging

Monitoring and logging are crucial for maintaining application reliability. Backend and frontend logs are captured using Google Cloud Logging, which includes information on API requests, authentication attempts, and errors.

A screenshot of a computer

AI-generated content may be incorrect.Google Cloud Monitoring tracks performance metrics, including CPU and memory usage, request latency, and error rates. Alerts are configured to notify administrators in case of failures, high latency, or unusual error patterns. Testing and verification are performed using Postman or curl to ensure all API endpoints respond correctly. Frontend interactions are verified to confirm that login, post creation, commenting, and liking functionality operate as intended, and database updates are confirmed in Cloud SQL.

1. Conclusion: In conclusion, the Social Media Application has been successfully deployed to Google Cloud using Docker, Cloud Run, and Cloud SQL. The deployment process is automated via CI/CD, security is enforced through JWT authentication and IAM access controls, and Cloud Run and Cloud SQL features support scalability. Comprehensive monitoring and logging ensure that the application is reliable and maintainable, making it production-ready and capable of handling increased user traffic efficiently.

| Reference: |
| --- |
| GitHub Docs. *Using Git*. Available: <https://docs.github.com/en/get-started/using-git>  Docker Docs. *Dockerfile reference*. Available: <https://docs.docker.com/engine/reference/builder/>  React Docs. *Getting Started with React*. Available: <https://reactjs.org/docs/getting-started.html>  GitHub Docs. *Using Git*. Available: https://docs.github.com/en/get-started/using-git  React Docs. *Getting Started with React*. Available: <https://reactjs.org/docs/getting-started.html>  Docker Docs. *Dockerfile reference*. Available: <https://docs.docker.com/engine/reference/builder/>  jwt.io. *Introduction to JWT*. Available: <https://jwt.io/introduction/> |

GitHub Repository: <https://github.com/JPretz/sweng861-2025-FALL-Social_Media_Application_to_Public_Could>

Google Cloud Project ID: sweng861-socialmediapc