Project Proposal

Monitoring a Containerized URL Shortener Webservice

Project Description:

This project aims to design, containerize, and monitor a functional URL shortener webservice using DevOps practices. The application will allow users to shorten long URLs, store mappings, and handle redirects efficiently. Once built, the service will be instrumented to expose custom performance metrics such as request latency, error rates, and request counts.
Monitoring and visualization will be implemented using Prometheus (for metrics collection) and Grafana (for dashboard visualization and alerting). The entire stack—application, database, and monitoring tools—will run locally using Docker Compose, ensuring an isolated, reproducible, and fully observable environment.

Group Members & Roles:

- Mohamed Adel Ali – Develop a functional URL shortener webservice with SQLite storage, exposing POST and GET endpoints for creating and redirecting short URLs.
- Ziad Mahmoud El sayed – Create a Dockerfile and an initial docker-compose.yml to build, deploy, and run the webservice locally for testing and validation.
- Mohamed Ayman Nabawy – Instrument the webservice with custom Prometheus metrics, configure scraping, and integrate Prometheus into the Docker Compose setup.
- · Ziad Ahmed Mahmoud – Add Grafana to the stack, connect it to Prometheus, and create a dashboard to visualize key webservice metrics in real time.
- Mostafa Khaled Mostafa – Configure alerting and persistent storage for all services, test data retention, and document the entire system and API in a README file.

Team Leader:

- Ziad Mahmoud El sayed

Objectives:

- Develop a fully functional and containerized URL shortener webservice.
- Implement monitoring and observability using Prometheus and Grafana.
- Automate deployment using Docker Compose.
- Visualize system health, latency, and performance metrics in real time.
- Ensure persistence and reliability through Docker volumes and alerting systems.

Tools & Technologies:

- Programming Framework: Python (Flask) or Node.js (Express)
- Containerization: Docker, Docker Compose
- Monitoring & Visualization: Prometheus, Grafana
- Database: SQLite
- Alerting: Grafana Alerts
- Version Control: Git & GitHub

Milestones & Deadlines:

Week	Milestone	Key Tasks & Deliverables
Week 1	Build & Containerize Webservice	Develop and containerize the URL shortener; create Dockerfile and docker-compose.yml.
Week 2	Add Prometheus Monitoring	Add metrics instrumentation, configure Prometheus, and integrate with Docker Compose.
Week 3	Implement Grafana Dashboards	Connect Grafana to Prometheus, build dashboards for metrics visualization.
Week 4	Configure Alerts & Persistence	Set up alerting in Grafana, add persistent volumes, test reliability, and finalize documentation.

KPIs (Key Performance Indicators):

1. Infrastructure & Automation

- Successful setup of Docker and Docker Compose for multi-container orchestration.
- Automated environment setup and teardown for testing and deployment.

2. Pipeline Efficiency & Performance

- Streamlined build and deployment workflows using Docker.
- Minimal downtime during container restarts or updates.

3. Code Integration & Testing

- Implementation of automated testing for API endpoints.
- Continuous integration through Git version control and local builds.

4. Deployment & Cloud Management

- Local deployment managed with Docker Compose.
- Future scalability plan for deployment to cloud (optional: Docker Hub or Kubernetes).

5. Monitoring & Reliability

- Real-time system metrics collection via Prometheus.
- Grafana dashboards for performance visualization.
- Configured alerting for high latency or error thresholds.
- System uptime and persistence verification through Docker volumes.

Conclusion:

This project integrates key DevOps principles—automation, observability, and reliability—into a complete microservice system. The end result will be a fully monitored, containerized URL shortener application with clear dashboards and alerts, providing hands-on experience in DevOps tools and modern software delivery pipelines.