1. Overview

This document describes the DevOps setup for the ERP project, including CI/CD workflows, Docker configuration, and container builds for Backend, Frontend, HR, and Finance services.

2. GitHub Actions Workflows

2.1 Backend CI/CD (Authentication Service)

Trigger: push on dev, pull_request on main, workflow_dispatch **Environment:** SERVICE_NAME: authentication-service

Jobs Overview

Job Name	Purpose
gitleaks	Detect secrets in code using GitLeaks
sonarqube	Perform SAST using SonarQube
depchecktest	Perform OWASP dependency check
build-scan	Build Docker image, run security scans, generate SBOM, push image to DockerHub
deploy	Update Docker image tag in docker-compose.yml, deploy to EC2 via deploy.sh
zap_scan	Run OWASP ZAP full scan against deployed service

2.2 Frontend CI/CD

Trigger: push on dev, pull_request on main, workflow_dispatch **Environment:** SERVICE_NAME: frontend

Jobs Overview

Job Name	Purpose
gitleaks	Secret detection
sonarqube	SAST for Angular application
depcheck	OWASP Dependency-Check for frontend
build-scan	Build Docker image, run security scans, generate SBOM, push image

Job Name	Purpose
deploy	Update Docker image tag, deploy frontend to EC2
zap_scan	Run OWASP ZAP scan on deployed frontend

2.3 HR Service CI/CD

Trigger: push on dev, pull_request on main, workflow_dispatch **Environment:** SERVICE_NAME: hr-service

Jobs Overview

Job Name	Purpose
gitleaks	Secret detection
sonarqube	Perform SAST for HR service
depchecktest	OWASP Dependency-Check
build-scan	Build Docker image, scan for security vulnerabilities, generate SBOM, push image
deploy	Deploy HR service to EC2, update image tags
zap_scan	Optional ZAP scan after deployment

2.4 Finance Service CI/CD

Trigger: push on dev, pull_request on main, workflow_dispatch **Environment:** SERVICE_NAME: finance-service

Jobs Overview

Job Name	Purpose
gitleaks	Secret detection
sonarqube	Perform SAST for Finance service
depchecktest	OWASP Dependency-Check
build-scan	Build Docker image, scan for security vulnerabilities, generate SBOM, push image
deploy	Deploy Finance service to EC2, update image tags
zap_scan	Optional ZAP scan after deployment

3. Docker Compose Setup

Version: 3.9

Services Overview

Service	Image/Build	Ports	Notes / Healthchecks
postgres	postgres:15	5432	Database with healthcheck pg_isready
authentication-service	./authentication-service	8080	Depends on Postgres, healthcheck via Spring Boot actuator
hr-service	./hr-service	8081	Depends on Postgres, healthcheck via actuator with basic auth
finance-service	./finance-service	8082	Depends on Postgres, healthcheck via actuator
frontend	./frontend	80	Depends on authentication-service, healthcheck on /

Networks and Volumes

- Network: erp-network (bridge)
- Volume: pg_data

4. Dockerfiles

4.1 Backend Services (Authentication, HR, Finance)

```
FROM maven:3.9.6-eclipse-temurin-21 AS build
WORKDIR /app
COPY pom.xml .
COPY src ./src
RUN mvn clean package -DskipTests

FROM eclipse-temurin:21-jre
WORKDIR /app
COPY --from=build /app/target/*.jar app.jar
EXPOSE 8080
CMD ["java", "-jar", "app.jar"]
```

4.2 Frontend Service

```
FROM node:18 AS build
WORKDIR /app
COPY package*.json ./
RUN npm ci --ignore-scripts
COPY . .
RUN npm run build
```

```
FROM nginx:alpine
RUN rm -rf /usr/share/nginx/html/*
COPY --from=build /app/dist/erp-frontend /usr/share/nginx/html
RUN mkdir -p /var/cache/nginx /var/run /var/log/nginx /tmp
    && chown -R 101:101 /usr/share/nginx/html /var/cache/nginx /var/run /var/
log/nginx /tmp
    && chmod -R 755 /usr/share/nginx/html /var/cache/nginx /var/run /var/log/
nginx /tmp
    && sed -i '/pid\s\+\/run\/nginx.pid;/d' /etc/nginx/nginx.conf
EXPOSE 80
USER 101
CMD ["nginx", "-g", "pid /tmp/nginx.pid; daemon off;"]
```

5. Security & Monitoring

- Gitleaks: Secret scanning
- SonarQube: SAST for code vulnerabilities
- OWASP Dependency Check: Dependency vulnerabilities
- Trivy: Docker image scanning & SBOM generation
- ZAP: Post-deployment automated security scan

6. Deployment

- Deployment script: `scripts/deploy.sh`
- Updates Docker image tags in docker-compose.yml
- Deploys backend, HR, Finance & frontend to EC2
- Health checks ensure deployment success

7. Issues & Solutions

Issue / Problem	Description	Solution / Fix
Backend service failing healthcheck	`authentication-service` was failing the healthcheck during startup	Adjusted the healthcheck endpoint to `curl -f http://localhost:8080/actuator/health || exit 1` with `start_period` 30s
Frontend container not starting	Angular build files were not copied correctly into Nginx	Used multi-stage build and corrected the copy path to `/usr/share/nginx/html`
Dependency scan failing	OWASP Dependency-Check was failing on some libraries	Added `--enableRetired` and `--enableExperimental` to bypass temporary errors

Issue / Problem	Description	Solution / Fix
Docker image tag mismatch	During deployment, the old Docker image was used	Added a step to update the image tag in `docker-compose.yml` before deployment
HR/Finance healthcheck issues	Services sometimes started before Postgres was ready	Configured proper `depends_on` with `condition: service_healthy` and healthchecks