

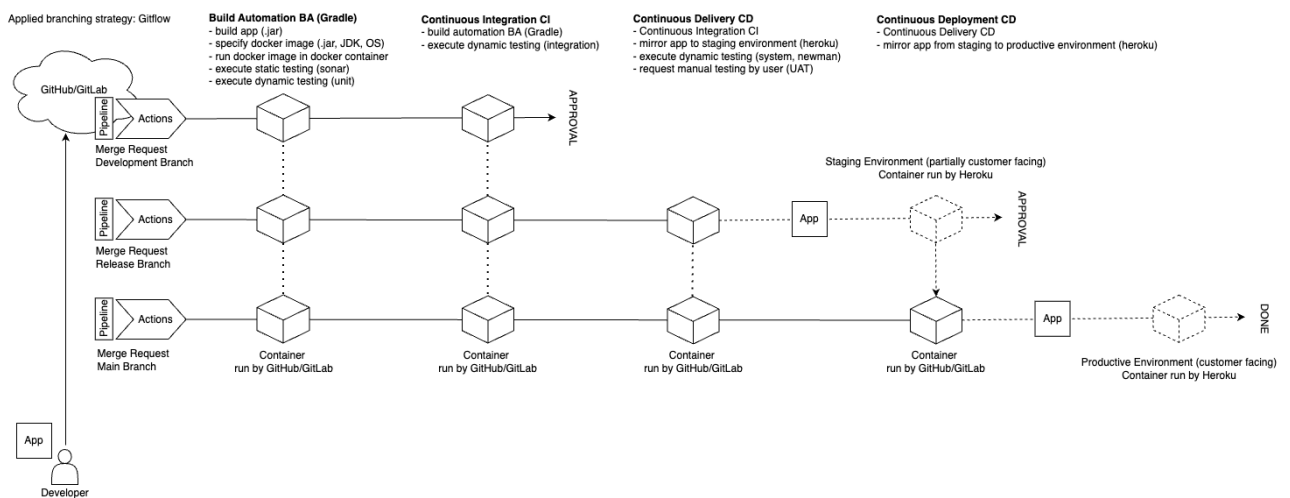
## Leistungsbeurteilung 151 Datenbanken in Web-Applikationen einbinden

Basierend auf LBV 151-1 seitens Ecole des Métiers Fribourg (EMF Informatique)

### Zusatzaufgabe DevOps / Kubernetes (12 Punkte, geschätzter Aufwand 12 Stunden)

Vorliegende Aufgabe sieht die Implementation einer Continuous Integration Pipeline sowie eine horizontale Skalierung der Spring Boot Applikation auf einem Kubernetes Cluster vor. Für beide Teile kann das React Frontend vernachlässigt werden.

### DevOps



### Anforderungen

- **A1 Implementation eines Gitflows (1 Punkt)**  
Der Gitflow sieht nebst Feature Branches das Führen eines Development, Release sowie Main Branches vor. Hotfix Branches können vernachlässigt werden.  
Siehe: <https://datasift.github.io/gitflow/IntroducingGitFlow.html>
- **A2 Implementation einer Pipeline Stage für die Build Automation (2.5 Punkte)**  
Schritt 'build app' = `./gradlew clean bootJar`  
Schritt 'specify docker image' = existierendes Dockerfile aus Kernaufgabe  
Schritt 'run docker image' = hier rennen wir in eine Abhängigkeit mit PostgreSQL. Es würde sich also anbieten auf unser docker-compose.yml zurückzugreifen und den Stack über docker compose up zu starten  
Schritt 'execute static testing (sonar)' = `./gradlew sonarqube`  
Schritt 'execute dynamic testing (unit)' = `./gradlew test --tests` [Referenz zu Unit Tests]
- **A3 Implementation einer Pipeline Stage für die Continuous Integration (2.5 Punkte)**  
Schritt 'execute dynamic testing (unit)' = `./gradlew test --tests` [Referenz zu Integration Tests]

## Kubernetes

### Anforderungen

- A1 Deployment (2 Punkte)  
Deployment der Spring Boot Applikation auf dem Cluster. Man stelle sicher, dass unter spec/replicas 3 pods gefordert sind.
- A2 Service (2 Punkte)  
Deployment eines Services auf dem Cluster. Man stelle sicher, dass vom Service unter spec/type ein LoadBalancing gefordert ist.
- A3 Ingress + Monitoring aller 3 Spring Boot Pods (2 Punkte)  
Deployment eines Ingress auf dem Cluster.  
Folgen der Logs aller 3 Spring Boot Pods. Dies kann über den Befehl `kubect! logs -f [name des Pods]` erreicht werden. Man stelle sicher, dass jeder Pod eingehende HTTP Requests logged. So kann das Loadbalancing der Requests verfolgt werden.