

# **GROEPSCONTRACT**

---

INTEGRATIEPROJECT J 3

# Rollen

DevOps Team (OP) - Younes, Rayan, Mateen, Matthias

Verantwoordelijk voor:

- Kubernetes deployment
- Docker & containerregistries
- Terraform (IaC)
- CD pipelines
- Monitoring & logging
- Deployment documentatie
- Operationele stabiliteit

## Dev Teams

- CI pipeline
- Frontend Team
  - React-frontend, API-calls
- Backend Services Team
  - Domain logic, API endpoints, database interacties
- AI Services Team
  - Model serving, ML logic, event flows

# Deployment Agreement

De Dev en Ops team komen **formeel overeen**:

Werk dat door **DevOps** uitgevoerd wordt

- Deployments naar test en productie (volledige opzet
- infrastructuur) Opzetten & onderhouden van K8s clusterstructuur
- Schrijven & beheren van deployment YAML's
- Terraform provisioning (cloud resources)
- Monitoring dashboards en loggingconfiguratie
- Technische ondersteuning bij integratie
- Services, poorten, protocollen en variabelen doorgeven aan het Dev team
  - op deze manier werken alle teams met eenzelfde manier van deployen

Wat **Dev** moet aanleveren (specificaties & requirements)

Voor elke service levert Dev correct en volledig:

- Dockerfile (productieklaar)
- Health endpoints
- Database requirements (migrations, schema's)
- API-documentatie / expected responses

## Deadline-afspraken

Er wordt enkel met duidelijke, voorspelbare **deadlines** gewerkt.

**Geen** last-minute werk. **Geen** nachtwerk (van onze kant). **Geen** "net op tijd".

## Documentatieplicht (GitLab + DevOps Discord)

- Elke beslissing → wordt in GitLab gedocumenteerd én vermeld in het DevOps-Discordkanaal.
- Elke wijziging → wordt in GitLab gelogd én aangekondigd in het DevOps-Discordkanaal.
- Elk probleem/blocker → moet als GitLab-issue bestaan en wordt gemeld in het DevOps-Discordkanaal.
- Elke deploy-aanvraag → moet via een officieel GitLab-issue gebeuren met labels & info, en eveneens in het DevOps-kanaal vermeld worden.

DevOps voert geen enkele deploy uit zonder officieel GitLab-issue én een melding in het DevOps-Discordkanaal.

# Aanleveringen en Verwachtingen

De **Dev-teams** leveren hun werk altijd op voorhand aan zodat DevOps voldoende tijd heeft om te testen, valideren en een correcte deployment-voorbereiding uit te voeren.

De volgende **aanleveringen** zijn **verplicht**:

Test container image:

- Het Dev-team levert de test image tijdig vooraf aan zodat DevOps deze kan testen, valideren en eventuele integratieproblemen kan detecteren.

Final container image:

- De final image wordt eveneens vooraf aangeleverd zodat DevOps deze zonder tijdsdruk kan deployen, documenteren en controleren richting de sprintdemo.

Refinement en user stories:

- Alle user stories moeten volledig refined, duidelijk en volledig zijn vóór er met de uitvoering wordt gestart. Last-minute refinement of incomplete user stories wordt niet geaccepteerd.

Service-informatie:

- Dev levert tijdig alle nodige informatie zoals environment variables en protocollen.
- DevOps bepaalt welke poorten gebruikt worden, en Dev volgt deze instructies zonder afwijkingen.

**DevOps** levert op zijn beurt de nodige deployment-onderdelen op een voorspelbare, **correcte** en **traceerbare manier**:

- DevOps voorziet de Kubernetes-YAML bestanden, deploymentconfiguratie en integratiedocumentatie op basis van de tijdig aangeleverde containers van de Dev-teams.
- DevOps valideert de aangeleverde images, controleert health endpoints, logging en servicecommunicatie.
- DevOps zorgt voor een stabiele testomgeving en een consistente workflow tussen Dev en Ops.
- DevOps communiceert duidelijke poortconfiguraties, vereiste protocollen en eventuele infrastructurele vereisten die Dev moet volgen.

# Review & Feedback

Er wordt regelmatig review gedaan op:

- Pipelines (build, test, packaging)
- Deployment readiness (volledigheid, stabiliteit, requirements)
- Feedback of afspraken moeten altijd bevestigd worden in het DevOps-Discordkanaal, zodat DevOps de status transparant kan volgen.

Geen enkel onderdeel gaat live (test of productie) zonder:

- Een **groene** pipeline
- Een finale **bevestiging** in het DevOps-Discordkanaal

# Conflicten & Escalaties

Wanneer er problemen of onduidelijkheden zijn, wordt deze procedure strikt gevolgd:

## Stap 1: Intern oplossen

- Dev en DevOps bespreken het probleem in het officiële DevOps-Discordkanaal.
- Het doel is om het probleem binnen het team op te lossen op basis van objectieve info.

## Stap 2: Dev–Ops overleg

- Indien de blokkade aanhoudt, wordt dit besproken in het volgende Dev–Ops overlegmoment, en:
  - Het besluit wordt bevestigd in het DevOps-Discordkanaal

## Stap 3: Escalatie naar docent

- Pas wanneer de interne escalatie mislukt, wordt de docent gecontacteerd via Teams.
- De docent wordt enkel aangesproken als:
  - er bewijs is van eerdere Discord-besprekingen
  - de interne overlegprocedure is gevolgd

# Handtekeningen

Onderstaande partijen **bevestigen** dat zij dit document **gelezen, begrepen** en **volledig goedgekeurd** hebben.

Door te ondertekenen gaan alle teams akkoord met de afspraken, verantwoordelijkheden, communicatievereisten en samenwerkingsregels zoals beschreven in dit contract.

## DevOps Team 4

Younes El Azzouzi



Mateen Murtaza



Matthias Adriaenssen

Rayan Boufker

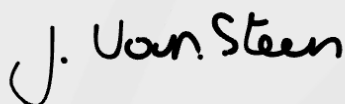


## Dev Team 12

Tim Noelmans



Jiri Van Steen



Nichola Bauwelinck



Siebe De Vos

Mohamed El Jafoufi



Gelezen en goedgekeurd op 21/11/2025 door alle betrokken leden  
van het DevOps-team en de Dev-teams van IP3.