

GINF4-B, DevOps, Contrôle 2

29.12.2025

Exercice 1

Vous devez développer une application web avec une équipe de 6 développeurs sur une période de 5 mois :

- A. Choisissez entre une stratégie Gitflow ou Trunk-based Development, en justifiant votre choix.
- B. Dans quel type de développement utilise-t-on principalement les Feature Flags ? à quoi servent-ils ?
- C. L'équipe devrait-elle travailler sur un seul dépôt partagé ou chaque développeur doit-il créer un fork (copie) du dépôt sur son compte GitHub ? Justifiez votre réponse.
- D. Vous devez développer une nouvelle fonctionnalité. sur quelle branche créez-vous votre code ? sur votre machine locale ou sur le dépôt distant ?
- E. Expliquez le processus de PR et code review avant merge.
- F. A quel moment devriez-vous faire un git pull ?
- G. Décrivez un scénario qui provoquerait un conflit de merge.

Exercice 2

Votre camarade a créé un fichier `.github/workflows/ci.yml` pour lancer les tests lint dans un premier job, puis lancer l'étape 'build and push' dans un deuxième job (si le premier job est effectué avec succès). Modifier ce fichier en respectant les consignes suivantes :

1. Lancer le workflow sur ubuntu et windows.
2. Corriger la séparation des jobs.
3. Ajouter un nom descriptif à chaque étape sans nom.
4. Ajouter un test de qualité de code des fichiers en JavaScript (.js), qui inclut la correction des erreurs identifiées par l'outil lint.
5. Ajouter la condition « `github.ref_name == 'main'` » au job docker.
6. Pousser vers DockerHub au lieu de GHCR.

Répondre séparément à ces questions :

- A. Est-ce que le workflow va se lancer si on modifie le fichier `.github/workflows/ci.yml` depuis la branche 'develop' directement sur GitHub ?
- B. à quoi servent les noms des jobs et des étapes ?
- C. à quoi sert la condition ajoutée dans la question 5, en pratique ?
- D. Où doit-on définir la variable 'secrets.GITHUB_TOKEN' ?
- E. Quelle est la valeur de chacune des variables [github.workflow, github.event_name] dans ce workflow ?
- F. Citer 2 fichiers de configurations nécessaires pour lancer les tests (lint ou unitaires). Spécifier leur chemin.
- G. 'ctrl_app' sera poussé sous quelle version ?

```
name: CI pipeline

on:
  push:
    branches: ['main', 'develop']
    paths: ["path/to/app/**"]
  pull_request:
    branches: ['main', 'develop']
    paths: ["path/to/app/**"]
  schedule:
    - cron: '30 8 1 * *'

jobs:
  job1:
    runs-on: ubuntu-latest
    defaults:
      run:
        working-directory: ./path/to/app/
    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Setup Node.js
        uses: actions/setup-node@v4
        with:
          node-version: 'latest'
          cache: 'npm'
          cache-dependency-path:
            ./path/to/app/package-lock.json

      - run: npm install

      - run: |
        npm run htmlhint **/*.html
        npm run stylelint **/*.css --fix

      - run: npm jest

      - uses: docker/login-action@v3
        with:
          registry: ghcr.io
          username: ${{ github.actor }}
          password: ${{ secrets.GITHUB_TOKEN }}

      - uses: docker/build-push-action@v5
        with:
          context: ./path/to/app/
          push: true
          tags: ghcr.io/${{ github.actor }}/ctrl_app
```

Exercice 3

Vous avez créé une image Docker nommée ‘myapp’ et lancé un conteneur ‘myapp_c’ à partir de cette image avec les caractéristiques suivantes :

- L'application à l'intérieur du conteneur écoute sur le port 3000.
 - Ce port doit être accessible via le port 5050 de votre machine locale
 - Le dossier ‘/data’ du conteneur est monté sur le volume ‘data-volume’,
 - et le dossier ‘/app’ est monté sur le dossier ‘\$PWD/app’ de votre machine locale.
- A. Ecrire les lignes de commande de création de ‘myapp’ et ‘myapp_c’.
- B. Lequel doit-on pousser à un répertoire distant ?
- C. Citer un avantage de Docker Hub et un autre de GHCR.
- D. Ecrire toutes les commandes nécessaires pour pousser vers DockerHub.
- E. Expliquer ces commandes utilisées dans les Dockerfile : RUN, ARG, FROM, WORKDIR.
- F. Décrire deux caractéristiques de chacune des stratégies de déploiement suivantes : Canary, Blue/Green.