

# TP 2 – Tests intégrés et CI/CD

## F68 - groupe 14009

### Objectif du laboratoire :

L'objectif de ce TP est d'évaluer la capacité des étudiants à :

- Écrire et exécuter des tests automatisés avec PyTest sur une API FastAPI.
- Mettre en place un pipeline d'intégration continue (CI) avec GitHub Actions.
- Comprendre les bases d'un pipeline de déploiement continu (CD) simulé.

À travers ce TP, l'étudiant doit démontrer sa maîtrise des outils de tests, de CI/CD et de l'automatisation des workflows, en s'appuyant sur le projet FastAPI utilisé dans les TD précédents.

### Préparation de l'environnement :

Ce TP se base sur les TD réalisés en classe pour la partie CI/CD.

Avant de commencer à répondre, remplacez le contenu de vos fichiers ci.yml et cd.yml par ceux fournis avec cet énoncé.

### Livrables attendus :

La remise doit être effectuée sur Léa au plus tard le **27 septembre 2025 à 23 h 59 min 59 s**.

Le livrable attendu est une archive compressée (.zip) contenant :

- 1) Un document PDF (ou Word) avec des captures d'écran montrant le résultat de chaque exécution dans GitHub Actions (les informations essentielles doivent être visibles, voir exemple ci-après).

The screenshot shows the GitHub Actions interface for a repository named 'TD-CI-DS-API-FastAPI-Docker'. The workflow 'CI - FastAPI' is displayed, triggered by a push to the 'dev' branch. The workflow status is 'Success' with a total duration of '2m 18s' and 3 artifacts. A sidebar on the left lists the jobs: lint, type-check, tests (3.10), tests (3.11), security, and docker-smoke. The main area shows a visual representation of the workflow steps: lint (12s), type-check (14s), a matrix test step (2 jobs completed), security (0s), and docker-smoke (1m 5s). A yellow box highlights the 'ci.yml' file in the workflow configuration.

- 2) Le fichier test\_api.py mis à jour.
- 3) Le fichier ci.yml mis à jour.
- 4) Le fichier cd.yml mis à jour.

## Exercice 1 : Tests intégrés

Dans le fichier test\_api.py, ajouter :

- 1) Un test qui valide une prédiction correcte avec les valeurs [1.0, 2.0, 3.0].
- 2) Un test qui valide une prédiction incorrecte (ex. : comparer avec un résultat attendu volontairement faux).
- 3) Un test qui envoie un JSON incorrect (exemple : champ features manquant, {[3.5, 1.2, 4.9]}).
- 4) Réaliser les étapes requises pour activer le pipeline CI-FastAPI et fournir une capture d'écran GitHub Actions avec les tests exécutés avec succès.

## Exercice 2 : Intégration continue

- 1) Modifier les déclencheurs de votre pipeline CI afin qu'il soit activé uniquement :
  - a) Sur push dans la branche dev,
  - b) Sur pull\_request vers la branche main.
- 2) Réaliser les étapes nécessaires et pousser le code sur la branche dev (capture d'écran GitHub Actions avec succès).

- 3) Réaliser un merge de la branche dev vers main via une pull request (capture d'écran GitHub Actions avec succès).
- 4) Bonifier le pipeline CI pour :
  - a) Mesurer la couverture (pytest --cov).
  - b) Ajouter une étape de vérification de la sécurité des dépendances (safety).
  - c) Mettre la vérification des dépendances dans un job séparé, exécuté en premier.
  - d) Faire dépendre le job de tests de ce job de sécurité.

### Exercice 3 : Déploiement continu

- 1) Expliquer le rôle de cette partie du pipeline CD :

```
on:
  workflow_run:
    workflows: ["CI - FastAPI"]
    types:
      - completed
```

- 2) Expliquer le rôle de cette partie du pipeline CD :

```
if: ${ github.event.workflow_run.conclusion == 'success' &&
github.event.workflow_run.head_branch == 'main' }
```

- 3) Expliquer pourquoi nous n'avons pas effectué de déploiement sur un serveur physique distant lors du cours (partie commentée dans le cd.yml).

### Grille d'évaluation :

Critère	Détails d'évaluation	Points
<b>Exercice 1 : Tests intégrés</b>		
1)	Test prédiction correcte [1.0, 2.0, 3.0]	5
2)	Test prédiction incorrecte	5
3)	Test JSON incorrect	5
4)	Capture d'écran GitHub Actions montrant les tests exécutés avec succès	5
<b>Exercice 2 : Intégration continue</b>		
1)	Modification correcte des déclencheurs	10

2) Exécution CI réussie après push sur dev	10
3) Exécution CI réussie après merge via PR sur main	10
4) Bonifier le pipeline CI	
a) Ajout de la couverture	5
b) Ajout de la vérification sécurité	5
c) Mise en place d'un job séparé pour la sécurité	5
d) Dépendance correcte des jobs	5
<b>Exercice 3 : Déploiement continu</b>	
1) Explication claire du déclencheur workflow_run	10
2) Explication claire de la condition if	10
3) Explication pertinente sur l'absence de déploiement réel	10
<b>TOTAL</b>	<b>100</b>