

Rapport LOG430 - Labo 0 : Infrastructure (Git, Docker, CI/CD)



ÉCOLE DE
TECHNOLOGIE
SUPÉRIEURE
Université du Québec

Rédacteur : Damien Sudre - SUDD91370200 - LOG430

Ecriture des Tests

On a **5 fonctions** à tester dans la classe `Calculator` de l'application `calculator.py` :

- `get_hello_message`
- `addition`
- `subtraction`
- `multiplication`
- `division`

Les tests sont rédigés dans le fichier `src/tests/test_calculator.py`. Il existe déjà le cas de test pour la fonction `get_hello_message`, nous nous concentrons alors sur les 4 autres fonctions.

On garde à l'esprit que le but du labo0 n'est pas de rédiger des tests pour une application complexe.

On a alors les 4 cas de tests suivants :

```
12
13 def test_addition():
14     calc = Calculator()
15     assert calc.addition(5, 3) == 8
16
17 def test_subtraction():
18     calc = Calculator()
19     assert calc.subtraction(205, 115) == 90
20
21 def test_multiplication():
22     calc = Calculator()
23     assert calc.multiplication(35, 43) == 1505
24
25 def test_division():
26     calc = Calculator()
27     assert calc.division(5, 2) == 2.5
```

Modification due à des erreurs d'imports

Avec le code et la structure de projet fournis, `pytest` ne parvenait pas à résoudre l'import `from calculator import Calculator`. En effet, l'exécution de `pytest` donnait systématiquement l'erreur suivante, et ce quelque soit le repertoire depuis lequel j'appelais `pytest`.

```
(Labo0) PS C:\Users\exter\Desktop\Etudes\Projet ETS\7 - Automne 2025\LOG430\Labo 0\log430-a25-labo0\sr
c> python -m pytest
===== test session starts =====
platform win32 -- Python 3.12.3, pytest-8.4.2, pluggy-1.6.0
rootdir: C:\Users\exter\Desktop\Etudes\Projet ETS\7 - Automne 2025\LOG430\Labo 0\log430-a25-labo0\src
collected 0 items / 1 error

===== ERRORS =====
_____ ERROR collecting tests/test_calculator.py _____
ImportError while importing test module 'C:\Users\exter\Desktop\Etudes\Projet ETS\7 - Automne 2025\LOG
430\Labo 0\log430-a25-labo0\src\tests\test_calculator.py'.
Hint: make sure your test modules/packages have valid Python names.
Traceback:
C:\Users\exter\AppData\Local\Programs\Python\Python312\Lib\importlib\__init__.py:98: in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
tests\test_calculator.py:7: in <module>
    from testcalculator import Calculator
E   ModuleNotFoundError: No module named 'testcalculator'

===== short test summary info =====
ERROR tests/test_calculator.py
!!!!!!!!!!!!!!!!!!!!!!!!!!!! Interrupted: 1 error during collection !!!!!!!!!!!!!!!!!!!!!!!!!!!!!
===== 1 error in 0.07s =====
(Labo0) PS C:\Users\exter\Desktop\Etudes\Projet ETS\7 - Automne 2025\LOG430\Labo 0\log430-a25-labo0\sr
c>
```

En souhaitant modifier le moins de code que possible, j'ai alors choisi de rajouter la section suivante dans le fichier `src/tests/test_calculator.py` afin de résoudre l'erreur en ajoutant le chemin absolu au `sys.path`:

```
3  SPDX - License - Identifier: LGPL - 3.0 - or -later
4  Auteurs : Gabriel C. Ullmann, Fabio Petrillo, 2025
5  ""
6
7  import os
8  import sys
9
10 sys.path.insert(0, os.path.abspath(os.path.join(os.path.dirname(__file__), "src")))
11
12 from calculator import Calculator
13
14 def test_app():
15     calc = Calculator()
16     assert calc.get_hello_message() == "== Calculatrice v1.0 =="
```

un fichier `__init__.py` a également été ajouté dans `src/tests` pour considérer le module de test.

On obtient alors bien l'exécution des tests :

```
(Labo0) PS C:\Users\exter\Desktop\Etudes\Projet ETS\7 - Automne 2025\LOG430\Labo 0\log430-a25-labo0> p
ytest
===== test session starts =====
platform win32 -- Python 3.12.3, pytest-8.4.2, pluggy-1.6.0
rootdir: C:\Users\exter\Desktop\Etudes\Projet ETS\7 - Automne 2025\LOG430\Labo 0\log430-a25-labo0
collected 5 items

src\tests\test_calculator.py ..... [100%]

===== 5 passed in 0.01s =====
(Labo0) PS C:\Users\exter\Desktop\Etudes\Projet ETS\7 - Automne 2025\LOG430\Labo 0\log430-a25-labo0> _
```

La méthode utilisée pour résoudre ce problème n'est pas recommandée en situation réelle, cependant une résolution plus poussée sortirait du cadre du laboratoire.

Question 1 : Conséquence de la rédaction d'un test éronné avec `pytest`

On modifie légèrement un cas de test pour forcer une erreur :

```
26 def test_multiplication():
27     calc = Calculator()
28     assert calc.multiplication(35, 43) == 1506
29
```

ici, on a modifié la valeur attendue à 1506 (au lieu de 1505)

On obtient alors la sortie suivante en lançant `pytest` :

```
(Labo0) PS C:\Users\exter\Desktop\Etudes\Projet ETS\7 - Automne 2025\LOG430\Labo 0\log430-a25-labo0\src> p
ytest
===== test session starts =====
platform win32 -- Python 3.12.3, pytest-8.4.2, pluggy-1.6.0
rootdir: C:\Users\exter\Desktop\Etudes\Projet ETS\7 - Automne 2025\LOG430\Labo 0\log430-a25-labo0\src
collected 5 items

tests\test_calculator.py ...F. [100%]

===== FAILURES =====
_____ test_multiplication _____

def test_multiplication():
    calc = Calculator()
>    assert calc.multiplication(35, 43) == 1506
E       assert 1505 == 1506
E       + where 1505 = multiplication(35, 43)
E       + where multiplication = <calculator.Calculator object at 0x0000020C0D5D0110>.multiplication

tests\test_calculator.py:28: AssertionError
===== short test summary info =====
FAILED tests/test_calculator.py::test_multiplication - assert 1505 == 1506
===== 1 failed, 4 passed in 0.04s =====
(Labo0) PS C:\Users\exter\Desktop\Etudes\Projet ETS\7 - Automne 2025\LOG430\Labo 0\log430-a25-labo0\src> _
```

On constate alors que l'on obtient un rapport d'erreur qui nous indique quelle assertion de quel test est concerné par l'échec. On a aussi un résumé en première ligne indiquant un "F" lorsqu'un test échoue (pratique pour figurer où se trouve le cas de test dans le fichier).

Pipeline CI + Versionnage du projet

Question 2 : Que font les étapes `Setup` et `Checkout` de la CI ainsi paramétrée :

On remarque que la question doit faire référence aux étapes suivantes :

```
4
5 jobs:
6   build:
7     runs-on: ubuntu-latest
8     environment: Labo0
9
10  steps:
11    - name: Checkout dépôt
12      uses: actions/checkout@v3
13
14    - name: Installer Python
```

1

2

A propos du Setup :

runs-on: ubuntu-latest initie la création d'une machine virtuel utilisant une distribution recente d'Ubuntu.

environment : Labo0 permet de lier le runner du job **build** (celui qui utilise la VM Ubuntu) a l'environnement GitHub 'Labo0' pour lui attribuer des autorisations ou autre parametres.

On a l'execution suivante sur GitHub Actions pour le Setup :

```
Set up job
1 Current runner version: '2.328.0'
2 ▼ Runner Image Provisioner
3   Hosted Compute Agent
4   Version: 20250829.383
5   Commit: 27cb235aab5b0e52e153a26cd86b4742e89dac5d
6   Build Date: 2025-08-29T13:48:48Z
7 ▼ Operating System
8   Ubuntu
9   24.04.3
10  LTS
11 ▼ Runner Image
12   Image: ubuntu-24.04
13   Version: 20250907.24.1
14   Included Software: https://github.com/actions/runner-images/blob/ubuntu24/20250907.24/images/ubuntu/Ubuntu2404-Readme.md
15  Image Release: https://github.com/actions/runner-images/releases/tag/ubuntu24%2F20250907.24
16 ▼ GITHUB_TOKEN Permissions
17   Actions: write
18   Attestations: write
19   Checks: write
20   Contents: write
21   Deployments: write
22   Discussions: write
23   Issues: write
24   Metadata: read
25   Models: read
26   Packages: write
27   Pages: write
28   PullRequests: write
29   RepositoryProjects: write
30   SecurityEvents: write
31   Statuses: write
32   Secret source: Actions
33   Prepare workflow directory
34   Prepare all required actions
35   Getting action download info
36   Download action repository 'actions/checkout@v3' (SHA:f43a0e5ff2bd294095638e18286ca9a3d1956744)
37   Download action repository 'actions/setup-python@v4' (SHA:7f4fc3e22c37d6ff65e88745f38bd3157c663f7c)
38   Complete job name: build
```

A propos du Checkout :

Cette première étape utilise une action officiel de github (avec **uses: actions/checkout@v3**) qui a pour role de cloner (ou plus precisement, d'initialiser) le repo de code sur la VM deployée (sinon il n'y a qu'une VM vide). C'est une étape essentiel pour ensuite pouvoir lancer des tests sur notre code (puisque pour tester le code, on a besoin du dit code).

On a la sortie suivante sur GitHub Actions :

```
Checkout dépôt 0s
1 ▶ Run actions/checkout@v3
14 Syncing repository: DamienSud/LOCAL-log430-a25-labo0
15 ▼ Getting Git version info
16 Working directory is '/home/runner/work/LOCAL-log430-a25-labo0/LOCAL-log430-a25-labo0'
17 /usr/bin/git version
18 git version 2.51.0
19 Temporarily overriding HOME='/home/runner/work/_temp/d17b63e0-ae4c-48be-8427-9a46c647bdc0' before making global git config changes
20 Adding repository directory to the temporary git global config as a safe directory
21 /usr/bin/git config --global --add safe.directory /home/runner/work/LOCAL-log430-a25-labo0/LOCAL-log430-a25-labo0
22 Deleting the contents of '/home/runner/work/LOCAL-log430-a25-labo0/LOCAL-log430-a25-labo0'
23 ▶ Initializing the repository
39 ▶ Disabling automatic garbage collection
41 ▶ Setting up auth
47 ▶ Fetching the repository
133 ▶ Determining the checkout info
134 ▶ Checking out the ref
138 /usr/bin/git log -1 --format='%H'
139 'c93a9d8ab4b07722d5f9a401bfe6d13a83e18c00'
```

Probleme avec la machine virtuelle

Desole mais je n'ai pas reussi a contacter la machine virtuelle meme avec les identifiants fournis ...