



AVIGNON
UNIVERSITÉ

Rapport TP2

(Dés)intégration continue

Étudiant
Michel Marie LAMAH

20 février 2023

Master Informatique
Intelligence Artificielle

UE Génie logiciel avancé
ECUE Techniques de test

Responsables

Daniel SALAS
Emmanuel FERREIRA

UFR
SCIENCES
TECHNOLOGIES
SANTÉ



CENTRE
D'ENSEIGNEMENT
ET DE RECHERCHE
EN INFORMATIQUE
ceri.univ-avignon.fr

Sommaire

Titre	1
Sommaire	2
1 Introduction	3
2 Épisode 1 - CircleCI	3
3 Épisode 2 - Circle.yml	5
4 Conclusion	9
Bibliographie	10

1 Introduction

Dans cet TP[1], il sera question de mettre en pratique les notions d'intégrations continues à travers la plateforme Circle CI.

Pour réaliser ce travail, j'utiliserai :

- **IDE** : IntelliJ IDEA¹
- **Dépôt de repository Git** : Github²
- **Intégration** : CircleCI³

Toutes les modifications réalisées ici se trouvent sur la branche **TP02** de mon repository [3].

Mais si vous souhaitez voir toutes les modifications effectuées depuis le TP1 jusqu'à maintenant vous pouvez voir la branche master [2].

2 Épisode 1 - CircleCI

Pour pouvoir utiliser CircleCI, nous allons devoir nous connecter et pour cela, nous nous rendons sur son [site officiel](#) et nous nous connectons avec notre compte Github.

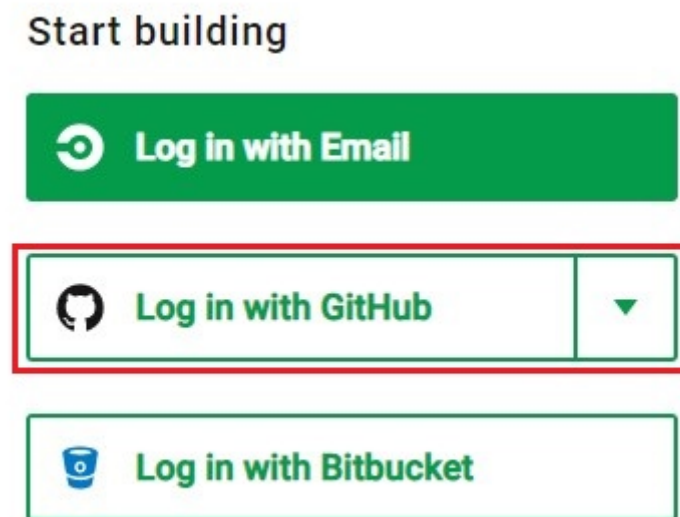


Figure 1. Se connecter avec GitHub

En cliquant sur **Log in with GitHub**, nous retrouverons sur cette page 2 dans laquelle on doit cliquer sur **Authorize circleci** pour pouvoir permettre à circleci d'accéder à nos projets sur GitHub

-
1. <https://www.jetbrains.com/fr-fr/idea/>
 2. <https://github.com/>
 3. <https://circleci.com/>

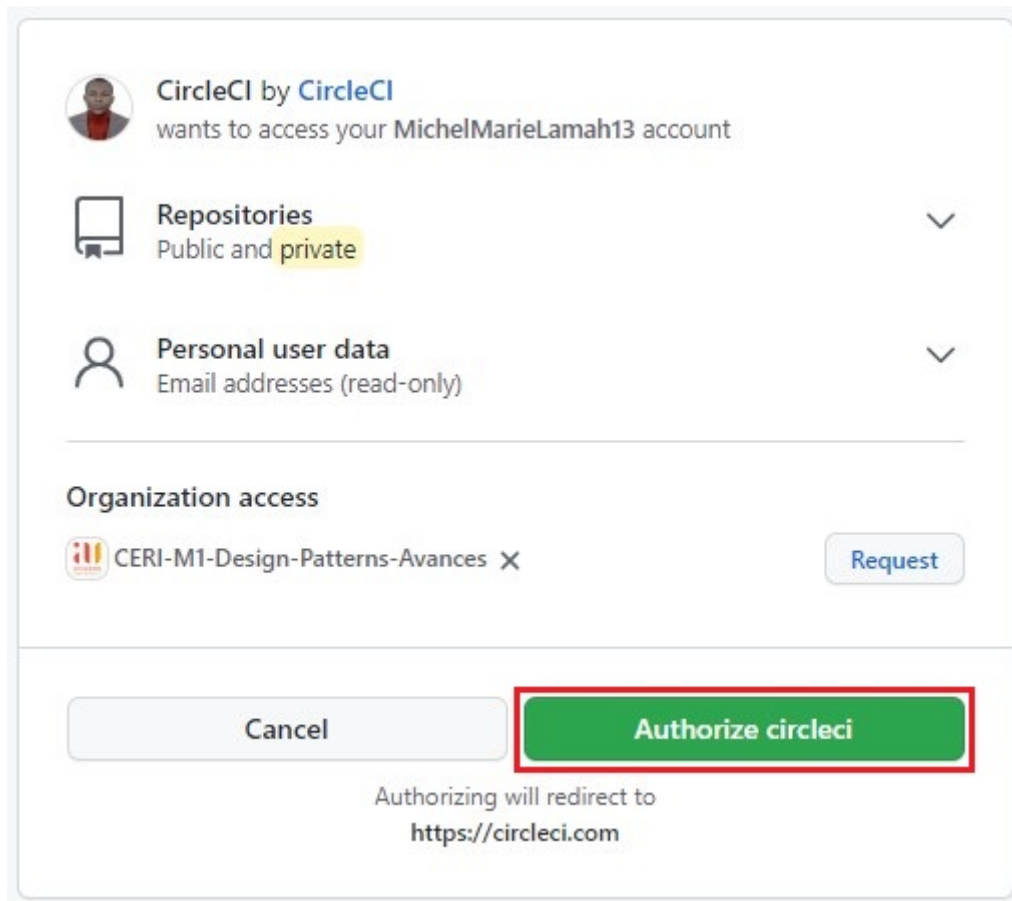



Figure 2. Autoriser circleci

Par la suite, nous spécifions notre mot de passe, sur la page indiquée à l'image 3



Confirm access

Password

.....

[Forgot password?](#)

Confirm

Tip: You are entering **sudo mode**. After you've performed a sudo-protected action, you'll only be asked to re-authenticate again after a few hours of inactivity.

Figure 3. Spécifier le mot de passe

Dans la page qui s'affiche [4](#), on nous demande de configurer notre projet ou de sauter la configuration, nous allons cliquer sur **Skip setup**, pour faire la configuration après

Not looking to set up?

Not ready to set up your project? Explore CircleCI and set up your project when you're ready.

Skip setup

Figure 4. Sauter la configuration

Par la suite on vous posera quelques questions, répondez les puis cliquez sur **Let's go** puis si on vous demande de choisir votre organisation, il suffira de choisir votre compte parmi la liste des comptes qui s'afficheront à l'écran.

3 Épisode 2 – Circle.yml

Par la suite, nous allons maintenant configurer notre projet pour permettre une intégration continue.

Pour cela dans la liste des projet, on selectionne le projet qu'on veut configurer en cliquant sur **Set up project**

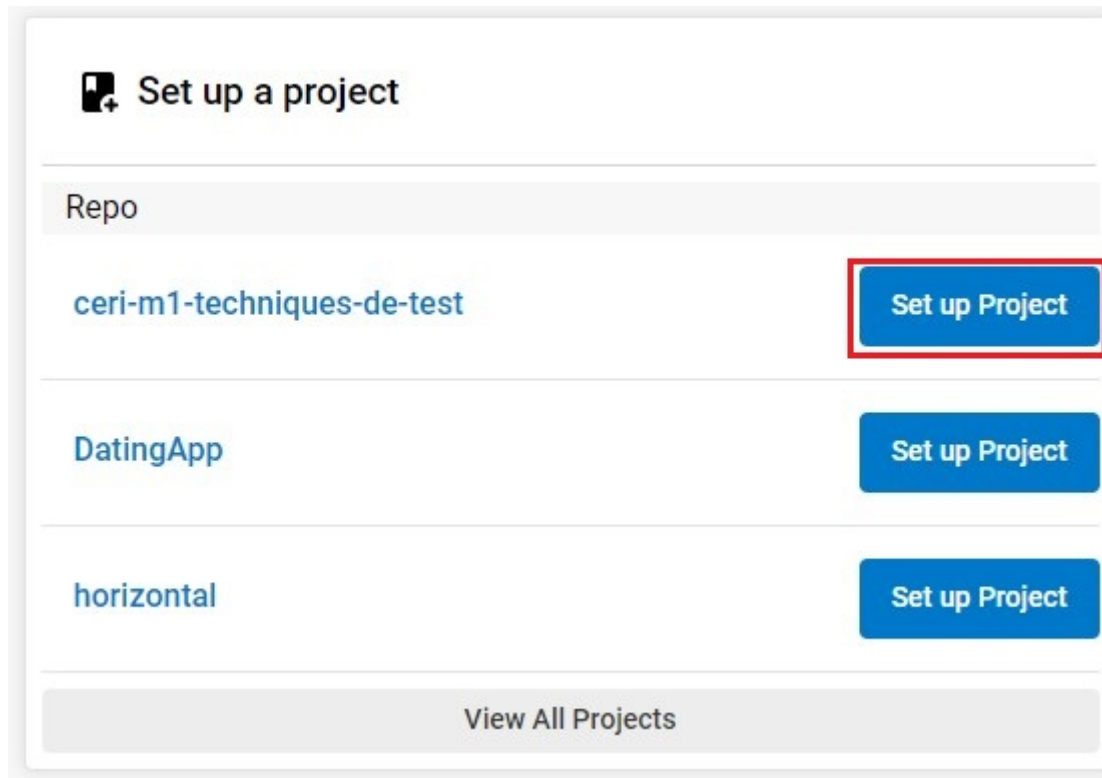


Figure 5. Configurer projet

Dans la fenêtre qui s'affiche, plusieurs options de configurations nous sont proposées, on choisit l'option **Fast**, qui nous fournit un fichier de configuration que nous pourrions par la suite modifier puis nous reviendrons par la suite pour cliquer sur l'option **Fastest**.

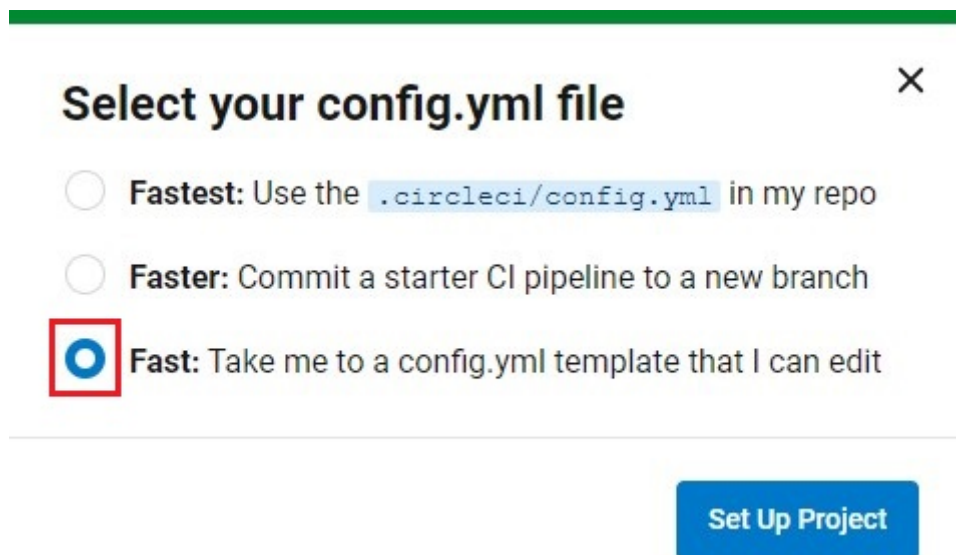


Figure 6. Choisir option de configuration

Par la suite, on choisit le type d'application de notre projet, dans notre cas, il s'agit d'un projet Maven

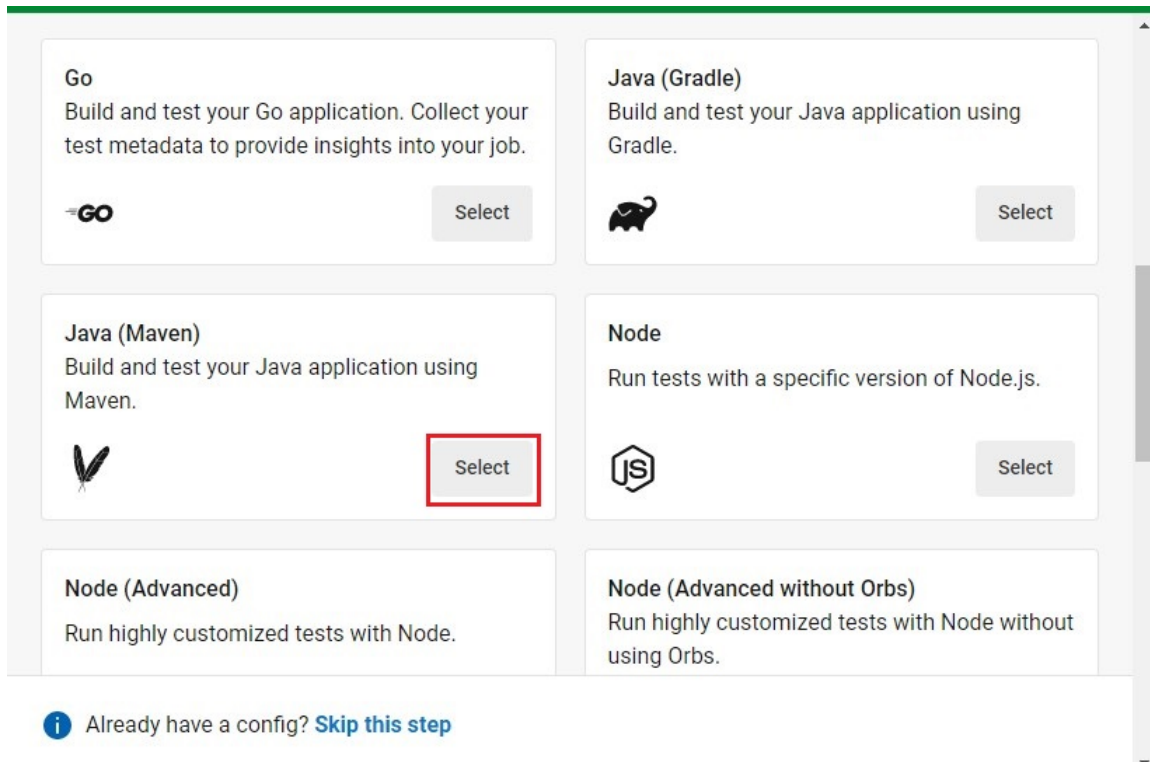


Figure 7. Choisir le type d'application du projet

Nous allons télécharger le fichier de configuration, qui nous est proposé.



Figure 8. Télécharger le fichier de configuration

Nous le modifions comme suit pour spécifier la version du jdk mais aussi la branche ainsi que les opérations qu'on veut effectuer, dans notre cas il s'agit des builds et tests.

```
version: 2.1

jobs:
  build-and-test:
    docker:
      - image: cimg/openjdk:11.0

    steps:
      - checkout
      - run:
          name: Build
          command: mvn -B -DskipTests clean package
      - run:
          name: Test
          command: mvn test

workflows:
  sample:
```

```
jobs:
  - build-and-test:
    filters:
      branches:
        only: master
```

Pour que circleci le reconnaisse, il va falloir le mettre à la racine de notre projet comme suit `.circleci\config.yml`

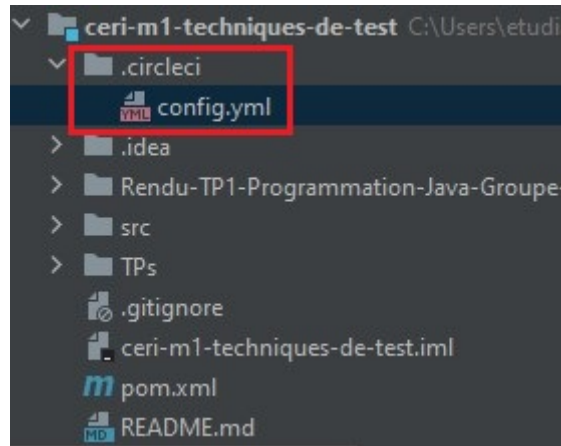


Figure 9. Repertoire de configuration circleci

Nous pouvons maintenant retourner à la configuration de notre projet puis choisir l'option **Fastest** puis choisir la branch master

Select your config.yml file

☒ **Fastest:** Use the `.circleci/config.yml` in my repo

ceri-m1-techniques-de-test

master

☒ `.circleci/config.yml` found on this branch

☐ **Faster:** Commit a starter CI pipeline to a new branch

☐ **Fast:** Take me to a config.yml template that I can edit

Set Up Project

Figure 10. Configuration du projet

En cliquant sur **Set Up Project**, on obtient alors le tableau de board ci-dessous :






Pipeline	Status	Workflow	Branch / Commit	Start	Duration	Actions
ceri-m1-techniques-de-test 1	✓ Success	sample	 master ca3a285	2m ago	 20s	   ...
Jobs		✓ build-and-test 1	16s			

Figure 11. Tableau de board

4 Conclusion

Dans cet tp, il était question de configurer notre projet pour une intégration continue. Ce travail a été utilisé en utilisant la plateforme CircleCi à travers un fichier .yaml de configuration.

Références

- [1] Daniel SALAS. *TP2 : (Dés)intégration continue*. Université Avignon. 2023. url : <https://github.com/Youkoulanda/ceri-m1-techniques-de-test/blob/master/TPs/TP2.md>.
- [2] LAMAH Michel Marie. *TP2 : (Dés)intégration continue [master]*. Université Avignon. 2023. url : <https://github.com/MichelMarieLamah13/ceri-m1-techniques-de-test/tree/master>.
- [3] LAMAH Michel Marie. *TP2 : (Dés)intégration continue [TP02]*. Université Avignon. 2023. url : <https://github.com/MichelMarieLamah13/ceri-m1-techniques-de-test/tree/TP02>.