

## TP 3

### Introduction à la CI/CD avec GitLab-CI

#### 1/ Préambule

Le but de ce TP est de vous montrer comment on peut créer et/ou se servir d'un workflow de traitements appelé « pipeline CI/CD ».

Le but d'un pipeline est de réaliser une chaîne de tâches consécutives sur un projet avec pour objectif global de générer un livrable qui sera déployé sur une infrastructure. Il est constitué d'un ensemble d'étapes (*stage*) et chaque *stage* peut rassembler un ou plusieurs tâches (*jobs*). Les *stages* sont traités selon un ordre séquentiel et à l'intérieur de chaque *stage*, les *jobs* sont également traités de façon séquentielle.

Maintenant que nous savons utiliser Git et GitLab vu au TP2 précédent, nous allons maintenant aborder une fonctionnalité importante de l'outil GitLab, appelée « GitLab-CI » ou « GitLab CI/CD » qui permet de construire et exécuter des pipelines CI/CD. Nous nous servirons des commandes git pour déclencher de façon automatique un pipeline.

#### 2/ Exploration

Dans cette partie, vous allez apprendre de façon individuelle et autonome comment on construit un pipeline en suivant le tutoriel officiel de GitLab : [https://docs.gitlab.com/ee/ci/quick\\_start/](https://docs.gitlab.com/ee/ci/quick_start/) (il est en Anglais. Vous devez vous y habituer, car la langue de l'Informatique c'est l'anglais).

Une fois que vous avez terminé la réalisation du tutoriel et exécuter graphiquement votre premier pipeline, réaliser les traitements suivants pour voir comment à travers un push git, on déclenche le pipeline :

- a) Clonez votre projet en local sur votre machine
- b) Ouvrez le fichier `.gitlab-ci.yml` et apportez-y une petite modification en ajoutant par exemple une ligne « echo ... » sur un job ; ou également ajoutez-y un fichier `test.java` dans le projet
- c) Commitez et poussez votre modification sur GitLab
- d) Repartez immédiatement sur GitLab dans l'onglet *Build > Pipelines*. Que constatez-vous ?

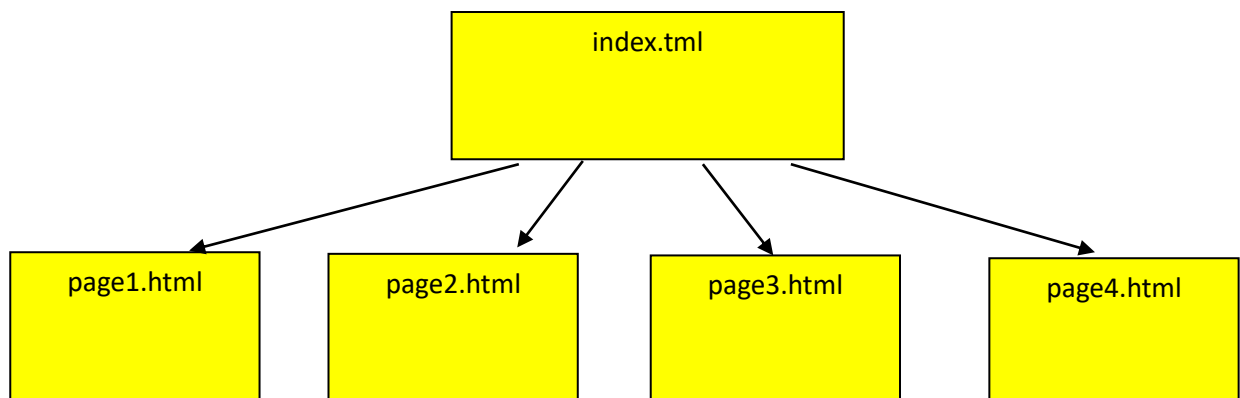
#### 3/ Projet : GitLab-CI & GitLab Pages

Dans cette partie, vous allez travailler en groupe.

Le but est de se servir d'un pipeline CI/CD pour construire et déployer un mini site web en ligne au travers de la fonctionnalité GitLab pages (<https://docs.gitlab.com/ee/user/project/pages/>). Vous allez partir d'un projet *template* existant que vous enrichirez au fur et à mesure de code HTML et le pipeline se chargera à chaque *commit && push* de construire et déployer votre site web à l'image du code source (HTML) que vous avez écrit dans le projet.

- a) Mettez-vous par groupe de travail de 4 personnes
- b) Sur GitLab, créez un Groupe pour votre projet
- c) Ajoutez les membres de votre groupe de travail dans le Groupe GitLab
  - ⇒ Lorsque les membres sont ajoutés directement, ils peuvent voir tous les projets du groupe
- d) Parmi vous, une personne doit cloner en local le projet template suivant : <https://gitlab.com/pages/plain-html>
- e) Pour éviter toute erreur, une fois en local, supprimez le sous dossier « .git », renommez le nom de votre dossier (ici plain-html) avec le nom de votre projet à réaliser, et réappliquez les commandes git du TP2 : « git init », « git remote add », « git add », « git commit », « git push » pour envoyer le projet dans votre groupe afin que les autres membres puissent le récupérer.
- f) L'idée est de coder un site web disposant d'une page principale disposant d'un menu à quatre boutons ou liens. Chaque lien redirige vers une page donnée. Pour éviter les problèmes de conflits de modification du même fichier dans ce TP, chaque développeur travaillera sur une page en ajoutant son code source html en fonction des fonctionnalités qui lui ont été demandées de réaliser dans le groupe.

L'idée est d'enrichir le dossier public/ du projet avec quatre autres fichiers *.html* en plus de l'*index.html* qui rassemblera l'ensemble.



Imaginez vous alors un vrai et sérieux projet de site web, tel que vous l'aviez fait au TP1 et mettez vous au travail.

- g) A chaque commit et push, vérifiez le déclenchement du pipeline et allez voir le rendu du site web ainsi déployé pour vérifier qu'il correspond bien au code source html de votre page web ; et au fur et à mesure, améliorez votre site web pour le rendre plus beau et professionnel.

NB : vous pourrez vous servir de Trello pour recenser et planifier l'ensemble de vos tâches de développement et savoir qui fait quoi et quelle tâche est en cours ou terminée. Recherchez sur le net de bons tutoriels qui vous expliquent comment utiliser faire du html si besoin (balises, formulaires, etc) et comment faire du css.