

# Intégration et déploiement continus de dépôts de documents multi-formats

Contact: Frédéric Herbreteau

## Contexte

Jekyll est un CMS (content management system) qui permet de construire un site web statique à partir de documents rédigés en format markdown. Il est notamment utilisé par GitHub Pages. Nous hébergeons notre propre instance à l'ENSEIRB-MATMECA pour les pages de cours (ex: logique et mathématiques discrètes).

Ces outils adossent Jekyll à un dépôt git. Chaque commit dans le dépôt lance la génération du site statique par Jekyll (via un hook git ou via un pipeline CI/CD) et son déploiement sur un serveur http.

## Problématique

De nombreux dépôts contiennent non seulement des pages markdown, mais aussi des supports rédigés dans d'autres formats (ex: LaTeX), ou compilés par des outils spécifiques (ex: Marp ou pandoc). Actuellement, ces supports ne sont pas compilés automatiquement. Ils doivent être générés puis déployés manuellement, à chaque modification. Ce manque d'automatisation est pénible et source d'erreurs.

## Objectifs

Ce projet a pour objectif d'automatiser la génération et le déploiement de l'ensemble des documents d'un dépôt git: pages markdown, documents LaTeX, etc. Pour simplifier, le prototype développé dans ce projet sera restreint aux sites Jekyll et aux documents LaTeX dans un premier temps. Mais l'étude portera également sur la généralisation des solutions proposées à d'autres formats de documents, ainsi qu'à des systèmes de génération et déploiement quelconques.

## Architecture minimale

Le point de départ consiste à recréer une architecture microservices à base de conteneurs permettant:

- d'héberger des dépôts git contenant des sites Jekyll et d'autres documents (ex: des sources LaTeX). On fera un minimum d'hypothèses sur le contenu de ces dépôts;
- de générer et déployer les sites web statiques via Jekyll à chaque commit sur les dépôts;
- de servir les sites web ainsi générées via un serveur http.



## Génération et déploiement de documents LaTeX

Il s'agira ensuite d'étendre cette architecture pour générer et déployer des fichiers PDF à partir des sources LaTeX présentes sur le dépôt, de manière automatique, à chaque commit.

- un fichier de configuration permettra de définir les sources LaTeX à compiler ainsi que le dossier de déploiement et les autres options nécessaires au fonctionnement du système. L'utilisation doit être la plus simple possible. En particulier, il doit être possible de définir des liens dans les pages markdown Jekyll, vers les documents PDF générés à partir des fichiers LaTeX, via des URL relatives.
- un service de compilation isolé (sandbox) permettra de générer des documents PDF à partir des sources LaTeX, et de déployer les documents PDF dans le dossier prédéterminé.
- la robustesse du système est un point clé. Une compilation LaTeX ne pas aboutir: le fichier source peut contenir des erreurs, et le compilateur peut s'arrêter en proposant une action à l'utilisateur. Par ailleurs, il devra être possible d'accéder au log produit par le compilateur LaTeX pour identifier d'éventuelles erreurs. Enfin, en cas d'erreur de compilation, les documents actuellement déployés ne doivent pas être écrasés.
- la souplesse du système est l'autre point clé. Il peut arriver que la compilation d'un document nécessite plusieurs phases (ex: génération d'images, etc). L'utilisateur devra donc pouvoir spécifier les phases de compilation des documents hébergés dans son dépôt git.
- la solution proposée s'appuyera sur une étude comparative de différentes solutions, notamment les approches CI/CD (ex: Jenkins) et git hook. La simplicité de la solution sera privilégiée, de même que l'utilisation de solutions logicielles existantes et éprouvées.

## Généralisation

La suite de l'étude portera sur la généralisation de la solution proposée à d'autres formats de documents, et d'autres systèmes de génération. La solution générale sera instanciée pour un autre système de compilation (ex: Marp ou pandoc) pour démontrer sa faisabilité.

## Restitution

S'agissant d'une étude de faisabilité, la solution proposée prendra la forme: - d'un prototype montrant la faisabilité et la pertinence des solutions retenues - de la documentation des solutions permettant leur mise en oeuvre sur un serveur de production