Qu'est-ce que GitLab ? Caractéristiques, avantages et comment commencer

Ce guide explore comment GitLab rationalise le développement, intègre CI/CD et améliore la collaboration.

Contenu Actualisé 14 mars 2025 · 15 min de lecture

Kurtis Pykes
Data Science & AI Blogger | Top 1000 Medium Writers on AI and Data Science

SUJETS

Git

La façon dont vous gérez vos projets joue un rôle essentiel dans le maintien de la productivité et la fourniture de solutions logicielles de haute qualité.

Imaginez un scénario dans lequel une équipe de développeurs travaille sur une application, apportant quotidiennement des mises à jour de code, mais sans assurer le cursus des modifications, la gestion des versions ou les processus de test et de déploiement. Il y a forcément un problème !

Ce scénario fictif ne risque pas de se produire de nos jours (je l'espère), car nous disposons d'outils tels que GitLab, dont nous allons parler dans ce billet de blog.

Nous aborderons les sujets suivants :

- Les principales fonctionnalités de GitLab
- Ses avantages
- Comment GitLab se compare-t-il aux autres plateformes d'hébergement Git ?

- Et comment commencer

Allons-y !

Qu'est-ce que GitLab ?

GitLab est une plateforme tout-en-un conçue pour le développement de logiciels et DevOps. Au départ, il s'agissait d'un gestionnaire de référentiel Git basé sur le web qui permettait aux équipes de collaborer sur le code. Cependant, il a depuis évolué pour devenir une solution entièrement intégrée pour la gestion de l'ensemble du cycle de vie du développement logiciel (SDLC).

Essentiellement, GitLab permet aux développeurs de :

- Gérer le contrôle des versions
- Automatiser les pipelines CI/CD
- Gérer les tâches du projet
- Contrôler les performances

Ces fonctionnalités sont disponibles au sein d'une interface unifiée, ce qui fait de GitLab un choix populaire pour rationaliser les processus de développement.

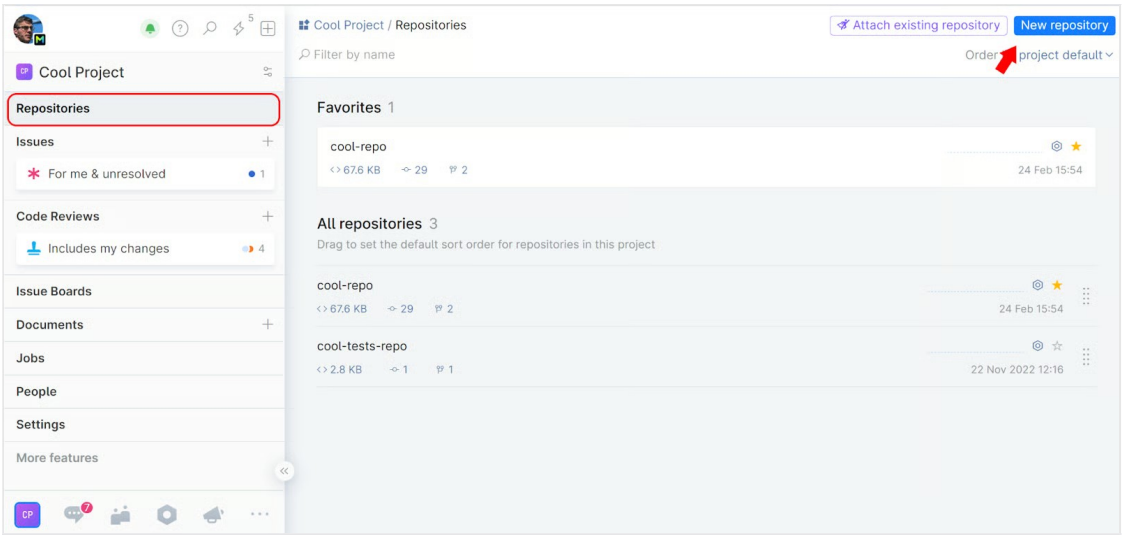
Fonctionnalités principales de GitLab

GitLab offre un large éventail de fonctionnalités qui s'adressent aux développeurs et aux organisations qui cherchent à rationaliser leurs flux de développement et d'opérations. Examinons quelques-unes des principales fonctionnalités de la plateforme.

Dépôts GitLab

Les dépôts GitLab servent de base à la gestion du code source basé sur Git. Les développeurs peuvent créer, cloner et gérer des référentiels qui prennent en charge les principales fonctions de contrôle de version (par exemple, les branches, l'historique des

livraisons et les demandes de fusion), ce qui aide les équipes à collaborer sur le code.



Exemple de référentiels GitLab ***Source :*** *Miroir d'un dépôt GitLab*

Intégration CI/CD

L'une des caractéristiques les plus remarquables de GitLab est son pipeline CI/CD intégré. Ces lignes de tuyaux CI/CDautomatisent les processus de test, de construction et de déploiement du code à l'aide de simples **fichiers de configuration YAML** qui définissent des pipelines pour automatiser ces tâches.

De plus, les pipelines de GitLab sont flexibles et s'intègrent à des outils populaires comme **Kubernetes et Docker** - ce qui simplifie le processus de gestion des déploiements dans le cloud et sur site.

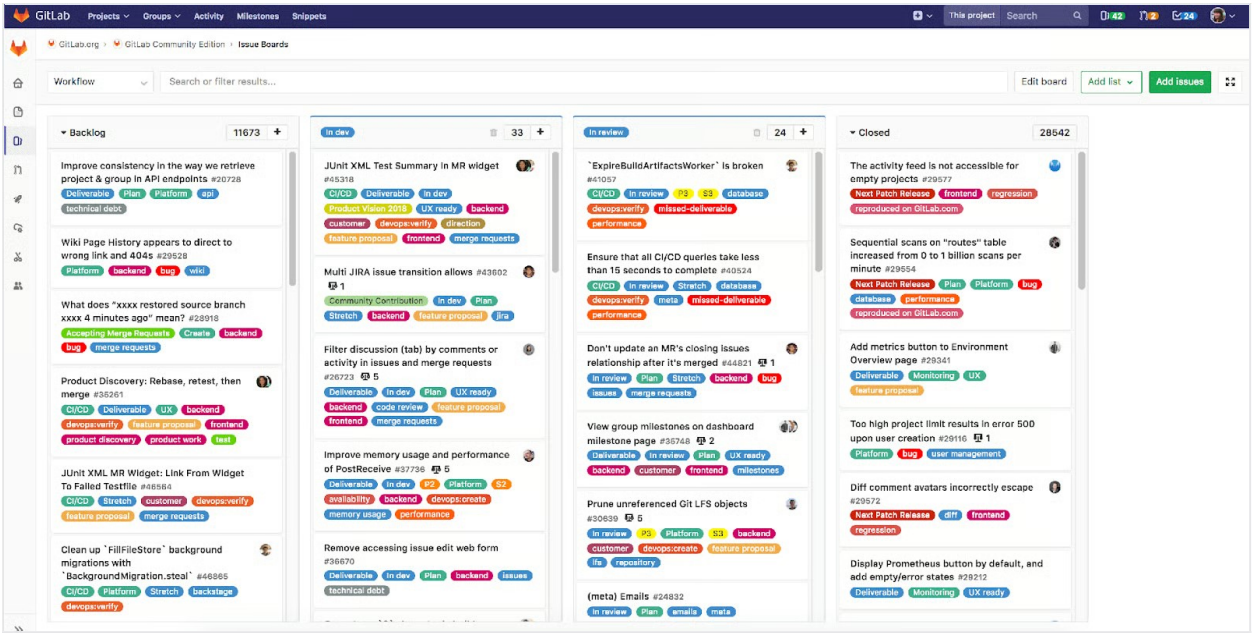
Quelle est la place de CI/CD dans votre flux de développement ? ***Alorsece :*** *Démarrez avec GitLab CI/CD*

GitLab Auto DevOps

Auto DevOps simplifie le processus CI/CD en proposant des modèles préconfigurés pour créer, tester et déployer des applications. Essentiellement, les développeurs peuvent automatiser des tâches avec un minimum d'installation.

Suivi des dossiers et gestion des projets

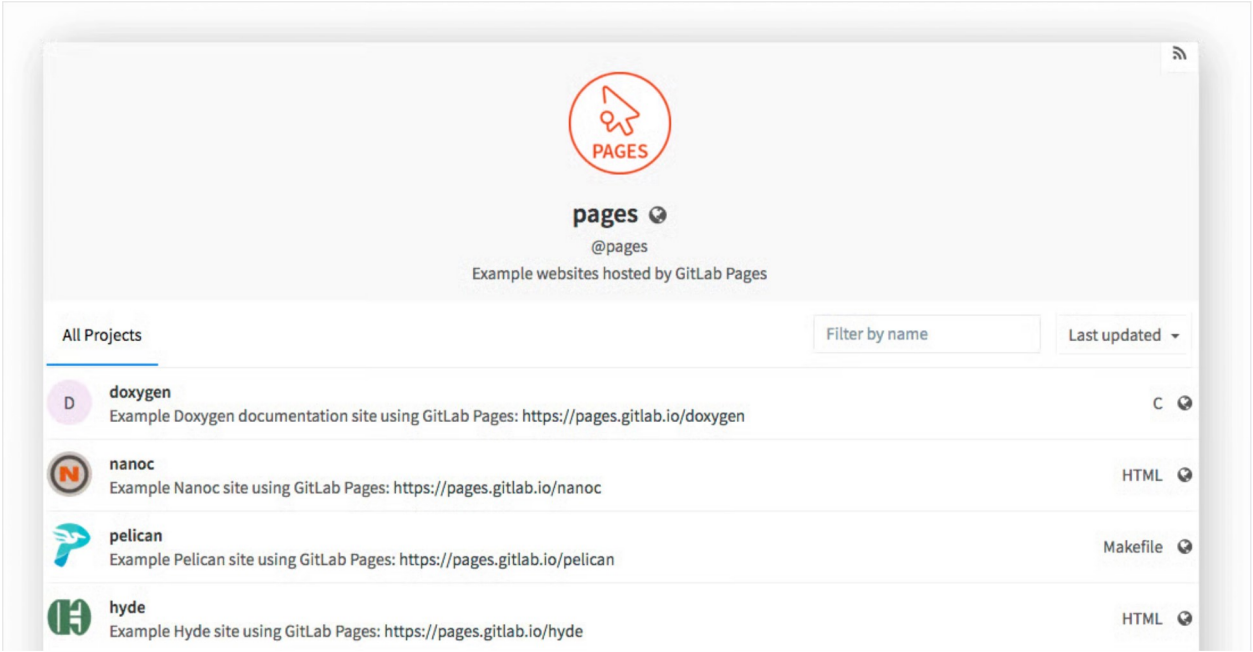
GitLab excelle également dans la gestion de projet, en fournissant des outils tels que des tableaux d'affichage des problèmes, des jalons, des étiquettes et des diagrammes d'évaluation pour soutenir les flux de travail agiles. Ces fonctionnalités permettent aux équipes de suivre l'évolution des problèmes depuis l'idéation jusqu'au déploiement, offrant ainsi une vue d'ensemble de la santé et du calendrier du projet.

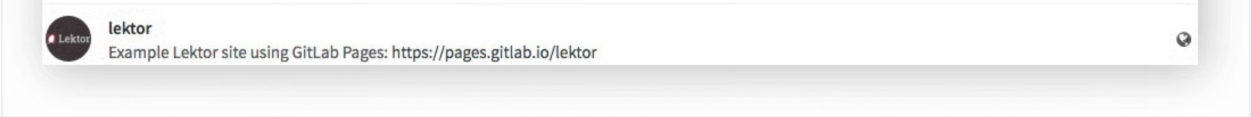


Un exemple de tableau d'affichage GitLab ***Source :*** *[4 façons d'utiliser les Issue Boards de GitLab]*

Pages GitLab

Pour les développeurs qui ont besoin d'une solution rapide et facile pour héberger des sites web statiques, GitLab Pages est la solution idéale. Vous pouvez déployer directement de la documentation, des sites de portfolio ou tout autre contenu web statique à partir de leur dépôt GitLab.





Ressources pour apprendre les pages GitLab ***Source :*** ***Commencez avec les pages GitLab***

Avantages de l'utilisation de GitLab

Qu'est-ce qui fait de GitLab un choix incontournable pour les équipes de développement ? Découvrons-le !

Plate-forme tout-en-un

L'un des principaux avantages de GitLab est sa nature intégrée. Vous n'avez pas besoin de passer d'un outil à l'autre pour effectuer le contrôle de version, le CI/CD, la gestion de projet et la surveillance de vos projets logiciels - tout est combiné dans une seule plateforme cohésive.

Collaboration et gestion d'équipe

GitLab a été conçu dans une optique de collaboration. Il vise à faciliter la communication et le travail d'équipe tout au long du processus de développement du logiciel. Dès le départ, GitLab a cherché à permettre aux développeurs et aux équipes de travailler ensemble plus efficacement en fournissant des outils qui rationalisent la collaboration et minimisent les goulots d'étranglement potentiels.

Évolutivité et flexibilité

GitLab est conçu pour s'adapter à votre équipe, que vous soyez un petit groupe de développement travaillant sur un projet secondaire ou une grande entreprise. La plateforme prend en charge les déploiements basés sur le cloud et auto-hébergés, ce qui signifie que les équipes peuvent adapter GitLab pour répondre à leurs besoins spécifiques en matière de sécurité et de performances.

Sécurité et conformité

Des fonctionnalités telles que l'analyse statique et dynamique du code, l'analyse des conteneurs et la gestion des vulnérabilités permettent aux utilisateurs de GitLab de s'assurer que leurs applications sont sécurisées dès le départ. Notez que GitLab intègre également les meilleures pratiques DevOps dans son écosystème pour s'assurer que la sécurité est un aspect important du processus de développement plutôt qu'une réflexion après coup.

GitLab vs GitHub vs Bitbucket

Les solutions alternatives à GitLab comprennent **GitHub** et **Bitbucket**. Lequel utiliser ?

Je répondrai à la manière typique d'un développeur : "Cela dépend. Il est essentiel de prendre en compte les exigences que vous attendez de votre outil de gestion du cycle de développement durable. Ceci étant dit, voici comment GitLab se positionne par rapport à ses concurrents.

GitLab vs GitHub

GitHub est une plateforme web de contrôle de version et de collaboration basée sur Git, un système populaire de contrôle de version distribué. Créé en 2008 par Tom Preston-Werner, Chris Wanstrath et PJ Hyett, GitHub a pour objectif premier de fournir une interface conviviale pour Git afin d'aider les développeurs à collaborer plus efficacement sur les projets.

D'abord populaire au sein de la communauté des logiciels libres, GitHub est devenu la plateforme de référence pour l'hébergement de dépôts publics et privés. Des millions de développeurs l'utilisent pour partager des bases de code et y contribuer dans le monde entier.

En ce qui concerne l'âge, GitHub a été lancé en 2008, trois ans avant GitLab, qui a été créé en 2011 par Dmitriy Zaporozhets et Valery Sizov. En termes de popularité, GitHub a traditionnellement bénéficié d'une base d'utilisateurs plus importante, en partie en raison de son entrée précoce sur le marché et de son adoption généralisée par la communauté des logiciels libres. Le vaste écosystème de dépôts publics de GitHub et son rôle important dans les projets open-source tels que Linux, Node.js et React ont encore renforcé sa popularité.

Bien que GitLab ait une base d'utilisateurs plus petite que GitHub, il a gagné une traction significative, en particulier parmi les équipes DevOps et les entreprises à la recherche d'une solution intégrée de développement et de déploiement. GitLab s'est taillé une place de choix en se concentrant sur la fourniture d'une plateforme DevSecOps tout-en-un, qui inclut des fonctions de CI/CD, de gestion de projet et de sécurité au sein d'un même écosystème. En revanche, GitHub s'appuie sur des intégrations tierces pour obtenir des fonctionnalités similaires, ce qui peut accroître la complexité des projets.

Si vous souhaitez en savoir plus sur GitHub, je vous recommande de commencer par le **cours GitHub Foundations**.

GitLab vs Bitbucket

Bitbucket est également un service d'hébergement de référentiels de contrôle de version basé sur le web, mais il a été créé à l'origine pour prendre en charge le système de contrôle de version Mercurial avant de passer entièrement à Git.

Comme GitHub, Bitbucket a été lancé en 2008 par Jesper Nøhr et a été racheté par Atlassian en 2010. Initialement, Bitbucket a été conçu pour fournir un environnement de

collaboration aux petites et grandes équipes, offrant une intégration profonde avec la suite d'outils d'Atlassian tels que Jira et Confluence. Ainsi, la plateforme est couramment utilisée par les équipes de développement au sein des organisations qui s'appuient sur l'écosystème Atlassian pour la gestion des projets et le cursus des problèmes.

En conséquence, GitLab a gagné une base d'utilisateurs plus large grâce à ses capacités DevOps tout-en-un. Bitbucket n'a peut-être pas la même popularité auprès du grand public que GitHub ou GitLab, principalement parce qu'il se concentre davantage sur les dépôts privés et les utilisateurs professionnels. Et malgré une bonne intégration avec Jira pour le cursus des projets, Bitbucket n'a pas la profondeur de GitLab en matière d'automatisation du cycle de vie du développement logiciel.

GitLab vs GitHub vs Bitbucket : Aperçu comparatif

Critères	GitLab	GitHub	Bitbucket
Année de lancement	2011	2008	2008
Les fondateurs	Dmitriy Zaporozhets, Valery Sizov	Tom Preston-Werner, Chris Wanstrath, PJ Hyett	Jesper Nøhr
Société mère	Indépendant (privé)	Microsoft (acquis en 2018)	Atlassian (acquis en 2010)
Objectif initial	Dépôt basé sur Git et plateforme DevOps intégrée	Hébergement d'un dépôt Git et collaboration	A l'origine, le logiciel Mercurial était utilisé, puis il a été remplacé par Git.