

GitLab AFB



<https://GitLab.afbiodiversite.fr/>

GitLab c'est quoi ?

Source Wikipédia :

GitLab est un logiciel libre de forge basé sur git proposant les fonctionnalités de wiki, un système de suivi des bugs, l'intégration continue et le déploiement en continue. Logiciel utilisé par plusieurs grandes entreprises informatiques incluant IBM, Sony, la NASA, Alibaba, Oracle, Boeing...

Source Journal du net :

GitLab est une plateforme de développement collaborative qui couvre l'ensemble des étapes du DevOps. Se basant sur les fonctionnalités du logiciel Git, elle permet de réaliser des dépôts et de gérer les versions de vos codes sources. Son usage est particulièrement indiqué pour les développeurs qui souhaitent disposer d'un outil réactif et accessible.

Les utilisations de GitLab

- Test de logiciels
- Configuration
- Monitoring
- Sécurité applicative
- Intégration et déploiement continu, etc.

Tutoriels GitLab

- Tutoriel sur GitLab proposé sur GitHub:
 - <https://github.com/SocialGouv/tutoriel-GitLab>
- Tutoriel pour utiliser GitLab:
 - <https://GitLab.com/jnaour/tutoriel-GitLab/wikis/tutoriel-GitLab>
- L'intégration continue avec GitLab:
 - <https://www.grafikart.fr/tutoriels/GitLab-ci-docker-808>

Redmine vs GitLab

Organisation et gestion des projets

- **Redmine** propose de gérer une arborescence de projets : une relation de type père-fils peut exister entre les projets, avec héritage optionnel des membres, tickets, versions. Toutefois, les identifiants des projets sont uniques sur l'intégralité de tous les projets et sous-projets.
- **GitLab** propose de gérer des projets par « espace de nommage », et son identifiés par « espace-de-nommage/nom-du-projet ». Chaque utilisateur et chaque groupe possède son propre espace de nommage, et les projets qu'il contient. Un utilisateur ne peut créer de projet que dans son espace personnel ou dans celui d'un groupe pour lequel il a la permission de le faire.

Redmine vs GitLab

Visibilité des projets

Redmine ne propose que deux niveaux de visibilité:

- **privé** : seuls les membres du projet peuvent le voir
- **public** : toutes les personnes connectées à la forge ont accès à une partie du projet

GitLab propose trois niveaux de visibilité :

- **privé** : seuls les membres du projet peuvent le voir
- **interne** : toutes les personnes connectées à la GitLab ont accès à une partie du projet
- **public** : tout le monde a accès à une partie du projet.

Redmine vs GitLab

Gestion des tickets

- **Redmine** propose un système de tickets un peu plus avancé que GitLab, avec la possibilité d'avoir plusieurs workflows configurables. Ces workflows sont communs à tous les projets, ce qui limite leur personnalisation.
- **GitLab** propose un système de tickets plus simplifié avec des tags. Il est toutefois possible d'utiliser un gestionnaire de tickets externe (Mantis, Jira, Redmine, ...) si nécessaire.

Redmine vs GitLab

Contrôle des sources et gestion des dépôts

- **Redmine** supporte Subversion, Darcs, Mercurial, CVS, Bazaar et Git, et plusieurs dépôts par projets mais tous les dépôts doivent avoir un identifiant unique. Redmine n'assure ni la gestion ni le contrôle d'accès aux dépôts, ce doit être délégué à des logiciels tiers.
- **GitLab** supporte exclusivement Git, mais il est parfaitement intégré : la création, la gestion et le contrôle d'accès des dépôts sont intégrés et automatisés. Toutefois, un projet n'a qu'un et un seul dépôt Git.

Les Plus de Redmine

- **Maitrise de l'outil en interne à l'AFB**
- **Internationalisation**
Redmine est traduit dans plusieurs langues, dont le français.
- **Champs personnalisés**
Redmine permet de définir des champs personnalisés qui pourront être affectés aux projets, tickets, ... Le type et les valeurs valides de chaque champ sont configurables. Il est ensuite possible de faire de recherches sur ces champs.

Les Plus de GitLab

- Plus orienté DevOps
- De nombreuses fonctionnalités par défaut
- La possibilité de trier les issues par « milestones » (jalons) afin de gérer les futures livraisons
- L'intégration continue teste votre code en instantané quand vous effectuez des « commit » et le déploiement en continue livre ce code si les tests sont bons.

Se connecter à GitLab

AFBiodiversité	Standard
AFBiodiversité Username	
<input type="text"/>	
Password	
<input type="password"/>	
<input type="checkbox"/> Remember me	
<input type="button" value="Sign in"/>	

Connexion via LDAP

Les utilisateurs dans le LDAP de l'AFB peuvent s'authentifier directement dans le GitLab avec leur compte AFB (le champ « AFBiodiversité Username » est l'adresse mail utilisé dans le LDAP de l'AFB).

Lors de la 1^{ère} connexion réussie au site gitlab.afbiodiversite.fr un compte est créé automatiquement sur le GitLab de l'AFB.

Comptes externes à l'AFB

Les utilisateurs qui ne sont pas dans le LDAP de l'AFB doivent être créés manuellement par un administrateur.

Les comptes non *@afbiodiversite.fr* sont créés en tant qu'« External » (ils peuvent être transformés en internes si le besoin est justifié).

Un compte **externe** ne voit que les projets publiques et les projets privés ou internes où il est affecté. Il ne peut pas créer de projets ou de groupes.

Un compte **interne** voit les projets publiques et internes et les projets privés où il est affecté.

Un utilisateur **non authentifié** ne voit que les projets publics.

Interface de GitLab

The screenshot shows the GitLab 'Projets' (Projects) page. The navigation bar at the top includes the GitLab logo, 'Projets' (selected), 'Groupes', 'Activité', 'Jalons', 'Extraits de code', and utility icons. A search bar is present with the text 'Rechercher ou aller à...'. Below the navigation bar, the page title 'Projets' is followed by a 'Nouveau projet' button. The main content area shows a list of projects under the 'Tous' tab. The projects listed are:

Project ID	Project Name	Role	Stars	Forks	Issues	Commits	Updated
1	SG / DSI / SPED / 1er projet SPED - TEST	Owner	0	0	0	0	Updated il y a une semaine
P	BOUDET Christian / Premier Projet AFB	Maintainer	0	0	0	2	Updated il y a une semaine
C	DUVEAU Gerald / cahiers-techniques	Maintainer	0	0	0	0	Updated il y a 2 mois
F	BOUDET Christian / First Project Premier projet test	Maintainer	0	0	0	0	Updated il y a 2 mois

La notion de Projet

- Créer un projet, c'est créer un espace public ou privé où vous pouvez versionner des fichiers et collaborer.
- Lorsque vous créez un nouveau projet, il peut résider :
 - soit dans votre espace personnel : [https://\[URL\]/prenom.nom/nom-du-projet](https://[URL]/prenom.nom/nom-du-projet)
 - soit dans un groupe existant : [https://\[URL\]/groupe/nom-du-projet](https://[URL]/groupe/nom-du-projet)

Visibilité d'un projet

Privé	Vous serez le seul à pouvoir accéder au projet et à donner des permissions à l'unité
Interne	Seuls les utilisateurs authentifiés peuvent voir le projet
Public	N'importe qui, même non authentifié peut voir le projet

Si le projet est privé et que vous souhaitez que des utilisateurs collaborent vous devez préalablement les inviter à rejoindre le projet.

Les différents rôles

Guest	Lire et commenter les tickets (issues)
Reporter	Gérer les tickets (issues), voir le code
Developer	Modifier les fichiers, les tickets (issues)
Maintainer	Responsables. Peuvent ajouter des utilisateurs et sous-groupes à un projet
Owner	Essentiellement administrateur d'un groupe (création/destruction)

Pour une vue plus détaillée : <https://docs.gitlab.com/ee/user/permissions.html>

Groupe AFB

Le groupe AFB rassemble les projets de l'Agence Française pour la Biodiversité.

Sauf besoin explicite (*ex: projets européens*), tout nouveau projet doit être créé dans le groupe AFB.

Ce groupe est à visibilité publique, ce qui signifie que tout le monde peut voir les projets publics qui sont dans ce groupe. Cependant, les projets sont par défaut créés avec une visibilité Interne.

Seuls les *Maintainers* du groupe AFB peuvent créer des sous-groupes dans le groupe AFB.

Les *Owners* et *Maintainers* des sous-groupes de AFB peuvent créer d'autres sous-groupes de niveaux inférieurs.

Il est possible d'ajouter des membres à un groupe. Les droits de ces membres se répercuteront sur les sous-groupes et projets de ce groupe.

Création d'un projet

Créer un projet « test »

Les projets de tests sont à créer dans son espace utilisateur (groupe nominatif), **attention, les projets opérationnels ne doivent pas être créés dans cet espace.**

Par défaut, les utilisateurs internes peuvent créer des projets dans leur espace utilisateur (groupe nominatif).

ex: Création d'un projet dans mon espace utilisateur □ <https://gitlab.afbiodiversite.fr/christian.boudet/test>

Créer un projet rattaché à une thématique

Les projets rattachés à une thématique (ex: SI-Geol, BSS, ADES...) sont à créer dans le groupe correspondant.

Seuls les utilisateurs ayant les droits *Maintainer* ou *Owner* peuvent créer des projets et/ou sous-groupes (correspondant la plupart du temps aux Chef de Projets AFB).

Créer un projet opérationnel « isolé » (ou créer un groupe dans AFB)

Les projets opérationnels « isolés », c'est-à-dire n'étant pas rattachés à un groupe particulier, doivent être créés dans le groupe AFB.

Seuls les *Maintainers* du groupe AFB (et les administrateurs) peuvent créer des projets dans le groupe AFB.

Création d'un ticket (issue)

Comment créer un ticket ?

Tous les utilisateurs voyant un projet peuvent ajouter des tickets à celui-ci, cependant seuls les personnes ayant des droits au minimum Reporter peuvent assigner des tickets à des utilisateurs, à des jalons, ajouter des labels ou une échéance.

Nouveau ticket

Titre

Add [description templates](#) to help your contributors communicate effectively!

Description

Write Aperçu B I " <> @ ☰ ☷ 📎 🗑

Write a comment or drag your files here...

Assignee [Assign to me](#) **Due date**

Milestone

Labels

Une façon commune de travailler

Par défaut, seuls les *Owners* et *Maintainers* peuvent écrire dans certaines branches protégées (dont la branche « master »).

Donner les droits d'écriture à tout le monde dans la branche « Master »:

Cette solution, bien que possible, n'est pas la meilleure. En effet, des pipelines de CI/CD peuvent exister sur le Master qui seraient alors exécutés à chaque *push*. De plus, les modifications seront moins facilement identifiables (rattachables à un ticket) dans l'historique des versions.

□ *Acceptable si le projet est assez peu collaboratif (i.e. assez peu d'intervenants).*

Fork

Le Fork consiste à faire une copie du projet dans son espace utilisateur et à travailler sur cette version. On peut par la suite faire une demande de fusion des deux branches (un « merge request ») pour ajouter les modifications réalisées dans la branche « master ».

□ *les externes ne peuvent pas créer de projets dans leur espace perso, et la multiplication des projets pourrait rendre le GitLab plus difficile à maintenir.*

Branche

La branche ressemble fortement au fork si ce n'est qu'au lieu de créer un nouveau projet, on crée seulement une branche dans le projet (comme ce qui pouvait se faire auparavant dans Redmine). De la même façon, une fois les modifications satisfaisantes, une demande de Merge Request est faite.

□ *Solution préconisée pour les projets AFB*

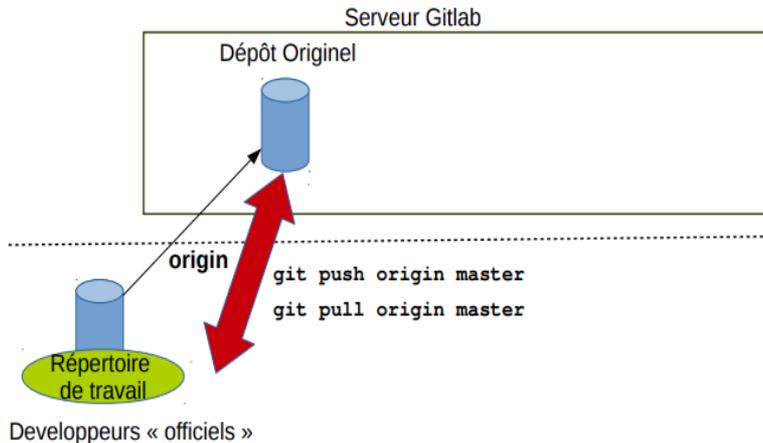
Merge Request

Demande pour merger une branche vers une autre. Cette demande doit être revue par un *Owner* ou un *Maintener*. Ceci permet de conserver une trace de ce qui a été fait et le cas échéant d'effectuer une revue de code.

Principes du Fork et Merge Request

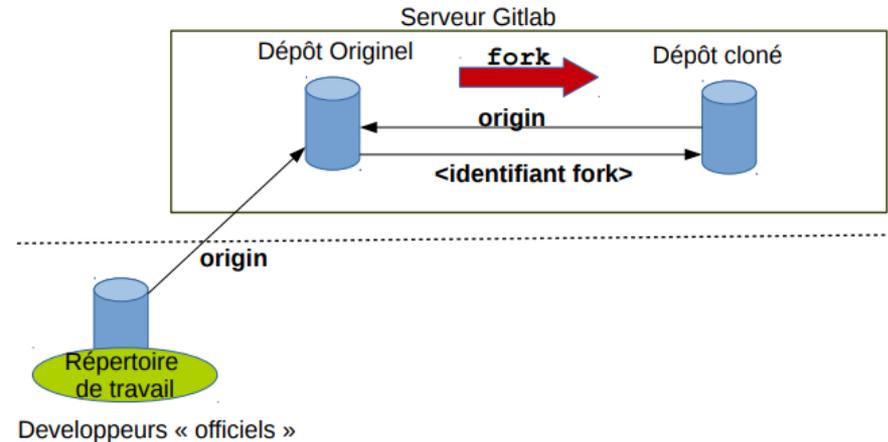
fork et merge request

- Situation initiale



forker

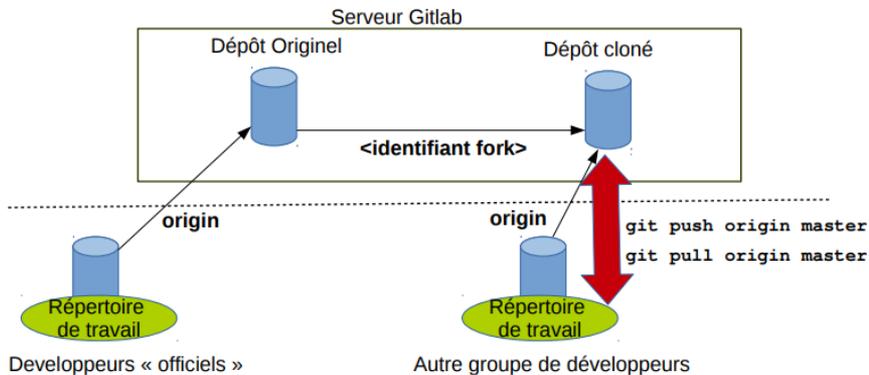
- Cloner un dépôt dans un autre *workspace*



Principes du Fork et Merge Request

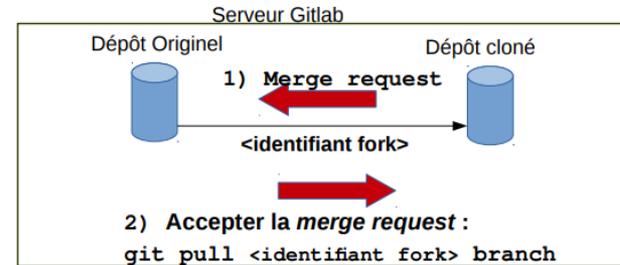
forker

- Développeurs extérieurs travaillent de manière isolée



merge request

- Proposition de **fusion de branches** depuis dépôt cloné vers dépôt originel
 - Développeurs du dépôt cloné proposent la fusion d'une branche.
 - Développeurs du dépôt originel décident si la fusion est réalisée.



Travailler sur un dépôt

Créer une branche (via GitLab)

Le nom de la nouvelle branche est par défaut constituée du nom du développeur suivi du nom de la fonctionnalité

BOUDET Christian > Test > Details

Test Identifiant de projet : 19

Mettre une étoile 0 Créer une divergence 0 Clone

Ajouter une licence 1 commit 1 branche 0 étiquette 133 ko fichiers

master test /

- Ce répertoire
- Nouveau fichier
- Téléverser un fichier
- Nouveau dossier
- Ce dépôt
- Nouvelle branche**
- Nouvelle étiquette

Ajout de README.md
BOUDET Christian a créé

ec9aa539

Historique Rechercher un fichier EDI Web

README Auto DevOp Ajouter un CONTRIBUTING

Nom	Dernière mise à jour
README.md	Ajout de README.md à l'instant

BOUDET Christian > Test > New Branch

New Branch

Branch name

Create from

Existing branch name, tag, or commit SHA

Sources:

gitlab.brgm.fr - présentation.pptx

https://romainlebreton.github.io/ProgWeb-CoteServeur/assets/initiation%20GIT_IUT.pdf

<https://blog.axopen.com/2017/02/gitlab-cest-quoi/>

<https://github.com/SocialGouv/tutoriel-GitLab#cr%C3%A9er-un-projet>