

Introduction à Gitlab

Table des matières

I. Présentation de GitLab	3
II. Exercice : Quiz	7
III. Installation et visualisation du fonctionnement	8
IV. Exercice : Quiz	13
V. Essentiel	15
VI. Auto-évaluation	15
A. Exercice	15
B. Test	15
Solutions des exercices	16

I. Présentation de GitLab

Contexte

La nécessité de faire évoluer du code et garder une surveillance de ce dernier à créer un besoin de créer des versions de ce dernier, mais aussi un besoin de faciliter le travail en équipe sur du code. C'est à partir de ces besoins qu'est sorti l'outil Git devenu une norme pour les développeurs.

Dans une majorité des cas, les versions de code sont mises en ligne avec Git sur le site / hébergeur GitHub mais il existe d'autres solutions répondant à ce service.

GitLab est donc l'une de ces solutions pouvant remplacer GitHub de façon totalement ou partiellement gratuite et beaucoup plus ouverte.

L'avantage de ce dernier étant qu'il s'agit d'une solution particulièrement complète pouvant être hébergée de façon autonome, ou via les concepteurs de celui-ci.

Nous verrons ce qu'est GitLab, son histoire, ses avantages et inconvénients vis-à-vis de ces concurrents et les spécificités de son offre.

Nous verrons aussi son installation via ses 2 méthodes principales, son interface et ce qu'elle propose.

Rappel Git

Dans l'environnement d'un développeur, un outil indispensable est celui de gestion de version du code, parmi les solutions, est facilement trouvable Git, pilier de ces outils, il existe depuis 2005 et est aujourd'hui la solution de référence.

L'utilisation de gestion de version apporte une sécurité et une facilité au niveau du code, cependant pour optimiser cette dernière, il est primordial de pouvoir déployer les versions sur un serveur d'hébergement.

Ces solutions hébergeant les différentes versions du code sont appelées forge.

Ainsi, il en existe de nombreuses avec pour chacune des spécificités propres, GitLab est l'une de ces solutions.

Existant depuis octobre 2011 pour sa première version, GitLab est une solution logicielle Open Source supportant git et permettant de réaliser du déploiement continu, du suivi de bug, la création d'un wiki et d'autres fonctionnalités grâce à des extensions externes.

Développé en Ruby sous licence MIT, il existe sous forme de site hébergeant comme GitHub ou sous forme d'application installable sur serveur.

La solution est par exemple utilisée par des grands groupes comme : NVidia, Drupal, Siemens, Goldman Sachs ou même ticketmaster.

GitLab met par ailleurs en avant toute ces fonctionnalités dès la première page de son site comme visible sur la capture d'écran ci-dessous :

 Manage	 Plan	 Create	 Verify	 Package	 Secure	 Release	 Configure	 Monitor	 Protect
Subgroups	Team Planning	Source Code Management	Continuous Integration (CI)	Package Registry	SAST	Continuous Delivery	Auto DevOps	Metrics	Container Scanning
Audit Events	Portfolio Management	Code Review	Code Testing and Coverage	Container Registry	Secret Detection	Pages	Kubernetes Management	Incident Management	Security Orchestration
Audit Reports	Service Desk	Wiki	Performance Testing	Helm Chart Registry	Code Quality	Advanced Deployments	Deployment Management	On-call Schedule Management	
Compliance Management	Requirements Management	Web IDE	Merge Trains	Dependency Proxy	DAST	Feature Flags	ChatOps	Tracing	
DevOps Reports	Quality Management	Snippets	Review Apps	Git LFS	API Security	Release Evidence	Infrastructure as Code	Error Tracking	
Value Stream Management	Design Management		Secrets Management		Fuzz Testing	Release Orchestration		Product Analytics	
					Dependency Scanning	Environment Management			
					License Compliance				
					Vulnerability Management				

Capture d'écran des fonctionnalités GitLab mis en avant sur le site officiel

Remarque

En plus d'être particulièrement utilisé de façon direct, GitLab a gagné en popularité grâce à des instances comme celle de Gnome et KDE, bien connu des utilisateurs linux ou même celle de Blender.

Instance en ligne ou privé

Un point important avec GitLab est qu'il s'agit d'une solution logicielle et non d'un service web contrairement, par exemple, à GitHub.

Cela signifie en d'autres termes, qu'il est possible d'héberger les projets sur le site gitlab.com, mais aussi d'auto-héberger sa solution de Forge. Ainsi les développeurs peuvent héberger leur instances GitLab sur leur propre serveur afin de ne pas dépendre d'un service extérieur.

Cette instance peut alors s'exécuter sous différente forme, en natif sur différent support Linux (Ubuntu, Debian, AlmaLinux, CentOS, OpenSuse), en natif sur Raspberry Pi OS, via un container Docker ou Kubernetes, ou même grâce à l'un des packages pré-crée pour les solutions de cloud : AWS, Google Cloud ou Azure.

Remarque

Il reste important de spécifier que GitLab utilise beaucoup de ressources sur un serveur, en tant que conteneur, par exemple, il sera donc nécessaire de prévoir une base hardware suffisamment solide si d'autres services tournent avec et partagent ces mêmes ressources.

Coût de GitLab

Évidemment, même sous licence MIT, GitLab peut être payant, cela permet de maintenir la société derrière le projet.

GitLab supporte alors gratuitement les projets Open Source sous licence MIT jusqu'à 5 GB de stockage, 10 GB de transfert par mois, 400 minutes de déploiement continu par mois et 5 utilisateurs par espace.

Au-delà, ou dans le cas de projet privé, GitLab propose 2 solutions :

- La Premium : incluant des reviews de code, des options avancées de déploiement continu, la gestion de planning d'entreprise agile, des contrôles accrus, un support et 10 000 minutes (soit presque 1 semaine non-stop) de déploiement continue par mois.
- La Ultimate : incluant les services Premium et des options de sécurité avancée, des options de gestion accrue, un management de portfolio, des utilisateurs invités gratuitement, un support avancé et 50 000 minutes (soit 1 mois non-stop environ) de déploiement continue par mois.

Complément

La Premium et la Ultimate coûtent respectivement, actuellement, environ 20€ et 100€ par mois, comme visible sur la capture ci-dessous.

Get the DevOps Platform

The screenshot displays the GitLab pricing page with three distinct columns for different service tiers. Each column has a header with a small icon, a title, a description, a price, and a list of features. The 'Free' tier is for individual users, 'Premium' is for team productivity, and 'Ultimate' is for organization-wide security and compliance. The 'Premium' and 'Ultimate' tiers are billed annually.

Tier	Price (per user/month)	Annual Billing (USD)
Free	\$0	-
Premium	\$19	\$228
Ultimate	\$99	\$1188

Free Tier Features:

- Spans the DevOps lifecycle
- Open Source - MIT License
- Includes free static websites
- 5GB storage
- 10GB transfer per month
- 400 CI/CD minutes per month
- 5 users per namespace

Premium Tier Features:

- Faster code reviews
- Advanced CI/CD
- Enterprise agile planning
- Release controls
- Self-managed reliability
- 10,000 CI/CD minutes per month
- Support

Ultimate Tier Features:

- Advanced security testing
- Security risk mitigation
- Compliance
- Portfolio management
- Value stream management
- Free guest users
- 50,000 CI/CD minutes per month
- Support

Capture d'écran des forfaits GitLab

À cela, GitLab propose aussi au paiement des crédits de temps de déploiement continue avec 1 000 minutes pour 10\$ et des crédits de stockage avec 10 GB pour 60\$.

Pour les versions auto-hébergées, les options restent assez similaires, en laissant une meilleure liberté pour les Repositories privés. La version auto-hébergée existe principalement en Edition Entreprise (EE) ou en Édition Communautaire (CE). La plupart de ces fonctionnalités présentes ou non peuvent être vues sur le site officiel à cette adresse : [Gitlab Features¹](https://about.gitlab.com/features/) grâce au bouton « *Filter by Plan & Hosting set-up* » en sélectionnant l'option Self-Hosted.

Comparaison avec les différents acteurs

GitLab n'est évidemment pas le seul service à fournir une Forge pour les repositories Git.

Le premier auquel pense la majorité des développeurs est GitHub, détenu par Microsoft mais il existe aussi :

- Azure Repos (aussi propriété de Microsoft),
- GitBucket (détenu par Alassian),
- Code Commit (une solution Amazon via AWS),
- SourceForge (propriété de Slashdot),
- Source Repositories (une solution Google Cloud),
- Gitea (Une solution Auto-hébergée sous licence MIT),
- Et énormément d'autres.

Il est donc important de pouvoir connaître les forces et faiblesses de GitLab pour sélectionner la forge la plus adaptée à un projet :

	Azure	GitBucket	Gitea	GitHub	GitLab
Open Source	Non	Oui	Oui	Non	Oui
Gratuité projet Open S.	Oui	Oui	Oui	Oui	Oui
Gratuité projet Privé	Oui	Oui	Oui	Non	Oui en SH
Fonctions Payantes	Oui	Oui	Non	Oui	Oui
Propose des options de CI/CD natifs	Oui	Oui	Non	Oui	Oui
Gestion des tests	Oui	Oui	Non	Oui	Oui
Hébergement	C	C / DC / SH	SH	DC	C / SH
Gestion de projet	Oui	Via Trello et Jira	Non	Oui	Oui
Gestion de ticket	Oui	Oui	Oui	Oui	Oui
	Complet	Complet	Légèreté	Complet	Complet

Les méthodes d'hébergement correspondent à :

- Cloud pour le C (Hébergé par la société propriétaire de la solution)
- Centre de donnée ou Data Center pour le DC (Hébergé par la société propriétaire de la solution)
- Auto Hébergé ou Self-Hosted pour le SH

¹ <https://about.gitlab.com/features/>

Ainsi avec GitBucket, GitLab est la solution Open Source la plus complète pouvant être auto-hébergée, cependant contrairement à ce dernier qui se base principalement sur sa solution Cloud, GitLab priorise énormément l'auto-hébergement.

Exercice : Quiz

[solution n°1 p.17]

Question 1

Comment s'appelle un hébergeur de Repositories en ligne ?

- ☐ Une Forge
- ☐ Une Herber
- ☐ Un FrauGit
- ☐ Un GitterLine

Question 2

GitLab est une solution :

- ☐ Propriétaire
- ☐ Open Source

Question 3

GitLab est gratuit dans tous les cas.

- ☐ Faux
- ☐ Vrai

Question 4

GitHub et GitLab sont des solutions dont le principal avantage est d'être :

- ☐ Complète
- ☐ Légère
- ☐ Gratuite

Question 5

GitLab peut être utilisé :

- ☐ En auto-hébergé
- ☐ En cloud
- ☐ En Datacenter pré-hébergé

III. Installation et visualisation du fonctionnement

GitLab peut donc être utilisé comme service en ligne, hébergé par la société GitLab, à la façon de GitHub ou peut être installé sur une machine ou un serveur dont l'utilisateur est propriétaire.

Ainsi une entreprise peut déployer une instance GitLab uniquement accessible sur son réseau pour ne pas déployer de code en ligne.

De même, une société voulant laisser la possibilité à ces utilisateurs de créer, par exemple, des extensions ouvertes pour son système, pourra créer une instance GitLab dédiés à cela.

Pour l'installation sur une machine locale, il existe différentes méthodes, dépendant principalement :

- Du système d'exploitation
- Des ressources allouées
- De la politique de déploiement envisagée

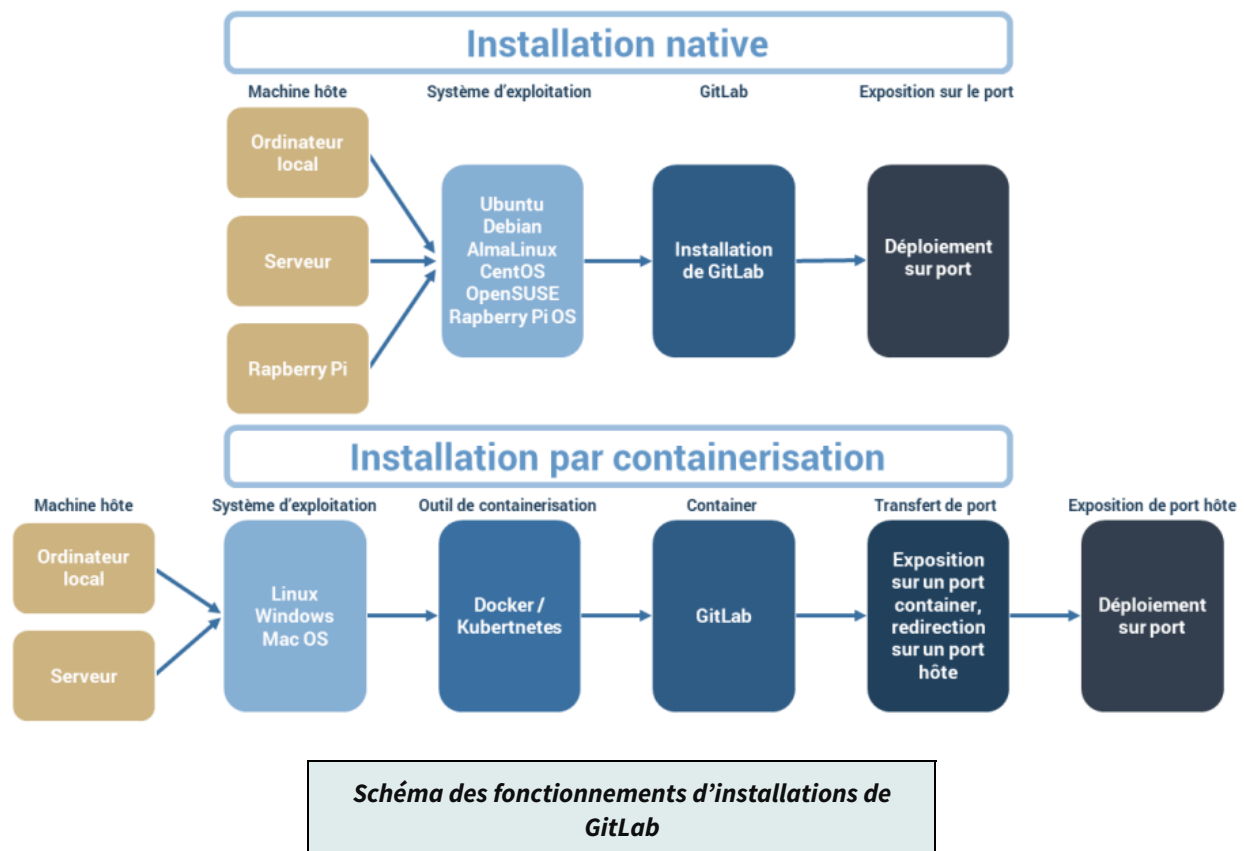
Ainsi, 2 méthodes ressortent principalement du lot :

- L'installation native, consistant à installer l'instance GitLab directement sur la machine,
- L'installation en container : consistant à installer GitLab dans un container docker ou kubernetes.

Pour la première méthode, seule une connaissance de l'OS de la machine hôte et un OS compatible sera nécessaire.

Pour la seconde méthode, la grande majorité des OS seront compatibles (dès lors qu'elle accepte la containerisation), cependant il sera préférable d'avoir des bases dans le domaine.

La différence de fonctionnement entre ces 2 méthodes peut être schématisé ainsi :



Méthode

Pour l'installation en containerisation, la méthode la plus simple est de passer par un fichier de composition docker, comme ci-dessous :

```

1 version: '3.6'
2 services:
3   web:
4     image: 'gitlab/gitlab-ee:latest'
5     restart: always
6     #hostname: '<adresse du server>.com'
7     #environment:
8       #GITLAB_OMNIBUS_CONFIG: |
9         #external_url 'https://<adresse du server>.com'
10    ports:
11      - '80:80' #Port HTTP
12      - '443:443' #Port HTTPS
13      - '22:22' #Port SSH
14    volumes:
15      - 'gitlab_config:/etc/gitlab'
16      - 'gitlab_logs:/var/log/gitlab'
17      - 'gitlab_data:/var/opt/gitlab'
18    shm_size: '256m'
19 volumes:
20   gitlab_config:
21   gitlab_logs:
22   gitlab_data:
23

```

Dans un fichier nommé 'docker-compose.yml'

À partir de cette configuration, GitLab sera le seul service déployé via l'adresse web, car ce dernier occupe les ports web par défaut (80 et 443), il sera tout de même possible de mettre des services sur d'autres ports.

Il est ensuite possible de créer le container en positionnant le terminal dans le dossier où se trouve le fichier et en renseignant la commande suivante :

```
1 $ sudo docker-compose up -d
```

Docker télécharge les images composant GitLab et compose le container avec les paramètres fournis précédemment comme sur la capture ci-dessous :

```

% sudo docker-compose up -d
[+] Running 9/9
  :: web Pulled                                98.2s
  :: e0b25ef51634 Pull complete                 3.9s
  :: 301fc011b74f Pull complete                 4.6s
  :: be4829809fe6 Pull complete                 4.7s
  :: a058cc0167e4 Pull complete                 4.8s
  :: 6df76736e784 Pull complete                 4.9s
  :: 6d214b17c866 Pull complete                 5.0s
  :: 2a89b6298276 Pull complete                 5.1s
  :: 55ff3103524c Pull complete                 96.2s
[+] Running 5/5
  :: Network gitlab_default                    Created          0.1s
  :: Volume "gitlab_gitlab_data"               Created           0.0s
  :: Volume "gitlab_gitlab_config"             Creat...         0.0s
  :: Volume "gitlab_gitlab_logs"               Created           0.0s
  :: Container gitlab-web-1                     Started           2.1s

```

**Capture d'écran Terminal Mac OS docker-compose
GitLab****Remarque**

GitLab est installé ici avec le docker-compose précédent via la version Enterprise (ee), cependant il est possible d'installer la version community(ce) en remplaçant la valeur gitlab/gitlab-ee:latest par gitlab/gitlab-ce:latest.

Exemple

Pour l'installation sur un système comme Debian, il faudra dans un premier temps se positionner sur le terminal, puis installer les dépendances nécessaires :

```
1 $ sudo apt-get update && apt-get install -y curl openssh-server ca-certificates perl
```

Cette ligne de commande mettra à jour les Repositories de Debian et installera curl, openssh-server, ca-certificates et perl.

À ce stade, il est possible d'ajouter le package 'postfix' à l'installation pour utiliser le serveur d'envoi et de réception de mail par défaut avec GitLab.

Il est ensuite nécessaire de rajouter le Repository GitLab via la commande curl suivante :

```
1 $ curl https://packages.gitlab.com/install/repositories/gitlab/gitlab-ee/script.deb.sh | sudo bash
```

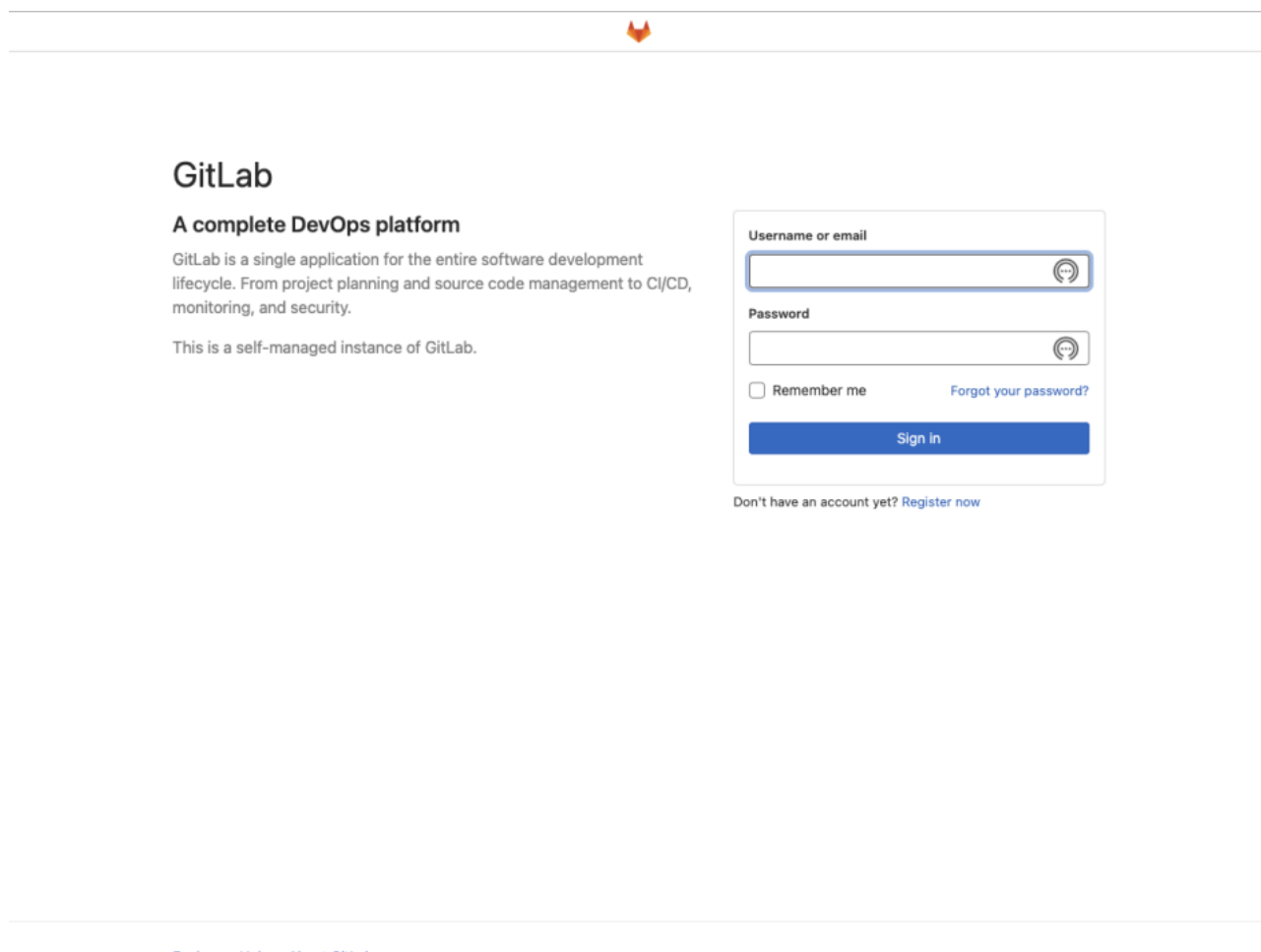
L'installation de GitLab peut alors être effectuée avec la commande suivante :

```
1 $ sudo EXTERNAL_URL="https://gitlab.example.com" apt-get install gitlab-ee
```

Cette commande installera GitLab en version enterprise et configurera l'adresse externe vers l'adresse <https://gitlab.example.com/>¹

¹ <https://gitlab.example.com/>

À la suite de cela, la page déployée sur l'adresse du serveur sera celle présente ci-dessous :



GitLab

A complete DevOps platform

GitLab is a single application for the entire software development lifecycle. From project planning and source code management to CI/CD, monitoring, and security.

This is a self-managed instance of GitLab.

Username or email

Password

☐ Remember me [Forgot your password?](#)

[Sign in](#)

Don't have an account yet? [Register now](#)

Capture d'écran première page d'une instance GitLab

Les identifiants par défaut sont :

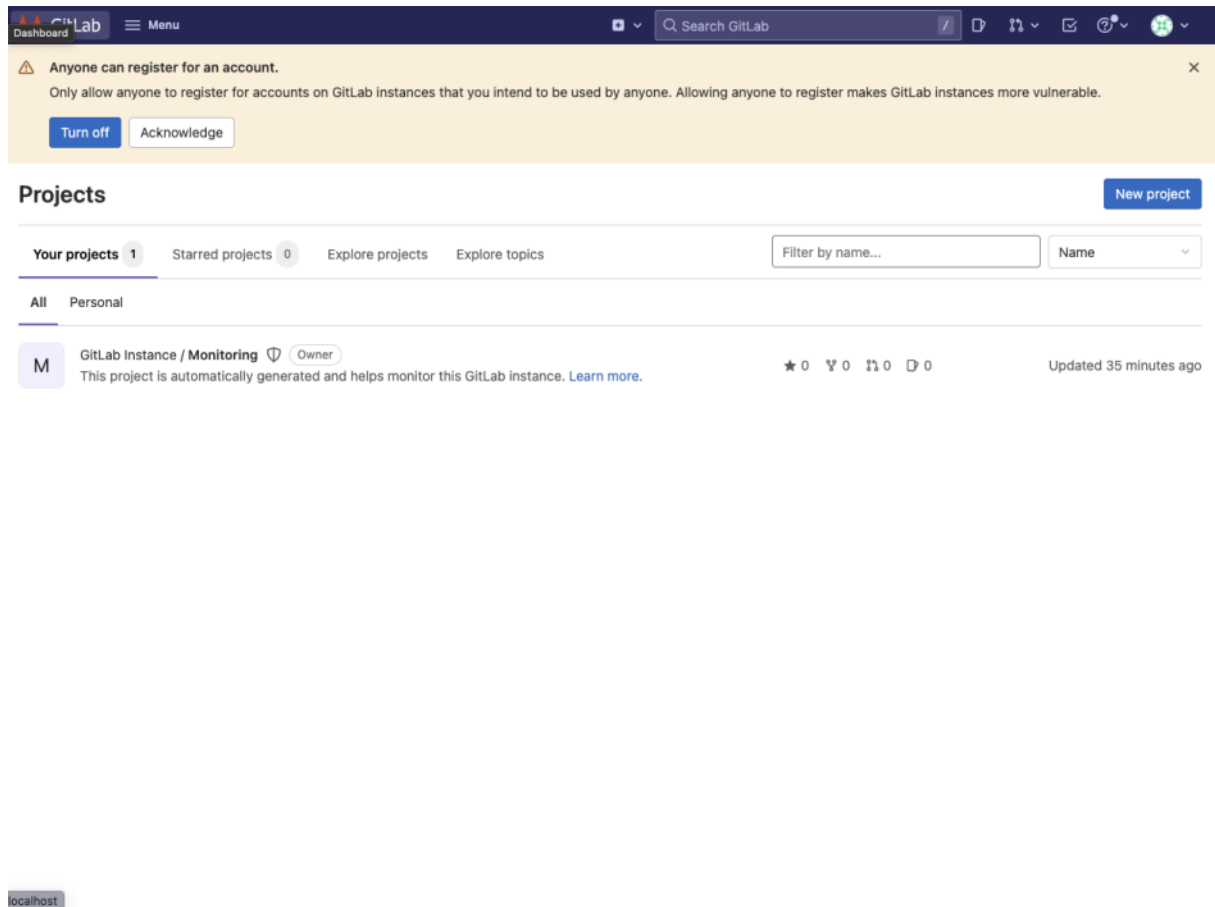
- Username : root,
- Pour le mot de passe, il se trouve dans le fichier initial_root_password à l'adresse /etc/gitlab.

Remarque

Dans le cas d'une installation via docker, il faudra se connecter sur l'instance Docker GitLab afin d'accéder au fichier.

Avec ces identifiants, la connexion se fait en tant qu'administrateur, il est alors recommandé de créer un nouveau compte avec des droits similaires, ou modifier les accès du compte existant afin de sécuriser ce dernier.

L'interface d'utilisation de GitLab s'affiche alors comme la capture ci-dessous :



Capture d'écran de l'interface GitLab

Attention

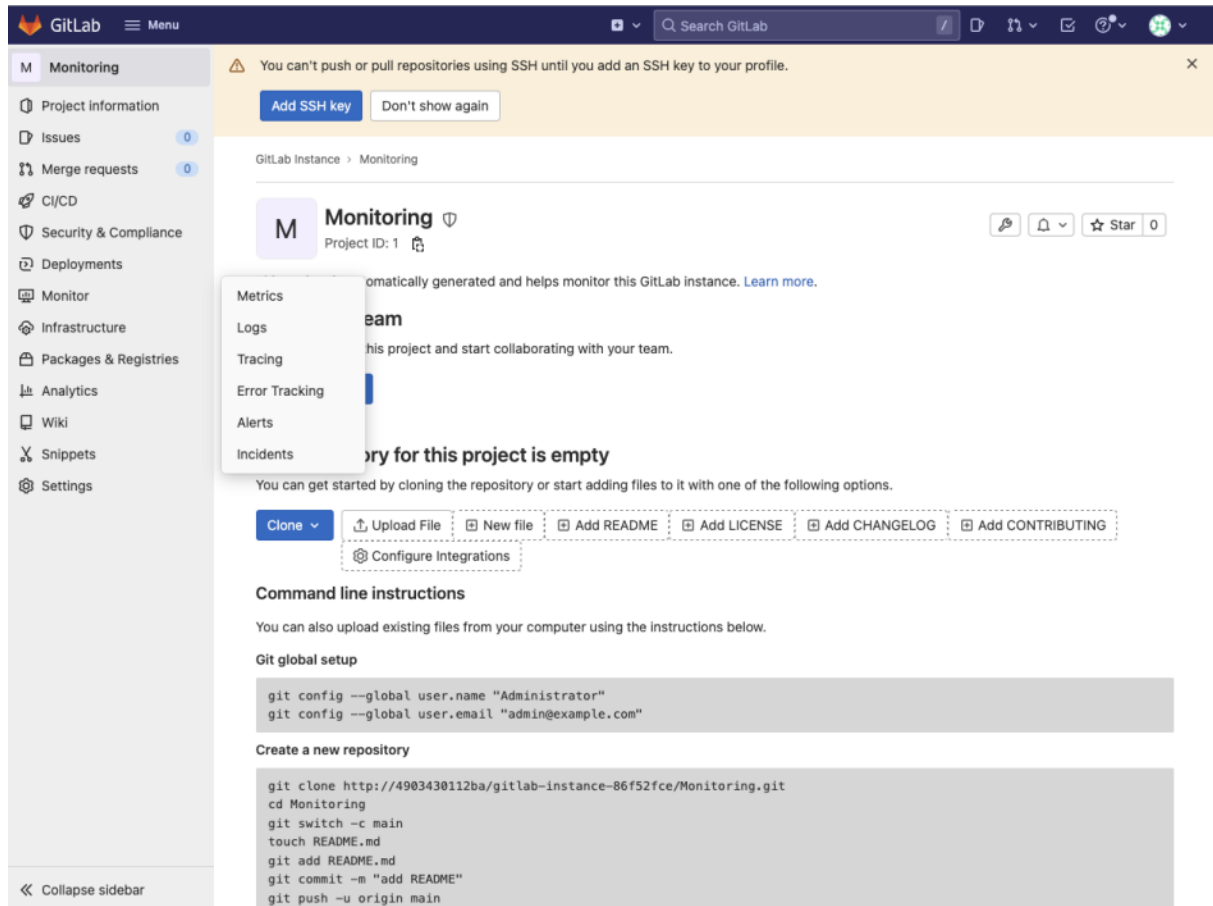
Comme indiqué sur la première page, n'importe quelle personne peut se créer un compte sur cette instance avec les paramètres actuels (option, pouvant être modifié en cliquant sur Turn off).

La page présentée est alors la première page de l'instance GitLab, elle présente les projets dont un 'Monitoring' créé par défaut.

Un projet possède alors son propre espace avec :

- Son Repository
- Ses issues
- Son espace déploiement continu
- Son moniteur
- Son propre wiki
- Ses snippets
- Et ses réglages

Ainsi un projet sur GitLab possède toutes ces caractéristiques propres et devient un espace propre au développement de chacun d'entre eux.



Capture d'écran projet de base sur GitLab

À partir de ce projet, les échanges avec git sont similaires à ceux avec GitHub.

Ainsi, il est possible de synchroniser le projet avec les commandes visibles dans la partie 'Command line instructions'.

Via les commandes suivantes pour synchroniser un Repository local existant :

```
1 $ cd projects
2 $ git remote add origin <adresse GitLab>/projects.git
3 $ git push -u origin main
4
```

Il est alors possible ensuite de créer un environnement autour de se projets, avec :

- Les tickets pour le suivi des fonctions, des bugs, etc.,
- Le déploiement et CI/CD pour le déploiement continu du projet,
- Le wiki pour créer une documentation complète du projet,
- Et d'autres fonctions avancées permettant l'optimisation du développement du projet.

Exercice : Quiz

[solution n°2 p.18]

Exercice

Les 2 solutions d'installations en self-hosted principal sont :

- ☐ L'installation native
- ☐ L'installation par container
- ☐ L'installation par support externe

Exercice

L'installation native est limitée par :

- ☐ Les ressources
- ☐ Les droits d'accès
- ☐ Le système d'exploitation

Exercice

Les 2 versions installables de GitLab sont :

- ☐ La Community Edition
- ☐ L'Enterprise Edition
- ☐ La Premium Edition
- ☐ L'Ultimate Edition

Exercice

Le mot de passe par défaut d'une instance GitLab est :

- ☐ Défini dans un fichier
- ☐ Admin
- ☐ Root
- ☐ Inexistant

Exercice

Un projet sur GitLab possède son propre :

- ☐ Espace de déploiement continu
- ☐ Repository
- ☐ Espace de ticket
- ☐ Wiki

V. Essentiel

GitLab est une solution de forge pour Repository git particulièrement complète et puissante. Cette solution possède l'avantage d'être Open Source et principalement gratuite à l'exception de certaines fonctions spécifiques.

GitLab peut aussi bien être une solution auto-hébergée, qu'une solution hébergée par la société propriétaire de GitLab via leur solution cloud et un compte sur leur site Gitlab¹.

À partir de ce site, 2 abonnements sont proposés afin de fournir toutes les fonctions que possèdent l'offre de GitLab en plus de l'abonnement gratuit, cependant l'abonnement gratuit ne fournit qu'un support des projets Open Source sous licence MIT.

L'installation quant à elle se passe via un container, ou une installation native, et permet un support gratuit des projets Open Source ou privé.

À partir de l'installation, GitLab se gère par projet. Chaque projet possède un environnement propre, gérant ses échanges git, son déploiement continu, ses tickets, son propre wiki, son tableau de bord d'analyse et ses propres configurations.

Enfin, une instance GitLab pourra être utilisée pour une gestion privée, mais aussi pour fournir une forge communautaire.

VI. Auto-évaluation

A. Exercice

Dans une société utilisant la containerisation, le responsable vous demande de créer un fichier afin de déployer GitLab sur l'un de leur serveur.

Question 1

[solution n°3 p.19]

Réalisez le fichier de composition Docker pour le serveur dont l'adresse est 208.65.153.238 et le nom DNS www.containgl.net² pour installer la version Community de GitLab. Le serveur n'étant pas HTTPS quelles seront les 2 adresses permettant d'accéder au service GitLab ?

Question 2

[solution n°4 p.19]

Le responsable vous demande quels avantages et désavantages il y a à une utilisation de GitLab contrairement à la solution Gitea, citez les principaux.

B. Test

Exercice 1 : Quiz

[solution n°5 p.20]

Question 1

La première version de GitLab est sortie en :

- ☐ 2011
- ☐ 2012
- ☐ 2013
- ☐ 2014

Question 2

1 <https://gitlab.com/>

2 <http://www.containgl.net/>

GitLab fonctionne sous :

- ☐ Go
- ☐ Java
- ☐ PHP
- ☐ Ruby

Question 3

GitLab est nativement installable sous Windows.

- ☐ Faux
- ☐ Vrai

Question 4

L'installation via un système de containerisation débloque les limites d'OS.

- ☐ Vrai
- ☐ Faux

Question 5


GitLab fonctionne avec un système principal de :

- ☐ Projets
- ☐ Repositories
- ☐ Steps
- ☐ Lignes

Solutions des exercices


Exercice p. 7 Solution n°1**Question 1**

Comment s'appelle un hébergeur de Repositories en ligne ?

- ☒ Une Forge
- ☐ Une Herber
- ☐ Un FrauGit
- ☐ Un GitterLine
-  Un hébergeur de Repositories en ligne se nomme une forge.


Question 2

GitLab est une solution :

- ☐ Propriétaire
- ☒ Open Source
-  GitLab est une solution Open Source.


Question 3

GitLab est gratuit dans tous les cas.

- ☒ Faux
- ☐ Vrai
-  GitLab propose des fonctionnalités payantes et d'autres gratuites.

Question 4


GitHub et GitLab sont des solutions dont le principal avantage est d'être :

- ☒ Complète
- ☐ Légère
- ☐ Gratuite
-  GitHub et GitLab sont des solutions complètes permettant de gérer plusieurs étapes du développement comme le développement continue par exemple.

Question 5

GitLab peut être utilisé :

- ☒ En auto-hébergé
- ☒ En cloud
- ☐ En Datacenter pré-hébergé

 GitLab peut être utilisé en cloud ou en self-hosted donc auto-hébergé.

Exercice p. 13 Solution n°2

Exercice

Les 2 solutions d'installations en self-hosted principal sont :


- ☒ L'installation native
- ☒ L'installation par container
- ☐ L'installation par support externe

 L'installation native et par container sont les 2 solutions d'installations principales.

Exercice

L'installation native est limitée par :


- ☐ Les ressources
- ☐ Les droits d'accès
- ☒ Le système d'exploitation

 L'installation native est principalement limitée par le système d'exploitation.

Exercice

Les 2 versions installables de GitLab sont :


- ☒ La Community Edition
- ☒ L'Enterprise Edition
- ☐ La Premium Edition
- ☐ L'Ultimate Edition

 Les 2 versions installables sont la Community Edition et l'Enterprise Edition.

Exercice

Le mot de passe par défaut d'une instance GitLab est :


- ☒ Défini dans un fichier
- ☐ Admin
- ☐ Root
- ☐ Inexistant

 Le mot de passe par défaut d'une instance GitLab est défini dans un fichier dans le dossier /etc/gitlab.

Exercice

Un projet sur GitLab possède son propre :

- ☒ Espace de déploiement continu
- ☒ Repository
- ☒ Espace de ticket
- ☒ Wiki

 Un projet sur GitLab possède son propre environnement notamment avec son Espace de déploiement continu, son Repository, son espace de ticket et son propre wiki.

p. 15 Solution n°3

Pour réaliser le fichier docker-compose, on utilise l'exemple vu dans le cours, en notifiant bien le bon nom de serveur et décommentant les lignes, et en modifiant le ee en ce, on obtient le script suivant :

```

1 version: '3.6'
2 services:
3   GitLab:
4     image: 'gitlab/gitlab-ce:latest'
5     restart: always
6     hostname: 'http://containgl.net'
7     environment:
8       GITLAB_OMNIBUS_CONFIG: |
9         external_url 'http://containgl.net'
10    ports:
11      - '80:80' #Port HTTP
12      - '22:22' #Port SSH
13    volumes:
14      - 'gitlab_config:/etc/gitlab'
15      - 'gitlab_logs:/var/log/gitlab'
16      - 'gitlab_data:/var/opt/gitlab'
17    shm_size: '256m'
18 volumes:
19   gitlab_config:
20   gitlab_logs:
21   gitlab_data:
22
```

Ainsi, les services de GitLab seront disponible aux adresses : <http://www.containgl.net>¹ ou 208.65.153.238³ ou plus explicitement sur le port 80 en ajoutant ':80' à la fin des adresses.

p. 15 Solution n°4

Le principal désavantage est que GitLab est énormément plus gourmand en ressource que Gitea. Cependant GitLab fourni des options bien plus complètes comme le développement continu, la gestion des tickets, le suivi via milestones, la création de wiki et énormément d'autres fonctions, là où Gitea est uniquement une forge pour les Repositories git.

1 <http://www.containgl.net/>

2 <http://www.containgl.net/>

3 <http://208.65.153.238/>

Exercice p. 15 Solution n°5

Question 1

La première version de GitLab est sortie en :

- ☒ 2011
- ☐ 2012
- ☐ 2013
- ☐ 2014

 La première version de GitLab est sortie en Octobre 2011.

Question 2

GitLab fonctionne sous :


- ☐ Go
- ☐ Java
- ☐ PHP
- ☒ Ruby

 GitLab est développé en Ruby.

Question 3

GitLab est nativement installable sous Windows.


- ☒ Faux
- ☐ Vrai

 Faux, il est installable sur quelques systèmes Linux uniquement : Ubuntu, Debian, AlmaLinux, CentOS et OpenSuse, ainsi que sur Raspberry pi os.

Question 4


L'installation via un système de containerisation débloque les limites d'OS.

- ☒ Vrai
- ☐ Faux

 Vrai, Grâce à la containerisation, GitLab est installable sur de plus nombreux système (notamment Windows et macOS).

Question 5

GitLab fonctionne avec un système principal de :

- ☒ Projets
- ☐ Repositories
- ☐ Steps
- ☐ Lignes
-  GitLab fonctionne avec un système principal de Projets qui contiendra : le Repository, les tickets, et tout l'environnement d'un projet.