Lab - installer et configurer GitLab CE sur CentOS 7

GitLab est un gestionnaire de référentiel open source basé sur Rails développé par GitLab Inc. Il s'agit d'un gestionnaire de référentiel git basé sur le Web qui permet à votre équipe de collaborer sur le codage, le test et le déploiement d'applications. GitLab fournit plusieurs fonctionnalités, notamment les wikis, le suivi des problèmes, les révisions de code et les flux d'activité.

GitLab Inc propose 4 produits:

- Gitlab CE (Community Edition) - auto-hébergé et gratuit; soutien du forum communautaire.
- Gitlab EE (Enterprise Edition) - auto-hébergé et payant; est livré avec des fonctionnalités supplémentaires.
- GitLab.com - SaaS et gratuit.
- GitLab.io - Instance GitLab privée gérée par GitLab Inc.

Conditions préalables

- Serveur CentOS 7 - 64 bits
- RAM minimum 2 Go
- Privilèges root

Étape 1 - Installer les packages

Installez certains packages nécessaires à l'installation de GitLab

```
yum -y install curl policycoreutils openssh-server openssh-clients postfix
```

Après cela, démarrez les services ssh et postfix.

```
systemctl start sshd
systemctl start postfix
```

Maintenant, permettez-leur de s'exécuter automatiquement au moment du démarrage.

```
systemctl enable sshd
systemctl enable postfix
```

Étape 2 - Installez GitLab

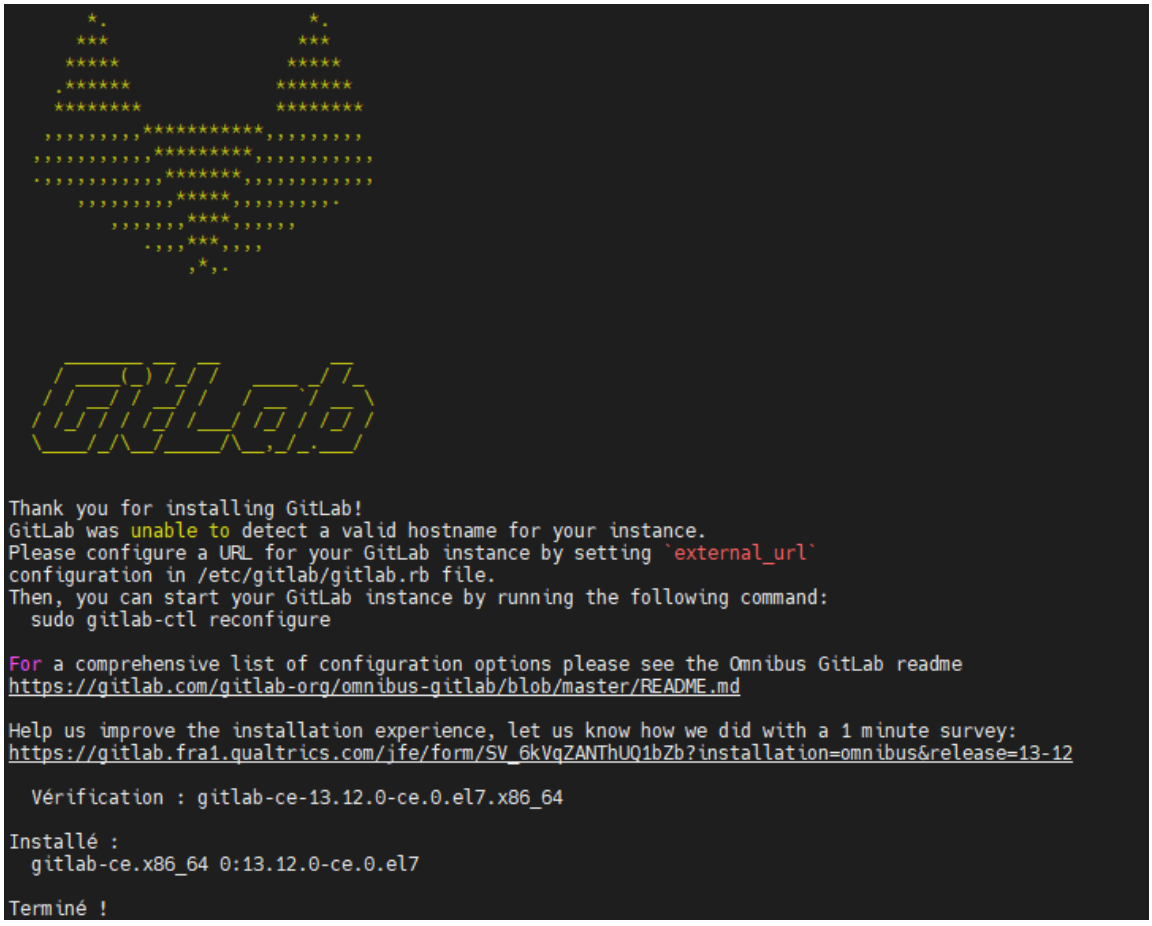
GitLab fournit un programme d'installation pour ajouter le référentiel GitLab CE. Téléchargez le programme d'installation avec curl et exécutez le script (comme indiqué ci-dessous) pour ajouter un nouveau référentiel GitLab CE.

```
curl -sS https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.rpm.sh | sudo bash
```

Le référentiel GitLab CE a été ajouté au système. Maintenant, installez GitLab avec la commande yum suivante.

```
yum -y install gitlab-ce
```

Une fois l'installation terminée, vous obtiendrez le résultat comme indiqué ci-dessous.



Avec cela, GitLab CE est maintenant installé sur le serveur CentOS 7.

Étape 3 - Configurer l'URL GitLab

Pour ce tutoriel, nous utiliserons un nom de domaine pour GitLab. Plus précisément, nous utiliserons le nom de domaine «gitlab.gk.fr».

Editez le fichier de configuration de GitLab '/etc/gitlab/gitlab.rb '.

```
vi /etc/gitlab/gitlab.rb
```

Modifiez la ligne external_url avec le nom de domaine «**gitlab.gk.fr**».

```
external_url 'http://gitlab.gk.fr'
```

Étape 4 - Générer le SSL Let's encrypt et le certificat DHPARAM

Pour la couche de sécurité de base, nous utiliserons le SSL pour notre site GitLab. Nous utiliserons un certificat SSL gratuit et générerons un certificat DHPARAM pour ajouter une couche de sécurité supplémentaire.

Pour générer le certificat SSL, nous devons installer le module SSL, qui est disponible dans le référentiel epel-release.

```
yum –y install epel-release
```

Installez l'outil Letsencrypt sur CentOS 7 avec la commande yum ci-dessous.

```
yum -y install mod_ssl
```

Créez un nouveau répertoire 'ssl' sous le répertoire de configuration de GitLab '/etc/gitlab/'.

```
mkdir -p /etc/gitlab/ssl/
```

Une fois l'installation terminée, générez un nouveau certificat SSL avec la commande ci-dessous.

```
[root@worker-node2 ~]# openssl req -x509 -nodes -days 365 -newkey rsa:2048
-keyout /etc/gitlab/ssl/gitlab-selfsigned.key -out /etc/gitlab/ssl/gitlab-
selfsigned.crt
Generating a 2048 bit RSA private key
.........................+++
.............................................+++
writing new private key to '/etc/gitlab/ssl/gitlab-selfsigned.key'
```

```
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:FR
State or Province Name (full name) []:France
Locality Name (eg, city) [Default City]:Paris
Organization Name (eg, company) [Default Company Ltd]:GK
Organizational Unit Name (eg, section) []:Training
Common Name (eg, your name or your server's hostname) []:gitlab.gk.fr
Email Address []:admin@gitlab.gk.fr
```

Générez maintenant le fichier pem du certificat DHPARAM à l'aide d'OpenSSL. Le plus gros est plus sûr.

```
sudo openssl dhparam -out /etc/gitlab/ssl/dhparams.pem 2048
```

Et une fois le certificat DHPARAM généré, modifiez l'autorisation du fichier de certificat sur 600.

```
chmod 600 /etc/gitlab/ssl/*
```

Ainsi, le certificat SSL et DHPARAM pour l'installation de GitLab a été généré.

Étape 5 - Activez Nginx HTTPS pour GitLab

À ce stade, nous disposons déjà de fichiers de certificats SSL gratuits et de certificats DHPARAM qui sont générés à l'aide de la commande OpenSSL. Et dans cette étape, nous activerons HTTPS pour le site GitLab. Nous activerons HTTPS et forcerons HTTP à la connexion HTTPS.

Tout d'abord, allez dans le répertoire de configuration de GitLab et éditez le fichier de configuration 'gitlab.rb'.

```
vi /etc/gitlab/gitlab.rb
```

Et changez HTTP en HTTPS sur la ligne external_url.

```
external_url 'https://gitlab.gk.fr'
```

Collez ensuite la configuration suivante sous la configuration de la ligne 'external_url'.

```
nginx['redirect_http_to_https'] = true
nginx['ssl_certificate'] = "/etc/gitlab/ssl/gitlab-selfsigned.crt"
nginx['ssl_certificate_key'] = "/etc/gitlab/ssl/gitlab-
selfsigned.key"
nginx['ssl_dhparam'] = "/etc/gitlab/ssl/dhparams.pem"
```

Enfin, appliquez la configuration GitLab à l'aide de la commande suivante.

```
gitlab-ctl reconfigure
```

Étape 6 - Configurer Firewalld

Démarrez firewalld et activez-le pour qu'il s'exécute automatiquement au moment du démarrage avec les commandes systemctl comme indiqué ci-dessous.

```
systemctl start firewalld
systemctl enable firewalld
```

Ensuite, ouvrez de nouveaux ports pour nos services. Nous ouvrirons les ports SSH, HTTP et HTTPS pour notre configuration GitLab. Exécutez les commandes firewall-cmd ci-dessous pour ouvrir les ports.

```
firewall-cmd --permanent --add-service ssh
firewall-cmd --permanent --add-service http
firewall-cmd --permanent --add-service https
```

Maintenant, rechargez le pare-feu et vérifiez la configuration de firewalld. Assurez-vous que SSH, HTTP et HTTPS figurent dans la liste.

```
firewall-cmd --reload
firewall-cmd --list-all
```

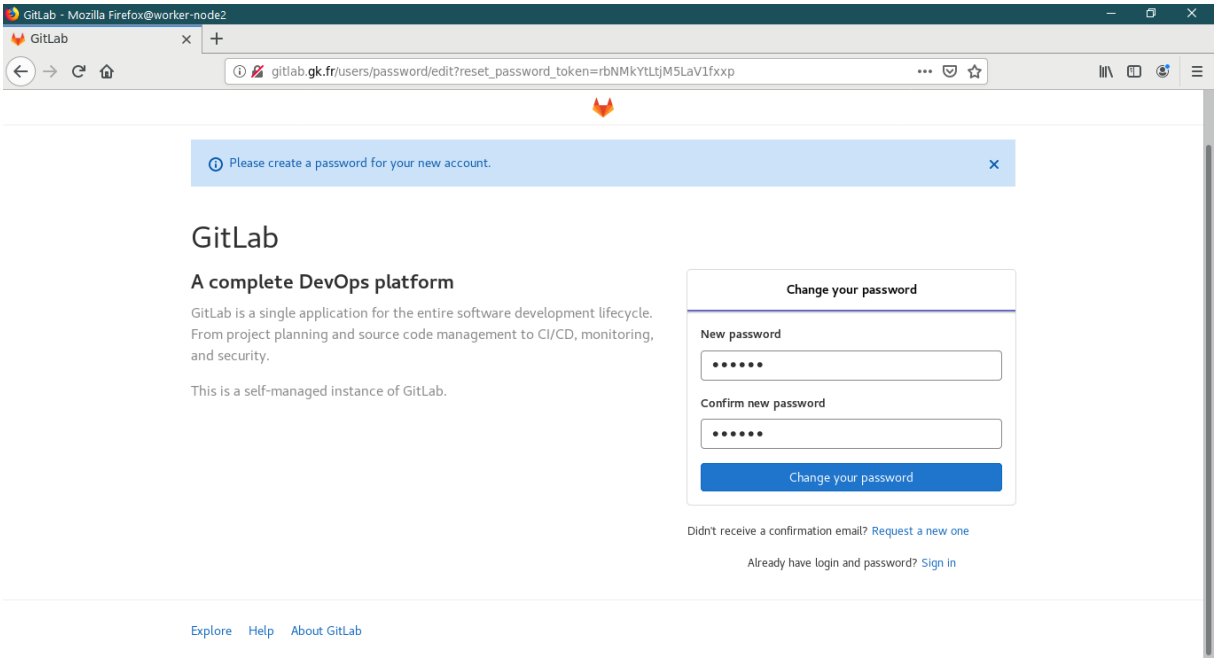
Ainsi, la configuration de Firewalld pour GitLab est terminée.

Étape 7 - Effectuer l'installation

Dans cette étape, nous effectuerons quelques réglages rapides après l'installation de GitLab sur le serveur.

Réinitialiser le mot de passe root GitLab

Ouvrez votre navigateur Web et saisissez l'URL gitlab «gitlab.gk.fr ». Changez le mot de passe root avec votre propre mot de passe et cliquez sur le bouton «Changer votre mot de passe» pour confirmer.



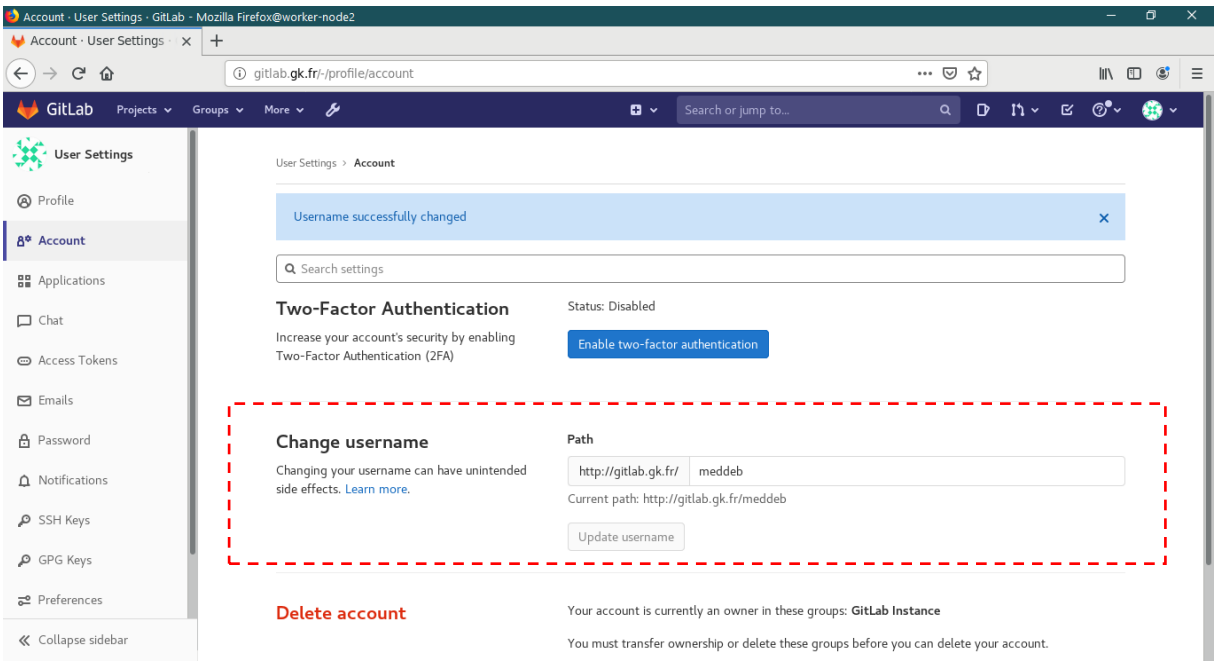
Vous pouvez maintenant vous connecter au tableau de bord GitLab avec l'utilisateur par défaut «root» et votre propre mot de passe.

Changer de profil et de nom d'utilisateur

Après vous être connecté au tableau de bord GitLab, cliquez sur en haut à droite de votre profil d'icône, puis sur l'icône "Paramètres" pour configurer votre profil.

Dans l'onglet "Profil", modifiez votre nom et votre adresse e-mail, puis cliquez sur le bouton "Mettre à jour les paramètres du profil" en bas pour confirmer.

Ensuite, allez dans l'onglet «Compte» et changez le nom d'utilisateur racine par défaut avec votre propre nom d'utilisateur, puis cliquez sur le bouton «Mettre à jour le nom d'utilisateur».



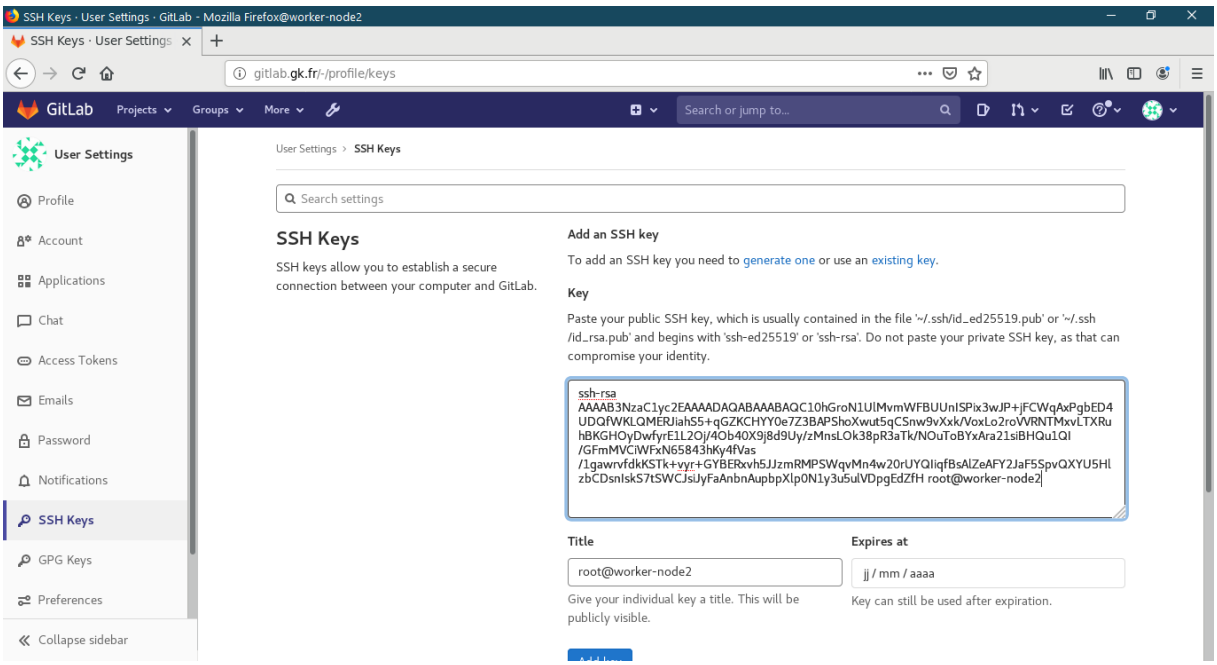
Ajouter une clé SSH

Assurez-vous que vous avez déjà une clé, si vous n'avez pas de clé SSH, vous pouvez en générer une à l'aide de la commande ci-dessous.

```
[root@gitlab ~]# ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:CGbA2AFUyKp5dQ+yRa4sfVr4arKFESznANpUOEOGV7s root@worker-node2
The key's randomart image is:
+---[RSA 2048]----+
|=OO*o            |
|=BB. ..          |
|=.=o=o           |
|.= +oo=.         |
|..o+EB.oS        |
|o oo* o .        |
| ....=           |
|  ..o .          |
|  .+..           |
+----[SHA256]-----+
```

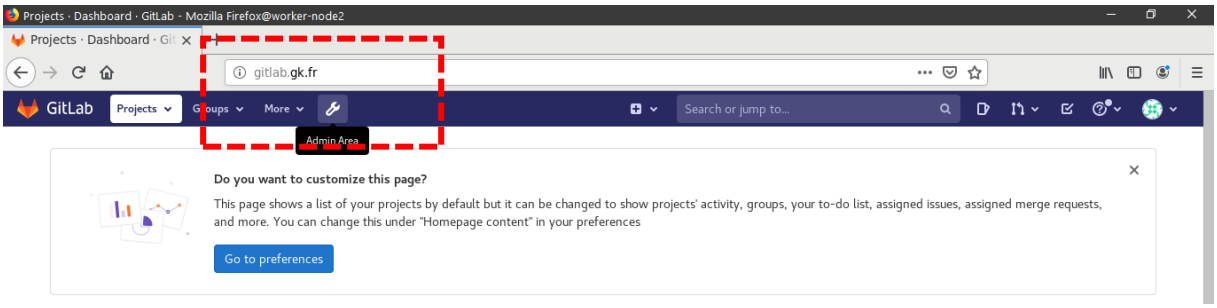
Et vous obtiendrez deux clés dans le répertoire ~ / .ssh /. «id_rsa» serait votre clé privée et «id_rsa.pub» serait votre clé publique.

Ensuite, revenez au navigateur Web et cliquez sur l'onglet «Clé SSH». Copiez le contenu du fichier 'id_rsa.pub' et collez-le dans la boîte à clés, puis cliquez sur 'Ajouter une clé'.

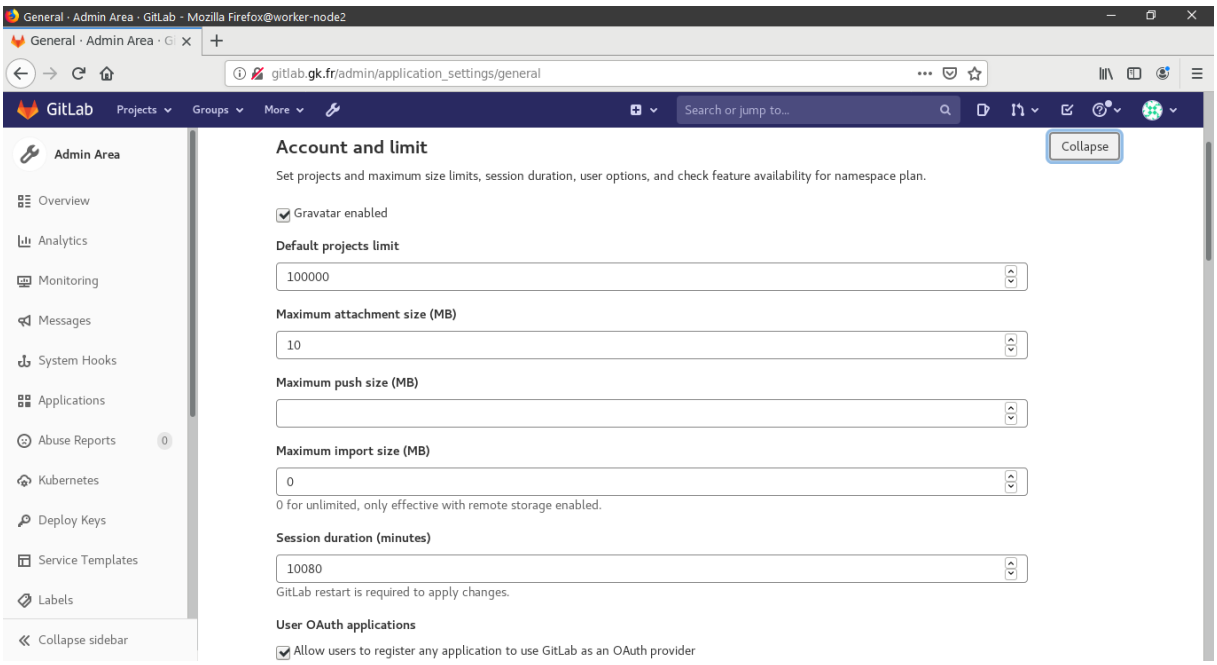


Les restrictions et les paramètres de limite

Cliquez sur l'icône «Zone d'administration», puis sur l'icône d'engrenage et choisissez «Paramètres».



Dans la section «Paramètres de compte et de limite», vous pouvez configurer le projet maximum par utilisateur. Et dans la section «Restrictions d'inscription», vous pouvez ajouter le nom de domaine de votre e-mail à la zone de liste blanche.



Effectuez des autres configurations personnalisées sur votre instance Gitlab.

La configuration de base de GitLab est terminée.

Étape 8 - Test

Maintenant, nous allons faire quelques tests avec notre GitLab auto-hébergé.

Créer un nouveau projet

Cliquez sur l'icône plus en haut à droite pour créer un nouveau référentiel de projet.

Saisissez le nom, la description et les paramètres de visibilité de votre projet pour votre projet. Et puis cliquez sur le bouton «Créer un projet».

Tester le premier commit

Une fois votre projet créé (demo1 dans notre cas), vous serez redirigé vers la page du projet. Maintenant, commencez à ajouter du nouveau contenu au référentiel.

Assurez-vous que Git est installé sur votre ordinateur.

Pour ce test, nous devons configurer le compte Git sur l'ordinateur, ce que vous pouvez faire à l'aide des commandes suivantes:

```
git config --global user.name "meddeb"
git config --global user.email "admin@gk.fr"
```

Clonez le référentiel et ajoutez un nouveau fichier README.md.

```
git clone http://gitlab.gk.fr/meddeb/demo.git
cd demo
touch README.md
```

Il vous sera demandé le mot de passe hakase. Veuillez saisir le même mot de passe que celui que nous avons utilisé lors de l'accès à GitLab pour la première fois, puis ajoutez un nouveau contenu au fichier README.md.

Validez les nouvelles modifications dans le référentiel à l'aide des commandes suivantes.

```
git add README.md
git commit -m "add README"
```

Ensuite, transférez le référentiel sur le serveur GitLab.

```
git push -u origin master
```

Tapez votre mot de passe et appuyez sur Entrée pour continuer. Vous devriez voir le résultat comme indiqué ci-dessous.