**Accéder à GitLab et gérer les membres**

1. **Aller sur GitLab** : Connectez-vous à votre compte GitLab.
2. **Cliquer sur le projet** : Sélectionnez le projet sur lequel vous souhaitez travailler (par exemple, "Demo").
3. **Choisir Members** : Dans la barre latérale gauche, cliquez sur "Members".
4. **Inviter des membres** : Cliquez sur "Invite members" pour ajouter de nouveaux membres au projet. Vous pourrez définir leur rôle (Maintainer, Developer, Reporter, etc.) lors de l'invitation.
5. **Envoyer les invitations** : Les membres recevront des invitations par e-mail. Assurez-vous que le serveur mail est correctement configuré pour envoyer les invitations.

**Configuration du serveur mail**

1. **Accéder au serveur mail** : Utilisez vim ou tout autre éditeur pour configurer le serveur mail. Vous devrez accepter les invitations dans le serveur mail.

vim /path/to/mail/server/config

**Accepter les invitations** : Les membres doivent accepter les invitations qu'ils reçoivent par e-mail pour rejoindre le projet.

**Configurer les branches protégées**

1. **Configurer le branchement** : Une fois les membres ajoutés, vous pouvez configurer les branches pour protéger certaines d'entre elles.
2. **Protéger une branche** : Pour protéger une branche, allez dans les paramètres du projet, sous "Repository", puis "Protected Branches". Cliquez sur "Protect" à côté de la branche que vous souhaitez protéger (par exemple, main ou master).

Settings > Repository > Protected Branches

1. **Configurer les permissions** : Vous pouvez définir qui peut pousser ou fusionner des modifications dans cette branche. Choisissez les rôles appropriés (Maintainers, Developers, etc.) qui peuvent avoir accès à ces actions.

**Résumé des commandes et configurations**

* Accès au projet et gestion des membres se fait via l'interface utilisateur de GitLab.
* Configuration du serveur mail via vim pour s'assurer que les invitations sont envoyées et reçues correctement.
* Protection des branches pour contrôler les permissions et éviter les modifications non autorisées.

**Configuration des branches protégées**

1. **Accéder au projet "Greeting"** :
   * Connectez-vous à votre compte GitLab.
   * Sélectionnez le projet "Greeting".
2. **Protéger les branches** :
   * Allez dans les paramètres du projet :

Settings > Repository > Protected Branches

* + Cliquez sur "Protect" à côté de la branche que vous souhaitez protéger (par exemple, main, development, etc.).
  + Configurez les permissions pour spécifier qui peut pousser ou fusionner des modifications dans cette branche. Définissez les rôles appropriés (Maintainers, Developers, etc.).

**Gestion des groupes et sous-groupes**

1. **Créer un groupe** :
   * Dans la barre latérale gauche, cliquez sur "Groups" et ensuite sur "Your groups".
   * Cliquez sur "Create group" pour créer un nouveau groupe.
   * Remplissez les informations nécessaires et définissez la visibilité du groupe (privé, interne ou public).

 **Créer un sous-groupe** :

* Accédez au groupe que vous venez de créer.
* Cliquez sur "New subgroup" pour créer un sous-groupe.
* Remplissez les informations nécessaires pour le sous-groupe.

 **Ajouter des membres aux groupes** :

* Une fois dans le groupe ou sous-groupe, allez dans "Group Members".
* Cliquez sur "Invite members" pour ajouter des utilisateurs au groupe ou sous-groupe. Définissez leur rôle (Owner, Maintainer, Developer, etc.).
* Les membres recevront une invitation par e-mail qu'ils devront accepter.

**Positionnement et configuration des branches dans les groupes**

1. **Accéder aux groupes créés** :
   * Allez dans "Groups" et sélectionnez le groupe souhaité.
   * Si vous avez des sous-groupes, sélectionnez le sous-groupe correspondant.
2. **Ajouter des projets aux groupes** :
   * Une fois dans le groupe ou sous-groupe, vous pouvez ajouter des projets en cliquant sur "New project" ou en transférant des projets existants vers le groupe.
3. **Positionner les branches protégées** :
   * Accédez au projet dans le groupe ou sous-groupe.
   * Configurez les branches protégées comme mentionné dans la section précédente.

**Résumé des étapes :**

1. **Accès et protection des branches** : Paramétrez les branches protégées dans le projet "Greeting".
2. **Gestion des groupes** : Créez des groupes et sous-groupes pour organiser les projets et gérer les permissions.
3. **Ajout de membres** : Invitez des membres et définissez leurs rôles pour gérer l'accès et les permissions.
4. **Positionnement des branches dans les groupes** : Ajoutez des projets aux groupes et configurez les branches protégées selon les besoins du groupe.

Pour gérer le projet, récupérer les fichiers depuis GitLab, installer les dépendances avec Composer, et exécuter les tests PHPUnit, voici les étapes détaillées :

**Préparer le projet et installer Composer**

1. **Récupérer le projet depuis GitLab** :
   * Clonez le projet depuis GitLab en utilisant Git.

git clone <URL\_DU\_PROJET\_GITLAB>

cd <NOM\_DU\_PROJET>

**Installer Composer** :

* Si Composer n'est pas déjà installé, utilisez la commande suivante pour l'installer :

php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"

php composer-setup.php

php -r "unlink('composer-setup.php');"

mv composer.phar /usr/local/bin/composer

**Vérifier l'installation de Composer** :

* Pour vérifier que Composer est correctement installé, utilisez :

composer -V

**nstaller les dépendances et configurer PHPUnit**

1. **Installer les dépendances** :
   * Naviguez dans le répertoire du projet et installez les dépendances définies dans composer.json :

composer install

**Installer PHPUnit** :

* Ajoutez PHPUnit comme dépendance de développement :

composer require --dev phpunit/phpunit:^10.5.20

**Exécuter les tests avec PHPUnit** :

* Lancez les tests en utilisant la commande suivante :

vendor/bin/phpunit

**Lancer le serveur PHP**

1. **Démarrer le serveur PHP** :
   * Utilisez la commande suivante pour lancer le serveur PHP en local sur le port 1500 :

php -S localhost:1500

**Gestion des utilisateurs**

1. **Changer d'utilisateur** :
   * Chaque fois que vous changez d'utilisateur, vous devrez réinstaller les dépendances :

**Vérification de la configuration réseau**

1. **Utiliser ifconfig pour voir les configurations réseau** :
   * La commande ifconfig affiche les configurations réseau de la machine. Si ifconfig n'est pas disponible (car obsolète sur certaines distributions), utilisez ip a.

ifconfig

**Générer l'autoload de Composer**

1. **Générer l'autoload** :
   * Utilisez la commande suivante pour générer l'autoload de Composer :

composer dump-autoload

**Exécuter les tests PHPUnit**

1. **Exécuter les tests avec PHPUnit** :
   * Utilisez la commande suivante pour lancer les tests situés dans le répertoire phome/leser/hanoi-php/tests :

vendor/bin/phpunit phome/leser/hanoi-php/tests

**Configurer une clé SSH pour se connecter en tant que développeur**

1. **Créer une clé SSH** :
   * Utilisez la commande suivante pour générer une clé SSH avec un algorithme RSA de 1024 bits :

ssh-keygen -b 1024 -t rsa -f ~/.ssh/id\_rsa\_devop -C "devop@your-domain.com"

*  Lorsque vous exécutez cette commande, vous serez invité à fournir un nom pour le fichier de clé et une phrase de passe si vous le souhaitez.

 **Ajouter la clé SSH à l'agent SSH** :

* Ajoutez la clé privée nouvellement générée à l'agent SSH :

eval "$(ssh-agent -s)"

ssh-add ~/.ssh/id\_rsa\_devop

**Ajouter la clé publique à GitLab** :

* Copiez le contenu de la clé publique générée :

cat ~/.ssh/id\_rsa\_devop.pub

Allez dans les paramètres de votre compte GitLab, puis dans "SSH Keys", et collez le contenu de la clé publique dans le champ prévu à cet effet.