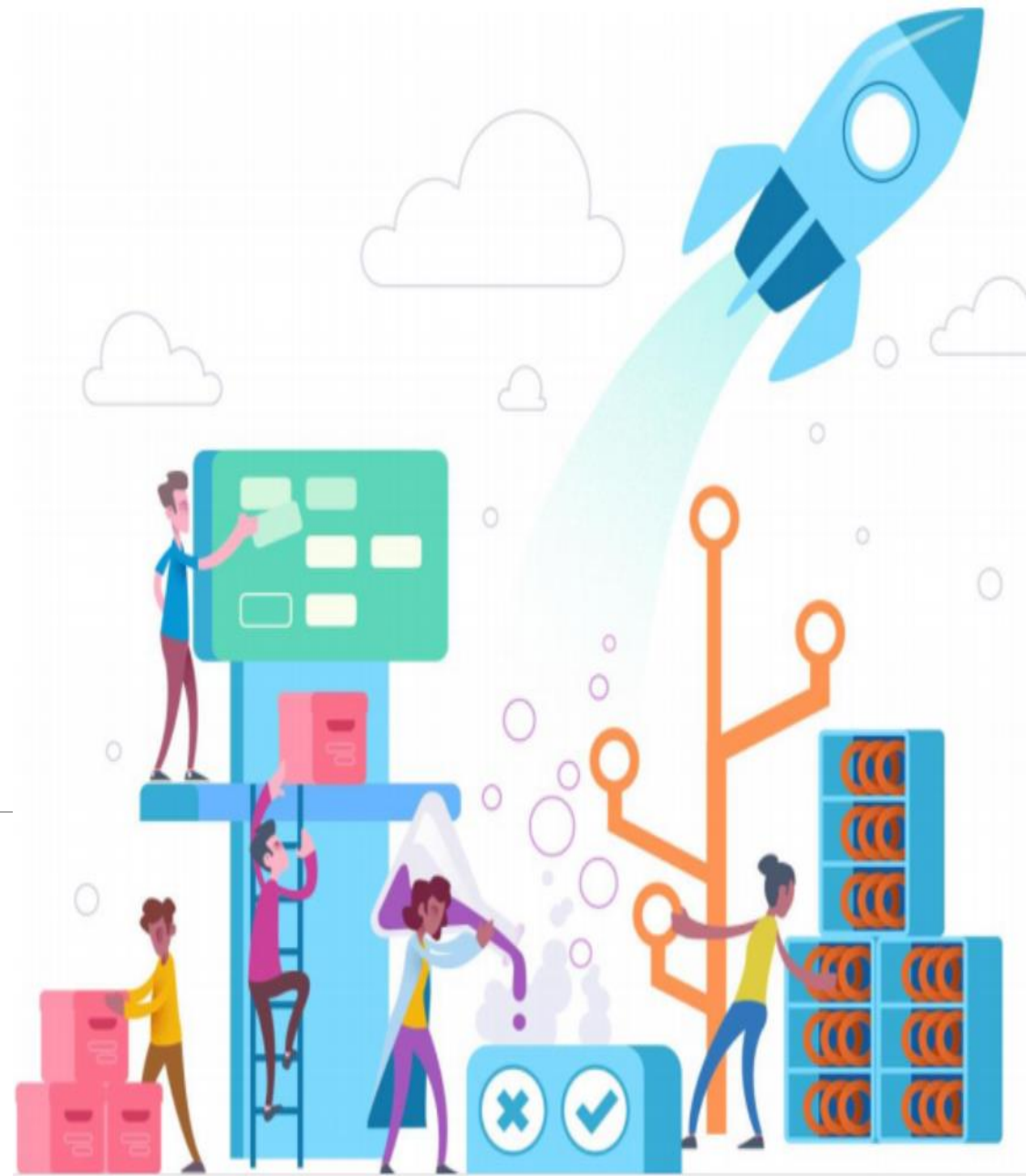


Guide Git et Visual Studio

Pr. EL-ALAOUY EL-ARBI

E.ELALAOUY.EMSI@GMAIL.COM



Objectifs

- ❑ Intégration de Git dans un projet Visual Studio
- ❑ Gestion de versions git d'un projet Visual Studio
- ❑ Gestion de communication entre un dépôt local et distant

Se connecter à son environnement préféré

Les dépôts Azure peuvent être connectés avec plusieurs environnements. Ci-dessous il y a une liste non exhaustive :

[Command-line](#)

[Visual Studio Code](#)

[Visual Studio](#)

[Xcode](#)

[Eclipse](#)

[IntelliJ](#)

Visual Studio

Les solutions Visual Studio peuvent être partagées en envoyant le code vers les dépôts Azure ou bien Github.

Ci-dessous des tâches communes que l'on peut réaliser sur Visual Studio:

- ☐ Ajouter un projet/solution à un dépôt git local
- ☐ Publier votre code vers un dépôt Azure ou un dépôt Github
- ☐ Réviser les changements de code avant son publication
- ☐ Lier les tâches (du composant Board d'Azure) avec les commits
- ☐ Faire des commits et des partages de code (i.e., git push)
- ☐ Gérer les mises à jour (i.e., pull ou fetch) depuis les branches de dépôts locaux.

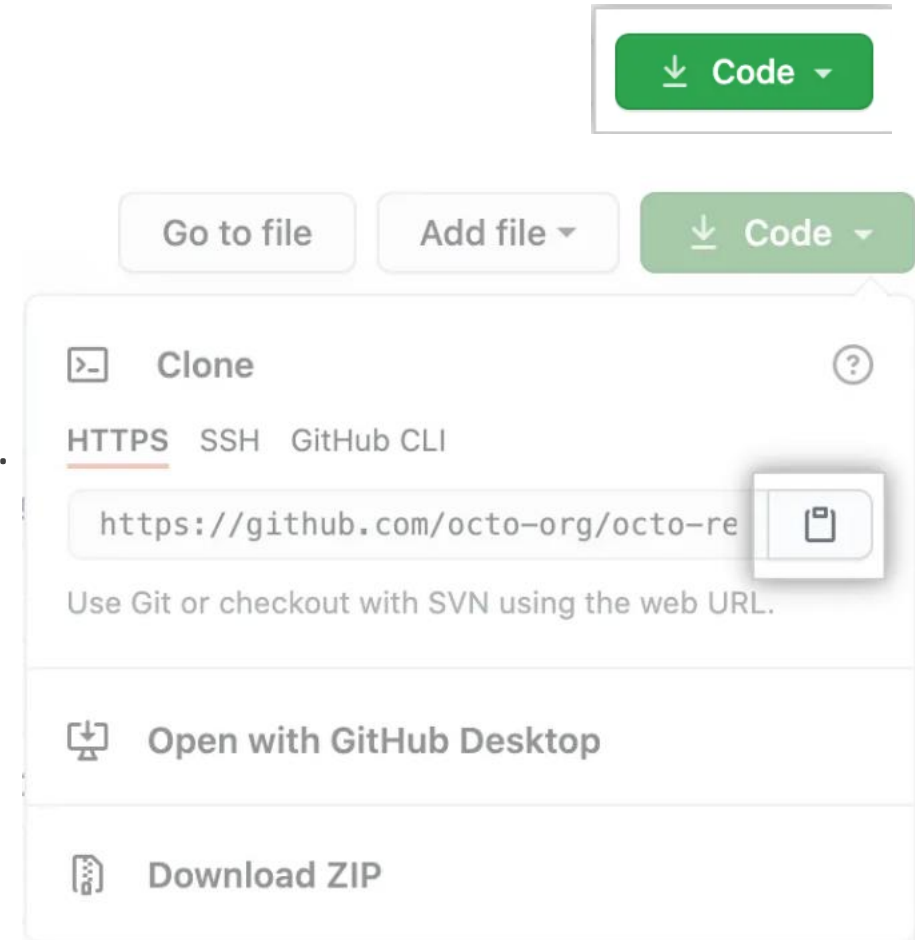
Cloner un dépôt Github

Dans GitHub.com, accédez à la page principale du **dépôt**. 1. Au-dessus de la liste des fichiers, cliquez sur

Pour cloner un dépôt, vous pouvez utiliser le protocole HTTPS, SSH ou aussi la CLI de Github.

Ouvrez Git Bash. Allez au répertoire de travail actuel, Tapez git clone, puis collez l'URL que vous avez copiée précédemment.

```
$ git clone https://github.com/YOUR-USERNAME/YOUR-REPOSITORY
```



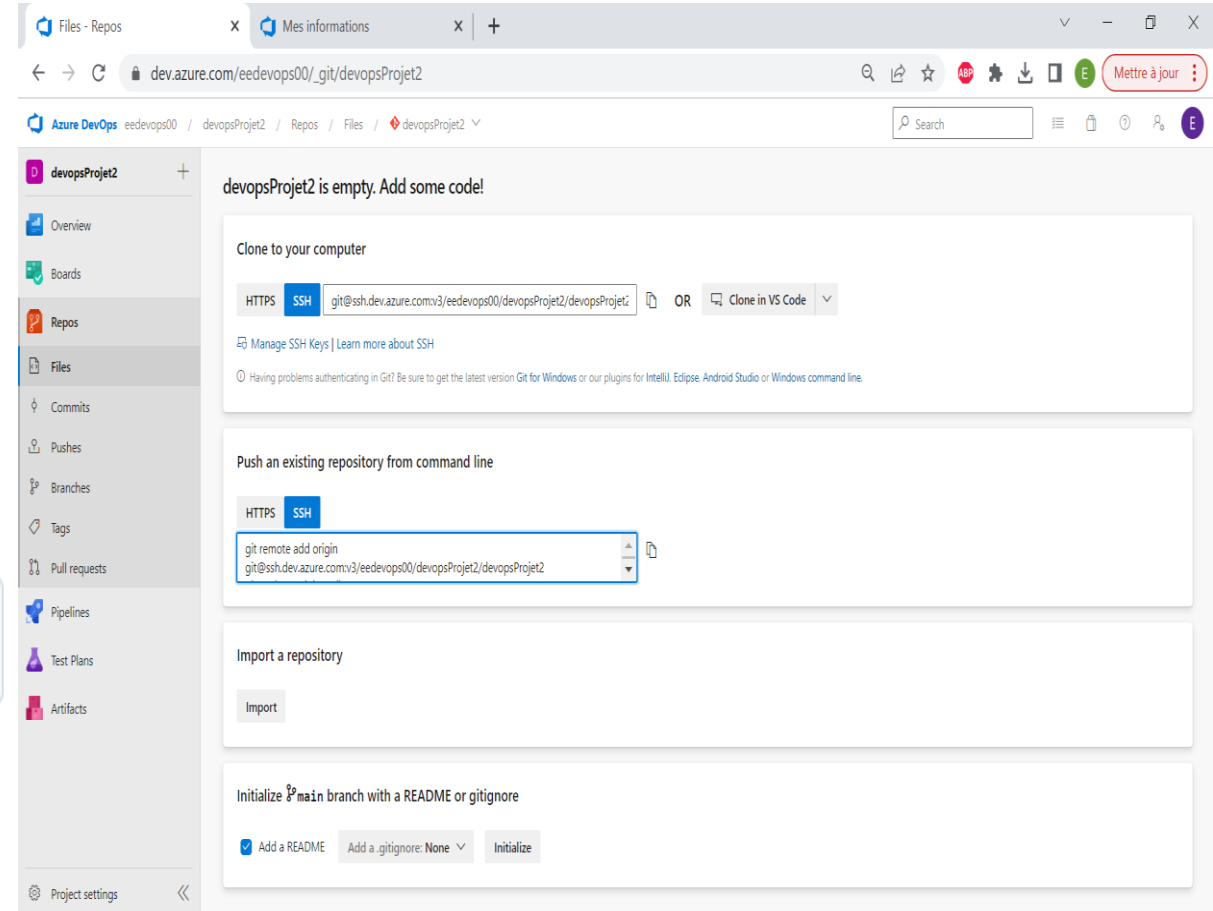
Cloner un dépôt Azure

Dans Azure DevOps, accédez à la page principale du **projet**. 1. Au-dessus de la liste des fichiers, cliquez sur

Pour cloner un dépôt, vous pouvez utiliser le protocole HTTPS, SSH.

Ouvrez Git Bash. Allez au répertoire de travail actuel, Tapez git clone, puis collez l'URL que vous avez copiée précédemment.

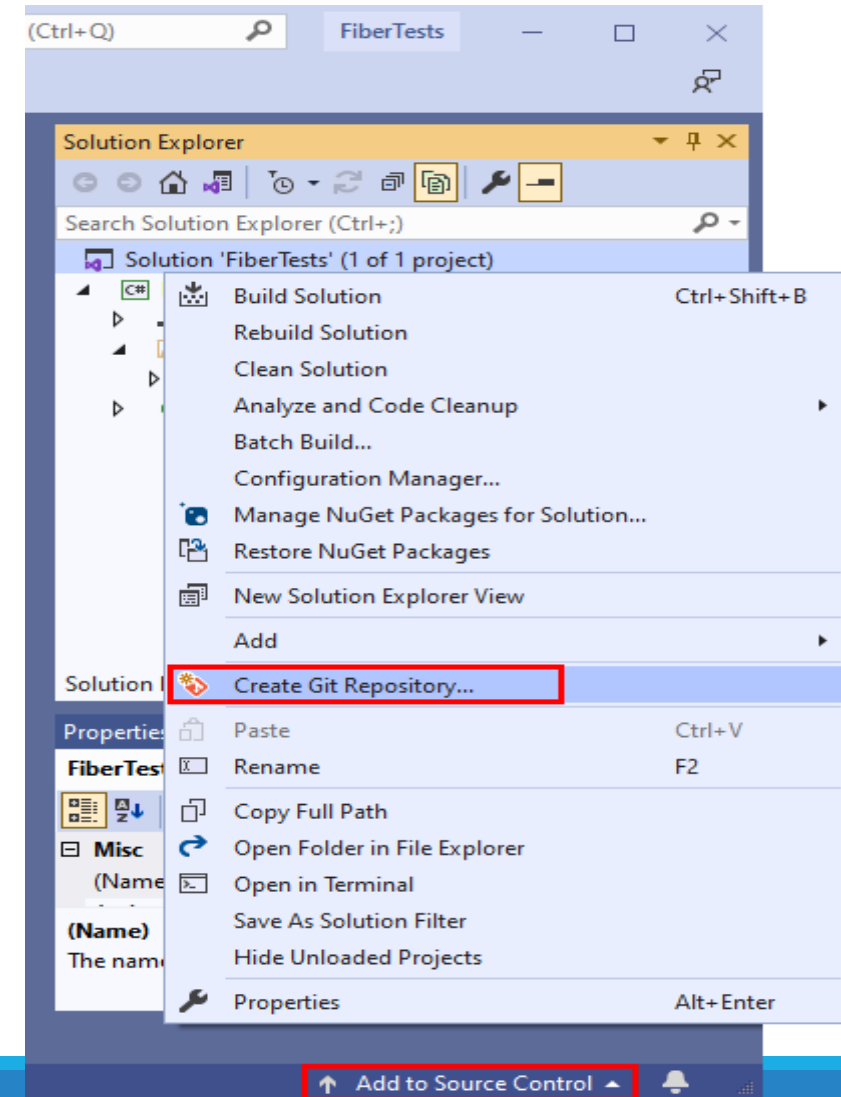
```
$ git clone https://github.com/YOUR-USERNAME/YOUR-REPOSITORY
```



Rendre un projet/solution un dépôt git local

1, Pour rendre une solution/projet un projet git, cliquer droit sur le nom de la solution dans l'explorateur de solution, puis **Create Git Repository**.

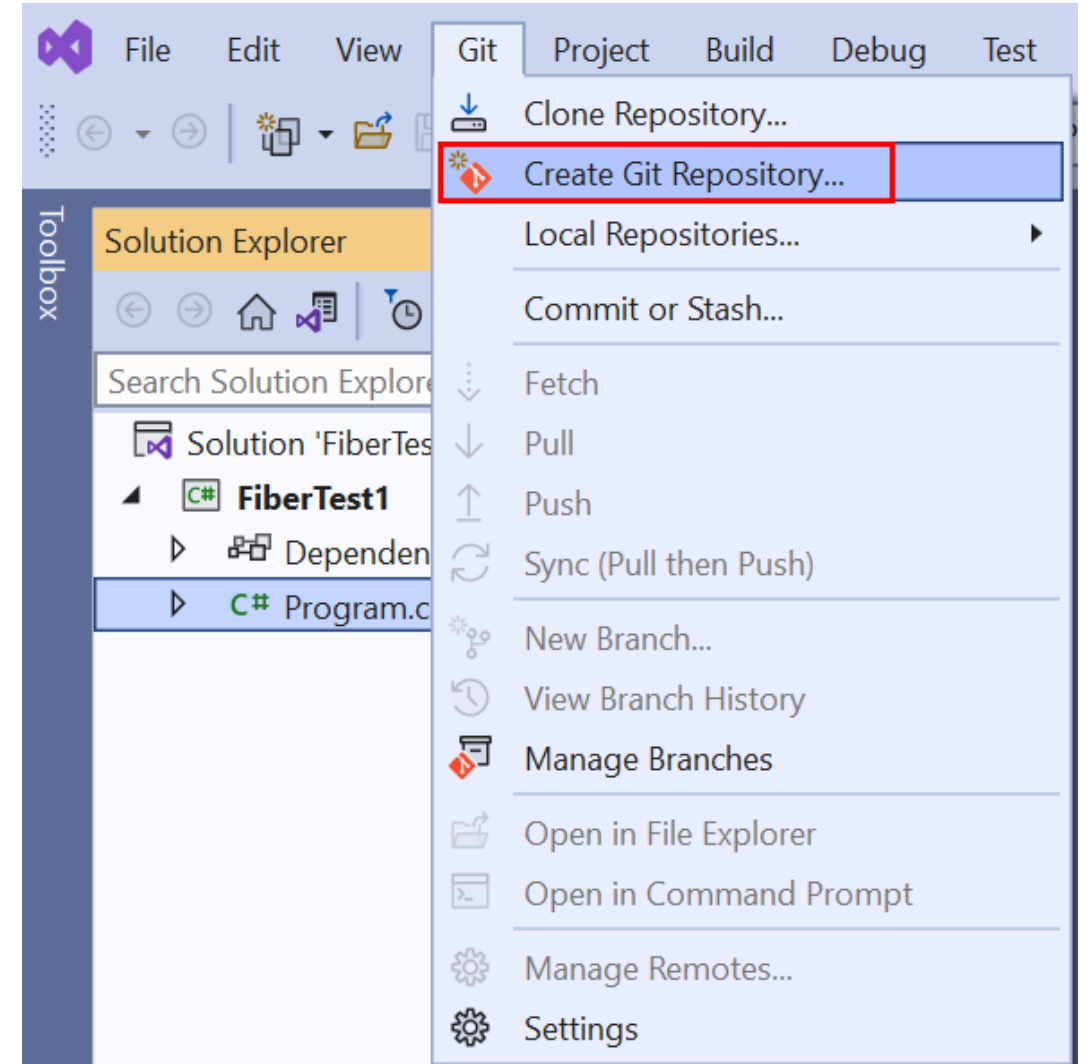
2, ou bien choisir **Add to Source Control** depuis la barre d'états, puis sélectionner Git. Si vous trouvez rien, c'est que la solution est déjà un dépôt git.



Rendre un projet/solution un dépôt git local

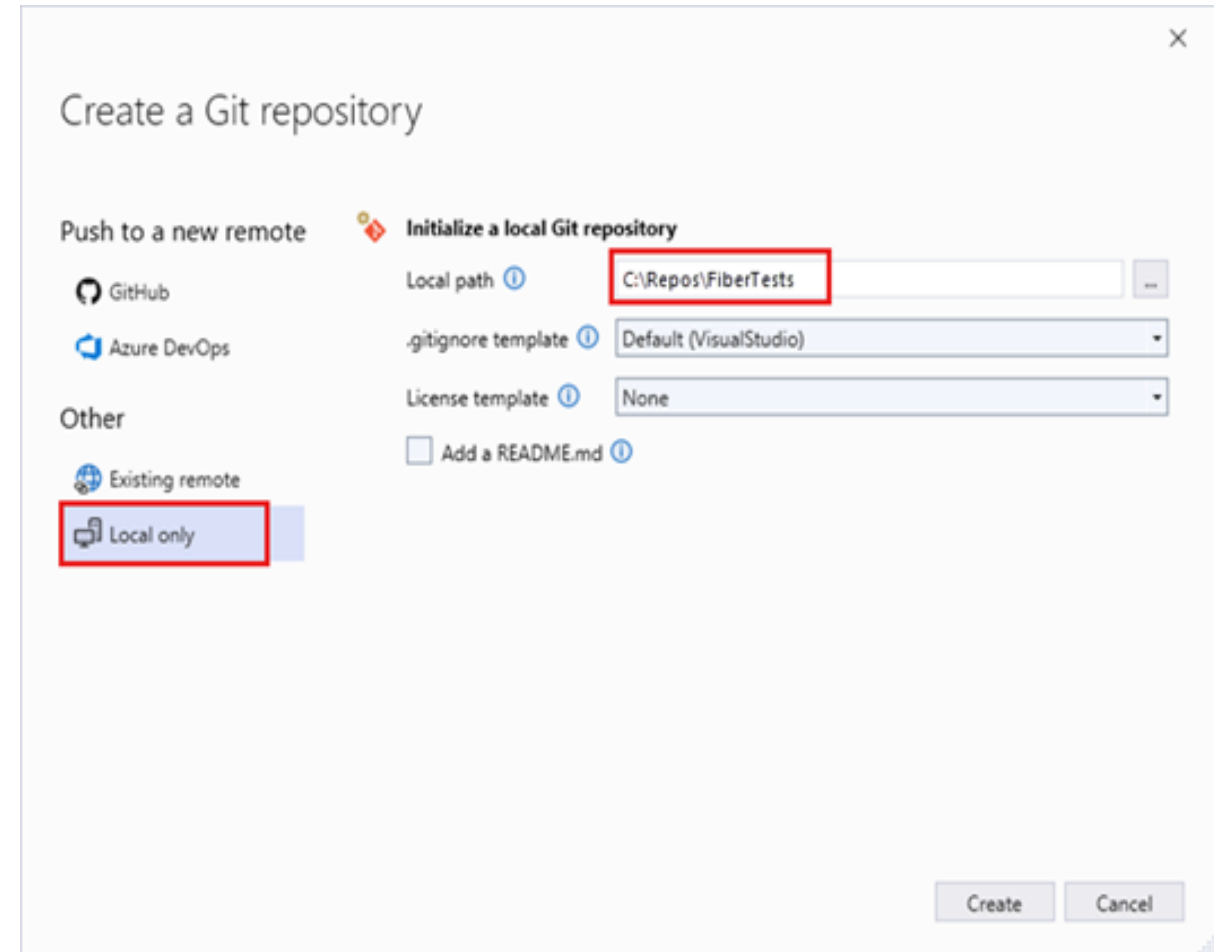
3, ou bien choisir **Git > Create Git Repository** depuis le menu Git dans la barre de menus pour lancer l'interface **Create a Git repository**.

Si vous voyez pas cette option, c'est que la solution est déjà un projet git.



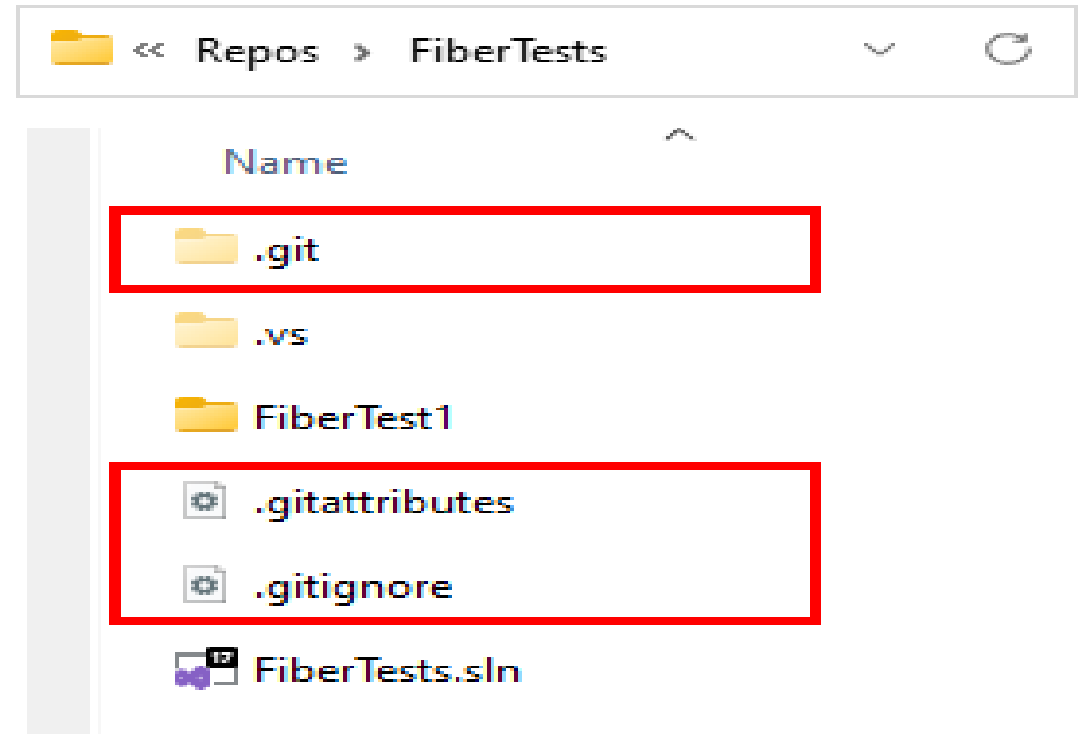
Ajouter un projet/solution à un nouveau dépôt git local

Cette étape s'applique uniquement à l'expérience de contrôle de version sur Visual Studio Git : dans la fenêtre **Create a Git repository**, choisissez **Local only**, vérifiez que le chemin d'accès local est correct, puis choisissez **Create**.



Ajouter un projet/solution à un nouveau dépôt git local

Vous avez maintenant créé un référentiel Git local dans le dossier de la solution Visual Studio et validé votre code dans ce référentiel. Votre référentiel Git local contient à la fois votre solution Visual Studio et les ressources Git.



Partager son travail

Vous pouvez partager votre travail avec d'autres personnes en publiant votre dépôt Git local dans un **dépôt Azure Repos Git**.

Visual Studio 2022 offre une expérience de contrôle de version Git à l'aide :

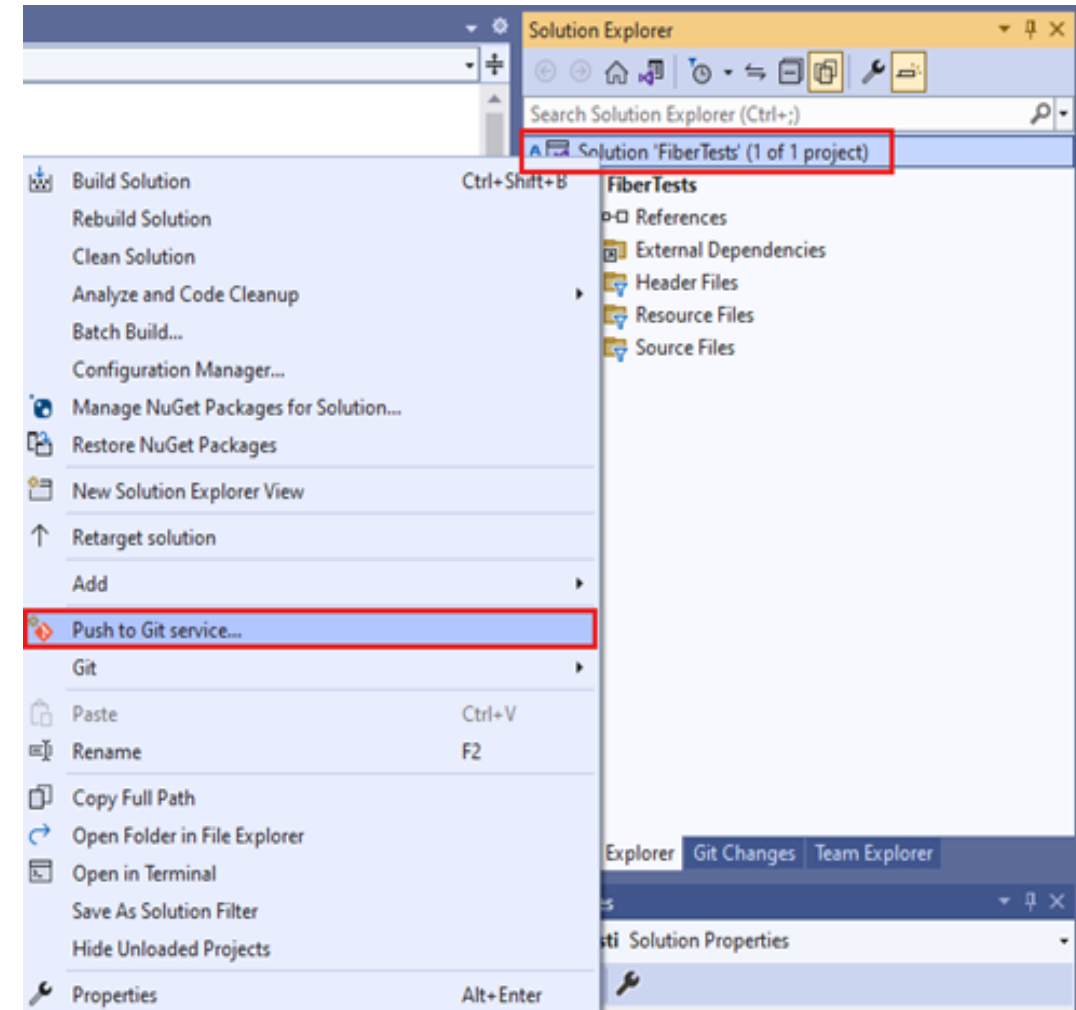
- du **menu Git**,
- des modifications Git dites **Git Changes**,
- et **des menus contextuels** de l'explorateur de solutions.

Dans ce qui suit, vous verrez les trois façons

Partager son travail

1,

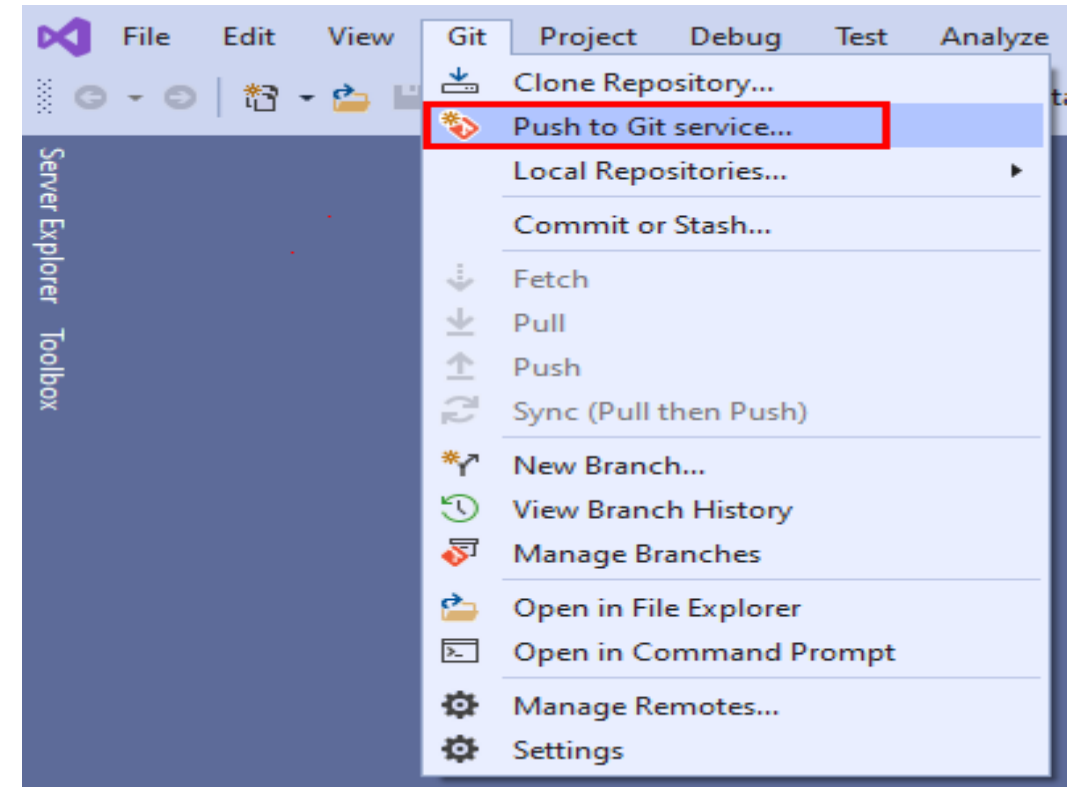
Connectez-vous à un référentiel Azure DevOps en cliquant avec le bouton droit sur le nom de la solution dans l'explorateur de solutions, puis en sélectionnant **Push to Git service** pour lancer la fenêtre **Create a Git repository**.



Partager son travail

2,

Ou choisissez **Git > Push to Git service** dans la barre de menus pour lancer la fenêtre **Create a Git repository**.



Partager son travail

Dans la boîte de dialogue **Create a Git repository**, sélectionnez **GitHub**. Saisissez les valeurs requises sous **Create a new GitHub repository**. Sélectionnez ensuite **Push**.

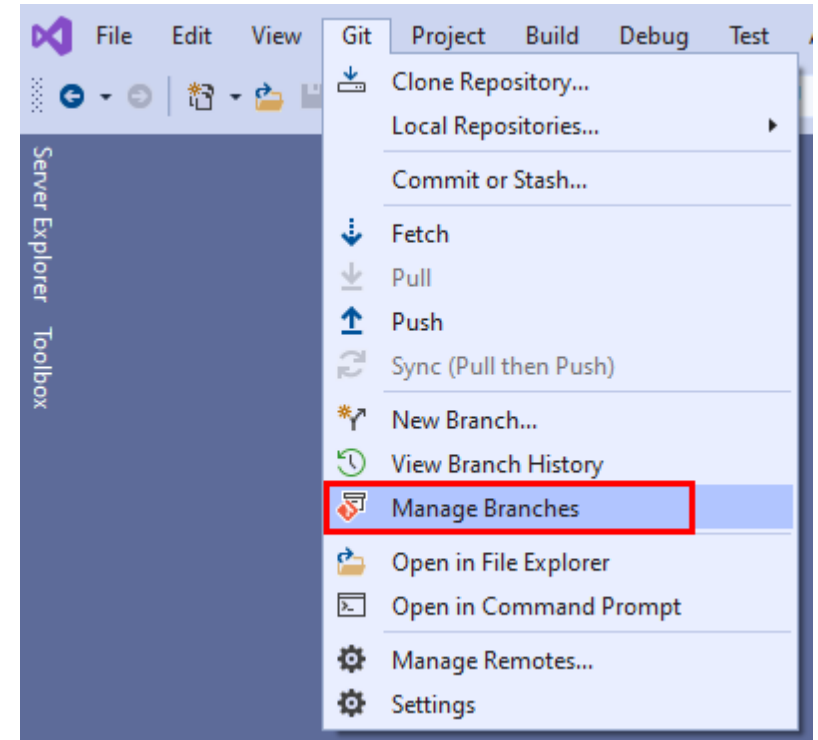
Vous pouvez vérifier que le référentiel est créé en accédant à GitHub et en recherchant le nom de votre référentiel dans votre compte.

The screenshot shows the 'Create a Git repository' dialog box. On the left, under 'Push to a new remote', the 'GitHub' option is selected. On the right, under 'Existing local Git repository', the 'Local path' is 'C:\Repos\FiberTests' and the 'License template' is 'None'. Below this, the 'Create a new GitHub repository' section is highlighted with a red box. It contains the following fields: 'Account' (FiberTestTeam (GitHub)), 'Owner' (FiberTestTeamAdmin), 'Repository name' (FiberTests), 'Description' (Fiber Tests project), and a checked 'Private repository' checkbox. At the bottom, the 'Push your code to GitHub' button is highlighted with a red box.

Révision de détails de commits

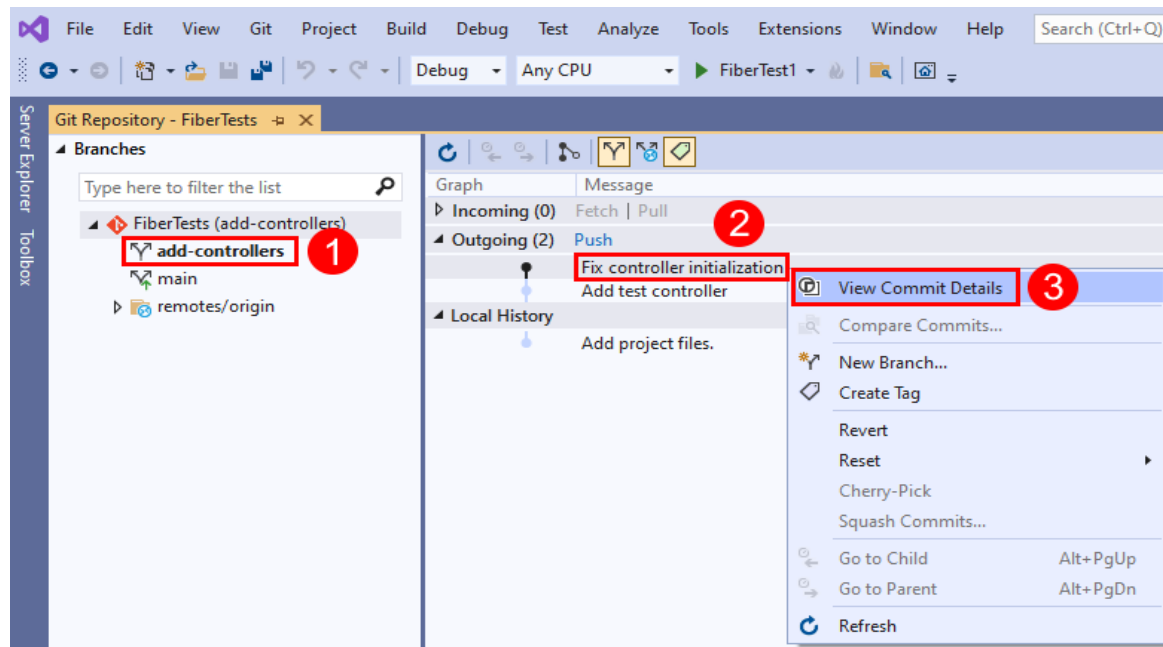
C'est une bonne pratique de revoir **les changements de code dans votre commit** avant de publier. Visual Studio fournit une vue de différence qui vous aide à voir toutes les modifications que vous avez apportées depuis la validation précédente (le dernier commit).

Pour voir ce qui a changé dans un commit, sélectionnez **Git > Manage Branches** pour ouvrir la fenêtre **Git Repository**.

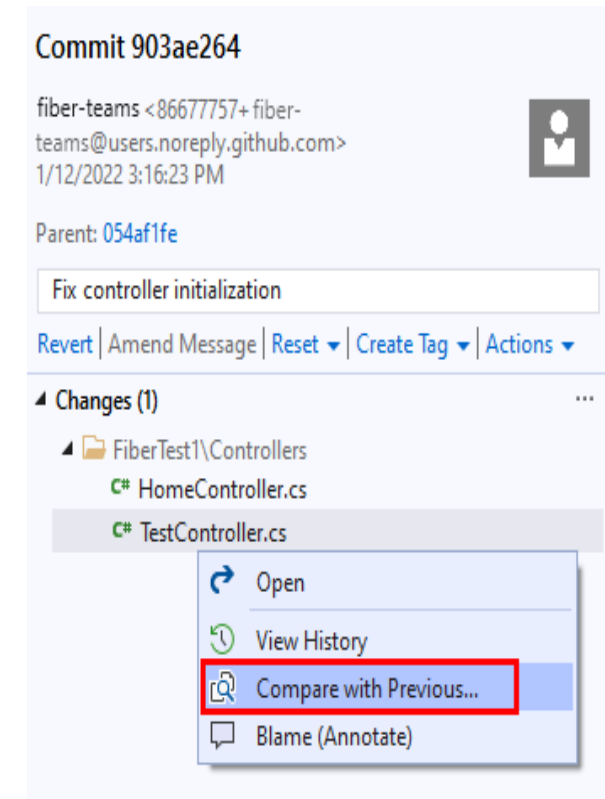


Révision de détails de commits

Dans la fenêtre **Git Repository**, sélectionnez votre branche, cliquez avec le bouton droit sur le commit, puis choisissez **View Commit Details** pour ouvrir la fenêtre **Commit**.

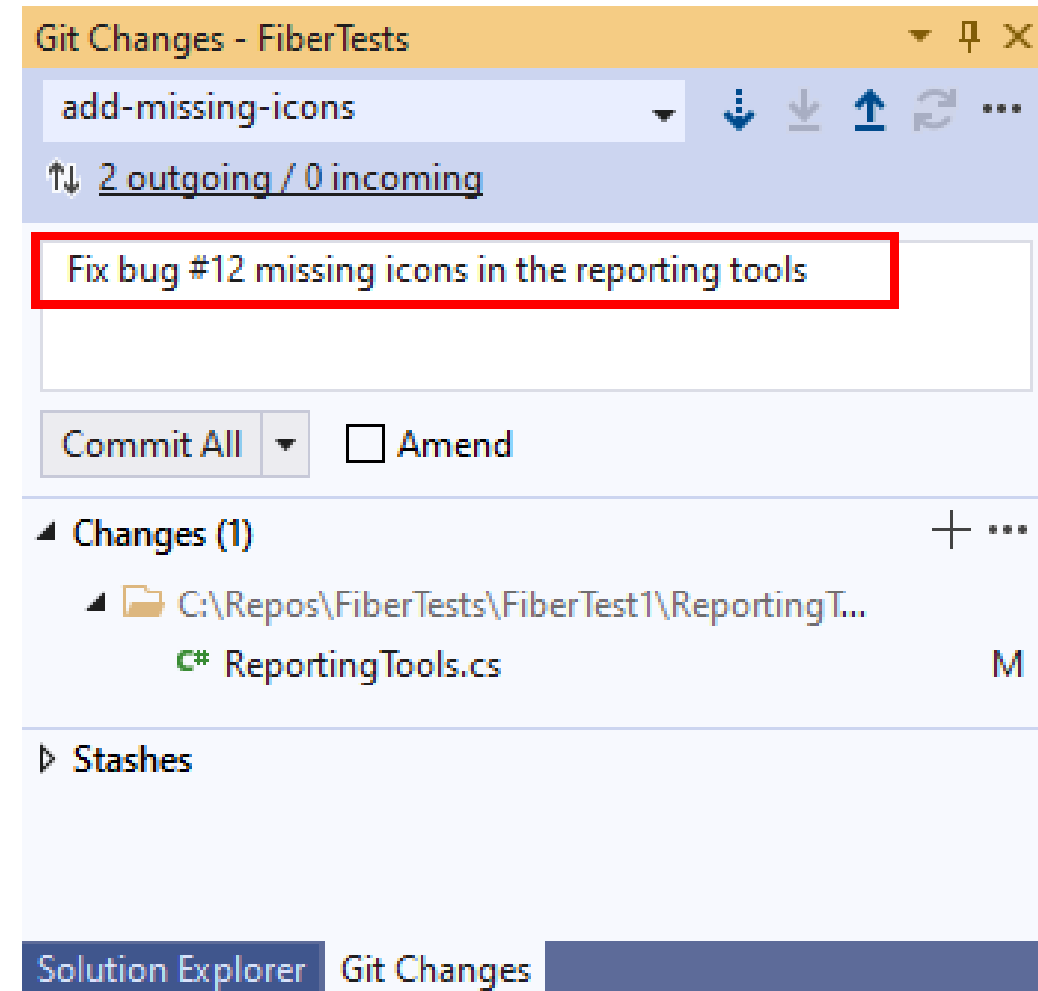


Dans la fenêtre **Commit**, vous pouvez cliquer avec le bouton droit sur n'importe quel fichier et sélectionner **Compare with Previous** pour afficher les modifications de fichier apportées par la validation (i.e. commit).



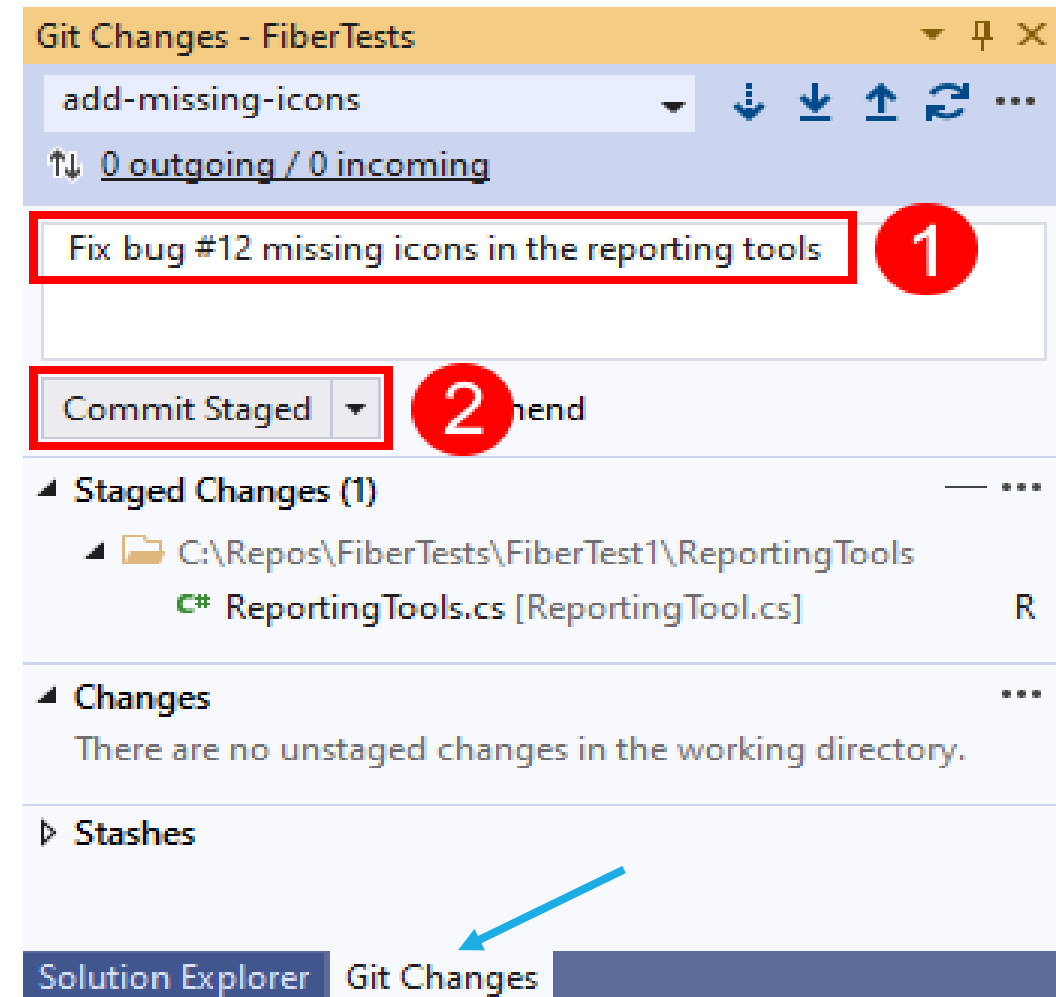
Lier les tâches avec les commits

Vous pouvez lier des éléments de travail (i.e., Work items) à des validations (i.e., commits) en ajoutant #ID dans le message de validation. Par exemple, le message de validation "Fix bug #12 Missing icons in the reporting tools" lierait l'élément de travail 12 à la validation lorsque la validation est transmise à **Azure Repos**.



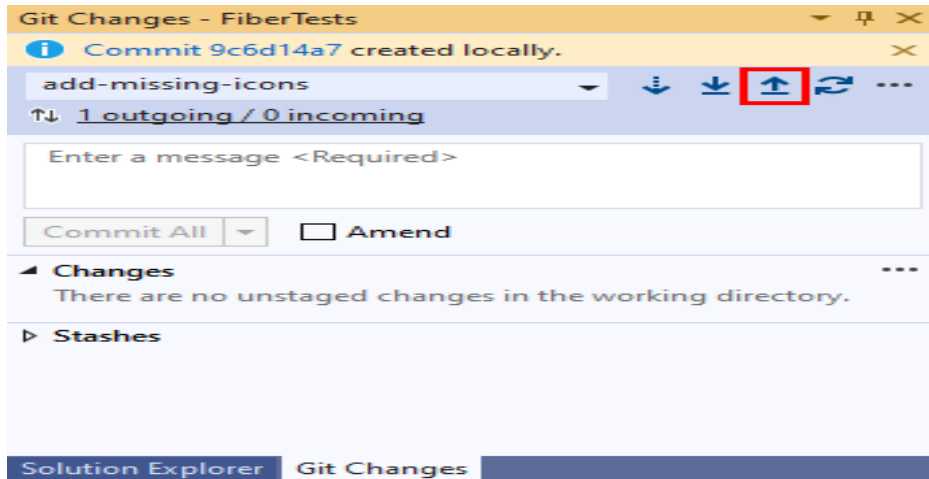
Envoyer au dépôt distant

Dans la fenêtre **Git Changes**, saisissez un message décrivant les modifications, puis choisissez **Commit All**. **Commit All** valide (i.e., `$git commit`) les modifications non indexées et ignore l'espace d'indexation. Vous pouvez choisir d'indexer tous les fichiers avant de valider en sélectionnant le bouton + (plus) en haut de la section **Changes** de la fenêtre **Git Changes**.

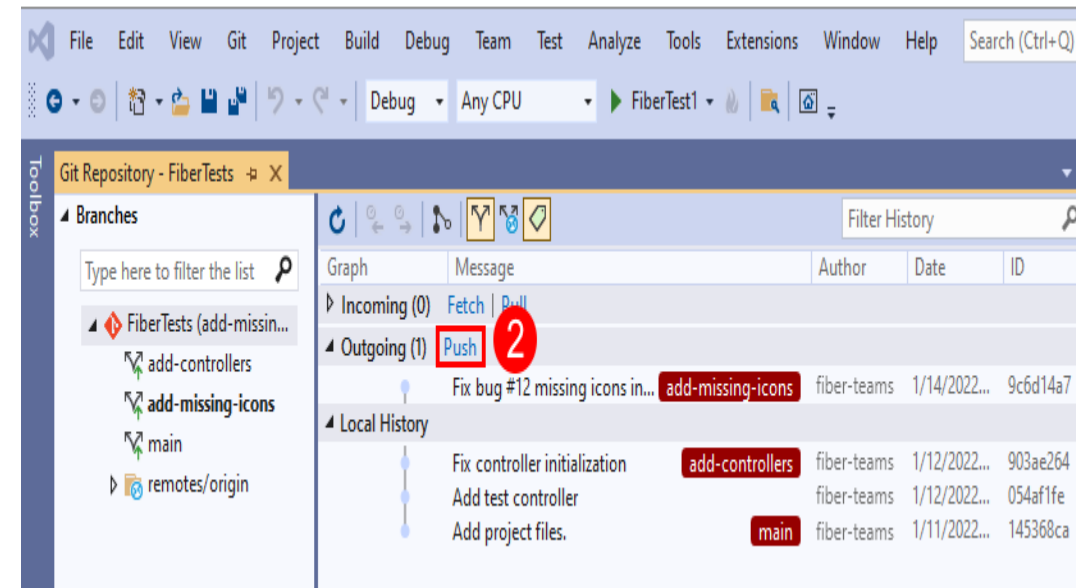
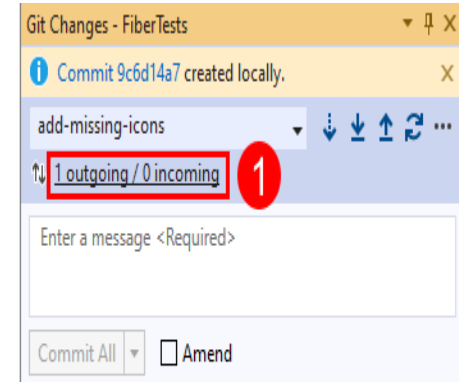


'''

Pour envoyer votre commit vers **Azure Repos**, sélectionnez le bouton flèche vers le haut.

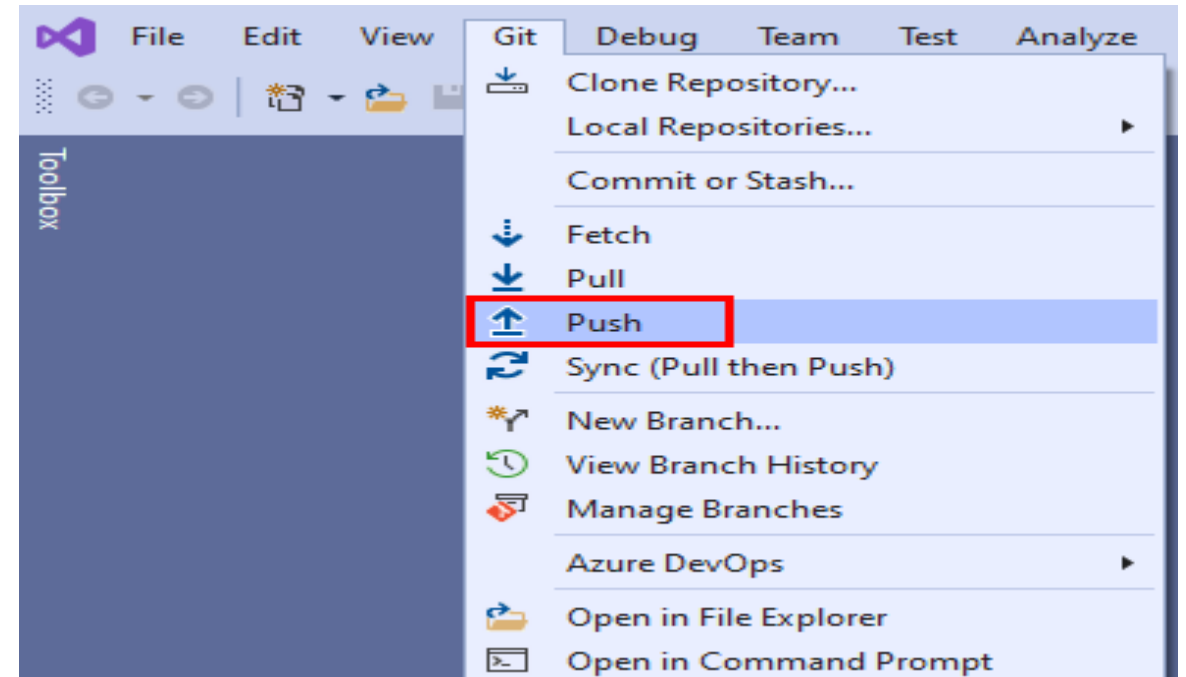


Ou, vous pouvez envoyer votre commit depuis la fenêtre du **Git Repository**. Pour ouvrir la fenêtre **Git Repository**, sélectionnez le lien **outgoing / incoming** dans la fenêtre **Git Changes**.



'''

Ou, vous pouvez choisir **Git > Push** dans la barre de menus.



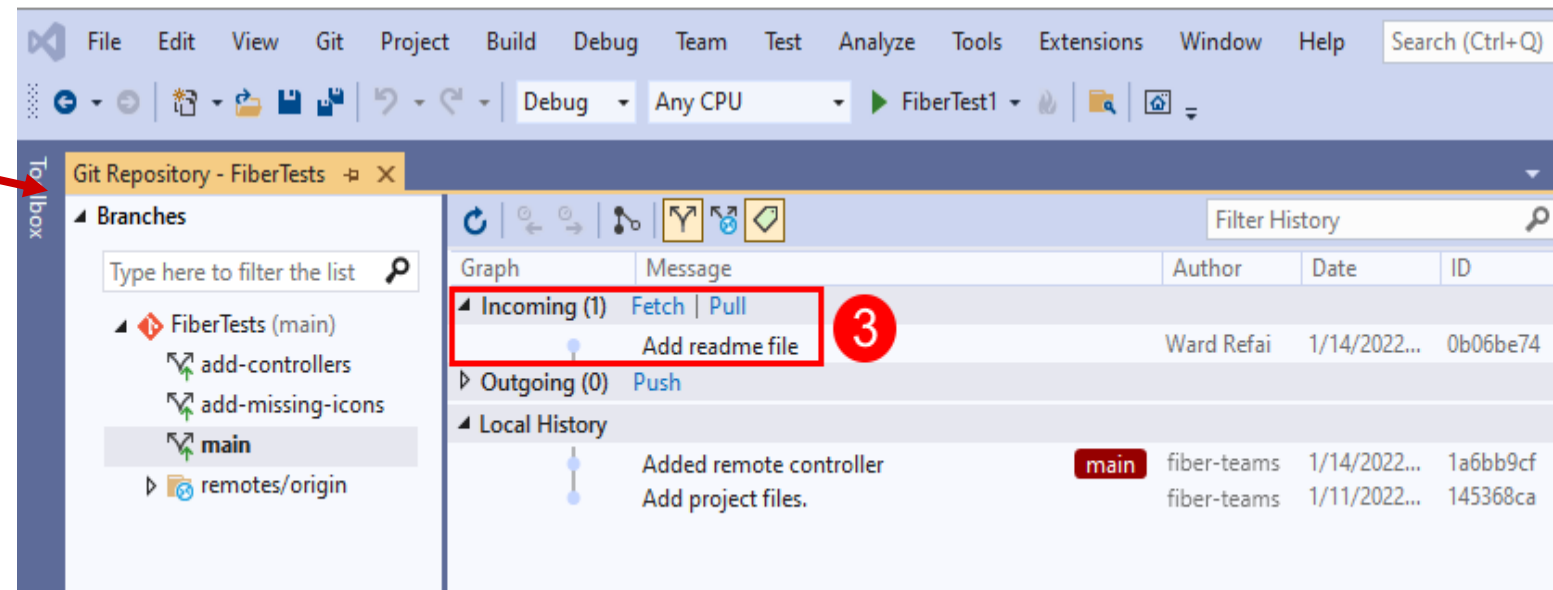
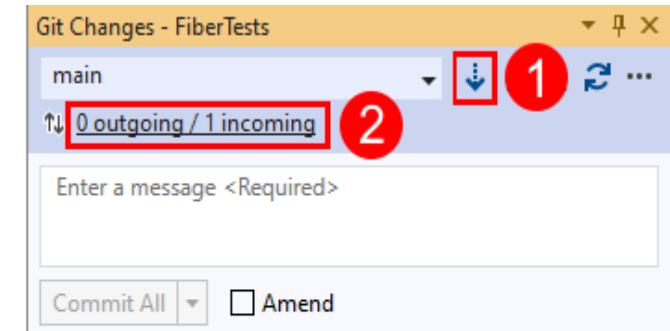
Mettre à jour son dépôt

Il est judicieux de basculer périodiquement vers votre branche principale (main/master) et de la mettre à jour avec la branche principale **distante**.

Utiliser la flèche en bas pour chercher les nouveautés dans la branche distante. Sélectionner ensuite le lien outgoing/incoming pour ouvrir la fenêtre **Git Repository** window.

Dans cette fenêtre, vous pouvez utiliser soit la commande **Fetch** ou la commande **Pull**.

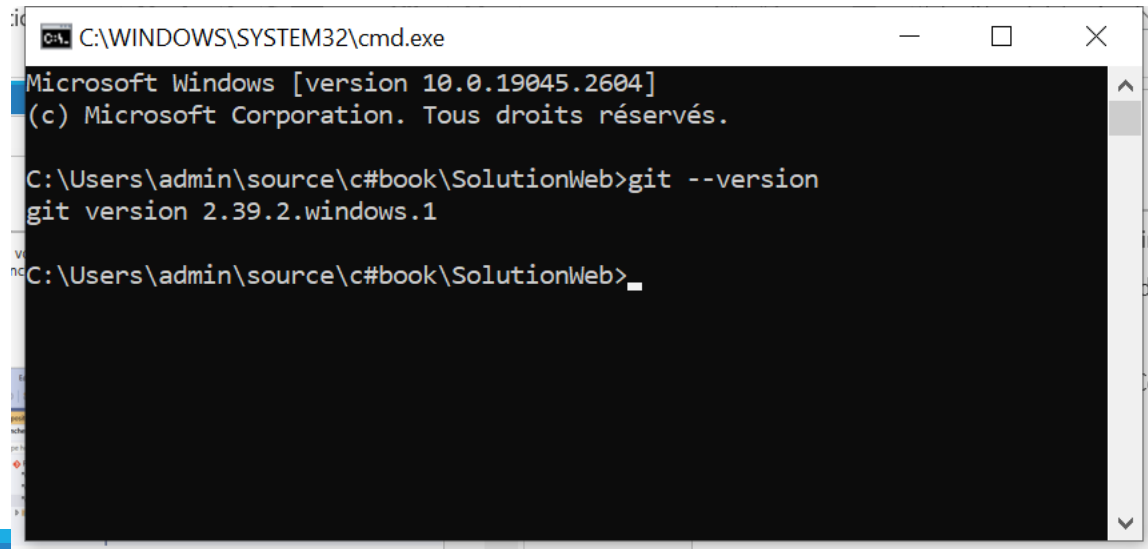
Les changements issues de Fetch peuvent être vues dans la section **Incomming Commits**, mais ils ne sont pas fusionner dans la branche locale.



Travailler en CMD

De temps en temps, vous pouvez avoir besoin de travailler avec l'invite de commande (e.g., blocage d'IDE, familiarisation, une commande absente sur IDE ou autres).

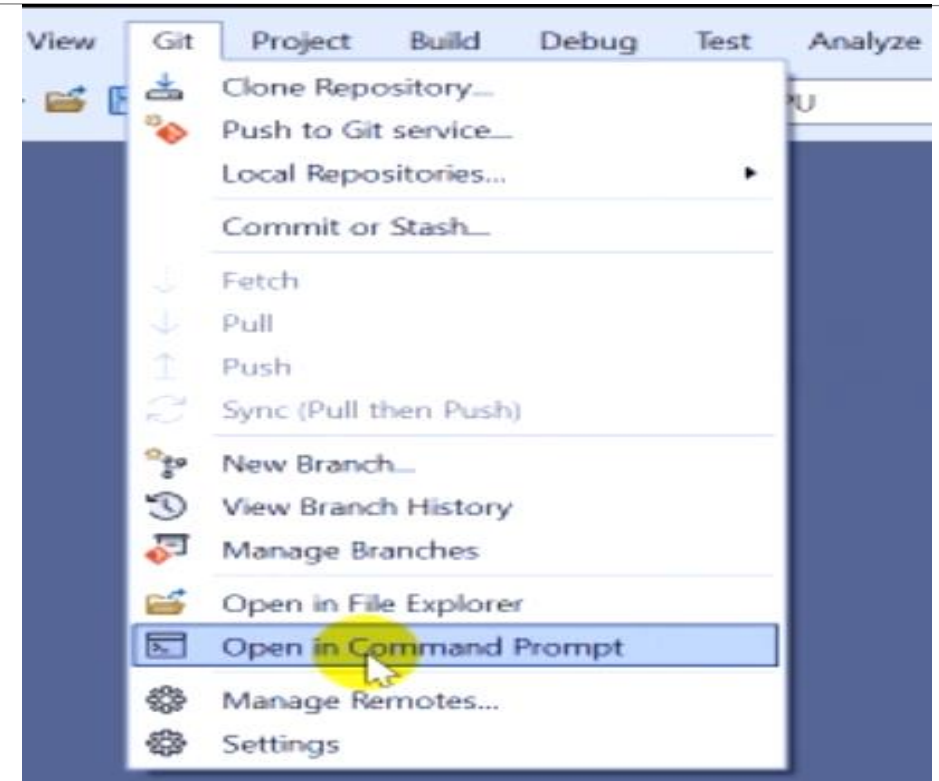
Dans ce cas, Aller vers Git > Open In Command Prompt.



```
C:\WINDOWS\SYSTEM32\cmd.exe
Microsoft Windows [version 10.0.19045.2604]
(c) Microsoft Corporation. Tous droits réservés.

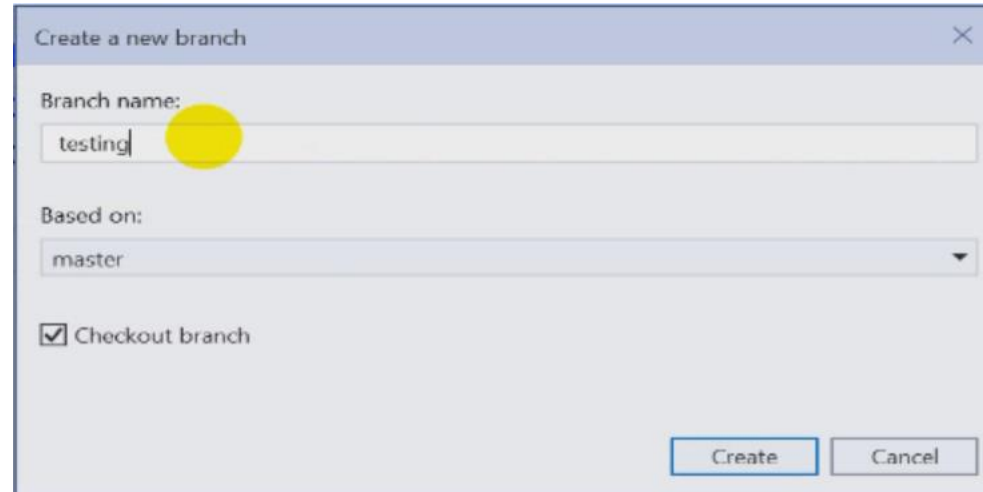
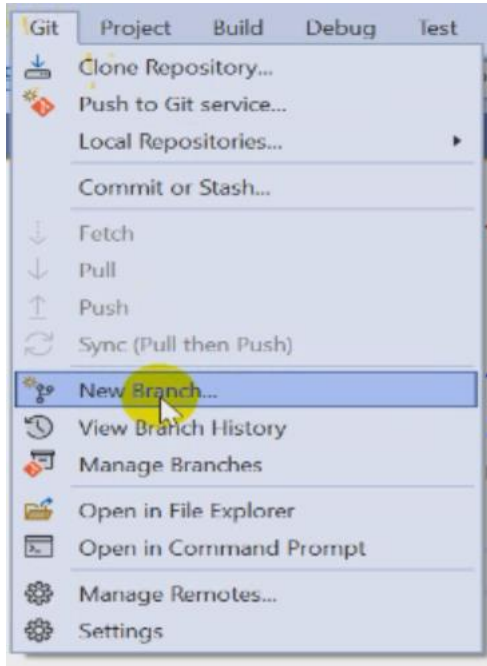
C:\Users\admin\source\c#book\SolutionWeb>git --version
git version 2.39.2.windows.1

C:\Users\admin\source\c#book\SolutionWeb>
```

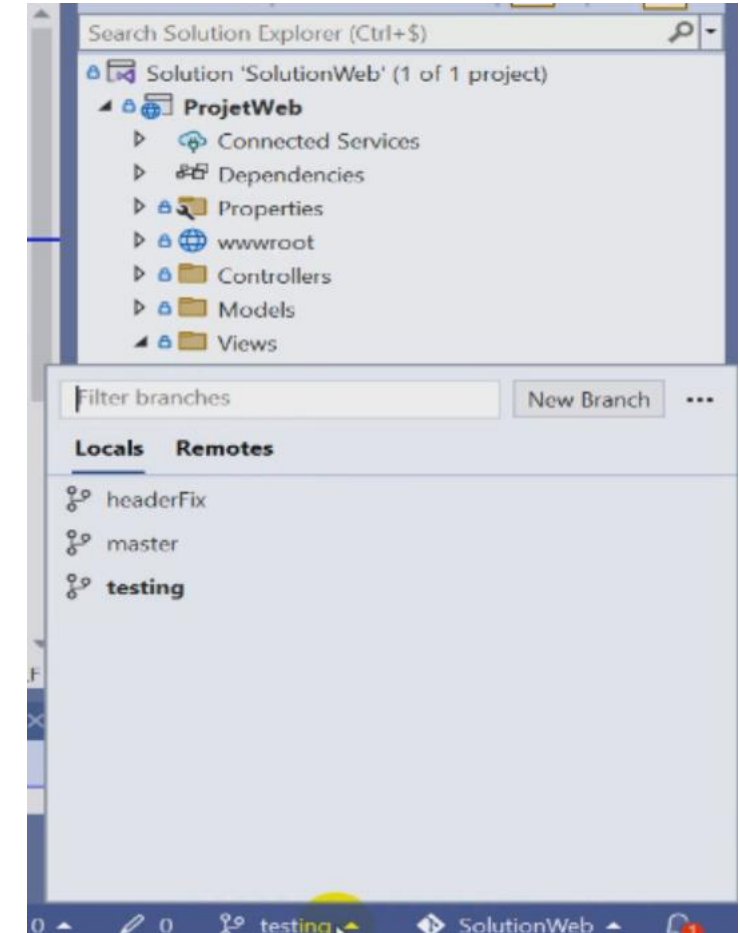


Gestion des branches

Créer une branche en allant sur Git > New Branch



Pour se déplacer vers une branche, utiliser la flèche en bas.

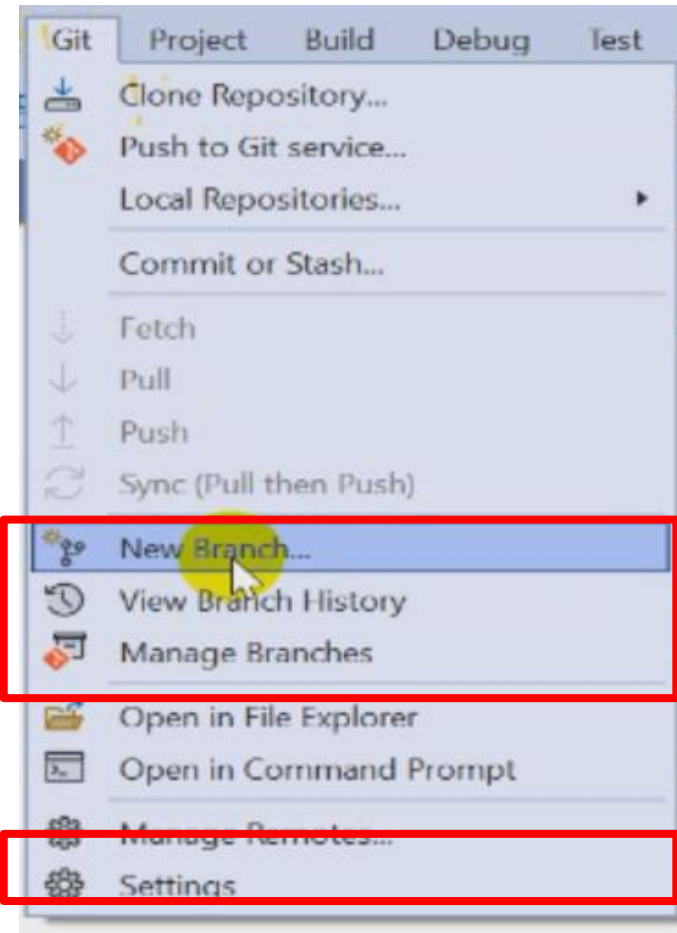


Gestion des branches

Pour gérer l'historique des branches, utiliser « View Branch History »

Pour gérer les branches locales, utiliser « Manage Branches »

Pour gérer les branches alias des branches distantes, utiliser « Manage Remotes »



Vous pouvez également gérer les branches locales et alias depuis cette fenêtre.

