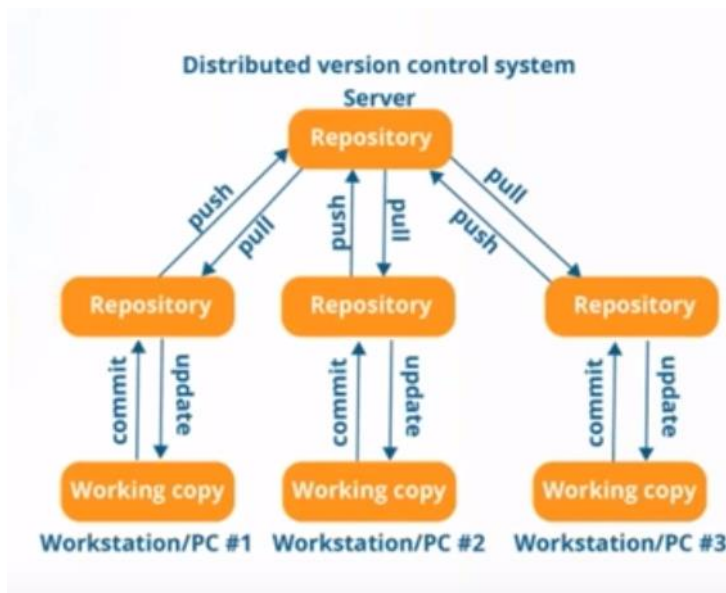


Tuto Git

QUELQUES INFOS POUR GIT

Bitbucket (tel que github) est un service pour héberger des dépôts (= répertoires pour projet) et git est l'outil qui permet de créer un dépôt local et de gérer les versions.

- Git est outil de système de gestion de version, il permet d'envoyer et de recevoir des données.
- Bitbucket est une entreprise qui vous permet d'héberger vos repositories et collaborer avec d'autres personnes, il permet aussi aux développeurs de partager des codes sources et des programmes.



Commandes Git

Git init

Cette commande permet de transformer un répertoire à un dépôt Git vide (initialisation)

Git commit

Cette commande permet d'enregistrer les changements effectués dans le dépôt local.

Git status

Une commande qui retourne à l'état actuel du dépôt

Si y a pas de changement, un message s'affiche: "nothing to commit, working directory clean".

Git add <fichier>

Git add permet de préparer le contenu pour faire un 'commit'

Git pull

Mettre à jour le contenu depuis le dépôt distant.

Git push

Mettre à jour le changement local dans le dépôt distant.

INDICATIONS POUR BITBUCKET /GIT

1. Création du repository on-line sur le serveur

Un des membres va créer le répertoire commun pour le projet : le repository ! Observez bien tous les manipulations !

Il va devoir tout d'abord **se créer un compte sur bitbucket avec le mail de l'IUT** et choisir la solution gratuite. (n'utilisez pas de compte google, créez un compte !)

Lors de la création, vous ferez attention à laisser un accès publique (décochez la case private repository)

Create a new repository [Import repository](#)

Owner nathalie.gruson

Repository name* SEM_SPE_NGRUS

Access level ☐ This is a private repository
Uncheck to make this repository public. Public repositories typically contain open-source code and can be viewed by anyone.

Include a README? No

Version control system ☒ Git ☐ Mercurial

[Advanced settings](#)

[Create repository](#) [Cancel](#)

Puis demandez la création du fichier readme et .gitignore.

Vous devriez arriver sur cette fenêtre, elle vous donne la commande pour pouvoir cloner le repository sur votre machine, vous pouvez à tout moment la retrouver. Veillez à sélectionner https.

SEM_SPE_TEST_NAT

Source

Pipelines

Downloads

Settings

Let's put some bits in your bucket

HTTPS

Get started quickly

Creating a README or a .gitignore is a quick and easy way to get something into your repository.

[Create a README](#) [Create a .gitignore](#)

Get your local Git repository on Bitbucket

Step 1: Switch to your repository's directory

```
1 cd /path/to/your/repo
```

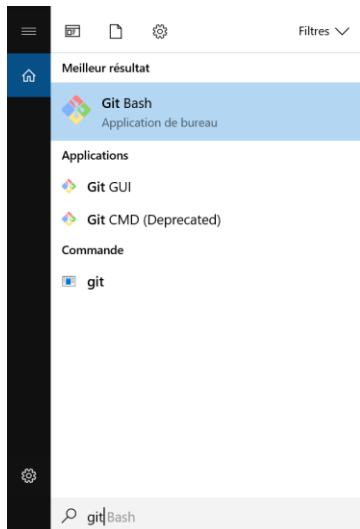
Step 2: Connect your existing repository to Bitbucket

```
1 git remote add origin git@bitbucket.org:ngrus/sem_spe_test_nat.git
2 git push -u origin master
```

2. Création du repository local sur votre machine

Pour tester, le créateur du repository va créer un répertoire de travail sur son P :

Puis déclenchez git bash.



Se positionnez à l'aide de `cd` dans son répertoire de travail

Puis copiez collez l'ordre pour cloner le repository

3. Communication repository local => repository on line

Exemple :

Maintenant, vous allez créer un fichier en local « `nom_prenom.txt` » contenant 2, 3 lignes de textes (Ces fichiers seront supprimés par la suite) puis vous allez l'exporter sur le repository partagé « on-line », pour cela, il faut :

- Ajoutez le fichier au projet git en local : `git add nom_prenom.txt`
- Confirmez la volonté de valider : `git commit -m « ajout nom_prenom.txt »`. Il faut toujours mettre un message.
- Envoyez les modifications : `git push`
- Observez l'ajout du fichier sur le site bitbucket

A chaque fois que vous êtes désireux d'envoyer toutes vos modifications, il faut faire ses 3 actions. Vous pouvez faire `git add .` si vous voulez prendre en considération tous les fichiers (c'est à ce moment là que le `.gitignore` peut être utile pour indiquer les fichiers ou répertoires que vous ne voulez pas envoyer. Ex : `bin/`)

4. Communication repository on line => repository local

Exemple : éditez le fichier « `nom_prenom.txt` » depuis le site bitbucket, ajoutez 1 ligne de texte. Sur votre machine, depuis git bash :

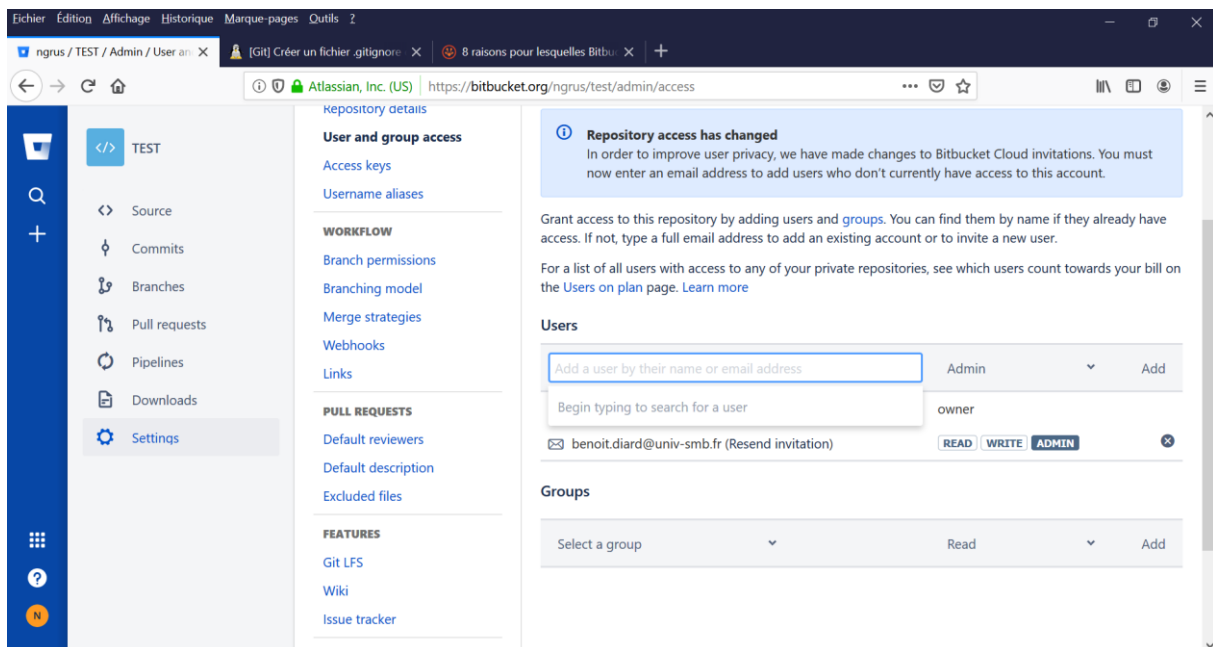
- Demandez une mise à jour : `git pull`
- Ouvrez ensuite votre fichier vous devriez voir les modifications

5. Création des autres utilisateurs

Le responsable du repository sur le serveur va ajouter des droits admin pour les autres membres du groupe qui vont recevoir un mail (utilisez le mail universitaire). A la réception du mail, les autres membres doivent juste créer un compte (pas de compte google) et valider l'invitation ce qui les emmène sur le repository automatiquement (si cela n'a pas marché, rouvrir le mail et cliquer sur le lien)

Les autres membres du groupe peuvent alors répéter les étapes 2 à 4.

Ensuite vous supprimerez depuis le serveur les fichiers nom_prenom.txt ou bien vous recréerez un autre repository.



Remarques importantes :

Si vous avez ce genre de message, c'est qu'il y a eu d'autres modifications sur le serveur.

Vous devez récupérer avant d'envoyer : faites un git pull puis à nouveau un git push.

```
MINGW64:/c/Users/ngrus/Desktop/SEM_SPE/sem_spe_test_nat
$ git add .

ngrus@pinf-pngrus MINGW64 ~/Desktop/SEM_SPE/sem_spe_test_nat (master)
$ git commit -m "dzdz"
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

ngrus@pinf-pngrus MINGW64 ~/Desktop/SEM_SPE/sem_spe_test_nat (master)
$ git push
To https://bitbucket.org/ngrus/sem_spe_test_nat.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://ngrus@bitbucket.org/ngrus/sem_spe_test_nat.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

ngrus@pinf-pngrus MINGW64 ~/Desktop/SEM_SPE/sem_spe_test_nat (master)
$ |
```

Si vous rencontrez un CONFLICT (comme ci-dessous), utilisez la commande :

Git mergetool

```
MINGW64:/c/Users/ngrus/Desktop/DEPOT/sem_spe_test_nat
ngrus@pinf-pngrus MINGW64 ~/Desktop/DEPOT/sem_spe_test_nat (master)
$ git pull
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From https://bitbucket.org/ngrus/sem_spe_test_nat
 53a2fb7..ac90726  master    -> origin/master
Auto-merging sources.txt
CONFLICT (content): Merge conflict in sources.txt
Automatic merge failed; fix conflicts and then commit the result.

ngrus@pinf-pngrus MINGW64 ~/Desktop/DEPOT/sem_spe_test_nat (master|MERGING)
$ git mergetool

This message is displayed because 'merge.tool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
'git mergetool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffuze diffmerge ecme
rge p4merge araxis bc codecompare smerge emerge vimdiff
Merging:
sources.txt
```

Tapez sur entree , cela va déclencher vimdiff (vi pour voir les différences entre la version serveur et local), les différences sont surlignées en rouge.

Contenu du fichier en local	Contenu du fichier en local avant la modification	Contenu du fichier sur le serveur
Le merge des 2 à modifier et sauvegarder		

Dans le cadre du dessous, vous avez les éléments en conflit <<<<<<HEAD est suivi du contenu en local, ===== est suivi du contenu sur le serveur.

A vous de faire le choix, tout garder, supprimer quelque chose, mais il faut enlever les <<<<<<HEAD =====>>>>>> .Pour modifier, Touche inser pour écrire puis echap, :x pour sauvegarder, puis :q ! pour fermer tous les fichiers.

Exemple ci-dessous : en gardant les 2 modifs

```
MINGW64:/c/Users/ngrus/Desktop/DEPOT/sem_spe_test_nat

ana
lucie
marc
martin
bruno
~
~
~
<:37 18/11/2019)5,6-1 Tout <:37 18/11/2019)4,7-1 Tout <:37 18/11/2019)5,9-1 Tout
ana
lucie
marc
martin
bruno
benjamin
~
~
~
liste.txt[+] [dos] (13:37 18/11/2019) 6,9 Tout
-- INSERTION --
```

Une fois le conflit résolu, pensez à git add, git commit -m « commentaires », git push, vous verrez ensuite sur bitbucket les « merges », branches // qui fusionnent.

Author	Commit	Message	Date
nathalie.gruson	68302b4	fusion	5 min
nathalie.gruson	dae1b8e	modif avec conflit	36 mi
nathalie.gruson	ac90726	sources.txt edited online with Bitbucket	37 mi
nathalie.gruson	53a2fb7	modif	43 mi
nathalie.gruson	25c0967	modif	49 mi
nathalie.gruson	b74c1d3	sources.txt edited online with Bitbucket	51 mi
nathalie.gruson	3656770	sources.txt edited online with Bitbucket	20 hc