# Initiation à l'utilisation de Git

## Les bases de Git

Gauthier BIEHLER

Président du BDTech
ENSEA

Présentation du 26 octobre 2023


# Plan de présentation

# C'est quoi Git ?

- Notions de VCS
- Installation de Git
- Fonctionnalités abordées

# Notions de VCS

## VCS

Version Control System, outil servant à suivre et gérer les modifications de fichiers au cours du temps.

## Pourquoi Git

- Plus utilisé des VCS depuis 2013.
- Disponible sur tous les OS.
- Facile à prendre en main.

# C'est quoi Git ?

- Notions de VCS
- Installation de Git
- Fonctionnalités abordées

# Installation de Git (Linux)

## Installation Debian

sudo apt-get install git

## Installation hors Debian

- Télécharger Git sur https://git-scm.com
- Lancer l'installation classique

## Tester l'installation

git –version

# Installation de Git (Max OS)

## Installation

- Télécharger Git sur https://git-scm.com
- Lancer l'installation classique
- Dans le terminal : alias git = "/usr/local/git/bin/git"

## Tester l'installation

git –version

# Installation de Git (Windows)

## Installation

- Installer Git Bash sur https://gitforwindows.org/
- Lancer l'installation classique

## Tester l'installation

git –version

# C'est quoi Git ?

- Notions de VCS
- Installation de Git
- Fonctionnalités abordées

# Fonctionnalités abordées

## Commandes de manipulation

- git init
- git add
- git rm
- git commit

## Commandes porcelaine

- git clone
- git pull
- git push

## Commandes plomberie

Lors d'autres séances. Certaines seront vues lors de cette séance mais non expliquées.

# Fonctionnement basique

- Initialisation du compte
- Vision des fichiers
- Les zones de fichiers

# Initialisation du compte et du dépôt

Avant tout, vous devrez vous créer un compte sur https://github.com.

## Configuration du compte

git config –global user.name <user_name>
git config –global user.email <user@mail.com>

## Créer un dépôt

- Cliquer sur New repository dans le dashboard

- git clone https ://github.com/<user_name/<repository>

OU

- git init , touch README.md , git add README.md
- git remote add origin https ://github.com/<user_name/<repository>
- git commit -m "Initial commit" , git push -u origin master

# Fonctionnement basique

- Initialisation du compte
- Vision des fichiers
- Les zones de fichiers

# Git et les fichiers

## Hash

Signature du fichier calculée en fonction du contenu du fichier. Chaîne de 40 caractères hexadécimaux (160 bits).
Configurable sur 64 caractères (256 bits) avec :

git init –object-format=sha256

## Conflit de Hash

Deux fichiers ayant un Hash identique. Quasiment impossible en pratique.

$$P_{conflit}(n) = P_{conflit}(n-1) + \frac{n(1 - P_{conflit}(n-1))}{2^{160}} \; ; \; P_{conflit}(1) = 0$$

# Fonctionnement basique

- Initialisation du compte
- Vision des fichiers
- Les zones de fichiers

# Répertoire de travail

Il s'agit du dossier dans lequel se trouve le fichier .git.

## Manipulation des fichiers

touch <file>
mkdir <dir>
rm <file>
rm -f -r <dir>
git mv <old file> <new file>

## Vision du contenu

ls
git status

# Index

Il s'agit d'une zone d'attente de commit, c'est la passerelle entre le répertoire de travail et le dépôt local.

## Manipulation des fichiers

git add <file>
git add -A
git rm <file>

## Vision du contenu

git ls-files –stage
git status

# Dépôt local

Il s'agit du dépôt présent sur la machine, c'est la version locale du dépôt distant (remote).

## Manipulation des fichiers

git commit

## Vision du contenu

git ls-tree -r HEAD
git log -<int>

# Dépôt distant

Il s'agit du dépôt coté serveur. C'est celui-ci qui est trouvable sur le site Github.

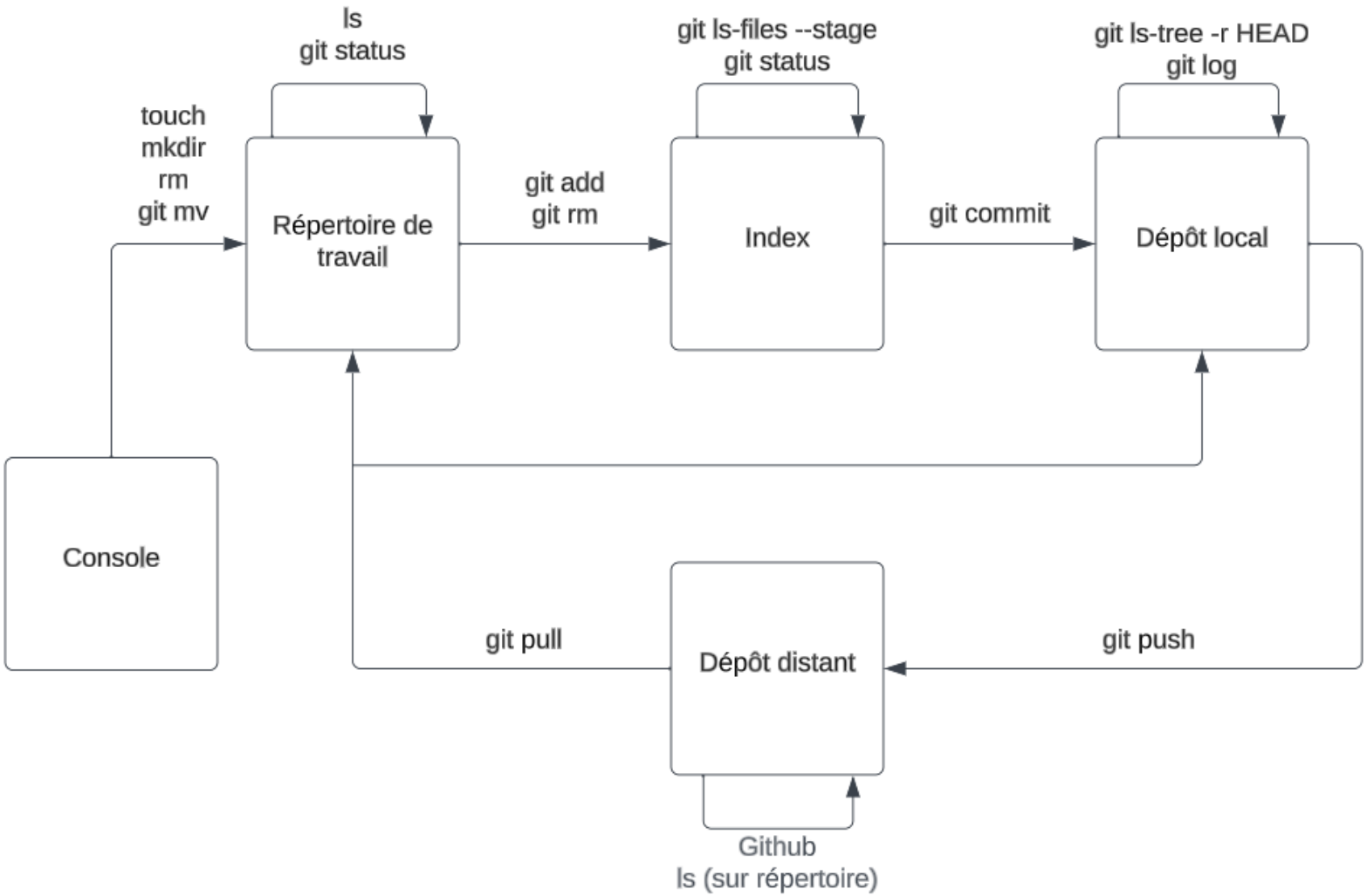
## Manipulation des fichiers

git push
git pull

## Vision du contenu

Sur Github
ls (si synchronisé au dépôt local)

# Résumé


ls
git status
git ls-files --stage
git status
git ls-tree -r HEAD
git log
touch
mkdir
rm
git mv
Répertoire de travail
git add
git rm
Index
git commit
Dépôt local
Console
git pull
Dépôt distant
git push
Github
ls (sur répertoire)

# Convention de communication

- Le readme
- Les commits

# Fichier readme

## Fichier readme

Fichier d'extension .md (MarkDown) contenant des informations importantes ou un guide d'utilisation pour le code fourni dans le dépôt.

## Pourquoi MarkDown

- Meilleure organisation des informations
- Style et police disponible
- Plus léger que du Text

## Ajout du readme

Sur Github, cocher "add a README file"
Sur le répertoire de travail :

```
touch README.md
```

# Les messages

Nous n'aborderons pas dans cette séance le commit décomposé de fichier (ie l'ajout d'une fonctionnalité précise d'un fichier).

## Message de commit

git commit -m "<message>"

Message de maximum 49 caractères (question de lisibilité et d'affichage).

## Exemple

~~git commit -m "Ajout de la fonctionnalité machin dans le fichier bidule"~~

git commit -m "Add F machin in bidule"
ou
git commit -m "+F machin -> bidule"

# Glossaire des commandes

## git add

git add <file> : Ajoute file et ses modifications à la zone d'index.
git add -A : Ajoute tout les fichiers modifiés à la zone d'index. git add .. réalise la même chose.

## git clone

git clone <url> : Créer, en dépôt local, une copie du dépôt distant se associé à l'url donnée.

## git commit

git commit : Créer un commit à partir des fichiers présent dans l'index. Ne pas mettre d'argument ouvrira une fenêtre pour écrire un commentaire.
git commit -m "<message>" : Créer un commit avec comme message la valeur de message.

# Glossaire des commandes

## git config

git config –global user.name/email <value> : Définit le pseudo et l'email associé au dépôt local à ceux donnés en argument et ce pour tous les dépôts existants (et futurs) sur la machine.
git config –global init.defaultBranch <name> : Définit le nom de la branche créée par défaut sur name. (Par défaut master).

## git init

git init : Initialise un dépôt local dans le dossier d'exécution en y créant un fichier .git.

## git log

git log -<int> : Affiche les int derniers commits effectués (déjà push ou non).

# Glossaire des commandes

## git ls-files

git ls-files –stage : Liste les fichiers du répertoire de travail ainsi que la zone d'index où ils se trouvent.

## git ls-tree

git ls-tree -r HEAD : Affiche la liste récursive des fichiers et dossiers dans l'arborescence du dépôt à partir de la branche HEAD. HEAD est la branche de travail active.

## git mv

git mv <old file> <new file> : Renomme ou déplace old file vers new file. old et new file peuvent être des chemins relatifs ou des noms de fichier.

# Glossaire des commandes

## git pull

git pull : Synchronise le dépôt local avec les changements effectués sur le dépôt distant. Inutile si vous travaillez seul sur votre projet.

## git push

git push : Synchronise le dépôt distant avec les changements réalisés sur le dépôt local.
git push -u origin master : Synchronise la branche master du dépôt distant associé à origin en spécifiant le suivi de celui-ci.

## git remote

git remote add origin <url> : Définit l'origine distante du dépôt local au dépôt distant lié à l'url donnée.

# Glossaire des commandes

## git rm

git rm <file> : Prépare la suppression du fichier file du dépôt. Le fichier sera également supprimé du répertoire de travail.

## git status

git status : Informe de l'état général du dépôt local en comparaison au dépôt distant. Indique de même la situation de la zone d'index et de dépôt.

## ls

ls : Liste l'ensemble des fichiers et dossiers présent dans le répertoire d'exécution. Le listing n'est pas récursif.

# Glossaire des commandes

## mkdir

mkdir <dir> : Crée le dossier dir dans le répertoire d'exécution.

## touch

touch <file> : Crée le fichier file dans le répertoire d'exécution.

## rm

rm <file> : Supprime le fichier file si il existe dans le répertoire d'exécution.
rm -f -r <dir> : Force la suppression, de manière récursive, du dossier dir si il existe dans le répertoire d'exécution.