git config --global user.name "MyName"

git config --global user.email "name@domaine.xx"

git config -l // regarder sa configuration

git init // Configurer un nouveau projet et initialiser Git

git status // Connaître le statut de vos fichiers

git add <nom_du_fichier_ou_du_répertoire> // dire quel fichier ou répertoire on suit et dont on conserve les versions

git add . // ajouter tous les fichiers du répertoire

git commit -m "Message qui indique sur quoi porte cette version" // on crée une version

git diff // regarder les différences

git log // avoir un historique des commit

git branch <nom_de_la_branche> // créer une branche

git branch // voir les branches

git checkout <nom_de_la_branche> // se mettre sur la branche

git diff master..<nom_de_la_branche> // voir la différence entre 2 branches

git merge <nom_de_la_branche> // rapatrier une branche vers le tronc

git branch -d <nom_de_la_branche> // supprimer une branche

git remote add <nom_du_remote> <url_projet> // configurer un remote repository.

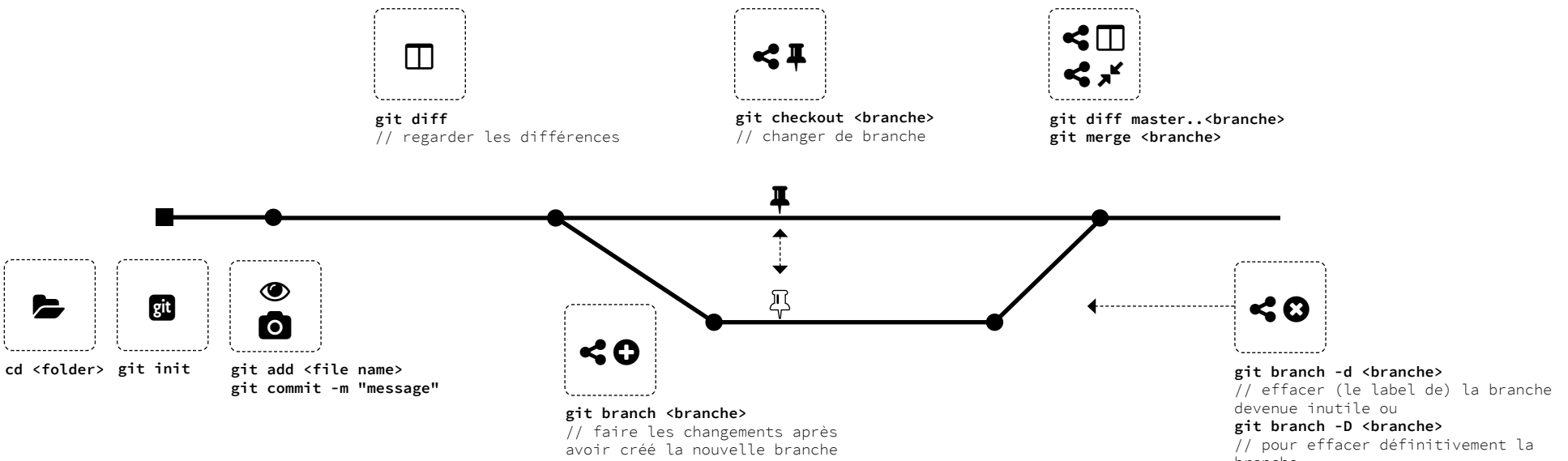
git remote -v // Vérifier qu'un remote repository est disponible pour le projet

git push <nom_du_remote> <nom_de_la_branche> // Envoyer son code vers le remote repository

git pull <nom_du_remote> <nom_de_la_branche> // Récupérer du code du remote repository = fetch + merge

git clone <url_projet> // Récupérer en local un projet existant

git fetch // Récupérer les derniers changements


git diff
// regarder les différences
git checkout <branche>
// changer de branche
git diff master..<branche>
git merge <branche>
cd <folder>
git init
git add <file name>
git commit -m "message"
git branch <branche>
// faire les changements après
avoir créé la nouvelle branche
git branch -d <branche>
// effacer (le label de) la branche
devenue inutile ou
git branch -D <branche>
// pour effacer définitivement la
branche