Guimier Tim

QUESTIONS :

1. détailler l’installation, l’histoire et l’utilité de Git.

Git est issue du noyau Linux, donc libre et de grande envergure. Les modifications étaient transmises sous forme de patchs et d’archives de fichier. En 2002, le projet du noyau Linux commença à utiliser un DVCS propriétaire appelé BitKeeper. En 2005, les relations entre la communauté développant le noyau Linux et la société en charge du développement de BitKeeper furent rompues, et le statut de gratuité de l'outil fut révoqué. Cela poussa la communauté du développement de Linux (et plus particulièrement Linus Torvalds, le créateur de Linux) à développer son propre outil en se basant sur les leçons apprises lors de l'utilisation de BitKeeper. Certains des objectifs du nouveau système étaient les suivants :

* vitesse ;
* conception simple ;
* support pour les développements non linéaires (milliers de branches parallèles) ;
* complètement distribué ;
* capacité à gérer efficacement des projets d'envergure tels que le noyau Linux (vitesse et compacité des données).

Depuis, Git a pu évolué en efficacité et en simplicité d’utilisation.il est très efficace pour de grands projets et il a un incroyable système de branches pour des développements non linéaires

Installation Git

* Télécharger Git
* Sélectionner les composants et laissez les options par défaut. (vérifier les composants supplémentaires désiré)
* Editeur de choix, par défaut utilisé par Git
* dans l’environnement Ajuster votre PATH, il est conseillé de conserver l’option Utiliser Git par défaut dans l’invite de commande
* Laisser les valeurs par défaut sélectionnée sous Utiliser OpenSSH

1. Explication des commandes : init, add, push, pull, diff, config, status branch, checkout, remote, merge, fetch, stash, log.

Commande de base

**git init** : Cela crée un nouveau sous-répertoire nommé .git qui contient tous les fichiers nécessaires au dépôt.

**git add** : Cette commande met à jour l’index en utilisant le contenu actuel trouvé dans l’arbre de travail, pour préparer le contenu de la prochaine validation.

**git commit** : Créer un nouveau commit concernant le contenu actuel de l’index et avec le message de validation décrivant la modification

**Git push** : Met à jour les références distantes avec les objets associés.

**git-pull** - Récupère et intègre un autre référentiel ou une branche locale

**git log** : Montre les log de commit

**git status** : Montre le statut de l’arbre de travail

1. Mettre ce fichier sur Git.

