**2.2 Installer Git**

Vérifiez si Git est installé avec la commande suivante :

git --version

Vous devez avoir des résultats du genre :

git version 2.29

Si ce n'est pas le cas. Alors, exécutez les commandes suivant :

sudo add-apt-repository ppa:git-core/ppa

sudo apt update

sudo apt install git

Vérifiez la version de Git pour s'assurer qu'il s'est bien installer.

**3. Configurer Git et GitHub**

**3.1 Configurer Git**

La configuration consiste à s'identifier localement à Git et lier notre compte Git avec notre compte GitHub. Commençant par vous identifier sur Git.

Exécuter les commandes suivantes en mettant vos informations au lieu de ce qui est écrit dans les double cotes:

git config --global user.name "Votre nom"

git config --global user.email "votre-email@email.com"

GitHub a changé la branche par défaut de master à main , faisons cela sur Git aussi:

git config --global init.defaultBranch main

Donnant un peu de vie à notre code Git:

git config --global color.ui auto

Vérifier si l'identification s'est effectuer correctement :

git config --get user.name

git config --get user.email

**3.2 Créer un compte GitHub**

Accédez au site officiel de GitHub et créer un compte, si vous n'en avez pas un, avec le même e-mail utiliser sur Git.

**3.3 Créer une clé SSH**

Une clé SSH est un identifiant crypté permettant aux serveur comme GitHub à s'identifier à votre machine. Ainsi, vous pouvez télécharger vos répertoire aux serveur sans avoir besoin de taper votre nom d'utilisateur et votre mot de passe à chaque fois.

Pour savoir si vous avez une clé SSH sur votre machine, tapez la commande suivante :

ls ~/.ssh/id\_rsa.pub

* **Questions**

Si vous avez reçu un message du genre :

ls: cannot access '/home/votre-nom-utilisateur/.ssh/id\_rsa.pub': No such file or directory

C'est que vous n'avez pas encore créer un. Donc, vous pouvez le faire en exécutant la commande suivante:

ssh-keygen -C votre-email@email.com

Gardez tous par défaut, juste tapez Entrer.

**3.4 Se lier à GitHub par clé SSH**

Cette liaison consiste à ajouter votre clé SSH à la liste des clé SSH sur GitHub.

Affichez votre clé SSH :

cat ~/.ssh/id\_rsa.pub

Votre clé, normalement, commence par ssh-rsa et fini par votre adresse e-mail.

Maintenant, retournez à GitHub dans la fenêtre de votre navigateur et collez la clé que vous avez copiée dans le champ clé. Ensuite, cliquez sur Ajouter une clé SSH. Vous avez terminé ! Vous avez ajouté votre clé SSH avec succès ! (faites une recherche Google si vous êtes perdu 😄).

**3.5. Tester votre clé**

Testez votre clé en suivant ce [tutoriel](https://help.github.com/en/articles/testing-your-ssh-connection).

**4.2 Pratique**

1. Créez un nouveau répertoire sur votre compte GitHUb et nommez le test\_git.
2. Copiez l'URL de ce répertoire sur votre machine locale on utilisant l'option SSH.
3. En ligne de commande sur votre machine locale, naviguez jusqu'à l'endroit où vous souhaitez stocker ce projet, puis clonez votre répertoire sur GitHub sur votre ordinateur avec git clone suivi de l'URL que vous avez copiée à la dernière étape. La commande complète devrait ressembler à git clone[git@github](mailto:git@github.com).com:nom-utilidsateur/test\_git.git.
4. Déplacez vous sur le répertoire cloné et vérifiez si la connexion s'est bien établie avec le répertoire sur GitHub: git remote -v
5. Décrire le résultat de cette commande?
6. Créez un nouveau fichier dans le dossier git\_test appelé README.md.
7. Tapez la commande git status dans votre terminal et décrivez le résultat de la commande.
8. Tapez la commande git add README.md.
9. Tapez la commande git status. Que remarquez vous? et pourquoi?
10. Tapez la commande git commit -m "Ajouter README.md".
11. Tapez la commande git status.
12. Avec vos propre mots, décrire le workflow que vous venez de pratiquer de l'étape 6 à 11.
13. Tapez git log et décrivez le résultat de la commande, ainsi le fonctionnement de la commande git log.
14. Créez un nouveau fichier hello\_world.txt dans le répertoire test\_git.
15. Vérifiez le statut de Git.
16. Ouvrir le fichier README.md avec votre éditeur de texte et ajouter une petite présentation de vous puis enregistrer les changement et fermer le fichier.
17. Vérifiez le statut de Git.
18. Ajoutez README.md à la staging area avec git add ..
19. Pouvez-vous deviner quel sera le statut de Git maintenant? Quelle est la différence entre git add . et git add README.md.
20. Valider tous les fichiers qui se trouvent dans la staging area et ajoutez un message de validation descriptif git commit -m "votre message".
21. Vérifiez le status de git maintenant.
22. Tapez git push origin main
23. Verifiez le statut de git
24. Jetez un coup d'oeil sur votre répertoire GihHub. Des remarques?