

TD N°1 –Sécurité Informatique & Cybersécurité



Systemes Git et GitHub

Présenté Par :
OCCINE ODNEY.

Professeur :
ISMAEL SAINT AMOOUR

Option:
RESEAUX

Niveau :
2e année

Date : 5/1/2026

1. Objectif du TD

L'objectif de ce travail dirigé est d'apprendre à utiliser Git et GitHub via la ligne de commande. Il consiste à créer un dépôt GitHub public, à le cloner localement, à organiser les dossiers, puis à héberger le rapport de travail au format PDF et les captures d'écran associées.

2. Démarche suivie

Pour réaliser ce TD, les étapes suivantes ont été suivies :

1. Création d'un dépôt GitHub public nommé *TD*
2. Clonage du dépôt sur le bureau à l'aide de la commande `git clone`
3. Création de la structure de dossiers demandée
4. Rédaction du rapport de travail et insertion des captures d'écran
5. Exportation du rapport en PDF et ajout des fichiers dans le dépôt local
6. Héberger tout sur GitHub à l'aide des commandes Git

3. Résultats de l'exécution des commandes

Commande 1 : `git status`

```
PS C:\Users\odney\Desktop\td\systeme> git add .
PS C:\Users\odney\Desktop\td\systeme> git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   image/capture (0).jpg
    new file:   image/capture (0).png
    new file:   image/capture (1).jpg
    new file:   image/capture (2).jpg
    new file:   image/capture (3).jpg
    new file:   presentation/rapport.pdf

PS C:\Users\odney\Desktop\td\systeme>
```

Explication : Cette commande permet de vérifier l'état des fichiers dans le dépôt.

Commande 2 : git add .

```
PS C:\Users\odney\Desktop\td\systeme> git add .  
>>
```

Explication : Cette commande ajoute tous les fichiers modifiés à l'index Git pour préparer le commit.

Commande 3 : git commit -m " rapport1"

```
odney@DESKTOP-K3FQT05 MINGW64 ~/Desktop/td/systeme (main)  
$ git push -u origin main  
Enumerating objects: 11, done.  
Counting objects: 100% (11/11), done.  
Delta compression using up to 4 threads  
Compressing objects: 100% (9/9), done.  
Writing objects: 100% (11/11), 7.15 MiB | 1.08 MiB/s, done.  
Total 11 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
To github.com:odyht/TD.git  
 * [new branch]      main -> main  
branch 'main' set up to track 'origin/main'.
```

Explication : Cette commande enregistre les modifications avec un message décrivant le contenu.

Commande 4 : git push -u origin main

```
PS C:\Users\odney\Desktop\td\systeme> git commit -m "Ajout rapport1 "  
[main (root-commit) ce6a3fc] Ajout rapport1  
6 files changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 systeme/image/capture (0).jpg  
create mode 100644 systeme/image/capture (0).png  
create mode 100644 systeme/image/capture (1).jpg  
create mode 100644 systeme/image/capture (2).jpg  
create mode 100644 systeme/image/capture (3).jpg  
create mode 100644 systeme/presentation/rapport.pdf  
PS C:\Users\odney\Desktop\td\systeme>
```

Explication : Cette commande envoie les commits locaux vers le dépôt distant Github sur la branche principale.

5. Conclusion

Ce TD ma permis de comprendre les bases de Git et Github, l'organisation de dossiers pour un projet, et l'importance de la documentation.

La tâche a été réalisée avec succès. Les difficultés rencontrées avec certaines commandes Git ont été résolues grâce à la pratique et à la vérification régulière de l'état du dépôt avec git status.

6. Tâche réussie, difficultés rencontrées et solutions

Tâche réussie :

La tâche a été réalisée avec succès, Le dépôt Github a été créé, la structure des dossiers a été mise en place correctement, le rapport au format PDF a été rédigé et les captures d'écran ont été hébergées sur GitHub via les commandes Git

Difficultés rencontrées :

Quelques erreurs de chemin en créant les dossiers

Vérification de l'emplacement correct des fichiers PDF et images

Solutions apportées :

Utilisation régulière de git status pour vérifier l'état du dépôt avant chaque commit.

Correction des noms de dossiers pour respecter exactement la consigne.

Organisation préalable des fichiers et vérification avant de faire le commit final