

 	Configurer une connexion SSH avec GitHub sous Windows	Réalisé le :	31/08/2024
		Modifié le :	11/11/2024 10:51

Nombre de
pages : 17

Configurer une connexion SSH avec GitHub sous Windows

Comment le faire en six étapes

1. LE PROTOCOLE SSH	2
1.1. POURQUOI UTILISER SSH ?	2
1.2. DEFINITION DE WIKIPEDIA	2
1.3. LOCALISATION DE SSH DANS LES MODELES OSI ET TCP/IP	2
1.3.1. Principaux éléments du protocole SSH	4
2. MODE OPERATOIRE EN SIX ETAPES	4
2.1. ÉTAPE 1/6 : VERIFIER LA PRESENCE DE CLES SSH	4
2.2. ÉTAPE 2/6 : GENERER UNE NOUVELLE CLE SSH	4
2.2.1. Génération de la clé SSH	4
2.2.2. Passphrase	5
2.3. ÉTAPE 3/6 : AJOUTER VOTRE CLE SSH A L'AGENT SSH (FACULTATIF : SI PASSPHRASE)	7
2.4. ÉTAPE 4/6 : AJOUTER LA CLE SSH PUBLIQUE A VOTRE COMPTE GITHUB	7
2.4.1. Compte GitHub / Settings	8
2.4.2. SSH and GPG keys	9
2.4.3. New SSH key	9
2.4.4. Add new SSH Key / Title	10
2.4.5. Le cas de plusieurs clés SSH	11
2.5. ÉTAPE 5/5 : TESTER LA CONNEXION SSH	11
2.5.1. Test de la liaison SSH et ajout du serveur	11
2.5.1.1. Confirmation avec un ping	12
2.5.2. Vérifier la présence du fichier known_hosts	12
2.5.3. Clonage d'un repository par SSH	12
2.5.3.1. Sélectionner le projet distant sur GitHub et cliquer sur le bouton « Code »	12
2.5.3.2. Cliquer sur l'onglet « SSH »	13
2.5.3.3. Cliquer sur le bouton pour copier l'url (en SSH) dans le presse-papiers	13
2.5.3.4. Cloner le projet en local avec la commande git clone + url	13
2.6. ÉTAPE 6/6 : CONFIGURER GIT POUR UTILISER SSH PAR DEFAULT	14
2.6.1. Example	14
2.6.2. Vérification sur le serveur GitHub	15
3. LINKS	15
3.1. IA	15
3.1.1. ChatGPT Open AI	15
3.1.2. Claude.ai	15
3.1.3. Gemini Google	15
3.1.4. Microsoft Copilot	15
3.1.5. Mistral.ai	15
3.1.6. You.com	15
3.2. TEXTE	15
3.2.1. Generating a new SSH key and adding it to the ssh-agent (GitHub)	15
3.2.2. Generating an SSH Key Pair (Oracle Help Center)	15
3.2.3. Utilisation d'une SSH key avec GitHub (IONOS)	15
3.3. VIDEO	15

 	Configurer une connexion SSH avec GitHub sous Windows	Réalisé le :	31/08/2024
		Modifié le :	11/11/2024 10:51

3.3.1. Configurer une connexion SSH avec Github -----15

4. SUPPORT-----16

4.1. VERIFIER QUE L'ON DISPOSE D'UNE CLE SSH EN COURS D'UTILISATION -----16

4.2. VERIFIER QUE LA CLE EST EN COURS D'UTILISATION -----16

1. Le protocole SSH

1.1. Pourquoi utiliser SSH ?

Configurer une connexion SSH avec GitHub est une excellente idée pour sécuriser et simplifier les interactions avec les dépôts Git, telles que les clonages de dépôt, les push et les pull, en utilisant des clés SSH.

De manière plus générale, SSH permet aux utilisateurs et administrateurs de se connecter de manière sécurisée à des serveurs ou d'autres ordinateurs, même sur des réseaux non sécurisés.

SSH chiffre les connexions, garantit l'authentification des utilisateurs, et permet d'exécuter des commandes, transférer des fichiers, et gérer des systèmes de manière sécurisée.

1.2. Définition de Wikipédia

Secure Shell (SSH) est un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite, tous les segments TCP sont authentifiés et chiffrés. Il devient donc impossible d'utiliser un analyseur de paquets (sniffer) pour voir ce que fait l'utilisateur¹.

- Date de création : 1995
- Port : TCP/22

1.3. Localisation de SSH dans les modèles OSI et TCP/IP

C'est dans la couche application que se situe SSH.

¹ https://fr.wikipedia.org/wiki/Secure_Shell

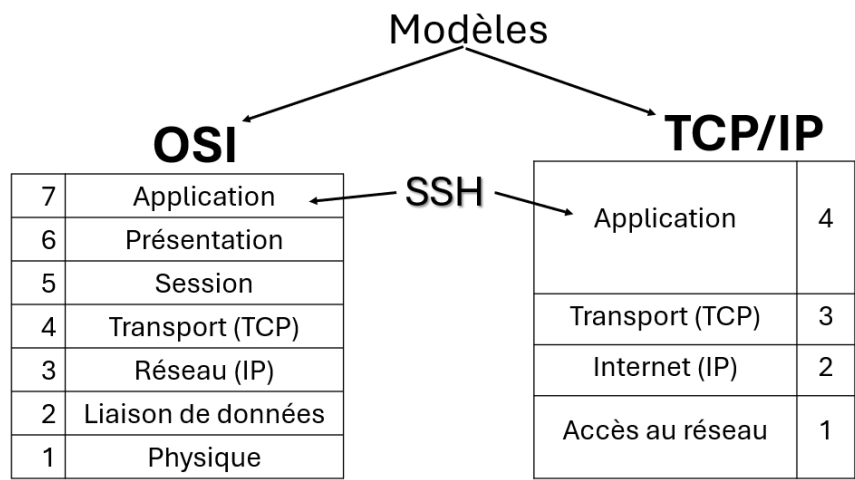
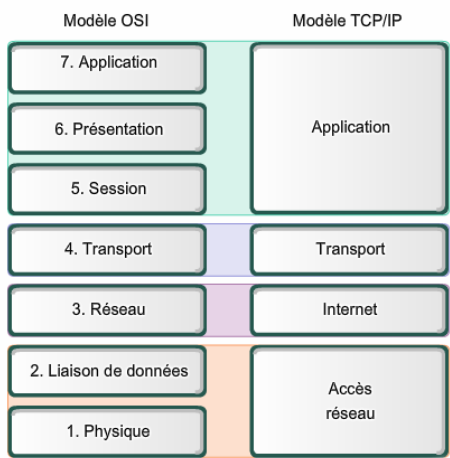


Figure 1: localisation du protocole SSH dans les modèles OSI et TCP/IP



Les applications fonctionnent généralement au-dessus de TCP ou d'UDP, et sont souvent associées à un port bien connu. Exemples :

- HTTP port TCP 80 ;
- SSH port TCP 22 ;
- DNS port UDP 53 (TCP 53 pour les transferts de zones et les requêtes supérieures à 512 octets) ;
- RIP port UDP 520 ;
- FTP port TCP 21 ;

	<h1>Configurer une connexion SSH avec GitHub sous Windows</h1>	Réalisé le :	31/08/2024
		Modifié le :	11/11/2024 10:51

Nombre de pages : 17

1.3.1. Principaux éléments du protocole SSH

../..

2. Mode opératoire en six étapes

Voici un mode opératoire en six étapes pour configurer une connexion SSH avec GitHub.

Étape 1 : Vérifier les clés SSH existantes

Étape 2 : Générer une nouvelle clé SSH

Étape 3 : Ajouter votre clé SSH à l'agent SSH

Étape 4 : Ajouter la clé SSH publique à votre compte GitHub

Étape 5 : Tester la connexion SSH

Étape 6 : Configurer Git pour utiliser SSH par défaut

2.1. Étape 1/6 : Vérifier la présence de clés SSH

Avant de créer une nouvelle clé SSH, il convient, au préalable, de vérifier si des clés SSH sont déjà configurées sur la machine. Pour cela il faut exécuter la commande suivante dans un terminal :

```
ls -al ~/.ssh
```

Cela affichera les fichiers dans le répertoire `.ssh`. Si des fichiers comme `id_rsa.pub` ou `id_ed25519.pub`, sont présents , cela signifie que des clés SSH sont déjà présentes sur la machine.

2.2. Étape 2/6 : Générer une nouvelle clé SSH

Génération de la clé SSH et passphrase.

2.2.1. Génération de la clé SSH

S'il n'y a pas de clé SSH ou si l'on souhaite en créer une nouvelle, il faut utiliser la commande suivante:

Note: If you are using a legacy system that doesn't support the Ed25519 algorithm, use:

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

2

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

en l'adaptant comme ceci :

```
ssh-keygen -t rsa -b 4096 -C "ikradesk@gmail.com"
```

² <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

  Nombre de pages : 17	Configurer une connexion SSH avec GitHub sous Windows	Réalisé le :	31/08/2024
		Modifié le :	11/11/2024 10:51

On copie et on colle cette commande dans n'importe quel répertoire.

```

MINGW64:/c/Users/Djamel/Desktop
Djamel@zBooki7 MINGW64 ~/Desktop
$ ssh-keygen -t rsa -b 4096 -C "ikradesk@gmail.com"

```

```

MINGW64:/c/Users/Djamel/Desktop
Djamel@zBooki7 MINGW64 ~/Desktop
$ ssh-keygen -t rsa -b 4096 -C "ikradesk@gmail.com"|
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Djamel/.ssh/id_rsa):

```

2.2.2. Passphrase

Ici on ne va pas taper une passphrase pour l'ajouter à la clé privée, car sinon à chaque connexion avec le compte GitHub il faudra retaper cette passphrase !

On laissera donc ce champ vide.

```

MINGW64:/c/Users/Djamel/Desktop
Djamel@zBooki7 MINGW64 ~/Desktop
$ ssh-keygen -t rsa -b 4096 -C "ikradesk@gmail.com"|
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Djamel/.ssh/id_rsa):
Created directory '/c/Users/Djamel/.ssh'.
Enter passphrase (empty for no passphrase):

```

```

MINGW64:/c/Users/Djamel/Desktop
Djamel@zBooki7 MINGW64 ~/Desktop
$ ssh-keygen -t rsa -b 4096 -C "ikradesk@gmail.com"|
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Djamel/.ssh/id_rsa):
Created directory '/c/Users/Djamel/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:

```

 	<h1>Configurer une connexion SSH avec GitHub sous Windows</h1>	Réalisé le :	31/08/2024
		Modifié le :	11/11/2024 10:51

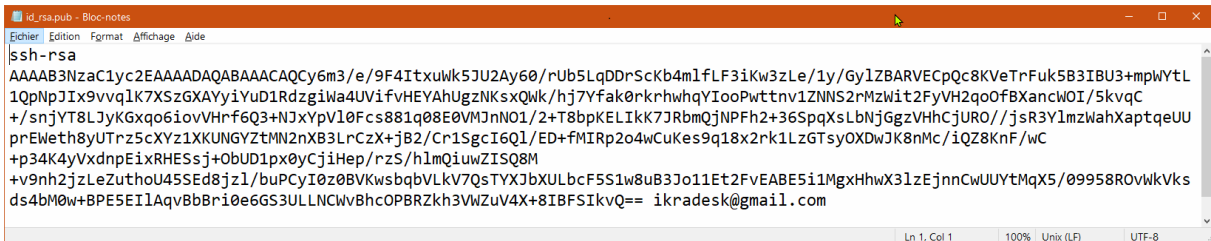
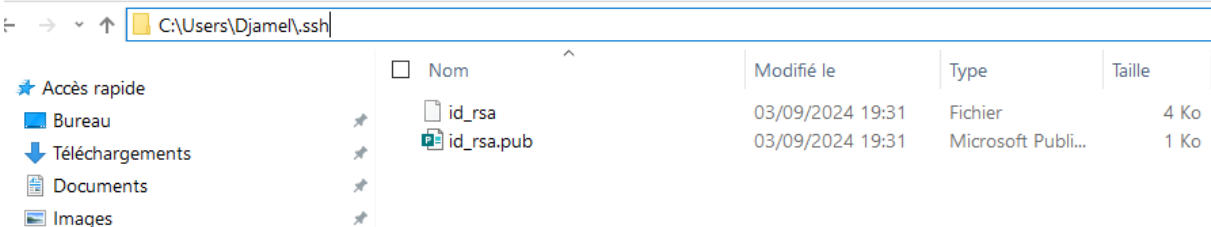
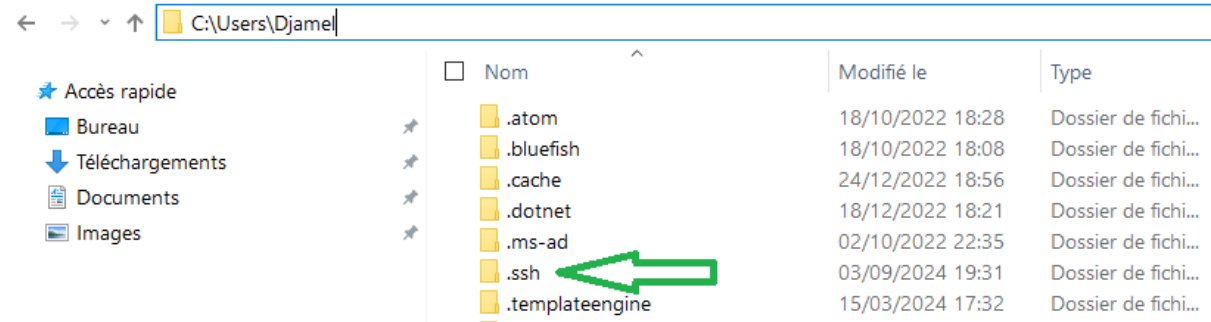
Nombre de pages : 17

```

MINGW64~/c/Users/Djamel/Desktop
Djamel@zBooki7 MINGW64 ~/Desktop
$ ssh-keygen -t rsa -b 4096 -C "ikradesk@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Djamel/.ssh/id_rsa):
Created directory '/c/Users/Djamel/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Djamel/.ssh/id_rsa
Your public key has been saved in /c/Users/Djamel/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:qaGrRCgcdXriiKJv9ypcbm8pTQS9yq8U6Ozl9PpZBLA ikradesk@gmail.com
The key's randomart image is:
+---[RSA 4096]-----+
|  ..O |
|  .+O. |
|  .oEo.. |
|o.+..O O. . |
|=ooo.O. S |
|++  +O.+ |
|..+O=+... |
|++==O=O |
|..+*=X* |
+---[SHA256]-----+

Djamel@zBooki7 MINGW64 ~/Desktop
$

```



 	<h1>Configurer une connexion SSH avec GitHub sous Windows</h1>	Réalisé le :	31/08/2024
		Modifié le :	11/11/2024 10:51

Nombre de pages : 17

```

MINGW64~/Users/Djamel/.ssh
Djamel@zBook17 MINGW64 ~//.ssh
$ ls
id_rsa id_rsa.pub
Djamel@zBook17 MINGW64 ~//.ssh
$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACy6m3/e/9F4ItxuWk5JU2Ay60/rub5LqDpScKb4m1fLF3ikw3zLe/1y/Gy1ZBARVECPqc8KVeTrFuk5
mpwYtL1opNpJIX9vvq1K7XSzGXAYyiyuD1Rdzgiwa4UVifvHEYAhUgzNksxQWk/hj7Yfak0rkRhwhqYIooPwttnv1ZNNS2rMzwit2FyVh2qoofBXancwoI
/snjYT8LJyKGXqo6ioVvHrf6Q3+NjXypVl0Fcs881q08E0VMJnn01/2+T8bpKELIkK7JRbmQjNPFh2+36SpqXSLbnJggzVHhCjURO//jsR3Y1mzwahXapt
weth8yUTrz5cXyz1XKUNGYZtMN2nXB3LrcZx+jB2/Cr1SgcI6Q1/ED+fMIRp2o4wCukes9q18x2rk1LZGTSyOXDwJK8nMc/iQZ8KnF/wc+p34K4yVxdnPE
j+ObUD1px0YcjHep/rzS/h1mqiuwZISQ8M+v9nh2jzLeZuthou45SEd8jz1/buPCyIOz0BVKwsbqbVLkv7QsTYXJbXULbcF5S1w8uB3Jo11Et2FvEABE5
wx3lZeJnnCWUUYtmQx5/09958ROvwkVksds4bm0w+BPE5E1lAqvBbBri0e6GS3ULLNCwVBhCOPBRZkh3VWZuV4x+8IBFSikvQ== ikradesk@gmail.com
Djamel@zBook17 MINGW64 ~//.ssh
$

```

C'est le contenu de cette clé : **id_rsa.pub** que l'on va copier au bon endroit du compte GitHub.

2.3. Étape 3/6 : Ajouter votre clé SSH à l'agent SSH (Facultatif : si passphrase)

Agent SSH: L'agent SSH peut vous éviter d'entrer votre phrase de passe à chaque fois.

Pour permettre à l'OS d'utiliser la clé SSH sans avoir à entrer le mot de passe à chaque fois, il faut ajouter la clé SSH à l'agent SSH.

Démarrer l'agent SSH et taper la commande :

```
eval "$(ssh-agent -s)"
```

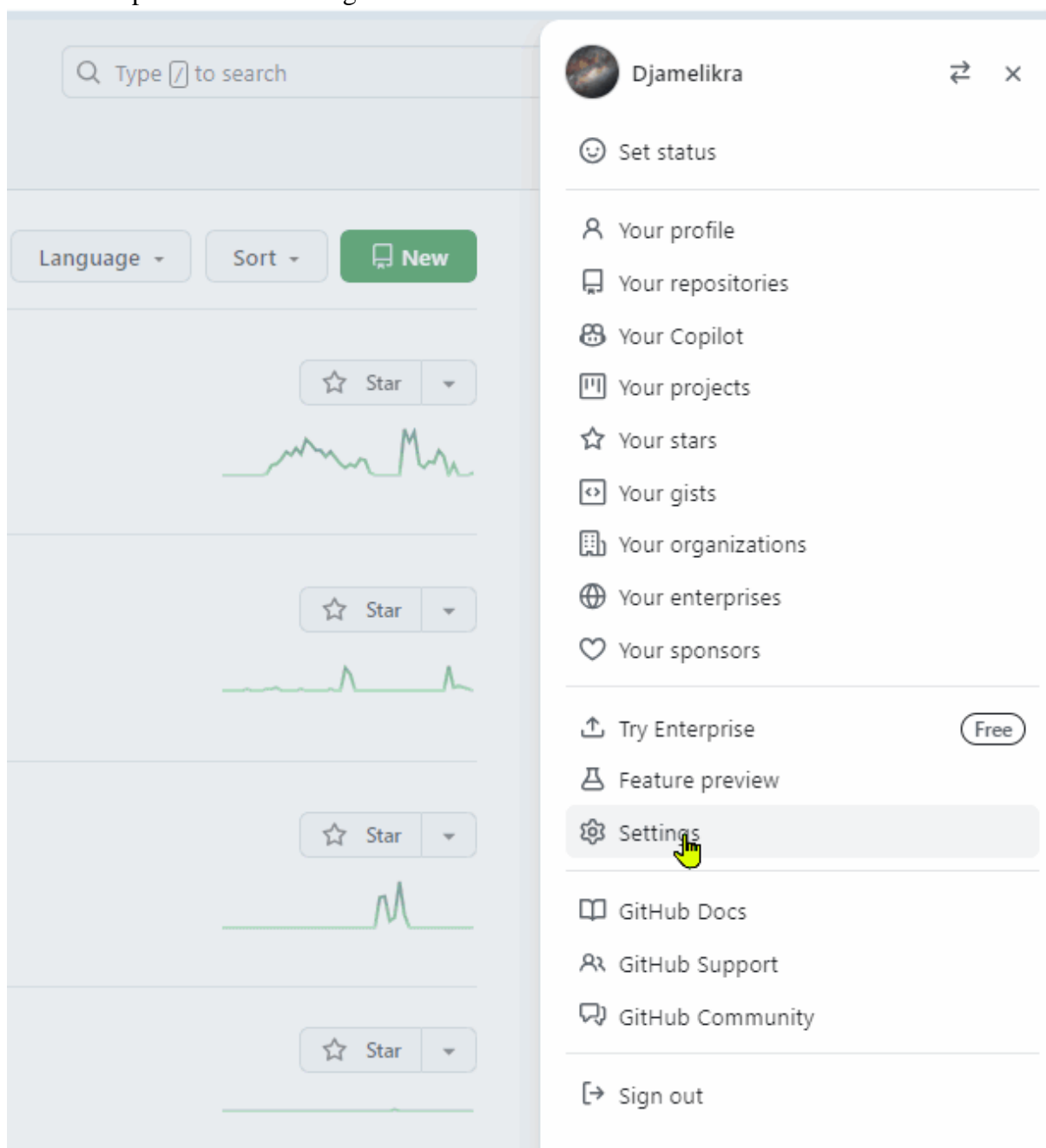
2.4. Étape 4/6 : Ajouter la clé SSH publique à votre compte GitHub

Aller dans Compte GitHub / Settings

 	Configurer une connexion SSH avec GitHub sous Windows	Réalisé le :	31/08/2024
		Modifié le :	11/11/2024 10:51

Nombre de
pages : 17

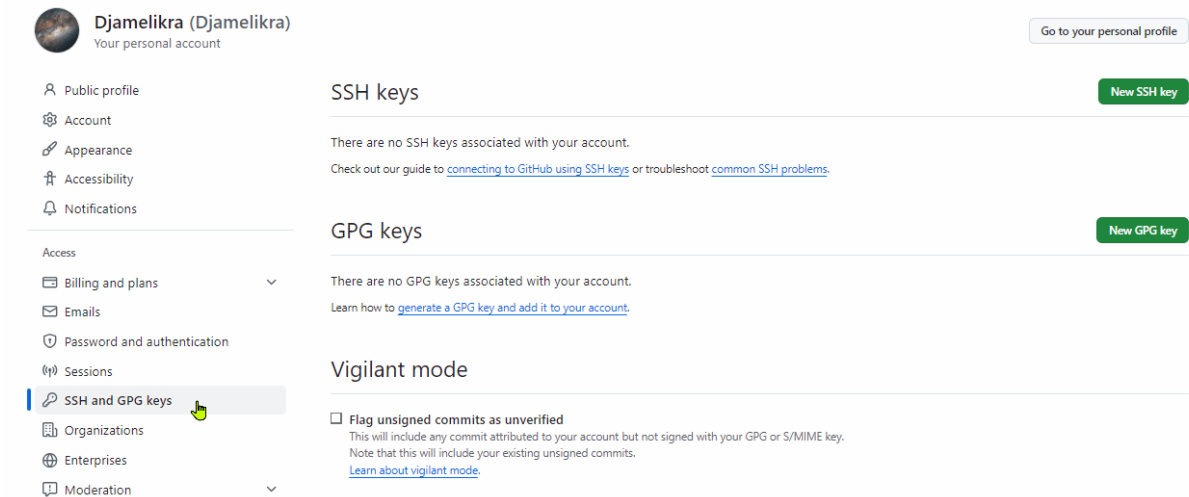
2.4.1. Compte GitHub / Settings



Ensuite aller dans SSH and GPG keys

 Nombre de pages : 17	<h2>Configurer une connexion SSH avec GitHub sous Windows</h2>	Réalisé le :	31/08/2024
		Modifié le :	11/11/2024 10:51

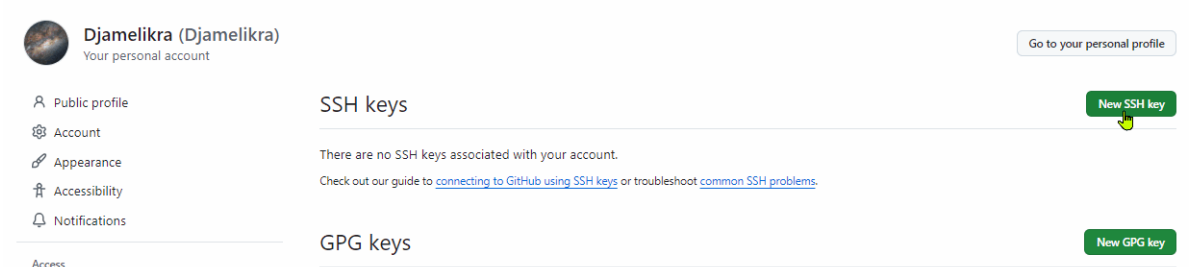
2.4.2. SSH and GPG keys



The screenshot shows the GitHub profile page for 'Djamelikra (Djamelikra)'. The left sidebar contains navigation links: Public profile, Account, Appearance, Accessibility, Notifications, Access, Billing and plans, Emails, Password and authentication, Sessions, SSH and GPG keys (highlighted with a cursor), Organizations, Enterprises, and Moderation. The main content area has three sections: 'SSH keys' with a 'New SSH key' button, 'GPG keys' with a 'New GPG key' button, and 'Vigilant mode' with a checkbox for 'Flag unsigned commits as unverified'.

Ensuite cliquer sur « New SSH key »

2.4.3. New SSH key




This screenshot is identical to the previous one, but with a yellow cursor pointing directly at the 'New SSH key' button in the SSH keys section.

Ensuite aller dans la section : « Add new SSH Key », champ « title » **donner un nom** à la clé SSH et on colle le contenu de la clé publique : **id_rsa.pub** dans le champ **key** :

 Nombre de pages : 17	<h2>Configurer une connexion SSH avec GitHub sous Windows</h2>	Réalisé le :	31/08/2024
		Modifié le :	11/11/2024 10:51

2.4.4. Add new SSH Key / Title


Djamelikra (Djamelikra)
 Your personal account

[Go to your personal profile](#)

[Public profile](#)
[Account](#)
[Appearance](#)
[Accessibility](#)
[Notifications](#)

[Access](#)
[Billing and plans](#)
[Emails](#)
[Password and authentication](#)
[Sessions](#)
[SSH and GPG keys](#)
[Organizations](#)
[Enterprises](#)
[Moderation](#)
[Code, planning, and automation](#)
[Repositories](#)

Add new SSH Key

Title

Key type


Authentication Key

Key


```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCAQCym3/e/9F4ltxuWk5JU2Ay60/rUb5LqDDrScKb4mflF3ikw3zLe/1y/GylZBARVECPqC8KVeTrFuk5B3IBU
3+mpWYtL1QpNpJlx9vvqIK7XszGXAYiYuD1RdZgiWa4UVfvHEyAhUgzNKsxQWk/hj7Yfak0rkrwhqYlaoPwttnv1ZNNs2rMzWit2FyVH2goOfbXan
cWOI/5kvqC+/snjVT8LJyKGxqo6ioVHrf6Q3+NjxYpVlOfcs881q08E0VMJnNO1/2+T8bpKElUk7JRbmQjNPFh2+36SpqXsLbNjGgzVHhCJURO//jsR
3YlmzWahXaptqeUUpREWeth8yUTrz5cXyZ1XKUNGYZtMN2nXB3LrCzX+jB2/Cr1Sgcl6QL/ED+fMIrP2o4wCuKes9q18x2rk1LzGTsyOXDwJK8nMc/iQ
Z8KnF/wC+p34K4yVxdnpEixRHESj+ObUD1px0yCjIHep/tz5/hlmQiuwZISQ8M+v9nh2jzLeZuthoU45SEd8jzl/buPCyl0z0BVKwsbqbVLKv7QsTVXJb
XULbcF5S1w8uB3Jo11Et2FvEABESi1MgxHhwX3LzEjnnCwUUYTMqX5/09958ROvWkVksd4bMow+BPESeIIAqVbBri0e6GS3ULLNCVwBhcOPBRZk
h3VWZuV4X+8IBFSikvQ== ikradesk@gmail.com]
```

Add SSH key

Une fois validé il faut rentrer le mot de passe du compte GitHub :



Confirm access


 Signed in as @Djamelikra

Password

[Forgot password?](#)

Confirm

Tip: You are entering [sudo mode](#). After you've performed a sudo-protected action, you'll only be asked to re-authenticate again after a few hours of inactivity.

Et on obtient un message de succès

 	<h1>Configurer une connexion SSH avec GitHub sous Windows</h1>	Réalisé le :	31/08/2024
		Modifié le :	11/11/2024 10:51

Nombre de pages : 17

Settings

You have successfully added the key 'zBooki7Win32Go'.

Djamelikra (Djamelikra)

Your personal account

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations

Enterprises

Moderation

Code, planning, and automation

SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

zBooki7Win32Go

SHA256:qaGrRCgc0Xr-i1K3v9ypcbm8pTQ59yq8U60z19PpZ8LA

Added on Sep 4, 2024

Never used — Read/write

Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

GPG keys

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

Vigilant mode

Maintenant le compte est bien configuré avec une connexion SSH.

2.4.5. Le cas de plusieurs clés SSH

Plusieurs clés SSH: on peut ajouter plusieurs clés SSH à un compte GitHub si l’on dispose de plusieurs ordinateurs ou environnements.

2.5. Étape 5/5 : Tester la connexion SSH

Pour s’assurer que tout fonctionne correctement, testez la connexion SSH avec GitHub avec la commande :

```
ssh -T git@github.com
```

2.5.1. Test de la liaison SSH et ajout du serveur

En tapant cette commande on obtient :

```
Djamel@zBooki7 MINGW64 ~/Desktop
$ ssh -T git@github.com
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCoQU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Hi Djamelikra! You've successfully authenticated, but GitHub does not provide shell access.

Djamel@zBooki7 MINGW64 ~/Desktop
$
```

Comme c’est la première fois que l’on se connecte en SSH à GitHub à partir de cet ordinateur, on est également invité à ajouter le serveur aux hôtes connus (« Known Hosts ») :

©Djamel CHABANE 2024 Configurer une connexion SSH avec Github sous Windows.docx

11 / 17

 	<h1>Configurer une connexion SSH avec GitHub sous Windows</h1>	Réalisé le :	31/08/2024
		Modifié le :	11/11/2024 10:51

Nombre de pages : 17

2.5.1.1. Confirmation avec un ping

```
Administrateur : Windows PowerShell
PS C:\Windows\system32> ping github.com

Envoi d'une requête 'ping' sur github.com [140.82.121.4] avec 32 octets de données :
Réponse de 140.82.121.4 : octets=32 temps=12 ms TTL=52
Réponse de 140.82.121.4 : octets=32 temps=12 ms TTL=52
Réponse de 140.82.121.4 : octets=32 temps=12 ms TTL=52
Réponse de 140.82.121.4 : octets=32 temps=11 ms TTL=52

Statistiques Ping pour 140.82.121.4:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 11ms, Maximum = 12ms, Moyenne = 11ms
PS C:\Windows\system32>
```

2.5.2. Vérifier la présence du fichier known_hosts

Comme tout s’est passé correctement, il y a la présence du fichier *known_hosts* dans le répertoire .ssh de l'utilisateur :

```
MINGW64:/c/Users/Djamel/.ssh
Djamel@zBooki7 MINGW64 ~/.ssh
$ ls
id_rsa id_rsa.pub known_hosts known_hosts.old
```

C:\Users\Djamel\.ssh				
	Nom	Modifié le	Type	Taille
Accès rapide				
Bureau	id_rsa	03/09/2024 19:31	Fichier	4 Ko
Téléchargements	id_rsa.pub	03/09/2024 19:31	Microsoft Publi...	1 Ko
Documents	known_hosts	05/09/2024 18:09	Fichier	1 Ko
Images	known_hosts.old	05/09/2024 18:09	Fichier OLD	1 Ko

2.5.3. Clonage d’un repository par SSH

Cela nécessite 4 étapes.

2.5.3.1. Sélectionner le projet distant sur GitHub et cliquer sur le bouton « Code »

AlgorithmicPrivate

Unwatch1Fork0Star0

main1 Branch0 Tags

Go to fileAdd fileCode

Djamelikra ssh + doc 1d13873f · 27 minutes ago557 Commits

AlgoVba

vba confirm 1last month

TSP

distance ok5 months ago

Access VBA Expert.docx

vba list 1last month

Catalogue Formation Audit VBA Access.docx

array 12 months ago

Configurer une connexion SSH avec Github N...

ssh + doc 127 minutes ago

About

Algorithmic in many languages

Readme

Activity

0 stars

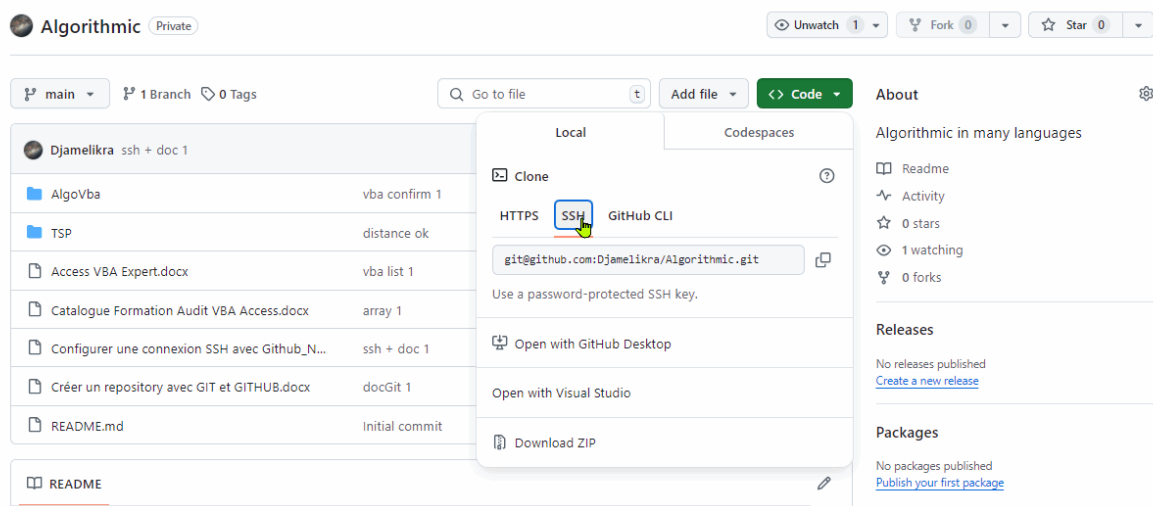
1 watching

0 forks

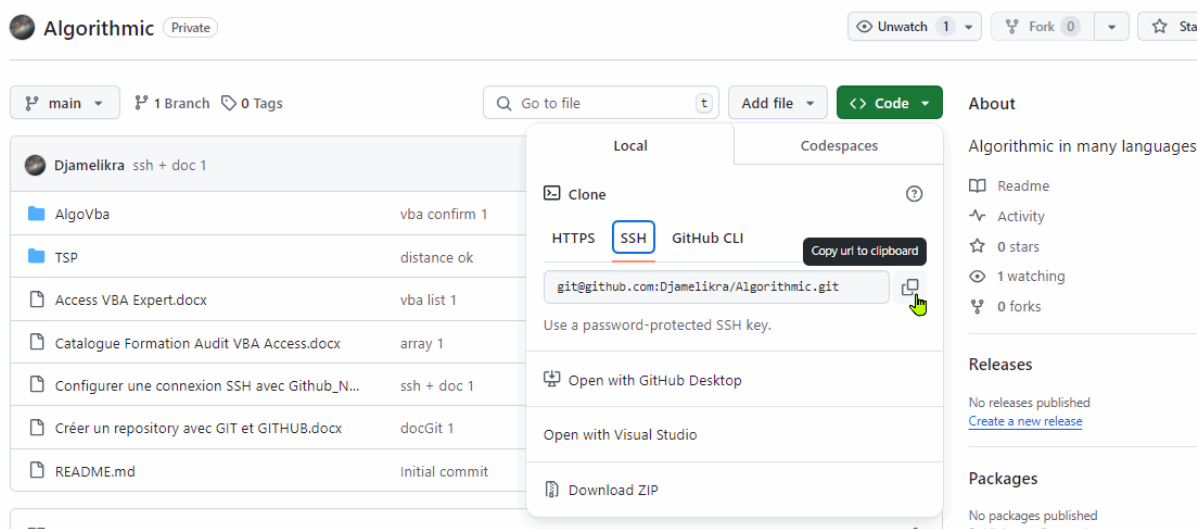
Releases

 Nombre de pages : 17	<h1>Configurer une connexion SSH avec GitHub sous Windows</h1>	Réalisé le :	31/08/2024
		Modifié le :	11/11/2024 10:51

2.5.3.2. Cliquer sur l'onglet « SSH »



2.5.3.3. Cliquer sur le bouton pour copier l'url (en SSH) dans le presse-papiers



2.5.3.4. Cloner le projet en local avec la commande git clone + url

Clonage d'un nouveau dépôt, en utilisant l'URL SSH de la forme :

```
git clone git@github.com:username/repo.git
```

```

MINGW64/c/Users/Djamel/Desktop
Djamel@zBooki7 MINGW64 ~/Desktop
$ git clone git@github.com:Djamelikra/Algorithmic.git
Cloning into 'Algorithmic'...
remote: Enumerating objects: 2699, done.
remote: Counting objects: 100% (647/647), done.
remote: Compressing objects: 100% (363/363), done.
remote: Total 2699 (delta 294), reused 615 (delta 262), pack-reused 2052 (from 1)
Receiving objects: 100% (2699/2699), 138.84 MiB | 17.48 MiB/s, done.
Resolving deltas: 100% (1276/1276), done.
Djamel@zBooki7 MINGW64 ~/Desktop
$

```

 	<h2>Configurer une connexion SSH avec GitHub sous Windows</h2>	Réalisé le :	31/08/2024
		Modifié le :	11/11/2024 10:51

Nombre de pages : 17

2.6. Étape 6/6 : Configurer Git pour utiliser SSH par défaut

Pour configurer Git afin qu'il utilise SSH par défaut au lieu de HTTPS, il faut, **à chaque fois**, modifier l'URL du dépôt concerné:

```
git remote set-url origin git@github.com:username/repo.git
```

2.6.1. Exemple

Voyons cela avec un exemple concret :

```

MINGW64:/g/GitHub/Algorithmic
Djamel@zBooki7 MINGW64 /g/GitHub/Algorithmic (main)
$ git remote set-url origin git@github.com:Djamelikra/Algorithmic.git

Djamel@zBooki7 MINGW64 /g/GitHub/Algorithmic (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

Djamel@zBooki7 MINGW64 /g/GitHub/Algorithmic (main)
$ touch sampleTexte.txt

Djamel@zBooki7 MINGW64 /g/GitHub/Algorithmic (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    sampleTexte.txt

nothing added to commit but untracked files present (use "git add" to track)

Djamel@zBooki7 MINGW64 /g/GitHub/Algorithmic (main)
$ git add sampleTexte.txt

Djamel@zBooki7 MINGW64 /g/GitHub/Algorithmic (main)
$ git commit -m "sample txt 1"
[main b6d2108] sample txt 1
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 sampleTexte.txt

Djamel@zBooki7 MINGW64 /g/GitHub/Algorithmic (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 279 bytes | 279.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:Djamelikra/Algorithmic.git
   f52ae44..b6d2108  main -> main
branch 'main' set up to track 'origin/main'.

Djamel@zBooki7 MINGW64 /g/GitHub/Algorithmic (main)
$ |

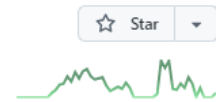
```

	Configurer une connexion SSH avec GitHub sous Windows	Réalisé le :	31/08/2024
		Modifié le :	11/11/2024 10:51

Nombre de pages : 17

2.6.2. Vérification sur le serveur GitHub

Algorithmic Private
 Algorithmic in many languages
 ● VBA Updated now



3. Links

3.1. IA

Le prompt utilisé :

Comment configurer une connexion SSH avec GitHub ?

3.1.1. ChatGPT Open AI

<https://chatgpt.com/share/d24a891e-5b33-4f47-bca1-ed26613cc76d>

3.1.2. Claude.ai

<https://claude.ai/chat/2bb8aba9-c2e8-4c3f-a721-db85b3b52a22>

3.1.3. Gemini Google

<https://g.co/gemini/share/8474ad680d2d>

3.1.4. Microsoft Copilot

<https://sl.bing.net/hIKI1Y6NTU>

3.1.5. Mistral.ai

<https://chat.mistral.ai/chat/7706f876-0c9a-4c1f-9e3a-6733474e8cdc>

3.1.6. You.com

https://you.com/search?q=comment+Configurer+une+connexion+SSH+avec+GitHub&cid=c1_fe5962ad-578b-44fb-8480-55970819be99&tbm=youchat

3.2. Texte

3.2.1. Generating a new SSH key and adding it to the ssh-agent (GitHub)

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

3.2.2. Generating an SSH Key Pair (Oracle Help Center)

<https://docs.oracle.com/cloud-machine/latest/stcompute/ELUSE/GUID-27253116-1FFF-4E1C-A7A7-BB2A03B5BAA5.htm#ELUSE-GUID-27253116-1FFF-4E1C-A7A7-BB2A03B5BAA5>

3.2.3. Utilisation d'une SSH key avec GitHub (IONOS)

<https://www.ionos.fr/digitalguide/sites-internet/developpement-web/ssh-key-avec-github/>

3.3. Vidéo

3.3.1. Configurer une connexion SSH avec Github

<https://youtu.be/sjcEruaUOC0?list=PLdKVEanRftb-H1p8ar5gt7EBPixFIVt-y>

 	<h1>Configurer une connexion SSH avec GitHub sous Windows</h1>	Réalisé le :	31/08/2024
		Modifié le :	11/11/2024 10:51

Nombre de pages : 17

4. Support

4.1. Vérifier que l'on dispose d'une clé SSH en cours d'utilisation

```
$ eval $(ssh-agent -s)
```

```
MINGW64:/c/Users/Djamel/Desktop
Djamel@zBooki7 MINGW64 ~/Desktop
$ eval $(ssh-agent -s)
Agent pid 1021
```

4.2. Vérifier que la clé est en cours d'utilisation

```
ssh -vT git@github.com
```

```
MINGW64:/c/Users/Djamel/Desktop
Djamel@zBooki7 MINGW64 ~/Desktop
$ ssh -vT git@github.com
OpenSSH_9.0p1, OpenSSL 1.1.1q  5 Jul 2022
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: Connecting to github.com [140.82.121.3] port 22.
debug1: Connection established.
debug1: identity file /c/Users/Djamel/.ssh/id_rsa type 0
debug1: identity file /c/Users/Djamel/.ssh/id_rsa-cert type -1
debug1: identity file /c/Users/Djamel/.ssh/id_ecdsa type -1
debug1: identity file /c/Users/Djamel/.ssh/id_ecdsa-cert type -1
debug1: identity file /c/Users/Djamel/.ssh/id_ecdsa.sk type -1
debug1: Next authentication method: publickey
debug1: Offering public key: /c/Users/Djamel/.ssh/id_rsa RSA SHA256:qaGrRCgcdXriiKJv9ypcbm8pTQS9yq8U6Oz19PpZBLA
debug1: Server accepts key: /c/Users/Djamel/.ssh/id_rsa RSA SHA256:qaGrRCgcdXriiKJv9ypcbm8pTQS9yq8U6Oz19PpZBLA
Authenticated to github.com ([140.82.121.3]:22) using "publickey".
debug1: channel 0: new [client-session]
debug1: Entering interactive session.
debug1: pledge: filesystem
debug1: client_input_global_request: rtype hostkeys-00@openssh.com want_reply 0
debug1: client_input_hostkeys: searching /c/Users/Djamel/.ssh/known_hosts for github.com / (none)
debug1: client_input_hostkeys: searching /c/Users/Djamel/.ssh/known_hosts2 for github.com / (none)
debug1: client_input_hostkeys: hostkeys file /c/Users/Djamel/.ssh/known_hosts2 does not exist
debug1: client_input_hostkeys: no new or deprecated keys from server
debug1: client_input_channel_req: channel 0 rtype exit-status reply 0
Hi Djamelikra! You've successfully authenticated, but GitHub does not provide shell access.
debug1: channel 0: free: client-session, nchannels 1
Transferred: sent 3512, received 3096 bytes, in 0.2 seconds
Bytes per second: sent 16245.0, received 14320.8
debug1: Exit status 1
Djamel@zBooki7 MINGW64 ~/Desktop
$
```

La sortie qui s'affiche montre bien que la connexion à GitHub fonctionne très bien 🍀 avec la paire de clés SSH, définies prudemment.

	Configurer une connexion SSH avec GitHub sous Windows	Réalisé le :	31/08/2024
Nombre de pages : 17		Modifié le :	11/11/2024 10:51