

 	<b>Configurer une connexion SSH avec GitHub sous Windows</b>	Réalisé le :	30/10/2024
		Modifié le :	12/11/2024

Nombre de  
pages : 15

# Configurer une connexion SSH avec GitHub sous Windows

Comment le faire en six étapes

<b>1. LE PROTOCOLE SSH</b>	<b>2</b>
1.1. POURQUOI UTILISER SSH ?	2
1.2. DEFINITION DE SSH	2
1.3. SSH DANS LES MODELES OSI ET TCP/IP	2
1.4. LE CLIENT SSH SOUS WINDOWS	3
1.4.1. Vérifier si le client SSH est déjà installé	3
1.4.2. Répertoire d'installation du client SSH	3
1.5. ÉTAPES D'UNE CONNEXION SSH	4
<b>2. MODE OPERATOIRE EN SIX ETAPES</b>	<b>4</b>
2.1. ÉTAPE 1/6 : VERIFIER LA PRESENCE DE CLES SSH	5
2.2. ÉTAPE 2/6 : GENERER UNE NOUVELLE CLE SSH	5
2.2.1. Génération de la clé SSH	5
2.2.2. Passphrase	6
2.2.3. Contenu de la clé publique : id_rsa.pub	7
2.3. ÉTAPE 3/6 : AJOUTER LA CLE SSH A L'AGENT SSH (FACULTATIF : SI PASSPHRASE)	7
2.4. ÉTAPE 4/6 : AJOUTER LA CLE SSH PUBLIQUE A VOTRE COMPTE GITHUB	8
2.4.1. Compte GitHub / Settings	8
2.4.2. SSH and GPG keys	8
2.4.3. New SSH key	9
2.4.4. Add new SSH Key / Title	9
2.4.5. Le cas de plusieurs clés SSH	10
2.5. ÉTAPE 5/6 : TESTER LA CONNEXION SSH	10
2.5.1. Test de la liaison SSH et ajout du serveur	11
2.5.1.1. Confirmation avec un ping	11
2.5.2. Vérifier la présence du fichier known_hosts	11
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 »	12
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	13
2.6.1. Exemple	13
2.6.2. Vérification sur le serveur GitHub	14
<b>3. ALLER PLUS LOIN</b>	<b>15</b>
3.1. GENERATING A NEW SSH KEY AND ADDING IT TO THE SSH-AGENT (GITHUB)	15
3.2. GENERATING AN SSH KEY PAIR (ORACLE HELP CENTER)	15
3.3. UTILISATION D'UNE SSH KEY AVEC GITHUB (IONOS)	15
3.4. COMPRENDRE ET MAITRISER SSH	15

 	<b>Configurer une connexion SSH avec GitHub sous Windows</b>	Réalisé le :	30/10/2024
		Modifié le :	12/11/2024

## 1. Le protocole SSH

**SSH**, autrement dit Secure Shell, a été développé dans les années 1990 pour remplacer des protocoles non sécurisés comme Telnet. Son objectif principal est de fournir une méthode sécurisée pour gérer les systèmes à distance en chiffrant les communications entre un client et un serveur.

### 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 pulls, 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 SSH

Secure Shell (SSH) est un protocole de communication sécurisé, apparu en 1995.

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'utilisateur<sup>1</sup>.

### 1.3. SSH dans les modèles OSI et TCP/IP

À quel niveau des stacks<sup>2</sup> des modèles OSI et TCP/IP, se trouve SSH ?

Il se situe dans la couche application des modèles OSI<sup>3</sup> et TCP/IP<sup>4</sup>.

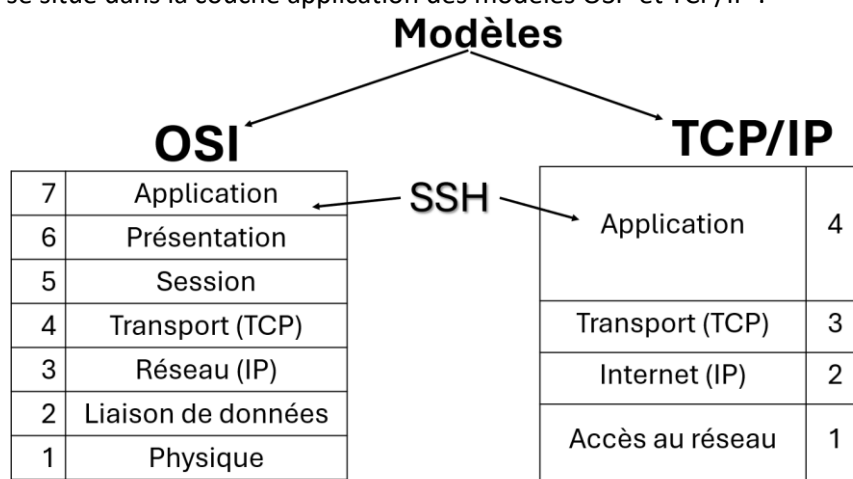


Figure 1: localisation du protocole SSH dans les modèles OSI et TCP/IP

<sup>1</sup> [https://fr.wikipedia.org/wiki/Secure\\_Shell](https://fr.wikipedia.org/wiki/Secure_Shell)

<sup>2</sup> Piles

<sup>3</sup> [https://fr.wikipedia.org/wiki/Mod%C3%A8le\\_OSI](https://fr.wikipedia.org/wiki/Mod%C3%A8le_OSI)

<sup>4</sup> [https://fr.wikipedia.org/wiki/Suite\\_des\\_protocoles\\_Internet](https://fr.wikipedia.org/wiki/Suite_des_protocoles_Internet)

 	<h1>Configurer une connexion SSH avec GitHub sous Windows</h1>	Réalisé le :	30/10/2024
		Modifié le :	12/11/2024

Nombre de pages : 15

### 1.4. Le client SSH sous Windows

Vérification du client SSH et localisation du dossier d’installation.

#### 1.4.1. Vérifier si le client SSH est déjà installé

Pour cela il faut lancer PowerShell et utiliser la commande :

```
Get-WindowsCapability -Online | Where{ $_.Name -like 'OpenSSH.Client*' }
```

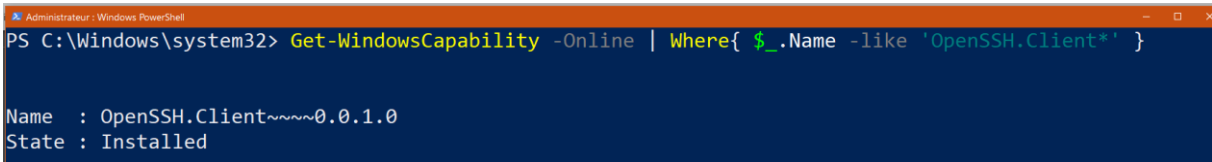


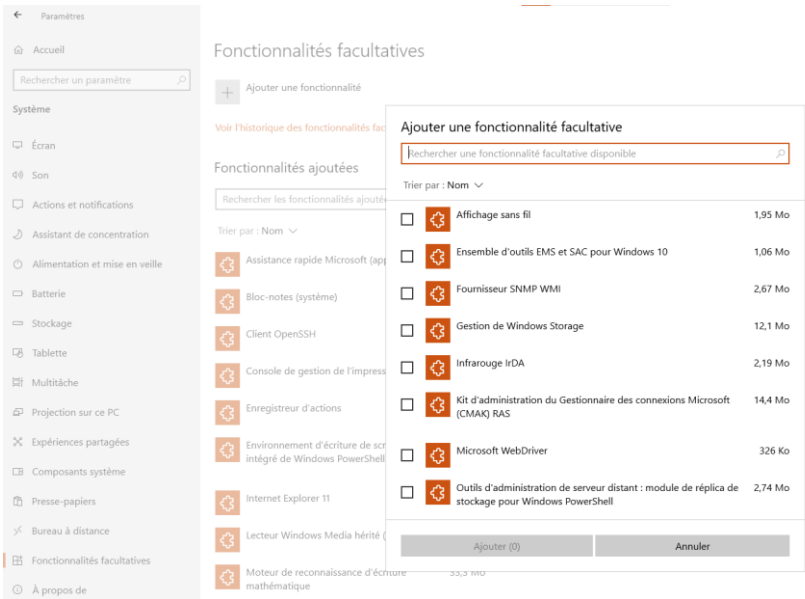
Image 1-Commande Powershell Get-WindowsCapability

S’il n’est pas installé il faudra le faire avec la commande :

```
Add-WindowsCapability -Online -Name OpenSSH.Client
```

Ou bien à partir de l'interface graphique de Windows à l'aide de l'ajout d'une fonctionnalité.

Rechercher : « Client OpenSSH ».



#### 1.4.2. Répertoire d’installation du client SSH

Il se trouve dans le répertoire OpenSSH. Le chemin est : C:\Windows\System32\OpenSSH\ et son contenu est :

 	<h1>Configurer une connexion SSH avec GitHub sous Windows</h1>	Réalisé le :	30/10/2024
		Modifié le :	12/11/2024

Nom	Taille
 LICENSE.txt	19 Ko
 NOTICE.txt	36 Ko
 scp.exe	417 Ko
 sftp.exe	444 Ko
 ssh.exe	1 215 Ko
 ssh-add.exe	585 Ko
 ssh-agent.exe	538 Ko
 ssh-keygen.exe	838 Ko
 ssh-keyscan.exe	644 Ko
 ssh-pkcs11-helper.exe	499 Ko
 ssh-sk-helper.exe	634 Ko

Pour utiliser le client SSH il faudra exécuter : "ssh.exe".

## 1.5. Étapes d'une connexion SSH

Le client SSH établit une connexion TCP/IP avec le serveur SSH sur le **port 22** par défaut. Ce processus comporte plusieurs étapes :

- Établissement de la connexion :**  
Le client initie une connexion vers le serveur SSH. Le serveur envoie alors sa clé publique au client pour que celui-ci puisse vérifier l'identité du serveur.
- Échange des clés :**  
SSH utilise l'algorithme **Diffie-Hellman**<sup>5</sup> pour établir une clé de session partagée et unique, qui sera utilisée pour chiffrer la session de communication.
- Authentification du serveur :**  
Le client vérifie l'identité du serveur en utilisant une clé publique.
- Authentification du client :**  
Le serveur vérifie l'identité du client en utilisant des méthodes telles que les mots de passe, les clés publiques, ou les certificats.
- Établissement d'un tunnel chiffré :**  
Une fois la connexion sécurisée, toutes les données échangées entre le client et le serveur sont cryptées par la clé de session. L'utilisateur peut maintenant exécuter des commandes ou transférer des fichiers de manière sécurisée.

## 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

<sup>5</sup> [https://fr.wikipedia.org/wiki/%C3%89change\\_de\\_cl%C3%A9s\\_Diffie-Hellman](https://fr.wikipedia.org/wiki/%C3%89change_de_cl%C3%A9s_Diffie-Hellman)

 	<h2>Configurer une connexion SSH avec GitHub sous Windows</h2>	Réalisé le :	30/10/2024
		Modifié le :	12/11/2024

Nombre de pages : 15

### 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"
```

source<sup>6</sup>

```
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"
```

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):
```

<sup>6</sup> <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

 	<h2>Configurer une connexion SSH avec GitHub sous Windows</h2>	Réalisé le :	30/10/2024
		Modifié le :	12/11/2024

Nombre de pages : 15

### 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:

```

```

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 :	30/10/2024
		Modifié le :	12/11/2024

Nombre de pages : 15

← → ↕ ↑

C:\Users\Djamel

Accès rapide	Nom	Modifié le	Type
Bureau	.atom	18/10/2022 18:28	Dossier de fichi...
Téléchargements	.bluefish	18/10/2022 18:08	Dossier de fichi...
Documents	.cache	24/12/2022 18:56	Dossier de fichi...
Images	.dotnet	18/12/2022 18:21	Dossier de fichi...
	.ms-ad	02/10/2022 22:35	Dossier de fichi...
	.ssh	03/09/2024 19:31	Dossier de fichi...
	.templateengine	15/03/2024 17:32	Dossier de fichi...

← → ↕ ↑

C:\Users\Djamel\.ssh

Accès rapide	Nom	Modifié le	Type	Taille
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				
Images				

2.2.3. Contenu de la clé publique : id\_rsa.pub

id\_rsa.pub - Bloc-notes

Fichier Edition Format Affichage Aide

ssh-rsa  
AAAAAB3NzaC1yc2EAAAADAQABAAQCAQCY6m3/e/9F4ItxuWk5JU2Ay60/rUb5LqDpRScKb4m1fLF3iKw3zLe/1y/Gy1ZBARVECPqC8KVeTrFuk5B3IBU3+mpWYtL1QpNpJIX9vvq1K7XSzGXAYyiYuD1Rdzgiwa4UVifvHEyAhUgzNKsxQWk/hj7Yfak0rkrhwhqYIooPwttnv1ZNNS2rMzWit2FyVH2qoOfBXancWOI/5kvqC+/snjYT8LJyKGXqo6iovVHrf6Q3+NjxypVl0Fcs881q08E0VMJnN01/2+T8bpKELIkK7JRbmQjNPFh2+36SpqXsLbnJGgzVHhCjURO//jsR3Y1mzwahXaptqeUUprEweth8yUTrz5cXYz1XKUNGYZtMN2nXB3LrCzX+jB2/Cr1SgcI6Q1/ED+fMIRp2o4wCuKes9q18x2rk1LzGTsyOXDwJK8nMc/iQZ8KnF/wC+p34K4yVxdnpEixRHESsj+ObUD1px0yCjiHep/rzS/hlmQiuwZISQ8M+v9nh2jzLeZuthou45SEd8jz1/buPcyI0z0BVKwsbqVlKv7QsTYXJbXULbcF5S1w8uB3Jo11Et2FvEABE5i1MgxHhwX31zEjnnCWUUYtMqX5/09958ROVwKvKsds4bM0w+BPE5E1IAqvBbBri0e6GS3ULLNCwVbhcOPBRZkh3VwZuV4X+8IBFSikvQ== ikradesk@gmail.com

MINGW64/c:/Users/Djamel/.ssh

Djamel@zBook17 MINGW64 ~/\.ssh

\$ ls  
id\_rsa id\_rsa.pub

Djamel@zBook17 MINGW64 ~/\.ssh  
\$ cat id\_rsa.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQCY6m3/e/9F4ItxuWk5JU2Ay60/rUb5LqDpRScKb4m1fLF3iKw3zLe/1y/Gy1ZBARVECPqC8KVeTrFuk5mpWYtL1QpNpJIX9vvq1K7XSzGXAYyiYuD1Rdzgiwa4UVifvHEyAhUgzNKsxQWk/hj7Yfak0rkrhwhqYIooPwttnv1ZNNS2rMzWit2FyVH2qoOfBXancWOI/snjYT8LJyKGXqo6iovVHrf6Q3+NjxypVl0Fcs881q08E0VMJnN01/2+T8bpKELIkK7JRbmQjNPFh2+36SpqXsLbnJGgzVHhCjURO//jsR3Y1mzwahXaptweth8yUTrz5cXYz1XKUNGYZtMN2nXB3LrCzX+jB2/Cr1SgcI6Q1/ED+fMIRp2o4wCuKes9q18x2rk1LzGTsyOXDwJK8nMc/iQZ8KnF/wC+p34K4yVxdnpEj+ObUD1px0yCjiHep/rzS/hlmQiuwZISQ8M+v9nh2jzLeZuthou45SEd8jz1/buPcyI0z0BVKwsbqVlKv7QsTYXJbXULbcF5S1w8uB3Jo11Et2FvEABE5wx31zEjnnCWUUYtMqX5/09958ROVwKvKsds4bM0w+BPE5E1IAqvBbBri0e6GS3ULLNCwVbhcOPBRZkh3VwZuV4X+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 la 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)"
```

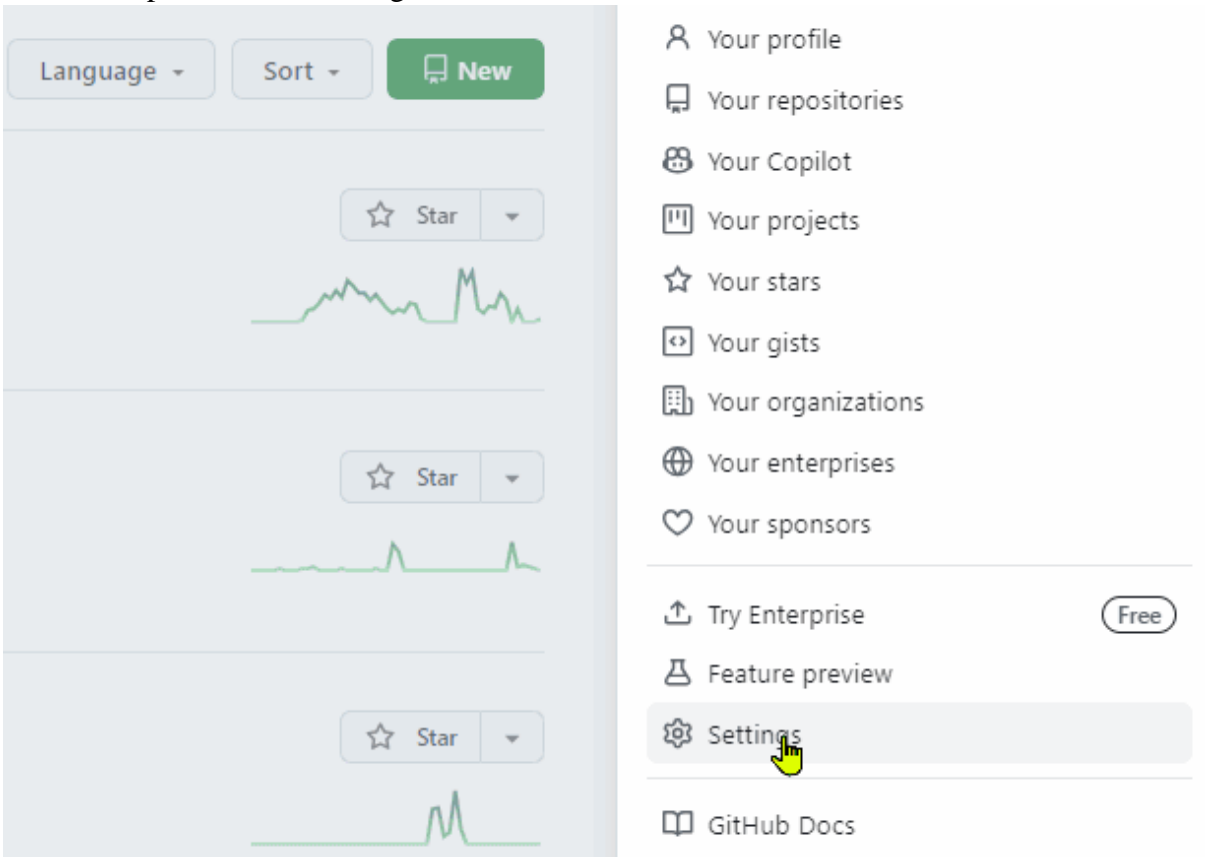
	<b>Configurer une connexion SSH avec GitHub sous Windows</b>	Réalisé le :	30/10/2024
		Modifié le :	12/11/2024

Nombre de pages : 15

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

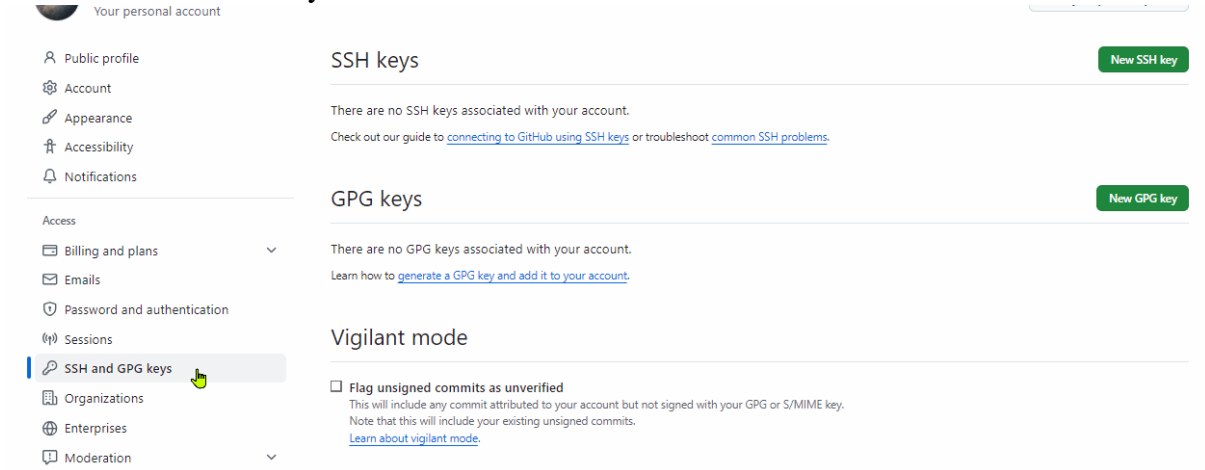
Aller dans Compte GitHub / Settings

### 2.4.1. Compte GitHub / Settings



Ensuite aller dans SSH and GPG keys

### 2.4.2. SSH and GPG keys




Ensuite cliquer sur « New SSH key »



	<h1>Configurer une connexion SSH avec GitHub sous Windows</h1>	Réalisé le :	30/10/2024
		Modifié le :	12/11/2024

Nombre de pages : 15

### 2.4.3. New SSH key



Djamelikra (Djamelikra)
Your personal account

Go to your personal profile

- Public profile
- Account
- Appearance
- Accessibility
- Notifications

### SSH keys

There are no SSH keys associated with your account.

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

New SSH key

### GPG keys

New GPG key

Ensuite aller dans la section : « Add new SSH Key », champ « title » **donner un nom** à la clé SSH et coller le contenu de la clé publique : **id\_rsa.pub** (vue précédemment)

id\_rsa.pub - Bloc-notes

Fichier Edition Format Affichage Aide

```


ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCAQCY6m3/e/9F4ItxuWk5JU2Ay60/rUb5LqDDrScKb4m1fLF3iKw3zLe/1y/GylZBARVECPqC8KVeTrFuk5B3IBU3+mpWYtL
1QpNpJIX9vvq1K7XSzGXAYyiYuD1RdzgiWa4UVifvHEYAhUgzNKsxQWk/hj7Yfak0rkrhwhqYIooPwttnv1ZNNS2rMzWit2FyVH2qoOfBXancWOI/5kvqC
+/snjYT8LJyKGxqo6ioVHrf6Q3+NjxYpVl0FcS881q08E0VMJnNO1/2+T8bpKELIK7JRbmQjNPFh2+36SpqXsLbNjGgzVHhCjURO//jsR3YlzmzWahXaptqeUU
prEweth8yUTrz5cXYz1XKUNGYZtMN2nXB3LrCzX+jB2/Cr1SgcI6Q1/ED+fMIRp2o4wCuKes9q18x2rk1LzGTsyOXDwJK8nMc/iQZ8KnF/wC
+p34K4yVxdnpEixRHESsj+ObUD1px0yCjiHep/rzS/hlmQiuwZISQ8M
+v9nh2jzLeZuthoU45SEd8jz1/buPCyI0z0BVKwsbqbVLkv7QsTYXJbXULbcF5S1w8uB3Jo11Et2FvEABE5i1MgxHhwX3lzeJnnCwUUYtMqX5/09958ROVwKvKs
ds4bM0w+BPE5EIIAqvBbBri0e6GS3ULLNCWvBhcOPBRZkh3VWZuV4X+8IBFSIkVQ== ikradesk@gmail.com

```

Ln 1, Col 1100% Unix (LF)UTF-8

dans le champ **key** :

### 2.4.4. Add new SSH Key / Title



Djamelikra (Djamelikra)
Your personal account

Go to your personal profile

- Public profile
- Account
- Appearance
- Accessibility
- Notifications

Add new SSH Key

Title

zBooki7Win32Go

Key type

Authentication Key

Key

```

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCAQCY6m3/e/9F4ItxuWk5JU2Ay60/rUb5LqDDrScKb4m1fLF3iKw3zLe/1y/GylZBARVECPqC8KVeTrFuk5B3IBU
3+mpWYtL1QpNpJIX9vvq1K7XSzGXAYyiYuD1RdzgiWa4UVifvHEYAhUgzNKsxQWk/hj7Yfak0rkrhwhqYIooPwttnv1ZNNS2rMzWit2FyVH2qoOfBXan
cWOI/5kvqC+/snjYT8LJyKGxqo6ioVHrf6Q3+NjxYpVl0FcS881q08E0VMJnNO1/2+T8bpKELIK7JRbmQjNPFh2+36SpqXsLbNjGgzVHhCjURO//jsR
3YlzmzWahXaptqeUUprEweth8yUTrz5cXYz1XKUNGYZtMN2nXB3LrCzX+jB2/Cr1SgcI6Q1/ED+fMIRp2o4wCuKes9q18x2rk1LzGTsyOXDwJK8nMc/iQ
Z8KnF/wC+p34K4yVxdnpEixRHESsj+ObUD1px0yCjiHep/rzS/hlmQiuwZISQ8M+v9nh2jzLeZuthoU45SEd8jz1/buPCyI0z0BVKwsbqbVLkv7QsTYXJb
XULbcF5S1w8uB3Jo11Et2FvEABE5i1MgxHhwX3lzeJnnCwUUYtMqX5/09958ROVwKvKsds4bM0w+BPE5EIIAqvBbBri0e6GS3ULLNCWvBhcOPBRZk
h3VWZuV4X+8IBFSIkVQ== ikradesk@gmail.com

```

Add SSH key


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

©Djamel CHABANE 2024


Configurer une connexion SSH avec GitHub sous Windows

9 / 15

 Nombre de pages : 15	<h1>Configurer une connexion SSH avec GitHub sous Windows</h1>	Réalisé le :	30/10/2024
		Modifié le :	12/11/2024



## 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, in fine, on obtient un message de succès

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:qaGrRCgcdxr11KJv9ypcbm8pTQ59yq8U60z19PpZ8LA  
 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

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

## 2.5. Étape 5/6 : Tester la connexion SSH

Pour s’assurer que tout fonctionne correctement, il faut tester la connexion SSH avec GitHub. Cette vérification peut s’effectuer de trois manières :

	<h2>Configurer une connexion SSH avec GitHub sous Windows</h2>	Réalisé le :	30/10/2024
		Modifié le :	12/11/2024

- En tapant une commande
- En vérifiant la présence du fichier `known_hosts`
- En clonant un repository

### 2.5.1. Test de la liaison SSH et ajout du serveur

En tapant la commande : `ssh -T git@github.com`

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 ») :

#### 2.5.1.1. Confirmation avec un ping

En lançant une console PowerShell<sup>7</sup> :

```
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 :

<sup>7</sup> <https://www.it-connect.fr/chapitres/powershell-prise-en-main-console-comment-utiliser/>

 	<h1>Configurer une connexion SSH avec GitHub sous Windows</h1>	Réalisé le :	30/10/2024
		Modifié le :	12/11/2024

```

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

```

← → ↶ ↷

C:\Users\Djamel\.ssh

	Nom	Modifié le	Type	Taille
	id_rsa	03/09/2024 19:31	Fichier	4 Ko
	id_rsa.pub	03/09/2024 19:31	Microsoft Publi...	1 Ko
	known_hosts	05/09/2024 18:09	Fichier	1 Ko

known\_hosts - Bloc-notes

Fichier Edition Format Affichage Aide

```
github.com ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOMqknkVzrm0SdG6U0oqKLsabgH5C9okwi0dh2l9GKJl
github.com ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCj7ndNxQowgcQnjsHcLrqPEiiphnt
+VTTvDP6mHBL9j1aNUkY4Ue1gvwnGLVl0hGeYrnZaMgRK6+PKCUXaDbC7qtbW8gIkhL7aGCSOr/C56SJMy/BCZfxd1nWzAOXS
```

Ln 1, Col 1 100% Unix (LF) UTF-8

2.5.3. Clonage d’un repository par SSH

Cette dernière méthode nécessite quatre étapes :

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

Algorithmic

Private

Unwatch 1 Fork 0 Star 0

main 1 Branch 0 Tags

Go to file Add file <> Code

Djamelikra ssh + doc 1 d13873f · 27 minutes ago 557 Commits

AlgoVba vba confirm 1 last month

TSP distance ok 5 months ago

Access VBA Expert.docx vba list 1 last month

Catalogue Formation Audit VBA Access.docx array 1 2 months ago

Configurer une connexion SSH avec Github N... ssh + doc 1 27 minutes ago

Algorithmic in many languages

Readme Activity 0 stars 1 watching 0 forks

Releases

### 2.5.3.2. Cliquer sur l’onglet « SSH »

Algorithmic

Private

Unwatch 1 Fork 0 Star 0

main 1 Branch 0 Tags

Go to file Add file <> Code

Djamelikra ssh + doc 1

AlgoVba vba confirm 1

TSP distance ok

Access VBA Expert.docx vba list 1

Catalogue Formation Audit VBA Access.docx array 1

Configurer une connexion SSH avec Github\_N... ssh + doc 1

Créer un repository avec GIT et GITHUB.docx docGit 1

README.md Initial commit

Local Codespaces

Clone

HTTPS SSH GitHub CLI

git@github.com:Djamelikra/Algorithmic.git

Use a password-protected SSH key.

Open with GitHub Desktop

Open with Visual Studio

Download ZIP

Algorithmic in many languages

Readme Activity 0 stars 1 watching 0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

©Djamel CHABANE 2024

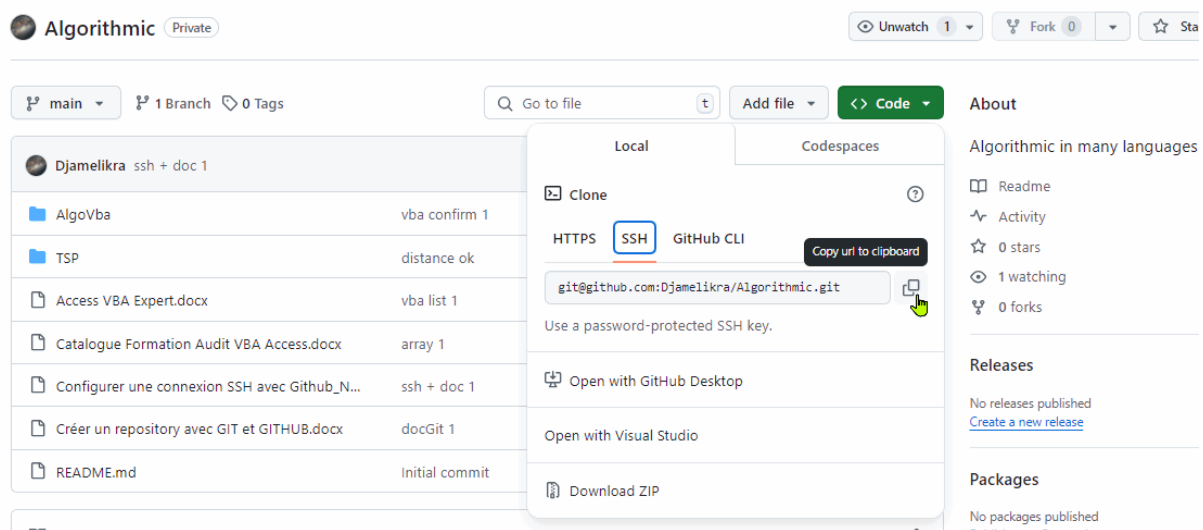
Configurer une connexion SSH avec GitHub sous Windows

12 / 15

 	<h2>Configurer une connexion SSH avec GitHub sous Windows</h2>	Réalisé le :	30/10/2024
		Modifié le :	12/11/2024

Nombre de pages : 15

### 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
$

```

## 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 avec la commande:

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

### 2.6.1. Exemple

Voyons cela avec un cas concret :

 	<h1>Configurer une connexion SSH avec GitHub sous Windows</h1>	Réalisé le :	30/10/2024
		Modifié le :	12/11/2024

Nombre de pages : 15

```
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)
$ |
```

2.6.2. Vérification sur le serveur GitHub

Algorithmic

Private

Algorithmic in many languages

VBA

Updated now

Star



Tout fonctionne comme prévu 👍

	<b>Configurer une connexion SSH avec GitHub sous Windows</b>	Réalisé le :	30/10/2024
Nombre de pages : 15		Modifié le :	12/11/2024

### 3. Aller plus loin

Voici quelques suggestions de liens pour approfondir le sujet :

#### 3.1. 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. Generating an SSH Key Pair (Oracle Help Center)

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

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

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

#### 3.4. Comprendre et maîtriser SSH

<https://www.it-connect.fr/cours/comprendre-et-maitriser-ssh/>