

# SSH tutorial

A tutorial **PowerShell**-hez készült, bizonyos részeihez **adminisztrátori** jogosultságra is szükség van!

## Előkészületek

Vizsgáljuk meg, hogy létezik -e a `.ssh` mappa. Ehhez navigáljunk az aktuális felhasználó home mappájába a `cd` parancs segítségével.

```
cd $HOME
```

Listázzuk ki a felhasználó home mappájának tartalmát az `ls`, vagy a `dir` parancsok egyikével.

```
ls
```

Amennyiben létezik a `.ssh` mappa, úgy lehet már korábban létre lett hozva kulcs. Ennek jele, hogy az alábbi fájlok közül valamelyik már létezik.

- `id_rsa.pub`
- `id_ecdsa.pub`
- `id_ed25519.pub`

Vizsgáljuk meg a `.ssh` mappa tartalmát.

```
ls .ssh
```

Amennyiben nem találtunk fájlt könnyen generálhatunk egy sajátot:

```
ssh-keygen -t ed25519 -C "otthoni saját gép"
```

- `-t`: **type**, azaz a kulcs típusa. A kódolási algoritmust határozza meg. Elhagyva RSA kulcsot generál.
- `-C`: **comment**, azaz megjegyzést adhatunk a kulcshoz.

A kulcs generálásakor meg kell adni, hogy mi legyen a neve. A típus alapján felajánl egy alapértelmezett nevet, ezt az enter lenyomásával elfogadhatjuk.

```
Generating public/private ed25519 key pair.  
Enter file in which to save the key (C:\Users\kepzes\.ssh\id_ed25519):
```

Amennyiben már létezne a megadott kulcs, úgy a fájl neve megadása után rákérdez, hogy szeretnénk-e felülírni a már ott lévőket. Ilyenkor célszerű lehet újat generálni más néven.

```
Overwrite (y/n)?
```

Amennyiben még nem létezett a `.ssh` mappa most létrehozza, ezt jelzi is.

```
C:\Users\kepzes\.ssh
```

A kulcs biztonságossá tételéhez célszerű megadni passphrase-t, azaz egy jelmondatot, majd ezt ismét meg kell adni.

```
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:
```

Amennyiben nem pont ugyanaz került begépelésre esetlegesen újra meg kell adni kétszer.

```
Passphrases do not match. Try again.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:
```

Siker esetén két fájlt hoz létre a `.ssh` mappában, jelent kódolás mellett a `id_ed25519` fájlt, ez lesz a privát kulcs és a `id_ed25519.pub` fájlt. Ez lesz a publikus. Utóbbi tartalmát ki is írja a konzolra.

```
Your identification has been saved in C: \Users\ Test Elek\.ssh\id_ed25519.  
Your public key has been saved in C: \Users\Test Elek\.ssh\id_ed25519.pub.  
The key fingerprint is:  
SHA256:LKvcIoFQ6MaJ7opG6Ickq2fwhlT+pca6Nin6RD/Ttt0 otthoni saját gép
```

## Publikus kulcs kiolvasása

Az `id_ed25519.pub` kulcs tartalmát a `cat` paranccsal megjeleníthetjük:

```
cat $HOME\.ssh\id_ed25519.pub
```

Amennyiben kényelmetlen a konzolról másolni, úgy a `clip` parancs segítségével rögtön a vágólapra is másolhatjuk.

```
cat $HOME\.ssh\id_ed25519.pub | clip
```

## Kulcs hozzáadása a github.com-hoz

A <http://github.com> oldalon bejelentkezés után az alábbi lépéseken kell végig menni.

Profil ikon > Settings > SSH and GPG keys

New SSH key (Új SSH kulcs) gombra kattintva megjelenik az űrlap, ahol fel lehet vinni a kulcsot.

Kitöltendő:

- Title (cím): Rövid leírás, hogy a kulcs mihez tartozik, pl.: OktatoLaptop
- Key (kulcs): A **publikus** SSH kulcs tartalma
- Add SSH key (SSH kulcs hozzáadása) gombra kattintva el is készült.

## Kulcs tesztelése

Az alábbi parancs segítségével megpróbál kapcsolatot létesíteni a github.com szerverével.

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

Első csatlakozáskor még nem ismer a github szerverét, így egy biztonsági kérdésen kell átesnünk.

```
The authenticity of host 'github.com (140.82.121.4)' can't be established.  
ECDSA key fingerprint is SHA256: p2QAMXNICITJYWeIOttrVc98/R1BUFWu/LiyKgUfOM.  
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Ellenőrizzük az ujjlenyomatot! A megfelelő fingerprint beilleszthető, és az ellenőrzést elvégezzük helyettünk.

- <https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/githubs-ssh-key-fingerprints>

Amennyiben mindent sikerült jól beállítani, úgy a felhasználónevünkön köszönt az oldal, hiszen a kulcs alapján be tudott azonosítani.

```
Hi gitkepzes2022!  
You've successfully authenticated, but GitHub does not provide shell access.
```

## SSH Agent

Az itt leírt parancsok futtatásához szükséges adminisztrátori jogosultság!

- Kényelmetlen lehet állandóan begépelni a jelszót, megoldás: SSH Agent
- SSH Agent: Kezeli az SSH kulcsokat, tárolja a passphrase-t (jelmondatot).
- A tárolás ideje megadható.

Fut-e az ssh-agent szolgáltatás?

```
Get-Service -Name ssh-agent
```

Status	Name	DisplayName
-----	-----	-----
Stopped	ssh-agent	OpenSSH Authentication Agent

Amennyiben nem fut beállíthatjuk, hogy az indítása legyen manuális

```
Get-Service -Name ssh-agent | Set-Service -StartupType Manual
```

Most már elindítható az ssh-agent. Használjuk a Windows telepítéséből származó Open SSH-t.

```
C:\Windows\System32\OpenSSH\ssh-agent.exe
```

Ellenőrizzük, hogy fut -e.

```
Get-Service -Name ssh-agent
```

Status	Name	DisplayName
-----	-----	-----
Running	ssh-agent	OpenSSH Authentication Agent

Adjuk hozzá **privát SSH kulcsunkat**.

```
C: \Windows\System32\OpenSSH\ssh-add.exe $HOME\.ssh\id_ed25519
```

Az ssh ügynökhöz hozzáadáskor meg kell adni a passphrase-t, azaz a jelmondatot.

```
Enter passphrase for C:\Users\kepzes\.ssh\id_ed25519
```

## Open SSH és a git kapcsolata

A Visual Studio Code és a konzolos git parancs akkor lesz működőképes, ha a git parancsokhoz ugyanazt az **ssh.exe**-t futtatják.

Olvassuk ki, hogy jelenleg mi a beállítás:

```
git config --get core.sshCommand
```

Állítsuk be **globálisan**, hogy a windowson található **Open SSH**-t használja:

```
git config --global core.sshCommand 'C:\\Windows\\System32\\OpenSSH\\ssh.exe'
```

*A dupla backslash karakterek azért van szükség, mert az elérési útvonal szövegként van megadva és a \ jelet, mint escape karaktert benyelné.*