

# Connexion automatique d'un utilisateur sous Debian

## Connexion automatique

Un programme permet de les terminaux : `getty`. L'idée est donc de lui demander de connecter automatiquement un utilisateur lorsqu'il est en `tty1`, autrement dit depuis le terminal 1 du PC.

La démarche est expliquée dans la documentation de *Arch* : [getty - ArchWiki](#).

Il convient d'abord de créer un fichier de configuration alternative afin de pas perdre ses réglages lors d'une mise à jour du système :

```
sudo systemctl edit getty@tty1.service
```

Cela va créer un fichier `override.conf` reprenant tous la configuration d'origine en commentaire. On décommente et modifie le fichier afin d'avoir les lignes suivantes :

```
[Service]
# # the VT is cleared by TTYVTDisallocate
# # The '-o' option value tells agetty to replace 'login' arguments with an
# # option to preserve environment (-p), followed by '--' for safety, and then
# # the entered username.
ExecStart=
ExecStart=-/sbin/agetty --noclear --autologin <username> %I $TERM
```

- `[Service]` permet d'indiquer l'emplacement des modifications à apporter.
- Le premier `[ExecStart]` permet de remettre à zéro le précédent réglage.
- La dernière ligne permet de connecter l'utilisateur `<username>` sans entrer de mot de passe.

Pour ma part, l'édition s'est faite dans *Nano*, au moment de l'enregistrement, j'ai du changer le nom du fichier : `/etc/systemd/system/getty@tty1.service.d/override.conf`.

Si on laisse les options `-o '-p -f -- \\u'`, le nom d'utilisateur sera pré-rempli, mais il faudra toujours entrer le mot de passe.

# Lancement automatique d'une commande

Cela se joue dans le fichier `.zshrc` de l'utilisateur.

Étant donné que je fais appel à une fonction dans mon script (c'est un peu imbriqué dans tous les sens mon affaire...), je le place à la toute fin de mon fichier.

```
[[ `tty` == /dev/tty1 ]] && <command>
```

La commande ne respecte pas le standard *POSIX*, mais ça va bien pour *zsh*. Autrement, j'aurais pu mettre la commande suivante, sans quoi *zsh* interprète `==` :

```
[ `tty` "==" /dev/tty1 ] && <command>
```

## Sources

- [linux - Automatic root login in Debian 8.0 \(console only\) - Super User](#)
- [getty - ArchWiki](#)
- [Why does `==` behave differently inside `...` in zsh and bash - Unix & Linux Stack Exchange](#)

---

Révision #2

Créé 9 janvier 2024 13:34:25 par Mickaël G.

Mis à jour 9 janvier 2024 13:57:12 par Mickaël G.