

Journal : Un irssi (ou autre chose) dans un tmux sur un serveur, avec systemd

Posté par nud le 26 mars 2019 à 22:02. Licence CC BY-SA.
Étiquettes : irssi, tmux, irc, serveur, init, service, debian



Sommaire

- [Comment démarrer irssi dans tmux?](#)
- [Avant, avec sysvinit](#)
- [Maintenant: une instance "utilisateur" de systemd](#)
 - [Créer une unit pour notre service](#)
 - [Démarrer lors du boot](#)
- [Pourquoi c'est cool?](#)

Bonjour Nal,

Cela fait quelques années que j'ai un irssi qui tourne gentiment sur un serveur à moi chez Claude. Il tourne dans un tmux, et pour être sûr de ne rien rater, j'ai toujours fait en sorte que ladite instance tmux soit démarrée automatiquement quand le serveur boote.

Avant, j'avais un script d'init qui faisait ça. Ce script était placé dans `/etc/init.d/` et ça marchait. Mais c'était un peu nul parce que les scripts d'init sont globaux et il n'est pas vraiment possible pour un utilisateur d'avoir ses propres scripts d'init sans faire des trucs louches. À la faveur d'une migration vers un nouveau serveur, et vu que ça fait deux versions de Debian qu'on a quand même systemd, je me suis dit que j'allais essayer de "faire ça bien" à la mode systemd.

Je me suis dit que c'était intéressant pour d'autres utilisations aussi, alors je me suis dit que j'allais écrire un petit journal pour la peine. C'est sans doute plein d'inexactitudes et de raccourcis :-)

Comment démarrer irssi dans tmux?

Alors ça, c'est facile:

```
/usr/bin/tmux new-session -s irc -d /usr/bin/i
```

La commande `new-session` permet de créer une nouvelle session dédiée, dont le nom est `irc`. Le paramètre `-d` permet de détacher immédiatement la session tmux, et ensuite suit la commande à exécuter. Jetez un œil au manuel si vous voulez voir les différentes possibilités offertes, là je reste dans la simplicité :-)

Après, pour me connecter à cette instance, j'utilise un alias qui fait ceci:

```
alias irc='ssh -t <server> tmux attach-sessior
```

Et, grosso-modo, ça juste marche, sauf si la machine est redémarrée évidemment.

Avant, avec sysvinit

J'ai déjà dit qu'avec sysvinit, il n'était pas vraiment possible d'avoir un init script défini au niveau de l'utilisateur. Le workaround évident pour quiconque est root sur sa machine, c'est simple, c'est de faire un script d'init traditionnel et de le mettre dans `/etc/init.d`. Voici donc mon vieux `/etc/init.d/tmux-irc`:

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          tmux-irc
# Should-Start:      $network
# Should-Stop:       $network
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
### END INIT INFO

set -e

NAME=tmux-irc
DESC="tmux session for IRC"
DAEMONUSER=nud
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin

. /lib/lsb/init-functions

start_it_up()
{
    log_daemon_msg "Starting $DESC" "$NAME"
    su $DAEMONUSER -c "/usr/bin/tmux new-session"
    log_end_msg $?
}

shut_it_down()
{
    log_daemon_msg "Stopping $DESC" "$NAME"
    su $DAEMONUSER -c "/usr/bin/tmux kill-sessic
    log_end_msg $?
}

case "$1" in
    start)
        start_it_up
        ;;
    stop)
        shut_it_down
        ;;
    restart|force-reload)
        shut_it_down
        start_it_up
        ;;
    *)
        echo "Usage: /etc/init.d/$NAME {start|stop|
    exit 2
    ;;
esac

exit 0
```

C'est du script d'init tout classique, la seule chose inhabituelle est probablement l'appel à `su` pour exécuter tmux comme utilisateur normal.

Une fois le script d'init en place, il suffit de l'activer à la mode de votre distribution. Ça marche aussi avec systemd, par ailleurs.

Maintenant: une instance "utilisateur" de systemd

Si tout le monde connaît le systemd qui tourne avec le PID 1, au niveau du système, on connaît moins le systemd qui tourne avec un PID quelconque, et qui gère les services liés à un utilisateur donné.

C'est quelque chose que j'avais découvert à l'époque où Gnome envisageait de l'utiliser à la place de `gnome-session`, et de nos jours, [pam-systemd](#) et [systemd-logind](#) permettent à systemd d'instancier des *slices* pour les différents utilisateurs et des *scopes* pour les différentes sessions. Le wiki de Archlinux contient [la documentation la plus complète que j'aie trouvée sur le sujet](#).

Si vous exécutez `systemctl status`, vous pourrez retrouver ce type d'info dans l'arborescence:

```
└─user.slice
   └─user-1000.slice
      └─user@1000.service
         └─tmux-irc.service
            │ │ └─5138 /usr/bin/tmux new-session -c
            │ │ └─5139 /usr/bin/irssi
            │ │ └─init.scope
            │ │ └─18920 /lib/systemd/systemd --user
            │ │ └─18921 (sd-pam)
```

C'est relativement anodin au jour le jour (le plus souvent ça permet de s'assurer que rien ne subsiste une fois qu'un utilisateur se délogue), mais via ce mécanisme, systemd permet aussi à certains utilisateurs de fournir leurs propres *units*. C'est cela que nous allons utiliser pour notre session tmux.

Créer une unit pour notre service

Il est possible de placer les *units* propres à un utilisateur dans le dossier `~/.config/systemd/user/`. Voici donc mon unit `tmux-irc.service` (un peu simpliste mais suffisante):

```
[Unit]
Description=tmux instance with irssi running i

[Service]
ExecStart=/usr/bin/tmux new-session -d -s irc
Type=forking
Restart=always

[Install]
WantedBy=default.target
```

Une fois ce fichier placé au bon endroit, il suffit de reloader `systemd` puis on peut démarrer le service:

```
$ systemctl --user daemon-reload
$ systemctl start daemon-tmux-irc
$ systemctl status --user tmux-irc
```

Par contre si vous rebootez la machine, ça ne marche pas. En fait, l'instance utilisateur de systemd n'est créée que lorsqu'un utilisateur se logue. Donc, pour le côté "présence constante", ça n'est pas suffisant.

Note: Théoriquement, il me semble tmux devrait également être arrêté lors du log-out mais ce n'est pas le cas, en tout cas sous Debian. Je n'ai pas creusé mais quelqu'un a peut-être l'explication.

Démarrer lors du boot

La configuration par défaut de systemd-logind est conçue afin de s'assurer que les processus lancés par un utilisateur ne lui survivent pas. Du coup, cette instance est lancée lors du login, et coupée lors du log out *de toutes les sessions de l'utilisateur* (une seule instance est lancée même si vous vous connectez 5 fois à la machine). Par ailleurs, si votre session tmux est tuée à chaque fois que vous vous déconnectez de votre serveur SSH, c'est à cause de ça.

Evidemment, ce n'est pas ce que l'on recherche ici, mais heureusement il est possible de dire à systemd que l'instance d'un utilisateur soit tout le temps présente, du boot au shutdown, ce qui permet d'avoir des processus (services) utilisateurs qui tournent sans avoir de session ouverte.

Il suffit de faire ceci (en root):

```
# loginctl enable-linger $user
```

Voilà. Rebootez la machine pour en avoir le cœur net. Ça marche :-)

Pourquoi c'est cool?

- vous pourrez encourager le petit dernier à créer son appli web sans lui donner les droits root, et ça c'est cool.
- vous pourrez utiliser tmux dans irssi (ou est-ce l'inverse?) et ça c'est cool.
- vous pourrez faire ce que vous voulez avec ça et ça c'est cool.

(17 commentaires), Markdown EPUB

Très utile
Posté par gnumdk (site web personnel) le 27 mars 2019 à 08:53. Évalué à 9.

J'ai beaucoup utilisé systemd lors de la mise en place des postes Ubuntu 18.04 dans mon école.

C'est vraiment mieux que l'ancien monde où il fallait jouer soit avec les scripts de session X soit avec cette daube de `~/config/autostart`

irssi ?
Posté par Jona le 27 mars 2019 à 09:14. Évalué à 3. Dernière modification le 27 mars 2019 à 09:14.

[irssi](#)^W est un client IRC.

[^] # Re: irssi ?
Posté par oinkoink_daotter le 27 mars 2019 à 11:08. Évalué à 1.

Moi qui pensais que c'était la version française d'un CISO connecté :)

Wiki ArchLinux
Posté par Glandos le 27 mars 2019 à 10:32. Évalué à 10.

Le wiki de Archlinux contient la documentation la plus complète que j'aie trouvée sur le sujet.

Comme toujours.

Je ne sais pas comment ils font. Mais la documentation de leur wiki est de qualité, et souvent bien à jour.

Cette communauté est géniale.

[^] # Re: Wiki ArchLinux
Posté par damaki le 27 mars 2019 à 14:00. Évalué à 3.

Leur OS aussi.

[^] # Re: Wiki ArchLinux
Posté par gUI (Mastodon) le 27 mars 2019 à 14:22. Évalué à 6.

Oui, de mon temps c'était Gentoo qui était la référence sur laquelle on finissait systématiquement quand on cherchait de la doc. Ils ont eu un gros soucis et ont tout perdu (je crois). Heureusement celle de Arch est au moins du même niveau.

...

En théorie, la théorie et la pratique c'est pareil. En pratique c'est pas vrai.

tmux + systemd
Posté par colmi le 27 mars 2019 à 15:20. Évalué à 2.

Pour ce qui est du sujet de la survie du process tmux à la session utilisateur, le sujet a déjà déclenché des réactions vives, [ici par exemple](#).

Du coup, sauf erreur de ma part, je suppose qu'il faut voir le paramétrage pour ta version de debian, pour savoir si c'est une version plus vieille de systemd avec des paramètres par défaut avec un comportement plus traditionnel, ou sinon voir si debian a surchargé les paramètres de systemd.

[^] # Re: tmux + systemd
Posté par nud le 27 mars 2019 à 15:36. Évalué à 2.

C'est Debian Buster, qui fournit systemd 241. Il me semble que c'est la dernière. Mais le comportement est le même avec Stretch et systemd 232.

Comme je le dis dans le journal, je n'ai pas (encore) creusé, même si le comportement décrit dans le journal que tu cites est celui auquel je me serais attendu.

[^] # Re: tmux + systemd
Posté par Shuba le 27 mars 2019 à 15:39. Évalué à 4.

Dans mes souvenirs debian avait choisi de mettre par défaut une option qui permettait de ne pas tuer les screens et autres tmux des utilisateurs, pour ne pas perturber la communauté habituée à fonctionner avec screen.

[^] # Re: **tmux + systemd**
Posté par nud le 27 mars 2019 à 18:33. Évalué à 5.

J'ai un peu fouillé, et effectivement, il y a un patch dans Debian qui change la valeur par défaut du paramètre `KillUserProcesses` dans `systemd-logind` (modifiable dans `/etc/systemd/logind.conf`). Par défaut c'est à yes et ils le mettent à no.



Après, je ne comprends pas trop pourquoi ils ont changé la valeur par défaut dans le démon plutôt que juste mettre la valeur à no dans le fichier de config (ça m'aurait paru plus simple), mais qui suis-je pour juger?

Correction / différence

Posté par --[silmaril]=- (site web personnel) le 27 mars 2019 à 17:55. Évalué à 4. Dernière modification le 27 mars 2019 à 17:56.

Sympa, je le savais possible mais n'avais pas pris le temps de regarder le sujet.

Chez moi (debian/stretch) il faut préciser `--user` pour chaque commande par contre:

```
$ systemctl start tmux-irc
```

```
$ systemctl start --user tmux-irc
```

et

```
$ systemctl status --user tmux-irc
```

pour avoir l'état

[^] # Re: **Correction / différence**

Posté par nud le 27 mars 2019 à 18:28. Évalué à 2.

Rha, effectivement, c'est une typo. Malheureusement on ne sait pas éditer le journal :(



Je me suis d'ailleurs fait la réflexion, en l'utilisant, que c'était plutôt chiant.

[^] # Re: **Correction / différence**

Posté par --[silmaril]=- (site web personnel) le 28 mars 2019 à 09:31. Évalué à 1.

solution: se faire un alias en mode utilisateur ?

Pour que le `tmux` soit lancé au boot j'ai du rajouter ça aussi:

```
$ systemctl enable --user tmux-irc
```

[^] # Re: **Correction / différence**

Posté par raph[(site web personnel) le 28 mars 2019 à 10:09. Évalué à 1.

Malheureusement on ne sait pas éditer le journal :(

Coucou la Belgique ? :-)

[^] # Re: **Correction / différence**

Posté par Benoit Sibaud (site web personnel) le 27 mars 2019 à 18:37. Évalué à 4.

Corrigé, merci.

Pourquoi tmux ?

Posté par NiKaro (site web personnel) le 28 mars 2019 à 06:35. Évalué à 1.

Pourquoi lancer `irssi` dans un `tmux` lui-même lancé par `systemd` ? Au lieu de lancer `irssi` directement avec `systemd` ?

[^] # Re: **Pourquoi tmux ?**

Posté par NiKaro (site web personnel) le 28 mars 2019 à 06:36. Évalué à 3.

Ah je n'ai rien dit, je viens de voir que `irssi` est client IRC, je pensais qu'il s'agissait d'un serveur.

Du coup ça a du sens.

Suivre le flux des commentaires

Note : les commentaires appartiennent à celles et ceux qui les ont postés. Nous n'en sommes pas responsables.

Revenir en haut de page

Derniers commentaires

- Re: Oxymore ?
- Re: karma et modér...
- Re: on dit FDN
- Re: Description de l'...
- Re: Description de l'...
- Re: Envisageable ou ...
- Re: Déprimant
- Re: imo/ama
- Re: Envisageable ou ...
- Belle pédagogie
- Re: Question sur la ...
- Re: Description de l'...

Étiquettes (tags) populaires

- [intelligence_artificielle](#)
- [grands_modèles_de...](#)
- [merdification](#)
- [hugging_face](#)
- [états-unis](#)
- [capitalisme_de_surv...](#)
- [administration_fran...](#)
- [jihad_butlérien](#)
- [capitalisme](#)
- [startup_nation](#)
- [bigtech](#)
- [slop](#)

Sites amis

- [Agenda du Libre](#)
- [April](#)
- [Éditions D-BookeR](#)
- [Éditions Diamond](#)
- [Éditions Eyrolles](#)
- [Éditions ENI](#)
- [En Vente Libre](#)
- [Framasoft](#)
- [La Quadrature du Net](#)
- [Lea-Linux](#)
- [Open Source Initiative](#)
- [Imprimerie Grafik Plus](#)

À propos de LinuxFr.org

- [Mentions légales](#)
- [Faire un don](#)
- [L'équipe de LinuxFr...](#)
- [Informations sur le s...](#)
- [Aide / Foire aux que...](#)
- [Suivi des suggestion...](#)
- [Wiki du site](#)
- [Règles de modération](#)
- [Statistiques](#)
- [API pour le développ...](#)
- [Code source du site](#)
- [Plan du site](#)