

Multiplexage de terminaux

Attribution : <http://creativecommons.org/licenses/by/4.0/fr/>

Table des matières

1. Objectifs	3
2. Introduction	4
3. Premiers Pas	5
3.1. Ouverture et Fermeture de sessions	5
3.2. Attacher et Détacher.....	6
3.2.1. Plusieurs connexions	6
3.3. Multi-Terminal	7
3.4. Zoomer	7
3.5. Fenêtres.....	7
3.6. Sessions	7
3.7. Fermeture.....	7
4. Conclusion	8

1. Objectifs

Installer et configurer tmux

Apprendre à utiliser tmux

2. Introduction

Lorsque l'on administre un serveur à distance, il est régulièrement utile d'ouvrir plusieurs terminaux. Plutôt que d'ouvrir plusieurs connexions en SSH dans plusieurs fenêtres différentes, il est possible de *multiplexer* sa connexion, c'est-à-dire de faire passer plusieurs terminaux dans une seule connexion. Cela permet en plus d'être plusieurs sur un seul terminal ou de laisser un terminal s'exécuter en tâche de fond.

Pour cela, plusieurs outils existent, comme *tmux* ou *screen*. Dans ce module, nous apprendrons à utiliser *tmux*.

3. Premiers Pas

Installation

Tmux s'installe sur la machine où il sera exécuté. Pour cela, il faut au préalable l'installer avec la commande suivante :

```
1 sudo apt install tmux
```

Configuration

Dans tmux, tout se fait par des raccourcis clavier. Ceux par défaut peuvent être difficiles à retenir. Notre première action sera donc de les remplacer par des plus simples. Dans un terminal, éditer le fichier de configuration

```
1 nano ~/.tmux.conf
```

et y ajouter le contenu suivant

```
1 # Activate color support
2 set -g default-terminal "screen-256color"
3
4 # Activate mouse support
5 set -g mouse on
6
7 # Add more history
8 set -g history-limit 100000
9
10 # Setup the status bar
11 set -g status-right ' #{?client_prefix,P,} "#{host}"'
12 set -g status-left-length 30
13 set -g status-right-length 30
14
15 # Change prefix key (Control+Space)
16 unbind C-b
17 set -g prefix C-Space
18
19 # Change key to reload (r)
20 unbind r
21 bind r source-file ~/.tmux.conf \; display "Reloaded ~/.tmux.conf"
22
23 # Add key to kill current session
24 unbind q
25 bind q kill-session
26
27 # Change key for splitting panes (h & v)
28 ## v and h are not binded by default, but we never know in the next versions...
29 unbind v
30 unbind h
31 unbind % # Split vertically
32 unbind '"' # Split horizontally
33 bind v split-window -h -c "#{pane_current_path}"
34 bind h split-window -v -c "#{pane_current_path}"
```

Nous verrons par la suite ce que font toutes ces lignes.

3.1. Ouverture et Fermeture de sessions

Ouverture

Pour ouvrir une session tmux, il suffit d'exécuter la commande `tmux` dans un terminal. Cela ouvrira un terminal dans le terminal, que l'on pourra reconnaître par la présence d'une barre verte en bas.

Ouverture en SSH

[+ Complément](#)

Tmux est plus souvent utilisé au sein d'une connexion SSH. Il est donc possible de se connecter en SSH directement à l'intérieur d'une session en tapant :

```
1 ssh login@host -t tmux
```

Fermeture

Une session tmux peut se fermer de plusieurs manières :

- En quittant le dernier terminal dedans en faisant un `exit` ou avec le raccourci clavier `control d` (`Ctrl-D`)

En demandant explicitement à tmux de terminer tous les terminaux de la sessions avec le raccourci `Ctrl-Space` puis `q`

Préfixe

[🔗 Fondamental](#)

Le raccourci que l'on vient d'effectuer (`Ctrl-Space` avant `q`) est ce que l'on appelle en tmux le *préfixe*. Il porte ce nom car toutes les commandes tmux commencent par un préfixe afin de les différencier des raccourcis classiques. On les dénotera `<prefix><key>` par la suite, avec `key` le raccourci rentré après (`q` dans notre dernier exemple).

3.2. Attacher et Détacher

On peut se *détacher* d'un terminal, c'est-à-dire le laisser tourner en arrière plan, en effectuant la commande suivante :

```
1 <prefix>d
```

avec `d` pour detach. Ensuite, on peut se rattacher à la dernière session ouverte avec la commande

```
1 tmux attach
2 # ou pour aller plus vite
3 tmux a
```

On peut donc se rattacher rapidement à une précédente session tmux depuis SSH avec la commande

```
1 ssh login@host -t "tmux a"
```

3.2.1. Plusieurs connexions

[👁 Exemple](#)

Plusieurs personnes peuvent également se connecter en même temps sur une seule connexion. Pour vérifier cela, connectez vous sur le VPS d'un camarade dans une session commune.

3.3. Multi-Terminal

Passons désormais au véritable pouvoir de tmux : *splitter* un terminal en plusieurs. Pour cela, on utilisera la commande

```
1 <prefix>v
```

pour séparer **v**erticalement et

```
1 <prefix>h
```

pour séparer **h**orizontalement.

Pour passer d'un terminal à l'autre, on peut utiliser la souris ou

```
1 <prefix>LeftArrow
2 <prefix>RightArrow
3 <prefix>UpArrow
4 <prefix>DownArrow
```

3.4. Zoomer

Il est parfois utile de mettre un seul terminal en "plein-écran" sans pour autant fermer les autres.

Pour cela, on peut zoomer sur le terminal courant avec le raccourci suivant :

```
1 <prefix>z
```

pour entrer et sortir du zoom.

3.5. Fenêtres

En plus d'avoir plusieurs terminaux en une vue, il est possible d'avoir plusieurs vues, appelées *fenêtres*. Une fenêtre contient plusieurs terminaux et il peut y avoir plusieurs fenêtres. Cela peut être utile quand vous auriez envie d'ouvrir un nouveau terminal pour avoir une vue dégagée par exemple. Pour créer une fenêtre, rentrer le raccourci suivant :

```
1 <prefix>c
```

ce qui ajoutera une sorte d'onglet en bas à gauche de la barre verte et nous placera dans la fenêtre nouvellement créée. Pour revenir à la fenêtre précédente, on peut soit cliquer sur son onglet à la souris, soit entrer le raccourci suivant :

```
1 <prefix>n
```

pour "next" (les fenêtres étant cycliques, passer à l'après-dernier revient à passer au premier).

3.6. Sessions

En fait il existe encore un niveau de multiplexage au dessus des fenêtres : les sessions. Nous ne les aborderons cependant pas car peu adaptées à notre usage. Pour notre culture et curiosité, notons donc qu'il est possible d'avoir plusieurs sessions en parallèle, comme par exemple un pour monitorer les services, une pour débayer les logs et une pour éditer les fichiers usuels.

3.7. Fermeture

Comme on l'a vu au début de ce module, il est possible de quitter tmux avec

```
1 <prefix>q
```

ce qui aura pour effet de forcer la fermeture de tous les terminaux de toutes les fenêtres de cette session.

4. Conclusion

Dans ce module nous avons découvert tmux pour faciliter le parcours de multiples console à distance.

Les terminaux ysont regroupés selon la hiérarchie suivante :

- Session
 - Fenêtre
 - Terminal

On parle avec tmux soit par CLI soit par raccourcis claviers. En particulier, le raccourci `<prefix>` (ici `Ctrl-Space`) préfixe tous les raccourcis propres a tmux :

```
1 tmux # Commence la session
2 tmux a # Attache une session démarrée
3 <prefix>d # Détache d'une session
4 <prefix>q # Quitte la session
5 <prefix>v # Vertical split
6 <prefix>h # Horizontal split
7 <prefix>Up/Down/Left/Right # Change de terminal (ou souris)
8 <prefix>z # Zoome sur un terminal
9 <prefix>c # Crée une nouvelle fenêtre
10 <prefix>n # Next fenêtre
```