

TMUX INSTALLATION/USE CASES

Wednesday, June 15, 2022 9:13 PM

TMUX INSTALLATION/USE CASES AND

So what is tmux?

Basically tmux is a terminal multiplexer, which allows a user to create multiple session and run different servers and close them **without actually killing** the session. So the big question is can't we just use background job(ctrl + z) to run the job in background and open multiple tabs to do the various jobs? The answer is yes you can do both in order to replace tmux, but the things you can do with tmux is limitless.

For example let's say someone ask you to debug or develop a different node, without tmux, you would probably close all the terminals and work on the new one. What if instead of killing those processes, you can "put them aside" (while they're still running), run whatever processes you need to complete your node task, then return to your original task as if you never left? That would be nice, wouldn't it? Tmux lets you do just that!

To be fair, you could argue that you could just open more tabs for node tasks instead of killing the previous rails processes. Although that would work, this could cause tab chaos. At some point, you'll start asking, "which of these 6 tabs did I just run the test?" or "which of these two Vim tabs was my config file?". When you start mixing tabs from completely unrelated tasks without clear boundaries, you can't tell which tab is for what. This also causes a grinding halt to your workflow.

So let's get started ?

As we have discussed tmux helps us to run multiple process, without having to kill them. So how does it do it? The answer is **SESSIONS**. Basically a session is nothing but a time frame for communication between devices, system etc. So does this mean, after I restart or reboot my computer will I be able to connect to those session? the answer is **no** you can't. But there is work around way which will be covered in this documentation.

SESSIONS

1. Creating a session

By using command "tmux", it creates a session with a session name "0" and the name gradually increases as well. In the screen shot below, 0:nagios-host@localhost:~*,

```
[nagios-host@localhost ~]$ tmux
```



```
nagios-host@localhost~  
[nagios-host@localhost ~]$  
  
[0] 0:nagios-host@localhost:~* "localhost.localdomain" 19:10 10-Jun-22
```

2. Attaching and Detaching session

```
[menz@localhost ~]$ tmux detach
```

Tmux detach is used to detach from a session or ctrl + D can also be used to detach from the session

```
[menz@localhost ~]$ tmux ls  
0: 1 windows (created Sat Jun 11 08:57:58 2022) [120x26]  
[menz@localhost ~]$ tmux attach -t 0
```

Tmux ls lists all the running sessions, and **tmux attach -t 0** attaches to the 0 session

3. Creating a session using a name

```
[menz@localhost ~]$ tmux new -s milan.yco
```

This creates a new session with the name "milan.yco"

4. Killing a session

```
[menz@localhost ~]$ tmux ls  
0: 1 windows (created Sat Jun 11 08:57:58 2022) [120x26]  
milan.yco: 1 windows (created Sat Jun 11 09:03:58 2022) [120x26]  
[menz@localhost ~]$ tmux kill-session -t 0  
[menz@localhost ~]$
```

WINDOWS

Tmux windows are subsets of tmux sessions. When you're creating a new tmux session (`tmux new -s MY_SESSION`), what you're seeing is technically a tmux window. Each terminal activity happens inside a window. So technically, a tmux session is a container for tmux windows.

You can have as many windows as you want in a session. Since a window contains a terminal on its own, you can have a window dedicated for running your application server, another window for logging, and another window for text-editing.

1. Creating a new window

You can only create a new window from inside a tmux session

```
tmux new-window -n fried-rice
```

2. Switching Windows

For switching to next window, prefix (ctrl) + n and to switch over to previous window Prefix(ctrl)+p

```
Prefix + n
```

```
Prefix + p
```

3. Listing out all the windows

```
Prefix + w
```

4. Killing windows

To kill the current window

```
tmux kill-window
```

To kill a different window

```
tmux kill-window -t ramen
```

Tmux Servers

When “tmux” command was used, it created a default server, where all the sessions, windows and panes are connected to the default server, by using “tmux” command not only it created tmux session but it also created new default server instance.

1. Creating a new named server

To create a new server with a name, by creating a new server it creates a default session as well

```
[menz@localhost ~]$ tmux -L YCO
```

```
[menz@localhost ~]$ tmux -L YCO ls
0: 1 windows (created Wed Jun 15 02:13:49 2022) [124x29]
[menz@localhost ~]$
```

2. Creating a new session in named server

```
[menz@localhost ~]$ tmux -L YCO new -s milan-test
```

```
[menz@localhost ~]$ tmux -L YCO ls
0: 1 windows (created Wed Jun 15 02:13:49 2022) [124x29]
milan-test: 1 windows (created Wed Jun 15 02:20:09 2022) [124x29]
```

3. Killing a named server

```
[menz@localhost ~]$ tmux -L YCO kill-server
[menz@localhost ~]$
```

Sharing Tmux server with a user

1. Starting a new server

```
tmux new-session
```

2. Using Sockets for different users to connect:

```
To Start a new session
tmux -S /tmp/socket
```

```
Change its permission for other users to access
chmod 777 /tmp/socket
```

```
for other users to attach to the session
tmux -S /tmp/socket attach
```