Professional Technical Notes

Windows and Linux troubleshooting and configuration notes

Comprehensive Tmux Tutorial for Beginners with a Cheat Sheet

July 22, 2020 How-To, Informational, Linux

Changing the color scheme, information in the status bar, installing themes via the plugin manager, enabling copy and paste via the mouse and to the clipboard, and mimicking the screen prefix in the config file are all covered in this tmux quick start guide.

However, in addition to providing the essential configuration options, this tutorial also provides a beginner-friendly introduction to the layout of tmux screens, windows, panes, and how each of these components works together.

This quick start guide is designed to coach a beginner from the installation and configuration of tmux, to installing custom powerline themes via a plugin manager by the time they reach the bottom of the page.

Why use Tmux?

Tmux stands for **Terminal Multiplexer**. Essentially tmux allows a user to create multiple terminal sessions within one actual terminal window or tab. A (probably bad) analogy could be if you have 10 different tabs open in Chrome or Firefox, and within each tab, you had 4 different web pages displayed.

For the actual definition, what better place to define tmux than the tmux man page

```
tmux is a terminal multiplexer: it enables a number of terminals tc
created, accessed, and controlled from a single screen.  tmux may b
detached from a screen and continue running in the background, then
later reattached.
```

Additional information can be obtained from the tmux github page.

Three Main Benefits of Tmux

The three main benefits of tmux also highlight the three different components that make up a tmux server instance. While there are other benefits (quickly resizing panes, color customization, themes, customization, hotkeys, etc.), the three **MAIN** benefits are listed below, with each component highlighted.

1. Attach/detach from **SESSIONS**
2. Multiple **WINDOWS** within a session
3. Multiple **PANES** within a window

In order to understand exactly what these mean, lets take a closer look into what makes up a tmux Session, Window, and Pane.

Sessions, Windows, and Panes

TMUX SESSIONS. Sessions are the highest level object that make up the interface. A user can create multiple sessions, kill sessions, rename them, and more, all without ever actually having to enter *into* the session. Sessions also allow users to attach and detach from them with ease.

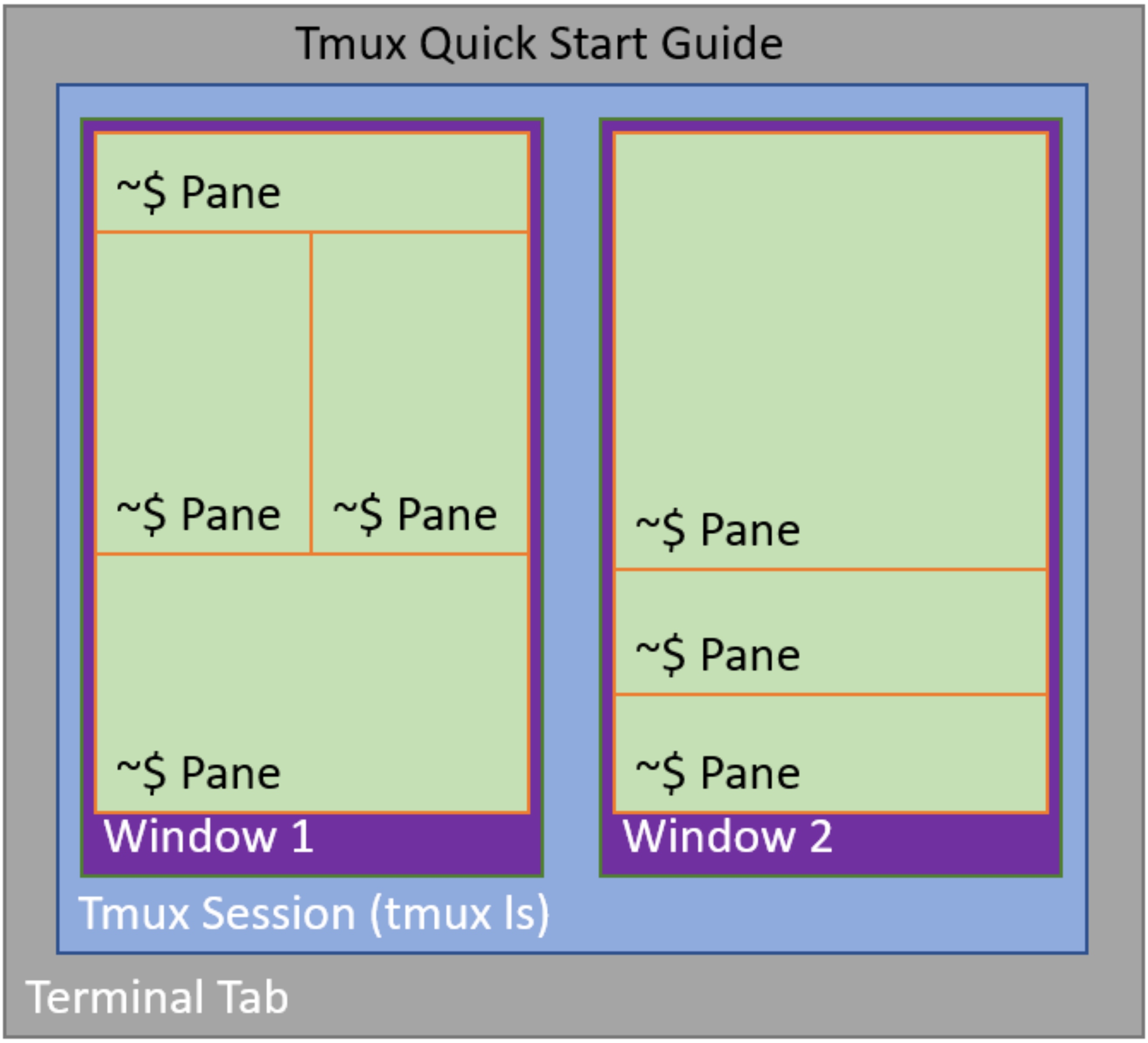
Sessions exemplify the power of tmux due to the ability to detach and attach. A single session could have 5 windows, with a total of 20 panes spread between them, with each pane already cd’ed into the proper directory, tailing the correct log, or executing the watch script, with a workspace pane also available. Need to do the same thing, but on another system? Just detach, and switch to the appropriate session with the same configuration. As you can imagine, this functionality can save a lot of time.

TMUX WINDOWS. Windows are the equivalent to tabs within your web browser, tabs within excel, or terminal tabs. Users can rename them and switch to them with a simple hotkey. They are displayed within the tmux status bar.

The benefit of tmux windows is obvious as even as you read this you probably have a minimum of 13 other tabs open. Example scenario: TAB1 is your troubleshooting tab with logs being tailed, monitoring cpu utilization, etc. TAB2 is where you are modifying your script or application configuration. Make a change in TAB2, switch to TAB1, and watch the stats/logs.

TMUX PANES. Panes are essentially another terminal instance, within the window, within the session. Users can shape panes horizontally and vertically with hotkeys, arranging them to optimize their screen space for the current task.

Tmux panes are the largest contributor to increased productivity. For example, you are testing a config change and want to see its impact on the system. You need to compare against a known good system. Within a span of 5 seconds you could open up two more panes, ssh to the good system, view the config in one pane, and look at the logs in another. All the while still viewing your test system logs and config on the other two panes you had open…. In the same Tmux Window…. In the Tmux Session… In the same Terminal Tab. Now that's increased productivity.



A Tmux Session with two tmux windows with multiple tmux panes within each

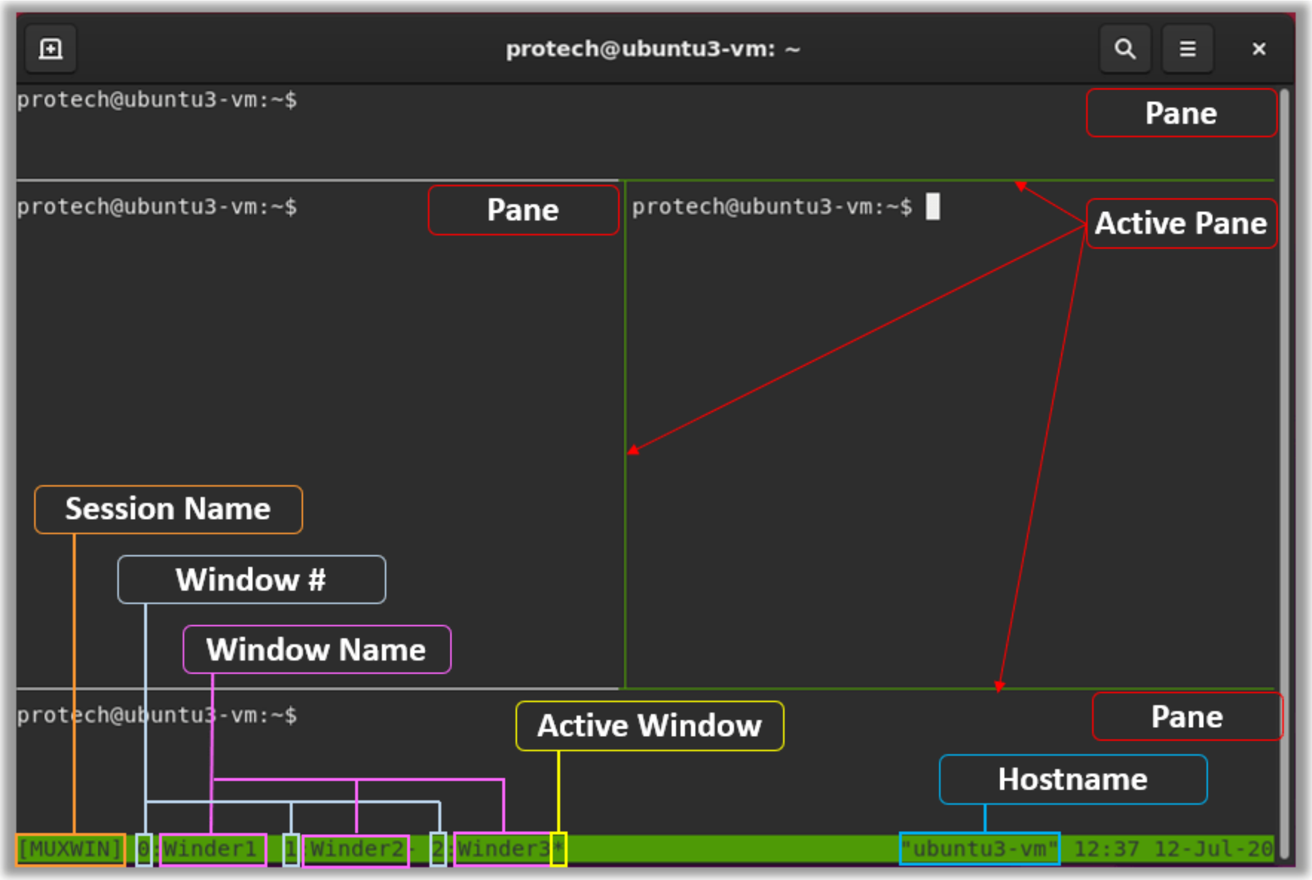
Tmux Status Bar

The tmux status bar deserves it's own section because understanding where you are is essential to using tmux correctly and not accidentally rm -rf'ing a directory on the wrong host. The status bar shows you all you need to know, and its important to know whats going on down there before we start muxing.

The Tmux status bar quickly shows you the following:

- **Session name**
- **Window number**
- **Window name**
- **Active Window**
- **Hostname**

There are a lot of configuration options available for the tmux status bar, but with a standard install, the default color of the status bar is green. The image below is an example of what the panes in Window1 of the image above could look like when configured the same way.



```
protech@ubuntu3-vm:~$

protech@ubuntu3-vm:~$                    protech@ubuntu3-vm:~$

protech@ubuntu3-vm:~$

[MUXWIN] 0:Winder1- 1:Winder2- 2:Winder3*                "ubuntu3-vm" 12:37 12-Jul-20
```

Tmux Status bar legend

With a basic understanding of what tmux is and how sessions, windows, and panes work together, lets quickly install it and start messing around.

Tmux Installation

Typical Install

- Installing tmux is quite simple:
    - **Ubuntu**: `sudo apt-get install tmux`
    - **Mac**: `brew install tmux`
    - **CentOS/RHEL**: `sudo yum install tmux`

Local Download

- If you'd like to have a local installation, you can download tmux, navigate to their github page and download the latest stable .tar.gz. Once downloaded, follow the configure/make/make install procedures.
    - Note: A local tmux download/install require ncurses and libevent 2.x. See the tmux README for additional information.

The **Tmux Prefix**

Before we start manipulating tmux sessions, windows, and panes, it is important to understand what the prefix is.

The tmux prefix is a combination of keys you press prior to pressing ANOTHER (command) key which will actually do the action you desire. It's the equivalent of pressing **`ctrl + alt`** before pressing **`ctrl + alt + del`**. In this example the **`ctrl + alt`** is the prefix, and the **`del`** is the command.

The **default prefix** for tmux is **`ctrl + b`**. For example:
Prefix: `ctrl + b`
Command: d
As we will learn further down in the article, this will detach from the current session using the command key **d**.

We learn how to change the prefix key within the tmux config section.

Tmux Sessions

There are a couple of methods to create, attach, detach, switch, and kill tmux sessions. We will practice creating one using the most basic method, and then with a bit more detail.

Creating a Simple Session

Starting with the very basics, lets simply start and attach to a new session, then exit it.

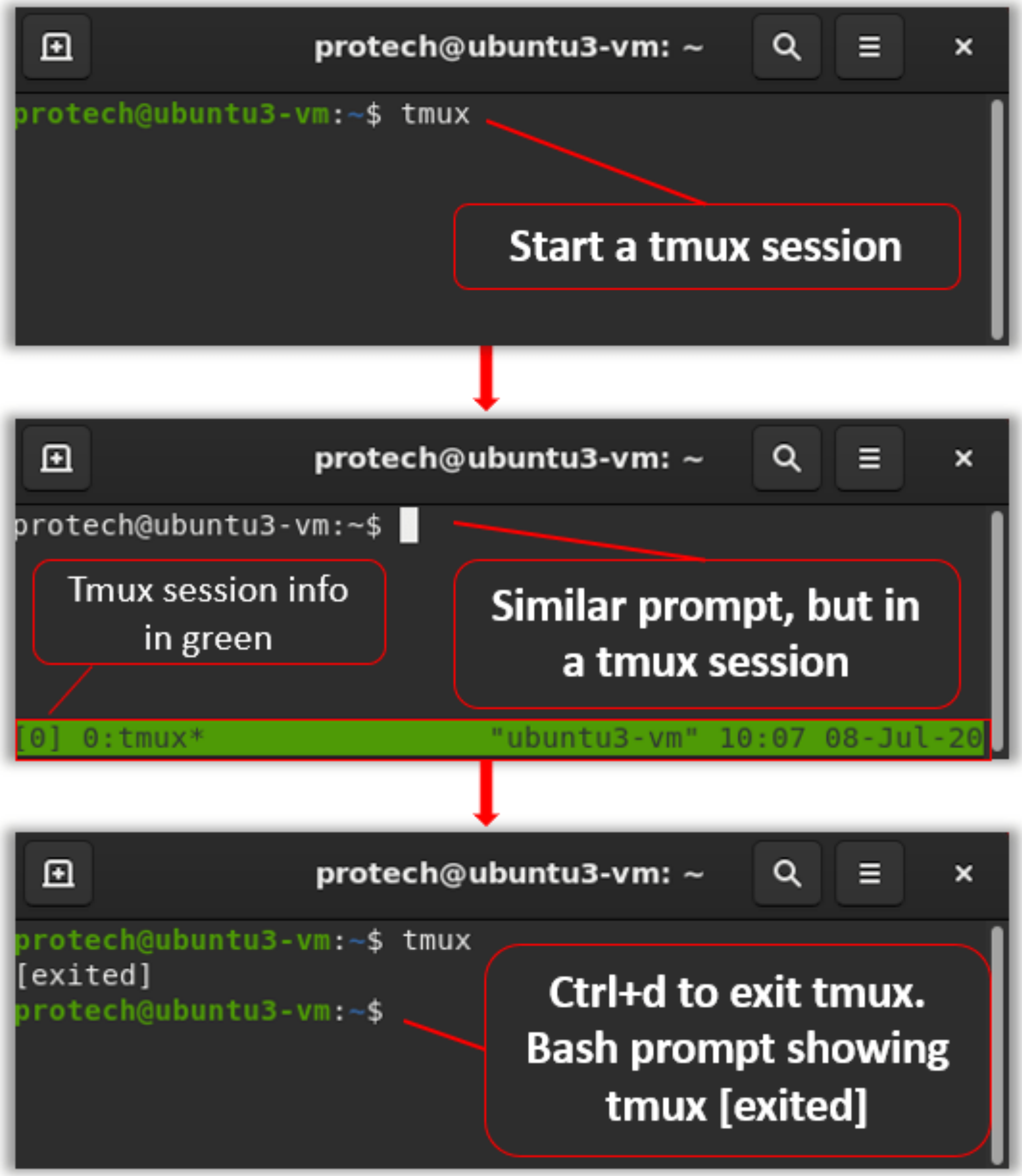
To **start and attach** to a new session, simply run the command:

```
tmux
```

Inside your first session, notice the view is almost identical to a normal terminal window. The tmux status bar at the bottom shows information regarding this session. Before we go further, lets **exit** this session (**not detach**).

Exit the session by pressing (When we exit the session, the server will also exit)

```
ctrl + d
```



```
protech@ubuntu3-vm:~$ tmux
```

Start a tmux session

```
protech@ubuntu3-vm:~$ █

[0] 0:tmux*                "ubuntu3-vm" 10:07 08-Jul-20
```

Tmux session info in green

Similar prompt, but in a tmux session

```
protech@ubuntu3-vm:~$ tmux
[exited]
protech@ubuntu3-vm:~$
```

Ctrl+d to exit tmux. Bash prompt showing tmux [exited]

Start a new tmux server session. Ctrl+d to exit it. Return to bash prompt with tmux server [exited] message.

Now, we are going to do the same thing, except this time we are going to **detach** from the session, leaving the tmux server, and therefor this session, still running. After detaching, we can attach to it again.

Start and attach to a new session

```
tmux
```

Now, **detach** from the session

```
ctrl + b d
```

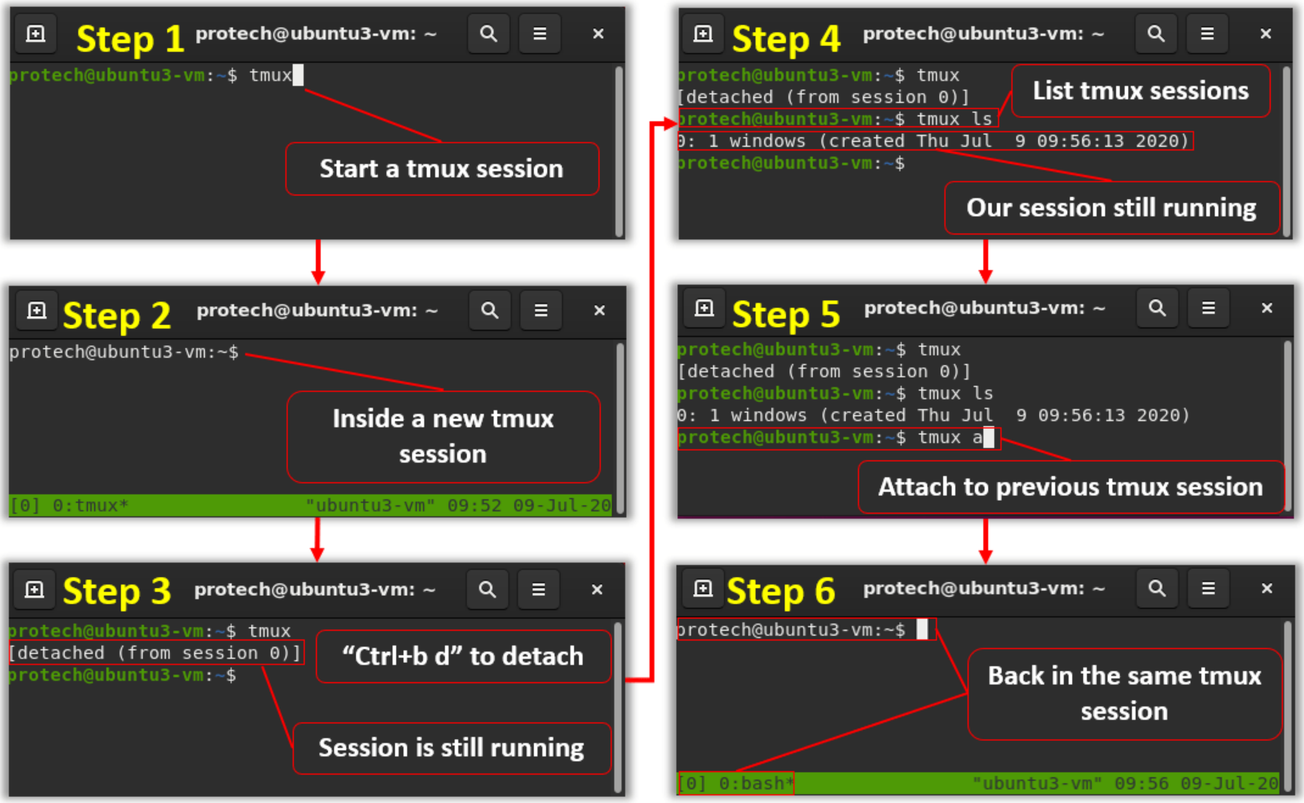
Next, let's **list running sessions**. We should see our previously created session

```
tmux ls
```

Now, we can **reattach** to the session in a variety of ways. One of the ways you can reattach is by attaching to your most previously used session. Run any of the following commands to **attach to your most previously used session**

```
tmux a
tmux at
tmux attach
tmux attach-session
```

Boom! You are back into the same session you created earlier. You can now exit with **`ctrl + d`** or detach with **`ctrl + b d`**.



Step 1 protech@ubuntu3-vm: ~

```
protech@ubuntu3-vm:~$ tmux
```

Start a tmux session

Step 2 protech@ubuntu3-vm: ~

```
protech@ubuntu3-vm:~$

[0] 0:tmux*                     "ubuntu3-vm" 09:52 09-Jul-20
```

Inside a new tmux session

Step 3 protech@ubuntu3-vm: ~

```
protech@ubuntu3-vm:~$ tmux
[detached (from session 0)]
protech@ubuntu3-vm:~$
```

“Ctrl+b d” to detach

Session is still running

Step 4 protech@ubuntu3-vm: ~

```
protech@ubuntu3-vm:~$ tmux
[detached (from session 0)]
protech@ubuntu3-vm:~$ tmux ls
0: 1 windows (created Thu Jul  9 09:56:13 2020)
protech@ubuntu3-vm:~$
```

List tmux sessions

Our session still running

Step 5 protech@ubuntu3-vm: ~

```
protech@ubuntu3-vm:~$ tmux
[detached (from session 0)]
protech@ubuntu3-vm:~$ tmux ls
0: 1 windows (created Thu Jul  9 09:56:13 2020)
protech@ubuntu3-vm:~$ tmux a
```

Attach to previous tmux session

Step 6 protech@ubuntu3-vm: ~

```
protech@ubuntu3-vm:~$

[0] 0:bash*                     "ubuntu3-vm" 09:56 09-Jul-20
```

Back in the same tmux session

Create, detach, list, and reattach to a tmux session.

These are the absolute basics of creating, exiting, detaching, and attaching to a single session. Next, we are going to practice creating and managing tmux sessions with a bit more detail.

Session Management

Now that we’ve created, detached, listed, and attached back to a single session, let us explore creating some with session names, killing sessions, switching to others.

First, let’s **create three new sessions with a specified name without attaching.** To do this, we will use the **-d** option to stay detached, and the **-s** option to specify a name. **tmux new -d -s <YourSessionName>**.

```
tmux new -d -s 1tmux
tmux new -d -s 2tmux
tmux new -d -s 3tmux
```

List tmux sessions. You should see all three newly created sessions.

```
tmux ls
```

Now, for testing purposes, lets:

1. Kill 1tmux
2. Attach to 2tmux

3. Change the session name from 2tmux to 2TMUXXX (from inside the session)
4. Kill all remaining sessions aside from the one we are in (which will mean killing 3tmux).
5. List the remaining sessions. There should only be 2TMUXX

Kill 1tmux session (From the terminal, not inside a session)

```
tmux kill-ses -t 1tmux
```

Attach to session 2tmux (From the terminal, not inside a session)

```
tmux a -t 2tmux
```

Rename session 2tmux to 2TMUXXX (Inside a session)

```
ctrl+b $ 2TMUXX
```

Kill all remaining tmux sessions aside from this one (Inside a session)

```
tmux kill-ses -a
```

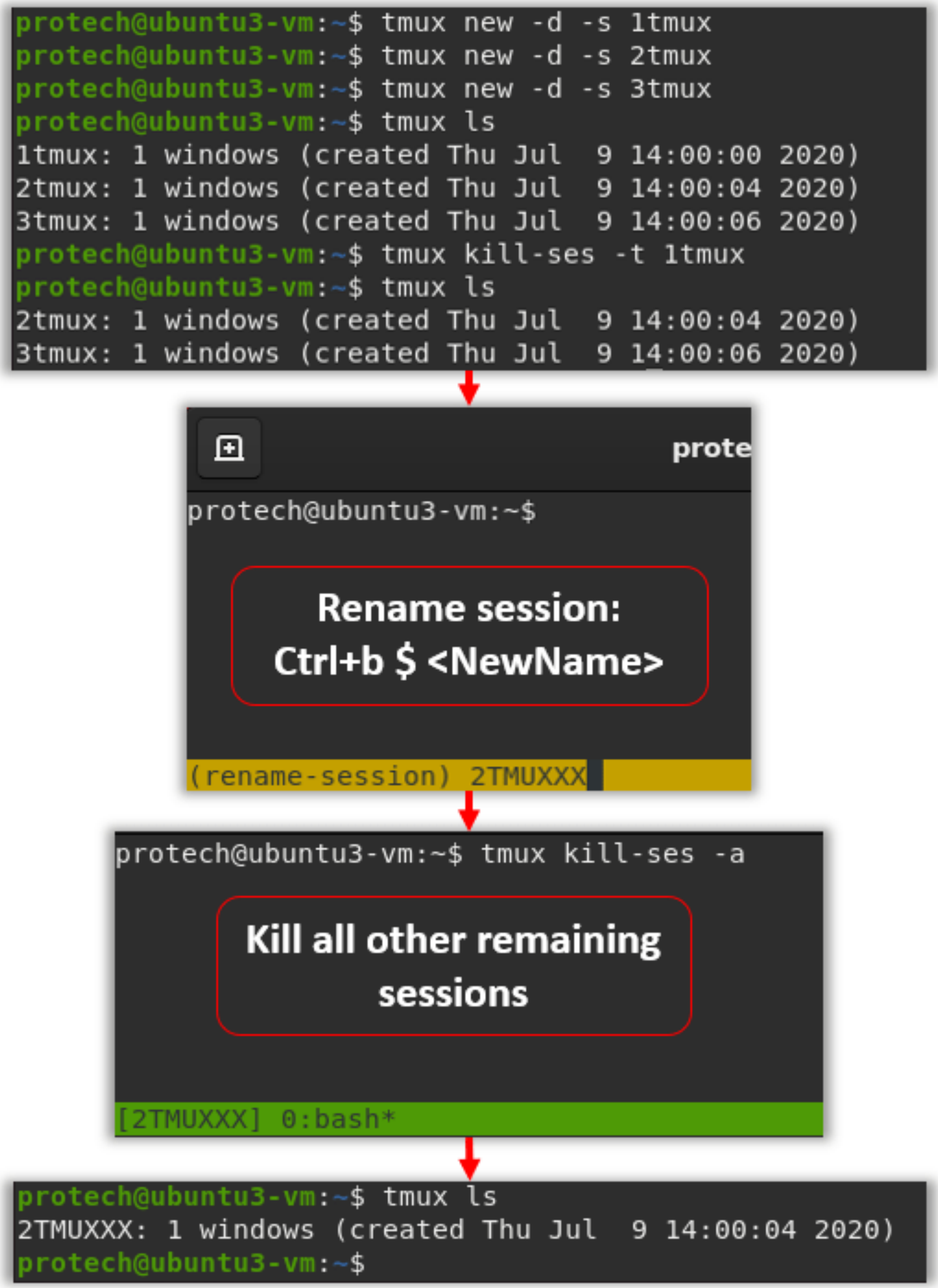
Detach (not exit) from this session (Inside a session)

```
ctrl + b d
tmux ls
```

List tmux sessions (From the terminal, not inside a session)

```
tmux ls
```

The only session remaining should be 2TMUXXX.



```
protech@ubuntu3-vm:~$ tmux new -d -s 1tmux
protech@ubuntu3-vm:~$ tmux new -d -s 2tmux
protech@ubuntu3-vm:~$ tmux new -d -s 3tmux
protech@ubuntu3-vm:~$ tmux ls
1tmux: 1 windows (created Thu Jul  9 14:00:00 2020)
2tmux: 1 windows (created Thu Jul  9 14:00:04 2020)
3tmux: 1 windows (created Thu Jul  9 14:00:06 2020)
protech@ubuntu3-vm:~$ tmux kill-ses -t 1tmux
protech@ubuntu3-vm:~$ tmux ls
2tmux: 1 windows (created Thu Jul  9 14:00:04 2020)
3tmux: 1 windows (created Thu Jul  9 14:00:06 2020)
```

```
protech@ubuntu3-vm:~$
```

Rename session:
Ctrl+b $ <NewName>

```
(rename-session) 2TMUXXX
```

```
protech@ubuntu3-vm:~$ tmux kill-ses -a
```

Kill all other remaining sessions

```
[2TMUXXX] 0:bash*
```

```
protech@ubuntu3-vm:~$ tmux ls
2TMUXXX: 1 windows (created Thu Jul  9 14:00:04 2020)
protech@ubuntu3-vm:~$
```

Creating, deleting, and renaming multiple tmux sessions

Switch sessions without exiting or detaching. Before continuing, you MUST know how to do this. This functionality will save you countless seconds every time you utilize it. Which, over the course of a few years, adds up to a few minutes!

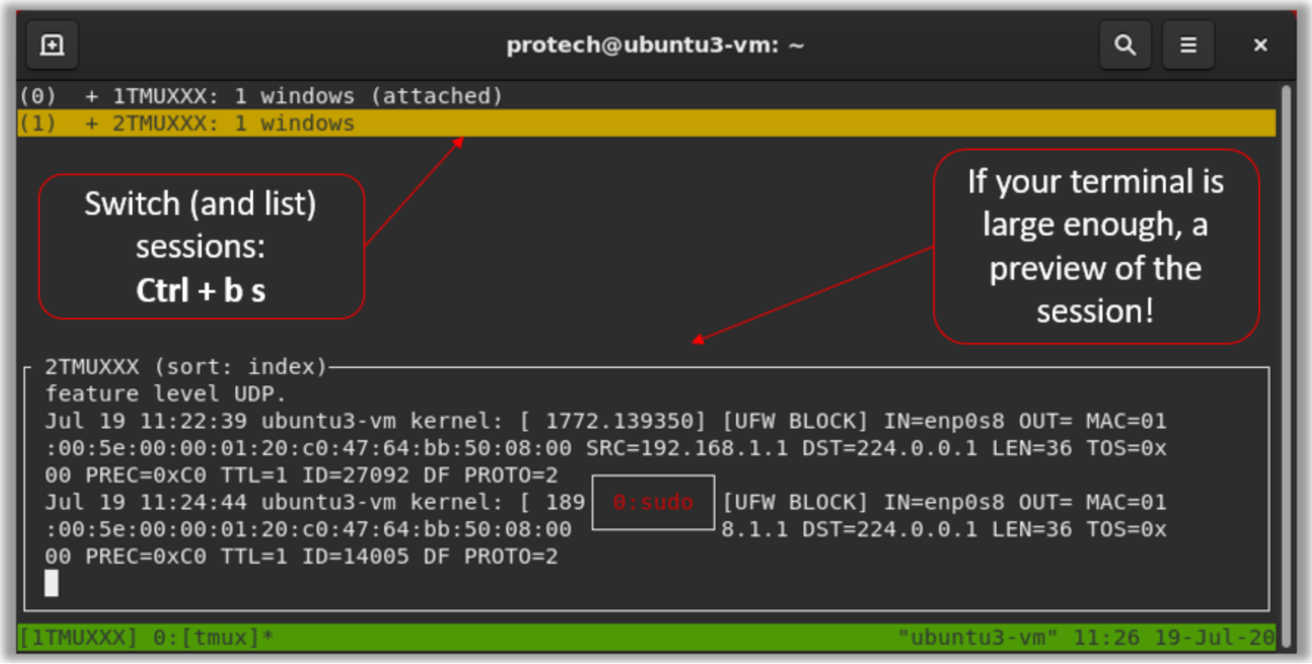
In addition to providing two features in one, invoking the ***tmux prefix + command key*** to switch sessions provides a preview of the other session. Great for verifying if you actually want to switch to a session before doing so.

Quick Tip! The switch session preview only occurs if your terminal window is large enough. If you don't see the preview, increase your terminal window size, and try to switch sessions again.

To **switch sessions** (and list them), enter your tmux prefix + s

```
Ctrl + b s
```

Use the arrow keys to navigate through the available sessions. Yep! That's right, this is the same as listing sessions. Press enter to complete the session switch. Use the preview to make an informed decision!



```
(0)  + 1TMUXXX: 1 windows (attached)
(1)  + 2TMUXXX: 1 windows
```

Switch (and list) sessions: **Ctrl + b s**

If your terminal is large enough, a preview of the session!

```
┌ 2TMUXXX (sort: index)─
 feature level UDP.
 Jul 19 11:22:39 ubuntu3-vm kernel: [ 1772.139350] [UFW BLOCK] IN=enp0s8 OUT= MAC=01
 :00:5e:00:00:01:20:c0:47:64:bb:50:08:00 SRC=192.168.1.1 DST=224.0.0.1 LEN=36 TOS=0x
 00 PREC=0xC0 TTL=1 ID=27092 DF PROTO=2
 Jul 19 11:24:44 ubuntu3-vm kernel: [ 189  0:sudo  [UFW BLOCK] IN=enp0s8 OUT= MAC=01
 :00:5e:00:00:01:20:c0:47:64:bb:50:08:00          8.1.1 DST=224.0.0.1 LEN=36 TOS=0x
 00 PREC=0xC0 TTL=1 ID=14005 DF PROTO=2
[1TMUXXX] 0:[tmux]*                                  "ubuntu3-vm" 11:26 19-Jul-20
```

List and switch tmux sessions with a preview of the session before switching.

Quick Tip! While inside a session, using the functionality to **switch** sessions is preferable to doing a **list** of sessions. For example, while working within tmux, you list the running sessions only to realize you want to switch to one now that you've seen what is available. You don't switch sessions every time, but sometimes you do end up switching to a session after listing them. Well, the hotkey for switching and listing sessions contains the benefit of listing sessions, with an additional benefit of giving you a preview of what is in the other sessions. This can save you an extra step while providing an additional feature. So basically, get in the habit of switching tmux sessions vs listing them!

Tmux Windows

Within each **session,** you can have multiple **windows**. Again, think of tmux windows like tabs in a browser or another terminal session.

For this example, we will do the following
1. Create and enter a new session

2. Rename the current window
3. Create two new windows within the session
4. Rename the two new windows

First, **create a new session** with the name MUXWIN

```
tmux new -s MUXWIN
```

In the MUXWIN tmux session, **rename the window** from its default (bash) to Winder1

```
ctrl + b ,
```

Create a new window

```
ctrl + b c
```

Rename this window to Winder2, then **create** one more window and rename it to Winder3

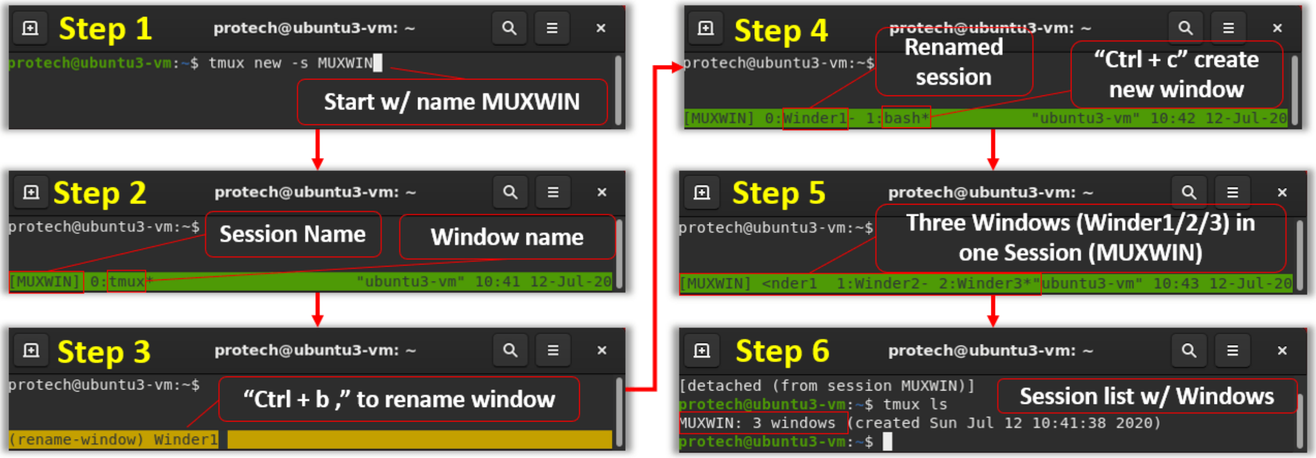
```
Ctrl + b , Winder2
Ctrl + b c
Ctrl + b , Winder3
```

Detach from the session (do not exit)

```
Ctrl + b d
```

List the sessions. Notice our session MUXWIN shows 3 Windows

```
tmux ls
```



Step 1

```
protech@ubuntu3-vm:~$ tmux new -s MUXWIN
```

Step 2

```
protech@ubuntu3-vm:~$

[MUXWIN] 0:tmux*                          "ubuntu3-vm" 10:41 12-Jul-20
```

Step 3

```
protech@ubuntu3-vm:~$

(rename-window) Winder1
```

Step 4

```
protech@ubuntu3-vm:~$

[MUXWIN] 0:Winder1- 1:bash*               "ubuntu3-vm" 10:42 12-Jul-20
```

Step 5

```
protech@ubuntu3-vm:~$

[MUXWIN] <nder1  1:Winder2- 2:Winder3*"ubuntu3-vm" 10:43 12-Jul-20
```

Step 6

```
[detached (from session MUXWIN)]
protech@ubuntu3-vm:~$ tmux ls
MUXWIN: 3 windows (created Sun Jul 12 10:41:38 2020)
protech@ubuntu3-vm:~$
```

protech@ubuntu3-vm: ~
Start w/ name MUXWIN
Session Name
Window name
“Ctrl + b ,” to rename window
Renamed session
“Ctrl + c” create new window
Three Windows (Winder1/2/3) in one Session (MUXWIN)
Session list w/ Windows

Create a new tmux session, and create and rename three windows within it.

Tmux Panes

Now that we can manage sessions and windows, let's start creating new panes ***within*** a tmux window.

First, **create a session** and attach to it. This one is called **1MUX**

```
tmux new -s 1MUX
```

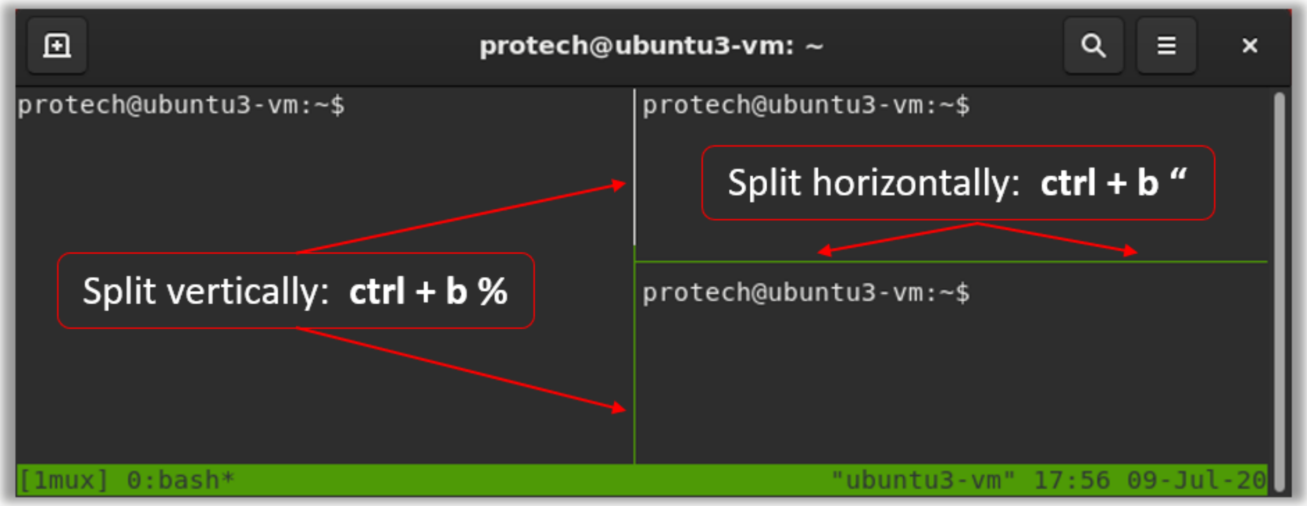
Now, **split the pane vertically**

```
ctrl + b %
```

Your cursor will now be on the right pane. Now, **split the pane horizontally**

```
ctrl + b "
```

Your cursor should now be on the bottom right of three panes within one tmux session!

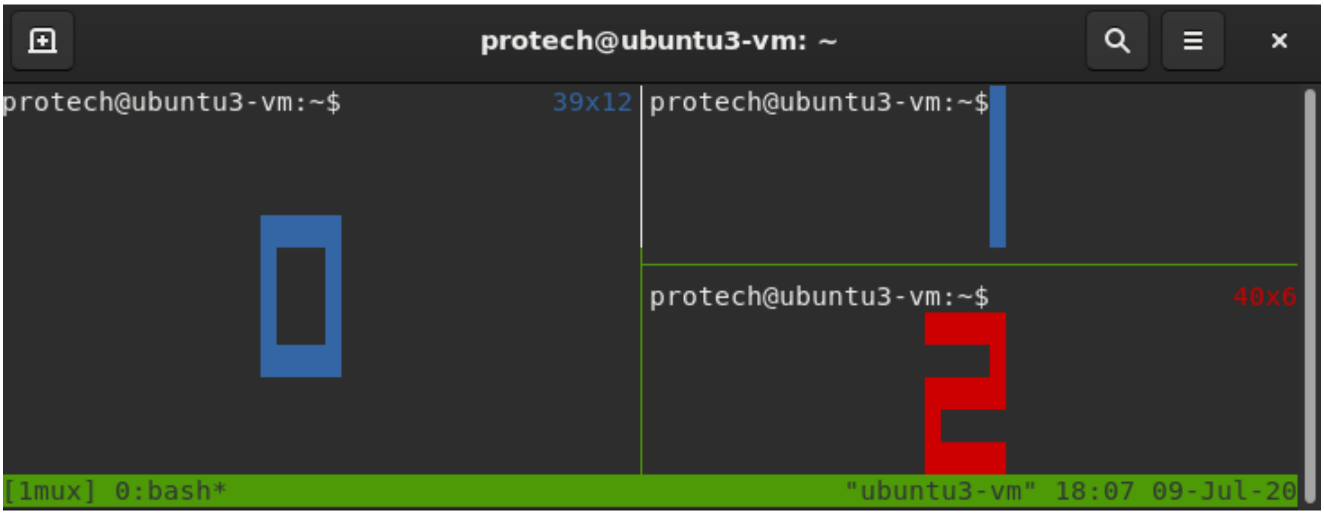


```
protech@ubuntu3-vm:~$          protech@ubuntu3-vm:~$
                               protech@ubuntu3-vm:~$
[1mux] 0:bash*                       "ubuntu3-vm" 17:56 09-Jul-20
```

Splitting tmux pane vertically and horizontally

Display the pane numbers by typing

```
ctrl + b q
```



```
protech@ubuntu3-vm:~$                     39x12
```

```
protech@ubuntu3-vm:~$
```

```
protech@ubuntu3-vm:~$                      40x6
```

```
[1mux] 0:bash*                    "ubuntu3-vm" 18:07 09-Jul-20
```

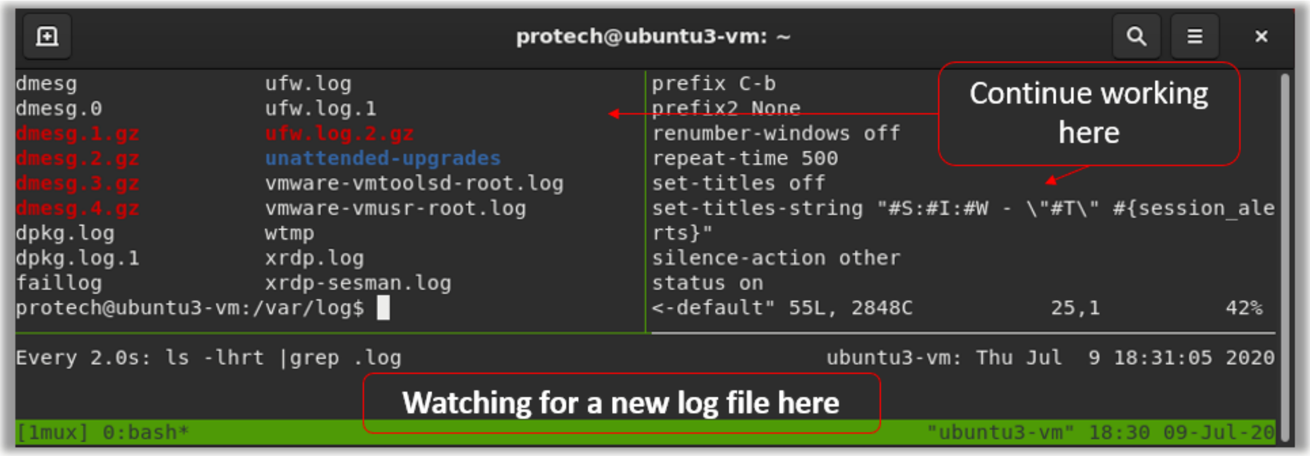
Ctrl+ b q to show tmux pane numbers

Knowing the numbers of the panes is useful for quickly switching to them. You can switch between the various panes by using the same key used to display the numbers, except this time add a number after the **q** to switch to that pane.

```
ctrl + b q <0..1..2..3..4..5..6..7..8..9>
```

While we only have three panes open right now, being able to quickly switch to a numbered pane can come in very handy when you have multiple sessions open with multiple panes within each session.

Quick tip: A very helpful way to use multiple panes within one session is to use the minimal amount of screen space required to obtain the information you need. For example, if you were running a watch command, and grepping for a new log file to drop into a directory, you could do that easily, while still using the other panes within the tmux session to continue working.



```
dmesg                 ufw.log
dmesg.0               ufw.log.1
dmesg.1.gz            ufw.log.2.gz
dmesg.2.gz            unattended-upgrades
dmesg.3.gz            vmware-vmtoolsd-root.log
dmesg.4.gz            vmware-vmusr-root.log
dpkg.log              wtmp
dpkg.log.1            xrdp.log
faillog               xrdp-sesman.log
protech@ubuntu3-vm:/var/log$
```

```
prefix C-b
prefix2 None
renumber-windows off
repeat-time 500
set-titles off
set-titles-string "#S:#I:#W - \"#T\" #{session_ale
rts}"
silence-action other
status on
<-default" 55L, 2848C          25,1          42%
```

Continue working here

```
Every 2.0s: ls -lhrt |grep .log              ubuntu3-vm: Thu Jul  9 18:31:05 2020
```

Watching for a new log file here

```
[1mux] 0:bash*                                      "ubuntu3-vm" 18:30 09-Jul-20
```

Using the smallest amount of space required to watch for a new file to drop, while still continuing to work within the other two panes within a single tmux session.

Tmux Copy / Paste & Clipboard

Being able to copy and paste is a basic human right, and no tmux tutorial would be complete without learning how to do it with the default config.

What's cool about this copy mode is that you can use while you are vim'ing a file, and also just at the command prompt.

What's not cool about this copy mode is that out of the box there is no hotkey to also copy it to our system's clipboard. So, keep in mind when you copy text here, it is only going to be relevant to the tmux buffer.

Copy & Paste – Default Settings

In a tmux window, enter tmux "**copy mode**"

```
ctrl + b [
```

You should see a little counter appear in the upper right with **`[0/0]`**. You are now in **copy mode**.

You still can't copy text yet. Move to where you want to START copying text.

Now, enter the "copy" mode of "copy mode".

```
ctrl + SPACEBAR
```

Now, using the arrow keys, move up/down. The highlighted text will be copied to your buffer.

Once you are done, exit copy mode by pressing

```
Alt + w
```

Everything that you just highlighted is now stored within the tmux buffer. Navigate to a new window or pane where you'd like to paste it. NOTE: You must still be in a session, you cannot paste this to a separate terminal window outside of tmux (for now).

Paste the text

```
ctrl + b ]
```

Thats it! Remember, you can use this within vim, nano, or your choice of editor, but also on the command prompt!

Copy Tmux Buffer to System Clipboard

Now let's modify our ***.tmux.conf*** so that we can copy the tmux buffer to our system's clipboard.

These settings are the most basic. As with anything, you can find more advanced versions, but for an easy way to configure copying the tmux buffer to the clipboard, these few lines work great.

For this configuration we use a program called **xclip**, so make sure you have it installed.

First, let's examine what each line of the config does, and then we'll piece it all together.

Allow copying with the **mouse**

```
set-option -g mouse on
```

Set tmux **copy mode vi**

```
setw -g mode-key vi
```

Set ***v*** to start copying, ***y*** to actually copy, and ***r*** to change to a rectangle copy

```
bind-key -T copy-mode-vi v send-keys -X begin-selection
bind-key -T copy-mode-vi y send-keys -X copy-selection
bind-key -T copy-mode-vi r send-keys -X rectangle-toggle
```

UNBIND the typical **"Enter"** key from Vi mode. We want to script this so that it will finish copying the text, but also send the text to the clipboard.

```
unbind -T copy-mode-vi Enter
```

Now, we need to **RE-BIND** the **copy-mode-vi "Enter"** to also include copying to the clipboard. A message will appear at the bottom of your tmux window once you press enter.

```
bind-key -T copy-mode-vi Enter send-keys -X copy-pipe-and-cancel 't
```

Finally, bind the **mouse pointer** so that you copy to the clipboard with it as well.

```
bind-key -T copy-mode-vi MouseDragEnd1Pane send-keys -X copy-pipe-a
```

When you are done, your ***.tmux.conf*** should look similar to this:

```
set-option -g mouse on
setw -g mode-key vi
set-option -s set-clipboard off
bind-key -T copy-mode-vi v send-keys -X begin-selection
```

```
bind-key -T copy-mode-vi y send-keys -X copy-selection
bind-key -T copy-mode-vi r send-keys -X rectangle-toggle
unbind -T copy-mode-vi Enter
bind-key -T copy-mode-vi Enter send-keys -X copy-pipe-and-cancel 't
bind-key -T copy-mode-vi MouseDragEnd1Pane send-keys -X copy-pipe-a
```

Now try it out!

1. Enter tmux copy mode with **`ctrl + b [`**
2. Select the text you would like to copy with **`v`**
3. Press **`Enter`** to copy the text to the tmux buffer AND the clipboard
4. Paste the text in another terminal window using standard paste commands (**`shift + insert`**)! BOOM! PASTED!

The **Tmux Config**

Wait, No Default Config?

Yep! With a fresh installation, there is no default tmux conf in /etc/ or somewhere else for us to examine. The man page for tmux says:

```
protech@ubuntu3-vm:~$ man tmux
... By default, tmux loads the system configuration
file from /etc/tmux.conf, if present, then looks for
a user configuration file at ~/.tmux.conf.
...
```

If you've freshly installed tmux, chances are there is no ***`tmux.conf`*** in **`/etc/`**, nor do you have a ***`~/.tmux.conf`*** config file.

While there is no tmux conf for us to examine after a fresh install, an example config located in `/usr/share/doc/tmux/`**`example_tmux.conf`**.

```
protech@ubuntu3-vm:~$ ll /usr/share/doc/tmux/
total 100
drwxr-xr-x    2 root root  4096 Jul  7 09:18 ./
drwxr-xr-x 1427 root root 69632 Jul  7 09:18 ../
-rw-r--r--    1 root root   776 Dec 31  2019 changelog.Debian.gz
-rw-r--r--    1 root root  6041 Dec  3  2019 copyright
-rw-r--r--    1 root root  1820 Jun 26  2019 example_tmux.conf
-rw-r--r--    1 root root   609 Dec  3  2019 NEWS.Debian.gz
-rw-r--r--    1 root root  1920 Jul 29  2019 README
```

Just because there is no default config available for viewing doesn't mean we can't see how tmux will behave once we start using it.

Your Tmux Config

Not sure how your tmux is configured? How do you view your Tmux config? You dump it of course! We can dump our tmux config to a file of our choosing using the **`tmux show-options -g`** command.

Before dumping the config, **you must have at least one tmux server running first.** If you try to dump your config without a tmux server running, you'd get the following message

```
protech@ubuntu3-vm:~$ tmux show-options -g > tmux.conf-default
no server running on /tmp/tmux-1000/default
```

To view your tmux config, do the following:

1. Attach to a new a session
2. Run the command **`tmux show-options -g > tmux.conf-default`**.
    A. Make sure you include the **`-g`**.
    B. This will drop a new file, **`tmux.conf-default`**, in your current working directory.

```
tmux
tmux show-options -g > tmux.conf-default
```

You can either stay attached or detach from the session.

Running **`ls -lhrt`** will show your newly created ***tmux.conf-default*** file, complete with your

```
protech@ubuntu3-vm:~$ ls -lhrt |tail -1
-rw-rw-r-- 1 protech protech 2.8K Jul  8 09:30 tmux.conf-default
```

The resulting tmux conf file contains 55 lines of configuration settings. Options for the tmux status bar color, buffers, display settings, history, pane, titles, detach settings, and much more can be found here.

However, this file <u>CANNOT</u> be used as your tmux config file.

Modifying Your Config

As stated in the man page, a user's tmux config comes from their ***`~/.tmux.conf`***.

If you compare the ***`tmux-conf.default`*** (what we dumped) with the example ***`.tmux.conf`*** file, you will notice that while they may contain SIMILAR information, the contents of the files are formatted differently. We **cannot** simply rename the dump to ***`~/.tmux.conf`***.

If you rename the dumped file to ***.tmux.conf***, and try to start tmux again, you will likely get an error message saying: **`unknown command: activity-action`**.

You can change your tmux config by adding and modifying lines to your ***~/.tmux.conf.*** Making changes within this file will automatically take effect the next time you start a new session. Conversely, you can reload your ***~/.tmux.conf*** from inside a tmux session by running ***tmux source ~/.tmux.conf***.

Essential Tmux.conf Settings

Before we start messing around with plugins and colors and themes, oh my!, Let's implement some quick changes to make our life easier down the road.

Change The Prefix

To change the tmux prefix from **`ctrl + b`** to **`ctrl + a`**, add the following to your ***tmux.conf*** (This is popular due to the same prefix being used with the screen program).

```
set-option -g prefix C-a
```

Then, from inside your tmux session, reload your newly updated config with

```
tmux source ~/.tmux.conf
```

Change the Scrollback Buffer Size

Another incredibly useful option to specify is increasing the tmux **scrollback buffer**. There is nothing worse than running a command and having 3000 lines of java error messages spew across your screen, filling the buffer (the default tmux buffer size is 2000). You can attempt to prevent this by increasing your buffer size to 5000 with the following:

```
set-option -g history-limit 5000
```

NOTE: This will only take effect for new panes. Existing panes will not inherit the increased buffer size. Remember to be consciences of the amount of buffer you wish to keep. The greater the buffer, the greater amount of RAM used, the great the amount of heat generated, the greater the amount of mountain dew consumed.

Change the Pane Navigation Settings

It's super helpful to not have to invoke the prefix in order to navigate through the tmux panes. Adding the following code to your ***.tmux.conf*** will allow you to navigate through the panes using nothing more than the ***Alt + Arrow***

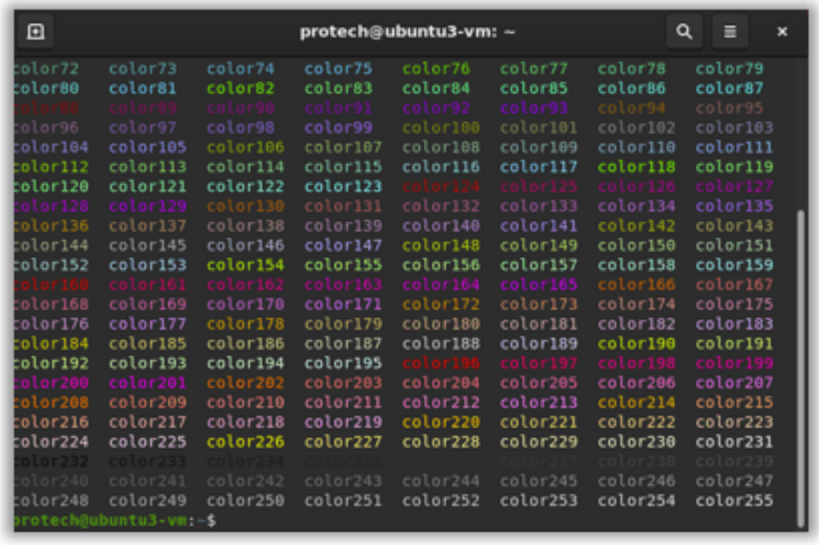
```
bind -n m-Left select-pane -L
bind -n m-Right select-pane -R
bind -n m-Up select-pane -U
bind -n m-Down select-pane -D
```

Tmux Colors & Display Options

Out of the box, the tmux colors are great, but you probably want to change them, which may be why you are on this page. We can change the background colors, status bar colors, pane colors, text colors, and much more. Add all of the following options to your ***~/.tmux.conf*** file, and reload (***`tmux source ~/.tmux.conf`***) it to implement changes in real-time.

Before we start messing around with colors, there is a great answer on superuser, with a short little bash script that I'll provide here. Run this script to get a nice output of all the available colors and their colour### available for tmux:

```
for x in {0..255}; do  printf "\x1b[38;5;${x}mcolor%-5i\x1b[0m" $x
```



```
color72   color73   color74   color75   color76   color77   color78   color79
color80   color81   color82   color83   color84   color85   color86   color87
color88   color89   color90   color91   color92   color93   color94   color95
color96   color97   color98   color99   color100  color101  color102  color103
color104  color105  color106  color107  color108  color109  color110  color111
color112  color113  color114  color115  color116  color117  color118  color119
color120  color121  color122  color123  color124  color125  color126  color127
color128  color129  color130  color131  color132  color133  color134  color135
color136  color137  color138  color139  color140  color141  color142  color143
color144  color145  color146  color147  color148  color149  color150  color151
color152  color153  color154  color155  color156  color157  color158  color159
color160  color161  color162  color163  color164  color165  color166  color167
color168  color169  color170  color171  color172  color173  color174  color175
color176  color177  color178  color179  color180  color181  color182  color183
color184  color185  color186  color187  color188  color189  color190  color191
color192  color193  color194  color195  color196  color197  color198  color199
color200  color201  color202  color203  color204  color205  color206  color207
color208  color209  color210  color211  color212  color213  color214  color215
color216  color217  color218  color219  color220  color221  color222  color223
color224  color225  color226  color227  color228  color229  color230  color231
color232  color233  color234  color235            color237  color238  color239
color240  color241  color242  color243  color244  color245  color246  color247
color248  color249  color250  color251  color252  color253  color254  color255
protech@ubuntu3-vm:~$
```

Tmux status bar color change options from bash script

Status Bar Colors & Options

GLOBAL STATUS BAR CHANGES

Lets change the **background** of the **tmux status bar** color to dark gray

```
set-option -g status-bg colour236
```

Change the color of the **font within the status bar** to yellow

```
set-option -g status-fg colour3
```

LEFT SIDE STATUS BAR CHANGES

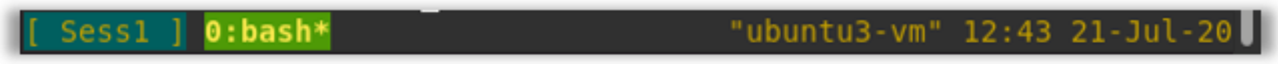
Change the **Session background** to green and display the name of the session surrounded by brackets using [#S]. Then, leave a space before listing the first window.

```
set-window-option -g status-left "#[bg=colour23][ #S ]#[bg=colour23
```

Now, let's start changing the colors of the windows within the status bar. Change the **active window** to a bright green **background**, with yellow **text**.

```
set-window-option -g window-status-current-style 'fg=colour11 bg=cc
```

At this point, we've changed your tmux status bar **background color, font text, session background color, window background color, and window text**. Here is what the status bar should currently look like (session name is Sess1). Note, the right side of the status bar is still the default.



```
[ Sess1 ] 0:bash*                    "ubuntu3-vm" 12:43 21-Jul-20
```

Current status of the tmux status bar with our color changes.

RIGHT SIDE STATUS BAR CHANGES

Now, let's start changing the right side of that status bar that currently says "ubuntu3-vm" Hour:minute day-month-yr.

Specifying the display on the right side is done in one line within the config. We will specify the hostname, month/day, and current time. Each field with a different color.

First, let's look at each part of the line individually, and then we'll combine it all together.

Specifying what options we are modifying

```
set-option -g status-right
```

Adding the **hostname** with ***#h***, with a background color of gray, and text of black

```
#=[fg=colour233,bg=colour8] #h
```

Add the date in a **month/day** format, with a background color of green, and text of black

```
#[fg=colour233,bg=colour23] %m/%d
```

Add the time in an **Hour:Minute:Second** format, with a background color of gray and text of black.

```
#[fg=colour233,bg=colour8] %H:%M:%S '
```

Now, let's combine all those together into one line, and inside your ***tmux.conf*** config file it will look like this:

```
set-option -g status-right '#[fg=colour233,bg=colour8] #h #[fg=colo
```

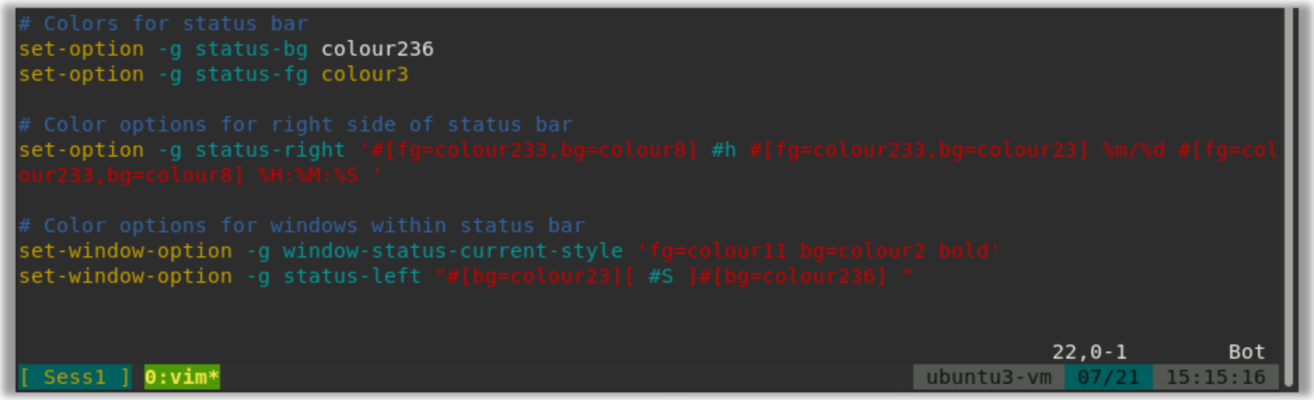
At this point we have added a total of five lines to our ***.tmux.conf***, changing the colors of the session, windows, hostname, time, and date. Our ***.tmux.conf*** will have the following code.

```
# Colors for status bar
set-option -g status-bg colour236
set-option -g status-fg colour3

# Color options for right side of status bar
set-option -g status-right '#[fg=colour233,bg=colour8] #h #[fg=colo

# Color options for windows within status bar
set-window-option -g window-status-current-style 'fg=colour11 bg=co
set-window-option -g status-left "#[bg=colour23][ #S ]#[bg=colour23
```

Our ***.tmux.conf*** and status bar will look like this



```
# Colors for status bar
set-option -g status-bg colour236
set-option -g status-fg colour3

# Color options for right side of status bar
set-option -g status-right '#[fg=colour233,bg=colour8] #h #[fg=colour233,bg=colour23] %m/%d #[fg=col
our233,bg=colour8] %H:%M:%S '

# Color options for windows within status bar
set-window-option -g window-status-current-style 'fg=colour11 bg=colour2 bold'
set-window-option -g status-left "#[bg=colour23][ #S ]#[bg=colour236] "



                                                                            22,0-1          Bot
[ Sess1 ] 0:vim*                                                    ubuntu3-vm 07/21 15:15:16
```

.tmux.conf config changes to modify tmux status bar colors and displayed information.

Pane Border Colors

Now lets change the colors of our panes and their borders.

First, lets change **border color** of **inactive panes** to white

```
set-option -g pane-border-style 'fg=colour255'
```

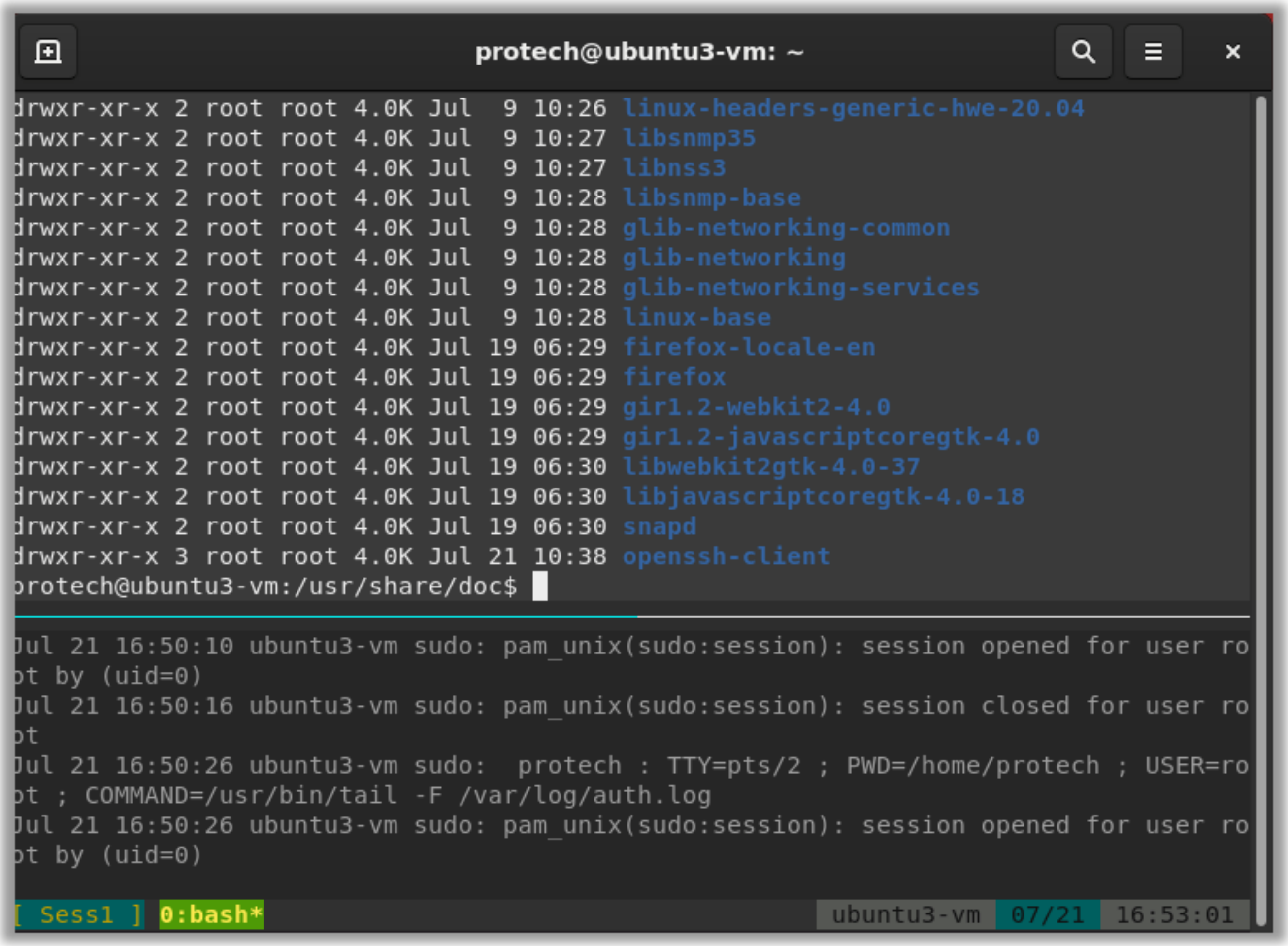
Next, lets change the **border color** of the **active pane** to bright cyan

```
set-option -g pane-active-border-style 'fg=colour51'
```

Next, change the **background** of the **active pane** we are working in is a different color so that it stands out.

```
set-window-option -g window-style 'fg=colour247,bg=colour235'
set-window-option -g window-active-style 'fg=colour255,bg=colour237
```

Now, we can clearly see which pane is active due to the different background colors. Our current settings will result in a session with two panes looking like the following image



```
drwxr-xr-x 2 root root 4.0K Jul  9 10:26 linux-headers-generic-hwe-20.04
drwxr-xr-x 2 root root 4.0K Jul  9 10:27 libsnmp35
drwxr-xr-x 2 root root 4.0K Jul  9 10:27 libnss3
drwxr-xr-x 2 root root 4.0K Jul  9 10:28 libsnmp-base
drwxr-xr-x 2 root root 4.0K Jul  9 10:28 glib-networking-common
drwxr-xr-x 2 root root 4.0K Jul  9 10:28 glib-networking
drwxr-xr-x 2 root root 4.0K Jul  9 10:28 glib-networking-services
drwxr-xr-x 2 root root 4.0K Jul  9 10:28 linux-base
drwxr-xr-x 2 root root 4.0K Jul 19 06:29 firefox-locale-en
drwxr-xr-x 2 root root 4.0K Jul 19 06:29 firefox
drwxr-xr-x 2 root root 4.0K Jul 19 06:29 gir1.2-webkit2-4.0
drwxr-xr-x 2 root root 4.0K Jul 19 06:29 gir1.2-javascriptcoregtk-4.0
drwxr-xr-x 2 root root 4.0K Jul 19 06:30 libwebkit2gtk-4.0-37
drwxr-xr-x 2 root root 4.0K Jul 19 06:30 libjavascriptcoregtk-4.0-18
drwxr-xr-x 2 root root 4.0K Jul 19 06:30 snapd
drwxr-xr-x 3 root root 4.0K Jul 21 10:38 openssh-client
protech@ubuntu3-vm:/usr/share/doc$
```

```
Jul 21 16:50:10 ubuntu3-vm sudo: pam_unix(sudo:session): session opened for user ro
ot by (uid=0)
Jul 21 16:50:16 ubuntu3-vm sudo: pam_unix(sudo:session): session closed for user ro
ot
Jul 21 16:50:26 ubuntu3-vm sudo:  protech : TTY=pts/2 ; PWD=/home/protech ; USER=ro
ot ; COMMAND=/usr/bin/tail -F /var/log/auth.log
Jul 21 16:50:26 ubuntu3-vm sudo: pam_unix(sudo:session): session opened for user ro
ot by (uid=0)
[ Sess1 ] 0:bash*                                     ubuntu3-vm  07/21  16:53:01
```

The active pane has a lighter background and text to help it stand out.

Plugin Manager

Before we start talking about tmux color schemes and themes, a supreme two-second meme about plugins should be gleaned. The reason for the is because

The tmux plugin manager is the most efficient way to handle tmux color schemes and themes. Just use it.

You do not want to be manually downloading and install themes. Install this plugin manager first.

Tmux plugins are handled via a plugin manager called **tpm**. Installation is simple (***git clone*** and adding 3 lines to ***.tmux.conf***). Follow the instructions found on their github installation section to install the plugin manager.

Yep, thats it. Now we can move onto the pretty colors and themes.

Themes & Color Schemes

Tmux themes and color schemes can be found all across the internet. Shoot, even the colors we defined above could be considered a color scheme if they were uploaded to github and gave them a name.

Anyway, let's install a particularly good looking theme that has a few built-in color schemes options based on powerline.

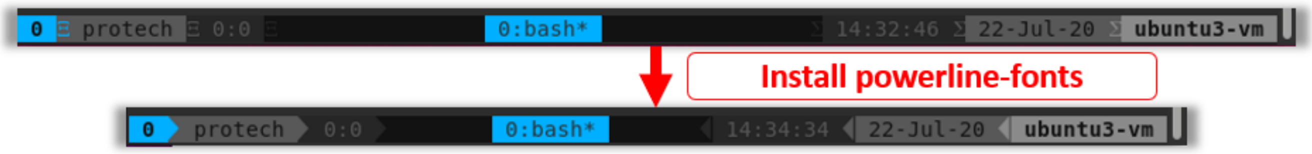
The Tmux Themepack has a super clean layout, in a great variety of colors. Installing it is easier than installing the tmux plugin manager.

Add the following to your ***.tmux.conf***.

```
set -g @plugin 'jimeh/tmux-themepack'
```

Then, run ***ctrl + b I*** to install the theme.

Once the theme is active, you may see some weird characters in the status bar. This is because the powerline fonts are probably not installed. Simply install them via ***sudo apt install powerline-fonts*** or the applicable method for your distro.



```
0  protech  0:0          0:bash*          14:32:46  22-Jul-20  ubuntu3-vm
0  protech  0:0          0:bash*          14:34:34  22-Jul-20  ubuntu3-vm
```

Install powerline-fonts

Install powerline-fonts to ensure tmux theme displays correctly.

Concluding Thoughts

And that's it! By now you should have a good idea of the basics of tmux, including why it is such a powerful tool, how to navigate through its different features, how to customize the colors and layout, how to download and install custom themes and color schemes. From here, the world is your oyster!

Last but not least, no tmux quick start guide would be complete without a cheat sheet, soooo…

Tmux Cheat Sheet

This cheat sheet assumes the default prefix has not been changed.

Session Commands – from the Terminal

Command	Function
tmux	Create and attach to a new session
tmux new -s	Create new session specifying a name
tmux kill-ses -t <Session Name>	Kill a session with name <SessionName>
tmux kill-session -a	Kill all sessions except the current
tmux kill-session -a -t <SessionName>	Kill all sessions except for <SessionName>
tmux ls	List running sessions
tmux a -t <SessionName>	Attach to session <SessionName>
tmux a	Attach to last session

Session commands – Inside a Session

Prefix + Command Key	Function
Ctrl + b $	Rename Session
Ctrl + b d	Detach from session
Ctrl + b (	Move to the previous session
Ctrl + b)	Move to the next session
Ctrl +b s	Show all sessions

Window Commands – Inside a Session

Prefix + Command Key	Function
Ctrl + b c	Create a new window

Prefix + Command Key	Function
Ctrl + b ,	Rename current window
Ctrl + b &	Close current window
Ctrl + d	Exit current window
Ctrl + b n	Jump to next window
Ctrl + b p	Jump to previous window
Ctrl + b <0..1..2..3…9>	Jump to Window number <0-9>

Pane Commands – Inside a Session

Prefix + Command Key	Function
Ctrl + b ;	Switch to previous active pane
Ctrl + b %	Vertically split pane
Ctrl + b “	Horizontally split pane
Ctrl + b {	Shift to the left pane
Ctrl + b }	Shift to the right pane
Ctrl + b <-	Switch to a pane to the left – the direction of arrow
Ctrl +b ->	Switch to a pane to the right – the direction of arrow
Ctrl + b Down arrow	Switch to a downwards pane – the direction of arrow
Ctrl + b Up arrow	Switch to an upwards pane – the direction of arrow
Ctrl + b q	Show the pane numbers
Ctrl + b q <0..1..2..3..0>	Jump to Pane number <0-9>
Ctrl + b !	Transform the current pane into a window
Ctrl + b + Ctrl Up/down Arrow	Resize Pane height up/down
Ctrl + b + Ctrl Left/Right Arrow	Resize Pane width left/right

Prefix + Command Key	Function
Ctrl + b x	Close pane

Linux, tmux. Bookmark.

← Remote Desktop Two Screens in Three Steps

How to Exit TigerVNC from Full Screen Mode →

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name

Email

Website

□ Save my name, email, and site URL in my browser for next time I post a comment.

Post Comment

Search

Recent Posts

Error Deleting Docker image

Create and Share ssh keys

Cannot ssh to localhost, Ubuntu 20.04

Categories

Docker (1)

How-To (12)

Informational (3)

Linux (10)

Powered by Tempera & WordPress.