# 4 tips for better tmux sessions

By Paul W. Frields on August 27, 2018

The tmux utility, a terminal multiplexer, lets you treat your terminal as a multi-paned window into your system. You can arrange the configuration, run different processes in each, and generally make better use of your screen. We introduced some readers to this powerful tool in this earlier article. Here are some tips that will help you get more out of tmux if you're getting started.

This article assumes your current prefix key is **Ctrl+b**. If you've remapped that prefix, simply substitute your prefix in its place.

## Set your terminal to automatically use tmux

One of the biggest benefits of tmux is being able to disconnect and reconnect to sesions at will. This makes remote login sessions more powerful. Have you ever lost a connection and wished you could get back the work you were doing on the remote system? With tmux this problem is solved.

However, you may sometimes find yourself doing work on a remote system, and realize you didn't start a session. One way to avoid this is to have tmux start or attach every time you login to a system with in interactive shell.

Add this to your remote system's *~/.bash_profile* file:

```
if [ -z "$TMUX" ]; then
    tmux attach -t default || tmux new -s default
fi
```

Then logout of the remote system, and log back in with SSH. You'll find you're in a tmux session named *default*. This session will be regenerated at next login if you exit it. But more importantly, if you detach from it as normal, your work is waiting for you next time you login — especially useful if your connection is interrupted.

Of course you can add this to your local system as well. Note that terminals inside most GUIs won't use the default session automatically, because they aren't login shells. While you can change that behavior, it may result in nesting that makes the session less usable, so proceed with caution.

# Use zoom to focus on a single process

While the point of tmux is to offer multiple windows, panes, and processes in a single session, sometimes you need to focus. If you're in a process and need more space, or to focus on a single task, the zoom command works well. It expands the current pane to take up the entire current window space.

Zoom can be useful in other situations too. For instance, imagine you're using a terminal window in a graphical desktop. Panes can make it harder to copy and paste multiple lines from inside your tmux session. If you zoom the pane, you can do a clean copy/paste of multiple lines of data with ease.

To zoom into the current pane, hit **Ctrl+b, z**. When you're finished with the zoom function, hit the same key combo to unzoom the pane.

# Bind some useful commands

By default tmux has numerous commands available. But it's helpful to have some of the more common operations bound to keys you can easily remember. Here are some examples you can add to your *~/.tmux.conf* file to make sessions more enjoyable:

```
bind r source-file ~/.tmux.conf \; display "Reloaded config"
```

This command rereads the commands and bindings in your config file. Once you add this binding, exit any tmux sessions and then restart one. Now after you make any other future changes, simply run **Ctrl+b, r** and the changes will be part of your existing session.

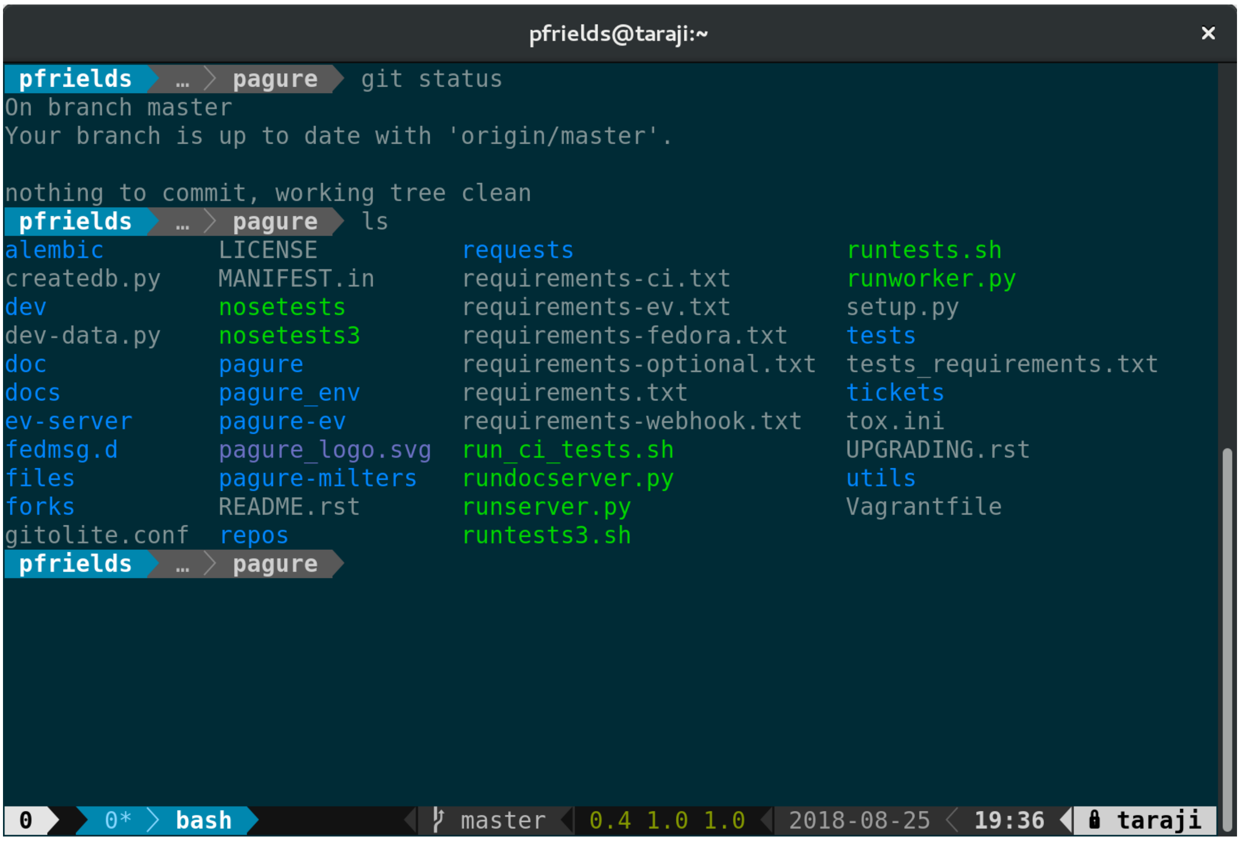
```
bind V split-window -h
bind H split-window
```

These commands make it easier to split the current window across a vertical axis (note that’s **Shift+V**) or across a horizontal axis (**Shift+H**).

If you want to see how all keys are bound, use **Ctrl+B, ?** to see a list. You may see keys bound in *copy-mode* first, for when you’re working with copy and paste inside tmux. The *prefix* mode bindings are where you’ll see ones you’ve added above. Feel free to experiment with your own!

# Use powerline for great justice

As reported in a previous Fedora Magazine article, the *powerline* utility is a fantastic addition to your shell. But it also has capabilities when used with tmux. Because tmux takes over the entire terminal space, the powerline window can provide more than just a better shell prompt.



```
pfrields … pagure  git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
pfrields … pagure  ls
alembic        LICENSE          requests                   runtests.sh
createdb.py    MANIFEST.in      requirements-ci.txt        runworker.py
dev            nosetests        requirements-ev.txt        setup.py
dev-data.py    nosetests3       requirements-fedora.txt    tests
doc            pagure           requirements-optional.txt  tests_requirements.txt
docs           pagure_env       requirements.txt           tickets
ev-server      pagure-ev        requirements-webhook.txt   tox.ini
fedmsg.d       pagure_logo.svg  run_ci_tests.sh            UPGRADING.rst
files          pagure-milters   rundocserver.py            utils
forks          README.rst       runserver.py               Vagrantfile
gitolite.conf  repos            runtests3.sh
pfrields … pagure

 0   0*  bash                      master  0.4 1.0 1.0  2018-08-25  19:36  taraji
```

If you haven't already, follow the instructions in the Magazine's powerline article to install that utility. Then, install the addon using sudo:

```
sudo dnf install tmux-powerline
```

Now restart your session, and you'll see a spiffy new status line at the bottom. Depending on the terminal width, the default status line now shows your current session ID, open windows, system information, date and time, and hostname. If you change directory into a git-controlled project, you'll see the branch and color-coded status as well.

Of course, this status bar is highly configurable as well. Enjoy your new supercharged tmux session, and have fun experimenting with it.

Photo by Pamela Saunders on Unsplash.

**Share:**