

★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



Make your Terminal Productive



Basant Soni · [Follow](#)

7 min read · 1 day ago



Listen



Share



More

When working with servers or command-line-based applications, we spend most of our time on the command line. A good-looking and productive terminal is better in many aspects than a GUI (Graphical User Interface) environment since the command line takes less time for most use cases. Today, we'll look at some features that make a terminal cool and productive.

You can use the following steps on Ubuntu. if you use a different operating system, your commands will likely differ. If you're using Windows, you can choose between Cygwin, WSL, and Git Bash.

Prerequisites

Let's upgrade the system and install some basic tools needed.

```
sudo apt update && sudo apt upgrade  
sudo apt install build-essential curl wget git
```

Z-Shell (ZSH)

Zsh is an extended Bourne shell with many improvements, including some features of Bash and other shells.

Let's install Z-Shell:

```
sudo apt install zsh
```

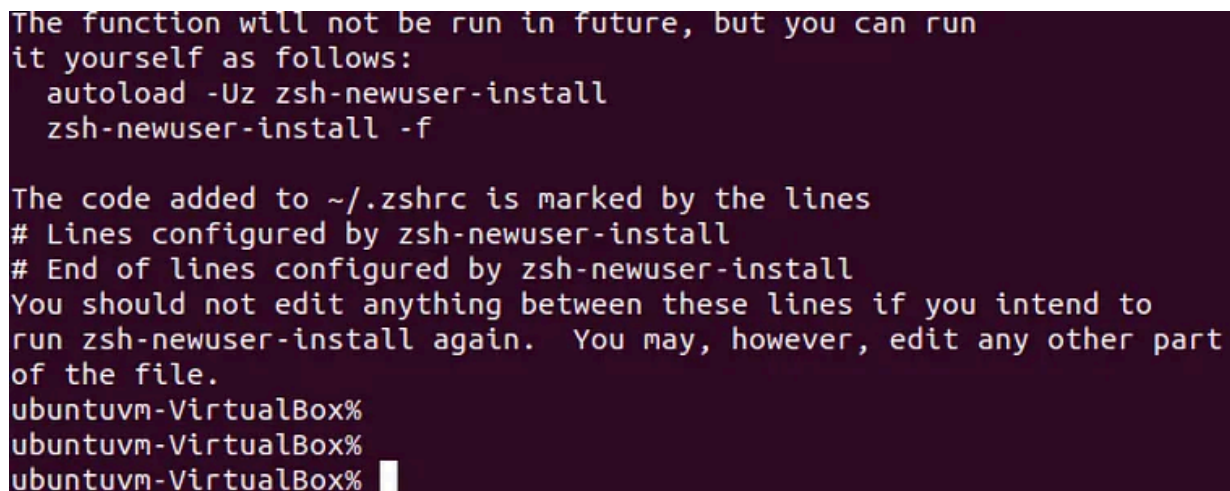
Make it our default shell for our terminal:

```
chsh -s $(which zsh)
```

Now restart the terminal to be welcomed by ZSH. Unlike other shells like Bash, ZSH requires some initial configuration, so it asks for some configuration options the first time we start it and saves them in a file called `.zshrc` in the home directory (`/home/user`) where the user is the current system user.

For now, we'll skip the manual work and get a head start with the default configuration. Press 2, and ZSH will populate the `.zshrc` file with some default options. We can change these later.

The initial configuration setup can be run again as shown in the below image



```
The function will not be run in future, but you can run
it yourself as follows:
  autoload -Uz zsh-newuser-install
  zsh-newuser-install -f

The code added to ~/.zshrc is marked by the lines
# Lines configured by zsh-newuser-install
# End of lines configured by zsh-newuser-install
You should not edit anything between these lines if you intend to
run zsh-newuser-install again. You may, however, edit any other part
of the file.
ubuntuvirtualbox%
ubuntuvirtualbox%
ubuntuvirtualbox% █
```

Oh-My-ZSH

Oh-My-ZSH is a community-driven, open-source framework to manage your ZSH configuration. It comes with many plugins and helpers. It can be installed with one single command as below.

Font

Once the font is installed, it will look like:



Theme

◀ ▶

To set this theme in `.zshrc`:

```
# Set name of the theme to load --- if set to "random", it will
# load a random theme each time oh-my-zsh is loaded, in which case,
# to know which specific one was loaded, run: echo $RANDOM_THEME
# See https://github.com/ohmyzsh/ohmyzsh/wiki/Themes
ZSH_THEME="powerlevel10k/powerlevel10k"

# Set list of themes to pick from when loading at random
```

Close the terminal and start it again. Powerlevel10k will welcome you with the initial setup, Go through the setup with the options you want. You can run this setup again by executing the below command:

```
p10k configure
```

Tools and plugins we can't live without

ZSH-Syntax-Highlighting

This enables the highlighting of commands as you type and helps you catch syntax errors before you execute them:

A terminal window with a dark purple background. The prompt is a green greater-than sign '>'. The command 'ls' is typed in green, followed by a white cursor block.A terminal window with a dark purple background. The prompt is a green greater-than sign '>'. The command 'lss' is typed in red, followed by a white cursor block.

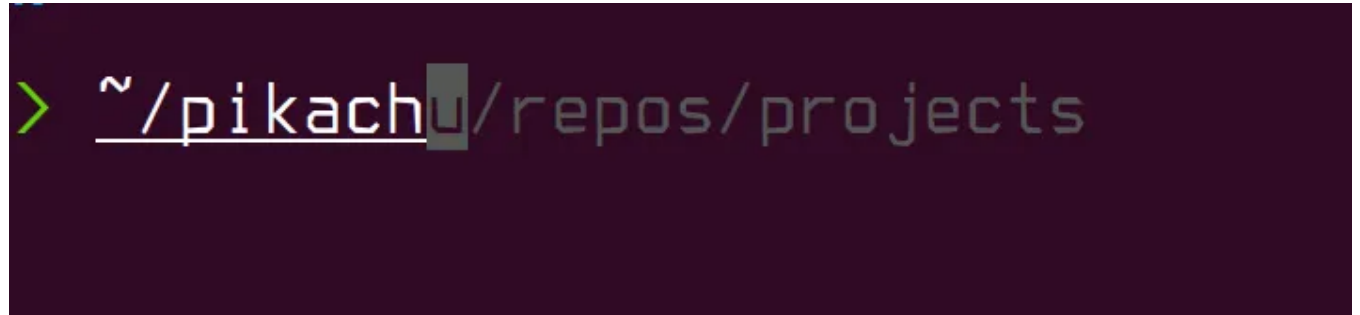
As you can see, “ls” is in green but “lss” is in red.

Execute below command to install it:

```
git clone https://github.com/zsh-users/zsh-syntax-highlighting.git ${ZSH_CUSTOM}
```

ZSH Autosuggestions

This suggests commands as you type based on your history:



The below command is how you can install it by cloning the git repo:

```
git clone https://github.com/zsh-users/zsh-autosuggestions ${ZSH_CUSTOM:~/.oh-
```

ZSH Completions

For some extra ZSH completion scripts, execute below command

```
git clone https://github.com/zsh-users/zsh-completions ${ZSH_CUSTOM:=~/.oh-my-z
```

autojump

It's a faster way of navigating the file system; it works by maintaining a database of directories you visit the most. More details can be found [here](#).

```
sudo apt install autojump
```

```
> j awesome-ui
/home/basant/pikachu/repos/projects/awesome-tools/packages/awesome-ui

~/pikachu/repos/projects/awesome-tools/packages/awesome-ui main *4 +3
> 
```

You can also use the [plugin Z](#) as an alternative if you're not able to install **autojump** or for any other reason.

Internal Plugins

Some plugins come installed with *oh-my-zsh*, and they can be included directly in *.zshrc* file without any installation.

copyfile

It copies the content of a file to the clipboard.

```
copyfile test.txt
```

copypath

It copies the absolute path of the current directory to the clipboard.

```
> copypath
/home/basant/pikachu/repos/courses/kent-c-dodds/create-open-source-library copied to clipboard.

~/pikachu/repos/courses/kent-c-dodds/create-open-source-library main *1
> 
```

copybuffer

This plugin copies the command that is currently typed in the command prompt to the clipboard. It works with the keyboard shortcut CTRL + o.

sudo

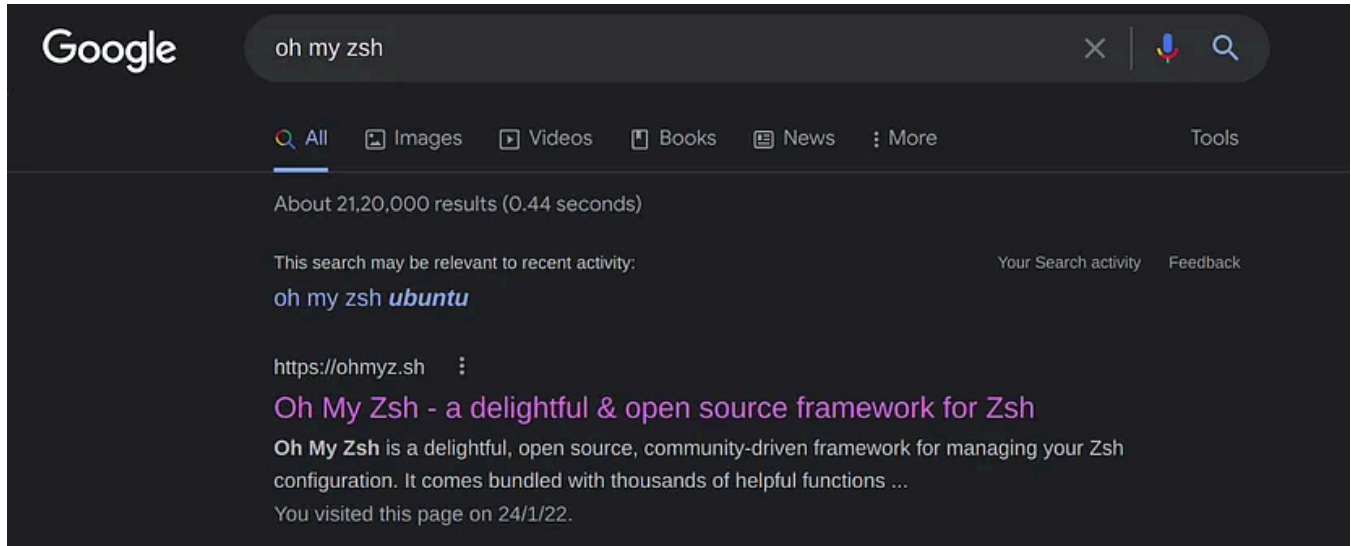
Sometimes, we forget to prefix a command with **sudo**, but that can be done in just a second with this plugin. When you hit the ESC key twice, it will prefix the command you've typed in the terminal with **sudo**.

web-search

This adds some aliases for searching with Google, Wikipedia, etc. For example, if you want to **web-search** with Google, you can execute the below command:

```
google oh my zsh
```

Doing so will open this search in Google:



More details can be found [here](#).

You'd have to add each of these plugins in the `~/.zshrc` file as well. So, in the end, this is how the plugins array in `~/.zshrc` file should look like:

```
# Add wisely, as too many plugins slow
plugins=(
    zsh-autosuggestions
    zsh-syntax-highlighting
    zsh-completions
    autojump
    copyfile
    copypath
    copybuffer
    history
    history-substring-search
    colored-man-pages
    dirhistory
    sudo
    web-search
    git
    docker
    docker-compose
    aws
)
```

I've added some more plugins like docker, aws, etc, which add docker and AWS CLI completion and some aliases. There are plugins for system admins as well, or for anything else you need. You can explore them [here](#).

Enhanced


```
> cd
>
/home/basant/pikachu/repos/projects/system-installer
/home/basant/pikachu/OS/ubuntu
/home/basant/pikachu/OS
/home/basant/Downloads
/home/basant/Downloads/solution
/home/basant/pikachu/repos/projects/awesome-tools/packages
/home/basant/pikachu/repos/projects/awesome-tools/packages/awesome-ui
/etc/apt
/home/basant/pikachu/repos/oss-contribution/chakra-ui
/home/basant/pikachu/repos/oss-contribution
```

Enhanced is the next-gen method to navigate file systems with cli. It works with a fuzzy finder; we'll install fzf for this purpose.

```
sudo apt install fzf
```

Enhanced can be installed as below

```
git clone https://github.com/b4b4r07/enhanced && source enhanced/init.sh
```

You'll have to source the init.sh file from .zshrc as below, replace installation-location with the location where you've installed enhanced

```
source <installation-location>/enhanced/init.sh
```

Colorls

Another tool that makes you say wow is **Colorls**. This tool colorizes the output of the ls command. This is how it looks once you install it:

```
~/pikachu/repos/projects/awesome-tools/packages/awesome-ui main *4 +3  
> ls  
  JS  babel.config.js      JS  jest.config.js        JS  prettier.config.js    ()  tsconfig.json  
  📁  coverage/             @  node_modules/         📄  README.md             📁  yarn.lock  
  📁  dist/                 ()  package.json          📁  src/  
  
~/pikachu/repos/projects/awesome-tools/packages/awesome-ui main *4 +3  
> 
```

It works with ruby, below is how you can install both ruby and colors:

```
sudo apt install ruby ruby-dev ruby-colorize  
sudo gem install colorls
```

Now, restart your terminal and execute the command `colorls` in your terminal to see the magic!

Bonus: We can add some **aliases** as well if we want the same output of `Colorls` when we execute the command `ls`. Note that we're adding another alias for `ls` to make it available as well.

Append to the file `~/.zshrc` with below

```
alias cl='ls'  
alias ls='colorls'  
alias la='colorls -a'  
alias ll='colorls -l'  
alias lla='colorls -la'
```

There're many more and I can't go through them all, but below are some more tools I use every day that you can explore

```
peco  
pcat  
copyq(clipboard manager)  
neovim
```

Automation

Do you want to repeat this process if you've bought a new laptop and want the same setup?

You can automate all of this, so I've created [Project Automator](#). This project does a lot more than just setting up a terminal, it works with Arch Linux as of now, but you can take the parts you need and make it work with almost any *nix system you like.

Explaining how it works is beyond the scope of this article, so I'll have to leave you guys here to explore it on your own.

Conclusion

We need to perform many tasks on our systems, and using a GUI (Graphical User Interface) tool for a task can consume a lot of your time, especially if you repeat the same task daily, like converting a media stream, setting up tools on a system, etc.

Using a command-line tool can save you a lot of time and you can automate repetitive tasks with scripting. It can be a great tool for your arsenal.

Your productivity doesn't just end here; you can install Arch Linux and make your own customized GUI. Visit [the home of *nix customization](#) and this [blog post](#) to explore more.

Zsh

Oh My Zsh

Linux Tutorial

Shell

Bash

[Open in app](#) ↗[Follow](#)

Written by Basant Soni

0 Followers

Software Developer, Linux Ninja