

- My Terminal
- Oh My Zsh
- Zsh config
  - History
  - Aliases
  - Plugins
  - zsh-autosuggestions
  - zsh-syntax-highlighting
- Zsh theme
  - Zsh P10K Theme configuration
- fzf — command-line fuzzy finder
- Links

## 1 My Terminal

This is my reminder for configuring my terminal with [Zsh](#), and gain in efficiency in day to day terminal operations.

The screenshot shows a terminal window titled "nborboen@T480s: ~/Dev/fsd/wp-dev/wp-ops". The window displays a directory listing of files with permissions, owner, group, size, date, and filename. The files listed are .coveragerc, docker, .., Makefile, ., ansible, and .git. Below the listing, a command prompt shows the user has entered a command to run a process-time-optimizations script. The output of the command shows a series of log entries for a test migration, including timestamps and file paths. The terminal window also shows a search bar and window control buttons in the top right corner.

```
nborboen@T480s: ~/Dev/fsd/wp-dev/wp-ops
-rw-rw-r-- 1 nborboen nborboen 503 Jun 16 20:31 .coveragerc
drwxrwxr-x 13 nborboen nborboen 4096 Jun 16 20:31 docker
drwxr-xr-x 8 nborboen nborboen 4096 Jun 17 09:08 ..
-rw-rw-r-- 1 nborboen nborboen 1557 Jun 24 16:18 Makefile
drwxrwxr-x 5 nborboen nborboen 4096 Jul 27 16:51 .
drwxrwxr-x 9 nborboen nborboen 4096 Aug 4 10:29 ansible
drwxrwxr-x 8 nborboen nborboen 4096 Aug 4 11:13 .git
~/Dev/fs/wp-dev/wp-ops process-time-optimizations ?2>>
18116 ./wpsible -l test_migration_wp__test_wpforms -e '{ "wp_destructive": { "te..
18120 ./wpsible -l test_migration_wp__test_wpforms -e '{ "wp_destructive": { "te..
18123 ./wpsible -l test_migration_wp__test_wpforms -e '{ "wp_destructive": { "te..
18124 ./wpsible -l test_migration_wp__test_wpforms -e '{ "wp_destructive": { "te..
18129 ./wpsible -l www__campus__services__canari2 -e '{ "wp_destructive": { "www..
18143 ./wpsible -l test_migration_wp__test_wpforms -e '{ "wp_destructive": { "te..
> 18898 ./wpsible -l test_migration_wp__labs__aaa -e '{ "wp_destructive": { "test..
18905 ./wpsible -l test_migration_wp__labs__aaa -e '{ "wp_destructive": { "test..
241/7115 +S
> ./wpsi
```

Figure 1: My Terminal

## 2 Oh My Zsh

Oh My Zsh is an open source, community-driven framework for managing your zsh configuration. <https://ohmyz.sh/>

```
sh -c "$(wget https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh -O -)"
```

## 3 Zsh config

### 3.1 History

In your ~/.zshrc file:

```
# User configuration
HISTFILE=~/.zsh_history
HISTSIZE=999999999
SAVEHIST=$HISTSIZE
setopt inc_append_history
setopt share_history
```

### 3.2 Aliases

Zsh comes with a ton of alias, just type alias in your terminal to study them.

If you want to use your own aliases, use your ~/.bash\_aliases in order to be able to use them in any shells.

In your ~/.zshrc file:

```
source ~/.bash_aliases
```

### 3.3 Plugins

In your ~/.zshrc file:

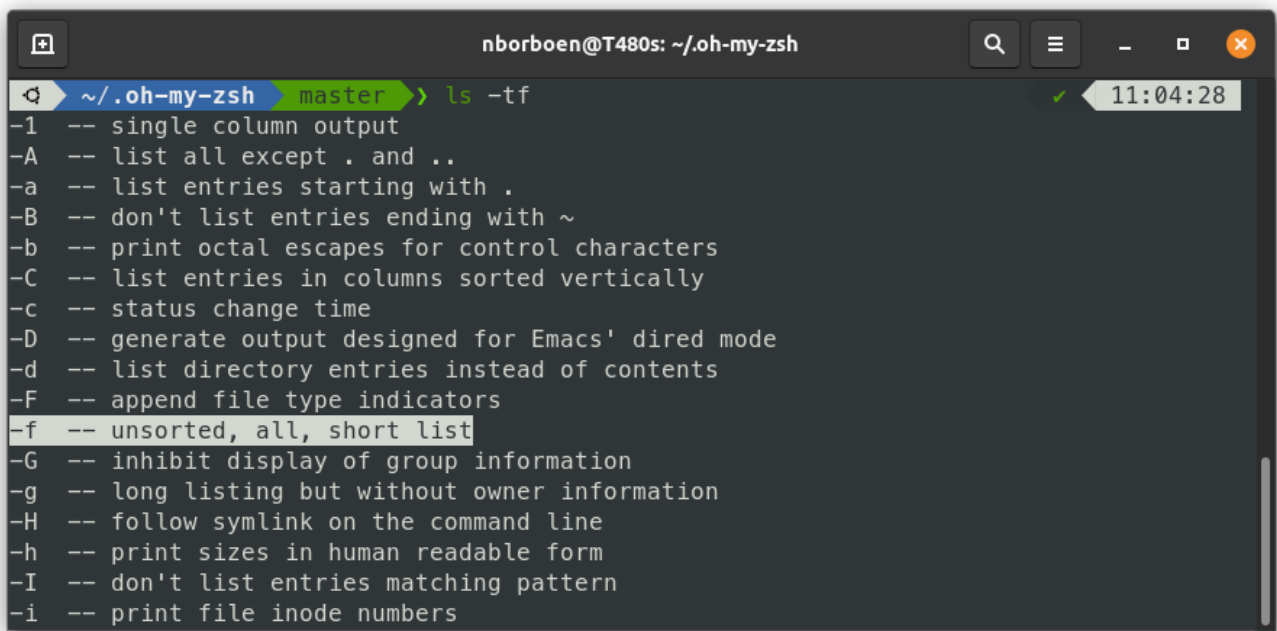
```
plugins=(
  git
  docker
  oc
  kubectl
  npm
  node
  wp-cli
  zsh-autosuggestions
  zsh-syntax-highlighting
)
```

Have a look to <https://github.com/ohmyzsh/ohmyzsh/wiki/Plugins> to find more!

### 3.4 zsh-autosuggestions

Fish-like autosuggestions for Zsh <https://github.com/zsh-users/zsh-autosuggestions>

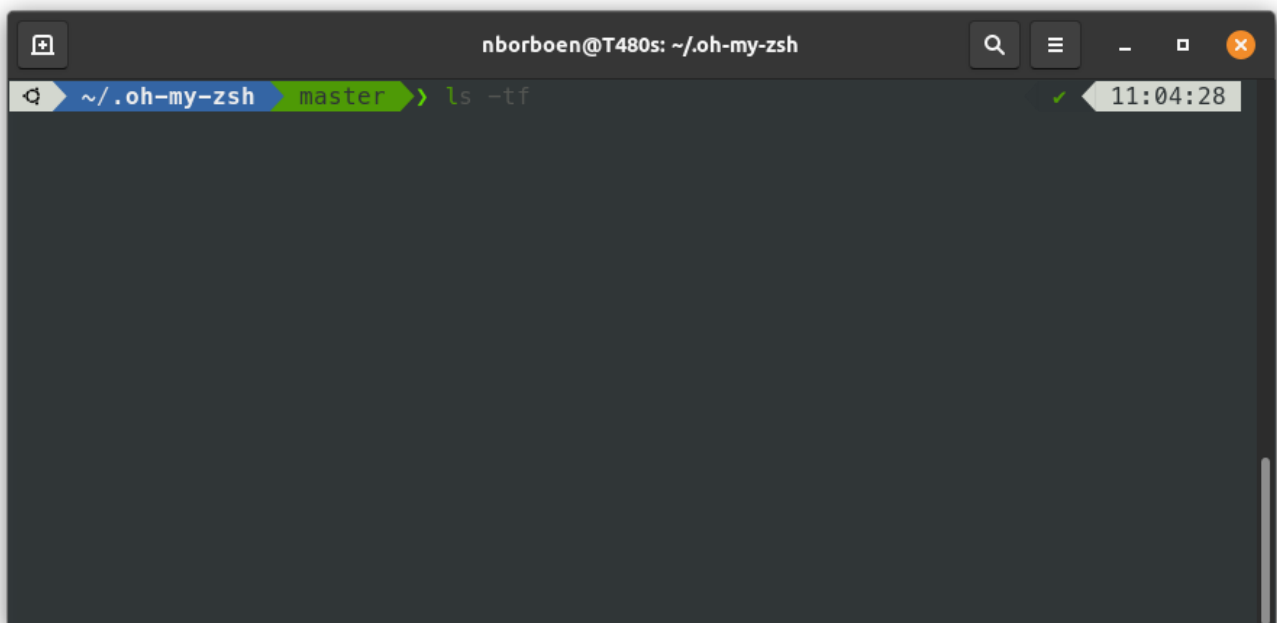
```
git clone https://github.com/zsh-users/zsh-autosuggestions \
  ${ZSH_CUSTOM:-~/.oh-my-zsh/custom}/plugins/zsh-autosuggestions
```



A terminal window titled "nborboen@T480s: ~/.oh-my-zsh" with a search icon, menu icon, and window controls. The prompt is "~/.oh-my-zsh master >>". The command "ls -tf" has been entered. Below the command, a list of options for the "ls" command is displayed, with "-f -- unsorted, all, short list" highlighted. The options are: -l -- single column output, -A -- list all except . and .., -a -- list entries starting with ., -B -- don't list entries ending with ~, -b -- print octal escapes for control characters, -C -- list entries in columns sorted vertically, -c -- status change time, -D -- generate output designed for Emacs' dired mode, -d -- list directory entries instead of contents, -F -- append file type indicators, -f -- unsorted, all, short list, -G -- inhibit display of group information, -g -- long listing but without owner information, -H -- follow symlink on the command line, -h -- print sizes in human readable form, -I -- don't list entries matching pattern, -i -- print file inode numbers. A green checkmark and the time "11:04:28" are in the top right corner.

```
nborboen@T480s: ~/.oh-my-zsh
~/.oh-my-zsh master >> ls -tf
-1 -- single column output
-A -- list all except . and ..
-a -- list entries starting with .
-B -- don't list entries ending with ~
-b -- print octal escapes for control characters
-C -- list entries in columns sorted vertically
-c -- status change time
-D -- generate output designed for Emacs' dired mode
-d -- list directory entries instead of contents
-F -- append file type indicators
-f -- unsorted, all, short list
-G -- inhibit display of group information
-g -- long listing but without owner information
-H -- follow symlink on the command line
-h -- print sizes in human readable form
-I -- don't list entries matching pattern
-i -- print file inode numbers
```

Figure 2: Zsh command completion



A terminal window titled "nborboen@T480s: ~/.oh-my-zsh" with a search icon, menu icon, and window controls. The prompt is "~/.oh-my-zsh master >>". The command "ls -tf" has been entered. The terminal is otherwise empty, showing the result of the command execution. A green checkmark and the time "11:04:28" are in the top right corner.

```
nborboen@T480s: ~/.oh-my-zsh
~/.oh-my-zsh master >> ls -tf
```

Figure 3: Zsh autosuggestions

### 3.5 zsh-syntax-highlighting

Fish shell-like syntax highlighting for Zsh <https://github.com/zsh-users/zsh-syntax-highlighting>

```
git clone https://github.com/zsh-users/zsh-syntax-highlighting.git \
  ${ZSH_CUSTOM:- ~/.oh-my-zsh/custom}/plugins/zsh-syntax-highlighting
```

## 4 Zsh theme

Powerlevel10k is a theme for Zsh. It emphasizes speed, flexibility and out-of-the-box experience <https://github.com/romkatv/powerlevel10k>

```
git clone --depth=1 https://github.com/romkatv/powerlevel10k.git \
  ${ZSH_CUSTOM:- $HOME/.oh-my-zsh/custom}/themes/powerlevel10k
```

### 4.1 Zsh P10K Theme configuration

In your ~/.zshrc file:

```
ZSH_THEME="powerlevel10k/powerlevel10k"
POWERLEVEL9K_LEFT_PROMPT_ELEMENTS+=(dir vsc)
POWERLEVEL9K_RIGHT_PROMPT_ELEMENTS+=(node_version php_version ram cpu_temp)
```

Once your ZSH\_THEME is set, use `p10k configure` to configure p10k.

## 5 fzf — command-line fuzzy finder

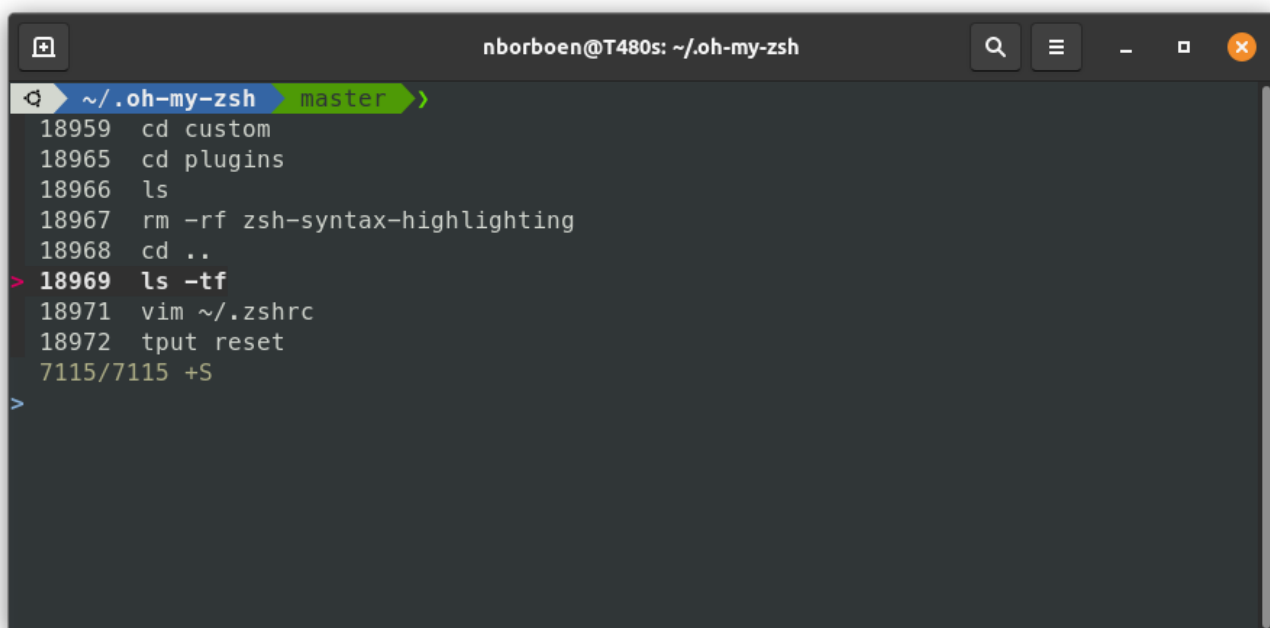
fzf is a general-purpose command-line fuzzy finder <https://github.com/junegunn/fzf>

```
git clone --depth 1 https://github.com/junegunn/fzf.git ~/.fzf
~/.fzf/install
```

Once installed, use `ctrl + r` to search your history.

## 6 Links

- [Your terminal can be much, MUCH more productive](#)
- [Power up your Terminal using Oh-my-Zsh + iTerm2](#)
- [How FZF and ripgrep improved my workflow](#)



A terminal window titled "nborboen@T480s: ~/.oh-my-zsh" with standard window controls. The prompt is "~/.oh-my-zsh master >>". The user has entered a series of commands, each preceded by a line number in red: 18959 cd custom, 18965 cd plugins, 18966 ls, 18967 rm -rf zsh-syntax-highlighting, 18968 cd .., 18969 ls -tf, 18971 vim ~/.zshrc, 18972 tput reset, and 7115/7115 +S. A blue prompt character ">" is visible at the bottom left.

```
nborboen@T480s: ~/.oh-my-zsh
~/.oh-my-zsh master >>
18959 cd custom
18965 cd plugins
18966 ls
18967 rm -rf zsh-syntax-highlighting
18968 cd ..
> 18969 ls -tf
18971 vim ~/.zshrc
18972 tput reset
7115/7115 +S
>
```

Figure 4: fzfc command-line fuzzy finder