

# Complete Guide: From Bash to Oh My Zsh

## Including GitHub Codespaces Configuration

### Terminal Enhancement Guide

August 2, 2025

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Prerequisites</b>	<b>3</b>
<b>3</b>	<b>Step 1: Install Zsh</b>	<b>3</b>
3.1	Ubuntu/Debian Systems . . . . .	3
3.2	CentOS/RHEL/Fedora Systems . . . . .	4
3.3	macOS Systems . . . . .	4
3.4	Verify Zsh Installation . . . . .	4
<b>4</b>	<b>Step 2: Install Oh My Zsh</b>	<b>4</b>
4.1	Method 1: Using curl . . . . .	4
4.2	Method 2: Using wget . . . . .	4
4.3	Method 3: Manual Installation . . . . .	4
<b>5</b>	<b>Step 3: Change Default Shell to Zsh</b>	<b>5</b>
5.1	Standard Linux/Unix Systems . . . . .	5
5.2	GitHub Codespaces - Method 1 . . . . .	5
5.3	GitHub Codespaces - Method 2 (Alternative) . . . . .	5
5.4	Verify Shell Change . . . . .	5
<b>6</b>	<b>Step 4: Restart Terminal</b>	<b>5</b>
<b>7</b>	<b>Step 5: Basic Configuration</b>	<b>6</b>
7.1	Edit Zsh Configuration . . . . .	6
7.2	Key Configuration Options . . . . .	6
<b>8</b>	<b>Step 6: Install Popular Plugins</b>	<b>6</b>
8.1	Zsh Autosuggestions . . . . .	6
8.2	Zsh Syntax Highlighting . . . . .	7
8.3	Powerlevel10k Theme (Optional) . . . . .	7
<b>9</b>	<b>Step 7: Apply Changes</b>	<b>7</b>
<b>10</b>	<b>Troubleshooting</b>	<b>7</b>
10.1	Common Issues and Solutions . . . . .	7
10.2	Codespaces-Specific Issues . . . . .	8
<b>11</b>	<b>Advanced Configuration</b>	<b>8</b>
11.1	Custom Prompt Configuration . . . . .	8
11.2	Environment Variables . . . . .	8
<b>12</b>	<b>Useful Aliases and Functions</b>	<b>8</b>
<b>13</b>	<b>Verification and Testing</b>	<b>9</b>
13.1	Test Your Setup . . . . .	9

<b>14 Conclusion</b>	<b>10</b>
14.1 Next Steps . . . . .	10
14.2 Resources . . . . .	10

# 1 Introduction

This guide provides step-by-step instructions for switching from the default bash shell to Zsh with the Oh My Zsh framework. We'll cover installation, configuration, and specific instructions for GitHub Codespaces environments.

## Information

### What is Oh My Zsh?

Oh My Zsh is an open-source framework for managing Zsh configuration that provides:

- Beautiful themes and prompts
- Hundreds of helpful plugins
- Auto-completion enhancements
- Git integration
- Customizable aliases and functions

# 2 Prerequisites

Before starting, ensure you have:

1. Terminal access (bash, zsh, or similar)
2. Internet connection for downloading packages
3. Basic command-line knowledge
4. Administrative privileges (sudo access)

## Warning

### Backup Your Current Configuration

Before making changes, backup your existing shell configuration:

#### Backup .bashrc

```
cp ~/.bashrc ~/.bashrc.backup
```

#### Backup .bash\_profile

```
cp ~/.bash_profile ~/.bash_profile.backup
```

# 3 Step 1: Install Zsh

## 3.1 Ubuntu/Debian Systems

### Update package list

```
sudo apt update
```

### Install Zsh

```
sudo apt install zsh
```

## 3.2 CentOS/RHEL/Fedora Systems

### Install Zsh (CentOS/RHEL)

```
sudo yum install zsh
```

### Install Zsh (Fedora)

```
sudo dnf install zsh
```

## 3.3 macOS Systems

### Install Zsh with Homebrew

```
brew install zsh
```

### Install Zsh with MacPorts

```
sudo port install zsh
```

## 3.4 Verify Zsh Installation

### Check Zsh version

```
zsh --version
```

### Find Zsh location

```
which zsh
```

Expected output should show Zsh version 5.0 or higher.

## 4 Step 2: Install Oh My Zsh

### 4.1 Method 1: Using curl

#### Install Oh My Zsh with curl

```
sh -c "$(curl -fsSL \
    https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
```

### 4.2 Method 2: Using wget

#### Install Oh My Zsh with wget

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

### 4.3 Method 3: Manual Installation

#### Clone Oh My Zsh repository

```
git clone https://github.com/ohmyzsh/ohmyzsh.git ~/.oh-my-zsh
```

#### Copy template configuration

```
cp ~/.oh-my-zsh/templates/zshrc.zsh-template ~/.zshrc
```

## 5 Step 3: Change Default Shell to Zsh

### 5.1 Standard Linux/Unix Systems

Change shell (standard method)

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

### 5.2 GitHub Codespaces - Method 1

For GitHub Codespaces environments, use this command:

Change shell (Codespaces method 1)

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

### 5.3 GitHub Codespaces - Method 2 (Alternative)

If the first method doesn't work, try:

Change shell (Codespaces method 2)

```
sudo chsh "$(id -un)" --shell "$(which zsh)"
```

#### Information

##### Why Different Commands for Codespaces?

GitHub Codespaces runs in a containerized environment where:

- Standard `chsh` might not have sufficient permissions
- `sudo` is required for shell changes
- User identification needs to be explicit
- Container restrictions may require alternative syntax

### 5.4 Verify Shell Change

Check current shell

```
echo $SHELL
```

Check user shell setting

```
grep $(whoami) /etc/passwd
```

The output should show the path to zsh (e.g., `/usr/bin/zsh`).

## 6 Step 4: Restart Terminal

After changing the shell, you need to restart your terminal session:

1. Close your current terminal
2. Open a new terminal session
3. Or logout and login again
4. For Codespaces: Restart the container or reload the window

#### Success

If successful, you should see the Oh My Zsh welcome message and a new colorful prompt!

## 7 Step 5: Basic Configuration

### 7.1 Edit Zsh Configuration

Edit `.zshrc` with nano

```
nano /.zshrc
```

Edit `.zshrc` with vim

```
vim /.zshrc
```

### 7.2 Key Configuration Options

Here's a basic `.zshrc` configuration:

```
# Oh My Zsh installation path
export ZSH="$HOME/.oh-my-zsh"

# Theme selection
ZSH_THEME="robbyrussell" # Default theme
# ZSH_THEME="agnoster"    # Popular alternative
# ZSH_THEME="powerlevel10k/powerlevel10k" # Advanced theme

# Plugins
plugins=(
  git
  zsh-autosuggestions
  zsh-syntax-highlighting
  docker
  kubectl
  node
  npm
)

# Source Oh My Zsh
source $ZSH/oh-my-zsh.sh

# Custom aliases
alias ll="ls -la"
alias la="ls -A"
alias l="ls -CF"
alias ..="cd .."
alias ...="cd ../.."

# Custom functions
mkcd() {
  mkdir -p "$1" && cd "$1"
}
```

## 8 Step 6: Install Popular Plugins

### 8.1 Zsh Autosuggestions

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

## 8.2 Zsh Syntax Highlighting

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

## 8.3 Powerlevel10k Theme (Optional)

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

Then set `ZSH_THEME="powerlevel10k/powerlevel10k"` in your `.zshrc`.

# 9 Step 7: Apply Changes

After making configuration changes:

```
Reload configuration
source ~/.zshrc
```

# 10 Troubleshooting

## 10.1 Common Issues and Solutions

### 1. Permission Denied

```
Fix permissions
sudo chsh -s $(which zsh) $USER
```

### 2. Zsh Not Found

```
Check Zsh installation
which zsh
```

If empty, reinstall zsh:

```
Reinstall Zsh (Ubuntu/Debian)
sudo apt install zsh
```

### 3. Oh My Zsh Installation Failed

Check internet connection and try manual installation:

```
Test connection
curl -fsSL https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh
```

### 4. Plugins Not Working

```
Check plugin directory
ls ~/.oh-my-zsh/custom/plugins/
```

Restart terminal after adding plugins.

## 10.2 Codespaces-Specific Issues

### Warning

#### Container Persistence

In GitHub Codespaces, remember that:

- Shell changes persist across container restarts
- Custom configurations in `/.zshrc` are preserved
- Installed plugins remain available
- Theme preferences are maintained

## 11 Advanced Configuration

### 11.1 Custom Prompt Configuration

```
# Custom prompt with git status
PROMPT='%F{cyan}%n@m%f:%F{yellow}%~%f$(git_prompt_info) %# '
ZSH_THEME_GIT_PROMPT_PREFIX=" %F{red}("
ZSH_THEME_GIT_PROMPT_SUFFIX=")%f"
ZSH_THEME_GIT_PROMPT_DIRTY="%F{yellow}*%f"
ZSH_THEME_GIT_PROMPT_CLEAN=""
```

### 11.2 Environment Variables

```
# Development environment
export EDITOR="code"
export BROWSER="google-chrome"
export TERM="xterm-256color"

# Path additions
export PATH="$HOME/.local/bin:$PATH"
export PATH="$HOME/bin:$PATH"

# Node.js
export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"
```

## 12 Useful Aliases and Functions

```
# Git aliases
alias gs="git status"
alias ga="git add"
alias gc="git commit"
alias gp="git push"
alias gl="git log --oneline"

# Docker aliases
alias dps="docker ps"
alias dpa="docker ps -a"
alias di="docker images"
alias drm="docker rm"
alias drmi="docker rmi"

# System aliases
```



```
alias h="history"
alias j="jobs"
alias c="clear"
alias e="exit"

# Directory navigation
alias home="cd ~"
alias root="cd /"
alias dtop="cd ~/Desktop"
alias docs="cd ~/Documents"

# File operations
alias cp="cp -i"
alias mv="mv -i"
alias rm="rm -i"
alias mkdir="mkdir -p"

# Network
alias ping="ping -c 5"
alias ports="netstat -tuln"
```

## 13 Verification and Testing

### 13.1 Test Your Setup

#### 1. Check Shell

```
Verify current shell
echo $SHELL
```

```
Check process
ps -p $$
```

#### 2. Test Oh My Zsh

```
Check Oh My Zsh version
omz version
```

```
List plugins
omz plugin list
```

#### 3. Test Autocompletion

Type `git <TAB><TAB>` or `docker <TAB><TAB>` to test autocompletion.

#### 4. Test Plugins

Type a command and see suggestions, use arrow keys to navigate history.

## 14 Conclusion

You have successfully:

### Success

- Installed Zsh shell
- Installed Oh My Zsh framework
- Changed default shell (including Codespaces methods)
- Configured themes and plugins
- Set up useful aliases and functions
- Verified the installation

### 14.1 Next Steps

1. Explore more themes: <https://github.com/ohmyzsh/ohmyzsh/wiki/Themes>
2. Discover plugins: <https://github.com/ohmyzsh/ohmyzsh/wiki/Plugins>
3. Customize your prompt further
4. Share your configuration with team members
5. Consider using dotfiles for configuration management

### 14.2 Resources

- Oh My Zsh GitHub: <https://github.com/ohmyzsh/ohmyzsh>
- Zsh Documentation: <https://zsh.sourceforge.io/Doc/>
- Awesome Zsh Plugins: <https://github.com/unixorn/awesome-zsh-plugins>
- Powerlevel10k: <https://github.com/romkatv/powerlevel10k>

### Information

#### Remember

Your terminal is now more powerful and user-friendly. Take time to explore the features and customize it to your workflow. Happy coding!