

# Zsh 安装与配置，使用 Oh-My-Zsh 美化终端

Leehow 收录于 技术

2024-01-02 2024-11-16 约 2100 字 预计阅读 5 分钟



传统的 bash 功能比较简陋，且不美观。本文基于 Ubuntu22.04 LTS 系统，安装 zsh，并使用 oh-my-zsh 对终端进行美化。Oh My Zsh 是基于 zsh 命令行的一个扩展工具集，提供了丰富的扩展功能。

## 1 环境配置

### 1.1 安装基本工具

✓ Bash

安装基本工具

- 1 # 更新软件源
- 2 sudo apt update && sudo apt upgrade -y
- 3 # 安装 zsh git curl
- 4 sudo apt install zsh git curl -y



设置默认终端为 zsh (**注意：不要使用 sudo**)。

✓ Bash

- 1 chsh -s /bin/zsh

### 1.2 安装 oh-my-zsh

官网: <http://ohmyz.sh/> 。安装方式任选一个即可。

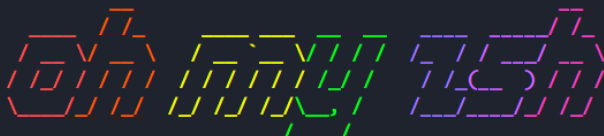
Method	Command
<b>curl</b>	<code>sh -c "\$(curl -fsSL https://install.ohmyz.sh/)"</code>
<b>wget</b>	<code>sh -c "\$(wget -O- https://install.ohmyz.sh/)"</code>
<b>fetch</b>	<code>sh -c "\$(fetch -o - https://install.ohmyz.sh/)"</code>
国内curl镜像 	<code>sh -c "\$(curl -fsSL https://gitee.com/pocmon/ohmyzsh/raw/master/tools/install.sh)"</code>
国内wget镜像 	<code>sh -c "\$(wget -O- https://gitee.com/pocmon/ohmyzsh/raw/master/tools/install.sh)"</code>

注意: 同意使用 Oh-my-zsh 的配置模板覆盖已有的 `.zshrc`。

```
hao@cse:~$ sh -c "$(curl -fsSL https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
Cloning Oh My Zsh...
remote: Enumerating objects: 1376, done.
remote: Counting objects: 100% (1376/1376), done.
remote: Compressing objects: 100% (1323/1323), done.
remote: Total 1376 (delta 31), reused 1124 (delta 28), pack-reused 0
Receiving objects: 100% (1376/1376), 3.19 MiB | 1.99 MiB/s, done.
Resolving deltas: 100% (31/31), done.
From https://github.com/ohmyzsh/ohmyzsh
* [new branch]      master    -> origin/master
Branch 'master' set up to track remote branch 'master' from 'origin'.
Already on 'master'
/home/hao

Looking for an existing zsh config...
Using the Oh My Zsh template file and adding it to /home/hao/.zshrc.

Time to change your default shell to zsh:
Do you want to change your default shell to zsh? [Y/n] Y
Changing your shell to /usr/bin/zsh...
[sudo] password for hao:
Shell successfully changed to '/usr/bin/zsh'.

 ....is now installed!

Before you scream Oh My Zsh! look over the `zshrc` file to select plugins, themes, and options.

• Follow us on Twitter: https://twitter.com/ohmyzsh
• Join our Discord community: https://discord.gg/ohmyzsh
• Get stickers, t-shirts, coffee mugs and more: https://shop.planetargon.com/collections/oh-my-zsh

→ ~
```

### 1.3 从 `.bashrc` 中迁移配置 (可选)

如果之前在使用 `bash` 时自定义了一些环境变量、别名等, 那么在切换到 `zsh` 后, 你需要手动迁移这些自定义配置。

### ✓ Bash

```
1 # 查看bash配置文件，并手动复制自定义配置
2 cat ~/.bashrc
3 # 编辑zsh配置文件，并粘贴自定义配置
4 nano ~/.zshrc
5 # 启动新的zsh配置
6 source ~/.zshrc
```

`root` 用户在执行 `sudo su` 命令后，再运行上述代码查看、手动复制、粘贴自定义配置。

## 2 配置主题

### 2.1 自定义主题

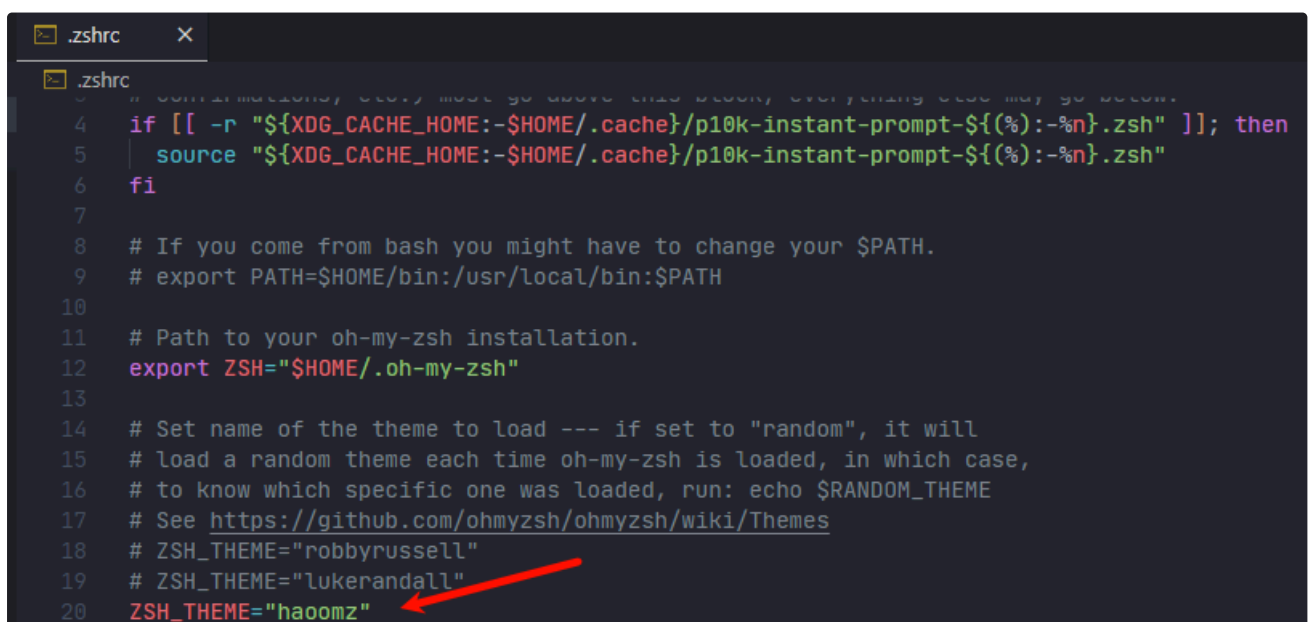
#### ✓ Bash

```
1 sudo wget -O $ZSH_CUSTOM/themes/haoomz.zsh-theme https://cdn.haoyep.com/gh/leegica
```

编辑 `~/.zshrc` 文件，将 `ZSH_THEME` 设为 `haoomz`。当然你也可以设置为其他主题，例如 `lukerandall`、`robbyrussell`。

#### ✓ Bash

```
1 nano ~/.zshrc
2
3 ZSH_THEME="haoomz"
4
5 source ~/.zshrc
```

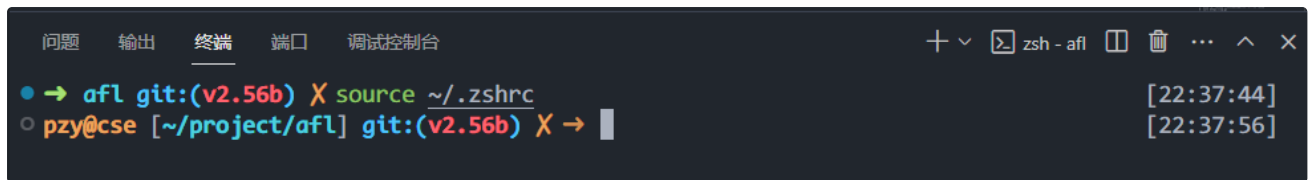


```
.zshrc
# If you come from bash you might have to change your $PATH.
# export PATH=$HOME/bin:/usr/local/bin:$PATH

# Path to your oh-my-zsh installation.
export ZSH="$HOME/.oh-my-zsh"

# 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="robbyrussell"
# ZSH_THEME="lukerandall"
ZSH_THEME="haoomz"
```

效果如下（`robbyrussell` → `haoomz`）：

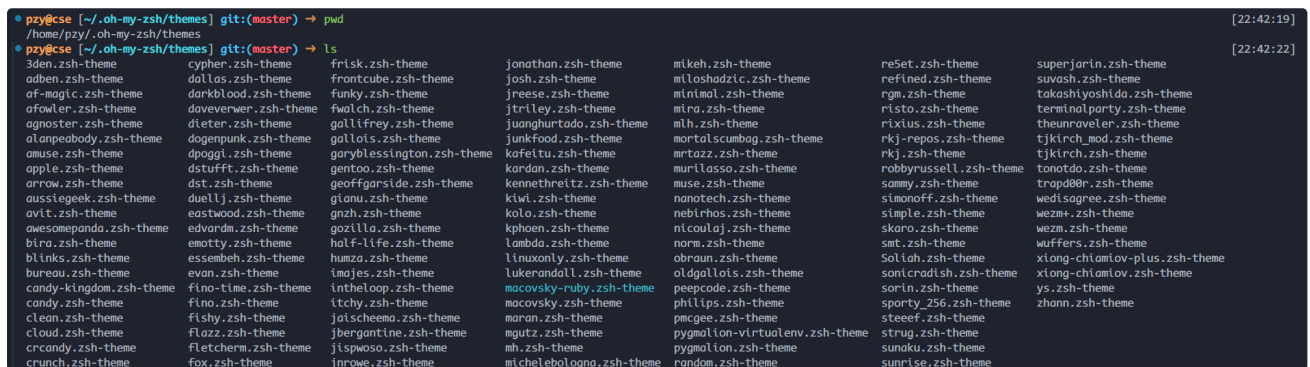


## 2.2 推荐主题

你可以在[内置主题样式截图](#)中查看所有 zsh 内置的主题样式和对应的主题名。这些内置主题已经放在 `~/.oh-my-zsh/themes` 目录下，不需要再下载。

### ▼ Bash

```
1 cd ~/.oh-my-zsh/themes && ls
```



### 2.2.1 powerlevel10k

根据 [What's the best theme for Oh My Zsh?](#) 中的排名，以及自定义化、美观程度，强烈建议使用 [powerlevel10k](#) 主题。

### ▼ Bash

```
1 git clone --depth=1 https://github.com/romkatv/powerlevel10k.git ${ZSH_CUSTOM:-$HOME}
2
3 # 中国用户可以使用 gitee.com 上的官方镜像加速下载
4 git clone --depth=1 https://gitee.com/romkatv/powerlevel10k.git ${ZSH_CUSTOM:-$HOME}
```

在 `~/.zshrc` 设置 `ZSH_THEME="powerlevel10k/powerlevel10k"`。接下来，终端会自动引导你配置 `powerlevel10k`。

## 3 安装插件

`oh-my-zsh` 已经内置了 `git` 插件，内置插件可以在 `~/.oh-my-zsh/plugins` 中查看，下面介绍一下我常用的插件，更多插件可以在 [awesome-zsh-plugins](#) 里查看。

### 3.1 插件推荐

#### 3.1.1 zsh -autosuggestions

[zsh-autosuggestions](#) 是一个命令提示插件，当你输入命令时，会自动推测你可能需要输入的命令，按下右键可以快速采用建议。效果如下：

```
ozy@cse [~/oh-my-zsh/themes] git:(master) → source ~/.zshrc [22:49:37]
```

安装方式：把插件下载到本地的 `~/oh-my-zsh/custom/plugins` 目录。

#### ▼ Bash

```
1 git clone https://github.com/zsh-users/zsh-autosuggestions ${ZSH_CUSTOM:-~/.oh-my-
2
3 # 中国用户可以使用下面任意一个加速下载
4 # 加速1
5 git clone https://github.moeyy.xyz/https://github.com/zsh-users/zsh-autosuggestion
6 # 加速2
7 git clone https://gh.xmlly.dev/https://github.com/zsh-users/zsh-autosuggestions ${Z
8 # 加速3
9 git clone https://gh.api.99988866.xyz/https://github.com/zsh-users/zsh-autosuggest:
```

### 3.1.2 zsh-syntax-highlighting

[zsh-syntax-highlighting](#) 是一个命令语法校验插件，在输入命令的过程中，若指令不合法，则指令显示为红色，若指令合法就会显示为绿色。效果如下：

```
ozy@cse [~/oh-my-zsh/themes] git:(master) → echo [22:49:37]
```

安装方式：把插件下载到本地的 `~/oh-my-zsh/custom/plugins` 目录。

#### ▼ Bash

```
1 git clone https://github.com/zsh-users/zsh-syntax-highlighting.git ${ZSH_CUSTOM:-~/.
2
3 # 中国用户可以使用下面任意一个加速下载
4 # 加速1
5 git clone https://github.moeyy.xyz/https://github.com/zsh-users/zsh-syntax-highlig
6 # 加速2
7 git clone https://gh.xmlly.dev/https://github.com/zsh-users/zsh-syntax-highlighting
8 # 加速3
9 git clone https://gh.api.99988866.xyz/https://github.com/zsh-users/zsh-syntax-high:
```

### 3.1.3 z

`oh-my-zsh` 内置了 `z` 插件。`z` 是一个文件夹快捷跳转插件，对于曾经跳转过的目录，只需要输入最终目标文件夹名称，就可以快速跳转，避免再输入长串路径，提高切换文件夹的效率。效果如下：

```

• pzy@cse [~] → cd /home/pzy/project/afl/afl-qemu-test [22:54:02]
• pzy@cse [~/project/afl/afl-qemu-test] git:(v2.56b) X → cd ~ [22:54:05]
• pzy@cse [~] → z test [22:54:12]
• pzy@cse [~/project/afl/afl-qemu-test] git:(v2.56b) X → pwd [22:54:15]
  /home/pzy/project/afl/afl-qemu-test
• pzy@cse [~/project/afl/afl-qemu-test] git:(v2.56b) X → [22:54:18]

```

### 3.1.4 extract

oh-my-zsh 内置了 `extract` 插件。`extract` 用于解压任何压缩文件，不必根据压缩文件的后缀名来记忆压缩软件。使用 `x` 命令即可解压文件，效果如下：

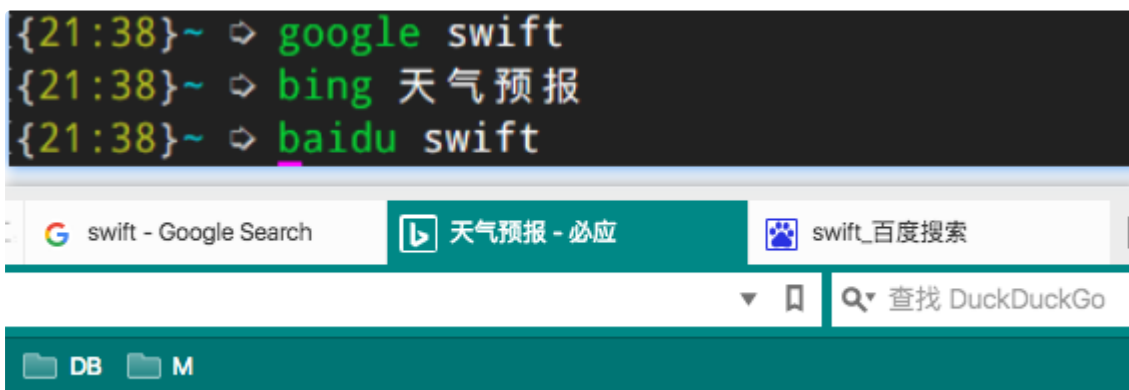
```

• pzy@cse [~/Documents] → ls
  qemu-2.10.0.tar.xz
• pzy@cse [~/Documents] → x qemu-2.10.0.tar.xz
  extract: extracting to qemu-2.10.0
  [
  qemu-2.10.0/device-hotplug.c
  qemu-2.10.0/.gdbinit
• pzy@cse [~/Documents] → ls
  qemu-2.10.0  qemu-2.10.0.tar.xz
• pzy@cse [~/Documents] → [

```

### 3.1.5 web-search

oh-my-zsh 内置了 `web-search` 插件。`web-search` 能让我们在命令行中使用搜索引擎进行搜索。使用 `搜索引擎关键字+搜索内容` 即可自动打开浏览器进行搜索。效果如下：



#### 重要

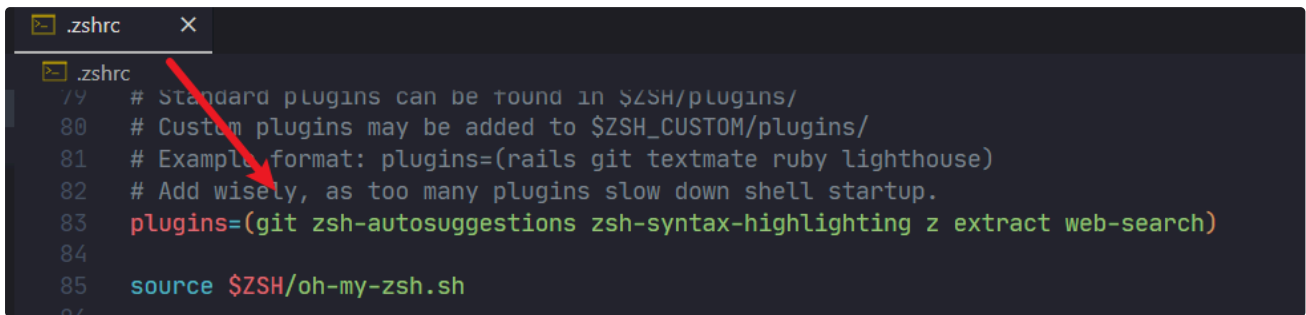
最后，记得启用所有插件。

## 3.2 启用插件

修改 `~/.zshrc` 中插件列表为：

## ▼ Bash

```
1 plugins=(git zsh-autosuggestions zsh-syntax-highlighting z extract web-search)
```



```
.zshrc
# Standard plugins can be found in $ZSH/plugins/
# Custom plugins may be added to $ZSH_CUSTOM/plugins/
# Example format: plugins=(rails git textmate ruby lighthouse)
# Add wisely, as too many plugins slow down shell startup.
plugins=(git zsh-autosuggestions zsh-syntax-highlighting z extract web-search)
source $ZSH/oh-my-zsh.sh
```

## 💡 提示

部分插件需要参考[插件介绍](#)进行安装。

开启新的 Shell 或执行 `source ~/.zshrc`，就可以开始体验插件。

# 4 Tips

## 4.1 root 用户

当你配置好登陆用户的 zsh 后，如果使用 `sudo su` 命令进入 `root` 用户的终端，发现还是默认的 `bash`。建议在 `root` 用户的终端下，也安装 `on my zsh`，设置与普通用户不同的主题以便区分，插件可以使用一样的。`root` 用户的 `~/.zshrc` 配置，仅供参考：

## ▼ Bash

```
1 ZSH_THEME="ys"
2 plugins=(git zsh-autosuggestions zsh-syntax-highlighting z extract web-search)
3 # 或
4 plugins=(git colored-man-pages colorize cp man command-not-found sudo suse ubuntu ;
```

## 4.2 配置本地代理

如果你配置了本地代理，并希望终端的 git 等命令使用代理，那么可以在 `~/.zshrc` 中添加：

## ▼ Bash

```
1 # 为 curl wget git 等设置代理
2 proxy () {
3     export ALL_PROXY="socks5://127.0.0.1:1089"
4     export all_proxy="socks5://127.0.0.1:1089"
5 }
6
7 # 取消代理
8 unproxy () {
```

```
9   unset ALL_PROXY
10  unset all_proxy
11 }
```

### 💡 提示

这里假设本地代理的端口是 `1089`。



```
.zshrc
112
113 # 为 curl wget git 等设置代理
114 proxy () {
115     export ALL_PROXY="socks5://127.0.0.1:1089"
116     export all_proxy="socks5://127.0.0.1:1089"
117 }
118
119 # 取消代理
120 unproxy () {
121     unset ALL_PROXY
122     unset all_proxy
123 }
124
```

以后在使用 `git` 等命令之前，只需要在终端中输入 `proxy` 命令，即可使用本地代理。

## 4.2.1 WSL 配置本地代理

### ▼ Bash

```
1 host_ip=$(cat /etc/resolv.conf |grep "nameserver" |cut -f 2 -d " ")
2 # 为 curl wget git npm apt 等设置代理
3 proxy () {
4     export ALL_PROXY="http://$host_ip:10811"
5     export all_proxy="http://$host_ip:10811"
6     # echo -e "Acquire::http::Proxy \"http://$host_ip:10811\";" | sudo tee -a /etc/ap
7     # echo -e "Acquire::https::Proxy \"http://$host_ip:10811\";" | sudo tee -a /etc/a
8 }
9
10 # 取消代理
11 unproxy () {
12     unset ALL_PROXY
13     unset all_proxy
14     # sudo sed -i -e '/Acquire::http::Proxy/d' /etc/apt/apt.conf
15     # sudo sed -i -e '/Acquire::https::Proxy/d' /etc/apt/apt.conf
16 }
```



## ⓘ 注意

这里假设宿主机局域网 http 代理的端口是 10811。

## 4.3 卸载 Oh My Zsh

- 终端输入：

### ✓ Bash

```
1 uninstall_oh_my_zsh
2 Are you sure you want to remove Oh My Zsh? [y/N] Y
```

- 终端提示信息：

### ✓ Bash

```
1 Removing ~/.oh-my-zsh
2 Looking for original zsh config...
3 Found ~/.zshrc.pre-oh-my-zsh -- Restoring to ~/.zshrc
4 Found ~/.zshrc -- Renaming to ~/.zshrc.omz-uninstalled-20170820200007
5 Your original zsh config was restored. Please restart your session.
6 Thanks for trying out Oh My Zsh. It's been uninstalled.
```

## 4.4 手动更新 Oh My Zsh

- **Oh My Zsh** 的自动更新提示误触关掉了解决办法
- 打开终端输入：

### ✓ Bash

```
1 upgrade_oh_my_zsh
```

(本文完)



## 相关内容

- 在 Word 中使用 Zotero 插入参考文献
- Ubuntu 22.04 LTS 64位系统安装 Clang10 版本
- Ubuntu 22.04 安装 AFL 及 Qemu Mode 实践
- 使用PicGo + GitHub 搭建 Obsidian 图床
- PTATM (一)：初始化工作

