

# 酿造升级日志

==> 完成 ln -s ../../Cellar/azure-cli/2.68.0/etc/bash\_completion.d/az az ln -s ../../Cellar/azure-cli/2.68.0/bin/az az ln -s ../../Cellar/azure-cli/2.68.0/share/fish/vendor\_completions.d/az.fish az.fish ln -s ../../Cellar/azure-cli/2.68.0/share/zsh/site-functions/\_az \_az ==> 警告 zsh 自动补全已安装到: /opt/homebrew/share/zsh/site-functions ==> 摘要 ✘ /opt/homebrew/Cellar/azure-cli/2.68.0: 24,507 个文件, 580.4MB ==> 运行 brew cleanup azure-cli… 正在移除: /opt/homebrew/Cellar/azure-cli/2.67.0\_1…(27,401 个文件, 647.1MB) 正在移除: /Users/lzwjava/Library/Caches/Homebrew/azure-cli\_bottle\_manifest-2.67.0\_1…(22.5KB) 正在移除: /Users/lzwjava/Library/azure-cli-2.67.0\_1…(54MB) ==> 警告 ==> openjdk 为了让系统 Java 包装器找到此 JDK, 请使用以下命令创建符号链接: sudo ln -sf /opt/homebrew/opt/openjdk/libexec/openjdk.jdk /Library/Java/JavaVirtualMachines/openjdk.jdk

openjdk 是 keg-only 的, 这意味着它没有被符号链接到 /opt/homebrew, 因为 macOS 提供了类似的软件, 并且并行安装此软件可能会导致各种问题。

如果您需要将 openjdk 放在 PATH 的首位, 请运行: echo ‘export PATH=“/opt/homebrew/opt/openjdk/bin:\$PATH”’ » ~/.zshrc

为了让编译器找到 openjdk, 您可能需要设置: export CPPFLAGS= “-I/opt/homebrew/opt/openjdk/include” ==> ruby 默认情况下, gem 安装的二进制文件将放置在: /opt/homebrew/lib/ruby/gems/3.4.0/bin

您可能需要将其添加到 PATH 中。

ruby 是 keg-only 的, 这意味着它没有被符号链接到 /opt/homebrew, 因为 macOS 已提供此软件, 并且并行安装另一个版本可能会导致各种问题。

如果您需要将 ruby 放在 PATH 的首位, 请运行: echo ‘export PATH=“/opt/homebrew/opt/ruby/bin:\$PATH”’ » ~/.zshrc

为了让编译器找到 ruby, 您可能需要设置: export LDFLAGS= “-L/opt/homebrew/opt/ruby/lib” export CPPFLAGS= “-I/opt/homebrew/opt/ruby/include” ==> yt-dlp zsh 自动补全已安装到: /opt/homebrew/share/zsh/site-functions ==> redis 升级后重新启动 redis: brew services restart redis 或者, 如果您不需要后台服务, 可以直接运行: /opt/homebrew/opt/redis/bin/redis-server /opt/homebrew/etc/redis.conf ==> perl 默认情况下, 非 Homebrew cpan 模块安装到 Cellar。如果您希望模块在更新后仍然存在, 我们建议使用 local::lib。

您可以这样设置: PERL\_MM\_OPT=“INSTALL\_BASE=\$HOME/perl5”cpanlocal :: lib shell /profile ~/.zshrc -I\$HOME/perl5/lib/perl5 -Mlocal :: lib =\$HOME/perl5” ==> awscli “examples” 目录已安装到: /opt/homebrew/share/awscli/examples

zsh 自动补全和函数已安装到: /opt/homebrew/share/zsh/site-functions ==> php 要在 Apache 中启用 PHP, 请将以下内容添加到 httpd.conf 并重新启动 Apache: LoadModule php\_module /opt/homebrew/opt/php/lib/httpd/modules/libphp

```
<FilesMatch \.php$>
  SetHandler application/x-httpd-php
</FilesMatch>
```

最后, 检查 DirectoryIndex 是否包含 index.php DirectoryIndex index.php index.html

php.ini 和 php-fpm.ini 文件位于： /opt/homebrew/etc/php/8.4/

立即启动 php 并使其在登录时重新启动： brew services start php 或者， 如果您不需要后台服务， 可以直接运行：  
/opt/homebrew/opt/php/sbin/php-fpm –nodaemonize ==> nginx 文档根目录是： /opt/homebrew/var/www

默认端口已在 /opt/homebrew/etc/nginx/nginx.conf 中设置为 8080， 以便 nginx 可以无需 sudo 运行。

nginx 将加载 /opt/homebrew/etc/nginx/servers/ 中的所有文件。

升级后重新启动 nginx： brew services restart nginx 或者， 如果您不需要后台服务， 可以直接运行：  
/opt/homebrew/opt/nginx/bin/nginx -g daemon off; ==> git-lfs 更新您的 git 配置以完成安装：

```
# 更新全局 git 配置 $ git lfs install
```

```
# 更新系统 git 配置 $ git lfs install --system
```

zsh 自动补全已安装到： /opt/homebrew/share/zsh/site-functions ==> wireshark 此公式默认仅安装命令行实用程序。

使用 Homebrew Cask 安装 Wireshark.app： brew install --cask wireshark

如果您的可用捕获接口列表为空（macOS 默认行为），请安装 ChmodBPF： brew install --cask wireshark-chmodbpf  
==> doctl zsh 自动补全已安装到： /opt/homebrew/share/zsh/site-functions ==> azure-cli zsh 自动补全已安装到：  
/opt/homebrew/share/zsh/site-functions