

醸造アップグレードログ

```
==> まとめて 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 ==> 注意点 zsh 補完機能は以下にインストールされました: /opt/homebrew/share/zsh/site-functions ==> 要約 ✘ /opt/homebrew/Cellar/azure-cli/2.68.0: 24,507 ファイル, 580.4MB ==> brew cleanup azure-cli を実行しています…削除: /opt/homebrew/Cellar/azure-cli/2.67.0_1…(27,401 ファイル, 647.1MB) 削除: /Users/lzwjava/Library/Caches/Homebrew/azure-cli_bottle_manifest-2.67.0_1…(22.5KB) 削除: /Users/lzwjava/Library/Caches/Homebrew/azure-cli-2.67.0_1…(54MB) ==> 注意点 ==> openjdk システム Java ラッパーがこの JDK を検出するには、シンボリックリンクを作成します。sudo ln -sfn /opt/homebrew/opt/openjdk/libexec/openjdk.jdk /Library/Java/JavaVirtualMachines/openjdk.jdk
```

openjdk は keg-only です。つまり、macOS が同様のソフトウェアを提供しており、並行してインストールすると様々な問題が発生する可能性があるため、/opt/homebrew にシンボリックリンクされませんでした。

PATH の先頭に openjdk を配置する必要がある場合は、以下を実行してください: echo 'export PATH="/opt/homebrew/opt/openjdk/bin:\$PATH"' » ~/.zshrc

コンパイラで openjdk を使用するには、以下を設定する必要があります: export CPPFLAGS="-I/opt/homebrew/opt/openjdk/include" ==> ruby デフォルトでは、gem によってインストールされたバイナリは以下に配置されます: /opt/homebrew/lib/ruby/gems/3.4.0/bin

PATH に追加することをお勧めします。

ruby は keg-only です。つまり、macOS すでにこのソフトウェアが提供されているため、別のバージョンを並行してインストールすると様々な問題が発生する可能性があるため、/opt/homebrew にシンボリックリンクされませんでした。

PATH の先頭に ruby を配置する必要がある場合は、以下を実行してください: echo 'export PATH="/opt/homebrew/opt/ruby/bin:\$PATH"' » ~/.zshrc

コンパイラで ruby を使用するには、以下を設定する必要があります: export LDFLAGS="-L/opt/homebrew/opt/ruby/lib" export CPPFLAGS="-I/opt/homebrew/opt/ruby/include" ==> yt-dlp zsh 補完機能は以下にインストールされました: /opt/homebrew/share/zsh/site-functions ==> redis アップグレード後に redis を再起動するには: brew services restart redis または、バックグラウンドサービスを必要としない場合は、以下を実行してください: /opt/homebrew/opt/redis/bin/redis-server /opt/homebrew/etc/redis.conf ==> perl デフォルトでは、brewed ではない cpan モジュールは Cellar にインストールされます。モジュールを更新後も保持したい場合は、local::lib を使用することをお勧めします。

以下の手順で設定できます: PERL_MM_OPT="INSTALL_BASE=\$HOME/perl5" cpanlocal :: lib : ./profile/.zshrc -I\$HOME/perl5/lib/perl5 -Mlocal :: lib =\$HOME/perl5" ==> awscli "examples" ディレクトリは以下にインストールされました: /opt/homebrew/share/awscli/examples

zsh 補完機能と関数は以下にインストールされました: /opt/homebrew/share/zsh/site-functions ==> php Apache で PHP を有効にするには、httpd.conf に以下を追加して Apache を再起動してください: LoadModule php_module /opt/homebrew/opt/php/lib/httpd/modules/libphp.so

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

最後に、DirectoryIndex に index.php が含まれていることを確認してください。DirectoryIndex index.php
index.html

php.ini と php-fpm.ini ファイルは以下にあります: /opt/homebrew/etc/php/8.4/

php を今すぐ起動し、ログイン時に再起動するには: brew services start php または、バックグラウンドサービスを必要としない場合は、以下を実行してください: /opt/homebrew/opt/php/sbin/php-fpm -nodaemonize ==> nginx ドキュルートは:/opt/homebrew/var/www です。

デフォルトポートは/opt/homebrew/etc/nginx/nginx.conf で 8080 に設定されているため、sudo なしで nginx を実行できます。

nginx は /opt/homebrew/etc/nginx/servers/ 内のすべてのファイルをロードします。

アップグレード後に nginx を再起動するには: brew services restart nginx または、バックグラウンドサービスを必要としない場合は、以下を実行してください: /opt/homebrew/opt/nginx/bin/nginx -g daemon off; ==> git-lfs インストールを完了するには、git の設定を更新してください:

```
# グローバル git 設定の更新 $ git lfs install
```

```
# システム git 設定の更新 $ git lfs install --system
```

zsh 補完機能は以下にインストールされました: /opt/homebrew/share/zsh/site-functions ==> wireshark このフ
ォーミュラはデフォルトでコマンドラインユーティリティのみインストールします。

Homebrew Cask を使用して Wireshark.app をインストールします: brew install --cask wireshark

使用可能なキャプチャインターフェースのリストが空の場合 (macOS のデフォルトの動作)、ChmodBPF をインストールします: brew install --cask wireshark-chmodbpf ==> doctl zsh 補完機能は以下にインストールされました: /opt/homebrew/share/zsh/site-functions ==> azure-cli zsh 補完機能は以下にインストールされました: /opt/homebrew/share/zsh/site-functions