

Le Journal des Mises à Jour de Brew

==> Finalisation `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` ==> Avertissements Les complétions zsh ont été installées à : `/opt/homebrew/share/zsh/site-functions` ==> Résumé `/opt/homebrew/Cellar/azure-cli/2.68.0` : 24 507 fichiers, 580,4 Mo ==> Exécution de `brew cleanup azure-cli`...Suppression : `/opt/homebrew/Cellar/azure-cli/2.67.0_1...`(27 401 fichiers, 647,1 Mo) Suppression : `/Users/lzwjava/Library/Caches/Homebrew/azure-cli_bottle_manifest-2.67.0_1...`(22,5 Ko) Suppression : `/Users/lzwjava/Library/Caches/Homebrew/azure-cli-2.67.0_1...`(54 Mo) ==> Avertissements ==> openjdk Pour que les wrappers Java système trouvent ce JDK, créez un lien symbolique avec `sudo ln -sf /opt/homebrew/opt/openjdk/libexec/openjdk.jdk /Library/Java/JavaVirtualMachines/openjdk.jdk`

openjdk est keg-only, ce qui signifie qu'il n'a pas été lié symboliquement dans `/opt/homebrew`, car macOS fournit un logiciel similaire et l'installation de ce logiciel en parallèle peut causer toutes sortes de problèmes.

Si vous avez besoin d'avoir openjdk en premier dans votre PATH, exécutez : `echo 'export PATH="/opt/homebrew/opt/openjdk/bin:$PATH"' >> ~/.zshrc`

Pour que les compilateurs trouvent openjdk, vous devrez peut-être définir : `export CPPFLAGS="-I/opt/homebrew/opt/openjdk/include"` ==> ruby Par défaut, les binaires installés par gem seront placés dans : `/opt/homebrew/lib/ruby/gems/3.4.0/bin`

Vous voudrez peut-être ajouter ceci à votre PATH.

ruby est keg-only, ce qui signifie qu'il n'a pas été lié symboliquement dans `/opt/homebrew`, car macOS fournit déjà ce logiciel et l'installation d'une autre version en parallèle peut causer toutes sortes de problèmes.

Si vous avez besoin d'avoir ruby en premier dans votre PATH, exécutez : `echo 'export PATH="/opt/homebrew/opt/ruby/bin:$PATH"' >> ~/.zshrc`

Pour que les compilateurs trouvent ruby, vous devrez peut-être définir : `export LDFLAGS="-L/opt/homebrew/opt/ruby/lib" export CPPFLAGS="-I/opt/homebrew/opt/ruby/include"` ==> yt-dlp Les complétions zsh ont été installées à : `/opt/homebrew/share/zsh/site-functions` ==> redis Pour redémarrer redis après une mise à niveau : `brew services restart redis` Ou, si vous ne voulez/avez pas besoin d'un service d'arrière-plan, vous pouvez simplement exécuter : `/opt/homebrew/opt/redis/bin/redis-server /opt/homebrew/etc/redis.conf` ==> perl Par défaut, les modules cpan non brassés sont installés dans le Cellar. Si vous souhaitez que vos modules persistent entre les mises à jour, nous vous recommandons d'utiliser `local::lib`.

Vous pouvez configurer cela comme ceci : `PERL_MM_OPT="INSTALL_BASE=HOME/perl5" cpanlocal :: lib` Et ajoutez ce qui suit à votre profil shell, par exemple `/.profile` ou `/.zshrc` `eval "$(perl -IHOME/perl5/lib/perl5 cpanlocal :: lib)"` ==> awscli Le répertoire « exemples » a été installé à : `/opt/homebrew/share/awscli/examples`

Les complétions et fonctions zsh ont été installées à : `/opt/homebrew/share/zsh/site-functions` ==> php Pour activer PHP dans Apache, ajoutez ce qui suit à `httpd.conf` et redémarrez Apache : `LoadModule php_module /opt/homebrew/opt/php/lib/httpd/modules/libphp.so`

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

Enfin, vérifiez que DirectoryIndex inclut index.php DirectoryIndex index.php index.html

Le fichier php.ini et php-fpm.ini se trouvent dans : /opt/homebrew/etc/php/8.4/

Pour démarrer php maintenant et redémarrer au démarrage : brew services start php Ou, si vous ne voulez/avez pas besoin d'un service d'arrière-plan, vous pouvez simplement exécuter : /opt/homebrew/opt/php/sbin/php-fpm -nodaemonize ==> nginx Le répertoire racine des documents est : /opt/homebrew/var/www

Le port par défaut a été défini dans /opt/homebrew/etc/nginx/nginx.conf sur 8080 afin que nginx puisse fonctionner sans sudo.

nginx chargera tous les fichiers dans /opt/homebrew/etc/nginx/servers/.

Pour redémarrer nginx après une mise à niveau : brew services restart nginx Ou, si vous ne voulez/avez pas besoin d'un service d'arrière-plan, vous pouvez simplement exécuter : /opt/homebrew/opt/nginx/bin/nginx -g daemon off; ==> git-lfs Mettez à jour votre configuration git pour terminer l'installation :

Mettez à jour la configuration git globale \$ git lfs install

Mettez à jour la configuration git système \$ git lfs install --system

Les complétions zsh ont été installées à : /opt/homebrew/share/zsh/site-functions ==> wireshark Cette formule n'installe que les utilitaires en ligne de commande par défaut.

Installez Wireshark.app avec Homebrew Cask : brew install --cask wireshark

Si votre liste d'interfaces de capture disponibles est vide (comportement macOS par défaut), installez ChmodBPF : brew install --cask wireshark-chmodbpf ==> doctl Les complétions zsh ont été installées à : /opt/homebrew/share/zsh/site-functions ==> azure-cli Les complétions zsh ont été installées à : /opt/homebrew/share/zsh/site-functions