

The Log of Brew Upgrade

==> Finishing up

```
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
```

==> Caveats

zsh completions have been installed to:

```
/opt/homebrew/share/zsh/site-functions
```

==> Summary

```
/opt/homebrew/Cellar/azure-cli/2.68.0: 24,507 files, 580.4MB
```

==> Running `brew cleanup azure-cli`...

```
Removing: /opt/homebrew/Cellar/azure-cli/2.67.0_1... (27,401 files, 647.1MB)
```

```
Removing: /Users/lzwjava/Library/Caches/Homebrew/azure-cli_bottle_manifest--2.67.0_1... (22.5KB)
```

```
Removing: /Users/lzwjava/Library/Caches/Homebrew/azure-cli--2.67.0_1... (54MB)
```

==> Caveats

==> openjdk

For the system Java wrappers to find this JDK, symlink it with

```
sudo ln -sfn /opt/homebrew/opt/openjdk/libexec/openjdk.jdk /Library/Java/JavaVirtualMachines/openjdk.jdk
```

openjdk is keg-only, which means it was not symlinked into /opt/homebrew, because macOS provides similar software and installing this software in parallel can cause all kinds of trouble.

If you need to have openjdk first in your PATH, run:

```
echo 'export PATH="/opt/homebrew/opt/openjdk/bin:$PATH"' >> ~/.zshrc
```

For compilers to find openjdk you may need to set:

```
export CPPFLAGS="-I/opt/homebrew/opt/openjdk/include"
```

==> ruby

By default, binaries installed by gem will be placed into:

```
/opt/homebrew/lib/ruby/gems/3.4.0/bin
```

You may want to add this to your PATH.

ruby is keg-only, which means it was not symlinked into /opt/homebrew, because macOS already provides this software and installing another version in parallel can cause all kinds of trouble.

If you need to have ruby first in your PATH, run:

```
echo 'export PATH="/opt/homebrew/opt/ruby/bin:$PATH"' >> ~/.zshrc
```

For compilers to find ruby you may need to set:

```
export LDFLAGS="-L/opt/homebrew/opt/ruby/lib"
export CPPFLAGS="-I/opt/homebrew/opt/ruby/include"
```

```
==> yt-dlp
```

zsh completions have been installed to:

```
/opt/homebrew/share/zsh/site-functions
```

```
==> redis
```

To restart redis after an upgrade:

```
brew services restart redis
```

Or, if you don't want/need a background service you can just run:

```
/opt/homebrew/opt/redis/bin/redis-server /opt/homebrew/etc/redis.conf
```

```
==> perl
```

By default non-brewed cpan modules are installed to the Cellar. If you wish for your modules to persist across updates we recommend using `local::lib`.

You can set that up like this:

```
PERL_MM_OPT="INSTALL_BASE=$HOME/perl5" cpan local::lib
```

And add the following to your shell profile e.g. ~/.profile or ~/.zshrc

```
eval "$(perl -I$HOME/perl5/lib/perl5 -Mlocal::lib=$HOME/perl5)"
```

```
==> awscli
```

The "examples" directory has been installed to:

```
/opt/homebrew/share/awscli/examples
```

zsh completions and functions have been installed to:

```
/opt/homebrew/share/zsh/site-functions
```

```
==> php
```

To enable PHP in Apache add the following to httpd.conf and restart Apache:

```
LoadModule php_module /opt/homebrew/opt/php/lib/httpd/modules/libphp.so
```

```
<FilesMatch \.php$>
```

```
    SetHandler application/x-httpd-php
```

```
</FilesMatch>
```

Finally, check DirectoryIndex includes index.php

```
DirectoryIndex index.php index.html
```

The php.ini and php-fpm.ini file can be found in:

```
/opt/homebrew/etc/php/8.4/
```

To start php now and restart at login:

```
brew services start php
```

Or, if you don't want/need a background service you can just run:

```
/opt/homebrew/opt/php/sbin/php-fpm --nodaemonize
```

```
==> nginx
```

Docroot is: /opt/homebrew/var/www

The default port has been set in /opt/homebrew/etc/nginx/nginx.conf to 8080 so that nginx can run without sudo.

nginx will load all files in /opt/homebrew/etc/nginx/servers/.

To restart nginx after an upgrade:

```
brew services restart nginx
```

Or, if you don't want/need a background service you can just run:

```
/opt/homebrew/opt/nginx/bin/nginx -g daemon\ off\;
```

```
==> git-lfs
```

Update your git config to finish installation:

```
# Update global git config
```

```
$ git lfs install
```

```
# Update system git config
```

```
$ git lfs install --system
```

zsh completions have been installed to:

```
/opt/homebrew/share/zsh/site-functions
```

```
==> wireshark
```

This formula only installs the command-line utilities by default.

Install Wireshark.app with Homebrew Cask:

```
brew install --cask wireshark
```

If your list of available capture interfaces is empty

(default macOS behavior), install ChmodBPF:

```
brew install --cask wireshark-chmodbpf
```

```
==> doctl
```

zsh completions have been installed to:

```
/opt/homebrew/share/zsh/site-functions  
==> azure-cli  
zsh completions have been installed to:  
/opt/homebrew/share/zsh/site-functions
```