# Following the noteapp-react-amplify tutorial

## Prerequisites

Starting this tutorial, I began with the prerequisites, which include installing node.js+npm, create-react-app, and aws-amplify CLI tools.

### Node.js+NPM: A Flat Brew

Generally, my preferred solution on the mac is to use ‘brew’ to manage installation of build tools and the like. In this case however, brew fizzled. I encountered two problems with Brew.

First, there seemed to be a permanent, previous installation of node tools on my mac machine that were locked down by root:wheel ownership which prevented the brew install from forming the proper links (i.e. most brew installations are performed within the ‘cellar’ directory, and then symbolically linked to the required directories in the macOS directories, e.g. /usr/local/include, etc). This first issue was resolved (unsatisfyingly) by changing all the folder permissions to ‘whoami’:admin, and allowing brew to overwrite existing files. In the end, this allowed the installation to complete.

Second, although both Node.js and npm seemed properly installed, there appeared to be a mismatch between the NPM and node.js versioning. This manifested itself when I tried to use npm to install the aws-amplify cli tools:

npm install -g @aws-amplify/cli

which resulted in:

npm WARN npm npm does not support Node.js v11.14.0

npm WARN npm You should probably upgrade to a newer version of node as we

npm WARN npm can't make any promises that npm will work with this version.

npm WARN npm Supported releases of Node.js are the latest release of 4, 6, 7, 8, 9.

npm WARN npm You can find the latest version at https://nodejs.org/

npm WARN deprecated kleur@2.0.2: Please upgrade to kleur@3 or migrate to 'ansi-colors' if you prefer the old syntax. Visit <https://github.com/lukeed/kleur/releases/tag/v3.0.0\> for migration path(s).

npm WARN notice [SECURITY] lodash has the following vulnerabilities: 1 moderate, 1 low. Go here for more details: https://nodesecurity.io/advisories?search=lodash&version=3.2.0 - Run `npm i npm@latest -g` to upgrade your npm version, and then `npm audit` to get more info.

npm WARN deprecated minimatch@2.0.10: Please update to minimatch 3.0.2 or higher to avoid a RegExp DoS issue

npm WARN notice [SECURITY] lodash has the following vulnerability: 1 low. Go here for more details: https://nodesecurity.io/advisories?search=lodash&version=3.5.0 - Run `npm i npm@latest -g` to upgrade your npm version, and then `npm audit` to get more info.

npm WARN notice [SECURITY] handlebars has the following vulnerability: 1 high. Go here for more details: https://nodesecurity.io/advisories?search=handlebars&version=4.0.11 - Run `npm i npm@latest -g` to upgrade your npm version, and then `npm audit` to get more info.

/usr/local/Cellar/node/11.14.0\_1/bin/amplify -> /usr/local/Cellar/node/11.14.0\_1/lib/node\_modules/@aws-amplify/cli/bin/amplify

> fsevents@1.2.8 install /usr/local/Cellar/node/11.14.0\_1/lib/node\_modules/@aws-amplify/cli/node\_modules/fsevents

> node install

node-pre-gyp WARN Using request for node-pre-gyp https download

[fsevents] Success: "/usr/local/Cellar/node/11.14.0\_1/lib/node\_modules/@aws-amplify/cli/node\_modules/fsevents/lib/binding/Release/node-v67-darwin-x64/fse.node" is installed via remote

Suffice it to say, the salient test was to test amplify with a version check which failed.

amplify --version

-bash: amplify: command not found

With this I resolved to start from scratch, i.e. use brew to uninstall node.js+npm, and start again with the mac binaries from <https://nodejs.org/en/>. Fortunately, the binary installation was quick and accomplished without a problem. The version checks with node and npm also successfully completed.

### Back to aws-amplify

This time, with node.js ‘successfully’ installed, the aws-amplify installation was pretty straightforward, despite a small hiccup. The first time around, aws-amplify completed mostly, however at the very end, I received the following:

npm WARN checkPermissions Missing write access to /usr/local/lib/node\_modules

npm ERR! path /usr/local/lib/node\_modules

npm ERR! code EACCES

npm ERR! errno -13

npm ERR! syscall access

npm ERR! Error: EACCES: permission denied, access '/usr/local/lib/node\_modules'

npm ERR! { [Error: EACCES: permission denied, access '/usr/local/lib/node\_modules']

npm ERR! stack:

npm ERR! 'Error: EACCES: permission denied, access \'/usr/local/lib/node\_modules\'',

npm ERR! errno: -13,

npm ERR! code: 'EACCES',

npm ERR! syscall: 'access',

npm ERR! path: '/usr/local/lib/node\_modules' }

npm ERR!

npm ERR! The operation was rejected by your operating system.

npm ERR! It is likely you do not have the permissions to access this file as the current user

npm ERR!

npm ERR! If you believe this might be a permissions issue, please double-check the

npm ERR! permissions of the file and its containing directories, or try running

npm ERR! the command again as root/Administrator (though this is not recommended).

npm ERR! A complete log of this run can be found in:

npm ERR! /Users/adam/.npm/\_logs/2019-04-27T13\_28\_53\_916Z-debug.log

The fix here turned out rather simple: changing user permissions to the /user/local/lib/node\_modules folder from root:wheel to ‘whoami’:admin. Retrying the installation, the installation worked flawlessly. Version check succeeded, showing installation of 1.6.5.

### Finally, create-react-app

Breaking away from the tutorial page, I used the instructions here (<https://facebook.github.io/create-react-app/docs/getting-started>) to create a new ‘noteapp’ directory. This is a slight break from the tutorial, where the assumption is that ‘create-react-app’ is preinstalled via NPM. Facebook’s updated approach recommends creating the app directory using npx so that the latest version of ‘create-react-app’ is used each time a new app is created.

Amplify Initial Configuration

Here the tutorial was uncomplicated to follow, wherein the steps instructed were exactly mirrored in practice. I downloaded the resulting credentials file (i.e. the file created by create the new IAM user for noteapp-react-amplify) to the top level of my ‘Projects’ directory so that it would not be added to the git repository.