# Edge Delivery Services Document-based Local Setup

## A document-based Edge Delivery Services site requires the use of Github to host the codebase and a cloud storage to host the documents.

This tutorial will take you through the steps to setup of setting up your IDE, a Github Repo and a SharePoint storage.

While different Github Repos and cloud storage locations may differ from project to project, the fundamental steps are the same.

Credit to the AEM.live Team for providing portions of the guide. Source: <https://www.aem.live/developer/tutorial>

## **Prerequisites**

Please ensure that you have the following accounts and software installed on the machine that you will be using during the workshop.

1. GitHub account: <https://github.com/signup>
2. SharePoint account: Use your Adobe one
3. Node.js and npm: <https://nodejs.org/en/download/package-manager>
4. Visual Studio Code application: <https://code.visualstudio.com/download>
5. Google Chrome application: <https://www.google.com/intl/en_ca/chrome/>
6. AEM Sidekick Browser Extension: <https://chromewebstore.google.com/detail/aem-sidekick/ccfggkjabjahcjoljmgmklhpaccedipo?pli=1>
7. AEM Command Line Interface <https://www.npmjs.com/package/@adobe/aem-cli>
   1. Run npm install -g @adobe/aem-cli in the Terminal App

This tutorial uses macOS, Chrome, and Visual Studio Code as the development environment and the screenshots and instructions reflect that setup. You can use a different operating system, browser, and code editor, but the UI you see and steps you must take may vary accordingly.

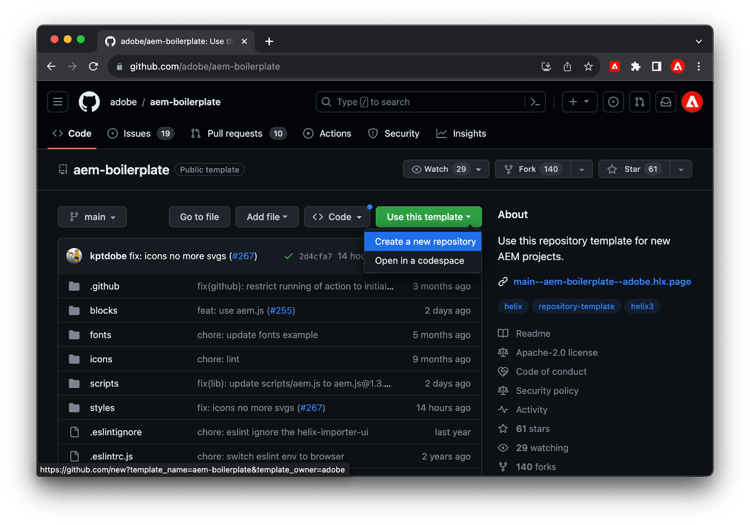
## 

## **Clone the AEM Boilerplate (fundamental step)**

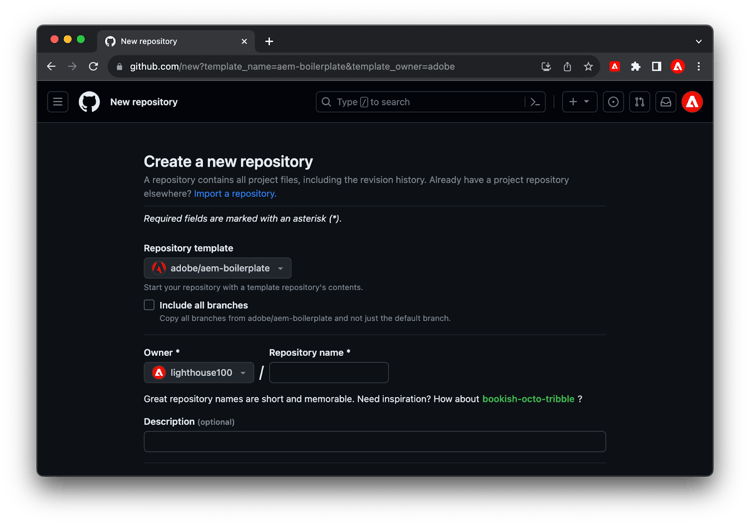
## 

| **Video** |
| --- |
| <https://www.hlx.live/developer/videos/tutorial-step1.mp4> |

The fastest and easiest way to get started following AEM best practices is to create your repository using the Boilerplate GitHub repository as a template. Even for new projects, it is recommended to start with the Boilerplate.   
  
Open <https://github.com/adobe/aem-boilerplate>

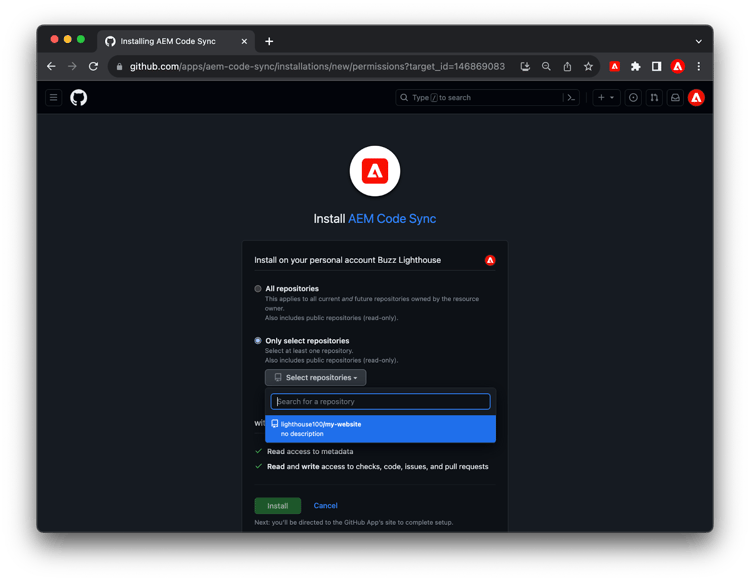


Just click the Use this template button and select Create a new repository, and select where you want to create this repository.



Set the repo to public.

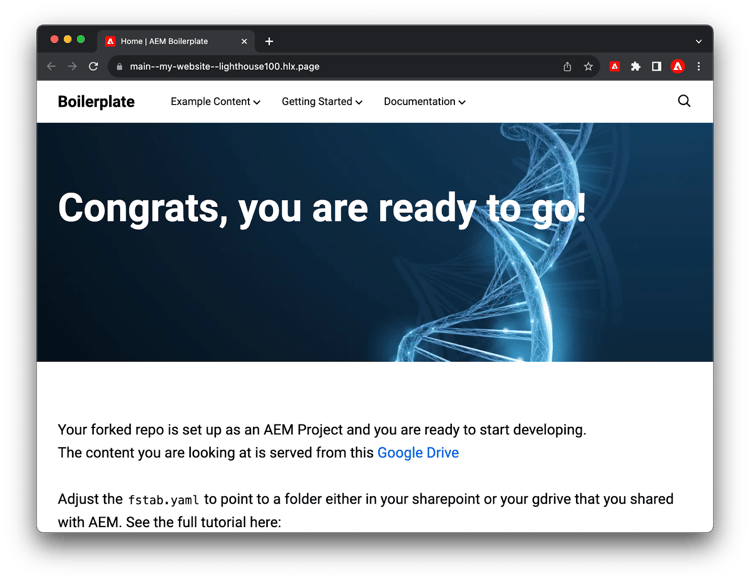
The only remaining step in GitHub is to install the [AEM Code Sync GitHub App](https://github.com/apps/aem-code-sync) on your repository by visiting this link:<https://github.com/apps/aem-code-sync/installations/new>



In the Repository access settings of the AEM Code Sync App, make sure you select Only select Repositories (not All Repositories). Then select your newly created repository, and click Save.

Congratulations! You have a new website running on https://<branch>--<repo>--<owner>.hlx.page/ You will determine this URL based on your git repositor.

<branch> = “main”  
<repo> = name of your git repo (all lower case)  
<owner> = git account user name

  
  
In the example above that’s https://main--my-website--lighthouse100.hlx.page/  
 

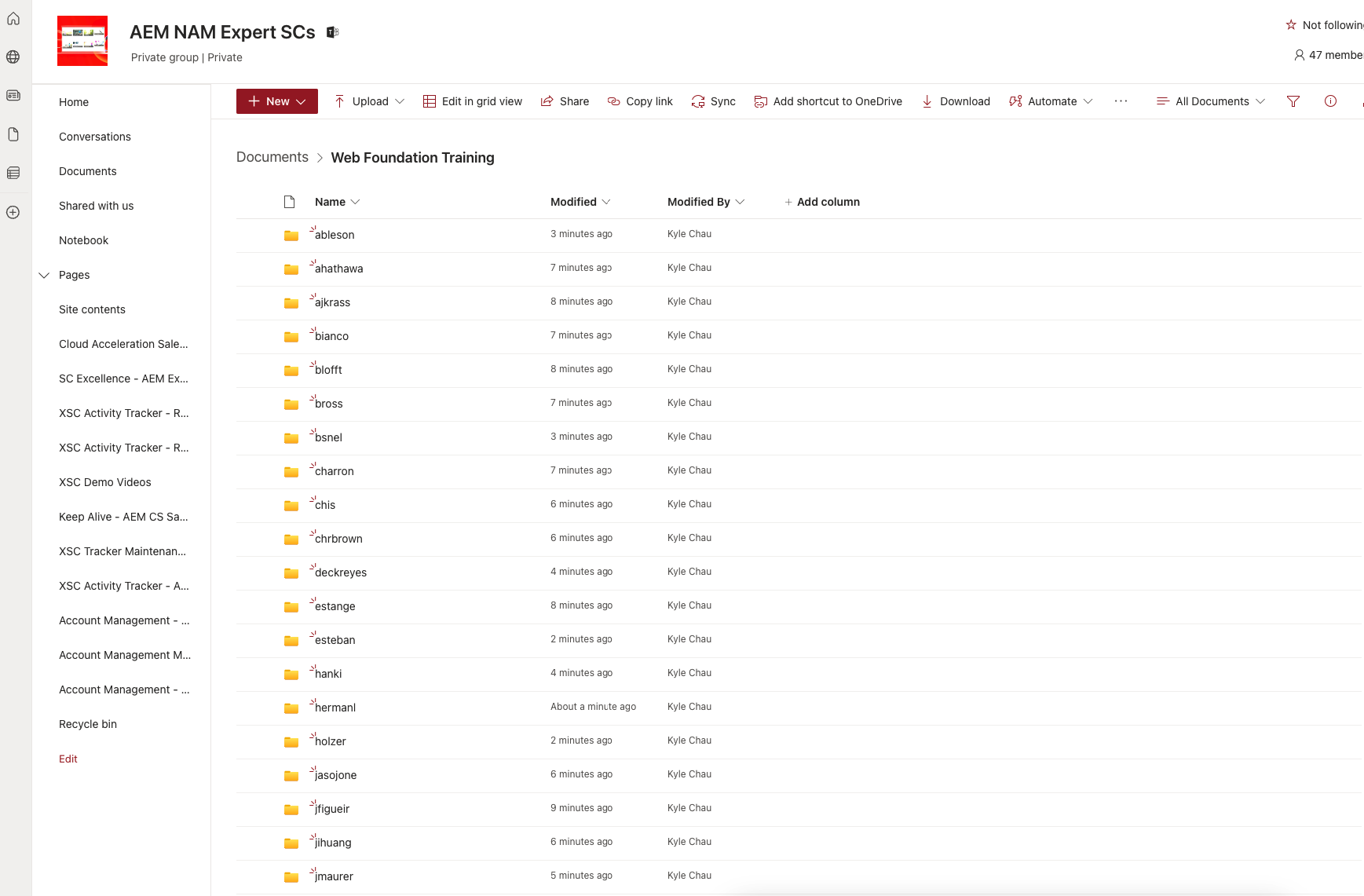
## 

## **Document Repository Setup (SharePoint)**

The following steps will use SharePoint.

Sign into Sharepoint <https://adobe.sharepoint.com/:f:/r/sites/AEMNAMExpertSCs/Shared%20Documents/Web%20Foundation%20Training?csf=1&web=1&e=Mn8Xod> using your Adobe Credentials.

This SharePoint site was created specifically for this workshop. Inside, you will find a folder named after your user ID. This is where you will create additional folders to store your document-based content. If you don't see a folder already created for you, create one yourself.

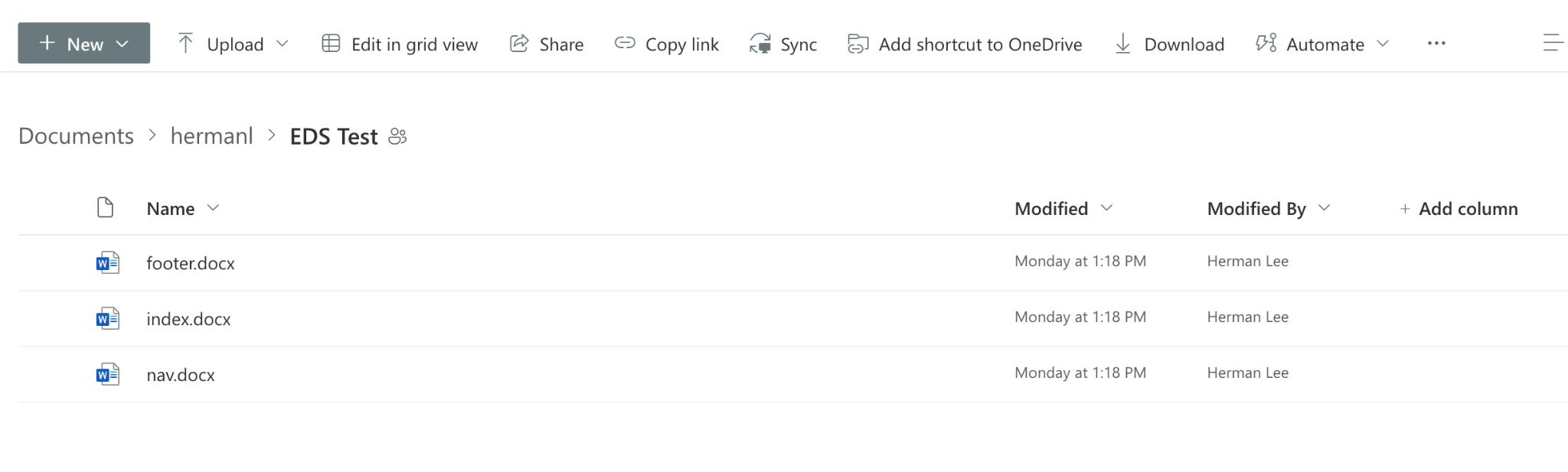


Inside your named folder, create another folder to hold the content for a test site, and name it EDS Test. Since multiple sites will source content from the main folder, organizing your folders is key.



Open the EDS Test folder. Add the starter content for your site by creating three separate Word documents. ***NOTE: please make sure that the filename are in lower case.***

1. index.docx - the content for the home page of your site
2. footer.docx - the footer content of your site
3. nav.docx - the navigation menu content of your site



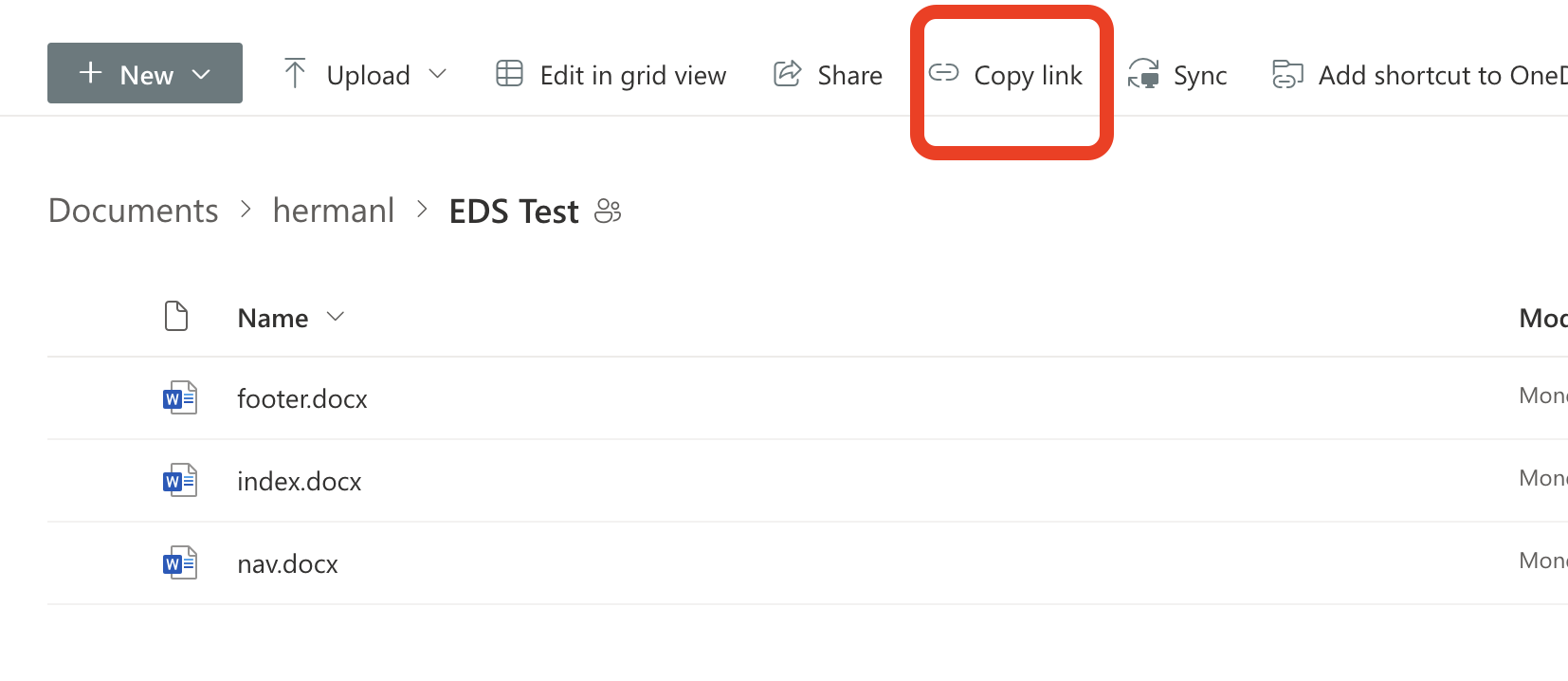
Open index.docx and add any text to the document. Hello World or Test Page will suffice.

Next you will share the folder with the Helix Bot. Doing so will allow the bot to access the folder and load it as pages of your site.

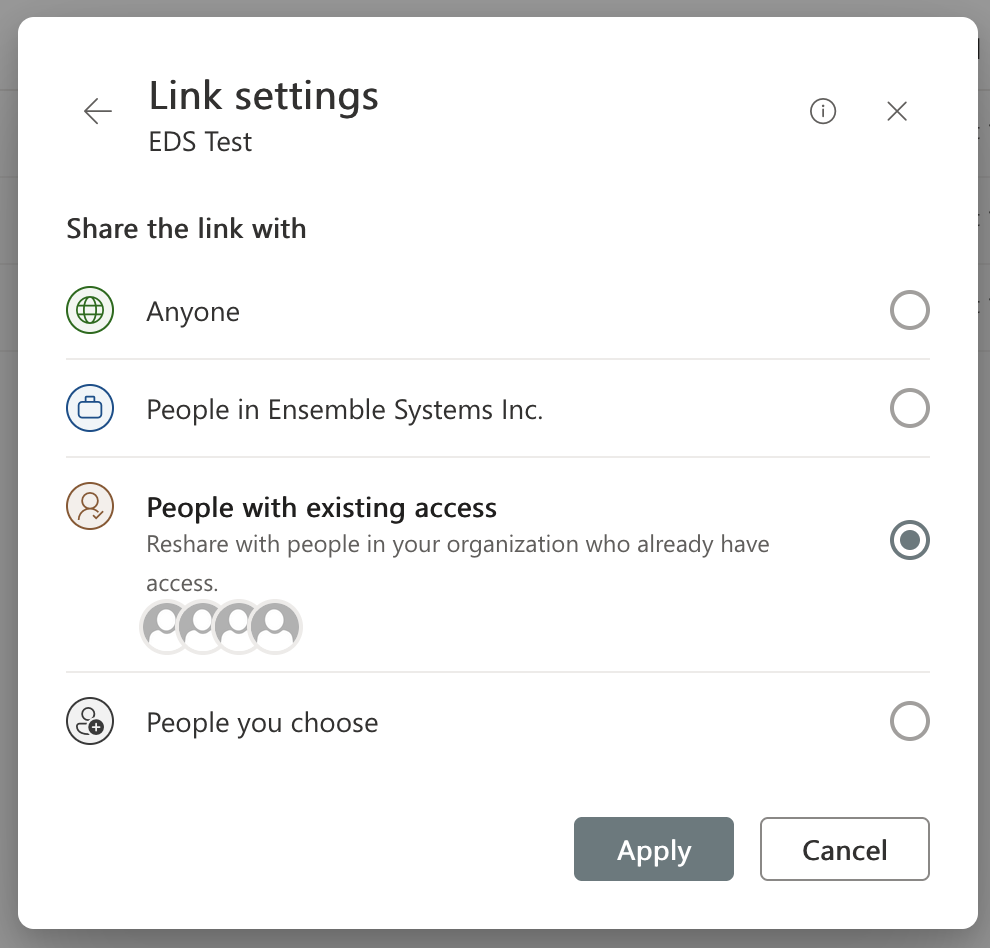
Be in the EDS Test folder. Click the Share button and share the folder with [helix@adobe.com](mailto:helix@adobe.com). (fundamental step)



Finally, copy the link to the folder, making sure that nothing in the folder is selected. Select the Copy link button.



Click the Settings icon and select Only people with the existing access.



If the dialog does not say People with existing access can use the link, try again.

## 

The link should resemble

<https://adobe.sharepoint.com/:f:/r/sites/AEMNAMExpertSCs/Shared%20Documents/Web%20Foundation%20Training/><username>/EDS Test?csf=1&web=1&e=Utyfso

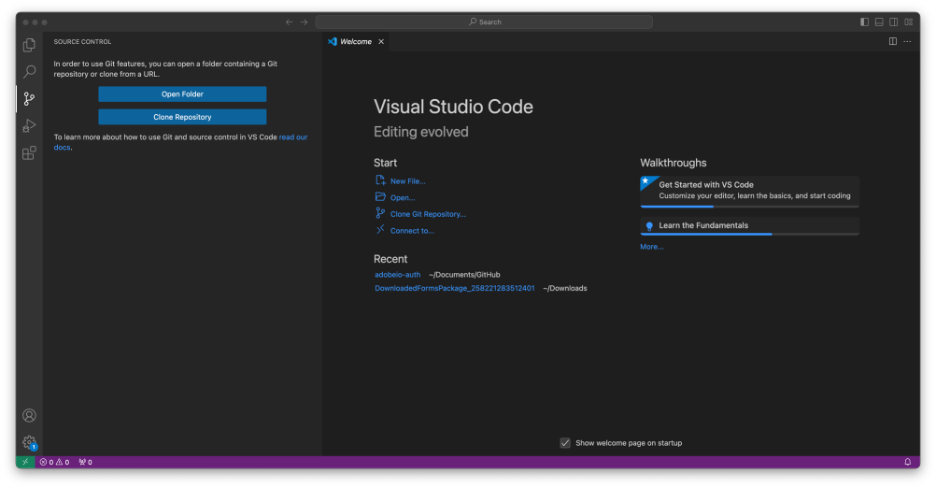
This is the document repo link that will go into the fstab.yaml file. More on that later.

**Visual Studio Code Setup**

Next, Git (source control) will be added to your machine. If you have ever installed xCode or Git before you will not need to do this.

Open Visual Studio Code and click on the Source Control icon located on the left rail.

If you see the Clone Repository button, then your computer does have Git installed and you may skip the steps to install Git.



If you see the Download Git for macOS button, then your computer does not have Git installed.



Installing Git Steps:

We will be following the instructions provided by the Download Git for macOS button.

Open the Terminal application and paste in /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)". Press the Enter key on your keyboard. Agree to the prompts to install Homebrew. Homebrew is a package manager for Mac. This might take more than a couple of minutes. Let it run until completion.

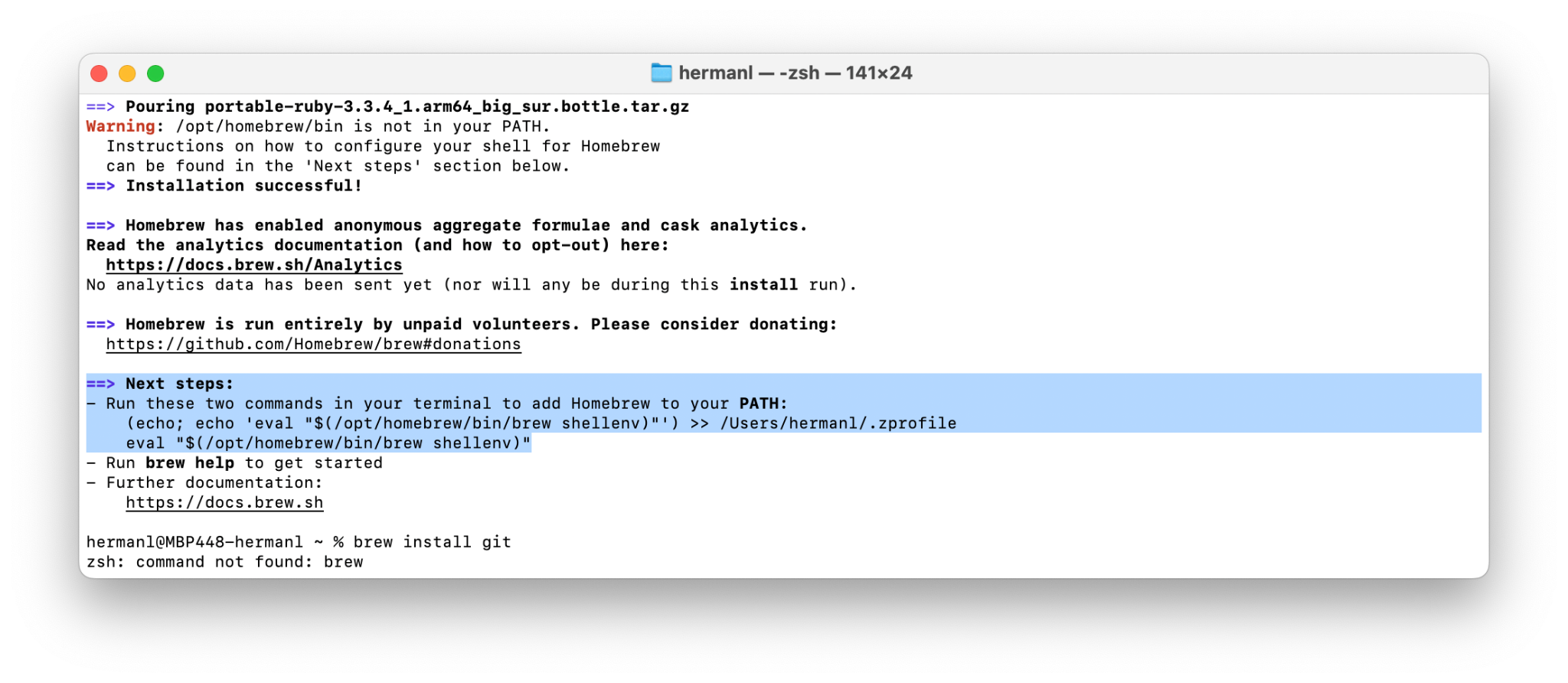


In the same Terminal window, paste in

(echo; echo 'eval "$(/opt/homebrew/bin/brew shellenv)"') >> /Users/hermanl/.zprofile

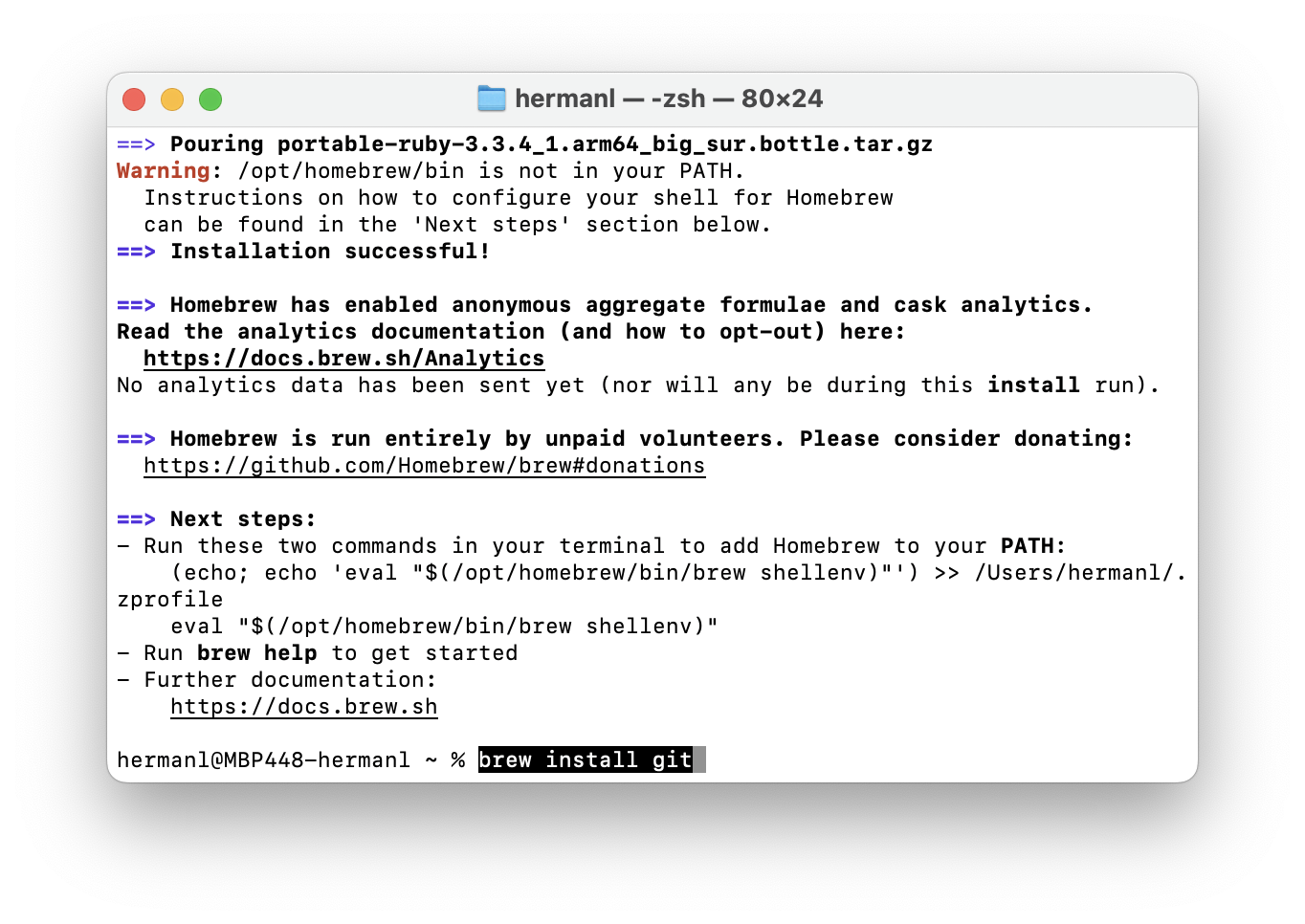
eval "$(/opt/homebrew/bin/brew shellenv)"v

Press the Enter key on your keyboard. This will add the brew command to your PATH so brew commands can be executed in the terminal. (You can also copy the two lines from the Terminal instead of copying from above)



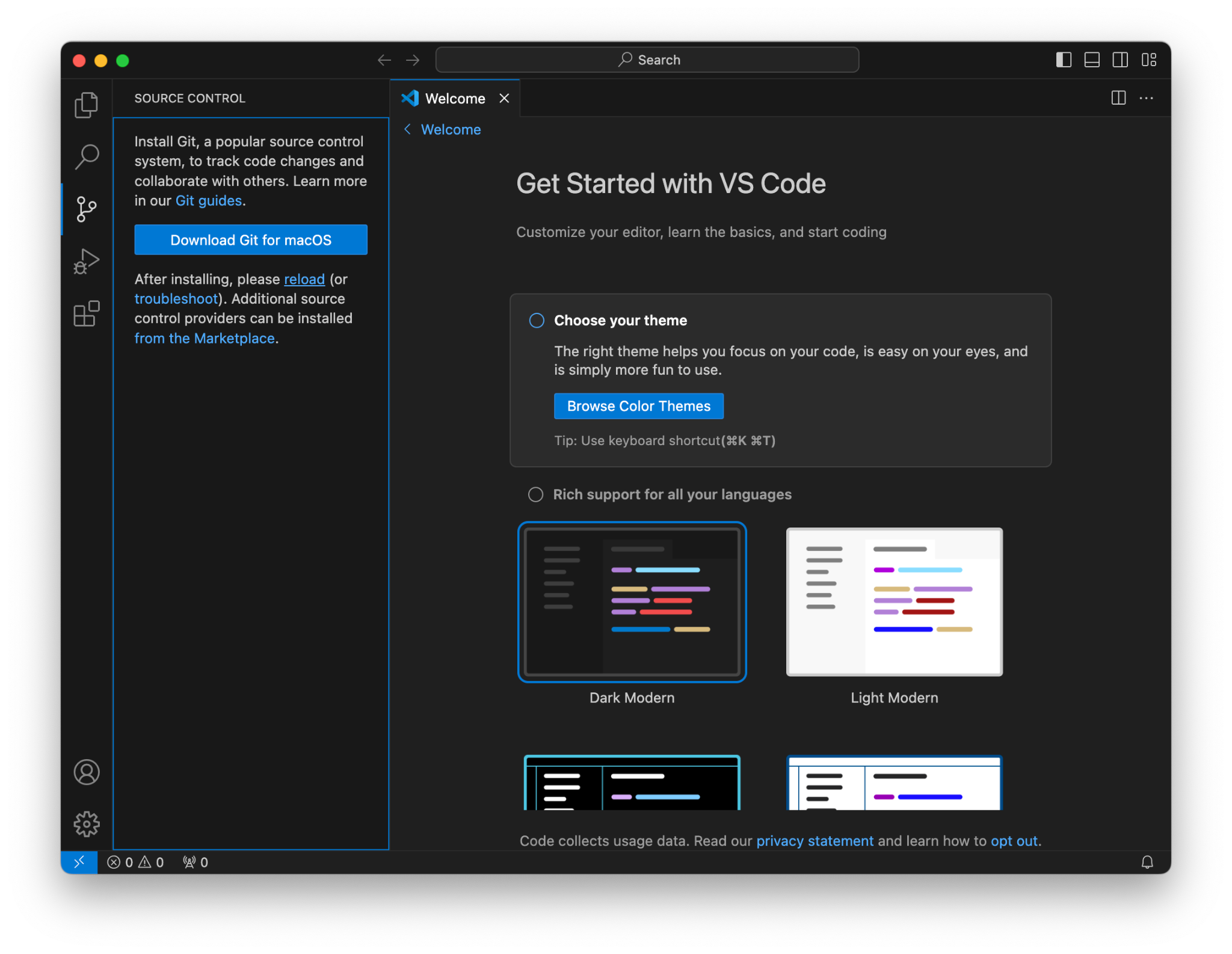
In the same Terminal window, paste in

brew install git

Press the Enter key on your keyboard.

When that has finished, type git -v into the Terminal. Confirm that git is working by checking for a version number.

Go back into Visual Studio Code and click on the reload link.



Git has been installed.

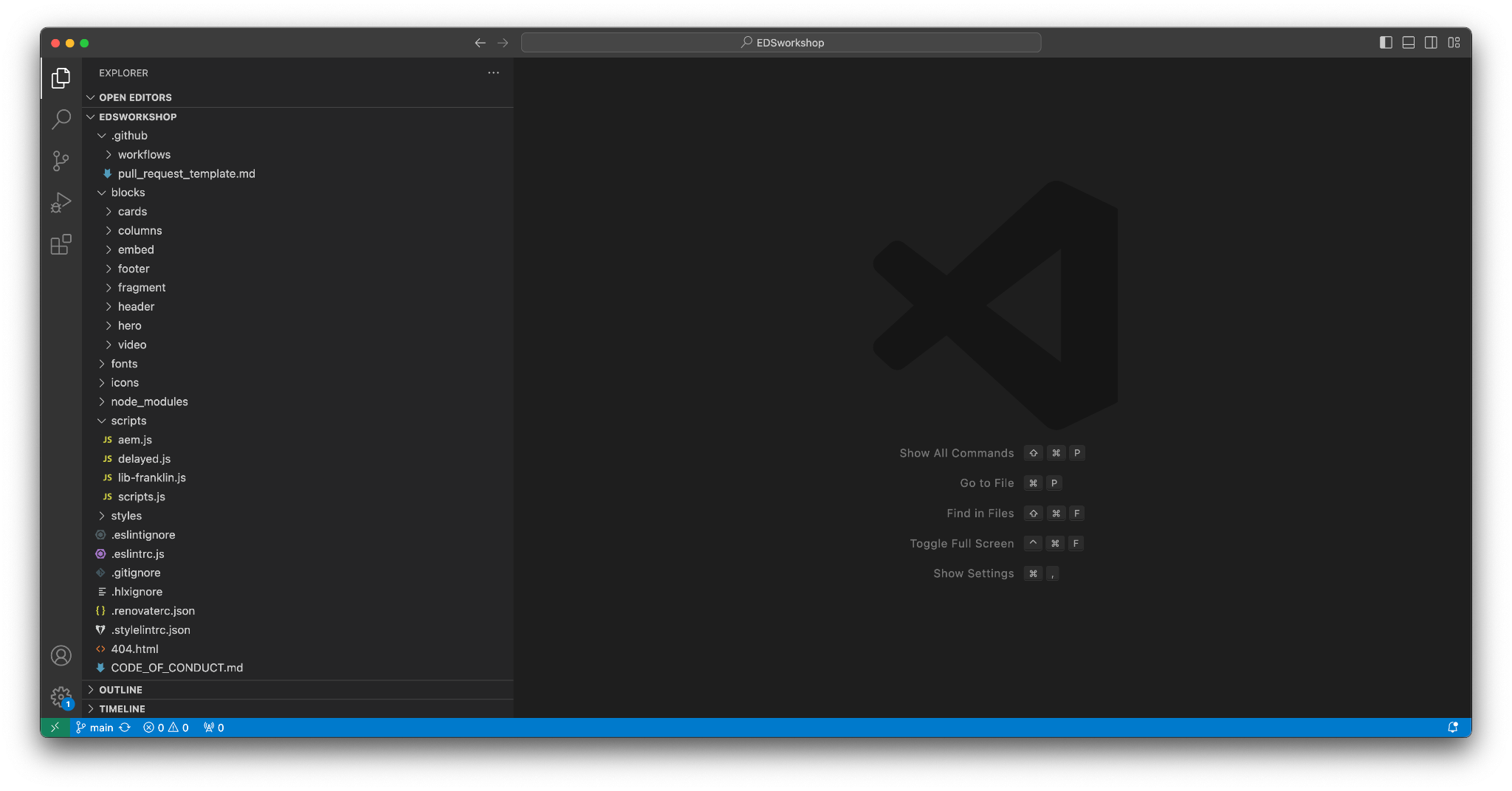
Click the Clone Repository button and paste the URL of your GIthub repo into the field.



It should be <https://github.com/><username>/<repo name>. In the example above, it is <https://github.com/hermanl/EDSworkshop>.

Select a location on your computer to save your code base.

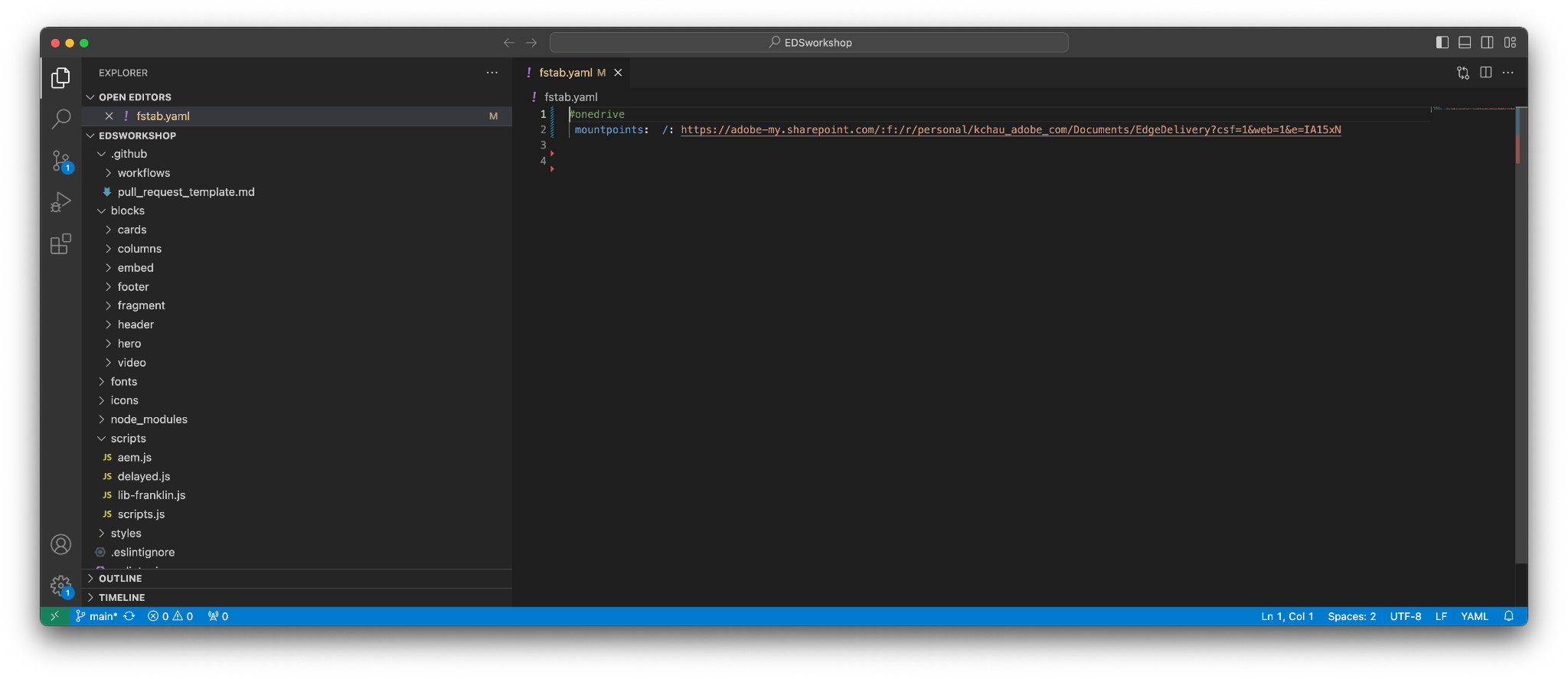
Once completed, you should see that your files are loaded in the Primary Side Bar.



## Link your own content source

Now that you have your content, you need to connect that content to your GitHub repo. You do this by changing the reference in fstab.yaml in your GitHub repo to the folder you just shared.

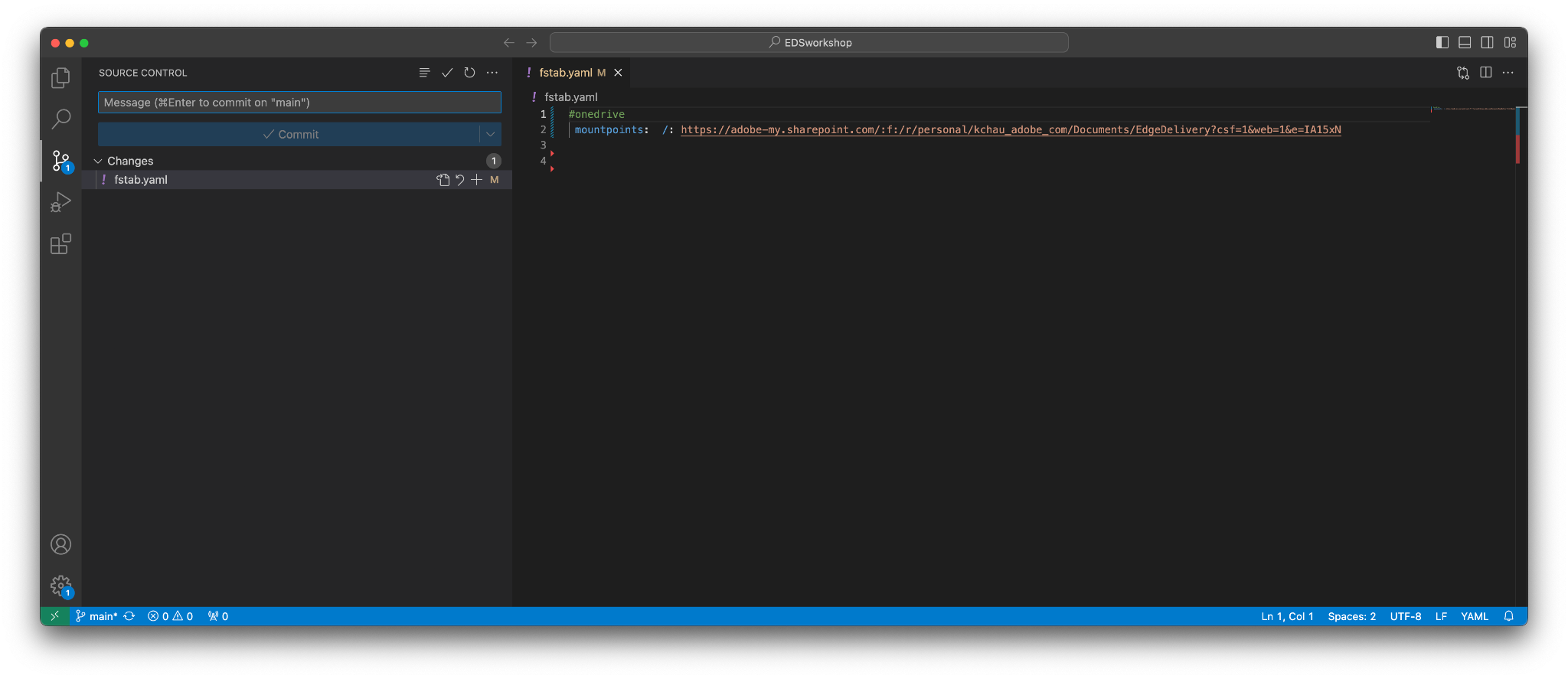
Open your codebase in Visual Studio Code. Open the fstab.yaml file.



Replace the existing URL with your document repository link. This link should open to the folder that contains the index.docx file. By doing this, you are instructing the code where to find the folder that houses all the document-based content for the site.

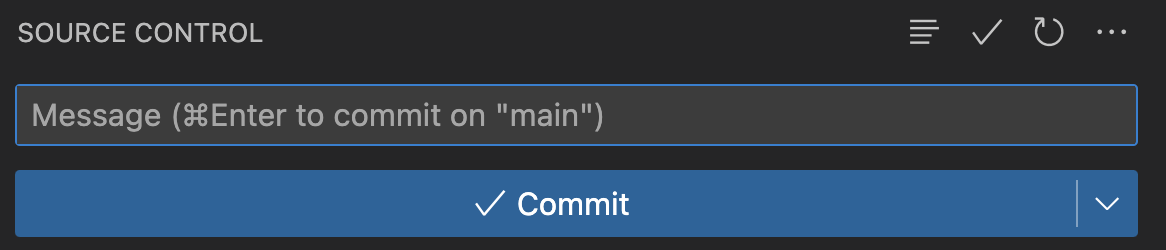
Typically, each site points to a single mount point. In this case, it is the EDS Test folder created in a previous section.

Next, you will commit the change to the Github repo. Select the Source Control button in the left rail.

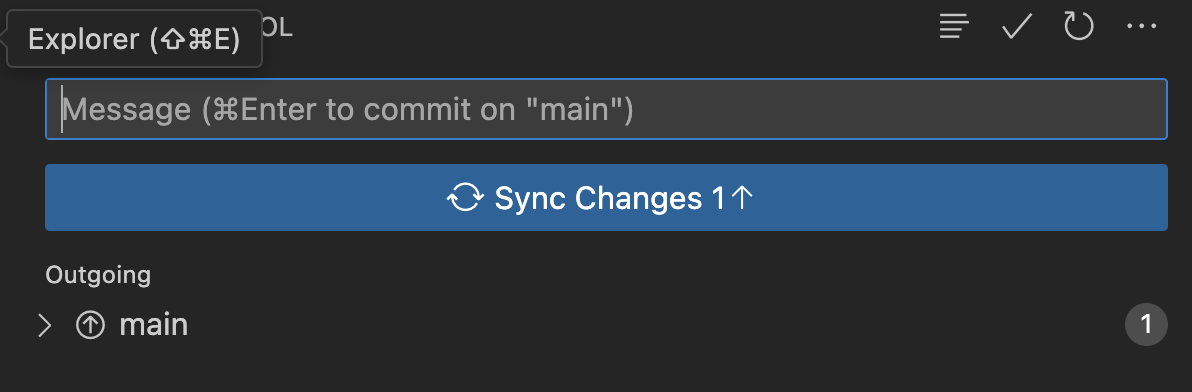


The next set of steps may differ depending on how your settings are.

Select the + button to the right of the fstab.yaml row. Enter a git commit message such as “Updated to point to SharePoint” into the field.



Select the Commit button.



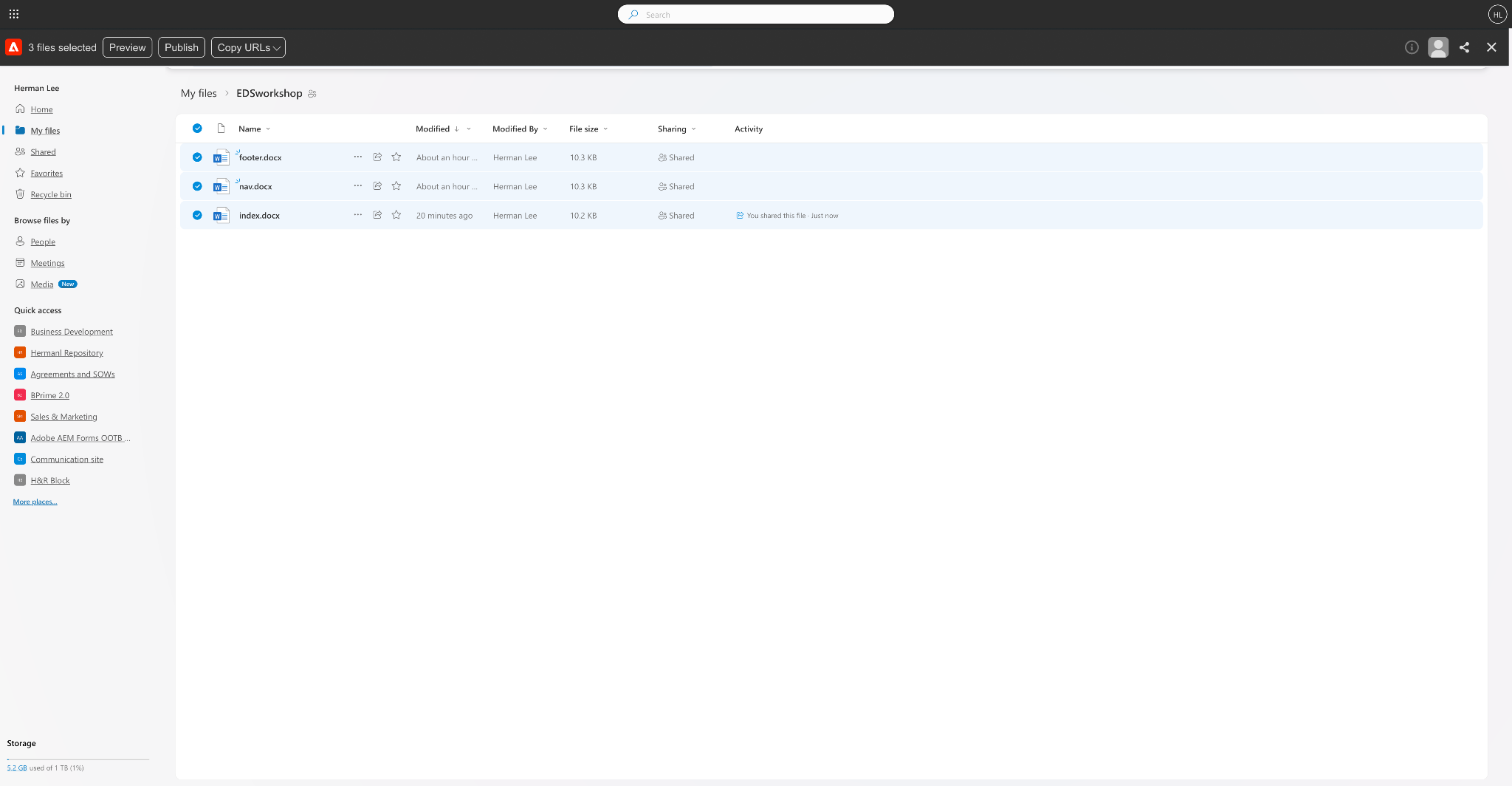
Enter a git commit message such as “Updated to point to SharePoint” into the field.

Select the Sync Changes button.

## Publish your content to the Preview Site

Open the newly created folder in SharePoint with the 3 Word documents. Select them all.

The AEM Sidekick should appear as a bar across the top of the page. Select the Preview button. Accept the prompts to publish the content to the Preview site.

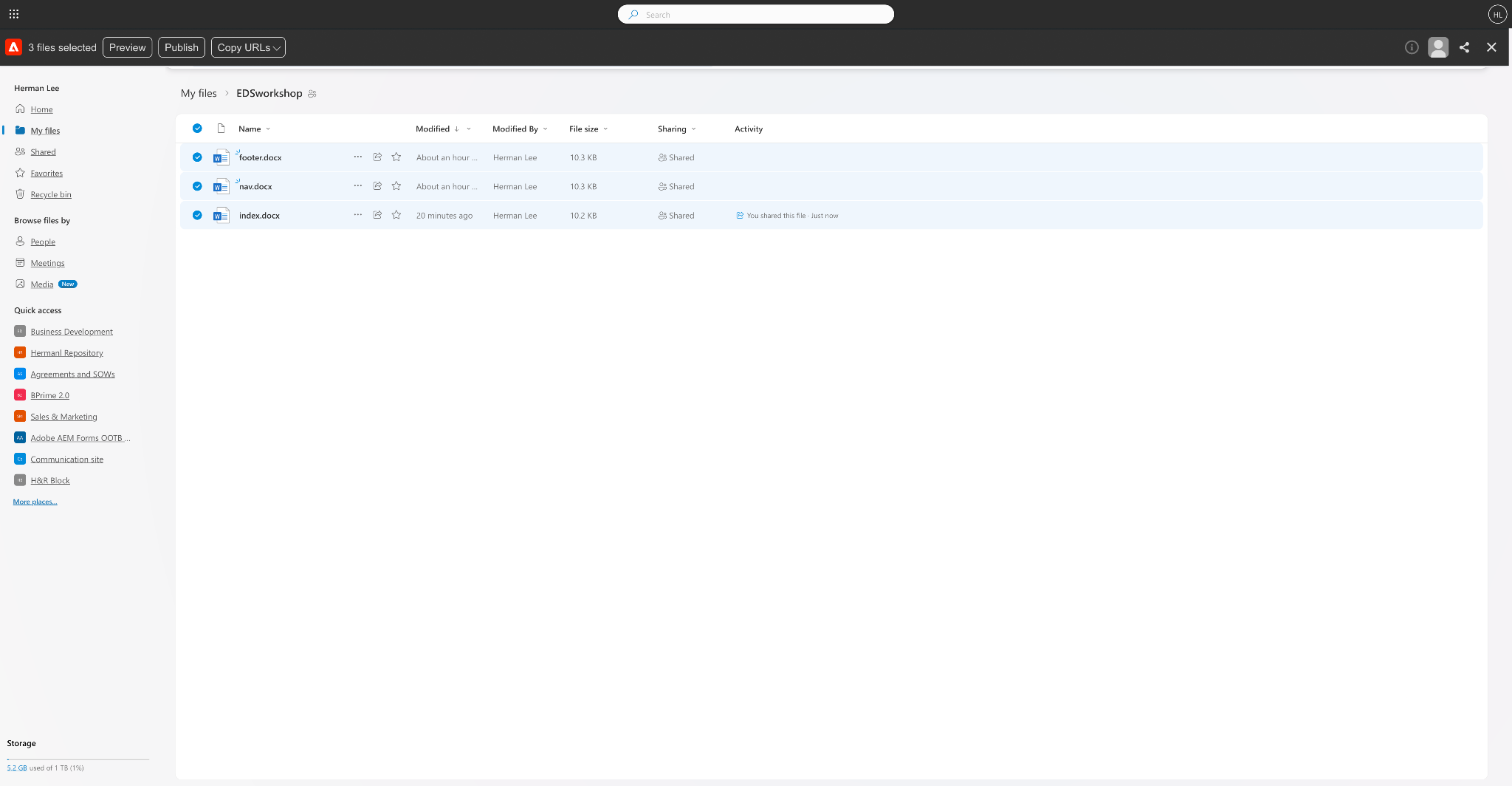
****

Open your preview site to confirm it worked. Your preview URL should be in the format [https://main–](about:blank)<repo name>--<Github username>.hlx.page. To confirm, you should see the text that you added to index.docx.

## Publish your content to the Production Site

Open the newly created folder in SharePoint with the 3 Word documents. Select them all.

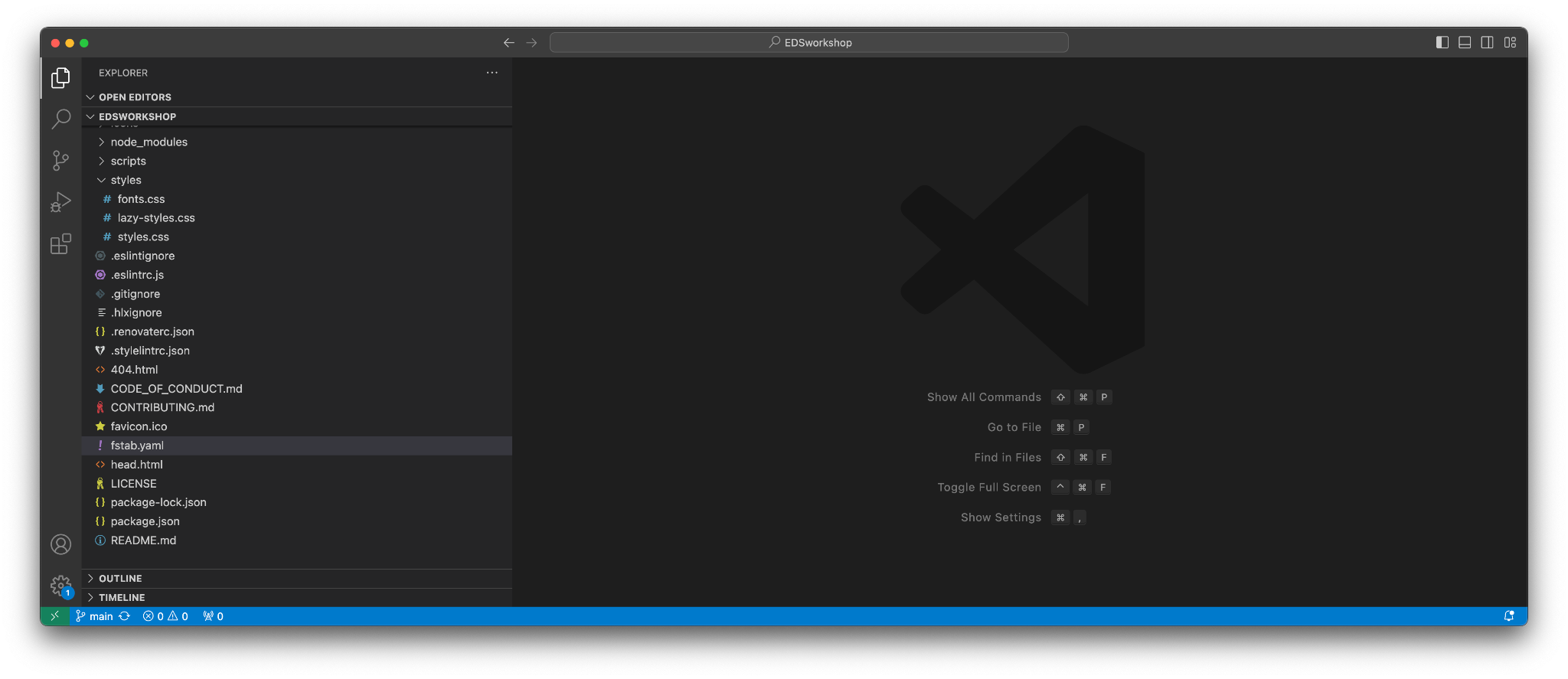
The AEM Sidekick should appear as a bar across the top of the page. Select the Publish button. Accept the prompts to publish the content to the Preview site.

****

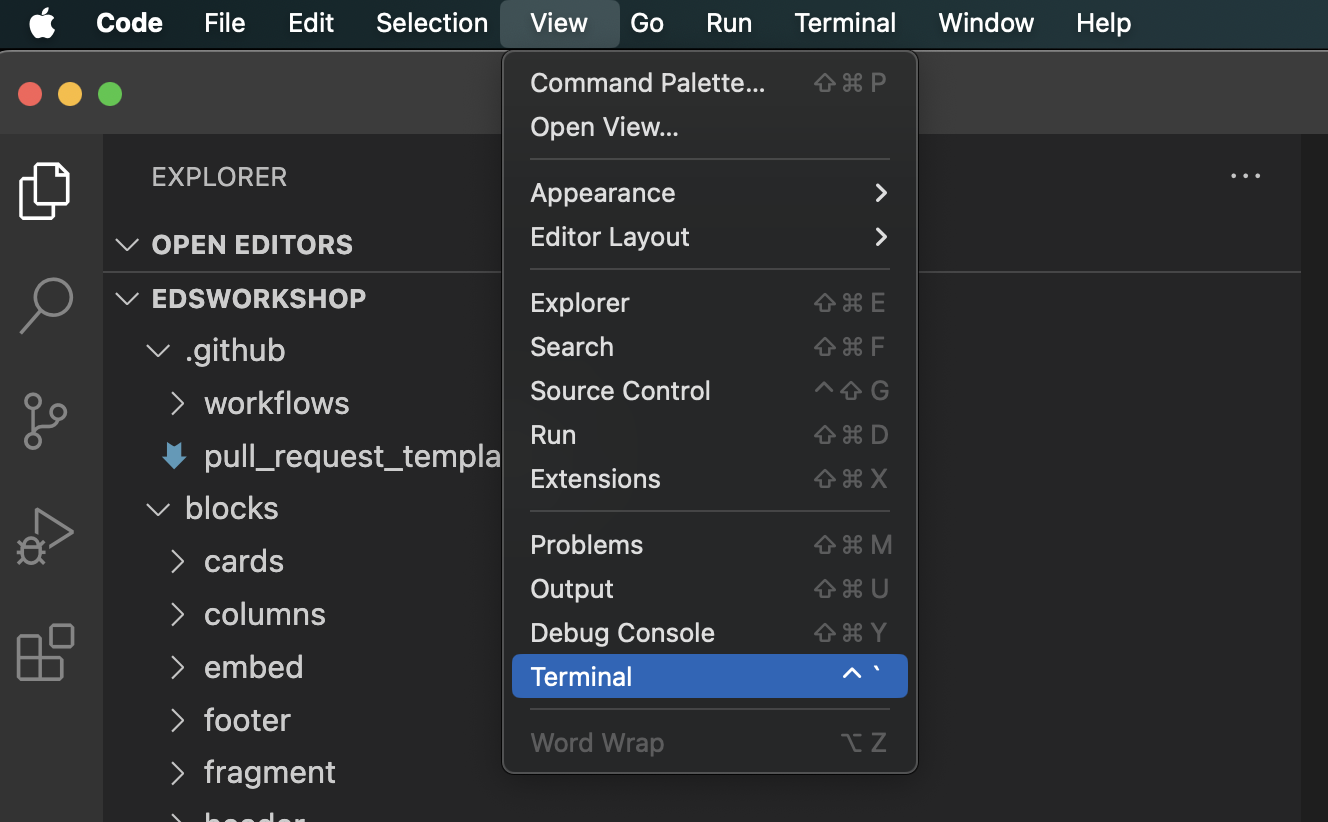
Open your preview site to confirm it worked. Your preview URL should be in the format [https://main–](about:blank)<repo name>--<Github username>.hlx.live. To confirm, you should see the text that you added to index.docx.

## Starting local version of the site

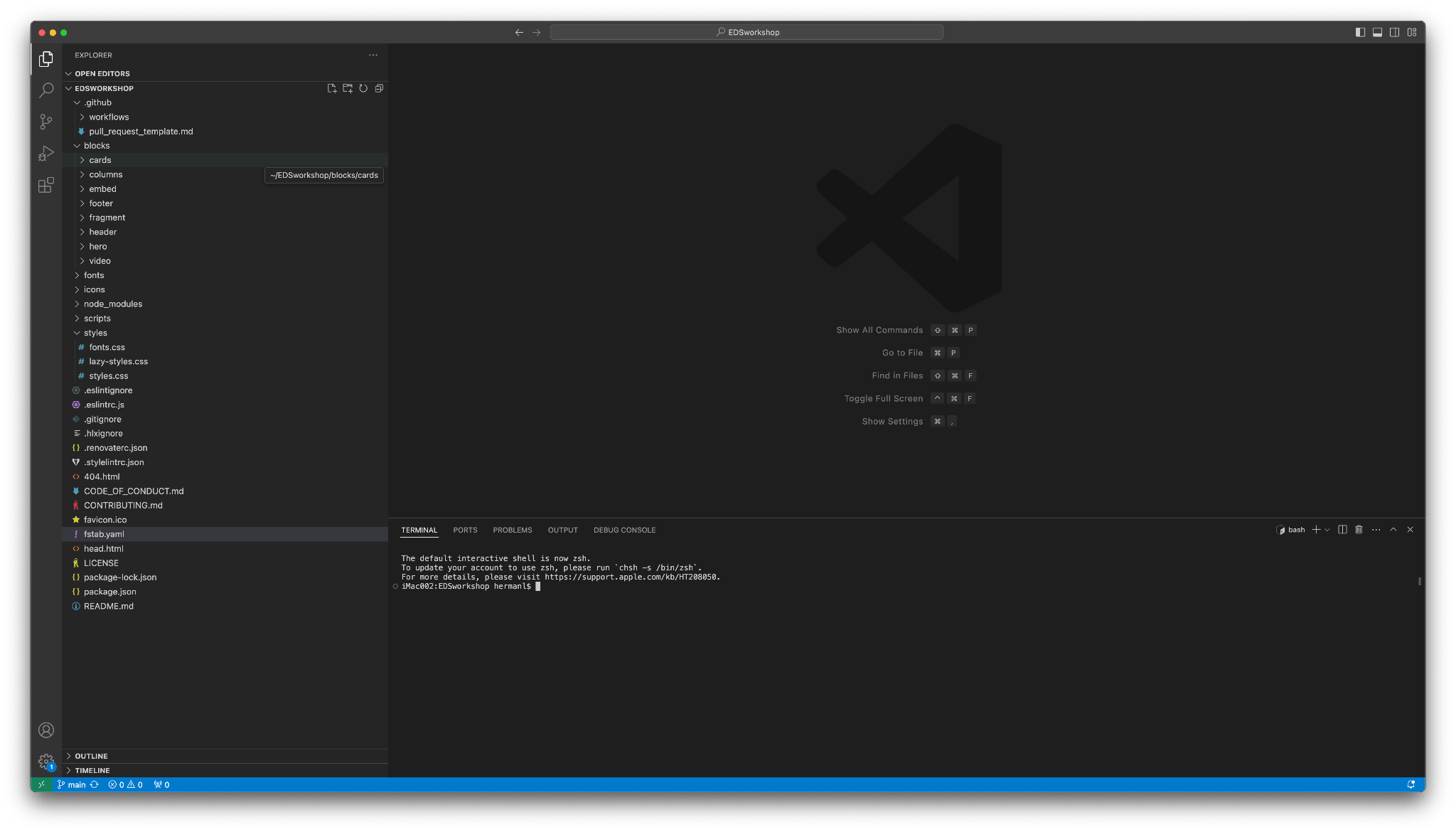
Open your codebase in Visual Studio Code. Confirm that it is the right codebase by looking at the name in the center of the menu bar.



Bring up the terminal by selecting View from the menu then Terminal.



The Terminal panel should appear at the bottom of the window.

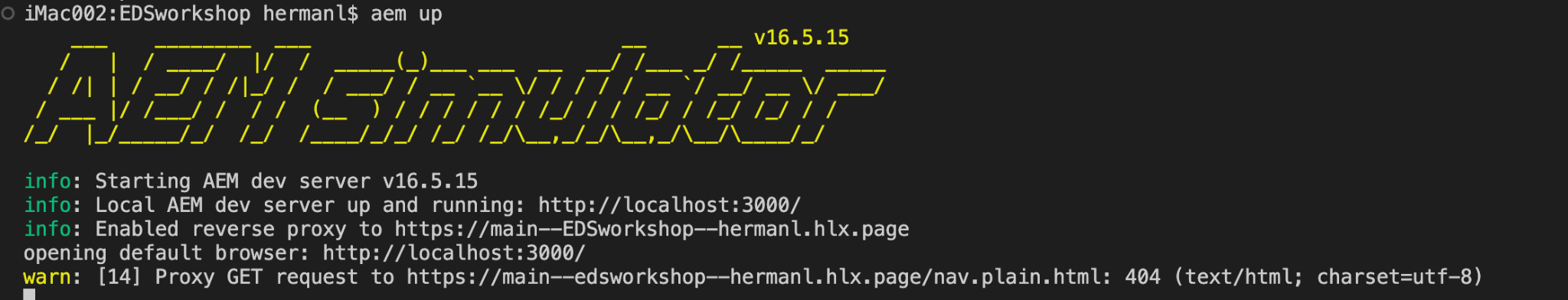


Type sudo npm install -g @adobe/aem-cli into the terminal and press the ENTER key on your keyboard. This will install the AEM Command Line Interface. (Skip if you have done this as part of the prerequisites.)

Type aem up into the terminal and press the ENTER key on your keyboard. This will turn on the tool.

**Note for Windows Users:** If you get a “...cannot be loaded because running scripts is disabled on this system” message when trying aem up, please try the solution mentioned in this [thread](https://stackoverflow.com/questions/56199111/visual-studio-code-cmd-error-cannot-be-loaded-because-running-scripts-is-disabl).

To confirm that it worked, you should see a similar message show up.



Your default browser should open localhost:3000. This is the local version of the site which displays content from your Sharepoint.

## This is the end of the setup guide.

Congratulations for making it to the end.

Remember that you will need to commit any changes to the code base to see changes. You will also need to preview any content changes to see updates to a page.

#### 

## Potential Issues You May Face

#### The Sidekick bar is not loading or the actions buttons are not available.

Open up index.docx and check if the bar or actions load. Report results in Slack.

#### aem up command results in command not found error

You may have issues with the Node Version Manager, wrong Node version, and permissions. See Slack for Kyle’s solution.

| **Metadata** | |
| --- | --- |
| Title | Local Development Setup |
| Description | This tutorial will get you up-and-running locally for document-based sites. |