



## CS 470 Module Three Assignment Guide

### Part 1 - Preparing the Angular Code

In Module Two, you forked and cloned the **learn-angular-from-scratch-step-by-step** code into a **lafs-web** directory on your computer. Now you need to do some work to make that codebase ready for use on Amazon Web Services.

1. Open a command shell, such as PowerShell.
2. Change the directory to **lafs-web**.
3. Build the Angular application for deployment.
  - a. `ng build --prod --base-href .`
  - b. If you receive the following error, then you need to install the SASS preprocessor. See steps C and D. Otherwise, go to step 4. For more information on this error see [Github Angular CLI: 7.0.3 Cannot Find Module 'node-sass'](#).

```
ERROR in ./src/styles.scss
Module build failed (from ./node_modules/mini-css-extract-plugin/dist/loader.js):
ModuleBuildError: Module build failed (from ./node_modules/sass-loader/lib/loader.js):
Error: Cannot find module 'node-sass'
```

- c. `npm install --save-dev node-sass`
  - d. `ng build --prod --base-href .`
4. You are now done preparing the code. What you did was tell Angular to use the current path (.) instead of the default root path (/) for routing of sub pages on your website. A good reference on the **base-href** and **deploy-url** build options for Angular can be found at [Angular 4: Use of base-href and deploy-url Build Options](#).

### Part 2 – Creating an S3 Bucket

In part 1, you prepared your code to be deployed to the cloud. Now you need to create the place to which it will be deployed. In this section, you will create an S3 bucket to be your deployment target.

1. Log into **AWS Academy**.
2. Access the **AWS Console** by clicking the “modules” button on the left, then “Learner Lab – Foundational Services”, then “Start Lab” to start the lab.
3. Once the Lab has started (you can tell from the Cloud Access message), and the “light” by the AWS at the top of the screen as turned green, click on the AWS link to start the management console.
4. Type “S3” in the **Find Services** search bar and select **S3**, or select the All Services link, then click **S3** under the **Storage** heading.
5. Click the orange **Create Bucket** button.
6. You will need to create a unique S3 Bucket name. See [Rules for Bucket Naming](#). Some key points:
  - a. Bucket names are unique by AWS Partition (World, China, and the Government). For our purposes, we operate in the "World" partition; your name must be globally unique.
  - b. Do not worry too much about the name. It has no real impact on your applications other than as a "locator" of your bucket. Many companies choose to generate names using a formula.



- c. If you cannot get your first choice, add some numbers or otherwise make the name slightly more unique.
7. Keep the region as **US East (N. Virginia) us-east-1**.
8. Keep the **Block All Public Access** check box checked.
9. Do not change anything in advanced settings.
10. Click **Create Bucket** on the bottom of the page.
11. The console should return you to the list of S3 buckets. You will see your new bucket listed, its region, that it is not public, and a timestamp showing when it was created.

Congratulations! You have now created your first serverless cloud “thing”!

### Part 3 – Deploying Your Website

In part 2, you created your S3 bucket to host your website. Now it is time to deploy your application.

1. Click on the link for the bucket name.
2. Click on the blue **Upload** button.
3. Drag and drop all the files from the **lafs-web/dist/learn-angular-from-scratch** directory. Make sure you select the contents of the directory and not the directory itself.
  - a. If you cannot or do not want to drag and drop your files, follow these steps to upload manually.
    - i. Select the **Create Folder** blue box.
    - ii. Type in “assets” (without quotes). Leave all other settings the same.
    - iii. Upload all the files in the **lafs-web/dist/learn-angular-from-scratch** directory to the S3 bucket root.
    - iv. Upload all the files in the **lafs-web/dist/learn-angular-from-scratch/assets** directory to the S3 bucket **assets** folder.
4. Click the blue **Upload** button in the lower-left corner. Leave all the settings as they are until the last page, then click the blue **Upload** button (this time it will be in the lower-right corner).

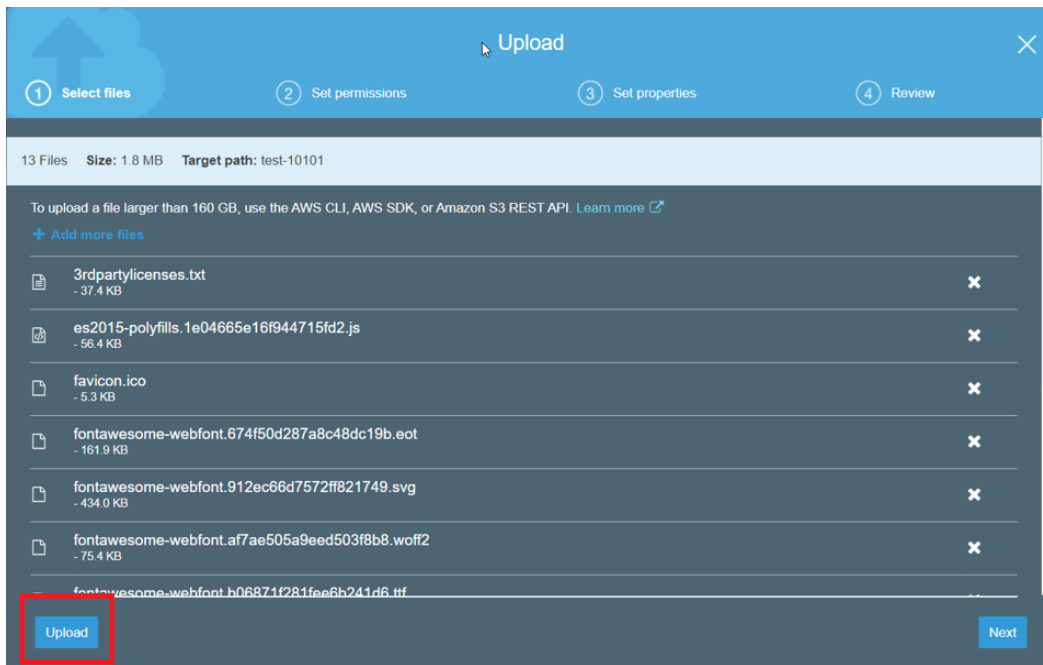


Figure 1: Upload Dialog Box

5. You should have 14 objects in your S3 folder.

cs470-sample-website

Overview	Properties	Permissions	Management	Access points
<input type="text" value="Type a prefix and press Enter to search. Press ESC to clear."/>				
<div> <input type="button" value="Upload"/> <input type="button" value="+ Create folder"/> <input type="button" value="Download"/> <input type="button" value="Actions"/> </div> <div style="text-align: right;">US East (N. Virginia) </div>				
Viewing 1 to 14				
<input type="checkbox"/> Name	Last modified	Size	Storage class	
<input type="checkbox"/> assets	--	--	--	
<input type="checkbox"/> 3rdpartylicenses.txt	May 3, 2020 4:15:19 PM GMT-0400	37.4 KB	Standard	
<input type="checkbox"/> es2015-polyfills.1e04665e16f944715fd2.js	May 3, 2020 4:15:19 PM GMT-0400	56.4 KB	Standard	
<input type="checkbox"/> favicon.ico	May 3, 2020 4:15:19 PM GMT-0400	5.3 KB	Standard	
<input type="checkbox"/> fontawesome-webfont.674f50d287a8c48dc19b.eot	May 3, 2020 4:15:19 PM GMT-0400	161.9 KB	Standard	
<input type="checkbox"/> fontawesome-webfont.912ec66d7572f821749.svg	May 3, 2020 4:15:19 PM GMT-0400	434.0 KB	Standard	
<input type="checkbox"/> fontawesome-webfont.af7ae505a9eed503f8b8.woff2	May 3, 2020 4:15:18 PM GMT-0400	75.4 KB	Standard	
<input type="checkbox"/> fontawesome-webfont.b06871f281fee6b241d6.ttf	May 3, 2020 4:15:18 PM GMT-0400	161.7 KB	Standard	
<input type="checkbox"/> fontawesome-webfont.fee6e712a8a08eef580.woff	May 3, 2020 4:15:18 PM GMT-0400	95.7 KB	Standard	
<input type="checkbox"/> index.html	May 3, 2020 4:15:18 PM GMT-0400	897.0 B	Standard	
<input type="checkbox"/> main.52621aeb4efe13ad729f.js	May 3, 2020 4:15:18 PM GMT-0400	614.2 KB	Standard	
<input type="checkbox"/> polyfills.8bbb231b43165d65d357.js	May 3, 2020 4:15:19 PM GMT-0400	41.0 KB	Standard	
<input type="checkbox"/> runtime.26209474bfa8dc87a77c.js	May 3, 2020 4:15:19 PM GMT-0400	1.4 KB	Standard	
<input type="checkbox"/> styles.0800ae2f50afe9efad98.css	May 3, 2020 4:15:19 PM GMT-0400	204.7 KB	Standard	
Viewing 1 to 14				

Figure 2: S3 Bucket After File Upload

That is it. You have deployed your application.

## Part 4 – Configuring Your S3 Bucket as a Website



In part 3, you deployed your application to your S3 bucket, but it is not a website yet. In this section, you will configure your S3 bucket to operate as a website.

1. Click on the orange Close button.
2. Select the **Properties** tab.
3. Select the “Edit” button on the **Static Website Hosting** card.
4. Select the “Enable” radio button on the “Static Website Hosting” section.
5. Select the radio button for **Host a Static Website**. The [Learn More link](#) will provide some details on what we are about to do.
6. You will also need to do the following:
  - a. Index document. This is your website’s home page. Angular created this file and it is in your folder. You need to type in “index.html” or the **Save** button will not be enabled.
  - b. Error document. You do not have a custom error page, but if you did, this would be where you would put in the name of it.
  - c. Redirection rules. Leave this blank.
7. Click **Save Changes**.
8. Copy and save the endpoint shown. It will look something like this: `http://{your bucket name}.s3-website-us-east-1.amazonaws.com`.

Congratulations! You have now configured a serverless web server. Go try it by putting in the endpoint you saved into a browser.

Note: If you forgot to save the endpoint, don’t worry – we all do that. Just click on the **Static Website Hosting** card and it will be right there again.

## 403 Forbidden

- Code: AccessDenied
- Message: Access Denied
- RequestId: CEF5E295593BBCF
- HostId: ReZ0/4XnIz+o4/uZVsb3XMtJGWolxcpFjMPndgWAAN1qEMT0EUbTYDH3yqWkKi6bez733xaDmaE=

*Figure 3: HTTP 403 Forbidden*

Oh No, a 403! Don't worry! You did not make a mistake; in fact, you did it right. Your website is secure by default, so no public access is allowed. We will make it public in the next set of steps.

### Part 5 – Configuring Security for Your S3 Bucket Website

In part 4, you configured your bucket to work as a website, but you are not able to access it from a browser. Not a very good website. We will now configure the security for your bucket so it is public and readable.

This is done in two separate steps: making your bucket public, and allowing public access to the read operations.

#### Making Your Bucket Public

1. Select the **Permissions** tab for your bucket.

2. This tab should have a **Block Public Access** section, with an Edit button in blue, and a box showing "Block All Public Access".
3. Select the **Edit** button in the upper-left of the Block public access box.
4. Uncheck the **Block All Public Access** check box.
5. Click **Save Changes**.
6. Type **confirm** in the dialog box and then click the orange **Confirm** button. This is confirming to AWS that you *really* want your S3 bucket to be public.
7. Test your end point again.

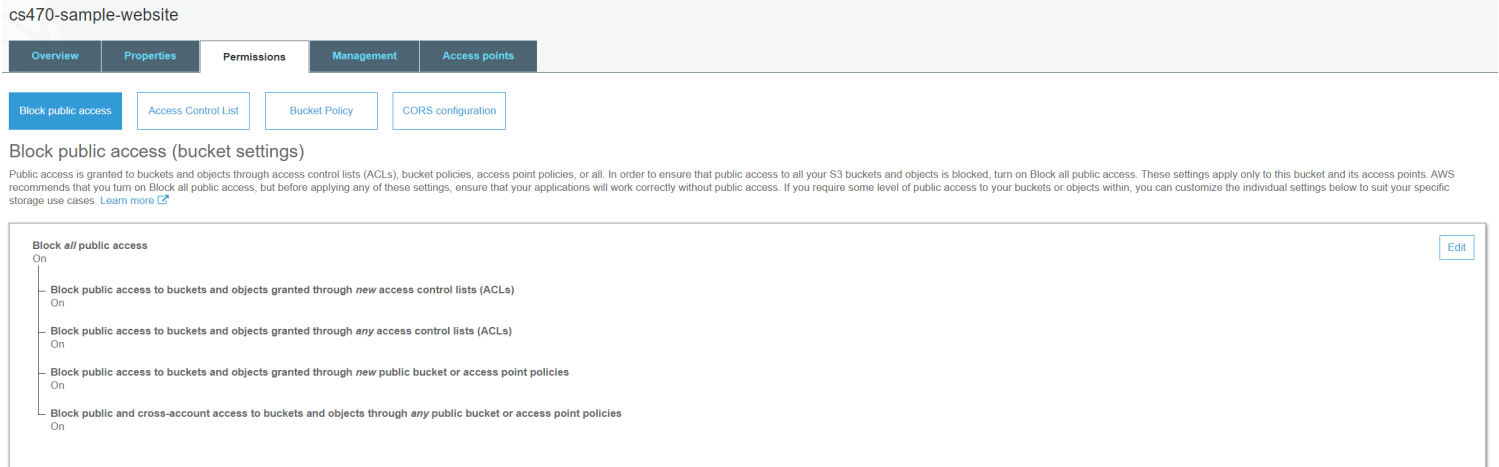


Figure 4: Block Public Access Settings

A 403! Again, don't worry! You still did not make a mistake; in fact, you are still doing it right. S3 has two layers of security: the bucket, and the operations on the bucket. You have just opened the security on the bucket, but you did not authorize any operations.

### Enabling the Read Access Operations

1. Create a bucket policy to enable read access.
2. Select the **Bucket Policy** button under **Permissions** for your bucket.
3. You will need to modify the following JSON segment and paste it into the box on the screen. You MUST replace **example.com** with your bucket name.

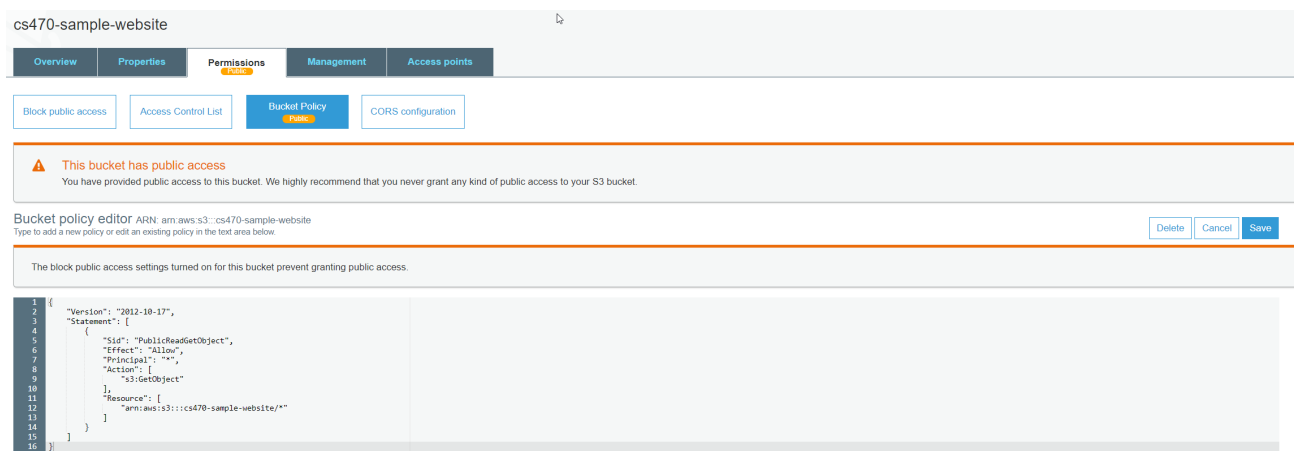
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ]
    }
  ]
}
```

```

    "Resource": [
      "arn:aws:s3:::example.com/*"
    ]
  }
}

```

4. When the JSON is pasted into the box, click **Save**.
5. If done correctly, the screen will change, and the **Permissions** tab and the **Bucket Policy** will both show the word "Public".
6. Now go try your website in the browser again.



cs470-sample-website

Overview Properties **Permissions** Management Access points

Block public access Access Control List **Bucket Policy** CORS configuration

**Warning** This bucket has public access  
You have provided public access to this bucket. We highly recommend that you never grant any kind of public access to your S3 bucket.

Bucket policy editor ARN: arn:aws:s3:::cs470-sample-website  
Type to add a new policy or edit an existing policy in the text area below. [Delete] [Cancel] [Save]

The block public access settings turned on for this bucket prevent granting public access.

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "PublicReadGetObject",
6       "Effect": "Allow",
7       "Principal": "*",
8       "Action": [
9         "s3:GetObject"
10      ],
11      "Resource": [
12        "arn:aws:s3:::cs470-sample-website/*"
13      ]
14    }
15  ]
16 }

```

Figure 5 - AWS Public access warnings

**Congratulations!** You did it: A totally serverless website migrated from desktop to containers and now to the cloud.