**Introduction**

You can create an S3 bucket using the AWS Management Console. As with many other AWS services, you can use the AWS API or CLI (command line interface) as well. This lab uses the AWS Management Console for all S3 related tasks. Once the bucket is created you will configure it for a static website.

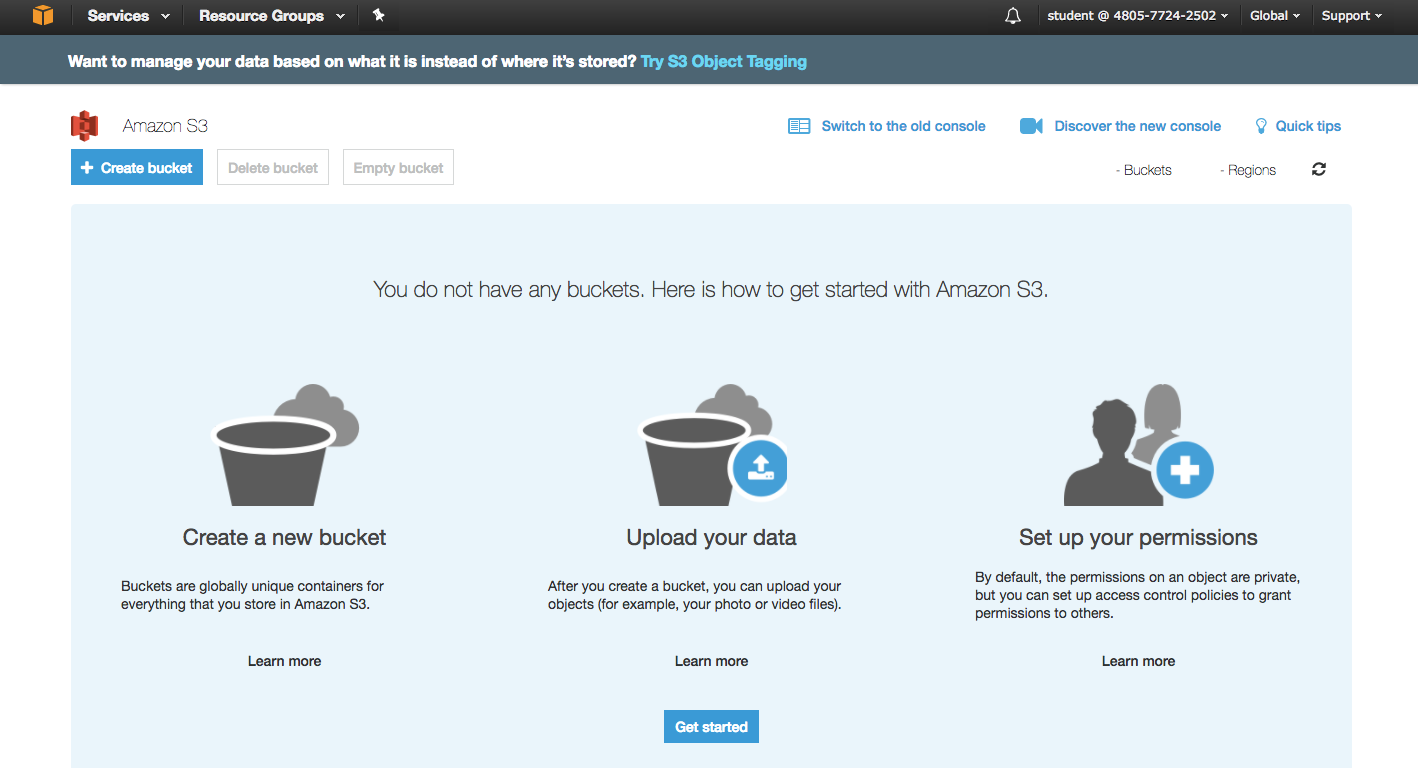
**Instructions**

1. Select the **S3** service from the AWS Management Console under the **Storage** section:

[](https://assets.cloudacademy.com/bakery/media/uploads/lab-step/blobid2-c82d92b8-a95d-4791-952b-48b8057ee047.png)

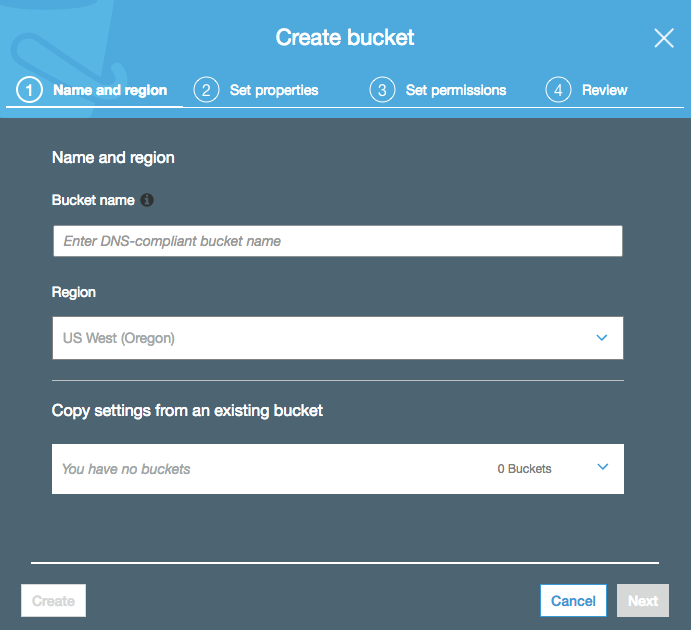
You will be placed in the S3 console.

2. From the S3 console, click the blue **+** **Create Bucket** button:



A four part **Create bucket** wizard starts. Screen 1 of 4 is for the **Name and region** information.

3. Enter a unique **Bucket name** on the **Name and region** screen of the wizard:



* **Bucket name**: Enter *calabs-bucket-<UniqueNumber>*(Append a unique number to the end of calabs-bucket- Examples: calabs-bucket-3, calabs-bucket-456)
* **Region**: US West (Oregon) (This should be set for you. If not, please select this region.)

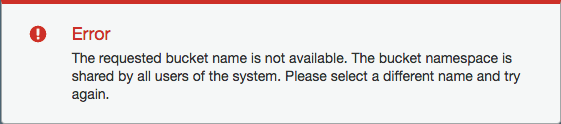
***Important!***Bucket names must be globally unique, regardless of the AWS region in which you create the bucket. Buckets must also be DNS-compliant.

The rules for DNS-compliant bucket names are:

* They must be at least 3 and no more than 63 characters long.
* They may contain lowercase letters, numbers, periods, and/or hyphens. Each label must start and end with a lowercase letter or a number.
* They *cannot* be formatted as an IP address (for example, 192.168.1.1).

The following examples are valid bucket names: calabs-bucket-1, cloudacademybucket , cloudacademy.bucket , calabs.1 or ca-labs-bucket.

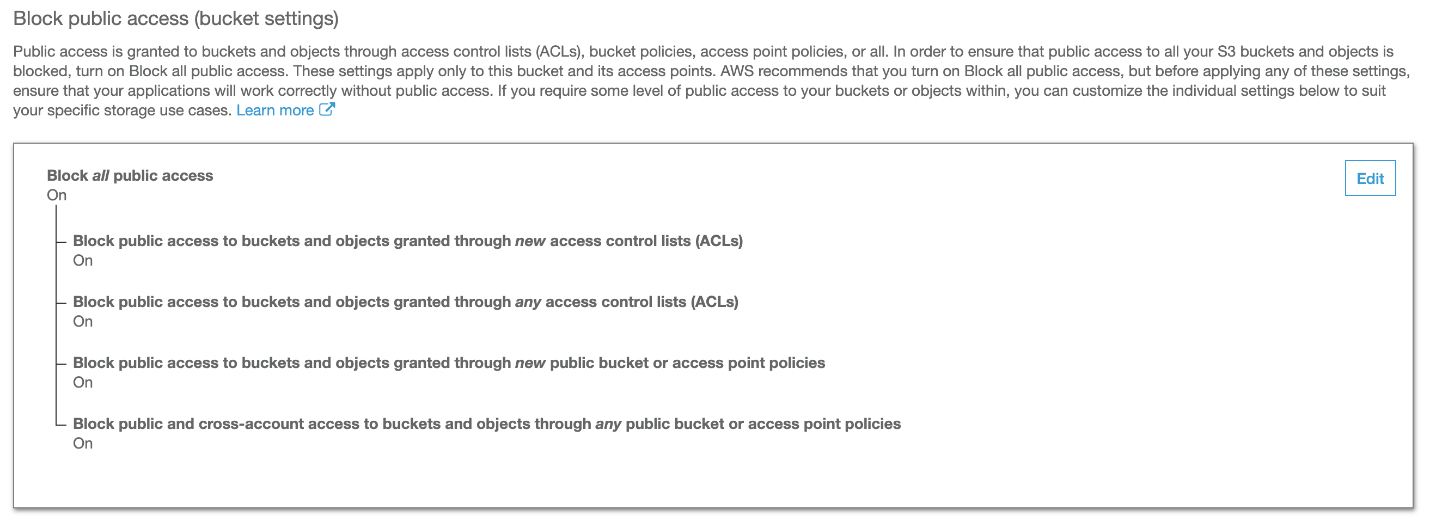
*Troubleshooting Tip*: If you receive an error because your bucket name is not unique, append a different unique number to the bucket name in order to guarantee its uniqueness:



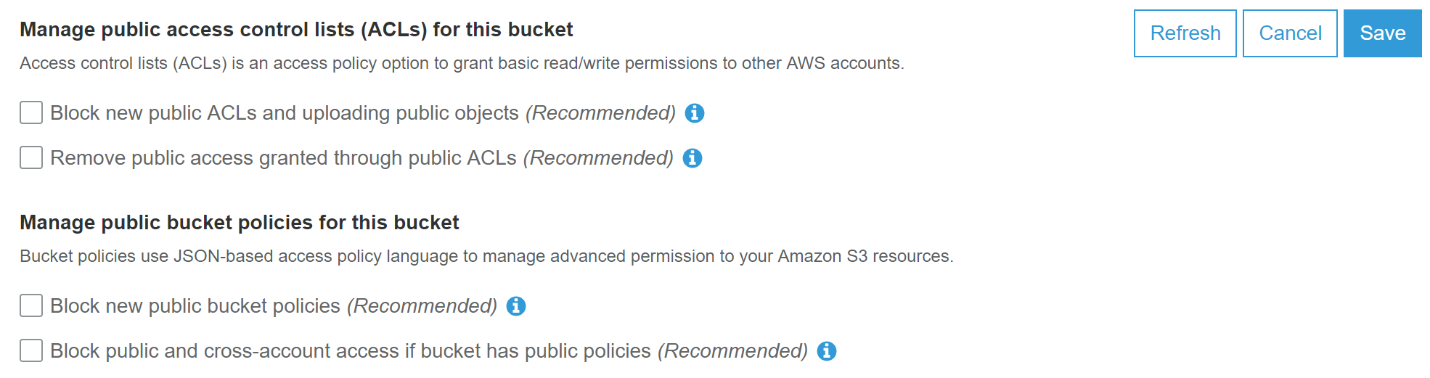
For example, change "calabs-bucket" to "calabs-bucket-1" (or a unique number/character string) and try again.

4. Click **Create** to create the new bucket, and then click on the name of the bucket you've just created.

5. Click the **Permissions** tab and click **Edit** for the **Public access settings**:

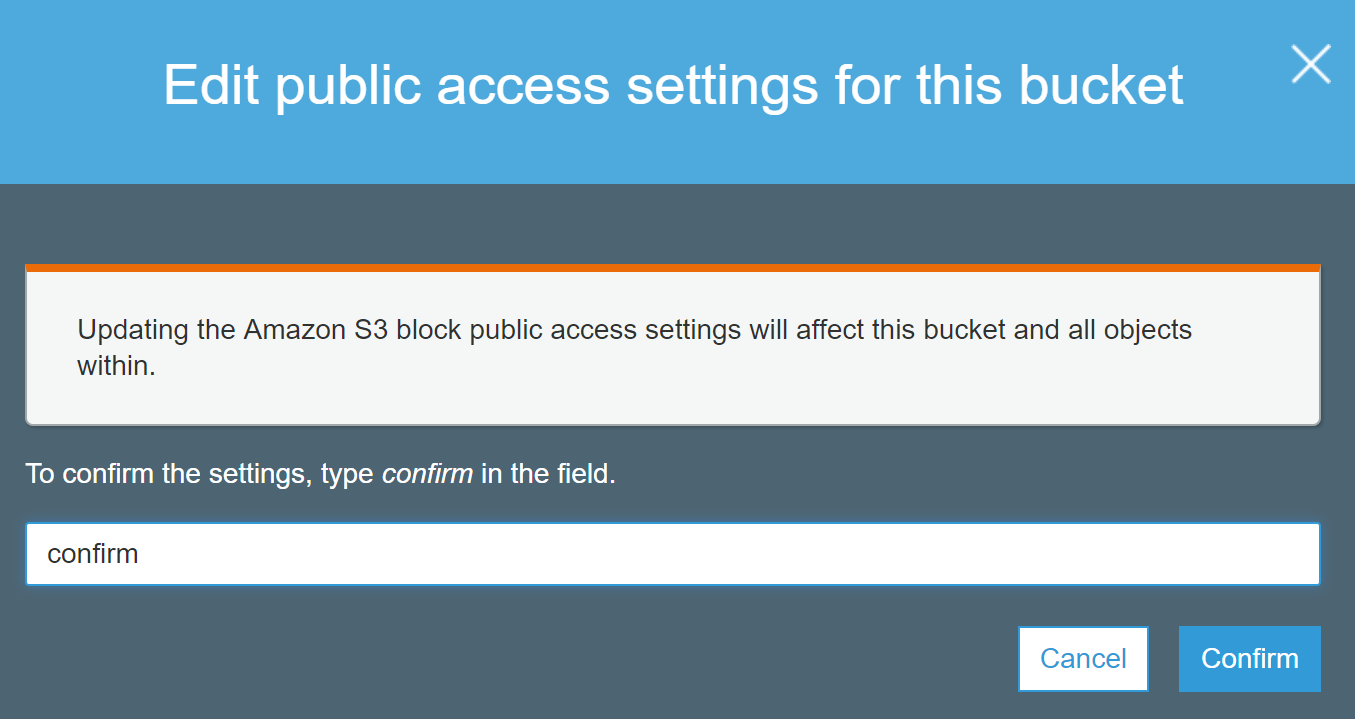


6. Uncheck all of the options to allow all kinds of public access and click **Save**:

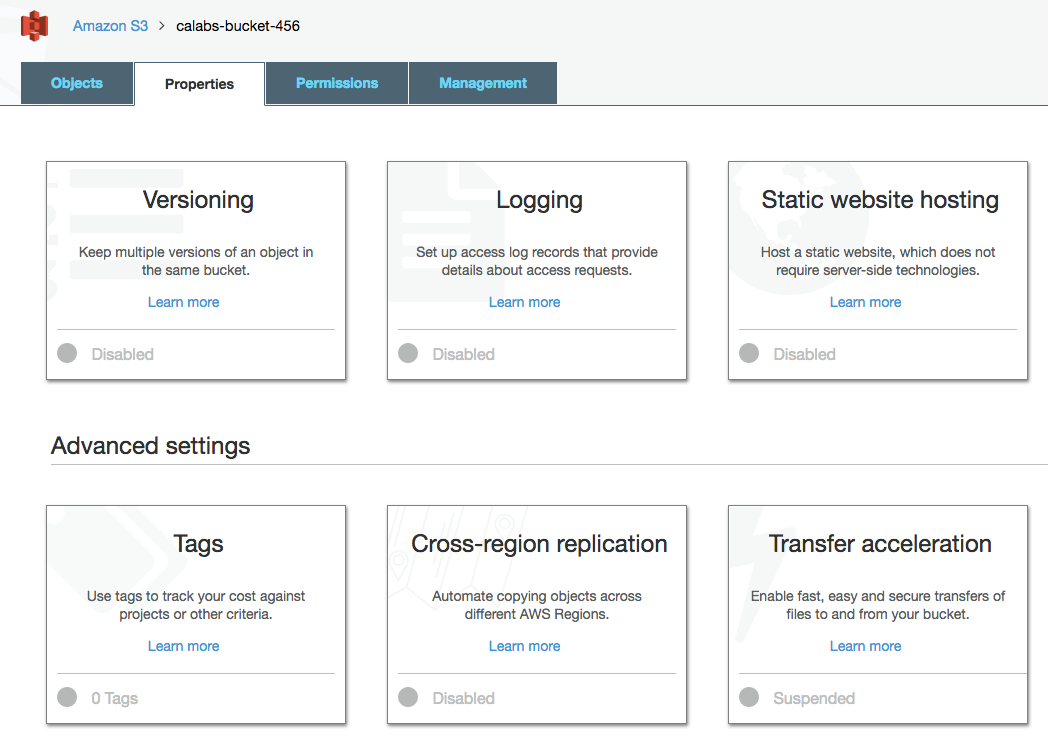


*Note*: You should carefully consider anytime you allow public access to S3 buckets. AWS has implemented these security features to help prevent data breaches. For this Lab, there is no sensitive data and you do want to allow public access.

7. Enter *confirm*in the confirmation modal to allow public access:



8. Go back to the S3 dashboard and then select your bucket. Click **Properties** to switch to the **Properties** tab for your new bucket:



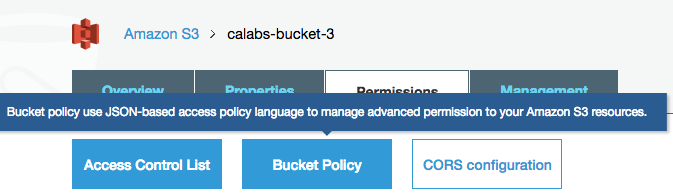
9. Click the **Static website hosting** tile. By default, static website hosting is disabled.

10. Click **Use this bucket to host a website**, then fill out:

* **Index document**: Enter *index.html*
* **Error document**: Enter *error/index.html*

Click **Save** when ready to proceed. Your S3 bucket is ready to host content. Rather than manually changing permissions on multiple objects in your bucket, next you will create a bucket policy. The policy will apply to all objects uploaded to your bucket.

11. Navigate to the **Permissions** tab of your bucket, then click **Bucket Policy:**

****

The policy is JSON based.

12. Copy and paste the following into your **Bucket policy editor**:

{  
   "Version": "2012-10-17",  
   "Statement": [  
      {  
         "Sid": "AddPerm",  
         "Effect": "Allow",  
         "Principal": "\*",  
         "Action": "s3:GetObject",  
         "Resource": "arn:aws:s3:::*YOUR\_BUCKET\_NAME*/\*"  
      }  
  ]  
}

Be sure to substitute your actual S3 bucket name for **YOUR\_BUCKET\_NAME** above. Click **Save** when ready to proceed. The JSON policy will allow public access to all objects beneath your S3 bucket. (Again, this is a very gratuitous policy. Your production policies may differ for security reasons.)

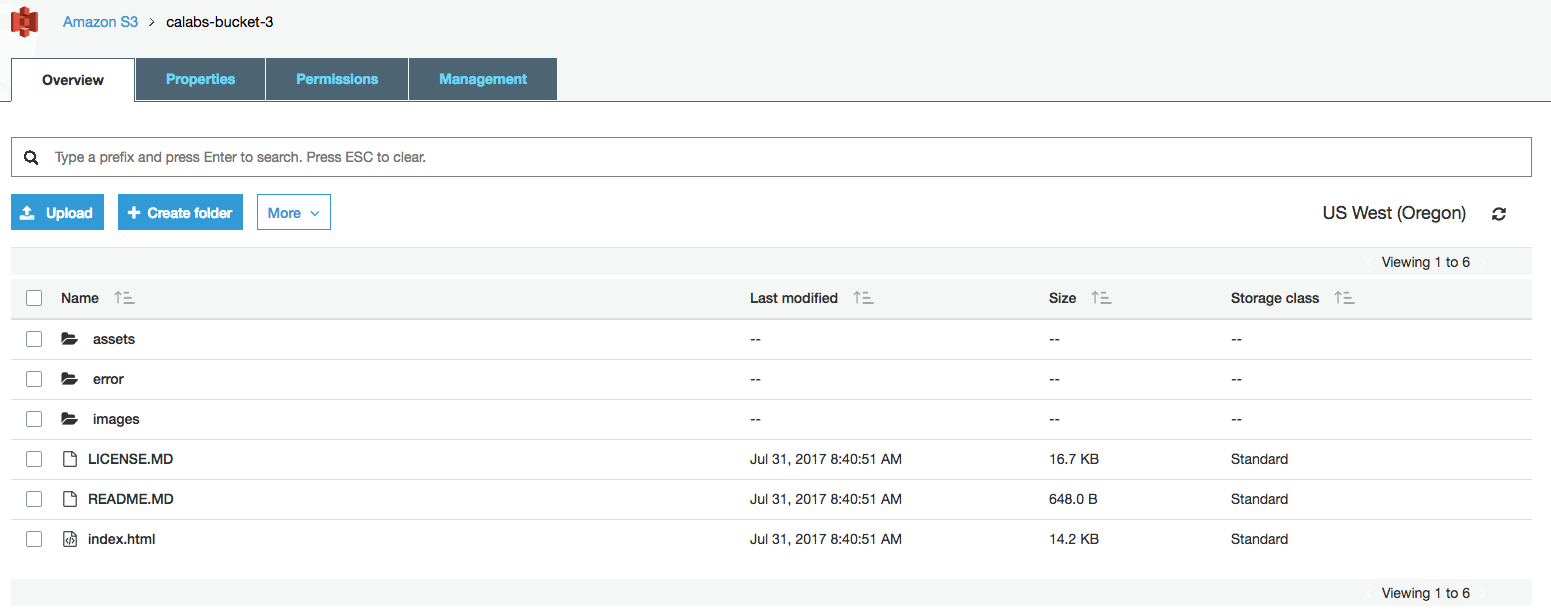
Next you will download a basic website from a public GitHub repository and load it into your S3 bucket.

13. Click [here](https://github.com/cloudacademy/static-website-example/archive/master.zip) to download a zip file of a basic website provided for you from the Cloud Academy public GitHub repository. Extract the contents of the zip archive to your local file storage.

14. From the top-level of your S3 bucket, click **Upload**. Navigate to where you placed the sample website content. Select or drag and drop all folders and files and click **Upload**.

The permissions on all files and folders uploaded to your S3 bucket will allow public read access.

15. Verify that you uploaded three top-level folders (assets, error, images) along with 3 top-level files as well (index.html, License, Readme) to your S3 bucket. It should look similar to the following:

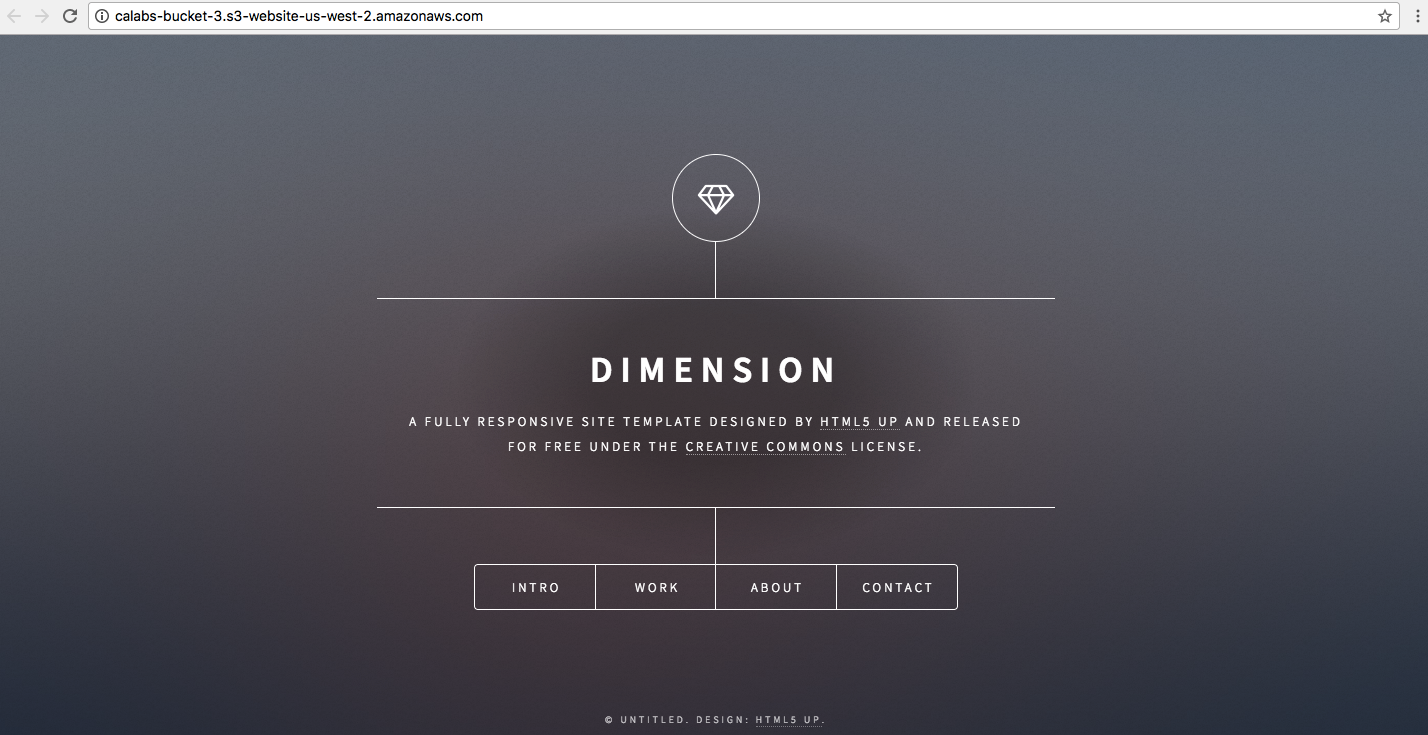


It's time to test everything out.

16. Open a new browser window (or tab) and navigate to the endpoint for your S3 static website.

*Reminder*: You can find the endpoint in the S3 console by navigating to your S3 bucket > **Properties** tab > **Static website hosting** tile (the endpoint is at the top).

The website should look similar to the following:



*Note*: The URL for your static S3 website endpoint will differ.

**Summary**

In this Lab Step you created an S3 bucket. You configured your S3 bucket to be a static website. You also established a bucket policy to apply to all objects uploaded to your bucket. Finally, you uploaded basic website content and confirmed that it was accessible on the web. At this point, you could simply maintain and use the basic website, or configure a content distribution network (CDN) such as Amazon CloudFront to allow global low-latency access to your website.