# **Exploring AWS S3 and CloudFront**

**Overview**

In this tutorial you will explore two Cloud Computing services with Amazon AWS: S3 and AWS CloudFront. These two services are often used together to create and deploy the static website through CDN.

Part 1 and Part 2 will show you how to deploy and publish a static website to the internet via AWS S3 and CloudFront.

You will need to use the Google Chrome web browser for this workshop.

## **Part 1. Creating an S3 Bucket to host a static web page**

Follow these steps to create an Amazon Simple Storage Service (Amazon S3) bucket to host a static website.

A static website is fixed and displays the same content for each user. In contrast, a dynamic website uses advanced programming to provide user interaction and display different content depending on the user's selections.

**Access the AWS Management Console**

1. Go to https://awsacademy.instructure.com/ and login.
2. Then go to "Modules" and then Learner Lab - Foundational Services.
3. To start the lab session, choose **Start Lab** in the upper-right corner of the page. o The lab session starts.

o A timer displays in the upper-right corner of the page and shows the time remaining in the session.

**Tip:** To refresh the session length at any time, choose **Start Lab** again before the timer reaches 0:00.

1. Before continuing, wait until the lab environment is ready. The environment is ready when the lab details appear on the right side of the page and the circle icon next to the **AWS** link in the upper-left corner turns green.
2. To connect to the AWS Management Console, choose the **AWS** link in the upper-left corner, above the terminal window.

A new browser tab opens and connects you to the AWS Management Console.

**Tip:** If a new browser tab does not open, a banner or icon is usually at the top of your browser with the message that your browser is preventing the site from opening pop-up windows. Choose the banner or icon, and then choose **Allow pop-ups**.

### **Task 1. Create an S3 bucket**

1. Choose the **Services** menu, locate the **Storage** services, and select **S3**.
2. Select **Create bucket** on the right side of the page.
3. For **Bucket name**, enter a unique Domain Name System (DNS)-compliant name for your new bucket that is specific to you, for example, "jeffreyting12345"

These are the DNS naming guidelines:

* The name must be unique across all existing bucket names in Amazon S3.
* The name must only contain lowercase characters.
* The name must start with a letter or number.
* The name must be between 3 and 63 characters long.
* After you create the bucket, you cannot change the name, so choose wisely.

1. For AWS **Region**, leave it as the selection (US East).
2. For Object Ownership, select ACLs enabled.
3. Uncheck(clear) the **Block *all* public access** box because you want to be able to test if the website is working.

A warning message similar to **Turning off block all public access might result in this bucket and the objects within becoming public** appear below the security setting you deselected.

1. Below the warning, check the box next to **I acknowledge that...**.
2. Scroll to the bottom of the page, and select **Create bucket**.

Your new bucket appears in the **Buckets** list.

### **Task 2. Add a bucket policy to make the content publicly available**

1. Click on the link for your bucket's name, and then select the **Permissions** tab.
2. In the **Bucket policy** section, choose **Edit**.
3. To grant public read access for your website, copy the following bucket policy, and paste it in the policy editor, replacing the default one that is already there:

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "PublicReadGetObject",

"Principal": "\*",

"Effect": "Allow",

"Action": [

"s3:GetObject"

],

"Resource": [

"arn:aws:s3:::jeffreyting12345/\*"

]

}

]

}

1. Obviously, replace "**jeffreyting12345"** with the name of your own bucket.
2. Select **Save changes**.

### **Task 3. Upload an HTML document**

In this task, you upload an HTML document to your new bucket.

1. Download the sample HTML web page from:

https://drive.google.com/file/d/1KvNqKfOJtso0lU5A6-ikxvkwNHU7eTQx/view?usp=drive\_lin k

and save it as : index.html

1. In the S3 buckets console, choose the **Objects** tab.
2. Upload the index.html file to your bucket.

* Choose **Upload**.
* Drag and drop the index.html file onto the upload page.
* As an alternative, choose **Add files**, navigate to the file, and choose **Open**.

1. Expand the **Permissions** section.
2. Under **Predefined ACLs**, select **Grant public-read access**.

A warning message similar to **Granting public-read access is not recommended** appears below the setting you selected.

1. Below the warning, check the box next to **I understand...**.
2. Expand the **Properties** section.

This section lists the storage classes that are available in Amazon S3..

Ensure that the **Standard** storage class is selected.

1. At the bottom of the page, choose **Upload**.
2. Choose **Close**.

The index.html file appears in the **Objects** list.

### **Task 4. Test your website**

1. Go back to your S3 Bucket console for your bucket.
2. Select the **Properties** tab, and scroll down to the **Static website hosting** section.
3. Choose **Edit**.
4. Select **Enable**.
5. In the **Index document** text box, enter index.html
6. Select **Save changes**.
7. Scroll down to the **Static website hosting** section again, and right click on the **Bucket website endpoint** URL and open it in a new tab:

The **Hello World** web page should display. You have successfully hosted a static website using an S3 bucket.

## **Part 2. Using CloudFront as a Content Delivery Network for a Website**

In this part, you will use Amazon CloudFront as a content delivery network (CDN) for a website that is stored in the Amazon Simple Storage Service (Amazon S3).

### **Task 1. Create a CloudFront distribution to serve your website**

1. Go to the main AWS Console
2. Choose the **Services** menu, locate the **Networking & Content Delivery** section, and choose **CloudFront**.
3. Choose **Create a CloudFront Distribution**.
4. Choose the text box next to **Origin Domain Name** and select the endpoint from your S3 bucket.
5. For **Viewer Protocol Policy**, ensure that **HTTP and HTTPS** is selected.
6. Scroll to the bottom of the page and select **Create Distribution**.
7. Go back to the CloudFront Distributions page.

A new CloudFront distribution displays in the distributions list. The **Status** will say *In Progress* until your website has been distributed. This may take a few minutes, sometimes longer. You will have to be patient.

Wait until the **Status** says *Deployed*, you can test your distribution.

1. Copy the **Domain Name** value for your distribution and save it to a text editor to use in a later step.
2. Create a new HTML file to test the distribution.

* Find and download an image from the internet.
* Navigate to your S3 bucket and upload the image file to it, making sure to grant public access as you did when uploading the HTML file earlier in this lab.
* Create a new text file using Notepad and copy the following text into it:

**<html>**

**<head>My CloudFront Test</head>**

**<body>**

**<p>My test content goes here.</p>**

**<p><img src="http://domain-name/object-name" alt="my test image">**

**</body>**

**</html>**

* Replace **domain-name** with the domain name that you copied earlier for your CloudFront distribution.
* Replace **object-name** with the file name of the picture file that you uploaded to your S3 bucket. The edited line of code should look similar to the following:

**<p><img src="http://d2g0a7p2vcceab.cloudfront.net/picture.jpg" alt="my test image"></p>**

* Obviously, replace my example URL with yours.
* Save the text file with an HTML extension.

1. Use a web browser to open the HTML file that you just created on CloudFront.

If the image that you uploaded shows, your CloudFront distribution was successful. If not, try again later or with a different image.

**Explore these two services further by creating and uploading other HTML pages and images and deploying them to the S3 and CloudFront services.**