In this lab, you use some of the Amazon Simple Storage Service (Amazon S3) features that you just learned about to create a static website.

Static websites can contain HTML pages, images, style sheets, and all files that are needed to render a website. Static websites do not use server-side scripting or a database. However, they may contain client-side scripts that run in a user's web browser.

You can host a static website on Amazon S3 by uploading the content and making it readable by users. No servers are needed, and you can use Amazon S3 to store and retrieve any amount of data at any time from anywhere on the web.

**Objectives**

After completing this lab, you will know how to do the following:

* Create a bucket in Amazon S3.
* Configure a bucket to host a static website.
* Upload content to a bucket.
* Turned on public access to bucket objects.
* Securely share a bucket object using a presigned URL.
* Secure a bucket using a bucket policy.
* Update the website.
* View object versions in the Amazon S3 console.

**Duration**

This lab requires approximately **30 minutes** to complete. You will have a total time of 180 minutes to complete this lab.

**Prerequisites**

This lab requires the following:

* A notebook computer running Microsoft Windows, macOS, or Linux (Ubuntu, SUSE, or Red Hat)
* An internet connection
* For Microsoft Windows users, administrator access to the computer
* An internet browser such as Chrome or Firefox

**Note**: This lab is incompatible with Internet Explorer 11. Use a different browser to launch this lab.

**AWS service restrictions**

In this lab environment, access to AWS services and service actions might be restricted to only the ones that you need to complete the lab instructions. You might encounter errors if you attempt to access other services or perform actions beyond the ones that this lab describes.

**Accessing the AWS Management Console**

1. At the top of these instructions, choose Start Lab to launch your lab.

A **Start Lab** panel opens, and it displays the lab status.

**Tip**: If you need more time to complete the lab, choose the **Start Lab** button again to restart the timer for the environment.

1. Wait until you see the message **Lab status: ready**, and then close the **Start Lab** panel by choosing the **X**.
2. At the top of these instructions, choose AWS

This opens the AWS Management Console in a new browser tab. The system automatically signs you in.

**Note:** If you find a dialog prompting you to switch to the new console home, choose **Switch to the new Console Home**.

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

1. Arrange the **AWS Management Console** tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time so that you can follow the lab steps.

**Task 1: Creating a bucket in Amazon S3**

In this task, you create an S3 bucket and configure it for static website hosting.

1. In the **AWS Management Console**, on the **Services** menu, choose **S3**.
2. Choose **Create bucket**

An S3 bucket name is globally unique, and all AWS accounts share the namespace. After you create a bucket, no other AWS accounts in any AWS Regions can use the name of that bucket unless you delete the bucket.

For this lab, you use a bucket name that includes a random number, such as **website-123**.

1. For **Bucket name**, enter website-<123> and replace *<123>* with a random number.

Public access to buckets is blocked by default. Because the files in your static website will need to be accessible through the internet, you must permit public access.

1. For **Object Ownership**, choose **ACLs enabled**.
2. Choose **Bucket owner preferred**.
3. For **Block Public Access settings for this bucket**, clear the check box for **Block *all* public access**, and then select the box that states **I acknowledge that the current settings might result in this bucket and the objects within becoming public.**
4. For **Bucket Versioning**, choose **Enable**.

**Note:** Once you turn on (enable) bucket versioning, you can't turn it off.

1. For **Tags**, choose **Add tag**, and enter the following:
   * **Key:** Department
   * **Value:** Marketing

You can use tags to add additional information to a bucket, such as a project code, cost center, or owner.

1. Choose **Create bucket**
2. In the **Buckets** section, choose the name of your new bucket.
3. Choose the **Properties** tab.

**Task 2: Configuring a static website on Amazon S3**

You will now configure the bucket for static website hosting.

1. Scroll to the **Static website hosting** panel.
2. Choose **Edit**
3. Configure the following settings:
   * **Static web hosting:** Choose **Enable**.
   * **Hosting type:** Choose **Host a static website**.
   * **Index document:** Enter index.html
   * **Error document:** Enter error.html

**Note**: You must enter index.html and error.html even though they are already displayed.

1. Choose **Save changes**
2. In the **Static website hosting** panel under **Bucket website endpoint**, choose the link.

You receive a *403 Forbidden* message because you have not yet configured the bucket permissions. Keep this tab open in your web browser so that you can return to it later.

You have configured your bucket to host a static website.

**Task 3: Uploading content to your bucket**

In this task, you upload the static files to your bucket.

1. Choose (right-click) each of the following links, and download the files to your computer:

Ensure that each file keeps the same file name, including the extension.

* [index.html](https://labs.vocareum.com/web/2259554/1007918.0/ASNLIB/public/docs/lang/assets/index.html)
* [script.js](https://labs.vocareum.com/web/2259554/1007918.0/ASNLIB/public/docs/lang/assets/script.js)
* [style.css](https://labs.vocareum.com/web/2259554/1007918.0/ASNLIB/public/docs/lang/assets/style.css)

1. Return to the Amazon S3 console, and choose the **Objects** tab.
2. Choose **Upload**
3. Choose **Add files**
4. Choose the three files that you downloaded.
5. Choose **Upload**

Your files are uploaded to the bucket.

1. Choose **Close**

**Task 4: Turning on public access to the objects**

Objects that are stored in Amazon S3 are private by default. This setting helps keep your organization's data secure.

In this task, you make the uploaded objects publicly accessible so users can view your website.

First, confirm that the objects are currently private.

1. Return to the browser tab that showed the *403 Forbidden* message.
2. Refresh the webpage.

If you accidentally closed this tab, go to the **Properties** tab, and in the **Static website hosting** panel, choose the **Bucket website endpoint** link again.

You should still see a *403 Forbidden* message. This response is expected! This message indicates that your static website is being hosted by Amazon S3 but that the content is private.

You can make Amazon S3 objects public through two different ways:

* To make either a whole bucket public or a specific directory in a bucket public, use a bucket policy.
* To make individual objects in a bucket public, use an access control list (ACL).

It is normally safer to make individual objects public because doing so avoids accidentally making other objects public. However, if you know that the entire bucket contains no sensitive information, you can use a bucket policy.

You now configure the individual objects to be publicly accessible.

1. Keep the website tab open, and return to the web browser tab with the Amazon S3 console.
2. Choose all three objects.
3. In the **Actions** menu, choose **Make public using ACL**.

A list of the three objects is displayed.

1. Choose **Make public**

Your static website is now publicly accessible.

1. Choose **Close**
2. Return to the web browser tab that has the *403 Forbidden* message.
3. Refresh the webpage.

You should now see the static website that is being hosted by Amazon S3.

Now you know how to share objects with everyone by making them public. However, there may be times when you need to share an individual object for a limited amount of time. In the next task, you learn how to temporarily share an object.

**Task 5: Securely sharing an object using a presigned URL**

When you need to temporarily and securely share an object with a person or group of people, you can create a presigned URL. When you create the URL, you must configure how long the URL will be valid. Then, you can share this URL with the users who should have access to the object.

As long as the presigned URL is valid, anyone who has it can get to the object. Avoid keeping the URL active longer than necessary, and only share the URL with people you trust.

1. Choose (right-click) the following link, and download the file to your computer: Ensure that the file keeps the same file name, including the extension.

* [new-report.png](https://labs.vocareum.com/web/2259554/1007918.0/ASNLIB/public/docs/lang/assets/new-report.png)

1. Return to the Amazon S3 console, and choose the **Objects** tab.
2. Choose **Upload**
3. Choose **Add files**
4. Choose the file that you downloaded.
5. Choose **Upload**

You have uploaded your file to the bucket.

1. Choose **Close**

Like when you first uploaded the website files, the **new-report.png** file is private by default. This time, instead of making the object public, you create a presigned URL to access the file.

1. In the **Objects** tab, choose **new-report.png**.
2. From the **Actions** menu, select **Share with a presigned URL**
3. In the pop-up window, configure the **Time interval until the presigned URL expires**:
   * Choose **Minutes**
   * For **Number of minutes**, enter 2
4. Choose **Create presigned URL**
5. From the banner at the top of the page, choose **Copy presigned URL**.
6. Open a new browser tab, and paste the URL you copied into the address bar.

A report is displayed in the web browser.

If you wait 5 minutes and use the link again, you will find that the URL has expired and no longer works.

**Task 6: Using a bucket policy to secure your bucket**

You want to protect your website files and make sure that no one can delete them. To do this, you apply a bucket policy that denies delete privileges on your website files.

1. Return to the Amazon S3 console, and choose the **Permissions** tab.
2. Under **Bucket policy**, choose **Edit**
3. Copy the following policy text. In the **Policy** text editor, replace the existing policy text with this text:



{

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

"Id": "MyBucketPolicy",

"Statement": [

{

"Sid": "BucketPutDelete",

"Effect": "Deny",

"Principal": "\*",

"Action": "s3:DeleteObject",

"Resource": [

"arn:aws:s3:::<bucket-name>/index.html",

"arn:aws:s3:::<bucket-name>/script.js",

"arn:aws:s3:::<bucket-name>/style.css"

      ]

}

]

}

This policy prevents everyone from deleting the three files that make your website work.

1. Next, you update the text in the policy editor. In the following lines of code in the policy editor, replace the placeholders with the name of your bucket.



"arn:aws:s3:::<bucket-name>/index.html",

"arn:aws:s3:::<bucket-name>/script.js",

"arn:aws:s3:::<bucket-name>/style.css"

Your updated code should look similar to the following:



"arn:aws:s3:::website-1234/index.html",

"arn:aws:s3:::website-1234/script.js",

"arn:aws:s3:::website-1234/style.css"

**Note:** Your bucket name will be different. Be sure to use the name of the bucket that you created.

1. Choose **Save changes**
2. Return to the the **Object tab**
3. Select **index.html**.
4. Choose **Delete**.
5. In the **Delete objects** panel, enter delete to confirm that you want to remove this file.
6. Choose **Delete objects**
7. Notice that the **index.html** file is listed in the **Failed to delete** pane.

This confirms that your policy is working and preventing the website's files from being deleted.

1. Choose **Close** to return to the **Objects** tab.

**Task 7: Updating the website**

Although you have configured a policy to prevent deletion of website files, you can still update the website by editing the HTML file and uploading it to the S3 bucket again.

Amazon S3 is an object storage service, so you must upload the whole file. This action replaces the existing object in your bucket. You cannot edit the contents of an object; instead, you must replace the whole object.

1. On your computer, load the **index.html** file into a text editor (for example, Notepad or TextEdit).
2. Find the text **Served from Amazon S3**, and replace it with Created by <YOUR-NAME>and substitute your name for (for example, **Created by Jane**).
3. Save the file.
4. Return to the Amazon S3 console, and upload the **index.html** file that you just edited.
5. Choose **index.html**, and in the **Actions** menu, choose the **Make public using ACL** option again.
6. Choose **Make public**.
7. Return to the web browser tab with the static website, and refresh the page.

Your name should now be on the page.

Your static website is now accessible on the internet. Because it is hosted on Amazon S3, the website has high availability and can serve high volumes of traffic without using any servers.

**Task 8: Exploring file versions**

Bucket versioning is turned off by default. When versioning is turned off, changes to objects can't be undone. For example, if you upload a new version of a file, the old file is replaced with the new one. The original file is lost. If you delete a file, it is permanently deleted, and you can't get it back.

However, when versioning is turned on, changed and deleted versions of files are saved. Previous versions of objects are not presented by default, but you can access them using the console or programmatically. Because you are keeping earlier versions of objects, you can recover them if you need to.

It is important to remember that once you turn on version, you cannot turn it off. However, you can suspend versioning. For more information on bucket versioning, see the [Amazon Simple Storage Service Users Guide](https://docs.aws.amazon.com/AmazonS3/latest/userguide/Versioning.html).

Recall that when you created your bucket, you turned on versioning. In this task, you view the object versions available in your bucket.

1. Return to the Amazon S3 console, and choose the **Objects** tab.
2. Choose **Show versions** to turn on bucket versioning.
3. Review the list of objects in the bucket.

* + Notice that each file has a **Version ID**. These IDs are automatically generated by Amazon S3 when versioning is turned on.
  + You should also find two versions of the **index.html** file because you uploaded a new version of the file. The current version is the file that you uploaded when you updated your website.

**Summary**

In this lab, you created a personalized, publicly accessible static website. You learned how to use a presigned URL to temporarily share objects in your bucket. You also protected your work with a bucket policy that prevents file deletion and turned on bucket versioning in case you need to recover previous versions of files. Excellent work!

**Lab complete**

Congratulations! You have completed the lab.

1. Choose End Lab at the top of this page, and then select **Yes** to confirm that you want to end the lab.

A panel indicates that *DELETE has been initiated... You may close this message box now.*

1. Select the **X** in the top right corner to close the panel.

*©2022 Amazon Web Services, Inc. and its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited.*