Simulation: Getting Started with Amazon S3

Simulation overview and objectives

In this simulation, 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 might 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 anytime from anywhere on the web.

Objectives

After completing this simulation, 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.
- Turn on public access to bucket objects.
- Securely share a bucket object by using a presigned URL.
- Secure a bucket by using a bucket policy.
- Update the website.
- View object versions in the Amazon S3 console.

Duration

This simulation requires approximately **30 minutes** to complete. You can take as long as you need.

Prerequisites

Before you begin this simulation, you should complete the *Getting Started with Storage* course content.

Amazon Web Services (AWS) service restrictions

In this simulation environment, you will be guided on which actions to perform. Because this environment is not live, you can only perform the actions that you are instructed to perform. If you choose any other actions, an error message will prompt you to the right actions. It is highly recommended that you review *How to use this simulation* in the introduction of the simulation.

Task 1: Creating a bucket in Amazon S3

In this task, you create an S3 bucket that you will use for static website hosting.

1. In the **AWS Management Console**, choose the search bar and enter *S3*.

   Note: To record your entry, press **Enter** on your keyboard or choose any place outside the entry field.

2. Then choose **S3** from the search results.
3. 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.

4. For **Bucket name**, enter `sim-website`.

   Note: To record your entry, press **Enter** on your keyboard or choose any place outside the entry field.

5. Choose the scroll bar to scroll down to **Object Ownership**.
6. For **Object Ownership**, choose **ACLs enabled**. Keep the default **Bucket owner preferred** selected.
7. Choose the scroll bar to scroll down to **Block Public Access settings for this bucket**.

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

8. For **Block Public Access settings for this bucket**, clear the checkbox for **Block *all* public access**. Then, select the box that states **I acknowledge that the current settings might result in this bucket and the objects within becoming public.**

9. Choose the scroll bar to scroll down to **Bucket Versioning**.
10. For **Bucket Versioning**, choose **Enable**.

    Note: As soon as you turn on (enable) bucket versioning, you can't turn it off.

11. For **Tags**, choose **Add tag**, and enter the following:

    - **Key:** `Department`
    - **Value:** `Marketing`

    Note: To record your entry, press **Enter** on your keyboard or choose any place outside the entry field.

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

12. Choose the scroll bar to scroll down.
13. Choose **Create bucket**.

    Your bucket appears in the list of buckets for your AWS account.

Task 2: Configuring a static website on Amazon S3

You will now configure the bucket for static website hosting.

17. In the list of your buckets, choose the name of the bucket that you just created, **sim-website**.
18. Choose the **Properties** tab.
19. Choose the scroll bar to scroll to the **Static website hosting** panel.
20. Choose **Edit** to the **Static website hosting** panel.
21. Choose **Enable**.
22. For **Hosting type**, keep the default setting **Host a static website**.
23. Configure the following settings:

- **Index document:** Enter `index.html`
- **Error document:** Enter `error.html`

Note: To record your entry, press **Enter** on your keyboard or choose any place outside the entry field.

Note: You must enter `index.html` and `error.html` even though they are already displayed. The display that you see is only an example of what the field is expecting.

24. Choose the scroll bar to scroll down.
25. Choose **Save changes**.
26. Choose the scroll bar to scroll to the **Static website hosting** panel.
27. In the **Static website hosting** panel under **Bucket website endpoint**, choose the link.

A new tab opens where you receive a *403 Forbidden* message because you have not yet configured the bucket permissions. You can return to it later.

28. Choose the **AWS Management Console** tab on your browser.

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.

22. Choose the scroll bar to scroll to the top of the page, and choose the **Objects** tab.
23. Choose **Upload**.
24. Choose **Add files**.
25. Choose the **Website files** folder, and choose **Open** to open the folder.
26. Use your mouse to choose each of the following files: **index.html**, **script.js**, and **style.css** (order does not matter). Then choose **Open**.
27. Choose the scroll bar to scroll down.
28. Choose **Upload**.

Your files are uploaded to the bucket.

27. Choose **Close**.

You have successfully uploaded your website file to your sim-website bucket.

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.

28. Return to the browser tab that showed the *403 Forbidden* message.
29. Choose the **Refresh** button for the webpage.

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 in 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.

30. Keep the website tab open, and return to the web browser tab with the **Amazon S3 console**.
31. Choose the **Name** checkbox to select all three objects.
32. In the **Actions** menu, choose **Make public using ACL**.

A list of the three objects is displayed.

33. Choose **Make public**.

Your static website is now publicly accessible.

34. Choose **Close**.
35. Return to the web browser tab that has the *403 Forbidden* message.
36. Refresh the webpage.

You should now see the static website that Amazon S3 is hosting.

37. On your browser, choose the **x** on the **My Static Website** tab.

Now you know how to share objects with everyone by making them public. However, at times, you might 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 by using a presigned URL

When you must 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.

If 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.

38. Choose **Upload**.
39. Choose **Add files**.
40. Choose the file **new-report** file and choose **Open**.
41. Choose the scroll bar to scroll down.
42. Choose **Upload**.

    You have uploaded your file to the bucket.

43. 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.

44. In the **Objects** tab, choose **new-report.png**.
45. From the **Actions** menu, select **Share with a presigned URL**.
46. In the pop-up window, keep the default **Minutes** selected for the **Time interval until the presigned URL expires**.
47. For **Number of minutes**, enter `2`.

    Note: To record your entry, press **Enter** on your keyboard or choose any place outside the entry field.

48. Choose **Create presigned URL**.
49. From the banner at the top of the page, choose **Copy presigned URL**.
50. Open a new browser tab.
51. Paste the URL that you copied into the address bar. Use these specific steps to paste and launch the URL:
    - Choose the browser URL search bar.
    - Press **Ctrl** + **v** on your keyboard.

        - **Note**: Mac users should also press **Ctrl** + **v** on their keyboard. This command is not the pasting command for Mac keyboards, but this simulation requires you to use your keyboard as a Windows keyboard.
    - Press **Enter** to load the page.

A report is displayed in the web browser.

If you wait 2 minutes and use the link again, you will find that the URL has expired and no longer works. Note: for the sake of the simulation, you do not need to wait 2 minutes.

52. Choose the Refresh icon on the browser.

    Now that the presigned URL is expired, you get an Access denied page.

53. Choose **x** to close the Access denied tab.

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 so, you apply a bucket policy that denies delete privileges on your website files.

50. Choose the **Permissions** tab.
51. Choose the scroll bar to scroll down to the **Bucket policy** panel.
52. In the **Bucket policy** panel, choose **Edit**.
53. Copy the following policy text and paste it in the **Policy** text editor field. To do so, follow these specific steps:
    - Open the context (right-click) menu for the Policy text editor field.
    - Choose **Paste**.

```
{
"Version": "2012-10-17",
  "Id": "MyBucketPolicy",
  "Statement": [
    {
      "Sid": "BucketPutDelete",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:DeleteObject",
      "Resource": [
        "arn:aws:s3:::sim-website/index.html",
        "arn:aws:s3:::sim-website/script.js",
        "arn:aws:s3:::sim-website/style.css"
        ]
    }
  ]
}
```

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

Note: If you use this code in your own AWS account, you must use the name of your bucket in place of this simulation's **sim-website** bucket.

53. Choose the scroll bar to scroll down.
54. Choose **Save changes.**
55. Return to the **Objects** tab.
56. Select **index.html**.
57. Choose **Delete**.
58. In the **Delete objects** panel, enter `delete` to confirm that you want to remove this file.

    Note: To record your entry, press **Enter** on your keyboard or choose any place outside the entry field.

59. Choose **Delete objects.**
60. Notice that the **index.html** file is listed in the **Failed to delete** pane.

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

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

    Your bucket policy is now protecting your website files from being deleted.

Task 7: Updating the website

Although you have configured a policy to prevent deletion of website files, you can still update the website. You can do so 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.

Next you edit the existing index.html file.

62. On your computer, load the **index.html** file into a text editor (in this simulation, you use Notepad). Follow these specific steps:
    1. Open the context (right-click) menu for the **index.html** file.
    2. Choose **Open with**.
    3. Choose **Notepad**.
63. Find the text **Served from Amazon S3**, and replace it with `Created by Jane`. Follow these specific steps:
    1. Choose the text **Served from Amazon S3.**
    2. Enter `Created by Jane`.

    Note: To record your entry, press **Enter** on your keyboard or choose any place outside the entry field.

64. Save the file. Follow these specific steps:
    1. Choose **File** from the Notepad menu.
    2. Choose **Save**.
65. Return to the **Amazon S3 console** by selecting the **Amazon S3 console** window in the background.

    Now you review the current website version.

66. Choose the **index.html** file name (choose the link, not the checkbox).
67. Choose the **Object URL** link.

    Served from Amazon S3 should still be visible on your website page because you have not yet uploaded the new version and made it public. Next, you will upload the index.html file that you edited and make it public.

68. Choose the **Back** arrow on your browser to return to the **Amazon S3 console**.
69. Choose the **sim-website** link from the navigation at the top of the page.
70. Upload the **index.html** file that you just edited. Follow these specific steps:
    1. Choose **Upload**.
    2. Choose **Add files**.
    3. Choose the **Website files** folder, and choose **Open**.
    4. Choose the **index** file and choose **Open**.
    5. Choose the scroll bar to scroll down.
    6. Choose **Upload**.
    7. Choose **Close**.

71. Select the **index.html** checkbox, and in the **Actions** menu, choose the **Make public using ACL** option again.
72. Choose **Make public**, and choose **Close**.

    Now you verify that your website is updated with your edits.

73. Choose the **index.html** file name (choose the link, not the checkbox).
74. Choose the **Object URL** link.

    Created by Jane should now be on the page in place of *Served from Amazon S3*.

    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.

75. Choose the Back arrow on your browser to return to the Amazon S3 console.
76. Choose the **sim-website** link from the navigation at the top of the page.

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 by 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 as soon as you turn on versioning, you cannot turn it off. However, you can suspend versioning. For more information about bucket versioning, see “Using versioning in S3 buckets” in the *Amazon Simple Storage Service Users Guide* at 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.

77. Choose **Show versions** to see which files have multiple versions.
78. Choose the scroll bar to scroll down.
79. Review the list of objects in the bucket.

- Notice that each file has a **Version ID**. Amazon S3 automatically generates these IDs 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 simulation, 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 you turned on bucket versioning to recover previous versions of files. Excellent work!

Simulation complete

Congratulations! You have completed the simulation.