

EXPERIMENT 6

Aim: To study and Implement Storage as a Service using OwnCloud/ AWS S3, Glaciers/ Azure Storage.

Objective:

- Discuss different types of storage solutions and their features and benefits.
- Discuss the features and concepts of Amazon S3.
- Describe Amazon S3 storage classes and associated use cases.
- Discuss how to use Amazon S3 to create a bucket, upload objects, and work with objects.
- Describe Amazon S3 configurations for cost savings and security.
- Identify other AWS storage solutions and their use cases.
- Use Amazon S3 to create a static website.

Theory: To embark on a study and implementation of Storage as a Service (STaaS) using OwnCloud, AWS S3, Glaciers, or Azure Storage, it's crucial to grasp the foundational concepts and functionalities inherent in these cloud storage platforms. Storage as a Service offers scalable, reliable, and cost-effective storage solutions, eliminating the need for organizations to manage on-premises storage infrastructure. The theory behind this experiment revolves around understanding the core components and features of STaaS offerings provided by OwnCloud, AWS S3, Glaciers, and Azure Storage. These platforms offer a variety of storage options, including object storage, file storage, and archival storage, catering to diverse storage requirements. Key concepts include data durability, availability, scalability, security, and cost-effectiveness. Participants will explore storage service features such as versioning, encryption, access control, and data lifecycle management. Practical exercises will involve provisioning storage buckets/containers, uploading/downloading data, setting access policies, and configuring storage tiers for cost optimization. Additionally, participants will analyze real-world use cases and case studies to understand how organizations leverage STaaS solutions to address storage challenges, improve data management workflows, and enhance business agility. Through this study and implementation, participants will gain hands-on experience in deploying, configuring, and managing cloud storage solutions, enabling them to effectively utilize STaaS offerings to meet storage needs, optimize costs, and drive innovation in their organizations.

Implementation And Output:

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**.

3. 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.
4. For **Object Ownership**, choose **ACLs enabled**.
5. Choose **Bucket owner preferred**.
6. 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**.
7. For **Bucket Versioning**, choose **Enable**.
Note: Once you turn on (enable) bucket versioning, you can't turn it off.
8. For **Tags**, choose **Add tag**, and enter the following:
 - a. **Key:** Department
 - b. **Value:** Marketing
9. You can use tags to add additional information to a bucket, such as a project code, cost center, or owner.
10. Choose **Create bucket**
11. In the **Buckets** section, choose the name of your new bucket.
12. Choose the **Properties** tab.

Workbench - Vocareum

Create S3 bucket | S3 | Global

s3.console.aws.amazon.com/s3/bucket/create?region=us-east-1&bucketType=general

AWS Services Search [Alt+S] Global Guest (2)

Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region: US East (N. Virginia) us-east-1

Bucket type: General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

Directory - New
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name:
Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

Copy settings from existing bucket - optional
Only the bucket settings in the following configurations are copied:

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

94°F Smoke 2:34 PM 2/29/2024

Workbench - Vocareum

Create S3 bucket | S3 | Global

s3.console.aws.amazon.com/s3/bucket/create?region=us-east-1&bucketType=general

AWS Services Search [Alt+S] Global Guest (2)

Encryption type: Server-side encryption with Amazon S3 managed keys (SSE-S3)
 Server-side encryption with AWS Key Management Service keys (SSE-KMS)
 Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the [Amazon S3 pricing page](#).

Bucket Key:
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

Disable
 Enable

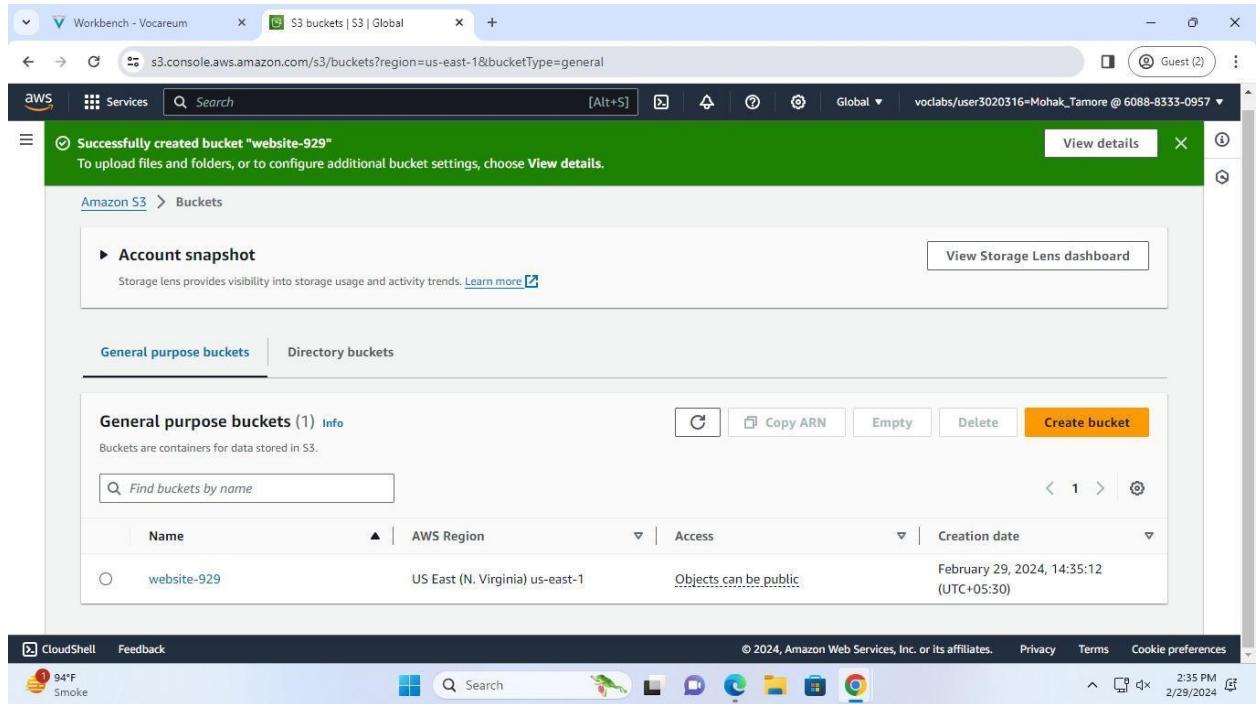
Advanced settings

After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

Cancel **Create bucket**

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

94°F Smoke 2:34 PM 2/29/2024



Task 2: Configuring a static website on Amazon S3

You will now configure the bucket for static website hosting.

13. Scroll to the **Static website hosting** panel.
14. Choose **Edit**
15. Configure the following settings:
 - a. **Static web hosting:** Choose **Enable**.
 - b. **Hosting type:** Choose **Host a static website**.
 - c. **Index document:** Enter `index.html`
 - d. **Error document:** Enter `error.html`
16. **Note:** You must enter `index.html` and `error.html` even though they are already displayed.
17. Choose **Save changes**
18. 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.

Static website hosting

Disable

Enable

Hosting type

Host a static website
Use the bucket endpoint as the web address. [Learn more](#)

Redirect requests for an object
Redirect requests to another bucket or domain. [Learn more](#)

For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see [Using Amazon S3 Block Public Access](#)

Index document

Specify the home or default page of the website.

index.html

Error document - optional

This is returned when an error occurs.

error.html

Redirection rules - optional

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

94°F Smoke

Search

2:36 PM 2/29/2024

Workbench - Vocareum | website-929 - S3 | 403 Forbidden

Not secure website-929.s3-website-us-east-1.amazonaws.com

Guest (2)

403 Forbidden

An Error Occurred While Attempting to Retrieve a Custom Error Document

- Code: AccessDenied
- Message: Access Denied
- RequestId: HA1CYSNZCVM4CEQ5
- HostId: LHBkJ5FzEapmy7xUmPIM0H9nObyri3a3A1KPFghlwLwRs8IClF5i9NvXzA5ihP6g3NkoJg+8V0=

Code: AccessDenied

Message: Access Denied



Task 3: Uploading content to your bucket

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

19. 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](#)
- [script.js](#)
- [style.css](#)

20. Return to the Amazon S3 console, and choose the **Objects** tab.

21. Choose **Upload**

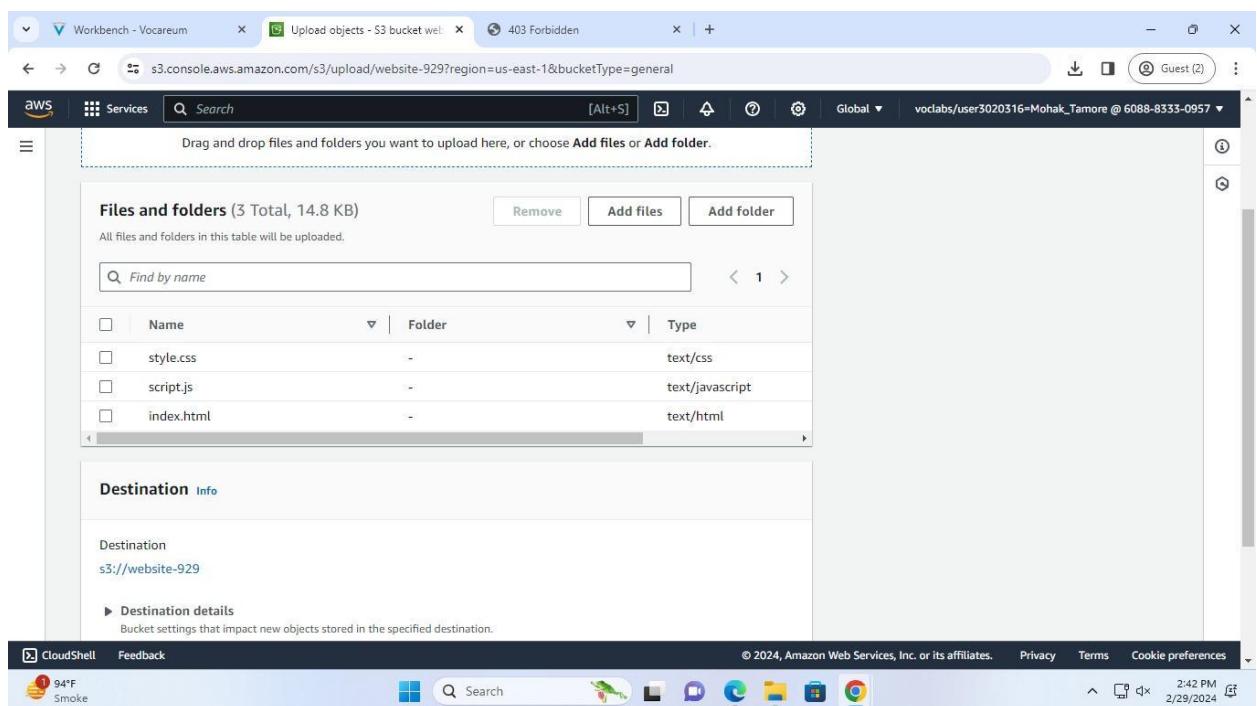
22. Choose **Add files**

23. Choose the three files that you downloaded.

24. Choose **Upload**

Your files are uploaded to the bucket.

25. Choose **Close**



The screenshot shows the AWS S3 console interface. At the top, a message indicates "Upload succeeded" with "View details below." Below this, a summary table shows the destination as "s3://website-929" with 3 files successfully uploaded (14.8 KB, 100.00%) and 0 failed files (0 B, 0%).

| Destination | Succeeded | Failed |
|------------------|----------------------------|-------------------|
| s3://website-929 | 3 files, 14.8 KB (100.00%) | 0 files, 0 B (0%) |

Below the summary, a "Files and folders" section displays the uploaded files:

| Name | Folder | Type | Size | Status | Error |
|------------|--------|-----------------|--------|-----------|-------|
| style.css | - | text/css | 3.1 KB | Succeeded | - |
| script.js | - | text/javascript | 1.9 KB | Succeeded | - |
| index.html | - | text/html | 9.8 KB | Succeeded | - |

At the bottom of the screenshot, a browser window shows the uploaded files at the URL s3.console.aws.amazon.com/s3/buckets/website-929?region=us-east-1&tab=objects.

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.

26. Return to the browser tab that showed the *403 Forbidden* message.
27. 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.

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

A list of the three objects is displayed.

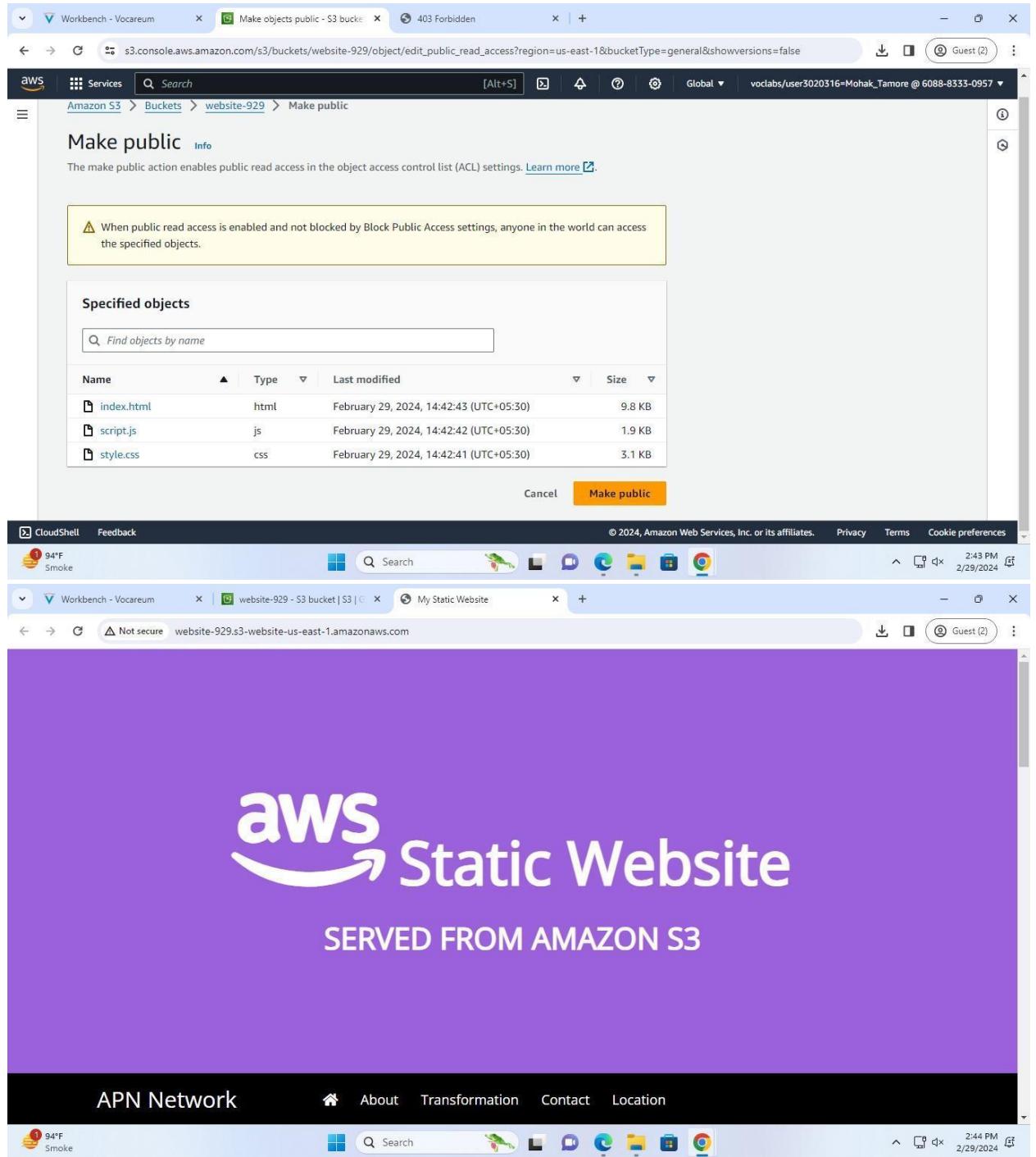
31. Choose **Make public**

Your static website is now publicly accessible.

32. Choose **Close**
33. Return to the web browser tab that has the *403 Forbidden* message.
34. 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.

Choose (right-click) the following link, and download the file to your computer:

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

- [new-report.png](#)

36. Return to the Amazon S3 console, and choose the **Objects** tab.

37. Choose **Upload**

38. Choose **Add files**

39. Choose the file that you downloaded.

40. Choose **Upload**

You have uploaded your file to the bucket.

41. 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.

42. In the **Objects** tab, choose **new-report.png**.

43. From the **Actions** menu, select **Share with a presigned URL**

44. In the pop-up window, configure the **Time interval until the presigned URL expires**:

- a. Choose **Minutes**

- b. For **Number of minutes**, enter 2

45. Choose **Create presigned URL**

46. From the banner at the top of the page, choose **Copy presigned URL**.

47. 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.

Workbench - Vocareum

s3.console.aws.amazon.com/s3/upload/website-929?region=us-east-1&bucketType=general

aws Services Search [Alt+S] Global v vodlabs/user3020316=Mahak_Tamore @ 6088-8333-0957 Guest (2)

Upload succeeded

View details below.

Summary

| Destination | Succeeded | Failed |
|------------------|---------------------------|-------------------|
| s3://website-929 | 1 file, 84.0 KB (100.00%) | 0 files, 0 B (0%) |

Files and folders Configuration

Files and folders (1 Total, 84.0 KB)

| Name | Folder | Type | Size | Status | Error |
|----------------|--------|-----------|---------|-----------|-------|
| new-report.... | - | image/png | 84.0 KB | Succeeded | - |

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 94°F Smoke 2:46 PM 2/29/2024

Workbench - Vocareum

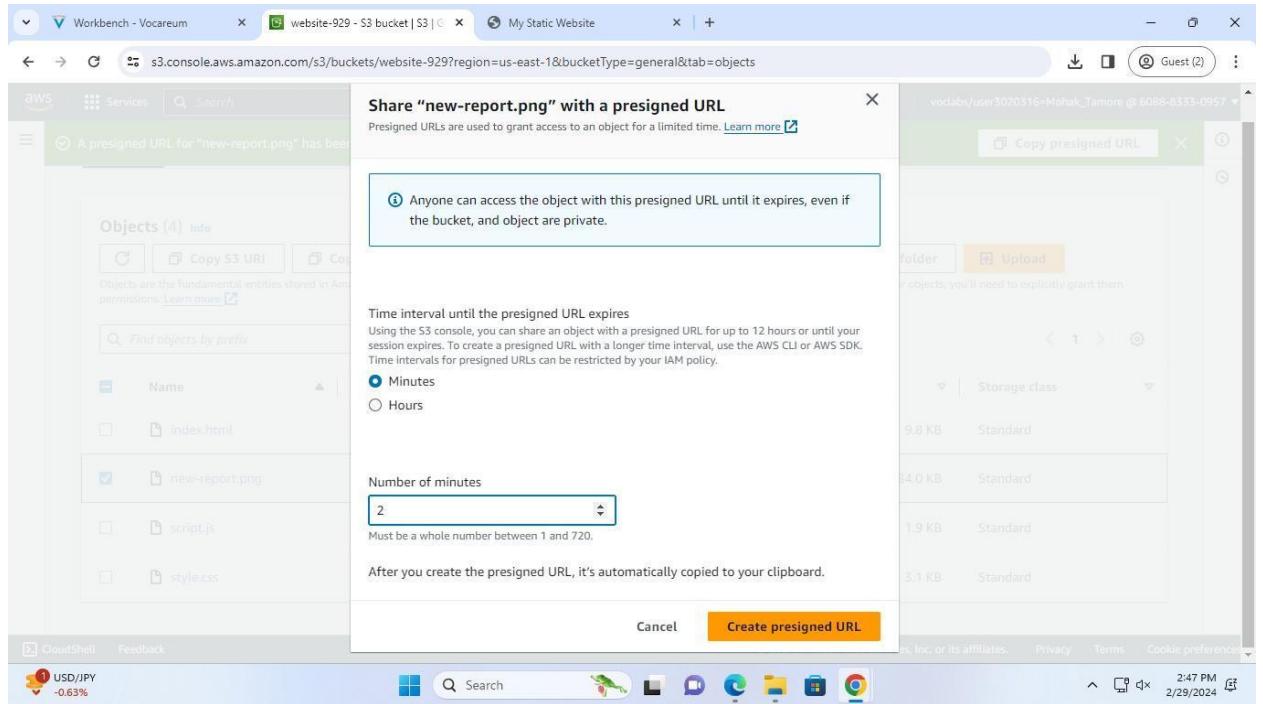
s3.console.aws.amazon.com/s3/buckets/website-929?region=us-east-1&bucketType=general&tab=objects

aws Services Search [Alt+S] Global v vodlabs/user3020316=Mahak_Tamore @ 6088-8333-0957 Guest (2)

Objects (4) Info

| Name | Type | Last modified | Size | Storage class |
|----------------|------|---|---------|---------------|
| index.html | html | February 29, 2024, 14:42:43 (UTC+05:30) | 9.8 KB | Standard |
| new-report.png | png | February 29, 2024, 14:46:07 (UTC+05:30) | 84.0 KB | Standard |
| script.js | js | February 29, 2024, 14:42:42 (UTC+05:30) | 1.9 KB | Standard |
| style.css | css | February 29, 2024, 14:42:41 (UTC+05:30) | 3.1 KB | Standard |

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 94°F Smoke 2:46 PM 2/29/2024



| Service | Operation | UsageType | Resource | StartTime | EndTime | UsageValue |
|----------|--------------------------|-------------------------------|--------------------|-----------------|------------------|------------|
| AmazonS3 | HeadBucket | USW2-C3DataTransfer-Out-Bytes | lab-test-bucket-77 | 10/31/2020 0:00 | 12/31/2020 11:59 | 15309 |
| AmazonS3 | PutObject | USW2-C3DataTransfer-In-Bytes | admin-test-77 | 10/31/2020 0:00 | 12/31/2020 11:59 | 19032 |
| AmazonS3 | HeadBucket | USW2-Requests-Tier2 | admin-test-77 | 10/31/2020 0:00 | 12/31/2020 11:59 | 128 |
| AmazonS3 | PutObjectForReplication | USW1-Request-SIA-Tier1 | mybucket-98765 | 10/31/2020 0:00 | 12/31/2020 11:59 | 56888 |
| AmazonS3 | GetObjectFor Replication | USW1-USW2-AWS-In-Bytes | mybucket-98766 | 10/31/2020 0:00 | 12/31/2020 11:59 | 254587 |
| AmazonS3 | GetObjectFor Replication | USW2-C3DataTransfer-Out-Bytes | mybucket-98767 | 10/31/2020 0:00 | 12/31/2020 11:59 | 235 |
| AmazonS3 | HeadBucket | USW2-C3DataTransfer-In-Bytes | mybucket-98768 | 10/31/2020 0:00 | 12/31/2020 11:59 | 25589 |
| AmazonS3 | PutObject | USW2-Requests-Tier2 | mybucket-98769 | 10/31/2020 0:00 | 12/31/2020 11:59 | 2348 |
| AmazonS3 | PutObjectForReplication | USW1-Request-SIA-Tier1 | mybucket-98770 | 10/31/2020 0:00 | 12/31/2020 11:59 | 15309 |
| AmazonS3 | GetObjectFor Replication | USW1-USW2-AWS-In-Bytes | mybucket-98771 | 10/31/2020 0:00 | 12/31/2020 11:59 | 19032 |
| AmazonS3 | GetObjectFor Replication | USW2-C3DataTransfer-Out-Bytes | lab-example-bucket | 10/31/2020 0:00 | 12/31/2020 11:59 | 128 |
| AmazonS3 | HeadBucket | USW2-C3DataTransfer-In-Bytes | lab-example-bucket | 10/31/2020 0:00 | 12/31/2020 11:59 | 56888 |
| AmazonS3 | PutObject | USW2-Requests-Tier2 | lab-example-bucket | 10/31/2020 0:00 | 12/31/2020 11:59 | 254587 |
| AmazonS3 | PutObjectForReplication | USW1-Request-SIA-Tier1 | lab-example-bucket | 10/31/2020 0:00 | 12/31/2020 11:59 | 235 |
| AmazonS3 | GetObjectFor Replication | USW1-USW2-AWS-In-Bytes | lab-example-bucket | 10/31/2020 0:00 | 12/31/2020 11:59 | 25589 |

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.

48. Return to the Amazon S3 console, and choose the **Permissions** tab.

49. Under **Bucket policy**, choose **Edit**

50. 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.

51. 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"

52. 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"

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

54. Choose **Save changes**

55. Return to the **Object 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.

59. Choose **Delete objects**

60. 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.

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

The screenshot shows the AWS S3 Bucket Policy editor for a bucket named 'website-929'. The 'Policy' section displays the following JSON code:

```

1 Version: "2012-10-17",
2 "Id": "MyBucketPolicy",
3 "Statement": [
4     {
5         "Sid": "BucketPutDelete",
6         "Effect": "Deny",
7         "Principal": "*",
8         "Action": "s3:DeleteObject",
9         "Resource": [
10             "arn:aws:s3:::website-929/index.html",
11             "arn:aws:s3:::website-929/script.js",
12             "arn:aws:s3:::website-929/style.css"
13         ]
14     }
15 ]
16
17 }

```

To the right of the policy editor, there is a sidebar with the following interface:

- Edit statement**: A text input field.
- Select a statement**: A placeholder text: "Select an existing statement in the policy or add a new statement."
- Add new statement**: A button.

The browser tabs at the top show 'Workbench - Vocareum', 'Edit bucket policy - S3 bucket...', 'My Static Website', and 'new-report.png (1045x602)'. The bottom navigation bar includes 'CloudShell', 'Feedback', and links to '© 2024, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

The screenshot shows the AWS S3 console interface. A modal dialog box is open, prompting the user to confirm the deletion of an object named 'index.html'. The dialog includes a message about adding delete markers, a search bar, and a table listing the object. Below the table is a text input field containing the word 'delete' for confirmation. At the bottom right of the dialog are 'Cancel' and 'Delete objects' buttons.

| Name | Type | Last modified | Size |
|------------|------|---|--------|
| index.html | html | February 29, 2024, 14:42:43 (UTC+05:30) | 9.8 KB |

Delete objects?

To confirm deletion, type *delete* in the text input field.

delete

Cancel Delete objects

The screenshot shows the AWS S3 console after a failed deletion attempt. A red banner at the top indicates 'Failed to delete objects'. Below it, a 'Summary' section shows the source as 's3://website-929' and lists 'Successfully deleted' 0 objects and 'Failed to delete' 1 object (9.8 KB). The 'Failed to delete' row has a red error icon next to it. At the bottom, a table titled 'Failed to delete (1 object, 9.8 KB)' shows the single failed entry with an 'Access denied' error message in the 'Error' column.

| Name | Folder | Type | Last modified | Size | Error |
|------------|--------|------|---|--------|---------------|
| index.html | - | html | February 29, 2024, 14:42:43 (UTC+05:30) | 9.8 KB | Access denied |

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

NH48 / Sh4 / M... Construction Search 2:50 PM 2/29/2024

The screenshot shows the AWS S3 console after a successful update. The 'Failed to delete' table now shows a single row with a green checkmark icon and the status 'Success' in the 'Error' column. The rest of the interface remains the same, showing the static website configuration and the 'Configuration' tab selected.

| Name | Folder | Type | Last modified | Size | Error |
|------------|--------|------|---|--------|---------|
| index.html | - | html | February 29, 2024, 14:42:43 (UTC+05:30) | 9.8 KB | Success |

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

NH48 / Sh4 / M... Construction Search 2:50 PM 2/29/2024

Task 7: Updating the website

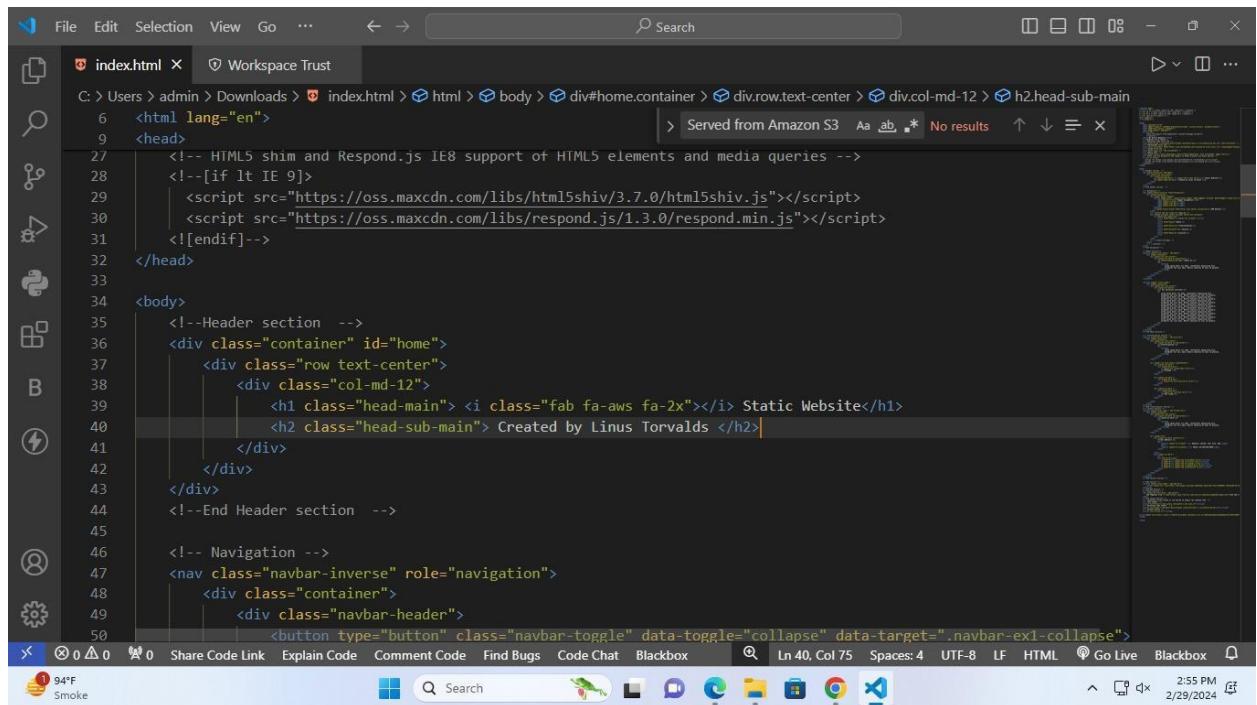
Although you have configured a policy to prevent deletion of website files, you can still update the website by editing the HTML file and uploading it to the S3 bucket again.

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.

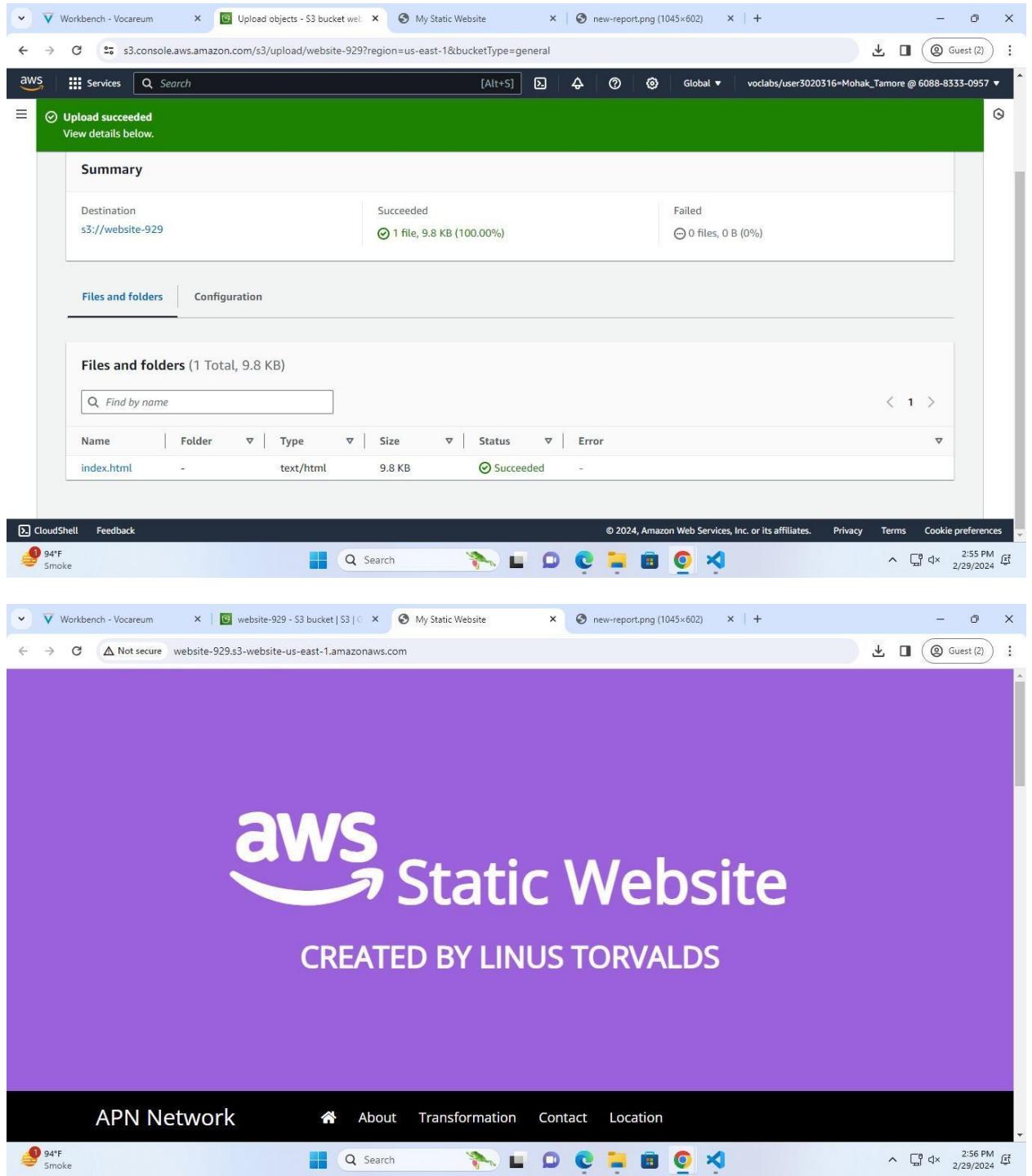
62. On your computer, load the **index.html** file into a text editor (for example, Notepad orTextEdit).
63. Find the text **Served from Amazon S3**, and replace it with **Created by <YOUR-NAME>**and substitute your name for (for example, **Created by Jane**).
64. Save the file.
65. Return to the Amazon S3 console, and upload the **index.html** file that you just edited.
66. Choose **index.html**, and in the **Actions** menu, choose the **Make public using ACL**option again.
67. Choose **Make public**.
68. 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.



```
C: > Users > admin > Downloads > index.html > html > body > div#home.container > div.row.text-center > div.col-md-12 > h2.head-sub-main
6   <html lang="en">
9   <head>
27   |   <!-- HTML5 shim and Respond.js IE8 support of HTML5 elements and media queries -->
28   |   <!--[if lt IE 9]>
29   |   |   <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
30   |   |   <script src="https://oss.maxcdn.com/libs/respond.js/1.3.0/respond.min.js"></script>
31   |   <![endif]-->
32   </head>
33
34   <body>
35   |   <!--Header section -->
36   |   <div class="container" id="home">
37   |   |   <div class="row text-center">
38   |   |   |   <div class="col-md-12">
39   |   |   |   |   <h1 class="head-main"> <i class="fab fa-aws fa-2x"></i> Static Website</h1>
40   |   |   |   |   <h2 class="head-sub-main"> Created by Linus Torvalds </h2>
41   |   |   </div>
42   |   </div>
43   |   </div>
44   |   <!!--End Header section -->
45
46   |   <!-- Navigation -->
47   |   <nav class="navbar-inverse" role="navigation">
48   |   |   <div class="container">
49   |   |   |   <div class="navbar-header">
50   |   |   |   |   <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-ex1-collapse">
```



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.

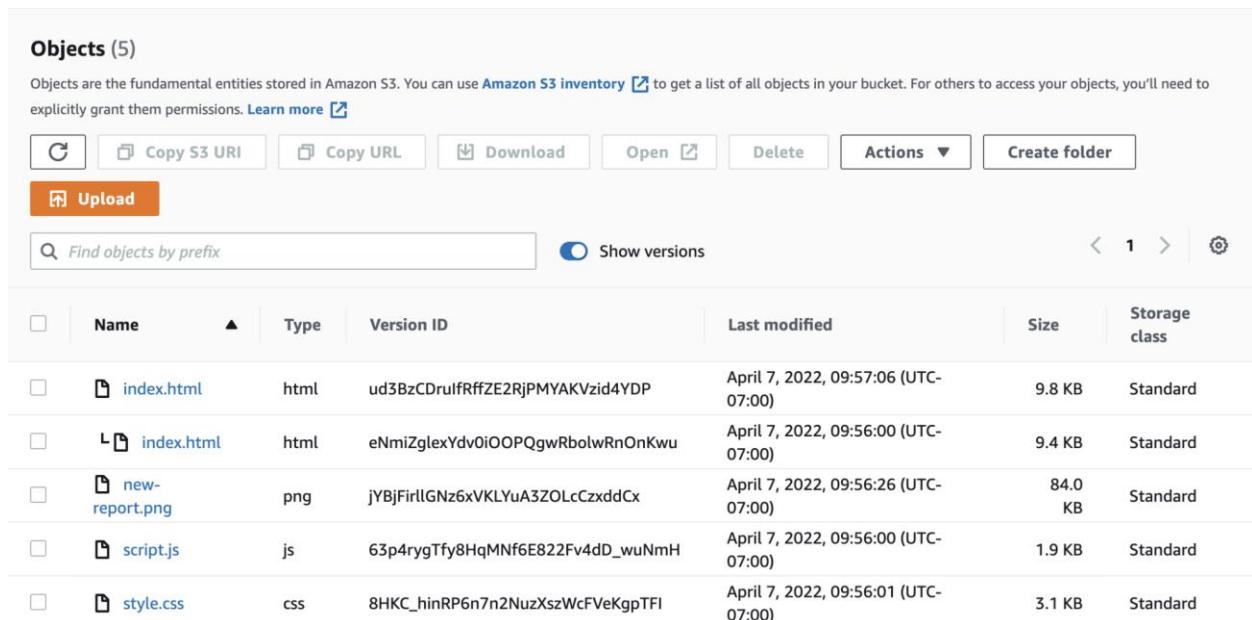
It is important to remember that once you turn on version, you cannot turn it off. However, you can suspend versioning. For more information on bucket versioning, see the [Amazon Simple Storage Service Users Guide](#).

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

69. Return to the Amazon S3 console, and choose the **Objects** tab.

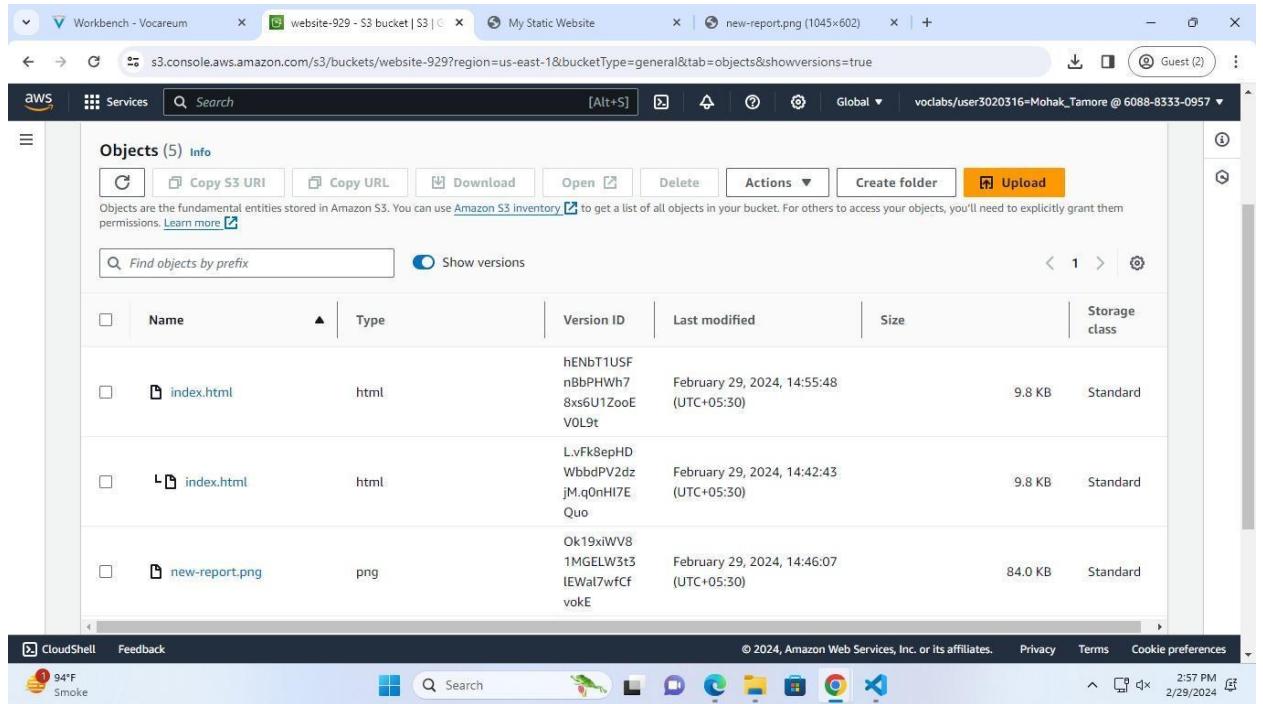
70. Choose **Show versions** to turn on bucket versioning.

71. Review the list of objects in the bucket.



| Objects (5) | | | | | | |
|--|--------------------------------|------|----------------------------------|-------------------------------------|---------|---------------|
| Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more | | | | | | |
| | Name | Type | Version ID | Last modified | Size | Storage class |
| <input type="checkbox"/> | index.html | html | ud3BzCDrlffRffZE2RjPMYAKVzid4YDP | April 7, 2022, 09:57:06 (UTC-07:00) | 9.8 KB | Standard |
| <input type="checkbox"/> | index.html | html | eNmiZglexYdv0iOOPQgwRbolwRnOnKwu | April 7, 2022, 09:56:00 (UTC-07:00) | 9.4 KB | Standard |
| <input type="checkbox"/> | new-report.png | png | jYBjFirllGNz6xVKLYuA3ZOLcCzxddCx | April 7, 2022, 09:56:26 (UTC-07:00) | 84.0 KB | Standard |
| <input type="checkbox"/> | script.js | js | 63p4rygTfy8HqMNf6E822Fv4dD_wuNmH | April 7, 2022, 09:56:00 (UTC-07:00) | 1.9 KB | Standard |
| <input type="checkbox"/> | style.css | css | 8HCK_hinRP6n7n2NuzXszWcFVeKgpTFI | April 7, 2022, 09:56:01 (UTC-07:00) | 3.1 KB | Standard |

- 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.



Conclusion: In conclusion, the exploration and implementation of Storage as a Service (STaaS) using AWS S3 have provided valuable insights into the capabilities and benefits of cloud-based storage solutions. Throughout this endeavor, we have delved into the core functionalities of Amazon S3, understanding its role in providing scalable, durable, and highly available object storage. By provisioning and configuring S3 buckets, we have gained practical experience in uploading, downloading, and managing data securely in the cloud. Real-world examples have showcased how organizations leverage S3 to store and access vast amounts of data, streamline data management processes, and enhance operational efficiency. As we reflect on our journey, it becomes evident that Amazon S3 offers a reliable and cost-effective solution for addressing storage challenges, facilitating data-driven decision-making, and supporting business growth. Armed with this knowledge and experience, we are better equipped to harness the power of AWS S3 effectively, optimizing storage resources, and driving innovation in our organizations.