**Cloud Computing Lab**

**Experiment No.: 4**

**Host a Static Website on Cloud**

**Subject: Cloud Computing Laboratory (DJ19DSL6011)**

**AY: 2022-23**

**Experiment 4**

**Name : Sarvagya Singh**

**SAPID : 60009200030 BATCH : K1**

**Aim: Host a Static Website on Cloud**

* Objectives:
  + 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.
* Outcomes: The learner will be able to create a personalized, publicly accessible static website and how use a presigned URL to share objects in your bucket temporarily. The learner will also protect his work with a bucket policy that prevents file deletion and turns on bucket versioning in case he needs to recover previous versions of files.
* Hardware / Software Required: Internet, AWS console

**Theory:**

Amazon S3 protects with versioning capability. Through versioning, you can save, retrieve, and restore every version of every object stored in your Amazon S3 bucket. This allows you to recover from unintended user actions and application failures quickly. It is a great backup mechanism. With the help of versioning, you can return to your previous version.

**Why do we need versioning in AWS S3?**

We use the versioning feature in the AWS S3 bucket to make multiple copies of the same file and ensure the original file’s availability. Multiple copies help keep the file modification record in the same bucket. Versioning of a file helps to recover the original file anytime because multiple files have the same name with different version IDs. These different versions indicate any modification made to the file.

**Benefits of Versioning**

The key benefits of versioning in the S3 bucket are that when you upgrade the file, it is saved in a versioned form, and you can have an option to get the original file whenever you want. Some other benefits of versioning are:

* + - It works as a backup tool
    - It provides the whole history of the file from original to upgrade copies
    - It provides an extra layer of security by using the Multi-Factor Authentication method

**Features of Versioning**

* + - Object Creation: Within the AWS S3 bucket, you can easily create an object by simply uploading any file
    - Creation of Version: If you want to do some modification to the uploaded file, the new file with modification could be saved with a different version ID in an S3 bucket
    - Deletion of Object: When you delete the uploaded file, the modified versions remain in the bucket.
    - Delete Marker: The deleted file will become a new version with the “Delete Marker” label. The file would not contain any data
    - Restore the Previous Version: If you saved the last copy of a file without any changes, the copied file would be saved with a different version ID. This means you could have duplicate copies of the file with different versions
    - Versioning Suspended: You cannot delete the version. You can only suspend versions. When you add a new copy of the file, the current file is saved with an empty version ID. When you delete this copy, the copied file with the empty version ID is shown with “Delete Marker.”
    - MFA Delete: Multi-Factor Authentication is used as an additional security layer to the bucket versioning. In order to permanently delete the object (file) or perform changes in versioning, some extra authentication is required in your standard security. The password generated by hardware or MFA device is temporary, an authentication code. The root account has the right to enable MFA Delete.

**Steps:**

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

* + - In the AWS Management Console, on the Services menu, choose S3.
    - Choose Create bucket. You use a bucket name for this lab that includes a random number, such as website-123.
    - For the Bucket name, enter the website-<123> and replace <123> with a random number.
    - For Object Ownership, choose ACLs enabled.
    - Choose Bucket owner preferred.
    - 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.
    - For Bucket Versioning, choose Enable.
    - For Tags, choose to Add tag, and enter the following: Key: Department, Value: Marketing
    - Choose Create bucket
    - In the Buckets section, choose the name of your new bucket.
    - Choose the Properties tab.

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

* + - Scroll to the Static website hosting panel.
    - Choose Edit
    - Configure the following settings:

1. Static web hosting: Choose Enable.
2. Hosting type: Choose Host a static website.
3. Index document: Enter index.html
4. Error document: Enter error.html
   * Choose Save changes
   * 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.

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

* + - Choose (right-click) each of the following links, and download the files to your computer: index.html, script.js, style.css
    - Return to the Amazon S3 console, and choose the Objects tab.
    - Choose Upload
    - Choose Add files
    - Choose the three files that you downloaded.
    - Choose Upload. Your files are uploaded to the bucket.
    - Choose Close

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

* + - Return to the browser tab that showed the 403 Forbidden message.
    - Refresh the webpage.
    - Keep the website tab open, and return to the web browser tab with the Amazon S3 console.
    - Choose all three objects.
    - In the Actions menu, choose Make public using ACL.
    - Choose Make public
    - Choose Close
    - Return to the web browser tab that has the 403 Forbidden message.
    - Refresh the webpage.

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

* + - Choose (right-click) the following link, and download the file to your computer: new-report.png
    - Return to the Amazon S3 console, and choose the Objects tab.
    - Choose Upload
    - Choose Add files
    - Choose the file that you downloaded.
    - Choose Upload
    - Choose Close
    - In the Objects tab, choose new-report.png.
    - From the Actions menu, select Share with a presigned URL
    - In the pop-up window, configure the Time interval until the presigned URL expires: Choose Minutes, For Number of minutes, enter 2
    - Choose Create presigned URL
    - From the banner at the top of the page, choose Copy presigned URL.
    - 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**

* + - Return to the Amazon S3 console, and choose the Permissions tab.
    - Under Bucket policy, choose Edit
    - Copy the following policy text. In the Policy text editor, replace the existing policy text with this text:
    - 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.
    - Choose Save changes
    - Return to the Object tab
    - Select index.html.
    - Choose Delete.
    - In the Delete objects panel, enter delete to confirm that you want to remove this file.
    - Choose Delete objects
    - Notice that the index.html file is listed in the Failed to delete pane.
    - Choose Close to return to the Objects tab.

**Task 7) Updating the website**

* + On your computer, load the index.html file into a text editor (for example, Notepad or TextEdit).
  + 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).

* + Save the file.
  + Return to the Amazon S3 console, and upload the index.html file that you just edited.
  + Choose index.html, and in the Actions menu, choose the Make public using ACL option again.
  + Choose Make public.
  + Return to the web browser tab with the static website, and refresh the page.

**Task 8) Exploring file versions**

* + Return to the Amazon S3 console, and choose the Objects tab.
  + Choose Show versions to turn on bucket versioning.
  + 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.

**Conclusion:**

Versioning of files helps to recover the original file anytime. When we enable versioning in the AWS S3 bucket, we can quickly create different copies of the same file in the bucket, and these multiple copies help to keep the record of file modification in the same bucket. These different versions indicate any modification made to the file. The key benefits of versioning in the S3 bucket are; we can upgrade the file that is saved in a versioned form, and we have an option to get the original file whenever we want.

**Viva Questions:**

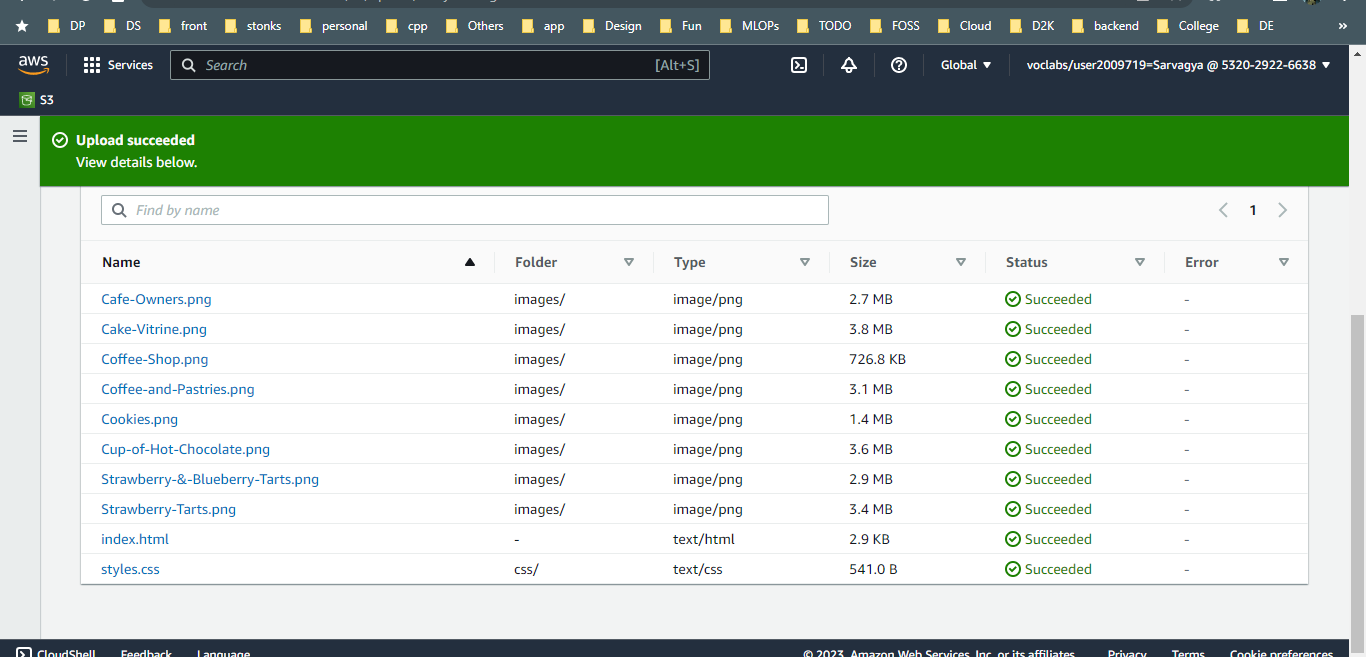
* + What is versioning, and why should I use it?
  + Explain S3 Versioning. What are the benefits of using versioning in S3?
  + How to Configure Versioning on a Bucket?

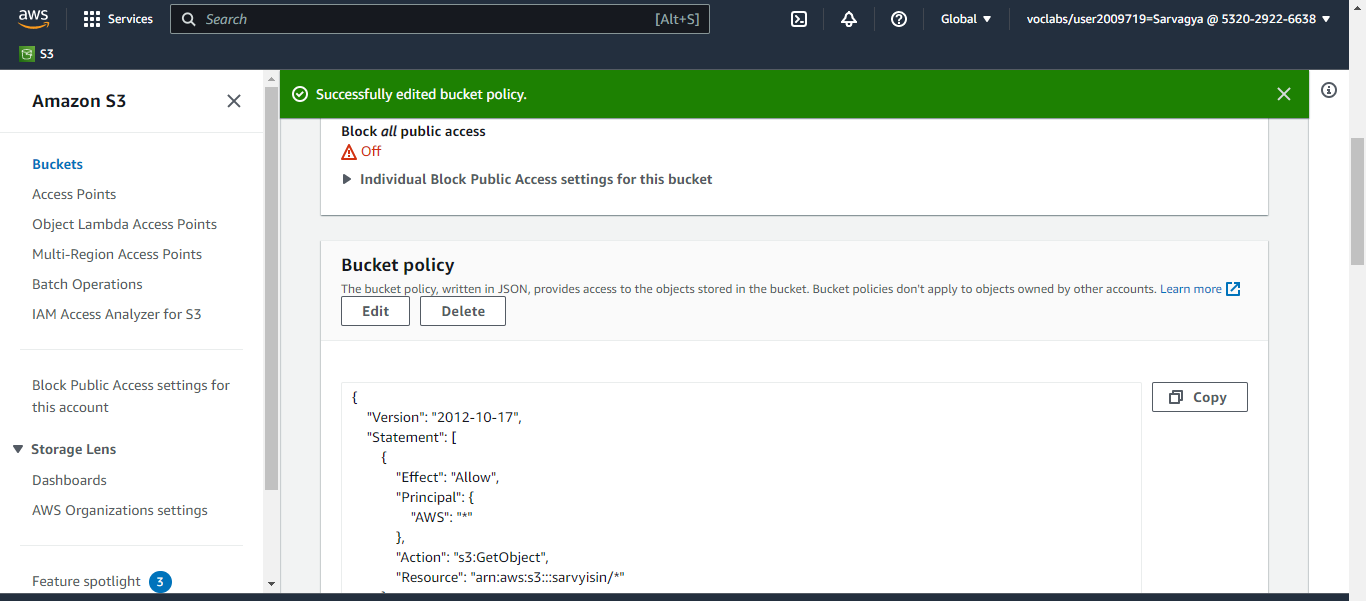
**References:** [**https://docs.aws.amazon.com/AmazonS3/latest/userguide/Versioning.html**](https://docs.aws.amazon.com/AmazonS3/latest/userguide/Versioning.html)

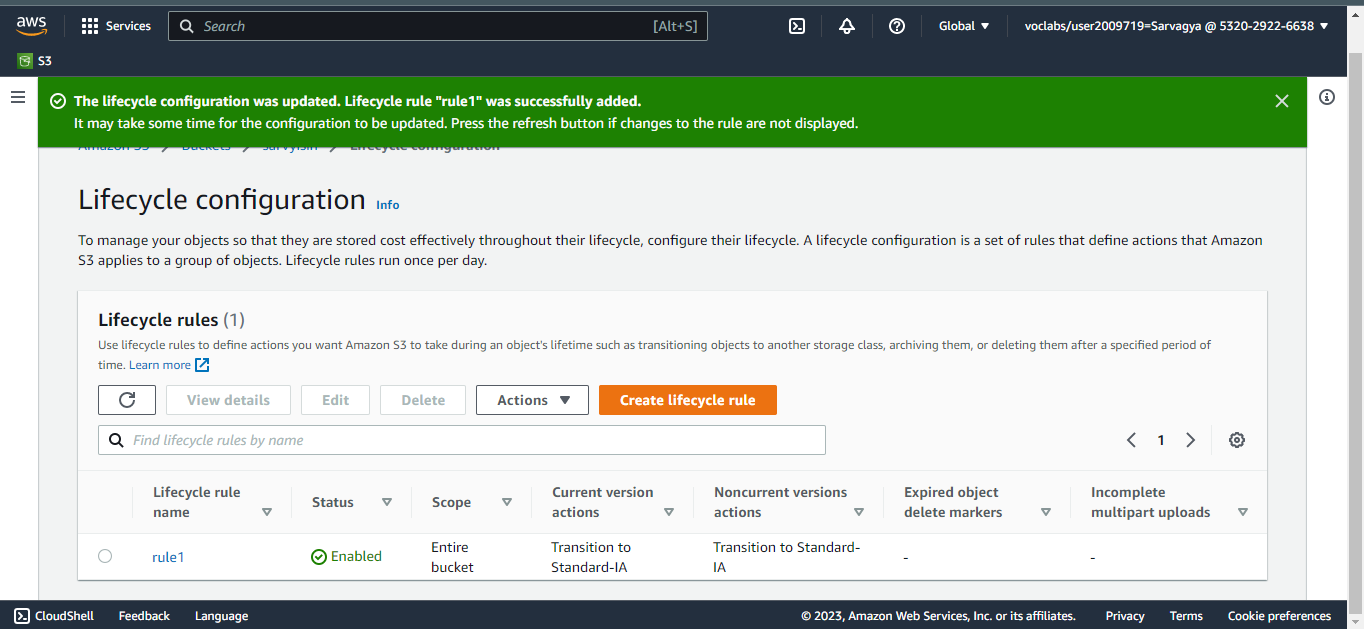
<https://www.youtube.com/watch?v=WipAKAHaDcM>

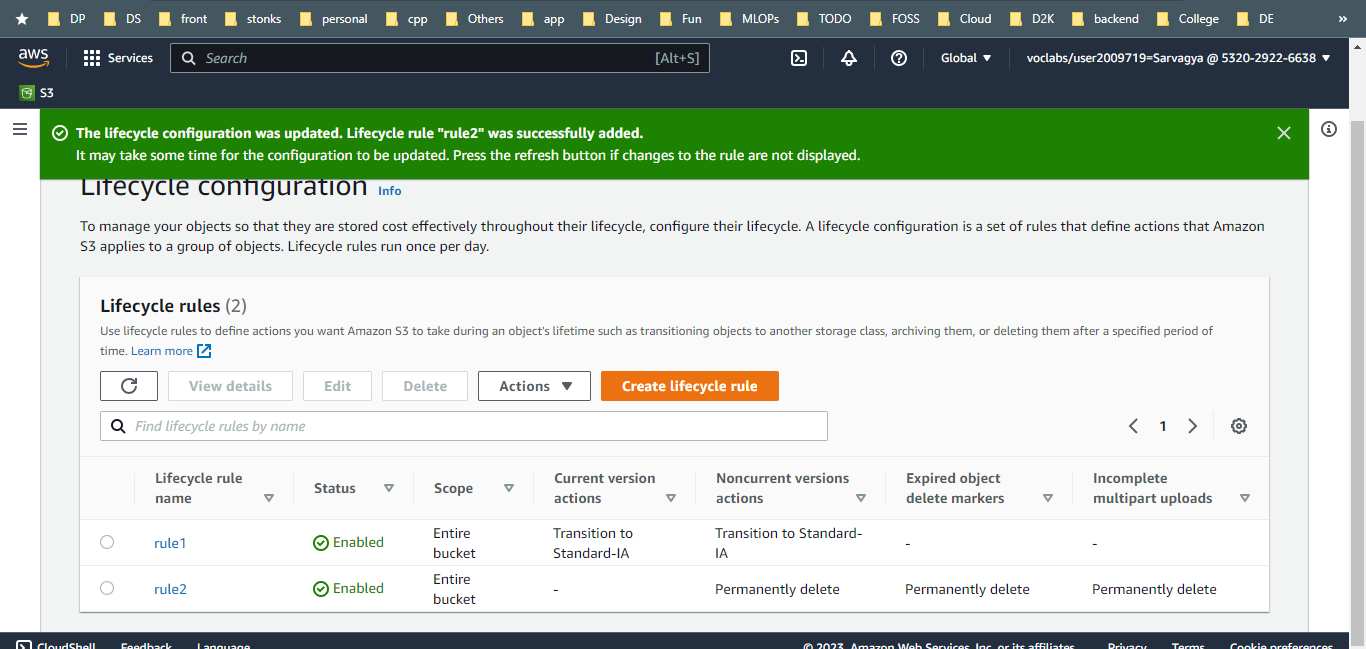
**Screenshots:**

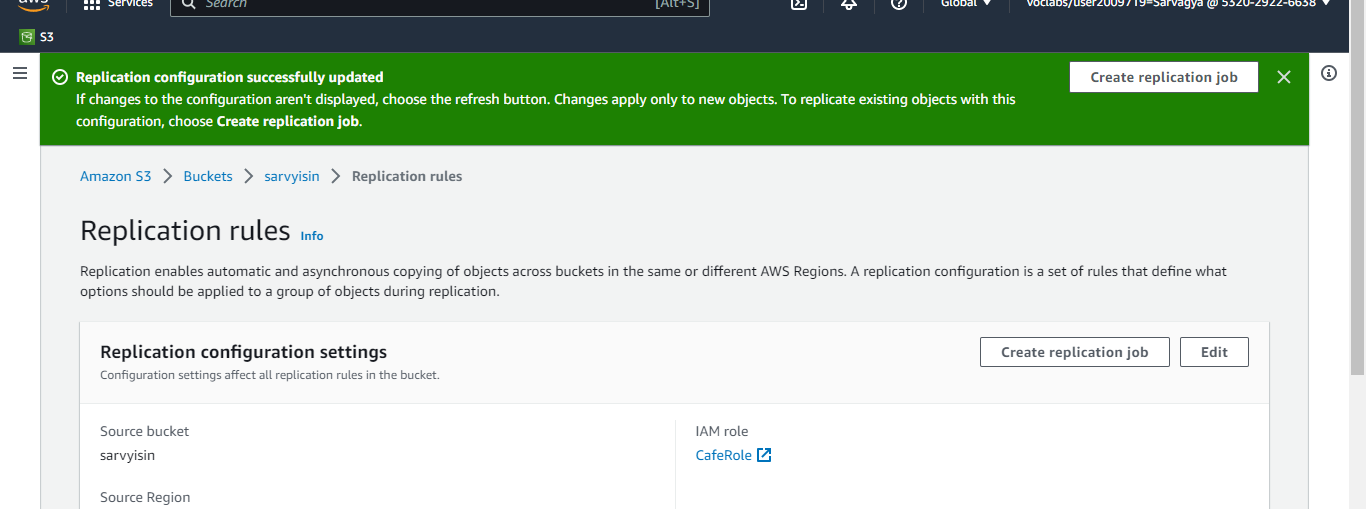
**Challenge lab:**

****

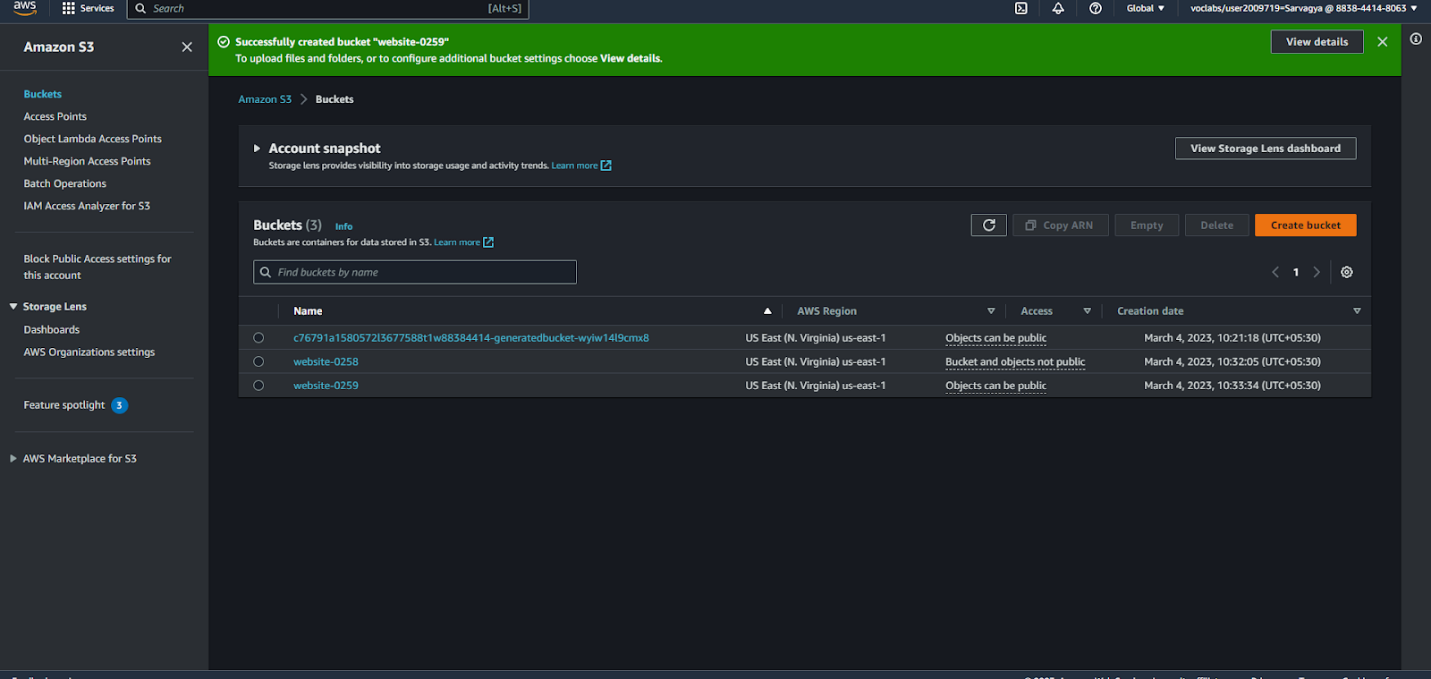
****

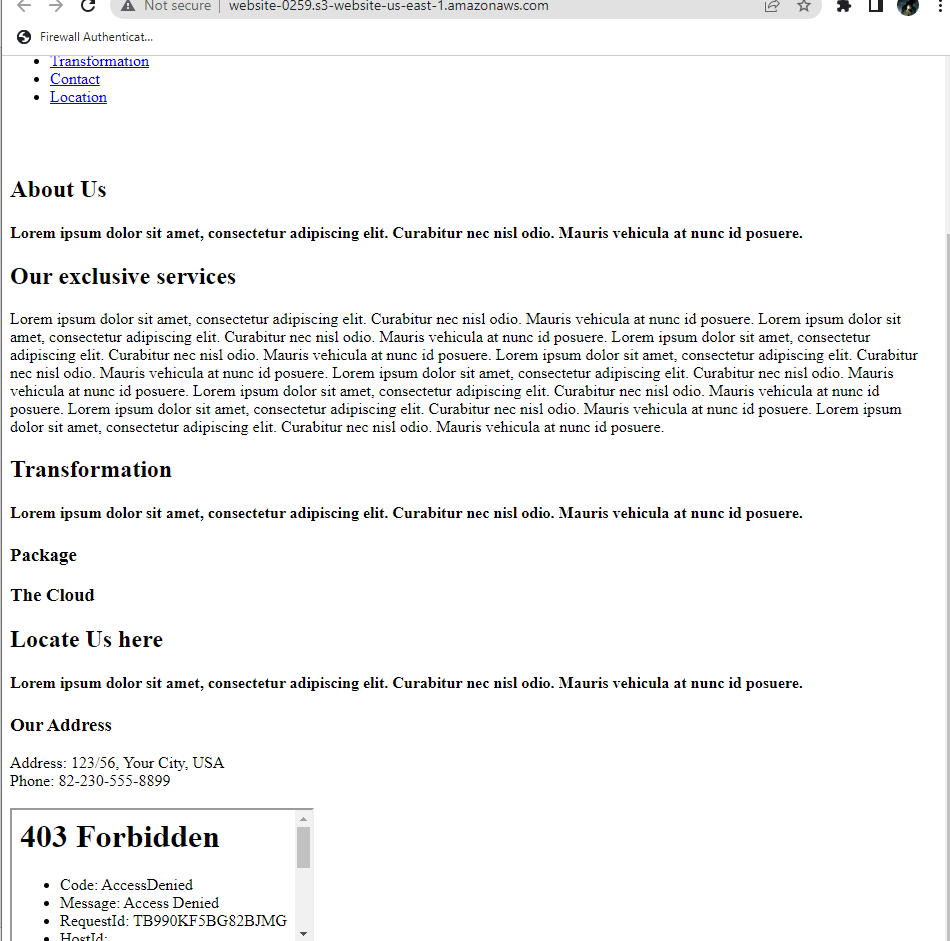
****

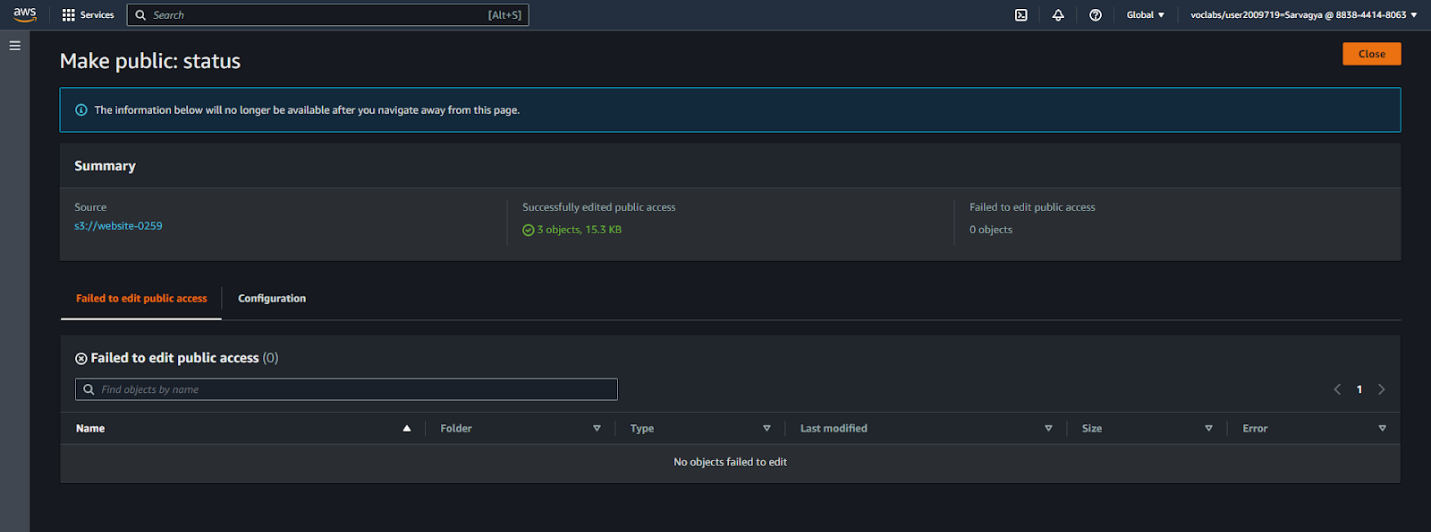
****

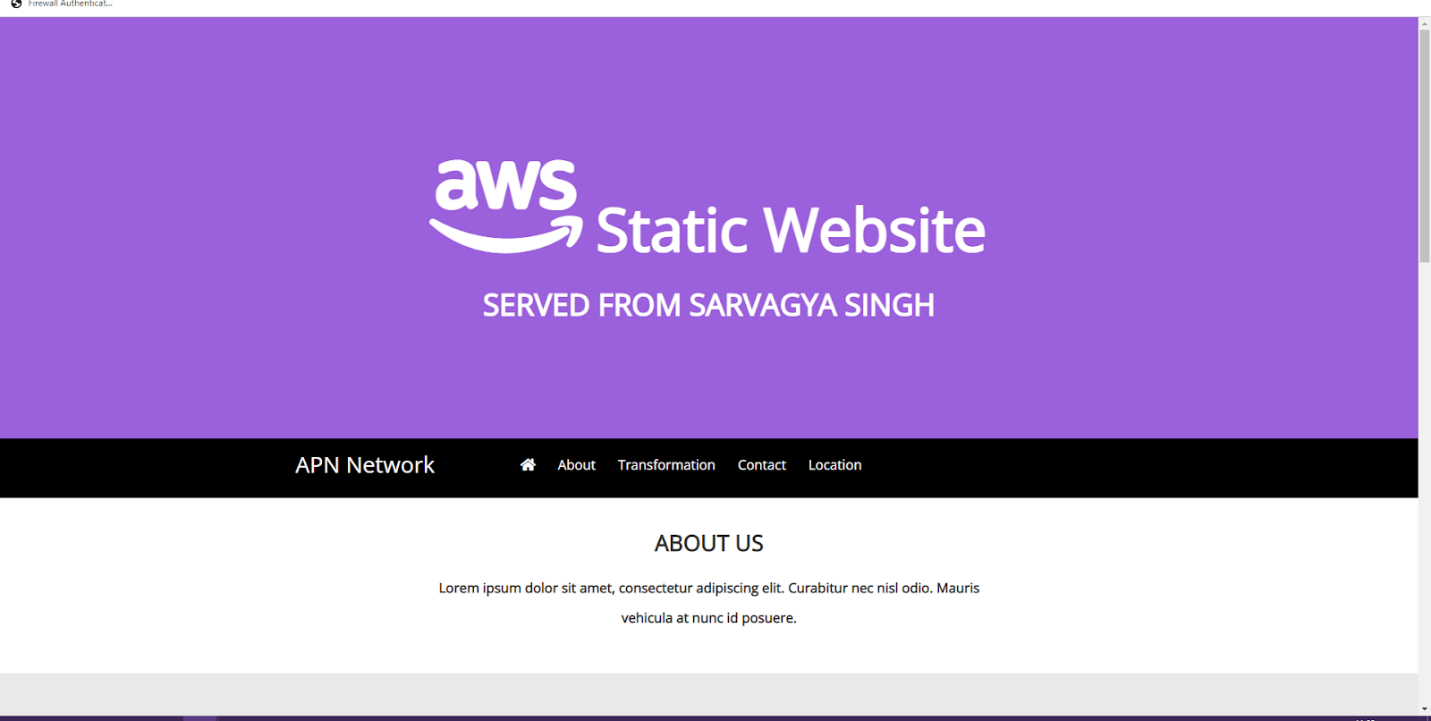
****

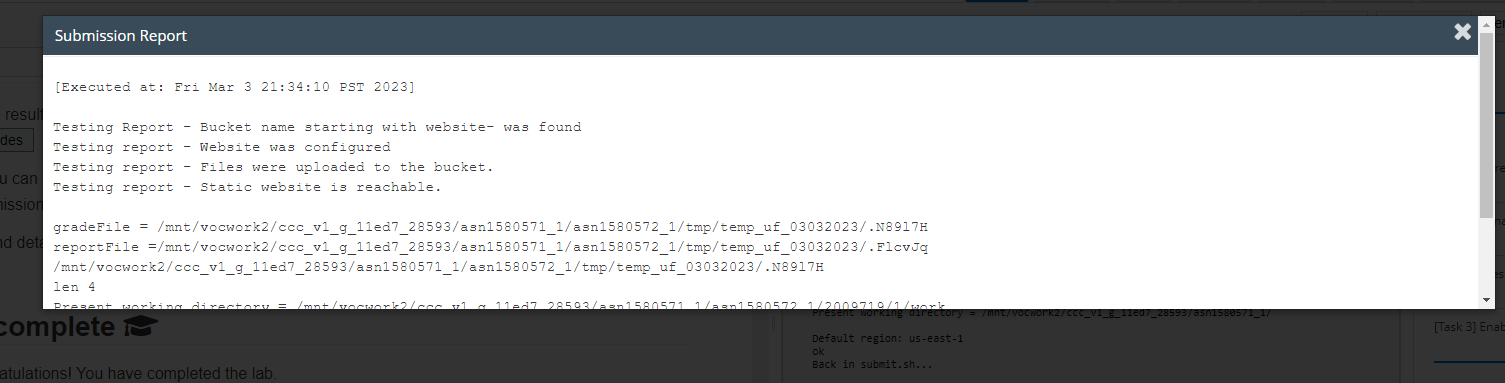
**Guided lab:**

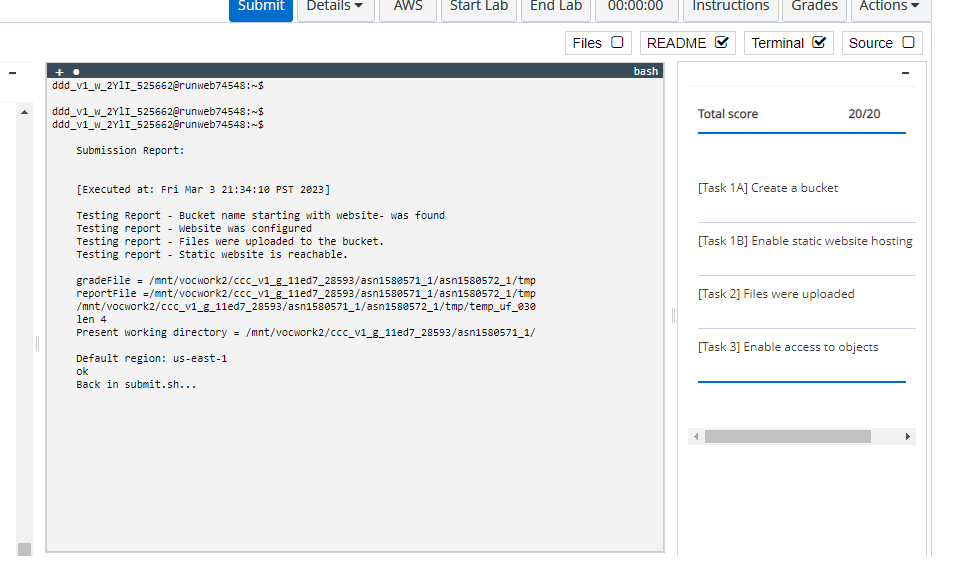
****

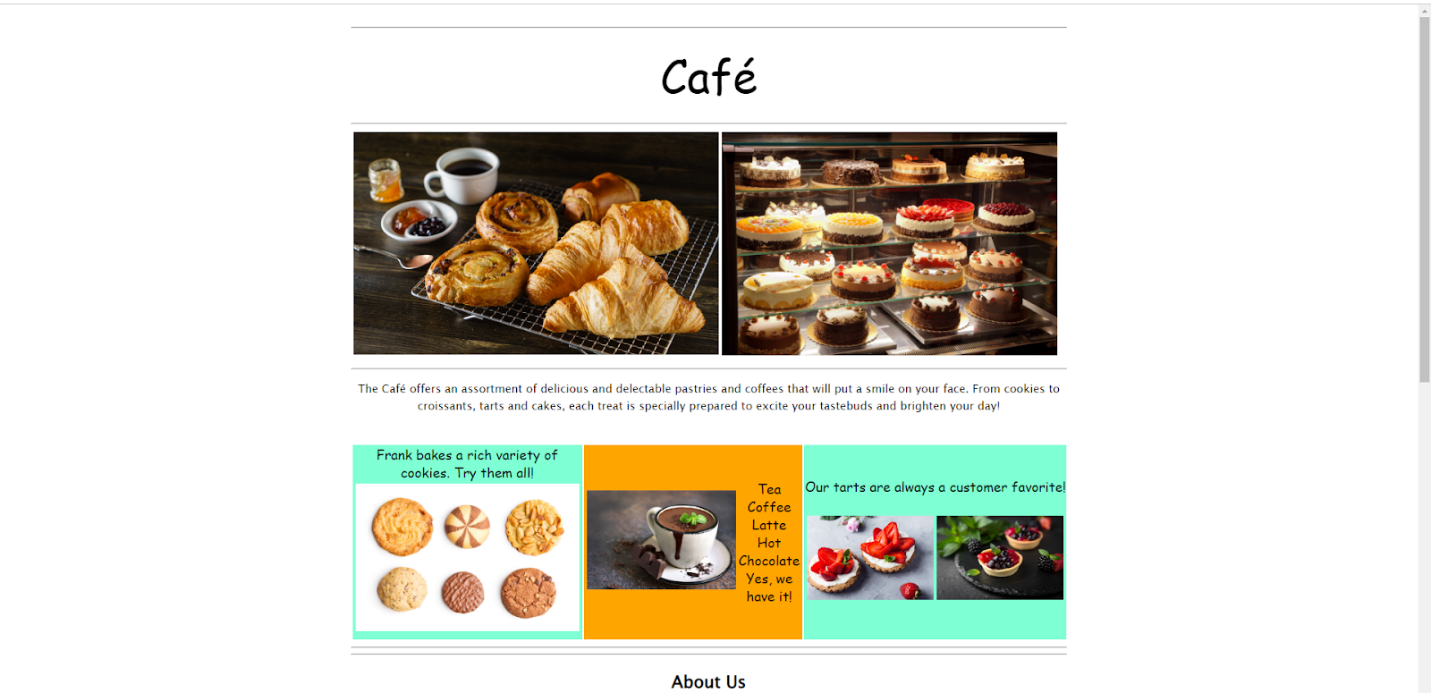
****

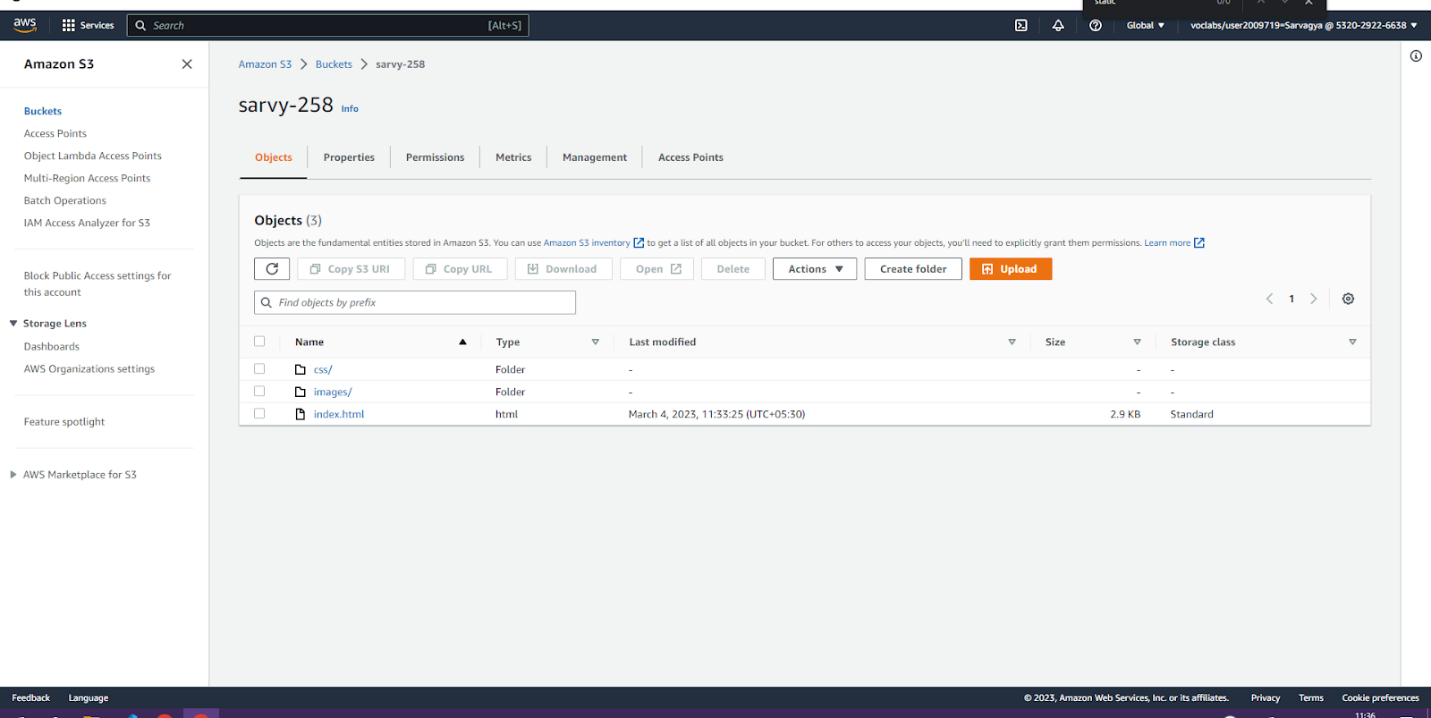
****

****

****

****

****

****

**Add a bucket policy that makes your bucket content publicly available**

{

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

"Statement": [

{

"Sid": "PublicReadGetObject",

"Effect": "Allow",

"Principal": "\*",

"Action": [

"s3:GetObject"

],

"Resource": [

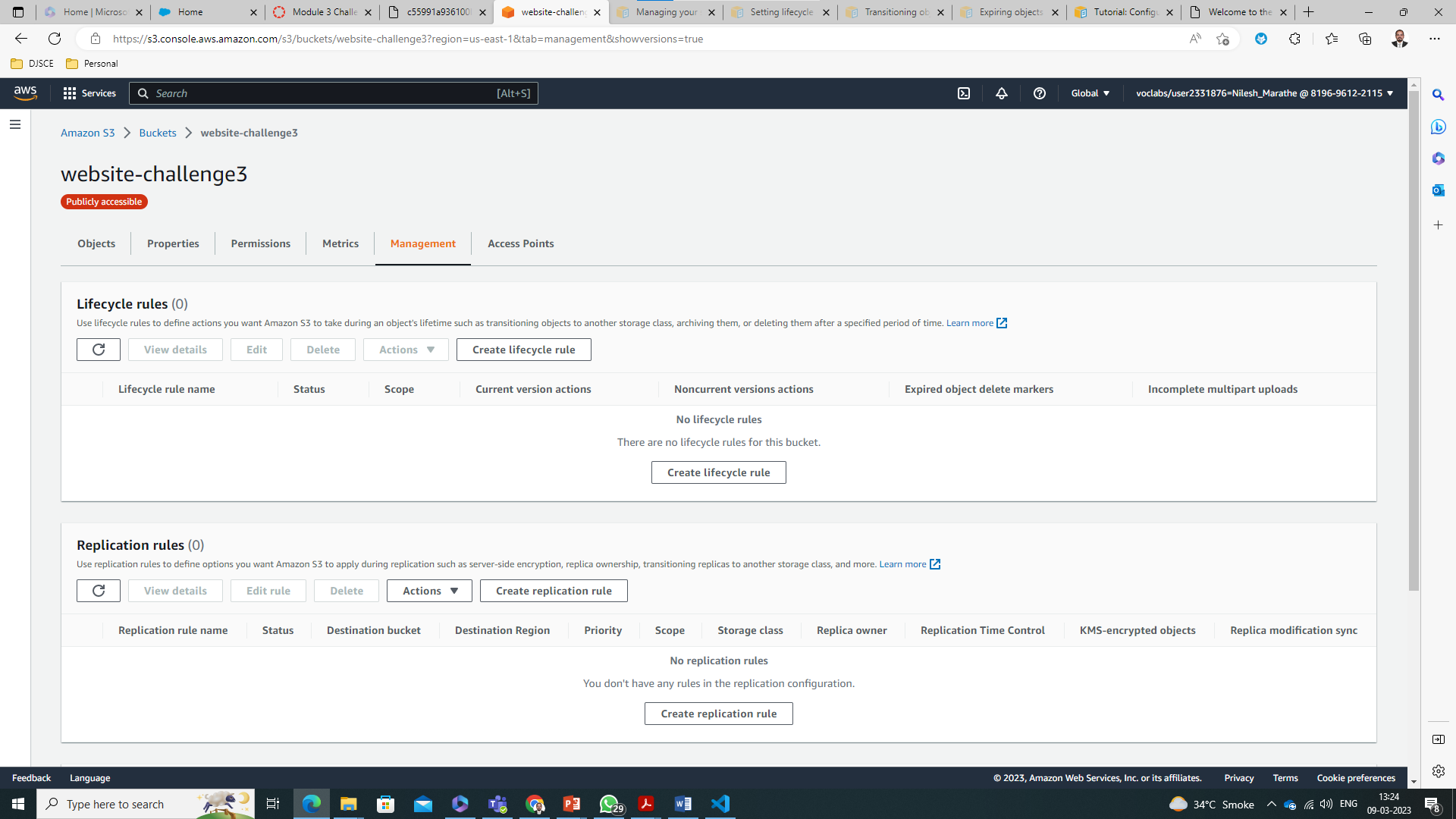
"arn:aws:s3:::*Bucket-Name*/\*"

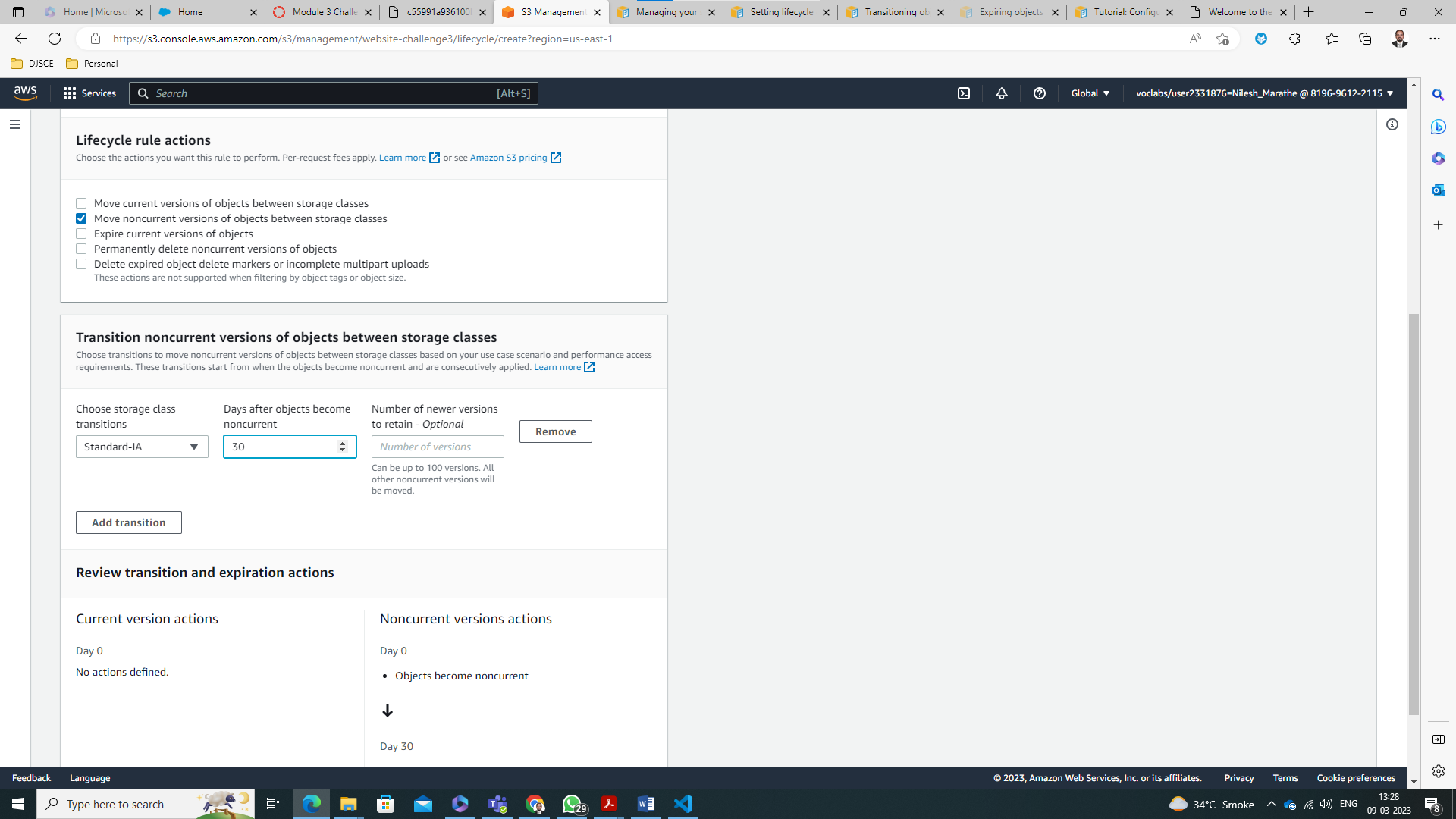
]

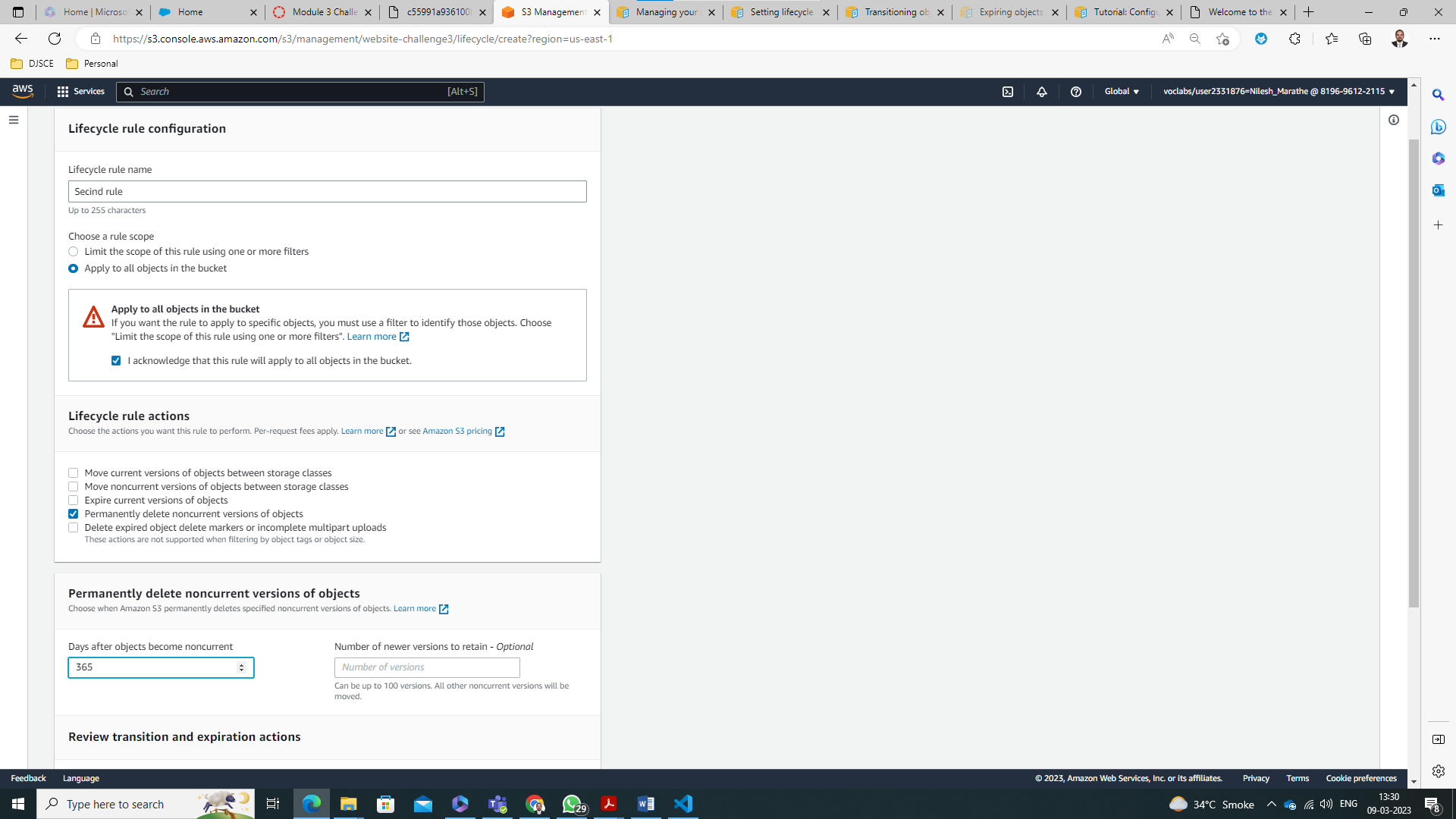
}

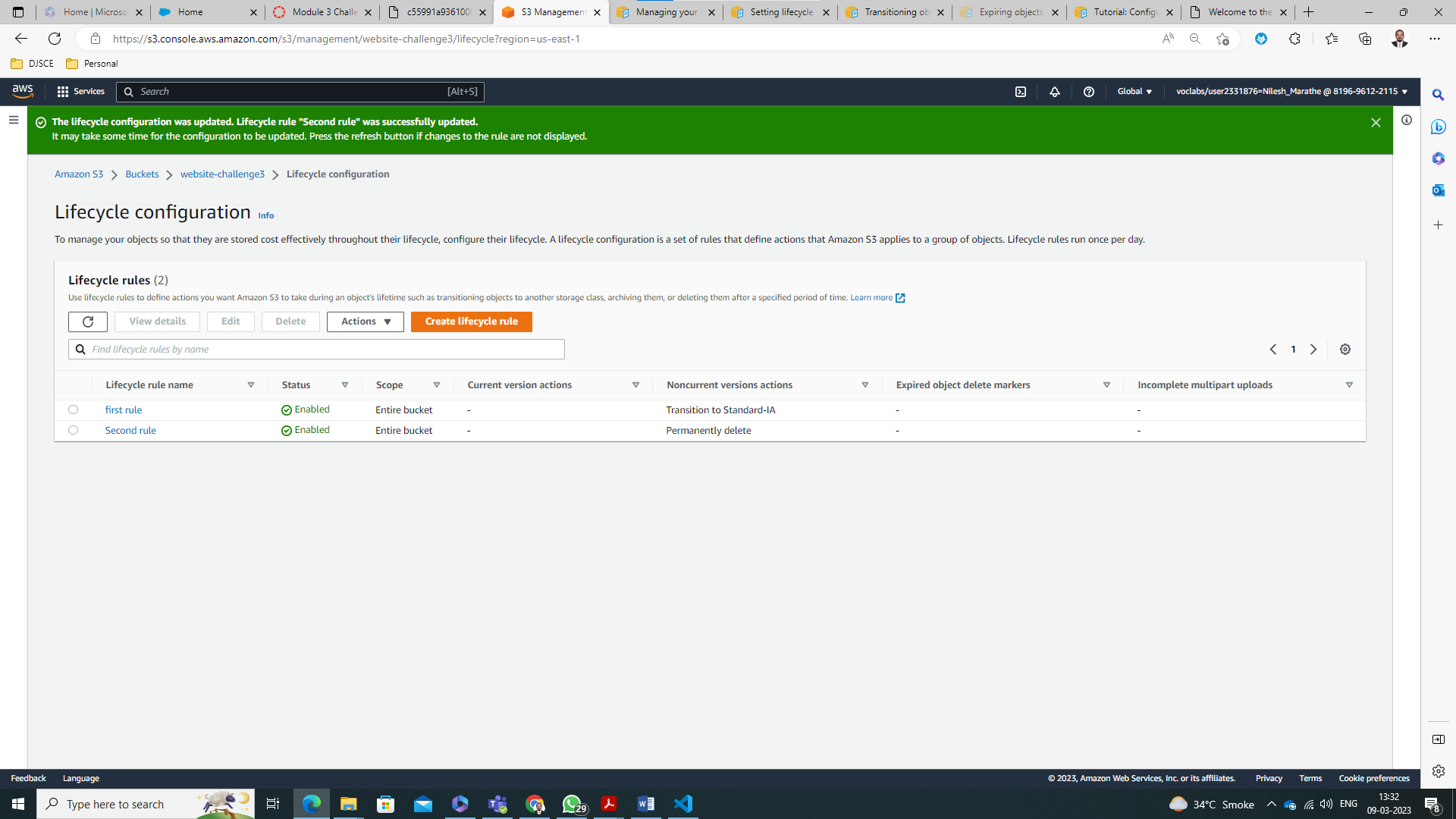
]

}

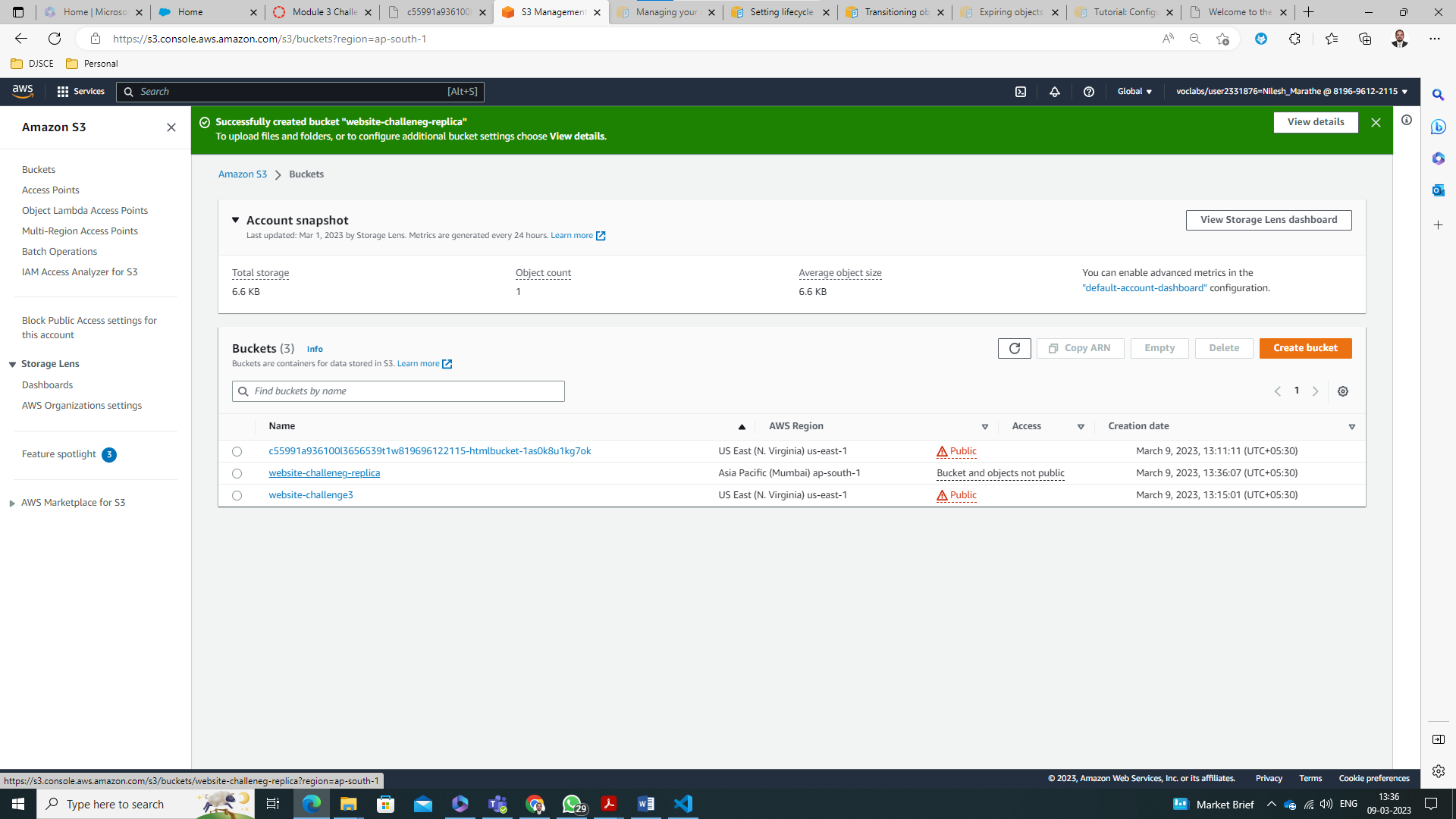








Created Second Bucket in Asia



Replication done

