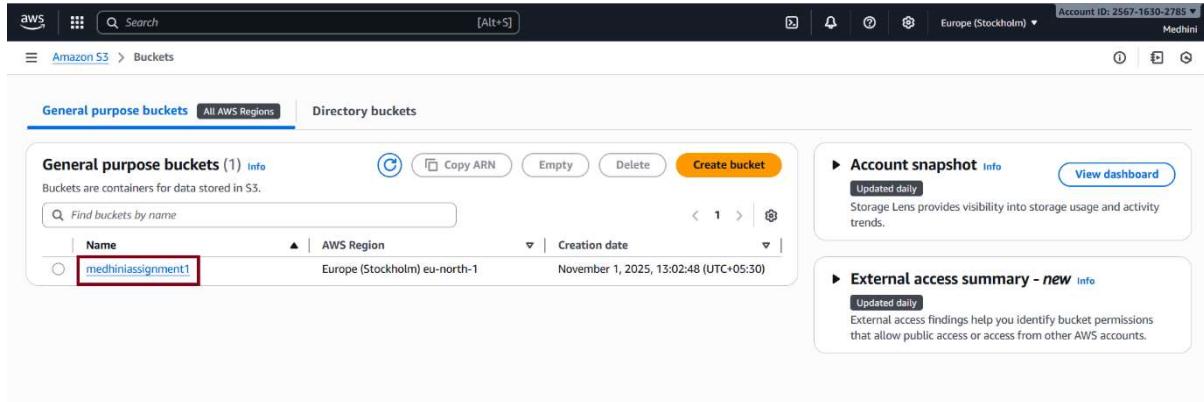


## Assignment S3 – Static Website and Lifecycle

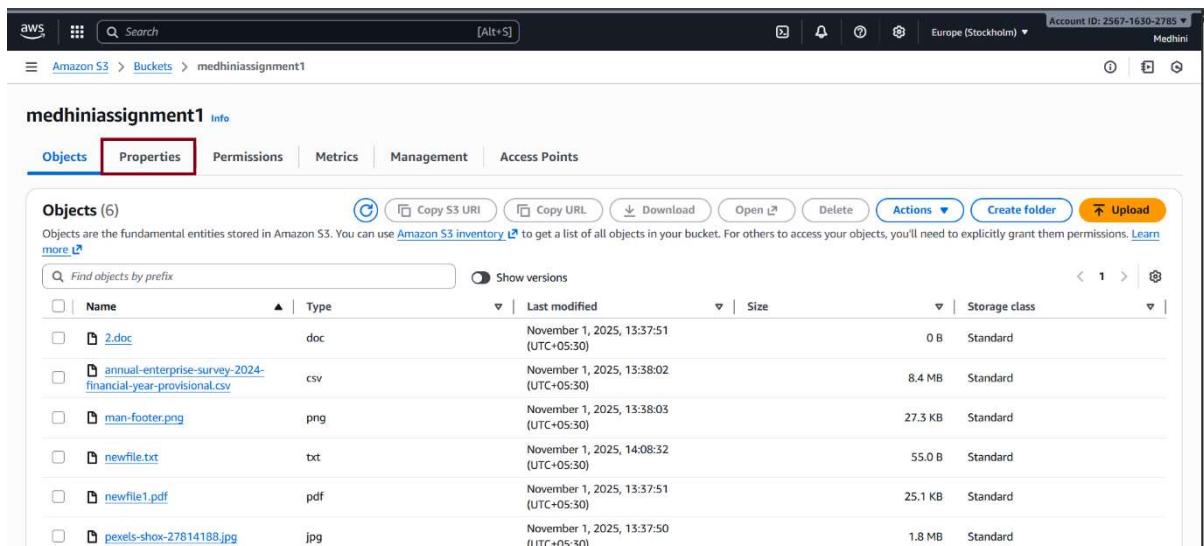
**Use the created bucket in the previous task to host static websites, upload an index.html and error.html page**

Step 1: Login to AWS account and click on created bucket



The screenshot shows the AWS S3 console with the 'General purpose buckets' tab selected. It lists one bucket, 'medhiniassignment1', which was created on November 1, 2025, at 13:02:48 UTC+05:30. The bucket is located in the 'Europe (Stockholm) eu-north-1' region. To the right of the bucket list, there are two informational boxes: 'Account snapshot' and 'External access summary - new'. The 'Account snapshot' box provides visibility into storage usage and activity trends, while the 'External access summary' box helps identify bucket permissions allowing public access or access from other AWS accounts.

Step 2: Click on “Properties”



The screenshot shows the AWS S3 console with the 'medhiniassignment1' bucket selected. The 'Properties' tab is active, indicated by a red box. Below it, the 'Objects' tab is also visible. The 'Objects' section displays six files: '2.doc', 'annual-enterprise-survey-2024-financial-year-provisional.csv', 'man-footer.png', 'newfile.txt', 'newfile1.pdf', and 'pixels-shox-27814188.jpg'. Each file's details like name, type, last modified, size, and storage class are listed. The 'Actions' dropdown menu is open, showing options like 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Create folder', and 'Upload'.

Step 3: You will get to see below tab and scroll down until you get “Static Web Hosting”

## Assignment S3 – Static Website and Lifecycle

Bucket overview

AWS Region: Europe (Stockholm) eu-north-1

Amazon Resource Name (ARN): arn:aws:s3:::medhiniasignment1

Creation date: November 1, 2025, 13:02:48 (UTC+05:30)

**Bucket Versioning**

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning: Enabled

**Multi-factor authentication (MFA) delete**

An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. [Learn more](#)

Multi-factor authentication (MFA) delete: Disabled

**Tags (0)**

You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

Currently this is disabled, so please click on “Edit” as shown in below picture

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

We recommend using AWS Amplify Hosting for static website hosting Deploy a fast, secure, and reliable website quickly with AWS Amplify Hosting. Learn more about [Amplify Hosting](#) or [View your existing Amplify apps](#)

S3 static website hosting

Disabled

Step 4: Click on “Enable” and “Host a Static website” give a file name as “index.html” and “error.html”

Edit static website hosting

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting

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.

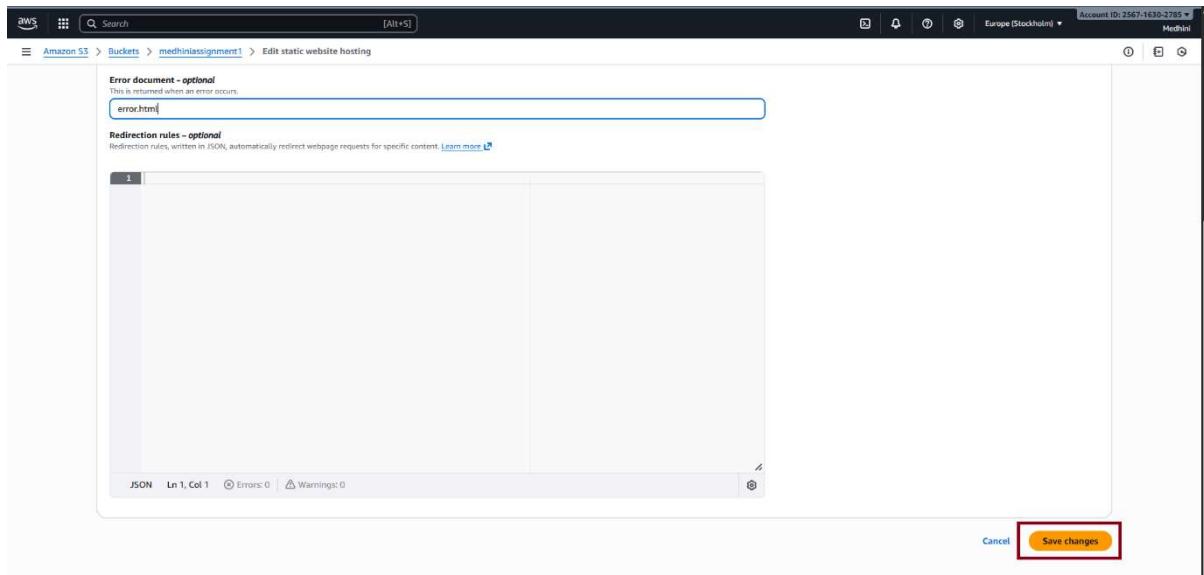
error.html

Redirection rules - optional

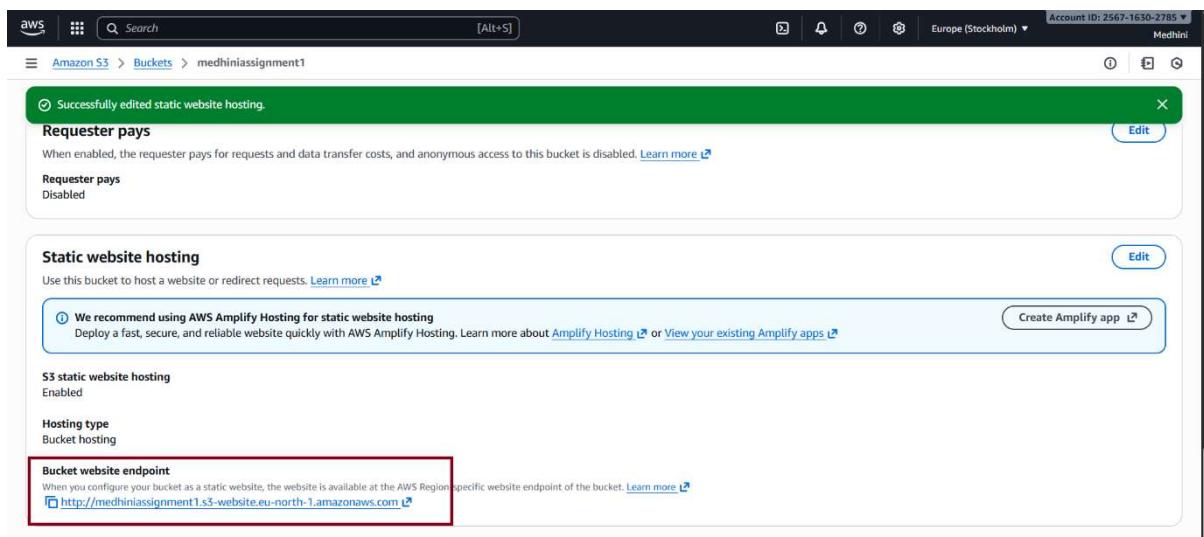
Redirection rules, written in JSON, automatically redirect webpage requests for specific content. [Learn more](#)

Step 5: Click on “Save changes”

## Assignment S3 – Static Website and Lifecycle



Step 6: Once you click on save changes scroll down at the bottom of page you will see the “**Bucket website endpoint**” to view the website



Click on it, the below page will be displayed



Step 7: Upload the index.html and error.html files on your created bucket

## Assignment S3 – Static Website and Lifecycle

The screenshot shows the AWS S3 'Upload' interface. In the 'Files and folders' section, two files are listed: 'error.html' and 'index.html'. Both files are highlighted with a red box. In the 'Destination' section, the bucket 'medhiniassignment1' is selected. At the bottom right, the 'Upload' button is highlighted with a red box.

You can see the files are uploaded on bucket

The screenshot shows the AWS S3 'Objects' page for the bucket 'medhiniassignment1'. The 'Objects (8)' section lists the following files:

Name	Type	Last modified	Size	Storage class
2.doc	doc	November 1, 2025, 13:37:51 (UTC+05:30)	0 B	Standard
annual-enterprise-survey-2024-financial-year-provisional.csv	csv	November 1, 2025, 13:38:02 (UTC+05:30)	8.4 MB	Standard
error.html	html	November 1, 2025, 14:52:26 (UTC+05:30)	416.0 B	Standard
index.html	html	November 1, 2025, 14:52:26 (UTC+05:30)	434.0 B	Standard
man-footer.png	png	November 1, 2025, 13:38:03 (UTC+05:30)	27.3 KB	Standard
newfile.txt	txt	November 1, 2025, 14:08:32 (UTC+05:30)	55.0 B	Standard
newfile1.pdf	pdf	November 1, 2025, 13:37:51 (UTC+05:30)	25.1 KB	Standard
pixels-shox-27814188.jpg	jpg	November 1, 2025, 13:37:50 (UTC+05:30)	1.8 MB	Standard

You still see the below page if you click on End point link from “Step 6”

The screenshot shows a 403 Forbidden error page. The error message is: "An Error Occurred While Attempting to Retrieve a Custom Error Document". Below it, there is a note: "Code: AccessDenied Message: Access Denied RequestId: 4DYZXAKFCHHSS1P1 HostId: mldIWCrzJfFHtQmMsGWjKaV80SLujtsR1rZVpHpDx+zLSDKguU2K+fY4rYLEDjxdAQyZSuM".

Note:

This is because to access publicly you should enable two things

1. Uncheck or remove “**Block all public access**”
2. Give a “**Bucket Policy**”

Step 8: Click on “**Permission**” tab and “**Block all public access**” is off here.

## Assignment S3 – Static Website and Lifecycle

The screenshot shows the AWS S3 console with the bucket 'medhiniassignment1'. The 'Permissions' tab is active. In the 'Block public access (bucket settings)' section, 'Block all public access' is set to 'Off'. Below that, under 'Bucket policy', it says 'Bucket policy is empty'. There are 'Edit' and 'Delete' buttons for the bucket policy.

Bucket policy is empty

The screenshot shows the AWS S3 console with the bucket 'medhiniassignment1'. The 'Permissions' tab is active. In the 'Block public access (bucket settings)' section, 'Block all public access' is set to 'Off'. Below that, under 'Bucket policy', it says 'No policy to display.' and has a 'Copy' button.

Step 9: Write bucket policy, click on “Edit”, click on “Save changes” as shown in below picture

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::medhiniassignment1/*"
        }
    ]
}
```

## Assignment S3 – Static Website and Lifecycle

The screenshot shows the AWS S3 Bucket Policy editor. On the left, a large text area displays a JSON policy document:

```
1 * Version: "2012-10-17",
2 * Statement: [
3 *     {
4 *         "Sid": "PublicReadGetObject",
5 *         "Effect": "Allow",
6 *         "Principal": "*",
7 *         "Action": "s3:GetObject",
8 *         "Resource": "arn:aws:s3:::medhinassignment1/*"
9 *     }
10 * ]
```

On the right, a modal window titled "Edit statement" is open, with a red box highlighting the "Save changes" button at the bottom.

Policy is created

The screenshot shows the AWS S3 Bucket Permissions overview page. A green success message at the top reads "Successfully edited bucket policy." Below it, the "Bucket policy" section displays the same JSON policy document as the previous screenshot:

```
1 * Version: "2012-10-17",
2 * Statement: [
3 *     {
4 *         "Sid": "PublicReadGetObject",
5 *         "Effect": "Allow",
6 *         "Principal": "*",
7 *         "Action": "s3:GetObject",
8 *         "Resource": "arn:aws:s3:::medhinassignment1/*"
9 *     }
10 * ]
```

Step 10: After updating policy, check on the url of “Bucket end point” now your website is visible as shown in below picture



To check the error.html, edit your url with some other name as shown in below picture you will see error.html file

## Assignment S3 – Static Website and Lifecycle



### Add a lifecycle rule for a bucket

- Transition from Standard-to-Standard IA in 60 days
- Expiration in 200 days

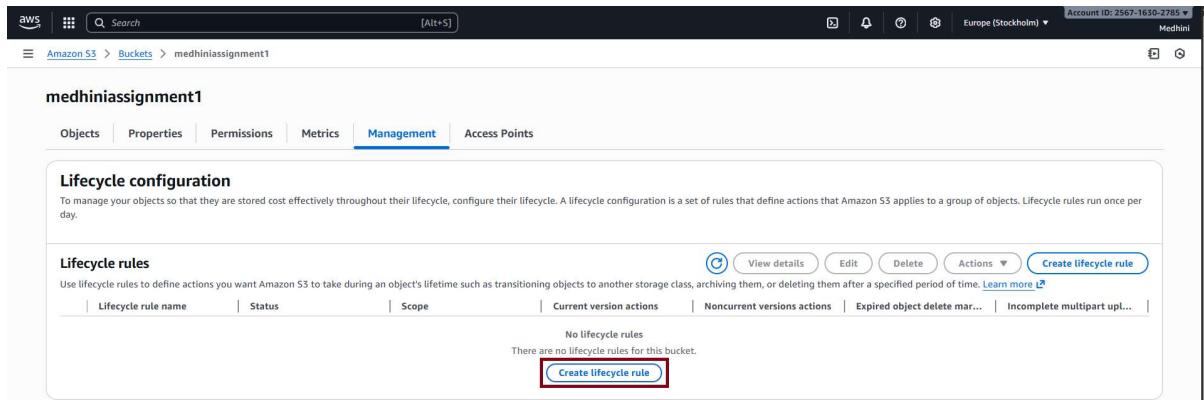
Step 1: Click on “Management” tab

Lifecycle is a automation policy that helps you to move, delete, archive objects based on how old they are.

A screenshot of the AWS S3 console showing the "Management" tab for a bucket named "medhinassignment1". The "Properties" tab is selected. The "Bucket overview" section shows the AWS Region as "Europe (Stockholm) eu-north-1". The "Amazon Resource Name (ARN)" is listed as "arn:aws:s3:::medhinassignment1". The "Creation date" is "November 1, 2025, 15:02:48 (UTC+05:30)". The "Bucket Versioning" section indicates it is "Enabled". The "Multi-factor authentication (MFA) delete" section shows it is "Disabled". The "Tags (0)" section indicates there are no tags present.

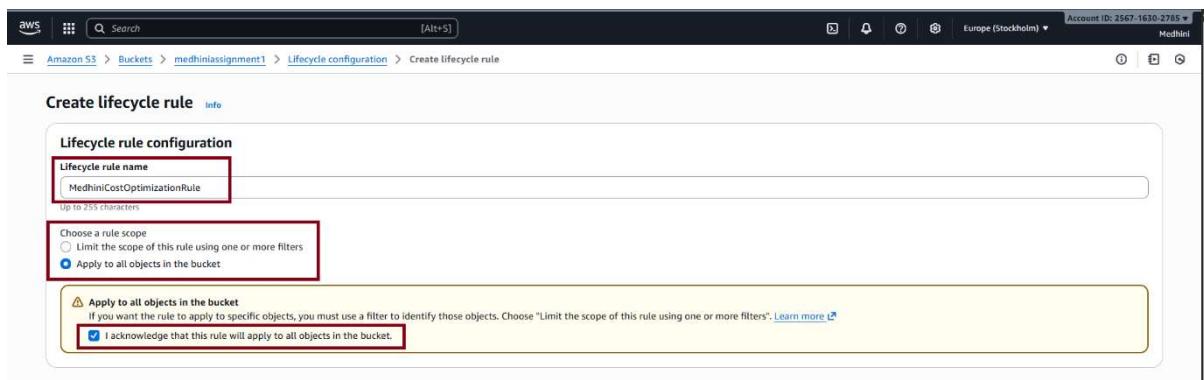
Step 2: Click on “Create lifecycle rule” as shown in below picture

## Assignment S3 – Static Website and Lifecycle



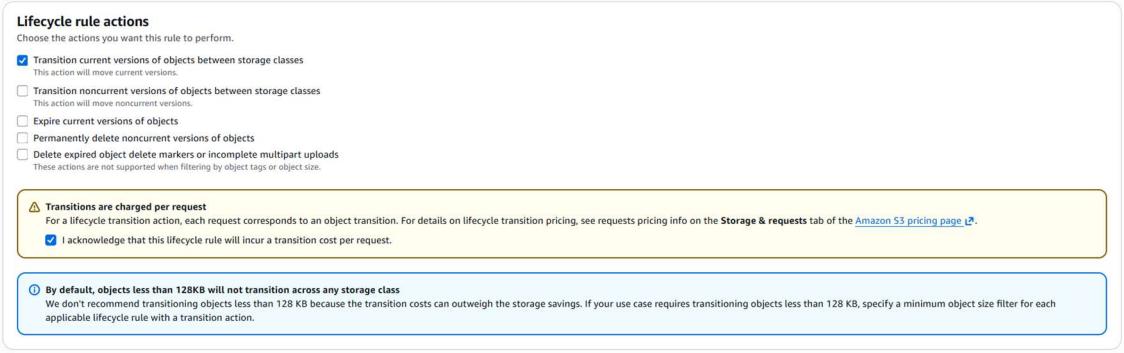
The screenshot shows the AWS S3 console with the 'medhiniasignment1' bucket selected. In the 'Lifecycle configuration' section, there is no existing lifecycle rule. A prominent 'Create lifecycle rule' button is highlighted with a red box.

Step 3: Give a name and choose “**Apply to all objects in the bucket**” and check on acknowledge as shown in below picture



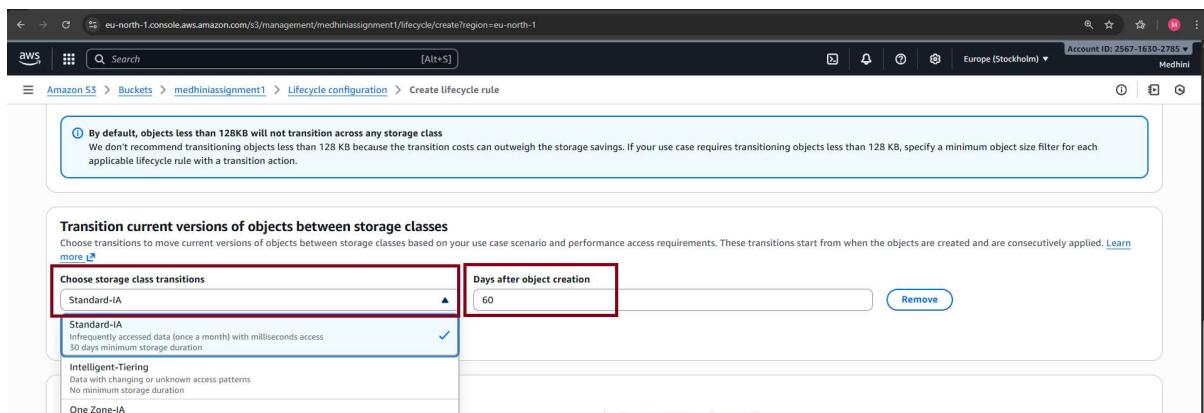
The screenshot shows the 'Create lifecycle rule' dialog. The 'Lifecycle rule name' field is set to 'MedhiniCostOptimizationRule'. Under 'Choose a rule scope', the option 'Apply to all objects in the bucket' is selected. At the bottom, the 'I acknowledge that this rule will apply to all objects in the bucket.' checkbox is checked and highlighted with a red box.

Step 4: As per the question, choose first option as shown in below picture



The screenshot shows the 'Lifecycle rule actions' section. It lists several actions: 'Transition current versions of objects between storage classes' (selected), 'Transition noncurrent versions of objects between storage classes', 'Expire current versions of objects', 'Permanently delete noncurrent versions of objects', and 'Delete expired object delete markers or incomplete multipart uploads'. Below these, a note states: 'By default, objects less than 128KB will not transition across any storage class'. The first action is highlighted with a red box.

Step 5: Choose “**Standard-IA**” and add “**60**” number of days



The screenshot shows the 'Transition current versions of objects between storage classes' dialog. It lists storage classes: 'Standard-IA' (selected), 'Standard-Tiering' (with a note about changing access patterns), and 'One Zone-IA'. The 'Days after object creation' input field is set to '60'. The 'Standard-IA' option is highlighted with a red box.

## Assignment S3 – Static Website and Lifecycle

Step 6: Check on “**Expire Current version of objects**” for Expiration in 200 days as per the question

The screenshot shows the 'Lifecycle rule actions' section of the AWS Lambda function configuration. It lists several actions:

- Transition current versions of objects between storage classes
- Transition noncurrent versions of objects between storage classes
- Expire current versions of objects
- Permanently delete noncurrent versions of objects
- Delete expired object delete markers or incomplete multipart uploads

A note states: "These actions are not supported when filtering by object tags or object size."

A warning box says: "Transitions are charged per request. For a lifecycle transition action, each request corresponds to an object transition. For details on lifecycle transition pricing, see requests pricing info on the Storage & requests tab of the Amazon S3 pricing page." A checkbox "I acknowledge that this lifecycle rule will incur a transition cost per request" is checked.

A note at the bottom says: "By default, objects less than 128KB will not transition across any storage class. We don't recommend transitioning objects less than 128 KB because the transition costs can outweigh the storage savings. If your use case requires transitioning objects less than 128 KB, specify a minimum object size filter for each applicable lifecycle rule with a transition action."

Please scroll down and add “200” under the below tab

The screenshot shows the 'Transition current versions of objects between storage classes' section with a note about transitioning objects less than 128KB. The 'Days after object creation' dropdown is set to 60.

The 'Expire current versions of objects' section has a note about version-enabled buckets. The 'Days after object creation' input field is highlighted with a red box and contains the value 200.

Verify all options and click on “Create rule”

The screenshot shows the 'Review transition and expiration actions' section. It displays two columns: 'Current version actions' and 'Noncurrent versions actions'. The 'Current version actions' column shows a timeline from Day 0 to Day 200. The 'Noncurrent versions actions' column shows a timeline from Day 0. A red box highlights the 'Create rule' button at the bottom right.

Step 7: The lifecycle rule has been created ssss

## Assignment S3 – Static Website and Lifecycle

The screenshot shows the AWS S3 console under the 'Lifecycle configuration' section for the bucket 'medhiniautomation'. A green success message at the top states: 'The rule "MediniCostOptimizationRule" has been successfully added and the lifecycle configuration has been updated. It may take some time for the configuration to be updated. Refresh the lifecycle rules list if changes to the configuration aren't displayed.' Below this, the 'Lifecycle rules (1)' section is shown with one rule named 'MediniCostOptimizationRule'. The rule is enabled and applies to the entire bucket, transitioning objects to Standard-IA after 30 days. The table headers for the rules list are: Lifecycle rule name, Status, Scope, Current version actions, Noncurrent versions ac..., Expired object delete ..., and Incomplete multipart u... .

Lifecycle rule name	Status	Scope	Current version actions	Noncurrent versions ac...	Expired object delete ...	Incomplete multipart u...
MediniCostOptimizationRule	Enabled	Entire bucket	Transition to Standard-IA, then	-	-	-