



Andhra Pradesh State Skill Development Corporation



AWS CLOUD COMPUTING

AMAZON SIMPLE STORAGE SERVICE (AMAZON S3)



Configuration of Amazon Simple Storage Service (Amazon S3)





Amazon Simple Storage Service (Amazon S3)

Amazon Simple Storage Service is storage for the Internet. It is designed to make web-scale computing easier for developers. Amazon S3 has a simple web services interface that you can use to store and retrieve any amount of data, at any time, from anywhere on the web. It gives any developer access to the same highly scalable, reliable, fast, inexpensive data storage infrastructure that Amazon uses to run its own global network of websites. The service aims to maximize benefits of scale and to pass those benefits on to developers.

Amazon S3 Bucket

A bucket is a logical unit of storage in Amazon Web Services (AWS) object storage service, Simple Storage Solution S3. Buckets are used to store objects, which consist of data and metadata that describes the data. When you subscribe to the data feed, you must specify an Amazon S3 bucket to store the data feed files.

Before you choose an Amazon S3 bucket for the data feed, consider the following:

- You must use a bucket from the US East.
- You must have FULL_CONTROL permission to the bucket.

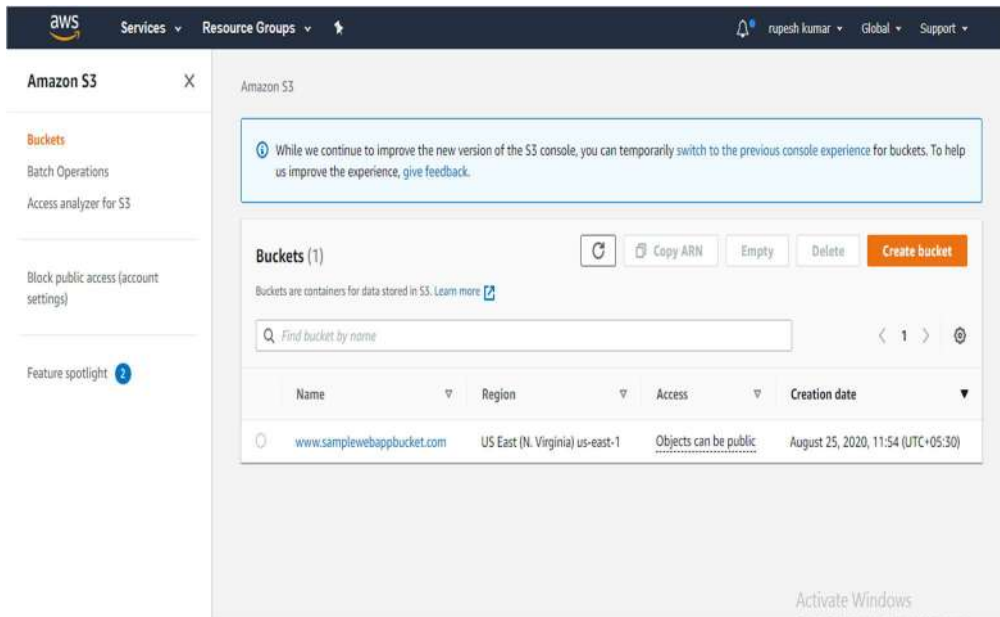
If you're the bucket owner, you have this permission by default. Otherwise, the bucket owner must grant your AWS account this permission. When you create your data feed subscription, Amazon S3 updates the ACL of the specified bucket to allow the AWS data feed account read and write permissions.

Removing the permissions for the data feed account does not disable the data feed. If you remove those permissions but don't disable the data feed, we restore those permissions the next time that the data feed account needs to write to the bucket. Each data feed file has its own ACL. The bucket owner has FULL_CONTROL permission to the data files.

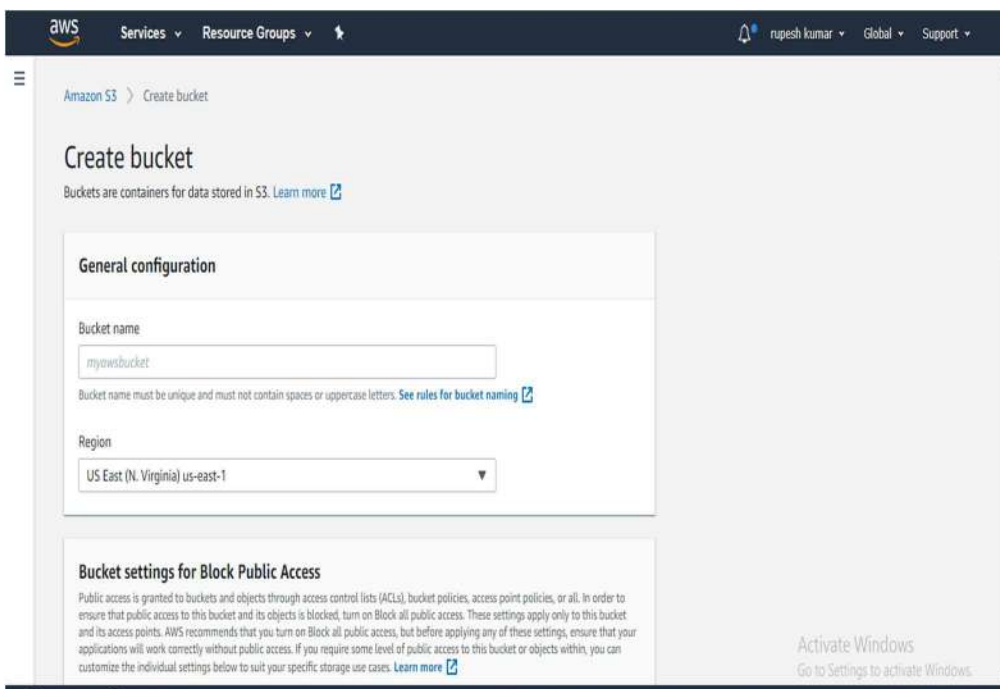
The data feed account has read and write permissions. If you delete your data feed subscription, Amazon EC2 does not remove the read and write permissions for the data feed account on either the bucket or the data files. You must remove these permissions yourself.

Step 1: To create an S3 bucket

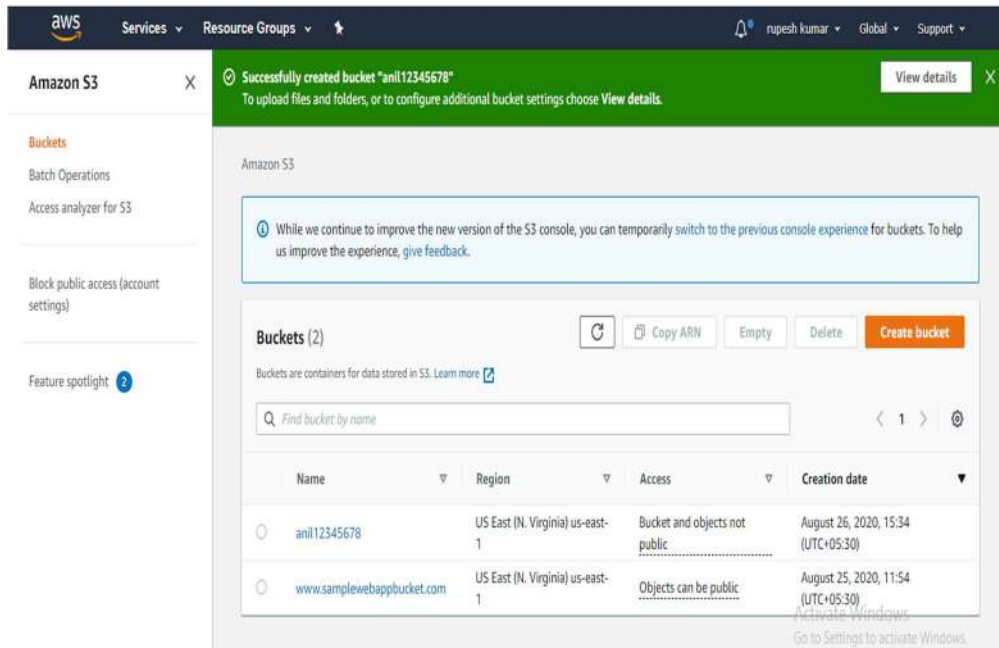
1. Sign in to the AWS Management Console and open the Amazon S3 console
2. Choose Create bucket.



- The bucket name you choose must be globally unique across all existing bucket names in Amazon S3 (that is, across all AWS customers).



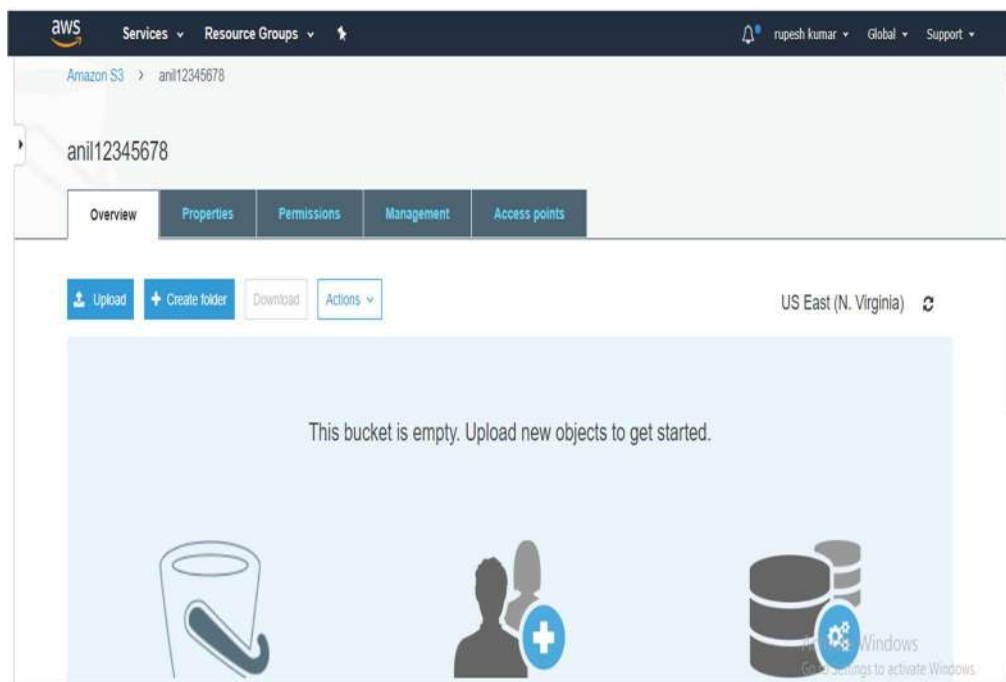
- In Region, choose Region and Click on Create. When Amazon S3 successfully creates your bucket, the console displays your empty bucket in the Buckets pane.

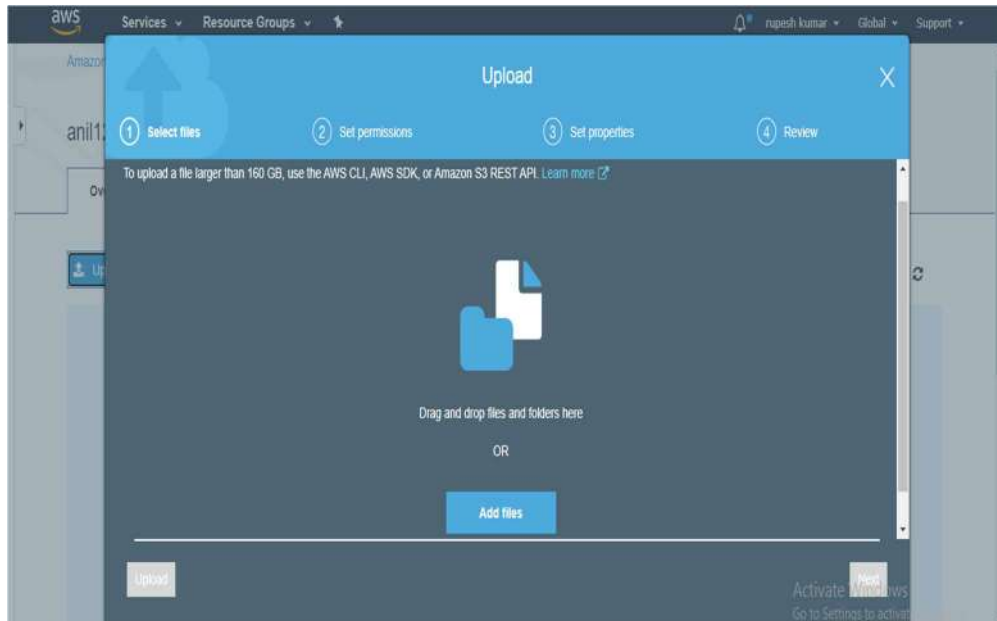


Step 2: Upload a File to Your Amazon S3 Bucket

Now that you've created a bucket, you're ready to add an object to it. An object can be any kind of file: a document, a photo, a video, a music file, or other file type.

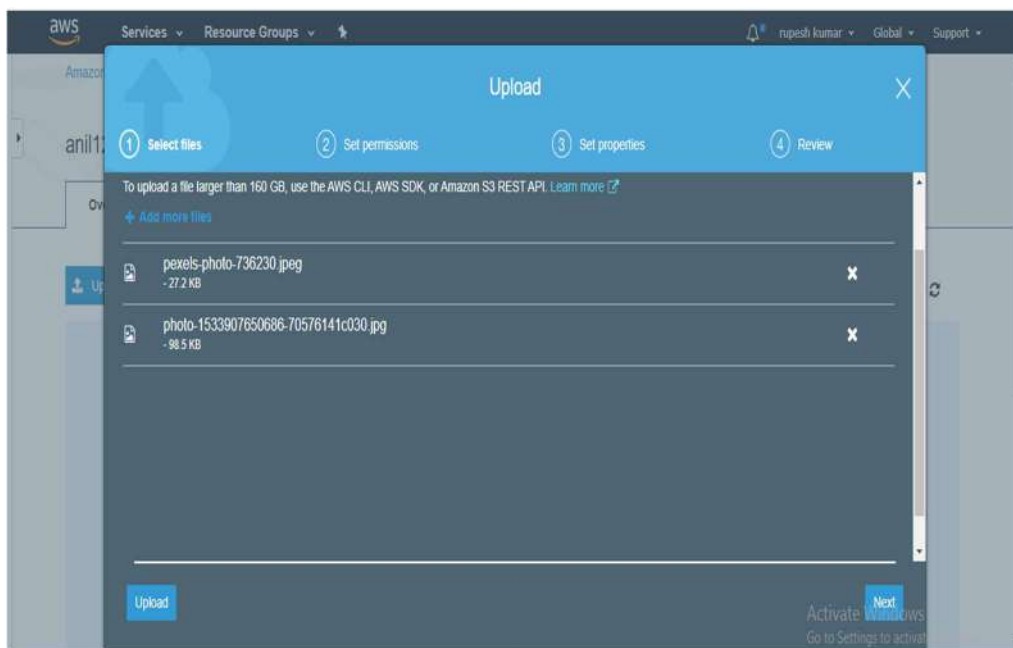
1. In the Amazon S3 console, choose the bucket where you want to upload an object, choose **Upload**, and then choose **Add Files**.



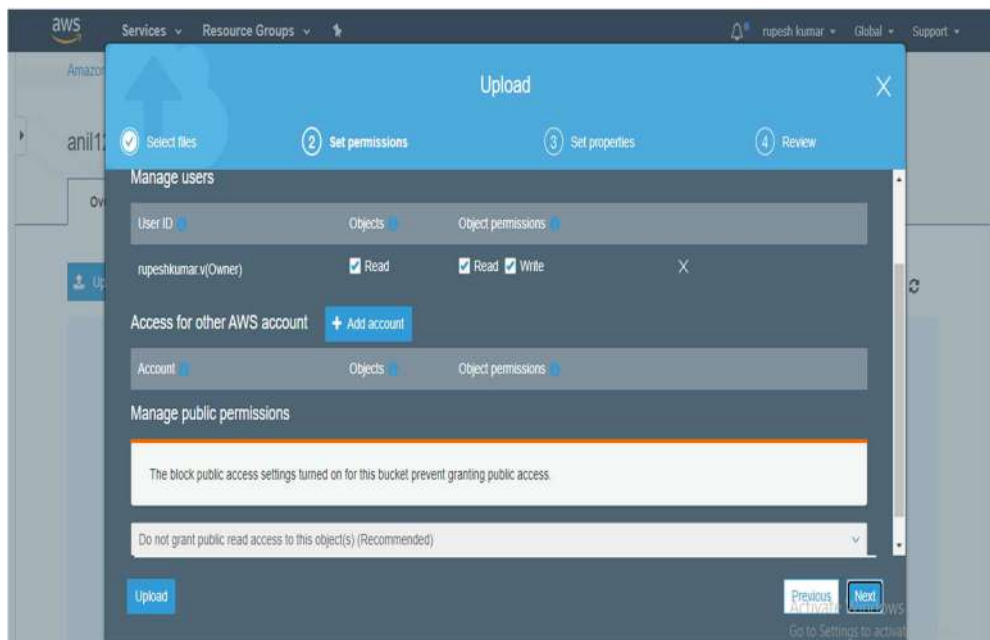


In the Upload dialog box, do one of the following:

- Drag and drop more files and folders to the console window that displays the **Upload** dialog box. To add more files, you can also choose **Add more files**. This option works only for files, not folders.
- To immediately upload the listed files and folders without granting or removing permissions for specific users or setting public permissions for all of the files that you're uploading, choose **Upload**.
- To set permissions or properties for the files that you are uploading, choose **Next**.



- On the **Set Permissions** page, under **Manage users** you can change the permissions for the AWS account owner. The owner refers to the AWS account root user, and not an AWS Identity and Access Management (IAM) user.
- Under **Manage public permissions** you can grant read access to your objects to the general public (everyone in the world), for all of the files that you're uploading. Granting public read access is applicable to a small subset of use cases such as when buckets are used for websites. We recommend that you do not change the default setting of **Do not grant public read access to this object(s)**. You can always make changes to object permissions after you upload the object.
- When you're done configuring permissions, choose **Next**.



- On the **Set Properties** page, choose the storage class and encryption method to use for the files that you are uploading. You can also add or modify metadata.
- Choose a storage class for the files you're uploading.

Upload

1 Select files 2 Set permissions 3 Set properties 4 Review

Storage class	Designed for	Availability Zones	Min storage duration	Min billable object size	Monitoring and automation fees	Retrieval fees
<input type="radio"/> Standard	Frequently accessed data	≥ 3	-	-	-	-
<input type="radio"/> Intelligent-Tiering	Long-lived data with changing or unknown access patterns	≥ 3	30 days	-	Per-object fees apply	-
<input type="radio"/> Standard-IA	Long-lived, infrequently accessed data	≥ 3	30 days	128KB	-	Per-GB fees apply
<input type="radio"/> One Zone-IA	Long-lived, infrequently accessed, non-critical data	≥ 1	30 days	128KB	-	Per-GB fees apply
<input type="radio"/> Glacier	Archive data with retrieval times ranging from minutes to hours	≥ 3	90 days	40KB	-	Per-GB fees apply
<input type="radio"/> Glacier Deep Archive	Archive data that rarely, if ever, needs to be accessed with retrieval times in	≥ 3	180 days	40KB	-	Per-GB fees apply

Upload Previous Next

Upload

1 Select files 2 Set permissions 3 Set properties 4 Review

<input type="radio"/> One Zone-IA	Long-lived, infrequently accessed, non-critical data	≥ 1	30 days	128KB	-	Per-GB fees apply
<input type="radio"/> Glacier	Archive data with retrieval times ranging from minutes to hours	≥ 3	90 days	40KB	-	Per-GB fees apply
<input type="radio"/> Glacier Deep Archive	Archive data that rarely, if ever, needs to be accessed with retrieval times in hours	≥ 3	180 days	40KB	-	Per-GB fees apply
<input type="radio"/> Reduced Redundancy (Not recommended)	Frequently accessed, non-critical data	≥ 3	-	-	-	-

Encryption

Protect data at rest by using Amazon S3 master-key or by using AWS KMS master-key.

☒ None ☐ Amazon S3 master-key ☐ AWS KMS master-key

Upload Previous Next

Choose the type of encryption for the files that you're uploading. If you don't want to encrypt them, choose **None**.

- To encrypt the uploaded files using keys that are managed by Amazon S3, choose **Amazon S3 master-key**.
- To encrypt the uploaded files using the AWS Key Management Service (AWS KMS), choose **KMS AWS master-key**. Then choose a master key from the list of AWS KMS master keys.

Note: To encrypt objects in a bucket, you can use only keys that are available in the same AWS Region as the bucket.

You can give an external account the ability to use an object that is protected by an AWS KMS key. To do this, select **Custom KMS ARN** from the list and enter the Amazon Resource Name (ARN) for the external account. Administrators of an external account that have usage permissions to an object protected by your AWS KMS key can further restrict access by creating a resource-level IAM policy.

Metadata for Amazon S3 objects is represented by a name-value (key-value) pair. There are two kinds of metadata: system-defined metadata and user-defined metadata.

If you want to add Amazon S3 system-defined metadata to all of the objects you are uploading, for **Header**, select a header. You can select common HTTP headers, such as **Content-Type** and **Content-Disposition**. Type a value for the header, and then choose **Save**.

Any metadata starting with prefix **x-amz-meta-** is treated as user-defined metadata. User-defined metadata is stored with the object, and is returned when you download the object.

To add user-defined metadata to all of the objects that you are uploading, type **x-amz-meta-** plus a custom metadata name in the Header field. Type a value for the header, and then choose **Save**. Both the keys and their values must conform to US-ASCII standards. User-defined metadata can be as large as 2 KB.



Object tagging gives you a way to categorize storage. Each tag is a key-value pair. Key and tag values are case sensitive. You can have up to 10 tags per object. To add tags to all of the objects that you are uploading, type a tag name in the Key field. Type a value for the tag, and then choose **Save**. A tag key can be up to 128 Unicode characters in length and tag values can be up to 255 Unicode characters in length.





- Choose **Next**.
- On the Upload review page, verify that your settings are correct, and then choose Upload. To make changes, choose Previous. To see the progress of the upload, choose In progress at the bottom of the browser window.

anil12345678

Overview Properties Permissions Management Access points

Q Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder Download Actions

US East (N. Virginia)

Viewing 1 to 2

Name	Last modified	Size	Storage class
pexels-photo-736230.jpeg	Aug 26, 2020 6:17:46 PM GMT+0530	27.2 KB	Standard
photo-1533907650686-70576141c030.jpg	Aug 26, 2020 6:17:46 PM GMT+0530	98.5 KB	Standard

Viewing 1 to 2

Operations 0 In progress 1 Success 0 Error

Activate Windows
Go to Settings to activate Windows.

- To see a history of your uploads and other operations, choose **Success**.

Viewing 1 to 2

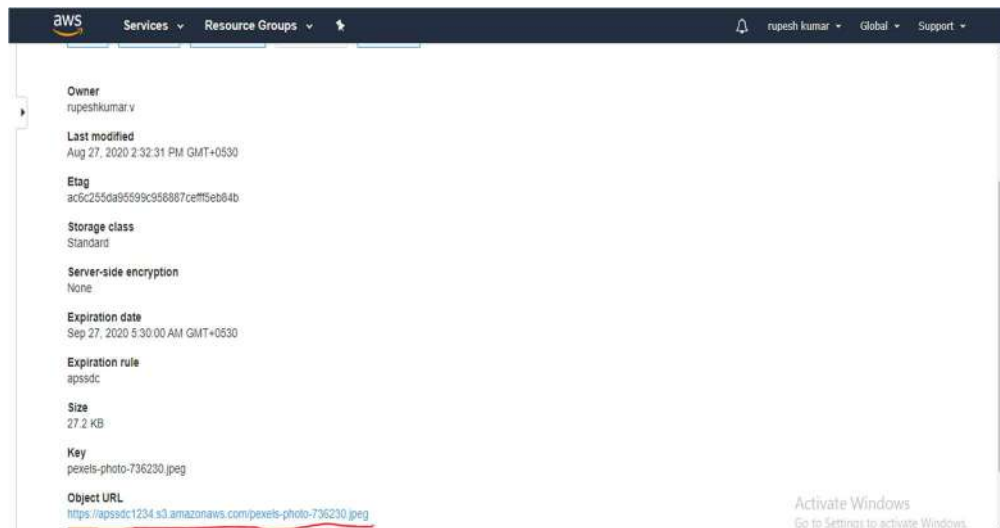
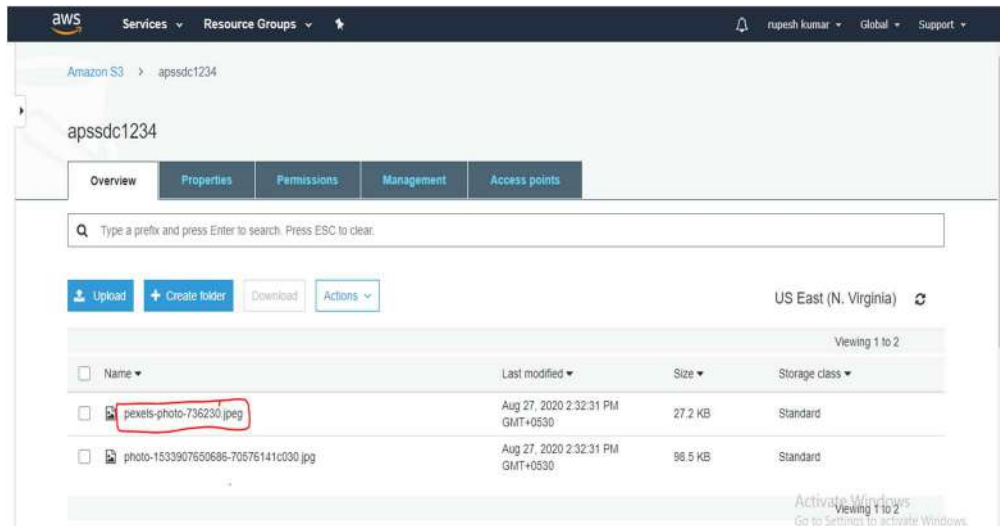
Upload View details

100% Successful

Operations 0 In progress 1 Success 0 Error

Activate Windows
Go to Settings to activate Windows.

- Now click on the **object name** and if you scroll down a little bit you'll find an **object URL** in the bottom of the page and access it.

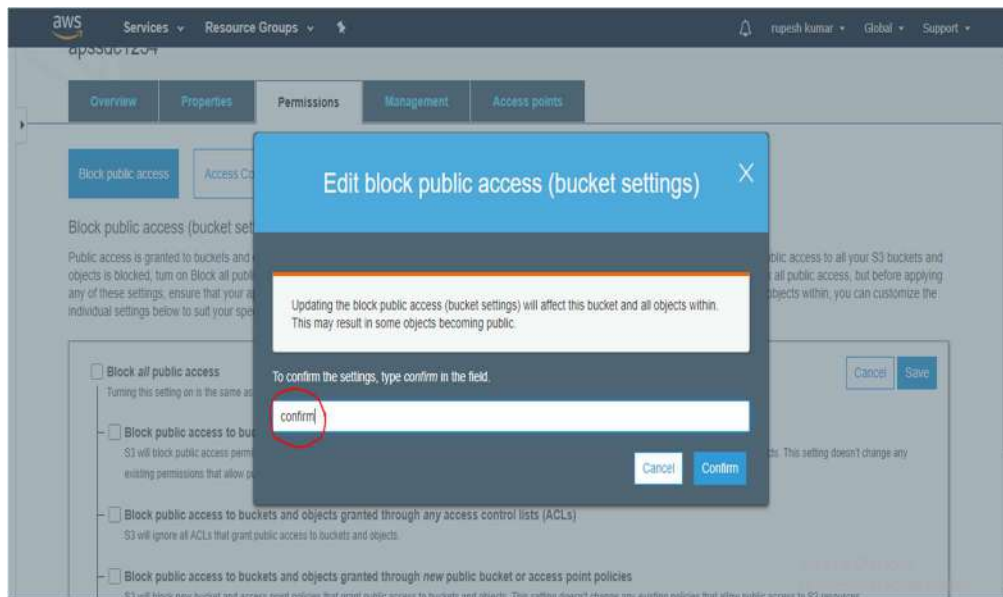


- You'll get an **xml error** when you access the object url, it means that your object doesn't have any permissions to the public. You need to change the permissions of the object as well as the bucket in order to make it public.

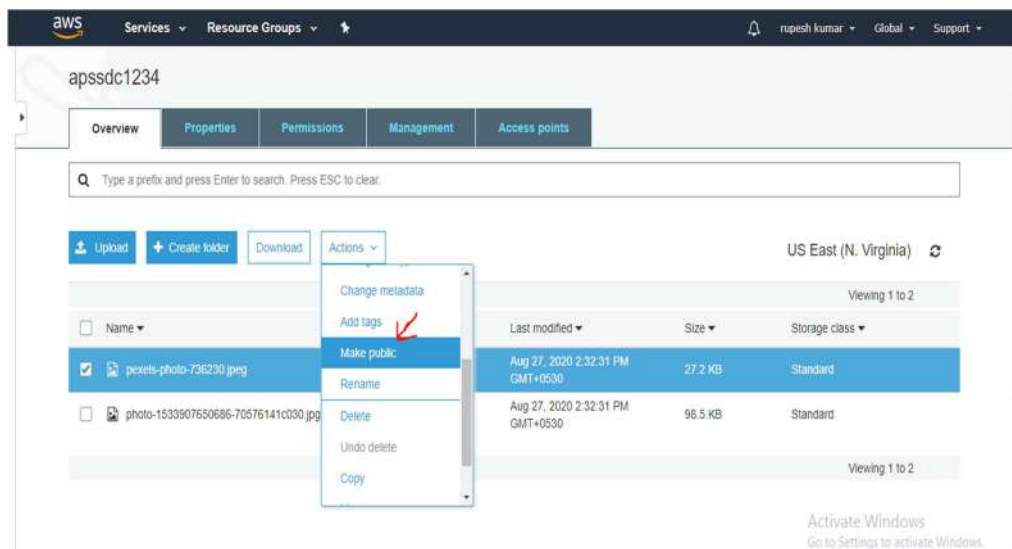
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>953840E91937CE33</RequestId>
  <HostId>hxrQcbPjRiL8GU/Muq7jt/sus2uP7CIPmNB/ueETIzhSsv1LDAPh+Y2Fgcj+V4yCXK7hI+gBuQ= </HostId>
</Error>
```

- You can see the bucket name at the top of the object overview page and click on it. Now click on the **permissions** tab and disable block all public access by clicking on the **edit button**.



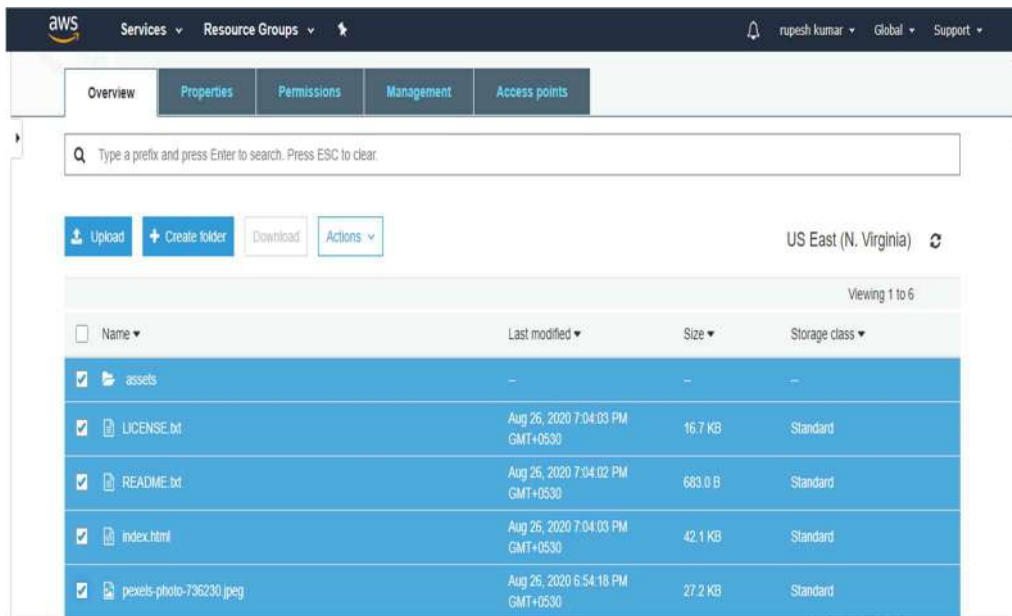
- After clicking on the save button type **confirm** in the box, now select the object by clicking on checkbox, click on **actions**, click on **make public** and check for the output by clicking on object URL. Now your object has public access and you can share the URL also.



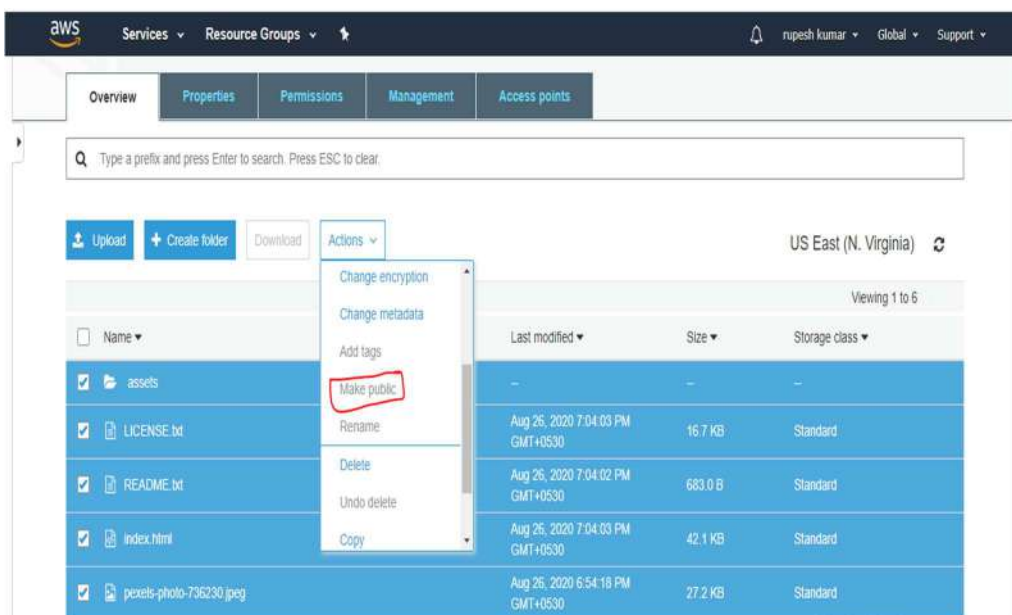
Hosting a Static Website on Amazon S3:

You can host a static website on Amazon Simple Storage Service (Amazon S3). On a static website, individual web pages include static content. They might also contain client-side scripts. By contrast, a dynamic website relies on server-side processing, including server-side scripts such as PHP, JSP, or ASP.NET. Amazon S3 does not support server-side scripting.

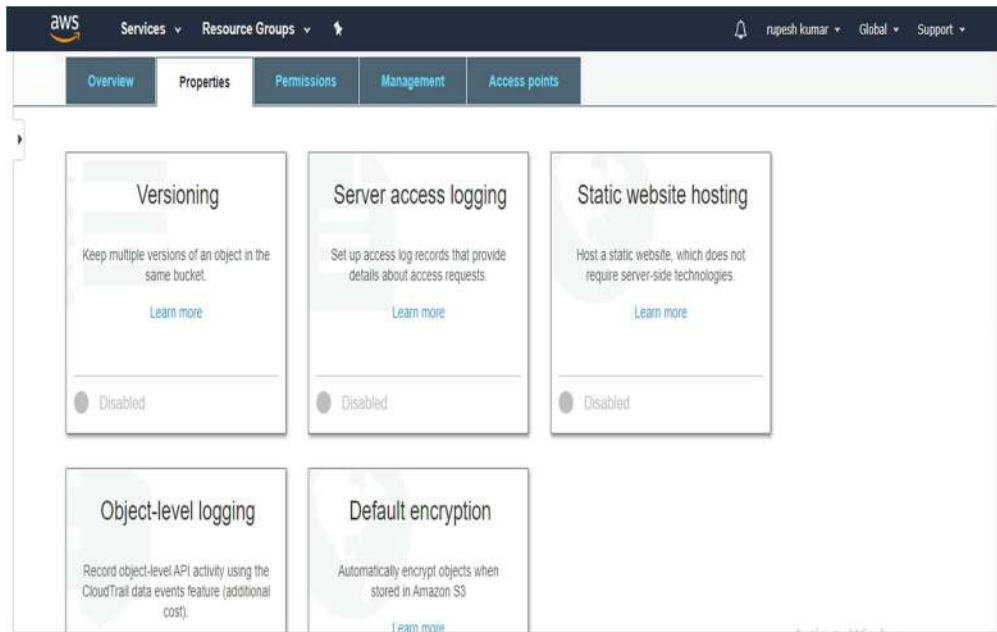
- To host a static website, you configure an Amazon S3 bucket for website hosting, and then upload your website content to the bucket. This bucket must have public read access. It is intentional that everyone in the world will have read access to this bucket. The website is then available at the AWS Region-specific website endpoint of the bucket.



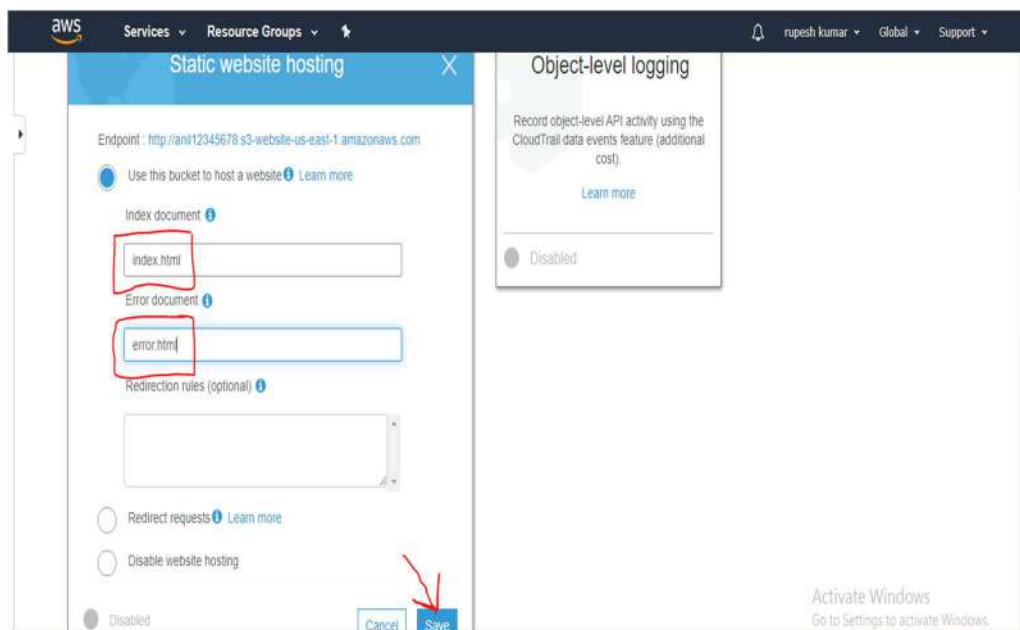
- After successful completion of code uploaded to S3, then select all object in bucket make it as public as shown below slide.



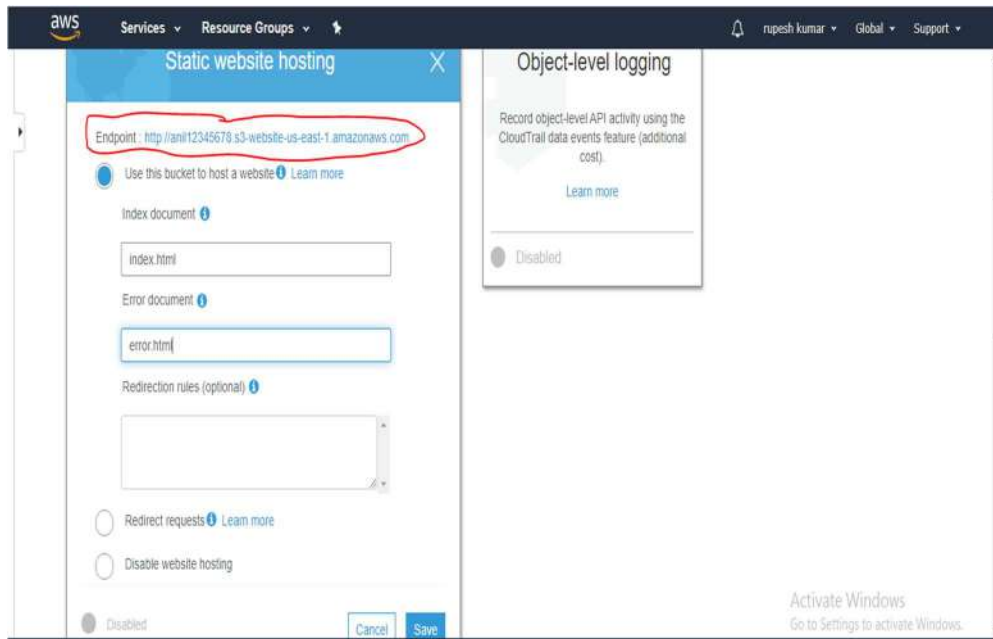
- Open the bucket Properties pane, choose Static Website Hosting, and do the following:



- Choose Use this bucket to host a website.
- In the Index Document box, type the name of your index document. The name is typically index.html.



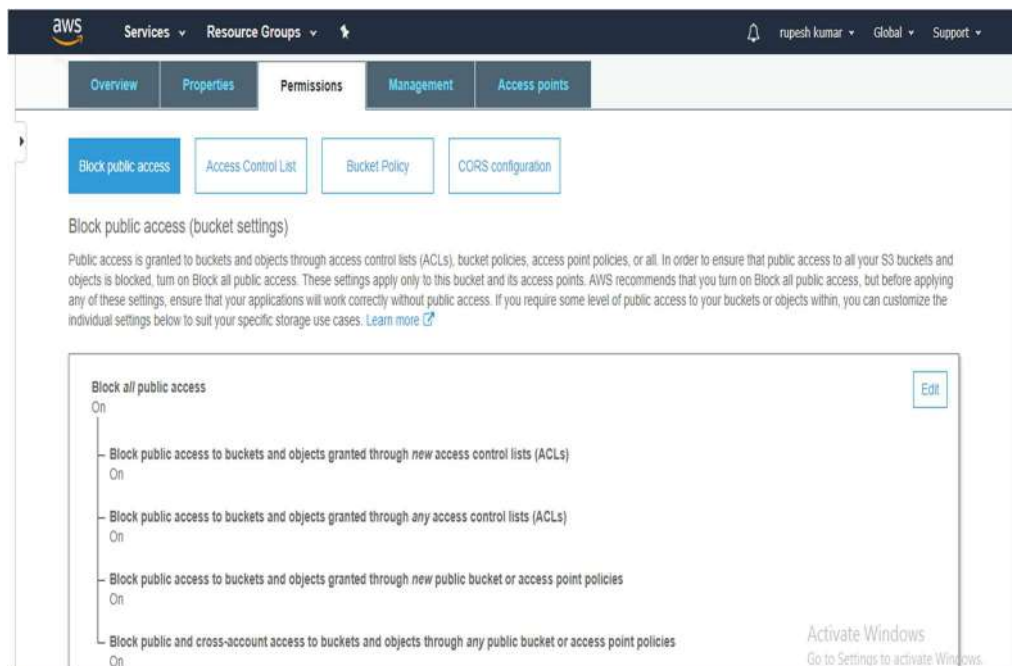
- Choose Save to save the website configuration.
- Write down the Endpoint.



The screenshot shows the AWS Static website hosting console. The 'Endpoint' field is highlighted with a red circle, showing the URL: `http://ani112345678.s3-website-us-east-1.amazonaws.com`. Below this, there are fields for 'Index document' (set to `index.html`) and 'Error document' (set to `error.html`). There are also options for 'Redirection rules (optional)', 'Redirect requests', and 'Disable website hosting'. The 'Object-level logging' section on the right is currently disabled.

Adding a Bucket Policy That Makes Your Bucket Content Publicly Available:

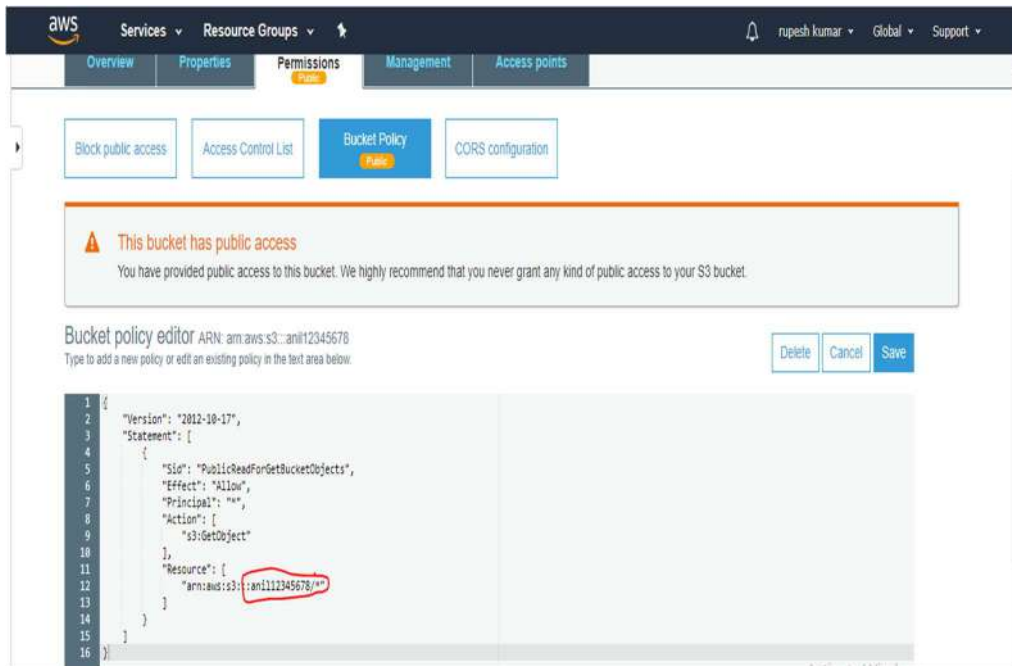
- In the **Properties** pane for the bucket, choose **Permissions**.



The screenshot shows the AWS S3 bucket 'Permissions' console. The 'Block public access' section is expanded, showing settings for blocking public access. The 'Block all public access' toggle is turned 'On'. Below it, there are four sub-toggles, all of which are also turned 'On':

- Block public access to buckets and objects granted through new access control lists (ACLs)
- Block public access to buckets and objects granted through any access control lists (ACLs)
- Block public access to buckets and objects granted through new public bucket or access point policies
- Block public and cross-account access to buckets and objects through any public bucket or access point policies

- Choose **Add Bucket Policy**.



- To host a website, your bucket must have public read access. It is intentional that everyone in the world will have read access to this bucket. Copy the following bucket policy, and then paste it in the Bucket Policy Editor.

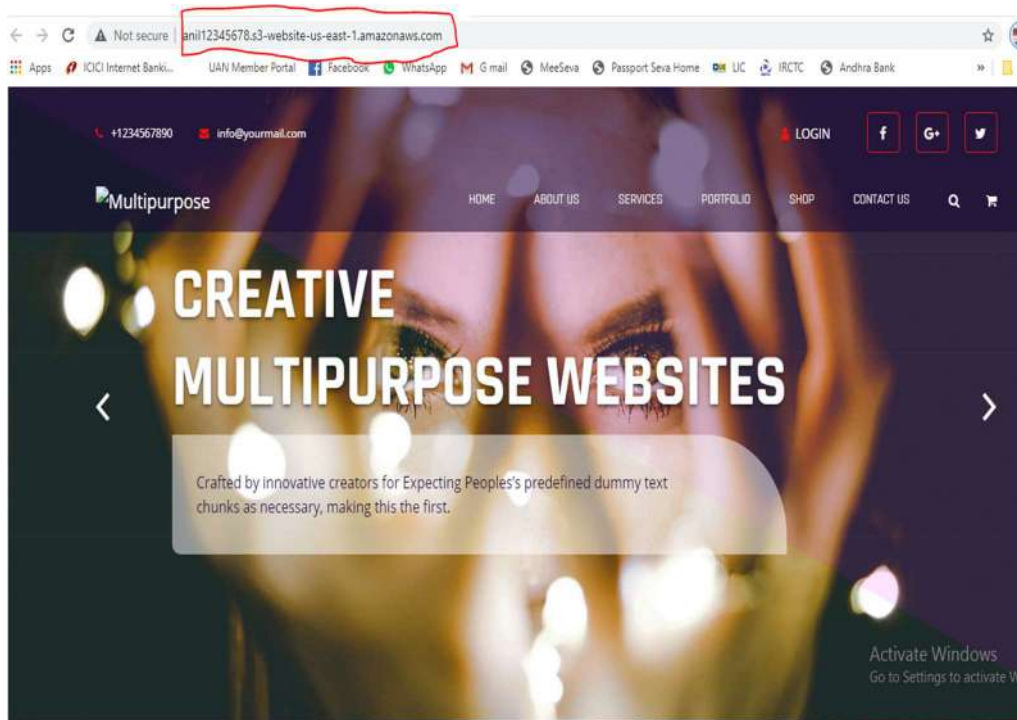
```

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

- In the policy, replace **example-bucket** with the name of your bucket.
- Choose **Save**.

Testing Your Website:

- Type the following URL in the browser, replacing **example-bucket** with the name of your bucket and **website-region** with the name of the AWS Region where you deployed your bucket.
- This is the Amazon S3-provided website endpoint for your bucket. You use this endpoint to test whether your website is running or not by pasting the Endpoint into your browser.



- If your browser displays your index.html page, the website was successfully deployed.

Note: HTTPS access to the website is not supported.

You now have a website hosted on Amazon S3. This website is available at the Amazon S3 website endpoint. However, you might have a domain, such as example.com, that you want to use to serve the content from the website you created. You might also want to use Amazon S3 root domain support to serve requests for both <http://www.example.com> and <http://example.com>.