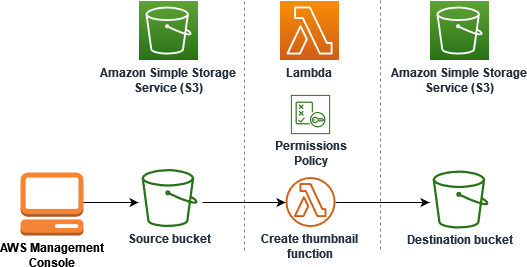
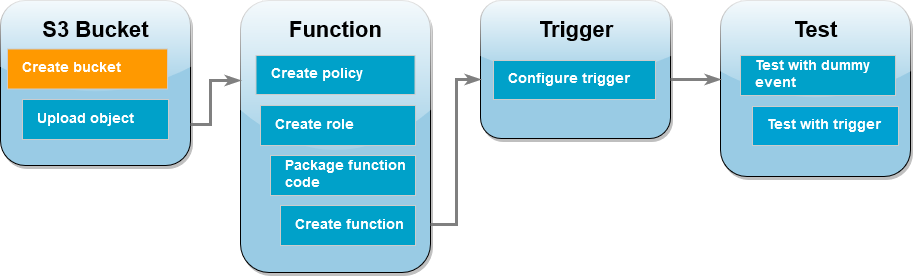
Project 1-Serverless image processing



**Step 1-** sign in to your aws account.

# **Step 2-** create a bucket.

****

## To create the Amazon S3 buckets (console)

1. Open the [Buckets](https://console.aws.amazon.com/s3/buckets) page of the Amazon S3 console.
2. Choose **Create bucket**.
3. Under **General configuration**, do the following:
   1. We create our source bucket ***msmainbucket.***
   2. For **AWS Region**, choose ***the Asia Pacific (Mumbai) south-ap-1***.
4. Leave all other options set to their default values and choose **Create bucket**.
5. Repeat steps 1 to 4 to create your destination bucket ***msresizebucket.***

|  |
| --- |
|  |

# **Step 3**-Upload a test image to your source bucket

# 

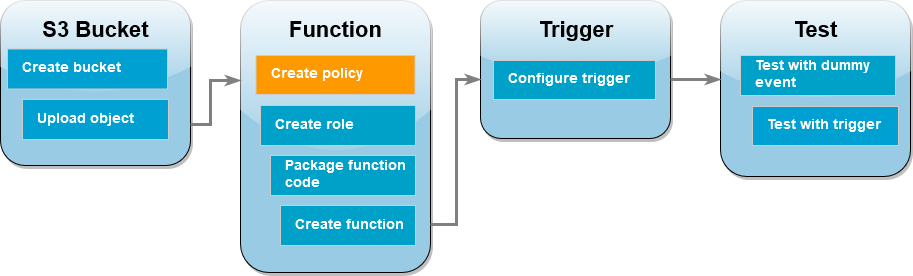
## To upload a jpg image to your source bucket (console)

1. Open the [Buckets](https://console.aws.amazon.com/s3/buckets) page of the Amazon S3 console.
2. Select the source bucket you created in the previous step.
3. Choose **Upload**.
4. Choose **Add files** and use the file selector to choose the object you want to upload.
5. Choose **Open**, then choose **Upload**.

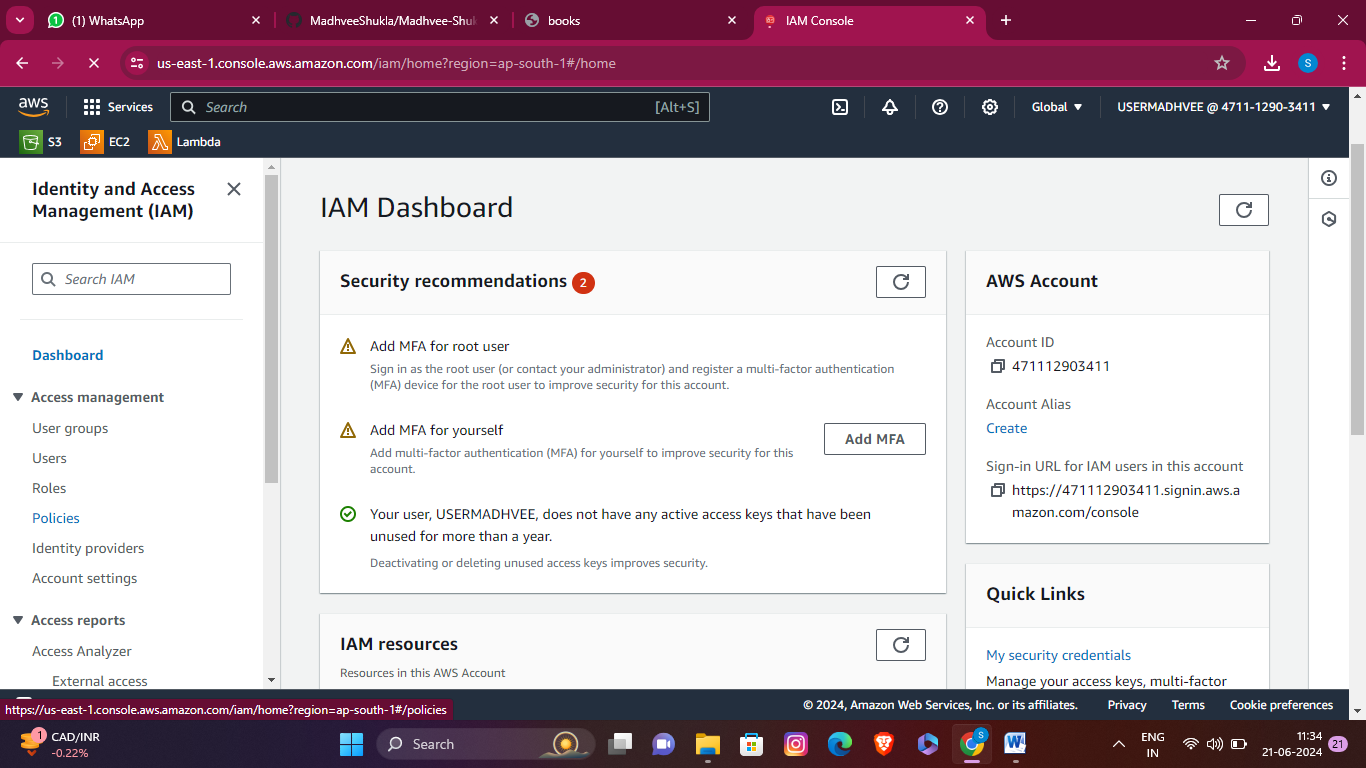
# 

# 

# **Step 4**-Create a permissions policy



1. Open the [Policies](https://console.aws.amazon.com/iamv2/home#policies) page of the AWS Identity and Access Management (IAM) console.



## 

1. Choose **Create policy**.
2. Choose the **JSON** tab, and then paste the following custom policy into the JSON editor.
3. {
4. "Version": "2012-10-17",
5. "Statement": [

7. {

1. "Effect": "Allow",
2. "Action": [
3. "logs:PutLogEvents",
4. "logs:CreateLogGroup",
5. "logs:CreateLogStream"

13. ],

14. "Resource": "arn:aws:logs:\*:\*:\*"

15. },

16. {

1. "Effect": "Allow",
2. "Action": [
3. "s3:GetObject"

20. ],

21. "Resource": "arn:aws:s3:::msmainbucket/\*"

22. },

23. {

1. "Effect": "Allow",
2. "Action": [
3. "s3:PutObject"

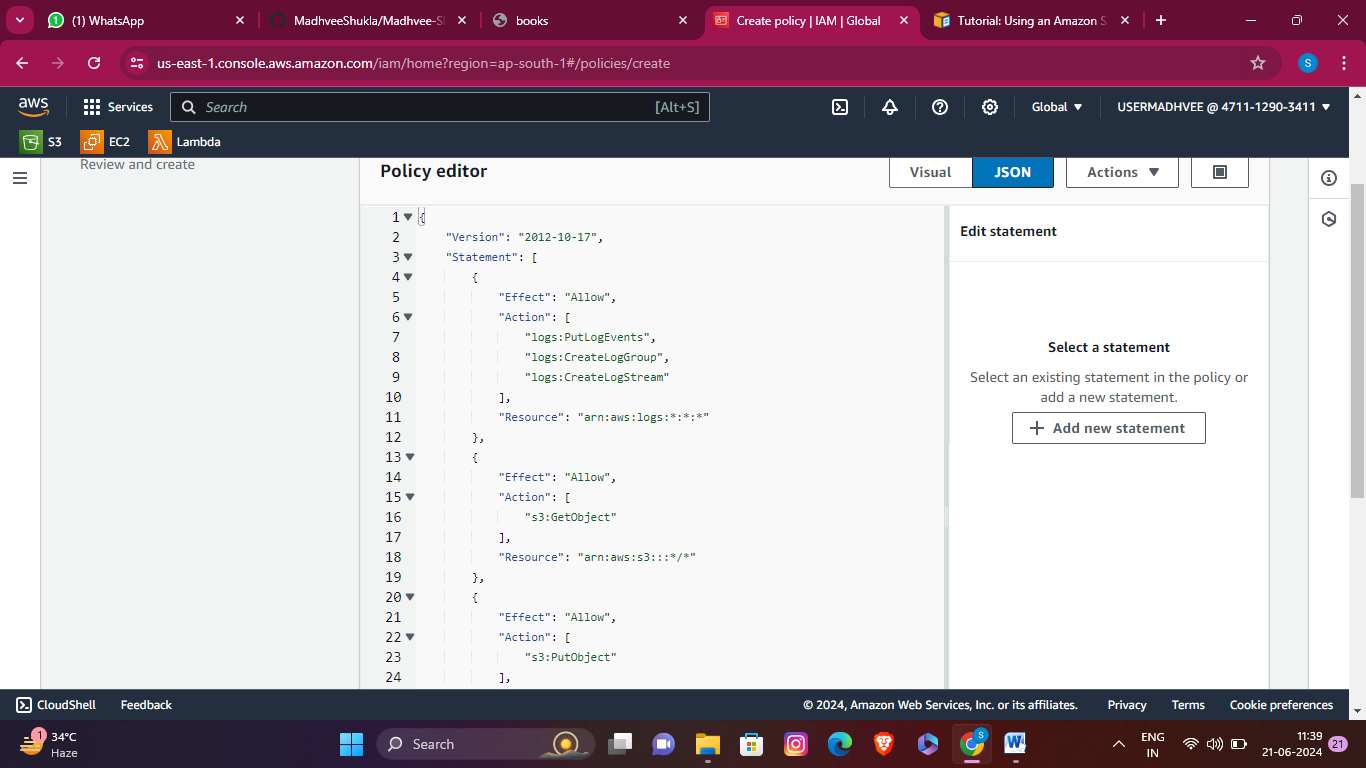
27. ],

28. "Resource": "arn:aws:s3:::msresizebucket/\*"

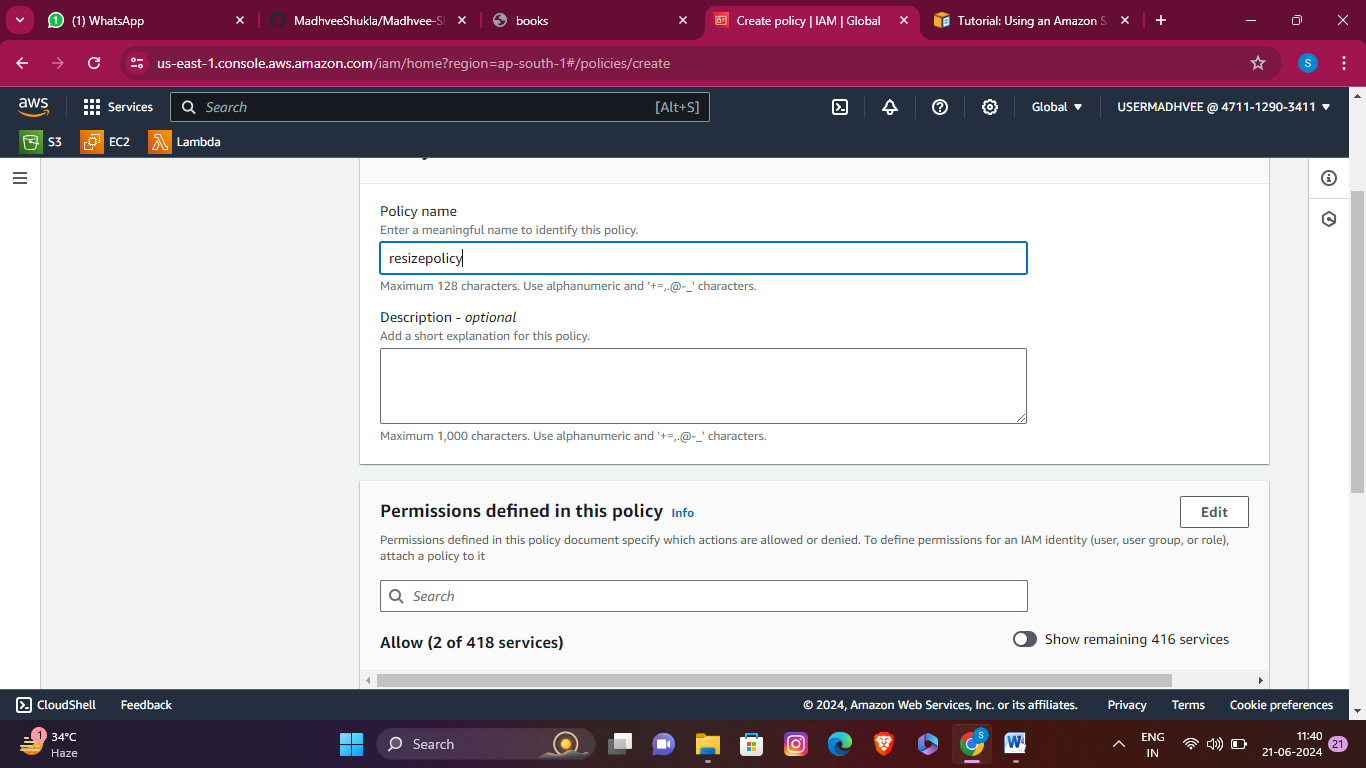
29. }

30. ]

}

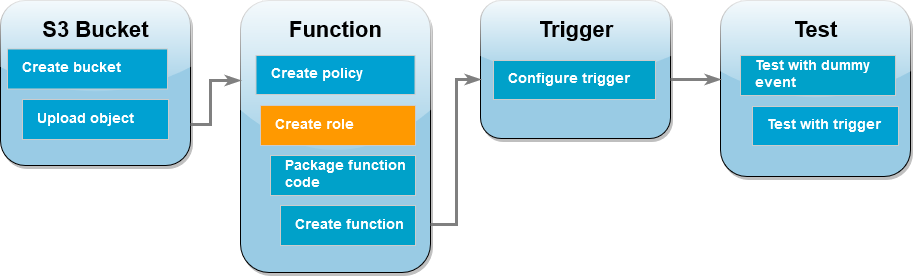


1. Choose **Next**.
2. Under **Policy details**, for **Policy name**, enter ***resizepolicy.***

******

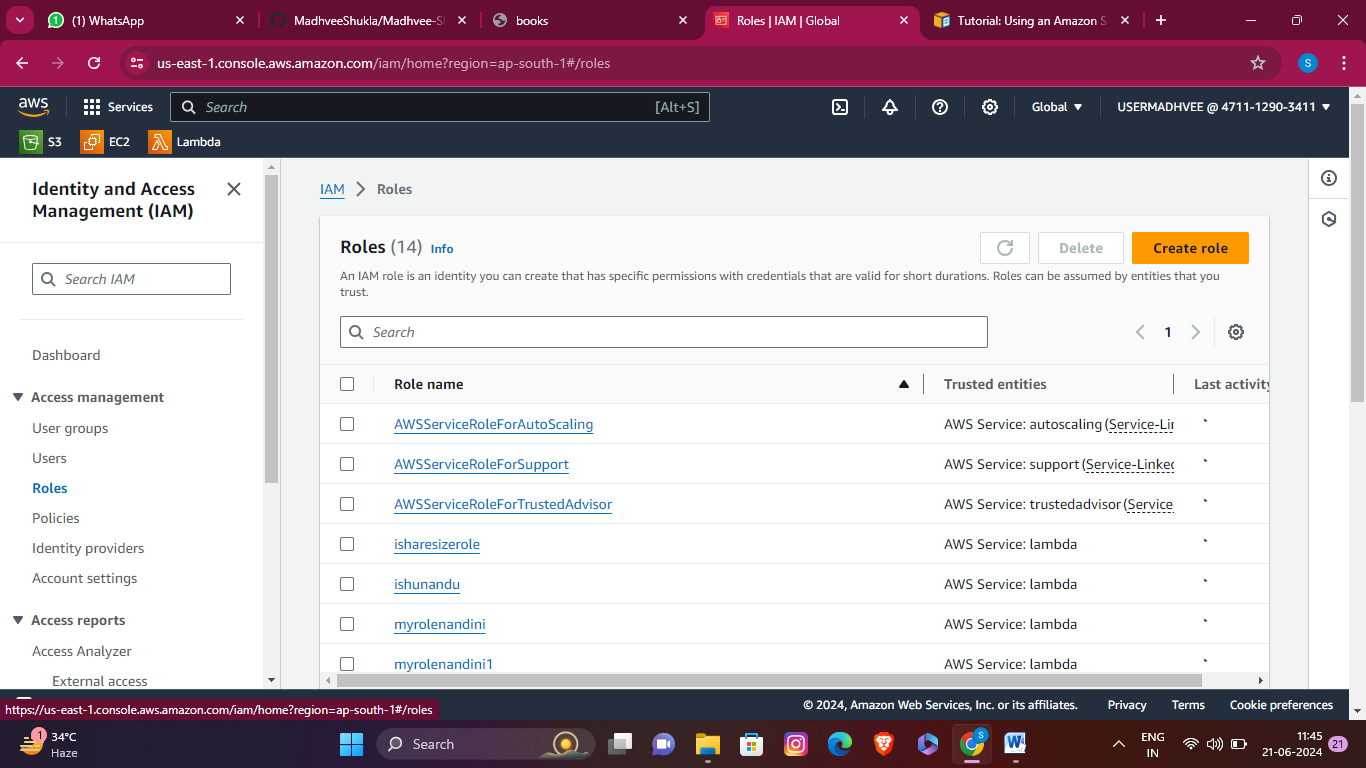
1. Choose **Create policy.**

**Step 5-**Create an execution role

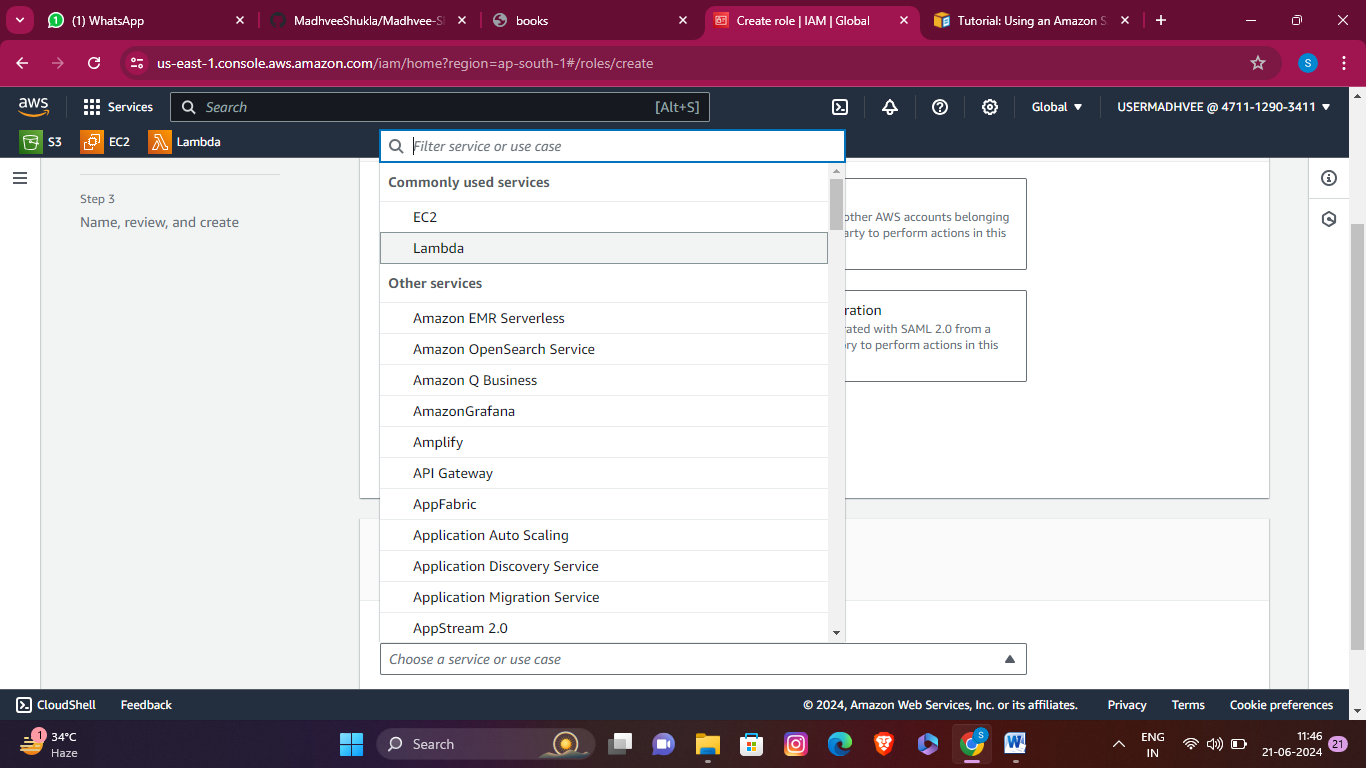


## To create an execution role and attach your permissions policy (console)

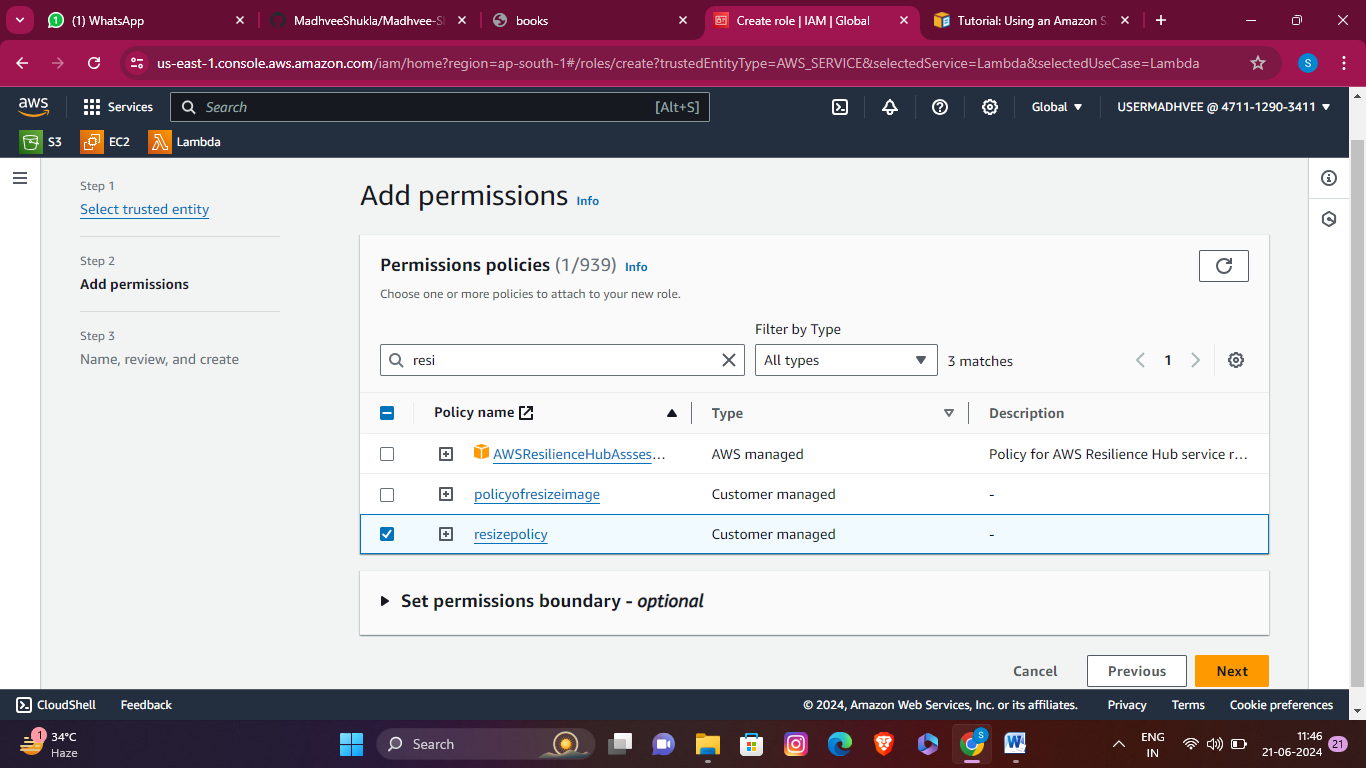
1. Open the [Roles](https://console.aws.amazon.com/iamv2/home#roles) page of the (IAM) console.



1. Choose **Create role**.
2. select **AWS service**, and for **Use case**, select **Lambda**.

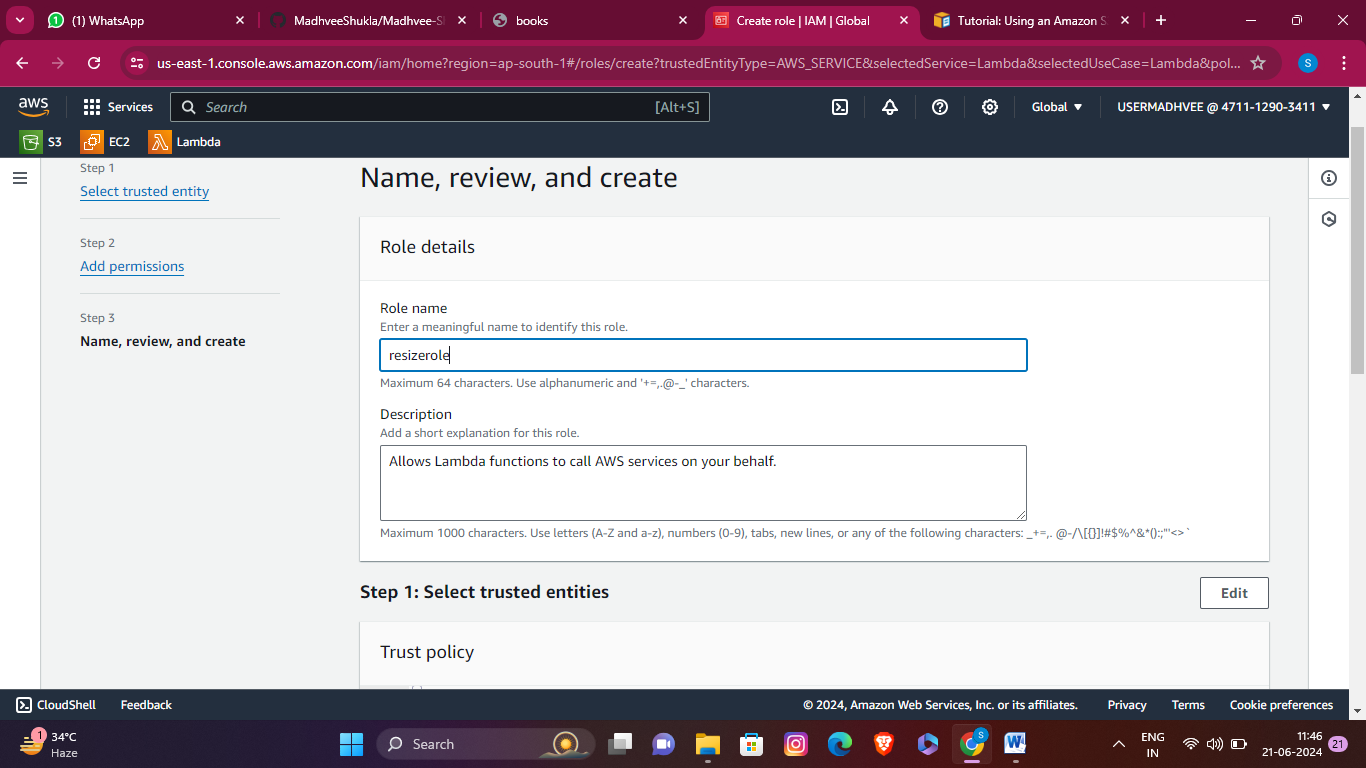


1. Choose **Next**.
2. Add the permissions policy you created in the previous step by doing the following:
   1. In the policy search box, enter ***resize policy*** and choose it.



* 1. Choose **Next**.

1. Under **Role details**, for the **Role name** enter ***resizerole.***

******

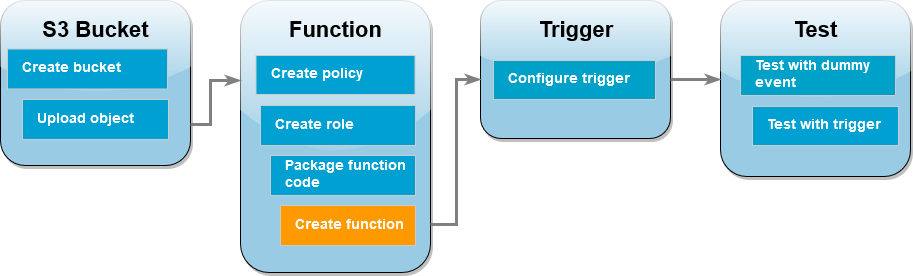
1. Choose **Create role**.

# **Step 6-**Create the function deployment package

# 

1. We create function.zip file.

**Step 7-**Create the Lambda function



1. Open the [Functions page](https://console.aws.amazon.com/lambda/home%23/functions) of the Lambda console.

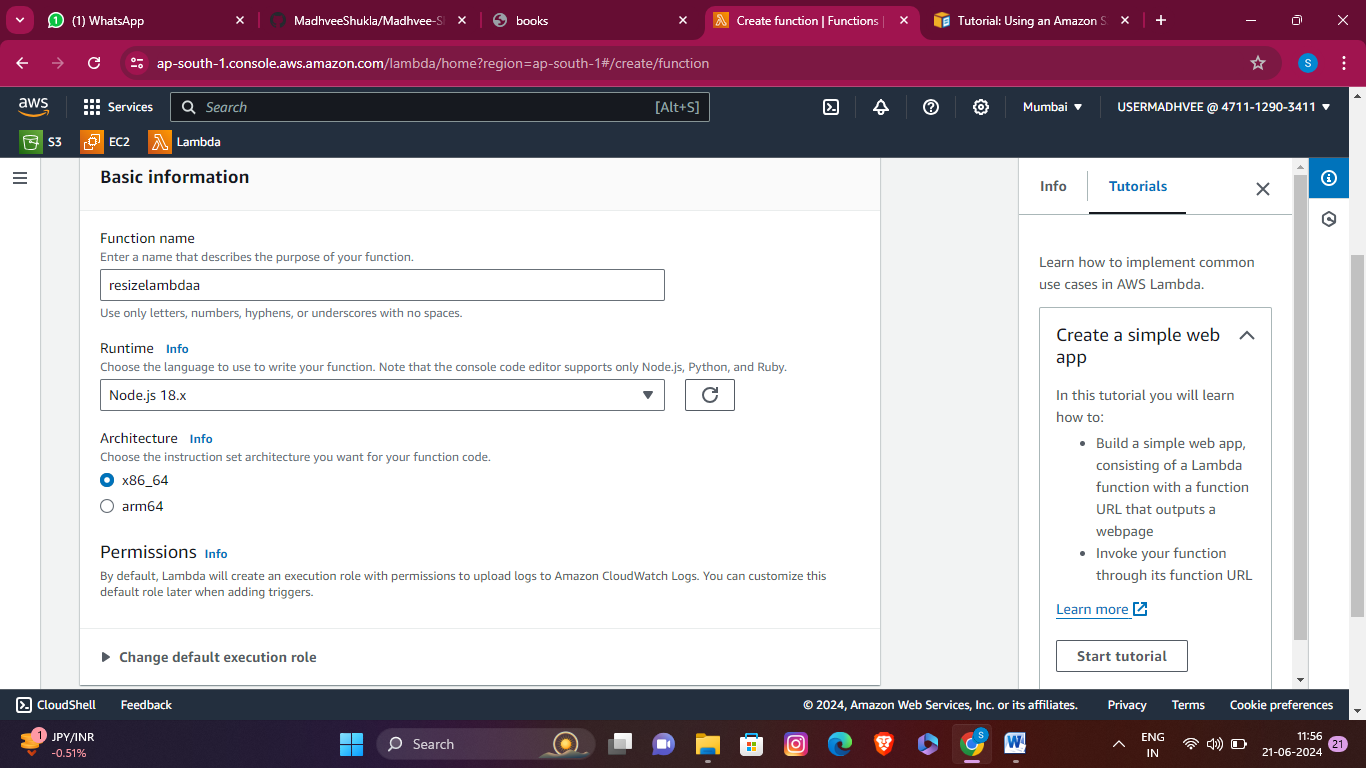
2. Choose **Create function**.

.

3. Under **Basic information**, do the following:

4. For **Function name**, enter **resizelambda**.

5. For **Runtime**choose either **Node.js 18.x** .

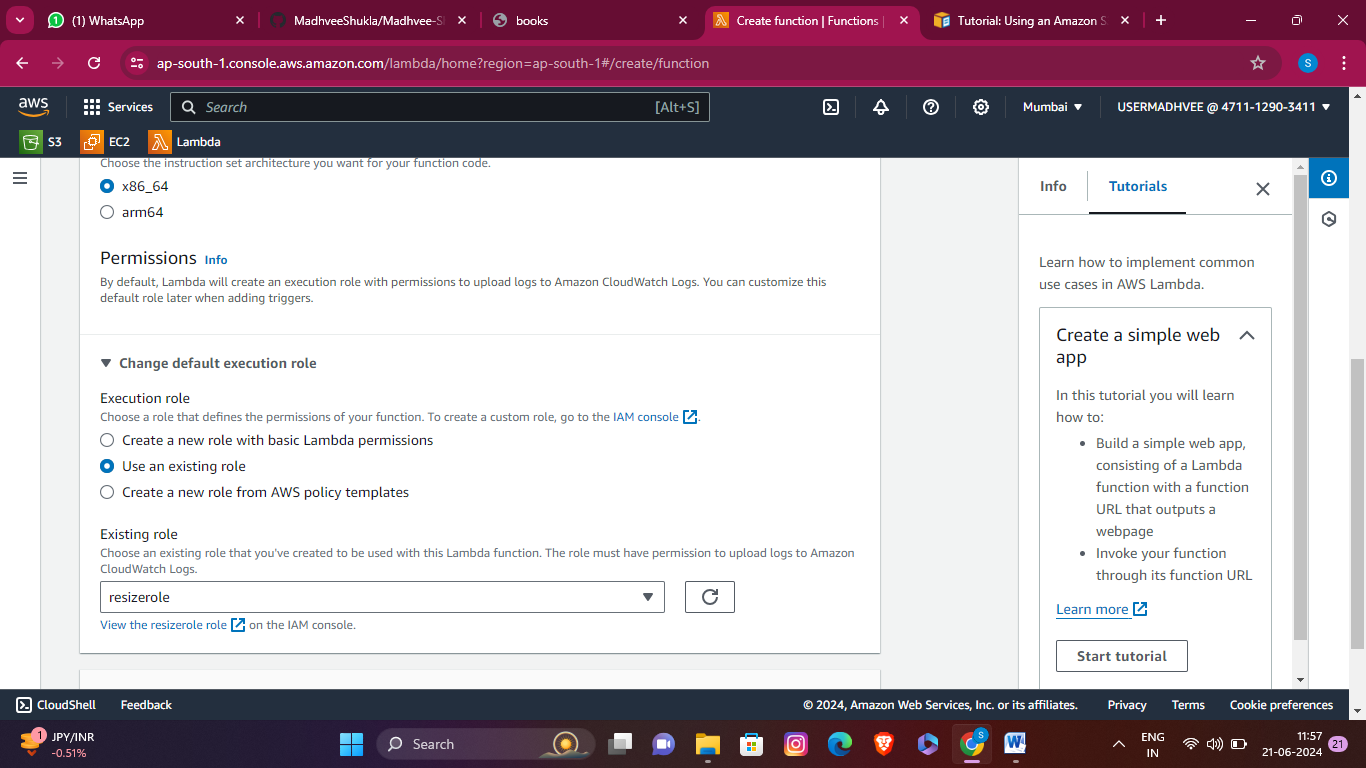


7. For **Architecture**, choose **x86\_64**.

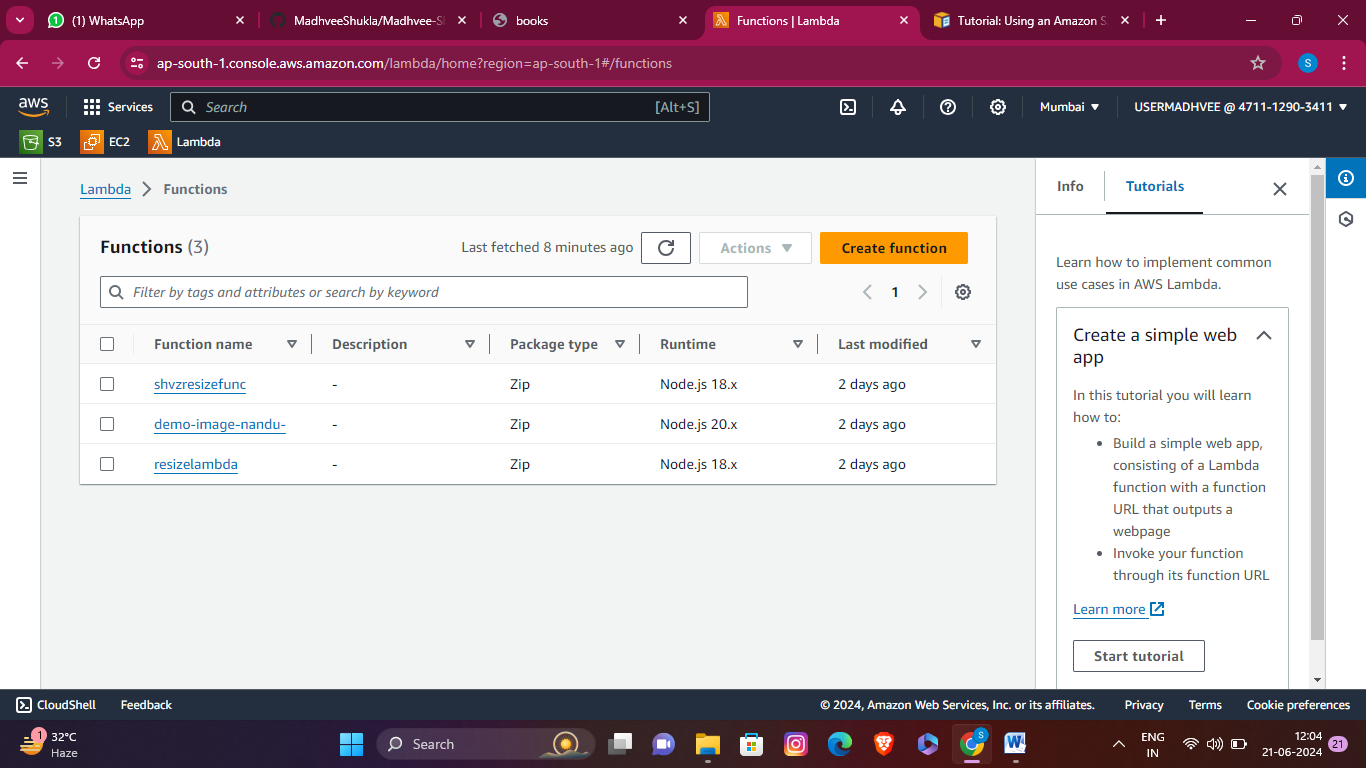
8. In the **Change default execution role** tab, do the following:

9. Expand the tab, then choose **Use an existing role**.

10. Select the ***resizerole*** you created earlier.



11. Choose **Create function**.



# **Step 8-**To upload the function code (console)

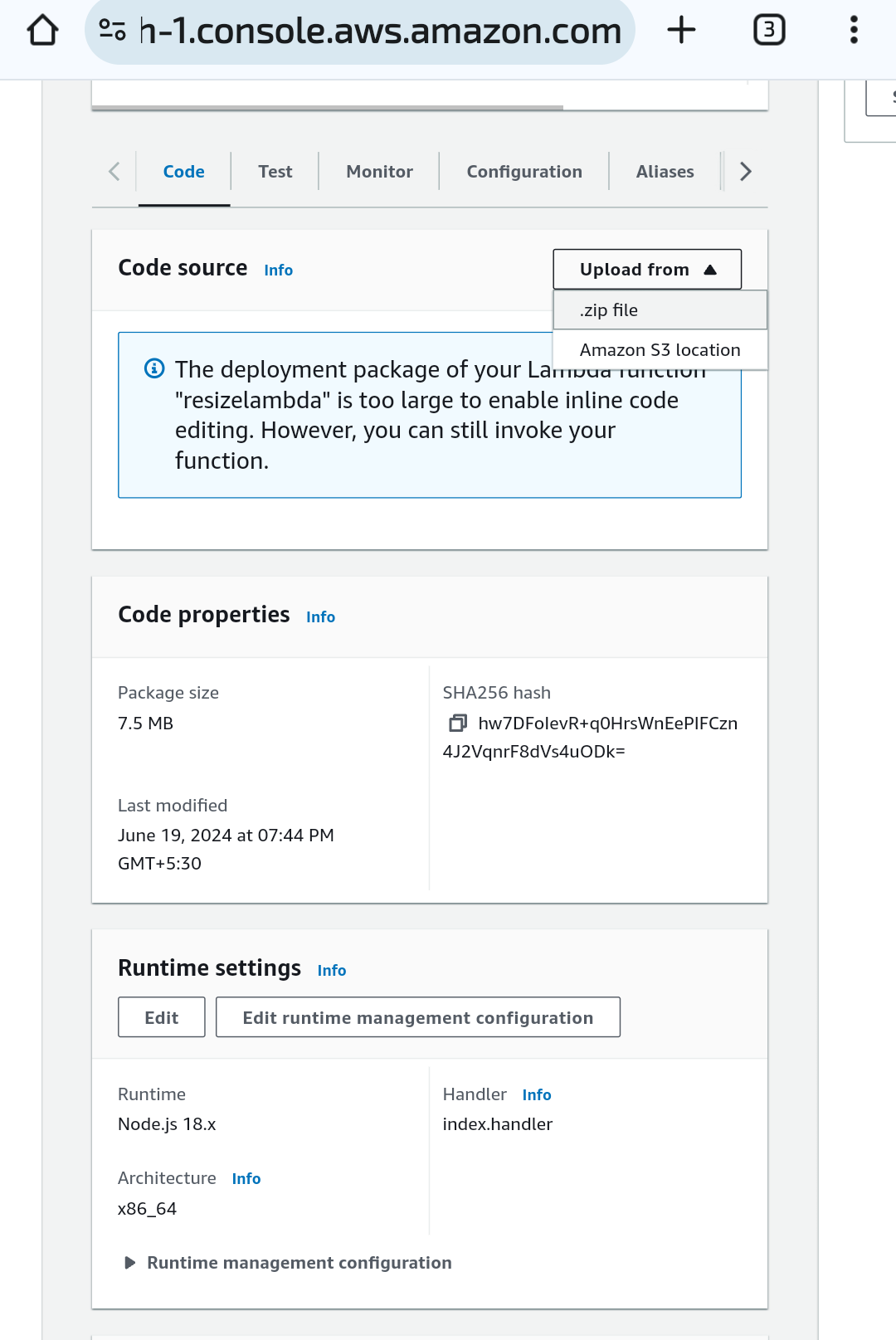
1. In the **Code source** pane, choose **Upload from**.

2. Choose **.zip file**.

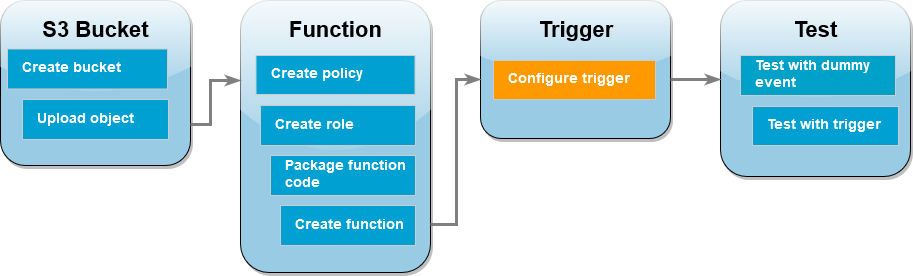
3. Choose **Upload**.

4. In the file selector, select your .zip file and choose **Open**.

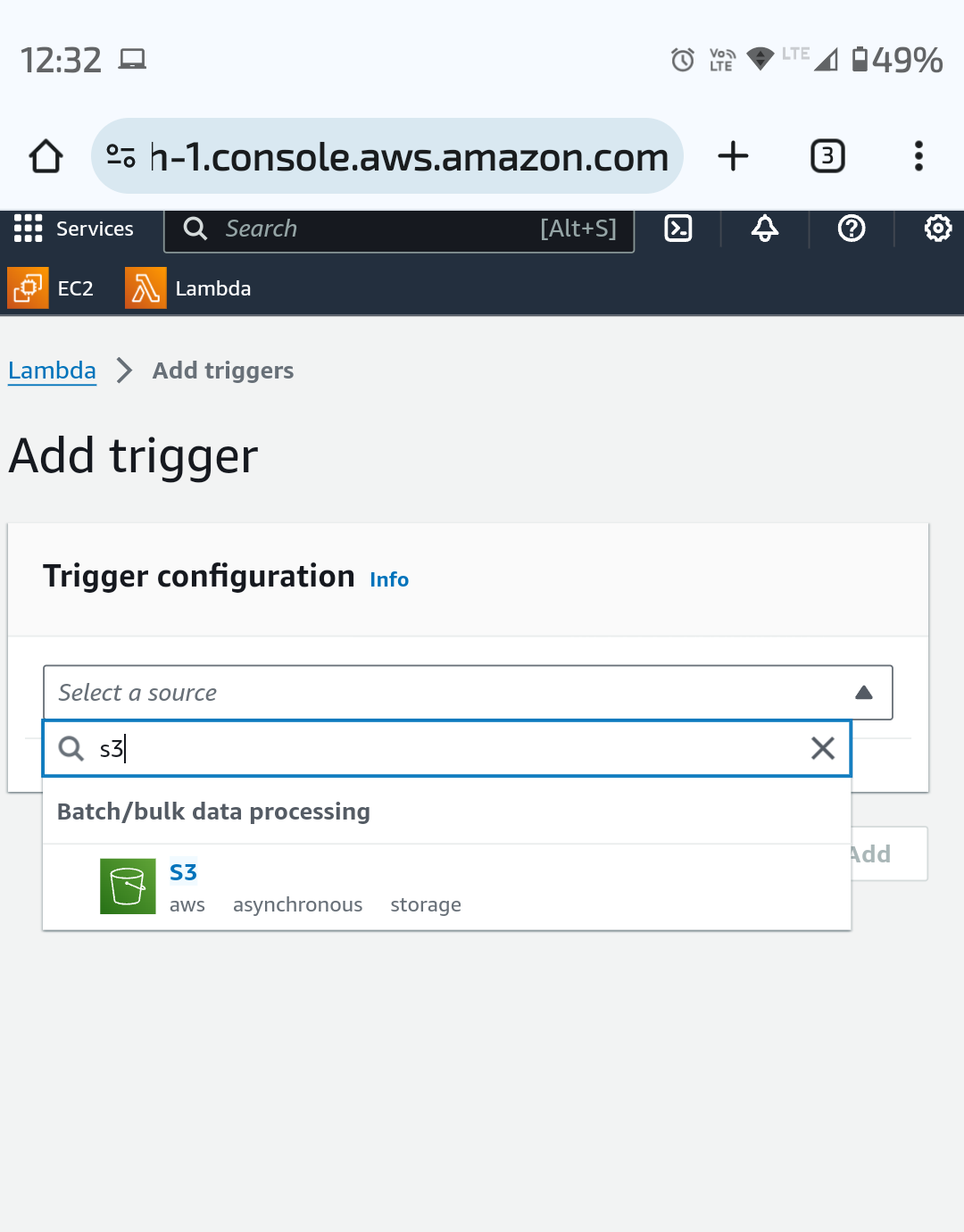
5. Choose **Save**.



**Step 9-**Configure Amazon S3 to invoke the function



* 1. Open the [Functions page](https://console.aws.amazon.com/lambda/home%23/functions) of the Lambda console and choose your function (resizelambda).
  2. Choose **Add trigger**.
  3. Select **S3**.



* 1. Under **Bucket**, select your source bucket.
  2. Set all options are as default.
  3. Choose **save**.

**Step 9-**Test your Lambda function with a dummy event



* + - Open the [Functions page](https://console.aws.amazon.com/lambda/home%23/functions) of the Lambda console and choose your function (resizelambda).
    - Choose the **Test** tab.
    - To create your test event, in the **Test event** pane, do the following:
      * Under **Test event action**, select **Create new event**.
      * For **Template**, select **S3 Put**.
      * Replace the values for the following parameters with your own values.
        + For awsRegion, replace us-east-1 with the AWS Region you created your Amazon S3 buckets in.
        + For name, replace DOC-EXAMPLE-BUCKET with the name of your own Amazon S3 source bucket.
        + For key, replace test%2Fkey with the filename of the test object you uploaded to your source bucket in the step [Upload a test image to your](https://docs.aws.amazon.com/lambda/latest/dg/with-s3-tutorial.html#with-s3-tutorial-test-image) [source bucket](https://docs.aws.amazon.com/lambda/latest/dg/with-s3-tutorial.html#with-s3-tutorial-test-image).

{

"Records": [

{

"eventVersion": "2.0",

"eventSource": "aws:s3",

"awsRegion": *"south-ap-1"*,

eventTime": "1970-01-01T00:00:00.000Z",

"eventName": "ObjectCreated:Put",

"userIdentity": {

"principalId": "EXAMPLE"

},

"requestParameters": {

"sourceIPAddress": "127.0.0.1"

},

"responseElements": {

"x-amz-request-id": "EXAMPLE123456789",

"x-amz-id-2": "EXAMPLE123/5678abcdefghijklambdaisawesome/mnopqrstuvwxyzABCDEFGH"

},

"s3": {

"s3SchemaVersion": "1.0",

"configurationId": "testConfigRule",

"bucket": {

"name": *"msmainbucket"*,

"ownerIdentity": {

"principalId": "EXAMPLE"

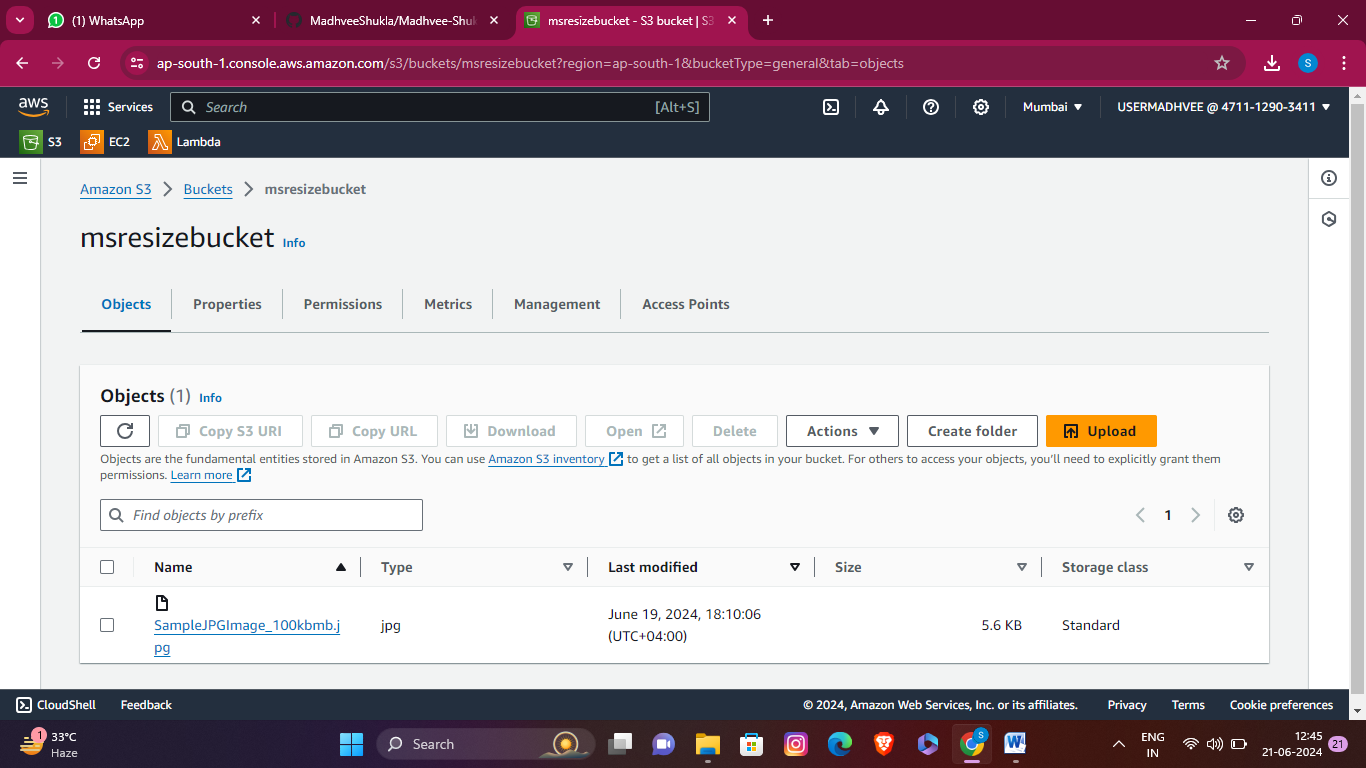
},

"arn": "arn:aws:s3:::msmainbucket "

},

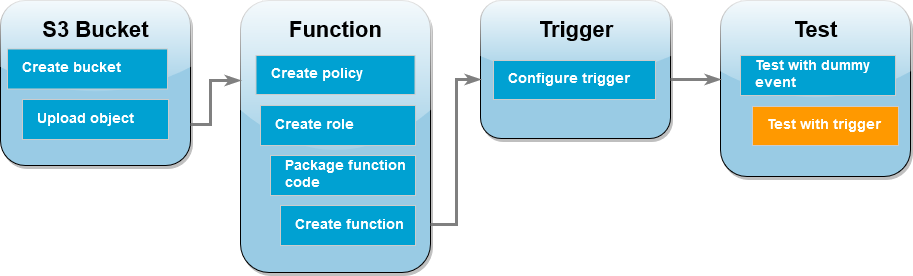
"object": {

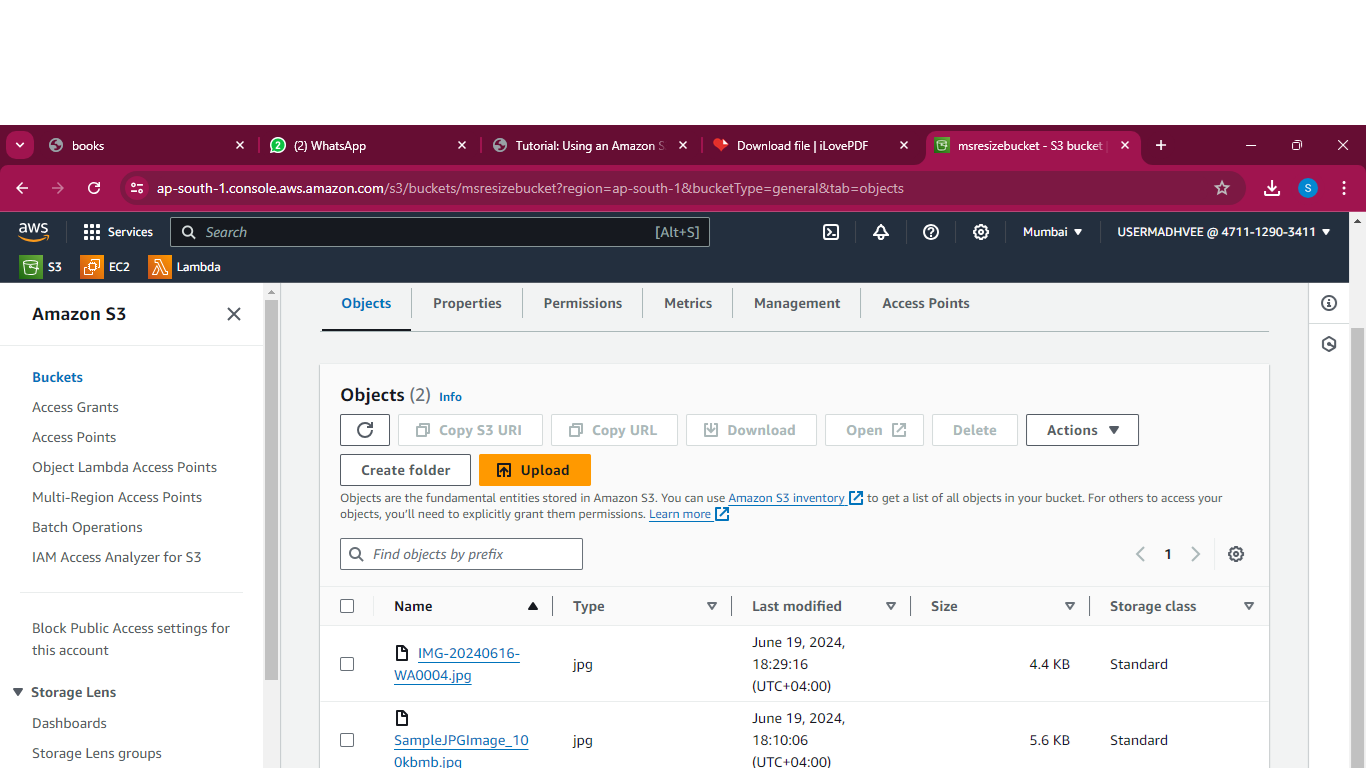
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | "key": *"SampleJPGImage\_100kbmb.jpg"*, |
|  |  |  |  |  | "size": 1024, |
|  |  |  |  |  | "eTag": "0123456789abcdef0123456789abcdef", |
|  | ] | } | } | } | "sequencer": "0A1B2C3D4E5F678901" |

* Choose **Save**..
* To check the your function has created a resized verison of your image and stored it in your target Amazon S3 bucket.
* Now go to the destinationbucket msresizebucket.

"

**Step 10-**Test your function using the Amazon S3 trigger





1. To upload an image to your Amazon S3 bucket, do the following:
2. Open the [Buckets](https://console.aws.amazon.com/s3/buckets) page of the Amazon S3 console and choose your source bucket.
3. Choose **Upload**.
4. Choose **Add files** and use the file selector to choose the image file you want to upload. Your image object can be any .jpg or .png file.
5. Choose **Open**, then choose **Upload**.
6. Verify that Lambda has saved a resized version of your image file in your target bucket by doing the following:
7. Navigate back to the [Buckets](https://console.aws.amazon.com/s3/buckets) page of the Amazon S3 console and choose your destination bucket.
8. In the **Objects** pane, you should now see two resized image files, one from each test of your Lambda function. To download your resized image, select the file, then

9.choose **Download**.

Now we can see that image is automatically resized.

Madhvee shukla

Email- Madhveeshukla9555@gmail.com