CSC PROJECT

M. Lohith

2100032244

Sec:31

Title: Serverless Speech-to-Text with AWS Transcribe and S3 Event Trigger using Lambda and CloudWatch

Step by step process:

Step:1 Creation of S3 bucket:

First, create an S3 bucket to store the audio files that you want to convert audio to text.

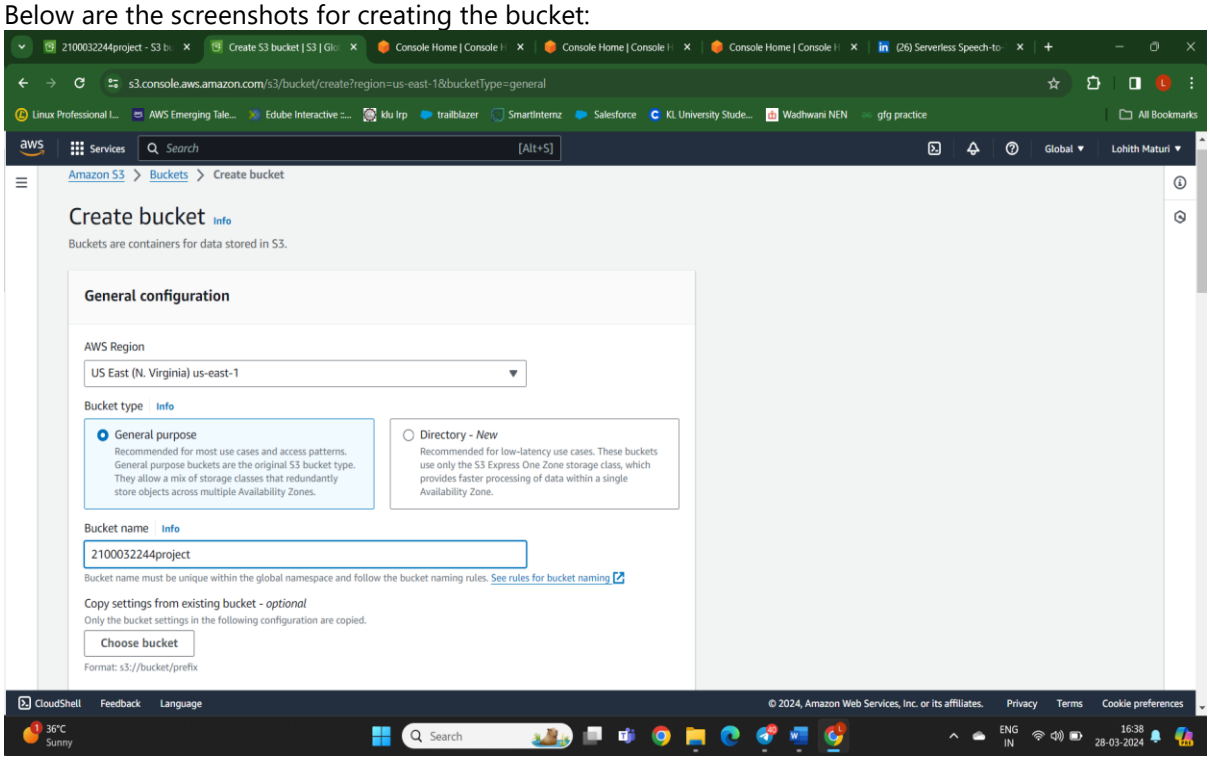
1.Give the unique bucket name

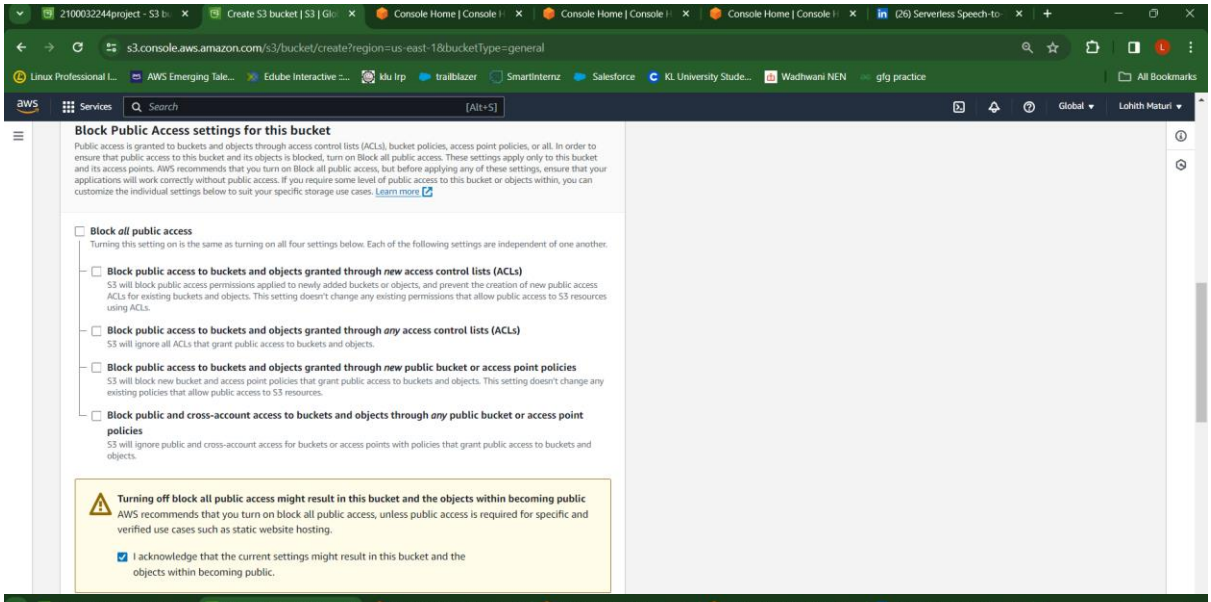
2.Select ACLs enabled

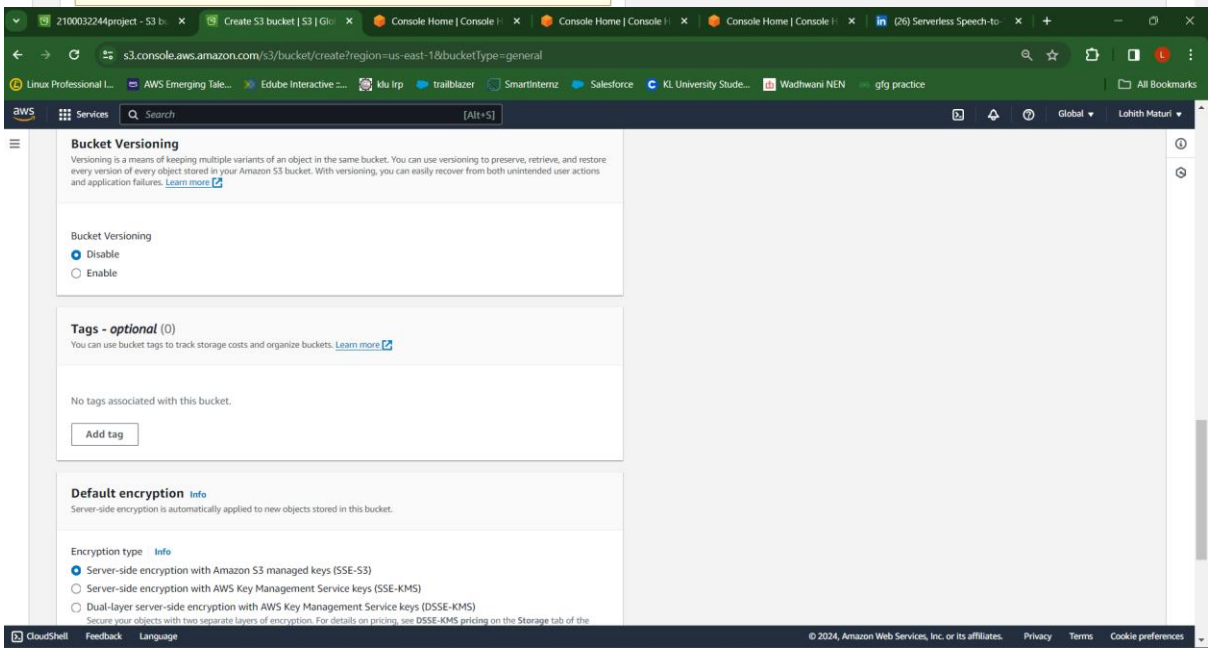
3.Then deselect block public access

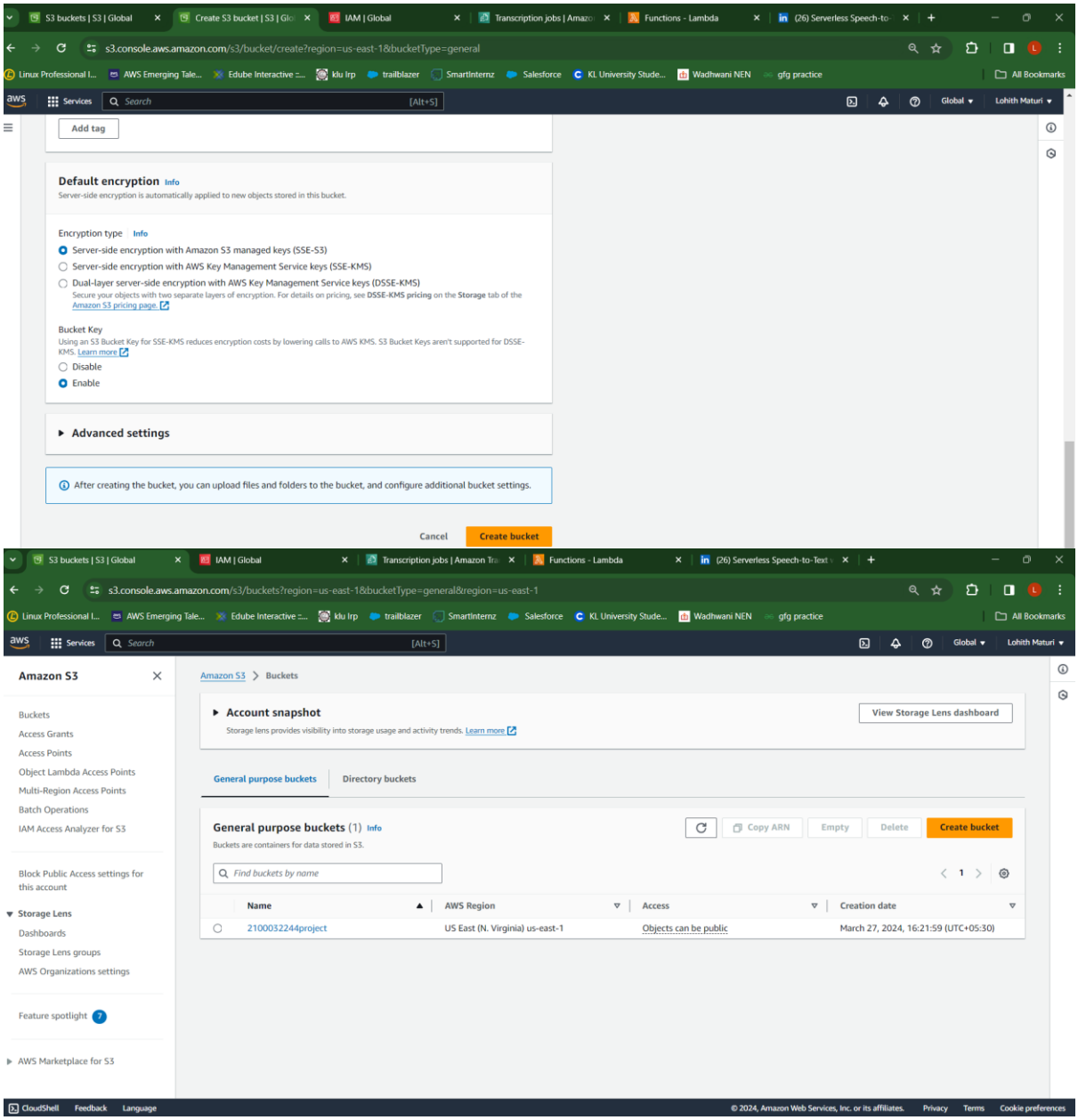
4.Then enable bucket versioning and click on create bucket

Below are the screenshots for creating the bucket:

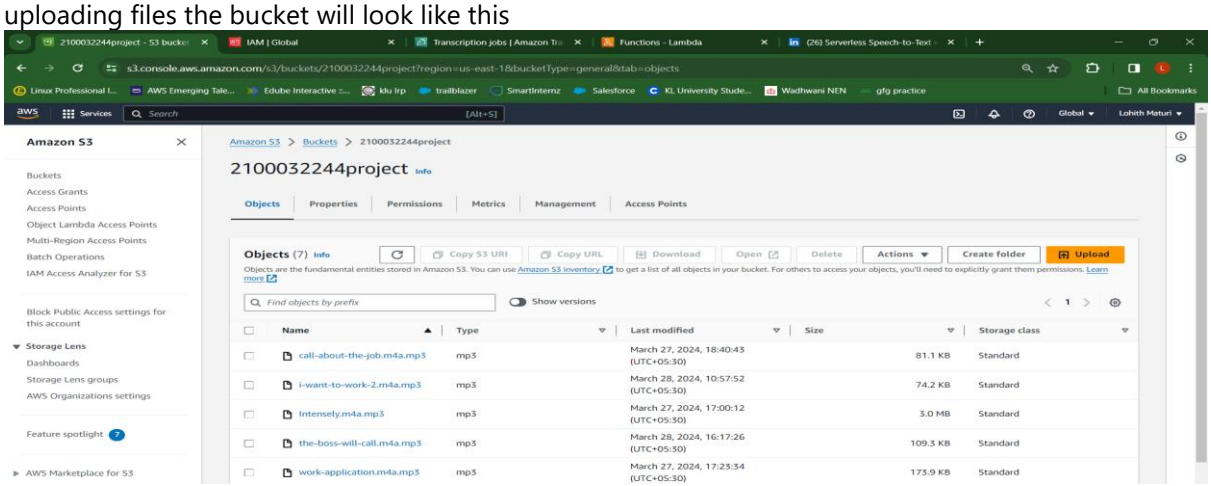



2100032244project - S3 b
Create S3 bucket | S3 | Glo
Console Home | Console
Console Home | Console
Console Home | Console
(26) Serverless Speech-to
s3.console.aws.amazon.com/s3/bucket/create?region=us-east-1&bucketType=general
Linux Professional I...
AWS Emerging Tale...
Edube Interactive :...
klu lrp
trailblazer
SmartInternz
Salesforce
KL University Stude...
Wadhwani NEN
gfg practice
All Bookmarks
aws
Services
Search
[Alt+S]
Global
Lohith Maturi
Block Public Access settings for this bucket
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
Block public access to buckets and objects granted through new access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
Block public access to buckets and objects granted through any access control lists (ACLs)
S3 will ignore all ACLs that grant public access to buckets and objects.
Block public access to buckets and objects granted through new public bucket or access point policies
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
Block public and cross-account access to buckets and objects through any public bucket or access point policies
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.
Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.
I acknowledge that the current settings might result in this bucket and the objects within becoming public.


2100032244project - S3 b
Create S3 bucket | S3 | Glo
Console Home | Console
Console Home | Console
Console Home | Console
(26) Serverless Speech-to
s3.console.aws.amazon.com/s3/bucket/create?region=us-east-1&bucketType=general
Linux Professional I...
AWS Emerging Tale...
Edube Interactive :...
klu lrp
trailblazer
SmartInternz
Salesforce
KL University Stude...
Wadhwani NEN
gfg practice
All Bookmarks
aws
Services
Search
[Alt+S]
Global
Lohith Maturi
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
Disable
Enable
Tags - optional (0)
You can use bucket tags to track storage costs and organize buckets. Learn more
No tags associated with this bucket.
Add tag
Default encryption Info
Server-side encryption is automatically applied to new objects stored in this bucket.
Encryption type Info
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
CloudShell
Feedback
Language
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

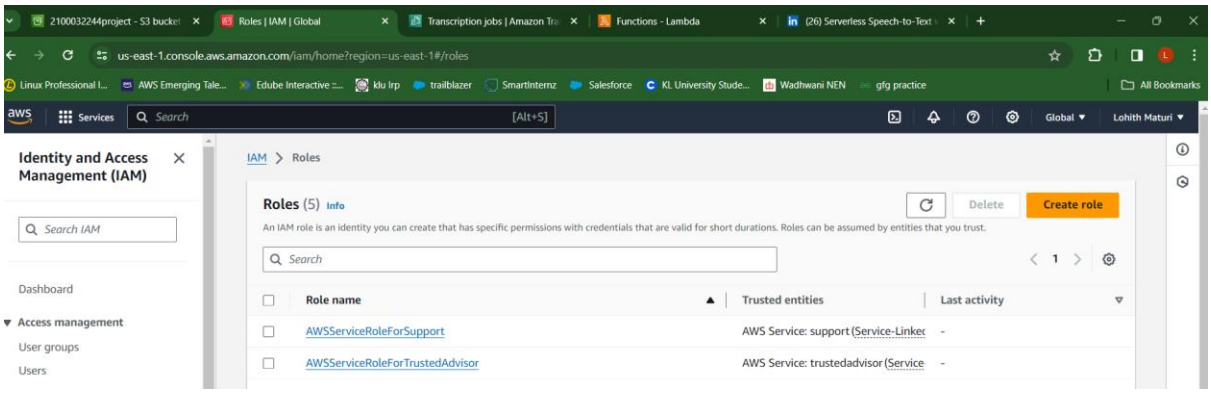


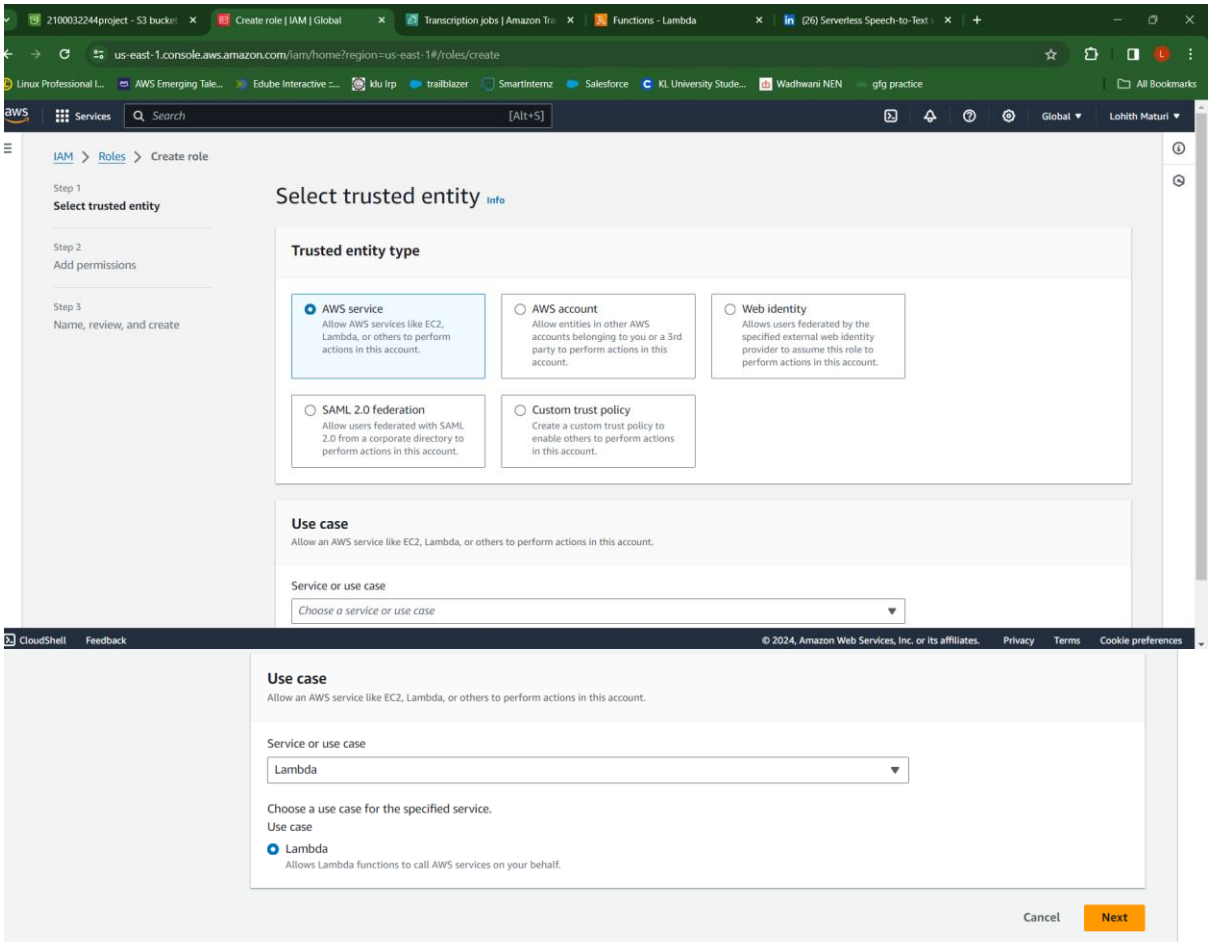
The bucket 2100032244project is successfully created.Now upload the audio files into the bucket.After uploading files the bucket will look like this



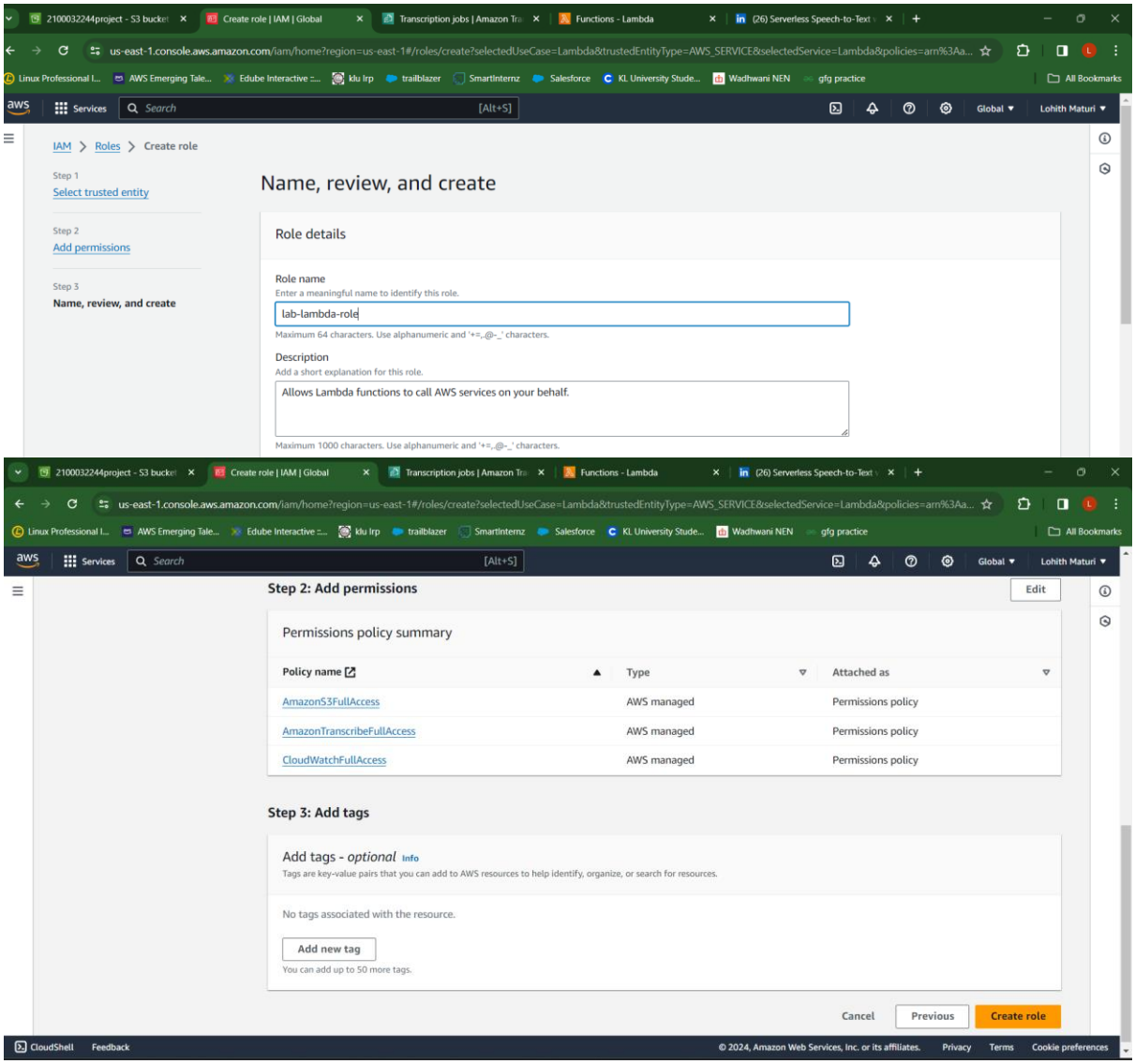
Step 2: Create an IAM role for our lambda function .we will create lambda function after creating the role.

Go to IAM console and click on crete role now we are creating role name called lab-lambda-role



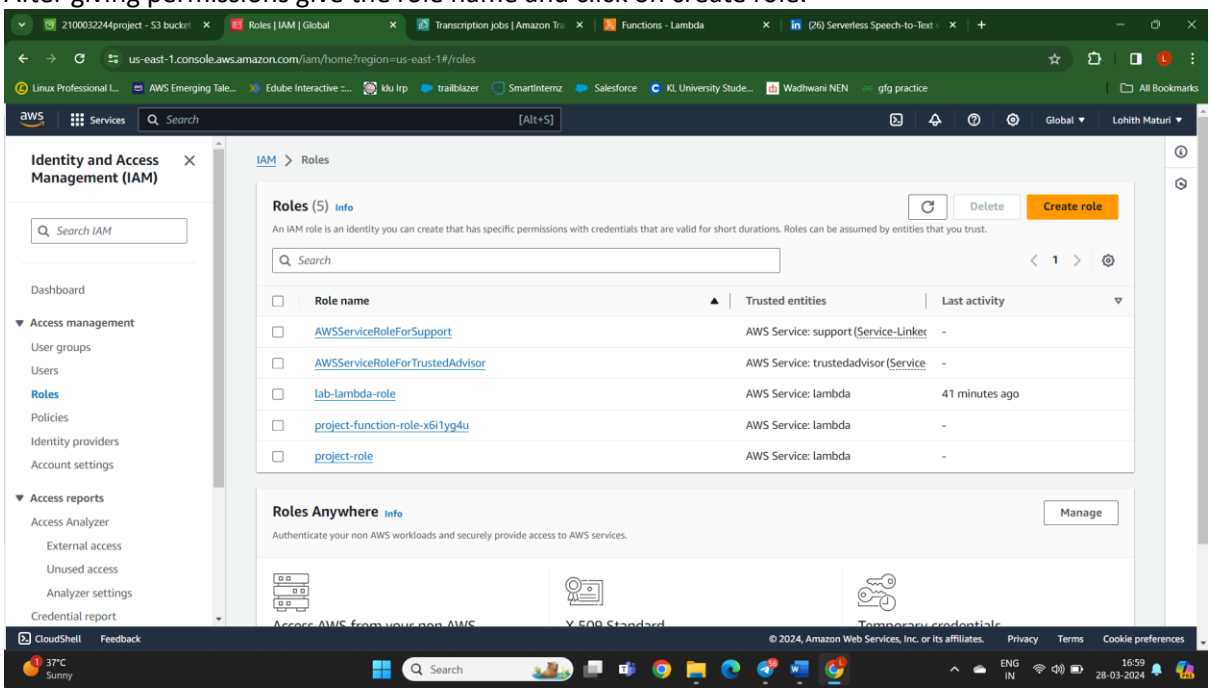


In above screenshot we select usecase as lambda then click on next .In the next step we have to select the permissions which we want to give our lambda function.


2100032244project - S3 bucket
Create role | IAM | Global
Transcription jobs | Amazon Tra
Functions - Lambda
(26) Serverless Speech-to-Text
us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles/create?selectedUseCase=Lambda&trustedEntityType=AWS_SERVICE&selectedService=Lambda&policies=arn%3Aa...
Linux Professional I...
AWS Emerging Tale...
Edube Interactive :...
klu lrp
trailblazer
SmartInternz
Salesforce
KL University Stude...
Wadhwani NEN
gfg practice
All Bookmarks
aws
Services
Search
[Alt+S]
Global
Lohith Maturi
IAM > Roles > Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Name, review, and create
Role details
Role name
Enter a meaningful name to identify this role.
lab-lambda-role
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.
Description
Add a short explanation for this role.
Allows Lambda functions to call AWS services on your behalf.
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

2100032244project - S3 bucket
Create role | IAM | Global
Transcription jobs | Amazon Tra
Functions - Lambda
(26) Serverless Speech-to-Text
us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles/create?selectedUseCase=Lambda&trustedEntityType=AWS_SERVICE&selectedService=Lambda&policies=arn%3Aa...
Linux Professional I...
AWS Emerging Tale...
Edube Interactive :...
klu lrp
trailblazer
SmartInternz
Salesforce
KL University Stude...
Wadhwani NEN
gfg practice
All Bookmarks
aws
Services
Search
[Alt+S]
Global
Lohith Maturi
Step 2: Add permissions
Edit
Permissions policy summary
Policy name
Type
Attached as
AmazonS3FullAccess
AWS managed
Permissions policy
AmazonTranscribeFullAccess
AWS managed
Permissions policy
CloudWatchFullAccess
AWS managed
Permissions policy
Step 3: Add tags
Add tags - optional Info
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.
No tags associated with the resource.
Add new tag
You can add up to 50 more tags.
Cancel
Previous
Create role
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

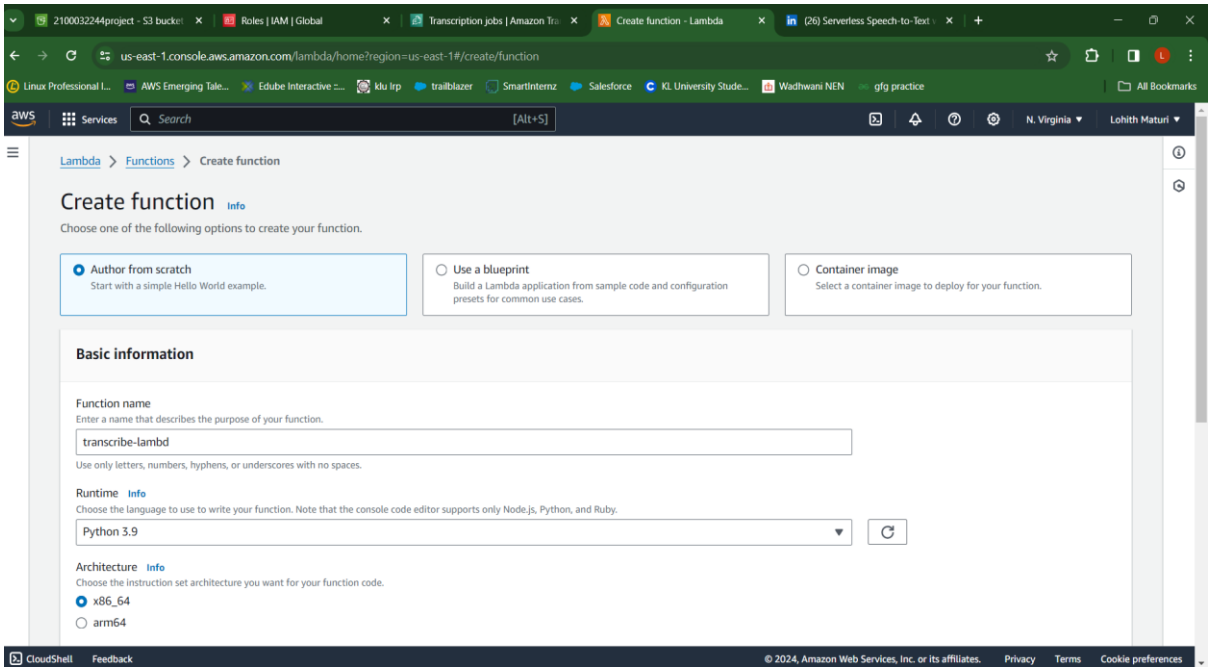
After giving permissions give the role name and click on create role.

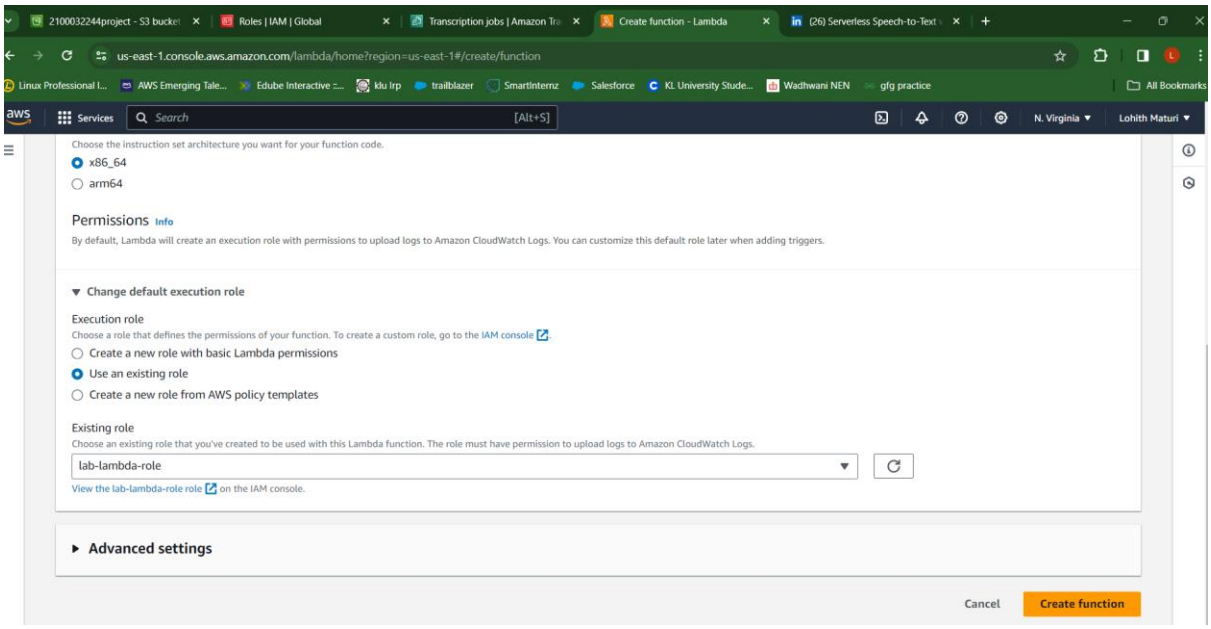


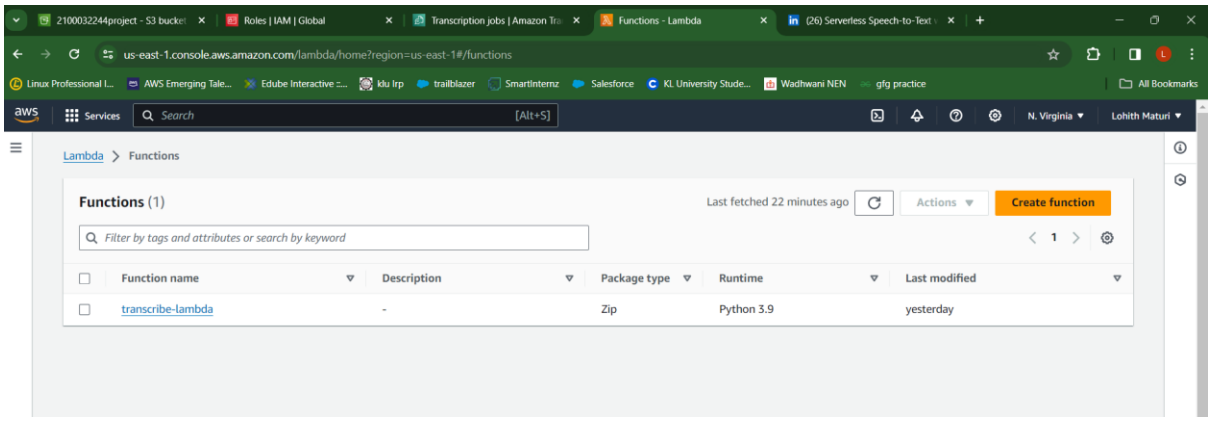
The IAM role lab-lambda-role was created.

Step3: Creating a lambda function

Create a new Lambda function in the AWS console. Choose "Python 3.9" as the runtime and select the IAM role you created in step 2. Use the code given below in the Lambda function editor.The lambda function name is lab-lambda-role. Below are the step by step process to create a lambda function. Select the existing IAM role which we were crated in above step.After selecting the role click on create the our lambda function will be created.







Our lambda function is created.

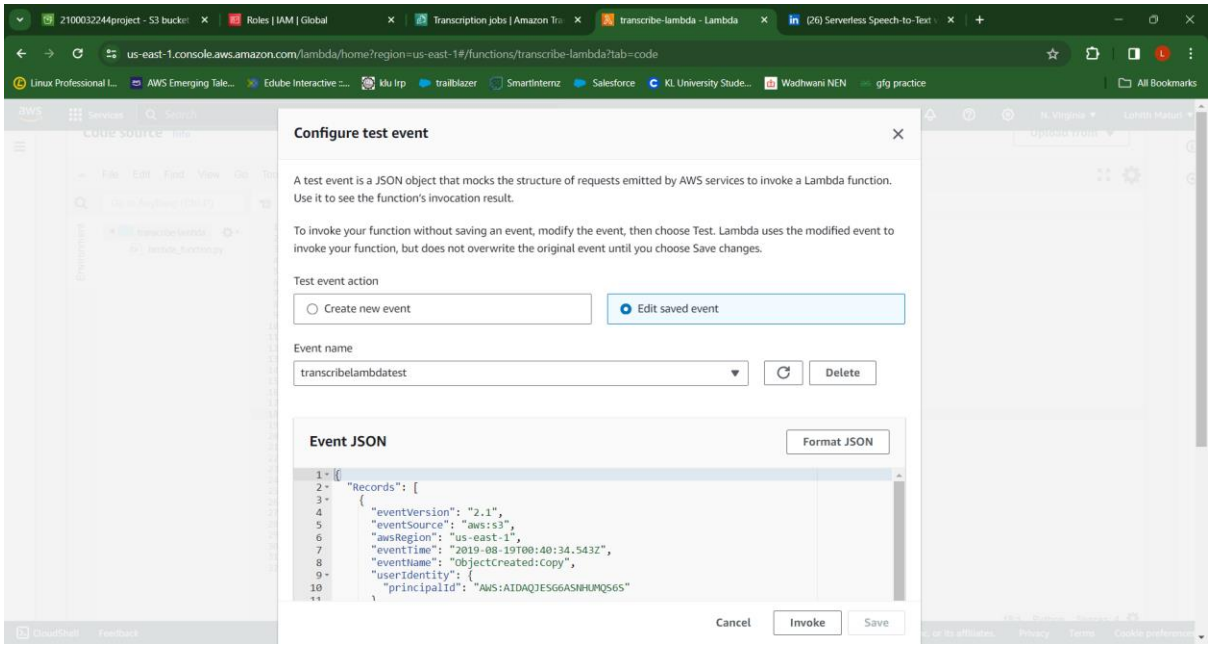
Lambda code:

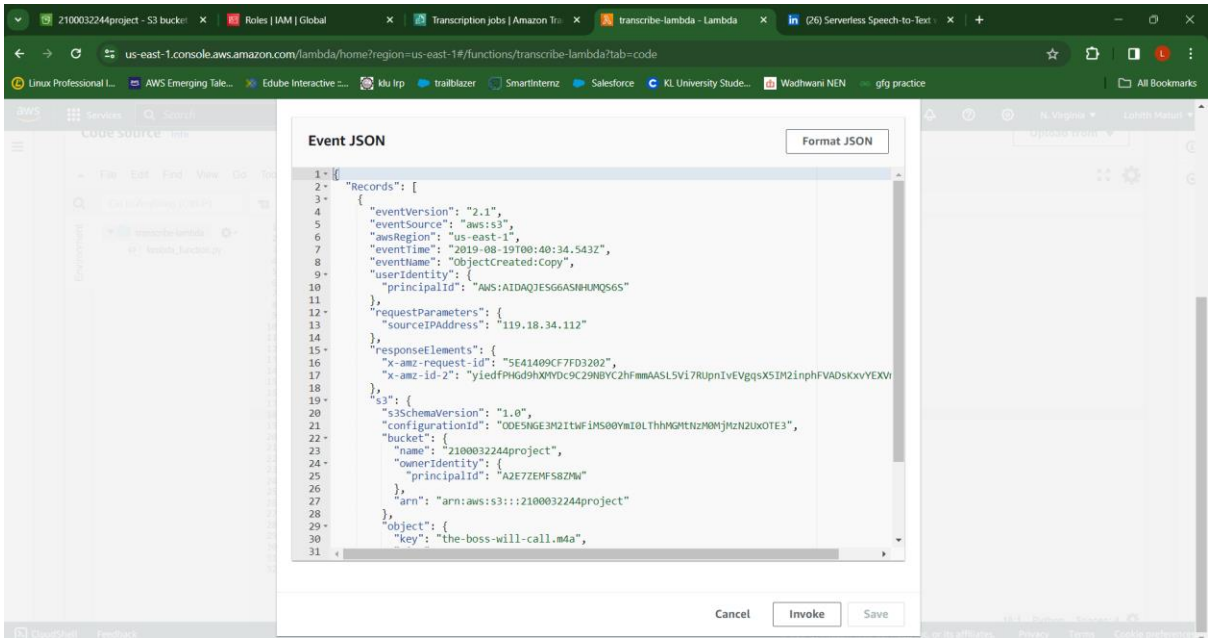
```
import boto3

import uuid

import json

def lambda_handler(event, context):

    print(json.dumps(event))

    record = event['Records'][0]

    s3bucket = record['s3']['bucket']['name']

    s3object = record['s3']['object']['key']
```

```
s3Path = "s3://" + s3bucket + "/" + s3object
jobName = s3object + '-' + str(uuid.uuid4())
client = boto3.client('transcribe')
response = client.start_transcription_job(
    TranscriptionJobName=jobName,
    LanguageCode='en-US',
    MediaFormat='mp4',
    Media={
        'MediaFileUri': s3Path
    }
)
print(json.dumps(response, default=str))
return {
    'TranscriptionJobName': response['TranscriptionJob']['TranscriptionJobName']
}
```

Now copy the code and paste it in lambda function

Now create the test event:





Code for test event is:


```
{
  "Records": [
    {
      "eventVersion": "2.1",
      "eventSource": "aws:s3",
      "awsRegion": "us-east-1",
      "eventTime": "2019-08-19T00:40:34.543Z",
      "eventName": "ObjectCreated:Copy",
      "userIdentity": {
        "principalId": "AWS:AIDAQJESG6ASNHUMQS6S"
      },
      "requestParameters": {
        "sourceIPAddress": "119.18.34.112"
      },
      "responseElements": {
        "x-amz-request-id": "5E41409CF7FD3202",
        "x-amz-id-2": "yiedfPHGd9hXMYDc9C29NBYC2hFmmAASL5Vi7RUpnIvEVgqsX5IM2inphFVADsKxvYEXVm9BzY="
      },
```

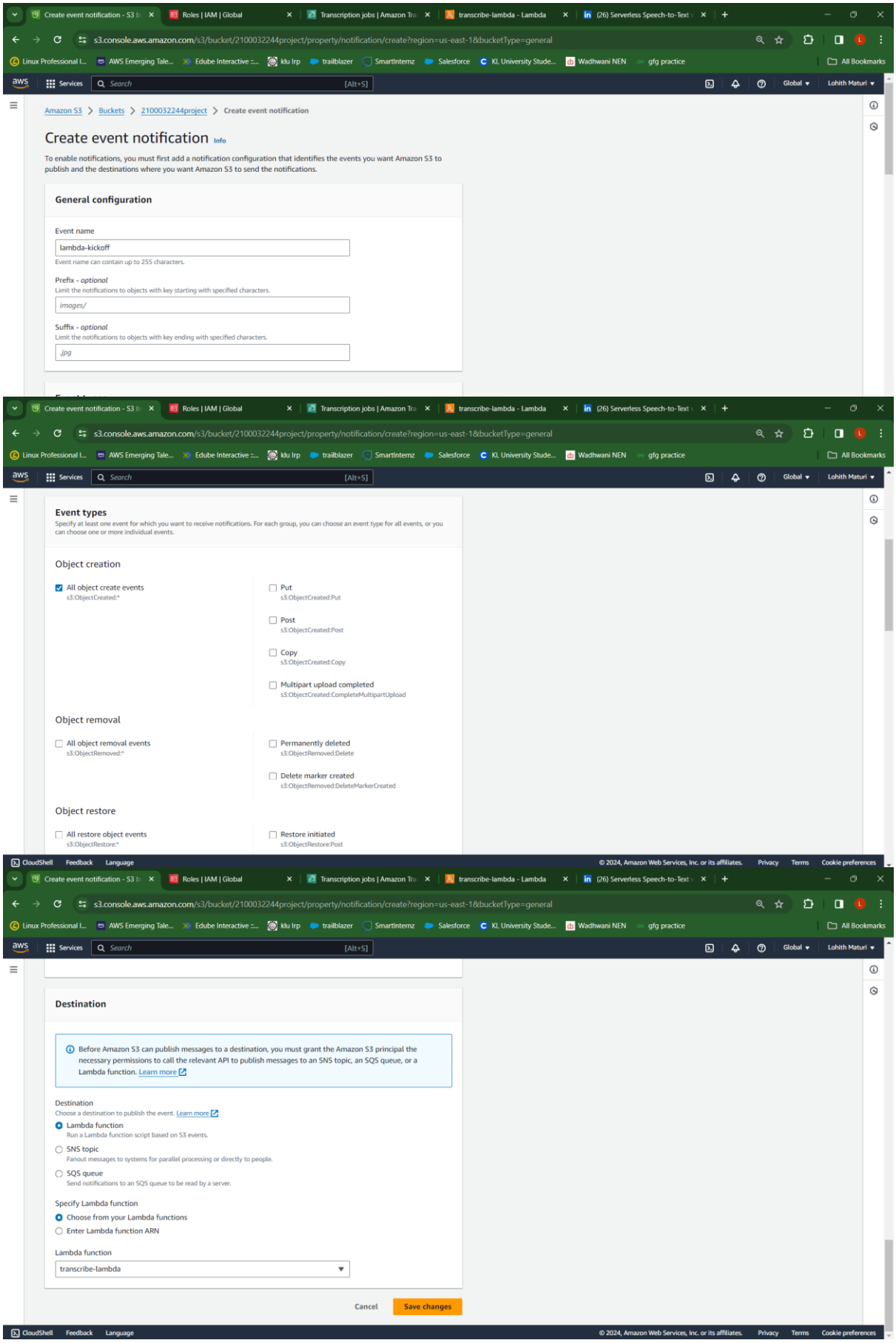
```
    "s3": {
      "s3SchemaVersion": "1.0",
      "configurationId": "ODE5NGE3M2ItWFiMS00YmI0LThhMGMtNzM0MjMzN2UxOTE3",
      "bucket": {
        "name": "2100032244project",
        "ownerIdentity": {
          "principalId": "A2E7ZEMFS8ZMW"
        },
        "arn": "arn:aws:s3:::2100032244project"
      },
      "object": {
        "key": "the-boss-will-call.m4a",
        "size": 1228405,
        "eTag": "7a6afa78089383ef7bfd343302560a2",
        "sequencer": "005D59F0025D9258B"
      }
    }
  }
 ]
}
```


Now the test event is created.

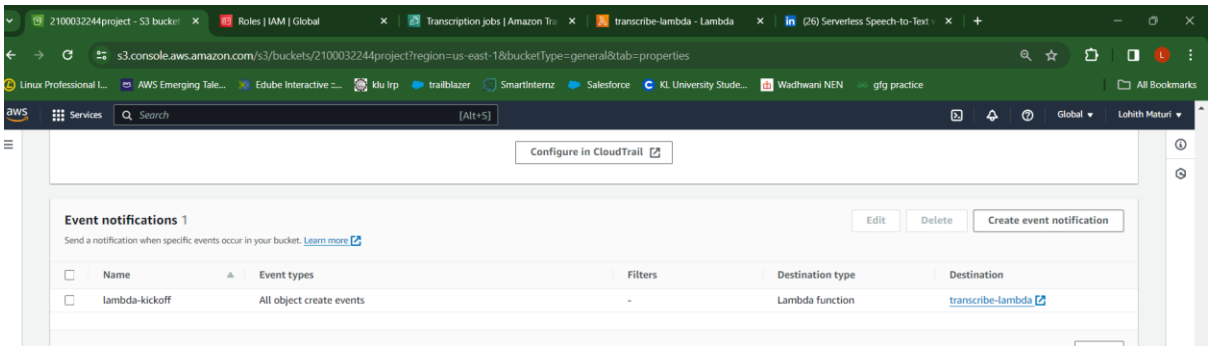
Step 4:Create a s3 event Notification to trigger an event to our lambda function

For this go to s3 bucket and go to properties and scroll down go to event notification and clicl on create event notification. In the Lambda function configuration, add a new trigger for S3 events. Choose the S3 bucket you created in step 1 and set the event type to "ObjectCreated". This will ensure that the Lambda function is triggered every time a new audio file is uploaded to the bucket.

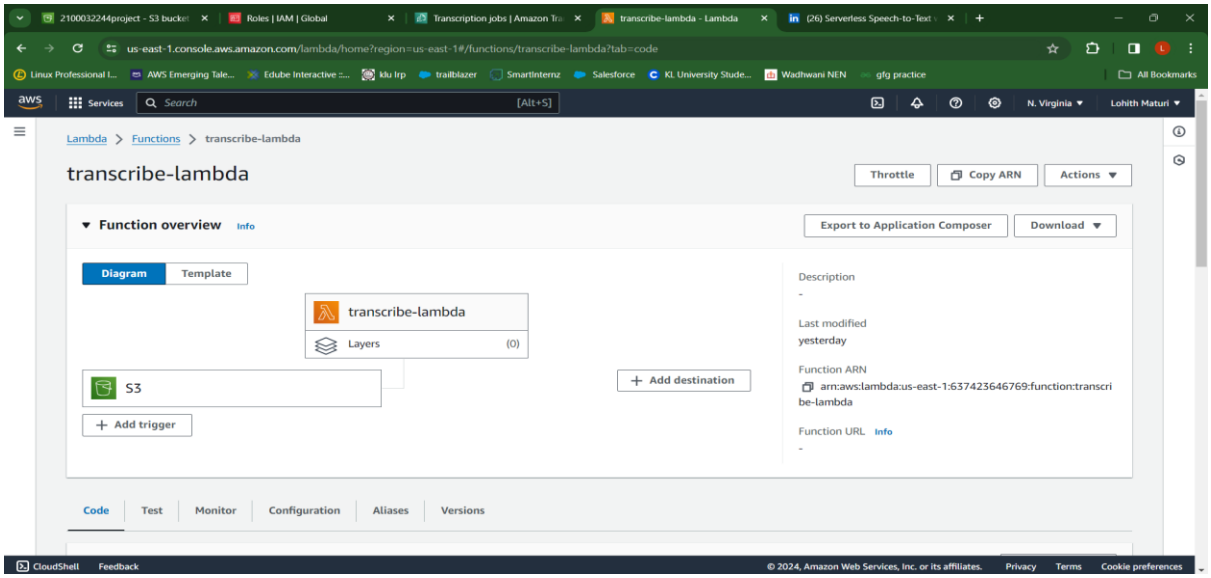
Below are the screenshots for how to create event notification for our lambda function.


Create event notification - S3 b
Roles | IAM | Global
Transcription jobs | Amazon Tra
transcribe-lambda - Lambda
(26) Serverless Speech-to-Text
s3.console.aws.amazon.com/s3/bucket/2100032244project/property/notification/create?region=us-east-1&bucketType=general
Linux Professional I...
AWS Emerging Tale...
Edube Interactive :...
klu lrp
trailblazer
SmartInternz
Salesforce
KL University Stude...
Wadhwani NEN
gfg practice
All Bookmarks
aws
Services
Search
[Alt+S]
Global
Lohith Maturi
Amazon S3 > Buckets > 2100032244project > Create event notification
Create event notification Info
To enable notifications, you must first add a notification configuration that identifies the events you want Amazon S3 to publish and the destinations where you want Amazon S3 to send the notifications.
General configuration
Event name
lambda-kickoff
Event name can contain up to 255 characters.
Prefix - optional
Limit the notifications to objects with key starting with specified characters.
images/
Suffix - optional
Limit the notifications to objects with key ending with specified characters.
.jpg
Event types
Specify at least one event for which you want to receive notifications. For each group, you can choose an event type for all events, or you can choose one or more individual events.
Object creation
All object create events
s3:ObjectCreated:*
Put
s3:ObjectCreated:Put
Post
s3:ObjectCreated:Post
Copy
s3:ObjectCreated:Copy
Multipart upload completed
s3:ObjectCreated:CompleteMultipartUpload
Object removal
All object removal events
s3:ObjectRemoved:*
Permanently deleted
s3:ObjectRemoved:Delete
Delete marker created
s3:ObjectRemoved:DeleteMarkerCreated
Object restore
All restore object events
s3:ObjectRestore:*
Restore initiated
s3:ObjectRestore:Post
CloudShell
Feedback
Language
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Destination
Before Amazon S3 can publish messages to a destination, you must grant the Amazon S3 principal the necessary permissions to call the relevant API to publish messages to an SNS topic, an SQS queue, or a Lambda function. Learn more
Destination
Choose a destination to publish the event. Learn more
Lambda function
Run a Lambda function script based on S3 events.
SNS topic
Fanout messages to systems for parallel processing or directly to people.
SQS queue
Send notifications to an SQS queue to be read by a server.
Specify Lambda function
Choose from your Lambda functions
Enter Lambda function ARN
Lambda function
transcribe-lambda
Cancel
Save changes

Click on save chenges now our event notification is created.



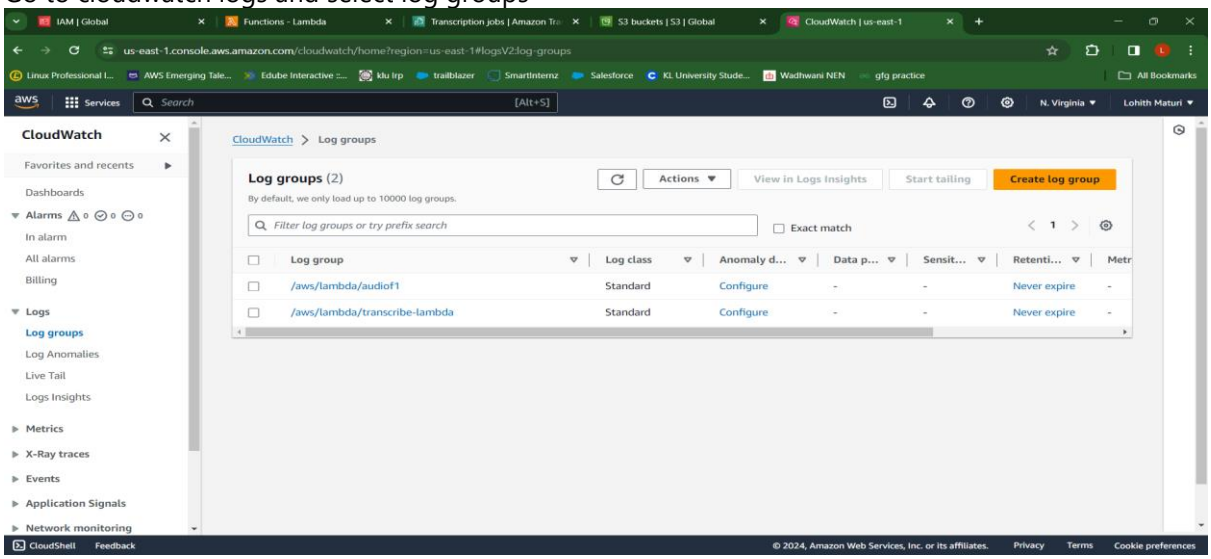
Now go to lamba function we see like this:



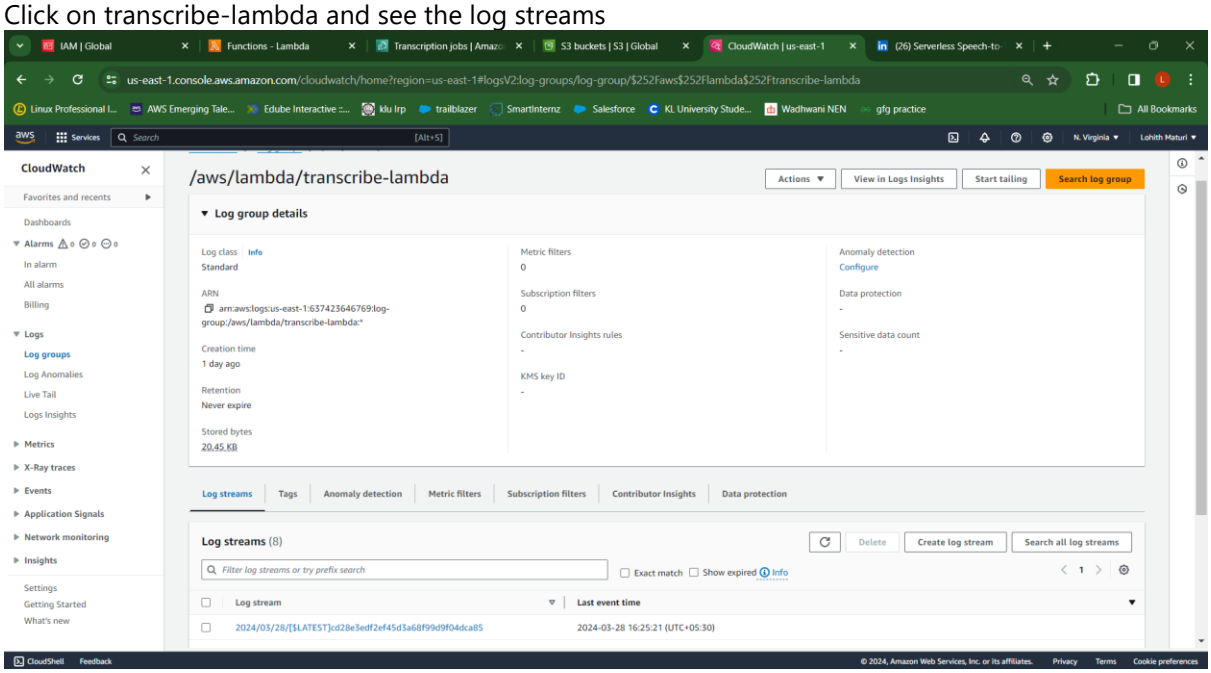
Now we successfully triggered an event to our lambda function.

Step:5 Create a new CloudWatch log group to store the logs generated by the Lambda function.

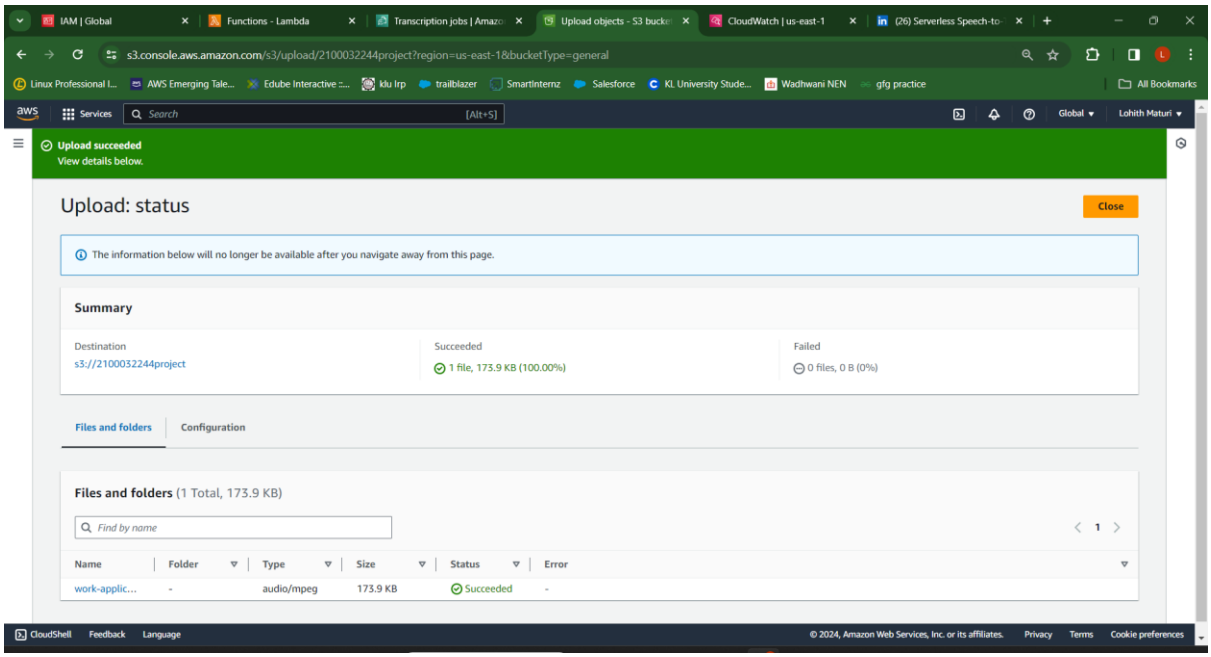
Go to cloudwatch logs and select log groups



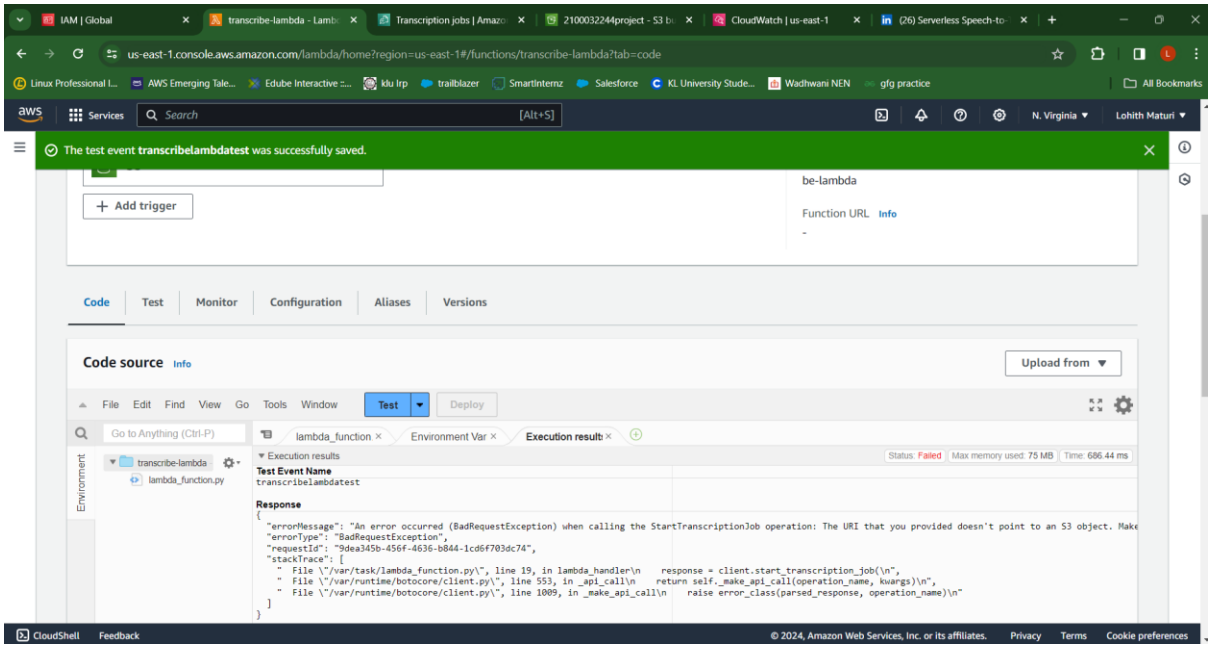
Click on transcribe-lambda and see the log streams



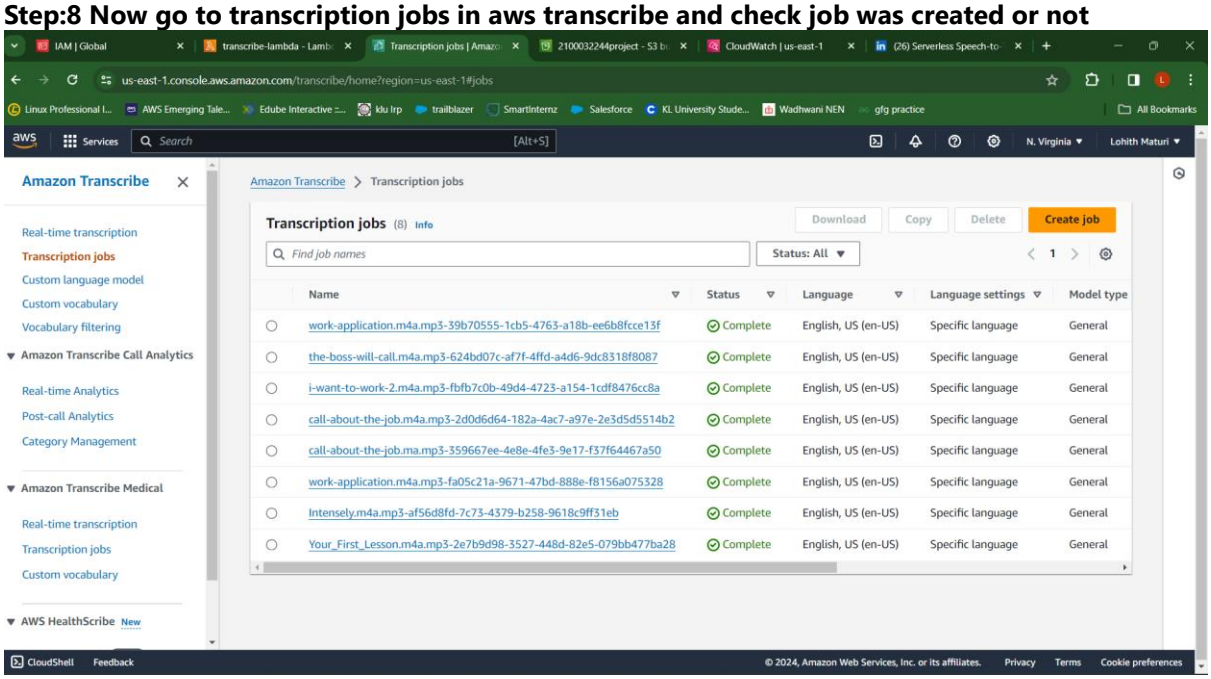
Step 6: Upload an audio file (in mp3 format) to your S3 bucket. Check the CloudWatch logs to confirm that the Lambda function was triggered and started a transcription job in AWS Transcribe.



Step:7: Once the transcription job is complete, you can view the transcribed text in the AWS Transcribe console.

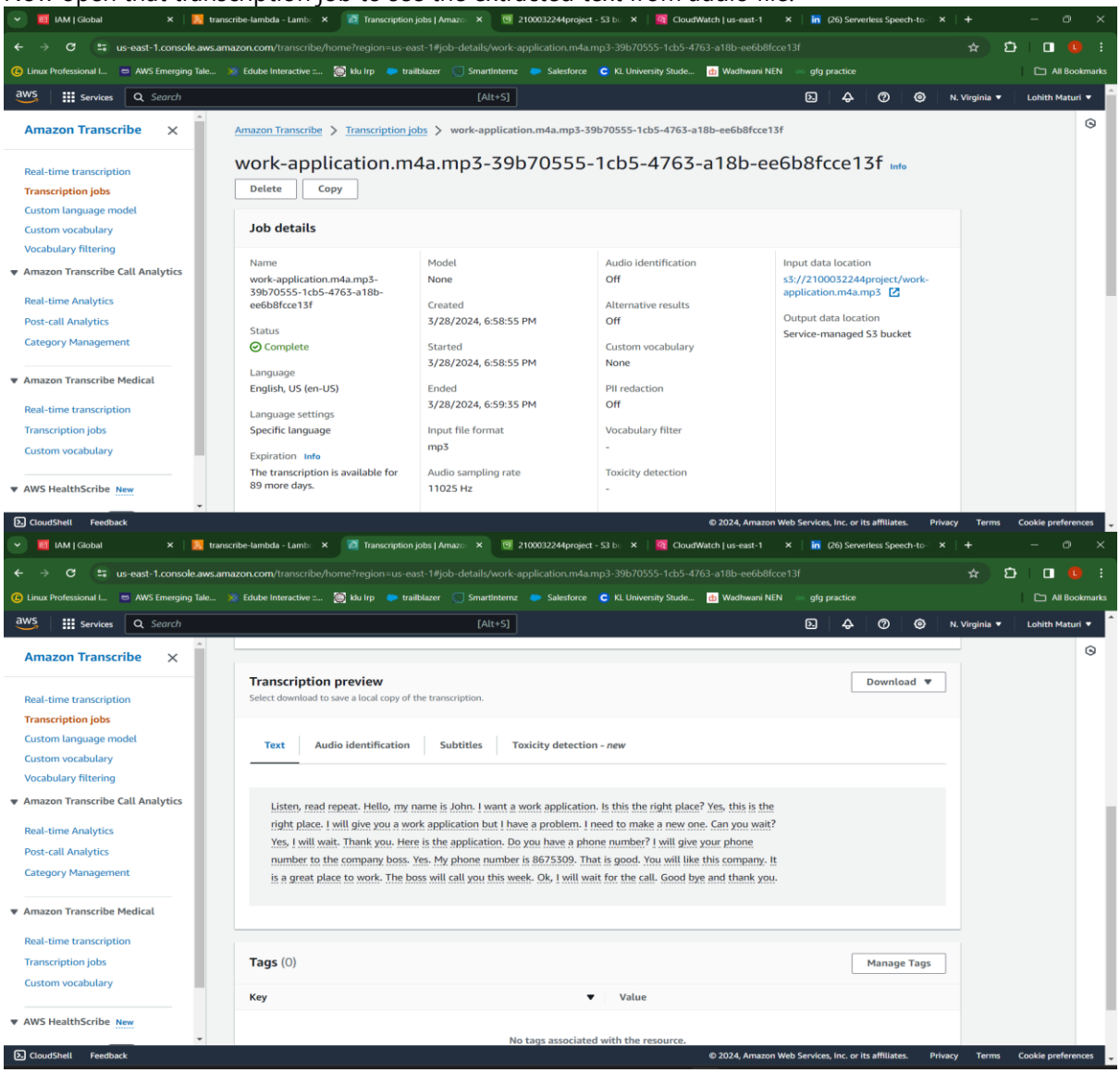


Step:8 Now go to transcription jobs in aws transcribe and check job was created or not



We uploaded the audio file called “work application” for that transcription job was created .

Now open that transcription job to see the extracted text from audio file.



In above photo the field called transcription preview contains the extracted text.

That's it! You have successfully set up a serverless Speech-to-Text experiment with AWS Transcribe and S3 Event Trigger using Lambda and CloudWatch.

GithubLink:

https://github.com/lohithmaturi/Cloud-Sereverless-Project

YouTubeLink:

https://youtu.be/3jKug-mxHBU?si=dxEryRomJPCjWNDZ

LinkedIn Article:

https://www.linkedin.com/pulse/aws-serverless-speech-to-text-transcribe-s3-event-trigger-maturi-hyrkc/?trackingId=tSVyUuhgReGHuf4lbLTxmw%3D%3D