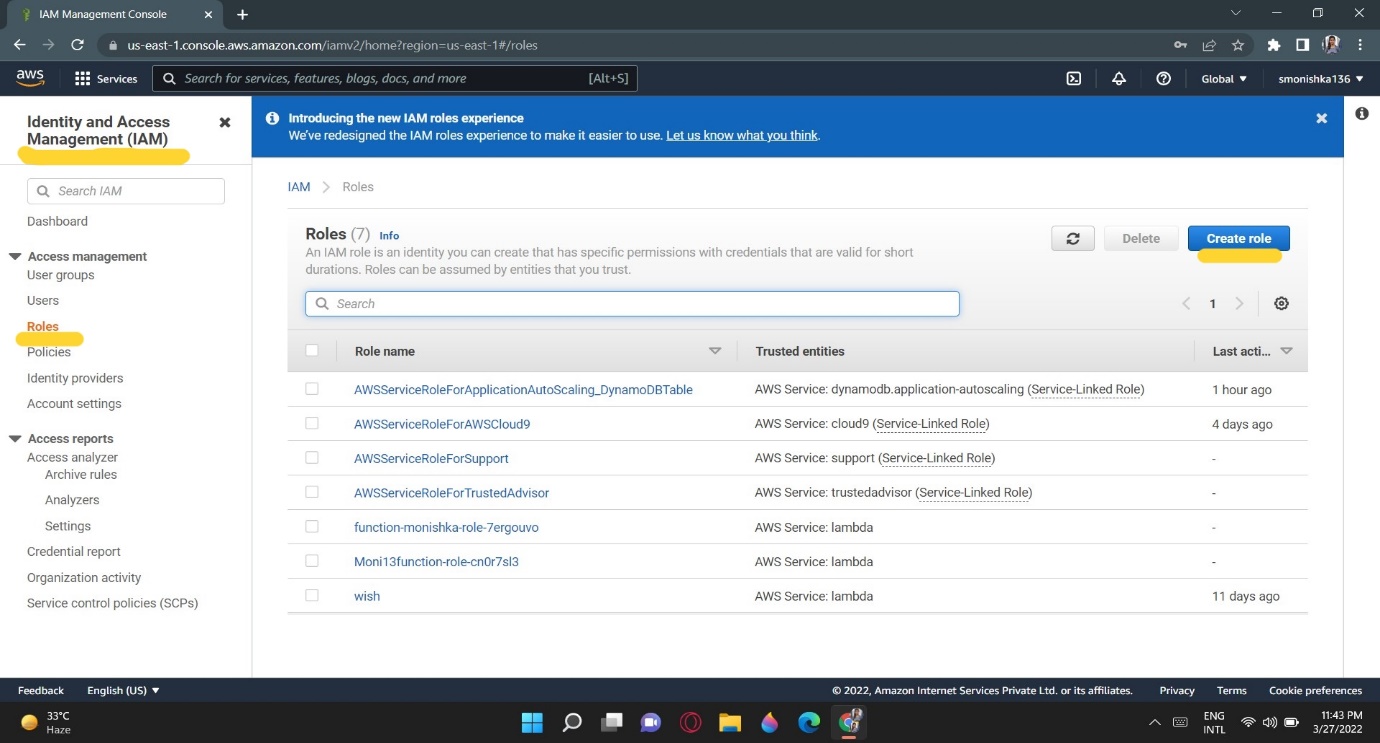
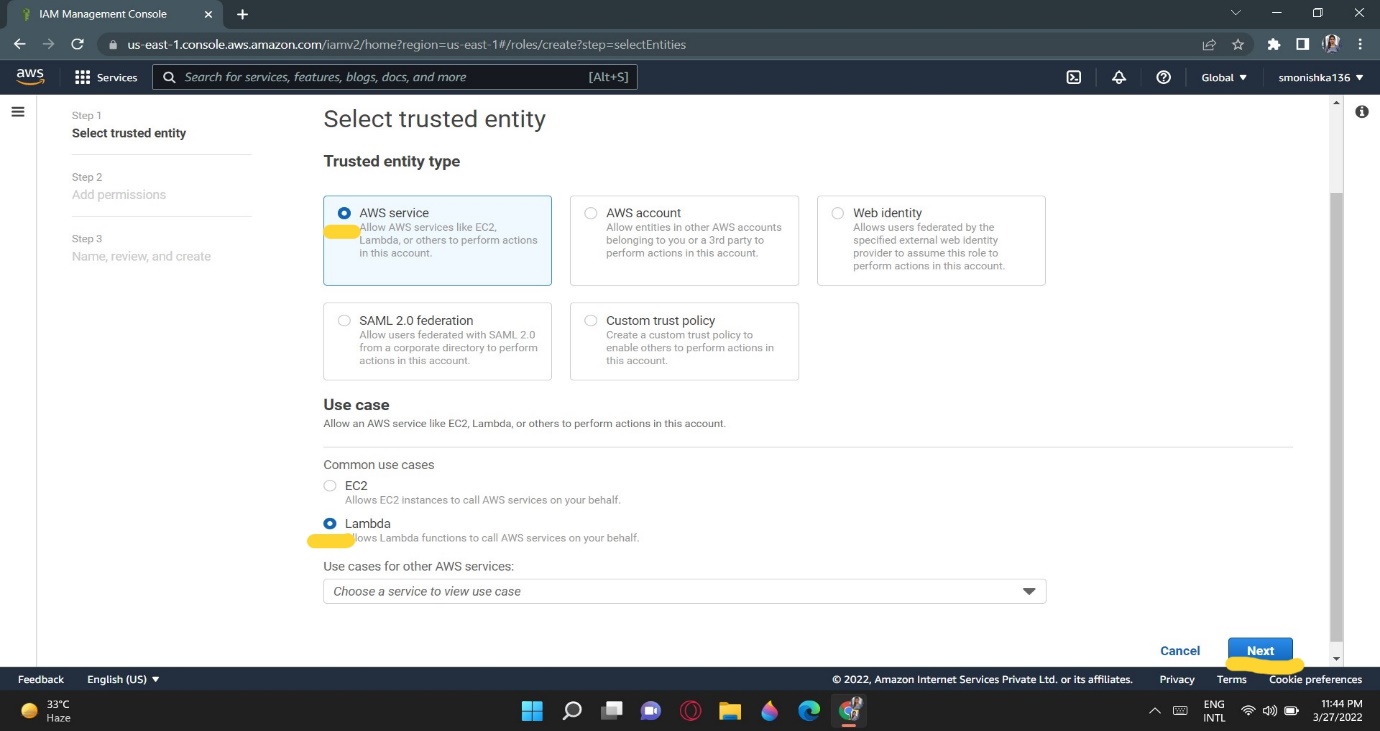
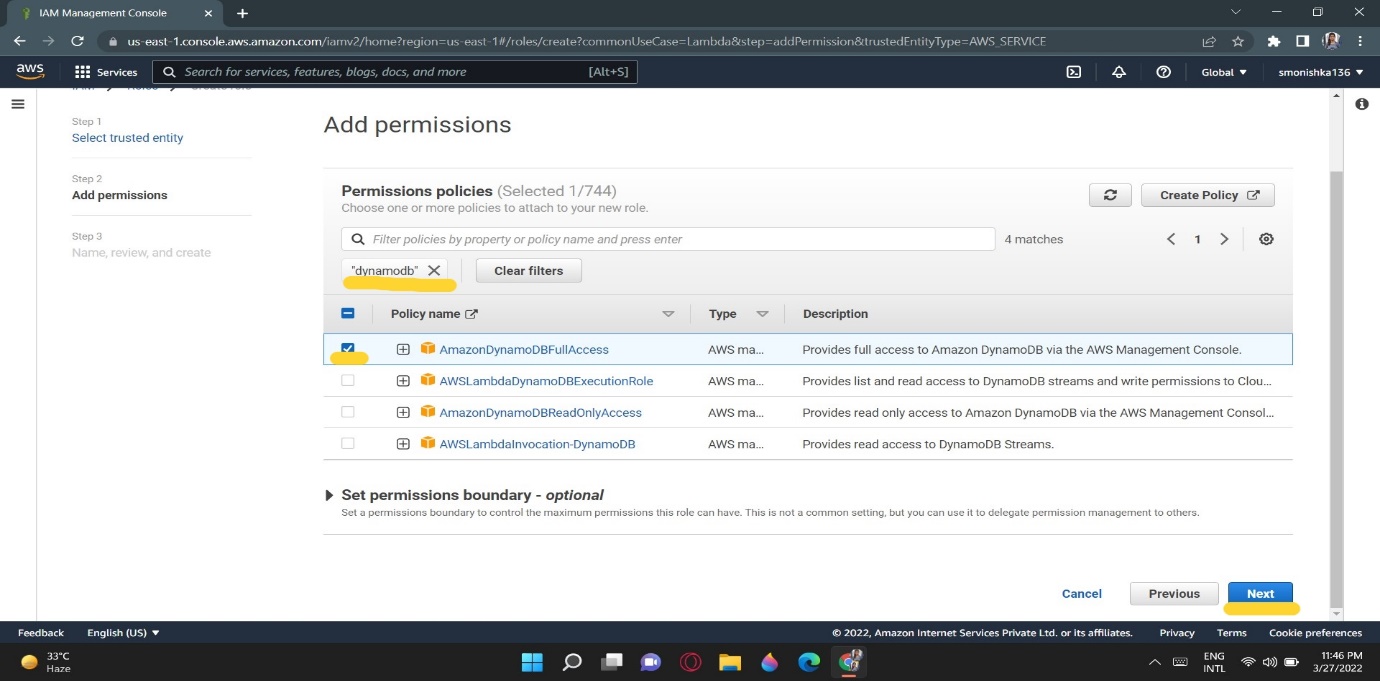
# AWS-Lambda-Trigger\_Automation

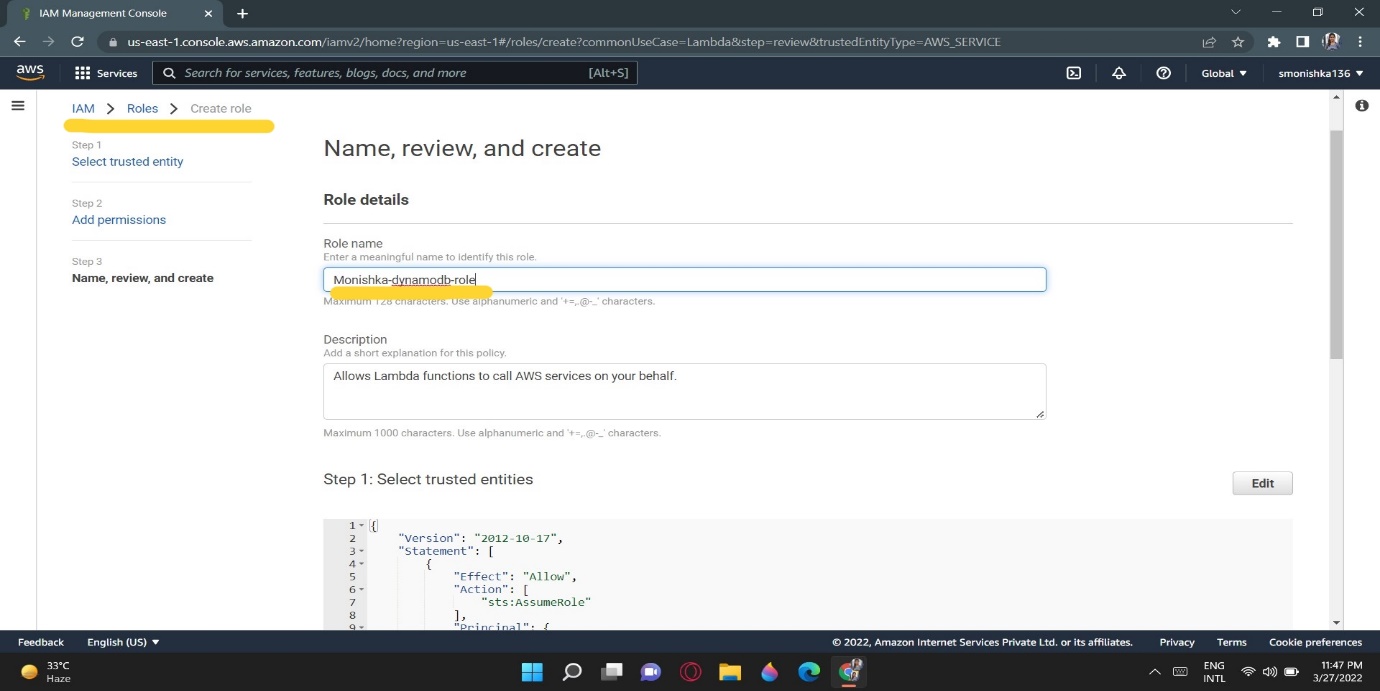
OBJECTIVE : S3 Trigger with Lambda and dynamoDB. Here, we will update the dynamo DB by fetching/triggering the new files added in the S3 bucket using the AWS lambda function which will we be complete automation.

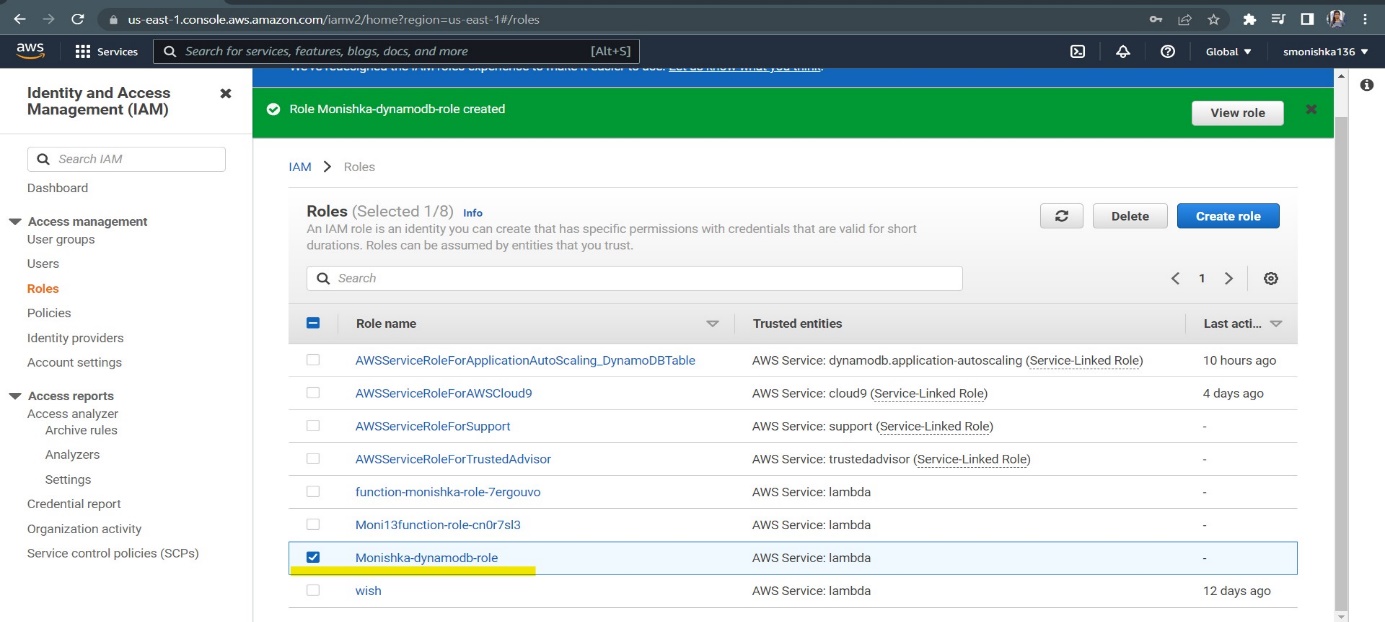
STEP -1 Create an IAM role (having specific permissions to access the AWS services).  
Go to Roles > create role > AWS services > lambda and select the policy. Here policy, we are attaching to our role is “AmazonDynamoDBFullAccess” and move a step ahead to complete further configurations. Add a role name, role name can be any, and click on create .Follow the below output.



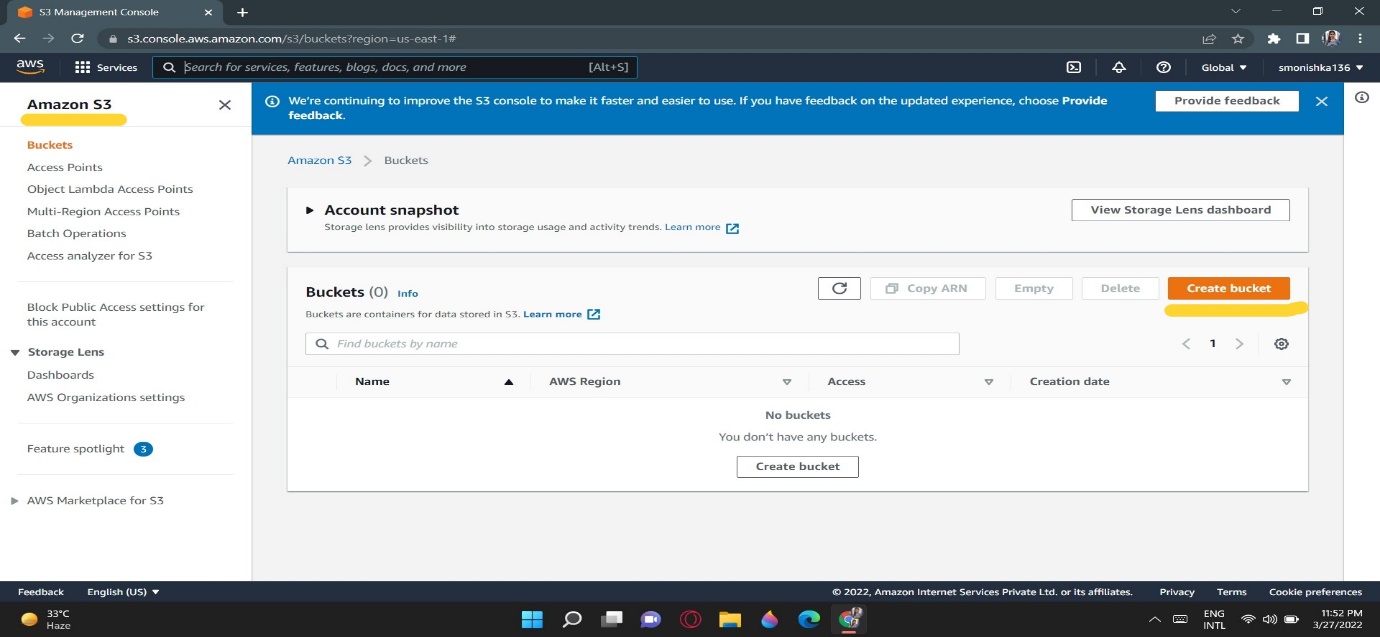


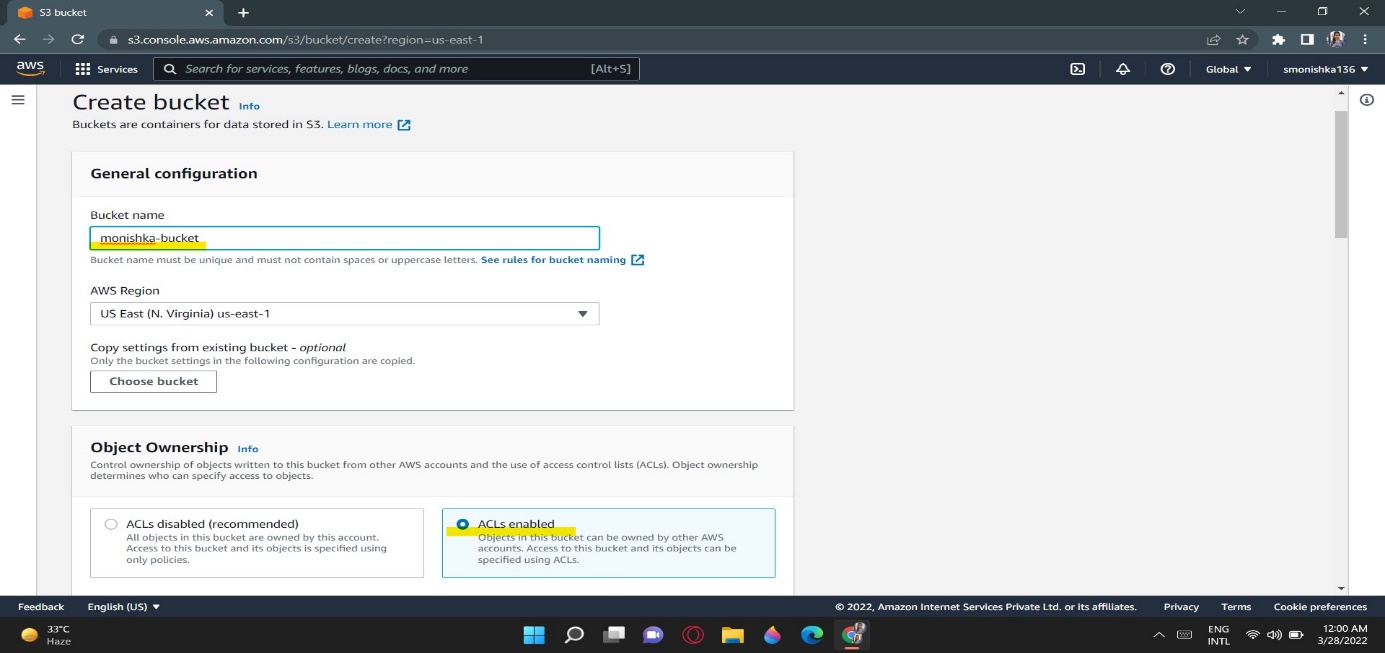


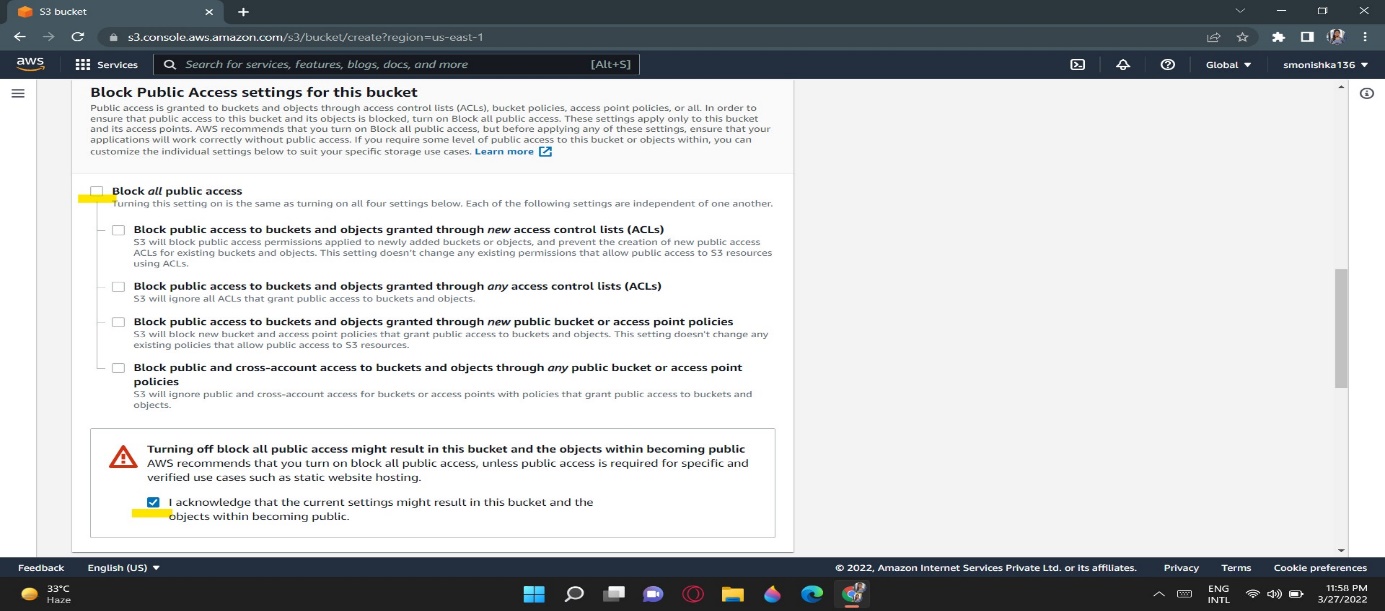




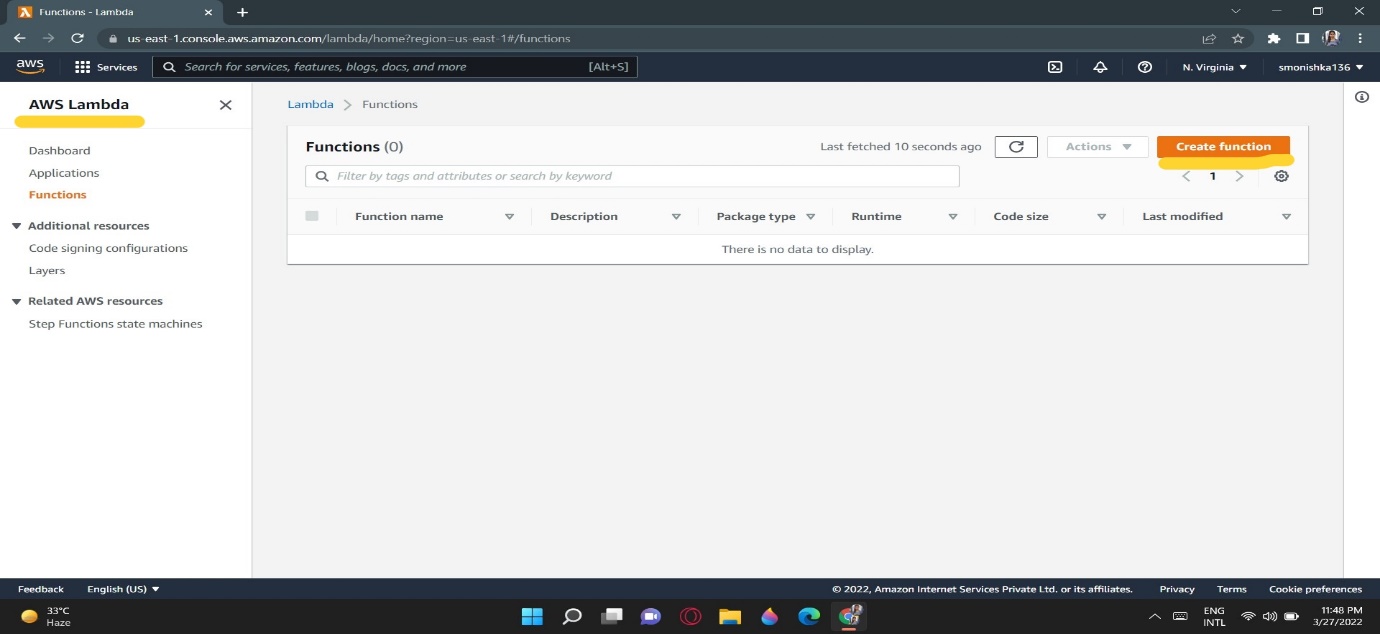
**Step-2 :** Create an S3 bucket to trigger from Lambda function. The name of bucket and region can be any for now. Allow all the public access as this is the learning phase.  
Finally, the bucket is created.

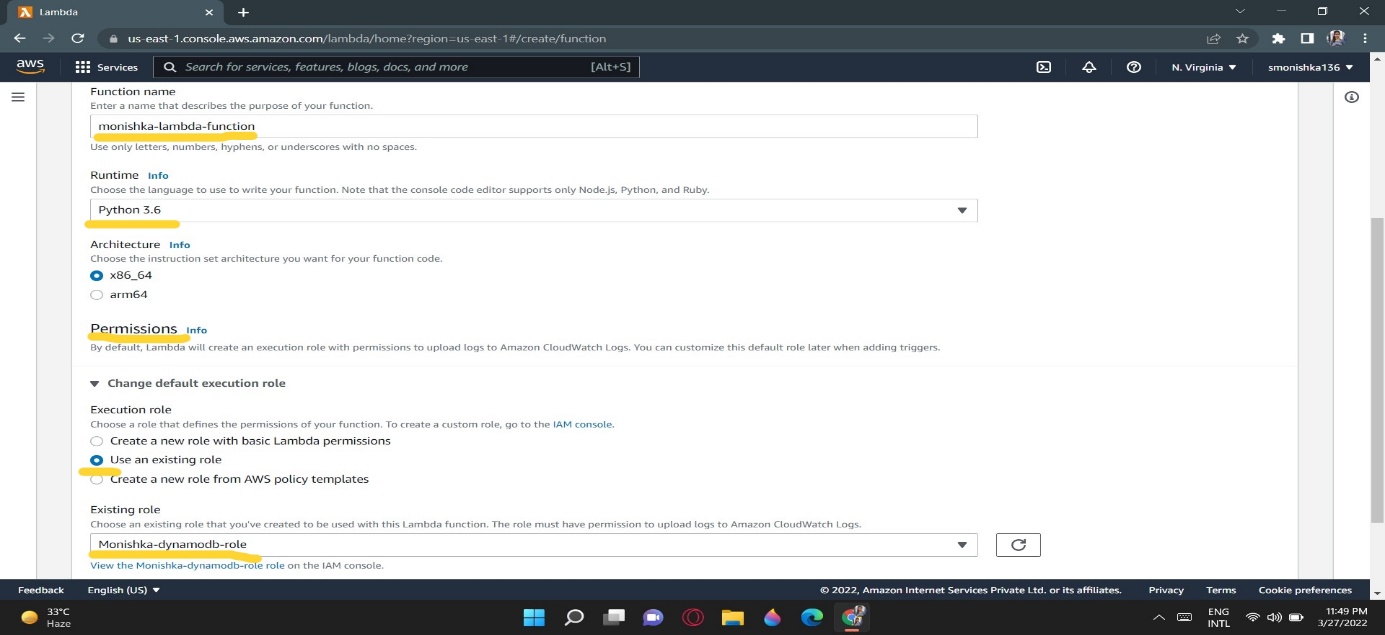


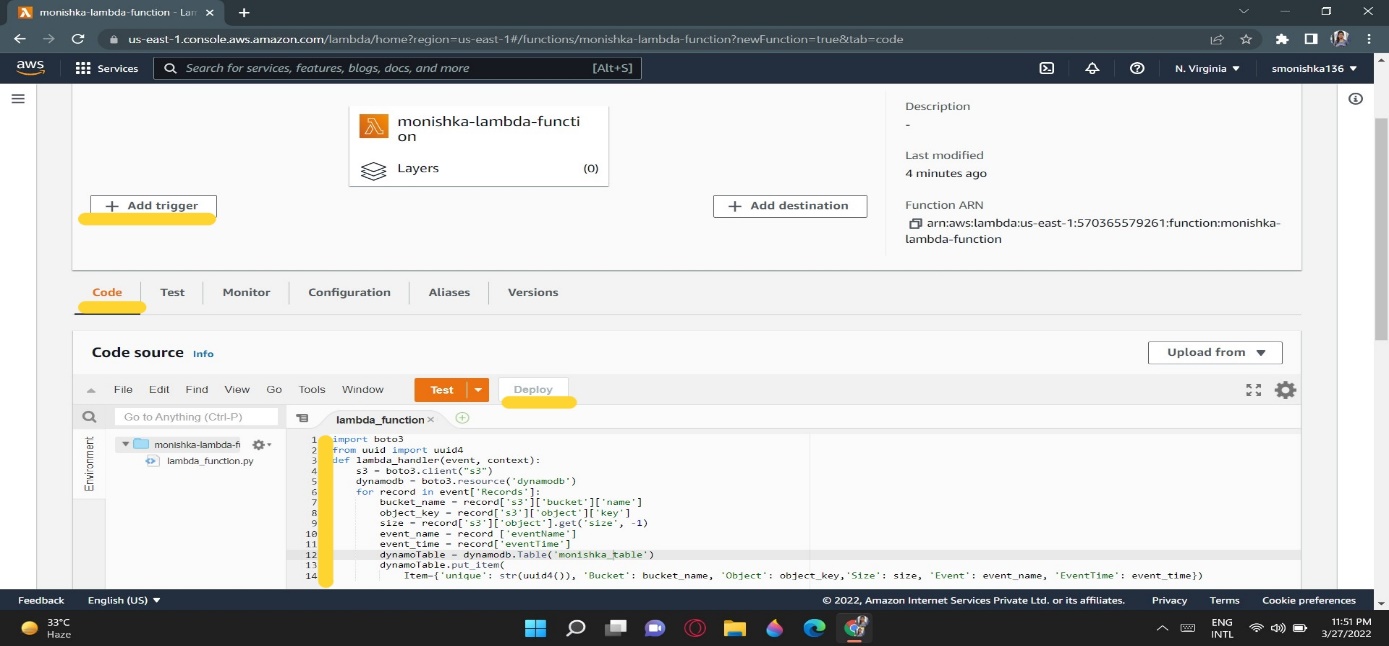




**Step-3 :** Create a Lambda function (this function will be used further to trigger the S3 bucket services).Here, we will integrate dynamo DB using the python 3.6 version.  
The role must be selected which was created in Step-1.







Inside the code source, a Python script is implemented to interact with Dynamo DB.

import boto3

from uuid import uuid4

def lambda\_handler(event, context):

s3 = boto3.client("s3")

dynamodb = boto3.resource('dynamodb')

for record in event['Records']:

bucket\_name = record['s3']['bucket']['name']

object\_key = record['s3']['object']['key']

size = record['s3']['object'].get('size', -1)

event\_name = record ['eventName']

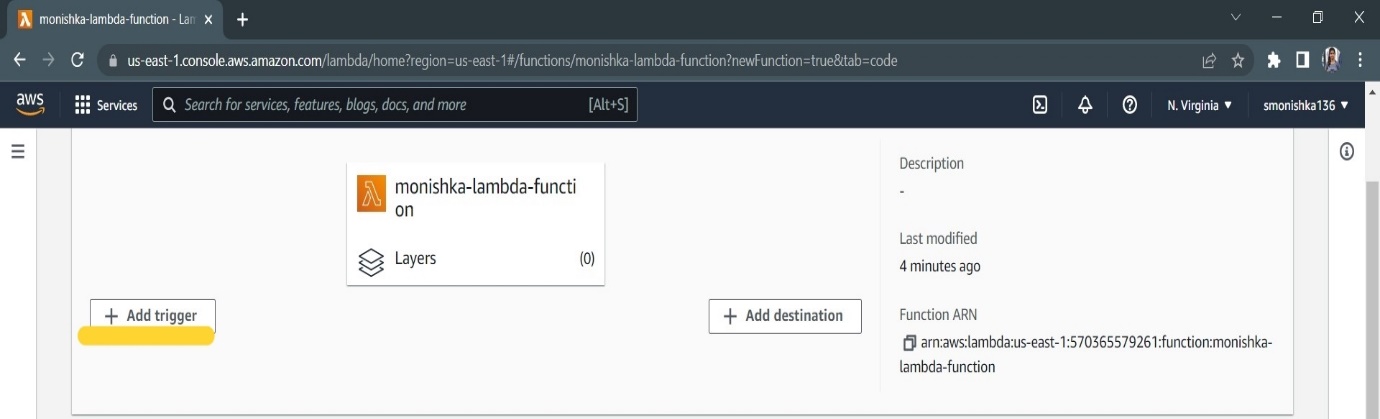
event\_time = record['eventTime']

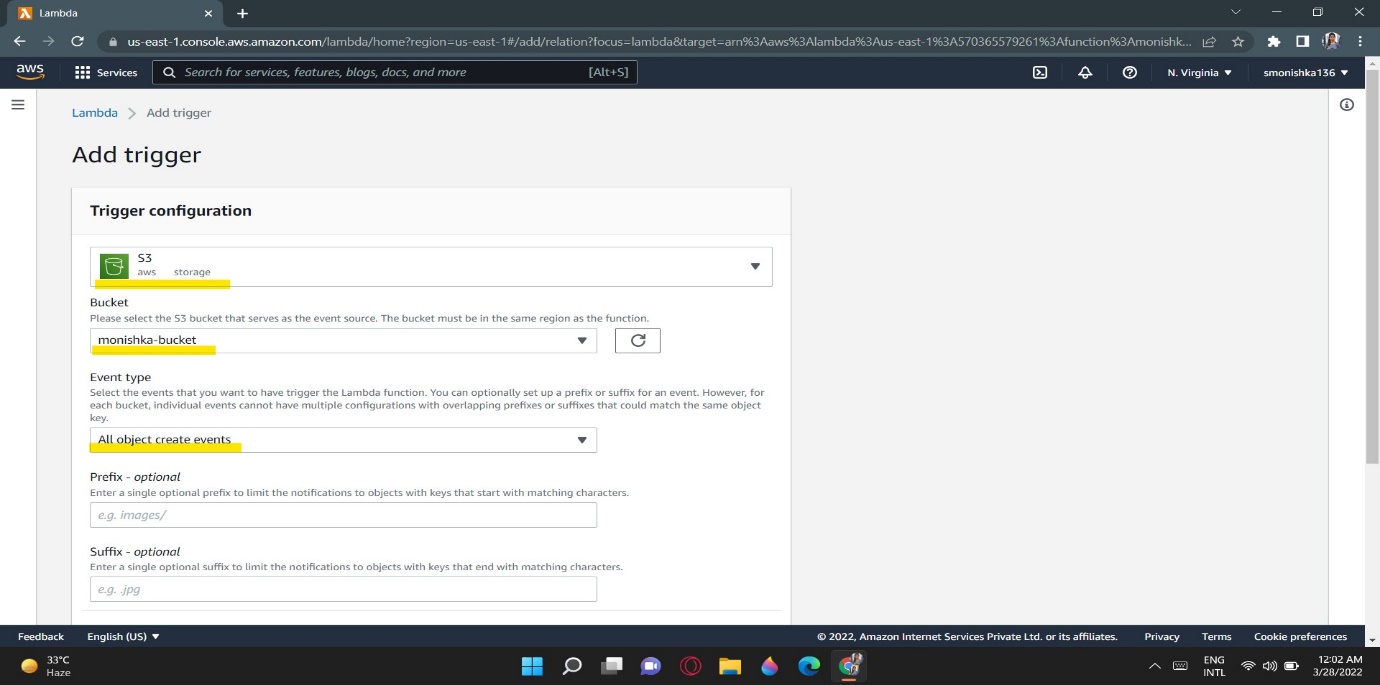
dynamoTable = dynamodb.Table('monishka\_table')

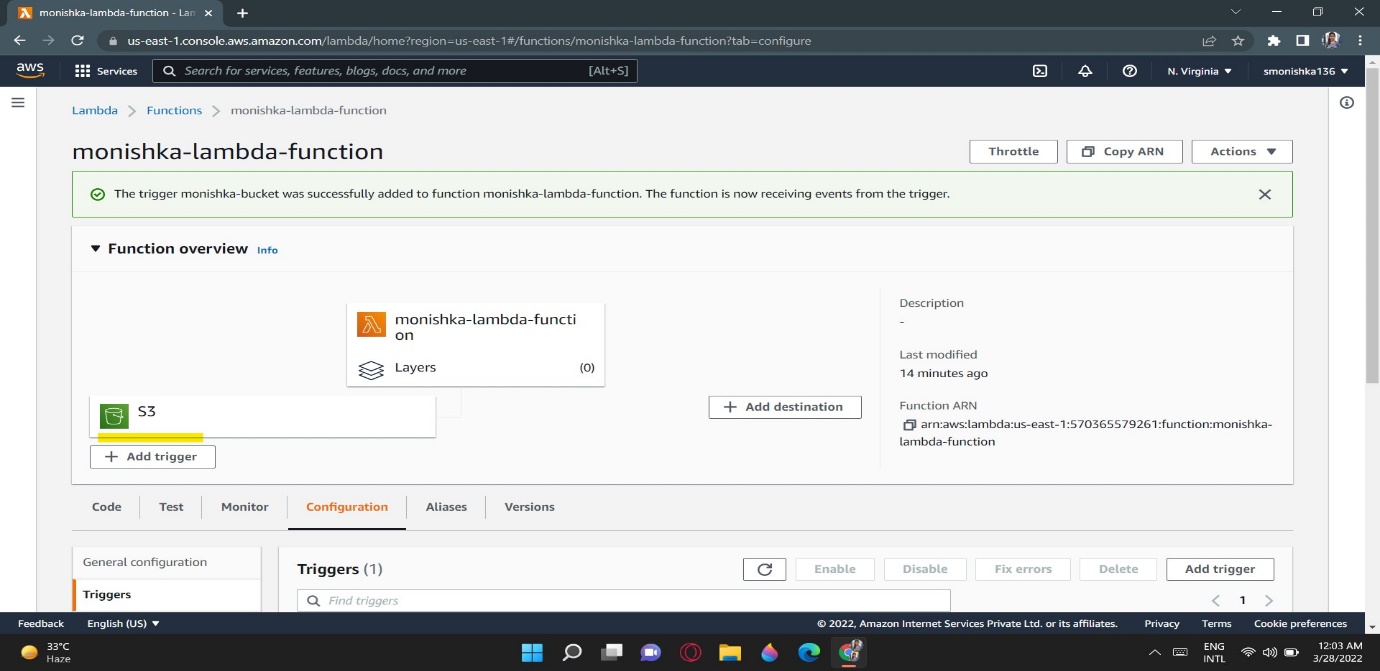
dynamoTable.put\_item(

Item={'unique': str(uuid4()), 'Bucket': bucket\_name, 'Object': object\_key,'Size': size, 'Event': event\_name, 'EventTime': event\_time})

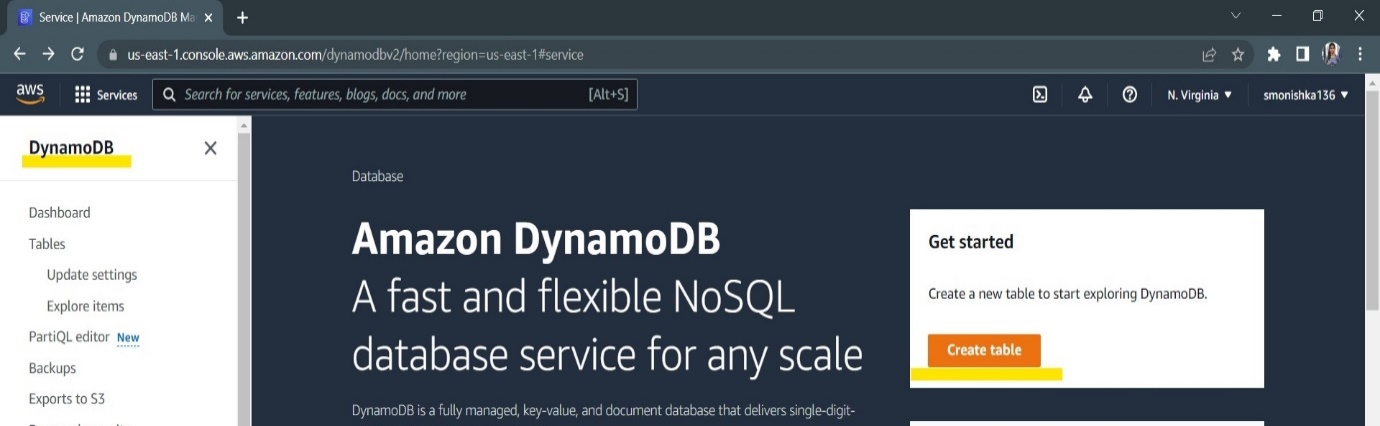
**Step-4 :** Now we will set the S3 bucket to trigger it by using the lambda function.  
Go to the “add trigger” and add the S3 bucket which we created earlier. Event type is basically a type of functionality we want like put delete and so on. Here we’ll select “All objects create events”. Prefix and suffix are used if we want to add a specific file of any extension.S3 bucket is ready to trigger using Lambda function.

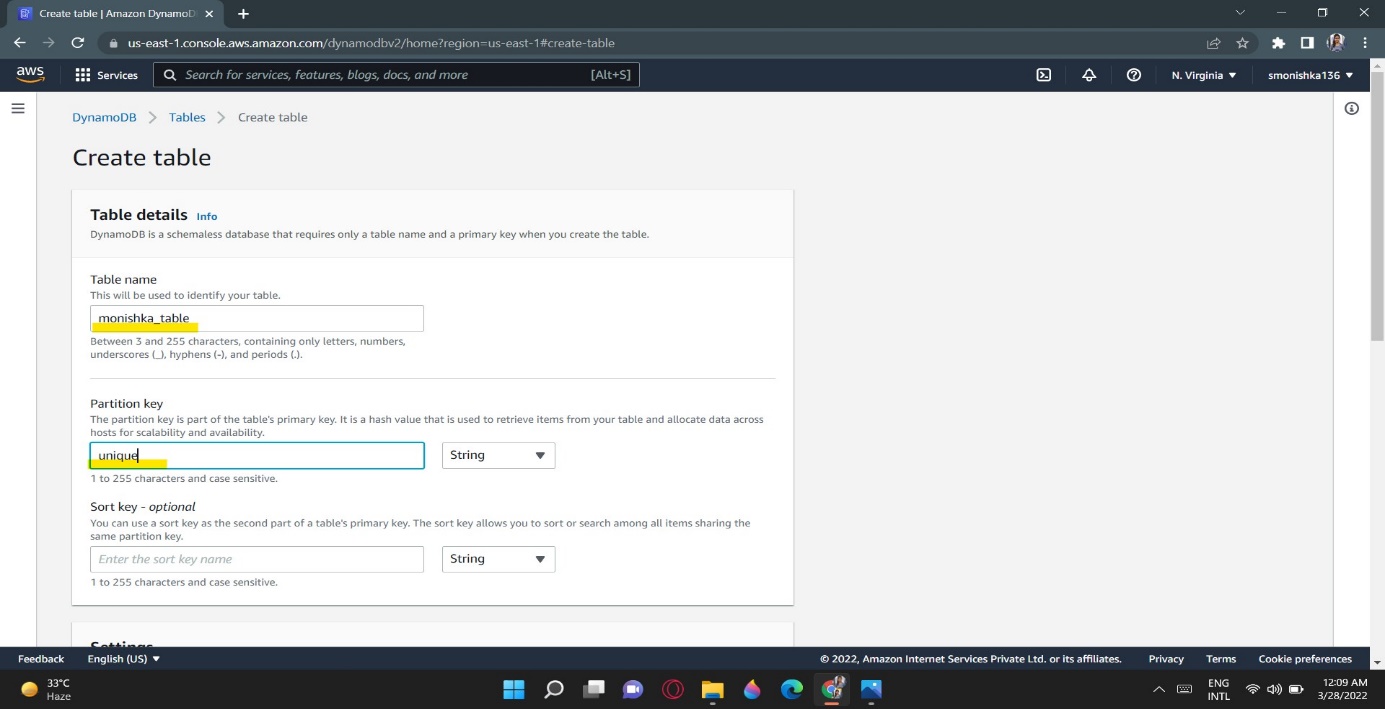


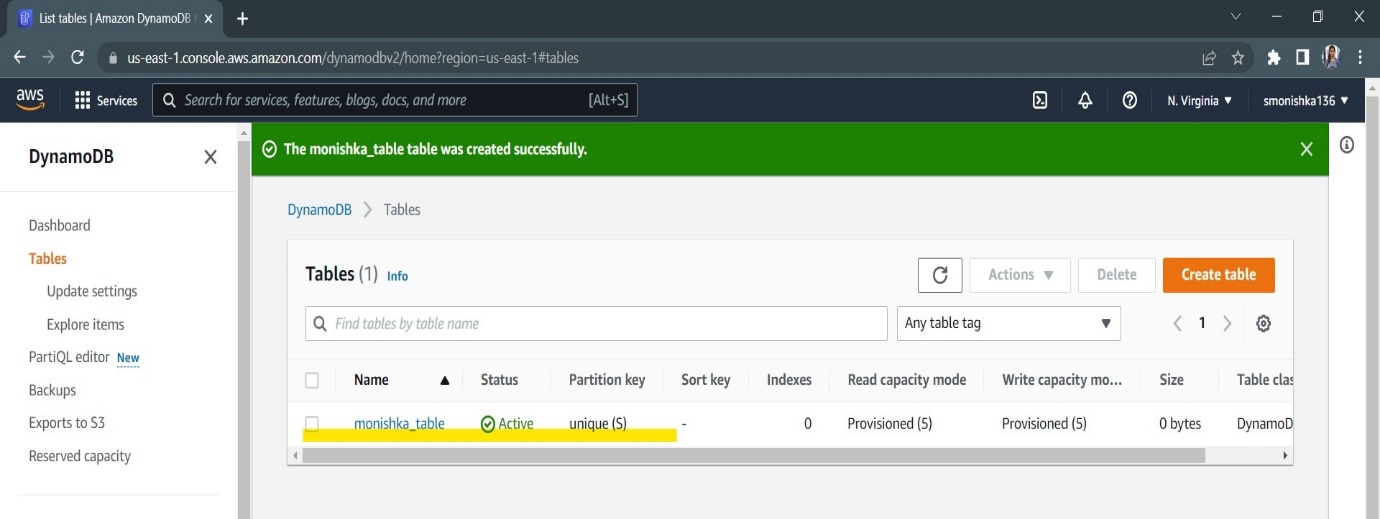




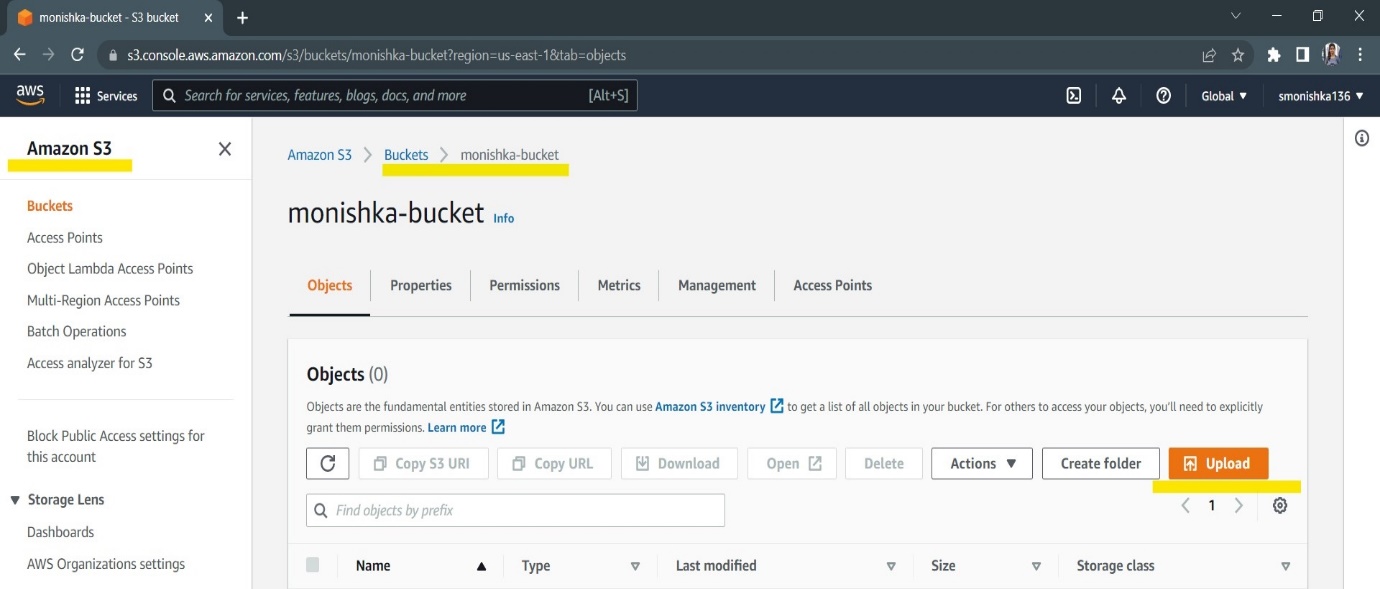
**Step-5 :** DynamoDB will be used to store the input and outputs of the data from the S3 bucket. Go to databases services and select Dynamo DB. Here we will take the name of the table i.e “newtable” and partition key “unique”. This value can be changed inside your code we used to integrate the dynamo DB. At first, you will see dynamo DB items list are empty.

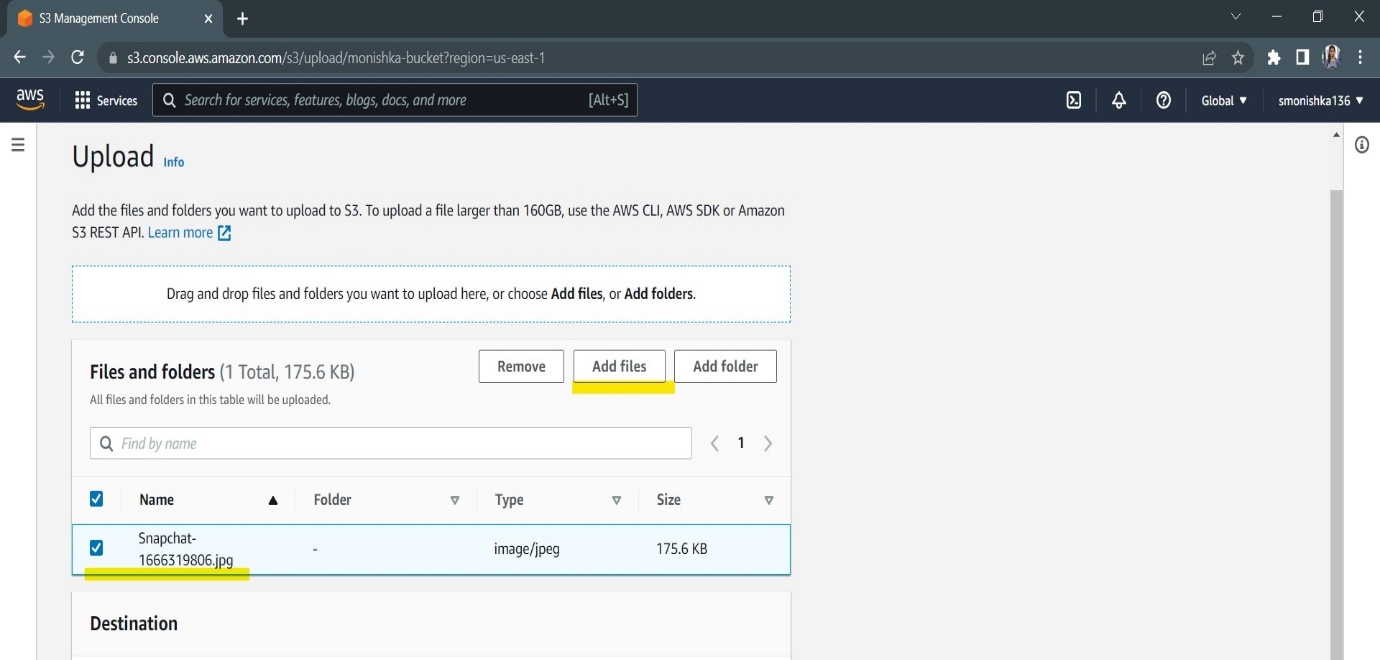


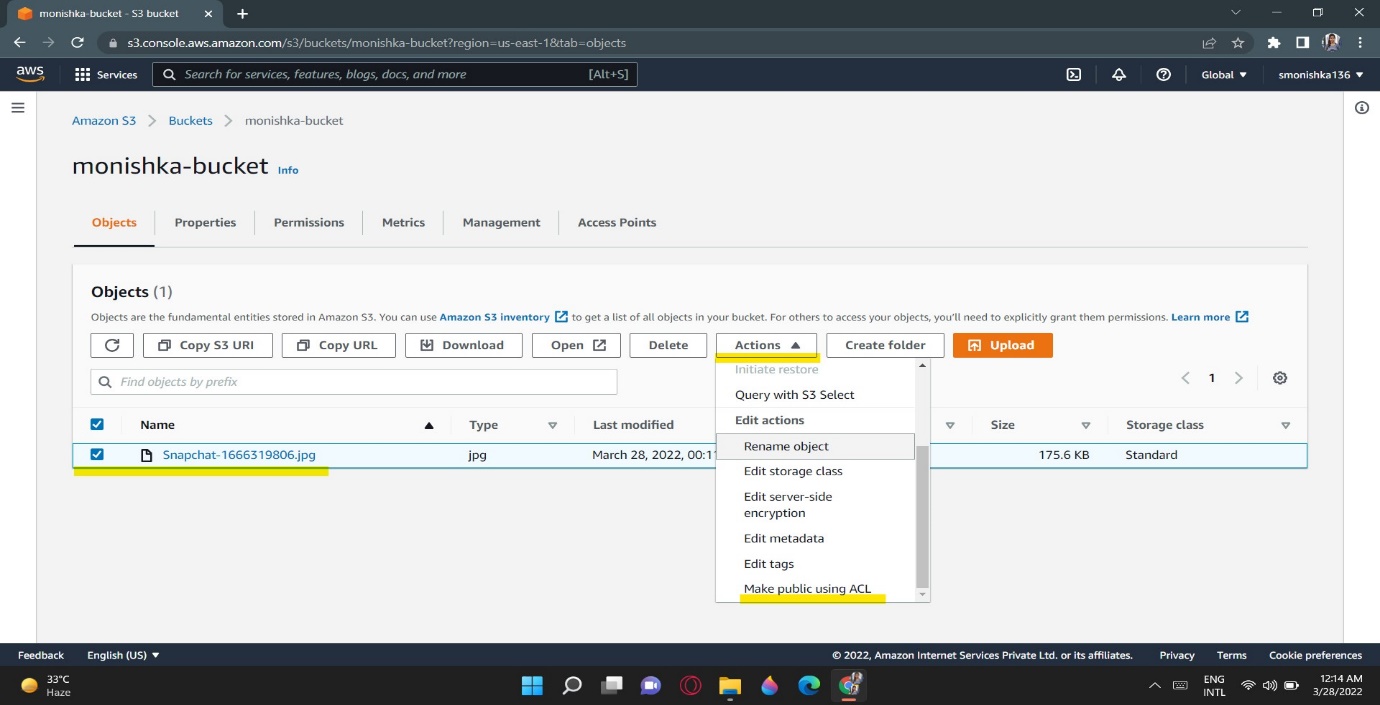




**Step-6 :** Go to the S3 bucket try adding the file/folder inside your bucket manually, we will see lambda function triggers and update the dynamo DB database. First, Below output shows File has been updated in S3 bucket .

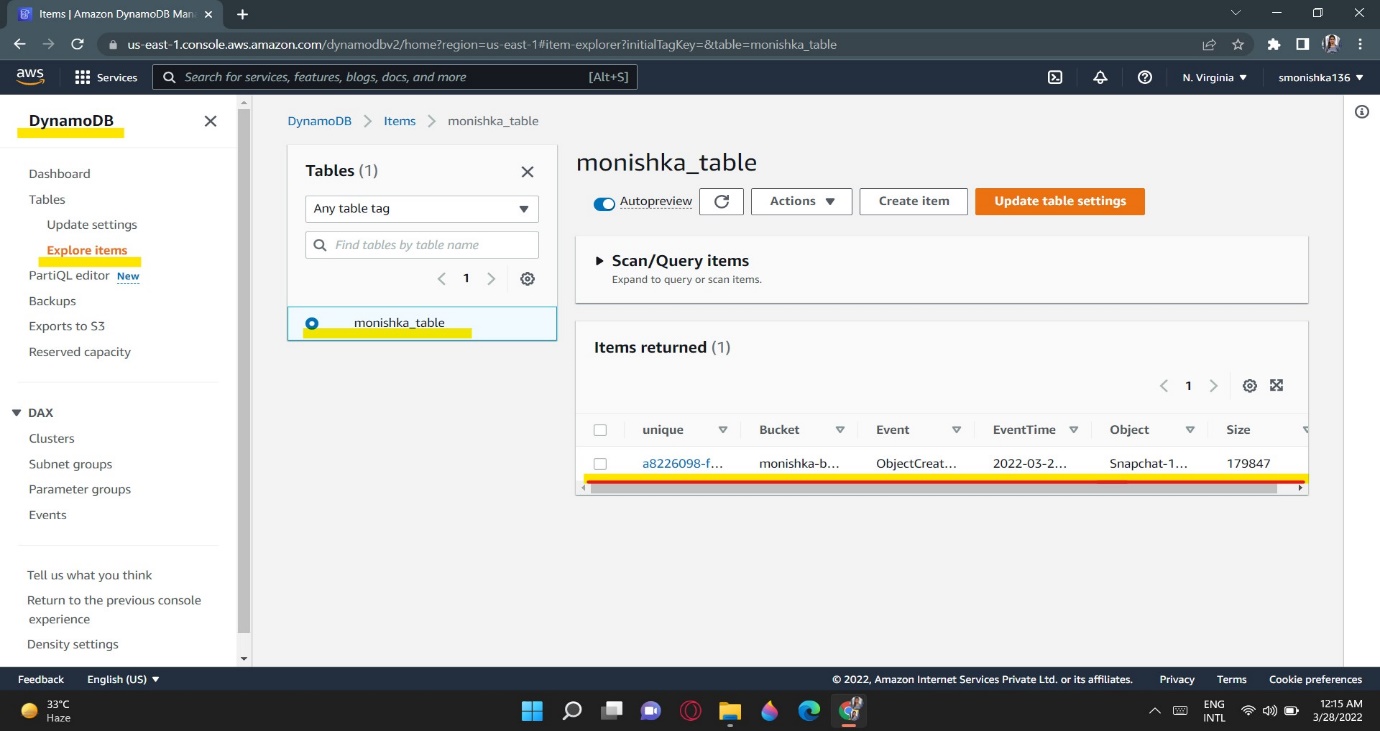






**Step-6 : Now the final step is we can see the S3 items which I Uploaded we see in Explore items then select table then we see the Items.**

**Hence also say -** we will see lambda function triggers and update the dynamo DB database. First, Below output shows File has been updated in S3 bucket .



Thankyou…