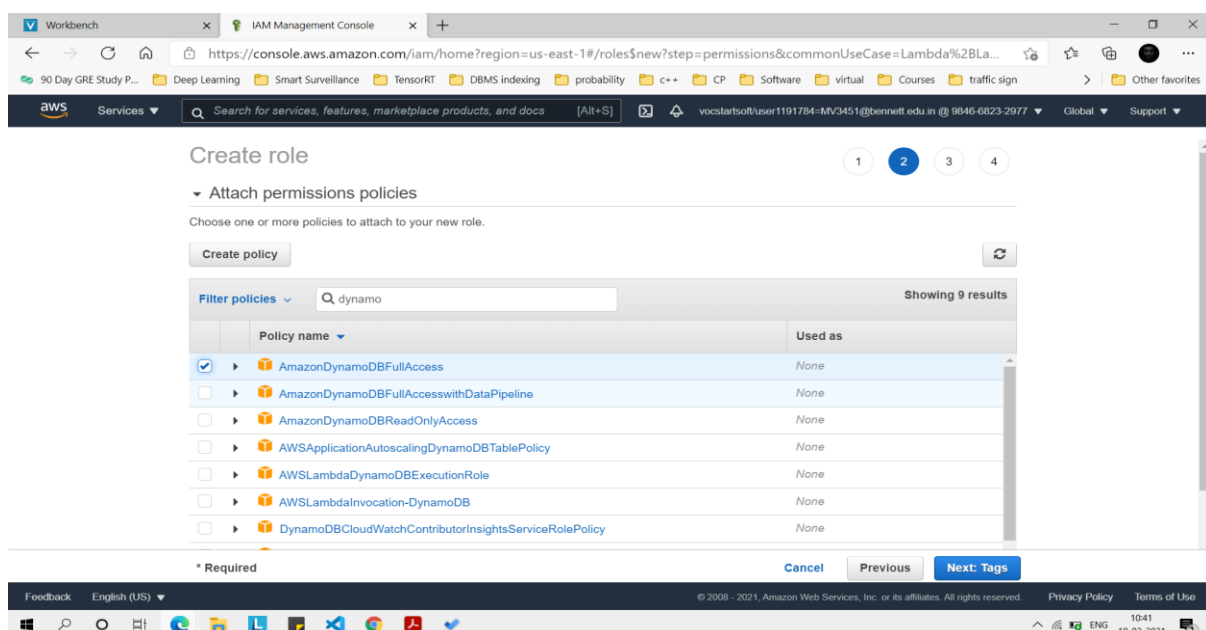
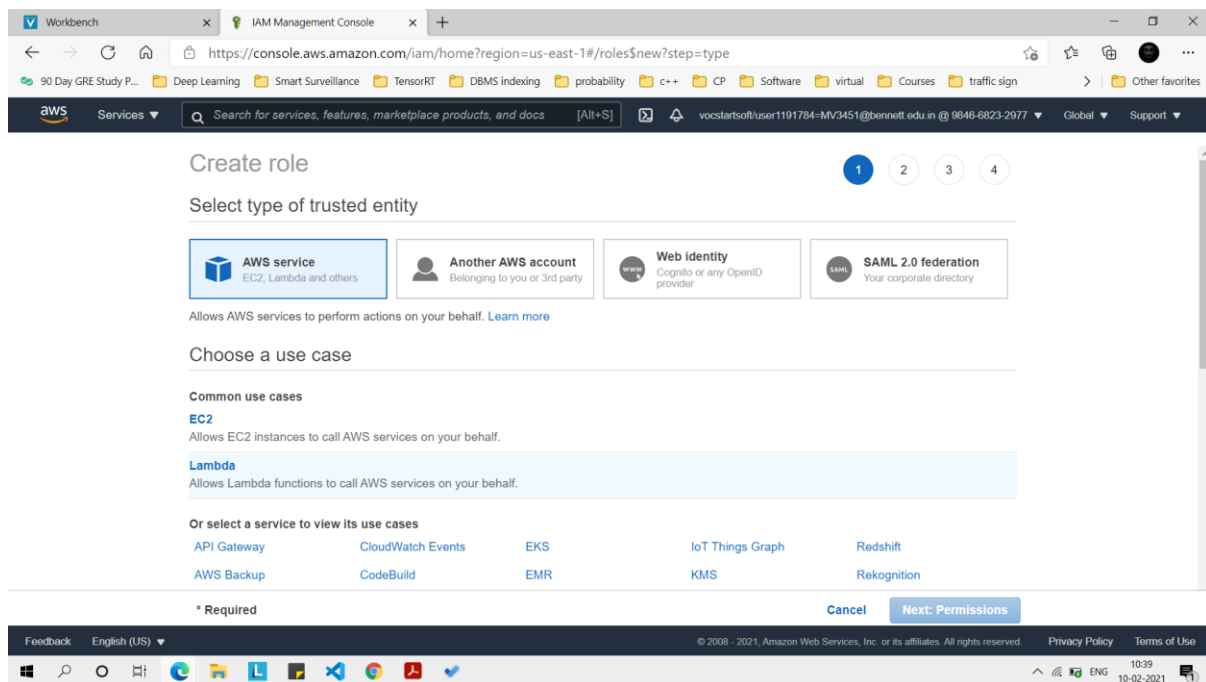


## Lab Assignment – 04

Create a Lambda function to trigger S3 bucket objects and store the object details in the DynamoDB Table.

Task – 1: Create the role of Full Dynamo DB service using IAM service.



**Create role**

**Review**

Provide the required information below and review this role before you create it.

**Role name\***

Use alphanumeric and '+', '@', '-' characters. Maximum 64 characters.

**Role description**

Maximum 1000 characters. Use alphanumeric and '+', '@', '-' characters.

**Trusted entities** AWS service: lambda.amazonaws.com

**Policies** AmazonDynamoDBFullAccess [View](#)

**Permissions boundary** Permissions boundary is not set

*No tags were added.*

\* Required

[Cancel](#) [Previous](#) [Create role](#)

Feedback English (US) © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

## Task – 2: Create Lambda function and configure it with S3 bucket and DynamoDB database.

**Author from scratch** ☒ Start with a simple Hello World example.

**Use a blueprint** ☐ Build a Lambda application from sample code and configuration presets for common use cases.

**Container image** ☐ Select a container image to deploy for your function.

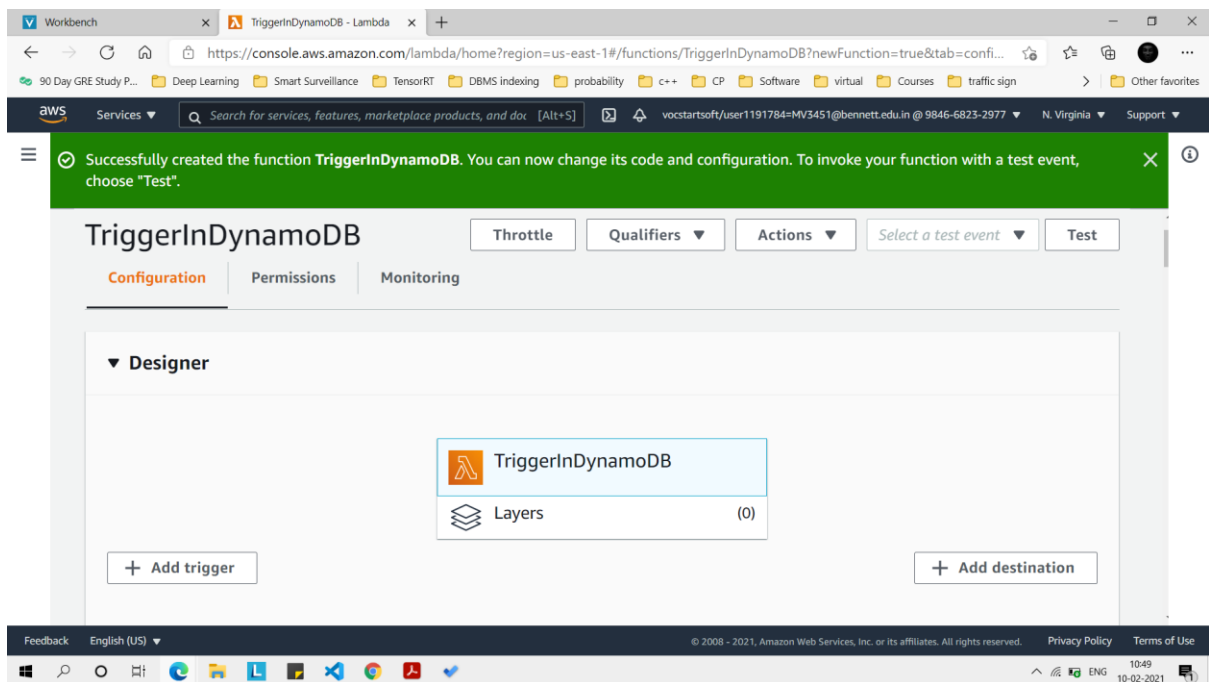
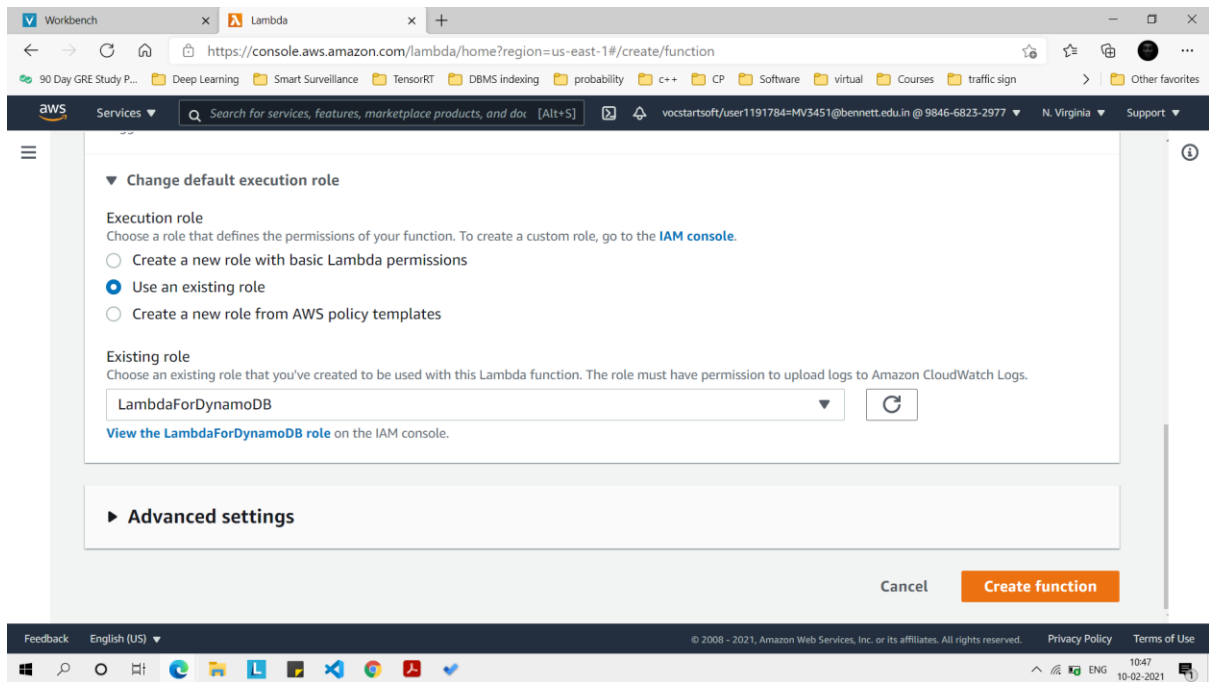
**Browse serverless app repository** ☐ Deploy a sample Lambda application from the AWS Serverless Application Repository.

**Basic information**

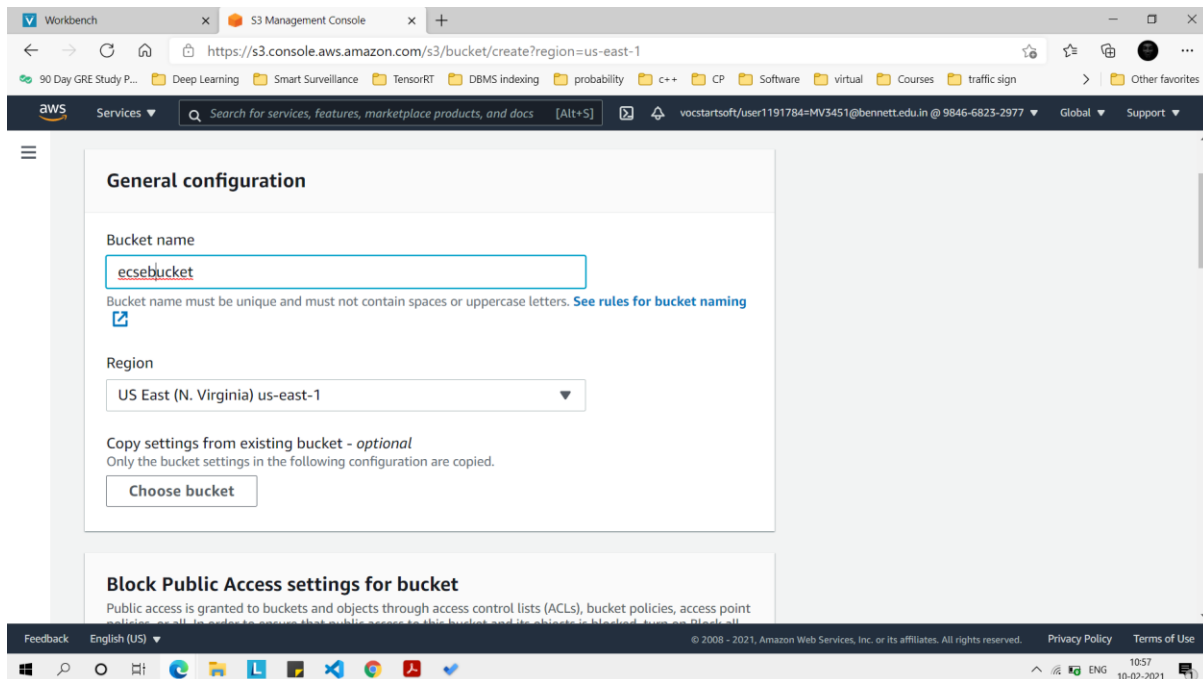
**Function name**  
Enter a name that describes the purpose of your function.  
  
Use only letters, numbers, hyphens, or underscores with no spaces.

**Runtime** [Info](#)  
Choose the language to use to write your function.

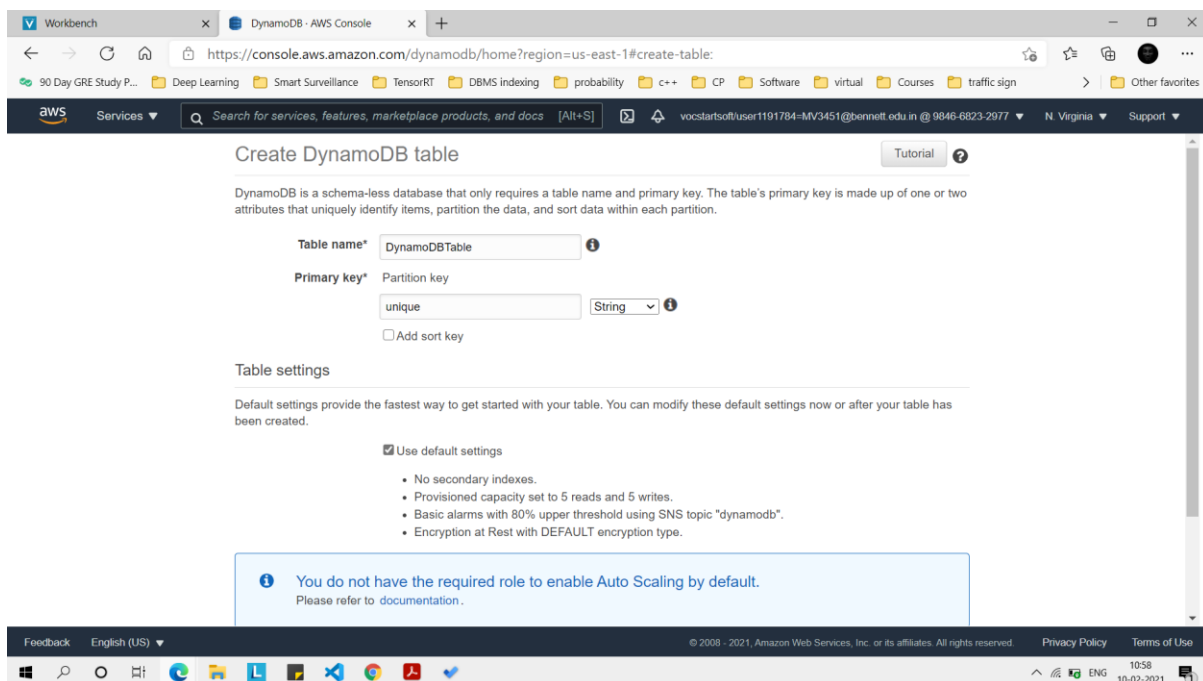
Feedback English (US) © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



## Create S3 Bucket:



## Task – 3: Create a table in DynamoDB and check the details of the items.



## Task – 4: Add a trigger for uploading objects in S3 bucket.

The screenshot shows the AWS Lambda console for the function **TriggerInDynamoDB**. The **Function code** tab is selected, displaying a Python script using `boto3` to interact with S3 and DynamoDB. The script processes an event from S3 and updates a DynamoDB table with the object's details.

```

1 import boto3
2 from uuid import uuid4
3 def lambda_handler(event, context):
4     s3 = boto3.client("s3")
5     dynamodb = boto3.resource('dynamodb')
6     for record in event['Records']:
7         bucket_name = record['s3']['bucket']['name']
8         object_key = record['s3']['object']['key']
9         size = record['s3']['object'].get('size', -1)
10        event_name = record['eventName']
11        event_time = record['eventTime']
12        dynamoTable = dynamodb.Table('DynamoDBTable')
13        dynamoTable.put_item(
14            Item={'unique': str(uuid4()), 'Bucket': bucket_name, 'Object': object_key, 'Size': size, 'Event': event_name, 'EventTime': event_time})
15

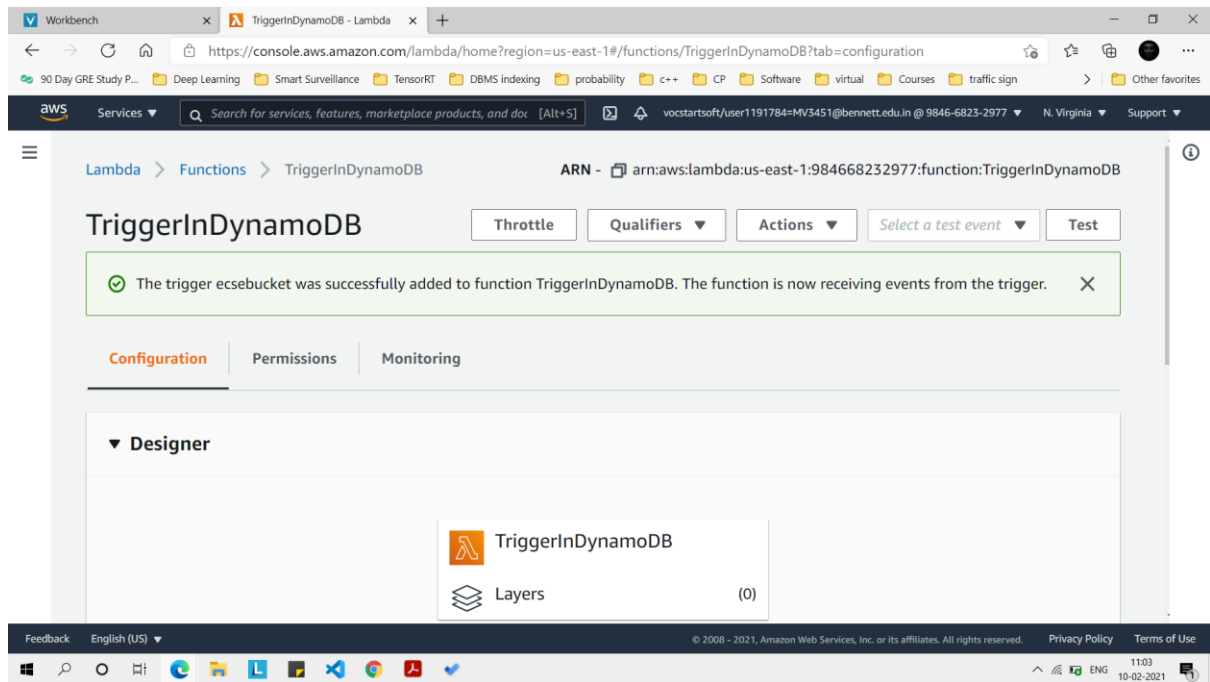
```

The console interface includes tabs for **Function code** and **Info**, and buttons for **Throttle**, **Qualifiers**, **Actions**, **Select a test event**, and **Test**. A **Deploy** button with a green **Changes deployed** status is also visible.

The screenshot shows the **Add trigger** configuration page in the AWS Lambda console. The **Trigger configuration** section is active, showing the following settings:

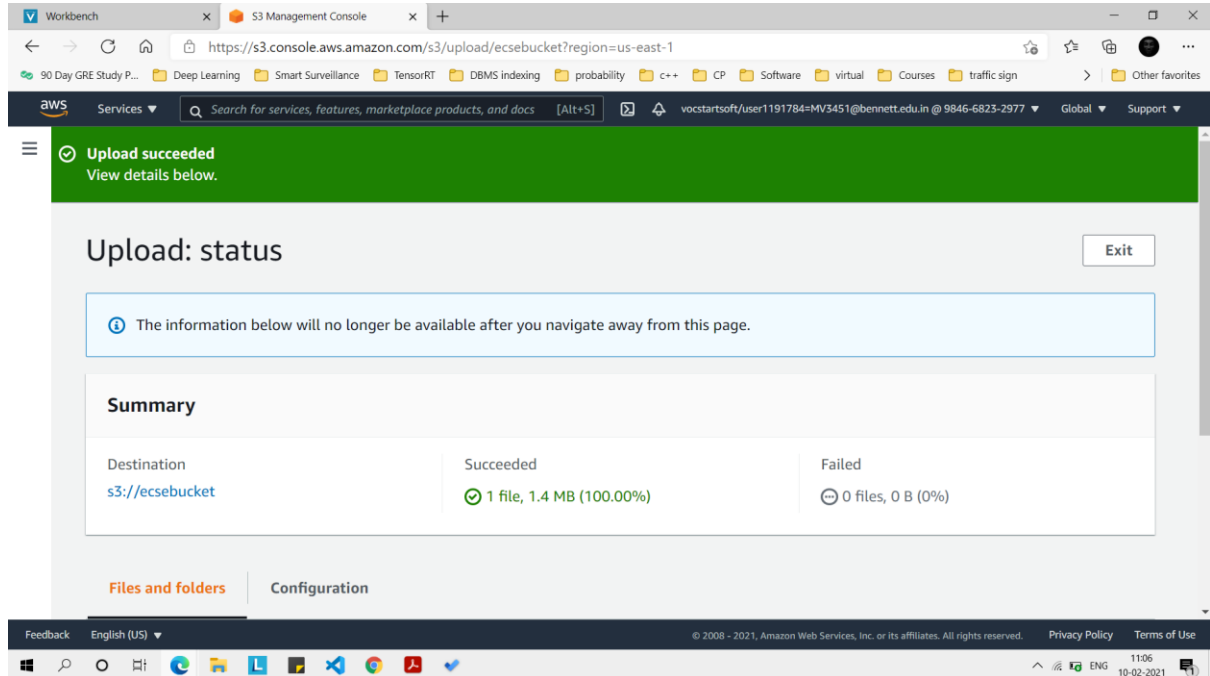
- Service:** S3 (aws storage)
- Bucket:** ecsebucket
- Event type:** All object create events

The **Bucket** section includes a note: "Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function." The **Event type** section includes a note: "Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key."



Task – 5: Check the proper working of the Lambda function.

Upload some files in S3 bucket:



We can observe the file upload details are stored in the DynamoDB Table indicating successful working of the Trigger.

