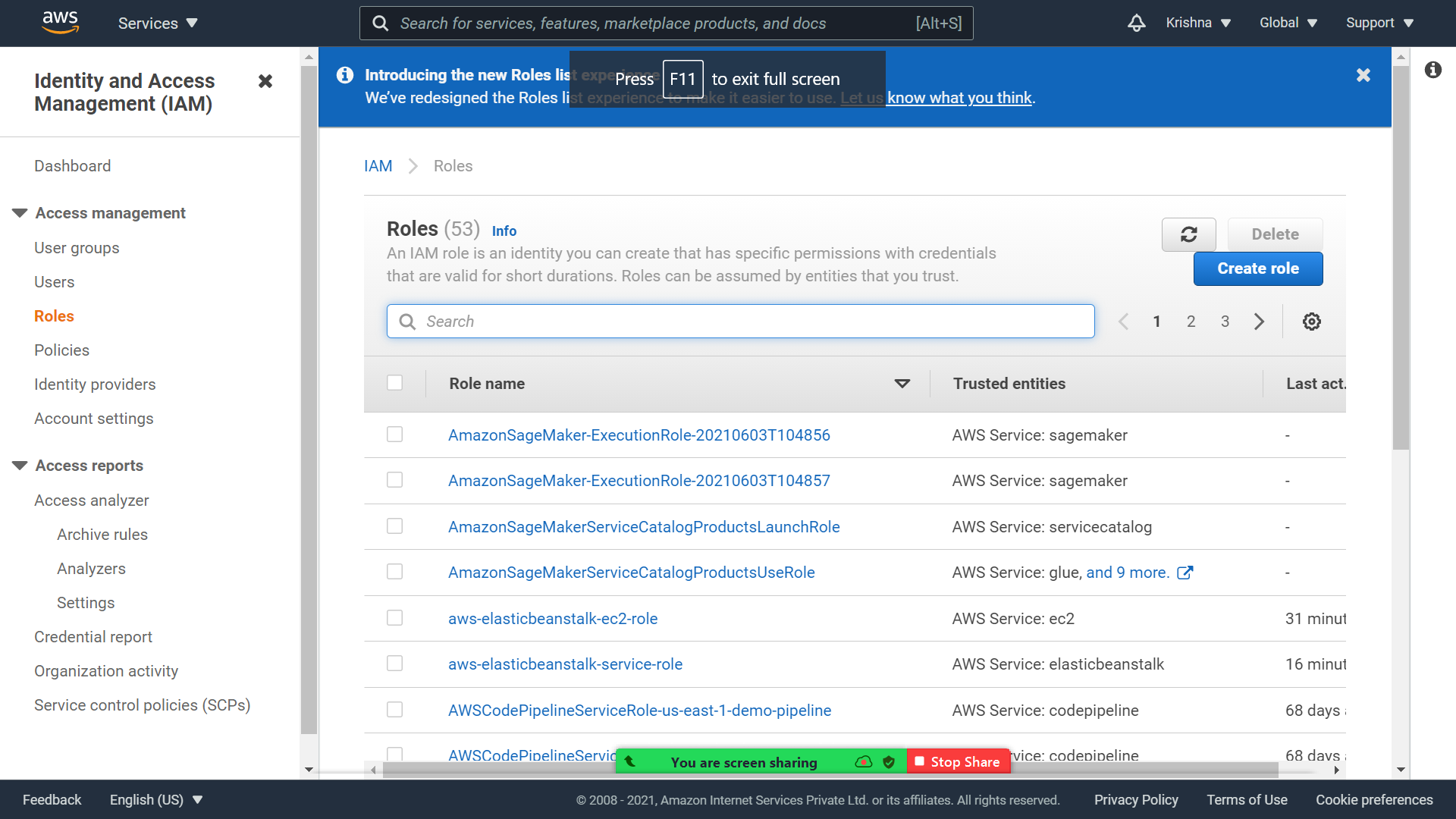
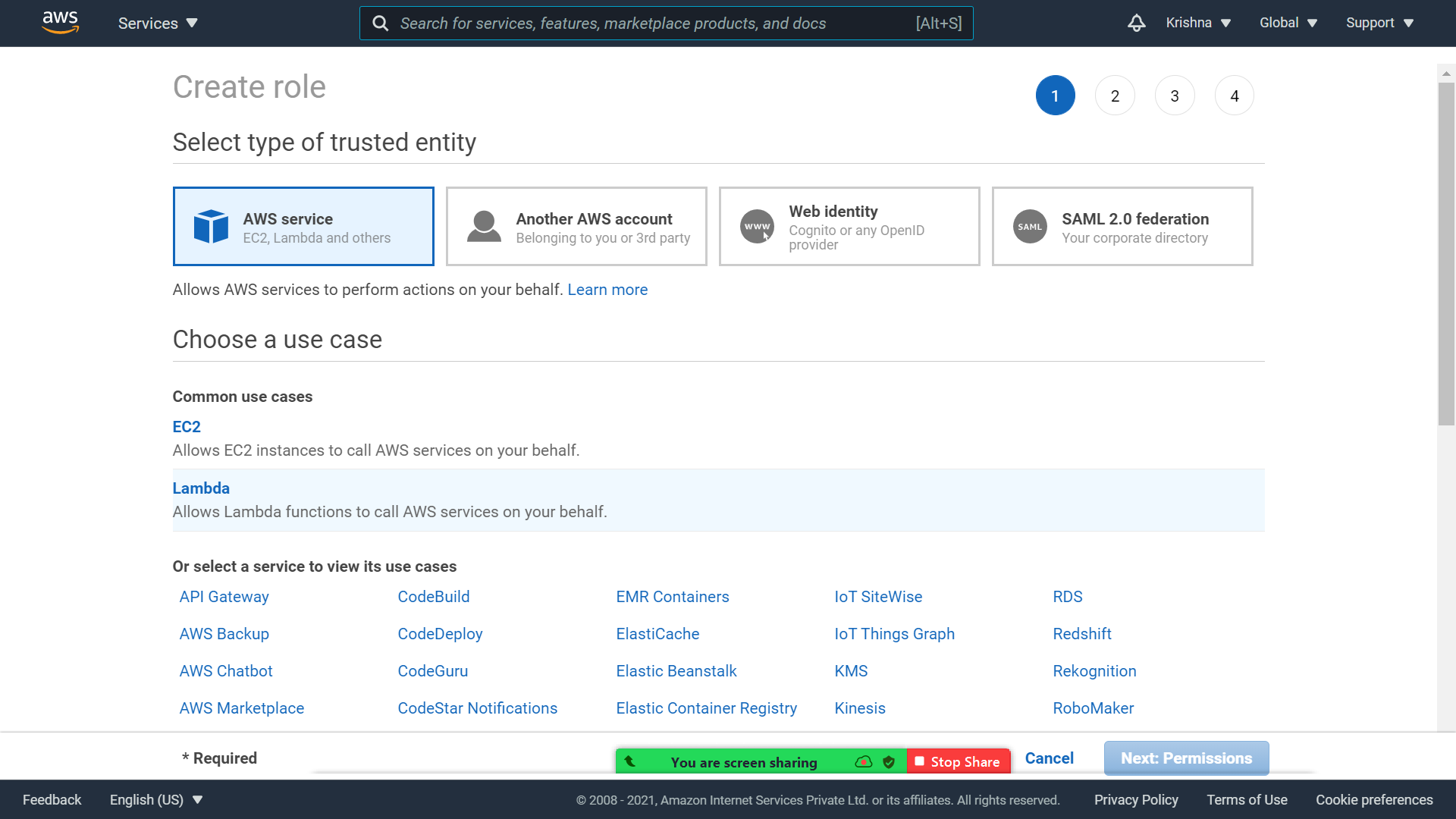
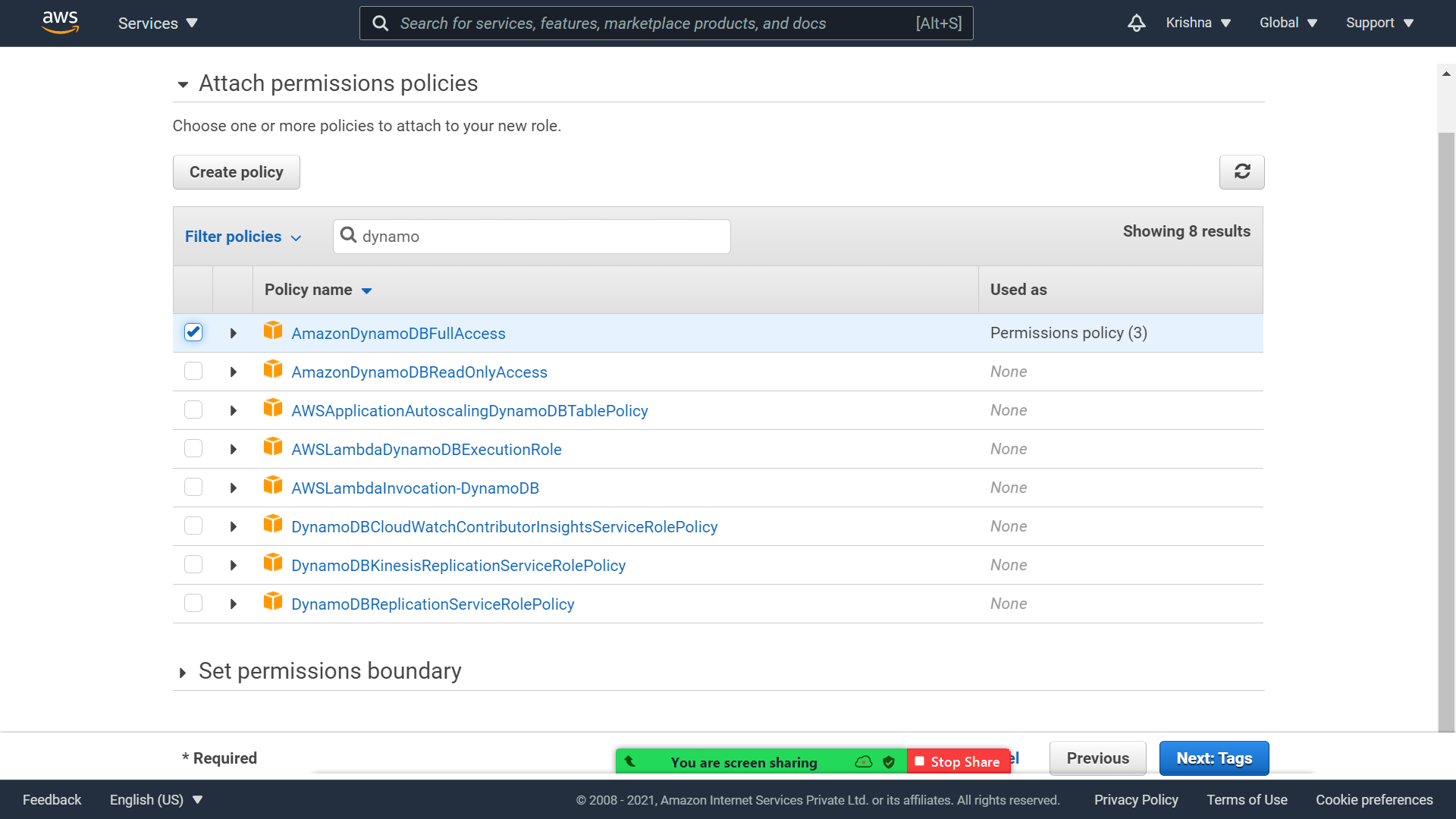
**Update metadata in Dynamo Db when any file is updated in S3 bucket**

1. **Create IAM role**

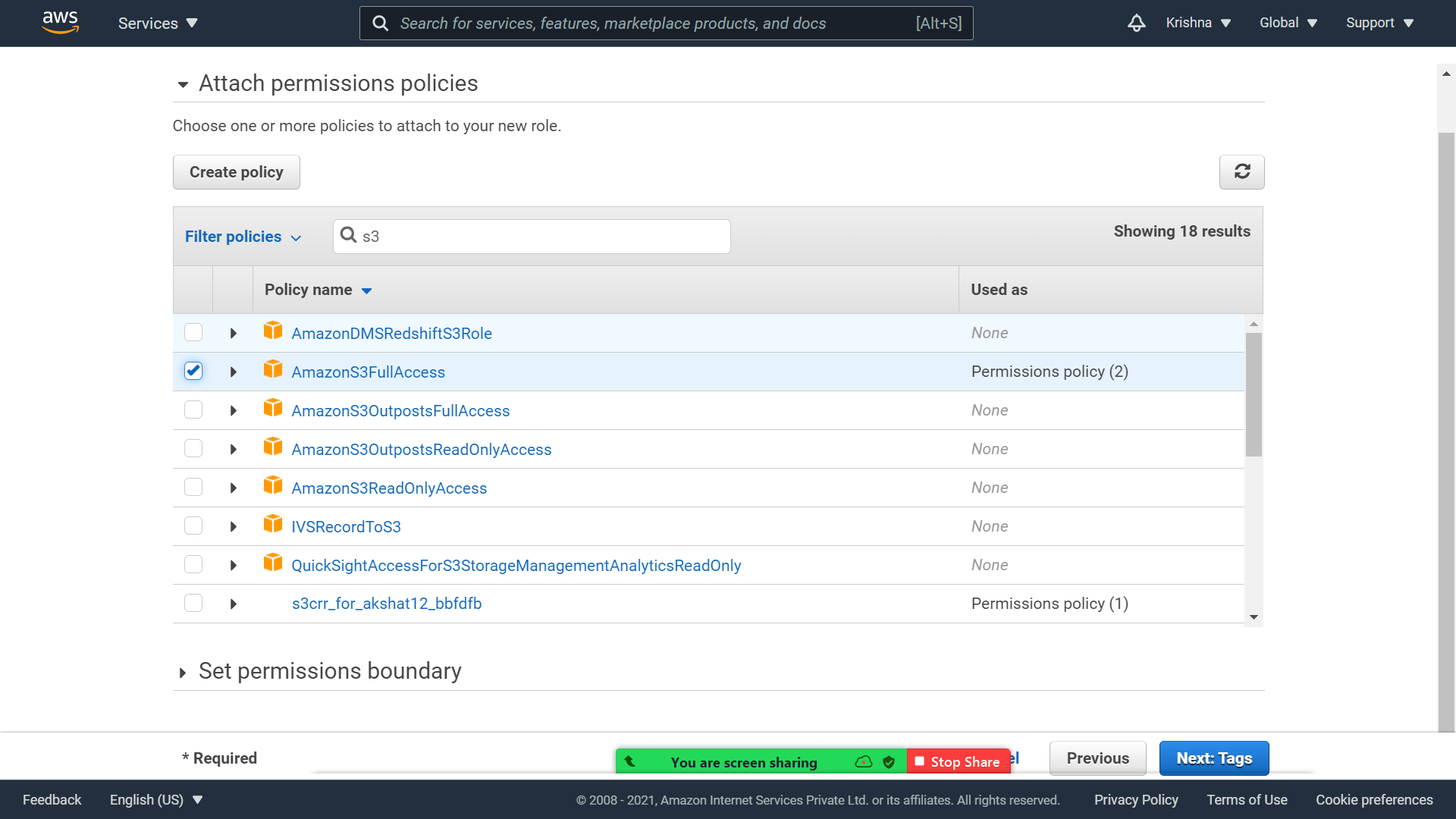


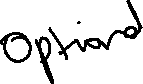


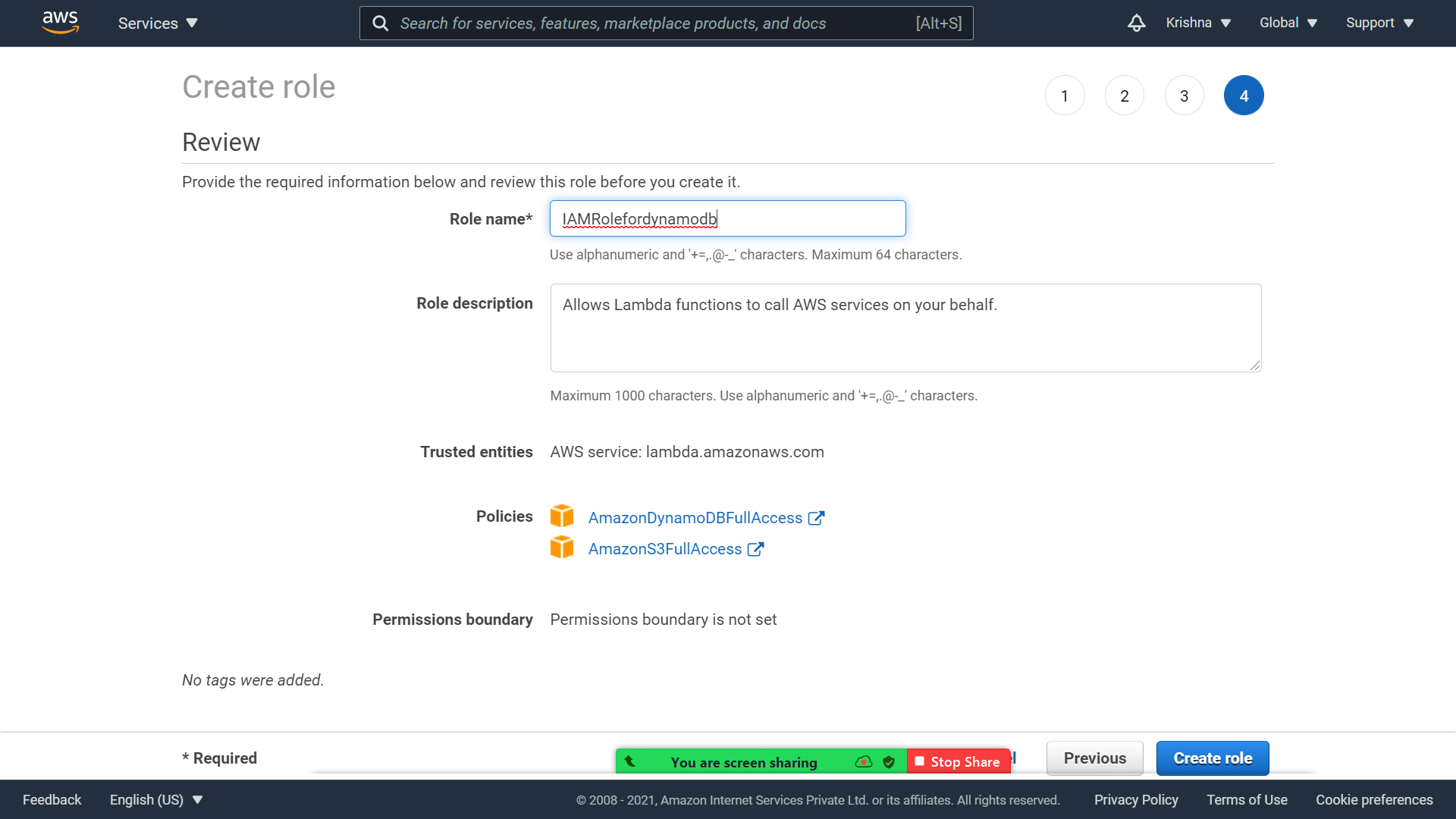


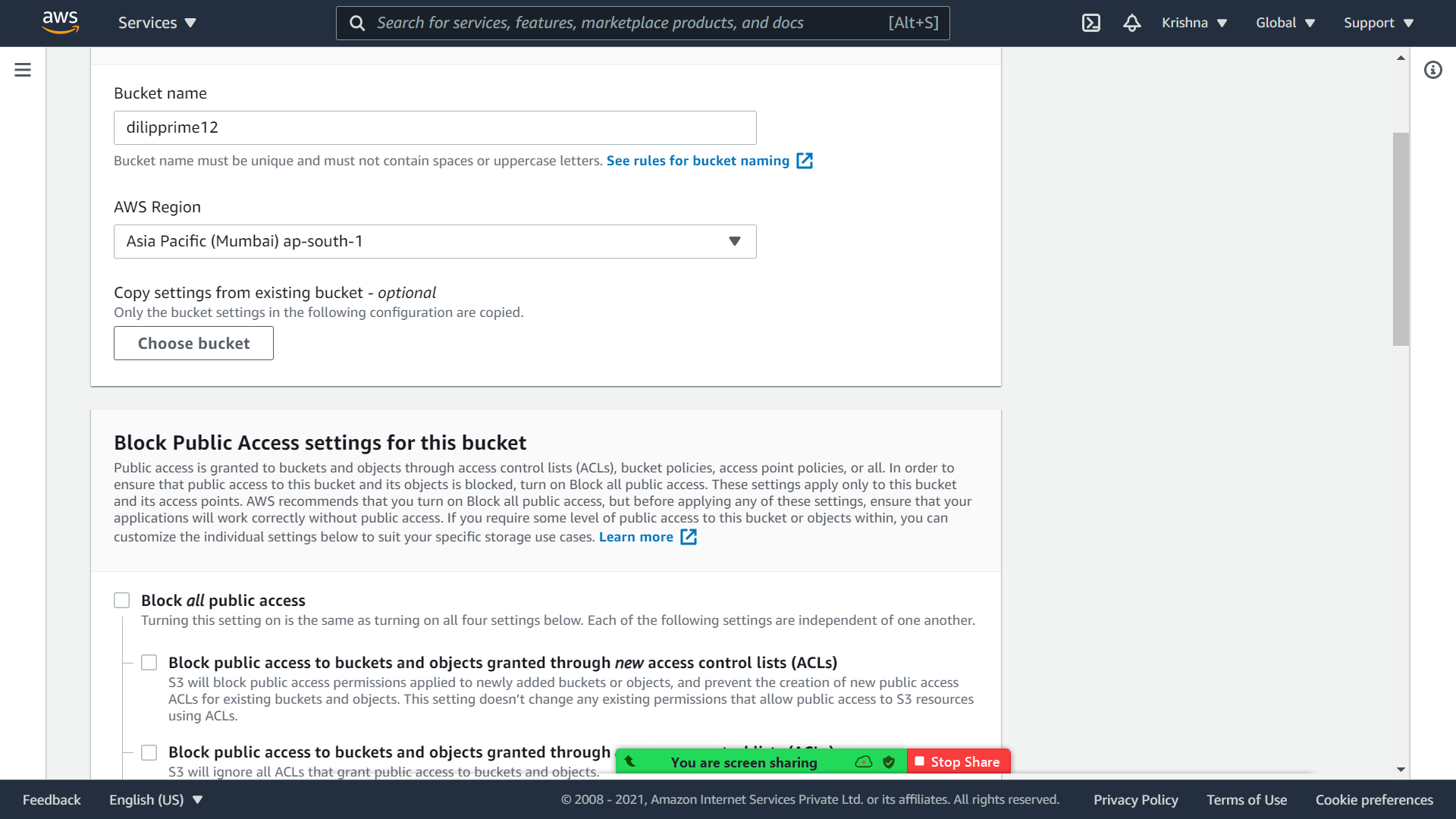






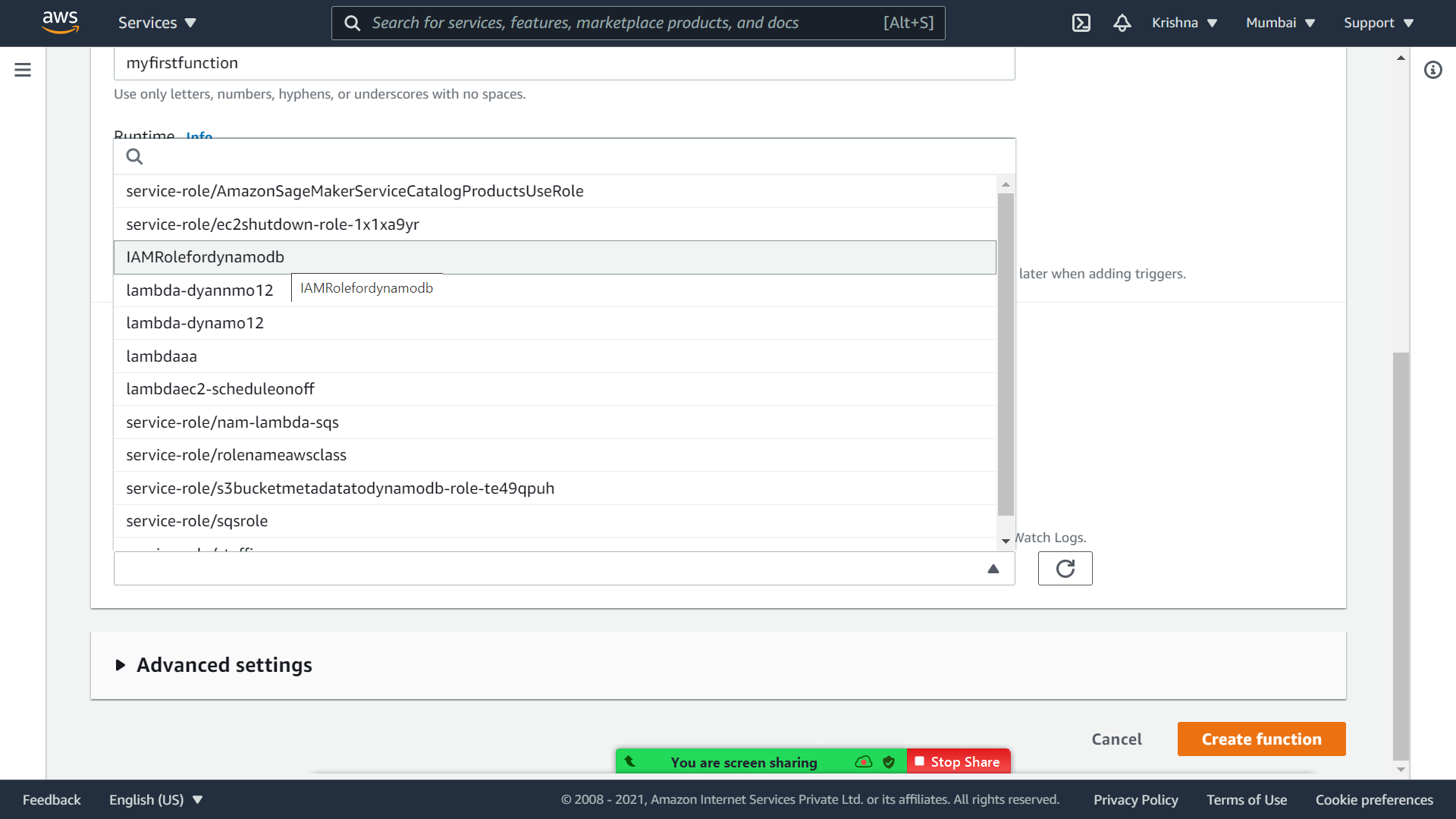
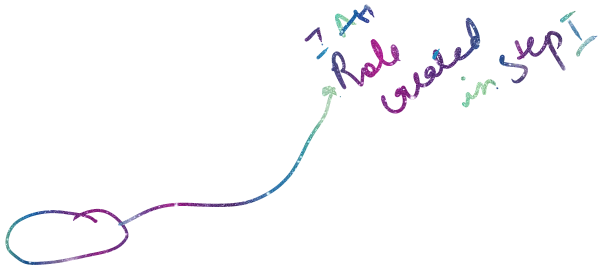




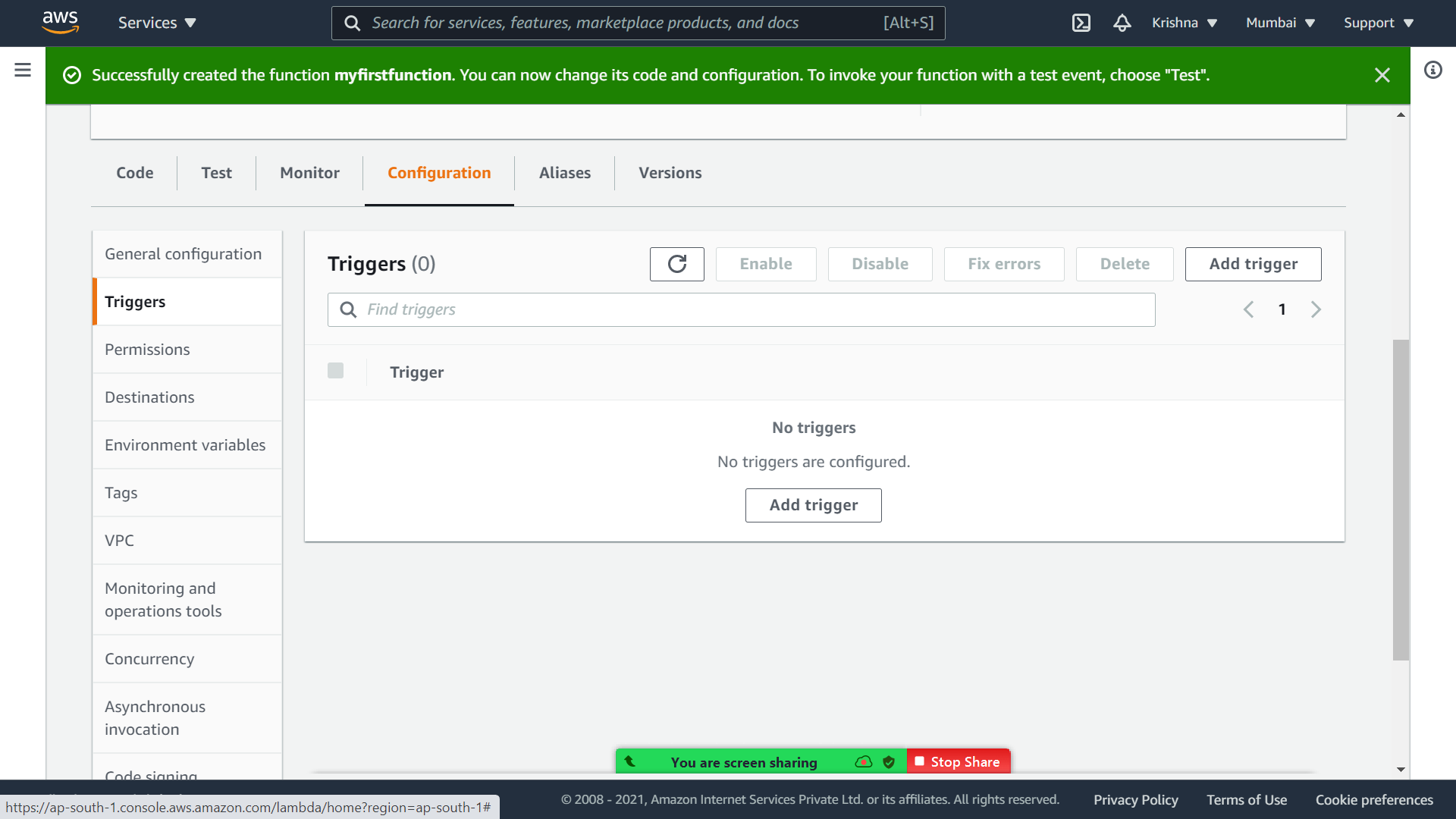
1. **Create s3 bucket and give public access.**

1. **Create Lambda**

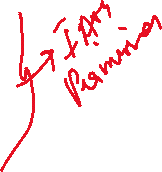




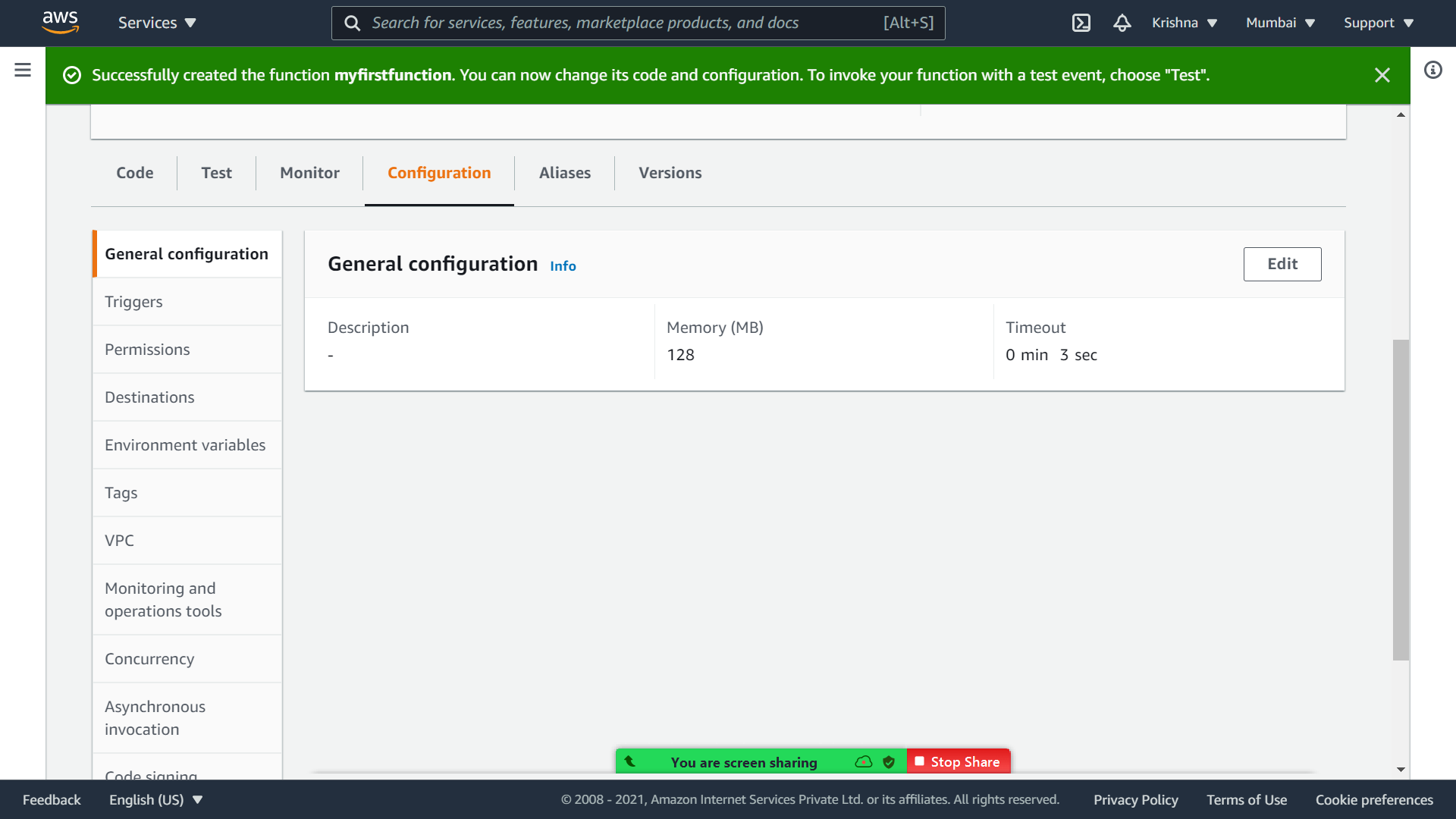
**Check configuration**

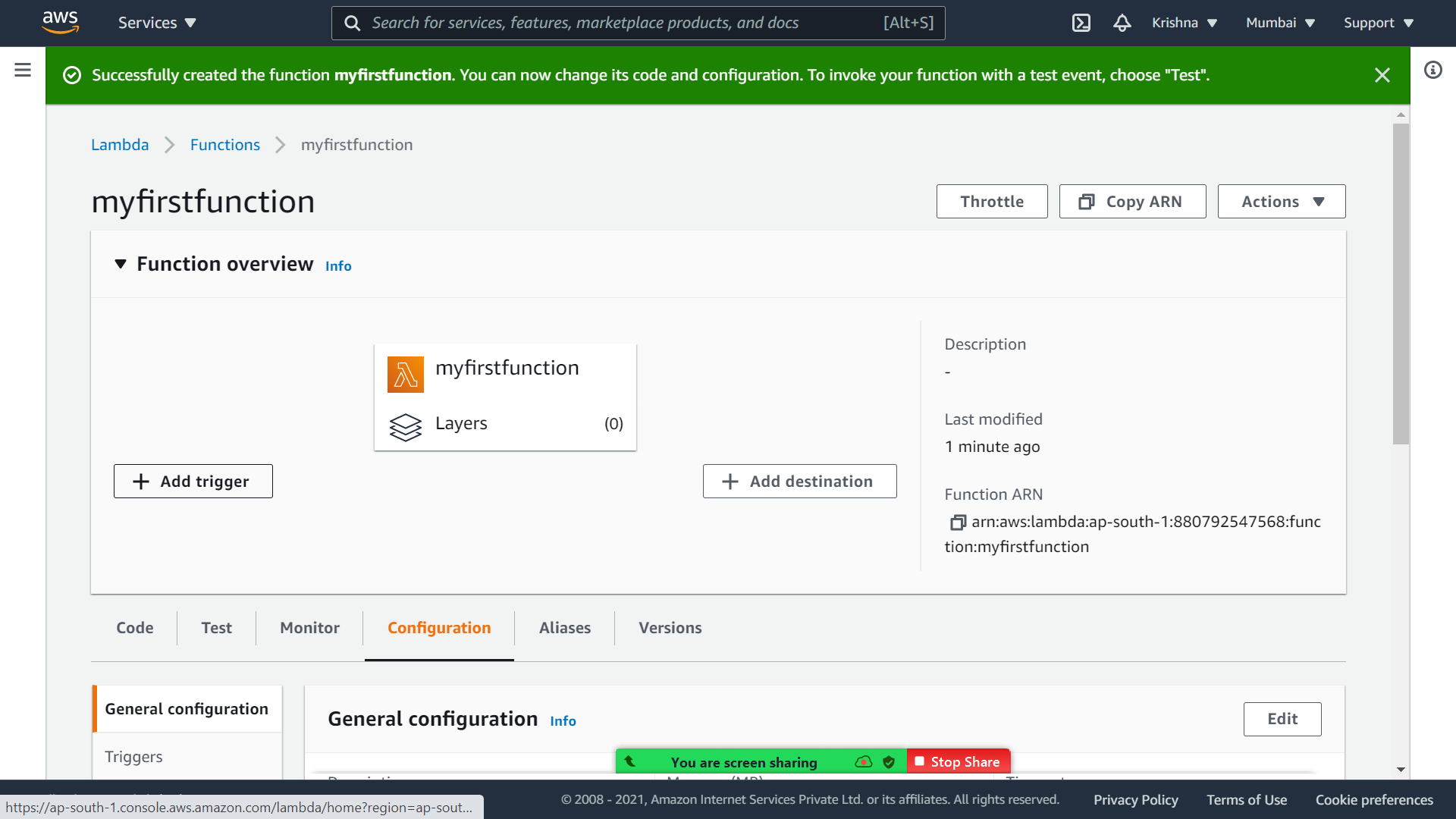






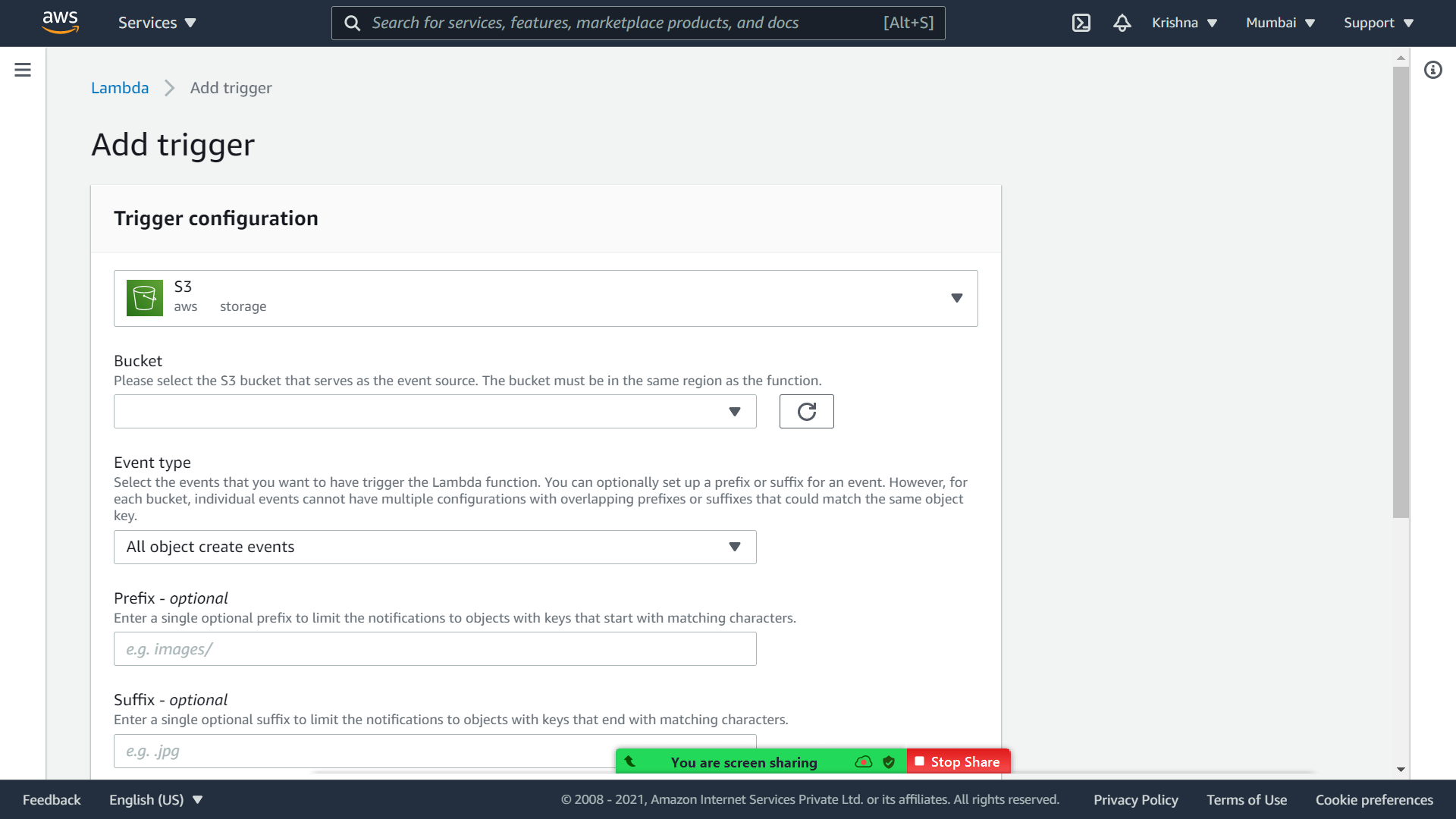
**Check timeout and max space from general config**

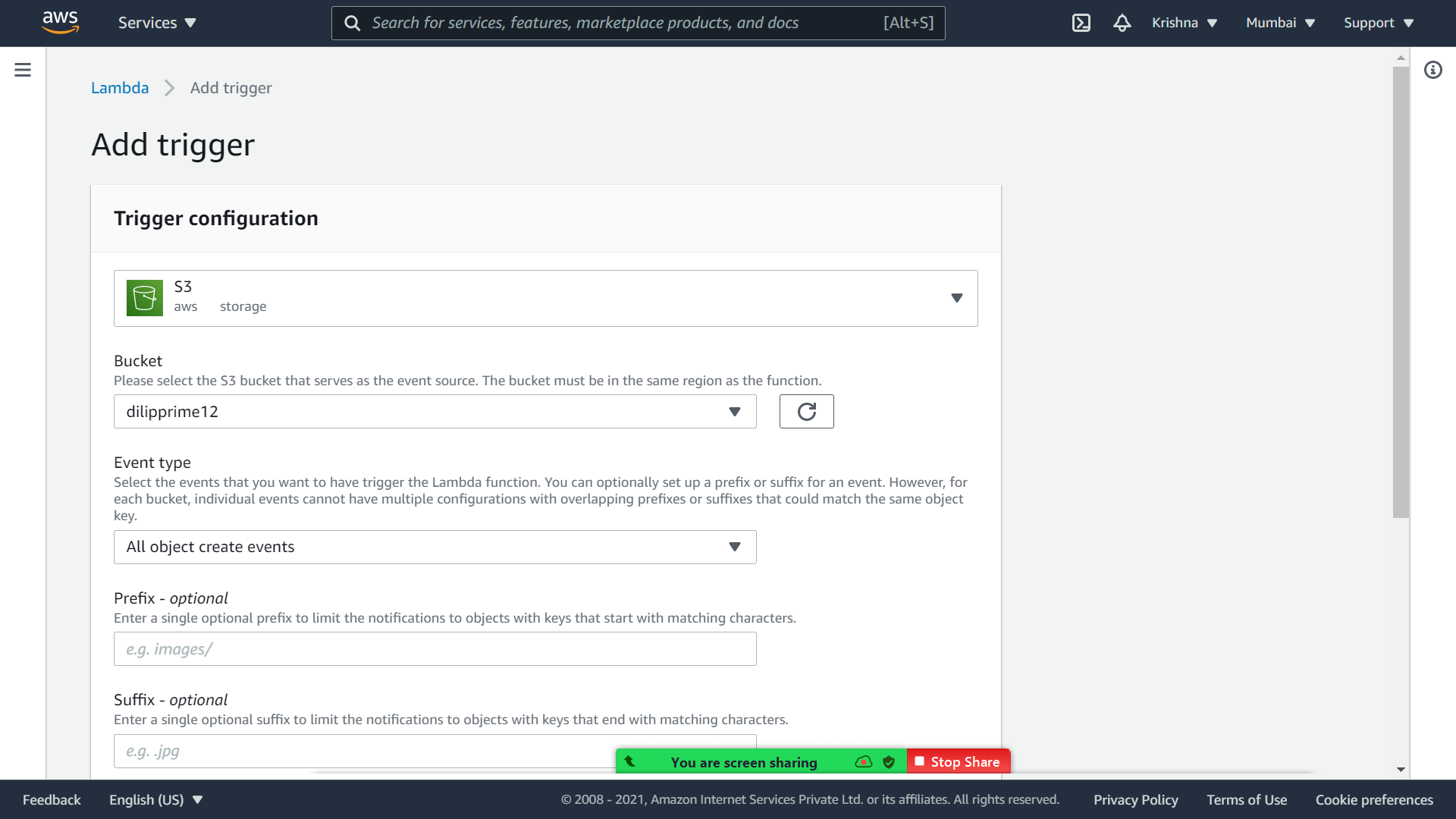
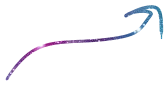
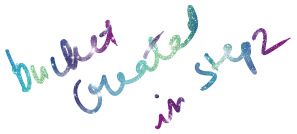




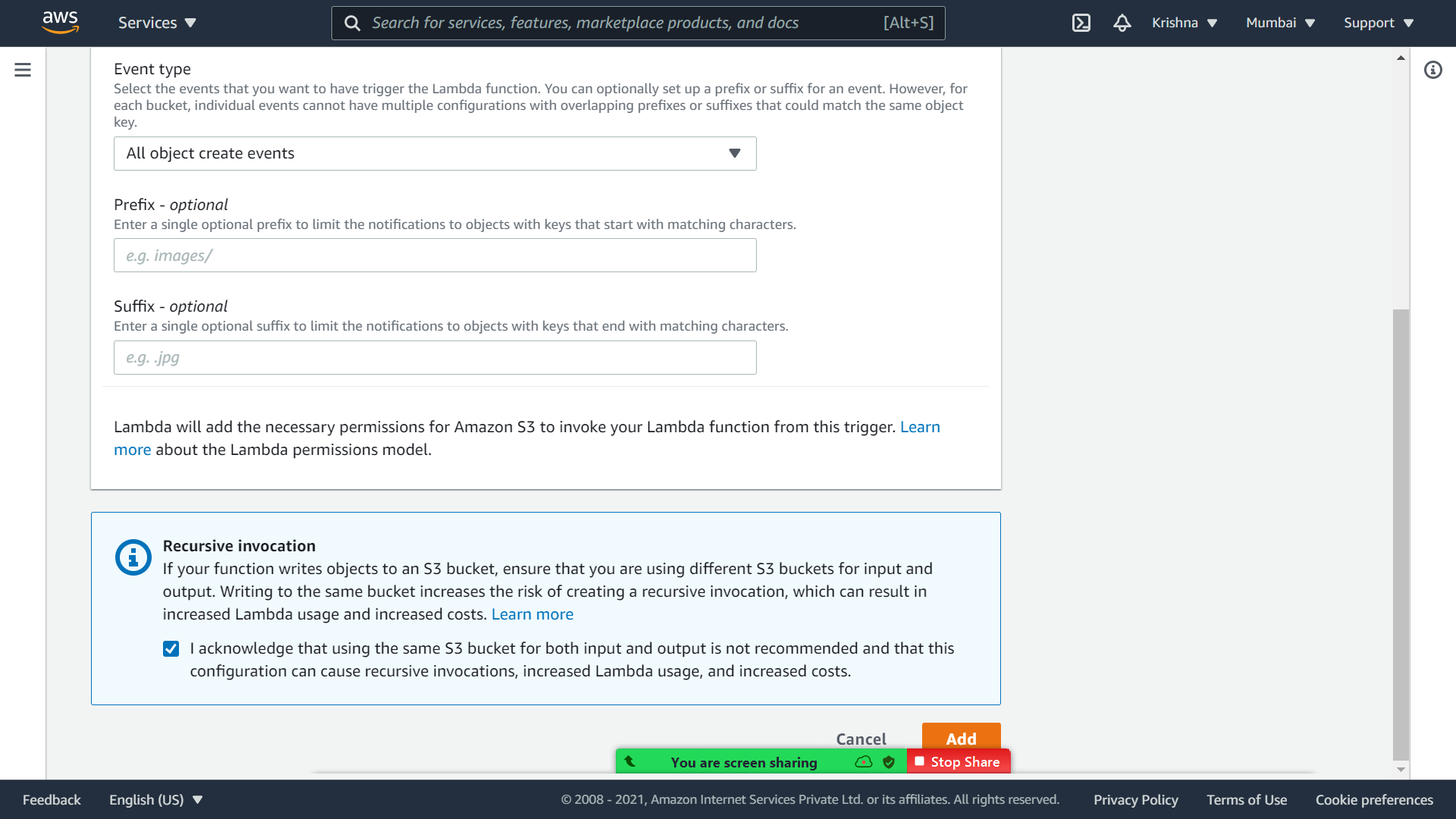
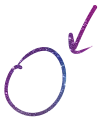
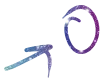


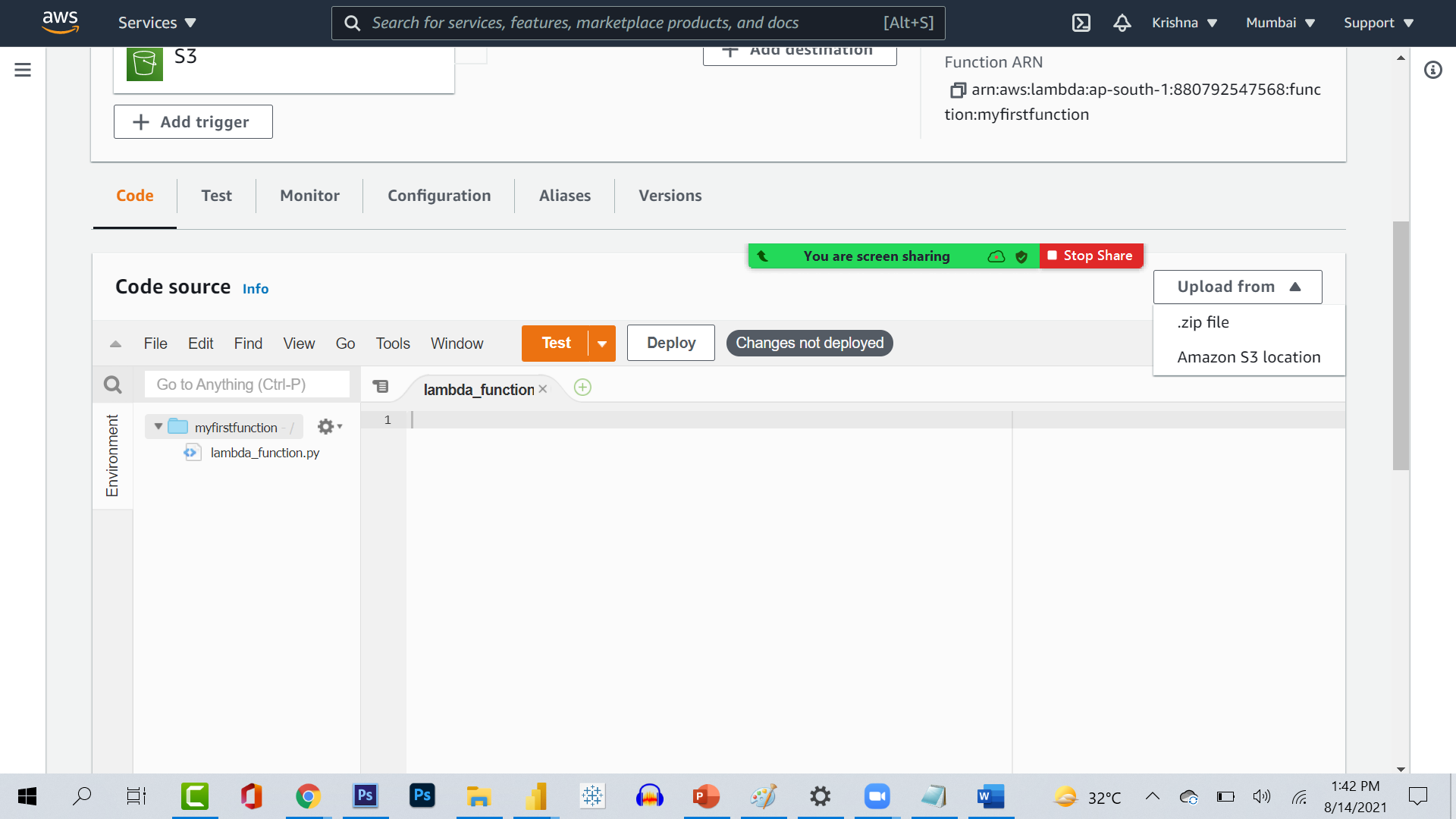
**After clicking on Add trigger**











**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)**

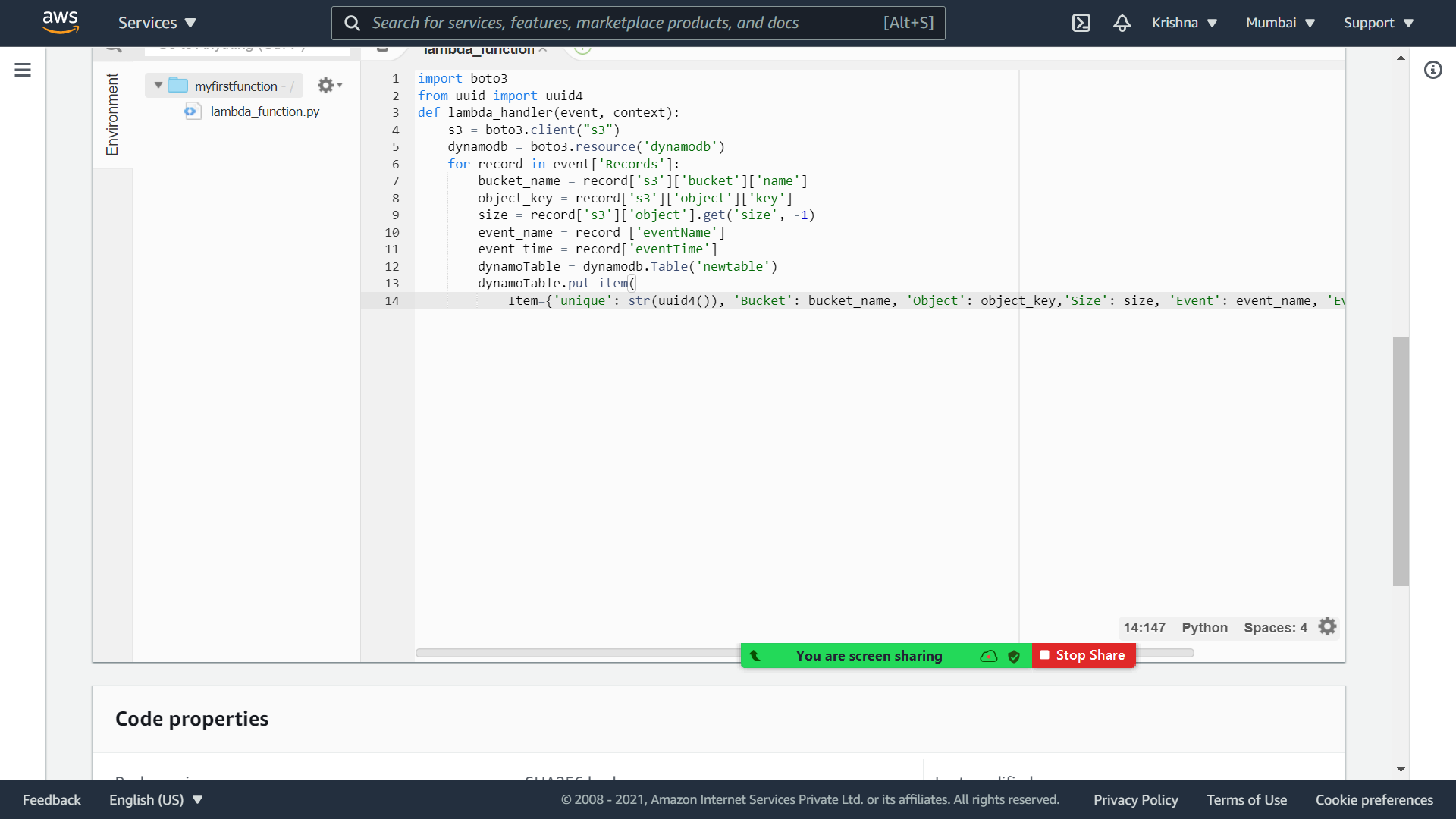
**event\_name = record ['eventName']**

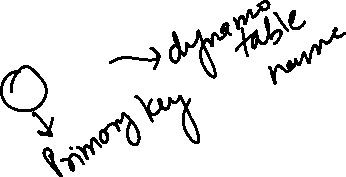
**event\_time = record['eventTime']**

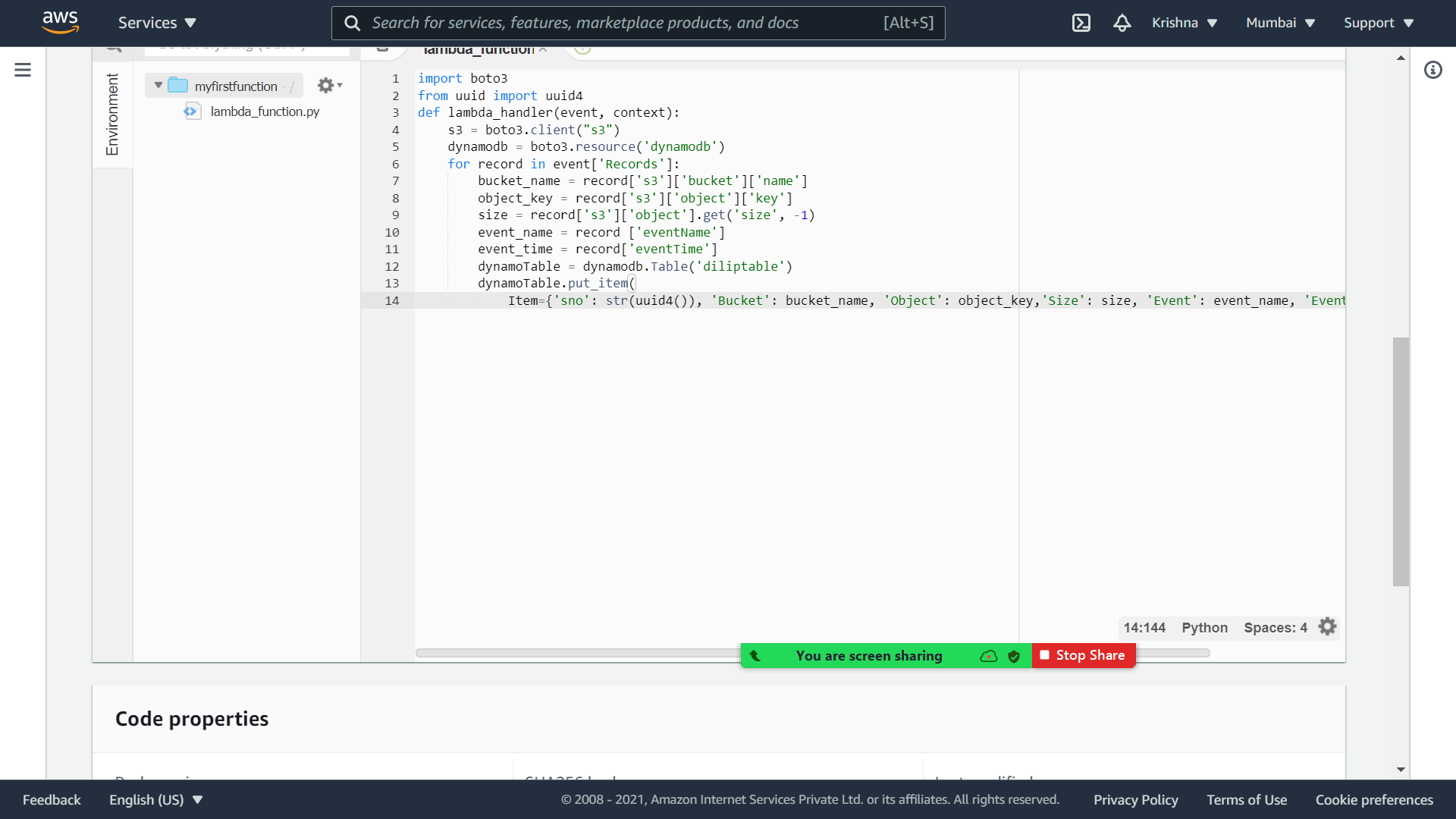
**dynamoTable = dynamodb.Table('newtable')**

**dynamoTable.put\_item(**

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



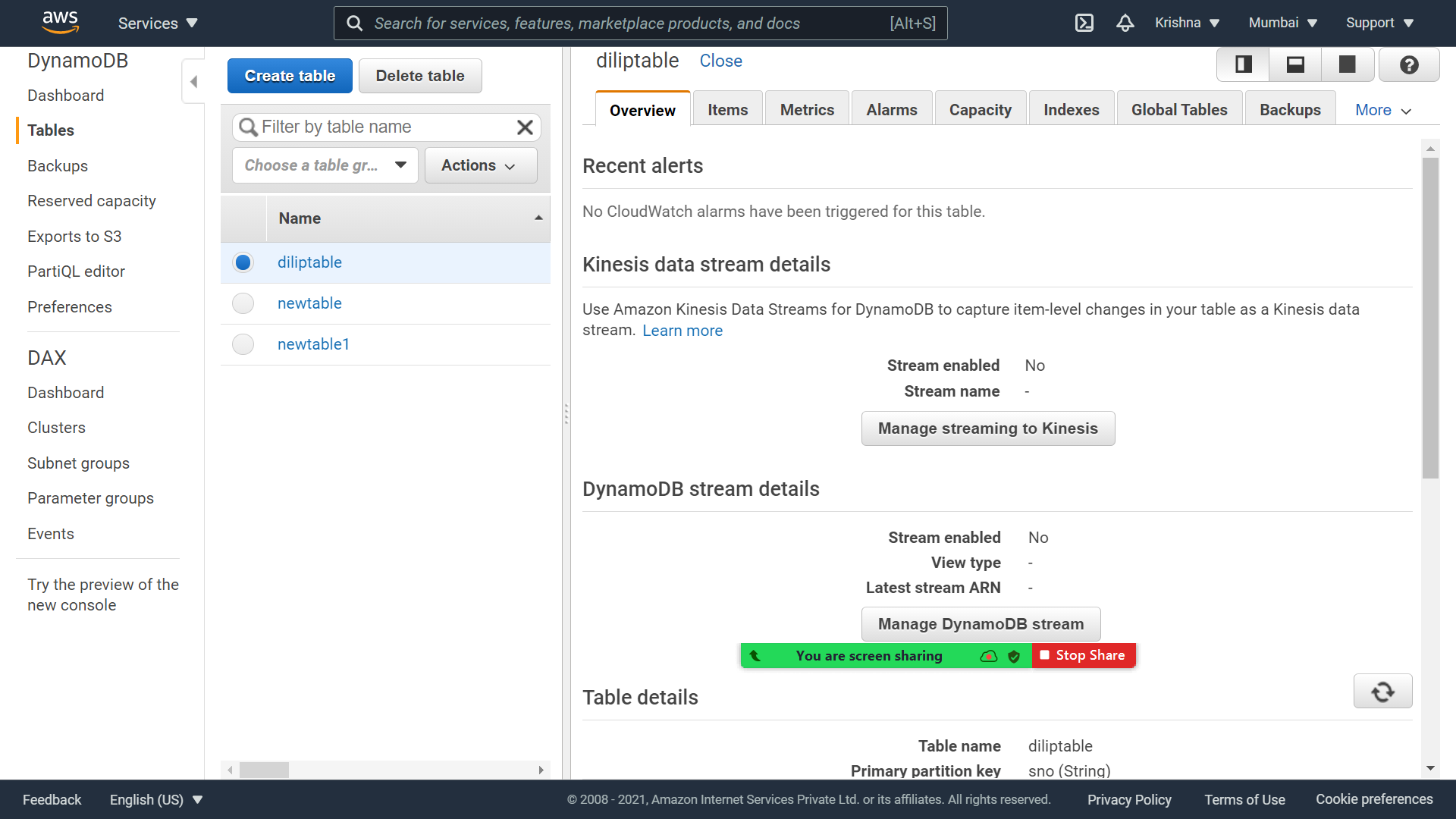




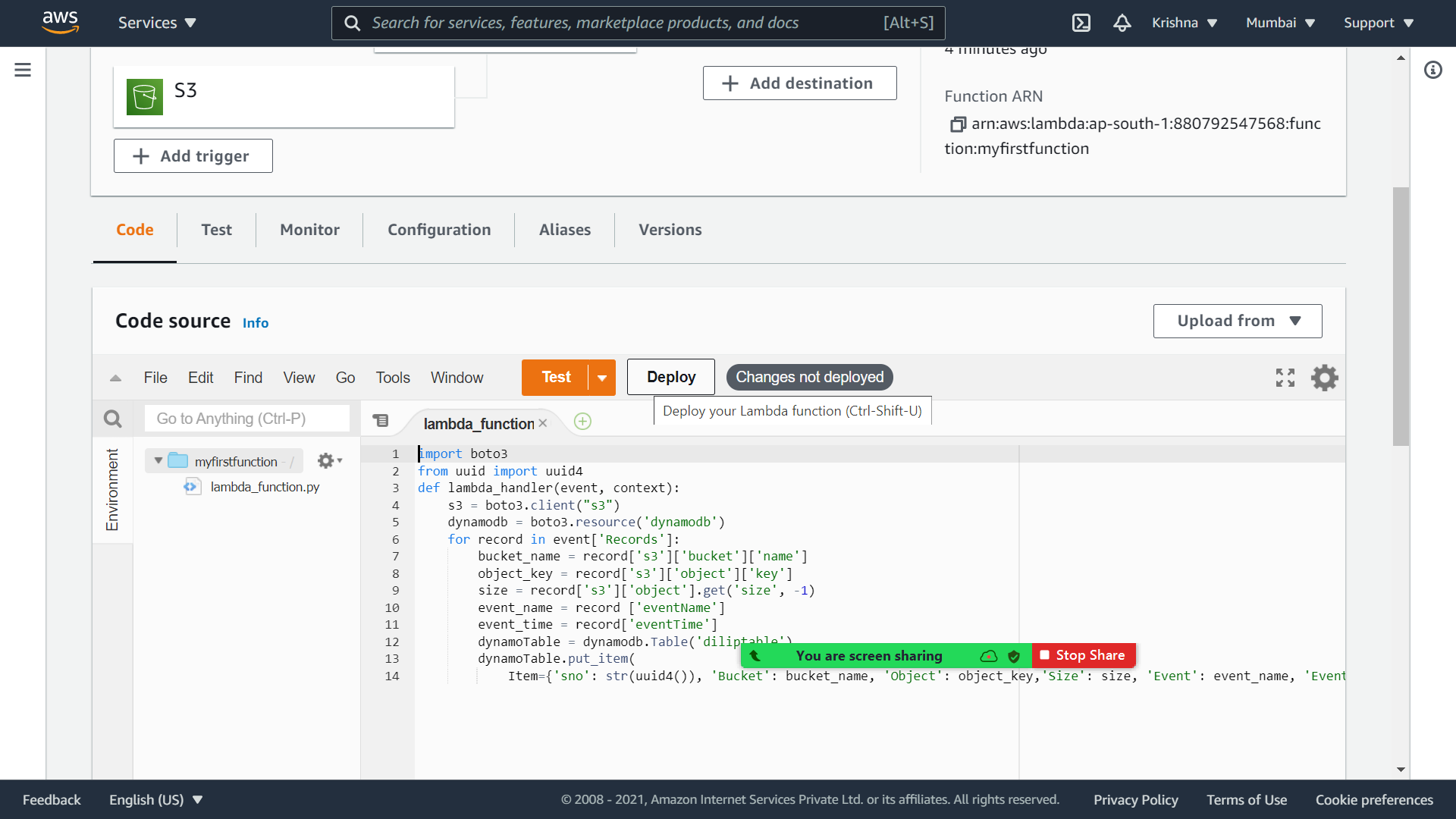


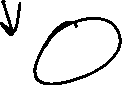
**Now go to dynamo db and create table with the name diliptable**

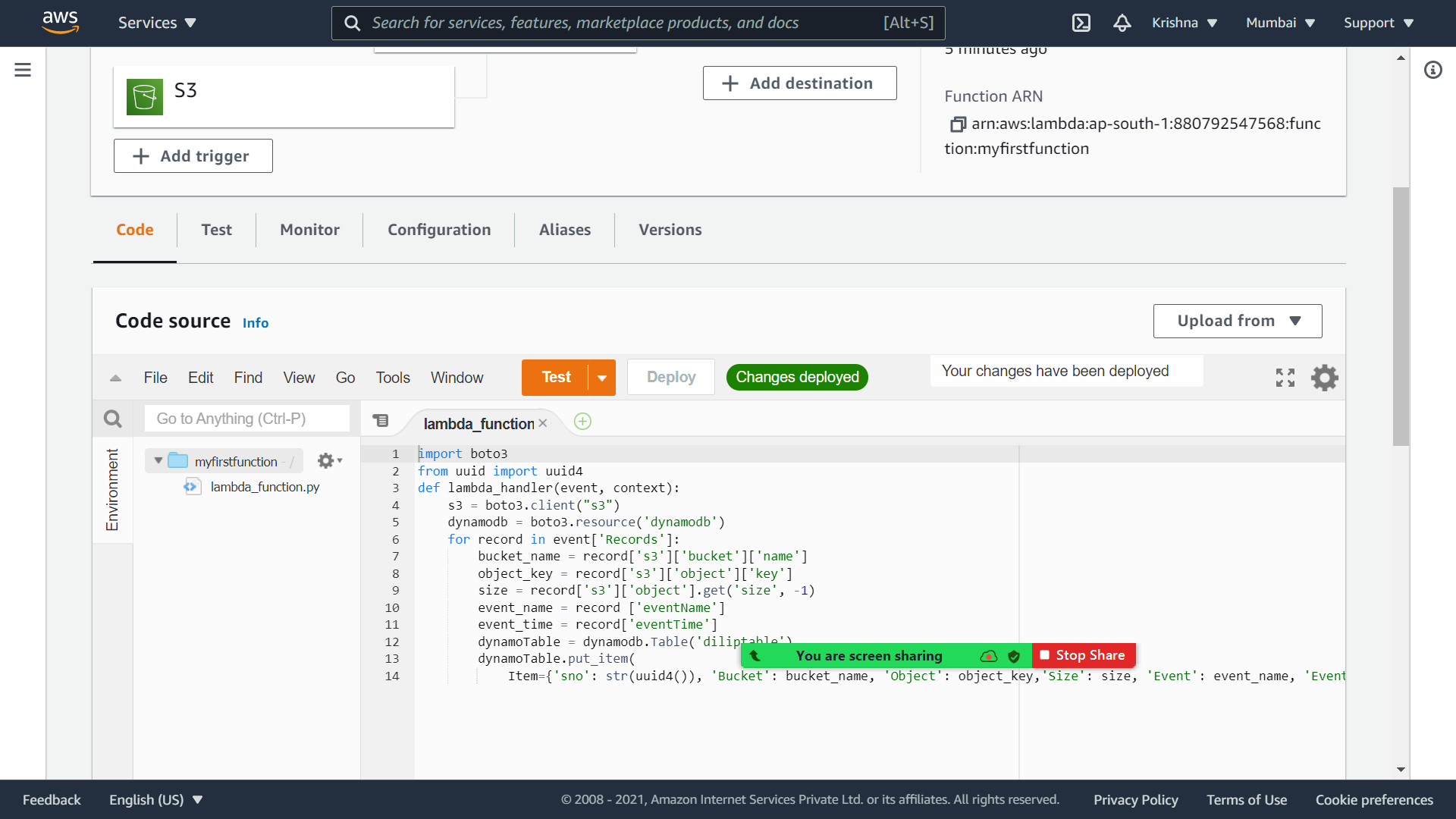
**And unique key as sno**



**Click on deploy in lambda (go to lambda and selection the function )**

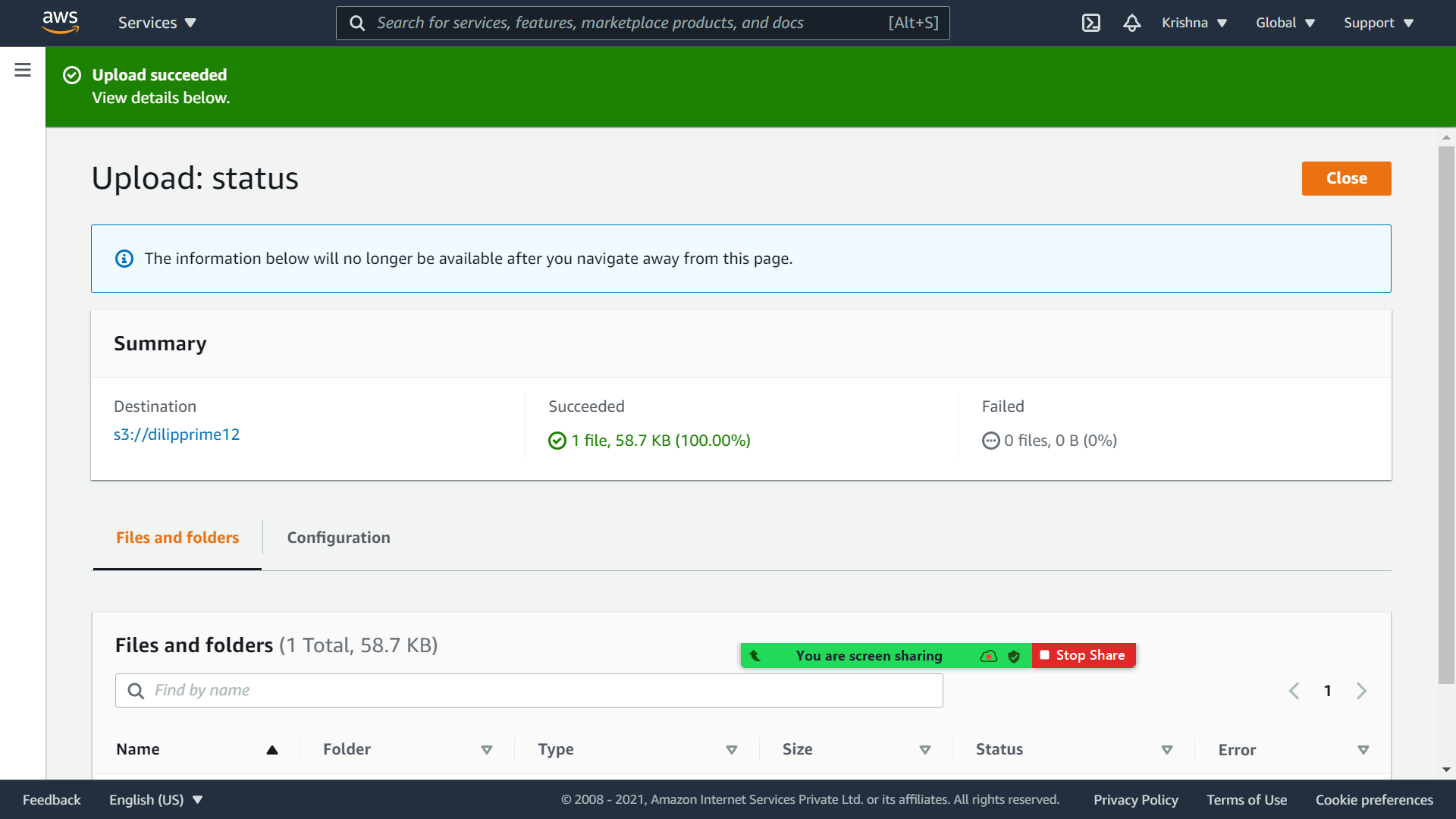








**Now go to s3 bucket and upload the content in s3 bucket**



**Check the dilip table in dynamo db**

