20CS3234AA – ADC

Name: B. Naveen

Reg.no: 2000031509

Experiments

Create one AWS Lambda function which can print the success message in CloudWatch after inserting data into DynamoDB and that lambda function need to be triggered by the Restful API Gateway.

The API Gateway need to be authorized by the token value of Amazon Cognito Login Page

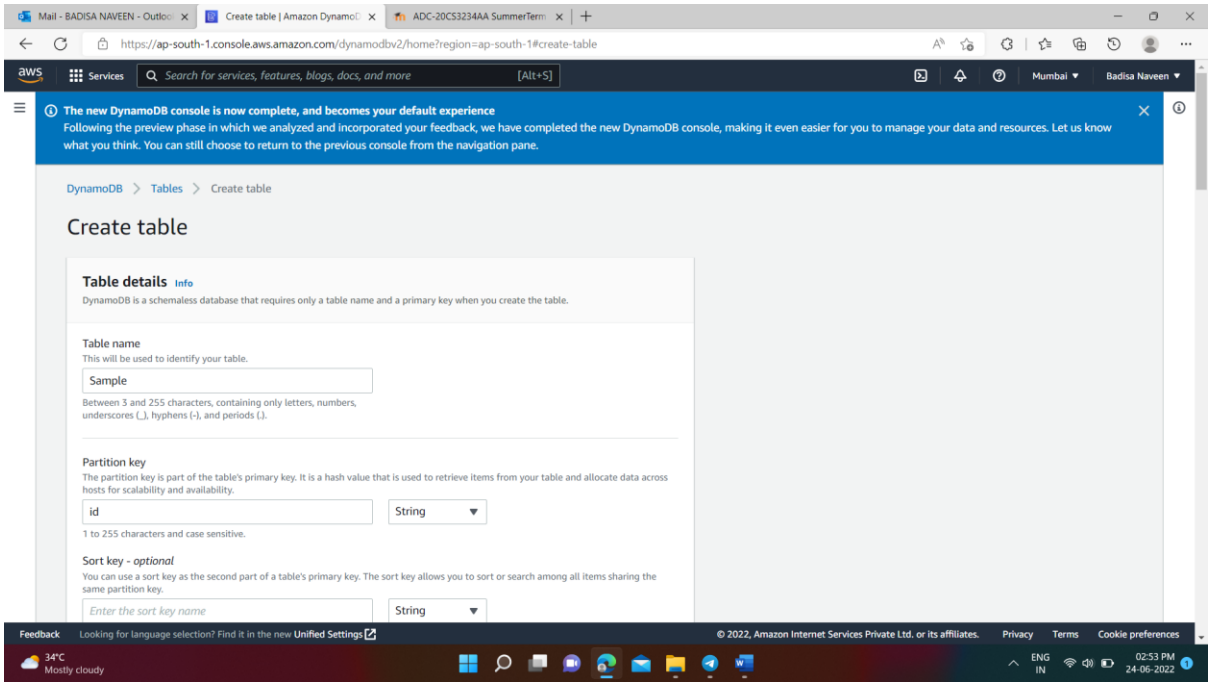
If request header has the proper token vale, then allow the lambda to display the success message and insert the data into DynamoDB. The success message need to be printed in Postman App and as well as Amazon CloudWatch.

If the request header is failed to provide the token value then display unauthorized

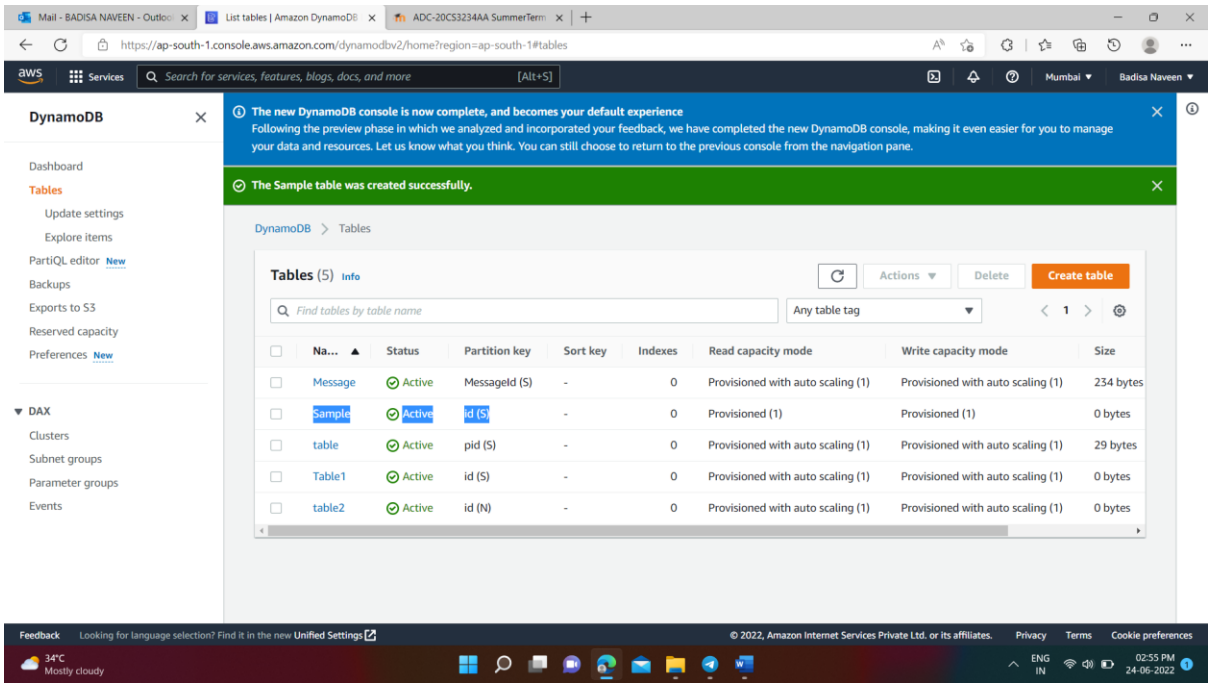
You can use Postman app (to set the header token), to display the output instead of the browser

A.

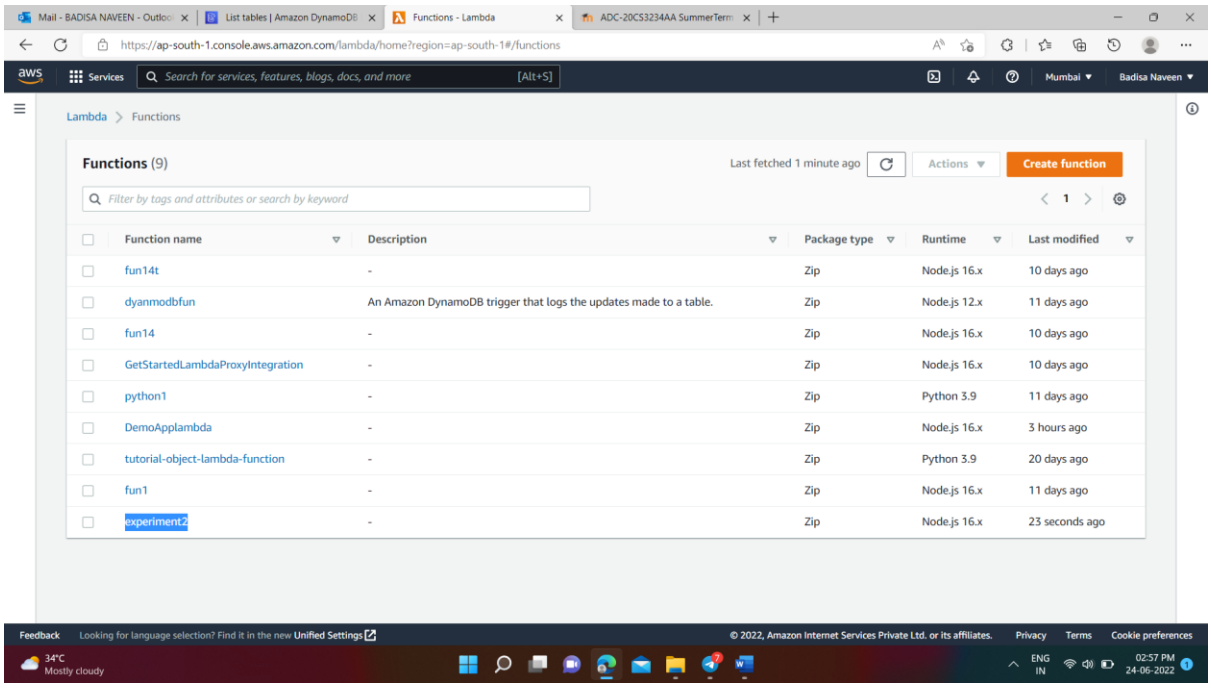
Step1: Create a table in Dynamodb



Step2: Dynamodb Table created



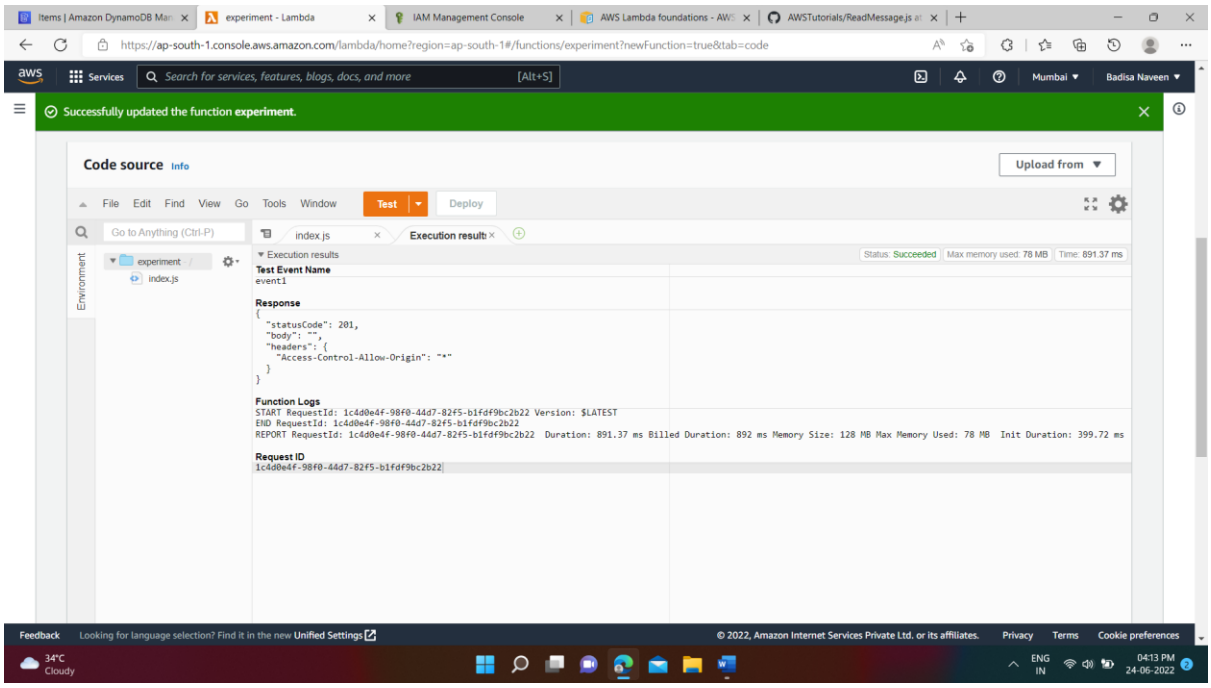
Step3: Created a lambda function



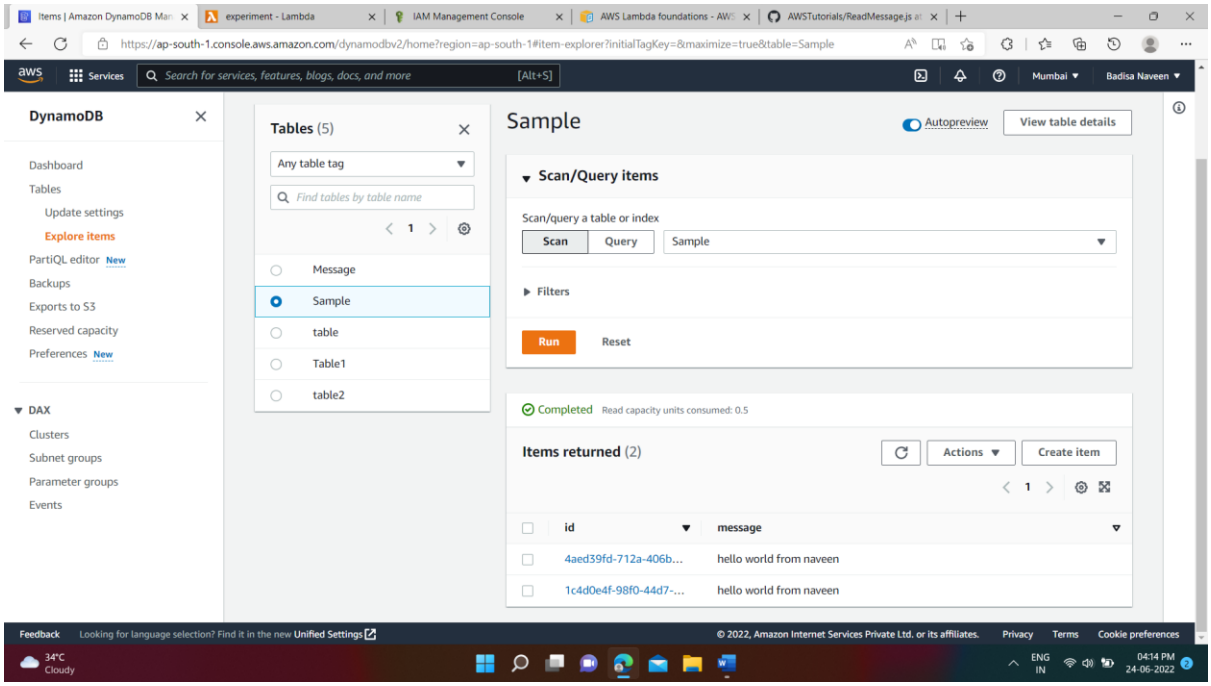
Step4: write the lambda code to write in dynamodb table

```
const AWS=require('aws-sdk');
const ddb=new AWS.DynamoDB.DocumentClient({region:'ap-south-1'});
exports.handler=async(event,context,callback)=>{
    const requestId=context.awsRequestId;
    await createMessage(requestId).then(()=>{
    callback(null,{
        statusCode:201,
        body:'',
        headers:{
            'Access-Control-Allow-Origin':'*'
        }
    })
    }).catch((err)=>{
        console.error(err)
    });
};
```

Step5: click on test



Step6: open dynamodb table and see the data inserted

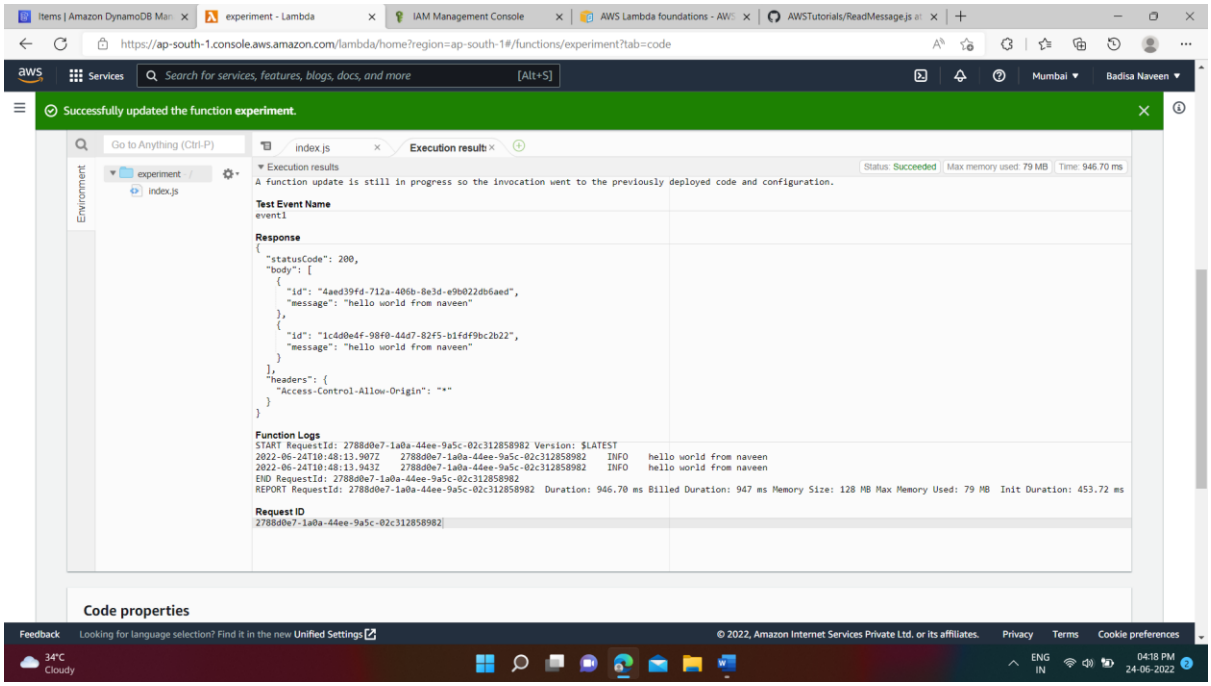


Step8: Read Message code

```
// Loads in the AWS SDK
const AWS = require('aws-sdk');


// Creates the document client specifing the region
// The tutorial's table is 'in us-east-1'
const ddb = new AWS.DynamoDB.DocumentClient({region: 'ap-south-1'});


exports.handler = async (event, context, callback) => {
    // Handle promise fulfilled/rejected states
    await readMessage().then(data => {
        data.Items.forEach(function(item) {
            console.log(item.message)
        });
        callback(null, {
            // If success return 200, and items
            statusCode: 200,
            body: data.Items,
            headers: {
                'Access-Control-Allow-Origin': '*',
            },
        })
    }).catch((err) => {
        // If an error occurs write to the console
        console.error(err);
```

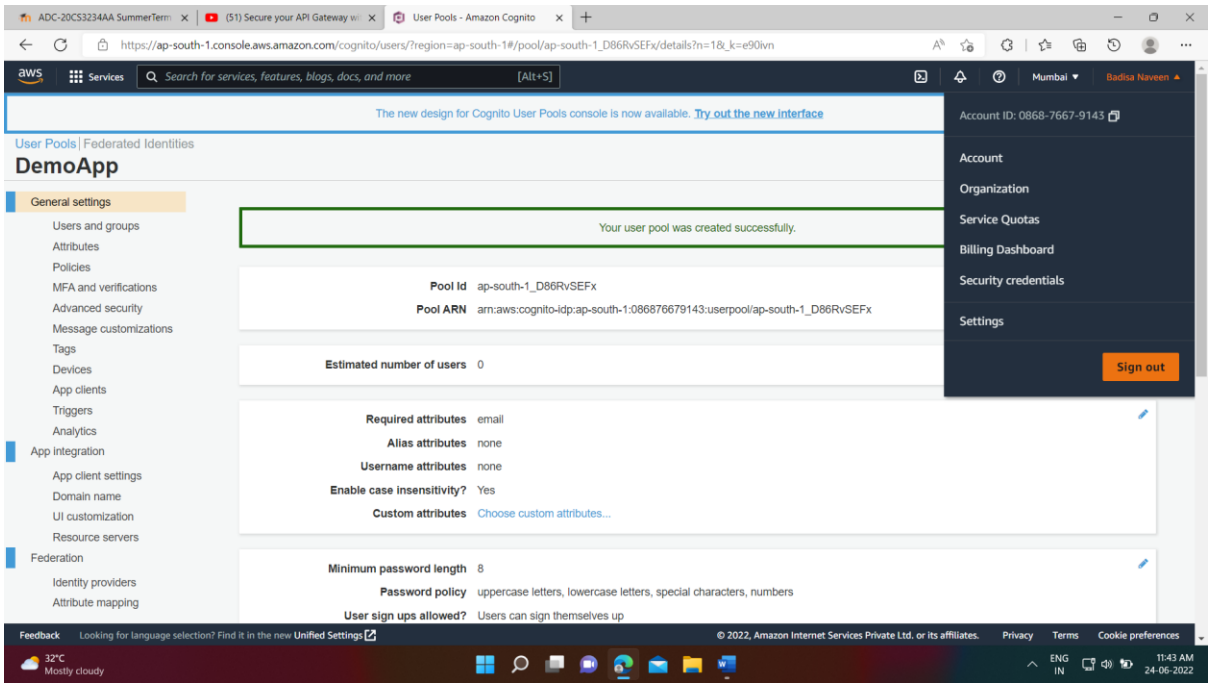
```
    })
};


// Function readMessage
// Reads 10 messages from the DynamoDb table Message
// Returns promise
function readMessage() {
    const params = {
        TableName: 'Sample',
        Limit: 10
    }
    return ddb.scan(params).promise();
}
```

Output:

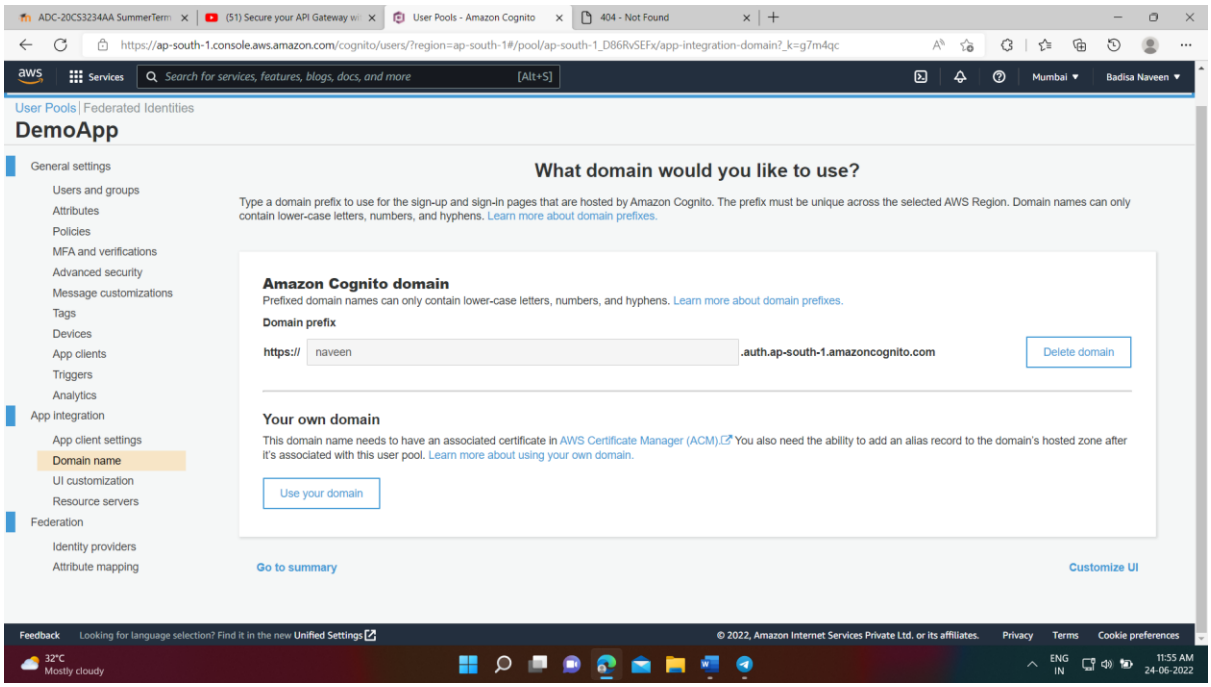


AWS COGNITO

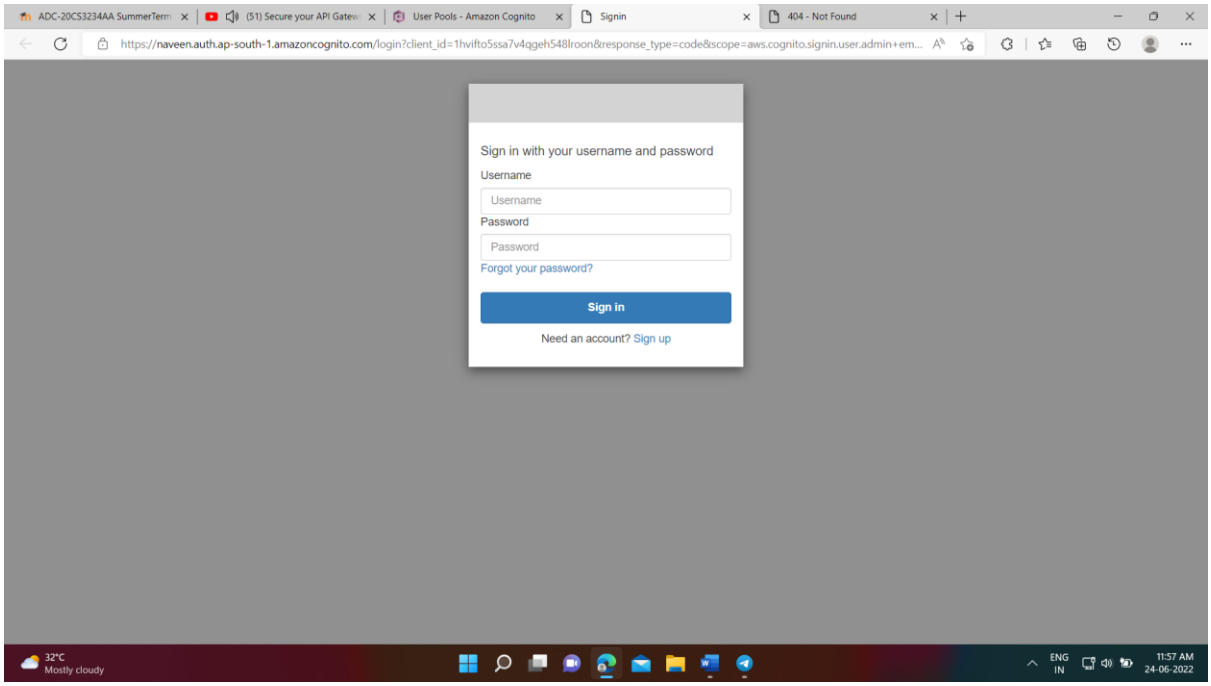
Step1: Open and create the Amazon cognito by creating user pool



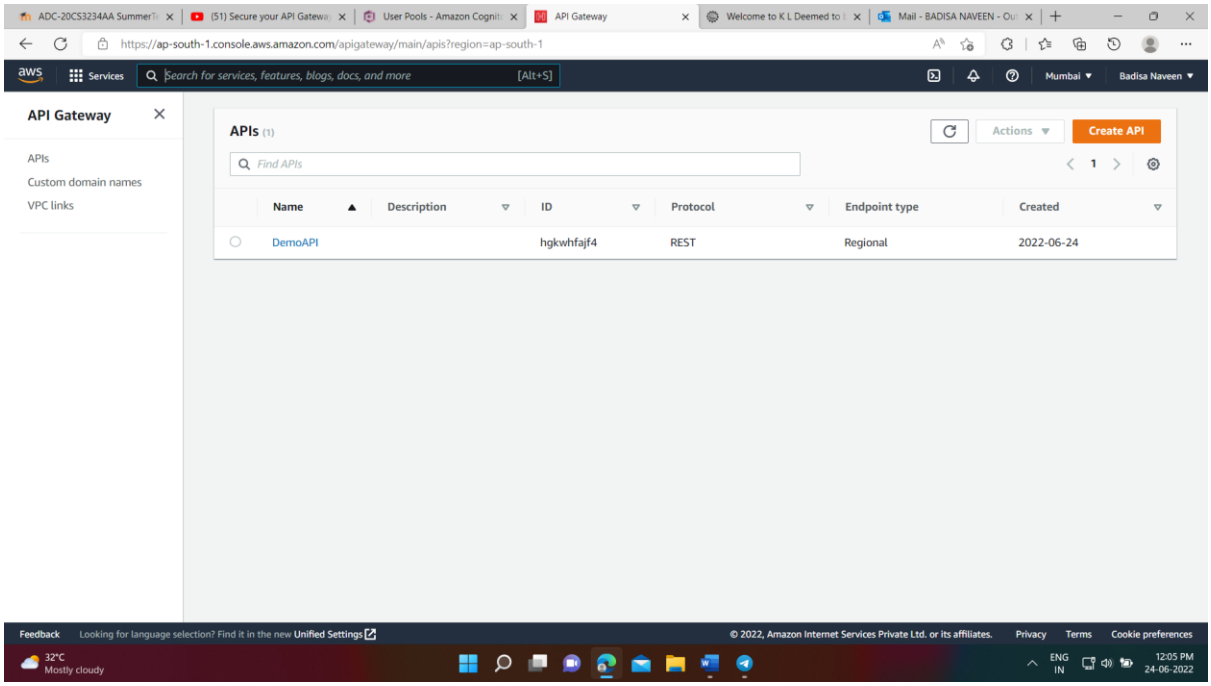
Step2: create a Domain Name



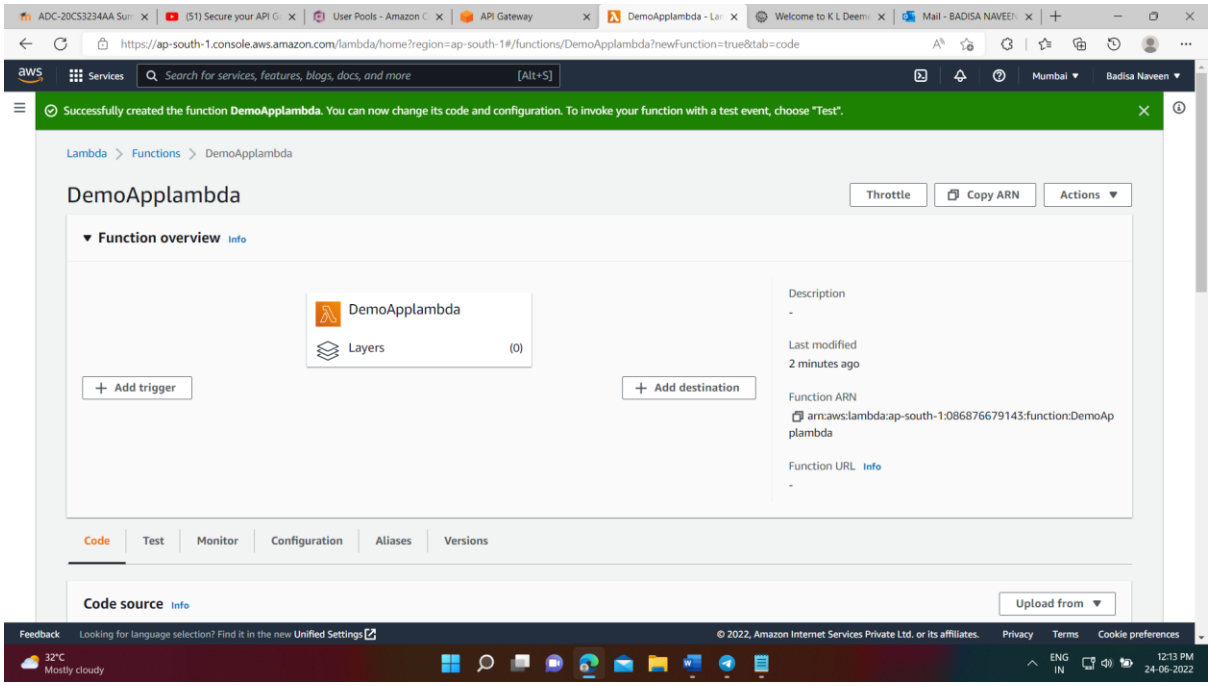
Step3: Signup and login



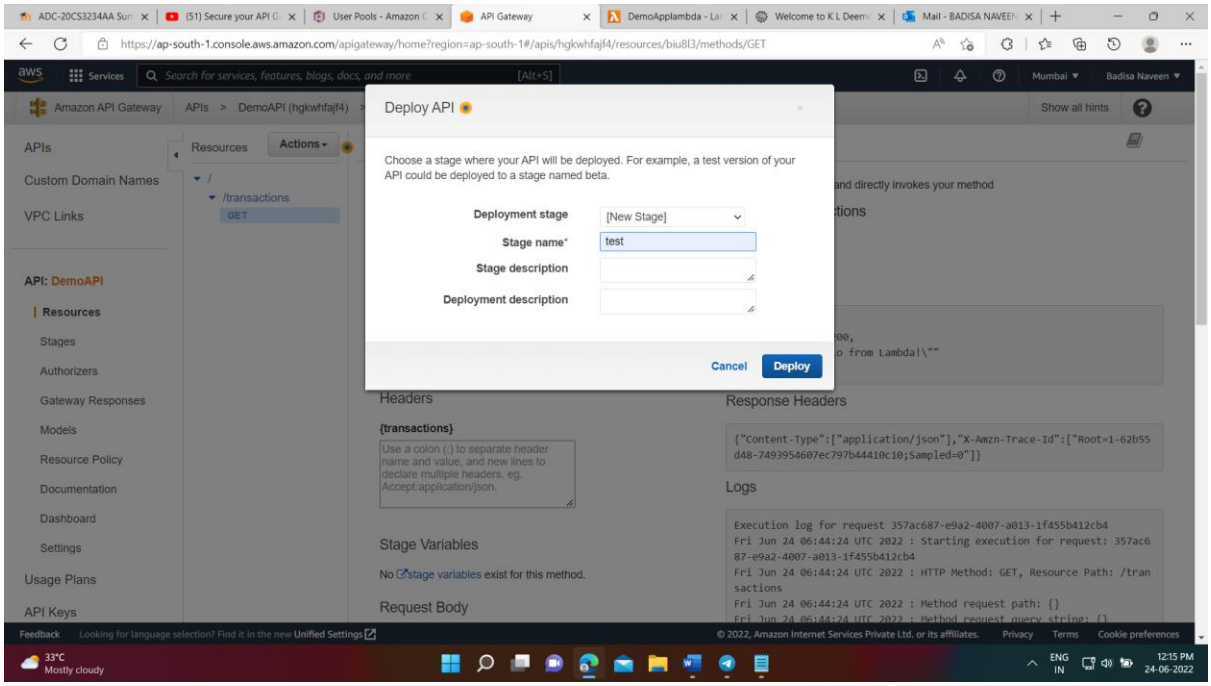
Step4: create API Gateway by choosing REST API



Step5: create Lambda function

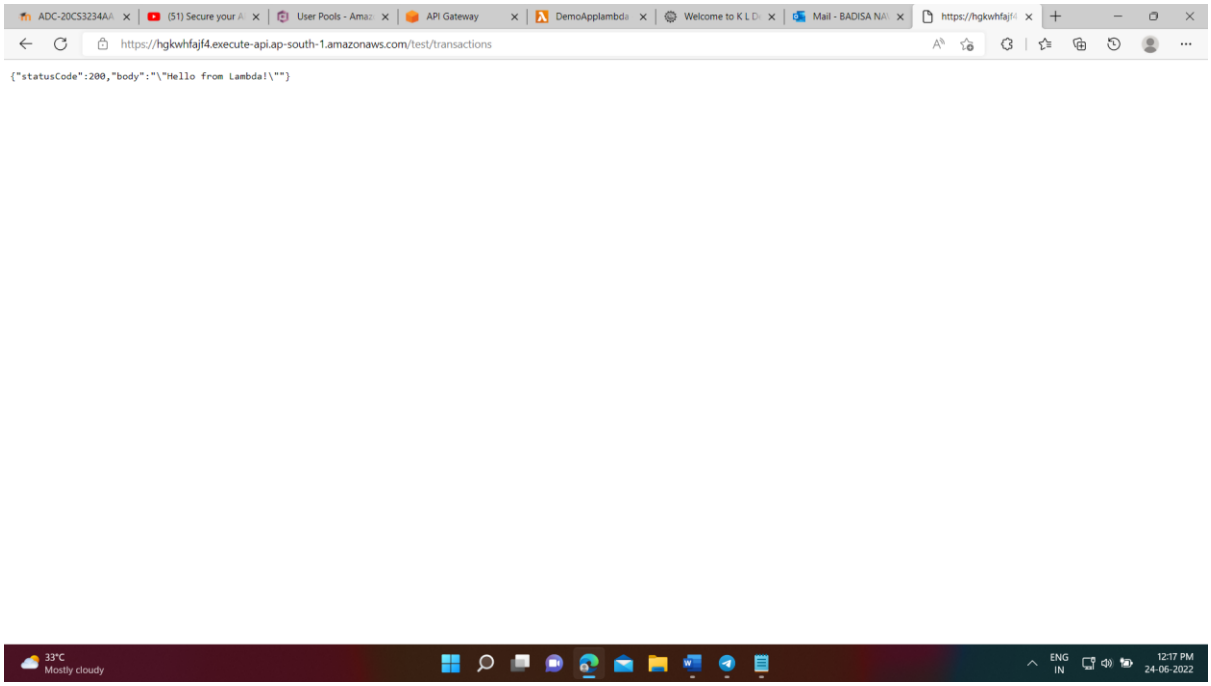


Step6: Deploy API

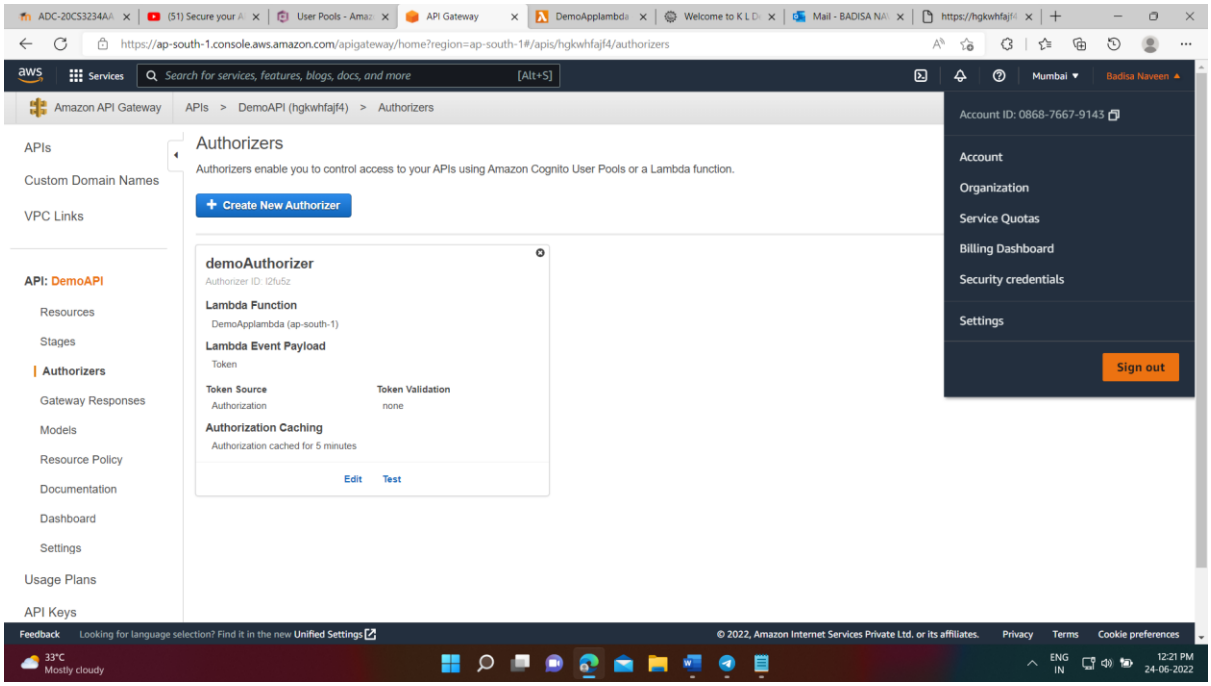


Step7: Open invoke url

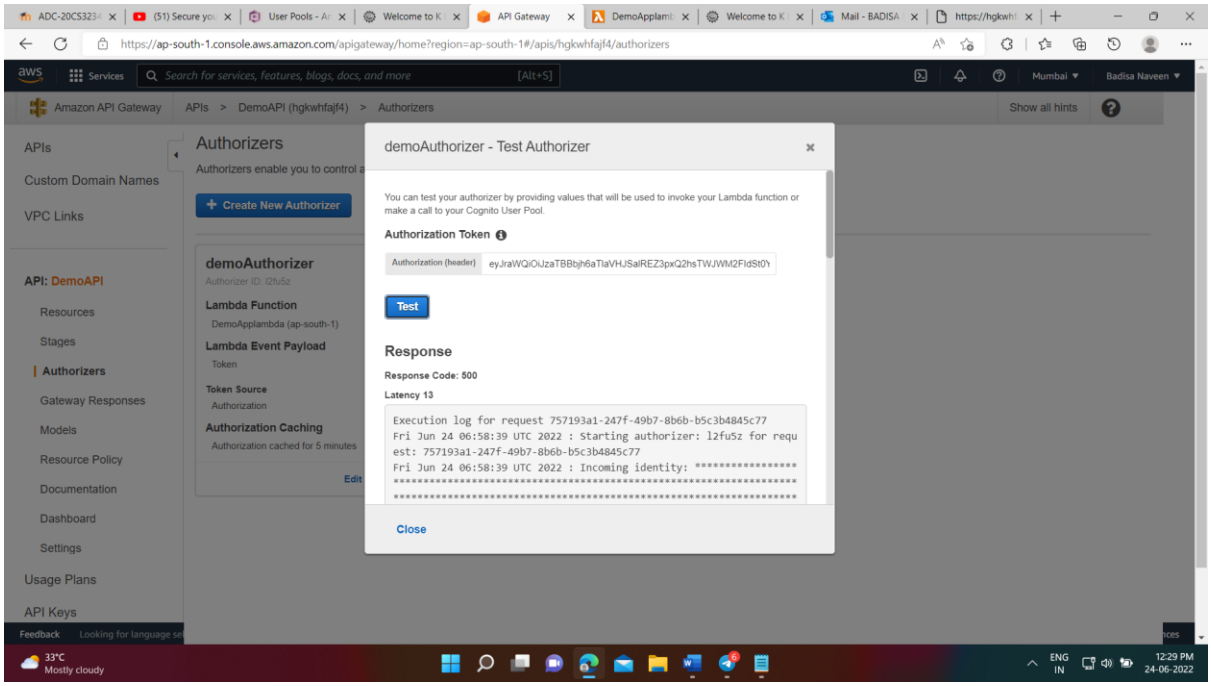
https://hgkwhfajf4.execute-api.ap-south-1.amazonaws.com/test/transactions



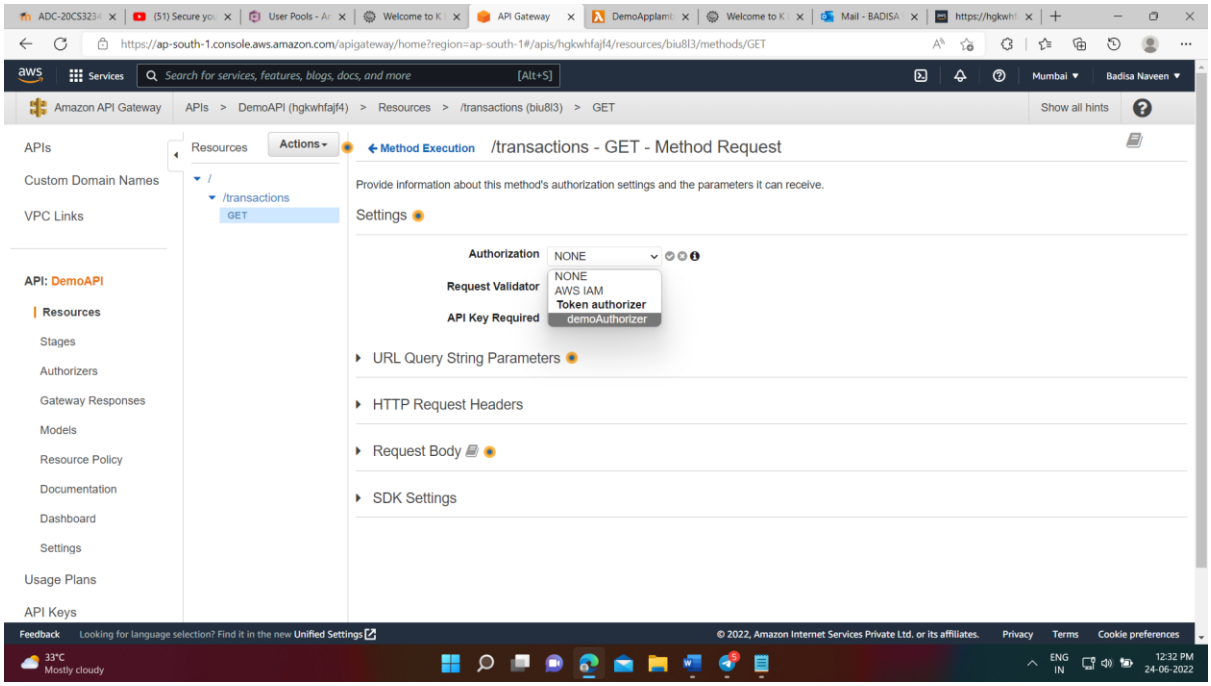
Step8: create Authorization



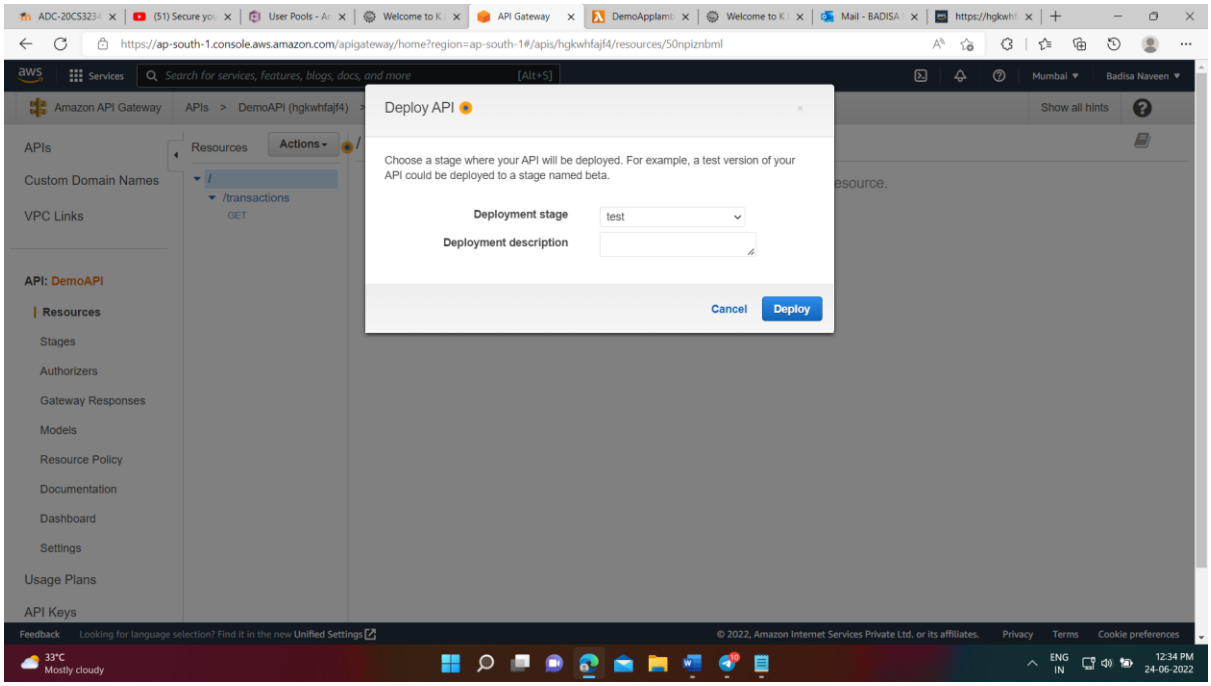
Step9: Test the Authorize token



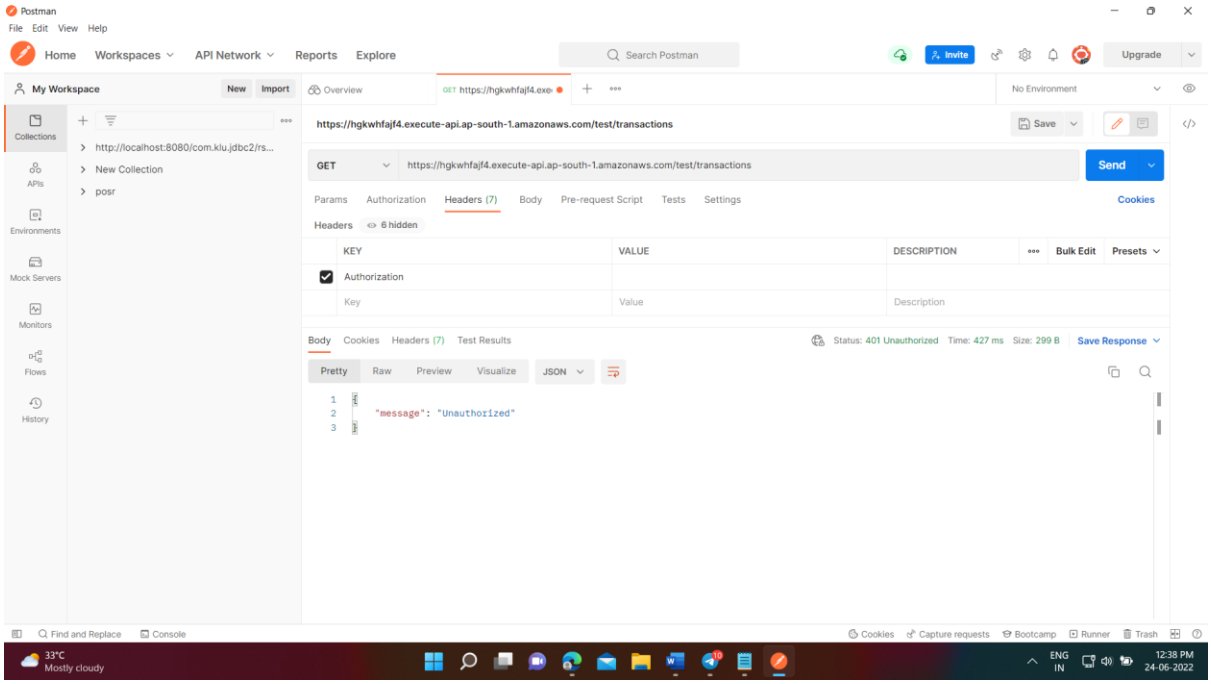
Step10: change the Authorization



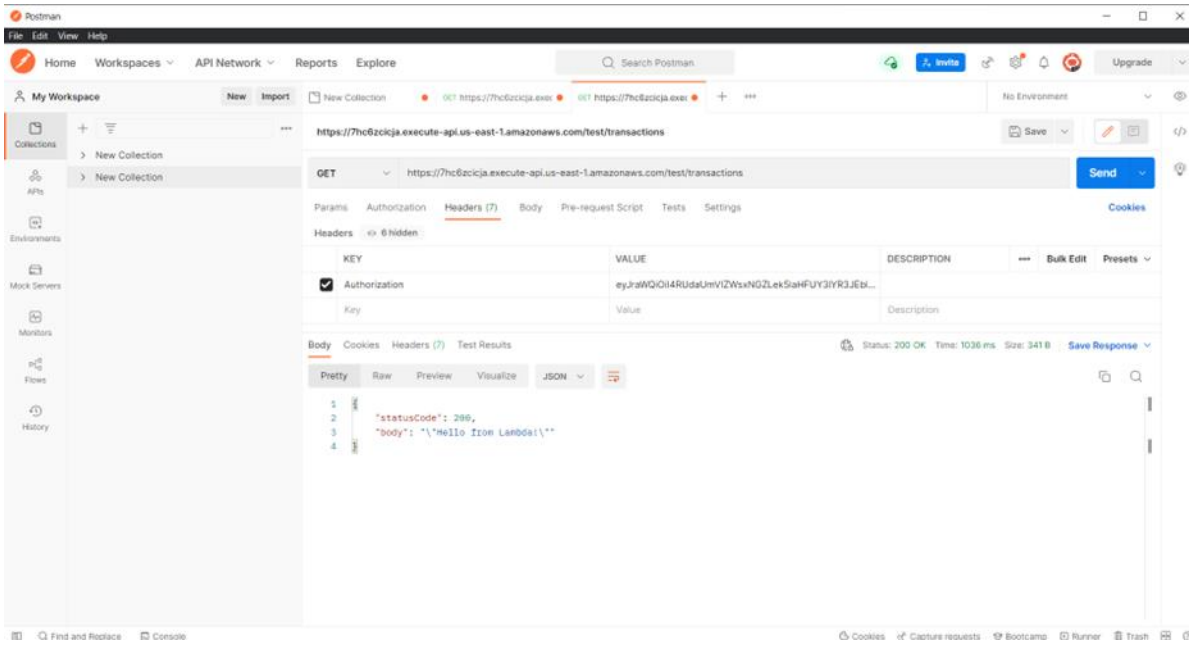
Step11: Deploy API



Step12: Open postman and paste the url



Step13: change the Authorization and provide key



Conclusion:

1. We can write and read data from Dynamodb using AWS lambda
2. AWS Cognito provides Authentication for WebApplications
3. I used Postman for setting header token and to display the output instead of the browser