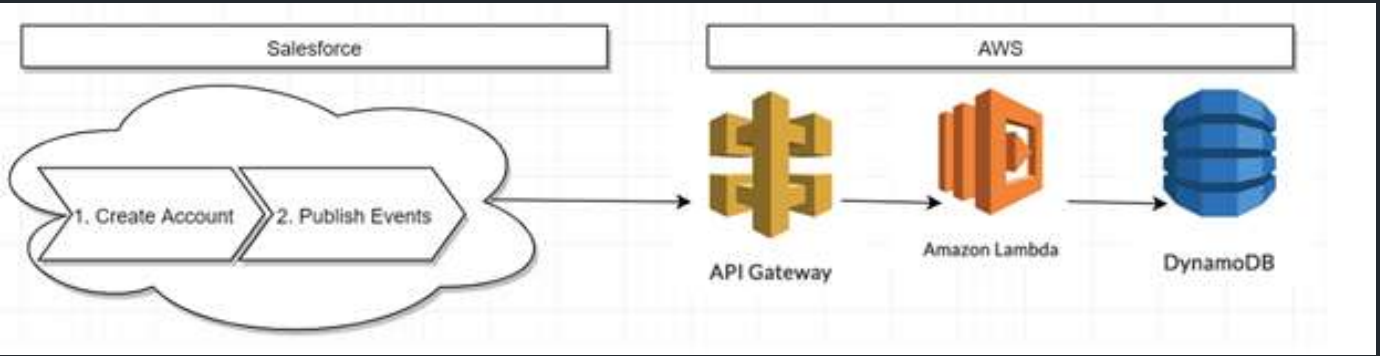
Salesforce AWS integration using API gateway and Lambda

Rajesh S.
Staff Engineer @ PayPal | Ex-Salesforce | Akamai | Morgan Stanley
Published May 27, 2019



S

Prerequisite

1. AWS Free Tier Account ,
2. Salesforce Developer Account
3. Basic to advance knowledge of Salesforce Architecture and Coding Pattern.
4. Basic to Intermediate knowledge AWS lambda, Dynamo DB and server less processing.

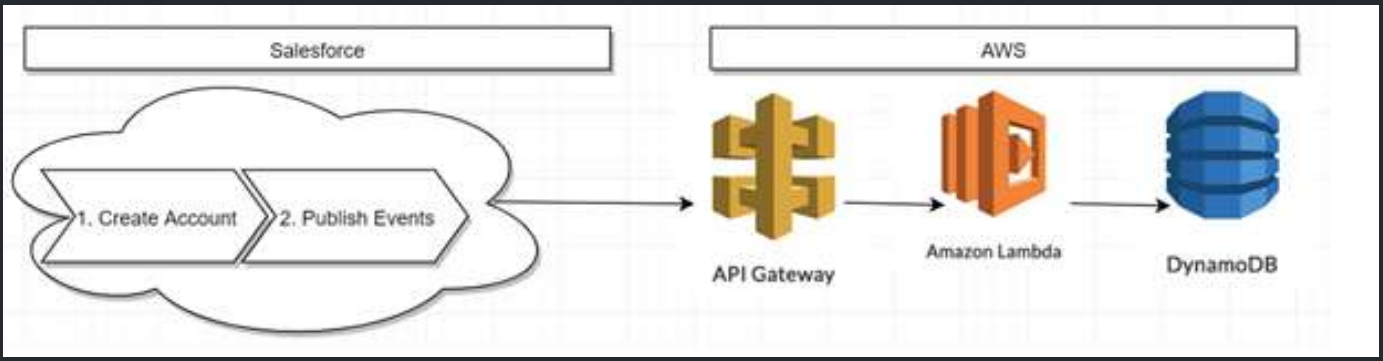
Consideration

1. All AWS service creation has been kept simple .. i.e most part like AWS lambda has been configured to have admin access role .

3. I will have separate article for security consideration b/w salesforce and AWS connection

Use Case Flow

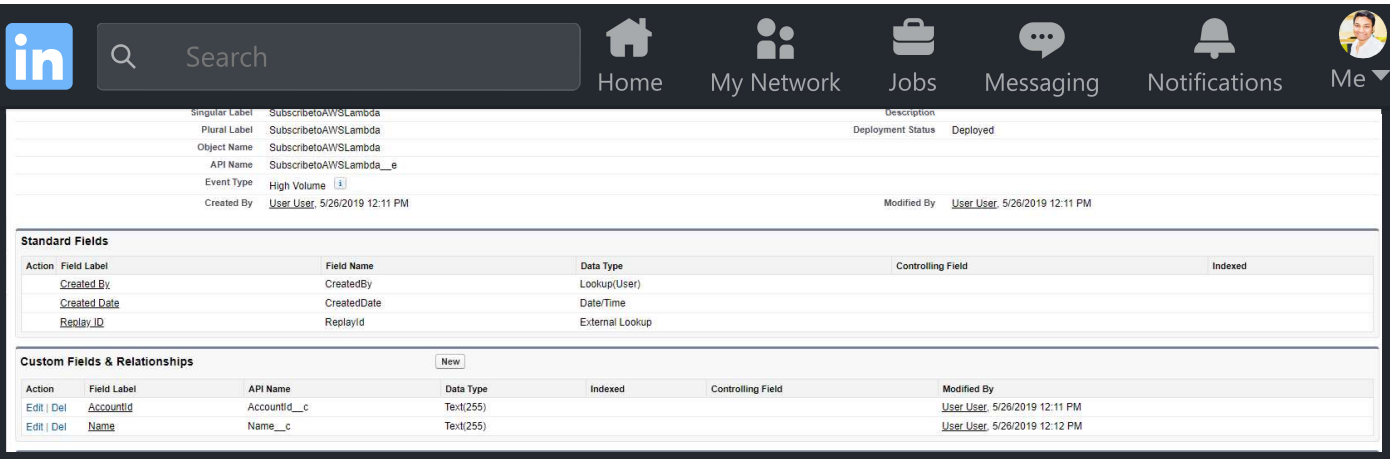
1. Account is created in Salesforce
2. Platform Event is subscribed and in turns it calls amazon gateway and pass data.
3. Amazon Gateway receives the data and transform and pass to amazon lambda.
4. Amazon Lambda has logic to pass this data to dynamo DB and insert



Salesforce Changes

a) Create Platform Even name **"SubscribetoAWSLambda"** (you can name whatever you want)

b) Create two custom field in platform event "AccountId" & Name in above event created.(Refer below image for more detail)



c) Create process builder to publish data in above vents when account is created ,

d) Create below class (Please note this class can be optimised for bulk operations and AWS can be read from custom Metadata/Custom Setting) . Please note that endpoint will be replaced when we create API gateway in later section.

```
public class AWSCallout

{

@future(callout = true)

public static void callAwsgateway(String recordId,String Name)

{

Http http = new Http();

HttpRequest request = new HttpRequest();

//Please set the endpoint of API gateway which we are going to create in later section

request.setEndpoint('awsapigateway');

request.setMethod('POST');
```

```
// Set the body as a JSON object

request.setBody('{"id": "'+recordId+'","Name": "'+Name+'"}');

HttpResponse response = http.send(request);

System.debug('response==='+response.getBody());

// Parse the JSON response

if (response.getStatusCode() != 201) {

    System.debug('The status code returned was not expected: ' +

        response.getStatusCode() + ' ' + response.getStatus());

} else {

    System.debug(response.getBody());



}

}



}
```

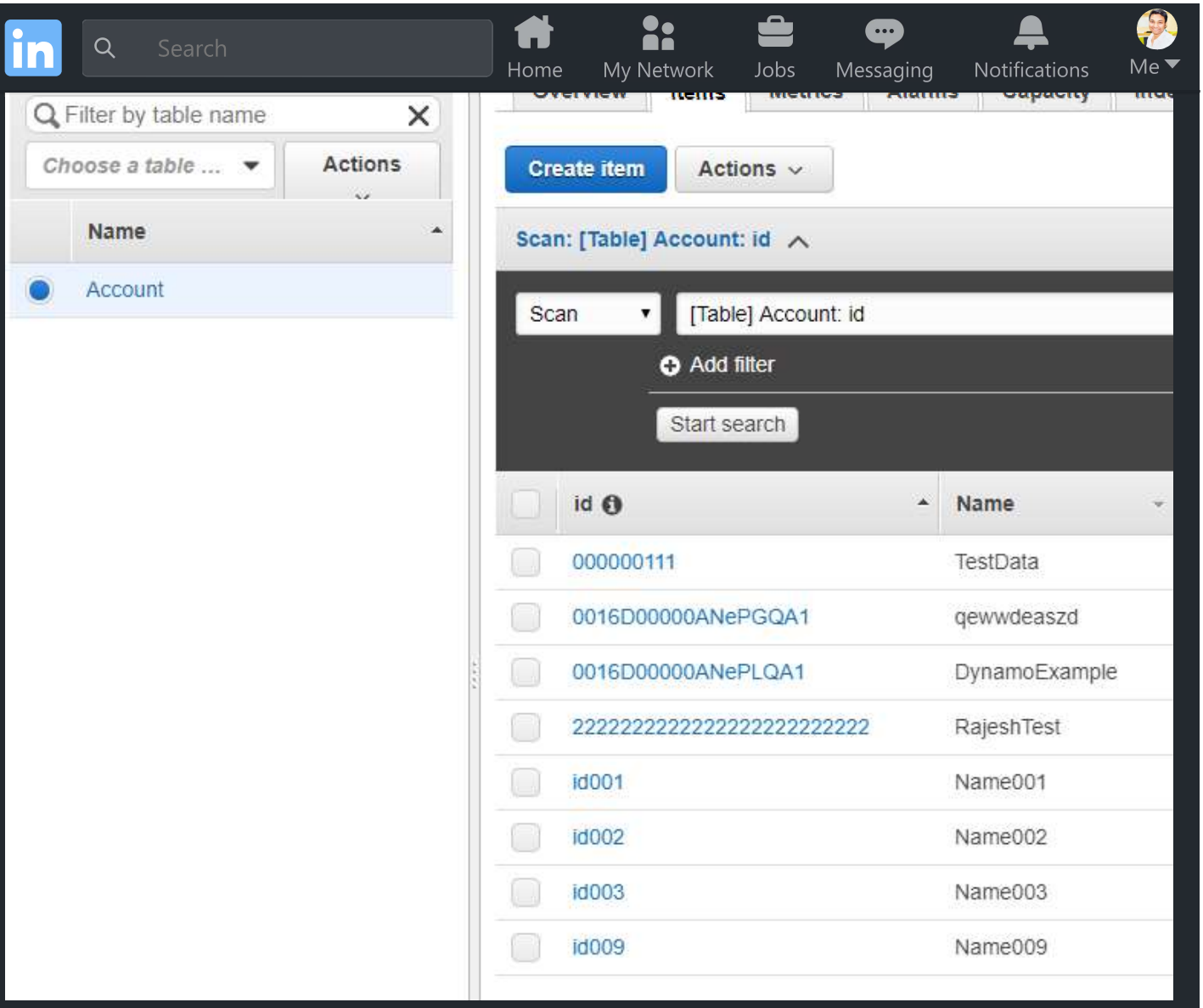
e) Create Subscriber on platform Event (basically platform event Trigger)

```
for (SubscribetoAWSLambda__e saws : Trigger.new)

{

  AWSCallout.callAwsgateway(saws.AccountId__c,saws.Name__c);

}



}
```

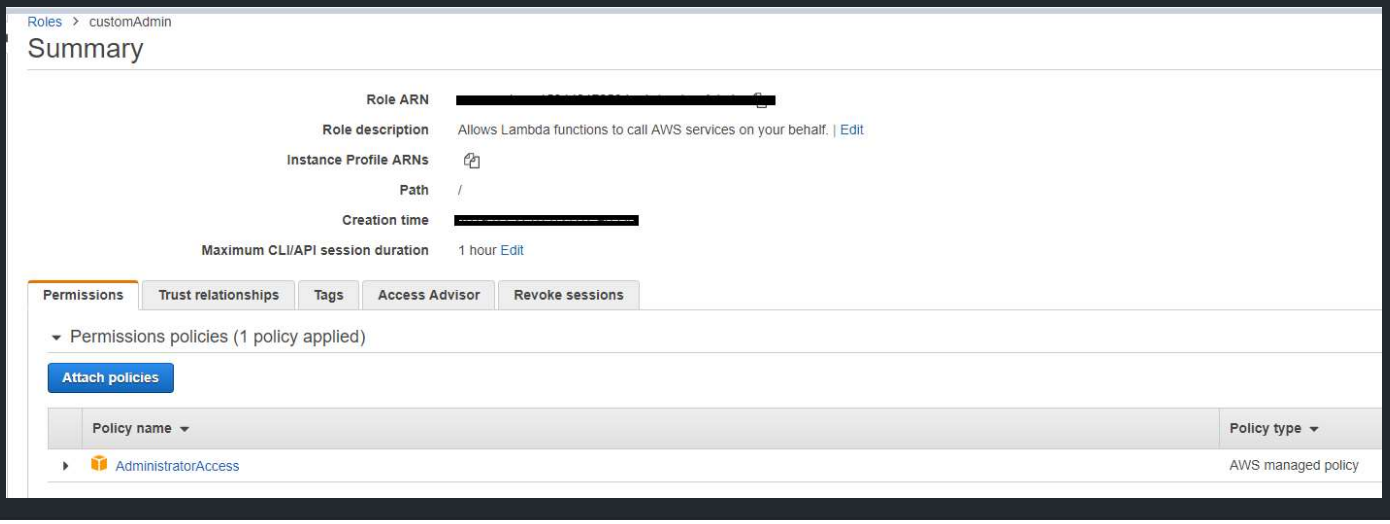
f) Do not forget to add the actual end point mentioned in Step c) to remote site setting.

AWS Changes

> Create below dynamodb table with "id" as Primary partition key

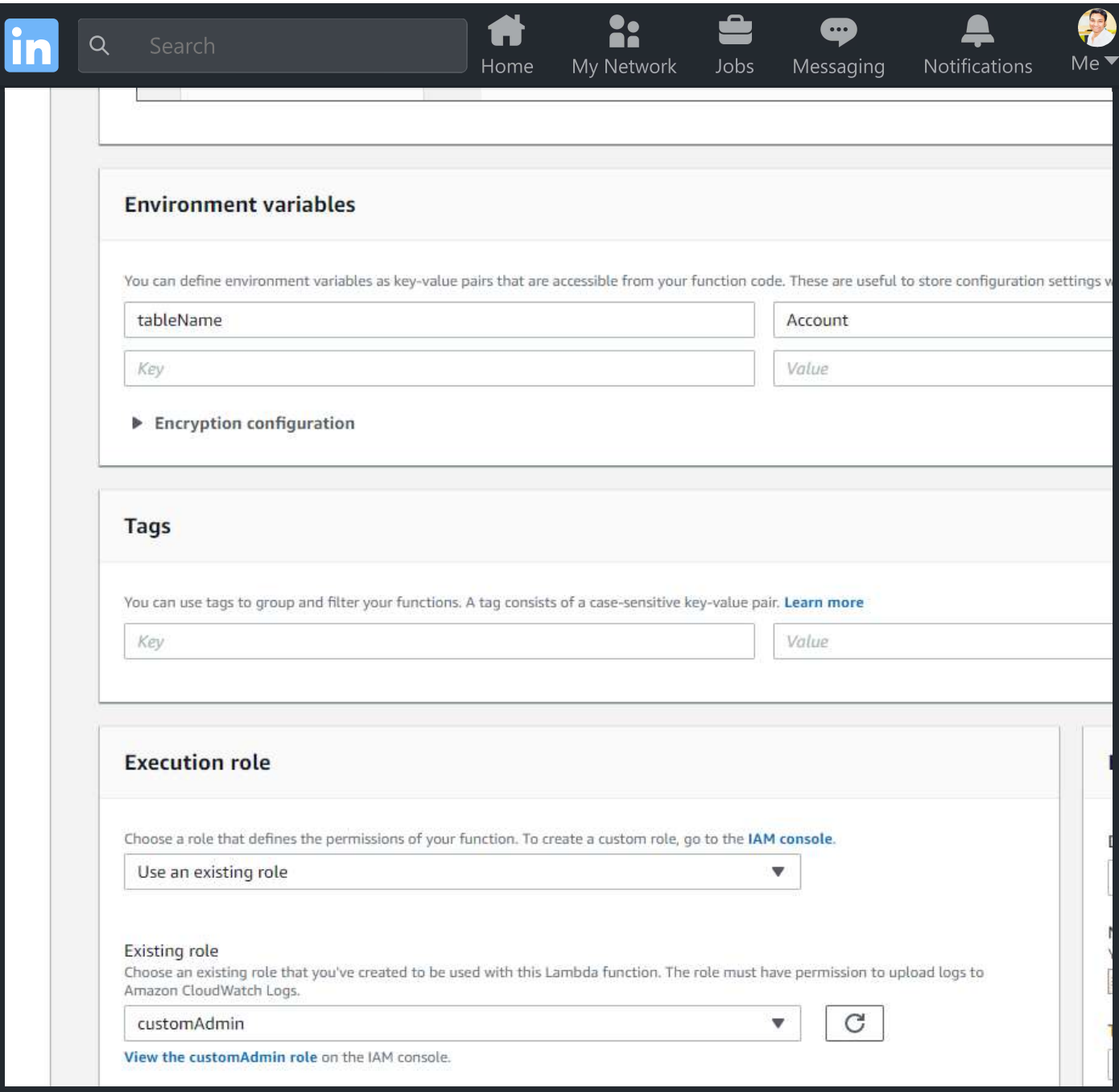


> Login to AWS and go to IAM and create custom role with Administrator access policy .I have used this role for AWS lambda , please note in real time situation we should not give admin access security should be limited here I have used just for demo purpose to avoid any access related issue .

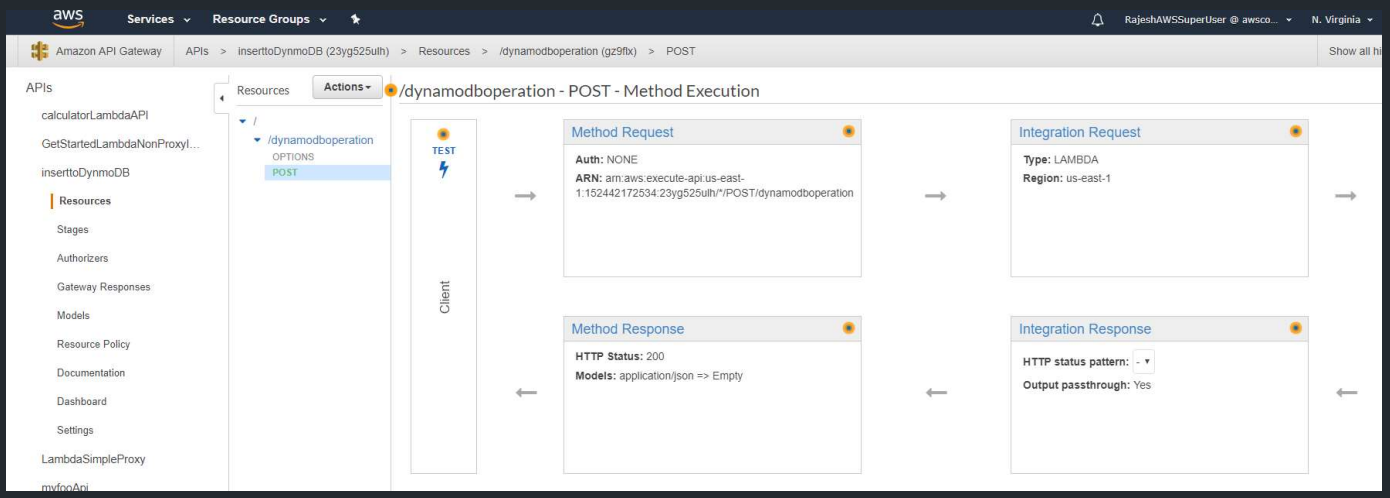


```
const AWS = require('aws-sdk');
const dynamodb = new AWS.DynamoDB({apiVersion: '2012-08-10'});

exports.handler = (event, context, callback) => {
    dynamodb.putItem({
        TableName: process.env.tableName,
        Item: {
            "id": {
                S: event.id
            },
            "Name": {
                S: event.Name
            }
        }
    }, function(err, data) {
        if (err) {
            console.log(err, err.stack);
            callback(null, {
                statusCode: '500',
                body: err
            });
        } else {
            callback(null, {
                statusCode: '200',
                body: 'Hello ' + event.id + event.Name + '!'
            });
        }
    })
};
```

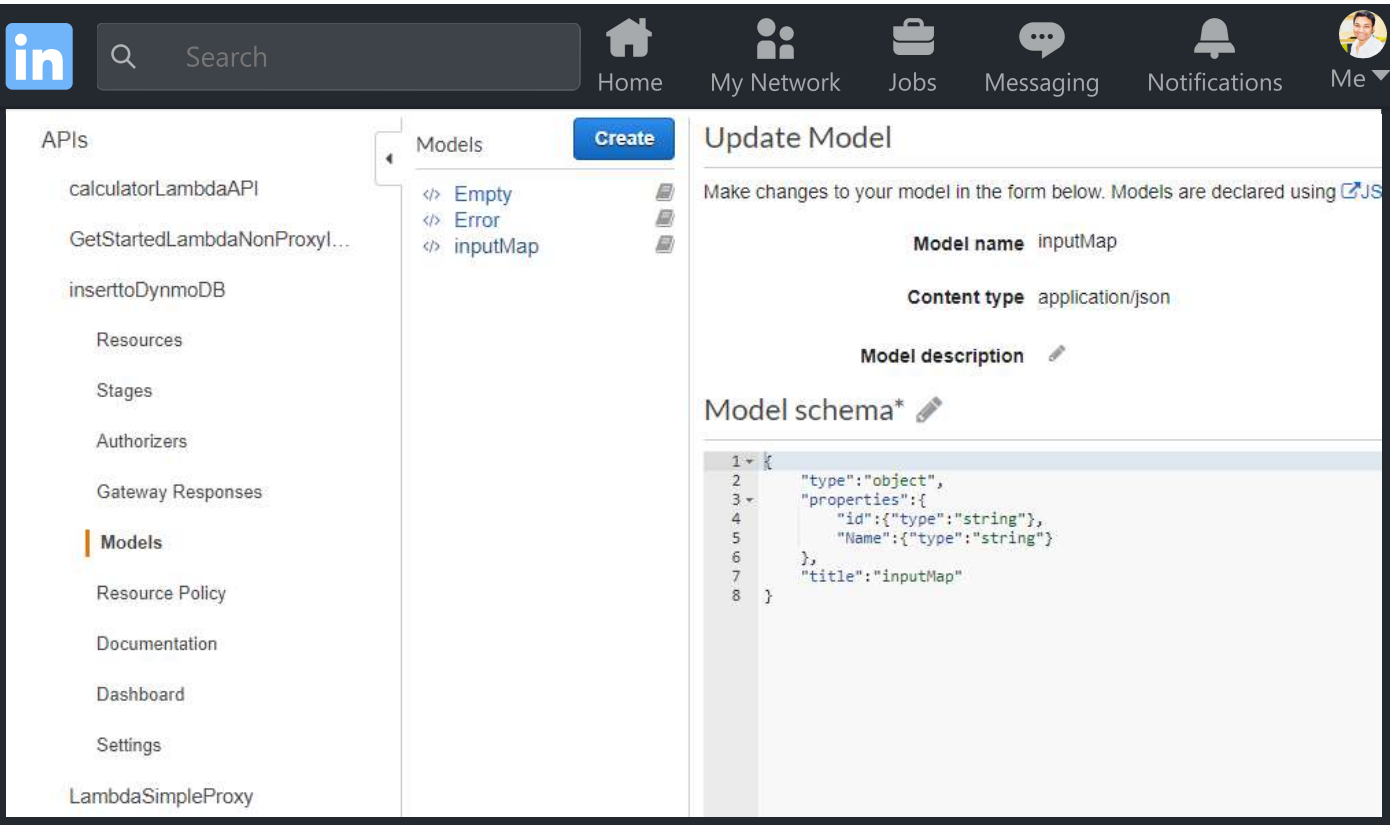
Below image shows the setup for lambda function



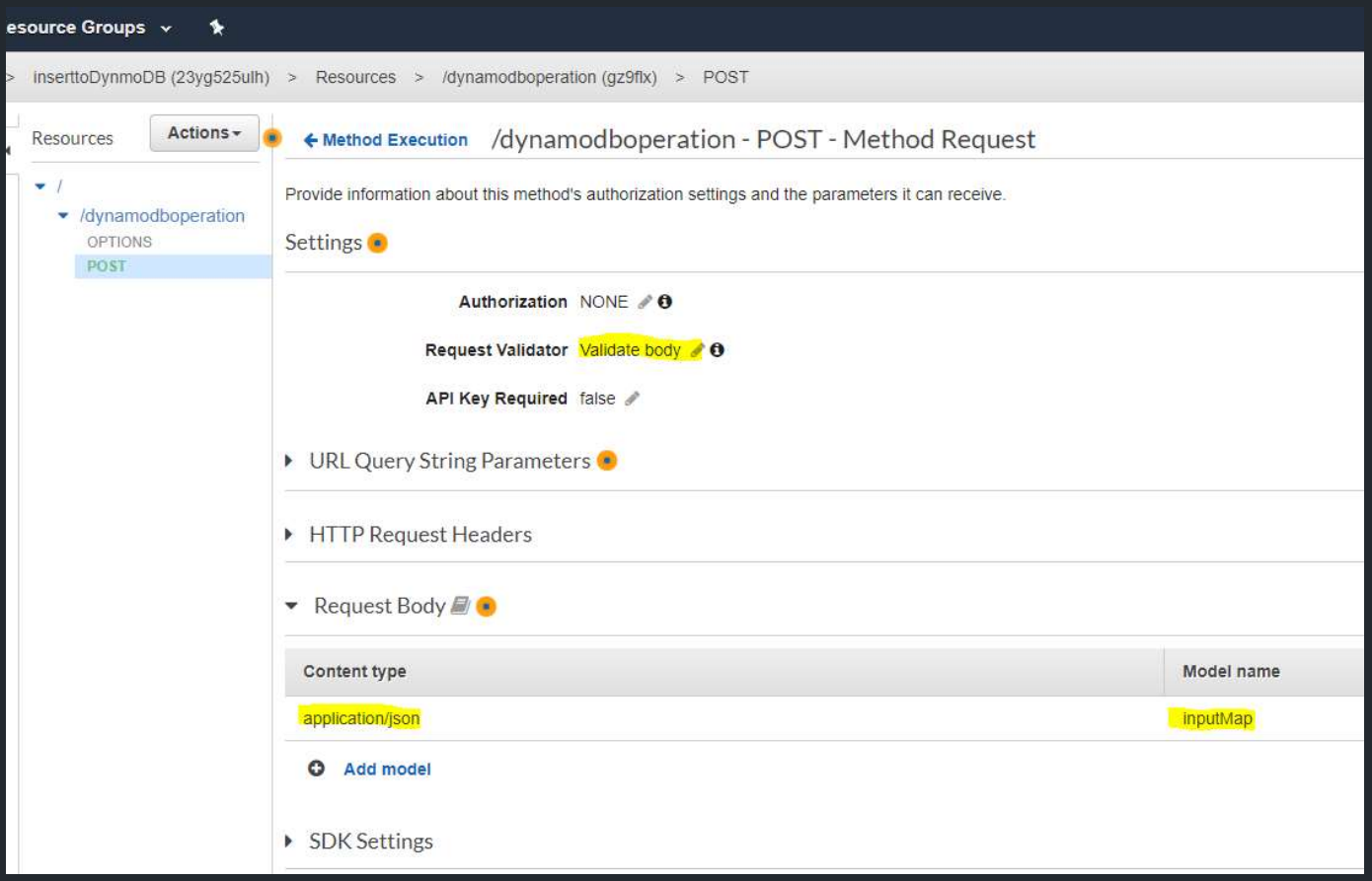
> Create below API gateway



> Create below model inside the gateway



> Configure post method as below .



> Deploy API gateway to stage (you can stage name whatever you want) I have named test.

> Test the functionality by creating account in Salesforce which will flow to dynamo db table created earlier .

Note : Code used here is not optimised do not use for production this is just for demo purpose for production code need to handle bulk scenario without any hard coding.

Report this

26 · 8 Comments

Like Comment Share

Caterina Murdock-Thorum 2y ···

I'm interested in data exchange between AWS S3 (data lake) and AWS Lambda and Salesforce, but recipient of data, is Salesforce. The data would be queried by agent on need basis on a lookup tab in a Salesforce DB. I found your article helpful though I need the reverse flow.

Like Reply

Siddharth Raj 3y ···

great job Rajesh..... really helpful

Like Reply

Anirban Goswami 3y ···

truly fine stuff

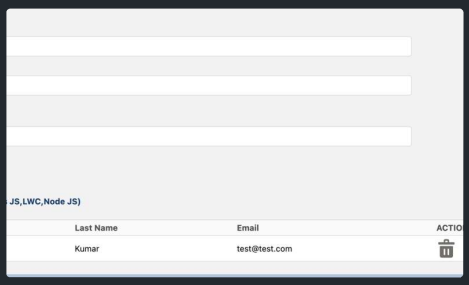
Like Reply

Vivek Mehta 3y ···

Like Reply

See more comments

More articles by this author



LWC With Express,Node,MongoDB...
Jul 5, 2021

Published on LinkedIn

Bulk API in Salesforce using JAVA Client
Jun 28, 2019

Published on LinkedIn

'Javascript Promises' in Lightning Web Compon
Jun 3, 2019