USE CASE – 3

Scenario: -

Let's take the case of an ecommerce application, users would be creating an account by providing their profile details. In this process the users table is populated with new items in the DynamoDB table (similar to rows in the RDBMS).

How should we start: -

When a new user details are populated in the users table, as part of the workflow we would like to get a notification via email or notify some other application. In this use case, once the front-end web application inserts an item into the DynamoDB table, the Lambda function will be automatically called which sends an email for the sake of notification. The functionality in the Lambda can be simply replaced with the code to notify some other application (like marketing application) via SQS.

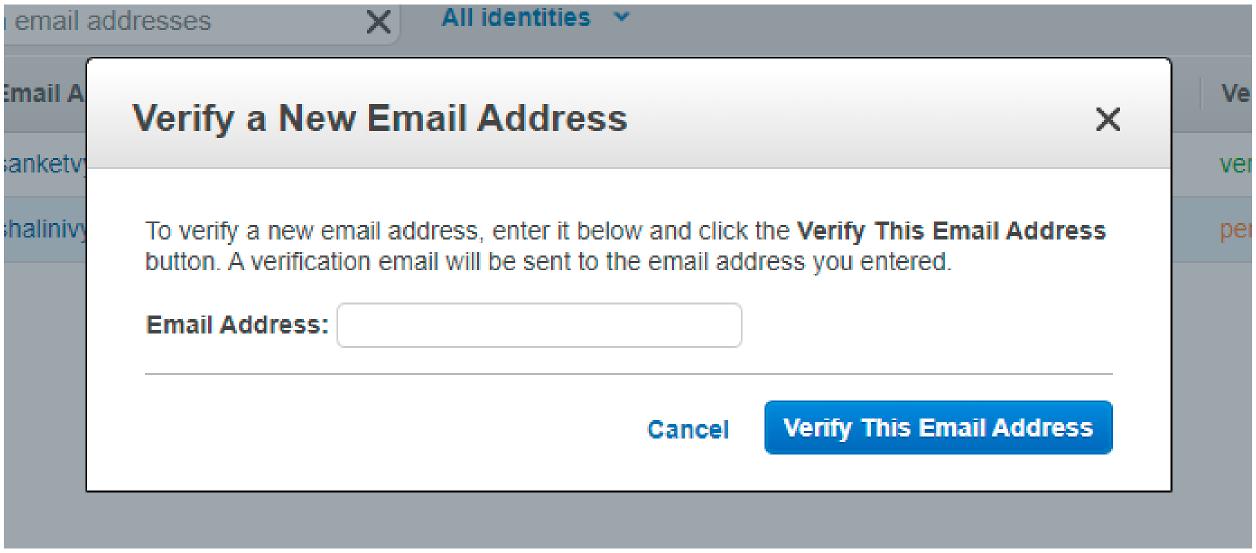
Services we will work with: -

DynamoDB, Lambda, IAM and SES

Follow Along: -

- **Step: - 1**

  Go to the SES Console, click on "Verify a New Email Address".



Verify a New Email Address

To verify a new email address, enter it below and click the **Verify This Email Address** button. A verification email will be sent to the email address you entered.

Email Address:

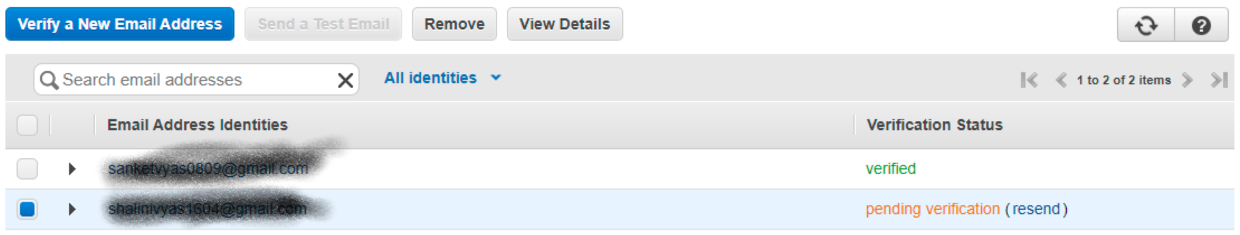
USE CASE – 3

- **Step: - 2**

  Enter the email address and click on "Verify This Email Address".

- **Step: - 3**

  An email will be sent to this address with a link for the sake of verification. Click on Close.



Verify a New Email Address | Send a Test Email | Remove | View Details

Search email addresses | All identities | 1 to 2 of 2 items

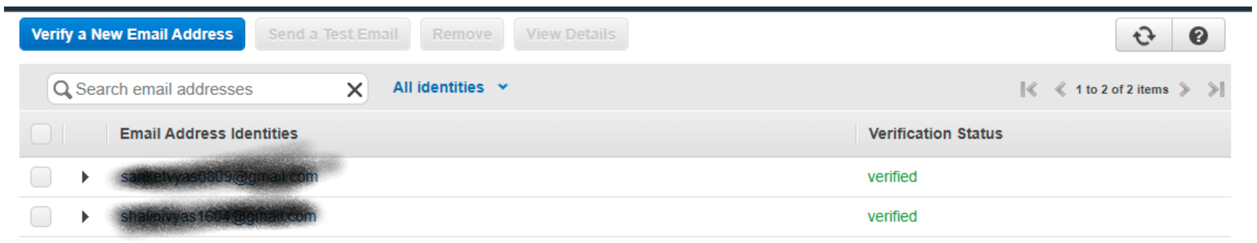
Email Address Identities	Verification Status
[redacted]	verified
[redacted]	pending verification (resend)

- **Step: -4**

  Initially the email will be in a "pending verification" status and after clicking on the link in the email, the status of the email will change to "verified". This is to make sure that the SES service is not used for spamming.

- **Step: - 5**

  Follow the same steps and verify another email address. Once of the email would be acting as the sender and the other as the receiver.



Verify a New Email Address | Send a Test Email | Remove | View Details

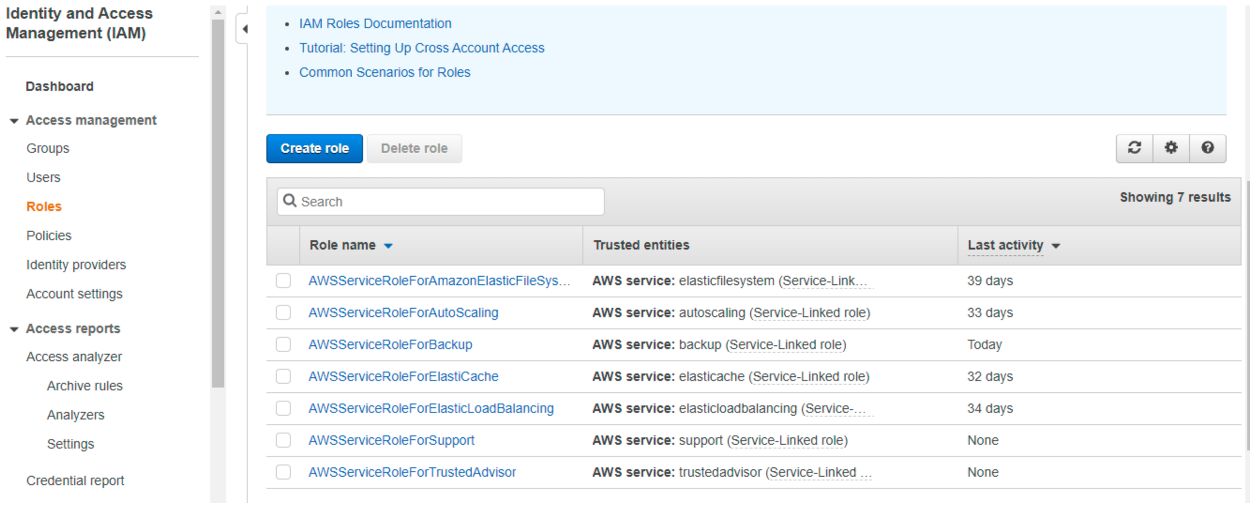
Search email addresses | All identities | 1 to 2 of 2 items

Email Address Identities	Verification Status
[redacted]	verified
[redacted]	verified

USE CASE – 3

- **Step: - 6**

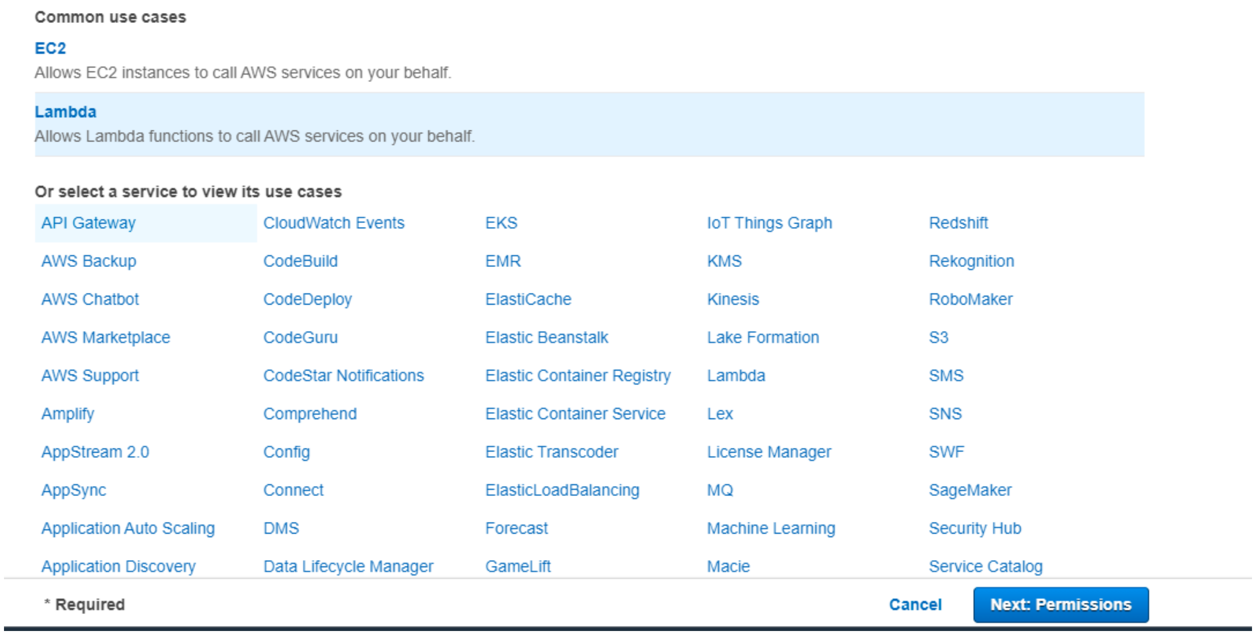
  Now it's time to create an IAM Role for Lambda. Go to the IAM Management Console, click on Roles and click on "Create role".



Role name	Trusted entities	Last activity
AWSServiceRoleForAmazonElasticFileSys...	**AWS service:** elasticfilesystem (Service-Link...	39 days
AWSServiceRoleForAutoScaling	**AWS service:** autoscaling (Service-Linked role)	33 days
AWSServiceRoleForBackup	**AWS service:** backup (Service-Linked role)	Today
AWSServiceRoleForElastiCache	**AWS service:** elasticache (Service-Linked role)	32 days
AWSServiceRoleForElasticLoadBalancing	**AWS service:** elasticloadbalancing (Service-...	34 days
AWSServiceRoleForSupport	**AWS service:** support (Service-Linked role)	None
AWSServiceRoleForTrustedAdvisor	**AWS service:** trustedadvisor (Service-Linked ...	None

- **Step: - 7**

  Select Lambda as the service which is going to use this Role. Click on "Next: Permissions".



Common use cases

EC2
Allows EC2 instances to call AWS services on your behalf.

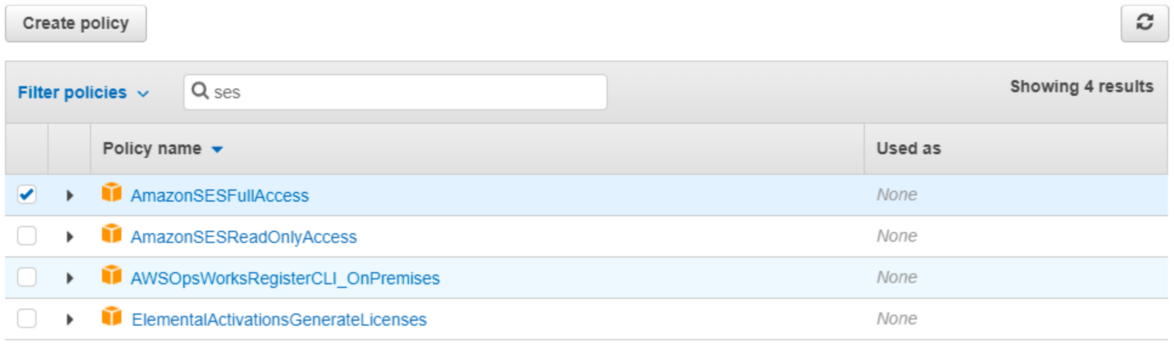
Lambda
Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

API Gateway	CloudWatch Events	EKS	IoT Things Graph	Redshift
AWS Backup	CodeBuild	EMR	KMS	Rekognition
AWS Chatbot	CodeDeploy	ElastiCache	Kinesis	RoboMaker
AWS Marketplace	CodeGuru	Elastic Beanstalk	Lake Formation	S3
AWS Support	CodeStar Notifications	Elastic Container Registry	Lambda	SMS
Amplify	Comprehend	Elastic Container Service	Lex	SNS
AppStream 2.0	Config	Elastic Transcoder	License Manager	SWF
AppSync	Connect	ElasticLoadBalancing	MQ	SageMaker
Application Auto Scaling	DMS	Forecast	Machine Learning	Security Hub
Application Discovery	Data Lifecycle Manager	GameLift	Macie	Service Catalog

- **Step: - 8**

  Select the AWSLambdaDynamoDBExecutionRole and AmazonSESFullAccess policies and click on "Next : Tags".



Create policy

Filter policies | ses | Showing 4 results

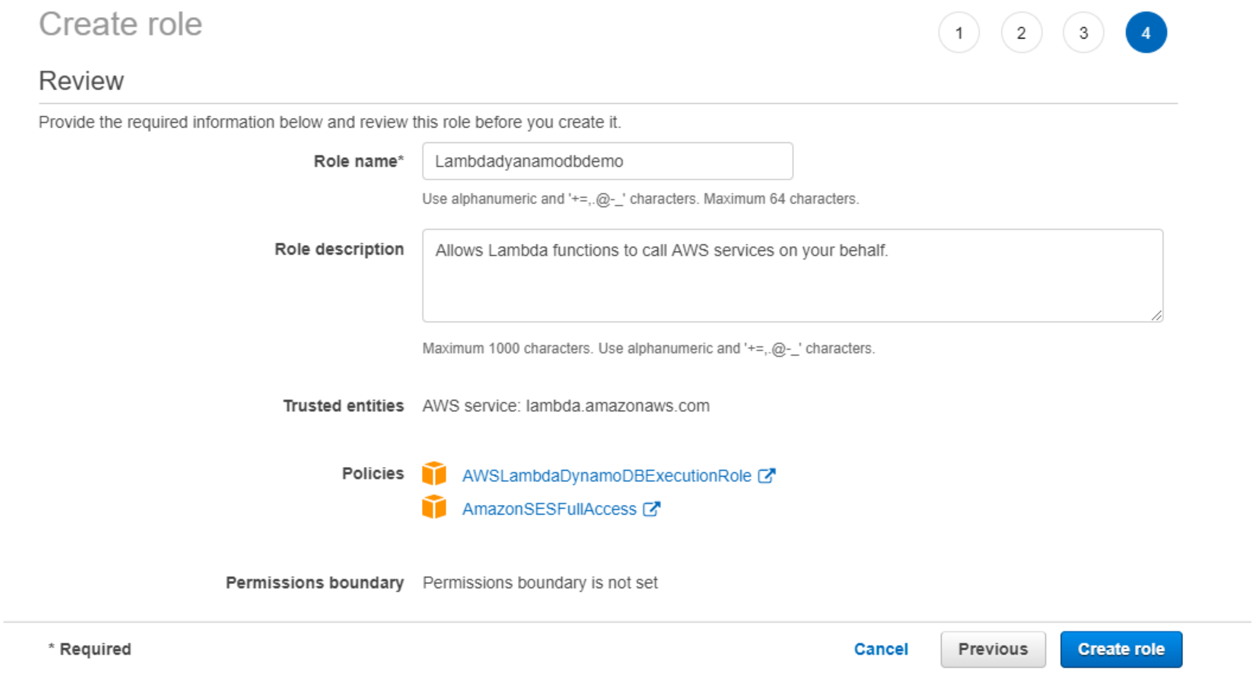
		Policy name	Used as
✓	▸	AmazonSESFullAccess	None
	▸	AmazonSESReadOnlyAccess	None
	▸	AWSOpsWorksRegisterCLI_OnPremises	None
	▸	ElementalActivationsGenerateLicenses	None

- **Step: - 9**

  Tags are optional. Simply, click on "Next : Review".

- **Step: - 10**

  Give the role a name and click on "Create role".



Create role

1 2 3 4

Review

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

Role name* Lambdadyanamodbdemo

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

Role description Allows Lambda functions to call AWS services on your behalf.

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

Trusted entities AWS service: lambda.amazonaws.com

Policies AWSLambdaDynamoDBExecutionRole
AmazonSESFullAccess

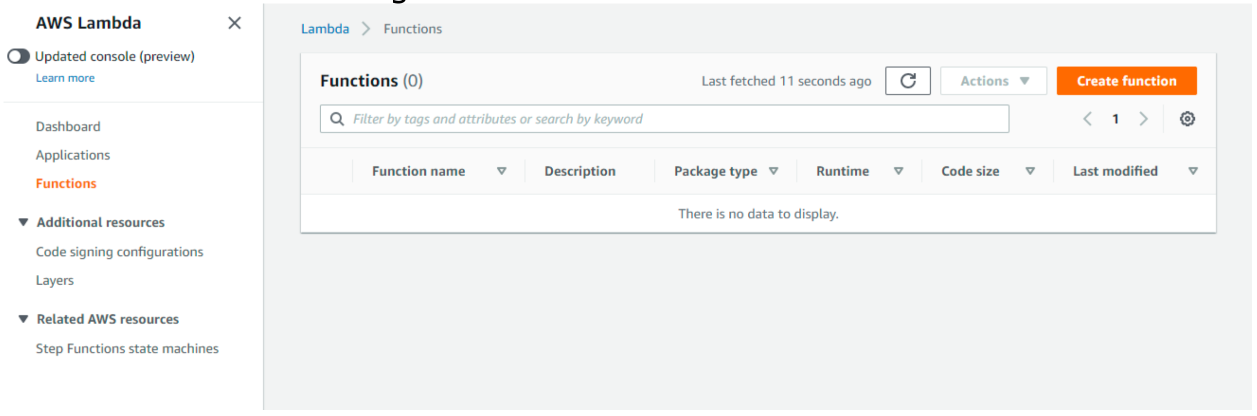
Permissions boundary Permissions boundary is not set

* Required

USE CASE – 3

- **Step: - 11**

  Go to the Lambda Management Console and click on "Create function".



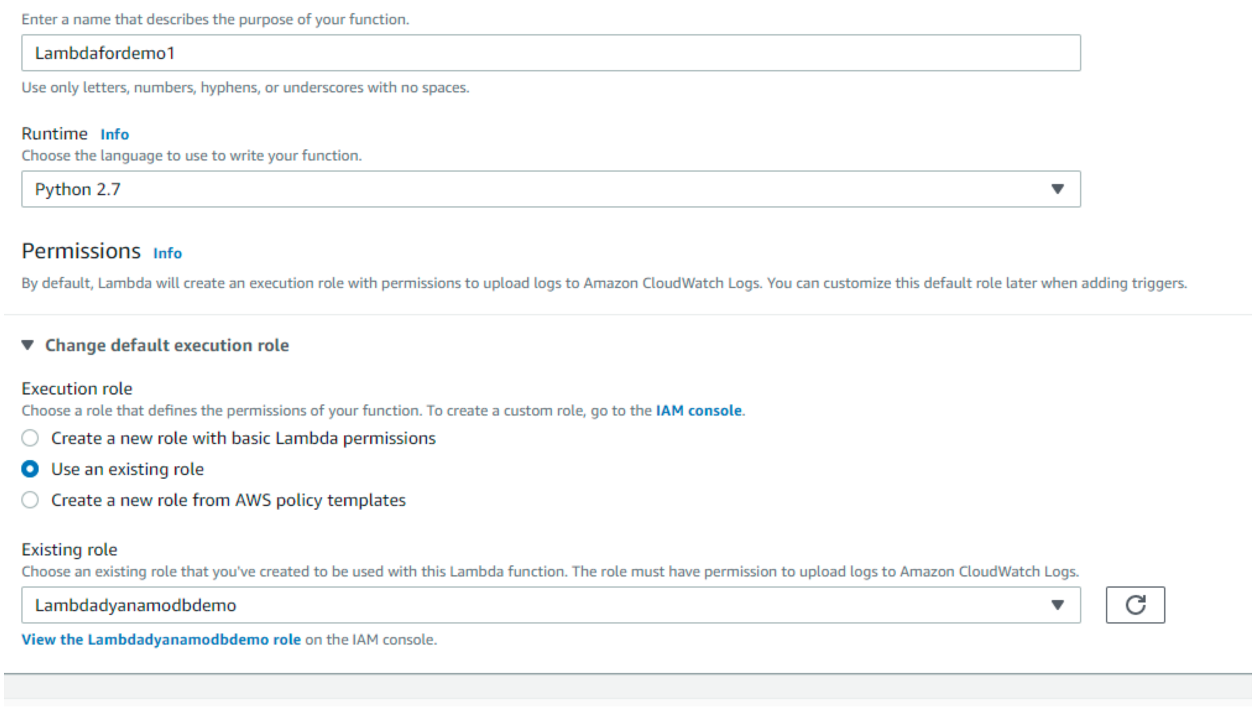
AWS Lambda
Updated console (preview)
Learn more
Dashboard
Applications
Functions
Additional resources
Code signing configurations
Layers
Related AWS resources
Step Functions state machines
Lambda > Functions
Functions (0)
Last fetched 11 seconds ago
Actions
Create function
Filter by tags and attributes or search by keyword
1

Function name	Description	Package type	Runtime	Code size	Last modified

There is no data to display.

- **Step: - 12**

  Enter the Function name, select the role and NodeJS 10.x or Python2.7 and select the role created earlier. The Python and NodeJS code for sending emails via SES has been mentioned in the next sections. Use either of these languages for the Lambda as per your comfort.



Enter a name that describes the purpose of your function.

Lambdafordemo1

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info

Choose the language to use to write your function.

Python 2.7

Permissions Info

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

- ○ Create a new role with basic Lambda permissions
- ● Use an existing role
- ○ Create a new role from AWS policy templates

Existing role

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

Lambdadyanamodbdemo

View the Lambdadyanamodbdemo role on the IAM console.

USE CASE – 3

- **Step: - 13**

  Copy the below NodeJS code. Make sure to replace the from and the to address with the email address which have been verified earlier. And finally click on Save.

- **Step: - 14**

  The IAM user will be created with the mentioned details. Note down the URL or the link mentioned in this screen. This is link to be used to login as an IAM user.

```
var aws = require('aws-sdk');
var ses = new aws.SES({region: 'us-east-1'});
exports.handler = function(event, context) {
        console.log("Incoming: ", event);
        // var output = querystring.parse(event);
        var eParams = {
                Destination: {
                        ToAddresses: ["abc@gmail.com"]//give the email ID which is
                        verified by SES
                },
                Message: {
                        Body: {Text: {
                                Data: "Hurray a new user has been created !!!!"
                                }
                                },
                Subject: { Data: "New User"}
        },
        Source: "xyz@gmail.com" //give the email ID which is verified by SES
};
console.log('===SENDING EMAIL===');
var email = ses.sendEmail(eParams, function(err, data){
        if(err) console.log(err);
        else {
                console.log("===EMAIL SENT===");
                console.log(data);
                console.log("EMAIL CODE END");
                console.log('EMAIL: ', email);
                context.succeed(event);
                }
        });
};
```

USE CASE – 3

- **Step: - 15**

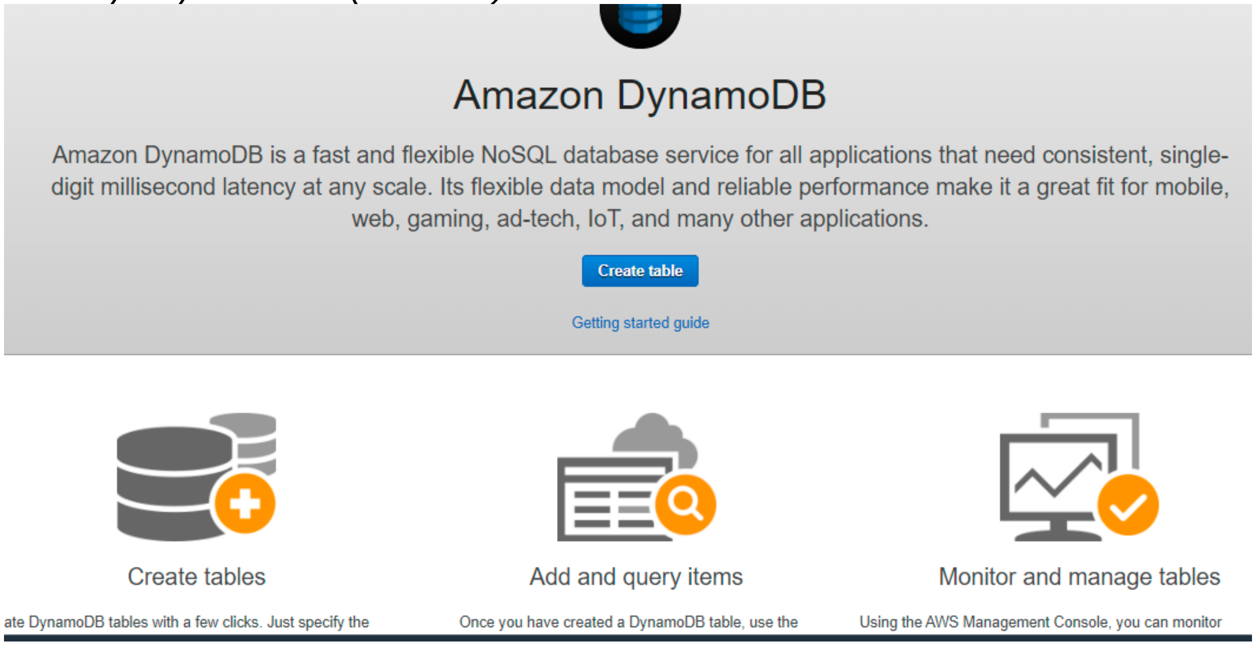
Copy the below Python code. Make sure to replace the from and the to address with the email address which have been verified earlier. And finally click on Save.

```
import boto3
from botocore.exceptions import ClientError
print('Loading function')
SENDER = "xyz@gmail.com"
RECIPIENT = "abc@gmail.com"
SUBJECT = "New User"
BODY_TEXT = "Hurray a new user has been created !!!!"
AWS_REGION = "us-east-1"
CHARSET = "UTF-8"
def lambda_handler(event, context):
        client = boto3.client('ses',region_name=AWS_REGION)
        try:
                response = client.send_email(
                        Destination={
                                'ToAddresses': [
                                                RECIPIENT,
                ],
},
Message={
        'Body': {
                'Text': {
                        'Charset': CHARSET,
                        'Data': BODY_TEXT,
                },
            },
                'Subject': {
                        'Charset': CHARSET,
                        'Data': SUBJECT,
                },
        },
        Source=SENDER
)
# Error handling
except ClientError as e:
        print(e.response['Error']['Message'])
else:
        print("Email sent! Message ID:"),
        print(response['MessageId'])
```

- **Step: - 16**

  Go to the DynamoDB Management Console and click on Create table. Enter the below details and click on Create. The table would be created in a few seconds.

  Table name – users
  Primary key – userid (Number)



Amazon DynamoDB

Amazon DynamoDB is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale. Its flexible data model and reliable performance make it a great fit for mobile, web, gaming, ad-tech, IoT, and many other applications.

ate DynamoDB tables with a few clicks. Just specify the

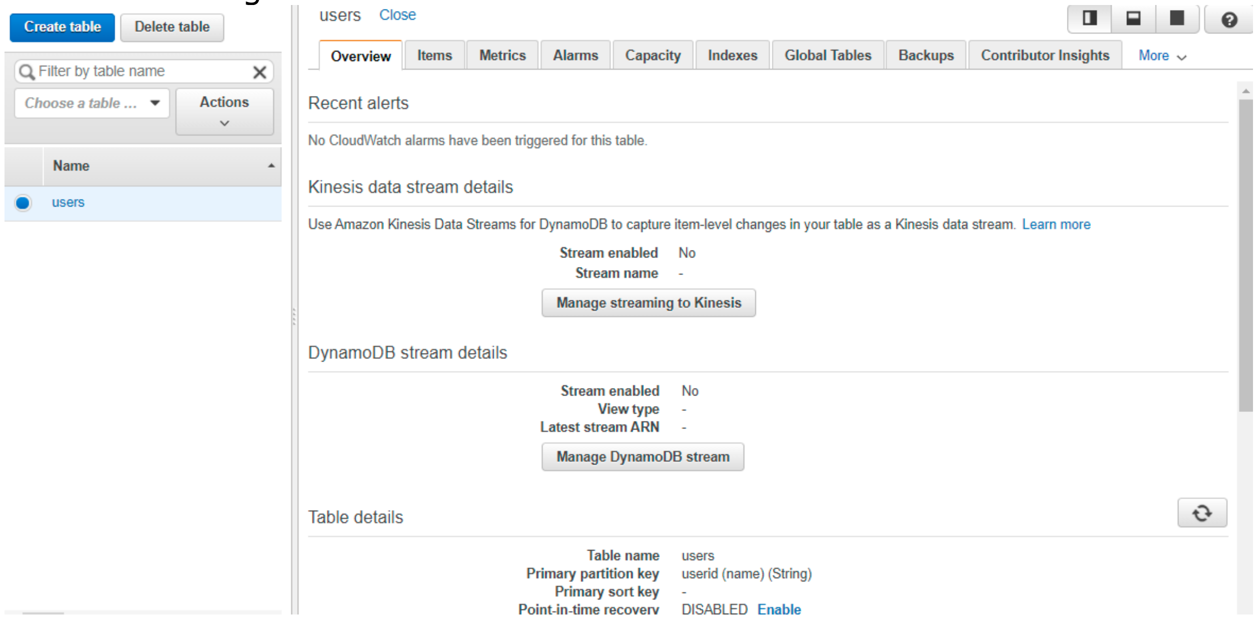
Once you have created a DynamoDB table, use the

Using the AWS Management Console, you can monitor

USE CASE – 3

- **Step: -17**

  Click on "Manage Streams".



Recent alerts

No CloudWatch alarms have been triggered for this table.

Kinesis data stream details

Use Amazon Kinesis Data Streams for DynamoDB to capture item-level changes in your table as a Kinesis data stream. Learn more

Stream enabled	No
Stream name	-

DynamoDB stream details

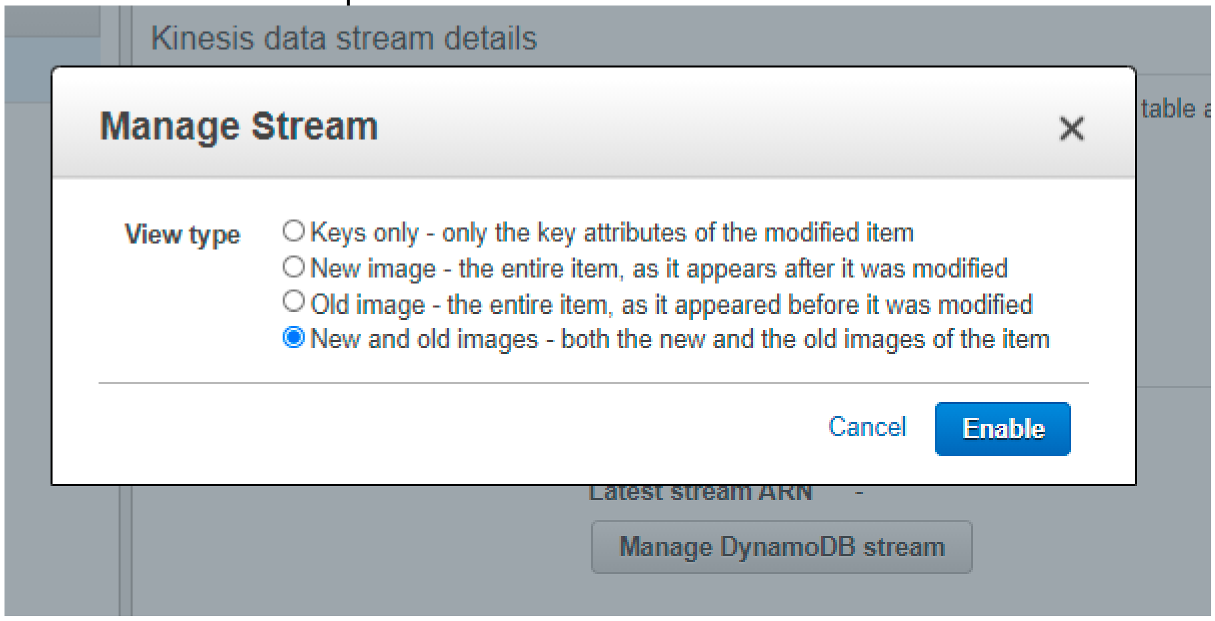
Stream enabled	No
View type	-
Latest stream ARN	-

Table details

Table name	users
Primary partition key	userid (name) (String)
Primary sort key	-
Point-in-time recovery	DISABLED Enable

- **Step: - 18**

  Go with the default options and click on "Enable".



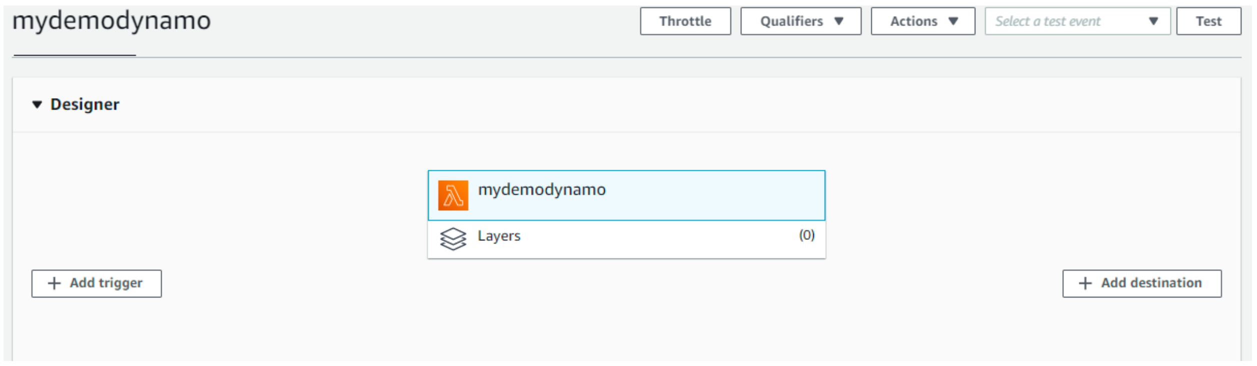
Manage Stream

View type
- Keys only - only the key attributes of the modified item
- New image - the entire item, as it appears after it was modified
- Old image - the entire item, as it appeared before it was modified
- [✓ New and old images - both the new and the old images of the item]

USE CASE – 3

- **Step: -19**

  Go back to the Lambda Management Console and click on "Add trigger".

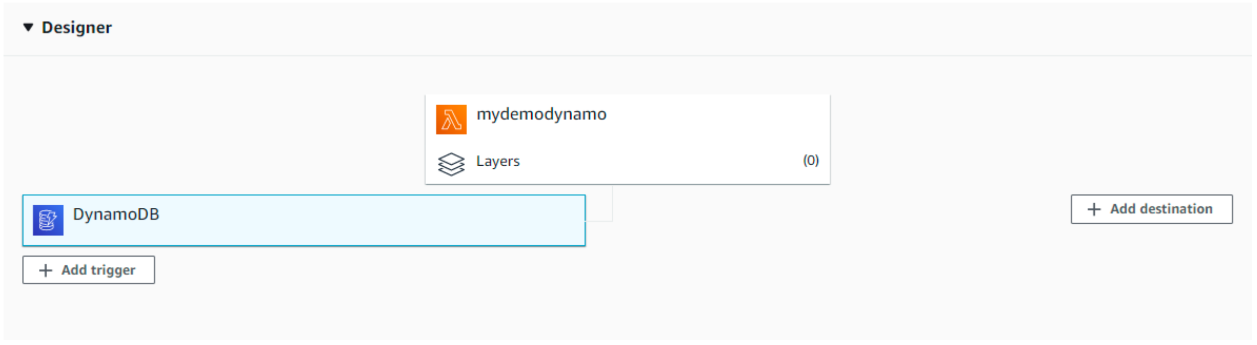


- **Step: - 20**

  Select the DynamoDB service and select the "users" table created earlier. Go with the rest of the default options and click on "Add". This integrates the DynamoDB table with the Lambda function via Triggers.

- **Step: - 21**

  The DynamoDB Trigger should be added to the Lambda function as shown below.



- **Step: -22**
  Create an IAM Role for EC2 with the AmazonDynamoDBFullAccess Policy attached to it.

USE CASE – 3

- **Step: - 23**

  Create an EC2 instance with the below details and connect to it.

  ```
  t2.micro
   Ubuntu OS
   Security Group with Port 22/Inbound allowed
  ```

- **Step: -24**

  Once connected to the EC2 instance execute the below commands. These commands will install Python, pip, boto3 (AWS Python SDK).

  ```
  #become root
          sudo su

  #get the list of softwares
          apt-get update

  #install python and pip
          apt-get install python2.7 python-pip -y

  #install python aws sdk
          pip install boto3
          exit
  mkdir .aws
          echo -e "[default]\nregion=us-east-1" > .aws/config
  ```

- **Step: - 25**

  Create a file called dynamodb-put.py with the below content. Finally execute the "python dynamodb-put.py" to execute the python program which inserts an item in the DynamoDB table. Here we are trying to mimic an application inserting an item in the DynamoDB table.

  ```
  import boto3
  if __name__ == '__main__':
  dynamodb = boto3.resource('dynamodb')
  table = dynamodb.Table('users')
  response = table.put_item(
  Item={
  'userid': 123,
  'name': "Praveen Sripati",
  ```

```
'city': "Hyderabad",
'country': "India"
}
)
        print("Put user succeeded:")
```

- **Step: - 26**

  The item should be inserted into the DynamoDB table as shown below. This should automatically trigger the Lambda function, which will send an email via AWS SES service.

- **Step: - 27**

  Check your email and there should be email from the SES service which has been triggered by the Lambda function. The function in the Lambda function can be replaced with any code for integration with other applications via AWS SQS Service.