



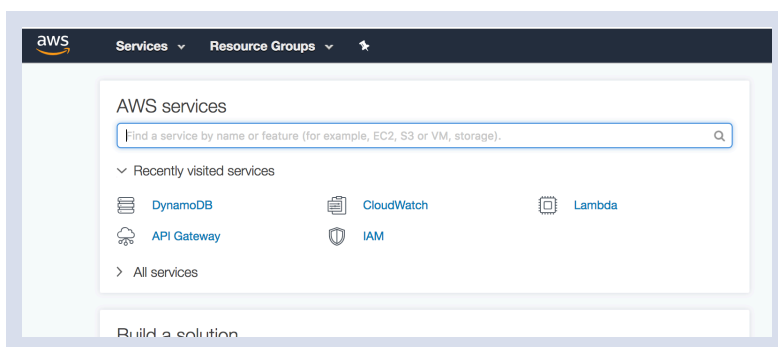
# Short Guide to DynamoDB + Lambda



## Creating a Dynamo DB (NoSQL) database and connection to a Lambda function (approx 5 minutes).

### Let's begin by creating a table within DynamoDB:

Navigate to AWS and type into the search input 'DynamoDB' or click the link on screen.



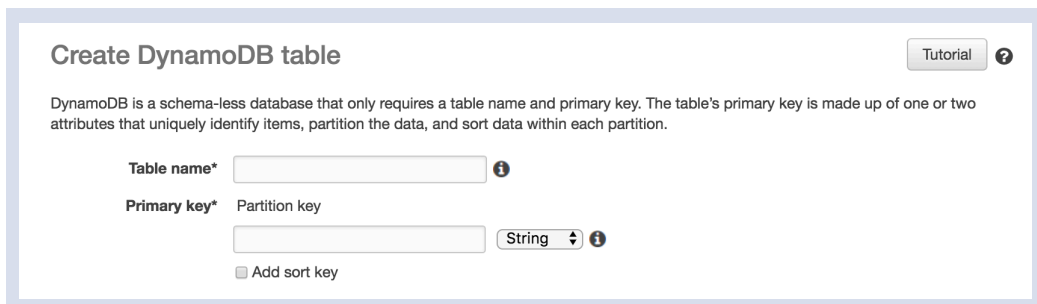
### To create the table:

Click the button from the centre section of the screen 'Create Table' as shown in the image below.



### Completing the details:

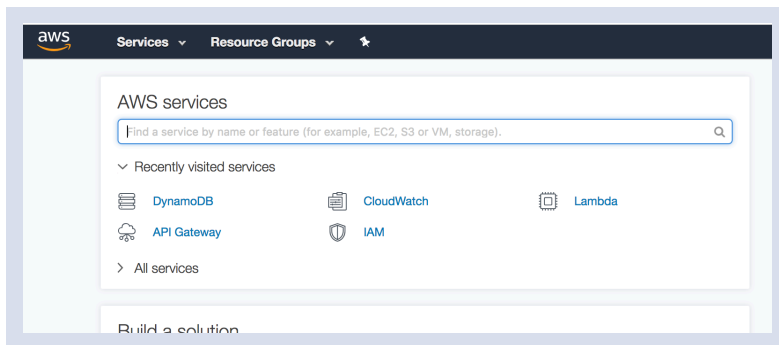
Include a table name and primary key.



**Ensure that the primary key's data type is correct before you select to 'Create Table'**

## Let's connect the database to a Lambda function.

Navigate to AWS and type into the search input 'Lambda' or click the link on screen.



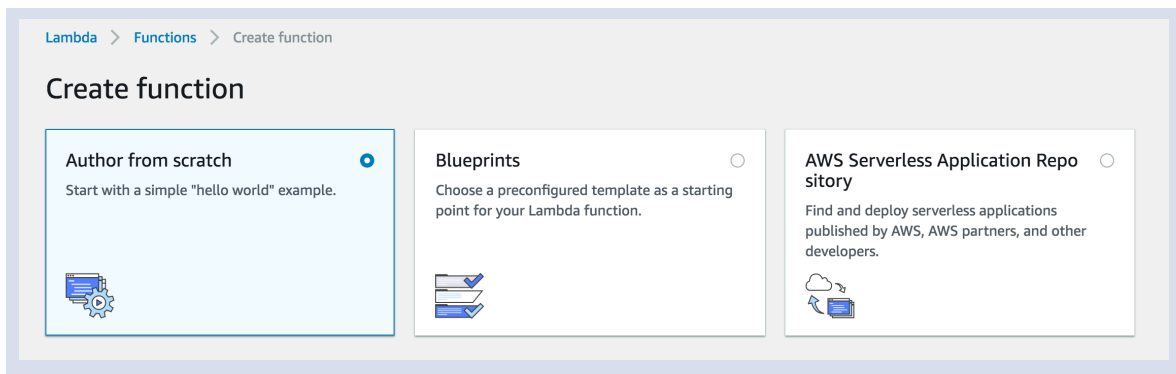
## Create a new Lambda function.

Click to create function.



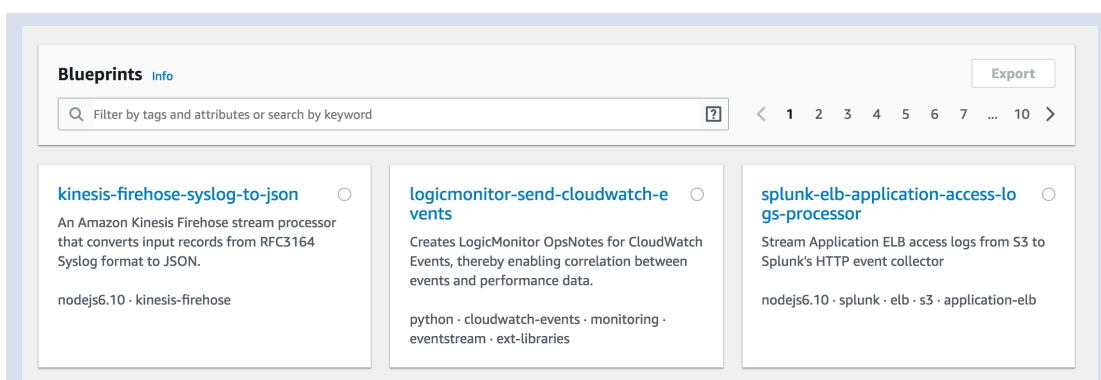
## Create new or template

Author from scratch will create the simplest version of Lambda with code to trigger 'hello world'. Alternatively you can select from Blueprints or AWS repository to kick start your project.



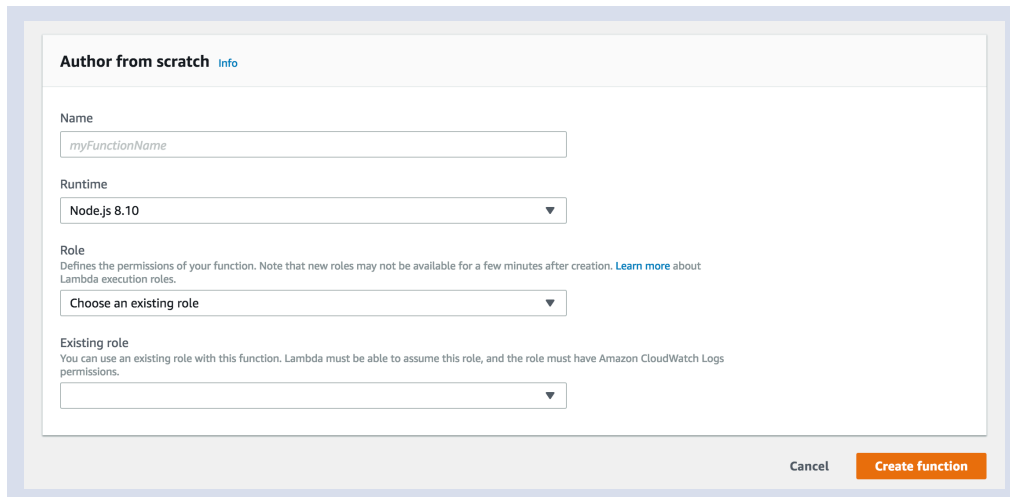
## Blueprints / AWS Repository

You can find Blueprints via the search input or select displayed lambda function blueprints.



## From scratch

This is recommended if you want to get a feel for Lambda and have no specific project in mind. Complete the form and click the 'Create function' button at the bottom.



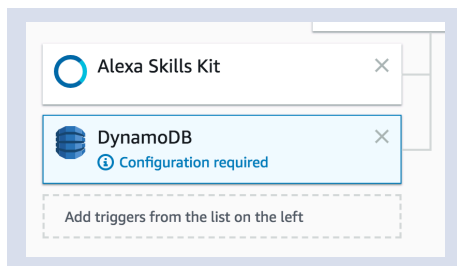
The screenshot shows the 'Author from scratch' form in the AWS Lambda console. The form is titled 'Author from scratch' with an 'Info' link. It contains the following fields:

- Name:** A text input field with the placeholder text 'myFunctionName'.
- Runtime:** A dropdown menu with 'Node.js 8.10' selected.
- Role:** A dropdown menu with 'Choose an existing role' selected. Below it, there is a link to 'Learn more about Lambda execution roles'.
- Existing role:** A text input field with a placeholder text 'You can use an existing role with this function. Lambda must be able to assume this role, and the role must have Amazon CloudWatch Logs permissions.'

At the bottom right of the form, there are two buttons: 'Cancel' and 'Create function'.

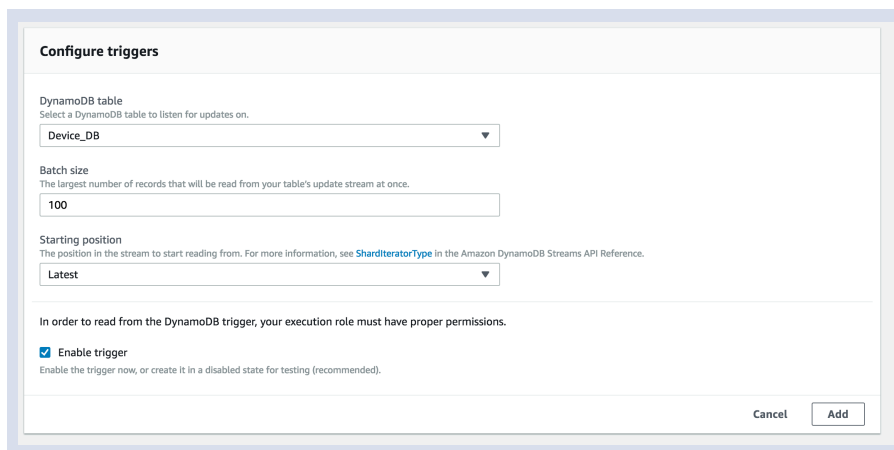
## Inside the Lambda

Lets connect the Database. First select 'DynamoDB' from the left hand side menu. This should then load a 'DynamoDB' block into the centre of the screen as shown below (you may or may not have an Alexa component to your app).



## Configure the DB

Scroll to the bottom of the page and complete the details below. Then click 'Add'. Then 'Save' your function.



The screenshot shows the 'Configure triggers' form in the AWS Lambda console. The form is titled 'Configure triggers'. It contains the following fields:

- DynamoDB table:** A dropdown menu with 'Device\_DB' selected. Below it, there is a link to 'Select a DynamoDB table to listen for updates on'.
- Batch size:** A text input field with the value '100'. Below it, there is a link to 'The largest number of records that will be read from your table's update stream at once'.
- Starting position:** A dropdown menu with 'Latest' selected. Below it, there is a link to 'The position in the stream to start reading from. For more information, see ShardIteratorType in the Amazon DynamoDB Streams API Reference'.

At the bottom of the form, there is a checkbox labeled 'Enable trigger' which is checked. Below it, there is a link to 'In order to read from the DynamoDB trigger, your execution role must have proper permissions. Enable the trigger now, or create it in a disabled state for testing (recommended)'.

At the bottom right of the form, there are two buttons: 'Cancel' and 'Add'.

## Write to the DB

Here is an example of how to write to the database.

TableName: Name of your Table

Key: Here you can provide keys

UpdateExpression:

[set / get / put etc] [item key] = :[temp var]  
“:[temp var] : value you wish to assign here”

### CODE Example:

```
var AWS = require('aws-sdk');
var docClient = new AWS.DynamoDB.DocumentClient();

var params = {
  TableName: "ENTER DB NAME HERE",
  Key:{
    "instance": 0
  },
  UpdateExpression: "set light = :lightValue",
  ExpressionAttributeValues:{
    ":lightValue" : "Hello World"
  },
  ReturnValues:"UPDATED_NEW"
};

docClient.update(params, ((err, data) => {
  if (err) {
    this.emit('ask', 'Sorry the request failed. Please try again');
  } else {
    this.emit('ask', 'Switching the light ' + slotValue);
  }
}));
```

## Read from the DB

Example of how to read from DB.

### CODE:

```
var AWS = require('aws-sdk');
var docClient = new AWS.DynamoDB.DocumentClient();

var params = {
  TableName: "ENTER DB NAME HERE",
  Key: {
    "instance": 0
  }
};

exports.handler = (event, context, callback) => {
  docClient.get(params, function(err, data) {
    if (err) { return console.error("that didn't work", data); }
    var payload = JSON.stringify(data, null, 2);
    var obj = JSON.parse(payload);
    var state = obj.Item;
    callback(null, state);
  });
};
```