

Module 6: Automate Monitoring And Event Management In AWS

Demo Document 3

edureka!

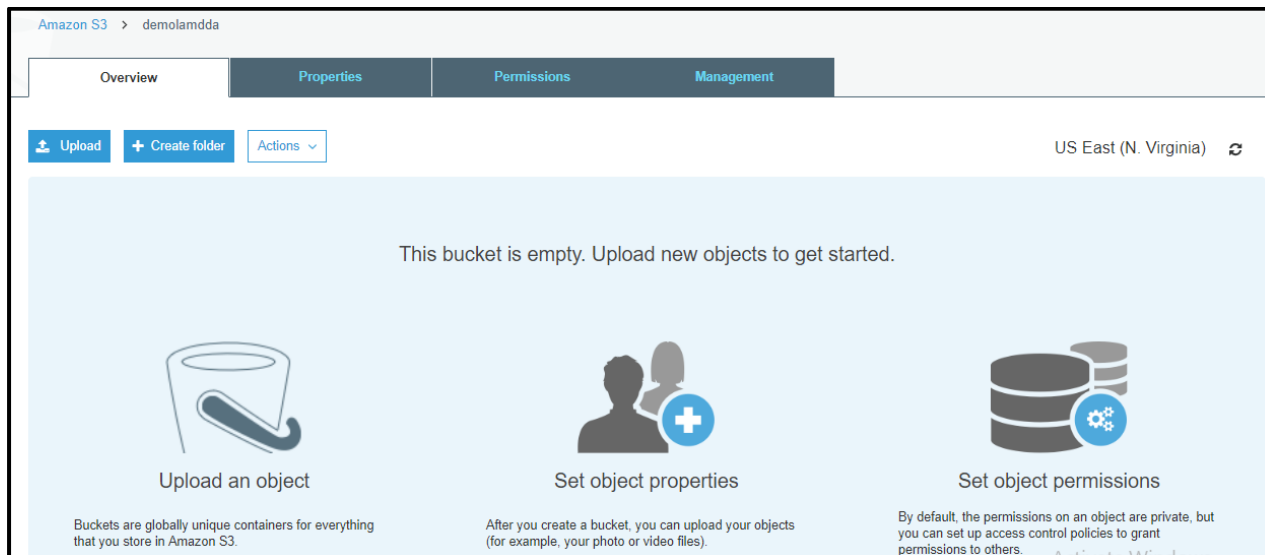
edureka!

© Brain4ce Education Solutions Pvt. Ltd.

Send an email when an object is added to S3 through Lambda

Demo steps:

Step1: Create A Bucket



Step2: Create an IAM role

- In IAM dashboard, select on roles and click on **create roles**
- In services, choose lambda and click on **permission**
- Choose administrator access and click on **review**
- Give a name for your role and click on **create role**


Review

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

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

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

Trusted entities AWS service: lambda.amazonaws.com

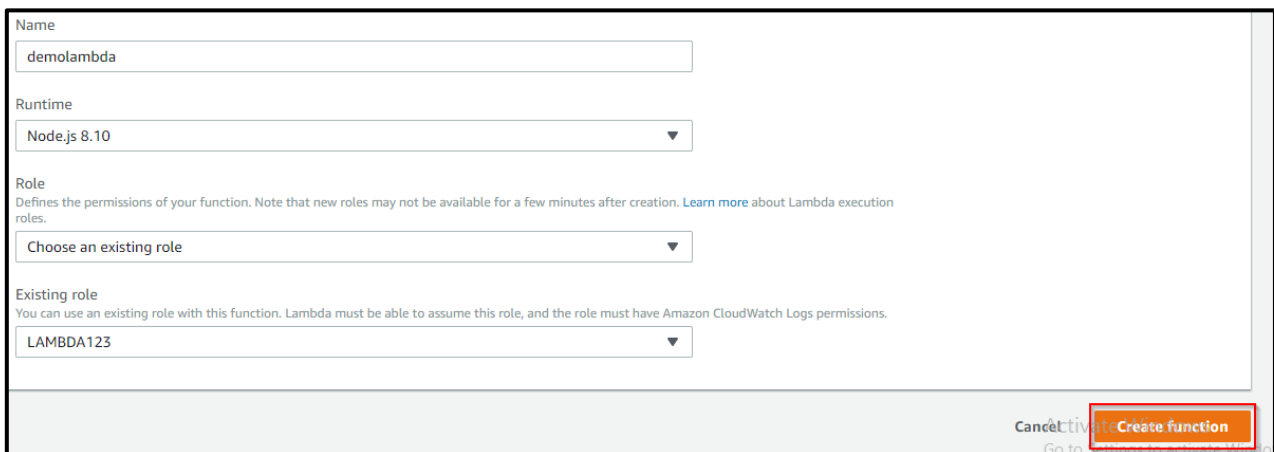
Policies  [AdministratorAccess](#)

Permissions boundary Permissions boundary is not set

* Required Cancel Previous Create role

Step3: Create a lambda function in the same region as the bucket

- In your management console, search for lambda service and select it
- Click on create lambda function
- Select author from scratch
- Give a name for lambda function
- Choose runtime as node.js 8.10
- In role, choose an existing role and select the name which you have created in pervious step
- Click on **create function**



Name
demolambda

Runtime
Node.js 8.10

Role
Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. [Learn more](#) about Lambda execution roles.
Choose an existing role

Existing role
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.
LAMBDA123

Cancel Go to **Create function**

Step 4: Type the below code in your lambda function and save it

```
var aws = require('aws-sdk');

var ses = new aws.SES({
  region: 'us-east-1'
});

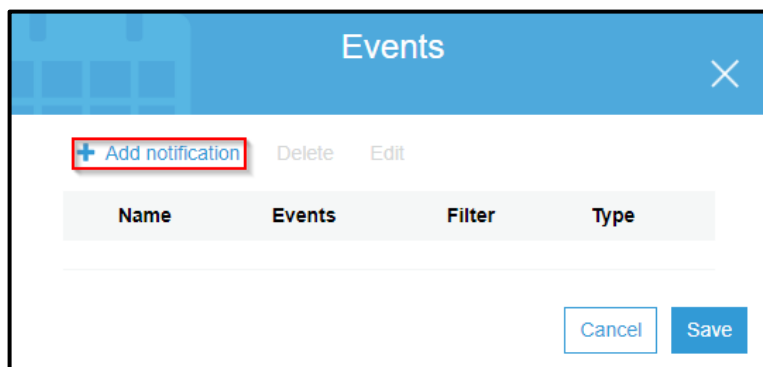
exports.handler = function(event, context) {
  console.log("Incoming: ", event);
  // var output = querystring.parse(event);

  var eParams = {
    Destination: {
      ToAddresses: ["ishwarya.a@edureka.co"] //give the email ID which is verified by SES
    },
```

```
Message: {
  Body: {Text: {
    Data: "lambda is working"
  }
},
  Subject: { Data: "mail from ses"}
},
  Source: ishu8297@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);
  }
});
};
```

Step 5: Add events in your buckets

- In properties, select **Events**
- Click on **add notifications**



- Select an event and under send to select lambda function

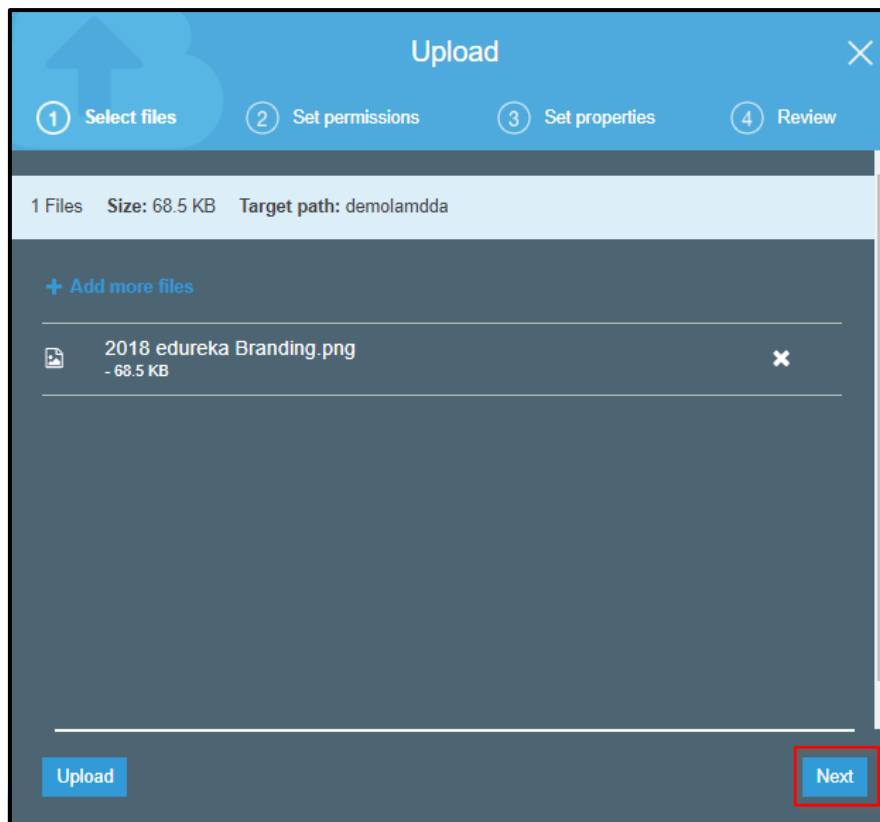
edureka!

- In lambda, choose the lambda function you have created and **save it**

Name	Events	Filter	Type
New event ×			
Name ⓘ			
<input type="text" value="e.g. MyEmailEventForPut"/>			
Events ⓘ			
<input type="checkbox"/> RRSObjectLost	<input type="checkbox"/> Delete		
<input checked="" type="checkbox"/> Put	<input type="checkbox"/> Delete Marker Created		
<input type="checkbox"/> Post	<input type="checkbox"/> ObjectCreate (All)		
<input type="checkbox"/> Copy	<input type="checkbox"/> ObjectDelete (All)		
<input type="checkbox"/> Complete Multipart Upload			
Prefix ⓘ			
<input type="text" value="e.g. images/"/>			
Suffix ⓘ			
<input type="text" value="e.g. .jpg"/>			
Send to ⓘ			
<input type="text" value="Lambda Function"/>			
Lambda			
<input type="text" value="demolambda"/>			
Activate Wire			
<input type="button" value="Cancel"/> <input type="button" value="Save"/>			

Step 6: Test the lambda function

- Upload an object to S3



- Check whether you have got an email

