

TRENDING: Hello world!

SEARCH

gefyrā

ABOUT PORTFOLIO TUTORIALS

TOPICS

Blog

Tutorials

- Data Engineering

- AWS CDK

- Web Development

GETTING STARTED WITH AWS CDK IN TYPESCRIPT

Posted by jonathan.moo | May 21, 2020 | AWS CDK, Data Engineering, Tutorials | 0 | ★★★★★

AWS Cloud Development Kit (AWS CDK) is toolkit for defining AWS cloud infrastructure in code. In short, we can manage the

<https://gefyra.co>

Pull requests Actions Projects Wiki Security Insights Settings

master 4 branches 0 tags Go to file Add file Code

Jonathan Moo and Jonathan Moo CDK-T1: Cleaned the README.md be6353f on Dec 26, 2020 3 commits

bin CDK-T1: Preparing CDK framework 9 months ago

lib CDK-T1: Preparing CDK framework 9 months ago

test CDK-T1: Preparing CDK framework 9 months ago

.gitignore CDK-T1: Preparing CDK framework 9 months ago

.npmignore CDK-T1: Preparing CDK framework 9 months ago

README.md CDK-T1: Cleaned the README.md 9 months ago

cdk.json CDK-T1: Preparing CDK framework 9 months ago

jest.config.js CDK-T1: Preparing CDK framework 9 months ago

-lock.json CDK-T1: Preparing CDK framework 9 months ago

About A demo on AWS CDK Readme

Releases No releases published Create a new release

Packages No packages published Publish your first package

Languages

<https://github.com/jonathan-moo/gefyra-cdk-demo/tree/master>

Never use your root credentials to access AWS services via the terminal!

This reduces the risk of exposing your root credentials to others.

**Pro Tip**



<https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html>

In the description down below.

<https://gefyra.co>



READEME.md — gefyra-cdk-demo

OPEN EDITORS

GEFYRA-CDK-DEMO

- bin
- gefyra-cdk-demo.d.ts
- gefyra-cdk-demo.js
- gefyra-cdk-demo.ts
- cdk.out
- lib
- node\_modules
- test
- .gitignore
- .npmignore
- cdk.json
- jest.config.js
- package-lock.json
- package.json
- README.md
- tsconfig.json

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE

```
(base) Jonathans-MacBook-Pro:gefyra-cdk-demo jonathanmoo$ aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [us-east-1]:
Default output format [None]:
(base) Jonathans-MacBook-Pro:gefyra-cdk-demo jonathanmoo$ npm install -g aws-cdk
added 5 packages, changed 185 packages, and audited 191 packages in 10s
found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 7.21.0 -> 7.24.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v7.24.0
npm notice Run npm install -g npm@7.24.0 to update!
npm notice
(base) Jonathans-MacBook-Pro:gefyra-cdk-demo jonathanmoo$
```

Rambling here.

It will give you a fresh install, but since I have CDK already, it would update my packages.

Had a long day.

## Configure your AWS account for access, then install the AWS CDK

package.json README.md tsconfig.json

```
(base) Jonathans-MacBook-Pro:gefyra-cdk-demo jonathanmoo$ ls
README.md          cdk.out           node_modules
bin                jest.config.js    package-lock.json
cdk.json          lib                package.json
tsconfig.json
```

.npmignore cdk.json jest.config.js package-lock.json package.json README.md tsconfig.json

```
(base) Jonathans-MacBook-Pro:Sites jonathanmoo$ cd cdk_test
(base) Jonathans-MacBook-Pro:cdk_test jonathanmoo$ ls
(base) Jonathans-MacBook-Pro:cdk_test jonathanmoo$ cdk init app --language typescript
Applying project template app for typescript
# Welcome to your CDK TypeScript project!

This is a blank project for TypeScript development with CDK.

The 'cdk.json' file tells the CDK Toolkit how to execute your app.

## Useful commands

* `npm run build` compile typescript to js
* `npm run watch` watch for changes and compile
* `npm run test` perform the jest unit tests
* `cdk deploy` deploy this stack to your default AWS account/Region
* `cdk diff` compare deployed stack with current state
* `cdk synth` emits the synthesized CloudFormation template

Initializing a new git repository...
Executing npm install...
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated sane@4.1.0: some dependency vulnerabilities fixed, support for node < 10 dropped, and ne
d
All done!
(base) Jonathans-MacBook-Pro:cdk_test jonathanmoo$ pwd
/Users/jonathanmoo/Sites/cdk_test
(base) Jonathans-MacBook-Pro:cdk_test jonathanmoo$
```



lib node\_modules test .gitignore .npmignore cdk.json jest.config.js package-lock.json package.json README.md tsconfig.json

```
(base) Jonathans-MacBook-Pro:cdk_test jonathanmoo$ pwd
/Users/jonathanmoo/Sites/cdk_test
(base) Jonathans-MacBook-Pro:cdk_test jonathanmoo$ ls -ali
total 1184
63737124 drwxr-xr-x 15 jonathanmoo staff 480 Sep 19 15:29 .
2375377 drwxr-xr-x 108 jonathanmoo staff 3456 Sep 19 15:28 ..
63737140 drwxr-xr-x 12 jonathanmoo staff 384 Sep 19 15:28 .git
63737127 -r--r--r-- 1 jonathanmoo staff 93 Sep 19 15:28 .gitignore
63737128 -r--r--r-- 1 jonathanmoo staff 65 Sep 19 15:28 .npmignore
63737129 -r--r--r-- 1 jonathanmoo staff 543 Sep 19 15:28 README.md
63737130 drwxr-xr-x 3 jonathanmoo staff 96 Sep 19 15:28 node_modules
63737132 -r--r--r-- 1 jonathanmoo staff 807 Sep 19 15:28 bin
63737133 -r--r--r-- 1 jonathanmoo staff 157 Sep 19 15:28 cdk.json
63737134 drwxr-xr-x 3 jonathanmoo staff 96 Sep 19 15:28 lib
63739845 drwxr-xr-x 374 jonathanmoo staff 11968 Sep 19 15:29 Jest.config.js
63760839 -r--r--r-- 1 jonathanmoo staff 576628 Sep 19 15:29 package-lock.json
63737136 -r--r--r-- 1 jonathanmoo staff 541 Sep 19 15:28 package.json
63737137 drwxr-xr-x 3 jonathanmoo staff 96 Sep 19 15:28 test
63737139 -r--r--r-- 1 jonathanmoo staff 650 Sep 19 15:28 tsconfig.json
(base) Jonathans-MacBook-Pro:cdk_test jonathanmoo$ cat README.md
# Welcome to your CDK TypeScript project!
```

This is a blank project for TypeScript development with CDK.

The 'cdk.json' file tells the CDK Toolkit how to execute your app.

## Useful commands

- \* `npm run build` compile typescript to js
- \* `npm run watch` watch for changes and compile
- \* `npm run test` perform the jest unit tests
- \* `cdk deploy` deploy this stack to your default AWS account/Region
- \* `cdk diff` compare deployed stack with current state
- \* `cdk synth` emits the synthesized CloudFormation template

(base) Jonathans-MacBook-Pro:cdk\_test jonathanmoo\$



The screenshot shows the VS Code interface with the file `gefyracdk-demo-stack.ts` open in the editor. The code defines a class `GefyraCdkDemoStack` that extends `cdk.Stack`. It includes a constructor that takes a scope, an id, and props. A comment indicates where the stack definition code goes. The file path is `gefyracdk-demo/lib/gefyracdk-demo-stack.ts`.

```
gefyracdk-demo-stack.ts — gefyra-cdk-demo
lib > TS gefyra-cdk-demo-stack.ts > ...
1 import * as cdk from '@aws-cdk/core';
2
3 export class GefyraCdkDemoStack extends cdk.Stack {
4   constructor(scope: cdk.Construct, id: string, props?: cdk.StackProps) {
5     super(scope, id, props);
6
7     // The code that defines your stack goes here
8   }
9 }
10
```

<https://docs.aws.amazon.com/cdk/api/latest/docs/aws-construct-library.html>

The screenshot shows the VS Code interface with the file `gefyracdk-demo.ts` open in the editor. It contains the main application setup code using AWS CDK. It imports node, source-map-support, and the stack definition from the lib directory. It creates a new `cdk.App` and adds the stack definition.

```
gefyracdk-demo.ts — gefyra-cdk-demo
bin > TS gefyra-cdk-demo.ts > ...
1 #!/usr/bin/env node
2 import '_source-map-support/register';
3 import * as cdk from '@aws-cdk/core';
4 import { GefyraCdkDemoStack } from '../lib/gefyracdk-demo-stack';
5
6 const app = new cdk.App();
7 new GefyraCdkDemoStack(app, 'GefyraCdkDemoStack');
8
```

The screenshot shows the VS Code interface with the file `gefyracdk-demo.test.ts` open in the editor. It contains unit tests for the application using Jest and AWS CDK assertions. It tests an empty stack by creating a new app and stack, then asserting they match the template.

```
gefyracdk-demo.test.ts — gefyra-cdk-demo
test > TS gefyra-cdk-demo.test.ts > ...
1 import { expect as expectCDK, matchTemplate, MatchStyle } from '@aws-cdk/assert';
2 import * as cdk from '@aws-cdk/core';
3 import * as GefyraCdkDemo from '../lib/gefyracdk-demo-stack';
4
5 test('Empty Stack', () => {
6   const app = new cdk.App();
7   // WHEN
8   const stack = new GefyraCdkDemo.GefyraCdkDemoStack(app, 'MyTestStack');
9   // THEN
10  expectCDK(stack).to(matchTemplate({
11    "Resources": {}
12  }, MatchStyle.EXACT));
13});
```

The screenshot shows the VS Code interface with the file `cdk.json` open in the editor. It contains the CDK configuration for the application, including the command to run the application (`app`), context settings for stack name duplicates, diff fail, and relative exports, and the stack name (`MyTestStack`).

```
cdk.json — gefyra-cdk-demo
cdk.json > ...
1 {
2   "app": "npx ts-node bin/gefyracdk-demo.ts",
3   "context": {
4     "@aws-cdk/core:enableStackNameDuplicates": "true",
5     "aws-cdk:enableDiffNoFail": "true",
6     "@aws-cdk/core:stackRelativeExports": "true"
7   }
8 }
```

The screenshot shows the VS Code interface with the 'jest.config.js' file selected in the Explorer sidebar. The code editor displays the following content:

```
jest.config.js — gefyra-cdk-demo
JS jest.config.js > [e] <unknown>
1 module.exports = {
2   roots: ['<rootDir>/test'],
3   testMatch: ['**/*.{test}.ts'],
4   transform: {
5     '^.+\\.tsx?$': 'ts-jest'
6   }
7 };
```

The screenshot shows the VS Code interface with the 'package.json' file selected in the Explorer sidebar. The code editor displays the following content:

```
package.json — gefyra-cdk-demo
JSON package.json > ...
1 {
2   "name": "gefyra-cdk-demo",
3   "version": "0.1.0",
4   "bin": {
5     "gefyra-cdk-demo": "bin/gefyra-cdk-demo.js"
6   },
7   "scripts": {
8     "build": "tsc",
9     "watch": "tsc -w",
10    "test": "jest",
11    "cdk": "cdk"
12  },
13  "devDependencies": {
14    "@aws-cdk/assert": "^1.80.0",
15    "@types/jest": "^26.0.10",
16    "@types/node": "^10.17.50",
17    "aws-cdk": "^1.80.0",
18    "jest": "^26.4.2",
19    "ts-jest": "^26.2.0",
20    "ts-node": "8.1.0",
21    "typescript": "~3.9.7"
22  },
23  "dependencies": {
24    "@aws-cdk/core": "^1.80.0",
25  }
26}
```

The screenshot shows the VS Code interface with the 'package.json' file selected in the Explorer sidebar. The code editor displays the following content:

```
package.json — gefyra-cdk-demo
JSON package.json > ...
1 },
2   "scripts": {
3     "build": "tsc",
4     "watch": "tsc -w",
5     "test": "jest",
6     "cdk": "cdk"
7   },
8   "devDependencies": {
9     "@aws-cdk/assert": "^1.80.0",
10    "@types/jest": "^26.0.10",
11    "@types/node": "^10.17.50",
12    "aws-cdk": "^1.80.0",
13    "jest": "^26.4.2",
14    "ts-jest": "^26.2.0",
15    "ts-node": "8.1.0",
16    "typescript": "~3.9.7"
17  },
18  "dependencies": {
19    "@aws-cdk/core": "^1.80.0",
20    "source-map-support": "^0.5.16"
21  }
22 },
23 },
24 }
```

```

tsconfig.json — gefyra-cdk-demo
tsconfig.json > ...
1  {
2    "compilerOptions": {
3      "target": "ES2018",
4      "module": "commonjs",
5      "lib": ["es2018"],
6      "declaration": true,
7      "strict": true,
8      "noImplicitAny": true,
9      "strictNullChecks": true,
10     "noImplicitThis": true,
11     "alwaysStrict": true,
12     "noUnusedLocals": false,
13     "noUnusedParameters": false,
14     "noImplicitReturns": true,
15     "noFallthroughCasesInSwitch": false,
16     "inlineSourceMap": true,
17     "inlineSources": true,
18     "experimentalDecorators": true,
19     "strictPropertyInitialization": false,
20     "typeRoots": ["./node_m
21   },
22   "exclude": ["cdk.out"]
23 }

```

cdk init app --language typescript

```

package-lock.json          1.90.1   1.12
jest                      26.6.3   26.
source-map-support        0.5.19   0.5.
ts-jest                   26.5.1   26.5
ts-node                   8.16.2   8.16
typescript                3.9.9    3.9.
tsconfig.json
(base) Jonathan's-MacBook-Pro:gefyra-cd

```

Your AWS profile, and this is especially needed if you're handling more than 1 default AWS account.

Amazon S3

Buckets

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- Access analyzer for S3

Block Public Access settings for this account

Storage Lens

- Dashboards
- AWS Organizations settings

Feature spotlight

AWS Marketplace for S3

Amazon S3

Account snapshot

Buckets (9) Info

Name	AWS Region	Access	Creation date
cdktoolkit-stagingbucket-ds		Bucket and objects not public	June 1, 2020, 08:38:00 (UTC-04:00)
cdktoolkit-stagingbucket-us		Bucket and objects not public	February 25, 2021, 04:31:45 (UTC-05:00)
[REDACTED]			

[View Storage Lens dashboard](#)

England Football Results Betting

football-data.co.uk/englandm.php

**Football-Data.co.uk**

Results / Odds / Tipsters

Network Site

Updated: 30/09/21

Bet365

Home Free Bets Livescores Odds Comparison Casino Poker Tennis Contact Like

**WORLD'S FAVOURITE**

**TOP RATED**

**Season 2021/2022**

**SITE RESOURCES**

**BETTING SYSTEMS**

**BETTING ARTICLES**

**BET CALCULATORS**

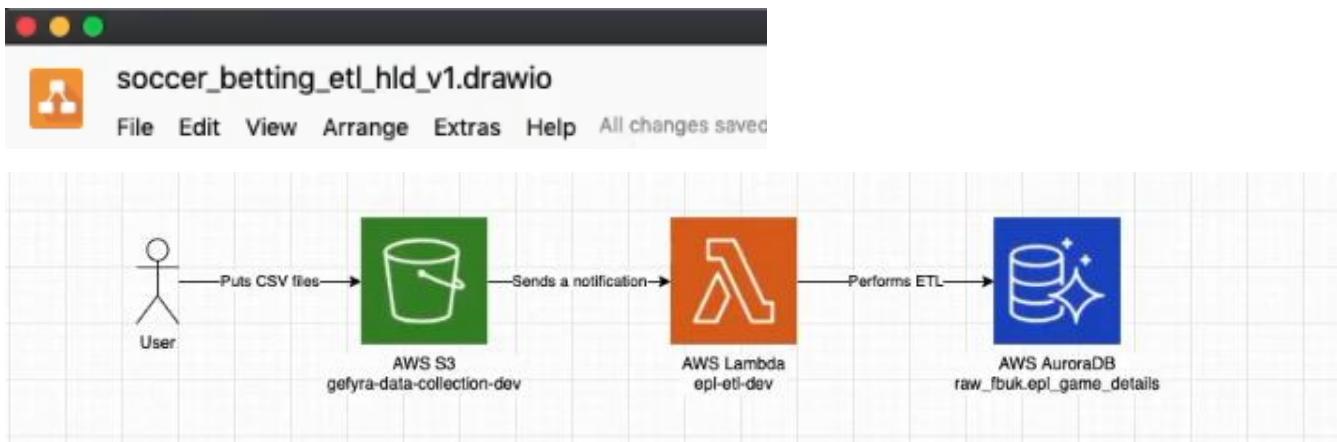
**ODDS & RESULTS**

Excel files aren't the file formats we use for data engineering and science. But this is subjective to your company's way of doing things.

We want this flat file data in S3 to be stored/ingested in some database like DynamoDB to allow querying.

Files, in and of itself, cannot be queried easily (unless you have crawled the data beforehand).  
Queries allow us to zoom into the segments of the data that we want for validation and cleaning.

## ETL - Extract, Transform, Load



An AWS Lambda is a container, hosted in a virtual machine environment, where code can be executed.

AWS Data Pipeline - <https://aws.amazon.com/datapipeline/>  
A managed service from AWS for ETLs.

<https://github.com/jonathan-moo/gefyra-cdk-demo/tree/CDK-T2>



A screenshot of a Mac OS X desktop showing a video call in progress. The video call window is on the right, showing a person in a blue t-shirt. On the left is a screenshot of the VS Code code editor. The Explorer sidebar shows a project structure with files like `bin`, `lib`, and `test`. The main editor tab is `s3-bucket-stack.ts` (M), which contains the following TypeScript code:

```

s3-bucket-stack.ts — gefyra-cdk-demo

lib > TS s3-bucket-stack.ts > S3BucketStack > constructor
1 import * as cdk from '@aws-cdk/core';
2 import * as s3 from "@aws-cdk/aws-s3";
3
4 export class S3BucketStack extends cdk.Stack {
5   public readonly bucket: s3.Bucket;
6
7   constructor(scope: cdk.Construct, id: string, props?: cdk.StackProps) {
8     super(scope, id, props);
9
10    // The code that defines your stack goes here
11    this.bucket = new s3.Bucket(this, "gefyra-data-collection-dev", [
12      versioned: false,
13      bucketName: "gefyra-data-collection-dev",
14      publicReadAccess: false,
15      blockPublicAccess: s3.BlockPublicAccess.BLOCK_ALL,
16      removalPolicy: cdk.RemovalPolicy.DESTROY
17    ]);
18  }
19}

```

The terminal at the bottom shows the command `(base) Jonathan's-MacBook-Pro:gefyra-cdk-demo jonathanmoo\$`.

This tells AWS CDK to configure for you, above is the S3 bucket stack with the exported class we create.



<https://docs.aws.amazon.com/cdk/api/latest/docs/aws-s3-readme.html>



A screenshot of a Mac OS X desktop showing a video call in progress. The video call window is on the right, showing a person in a blue t-shirt. On the left is a screenshot of the VS Code code editor. The Explorer sidebar shows a project structure with files like `bin`, `lib`, and `test`. The main editor tab is `gefyra-cdk-demo.ts` (M), which contains the following TypeScript code:

```

gefyra-cdk-demo.ts — gefyra-cdk-demo

bin > TS gefyra-cdk-demo.ts > [e] bucket
1 #!/usr/bin/env node
2 import 'source-map-support/register';
3 import * as cdk from '@aws-cdk/core';
4 //import { GefyraCdkDemoStack } from '../lib/gefyra-cdk-demo-stack';
5 import { S3BucketStack } from '../lib/s3-bucket-stack';
6
7 const app = new cdk.App();
8 //new GefyraCdkDemoStack(app, 'GefyraCdkDemoStack');
9
10 // Creating an S3 bucket stack
11 const s3_bucket_stack = new S3BucketStack(app, 'gefyraS3Stack');
12
13 // Re-using assets
14 const bucket = s3_bucket_stack.bucket;
15

```

The terminal at the bottom shows the command `(base) Jonathan's-MacBook-Pro:gefyra-cdk-demo jonathanmoo\$ npm run build && cdk synth` followed by `> gefyra-cdk-demo@0.1.0 build` and `> tsc`.

We are assigning a name to the S3 bucket on line 14 to allow us to refer to it when reusing it with other integrations. IAM roles that are unique to your stack can be defined if different from the common IAM role created by default.

```

> cdk.out
  lib
    TS basic-lambda-stack.d.ts
    JS basic-lambda-stack.js
    TS gefyra-cdk-demo-stack.d.ts
    JS gefyra-cdk-demo-stack.js
    TS gefyra-cdk-demo-stack.ts
    JS s3-bucket-stack.js
    TS s3-bucket-stack.ts M

  > node_modules
  > test
    TS gefyra-cdk-demo.test.d.ts
    JS gefyra-cdk-demo.test.js
    TS gefyra-cdk-demo.test.ts
    .gitignore
    .npmignore
    cdk.json
    jest.config.js
    package-lock.json
    package.json M M

  > OUTLINE
  > TIMELINE
  > NPM SCRIPTS

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE
  - Ref: AWS::Region
  - us-east-1
  - Fn::Equals:
    - Ref: AWS::Region
    - us-east-2
  - Fn::Or:
    - Fn::Equals:
      - Ref: AWS::Region
      - us-west-1
    - Fn::Equals:
      - Ref: AWS::Region
      - us-west-2

(base) Jonathans-MacBook-Pro:gefyra-cdk-demo jonathanmoo$ cdk destroy --profile gefyra_terraform
Are you sure you want to delete: gefyraS3Stack (y/n)? y
gefyraS3Stack: destroying...

✓ gefyraS3Stack: destroyed
(base) Jonathans-MacBook-Pro:gefyra-cdk-demo jonathanmoo$ cdk deploy --profile gefyra_terraform
gefyraS3Stack: deploying...
gefyraS3Stack: creating CloudFormation changeset...

✓ gefyraS3Stack

Stack ARN:
arn:aws:cloudformation:us-east-1:101699910734:stack/gefyraS3Stack/519fbfc0-344e-11ec-ae69-0e573fcf284d
(base) Jonathans-MacBook-Pro:gefyra-cdk-demo jonathanmoo$ []

```

Amazon S3

Buckets

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

Access analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight

AWS Marketplace for S3

Amazon S3

▶ Account snapshot

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

[View Storage Lens dashboard](#)

Buckets (10) Info

Buckets are containers for data stored in S3. [Learn more](#)

Find buckets by name

Name
bombora-raw
cdktoolkit-stagingbucket-d5pd83ynqq5u
cdktoolkit-stagingbucket-u9b8ane4zh95
cdn.livinggroup.asia
cdn.webmusing.com
fe-store-asia
gefyra-data-collection-dev
gefyra-data-collection-test
gefyra-raw
gefyra-test
gefyra-test-1

[Create bucket](#)

```

> cdk.out
  lib
    TS basic-lambda-stack.d.ts
    JS basic-lambda-stack.js
    TS gefyra-cdk-demo-stack.d.ts
    JS gefyra-cdk-demo-stack.js
    TS gefyra-cdk-demo-stack.ts
    JS s3-bucket-stack.js
    TS s3-bucket-stack.ts M

  > node_modules
  > test
    TS gefyra-cdk-demo.test.d.ts
    JS gefyra-cdk-demo.test.js
    TS gefyra-cdk-demo.test.ts
    .gitignore
    .npmignore
    cdk.json
    jest.config.js
    package-lock.json
    package.json M M

  > OUTLINE
  > TIMELINE
  > NPM SCRIPTS

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE
  this.bucket = new S3.Bucket(this, 'gefyra-data-collection-dev', {
    ... }

(base) Jonathans-MacBook-Pro:gefyra-cdk-demo jonathanmoo$ cdk destroy --profile gefyra_terraform
Are you sure you want to delete: gefyraS3Stack (y/n)? y
gefyraS3Stack: destroying...

✓ gefyraS3Stack: destroyed
(base) Jonathans-MacBook-Pro:gefyra-cdk-demo jonathanmoo$ cdk deploy --profile gefyra_terraform
gefyraS3Stack: deploying...
gefyraS3Stack: creating CloudFormation changeset...

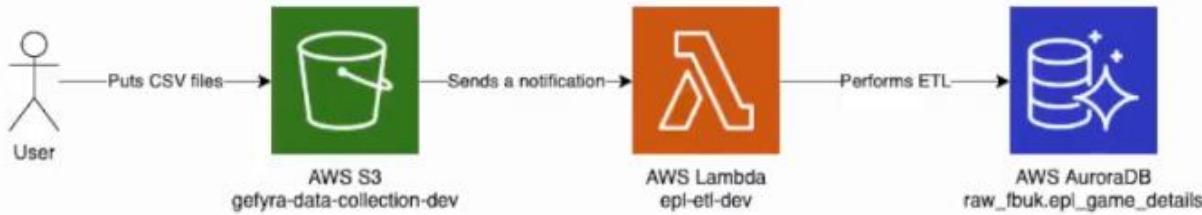
✓ gefyraS3Stack

Stack ARN:
arn:aws:cloudformation:us-east-1:101699910734:stack/gefyraS3Stack/519fbfc0-344e-11ec-ae69-0e573fcf284d
(base) Jonathans-MacBook-Pro:gefyra-cdk-demo jonathanmoo$ cdk destroy --profile gefyra_terraform
Are you sure you want to delete: gefyraS3Stack (y/n)? y
gefyraS3Stack: destroying...

✓ gefyraS3Stack: destroyed
(base) Jonathans-MacBook-Pro:gefyra-cdk-demo jonathanmoo$ []

```

The screenshot shows the AWS S3 Management Console. On the left, there's a sidebar with various AWS services like Storage Lens, Dashboards, and AWS Organizations settings. The main area is titled "Amazon S3" and shows an "Account snapshot" with a link to "Storage lens provides visibility into storage usage and activity trends". Below this is a section for "Buckets" with a count of 9. A search bar says "Find buckets by name". To the right are buttons for "Copy ARN", "Empty", "Delete", and "Create bucket". A large black rectangular redaction box covers the list of bucket names.



We deploy a Lambda to get a S3 new file notification, retrieve the file, process the file contents, then store results in DB.

The screenshot shows the AWS CDK development environment in VS Code. The Explorer sidebar shows a project structure for "GEFYRA-CDK-DEMO" with files like "basic-lambda-stack.ts", "lambda\_function.py", and "basic-lambda-stack.js". The terminal window shows the command "npm install @aws-cdk/aws-lambda" being run, followed by audit results. A video overlay shows a person speaking.

```

npm install @aws-cdk/aws-lambda
[...]
(base) Jonathan-MacBook-Pro:gefyra-cdk-demo jonathanmoo$ npm install @aws-cdk/aws-lambda
removed 7 packages, and audited 793 packages in 6s
25 packages are looking for funding
  run `npm fund` for details
27 high severity vulnerabilities
To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
(base) Jonathan-MacBook-Pro:gefyra-cdk-demo jonathanmoo$ npm outdated
[...]
  package@ Current Wanted Latest Location Dependents
@types/jest 26.0.24 26.0.24 27.0.7 node_modules/@types/jest gefyra-cdk-demo
@types/node 10.17.68 10.17.68 16.11.6 node_modules/@types/node gefyra-cdk-demo
jest 26.6.3 26.6.3 27.3.1 node_modules/jest gefyra-cdk-demo
ts-jest 26.5.6 26.5.6 27.0.7 node_modules/ts-jest gefyra-cdk-demo
ts-node 8.10.2 8.10.2 18.4.0 node_modules/ts-node gefyra-cdk-demo
typescript 3.9.10 3.9.10 4.4.4 node_modules/typescript gefyra-cdk-demo
(base) Jonathan-MacBook-Pro:gefyra-cdk-demo jonathanmoo$ npm install -g aws-cdk
changed 180 packages, and audited 187 packages in 5s
found 0 vulnerabilities
(base) Jonathan-MacBook-Pro:gefyra-cdk-demo jonathanmoo$ 
  
```

<https://github.com/jonathan-moo/gefyra-cdk-demo/tree/CDK-T3>

The screenshot shows a video call interface with a person in the top right corner. In the foreground, a code editor window is displayed. The left sidebar shows a file tree for a project named 'GEFYRA-CDK-DEMO'. The main area shows two files: 'lambda\_function.py' and 'basic-lambda-stack.ts'. The 'basic-lambda-stack.ts' file contains TypeScript code for a Lambda function stack. The terminal tab at the bottom shows the command 'python lambda\_function.py' being run, with the output showing a successful execution.

Now that we have our basic function, it is time to write the infrastructure code to create a Lambda to run the code.

The screenshot shows a video call interface with a person in the top right corner. In the foreground, a code editor window is displayed. The left sidebar shows a file tree for a project named 'GEFYRA-CDK-DEMO'. The main area shows the 'basic-lambda-stack.ts' file. The terminal tab at the bottom shows the command 'cdk deploy' being run.

The screenshot shows a video call interface with a person in the top right corner. In the foreground, a code editor window is displayed. The left sidebar shows a file tree for a project named 'GEFYRA-CDK-DEMO'. The main area shows the 'gefyra-cdk-demo.ts' file. The terminal tab at the bottom shows the command 'cdk synth' being run.

We then add the LambdaStack to our app on line 18 for deployment

The screenshot shows a code editor window with the 'gefyra-cdk-demo.ts' file open. On line 18, there is a new line of code: 'const basic\_lambda\_stack = new basicLambdaStack(app, 'basicLambdaStack');'. The terminal tab at the bottom shows the command 'cdk deploy' being run.

We can choose which of the 2 stacks (S3BucketStack and the basicLambdaStack) we have that needs to be deployed.

The screenshot shows the AWS Lambda console interface. At the top, there's a navigation bar with tabs like 'Getting Started' and 'Services'. Below it, a search bar says 'Search for services, features, marketplace products, and docs [Option+5]'. The main area is titled 'Lambda > Functions' and shows a table header for 'Functions (0)'. The columns are 'Function name', 'Description', 'Package type', 'Runtime', 'Code size', and 'Last modified'. A message at the bottom of the table says 'There is no data to display.' On the right side of the screen, there's a video feed of a person speaking.

This screenshot shows a terminal window in VS Code with the title 'gefyracdk-demo.ts — gefyracdk-demo'. The code in the editor is related to deploying a basic Lambda stack using AWS CDK. The terminal output shows the deployment process:

```
(base) Jonathanhs-MacBook-Pro:gefyracdk-demo jonathanmoo$ cdk deploy basicLambdaStack --profile gefyra_us_east_2
This deployment will make sensitive changes according to your current security approval level (--require-approval bro
adening).
Please confirm you intend to make the following modifications:
```

Below the terminal, there are sections for 'IAM Statement Changes' and 'IAM Policy Changes', both listing a single entry for the service role. A note at the bottom of the terminal says '(NOTE: There may be security-related changes not in this list. See https://github.com/aws/aws-cdk/issues/1299)'.

This screenshot continues from the previous one, showing the deployment process. The terminal output shows the stack publishing and creating changesets:

```
[0%] start: Publishing 603d769490796a8f3ca95023f9a5a25bd872343fb8c98ad038a44329b413e28b:current
[100%] success: Published 603d769490796a8f3ca95023f9a5a25bd872343fb8c98ad038a44329b413e28b:current
basicLambdaStack: creating CloudFormation changeset...
```

At the bottom of the terminal, a green checkmark icon appears next to the stack name 'basicLambdaStack', indicating successful deployment. The status bar at the bottom of the VS Code window shows 'Python 3.8.5 64-bit' and other development details.

The screenshot shows the AWS Lambda Functions page. A search bar at the top right contains the text 'Search for services, features, marketplace products, and docs [Option+S]'. Below it, a table lists one function:

Function name	Description	Package type	Runtime	Code size	Last modified
gefyra-basic-lambda	-	Zip	Python 3.8	301.0 byte	31 seconds ago

The screenshot shows the 'gefyra-basic-lambda' function details page. The 'Function overview' section includes:

- A thumbnail for the function.
- A 'Layers' section with '(0)'.
- A 'Description' field containing '-'.
- A 'Last modified' field showing '39 seconds ago'.
- A 'Function ARN' field: 'arn:aws:lambda:us-east-2:101699910734:function:gefyra-basic-lambda'.
- An 'Application' field: 'basicLambdaStack'.

The 'Code' tab is selected, showing the code editor with the following Python code:

```
lambda_function
1 def lambda_handler(event, context):
2     return {
3         "statusCode":200
4     }
5 # For direct invocation and testing on the local machine
6 if __name__ == '__main__':
7     print(lambda_handler(None,None))
```

The screenshot shows the 'gefyra-basic-lambda' function details page again, with the 'Code' tab selected. The code editor displays the same Python code as before:

```
lambda_function
1 def lambda_handler(event, context):
2     return {
3         "statusCode":200
4     }
5 # For direct invocation and testing on the local machine
6 if __name__ == '__main__':
7     print(lambda_handler(None,None))
```

The screenshot shows the AWS Lambda console interface. At the top, the function name 'gefyra-basic-lambda' is visible. Below it, the 'Test' tab is selected. The 'Test event' section contains a JSON editor with the following content:

```
1 - {  
2   "key1": "value1",  
3   "key2": "value2",  
4   "key3": "value3"  
5 }
```

The screenshot shows the AWS Lambda console interface. A green success message at the top states: 'The test event initializationTest was successfully saved.' The 'Test' tab is selected. The 'Test event' section shows the 'Saved event' option selected, and the event name 'initializationTest' is entered. The JSON editor is empty.

The screenshot shows the AWS Lambda console interface. A green success message at the top states: 'The test event initializationTest was successfully saved.' The 'Test' tab is selected. The 'Execution result: succeeded (logs)' section displays the following log output:

```
{  
  "statusCode": 200  
}
```

Summary details include:

- Code SHA-256: DYJ3grUIMu8G7Hf2XFxG/tbrt/ESw/fHPe+QN47QfvY=
- Request ID: f393392e-aea8-43c1-ac72-de4f89e2de24
- Init duration: 122.60 ms
- Billed duration: 2 ms
- Max memory used: 38 MB
- Duration: 1.38 ms
- Resources configured: 128 MB

The 'Log output' section shows the following CloudWatch logs:

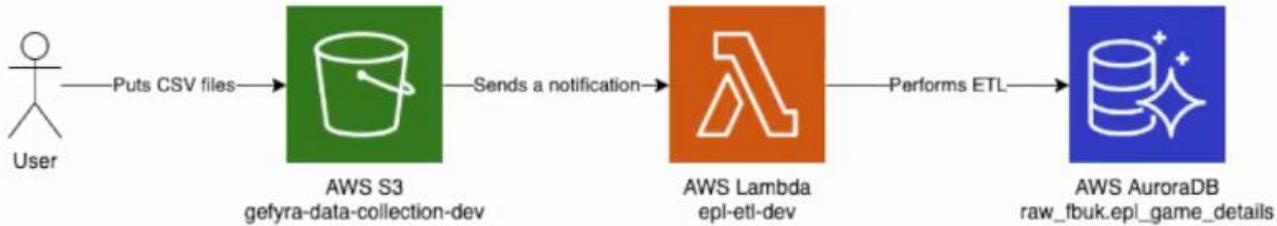
```
START RequestId: f393392e-aea8-43c1-ac72-de4f89e2de24 Version: $LATEST  
END RequestId: f393392e-aea8-43c1-ac72-de4f89e2de24  
REPORT RequestId: f393392e-aea8-43c1-ac72-de4f89e2de24 Duration: 2 ms Memory Size: 128 MB Max Memory Used: 38 MB Init Duration: 122.60 ms
```

```
(base) Jonathans-MacBook-Pro:gefyra-cdk-demo jonathanmoo$ cdk destroy basicLambdaStack --profile gefyra_us_east_2
Are you sure you want to delete: basicLambdaStack (y/n)? y
basicLambdaStack: destroying...
basicLambdaStack: destroyed
```

Last fetched in 0

Function name	Description	Package type	Runtime	Code size	Last modified

There is no data to display.



We now need to connect our S3 bucket to the Lambda, we need to make sure that the Lambda captures the S3 event.

```

lambda_function.py M X
src > lambda > basic_lambda > lambda_function.py > lambda_handler
1 import logging
2
3 logger = logging.getLogger()
4 logger.setLevel(logging.DEBUG)
5
6 def lambda_handler(event, context):
7     # Outputs the incoming event into CW logs
8     logger.info("Event:")
9     logger.info(event)
10    print(event)
11
12    return {
13        "statusCode": 200
14    }
15
16    # For direct invocation and testing on the local machine
17    if __name__ == '__main__':
18        print(lambda_handler(None,None))
  
```

(gefyra-cdk-demo) (base) Jonathans-MacBook-Pro:basic\_lambda jonathanmoo\$ python lambda\_function.py

None

{'statusCode': 200}

(gefyra-cdk-demo) (base) Jonathans-MacBook-Pro:basic\_lambda jonathanmoo\$

When we run the function, we won't see the logger logs on the console BUT they should be in CloudWatch where we can use it to trigger an event. The event is **None** since we haven't set an event up yet.

The screenshot shows a video call interface with a person on the right and a code editor window on the left. The code editor displays a file named `lambda_function.py` with the following content:

```

lib > TS basic-lambda-stack.ts > basicLambdaStack > constructor
1 import * as cdk from '@aws-cdk/core';
2 import * as lambda from '@aws-cdk/aws-lambda';
3
4 export class basicLambdaStack extends cdk.Stack{
5   // Making the object accessible for reusability
6   public readonly lambdaFunction: lambda.Function;
7
8   constructor(scope: cdk.Construct, id: string, props?: cdk.StackProps) {
9     super(scope, id, props);
10
11    const function_name = 'gefyra-basic-lambda';
12    const lambda_path = 'src/lambda/basic_lambda';
13
14    // Initialization of the lambda function
15    this.lambdaFunction = new lambda.Function(this, function_name, {
16      functionName: function_name,
17      runtime: lambda.Runtime.PYTHON_3_8,
18      code: lambda.Code.fromAsset(lambda_path),
19      handler: "lambda_function.lambda_handler"
20    });
21  }
22}

```

We can now expose our Lambda resource as an object within the AWS CDK, we define the `basicLambdaStack`. We declare a new `LambdaFunction` as a variable that can be reused in other stack on line 6 as `public readonly`.

The screenshot shows a video call interface with a person on the right and a code editor window on the left. The code editor displays a file named `s3-bucket-stack.ts` with the following content:

```

lib > TS s3-bucket-stack.ts > S3BucketStack > constructor
1 import * as cdk from '@aws-cdk/core';
2 import * as s3 from "@aws-cdk/aws-s3";
3 import * as s3n from '@aws-cdk/aws-s3-notifications';
4 import * as lambda from '@aws-cdk/aws-lambda';
5
6 // Allows the stack to receive a lambda.function object
7 export interface S3Props extends cdk.StackProps{
8   readonly lambdaFunction: lambda.Function;
9 }
10
11 export class S3BucketStack extends cdk.Stack {
12   public readonly bucket: s3.Bucket;
13
14   // Replace the props with our new interface
15   constructor(scope: cdk.Construct, id: string, props: S3Props) {
16     super(scope, id, props);
17
18     // The code that defines your stack goes here
19     this.bucket = new s3.Bucket(this, "gefyra-data-collection-dev",{
20       versioned: false,
21       bucketName: "gefyra-data-collection-dev",
22       publicReadAccess: false,
23       blockPublicAccess: s3.BlockPublicAccess.BLOCK_ALL,
24       removalPolicy: cdk.RemovalPolicy.DESTROY
25     });
26
27     // Assigning notifications to be sent to the Lambda function
28     this.bucket.addEventNotification(s3.EventType.OBJECT_CREATED, new s3n.LambdaDestination(props.lambdaFunction));
29   }
30 }

```

Our S3 bucket needs to know where to send the notifications to, we define an interface on line 7 to extend the class and passed as the props on line 15. We then add the event notification on line 28.

Run on your terminal at your project root folder:  
`npm install @aws-cdk/aws-s3-notifications`

The code editor shows a tooltip for the `s3.EventType.OBJECT_CREATED` enum member. The tooltip contains the following information:

- (enum member) `s3.EventType.OBJECT_CREATED` = "s3:ObjectCreated:\*
- Amazon S3 APIs such as PUT, POST, and COPY can create an object.
- Using these event types, you can enable notification when an object is created using a specific API, or you can use the `s3:ObjectCreated:*` event type to request notification regardless of the API that was used to create an object.
- `@stability — stable`

The screenshot shows a video call interface on the right and a code editor on the left. The code editor displays a file named `gefyra-cdk-demo.ts` with the following content:

```

bin > TS gefyra-cdk-demo.ts ...
1  #!/usr/bin/env node
2  import `source-map-support/register`;
3  import * as cdk from '@aws-cdk/core';
4  //import { GefyraCdkDemoStack } from '../lib/gefyra-cdk-demo-stack';
5  import { S3BucketStack } from '../lib/s3-bucket-stack';
6  import { basicLambdaStack } from '../lib/basic-lambda-stack';
7
8  const app = new cdk.App();
9  //new GefyraCdkDemoStack(app, 'GefyraCdkDemoStack');
10
11 // Deploying basic Lambda function
12 const basic_lambda_stack = new basicLambdaStack(app, 'basicLambdaStack');
13
14 // Creating an S3 bucket stack
15 const s3_bucket_stack = new S3BucketStack(app, 'gefyraS3Stack', {
16   | lambdaFunction: basic_lambda_stack.lambdaFunction
17 });
18
19 // Re-using assets
20 const bucket = s3_bucket_stack.bucket;
21

```

We rearrange to create the lambda stack, then the S3 bucket to use the Lambda stack as above

The screenshot shows a video call interface on the right and a code editor on the left. The code editor displays a terminal window with the following output:

```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE
(gefyra-cdk-demo) (base) Jonathans-MacBook-Pro:gefyra-cdk-demo jonathanmoo$ npm run build && cdk synth
> gefyra-cdk-demo@0.1.0 build
> tsc

Successfully synthesized to /Users/jonathanmoo/Sites/gefyra-cdk-demo/cdk.out
Supply a stack id (basicLambdaStack, gefyraS3Stack) to display its template.
(gefyra-cdk-demo) (base) Jonathans-MacBook-Pro:gefyra-cdk-demo jonathanmoo$ 

```

Below the terminal, there is a deployment summary and IAM policy changes table:

Resource	Effect	Action	Principal	Condition
+ \${gefyra-basic-lambda/ServiceRole Arn}	Allow	sts:AssumeRole	Service:lambda.amazonaws.com	

**IAM Policy Changes**

Resource	Managed Policy ARN
+ \${gefyra-basic-lambda/ServiceRole}	arn:aws:partition::iam::aws:policy/service-role/AWSLambdaBasicExecutionRole

(NOTE: There may be security-related changes not in this list. See <https://github.com/aws/aws-cdk/issues/1299>)

Do you wish to deploy these changes (y/n)?

The screenshot shows a video call interface on the right and a code editor on the left. The code editor displays a terminal window with the following output:

```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE
bin > TS gefyra-cdk-demo.ts > s3_bucket_stack > lambdaFunction
1  #!/usr/bin/env node
2  import `source-map-support/register`;
3  import * as cdk from '@aws-cdk/core';
4  //import { GefyraCdkDemoStack } from '../lib/gefyra-cdk-demo-stack';
5  import { S3BucketStack } from '../lib/s3-bucket-stack';
6  import { basicLambdaStack } from '../lib/basic-lambda-stack';
7
8  const app = new cdk.App();
9  //new GefyraCdkDemoStack(app, 'GefyraCdkDemoStack');
10

Outputs:
basicLambdaStack.ExportsOutputFnGetAttgefyrabasiclambda456DE80BArn787BDAEE = arn:aws:lambda:us-east-2:101699910734:function:gefyra-basic-lambda

Stack ARN:
arn:aws:cldformation:us-east-2:101699910734:stack/basicLambdaStack/f9226dc0-4005-11ec-a051-0ab778f82978
gefyraS3Stack
This deployment will make potentially sensitive changes according to your current security approval level (--require-approval broadening).
Please confirm you intend to make the following modifications:
```

**IAM Statement Changes**

Resource	Effect	Action	Principal	Condition
+ \${BucketNotificationsHandler050a0587b7544547bf325f094a3db834/Role.Arn}	Allow	sts:AssumeRole	Service:lambda.amazonaws.com	
+ *	Allow	s3:PutBucketNotification	AWS:\${BucketNotificationsHandler050a0587b7544547bf325f094a3db834/Role}	
+ {"Fn::ImportValue": "basicLambdaStack:ExportsOutputFnGetAttgefyrabasiclambda456DE80BArn7"}	Allow	lambda:InvokeFunction	Service:s.s3.amazonaws.com	"ArnLike": {"AWS:SourceArn": "\${gefyra-data-collection-dev.Arn}"}



Code editor showing Gefyra-CDK-Demo code and IAM Policy Changes:

```

bin > TS gefyra-cdk-demo.ts > (s) s3_bucket_stack.ts > lambdaFunction
1  #!/usr/bin/env node
2  import 'source-map-support/register';
3  import * as cdk from '@aws-cdk/core';
4  //import { GefyraCdkDemoStack } from '../lib/gefyra-cdk-demo-stack';
5  import { S3BucketStack } from '../lib/s3-bucket-stack';
6  import { basicLambdaStack } from '../lib/basic-lambda-stack';
7
8  const app = new cdk.App();
9  //new GefyraCdkDemoStack(app, 'GefyraCdkDemoStack');
10

```

**IAM Policy Changes**

Resource	Managed Policy ARN
+\$BucketNotificationsHandler050a0587b7544547bf325f094a3db834/Role	arn:\${AWS::Partition}:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole

(NOTE: There may be security-related changes not in this list. See <https://github.com/aws/aws-cdk/issues/1299>)

Do you wish to deploy these changes (y/n)? y  
gefyraS3Stack: deploying...



AWS S3 console showing the gefyra-data-collection-dev bucket:

- Buckets
  - Access Points
  - Object Lambda Access Points
  - Multi-Region Access Points
  - Batch Operations
  - Access analyzer for S3
- Block Public Access settings for this account
- Storage Lens
- Dashboards
- AWS Organizations settings
- Feature spotlight
- AWS Marketplace for S3

Objects (0)

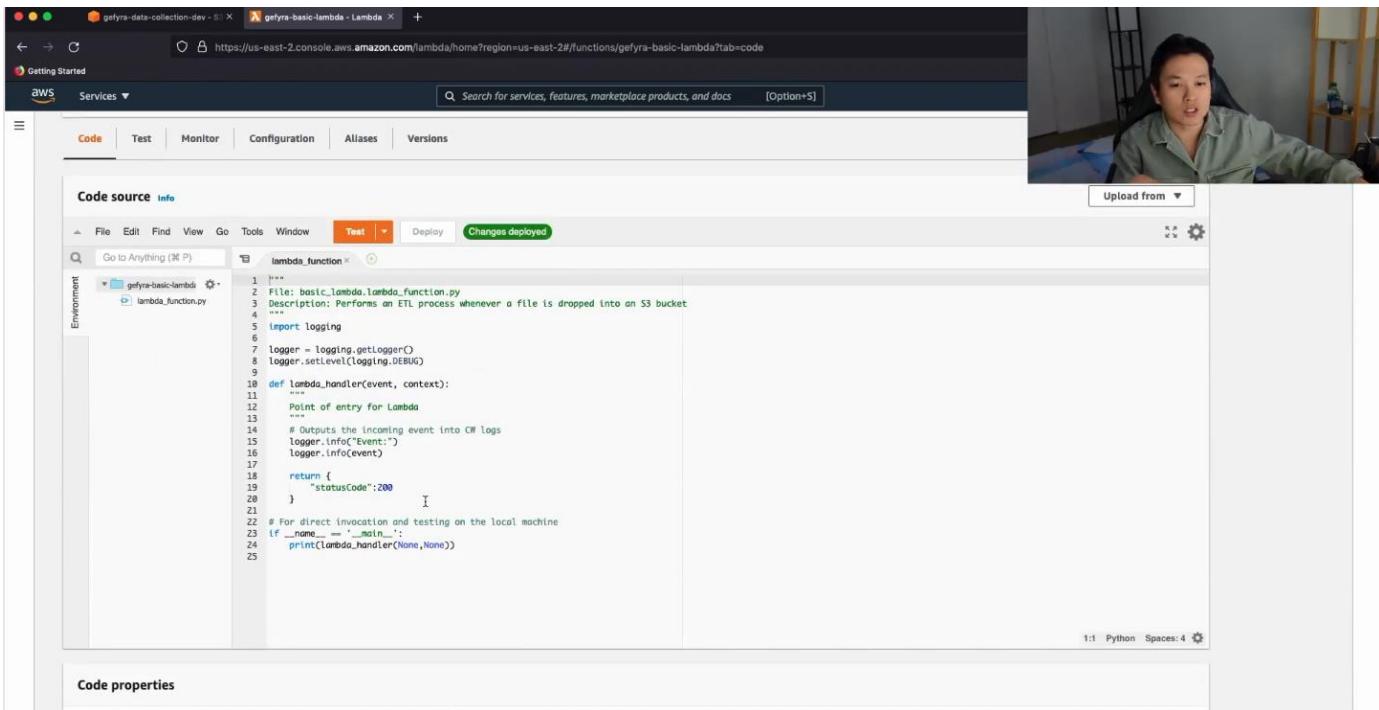
No objects

Upload



AWS Lambda console showing Functions (2):

Function name	Description	Package type	Runtime	Code size	Last modified
gefyraS3Stack-BucketNotificationsHandler050a0587b7-4FGvrNAe7u6	AWS CloudFormation handler for "Custom:S3BucketNotifications" resources (@aws-cdk/aws-s3)	Zip	Python 3.7	1.4 kB	1 day ago
gefyra-basic-lambda	-	Zip	Python 3.8	392.0 byte	1 day ago



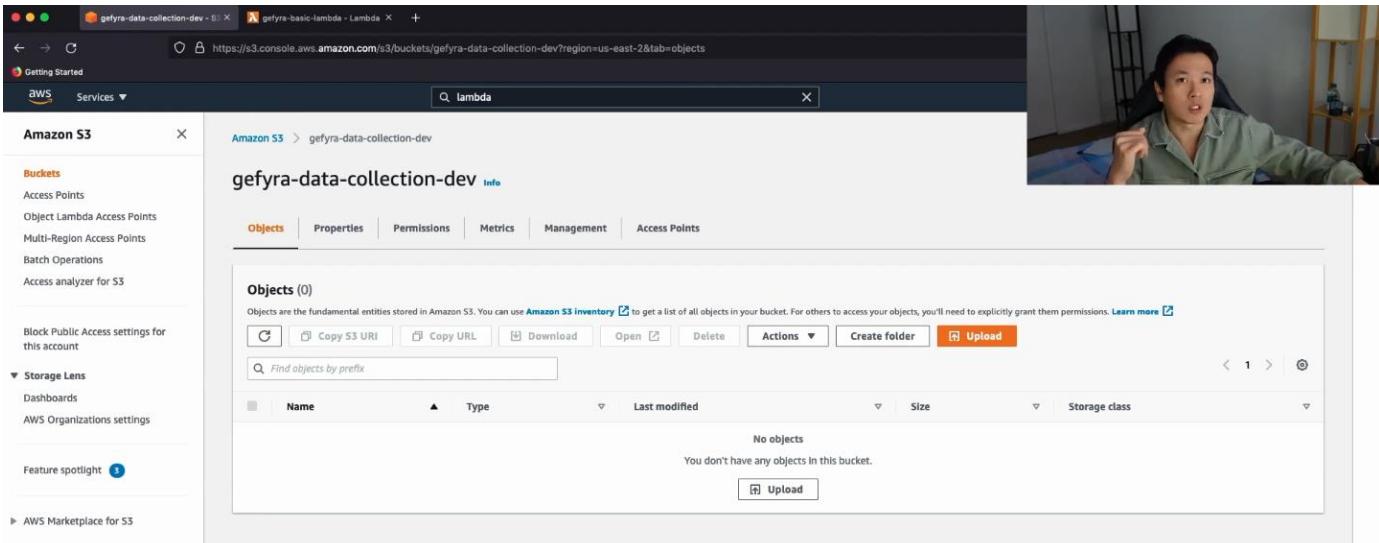
The screenshot shows the AWS Lambda console interface. In the top navigation bar, the URL is https://us-east-2.console.aws.amazon.com/lambda/home?region=us-east-2#/functions/gefyra-basic-lambda?tab=code. The main area displays the code source for the 'gefyra-basic-lambda' function. The code is a Python script named 'lambda\_function.py':

```

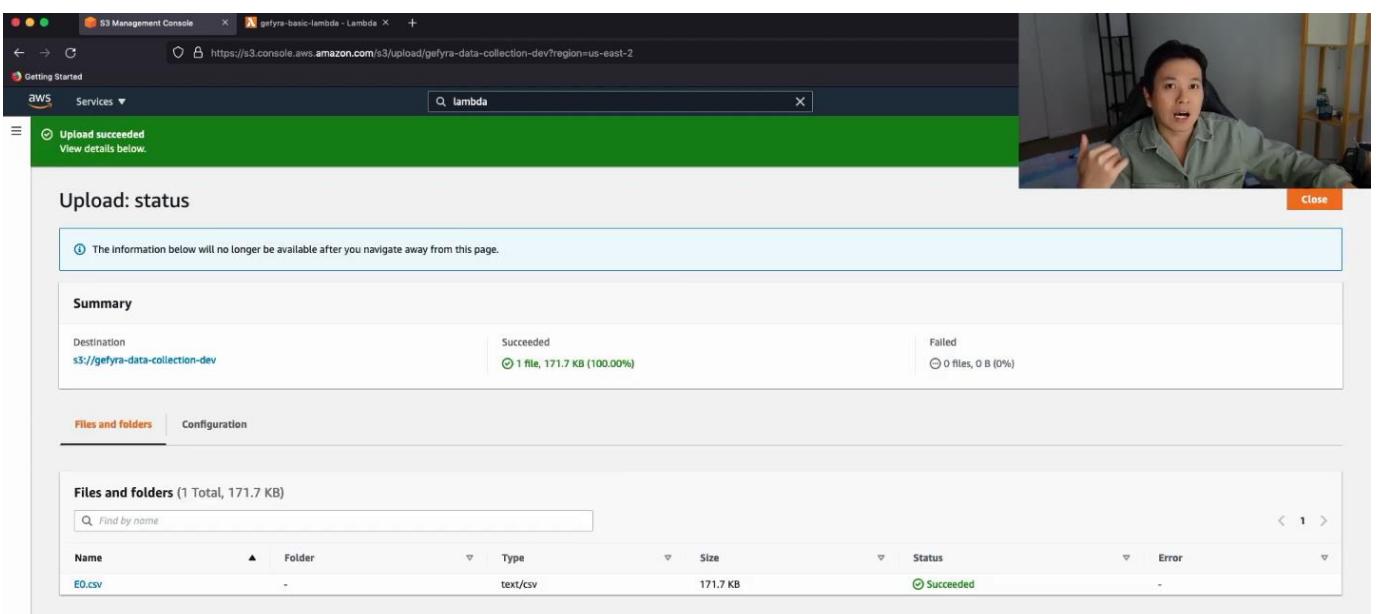
1  """
2  File: basic_lambda_function.py
3  Description: Performs an ETL process whenever a file is dropped into an S3 bucket
4  """
5  import logging
6
7  logger = logging.getLogger()
8  logger.setLevel(logging.DEBUG)
9
10 def lambda_handler(event, context):
11     """
12     Point of entry for Lambda
13
14     # Outputs the incoming event into CloudWatch logs
15     logger.info("Event: %s")
16     logger.info(event)
17
18     return {
19         "statusCode": 200
20     }
21
22 # For direct invocation and testing on the local machine
23 if __name__ == '__main__':
24     print(lambda_handler(None, None))

```

The interface includes tabs for 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions'. A 'Test' button is highlighted. Below the code editor, there's a 'Code properties' section.



The screenshot shows the AWS S3 console interface. The URL is https://s3.console.aws.amazon.com/s3/buckets/gefyra-data-collection-dev?region=us-east-2&tab=objects. The left sidebar shows the 'Amazon S3' navigation pane with options like 'Buckets', 'Access Points', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', 'Access analyzer for S3', 'Block Public Access settings for this account', 'Storage Lens', 'Dashboards', 'AWS Organizations settings', 'Feature spotlight', and 'AWS Marketplace for S3'. The main area shows the 'gefyra-data-collection-dev' bucket. The 'Objects' tab is selected, displaying a table with columns: Name, Type, Last modified, Size, and Storage class. A message at the bottom states 'No objects' and 'You don't have any objects in this bucket.'

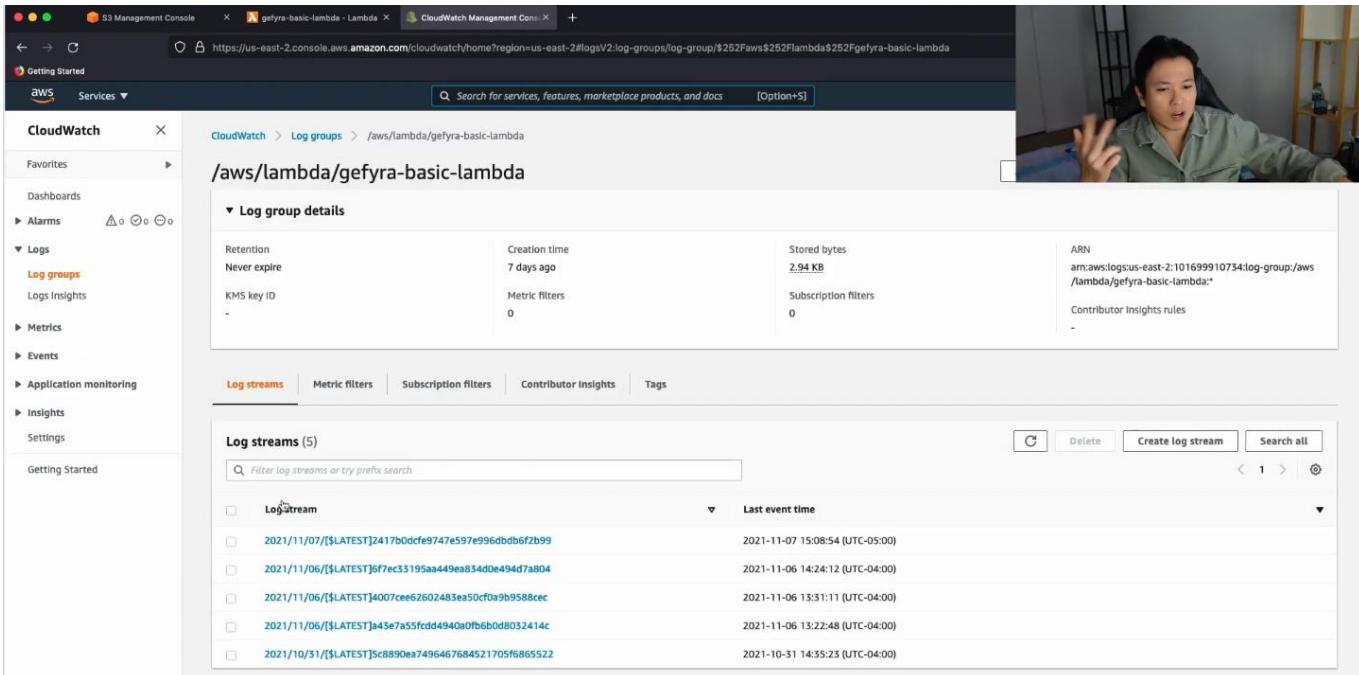
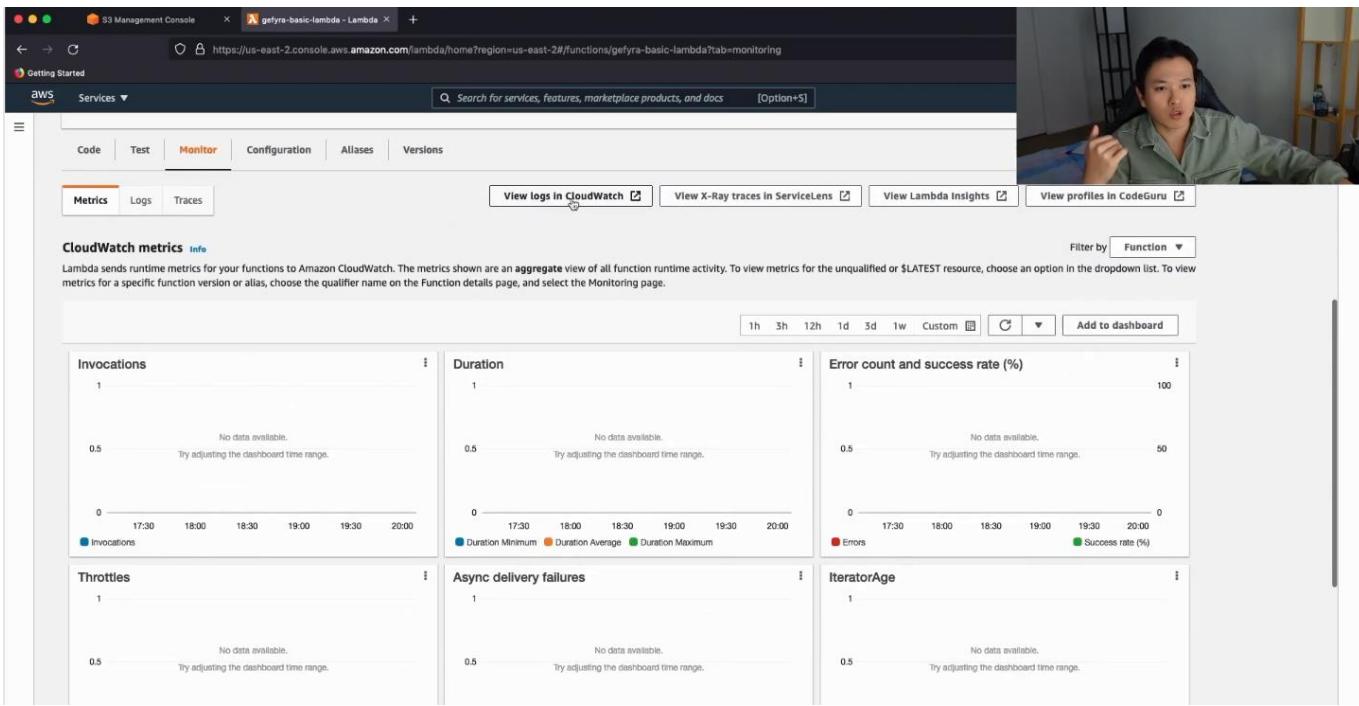


The screenshot shows the AWS S3 Management Console interface. The URL is https://s3.console.aws.amazon.com/s3/upload/gefyra-data-collection-dev?region=us-east-2. The top bar shows a green success message: 'Upload succeeded' with a link to 'View details below.' The main area shows the 'Upload: status' summary. It indicates 'Destination' as 's3://gefyra-data-collection-dev' and 'Status' as 'Succeeded' with '1 file, 171.7 KB (100.00%)'. Below this, the 'Files and folders' section shows a table with one item: 'E0.csv' (text/csv, 171.7 KB, Status: Succeeded).

The screenshot shows the AWS S3 Management Console with a successful upload message: "Upload: status" and "Upload succeeded". The summary table shows one succeeded file (EO.csv) and zero failed files. The "Files and folders" tab is selected, displaying a single file named EO.csv with a size of 171.7 KB and a status of "Succeeded".

We drop a file into the S3 bucket, let us check to see if the Lambda notification event was triggered

The screenshot shows the AWS Lambda function details page for "gefyra-basic-lambda". The "Function overview" section shows the function name, ARN, and application stack. It also lists triggers, including an S3 trigger. The "Monitor" tab is selected, showing CloudWatch metrics and logs. The CloudWatch metrics section includes links to view logs, X-Ray traces, Lambda Insights, and CodeGuru.



S3 Management Console    gefyra-basic-lambda - Lambda    CloudWatch Management Con...

Getting Started

CloudWatch Services

CloudWatch Log groups /aws/lambda/gefyra-basic-lambda 2021/11/07/[LATEST]2417b0dcfe9747e597e996dbd6f2b99

Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

Filter events Clear 1m 30m 1h 12h Custom

	Timestamp	Message
▶	2021-11-07T15:08:54.078-05:00	START RequestId: 9bc784d5-f917-4dc9-8869-8fd3f7b03f10 Version: \$LATEST
▶	2021-11-07T15:08:54.079-05:00	[INFO] 2021-11-07T20:08:54.079Z 9bc784d5-f917-4dc9-8869-8fd3f7b03f10 Event:
▼	2021-11-07T15:08:54.079-05:00	[INFO] 2021-11-07T20:08:54.079Z 9bc784d5-f917-4dc9-8869-8fd3f7b03f10 [Records]: [{"eventVersion": "2.1", "eventSource": "aws:s3", "awsRegion": "us-east-2", "eventTime": "2021-11-07T20:08:54.079Z", "requestParameters": {"sourceIPAddress": "192.168.1.10"}, "responseElements": {"x-amz-request-id": "F4E9Y5MHRQWBMV5IN", "x-amzn-requestid": "7C72F7D77177B1A"}, "configurationId": "0GRLYz12Y2kHOUW100Yz2NxLTpx2TxKMGN12M0NzRn2JZM2", "bucket": {"name": "gefyra-data-collection-dev", "ownerIdentity": {"principalId": "AUYF9QMF8Z1SP"}, "arn": "arn:aws:s3:::gefyra-data-collection-dev", "object": {"key": "10.csv", "size": 175815, "eTag": "#86f6fc92228108bb909e59f53aa", "sequencer": "#061883254122842c3"}}]}]
▶	2021-11-07T15:08:54.088-05:00	END RequestId: 9bc784d5-f917-4dc9-8869-8fd3f7b03f10
▶	2021-11-07T15:08:54.088-05:00	REPORT RequestId: 9bc784d5-f917-4dc9-8869-8fd3f7b03f10 Duration: 1.50 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 39 MB Init Duration: 111.20 ms

No newer events at this moment. Auto retry paused. Resume

Getting Started

Code Test Monitor Configuration Aliases Versions

Test event

Invoke your function with a test event. Choose a template that matches the service that triggers your function, or enter your event document in JSON.

New event

Saved event

Template

hello-world

Name

s3notification

```
1 {"Records": [{"eventVersion": "2.1", "eventSource": "aws:s3", "awsRegion": "us-east-2", "eventTime": "2021-11-07T20:08:52.119Z", "eventName": "ObjectCreated:Put", "userIdentity": {}}
```

Getting Started

Code Test Monitor Configuration Aliases Versions

Test event

Invoke your function with a test event. Choose a template that matches the service that triggers your function, or enter your event document in JSON.

New event

Saved event

Feedback English (US) ▾

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application

Cookie "notflush\_awsecm" will be soon rejected because it has the "SameSite" attribute set to "None" or an invalid value, without the "secure" attribute. To know more about the "SameSite" attribute, read <https://developer.mozilla.org/docs/Web/HTTP/Headers> [aws-global-nav.js:2:30566](#)

Cookie "aws-authTimer" has been rejected because it is already expired.

The script from "<https://phd.awx.amazon.com/auth/state/hashHeaders>" was loaded even though its MIME type ("text/plain") is not a valid JavaScript MIME type. [\[Learn More\]](#)

An iframe which has both allow-scripts and allow-same-origin for its sandbox attribute can remove its sandboxing.

Content Security Policy: Ignoring "x-frame-options" because of "frame-ancestors" directive.

An iframe which has both allow-scripts and allow-same-origin for its sandbox attribute can remove its sandboxing.

Content Security Policy: Ignoring "x-frame-options" because of "frame-ancestors" directive.

An iframe which has both allow-scripts and allow-same-origin for its sandbox attribute can remove its sandboxing.

Content Security Policy: Ignoring "x-frame-options" because of "frame-ancestors" directive.

An iframe which has both allow-scripts and allow-same-origin for its sandbox attribute can remove its sandboxing.

This page uses the non standard property "zoom". Consider using calc() in the relevant property values, or using "transform" along with "transform-origin: 0 0".

Window.controllers/Controllers are deprecated. Do not use it for UA detection.

Element.setCapture() is deprecated. Use Element.setPointerCapture() instead. For more help <https://developer.mozilla.org/docs/Web/API/Element/releasePointerCapture>

```
console.log(DOM.getAttribute("Records", "eventVersion", "2.1", "eventName", "aws:s3", "awsRegion", "us-east-2", "eventTime", "2021-11-07T20:08:52.119Z", "eventName", "ObjectCreated:Put", "userIdentity", {"principalId": "AUYF9QMF8Z1SP"}, "requestParameters", {"sourceIPAddress": "192.168.1.10"}, "responseElements", {"x-amz-request-id": "7C72F7D77177B1A"}, "configurationId", "0GRLYz12Y2kHOUW100Yz2NxLTpx2TxKMGN12M0NzRn2JZM2", "bucket", {"name": "gefyra-data-collection-dev", "ownerIdentity": {"principalId": "AUYF9QMF8Z1SP"}, "arn": "arn:aws:s3:::gefyra-data-collection-dev", "object": {"key": "10.csv", "size": 175815, "eTag": "#86f6fc92228108bb909e59f53aa", "sequencer": "#061883254122842c3"}))
```

Records: [{"eventVersion": "2.1", "eventSource": "aws:s3", "awsRegion": "us-east-2", "eventTime": "2021-11-07T20:08:52.119Z", "eventName": "ObjectCreated:Put", "userIdentity": {"principalId": "AUYF9QMF8Z1SP"}, "requestParameters": {"sourceIPAddress": "192.168.1.10"}, "responseElements": {"x-amz-request-id": "7C72F7D77177B1A"}, "configurationId": "0GRLYz12Y2kHOUW100Yz2NxLTpx2TxKMGN12M0NzRn2JZM2", "bucket": {"name": "gefyra-data-collection-dev", "ownerIdentity": {"principalId": "AUYF9QMF8Z1SP"}, "arn": "arn:aws:s3:::gefyra-data-collection-dev", "object": {"key": "10.csv", "size": 175815, "eTag": "#86f6fc92228108bb909e59f53aa", "sequencer": "#061883254122842c3"}})}]

undefined

The screenshot shows the AWS Lambda console interface. At the top, there are three tabs: 'gefyr-data-collection-dev' (selected), 'gefyr-basic-lambda - Lambda' (highlighted in blue), and 'CloudWatch Management Console'. Below the tabs, a banner indicates 'The test event s3notification was successfully saved.' The main area is titled 'Test event' and contains a code editor with the following JSON input:

```
1 - {
2 -   "Records": [
3 -     {
4 -       "eventVersion": "2.1",
5 -       "eventSource": "aws:s3",
6 -       "awsRegion": "us-east-2",
7 -       "eventTime": "2021-11-07T20:08:52.119Z",
8 -       "eventName": "ObjectCreated:Put",
9 -       "userIdentity": {
10 -         "principalId": "A1YF9QMFZ15BP"
11 -       },
12 -       "requestParameters": {
13 -         "sourceIPAddress": "73.125.109.75"
14 -       },
15 -       "responseElements": {
16 -         "x-amz-request-id": "EKE0V5MNROQMF51N",
17 -         "x-amz-id-2": "LBCPvva5/YbZWNLyG37b271Nm6rJJsrssZnezoTl+KVF7t2FkkWxPMmf/wNNsF5zdulhCh/TywfDUxaUoT8huoKK1C"
18 -     }
19 -   ]
20 - }
```

Below the code editor are buttons for 'Delete', 'Format', 'Save changes', and 'Test'. The 'Test' button is highlighted in orange. The bottom of the page includes standard AWS navigation links: Feedback, English (US), Inspector, Console, Debugger, Network, Style Editor, Performance, Memory, Storage, Accessibility, Application, Errors, Warnings, Logs, Info, Debug, CSS, XHR, Requests, and links for Privacy Policy, Terms of Use, and Cookie preferences.

This screenshot is similar to the one above, showing the AWS Lambda console after a successful test execution. The main difference is the presence of a modal window titled 'Execution result: succeeded (logs)'. The modal contains a single link labeled 'Details'. The rest of the interface is identical to the first screenshot, including the test event configuration and the browser's developer tools at the bottom.

The test event s3notification was successfully saved.

The area below shows the result returned by your function execution. [Learn more](#) about returning results from your function.

```
{ "statusCode": 200 }
```

### Summary

Code SHA-256	Request ID
Lw/X04Hjb9XhB8HISDdvZT3dcqkPYws7JhlqTTvGUYM=	8b5f0a75-5015-4635-bc83-e6025614ab55
Duration	Billed duration
1.43 ms	2 ms
Resources configured	Max memory used
128 MB	39 MB

### Log output

The section below shows the logging calls in your code. [Click here](#) to view the corresponding CloudWatch log group.

```
START RequestId: 8b5f0a75-5015-4635-bc83-e6025614ab55 Version: $LATEST
[INFO] 2021-11-07T02:12:18.843Z          8b5f0a75-5015-4635-bc83-e6025614ab55  Event:
[INFO] 2021-11-07T02:12:18.843Z          8b5f0a75-5015-4635-bc83-e6025614ab55  {"Records": [{"eventVersion": "2.1", "eventSource": "aws:s3", "awsRegion": "us-east-2", "eventTime": "2021-11-07T02:12:08:52.119Z", "eventName": "ObjectCreated:Put", "userIdentity": {"principalId": "A1YF9MBPZ18BP"}, "requestParameters": {"sourceIPAddress": "73.125.169.75"}, "responseElements": {"x-amz-request-id": "EKE0W5NRRQMF51N", "x-amz-id-2": "LBCPvw3y/Yz2mNLy3gtZ71m6rJlsrsszNezo2TL+KVF7z2fkWnPMMf/wBNsF5zdulhCh/lyfuDUxalo08ueoKK1C"}, "s3": {"s3SchemaVersion": "1.0", "configurationId": "0GRYiUz0YzktMDUu000YzWmLTxpZTkMGNyIz0M3NzNm2JM2", "bucket": {"name": "gefrya-data-collection-dev", "ownerIdentity": {"principalId": "A1YF9MBPZ18BP"}, "arn": "arn:aws:s3:::gefrya-data-collection-dev"}, "object": {"key": "E0.csv", "size": 175815, "eTag": "f8a6fc6922c8108bbdb96e5e59f53aa0", "sequencer": "0061883254122842C3"}}]}
END RequestId: 8b5f0a75-5015-4635-bc83-e6025614ab55
REPORT RequestId: 8b5f0a75-5015-4635-bc83-e6025614ab55 Duration: 1.43 ms          Billed Duration: 2 ms      Memory Size: 128 MB      Max Memory Used: 39 MB
```

Feedback English (US) ▾ © 2008–2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application Errors Warnings Logs Info Debug CSS XHR Requests

Filter Output

```
["#0aefc6922c8108bbdb96e5e59f53aa0", "sequencer": "0061883254122842C3"}]}))}))
```

```
{"Records": [{"eventVersion": "2.1", "eventSource": "aws:s3", "awsRegion": "us-east-2", "eventTime": "2021-11-07T02:12:08:52.119Z", "eventName": "ObjectCreated:Put", "userIdentity": {"principalId": "A1YF9MBPZ18BP"}, "requestParameters": {"sourceIPAddress": "73.125.169.75"}, "responseElements": {"x-amz-request-id": "EKE0W5NRRQMF51N", "x-amz-id-2": "LBCPvw3y/Yz2mNLy3gtZ71m6rJlsrsszNezo2TL+KVF7z2fkWnPMMf/wBNsF5zdulhCh/lyfuDUxalo08ueoKK1C"}, "s3": {"s3SchemaVersion": "1.0", "configurationId": "0GRYiUz0YzktMDUu000YzWmLTxpZTkMGNyIz0M3NzNm2JM2", "bucket": {"name": "gefrya-data-collection-dev", "ownerIdentity": {"principalId": "A1YF9MBPZ18BP"}, "arn": "arn:aws:s3:::gefrya-data-collection-dev"}, "object": {"key": "E0.csv", "size": 175815, "eTag": "f8a6fc6922c8108bbdb96e5e59f53aa0", "sequencer": "0061883254122842C3"}}]}))})
```

```
< undefined
```

This helps with unit testing. We have now connected the S3 bucket to the Lambda function and made sure that it works.