

Develop, Test & Deploy TypeScript Lambda Functions Using AWS CDK



Prasad Domala
11.1K subscribers

Subscribe

87



Share

Download



7,194 views Apr 21, 2022 AWS CDK

This video explains the process of developing, testing and deploying TypeScript based Lambda functions using AWS CDK.

Please refer to below Git repo for the code I used in this video.

You can find all the commands and URLs in the README.md file within the repo.

<https://github.com/kheriox-technologi...>

Develop, Test & Deploy TypeScript Lambda Functions Using AWS CDK



https://github.com/kheriox-technologies/yt-typescript-lambda-cdk/tree/develop

kheriox-technologies / yt-typescript-lambda-cdk

Type to search

Code Issues Pull requests Actions Projects Security Insights

yt-typescript-lambda-cdk Public

Watch 1 Fork 2 Star 6

develop 1 branch 0 tags

Go to file Add file Code

kheriox-technologies Added Youtube video link a249e4d on Apr 21, 2022 3 commits

File	Description	Last Commit
bin	Initial upload	last year
lambda-handlers	Initial upload	last year
lambda-layer	Initial upload	last year
lib	Initial upload	last year
.gitignore	Initial upload	last year
.npmignore	Initial upload	last year
README.md	Added Youtube video link	last year
cdk.context.json	Initial upload	last year
cdk.json	Initial upload	last year
jest.config.js	Initial upload	last year
package-lock.json	Initial upload	last year
package.json	Initial upload	last year
template.yaml	Initial upload	last year
tsconfig.json	Initial upload	last year
types.d.ts	Initial upload	last year

About

Deploy Typescript Based Lambda Functions Using CDK

Readme Activity 6 stars 1 watching 2 forks Report repository

Releases

No releases published

Packages

No packages published

Languages

TypeScript 83.7% JavaScript 16.3%

<https://github.com/kheriox-technologies/yt-typescript-lambda-cdk/tree/develop>

<https://github.com/kheriox-technologies/yt-typescript-lambda-cdk/blob/develop/README.md>

kheriox-technologies / yt-typescript-lambda-cdk

Code Issues Pull requests Actions Projects Security Insights

Files

develop

Go to file

- bin
- lambda-handlers
- lambda-layer
- lib
- .gitignore
- .npmignore
- README.md
- cdk.context.json
- cdk.json
- jest.config.js
- package-lock.json
- package.json
- template.yaml
- tsconfig.json
- types.d.ts

yt-typescript-lambda-cdk / README.md

kheriox-technologies Added Youtube video link

a249e4d · last year History

Preview Code Blame 43 lines (27 loc) · 1007 Bytes Code 55% faster with GitHub Copilot

Raw

Deploy Typescript based Lambda functions using AWS CDK

This is a companion repo for my youtube video explaining the process of developing, testing and deploying Typescript based Lambda functions using AWS CDK

You can find all the command and related links I used in the video

Youtube Video: <https://youtu.be/DUNEuhQ1k4>

URLs

Multi Env CDK Youtube Video: https://youtu.be/H7Ynxkk_jss

Git Repo: <https://github.com/kheriox-technologies/yt-typescript-lambda-cdk>

AWS Profile NPM Package: <https://www.npmjs.com/package/awsprofile>

Git Branch NPM Package: <https://www.npmjs.com/package/git-branch>

AWS CDK Documentation: <https://docs.aws.amazon.com/cdk/api/v2/docs/aws-construct-library.html>

Tools & Technologies

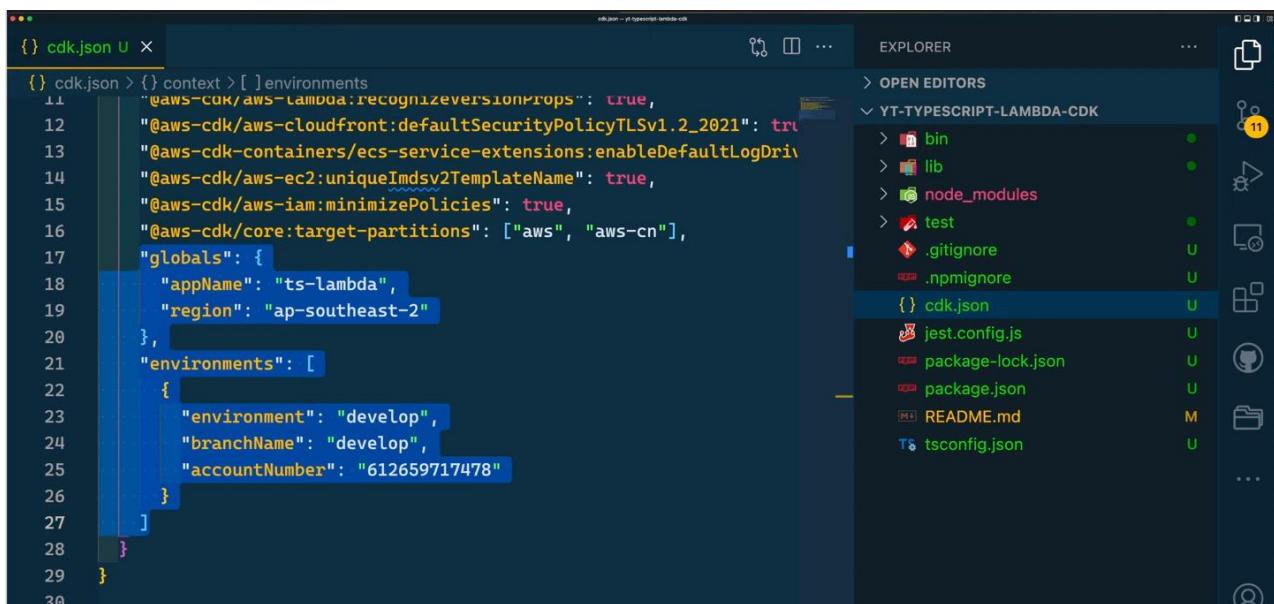
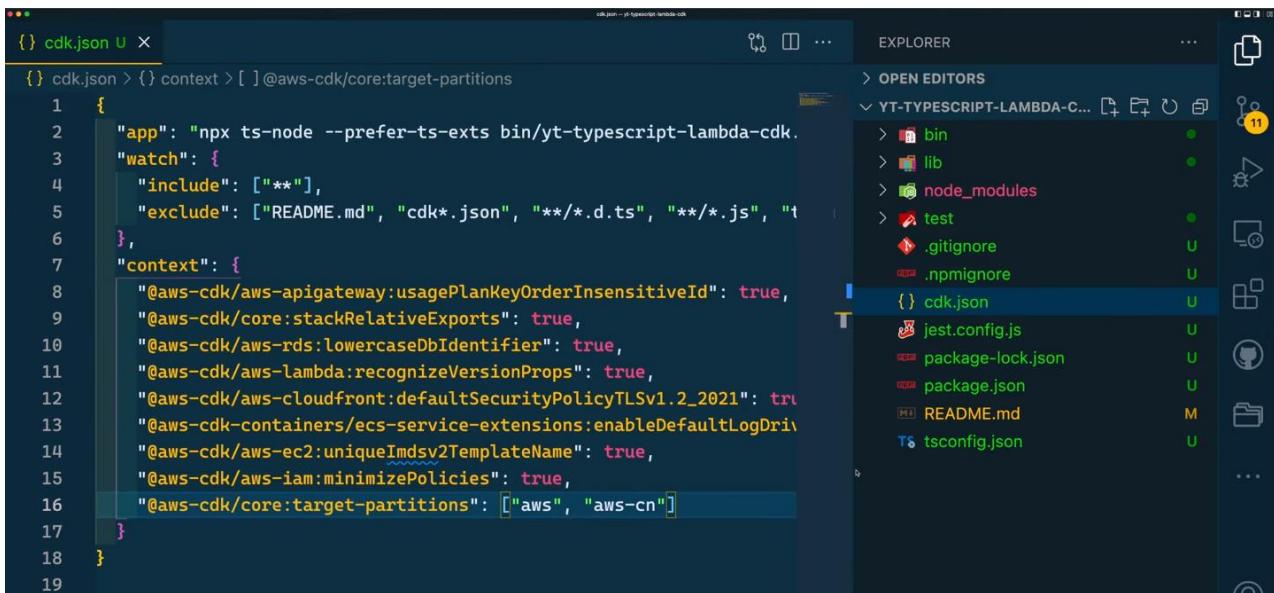
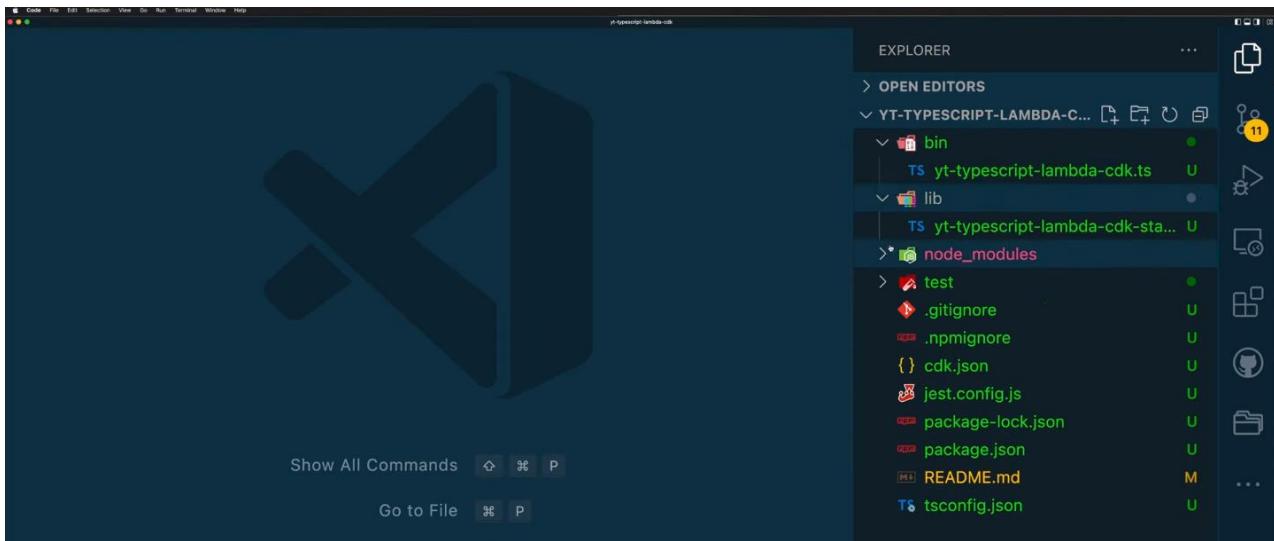


NODEJS + TypeScript

Public & Private (VPC)
Lambda Layers
SAM Local

CDK Version 2

CDK PROJECT STRUCTURE



The screenshot shows the VS Code interface with the 'types.d.ts' file open in the editor. The code defines a type 'CDKContext' with properties: appName, region, environment, branchName, and accountNumber. The Explorer sidebar shows the project structure for 'YT-TYPESCRIPT-LAMBDA-CDK', including files like cdk.json, jest.config.js, package-lock.json, package.json, README.md, tsconfig.json, and types.d.ts.

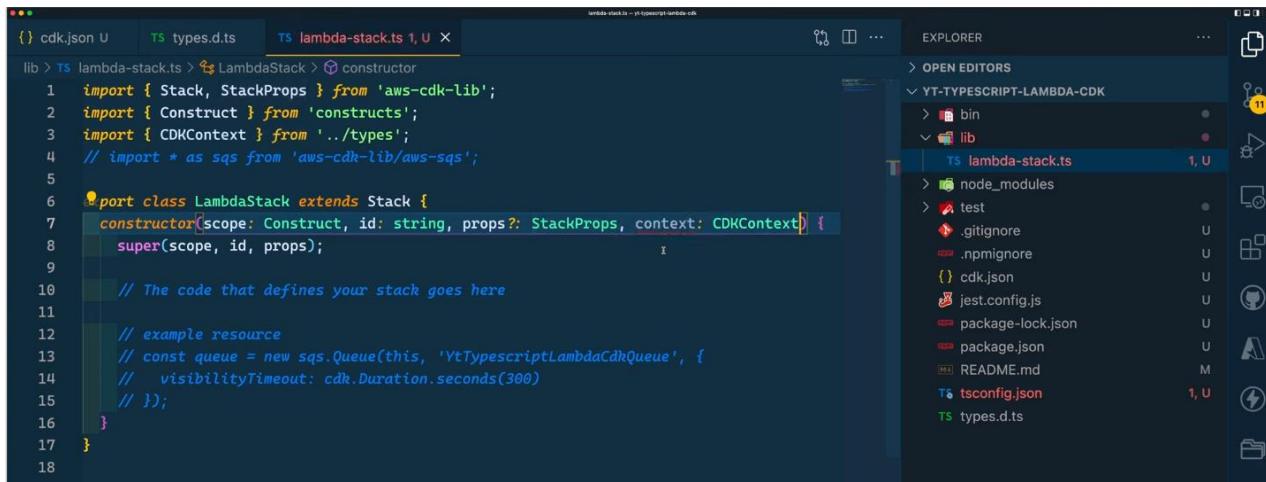
```
1 export type CDKContext = {
2   appName: string;
3   region: string;
4   environment: string;
5   branchName: string;
6   accountNumber: string;
7 };
```

The screenshot shows the 'yt-typescript-lambda-cdk-stack.ts' file open in the editor. It imports 'Stack' and 'Construct' from 'aws-cdk-lib'. It defines a class 'YtTypescriptLambdaCdkStack' that extends 'Stack'. The constructor takes a 'scope', 'id', and optional 'props'. The code then defines a stack, creating a queue with a visibility timeout of 300 seconds. The Explorer sidebar shows the project structure for 'YT-TYPESCRIPT-LAMBDA-CDK'.

```
1 import { Stack, StackProps } from 'aws-cdk-lib';
2 import { Construct } from 'constructs';
3 // import * as sqs from 'aws-cdk-lib/aws-sqs';
4
5 export class YtTypescriptLambdaCdkStack extends Stack {
6   constructor(scope: Construct, id: string, props?: StackProps) {
7     super(scope, id, props);
8
9     // The code that defines your stack goes here
10
11    // example resource
12    // const queue = new sqs.Queue(this, 'YtTypescriptLambdaCdkQueue', {
13    //   visibilityTimeout: cdk.Duration.seconds(300)
14    // });
15  }
16}
17
```

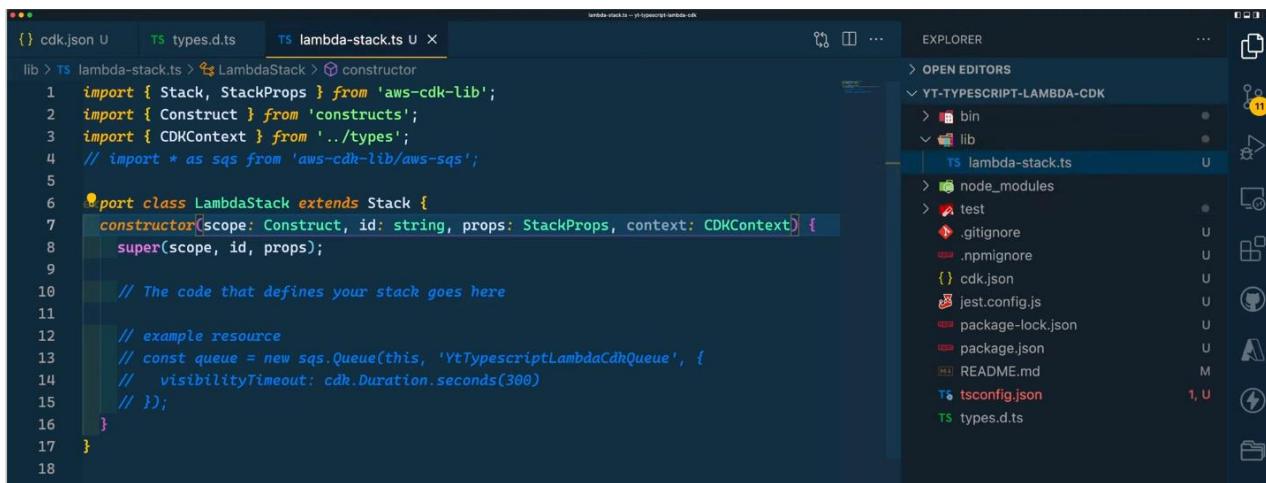
The screenshot shows the 'lambda-stack.ts' file open in the editor. It imports 'Stack' and 'Construct' from 'aws-cdk-lib'. It defines a class 'LambdaStack' that extends 'Stack'. The constructor takes a 'scope', 'id', and optional 'props'. The code then defines a stack, creating a queue with a visibility timeout of 300 seconds. The Explorer sidebar shows the project structure for 'YT-TYPESCRIPT-LAMBDA-CDK'.

```
1 import { Stack, StackProps } from 'aws-cdk-lib';
2 import { Construct } from 'constructs';
3 // import * as sqs from 'aws-cdk-lib/aws-sqs';
4
5 export class LambdaStack extends Stack {
6   constructor(scope: Construct, id: string, props?: StackProps) {
7     super(scope, id, props);
8
9     // The code that defines your stack goes here
10
11    // example resource
12    // const queue = new sqs.Queue(this, 'YtTypescriptLambdaCdkQueue', {
13    //   visibilityTimeout: cdk.Duration.seconds(300)
14    // });
15  }
16}
17
```



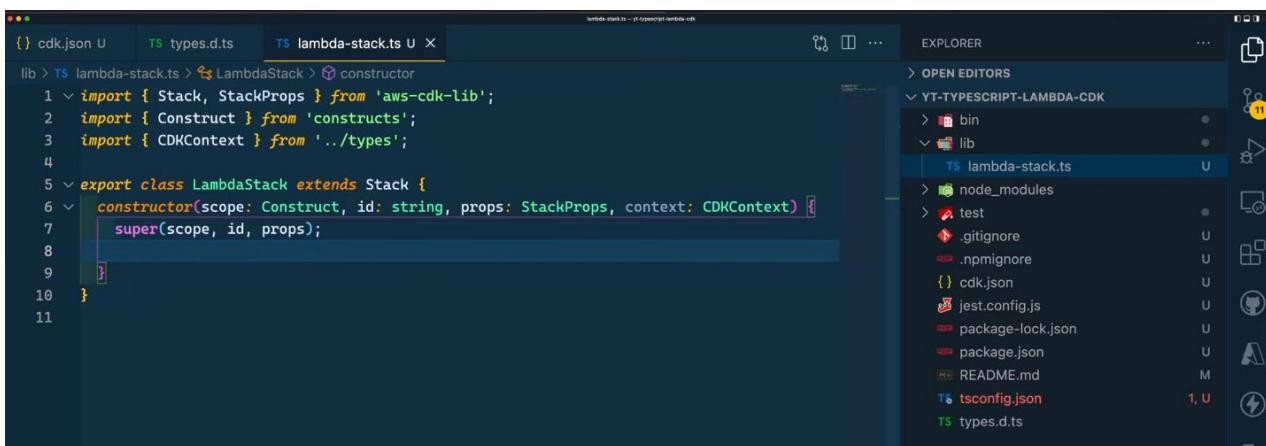
```
lambda-stack.ts - yt-typescript-lambda-cdk
{} cdk.json U TS types.d.ts TS lambda-stack.ts 1, U ...
lib > TS lambda-stack.ts > LambdaStack > constructor
1 import { Stack, StackProps } from 'aws-cdk-lib';
2 import { Construct } from 'constructs';
3 import { CDKContext } from '../types';
4 // import * as sqs from 'aws-cdk-lib/aws-sqs';
5
6 export class LambdaStack extends Stack {
7   constructor(scope: Construct, id: string, props?: StackProps, context: CDKContext) {
8     super(scope, id, props);
9
10    // The code that defines your stack goes here
11
12    // example resource
13    // const queue = new sqs.Queue(this, 'YtTypescriptLambdaCdkQueue', {
14    //   visibilityTimeout: cdk.Duration.seconds(300)
15    // });
16  }
17}
18
```

The screenshot shows the VS Code interface with the LambdaStack.ts file open. The code defines a LambdaStack class that extends the Stack class from the aws-cdk-lib library. It includes a constructor that takes a scope, id, props, and context. A comment indicates where the stack definition code will go. An example resource, a SQS queue, is shown but commented out.



```
lambda-stack.ts - yt-typescript-lambda-cdk
{} cdk.json U TS types.d.ts TS lambda-stack.ts U ...
lib > TS lambda-stack.ts > LambdaStack > constructor
1 import { Stack, StackProps } from 'aws-cdk-lib';
2 import { Construct } from 'constructs';
3 import { CDKContext } from '../types';
4 // import * as sqs from 'aws-cdk-lib/aws-sqs';
5
6 export class LambdaStack extends Stack {
7   constructor(scope: Construct, id: string, props: StackProps, context: CDKContext) {
8     super(scope, id, props);
9
10    // The code that defines your stack goes here
11
12    // example resource
13    // const queue = new sqs.Queue(this, 'YtTypescriptLambdaCdkQueue', {
14    //   visibilityTimeout: cdk.Duration.seconds(300)
15    // });
16  }
17}
18
```

This screenshot is identical to the first one, showing the same code in the LambdaStack.ts file.



```
lambda-stack.ts - yt-typescript-lambda-cdk
{} cdk.json U TS types.d.ts TS lambda-stack.ts U ...
lib > TS lambda-stack.ts > LambdaStack > constructor
1 import { Stack, StackProps } from 'aws-cdk-lib';
2 import { Construct } from 'constructs';
3 import { CDKContext } from '../types';
4
5 export class LambdaStack extends Stack {
6   constructor(scope: Construct, id: string, props: StackProps, context: CDKContext) {
7     super(scope, id, props);
8
9   }
10}
11
```

This screenshot is identical to the previous ones, showing the same code in the LambdaStack.ts file.

We will be defining all our Lambda resources in this file later

The screenshot shows the Visual Studio Code interface. The code editor on the left displays a TypeScript file named `bin/yt-typescript-lambda-cdk.ts`. The file contains code for setting up a Lambda function using AWS CDK. The Explorer sidebar on the right lists the project structure, including files like `cdk.json`, `lambda-stack.ts`, and `tsconfig.json`. A status bar at the bottom indicates the file is 1,000 lines long.

```
1 //!/usr/bin/env node
2 import 'source-map-support/register';
3 import * as cdk from 'aws-cdk-lib';
4 import { YtTypescriptLambdaCdkStack } from '../lib/yt-typescript-lambda-cdk-stack';
5
6 const app = new cdk.App();
7 new YtTypescriptLambdaCdkStack(app, 'YtTypescriptLambdaCdkStack', {
8   /* If you don't specify 'env', this stack will be environment-agnostic.
9    * Account/Region-dependent features and context lookups will not work,
10   * but a single synthesized template can be deployed anywhere. */
11
12   /* Uncomment the next line to specialize this stack for the AWS Account
13    * and Region that are implied by the current CLI configuration. */
14   // env: { account: process.env.CDK_DEFAULT_ACCOUNT, region: process.env.CDK_DEFAULT_REGION },
15
16   /* Uncomment the next line if you know exactly what Account and Region you
17    * want to deploy the stack to. */
18   // env: { account: '123456789012', region: 'us-east-1' },
19
20   /* For more information, see https://docs.aws.amazon.com/cdk/latest/guide/environments
21 });

```

The screenshot shows the Visual Studio Code interface with the following details:

- Code Editor:** The main area displays the `bin/yt-typescript-lambda-cdk.ts` file. The code is a TypeScript script for a Lambda CDK stack, defining an application (`app`) and a stack (`YtTypescriptLambdaCdkStack`). It includes comments about environment agnosticism and account/region configuration.
- File Explorer:** On the right, the "EXPLORER" view shows the project structure:
 - bin:** Contains `yt-typescript-lambda-cdk.ts` (marked as 2 changes), `lambda-stack.ts`, and `node_modules`.
 - lib:** Contains `cdk.json`, `.gitignore`, `.npmignore`, `jest.config.js`, `package-lock.json`, `package.json`, `README.md`, `tsconfig.json`, and `types.d.ts` (marked as 1 change).
 - test:** Contains `gitBranch` and `CDKContext` imports.

The screenshot shows three separate terminal windows within VS Code, each displaying the output of an `npm install` command. The first terminal at the top shows the command `npm i git-branch` being run, followed by audit results: "added 22 packages, and audited 564 packages in 4s", "27 packages are looking for funding", and "found 0 vulnerabilities". The second terminal in the middle shows the command `npm i -D @types/git-branch` being run, followed by audit results: "added 1 package, and audited 565 packages in 1s", "27 packages are looking for funding", and "found 0 vulnerabilities". The third terminal at the bottom shows the command `npm i` being run, followed by audit results: "added 22 packages, and audited 564 packages in 4s", "27 packages are looking for funding", and "found 0 vulnerabilities". The VS Code interface includes tabs for code files like `lambda-stack.ts` and `tsconfig.json`, and an Explorer sidebar showing the project structure.

```
{ } cdk.json U TS types.d.ts TS lambda-stack.ts U TS yt-typescript-lambda-cdk.ts 2, U bin > TS yt-typescript-lambda-cdk.ts > ... 1 #!/usr/bin/env node PROBLEMS SEARCH OUTPUT DEBUG CONSOLE TERMINAL GITLENS zsh + ^ x bin > ~/Doc/Y/045/git-repo/yt-typescript-lambda-cdk > ⌂ develop !1 ?10 22:45:18 ⚡ > npm i git-branch added 22 packages, and audited 564 packages in 4s 27 packages are looking for funding run `npm fund` for details found 0 vulnerabilities bin > ~/Doc/Y/045/git-repo/yt-typescript-lambda-cdk > ⌂ develop !1 ?10 4s 23:56:37 ⚡ > npm i -D @types/git-branch added 1 package, and audited 565 packages in 1s 27 packages are looking for funding run `npm fund` for details found 0 vulnerabilities bin > ~/Doc/Y/045/git-repo/yt-typescript-lambda-cdk > ⌂ develop !1 ?10 23:56:51 ⚡ >
```

```
bin > TS yt-typescript-lambda-cdk.ts ...
1  #!/usr/bin/env node
2  import '!source-map-support/register';
3  import * as cdk from 'aws-cdk-lib';
4  import { LambdaStack } from '../lib/lambda-stack';
5  import gitBranch from 'git-branch';
6  import { CDKContext } from '../types';
7
8  const app = new cdk.App();
9  new YtTypescriptLambdaCdkStack(app, 'YtTypescriptLambdaCdkStack', {
10    /* If you don't specify 'env', this stack will be environment-agnostic.
11     * Account/Region-dependent features and context lookups will not work,
12     * but a single synthesized template can be deployed anywhere. */
13    // Uncomment the next line to specialize this stack for the AWS Account
14    // and Region that are implied by the current CLI configuration. */
15    // env: { account: process.env.CDK_DEFAULT_ACCOUNT, region: process.env.CDK_DEFAULT_REGION },
16    /* Uncomment the next line if you know exactly what Account and Region you
17     * want to deploy the stack to. */
18    // env: { account: '123456789012', region: 'us-east-1' },
19    /* For more information, see https://docs.aws.amazon.com/cdk/latest/guide/environments
20   });
21
```

```
bin > TS yt-typescript-lambda-cdk.ts ...
1  #!/usr/bin/env node
2  import '!source-map-support/register';
3  import * as cdk from 'aws-cdk-lib';
4  import { LambdaStack } from '../lib/lambda-stack';
5  import gitBranch from 'git-branch';
6  import { CDKContext } from '../types';
7
8  // Get CDK Context based on git branch
9  export const getContext = async (app: cdk.App): Promise<CDKContext> => {
10    return new Promise(async (resolve, reject) => {
11      try {
12        const currentBranch = await gitBranch();
13
14        const environment = app.node.tryGetContext('environments').find((e: any) => e.branch === currentBranch);
15
16        const globals = app.node.tryGetContext('globals');
17
18        return resolve({ ...globals, ...environment });
19      } catch (error) {
20        console.error(error);
21        return reject();
22      }
23    });
24  };
25
```

```
bin > TS yt-typescript-lambda-cdk.ts > ⚡ getContext > ⚡ <function>
1  /usr/bin/env node
2  port '!source-map-support/register';
3  port * as cdk from 'aws-cdk-lib';
4  port { LambdaStack } from '../lib/lambda-stack';
5  port gitBranch from 'git-branch';
6  port { CDKContext } from '../types';
7
8  Get CDK Context based on git branch
9  port const getContext = async (app: cdk.App): Promise<CDKContext> => {
10    return new Promise(async (resolve, reject) => {
11      try {
12        const currentBranch = await gitBranch();
13
14        const environment = app.node.tryGetContext('environments').find((e: any) => e.branch === currentBranch);
15
16        const globals = app.node.tryGetContext('globals');
17
18        return resolve({ ...globals, ...environment });
19      } catch (error) {
20        console.error(error);
21        return reject();
22      }
23    });
24  };
25
```

PRASAD DOMALA

```

{} cdk.json U   TS types.d.ts   TS lambda-stack.ts U   TS yt-typescript-lambda-cdk.ts X
bin > TS yt-typescript-lambda-cdk.ts > ...
24
25
26 // Create Stacks
27 const createStacks = async () => {
28   try {
29     const app = new cdk.App();
30     const context = await getContext(app);
31
32     const tags: any = {
33       Environment: context.environment,
34     };
35
36     const stackProps: cdk.StackProps = {
37       env: {
38         region: context.region,
39         account: context.accountNumber,
40       },
41       stackName: `${context.appName}-stack-${context.environment}`,
42       description: 'This is the Stack Description',
43       tags,
44     };
45
46     new LambdaStack(app, `${context.appName}-stack-${context.environment}`, stackProps,
47   } catch (error) {
48     console.error(error);
49   }
50 };

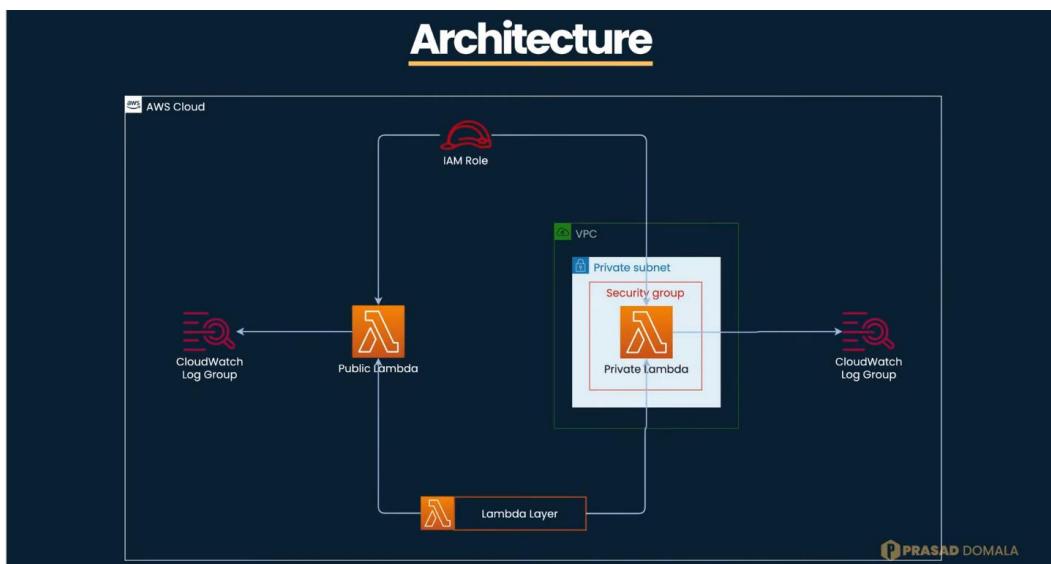
```

PRASAD DOMALA

```

{} cdk.json U   TS types.d.ts   TS lambda-stack.ts U   TS yt-typescript-lambda-cdk.ts X
bin > TS yt-typescript-lambda-cdk.ts > ...
29   const app = new cdk.App();
30   const context = await getContext(app);
31
32   const tags: any = {
33     Environment: context.environment,
34   };
35
36   const stackProps: cdk.StackProps = {
37     env: {
38       region: context.region,
39       account: context.accountNumber,
40     },
41     stackName: `${context.appName}-stack-${context.environment}`,
42     description: 'This is the Stack Description',
43     tags,
44   };
45
46   new LambdaStack(app, `${context.appName}-stack-${context.environment}`, stackProps, context);
47 } catch (error) {
48   console.error(error);
49 }
50
51 createStacks();

```



The screenshot shows the VS Code interface with multiple tabs open. The active tab is `lambda-stack.ts`, which contains the following code:

```

1 import { Stack, StackProps } from 'aws-cdk-lib';
2 import { Construct } from 'constructs';
3 import { CDKContext } from '../types';
4
5 export class LambdaStack extends Stack {
6   constructor(scope: Construct, id: string, props: StackProps, context: CDKContext) {
7     super(scope, id, props);
8   }
9 }
10

```

The Explorer sidebar on the right shows the project structure for `YT-TYPESCRIPT-LAMBDA-CDK`, including files like `lambda-stack.ts`, `node_modules`, and `tsconfig.json`.

This screenshot shows the same VS Code environment with the code for the `LambdaStack` constructor expanded. It includes imports for `aws-cdk-lib/iam` and the definition of a `lambdaRole` using the `Role` class from the AWS CDK library.

```

1 import { Stack, StackProps } from 'aws-cdk-lib';
2 import { Construct } from 'constructs';
3 import { CDKContext } from '../types';
4
5 import * as iam from 'aws-cdk-lib/aws-iam';
6
7 export class LambdaStack extends Stack {
8   constructor(scope: Construct, id: string, props: StackProps, context: CDKContext) {
9     super(scope, id, props);
10
11     // Lambda Role
12     const lambdaRole = new iam.Role(this, 'lambdaRole', {
13       roleName: `${context.appName}-lambda-role-${context.environment}`,
14       description: `Lambda role for ${context.appName}`,
15       assumedBy: new iam.ServicePrincipal('lambda.amazonaws.com'),
16       managedPolicies: [iam.ManagedPolicy.fromAwsManagedPolicyName('ReadOnlyAccess')],
17     });
18   }
19 }
20

```

The screenshot shows the AWS CDK API Reference page at <https://docs.aws.amazon.com/cdk/api/latest/typescript/index.html>. The left sidebar lists various service modules like `aws-cdk-lib`, `aws-cdk-lib.alexa_ask`, etc. The main content area is titled "API Reference" and "Modules". It explains that the AWS Construct Library is organized into modules and provides a list of module types:

- aws-xxx**: service package for the indicated service. This package will contain constructs to work with the given service.
- aws-xxx¹**: a little superscript 1 indicates that his package only contains CloudFormation Resources (for now).
- aws-xxx-targets**: integration package for the indicated service. This package will contain classes to connect the constructs in the "aws-xxx" package to other AWS services it can work with.
- xxx**: packages that don't start "aws-" are AWS CDK framework packages.

The "Module Contents" section lists the following types:

- Constructs** - All higher-level constructs in this library.
- Other Types** - All non-construct classes, interfaces, structs and enums that exist to support the constructs.
- CloudFormation Resources** - All constructs that map directly onto CloudFormation Resources. We recommend that you read the [CloudFormation Resource](#) and [Property Type Reference](#) for details on these resources.
- CloudFormation Property Types** - All structs that are used by the CloudFormation Resource constructs.

A watermark for "PRASAD DOMALA" is visible in the bottom right corner.

The screenshot shows the VS Code interface with the following details:

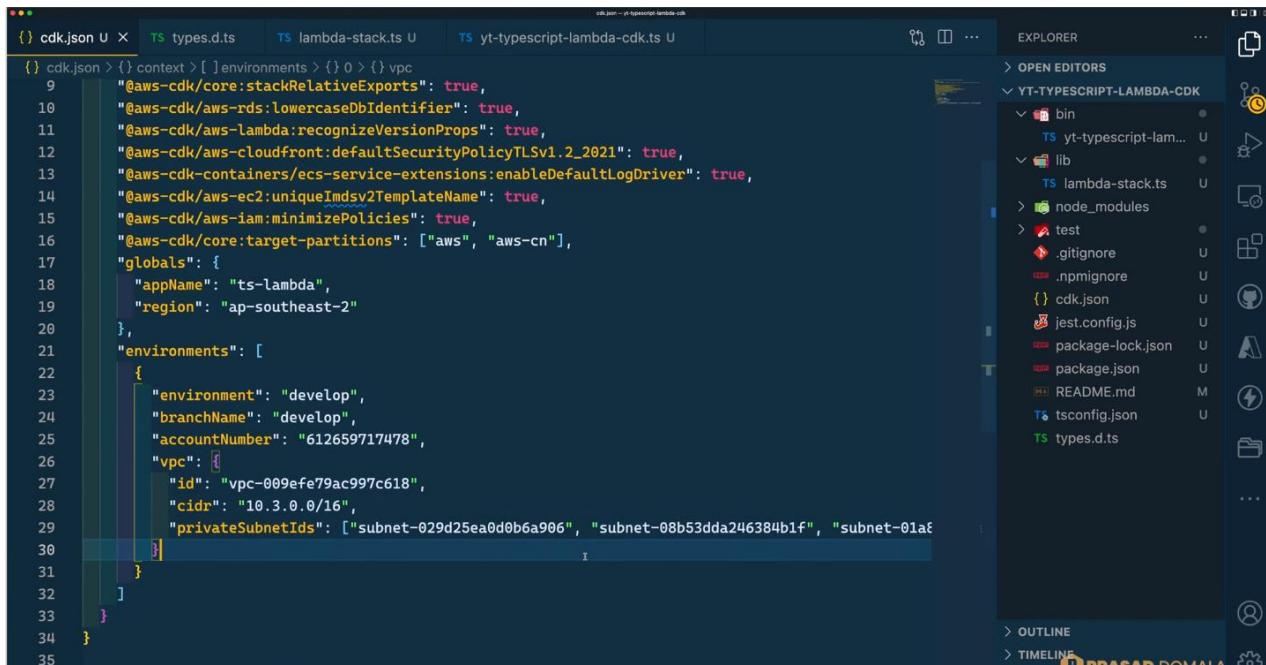
- Code Editor:** The main area displays the `lambda-stack.ts` file. The code defines an IAM role named `lambdaRole` with a policy named `'lambdaExecutionAccess'` that allows various log-related actions.
- Explorer Sidebar:** On the right, the **OPEN EDITORS** section shows the project structure:
 - `YT-TYPESCRIPT-LAMBDA-CDK` (selected)
 - `bin`
 - `lib`
 - `lambda-stack.ts` (selected)
 - `node_modules`
 - `test`
 - `.gitignore`
 - `.npmignore`
 - `cdk.json`
 - `jest.config.js`
 - `package-lock.json`
 - `package.json`
 - `README.md`
 - `tsconfig.json`
 - `types.d.ts`
- Bottom Status Bar:** Shows file paths like `lambda-stack.ts > ts lambda-stack.ts`, line numbers (e.g., L 27 Col 7), and other status indicators.

These completes adding our IAM resources from our architecture.

The screenshot shows the VS Code interface with the following details:

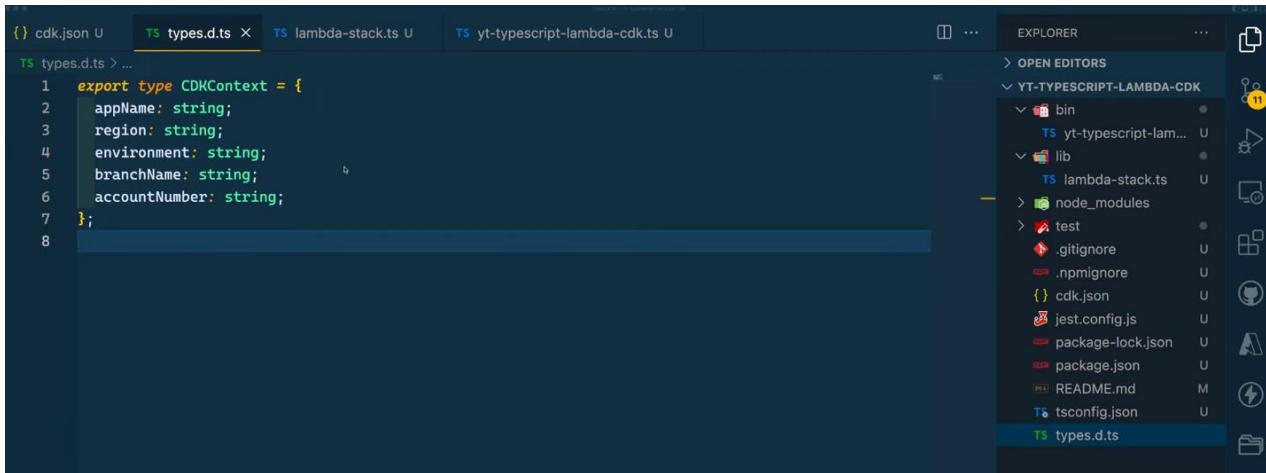
- Code Editor:** The main area displays the `cdk.json` file. It defines a context object with various AWS CDK configuration options, including `app`, `watch`, `include`, `exclude`, `context`, and `environments`.
- Explorer Sidebar:** On the right, the **OPEN EDITORS** section shows the project structure:
 - `YT-TYPESCRIPT-LAMBDA-CDK` (selected)
 - `bin`
 - `lib`
 - `lambda-stack.ts` (selected)
 - `node_modules`
 - `test`
 - `.gitignore`
 - `.npmignore`
 - `cdk.json`
 - `jest.config.js`
 - `package-lock.json`
 - `package.json`
 - `README.md`
 - `tsconfig.json`
 - `types.d.ts`
- Bottom Status Bar:** Shows file paths like `cdk.json > {} context > [] environments > {} 0`, line numbers (e.g., L 22 Col 8), and other status indicators.

Next, let us import our VPC and subnets for our private Lambda functions and create a security group



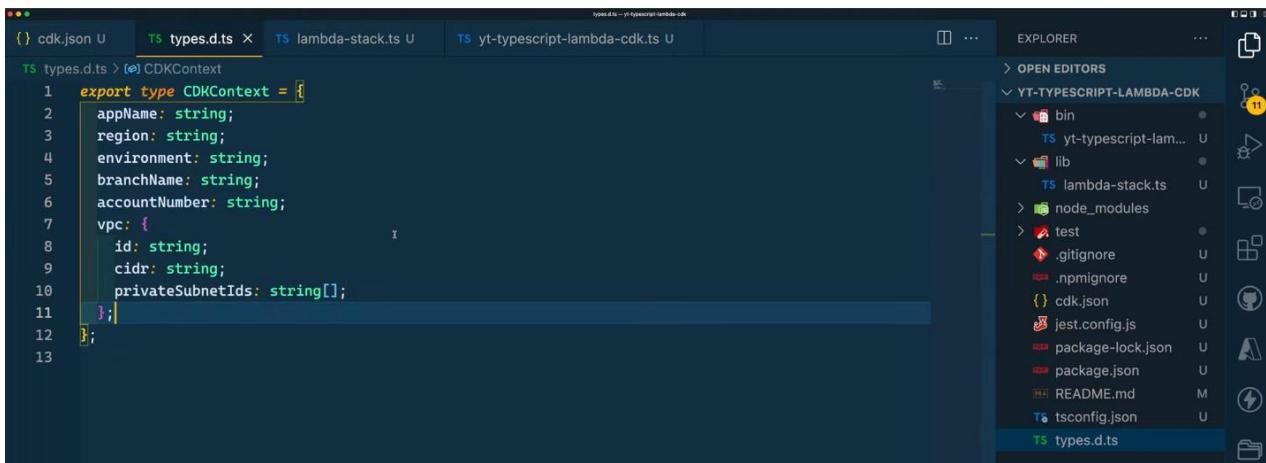
```
{}
cdk.json U TS types.d.ts U TS lambda-stack.ts U TS yt-typescript-lambda-cdk.ts U
{} cdk.json > {} context > [ ] environments > {} 0 > {} vpc
9   "@aws-cdk/core:stackRelativeExports": true,
10  "@aws-cdk/aws-rds:lowercaseDbIdentifier": true,
11  "@aws-cdk/aws-lambda:recognizeVersionProps": true,
12  "@aws-cdk/aws-cloudfront:defaultSecurityPolicyTLSv1.2_2021": true,
13  "@aws-cdk/aws-service-extensions:enableDefaultLogDriver": true,
14  "@aws-cdk/aws-ec2:uniqueImdsv2TemplateName": true,
15  "@aws-cdk/aws-iam:minimizePolicies": true,
16  "@aws-cdk/core:target-partitions": ["aws", "aws-cn"],
17  "globals": {
18    "appName": "ts-lambda",
19    "region": "ap-southeast-2"
20  },
21  "environments": [
22    {
23      "environment": "develop",
24      "branchName": "develop",
25      "accountNumber": "612659717478",
26      "vpc": [
27        {
28          "id": "vpc-009efe79ac97c618",
29          "cidr": "10.3.0.0/16",
30          "privateSubnetIds": ["subnet-029d25ea0d0b6a906", "subnet-08b53dda246384b1f", "subnet-01a8"]
31        }
32      ]
33    }
34  ]
35 }
```

We need to add our VPC and subnet details to the environments as above



```
{}
cdk.json U TS types.d.ts X TS lambda-stack.ts U TS yt-typescript-lambda-cdk.ts U
{} types.d.ts > ...
1 export type CDKContext = {
2   appName: string;
3   region: string;
4   environment: string;
5   branchName: string;
6   accountNumber: string;
7 };
8 
```

We also need to update our CDKContext definition to include our VPC details as below



```
{}
cdk.json U TS types.d.ts X TS lambda-stack.ts U TS yt-typescript-lambda-cdk.ts U
{} types.d.ts > [e] CDKContext
1 export type CDKContext = {
2   appName: string;
3   region: string;
4   environment: string;
5   branchName: string;
6   accountNumber: string;
7   vpc: {
8     id: string;
9     cidr: string;
10    privateSubnetIds: string[];
11  };
12};
13 
```

The screenshot shows the VS Code interface with the Explorer, Outline, and Timeline panes visible on the right. The code editor displays the `lambda-stack.ts` file, which contains the following code:

```
lib > TS lambda-stack.ts > ...
1 import { Stack, StackProps } from 'aws-cdk-lib';
2 import { Construct } from 'constructs';
3 import { CDKContext } from '../types';
4
5 export * as iam from 'aws-cdk-lib/aws-iam';
6 import * as ec2 from 'aws-cdk-lib/aws-ec2';
7
8 export class LambdaStack extends Stack {
9   constructor(scope: Construct, id: string, props: StackProps, context: CDKContext) {
10     super(scope, id, props);
11
12     // Lambda Role
13     const lambdaRole = new iam.Role(this, 'lambdaRole', {
14       roleName: `${context.appName}-lambda-role-${context.environment}`,
15       description: `Lambda role for ${context.appName}`,
16       assumedBy: new iam.ServicePrincipal('lambda.amazonaws.com'),
17       managedPolicies: [iam.ManagedPolicy.fromAwsManagedPolicyName('ReadOnlyAccess')],
18     });
19
20     // Attach inline policies to Lambda role
21     lambdaRole.attachInlinePolicy(
22       new iam.Policy(this, 'LambdaExecutionAccess', {
23         policyName: 'lambdaExecutionAccess',
24         statements: [
25           new iam.PolicyStatement({
26             effect: iam.Effect.ALLOW,
27             resources: ['*'],
28             actions: [
29               'logs>CreateLogGroup',
30               'logs>CreateLogStream',
31               'logs>DescribeLogGroups',
32               'logs>DescribeLogStreams',
33               'logs>PutLogEvents',
34             ],
35           }),
36         ],
37       });
38     );
39
40     // Import existing VPC based on VPC ID.
41     const vpc = ec2.Vpc.fromLookup(this, 'vpc', {
42       vpcId: context.vpc.id,
43     });
44   }
45 }
```

VPC concept is part of EC2 library so we need to import it as in line 6 above

The screenshot shows the VS Code interface with the Explorer, Outline, and Timeline panes visible on the right. The code editor displays the `lambda-stack.ts` file, which now includes the import statement from the EC2 library:

```
lib > TS lambda-stack.ts > ...
1 import { Stack, StackProps } from 'aws-cdk-lib';
2 import { Construct } from 'constructs';
3 import { CDKContext } from '../types';
4
5 import * as iam from 'aws-cdk-lib/aws-iam';
6 import * as ec2 from 'aws-cdk-lib/aws-ec2';
7
8 export class LambdaStack extends Stack {
9   constructor(scope: Construct, id: string, props: StackProps, context: CDKContext) {
10     super(scope, id, props);
11
12     // Lambda Role
13     const lambdaRole = new iam.Role(this, 'lambdaRole', {
14       roleName: `${context.appName}-lambda-role-${context.environment}`,
15       description: `Lambda role for ${context.appName}`,
16       assumedBy: new iam.ServicePrincipal('lambda.amazonaws.com'),
17       managedPolicies: [iam.ManagedPolicy.fromAwsManagedPolicyName('ReadOnlyAccess')],
18     });
19
20     // Attach inline policies to Lambda role
21     lambdaRole.attachInlinePolicy(
22       new iam.Policy(this, 'LambdaExecutionAccess', {
23         policyName: 'lambdaExecutionAccess',
24         statements: [
25           new iam.PolicyStatement({
26             effect: iam.Effect.ALLOW,
27             resources: ['*'],
28             actions: [
29               'logs>CreateLogGroup',
30               'logs>CreateLogStream',
31               'logs>DescribeLogGroups',
32               'logs>DescribeLogStreams',
33               'logs>PutLogEvents',
34             ],
35           }),
36         ],
37       });
38     );
39
40     // Import existing VPC based on VPC ID.
41     const vpc = ec2.Vpc.fromLookup(this, 'vpc', {
42       vpcId: context.vpc.id,
43     });
44   }
45 }
```

The screenshot shows the VS Code interface with the Explorer, Outline, and Timeline panes visible on the right. The code editor displays the `lambda-stack.ts` file, which includes additional imports and logic for private subnets:

```
lib > TS lambda-stack.ts > ...
1 import { Stack, StackProps } from 'aws-cdk-lib';
2 import { Construct } from 'constructs';
3 import { CDKContext } from '../types';
4
5 import * as iam from 'aws-cdk-lib/aws-iam';
6 import * as ec2 from 'aws-cdk-lib/aws-ec2';
7
8 export class LambdaStack extends Stack {
9   constructor(scope: Construct, id: string, props: StackProps, context: CDKContext) {
10     super(scope, id, props);
11
12     // Lambda Role
13     const lambdaRole = new iam.Role(this, 'lambdaRole', {
14       roleName: `${context.appName}-lambda-role-${context.environment}`,
15       description: `Lambda role for ${context.appName}`,
16       assumedBy: new iam.ServicePrincipal('lambda.amazonaws.com'),
17       managedPolicies: [iam.ManagedPolicy.fromAwsManagedPolicyName('ReadOnlyAccess')],
18     });
19
20     // Attach inline policies to Lambda role
21     lambdaRole.attachInlinePolicy(
22       new iam.Policy(this, 'LambdaExecutionAccess', {
23         policyName: 'lambdaExecutionAccess',
24         statements: [
25           new iam.PolicyStatement({
26             effect: iam.Effect.ALLOW,
27             resources: ['*'],
28             actions: [
29               'logs>CreateLogGroup',
30               'logs>CreateLogStream',
31               'logs>DescribeLogGroups',
32               'logs>DescribeLogStreams',
33               'logs>PutLogEvents',
34             ],
35           }),
36         ],
37       });
38     );
39
40     // Import existing VPC based on VPC ID.
41     const vpc = ec2.Vpc.fromLookup(this, 'vpc', {
42       vpcId: context.vpc.id,
43     });
44
45     // Import privateSubnets
46     const privateSubnets = context.vpc.privateSubnetIds.map((id, index) => ec2.Subnet.fromSubnetId(this, `privateSubnet${index}`));
47   }
48 }
```

```

39
40     existing VPC based on VPC ID.
41     = ec2.Vpc.fromLookup(this, 'vpc', {
42         context.vpc.id,
43     }
44
45     privateSubnets
46     const privateSubnets = context.vpc.privateSubnetIds.map((id, index) => ec2.Subnet.fromSubnetId(this, `privateSubnet${index}`, id));
47
48
49

```

```

1

```

Next, we start to create our Lambda layer

```

1 // Logger Functions
2 export const logInfo = (message: string | any, title: string | undefined = undefined): void => {
3     if (typeof message === 'string') {
4         title ? console.info(`${title}: ${message}`) : console.info(message);
5     } else {
6         title ? console.info(`${title}:`, JSON.stringify(message, null, 2)) : console.info(JSON.stringify(message));
7     }
8 };
9 export const logError = (message: string | any, title: string | undefined = undefined): void => {
10    if (typeof message === 'string') {
11        title ? console.error(`${title}: ${message}`) : console.error(message);
12    } else {
13        title ? console.error(`${title}:`, JSON.stringify(message, null, 2)) : console.error(JSON.stringify(message));
14    }
15 };
16 export const logWarn = (message: string | any, title: string | undefined = undefined): void => {
17    if (typeof message === 'string') {
18        title ? console.warn(`${title}: ${message}`) : console.warn(message);
19    } else {
20        title ? console.warn(`${title}:`, JSON.stringify(message, null, 2)) : console.warn(JSON.stringify(message));
21    }
22 };
23 export const logDebug = (message: string | any, title: string | undefined = undefined): void => {
24    if (process.env.LOG_LEVEL === 'debug') {
25        if (typeof message === 'string') {
26            title ? console.debug(`${title}: ${message}`) : console.debug(message);
27        } else {

```

We define several functions we need, then we will use them from within our Lambdas as below

```
lib > TS lambda-stack.ts ...  
1 import { Stack, StackProps } from 'aws-cdk-lib';  
2 import { Construct } from 'constructs';  
3 import { CDKContext } from '../types';  
4  
5 import * as iam from 'aws-cdk-lib/aws-iam';  
6 import * as ec2 from 'aws-cdk-lib/aws-ec2';  
7 import * as lambda from 'aws-cdk-lib/aws-lambda';  
8 import { NodejsFunction } from 'aws-cdk-lib/aws-lambda-nodejs';  
9  
10 export class LambdaStack extends Stack {  
11   constructor(scope: Construct, id: string, props: StackProps, context: CDKContext) {  
12     super(scope, id, props);  
13  
14     // Lambda Role  
15     const lambdaRole = new iam.Role(this, 'lambdaRole', {  
16       roleName: `${context.appName}-lambda-role-${context.environment}`,  
17       description: 'Lambda role for ${context.appName}',  
18       assumedBy: new iam.ServicePrincipal('lambda.amazonaws.com'),  
19       managedPolicies: [iam.ManagedPolicy.fromAwsManagedPolicyName('ReadOnlyAccess')],  
20     });  
21  
22     // Attach inline policies to Lambda role  
23     lambdaRole.attachInlinePolicy(  
24       new iam.Policy(this, 'lambdaExecutionAccess', {  
25         policyName: 'lambdaExecutionAccess',  
26         statements: [  
27           new iam.PolicyStatement({  
28             effect: iam.Effect.Allow, ...  
29           })  
30         ]  
31       })  
32     );  
33  
34     // Import existing VPC based on VPC ID.  
35     const vpc = ec2.Vpc.fromLookup(this, 'vpc', {  
36       vpcId: context.vpc.id,  
37     });  
38  
39     // Import privateSubnets  
40     const privateSubnets = context.vpc.privateSubnetIds.map((id, index) => ec2.Subnet.fromSubnetId(  
41       id,  
42       index  
43     ));  
44  
45     // Lambda Security Group  
46     const lambdaSG = new ec2.SecurityGroup(this, 'lambdaSG', {  
47       vpc,  
48       allowAllOutbound: true,  
49       securityGroupName: `${context.appName}-lambda-security-group-${context.environment}`,  
50     });  
51     lambdaSG.addIngressRule(ec2.Peer.ipv4(context.vpc.cidr), ec2.Port.allTcp(), 'Allow internal VPC  
52     traffic');  
53  
54     // Lambda Layer  
55     const lambdaLayer = new lambda.LayerVersion(this, 'lambdaLayer', {  
56       code: lambda.Code.fromAsset('lambda-layer'),  
57       compatibleRuntimes: [lambda.Runtime.NODEJS_14_X],  
58       description: 'Lambda Layer for ${context.appName}',  
59     });  
60  
61   }  
62 }  
63  
64  
65 }  
66
```

```
lib > TS lambda-stack.ts ...  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66
```

These are the resources required for our Lambda functions based on our architecture

```
lib > TS lambda-config.ts ...  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66
```

Next, let us define the actual Lambda functions themselves. We will define our Lambdas as a group using a config that the stack file can read to generate the specific Lambda function definitions dynamically. The definition will hold all the props required to define our Lambda functions.

```

{} cdk.json U   TS types.d.ts X   TS lambda-stack.ts U   TS lambda-config.ts U   TS utils.ts
TS yt-typescript-lambda-cdk.ts U

TS types.d.ts > e] LambdaDefinition
1  export type CDKContext = {
2    appName: string;
3    region: string;
4    environment: string;
5    branchName: string;
6    accountNumber: string;
7    vpc: {
8      id: string;
9      cidr: string;
10     privateSubnetIds: string[];
11   };
12 };
13
14 export type LambdaDefinition = {
15   name: string;
16   memoryMB?: number;
17   timeoutMins?: number;
18   environment?: {
19     [key: string]: string;
20   };
21   isPrivate?: boolean;
22 };

```

```

{} cdk.json U   TS types.d.ts   TS lambda-stack.ts U   TS lambda-config.ts U   TS utils.ts
TS yt-typescript-lambda-cdk.ts U

lib > TS lambda-config.ts > ...
1 import { LambdaDefinition, CDKContext } from '../types';
2
3 // Constants
4 const DEFAULT_LAMBDA_MEMORY_MB = 1024;
5 const DEFAULT_LAMBDA_TIMEOUT_MINS = 15;

```

```

{} cdk.json U   TS types.d.ts   TS lambda-stack.ts U   TS lambda-config.ts U   TS utils.ts
TS yt-typescript-lambda-cdk.ts U

lib > TS lambda-config.ts > ...
7 // Returns lambda definitions with custom env
8 export const getLambdaDefinitions = (context: CDKContext): LambdaDefinition[] => {
9   const lambdaDefinitions: LambdaDefinition[] = [
10     {
11       name: 'public-function',
12       environment: {
13         REGION: context.region,
14         ENV: context.environment,
15         GIT_BRANCH: context.branchName,
16       },
17       isPrivate: false,
18     },
19     {
20       name: 'private-function',
21       memoryMB: 2048,
22       timeoutMins: 5,
23       environment: {
24         REGION: context.region,
25         ENV: context.environment,
26         GIT_BRANCH: context.branchName,
27       },
28       isPrivate: true,
29     },
30   ];
31   return lambdaDefinitions;
32 };

```

```
lambda-config.ts -- yt-typescript-lambda-cdk
```

```
{} cdk.json U TS types.d.ts TS lambda-stack.ts U TS lambda-config.ts U TS utils.ts
```

```
TS yt-typescript-lambda-cdk.ts U
```

```
lib > TS lambda-config.ts > ...
```

```
1 import { LambdaDefinition, CDKContext } from '../types';
2 import { NodejsFunctionProps } from 'aws-cdk-lib/aws-lambda-nodejs';
3 export * as lambda from 'aws-cdk-lib/aws-lambda';
4 import * as iam from 'aws-cdk-lib/aws-iam';
```

```
5
6 // Constants
7 const DEFAULT_LAMBDA_MEMORY_MB = 1024;
8 const DEFAULT_LAMBDA_TIMEOUT_MIN = 15;
```

```
9
10 // Returns lambda definitions with custom env
11 export const getLambdaDefinitions = (context: CDKContext): LambdaDefinition[] => {
12   const lambdaDefinitions: LambdaDefinition[] = [
13     {
14       name: 'public-function',
15       environment: {
16         REGION: context.region,
17         ENV: context.environment,
18         GIT_BRANCH: context.branchName,
19       },
20       isPrivate: false,
21     },
22     {
23       name: 'private-function',
24       memoryMB: 2048,
25       timeoutMins: 5,
26       environment: {
27         memoryMB: 2048,
28         timeoutMins: 5,
29         environment: {
30           REGION: context.region,
31           ENV: context.environment,
32           GIT_BRANCH: context.branchName,
33         },
34         isPrivate: true,
35       },
36     },
37   ];
38   return lambdaDefinitions;
39 }
```

```
ln 4 Col 44 Spaces: 2 UTF-8 LF (1) TypeScript Go live Spell Prettier
```

```
EXPLORER OPEN EDITORS ...
```

```
YT-TYPESCRIPT-LAMBDA-CDK bin
  TS yt-typescript-lam... U
  lambda-layer
    TS utils.ts
  lib
    TS lambda-config.ts U
    TS lambda-stack.ts U
node_modules
test
  .gitignore U
  .npmignore U
  cdk.json U
  jest.config.js U
  package-lock.json U
  package.json U
  README.md M
  tsconfig.json U
  types.d.ts
```

```
OUTLINE TIMELINE NPM SCRIPTS
```

```
PRASAD DOMALA
```

```
lambda-config.ts -- yt-typescript-lambda-cdk
```

```
{} cdk.json U TS types.d.ts TS lambda-stack.ts U TS lambda-config.ts U TS utils.ts
```

```
TS yt-typescript-lambda-cdk.ts U
```

```
lib > TS lambda-config.ts > ...
```

```
25   memoryMB: 2048,
26   timeoutMins: 5,
27   environment: {
28     REGION: context.region,
29     ENV: context.environment,
30     GIT_BRANCH: context.branchName,
31   },
32   isPrivate: true,
33 },
34 ];
35 return lambdaDefinitions;
36 };
37
38
```

```
ln 4 Col 44 Spaces: 2 UTF-8 LF (1) TypeScript Go live Spell Prettier
```

```
EXPLORER OPEN EDITORS ...
```

```
YT-TYPESCRIPT-LAMBDA-CDK bin
  TS yt-typescript-lam... U
  lambda-layer
    TS utils.ts
  lib
    TS lambda-config.ts U
    TS lambda-stack.ts U
node_modules
test
  .gitignore U
  .npmignore U
  cdk.json U
  jest.config.js U
  package-lock.json U
  package.json U
  README.md M
  tsconfig.json U
  types.d.ts
```

```
OUTLINE TIMELINE NPM SCRIPTS
```

```
PRASAD DOMALA
```

```
lambda-config.ts -- yt-typescript-lambda-cdk
```

```
{} cdk.json U TS types.d.ts TS lambda-stack.ts U TS lambda-config.ts U TS utils.ts
```

```
TS yt-typescript-lambda-cdk.ts U
```

```
lib > TS lambda-config.ts > ...
```

```
37
38 // Returns Lambda Function properties with defaults and overwrites
39 export const getFunctionProps = (
40   lambdaDefinition: LambdaDefinition,
41   lambdaRole: iam.Role,
42   lambdaLayer: lambda.LayerVersion,
43   context: CDKContext
44 ): NodejsFunctionProps => {
45   const functionProps: NodejsFunctionProps = {
46     functionName: `${context.appName}-${lambdaDefinition.name}-${context.environment}`,
47     entry: `lambda-handlers/${lambdaDefinition.name}.ts`,
48     runtime: lambda.Runtime.NODEJS_14_X,
49     memorySize: lambdaDefinition.memoryMB ? lambdaDefinition.memoryMB : DEFAULT_LAMBDA_MEMORY_MB,
50     timeout: lambdaDefinition.timeoutMins ? Duration.minutes(lambdaDefinition.timeoutMins) : Duration.seconds(10),
51     environment: lambdaDefinition.environment,
52     role: lambdaRole,
53     layers: [lambdaLayer],
54   };
55   return functionProps;
56 }
```

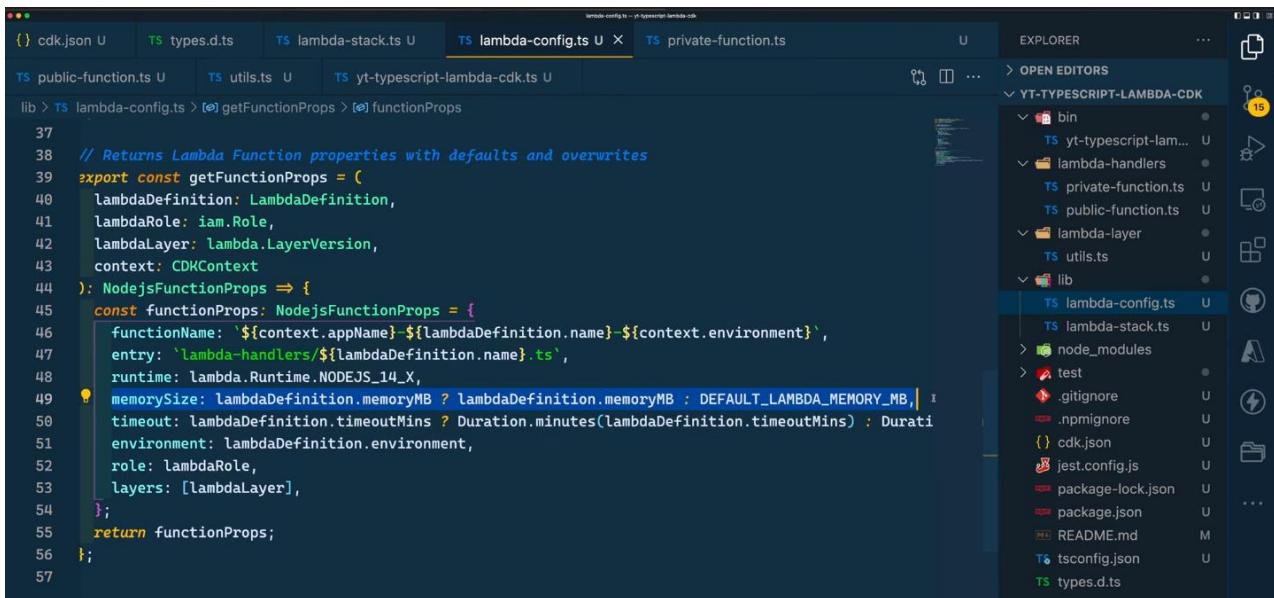
```
ln 4 Col 44 Spaces: 2 UTF-8 LF (1) TypeScript Go live Spell Prettier
```

```
EXPLORER OPEN EDITORS ...
```

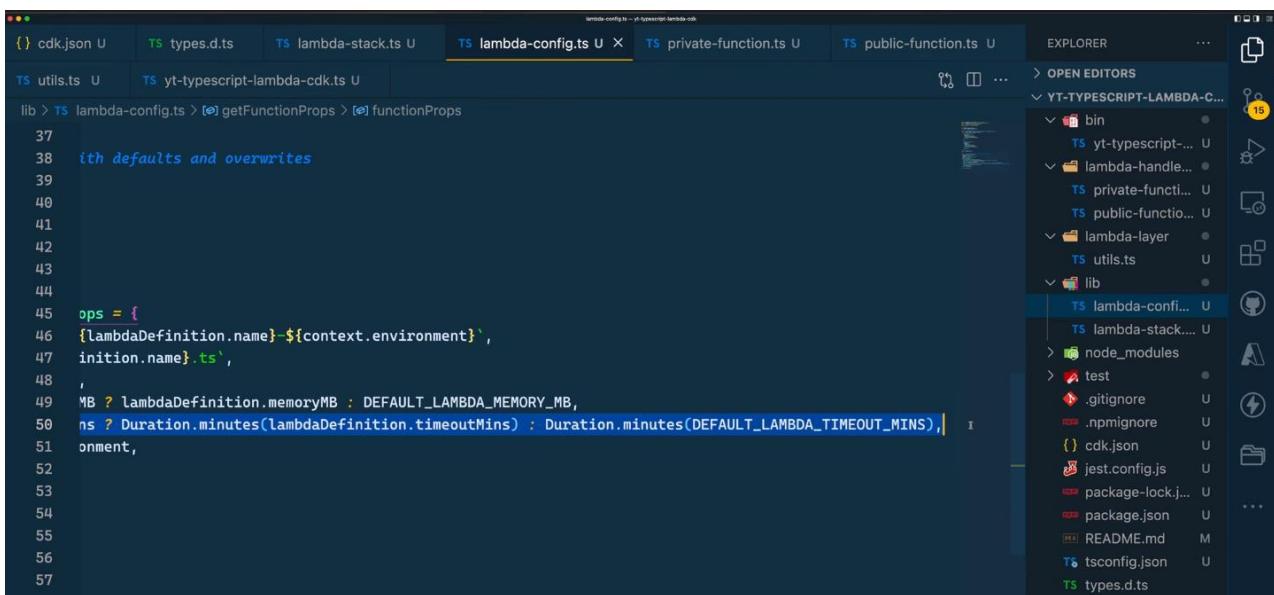
```
YT-TYPESCRIPT-LAMBDA-CDK bin
  TS yt-typescript-lam... U
  lambda-layer
    TS utils.ts
  lib
    TS lambda-config.ts U
    TS lambda-stack.ts U
node_modules
test
  .gitignore U
  .npmignore U
  cdk.json U
  jest.config.js U
  package-lock.json U
  package.json U
  README.md M
  tsconfig.json U
  types.d.ts
```

```
OUTLINE TIMELINE NPM SCRIPTS
```

```
PRASAD DOMALA
```



```
lib > TS lambda-config.ts > [e] getFunctionProps > [e] functionProps
37
38 // Returns Lambda Function properties with defaults and overwrites
39 export const getFunctionProps = (
40   lambdaDefinition: LambdaDefinition,
41   lambdaRole: iam.Role,
42   lambdaLayer: lambda.LayerVersion,
43   context: CDKContext
44 ): NodejsFunctionProps => {
45   const functionProps: NodejsFunctionProps = {
46     functionName: `${context.appName}-${lambdaDefinition.name}-${context.environment}`,
47     entry: `lambda-handlers/${lambdaDefinition.name}.ts`,
48     runtime: lambda.Runtime.NODEJS_14_X,
49     memorySize: lambdaDefinition.memoryMB ? lambdaDefinition.memoryMB : DEFAULT_LAMBDA_MEMORY_MB,
50     timeout: lambdaDefinition.timeoutMins ? Duration.minutes(lambdaDefinition.timeoutMins) : Duration.seconds(DEFAULT_LAMBDA_TIMEOUT_SECS),
51     environment: lambdaDefinition.environment,
52     role: lambdaRole,
53     layers: [lambdaLayer],
54   };
55   return functionProps;
56 };
57
```



```
lib > TS lambda-config.ts > [e] getFunctionProps > [e] functionProps
37
38 with defaults and overwrites
39
40
41
42
43
44
45   ops = [
46     {LambdaDefinition.name}-${context.environment},
47     inition.name}.ts`,
48   ,
49   MB ? lambdaDefinition.memoryMB : DEFAULT_LAMBDA_MEMORY_MB,
50   ns ? Duration.minutes(lambdaDefinition.timeoutMins) : Duration.minutes(DEFAULT_LAMBDA_TIMEOUT_MINS),
51   onment,
```

We can now use the functions in our stack file to define our dynamic Lambda functions

The screenshot shows the VS Code interface with the 'lambda-stack.ts' file selected in the tab bar. The code implements the constructor for the LambdaStack class, which extends the Stack class from aws-cdk-lib. It creates a LambdaRole using iam.Role and attaches an inline policy for lambdaExecutionAccess.

```
import { Stack, StackProps } from 'aws-cdk-lib';
import { Construct } from 'constructs';
import { CDKContext } from './types';

import * as iam from 'aws-cdk-lib/aws-iam';
import * as ec2 from 'aws-cdk-lib/aws-ec2';
import * as lambda from 'aws-cdk-lib/aws-lambda';
import { NodejsFunction } from 'aws-cdk-lib/aws-lambda-nodejs';

import { getLambdaDefinitions, getFunctionProps } from './lambda-config';

export class LambdaStack extends Stack {
  constructor(scope: Construct, id: string, props: StackProps, context: CDKContext) {
    super(scope, id, props);

    // Lambda Role
    const lambdaRole = new iam.Role(this, 'LambdaRole', {
      roleName: `${context.appName}-lambda-role-${context.environment}`,
      description: 'Lambda role for ${context.appName}',
      assumedBy: new iam.ServicePrincipal('lambda.amazonaws.com'),
      managedPolicies: [iam.ManagedPolicy.fromAwsManagedPolicyName('ReadOnlyAccess')],
    });

    // Attach inline policies to Lambda role
    lambdaRole.attachInlinePolicy(
      new iam.Policy(this, 'lambdaExecutionAccess', [
        ...
      ])
    );
  }
}
```

The screenshot shows the same VS Code interface with the 'lambda-stack.ts' file selected. The code now includes logic to add an ingress rule to the Lambda function's VPC configuration and to loop through lambda definitions to create lambda functions.

```
lambdaSG.addIngressRule(ec2.Peer.ipv4(context.vpc.cidr), ec2.Port.allTcp(), 'Allow internal VPC traffic');

// Lambda Layer
const lambdaLayer = new lambda.LayerVersion(this, 'lambdaLayer', {
  code: lambda.Code.fromAsset('lambda-layer'),
  compatibleRuntimes: [lambda.Runtime.NODEJS_14_X],
  description: `Lambda Layer for ${context.appName}`,
});

// Get Lambda definitions
const lambdaDefinitions = getLambdaDefinitions(context);

// Loop through the definitions and create lambda functions
for (const lambdaDefinition of lambdaDefinitions) {
  // Get function props based on lambda definition
  let functionProps = getFunctionProps(lambdaDefinition, lambdaRole, lambdaLayer, context);
}

}
```

The screenshot shows the VS Code interface with the Explorer sidebar open. The current file is `lambda-stack.ts`. The code implements a constructor for `LambdaStack` that iterates through `lambdaDefinitions` to create `lambda functions`. It checks if each function is private and adds VPC, SG, and Subnets accordingly. The `functionProps` object is updated with `vpc`, `securityGroups`, and `vpcSubnets`.

```
// Get Lambda definitions
const lambdaDefinitions = getLambdaDefinitions(context);

// Loop through the definitions and create lambda functions
for (const lambdaDefinition of lambdaDefinitions) {
    // Get function props based on lambda definition
    let functionProps = getFunctionProps(lambdaDefinition, lambdaRole, lambdaLayer, context);

    // Check if function is private and add VPC, SG and Subnets
    if (lambdaDefinition.isPrivate) {
        functionProps = {
            ...functionProps,
            vpc: vpc,
            securityGroups: [lambdaSG],
            vpcSubnets: {
                subnets: privateSubnets,
            },
        };
    }
}
```

The screenshot shows the VS Code interface with the Explorer sidebar open. The current file is `lambda-stack.ts`. The code creates a new Lambda function using `new NodejsFunction`, passing the function name and `functionProps`.

```
// Check if function is private and add VPC, SG and subnets
if (lambdaDefinition.isPrivate) {
    functionProps = {
        ...functionProps,
        vpc: vpc,
        securityGroups: [lambdaSG],
        vpcSubnets: {
            subnets: privateSubnets,
        },
    };
}

// Lambda Function
new NodejsFunction(this, `${lambdaDefinition.name}-function`, functionProps);
```

Our stack file is clean and concise. All we need to do is to modify our Lambda definitions array in our config file to add new Lambda functions.

```
lib > TS lambda-stack.ts > ...
1 import { Stack, StackProps } from 'aws-cdk-lib';
2 import { Construct } from 'constructs';
3 import { CDKContext } from './types';
4
5 import * as iam from 'aws-cdk-lib/aws-iam';
6 import * as ec2 from 'aws-cdk-lib/aws-ec2';
7 import * as lambda from 'aws-cdk-lib/aws-lambda';
8 import { NodejsFunction } from 'aws-cdk-lib/aws-lambda-nodejs';
9 import * as cwLogs from 'aws-cdk-lib/aws-logs';
10
11 import { getLambdaDefinitions, getFunctionProps } from './lambda-config';
12
13 export class LambdaStack extends Stack {
14   constructor(scope: Construct, id: string, props: StackProps, context: CDKContext) {
15     super(scope, id, props);
16
17     // Lambda Role
18     const lambdaRole = new iam.Role(this, 'lambdaRole', {
19       roleName: `${context.appName}-lambda-role-${context.environment}`,
20       description: `Lambda role for ${context.appName}`,
21       assumedBy: new iam.ServicePrincipal('lambda.amazonaws.com'),
22       managedPolicies: [iam.ManagedPolicy.fromAwsManagedPolicyName('ReadOnlyAccess')],
23     });
24
25     // Attach inline policies to Lambda role
26     lambdaRole.attachInlinePolicy([

```

```
for (const lambdaDefinition of lambdaDefinitions) {
  // Get function props based on lambda definition
  let functionProps = getFunctionProps(lambdaDefinition, lambdaRole, lambdaLayer, context);

  // Check if function is private and add VPC, SG and Subnets
  if (lambdaDefinition.isPrivate) {
    functionProps = {
      ...functionProps,
      vpc: vpc,
      securityGroups: [lambdaSG],
      vpcSubnets: {
        subnets: privateSubnets,
      },
    };
  }

  // Lambda Function
  new NodejsFunction(this, `${lambdaDefinition.name}-function`, functionProps);

  // Create corresponding Log Group with one month retention
  new cwLogs.LogGroup(this, `fn-${lambdaDefinition.name}-log-group`, {
    logGroupName: `/aws/lambda/${context.appName}-${lambdaDefinition.name}-${context.environment}`,
    retention: cwLogs.RetentionDays.ONE_MONTH,
    removalPolicy: RemovalPolicy.DESTROY,
  });
}
```

This completes our stack file.

```

{} cdk.json U TS types.d.ts TS lambda-stack.ts U TS lambda-config.ts U TS private-function.ts U TS public-fun ... PROBLEMS 2 SEARCH OUTPUT DEBUG CONSOLE TERMINAL GITLENS
lib > TS lambda-stack.ts > LambdaStack > constructor
  vpcSubnets: f
  npm i aws-lambda
  npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. new code should use the URLSearchParams API instead.
  npm WARN deprecated uuid@3.3.2: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.

  added 17 packages, and audited 582 packages in 3s

  28 packages are looking for funding
    run `npm fund` for details

  found 0 vulnerabilities

  npm i -D @types/aws-lambda
  added 1 package, and audited 583 packages in 3s

  28 packages are looking for funding
    run `npm fund` for details

  found 0 vulnerabilities

```

Let's add some sample code to our Lambda functions, we need the above library that provides us all the TS types

```

{} cdk.json U TS types.d.ts TS lambda-stack.ts U TS lambda-config.ts U TS private-function.ts 2, U ...
TS public-function.ts U TS utils.ts U TS yt-typescript-lambda-cdk.ts U
lambda-handlers > TS private-function.ts
1 import { Handler, Handler } from 'aws-lambda';
  ↗ CodePipelineCloudWatchActionHandler
  ↗ APIGatewayRequestIAMAuthorizerHandlerV2
  ↗ CodePipelineCloudWatchPipelineHandler
  ↗ APIGatewayAuthorizerWithContextHandler
  ↗ APIGatewayRequestSimpleAuthorizerHandlerV2
  ↗ APIGatewayRequestSimpleAuthorizerHandlerV2With...
  ↗ VerifyAuthChallengeResponseTriggerHandler
  ↗ APIGatewayProxyWithLambdaAuthorizerHandler
  ↗ APIGatewayProxyWithCognitoAuthorizerHandler
  ↗ APIGatewayTokenAuthorizerWithContextHandler
  ↗ APIGatewayRequestAuthorizerWithContextHandler
  ↗ APIGatewayRequestIAMAuthorizerV2WithContextHan...
[AI] import Handler (Auto-Import)
[AI] Import Handler from aws-lambda

```

```

{} cdk.json U TS types.d.ts TS lambda-stack.ts U TS lambda-config.ts U TS private-function.ts U TS public-function.ts U ...
TS utils.ts U TS yt-typescript-lambda-cdk.ts U
lambda-handlers > TS private-function.ts > ...
1 import { Handler } from 'aws-lambda';
2
3 export const handler: Handler = async (event, context) => {
4   return new Promise<string>(async (resolve, reject) => {
5     try {
6       // Lambda handler code goes here
7       return resolve('This is a Private Function');
8     } catch (error) {
9       reject();
10    }
11  });
12};
13

```

We can utilize the logging functions we defined in our Lambda layers here. Our lambda layers will be mapped to the `/opt` directory in our Lambda runtime.

private-function.ts

```
1  export { Handler } from 'aws-lambda';
2  import * as utils from '/opt/utils';
3
4  export const handler: Handler = async (event, context) => {
5    return new Promise<string>(async (resolve, reject) => {
6      try {
7        // Lambda handler code goes here
8        return resolve('This is a Private Function');
9      } catch (error) {
10        reject();
11      }
12    });
13  };
14
```

EXPLORER

- YT-TYPESCRIPT-LAMBDA-C...
- bin
- lambda-handle...
- private-fun... 1, U
- public-functio...
- lambda-layer
- utils.ts
- lib
- lambda-confi...
- lambda-stack.... U
- node_modules
- test
- .gitignore
- .npmignore
- cdk.json

private-function.ts

```
1  export { Handler } from 'aws-lambda';
2  import * as utils from '/opt/utils';
3
4  'utils' is declared but its value is never read. ts(6133)
5  Cannot find module '/opt/utils' or its corresponding type
6  declarations. ts(2307)
7  View Problem Quick Fix... (%)
8
9  return resolve('This is a Private Function');
10 } catch (error) {
11   reject();
12 }
13 };
14
```

EXPLORER

- YT-TYPESCRIPT-LAMBDA-C...
- bin
- lambda-handle...
- private-fun... 1, U
- public-functio...
- lambda-layer
- utils.ts
- lib
- lambda-confi...
- lambda-stack.... U
- node_modules
- test
- .gitignore
- .npmignore
- cdk.json

tsconfig.json

```
1  "compilerOptions": {
2    "declaration": true,
3    "strict": true,
4    "noImplicitAny": true,
5    "strictNullChecks": true,
6    "noImplicitThis": true,
7    "alwaysStrict": true,
8    "noUnusedLocals": false,
9    "noUnusedParameters": false,
10   "noImplicitReturns": true,
11   "noFallthroughCasesInSwitch": false,
12   "inlineSourceMap": true,
13   "inlineSources": true,
14   "experimentalDecorators": true,
15   "strictPropertyInitialization": false,
16   "typeRoots": ["./node_modules/@types"],
17   "esModuleInterop": true,
18   "baseUrl": ".",
19   "paths": [
20     "/opt/*": ["lambda-layer/*"]
21   ],
22   "exclude": ["node_modules", "cdk.out"]
23 }
24
25
26
27
28 }
```

EXPLORER

- YT-TYPESCRIPT-LAMBDA-C...
- bin
- lambda-handle...
- private-fun... 1, U
- public-functio...
- lambda-layer
- utils.ts
- lib
- lambda-confi...
- lambda-stack.... U
- node_modules
- test
- .gitignore
- .npmignore
- cdk.json
- jest.config.js
- package-lock.j...
- package.json
- README.md
- tsconfig.json
- types.d.ts

```

private-function.ts - yt-typescript-lambda-cdk
1 import { Handler } from 'aws-lambda';
2 import * as utils from '/opt/utils';
3
4 export const handler: Handler = async (event, context) => {
5   return new Promise<string>(async (resolve, reject) => {
6     try {
7       // Lambda handler code goes here
8       return resolve('This is a Private Function');
9     } catch (error) {
10      reject();
11    }
12  });
13};
14

```

```

private-function.ts - yt-typescript-lambda-cdk
1 import { Handler } from 'aws-lambda';
2 import * as utils from '/opt/utils';
3
4 export const handler: Handler = async (event, context) => {
5   return new Promise<string>(async (resolve, reject) => {
6     try {
7       // Lambda handler code goes here
8       utils.logInfo('This is a Private Function');
9       return resolve('This is a Private Function');
10    } catch (error) {
11      utils.LogError(error);
12      reject();
13    }
14  });
15};
16

```

```

public-function.ts - yt-typescript-lambda-cdk
1 import { Handler } from 'aws-lambda';
2 import * as utils from '/opt/utils';
3
4 export const handler: Handler = async (event, context) => {
5   return new Promise<string>(async (resolve, reject) => {
6     try {
7       // Lambda handler code goes here
8       utils.logInfo('This is a Public Function');
9       return resolve('This is a Public Function');
10    } catch (error) {
11      utils.LogError(error);
12      reject();
13    }
14  });
15};
16

```

Next, let's see how we can execute these private and public Lambdas locally for testing purposes.

```

Profile 'kheriox-demo-dev' switched successfully
AWS export command is copied to your clipboard. Please paste (Cmd-V) the command to set your profile

AWS_PROFILE=kheriox-demo-dev

```

```

export AWS_PROFILE=kheriox-demo-dev

```

```

pressd@pressd-MBP-045-typescript-lambda-cdk:~/Documents/Music/045-typescript-lambda-cdk> cdk develop !1 ?13  kheriox-demo-dev ap-southeast-2 8 00:23:48 ⏎
> cdk list
ts-lambda-stack-develop

apple ~ /Doc/Y/045-typescript-lambda-cdk/git-repo/yt-typescript-lambda-cdk> cdk develop !1 ?13  4s ✘ kheriox-demo-dev ap-southeast-2 8 00:24:01 ⏎
> cdk synth ts-lambda-stack-develop --no-staging > template.yaml
Bundling asset ts-lambda-stack-develop/public-function-function/Code/Stage...
cdk.out/bundling-temp-74d7a518c7ee626d829112ced1387b81d1c54c0da335c4de8c86acc6a6ec7ee3/index.js 2.3kb
⚡ Done in 6ms
Bundling asset ts-lambda-stack-develop/private-function-function/Code/Stage...
cdk.out/bundling-temp-577610e0cea7fcf1df7a94cdab8e92af02ca4c1ddf2d5b8023602649725c3e76/index.js 2.3kb
⚡ Done in 1ms
[Warning at /ts-lambda-stack-develop/privateSubnet0] No routeTableId was provided to the subnet 'subnet-029d25ea0d0b6a906'. Attempting to read its .routeTableId will return null/undefined. (More info: https://github.com/aws/aws-cdk/pull/3171)
[Warning at /ts-lambda-stack-develop/privateSubnet1] No routeTableId was provided to the subnet 'subnet-08b53dda246384b1f'. Attempting to read its .routeTableId will return null/undefined. (More info: https://github.com/aws/aws-cdk/pull/3171)
[Warning at /ts-lambda-stack-develop/privateSubnet2] No routeTableId was provided to the subnet 'subnet-01a855a9fe96b9ded'. Attempting to read its .routeTableId will return null/undefined. (More info: https://github.com/aws/aws-cdk/pull/3171)

apple ~ /Doc/Y/045-typescript-lambda-cdk/git-repo/yt-typescript-lambda-cdk> cdk develop !1 ?14  4s ✘ kheriox-demo-dev ap-southeast-2 8 00:24:34 ⏎
>

```

```

{
  "Description": "This is the Stack Description",
  "Resources": {
    "lambdaRoleC844FDB1": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [
            {
              "Action": "sts:AssumeRole",
              "Effect": "Allow",
              "Principal": "Service:lambda.amazonaws.com"
            }
          ],
          "Version": "2012-10-17"
        },
        "Description": "Lambda role for ts-lambda",
        "ManagedPolicyArns": [
          {
            "Fn::Join": [
              "",
              [
                "-arn:",
                {
                  "Ref": "AWS::Partition"
                },
                ":iam::aws:policy/ReadOnlyAccess"
              ]
            ],
            "RoleName": "ts-lambda-lambda-role-develop"
          }
        ],
        "Metadata": {
          "aws:cdk:path": "ts-lambda-stack-develop/lambdaRole/Resource"
        },
        "lambdaExecutionAccess82ECF6E2": {
          "Type": "AWS::IAM::Policy",
          "Properties": {
            "PolicyDocument": ...
          }
        }
      }
    }
  }
}

```

```
template.yaml
20   RoleName: ts-lambda-lambda-role-develop
21   Metadata:
22     aws:cdk:path: ts-lambda-stack-develop/lambdaRole/Resource
23   lambdaExecutionAccess82ECF6E2:
24     Type: AWS::IAM::Policy
25     Properties:
26       PolicyDocument:
27         Statement:
28           - Action:
29             - logs>CreateLogGroup
29             - logs>CreateLogStream
29             - logs>DescribeLogGroups
29             - logs>DescribeLogStreams
29             - logs>PutLogEvents
29           Effect: Allow
29           Resource: "*"
29         Version: "2012-10-17"
30       PolicyName: lambdaExecutionAccess
31     Roles:
32       - Ref: lambdaRoleC844FDB1
33     Metadata:
34       aws:cdk:path: ts-lambda-stack-develop/lambdaExecutionAccess/Resource
35   lambdaSG81E771DE:
36     Type: AWS::EC2::SecurityGroup
37     Properties:
38       GroupDescription: ts-lambda-stack-develop/lambdaSG
```

The screenshot shows the VS Code interface with the following details:

- File Explorer (Left):** Shows the project structure with files like `lambda-handle...`, `private-functi...`, `public-functio...`, `lambda-layer...`, `utils.ts`, `lib`, `lambda-confi...`, `lambda-stack....`, `node_modules`, `test`, `.gitignore`, `.npmignore`, `cdk.context.json`, `cdk.json`, `jest.config.js`, `package-lock.j...`, `package.json`, `README.md`, `template.yaml` (highlighted in blue), `tsconfig.json`, and `types.d.ts`.
- Editor (Center):** Displays the `template.yaml` file content in YAML format, defining a security group for Lambda execution.
- Bottom Status Bar:** Shows the current file is `template.yaml`, the line and column are `Line 43, Col 5 / 29 selected`, and there are `Spaces: 2`, `UTF-8`, `LF`, `YAML` encoding options.

```

49     Description: Allow all outbound traffic by default
50     IpProtocol: "-1"
51   SecurityGroupIngress:
52     - CidrIp: 10.3.0.0/16
53       Description: Allow internal VPC traffic
54       FromPort: 0
55       IpProtocol: tcp
56       ToPort: 65535
57     VpcId: vpc-009e79ac997c618
58   Metadata:
59     aws:cdk:path: ts-lambda-stack-develop/lambdaSG/Resource
60   lambdaLayer857D4AF4:
61     Type: AWS::Lambda::LayerVersion
62     Properties:
63       Content:
64         S3Bucket: cdk-hnb659fds-assets-612659717478-ap-southeast-2
65         S3Key: 5947d7a0930311fd47fce02ef60bb06e6dbd1f3dc3c89b1903215a7a66115127.zip
66       CompatibleRuntimes:
67         - nodejs14.x
68       Description: Lambda Layer for ts-lambda
69     Metadata:
70       aws:cdk:path: ts-lambda-stack-develop/lambdaLayer/Resource
71       aws:asset:path: /Users/prasaddomala/Documents/Youtube/045-typescript-lambda-cdk/git-repo/yt-type
72       aws:asset:is-bundled: false
73       aws:asset:property: Content

```

```

58   Metadata:
59     aws:cdk:path: ts-lambda-stack-develop/lambdaSG/Resource
60   lambdaLayer857D4AF4:
61     Type: AWS::Lambda::LayerVersion
62     Properties:
63       Content:
64         S3Bucket: cdk-hnb659fds-assets-612659717478-ap-southeast-2
65         S3Key: 5947d7a0930311fd47fce02ef60bb06e6dbd1f3dc3c89b1903215a7a66115127.zip
66       CompatibleRuntimes:
67         - nodejs14.x
68       Description: Lambda Layer for ts-lambda
69     Metadata:
70       aws:cdk:path: ts-lambda-stack-develop/lambdaLayer/Resource
71       aws:asset:path: /Users/prasaddomala/Documents/Youtube/045-typescript-lambda-cdk/git-repo/yt-type
72       aws:asset:is-bundled: false
73       aws:asset:property: Content
74     publicfunctionfunction65B26B99:
75       Type: AWS::Lambda::Function
76       Properties:
77         Code:
78           S3Bucket: cdk-hnb659fds-assets-612659717478-ap-southeast-2
79           S3Key: 6524c803778d6198de0d33beb43a5d9d948c0bdcb8e6209746f313c24c5d075a.zip
80         Role:
81           Fn::GetAtt:
82             - lambdaRoleC844FDB
83             - Arn

```

We can use SAM local to execute the function id shown above locally. Install SAM CLI and Docker before invoking the command below

```

presaddomala@Prasad-M1-MBP:~/Documents/Youtube/045-typescript-lambda-cdk$ sam local invoke publicfunctionfunction65B26B99 --event event.json

```

```

Terminal Shell Edit View Session Scripts Profiles Toolbar Window Help
prasad@prasad-M1:~/Doc/Youtube/045-typescript-lambda-cdk/git-repo/yt-typescript-lambda-cdk$ sam local invoke publicfunctionfunction65B26B99 --no-event
> sam local invoke publicfunctionfunction65B26B99 --no-event
Invoking index.handler (nodejs14.x)
lambdaLayer857D4AF4 is a local Layer in the template
Building image......
Skip pulling image and use local one: samcli/lambda:nodejs14.x-x86_64-f652e5eff77eee496e6691153.

Mounting /Users/prasadomala/Documents/Youtube/045-typescript-lambda-cdk/git-repo/yt-typescript-lambda-cdk/cdk.out/asset.6524c803778d6198de0d33beb43a5d9d948c
0bdcbb8e6209746f313c24c5d075a as /var/task:ro, delegated inside runtime container
START RequestId: 8797a216-627b-4378-9e31-3ddce0ba8a42 Version: $LATEST
2022-04-19T14:26:45.030Z 8797a216-627b-4378-9e31-3ddce0ba8a42 INFO This is a Public Function!
END RequestId: 8797a216-627b-4378-9e31-3ddce0ba8a42
REPORT RequestId: 8797a216-627b-4378-9e31-3ddce0ba8a42 Init Duration: 1.09 ms Duration: 993.19 ms Billed Duration: 994 ms Memory Size: 1024 MB Max M
emory Used: 1024 MB
"This is a Public Function"*

```

This is how we execute individual Lambda functions locally using SAM

```

Terminal Shell Edit View Session Scripts Profiles Toolbar Window Help
cdk deploy
Bundling asset ts-lambda-stack-develop/public-function-function/Code/Stage...
cdk.out/bundling-temp-74d7a518c7ee626d829112ced1387b81d1c54c0da335c4de8c86acc6a6ec7ee3/index.js 2.3kb
⚡ Done in 1ms
Bundling asset ts-lambda-stack-develop/private-function-function/Code/Stage...
cdk.out/bundling-temp-577610e0cea7fcf1df7a94cdab8e92af02ca4c1ddf2d5b8023602649725c3e76/index.js 2.3kb
⚡ Done in 3ms
[Warning at /ts-lambda-stack-develop/privateSubnet0] No routeTableId was provided to the subnet 'subnet-029d25ea0d0b6a906'. Attempting to read its .routeTableId will return null/undefined. (More info: https://github.com/aws/aws-cdk/pull/3171)
[Warning at /ts-lambda-stack-develop/privateSubnet1] No routeTableId was provided to the subnet 'subnet-08b53dda246384b1f'. Attempting to read its .routeTableId will return null/undefined. (More info: https://github.com/aws/aws-cdk/pull/3171)
[Warning at /ts-lambda-stack-develop/privateSubnet2] No routeTableId was provided to the subnet 'subnet-01a855a9fe96b9ded'. Attempting to read its .routeTableId will return null/undefined. (More info: https://github.com/aws/aws-cdk/pull/3171)
⚡ Synthesis time: 2.8s

```

```

cdk deploy
e.routeTableId will return null/undefined. (More info: https://github.com/aws/aws-cdk/pull/3171)
[Warning at /ts-lambda-stack-develop/privateSubnet0] No routeTableId was provided to the subnet 'subnet-029d25ea0d0b6a906'. Attempting to read its .routeTableId will return null/undefined. (More info: https://github.com/aws/aws-cdk/pull/3171)
[Warning at /ts-lambda-stack-develop/privateSubnet1] No routeTableId was provided to the subnet 'subnet-08b53dda246384b1f'. Attempting to read its .routeTableId will return null/undefined. (More info: https://github.com/aws/aws-cdk/pull/3171)
[Warning at /ts-lambda-stack-develop/privateSubnet2] No routeTableId was provided to the subnet 'subnet-01a855a9fe96b9ded'. Attempting to read its .routeTableId will return null/undefined. (More info: https://github.com/aws/aws-cdk/pull/3171)
⚡ Synthesis time: 2.8s

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 |
|------------------|--------|-----------------------------------------------------------------------------------------------------------------------|------------------------------|-----------|
| +\$lambdaRoleArn | Allow  | sts:AssumeRole                                                                                                        | Service:lambda.amazonaws.com |           |
| +\$lambdaRole    | Allow  | logs:CreateLogGroup<br>logs:CreateLogStream<br>logs:DescribeLogGroups<br>logs:DescribeLogStreams<br>logs:PutLogEvents | AWS:\$lambdaRole             |           |



IAM Policy Changes


| Resource      | Managed Policy ARN                                    |
|---------------|-------------------------------------------------------|
| +\$lambdaRole | arn:\${AWS::Partition}:iam::aws:policy/ReadOnlyAccess |



Security Group Changes


| Group               | Dir | Protocol    | Peer            |
|---------------------|-----|-------------|-----------------|
| +\$lambdaSG.GroupId | In  | TCP 0-65535 | 10.3.0.0/16     |
| +\$lambdaSG.GroupId | Out | Everything  | Everyone (IPv4) |



(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_

```

```

  IAM Statement Changes
  + ${lambdaRole.Arn}      Allow   sts:AssumeRole
  + *                      Allow   logs:CreateLogGroup
                                logs:CreateLogStream
                                logs:DescribeLogGroups
                                logs:DescribeLogStreams
                                logs:PutLogEvents

```

IAM Policy Changes

Resource	Managed Policy ARN
+ \${lambdaRole}	arn:\${AWS::Partition}:iam::aws:policy/ReadOnlyAccess

Security Group Changes

Group	Dir	Protocol	Peer
+ \${lambdaSG.GroupId}	In	TCP 0-65535	10.3.0.0/16
+ \${lambdaSG.GroupId}	Out	Everything	Everyone (IPv4)

(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

ts-lambda-stack-develop: deploying...

```

[0%] start: Publishing 5947d7a0930311fd47fce02ef60bb06e6dbd1f3dc3c89b1903215a7a66115127:612659717478-ap-southeast-2
[0%] start: Publishing 6524c803778d6198de0d3beb43a5d9d948c0bdb8e6209746f313c24c5d075a:612659717478-ap-southeast-2
[0%] start: Publishing c993b10b13f96d4647db0cf6a9a7019710b34f61600695d887ad66e144f8a93c:612659717478-ap-southeast-2
[0%] start: Publishing 927a9977ebaabe440a1681101b202d46f1ae703cf491090e4ba01bf4890eb4d4:612659717478-ap-southeast-2
[25%] success: Published c993b10b13f96d4647db0cf6a9a7019710b34f61600695d887ad66e144f8a93c:612659717478-ap-southeast-2
[50%] success: Published 927a9977ebaabe440a1681101b202d46f1ae703cf491090e4ba01bf4890eb4d4:612659717478-ap-southeast-2
[75%] success: Published 6524c803778d6198de0d3beb43a5d9d948c0bdb8e6209746f313c24c5d075a:612659717478-ap-southeast-2
[100%] success: Published 5947d7a0930311fd47fce02ef60bb06e6dbd1f3dc3c89b1903215a7a66115127:612659717478-ap-southeast-2

```

ts-lambda-stack-develop: creating CloudFormation changeset...



```

  logs:DescribeLogStreams
  logs:PutLogEvents

```

IAM Policy Changes

Resource	Managed Policy ARN
+ \${lambdaRole}	arn:\${AWS::Partition}:iam::aws:policy/ReadOnlyAccess

Security Group Changes

Group	Dir	Protocol	Peer
+ \${lambdaSG.GroupId}	In	TCP 0-65535	10.3.0.0/16
+ \${lambdaSG.GroupId}	Out	Everything	Everyone (IPv4)

(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

ts-lambda-stack-develop: deploying...

```

[0%] start: Publishing 5947d7a0930311fd47fce02ef60bb06e6dbd1f3dc3c89b1903215a7a66115127:612659717478-ap-southeast-2
[0%] start: Publishing 6524c803778d6198de0d3beb43a5d9d948c0bdb8e6209746f313c24c5d075a:612659717478-ap-southeast-2
[0%] start: Publishing c993b10b13f96d4647db0cf6a9a7019710b34f61600695d887ad66e144f8a93c:612659717478-ap-southeast-2
[0%] start: Publishing 927a9977ebaabe440a1681101b202d46f1ae703cf491090e4ba01bf4890eb4d4:612659717478-ap-southeast-2
[25%] success: Published c993b10b13f96d4647db0cf6a9a7019710b34f61600695d887ad66e144f8a93c:612659717478-ap-southeast-2
[50%] success: Published 927a9977ebaabe440a1681101b202d46f1ae703cf491090e4ba01bf4890eb4d4:612659717478-ap-southeast-2
[75%] success: Published 6524c803778d6198de0d3beb43a5d9d948c0bdb8e6209746f313c24c5d075a:612659717478-ap-southeast-2
[100%] success: Published 5947d7a0930311fd47fce02ef60bb06e6dbd1f3dc3c89b1903215a7a66115127:612659717478-ap-southeast-2

```

ts-lambda-stack-develop: creating CloudFormation changeset...

[.....] (0/10)

```

12:27:31 am | CREATE_IN_PROGRESS | AWS::CloudFormation::Stack | ts-lambda-stack-develop
12:27:36 am | CREATE_IN_PROGRESS | AWS::Lambda::LayerVersion | lambdaLayer
12:27:36 am | CREATE_IN_PROGRESS | AWS::IAM::Role          | lambdaRole
12:27:36 am | CREATE_IN_PROGRESS | AWS::EC2::SecurityGroup | lambdaSG
12:27:36 am | CREATE_IN_PROGRESS | AWS::Logs::LogGroup    | fn-private-function-log-group
12:27:36 am | CREATE_IN_PROGRESS | AWS::Logs::LogGroup    | fn-public-function-log-group
12:27:38 am | CREATE_IN_PROGRESS | AWS::CDK::Metadata     | CDKMetadata/Default

```

Prasad Domala

```

Resource Managed Policy ARN
+ ${lambdaRole} arn:${AWS::Partition}:iam::aws:policy/ReadOnlyAccess

Security Group Changes
Group Dir Protocol Peer
+ ${lambdaSG.GroupId} In TCP 0-65535 10.3.0.0/16
+ ${lambdaSG.GroupId} Out Everything Everyone (IPv4)

(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
ts-lambda-stack-develop: deploying...
[0%] start: Publishing 5947d7a0930311fd47fce02ef60bb06e6dbd1f3dc3c89b1903215a7a66115127:612659717478-ap-southeast-2
[0%] start: Publishing 6524c803778d6198de0d33beb43a5d9d948c0bdcbe6209746f13c24c5d075a:612659717478-ap-southeast-2
[0%] start: Publishing c993b10b13f96d4647db0c6fa9a7019710b34f61600695d887ad66e144f8a93c:612659717478-ap-southeast-2
[0%] start: Publishing 927a9977ebaabe440a1681101b202d46f1ae703cf491090e4ba01bf4890eb4d4:612659717478-ap-southeast-2
[25%] success: Published c993b10b13f96d4647db0c6fa9a7019710b34f61600695d887ad66e144f8a93c:612659717478-ap-southeast-2
[50%] success: Published 927a9977ebaabe440a1681101b202d46f1ae703cf491090e4ba01bf4890eb4d4:612659717478-ap-southeast-2
[75%] success: Published 6524c803778d6198de0d33beb43a5d9d948c0bdcbe6209746f31c24c5d075a:612659717478-ap-southeast-2
[100%] success: Published 5947d7a0930311fd47fce02ef60bb06e6dbd1f3dc3c89b1903215a7a66115127:612659717478-ap-southeast-2
ts-lambda-stack-develop: creating CloudFormation changeset...
12:28:21 am | CREATE_FAILED | AWS::Lambda::Function | privatefunctionfunctionB21CF630

✖ ts-lambda-stack-develop failed: Error: The stack named ts-lambda-stack-develop failed creation, it may need to be manually deleted from the AWS console: ROLLBACK_COMPLETE
at waitForStackDeploy (/Users/prasadomala/.nvm/versions/node/v16.13.0/lib/node_modules/aws-cdk/lib/api/util/cloudformation.ts:311:11)
at processTicksAndRejections (node:internal/process/task_queues:96:5)
at prepareAndExecuteChangeSet (/Users/prasadomala/.nvm/versions/node/v16.13.0/lib/node_modules/aws-cdk/lib/api/deploy-stack.ts:376:26)
at CdkToolkit.deploy (/Users/prasadomala/.nvm/versions/node/v16.13.0/lib/node_modules/aws-cdk/cdk-toolkit.ts:209:24)
at initCommandLine (/Users/prasadomala/.nvm/versions/node/v16.13.0/lib/node_modules/aws-cdk/lib/cli.ts:342:12)

The stack named ts-lambda-stack-develop failed creation, it may need to be manually deleted from the AWS console: ROLLBACK_COMPLETE

```

We need to provide the necessary permissions for our Private Lambda function with the network role for creating network interfaces. AWS provides a managed policy with the necessary permissions to create private Lambda functions

```

lib > TS lambda-stack.ts > LambdaStack > constructor > lambdaRole
  11 import { getLambdaDefinitions, getFunctionProps } from './lambda-config';
  12
  13 export class LambdaStack extends Stack {
  14   constructor(scope: Construct, id: string, props: StackProps, context: CDKContext) {
  15     super(scope, id, props);
  16
  17     // Lambda Role
  18     const lambdaRole = new iam.Role(this, 'lambdaRole', {
  19       roleName: `${context.appName}-lambda-role-${context.environment}`,
  20       description: `Lambda role for ${context.appName}`,
  21       assumedBy: new iam.ServicePrincipal('lambda.amazonaws.com'),
  22       managedPolicies: [
  23         iam.ManagedPolicy.fromAwsManagedPolicyName('ReadOnlyAccess'),
  24         iam.ManagedPolicy.fromManagedPolicyArn(
  25           this,
  26           'lambdaVPCAccessPolicy',
  27           'arn:aws:iam::aws:policy/service-role/AWSLambdaVPCAccessExecutionRole'
  28         ),
  29       ],
  30     });
  31
  32     // Attach inline policies to Lambda role
  33     lambdaRole.attachInlinePolicy(
  34       new iam.Policy(this, 'lambdaExecutionAccess', {
  35         policyName: 'lambdaExecutionAccess',
  36       })
  37     );
  38   }
  39 }

  // Attach inline policies to Lambda role
  lambdaRole.attachInlinePolicy(
    new iam.Policy(this, 'lambdaExecutionAccess', {
      policyName: 'lambdaExecutionAccess',
    })
  );
}

```

We add the role as above

```

cdk deploy
Bundling asset ts-lambda-stack-develop/public-function-function/Code/Stage...
cdk.out/bundling-temp-74d7a518c7ee626d829112ced1387b81d1c54c0da335c4de8c86acc6a6ec7ee3/index.js 2.3kb
⚡ Done in 10ms
Bundling asset ts-lambda-stack-develop/private-function-function/Code/Stage...
cdk.out/bundling-temp-577610e0cea7fcf1df7a94cdab8e92af02ca4c1ddf2d5b8023602649725c3e76/index.js 2.3kb
⚡ Done in 1ms
[Warning at /ts-lambda-stack-develop/privateSubnet0] No routeTableId was provided to the subnet 'subnet-029d25ea0d0b6a906'. Attempting to read its .routeTable.routeTableId will return null/undefined. (More info: https://github.com/aws/aws-cdk/pull/3171)
[Warning at /ts-lambda-stack-develop/privateSubnet1] No routeTableId was provided to the subnet 'subnet-08b53dda246384b1f'. Attempting to read its .routeTable.routeTableId will return null/undefined. (More info: https://github.com/aws/aws-cdk/pull/3171)
[Warning at /ts-lambda-stack-develop/privateSubnet2] No routeTableId was provided to the subnet 'subnet-01a855a9fe96b9ded'. Attempting to read its .routeTable.routeTableId will return null/undefined. (More info: https://github.com/aws/aws-cdk/pull/3171)

+ Synthesis time: 3.45s
-
```

IAM Policy Changes

Resource	Managed Policy ARN
+ \${lambdaRole.Arn}	arn:\${AWS::Partition}:iam::aws:policy/ReadOnlyAccess
+ *	arn:aws:iam::aws:policy/service-role/AWSLambdaVPCAccessExecutionRole

Security Group Changes

Group	Dir	Protocol	Peer
+ \${lambdaSG.GroupId}	In	TCP 0-65535	10.3.0.0/16
+ \${lambdaSG.GroupId}	Out	Everything	Everyone (IPv4)

(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

ts-lambda-stack-develop: deploying...

[0%] start: Publishing 5947d7a0930311fd47fce02ef60bb06e6dbd1f3dc3c89b1903215a7a66115127:612659717478-ap-southeast-2
[0%] start: Publishing 6524c803778d6198de0d33beb43a5d9d48c0bdb8e6209746f313c24c5d075a:612659717478-ap-southeast-2
[0%] start: Publishing c993b10b13f96d4647db0cfc9a7019710b34f61600695d887ad66e144f8a93c:612659717478-ap-southeast-2
[0%] start: Publishing 48909bda17f130fcfa868f6bb8c95ec371b987071a164487083a1142f016566f:612659717478-ap-southeast-2
[25%] success: Published c993b10b13f96d4647db0cfc9a7019710b34f61600695d887ad66e144f8a93c:612659717478-ap-southeast-2
[50%] success: Published 5947d7a0930311fd47fce02ef60bb06e6dbd1f3dc3c89b1903215a7a66115127:612659717478-ap-southeast-2
[75%] success: Published 6524c803778d6198de0d33beb43a5d9d48c0bdb8e6209746f313c24c5d075a:612659717478-ap-southeast-2
[100%] success: Published 48909bda17f130fcfa868f6bb8c95ec371b987071a164487083a1142f016566f:612659717478-ap-southeast-2

ts-lambda-stack-develop: creating CloudFormation changeset...

IAM Policy Changes

Resource	Managed Policy ARN
+ \${lambdaRole}	arn:\${AWS::Partition}:iam::aws:policy/ReadOnlyAccess
+ \${lambdaRole}	arn:aws:iam::aws:policy/service-role/AWSLambdaVPCAccessExecutionRole

Security Group Changes

Group	Dir	Protocol	Peer
+ \${lambdaSG.GroupId}	In	TCP 0-65535	10.3.0.0/16
+ \${lambdaSG.GroupId}	Out	Everything	Everyone (IPv4)

(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

ts-lambda-stack-develop: deploying...

[0%] start: Publishing 5947d7a0930311fd47fce02ef60bb06e6dbd1f3dc3c89b1903215a7a66115127:612659717478-ap-southeast-2
[0%] start: Publishing 6524c803778d6198de0d33beb43a5d9d48c0bdb8e6209746f313c24c5d075a:612659717478-ap-southeast-2
[0%] start: Publishing c993b10b13f96d4647db0cfc9a7019710b34f61600695d887ad66e144f8a93c:612659717478-ap-southeast-2
[0%] start: Publishing 48909bda17f130fcfa868f6bb8c95ec371b987071a164487083a1142f016566f:612659717478-ap-southeast-2
[25%] success: Published c993b10b13f96d4647db0cfc9a7019710b34f61600695d887ad66e144f8a93c:612659717478-ap-southeast-2
[50%] success: Published 5947d7a0930311fd47fce02ef60bb06e6dbd1f3dc3c89b1903215a7a66115127:612659717478-ap-southeast-2
[75%] success: Published 6524c803778d6198de0d33beb43a5d9d48c0bdb8e6209746f313c24c5d075a:612659717478-ap-southeast-2
[100%] success: Published 48909bda17f130fcfa868f6bb8c95ec371b987071a164487083a1142f016566f:612659717478-ap-southeast-2

ts-lambda-stack-develop: creating CloudFormation changeset...

12:54:25 am | CREATE_IN_PROGRESS | AWS::CloudFormation::Stack | ts-lambda-stack-develop
12:55:09 am | CREATE_IN_PROGRESS | AWS::IAM::Policy | lambdaExecutionAccess
12:55:10 am | CREATE_IN_PROGRESS | AWS::Lambda::Function | private-function-function
12:55:10 am | CREATE_IN_PROGRESS | AWS::Lambda::Function | public-function-function

Resource	Managed Policy ARN	
+	<code>#{lambdaRole}</code>	<code>arn:\${AWS::Partition}::iam::aws:policy/ReadOnlyAccess</code>
+	<code>#{lambdaRole}</code>	<code>arn:aws:iam::aws:policy/service-role/AWSLambdaVPCAccessExecutionRole</code>

Security Group Changes

	Group	Dir	Protocol	Peer
+	<code>#{lambdaSG.GroupId}</code>	In	TCP 0-65535	10.3.0.0/16
+	<code>#{lambdaSG.GroupId}</code>	Out	Everything	Everyone (IPv4)

(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
ts-lambda-stack-develop: deploying...
[%] start: Publishing 5947d7a0930311fd47fce02ef60bb06e6dbd1f3dc3c89b1903215a7a66115127:612659717478-ap-southeast-2
[%] start: Publishing 6524c803778d6198de0d33beb43a5d9d948c0bdcbe6209746f313c24c5d075a:612659717478-ap-southeast-2
[%] start: Publishing c993b10b13f96d4647db0cf6a9a7019710b34f616006950887ad66e144f8a93c:612659717478-ap-southeast-2
[%] start: Publishing 48909bda17f130fcfa86f6bb08c95ec371b987071a16487083a1142f016566f:612659717478-ap-southeast-2
[25%] success: Published c993b10b13f96d4647db0cf6a9a7019710b34f616006950887ad66e144f8a93c:612659717478-ap-southeast-2
[50%] success: Published 5947d7a0930311fd47fce02ef60bb06e6dbd1f3dc3c89b1903215a7a66115127:612659717478-ap-southeast-2
[75%] success: Published 6524c803778d6198de0d33beb43a5d9d948c0bdcbe6209746f313c24c5d075a:612659717478-ap-southeast-2
[100%] success: Published 48909bda17f130fcfa86f6bb08c95ec371b987071a164487083a1142f016566f:612659717478-ap-southeast-2
ts-lambda-stack-develop: creating CloudFormation changeset...
✓ ts-lambda-stack-develop

* Deployment time: 183.34s

Stack ARN:
arn:aws:cloudformation:ap-southeast-2:612659717478:stack/ts-lambda-stack-develop/97613830-bff0-11ec-8bbd-0ac3edc67df0

* Total time: 186.78s
```

PRASAD DOMALA

Now the deployment is successful.

Console Home

Welcome to AWS

- Getting started with AWS
- Training and certification
- What's new with AWS?

CloudFormation

Stacks (2)

Stack name	Status	Created time	Description
ts-lambda-stack-develop	CREATE_COMPLETE	2022-04-20 00:54:19 UTC+1000	This is the Stack Description
CDKToolkit	CREATE_COMPLETE	2022-04-11 22:11:46 UTC+1000	This stack includes resources needed to deploy AWS CDK apps into this environment

CloudFormation > Stacks > ts-lambda-stack-develop

ts-lambda-stack-develop

Stack info

Overview

Stack ID arn:aws:cloudformation:ap-southeast-2:612659717478:stack/ts-lambda-stack-develop/97613830-bff0-11ec-8bbd-0ac3edc67df0	Description This is the Stack Description
Status CREATE_COMPLETE	Status reason
Root stack	Parent stack
Created time 2022-04-20 00:54:19 UTC+1000	Deleted time
Updated time 2022-04-20 00:54:25 UTC+1000	
Drift status NOT_CHECKED	Last drift check time
Termination protection	IAM role



CloudFormation > Stacks > ts-lambda-stack-develop

ts-lambda-stack-develop

Resources

Logical ID	Physical ID	Type	Status	Status reason	Module
CDKMetadata	97613830-bff0-11ec-8bbd-0ac3edc67df0	AWS::CDK::Metadata	CREATE_COMPLETE		
fnpprivatefunctionoggrouppFAA6666E	/aws/lambda/ts-lambda-private-function-develop	AWS::Logs::LogGroup	CREATE_COMPLETE		
fnppublicfunctionoggrouppA0A81846	/aws/lambda/ts-lambda-public-function-develop	AWS::Logs::LogGroup	CREATE_COMPLETE		
lambdaExecutionAccess\$B2ECF6E2	ts-lambda-lamb-1WX45LYXHHM8K	AWS::IAM::Policy	CREATE_COMPLETE		



Screenshot of the AWS CloudFormation console showing the Stacks page for the 'ts-lambda-stack-develop' stack.

Stack details:

- Stack ID: ts-lambda-stack-develop
- Created: 2022-04-20 00:54:19 UTC+1000
- Status: CREATE_COMPLETE

Resources (9):

Logical ID	Physical ID	Type	Status	Status reason	Module
lambdaExecutionAccess82ECF6E2	ts-la-lamb-1WX45LYXHHM8K	AWS::IAM::Policy	CREATE_COMPLETE	-	-
lambdaLayer857D4AF4	arn:aws:lambda:ap-southeast-2:612659717478:layer:lambdaLayer857D4AF4:5	AWS::Lambda::LayerVersion	CREATE_COMPLETE	-	-
lambdaRoleC844FDB1	ts-lambda-lambda-role-develop	AWS::IAM::Role	CREATE_COMPLETE	-	-
lambdaSG81E771DE	sg-03c4607e5334576e1	AWS::EC2::SecurityGroup	CREATE_COMPLETE	-	-
privatefunctionfunctionB21CF630	ts-lambda-private-function-develop	AWS::Lambda::Function	CREATE_COMPLETE	-	-
publicfunctionfunction65B26B99	ts-lambda-public-function-develop	AWS::Lambda::Function	CREATE_COMPLETE	-	-

Screenshot of the AWS Lambda console showing the function details for 'ts-lambda-public-function-develop'.

Function overview:

- Function name: ts-lambda-public-function-develop
- Related functions: Select a function
- Description: Last modified 4 minutes ago
- Function ARN: arn:aws:lambda:ap-southeast-2:612659717478:function:ts-lambda-public-function-develop
- Application: ts-lambda-stack-develop
- Function URL: Info

Function tabs:

- Code
- Test
- Monitor
- Configuration
- Aliases
- Versions

Your applications Documentation - Stack to learn AWS Lambda - ts-lambda-public-function-dev

https://ap-southeast-2.console.aws.amazon.com/lambda/home?region=ap-southeast-2/functions/lambda-public-function-develop?tab=code

AWS Services Search for services, features, blogs, docs, and more [Option+S]

Sydney AdministratorAccess/prasad.domala@kheriox.tech

ts-lambda-stack-develop

Function URL Info

Code Test Monitor Configuration Aliases Versions

Code source Info

Upload from

File Edit Find View Go Tools Window Test Deploy

Go to Anything (% P)

Environment ts-lambda-public-fu index.js

```
1 var __defProp = Object.defineProperty;
2 var __getOwnPropertyDesc = Object.getOwnPropertyDescriptor;
3 var __getOwnPropertyNames = Object.getOwnPropertyNames;
4 var __hasOwnProp = Object.prototype.hasOwnProperty;
5 var __markAsModule = (target) => __defProp(target, "__esModule", { value: true });
6 var __export = (target, all) => {
7   for (var name in all)
8     __defProp(target, name, { get: all[name], enumerable: true });
9 };
10 var __reExport = (target, module2, copyDefault, desc) => {
11   if (!module2 && typeof module2 === "object" || typeof module2 === "function") {
12     for (let key of __getOwnPropertyNames(module2)) {
13       if (!__hasOwnProp.call(target, key) && (copyDefault || key !== "default"))
14         __defProp(target, key, { get: () => module2[key], enumerable: !(desc = __getOwnPropertyDesc(module2, key)) || desc.enumerable });
15     }
16     return target;
17   }
18 var __toCommonJS = /* @__PURE__ */ ((cache) => {
19   return (module2, temp) => {
20     return cache && cache.get(module2) || (temp = __reExport(__markAsModule({}), module2, 1), cache && cache.set(module2, temp), temp);
21   };
22 })(typeof WeakMap !== "undefined" ? /* @__PURE__ */ new WeakMap() : {});
```

Your applications Documentation - Stack to learn AWS Lambda - ts-lambda-public-function-dev

https://ap-southeast-2.console.aws.amazon.com/lambda/home?region=ap-southeast-2/functions/lambda-public-function-develop?tab=code

AWS Services Search for services, features, blogs, docs, and more [Option+S]

Sydney AdministratorAccess/prasad.domala@kheriox.tech

ts-lambda-stack-develop

Function URL Info

Code Test Monitor Configuration Aliases Versions

Code source Info

Upload from

File Edit Find View Go Tools Window Test Deploy

Go to Anything (% P)

Environment ts-lambda-public-fu index.js

```
12 for (let key of __getOwnPropertyNames(module2))
13   if (!__hasOwnProp.call(target, key) && (copyDefault || key !== "default"))
14     __defProp(target, key, { get: () => module2[key], enumerable: !(desc = __getOwnPropertyDesc(module2, key)) || desc.enumerable });
15   }
16   return target;
17 }
18 var __toCommonJS = /* @__PURE__ */ ((cache) => {
19   return (module2, temp) => {
20     return cache && cache.get(module2) || (temp = __reExport(__markAsModule({}), module2, 1), cache && cache.set(module2, temp), temp);
21   };
22 })(typeof WeakMap !== "undefined" ? /* @__PURE__ */ new WeakMap() : {});
23 // lambda-handlers/public-function.ts
25 var public_function_exports = {};
26 __export(public_function_exports, {
27   handler: () => handler
28 });
29 // lambda-layer/utils.ts
31 var logInfo = (message, title = void 0) => {
32   if (typeof message === "string") {
33     title ? console.info(`${title}: ${message}`) : console.info(message);
34   } else if (Array.isArray(message)) {
35     message.forEach(item => logInfo(item));
36   }
37 }
```

Your applications

Search for services, features, blogs, docs, and more [Option+S]

Sydney AdministratorAccess/prasad.domala@kheriox.tech

Code Test Monitor Configuration Aliases Versions

Code source Info

File Edit Find View Go Tools Window Test Deploy

Go to Anything (% P)

index.js

```

1 var __defProp = Object.defineProperty;
2 var __getOwnPropDesc = Object.getOwnPropertyDescriptor;
3 var __getOwnPropertyNames = Object.getOwnPropertyNames;
4 var __hasOwnProp = Object.prototype.hasOwnProperty;
5 var __markAsModule = (target) => __defProp(target, "__esModule", { value: true });
6 var __export = (target, all) => {
7   for (var name in all)
8     __defProp(target, name, { get: all[name], enumerable: true });
9 };
10 var __reExport = (target, module2, copyDefault, desc) => {
11   if (!module2 || typeof module2 === "object" || typeof module2 === "function") {
12     if (let key of __getOwnPropertyNames(module2))
13       if (!__hasOwnProp.call(target, key) && (copyDefault || key !== "default"))
14         __defProp(target, key, { get: () => module2[key], enumerable: !(desc = __getOwnPropDesc(module2, key)) || desc.enumerable });
15   }
16   return target;
17 };
18 var __toCommonJS = /* @__PURE__ */ ((cache) => {
19   return (module2, temp) => {
20     return cache && cache.get(module2) || (temp = __reExport(__markAsModule({}), module2, 1), cache && cache.set(module2, temp), temp);
21   };
22 })(typeof WeakMap !== "undefined" ? /* @__PURE__ */ new WeakMap() : 0);
23 // lambda-handlers/public-function.ts
25 var public_function_exports = {};
26 __export(public_function_exports, {
27   handler: () => handler
28 });
29
30 // lambda-layer/utils.ts
31

```

PRASAD DOMALA

Search for services, features, blogs, docs, and more [Option+S]

Sydney AdministratorAccess/prasad.domala@kheriox.tech

Code Test Monitor Configuration Aliases Versions

General configuration

Triggers

Permissions

Destinations

Function URL - new

Environment variables

Tags

VPC

Monitoring and operations tools

General configuration Info

Description -

Memory 1024 MB

Ephemeral storage 512 MB

AWS Compute Optimizer Opt in to see memory recommendations for your Lambda functions. [View details](#)

Search for services, features, blogs, docs, and more [Option+S]

Sydney AdministratorAccess/prasad.domala@kheriox.tech

Code Test Monitor Configuration Aliases Versions

General configuration

Triggers

Permissions

Destinations

Function URL - new

Environment variables (4)

The environment variables below are encrypted at rest with the default Lambda service key.

Key	Value
AWS_NODEJS_CONNECTION_REUSE_ENABLED	1
ENV	develop
GIT_BRANCH	develop
REGION	ap-southeast-2

Sydney | AdministratorAccess/prasad.domala@kheriox.tech

Services Search for services, features, blogs, docs, and more [Option+S]

Code Test Monitor Configuration Aliases Versions

General configuration

Tags (4) Info

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value
aws:cloudformation:stack-name	ts-lambda-stack-develop
aws:cloudformation:stack-id	arn:aws:cloudformation:ap-southeast-2:612659717478:stack/ts-lambda-stack-develop/97613830-bff0-11ec-8bbd-0ac3edc67df0
aws:cloudformation:logical-id	publicfunctionfunction65B26B99
Environment	develop

Manage tags

VPC Monitoring and operations tools

Sydney | AdministratorAccess/prasad.domala@kheriox.tech

Lambda > Functions > ts-lambda-private-function-develop

ts-lambda-private-function-develop

This function belongs to an application. Click here to manage it.

Function overview Info

Related functions: Select a function

Description: -

Last modified: 6 minutes ago

Function ARN: arn:aws:lambda:ap-southeast-2:612659717478:function:ts-lambda-private-function-develop

Application: ts-lambda-stack-develop

Function URL: Info

Code Test Monitor Configuration Aliases Versions

Sydney | AdministratorAccess/prasad.domala@kheriox.tech

Services Search for services, features, blogs, docs, and more [Option+S]

Code Test Monitor Configuration Aliases Versions

General configuration

VPC Info

VPC: vpc-009efe79ac997c618 (10.3.0.0/16) | kheriox-demo-dev-vpc

Subnets:

- subnet-029d25ea0db6a906 (10.3.96.0/20) | ap-southeast-2a, Private-2a
- subnet-08b53dd246384b1f (10.3.112.0/20) | ap-southeast-2b, Private-2b
- subnet-01a855a9fe96b9ded (10.3.128.0/20) | ap-southeast-2c, Private-2c

Security groups:

- sg-03c4607e5334576e1 (ts-lambda-lambda-security-group-develop)

Inbound rules | **Outbound rules**

Security group ID	Protocol	Ports	Source
sg-03c4607e5334576e1	All TCP	0 - 65535	10.3.0.0/16

Sydney | AdministratorAccess/prasad.domala@kheriox.tech

Code Test Monitor Configuration Aliases Versions

General configuration

Triggers Permissions Destinations Function URL - new Environment variables Tags VPC Monitoring and operations tools

General configuration Info

Description: - Memory: 2048 MB Ephemeral storage: 512 MB

Timeout: 5 min 0 sec

AWS Compute Optimizer: Opt in to see memory recommendations for your Lambda functions. View details

Sydney | AdministratorAccess/prasad.domala@kheriox.tech

Code Test Monitor Configuration Aliases Versions

General configuration Triggers Permissions Destinations Function URL - new Environment variables Tags VPC Monitoring and operations tools

Tags (4) Info

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value
aws:cloudformation:stack-name	ts-lambda-stack-develop
aws:cloudformation:stack-id	arn:aws:cloudformation:ap-southeast-2:612659717478:stack/ts-lambda-stack-develop/97613830-bff0-11ec-8bbd-0ac3edc67df0
aws:cloudformation:logical-id	publicfunctionfunction65B26B99
Environment	develop

Sydney | AdministratorAccess/prasad.domala@kheriox.tech

Code Test Monitor Configuration Aliases Versions

General configuration

Triggers Permissions Destinations Function URL - new Environment variables Tags VPC Monitoring and operations tools

General configuration Info

Description: - Memory: 1024 MB Ephemeral storage: 512 MB

Timeout: 15 min 0 sec

AWS Compute Optimizer: Opt in to see memory recommendations for your Lambda functions. View details