

AWS CDK share resources between stacks



Concise Developer
307 subscribers

Subscribe

20



530 views Mar 21, 2023

Here we will see how we can share resources with AWS CDK in a TypeScript project

Full CDK course:

<https://www.udemy.com/course/aws-type...>

0:00 intro

0:46 import props inside stack

2:47 inject props inside stack

3:10 export info from stack

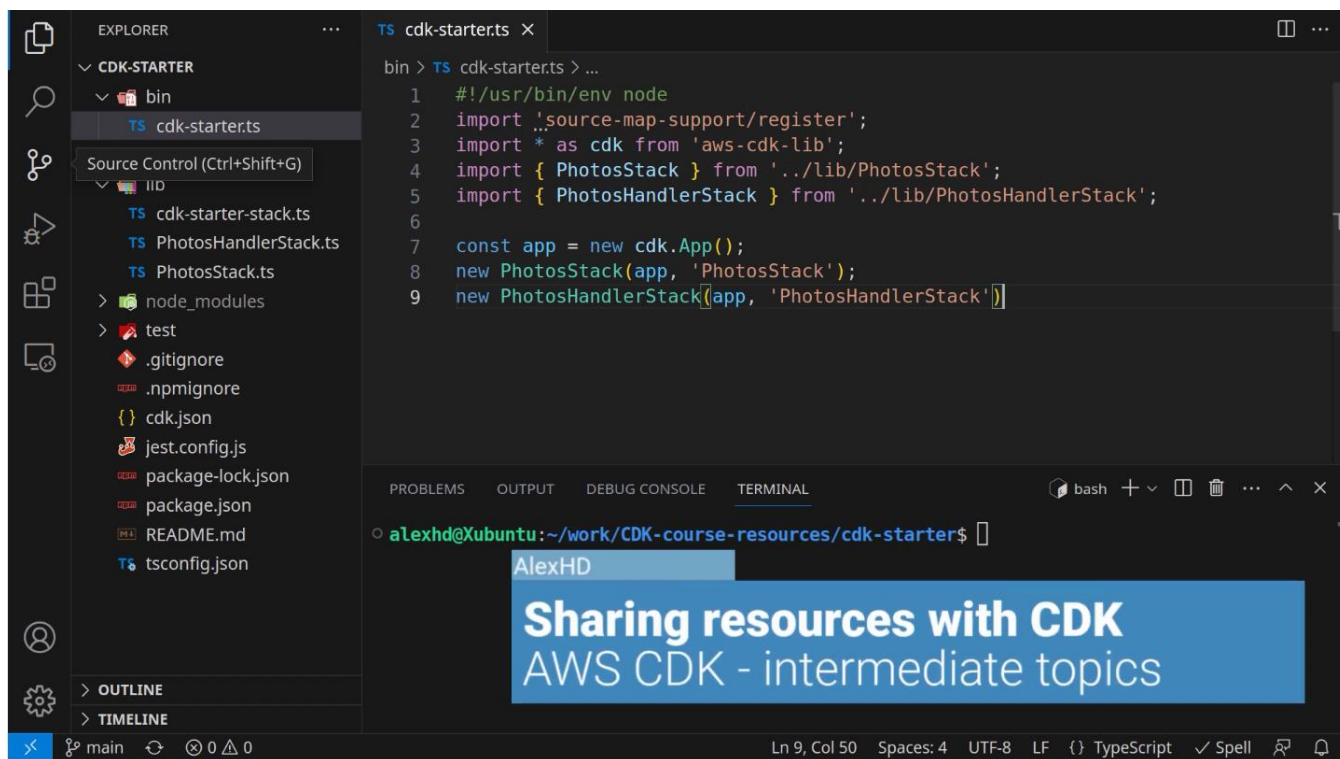
4:34 use info when stack is created

5:12 cdk synth

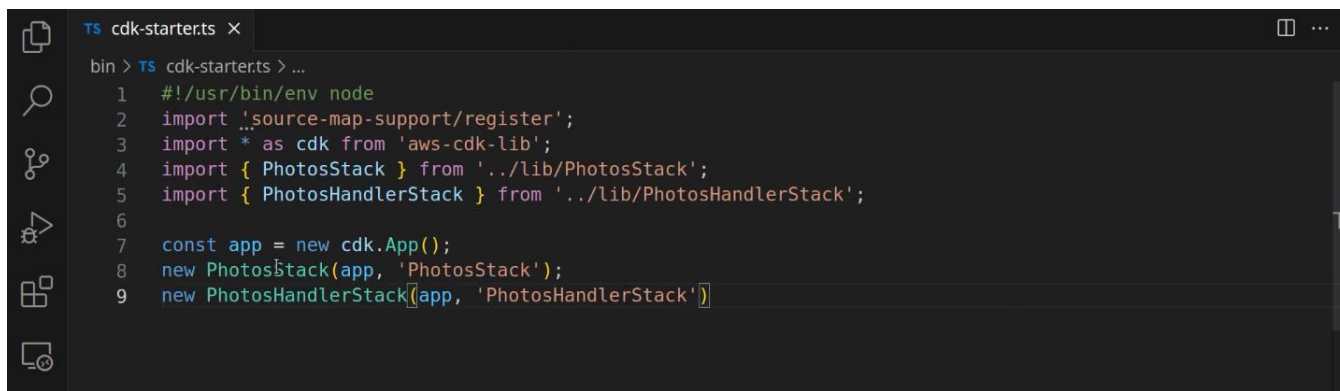
5:27 cdk deploy

6:29 stacks inside the console

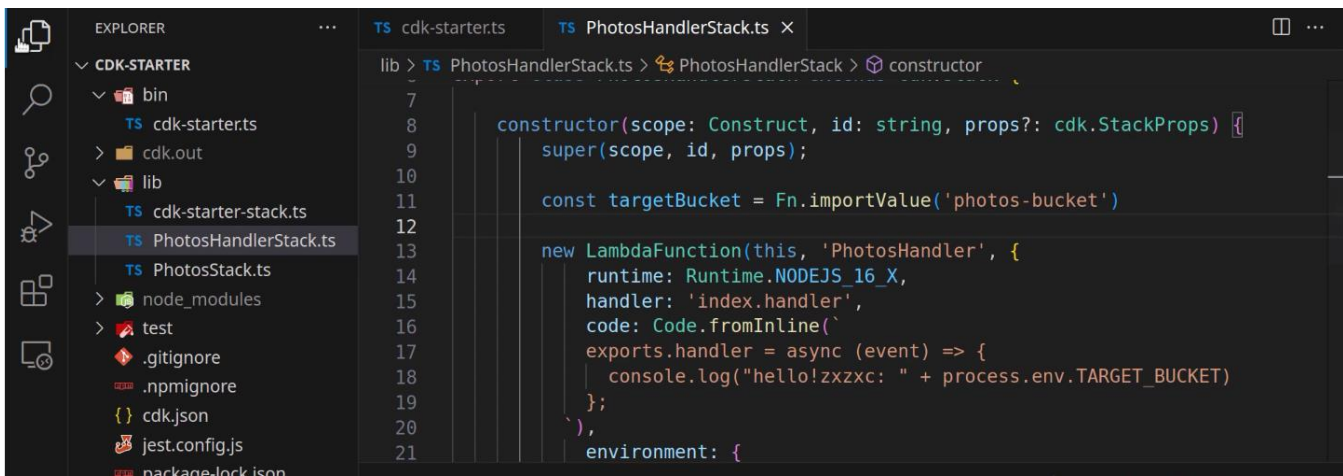
7:14 conclusions



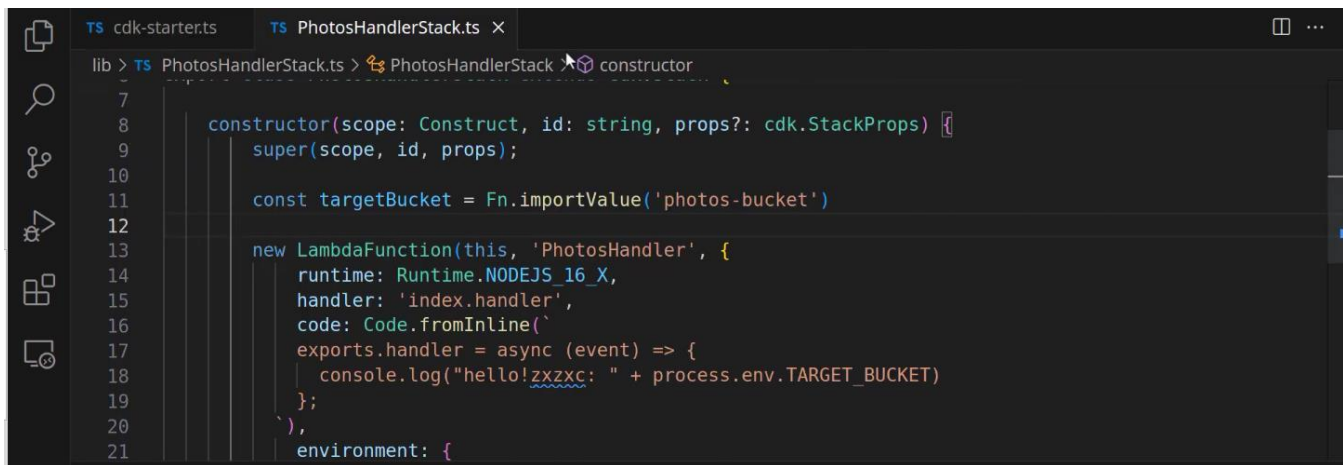
Let us see how we can share data between different CFN stacks using CDK



We have 2 stacks, **PhotosStack** that creates an S3 bucket and a **PhotosHandlerStack** that needs that S3 bucket to put data into it. How can we share data between these 2 stacks using only TypeScript features?

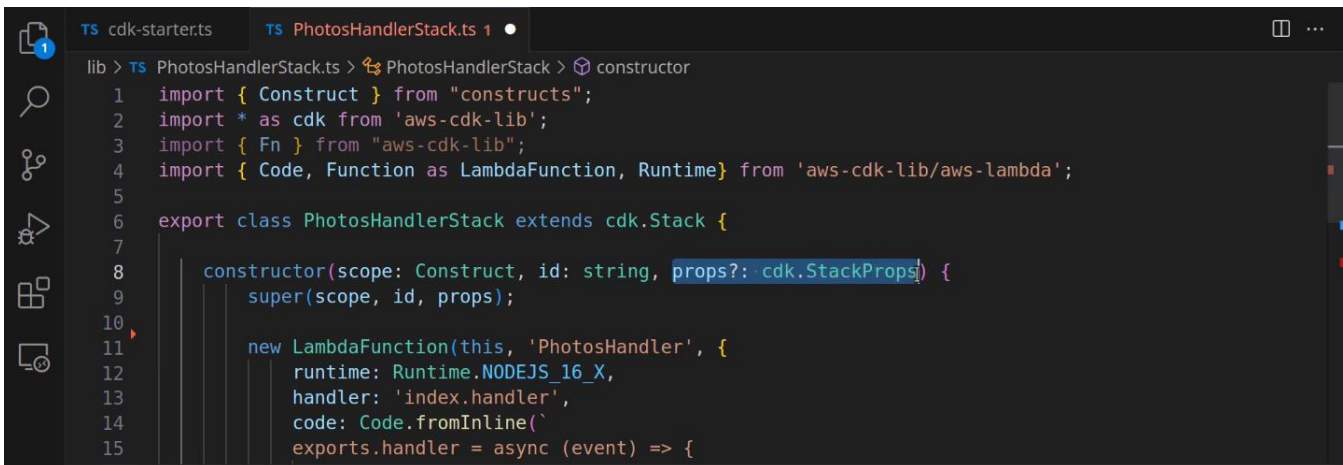


```
lib > TS PhotosHandlerStack.ts > PhotosHandlerStack > constructor
7
8     constructor(scope: Construct, id: string, props?: cdk.StackProps) {
9         super(scope, id, props);
10
11         const targetBucket = Fn.importValue('photos-bucket')
12
13         new LambdaFunction(this, 'PhotosHandler', {
14             runtime: Runtime.NODEJS_16_X,
15             handler: 'index.handler',
16             code: Code.fromInline(`
17                 exports.handler = async (event) => {
18                     console.log("hello!zxxxc: " + process.env.TARGET_BUCKET)
19                 };
20             `),
21             environment: {
```



```
lib > TS PhotosHandlerStack.ts > PhotosHandlerStack > constructor
7
8     constructor(scope: Construct, id: string, props?: cdk.StackProps) {
9         super(scope, id, props);
10
11         const targetBucket = Fn.importValue('photos-bucket')
12
13         new LambdaFunction(this, 'PhotosHandler', {
14             runtime: Runtime.NODEJS_16_X,
15             handler: 'index.handler',
16             code: Code.fromInline(`
17                 exports.handler = async (event) => {
18                     console.log("hello!zxxxc: " + process.env.TARGET_BUCKET)
19                 };
20             `),
21             environment: {
```

To use shared data, we no longer need line 11 above



```
lib > TS PhotosHandlerStack.ts > PhotosHandlerStack > constructor
1  import { Construct } from "constructs";
2  import * as cdk from 'aws-cdk-lib';
3  import { Fn } from "aws-cdk-lib";
4  import { Code, Function as LambdaFunction, Runtime } from 'aws-cdk-lib/aws-lambda';
5
6  export class PhotosHandlerStack extends cdk.Stack {
7
8      constructor(scope: Construct, id: string, props?: cdk.StackProps) {
9          super(scope, id, props);
10
11          new LambdaFunction(this, 'PhotosHandler', {
12              runtime: Runtime.NODEJS_16_X,
13              handler: 'index.handler',
14              code: Code.fromInline(`
15                  exports.handler = async (event) => {
```

```
TS cdk-starters.ts TS PhotosHandlerStack.ts 1
lib > TS PhotosHandlerStack.ts > ...
1 import { Construct } from "constructs";
2 import * as cdk from 'aws-cdk-lib';
3 import { Fn } from "aws-cdk-lib";
4 import { Code, Function as LambdaFunction, Runtime } from 'aws-cdk-lib/aws-lambda';
5
6
7
8 export class PhotosHandlerStack extends cdk.Stack {
9     interface StackProps
10     constructor(scope: Construct, id: string, props?: cdk.StackProps) {
11         super(scope, id, props);
12
13         new LambdaFunction(this, 'PhotosHandler', {
14             runtime: Runtime.NODEJS_16_X,
15             handler: 'index.handler',
```

```
TS cdk-starters.ts TS PhotosHandlerStack.ts 1
lib > TS PhotosHandlerStack.ts > PhotosHandlerStackProps
1 import { Construct } from "constructs";
2 import * as cdk from 'aws-cdk-lib';
3 import { Fn } from "aws-cdk-lib";
4 import { Code, Function as LambdaFunction, Runtime } from 'aws-cdk-lib/aws-lambda';
5
6 interface PhotosHandlerStackProps extends cdk.StackProps {
7
8 }
9
10 export class PhotosHandlerStack extends cdk.Stack {
11
12     constructor(scope: Construct, id: string, props?: cdk.StackProps) {
13         super(scope, id, props);
14
15         new LambdaFunction(this, 'PhotosHandler', {
```

```
TS cdk-starters.ts 1 TS PhotosHandlerStack.ts 1
lib > TS PhotosHandlerStack.ts > PhotosHandlerStack > constructor
4 import { Code, Function as LambdaFunction, Runtime } from 'aws-cdk-lib/aws-lambda';
5
6 interface PhotosHandlerStackProps extends cdk.StackProps {
7     targetBucketArn: string
8 }
9
10 export class PhotosHandlerStack extends cdk.Stack {
11
12     constructor(scope: Construct, id: string, props: PhotosHandlerStackProps) {
13         super(scope, id, props);
14
15         new LambdaFunction(this, 'PhotosHandler', {
16             runtime: Runtime.NODEJS_16_X,
17             handler: 'index.handler',
18             code: Code.fromInline(`
19             exports.handler = async (event) => {
```

```
lib > TS PhotosHandlerStack.ts > PhotosHandlerStack > constructor > environment > TARGET_BUCKET
11
12 constructor(scope: Construct, id: string, props: PhotosHandlerStackProps) {
13     super(scope, id, props);
14
15     new LambdaFunction(this, 'PhotosHandler', {
16         runtime: Runtime.NODEJS_16_X,
17         handler: 'index.handler',
18         code: Code.fromInline(`
19             exports.handler = async (event) => {
20                 console.log("hello!zxzxc: " + process.env.TARGET_BUCKET)
21             };
22         `),
23         environment: {
24             TARGET_BUCKET: targetBucket,
25         },
26     });
27 }
```

```
lib > TS PhotosHandlerStack.ts > PhotosHandlerStack > constructor > environment > TARGET_BUCKET
6 interface PhotosHandlerStackProps extends cdk.StackProps {
7     targetBucketArn: string
8 }
9
10 export class PhotosHandlerStack extends cdk.Stack {
11
12     constructor(scope: Construct, id: string, props: PhotosHandlerStackProps) {
13         super(scope, id, props);
14
15         new LambdaFunction(this, 'PhotosHandler', {
16             runtime: Runtime.NODEJS_16_X,
17             handler: 'index.handler',
18             code: Code.fromInline(`
19                 exports.handler = async (event) => {
20                     console.log("hello!zxzxc: " + process.env.TARGET_BUCKET)
21                 };
22             `),
23             environment: {
24                 TARGET_BUCKET: props.targetBucketArn,
25             },
26         });
27     }
28 }
29
30
31 }
```

This is how we can retrieve the information from the stack properties as shared data instead

```
bin > TS cdk-starters.ts > ...
1 #!/usr/bin/env node
2 import 'source-map-support/register';
3 import * as cdk from 'aws-cdk-lib';
4 import { PhotosStack } from '../lib/PhotosStack';
5 import { PhotosHandlerStack } from '../lib/PhotosHandlerStack';
6
7 const app = new cdk.App();
8 new PhotosStack(app, 'PhotosStack');
9 new PhotosHandlerStack(app, 'PhotosHandlerStack');
```

We now need to change the stack being initialized


```
TS cdk-starters 1 • TS PhotosHandlerStack.ts M
bin > TS cdk-starters > ...
1  #!/usr/bin/env node
2  import 'source-map-support/register';
3  import * as cdk from 'aws-cdk-lib';
4  import { PhotosStack } from '../lib/PhotosStack';
5  import { PhotosHandlerStack } from '../lib/PhotosHandlerStack';
6
7  const app = new cdk.App();
8  new PhotosStack(app, 'PhotosStack');
9  new PhotosHandlerStack(app, 'PhotosHandlerStack', {
10     targetBucketArn:
11  })
```

We now need to export the targetBucketArn value from the PhotoStack object that will create it


```
TS cdk-starters 1 • TS PhotosHandlerStack.ts M
bin > TS cdk-starters > ...
1  #!/usr/bin/env node
2  import 'source-map-support/register';
3  import * as cdk from 'aws-cdk-lib';
4  import { PhotosStack } from '../lib/PhotosStack';
5  import { PhotosHandlerStack } from '../lib/PhotosHandlerStack';
6
7  const app = new cdk.App();
8  new PhotosStack(app, 'PhotosStack');
9  new
10  (alias) new PhotosStack(scope: Construct, id: string, props?:
11  cdk.StackProps | undefined): PhotosStack
import PhotosStack
constructor(scope: Construct, id: string, props?: cdk.StackProps) {
    super(scope, id, props);

    this.initializeSuffix();


    const photosBucket = new Bucket(this, 'PhotosBucket2', {
        bucketName: `photos-bucket-${this.stackSuffix}`
    });
```

```
TS cdk-starters 1 • TS PhotosStack.ts X TS PhotosHandlerStack.ts M
lib > TS PhotosStack.ts > PhotosStack > constructor
10  constructor(scope: Construct, id: string, props?: cdk.StackProps) {
11      super(scope, id, props);
12
13      this.initializeSuffix();
14
15      const photosBucket = new Bucket(this, 'PhotosBucket2', {
16          bucketName: `photos-bucket-${this.stackSuffix}`
17      });
18
19      new CfnOutput(this, 'photos-bucket', {
20          value: photosBucket.bucketArn,
21          exportName: 'photos-bucket'
22      })
23  }
24
25  private initializeSuffix(){
26      const shortStackId = Fn.select(2, Fn.split('/', this.stackId))
27      this.stackSuffix = Fn.select(4, Fn.split('-', shortStackId))
28  }
29  }
```

```
lib > TS PhotosStack.ts > PhotosStack > constructor
3 import { Bucket, CfnBucket } from "aws-cdk-lib/aws-s3";
4 import { CfnOutput, Fn } from "aws-cdk-lib";
5
6 export class PhotosStack extends cdk.Stack {
7
8     private stackSuffix: string;
9
10    constructor(scope: Construct, id: string, props?: cdk.StackProps) {
11        super(scope, id, props);
12
13        this.initializeSuffix();
14
15        const photosBucket = new Bucket(this, 'PhotosBucket2', {
16            bucketName: `photos-bucket-${this.stackSuffix}`
17        });
18
19        new CfnOutput(this, 'photos-bucket', {
20            value: photosBucket.bucketArn,
21            exportName: 'photos-bucket'
22        });
23    }
24
25    private initializeSuffix(){
26        const shortStackId = Fn.select(2, Fn.split('/', this.stackId))
27        this.stackSuffix = Fn.select(4, Fn.split('-', shortStackId))
28    }
}
```



```
lib > TS PhotosStack.ts > PhotosStack > stackSuffix
3 import { Bucket, CfnBucket } from "aws-cdk-lib/aws-s3";
4 import { CfnOutput, Fn } from "aws-cdk-lib";
5
6 export class PhotosStack extends cdk.Stack {
7
8     private stackSuffix: string;
9
10
11    constructor(scope: Construct, id: string, props?: cdk.StackProps) {
12        super(scope, id, props);
13
14        this.initializeSuffix();
15
16        const photosBucket = new Bucket(this, 'PhotosBucket2', {
17            bucketName: `photos-bucket-${this.stackSuffix}`
18        });
19    }
20
21    private initializeSuffix(){
22        const shortStackId = Fn.select(2, Fn.split('/', this.stackId))
23        this.stackSuffix = Fn.select(4, Fn.split('-', shortStackId))
24    }
25 }
```



```
TS cdk-starters.ts 1 • TS PhotosStack.ts • TS PhotosHandlerStack.ts M
lib > TS PhotosStack.ts > PhotosStack > photosBucketArn
3 import { Bucket, CfnBucket } from "aws-cdk-lib/aws-s3";
4 import { CfnOutput, Fn } from "aws-cdk-lib";
5
6 export class PhotosStack extends cdk.Stack {
7
8     private stackSuffix: string;
9     public readonly photosBucketArn: string;
10
11     constructor(scope: Construct, id: string, props?: cdk.StackProps) {
12         super(scope, id, props);
13
14         this.initializeSuffix();
15
16         const photosBucket = new Bucket(this, 'PhotosBucket2', {
17             bucketName: `photos-bucket-${this.stackSuffix}`
18         });
19     }
20
21     private initializeSuffix(){
22         const shortStackId = Fn.select(2, Fn.split('/', this.stackId))
23         this.stackSuffix = Fn.select(4, Fn.split('-', shortStackId))
24     }
25 }
```



```
TS cdk-starters.ts 1 • TS PhotosStack.ts M X TS PhotosHandlerStack.ts M
lib > TS PhotosStack.ts > PhotosStack > constructor
3 import { Bucket, CfnBucket } from "aws-cdk-lib/aws-s3";
4 import { CfnOutput, Fn } from "aws-cdk-lib";
5
6 export class PhotosStack extends cdk.Stack {
7
8     private stackSuffix: string;
9     public readonly photosBucketArn: string;
10
11     constructor(scope: Construct, id: string, props?: cdk.StackProps) {
12         super(scope, id, props);
13
14         this.initializeSuffix();
15
16         const photosBucket = new Bucket(this, 'PhotosBucket2', {
17             bucketName: `photos-bucket-${this.stackSuffix}`
18         });
19         this.photosBucketArn = photosBucket.bucketArn;
20     }
21
22     private initializeSuffix(){
23         const shortStackId = Fn.select(2, Fn.split('/', this.stackId))
24         this.stackSuffix = Fn.select(4, Fn.split('-', shortStackId))
25     }
26 }
```



We now have the information available; we need to pass it to the other stack next.

```
TS cdk-starters.ts 1 • TS PhotosStack.ts M TS PhotosHandlerStack.ts M
bin > TS cdk-starters.ts > ...
1 #!/usr/bin/env node
2 import 'source-map-support/register';
3 import * as cdk from 'aws-cdk-lib';
4 import { PhotosStack } from '../lib/PhotosStack';
5 import { PhotosHandlerStack } from '../lib/PhotosHandlerStack';
6
7 const app = new cdk.App();
8 new PhotosStack(app, 'PhotosStack');
9 new PhotosHandlerStack(app, 'PhotosHandlerStack', {
10     targetBucketArn:
11 }
```



```
TS cdk-starters.ts 2 • TS PhotosStack.ts M TS PhotosHandlerStack.ts M
bin > TS cdk-starters.ts > ...
1  #!/usr/bin/env node
2  import { sourceMapSupport } from 'source-map-support/register';
3  import * as cdk from 'aws-cdk-lib';
4  import { PhotosStack } from '../lib/PhotosStack';
5  import { PhotosHandlerStack } from '../lib/PhotosHandlerStack';
6
7  const app = new cdk.App();
8  new PhotosStack(app, 'PhotosStack').photosBucketArn;
9  new PhotosHandlerStack(app, 'PhotosHandlerStack', {
10     targetBucketArn:
11 })
```

```
TS cdk-starters.ts 1 • TS PhotosStack.ts M TS PhotosHandlerStack.ts M
bin > TS cdk-starters.ts > photosStack
1  #!/usr/bin/env node
2  import { sourceMapSupport } from 'source-map-support/register';
3  import * as cdk from 'aws-cdk-lib';
4  import { PhotosStack } from '../lib/PhotosStack';
5  import { PhotosHandlerStack } from '../lib/PhotosHandlerStack';
6
7  const app = new cdk.App();
8  const photosStack = new PhotosStack(app, 'PhotosStack');
9  new PhotosHandlerStack(app, 'PhotosHandlerStack', {
10     targetBucketArn:
11 })
```

We are creating this reference so that we use it in the other stack

```
TS cdk-starters.ts × TS PhotosStack.ts M TS PhotosHandlerStack.ts M
bin > TS cdk-starters.ts > targetBucketArn
1  #!/usr/bin/env node
2  import { sourceMapSupport } from 'source-map-support/register';
3  import * as cdk from 'aws-cdk-lib';
4  import { PhotosStack } from '../lib/PhotosStack';
5  import { PhotosHandlerStack } from '../lib/PhotosHandlerStack';
6
7  const app = new cdk.App();
8  const photosStack = new PhotosStack(app, 'PhotosStack');
9  new PhotosHandlerStack(app, 'PhotosHandlerStack', {
10     targetBucketArn: photosStack.photosBucketArn
11 })
```

This is what we need to do. Let's test to see if it worked.

The screenshot shows the AWS CloudFormation console for the 'CDKToolkit' stack. On the left, a sidebar lists the stacks, with 'CDKToolkit' selected and showing a 'CREATE_COMPLETE' status. The main panel displays the 'Events' tab for the 'CDKToolkit' stack, showing a list of 36 events. The events table includes columns for Timestamp, Logical ID, Status, and Status reason. The first event is 'CDKToolkit' at '2023-03-17 14:42:20 UTC+0200' with a status of 'CREATE_COMPLETE'. Subsequent events show the creation of 'DeploymentActionRole', 'FilePublishingRoleDefaultPolicy', and 'ImagePublishingRoleDefaultPolicy', all with 'CREATE_COMPLETE' status. A circular video feed of a person is visible on the right side of the events table.

Timestamp	Logical ID	Status	Status reason
2023-03-17 14:42:20 UTC+0200	CDKToolkit	CREATE_COMPLETE	-
2023-03-17 14:42:18 UTC+0200	DeploymentActionRole	CREATE_COMPLETE	
2023-03-17 14:42:17 UTC+0200	FilePublishingRoleDefaultPolicy	CREATE_COMPLETE	
2023-03-17 14:42:16 UTC+0200	ImagePublishingRoleDefaultPolicy	CREATE_COMPLETE	
2023-03-17 14:41:58			

We need to delete the other stacks so that we can test our new code.

The screenshot shows a code editor with three tabs: 'cdk-starters.ts', 'PhotosStack.ts', and 'PhotosHandlerStack.ts'. The 'cdk-starters.ts' tab is active, displaying the following TypeScript code:

```
1  #!/usr/bin/env node
2  import 'source-map-support/register';
3  import * as cdk from 'aws-cdk-lib';
4  import { PhotosStack } from '../lib/PhotosStack';
5  import { PhotosHandlerStack } from '../lib/PhotosHandlerStack';
6
7  const app = new cdk.App();
8  const photosStack = new PhotosStack(app, 'PhotosStack');
9  new PhotosHandlerStack(app, 'PhotosHandlerStack', {
10     targetBucketArn: photosStack.photosBucketArn
11 })
```

Below the code editor, a terminal window shows the output of the 'cdk synth' command:

```
alexhd@Xubuntu:~/work/CDK-course-resources/cdk-starter$ cdk synth
Successfully synthesized to /home/alexhd/work/CDK-course-resources/cdk-starter/cdk.out
Supply a stack id (PhotosStack, PhotosHandlerStack) to display its template.
alexhd@Xubuntu:~/work/CDK-course-resources/cdk-starter$
```

A circular video feed of a person is visible in the bottom right corner of the terminal window.

We can call the **cdk synth** command to check that our code is okay and will work.

node + - [] [] ... ^ x

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

-approval broadening).
Please confirm you intend to make the following modifications:


IAM Statement Changes

	Resource	Effect	Action	Principal	Condition
+	<code>\${PhotosHandler/ServiceRole.Arn}</code>	Allow	sts:AssumeRole	Service:lambda.amazonaws.com	

IAM Policy Changes

	Resource	Managed Policy ARN
+	<code>\${PhotosHandler/ServiceRole}</code>	<code>arn:\${AWS::Partition}:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole</code>

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



node + - [] [] ... ^ x


PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

IAM Policy Changes

	Resource	Managed Policy ARN
+	<code>\${PhotosHandler/ServiceRole}</code>	<code>arn:\${AWS::Partition}:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole</code>

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

Do you wish to deploy these changes (y/n)? y
PhotosHandlerStack: deploying... [2/2]
[0%] start: Publishing 00b51b8e418bef3753079a7099ca1af0f14b4a191c764b55e44cae4dac1edc0c:current_account-current_region
[100%] success: Published 00b51b8e418bef3753079a7099ca1af0f14b4a191c764b55e44cae4dac1edc0c:current_account-current_region
PhotosHandlerStack: creating CloudFormation changeset...



bash + - [] [] ... ^ x

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

[0%] start: Publishing 00b51b8e418bef3753079a7099ca1af0f14b4a191c764b55e44cae4dac1edc0c:current_account-current_region
[100%] success: Published 00b51b8e418bef3753079a7099ca1af0f14b4a191c764b55e44cae4dac1edc0c:current_account-current_region
PhotosHandlerStack: creating CloudFormation changeset...


✔ PhotosHandlerStack

⚡ Deployment time: 63.18s

Stack ARN:
arn:aws:cloudformation:eu-west-1:151873756016:stack/PhotosHandlerStack/945529b0-c7e9-11ed-ace1

⚡ Total time: 70.57s

alexhd@Xubuntu: ~/work/CDK-course-resources/cdk-starter\$



CloudFormation - Stacks

eu-west-1.console.aws.amazon.com/cloudformation/home?region=eu-west-1#/stacks/events?filteringText=&filteringStatus=active&vi...

Services Search [Alt+S]

Resource Groups & Tag Editor

CloudFormation > Stacks > CDKToolkit

Stacks (3)

Filter by stack name

Active View nested

Stacks

- PhotosHandlerStack
2023-03-21 15:09:15 UTC+0200
CREATE_COMPLETE
- PhotosStack
2023-03-21 15:08:16 UTC+0200
CREATE_COMPLETE
- CDKToolkit
2023-03-17 14:41:21 UTC+0200
CREATE_COMPLETE

CDKToolkit

Delete Update Stack actions Create stack

Stack info Events Resources Outputs Parameters Template Change sets

Events (36)

Search events

Timestamp	Logical ID	Status	Status reason
2023-03-17 14:42:20 UTC+0200	CDKToolkit	CREATE_COMPLETE	-
2023-03-17 14:42:18 UTC+0200	DeploymentActionRole	CREATE_COMPLETE	-
2023-03-17 14:42:17 UTC+0200	FilePublishingRoleDefaultPolicy	CREATE_COMPLETE	-
2023-03-17 14:42:16 UTC+0200	ImagePublishingRoleDefaultPolicy	CREATE_COMPLETE	-
2023-03-17 14:41:58			

Feedback Language

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudFormation - Stacks

eu-west-1.console.aws.amazon.com/cloudformation/home?region=eu-west-1#/stacks/events?filteringText=&filteringStatus=active&vi...

Services Search [Alt+S]

Resource Groups & Tag Editor

CloudFormation > Stacks > PhotosStack

Stacks (3)

Filter by stack name

Active View nested

Stacks

- PhotosHandlerStack
2023-03-21 15:09:15 UTC+0200
CREATE_COMPLETE
- PhotosStack
2023-03-21 15:08:16 UTC+0200
CREATE_COMPLETE
- CDKToolkit
2023-03-17 14:41:21 UTC+0200
CREATE_COMPLETE

PhotosStack

Delete Update Stack actions Create stack

Stack info Events Resources Outputs Parameters Template Change sets

Events (9)

Search events

Timestamp	Logical ID	Status	Status reason
2023-03-21 15:08:50 UTC+0200	PhotosStack	CREATE_COMPLETE	-
2023-03-21 15:08:48 UTC+0200	PhotosBucket20FA6ECA3	CREATE_COMPLETE	-
2023-03-21 15:08:28 UTC+0200	CDKMetadata	CREATE_COMPLETE	-
2023-03-21 15:08:28 UTC+0200	CDKMetadata	CREATE_IN_PROGRESS	-
2023-03-21 15:08:27			

Feedback Language

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudFormation - Stacks

eu-west-1.console.aws.amazon.com/cloudformation/home?region=eu-west-1#/stacks/outputs?filteringText=&filteringStatus=active&...

Services Search [Alt+S]

Resource Groups & Tag Editor

CloudFormation > Stacks > PhotosStack

Stacks (3)

Filter by stack name

Active View nested

Stacks

- PhotosHandlerStack
2023-03-21 15:09:15 UTC+0200
CREATE_COMPLETE
- PhotosStack**
2023-03-21 15:08:16 UTC+0200
CREATE_COMPLETE
- CDKToolkit
2023-03-17 14:41:21 UTC+0200
CREATE_COMPLETE

PhotosStack

Delete Update Stack actions Create stack

Stack info Events Resources **Outputs** Parameters Template Change sets

Outputs (1)

Search outputs

Key	Value	Description	Export name
ExportsOutputFnGetAttPhotosBucket20FA6ECA3ArnBBC E0F84	arn:aws:s3:::photos-bucket-0ab63adc97bf	-	PhotosStack:ExportsOutputFnGetAttPhotosBucket20FA6ECA3ArnBBC E0F84

Feedback Language

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudFormation - Stacks

eu-west-1.console.aws.amazon.com/cloudformation/home?region=eu-west-1#/stacks/outputs?filteringText=&filteringStatus=active&...

Services Search [Alt+S]

Resource Groups & Tag Editor

CloudFormation > Stacks > PhotosHandlerStack

Stacks (3)

Filter by stack name

Active View nested

Stacks

- PhotosHandlerStack**
2023-03-21 15:09:15 UTC+0200
CREATE_COMPLETE
- PhotosStack
2023-03-21 15:08:16 UTC+0200
CREATE_COMPLETE
- CDKToolkit
2023-03-17 14:41:21 UTC+0200
CREATE_COMPLETE

PhotosHandlerStack

Delete Update Stack actions Create stack

Stack info Events Resources **Outputs** Parameters Template Change sets

Outputs

Search outputs

No outputs

There are no outputs defined in your template

Feedback Language

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudFormation - Stack

eu-west-1.console.aws.amazon.com/cloudformation/home?region=eu-west-1#/stacks/resources?filteringText=&filteringStatus=active...

Services Search [Alt+S]

Resource Groups & Tag Editor

CloudFormation > Stacks > PhotosHandlerStack

Stacks (3)

Filter by stack name

Active View nested

1

Stacks

PhotosHandlerStack

2023-03-21 15:09:15 UTC+0200

CREATE_COMPLETE

PhotosStack

2023-03-21 15:08:16 UTC+0200

CREATE_COMPLETE

CDKToolkit

2023-03-17 14:41:21 UTC+0200

CREATE_COMPLETE

PhotosHandlerStack

Delete Update Stack actions Create stack

Stack info Events Resources Outputs Parameters Template Change sets

Resources (3)

Tree view Flat view

Search resources

Logical ID	Physical ID	Type	Status	Module
PhotosHandler	-	-	CREATE_COMPLETE	-
CDKMetadata	-	-	CREATE_CO	-

Feedback

Language

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudFormation - Stack

eu-west-1.console.aws.amazon.com/cloudformation/home?region=eu-west-1#/stacks/resources?filteringText=&filteringStatus=active...

Services Search [Alt+S]

Resource Groups & Tag Editor

CloudFormation > Stacks > PhotosHandlerStack

Stacks (3)

Filter by stack name

Active View nested

1

Stacks

PhotosHandlerStack

2023-03-21 15:09:15 UTC+0200

CREATE_COMPLETE

PhotosStack

2023-03-21 15:08:16 UTC+0200

CREATE_COMPLETE

CDKToolkit

2023-03-17 14:41:21 UTC+0200

CREATE_COMPLETE

PhotosHandlerStack

Delete Update Stack actions Create stack

Stack info Events Resources Outputs Parameters Template Change sets

Resources (3)

Tree view Flat view

Search resources

Logical ID	Physical ID	Type	Status	Module
PhotosHandler	-	-	CREATE_COMPLETE	-
ServiceRole	-	-	CREATE_CO	-
PhotosHandler7038E0D9	PhotosHandlerStack-PhotosHandler7038E0D9-EAcL7xYoA...	AWS::Lambda::Function	CREA	-
CDKMetadata	-	-	CREA	-

Feedback

Language

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudFormation - Stack PPhotosHandlerStack-P

eu-west-1.console.aws.amazon.com/lambda/home?region=eu-west-1#/functions/PhotosHandlerStack-PhotosHandler7038E0D9-EAcL... | Ireland | AlexH

Resource Groups & Tag Editor

ApplicationPhotosHandlerStackFunction URL Info

CodeTestMonitorConfigurationAliasesVersions

General configurationTriggersPermissionsDestinationsFunction URLEnvironment variablesTagsVPC

Environment variables (1)

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

Key	Value
TARGET_BUCKET	arn:aws:s3::photos-bucket-0ab63adc97bf

Edit

FeedbackLanguage© 2023, Amazon Web Services, Inc. or its affiliates. PrivacyTermsCookie preferences

A circular inset video feed showing a man with short brown hair and glasses, wearing a dark jacket, speaking into a professional microphone. He is positioned in front of a wooden wall and a guitar is partially visible behind him.