IGOR SOROKA • JAWS PANKRATION • NOV. 20, 2021

IGOR SOROKA • JAWS PANKRATION • NOV. 20, 2021

IGOR SOROKA • JAWS PANKRATION • NOV. 20, 2021

サーバレスユースケースのための実コードとしてのインフラストラクチャ

INFRASTRUCTURE AS REAL CODE

Serverless use case

JAWS PANKRATION • NOV. 20, 2021

PRESENTATION HIGHLIGHTS

- CDK introduction
- Other tools for IaC
- CDK mechanism
- Serverless examples with CDK
- Conclusions

IGOR SOROKA

- Based in **Helsinki, Finland**
- **Serverless** Contractor
- Long-distance **runner**

@grenguar

AWS community builders

aws certified Developer Associate

aws certified Solutions Architect Associate

aws certified Cloud Practitioner

MY AWS STORY

2018 First look at AWS Lambda

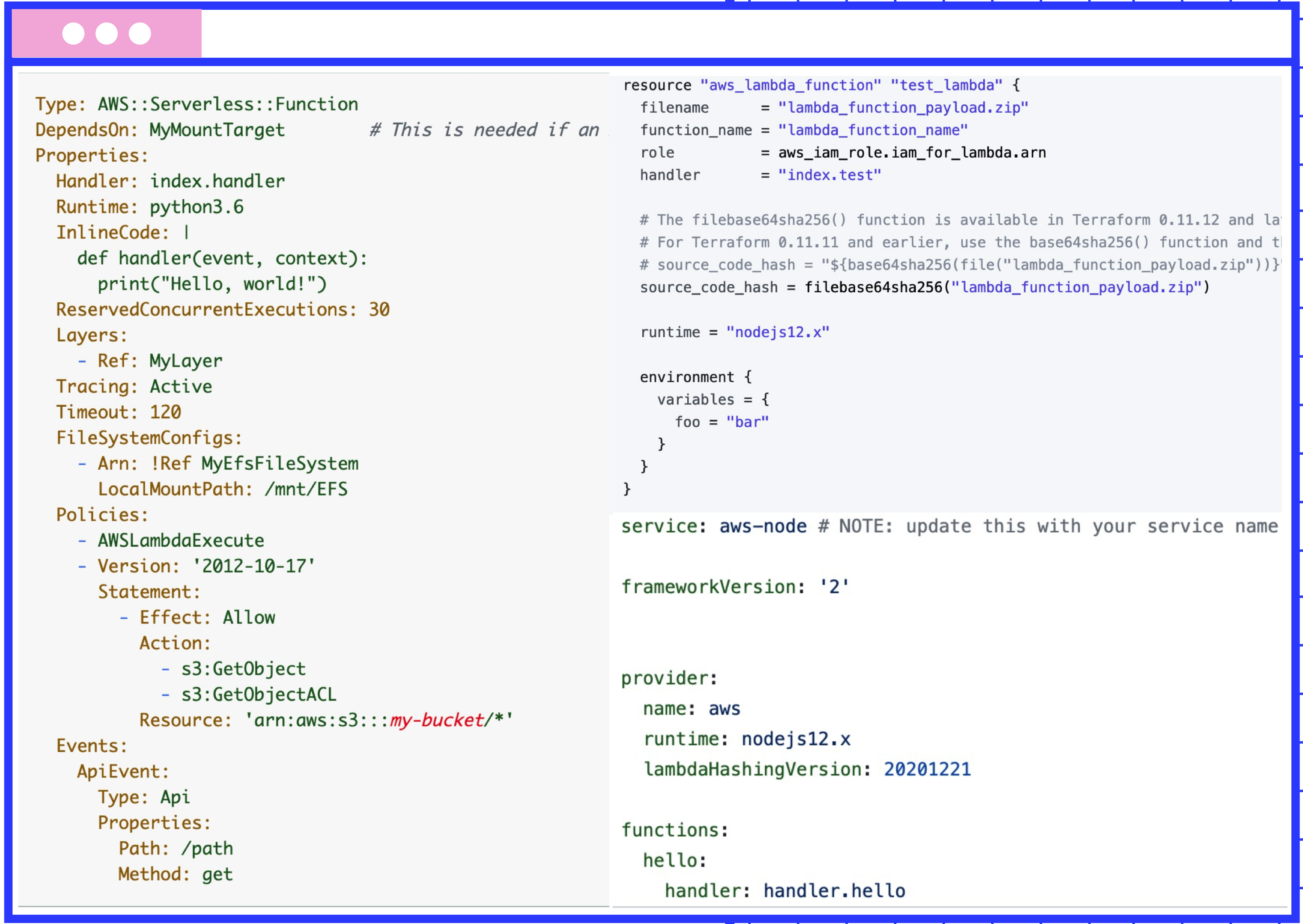
2019 Serverless framework and Terraform

2020 Discovered CDK

2021 Launched 3 projects done with CDK (including CI/CD)

POPULAR DEPLOYMENT TOOLS

- CloudFormation/SAM
- Terraform
- Serverless
- Pulumi



```yaml
Type: AWS::Serverless::Function
DependsOn: MyMountTarget        # This is needed if an
Properties:
  Handler: index.handler
  Runtime: python3.6
  InlineCode: |
    def handler(event, context):
      print("Hello, world!")
  ReservedConcurrentExecutions: 30
  Layers:
    - Ref: MyLayer
  Tracing: Active
  Timeout: 120
  FileSystemConfigs:
    - Arn: !Ref MyEfsFileSystem
      LocalMountPath: /mnt/EFS
  Policies:
    - AWSLambdaExecute
    - Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Action:
            - s3:GetObject
            - s3:GetObjectACL
          Resource: 'arn:aws:s3:::my-bucket/*'
  Events:
    ApiEvent:
      Type: Api
      Properties:
        Path: /path
        Method: get
```

```hcl
resource "aws_lambda_function" "test_lambda" {
  filename      = "lambda_function_payload.zip"
  function_name = "lambda_function_name"
  role          = aws_iam_role.iam_for_lambda.arn
  handler       = "index.test"

  # The filebase64sha256() function is available in Terraform 0.11.12 and la
  # For Terraform 0.11.11 and earlier, use the base64sha256() function and t
  # source_code_hash = "${base64sha256(file("lambda_function_payload.zip"))}
  source_code_hash = filebase64sha256("lambda_function_payload.zip")

  runtime = "nodejs12.x"

  environment {
    variables = {
      foo = "bar"
    }
  }
}
```

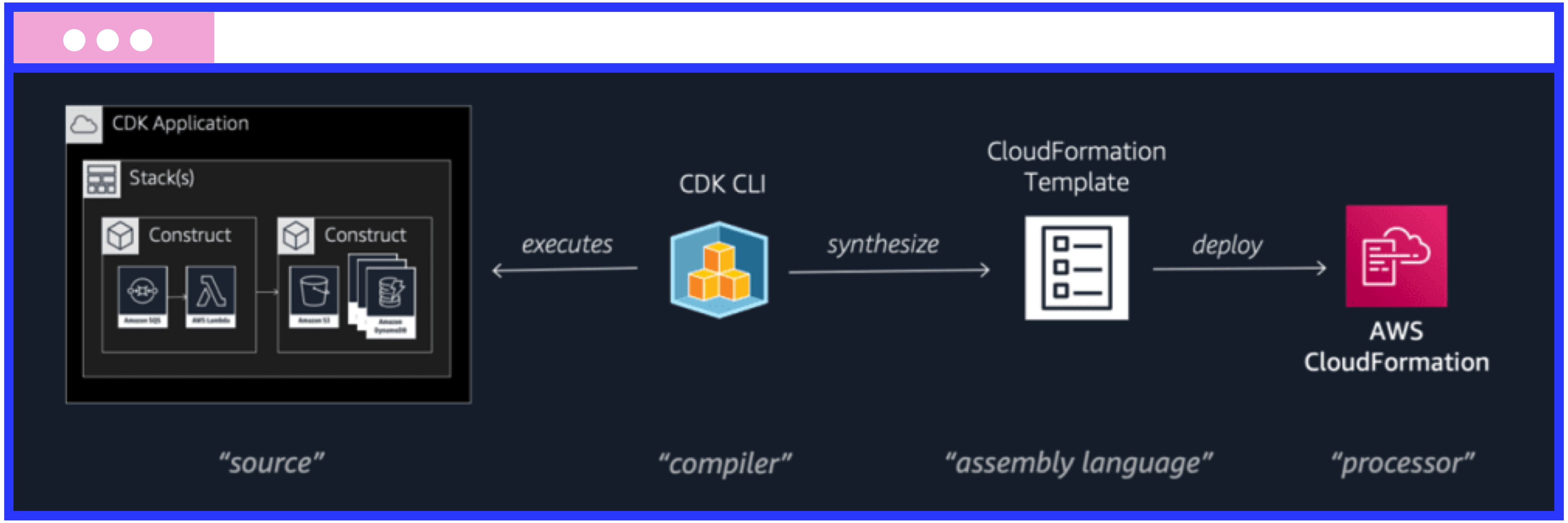
```yaml
service: aws-node # NOTE: update this with your service name

frameworkVersion: '2'


provider:
  name: aws
  runtime: nodejs12.x
  lambdaHashingVersion: 20201221

functions:
  hello:
    handler: handler.hello
```

WHAT IS CDK?

BRIEF INTRODUCTION

"An open-source software development framework to define your cloud application resources using familiar programming languages.”
AWS CDK introduction page

WHAT CDK DOES FOR YOU?



CONSTRUCTS

LOGICALLY SEPARATED PIECES OF INFRASTRUCTURE

Level 1 – AWS CloudFormation Only

Level 2 – Curated

Level 3 – Patterns and whole architectures

WHAT IS INSIDE A SERVERLESS PROJECT?

A classic serverless architecture will consist of:

- AWS Lambda – compute service for the business logic
- API Gateway – for REST endpoints
- S3 – for storage
- SNS/SQS – async tasks
- EventBridge/CloudWatch Events
- DynamoDB – NoSQL Database

BILLING ALERT CDK EXAMPLE

1/2

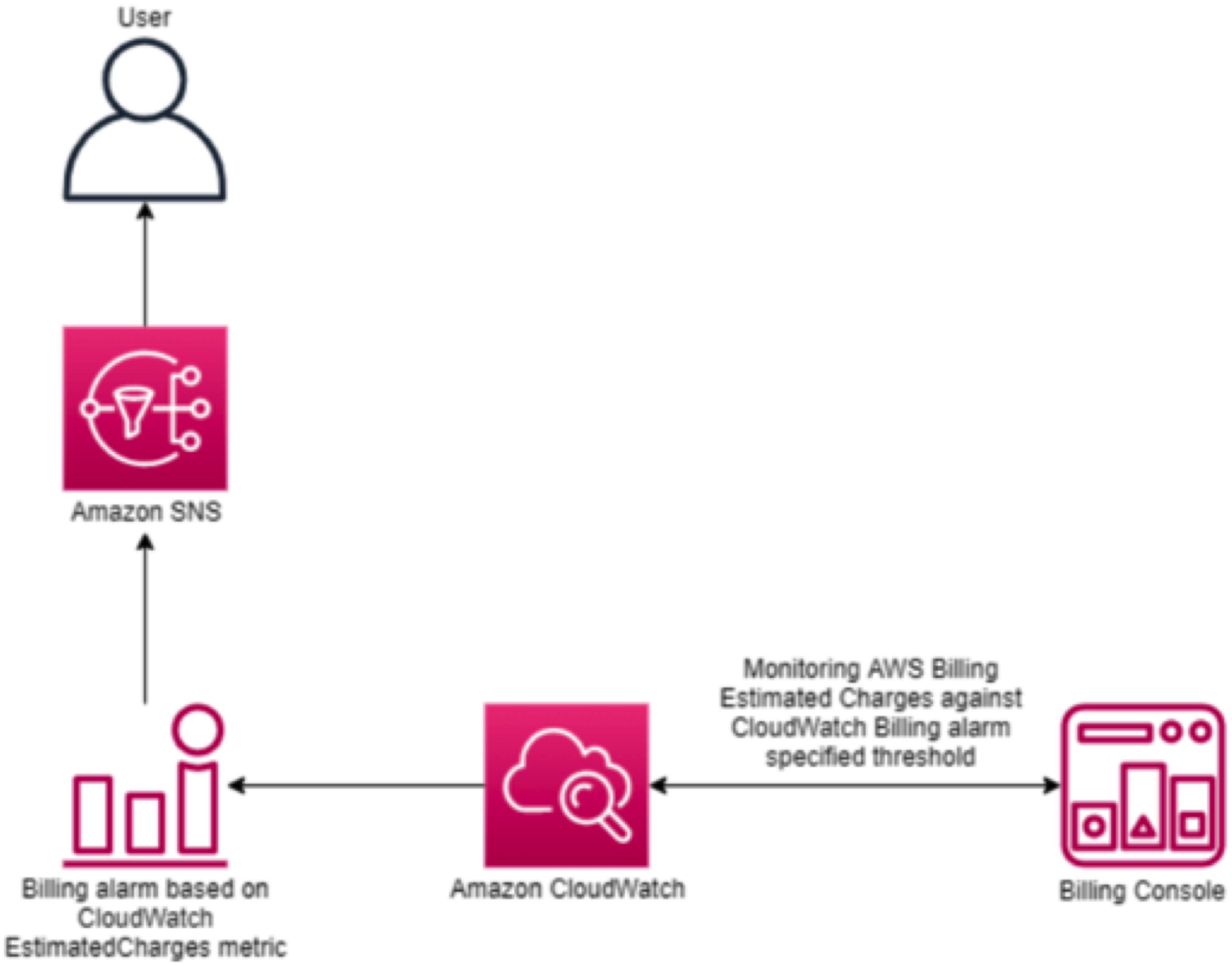
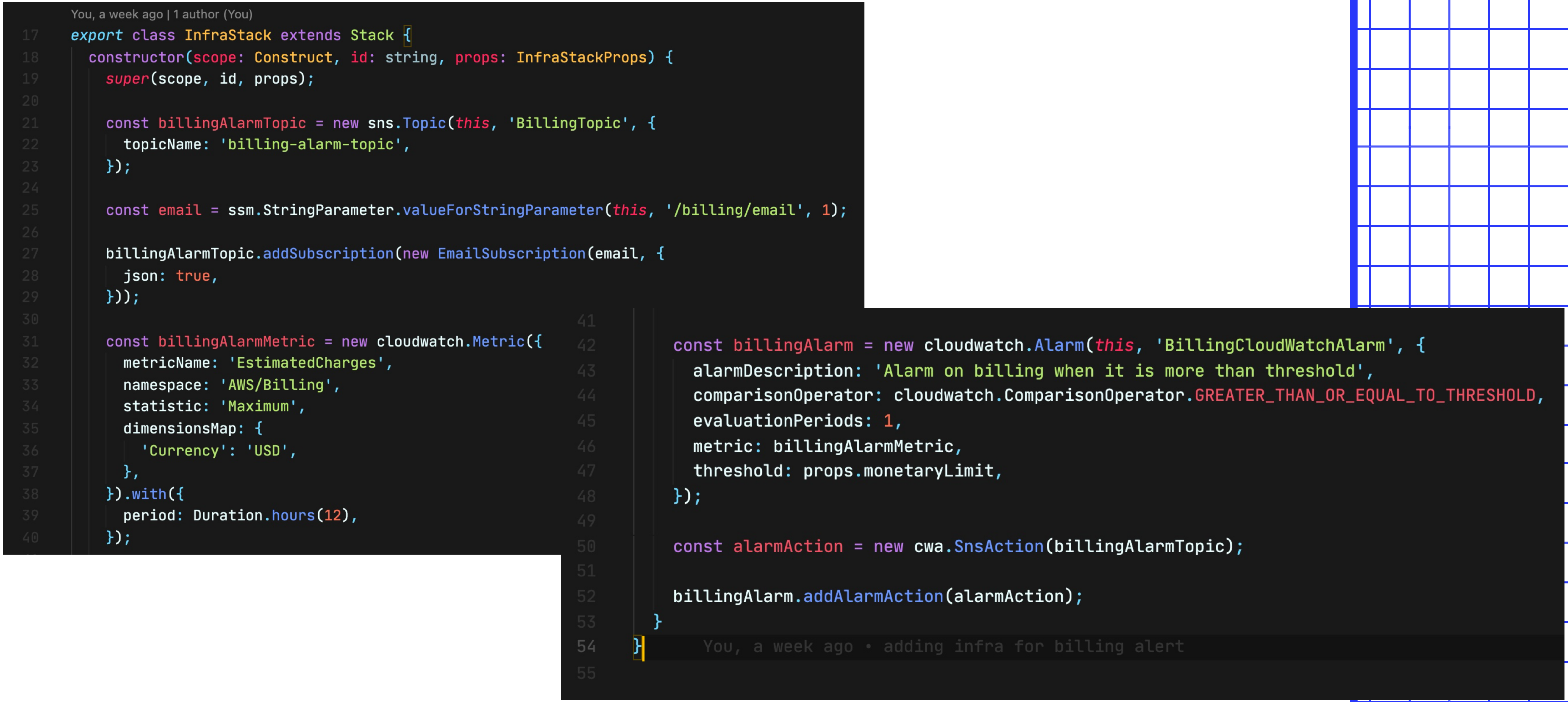

User
Amazon SNS
Monitoring AWS Billing Estimated Charges against CloudWatch Billing alarm specified threshold
Billing alarm based on CloudWatch EstimatedCharges metric
Amazon CloudWatch
Billing Console

Figure 1: CloudWatch Alarm Solution Overview

BILLING ALERT CDK EXAMPLE

2/2

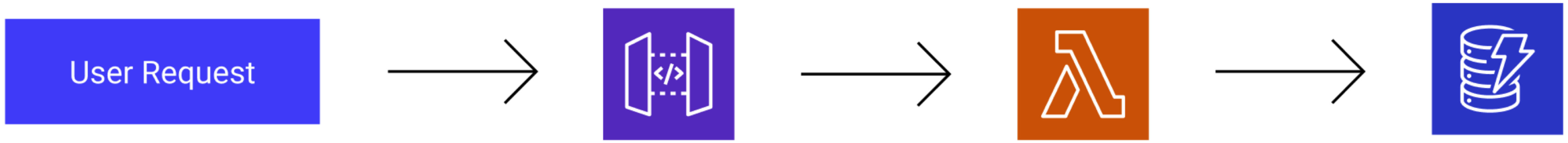


You, a week ago | 1 author (You)

```
export class InfraStack extends Stack {
  constructor(scope: Construct, id: string, props: InfraStackProps) {
    super(scope, id, props);

    const billingAlarmTopic = new sns.Topic(this, 'BillingTopic', {
      topicName: 'billing-alarm-topic',
    });

    const email = ssm.StringParameter.valueForStringParameter(this, '/billing/email', 1);

    billingAlarmTopic.addSubscription(new EmailSubscription(email, {
      json: true,
    }));

    const billingAlarmMetric = new cloudwatch.Metric({
      metricName: 'EstimatedCharges',
      namespace: 'AWS/Billing',
      statistic: 'Maximum',
      dimensionsMap: {
        'Currency': 'USD',
      },
    }).with({
      period: Duration.hours(12),
    });

    const billingAlarm = new cloudwatch.Alarm(this, 'BillingCloudWatchAlarm', {
      alarmDescription: 'Alarm on billing when it is more than threshold',
      comparisonOperator: cloudwatch.ComparisonOperator.GREATER_THAN_OR_EQUAL_TO_THRESHOLD,
      evaluationPeriods: 1,
      metric: billingAlarmMetric,
      threshold: props.monetaryLimit,
    });

    const alarmAction = new cwa.SnsAction(billingAlarmTopic);

    billingAlarm.addAlarmAction(alarmAction);
  }
}     You, a week ago • adding infra for billing alert

```

Source Code: https://github.com/Grenguar/cdk-billing-alert

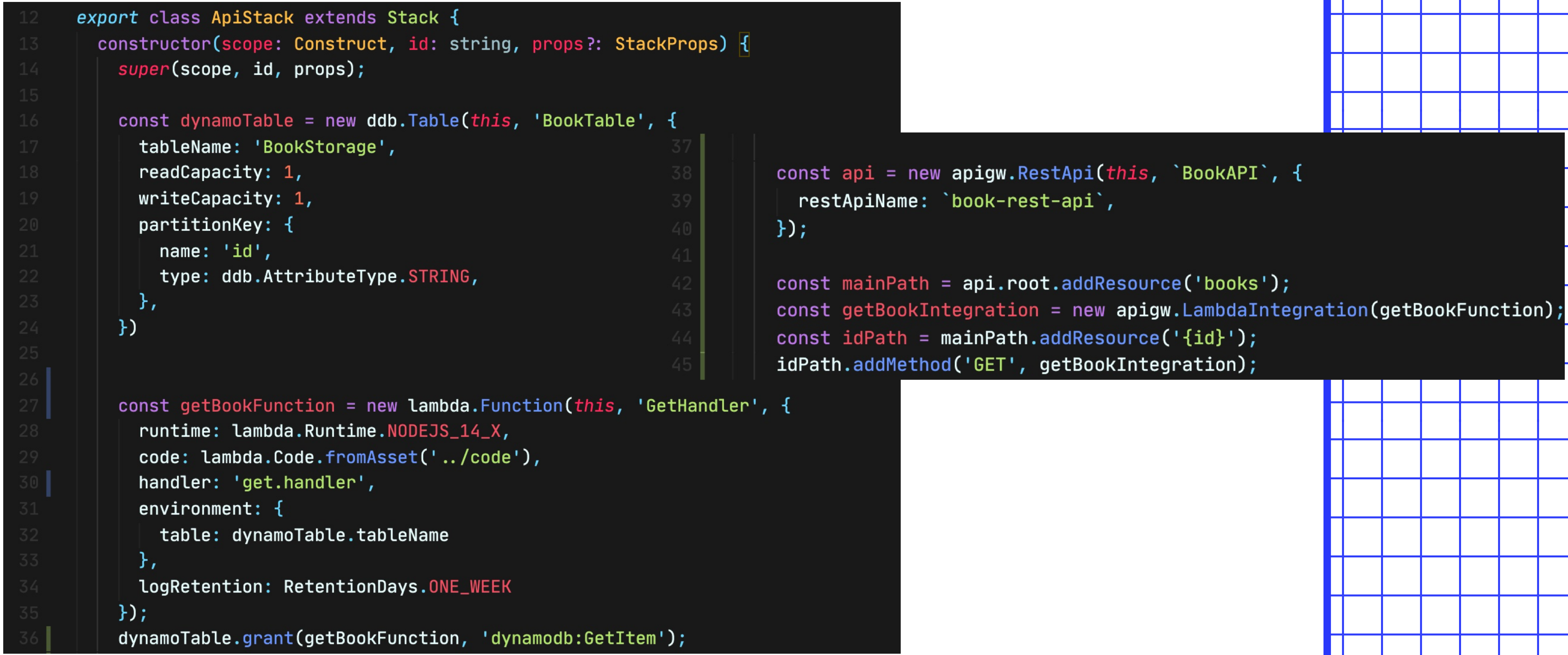
SERVERLESS API CDK EXAMPLE



User Request

SERVERLESS API CDK EXAMPLE

2/2



```
export class ApiStack extends Stack {
  constructor(scope: Construct, id: string, props?: StackProps) {
    super(scope, id, props);

    const dynamoTable = new ddb.Table(this, 'BookTable', {
      tableName: 'BookStorage',
      readCapacity: 1,
      writeCapacity: 1,
      partitionKey: {
        name: 'id',
        type: ddb.AttributeType.STRING,
      },
    })


    const getBookFunction = new lambda.Function(this, 'GetHandler', {
      runtime: lambda.Runtime.NODEJS_14_X,
      code: lambda.Code.fromAsset('../code'),
      handler: 'get.handler',
      environment: {
        table: dynamoTable.tableName
      },
      logRetention: RetentionDays.ONE_WEEK
    });
    dynamoTable.grant(getBookFunction, 'dynamodb:GetItem');

    const api = new apigw.RestApi(this, `BookAPI`, {
      restApiName: `book-rest-api`,
    });

    const mainPath = api.root.addResource('books');
    const getBookIntegration = new apigw.LambdaIntegration(getBookFunction);
    const idPath = mainPath.addResource('{id}');
    idPath.addMethod('GET', getBookIntegration);
```

Source Code: https://github.com/Grenguar/aws-cdk-api-workshop

ADVANTAGES OF CDK WITH SERVERLESS

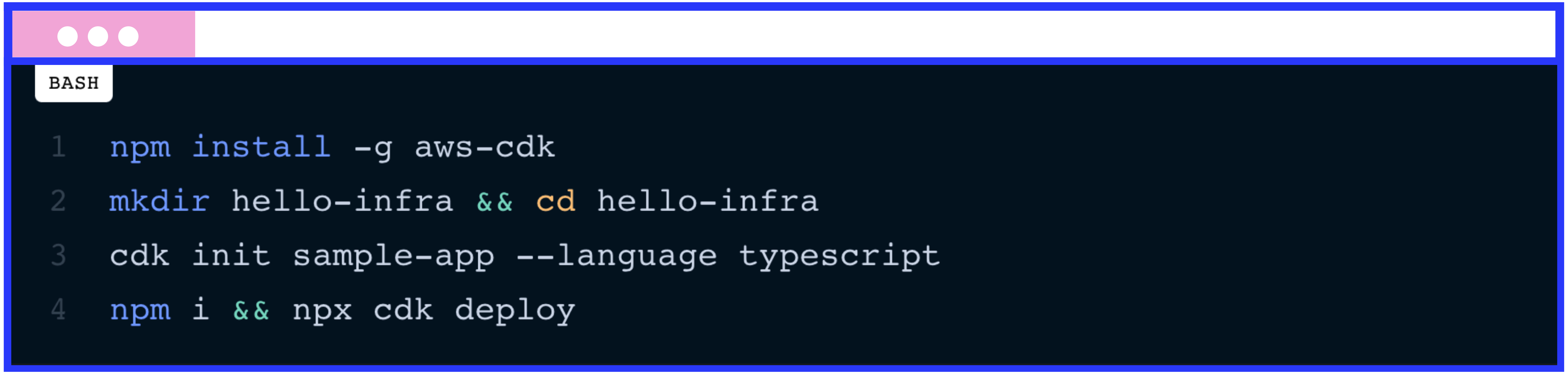
- CDK Pipelines
- Ready-made typed constructs
- Internal reusable components
- Permissions and linking resources

LESSONS LEARNED

How to start adopting CDK?

- Small project (event + action)
- Gradual migration (CDK + SAM for one deployment)
- Experiment and share knowledge

WRITE, BUILD, DEPLOY



```bash
npm install -g aws-cdk
mkdir hello-infra && cd hello-infra
cdk init sample-app --language typescript
npm i && npx cdk deploy
```

THANK YOU FOR YOU ATTENTION

ご清聴ありがとうございました。

TWITTER

@grenguar

LINKEDIN

Igor Soroka

E-MAIL

igor@soroka.tech