Build a serverless API in AWS in minutes:

Boosting your stack with Chalice and CDK

Juan David Alzate Cardona

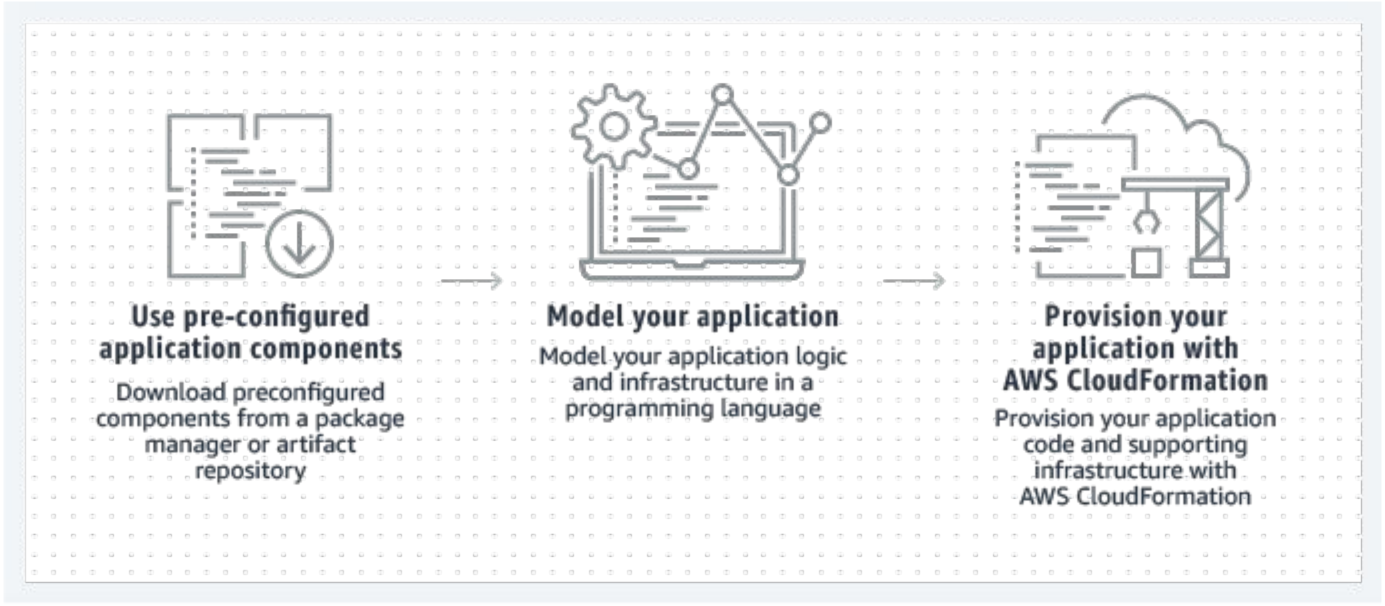
Facts:

- Physics Engineer.
- M.Sc. Physics.
- Reading lover.
- Teacher by passion.
- Developer by coincidence.
- A human being (most of the time).

Content

What is CDK?

$$\begin{tikzcd}\boxed{\text{CDK}} \arrow[r] & \boxed{\text{AWS Cloud Development Kit}}\end{tikzcd}$$



Use pre-configured application components

Download preconfigured components from a package manager or artifact repository

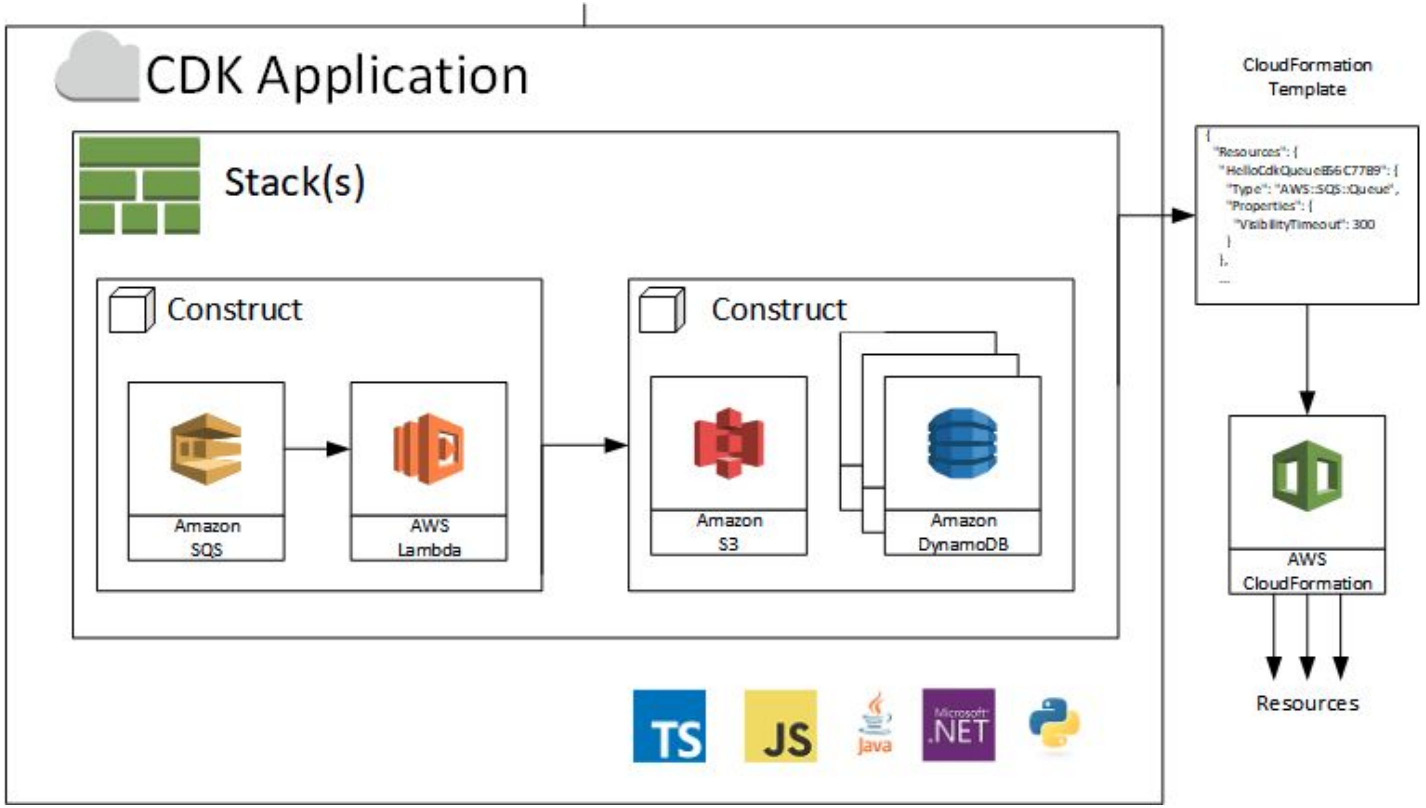
Model your application

Model your application logic and infrastructure in a programming language

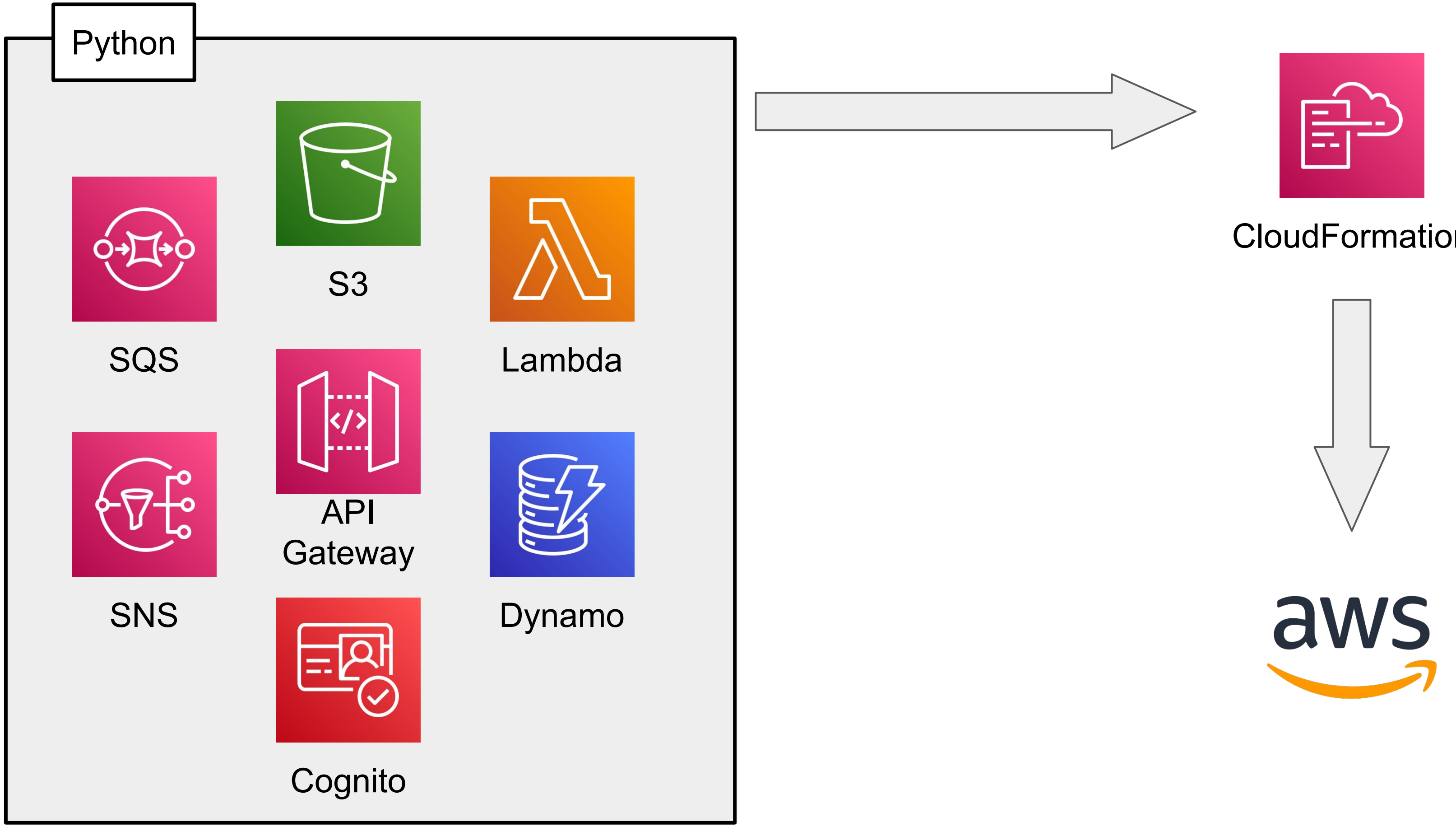
Provision your application with AWS CloudFormation

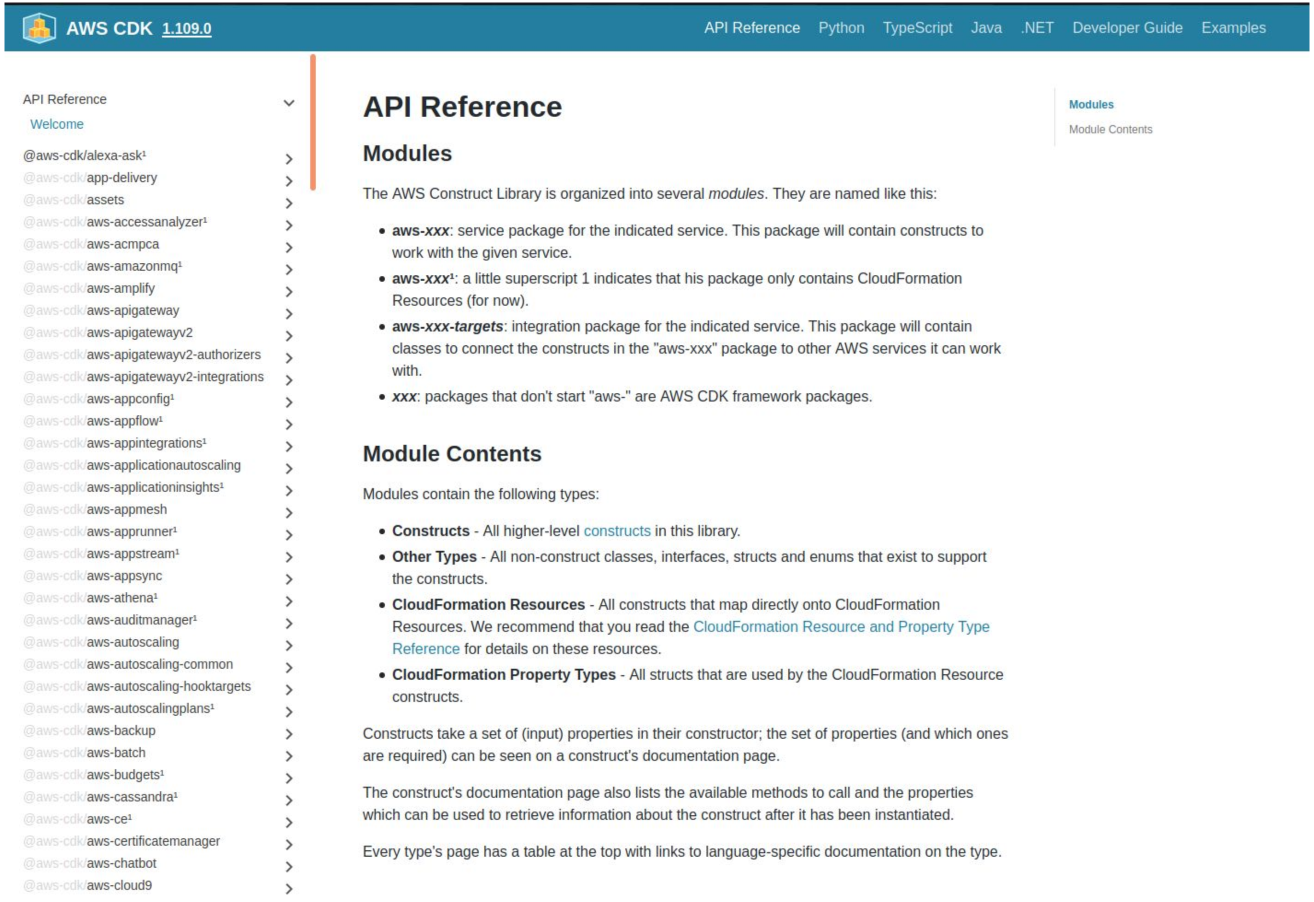
Provision your application code and supporting infrastructure with AWS CloudFormation

What is CDK for?



```
{
  "Resources": {
   "HelloCdkQueue856C7789": {
    "Type": "AWS::SQS::Queue",
    "Properties": {
     "VisibilityTimeout": 300
    }
  },
  ...
```


Python
S3
SQS
Lambda
API Gateway
SNS
Dynamo
Cognito
CloudFormation
aws



API Reference

Modules

The AWS Construct Library is organized into several *modules*. They are named like this:

- **aws-*xxx***: service package for the indicated service. This package will contain constructs to work with the given service.
- **aws-*xxx*¹**: 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.

Module Contents

Modules contain 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.

Constructs take a set of (input) properties in their constructor; the set of properties (and which ones are required) can be seen on a construct's documentation page.

The construct's documentation page also lists the available methods to call and the properties which can be used to retrieve information about the construct after it has been instantiated.

Every type's page has a table at the top with links to language-specific documentation on the type.

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

Serverless (AWS)

Compute

AWS Lambda

Containers

AWS Fargate

Application Integration

Amazon SQS

Amazon SNS

Amazon EventBridge

AWS Step Functions

AWS AppSync

Storage

Amazon S3

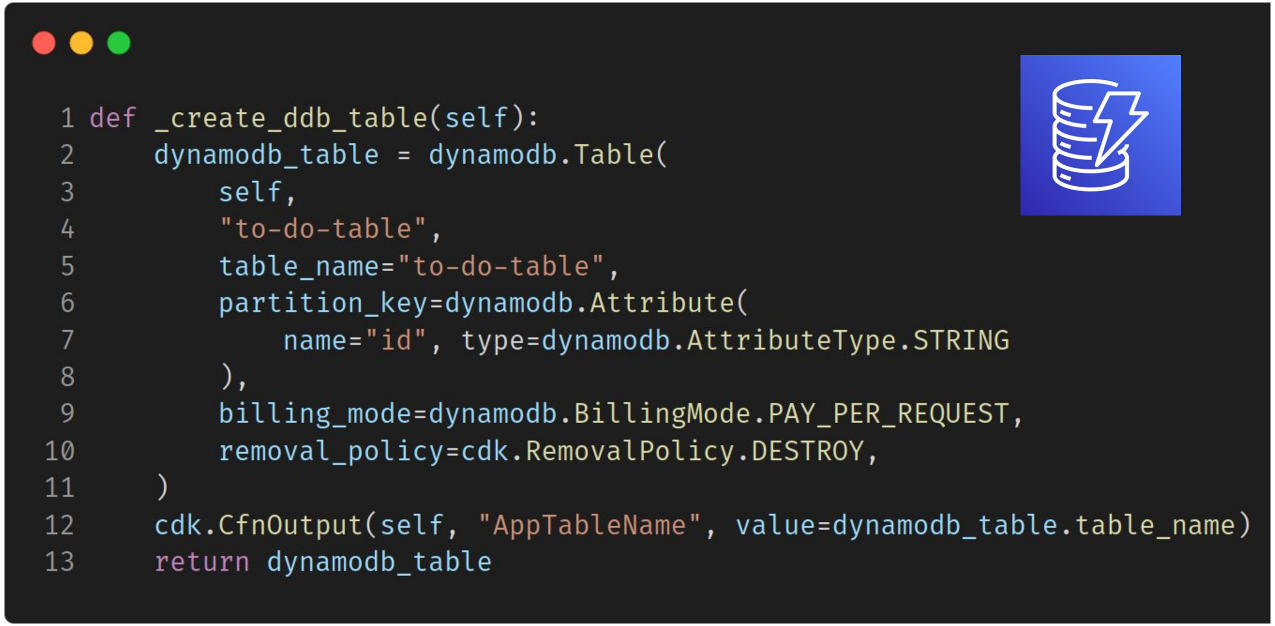
Database

Amazon DynamoDB

Amazon RDS Proxy

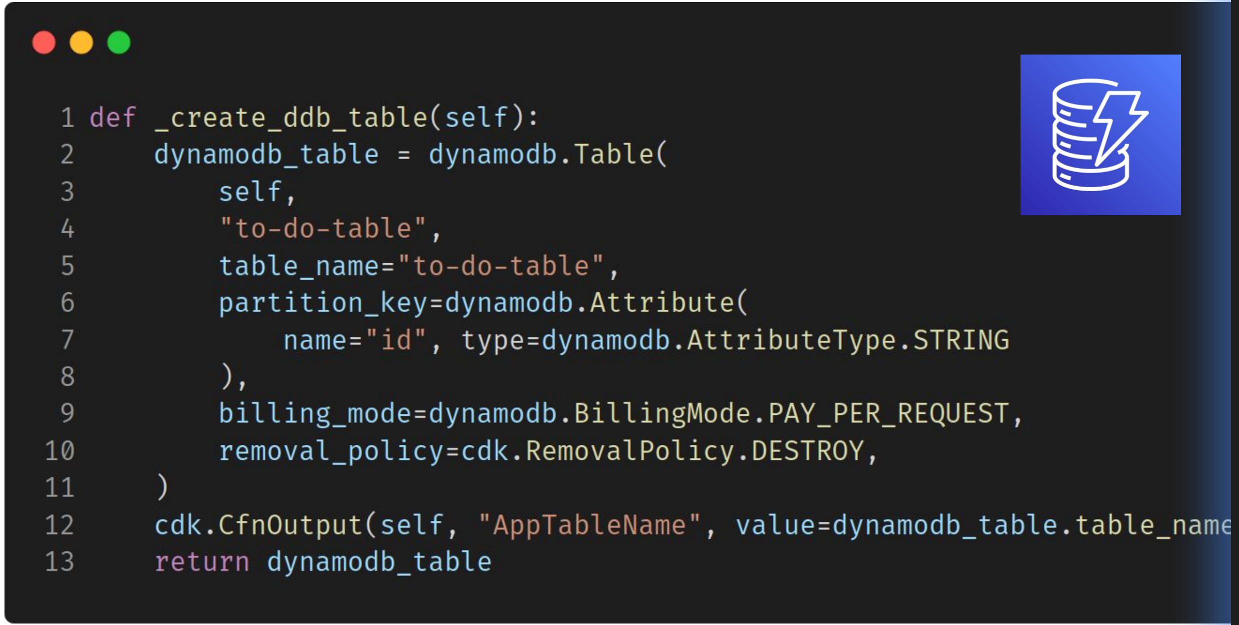
Amazon Aurora Serverless

How does CDK work?

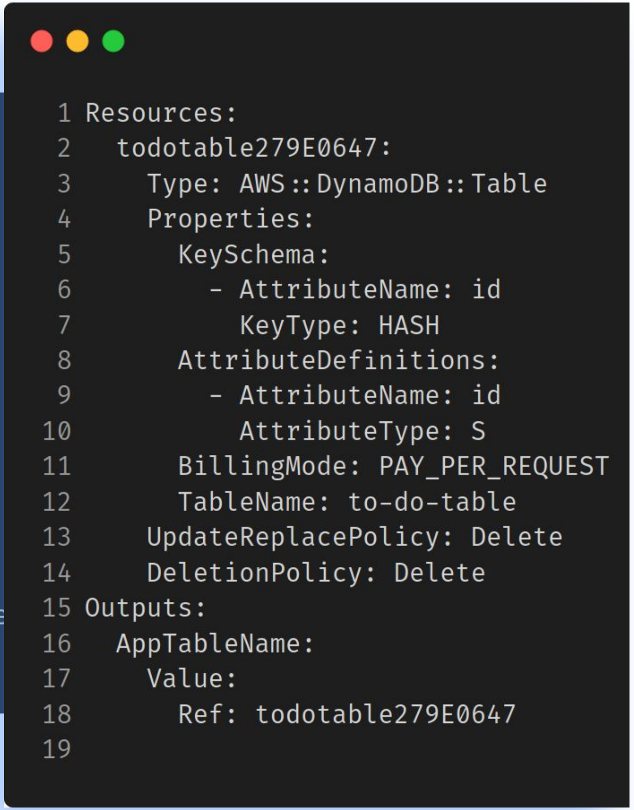


```python
def _create_ddb_table(self):
    dynamodb_table = dynamodb.Table(
        self,
        "to-do-table",
        table_name="to-do-table",
        partition_key=dynamodb.Attribute(
            name="id", type=dynamodb.AttributeType.STRING
        ),
        billing_mode=dynamodb.BillingMode.PAY_PER_REQUEST,
        removal_policy=cdk.RemovalPolicy.DESTROY,
    )
    cdk.CfnOutput(self, "AppTableName", value=dynamodb_table.table_name)
    return dynamodb_table
```

How does CDK work?



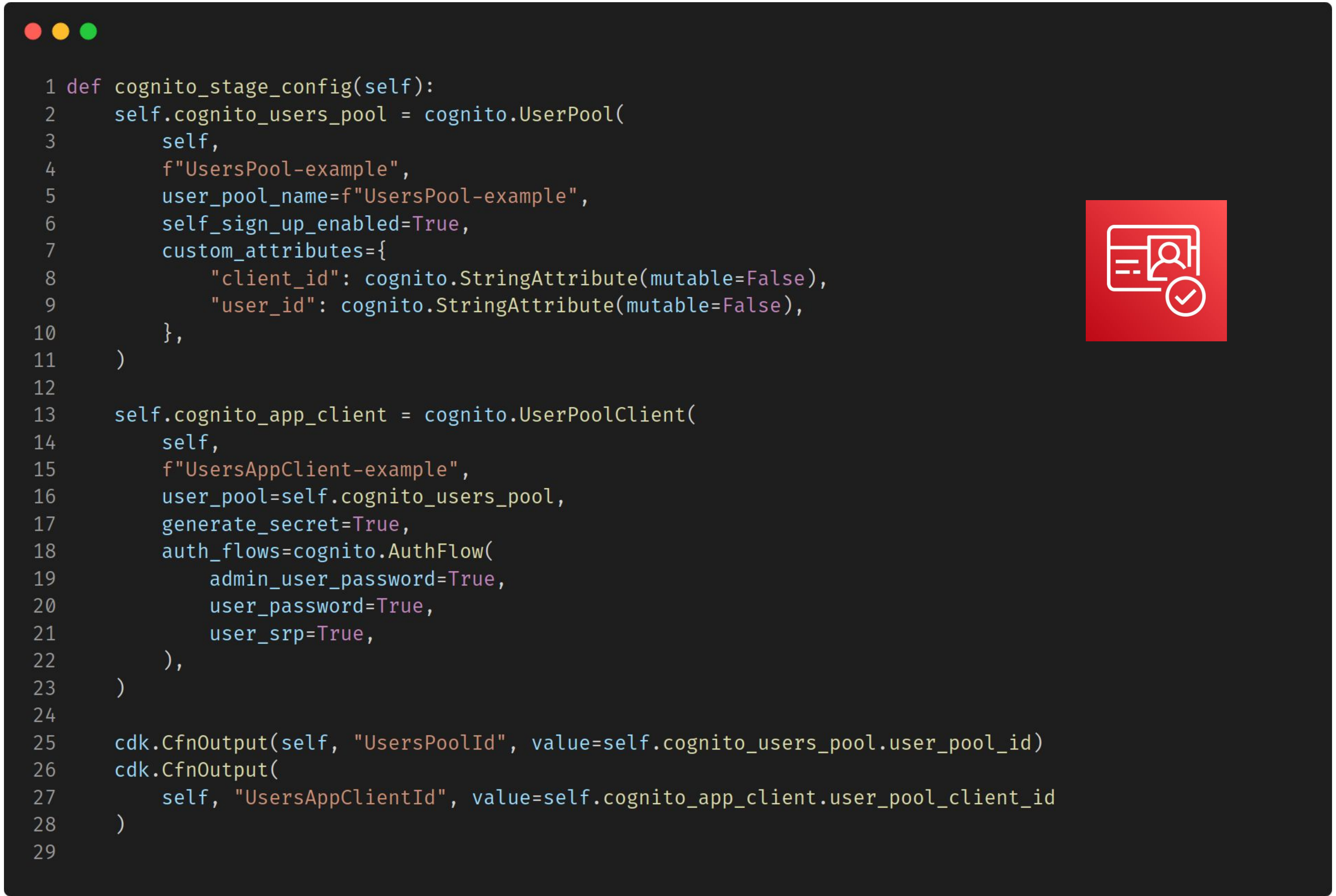
```python
def _create_ddb_table(self):
    dynamodb_table = dynamodb.Table(
        self,
        "to-do-table",
        table_name="to-do-table",
        partition_key=dynamodb.Attribute(
            name="id", type=dynamodb.AttributeType.STRING
        ),
        billing_mode=dynamodb.BillingMode.PAY_PER_REQUEST,
        removal_policy=cdk.RemovalPolicy.DESTROY,
    )
    cdk.CfnOutput(self, "AppTableName", value=dynamodb_table.table_name
    return dynamodb_table
```



```yaml
Resources:
  todotable279E0647:
    Type: AWS::DynamoDB::Table
    Properties:
      KeySchema:
        - AttributeName: id
          KeyType: HASH
      AttributeDefinitions:
        - AttributeName: id
          AttributeType: S
      BillingMode: PAY_PER_REQUEST
      TableName: to-do-table
    UpdateReplacePolicy: Delete
    DeletionPolicy: Delete
Outputs:
  AppTableName:
    Value:
      Ref: todotable279E0647

```

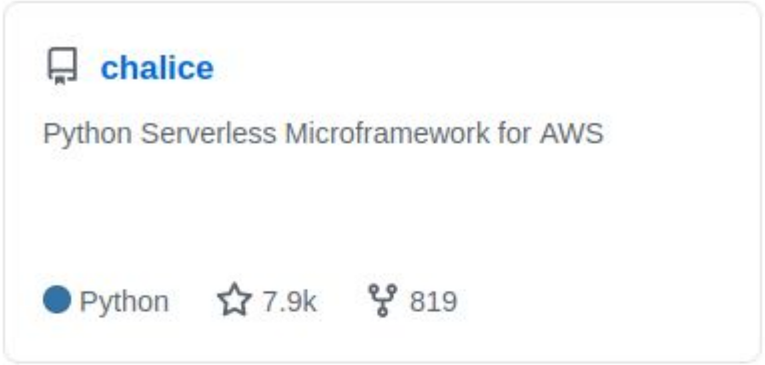
Build a serverless API in AWS in minutes

June 19, 2021



```python
def cognito_stage_config(self):
    self.cognito_users_pool = cognito.UserPool(
        self,
        f"UsersPool-example",
        user_pool_name=f"UsersPool-example",
        self_sign_up_enabled=True,
        custom_attributes={
            "client_id": cognito.StringAttribute(mutable=False),
            "user_id": cognito.StringAttribute(mutable=False),
        },
    )

    self.cognito_app_client = cognito.UserPoolClient(
        self,
        f"UsersAppClient-example",
        user_pool=self.cognito_users_pool,
        generate_secret=True,
        auth_flows=cognito.AuthFlow(
            admin_user_password=True,
            user_password=True,
            user_srp=True,
        ),
    )

    cdk.CfnOutput(self, "UsersPoolId", value=self.cognito_users_pool.user_pool_id)
    cdk.CfnOutput(
        self, "UsersAppClientId", value=self.cognito_app_client.user_pool_client_id
    )

```

What is Chalice?


chalice

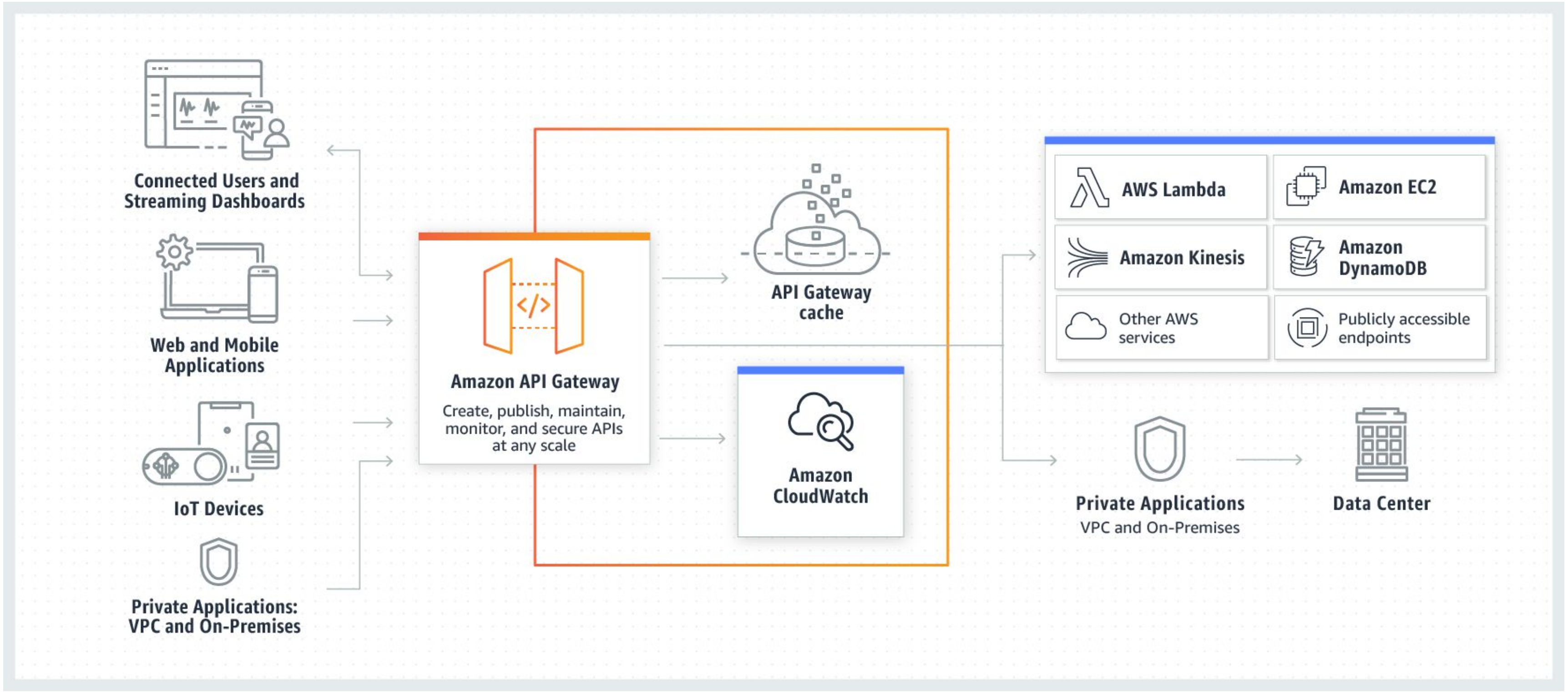
Python Serverless Microframework for AWS

Python 7.9k 819

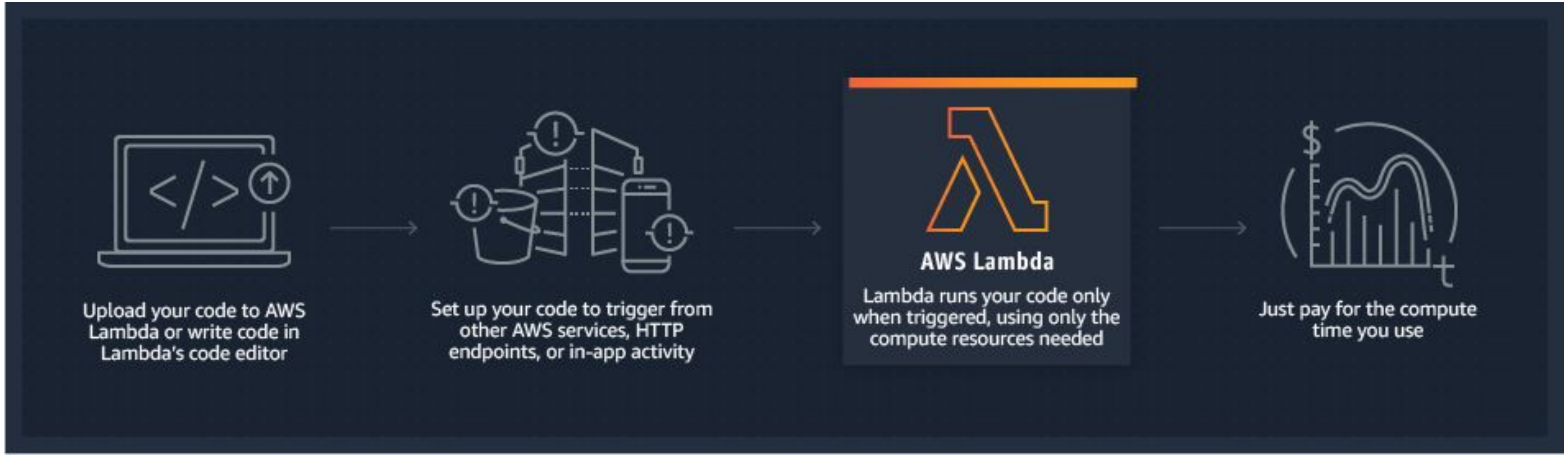
AWS Chalice allows you to quickly create and deploy applications that use Amazon API Gateway and AWS Lambda. It provides:

- A command line tool for creating, deploying, and managing your app
- A familiar and easy to use API for declaring views in python code
- Automatic IAM policy generation

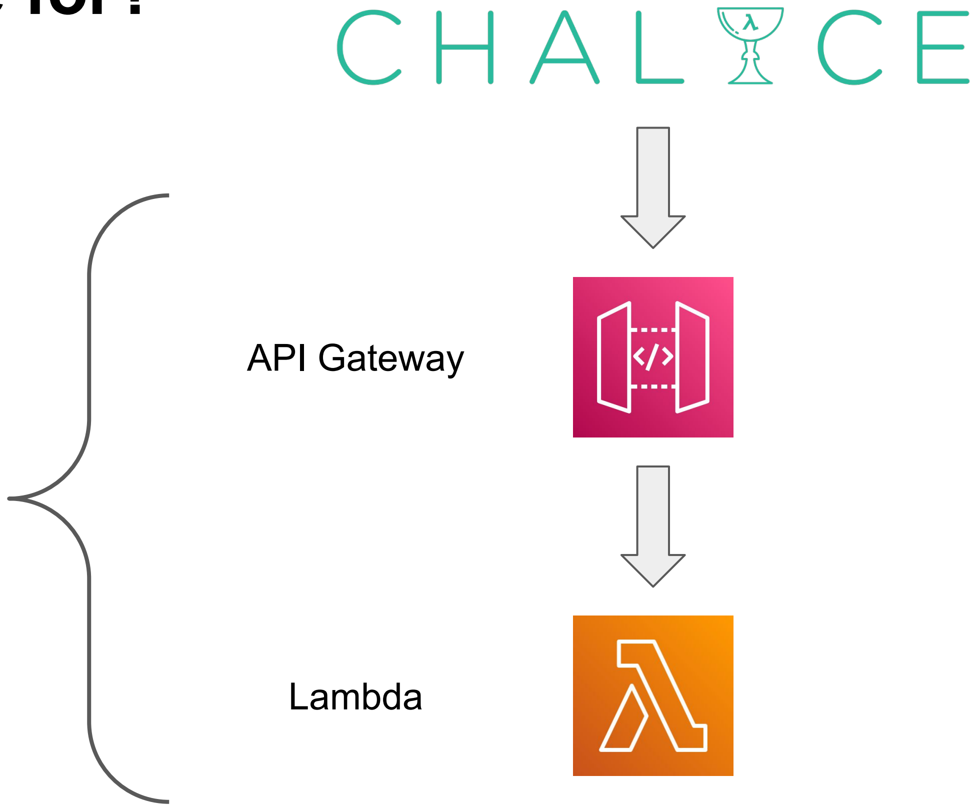
API gateway


Connected Users and Streaming Dashboards
Web and Mobile Applications
IoT Devices
Private Applications: VPC and On-Premises
Amazon API Gateway
Create, publish, maintain, monitor, and secure APIs at any scale
API Gateway cache
Amazon CloudWatch
AWS Lambda
Amazon EC2
Amazon Kinesis
Amazon DynamoDB
Other AWS services
Publicly accessible endpoints
Private Applications
VPC and On-Premises
Data Center

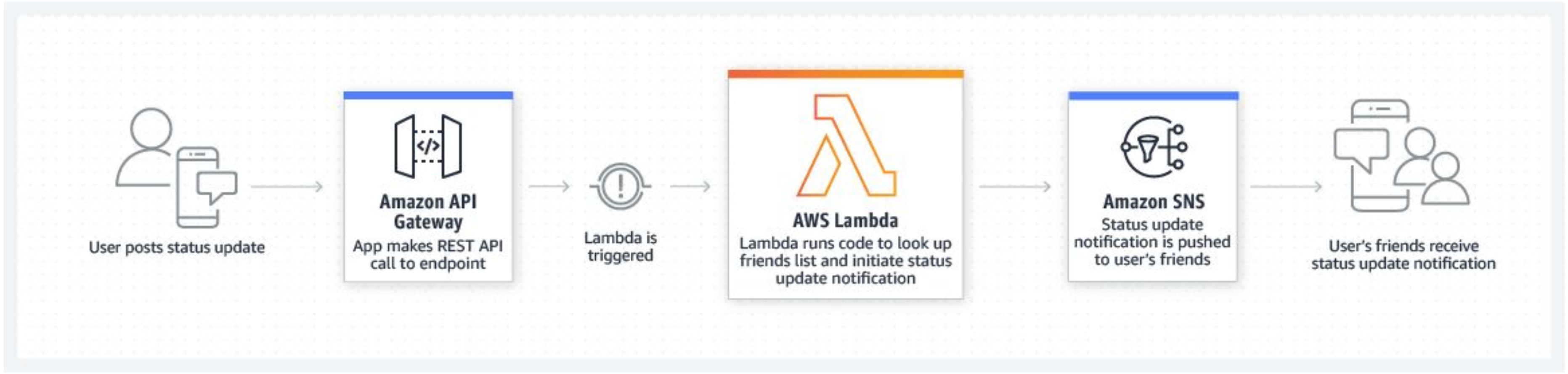
Lambda


Upload your code to AWS Lambda or write code in Lambda's code editor
Set up your code to trigger from other AWS services, HTTP endpoints, or in-app activity
AWS Lambda
Lambda runs your code only when triggered, using only the compute resources needed
Just pay for the compute time you use

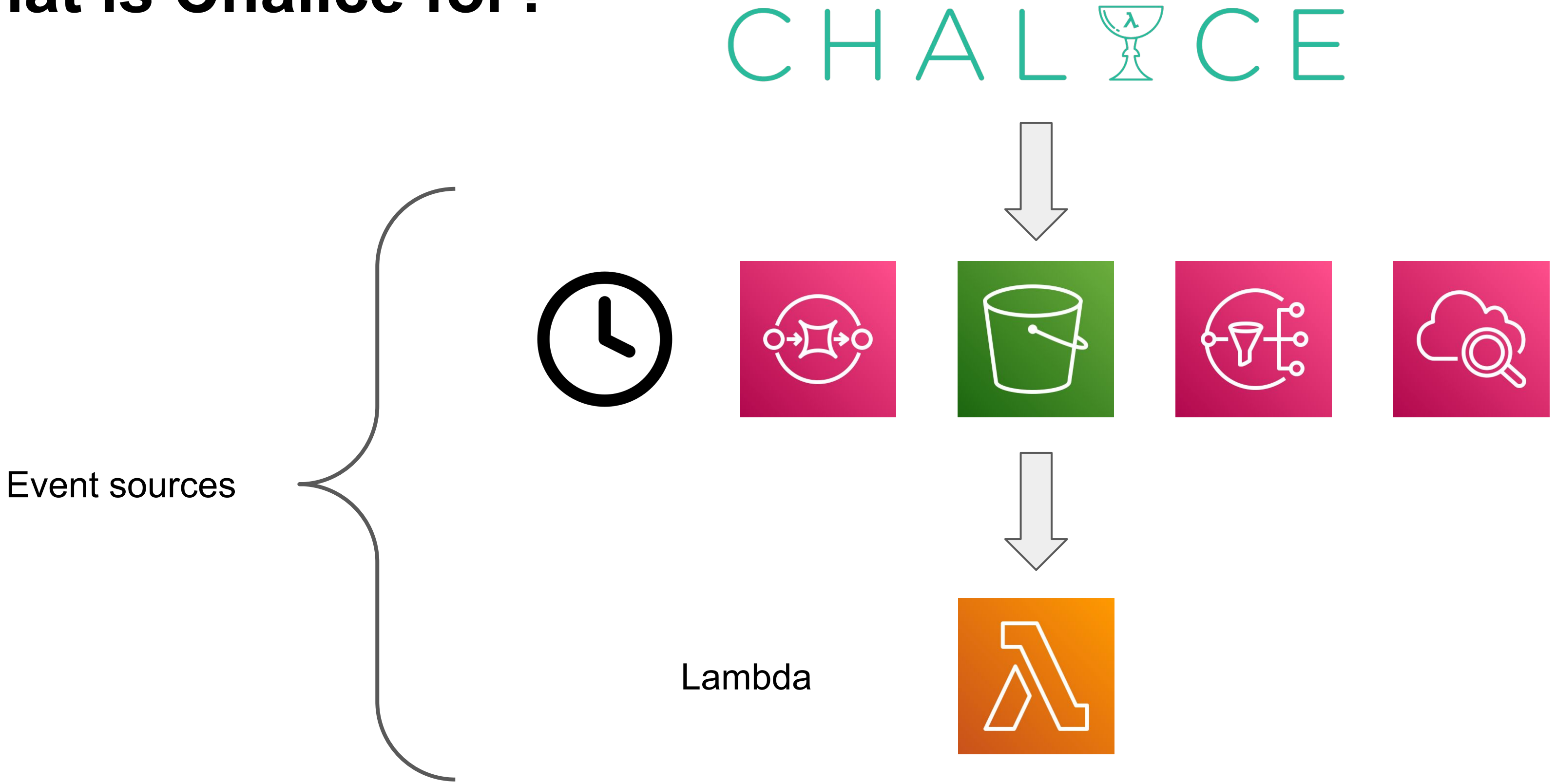
What is Chalice for?



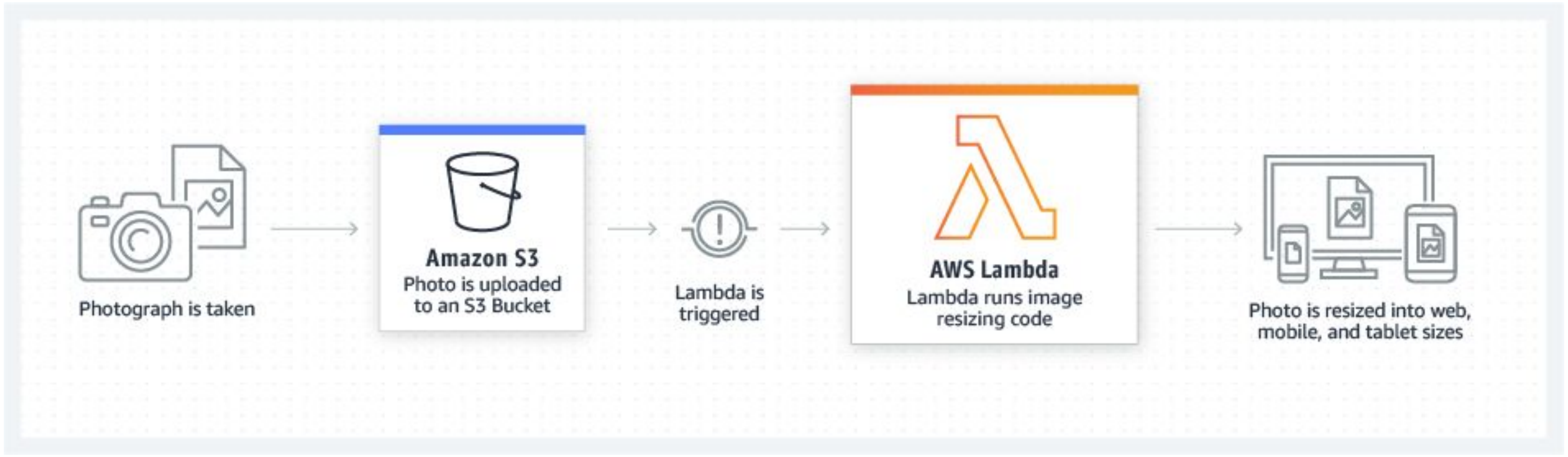
What is Chalice for?


User posts status update
Amazon API Gateway
App makes REST API call to endpoint
Lambda is triggered
AWS Lambda
Lambda runs code to look up friends list and initiate status update notification
Amazon SNS
Status update notification is pushed to user's friends
User's friends receive status update notification

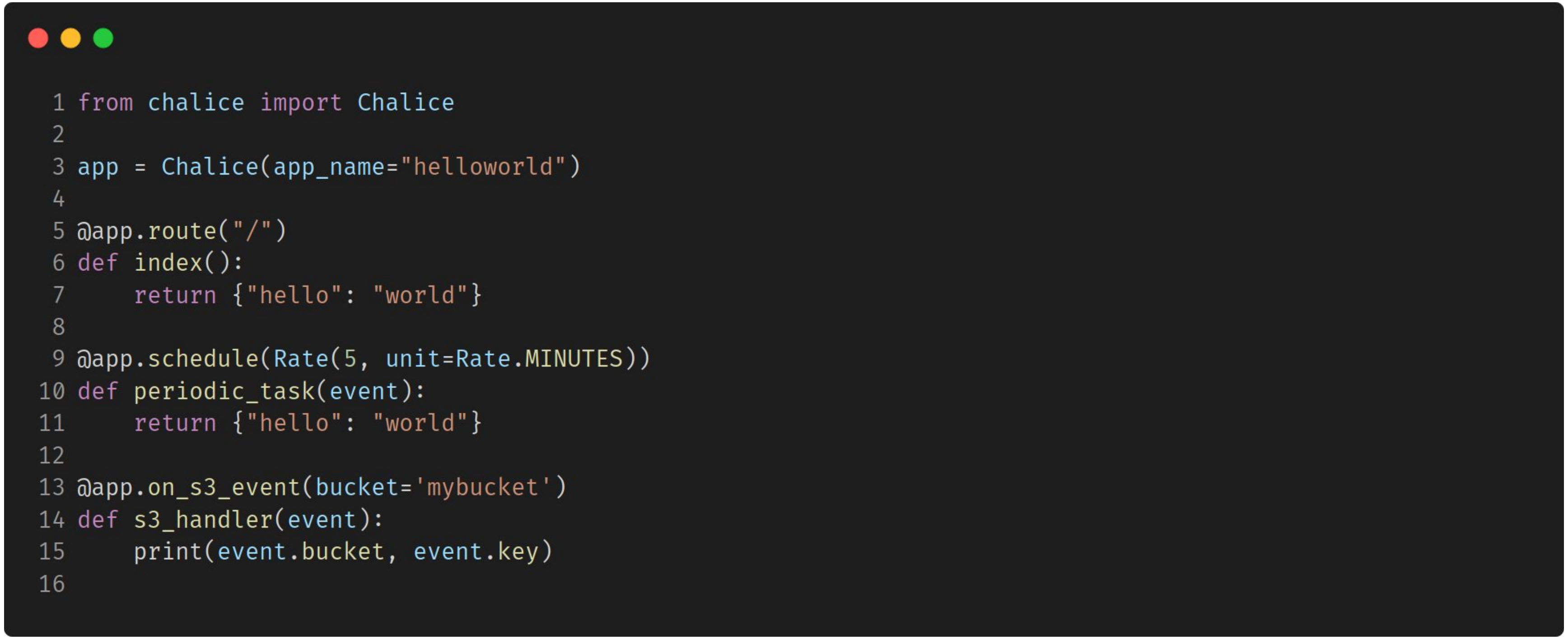
What is Chalice for?



What is Chalice for?

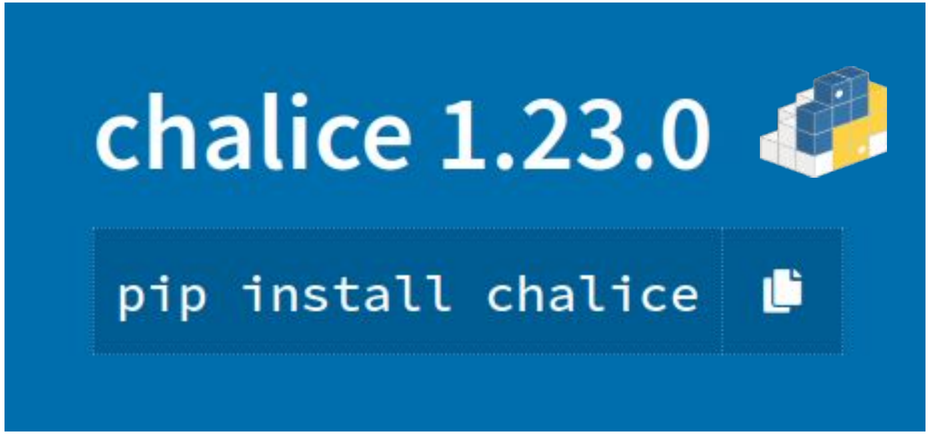

Photograph is taken
Amazon S3
Photo is uploaded to an S3 Bucket
Lambda is triggered
AWS Lambda
Lambda runs image resizing code
Photo is resized into web, mobile, and tablet sizes

How does Chalice work?



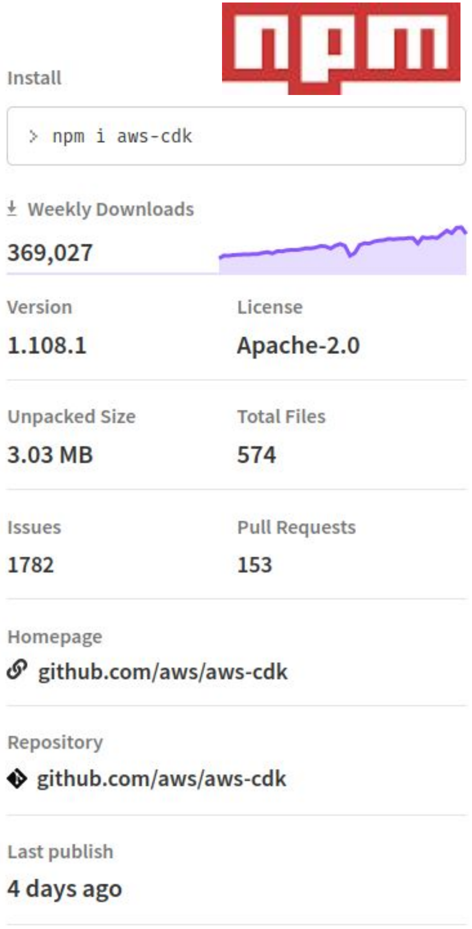
```python
from chalice import Chalice

app = Chalice(app_name="helloworld")

@app.route("/")
def index():
    return {"hello": "world"}

@app.schedule(Rate(5, unit=Rate.MINUTES))
def periodic_task(event):
    return {"hello": "world"}

@app.on_s3_event(bucket='mybucket')
def s3_handler(event):
    print(event.bucket, event.key)

```

Dependencies



chalice 1.23.0

```
pip install chalice
```



Install

```
> npm i aws-cdk
```

Weekly Downloads

369,027

Version	License
1.108.1	Apache-2.0
Unpacked Size	Total Files
3.03 MB	574
Issues	Pull Requests
1782	153

Homepage

github.com/aws/aws-cdk

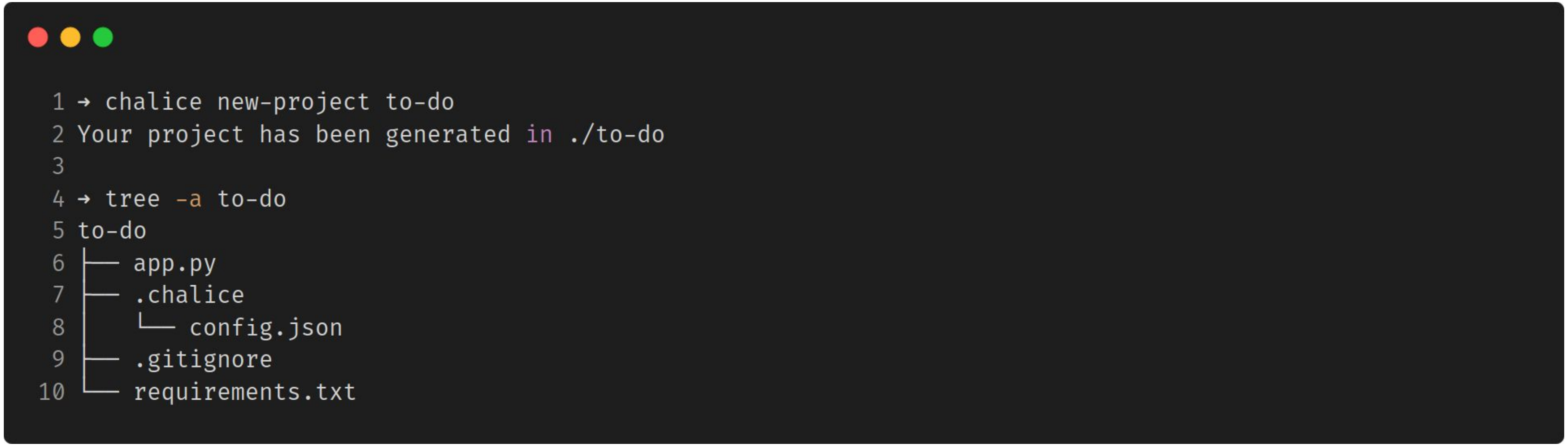
Repository

github.com/aws/aws-cdk

Last publish

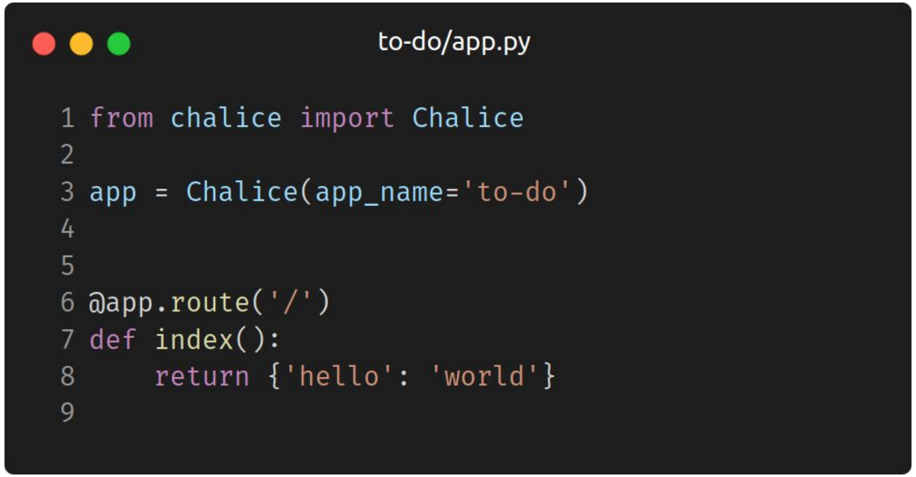
4 days ago

Our first app



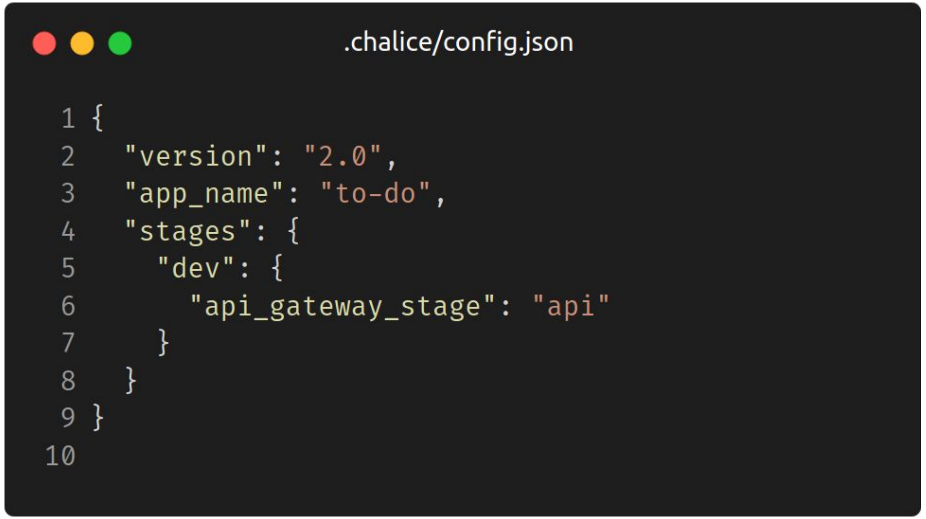
```
→ chalice new-project to-do
Your project has been generated in ./to-do

→ tree -a to-do
to-do
├── app.py
├── .chalice
│   └── config.json
├── .gitignore
└── requirements.txt
```

Our first app



to-do/app.py

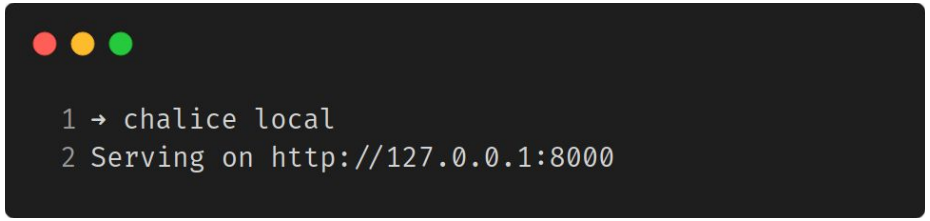
```python
from chalice import Chalice

app = Chalice(app_name='to-do')


@app.route('/')
def index():
    return {'hello': 'world'}

```



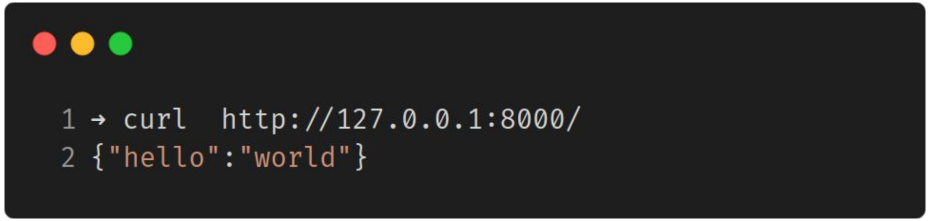
.chalice/config.json

```json
{
  "version": "2.0",
  "app_name": "to-do",
  "stages": {
    "dev": {
      "api_gateway_stage": "api"
    }
  }
}

```

Our first app

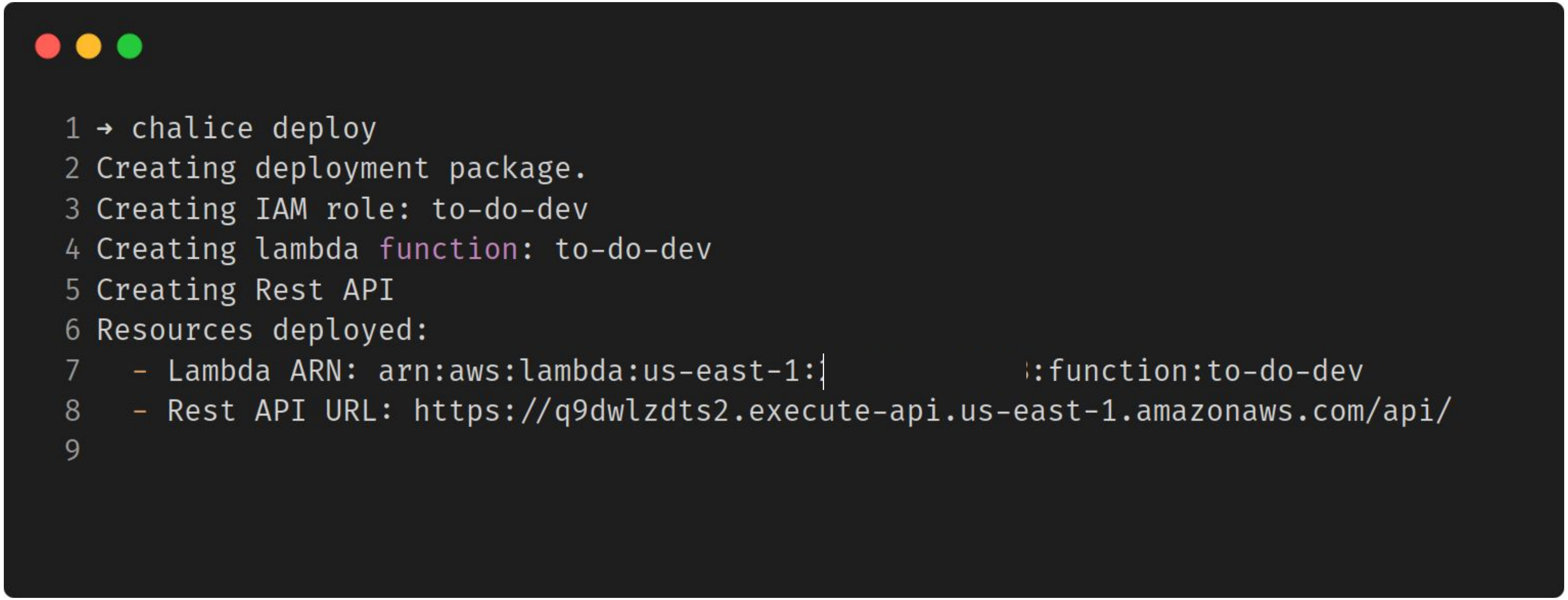


```
→ chalice local
Serving on http://127.0.0.1:8000
```



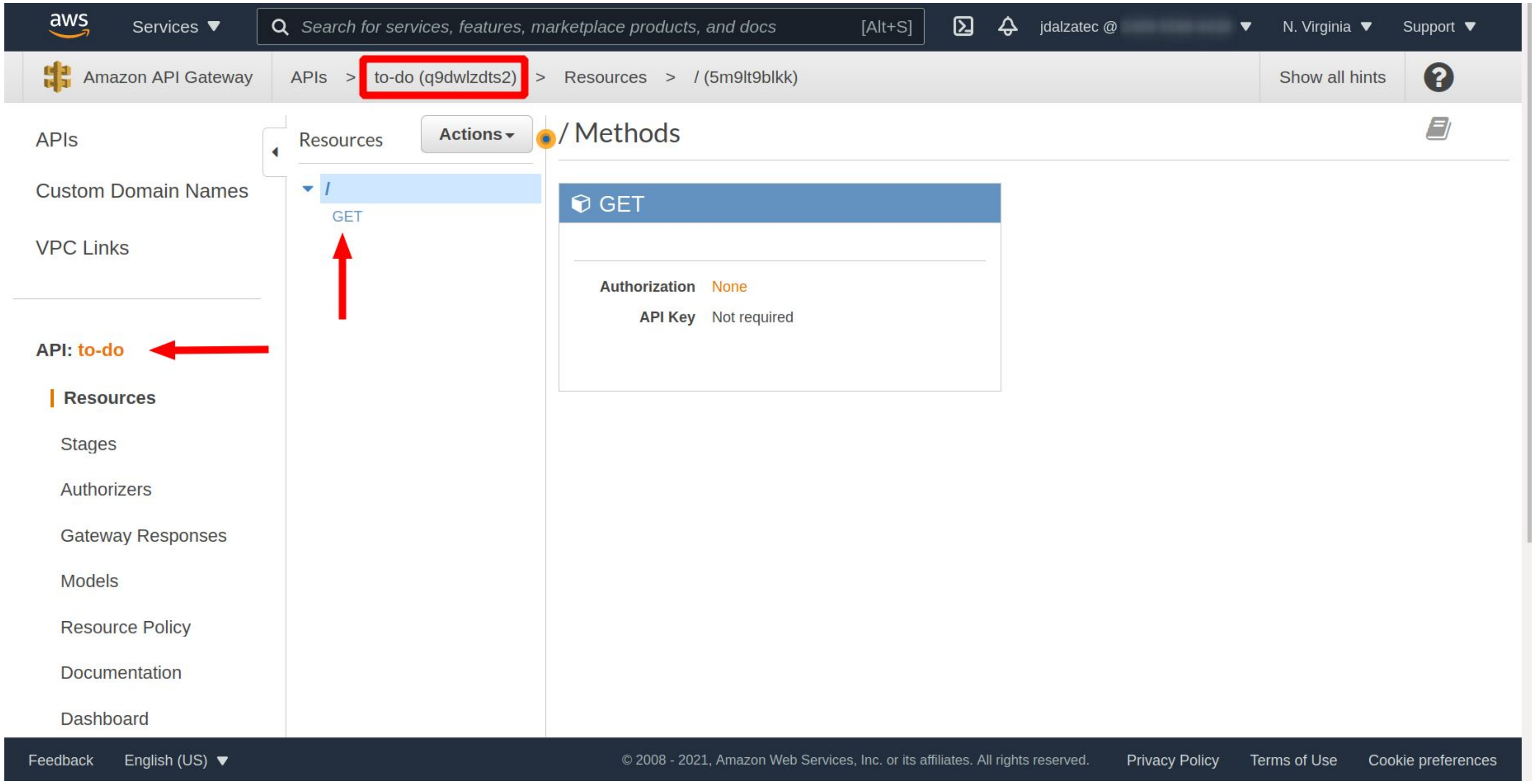
```
→ curl  http://127.0.0.1:8000/
{"hello":"world"}
```

Our first app



```
→ chalice deploy
Creating deployment package.
Creating IAM role: to-do-dev
Creating lambda function: to-do-dev
Creating Rest API
Resources deployed:
  - Lambda ARN: arn:aws:lambda:us-east-1:[redacted]:function:to-do-dev
  - Rest API URL: https://q9dwlzdts2.execute-api.us-east-1.amazonaws.com/api/
```

Our first app



aws Services ▼ Search for services, features, marketplace products, and docs [Alt+S] jdalzatec @ [redacted] ▼ N. Virginia ▼ Support ▼

Amazon API Gateway APIs > to-do (q9dwlzdts2) > Resources > / (5m9lt9blkk) Show all hints

APIs

Custom Domain Names

VPC Links

API: to-do

Resources

Stages

Authorizers

Gateway Responses

Models

Resource Policy

Documentation

Dashboard

Resources Actions▾

/

GET

/ Methods

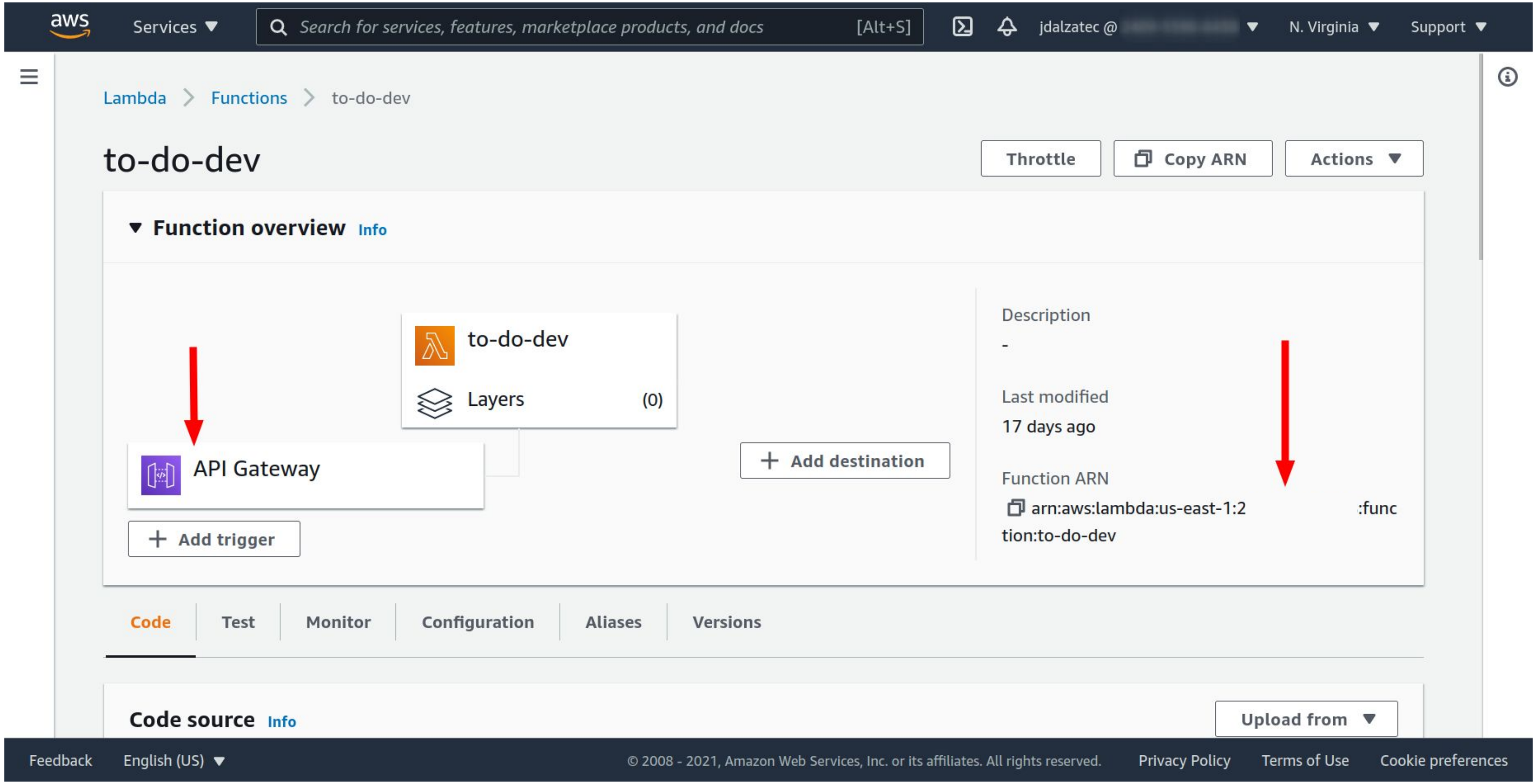
GET

Authorization None

API Key Not required

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

Our first app



Services ▼ Search for services, features, marketplace products, and docs [Alt+S]

jdalzatec @ [redacted] ▼ N. Virginia ▼ Support ▼

Lambda > Functions > to-do-dev

to-do-dev

Throttle | Copy ARN | Actions ▼

▼ Function overview Info

to-do-dev

Layers (0)

API Gateway

+ Add trigger

+ Add destination

Description

-

Last modified

17 days ago

Function ARN

arn:aws:lambda:us-east-1:2 :function:to-do-dev

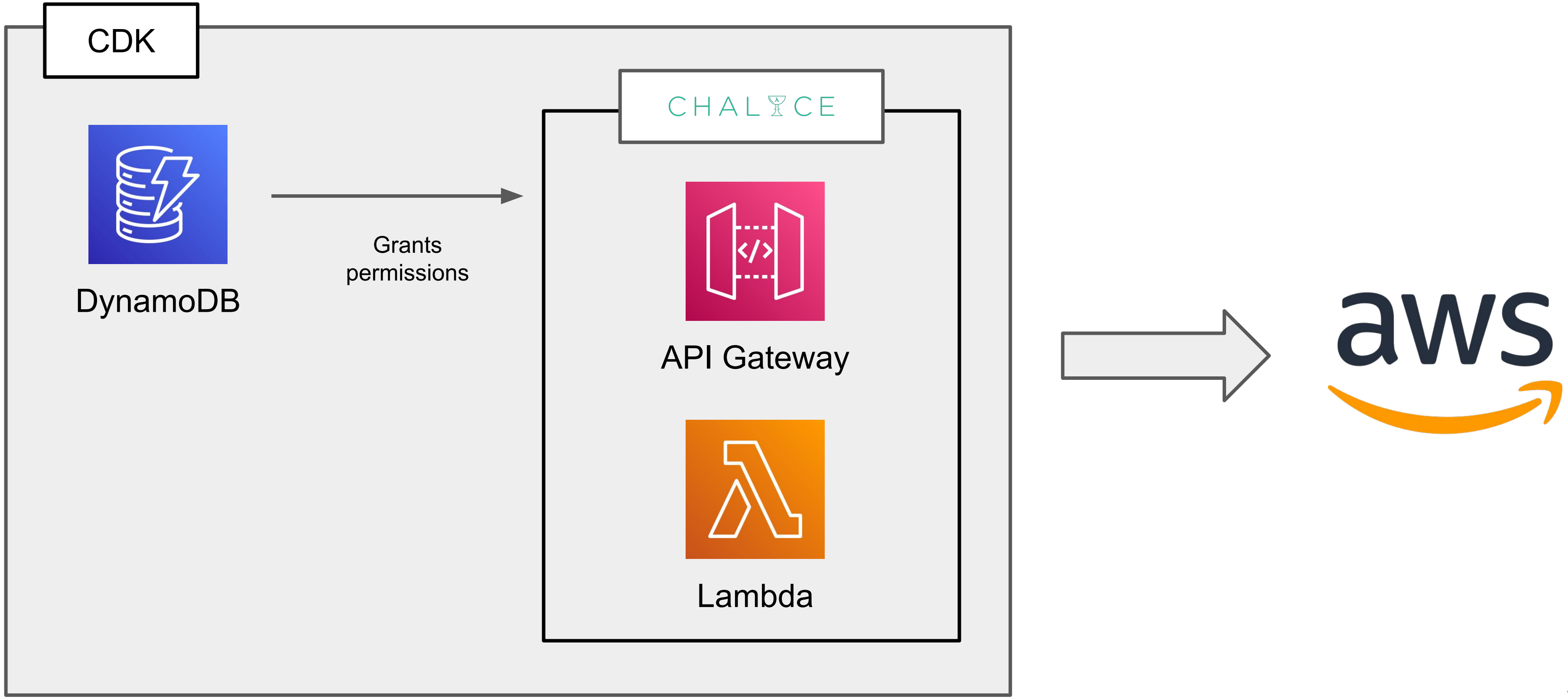
Code | Test | Monitor | Configuration | Aliases | Versions

Code source Info

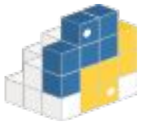
Upload from ▼

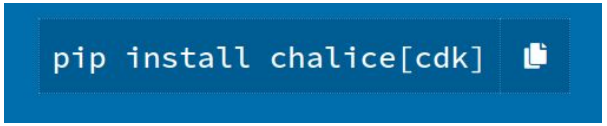
Feedback English (US) ▼ © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

CDK + Chalice


CDK
DynamoDB
Grants permissions
CHALICE
API Gateway
Lambda
aws

to-do app

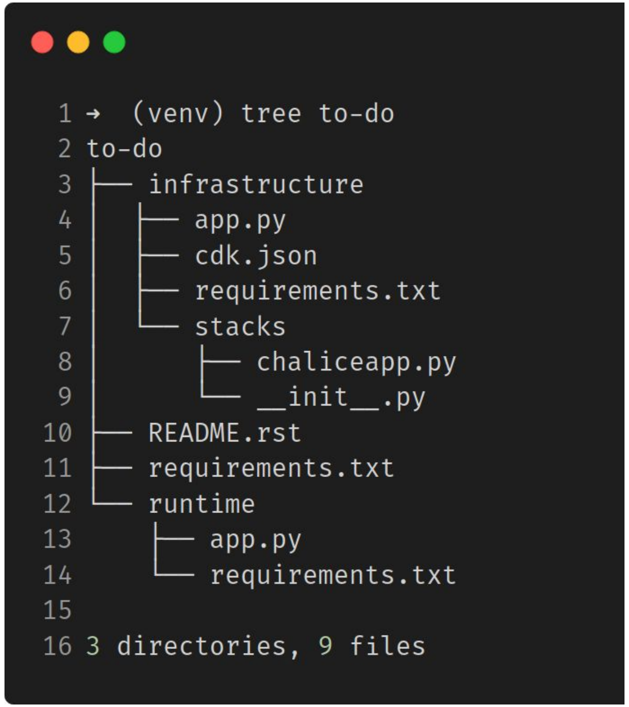




```
pip install chalice[cdk]
```

```
(venv) ➜ proj
```

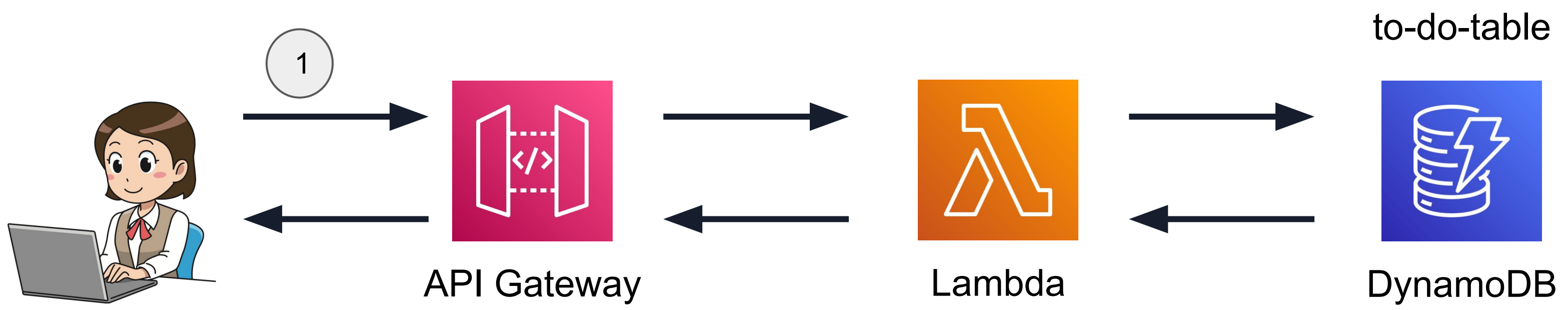
to-do app



```
→  (venv) tree to-do
to-do
├── infrastructure
│   ├── app.py
│   ├── cdk.json
│   ├── requirements.txt
│   └── stacks
│       ├── chaliceapp.py
│       └── __init__.py
├── README.rst
├── requirements.txt
└── runtime
    ├── app.py
    └── requirements.txt

3 directories, 9 files
```

Full example

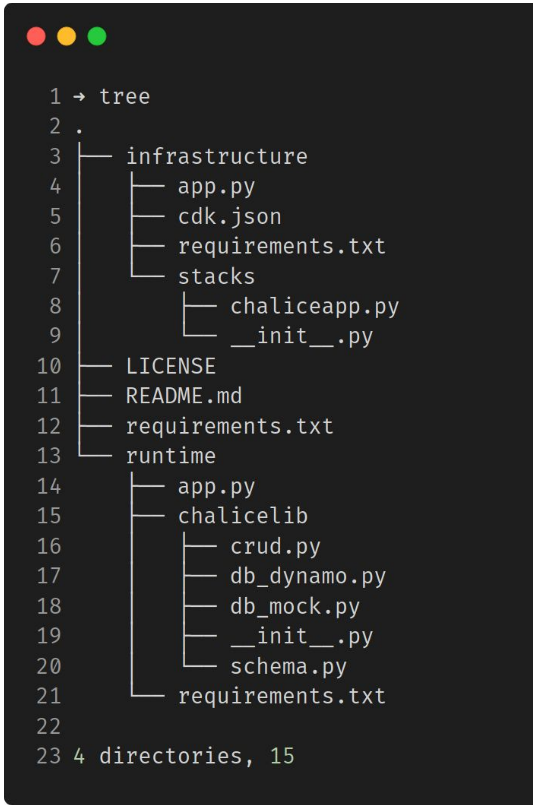
https://github.com/jdalzatec/pycon-2021-code



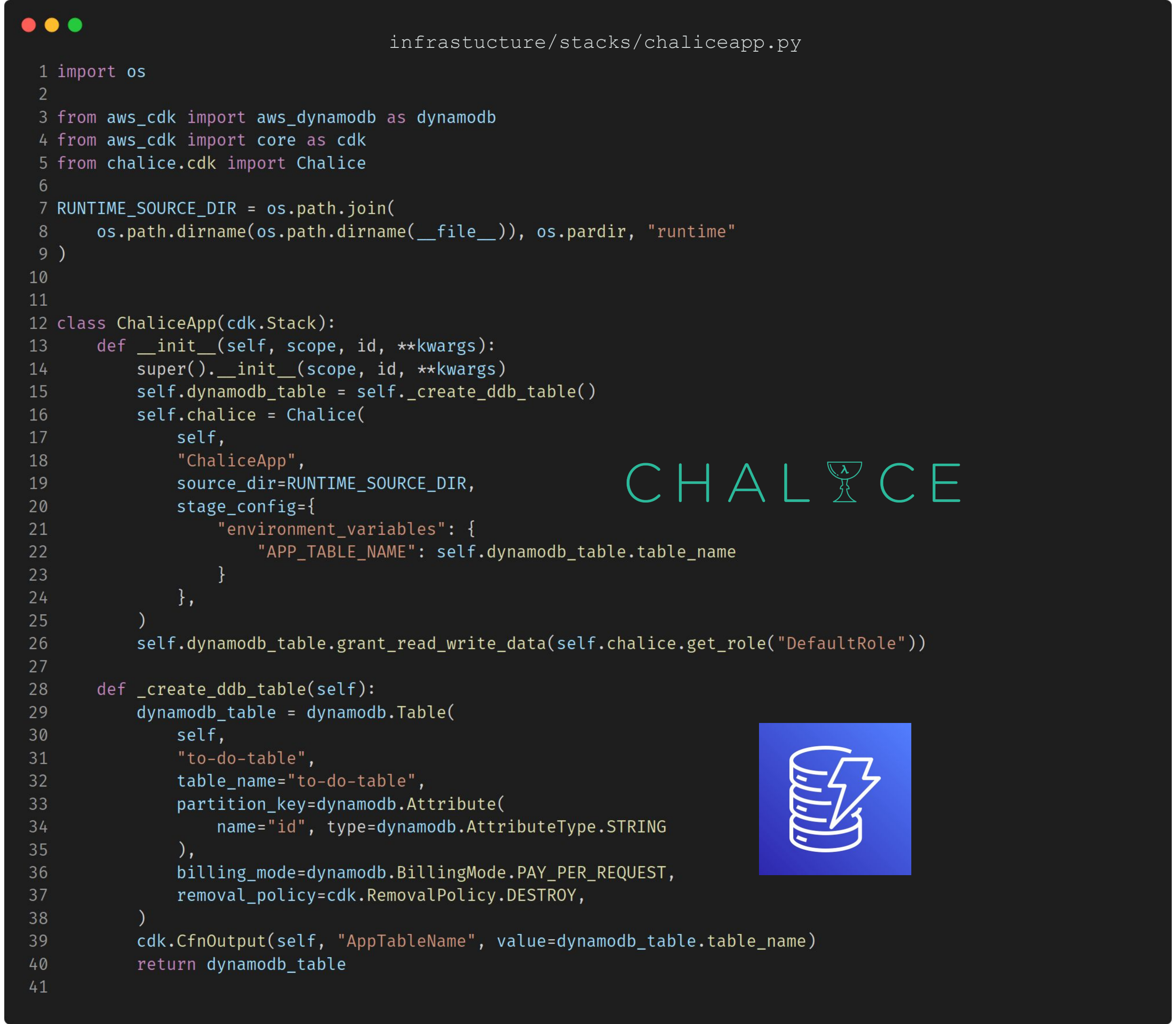
Full example

https://github.com/jdalzatec/pycon-2021-code

```
(venv) → proj
```



```
→ tree
.
├── infrastructure
│   ├── app.py
│   ├── cdk.json
│   ├── requirements.txt
│   └── stacks
│       ├── chaliceapp.py
│       └── __init__.py
├── LICENSE
├── README.md
├── requirements.txt
└── runtime
    ├── app.py
    ├── chalicelib
    │   ├── crud.py
    │   ├── db_dynamo.py
    │   ├── db_mock.py
    │   ├── __init__.py
    │   └── schema.py
    └── requirements.txt

4 directories, 15
```

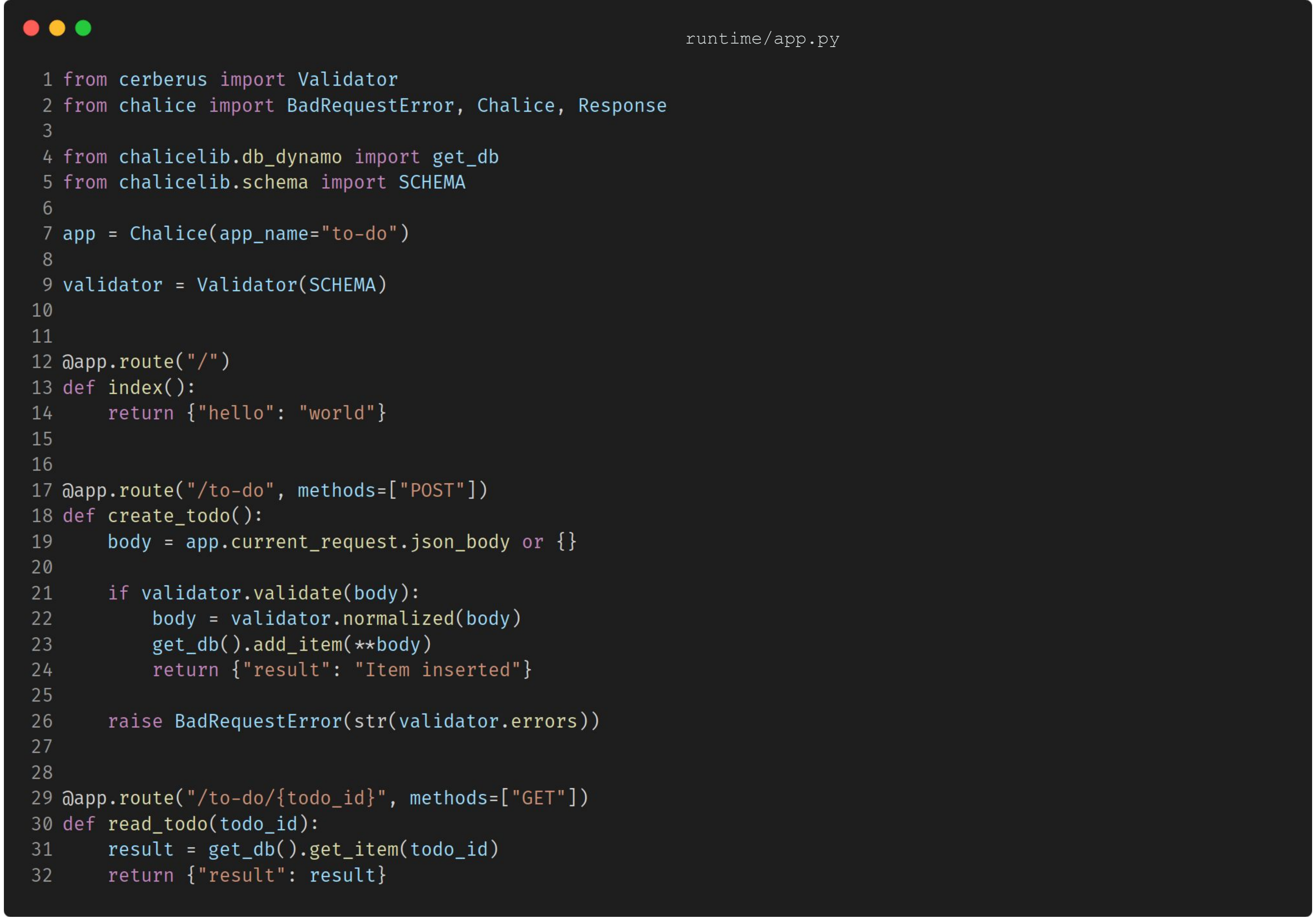


infrastucture/stacks/chaliceapp.py

```python
import os

from aws_cdk import aws_dynamodb as dynamodb
from aws_cdk import core as cdk
from chalice.cdk import Chalice

RUNTIME_SOURCE_DIR = os.path.join(
    os.path.dirname(os.path.dirname(__file__)), os.pardir, "runtime"
)


class ChaliceApp(cdk.Stack):
    def __init__(self, scope, id, **kwargs):
        super().__init__(scope, id, **kwargs)
        self.dynamodb_table = self._create_ddb_table()
        self.chalice = Chalice(
            self,
            "ChaliceApp",
            source_dir=RUNTIME_SOURCE_DIR,
            stage_config={
                "environment_variables": {
                    "APP_TABLE_NAME": self.dynamodb_table.table_name
                }
            },
        )
        self.dynamodb_table.grant_read_write_data(self.chalice.get_role("DefaultRole"))

    def _create_ddb_table(self):
        dynamodb_table = dynamodb.Table(
            self,
            "to-do-table",
            table_name="to-do-table",
            partition_key=dynamodb.Attribute(
                name="id", type=dynamodb.AttributeType.STRING
            ),
            billing_mode=dynamodb.BillingMode.PAY_PER_REQUEST,
            removal_policy=cdk.RemovalPolicy.DESTROY,
        )
        cdk.CfnOutput(self, "AppTableName", value=dynamodb_table.table_name)
        return dynamodb_table
```

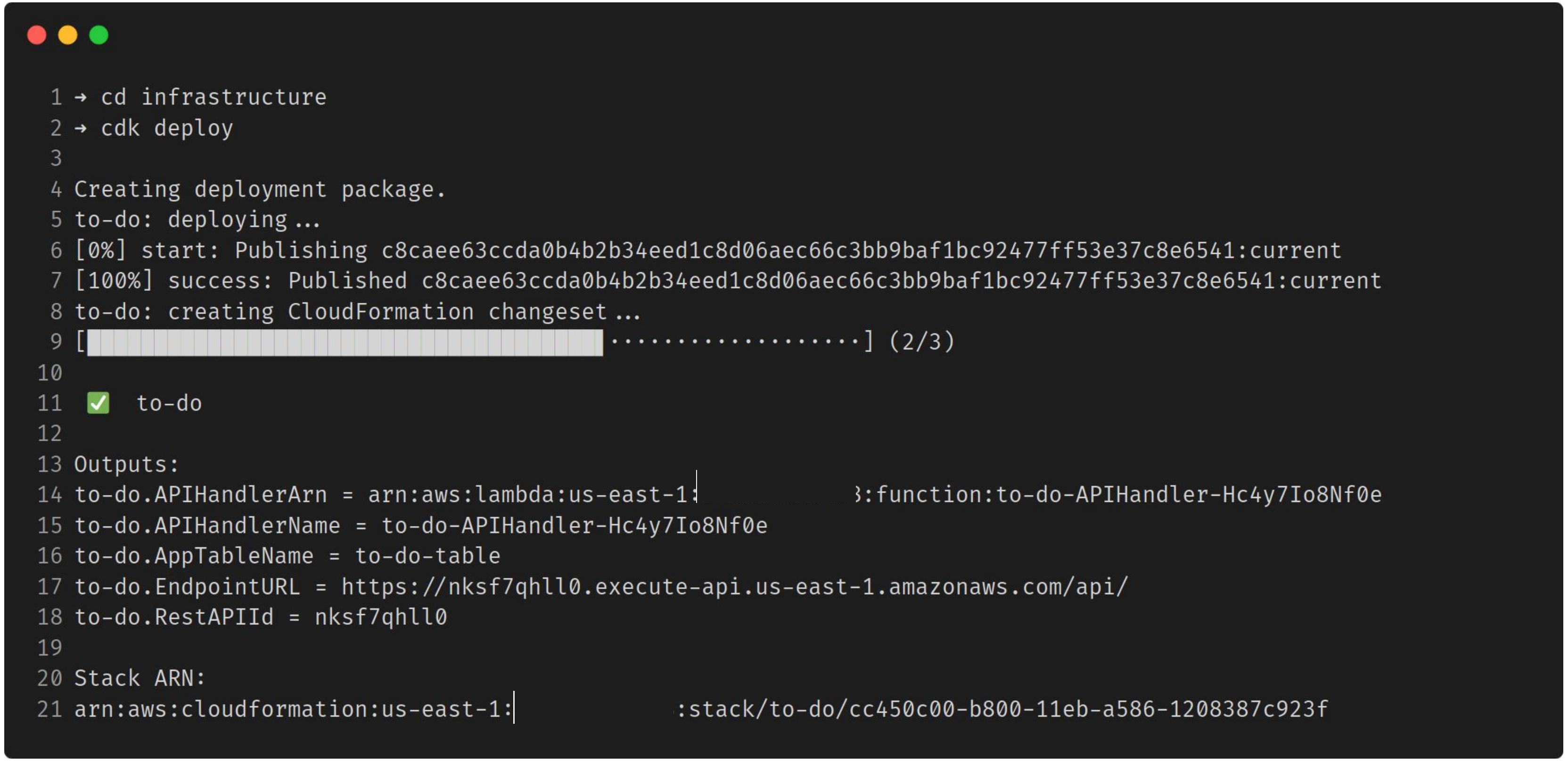
CHALICE

aws


runtime/app.py

```python
from cerberus import Validator
from chalice import BadRequestError, Chalice, Response

from chalicelib.db_dynamo import get_db
from chalicelib.schema import SCHEMA

app = Chalice(app_name="to-do")

validator = Validator(SCHEMA)


@app.route("/")
def index():
    return {"hello": "world"}


@app.route("/to-do", methods=["POST"])
def create_todo():
    body = app.current_request.json_body or {}

    if validator.validate(body):
        body = validator.normalized(body)
        get_db().add_item(**body)
        return {"result": "Item inserted"}

    raise BadRequestError(str(validator.errors))


@app.route("/to-do/{todo_id}", methods=["GET"])
def read_todo(todo_id):
    result = get_db().get_item(todo_id)
    return {"result": result}
```

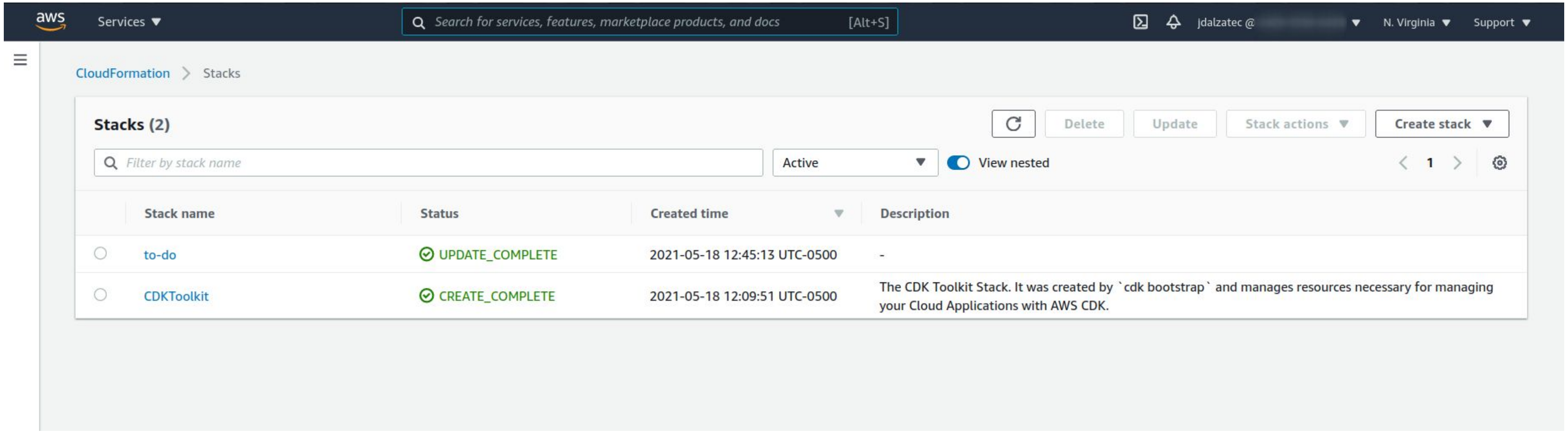
Full example



```
→ cd infrastructure
→ cdk deploy

Creating deployment package.
to-do: deploying...
[0%] start: Publishing c8caee63ccda0b4b2b34eed1c8d06aec66c3bb9baf1bc92477ff53e37c8e6541:current
[100%] success: Published c8caee63ccda0b4b2b34eed1c8d06aec66c3bb9baf1bc92477ff53e37c8e6541:current
to-do: creating CloudFormation changeset...
[██████████████████████████████████████····················] (2/3)

 ✅  to-do

Outputs:
to-do.APIHandlerArn = arn:aws:lambda:us-east-1:            :function:to-do-APIHandler-Hc4y7Io8Nf0e
to-do.APIHandlerName = to-do-APIHandler-Hc4y7Io8Nf0e
to-do.AppTableName = to-do-table
to-do.EndpointURL = https://nksf7qhll0.execute-api.us-east-1.amazonaws.com/api/
to-do.RestAPIId = nksf7qhll0

Stack ARN:
arn:aws:cloudformation:us-east-1:            :stack/to-do/cc450c00-b800-11eb-a586-1208387c923f
```

Full example

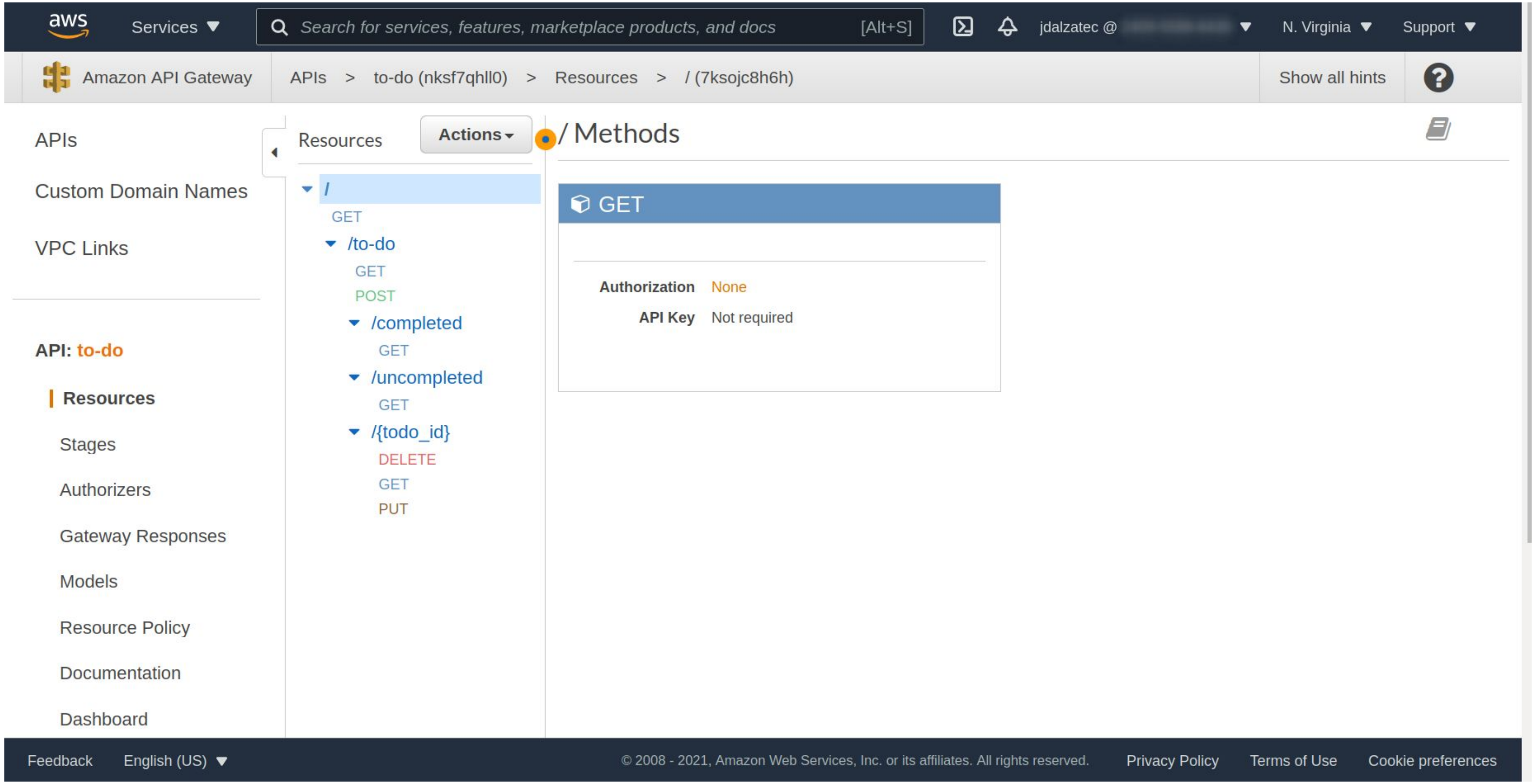
https://github.com/jdalzatec/pycon-2021-code



Stack name	Status	Created time	Description
to-do	UPDATE_COMPLETE	2021-05-18 12:45:13 UTC-0500	-
CDKToolkit	CREATE_COMPLETE	2021-05-18 12:09:51 UTC-0500	The CDK Toolkit Stack. It was created by `cdk bootstrap` and manages resources necessary for managing your Cloud Applications with AWS CDK.

Full example

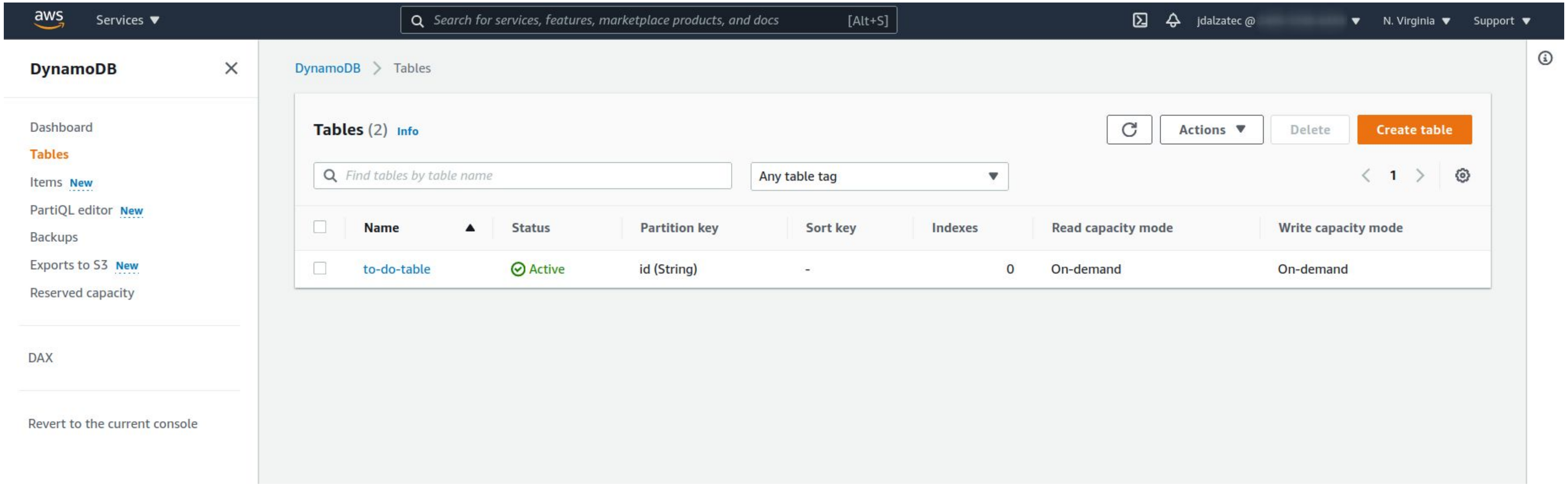
https://github.com/jdalzatec/pycon-2021-code



Authorization	None
API Key	Not required

Full example

https://github.com/jdalzatec/pycon-2021-code



Tables (2) Info

Name	Status	Partition key	Sort key	Indexes	Read capacity mode	Write capacity mode
to-do-table	✓ Active	id (String)	-	0	On-demand	On-demand

¡Gracias!