

Best Practices for Deploying Well- Architected, Serverless Python Applications to AWS with CDK

Kristi Perreault

Principal Software Engineer
Liberty Mutual Insurance



Organized by:



python
& cloud
expert
consulting

Agenda

Introduction

Why AWS CDK?

Python CDK specifics

Well-Architected Best Practices

More resources

About Liberty Mutual Insurance



Our purpose: We exist to help people embrace today and confidently pursue tomorrow.



Our promise: We promise protection for the unexpected, delivered with care.





For more information:
lmi.coltech



Kristi Perreault

Principal Software Engineer @ LM

Cloud & Serverless Engineering

5 years experience

AWS Serverless Hero

June 2022

MS in Electrical & Computer Eng.

Boston University, January 2021

New Hampshire, USA

AWS Portsmouth User Group Organizer

Women in Tech Advocate



Cloud Development

Business logic code

- Application code
- Written in language of choice

Infrastructure as Code

- Building blocks of your application
- Define the resources you want to use

You Have Options!

CloudFormation, SAM,
Serverless Framework,
Terraform

Define in YAML/JSON

Pulumi & CDK

Define in your language of
choice (python!)

There's No Lock In

CDK
constructs in
Pulumi

Import SAM
templates

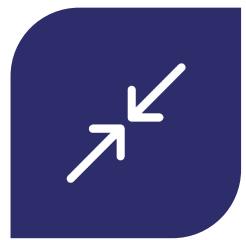
CDKTF
module for
Terraform

“AWS Cloud Development Kit (CDK) accelerates cloud development using common programming languages to model your applications”

- AWS Website



Why CDK?



EASE OF USE



FLEXIBLE



ROBUST
DOCUMENTATION

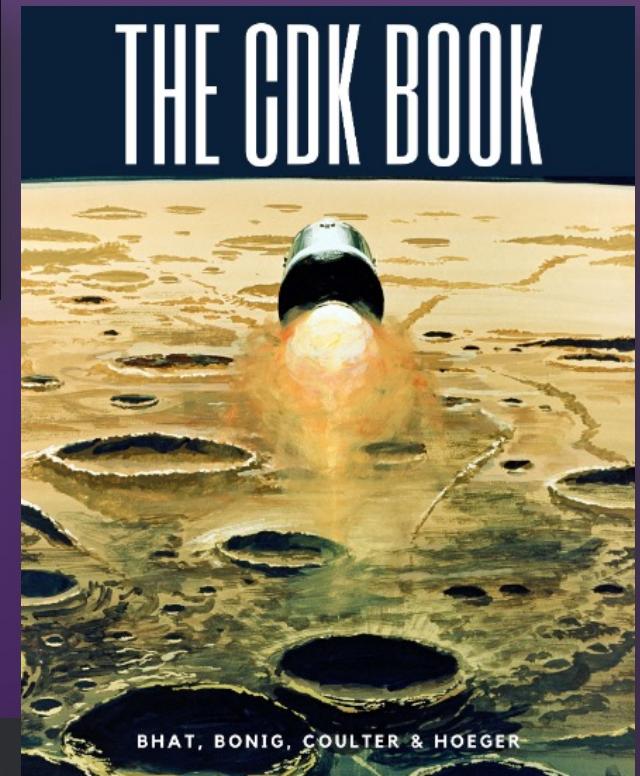


OPEN SOURCE,
MANAGED BY AWS



STRONG
COMMUNITY

CDK Support



Simplify cloud development with constructs

Find and use open-source Cloud Development Kit (CDK) libraries

Search 1400+ construct libraries

[Find constructs](#)

AWS CDK Workshop

Search

English

Prerequisites

Welcome Developers!

Hey there, and thanks for joining us! Hope you can't wait to play with this new thing we call the "AWS Cloud Development Kit" or in short, the AWS CDK.



Less Talk, More Code



What does this look like?

```
queue = sqs.Queue(  
    self, "CdkExampleQueue",  
    visibility_timeout=Duration.seconds(300),  
)  
  
topic = sns.Topic(  
    self, "CdkExampleTopic"  
)  
  
topic.add_subscription(subs.SqsSubscription(queue))
```

CDK Sample App



CDK Synth

```
{  
  "Resources": {  
    "CdkExampleQueue7618E31B": {  
      "Type": "AWS::SQS::Queue",  
      "Properties": {  
        "VisibilityTimeout": 300  
      },  
      "UpdateReplacePolicy": "Delete",  
      "DeletionPolicy": "Delete",  
      "Metadata": {  
        "aws:cdk:path": "cdk-example/CdkExampleQueue/Resource"  
      }  
    },  
    "CdkExampleQueuePolicy839151B5": {  
      "Type": "AWS::SQS::QueuePolicy",  
      "Properties": {  
        "PolicyDocument": {  
          "Statement": [  
            {  
              "Action": "sns:SendMessage",  
              "Condition": {  
                "ArnEquals": {  
                  "aws:SourceArn": {  
                    "Ref": "CdkExampleTopicDAE4807D"  
                  }  
                }  
              },  
              "Effect": "Allow",  
              "Principal": {  
                "Service": "sns.amazonaws.com"  
              },  
              "Resource": {  
                "Fn::GetAtt": [  
                  "CdkExampleQueue7618E31B",  
                  "Arn"  
                ]  
              }  
            ],  
            "Version": "2012-10-17"  
          },  
          "Queues": [  
            {  
              "Ref": "CdkExampleQueue7618E31B"  
            }  
          ]  
        },  
        "Metadata": {  
          "aws:cdk:path": "cdk-example/CdkExampleQueue/Policy/Resource"  
        }  
      },  
      "CdkExampleQueuecdkexampleCdkExampleTopic69C35F67F10BBE4B": {  
        "Type": "AWS::SNS::Subscription",  
        "Properties": {  
          "Protocol": "sq",  
          "TopicArn": {  
            "Ref": "CdkExampleTopicDAE4807D"  
          },  
          "Endpoint": {  
            "Fn::GetAtt": [  
              "CdkExampleQueue7618E31B",  
              "Arn"  
            ]  
          }  
        }  
      }  
  },  
  "Outputs": {}  
}
```

Project Structure

Configuration files at project root

Organize directory by logical units

Constructs, not stacks

```
# The recommended project structure example

.
| '-- backend
|   | '-- api
|   |   | '-- runtime
|   |   |   | '-- lambda_function.py
|   |   |   | '-- requirements.txt
|   |   |   '-- infrastructure.py
|   | '-- database
|   |   '-- infrastructure.py
| '-- app.py
| '-- constants.py
| '-- requirements.txt
```

Let's Talk Python Tools



Boto

- AWS SDK for Python
- Interact with AWS services
- Create, update, configure, delete

Code Examples from Boto Documentation

```
# Get the service resource
sns = boto3.resource('sns')

# Create the queue. This returns an SQS.Queue instance
queue = sns.create_queue(QueueName='test', Attributes={'DelaySeconds': '5'})
```

```
import boto3

# Create CloudWatch client
cloudwatch = boto3.client('cloudwatch')

# Delete alarm
cloudwatch.delete_alarms(
    AlarmNames=['Web_Server_CPU_Utilization'],
)
```

```
# Retrieve the list of existing buckets
s3 = boto3.client('s3')
response = s3.list_buckets()
```

Lambda Powertools

Library for observability
best practices

Tracer

```
@tracer.capture_lambda_handler
def lambda_handler(event, context):
```

Metrics

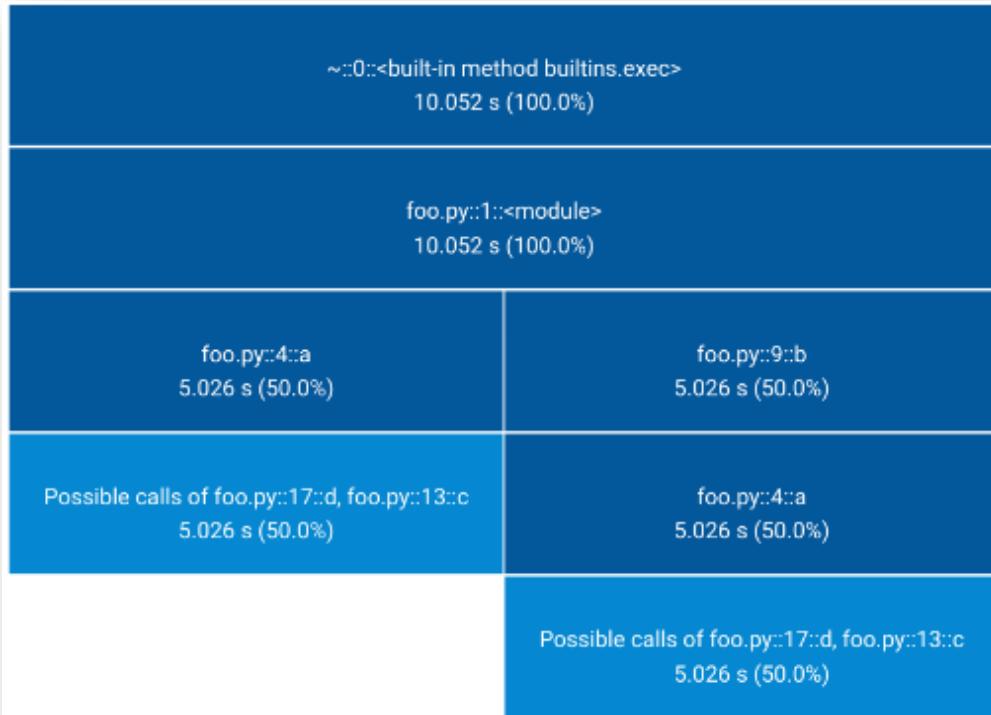
```
@metrics.log_metrics(capture_cold_start_metric=True)
def lambda_handler(event, context):
...
check_out_cart() # Function to process the checkout
metrics.add_metric(name="CartCheckedOut", unit="Count", value=1)
```

Logger

```
@logger.inject_lambda_context(log_event=True)
def lambda_handler(event, context):
{
    "timestamp": "2020-06-15 08:27:42,730",
    "level": "INFO",
    "location": "<module>:17",
    "service": "shopping-cart",
    "sampling_rate": 0,
    "cold_start": false,
    "function_name": "aws-serverless-shopping-cart",
    "function_memory_size": "512",
    "function_arn": "arn:aws:lambda:us-east-1:123456789012:function:shopping-cart",
    "function_request_id": "de9aee44-12fc-4e1d-85c3-123456789012",
    "message": "Initializing DDB Table aws-serverless-shopping-cart"
}
```

Credit: Tom McCarthy, 2020

Tuna



<https://github.com/nschloe/tuna>

- Python performance analysis
- Profiler visualizer for cold starts
- CLI install
- Create or import a profile
- Use ‘tuna’ to generate your tree

Testing

Pytest

- Out-of-the-box
- Typical ‘Assert’ tests
- CDK Assertions with ‘Template’

```
template.has_resource_properties(  
    "AWS::Lambda::Function",  
    {  
        "Handler": "handler",  
        "Runtime": "nodejs14.x",  
    },  
)
```

<https://docs.aws.amazon.com/cdk/v2/guide/testing.html>

Moto

- Boto’s mock testing library
- Decorators, context manager, raw, unit tests

```
@pytest.fixture(scope="function")  
def s3(aws_credentials):  
    with mock_s3():  
        yield boto3.client("s3", region_name="us-east-1")
```

Tox

- Virtual Env Manager for running tests
- Config with a tox.ini file

```
[tox]  
min_version = 4.0  
env_list =  
    py310  
    py39  
    type  
  
[testenv]  
deps = pytest  
commands = pytest tests  
  
[testenv:type]  
deps = mypy  
commands = mypy src
```

Best Practices



Well Architected Pillars



Security



Cost



Operational
Excellence



Reliability



Performance



Sustainability

Well Architected Pillar Examples

Security

IAM roles & policies

Cost

Minimize API calls

Operational Excellence

Lambda Powertools Tracer & CFN-Lint

Reliability

Multi-region & throttling

Performance

Lambda Powertools Tracer & Metrics

Sustainability

Optimize lambda usage with....Powertools!

CDK Starter Pattern



BEST PRACTICES OUT-
OF-THE-BOX



SECURITY &
COMPLIANCE ARE
BAKED IN



CUSTOMIZE & BUILD
ON THE BASICS



PROMOTES REUSE &
COLLABORATION



LEADS TO
COMMUNITY DRIVEN
DEVELOPMENT

Looking for More?



Further Reading & References

- AWS CDK Website: <https://aws.amazon.com/cdk/>
- CDK Documentation: <https://docs.aws.amazon.com/cdk/api/v2/>
- CDK Workshop: <https://cdkworkshop.com/>
- CDK Construct Hub: <https://constructs.dev/>
- CDK Book: <https://www.thecdkbook.com/>
- CDK Day: <https://www.cdkday.com/>
- Project structure blog: <https://aws.amazon.com/blogs/developer/recommended-aws-cdk-project-structure-for-python-applications/>
- Boto: <https://boto3.amazonaws.com/v1/documentation/api/latest/index.html>
- Lambda Powertools: <https://awslabs.github.io/aws-lambda-powertools-python/2.9.1/> & <https://aws.amazon.com/blogsopensource/simplifying-serverless-best-practices-with-lambda-powertools/>
- Tuna: <https://github.com/nschloe/tuna>
- Moto: <http://docs.getmoto.org/en/latest/>
- Tox: <https://tox.wiki/en/latest/>
- Well Architected: <https://docs.aws.amazon.com/wellarchitected/latest/framework/welcome.html>

Thank You!

Twitter @kperreault95
LinkedIn Kristi-Perreault