Functional IaC

Infrastructure as Code using strongly typed functional languages

Matteo Di Pirro & Andrea Zoleo, Kynetics

Agenda

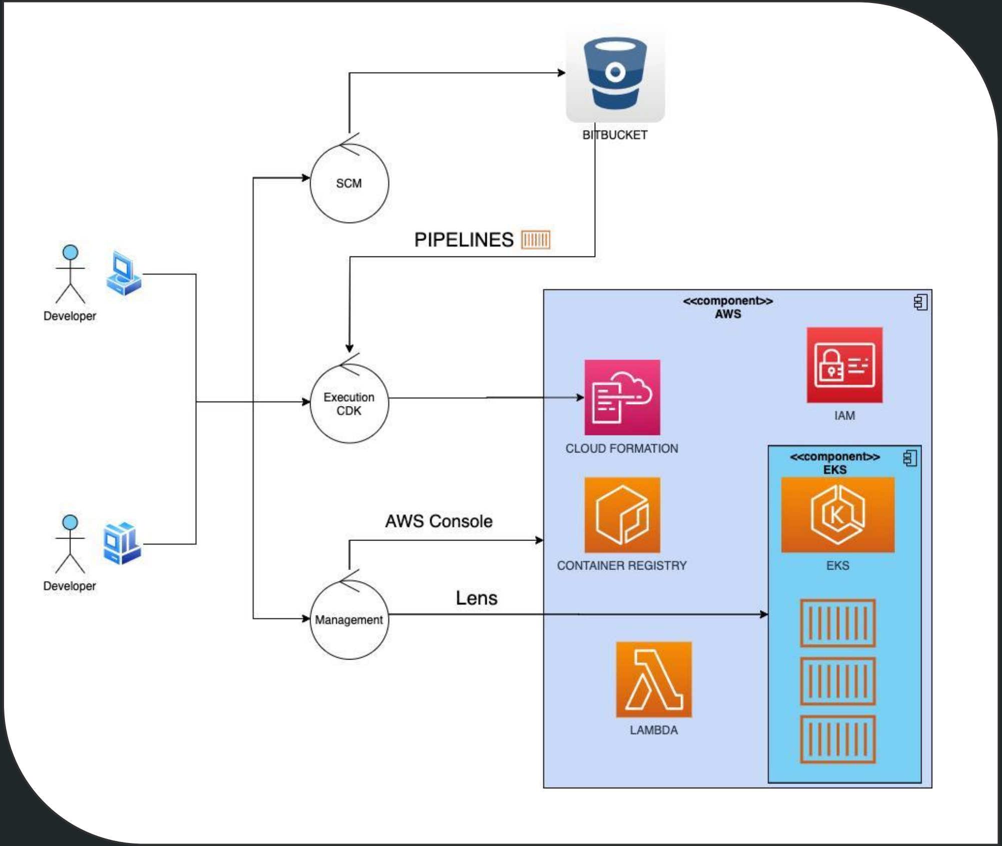
- Background
- Infrastructure as Code in AWS
- Functional IaC
- Conclusions

“The enabling idea of infrastructure as code is that the ***systems and devices which are used to run software can be treated as*** *if they, themselves, are* ***software****.”*

Kief Morris

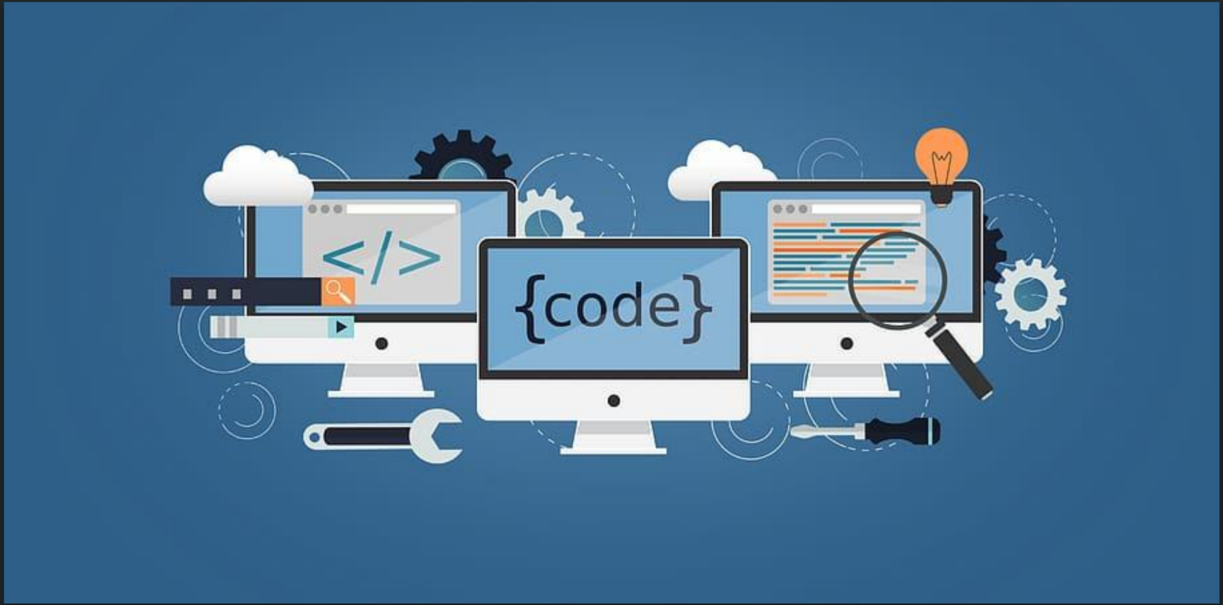
Context of this Work

UpdateFactory is our artifact content and software update delivery IoT Platform, powered by Eclipse hawkBit.

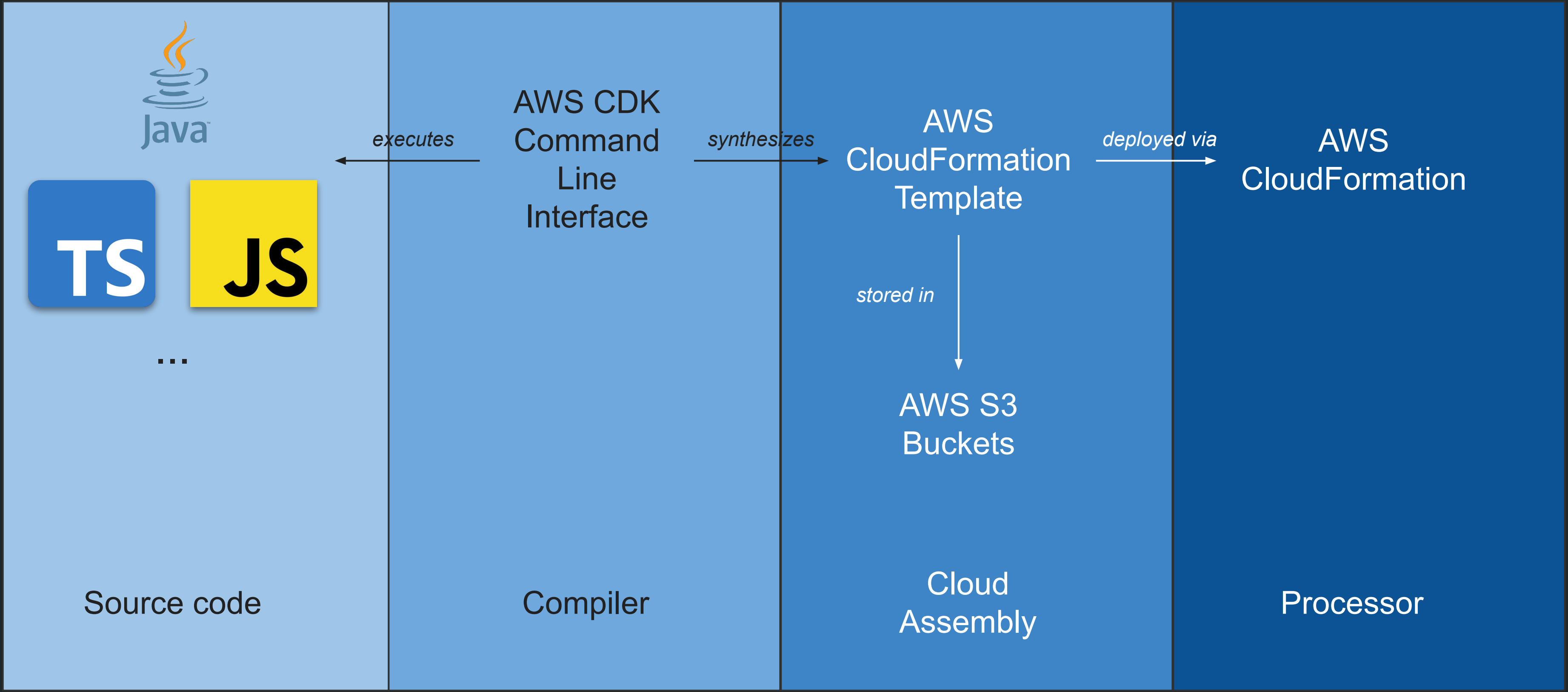


AWS Infrastructure as Code

- REST
- Language SDK
- Terraform
- CloudFormation
- CDK



AWS CDK



www.kynetics.com

A look at the code

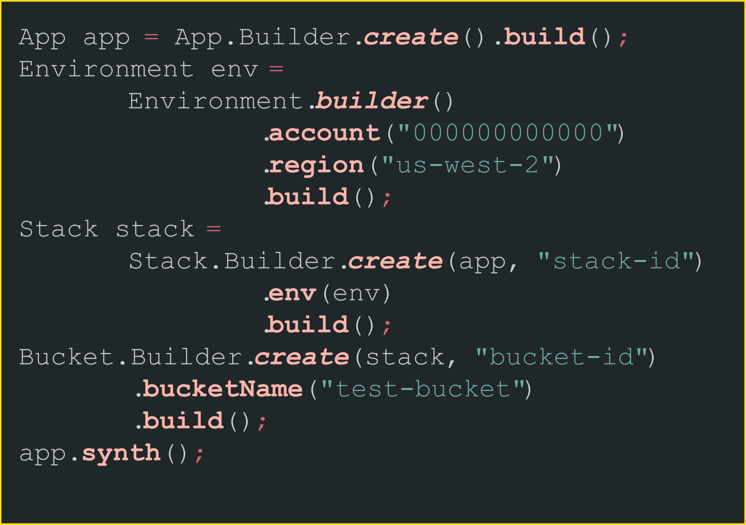
Scala & Kotlin

Why Scala for lambdas and Kotlin for CDK

1. Functional and Typed (Scala more than Kotlin)
2. Kotlin instead of TypeScript in CDK because of popularity (and also Gradle ;-))
3. Scala for lambdas because we love it

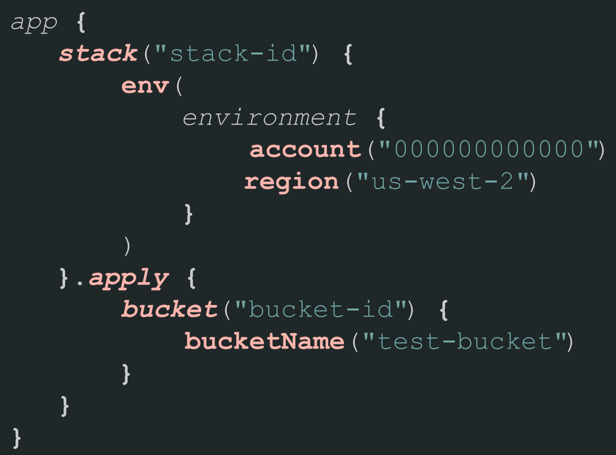
Declarative IaC

AWS CDK (Java)



```
App app = App.Builder.create().build();
Environment env =
       Environment.builder()
               .account("000000000000")
               .region("us-west-2")
               .build();
Stack stack =
       Stack.Builder.create(app, "stack-id")
               .env(env)
               .build();
Bucket.Builder.create(stack, "bucket-id")
       .bucketName("test-bucket")
       .build();
app.synth();
```

AWS CDK (Kotlin, sugared)



```
app {
    stack("stack-id") {
        env(
            environment {
                account("000000000000")
                region("us-west-2")
            }
        )
    }.apply {
        bucket("bucket-id") {
            bucketName("test-bucket")
        }
    }
}
```

Declarative IaC

AWS CDK (Java)

```
App app = App.Builder.create().build();
Environment env =
       Environment.builder()
               .account("000000000000")
               .region("us-west-2")
               .build();
Stack stack =
       Stack.Builder.create(app, "stack-id")
               .env(env)
               .build();
Bucket.Builder.create(stack, "bucket-id")
       .bucketName("test-bucket")
       .build();
app.synth();
```

AWS CDK (Kotlin, sugared)

```
app {
   stack("stack-id") {
       env(
           environment {
               account("000000000000")
               region("us-west-2")
           }
       )
   }.apply {
       bucket("bucket-id") {
           bucketName("test-bucket")
       }
   }
}
```

A look at the code

AWS Lambda

- Function as a Service
  - Serverless, event-driven compute service
  - Well suited for one-shot computations or recurring tasks
- Used to run SDK calls from CDK code (lambda-based custom resources)
- Different runtimes for different languages
  - Built-in (Java, Python, Node.js, etc)
  - Custom (from Docker image)

GraalVM

- Key Features
  - High Performance
  - AOT Native Image Compilation
  - Polyglot Programming
  - Advanced Tools
- Native-image
  - Small Footprint
  - Improved Security
  - Fast Startup
  - Ideal for Containers

```
BUG/65 Version 2.0
  Copyright 1982 McStuff Co.
>D 0600
0600 D8 18 A9 02 69 02 85 CB
>L 0600

>H
0000 0000
>Y 0600 0608
0600   CLD          D8
0601   CLC          18
0602   LDA #$02     A9 02
0604   ADC #$02     69 02
0606   STA $CB      85 CB
0608   RTS          60
>
```

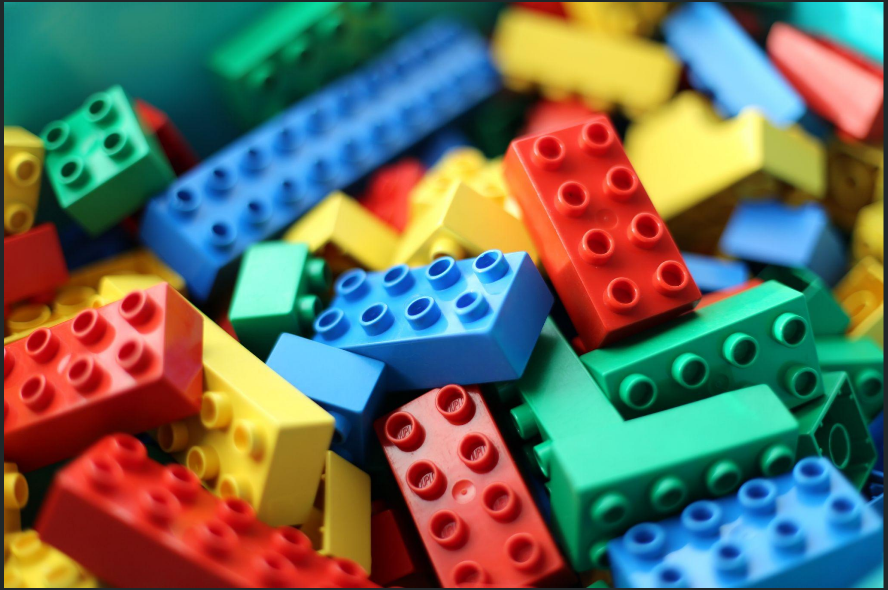
A look at the code

Conclusions

CDK → advanced IaC technology, but no multi-cloud

CDK → TypeScript vs JVM

JVM → Java vs Kotlin/Scala



Thank you

Andrea Zoleo
andrea.zoleo@kynetics.com

Matteo Di Pirro
matteo.dipirro@kynetics.com