AWS Console is Gross
or
Intro to Infrastructure as Code with AWS CDK

WORKSHOP BY: USMAN TAHIR

WWW.ANTEMATTER.IO

AGENDA

- What is Infrastructure as Code (IaC)
- Overview of AWS CDK
  - Benefits of using the CDK
  - How it works
  - Important CDK concepts
  - Some important CDK commands
- Practical
  - Infra 1
  - infra 2
  - infra 3
- Q&A

INFRASTRUCTURE AS CODE?!

Lets you manage and provision tech-infrastructures through code rather than having to manually configure resources

or: Run a single command in your shell and everything gets setup automatically

AWS CDK OVERVIEW

BENEFITS?

VERSION CONTROL

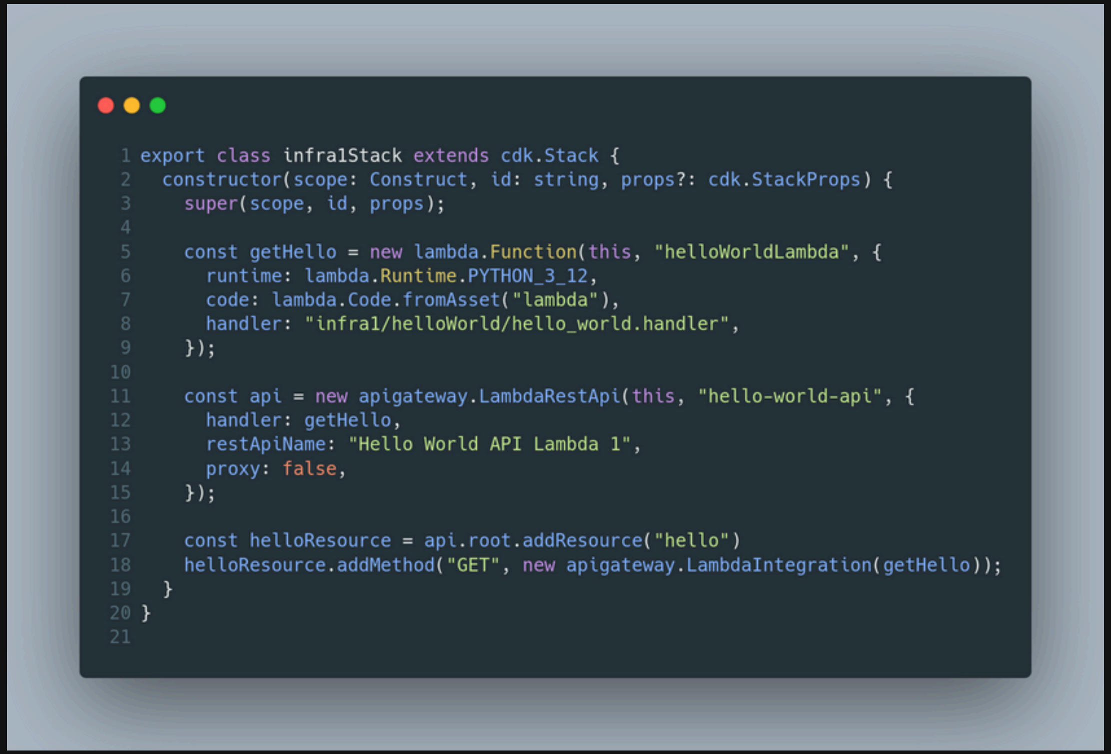
EASIER MIGRATIONS

LESS PRONE TO
HUMAN ERRORS

EASILY SCALEABLE

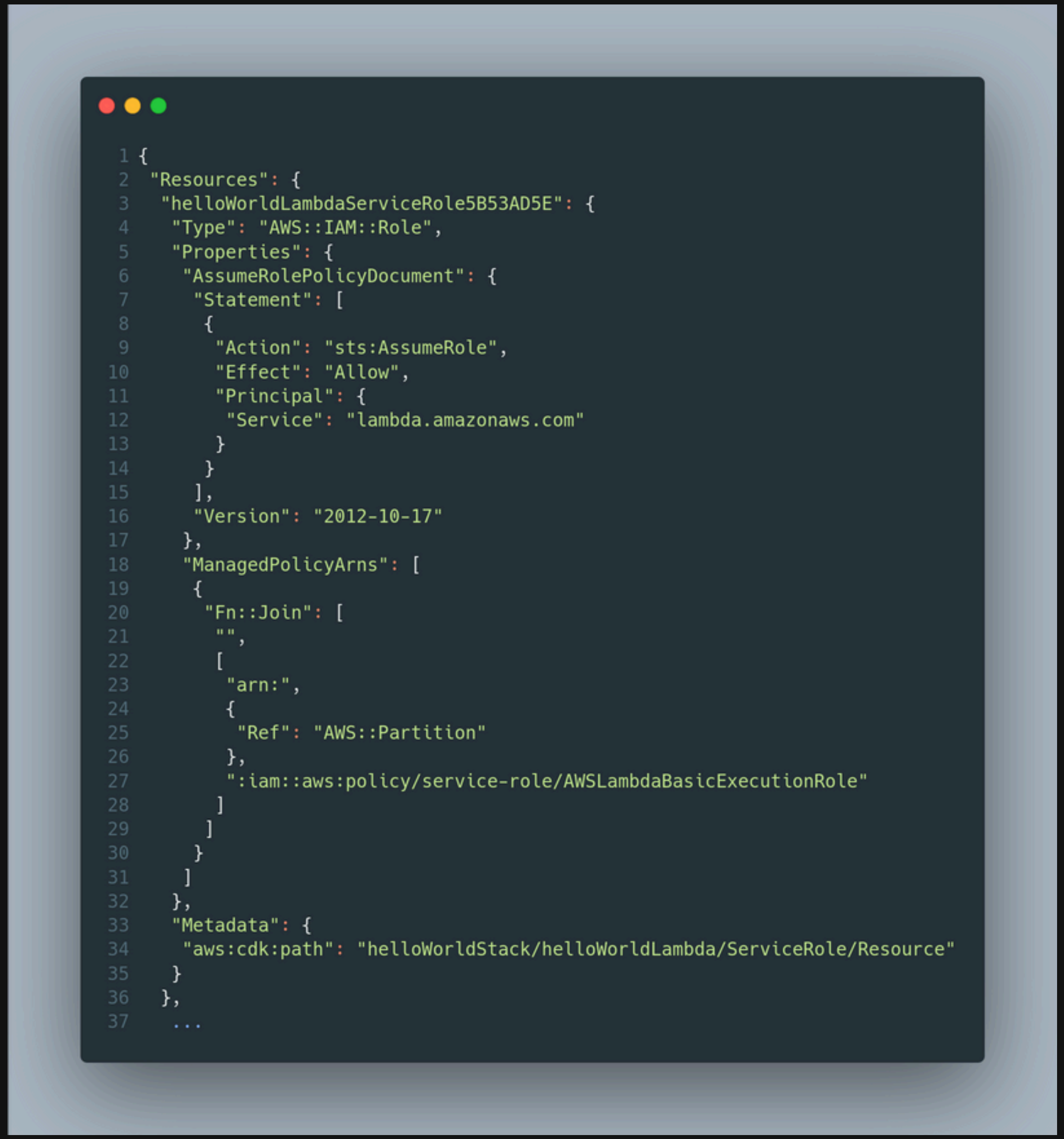
SAVE TIME

HOW THE CDK WORKS



```typescript
export class infra1Stack extends cdk.Stack {
  constructor(scope: Construct, id: string, props?: cdk.StackProps) {
    super(scope, id, props);

    const getHello = new lambda.Function(this, "helloWorldLambda", {
      runtime: lambda.Runtime.PYTHON_3_12,
      code: lambda.Code.fromAsset("lambda"),
      handler: "infra1/helloWorld/hello_world.handler",
    });

    const api = new apigateway.LambdaRestApi(this, "hello-world-api", {
      handler: getHello,
      restApiName: "Hello World API Lambda 1",
      proxy: false,
    });

    const helloResource = api.root.addResource("hello")
    helloResource.addMethod("GET", new apigateway.LambdaIntegration(getHello));
  }
}

```

Your Infrastructure Code



```json
{
 "Resources": {
  "helloWorldLambdaServiceRole5B53AD5E": {
   "Type": "AWS::IAM::Role",
   "Properties": {
    "AssumeRolePolicyDocument": {
     "Statement": [
      {
       "Action": "sts:AssumeRole",
       "Effect": "Allow",
       "Principal": {
        "Service": "lambda.amazonaws.com"
       }
      }
     ],
     "Version": "2012-10-17"
    },
    "ManagedPolicyArns": [
     {
      "Fn::Join": [
       "",
       [
        "arn:",
        {
         "Ref": "AWS::Partition"
        },
        ":iam::aws:policy/service-role/AWSLambdaBasicExecutionRole"
       ]
      ]
     }
    ]
   },
   "Metadata": {
    "aws:cdk:path": "helloWorldStack/helloWorldLambda/ServiceRole/Resource"
   }
  },
   ...
```

Cloudformation Stack

Deployed to AWS

IMPORTANT CDK CONCEPTS

APP

THE ENTRY POINT OF YOUR CDK PROJECT
ORCHESTRATES DEPLOYMENTS
ONE PER PROJECT

STACK

A SINGLE PIECE OF AWS INFRASTRUCTURE FOR YOUR APP
MULTIPLE STACKS CAN BE DEPLOYED PER PROJECT

CONSTRUCT

REPRESENTS ONE OR MORE AWS RESOURCES
REUSABLE
USED WITHIN STACKS TO CREATE INFRASTURES

CONTRUCTS

L1

LOW LEVEL
REQUIRE DETAILED CONFIGURATION
PREFIXED WITH “CFN”
EG: S3.CFNBUCKET

L2

OPINIONATED MEANINGFUL DEFAULTS AND USEFUL METHODS
REQUIRE FEWER LINES OF CODES TO SETUP
ABSTRACTION OVER L1 CONSTRUCTS
EG: S3.BUCKET

L3

PROVIDE COMPLETE ARCHITECTURAL PATTERNS.
IMPLEMENT AWS BEST PRACTICES.
INVOLVE MULTIPLE RESOURCES
EG: S3TOLAMBDA

MOST ABSTRACTION → **LEAST ABSTRACTION**

MOST CUSTOMIZABLE → **LEAST CUSTOMIZABLE**

ANTEMATTER

IMPORTANT CDK COMMANDS

CDK INIT APP --LANGUAGE=<LANGUAGE NAME>

CDK BOOTSTRAP

CDK SYNTH

CDK DEPLOY

CDK DEPLOY --HOTSWAP*

CDK DESTROY

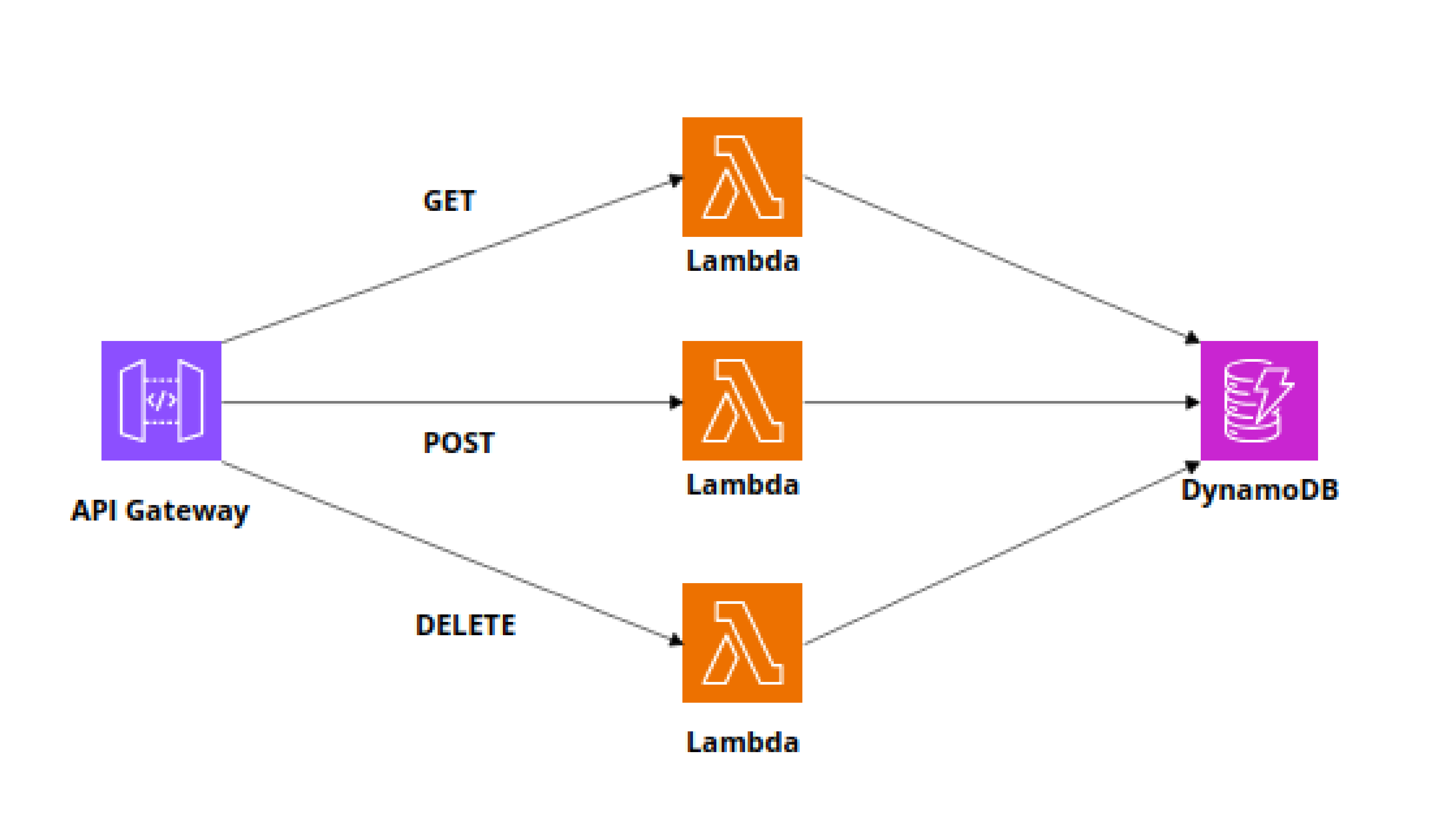
NOTE: NEVER USE --HOTSWAP FLAG WHEN DEPLOYING PRODUCTION INFRAS

PRACTICAL

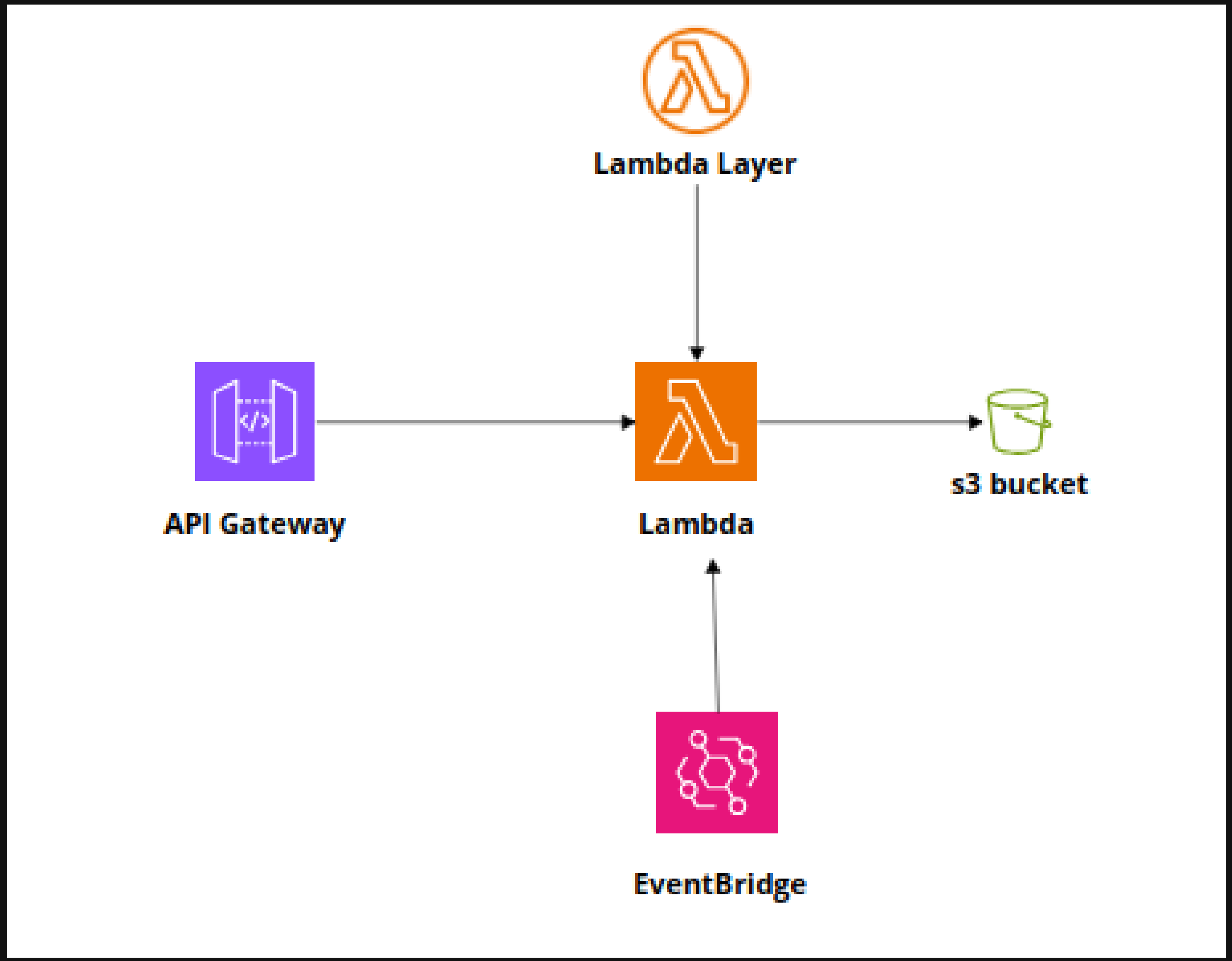
PRE-REQUISITES

- Have the AWS CLI setup on your machine.
- Have the aws-cdk installed: `npm install -g aws-cdk`
- Know typescript or javascript or python or java or C# or Go

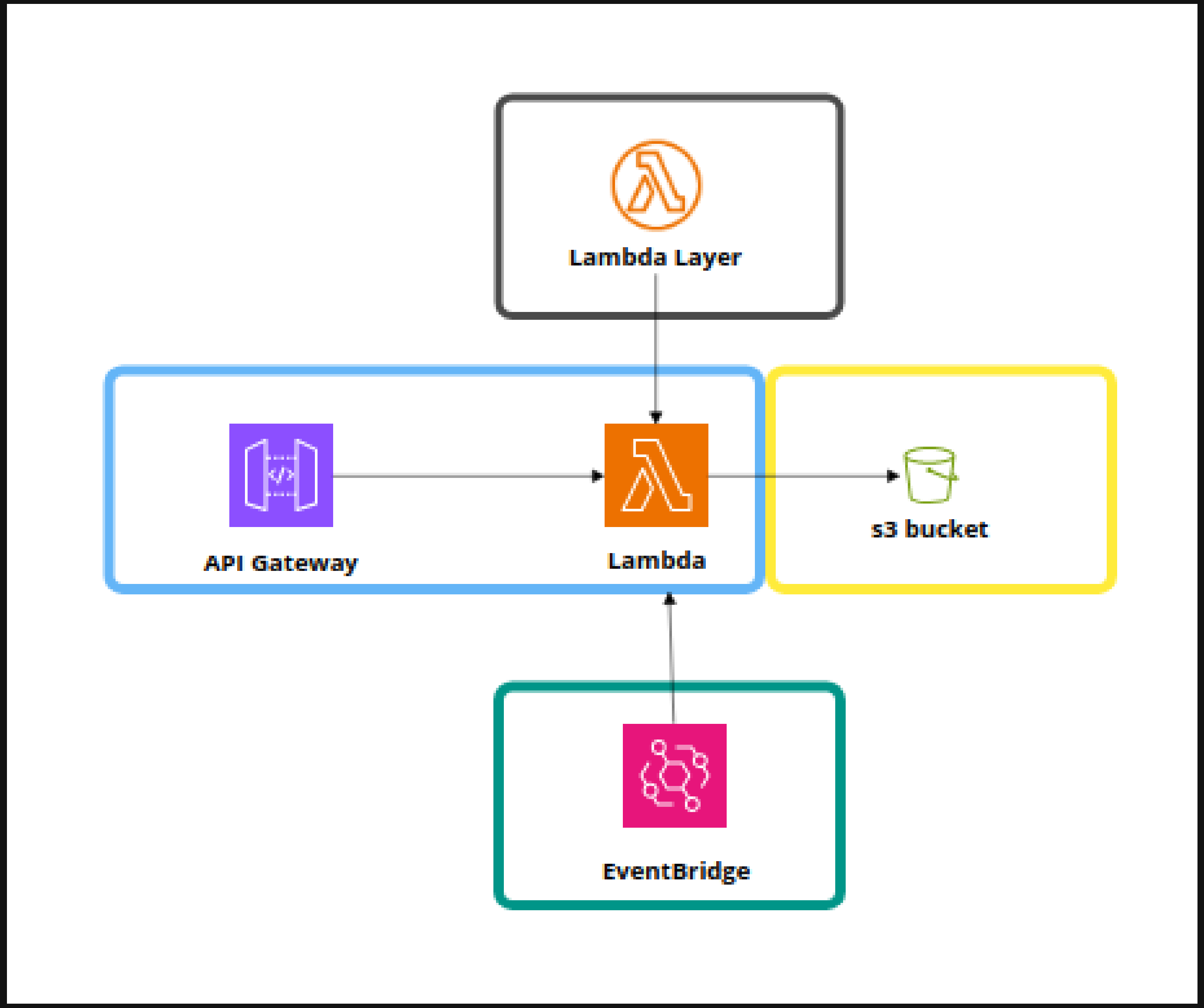
INFRA 1



INFRA 2



INFRA 2



QUESTIONS?