
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

ARC311-R

Build production-ready prototypes rapidly using serverless patterns

Kwonyul Choi (he/him)

Senior Prototyping Engineer/SDE
AWS

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Prototype definition

Prototyping scenario

Build prototype – architecture

Build prototype – tool

Build prototype – implementation

Build prototype – demo

After prototyping

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Prototype definition

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Launch a new product or service

Functional requirements

Business problem
User experience



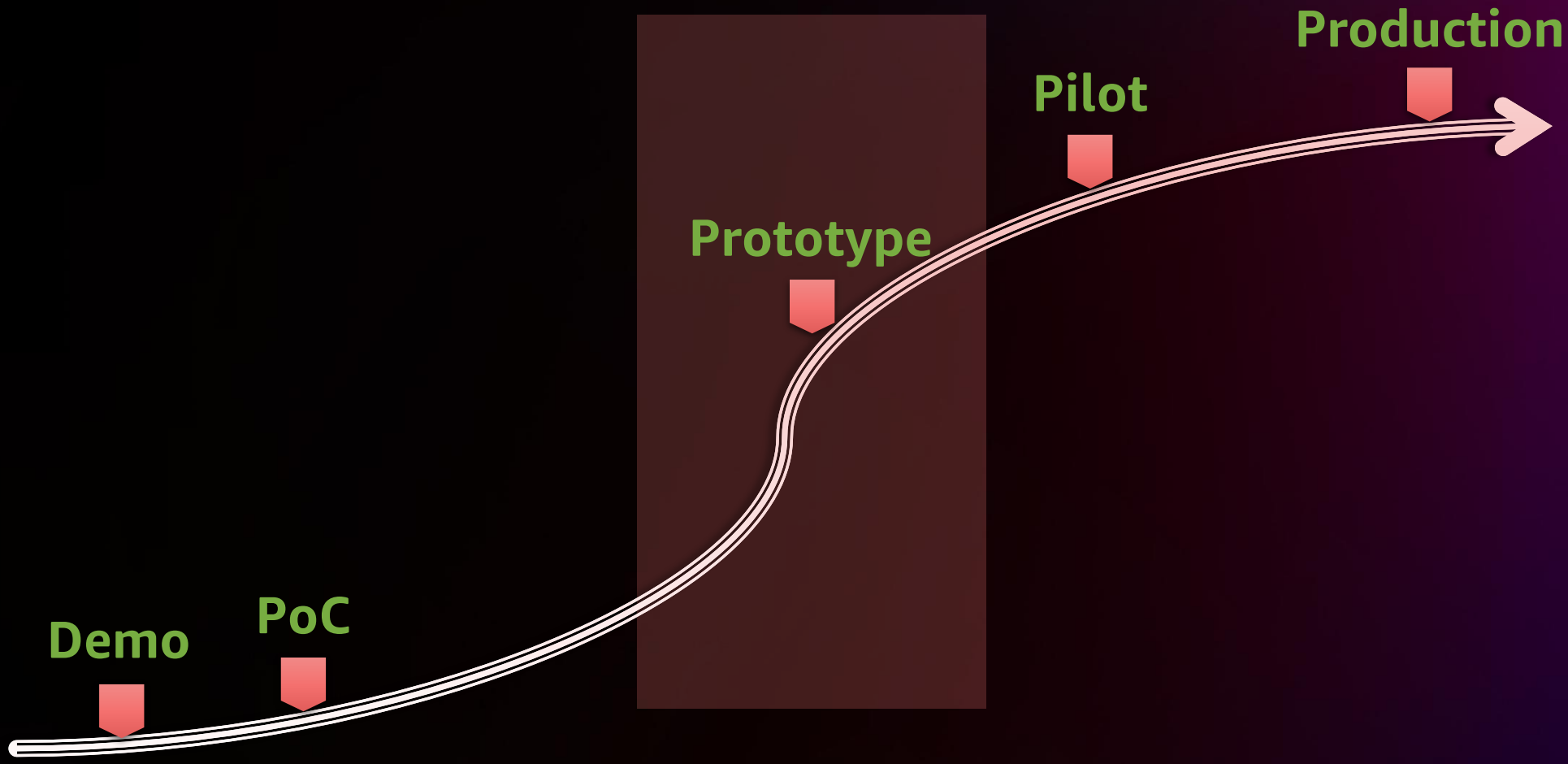
Non-functional requirements

AWS Well-Architected Framework

A new production or service

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

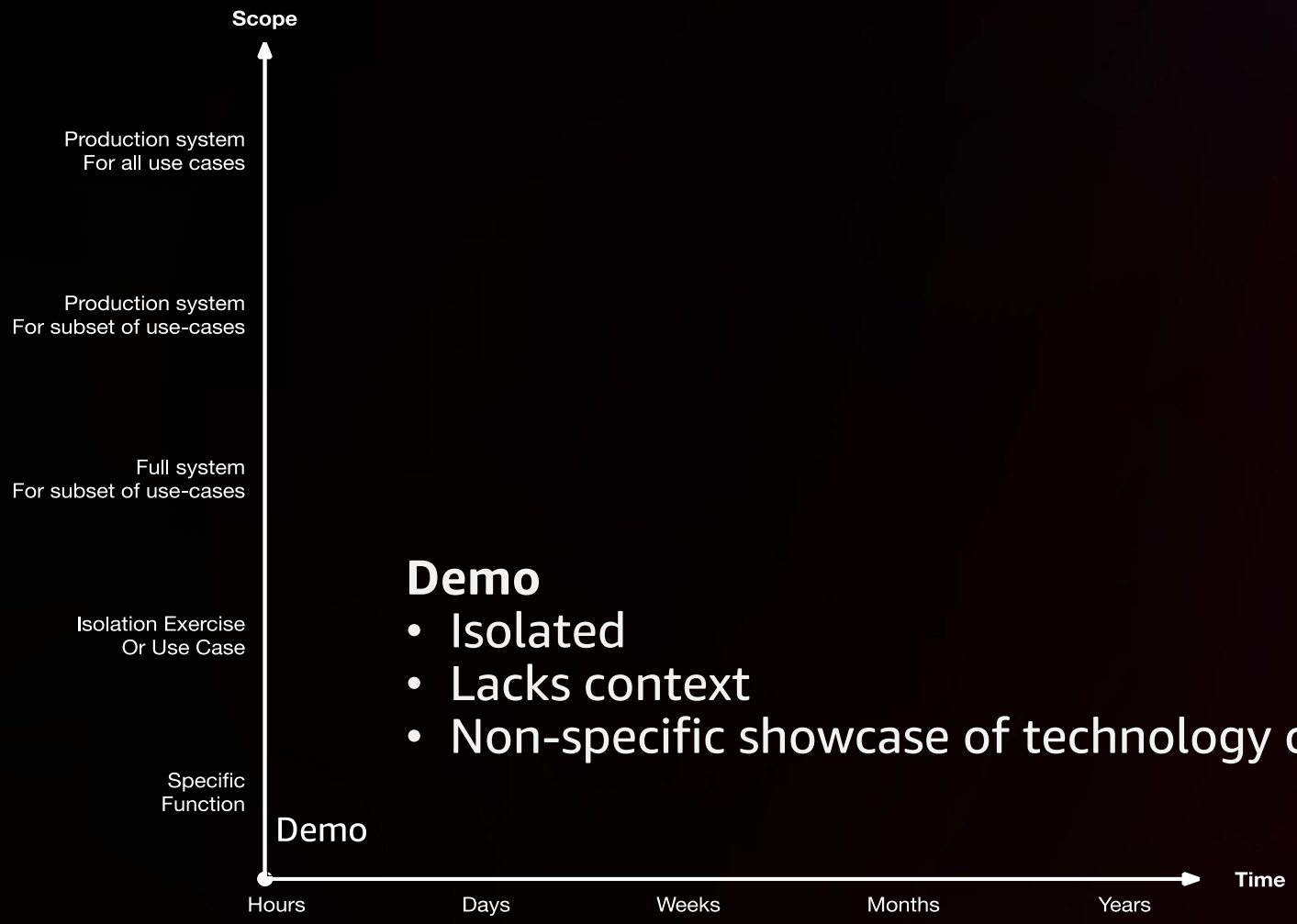
What is prototyping?



The journey from development to production

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What is prototyping?

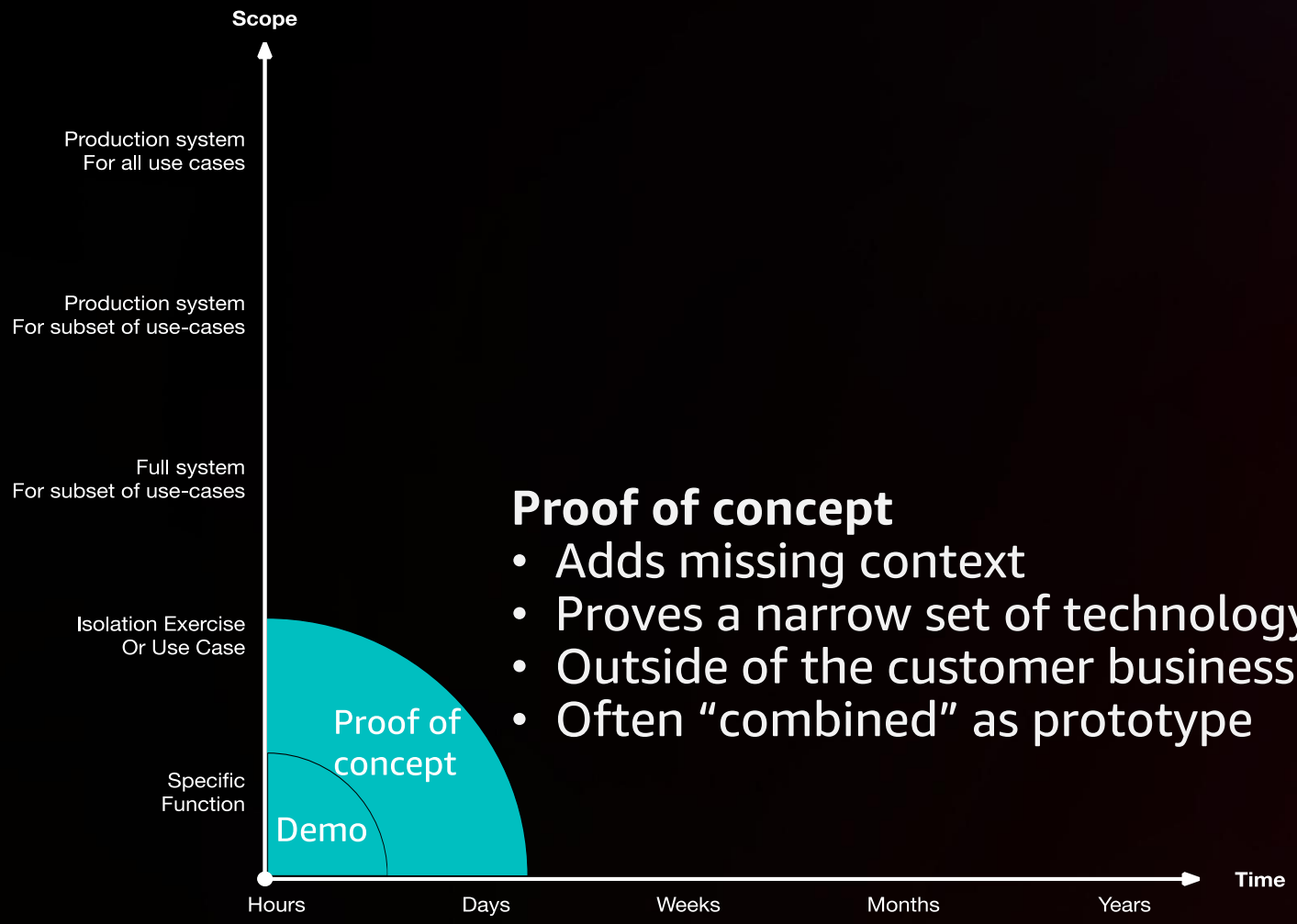


Demo

- Isolated
- Lacks context
- Non-specific showcase of technology capabilities

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What is prototyping?

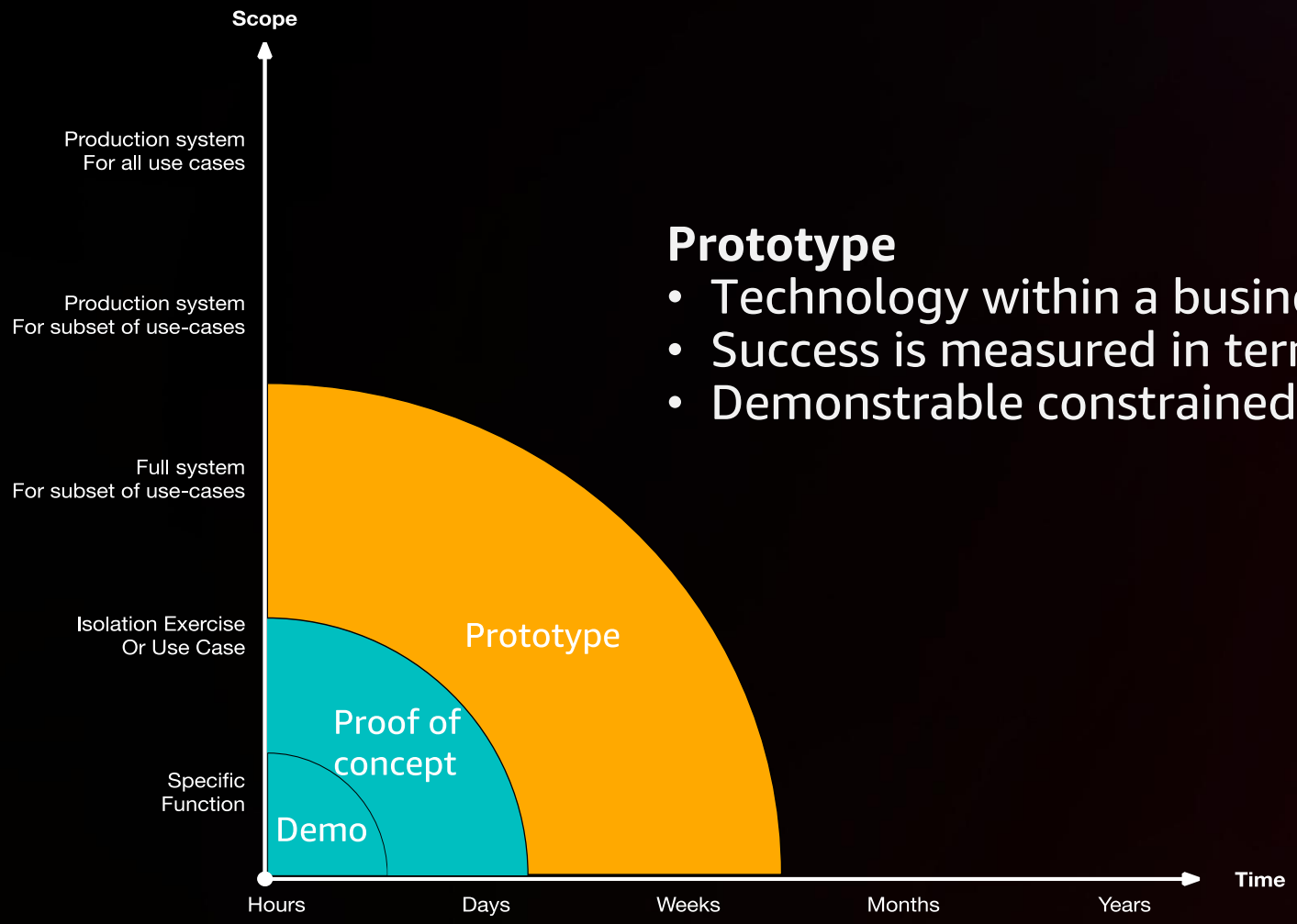


Proof of concept

- Adds missing context
- Proves a narrow set of technology outcomes
- Outside of the customer business framework
- Often “combined” as prototype

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What is prototyping?

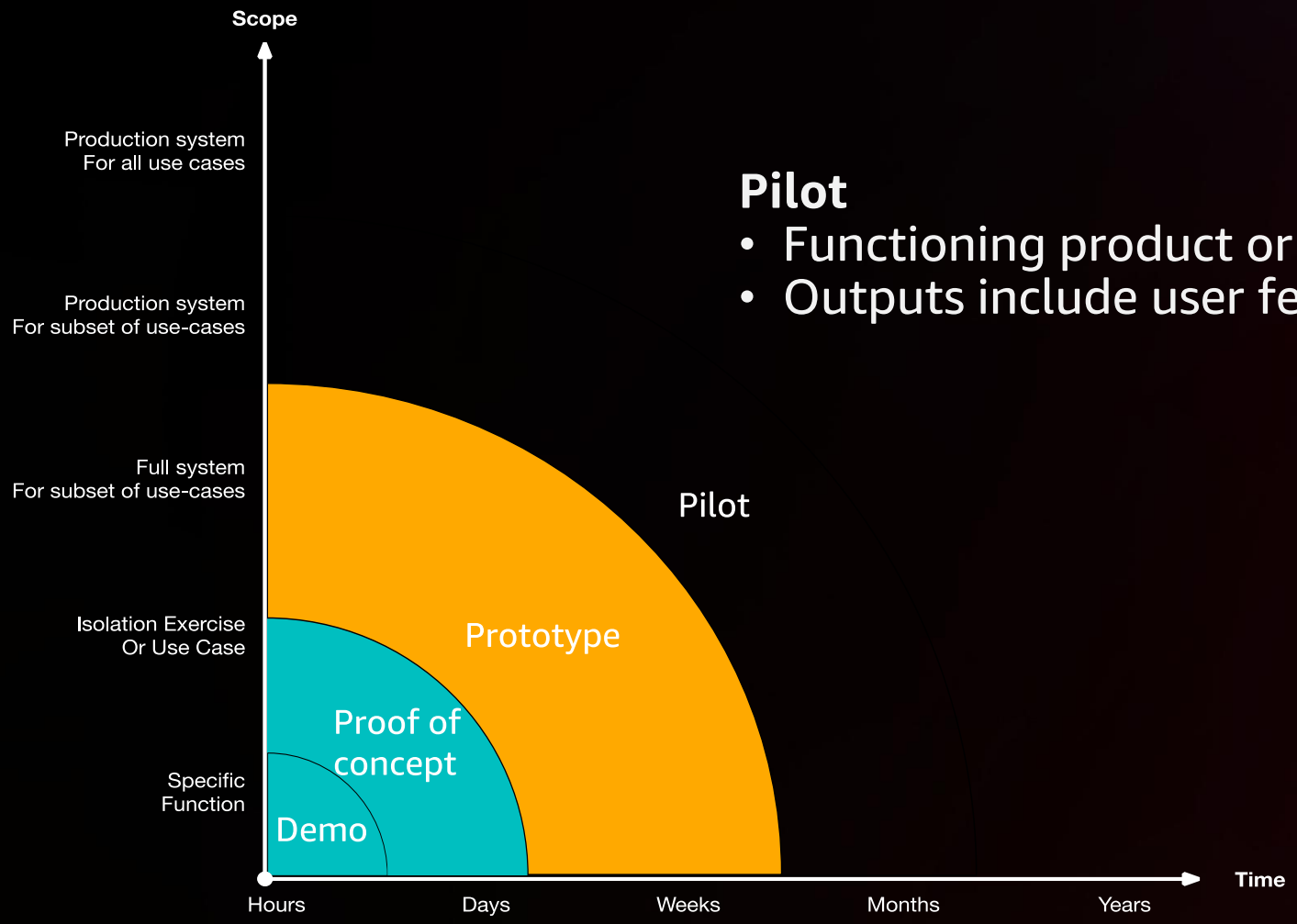


Prototype

- Technology within a business benefits framework
- Success is measured in terms of the business outcomes
- Demonstrable constrained user stories within a business context

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What is prototyping?

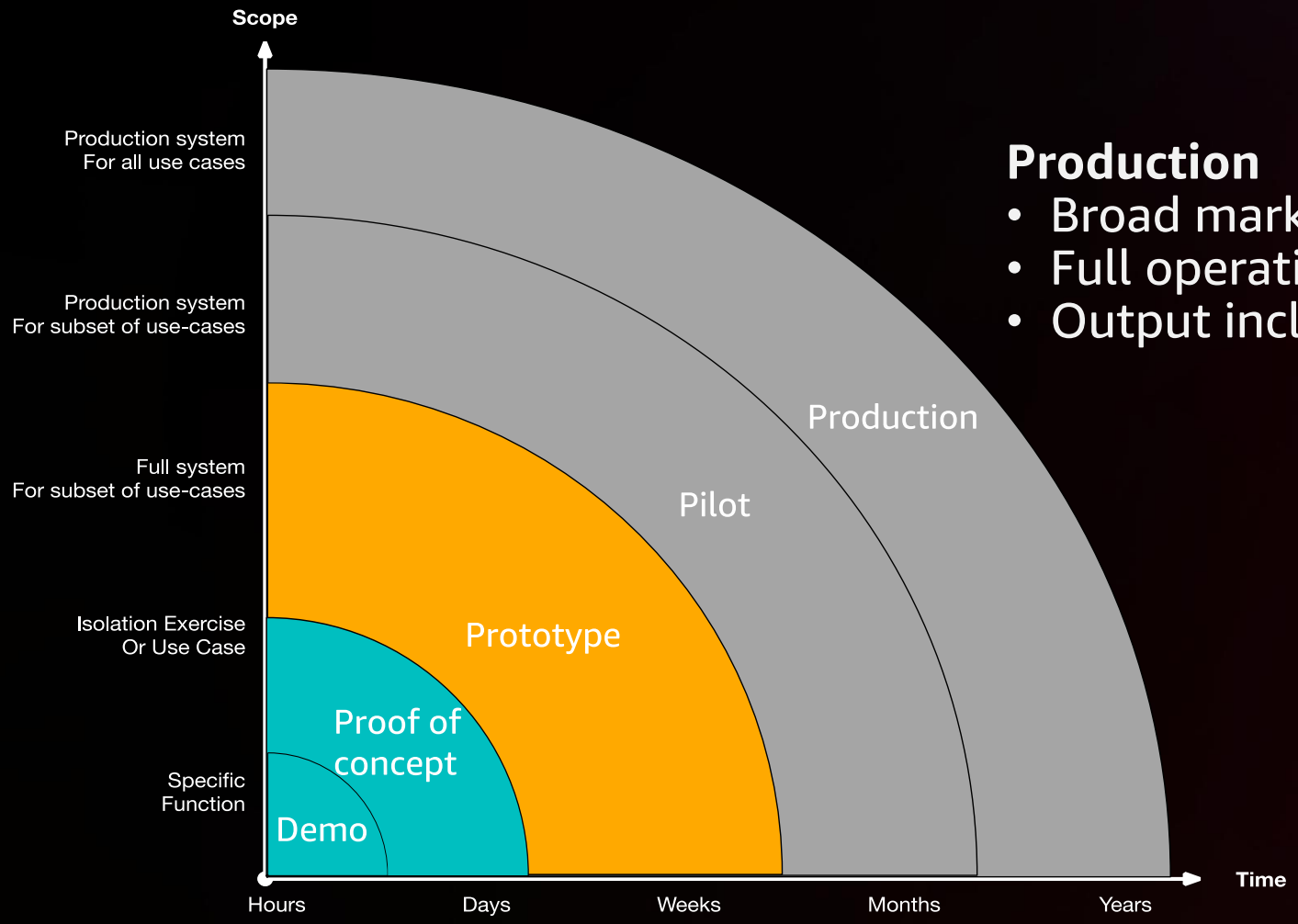


Pilot

- Functioning product or service released to a small set of users
- Outputs include user feedback

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What is prototyping?

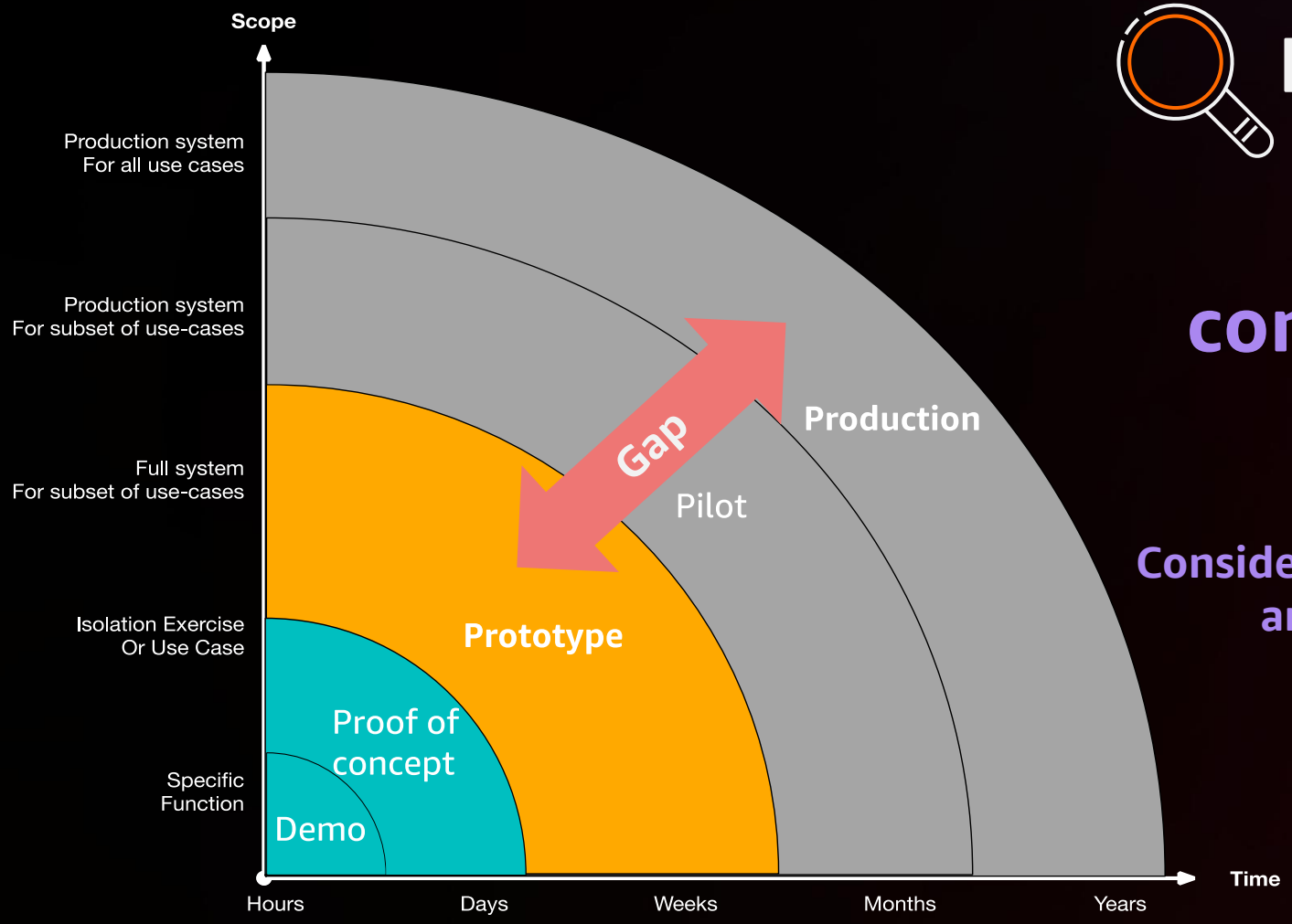


Production

- Broad market launch
- Full operational framework
- Output includes business metrics

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What is prototyping?



Prototyping concern

is a trade-off between

completeness and **velocity**

Consider non-functional areas as well

Develop quickly by focusing only on functions

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Effective prototyping approach: Practices

Modern application



Cloud-native approach

Agile/Scrum



Iterative approach

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

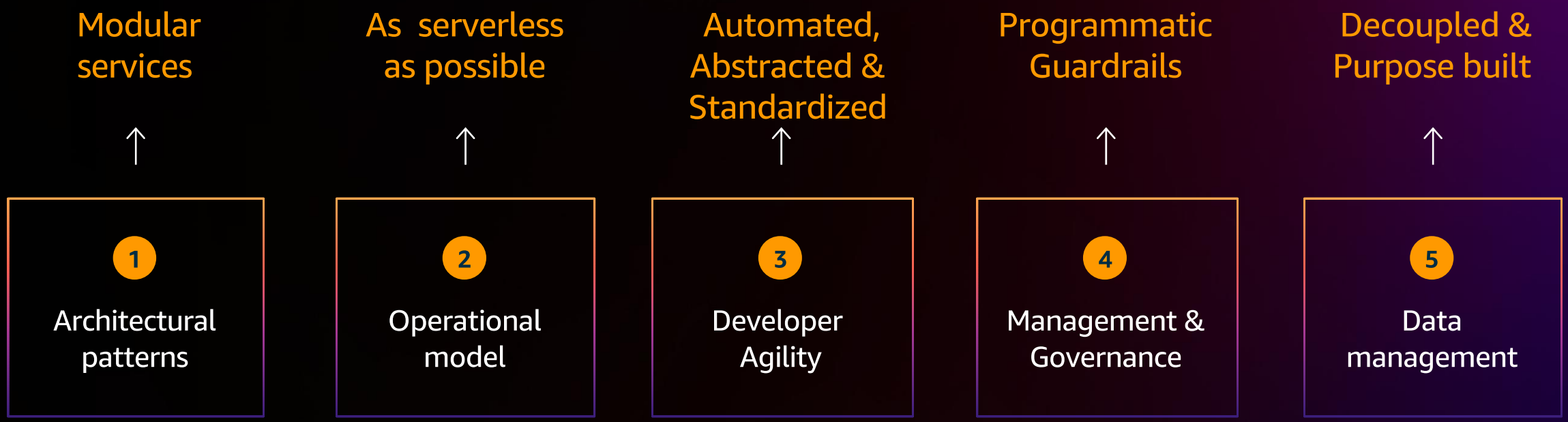
Effective prototyping approach: Cloud-native

1. Architectural patterns
2. Operational model
3. Developer Agility
4. Management & Governance
5. Data management

The best way to build a modern application

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Effective prototyping approach: Cloud-native



The best way to build a modern application

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Effective prototyping approach: Cloud-native


Amazon API Gateway


AWS Lambda


AWS CDK


Amazon Cognito


Amazon DynamoDB


Amazon SNS


Amazon Comprehend


AWS X-Ray


Amazon Kinesis Data Streams

Modular services

As serverless as possible

Automated, Abstracted & Standardized

Programmatic Guardrails

Decoupled & Purpose built

1 Architectural patterns

2 Operational model

3 Developer Agility

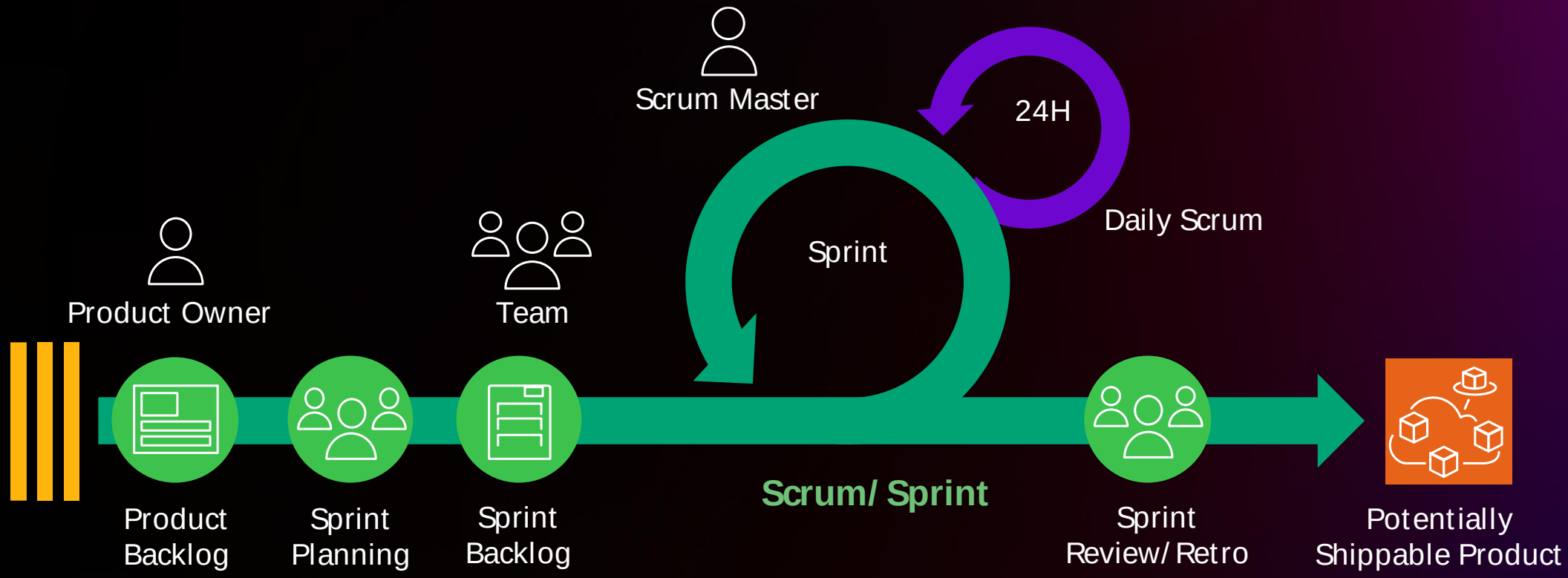
4 Management & Governance

5 Data management

The best way to build a modern application

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Effective prototyping approach: Iterative



Agile/Scrum

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Prototyping scenario

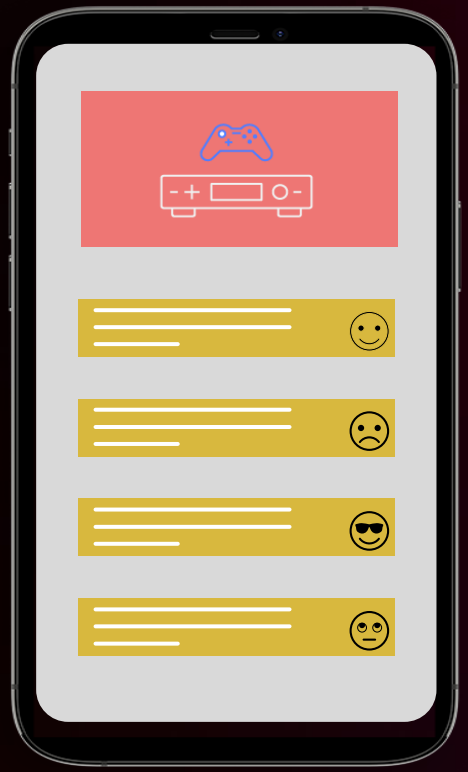
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

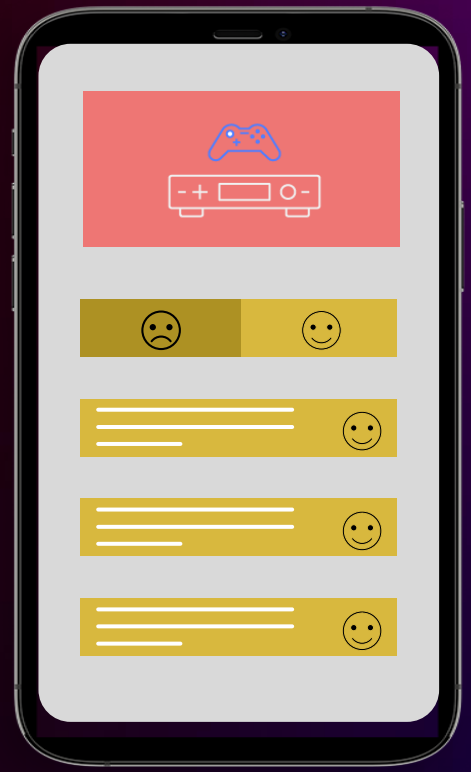
Requirement

Let’s provide a review service with real-time sentiment analysis, so that customers can get more information to make a purchase



Product owner





aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Demo and PoC

Demo: sentiment analysis



Amazon Comprehend



Amazon S3

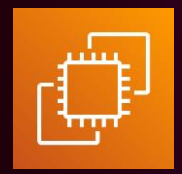


Amazon SageMaker

PoC: review request with sentiment analysis



Amazon Comprehend



Amazon EC2

Web framework + RDMS



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Demo and PoC

PoC: review request with sentiment analysis

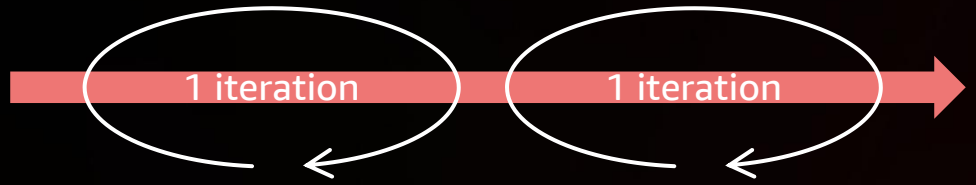


Amazon Comprehend



Amazon EC2

Web framework + RDMS



Will the prototype be ready for production within a month?

How can we guarantee our sentiment analysis or effect?



Product owner

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Build prototype: Architecture

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture: Review REST API

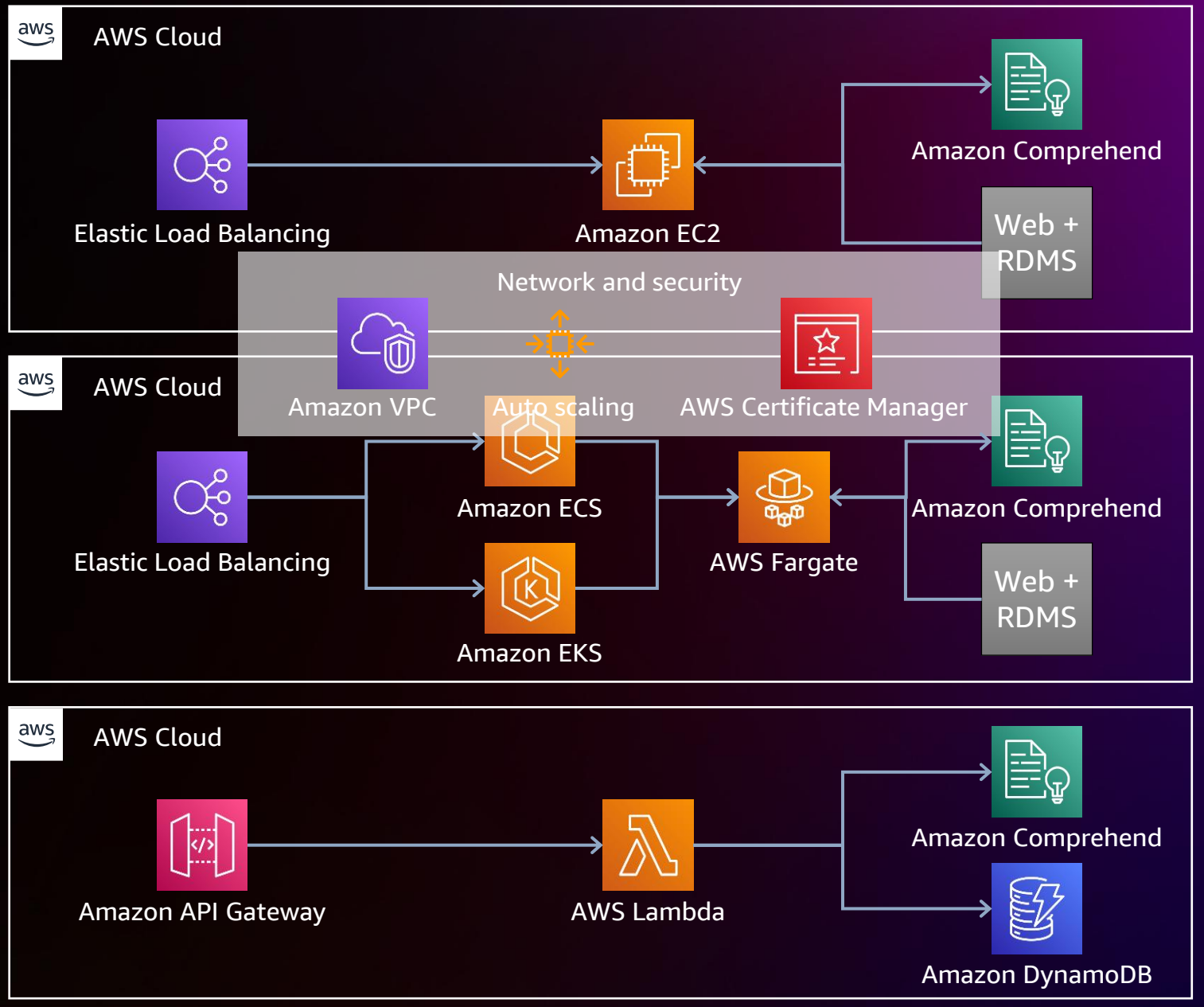


Web framework + RDMS

case1

case2

case3



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

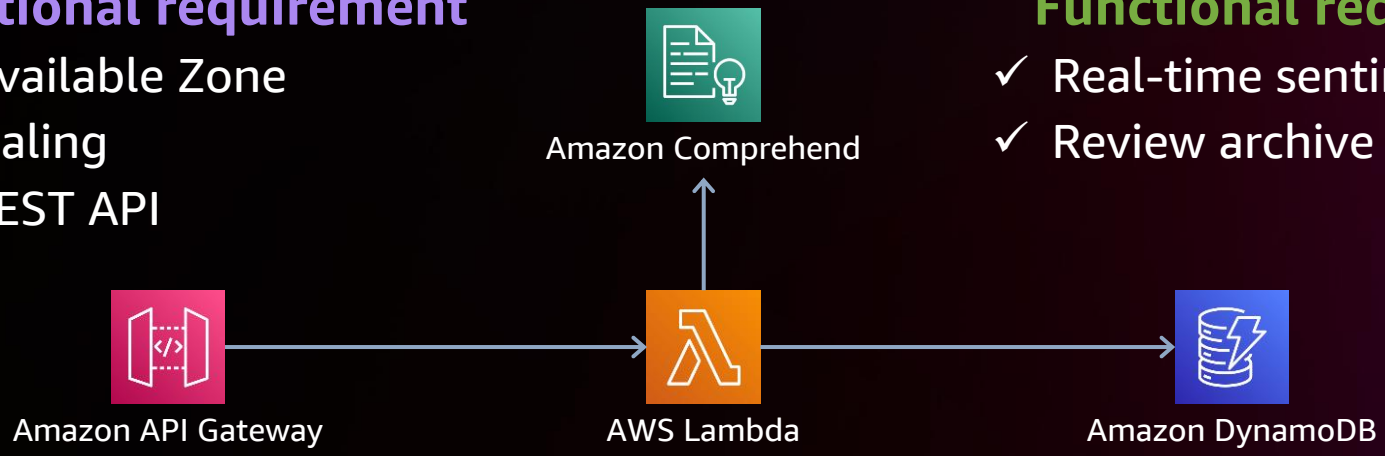
Architecture: Serverless review REST API

Non-functional requirement

- ✓ Multi-Available Zone
- ✓ Auto Scaling
- ✓ HTTS REST API

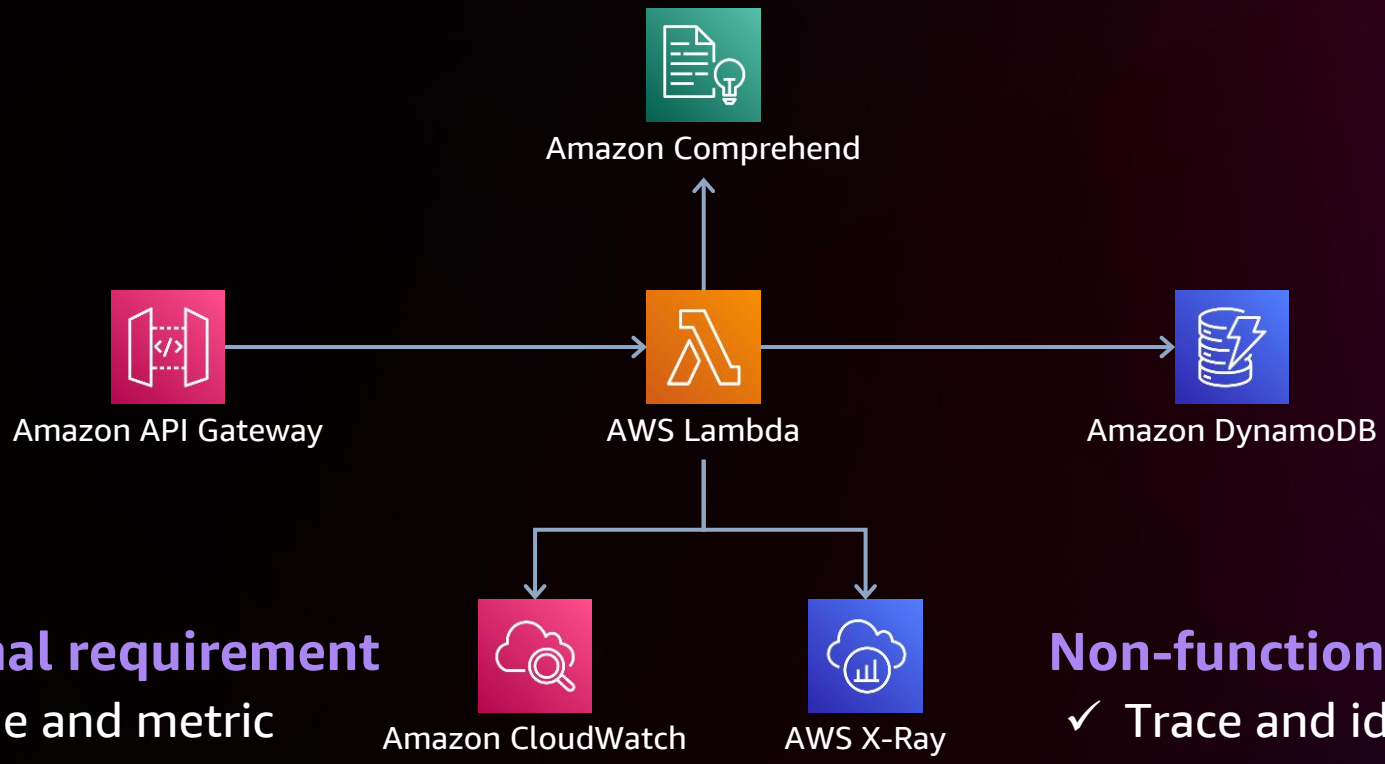
Functional requirement

- ✓ Real-time sentiment analysis
- ✓ Review archive (insert)



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture: Observable serverless REST API



Non-functional requirement

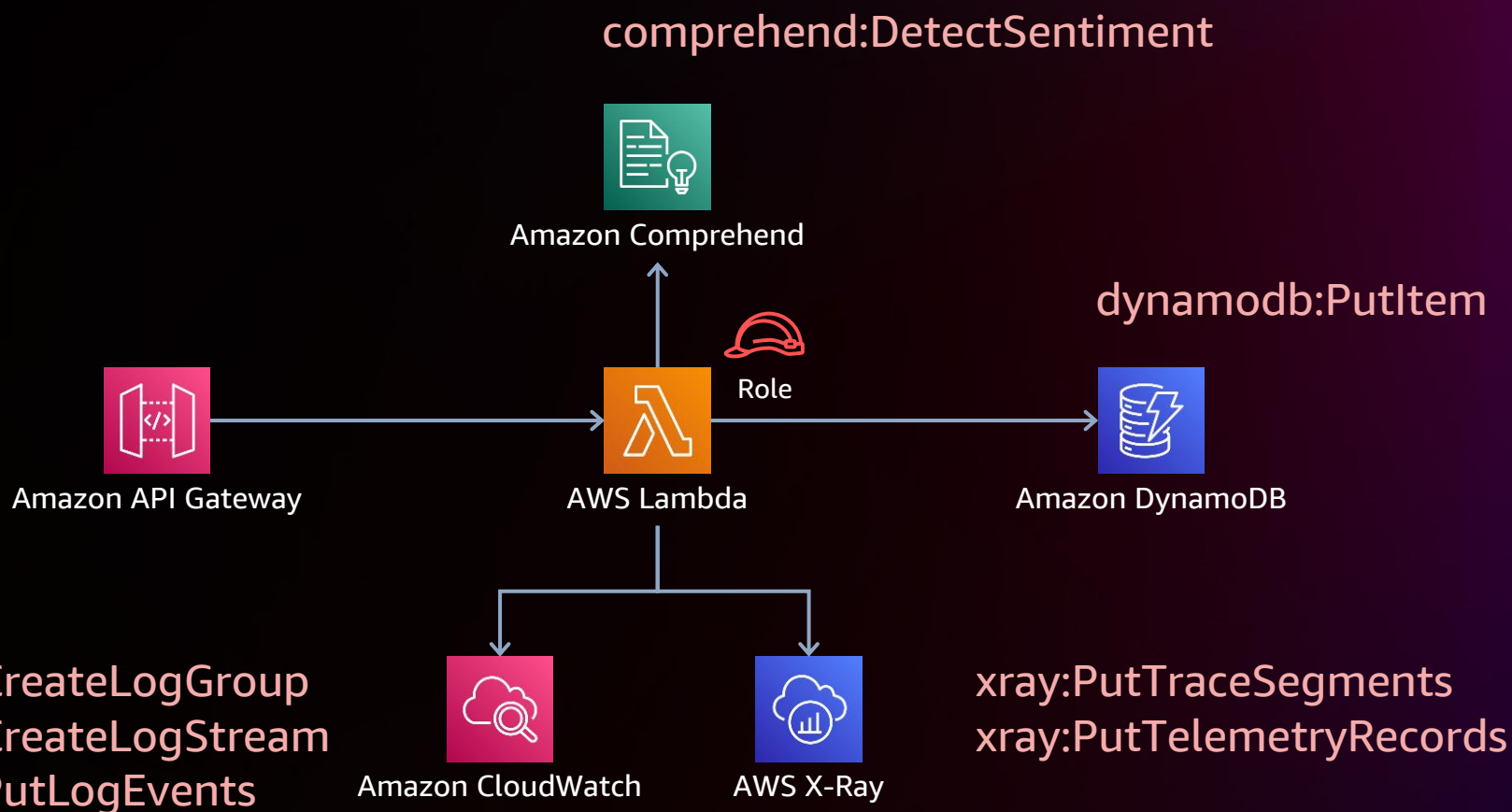
- ✓ Log message and metric
- ✓ Alarm and notification

Non-functional requirement

- ✓ Trace and identify bottlenecks

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture: Secure serverless REST API

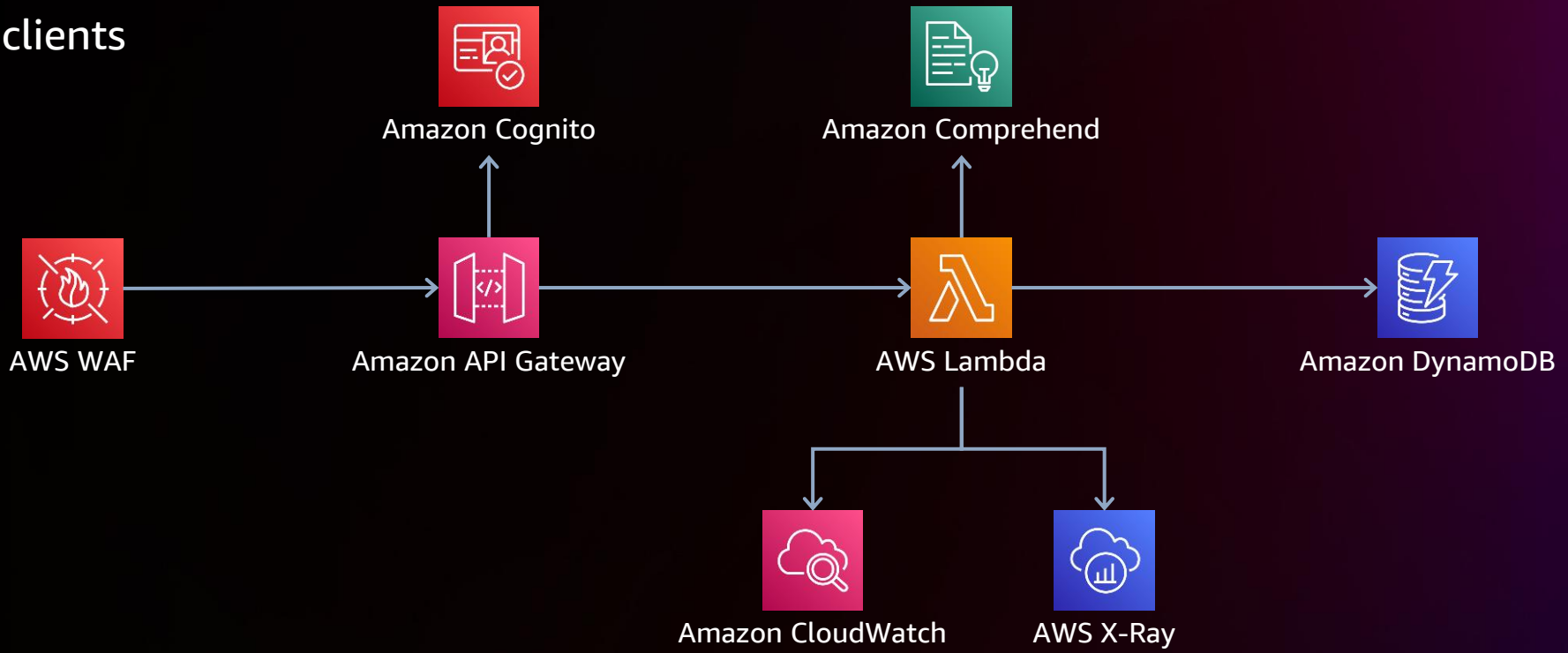


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture: Secure serverless REST API

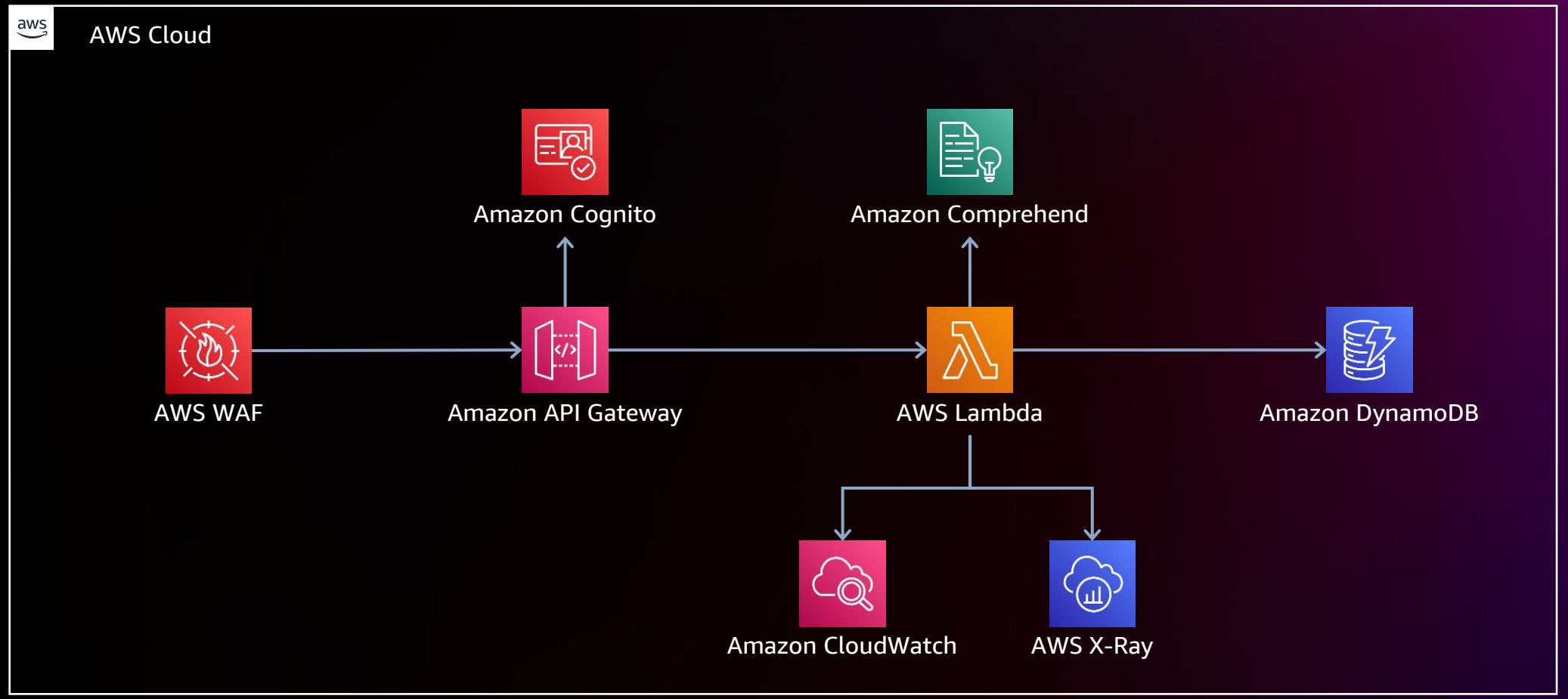
Non-functional requirement

- ✓ Protect API (exploits/bot)
- ✓ AuthN/Z clients



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture: Secure serverless REST API

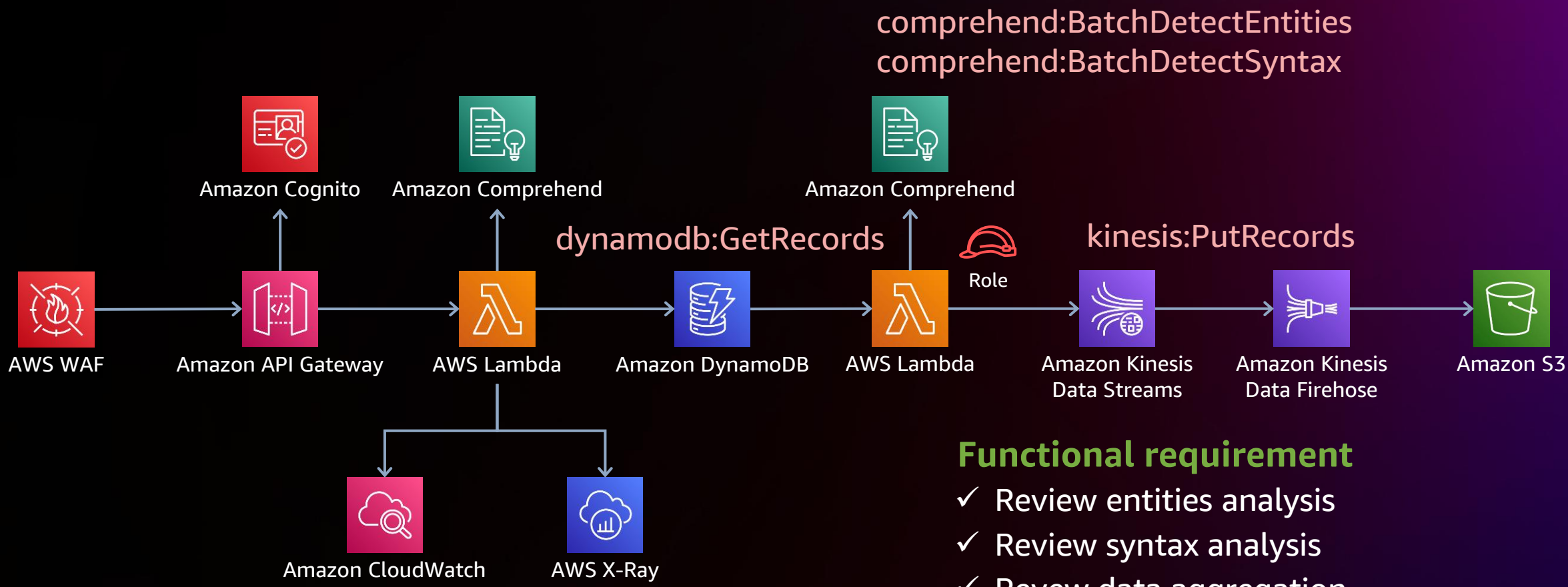


How can we guarantee our solution is ready for production within a month?

Will the pentest be ready?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture: Review stream-batch analysis



Functional requirement

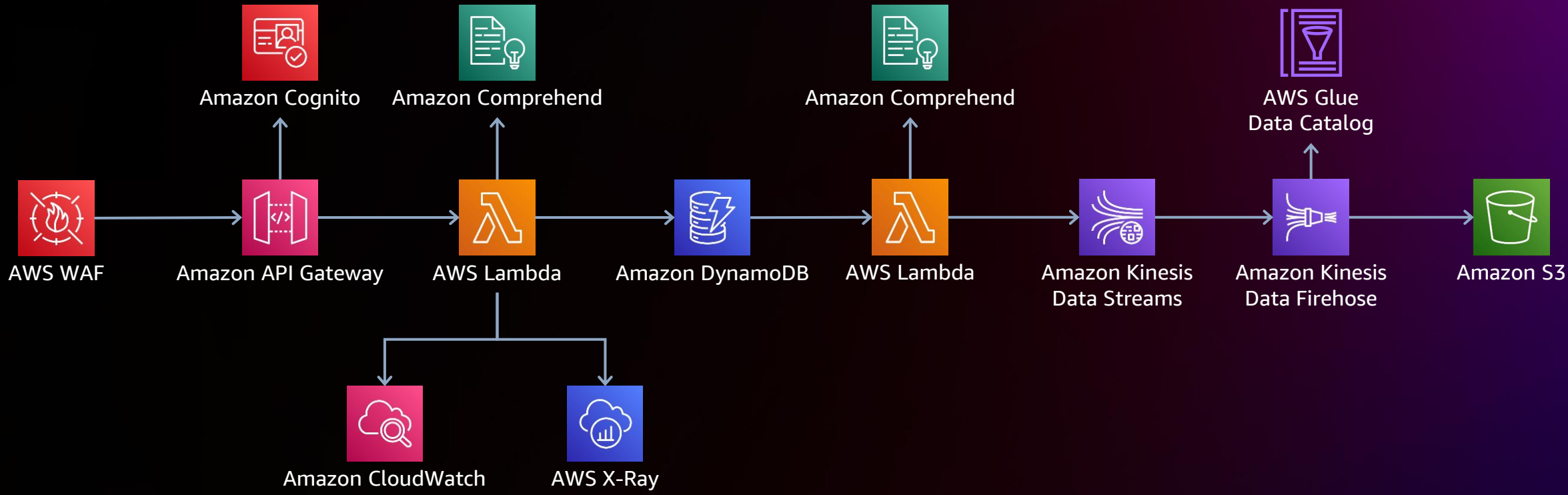
- ✓ Review entities analysis
- ✓ Review syntax analysis
- ✓ Revew data aggregation
- ✓ Analysis result ingestion
- ✓ Analysis result archive

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture: Review stream-batch analysis

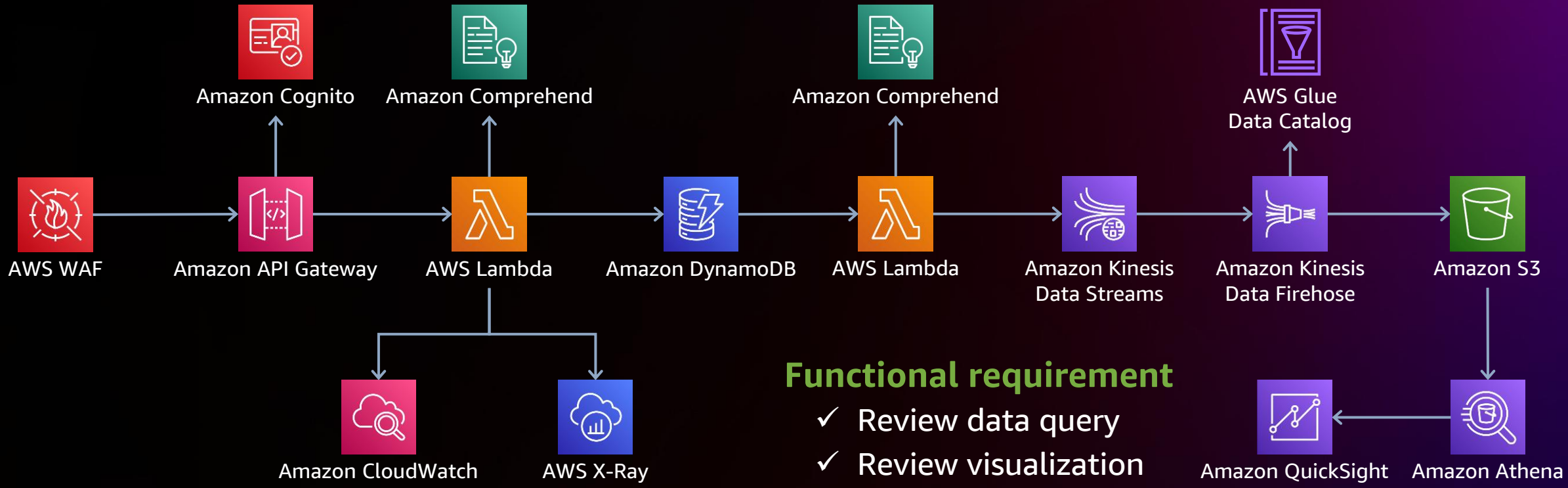
Non-functional requirement

- ✓ No coding
- ✓ High performance



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture: Review query and visualization

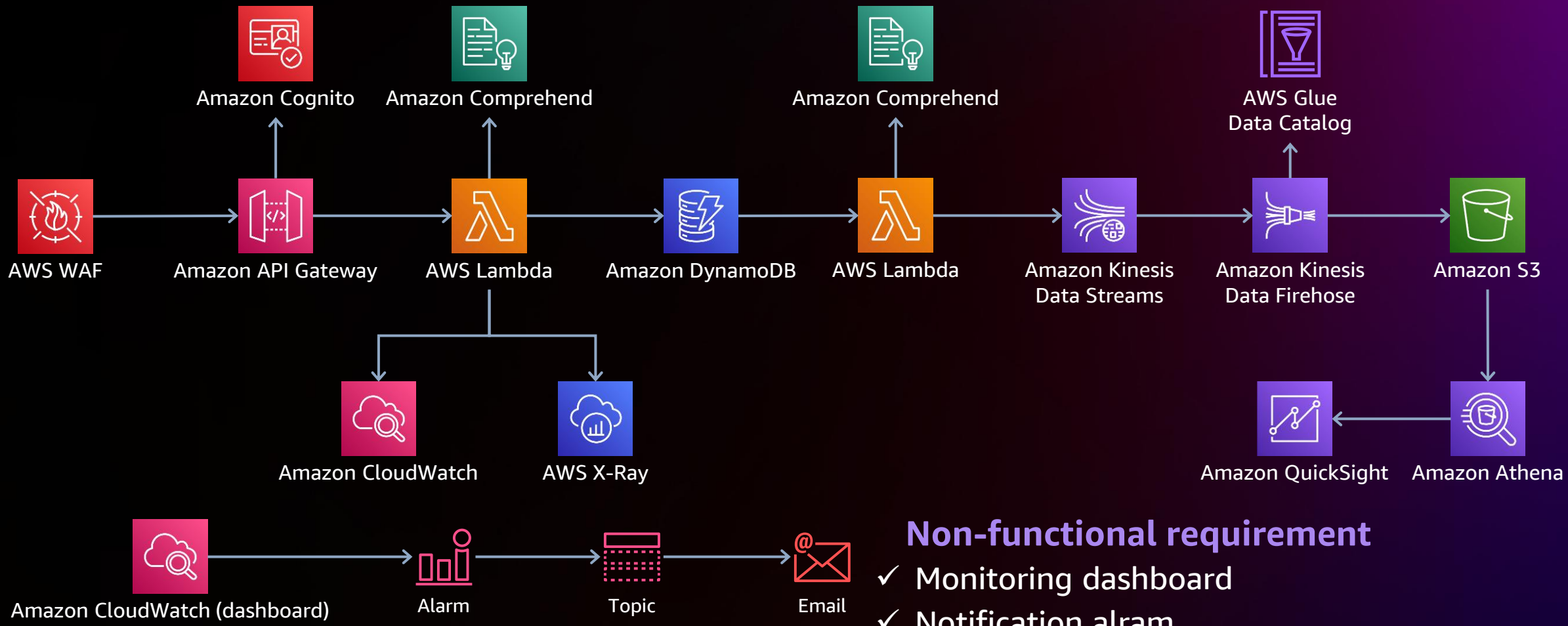


Functional requirement

- ✓ Review data query
- ✓ Review visualization
- ✓ Review dashboard

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture: Review service monitoring

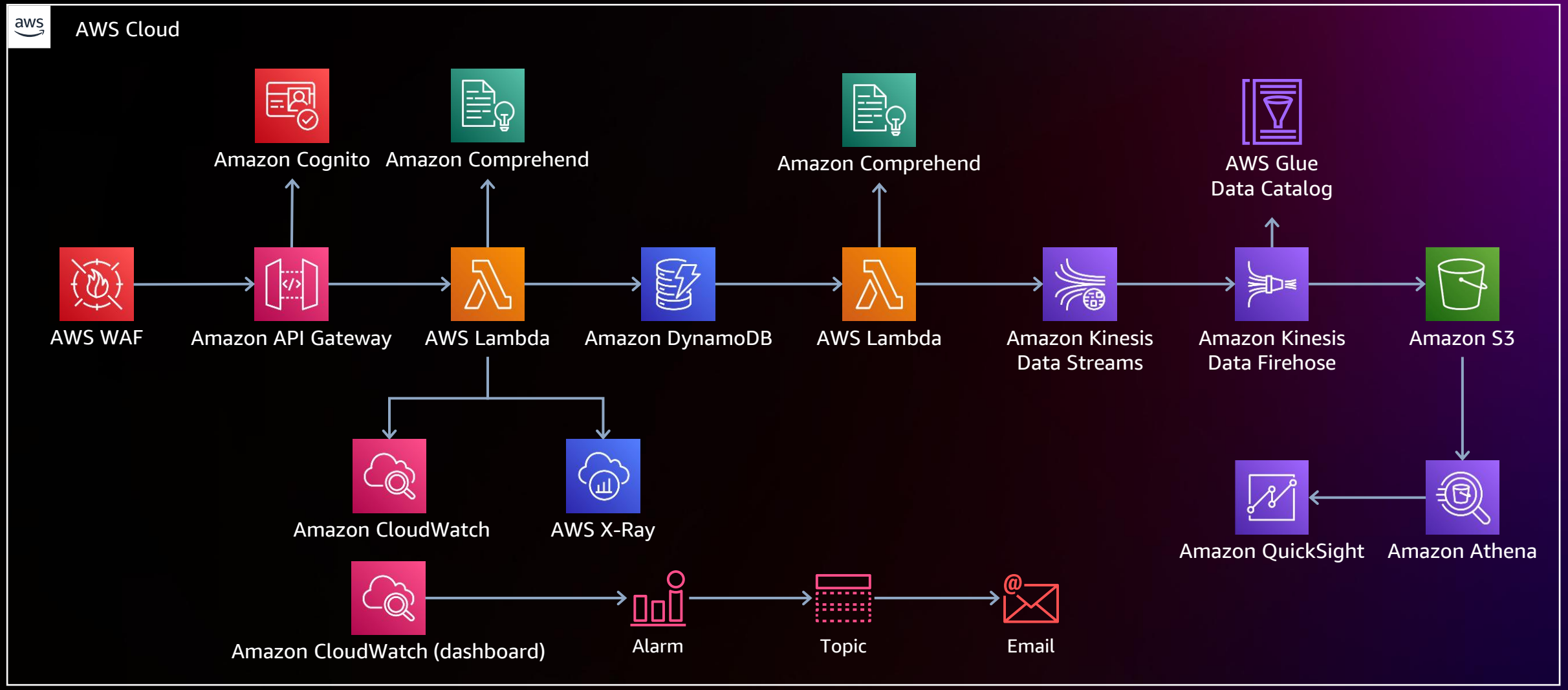


Non-functional requirement

- ✓ Monitoring dashboard
- ✓ Notification alram

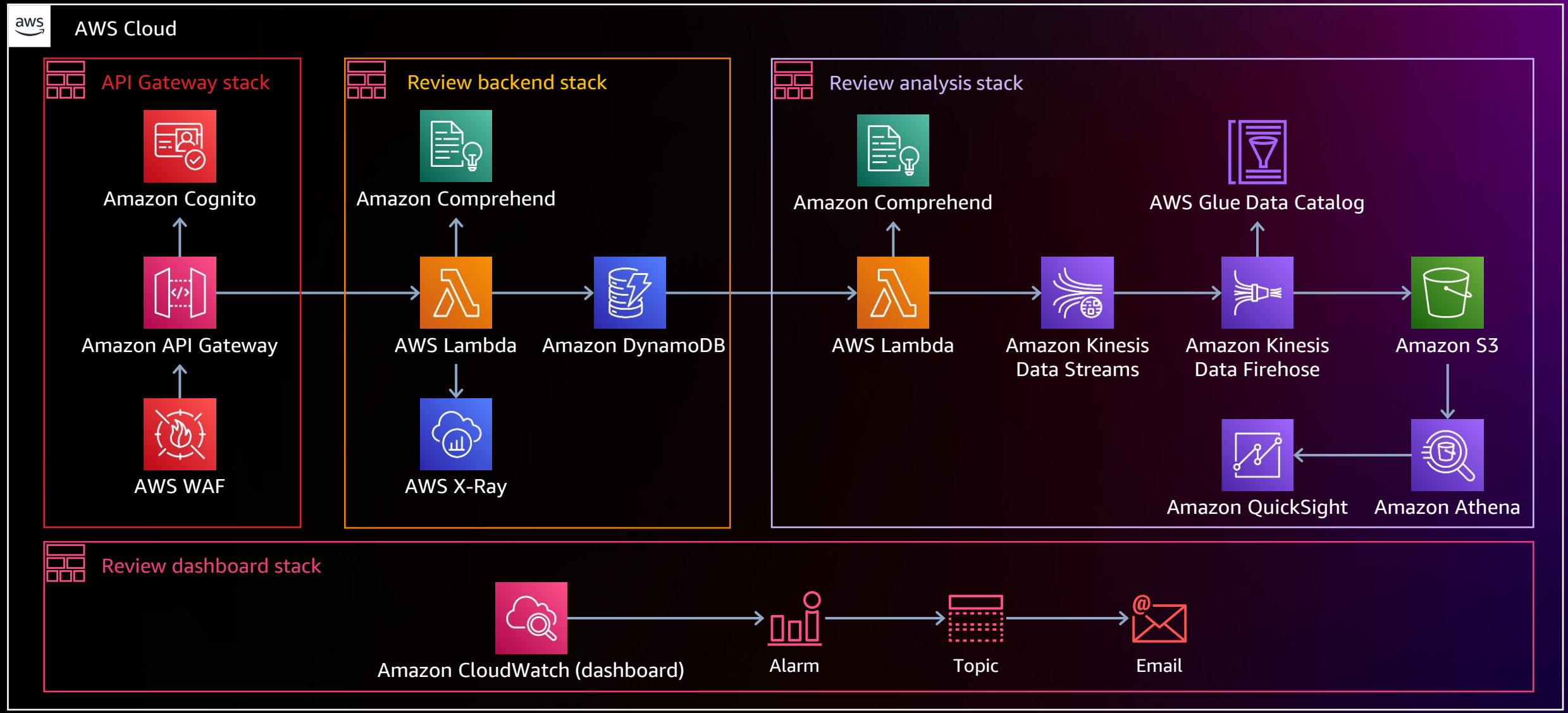
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture: Semi-final version



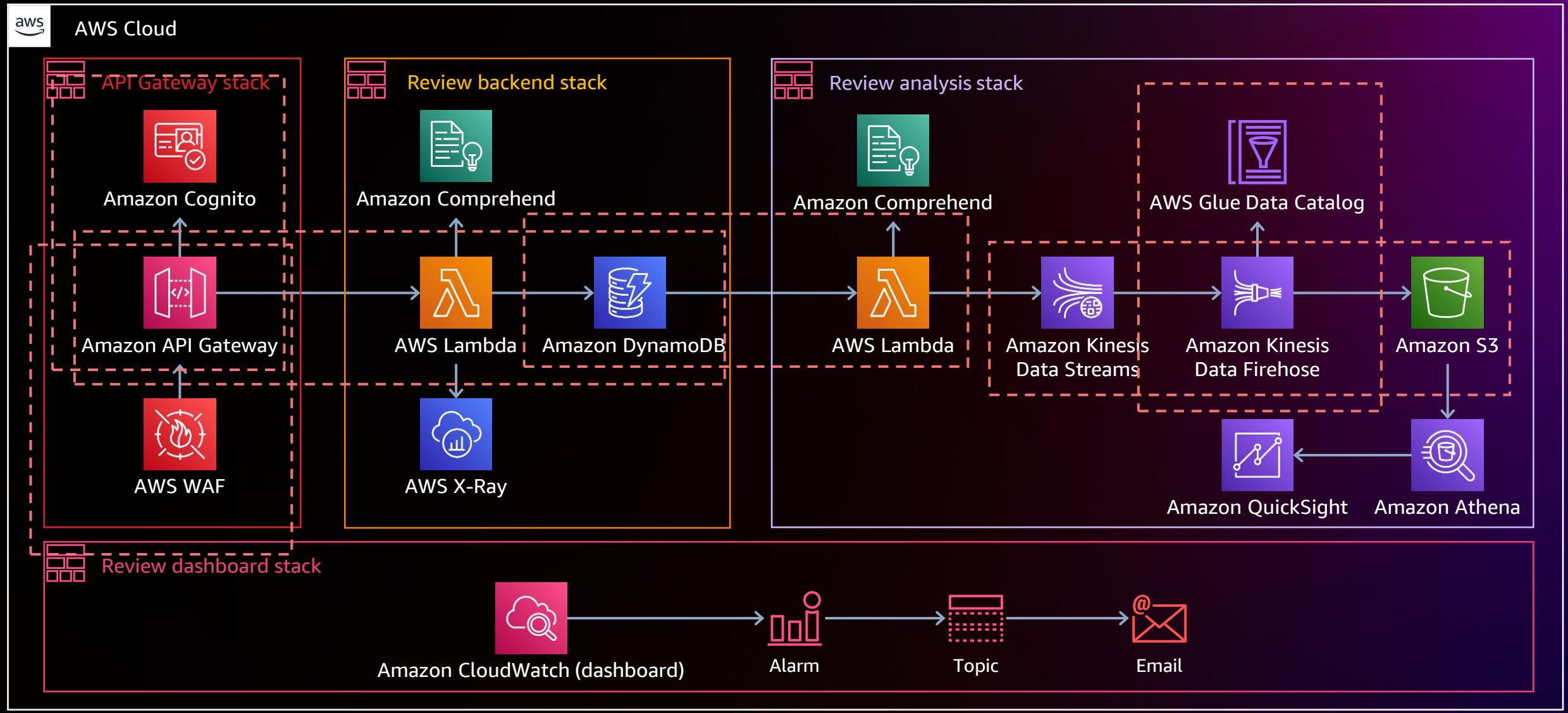
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture: Final version



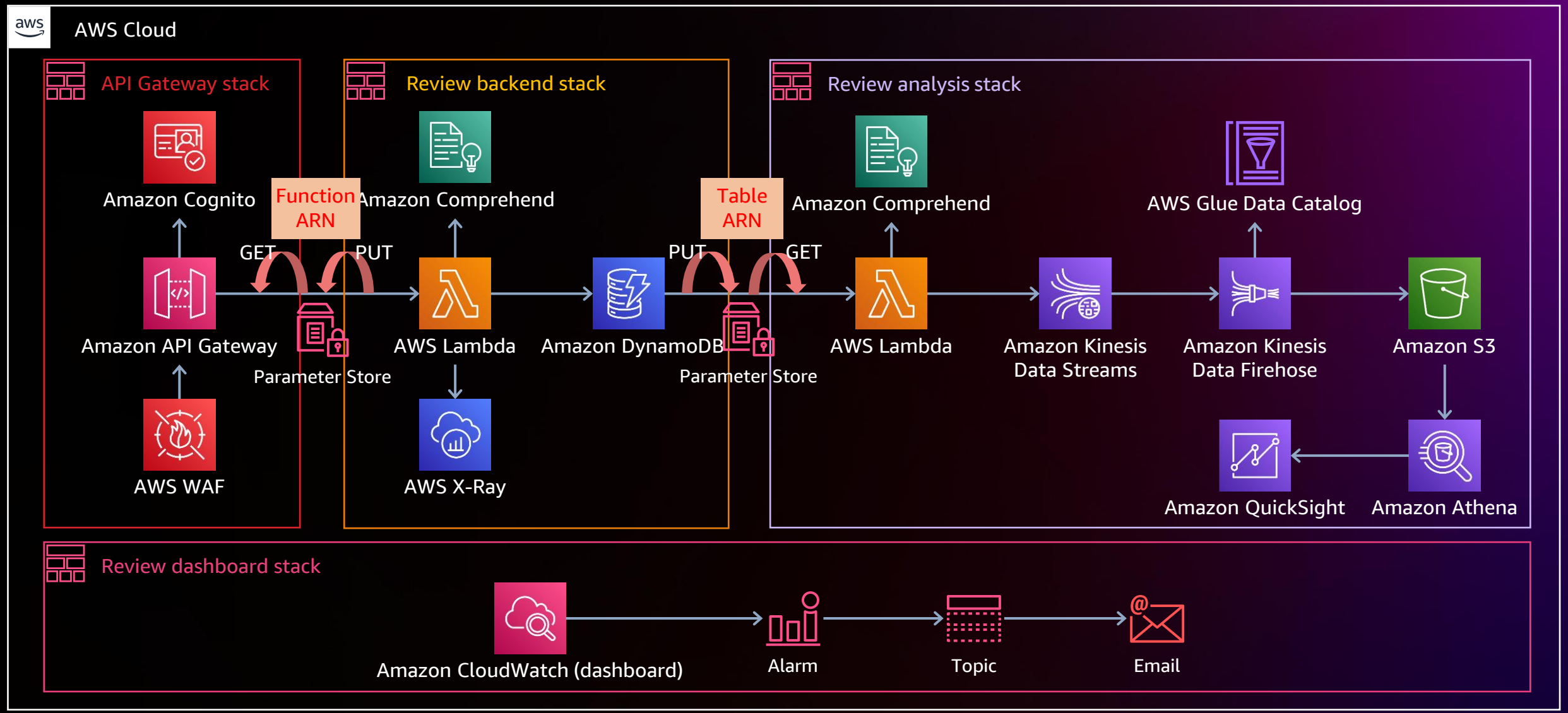
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture: Serverless patterns



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Architecture: Stack dependency



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Build prototype: Tool

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Tools



AWS CDK

aws / aws-cdk

https://github.com/aws/aws-cdk
https://docs.aws.amazon.com/cdk/api/v2/

awslabs / aws-solutions-constructs

https://github.com/awslabs/aws-solutions-constructs
https://docs.aws.amazon.com/solutions/latest/constructs/welcome.html

aws-samples / aws-cdk-project-template-for-devops

https://github.com/aws-samples/aws-cdk-project-template-for-devops



AWS SDKs

boto / boto3

https://github.com/boto/boto3
https://boto3.amazonaws.com/v1/documentation/api/latest/index.html

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Tools: IaC for serverless

Server



Web servers

Presentation layers



Application servers

Business logic

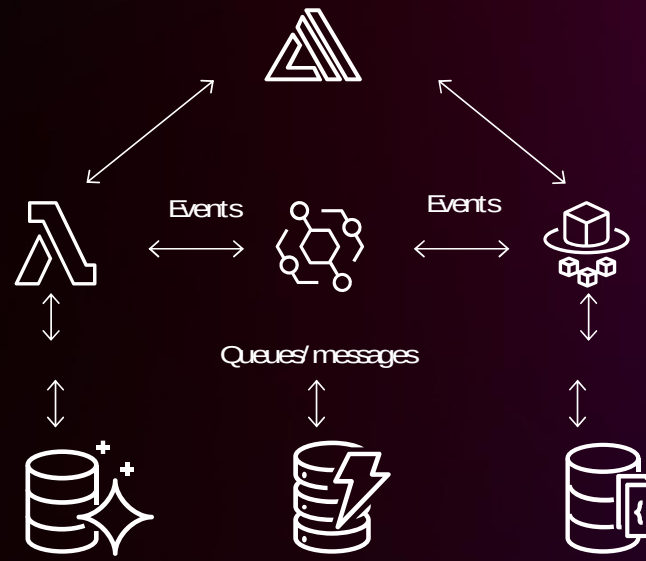


Database servers

Data layer

Infrastructure Configuration/ Orchestration

Serverless



Presentation

Business logic

Data

Cloud Resource Management

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Tools: IaC for serverless

IaC on AWS

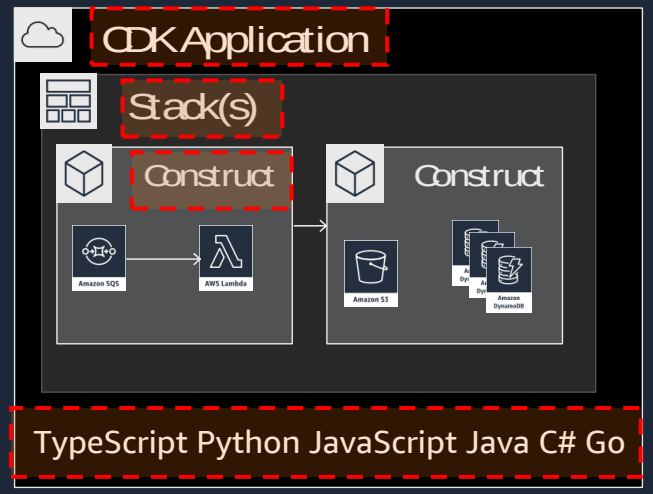
- **AWS CloudFormation**: a JSON or YAML formatted text file, the most mature and widely used
- **AWS Serverless Application Model (AWS SAM):** an extension of AWS CloudFormation, shorthand syntax to express serverless sources such as functions, APIs, databases, and event source mappings
- **AWS Cloud Development Kit (AWS CDK**): familiar OOP programming languages, high-level components called constructs that preconfigure cloud resources with proven defaults



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS CDK components

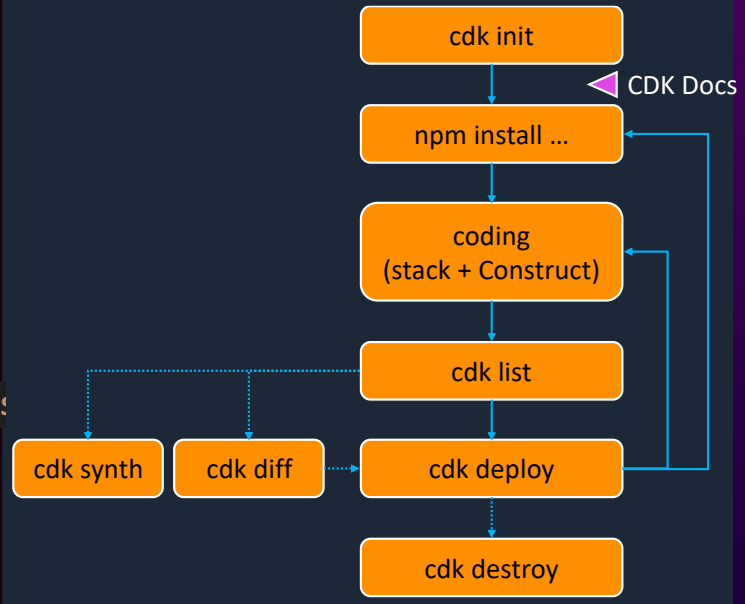
Core Framework



AWS Construct Library

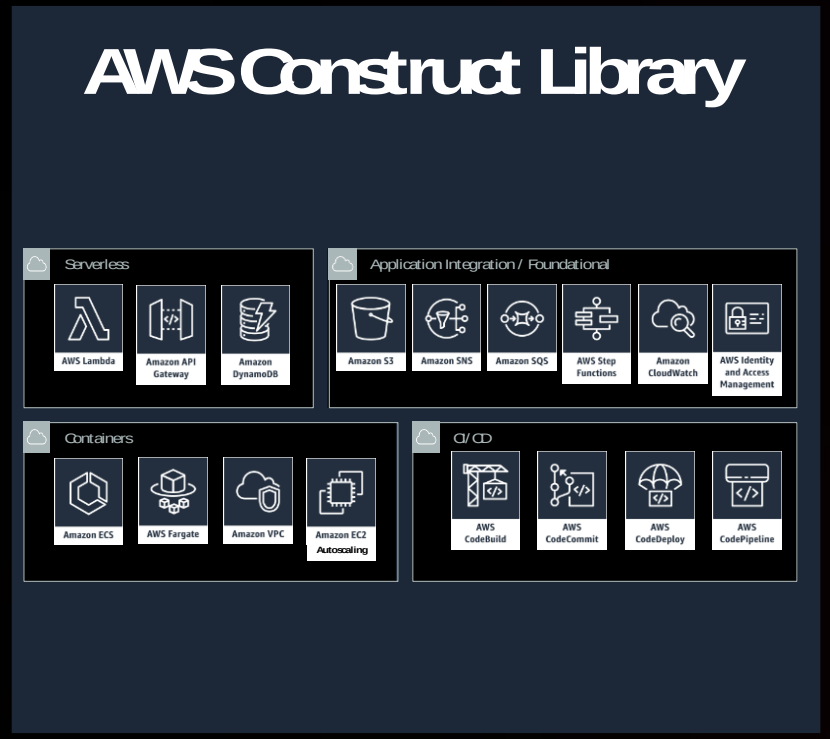
```
new ec2.Vpc(this, 'BaseVPC',
new lambda.Function(this, 'func',
new ecs.FargateTaskDefinition(this, 'def',
new rds.ServerlessCluster(this, 'serverless
```

AWS CDK CLI



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS CDK components: AWS Construct Library




```
import { Construct } from 'constructs';
import * as cdk from 'aws-cdk-lib';
import * as ecs from 'aws-cdk-lib/aws-ecs';
import * as ecsPatterns from 'aws-cdk-lib/aws-ecs-patterns';

export class EcsSampleStack extends cdk.Stack {

    constructor(scope: Construct, id: string,) {
        super();

        new ecs.CfnService(scope, 'Layer-1', {...});

        new ecs.Ec2Service(scope, 'Layer-2', {...});

        new ecsPatterns.ApplicationLoadBalancedEc2Service(scope, 'Layer-3', {...});
    }
}
```

L1 CloudFormation Resources

L2 AWS Constructs

L3+ Purpose-built constructs

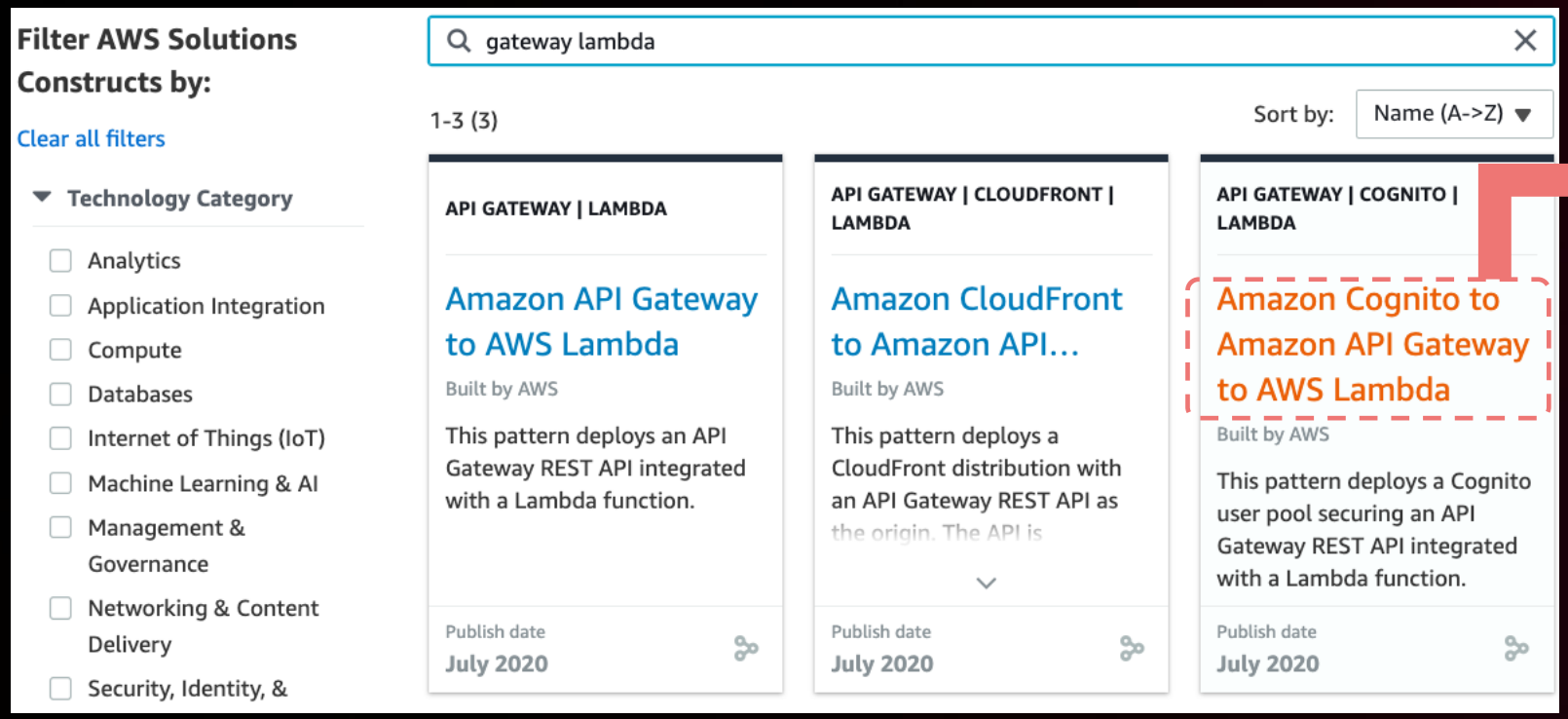
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

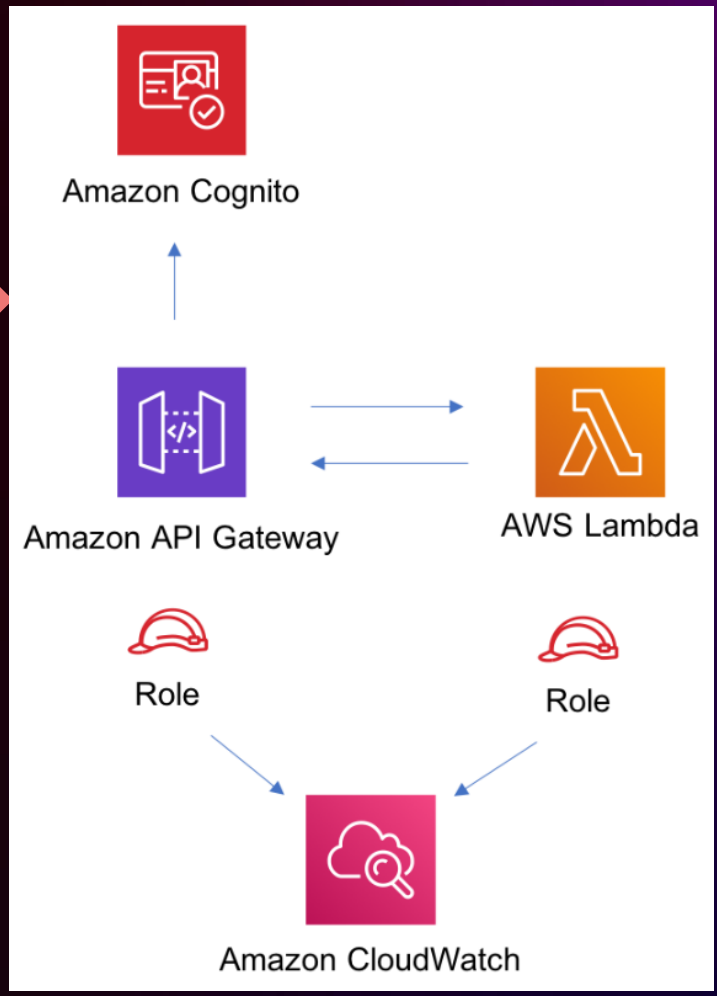
AWS CDK components: Solutions Constructs

awslabs / aws-solutions-constructs

https://github.com/awslabs/aws-solutions-constructs
https://docs.aws.amazon.com/solutions/latest/constructs/welcome.html

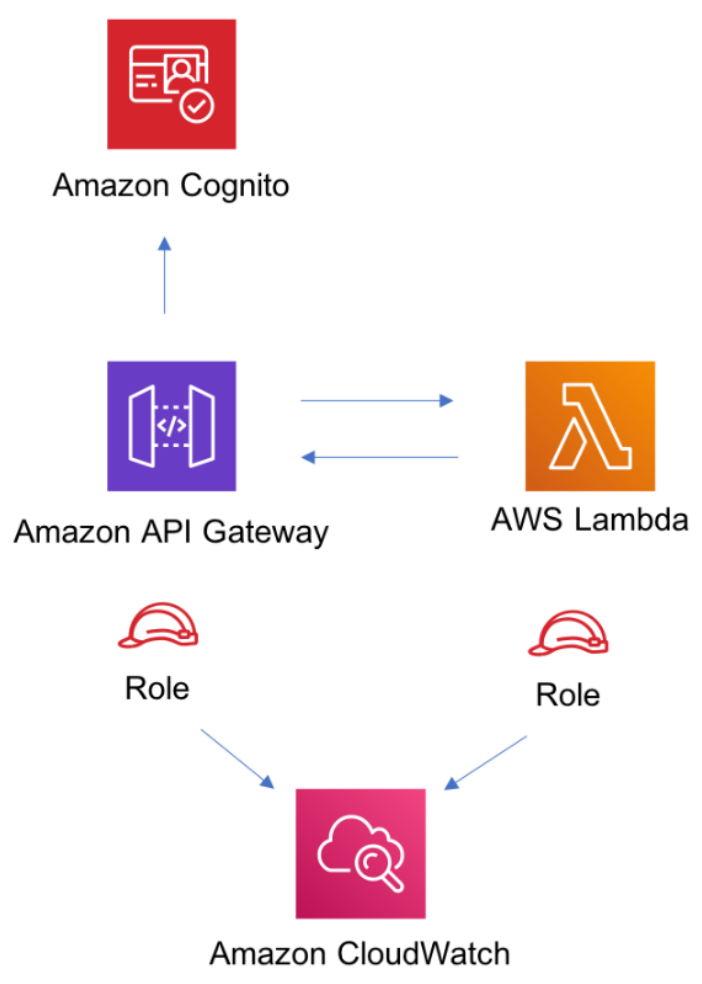
L3+ Purpose-built constructs





© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS CDK components: Solutions Constructs




```
new CognitoToApiGatewayToLambda(this, 'Layer-3', {
    lambdaFunctionProps: {
        code: lambda.Code.fromAsset(`lambda`),
        runtime: lambda.Runtime.NODEJS_16_X,
        handler: 'index.handler'
    }
});
```

AWS Solutions Constructs

```
const func = new lambda.Function(this, 'func', {
    code: lambda.Code.fromAsset(`lambda`),
        runtime: lambda.Runtime.NODEJS_16_X,
        handler: 'index.handler'
});

const pool = new cognito.UserPool(this, 'pool');
new cognito.UserPoolClient(this, 'client', {
    userPool: pool
});

new gateway.LambdaRestApi(this, 'rest', {
    handler: func
});
new gateway.CognitoUserPoolsAuthorizer(this, 'auth', {
    cognitoUserPools: [pool]
});
```

AWS CDK Construct

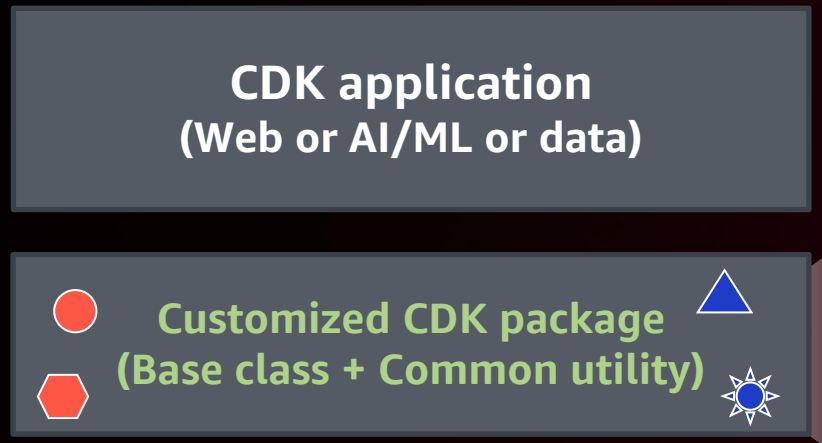
aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS CDK: Customized package

aws-samples / aws-cdk-project-template-for-devops

https://github.com/aws-samples/aws-cdk-project-template-for-devops





- Configuration injection
- multi-target environment injection
- Prefix-naming management
- Stack dependency management
- Common pattern resources





© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS CDK: Technical debt (config-hard coding)

```
const stage = 'dev';
// const stage = 'prd';
```

```
const app = new App();

const prefix = 'ReviewServiceDev';
// const prefix = 'ReviewServicePrd';
```

```
new MyFirstStack(app, `${prefix}-hello-cdk`, {
    env: {
        account: '111111111111',
        region: 'eu-central-1'
    }
    // env: {
    //     account: '222222222222',
    //     region: 'us-east-1'
    // }
});
```

```
class MyFirstStack extends Stack {
    constructor(scope: Construct, id: string, props: StackProps) {
        super(scope, id);
        new lambda.Function(this, 'lambda-func', {
            runtime: lambda.Runtime.PYTHON_3_9,
            handler: 'handler.handle',
            code: lambda.Code.fromAsset('codes/ver1')
            // code: lambda.Code.fromAsset('codes/ver2')
        });
    }
}
```

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS CDK: Technical debt (config-hard coding)

```
import { App, Stack, StackProps } from 'aws-cdk-lib'
import { Construct } from 'constructs'
import * as lambda from 'aws-cdk-lib/aws-lambda'

class MyFirstStack extends Stack {
    constructor(scope: Construct, id: string, props: StackProps) {
        super(scope, id);
        new lambda.Function(this, 'lambda-func', {
            runtime: lambda.Runtime.PYTHON_3_9,
            handler: 'handler.handle',
            code: lambda.Code.fromAsset('codes/ver1')
            // code: lambda.Code.fromAsset('codes/ver2')
        });
    }
}

const app = new App();

const prefix = 'ReviewServiceDev';
// const prefix = 'ReviewServicePrd';

new MyFirstStack(app, `${prefix}-hello-cdk`, {
    env: {
        account: '111111111111',
        region: 'eu-central-1'
    }
    // env: {
    //     account: '222222222222',
    //     region: 'us-east-1'
    // }
});
```

Abstract extract

Development configuration

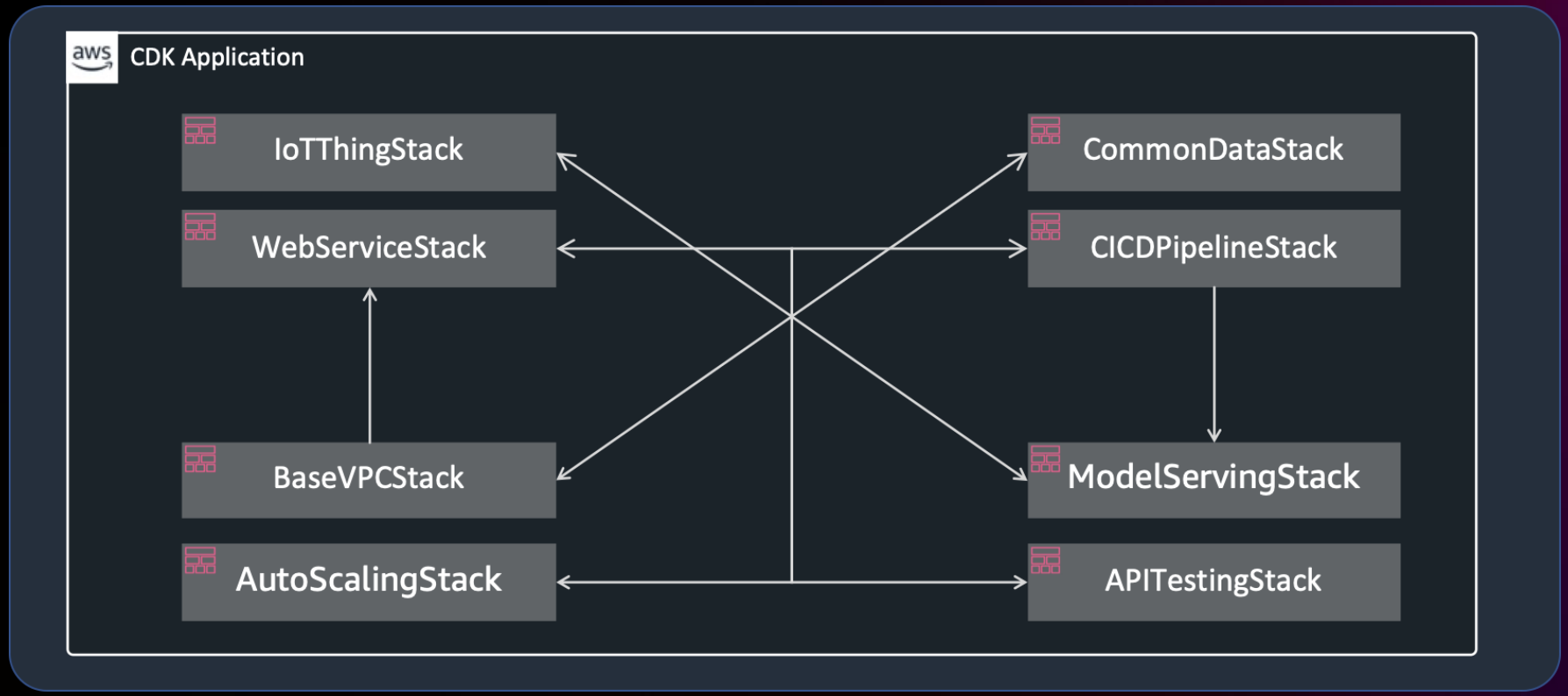
- target environment
- lambda code path
- lambda memory size
- Name Prefix

Production configuration

- target environment
- lambda code path
- lambda memory size
- Name Prefix

aws © 20

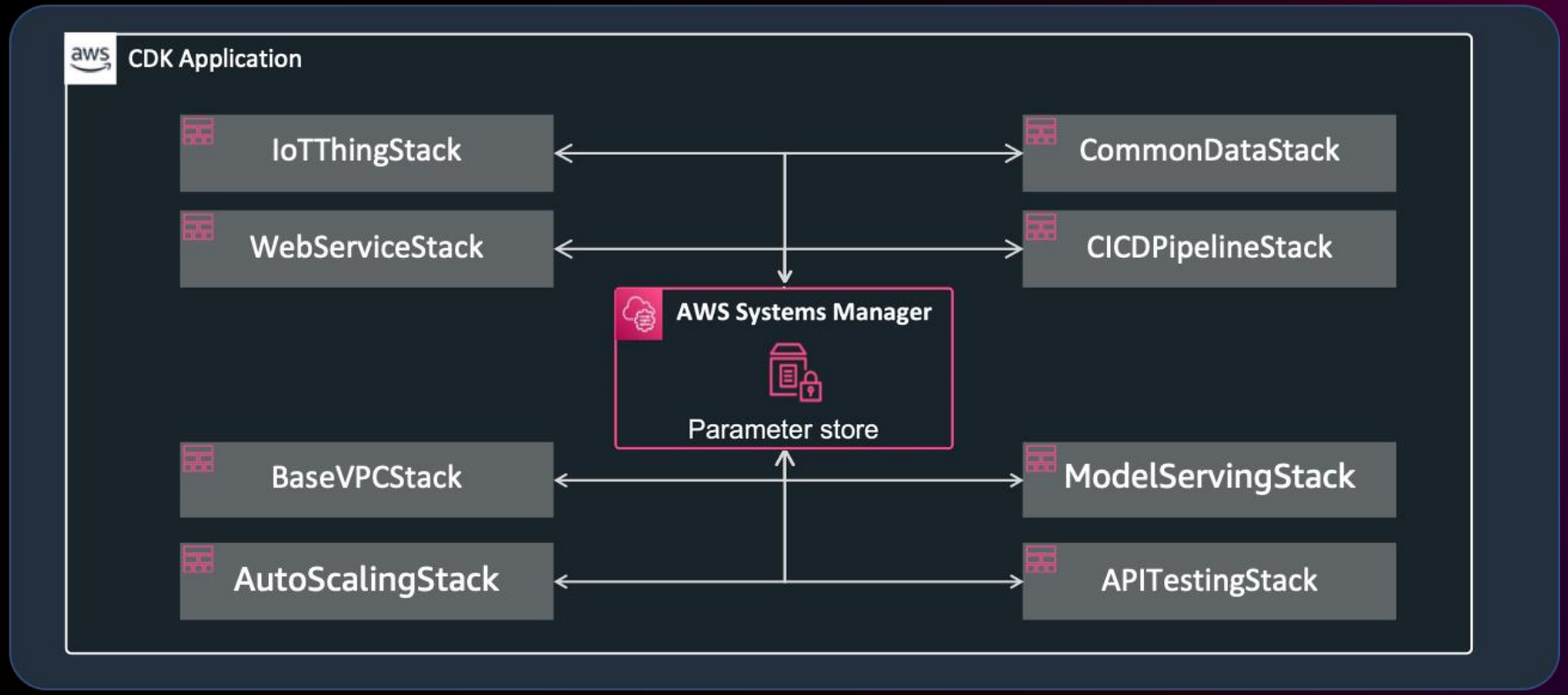
AWS CDK: Stack dependency



Direct dependency – tightly coupled

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS CDK: Stack dependency



Indirect dependency – loosely coupled

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS CDK: Customized CDK package

aws-samples / aws-cdk-project-template-for-devops

https://github.com/aws-samples/aws-cdk-project-template-for-devops

app-config-prd.json

app-config-dev.json

```
{
    "Project": {
        "Name": "ReviewService",
        "Stage": "Dev",
        "Account": "111111111111",
        "Region": "eu-central-1",
        "Profile": "cdk-v2"
    },

    "Stack": {
        "ReviewBackend": {
            "Name": "ReviewBackendStack",

            "LambdaFuncName": "BackendFunc",
            "LambdaFuncMemory": 128,
            "LambdaFuncCode": "codes/ver1",
            "LambdaFuncHandler": "handler.handle",

            "DdbTableName": "ReviewHistoryTable",
            "DdbTablePartitionKey": "ProductId",
            "DdbTableSortKey": "ReviewId"
        }
    }
}
```

Multi-target environment injection

Prefix-naming management

Stack name
ReviewServicePrd-Review
ReviewServicePrd-ApiGat
ReviewServicePrd-ReviewBackendStack

Stack name
ReviewServiceDev-ApiGatewayStack
ReviewServiceDev-ReviewAnalysisStack
ReviewServiceDev-ReviewBackendStack

Configuration injection

BaseStack

+ stackConfig: json

+ withStackName(baseName)
+ withProjectPrefix(baseName)
+ createS3BucketName(baseName)

+ getParameter(key)
+ putParameter(key, value)

Stack dependency management

aws

Build prototype: Implementation

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

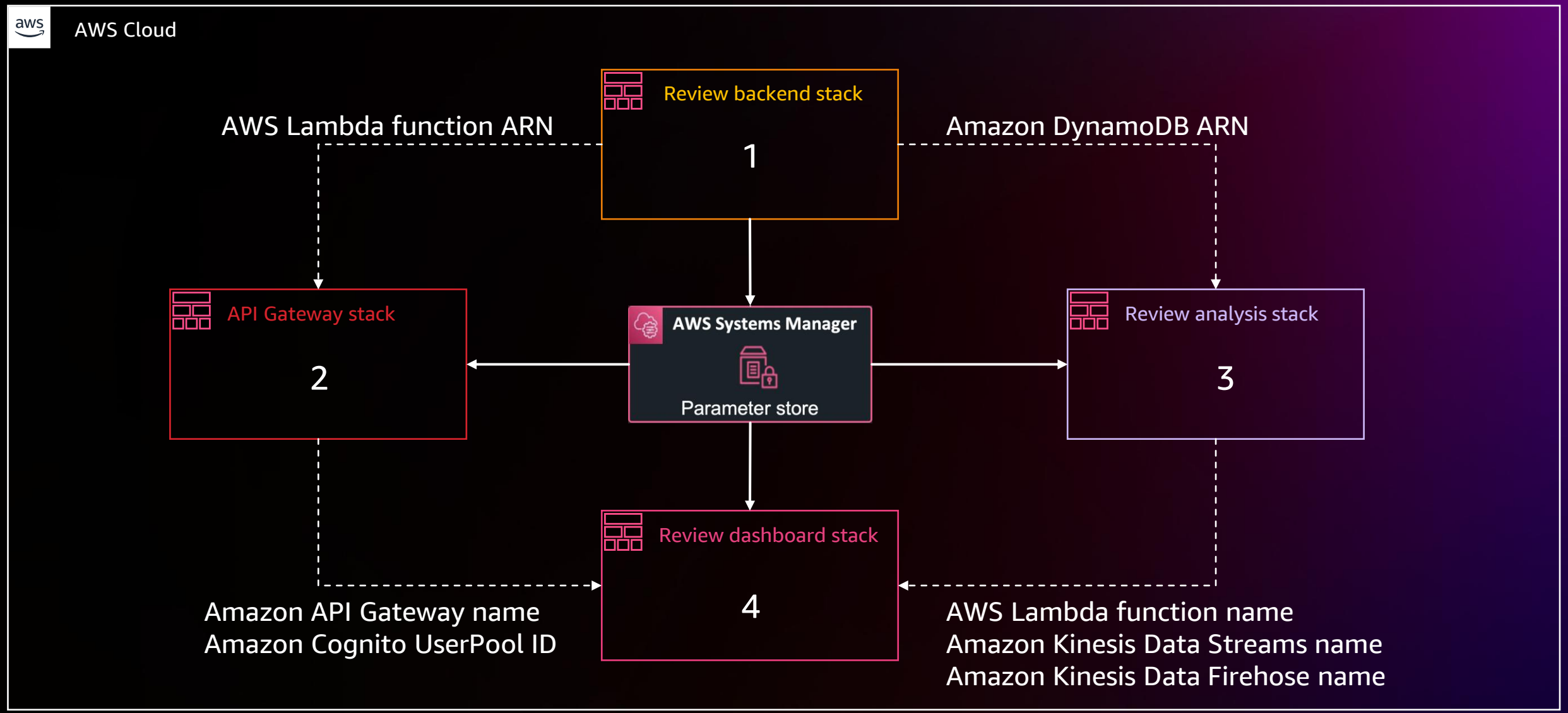
Implementation: Source code



https://github.com/aws-samples/aws-serverless-nlp-comprehend-using-aws-cdk

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

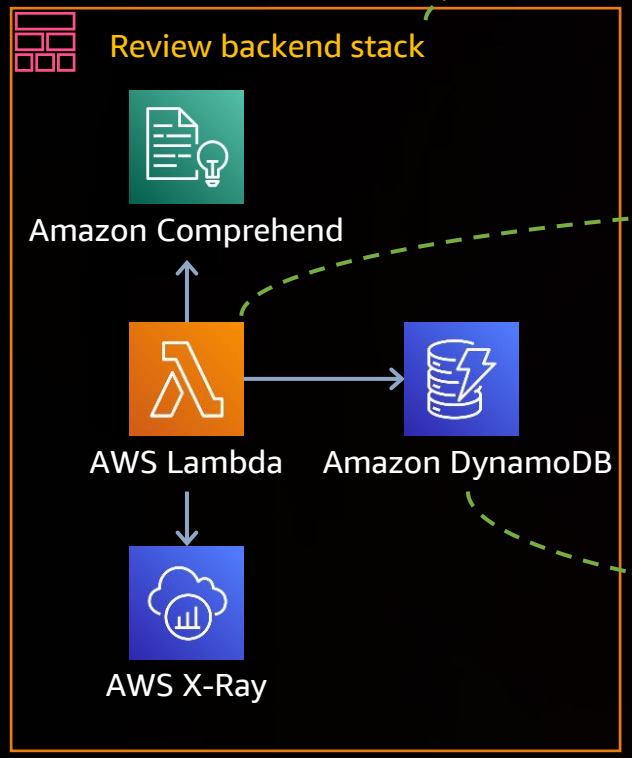
Implementation: Stack dependency



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation: 1. Review backend stack

```
new ReviewBackendStack(appContext, appContext.appConfig.Stack.ReviewBackend);
```




```
"Stack": {
    "ReviewBackend": {
        "Name": "ReviewBackendStack",

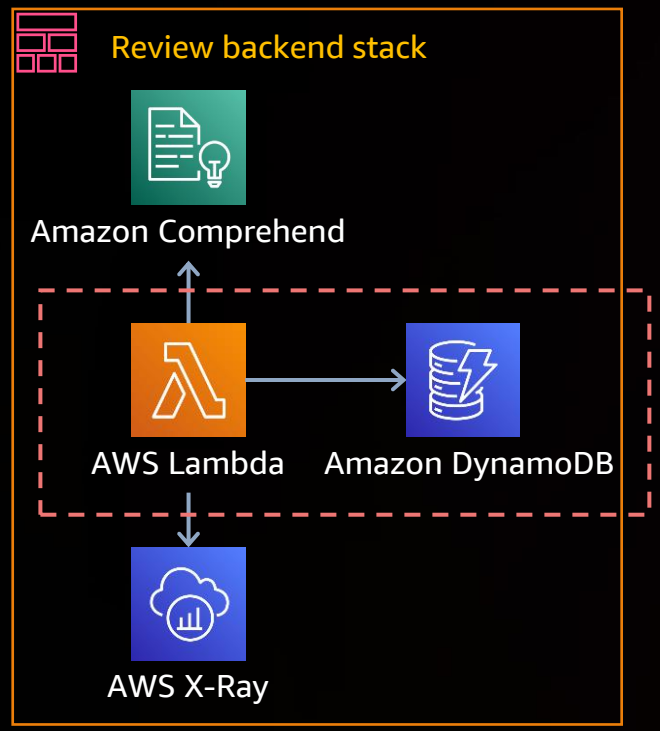
        "LambdaConfig": {
            "LambdaFuncName": "BackendFunc",
            "LambdaFuncMemory": 128,
            "LambdaFuncCode": "codes/lambda/review-backend/src",
            "LambdaFuncHandler": "handler.handle"
        },

        "DdbConfig": {
            "DdbTableName": "ReviewHistoryTable",
            "DdbTablePartitionKey": "ProductId",
            "DdbTableSortKey": "ReviewId"
        }
    }
}
```

app-config-dev.json

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation: 1. Review backend stack

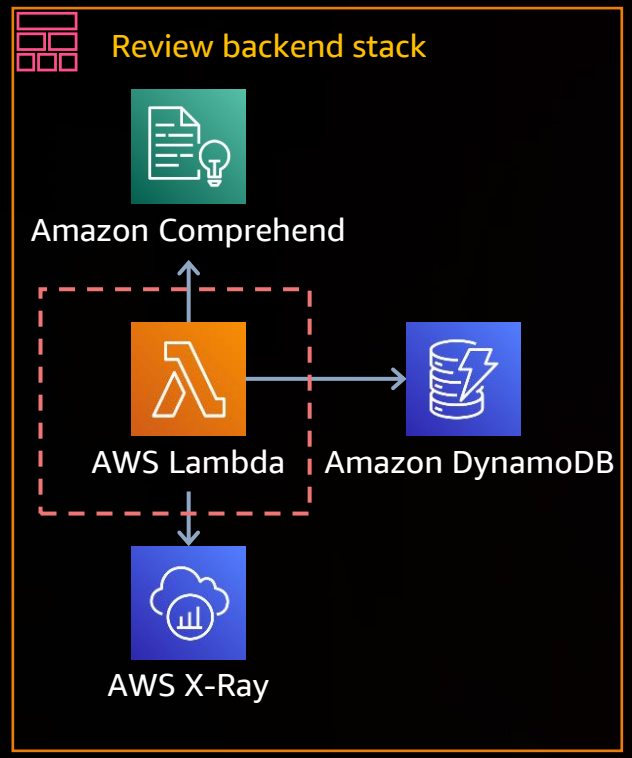



```
const ddbLambda = new LambdaToDynamoDB(this, 'lambda-dynamodb', {
    dynamoTableProps: {
        tableName: this.withStackName(stackConfig.DdbTableName),
        partitionKey: {
            name: stackConfig.DdbTablePartitionKey,
            type: ddb.AttributeType.STRING
        },
        sortKey: {
            name: stackConfig.DdbTableSortKey,
            type: ddb.AttributeType.STRING
        },
        stream: ddb.StreamViewType.NEW_IMAGE
    },
    lambdaFunctionProps: {
        functionName: this.withStackName(stackConfig.LambdaFuncName),
        runtime: lambda.Runtime.PYTHON_3_9,
        code: lambda.Code.fromAsset(stackConfig.LambdaFuncCode),
        handler: stackConfig.LambdaFuncHandler,
        memorySize: stackConfig.LambdaFuncMemory,
```

BaseStack
+ stackConfig: json
+ withStackName(baseName) + withProjectPrefix(baseName) + createS3BucketName(baseName)

aws

Implementation: 1. Review backend stack



CDK stack

```
const ddbLambda = new LambdaToDynamoDB(this, 'lambda-dynamodb', {
});
ddbLambda.lambdaFunction.addEnvironment('TABLE_NAME', ddbLambda.dynamoTable
ddbLambda.lambdaFunction.addEnvironment('TABLE_PARTITION', ddbConfig.DdbTab
ddbLambda.lambdaFunction.addEnvironment('TABLE_SORT', ddbConfig.DdbTableSor
```

```
_table_name = os.environ.get('TABLE_NAME', 'ReviewServiceDev
_table_partition = os.environ.get('TABLE_PARTITION', 'Produc
_table_sort = os.environ.get('TABLE_SORT', 'ReviewId')

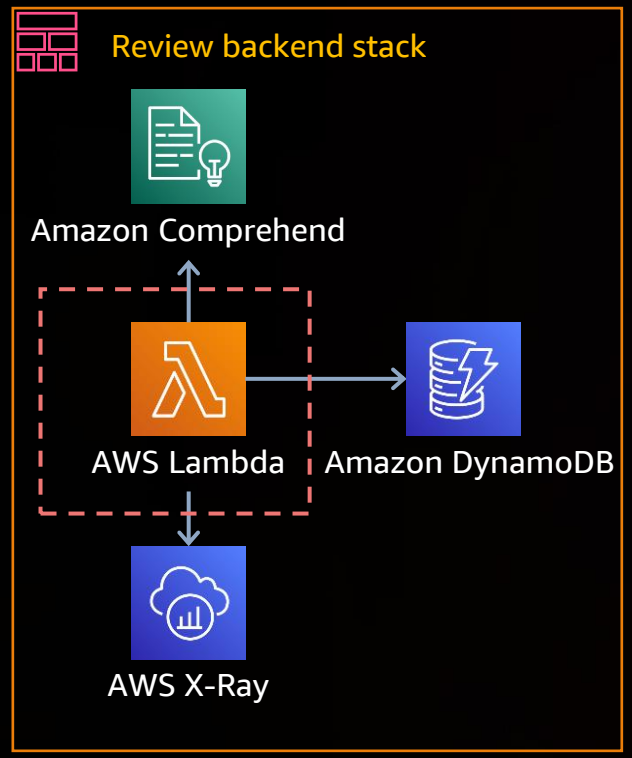
_comp = None
_table = None


def get_comprehend():
    global _comp
    if _comp is None:
        _comp = boto3.client('comprehend')
    return _comp
```

lambda handler

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation: 1. Review backend stack



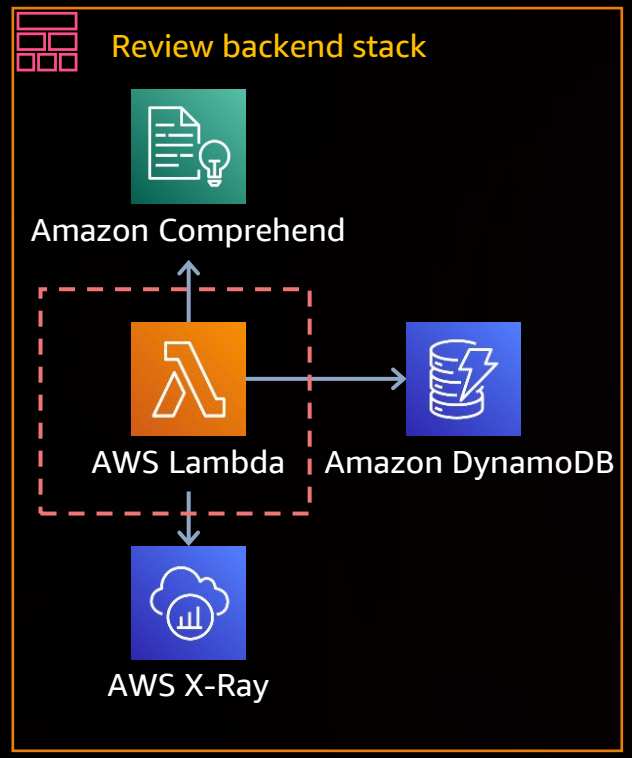

```python
def analyze_sentiment(review: str, language: str) -> dict:
    try:
        response = get_comprehend().detect_sentiment(
                    Text=review,
                    LanguageCode=language
                )
        del response['ResponseMetadata']
        return
    except Clie
        print('
        raise e
```

```python
def insert_item(request: dict, sentiment: dict) -> bool :
    ts_format = '%Y/%m/%d-%H:%M:%S'
    timestamp = datetime.now().strftime(ts_format)
    random_suffix = str(uuid.uuid1()).replace('-', '')

    item = {…
    try:
        get_table().put_item(
                Item=item
            )
        return True
    except ClientError as e:
        print('[EXCEPT] insert_item', e)
        raise e
```

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation: 1. Review backend stack




```python
def test_003a(self):
    logging.info('---test_003a: get_table----')

    table = handler.get_table()
    assert(table is not None)
```

```python
def test_004a(self):
    logging.info('---test_004a: analyze_sentiment happy case----')

    review = 'This is good.'
    language = 'en'

    result = handler.analyze_sentiment(review, language)

    assert(result is not None)
    assert('Sentiment' in result)
    assert(result['Sentiment'] == 'POSITIVE')
```

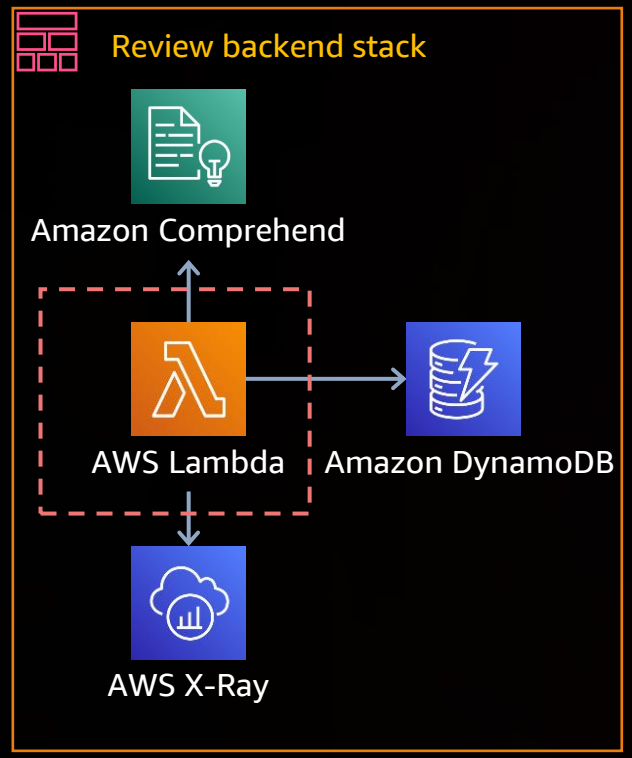
```python
def test_004b(self):
    logging.info('---test_004b: analyze_sentiment except----')

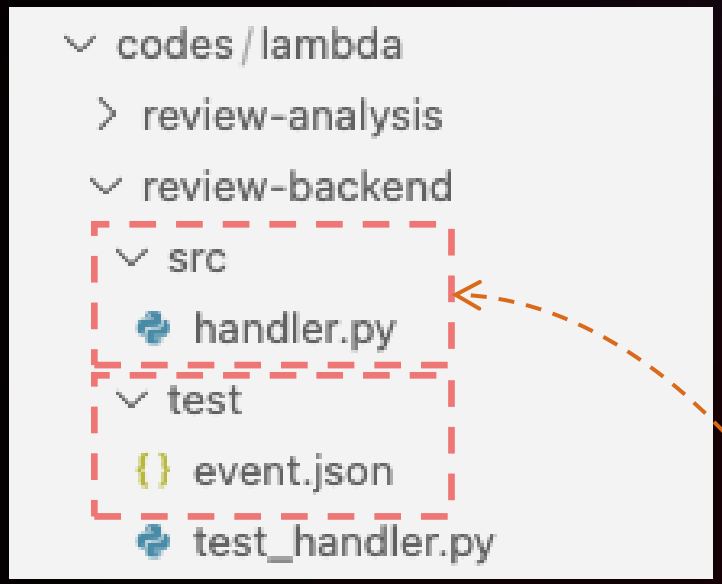
    review = ''
    language = ''

    with pytest.raises(ParamValidationError):
        result = handler.analyze_sentiment(review, language)
```

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation: 1. Review backend stack



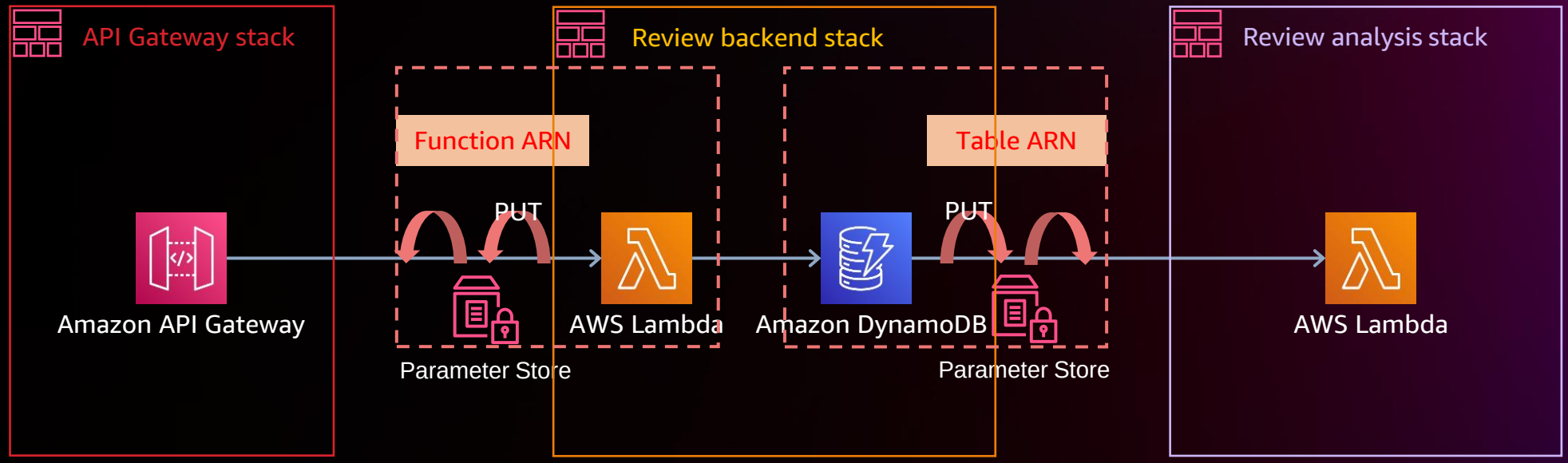



```
const ddbLambda = new LambdaToDynamoDB(this, 'lambda-dynamodb', {
    dynamoTableProps: {…
    },
    lambdaFunctionProps: {
        functionName: this.withStackName(lambdaConfig.LambdaFuncName),
        runtime: lambda.Runtime.PYTHON_3_9,
        code: lambda.Code.fromAsset('lambda/review-backend/src'),
        handler: lambdaConfig.LambdaFuncHandler,
        memorySize: lambdaConfig.LambdaFuncMemory,
    }
});
```

CDK stack

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation: 1. Review backend stack



BaseStack

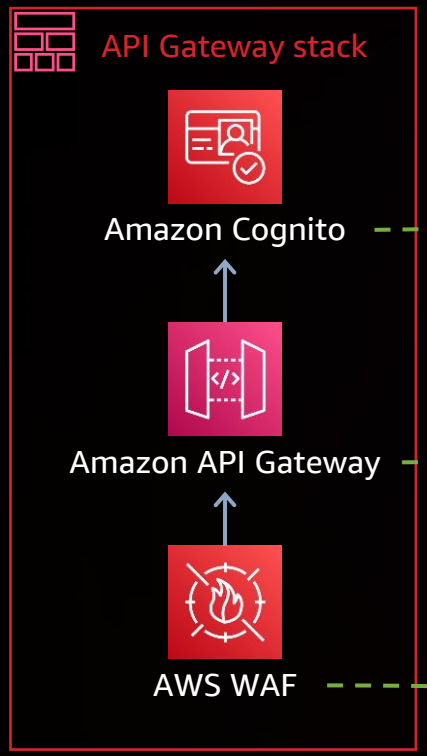
+ getParameter(key)
+ putParameter(key, value)

```
this.putParameter(`${stackConfig.DdbTableName}TableArn`, ddbLambda.dynamoTa
this.putParameter(`${stackConfig.DdbTableName}TableStreamArn`, ddbLambda.dynamoTable.tableStreamArn!);

this.putParameter(`${stackConfig.LambdaFuncName}FunctionArn`, ddbLambda.lambdaFunction.functionArn);
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation: 2. API Gateway stack




```
"Stack": {
    "ApiGateway": {
        "Name": "ApiGatewayStack",

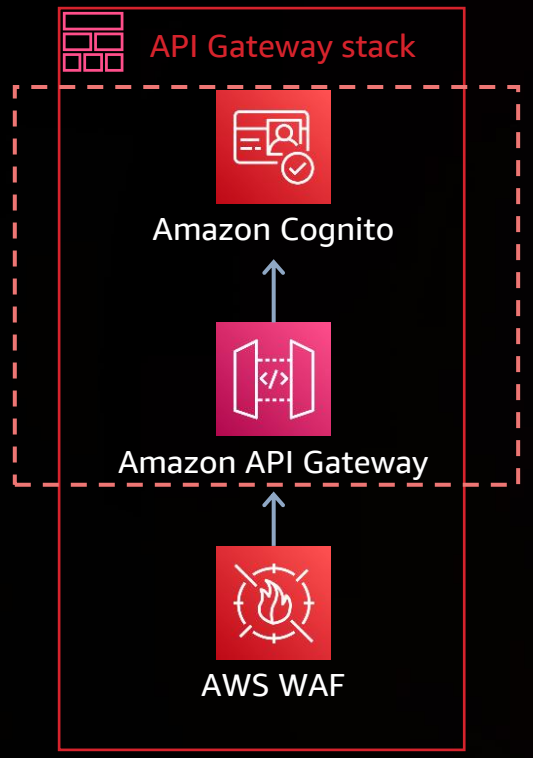
        "CognitoConfig": {
            "CognitoUserPoolName": "ReviewUserPool"
        },

        "ApiConfig": {
            "ApiGatewayName": "ReviewApis",
            "ResourceMapping": [
                {
                    "ResourceName": "review",
                    "LambdaFuncName": "BackendFunc",
                    "Methods": ["POST"]
                }
            ]
        },

        "WafConfig": {
            "WebAclName": "Waf",
            "WafAwsManagedRules": [
                "AWSManagedRulesCommonRuleSet"
            ]
        }
    }
}
```

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

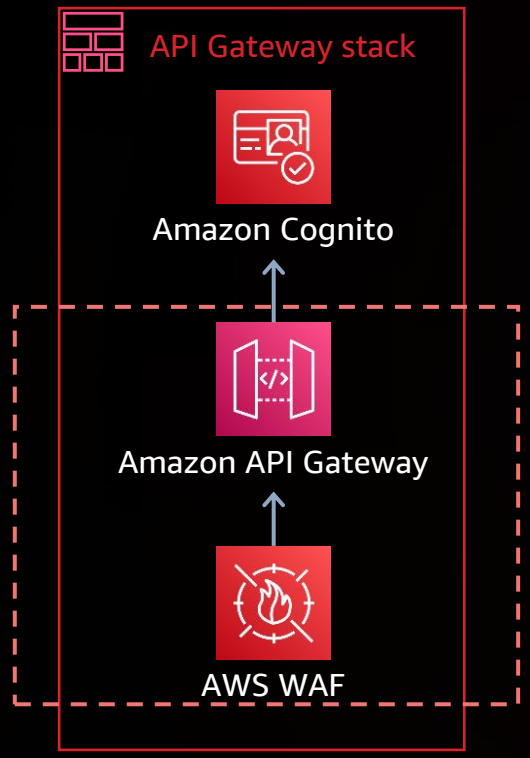
Implementation: 2. API Gateway stack




```
const apiLambda = new CognitoToApiGatewayToLambda(this, 'apigateway-cognito-lambda', {
    apiGatewayProps: {
        restApiName: this.withStackName(stackConfig.ApiGatewayName),
        endpointConfiguration: {
            types: [apigateway.EndpointType.REGIONAL]
        },
        proxy: false,
        deployOptions: {
            loggingLevel: apigateway.MethodLoggingLevel.ERROR,
        },
    },
    cognitoUserPoolProps: {
        userPoolName: this.withStackName(stackConfig.CognitoUserPoolName),
    },
    cognitoUserPoolClientProps: {
        authFlows: {
            userPassword: true,
            userSrp: true,
            custom: true,
        }
    },
    existingLambdaObj: this.createDefaultHandler()
});
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation: 2. API Gateway stack




```
new WafwebaclToApiGateway(this, 'wafwebacl-apigateway', {
    existingApiGatewayInterface: apiLambda.apiGateway,
    webaclProps: this.createWafwebaclProps('REGIONAL', stackConfig.WafAwsManagedRules)
});
```

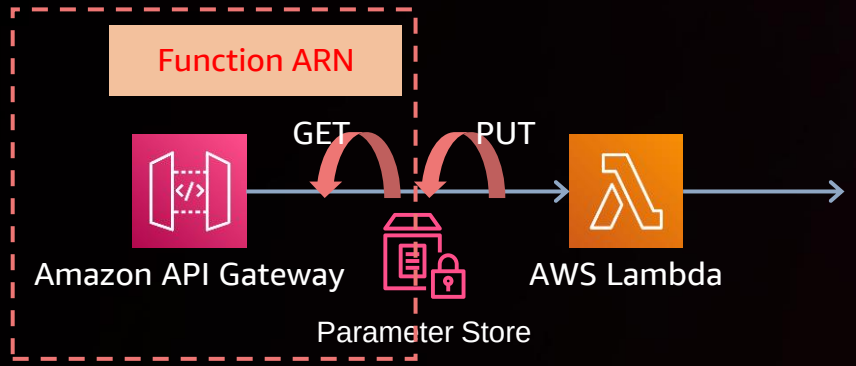
```
"Stack": {
    "ApiGateway": {
        "Name": "ApiGatewayStack",

        "CognitoUserPoolName": "ReviewUserPool",
        "ApiGatewayName": "ReviewApis",
        "WebAclName": "Waf",

        "ResourceMapping": [
            {
                "ResourceName": "review",
                "LambdaFuncName": "BackendFunc",
                "Methods": ["POST"]
            }
        ],
        "WafAwsManagedRules": [
            "AWSManagedRulesCommonRuleSet"
        ]
    }
}
```

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation: 2. API Gateway stack




```
"ResourceMapping": [
    {
        "ResourceName": "review",
        "LambdaFuncName": "BackendFunc",
        "Methods": ["POST"]
    }
],
```

```
stackConfig.ResourceMapping.forEach(item => {
    const resource = apiLambda.apiGateway.root.addResource(item.ResourceName);
    this.addCorsOptions(resource);

    const lambdaFuncArn = this.getParameter(`${item.LambdaFuncName}FunctionArn`);
    const lambdaFunc = lambda.Function.fromFunctionArn(this, item.LambdaFuncName, lambdaFuncArn);
    const lambdaFuncIntegration = new apigateway.LambdaIntegration(lambdaFunc, {
        credentialsRole: this.getCredentialRole(item.LambdaFuncName, lambdaFuncArn)
    });
    for (let method of item.Methods) {
        resource.addMethod(method, lambdaFuncIntegration);
    }
});
apiLambda.addAuthorizers();
```

aws

Implementation: 2. API Gateway stack

```
const apiLambda = new CognitoToApiGatewayToLambda(this, 'apigateway-cognito-lambda', {
    apiGatewayProps: {
        restApiName: this.withStackName(stackConfig.ApiGatewayName),
        endpointConfiguration: {
            types: [apigateway.EndpointType.REGIONAL]
        },
        proxy: false,
        deployOptions: {
            loggingLevel: apigateway.MethodLoggingLevel.ERROR,
        },
    },
    cognitoUserPoolProps: {
        userPoolName: this.withStackName(stackConfig.CognitoUserPoolName),
    },
    cognitoUserPoolClientProps: {
        authFlows: {
            userPassword: true,
            userSrp: true,
            custom: true,
        }
    },
    existingLambdaObj: this.createDefaultHandler()
});
```

aws

Implementation: 2. API Gateway stack

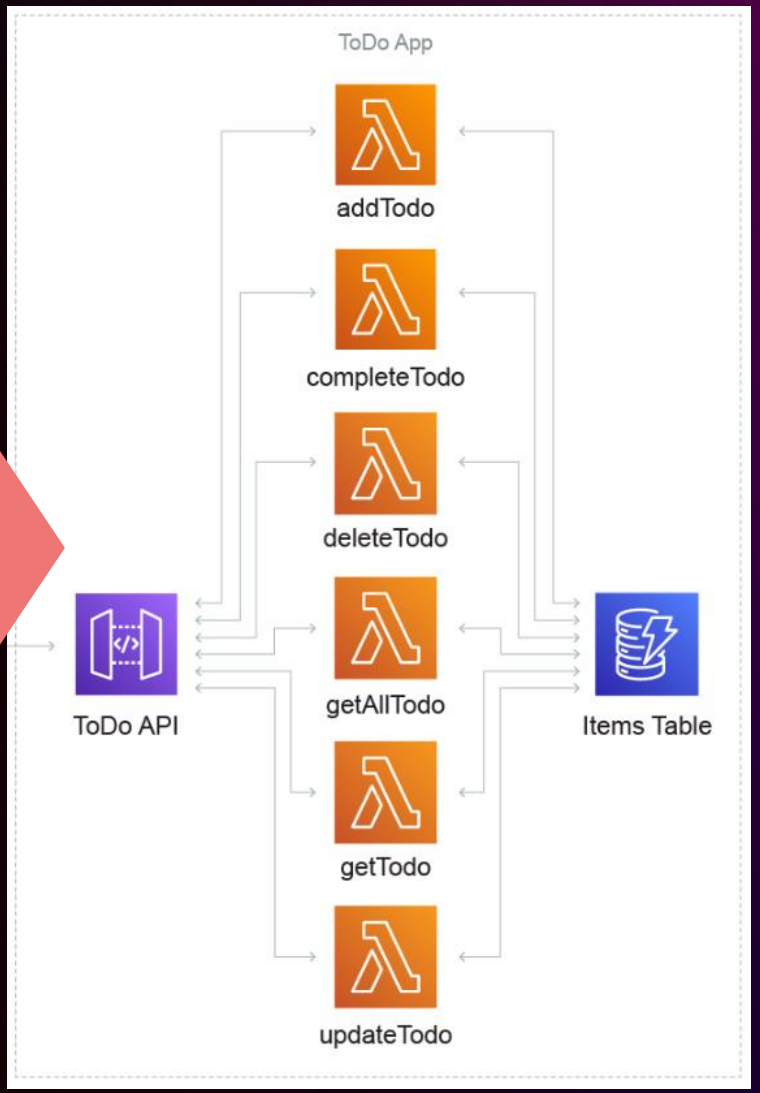
```
"ApiGateway": {
    "Name": "ApiGatewayStack",

    "ApiConfig": {
        "ApiGatewayName": "ToDoApis",
        "ResourceMapping": [
            {
                "ResourceName": "addTodo",
                "LambdaFuncName": "AddTodoFunc",
                "Methods": ["POST"]
            },
            {
                "ResourceName": "getTodo",
                "LambdaFuncName": "GetTodoFunc",
                "Methods": ["GET"]
            },
            {
                "ResourceName": "deleteTodo",
                "LambdaFuncName": "DeleteTodoFunc
                "Methods": ["DELETE"]
            },
            {
                "ResourceName": "updateTodo",
                "LambdaFuncName": "UpdateTodoFunc",
                "Methods": ["UPDATE"]
            }
        ]
    },

    "CognitoConfig": {…
    },

    "WafConfig": {…
    }
},
```

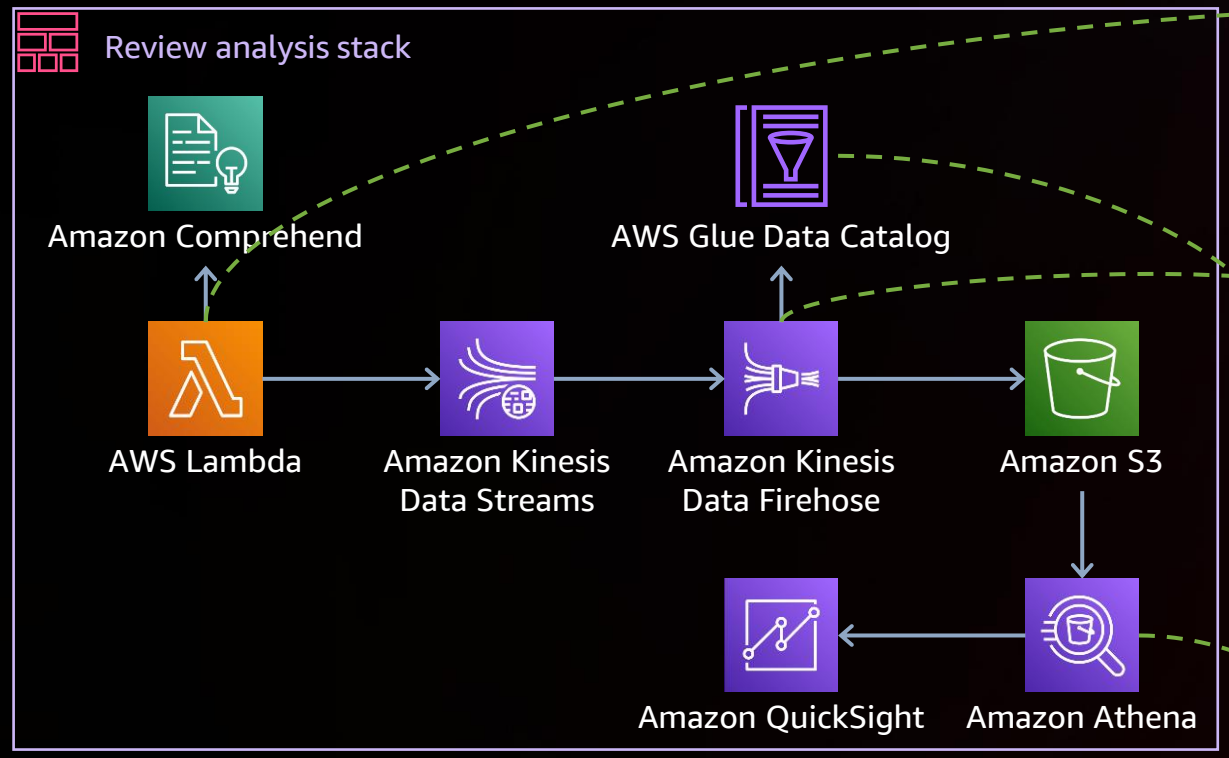
Configuration injection



https://aws.amazon.com/serverless/

aws

Implementation: 3. Review analysis stack




```
"Stack": {
    "ReviewAnalysis": {
        "Name": "ReviewAnalysisStack",

        "LambdaConfig": {
            "LambdaFuncName": "AnalysisFunc",
            "LambdaFuncMemory": 128,
            "LambdaFuncCode": "codes/lambda/review-analysis/src",
            "LambdaFuncHandler": "handler.handle",
            "LambdaFuncBatch": 10,
            "LambdaFuncWindow": 1,
            "StreamBatchSize": "10"
        },

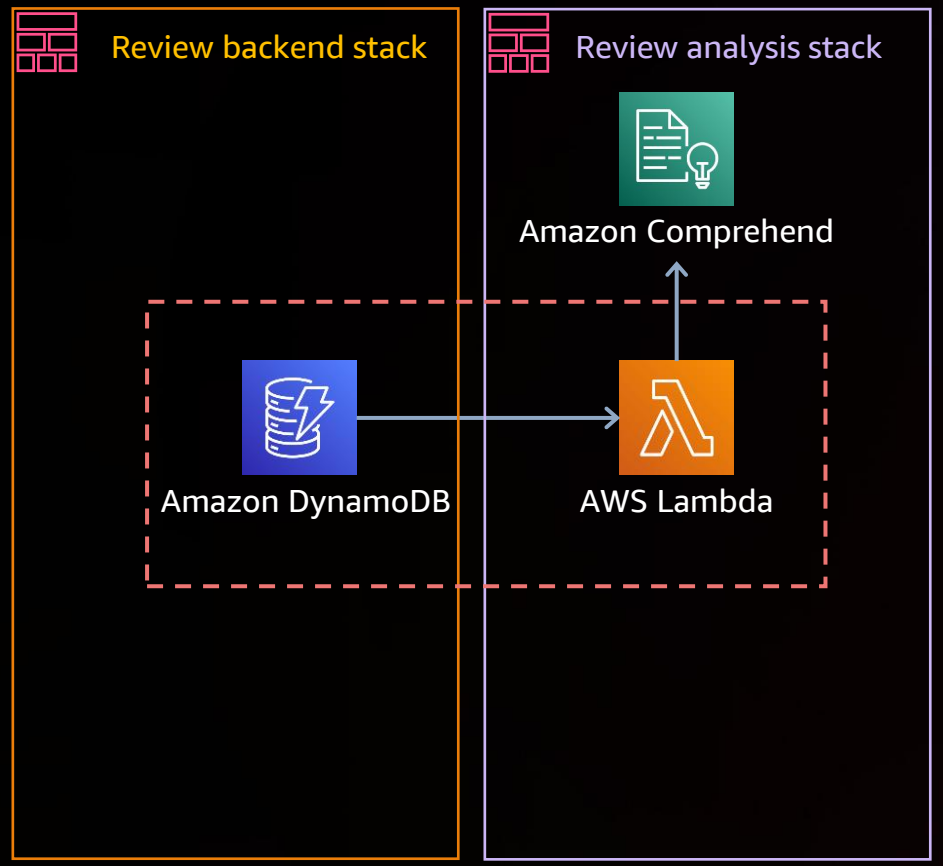
        "KinesisConfig": {
            "KinesisStreamName": "Stream",
            "KinesisFireHoseName": "Hose",
            "KinesisBucketName": "kinesis"
        },

        "GlueConfig": {
            "GlueDatabaseName": "Analysis-Database",
            "GlueTableName": "Comprehend-Table"
        },

        "AthenaConfig": {
            "AtheanGroupName": "Group",
            "AtheanBucketName": "athena",
            "AtheanQuerySentiment": "sentiment-table",
            "AtheanQueryEntities": "entities-table",
            "AtheanQuerySyntax": "syntax-table"
        }
    }
}
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

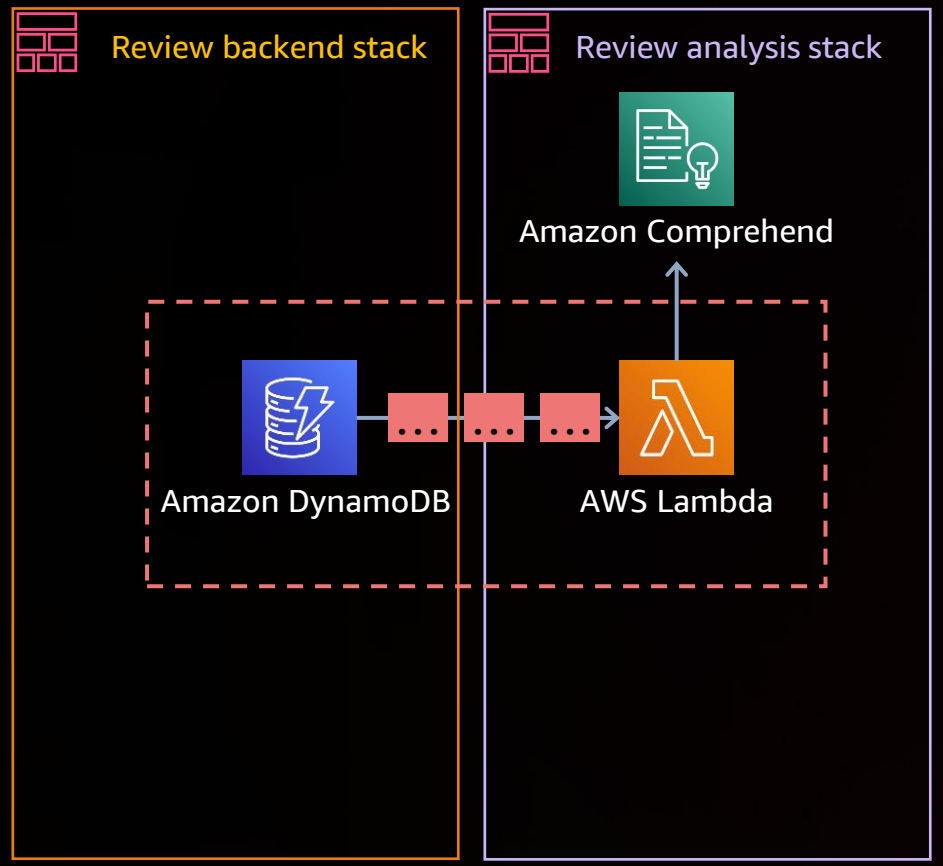
Implementation: 3. Review analysis stack




```
const streamLambda = new DynamoDBStreamsToLambda(this, 'dynamoc
    lambdaFunctionProps: {
        functionName: this.withStackName(stackConfig.LambdaFunc
        runtime: lambda.Runtime.PYTHON_3_9,
        code: lambda.Code.fromAsset(stackConfig.LambdaFuncCode)
        handler: stackConfig.LambdaFuncHandler,
        memorySize: stackConfig.LambdaFuncMemory,
        environment: {
            STREAM_NAME: kinesisStream.streamName
        }
    },
    existingTableInterface: backendTable,
    dynamoEventSourceProps: {
        startingPosition: lambda.StartingPosition.LATEST,
        batchSize: stackConfig.LambdaFuncBatch,
        maxBatchingWindow: cdk.Duration.minutes(stackConfig.Lam
        onFailure: new lambdaEvent.SqsDlq(new sqs.Queue(this,
    }
});
```

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation: 3. Review analysis stack




```python
def batch_detect(batch_array: list):
    reviews = [item['Review'] for item in batch_array]

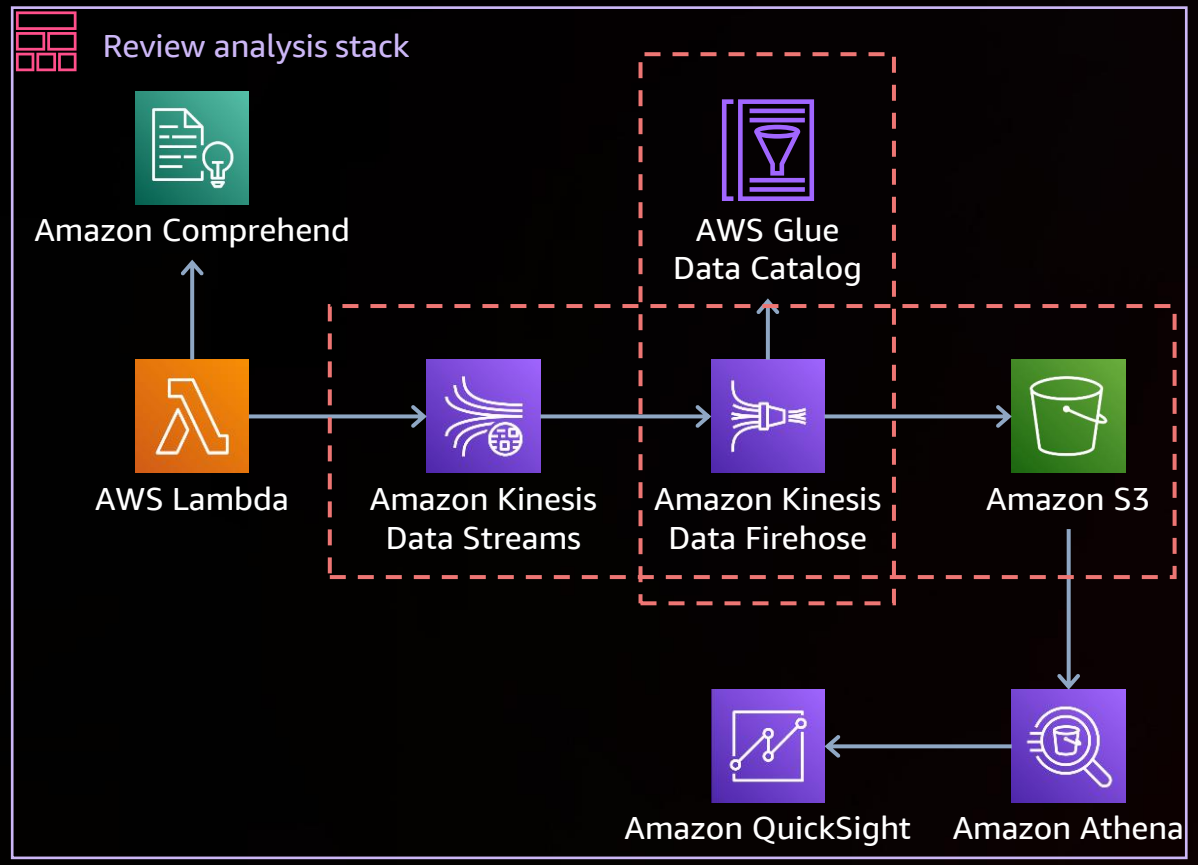
    try:
        comp = get_comprehend()
        response_entities = comp.batch_detect_entities(
                    TextList=reviews,
                    LanguageCode='en'
                )
        response_syntax = comp.batch_detect_syntax(
                    TextList=reviews,
                    LanguageCode='en'
                )

        records = []
        for index, item in enumerate(reviews):
            batch_array[index]['Entities'] = json.loads(json.dumps(res
            batch_array[index]['Syntax'] = json.loads(json.dumps(respo
            records.append({
                'Data': json.dumps(batch_array[index], default=str),
                'PartitionKey': batch_array[index]['ProductId']
            })

        get_kinesis().put_records(
                    StreamName=_stream_name,
```

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

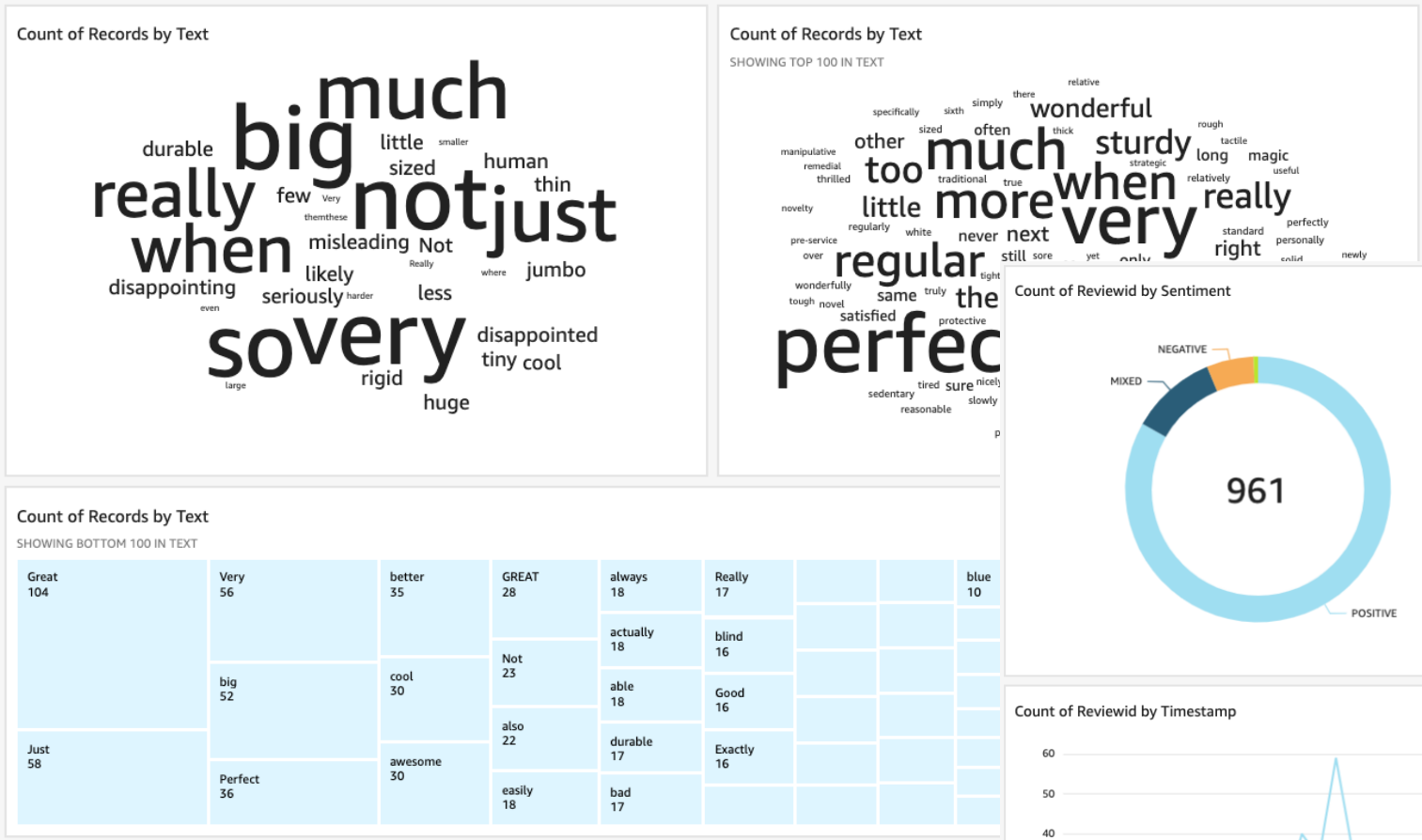
Implementation: 3. Review analysis stack




```
const pipeline = new KinesisStreamsToKinesisFirehoseToS3(this, 'st
    kinesisStreamProps: {
        streamName: this.withStackName(config.KinesisStreamName),
        encryption: kinesis.StreamEncryption.MANAGED
    },
    kinesisFirehoseProps: {
        deliveryStreamType: 'KinesisStreamAsSource',
        deliveryStreamName: this.withStackName(config.KinesisFireH
        extendedS3DestinationConfiguration: {
            bucketArn: kinesisBucket.bucketArn,
            compressionFormat: 'UNCOMPRESSED',
            bufferingHints: {…
            },
            dataFormatConversionConfiguration: {
                enabled: true,
                inputFormatConfiguration: {
                    deserializer: {  openXJsonSerDe: {} }
                },
                outputFormatConfiguration: {
                    serializer: {  parquetSerDe: {} }
                },
                schemaConfiguration: {
                    region: this.commonProps.appConfig.Project.Reg
                    roleArn: this.createFireHoseRole().roleArn,
                    versionId: 'LATEST',
```

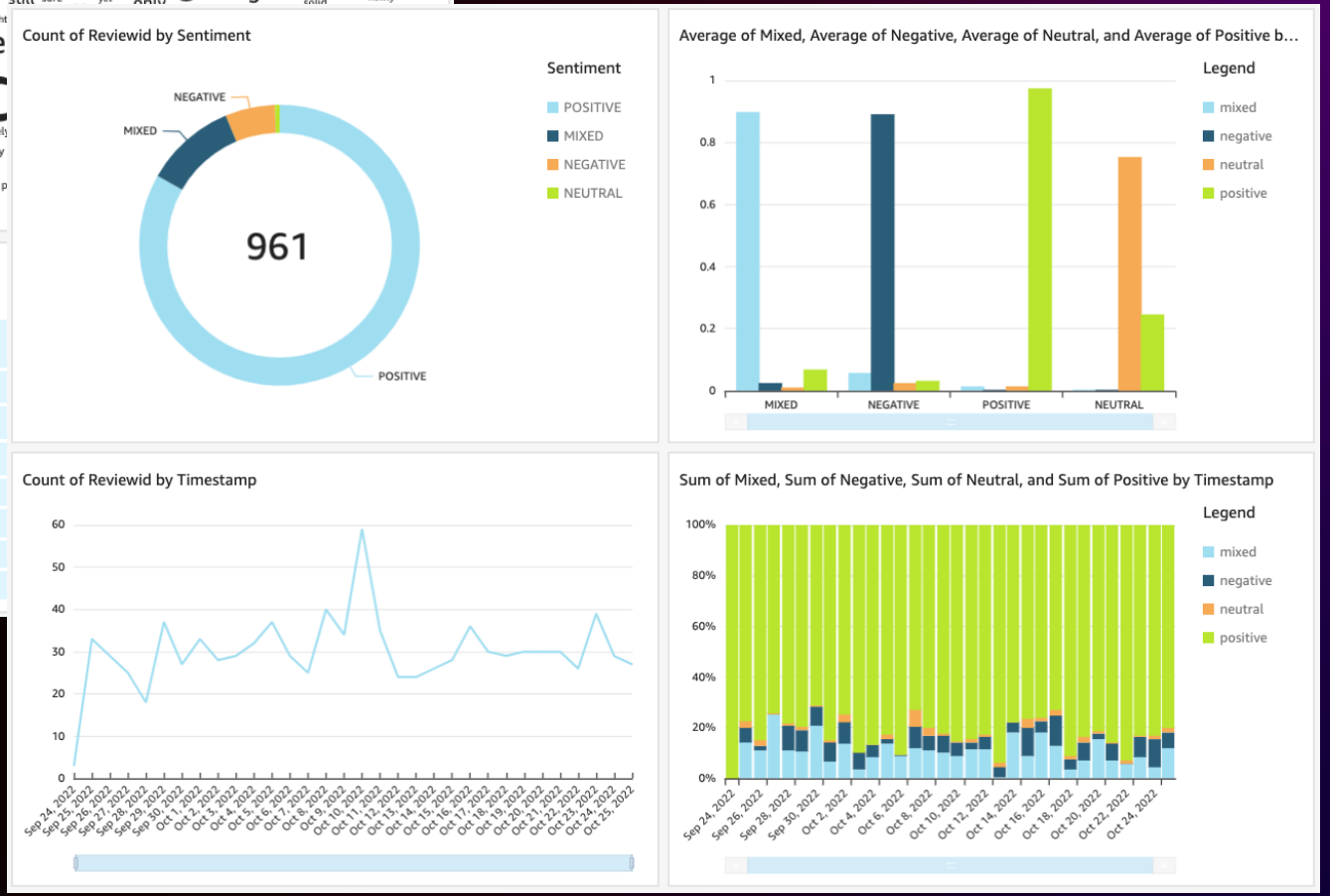
aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation: 3. Review analysis stack



Syntax analysis table

Sentiment analysis table



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

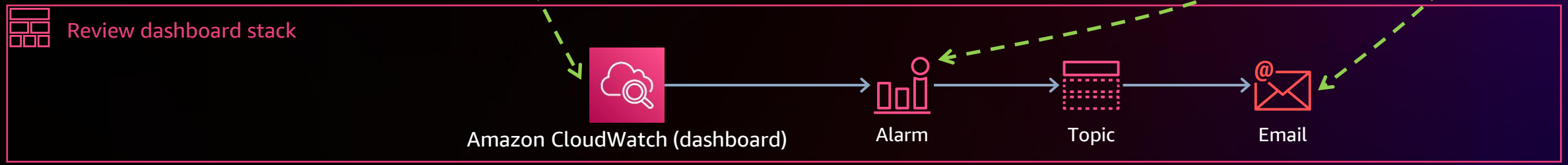
Implementation: 4. Review dashboard stack

```
"ReviewDashboard": {
    "Name": "ReviewDashboardStack",

    "DashboardName": "ReviewDashboard",

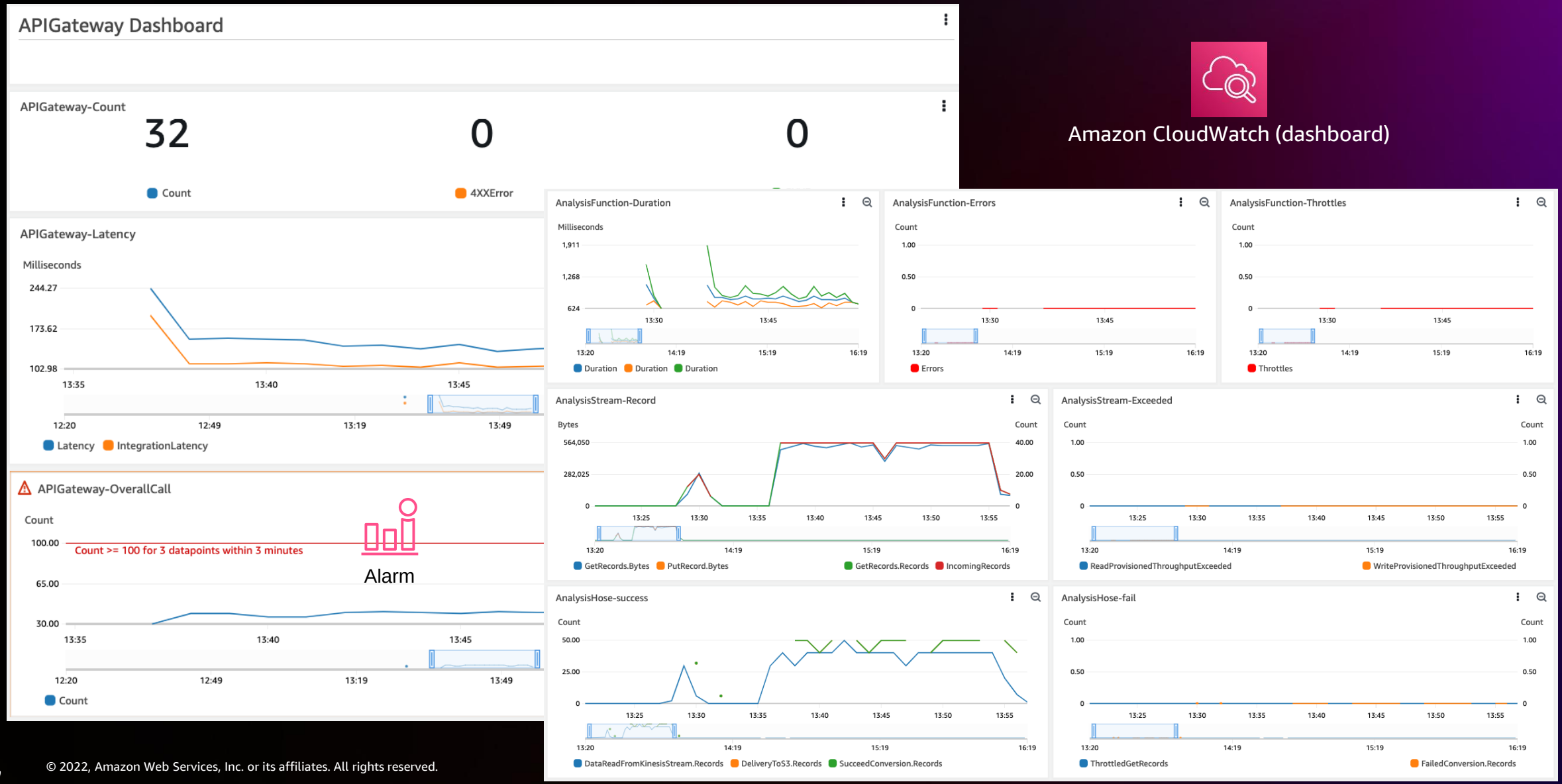
    "SubscriptionEmails": [
        "devops1@company.com",
        "devops2@company.com",
        "devops3@company.com"
    ],

    "ApiGatewayOverallCallThreshold": 100,
    "ApiGatewayError4xxCallThreshold": 20,
    "ApiGatewayError5xxCallThreshold": 20
}
```



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementation: 4. Review dashboard stack



Amazon CloudWatch (dashboard)

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Build Prototype: Demo

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws-samples / aws-serverless-nlp-comprehend-using-aws-cdk

Public

Notifications Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights

main 1 branch 0 tags

Go to file Code

kwonyulchoi Update scripts 8edaafe 2 hours ago 6 commits

codes/lambda	initial commit	5 days ago
config	Update config file	2 hours ago
docs/asset	initial commit	5 days ago
infra	initial commit	5 days ago
lib/template	[Framework] Add a new option to turn off S3BucketAccessLog	4 days ago
script	Update scripts	2 hours ago
test	initial commit	5 days ago
.gitignore	initial commit	5 days ago
.npmignore	initial commit	5 days ago
CODE_OF_CONDUCT.md	Initial commit	16 days ago

About

This repository describes how to design and implement Natural Language Processing(NLP)-based service using AWS Serverless, Amazon Comprehend and AWS Cloud Development Kit(CDK)

nlp serverless kinesis cdk comprehend solutions-constructs

Readme

MIT-0 license

Code of conduct

0 stars

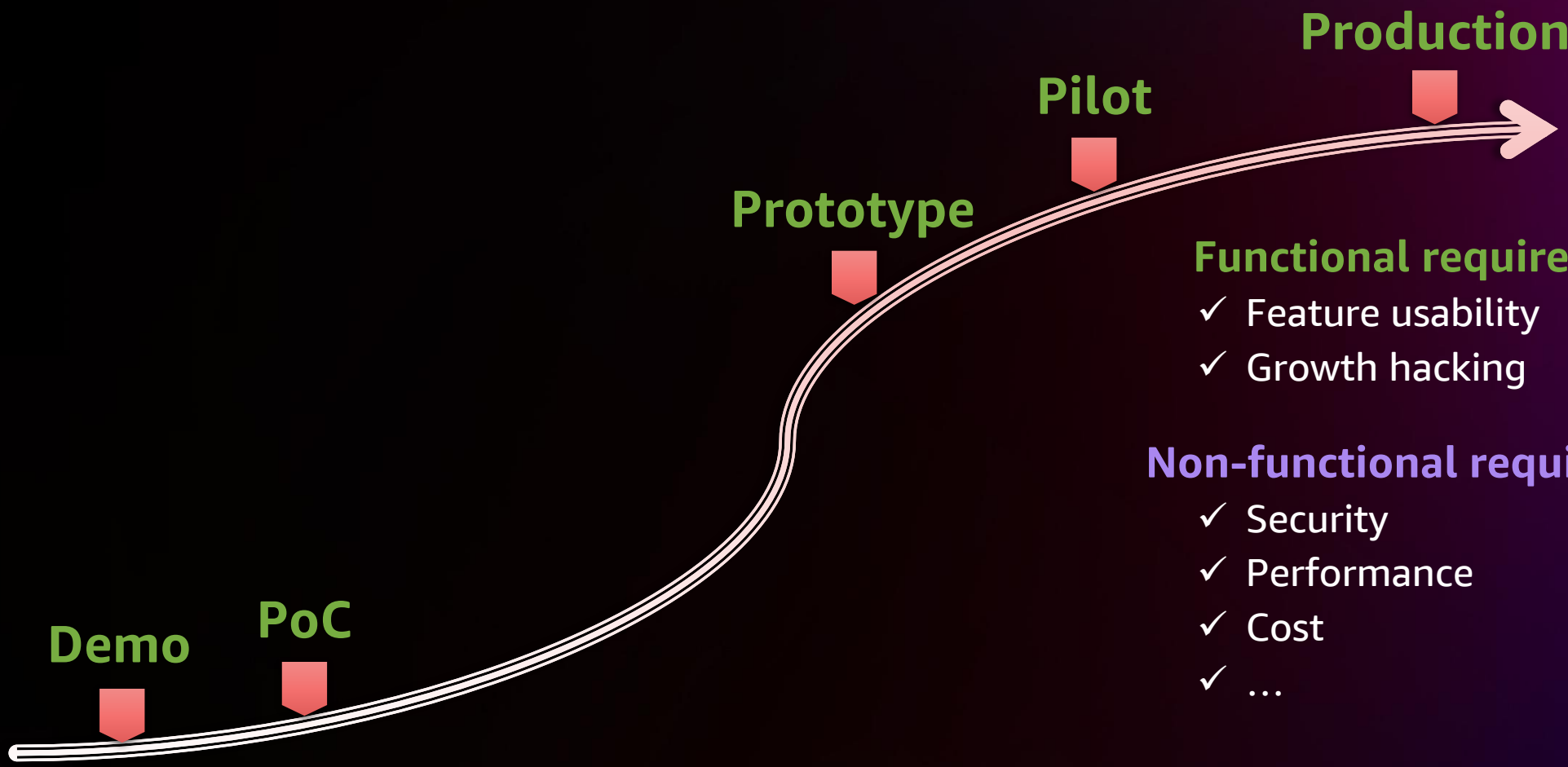
5 watching

0 forks

After prototyping

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Production journey



Functional requirement

- ✓ Feature usability
- ✓ Growth hacking

Non-functional requirement

- ✓ Security
- ✓ Performance
- ✓ Cost
- ✓ …

The journey from development to production

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Security review

https://github.com/stelligent/cfn_nag

Background

The cfn-nag tool looks for patterns in CloudFormation templates that may indicate insecure infrastructure. Roughly speaking, it will look for:

- IAM rules that are too permissive (wildcards)
- Security group rules that are too permissive (wildcards)
- Access logs that aren't enabled
- Encryption that isn't enabled
- Password literals

https://github.com/cdklabs/cdk-nag

Available Packs

See RULES for more information on all the available packs.

1. AWS Solutions
2. HIPAA Security
3. NIST 800-53 rev 4
4. NIST 800-53 rev 5
5. PCI DSS 3.2.1

app-config-demo.json

```
{
    "Project": {…
    },

    "Global": {
        "CdkNagEnable": true
    },

    "Stack": {…
    }
}
```

```
describe('cdk-nag AwsSolutionsPack ReviewBackendStac
    let stack: Stack;

    beforeAll(() => {
        // GIVEN
        const appContext = new AppContext({…
        });
        stack = new ReviewBackendStack(appContext, appContext.appConfig.Stack.ReviewBackend);

        // WHEN
        Aspects.of(stack).add(new AwsSolutionsChecks());
    });

    // THEN
    Run | Debug
    test('No unsuppressed Warnings', () => {
        const warnings = Annotations.fromStack(stack).findWarning(
            '*',
            Match.stringLikeRegexp('AwsSolutions-.*')
        );
        expect(warnings).toHaveLength(0);
    });

    Run | Debug
    test('No unsuppressed Errors', () => {
        const errors = Annotations.fromStack(stack).findError(
            '*',
            Match.stringLikeRegexp('AwsSolutions-.*')
        );
        expect(errors).toHaveLength(0);
    });
});
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Pilot

```
"ReviewBackend": {
    "Name": "ReviewBackendStack",

    "LambdaConfig": {
        "LambdaFuncName": "BackendFunc",
        "LambdaFuncMemory": 128,
        "LambdaFuncCode": "codes/lambda/review-backend/src",
        "LambdaFuncHandler": "handler.handle",
        "PilotProductIds": "id-001,id-002,id-003"
    },

    "DdbConfig": {…
    }
},
```

```
const ddbLambda = new LambdaToDynamoDB(this, 'lambda-dynamodb', {
    dynamoTableProps: {…
    },
    lambdaFunctionProps: {
        functionName: this.withStackName(lambdaConfig.LambdaFuncName),
        runtime: lambda.Runtime.PYTHON_3_9,
        code: lambda.Code.fromAsset(lambdaConfig.LambdaFuncCode),
        handler: lambdaConfig.LambdaFuncHandler,
        memorySize: lambdaConfig.LambdaFuncMemory,
        environment: {
            PILOT_PRODUCT_IDS: lambdaConfig.PilotProductIds
        }
    }
});
```

```
_pilot_product_ids = os.environ.get('PILOT_PRODUCT_IDS', '').split(',')
```

```
if success:
    if request['ProductId'] in _pilot_product_ids:
        sentiment = analyze_sentiment(request['Review'], request['Language'])
    else:
        sentiment = {}
    insert_item(request, sentiment)
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Performance/Cost

Default Method Throttling

Choose the default throttling level for the methods in this stage. Each method in this stage will respect these rate and burst settings. Your current account l tling rate is **10000** requests per second with a burst of **5000** requests. Read mo API Gateway throttling

Enable throttling

Rate 10000 requests per second

Burst 5000 requests



Amazon API Gateway

General configuration Info

Edit

Description	Memory	Ephemeral storage
-	128 MB	512 MB

Timeout: 0 min 3 sec



AWS Lambda

Concurrency

Edit

Function concurrency	Unreserved account concurrency
Use unreserved account concurrency	1000

Data stream capacity Info

Capacity mode

- On-demand
  Use this mode when your data stream's throughput requirements are unpredictable and variable. With on-demand mode, your data stream's capacity scales automatically.
- Provisioned
  Use provisioned mode when you can reliably estimate throughput requirements of your data stream. With provisioned mode, your data stream's capacity is fixed.

Provisioned shards
The total capacity of a stream is the sum of the capacities of its shards. You can update provisioned shards after updating capacity mode.
1

Total data stream capacity
Shard capacity is determined by the number of provisioned shards. Each shard ingests up to 1 MiB/second and 1,00 d and emits up to 2 MiB/second. If writes and reads exceed capacity, the application will receive throttles.

Write capacity	Read capacity
Maximum	Maximum
1 MiB/second and 1,000 records/second	2 MiB/second



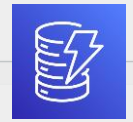
Amazon Kinesis Data Streams

Edit read/write capacity

Capacity mode Info

- On-demand
  Simplify billing by paying for the actual reads and writes your application performs.
- Provisioned
  Manage and optimize your costs by allocating read/write capacity in advance.

▸ Capacity calculator



Amazon DynamoDB

Table capacity

Read capacity

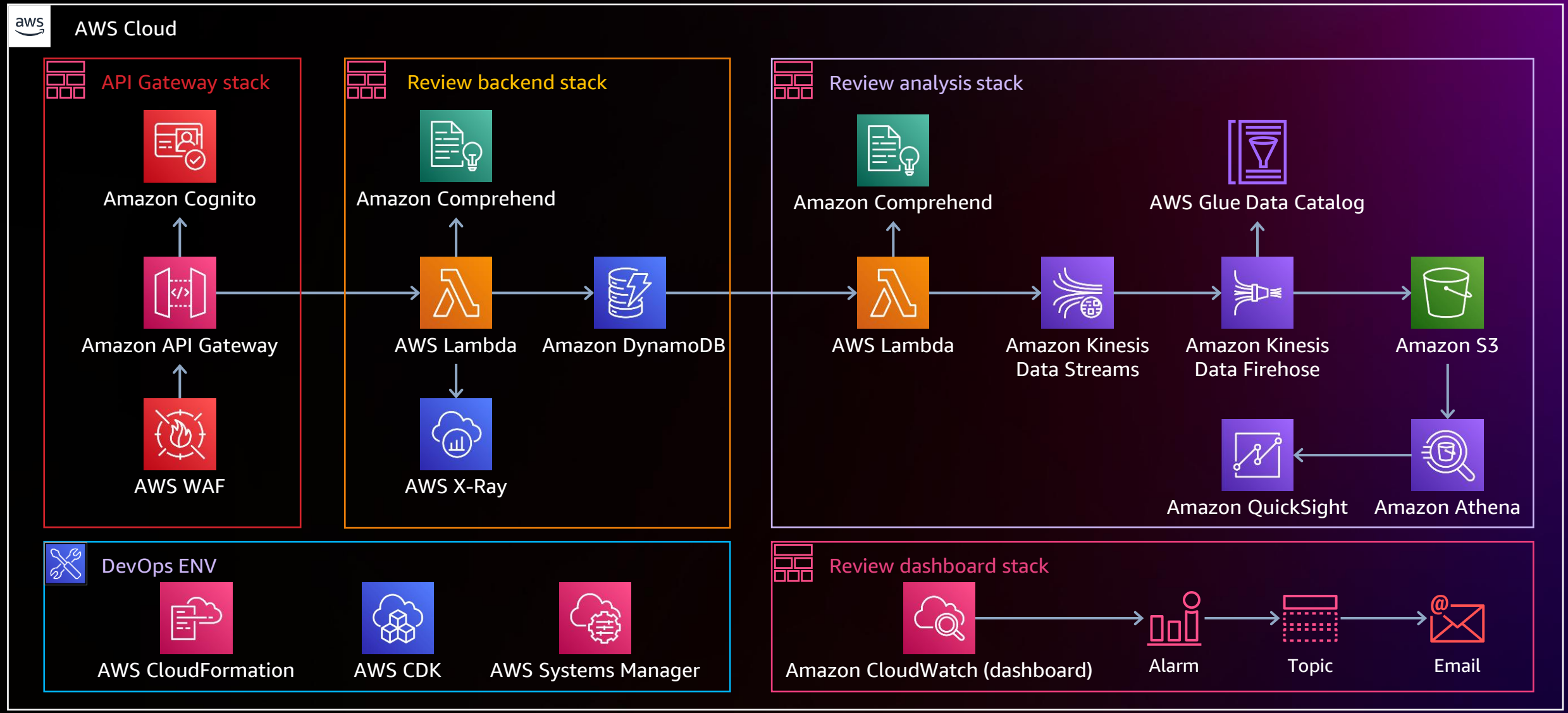
Auto scaling Info
Dynamically adjusts provisioned throughput capacity on your behalf in response to actual traffic patterns.

- On
- Off

Minimum capacity units	Maximum capacity units	Target utilization (%)
1	10	70

Initial provisioned units Info
5

Wrap-up: Reusable template



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Wrap-up: Production-ready prototype

Culture

Tool

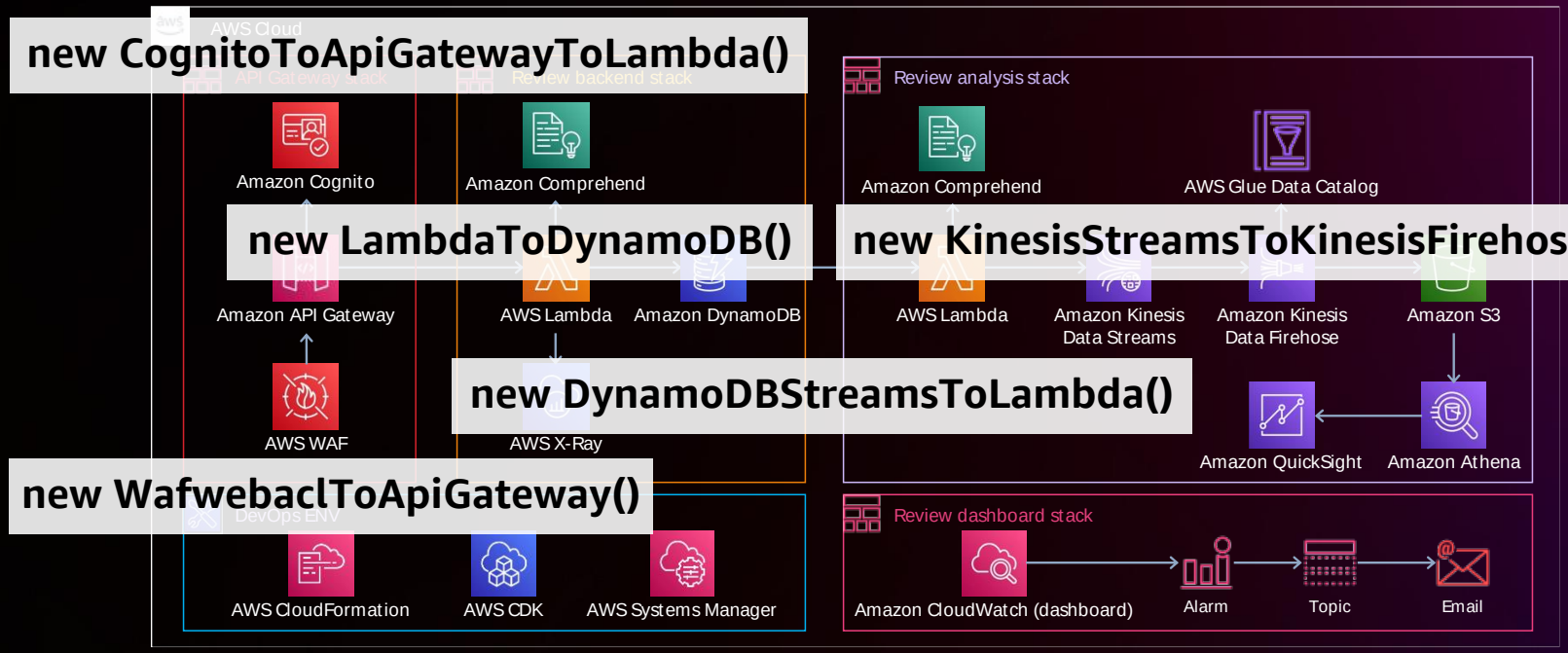
Practice

Functional requirements

Business context

Non-functional requirements

AWS Well-Architected



Cloud-native approach

Modern application

Iterative approach

Agile/Scrum

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Kwonyul Choi
kwonyulchoi (GitHub)
Kwonyul Choi (LinkedIn)



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.