aws DEV DAY
ONLINE JAPAN

B - 1

# AWS Amplify実践編

Jaga (Daisuke Nagayama) @jagaimogmog
Amazon Web Services Japan
Startup Solutions Architect

In Partnership with intel

## このセッションの想定視聴者

- すでにAWS Amplifyを触ったことがある人
- AWS Amplifyを本番環境でつかってみたいと思っている方

## このセッションでお伝えしたいこと

- AWS Amplifyをチーム開発で使うためのノウハウ
- AWS Amplify を使用して高速に開発するためのTips
  - API (GraphQL)カテゴリ
  - Functions カテゴリ
  - Amplify Mocking
- AWS Amplify 拡張シナリオ

aws DEV DAY
ONLINE JAPAN

# AWS Amplifyをチーム開発で使うノウハウ

In Partnership with intel

# プロダクトの成長とチーム開発

環境の分割
（prod/stg/dev）

環境の競合の防止

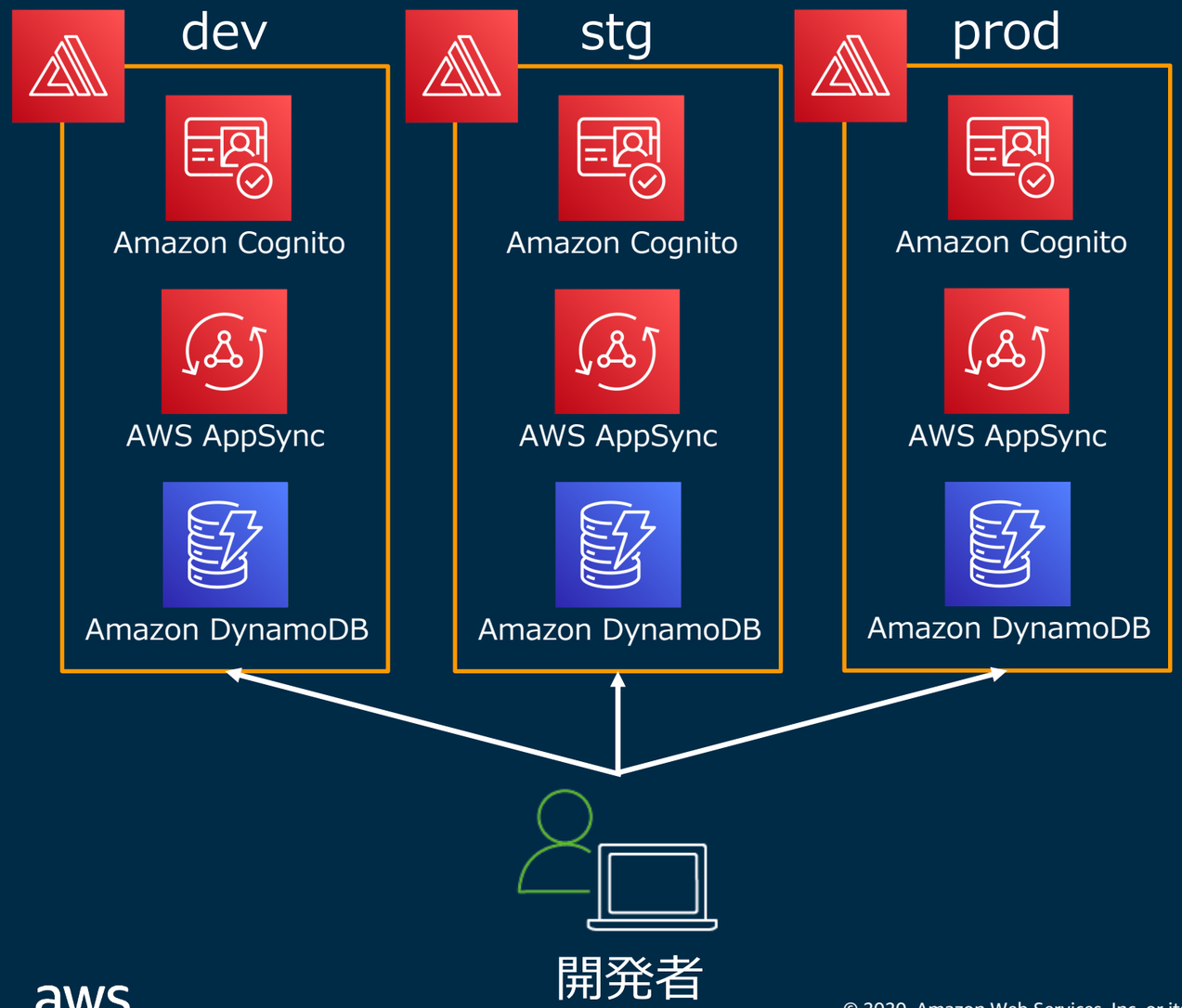
様々なステークホルダの加入

アカウント分割

開発環境の統制

# Amplify の multi env 機能で環境を分割する

環境の分割
(prod/stg/dev)




$ amplify env add
env を追加 (複数のアカウントにまたがることも可能)

$ amplify env checkout
env を切り替え

$ amplify env import
他の開発者が作成した env を参照

https://aws-amplify.github.io/docs/cli-toolchain/quickstart#environments-and-teams



In Partnership with intel

# envとブランチの紐付け

環境の分割
(prod/stg/dev)


env と Amplify Consoleで接続するブランチは1:Nの関係

Amplify Console でブランチを紐づける際、どの env と対応づけるかを選択可能( env の新規作成も可)

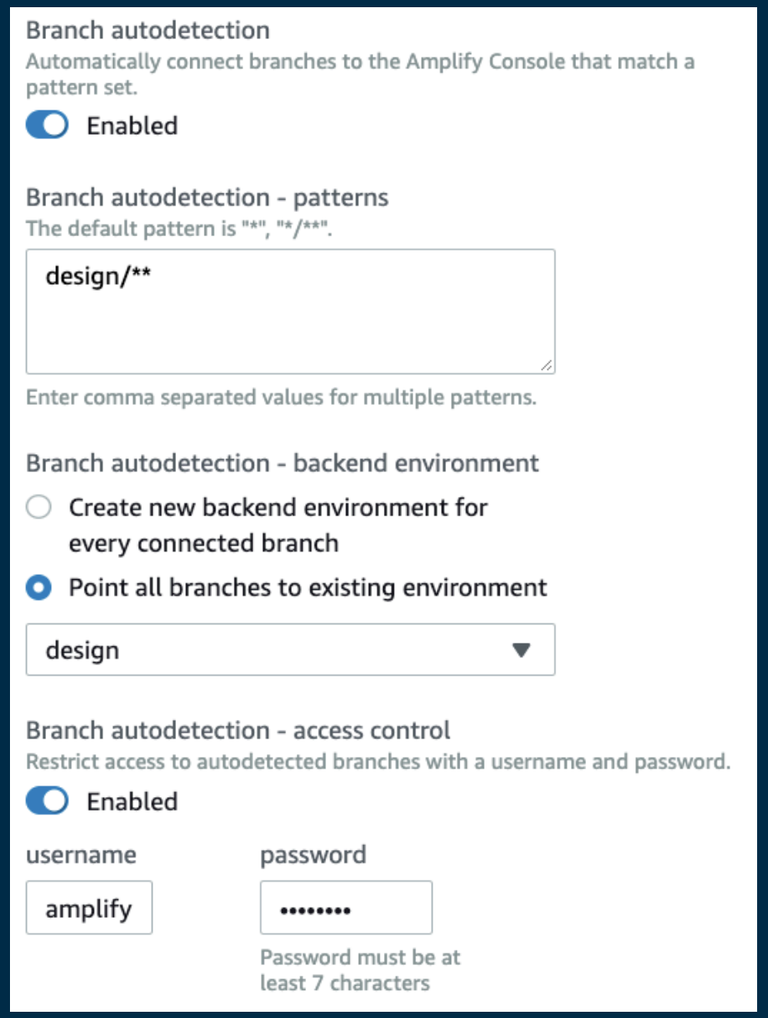
https://aws-amplify.github.io/docs/cli-toolchain/quickstart#environments-and-teams



In Partnership with intel

# ブランチベースの新しいenvのデプロイ

環境の分割
(prod/stg/dev)




特定のパターンに一致するブランチの作成時、自動的に
Amplify コンソールでデプロイ

「design/**」のようなパターンを定義して、
「design/」で始まる Git ブランチを自動的にデプロイ

接続するバックエンドの env は
新規作成 or 既存 env から選択

自動作成されたアプリにベーシック認証

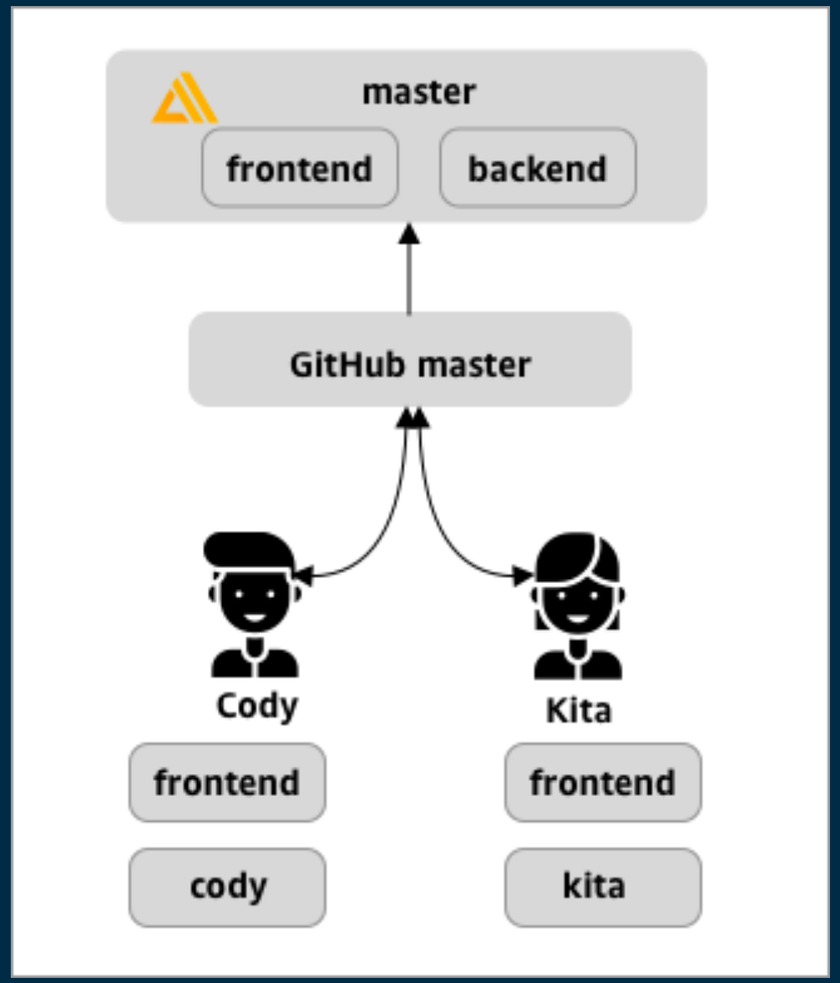
https://docs.aws.amazon.com/ja_jp/amplify/latest/userguide/multi-environments.html#pattern-based-branch-feature-branch-deployments



In Partnership with intel

# 開発者固有のenvで開発環境の競合を防ぐ

環境の競合の防止




Amplify では開発者ごとに env を分けることを推奨

開発者は他のチームメンバーの変更により競合することなく、互いに独立して作業することが可能

フロントエンドとバックエンドの両方に変更が入るような開発が並列に進めやすくなる

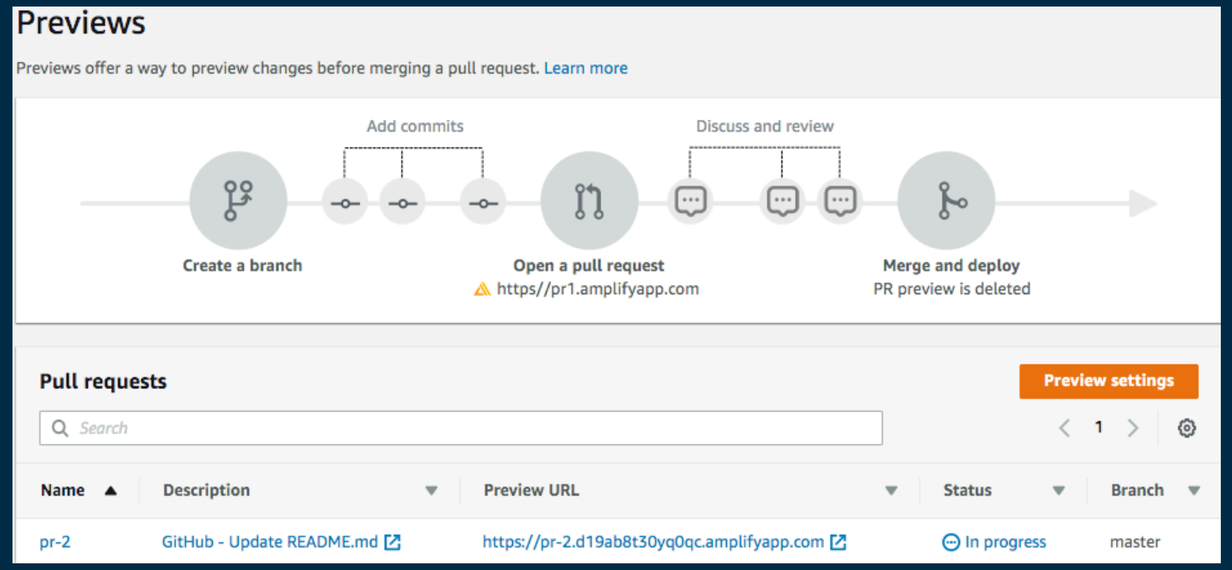
https://docs.aws.amazon.com/ja_jp/amplify/latest/userguide/multi-environments.html#sandbox

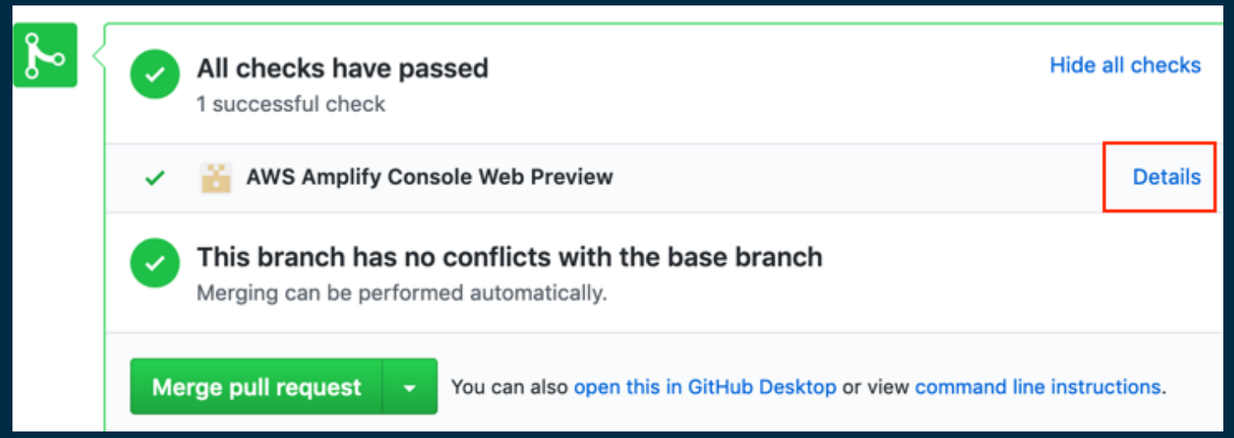


In Partnership with intel

# Pull Request Previews でレビューを高速化

様々なステークホルダの加入




Pull Request Previews を設定すると、Pull Request が作成されるたびに一時的なウェブサイトをホスト

ホストした URL を Git Hub の Pull Request ページに載せる

Pull Request レビューが非常に楽になり、非エンジニア職でもレビューに参加することが可能

Pull Request が閉じたら一時的なウェブサイトのホストも消える

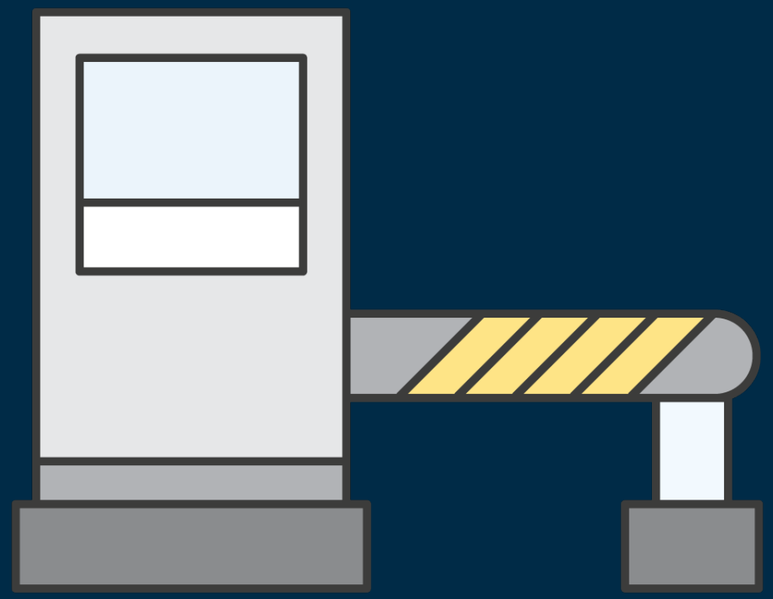
https://docs.aws.amazon.com/amplify/latest/userguide/pr-previews.html



In Partnership with intel

# Developer Experienceを損なわない環境統制

Gatekeeper



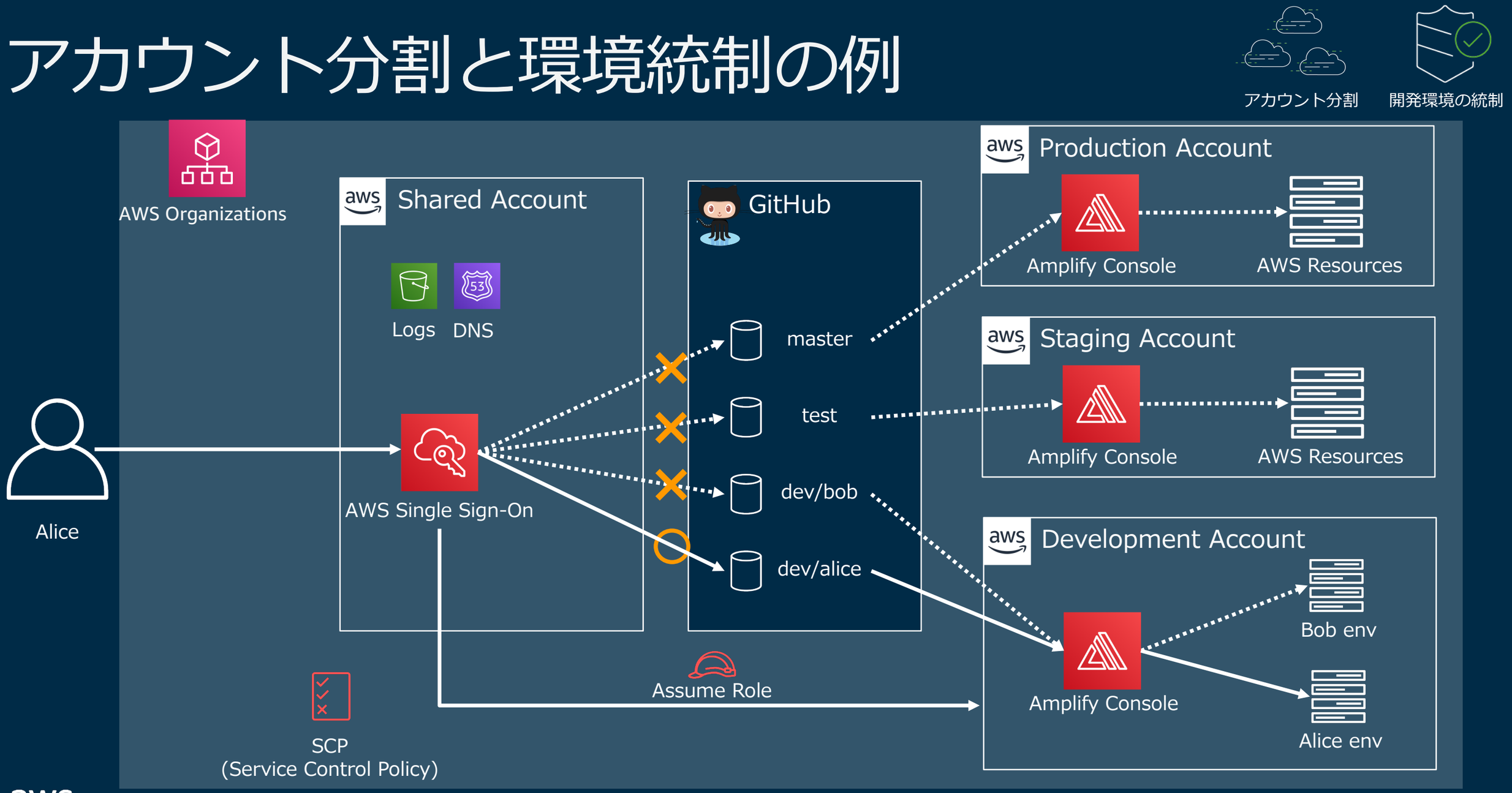
V.S.

Guardrail



In Partnership with intel

# アカウント分割と環境統制の例



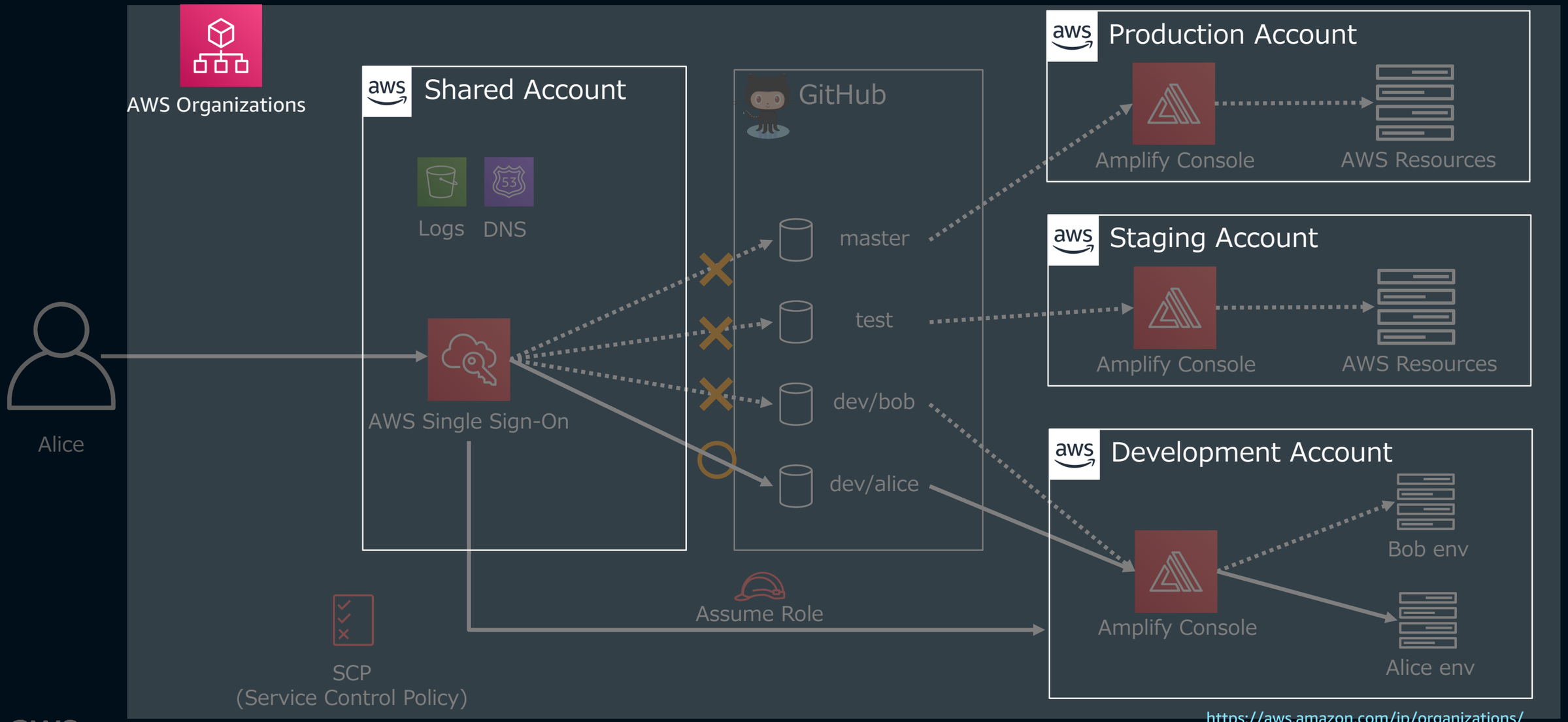



In Partnership with intel

# アカウント分割と環境統制の例


アカウント分割


開発環境の統制




https://aws.amazon.com/jp/organizations/

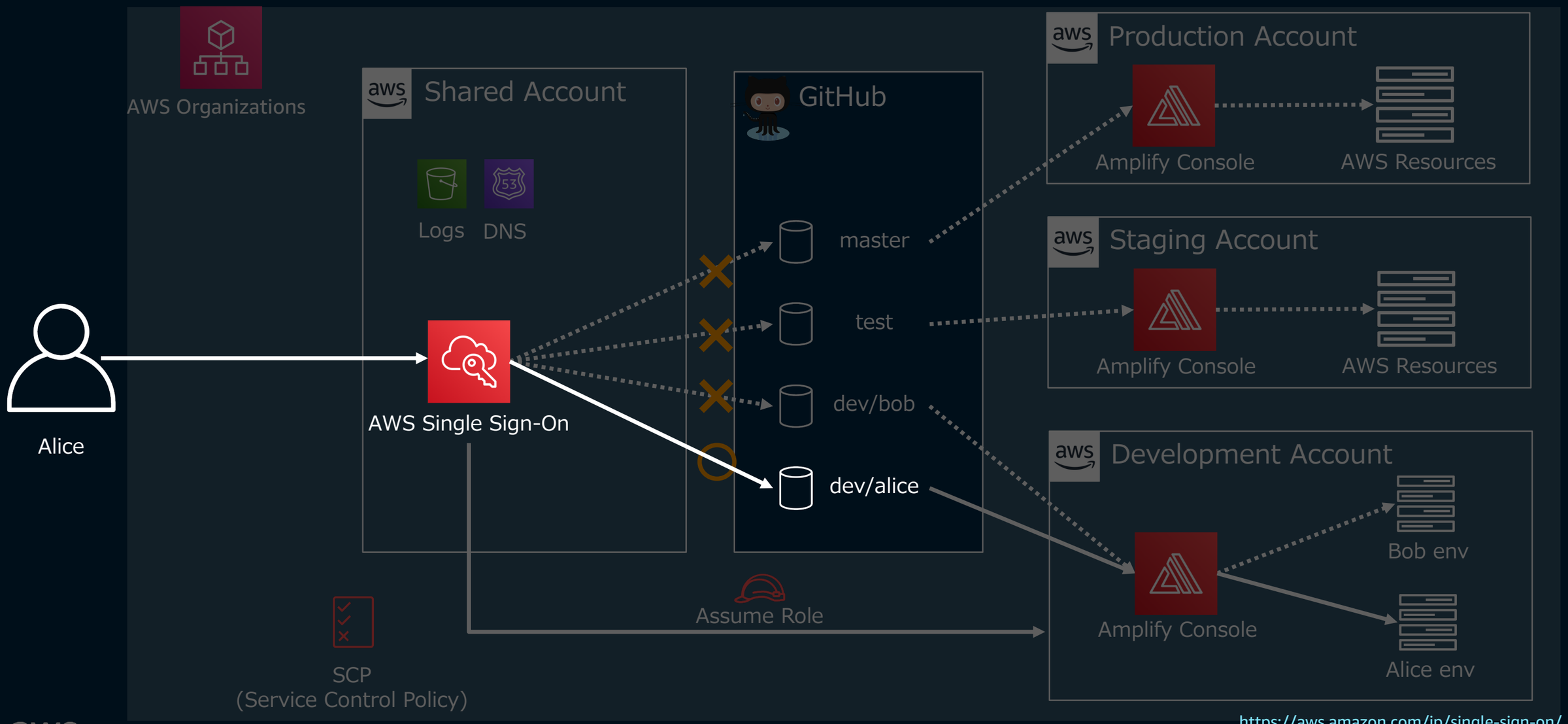


In Partnership with intel

# アカウント分割と環境統制の例


アカウント分割


開発環境の統制




https://aws.amazon.com/jp/single-sign-on/

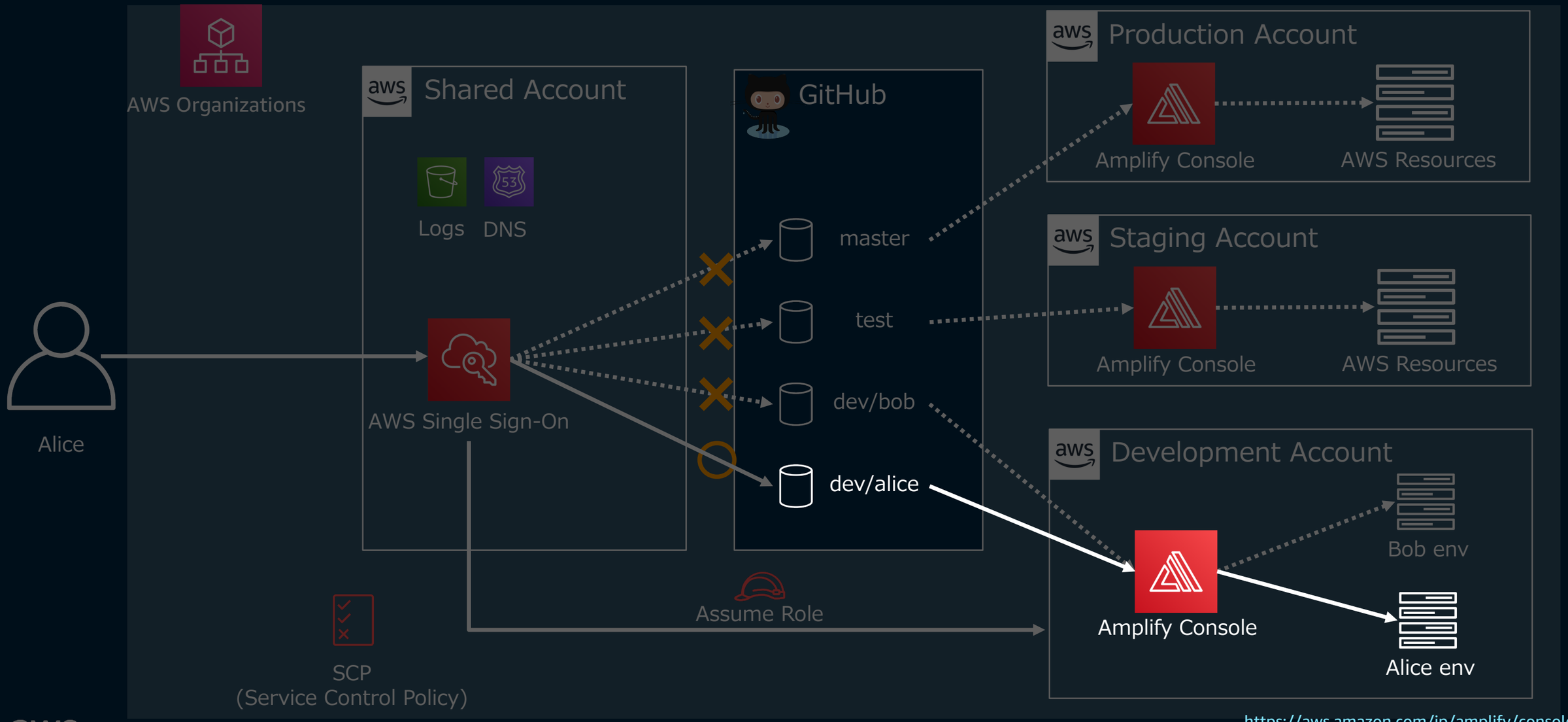


In Partnership with intel

# アカウント分割と環境統制の例


アカウント分割


開発環境の統制




https://aws.amazon.com/jp/amplify/console/

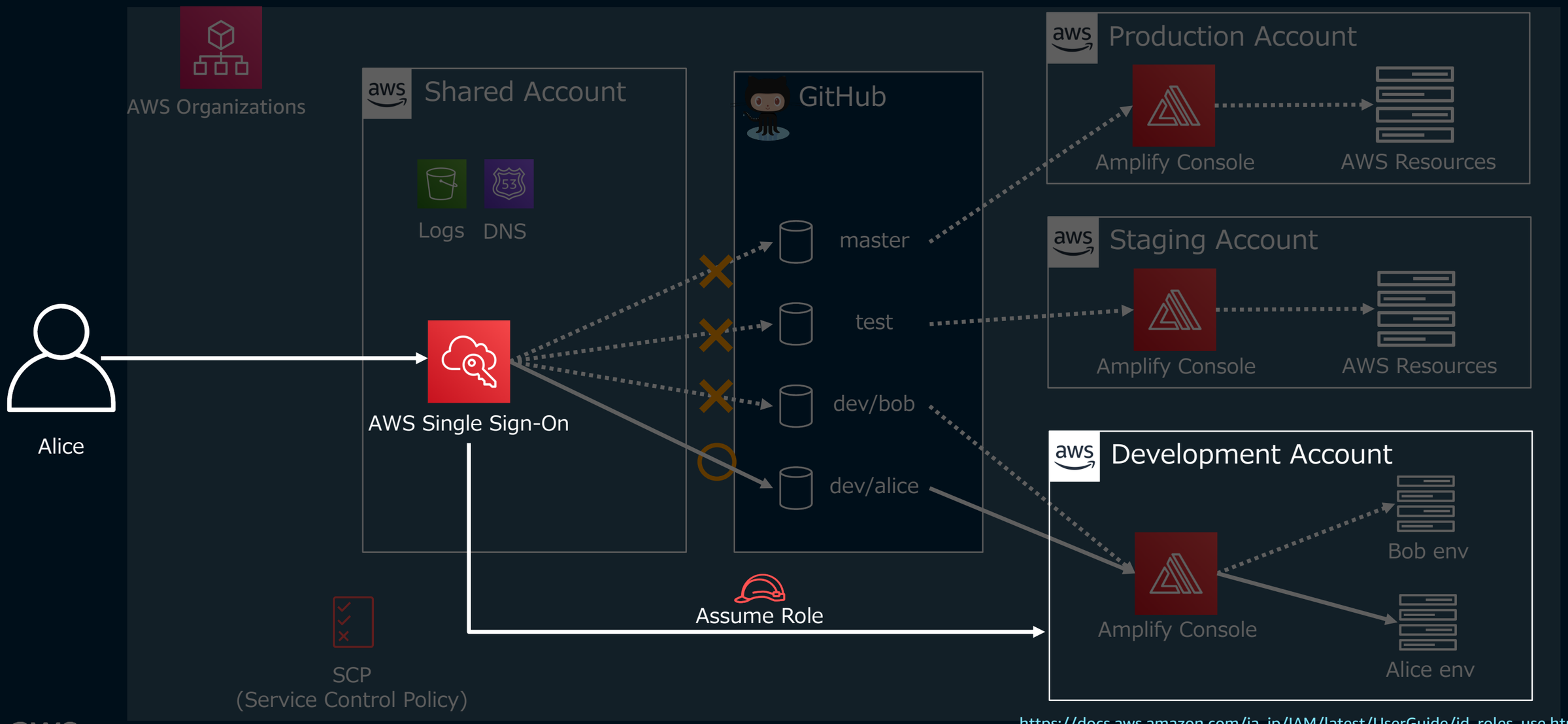


In Partnership with intel

# アカウント分割と環境統制の例


アカウント分割


開発環境の統制




https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/id_roles_use.html

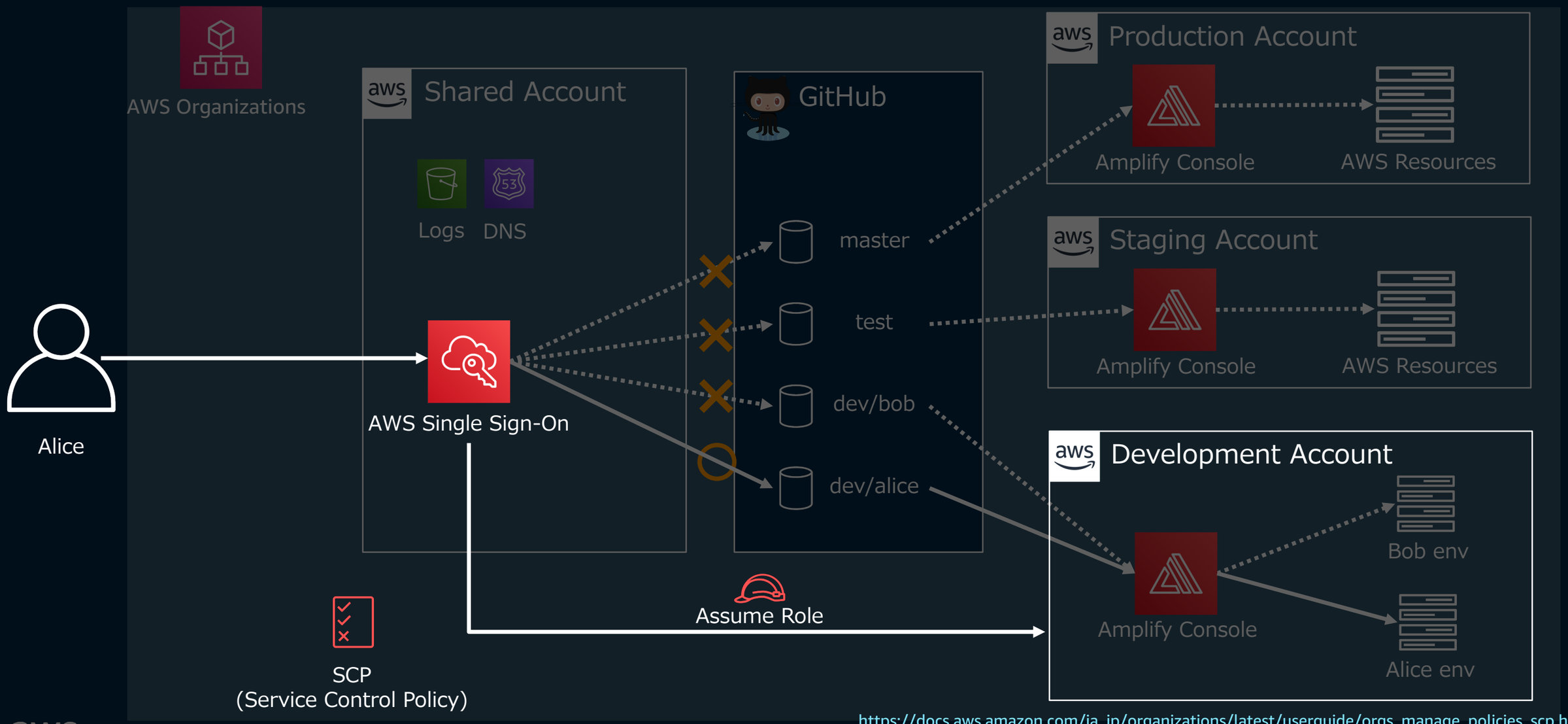


In Partnership with intel

# アカウント分割と環境統制の例


アカウント分割


開発環境の統制

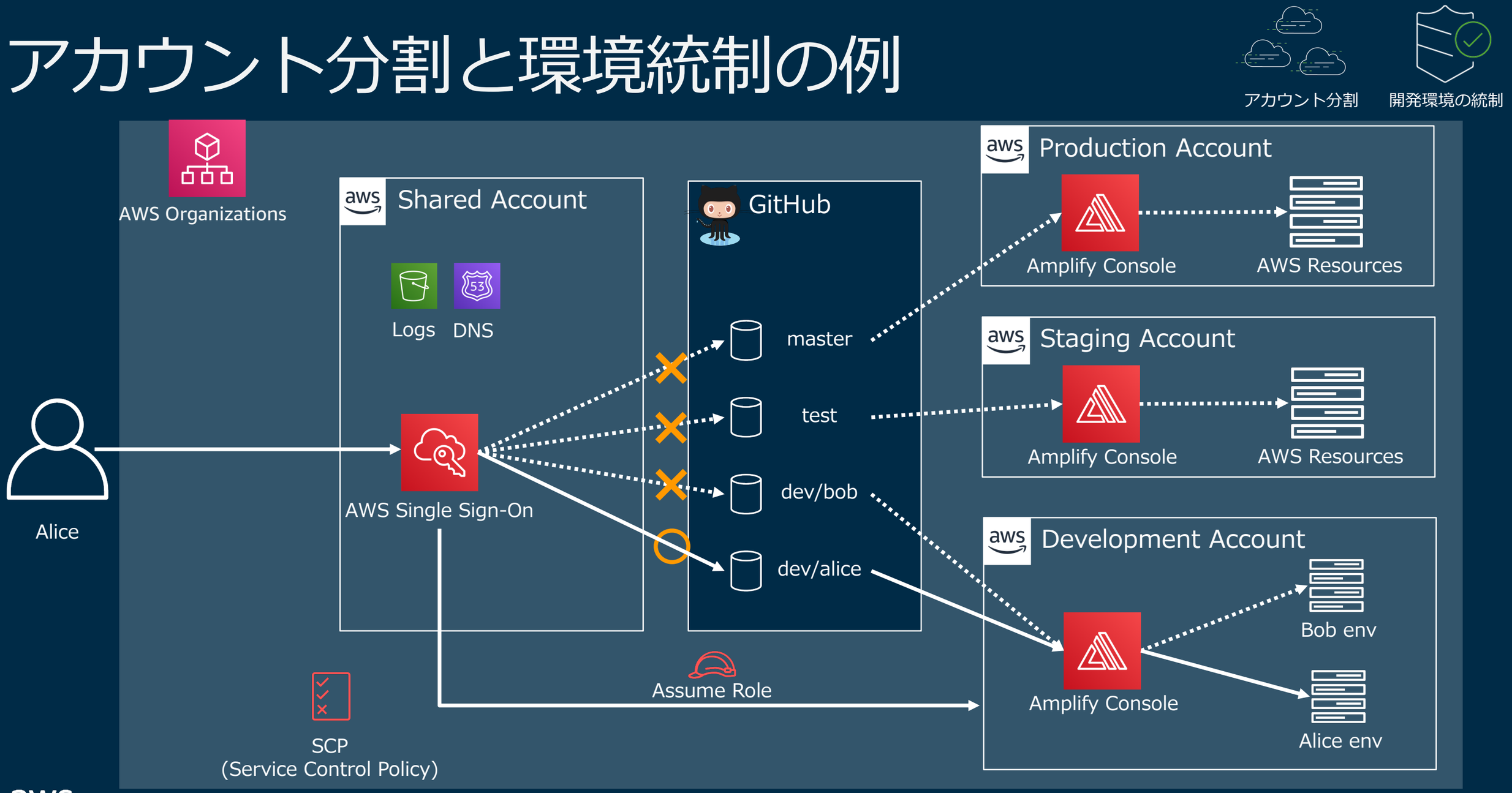



https://docs.aws.amazon.com/ja_jp/organizations/latest/userguide/orgs_manage_policies_scp.html



In Partnership with intel

# アカウント分割と環境統制の例





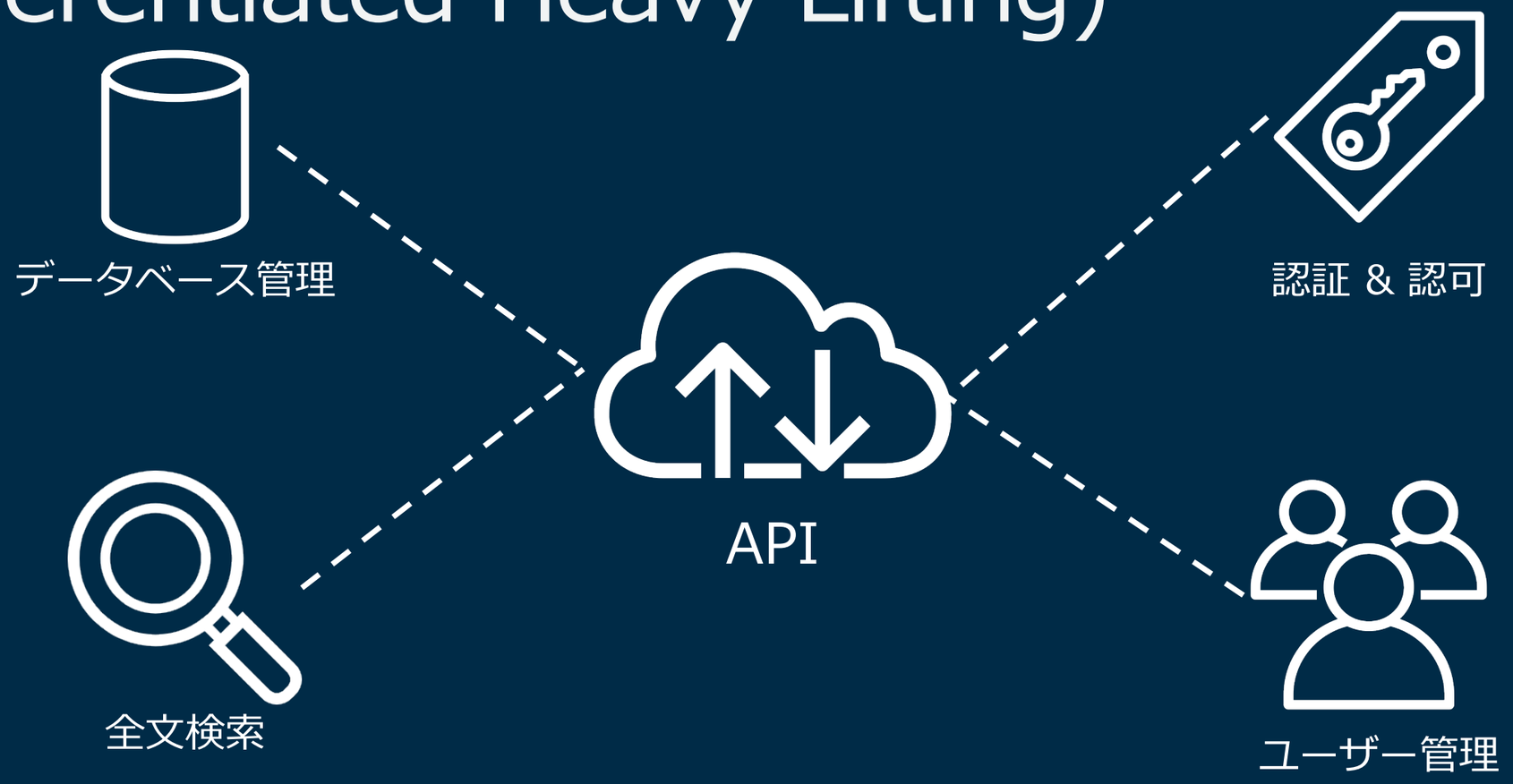

In Partnership with intel

aws DEV DAY
ONLINE JAPAN

# Amplifyで開発を高速化しよう！
# - API(GraphQL) -

In Partnership with intel

# API開発と付加価値を生まない重労働 (Undifferentiated Heavy Lifting)



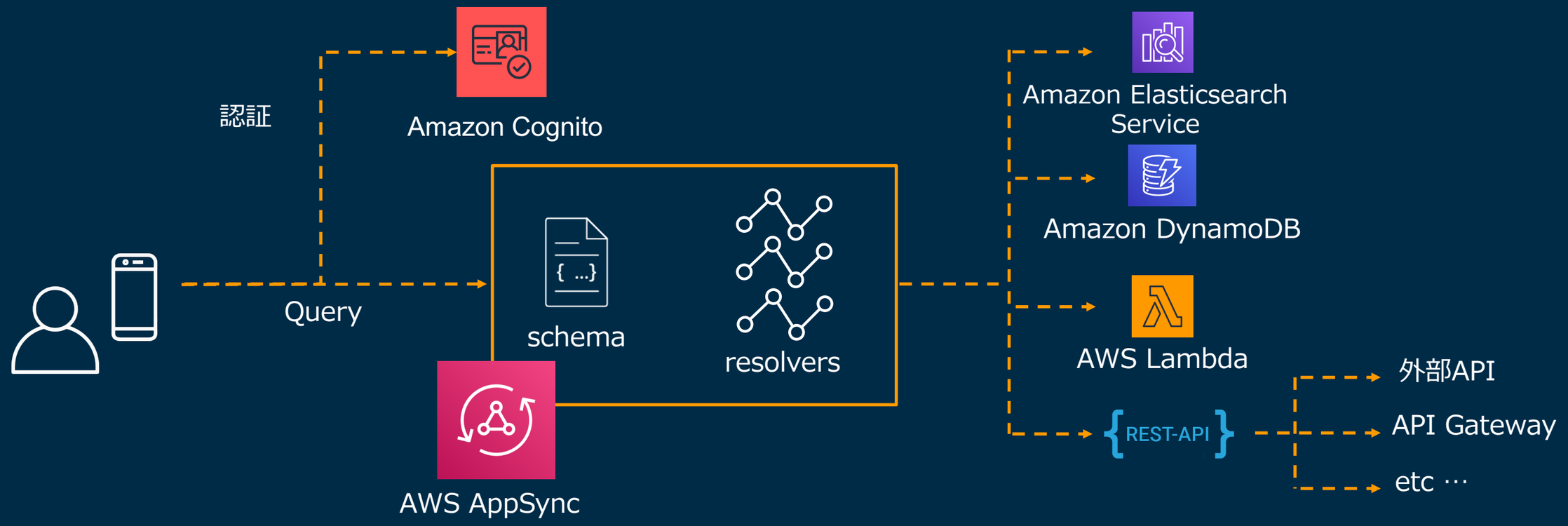

付加価値を生まない重労働を Amplify に任せて 価値提供にフォーカスしたい



In Partnership with intel

# API (GraphQL)

- API カテゴリ (GraphQL) は AWS AppSync と統合された API を構築
- GraphQL サーバーでは通常、schema と resolver (VTLファイル) を編集
- API (GraphQL) カテゴリを使うことでシンプルに設定が可能






In Partnership with intel

# 便利な Directive を活用する

```
type Post
  @model
{
  id: ID!
  name: String!
  description: String!
  owner: String!
  comments: [Comment]
  updatedAt: AWSDateTime!
  createdAt: AWSDateTime!
}
```

@model のように、schema.grahql ファイルにディレクティブを付与することで様々な機能を GraphQL APIに付与することが可能

本セクションでは以下5つを深掘り

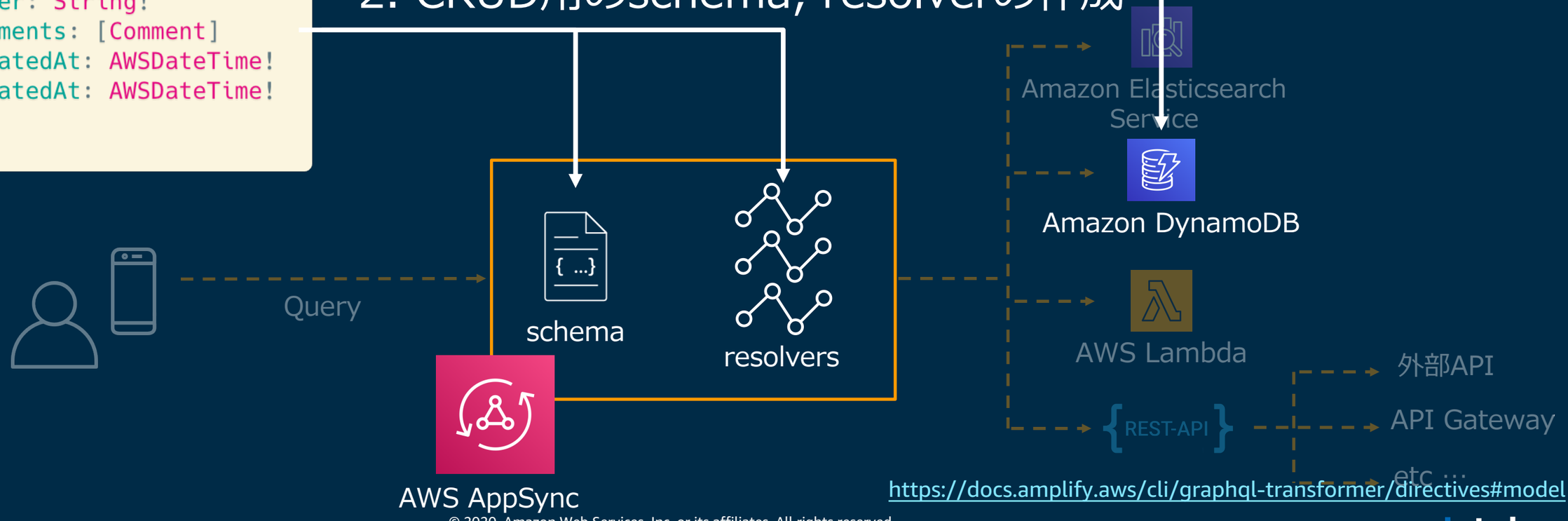
- @model
- @auth
- @key
- @searchable
- @function

# Directive #1 @model

```
type Post
  @model
{
  id: ID!
  name: String!
  description: String!
  owner: String!
  comments: [Comment]
  updatedAt: AWSDateTime!
  createdAt: AWSDateTime!
}
```

1. Postテーブルの作成

2. CRUD用のschema, resolverの作成




https://docs.amplify.aws/cli/graphql-transformer/directives#model



In Partnership with intel

# おまけ: @modelが生成するschema.graphql

```
type Mutation {
  createPost(input: CreatePostInput!, condition: ModelPostConditionInput): Post
  updatePost(input: UpdatePostInput!, condition: ModelPostConditionInput): Post
  deletePost(input: DeletePostInput!, condition: ModelPostConditionInput): Post
}

input ModelPostFilterInput {
  id: ModelIDInput
  name: ModelStringInput
  description: ModelStringInput
  owner: ModelStringInput
  updatedAt: ModelStringInput
  createdAt: ModelStringInput
  and: [ModelPostFilterInput]
  or: [ModelPostFilterInput]
  not: ModelPostFilterInput
}
```

https://docs.amplify.aws/cli/graphql-transformer/directives#model

# おまけ: @modelが生成するresolvers

```
$ tree amplify/backend/api/devday2020/build/resolvers/
amplify/backend/api/devday2020/build/resolvers/
├── Mutation.createPost.req.vtl
├── Mutation.createPost.res.vtl
├── Mutation.deletePost.req.vtl
├── Mutation.deletePost.res.vtl
├── Mutation.updatePost.req.vtl
├── Mutation.updatePost.res.vtl
├── Query.getPost.req.vtl
├── Query.getPost.res.vtl
├── Query.listPosts.req.vtl
└── Query.listPosts.res.vtl
```

```
## [Start] Set default values. **
$util.qr($context.args.input.put("id", $util.defaultIfNull($ctx.args.input.id, $util.autoId())))
#set( $createdAt = $util.time.nowISO8601() )
## Automatically set the createdAt timestamp. **
$util.qr($context.args.input.put("createdAt", $util.defaultIfNull($ctx.args.input.createdAt, $createdAt)))
## Automatically set the updatedAt timestamp. **
$util.qr($context.args.input.put("updatedAt", $util.defaultIfNull($ctx.args.input.updatedAt, $createdAt)))
## [End] Set default values. **
## [Start] Prepare DynamoDB PutItem Request. **
$util.qr($context.args.input.put("__typename", "Post"))
#set( $condition = {
  "expression": "attribute_not_exists(#id)",
  "expressionNames": {
      "#id": "id"
  }
} )
```

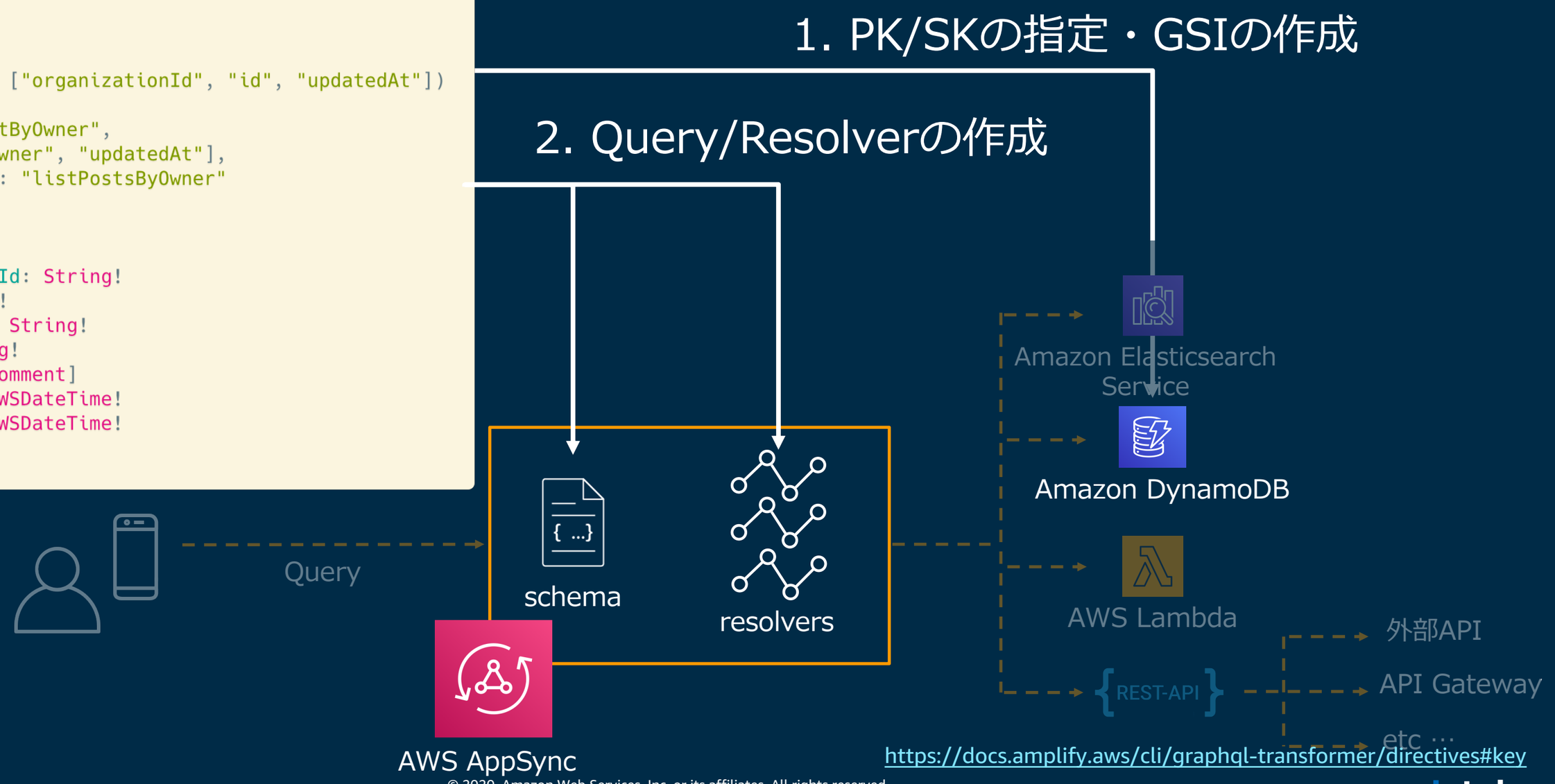
https://docs.amplify.aws/cli/graphql-transformer/directives#model



In Partnership with intel

# Directive #2 @key

```
type Post
  @model
  @key(fields: ["organizationId", "id", "updatedAt"])
  @key(
    name: "listByOwner",
    fields:["owner", "updatedAt"],
    queryField: "listPostsByOwner"
  )
{
  id: ID!
  organizationId: String!
  name: String!
  description: String!
  owner: String!
  comments: [Comment]
  updatedAt: AWSDateTime!
  createdAt: AWSDateTime!
}
```




https://docs.amplify.aws/cli/graphql-transformer/directives#key



In Partnership with intel

# Directive #2 @key

schema.graphql

```
type Post
  @model
  @key (fields:[organizationId, createdAt])
  @key (name: "listByOwner",
        fields:[owner, createdAt]
        queryField: "listPostsByOwner")
{
  id: ID!
  name: String!
  organizationId: Staring!
  description: String!
  owner: String!
  comments: [Comment]
  updatedAt: AWSDateTime
  createdAt: AWSDateTime
}
```

queryFieldを指定しない@keyではlistPostsなど@model で追加したCRUDに使える引数を指定することが可能

- Partition Key: organizationId
- Sort Key: createdAt

queryFieldを指定した@keyは、listPostsとは別のフィールドを用いたQueryを追加することが可能

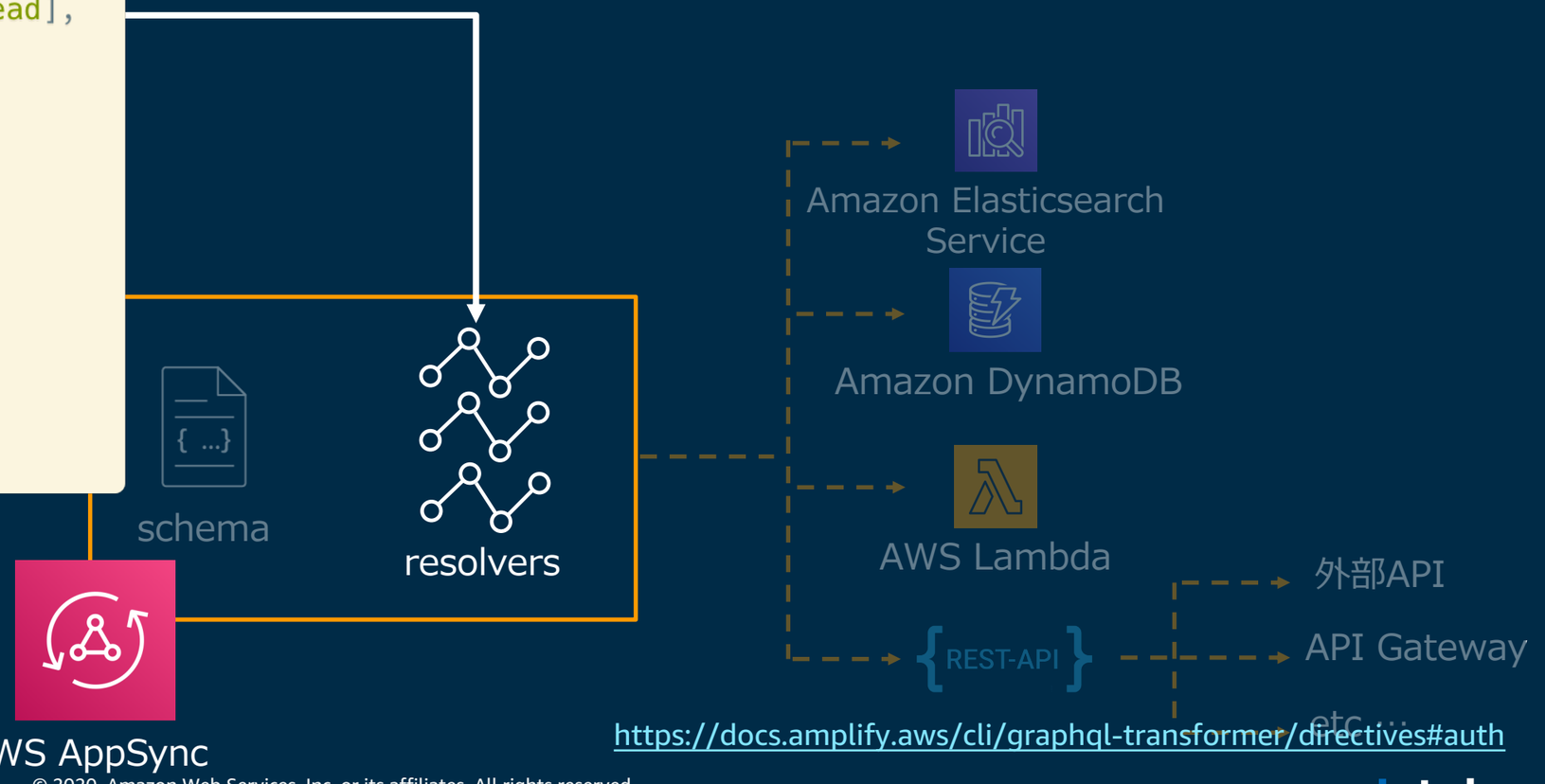
- Partition Key: owner
- Sort Key: createdAt
- インデックスの名前: listByOwner
- queryの名前: listPostsByOwner

https://docs.amplify.aws/cli/graphql-transformer/directives#key

# Directive #3 @auth

```
type Post
  @model
  @auth(rules: [
    {
      allow: owner,
      provider: userPools
      ownerField: "owner",
      operations: [create, update, delete, read],
    },
  ])
{
  id: ID!
  name: String!
  description: String!
  owner: String!
  comments: [Comment] @auth(rules:[{}])
  updatedAt: AWSDateTime!
  createdAt: AWSDateTime!
}
```

認可ロジックを Resolver に追加




https://docs.amplify.aws/cli/graphql-transformer/directives#auth



In Partnership with intel

# Directive #3 @auth

schema.graphql

```
type Post
  @model
  @auth(rules: [
    {
      allow: owner,
      provider: userPools
      ownerField: "owner",
      operations: [create, update, delete, read],
    },
  ])
{
  id: ID!
  name: String!
  description: String!
  owner: String!
  comments: [Comment] @auth(rules:[{}])
  updatedAt: AWSDateTime!
  createdAt: AWSDateTime!
}
```

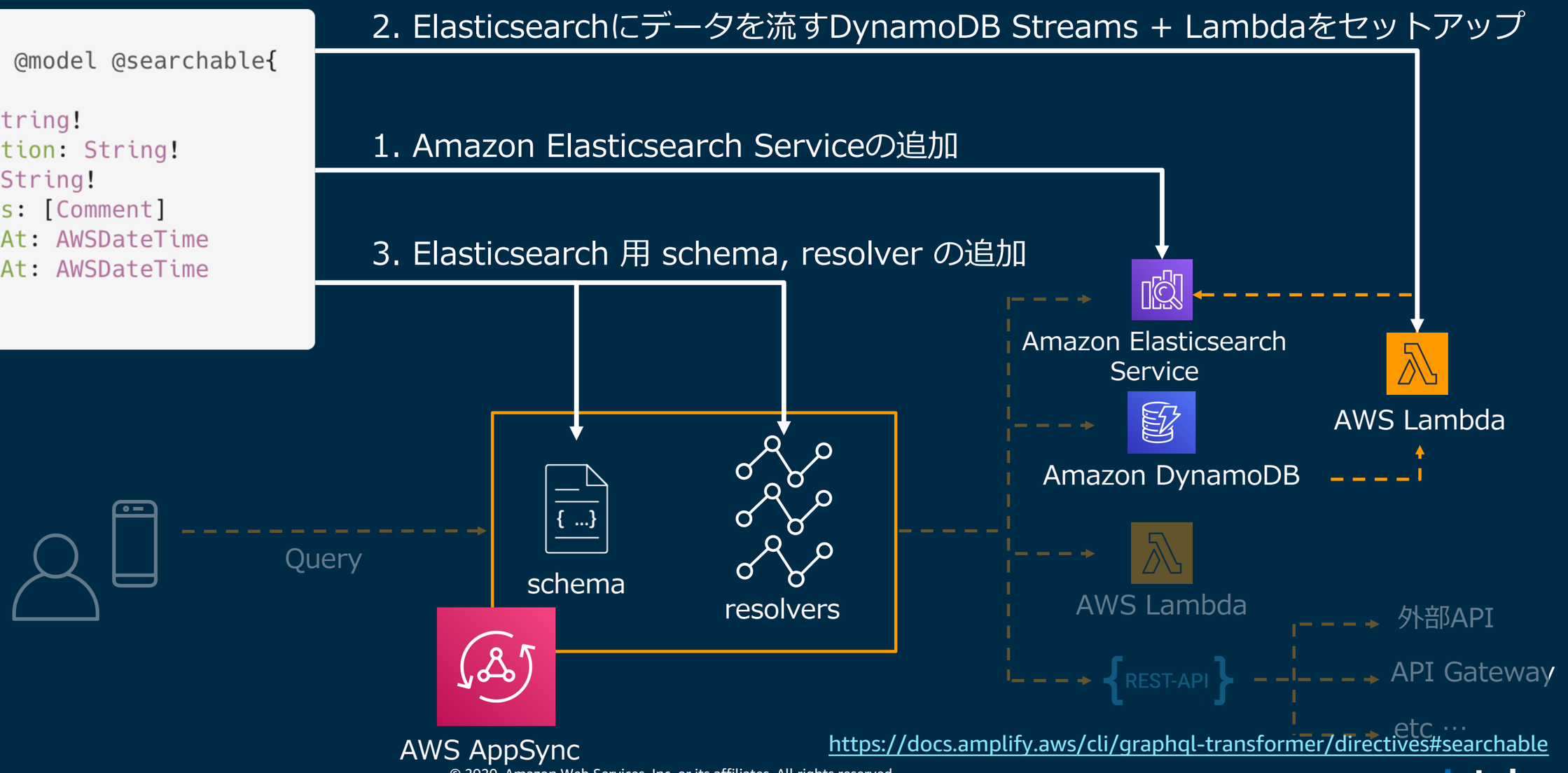
Amazon Cognito User Poolやサードパーティのの OIDCプロバイダによって認証されたユーザーに対し、ユーザーの認証メタデータを使用してGraphQL APIのアクションに対する認可ルールを設定

- allow: owner groups private public
- provider: apiKey iam oidc userPools
- operations: create update delete read
- Field Level Authorization

https://docs.amplify.aws/cli/graphql-transformer/directives#auth

# Directive #4 @searchable

```
type Post @model @searchable{
  id: ID!
  name: String!
  description: String!
  owner: String!
  comments: [Comment]
  updatedAt: AWSDateTime
  createdAt: AWSDateTime
}
```




https://docs.amplify.aws/cli/graphql-transformer/directives#searchable



In Partnership with intel

# Directive #4 @searchable

GraphQL Query

```
query SearchPosts {
  searchPost(filter: {
    title: { wildcard: "S*" }
    or: [
      { createdAt: { eq: "08/20/2018" } },
      { updatedAt: { eq: "08/20/2018" } }
    ]
  }) {
    items {
      id
      title
    }
  }
}
```

@searchableをつけるとElasticsearchの構文を利用した全文検索が可能に

Search<type名> という名前のQueryで呼び出し可能

裏ではAmazon Elasticsearch ServiceとDynamoDB Streamsをセットアップ

@searchableをつけた後のデータしか検索できないので注意

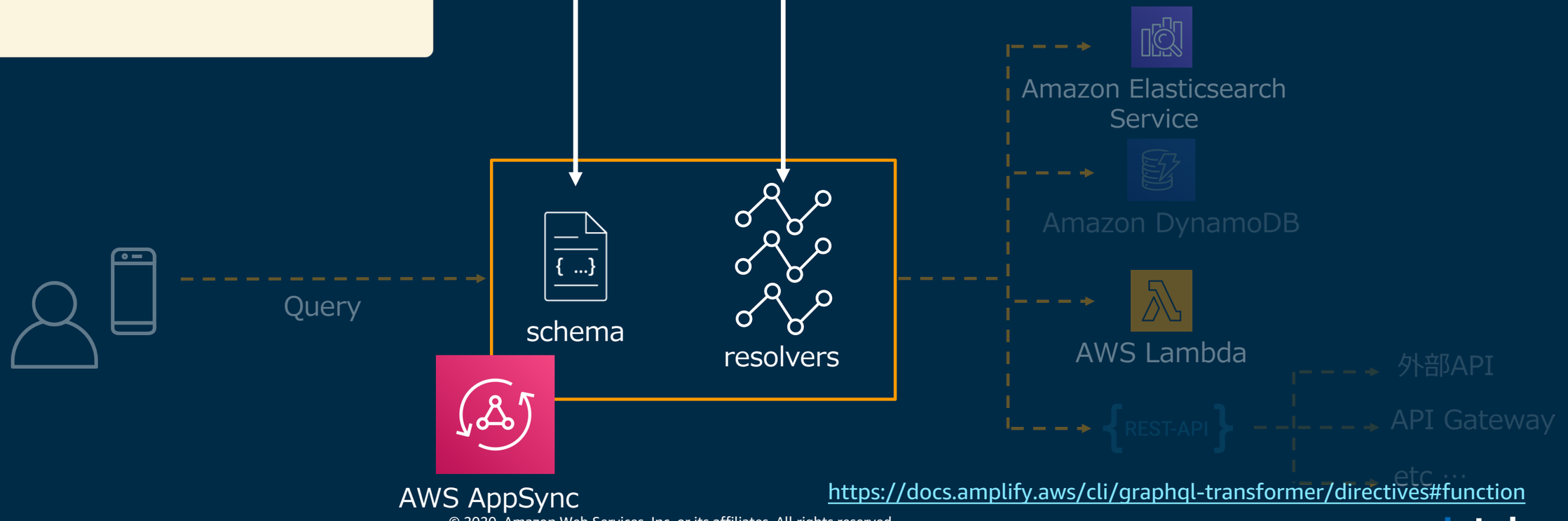
https://docs.amplify.aws/cli/graphql-transformer/directives#searchable

# Directive #5 @function

schema.graphql

```
type Mutation {
  addComment(comment: String): Comment
    @function(name: "authorizer-${env}")
    @function(name: "addComment-${env}")
}
```

Lambda 関数を呼び出すための schema、resolver の追加
(Lambda関数自体はFunctionsカテゴリで作成)




https://docs.amplify.aws/cli/graphql-transformer/directives#function



In Partnership with intel

# Directive #5 @function

schema.graphql

```
type Mutation {
  addComment(comment: String): Comment
    @function(name: "authorizer-${env}")
    @function(name: "addComment-${env}")
}
```

$ amplify add function で作成したLambda関数を呼び出し

末尾に env 名をつけることに注意

複数の@functionをチェインして、処理をパイプすることが可能 (authorizer を実行してから addCommentを実行するなど)

https://docs.amplify.aws/cli/graphql-transformer/directives#function

# Directive #5 @function

function/authorizer/arc/index.js (Lambda)

```
exports.handler = async (event) => {
  const owner = event.identify.claims
    [COGNITO_USERNAME_CLAIM_KEY];
  const comment = event.arguments.comment;
  ...
};
```

Lambda関数には event 引数で呼び出し元の認証情報や、GraphQLのオペレーションを呼ぶ際に渡した引数が格納されている

https://docs.amplify.aws/cli/graphql-transformer/directives#function



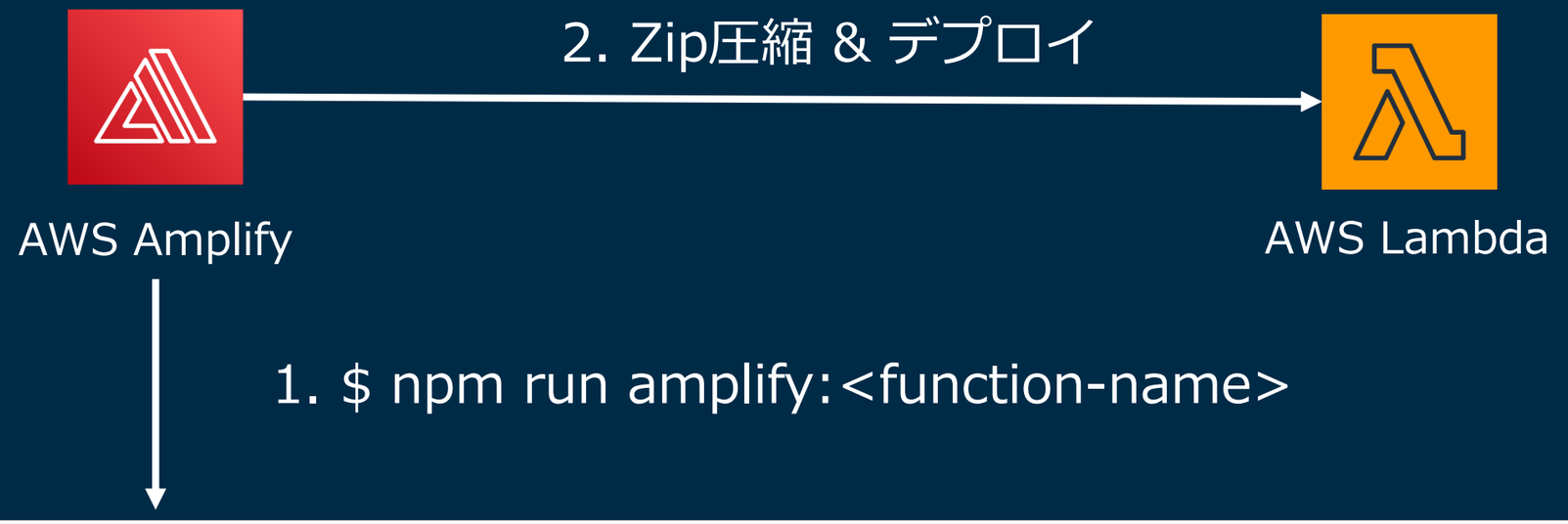
In Partnership with intel

aws DEV DAY
ONLINE JAPAN

# Amplifyで開発を高速化しよう！
# - function -

In Partnership with intel

# デプロイ時にTypeScriptのbuildを自動で走らせる



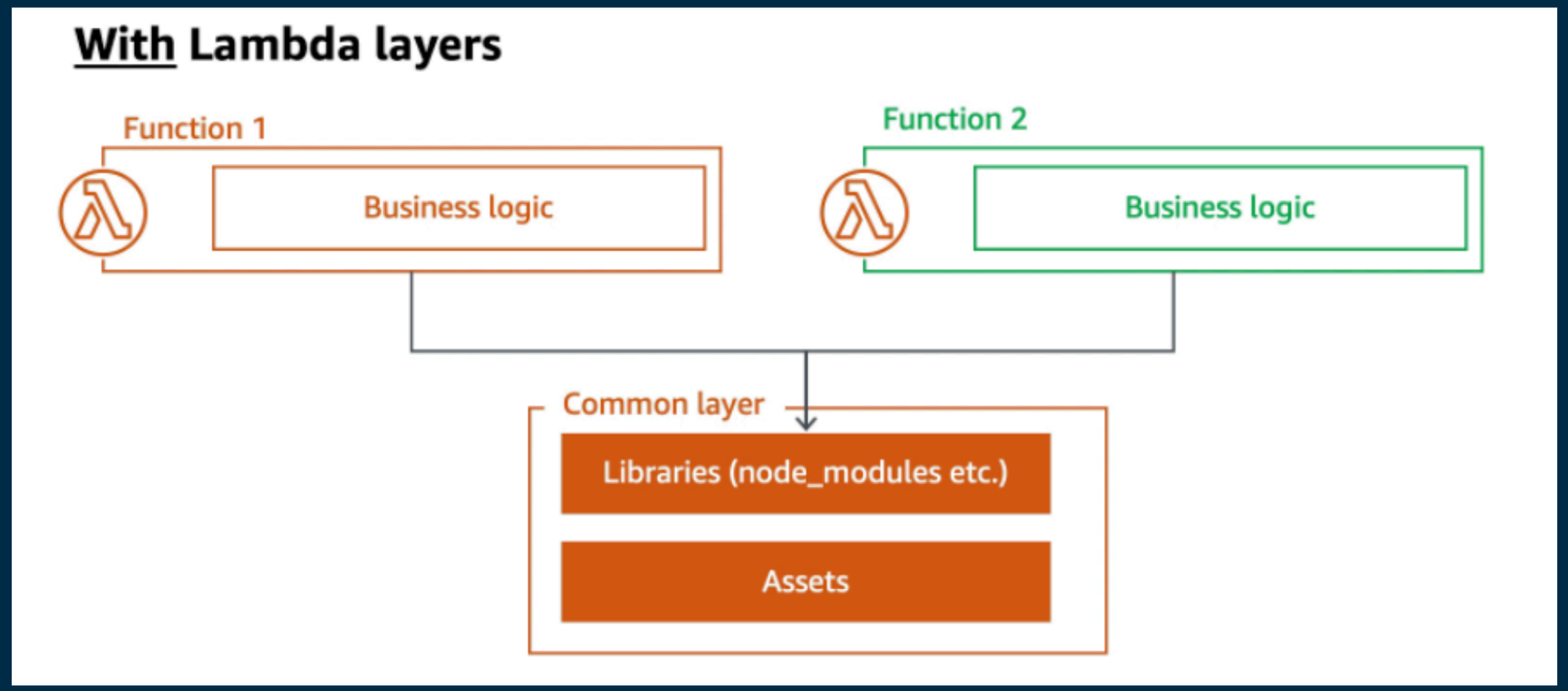

```
{
  "scripts": {
    "amplify:<function-name>": "cd amplify/backend/function/<function-name> && tsc -p ./tsconfig.json
&& cd -"
  },
}
```

https://docs.amplify.aws/cli/function/build-options



In Partnership with intel

# Lambda Layers でライブラリや共通処理をまとめる




- 現時点で Node.js と Python に対応
- 1つのLambda関数に5つまでのLambda Layersを追加可能
- Amplify CLI で Lambda Layers を作成することも、Amplify CLI 外で作成した Lambda Layers を import して使うことも可能

https://docs.amplify.aws/cli/function/layers



In Partnership with intel

# Lambda Layers でライブラリや共通処理をまとめる

```
$ amplify add function
? Select which capability you want to add:
  Lambda function (serverless function)
> Lambda layer (shared code & resource used across functions)
```

```
$ amplify add function
? Select which capability you want to add: Lambda function (serverless function)
? Provide a friendly name for your resource to be used as a label for this category in the project: devday202004b7eed0
? Provide the AWS Lambda function name: devday202004b7eed0
? Choose the runtime that you want to use: NodeJS
? Choose the function template that you want to use: Hello World
? Do you want to access other resources in this project from your Lambda function? No
? Do you want to invoke this function on a recurring schedule? No
? Do you want to configure Lambda layers for this function? Yes
? Provide existing layers or select layers in this project to access from this function (pick up to 5):
 ◯ Provide existing Lambda layer ARNs
>◉ devday202002fad996
 ◯ devday202074728e4a
```

https://docs.amplify.aws/cli/function/layers



In Partnership with intel

# 定期的に実行するLambda関数の作成

```
$ amplify add function
? Select which capability you want to add: Lambda function (serverless function)
? Provide a friendly name for your resource to be used as a label for this category in the project: devday2020f8dcc061
? Provide the AWS Lambda function name: devday2020f8dcc061
? Choose the runtime that you want to use: NodeJS
? Choose the function template that you want to use: Hello World
? Do you want to access other resources in this project from your Lambda function? No
? Do you want to invoke this function on a recurring schedule? Yes
? At which interval should the function be invoked: (Use arrow keys)
❯ Minutes
  Hourly
  Daily
  Weekly
  Monthly
  Yearly
  Custom AWS cron expression
```

https://aws.amazon.com/blogs/mobile/how-to-schedule-recurring-lambda-functions-using-the-amplify-cli/



In Partnership with intel

# Lambda Trigger の作成

```
$ amplify add function
? Select which capability you want to add: Lambda function (serverless function)
? Provide a friendly name for your resource to be used as a label for this category in the project: devday2020c5b0555c
? Provide the AWS Lambda function name: devday2020c5b0555c
? Choose the runtime that you want to use: NodeJS
? Choose the function template that you want to use: Lambda trigger
? What event source do you want to associate with Lambda trigger? Amazon Kinesis Stream
? Choose a Kinesis event source option (Use arrow keys)
❯ Use Analytics category kinesis stream in the current Amplify project
  Provide the ARN of Kinesis stream directly
```

```
$ amplify add function
? Select which capability you want to add: Lambda function (serverless function)
? Provide a friendly name for your resource to be used as a label for this category in the project: devday2020a1f57cce
? Provide the AWS Lambda function name: devday2020a1f57cce
? Choose the runtime that you want to use: NodeJS
? Choose the function template that you want to use: Lambda trigger
? What event source do you want to associate with Lambda trigger? Amazon DynamoDB Stream
? Choose a DynamoDB event source option (Use arrow keys)
❯ Use API category graphql @model backed DynamoDB table(s) in the current Amplify project
  Use storage category DynamoDB table configured in the current Amplify project
  Provide the ARN of DynamoDB stream directly
```

aws DEV DAY
ONLINE JAPAN

# Amplify で開発を高速化しよう！
# - Amplify Mocking -

In Partnership with intel

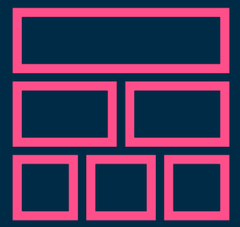
# APIの開発を効率化するには？

schema.graphql

```
type Post
  @model
{
  id: ID!
  name: String!
  description: String!
  owner: String!
  comments: [Comment]
  updatedAt: AWSDateTime!
  createdAt: AWSDateTime!
}
```

$ amplify push

1. クラウドリソースの参照と差分確認
2. 設定ファイルの書き出し
3. クラウドリソースへ変更を反映



AWS CloudFormation Stacks

時間をかけずに変更結果の確認がしたい！

# Amplify CLI – Mock commands

```
$ amplify mock

$ amplify mock api

$ amplify mock storage

$ amplify mock function <function-name>
```

https://aws-amplify.github.io/docs/cli-toolchain/usage?sdk=js#mocking-and-testing



In Partnership with intel

# API Mocking (1)

## $ amplify mock api すると…

### ①GraphQL transformationを実行

- schema.graphqlから以下を作成
  - AppSyncに関する設定ファイル (schema, resolvers, CFn templateなど)
  - フロントエンドからGraphQLを呼び出す際に使うJavaScriptのコード

### ②Amazon DynamoDB Localの立ち上げ

- データはamplify/mock-data/fake-****.db のSQLiteに保存
- データはvscode-sqliteなどのプラグインで確認・編集可能

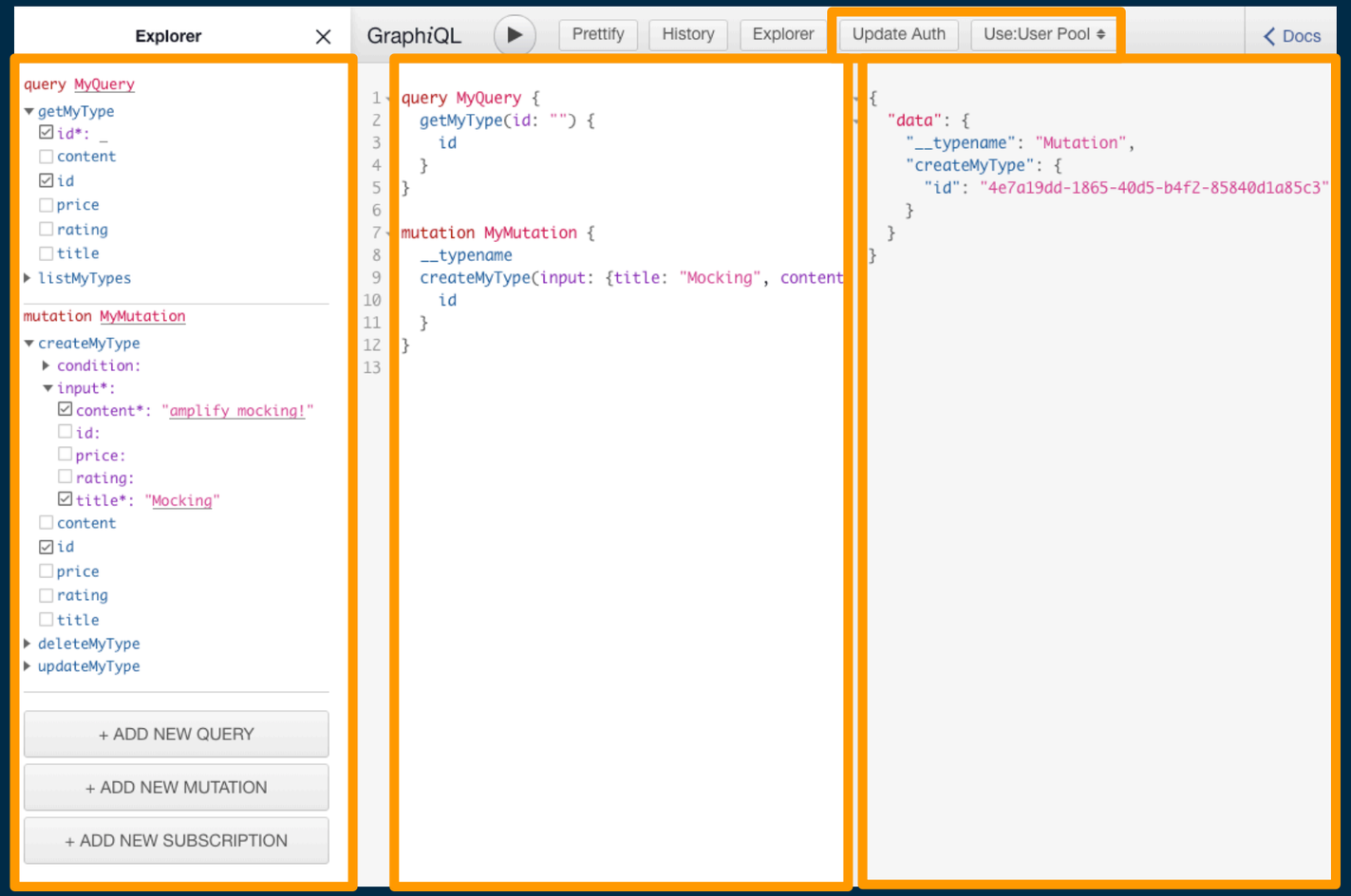
https://aws.amazon.com/jp/blogs/aws/new-local-mocking-and-testing-with-the-amplify-cli/

# API Mocking (2)

## $ amplify mock api すると…

### ③Amplify GraphiQL Explorerの立ち上げ

- OneGraph graphiql-explorer をベースに開発
- http://localhost:20002で立ち上る
- Query, Mutation, Subscriptionを簡単に実行可能なインタフェース
- Update AuthでAPI(GraphQL)の認証方法(Amazon Cognito User Pool/OIDC/API KEY/IAM)を切り替え可能




https://aws.amazon.com/jp/blogs/aws/new-local-mocking-and-testing-with-the-amplify-cli/



In Partnership with intel

# API Mocking (3)

## $ amplify mock api すると…

④GraphQL Endpointの立ち上げ

⑤aws-exportsをMock用に更新

- Amplify Frameworkでは、Amplify CLIから吐き出されるaws-exportsでクライアントの初期設定を行う
- $ amplify mock api してる間だけ、mockが終わるとクラウドのリソースを指す

src/index.jsx

```
import awsmobile from 'aws-exports';
import Amplify from 'aws-amplify';
Amplify.configure(awsmobile);
```

src/aws-exports.js

```
const awsmobile = {
    "aws_project_region": "us-east-1",
    "aws_appsync_graphqlEndpoint":
        "https://example.appsync-api.us-west-2.amazonaws.com/graphql",
    "aws_appsync_region": "us-east-1",
    "aws_appsync_authenticationType": "API_KEY",
    "aws_appsync_apiKey": "da2-XXXXXXXXXXX",
};
```

Mock

src/aws-exports.js

```
const awsmobile = {
    "aws_project_region": "us-east-1",
    "aws_appsync_graphqlEndpoint": "http://192.168.1.23:20002/graphql",
    "aws_appsync_region": "us-east-1",
    "aws_appsync_authenticationType": "API_KEY",
    "aws_appsync_apiKey": "da2-fakeApiId123456",
    "aws_appsync_dangerously_connect_to_http_endpoint_for_testing": true
};
```



https://aws.amazon.com/jp/blogs/aws/new-local-mocking-and-testing-with-the-amplify-cli/

aws DEV DAY
ONLINE JAPAN

# Amplify 拡張シナリオ

In Partnership with intel

# プロダクトの成長と機能開発

SNS認証

分析基盤

カスタマー
エンゲージメント

機械学習



In Partnership with intel

# OAuthでSNS認証を組み込みたい(1)

```
$ amplify add auth
Using service: Cognito, provided by: awscloudformation

 The current configured provider is Amazon Cognito.

 Do you want to use the default authentication and security configuration? Default configuration with Social Provider (Federation)
 Warning: you will not be able to edit these selections.
 How do you want users to be able to sign in? Username
 Do you want to configure advanced settings? No, I am done.
 What domain name prefix do you want to use? devday2020testc03d9135-c03d9135
 Enter your redirect signin URI: https://example.com/signin/
 Do you want to add another redirect signin URI No
 Enter your redirect signout URI: https://example.com/signout/
 Do you want to add another redirect signout URI No
 Select the social providers you want to configure for your user pool: Facebook

 You've opted to allow users to authenticate via Facebook.  If you haven't already, you'll need to go to https://developers.facebook.co
m and create an App ID.

 Enter your Facebook App ID for your OAuth flow:  test_facebook_ID
 Enter your Facebook App Secret for your OAuth flow:  test_facebook_secret
```

https://aws-amplify.github.io/docs/js/authentication#oauth-and-hosted-ui



In Partnership with intel

# OAuthでSNS認証を組み込みたい(2)

```
import Auth from '@aws-amplify/auth';

...

<div className="App">
   <button onClick={() => Auth.federatedSignIn({provider: 'Facebook'})}>Facebook Sign-in</button>
   <button onClick={() => Auth.signOut()}>Sign Out</button>
</div>
```

https://aws-amplify.github.io/docs/js/authentication#oauth-and-hosted-ui

# カスタマーエンゲージメントを組み込みたい

$ amplify add analytics

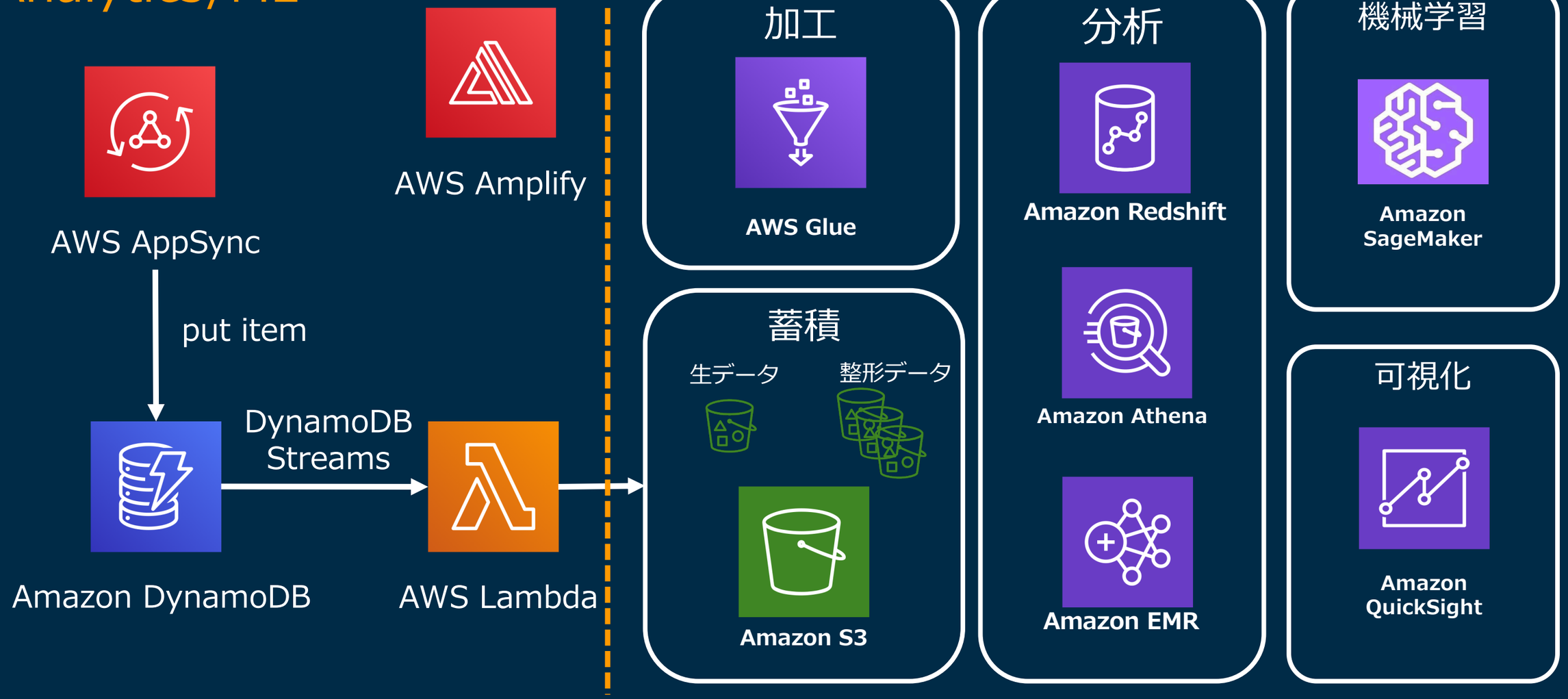

https://speakerdeck.com/jaguar_imo/startup-fm-amazon-pinpoint



In Partnership with intel

# アプリから得られたデータを分析したい

## Analytics/ML




https://aws.amazon.com/jp/big-data/datalakes-and-analytics/what-is-a-data-lake/



In Partnership with intel

# Custom CloudFormation Stacks

## amplify/backend/backend-config.json

```
{
  "auth": {
    "startupday202037dad6d6": {
      "service": "Cognito",
      "providerPlugin": "awscloudformation",
      "dependsOn": []
    }
  },
  "<custom-category-name>": {
    "<custom-resource-name>": {
      "service": <custom-aws-service-name>,
      "providerPlugin": "awscloudformation",
      "dependsOn": [{
            "category": "auth",
            "resourceName": "startupday202037dad6d6",
            "attributes": [
                "UserPoolId"
            ]
      }]
    }
  },
}
```

## Directory Structure

```
amplify/
└backend/
  ├backend-config.json
  ├auth/
  │ └startupday202037dad6d6/
  │   ├parameters.json
  │   └startupday202037dad6d6-cloudformation-template.yml
  └<custom-category-name>/
    └<custom-resource-name>/
      ├parameters.json
      └template.yml
```

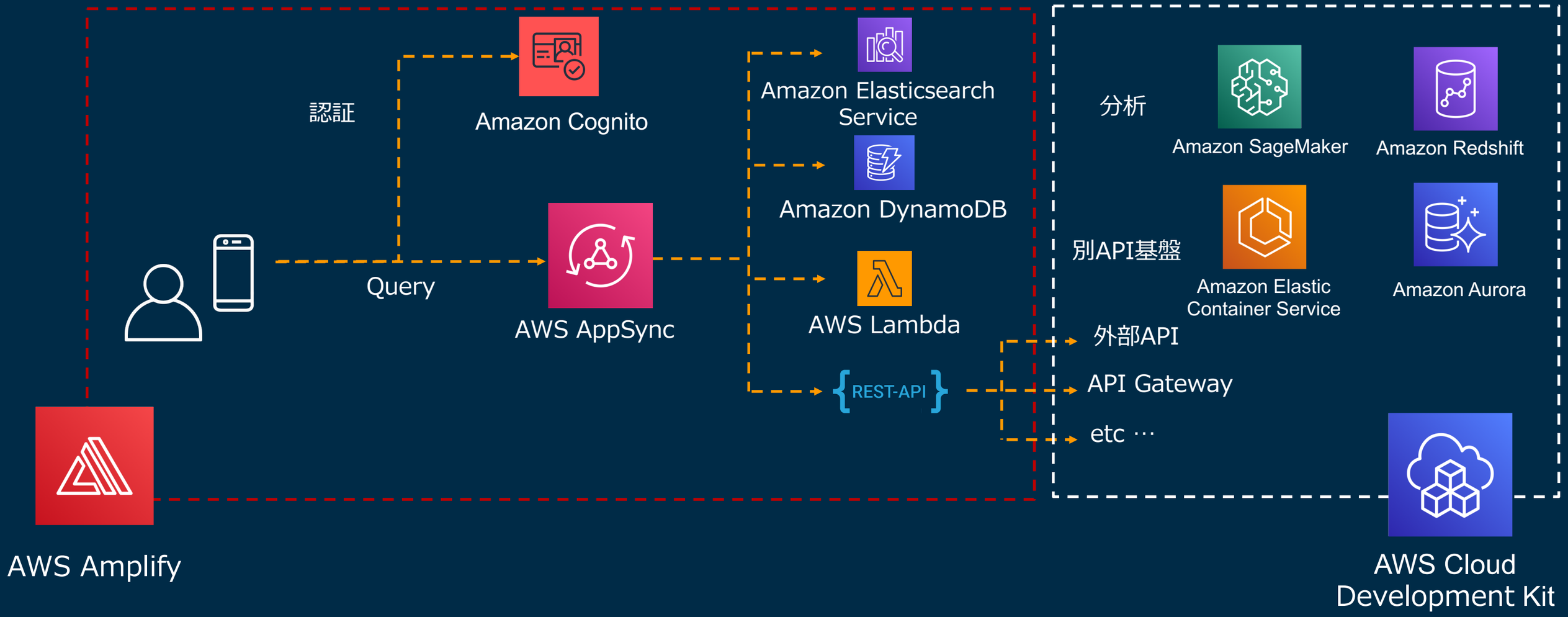
backend-config.json にリソースの情報を書く

CloudFormation Template を自由に書いてリソースを足す

env 変数を参照し、env 間でのリソース名の衝突を防ぐ

https://aws-amplify.github.io/docs/cli-toolchain/quickstart?sdk=js#custom-cloudformation-stacks

# Tips: ユースケースに応じて使い分けることが大事






In Partnership with intel

aws DEV DAY
ONLINE JAPAN

# Next Step

In Partnership with intel

# Q: Amplifyを本番環境で使ってる事例が知りたい

## A: Amplify Meetup にぜひご参加を！


Connpassグループ

https://aws-amplify-jp.connpass.com/

Amplify Meetup #01 登壇資料集
「Amplifyで実現した開発期間15日の路面電車乗務員勤務管理システムのフィールドテスト事例について」時田 明典 さん（KDDI株式会社）
「Amplify Consoleのビルド通知をSlackで受け取るためにやったこと」荻野 陽太 さん（株式会社JustInCaseTechnologies）
「痒いところに手が届くAmplify」吉田 祐樹（アマゾンウェブサービスジャパン株式会社）
「Amplify 山あり、谷あり奮闘記」山本 準 さん（ナイル株式会社）
「AmplifyのAPI機能を使う際のTips」Taewoo Kim（キムテウ） さん（クラスメソッド株式会社）
「わたしがAWS Amplifyを使い続ける理由。」資延 香里 さん（アビームコンサルティング株式会社）

https://aws.amazon.com/jp/blogs/news/amplify-meetup-01/



In Partnership with intel

# Q: Amplifyでの開発を手を使って学びたい


https://amplify-sns.workshop.aws/ja/

A: CTOになりきってTwitterライクなアプリを開発するAmplify SNS Workshopがおすすめです！

@model, @auth, @key, @function, @searchableといったディレクティブの活用方法

チーム開発でのAmplify Console、multi env機能の使い方



In Partnership with intel

aws DEV DAY
ONLINE JAPAN

# Closing

In Partnership with intel

# まとめ

AWS Amplifyを本番環境で使うためのノウハウ

- amplify env、Amplify Consoleを用いたWorkflow
- AWS上の環境統制

AWS Amplify を使用して高速に開発するためのTips

- @model, @key, @auth, @searchable, @auth
- Amplify Mocking

AWS Amplify 拡張シナリオ

- OAuthでSNS認証を組み込む
- カスタマーエンゲージメントツールを組み込む
- 分析基盤を組み込む
- AmplifyでサポートされていないAWSのサービスをAmplifyで管理する



In Partnership with intel

aws DEV DAY
ONLINE JAPAN

# Thank you!

Jaga (Daisuke Nagayama)
Twitter@jagaimogmog

In Partnership with intel