

aws **DEV DAY**  
ONLINE JAPAN

# DEV DAY

20-22.10.2020

In Partnership with **intel**

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

F - 7

# Amplify CLI Deep Dive

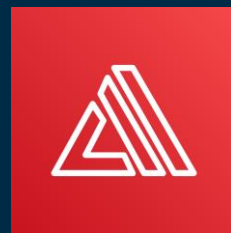
Jaga (Daisuke Nagayama) @jagaimogmog  
Amazon Web Services Japan  
Startup Solutions Architect

# Jaga (Daisuke Nagayama)

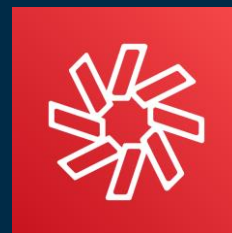
@jagaimogmog

Startup Solutions Architect  
Amazon Web Service Japan K.K.

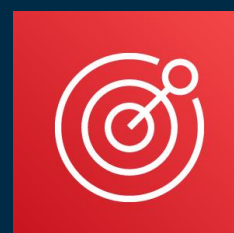
- > Health Care Startupでエンジニア・事業開発
- > AWSでStartupの技術支援・Mobile Specialist



AWS Amplify



Amazon Chime SDK



Amazon Pinpoint



## 想定視聴者

- AWS Amplify、特にAmplify CLIを利用したことがある方
- 特にAmplify CLIを使った開発に慣れてきたが、ちょっと手の届かないところがあると感じる方

## 目的

- Amplify CLIでわからないことが起きた時、どのファイルを調べれば良いか見当がつくようになる
- Amplify CLIのGit Hub Issueのディスカッション内容が理解できる

# Agenda

1. Amplify CLIのコアファイル
2. amplify/backendディレクトリの世界
3. カテゴリのファイル構造の例 - api(GraphQL) -
4. カテゴリのファイル構造の例 - function -
5. Closing



# 注意事項

検証環境は @aws-amplify/cli@4.29.6です

aws-exports.jsなどの出力ファイルはJavaScriptとなっていますが、内容自体はiOS/Androidなど別プラットフォームのバックエンドでも適用可能です

一部コマンド結果の文字が小さい部分は雰囲気感じ取っていただいで、ダウンロード資料の方で確認いただければと思います

Amplifyは開発が盛んなため、予告なくファイル構成が変更される可能性があります

Amplifyのコマンドについての説明は必要最小限とします



# 1. Amplify CLIのコアファイル

2. amplify/backendディレクトリの世界

3. カテゴリのファイル構造の例 - api(GraphQL) -

4. カテゴリのファイル構造の例 - function -

5. Closing

# amplify init直後のディレクトリ構造

```
$ tree -a
```

```
.
├── .gitignore
├── amplify
│   ├── #current-cloud-backend
│   │   └── amplify-meta.json
│   ├── .config
│   │   ├── local-aws-info.json
│   │   ├── local-env-info.json
│   │   └── project-config.json
│   ├── backend
│   │   ├── amplify-meta.json
│   │   ├── backend-config.json
│   │   └── tags.json
│   ├── cli.json
│   └── team-provider-info.json
└── src
    └── aws-exports.js
```

```
5 directories, 11 files
```

## amplify

- Amplify CLIによって作成される設定ファイルの置き場所
- 基本はこの中にあるファイルを編集していく

## src/exports.js

- Amplify CLIが自動で生成してくれる設定ファイル
- フロントエンドでライブラリの初期設定をするのに使う

## .gitignore

- Amplifyがgit管理下に置かないことを推奨している項目を追加
- ファイルがない場合は新規作成



# .gitignore

```
#amplify
amplify/\#current-cloud-backend
amplify/.config/local-*
amplify/mock-data
amplify/backend/amplify-meta.json
amplify/backend/awscloudformation
build/
dist/
node_modules/
aws-exports.js
awsconfiguration.json
amplifyconfiguration.json
amplify-build-config.json
amplify-gradle-config.json
amplifytools.xcconfig
```

- チームでの共有する必要がなかったり、Amplify ConsoleでCI/CDする際に必要のないファイルはgitignoreされる
- 例外として、team-provider-info.json（後述）はデフォルトで.gitignoreに含まれないが、プライベートリポジトリの場合以外は.gitignoreに追加することが推奨
- どのファイルを手動で変更して良いか、バージョンコントロールに含めるべきかは下記ドキュメントに記載

<https://docs.amplify.aws/cli/reference/files>



# amplifyディレクトリ

```
$ tree amplify -a
amplify
├── #current-cloud-backend
│   └── amplify-meta.json
├── .config
│   ├── local-aws-info.json
│   ├── local-env-info.json
│   └── project-config.json
├── backend
│   ├── amplify-meta.json
│   ├── backend-config.json
│   └── tags.json
├── cli.json
└── team-provider-info.json

3 directories, 9 files
```

## #current-cloud-backend

- AWS上のAWS CloudFormationスタックの状態の保存場所
- \$ amplify pushや\$ amplify statusコマンド発行時に更新

## .config

- ローカル開発環境の設定の保存場所

## backend

- CloudFormationテンプレートはじめとした、バックエンドの設定項目の置き場所

# team-provider-info.json

envごとに情報が記述

```
{
  "dev": {
    "awscloudformation": {
      "AuthRoleName": "amplify-devday2020-dev-115715-authRole",
      "UnauthRoleArn": "arn:aws:iam::XXXXXXXXXXXX:role/amplify-devday2020-dev-115715-unauthRole",
      "AuthRoleArn": "arn:aws:iam::XXXXXXXXXXXX:role/amplify-devday2020-dev-115715-authRole",
      "Region": "us-east-1",
      "DeploymentBucketName": "amplify-devday2020-dev-115715-deployment",
      "UnauthRoleName": "amplify-devday2020-dev-115715-unauthRole",
      "StackName": "amplify-devday2020-dev-115715",
      "StackId": "arn:aws:cloudformation:us-east-1:XXXXXXXXXXXX:stack/amplify-devday2020-dev-115715/50361cd0-0aa4-11eb-9427-0a2de5ab949d",
      "AmplifyAppId": "d3nnzzu02irqny"
    }
  }
}
```

設定ファイルの置き場所

CloudFormation Stackの情報

Amplify ConsoleのAppId

- 手元のAmplifyプロジェクトとクラウド上のリソースの対応を確認可能
- ARNに加えOIDC認証のID, Secretといった情報も書き込まれるため、パブリックリポジトリの場合gitignoreが推奨
- パブリックリポジトリ利用の場合、開発者間のenv情報受け渡しは \$ amplify pullコマンドを利用



# cli.json – Feature Flags機能

デフォルト

```
{  
  "features": {}  
}
```

使用例

```
{  
  "features": {  
    "graphqlTransformer": {  
      "transformerVersion": 5  
    },  
    "keyTransformer": {  
      "defaultQuery": true  
    },  
    "experimental": {  
      "feature1": true,  
      "feature2": false  
    },  
    "release": {  
      "lambdaLayers": true  
    }  
  }  
}
```

## 概要

- 4.26.0で追加 (現状利用可能なFlagはなし)
- Amplify CLIの機能の有効化・無効化、バージョンの指定などが可能

## 種類

- Release: アクティブに変更が入る機能
- Feature: 後方互換性のない変更の反映
- Experimental: 実験段階の機能

## ライフサイクル

- ドキュメント・リリースノートで廃止時期と実装の詳細を告知
- 実装後はFeature Flagも消える

<https://docs.amplify.aws/cli/reference/feature-flags>

# amplify/.config

amplify/.config/local-env-info.json

```
{
  "projectPath": "/Users/jaga/devday2020",
  "defaultEditor": "vim",
  "envName": "dev"
}
```

amplify/.config/local-aws-info.json

```
{
  "dev": {
    "configLevel": "project",
    "useProfile": true,
    "profileName": "amplify-profile"
  }
}
```

amplify/.config/project-config.json

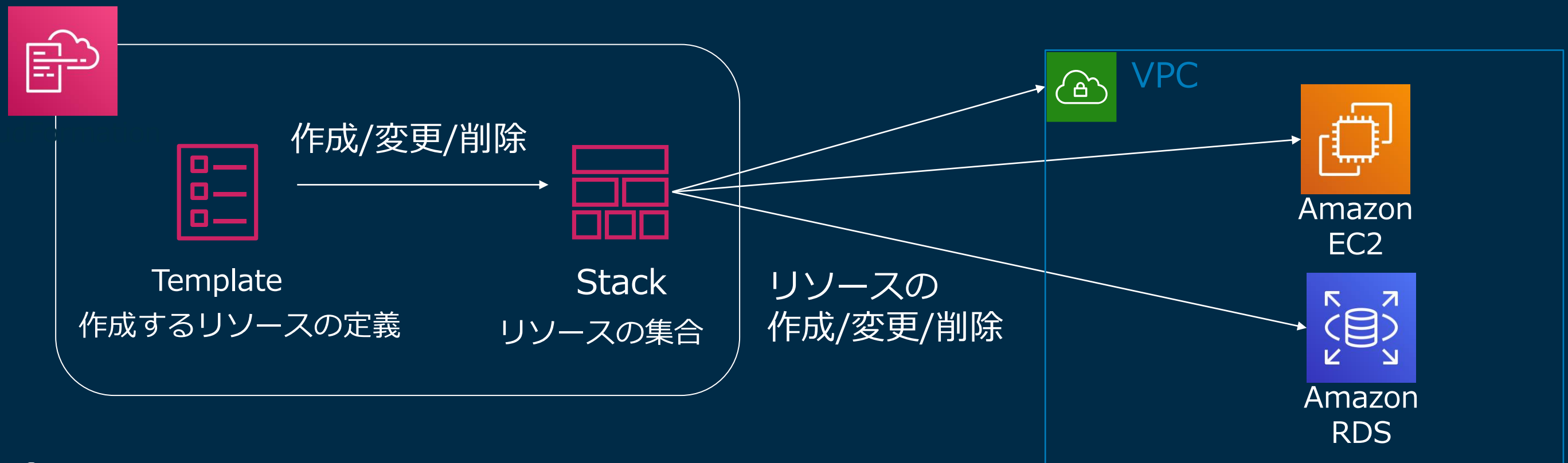
```
{
  "projectName": "devday2020",
  "version": "3.0",
  "frontend": "javascript",
  "javascript": {
    "framework": "none",
    "config": {
      "SourceDir": "src",
      "DistributionDir": "dist",
      "BuildCommand": "npm run-script build",
      "StartCommand": "npm run-script start"
    }
  },
  "providers": [
    "awscloudformation"
  ]
}
```

- envとprofileの対応はlocal-aws-info.jsonで確認
- 手動での変更は非推奨のため、\$ amplify configureコマンドを利用
- project-config.jsonのみgit管理推奨。残り2ファイルは開発者ごとの設定

# AWS CloudFormation

EC2 や ELB といった AWS リソースの環境構築を、設定ファイル (テンプレート) を元に自動化できるサービス

- テンプレートには起動すべきリソースの情報を JSON や YAML フォーマットのテキスト形式で記述
- 追加料金はなくプロビジョニングされたAWSリソース分の料金のみ発生)
- Amplify CLIはAWS CloudFormationをラップし、対話的にリソースの作成・変更・削除を行う



# テンプレートの要素(抜粋)

実行時 (スタック作成/更新) にユーザ入力を求めるパラメータ (KeyPairの名前や、DBのユーザ名など)

**(必須パラメータ)** AWS AppSyncやAmazon DynamoDB など、スタックを構成するリソースとプロパティ

スタック構築後にAWS CloudFormationから出力させる値 (AWS AppSyncのエンドポイントのURLなど)

```
AWSTemplateFormatVersion: "version date"  
Description:  
  String  
Parameters:  
  set of parameters  
Resources:  
  set of resources  
Outputs:  
  set of outputs
```

# AmplifyのCloudFormation Stack

CloudFormation > スタック > amplify-devday2020-dev-115715

amplify-devday2020-dev-115715 削除 更新する スタックアクション スタックの作成

スタックの情報 | イベント | **リソース** | 出力 | パラメータ | テンプレート | 変更セット

### リソース (10)

リソースの検索

論理 ID	物理 ID	タイプ	ステータス	状況の理由
AuthRole	<a href="#">amplify-devday2020-dev-115715-authRole</a>	AWS::IAM::Role	✔ UPDATE_COMPLETE	-
DeploymentBucket	<a href="#">amplify-devday2020-dev-115715-deployment</a>	AWS::S3::Bucket	✔ UPDATE_COMPLETE	-
UnauthRole	<a href="#">amplify-devday2020-dev-115715-unauthRole</a>	AWS::IAM::Role	✔ UPDATE_COMPLETE	-
UpdateRolesWithIDPFunction	<a href="#">amplify-devday2020-dev-11-UpdateRolesWithIDPFunci-1096JXECDUJK</a>	AWS::Lambda::Function	✔ CREATE_COMPLETE	-
UpdateRolesWithIDPFunctionOutputs	2020/10/10/[\$LATEST]3c5600597f13494c8c8cd5b3869198bc	Custom::LambdaCallout	✔ CREATE_COMPLETE	-
UpdateRolesWithIDPFunctionRole	<a href="#">amplify-devday2020-dev-115715-authRole-idp</a>	AWS::IAM::Role	✔ CREATE_COMPLETE	-
apidevday2020	<a href="#">arn:aws:cloudformation:us-east-1:620456053892:stack/amplify-devday2020-dev-115715-apidevday2020-U7RHJ51E7GLP/41b0c460-0ac9-11eb-8fc0-0aa9a390fc79</a>	AWS::CloudFormation::Stack	✔ UPDATE_COMPLETE	-

**スタック (8)**

amplify-devday2020-dev-115715

アクティブ  ネスト表示

- ネストされた amplify-devday2020-dev-115715-apidevday2020-U7RHJ51E7GLP-CustomResourcesjson-5NSJOSPY7YO8  
2020-10-10 16:23:57 UTC+0900  
✔ CREATE\_COMPLETE
- ネストされた amplify-devday2020-dev-115715-apidevday2020-U7RHJ51E7GLP-Todo-151EQYOH4Z51S  
2020-10-10 16:23:06 UTC+0900  
✔ UPDATE\_COMPLETE
- ネストされた amplify-devday2020-dev-115715-apidevday2020-U7RHJ51E7GLP  
2020-10-10 16:21:47 UTC+0900  
✔ UPDATE\_COMPLETE
- ネストされた amplify-devday2020-dev-115715-authdevday2020-0ac75fde8-R7R4O0GIZXT  
2020-10-10 16:20:08 UTC+0900  
✔ CREATE\_COMPLETE
- amplify-devday2020-dev-115715  
2020-10-10 11:57:20 UTC+0900  
✔ UPDATE\_COMPLETE





1. Amplify CLIのコアファイル

## 2. amplify/backendディレクトリの世界

3. カテゴリのファイル構造の例 - api(GraphQL) -

4. カテゴリのファイル構造の例 - function -

5. Closing

# \$ amplify add api後のamplify/backend

```
$ tree -a -L 3 amplify/backend
amplify/backend
├── amplify-meta.json
├── api
│   └── devday2020
│       ├── build
│       ├── parameters.json
│       ├── resolvers
│       ├── schema.graphql
│       ├── stacks
│       └── transform.conf.json
├── auth
│   └── devday2020ac75fde8
│       ├── devday2020ac75fde8-cloudformation-template.yml
│       └── parameters.json
├── backend-config.json
└── tags.json

13 directories, 15 files
```

- \$ amplify add apiで認証にauthカテゴリを利用するGraphQL APIを作成した様子
- backendディレクトリの直下にapiやauthといったカテゴリごとのディレクトリが生成
- functionのようにカテゴリが複数のリソースをもつこともあるため、各カテゴリの配下に、リソース(サービスと呼ばれる)が配置
- 各サービスの配下にはCloudFormationのテンプレートと、CloudFormationに渡すパラメータを記述したparameters.jsonが存在

# amplify/backend/backend-config.json

```
{
  "auth": "略",
  "api": {
    "devday2020": {
      "service": "AppSync",
      "providerPlugin": "awscloudformation",
      "output": {
        "authConfig": {
          "defaultAuthentication": {
            "authenticationType": "AMAZON_COGNITO_USER_POOLS",
            "userPoolConfig": {
              "userPoolId": "authdevday2020ac75fde8"
            }
          }
        },
        "additionalAuthenticationProviders": []
      }
    }
  },
  "function": {
    "devday2020466f68db": {
      "build": true,
      "providerPlugin": "awscloudformation",
      "service": "Lambda",
      "dependsOn": [
        {
          "category": "api",
          "resourceName": "devday2020",
          "attributes": [
            "GraphQLAPIIdOutput",
            "GraphQLAPIEndpointOutput"
          ]
        }
      ]
    }
  }
}
```

- \$ amplify add functionでGraphQL APIを参照するfunctionを追加した様子
- 各カテゴリのリソースをデプロイする上で必要なメタ情報が記述
- dependsOnパラメーターを使用することで、他カテゴリのサービスのOutput Attributeを参照可能
- Custom CloudFormation Stacksを使うときのみ編集し、git管理対象

# amplify/backend/amplify-meta.json

```
{
  "providers": {
    .....
  },
  "auth": {
    .....
  },
  "api": {
    "devday2020": {
      "service": "AppSync",
      "providerPlugin": "awscloudformation",
      "output": {
        "authConfig": {
          "defaultAuthentication": {
            "authenticationType": "AMAZON_COGNITO_USER_POOLS",
            "userPoolConfig": {
              "userPoolId": "authdevday2020ac75fde8"
            }
          },
          "additionalAuthenticationProviders": []
        },
        "GraphQLAPIIdOutput": "bwmpu2kmsbfi3biwt6kywnvnm",
        "GraphQLAPIEndpointOutput": "https://cwm64lfjujdl7notuhsbb3n2fe.appsync-api.us-east-1.amazonaws.com/graphql"
      },
      "providerMetadata": {
        "s3TemplateURL": "https://s3.amazonaws.com/amplify-devday2020-dev-115715-deployment/amplify-cfn-templates/api/cloudformation-template.json",
        "logicalId": "apidevday2020"
      },
      "lastPushTimeStamp": "2020-10-10T07:24:58.599Z",
      "lastPushDirHash": "H03/oyuqliijkPAGrvRWNbL+SEE="
    }
  }
}
```

- Git管理には含めず、手動変更も非推奨なCLIのコアやカテゴリ間で情報連携するためのファイル
- Outputには他のカテゴリから参照可能なパラメータを出力(GraphQL APIの場合エンドポイントなど)

# おまけ: functionカテゴリからの参照方法

```
"Parameters": {
  "CloudWatchRule": {
    "Type": "String",
    "Default": "NONE",
    "Description": "Schedule Expression"
  },
  "env": {
    "Type": "String"
  },
  "apidevday2020GraphQLAPIIdOutput": {
    "Type": "String",
    "Default": "apidevday2020GraphQLAPIIdOutput"
  },
  "apidevday2020GraphQLAPIEndpointOutput": {
    "Type": "String",
    "Default": "apidevday2020GraphQLAPIEndpointOutput"
  }
},
```

- CloudFormationでは実行時、Parameterを渡すことができ、Stack作成時に参照可能
- amplify push時、<category><resource-name><parameter-name>というkeyで、Amplify CLIがdependsOnで指定した情報を引き渡す
- \$ amplify update apiコマンドで他カテゴリのリソースを参照する設定を追加すると自動で足してくれるので、普段は意識しなくて良い

# おまけ: amplify-meta.jsonのBefore After

```
{
  "providers": {
    "awscloudformation": {
      "AuthRoleName": "amplify-devday2020-dev-115715-authRole",
      .....
    }
  },
  "auth": {
    "devday2020ac75fde8": {
      "service": "Cognito",
      "providerPlugin": "awscloudformation",
      "dependsOn": [],
      "customAuth": false
    }
  },
  "api": {
    "devday2020": {
      "service": "AppSync",
      "providerPlugin": "awscloudformation",
      "output": {
        "authConfig": {
          "defaultAuthentication": {
            "authenticationType": "AMAZON_COGNITO_USER_POOLS",
            "userPoolConfig": {
              "userPoolId": "authdevday2020ac75fde8"
            }
          },
          "additionalAuthenticationProviders": []
        }
      }
    }
  }
}
```

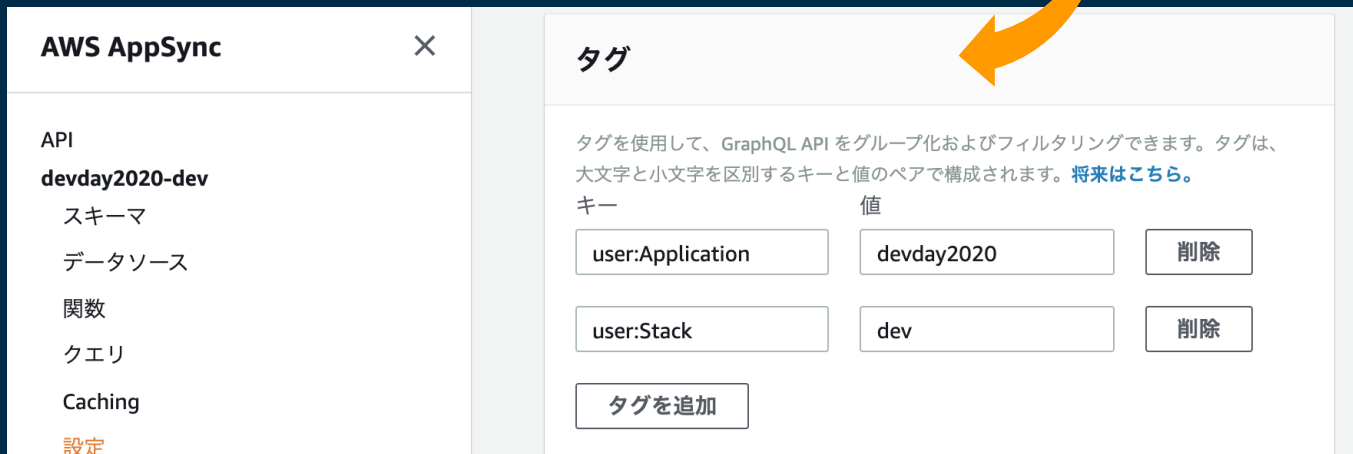
\$ amplify push

```
{
  "providers": {
    .....
  },
  "auth": {
    .....
  },
  "api": {
    "devday2020": {
      "service": "AppSync",
      "providerPlugin": "awscloudformation",
      "output": {
        "authConfig": {
          "defaultAuthentication": {
            "authenticationType": "AMAZON_COGNITO_USER_POOLS",
            "userPoolConfig": {
              "userPoolId": "authdevday2020ac75fde8"
            }
          },
          "additionalAuthenticationProviders": []
        },
        "GraphQLAPIIdOutput": "bwmpu2kmsbfi3biwt6kywnvnm",
        "GraphQLAPIEndpointOutput": "https://cwm64lfjujdl7notuhsbb3n2fe.appsync-api.us-east-1.amazonaws.com/graphql"
      },
      "providerMetadata": {
        "s3TemplateURL": "https://s3.amazonaws.com/amplify-devday2020-dev-115715-deployment/amplify-cfn-templates/api/cloudformation-template.json",
        "logicalId": "apidevday2020"
      },
      "lastPushTimeStamp": "2020-10-10T07:24:58.599Z",
      "lastPushDirHash": "H03/oyuqliijkPAGrvRWNbL+SEE="
    }
  }
}
```

# amplify/backend/tags.json – Resource Tagging

```
[
  {
    "Key": "user:Stack",
    "Value": "{project-env}"
  },
  {
    "Key": "user:Application",
    "Value": "{project-name}"
  }
]
```

\$ amplify push



AWS AppSync ×

タグ

タグを使用して、GraphQL API をグループ化およびフィルタリングできます。タグは、大文字と小文字を区別するキーと値のペアで構成されます。将来はこちら。

キー	値	
user:Application	devday2020	削除
user:Stack	dev	削除

タグを追加

## 概要

- 4.28.0で追加されたAWSリソースへのタグ付け機能
- 請求時の計算などに利用可能

## 使い方

- tags.jsonの配列に追加することで最大50までのタグを使用可能
- {project-env}と{project-name}変数はamplify push時に代入される

## 注意事項

- 4.28.0以前のバージョンで作成されたプロジェクトの場合、tags.jsonファイルを作成する必要がある

1. Amplify CLIのコアファイル

2. amplify/backendディレクトリの世界

**3. カテゴリのファイル構造の例 - api(GraphQL) -**

4. カテゴリのファイル構造の例 - function -

5. Closing



# api(GraphQL)のディレクトリ構成

```
$ tree -a amplify/backend/api
amplify/backend/api
├── devday2020
│   ├── build
│   │   ├── cloudformation-template.json
│   │   ├── functions
│   │   ├── parameters.json
│   │   ├── pipelineFunctions
│   │   ├── resolvers
│   │   │   ├── Mutation.createTodo.req.vtl
│   │   │   ├── Mutation.createTodo.res.vtl
│   │   │   ├── Mutation.deleteTodo.req.vtl
│   │   │   ├── Mutation.deleteTodo.res.vtl
│   │   │   ├── Mutation.updateTodo.req.vtl
│   │   │   ├── Mutation.updateTodo.res.vtl
│   │   │   ├── Query.getTodo.req.vtl
│   │   │   ├── Query.getTodo.res.vtl
│   │   │   ├── Query.listTodos.req.vtl
│   │   │   └── Query.listTodos.res.vtl
│   │   ├── schema.graphql
│   │   └── stacks
│   │       ├── CustomResources.json
│   │       └── Todo.json
│   ├── parameters.json
│   ├── resolvers
│   ├── schema.graphql
│   ├── stacks
│   │   ├── CustomResources.json
│   │   └── transform.conf.json
└── 8 directories, 19 files
```

## schema.graphql (リソース名ディレクトリ直下)

- AmplifyでGraphQL APIを管理する際に一番触るファイル
- GraphQL APIのロジックはこのファイルに集約される

## resolvers

- カスタムリゾルバーを作る際にマッピングテンプレート(VTLファイル)を置く場所

## build

- gitignoreされており、\$ amplify mock apiや\$ amplify push時にAmplify CLIが自動的に更新
- Service直下のschema.graphqlを元に、AppSyncの使うschema.graphqlや、@modelを使った場合にCRUD用のresolverを生成

# api(GraphQL)のparameters.json

```
{
  "AppSyncApiName": "devday2020",
  "DynamoDBBillingMode": "PAY_PER_REQUEST",
  "DynamoDBEnableServerSideEncryption": false,
  "AuthCognitoUserPoolId": {
    "Fn::GetAtt": [
      "authdevday2020ac75fde8",
      "Outputs.UserPoolId"
    ]
  },
  "ElasticSearchInstanceType": "t2.small.elasticsearch",
  "ElasticSearchInstanceCount": 1
}
```

## DynamoDBBillingMode

- 定額課金で一定スループットを確保する“PROVISIONED”(プロビジョニングモード)か、従量課金の“PAY\_PER\_REQUEST”(オンデマンドモード)が選択可能
- デフォルトは“PAY\_PER\_REQUEST”

## ElasticSearchInstanceType

- @searchableによって作成した全文検索用のAmazon Elasticsearch Serviceのインスタンスタイプ

<https://docs.amplify.aws/cli/graphql-transformer/config-params>

# Amplify Datastore

- AWS re:Invent 2019 で発表されたマルチプラットフォーム (iOS/Android/Web) な **デバイス側ストレージエンジン**
- アプリ実装者が端末の**オンライン/オフラインを意識せず**に実装が行える
- オンライン復旧後のデータ更新時における競合検知 & 自動マージ
- Amplify Datastore では GraphQL のクエリを記述することなく、各開発言語に最適化されたインターフェースで **AWS AppSync** にアクセスが可能



データの競合検知、  
自動解決



馴染みのあるローカル  
ファーストプログラミングモデル



デルタ同期と自動  
マージ



AWS AppSync、  
GraphQL との連携

<https://docs.amplify.aws/lib/datastore/getting-started/q/platform/js>



# transform.conf.json

## DataStore無効API

```
$ amplify update api
? Please select from one of the below mentioned services: GraphQL
? Select from the options below Enable DataStore for entire API
```

```
{
  "Version": 5,
  "ElasticsearchWarning": true
}
```

## DataStore有効API

```
{
  "Version": 5,
  "ElasticsearchWarning": true,
  "ResolverConfig": {
    "project": {
      "ConflictHandler": "AUTOMERGE",
      "ConflictDetection": "VERSION"
    }
  }
}
```

```
$ amplify update api
? Please select from one of the below mentioned services: GraphQL
? Select from the options below Disable DataStore for entire API
```

# Tips: amplify/mock-data

```
$ tree -a amplify/mock-data
amplify/mock-data
├── dynamodb
│   └── fake_us-fake-1.db
└──
1 directory, 1 file
```

- amplify mockにより生成されるデータの置き場所
- 例えば\$ amplify mock apiで立ち上がるDynamoDB Localのデータの実体はmock-data/dynamodb/fake\_us-fake-1.db
- @keyの変更により、最新のスキーマと整合性の取れないデータがDB内にあると、エラーが起こる
- その場合はmockディレクトリのfake\_us-fake-1.db ファイルを消すことで、リセット可能

1. Amplify CLIのコアファイル

2. amplify/backendディレクトリの世界

3. カテゴリのファイル構造の例 - api(GraphQL) -

**4. カテゴリのファイル構造の例 - function -**

5. Closing

# amplify/backend/function

```
amplify/backend/function
├── devday2020466f68db
│   ├── amplify.state
│   ├── devday2020466f68db-cloudformation-template.json
│   ├── dist
│   │   └── latest-build.zip
│   ├── function-parameters.json
│   ├── parameters.json
│   └── src
│       ├── event.json
│       ├── index.js
│       ├── package-lock.json
│       └── package.json
```

3 directories, 9 files

## amplify.state

- Amplify CLIが内部で使用するメタデータ。Git管理推奨で手動管理は非推奨

## dist

- Lambdaへのデプロイ用にsrc内のファイル群を圧縮したzipファイルの置き場

## parameters.json

- CloudFormationに引き渡すパラメータ

## src

- Lambdaで実行するソースファイルの置き場所

<https://docs.amplify.aws/cli/graphql-transformer/config-params>



# Tips: parameters.jsonで環境変数を渡す(1)

```
{  
  "MyRESTEndpoint": "example.com/api",  
  "MyGraphQLEndpoint": "example.com/graphql"  
}
```

## envで共通のカスタムパラメータ

- parameters.jsonに書き足すことで、CloudFormationで参照可能
- envごとに固有のパラメータを使いたい場合はteam-provider-info.jsonを利用

Recurring function: <https://docs.amplify.aws/cli/function#schedule-recurring-lambda-functions>





# Tips: parameters.jsonで環境変数を渡す(2)

amplify/backend/function/<service-name>/<service-name>-cloudformation.yml

```
"Parameters": {
  "CloudWatchRule": {
    "Type": "String",
    "Default": "NONE",
    "Description": " Schedule Expression"
  },
  "env": {
    "Type": "String"
  },
  "apidevday2020GraphQLAPIIdOutput": {
    "Type": "String",
    "Default": "apidevday2020GraphQLAPIIdOutput"
  },
  "apidevday2020GraphQLAPIEndpointOutput": {
    "Type": "String",
    "Default": "apidevday2020GraphQLAPIEndpointOutput"
  },
  "functiondevday202002fad996Arn": {
    "Type": "String",
    "Default": "functiondevday202002fad996Arn"
  },
  "MyRESTEndpoint": {
    "Type": "String",
    "Default": "MyRESTEndpoint"
  },
  "MyGraphQLEndpoint": {
    "Type": "String",
    "Default": "MyGraphQLEndpoint"
  }
},
```

```
"LambdaFunction": {
  "Type": "AWS::Lambda::Function",
  "Metadata": {
    .....
  },
  "Properties": {
    "Handler": "index.handler",
    "FunctionName": {
      .....
    },
    "Environment": {
      "Variables": {
        "ENV": {
          "Ref": "env"
        },
        "REGION": {
          "Ref": "AWS::Region"
        },
        "API_DEVDAY2020_GRAPHQLAPIIDOUTPUT": {
          "Ref": "apidevday2020GraphQLAPIIdOutput"
        },
        "API_DEVDAY2020_GRAPHQLAPIENDPOINTOUTPUT": {
          "Ref": "apidevday2020GraphQLAPIEndpointOutput"
        },
        "MyRESTEndpoint": {
          "Ref": "MyRESTEndpoint"
        },
        "MY_GRAPHQL_ENDPOINT": {
          "Ref": "MyGraphQLEndpoint"
        }
      }
    }
  }
},
```

# Tips: parameters.jsonで環境変数を渡す(3)

## 環境変数 (6) 編集

以下の環境変数はデフォルトの Lambda サービスキーを使用して保存時に暗号化されました。

キー	値
API_DEVDAY2020_GRAPHQLAPIENDPOINTOUTPUT	https://cwm64lfjuidl7notuhsbb3n2fe.appsync-api.us-east-1.amazonaws.com/graphql
API_DEVDAY2020_GRAPHQLAPIIDOUTPUT	bwmpu2kmsbfi3biwt6kywnvnm
ENV	dev
MY_GRAPHQL_ENDPOINT	example.com/graphql
MyRESTEndpoint	example.com/api
REGION	us-east-1

# Tips: env固有のパラメータの指定 (1)

```
{
  "dev": {
    "awscloudformation": {
      ...
    },
    "categories": {
      "function": {
        "devday2020466f68db": {
          "MyRESTEndpoint": "dev.api.example.com"
        }
      }
    }
  },
  "prod": {
    "awscloudformation": {
      ...
    },
    "categories": {
      "function": {
        "devday2020466f68db": {
          "MyRESTEndpoint": "api.example.com"
        }
      }
    }
  }
}
```

## 使い方

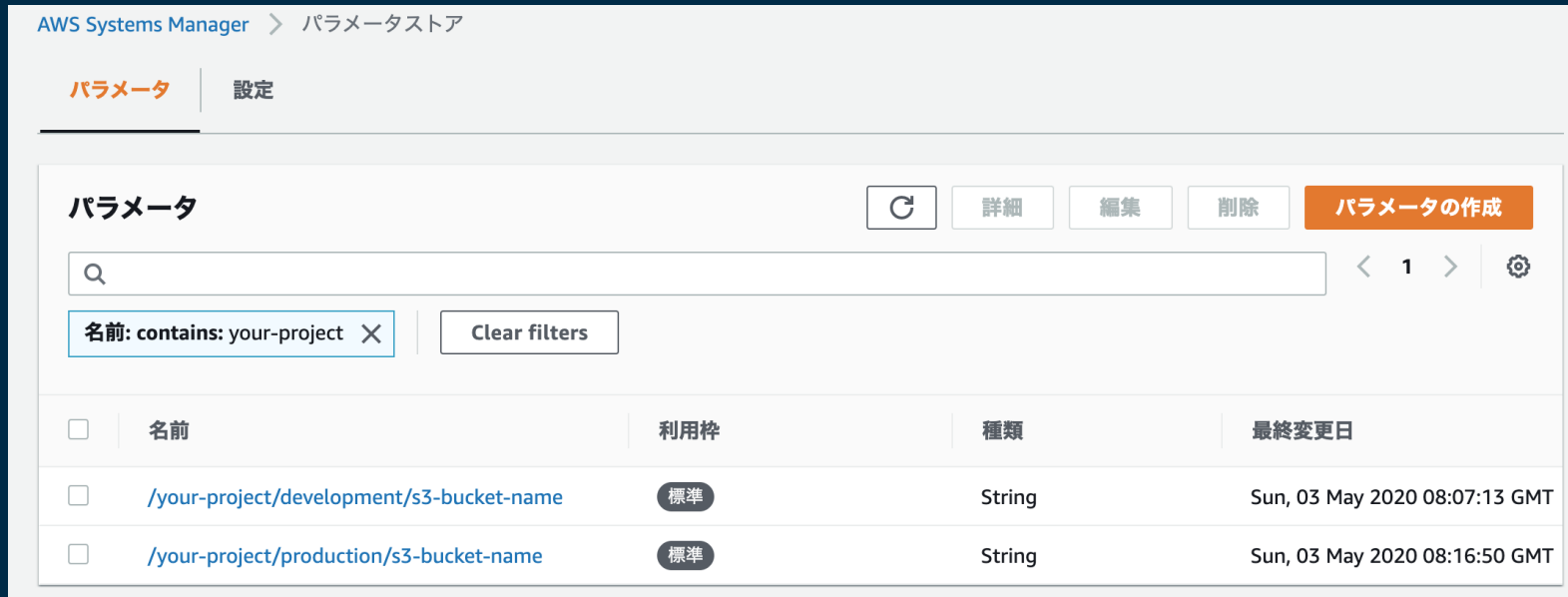
- team-provider-info.jsonでenv固有のパラメータを追記可能

## 注意事項

- team-provider-info.jsonとparameters.jsonに同じキー名のパラメータが指定された場合、team-provider-info.jsonの内容が優先される
- \$ amplify pullや\$ amplify env checkout時、amplify consoleでのデプロイでは手動で書き加えたパラメータは反映されないため、team-provider-info.jsonをgit管理する必要がある

<https://github.com/aws-amplify/amplify-cli/issues/3240#issuecomment-577832717>

# Tips: env固有のパラメータの指定 (2)



AWS Systems Manager > パラメータストア

パラメータ | 設定

パラメータ 🔄 詳細 編集 削除 パラメータの作成

🔍

名前: contains: your-project × Clear filters

<input type="checkbox"/>	名前	利用枠	種類	最終変更日
<input type="checkbox"/>	/your-project/development/s3-bucket-name	標準	String	Sun, 03 May 2020 08:07:13 GMT
<input type="checkbox"/>	/your-project/production/s3-bucket-name	標準	String	Sun, 03 May 2020 08:16:50 GMT

## 使い方

- AWS Systems Managerのパラメータストアでenvの名前を含んだkeyのパラメータを作成
- CloudFormation テンプレートで `resolve:ssm`関数を使って参照

```
Parameters:
  env:
    Default: production
    Type: String

Resources:
  S3Bucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: !Sub '{{resolve:ssm:/your-project/${env}/s3-bucket-name:2}}
```

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/dynamic-references.html>



# Tips: env固有のパラメータの指定 (3)



<https://github.com/aws-amplify/amplify-cli/issues/3240#issuecomment-577832717>



1. Amplify CLIのコアファイル
2. amplify/backendディレクトリの世界
3. カテゴリのファイル構造の例 - api(GraphQL) -
4. カテゴリのファイル構造の例 - function -

# 5. Closing

# まとめ

- やりたいことがドキュメントに書いてない場合はGit Hub Issueで検索！
- 本セッションの内容があれば、ディスカッションの内容が理解しやすい（はず）
- Amplify CLIをハックして快適なAmplifyライフを！

# 関連セッション

## A-3 : フロントエンドエンジニアのための AWS アーキテクチャ

講師：水馬 拓也（アマゾン ウェブ サービス ジャパン株式会社）

セッション概要：本セッションでは主にフロントエンドエンジニアに向けて Single Page Application (SPA) や Server Side Rendering (SSR)、JAMstack などのトピックを中心に AWS をどのようにご活用いただけるかについて紹介します。

## B-1 : AWS Amplify 実践編

講師：Jaga (Daisuke Nagayama)（アマゾン ウェブ サービス ジャパン株式会社）

セッション概要：AWS Amplify 実践編と称し、より実践的な Amplify を用いた開発についてお話しします。少人数で MVP をスピーディに開発するのに留まらず、事業が成長してチームが大きくなってきたときにどう Amplify を使えばよいか？ビジネスの敏捷性をエンジニアリングで支えるために、Amplify はどのような効率的なツール群を提供するか？広範なビジネスニーズに答えるための柔軟性を、Amplify と AWS の 175以上のサービス群で実現するには？こうした質問に答える形で、Amplify の魅力をお伝えします。





# Thank you!

Jaga (Daisuke Nagayama)

Twitter@jagaimogmog



Please complete the session  
survey in the mobile app.