




Day 1 Session Track 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

Takuya Mizuma
Prototyping Specialist Solution Architect
Amazon Web Service Japan

In Partnership with 

水馬 拓也 (みずま たくや)

Prototyping Specialist Solution Architect
Amazon Web Service Japan



- ERP パッケージベンダー にて会計系システムの開発
- 日系システムコンサル会社 にて金融システムの保守、クラウド推進
- Start up で AWS を中心に自社サービスの開発、保守

好きな技術領域：フロントエンド (特に Typescript や React)

好きな AWS サービス：AWS Amplify



本セッションについて

想定聴講者：

- フロントエンドエンジニアの方
- **Single Page Application (SPA)** や **Server Side Rendering (SSR)** **JAMstack** といったモダンなフロントエンドアプリケーションを AWS 上で構築されたい方

ゴール：

- フロントエンドに特に関係する AWS サービスを知る
- AWS でモダンなフロントエンドアプリケーションを運用するためのプラクティスを知る



Agenda

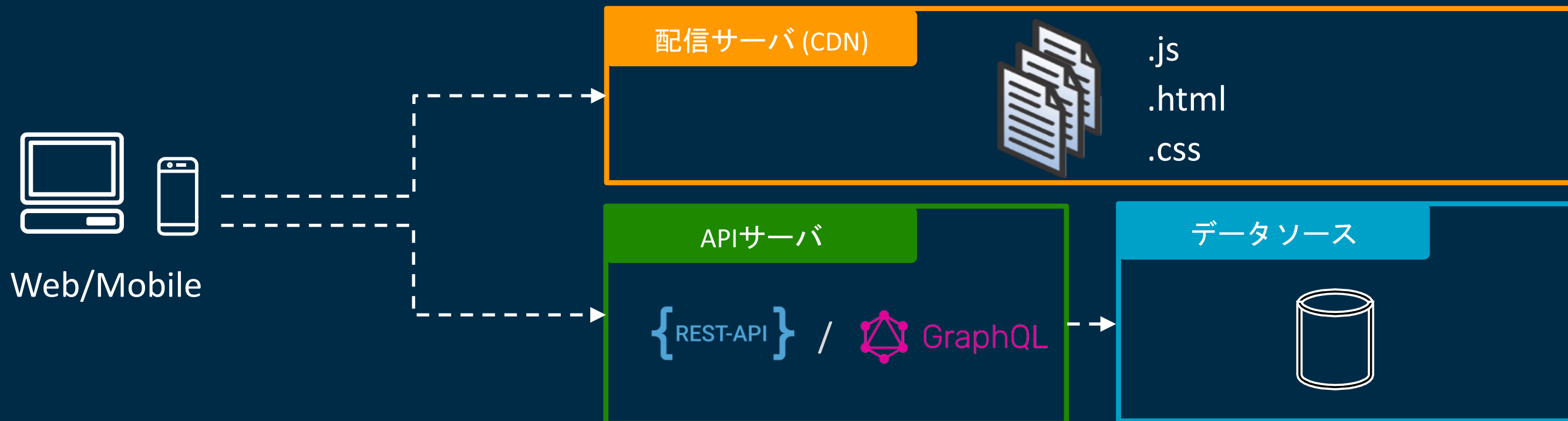
1. 代表的なフロントエンドの構成のおさらい
2. フロントエンドに関する AWS サービス
3. AWS Amplify のご紹介
4. 具体的なフロントエンドアーキテクチャの構築方法



代表的なフロントエンドの構成のおさらい

フロントエンドのアーキテクチャ

Single Page Application (SPA) の構成

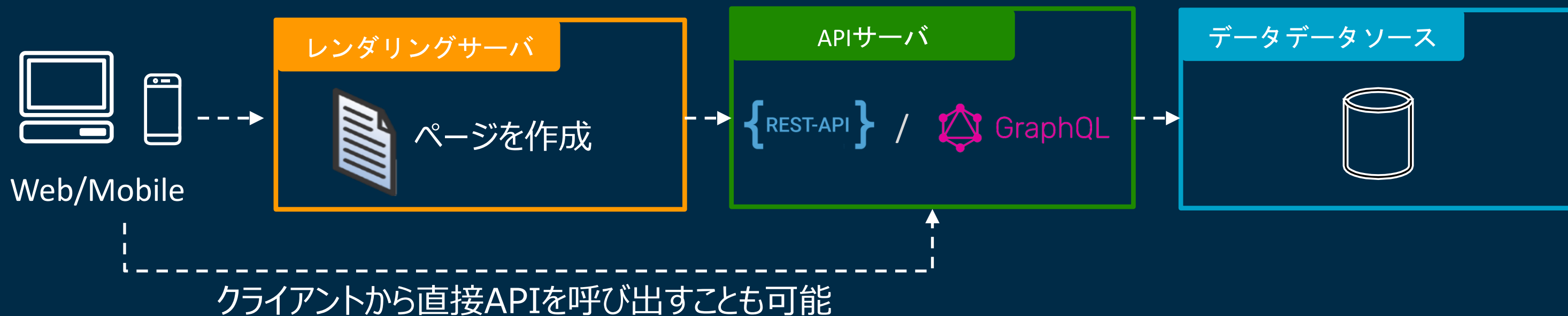


[SPAとの特徴]

- ✓ 初回アクセス時に全てのアセットを読み込んで単一のページを JavaScript で切り替えて表示させるので画面遷移が高速。ダイナミックな描画が可能。
- ✓ 初回アクセス時の表示が遅く、SEOの観点で不利になる場合がある。

フロントエンドのアーキテクチャ

Server Side Rendering (SSR) の構成

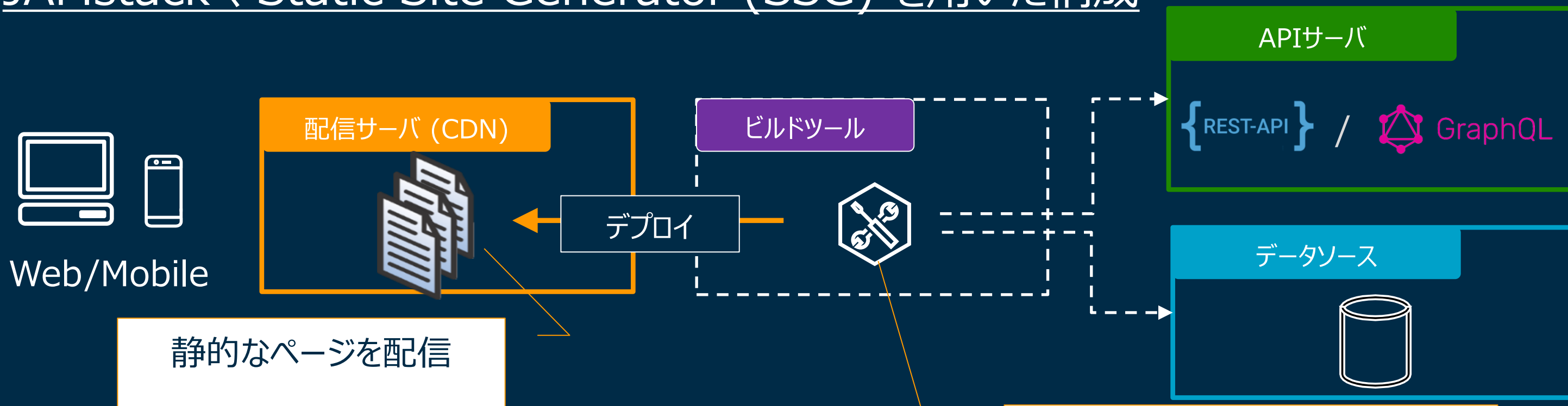


[SPAとの比較]

- ✓ サーバ側でページの構築を行うため、SPA と比較して初回表示が高速
- ✓ 初回表示の段階で構築済みのページが返却されるため SEO の観点で有利
- ✓ API サーバとは別にレンダリング用のサーバが必要

フロントエンドのアーキテクチャ

JAMstack、Static Site Generator (SSG) を用いた構成



[SPA、SSR との比較]

- ✓ 静的なHTMLを配信するため高速、SEO の観点で有利
- ✓ ページのデータが更新されるのはビルド時のみ
- ✓ ページ数が多くなるとビルドに時間がかかる

フロントエンドに関するAWSサービス

フロントエンドに関連するサービス

認証機能



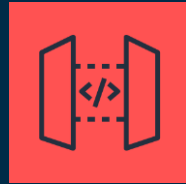
Amazon Cognito

データストア



Amazon DynamoDB

API 基盤



Amazon
API Gateway



AWS AppSync

ストレージ



Amazon Simple
Storage Service

コンピューティング



AWS Lambda

エッジサービス



Amazon
CloudFront


CI/CD パイプライン



Amplify Console



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

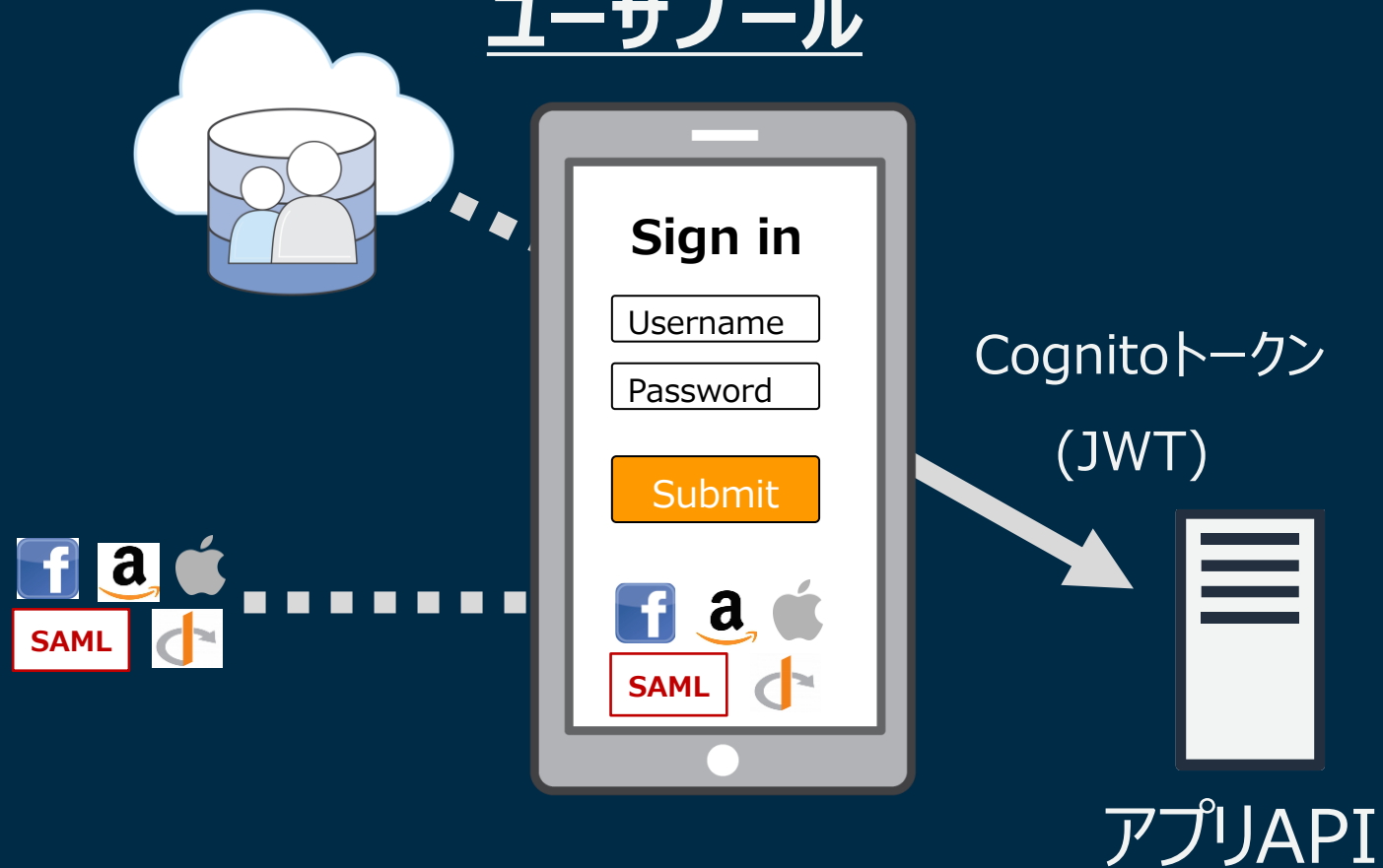
In Partnership with intel.

Amazon Cognito

認証機能

APIベースで実装されるモバイル、Webアプリにユーザ認証機能を提供する

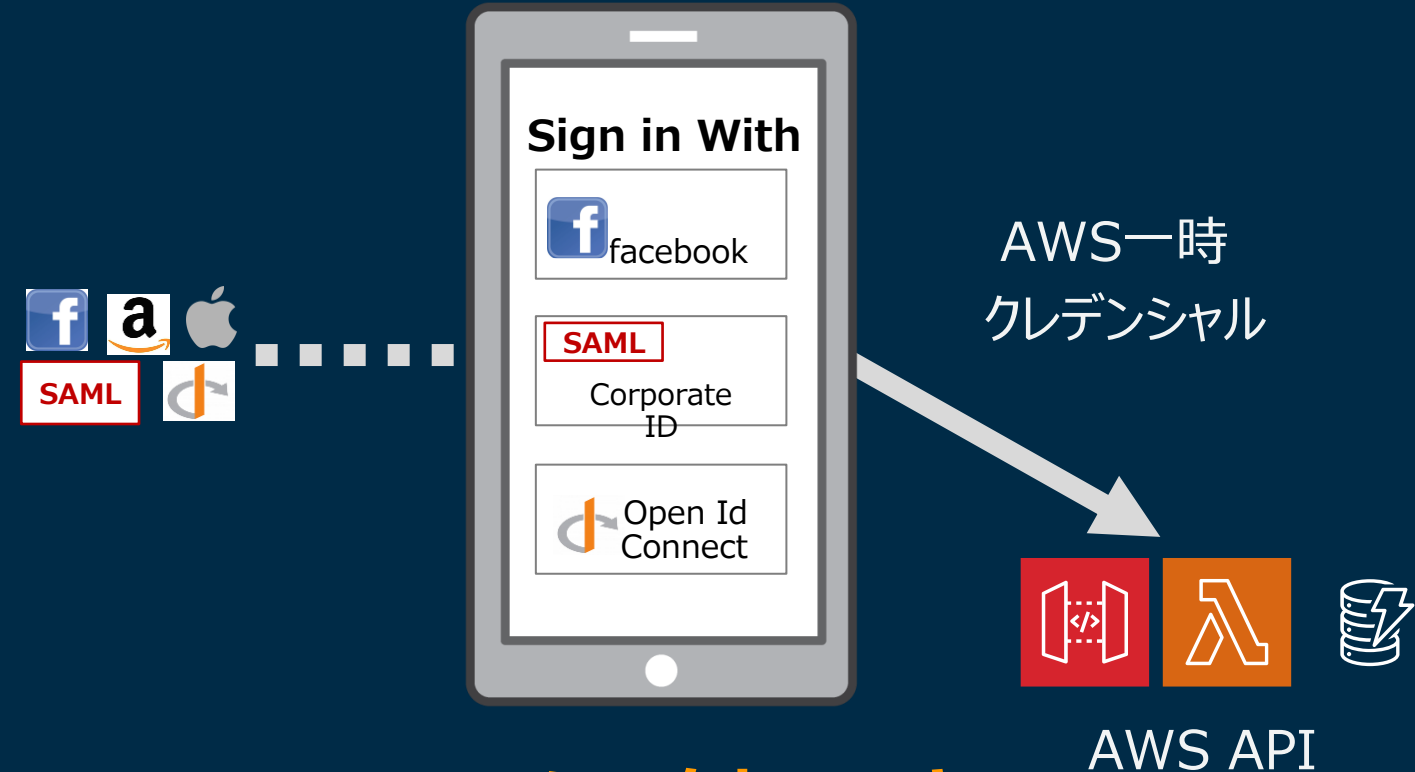
ユーザプール



アプリへのアクセスに利用できる
トークンを提供

ID プール

(フェデレーテッド アイデンティティ)



AWSにアクセスできる
クレデンシャルを提供



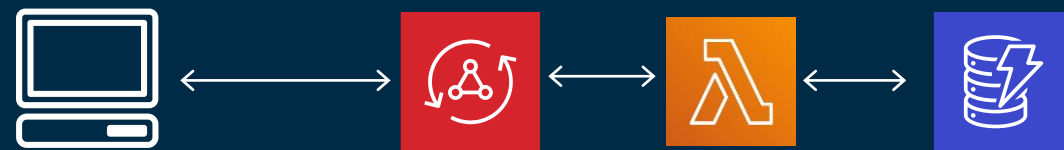
AWS Lambda

コンピューティング

イベント発生時にプログラムコードを実行するサーバレスアーキテクチャ基盤。サーバのプロビジョニングや管理なしでコードを実行可能。Python、Node.js、Java、C#、Go、PowerShell、Rubyに対応。カスタムランタイムの機能を用いることで 上記以外の言語にも対応可能。

よくあるデザインパターン

Web、モバイルバックエンド



- API のビジネスロジックの実装

画像処理・データ加工



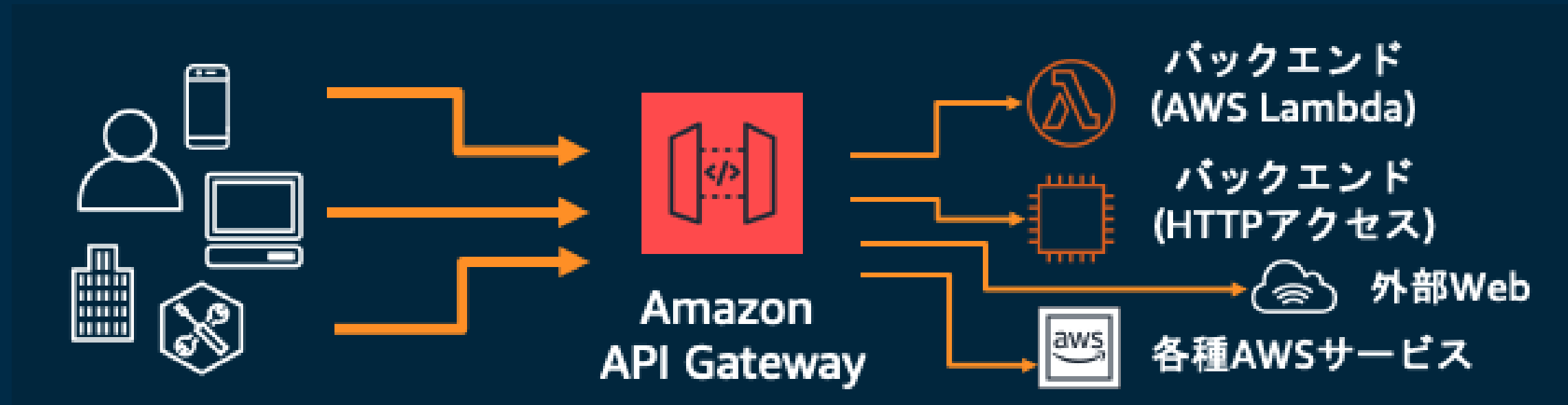
- 加工処理
- 画像圧縮、リサイズ...
 - シンプルな数値計算
 - 文字変換...

- データ投入をきっかけにファイル情報を引き渡して処理を起動

Amazon API Gateway

API 基盤

クライアントからのリクエストを受け付けるエンドポイントを提供し、各種バックエンドとの統合



フロントエンドのユースケース

The diagram shows a front-end use case where a laptop sends a `GET: /user` request to the Amazon API Gateway, which then routes the request to an AWS Lambda function.

- ✓ API Gateway を REST API のエンドポイントとして公開
- ✓ REST API のリソースとメソッドに統合された Lambda関数を実行

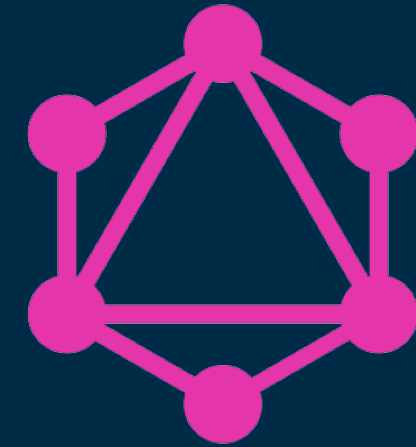
AWS AppSync (1)

API 基盤

Amazon DynamoDBをはじめとする様々な AWSのデータリソースにアクセスするためのマネージドなGraphQL 基盤を提供

GraphQL とは？

- ✓ GraphQL API 用 OSS の Query 言語
- ✓ 処理形態は、取得 (**Query**)、変更 (**Mutation**)、購読 (**Subscription**)



AWS AppSync

GraphQL の特徴

- ✓ 型指定されたスキーマ
- ✓ クライアントからレスポンスの形式を指定



AWS AppSync (2)

API 基盤

データソースにアクセスするためのスキーマが定義される

```
type User {  
  id: ID!  
  name: String  
  email: String  
}
```

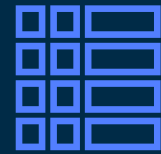
スキーマとデータソースの紐付けを行う resolver



/graphql



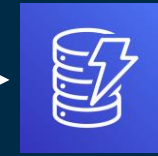
resolver



AWS AppSync



AWS Lambda



Amazon DynamoDB



Amazon Aurora Serverless



Amazon Elasticsearch Service



HTTP Endpoint

各種データソース

```
query GetUser($id: ID!) {  
  Execute Query  
  getUser(id: $id) {  
    id  
    name  
    email  
  }  
}
```

一つのエンドポイント(/graphql) にクエリを発行する



Amazon DynamoDB

データストア

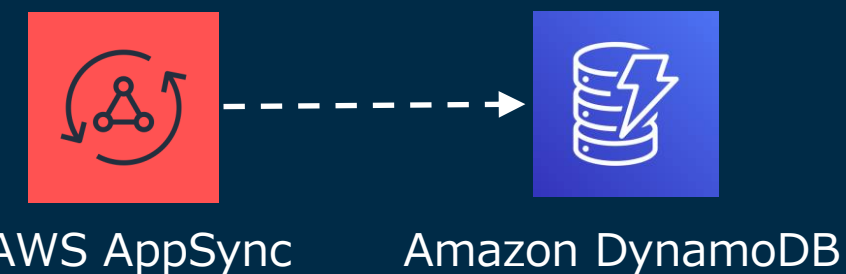
フルマネージドの NoSQL データベース。高速で予測可能なパフォーマンスとシームレスなスケールビリティを提供する。

- ✓ 無制限のストレージ容量
- ✓ スケールに応じたパフォーマンスの拡張が可能
- ✓ データは自動的に3カ所のAZに分散され、高い信頼性を担保

フロントエンドのユースケース



✓ REST API のデータ保存先として



✓ AppSync のデータソースとして

Amazon CloudFront

エッジサービス

低レイテンシーの高速転送によりデータ、ビデオ、アプリケーション、APIを安全に配信するグローバル配信ネットワーク(CDN)

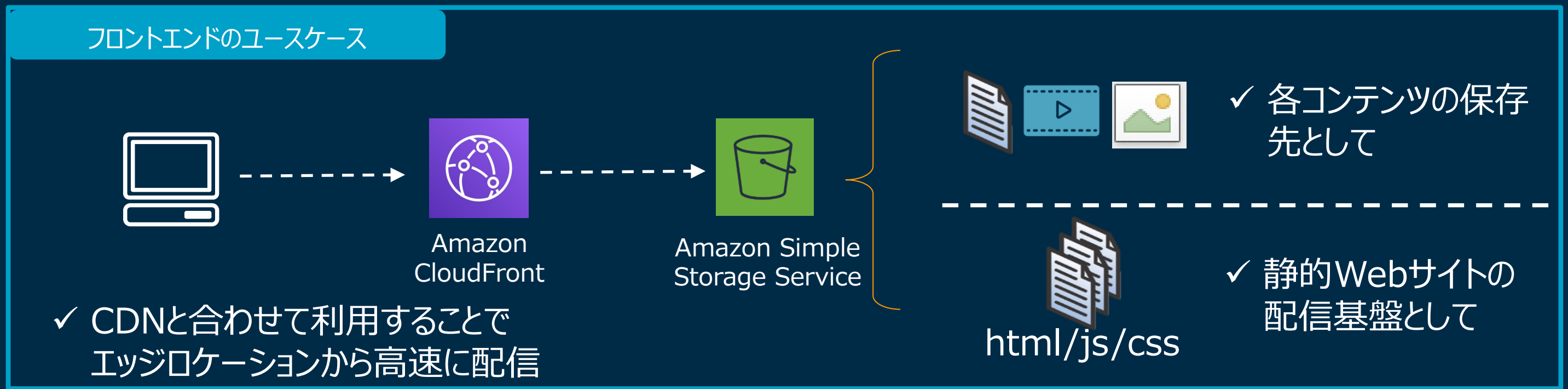
- ✓ 高性能な分散配信 (世界 216 拠点) ※ 2020年 6月時点
- ✓ 高いキャッシュヒット率
- ✓ ビルドインのセキュリティ機能(DDoS 対策や WAF 連携)
- ✓ 完全従量課金制



Amazon Simple Storage Service (Amazon S3)

ストレージ

99.9999999999% の耐久性を持つオブジェクトストレージ。バックアップやファイルの保存に最適。
静的機能の配信機能があり、画像やデジタルコンテンツ、静的 Web サイトの配信利用も。

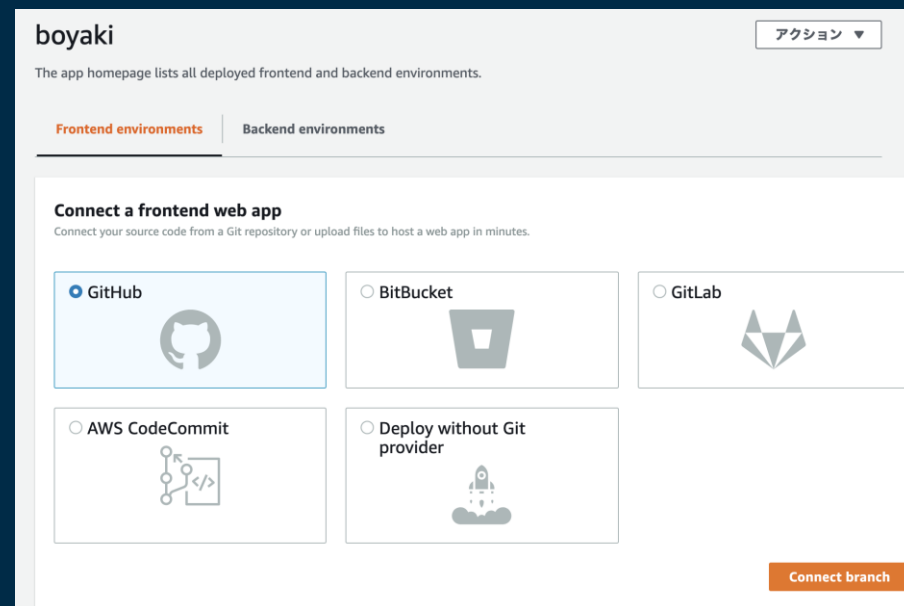


Amplify Console

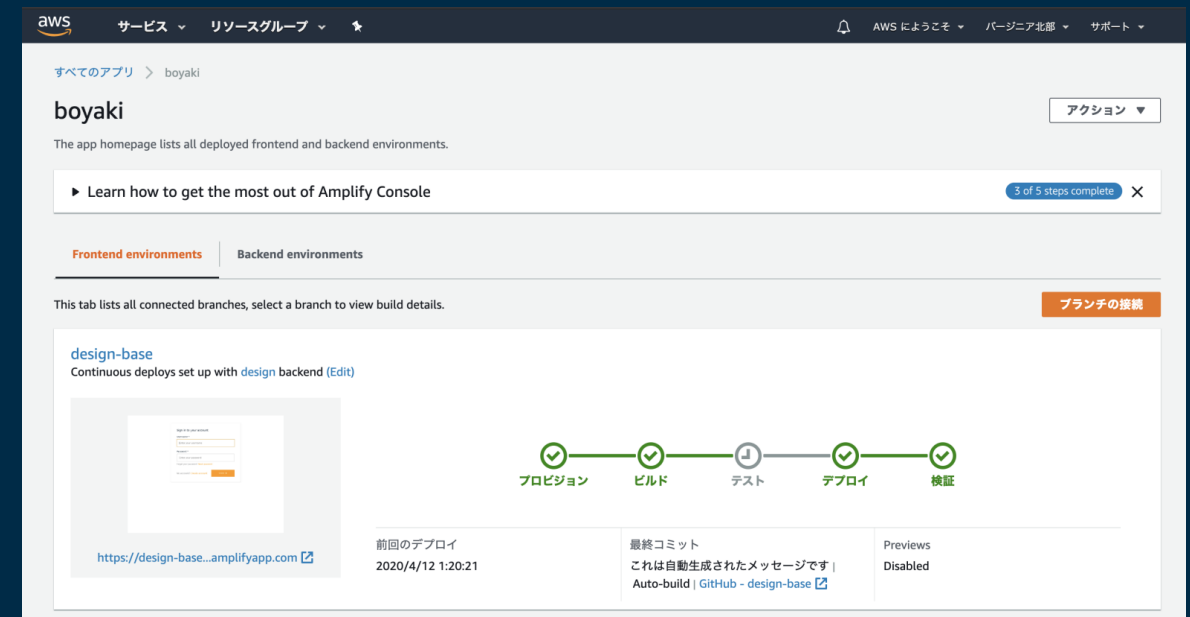
CI/CD パイプライン

SPA などの静的サイトの開発ワークロードを提供するフルスタックなホスティングサービス

- ✓ Git リポジトリと連携するだけで CI/CD パイプラインが構築可能
- ✓ CloudFront、S3 を意識することなくホスティング



ソースリポジトリの選択



CI/CD パイプラインの生成



フロントエンドに関連するサービス

認証機能



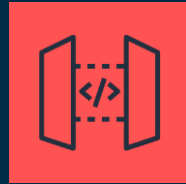
Amazon Cognito

データストア



Amazon DynamoDB

API 基盤



Amazon
API Gateway



AWS AppSync

ストレージ



Amazon Simple
Storage Service

コンピューティング



AWS Lambda

エッジサービス



Amazon
CloudFront


CI/CD パイプライン



Amplify Console



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

In Partnership with 

フロントエンドに関連するサービス

認証機能



Amazon Cognito

データストア



Amazon DynamoDB

API 基盤



Amazon API Gateway



AWS AppSync

ストレージ



Amazon Simple Storage Service

コンピューティング



AWS Lambda

エッジサービス



Amazon CloudFront

CI/CD パイプライン



Amplify Console

これらの仕組みをゼロから構築するのは
多くの労力が必要になる...

フロントエンドに関連するサービス

認証機能



Amazon Cognito

データストア



Amazon DynamoDB

API 基盤



Amazon API Gateway



AWS AppSync

ストレージ



Amazon Simple Storage Service

コンピューティング



AWS Lambda

バックエンドを強く意識することなく、アプリケーション開発に集中するための仕組みが必要



Amazon CloudFront

CI/CD パイプライン



Amplify Console






AWS Amplify という選択肢



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

In Partnership with  intel.

AWS Amplify のご紹介

AWS Amplify とは？

バックエンドを強く意識することなく Web アプリケーションの 実装、インフラの構築、アプリケーションのデプロイが可能なプラットフォーム



Amplify CLI

- ✓ バックエンドを強く意識することなくAWS のリソースを構築

Amplify Library

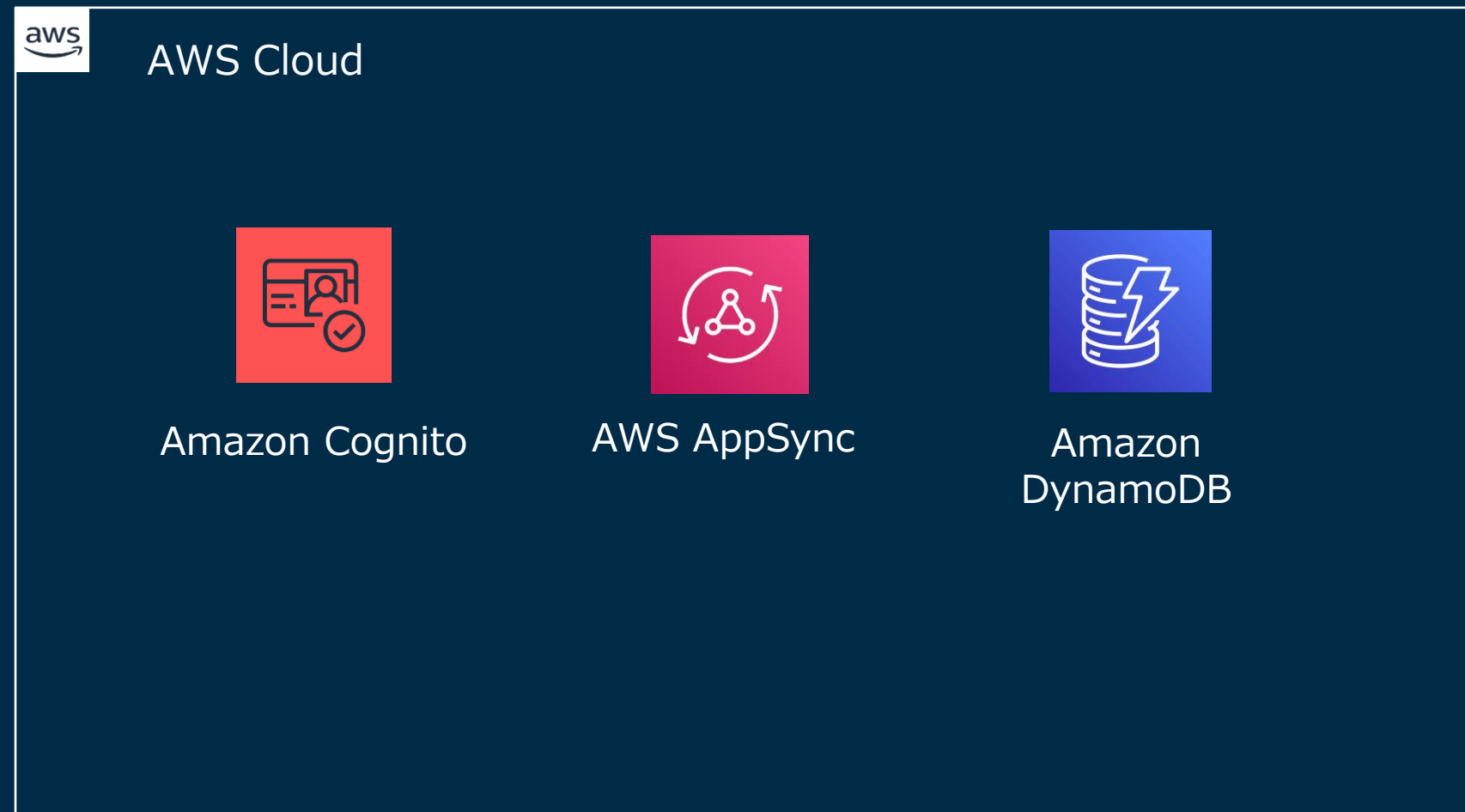
- ✓ バックエンドのリソースに簡単にアクセスできるライブラリの提供

Amplify Console

- ✓ Webアプリケーションのホスティング

AWS Amplify で実現できること ～ CLI 編 ～

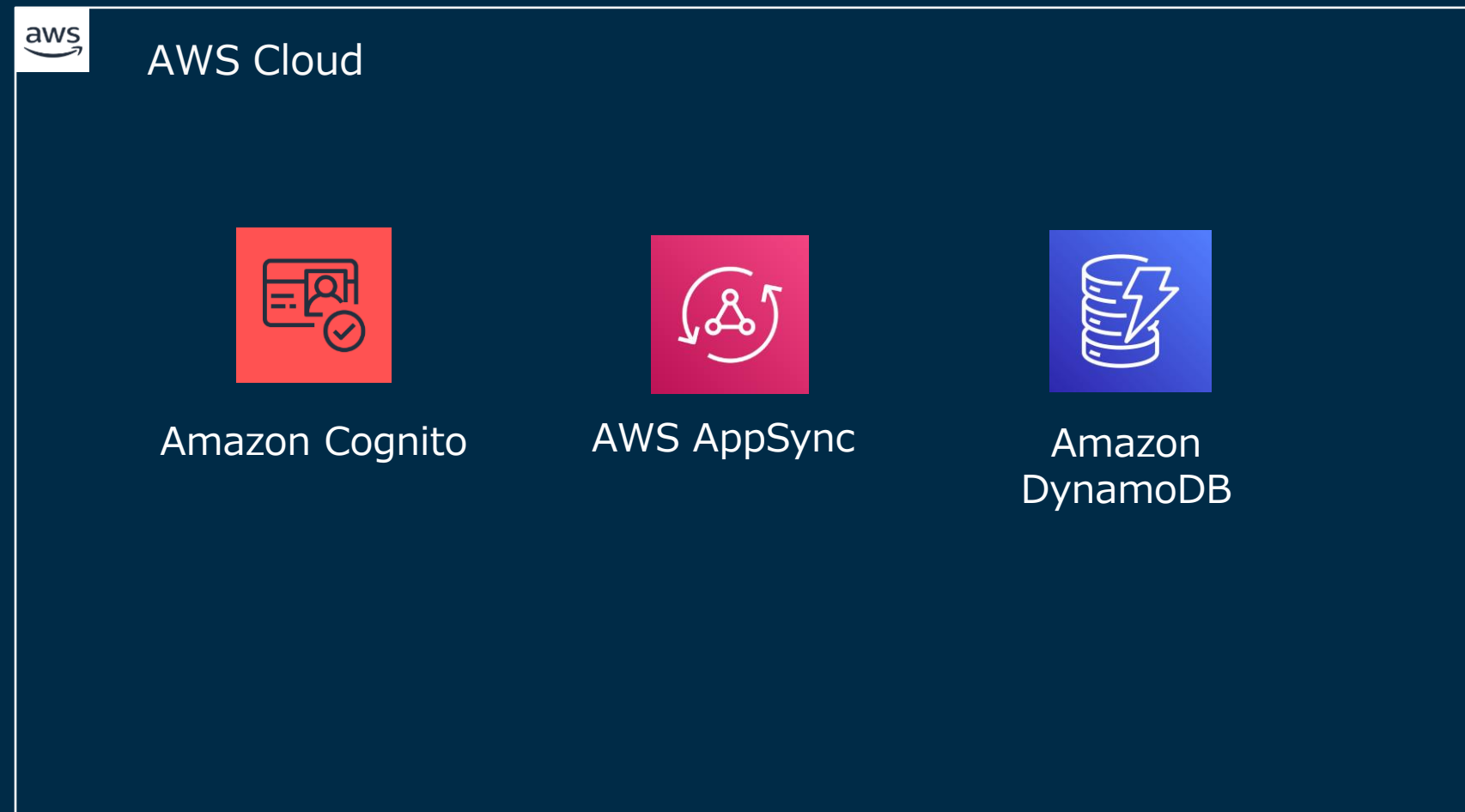
Amplify CLI を用いて様々なバックエンドをコマンドラインで操作、管理が可能



数コマンドでバックエンドを構築

AWS Amplify で実現できること ～ CLI 編 ～

Amplify CLI を用いて様々なバックエンドをコマンドラインで操作、管理が可能



数コマンドでバックエンドを構築

AWS Amplify で実現できること ～ CLI 編 ～

Amplify CLI を用いて様々なバックエンドをコマンドラインで操作、管理が可能



```
$ amplify add api  
$ amplify add auth  
$ amplify push
```

Amplify CLI

数コマンドでバックエンドを構築

対話的にバックエンドの設定が可能

- ✓ 一つ一つの質問に回答することでベストプラクティスに則ったバックエンドが構築
- ✓ AWS CLI のように詳細なオプションを指定する必要がない

```
38f9d358282c:amplify-cli tamizuma$ amplify add api  
? Please select from one of the below mentioned services: GraphQL  
? Provide API name: amplifycli  
? Choose the default authorization type for the API Amazon Cognito User Pool  
Use a Cognito user pool configured as a part of this project.  
? Do you want to configure advanced settings for the GraphQL API No, I am done.  
? Do you have an annotated GraphQL schema? (y/N) █
```

AWS Amplify で実現できること ～ Library 編 ～

- ✓ **Amplify Library** と呼ばれる、直感的なインターフェースで AWS のバックエンドに接続できるライブラリ
- ✓ JavaScript、iOS、Android に対応

例) 写真に写っているオブジェクトを判定する

```
Predictions.identify({
  labels: {
    source: {
      file: files[0]
    },
    type: "ALL"
  }
}).then(
  result => this.labels = result.labels
).catch(
  error => this.error = JSON.stringify(error)
)
```

ファイルを選択 IMG_0334.JPG



約98%の精度で猫(cat)と判定している

Name	Confidence[%]
Cat	98.54254913330078
Mammal	98.54254913330078
Pet	98.54254913330078
Animal	98.54254913330078
Black Cat	94.8594970703125
Manx	55.98295593261719

AWS Amplify で実現できること ～ Console 編 ～

- ✓ Git ベースの SPA (Single Page Application) や静的サイトの開発ワークロードを提供するフルスタックなホスティングサービス
- ✓ マネジメントコンソールから以下のワークロードの管理が可能

CI/CD パイプラインの構築/ホスティング以外にも..

- ✓ CDN、HTTPS のカスタムドメインの構築
- ✓ branch の Auto detection
- ✓ Basic認証によるテスト環境の保護
- ✓ E2E テスト環境を容易に構築 ...etc

■ Amplify Console

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

aws サービス リソースグループ

すべてのアプリ > boyaki

boyaki アクション

The app homepage lists all deployed frontend and backend environments.

▶ Learn how to get the most out of Amplify Console 3 of 5 steps complete

Frontend environments Backend environments

This tab lists all connected branches, select a branch to view build details. ブランチの接続

design-base
Continuous deploys set up with design backend (Edit)

https://design-base...amplifyapp.com

前回のデプロイ
2020/4/12 1:20:21

最終コミット
これは自動生成されたメッセージです | Auto-build | GitHub - design-base

Previews
Disabled

プロビジョニング ビルド テスト デプロイ 検証

具体的なフロントエンド アーキテクチャの構築方法

Big Data

[Architecture. 1]

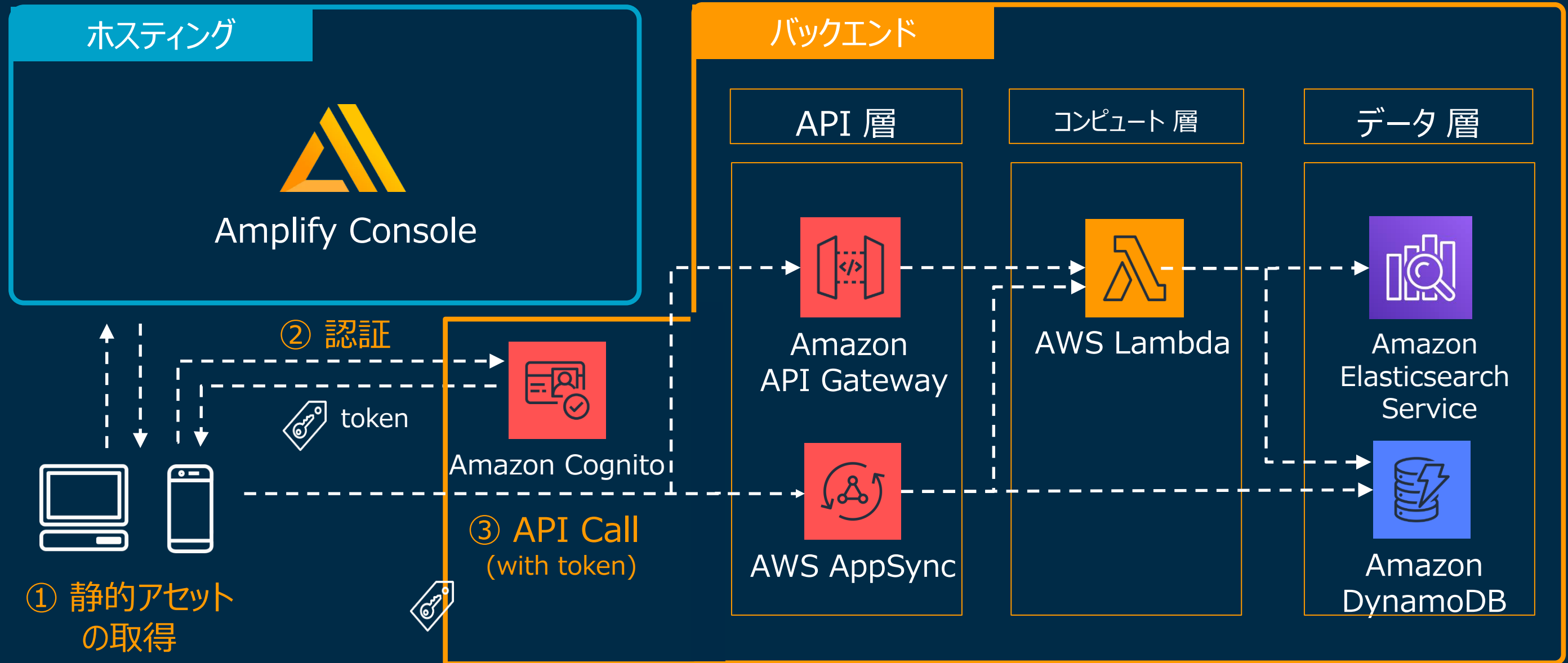
Single Page Application の構成を考える

Architecture Key Points

- ✓ Amplify CLI を用いてバックエンドを構築する
- ✓ Amplify Console を用いて CI/CD を構築する



Single Page Application Architecture



Single Page Application Architecture

[デプロイ方法]

✓ Amplify CLI から対話的にデプロイ

プロジェクトにAmplifyをセットアップ

```
$ amplify init
```

amplify add コマンドで構築する機能の設定を追加
例) 認証機能、API 機能の設定を追加

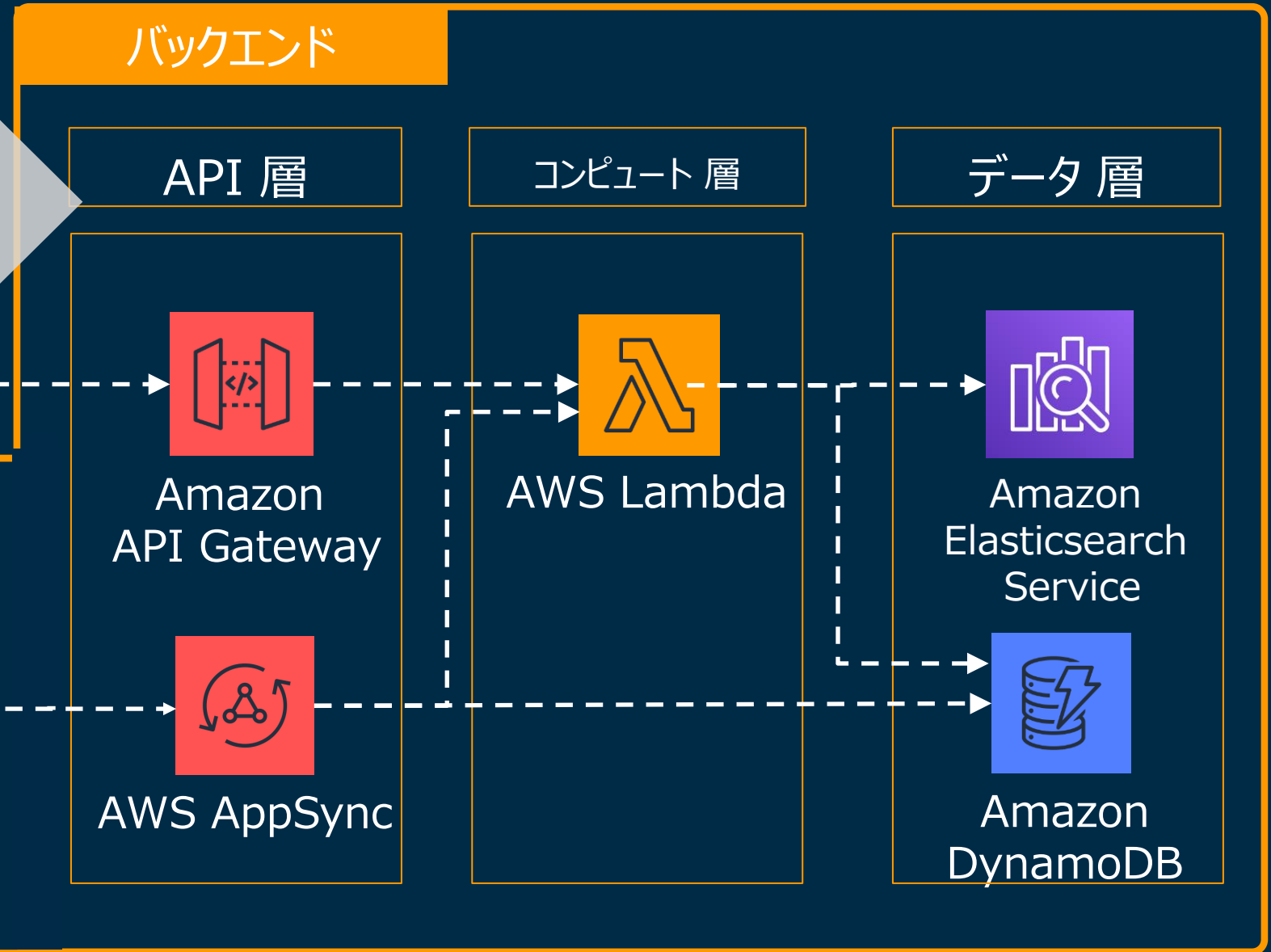
```
$ amplify add auth
```

```
$ amplify add api
```

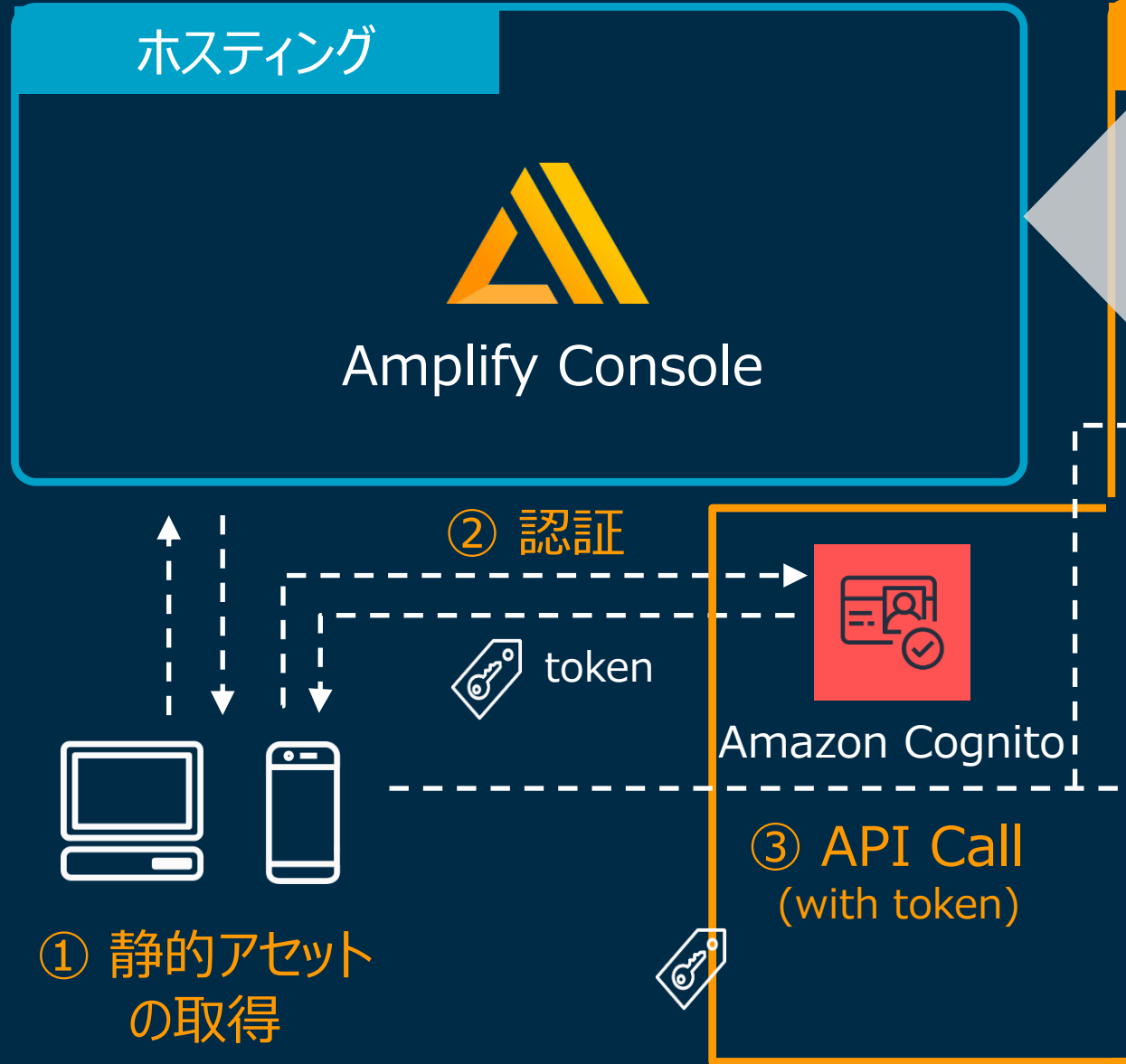
amplify push コマンドで設定を反映

```
$ amplify push
```

① 静的アセット
の取得

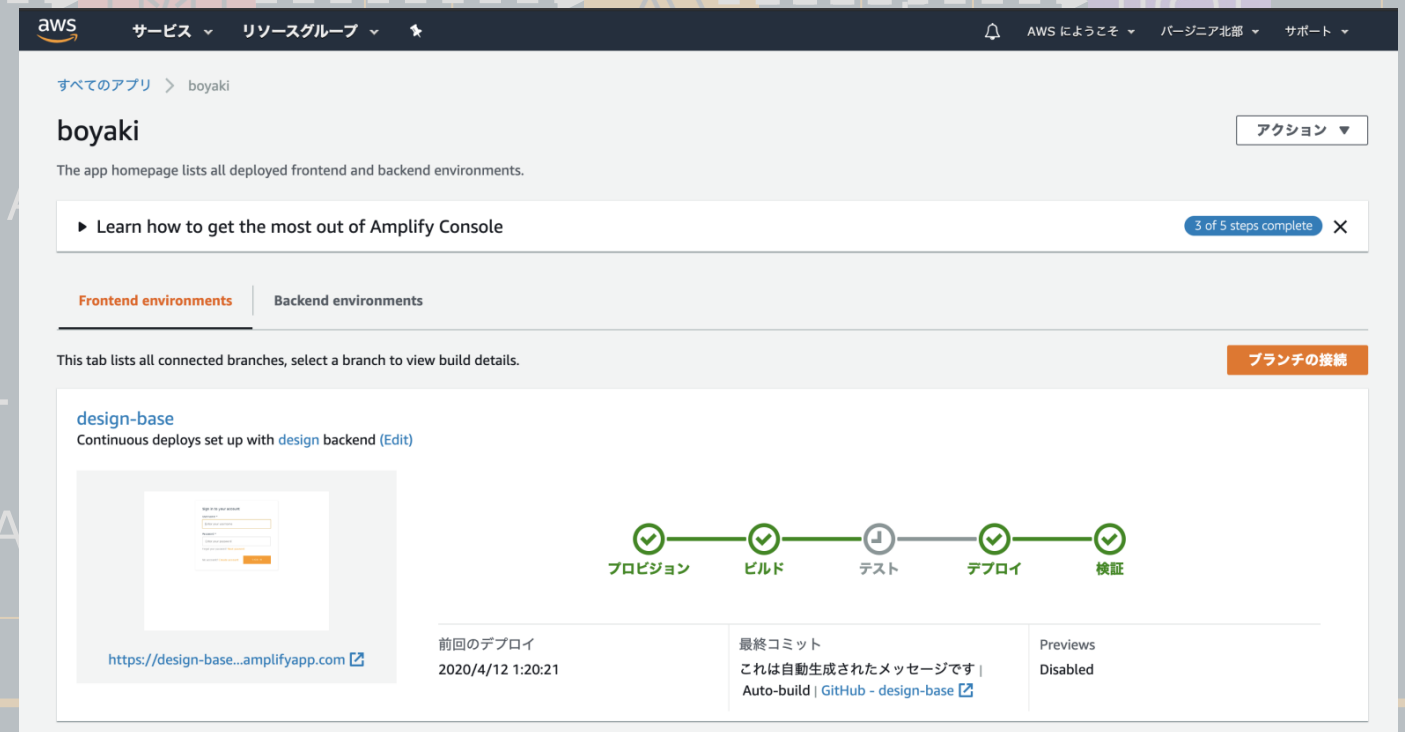


Single Page Application Architecture



[デプロイ方法]

- ✓ Amplify Console を用いてデプロイ
- ✓ ソースリポジトリと連携し、CI/CDパイプラインを構築
- ✓ git push でアプリケーションのデプロイが可能



Single Page Application Architecture Key Points

- ✓ Amplify が想定する最もオーソドックスなアーキテクチャ
- ✓ バックエンドのデプロイには **Amplify CLI**
- ✓ アプリケーションのデプロイには **Amplify Console**



Big Data

[Architecture. 2]

Server Side Rendering の構成を考える

Architecture Key Points

- ✓ SPAと比較して、インフラのコンポーネントが一つ増える
- ✓ AWS 環境へのデプロイに **Serverless Framework** を用いる

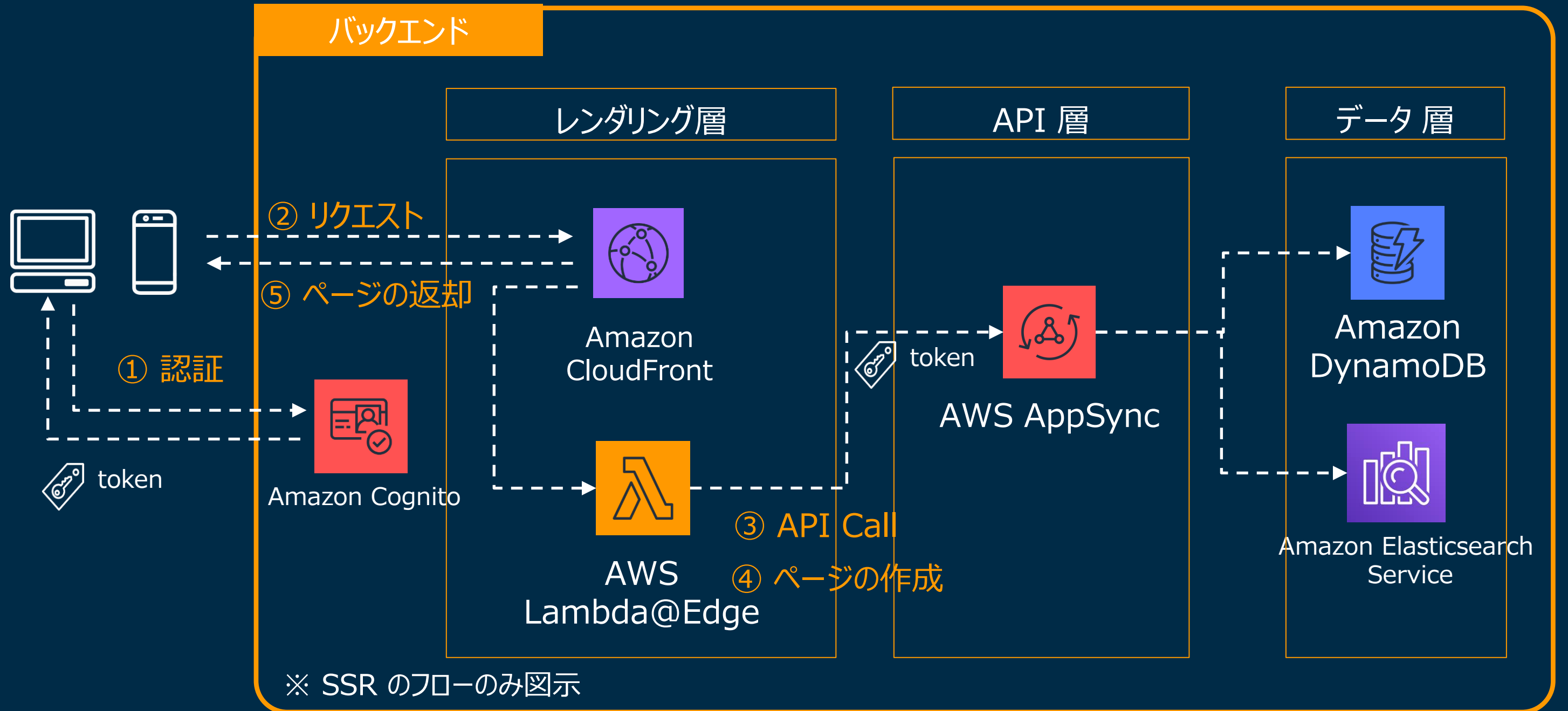


Amplify Library が Server Side Rendering をサポート

New!

- ✓ 2020年9月に、Amplify Library が Next.js や Nuxt.js などのフレームワークを用いた Server Side Rendering をサポート
- ✓ レンダリング層に **CloudFront**、**Lambda@Edge** を使用
- ✓ レンダリング層のデプロイには **Serverless Framework** を用いる
 - ※ Amplify Console を用いた静的ファイルのデプロイとは異なる点に注意！

Server Side Rendering Architecture



Server Side Rendering Architecture

[デプロイ方法]

✓ Amplify CLI から対話的にデプロイ

例) 認証機能、API 機能の設定を追加

```
$ amplify add auth
```

```
$ amplify add api
```

amplify push コマンドで設定を反映

```
$ amplify push
```



① 認証



token



Amazon Cognito

CloudFront



AWS
Lambda@Edge

③ API Call

④ ページの作成

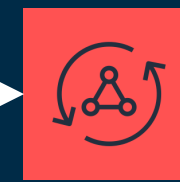
レンダリング層

API 層

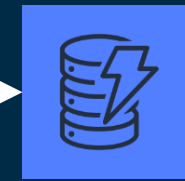
データ層



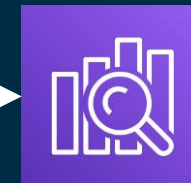
token



AWS AppSync



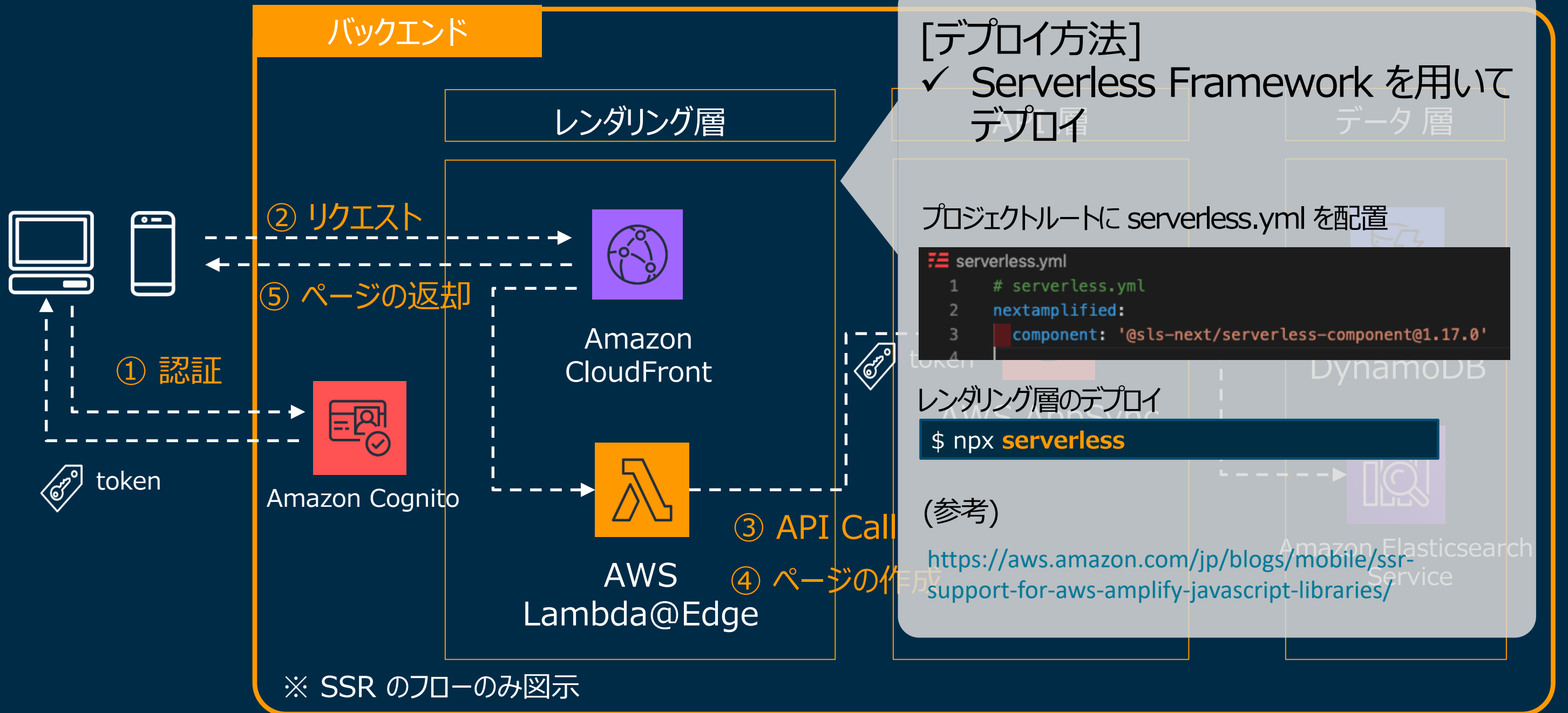
Amazon
DynamoDB



Amazon Elasticsearch
Service

※ SSR のフローのみ図示

Server Side Rendering Architecture



Server Side Rendering Architecture Key Points

- ✓ SPA と比較して レンダリング層がバックエンドの要素として増える
- ✓ バックエンドのデプロイには **Amplify CLI**
- ✓ アプリケーションのデプロイには **Serverless Framework**
- ✓ **Amplify Console** を用いた静的ファイルのデプロイと方式が異なる点に注意

Big Data

[Architecture. 3]

Static Site Generator を用いた JAMstack 構成を考える

Architecture Key Points

- ✓ 静的サイト構築のフレームワークに **Gatsby** を使用
- ✓ **Amplify Console** を用いて静的ファイルを生成する
- ✓ 静的ファイル生成のためのデータソースへの参照には Gatsby のプラグインを用いる



Gatsby について



- React ベース SSG フレームワーク
- ビルド時のデータソースへのアクセスに GraphQL が使用可能
- 豊富な Plugin を用いて様々なデータソースにアクセスが可能

[AWS のリソースにアクセスするためのプラグイン例] ※ 2020/10 現在

- gatsby-source-dynamodb

<https://github.com/wmlutz/gatsby-source-dynamodb>

- gatsby-source-s3

<https://github.com/robinmetral/gatsby-source-s3>

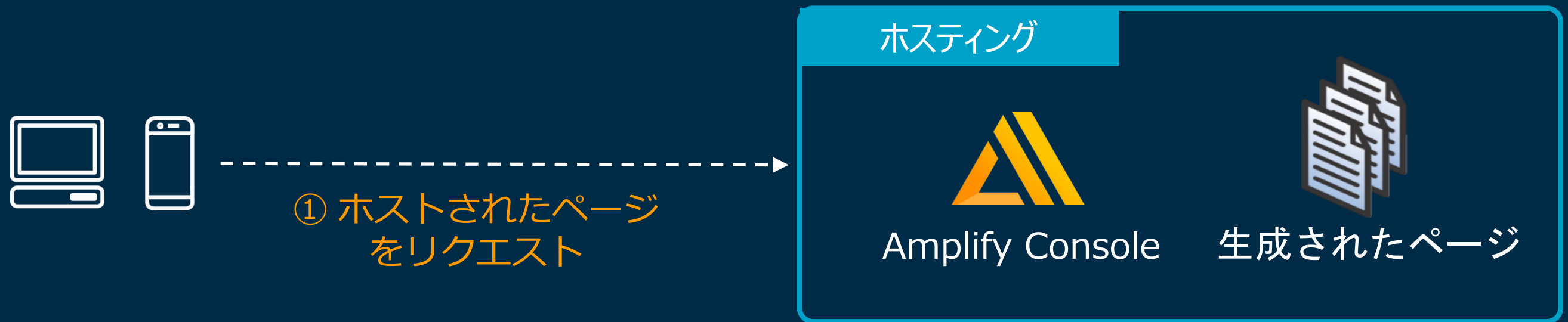
- gatsby-source-serverless-aurora

<https://github.com/saikojosh/gatsby-source-serverless-aurora>

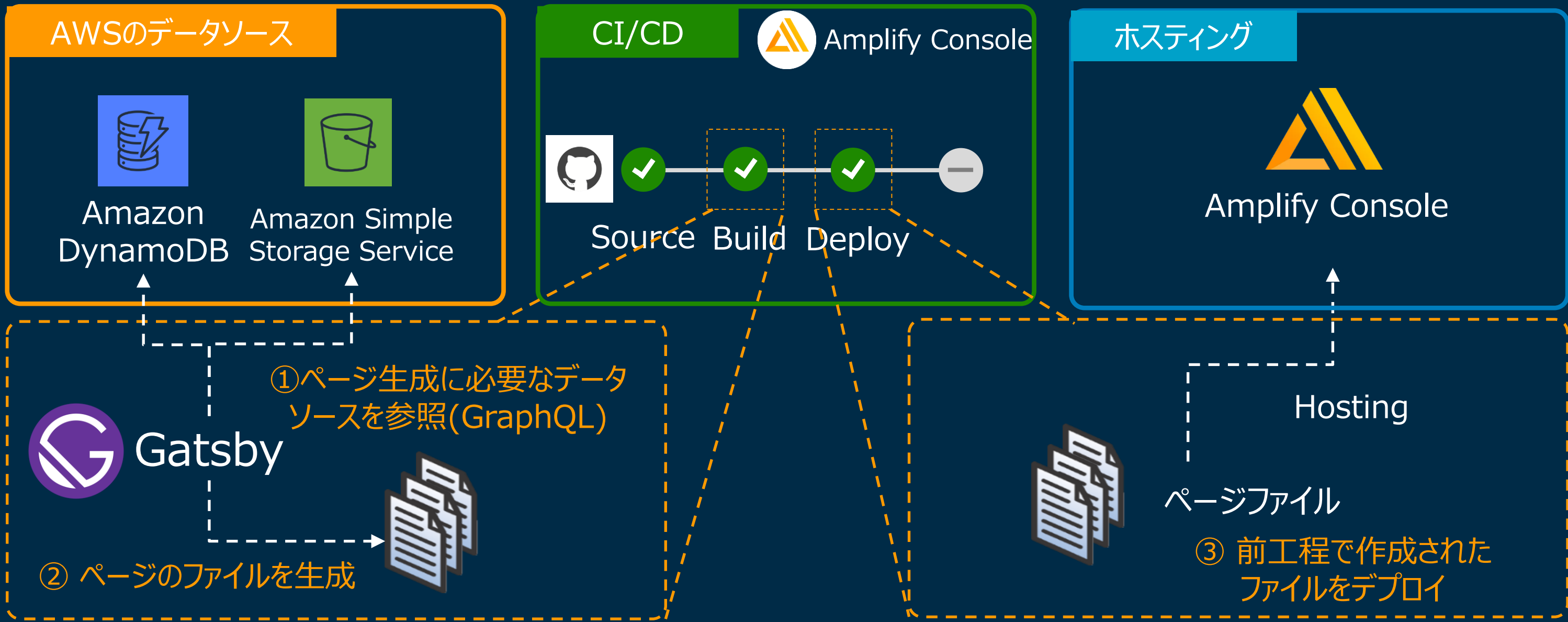
※ いずれもAWS が公式にサポートしているものではありません



JAMstack, Static Site Generator Architecture



JAMstack, Static Site Generator Architecture



JAMstack, Static Site Generator Architecture Key Points

- ✓ **Amplify Console** で静的サイトの作成、ホスティングを行う
- ✓ 静的サイト生成時のデータソースには Gatsby のプラグインを使用
- ✓ データソースの参照はビルド時にのみ実施されるため、リアルタイムにデータをS取得する場合はクライアントからAPIを呼び出す

本日のまとめ

- **AWS Amplify** はフロントエンドエンジニアの強い味方
- バックエンドを強く意識することなくアプリケーションの開発、リリースが可能

Amplify 関連セッション

Day 1, 10/20

AWS Amplify 実践編

11:45 – 12:25 | Track B

Day 2, 10/21

Amplify CLI Deep Dive

14:30 – 15:00 | Track F

Day 3, 10/22

あなたもCTOになろう！AWS Amplify で爆速開発ワークショップ

13:00 – 15:00 | WSB-2



Amplify についてさらに知りたい方に



aws

[AWS Black Belt Online Seminar]

AWS Amplify

サービスカットシリーズ

Solutions Architect 水馬 拓也
2020/05/20

AWS 公式 Webinar
<https://amzn.to/JPWebinar>

過去資料
<https://amzn.to/JPArchive>

AWS Amplify Black Belt

- ✓ Amplify の概要について解説
- ✓ まずは体系的に Amplify を学びたい方にオススメ

<https://aws.amazon.com/jp/blogs/news/webinar-bb-aws-amplify-2020/>



aws

Search...

1. 前提条件
2. はじめに
3. MVPを作ろう！
4. ウェブサイトホスティング
5. Follow/Timeline機能の実装
6. 全文検索機能の追加
7. 複数メンバーでの開発
8. E2Eテスト
9. 終わりに
10. 補足

AMPLIFY SNS WORKSHOP

Amplify SNS Workshopへようこそ！本ワークショップではTwitterライクなソーシャルメディアアプリケーションの開発を通して、実践的に AWS Amplify について学ぶことができます。

本ハンズオンは、AWS Startup Day で聴講頂ける「これからはじめるAWS Amplify」および「AWS Amplify 実践編」をご覧になった上での参加を推奨しております。セッションとハンズオンを通して、AWS Amplify の理解を深めていきましょう！

対象者

- 爆速で開発したいスタートアップエンジニア

Amplify SNS Workshop

- ✓ React で作る実践型 Workshop
- ✓ Amplify の概要だけでなく、実運用を見据えたテクニックが知りたい方にオススメ

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

Thank you!

水馬 拓也

Contact information



@mizuma_t

tamizuma@amazon.co.jp