[AWS Black Belt Online Seminar]

Amazon API Gateway

サービスカットシリーズ

Solutions Architect 松原 武司

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

過去資料
https://amzn.to/JPArchive

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

自己紹介

- 名前
  - 松原 武司
- 所属
  - アマゾン ウェブ サービス ジャパン 株式会社
  - 技術統括本部 金融ソリューション部
  - ソリューション アーキテクト
- 好きなAWSのサービス

Amazon API Gateway

AWS Cloud9

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

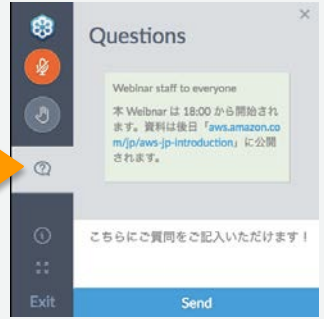
AWS Black Belt Online Seminar とは

「サービス別」「ソリューション別」「業種別」のそれぞれのテーマに分かれて、アマゾン ウェブ サービス ジャパン株式会社が主催するオンラインセミナーシリーズです。

質問を投げることができます！

- 書き込んだ質問は、主催者にしか見えません
- 今後のロードマップに関するご質問は
  お答えできませんのでご了承下さい

① 吹き出しをクリック
② 質問を入力
③ Sendをクリック



Twitter ハッシュタグは以下をご利用ください
#awsblackbelt

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

内容についての注意点

- 本資料では2019年5月14日時点のサービス内容および価格についてご説明しています。最新の情報はAWS公式ウェブサイト(http://aws.amazon.com)にてご確認ください。

- 資料作成には十分注意しておりますが、資料内の価格とAWS公式ウェブサイト記載の価格に相違があった場合、AWS公式ウェブサイトの価格を優先とさせていただきます。

- 価格は税抜表記となっています。日本居住者のお客様が東京リージョンを使用する場合、別途消費税をご請求させていただきます。

- AWS does not offer binding price quotes. AWS pricing is publicly available and is subject to change in accordance with the AWS Customer Agreement available at http://aws.amazon.com/agreement/. Any pricing information included in this document is provided only as an estimate of usage charges for AWS services based on certain information that you have provided. Monthly charges will be based on your actual use of AWS services, and may vary from the estimates provided.

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

本日のアジェンダ

- はじめに - API / Web API とは
- Amazon API Gateway 概要
    - 全体イメージ
    - ユースケース
    - REST API / WebSocket API
    - その他主要な機能
    - 補足的な機能 や ご使用料金の情報など
- まとめ

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

本日のアジェンダ

- はじめに - API / Web API とは
- Amazon API Gateway 概要
    - 全体イメージ
    - ユースケース
    - REST API / WebSocket API
    - その他主要な機能
    - 補足的機能 や ご使用料金の情報など
- まとめ

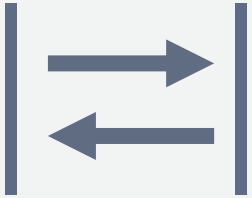
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

今日お話しするテーマ の “API”

“Application Programming Interface”：
プログラムやソフトウェア同士がやり取りするための取り決め・仕様

言語処理系API
[SDK/ライブラリ等]

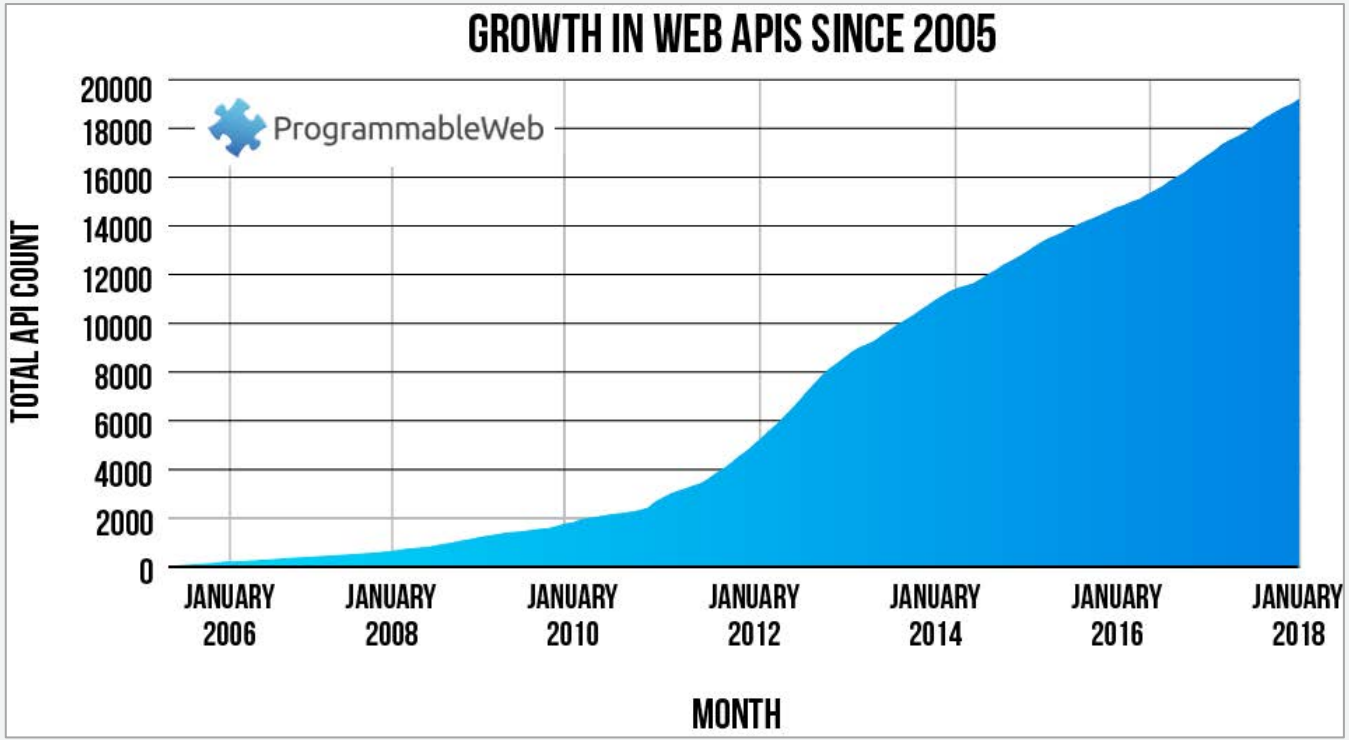


ネットワーク
呼び出しAPI

Web API

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

増え続けるWeb API

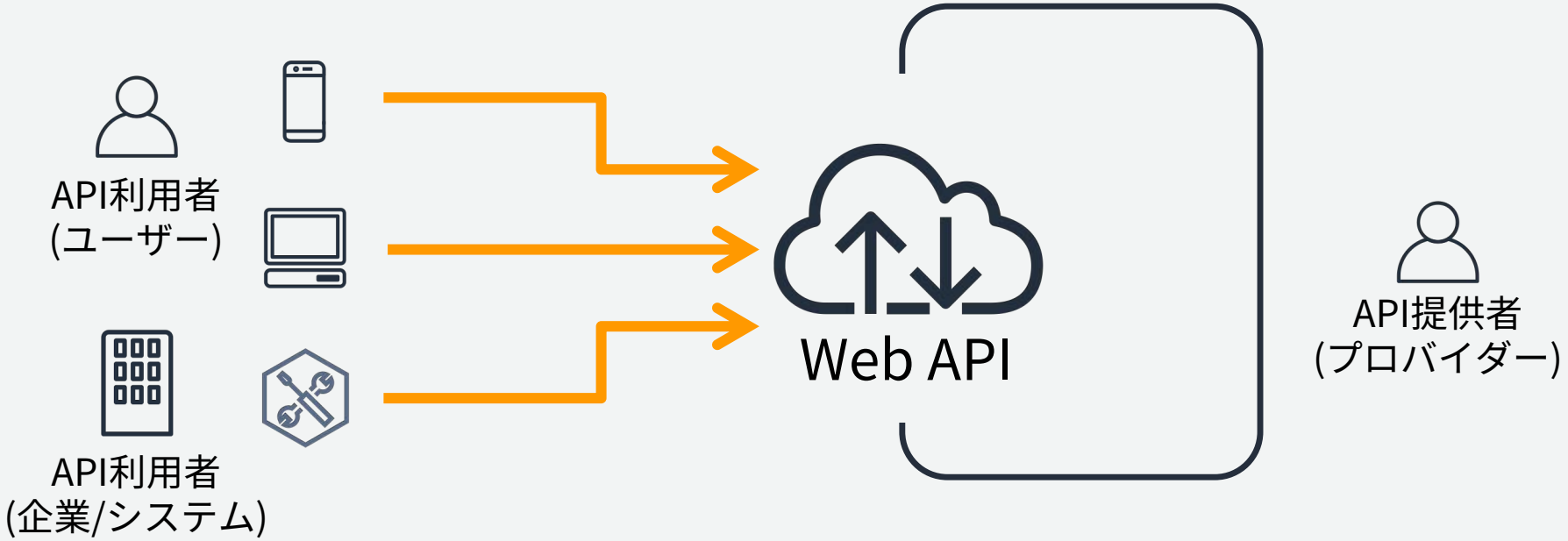


(出典) programmableweb.com 記事「Research Shows Interest in Providing APIs Still High」(2018/2/23) より
https://www.programmableweb.com/news/research-shows-interest-providing-apis-still-high/research/2018/02/23

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Web API 増加の背景

緩やかにシステムが連携し APIによるエコシステム を構築するスタイルが
Webの発展に上手く合致



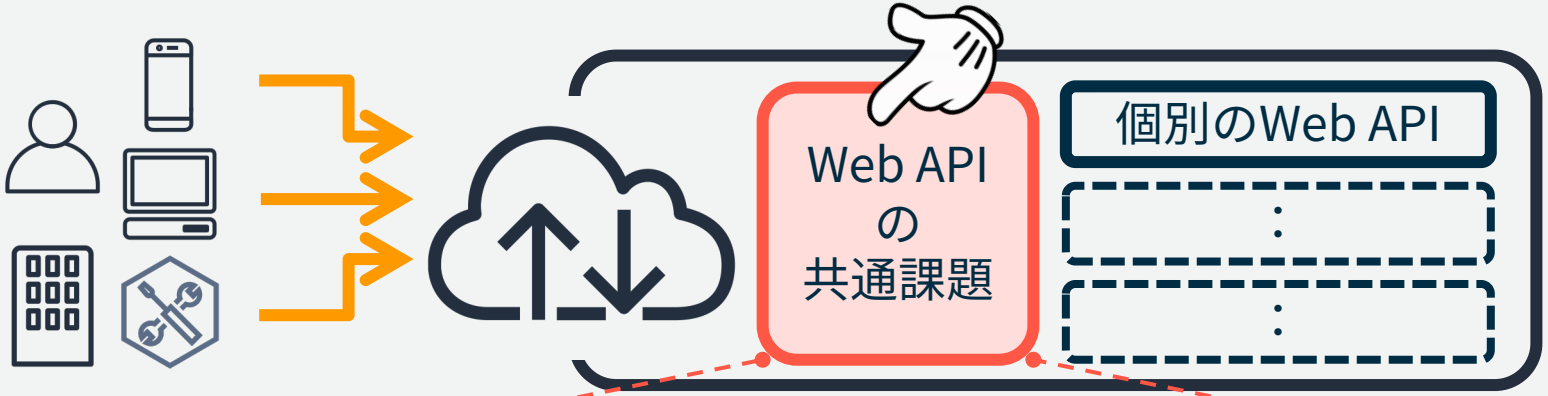
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

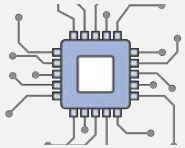
AWS と Web API

AWSもまた、数多くの Web API をご提供しています

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Web API提供時の課題





1. インフラの管理
(可用性とスケーラビリティ)

2. APIの管理
(設定やデプロイの制御)

3. 認証と認可
(アクセスの制御)

4. 流量制御と保護
(スロットリング)

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

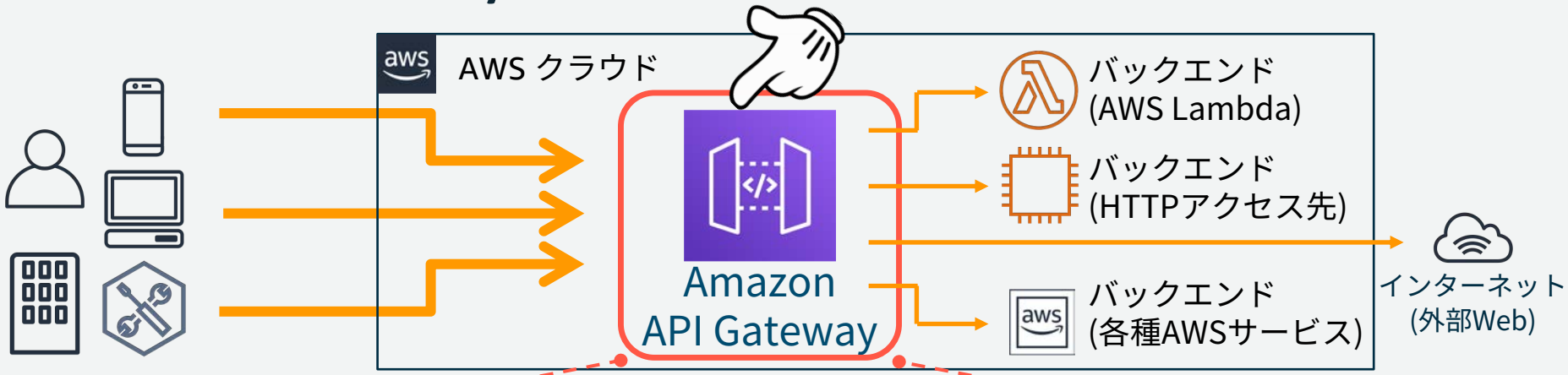
本日のアジェンダ

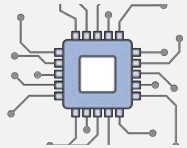
- はじめに - API / Web API とは
- Amazon API Gateway 概要
    - 全体イメージ
    - ユースケース
    - REST API / WebSocket API
    - その他主要な機能
    - 補足的機能 や ご使用料金の情報など
- まとめ

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Amazon API Gateway の位置づけ





1. インフラの管理
(可用性とスケーラビリティ)

2. APIの管理
(設定やデプロイの制御)

3. 認証と認可
(アクセスの制御)

4. 流量制御と保護
(スロットリング)

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Amazon API Gateway が利用可能なリージョン

Amazon API Gateway は 21 のリージョン全てでご利用可能
(2019/05/14現在、大阪ローカルリージョンを除く)

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Amazon API Gateway の主なユースケース概要

1. インターネットからアクセス可能なパブリックなWeb APIの基盤を提供する

2. 自社企業・企業グループ内でのプライベートなWeb APIの基盤を提供する

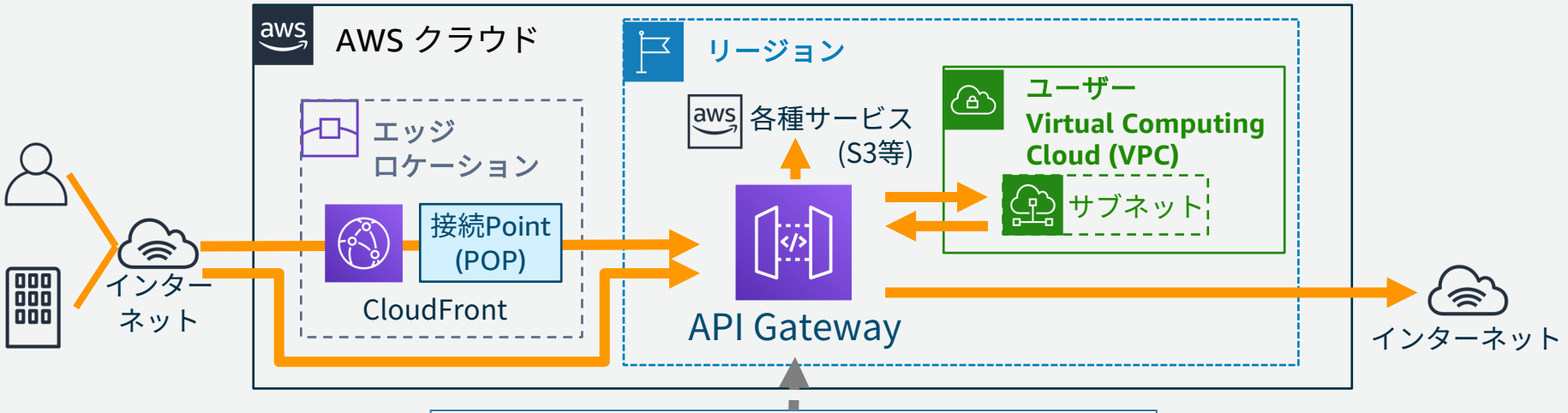
3. AWSサービス(例：Amazon DynamoDB 等)を独自のWeb API化する手段として利用する

4. サーバーレスアーキテクチャを実現する手段として利用する

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

AWSサービスとしての Amazon API Gateway

Amazon API Gateway (以降、「API Gateway」と記載) は、
フルマネージド型のサービス

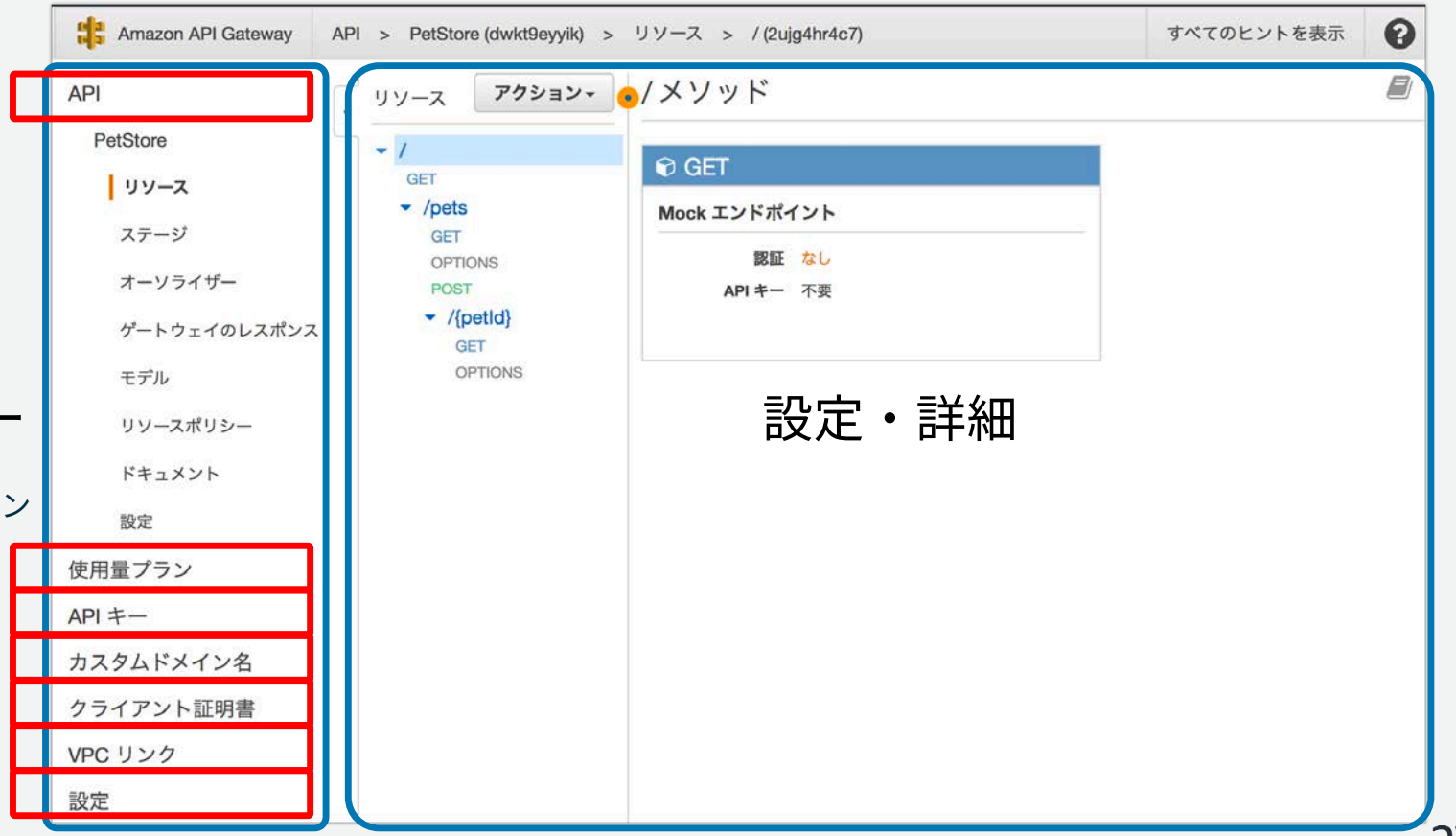


マネージドサービスのメリット

- 自動でスケール
- 仮想サーバー(OS)管理不要
- 使用量に応じた課金

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

API Gateway 管理コンソール

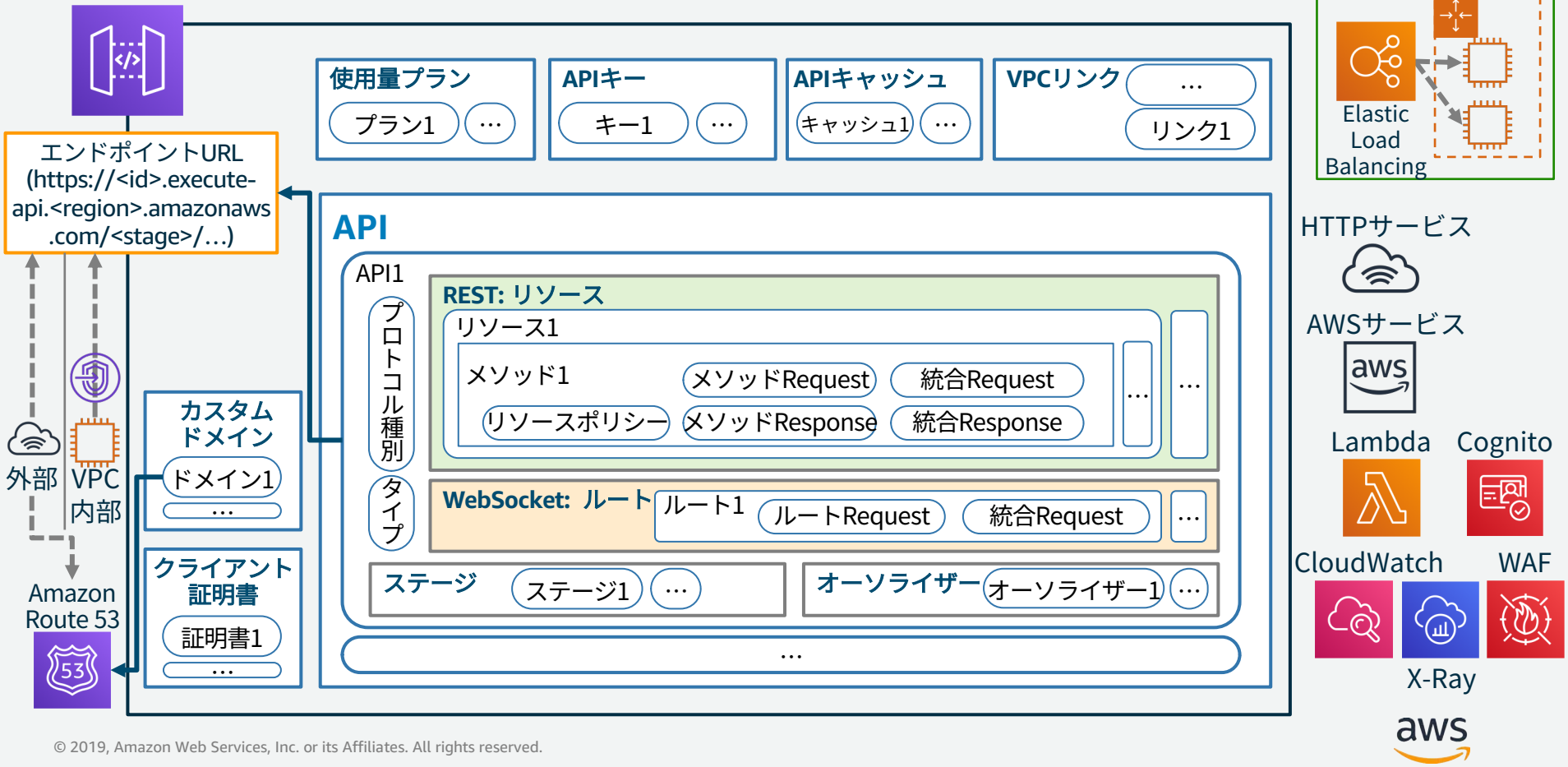


メニュー
(プライマリナビゲーションペイン)

設定・詳細

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

API Gateway の概要図



© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

API Gateway が扱うAPI

API Gatewayは「REST」「WebSocket」の2種類のAPIを扱うことが可能

(※どちらもセキュア接続のみ)

REST

“ステートレス”

- REST= **Re**presentational **s**tate **t**ransfer
- 単一HTTPメッセージで1つの操作に関する情報を(理想では)含む
- 扱う情報をURIで表現する「リソース」として定義し、それらをHTTPメソッド(PUT, GET, POST, DELETE, …)の表現で操作

WebSocket

“ステートフル”

2018/12/19 Updateで対応

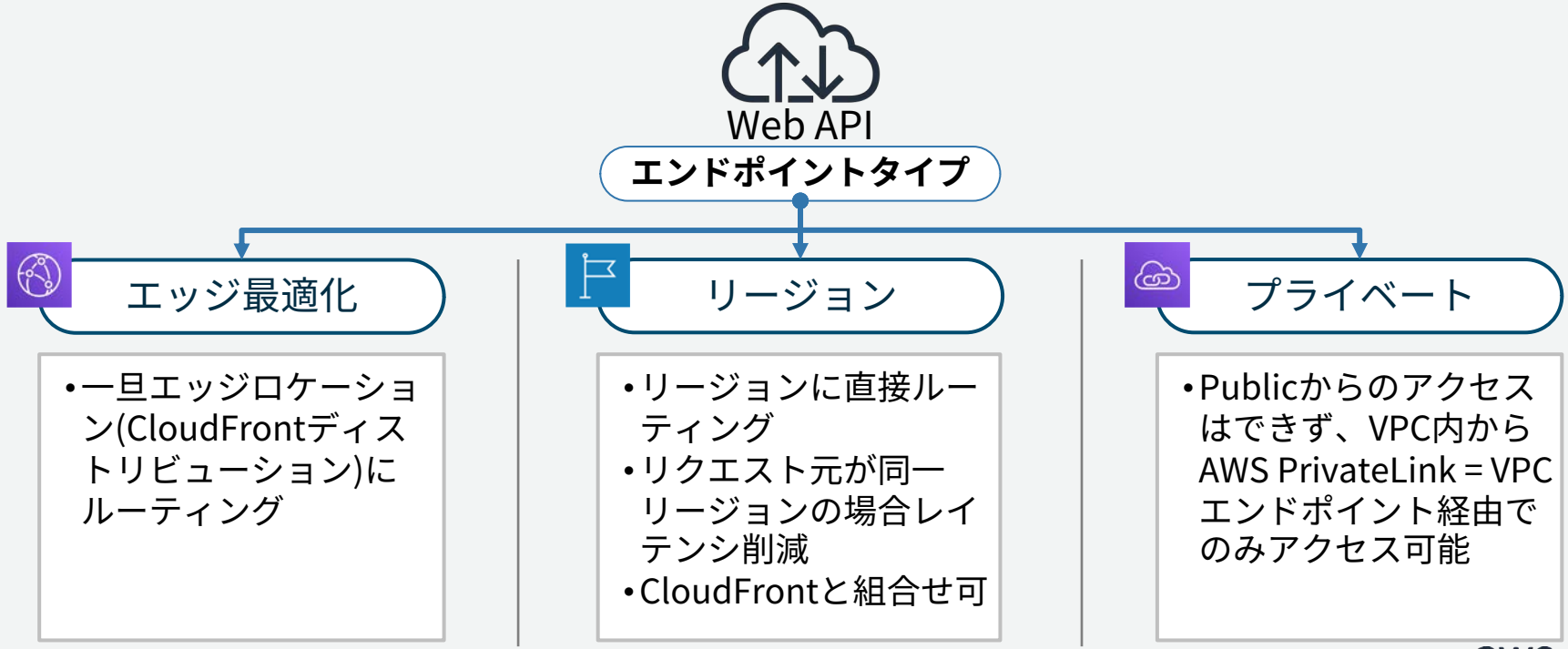
- HTTPの上で、クライアントとサーバー間の双方向通信を実現するための通信仕様
- 一つのコネクションで継続的なデータ送受信が可能
- URIスキームは セキュアWebSocket用の「wss://~」を利用

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

API Gateway が扱うAPI - REST エンドポイントタイプ

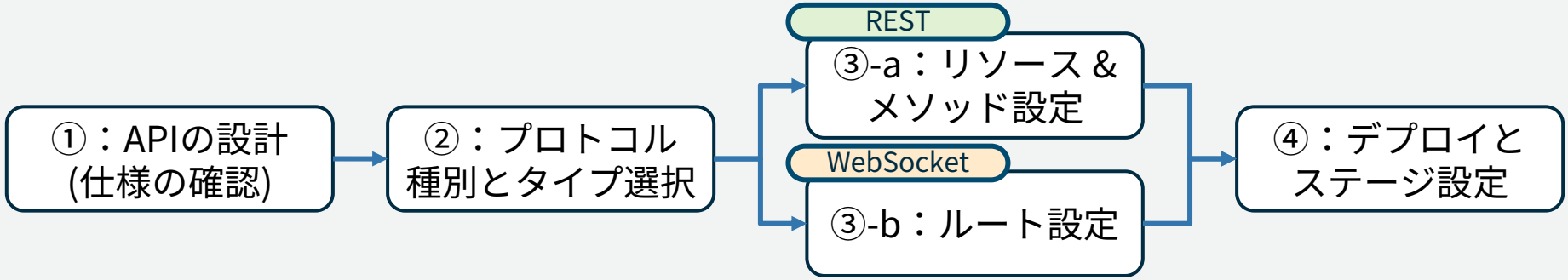
REST

REST APIの場合、3種類のエンドポイントタイプから1つを選択
(クライアントから見たアクセス先エンドポイントとしての性質を決定)



© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

API 作成の流れ (新規作成)



- 作成するAPIの要求・仕様を確認
- 以降の作業で行う設定項目を検討

- API仕様に基づきREST or WebSocketを選択
- RESTの場合はエンドポイントタイプを選択
  - ✓ リージョン
  - ✓ エッジ最適化
  - ✓ プライベート

- プロトコル種別に応じたAPIの設定を実施
- 設定後、APIとしてのテスト呼び出しが可能
(公開前の状態におけるテスト)

- デプロイ操作(API単位)によりクライアントから呼び出し可能な状態へ
- デプロイ先ステージを新規または既存から選択し、ステージの設定と接続テスト

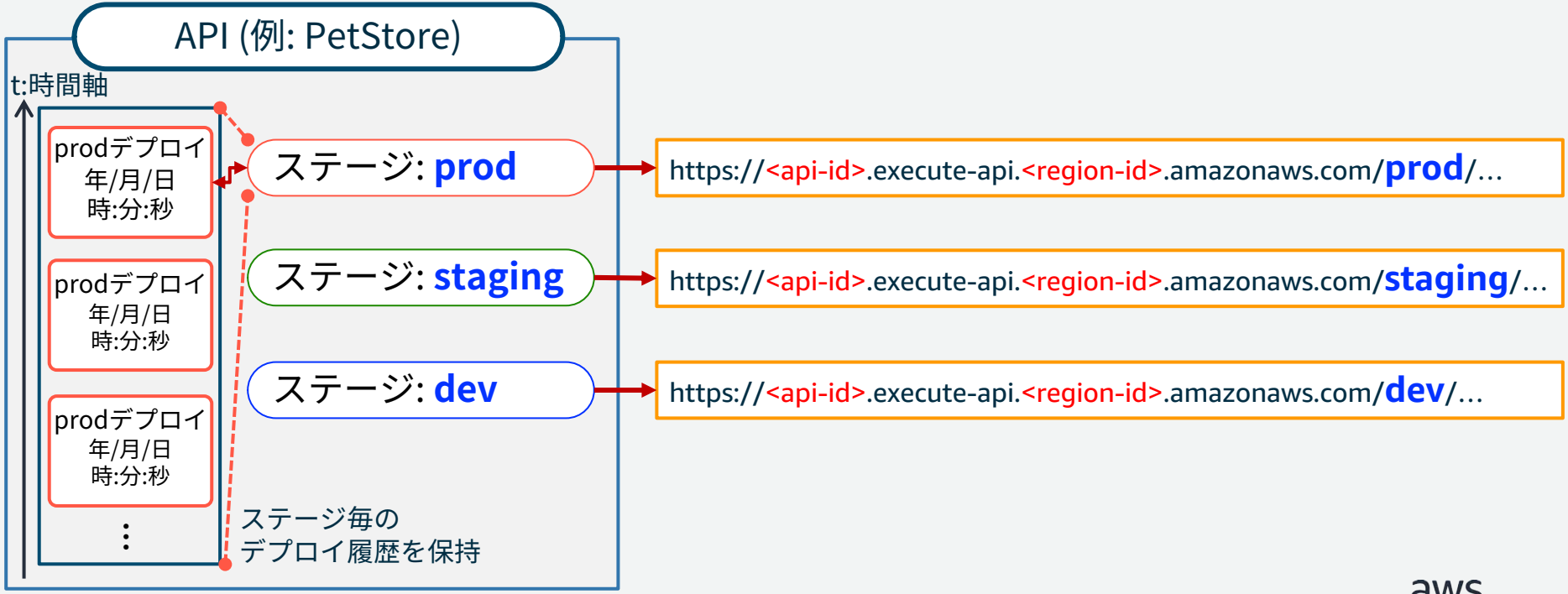
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 デプロイとステージ

REST WebSocket

APIは ステージ に“デプロイ”することでクライアントから呼び出し可能に
(ステージ名はエンドポイントURL(パス)の一部として利用される)



© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

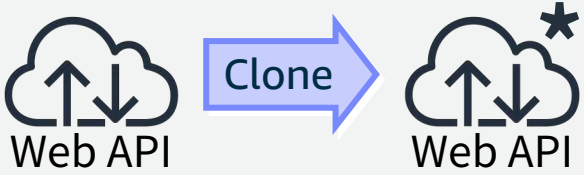
aws

API 作成の流れ (新規作成 - RESTの場合のみ)

REST

「既存APIのクローン」または「Swagger(OpenAPI)ファイルのインポート」による作成が可能

既存APIの
クローン



- 内容をコピーしつつ、別の名前とIDを持つ新規のAPIとして作成
- 互換性がない変更が行われたAPIを別APIとして公開したい場合に有用

Swaggerファイル
のインポート

OpenAPI 2.0 または 3.0

- Swagger2.0/3.0(OAS2.0/3.0) のファイル(JSON, YAML)をインポート可能
- API Gateway固有の定義は x-amazon-apigateway-* で指定可能

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

本日のアジェンダ

- はじめに - API / Web API とは
- Amazon API Gateway 概要
    - 全体イメージ
    - ユースケース
    - REST API / WebSocket API
    - その他主要な機能
    - 補足的機能 や ご使用料金の情報など
- まとめ

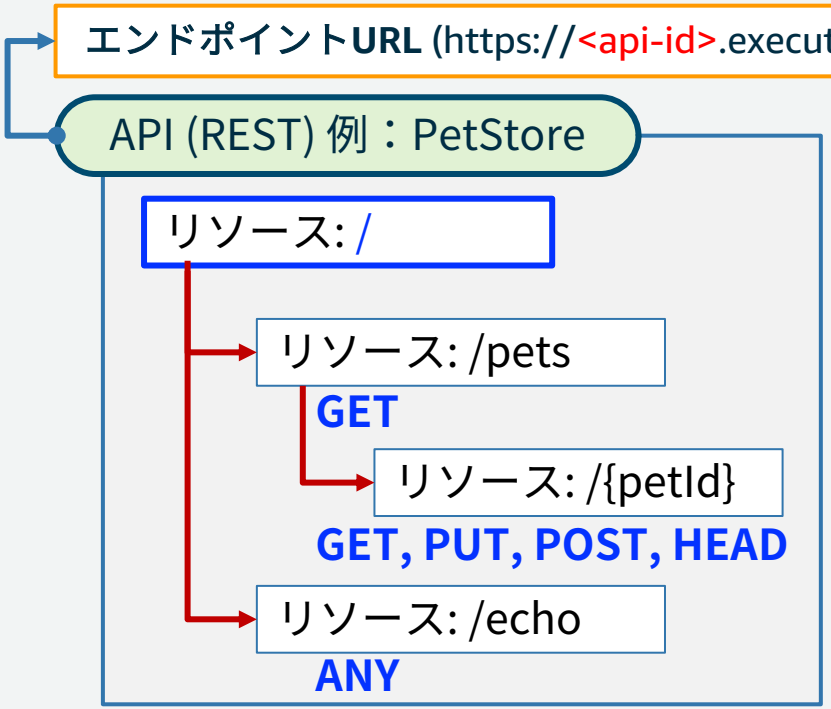
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 (REST) - リソースとメソッド

REST

「/」を最上位としたツリー構造にて「リソース」を定義。
各リソースに受け付けるHTTPメソッドを指定 (パスパラメータの利用も可能)



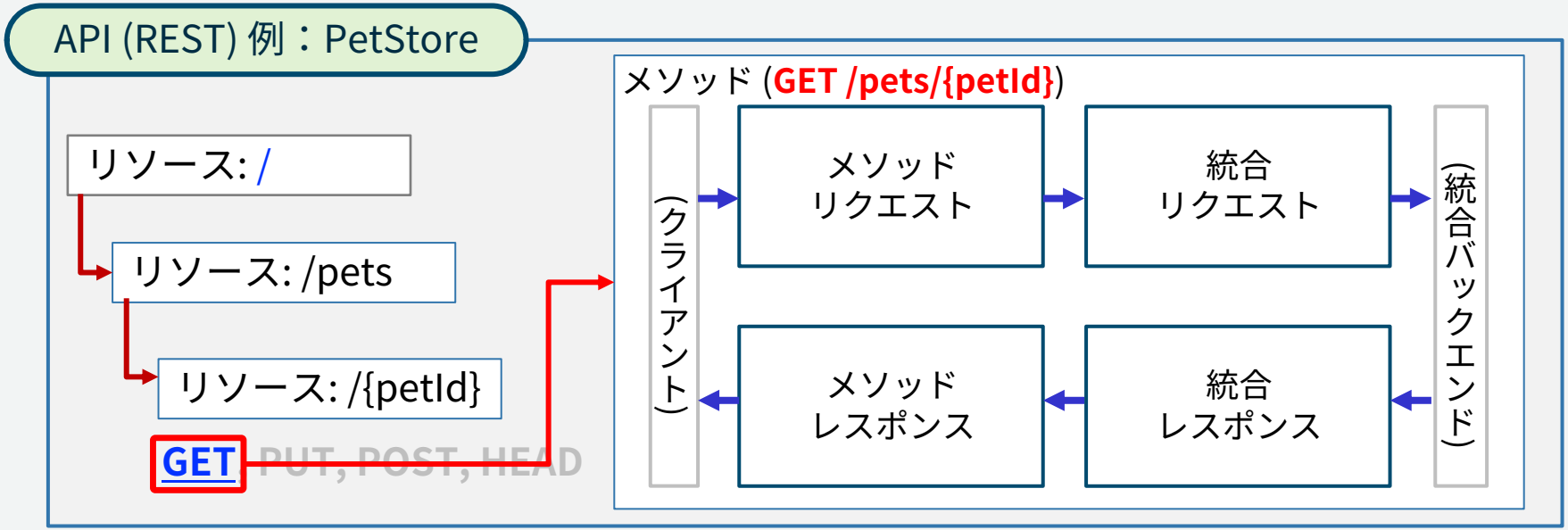
- 各APIには「英数小文字からなる識別子」が割り振られ、URL(ドメイン名の一部)を構成
  - 各APIは常に"/"リソースを起点として持つ
- "/"リソース配下にリソースを複数構成可能
  - リソース名がURL(パス)の一部を構成
- 各リソースにメソッドを複数構成可能
  - メソッド毎にフローを構成
- メソッドは7つのHTTPメソッドとANYから選択
  - GET, POST, PUT, HEAD, DELETE, OPTIONS, PATCH または ANY

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 (REST) - メソッド設定

REST



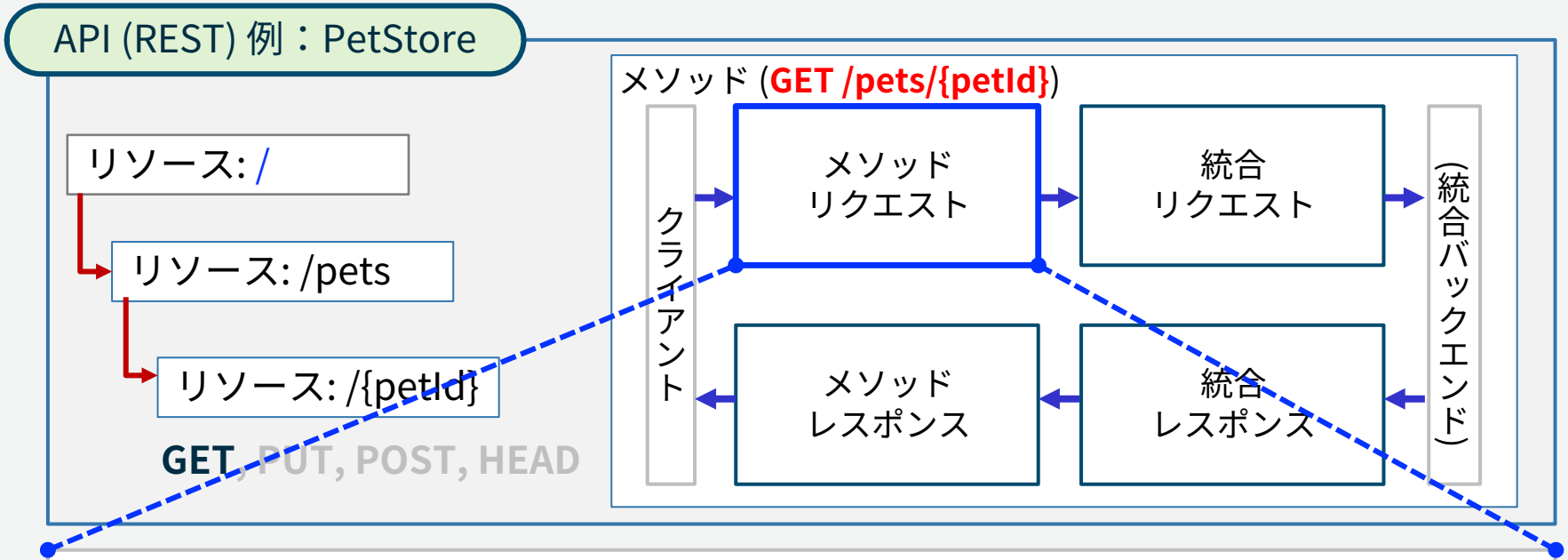
- リソースとメソッドの組合わせ毎に、合計4つの設定箇所で1つのフロー処理を定義
  - メソッドリクエスト→統合リクエスト → 統合レスポンス→メソッドレスポンス

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 (REST) - メソッド設定

REST



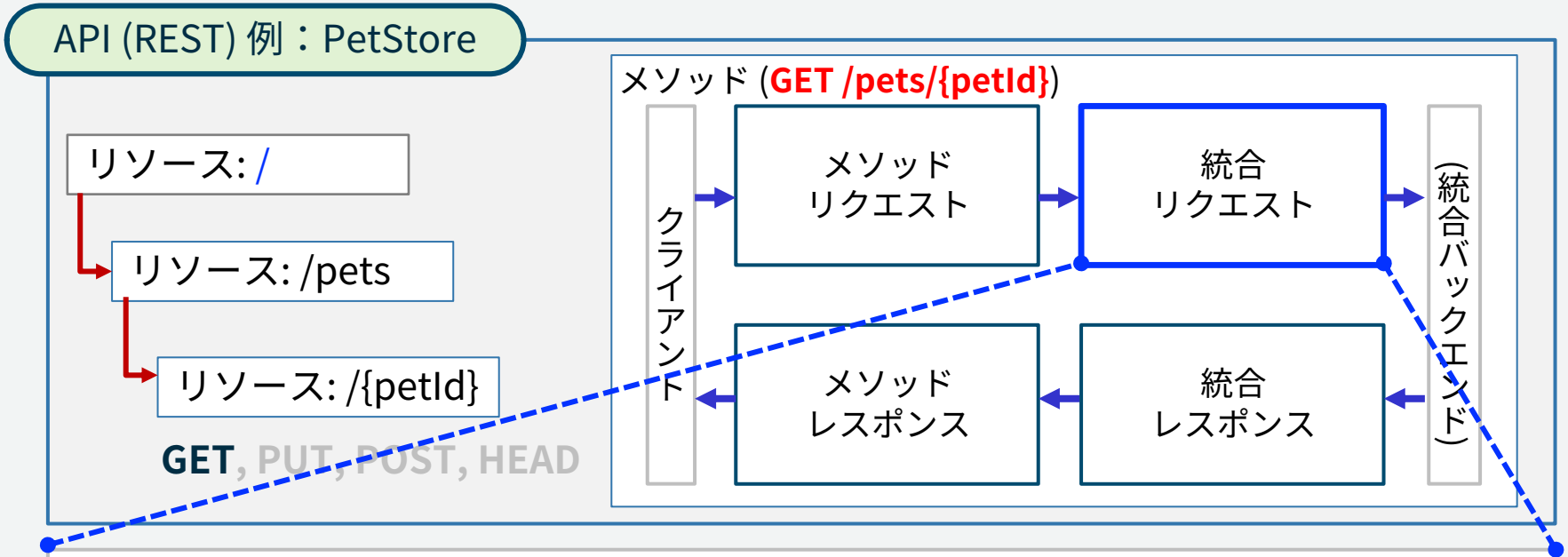
- 認証の設定や、受け付けるクエリパラメータ、必須とするHTTPヘッダなど「リクエストの受付」に関する設定
  - 設定に合致するリクエストかの検証の有無
  - リクエスト時にAPIキー(HTTPヘッダ x-api-key で指定) を必須とするかどうか

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 (REST) - メソッド設定

REST



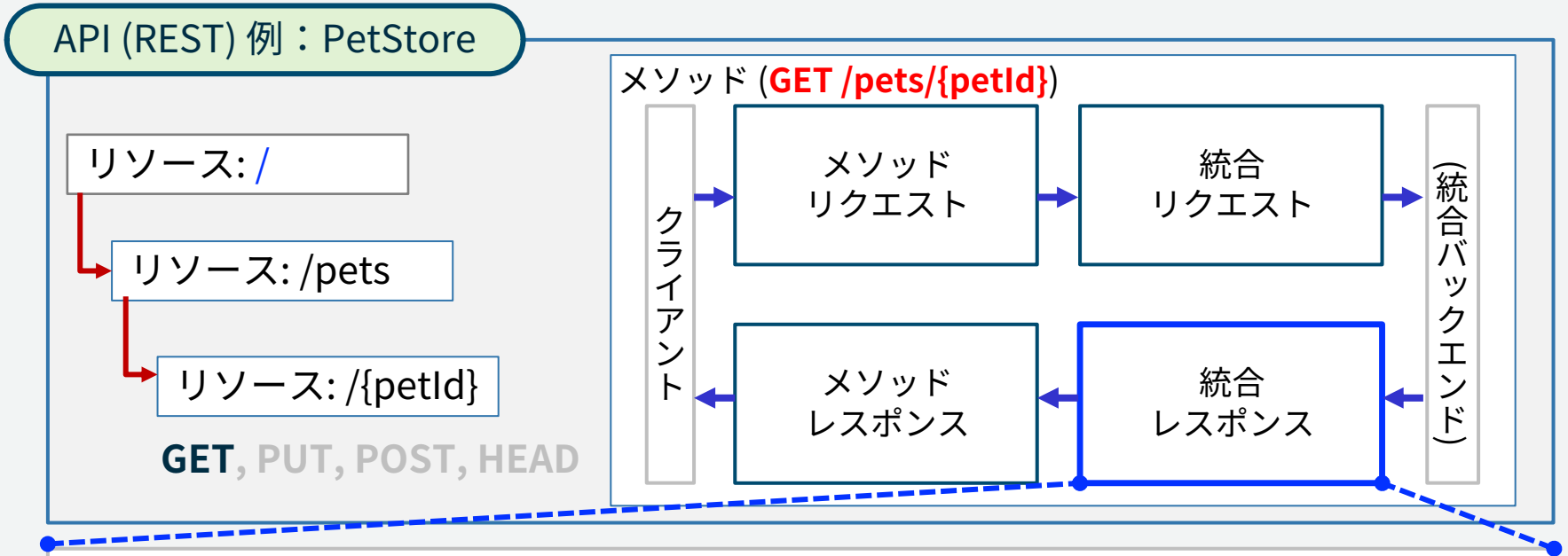
- バックエンドへのリクエストに関する設定
  - ルーティング先バックエンドの指定、リクエストの変換など
  - バックエンドの種別(=統合タイプ)として、「Lambda関数」「HTTP」「Mock」「AWSサービス」「VPCリンク」から選択 (後述)

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 (REST) - メソッド設定

REST



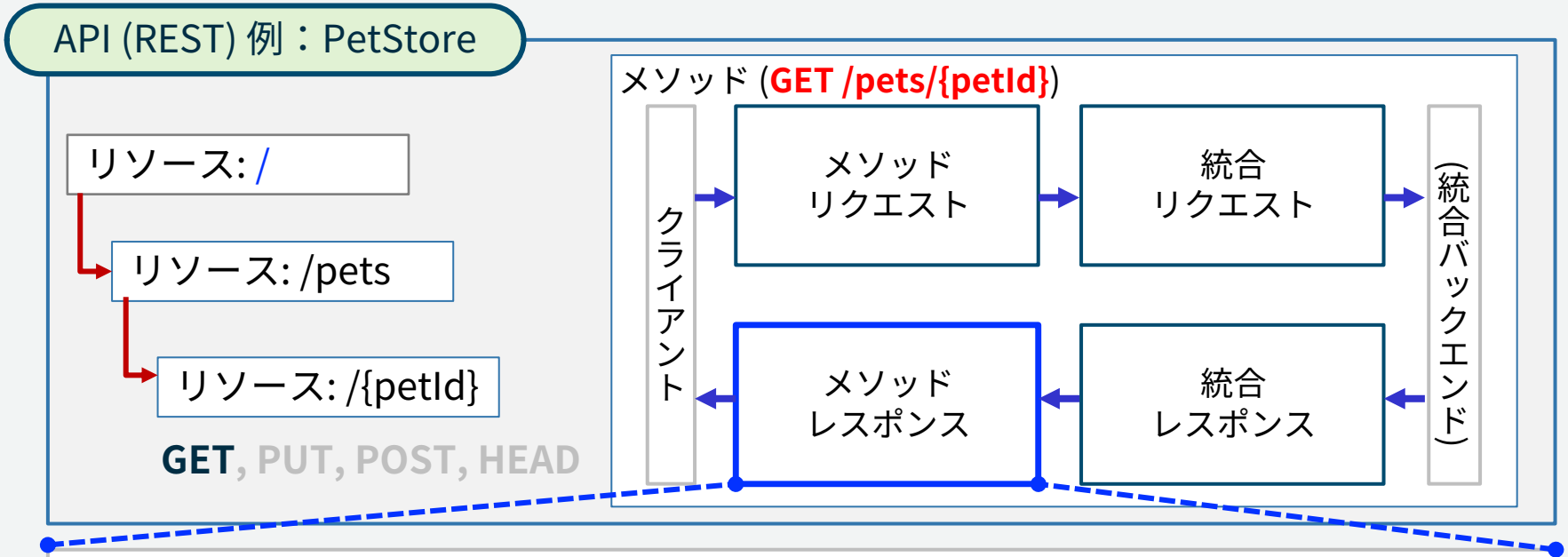
- バックエンドからのレスポンスに関する設定
  - レスポンス内容の変換やステータスコードのマッピングなど
  - (統合リクエストに「統合プロキシ」を指定した場合は設定不可)

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 (REST) - メソッド設定

REST



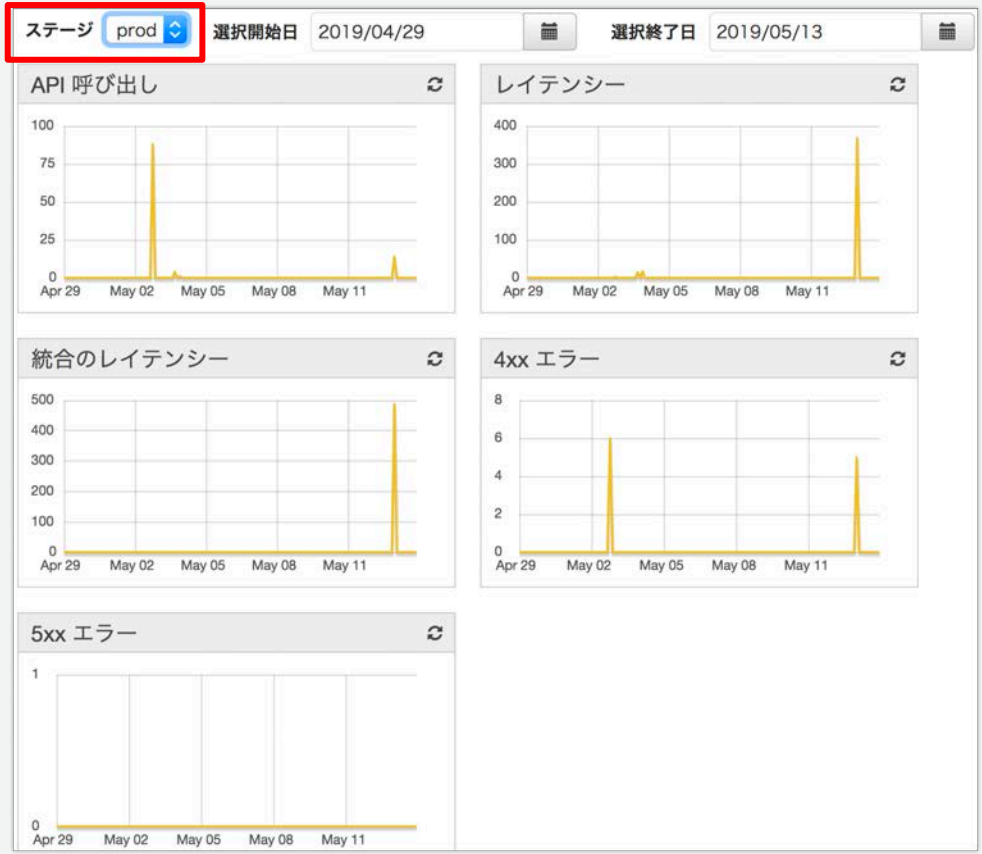
- リクエストに対する最終的なAPI Gatewayとしてのレスポンスに関する設定
  - どういうステータスコードやHTTPレスポンスヘッダを返すかなど

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API ダッシュボード (REST)

REST



- API呼び出し
  - そのAPIの呼び出し回数
- レイテンシー
  - リクエストを受け取ってから応答するまでの時間(ms)
- 統合のレイテンシー
  - バックエンドとのやり取りに要した時間(ms)
- 4xx エラー
  - HTTP応答 4xx を返した回数
- 5xx エラー
  - HTTP応答 5xx を返した回数

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

本日のアジェンダ

- はじめに - API / Web API とは
- Amazon API Gateway 概要
    - 全体イメージ
    - ユースケース
    - REST API / WebSocket API
    - その他主要な機能
    - 補足的機能 や ご使用料金の情報など
- まとめ

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

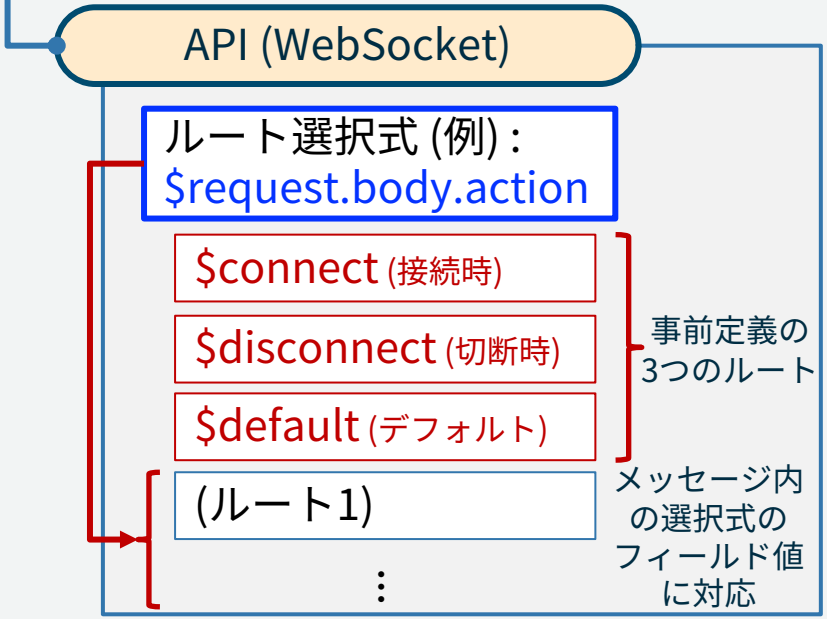
aws

API設定 (WebSocket) - ルート

WebSocket

WebSocketの状態に関するイベントを「ルート(Route)」として定義。
3つの事前定義ルートに加えてメッセージに応じたルートを指定可能

1. **WebSocket URL** (**wss**://<api-id>.execute-api.<region-id>.amazonaws.com/<stage-name>)
2. **Callback URL** (**https**://<api-id>.execute-api.<region-id>.amazonaws.com/<stage-name>/@connections/<connetion-id>)



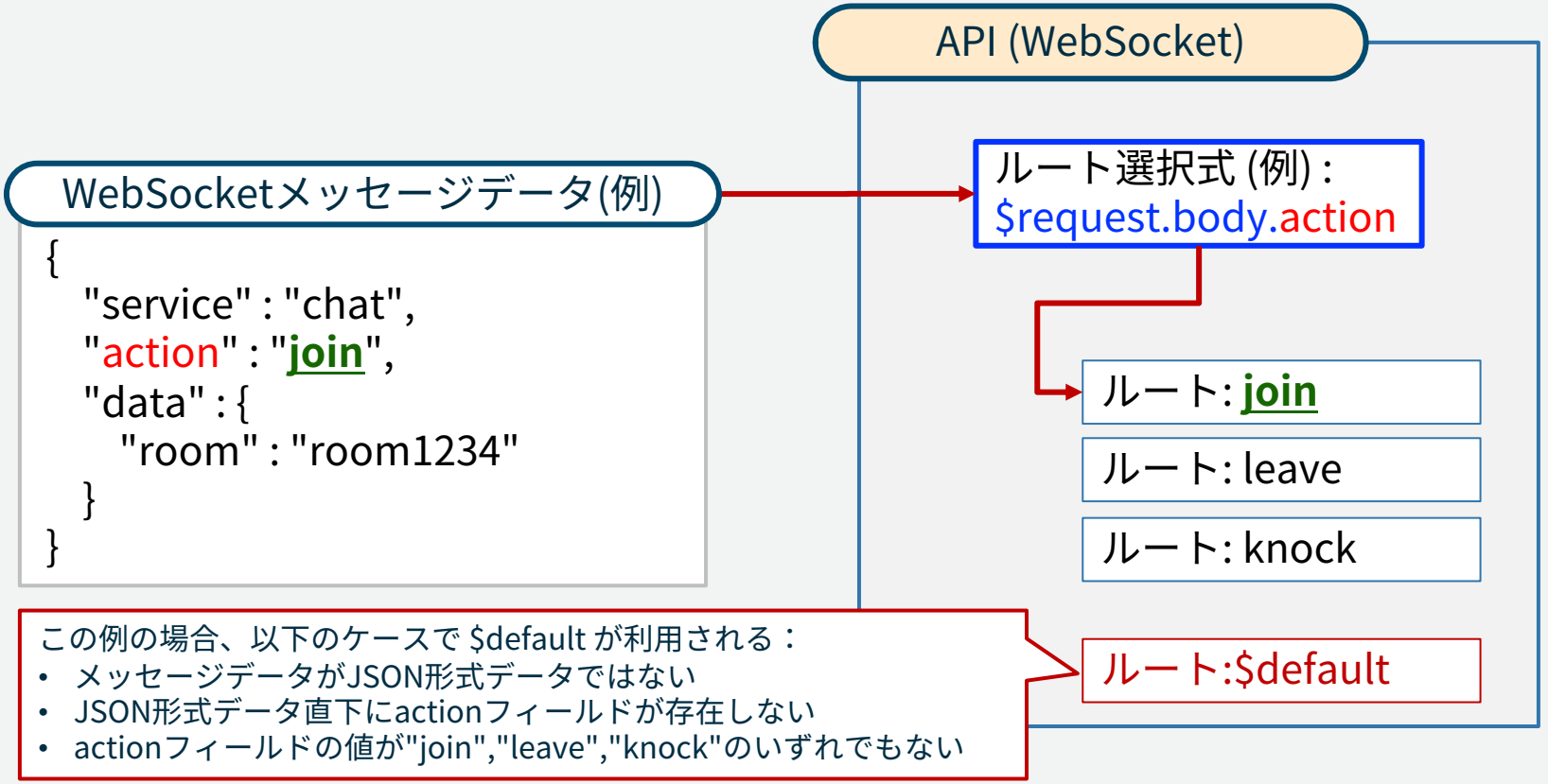
- 各APIには「英数小文字からなるID(api-id)」が割り振られ、URL(ドメイン名の一部)を構成
  - 接続とWebSocket用に2つのURLが提供
- 3つの「ルート」が事前定義
  - WebSocket接続・切断時の処理
  - ルート選択式で評価するデータが無かった場合 $default が利用(非JSONの場合含む)
- メッセージデータ(JSON)内の式が示すフィールドの値としてカスタムルートが定義されていた場合はそのルートが選択

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 (WebSocket) - ルート選択式

WebSocket

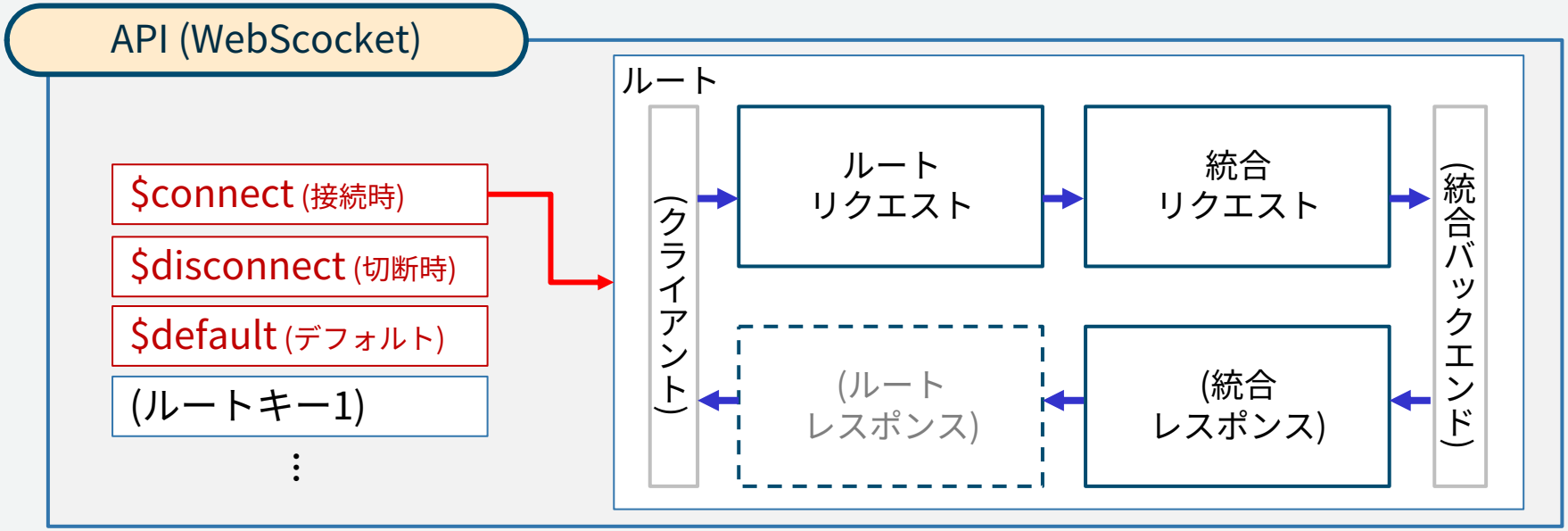


© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 (WebSocket) - ルート設定

WebSocket



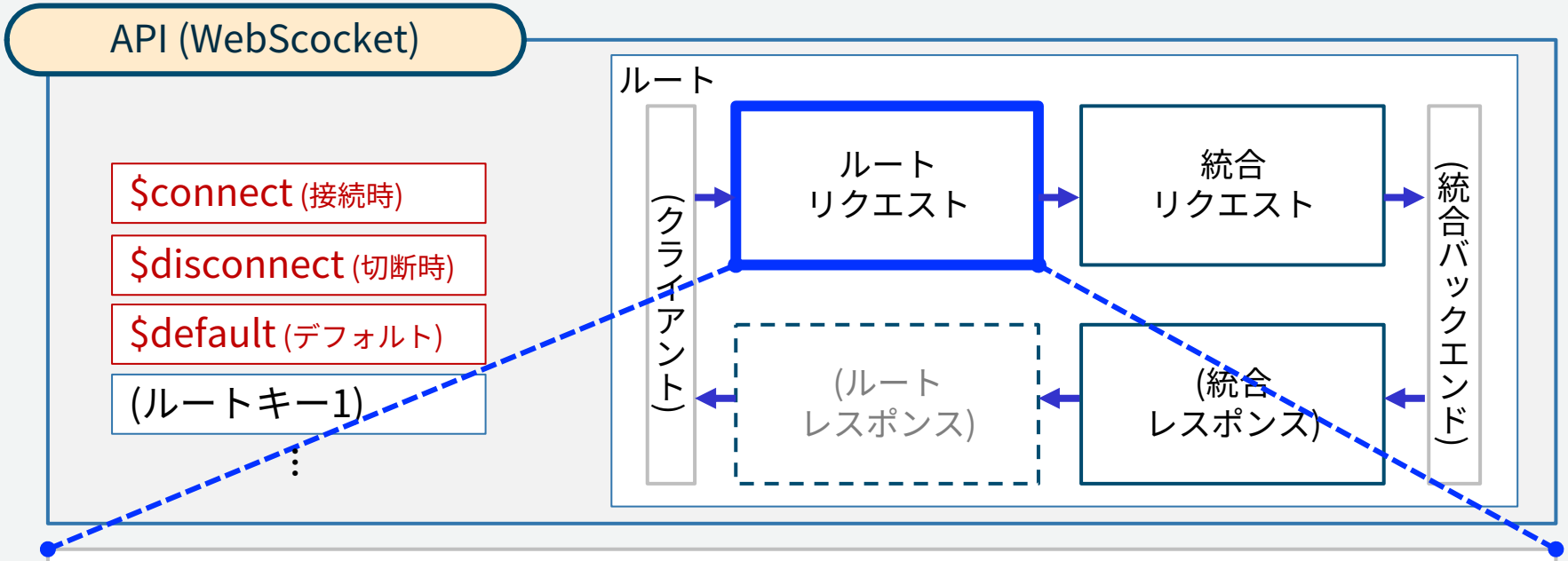
- ルート毎に、合計3つの設定箇所で1つのフロー処理を定義
  - ルートリクエスト→統合リクエスト → 統合レスポンス→(ルートレスポンス)
  - 「統合レスポンス」の設定はオプション
  - 「ルートレスポンス」の設定はオプション

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 (WebSocket) - ルート設定

WebSocket



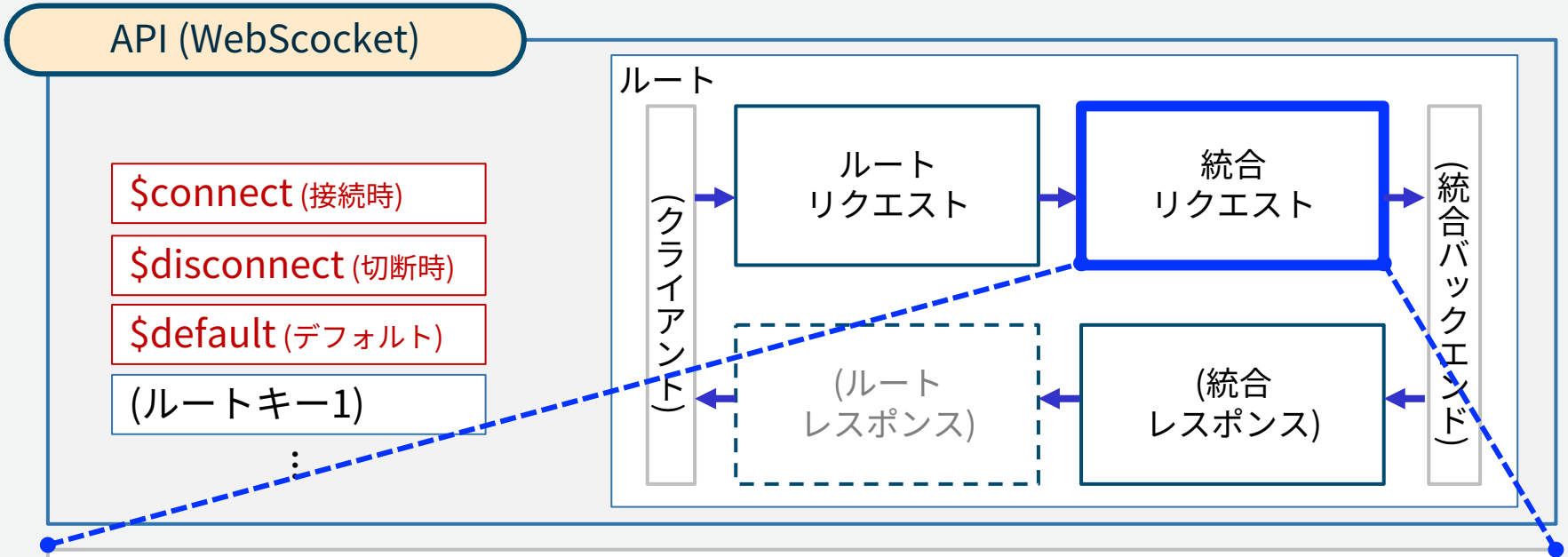
- $connect(接続時)ルートの場合　：認証の有無、APIキー必須、
- $disconnect(切断時)ルートの場合　：設定項目なし
- それ以外のルートの場合　：「モデル選択式」を指定(オプション)
    - モデル選択式は、当ルートでの受信メッセージのモデル(後述)を指定

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 (WebSocket) - ルート設定

WebSocket



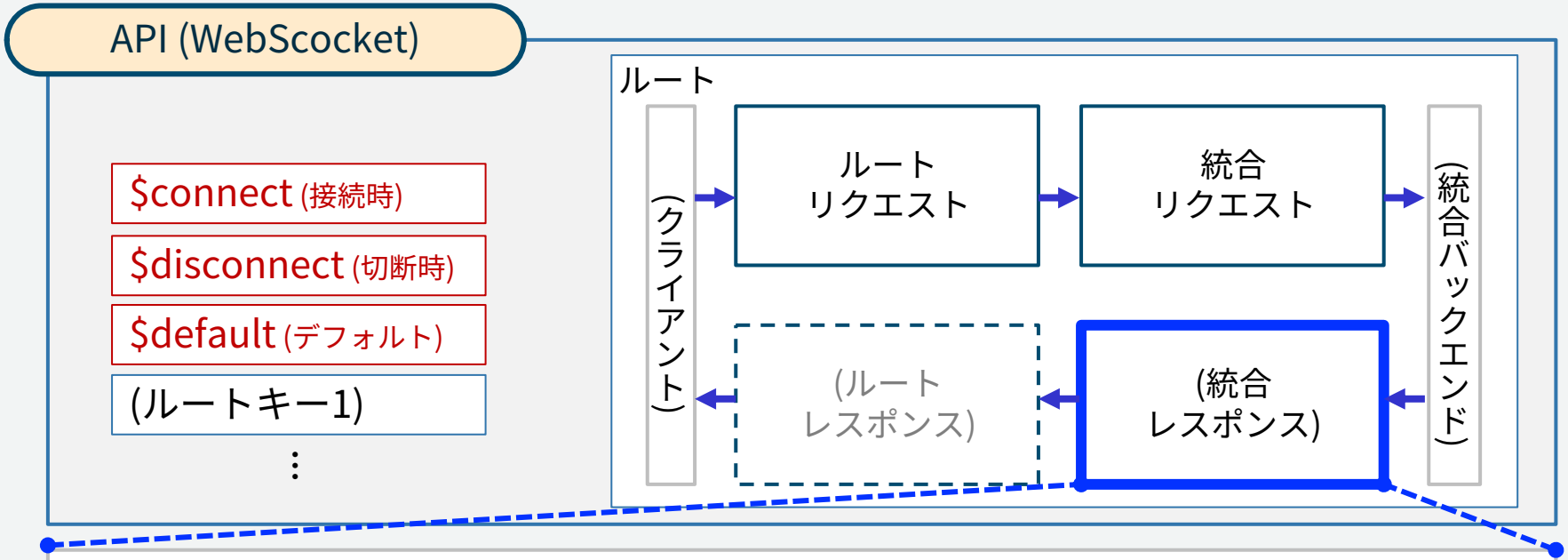
- バックエンドへのリクエストに関する設定
  - バックエンドの種別(=統合タイプ)として、「Lambda関数」「HTTP」「Mock」「AWSサービス」「VPCリンク」から選択 (REST APIと同様)
  - オプションとしてテンプレート選択式を指定可能

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 (WebSocket) - ルート設定

WebSocket



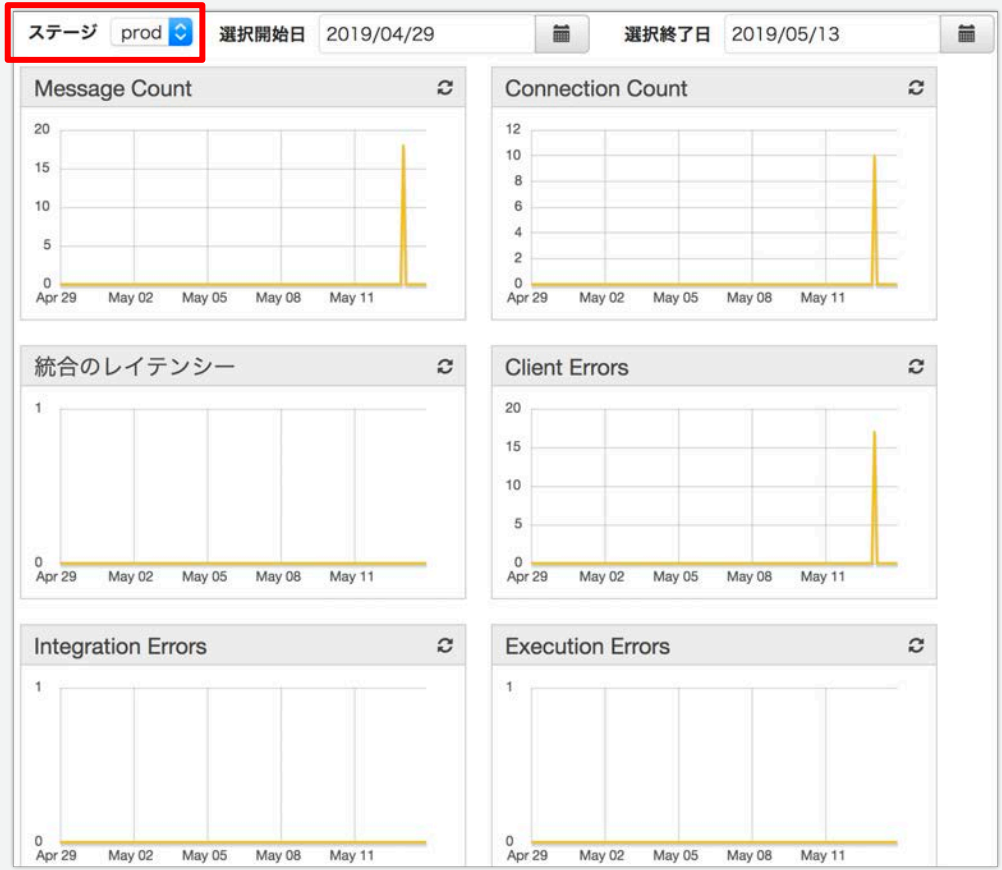
- 「統合レスポンス」の設定はオプション
  - 「レスポンス選択式」を指定
  - その上で、“「レスポンスキー」とそれに対応する「テンプレート選択式」および「テンプレートキー」と「テンプレート」 のセット”を複数指定可能

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API ダッシュボード (WebSocket)

WebSocket



- Message Count
  - 送受信メッセージ数
- Connection Count
  - 接続($connect)メッセージ数
- 統合のレイテンシー
  - バックエンドとのやり取りに要した時間
- Client Errors
  - バックエンド呼び出し前に4xx応答を返したリクエストの数
- Integration Errors
  - 統合から4xx/5xx応答を返したリクエストの数
- Execution Errors
  - 統合呼び出し時のエラー数

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

本日のアジェンダ

- はじめに - API / Web API とは
- Amazon API Gateway 概要
    - 全体イメージ
    - ユースケース
    - REST API / WebSocket API
    - その他主要な機能
    - 補足的機能 や ご使用料金の情報など
- まとめ

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 - 認証・認可

REST　WebSocket

メソッドリクエスト
「認証」設定

$connect のルートリクエスト
「認証」設定

1 IAM
アクセス権限

- AWS 署名 v4による認証
- IAMユーザーのクレデンシャル情報を利用してリクエストに署名データを指定
- CognitoフェデレーテッドID認証でIAMロールのアクセス権を使う場合もこの認証方式を選択

2 Lambda
オーソライザー

- 認証処理を委譲するLambda関数を指定してAPI へのアクセスを制御(認証成功時にプリンシパルIDとそのポリシードキュメントを返す仕様)
- 独自の認証方式やOAuthなどの認証方式を組み込む方法として利用
- (以前の名称は「カスタムオーソライザー」)

3 Cognito
オーソライザー

- 指定したCognitoユーザープールを元に事前に認証
- Cognito認証を行い取得したトークン (IDトークン または アクセストークン)を、Authorization ヘッダーに指定してAPI Gatewayへリクエストする

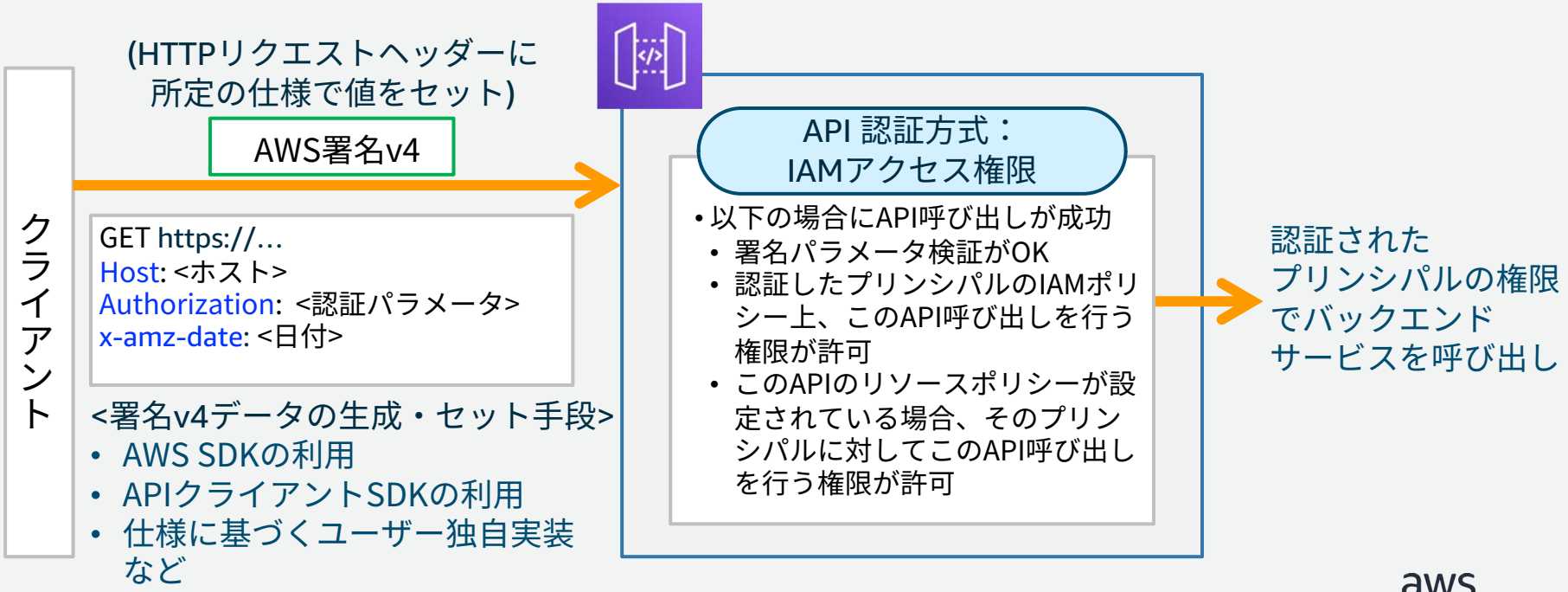
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 - 認証・認可(IAMアクセス権限)

REST WebSocket

APIクライアントが認証対象のIAM ユーザーの
アクセスキー等から生成したAWS署名v4を送信することを要求する方式

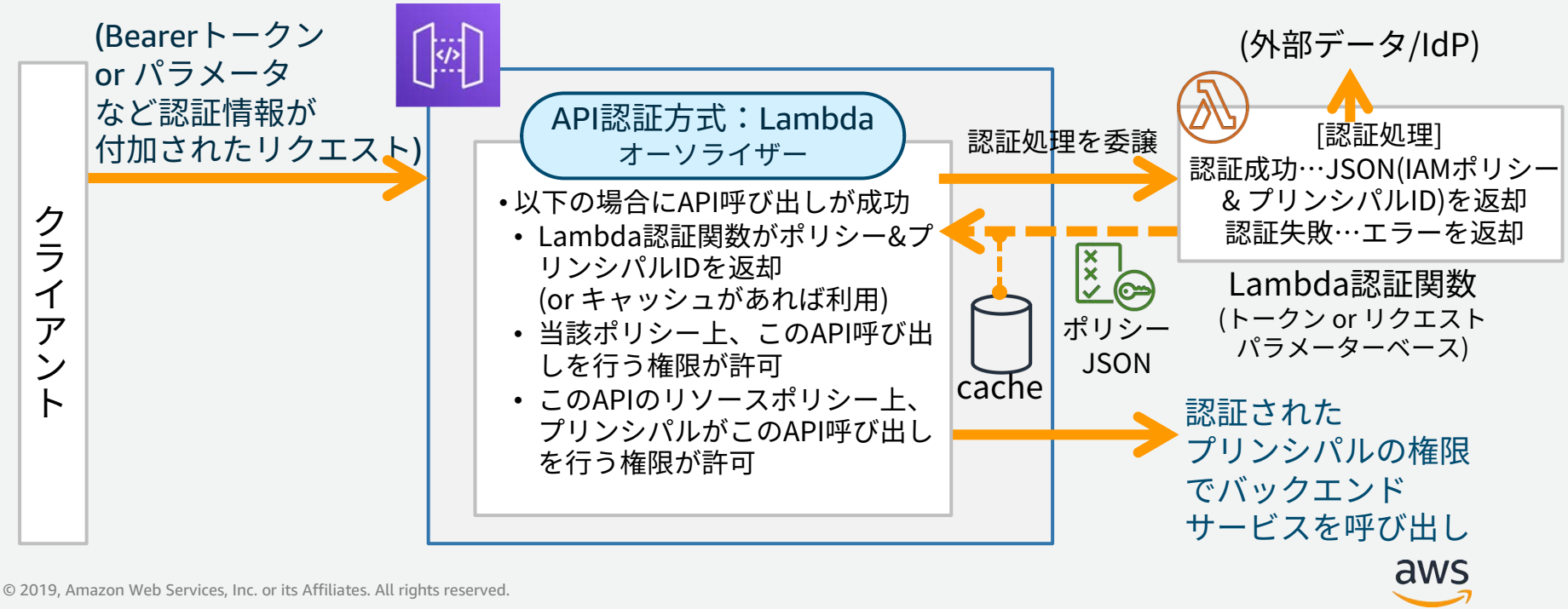


© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

API設定 - 認証・認可(Lambdaオーソライザー)

REST WebSocket

所定の入出力仕様を満たすよう実装されたLamda関数を用いる方式
(リクエストデータを元に認証し、認証結果としてIAMポリシーJSONを返す)

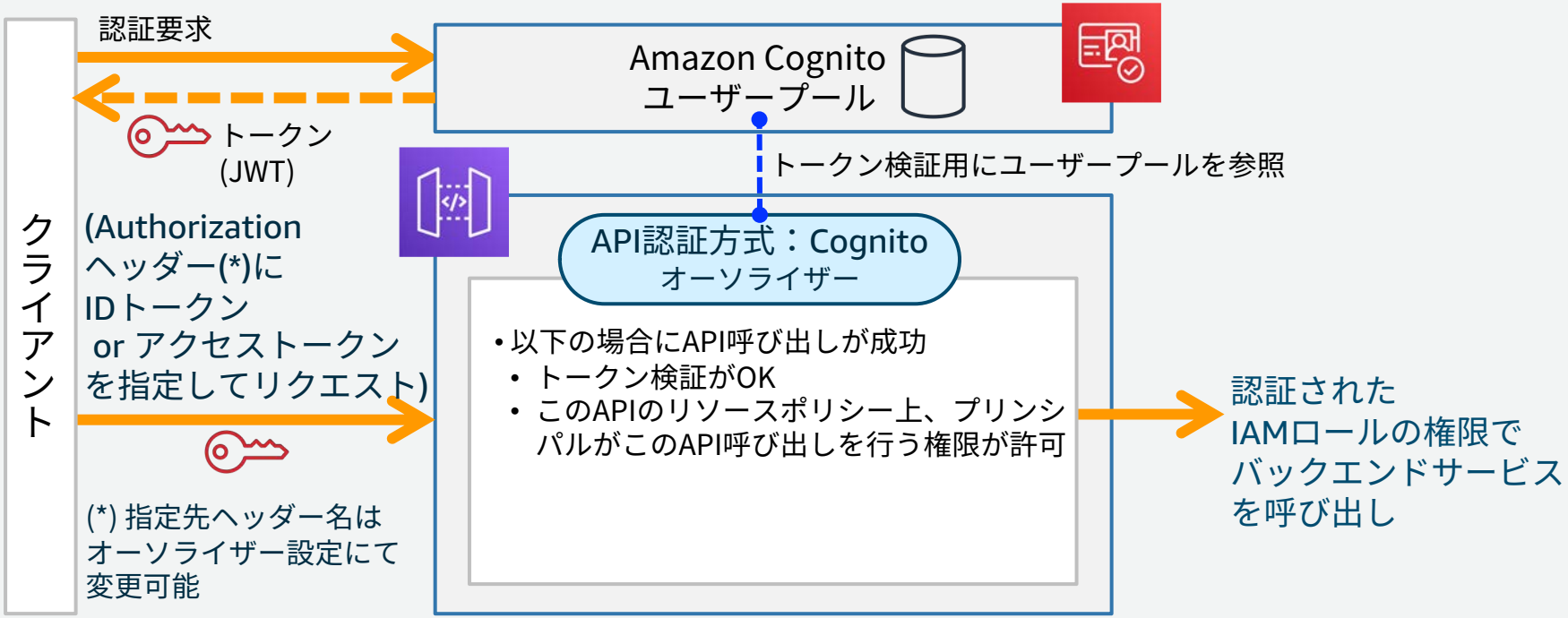


© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

API設定 - 認証・認可(Cognitoオーソライザー)

REST

事前にAmazon Cognitoユーザープールで認証を行い、
取得したトークンをHTTPヘッダーに指定してAPIに渡すことを要求する方式



© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 - 統合タイプ

REST WebSocket

統合リクエスト
統合タイプ

※REST APIの主な制限(ハードリミット)：
- 統合タイムアウト … 最長29秒
- HTTPヘッダー値のサイズ … 最大10,240 Bytes
- ペイロードサイズ … 最大10 MB

※WebSocket APIの主な制限(ハードリミット)：
- フレームサイズ … 最大32KB
- 接続時間 … 最大2時間
- アイドル接続タイムアウト … 最大10分
- ペイロードサイズ … 最大128KB

1 Lambda関数
- Lambda関数を指定して呼び出す
- クロスリージョン、クロスアカウントでの呼び出しに対応

2 HTTP
- インターネット経由で呼び出し可能なURL(HTTPS)とメソッドを指定して呼び出す
- Publicに到達可能なエンドポイントであれば実装場所は問わない

3 Mock
- モックとしてAPI Gatewayで直接、固定的な応答を返す
- 統合レスポンスのマッピングテンプレートで実際のモック応答データを設定

4 AWSサービス
- AWSのサービスを直接呼び出して、各サービスで用意されているアクションを実行
- 例：Kinesis Data Streamのエンドポイントを独自Web APIとして対外的に公開

5 VPCリンク
- Network Load Balancer(NLB)を参照する「VPCリンク」を登録しそれを経由して VPC内リソースへアクセス。エンドポイントURLとしてNLBホスト名を指定

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

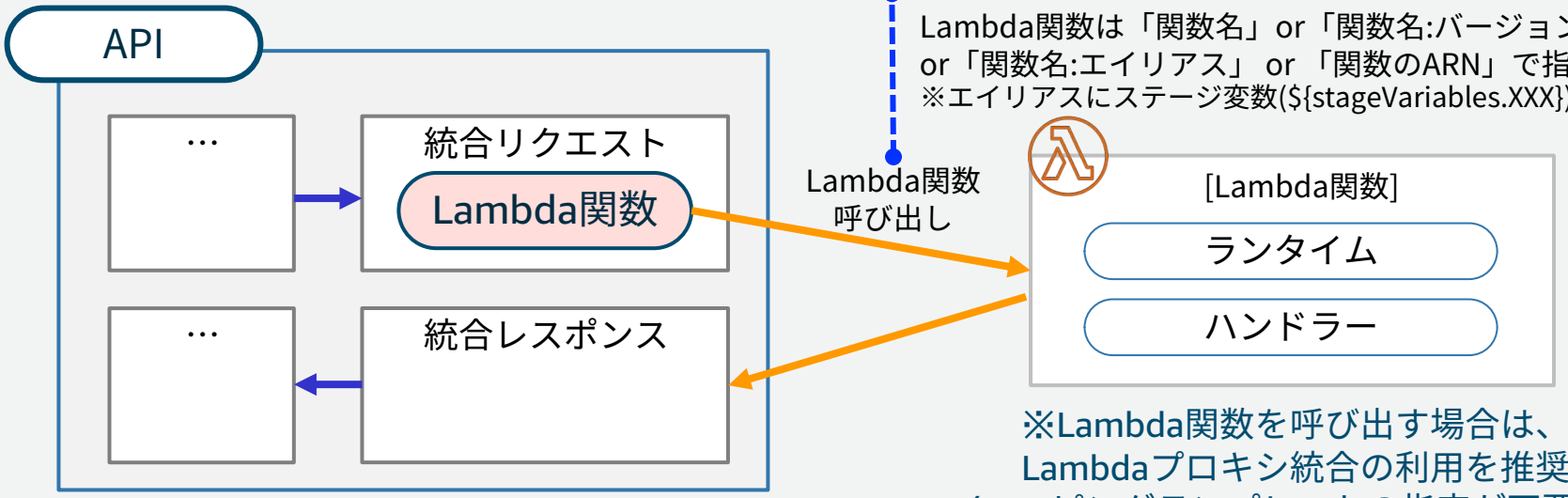
API設定 - 統合タイプ(Lambda関数)

REST WebSocket

Lambda関数をバックエンドとして呼び出すタイプ (別リージョン・別アカウントの関数を指定可能)

※対象Lambda関数に対する呼び出し権限を API Gatewayサービス に与える必要あり(管理コンソールから設定を行うと自動的に追加可能)

Lambda関数は「関数名」or「関数名:バージョン」 or「関数名:エイリアス」 or 「関数のARN」で指定
※エイリアスにステージ変数(${stageVariables.XXX})を指定可



※Lambda関数を呼び出す場合は、Lambdaプロキシ統合の利用を推奨 (マッピングテンプレートの指定が不要に)

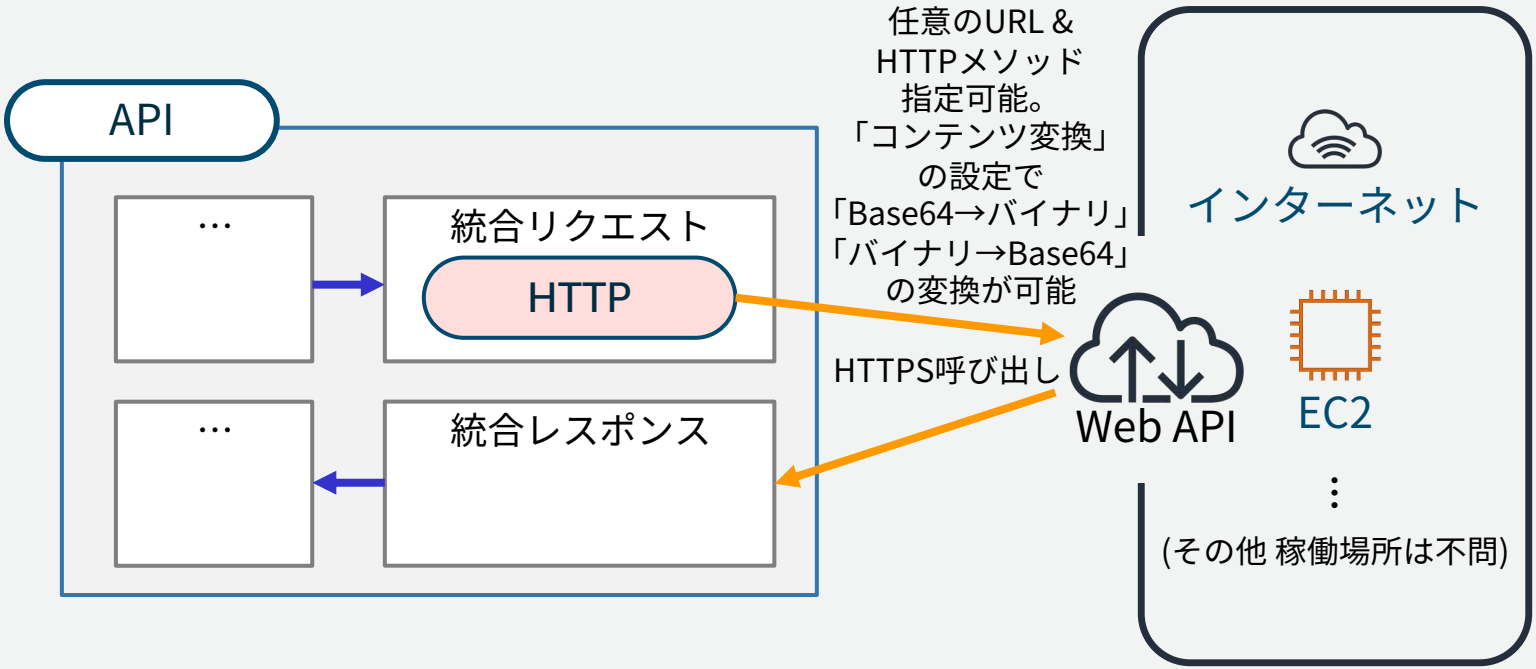
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 - 統合タイプ(HTTP)

REST WebSocket

任意のHTTPエンドポイントURLをバックエンドとして呼び出すタイプ
(Publicに到達可能であれば、インターネットやEC2等を問わない)



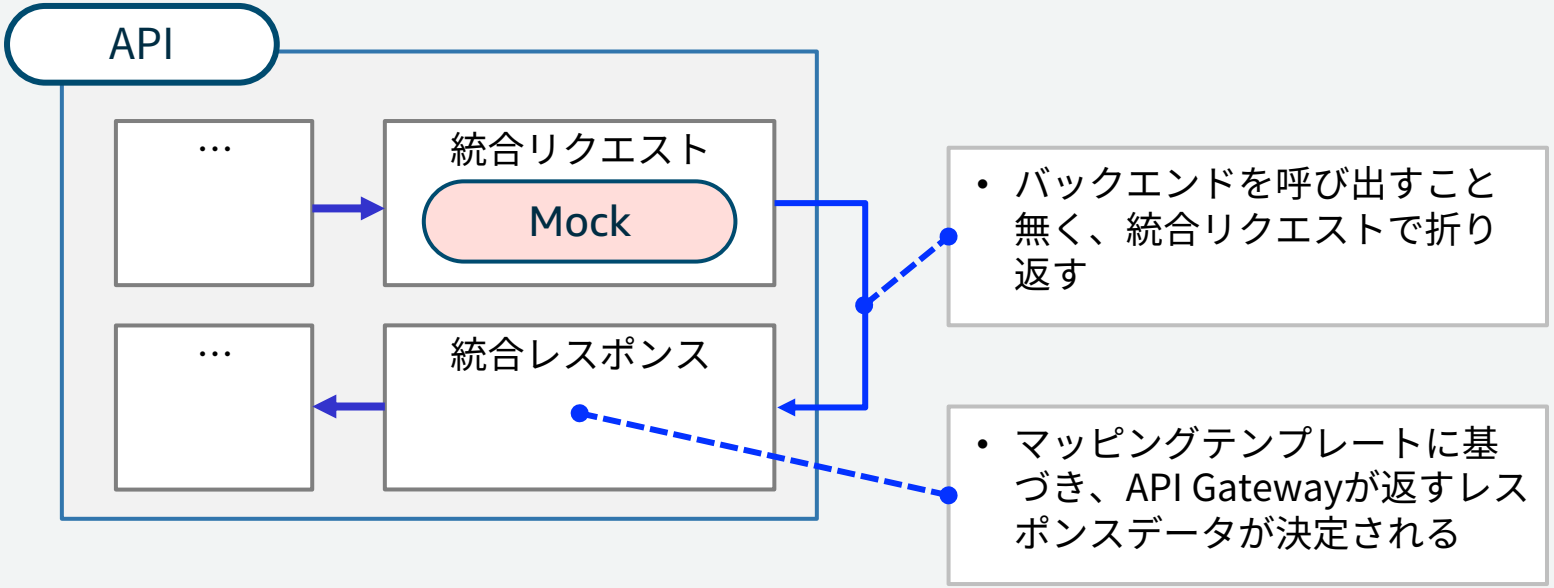
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 - 統合タイプ(Mock)

REST WebSocket

モックデータ(マッピングテンプレートで指定)を返すタイプ
テスト実装やAPIクライアントの先行実装のためのモック実装が主な用途

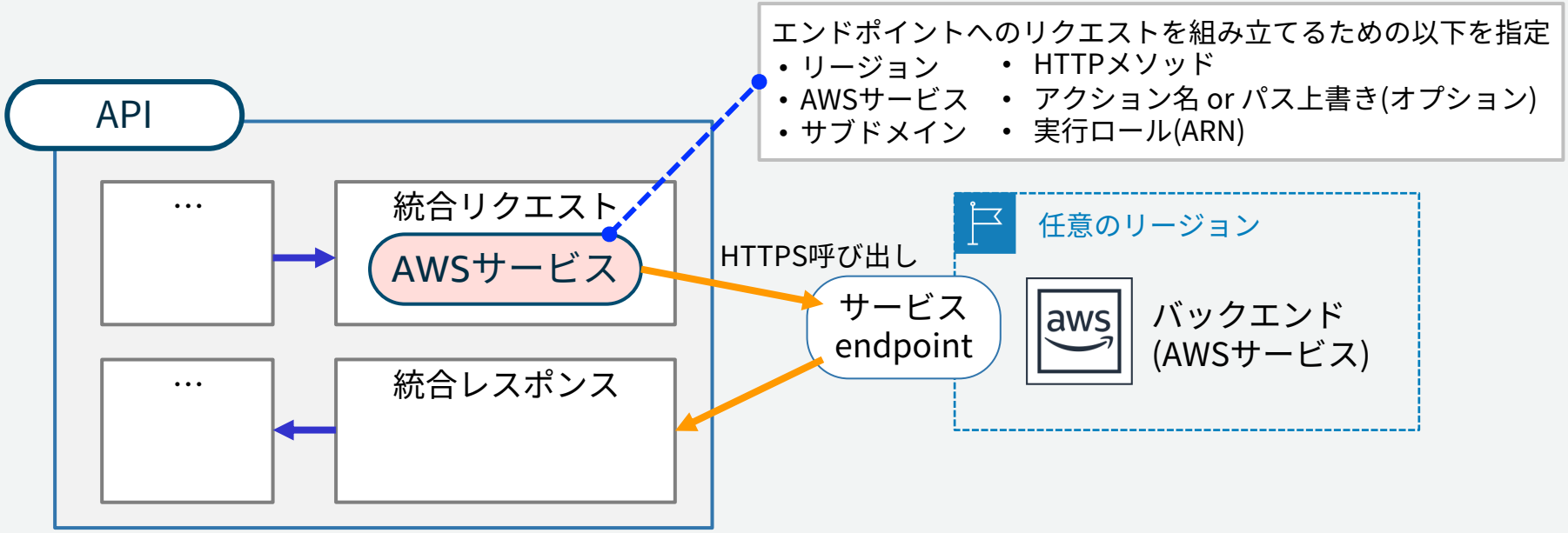


© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 - 統合タイプ(AWSサービス)

バックエンドとしてAWSサービスを指定 (実行ロールのARNを指定可能)
AWS サービスAPIを独自仕様で提供する用途で利用

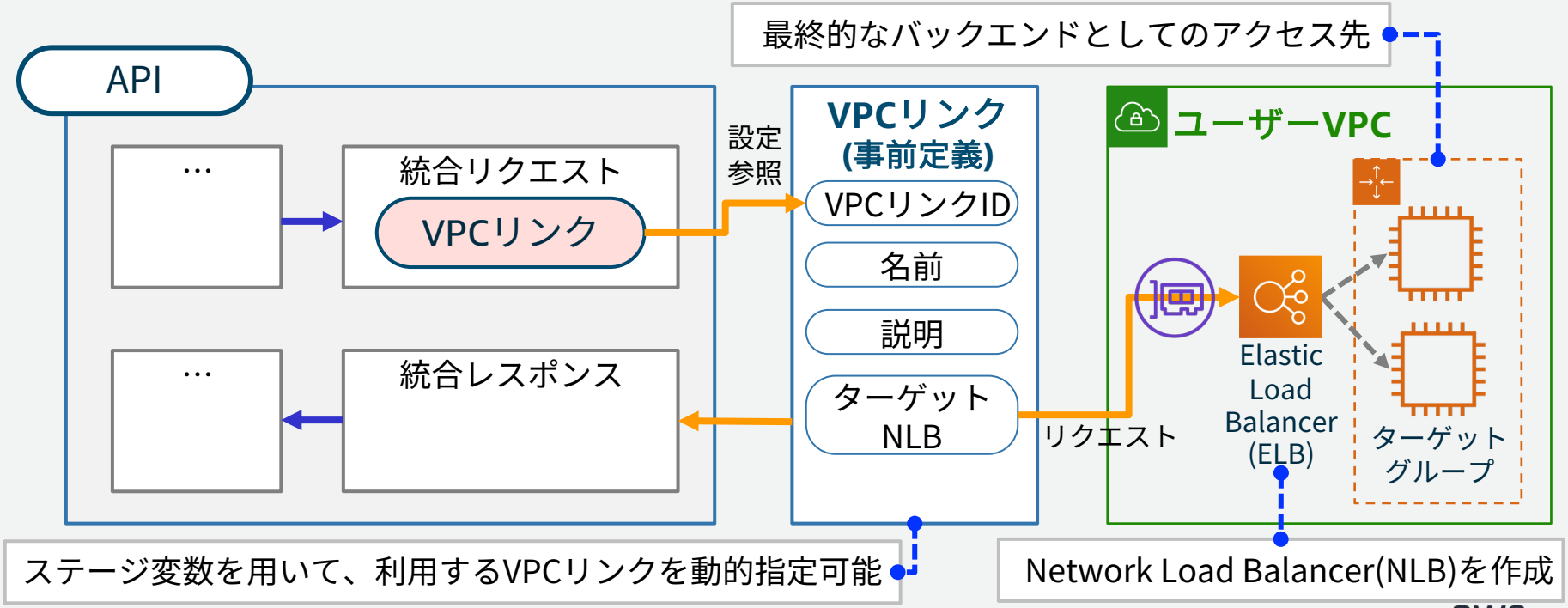


© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

API設定 - 統合タイプ(VPCリンク)

REST WebSocket

同一リージョンのVPC内リソースをバックエンドとしてアクセスする形式
(「VPCリンク」を事前定義し、NLBと関連付けてそれを参照)

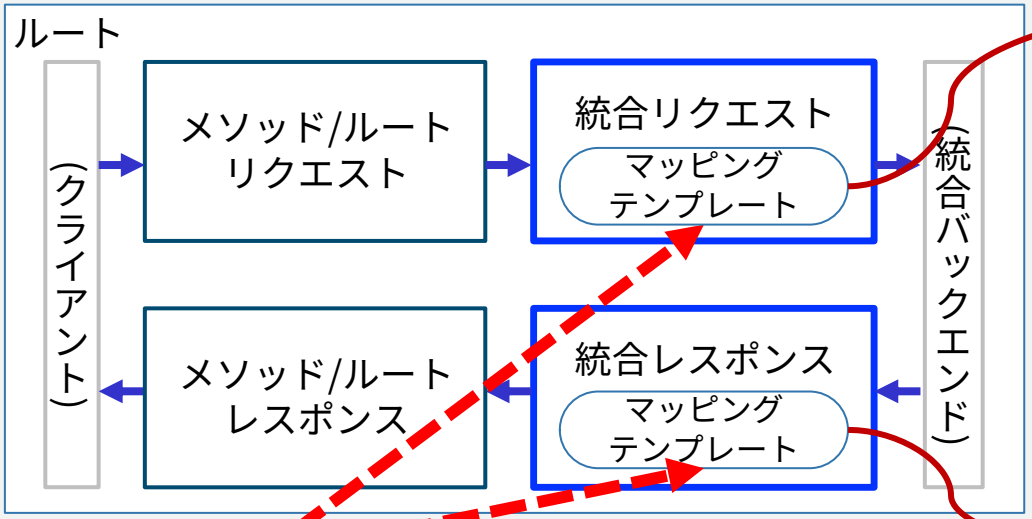


© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 - リクエスト/レスポンス変換

REST WebSocket



例：XMLをJSONに変換 ($input.json関数を利用)

```
{
   "body" : $input.json('$')
}
```

例：JSONから一部フィールドを抜き出しXMLに変換

```
#set($root = $input.path('$'))
<xml>
  <name>
    $root.name
  </name>
</xml>
```

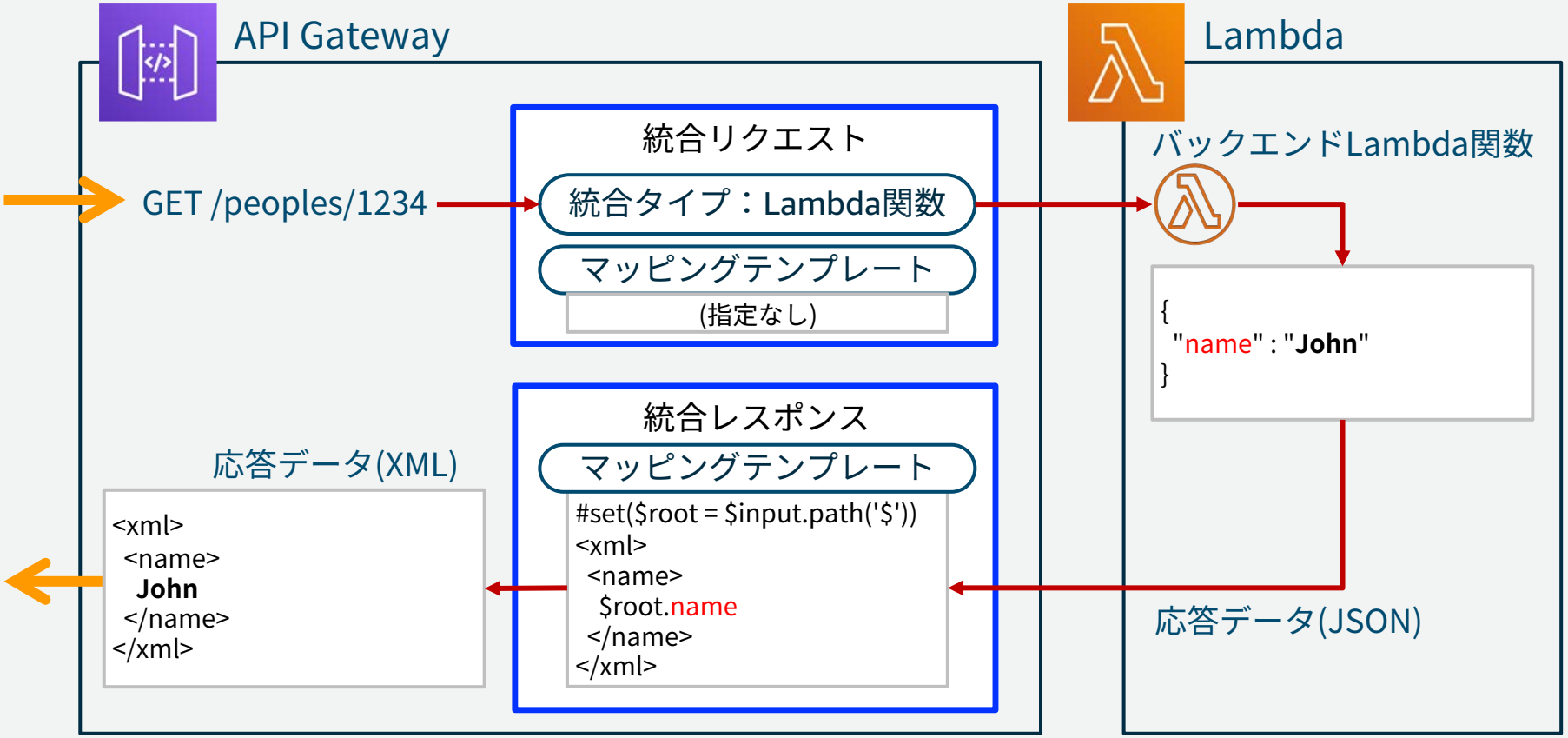
マッピングテンプレート

- 統合リクエスト/統合レスポンスにて指定するテンプレートドキュメント
  - VTL(Velocity Template Language)とJSONPathや予約変数($input等)を利用し記述
  - リクエストやレスポンスのデータを別の形式に変換することが可能
- 用途：不要フィールドの削除、XML⇔JSON変換、Mock統合タイプでのデータ指定、…

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 - リクエスト/レスポンス変換(REST APIでの例)

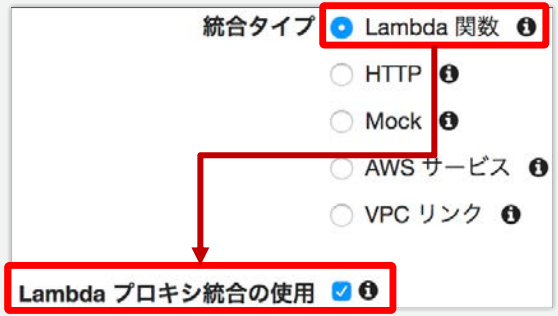


© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

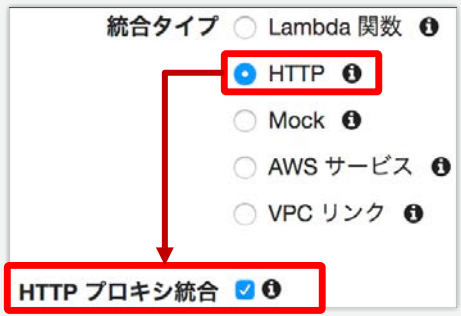
API設定 - 統合タイプ (プロキシ統合)

REST　WebSocket

Lambda関数



HTTP



VPCリンク

「統合レスポンス」でマッピングテンプレートは指定不可 &「統合レスポンス」は全体が設定対象外に

- Lambda関数は以下を含む構造でJSONデータを返す必要あり

```
{
  "statusCode" : <数値>,
  "headers" : { "ヘッダ名" : "値", … },
  "body" : "<応答ボディ文字列>"
}
```

- バックエンドから返されたHTTP応答データをそのまま統合レスポンスのデータとして返す

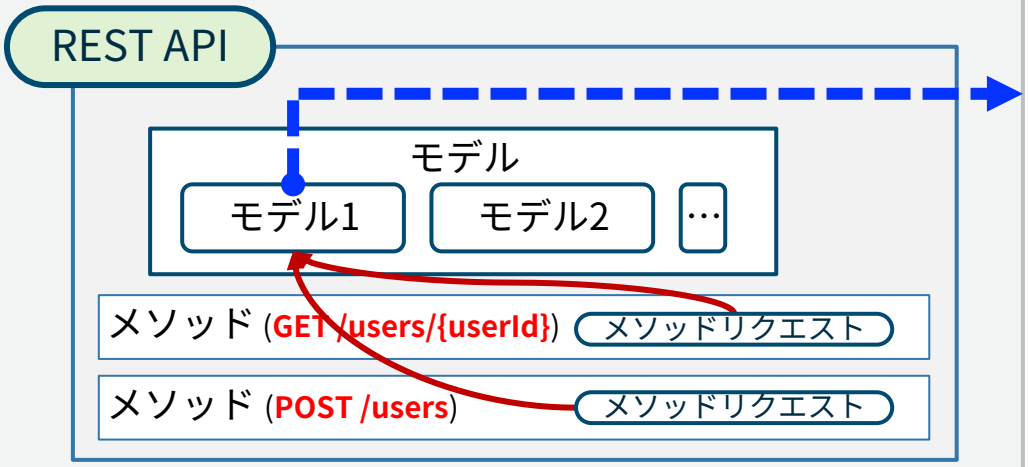
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 - モデル

REST WebSocket

- リクエスト/レスポンスで扱われるデータ構造(スキーマ)を
  JSON Schema draft 4形式で定義したもの (API Gateway利用上、必須ではない)
  - 1つのAPI内で複数のメソッド/ルートを跨がって共通的に利用可能
- リクエスト検証やクライアントSDK生成、マッピングテンプレートで活用



モデル定義例

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "User",
  "type": "object",
  "properties": {
    "UserID": { "type": "string" },
    "Name": { "type": "string" },
    "Age": {
      "description": "Age in years",
      "type": "integer",
      "minimum": 18
    }
  },
  "required": ["UserID", "Name"]
}
```

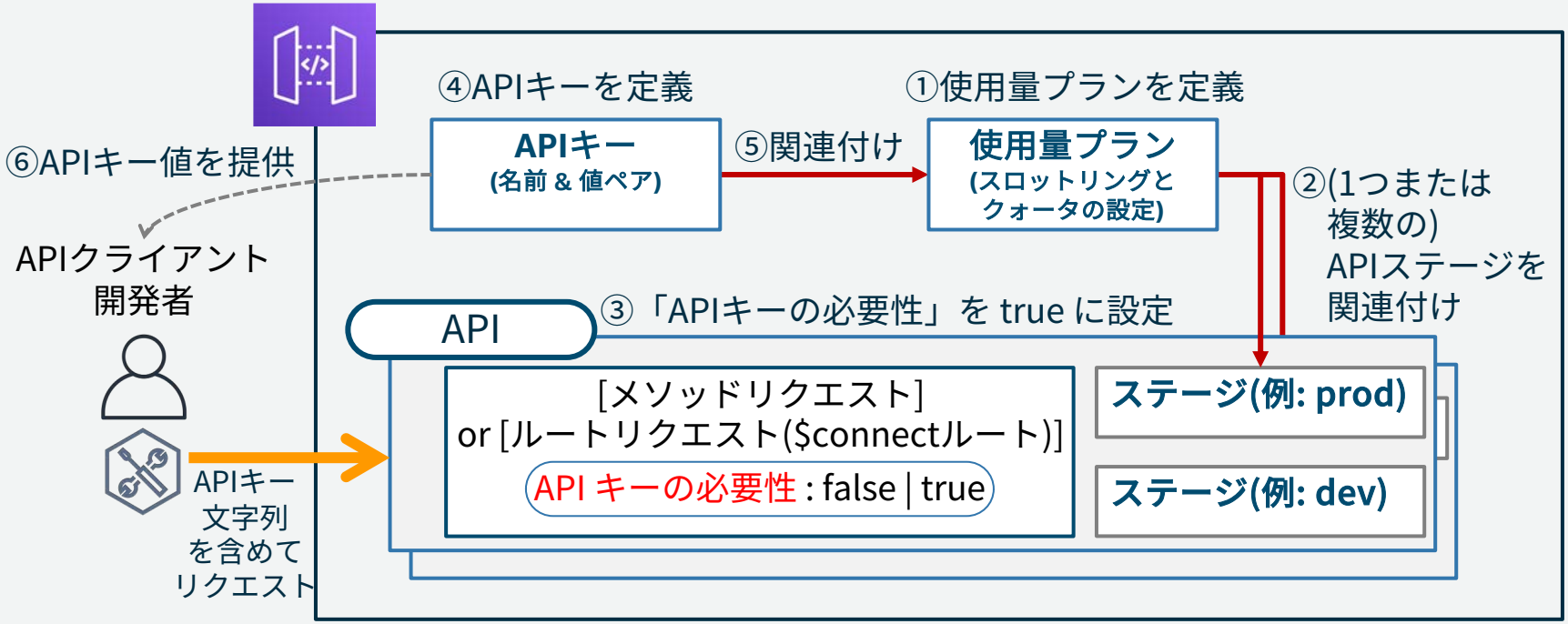
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

APIキーと使用量プラン

REST WebSocket

APIキーと使用量プランにより、APIのメータリングとスロットリングを実現



© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

APIキーと使用量プラン の補足

REST WebSocket

しきい値を超えた場合のリトライやAPIキーの用途にご注意

- 使用量プランについて
  - 閾値を超えるとHTTPステータスコード429(Too Many Requests)エラー
    - リトライ実装はAPIクライアント開発者の責務 (またはクライアントSDKを利用)
  - クォータ(呼び出し回数制限)の設定が可能：(日/週/月 あたりの要求回数)
- APIキーについて
  - APIキーを「認証」目的に利用しないこと
    - 計測のための識別用途のため
  - API側の設定項目(APIキーの必要性)はデフォルト値がfalse
    - メータリング・スロットリングを有効にする際には原則としてtrueを指定
      - 所定のHTTPリクエストヘッダー(x-api-key)またはLambdaオーソライザー(usageIdentifierKey)で指定されない場合 403 Forbidden エラー

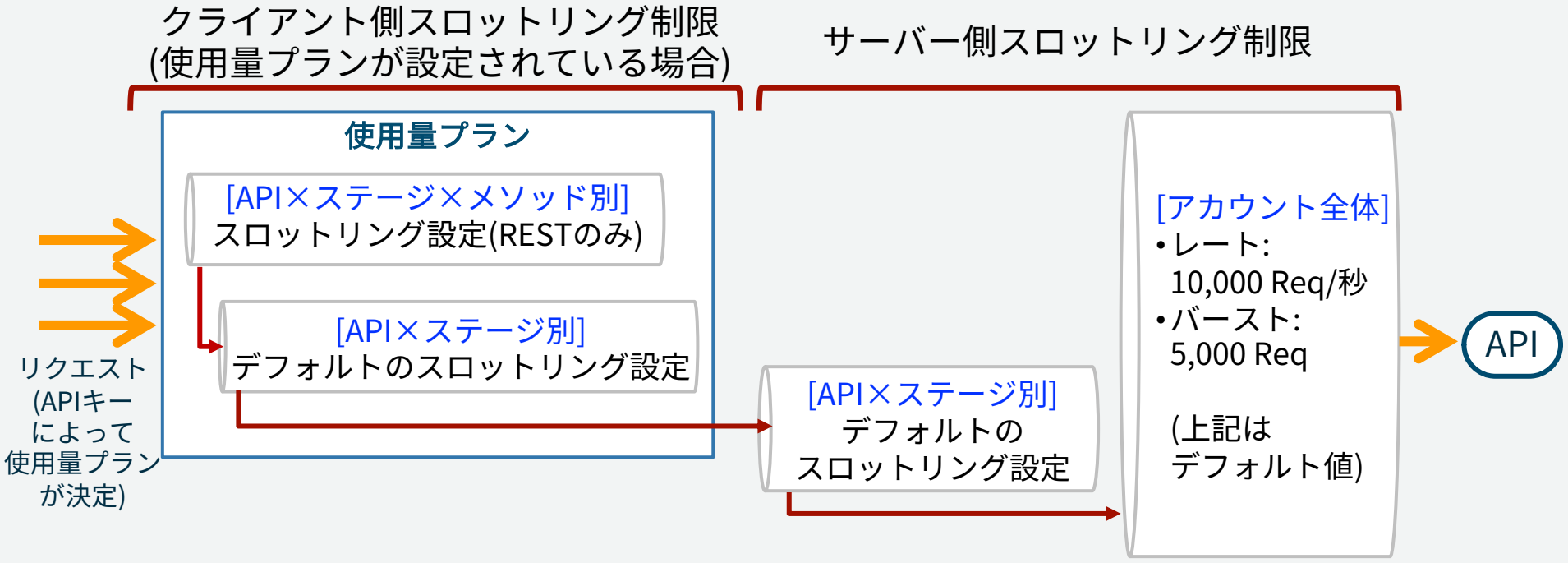
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

スロットリング

REST WebSocket

スロットリングは 「サーバー側のスロットリング制限」と
「クライアント側のスロットリング制限」によって行われる

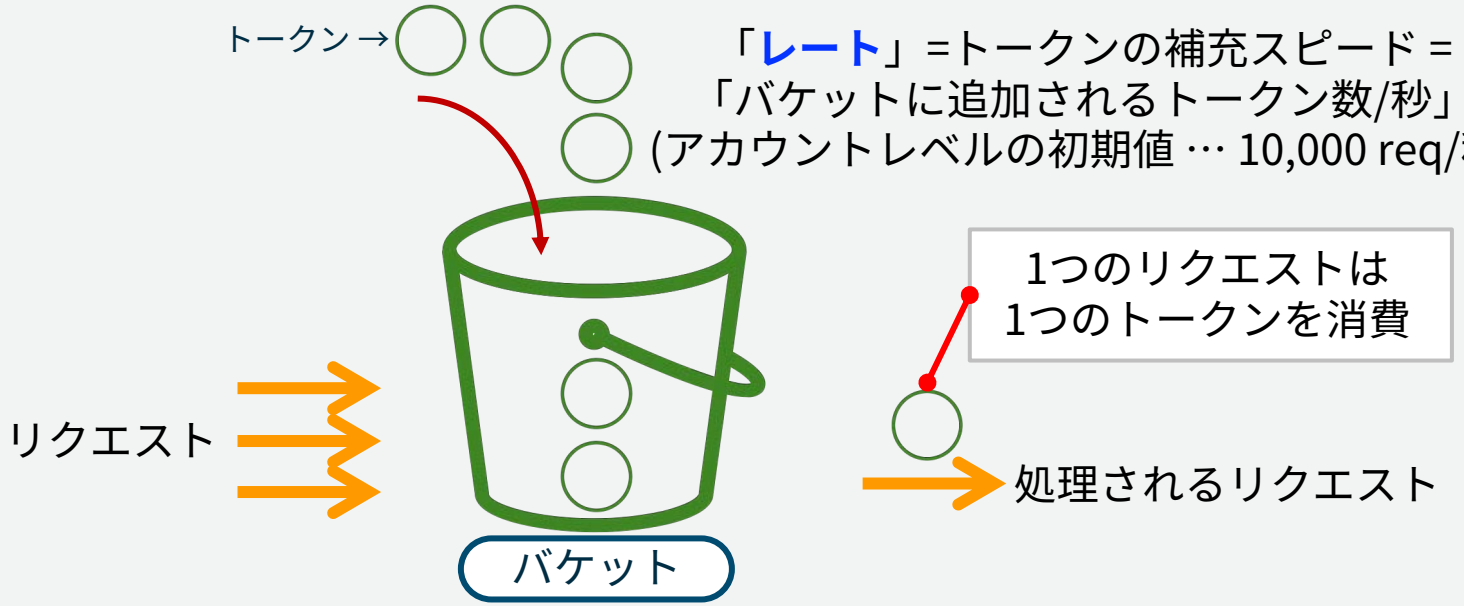


© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

(補足)：スロットリング - トークンバケットアルゴリズム

スロットリングはトークンバケットアルゴリズムに基づく
「レート」と「バースト」の2つの設定値に従って行われる



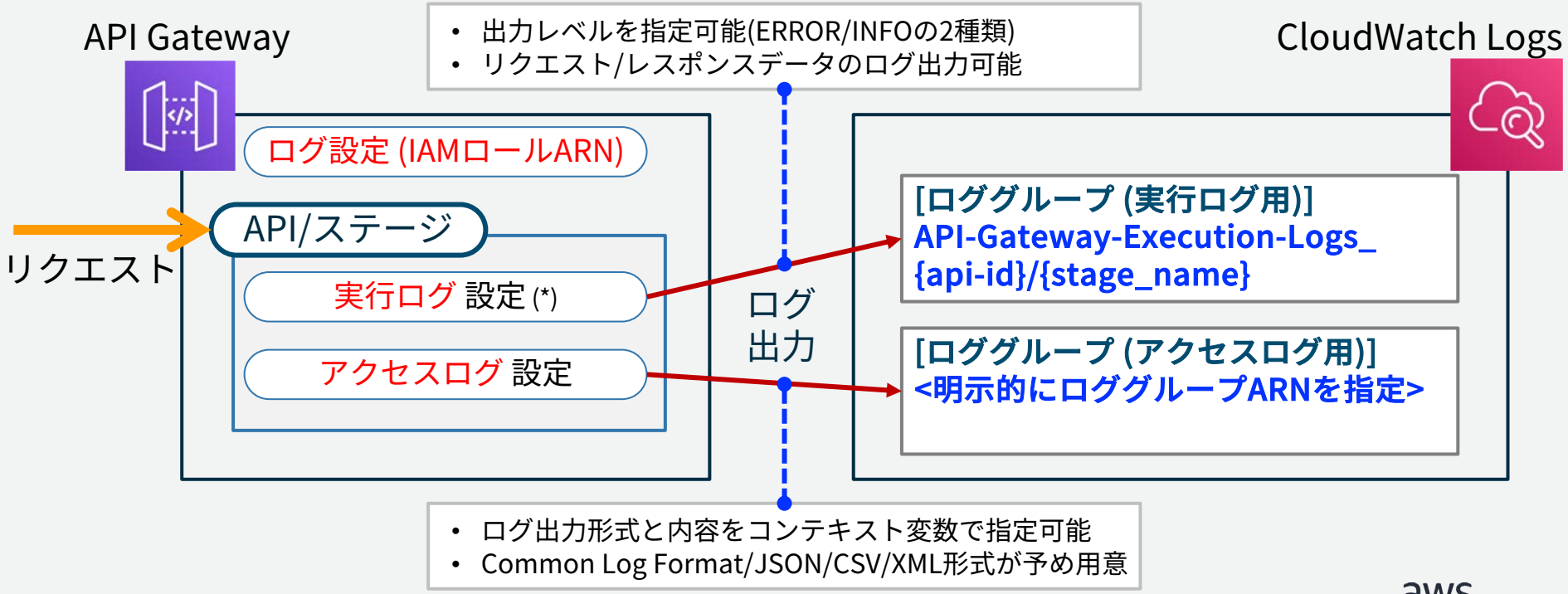
「**バースト(バースト上限)**」= バケット内のトークンの初期値 兼 バケット最大サイズ
(アカウントレベルの初期値 … 5,000)

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

API設定 - ログとモニタリング

(*)REST APIの場合、メソッドレベルでステージ設定を個別に上書き可能

API/ステージ単位で「実行ログ」と「アクセスログ」をCloudWatch Logsに出力可能



© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

API設定 - ログとモニタリング

REST WebSocket

API/ステージ単位でメトリクスを CloudWatch メトリクスに出力可能

メトリクス	意味	単位
4XXError	ステータスコード4XXのエラー数	個
5XXError	ステータスコード5XXのエラー数	個
CacheHit Count	キャッシュにヒットしたリクエスト数	個
CacheMiss Count	キャッシュにヒットせず、バックエンドからレスポンスされたリクエスト数	個
Count	API メソッド呼び出し数（リクエスト数）	個
Integration Latency	API Gateway がバックエンドにリクエストを中継してから、バックエンドからレスポンスを受け取るまでの時間	ms
Latency	API Gateway がクライアントからリクエストを受け取ってから、クライアントにレスポンスを返すまでの時間	ms

メトリクス	意味	単位
Connect Count	$connect ルートの統合に送信されたメッセージ数	個
Message Count	クライアントとの間で送受信される、送信されたメッセージ数	個
Integration Error	バックエンド統合からの ステータスコード4XX/5XX のエラー数	個
ClientError	統合が呼び出される前に API Gateway によって4XXを返したエラー数	個
Execution Error	統合を呼び出す際に発生したエラー数	個
Integration Latency	API Gateway がバックエンドにリクエストを中継してから、バックエンドからレスポンスを受け取るまでの時間	ms

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 - カスタムドメインの設定

「カスタムドメイン」を登録し、証明書とDNSを設定することで
独自ドメイン名で登録したAPIを呼び出し可能 (※エンドポイントタイプが「プライベート」以外)

エンドポイントURL (https://<api-id>.execute-api.<region-id>.amazonaws.com/<stage-name>)

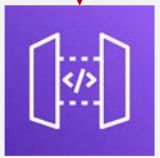
カスタムドメイン (例： https://**api.example.com**/my-api-id-service)

①ドメイン(サブドメイン)のパブリック証明書を取得

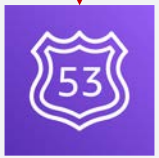
AWS Certificate Manager

※エンドポイントタイプが「エッジ最適化」の場合は
米国東部(us-east-1)リージョンで証明書を作成
or インポート(「リージョン」の場合はそのリージョン)

②「カスタムドメイン」に
証明書とベースパスを登録



API Gateway



③ 利用DNSにエイリアスや
CNAMEとして登録

Amazon Route 53
(または利用DNSサービス)

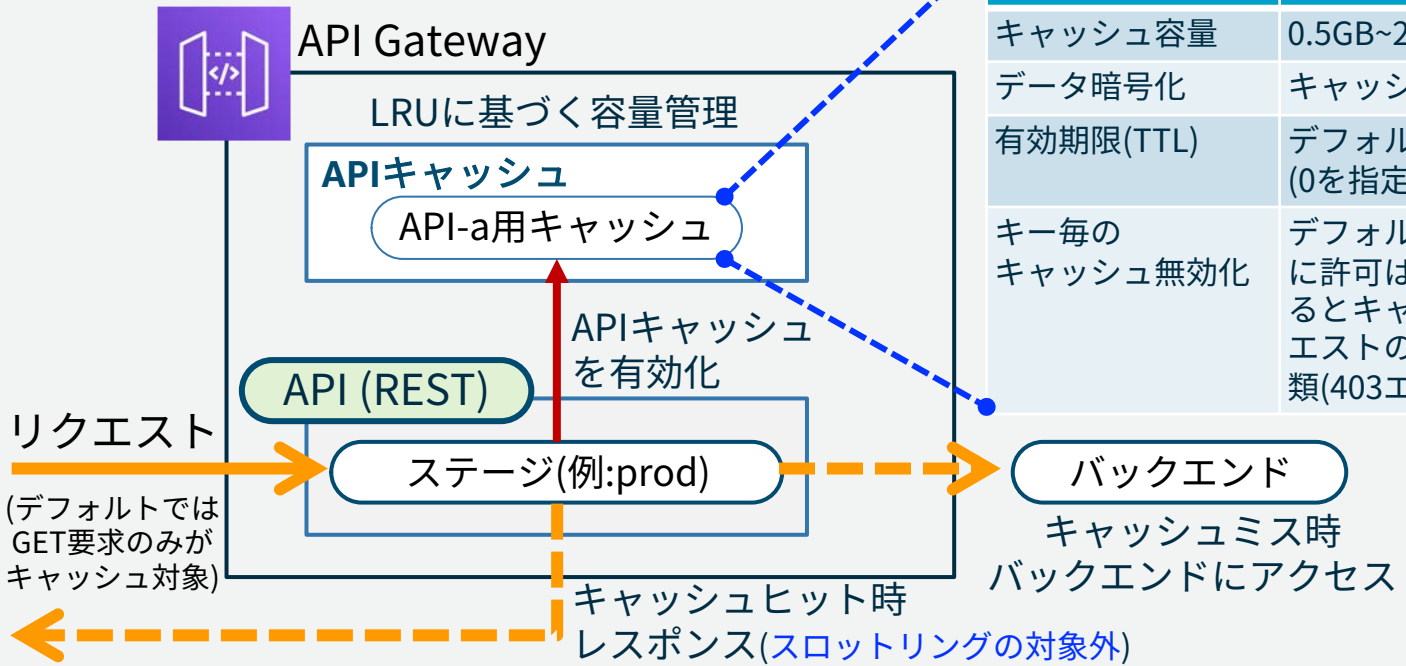
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 - キャッシュ

REST

REST APIではAPIのステージ毎にキャッシュを定義し、
バックエンドへのトラフィック削減と低レイテンシの実現に利用可能



設定項目	説明
キャッシュ容量	0.5GB~237GBでの8種類から選択
データ暗号化	キャッシュデータの暗号化有無を指定
有効期限(TTL)	デフォルト300、最長3600(単位:秒) (0を指定するとキャッシュ無効)
キー毎の キャッシュ無効化	デフォルトではキャッシュ無効化指示に許可は不要の設定(false)。trueにするとキャッシュ無効化を指示するリクエストの扱いを、予め用意された3種類(403エラー/警告/無視)から選択

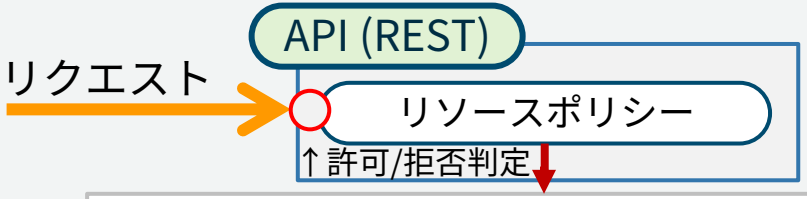
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 - リソースポリシー

REST

REST APIではAPIのリソースポリシーを指定可能 (IAMポリシーと併用可能)




```
{
  "Version": "2012-10-17",
  "Statement": [
   {
    "Effect": "<Allow または Deny>",
    "Principal": "…",
    "Action": "<①~③のいずれか>",
    "Resource": "arn:aws:execute-api:<region>:<account-id>:*",
    "Condition":{
       … (ソースIPやVPCエンドポイントなどの条件演算子) …
    }
   }, …
  ]
}
```

① "execute-api:Invoke" … 呼び出し操作
② "ececute-api:InvalidateCache"… キャッシュ無効化操作
③ "execute-api:*" … ① + ② を意味

※ APIを示すARN形式文字列を指定(ワイルドカード利用可)
「arn:aws:execute-api:<region>:<account-id>:<api-id>/<stage-name>/<HTTP-Verb>/<resource-path-specifier>」

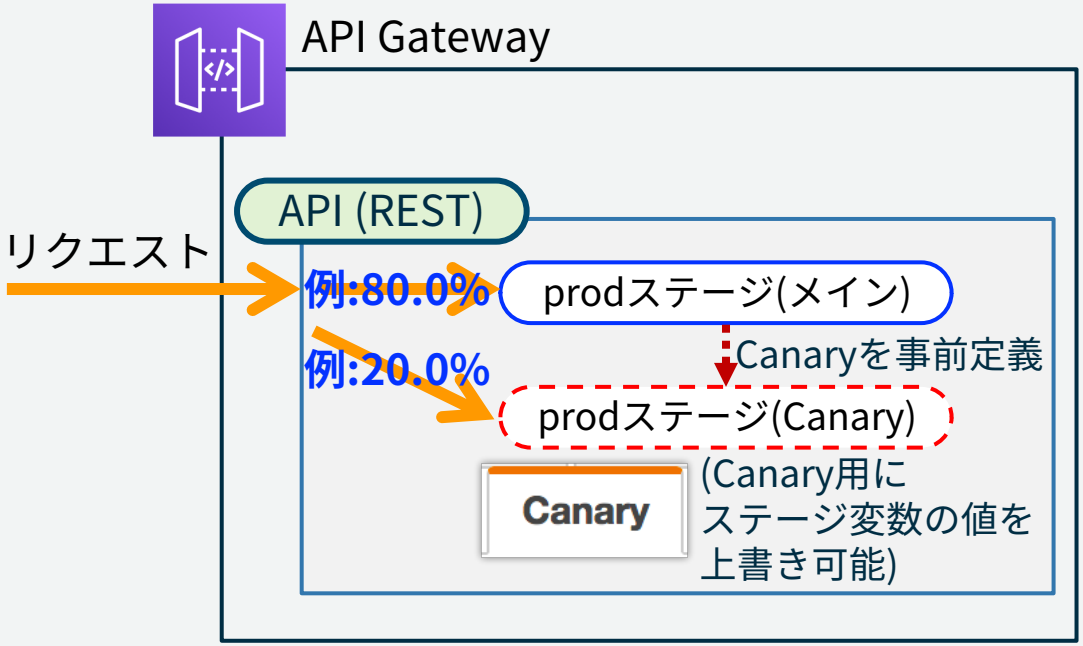
- リソースポリシーの利用例：
  - アクセス元を制限する
    - 指定AWS アカウントのユーザ
    - 指定ソース IP アドレス範囲または CIDR ブロック
    - 指定VPC または 任意アカウントの VPC エンドポイント

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

API設定 - カナリアリリース

REST

REST APIの各ステージ(メイン)に紐付く特別なステージ「Canary」を作成し、リクエストを指定の比率でCanaryへ流すことが可能



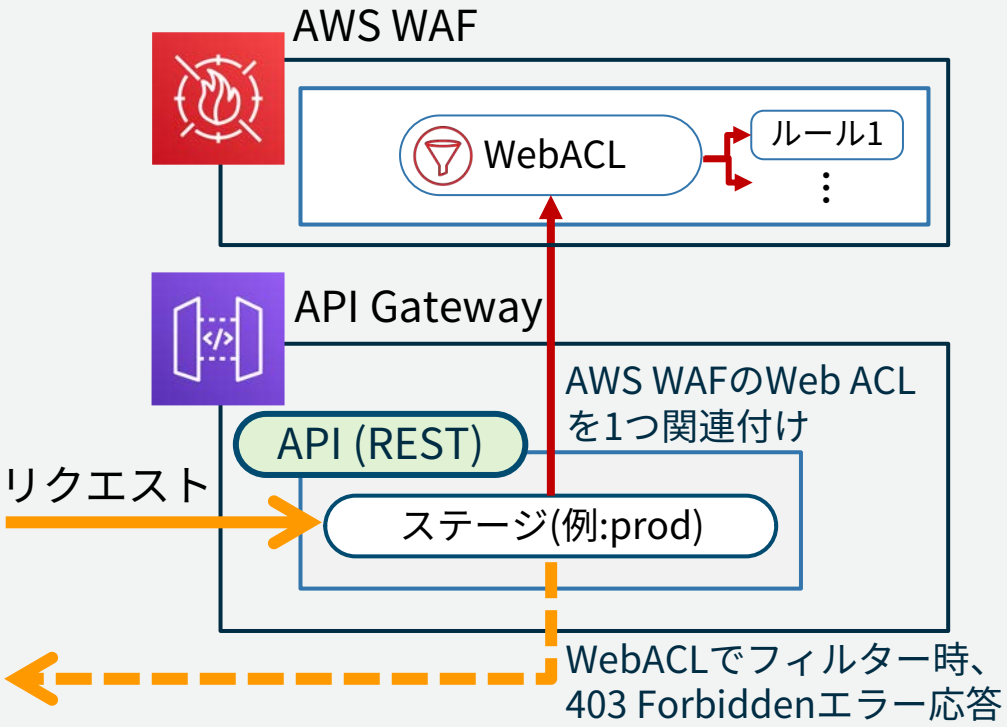
- あるステージに対して「Canary」を定義すると、APIの「デプロイ」操作でそのステージを指定した際に一旦Canary に対してのみ行われるようになる(メインの内容は変わらない)
- Canaryに対して以下の操作が可能：
  - 昇格
    - Canaryのデプロイ内容をメインのステージに反映
  - 削除
    - Canaryを削除して、メインステージのみの構成に戻す

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

API設定 - AWS WAF連携(WebACLとの関連づけ) REST

REST APIではAPIステージにAWS WAFのWebACLを指定可能



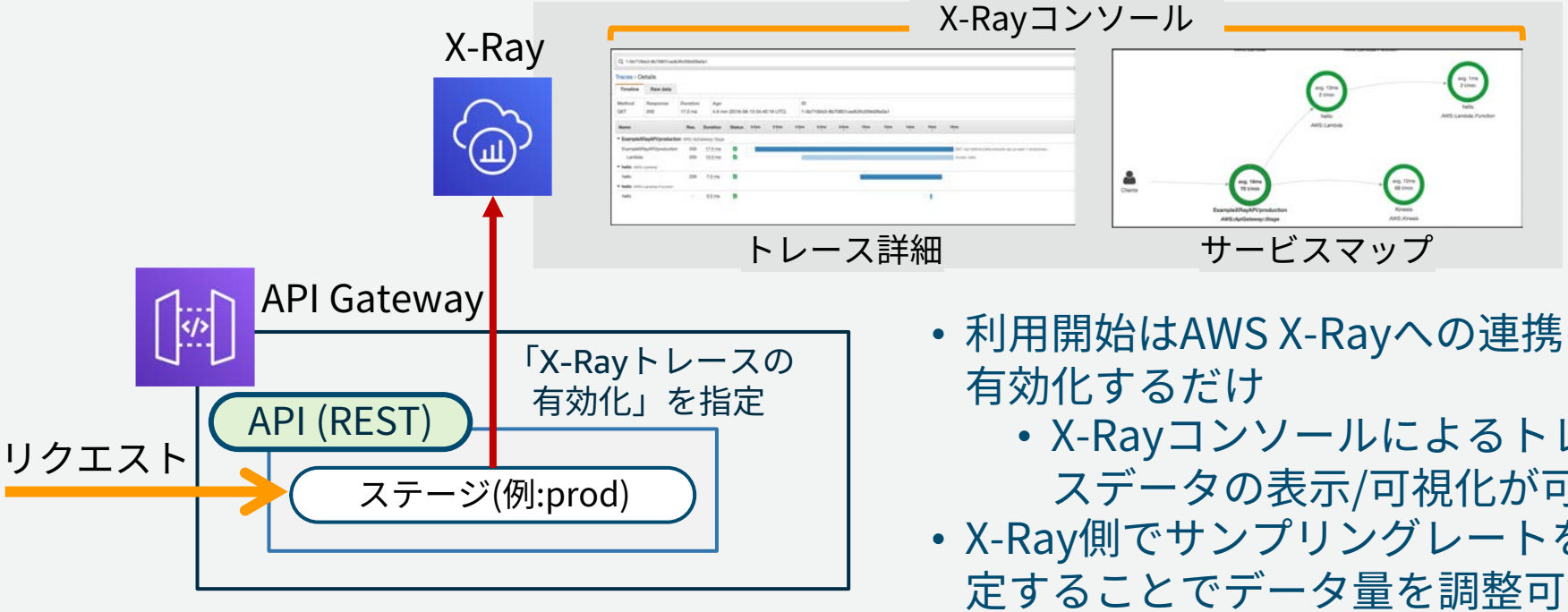
- AWS WAFのWeb API保護を利用
  - SQLインジェクションやXSSなどの攻撃からAPIを保護
- アクセスポリシーや認証など、その他のアクセス制御よりも前にWebACLによって評価される
- APIステージ設定UI または WAF WebACLの新規作成画面でAPI GatewayのAPI/ステージを指定

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

API設定 - AWS X-Ray連携

REST

REST APIではAPIステージのログ設定としてAWS X-Rayへの
連携によるリクエストのトレースと分析およびデバッグが可能



- 利用開始はAWS X-Rayへの連携を有効化するだけ
  - X-Rayコンソールによるトレースデータの表示/可視化が可能
- X-Ray側でサンプリングレートを設定することでデータ量を調整可能

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

本日のアジェンダ

- はじめに - API / Web API とは
- Amazon API Gateway 概要
    - 全体イメージ
    - ユースケース
    - REST API / WebSocket API
    - その他主要な機能
    - 補足的機能 や ご使用料金の情報など
- まとめ

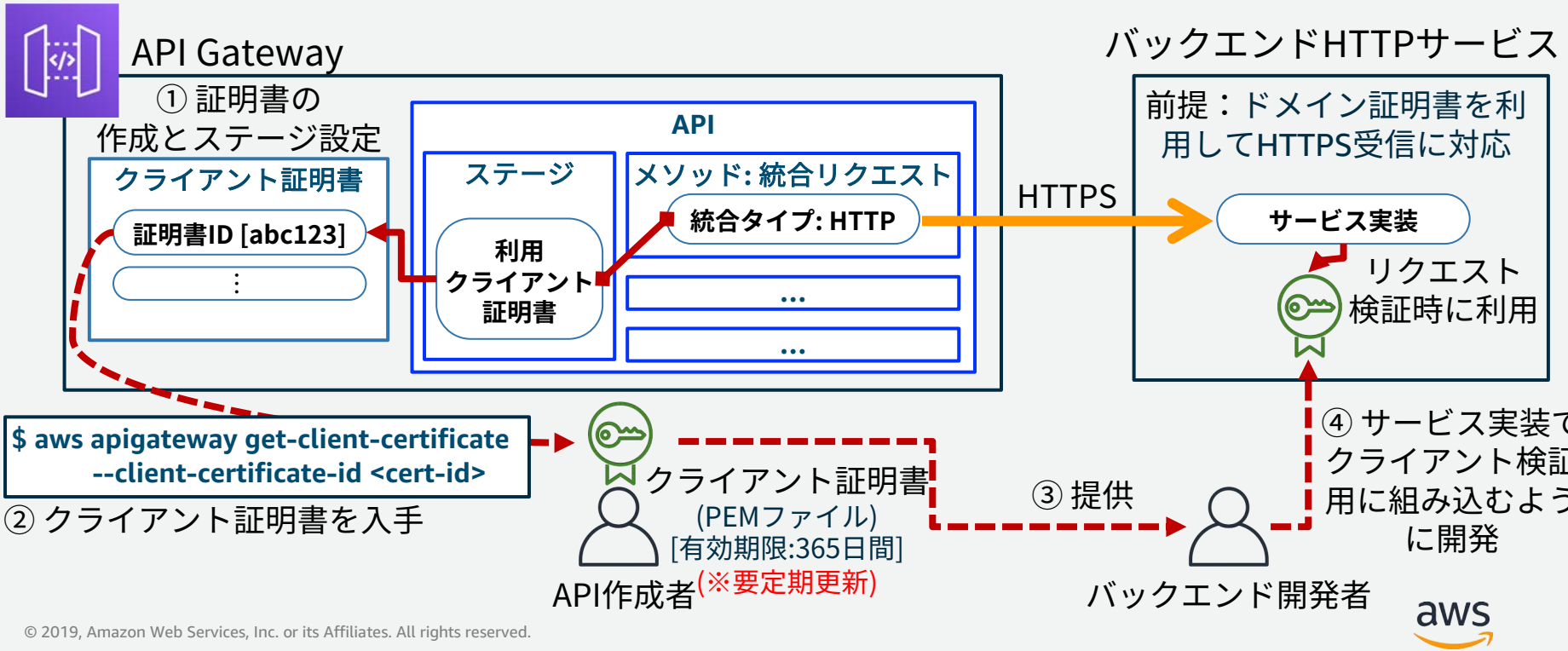
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

クライアント証明書

REST

HTTPエンドポイントタイプを利用する際に、API Gatewayからのアクセスであることを示すクライアント証明書を作成・利用可能

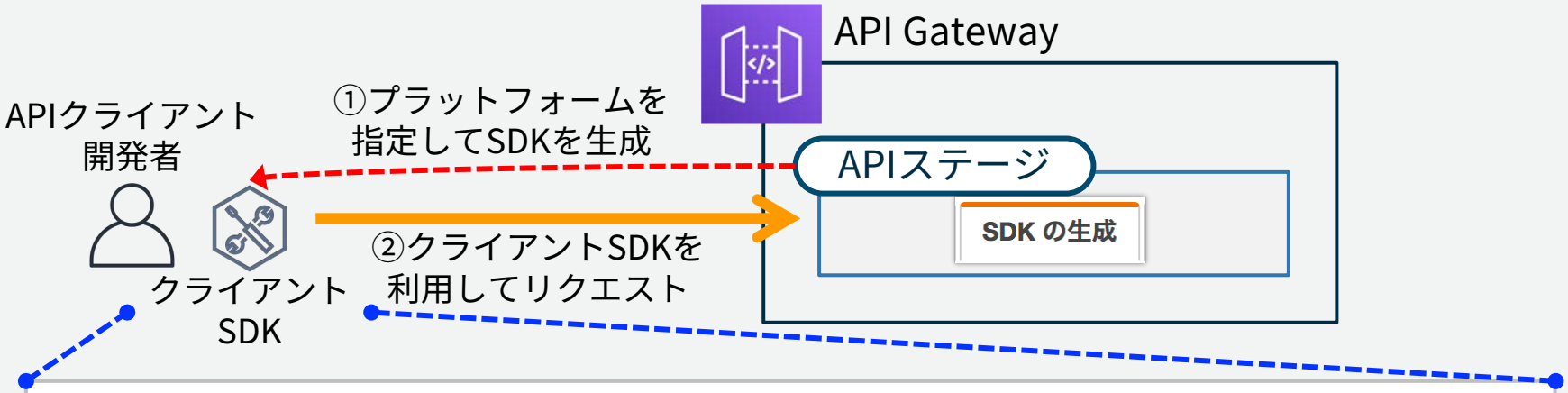


© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

クライアントSDK

REST

APIステージ単位でREST API呼び出しクライアントをSDKとして生成可能 (クライアント開発作業を効率化)



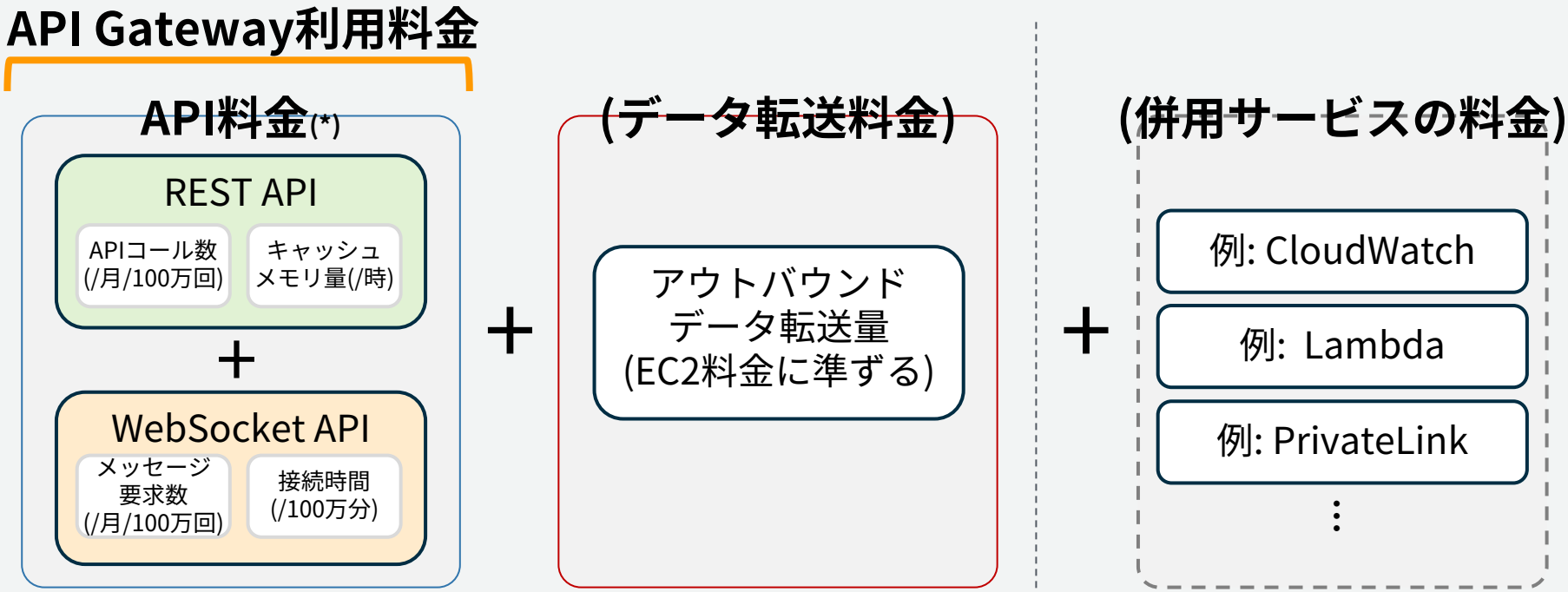
- 対応プラットフォーム：Android, JavaScript, iOS(Objective-C/Swift), Java, Ruby
- クライアントSDKを利用する動機(メリット)：
  - スロットリング時のリトライやAWS署名v4生成・セット処理が実装済み
  - REST APIのIN/OUTで必要なモデルのマーシャル/アンマーシャル処理にも対応

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws

Amazon API Gateway のご使用料金の構造

API Gatewayは、API が使用されている分のみのご使用料金
(最低利用料金 や 初期費用 なし)



(*) AWSサインアップ日より12ヶ月間内での無料枠あり

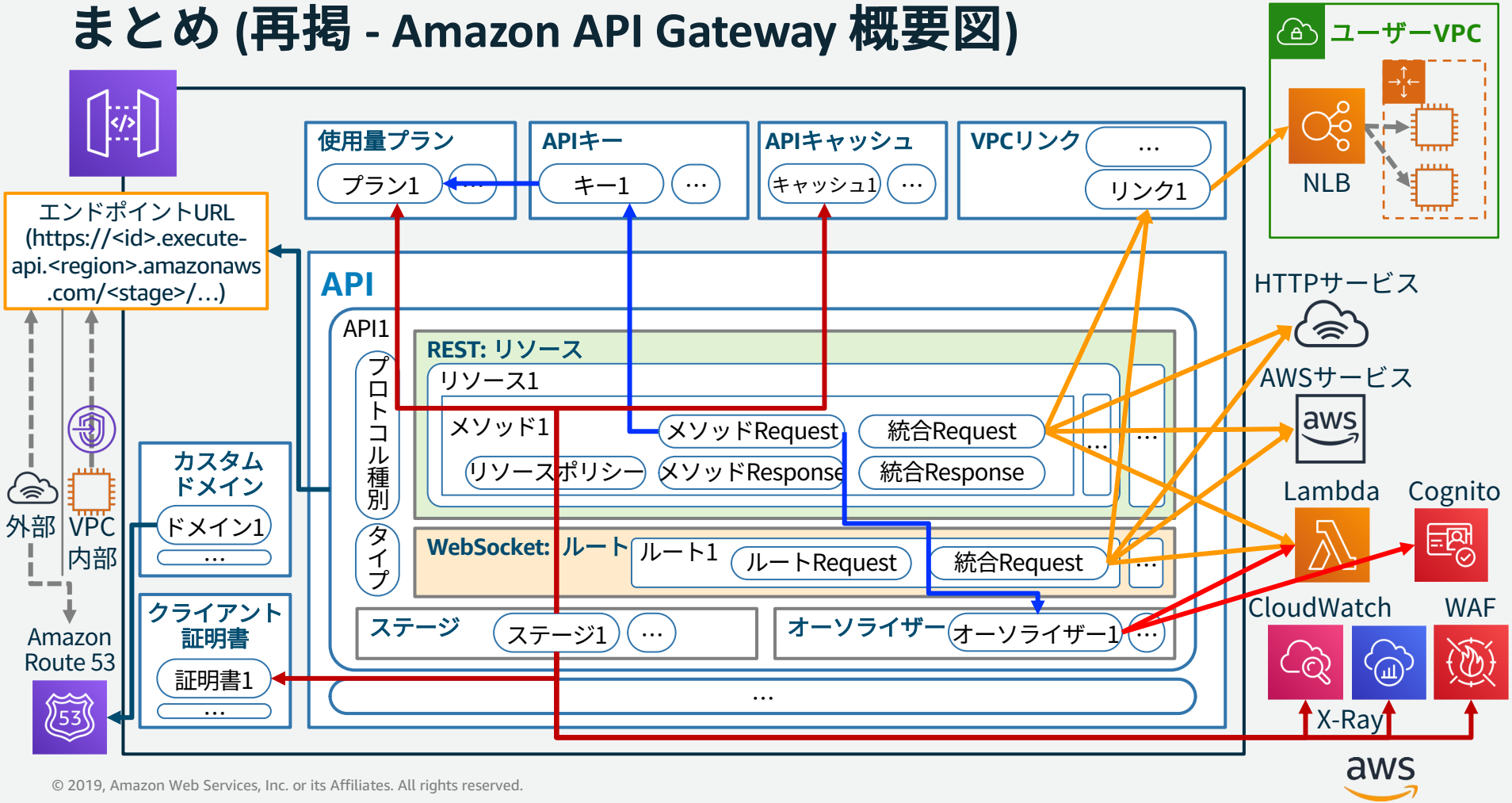
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

本日のアジェンダ

- はじめに - API / Web API とは
- Amazon API Gateway 概要
    - 全体イメージ
    - ユースケース
    - REST API / WebSocket API
    - その他主要な機能
    - 補足的機能 や ご使用料金の情報など
- まとめ

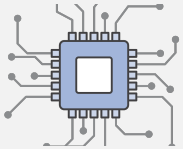
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

まとめ (再掲 - Amazon API Gateway 概要図)



© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

まとめ



1. インフラの管理
(可用性とスケーラビリティ)

- マネージドサービスとしてAWSが管理
- サーバーレスアーキテクチャの実現にも適した基盤

2. APIの管理
(設定やデプロイの制御)

- ステージや統合リクエスト/レスポンスなどの構成要素を通じた設定
- カナリアリリースも設定で実現

3. 認証と認可
(アクセスの制御)

- オーソライザーやAWS署名v4を使った認証と認可
- リソースポリシーによる細かなアクセス制御

4. 流量制御と保護
(スロットリング)

- APIキーと使用量プランによるスロットリング
- WAF連携によるAPIの保護

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

ご視聴ありがとうございました

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

過去資料
https://amzn.to/JPArchive

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.