

AWSで名前解決する技術 ～WebサービスでのRoute 53の構成一覧～

NW-JAWS #15 ～改めて振り返るRoute 53～

村田亜弥 アクセンチュア株式会社

0.自己紹介



村田 亜弥 (alichan)

仕事：AWSを用いた公共サービスやメディアサービスの基盤刷新やPoC開発でアプリケーションエンジニア/クラウドエンジニアとして仕事しています。

好きなこと：個人でAWSを用いた個人開発や技術記事執筆/登壇、イベントの開催等を行なっています。AWSとプログラミングが好きです！

個人ではalichanという名前で活動しています。

資格：Solutions Architect - Professional



: <https://github.com/alichan-69>



: <https://zenn.dev/alichan>



: <https://x.com/alichan0609>



1.概要

AWSで名前解決と言えばRoute 53



Route 53



グローバルに分散されたDNSサーバーによる**可用性**

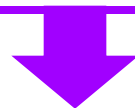


エイリアスレコードによるAWSサービスへのルーティングの**容易性**

example.com



ドメイン購入、レコード管理、リソースのヘルスチェックを一つのサービスで行える**網羅性**



WebサービスでのRoute 53構成パターンを紹介



1.概要

Route 53振り返り

パブリックホストゾーン

購入したドメイン名と同じパブリックホストゾーンが自動で作成される

AWS内外へのリソースへのルーティングや証明書情報等を含んだレコードを登録することができる物

NSレコード

ホストゾーンに割り振られた4つのネームサーバー情報が格納されたレコード

SOAレコード

ドメインに設定されたDNS情報が格納されたレコード

CNAMEレコード

あるドメインに対するクエリを別のドメインにルーティングするレコード

エイリアスレコード

AWSリソースへのルーティングを簡単に登録/管理できるレコード
Aレコードとして登録される

レコード

AWS内外へのリソースへのルーティングや証明書情報等ドメインやサブドメインへのルーティングに必要な情報を登録する物

Route 53 > ホストゾーン > [ドメイン]

パブリック [ドメイン].com 情報

▶ ホストゾンの詳細

レコード (5)

DNSSEC 署名

ホストゾンのタグ

レコード (5) 情報

Automatic モードは最適なフィルタ結果に最適化された現在の検索

Q プロパティまたは値でレコードをフィルタリングする

☐	レコー...	タ...	ルーテ...	差別...	エイ...	値/トラフィックのルーテ...	TTL (秒)	パル...
☐	[ドメイン]	A	シンプル	-	はい	[ドメイン].cloudfront.net.	-	
☐	[ドメイン]	NS	シンプル	-	いいえ		72800	-
☐	[ドメイン]	SOA	シンプル	-	いいえ		900	-
☐	[ドメイン]	CNAME	シンプル	-	いいえ	[ドメイン]	300	-
☐	[ドメイン]	A	シンプル	-	はい	[ドメイン]	-	-

> 今回使用するのはこちら

2.構成パターン一覧

Webサービス構成パターン一覧

※エイリアスレコードに対応しているサービスを中心に構成

A

サーバーレス

+

リクエスト
従量課金

+

カスタマイズ性
高

=

1. Route 53 + CloudFront + S3、Route 53 + (CloudFront) + API Gateway + Lambda
2. Route 53 + CloudFront + API Gateway + Lambda with Lambda Web Adapter

B

サーバーレス

+

リクエスト
従量課金

+

カスタマイズ性
低

=

1. Route 53 + Amplify
2. Route 53 + App Sync

C

サーバーレス

+

時間
従量課金

+

カスタマイズ性
高

=

1. Route 53 + (CloudFront) + ALB/NLB + ECS/EKS

D

サーバーレス

+

時間
従量課金

+

カスタマイズ性
低

=

1. Route 53 + (CloudFront) + AppRunner

E

サーバー
管理あり

+

時間
従量課金

+

カスタマイズ性
高

=

1. Route53 + (CloudFront) + Elastic Beanstalk
2. Route53 + (CloudFront) + ALB/NLB + EC2



Route 53 + CloudFront + S3

Route 53 + (CloudFront) + API Gateway + Lambda

エイリアスレコード1つ



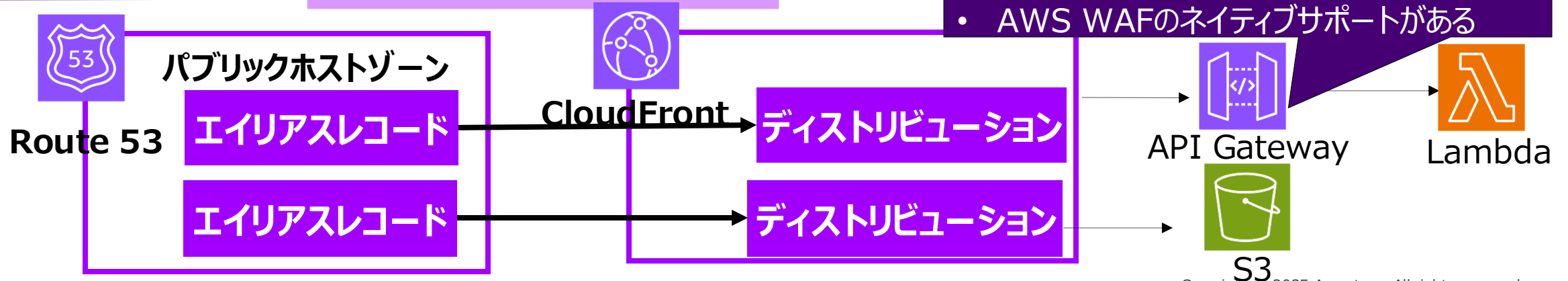
ディストリビューション分割論点

- ログ分離の必要性
- キャッシュの扱いの慎重度...etc

API Gatewayへの直接ルーティングもあり

- 頻繁なデータ更新によりキャッシュし辛い
- AWS WAFのネイティブサポートがある

エイリアスレコード2つ



A-2.サーバーレス+リクエスト従量課金+カスタマイズ性高の構成パターン

Route 53論点 : 
構成パターン論点 : 

Route 53 + CloudFront + API Gateway + Lambda with Lambda Web Adapter

モルポ構成の場合の考慮点

- レスポンスに静的資源と動的レスポンスが入り混じるのでCDNキャッシュの設定に慎重さが必要

Webフレームワーク製Webサービス



NEXT.js



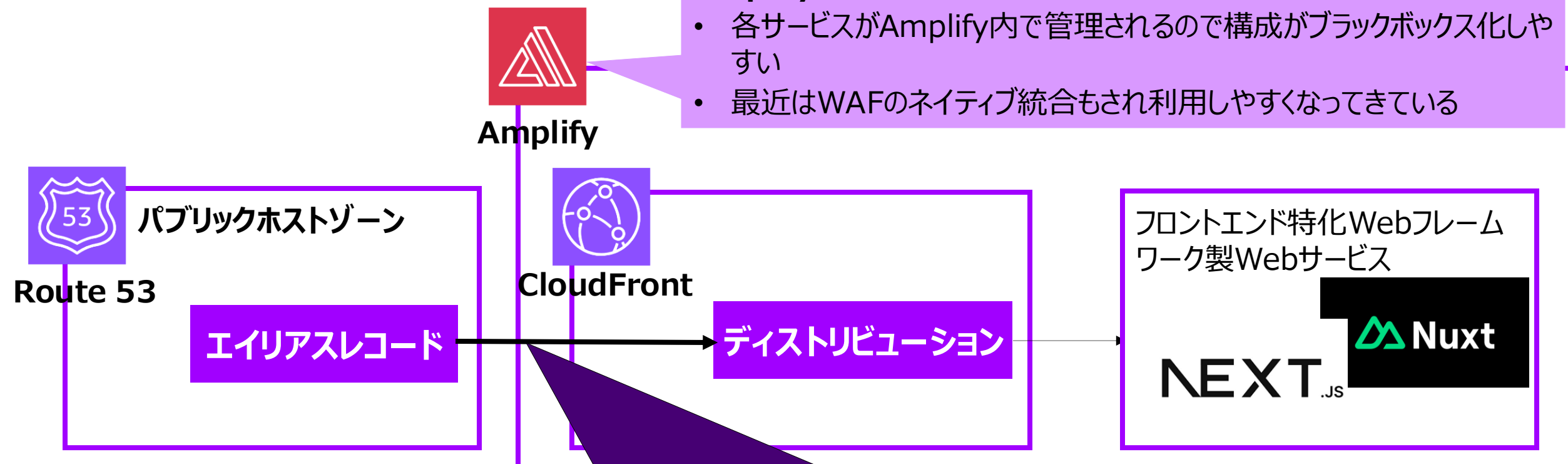
B-1.サーバーレス+リクエスト従量課金+カスタマイズ性低の構成パターン

Route 53論点 : 
構成パターン論点 : 

Route 53 + Amplify

Amplify利用時の考慮点

- 各サービスがAmplify内で管理されるので構成がブラックボックス化しやすい
- 最近はWAFのネイティブ統合もされ利用しやすくなってきている



Amplifyでの名前解決の実装

- Route 53で購入したドメインをカスタムドメインとしてAmplifyに登録
- AmplifyがパブリックホストゾーンにAmplify管理下のCloudFront ディストリビューションエイリアスレコードを発行



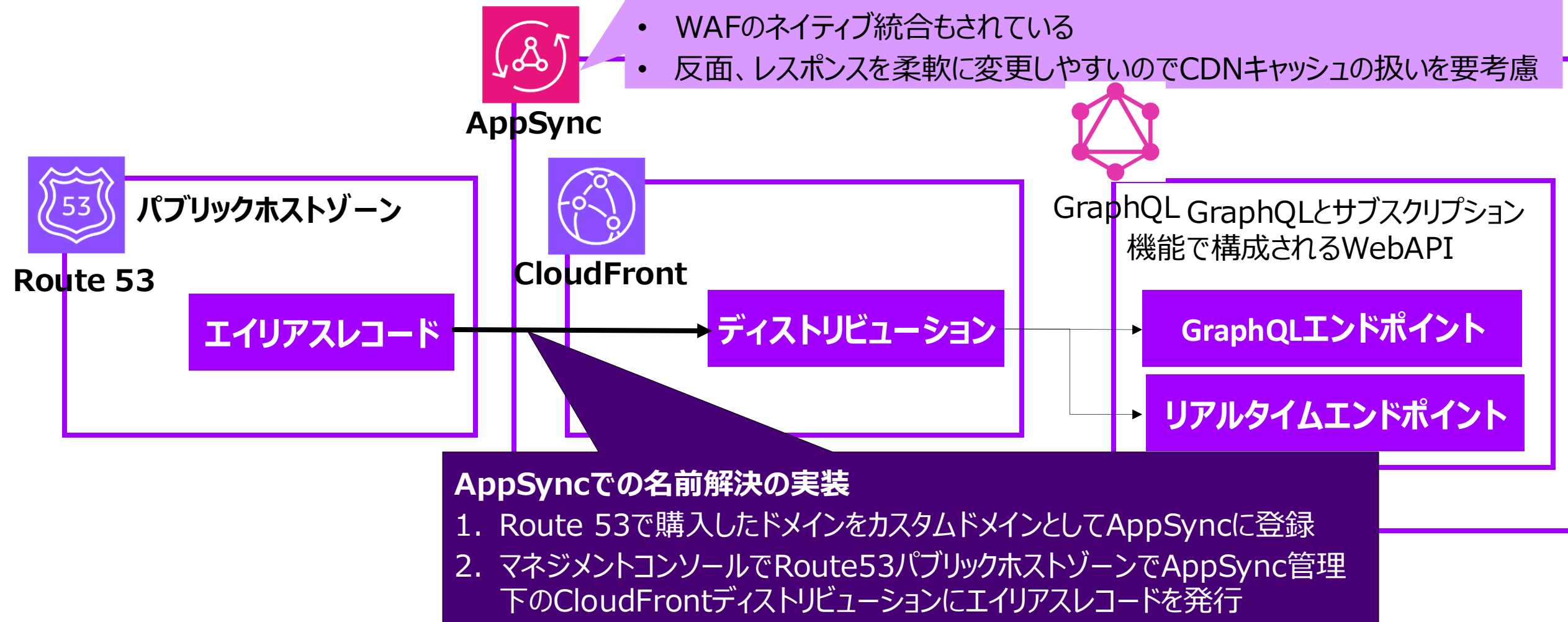
B-2.サーバーレス+リクエスト従量課金+カスタマイズ性低の構成パターン

Route 53論点 : 
構成パターン論点 : 

Route 53 + AppSync

AppSync利用時の考慮点

- Lambdaを挟まなくてもAppSync内でのスキーマ定義やリゾルバー定義のみでWeb APIを構成可能
- WAFのネイティブ統合もされている
- 反面、レスポンスを柔軟に変更しやすいのでCDNキャッシュの扱いを要考慮



C-1.サーバーレス+時間従量課金+カスタマイズ性高の構成パターン

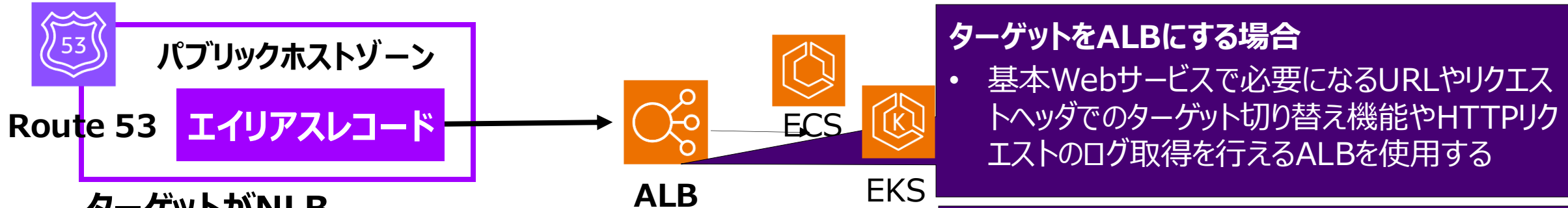
Route 53論点: 
構成パターン論点: 

Route 53 + (CloudFront) + ALB/NLB + ECS/EKS

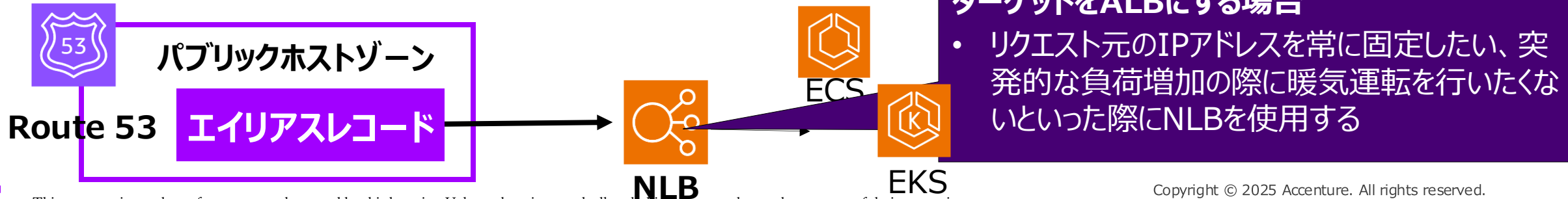
ターゲットがCloudFront



ターゲットがALB



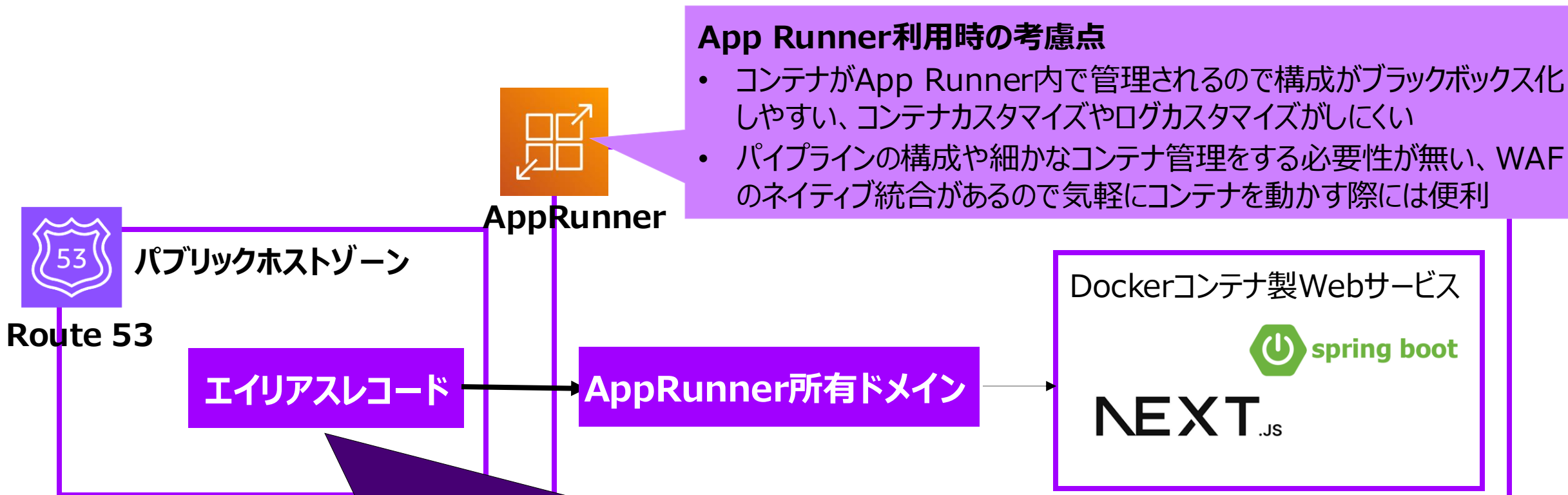
ターゲットがNLB



D-1.サーバーレス+時間従量課金+カスタマイズ性低の構成パターン

Route 53論点 : 
構成パターン論点 : 

Route 53 + (CloudFront) + App Runner



App Runner利用時の考慮点

- コンテナがApp Runner内で管理されるので構成がブラックボックス化しやすい、コンテナカスタマイズやログカスタマイズがしにくい
- パイプラインの構成や細かなコンテナ管理をする必要性が無い、WAFのネイティブ統合があるので気軽にコンテナを動かす際には便利

App Runnerでの名前解決の実装

1. Route 53で購入したドメインをカスタムドメインとしてApp Runnerに登録し、エイリアスレコードを発行するように設定
2. App RunnerがパブリックホストゾーンにApp Runner所有ドメインのエイリアスレコードを発行



Route53 + (CloudFront) + Elastic Beanstalk



- 最近はECS等のコンテナオーケストレーションツールでコンテナ管理してアプリケーションを動かすことが主流になっている
- クラウドリフト&シフトの文脈でDockerコンテナ化されていないWebサービスを一度EC2上に移動して管理する際に便利

仮想サーバー上Webサービス



spring boot

NEXT.JS

NEXT.JS

...

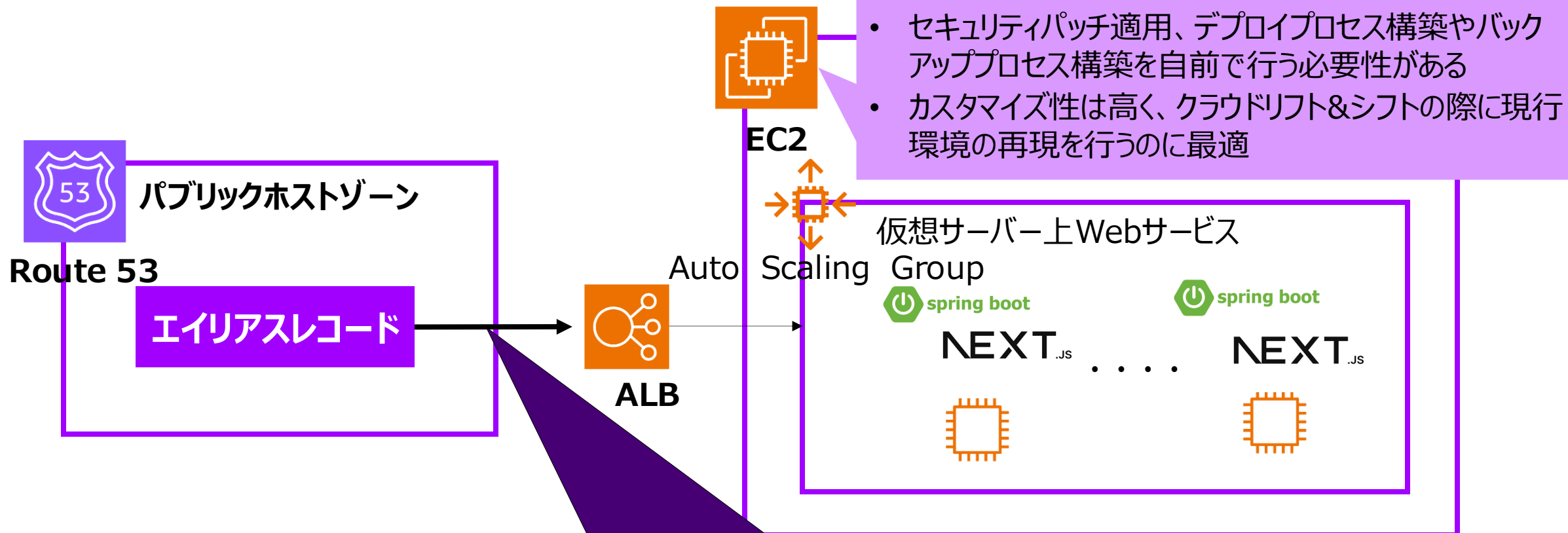
1. Elastic Beanstalkでアプリケーションを作成後Elastic Beanstalk所有ドメインが発行

2. Route 53からElastic Beanstalkに対してエイリアスレコードが発行される

E-2.サーバー管理あり+時間従量課金+カスタマイズ性高の構成パターン

Route 53論点 : 
構成パターン論点 : 

Route53 + (CloudFront) + ALB/NLB + EC2



EC2での名前解決の実装

1. Auto Scaling GroupやEC2の前段にALBを配置
2. Route 53がALBに対してエイリアスレコードを発行



7.まとめ

Route53の可能性は無限大

- グローバルに分散されたDNSサーバーにより**可用性が高い**、視覚的に分かりやすいUIにより入門ハードルの高いDNSの**構造が分かりやすい**、ルーティングターゲットへの**ヘルスチェック完備**と安心して利用できる
- エイリアスレコード対応サービスは続々と増えており、**他AWSサービスと連携がし易く**、構成パターンも豊富に考えられる
- インターネット公開DNSサーバーとしてではなくプライベートDNSサーバーとしても使用でき、**オンプレミスとの親和性も高い**

⇒これからもAWSサービスの主力の一つであり続ける