



このコンテンツは公開から3年以上経過しており内容が古い可能性があります
最新情報については[サービス別資料](#)もしくはサービスのドキュメントをご確認ください

[AWS Black Belt Online Seminar]

Amazon CloudFront deep dive

サービスカットシリーズ

Solutions Architect 藤原 吉規
2020/10/28



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



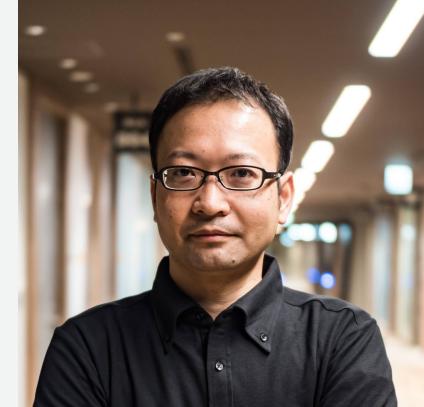
過去資料
<https://amzn.to/JPArchive>



自己紹介

藤原 吉規 (ふじわら よしのり)

- 西日本担当 ソリューションアーキテクト
- AWS 大阪オフィスにいます
- 関西のビジネスチャットスタートアップ企業で
6 年間 AWS を活用
- AWS サムライ 2012
- 好きな AWS サービス: **Amazon CloudFront, AWS 技術サポート**



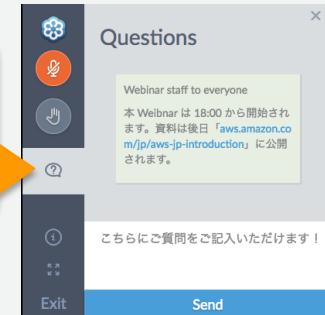
AWS Black Belt Online Seminar とは

「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、Amazon 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

内容についての注意点

- 本資料では2020年10月28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には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.

本日のアジェンダ

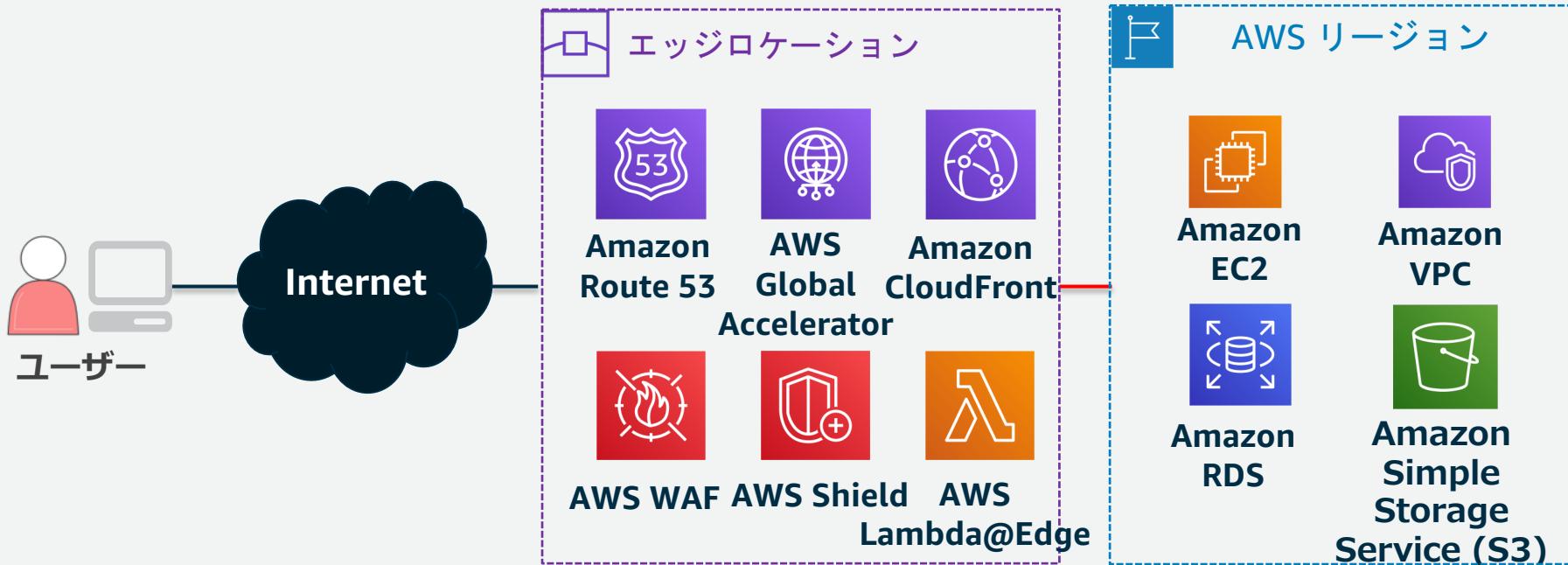
- AWS のエッジサービス と Amazon CloudFront
- CloudFront deep dive
 - Distribution
 - Origin
 - Behavior
 - Distribution に関する機能
- まとめ



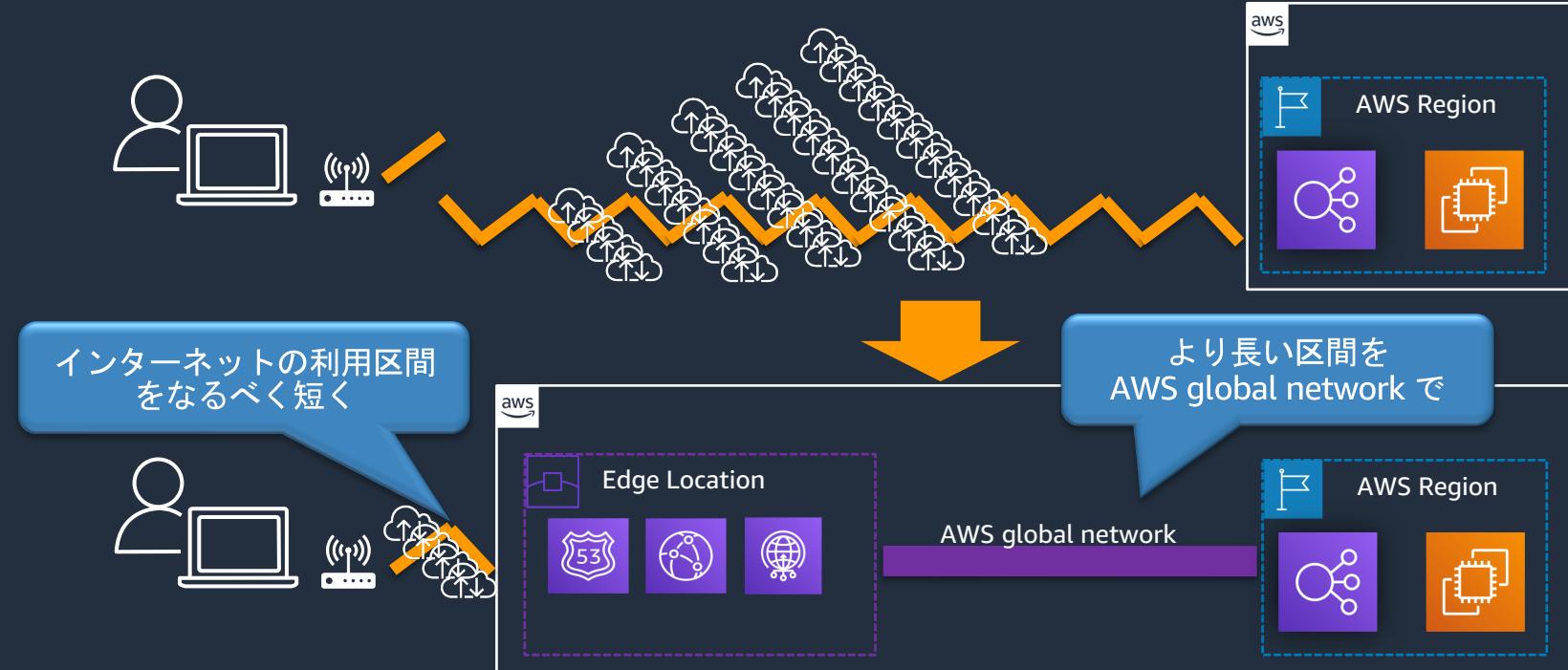
AWS のエッジサービス と Amazon Cloud Front

エッジサービス

AWS のエッジロケーションから提供されるサービス群
AWS のサービスへのアクセスをユーザーに近い場所から提供



エッジサービス

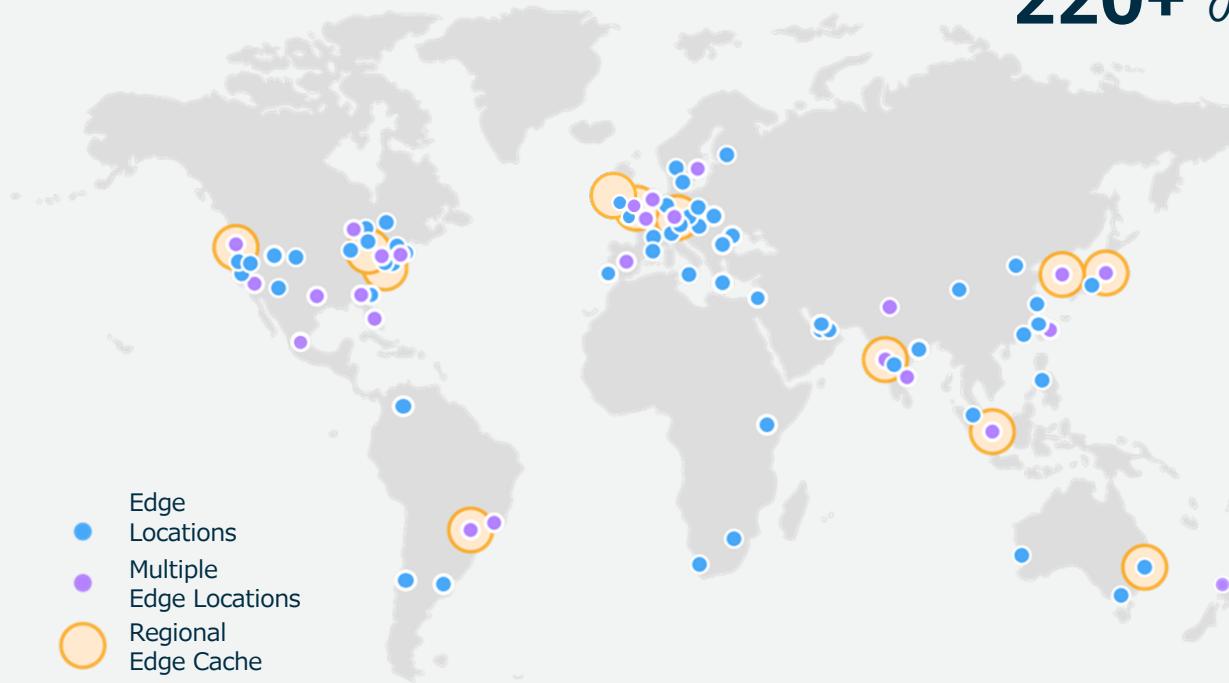


- AWS global network を利用することにより、より高速で安定したユーザ体験を提供
- CloudFront のキャッシュも有効活用

220+ の POP (Point Of Presence)

12 のリージョン別
エッジキャッシュ

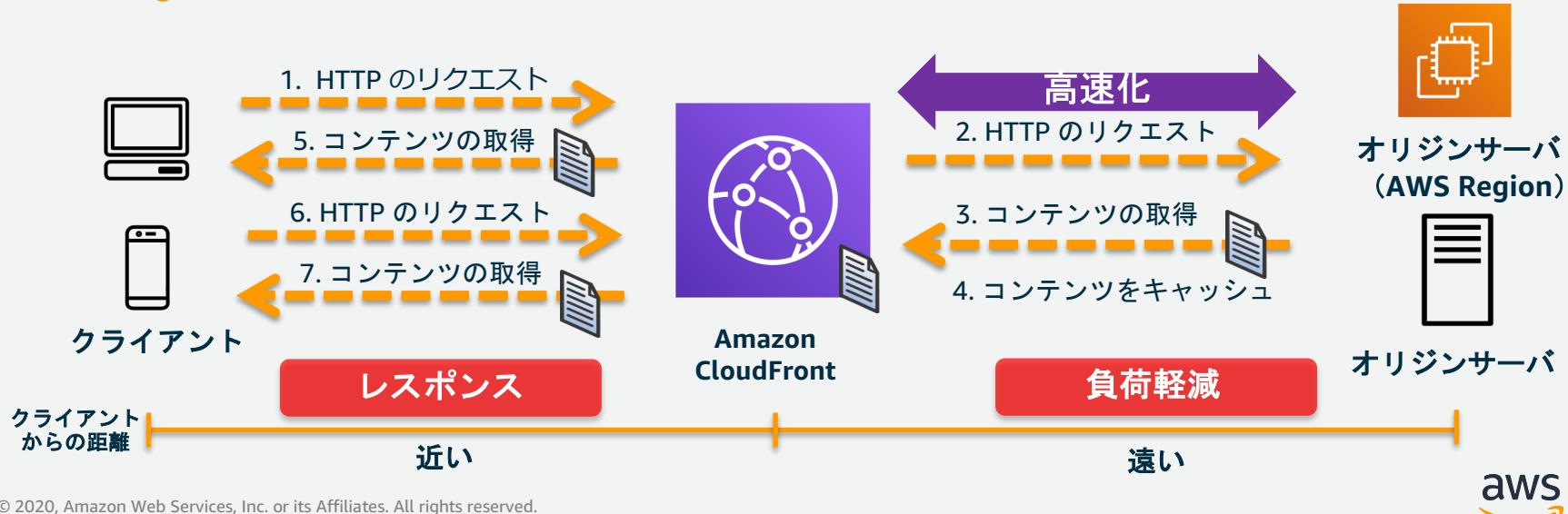
44 か国 **87** 都市に展開



CloudFront

Fast, highly secure and programmable content delivery network (CDN)
高い安全性と高性能を実現するプログラム可能なコンテンツデリバリー・ネットワーク

- ユーザーを一番近いエッジロケーションに誘導することで **配信を高速化**
- エッジサーバでコンテンツのキャッシングを行い **オリジンの負荷をオフロード**
- AWS global network を利用することによる非キャッシングコンテンツの高速化**



ユーザー事例 - API アクセラレーション

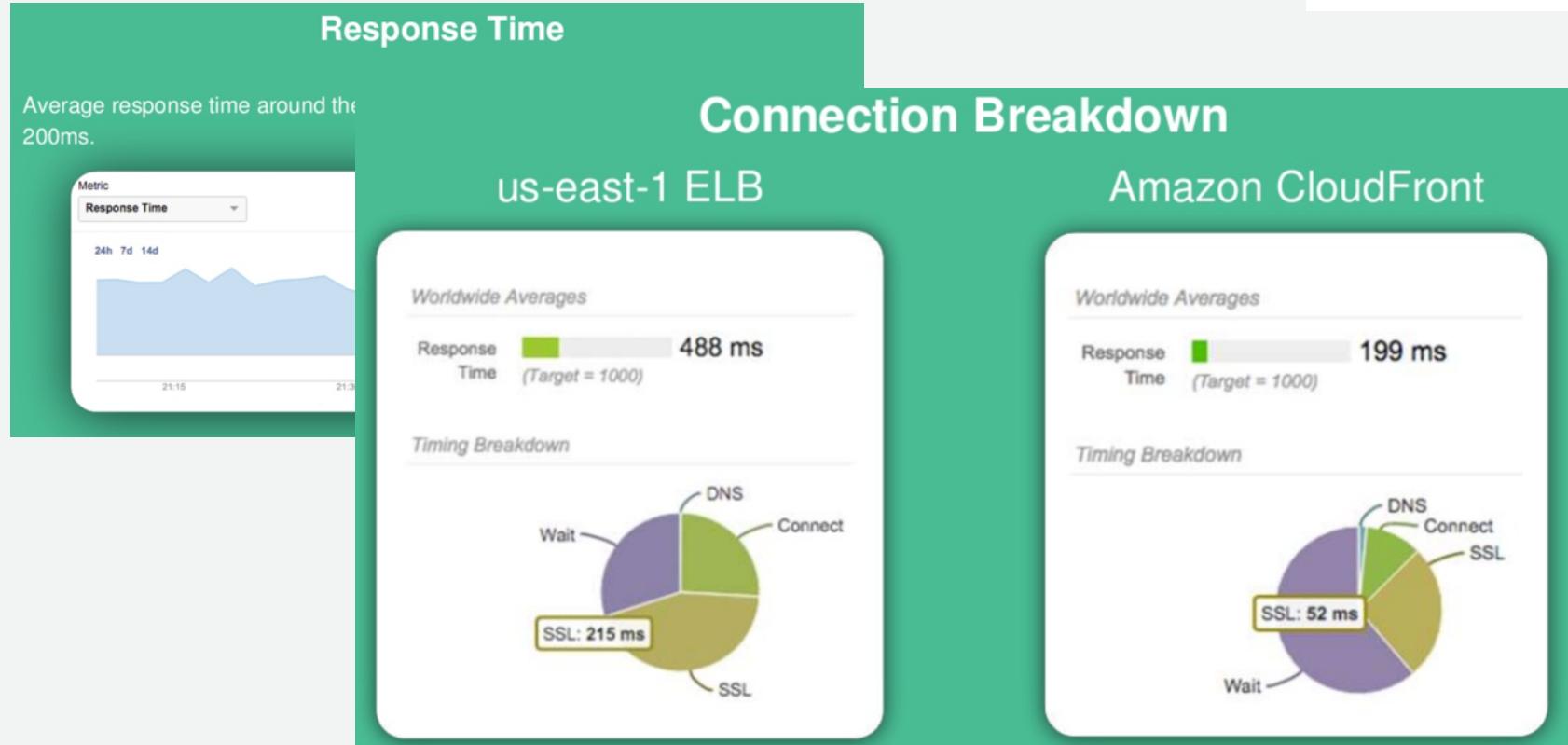


Slack Web API

- HTTPS エンドポイントに対して POST / GET
- レスポンスは JSON オブジェクト
- Amazon CloudFront 利用して、グローバルな API アクセラレーションを実現

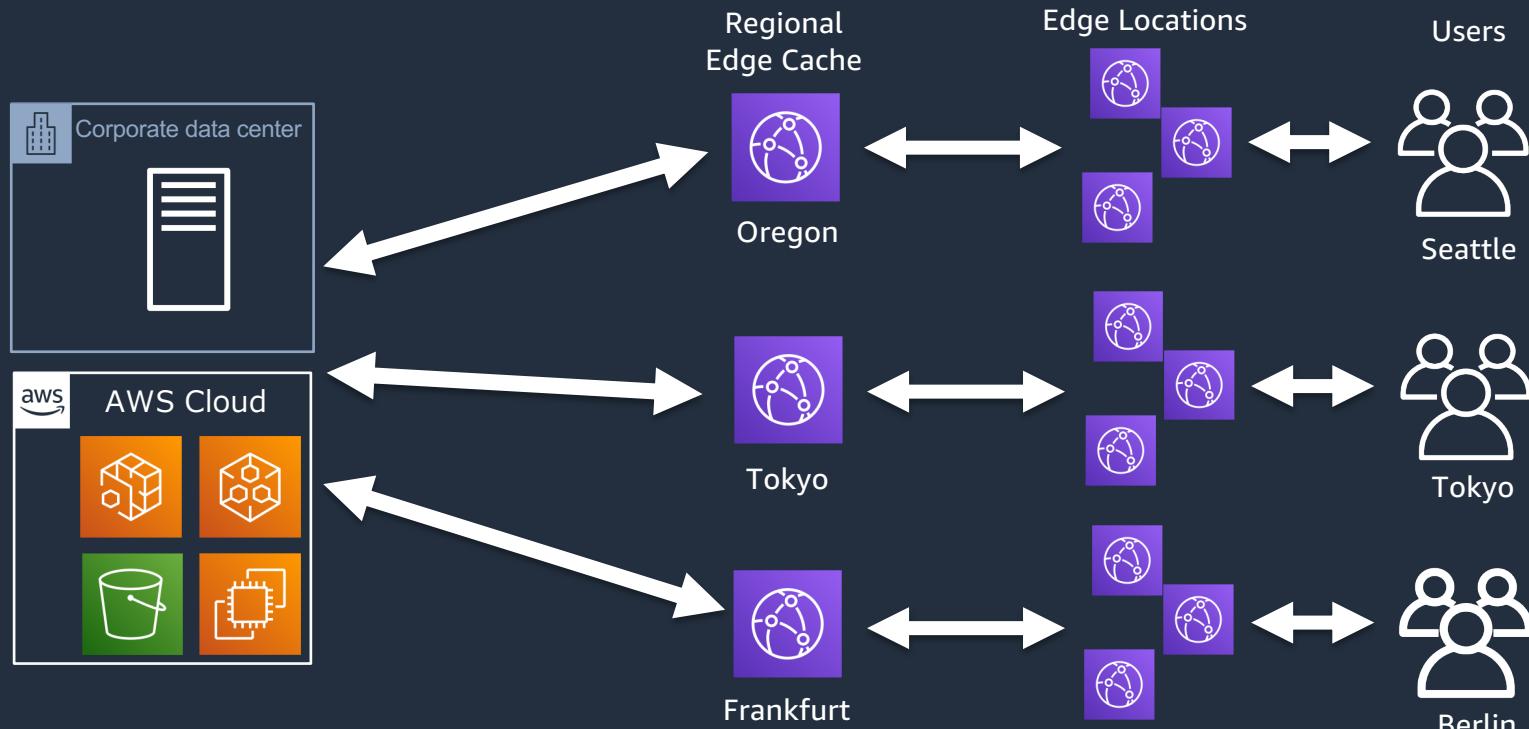


ユーザー事例 - API アクセラレーション



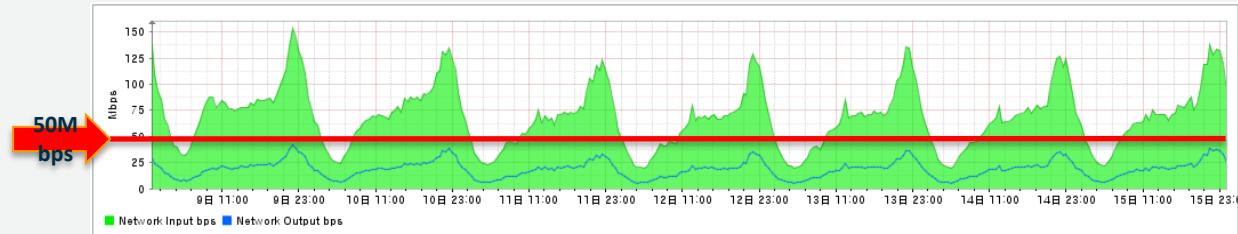
リージョン別エッジキャッシュ (REC)

- ・ ユーザーからのリクエストを REC で集約しオリジンにリクエスト
- ・ エッジロケーションに近い REC を利用



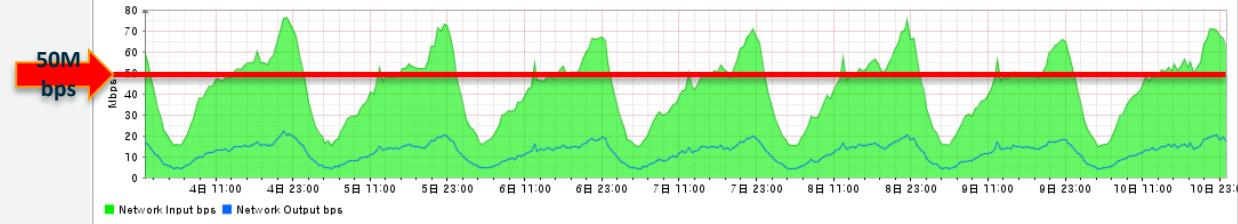
他 CDN から CloudFront に移行後オリジントラフィックが約 7 分の 1 に減少したお客様事例

1. 他 CDN 使用時 10/9 ~ 10/15

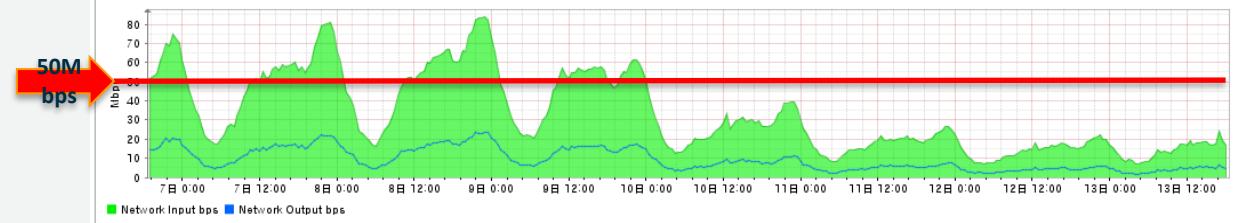


2. CFへMigration後

12/4 ~ 12/10

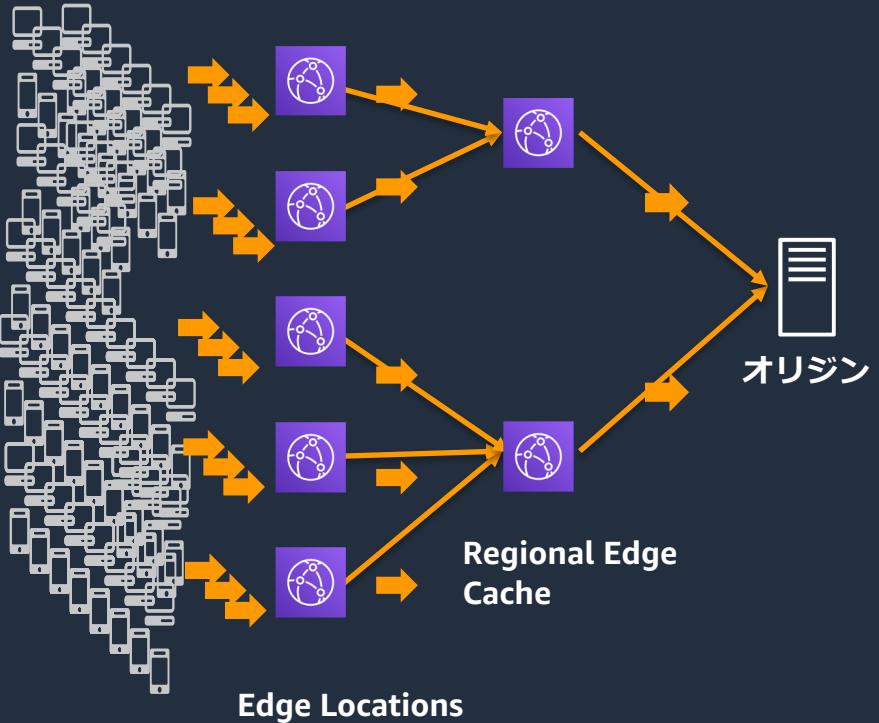


3. リージョン別エッジキャッシュ機能のリリース後 1/6 ~ 1/12



オリジンインフラの保護

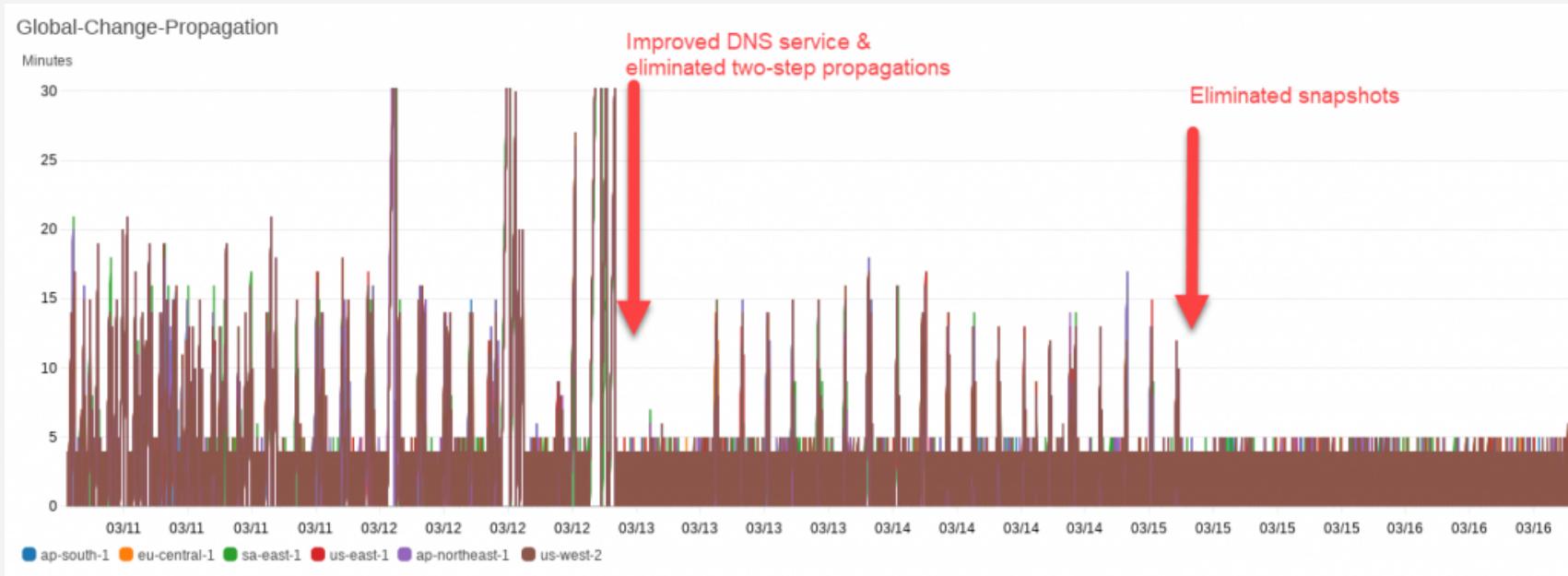
Automatic Flash Crowd Protection



- 同時に大量リクエストが発生（フラッシュクラウド/Flash Crowd）した場合、最初のリクエストのみをオリジンに送り、負荷低減を実現する仕組み
- オリジンが AWS にある場合は AWS Global Network を使用
- AWS 以外のオリジンでも同様の機能を提供

CloudFront の変更反映の改善

New



Slashing CloudFront change propagation times in 2020 – recent changes and looking forward

<https://aws.amazon.com/jp/blogs/networking-and-content-delivery/slashing-cloudfront-change-propagation-times-in-2020-recent-changes-and-looking-forward/>

CloudFront クオータの変更

New

エンティティ	変更前	変更後
ディストリビューションごとのデータ転送レート	40 Gbps	150 Gbps
1 秒あたり、ディストリビューションあたりのリクエスト	100,000 rps	250,000 rps

上記はデフォルトの制限値、チケットを起票しクオータ引き上げリクエストを行うことで更に大規模なトラフィックにも対応可能

Amazon CloudFront 開発者ガイド(クオータ):

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/cloudfront-limits.html

CloudFront deep dive

CloudFront 用語集



- **Viewer** (ビューワー): クライアント / Web ブラウザ
- **Distribution** (ディストリビューション): コンテンツ配信の設定単位、CloudFront ドメイン名、代替ドメイン名毎に作成
 - **Origin** (オリジン): コンテンツ提供元の Web サーバー毎に作成
 - カスタムオリジン: Amazon VPC やオンプレミスの Web サーバー
 - S3 オリジン: 静的コンテンツを提供する S3 バケット
 - **Behavior** (ビヘイビア): キャッシュ動作設定、 URL パスパターン毎に作成

※ Origin, Behavior は用途毎に複数設定が可能

CloudFront 設定

1. Distribution に関するリソースの準備と設定

- Route 53 ホストゾーン, AWS Certificate Manager (ACM) SSL/TLS 証明書, WAF Web ACL, ログ用 S3 バケット, CloudWatch メトリクス

2. Origin に関するリソースの準備と設定

- カスタムオリジンの Web サーバー
- S3 オリジンの S3 バケット, オリジンアクセスアイデンティティ (OAI)
- Origin Group

3. Behavior に関するリソースの準備と設定

- Cache Policy (キャッシュポリシー), Origin Request Policy (オリジンリクエストポリシー)
- Realtime Log config (リアルタイムログ): Amazon Kinesis Data Streams
- Key groups (署名付き URL, Cookie 用キー)
- Field-level encryption config (フィールドレベル暗号化設定)
- Lambda@Edge 関数

CloudFront 設定

続き

4. Distribution に関する機能

- Custom Error Responses: エラーレスポンス動作のカスタマイズ
- Restrictions: 特定の国のユーザーに対するアクセス制限
- Invalidation: キャッシュファイルの無効化

Distribution

CloudFront 設定

1. Distribution に関するリソースの準備と設定

- Route 53 ホストゾーン, AWS Certificate Manager (ACM) SSL/TLS 証明書, WAF Web ACL, ログ用 S3 バケット, CloudWatch メトリクス

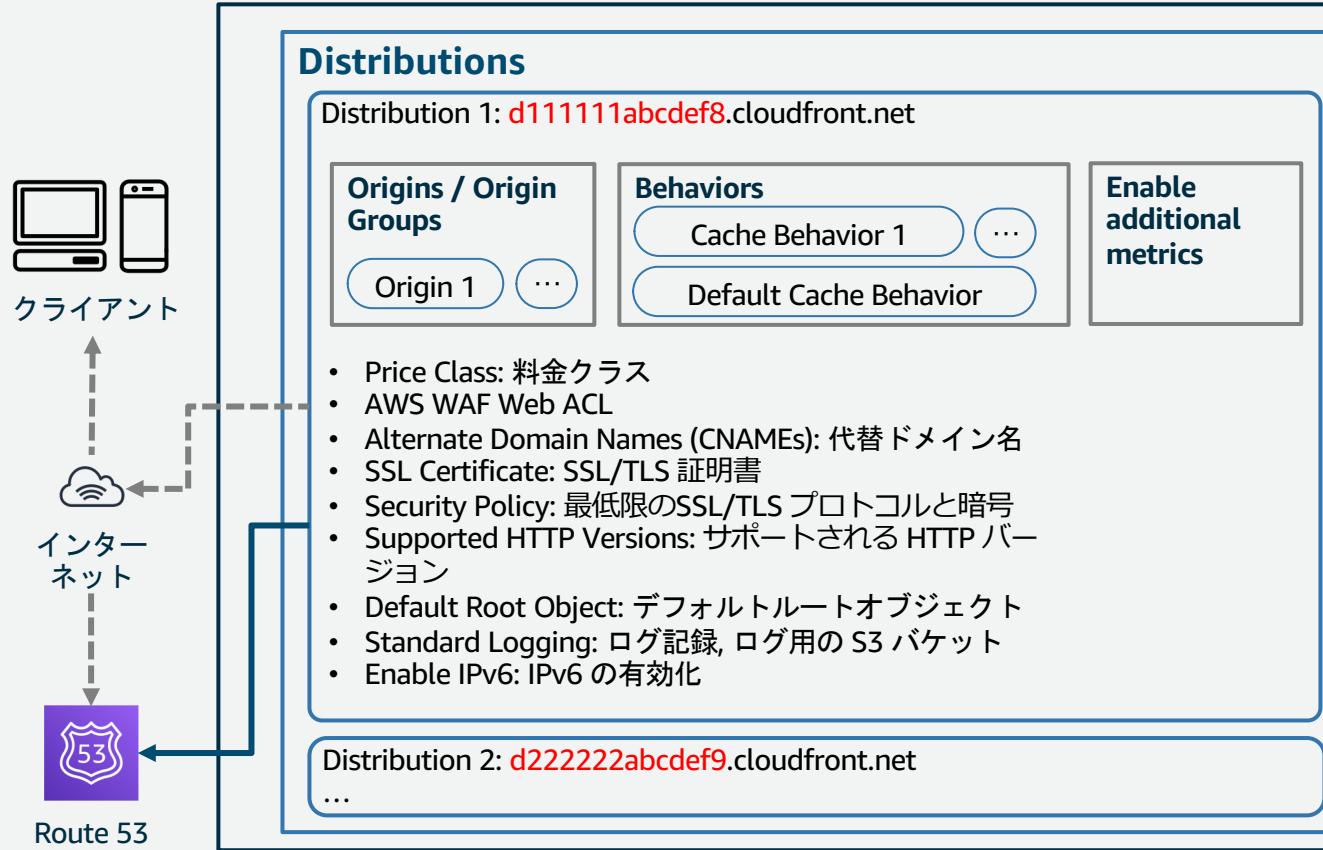
2. Origin に関するリソースの準備と設定

- カスタムオリジンの Web サーバー
- S3 オリジンの S3 バケット, オリジンアクセスアイデンティティ (OAI)
- Origin Group

3. Behavior に関するリソースの準備と設定

- Cache Policy (キャッシュポリシー), Origin Request Policy (オリジンリクエストポリシー)
- Realtime Log config (リアルタイムログ): Amazon Kinesis Data Streams
- Key groups (署名付き URL, Cookie 用キー)
- Field-level encryption config (フィールドレベル暗号化設定)
- Lambda@Edge 関数

Distribution 概要図



Certificate Manager



WAF



AWS Shield



S3
ログ



CloudWatch



Distribution 概要



- AWS Management Console もしくは API で**即時作成可能**
- HTTP/1.0, HTTP/1.1, HTTP/2, WebSocket 対応
 - HTTP/2 使用時はクライアントが TLS 1.2 以降と SNI (Server Name Identification) サポート必要
- **TLS 1.3** クライアント接続に対応、デフォルトで有効化 New
- IPv6 対応
- [ランダム文字列].cloudfront.net がドメイン名として割り当てられる
- CNAME エリアスを利用して代替ドメイン名の指定が可能
 - 有効な **SSL/TLS 証明書のコモンネームとの一致** が条件
 - ACM で無償の証明書を発行可能
 - ワイルドカード指定もサポート (例: *.example.com など)
 - Route53 Alias レコードと組み合わせた Zone Apex (例: example.com など) が利用可能

AWS WAF 連携



AWS WAF で定義した Web ACL を Distribution に適用

- CloudFront をサービスの前段に配置することでサイトの保護を実現
- AWS WAF での制御
 - XSS / GEO 制限 / IP アドレス制限 / サイズ制限 / SQLインジェクション / ヘッダー, クエリ, リクエストボディの文字列, 正規表現マッチング
 - WAFv2 のビルトインマネジドルール: AWS Managed Rules for AWS WAF**
 - パートナーのマネジドルール
- ブロック時は 403(FORBIDDEN) を応答



Edit Distribution

Distribution Settings

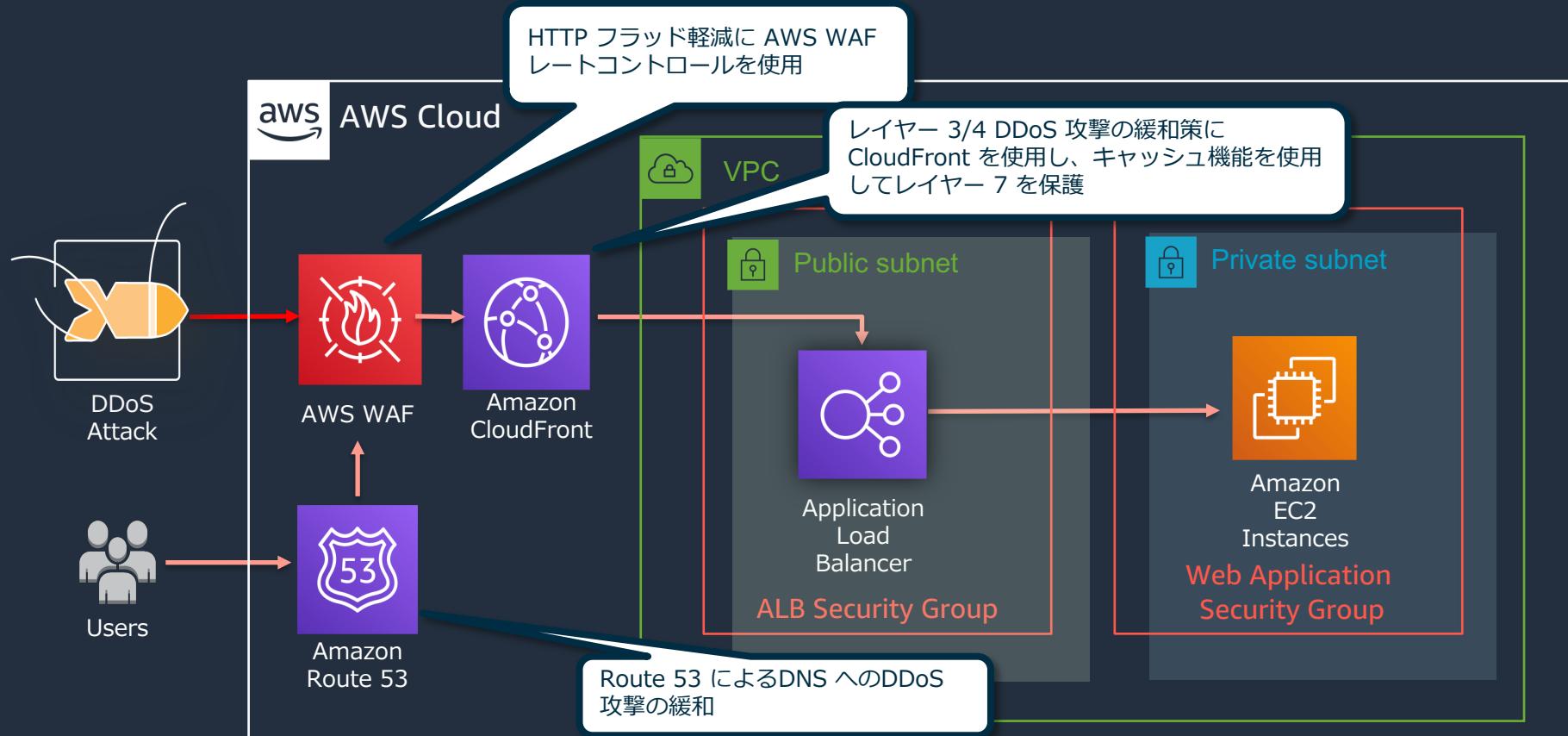
Price Class: Use All Edge Locations (Best Performance)

AWS WAF Web ACL: None

Alternate Domain Names (CNAMEs): .com

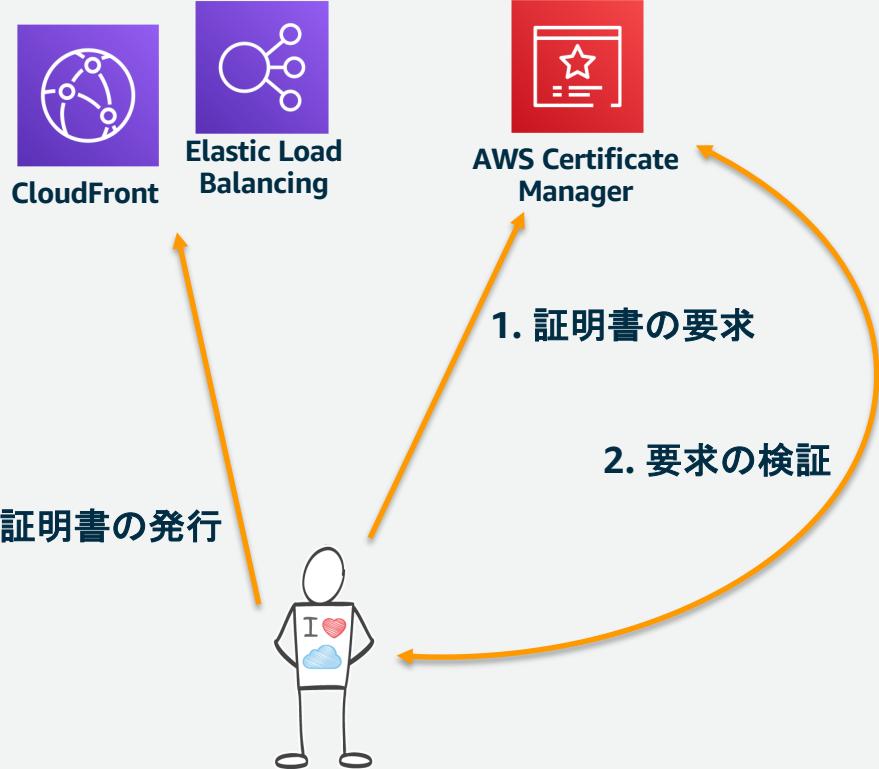
SSL Certificate: Default CloudFront Certificate (*.cloudfront.net)

DDoS 耐性の高いアーキテクチャ

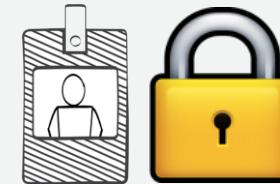


AWS Certificate Manager (ACM) との統合

- 新規の SNI 証明書を数分で発行し、CloudFront コンソールから直接デプロイ
- 生成された SNI 証明書は無償利用が可能
- 生成された SNI 証明書は自動更新が可能
- 既存証明書のインポートも可能



Viewer 接続 SSL セキュリティポリシー



Viewer と CloudFront 間の SSL/TLS バージョンと暗号の組み合わせを選択

New

TLsv1.2_2019(推奨), TLsv1.2_2018, TLsv1.1_2016, TLsv1_2016, TLsv1 から選択

New

すべてのポリシーで TLS1.3 が有効

- 代替ドメイン名を使用する独自 SSL 証明書の利用時のみ指定可能
 - SNI SSL 証明書は TLsv1 以降のみ指定可能
 - SSLv3 は専用 IP アドレス SSL 証明書のみ指定可能



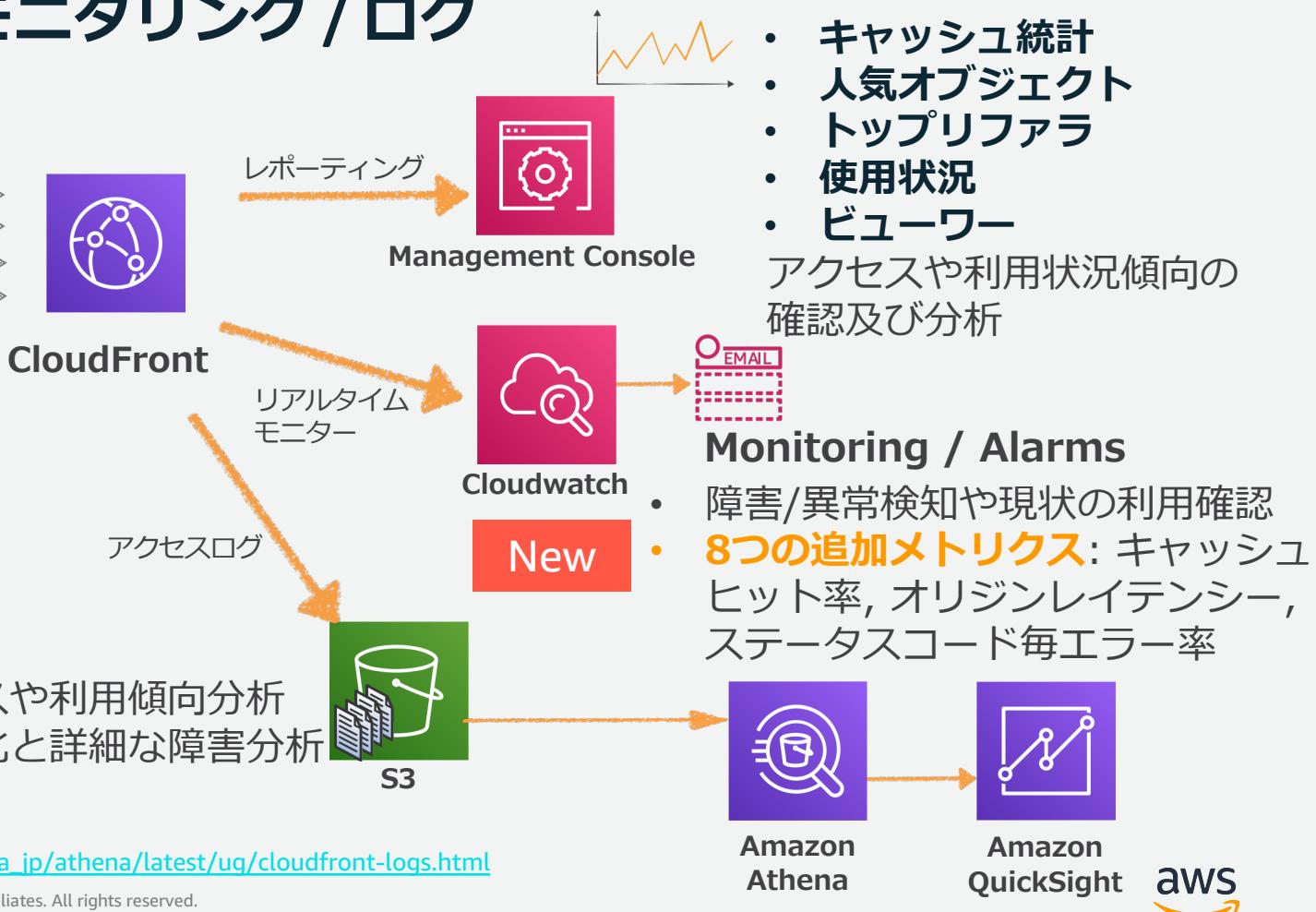
CloudFront エッジ

Security Policy

- TLsv1
- TLsv1_2016
- TLsv1.1_2016
- TLsv1.2_2018
- TLsv1.2_2019 (recommended)

See the [list of protocols and ciphers](#) that CloudFront uses for each security policy.

レポート / モニタリング / ログ



Origin

CloudFront 設定

1. Distribution に関するリソースの準備と設定

- Route 53 ホストゾーン, ACM SSL/TLS 証明書, WAF Web ACL, ログ用 S3 バケット, CloudWatch メトリクス

2. Origin に関するリソースの準備と設定

- カスタムオリジンの Web サーバー
- S3 オリジンの S3 バケット, オリジンアクセスアイデンティティ (OAI)
- Origin Group

3. Behavior に関するリソースの準備と設定

- Cache Policy (キャッシュポリシー), Origin Request Policy (オリジンリクエストポリシー)
- Realtime Log config (リアルタイムログ): Amazon Kinesis Data Streams
- Key groups (署名付き URL, Cookie 用キー)
- Field-level encryption config (フィールドレベル暗号化設定)
- Lambda@Edge 関数

Origin 概要図

Distribution 1: d111111abcdef8.cloudfront.net

Origins

カスタムオリジン 1

- Origin Domain Name: ドメイン名
 - Origin Path: パス
 - Enable Origin Shield: Origin Shield の有効化
 - Origin Connection Attempts: オリジン接続の試行回数
 - Origin Connection Timeout: オリジン接続タイムアウト
 - Origin Custom Headers: オリジンカスタムヘッダー
- ※ ここまで S3 オリジンと共通
- Minimum Origin SSL Protocol: 最低限の SSL/TLS プロトコル
 - Origin Protocol Policy: オリジンプロトコルポリシー
 - Origin Response Timeout: オリジン応答タイムアウト
 - Origin Keep-alive Timeout: オリジン持続的接続のタイムアウト
 - HTTP Port: HTTP ポート
 - HTTPS Port: HTTPS ポート

カスタムオリジン 2

S3 オリジン 1

- Origin Access Identity: OAI

Origin Groups

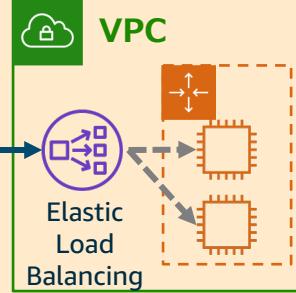
- オリジン
グループ 1
- Origins
 - Failover criteria

...

OAI

OAI 1

カスタムオリジン 1



カスタムオリジン 2



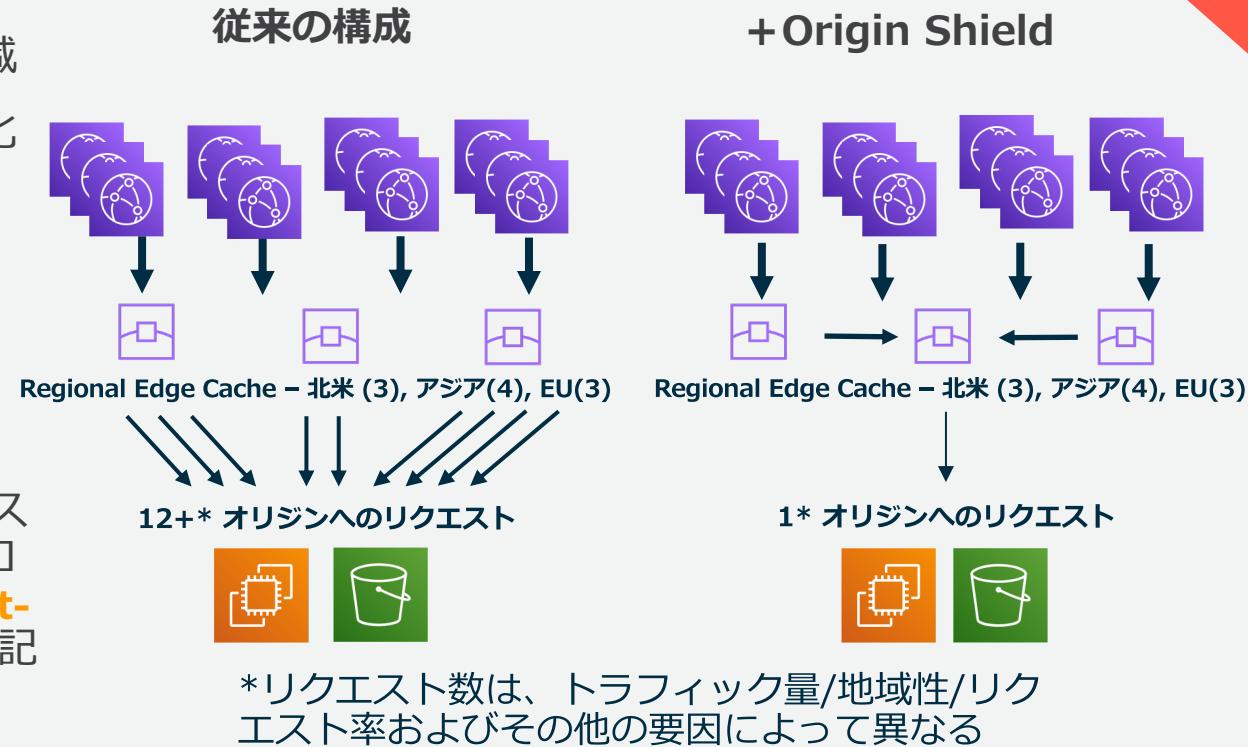
S3 オリジン 1



Origin Shield

New

- ・オリジンの負荷をさらに低減
- ・オリジン関連のコスト最適化
 - ・リクエスト数の削減
 - ・データ転送量の削減 他
- ・ユーザー視聴体験の向上
- ・キャッシュ効率の向上
- ・Origin Shield からレスポンスが返された場合はアクセスログの **x-edge-detailed-result-type** に **OriginShieldHit** が記録される



Amazon CloudFront 開発者ガイド(Using Amazon CloudFront Origin Shield):

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/origin-shield.html

Origin Shield の料金

アクセス先 REC と Origin Shield が同一リージョンの場合は無料
他リージョンの REC から Origin Shield へのアクセスは 1 万リクエスト
毎に料金が発生する

Origin Shield 料金 (1 万リクエスト毎)

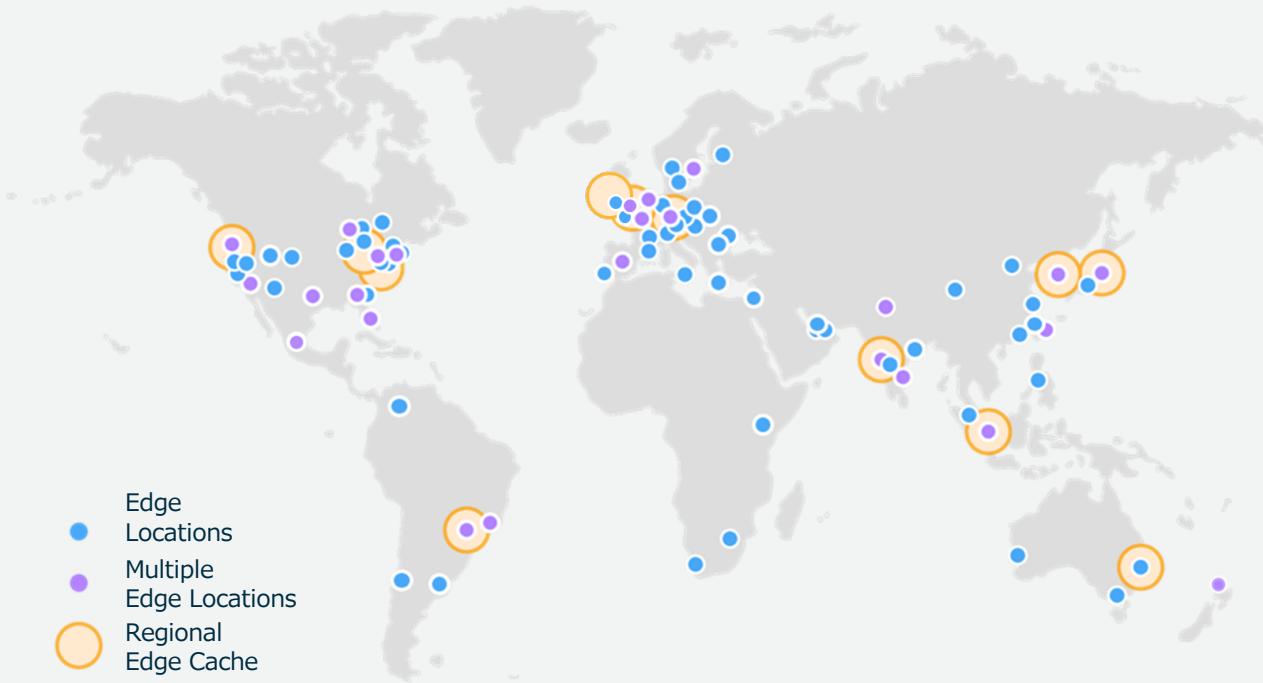
米国	欧州	南アメリカ	日本	オーストラリア	シンガポール	韓国	インド
\$0.0075	\$0.0090	\$0.0160	\$0.0090	\$0.0090	\$0.0090	\$0.0090	\$0.0090

例: Origin Shield が Tokyo リージョンの場合、日本のエッジロケーション経由の追加コストは発生しない

Origin Shield ロケーション

As of 10/28/2020

選択可能な Origin Shield
ロケーション



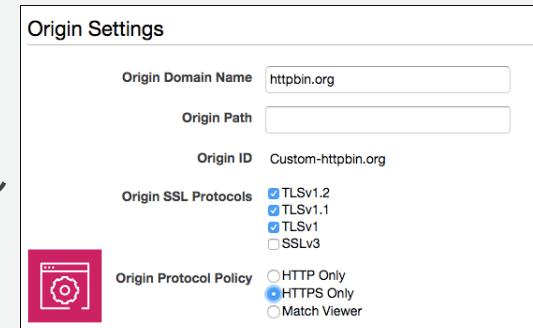
- US East (N. Virginia)
- US East (Ohio)
- US West (Oregon)
- South America (Sao Paulo)
- EU (Ireland)
- EU (London)
- EU (Frankfurt),
- Asia Pacific (Seoul)
- **Asia Pacific (Tokyo)**
- Asia Pacific (Singapore)
- Asia Pacific (Mumbai)
- Asia Pacific (Sydney)

カスタムオリジンの通信ポリシー

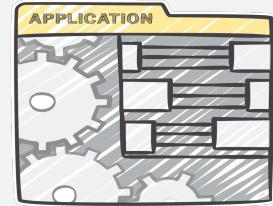


CloudFront エッジとオリジン間の通信方式を制御

- SSL/TLS プロトコル方式
 - TLSv1.2, TLSv1.1, TLSv1, SSLv3 から複数指定可能
- オリジンとの通信プロトコル
 - HTTP のみ、HTTPS のみ、クライアントからの通信プロトコルに合わせる
- S3 オリジンは標準で HTTPS を利用



カスタムオリジンのタイムアウト



オリジンの応答タイムアウト

- CloudFront がカスタムオリジンからの応答を待つ時間を指定
- ビジー状態の負荷を軽減したり、Viewer にエラー応答をより迅速に表示したりする場合は、応答タイムアウトを小さくする
- デフォルトのタイムアウトは 30 秒、4~60 秒の範囲で設定可能**

持続的接続のタイムアウト

- 接続を閉じる前に CloudFront がカスタムオリジンサーバーとの持続的接続を維持する最大時間を指定
- デフォルトの Keep-alive Timeout は5秒、1~60秒の範囲で設定可能

オリジンカスタムヘッダー



オリジンへの通信時にカスタム HTTP ヘッダーを追加

- オリジン毎に固定ヘッダーの追加もしくは、クライアントからのリクエストヘッダーの上書きが可能
- Shared-Secret
 - CloudFront とオリジン間で任意の HTTP ヘッダーおよび値を取り決め、オリジン側でヘッダー値のチェックを行うことで、カスタムオリジンは CloudFrontからのアクセスのみに制御する

Origin Custom Headers	Header Name	Value	⋮
	X-CloudFront-Distribution-Id	123	×
	X-Shared-Secret	cf9db9688fff28c2624fd3a321948c51	+



オリジンサーバの保護

S3 オリジン

- Origin Access Identity(OAI) を利用
 - S3 バケットへのアクセスを CloudFront からのみに制限

カスタムオリジンは下記の2種類が選択可能

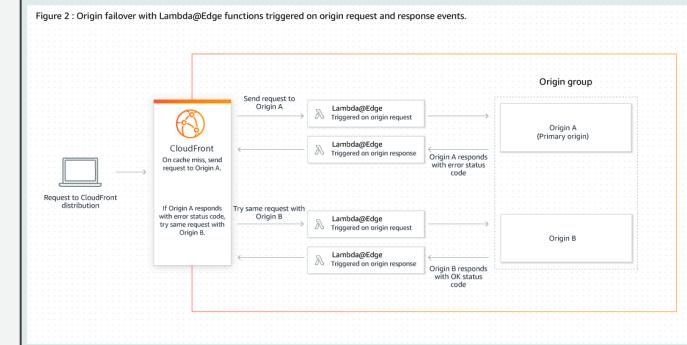
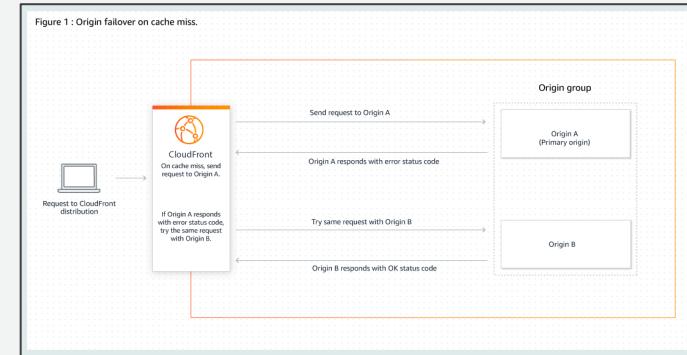
- オリジンカスタムヘッダーを利用し、指定された任意のヘッダーをオリジン側でチェック
 - **ALB のカスタムルールにて、HTTP ヘッダーのチェックが可能**
- オリジン側のアドレスを公開しないとともに、CloudFront が利用する IP アドレスのみの許可させる
 - CloudFront が利用する IP アドレスは下記 URL から取得可能: <https://ip-ranges.amazonaws.com/ip-ranges.json>
 - JSON フォーマット: Service キーの “CLOUDFRONT” でフィルタすることで抽出可能



Origin Group によるオリジンフェイルオーバー

オリジンの高可用性を実現

- オリジングループを作成し、プライマリ・セカンダリオリジンを指定
- フェイルオーバー基準: オリiginがフェイルオーバー用に設定した 500, 502, 503 等の HTTP ステータスコードを返した場合や、接続タイムアウト/接続試行回数を超過/応答タイムアウトした場合にバックアップオリジンにルーティング
- Lambda@Edge 関数やカスタムエラーページでもオリジンフェイルオーバーが可能



Behavior

CloudFront 設定

1. Distribution に関するリソースの準備と設定

- Route 53 ホストゾーン, ACM SSL/TLS 証明書, WAF Web ACL, ログ用 S3 バケット, CloudWatch メトリクス

2. Origin に関するリソースの準備と設定

- カスタムオリジンの Web サーバー
- S3 オリジンの S3 バケット, オリジンアクセスアイデンティティ (OAI)
- Origin Group

3. Behavior に関するリソースの準備と設定

- Cache Policy (キャッシュポリシー), Origin Request Policy (オリジンリクエストポリシー)
- Realtime Log config (リアルタイムログ): Amazon Kinesis Data Streams
- Key groups (署名付き URL, Cookie 用キー)
- Field-level encryption config (フィールドレベル暗号化設定)
- Lambda@Edge 関数

Cache Policy / Origin Request Policy / Behavior 概要図



Cache Policy

- Min TTL: 最小TTL
- Max TTL: 最大TTL
- Default TTL: デフォルトTTL
- Headers: キャッシュキー HTTP ヘッダー
- Cookies: キャッシュキー Cookie
- Query Strings: キャッシュキークエリ文字列
- Gzip: Gzip 圧縮サポート
- Brotli: Brotli 圧縮サポート

Managed-*: マネージドキャッシュポリシー

Origin Request Policy

オリジンリクエストポリシー 1

- Headers: オリigin転送 HTTP ヘッダー
- Cookies: オリigin転送 Cookie
- Query Strings: オリigin転送クエリ文字列

Managed-*: マネージドオリジンリクエスト
ポリシー

Distribution 1: d111111abcdef8.cloudfront.net

Behaviors

Cache Behavior 1: api/item* → カスタムオリジン 1

- Path Pattern: URL パスパターン
- Target Origin or Origin Group: オリジン/オリジングループ
- Viewer Protocol Policy: ビューワープロトコルポリシー
- Allowed HTTP Methods: 許可される HTTP メソッド
- Cache Policy: キャッシュポリシー
- Origin Request Policy: オリジンリクエストポリシー
- Compress Objects Automatically: オブジェクトを自動的に圧縮する

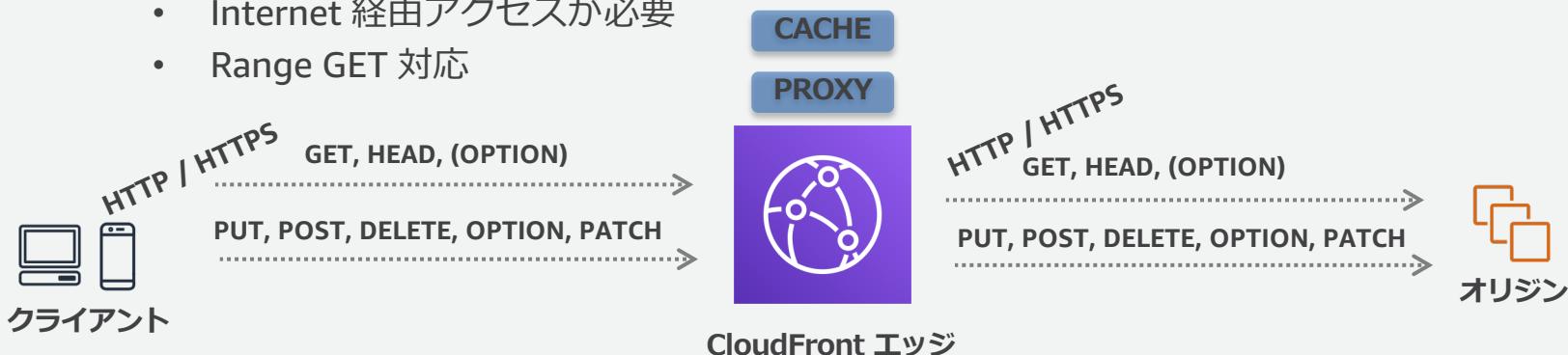
Cache Behavior 2: img/* → S3 オリジン 1

Default Cache Behavior: * → カスタムオリジン 1

Behavior: プロトコルポリシー / HTTP メソッド



- Viewer プロトコルポリシー
 - HTTP / HTTPS
 - HTTP から HTTPS への Redirect
 - HTTPS のみ
- 許可される HTTP メソッド
 - キャッシュモード: GET, HEAD, OPTION (選択可能)
 - プロキシモード: GET, HEAD, OPTION, PUT, POST, PATCH, DELETE
- オリジンへのアクセス
 - Internet 経由アクセスが必要
 - Range GET 対応



Behavior: キャッシュコントロール機能



キャッシュコントロール

キャッシュヒット率を向上させることが CDN 導入におけるポイント

- GET / HEAD / OPTION (選択可能) のリクエストがキャッシュ対象
- 単一リクエスト (FULL または RANGE GET) のキャッシングは最大 20GB まで

URL および Cache Policy で有効化した HTTPヘッダー, クエリ文字列, Cookie パラメータ値の**完全一致**でキャッシュが再利用される

Cache Policy / Origin Request Policy

New

- オリジンに転送するリクエストとキャッシュキーを分離して取り扱うことにより、より柔軟なキャッシング設定が可能に
 - 従来のインターフェイスも継続して利用可能
- 事前定義済みのマネージドポリシーの他に、カスタムポリシーの作成・適用が可能
- より高いキャッシング効率が実現可能

The screenshot shows the AWS CloudFront Policies page. A red oval highlights the 'Policies' link in the left sidebar. The main area displays a table of Cache policies with columns for Policy name, Policy comment, Last modified, Distributions, and Policy ID. Policies listed include Canned-DefaultCachePolicy, Canned-DisabledCachePolicy, DefaultSetting, NoCache, and PolicyLinked.

Policy name	Policy comment	Last modified	Distributions	Policy ID
Canned-DefaultCachePolicy	Preconfigured default cache policy	-	5	658327ea-f89d-4fb-a535-7e8893e58f5
Canned-DisabledCachePolicy	Preconfigured cache policy	-	3	4135ea2d-6dfb-44b3-9d3-4b5a54be3fad
DefaultSetting	Default policy	-	6	1aa49186-2160-4e93-b4fa-2e5caead53e
NoCache	Disabled cache key	-	2	5712a626-742d-47c8-8209-71cc0121d9e
PolicyLinked	Test	-	3	f5ca1280-e8b6-42f0-94c8-8394284c1014



Cache Policy
(キャッシングキーに含めるヘッダー
やクエリ文字列、TTLなどを定義)



Amazon
CloudFront



Origin Request Policy
(オリジンが一意のコンテンツを作る
のに必要なヘッダーやクエリ文字列な
どを定義)



オリジンサーバ
(AWS Region or
Custom Origin)

Amazon CloudFront 開発者ガイド(ポリシーの使用):

https://docs.aws.amazon.com/ja_ip/AmazonCloudFront/latest/DeveloperGuide/working-with-policies.html

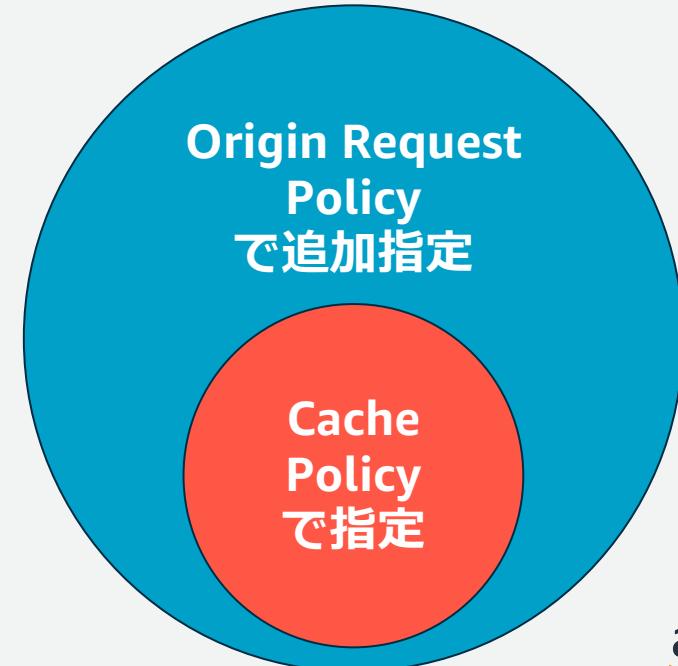
© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



Policy 使用時のオリジンリクエストの制御

- Cache Policy を使用したキャッシュキーに含めるすべての HTTP ヘッダー, Cookie, URL クエリ文字列は、**オリジンリクエストに自動的に含まれる**
- Origin Request Policy では、オリジンリクエストに含めるが、キャッシュキーには含めないデータを指定
- クエリ文字列以外の URL パス、リクエスト Body、Host, User-Agent: “**Amazon CloudFront**”, X-Amz-Cf-Id ヘッダは自動的に付与

オリジンリクエストに含まれる
HTTPヘッダー, クエリ文字列, Cookie



Cache Policy / Origin Request Policy の使用例

キャッシュキーに User-Agent と Referer を含めずに、オリジンリクエストへ転送

Behavior の Cache Policy:

- カスタム Cache Policy のキャッシュキー
HTTP ヘッダー: **Accept-Language**

Viewer の HTTP リクエスト

```
GET /content/stories/example-story.html
Host: d111111abcdef8.cloudfront.net
User-Agent: Mozilla/5.0 Gecko/20100101 Firefox/68.0
Accept: text/html, */*
Accept-Language: ja,en-US,en
Referer: https://news.example.com/
```

オリジンに転送された HTTP リクエスト

Behavior の Origin Reuest Policy:

- カスタム Origin Request Policy のオリジン転送 HTTP ヘッダー: **指定なし**

```
GET /content/stories/example-story.html
Host: cf-backend.example.com
User-Agent: Amazon CloudFront
X-Amzn-Trace-Id: Root=1-11111111-123456789abcdefghijklmnno
Accept-Language: ja,en-US,en
```

Behavior の Origin Reuest Policy:

- Managed-UserAgentRefererHeaders**

```
GET /content/stories/example-story.html
Host: cf-backend.example.com
User-Agent: Mozilla/5.0 Gecko/20100101 Firefox/68.0
Referer: https://news.example.com/
X-Amzn-Trace-Id: Root=1-22222222-123456789abcdefghijklmnno
Accept-Language: ja,en-US,en
```

Cache Policy: Cache-Control ヘッダー



- オリジンの Cache-Control ヘッダーでキャッシュ時間の設定が可能
- オリジンが Cache-Control ヘッダーを付与しない場合でも上書きが可能
- Behavior 每に異なる設定を行うことで、URL パスパターン毎にキャッシュ期間を変えることが可能
 - デフォルト TTL : オリiginが Cache-Control ヘッダーを指定しない場合に利用(デフォルト 24 時間)
 - 最小 TTL : CloudFront でキャッシュすべき最小期間
 - 最大 TTL : CloudFront でキャッシュすべき最大期間

Cache Policy Minimum TTL 設定					
オリジン HTTP ヘッダー	最小 TTL = 0 秒		最小 TTL > 0 秒を設定		
	Cache-Control max-age を指定	指定された max-age と最大 TTL で小さい値の期間キャッシュ	最小 TTL < max-age < 最大 TTL	max-age 期間	最小 TTL 期間
	Cache-Control 設定なし	デフォルト TTL 期間キャッシュ (標準 24 時間)	max-age < 最小 TTL	最大 TTL 期間	最大 TTL 期間
			最小 TTL またはデフォルト TTL で大きい値の期間キャッシュ		

Cache Policy: Cache-Control ヘッダー



オリジン HTTP ヘッダー

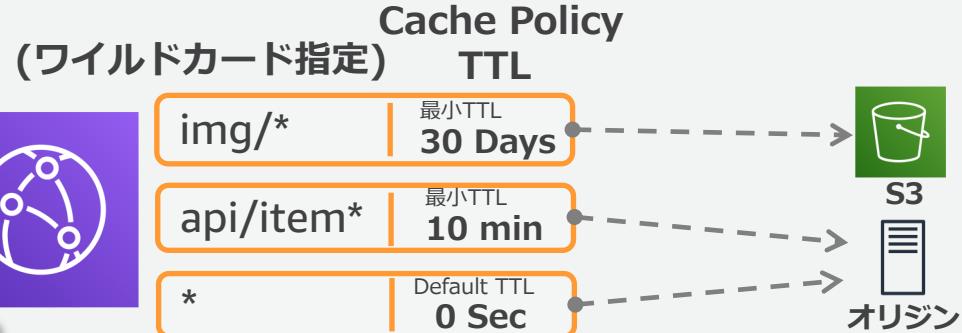
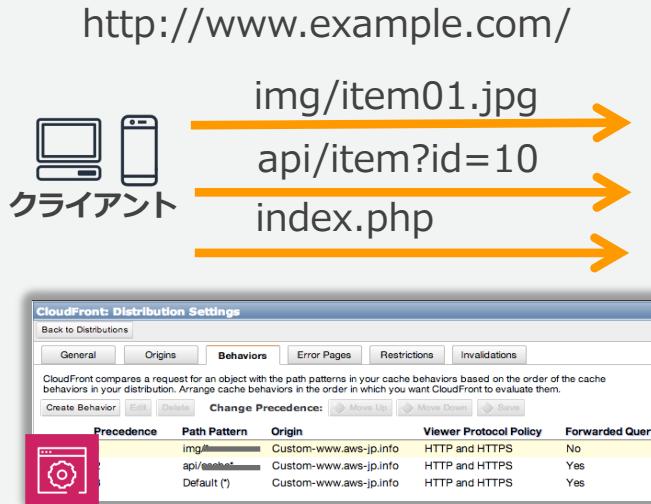
Cache Policy Minimum TTL 設定

	最小TTL = 0秒	最小TTL >0秒を設定
Cache-Control max-age と s-maxage を指定	指定された s-max-age と最大 TTL で 小さい値の期間キャッシング	最小TTL < s-max-age < 最大 TTL s-max-age < 最小 TTL 最大 TTL < s-max-age
Expires を指定	指定された Expires 日付と最大 TTL で 早い日付の期間キャッシング	最小 TTL << 最大 TTL Expires < 最小 TTL 最大 TTL < Expires
Cache-Control no-cache, no-store を指定	キャッシングされない	最小 TTL の期間キャッシング

- S3 オリジンの場合は S3 オブジェクト Metadata に Cache-Control, Expires を指定可能
- HTML Meta タグに Cache-Control もしくは Pragma を指定しても CloudFront は利用しない

きめ細やかなキャッシングの実現

- Cache Policy / Origin Request Policy を組み合わせ、HTTP ヘッダー, Cookie, クエリ文字列をオリジンリクエストへ含めることで、**動的コンテンツ**の配信に対応
- クライアントのリクエストパターンをもとに、**複数の URL パスパターンの Behavior** と **マルチオリジン**を組み合わせ、**きめ細かなキャッシングコントロール**を実現



Behaviors Path Pattern の記述方法

- 「*」 0もしくはそれ以上の文字列
 - 「?」 1文字
- 例) /*.jpg, /image/*, /image/a*.jpg, /a???.jpg

CloudFront Header の拡張

New

- デバイスや地域情報の取得に使われていた CloudFront Header が拡張
- Cache Policy / Origin Request Policy でも従来のインターフェイスでも利用が可能

デバイス情報 Header 例

- CloudFront-Is-Android-Viewer
- CloudFront-Is-Desktop-Viewer
- CloudFront-Is-IOS-Viewer
- CloudFront-Is-Mobile-Viewer
- CloudFront-Is-SmartTV-Viewer
- CloudFront-Is-Tablet-Viewer

地域情報 Header 例

- CloudFront-Viewer-City
- CloudFront-Viewer-Country
- CloudFront-Viewer-Country-Name
- CloudFront-Viewer-Country-Region
- CloudFront-Viewer-Country-Region-Name
- CloudFront-Viewer-Latitude
- CloudFront-Viewer-Longitude
- CloudFront-Viewer-Metro-Code ※ US のみ
- CloudFront-Viewer-Postal-Code
- CloudFront-Viewer-Time-Zone

Amazon CloudFront 開発者ガイド(CloudFront HTTP ヘッダーを使用する):

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/using-cloudfront-headers.html

Behavior, Cache Policy: エッジでの Gzip, Brotli 圧縮

New

- Cache Policy の Gzip, Brotli (ブレトリ) を有効に、各 TTL は 1 以上に設定
- Behavior の Compress Objects Automatically を有効に
- リクエストヘッダーに Accept-Encoding:gzip, br が指定されており、オリジンがレスポンス圧縮に対応していない場合は、CloudFront エッジにて Gzip, Brotli 圧縮を行い配信
- S3 はレスポンス圧縮をサポートしていないため、有効なオプション



Behavior 概要図

Distribution 1: d111111abcdef8.cloudfront.net

Behaviors

Cache Behavior 1: api/item* → カスタムオリジン 1

- Enable Real-time Logs: リアルタイムログの有効化
- Restrict Viewer Access: 署名付き URL, Cookie の使用
 - Trusted Key Groups: Key Group の指定
- Field-level Encryption Config: フィールドレベル暗号化の設定
- Lambda Function Associations: Lambda の ARN 関連付け
 - Viewer Request: ビューワーリクエストの Lambda
 - Viewer Response: ビューワーレスポンスの Lambda
 - Origin Request: オリジンリクエストの Lambda
 - Origin Response: オリジンレスポンスの Lambda

Cache Behavior 2: img/* → S3 オリジン 1

Default Cache Behavior: * → カスタムオリジン 1

Realtime Log Config

ログ 1

- Sampling Rate: ログサンプリングレートの %
- End Points: Kinesis の ARN
- Fields: ログフィールド



Kinesis Data Streams

Key groups

Key group 1

- Public keys: パブリックキー



Field Level Encryption Config

Field Level Encryption Config 1

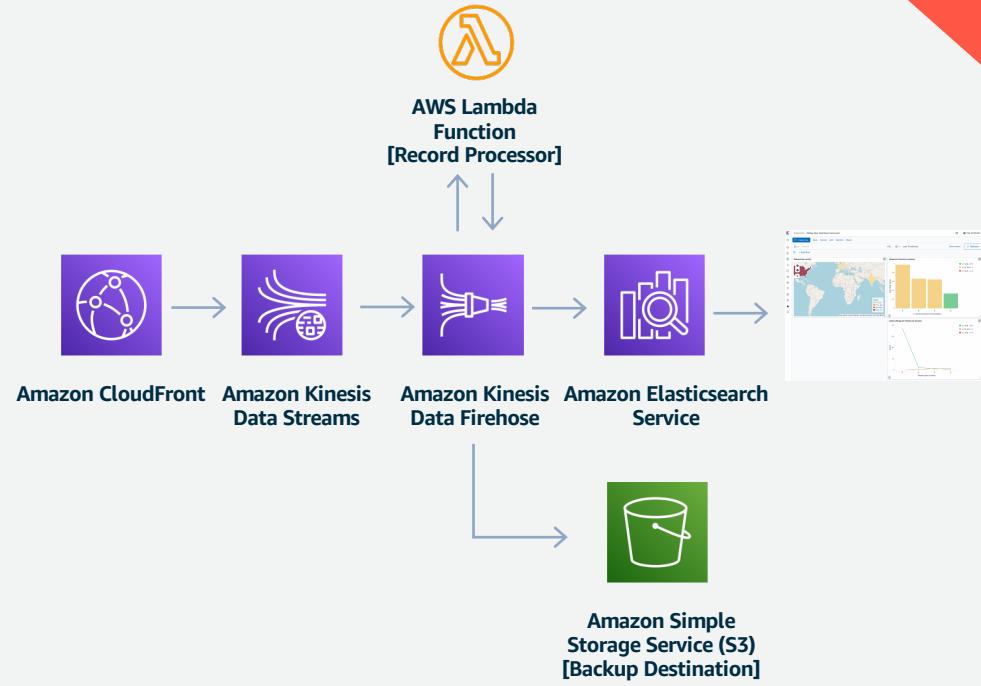
- Public keys: パブリックキー
- Provider Name: プロバイダ名
- Field name pattern to match: 暗号化フィールド



リアルタイムログ

New

- Kinesis Data Streams 経由で 1 分以内のニアリアルタイムでログ処理が可能
- サンプリングレートとログフィールドを選択可能
- Amazon Kinesis Data Firehose 経由で Amazon S3, Amazon Redshift, Amazon Elasticsearch Service および、サードパーティのログ処理サービスにログを配信可能



Amazon CloudFront 開発者ガイド(リアルタイムログ):

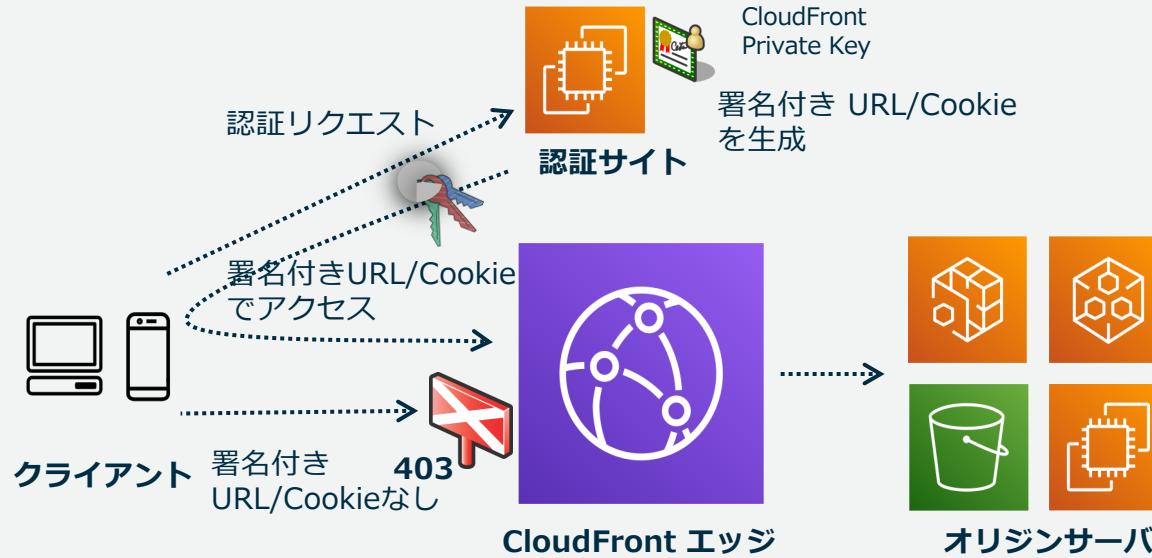
https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/real-time-logs.html

Amazon CloudFront ログを使用したリアルタイムダッシュボードの作成

<https://aws.amazon.com/jp/blogs/news/cloudfront-realtime-dashboard/>

署名付き URL / 署名付き Cookie

New

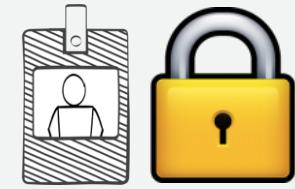


- IAM アカウントで署名付き URL / 署名付き Cookie のキー設定が可能に
- 単一コンテンツアクセスの場合は署名付き URL、HLS 動画配信などの複数コンテンツアクセスの場合は、署名付き Cookie の利用を推奨

Amazon CloudFront 開発者ガイド(Choosing between trusted key groups (recommended) and AWS accounts):

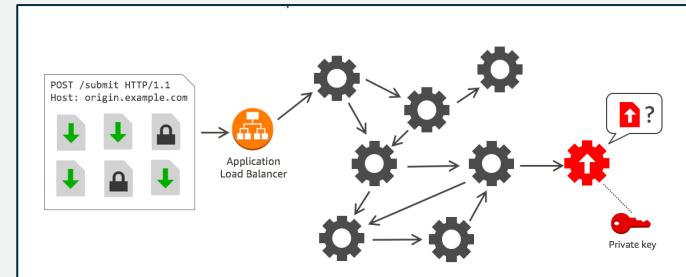
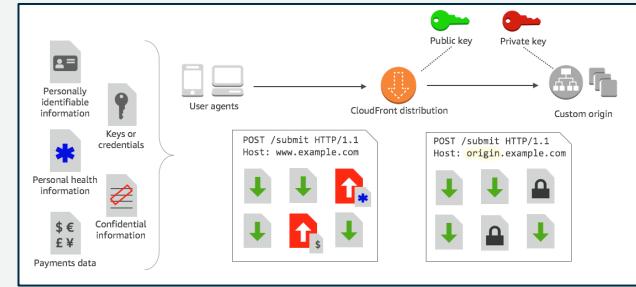
<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/private-content-trusted-signers.html#choosing-key-groups-or-AWS-accounts>

フィールドレベル暗号化



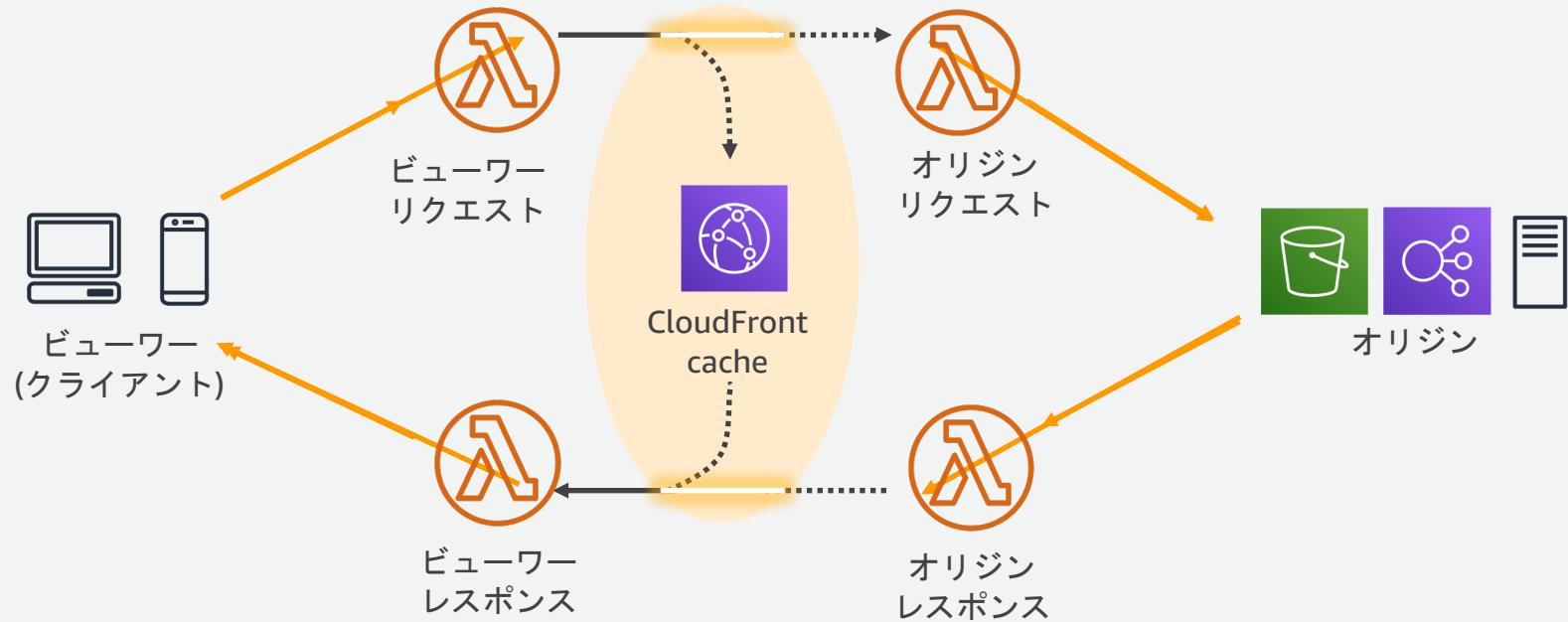
POST リクエストの特定データフィールドを特定のアプリケーションのみアクセスできるように保護

- 公開鍵暗号方式
- 設定方法
 1. RSA キーペアを取得
 2. パブリックキーを CloudFront に追加
 3. フィールドレベル暗号化のプロファイルを作成
 4. 暗号化を行うリクエストのコンテンツタイプを指定する設定を作成
 5. Behavior に設定を追加
 6. オリジンでデータフィールドを復号化
 - AWS Encryption SDK を使用
 - C, Java, Python, JavaScript, CLI を使用可能



Lambda@Edge イベント

Lambda 関数を使用して CloudFront リクエストとレスポンスを変更



Lambda 関数をトリガーできる CloudFront イベント

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/lambda-cloudfront-trigger-events.html

Lambda@Edge のプログラミングモデル

イベント・ドリブン

- 関数はイベントに関連付けられる
 - viewer-request -> my_function:1
- 関数はイベント発生時に実行される
 - viewer-request は CloudFront がリクエストを受信した時に実行される
- 関数は入力イベントの内容を受け取って実行される
 - my_function:1 はリクエストオブジェクトを受け取って実行される
- 関数は呼び出し元に変更した結果を返す必要がある
 - callback(null, request)

リクエストイベントごとの機能

ビューウー

- Header 読み取り/書き込み
- URL 読み取り/書き込み
- クエリ文字列 読み取り/書き込み
- Request Body 読み取り
- Response 生成
- Network 呼び出し

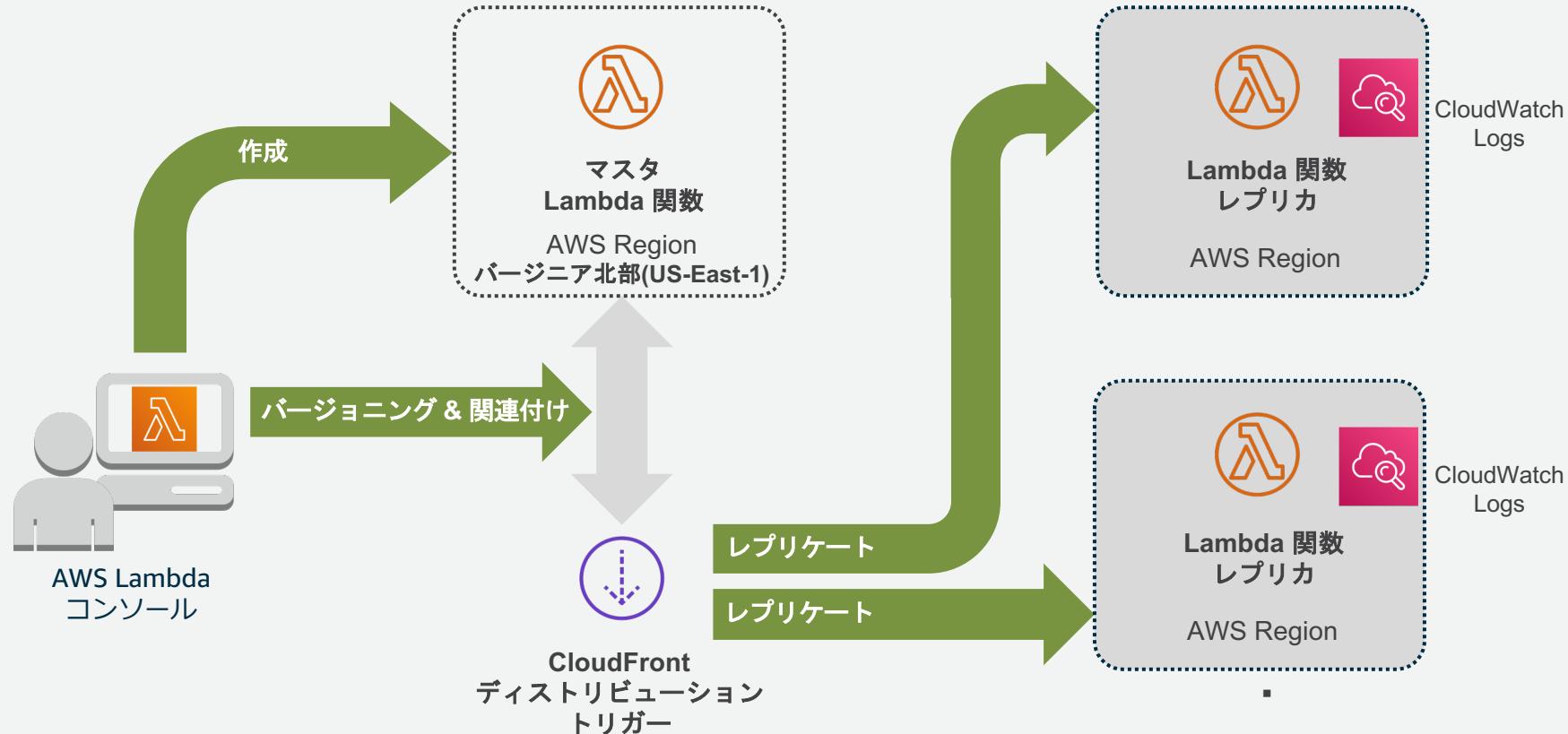
リクエスト

- Header 読み取り/書き込み
- Request Body 読み取り
- URL 読み取り/書き込み
- クエリ文字列 読み取り/書き込み
- CloudFront-* 追加 Header 読み取り
- バイナリを含む Response 生成
- Network 呼び出し
- S3オリジン,カスタムオリジンの変更
- 関数タイムアウト 30 秒

オリジン

- Header 読み取り/書き込み
- Request object 読み取り
- Network 呼び出し
- Header 読み取り/書き込み
- Request object 読み取り
- エラーステータス時の Response 更新
- Network 呼び出し
- 関数タイムアウト 30 秒

Lambda@Edge 用 Lambda 関数のデプロイフロー



Lambda@Edge 関数の作成と使用の開始

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/lambda-edge-how-it-works.html

テストとデバッグ

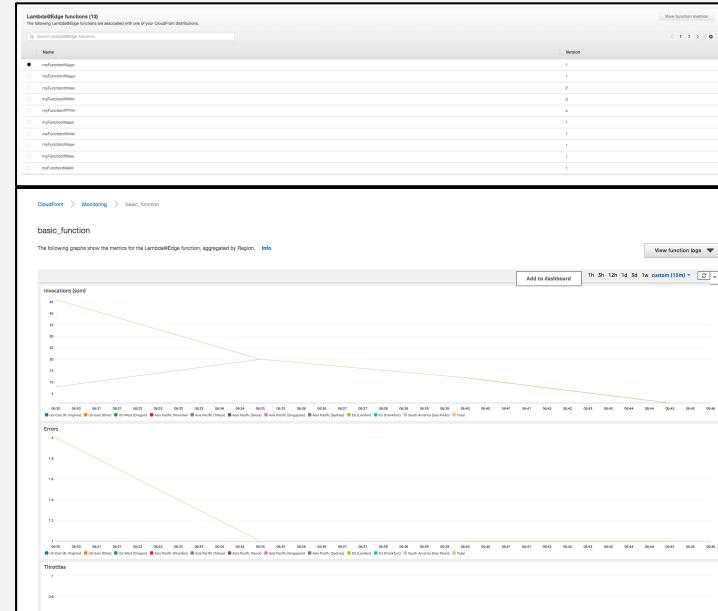
- ・ 「Lambda@Edge 関数のテストとデバッグ」 のドキュメントを確認
 - ・ <https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/lambda-edge-testing-debugging.html>
- ・ CloudFront エラーレスポンスの X-Cache ヘッダーを確認
 - ・ HTTP 502 Status Code (X-Cache: LambdaValidationError from CloudFront)
 - ・ HTTP 500 Status Code (X-Cache: LambdaExecutionError from CloudFront)
 - ・ HTTP 503 Status Code (X-Cache: LambdaLimitExceeded from CloudFront)
- ・ CloudFront アクセスログの確認 (x-edge-result-type)
 - ・ LambdaValidationError
 - ・ LambdaExecutionError
 - ・ LambdaLimitExceeded
- ・ CloudWatch Lambda@Edge 関数メトリクス (後述) を確認

CloudWatch Lambda@Edge 関数メトリクス

CloudFront Reports & Analytics の
Monitoring から、Lambda@Edge 関数の
メトリクスを確認

全リージョン Lambda@Edge 関数の
CloudWatch メトリクスを一覧で確認可能

- Invocations
- Errors
- Throttles
- Success rate
- Duration



Lambda@Edge 実行環境

		オリジン	ビューワー
ランタイム	New	Node.js 12.x & Python 3.8	←
メモリ		Lambda と同じ	128 MB
関数タイムアウト		30 秒	5 秒
Lambda 関数および組み込みライブラリの最大圧縮サイズ		50 MB	1 MB
レスポンスサイズ (request events)		1 MB	40 KB
同時実行数のデフォルト (Region毎) ※上限緩和可能		Lambda と同じ ※ Tokyo Region: 1,000	←
/tmp, 環境変数, DLQ, VPC, Layer, X-Ray		使用不可	←

Lambda@Edge 関数がサポートするランタイムと設定

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/lambda-requirements-limits.html#lambda-requirements-lambda-function-configuration

Distribution に関する機能

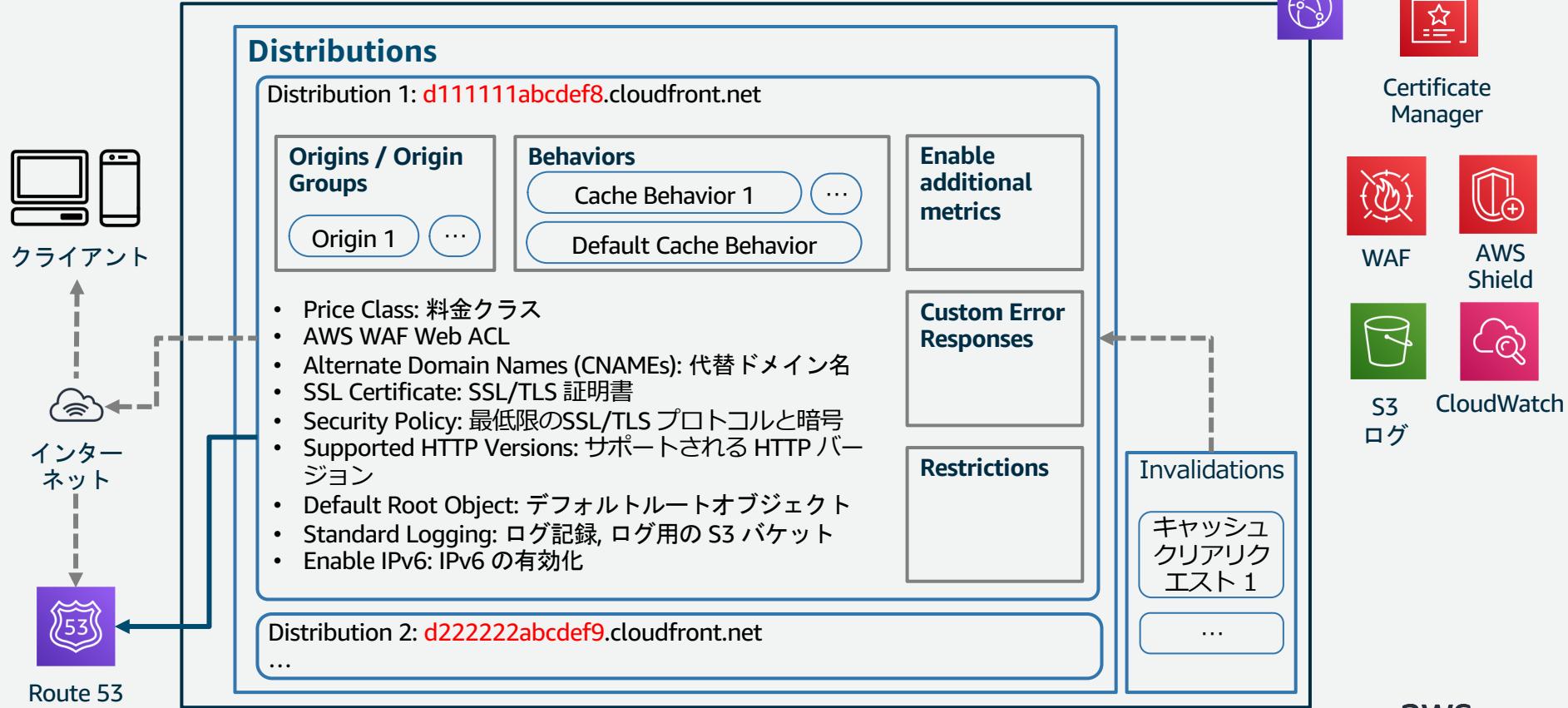
CloudFront 設定

続き

4. Distribution に関する機能

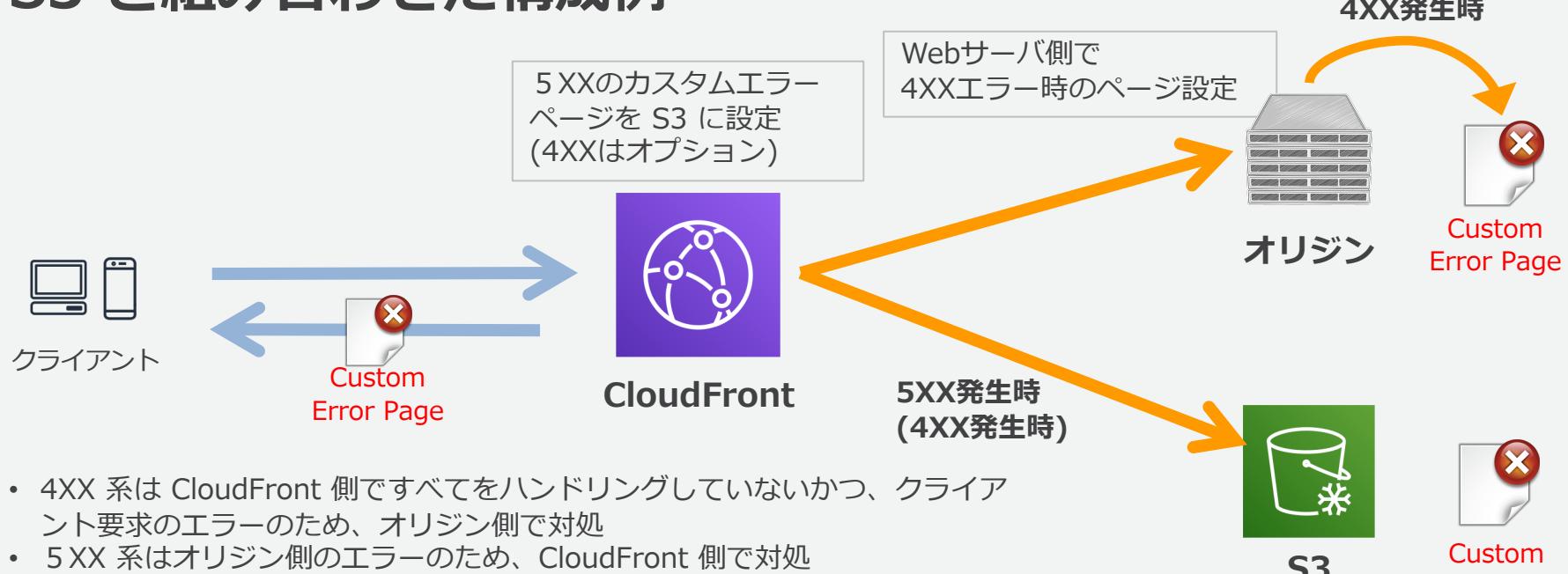
- Custom Error Responses: エラーレスポンス動作のカスタマイズ
- Restrictions: 特定の国のユーザー
- Invalidation: キャッシュファイルの無効化

Distribution 概要図



エラーレスポンス動作のカスタマイズ

S3 と組み合わせた構成例



地域 (GEO) 制限



特定の国のユーザーに対するアクセス制御

- 接続されるクライアントの地域情報を元に、エッジでアクセス判定
- 無効リストもしくは有効リストで指定可能
- Distribution 全体に対して適用される
- 制限されたアクセスには **403** を応答

GEO Restriction有効



Edit Geo-Restrictions

Geo-Restriction Settings

Enable Geo-Restriction Yes No

Restriction Type Whitelist Blacklist

Countries

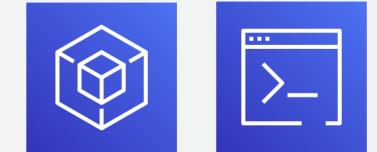
IT -- ITALY
JM -- JAMAICA
JP -- JAPAN
JE -- JERSEY
JO -- JORDAN
KZ -- KAZAKHSTAN

Add >> JP -- JAPAN << Remove

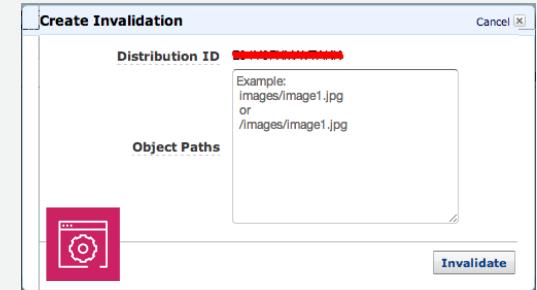
Cancel Yes, Edit

キャッシュファイルの無効化 (Invalidation)

- ・ コンテンツ毎の無効化パス指定
 - ・ 同時に最大 3,000 個までのパス指定が可能
- ・ ワイルドカードを利用した無効化パス指定
 - ・ 同時に最大 15 個まで無効化パスリクエストが指定可能
 - ・ オブジェクト数の制限無し
- ・ AWS Management Console もしくは API で実行可能
- ・ **キャッシュファイルの無効リクエストは有償**のため、Cache Policy の各 TTL や、オリジンで指定する Cache-Control レスポンスヘッダで適切なキャッシュ期間を設定することを推奨
 - ・ 有償: 最初の 1,000 パスまでは追加料金無し、それ以降は、無効をリクエストしたパスごとに \$0.005



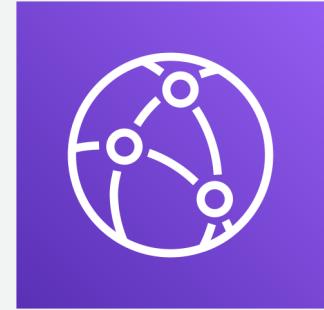
AWS SDK / CLI / API



まとめ

CloudFront の特徴

- ・ 高性能な分散配信 (220+ の POP) ※ 2020 年10月時点
- ・ 高いキャッシュヒット率
- ・ 予測不可能なフラッシュクラウドへの対応
- ・ キャッシュしないコンテンツについても高速化を実現
- ・ ビルトインのセキュリティ機能 (WAF 連携、DDoS 対策)
- ・ AWS Certificate Manager (ACM) との統合による SSL/TLS 証明書の迅速なデプロイとローテーション
- ・ 充実したレポーティング (ログ、ダッシュボード、通知機能)
- ・ Lambda@Edge により柔軟な処理を実行可能
- ・ 完全従量課金 (初期費用がなく安価、一時的な利用も可能)



Appendix

CloudFront 料金モデル

①データ転送アウト(GBあたり)

	米国,メキシコ,カナダ	欧州,イスラエル	南アフリカ,ケニア,中東	南米	日本	オーストラリア,ニュージーランド	シンガポール,韓国,台湾,香港,フィリピン	インド	予約容量の価格
最初の 10 TB/月	\$0.085	\$0.085	\$0.110	\$0.110	\$0.114	\$0.114	\$0.140	\$0.170	問い合わせ
次の40TB/月	\$0.080	\$0.080	\$0.105	\$0.105	\$0.089	\$0.098	\$0.135	\$0.130	問い合わせ
次の100TB/月	\$0.060	\$0.060	\$0.090	\$0.090	\$0.086	\$0.094	\$0.120	\$0.110	問い合わせ
次の350TB/月	\$0.040	\$0.040	\$0.080	\$0.080	\$0.084	\$0.092	\$0.100	\$0.100	問い合わせ
次の524TB/月	\$0.030	\$0.030	\$0.060	\$0.060	\$0.080	\$0.090	\$0.080	\$0.100	問い合わせ
次の4PB/月	\$0.025	\$0.025	\$0.050	\$0.050	\$0.070	\$0.085	\$0.070	\$0.100	問い合わせ
次の5PB/月以上	\$0.020	\$0.020	\$0.040	\$0.040	\$0.060	\$0.080	\$0.060	\$0.100	問い合わせ

②リクエスト(10,000件あたり)

	米国,メキシコ,カナダ	欧州,イスラエル	南アフリカ,ケニア,中東	南米	日本	オーストラリア,ニュージーランド	シンガポール,韓国,台湾,香港,フィリピン	インド	予約容量の価格
HTTP リクエスト	\$0.0075	\$0.0090	\$0.0090	\$0.0160	\$0.0090	\$0.0090	\$0.0090	\$0.0090	問い合わせ
HTTPS リクエスト	\$0.0100	\$0.0120	\$0.0120	\$0.0220	\$0.0120	\$0.0125	\$0.0120	\$0.0120	問い合わせ

③専用IP 独自 SSL 証明書

ディストリビューションに関連付けられた証明書1通につき、月\$600 ※SNIの場合は不要

④オリジンへのデータ転送アウト(GBあたり)

	米国,カナダ	欧州,イスラエル	南アフリカ,ケニア,中東	南米	日本	オーストラリア,ニュージーランド	シンガポール,韓国,台湾,香港,フィリピン	インド	予約容量の価格
すべてのデータ転送	\$0.020	\$0.020	\$0.060	\$0.125	\$0.060	\$0.080	\$0.060	\$0.160	問い合わせ

⑤CloudFrontへのデータ転送アウト(GBあたり)

別の AWS リージョンまたは Amazon CloudFront、\$0.000

⑥無効リクエスト

最初の 1,000 ファイルまで追加料金なし。それ以上はリクエスト毎に \$0.005

<https://aws.amazon.com/jp/cloudfront/pricing/>

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

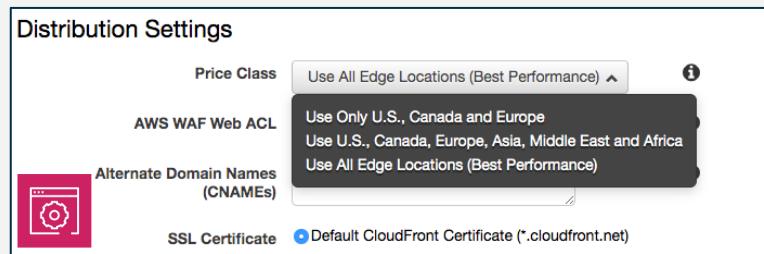


CloudFront 料金クラス

料金クラスを指定することで、安価なエッジロケーションのみを利用した配信が可能

- 料金クラスの変更により、ユーザへの配信速度に影響が出る可能性があるため利用の際は注意が必要

以下に含まれるエッジロケーション	米国, メキシコ, 欧州, イスラエル	カナダ	南アフリカ, ケニア, 中東	南米	日本	オーストラリア, ニュージーランド	シンガポール, 韓国, 台湾, 香港, フィリピン	インド
料金クラス すべて	有	有	有	有	有	有	有	有
料金クラス 200	有	有	有	x	有	x	有	有
料金クラス 100	有	有	x	x	x	x	x	x



Q&A

お答えできなかったご質問については

AWS Japan Blog 「<https://aws.amazon.com/jp/blogs/news/>」にて
後日掲載します。

AWS の日本語資料の場所 「AWS 資料」で検索

The screenshot shows the AWS Japan Language Resources homepage. At the top, there's a navigation bar with the AWS logo, search bar, and links for "日本担当チームへお問い合わせ", "サポート", "日本語", "アカウント", and "コンソールにサインイン". Below the navigation is a main menu with links for "製品", "ソリューション", "料金", "ドキュメント", "学習", "パートナー", "AWS Marketplace", "その他", and a search icon. The main content area features a large title "AWS クラウドサービス活用資料集トップ" and a descriptive paragraph about the resources available. At the bottom, there are four buttons: "AWS Webinar お申込", "AWS 初心者向け", "業種・ソリューション別資料", and "サービス別資料".

日本担当チームへお問い合わせ サポート 日本語 アカウント コンソールにサインイン

製品 ソリューション 料金 ドキュメント 学習 パートナー AWS Marketplace その他 Q

AWS クラウドサービス活用資料集トップ

Amazon Web Services (AWS) は安全なクラウドサービスプラットフォームで、ビジネスのスケールと成長をサポートする処理能力、データベースストレージ、およびその他多種多様な機能を提供します。お客様は必要なサービスを選択し、必要な分だけご利用いただけます。それらを活用するために役立つ日本語資料、動画コンテンツを多数ご提供しております。(本サイトは主に、AWS Webinar で使用した資料およびオンデマンドセミナー情報を掲載しています。)

AWS Webinar お申込 » AWS 初心者向け » 業種・ソリューション別資料 » サービス別資料 »

<https://amzn.to/JPArchive>

AWS Well-Architected 個別技術相談会

毎週”W-A個別技術相談会”を実施中

- AWSのソリューションアーキテクト(SA)に
対策などを相談することも可能

- 申込みはイベント告知サイトから

(<https://aws.amazon.com/jp/about-aws/events/>)

AWS イベント で[検索]

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>





Amazon CloudFront

(Cache Control 編)

AWS Black Belt Online Seminar

文珠 啓介

Solutions Architect
2023/04

AWS Black Belt Online Seminar とは

- ・ 「サービス別」「ソリューション別」「業種別」などのテーマに分け、アマゾン ウェブ サービス ジャパン合同会社が提供するオンラインセミナーシリーズです
- ・ AWSの技術担当者が、AWS の各サービスやソリューションについてテーマごとに動画を公開します
- ・ 動画を一時停止・スキップすることで、興味がある分野・項目だけの聴講も可能、スキマ時間の学習にもお役立ていただけます
- ・ 以下の URL より、過去のセミナー含めた資料などをダウンロードすることができます
 - ・ <https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-service-cut/>
 - ・ <https://www.youtube.com/playlist?list=PLzWGOASvSx6FIwIC2X1nObr1KcMCBBlqY>

内容についての注意点

- ・ 本資料では 2023 年 3 月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 (<https://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には別途消費税をご請求させていただきます

自己紹介

名前：文珠 啓介 (Keisuke Monju)

所属：ソリューションアーキテクト

経歴：

- ・オンプレミスのインフラ構築・運用
- ・AWSへのマイグレーション対応・運用

好きなAWSサービス：

Amazon CloudFront、AWS WAF、AWS Firewall Manager



本セミナーの対象者

例：

Amazon CloudFront の概要は知っているが、自社のサービス向けに導入する
メリットとして何があるのかが不明な方

Amazon CloudFront は既に活用しているが、コンテンツキャッシングの用途で
は利用していない方

Amazon CloudFront のコンテンツキャッシングを利用しているが、もっとコン
テンツキャッシングについて知って、より良い運用に繋げたい方

アジェンダ

1. Amazon CloudFrontについて
2. Amazon CloudFront 活用のメリットと今回のメイントピックについて
3. Web サイトを構成する要素
(静的コンテンツと動的コンテンツについて)
4. コンテンツキャッシュの運用方法について
5. Amazon CloudFront とオリジンで行うキャッシングの設定について
6. Amazon CloudFront を利用したキャッシングの運用における Tips
7. クライアント側のキャッシングについて
8. その他

Amazon CloudFront について

450+

Amazon
CloudFront
Points of
Presence

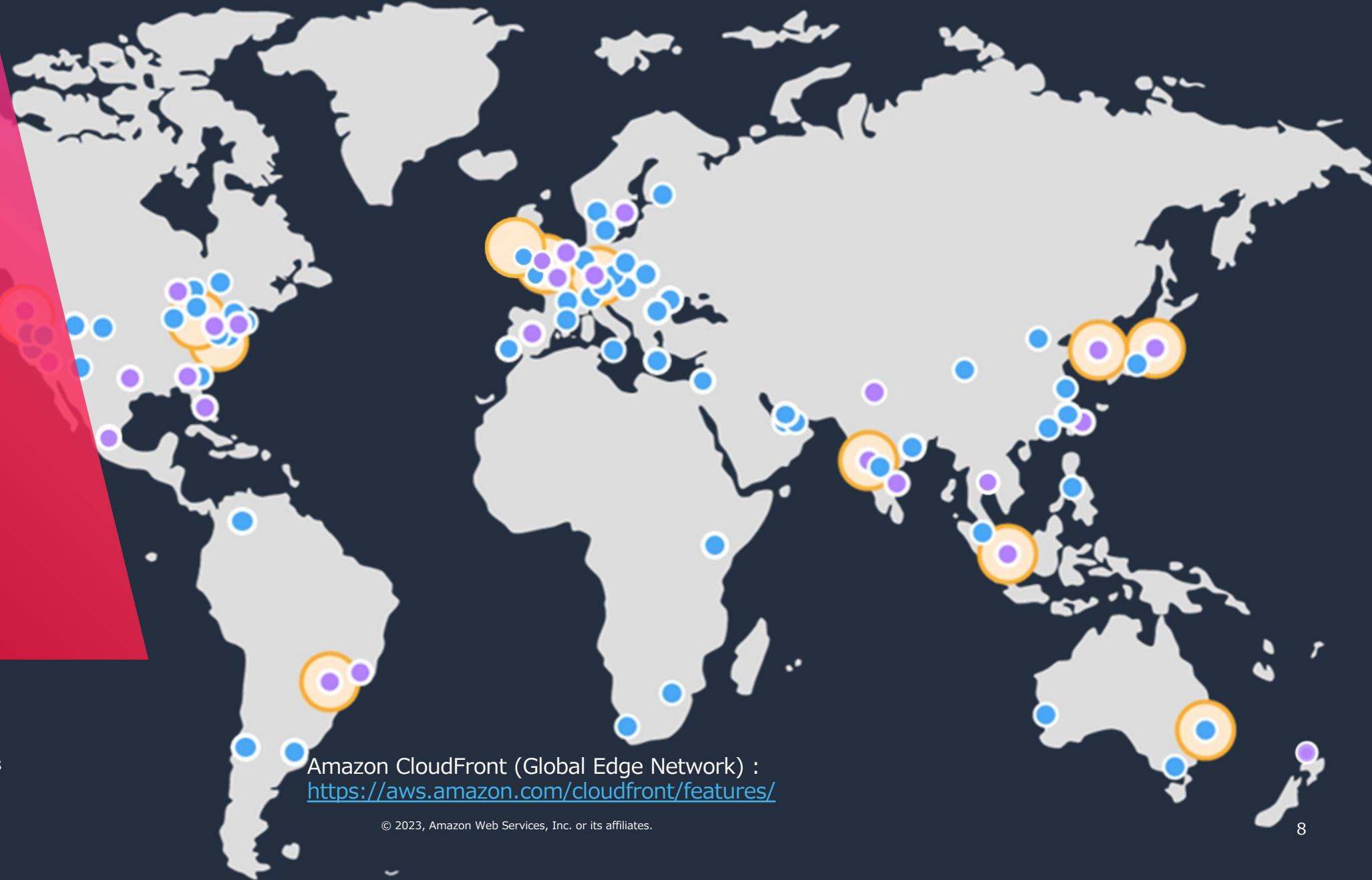
13 Regional
Edge Caches

90+ cities,
48 countries

China CDN

As of 03/09/2023

- Edge Locations
- Multiple Edge Locations
- aws  ● Regional Edge Caches



Amazon CloudFront (Global Edge Network) :
<https://aws.amazon.com/cloudfront/features/>

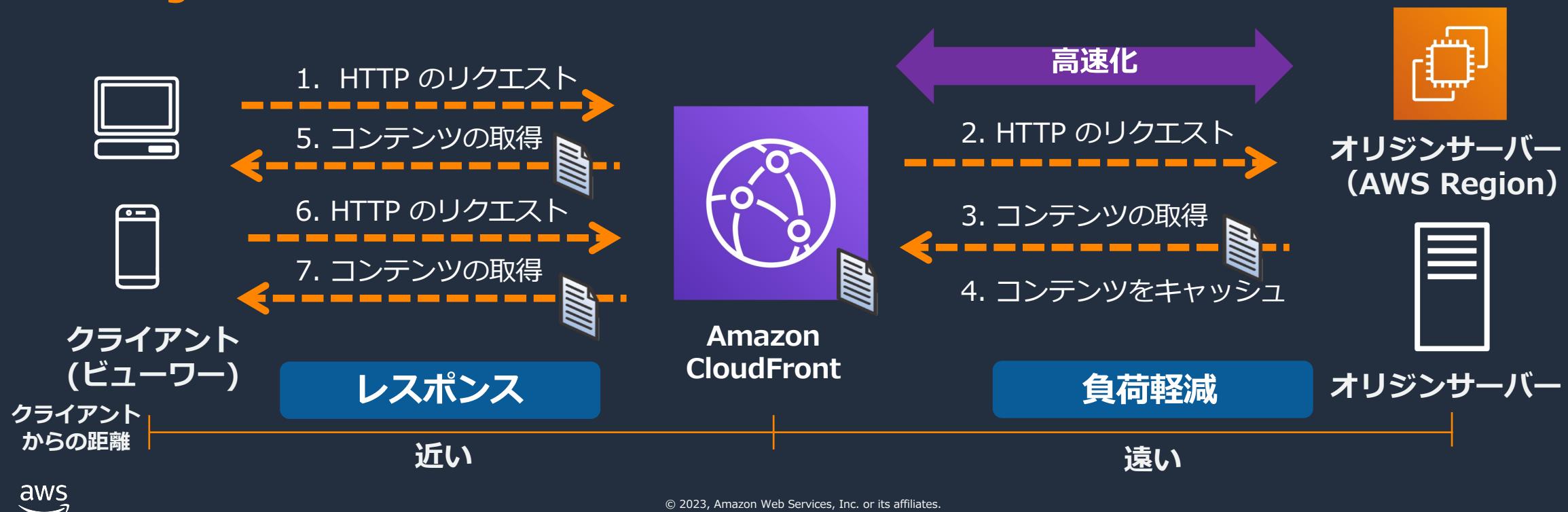
© 2023, Amazon Web Services, Inc. or its affiliates.

Amazon CloudFront の概要

Fast, highly secure and programmable content delivery network (CDN)

高い安全性と高性能を実現するプログラム可能なコンテンツデリバリー・ネットワーク

- ユーザーを一番近いエッジ・ロケーションに誘導することで **配信を高速化**
- エッジ・サーバーでコンテンツのキャッシングを行い **オリジンの負荷をオフロード**
- AWS global network を利用することによる非キャッシングコンテンツの高速化**



Amazon CloudFront 活用のメリットと 今回のメイントピックについて

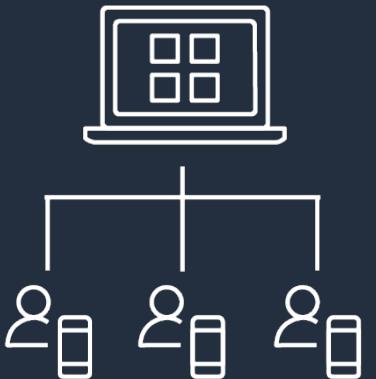
Amazon CloudFront を利用するメリット

AWS のグローバル インフラストラクチャ



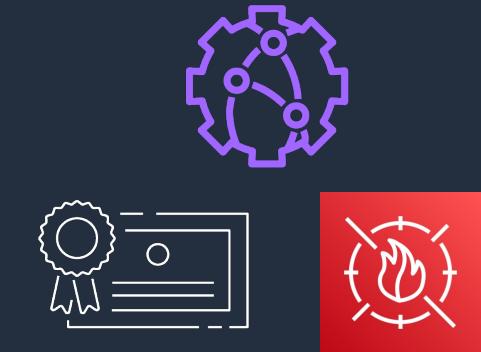
世界のどこからアクセスしても
最適な経路でオリジンまで
アクセスが可能なため
ネットワークのレイテンシーを
最適化できる

エッジロケーションを キャッシュサーバーとして活用



キャッシュされたコンテンツを
エッジロケーションから返すことで
エンドユーザーはレスポンスを早く
受けることができ、サーバーサイド
はコンピュートリソース節約できる

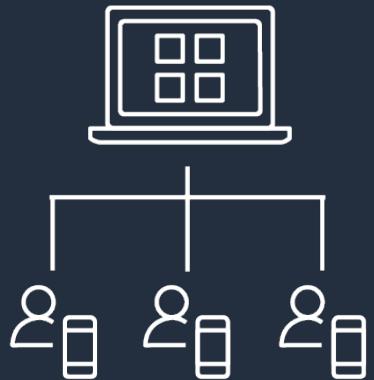
CloudFront の様々な機能や 他のAWSサービスとの連携



エッジ関数の実行、SSL/TLS
通信の終端、AWS WAFによる
セキュリティ対策など、アプリ
ケーション側の変更を必要とせず
動作のカスタマイズができる

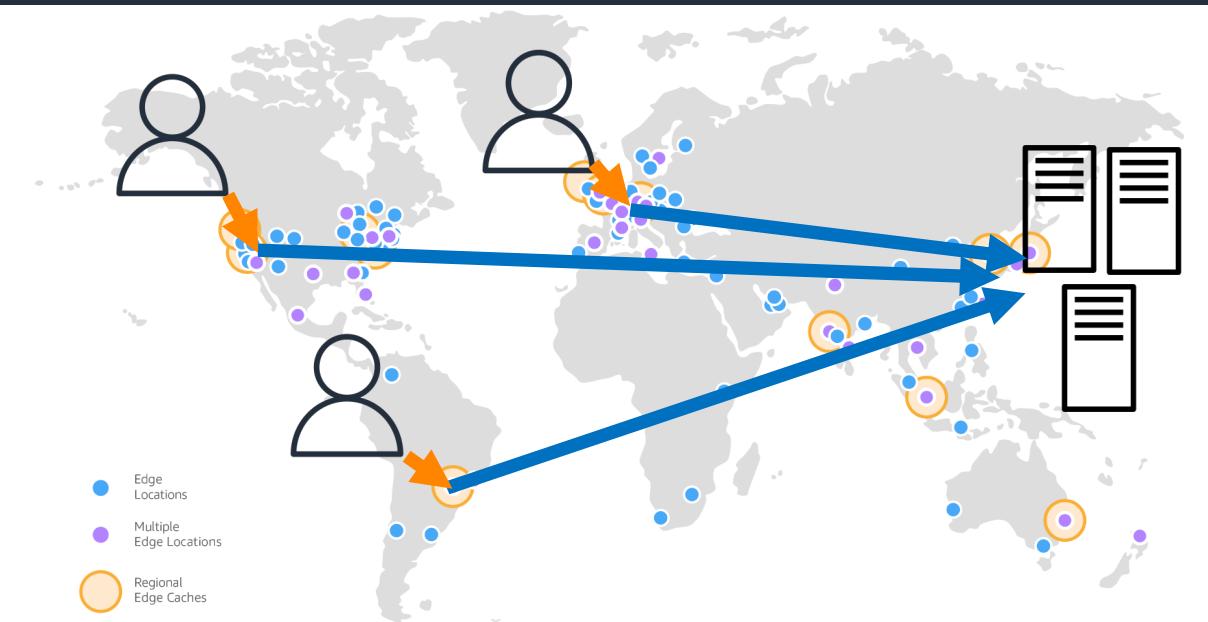
Amazon CloudFront を利用するメリット

エッジロケーションを
キャッシュサーバーとして活用

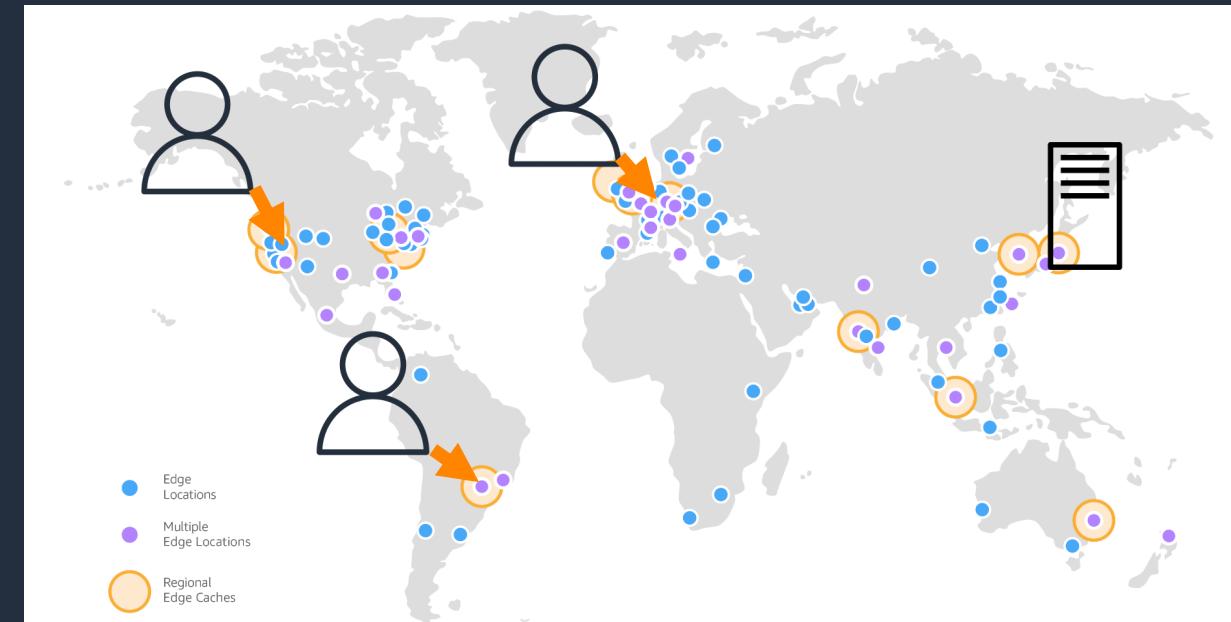


キャッシュされたコンテンツを
エッジロケーションから返すことで
エンドユーザーはレスポンスを早く
受けることができ、サーバーサイド
はコンピュートリソース節約できる

キャッシュがある場合と無い場合の違い



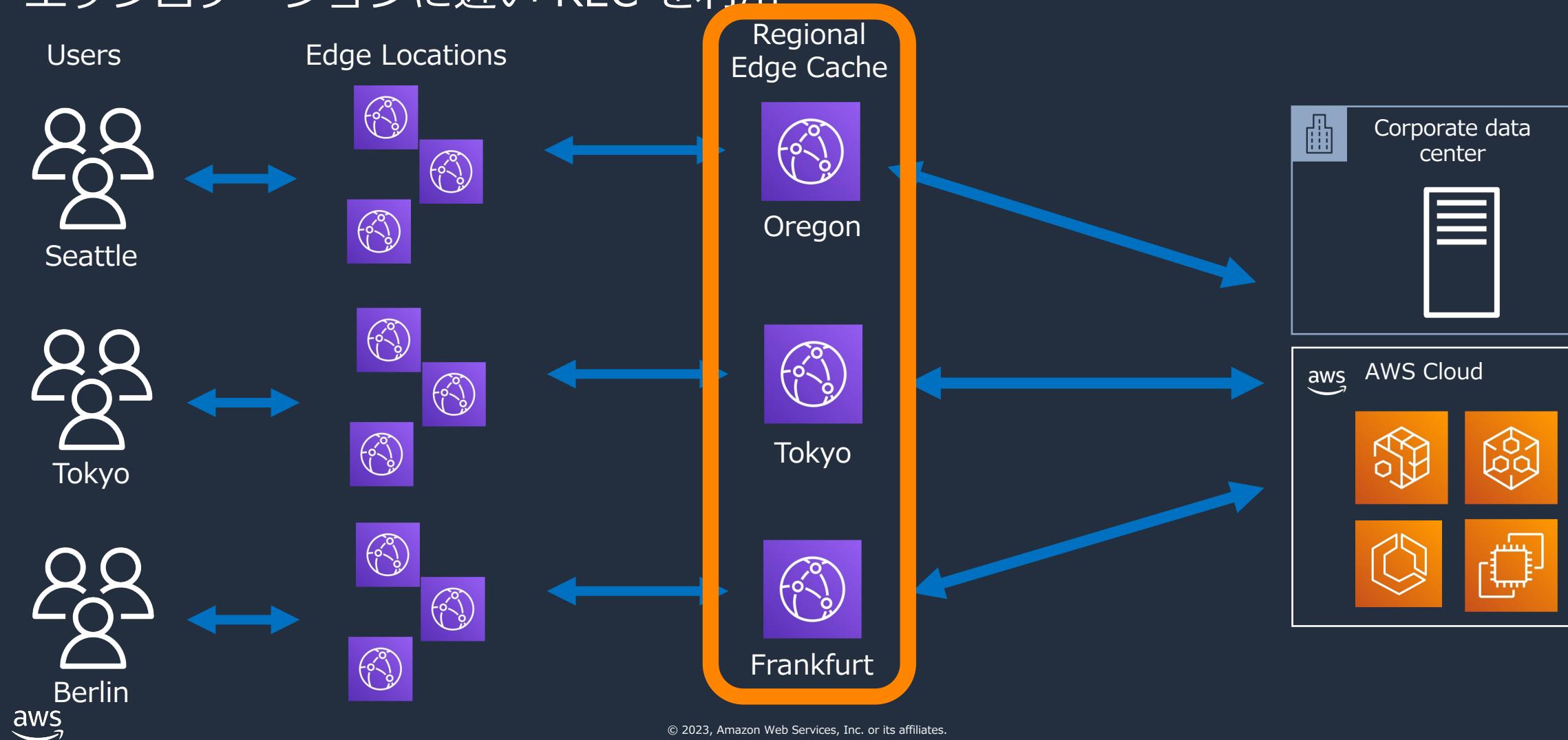
キャッシュが無いアクセス



キャッシュがあるアクセス

リージョナルエッジキャッシュ (REC)

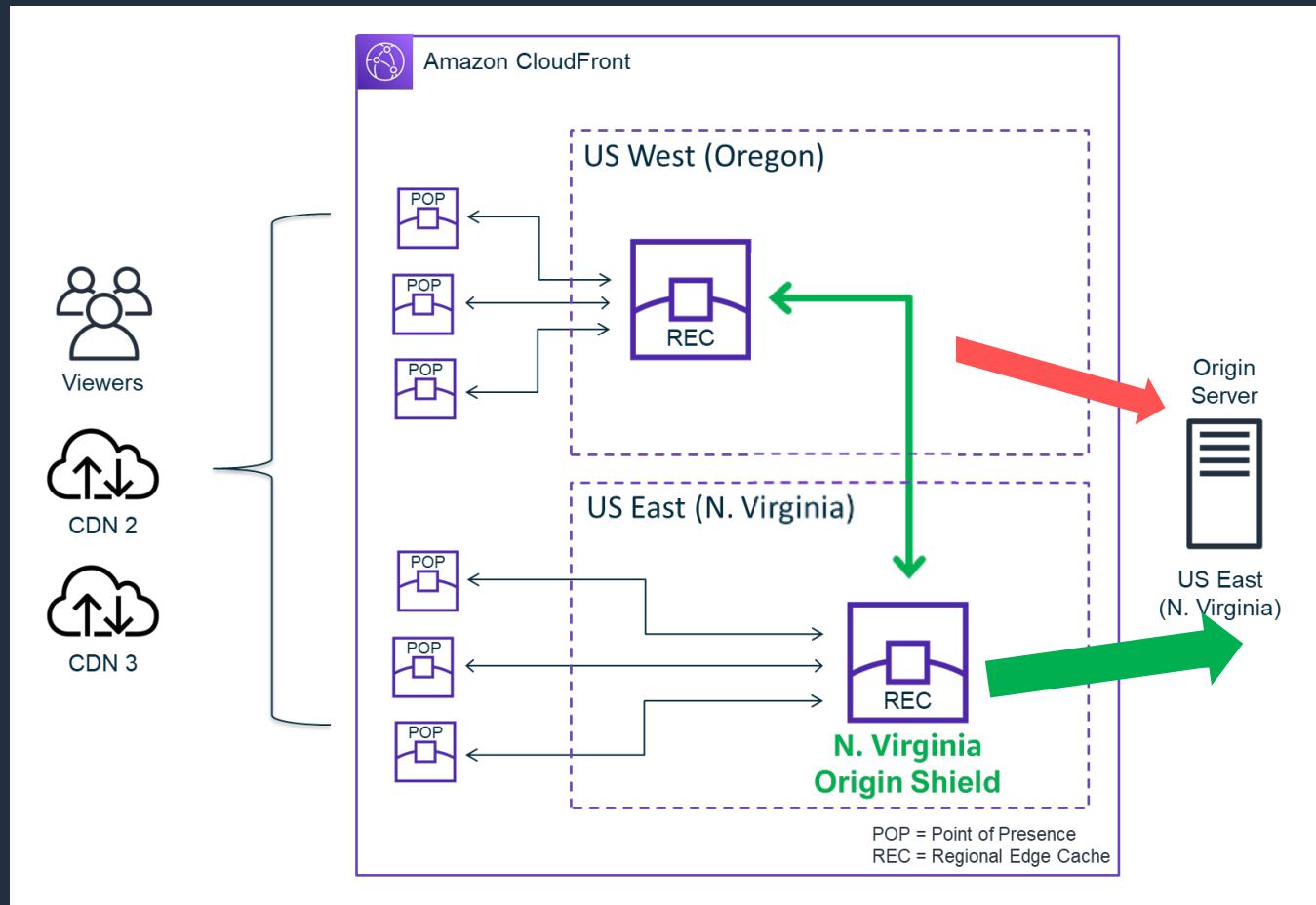
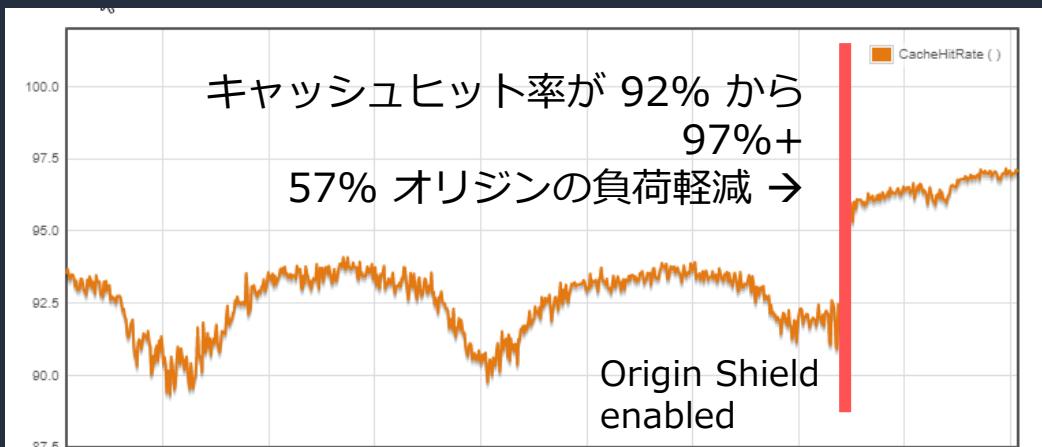
- ユーザーからのリクエストを REC で集約しオリジンにリクエスト
- エッジロケーションに近い REC を利用



Origin Shield

Origin Shield は、**オリジンの負荷と運用コストを削減する**ために、
オリジンの前に配置される**キャッシュレイヤー**

- キャッシュヒット率の向上
- リージョン間の重複したリクエストを集約
- オリジンの可用性向上
- 運用コストの削減 データ転送アウト、
ライブストリーミングのパッケージング処理
やオンザフライのイメージ変換



実際どれくらい早くなるのか？

The diagram illustrates a web application interface on the left and its corresponding Network tab in the developer tools on the right. The interface includes an input message box ('Input Message') containing 'こんにちは世界!!', a send button ('Send'), and an output message box ('Output Message') displaying 'hello world!!'. A red circle highlights the CloudFront logo in the header. Arrows point from the interface components to the developer tools: a blue arrow from the input message to the Network tab, a red arrow from the send button to the Network tab, and a blue arrow from the output message to the Network tab. The developer tools show a list of resources and a waterfall chart. A legend at the top right indicates that red bars represent 'キャッシュコンテンツ' (Cached Content) and blue bars represent '非キャッシュコンテンツ' (Non-Cached Content). In the waterfall chart, the first four items ('document', 'stylesheet', 'script', 'png') are highlighted in green, while the API request ('xhr') is highlighted in red.

HTML
画像
CSS
JavaScript

—— キャッシュコンテンツ
—— 非キャッシュコンテンツ

Input Message
こんにちは世界!!
Send
All Clear
Output Message
hello world!!

キャッシュコンテンツ
非キャッシュコンテンツ

API の結果(動的コンテンツ)

Name	Status	Type	Initiator	Size	Time
d3lnc4nmpnao2n.cloudfront.net	200	document	Other	1.3 kB	611 ms
styles.css	200	stylesheet	(index)	394 B	16 ms
script.js	200	script	(index)	865 B	16 ms
cloudfront.png	200	png	(index)	20.6 kB	9 ms
cloudfront.png	200	png	Other	20.6 kB	10 ms
api?input_text=%26%2312371%...	200	xhr	script.js:11	355 B	916 ms

AWS Hands-on for Beginners (Amazon CloudFrontおよびAWS WAFを用いて エッジサービスの活用方法を学ぼう)
https://pages.awscloud.com/JAPAN-event-OE-Hands-on-for-Beginners-CF_WAF-2022-reg-event.html



ただし、注意も必用



- ・間違えて古いコンテンツをキャッシュしたまま
新規の画面を公開してしまい、表示は古い画面のままだった
※まだ、取り返しがつく



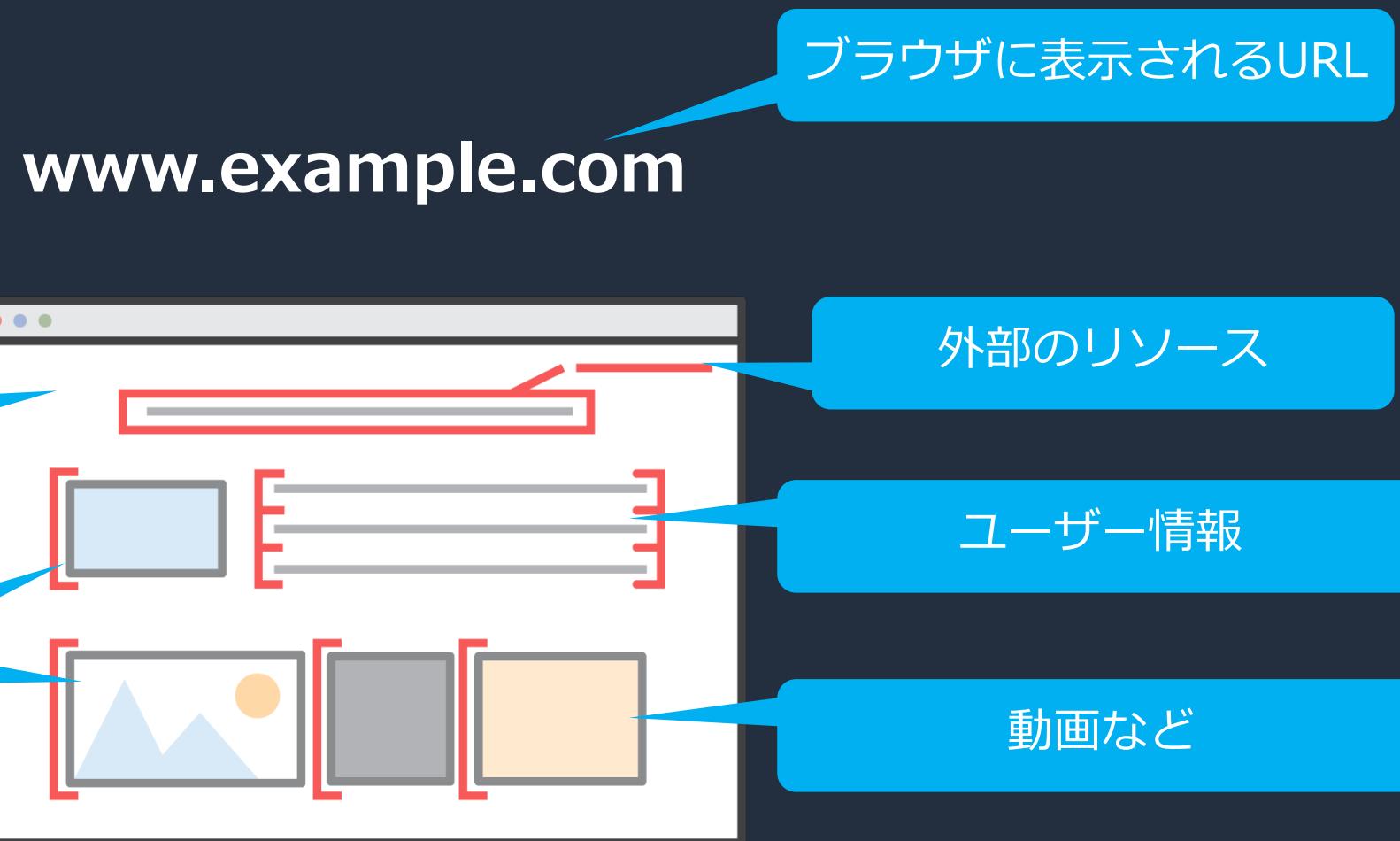
- ・お客様の個人情報をキャッシュしており、設定のミスで
他のお客様に誤って、個人情報が公開されてしまった
※これは、取り返しがつかない



※どんなコンテンツをキャッシュさせるかの取捨選択が大事
(個人情報を含んだコンテンツのキャッシュは避ける！！)

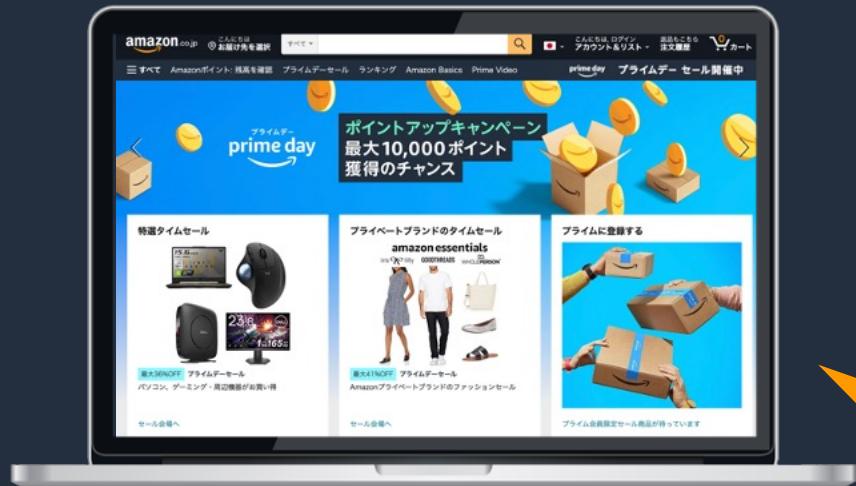
Web サイトを構成する要素 (静的コンテンツと動的コンテンツについて)

Web サイトの構成要素について



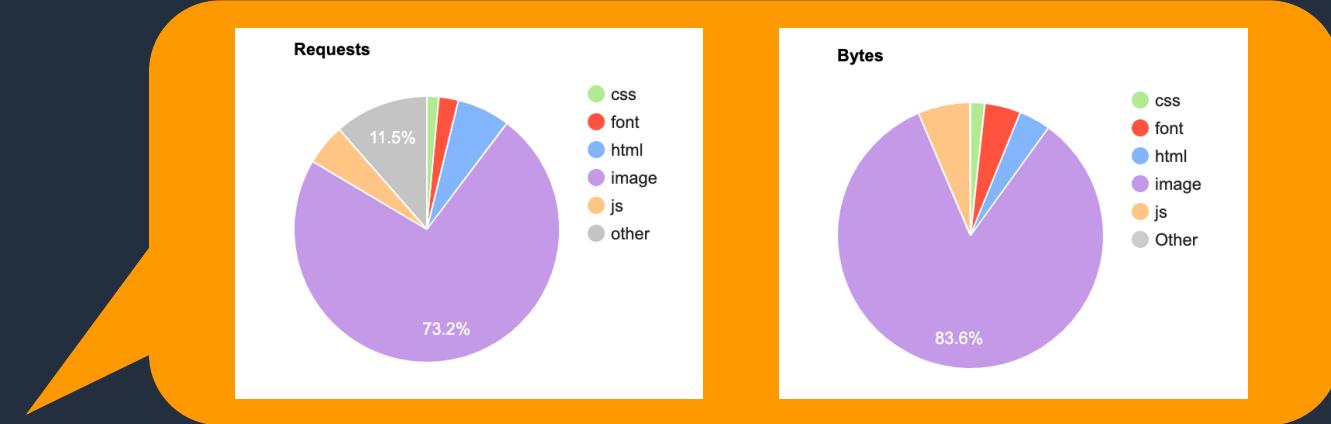
実際の Web サイトの例

画像などの静的コンテンツがリクエスト数、データ量の大半を占める

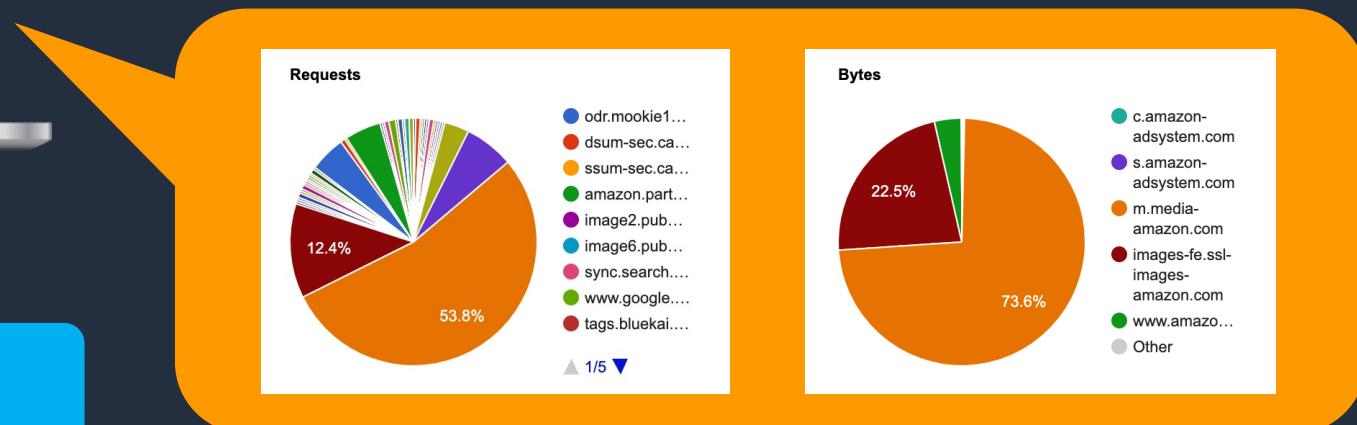


<https://www.amazon.co.jp/>

90-95% はキャッシュ可能



複数のドメインから様々なコンテンツ、アセットがダウンロードされる



Web サイトの様々な要素

- **動的コンテンツ**

ユーザーのリクエストによって変わるコンテンツのこと
(例) ユーザー名や、カートの情報など



- **静的コンテンツ**

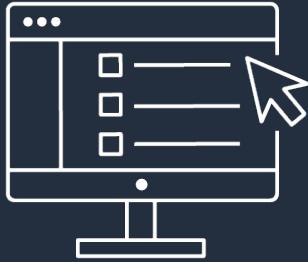
ユーザーのリクエストによって変わらないコンテンツのこと
(例) HTML、CSS、画像、動画など



※ Web サイトの大半は**静的なコンテンツ**で構成される

コンテンツキャッシュの 運用方法について

コンテンツキャッシュの運用において最初に考えること



自分のサイトにどういう コンテンツがあるか整理

どのような静的コンテンツや
動的コンテンツによって
自分のサイトが構成されているのか。
また、それらのコンテンツを
運用している部署はどこか。



コンテンツ更新の頻度 (スケジュール)

各コンテンツの更新は
どのタイミングによって発生するか。
その頻度はどれくらいか。
キャッシングするコンテンツと
しないコンテンツの割合はどうか。



誰がどういう方法でコンテンツ のキャッシング更新を行うか

実際に誰がコンテンツの
更新を行うのか。
コンテンツを更新した時に
キャッシングをどのような方法
で更新するのか。

コンテンツキャッシュの運用において最初に考えること

※ Web サイトのコンテンツに、様々な部署が関わるのは一般的
部署ごとにコンテンツを保持しているサーバーが異なることも
(コンテンツを更新するスケジュールも方法もバラバラ)



www.example.com



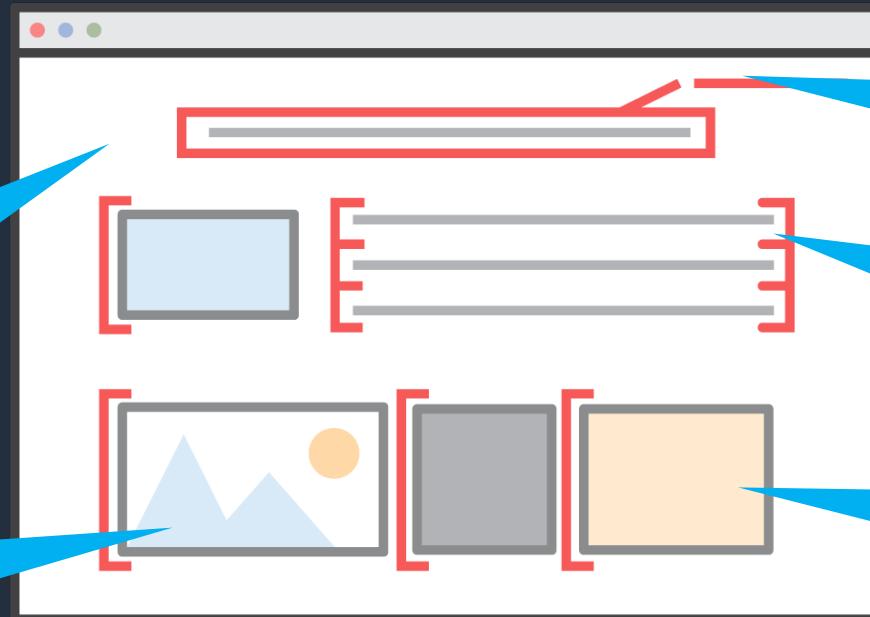
サイト全般の運営
(企画統括部が管理)



HTML/CSS
(デザイン部が管理)



商品画像
(マーケティング部が管理)



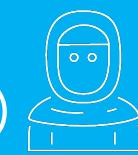
外部のリソース
(協力会社が管理)



動的コンテンツ
(IT開発部が管理)



動画
(メディア開発部が管理)

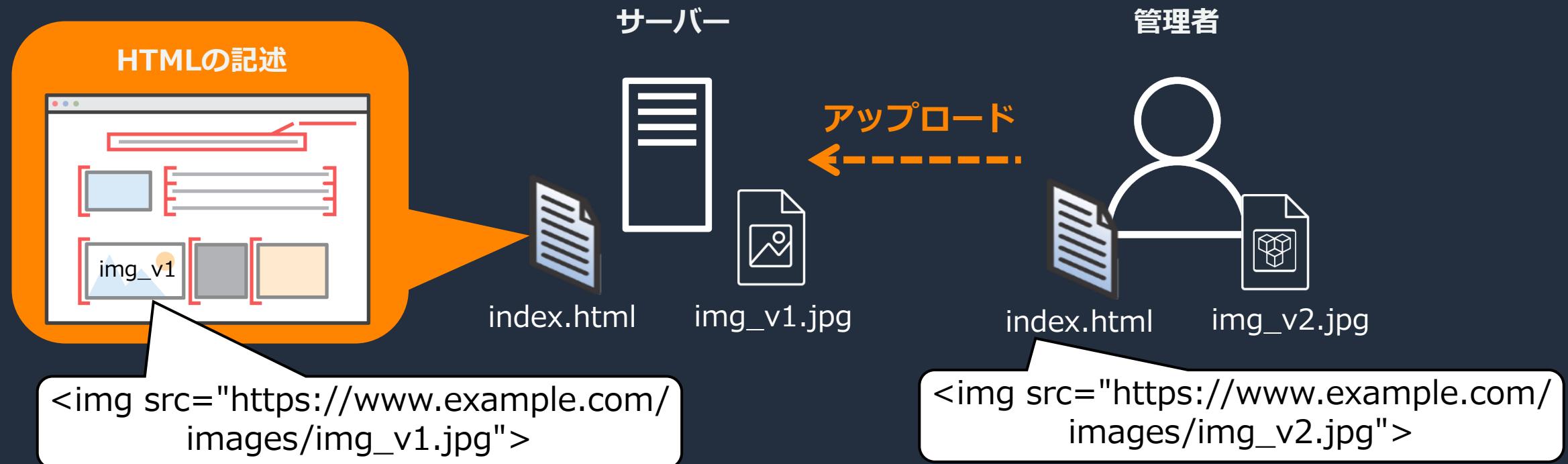


既存コンテンツのキャッシュを更新するための 2 つの方法

- ファイル名にバージョン識別子を使用して更新する
- 同じ名前を使用してファイルを更新する

既存コンテンツのキャッシュを更新するための 2 つの方法

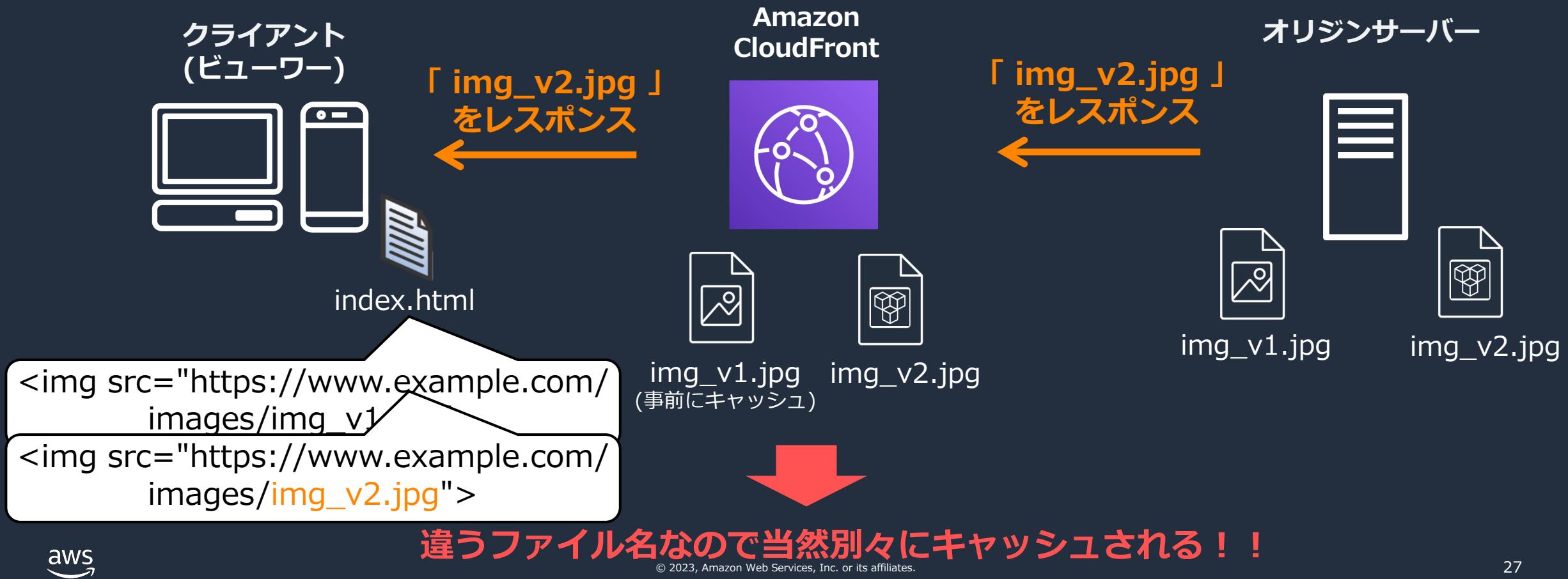
- ファイル名にバージョン識別子を使用して更新する
- 同じ名前を使用してファイルを更新する



既存のコンテンツと CloudFront ディストリビューションを更新する
aws https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/UpdatingExistingObjects.html

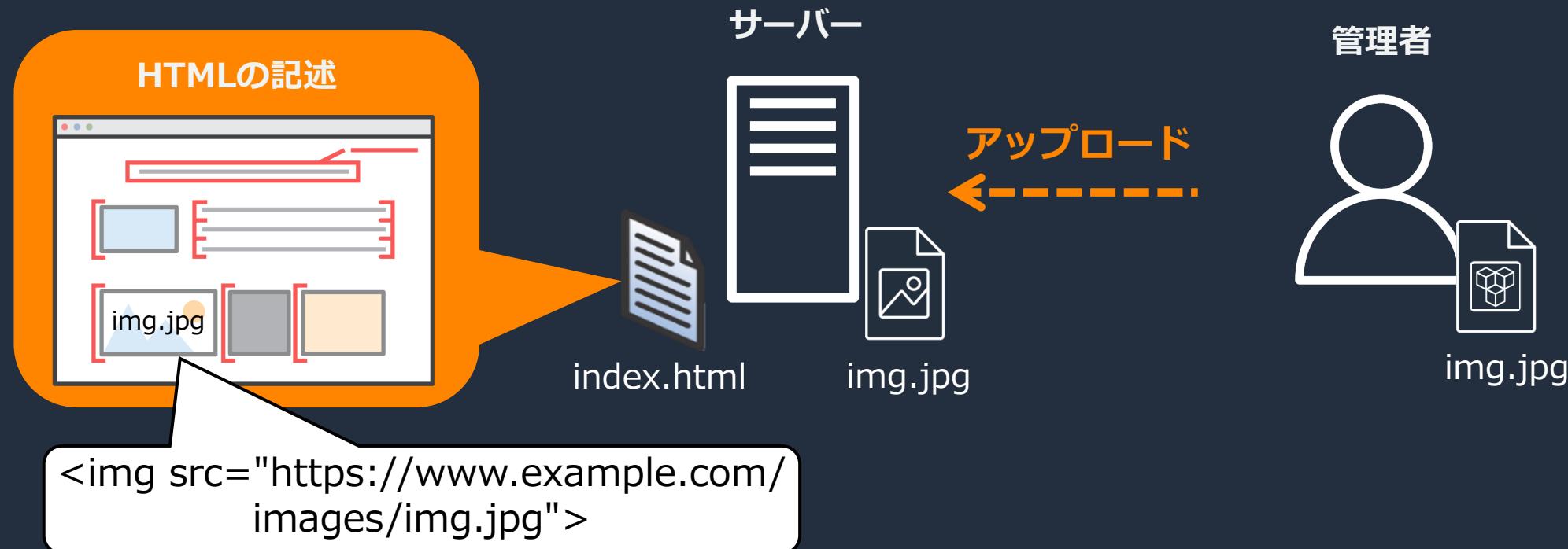
既存コンテンツのキャッシュを更新するための 2 つの方法

- ・ ファイル名にバージョン識別子を使用して更新する
- ・ 同じ名前を使用してファイルを更新する



既存コンテンツのキャッシュを更新するための 2 つの方法

- ファイル名にバージョン識別子を使用して更新する
- 同じ名前を使用してファイルを更新する



既存のコンテンツと CloudFront ディストリビューションを更新する
aws https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/UpdatingExistingObjects.html

既存コンテンツのキャッシュを更新するための 2 つの方法

- ファイル名にバージョン識別子を使用して更新する
- 同じ名前を使用してファイルを更新する



1つのコンテンツに紐付いているメタ情報について

※実は1つのコンテンツに対して大量のメタ情報が付与されている！！
(ファイル名が同じでも、これらの情報を用いて別物だと認識させることが可能)



1コンテンツ

Name: storefront?storeType=browse&node=2275256051
Request URL: https://www.amazon.co.jp/kindle-dbs/storefront?storeType=browse&node=2275256051
Request Method: GET
Status Code: 200
Remote Address: 13.249.153.129:443
Referrer Policy: strict-origin-when-cross-origin

General
Headers
Payload
Preview
Response
Initiator
Timing
Cookies

Request URL: https://www.amazon.co.jp/kindle-dbs/storefront?storeType=browse&node=2275256051
Request Method: GET
Status Code: 200
Remote Address: 13.249.153.129:443
Referrer Policy: strict-origin-when-cross-origin

Response Headers
accept-ch: ect,rtt,downlink,device-memory,sec-ch-device-memory,viewport-width,sec-ch-viewport-width,dpr,sec-ch-dpr,sec-ch-ua-platform,sec-ch-ua-platform-version
accept-ch-lifetime: 86400
cache-control: no-cache
content-encoding: gzip
content-language: ja-JP
content-security-policy: upgrade-insecure-requests;report-uri https://metrics.media-amazon.com/
content-security-policy-report-only: default-src 'self' blob: https: data: mediastream: 'unsafe-eval' 'unsafe-inline';report-uri https://metrics.media-amazon.com/
content-type: text/html; charset=UTF-8

1つのコンテンツに紐づく情報
(ブラウザの開発者ツールより。)

Cookies

クエリ文字列

ヘッダー

etag: "1be29193e65f4fee5fa62d7a4d7d9305"
last-modified: Tue, 21 Mar 2023 13:28:52 GMT

(補足) キャッシュ期間が切れたあとの挙動について

キャッシュしたコンテンツに変更がない場合

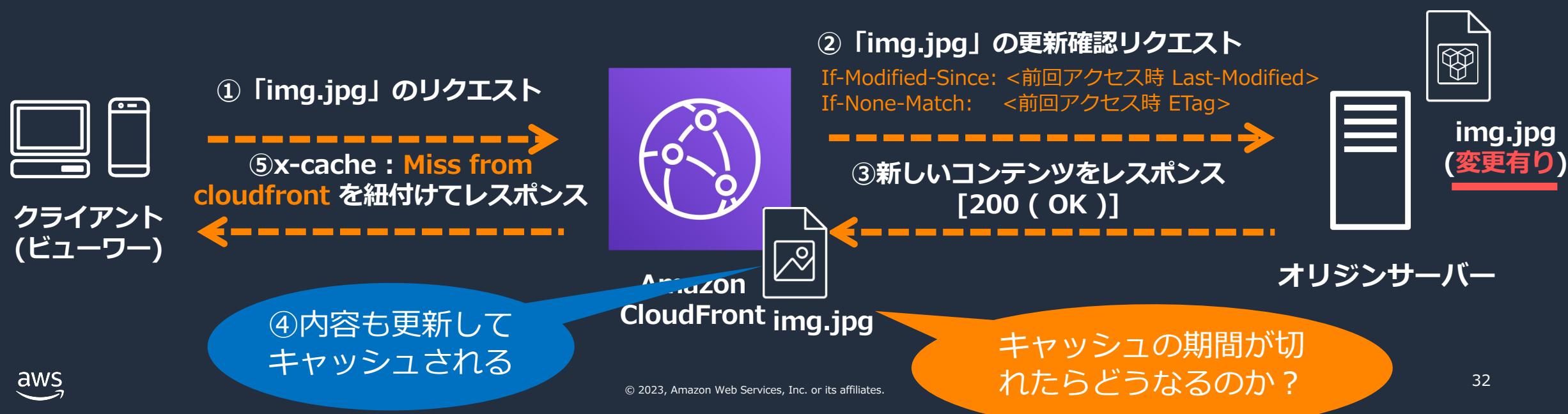
- CloudFront にてキャッシュしているコンテンツの期間がきた際、
Etag / LastModified といったコンテンツに紐付いている HTTP ヘッダー
の値を用いてオリジンサーバーに確認のリクエストを送る (If-Modified-Since / If-None-Match)
- レスポンスとして上記の値に変更がなければステータスコード 304 (Not Modified) が
返ってきてコンテンツのキャッシュ期間は更新される
(x-cache : RefreshHit from cloudfront をレスポンス)



(補足) キャッシュ期間が切れたあとの挙動について

キャッシュしたコンテンツに変更がある場合

- CloudFront にてキャッシュしているコンテンツの期間がきた際、
Etag / LastModified といったコンテンツに紐付いている HTTP ヘッダー
の値を用いてオリジンサーバーに確認のリクエストを送る (If-Modified-Since / If-None-Match)
- レスポンスとして上記の値に変更があれば、コンテンツを新しいものに差し替えて
ステータスコード 200 (OK) とともに新しいコンテンツをレスポンス
(x-cache : Miss from cloudfront を紐付けてレスポンス)



既存コンテンツのキャッシュを更新するための 2 つの方法

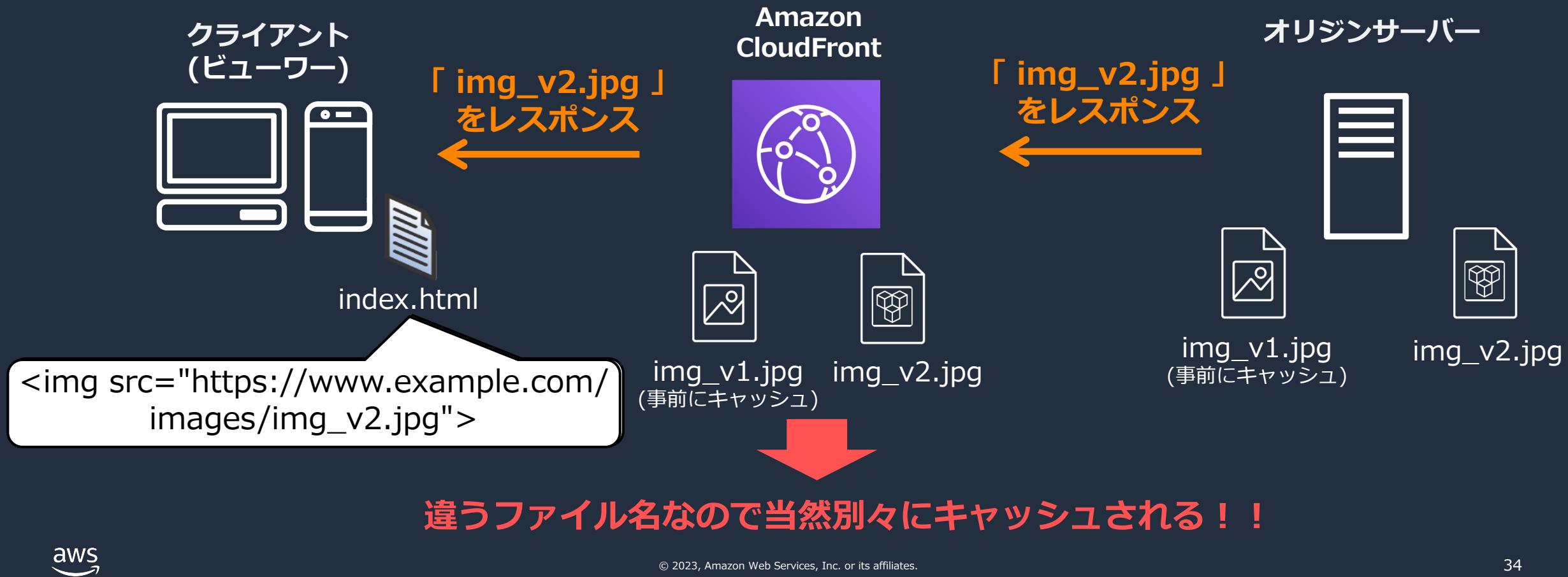
- ファイル名にバージョン識別子を使用して更新する
- 同じ名前を使用してファイルを更新する

注意点

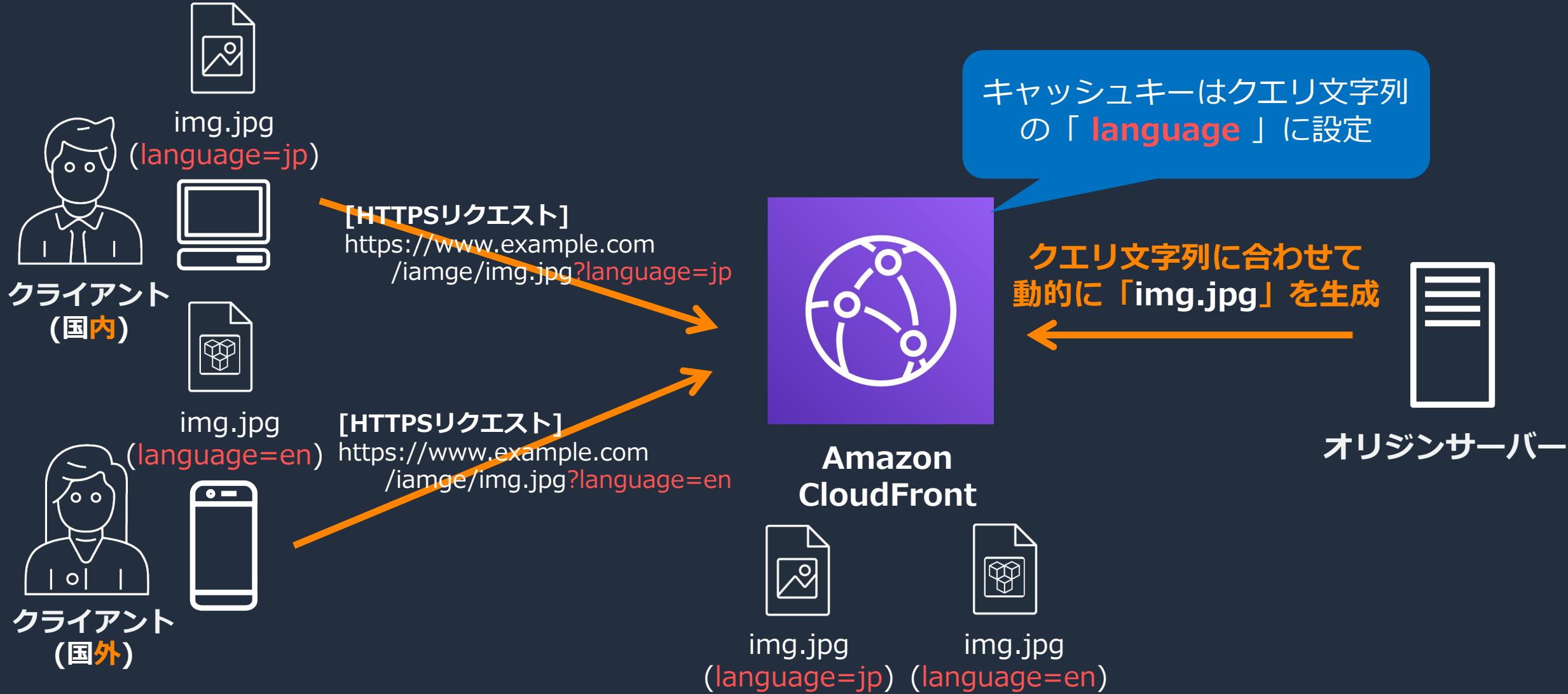
- キャッシュの有効期限を上手く制御して、コンテンツごとにどれくらいの期間をキャッシュさせるのか決めて運用する必要がある
- キャッシュの有効期限を長くすることで、オリジンサーバーへのリクエストを減らせるが、コンテンツの更新が頻繁に発生する場合には、有効期限を短くする必要がある
- 指定したタイミングで一斉にキャッシュの更新をしたい場合などには、キャッシュの無効化対応が必要になる

既存コンテンツのキャッシュを更新するための 2 つの方法

- ファイル名にバージョン識別子を使用して更新する
- 同じ名前を使用してファイルを更新する



キャッシュキーを用いて別のコンテンツとしてキャッシュする



CloudFront におけるキャッシュの設定について

大量に付与されているメタ情報の何をキーにして CloudFront は
“同じ名前のファイルを別のキャッシュとして保管する”のか？

重要！

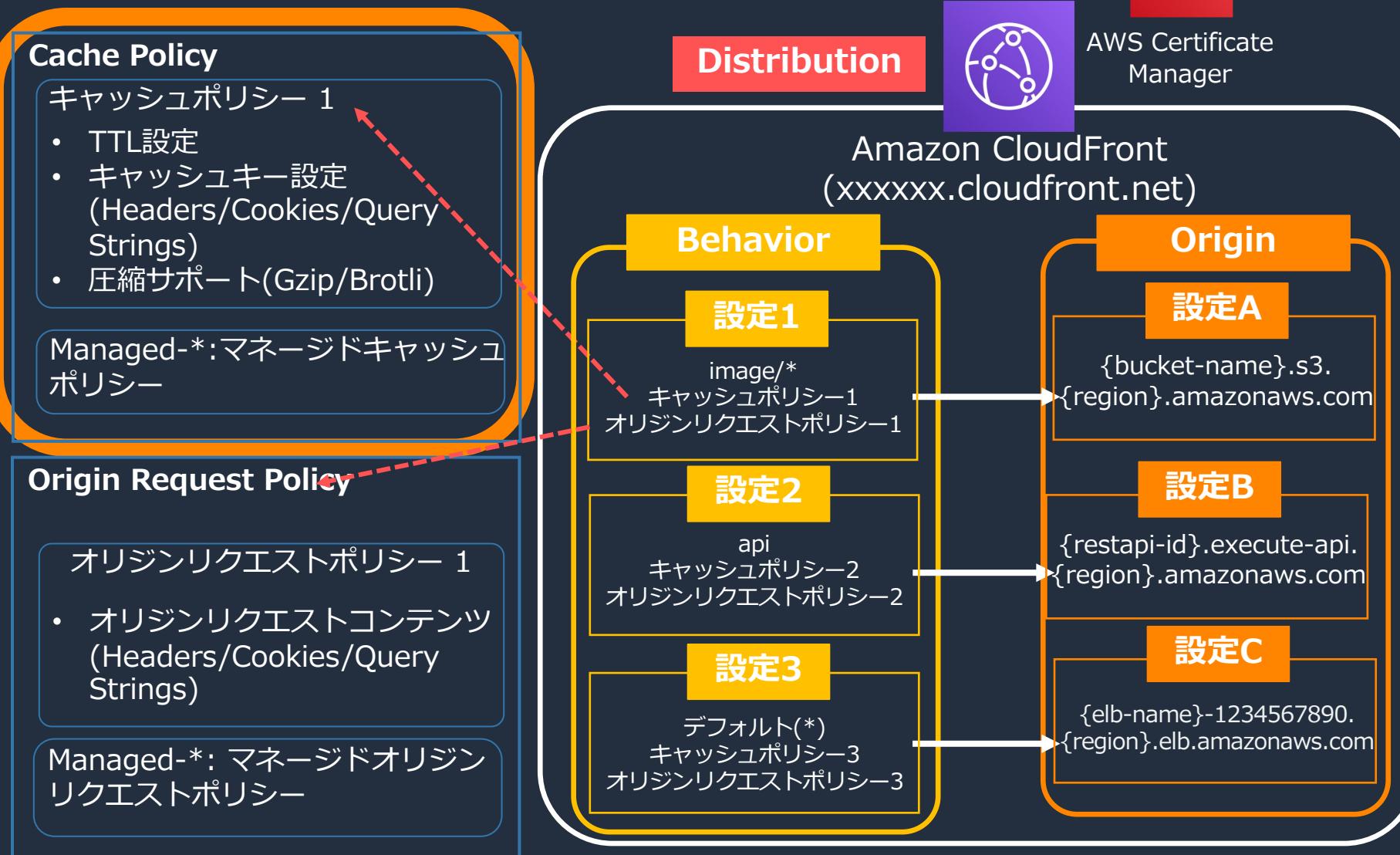


Amazon
CloudFront

Cache Policy
(キャッシュポリシー)

Amazon CloudFront とオリジンで 行うキャッシュの設定について

CloudFront の設定イメージ



AWS Certificate Manager

オリジンサーバー



Amazon S3



Amazon API Gateway



AWS Lambda



Application Load Balancer



Amazon EC2

Cache Policy の設定画面

● 3つのセクション

・ TTL (Time to live) 設定

CloudFront キャッシュ内のオブジェクトの有効期間を決定する（オリジン側の「Cache-Control」および「Expires」HTTP ヘッダーに連動する）

・ キャッシュキー設定

CloudFront がコンテンツをキャッシュする際に、一意のコンテンツであることを判断するための、キーを指定する（ヘッダー、クエリストリング、Cookie の要素を指定できる）

・ 圧縮サポート

ビューウィーからのリクエストに基づいて、特定のタイプのコンテンツを必要に応じて自動的に圧縮し、キャッシュ及びレスポンスするための設定

圧縮ファイルの供給

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/ServingCompressedFiles.html



[補足] Cache Policy における「Amazon マネージドポリシー」

マネージドポリシーについて

- AWS 側が用途に合わせて予め用意しているテンプレート
- そのまま使えそうなら使っても良い(参考として、中身を覗いてみるのをオススメ)
- 自身で作成するポリシーはカスタムポリシーと呼ばれる

The screenshot shows the CloudFront Cache Policy configuration interface. The top navigation bar includes 'CloudFront > ポリシー > キャッシュ' (CloudFront > Policies > Cache). Below the navigation, there are three tabs: 'キャッシュ' (Cache), 'オリジンリクエスト' (Origin Request), and 'レスポンスヘッダー' (Response Header). The 'キャッシュ' tab is selected. A large orange box highlights the 'Amazon マネージドポリシー' (Amazon Managed Policies) section. This section lists five managed policies with their descriptions:

名前	説明
Amplify	Policy for Amplify Origin
CachingDisabled	Policy with caching disabled
CachingOptimized	Policy with caching enabled. Supports Gzip and Brotli
CachingOptimizedForUncompressedObjects	Default policy when compression is disabled
Elemental-MediaPackage	Policy for Elemental MediaPackage Origin

At the bottom of the highlighted section, there is a red box containing the text 'カスタムポリシー (3) 情報' (Custom Policies (3) Information) and three buttons: '編集' (Edit), '削除' (Delete), and 'キャッシュポリシーを作成' (Create Cache Policy). The entire 'Amazon マネージドポリシー' section is also enclosed in a red box.

名前	説明
Amplify	Policy for Amplify Origin
CachingDisabled	Policy with caching disabled
CachingOptimized	Policy with caching enabled. Supports Gzip and Brotli
CachingOptimizedForUncompressedObjects	Default policy when compression is disabled
Elemental-MediaPackage	Policy for Elemental MediaPackage Origin

カスタムポリシー (3) 情報 編集 削除 キャッシュポリシーを作成

[補足] キャッシュコントロール機能



CloudFront のキャッシュコントロールにおける注意事項

- GET / HEAD / OPTION (選択可能) のリクエストがキャッシュ対象
(CloudFront はその他のメソッドを使用するリクエストへのレスポンスをキャッシュしない)
 - 単一リクエストで取得できるキャッシュの最大サイズは 30GB (範囲リクエストを使用しない場合)
※範囲リクエストを使用することで 30GB を超えるオブジェクトを分割してキャッシュが可能
 - URL および Cache Policy で有効化した HTTP ヘッダー、クエリ文字列、
Cookie パラメータ値の完全一致でキャッシュが再利用される
- ※ **Cache Policy** にて、全ての TTL の値が 0 の場合は
CloudFront ではキャッシュをしない
(マネージドポリシーの **CachingDisabled** の利用)

再掲

- 「同じ名前を使用してファイルを更新する」際の話
- 大量に付与されているメタ情報の何をキーにして CloudFront は
“同じ名前のファイルを別のキャッシュとして保管する”のか？

重要！



Cache Policy
(キャッシュポリシー)

CloudFront がオブジェクトの部分的リクエスト (レンジ GET) を処理する方法
https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/RangeGETs.html



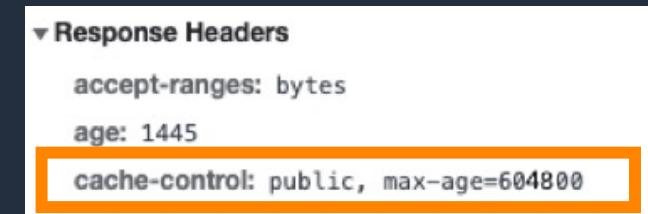
CloudFront の TTL と オリジン側の Cache-Control ヘッダーの関係について

Q. 例えば下記の各々の設定をした際にコンテンツはどれくらいの間
CloudFront にキャッシュされるだろうか？

「*.jpg」のパスに対するビヘイビアのキャッシュポリシーの設定



Apache HTTP Server の
Cache-Control ヘッダーの設定

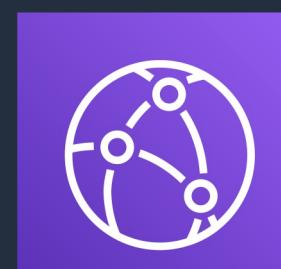


① 「img.jpg」を初めてリクエスト



④ 「img.jpg」をレスポンス

クライアント
(ビューウィー)



Amazon
CloudFront
img.jpg

② 「img.jpg」を初めてリクエスト



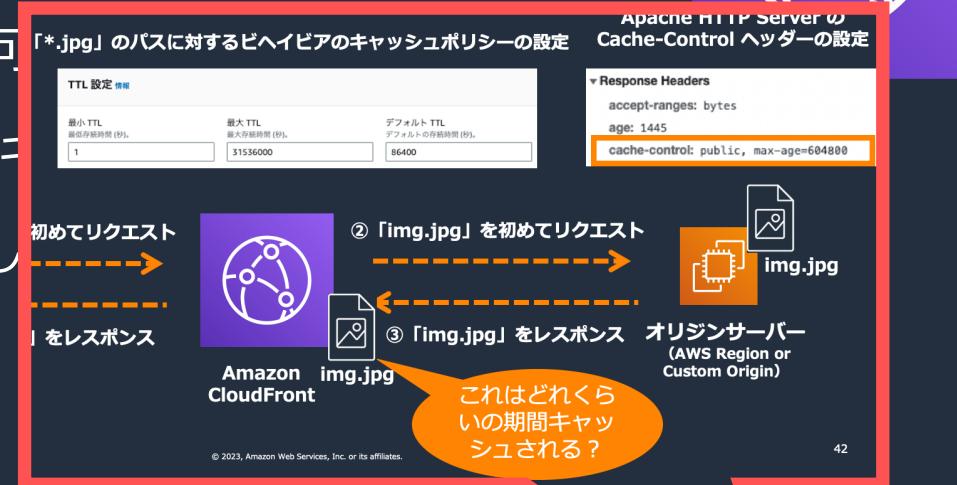
③ 「img.jpg」をレスポンス オリジンサーバー
(AWS Region or Custom Origin)

これはどれくらいの期間キャッシュされる？

CloudFront 側のキャッシングについて



- オリジンの Cache-Control ヘッダーでキャッシング時間の設定が可
 - Behavior 毎に異なる設定を行うことで、URL パスパターン毎に設定可能
- デフォルト TTL : オリジンが Cache-Control ヘッダーを指定しない場合
- 最小 TTL : CloudFront でキャッシングすべき最小期間
- 最大 TTL : CloudFront でキャッシングすべき最大期間



Cache Policy 最小 TTL 設定

オリジン HTTP ヘッダー	最小 TTL = 0 秒		最小 TTL > 0 秒を設定	
	Cache-Control max-age を指定	指定された max-age と最大 TTL で小さい値の期間キャッシング	最小 TTL < max-age < 最大 TTL	max-age 期間
Cache-Control 設定なし		デフォルト TTL 期間キャッシング	max-age < 最小 TTL	最小 TTL 期間
			最大 TTL < max-age	最大 TTL 期間

CloudFront 側のキャッシュについて



Q. 「Cache-Control : max-age/s-maxage/no-cache/no-store /private」、「Expires」HTTP ヘッダーとは何者なのか？

ヘッダーと HTTP リクエスト

Cache-Control max-age と s-maxage を指定

Expires を指定

Cache-Control no-cache, no-store、および(または) private ディレクティブを追加

Cache Policy 最小 TTL 設定

	最小TTL = 0秒	最小TTL > 0秒を設定	
Cache-Control max-age と s-maxage を指定	指定された s-max-age と最大 TTL で小さい値の期間キャッシュ	最小TTL < s-max-age < 最大 TTL	s-max-age 期間
Expires を指定	指定された Expires 日付と最大 TTL で早い日付の期間キャッシュ	s-max-age < 最小 TTL	最小 TTL 期間
Cache-Control no-cache, no-store、および(または) private ディレクティブを追加	ヘッダーを優先させる	最大 TTL < s-max-age	最大 TTL 期間
		最小 TTL < 最大 TTL	Expires 日付
		Expires < 最小 TTL	最小 TTL 期間
		最大 TTL < Expires	最大 TTL 期間
		最小 TTL の期間キャッシュ	

※ S3 がオリジンの場合は S3 オブジェクト Metadata に Cache-Control, Expires を指定可能

コンテンツがキャッシュに保持される期間(有効期限)の管理

aws https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/Expiration.html

© 2023, Amazon Web Services, Inc. or its affiliates.

オリジンサーバー側の「Cache-Control」および「Expires」HTTP ヘッダーの設定

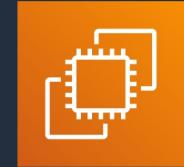
Apache HTTP Server (ver 2.4) での設定例

- **Headers モジュール [mod_headers]**

```
Header set Cache-Control "public,max-age=604800"
```

- **Expires モジュール [mod_expires]**

```
ExpiresByType text/css "access plus 1 month 15 days 2 hours"
```



オリジンサーバー
(AWS Region or
Custom Origin)

- **Cache-Control ヘッダー**

- max-age . . . レスポンスが生成されてから N 秒後まで、レスポンスが新鮮なままであることを示す。
- no-cache . . . キャッシュに保存できることを示す。キャッシュがオリジンサーバーから切断された場合でも、再利用の前にオリジンサーバーで検証を行うことを指示する。
- no-store . . . あらゆる種類のキャッシュが、このレスポンスを保存しないようにすることを指示する。

- **Expires ヘッダー** . . . レスポンスが古くなると見なされる日時を指定する。

※Cache-Control max-age と Expires の両方の値を指定した場合、CloudFront は 前者の値のみを使用。

Apache HTTP Server バージョン 2.4 ドキュメント
https://httpd.apache.org/docs/2.4/ja/mod/mod_expires.html

MDN Web Docs (Cache-Control)
<https://developer.mozilla.org/ja/docs/Web/HTTP/Headers/Cache-Control>

Amazon CloudFront を利用した キャッシュの運用における Tips

CloudFront の設定イメージ

Cache Policy

- キャッシュポリシー 1
 - TTL設定
 - キャッシュキー設定 (Headers/Cookies/Query Strings)
 - 圧縮サポート(Gzip/Brotli)
- Managed-*:マネージドキャッシュポリシー

Origin Request Policy

- オリジンリクエストポリシー 1
 - オリジンリクエストコンテンツ (Headers/Cookies/Query Strings)
- Managed-*: マネージドオリジンリクエストポリシー

Distribution



AWS Certificate Manager



Amazon CloudFront
(xxxxxxxx.cloudfront.net)

Behavior

設定1

image/*
キャッシュポリシー1
オリジンリクエストポリシー1

設定2

api
キャッシュポリシー2
オリジンリクエストポリシー2

設定3

デフォルト(*)
キャッシュポリシー3
オリジンリクエストポリシー3

Origin

設定A

{bucket-name}.s3.
{region}.amazonaws.com

設定B

{restapi-id}.execute-api.
{region}.amazonaws.com

設定C

{elb-name}-1234567890.
{region}.elb.amazonaws.com

オリジンサーバー



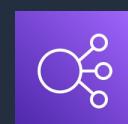
Amazon S3



Amazon API
Gateway



AWS Lambda



Application
Load Balancer



Amazon EC2

[補足] Cache Policy と Origin Request Policy の連携

- オリジンに転送するリクエストとキャッシュキーを分離して取り扱うことにより、より柔軟なキャッシュ設定が可能
- 事前定義済みのマネージドポリシーの他に、カスタムポリシーの作成・適用が可能

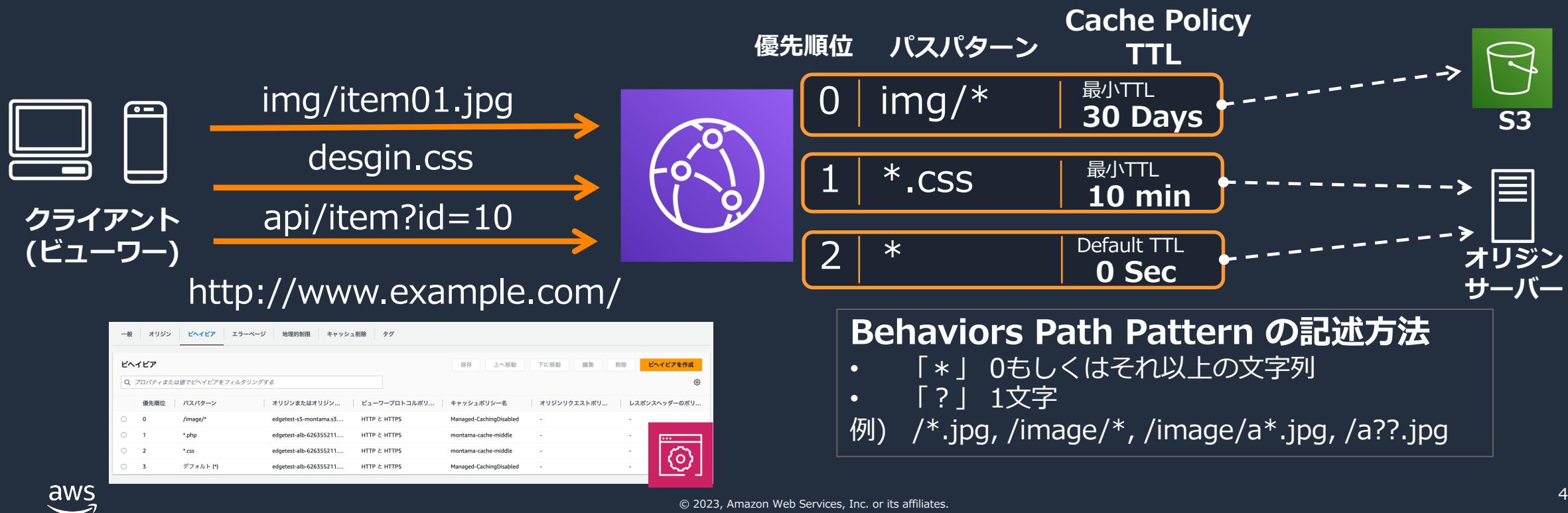


※ Origin Request Policy の設定画面



きめ細やかなキャッシングの実現

- Cache Policy / Origin Request Policy を組み合わせ、HTTP ヘッダー, Cookie, クエリ文字列をオリジンリクエストに含めるか否かをパスパターン毎に自由にカスタマイズできる
- クライアントのリクエストパターンをもとに、複数の URL パスパターンの Behavior とマルチオリジンを組み合わせ、きめ細かなキャッシングコントロールを実現



キャッシュファイルの無効化 (Invalidation)

- ・ コンテンツ毎の無効化パス指定
同時に最大 3,000 個までのパス指定が可能
- ・ ワイルドカードを利用した無効化パス指定
 - ・ 同時に最大 15 個まで無効化パスリクエストが指定可能
 - ・ オブジェクト数の制限無し
- ・ AWS Management Console / CLI / SDK で実行可能
- ・ キャッシュファイルの無効リクエストは有償のため、オリジンレスポンスヘッダーの `Cache-Control: max-age` または `s-maxage` や `Expires` の指定、キャッシュポリシーのデフォルト TTL で適切なキャッシュ期間を設定することを推奨
 - ・ 最初の 1,000 パスまでは追加料金無し、それ以降は、無効をリクエストしたパスごとに \$0.005

無効化のクォータ

エンティティ	デフォルトのクォータ
ファイルの無効化: ワイルドカードの無効化を除く、アクティブな無効化リクエストで許可されるファイルの最大数	3,000
詳細については、「 ファイルの無効化 」を参照してください。	
ファイルの無効化: 許可されるアクティブなワイルドカード無効化の最大数	15
ファイルの無効化: 1 つのワイルドカードの無効化で処理できるファイルの最大数	クォータなし

キャッシュ削除を作成

オブジェクトパス

オブジェクトパスを追加
CloudFront キャッシュから削除する各オブジェクトのパスを追加します。ワイルドカード (*) を使用できます。

② オブジェクトパスを個別に追加するには、[標準エディタ](#) を使用します。

キャンセル

キャッシュ削除を作成

ファイルの無効化



https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/Invalidation.html

© 2023, Amazon Web Services, Inc. or its affiliates.

CloudFront がオリジンからの HTTP 4xx および 5xx ステータスコードを処理してキャッシュする方法について

- エラーキャッシュの最小 TTL はデフォルトで 10秒 である
 - カスタムエラーレスポンスの定義で変更可能
- 4xx Error に関しては、ステータスコードによってキャッシュの挙動が異なる
 - 一部に関しては Cache-Control ヘッダーの max-age または s-maxage が返ってきたときのみキャッシュする
- CloudFront に既存のキャッシュが残っているか(有効期限切れを含む)、否かで CloudFront のレスポンスの挙動が変わる
- 5xx Error か 4xx Error かでキャッシュの有効期限が切れた際に、CloudFront が保持しているキャッシュをそのままレスポンスするか否かが異なる
- カスタムエラーレスポンス用のページが設定されているか否かで挙動が変わる

CloudFront が常にキャッシュする
HTTP ステータスコード

404	Not Found
414	Request-URI Too Large
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Time-out

Cache-Control ヘッダーに基づいて
キャッシュする HTTP ステータスコード

400	Bad Request
403	Forbidden
405	Method Not Allowed
412	Precondition Failed
415	Unsupported Media Type

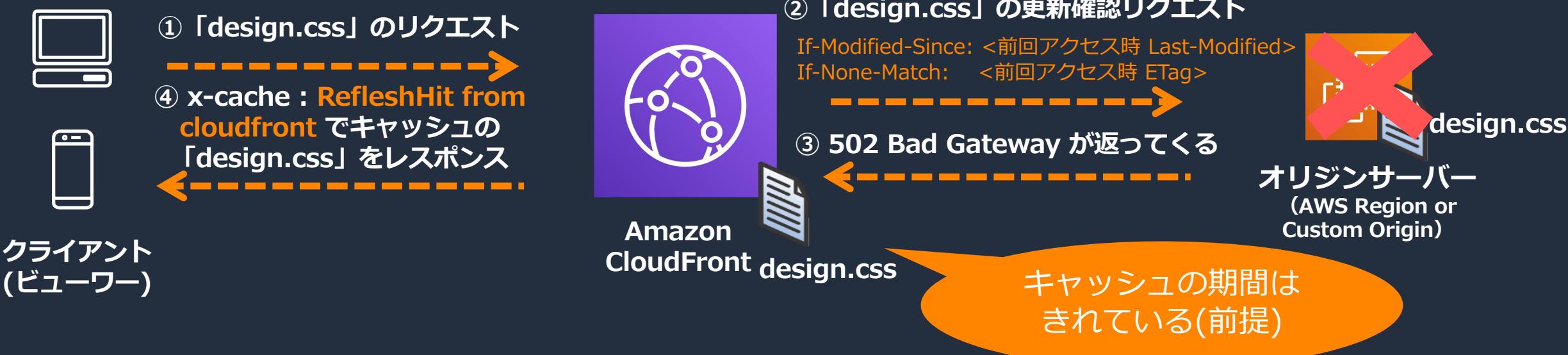
CloudFront がオリジンからの HTTP 4xx および 5xx ステータスコードを処理してキャッシュする方法について

具体的なフロー (キャッシュが CloudFront に既にある場合)

※下記は 5xx Error のフロー なので注意

→ 4xx Error の場合、リクエストされたオブジェクトではなくステータスコードをビューワーに返す

“CloudFront にあるコンテンツのキャッシュは期間がきれているが、レスポンスとして返ってくる”
(これを回避するには「Cache-Control: stale-if-error=0」を含めるようにする必要がある)



CloudFront がオリジンからの HTTP 4xx および 5xx ステータスコードを処理してキャッシュする方法について

具体的なフロー (キャッシュが CloudFront に元々無い場合)

※エラーが CloudFront にキャッシュされる
(下記の工程の⑥と⑦は エラーのキャッシュがある間は、502 Bad Gateway を返し続ける)



コンテンツがキャッシュに保持される期間 (有効期限) の管理

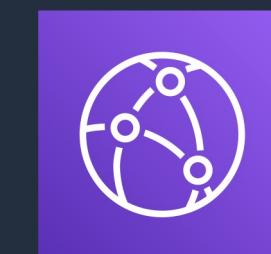
https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/Expiration.html

© 2023, Amazon Web Services, Inc. or its affiliates.

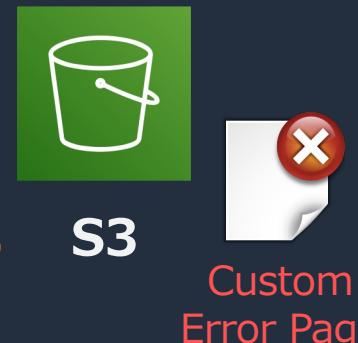
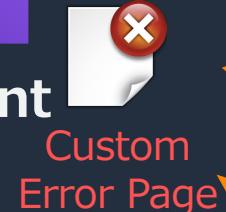
エラーレスポンス動作のカスタマイズ (Custom Error Page)



予め、4xx, 5xx のカスタム
レスポンスを設定(エラーぺ
ージの遷移先を S3 に設定)



CloudFront



- CloudFront は、エラーレスポンスをデフォルト 10 秒キャッシュ
- 4xx および 5xx ステータスコードそれぞれに対して、
 - エラーキャッシュ期間 (エラーキャッシュ最小 TTL) は 0 秒以上を指定可能、レスポンスヘッダーに Cache-Control: max-age または s-maxage や Expires を指定しオブジェクト毎にカスタマイズも可能
 - エラーレスポンスページおよびステータスコードのカスタマイズが可能

キャッシュの挙動を確認するための方法について(ブラウザ編)

x-cache レスポンスヘッダー

- CloudFront がレスポンス返す時に生成してくれるヘッダー
- CloudFront で保持しているキャッシュの状況に合わせて x-cache の内容が変わる

代表的な x-cache の内容

- | | |
|------------------------------|--|
| • Miss From cloudfront | • • • 現在 CloudFront にキャッシュが無い状態なので、オリジンサーバーにコンテンツを取りに行って返した |
| • Hit from cloudfront | • • • CloudFront にキャッシュがあったので、それを返した |
| • RefreshHit from cloudfront | • • • CloudFront にキャッシュはあったが、有効期限が切れたり、オリジンサーバーに最新か確認した後に返した |

とあるコンテンツのレスポンスヘッダ
(ブラウザの開発者ツールより。)

```
▼ Response Headers
accept-ranges: bytes
content-length: 5347
content-type: image/png
date: Tue, 07 Mar 2023 04:49:27 GMT
etag: "33dbdd0177549353eeeb785d02c294af"
last-modified: Sat, 11 Feb 2023 14:30:17 GMT
server: AmazonS3
via: 1.1 18fb8bbcd8ce7c8581681ccc40c56f10.cloudflare.net (CloudFront)
x-amz-cf-id: -1jzFdzrUwgHQfYwMo8CM0vQ0D978PA1eWpkf09E15SZE3I5gacQ2g==
x-amz-cf-pop: NRT57-P3
x-amz-server-side-encryption: AES256
x-cache: Miss from cloudfront
```

キャッシュの挙動を確認するための方法について(ログ編)

x-edge-result-type / x-edge-response-result-type

- HIT

CloudFront がキャッシュからビューワーにオブジェクトを渡したことを示す

- RefreshHit

CloudFront のキャッシュにてオブジェクトを検出したが、キャッシュの有効期限が切れていたため、オリジンに問い合わせて、最新バージョンのオブジェクトがあるかどうかを確認したことを示す

- Miss

CloudFront のキャッシュにあるオブジェクトでリクエストに対応できなかったため、リクエストをオリジンサーバーに転送して結果をビューワーに返したことを示す

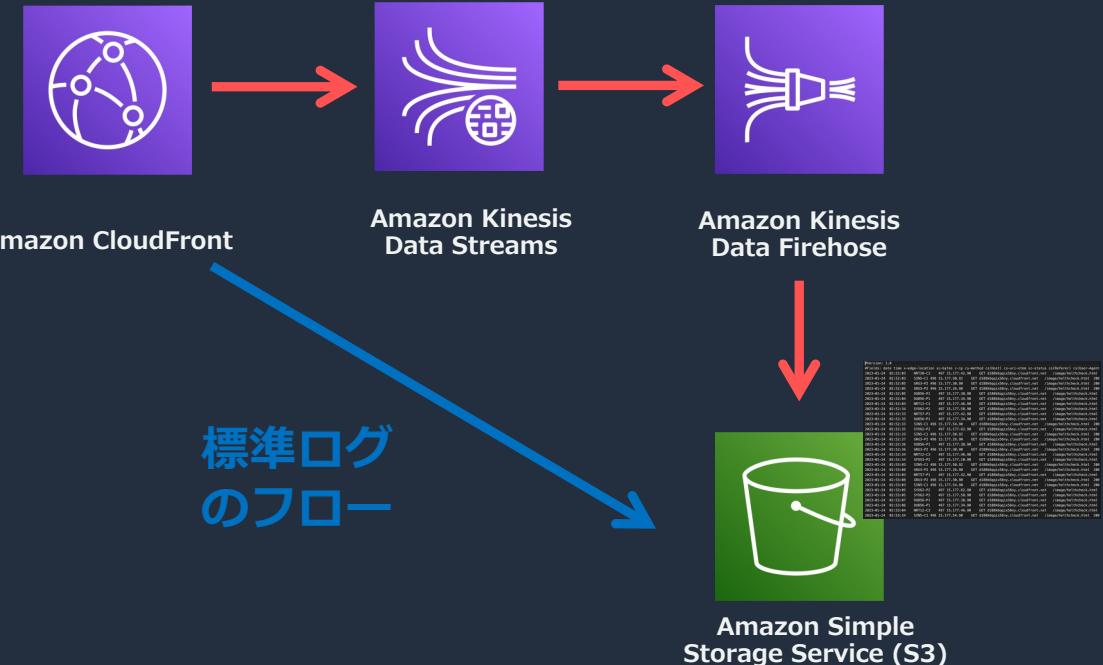
※上記以外にも LimitExceeded、CapacityExceeded、Error、Redirect などの値がある

標準ログ (アクセスログ) の設定および使用

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/AccessLogs.html



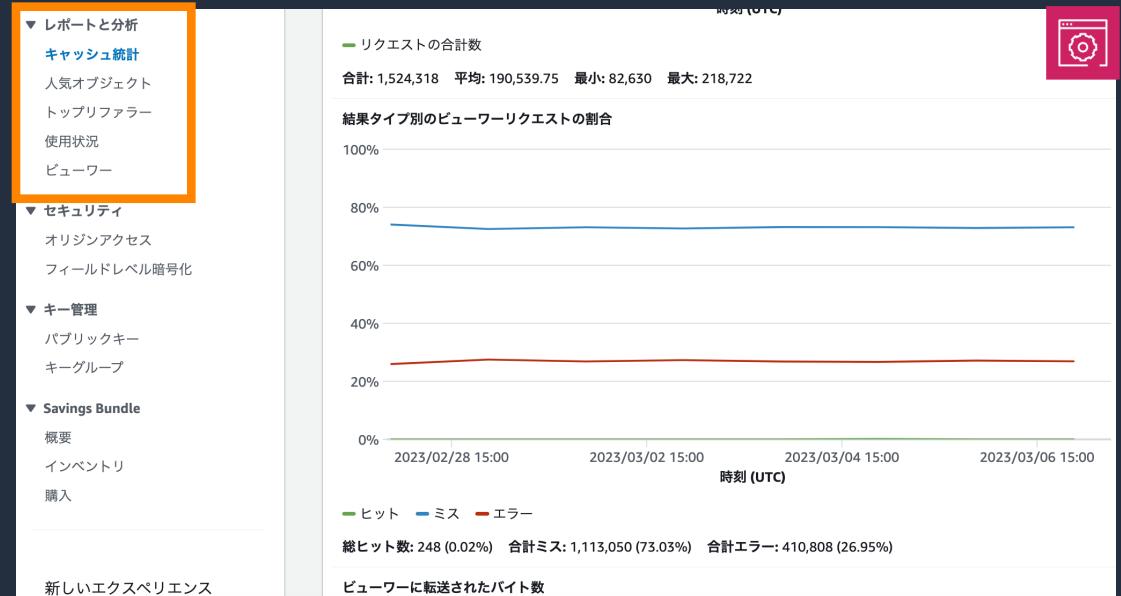
リアルタイムログのフロー



CloudFront レポートと分析(マネージメントコンソール)

マネージメントコンソールにてデフォルトで様々な情報が確認できるようになっている

- キャッシュ統計
 - キャッシュの統計情報
- 人気オブジェクト
 - 人気コンテンツの統計情報
- トップリファラー
 - リファラーの統計情報
- 使用状況
 - リクエスト数およびデータ転送量
- ビューウー
 - クライアントデバイスの統計情報



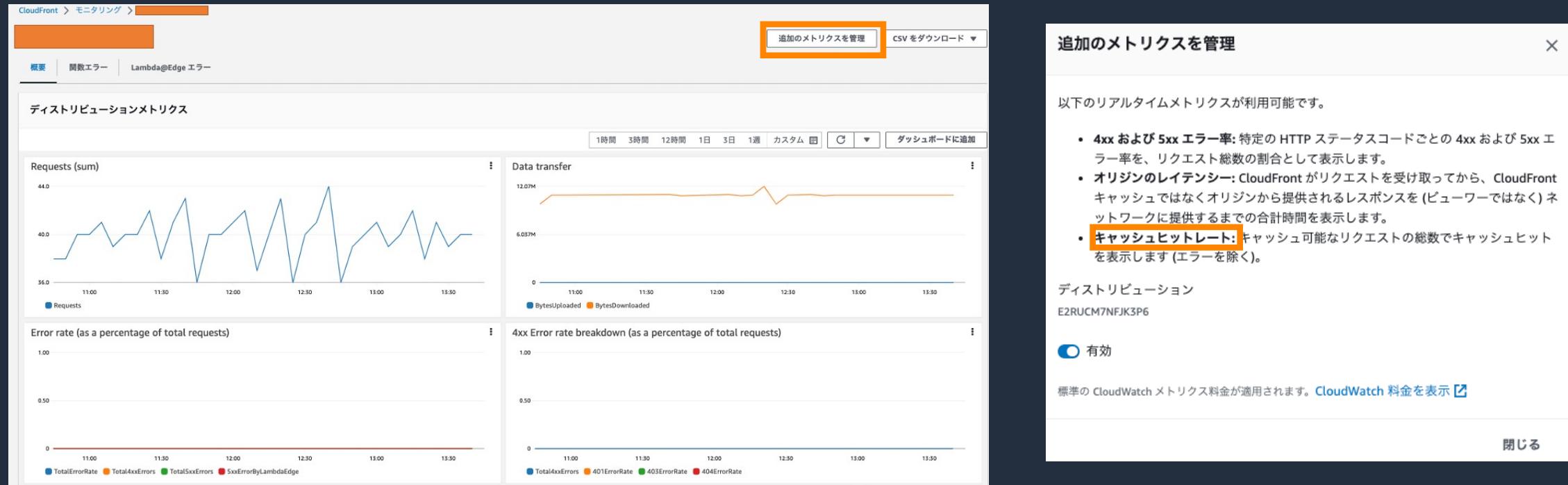
キャッシュ統計 / 人気オブジェクト /
トップリファラー / 使用状況 / ビューウーは
AWS Management Console のみで参照可能

コンソールの CloudFront レポート

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/reports.html

CloudWatch で管理する Cache hit rate メトリクスについて

デフォルトでは CloudFront のキャッシュヒットレートは CloudWatch のメトリクスとして取得されない（マネージメントコンソールのモニタリングの画面から追加が可能）



CloudFront 関数およびエッジ関数のメトリクスの表示

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/viewing-cloudfront-metrics.html

クライアント側のキャッシュについて

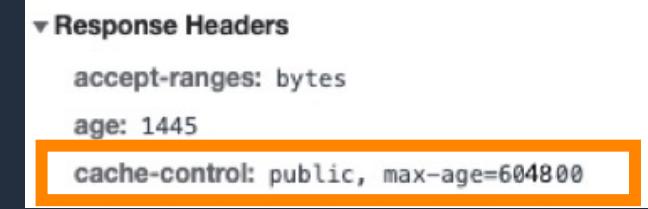
CloudFront の TTL と オリジン側の Cache-Control ヘッダーの関係について

Q. クライアント側にもキャッシュはされる、これはどの期間キャッシュされるのか？

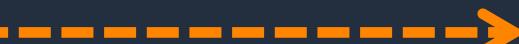
「*.jpg」のパスに対するビヘイビアのキャッシュポリシーの設定



Apache HTTP Server の
Cache-Control ヘッダーの設定



① 「img.jpg」を初めてリクエスト



④ 「img.jpg」をレスポンス

img.jpg

クライアント
(ビューウィー)



これはどれくらいの期間キャッシュされる？



Amazon
CloudFront

② 「img.jpg」を初めてリクエスト



③ 「img.jpg」をレスポンス オリジンサーバー
(AWS Region or Custom Origin)

クライアント側のキャッシュについて



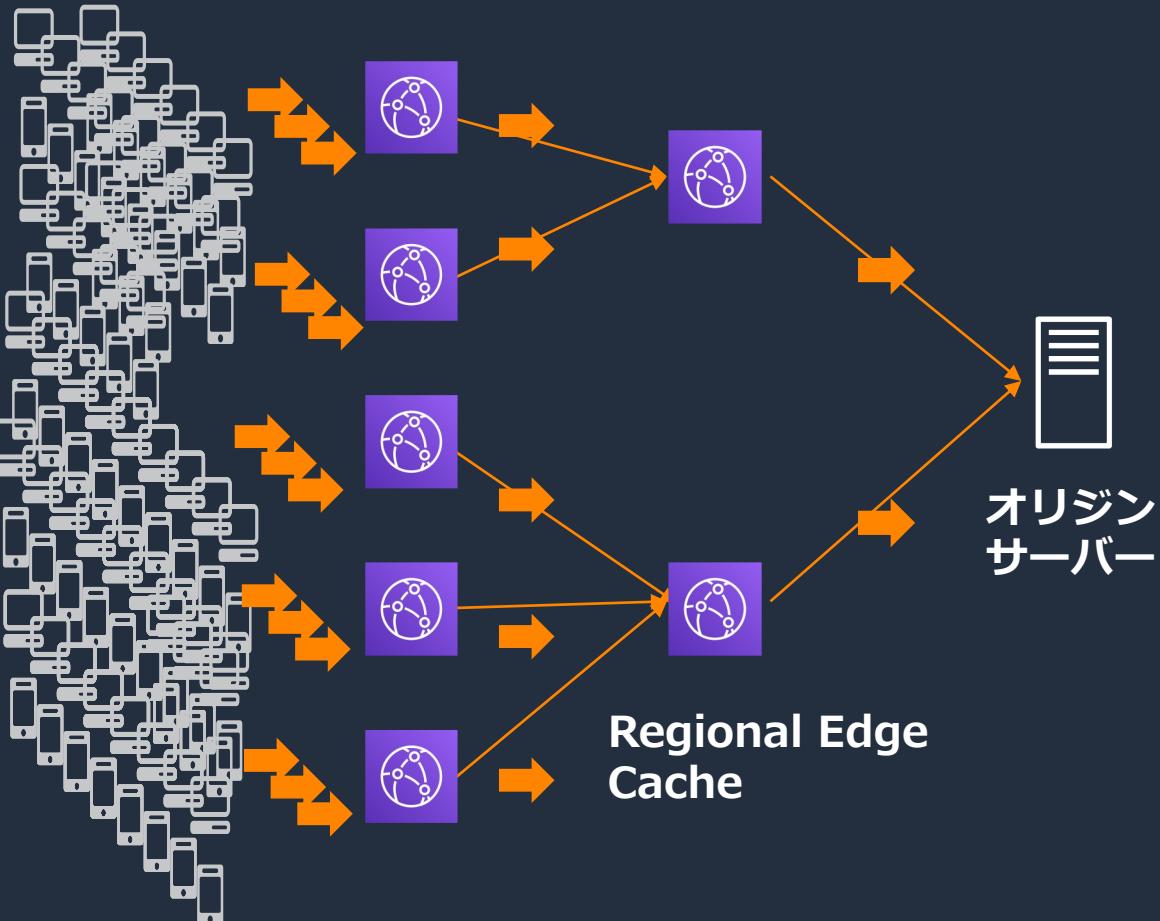
Cache Policy Minimum TTL 設定

オリジン HTTP ヘッダー	Cache Policy Minimum TTL 設定	
	最小 TTL = 0 秒	最小 TTL > 0 秒を設定
	Cache-Control max-age を指定	Cache-Control: max-age ディレクティブの値に 対応する期間、オブジェクトをキャッシュ
	Cache-Control 設定なし	ブラウザによって異なる
	Cache-Control max-age と s-maxage を指定	Apache HTTP Server の Cache-Control ヘッダーの設定
	Expires を指定	max-age ディレクティブの値 オブジェクトをキャッシュ
Cache-Control no-cache, no-store 、および (または) private ディレクティブを追加	Cache	① 「img.jpg」を初めてリクエスト ② 「img.jpg」を初めてリクエスト ③ 「img.jpg」をレスポンス オリジンサーバー (AWS Region or Custom Origin) ④ 「img.jpg」をレスポンス これはどれくらいの期間キャッシュされる?

その他

オリジンインフラの保護

Automatic Flash Crowd Protection

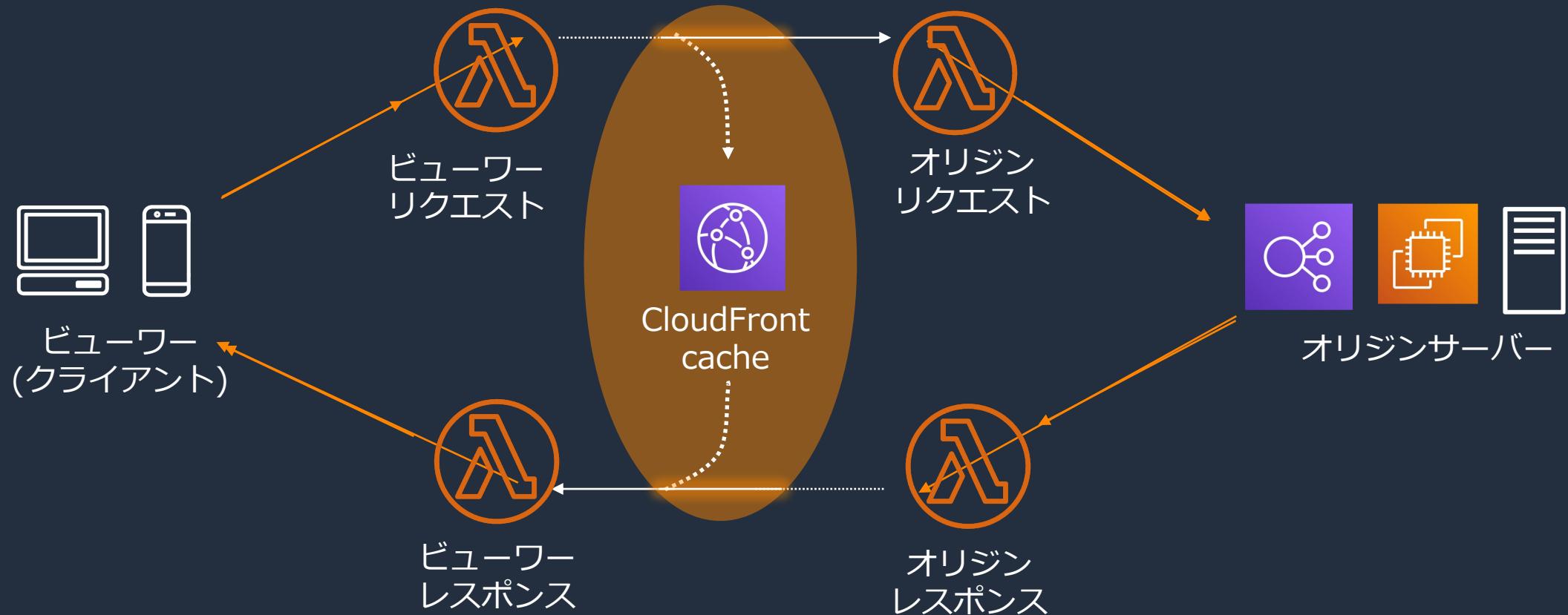


同一オブジェクトへの同時リクエスト（リクエストを折りたたむ）

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/RequestAndResponseBehaviorS3Origin.html

CloudFront のキャッシュヒットした時の Lambda@Edge の挙動

Q. CloudFront のキャッシュがヒットした際に、Lambda@Edge は実行されるのか？



Lambda 関数をトリガーできる CloudFront イベント

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/lambda-cloudfront-trigger-events.html

キャッシングに関する CloudFront のクオータ

キャッシングに関する CloudFront のクオータは、主にキャッシングポリシーに関するものである(下記抜粋)

- ・作成できるキャッシングポリシーの数
- ・キャッシングポリシーあたりのクエリ文字列
- ・キャッシングポリシーあたりのヘッダー
- ・キャッシングポリシーあたりの Cookie

など

ポリシーの一般的なクオータ	
	デフォルトのクオータ
エンティティ	20
AWS アカウントあたりのキャッシングポリシー	100
同じキャッシングポリシーに関連付けられたディストリビューション	10
キャッシングポリシーあたりのクエリ文字列	クォータ引き上げのリクエスト
キャッシングポリシーあたりのヘッダー	10
キャッシングポリシーあたりの Cookie	クォータ引き上げのリクエスト
キャッシングポリシー内のすべてのクエリ文字列、ヘッダー、および Cookie 名の合計長	1024
AWS アカウントあたりのオリジンリクエストポリシー	20
同じオリジンリクエストポリシーに関連付けられたディストリビューション	100
オリジンリクエストポリシーあたりのクエリ文字列	10
オリジンリクエストポリシーあたりのヘッダー	クォータ引き上げのリクエスト
オリジンリクエストポリシーあたりの Cookie	クォータ引き上げのリクエスト
キャッシングリクエストポリシー内のすべてのクエリ文字列、ヘッダー、および Cookie 名の合計長	1024
AWS アカウントあたりのレスポンスヘッダーポリシー	20
同じレスポンスヘッダーポリシーに関連付けられたディストリビューション	100
レスポンスヘッダーポリシーごとのカスタムヘッダー	10
AWS アカウントあたりの継続的デプロイポリシー	クォータ引き上げのリクエスト
	クォータ引き上げのリクエスト



クオータ

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/cloudfront-limits.html

© 2023, Amazon Web Services, Inc. or its affiliates.

参考 URL

メディア配信におけるキャッシュの活用方法

- ・ ビデオオンデマンド (VOD) を配信
https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/on-demand-video.html
- ・ ライブストリーミングビデオの配信
https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/live-streaming.html

AWS Hands-on for Beginners

- ・ AWS 上で静的な Web サイトを公開しよう !
https://pages.awscloud.com/JAPAN-event-OE-Hands-on-for-Beginners-CF_WAF-2022-reg-event.html
- ・ Amazon CloudFrontおよびAWS WAFを用いて エッジサービスの活用方法を学ぼう
https://pages.awscloud.com/JAPAN-event-OE-Hands-on-for-Beginners-CF_WAF-2022-reg-event.html

まとめ

- ✓ CloudFront は AWS のグローバルレインフラストラクチャを利用した CDN (Content Delivery Network) サービス
- ✓ キャッシュの活用は CloudFront を利用するメリットの一つ
- ✓ Web サイトのコンテンツには沢山の種類が、あるがその殆どが静的コンテンツ
- ✓ コンテンツキャッシュの運用は、コンテンツ毎に誰がいつどうやって行うのかを決めた上で検討する
- ✓ CloudFront でキャッシュを活用する際には Cache Policy で設定し、オリジン側の Cache Control の設定もキャッシュ期間に影響する
- ✓ エラーレスポンスについても CloudFront にキャッシュをさせることができる

本資料に関するお問い合わせ・ご感想

技術的な内容に関しては、有料の AWS サポート窓口へ
お問い合わせください

<https://aws.amazon.com/jp/premiumsupport/>

料金面でのお問い合わせに関しては、カスタマーサポート窓口へ
お問い合わせください（マネジメントコンソールへのログインが必要です）

<https://console.aws.amazon.com/support/home#/case/create?issueType=customer-service>

具体的な案件に対する構成相談は、後述する個別相談会をご活用ください



ご感想は Twitter へ！ハッシュタグは以下をご利用ください
#awsblackbelt



その他コンテンツのご紹介

ウェビナーなど、AWS のイベントスケジュールをご参照いただけます

<https://aws.amazon.com/jp/events/>

ハンズオンコンテンツ

<https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-hands-on/>

AWS 個別相談会

AWS のソリューションアーキテクトと直接会話いただけます

<https://pages.awscloud.com/JAPAN-event-SP-Weekly-Sales-Consulting-Seminar-2021-reg-event.html>



Thank you!



Amazon CloudFront

(レポート / モニタリング / ロギング編)

AWS Black Belt Online Seminar

長谷川 純也

Solutions Architect
2023/08

AWS Black Belt Online Seminarとは

- ・ 「サービス別」「ソリューション別」「業種別」などのテーマに分け、
アマゾン ウェブ サービス ジャパン合同会社が提供するオンラインセミナー
シリーズです
- ・ AWS の技術担当者が、AWS の各サービスやソリューションについてテーマ
ごとに動画を公開します
- ・ 動画を一時停止・スキップすることで、興味がある分野・項目だけの聴講も
可能、スキマ時間の学習にもお役立ていただけます
- ・ 以下の URL より、過去のセミナー含めた資料などをダウンロードすることができます
 - ・ <https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-service-cut/>
 - ・ <https://www.youtube.com/playlist?list=PLzWGOASvSx6FIwIC2X1nObr1KcMCBBlqY>

内容についての注意点

- ・ 本資料では 2023 年 07 月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 (<https://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には別途消費税をご請求させていただきます

自己紹介

名前：長谷川 純也（はせがわ じゅんや）

所属：アマゾン ウェブ サービス ジャパン合同会社

技術統括本部 エンタープライズ技術本部

通信・メディアグループ

メディアソリューション部

経歴：外資系 CDN/WAF ベンダーのプリセールスエンジニアを経て、
2019 年より現職、現在はメディア系企業のお客様の技術支援を担当

好きなAWSサービス：Amazon CloudFront , AWS WAF , Amazon Route 53



本セミナーの対象者

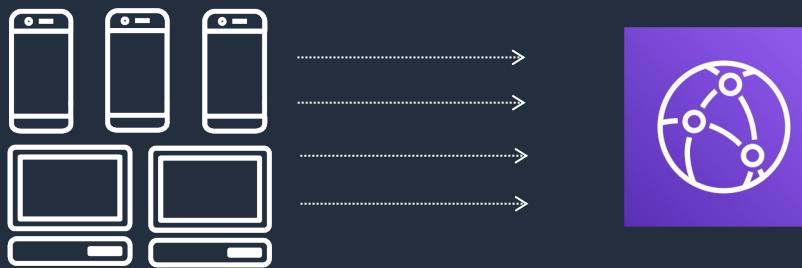
- 基本的な AWS サービスについて理解されている方
- Amazon CloudFront の概要を理解されている方
- Amazon CloudFront を現在運用中の方
- Amazon CloudFront の監視やロギングの仕組みについてご興味のある方

アジェンダ

- Amazon CloudFront のレポート & モニタリング機能
- Amazon CloudFront のロギング機能
- まとめ

レポート & モニタリング機能

Amazon CloudFront レポート & モニタリング機能概要



クライアント



レポーティング

キャッシュ統計
人気オブジェクト
トップリファラー
使用状況
ビューウェー

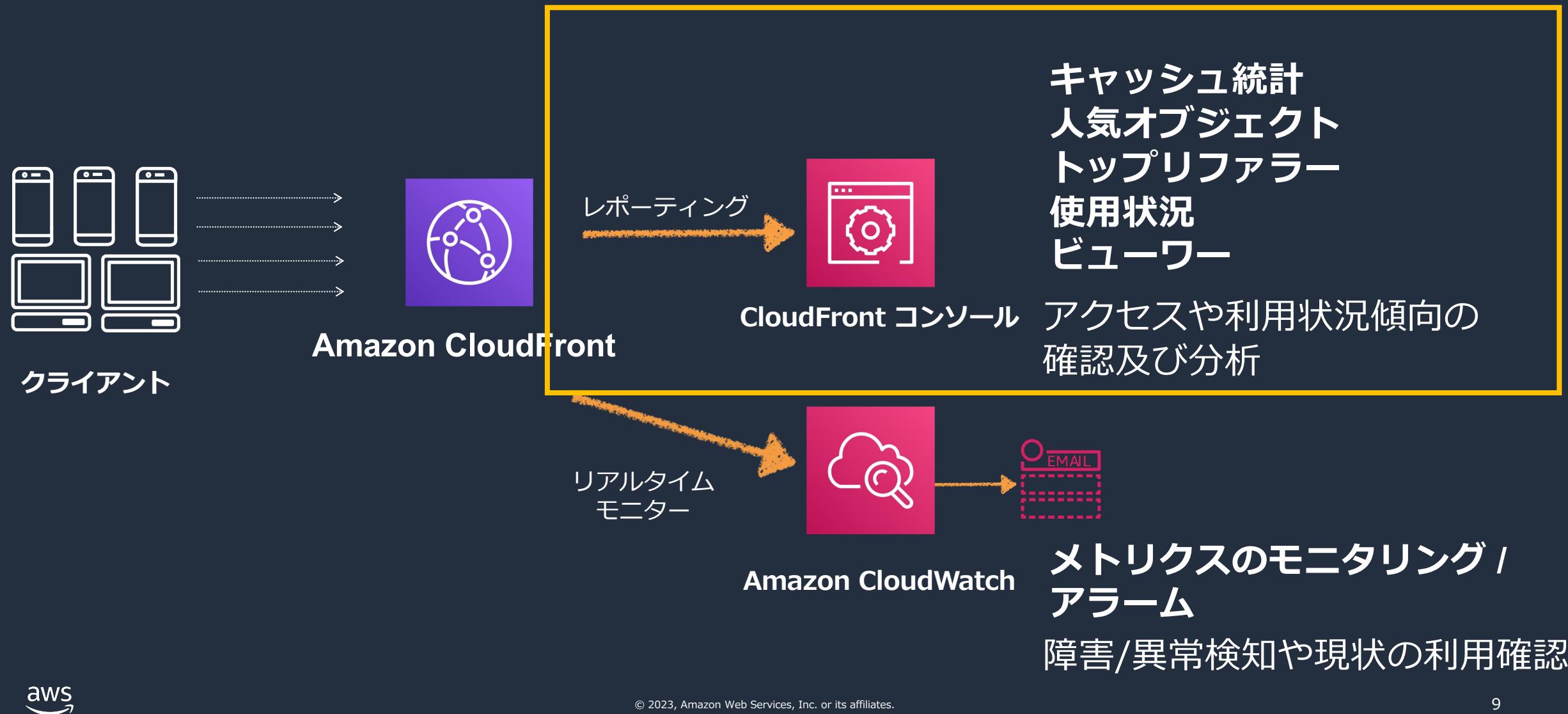
アクセスや利用状況傾向の
確認及び分析



リアルタイム
モニター

メトリクスのモニタリング /
アラーム
障害/異常検知や現状の利用確認

Amazon CloudFront レポート 機能



CloudFront コンソール：レポートと分析

CloudFront の利用状況における傾向分析として利用

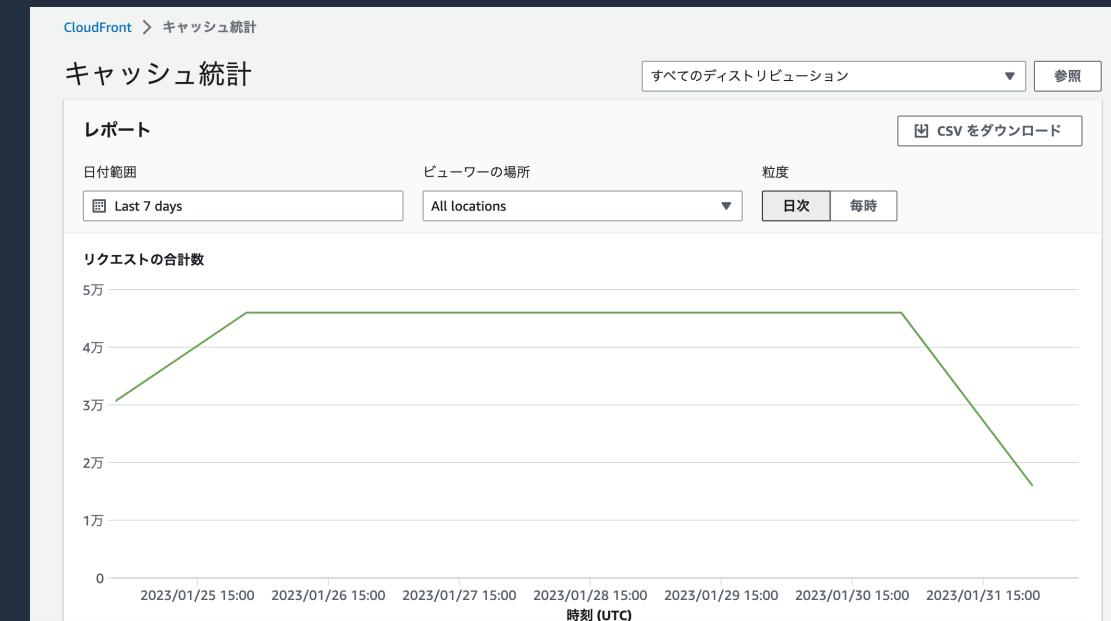
- キャッシュ統計 / 人気オブジェクト / トップリファラー / 使用状況 / ビューウィーの情報が確認可能
- 過去 60 日間までのデータを保持し、グラフで参照可能
- 1 時間単位もしくは日単位でのグラフ表示 (UTC)
- CSV へのエクスポートも可能
- フィルタリング
 - 全ディストリビューションもしくは ディストリビューション単位
 - 期間指定
 - エッジ地域
 - ビューウィーの場所
 - 請求リージョンなど



CloudFront コンソール：レポートと分析

キャッシュ統計

- リクエストの合計数
 - すべての HTTP ステータスコードとメソッドを含むリクエスト総数
- 結果タイプ別のビューワーリクエストの割合
 - キャッシュのヒット、ミス、エラーの割合
- ビューワーに転送されたバイト数
 - ビューワーに提供された合計バイト数
 - ミスヒットしたリクエストに対するバイト数
- HTTP ステータスコード
 - 2XX, 3XX, 4XX, 5XX 毎の応答数
- ダウンロードを完了しなかった GET リクエストの割合
 - ダウンロードを完了出来なかった GET リクエストの割合



CloudFront コンソール：レポートと分析

人気オブジェクト

ディストリビューション毎のリクエスト数の多いTop 50コンテンツリスト

- オブジェクト
 - オブジェクトへのアクセスにビューウィーが使用する URL の末尾 500 文字
- リクエスト
- ヒット, ヒット %, ミス
- 合計バイト数, ミスからのバイト
- 不完全なダウンロード
- HTTP ステータスコード
 - 2xx,3xx,4xx,5xx

The screenshot shows the 'Popular Objects' report in the CloudFront console. The report lists the most popular objects based on request count over the last 7 days. The columns include Object, Request, Hit, Hit %, Miss, Miss bytes, Total bytes, and Incomplete downloads. The top object is '/inputform.html'.

オブジェクト	リクエスト	ヒット	ヒット %	ミス	ミスからのバイト	合計バイト数	不完全なダウンロー
/inputform.html	340,994	42,554	12.48%	55	152.16 KB	494.52 MB	5

CloudFront コンソール：レポートと分析

トップリファラー

ディストリビューション毎のリクエスト数の多い Top 25 のリファラー

- リファラー
 - リファラーのドメイン名
- リクエスト数
 - リファラー列のドメイン名からのリクエストの総数
- リクエストの割合
 - 指定した期間のリクエストの総数に対してリファラーによって送信されたリクエストの数の割合

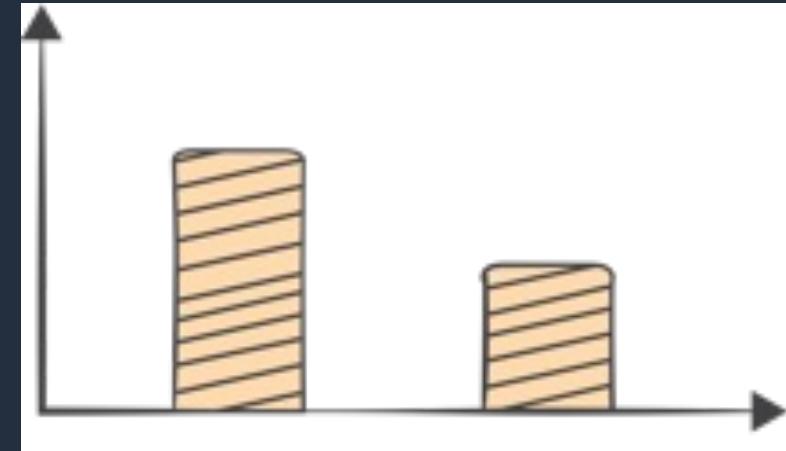


CloudFront コンソール：レポートと分析

使用状況

使用状況をグラフィカルに表示

- プロトコル別データ転送量
 - 各リージョンのエッジロケーションから転送されたデータの合計量を、プロトコル別（HTTP または HTTPS）、に分けて表示
- 送信先別データ転送量
 - CloudFront から送信先別（ユーザーまたはオリジン）の転送量
- リクエスト数
 - 指定されたディストリビューションごとの CloudFront が応答したリクエストの総数をプロトコル別（HTTP または HTTPS）、に分けて表示
- フィールドレベル暗号化リクエスト数
 - フィールドレベル暗号化を使用したリクエスト数

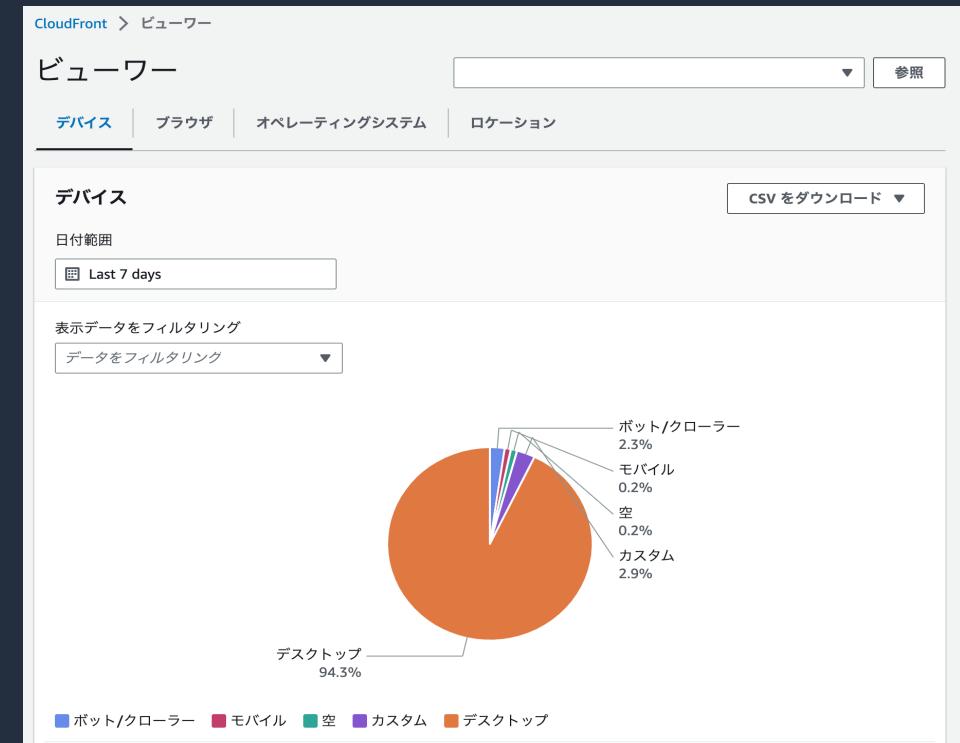


CloudFront コンソール：レポートと分析

ビューウー

物理デバイス（デスクトップコンピュータ、モバイルデバイス）およびコンテンツにアクセスするビューウナー（通常はウェブブラウザ）に関するレポートを表示

- デバイス
 - デバイス種別の比率と傾向
- ブラウザ
 - ブラウザ種別の比率と傾向
- オペレーティングシステム
 - OS 種別の比率と傾向
- ロケーション
 - 国ごとのリクエスト数、サイズと傾向



Amazon CloudFront モニタリング機能



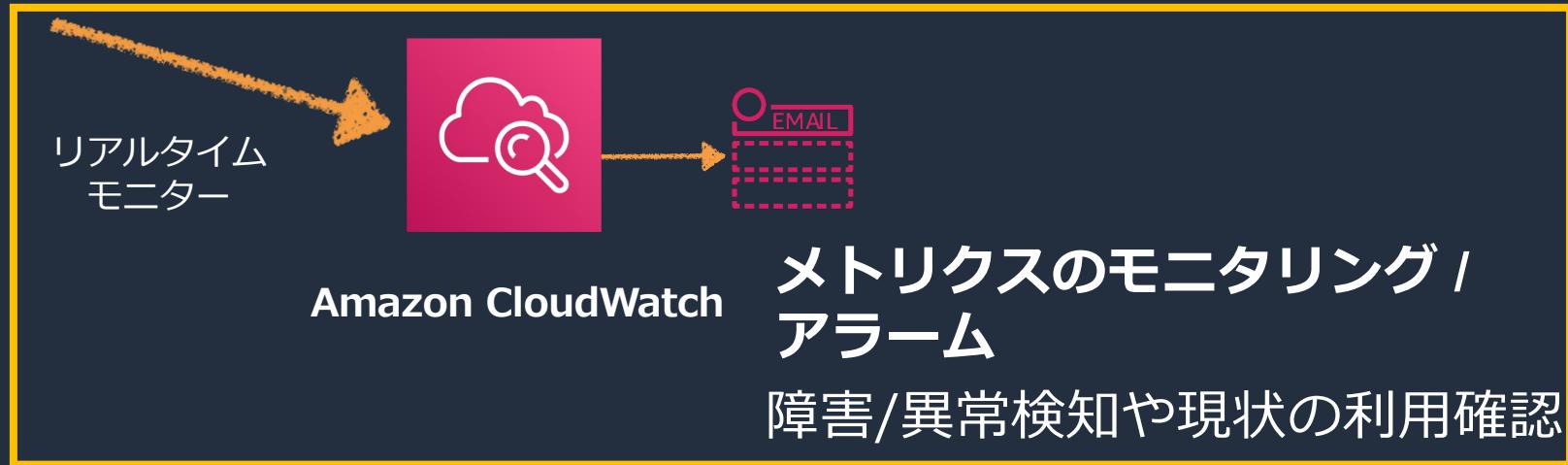
クライアント



レポーティング

キャッシュ統計
人気オブジェクト
トップリファラー
使用状況
ビューウー

アクセスや利用状況傾向の
確認及び分析



リアルタイム
モニター

Amazon CloudWatch

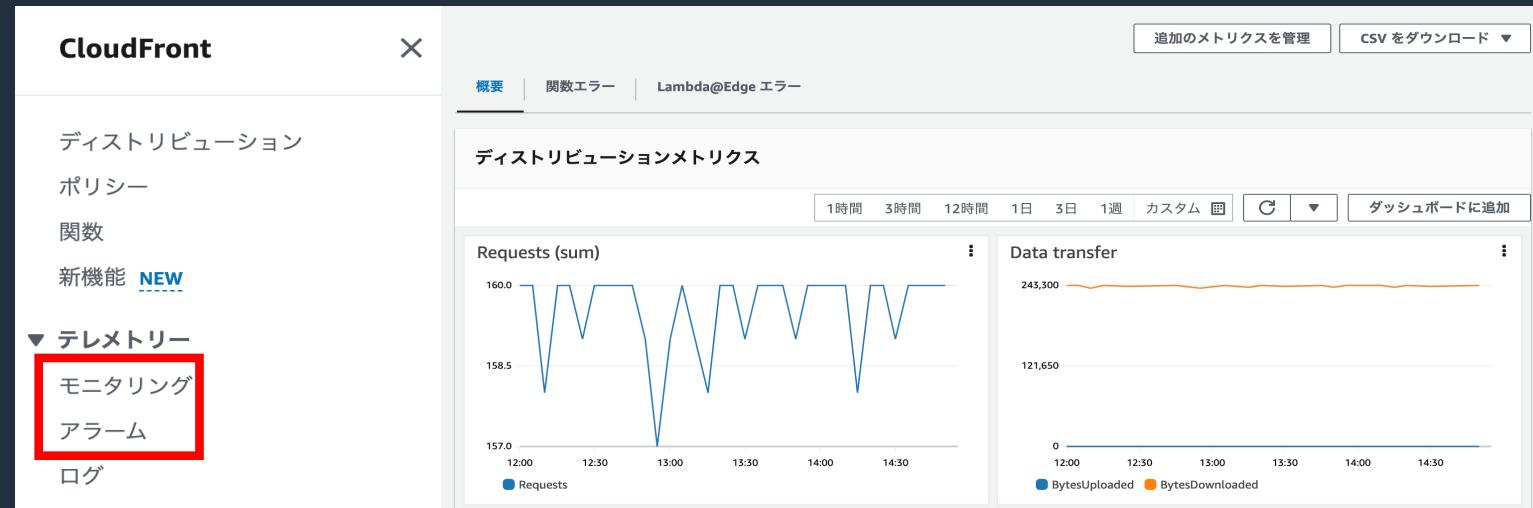
メトリクスのモニタリング /
アラーム

障害/異常検知や現状の利用確認

Amazon CloudWatchによる モニタリング / アラーム

ディストリビューション運用のメトリクスとエッジ関数 (CloudFront Functions と Lambda@Edge) のメトリクスを公開

- CloudFront の CloudWatch メトリクスは 米国東部 (バージニア北部) リージョンに出力
- メトリクスは CloudFront コンソールの一連のグラフに表示され、API または CLI を使用してアクセスすることが可能
- メトリクスは数分の遅延で利用状況を把握可能
- メトリクスデータは 15 ヶ月間保存
- CloudWatch のアラート機能を利用し、突発的なアクセスやエラーレートの上昇を検知して通知可能
- しきい値設定によるアラート連携も可能



CloudFront コンソール：モニタリング

概要：ディストリビューションメトリクス（デフォルトメトリクス）

CloudFront ディストリビューションについて、デフォルトメトリクスが追加料金なしで表示

- Requests (sum)
 - すべての HTTP メソッド、および HTTP リクエストと HTTPS リクエストの両方について CloudFront が受信したビューワーリクエストの総数
- Data transfer
 - BytesDownloaded: GET リクエスト、HEAD リクエスト、および OPTIONS リクエストに対してビューワーがダウンロードしたバイト総数
 - BytesUploaded: POST リクエストと PUT リクエストを使用して CloudFront でビューワーがオリジンにアップロードしたバイト総数
- Error rate (as a percentage of total requests)
 - 合計エラーの割合 (%)、レスポンスの HTTP ステータスコードが 4xx または 5xx であるすべてのビューワーリクエストの割合 (%)、5xxErrorByLambdaEdge の割合 (%)

CloudFront コンソール：モニタリング

概要：ディストリビューションメトリクス（追加メトリクス）

追加メトリクス（有償）を有効化することで、さらなる詳細なモニタリングが可能

- 4xx Error rate breakdown (as a percentage of total requests)
 - レスポンスの HTTP ステータスコードが 4xx (401、403、404) であるすべてのビューワリクエストの割合 (%)
- 5xx Error rate breakdown (as a percentage of total requests)
 - レスポンスの HTTP ステータスコードが 5xx (502、503、504) であるすべてのビューワリクエストの割合 (%)
- Origin latency
 - CloudFront キャッシュではなくオリジンから送信されたリクエストについて、CloudFront がリクエストを受信してからネットワーク（ビューワーではなく）にレスポンスを提供し始めるまでに費やした合計時間
- Cache hit rate
 - CloudFront がそのキャッシュからコンテンツを送信した対象のすべてのキャッシュ可能なリクエストの割合 (%)
HTTP POST/PUT リクエストおよびエラーは、キャッシュ可能なリクエストとは見なされない

CloudFront コンソール：モニタリング

ディストリビューション：関数エラー

対象ディストリビューションに関連付けられた CloudFront Functions のメトリクス

- 実行エラー
 - Global で発生した関数に処理されない例外またはコードにエラーがあり、CloudFront が関数からレスポンスを得られないときの実行エラー数
- 関数レスポンスが無効です
 - Global で関数は正常に実行されたが、CloudFront が関数から受け取るレスポンスのオブジェクト構造がイベント構造に従わない場合、またはレスポンスに無効なヘッダーや他の無効なフィールドが含まれている場合のエラー数
- 関連付けられた CloudFront Functions
 - 対象ディストリビューションに関連付けられた CloudFront Functions

CloudFront > モニタリング >

概要

関数エラー

Lambda@Edge エラー

CloudFront コンソール：モニタリング

ディストリビューション：Lambda@Edge エラー

対象ディストリビューションに関連付けられた Lambda@Edge のメトリクス

- 実行エラー
 - リージョン毎に発生した関数に処理されない例外またはコードにエラーがあり、CloudFront が関数からレスポンスを得られないときの実行エラー数
- 関数レスポンスが無効です
 - リージョン毎に関数は正常に実行されたが、CloudFront が関数から受け取るレスポンスのオブジェクト構造がイベント構造に従わない場合、またはレスポンスに無効なヘッダーや他の無効なフィールドが含まれている場合のエラー数
- スロットル
 - 関数の実行数が各リージョンのクオータに達した時にエラーが返される数
- 関連付けられた Lambda@Edge
 - 対象ディストリビューションに関連付けられた Lambda@Edge



CloudFront コンソール：モニタリング

CloudFront Functions ①

Amazon CloudWatch に運用メトリクスを送信し、関数をリアルタイムにモニタリング可能

- Invocations (sum)
 - 5 分ごとに関数が呼び出された回数
- Validation Errors
 - リージョン毎に関数は正常に実行されたが、CloudFront が関数から受け取るレスポンスのオブジェクト構造がイベント構造に従わない場合、またはレスポンスに無効なヘッダーや他の無効なフィールドが含まれている場合のエラー数
- Execution Errors
 - リージョン毎に発生した関数に処理されない例外またはコードにエラーがあり、CloudFront が関数からレスポンスを得られないときの実行エラー数

CloudFront > モニタリング

モニタリング

ディストリビューション

CloudFront Functions

Lambda@Edge

CloudFront コンソール：モニタリング

CloudFront Functions ②

- Throttles
 - 指定された期間に関数がスロットリングされた回数
 - スロットリングされる理由は、実行に許容される最大時間を継続的に超えている、コンパイルエラーが発生する、1 秒あたりのリクエスト数が多い場合
- Compute Utilization
 - 関数の実行にかかった時間（最大許容時間に対するパーセンテージ）たとえば、値 35 は、関数が最大許容時間（1ms 未満）の 35% で完了したことを意味する。このメトリクスは、0 から 100 までの数値
- 関連付けられているディストリビューション
 - 対象となる CloudFront Functions が関連付けられたディストリビューション

CloudFront > モニタリング

モニタリング

ディストリビューション

CloudFront Functions

Lambda@Edge

CloudFront コンソール：モニタリング

Lambda@Edge ①

Amazon CloudWatch に運用メトリクスを送信し、関数をリアルタイムにモニタリング可能

- Invocations (sum)
 - リージョン毎で 5 分ごとに関数が呼び出された回数、合計数 Global(sum) での表示も可能
- Errors
 - リージョン毎で発生した関数に処理されない例外またはコードにエラーがあり、CloudFront が関数からレスポンスを得られないときの実行エラー数、合計数 Global(sum) での表示も可能
- Throttles
 - 関数の実行数が各リージョンのクオータに達した時にエラーが返される数、合計数 Global(sum) での表示も可能

CloudFront > モニタリング

モニタリング

ディストリビューション

CloudFront Functions

Lambda@Edge

CloudFront コンソール：モニタリング

Lambda@Edge ②

- Success rate (%)
 - リージョン毎の関数の成功率、Global での表示も可能
- Duration (ms)
 - リージョン毎の関数の実行時間、Global での平均値 Global (avg) 表示も可能
- 関連付けられているディストリビューション
 - 対象となる Lambda@Edge が関連付けられたディストリビューション

The screenshot shows the CloudFront Lambda@Edge monitoring interface. At the top, there's a breadcrumb navigation: CloudFront > モニタリング. Below that is a header with tabs: ディストリビューション, CloudFront Functions, and Lambda@Edge, with Lambda@Edge being the active tab and highlighted by a yellow border. The main content area displays monitoring data for Lambda@Edge functions.

CloudFront コンソール： アラーム

アラーム

CloudFront コンソールで、CloudFront の特定のメトリクスに基づいて条件を設定し、Amazon Simple Notification Service (Amazon SNS) から通知を受け取るよう設定可能

The screenshot shows the CloudFront console's 'Alarms' creation interface. On the left sidebar, under the 'Monitoring' section, the 'Alarms' item is highlighted with a red box. The main area is titled 'CloudWatch アラームを作成する' (Create CloudWatch Alarm). It includes sections for 'Details' (Alarm Name, Distribution), 'Metrics' (Request count condition), and 'Notifications' (Recipient selection). The 'Metrics' section is expanded, showing a condition where the request count must be greater than 100 over a 1-minute period.

CloudFront > アラーム > 作成

CloudWatch アラームを作成する

詳細

アラーム名
アラームの名前。
アラーム名を入力

ディストリビューション
このアラームが対象とするディストリビューション。
参照

条件 情報

メトリクス
アラームの基準となるメトリクス。

リクエスト

「IF」リクエストの合計数
「If」リクエストの合計数 条件を設定します。
> 100
しきい値は数値でなければなりません。

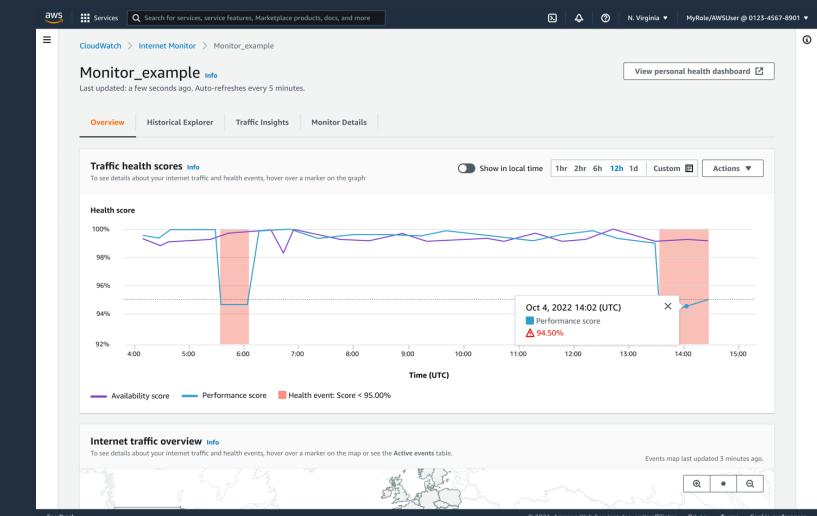
「FOR」連続期間
連続する期間に対して「for」条件を設定します。
1 分
期間は正の数でなければなりません。

通知 - オプション
通知する送信先を選択します。
送信先を選択

キャンセル アラームを作成

Amazon CloudWatch Internet Monitor を使った監視

- AWS でホストされたアプリケーションとアプリケーションエンドユーザーの間におけるインターネットの可用性とパフォーマンスのメトリクスを、簡単にモニタリングすることが可能
- 問題の影響を迅速に可視化して影響を受ける場所やプロバイダーを特定し、エンドユーザーのネットワークエクスペリエンスを向上
- トラフィックパターンとヘルスイベントをグローバルに把握し、さまざまな地理的詳細レベルでイベントに関する情報を掘り下げることが可能

This screenshot shows a modal dialog titled 'リソースを追加 (60)'. It contains a search bar and a table with six rows of CloudFront resources. The columns are: リソース ID, リソース名, AWS リージョン, and リソースタイプ. The resources listed are: E155VWISG0D9P, dc21+7zdu6vcloudfront, fr, CloudFront; E, d2fr, clo, CloudFront; E, d1uc, clo, CloudFront; and E, d2ugr9nq, clo, CloudFront.

ログイン機能

CloudFront から出力されるログの種類

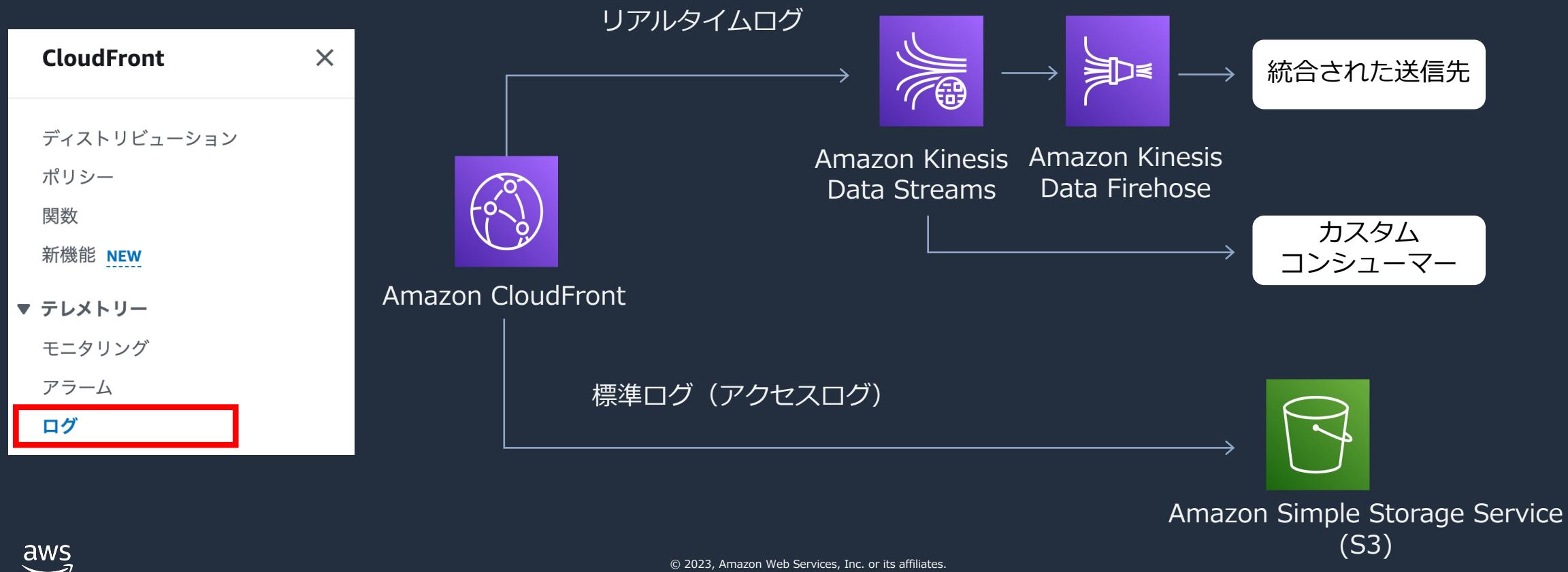
- リクエストのログ
- エッジ関数のログ
- サービスアクティビティのログ

CloudFront から出力されるログの種類

- リクエストのログ
 - 標準ログ（アクセスログ）
 - リアルタイムログ
- エッジ関数のログ
- サービスアクティビティのログ

Amazon CloudFront リクエストログ概要

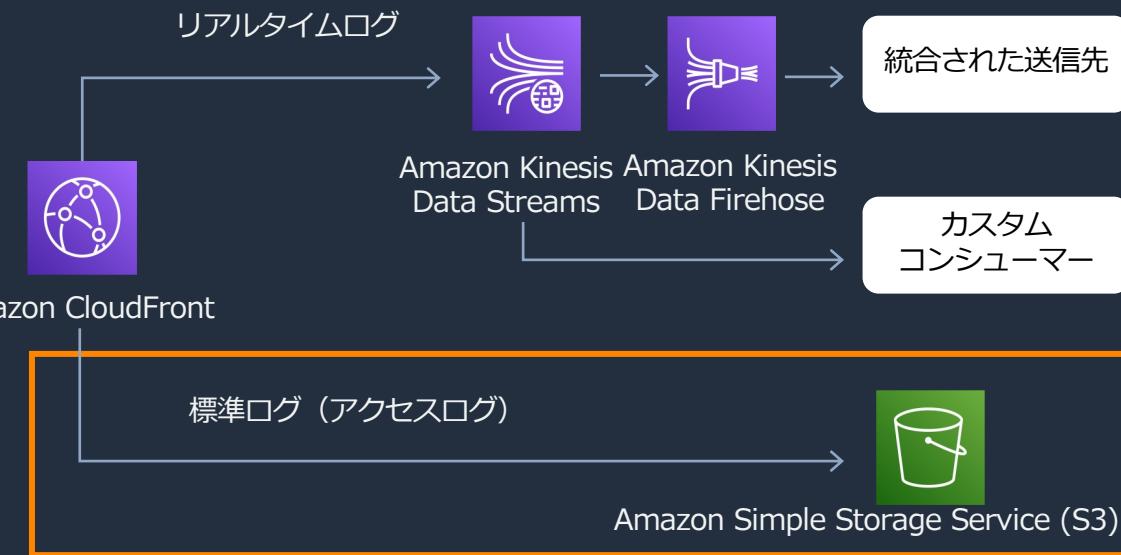
- リクエストを記録するために標準ログ（アクセスログ）とリアルタイムログを提供
- CloudFront コンソールから設定が可能



標準ログ (アクセスログ)

複雑なアクセスや利用傾向分析、データの可視化と詳細な障害分析

- リクエストに関する詳細なレコードを提供
- セキュリティやアクセスの監査などで利用可能
- 選択した Amazon S3 バケットに配信
- ログは 1 時間に最大で数回配信
- 標準ログの料金は発生しないが、ファイルの保存とアクセスについては Amazon S3 の料金が発生
- 期間中にオブジェクトに対してユーザーによるリクエストがなければ、その期間のログファイルは配信されない



標準ログ（アクセスログ）の設定方法

- CloudFront コンソールからログを選択し、スタンダードログのタブからロギング対象とするディストリビューションを選択し、標準ログの設定
- S3 バケットの指定とステータスを有効化

The screenshot shows the CloudFront console with the 'Logs' section selected. A distribution named 'E1E3YBI68RPMBE' is listed. The 'Standard' tab is active. A red box highlights the 'Edit' button in the 'Standard Log' row under the 'Log Status' column.

This dialog box allows editing standard log settings. It includes fields for 'S3 Bucket' and 'S3 Bucket Prefix'. Under 'Cookie Log Record', the 'Status' is set to 'Ineffective'. The 'Change to Effective' button is highlighted with a red box.

参考：標準ログ（アクセスログ）のログフィールド①

項目	説明
date	イベントが発生した日付 (UTC)
time	サーバーがリクエストへの対応を完了した時刻 (UTC)
x-edge-location	リクエストを処理したエッジロケーション
sc-bytes	サーバーがリクエストに応じてビューアーに送信したデータ (ヘッダーを含む) のバイトの合計数
c-ip	リクエスト元のビューアーの IP アドレス
cs-method	ビューアーから受信した HTTP リクエストメソッド
cs(Host)	ディストリビューションのドメイン名
cs-uri-stem	パスとオブジェクトを識別するリクエスト URL (/images/cat.jpg など)
sc-status	サーバーのレスポンスの HTTP ステータスコード もしくは 000 (サーバーがリクエストに応答する前に、ビューアーが接続を閉じた)
cs(Referer)	リクエスト内の Referer ヘッダーの値
cs(User-Agent)	リクエスト内の User-Agent ヘッダーの値
cs-uri-query	リクエスト URL のクエリ文字列の部分 (ある場合)
cs(Cookie)	名前と値のペアおよび関連属性を含む、リクエスト内の Cookie ヘッダー

Amazon CloudFront デベロッパーガイド：標準ログ（アクセスログ）の設定および使用
https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/AccessLogs.html



参考：標準ログ（アクセスログ）のログフィールド②

項目	説明
x-edge-result-type	サーバーが、最後のバイトを渡した後で、レスポンスを分類した方法 Hit : キャッシュヒット RefreshHit : キャッシュが Expire されていた Miss : キャッシュミス LimitExceeded : CloudFront のリミットオーバー CapacityExceeded : エッジのキャパシティ不足 Error : クライアントもしくはオリジンによるエラーなど
x-edge-request-id	リクエストを一意に識別する文字列
x-host-header	ビューウーが、このリクエストの Host ヘッダーに追加した値
cs-protocol	ビューウーリクエストのプロトコル (http、https、ws、wss のいずれか)
cs-bytes	ビューウーがリクエストに含めたデータ (ヘッダーを含む) のバイトの合計数
time-taken	サーバーが、ビューウーリクエストを受信してからレスポンスの最後のバイトを出力キューに書き込むまでの秒数。サーバーで 1,000 分の 1 秒単位まで測定されます (例: 0.082)
x-forwarded-for	ビューウーが HTTP プロキシなどを利用した場合の元ビューウー IP (IPv4 または IPv6)
ssl-protocol	リクエストとレスポンスを送信するためにビューウーとサーバーがネゴシエートした SSL/TLS プロトコル
ssl-cipher	リクエストとレスポンスを暗号化するためにビューウーとサーバーがネゴシエートした SSL/TLS 暗号

Amazon CloudFront デベロッパーガイド：標準ログ（アクセスログ）の設定および使用
https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/AccessLogs.html



参考：標準ログ（アクセスログ）のログフィールド③

項目	説明
x-edge-response-result-type	ビューウィーにレスポンスを返す直前にサーバーがレスポンスを分類した方法 ※分類は x-edge-result-type と同様
cs-protocol-version	ビューウィーがリクエストで指定した HTTP バージョン
fle-status	フィールドレベル暗号化設定時に、リクエストボディが正常に処理されたかどうかを示すコード
fle-encrypted-fields	サーバーが暗号化してオリジンに転送したフィールドレベル暗号化フィールドの数
c-port	閲覧者からのリクエストのポート番号
time-to-first-byte	サーバー上で測定される、要求を受信してから応答の最初のバイトを書き込むまでの秒数
x-edge-detailed-result-type	x-edge-result-type フィールドの値が Error である場合、このフィールドには特定のタイプのエラーが含まれる。オブジェクトが Origin Shield キャッシュからビューウィーに渡された場合、OriginShieldHit が含まれる
sc-content-type	レスポンスの HTTP Content-Type ヘッダーの値
sc-content-len	レスポンスの HTTP Content-Length ヘッダーの値
sc-range-start	レスポンスに HTTP Content-Range ヘッダーが含まれている場合、このフィールドには範囲の開始値が含まれる
sc-range-end	レスポンスに HTTP Content-Range ヘッダーが含まれている場合、このフィールドには範囲の終了値が含まれる

Amazon CloudFront デベロッパーガイド：標準ログ（アクセスログ）の設定および使用
https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/AccessLogs.html



Amazon S3 Select を使用した標準ログ（アクセスログ）のクエリ

Amazon S3 に出力された標準ログ（アクセスログ）に対して
Amazon S3 Select を使用してクエリを実行することが可能

The screenshot shows the Amazon S3 console interface. On the left, the navigation bar includes 'Amazon S3', 'Amazon S3 > パケット >', and the file name '-21-05.a75c3223.gz'. The main area displays the object's properties: 'オブジェクトの概要' (Owner: awslogsdelivery+s3_us-east-1). A dropdown menu titled 'オブジェクトアクション' is open, with the option 'S3 Select を使用したクエリ' highlighted with a red box. To the right, the 'SQL クエリ' section contains sample SQL code:

```
1 /* SQL クエリを記述するためのリファレンスポイントを作成するには、次の SQL クエリを実行して入力データの最初の 5 つのレコードを表示できます。
   SELECT * FROM $object s LIMIT 5 */
2 SELECT * FROM $object s LIMIT 5
```

The 'クエリ結果' section shows the output of the query: 'ステータス: 5 個のレコードを 282 ミリ秒で正常に返しました' and '返されたバイ字数: 1825 B'. The results are displayed in 'Raw' format.

Amazon Simple Storage Service (S3) ユーザーガイド : Amazon S3 Select を使用したデータのフィルタリングと取得
https://docs.aws.amazon.com/ja_jp/AmazonS3/latest/userguide/selecting-content-from-objects.html



Amazon Athena を使用した標準ログ（アクセスログ）のクエリ

Amazon S3 に出力された複数の標準ログ（アクセスログ）に対して
Amazon Athena を使用してクエリを実行することが可能



Amazon Athena ユーザーガイド : Amazon CloudFront ログのクエリ
https://docs.aws.amazon.com/ja_jp/athena/latest/ug/cloudfont-logs.html

参考：CloudFront ログ用のテーブル定義例

```
CREATE EXTERNAL TABLE IF NOT EXISTS default.cloudfront_logs (
    `date` DATE, time STRING, location STRING, bytes BIGINT, request_ip STRING, method STRING, host
    STRING, uri STRING, status INT, referrer STRING, user_agent STRING, query_string STRING, cookie STRING,
    result_type STRING, request_id STRING, host_header STRING, request_protocol STRING, request_bytes
    BIGINT, time_taken FLOAT, xforwarded_for STRING, ssl_protocol STRING, ssl_cipher STRING,
    response_result_type STRING, http_version STRING, file_status STRING, file_encrypted_fields INT, c_port INT,
    time_to_first_byte FLOAT, x_edge_detailed_result_type STRING, sc_content_type STRING, sc_content_len
    BIGINT, sc_range_start BIGINT, sc_range_end BIGINT
)

ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
LOCATION 's3://CloudFront_bucket_name/CloudFront/'
TBLPROPERTIES ( 'skip.header.line.count'=2' )
```

※ LOCATION をログを保存する Amazon S3 バケットに変更

Amazon Athena ユーザーガイド：Amazon CloudFront ログのクエリ
https://docs.aws.amazon.com/ja_jp/athena/latest/ug/cloudfront-logs.html



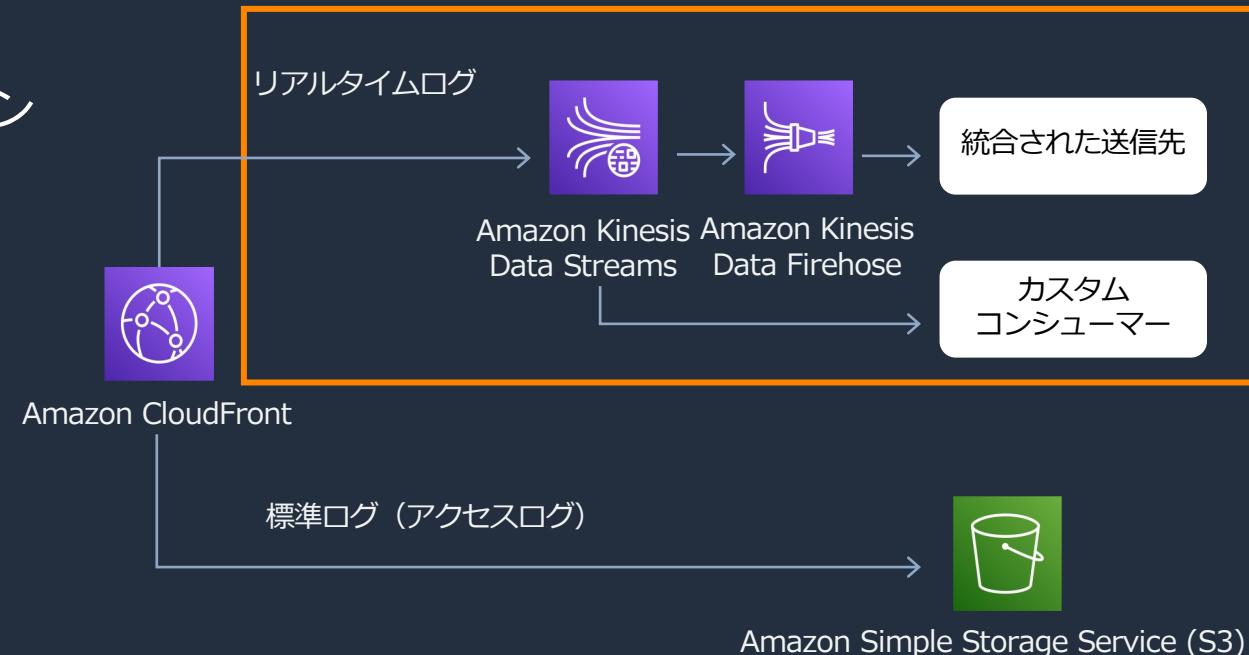
Amazon QuickSight を使用した可視化

ビジネスインテリジェンスサービスである Amazon QuickSight から Amazon Athena で定義したテーブル情報を可視化



リアルタイムログ

- ディストリビューションに対して行われたリクエストに関する情報をニアリアルタイムで提供
- ログレコードはリクエストを受信してから数秒以内に配信
- キャッシュビヘイビア毎に設定し、サンプリングレートと特定のフィールドを選択可能
- Amazon Kinesis Data Streams もしくは Amazon Kinesis Data Firehose 経由で Amazon S3, Amazon Redshift, Amazon OpenSearch Service および、サードパーティのログ処理サービスにログを配信可能
- Amazon Kinesis Data Streams で選択したデータストリームに配信。Kinesis Data Streams の使用料金に加えて、CloudFrontでのリアルタイムログの料金が発生



リアルタイムログの設定方法

- 事前に Amazon Kinesis のデータストリームを作成
- CloudFront コンソールからログを選択し、リアルタイム設定のタブを選択し、設定を作成を選択
- サンプリングレートなど設定情報を入力、エンドポイントで事前に作成したデータストリームを選択
- アタッチするディストリビューションとキャッシュビヘイビアを選択

The screenshot shows two overlapping windows from the AWS CloudFront console.

Left Window (Logs Overview):

- Shows the CloudFront navigation bar with "Logs" selected.
- Shows the "Logs" section with tabs for "Standard" and "Realtime Settings".
- Shows a table with one row: "Name" (名前) "Sampling Rate" (サンプリングレート) "Kinesis Data Stream" (Kinesis データストリーム).
- A red box highlights the "Create Setting" button (設定を作成).

Right Window (Create Log Setting):

- Shows the "Create Setting" dialog.
- Setting:**
 - "Name": A text input field.
 - "Sampling Rate": A dropdown menu showing "Sampling Rate" with options like "1%".
 - "Fields": A dropdown menu showing "Fields" with options like "asn", "c-country", "c-ip", and "c-ip-version".
 - "Endpoint": A dropdown menu showing "Endpoint" with options like "1つまたは複数の Kinesis Data Streams を選択して、リアルタイムログを送信します." (Select one or more Kinesis Data Streams to send real-time logs). A radio button is selected for "Create a new service role CloudFrontRealtimeLogConfigRole-name".
- Distribution:**
 - "Distribution": A dropdown menu showing "Distribution" with options like "ディストリビューション - オプション" (Distribution - Options).
 - "Behavior": A dropdown menu showing "Behavior" with options like "キャッシュビヘイビア" (Cache Behavior).
- Buttons:**
 - "Cancel" (キャンセル)
 - "Create Setting" (設定を作成)

参考：リアルタイムログのログフィールド①

項目	説明
timestamp	エッジサーバーがリクエストへの応答を終了した日時
c-ip	リクエスト元のビューウィーの IP アドレス
time-to-first-byte	サーバー上で測定される、要求を受信してから応答の最初のバイトを書き込むまでの秒数
sc-status	サーバーのレスポンスの HTTP ステータスコード
sc-bytes	サーバーがリクエストに応じてビューウィーに送信したデータ (ヘッダーを含む) のバイトの合計数
cs-method	ビューウィーから受信した HTTP リクエストメソッド
cs-protocol	ビューウィーのリクエストのプロトコル (http、https、ws、wss のいずれか)
cs-host	ビューウィーが、このリクエストの Host ヘッダーに追加した値
cs-uri-stem	クエリ文字列 (存在する場合) を含むが、ドメイン名を含まないリクエスト URL 全体
cs-bytes	ビューウィーがリクエストに含めたデータ (ヘッダーを含む) のバイトの合計数
x-edge-location	リクエストを処理したエッジロケーション
x-edge-request-id	リクエストを一意に識別する文字列
x-host-header	ディストリビューションのドメイン名
time-taken	サーバーが、ビューウィーのリクエストを受信してからレスポンスの最後のバイトを出力キューに書き込むまでの秒数
cs-protocol-version	ビューウィーがリクエストで指定した HTTP バージョン



Amazon CloudFront デベロッパーガイド : リアルタイムログ

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/real-time-logs.html

© 2023, Amazon Web Services, Inc. or its affiliates.

参考：リアルタイムログのログフィールド②

項目	説明
c-ip-version	リクエストの IP バージョン (IPv4 または IPv6)
cs-user-agent	リクエスト内の User-Agent ヘッダーの値
cs-referer	リクエスト内の Referer ヘッダーの値
cs-cookie	名前と値のペアおよび関連属性を含む、リクエスト内の Cookie ヘッダー
cs-uri-query	リクエスト URL のクエリ文字列の部分 (ある場合)
x-edge-response-result-type	ビューワーにレスポンスを返す直前にサーバーがレスポンスを分類した方法 (Hit, Missなど)
x-forwarded-for	ビューワーが HTTP プロキシなどを利用した場合の元ビューワー IP (IPv4 または IPv6)
ssl-protocol	リクエストとレスポンスを送信するためにビューワーとサーバーがネゴシエートした SSL/TLS プロトコル
ssl-cipher	リクエストとレスポンスを暗号化するためにビューワーとサーバーがネゴシエートした SSL/TLS 暗号
x-edge-result-type	サーバーが、最後のバイトを渡した後で、レスポンスを分類した方法 (Hit, Miss など)
fle-encrypted-fields	サーバーが暗号化してオリジンに転送したフィールドレベル暗号化フィールドの数
fle-status	フィールドレベル暗号化設定時に、リクエストボディが正常に処理されたかどうかを示すコード
sc-content-type	レスポンスの HTTP Content-Type ヘッダーの値
sc-content-len	レスポンスの HTTP Content-Length ヘッダーの値



Amazon CloudFront デベロッパーガイド : リアルタイムログ

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/real-time-logs.html

© 2023, Amazon Web Services, Inc. or its affiliates.

参考：リアルタイムログのログフィールド③

項目	説明
sc-range-start	レスポンスに HTTP Content-Range ヘッダーが含まれている場合、このフィールドには範囲の開始値が含まれる
sc-range-end	レスポンスに HTTP Content-Range ヘッダーが含まれている場合、このフィールドには範囲の終了値が含まれる
c-port	閲覧者からのリクエストのポート番号
x-edge-detailed-result-type	x-edge-result-type フィールドの値が Error である場合、このフィールドには特定のタイプのエラーが含まれる。オブジェクトが Origin Shield キャッシュからビューワーに渡された場合、OriginShieldHit が含まれる
c-country	ビューワーの IP アドレスによって決定される、ビューワーの地理的位置を表す国コード
cs-accept-encoding	ビューワーリクエスト内の Accept-Encoding ヘッダーの値
cs-accept	ビューワーリクエスト内の Accept ヘッダーの値
cache-behavior-path-pattern	ビューワーリクエストに一致したキャッシュ動作を識別するパスパターン
cs-headers	ビューワーリクエスト内の HTTP ヘッダー の名前と値 (800 バイトに切り捨て)
cs-header-names	ビューワーリクエスト内の HTTP ヘッダーの名前 (値ではない) (800 バイトに切り捨て)
cs-headers-count	ビューワーリクエスト内の HTTP ヘッダーの数
origin-fbl	CloudFront とオリジン間の先頭バイトのレイテンシーの秒数
origin-lbl	CloudFront とオリジン間の最終バイトのレイテンシーの秒数
asn	ビューワーの AS 番号 (ASN)



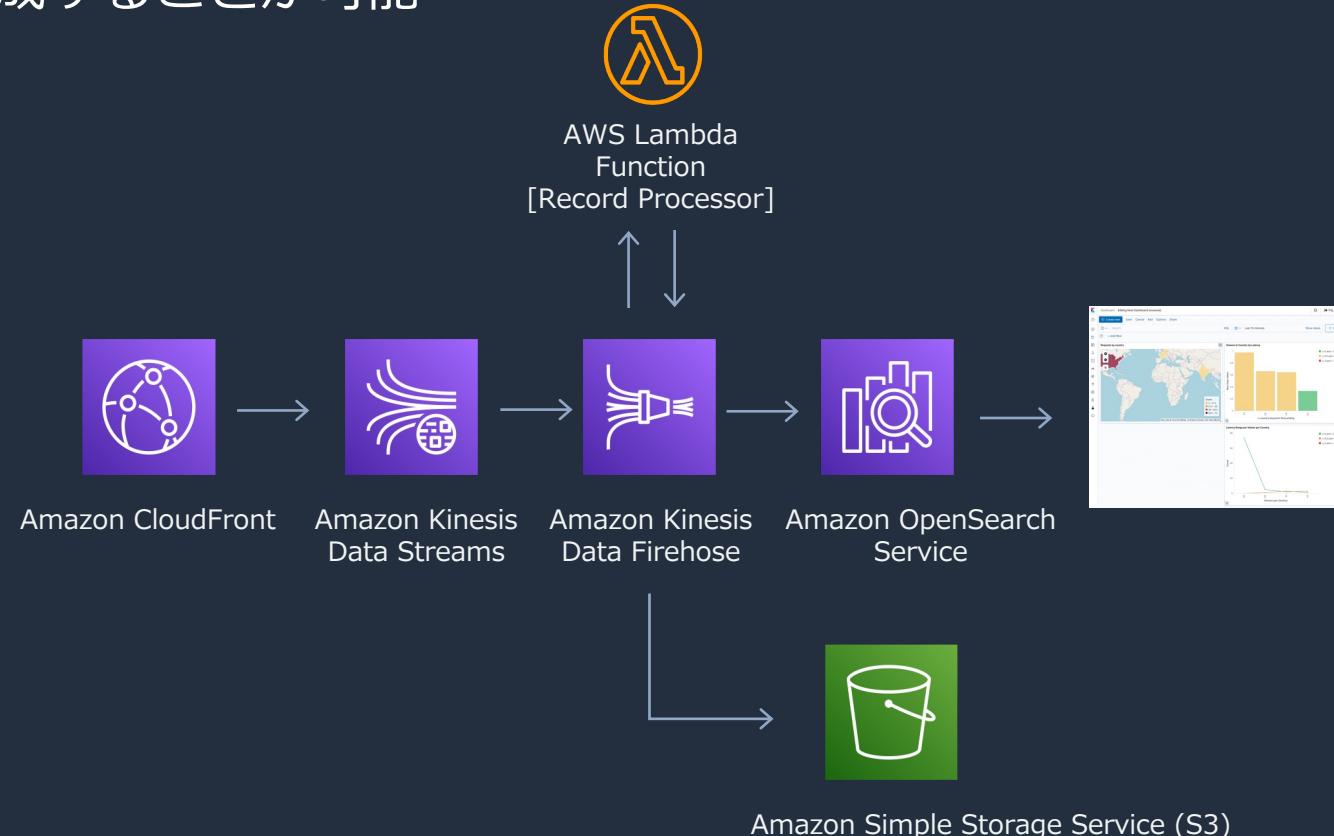
Amazon CloudFront デベロッパーガイド : リアルタイムログ

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/real-time-logs.html

© 2023, Amazon Web Services, Inc. or its affiliates.

リアルタイムログを使用したリアルタイムダッシュボード

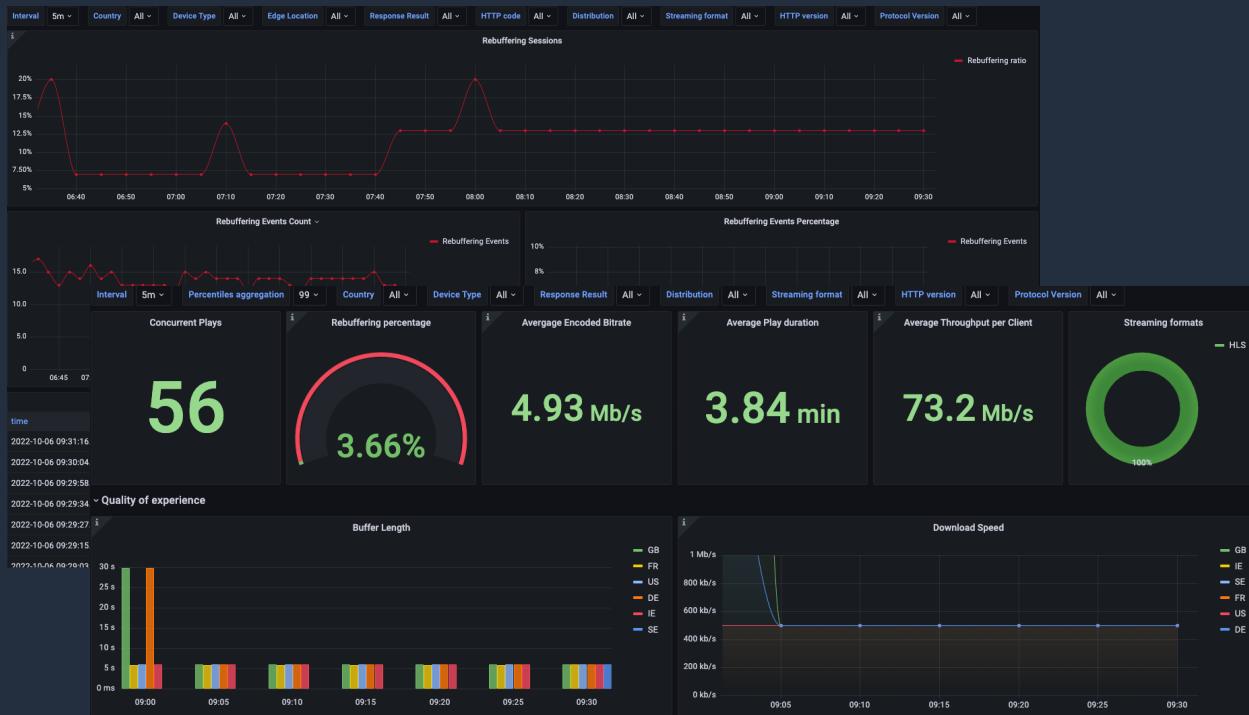
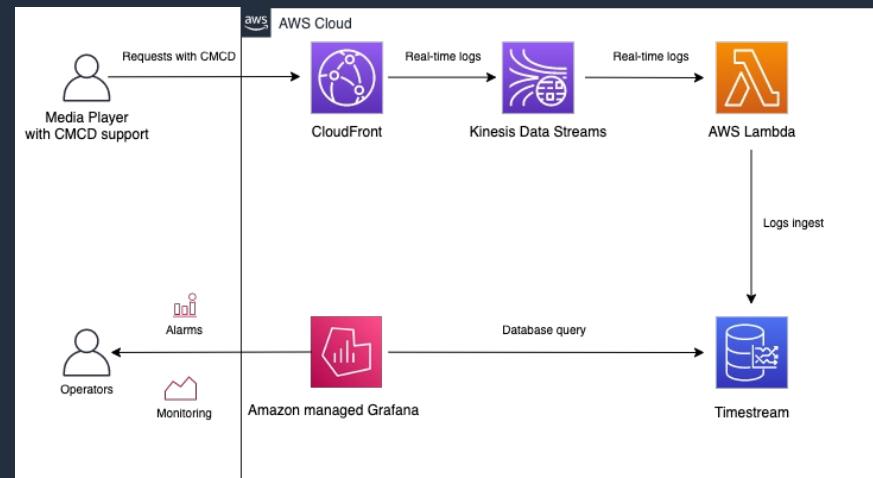
リアルタイムログと Amazon OpenSearch Service 組み合わせて
運用ダッシュボードを作成することが可能



Amazon Web Services ブログ : Amazon CloudFront ログを使用したリアルタイムダッシュボードの作成
<https://aws.amazon.com/jp/blogs/news/cloudfront-realtime-dashboard/>

CMCD と CloudFront による動画の可観測性向上

リアルタイムログと Amazon Managed Grafana と Common Media Client Data (CMCD) 仕様を組み合わせてユーザー体験を監視するダッシュボードを作成することが可能



Amazon Web Services ブログ : CMCD と CloudFront による動画の可観測性向上

<https://aws.amazon.com/jp/blogs/news/networking-and-content-delivery-improving-video-observability-with-cmcd-and-cloudfront/>

CloudFront から出力されるログの種類

- リクエストのログ
- エッジ関数のログ
 - **Lambda@Edge のログ**
 - **CloudFront Functions のログ**
- サービスアクティビティのログ

Lambda@Edge のログ

- 関数ログを CloudWatch Logs に自動的に送信し、関数が実行される AWS リージョンにログストリームを作成。ロググループ名は `/aws/lambda/us-east-1.function-name` の形式、function-name は作成時に関数に付けた名前になる
- 関数ログファイルを確認するためには、関数が実行された AWS リージョンで CloudWatch ログファイルを確認することが必要
- CloudFront コンソールで関数のメトリクスのグラフを表示し、同じページからリージョンを選択することで、該当するログファイルを表示することが可能

The screenshot shows the AWS CloudFront Metrics page. On the left, there's a sidebar with navigation links: ディストリビューション, ポリシー, 関数, 新機能 NEW, テレメトリー (expanded), モニタリング (highlighted with a red box), アラーム, and ログ.

The main content area has a breadcrumb navigation: CloudFront > モニタリング > arn:aws:lambda. It displays a chart titled "関数メトリクス" (Function Metrics) for "Invocations (sum)". The chart shows two data points: 1.00 and 0.80. Below the chart are time selection buttons: 1時間, 3時間, 12時間, 1日, 3日, 1週.

To the right of the chart, there's a "関数ログを表示" (View Function Log) button. A dropdown menu lists AWS regions and their corresponding AWS Lambda ARNs:

- 米国東部 (バージニア北部) us-east-1 [x]
- 米国東部 (オハイオ) us-east-2 [x]
- 米国西部 (北カリフォルニア) us-west-1 [x]
- 米国西部 (オレゴン) us-west-2 [x]
- アジアパシフィック (マンバイ) ap-south-1 [x]
- アジアパシフィック (東京) ap-northeast-1 [x]** (This item is highlighted with a blue border)
- アジアパシフィック (ソウル) ap-northeast-2 [x]
- アジアパシフィック (シンガポール) ap-southeast-1 [x]

CloudFront Functions のログ

- CloudFront Functions のコードに `console.log()` ステートメントが含まれている場合はログ行を CloudWatch Logs に自動的に送信、`console.log()` ステートメントがない場合は CloudWatch Logs には何も送信されない
- CloudFront Functions は、関数が実行されたエッジロケーションに関係なく、常に米国東部（バージニア北部）リージョン（us-east-1）にログストリームを作成
- ロググループ名は `/aws/cloudfront/function/FunctionName` の形式、`FunctionName` は関数を作成した際に指定した名前になる

ロググループ	アクション	アカウント ID	状況
/aws/cloudfront/function/	データ...	モニタリングア...	失効しない
/aws/cloudfront/function/	データ...	モニタリングア...	失効しない
/aws/cloudfront/function/	データ...	モニタリングア...	失効しない
/aws/cloudfront/function/	データ...	モニタリングア...	失効しない

ログメッセージの例：

- 各行はリクエストを一意に識別する ID で始まる
- メッセージは、ディストリビューション ID を含む START 行で始まり、END 行で終わる
- START 行と END 行の間には、関数の `console.log()` ステートメントによって生成されるログ行がある

```
U7b4hR_RaxMADupvKAvr8_m9gsGXvioUggLV5Oyq-vmAtH8HADpjhw== START DistributionID: E3E5D42GADAXZZ  
U7b4hR_RaxMADupvKAvr8_m9gsGXvioUggLV5Oyq-vmAtH8HADpjhw== Example function log output  
U7b4hR_RaxMADupvKAvr8_m9gsGXvioUggLV5Oyq-vmAtH8HADpjhw== END
```

CloudFront から出力されるログの種類

- リクエストのログ
- エッジ関数のログ
- サービスアクティビティのログ
 - CloudTrail による API リクエストのキャプチャ

CloudTrail による API リクエストのキャプチャ

- CloudFront は CloudTrail と統合され、リクエストのログファイルを定期的に指定の Amazon S3 バケットに保存
- リクエストが CloudFront コンソール、CloudFront API、AWS SDK、CloudFront CLI、または AWS CloudFormation などの別のサービスによって行われたかどうかにかかわらず、すべてのリクエストに関する情報をキャプチャ。CloudFront に対して発行されたリクエストの種類、リクエストの発行元 IP アドレス、発行者、発行日時などを判断可能
- CloudFront でアクティビティが発生すると、CloudTrail イベントに記録。CloudFront はグローバルサービスであるため、サービスのイベントはすべて米国東部 (バージニア北部) に記録

The screenshot shows the AWS CloudTrail console with the following details:

- Left Sidebar:** Includes links for CloudTrail Dashboard, Event History (which is selected), Insights, Lake, Query, Event Data Store, Integration (New), Trace, and Settings.
- Top Bar:** Shows the AWS logo, a search bar with placeholder "検索" (Search), and a button labeled "[オプション+S]" (Options+S).
- Page Header:** Displays "CloudTrail > イベント履歴" (Event History) and a breadcrumb trail.
- Main Content Area:**
 - Event History Section:** Shows "イベント履歴 (3) 情報" (Event History (3) Information). It includes a "Download Events" button, an "Athena Table Creation" button, and a search bar for "Event Source" set to "cloudfront.amazonaws.com".
 - Filter Section:** Includes a "ルックアップ属性" (Lookup Attribute) dropdown, a search bar for "Event Name" containing "ListDistributions", and time filters for "30m", "1h", "3h", "12h", "Clear", and "Custom".
 - Table Section:** A table listing three events:

イベント名	イベント時間	ユーザー名
ListDistributions	March 02, 2023, 12:57:36 (UTC+09:00)	
ListDistributions	March 02, 2023, 12:57:01 (UTC+09:00)	
ListFunctions	March 02, 2023, 12:56:16 (UTC+09:00)	resource-explorer-2

まとめ



まとめ

- Amazon CloudFront のキャッシュ統計情報やアクセス利用状況などのレポート情報は CloudFront コンソールから確認することができ、アクセスや利用状況傾向の確認及び分析に利用することができる
- Amazon CloudFront は CloudWatch に統合され、ディストリビューション運用のメトリクスやエッジ関数などのモニタリング情報を CloudFront コンソールから確認することができ、運用に利用することができる
- ログにはリクエスト、エッジ関数、サービスアクティビティのログの 3 種類があり、アクセスやセキュリティの監査など目的ごとに使い分けることができる

参考情報

- Amazon CloudFront デベロッパーガイド
レポート、メトリクス、ログ
https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/reports-and-monitoring.html
- Amazon CloudFront デベロッパーガイド
CloudFront とエッジ関数のログ記録
https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/logging.html

本資料に関するお問い合わせ・ご感想

技術的な内容に関しては、有料の AWS サポート窓口へ
お問い合わせください

<https://aws.amazon.com/jp/premiumsupport/>

料金面でのお問い合わせに関しては、カスタマーサポート窓口へ
お問い合わせください（マネジメントコンソールへのログインが必要です）

<https://console.aws.amazon.com/support/home#/case/create?issueType=customer-service>

具体的な案件に対する構成相談は、後述する個別相談会をご活用ください



ご感想は Twitter へ！ハッシュタグは以下をご利用ください
#awsblackbelt



その他コンテンツのご紹介

ウェビナーなど、AWS のイベントスケジュールをご参照いただけます

<https://aws.amazon.com/jp/events/>

ハンズオンコンテンツ

<https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-hands-on/>

AWS 個別相談会

AWS のソリューションアーキテクトと直接会話いただけます

<https://pages.awscloud.com/JAPAN-event-SP-Weekly-Sales-Consulting-Seminar-2021-reg-event.html>



Thank you!

AWS Black Belt Online Seminar

Amazon CloudFront 基礎編

鈴木 隆昭

Professional Services

2025/07



AWS Black Belt Online Seminar とは

- ・ 「サービス別」「ソリューション別」「業種別」などのテーマに分け、
アマゾン ウェブ サービス ジャパン合同会社が提供するオンラインセミナーシリーズです
- ・ AWS の技術担当者が、 AWS の各サービスやソリューションについてテーマごとに動画を公開します
- ・ 以下の URL より、過去のセミナー含めた資料などをダウンロードすることができます
 - > <https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-service-cut/>
 - > <https://www.youtube.com/playlist?list=PLzWGOASvSx6FIwIC2X1nObr1KcMCBBlqY>



ご感想は X (Twitter) へ！ハッシュタグは以下をご利用ください
#awsblackbelt

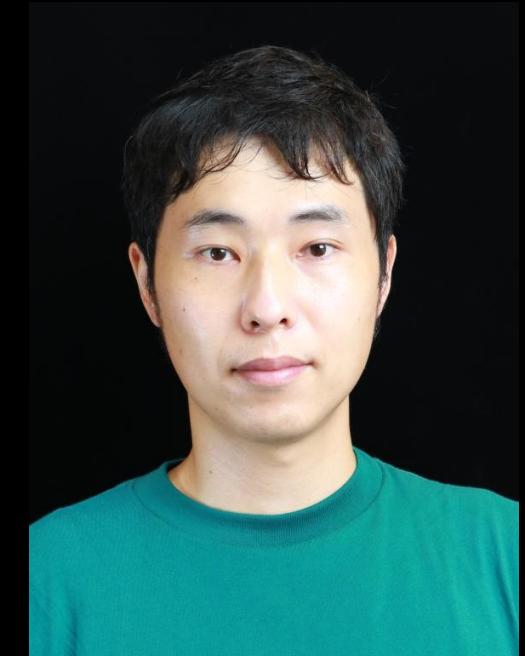
内容についての注意点

- ・ 本資料では資料作成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。AWS のサービスは常にアップデートを続けているため、最新の情報は AWS 公式ウェブサイト (<https://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には別途消費税をご請求させていただきます
- ・ 技術的な内容に関しては、有料の [AWS サポート窓口](#) へお問い合わせください
- ・ 料金面でのお問い合わせに関しては、[カスタマーサポート窓口](#) へお問い合わせください (マネジメントコンソールへのログインが必要です)

自己紹介

鈴木 隆昭

アマゾンウェブサービスジャパン
プロフェッショナルサービス
アプリケーション開発コンサルタント



AWS サービスを組み合わせたワークフローのプロトタイピングや、開発チームの内製化支援を行なっています。

好きな AWS サービス

Lambda@Edge, Amazon Aurora, AWS Certificate Manager

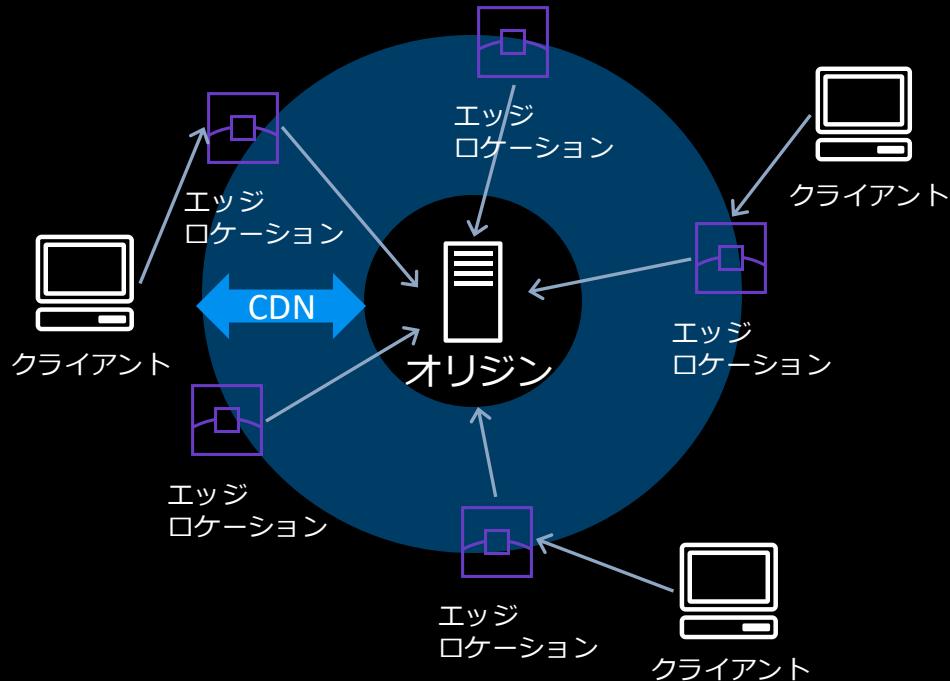
アジェンダ

1. Amazon CloudFront の概要
2. CloudFront の基本アーキテクチャ
3. CloudFront の設定の流れ
4. 主要機能についての補足
5. まとめ

Amazon CloudFront の概要

Content Delivery Network (CDN) とは

クライアントに地理的に近いエッジロケーションから、
速く・効率的にコンテンツ配信を行うネットワークを提供するサービス



グローバルに分散したエッジロケーション

- ✓ 広域な負荷分散
- ✓ 高可用なネットワークインフラストラクチャ
- ✓ 大容量の通信帯域、安定したネットワーク

様々な付帯機能

キャッシング：

- ✓ レスポンスタイムの削減
- ✓ オリジンの負荷軽減
- ✓ オリジンの保護

セキュリティサービスとの統合：

- ✓ DDoS 攻撃対策
- ✓ WAF

エッジコンピューティング：

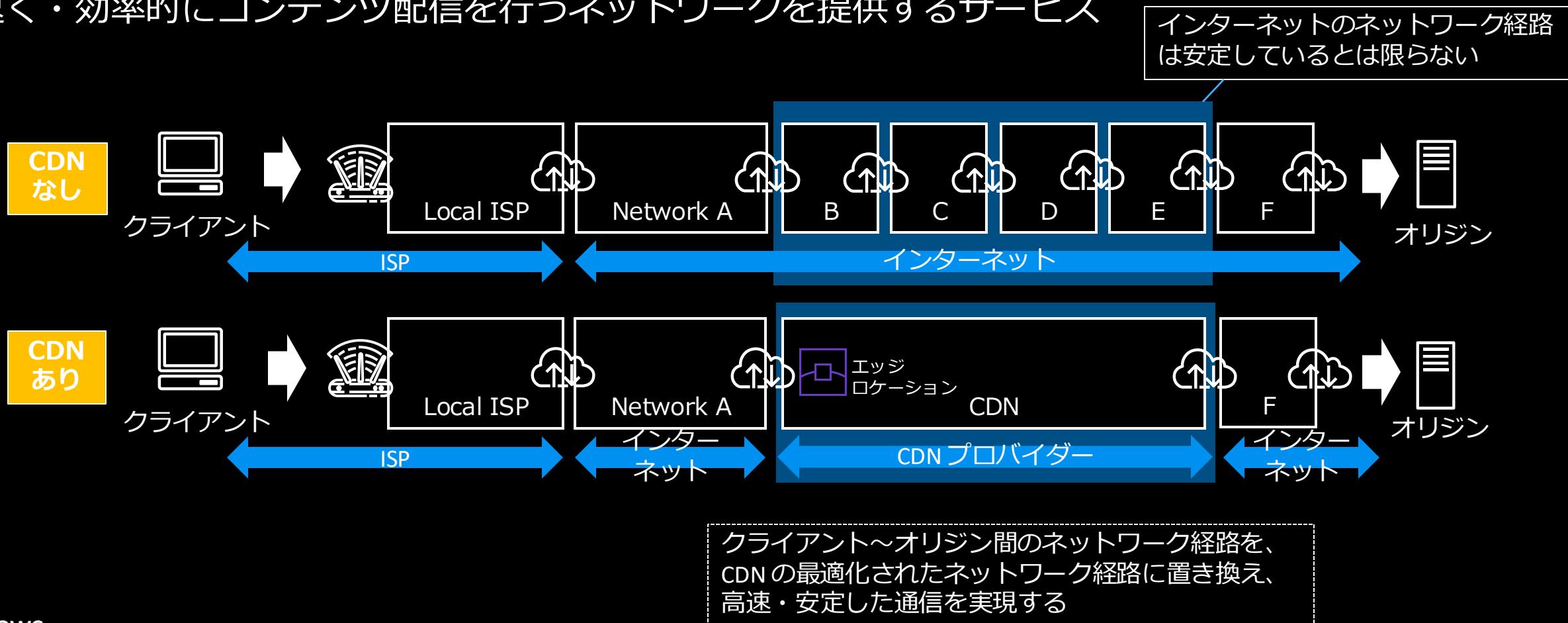
- ✓ コンテンツの書き換え
- ✓ 認可処理

+

等

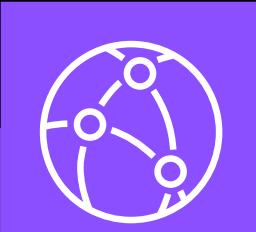
Content Delivery Network (CDN) とは

クライアントに地理的に近いエッジロケーションから、
速く・効率的にコンテンツ配信を行うネットワークを提供するサービス

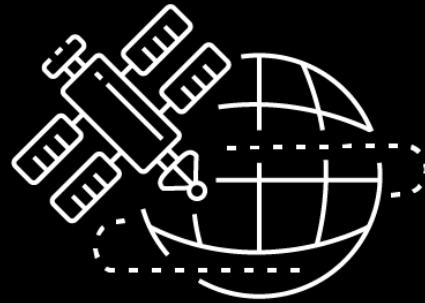


Amazon CloudFront

Fast, highly secure and programmable content delivery network (CDN)
高い安全性と性能を実現するプログラム可能なコンテンツデリバリー・ネットワーク



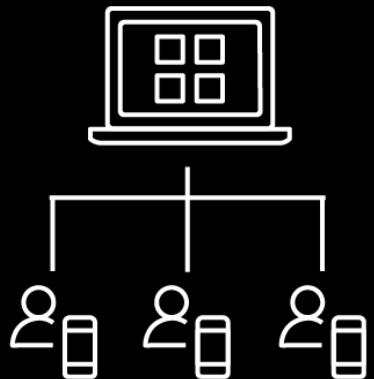
AWS のグローバル インフラストラクチャ



世界中から最適な経路での
高速・安定したアクセス

- ✓ ネットワークレイテンシーの低減
- ✓ ネットワーク帯域幅の確保
- ✓ 高可用なインフラストラクチャ

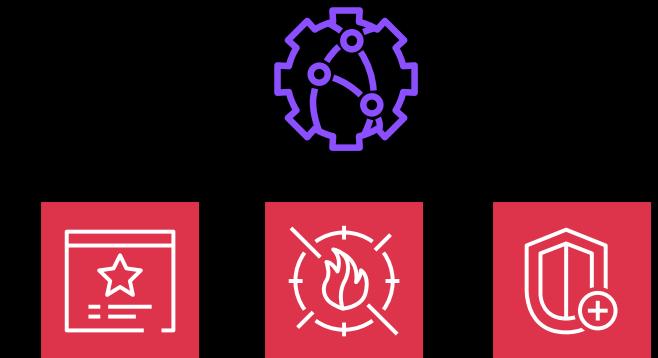
エッジロケーションを キャッシュサーバーとして活用



キャッシュされたコンテンツを
エッジロケーションから返却

- ✓ レスポンスタイムの低減
- ✓ サーバーのコンピュートリソースの節約
- ✓ ネットワークコストの節約

CloudFront の様々な機能や 他の AWS サービスとの連携

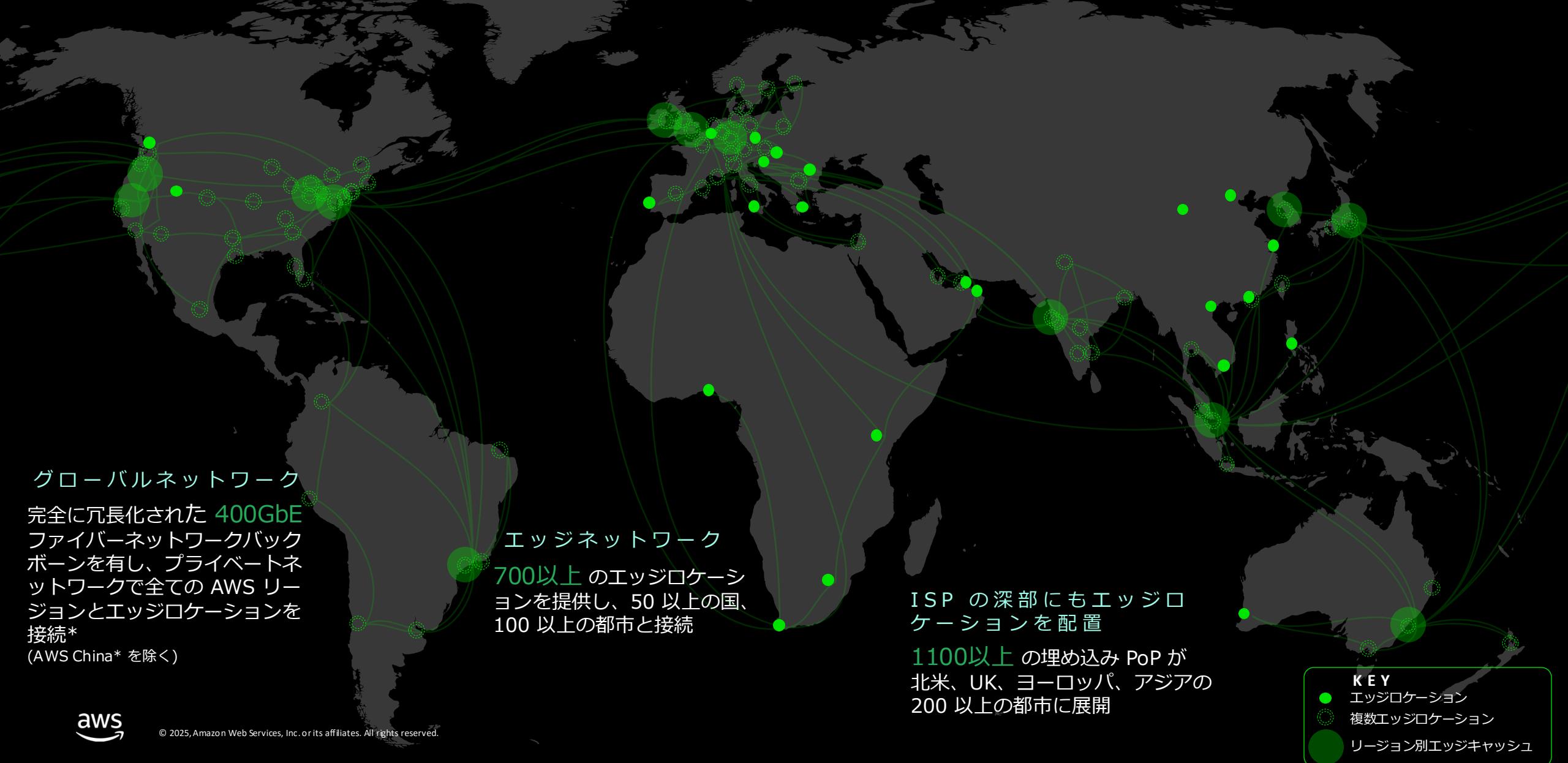


アプリケーションを変更をせず
動作をカスタマイズ

- ✓ エッジコンピューティング
- ✓ DDoS 攻撃対策、WAF
- ✓ SSL/TLS 終端等

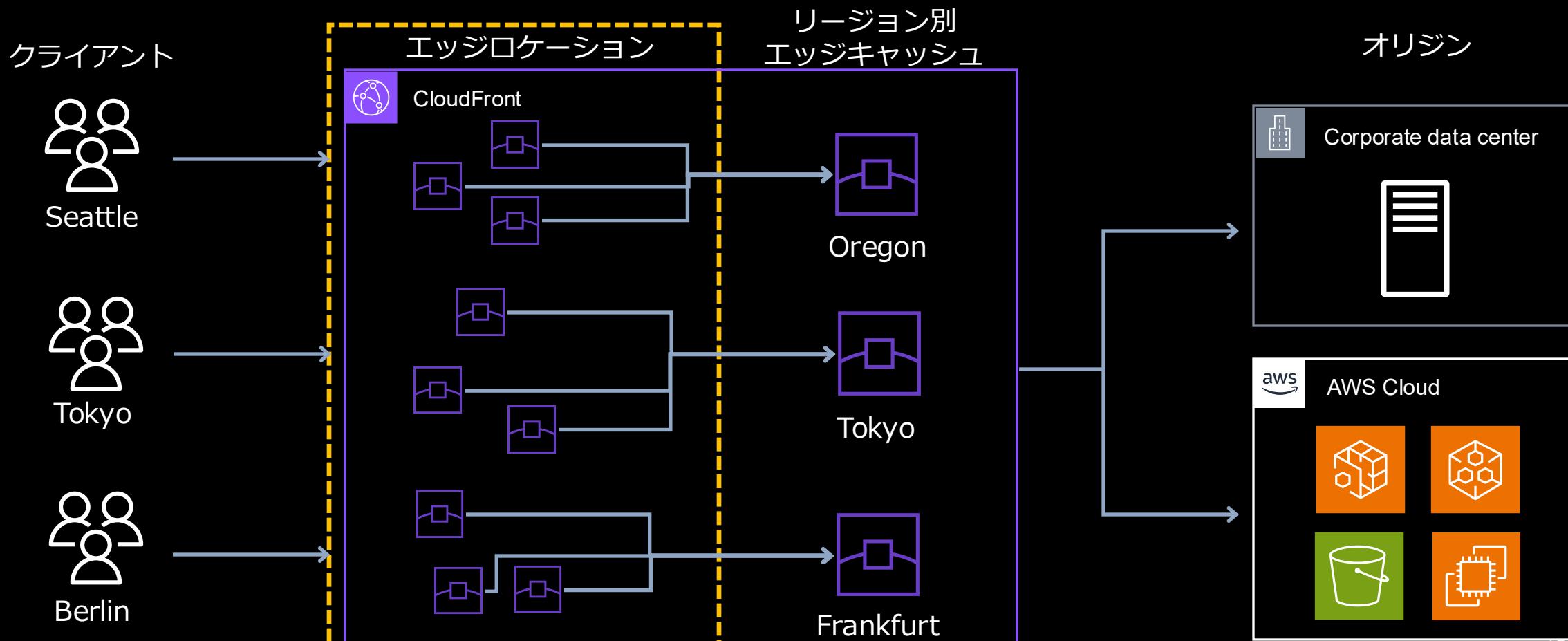
CloudFront の 基本アーキテクチャ

AWS グローバルインフラストラクチャ



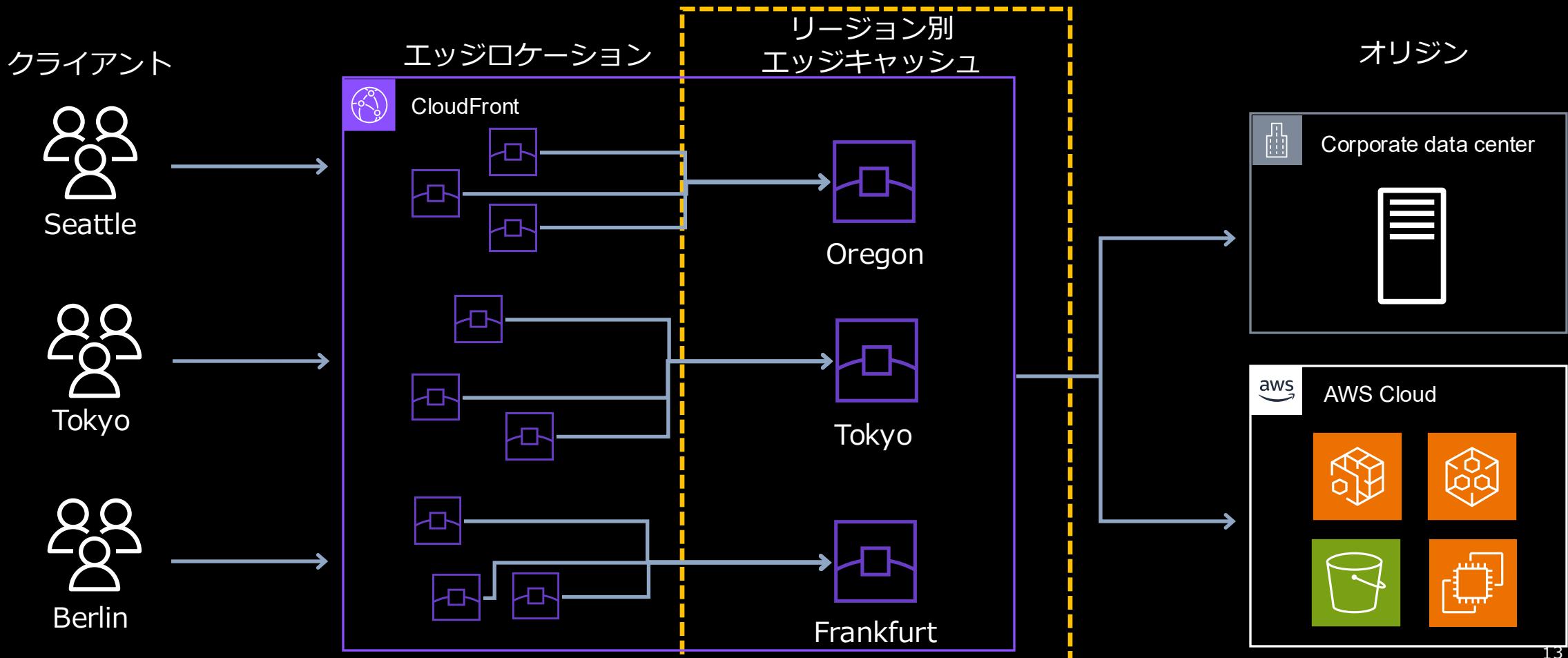
エッジキャッシング

- ✓ クライアントに最も近い場所からコンテンツを配信
- ✓ 各エッジキャッシングはキャッシュサーバーとして機能
- ✓ エッジキャッシング以降のリクエストの低レイテンシー化を実現



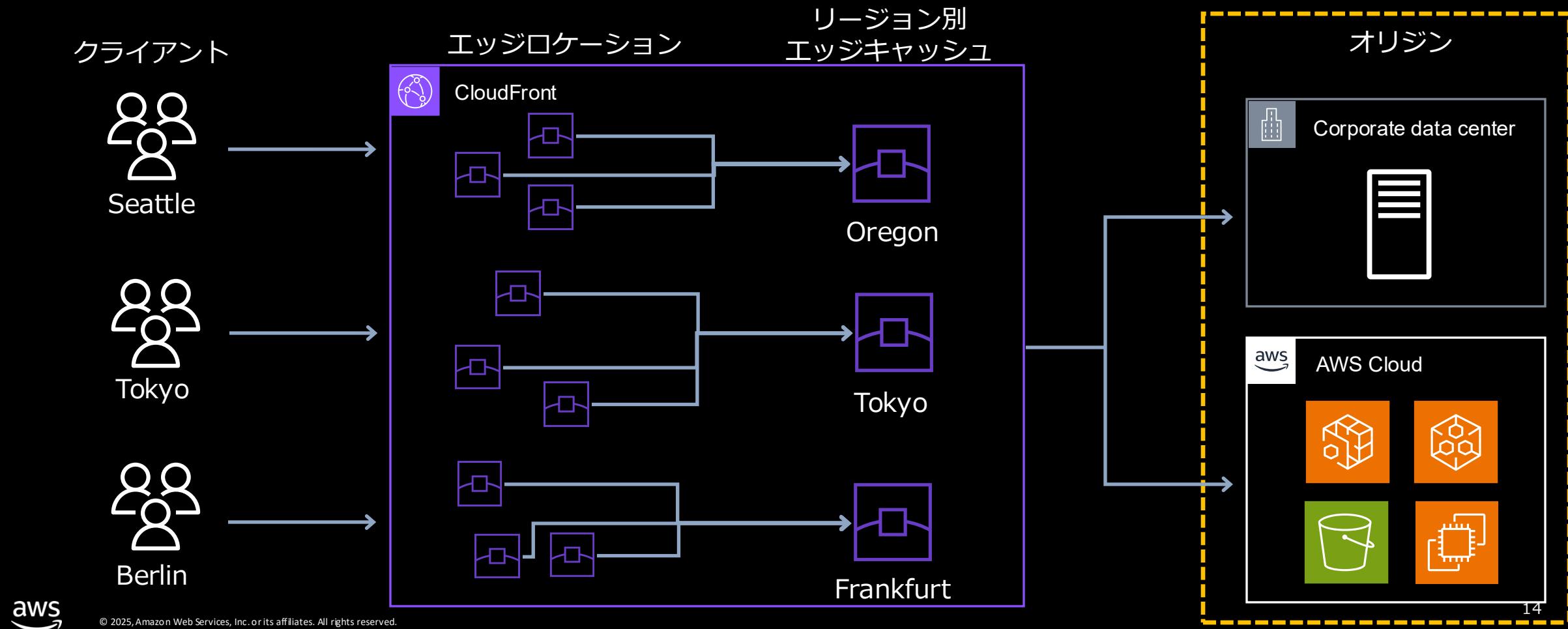
リージョン別エッジキャッシュ (REC)

- ✓ エッジロケーションとオリジン間に配置されるキャッシュ
- ✓ キャッシュ設定を無効化した場合、REC は使用されない
- ✓ 一部の HTTP メソッドはエッジロケーションから直接オリジンへ転送される



オリジン

- ✓ オリジナルのコンテンツを保持するソース
- ✓ CloudFront のキャッシュにヒットしなかった時にオリジンへのアクセスが発生
- ✓ オリジンは AWS リソースだけでなくオンプレミスサーバーも利用可能



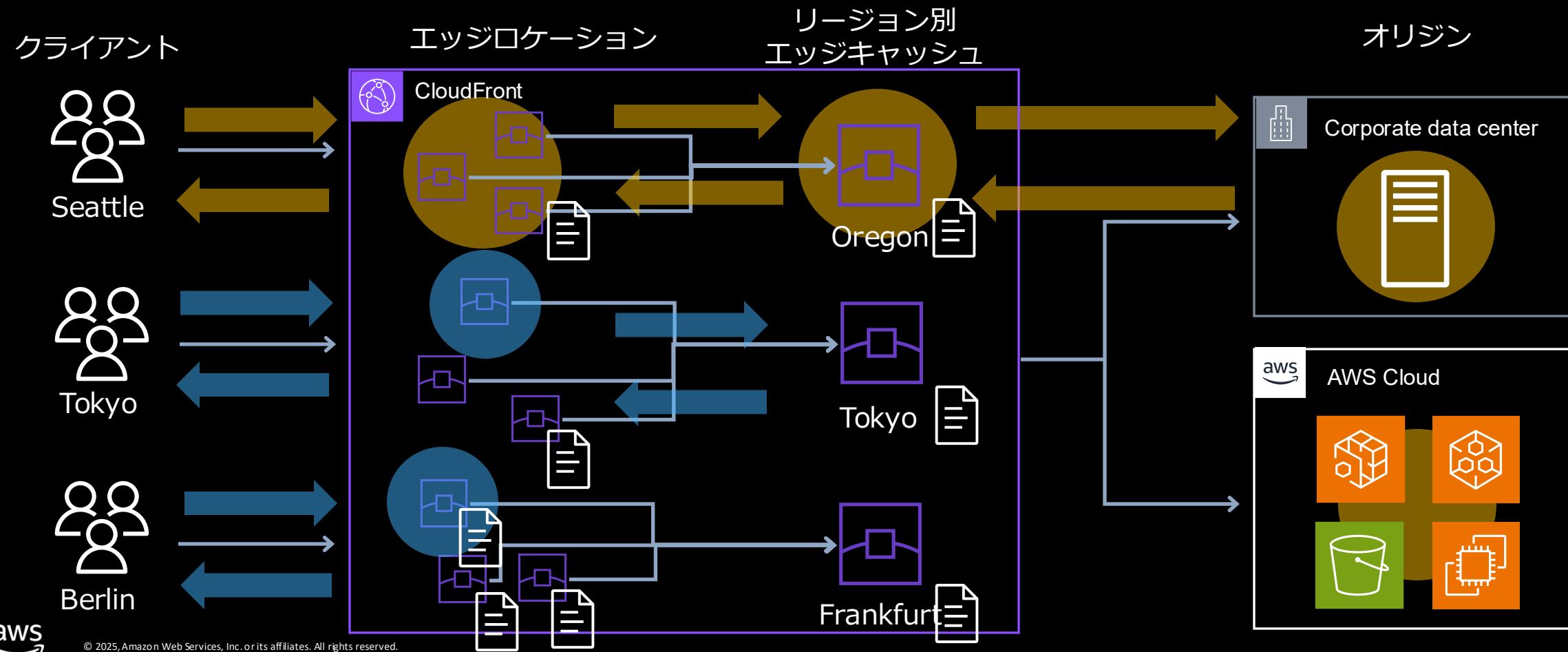
コンテンツ配信の仕組み

初回リクエスト時

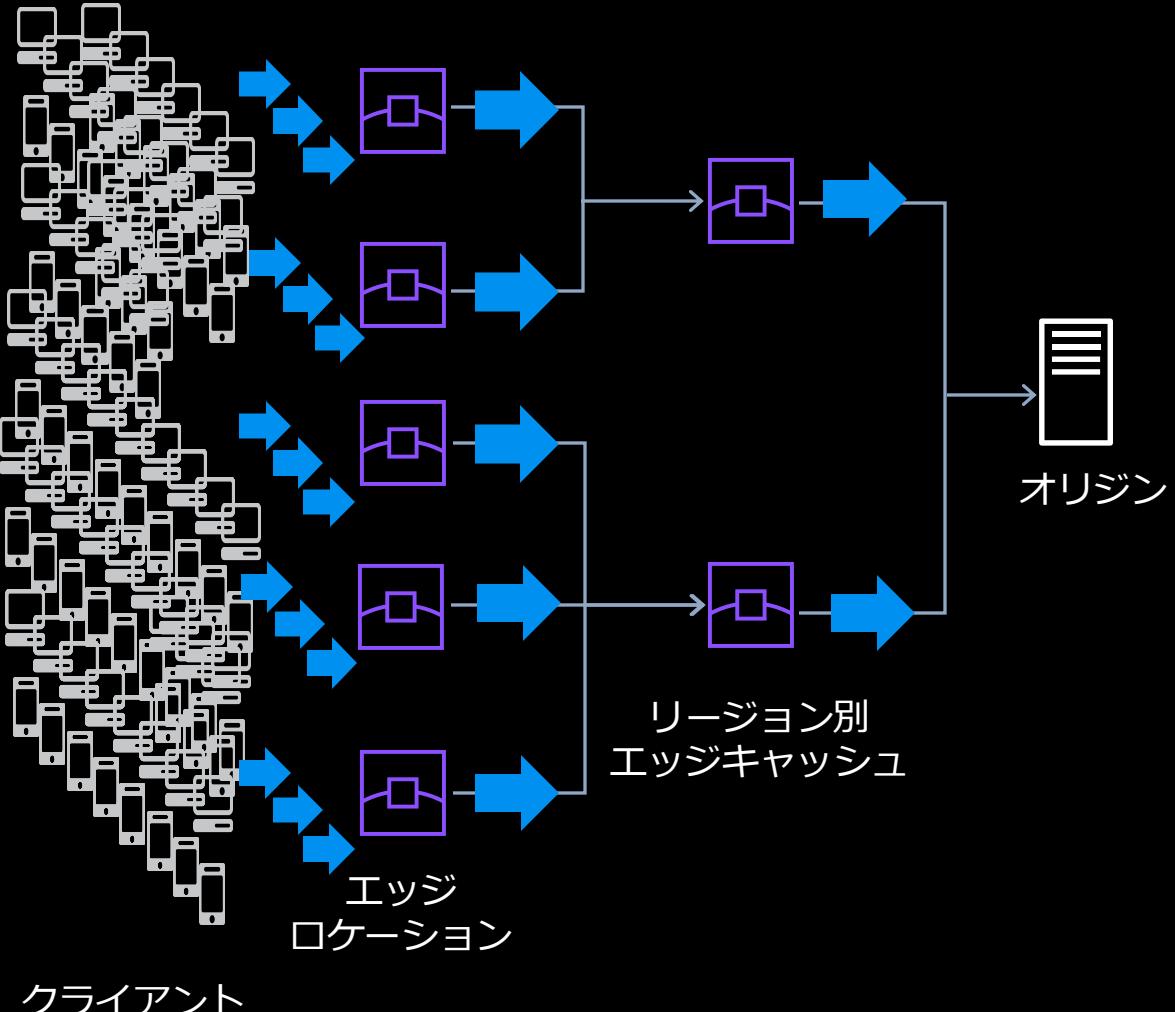
1. クライアントからのリクエストを最寄りのエッジロケーションが受信
2. エッジロケーションのキャッシュになければ、リージョン別エッジキャッシュを確認
3. リージョン別エッジキャッシュになければオリジンからコンテンツを取得
4. 取得したコンテンツをキャッシュしつつ、クライアントにレスポンス

2回目以降リクエスト時

1. クライアントからのリクエストを最寄りのエッジロケーションが受信
2. エッジロケーションのキャッシュからコンテンツを返却
(なければリージョン別エッジキャッシュからコンテンツを返却)



"フラッシュクラウド"からのオリジンの保護



- 同時に大量リクエストが発生（フラッシュクラウド/Flash Crowd）した場合、最初のリクエストのみをオリジンに送り、負荷低減を実現する仕組み
- オリジンが AWS 上にある場合は AWS グローバルネットワークを使用
- AWS 以外のオリジンに対しても同様の機能を提供

CloudFront の設定の流れ

CloudFront : 用語集

- ・ ビューウー: ユーザー / クライアント / Web ブラウザ
- ・ ディストリビューション: コンテンツ配信の設定単位
CloudFront ドメイン名、代替ドメイン名毎に作成
- ・ ビヘイビア: 振る舞いの設定
URL パスパターン毎に作成

キャッシュ設定：
キャッシュキー
TTL
コンテンツ圧縮

リクエスト/レスポンス制御：
ビューウー ~ CloudFront 間
CloudFront ~ オリジン間

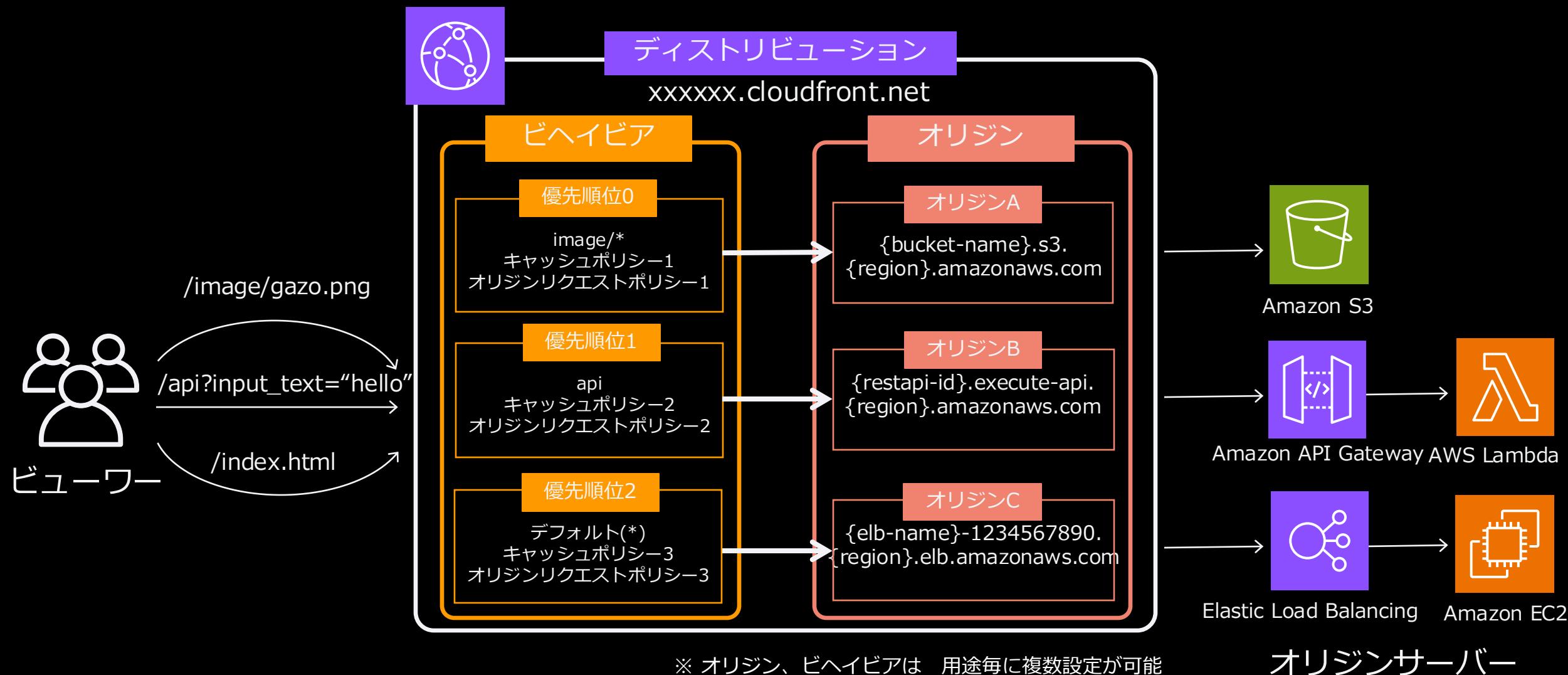
エッジコンピューティングの関連づけ：
CloudFront Functions
Lambda@Edge

- ・ オリジン: コンテンツ提供元の設定
コンテンツ提供元毎に作成

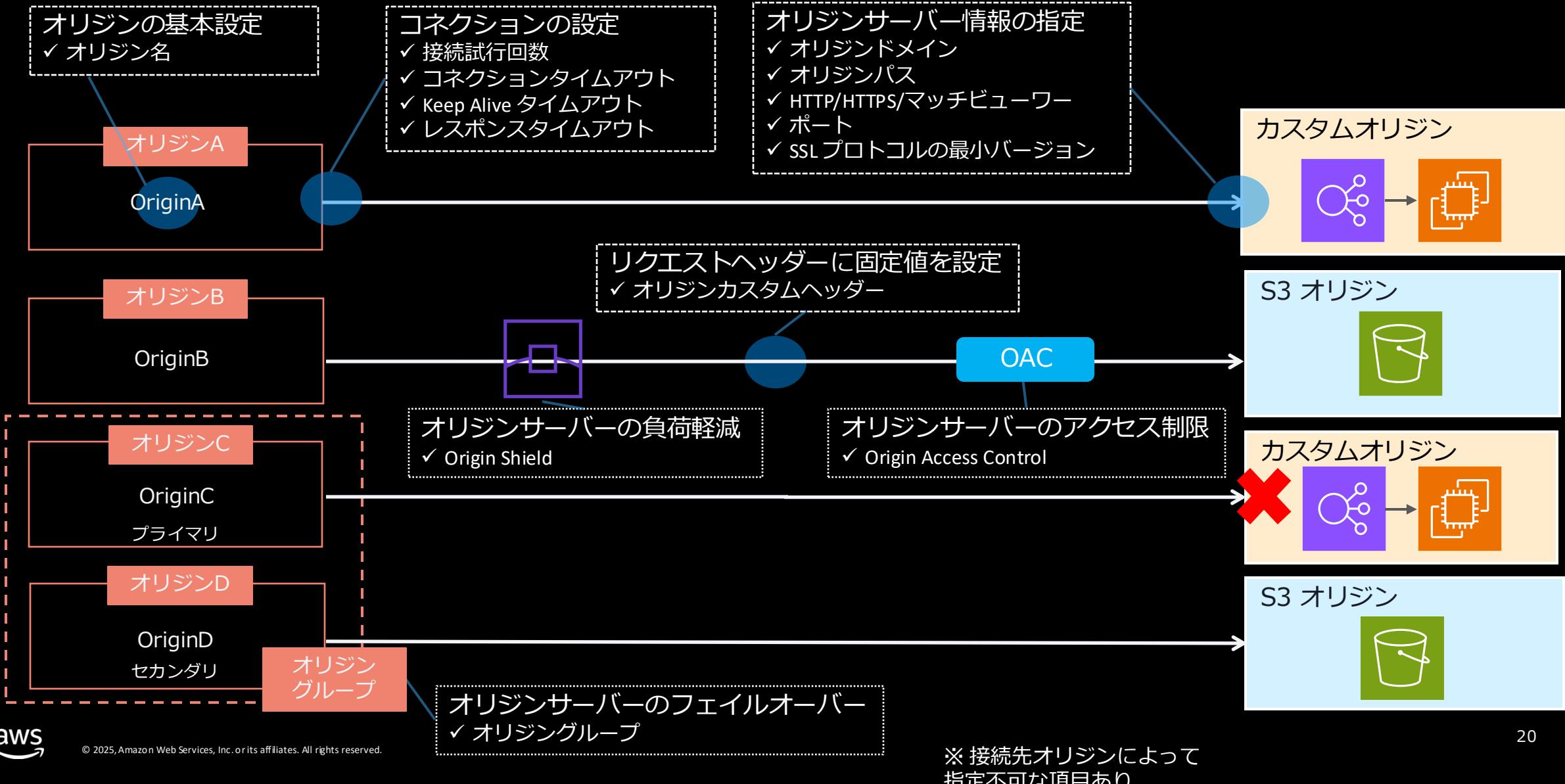
カスタムオリジン:
ALB、EC2 や Amazon API Gateway、
Lambda 関数 URL、オンプレミスの Web サーバー 等

S3 オリジン:
静的コンテンツを提供する S3 バケット

CloudFront の構成要素イメージ



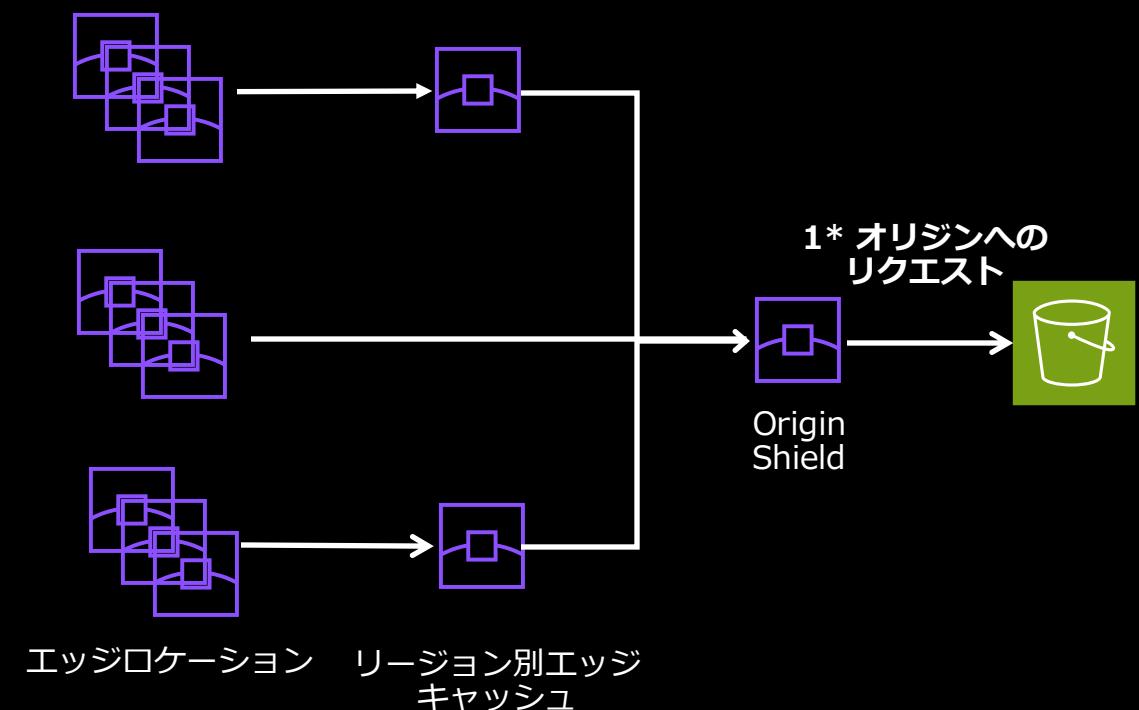
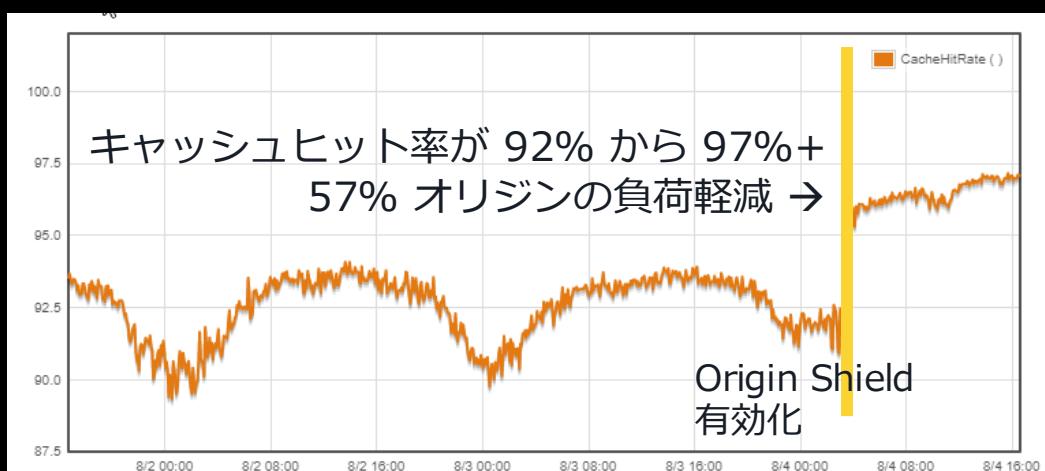
オリジンの設定概要



Origin Shield による負荷軽減

オリジンに最も近いリージョン別エッジキャッシュにアクセス拠点を限定してオリジンへの負荷を最小限に

- リージョン間の重複したリクエストを集約
- キャッシュヒット率の向上
- オリジンのコスト最適化
 - リクエスト数削減
 - データ転送量削減



Amazon CloudFront Origin Shield の使用

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/origin-shield.html

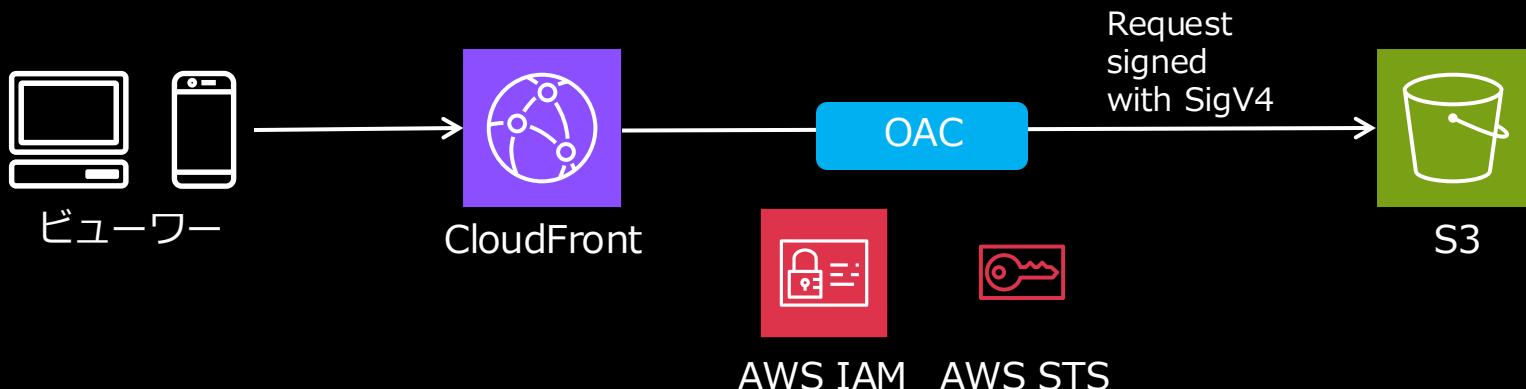


© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Origin Access Control (OAC)

IAM Service Principal による一貫したアクセス制御

- ・ 短期間のクレデンシャル、頻繁なクレデンシャルのローテーションおよびリソースベースのポリシーなど、強化されたセキュリティプラクティスを実装
- ・ GET, HEAD, OPTIONS に加え、PUT, POST, PATCH, DELETE メソッドをサポート
- ・ SSE-KMS で暗号化された S3 オブジェクトのダウンロード/アップロードをサポート
- ・ Origin Access Identity (OAI) は 2022 年 12 月までに開始された既存の AWS リージョンでのみサポートされるため、OAC の利用を推奨



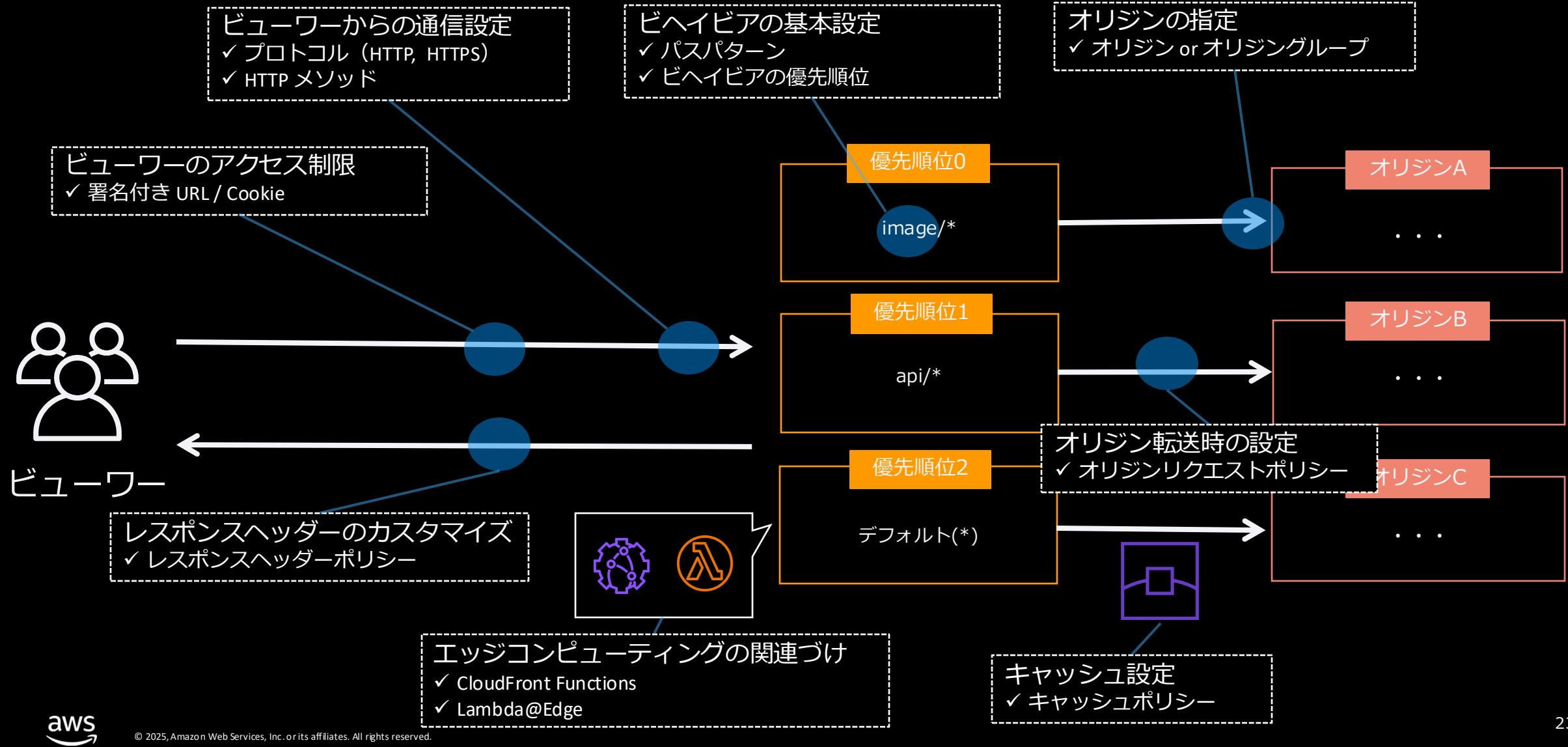
Amazon S3 オリジンへのアクセスの制限:

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/private-content-restricting-access-to-s3.html

Create control setting

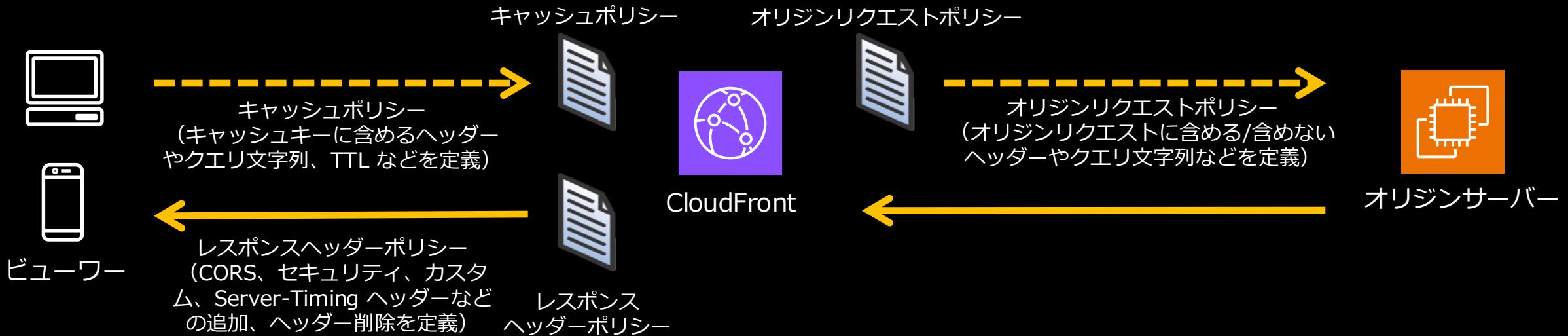
名前: origin-access-control-for-[region]-[bukcet]
説明 - オプション
署名動作
 リクエストに署名しない
 署名リクエスト (推奨)
 認証ヘッダーを上書きしない
オリジンタイプ
S3
キャンセル 作成

ビヘイビアの設定概要



ポリシー (キャッシュ / オリジンリクエスト / レスポンスヘッダー)

定義済みのマネージドポリシー、再利用可能なカスタムポリシーを定義



ポリシーの使用:

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/working-with-policies.html

Screenshot of the AWS CloudFront console showing the Policies section under Distribution settings. The left sidebar shows navigation options like Distributions, Policies, Metrics, Reports & Analytics, and Amplify. The main panel displays a list of managed policies:

名前	説明
Amplify	Policy for Amplify Origin
CachingDisabled	Policy with caching disabled
CachingOptimized	Default policy when CF compression is enabled
CachingOptimizedForUncompressedObjects	Default policy when compression is disabled
Elemental-MediaPackage	Policy for Elemental MediaPackage Origin

- オリジンに転送するリクエストとキャッシュキーを分離して取り扱うことにより、柔軟なキャッシュ設定が可能
- レスポンスヘッダーのカスタマイズが可能

署名付き URL / Cookie

クライアントのコンテンツへのアクセス制限



- IAM アカウントで署名付き URL / Cookie のキー設定を行う
- 単一コンテンツアクセスの場合は署名付き URL、HLS 動画配信などの複数コンテンツアクセスの場合は、署名付き Cookie の利用を推奨

署名付き URL と署名付き Cookie を使用したプライベートコンテンツの提供

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/PrivateContent.html

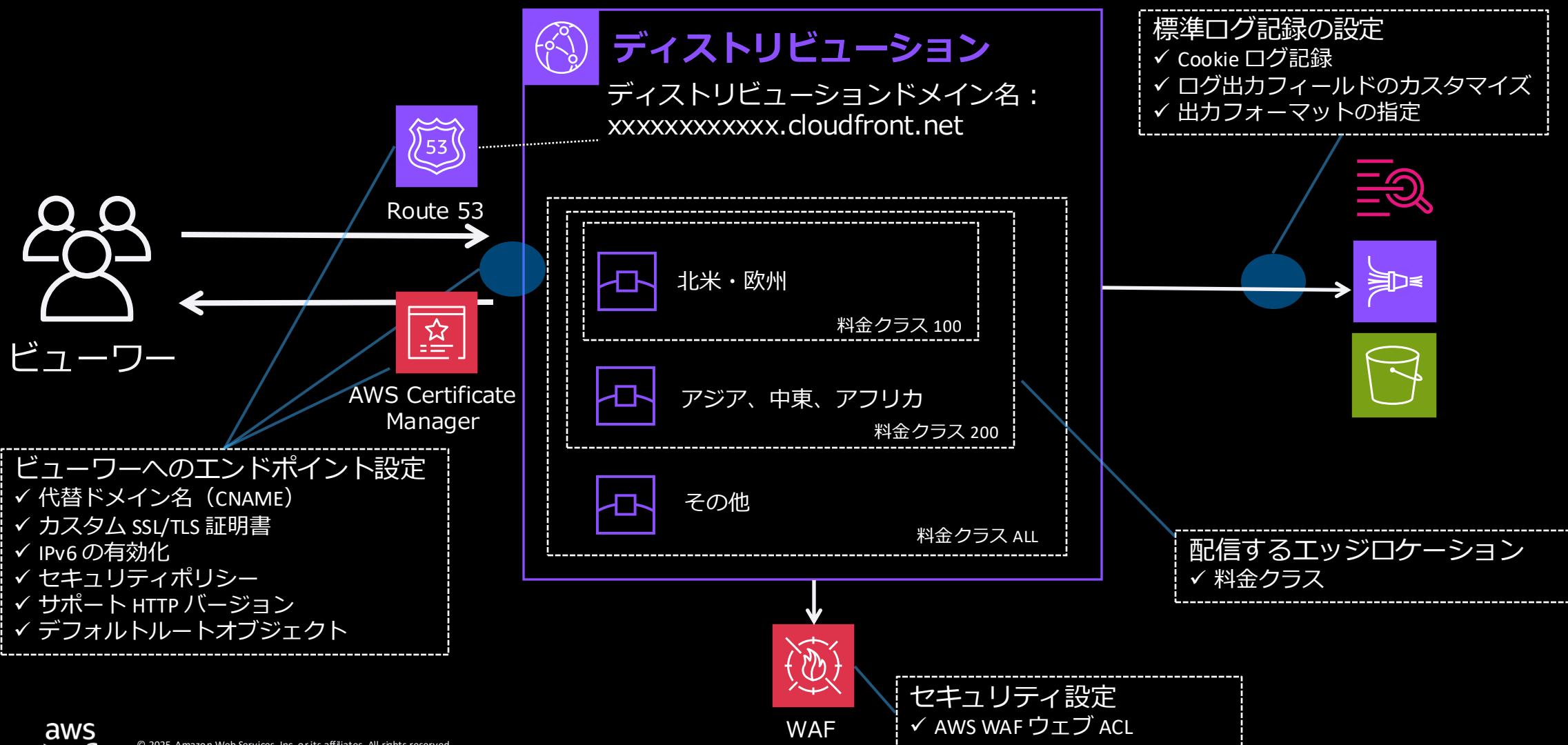
エッジコンピューティングの関連付け

ビューウーに近い場所で任意の処理を実行



- ・ ビューウーに近いエッジロケーション/リージョン別エッジキャッシュでコードを実行するサービス
- ・ HTTP(s) リクエストとレスポンスをトリガーする
- ・ サーバーレスでインフラストラクチャの管理は不要
- ・ トリガーに応じて自動でスケール
- ・ 使った分だけの従量課金
- ・ コンピューティングリソースをグローバルに利用可能

ディストリビューションの設定概要



代替ドメイン名、カスタム SSL/TLS 証明書の設定

独自のドメインでセキュアにコンテンツを配信

- クライアントからの SSL/TLS 接続はエッジロケーションで終端
- 代替ドメイン名 (CNAME) の追加には、ドメイン名を含む **有効なカスタム SSL/TLS 証明書を設定する**
 - > CloudFront は AWS Certificate Manager (ACM) と統合されており、**無償のドメイン認証 (DV)** タイプ証明書を数分で発行、自動更新も可能
 - > ワイルドカード指定 (例: *.example.com) や、Route53 エイリアスレコードと組み合わせた Zone Apex (例: example.com) をサポート



サポート HTTP バージョン、セキュリティポリシー

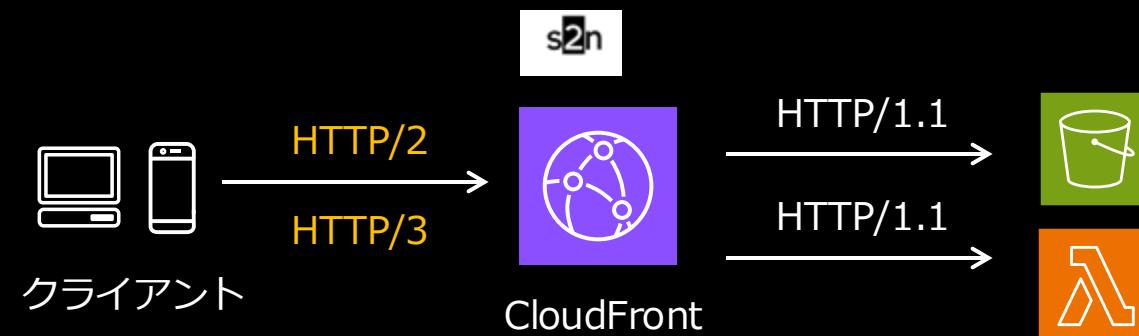
CloudFrontへの接続要件を指定

サポート HTTP バージョン

- TLS 1.3 クライアント接続をデフォルトサポート
- HTTP/1.0, HTTP/1.1, WebSocket プロトコルをデフォルトサポート
- HTTP/2, HTTP/3 の追加サポートが可能

セキュリティポリシー

- 最低限の SSL/TLS プロトコルと暗号の組み合わせ
 - ✓ TLSv1.2_2021
 - ✓ TLSv1.2_2019
 - ✓ TLSv1.2_2018
 - ✓ TLSv1.1_2016
 - ✓ TLSv1_2016
 - ✓ TLSv1
 - ✓ SSLv3 (非推奨)

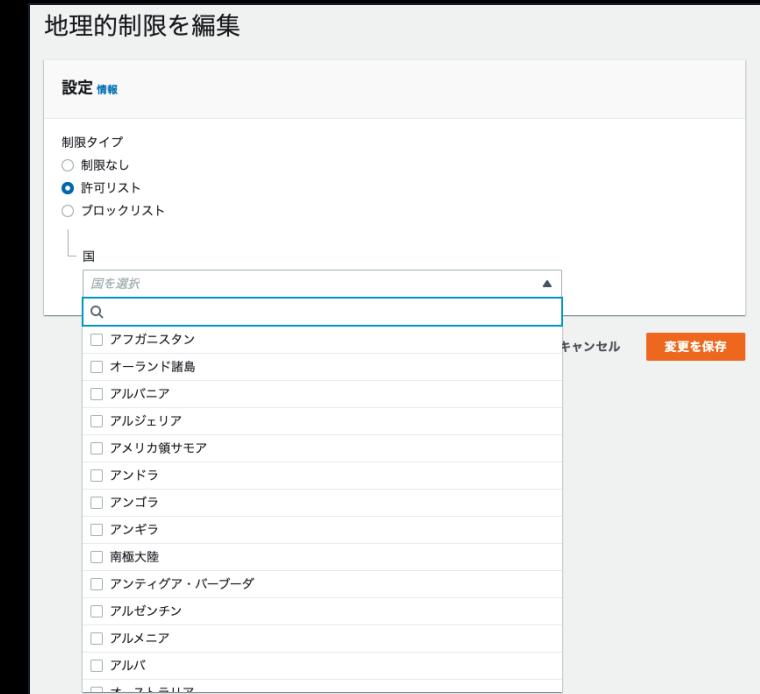
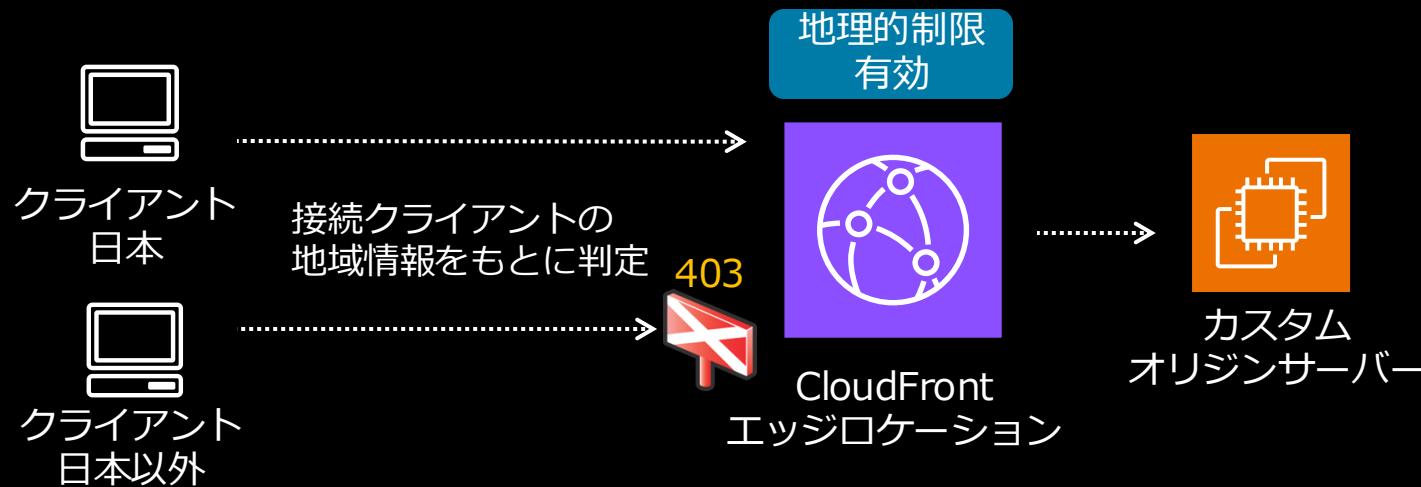


主要機能についての補足

地理的 (GEO) 制限

特定の国のユーザーに対するアクセス制御

- クライアント IP アドレスの地域情報を元に、エッジロケーションでアクセス判定
- 許可リストまたはブロックリストで指定可能
- ディストリビューション全体に適用される
- 制限されたアクセスには **403** を応答



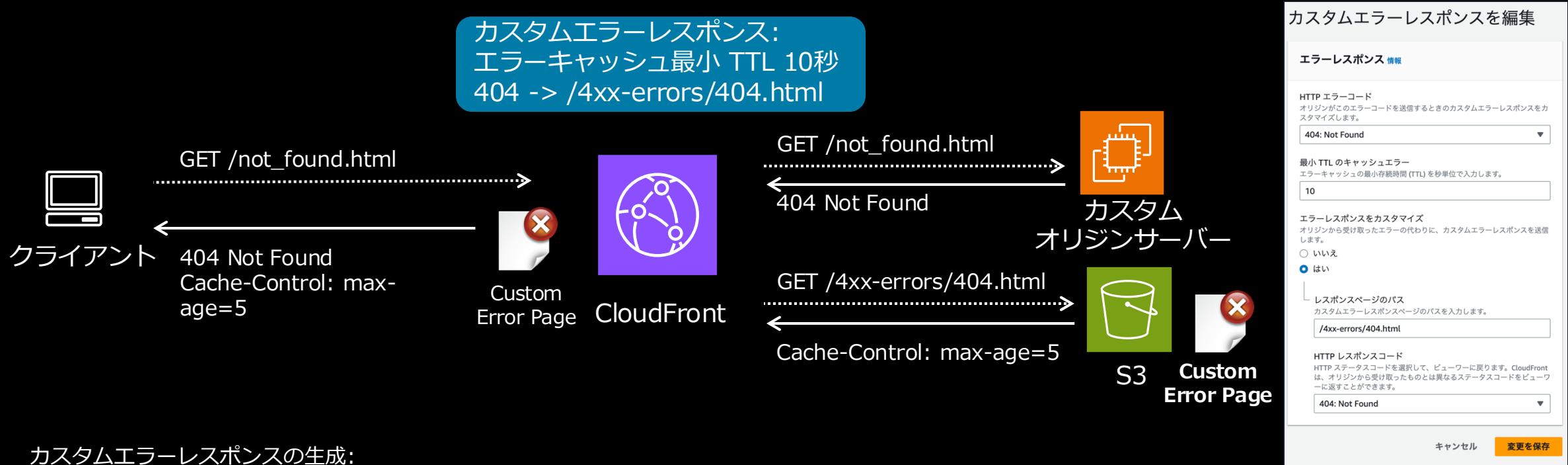
コンテンツの地理的ディストリビューションの制限

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/georestrictions.html

カスタムエラーレスポンス

オリジンサーバーのエラー受信時にレスポンスをカスタマイズ

- CloudFront は、エラーレスポンスをデフォルト 10 秒キャッシュ
- 4xx および 5xx ステータスコードそれぞれに対して、エラーレスポンスおよびレスポンスステータスコードのカスタマイズが可能



カスタムエラーレスポンスの生成:

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/GeneratingCustomErrorResponses.html



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

キャッシュファイルの無効化 (Invalidation)

CloudFront のキャッシュをパス単位でページ

- コンテンツ毎の無効化パス指定
 - 同時に最大 3,000 個までのパス指定が可能
- ワイルドカードを利用した無効化パス指定
 - 同時に最大 15 個まで無効化パスリクエストが指定可能
 - オブジェクト数は制限無く、1 無効化パスとして計算
- 月間最初の 1,000 パスまでは追加料金無し、それ以降は、無効リクエストしたパスごとに \$0.005



ファイルの無効化:

https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/Invalidation.html

ログ / メトリクス / レポート



CloudFront ログのクエリ:

https://docs.aws.amazon.com/ja_jp/athena/latest/ug/cloudfront-logs.html

まとめ

CloudFront の特徴

① AWS グローバルネットワークの利用

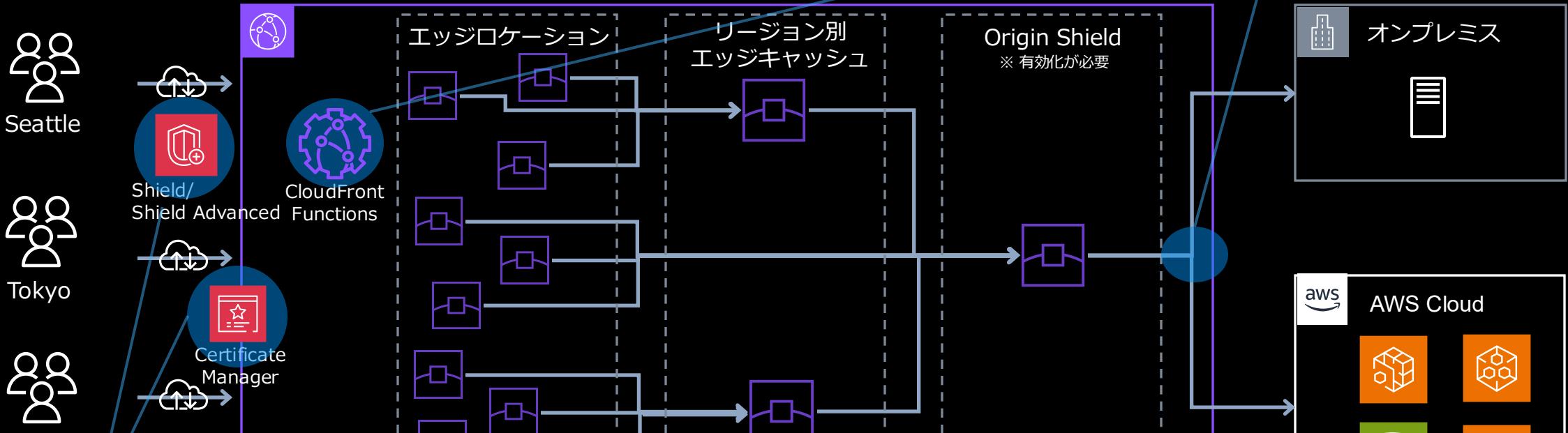
- ・ 高速かつ安定したネットワークでコンテンツ配信
- ・ キャッシュしない動的コンテンツでも高速化が見込める

⑤ エッジコンピューティングの利用

- ・ CloudFront Functions
- ・ Lambda@Edge

④ オリジンへのアクセス制御

- ・ OAC



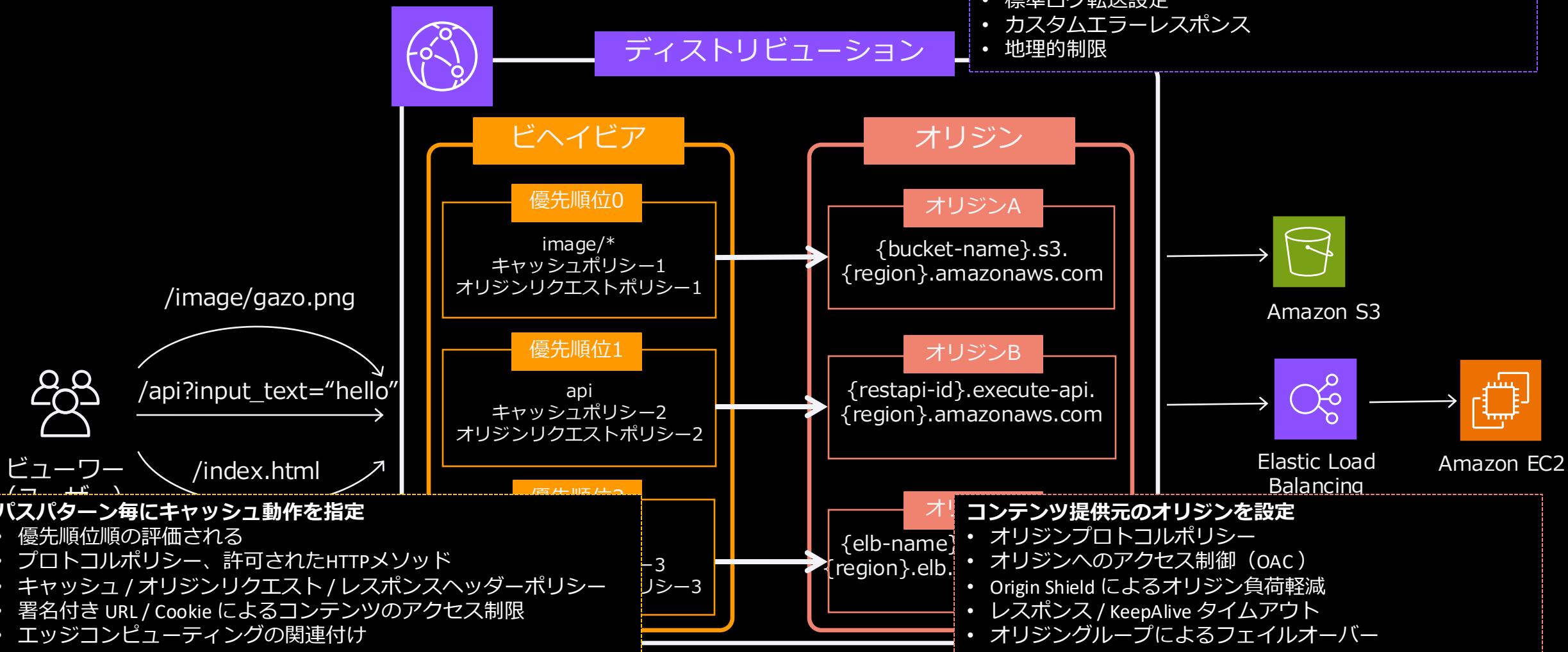
③ セキュリティサービスとの統合

- ・ DDoS 攻撃対策
- ・ SSL 証明書
- ・ WAF

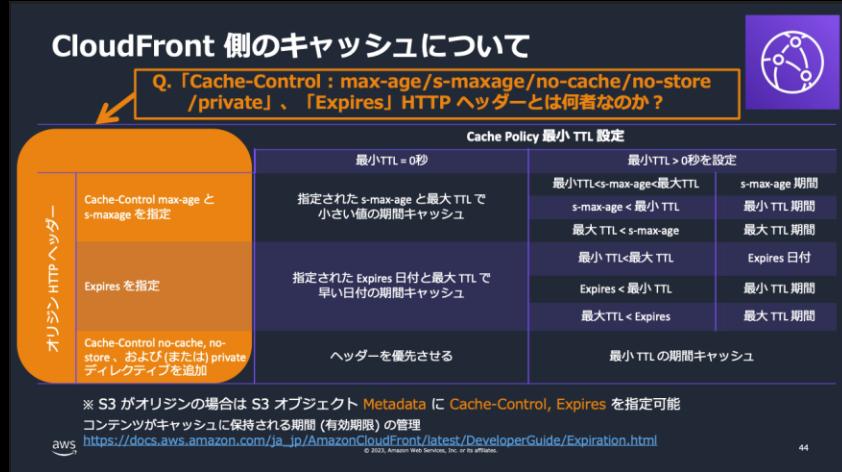
② 多段キャッシュ

- ・ レスポンスタイムの高速化
- ・ オリジン負荷の低減
- ・ フラッシュクラウドからのオリジン保護

CloudFront の設定



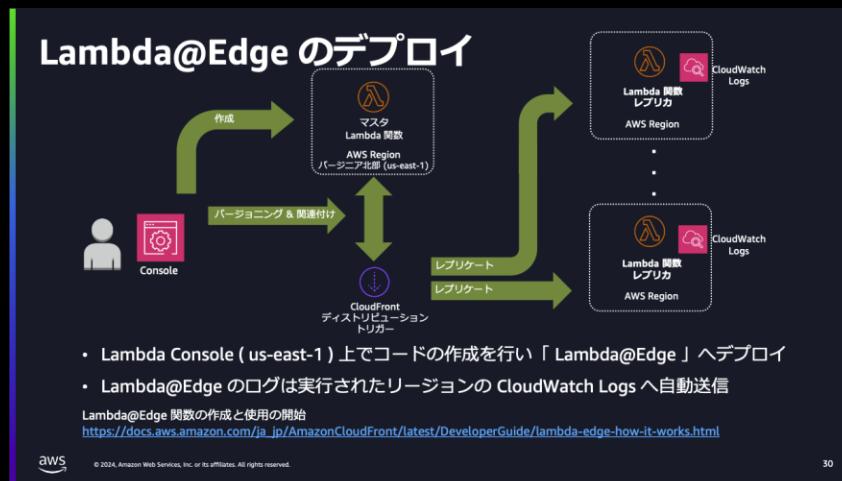
CloudFront のテーマ別 Blackbelt の紹介



Cache Control 編



レポート / モニタリング / ロギング 編



Edge Computing 編

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS サービス別資料

Thank you!