[AWS Black Belt Online Seminar]

Amazon CloudFront deep dive

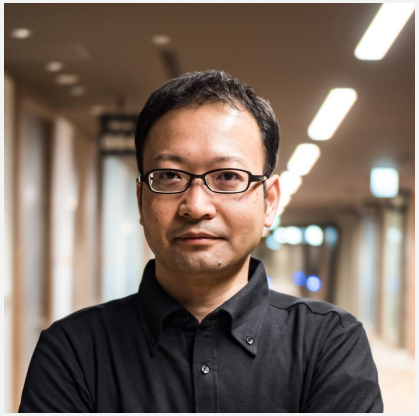
サービスカットシリーズ

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**

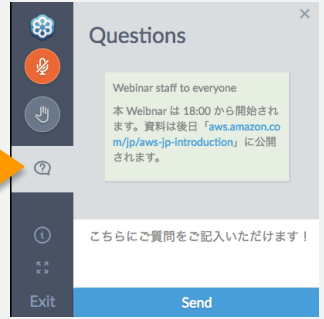
AWS Black Belt Online Seminar とは

「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、アマゾン ウェブ サービス ジャパン株式会社が主催するオンラインセミナーシリーズです。

質問を投げることができます！

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

① 吹き出しをクリック
② 質問を入力
③ Sendをクリック



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

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。

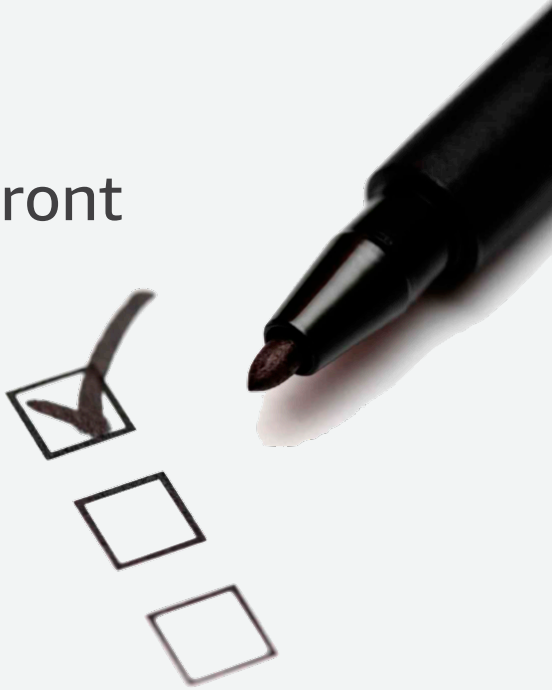
- 資料作成には十分注意しておりますが、資料内の価格とAWS公式ウェブサイト記載の価格に相違があった場合、AWS公式ウェブサイトの価格を優先とさせていただきます。

- 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 に関連する機能
- まとめ

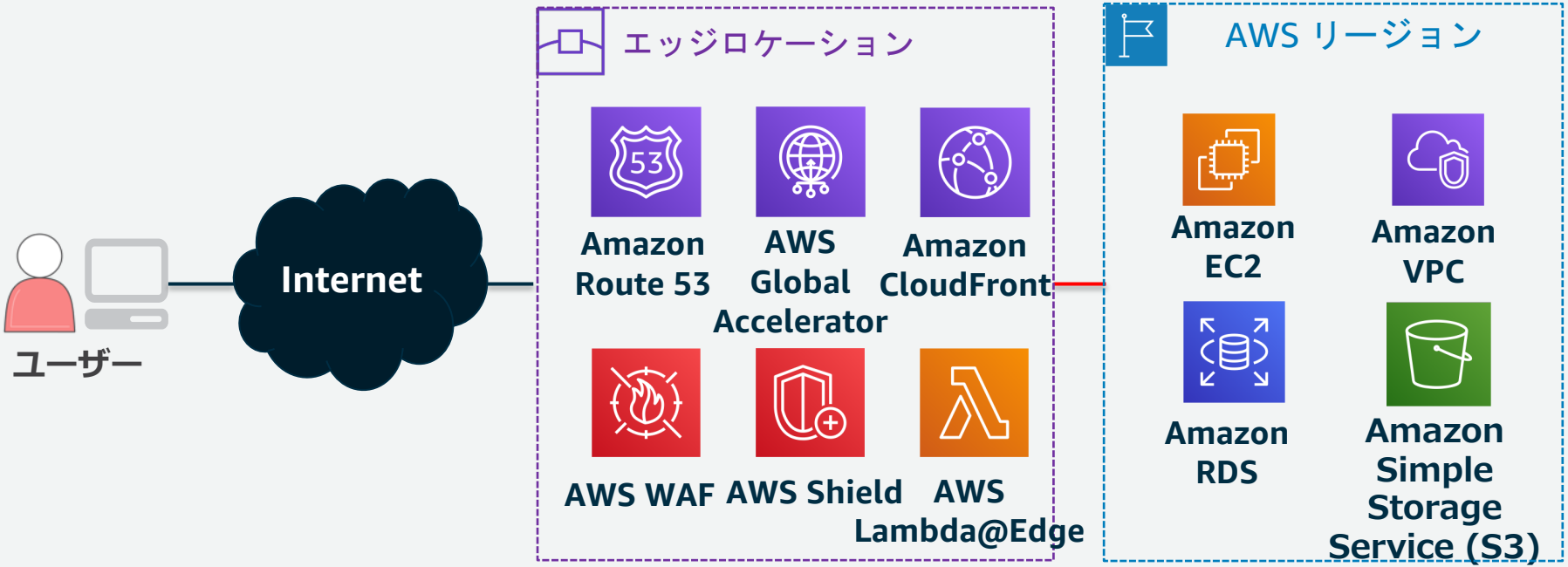


AWS のエッジサービス と Amazon Cloud Front

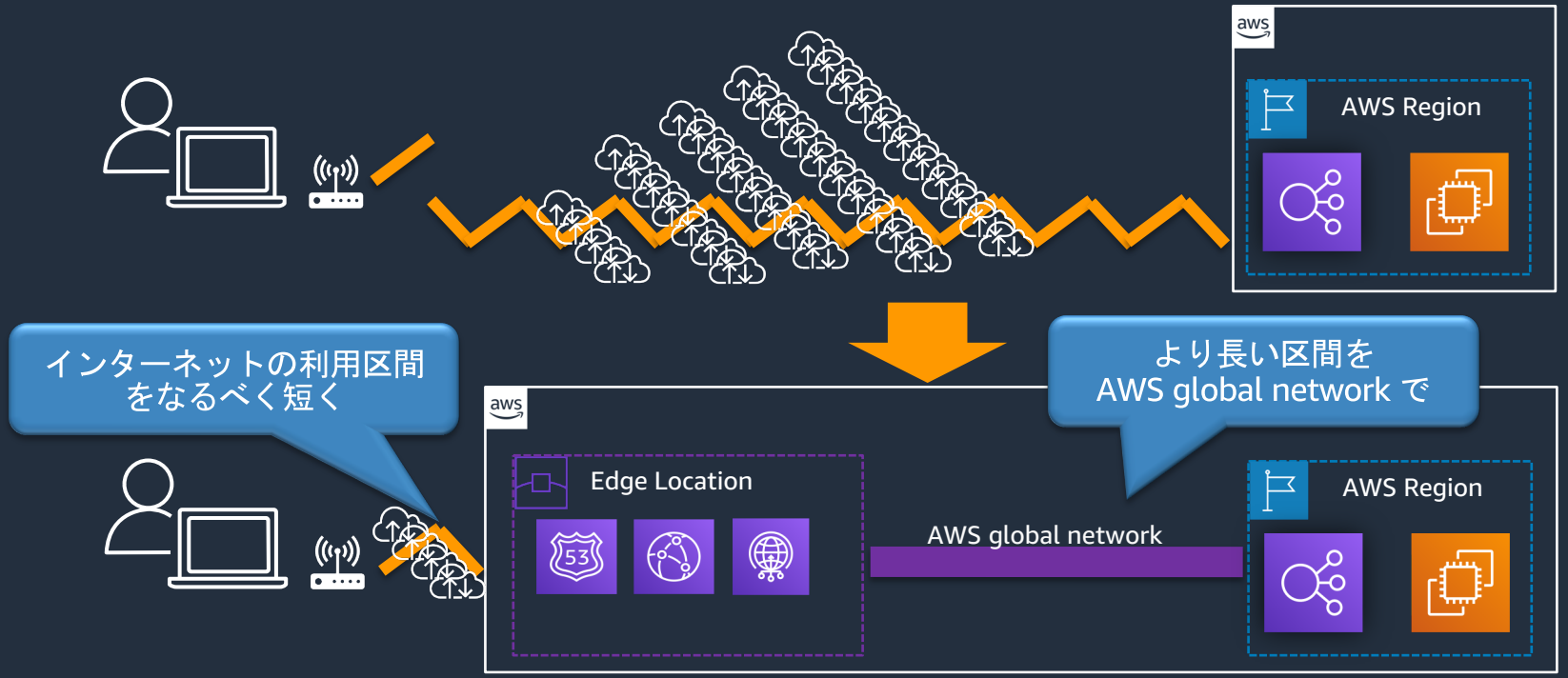
aws

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

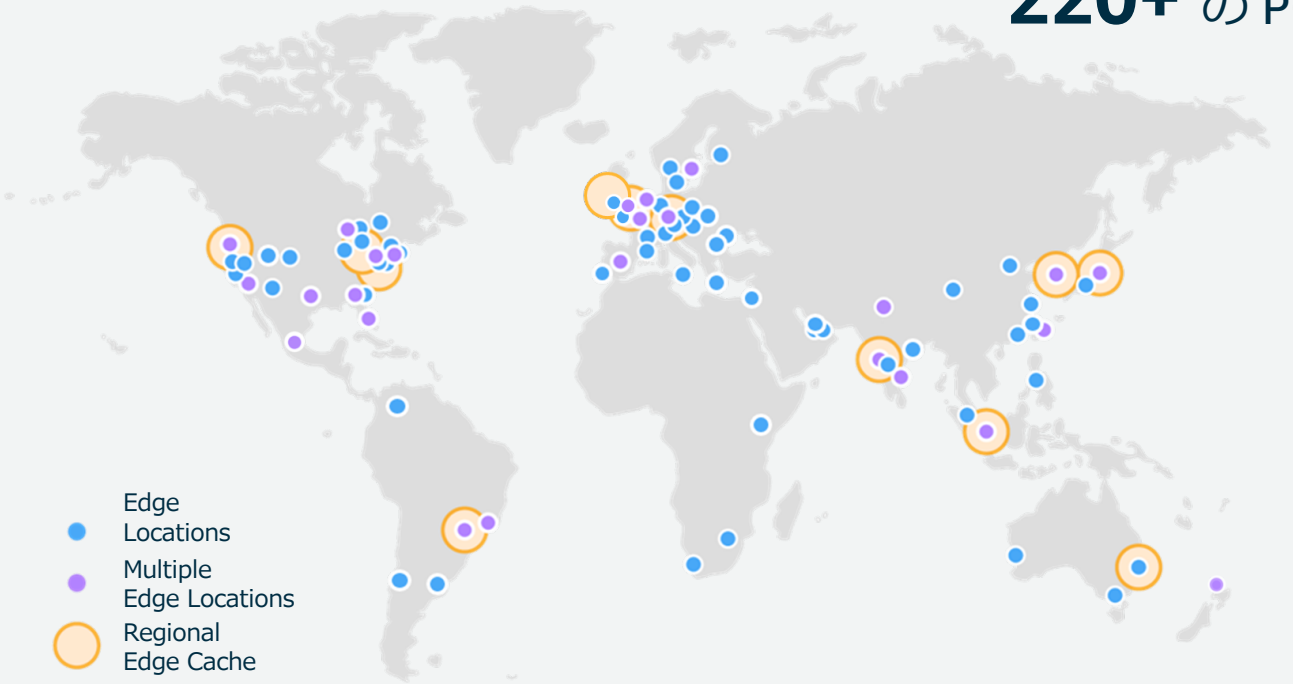


エンドユーザーの近くで DNS 名前解決、コンテンツ配信

- 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

エッジロケーション

As of 10/28/2020



220+ の POP (Point Of Presence)

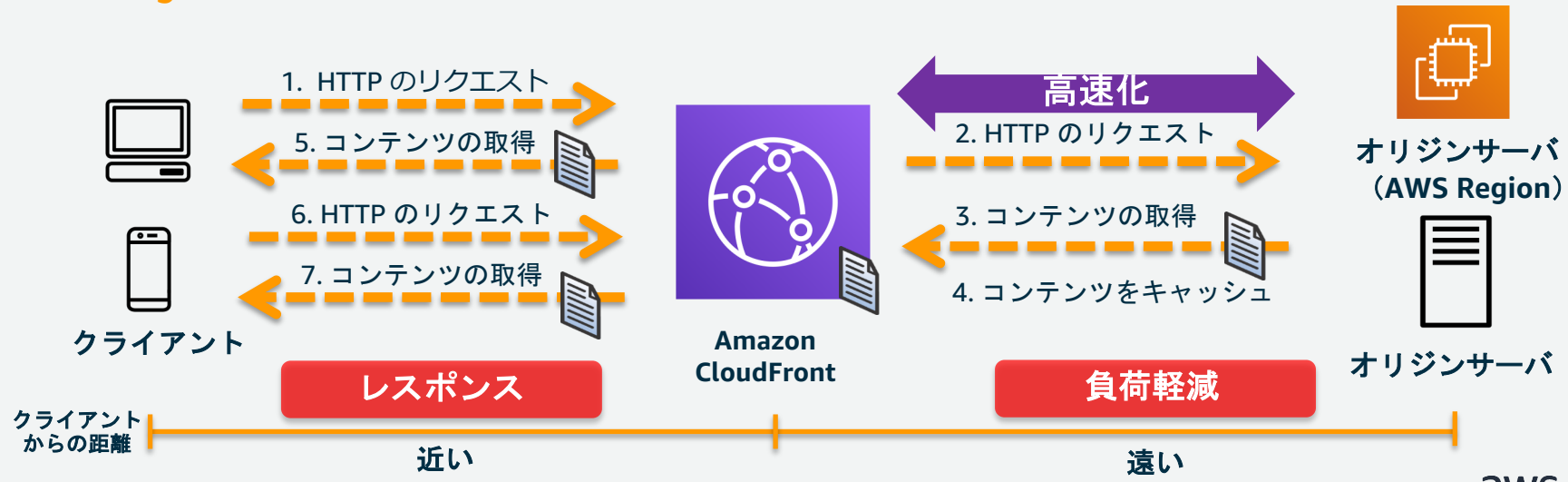
12 のリージョン別
エッジキャッシュ

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ュコンテンツの高速化**



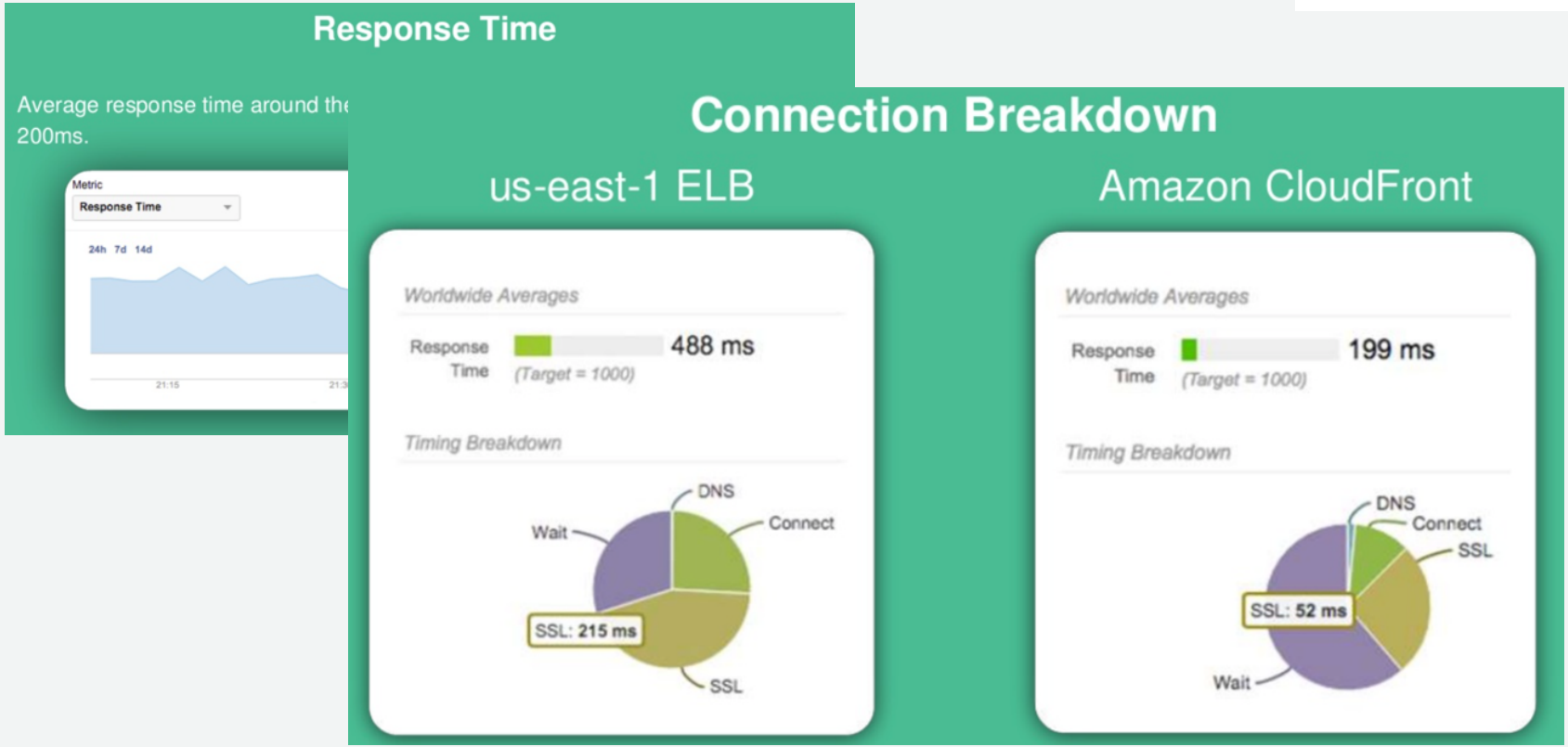
ユーザー事例 - API アクセラレーション

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

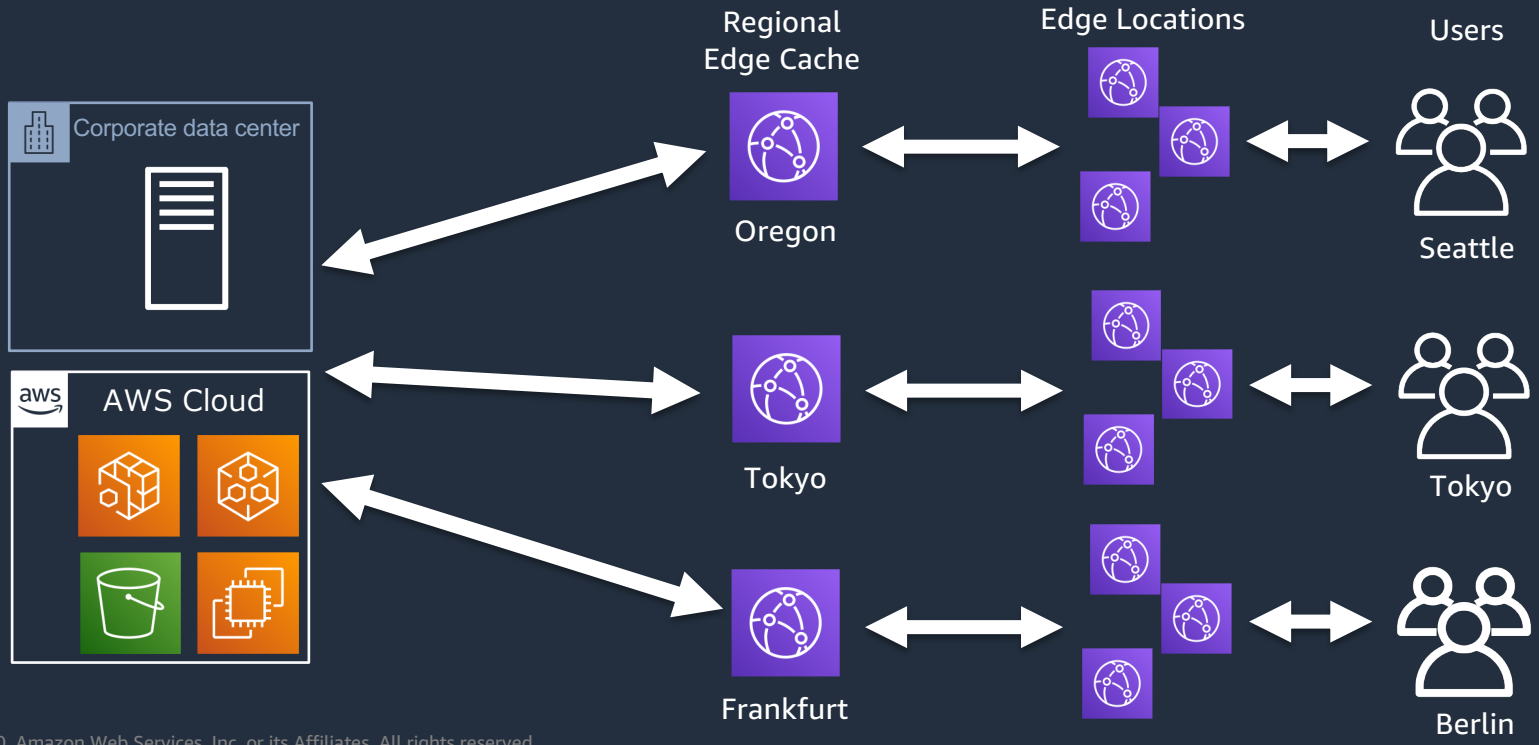
slack



aws

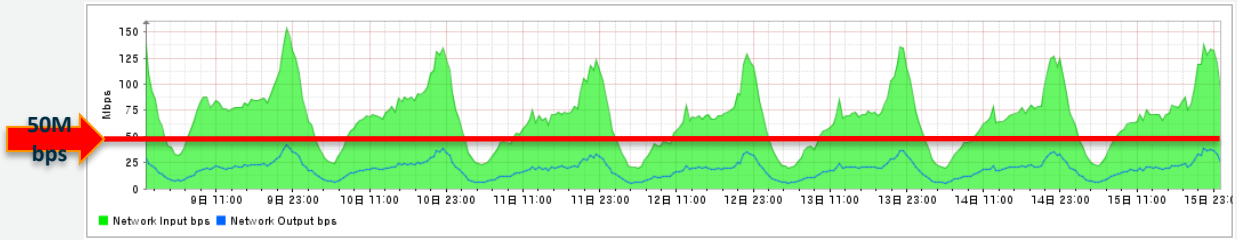
リージョン別エッジキャッシュ (REC)

- 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

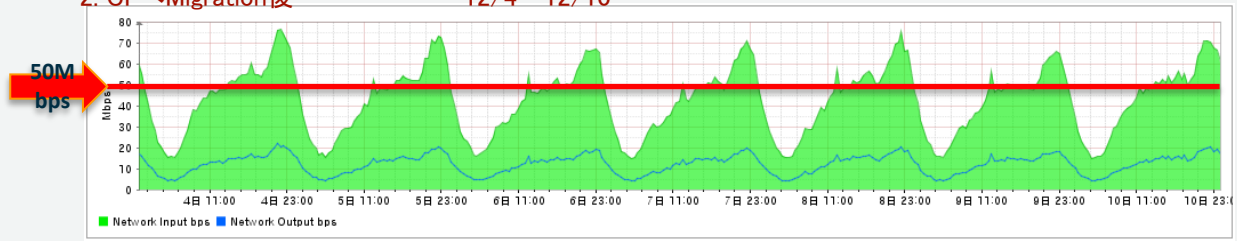


他 CDN から CloudFront に移行後オリジントラフィックが約 7 分の 1 に減少したお客様事例

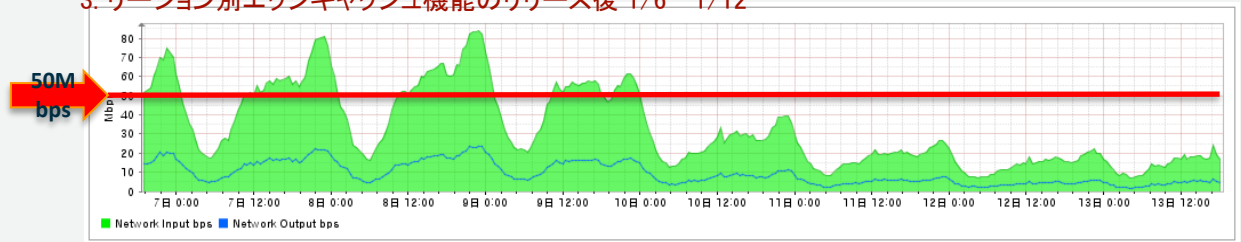
1. 他CDN使用時 10/9 ~ 10/15



2. CFへMigration後　　12/4 ~ 12/10

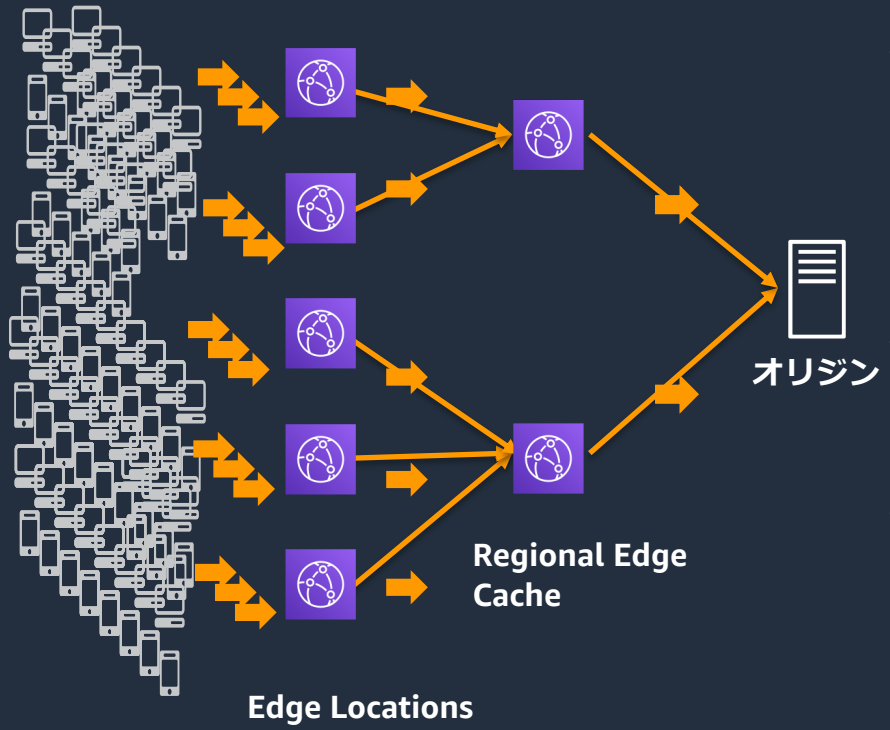


3. リージョン別エッジキャッシュ機能のリリース後 1/6 ~ 1/12



オリジンインフラの保護

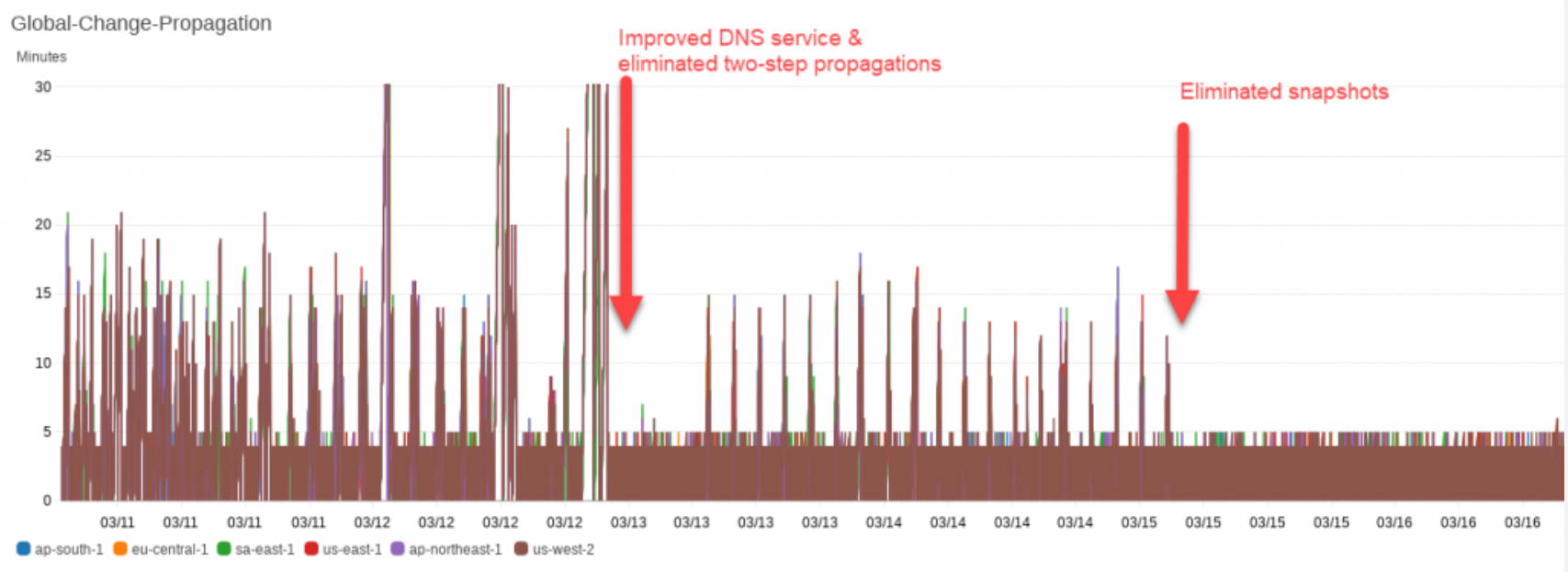
Automatic Flash Crowd Protection



- 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/

aws

CloudFront クォータの変更

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ォ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

aws

CloudFront deep dive

aws

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

aws

CloudFront 設定

1. Distribution に関連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 に関連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 に関連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

aws

CloudFront 設定

続き

4. Distribution に関連する機能
    - 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

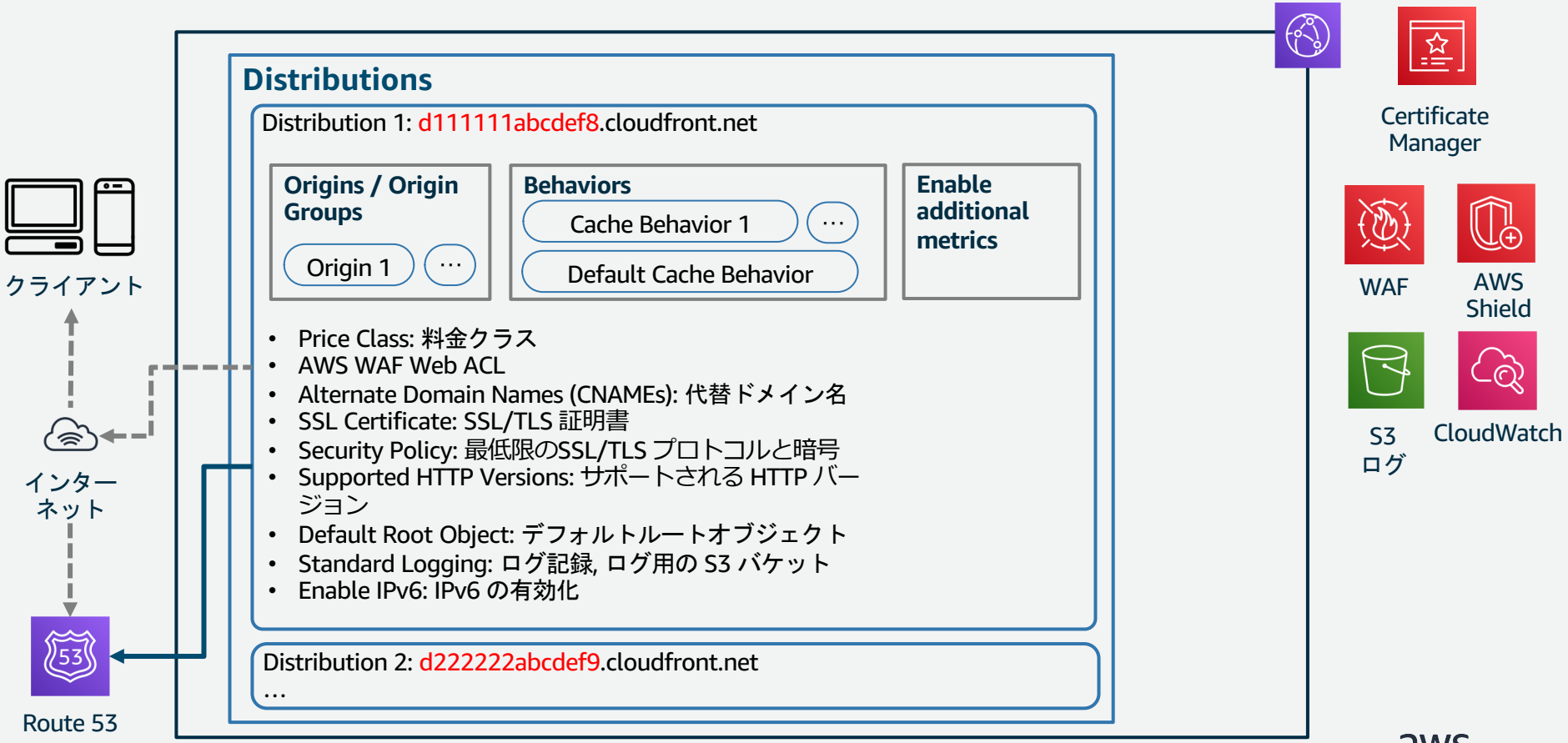
Distribution

aws

CloudFront 設定

1. Distribution に関連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 に関連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 に関連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

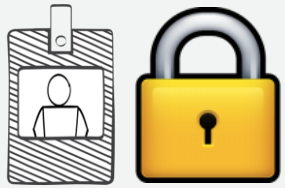


Distribution 概要

- 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

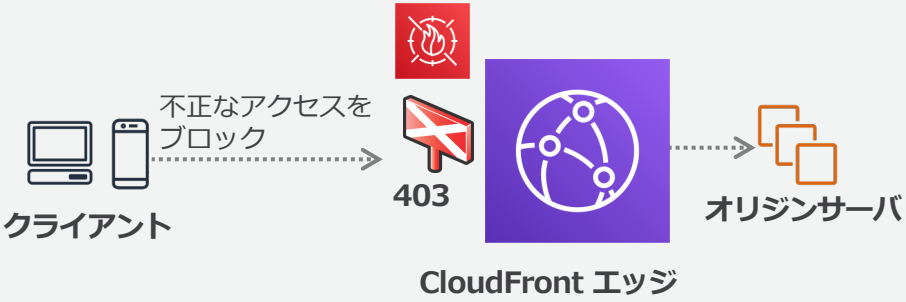
aws

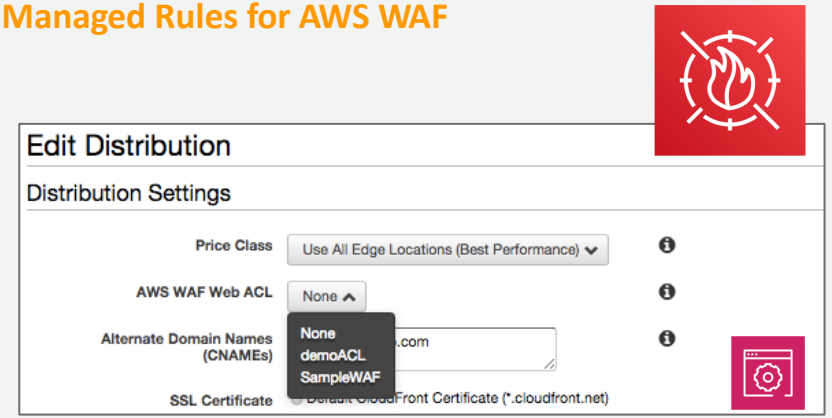
AWS WAF 連携



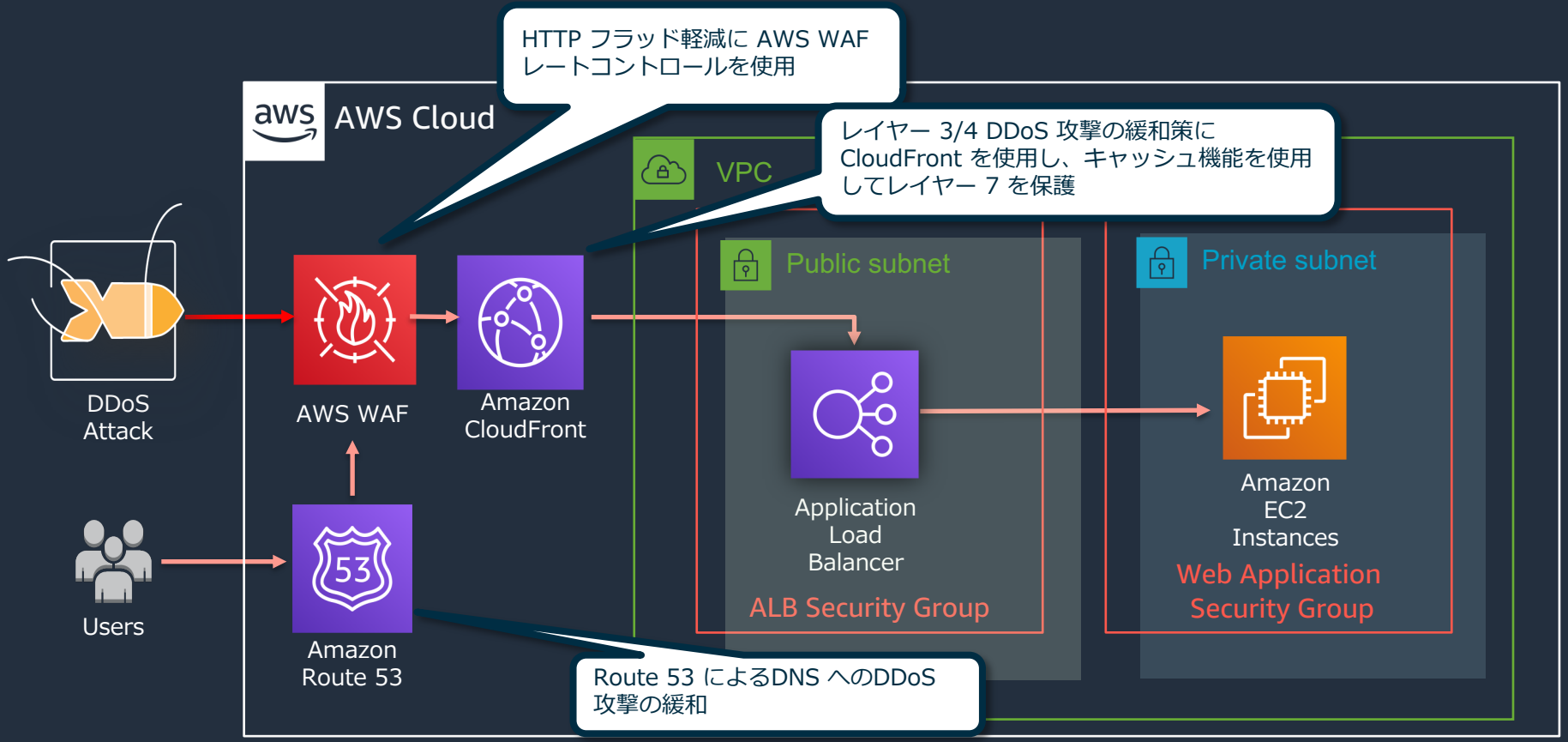
AWS WAF で定義した Web ACL を Distribution に適用

- 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 のビルトインマネージドルール: AWS Managed Rules for AWS WAF**
  - パートナーのマネージドルール
- ブロック時は 403(Forbidden) を応答



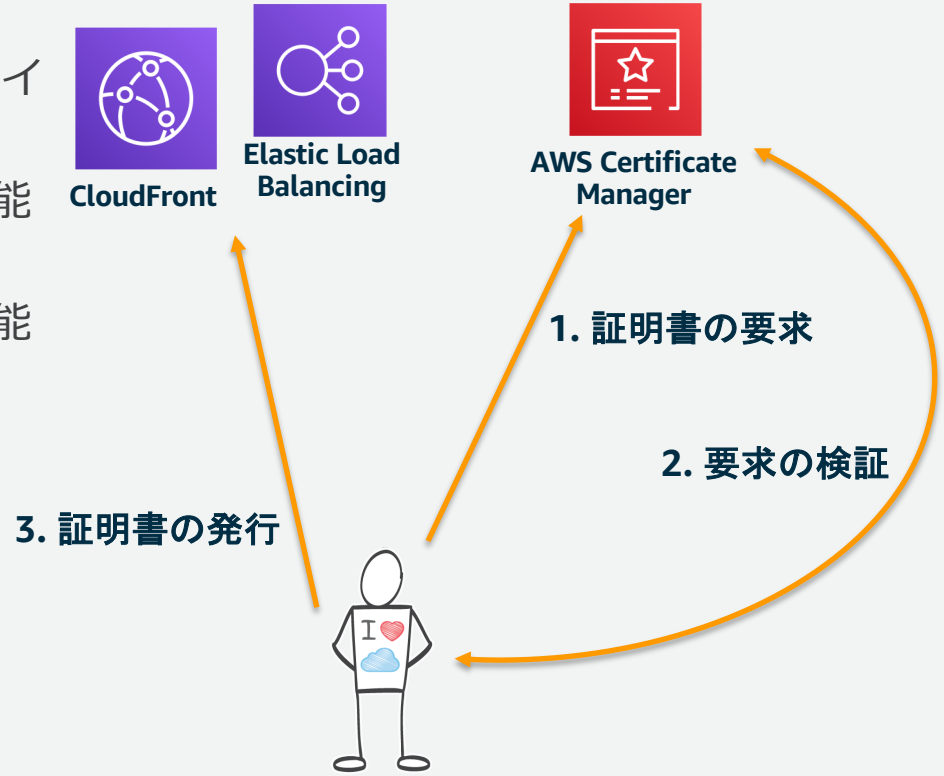


DDoS 耐性の高いアーキテクチャ

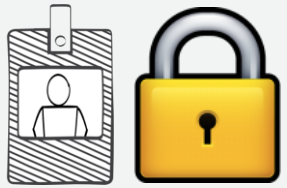


AWS Certificate Manager (ACM) との統合

- 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

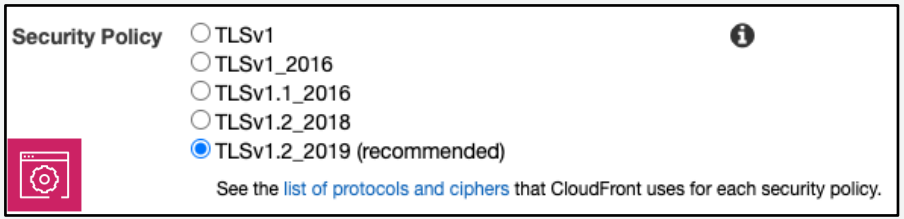


Viewer と CloudFront 間の SSL/TLS バージョンと暗号の組み合わせを選択

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

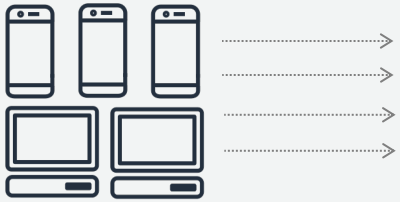
- 代替ドメイン名を使用する独自 SSL 証明書の利用時のみ指定可能
  - SNI SSL 証明書は TLSv1 以降のみ指定可能
  - SSLv3 は専用 IP アドレス SSL 証明書のみ指定可能



レポート / モニタリング / ログ

- キャッシュ統計
- 人気オブジェクト
- トップリファラ
- 使用状況
- ビューワー

アクセスや利用状況傾向の確認及び分析



クライアント

CloudFront

レポーティング

Management Console

リアルタイム
モニター

Cloudwatch

Monitoring / Alarms

- 障害/異常検知や現状の利用確認
- **8つの追加メトリクス**: キャッシュヒット率, オリジンレイテンシー, ステータスコード毎エラー率

New

アクセスログ

Access Log

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

S3

Amazon Athena

Amazon QuickSight

CloudFront ログのクエリ
https://docs.aws.amazon.com/ja_jp/athena/latest/ug/cloudfront-logs.html

Origin

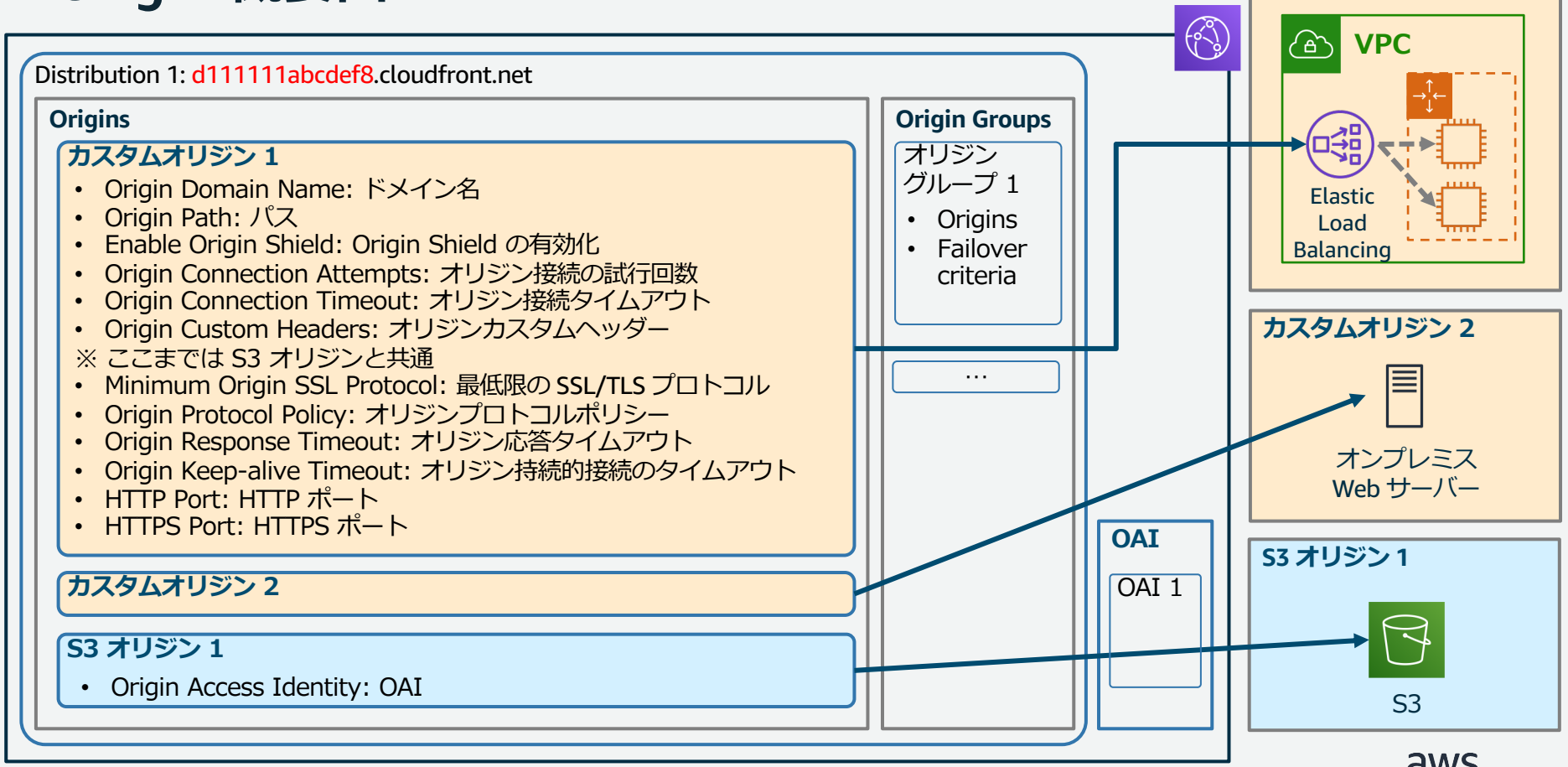
aws

CloudFront 設定

1. Distribution に関連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 に関連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 に関連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

aws

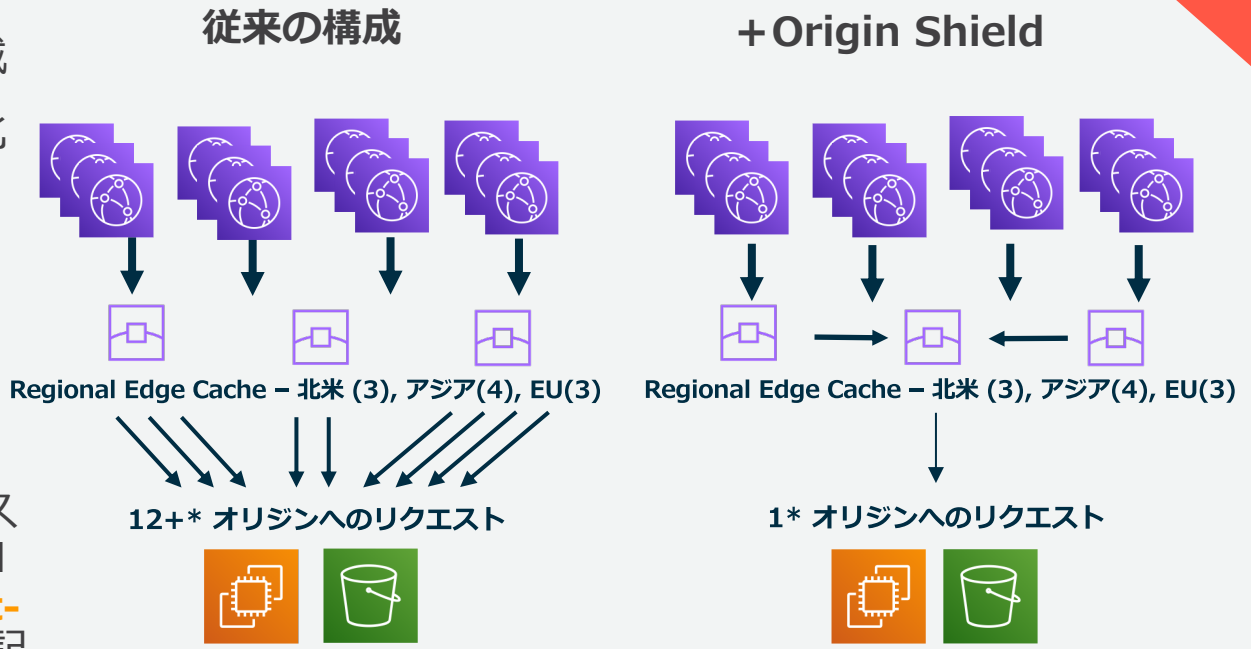
Origin 概要図



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



*リクエスト数は、トラフィック量/地域性/リクエスト率およびその他の要因によって異なる

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

aws

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

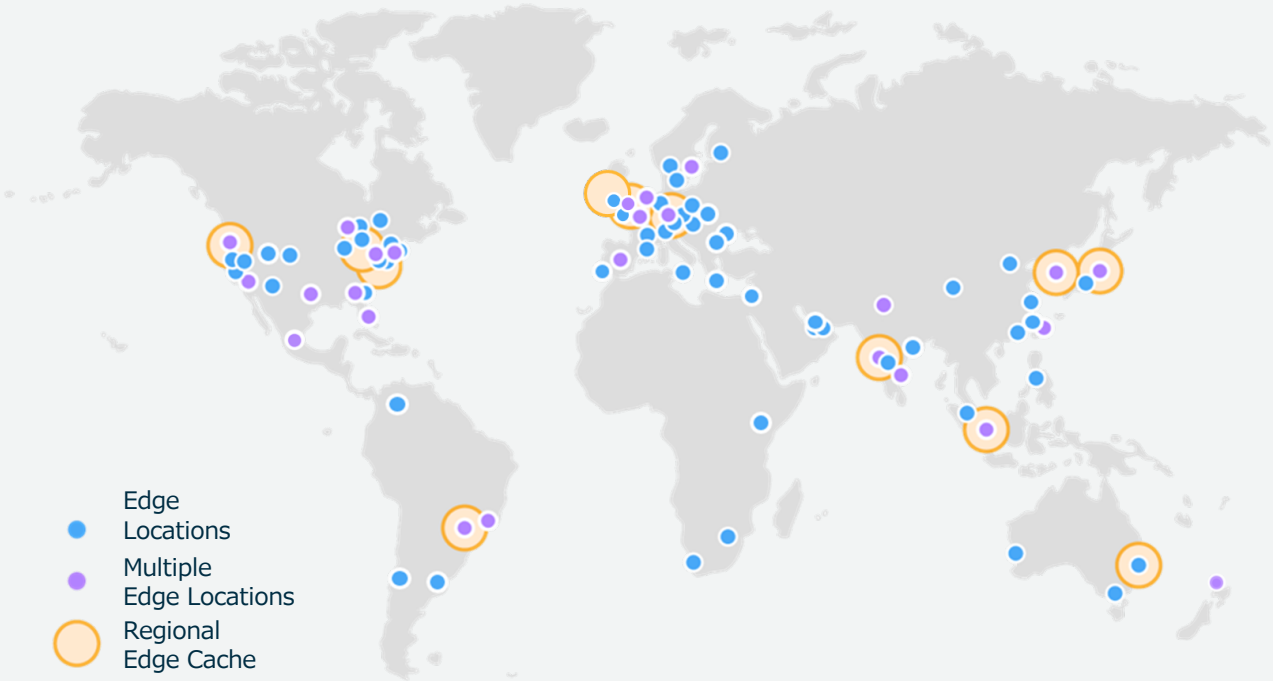
他リージョンの REC から Origin Shield へのアクセスは 1 万リクエスト毎に料金が発生する

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

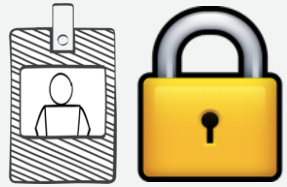
As of 10/28/2020



選択可能な Origin Shield ロケーション

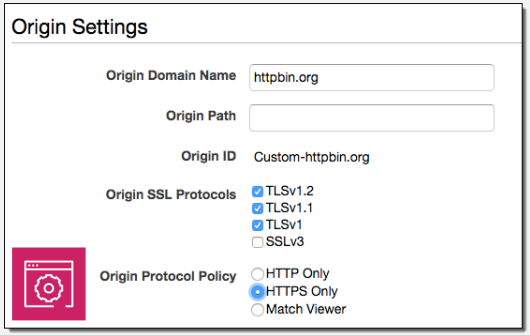
- US East (N. Virginia)
- US East (Ohio)
- US West (Oregon)
- South America (Sao Paulo)
- EU (Ireland)
- EU (London)
- EU (Frankfurt),
- Asia Pacific (Seoul)
- **Asia Pacific (Tokyo)**
- Asia Pacific (Singapore)
- Asia Pacific (Mumbai)
- Asia Pacific (Sydney)

カスタムオリジンの通信ポリシー

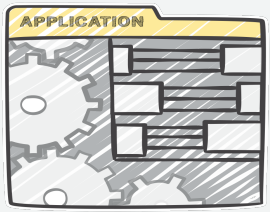


CloudFront エッジとオリジン間の通信方式を制御

- SSL/TLS プロトコル方式
  - TLSv1.2, TLSv.1.1, 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

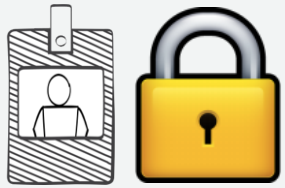
- CloudFront がカスタムオリジンからの応答を待つ時間を指定
- ビジー状態の負荷を軽減したり、 Viewer にエラー応答をより迅速に表示したりする場合は、応答タイムアウトを小さくする
- **デフォルトのタイムアウトは 30 秒**、 4～60 秒の範囲で設定可能

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

aws

オリジンカスタムヘッダー

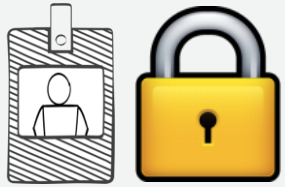


オリジンへの通信時にカスタム HTTP ヘッダーを追加

- 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

オリジンサーバの保護

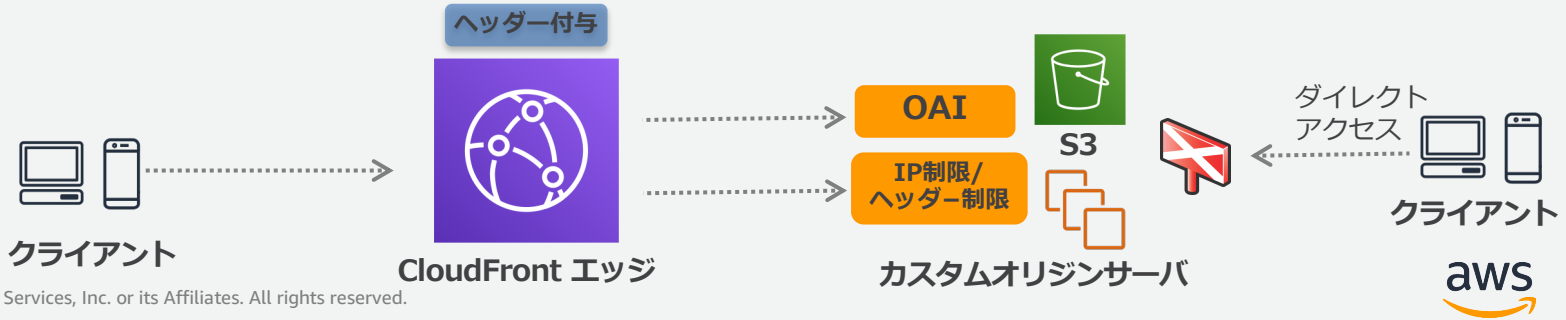


S3 オリジン

- 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

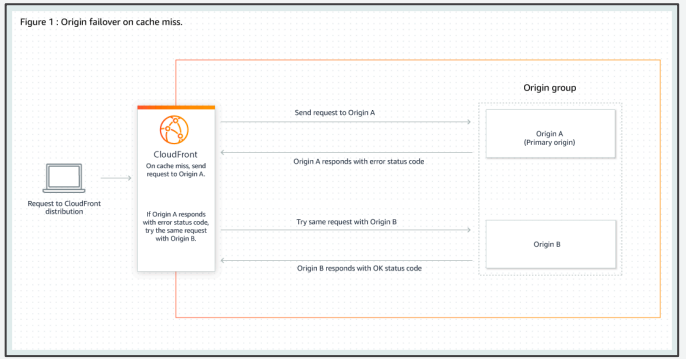
- オリジンカスタムヘッダーを利用し、指定された任意のヘッダーをオリジン側でチェック
  - **ALB のカスタムルールにて、 HTTP ヘッダーのチェックが可能**
- オリジン側のアドレスを公開しないとともに、CloudFront が利用する IP アドレスのみの許可させる
  - CloudFront が利用する IP アドレスは下記 URL から取得可能: https://ip-ranges.amazonaws.com/ip-ranges.json
  - JSON フォーマット: Service キーの “CLOUDFRONT” でフィルタすることで抽出可能

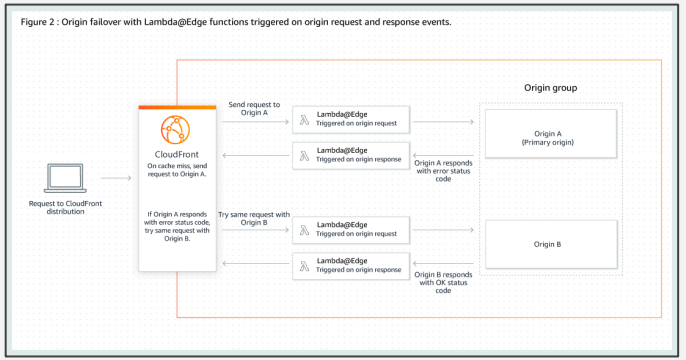


Origin Group によるオリジンフェイルオーバー

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: オリジン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

aws

CloudFront 設定

1. Distribution に関連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 に関連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 に関連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

キャッシュポリシー 1

- 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: Brotoli 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: オリジン転送 HTTP ヘッダー
- Cookies: オリジン転送 Cookie
- Query Strings: オリジン転送クエリ文字列

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

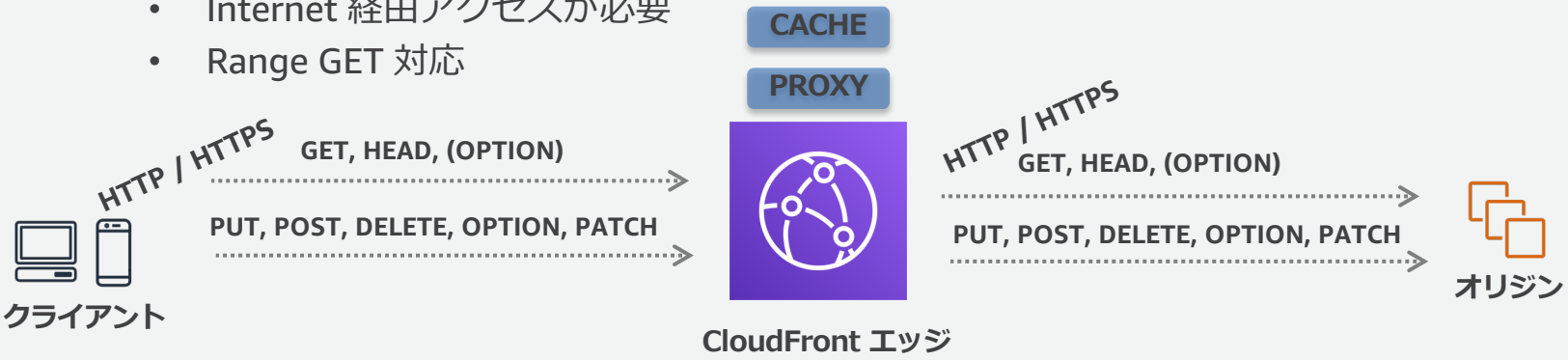
Cache Behavior 2: img/* → **S3 オリジン 1**

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

aws

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)
  - プロキシーモード: 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

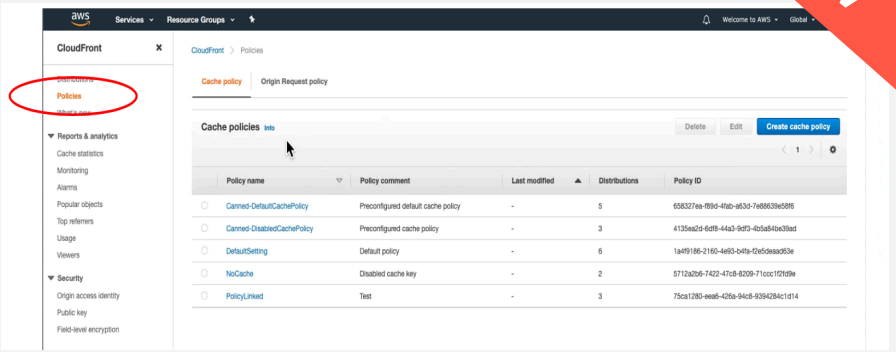
URL および Cache Policy で有効化した HTTPヘッダー, クエリ文字列, Cookie パラメータ値の**完全一致**でキャッシュが再利用される

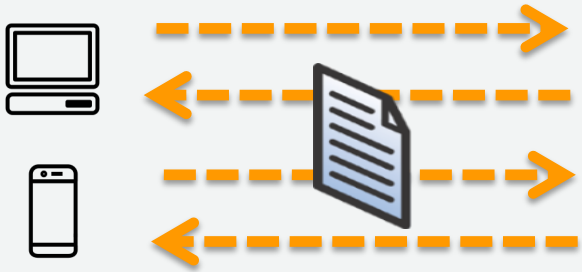
aws

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ュ設定が可能に
  - 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
- より高いキャッシュ効率が実現可能





Cache Policy
（キャッシュ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）

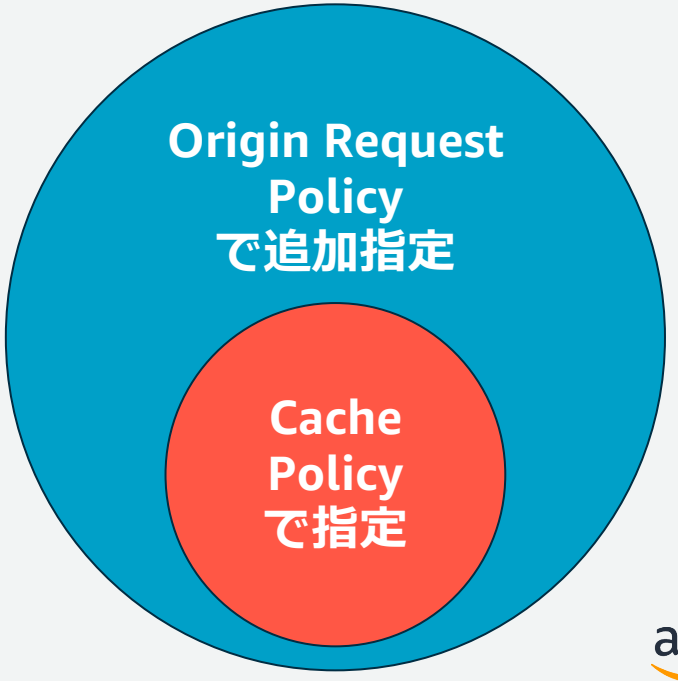
オリジンサーバ
（AWS Region or Custom Origin）

Amazon CloudFront 開発者ガイド(ポリシーの使用):
https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/working-with-policies.html

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

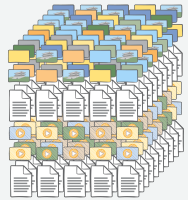
Behavior の **Origin Reuest Policy**:
- カスタム Origin Request Policy のオリジン転送 HTTP ヘッダー: **指定なし**

```
GET /content/stories/example-story.html
Host: cf-backend.example.com
User-Agent: Amazon CloudFront
X-Amzn-Trace-Id: Root=1-11111111-123456789abcdefghijklmno
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=1-22222222-123456789abcdefghijklmno
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 毎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：オリジンが 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		
		最小 TTL = 0 秒	最小 TTL >0 秒を設定	
オリジン HTTP ヘッダー	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
			max-age < 最小 TTL	最小 TTL 期間
			最大 TTL < max-age	最大 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)	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

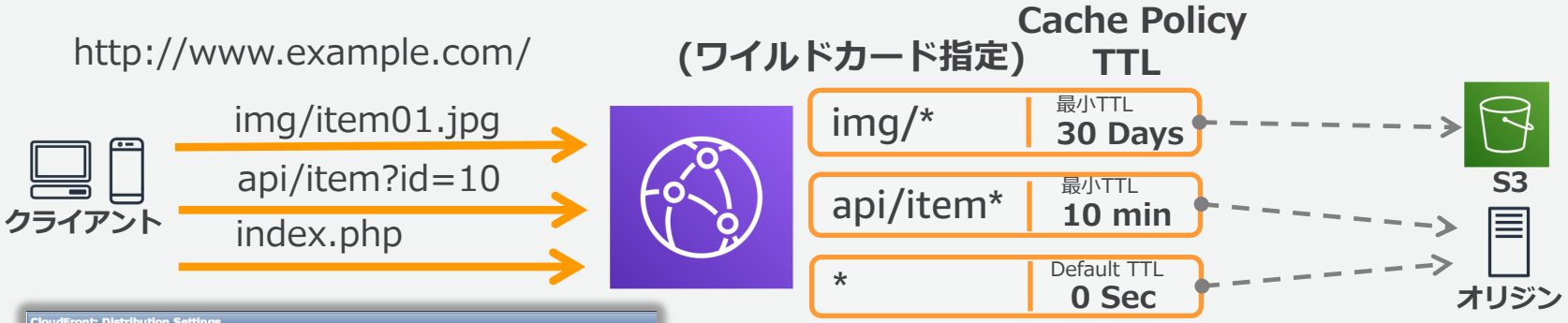
		Cache Policy Minimum TTL 設定		
		最小TTL = 0秒	最小TTL >0秒を設定	
オリジン HTTP ヘッダー	Cache-Control max-age とs-maxage を指定	指定された s-max-age と最大 TTL で小さい値の期間キャッシュ	最小TTL<s-max-age<最大TTL	s-max-age 期間
			s-max-age < 最小 TTL	最小 TTL 期間
			最大 TTL < s-max-age	最大 TTL 期間
	Expires を指定	指定された Expires 日付と最大 TTL で早い日付の期間キャッシュ	最小 TTL<<最大 TTL	Expires 日付
			Expires < 最小 TTL	最小 TTL 期間
			最大TTL < Expires	最大 TTL 期間
	Cache-Control no-cache, no-store を指定	キャッシュされない	最小 TTL の期間キャッシュ	

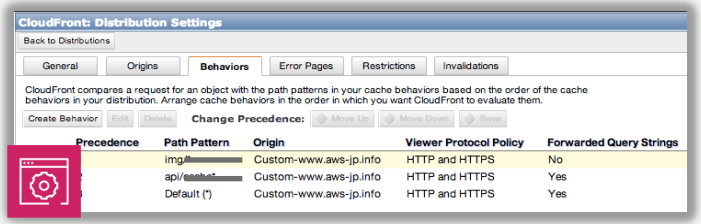
- S3 オリジンの場合は S3 オブジェクト Metadata に Cache-Control, Expires を指定可能
- HTML Meta タグに Cache-Control もしくは Pragma を指定しても CloudFront は利用しない

aws

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ュコントロール**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

aws

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

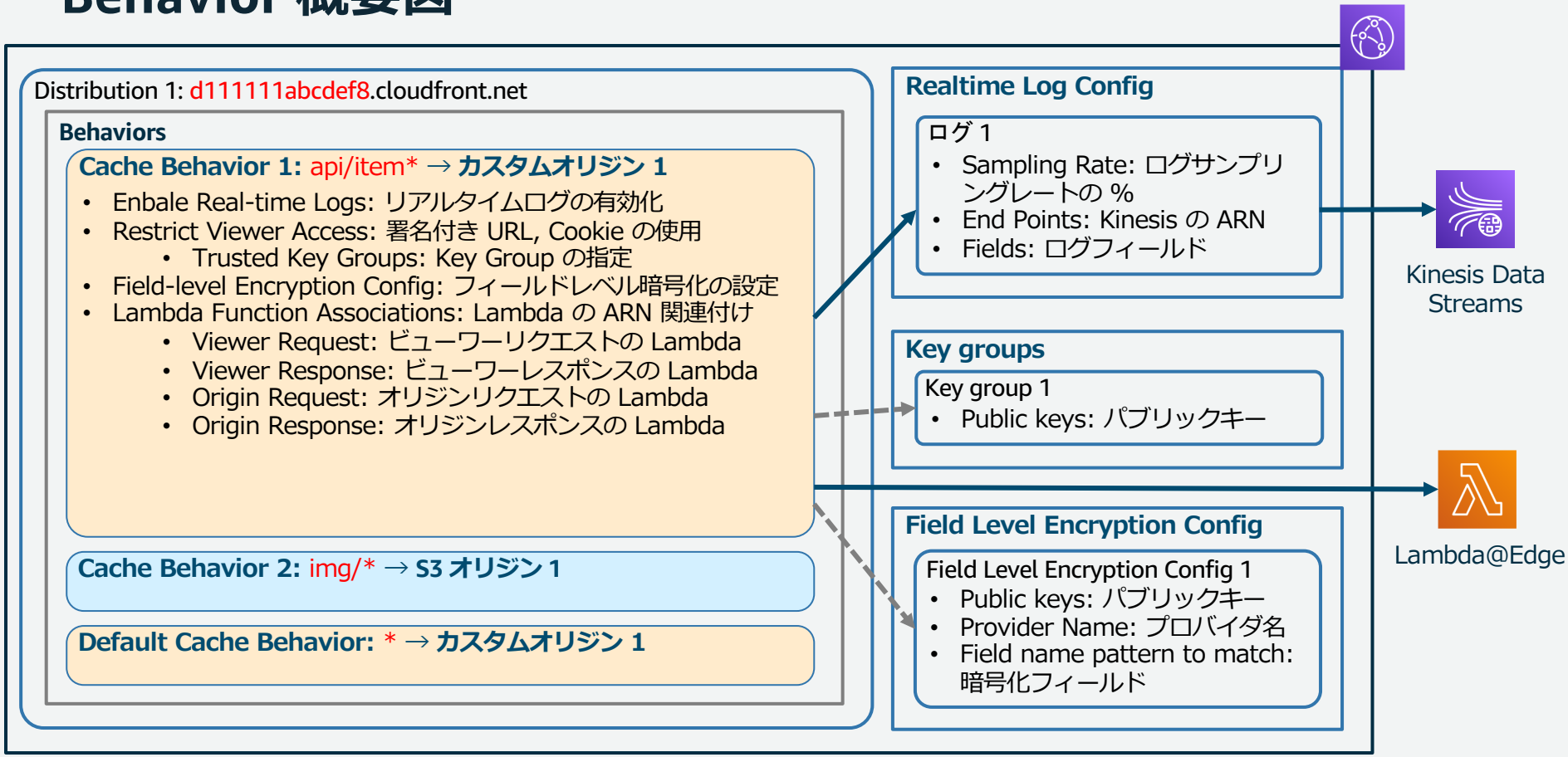
New

- 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

aws

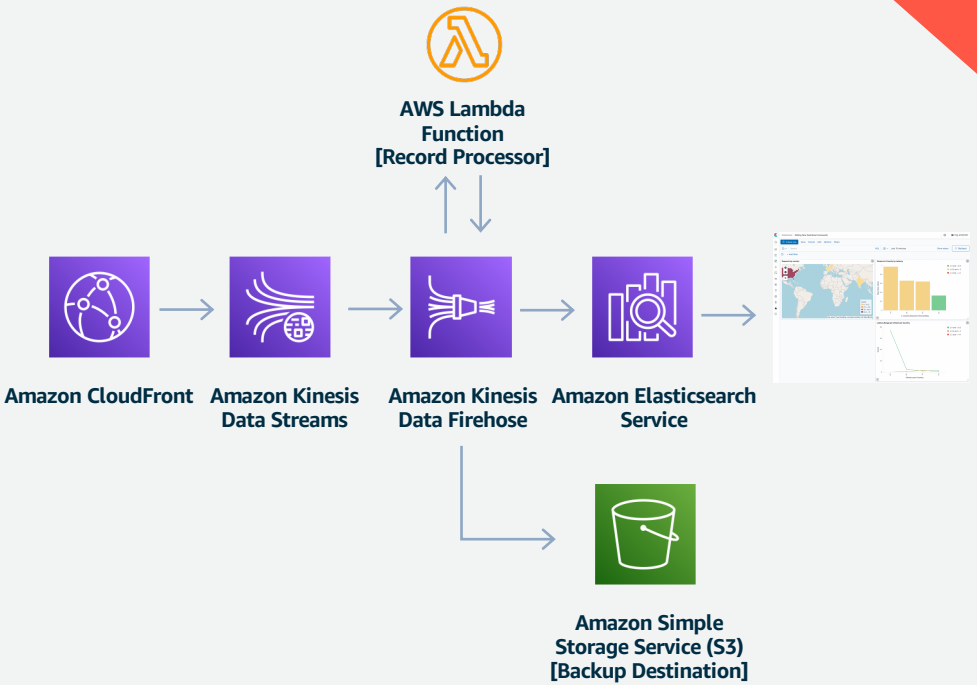
Behavior 概要図



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、サードパーティーのログ処理サービスにログを配信可能

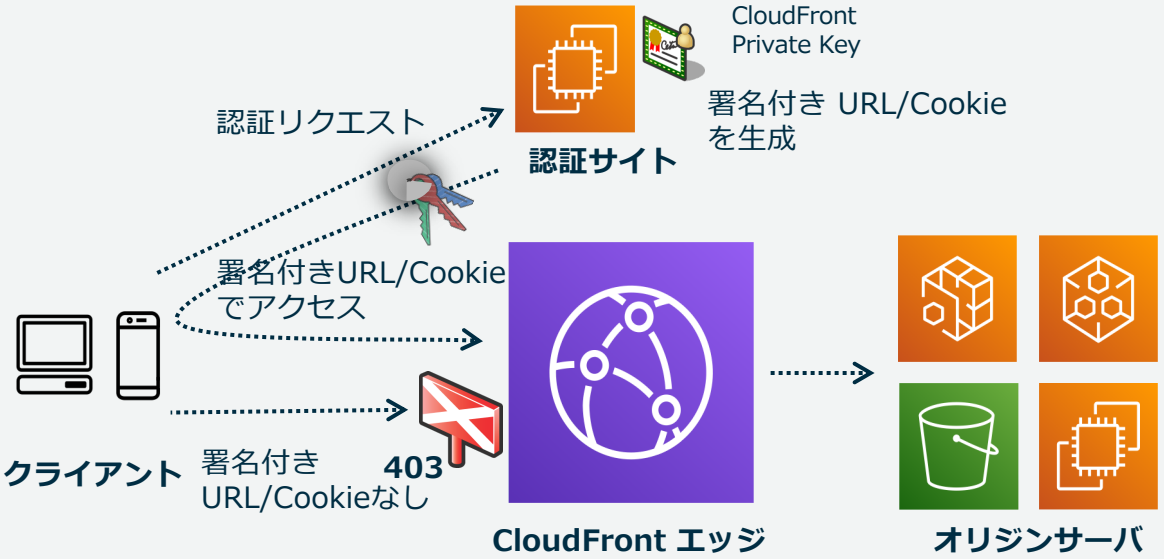


Amazon CloudFront 開発者ガイド(リアルタイムログ):
https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/real-time-logs.html

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

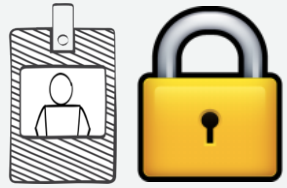
New



- **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

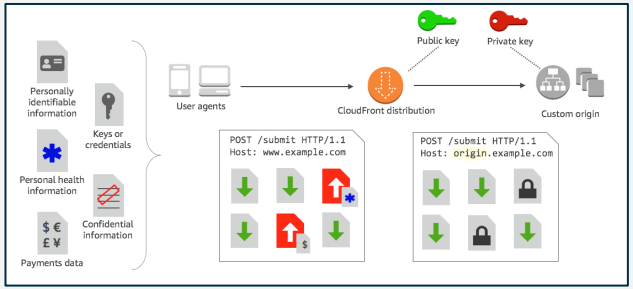
Amazon CloudFront 開発者ガイド(Choosing between trusted key groups (recommended) and AWS accounts):
https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/private-content-trusted-signers.html#choosing-key-groups-or-AWS-accounts

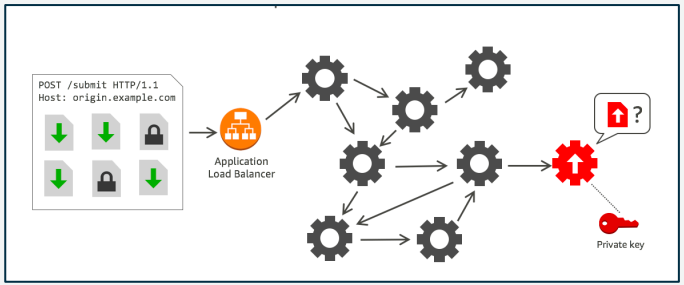
フィールドレベル暗号化



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

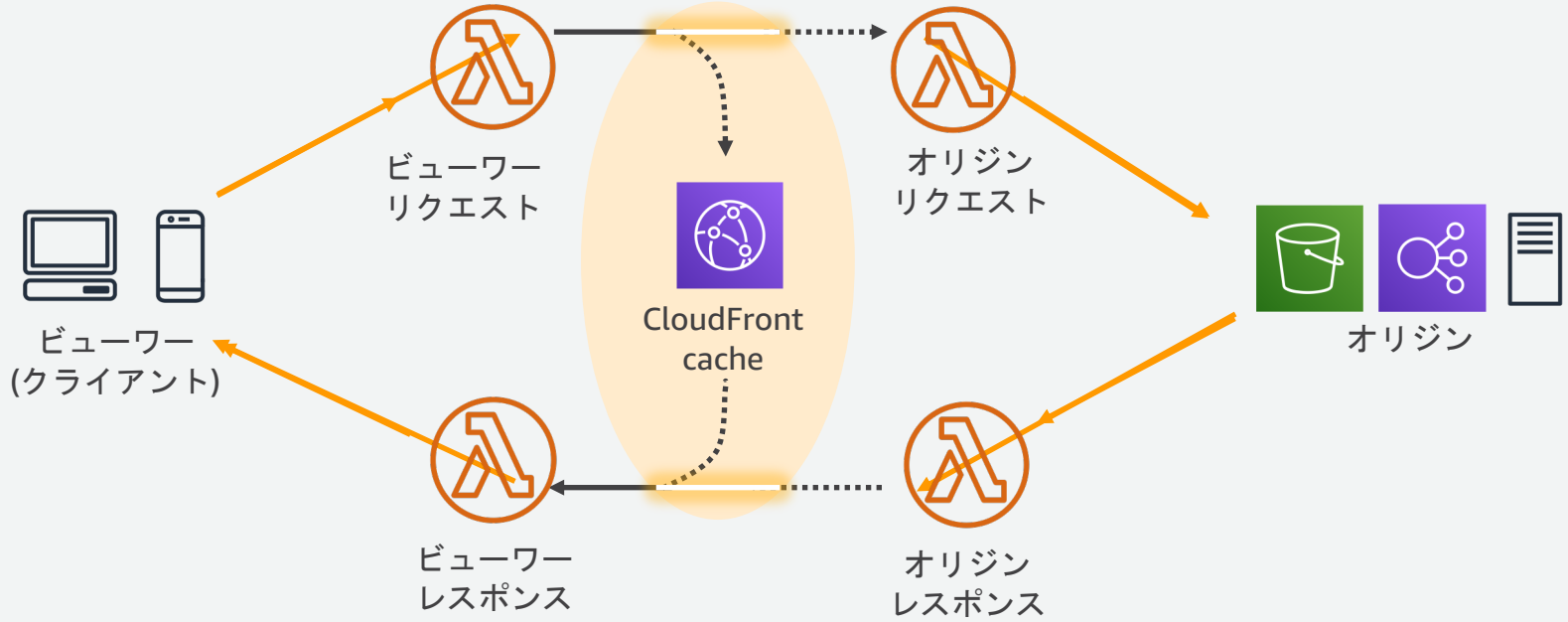
- 公開鍵暗号方式
- 設定方法
  1. RSA キーペアを取得
  2. パブリックキーを CloudFront に追加
  3. フィールドレベル暗号化のプロファイルを作成
  4. 暗号化を行うリクエストのコンテンツタイプを指定する設定を作成
  5. Behavior に設定を追加
  6. オリジンでデータフィールドを復号化
     - AWS Encryption SDK を使用
     - C, Java, Python, JavaScript, CLI を使用可能





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

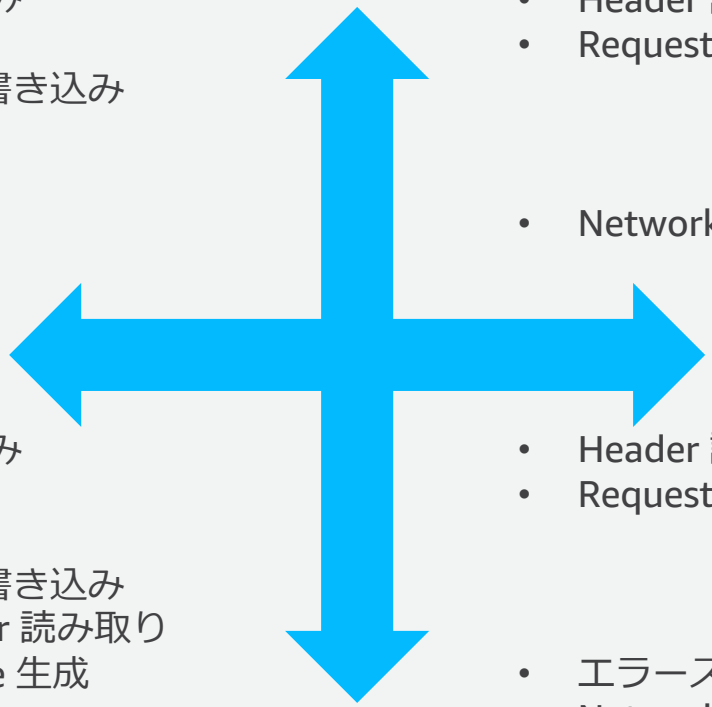
Lambda@Edge のプログラミングモデル

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)

aws

リクエストイベントごとの機能



- 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

- Header 読み取り/書き込み
- Request object 読み取り

- Network 呼び出し

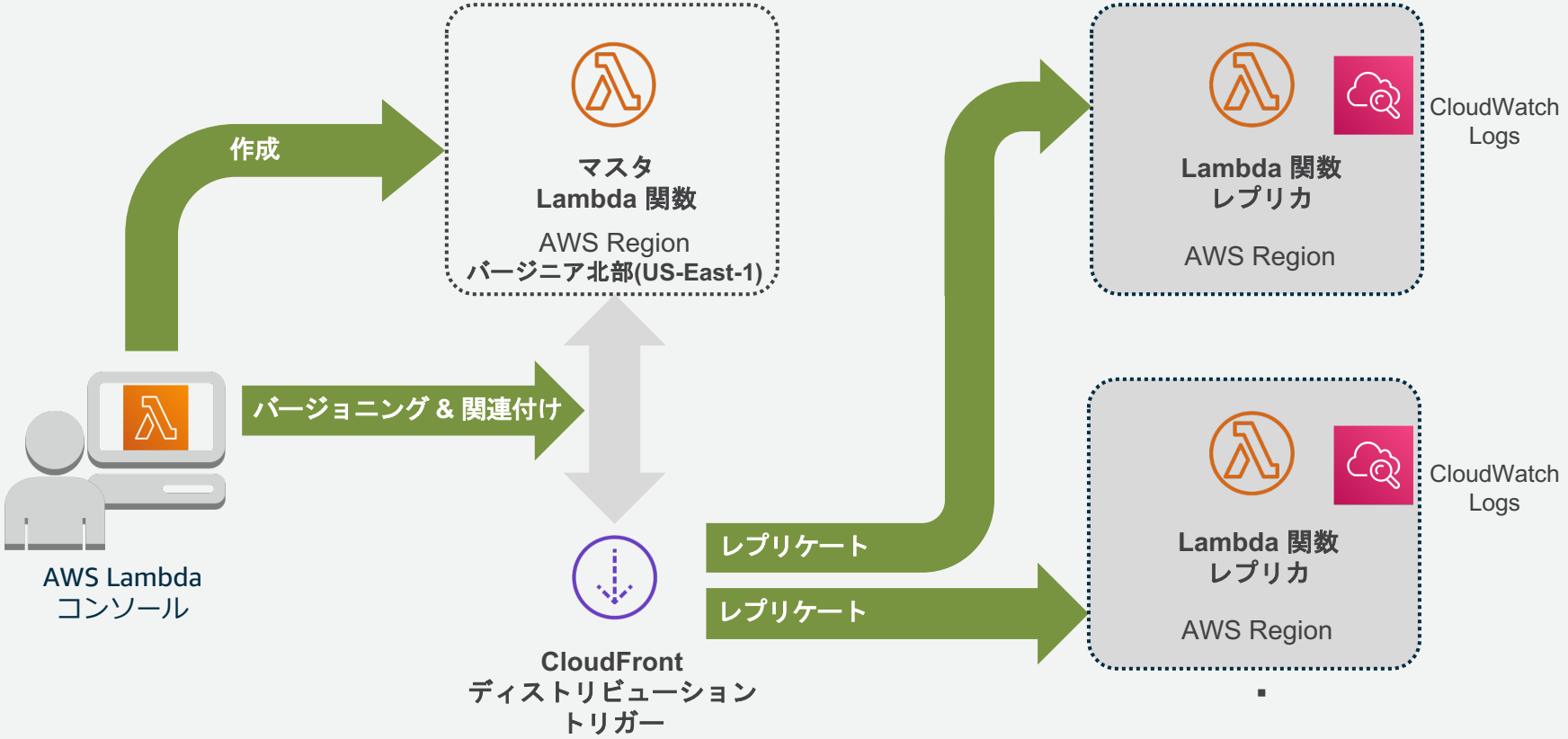
- Header 読み取り/書き込み
- Request Body 読み取り
- URL 読み取り/書き込み
- クエリ文字列 読み取り/書き込み
- CloudFront-* 追加 Header 読み取り
- バイナリを含む Response 生成
- Network 呼び出し
- S3オリジン,カスタムオリジンの変更
- 関数タイムアウト 30 秒

- 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

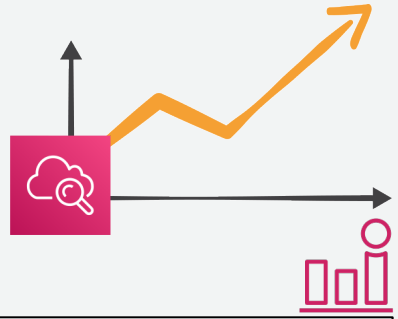


Lambda@Edge 関数の作成と使用の開始
https://docs.aws.amazon.com/ja_jp/AmazonCloudFront/latest/DeveloperGuide/lambda-edge-how-it-works.html

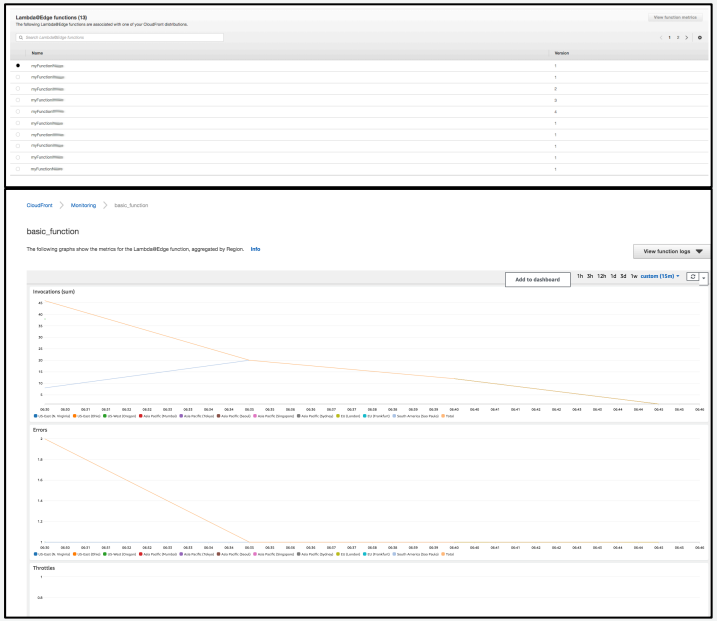
テストとデバッグ

- 「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 に関連する機能

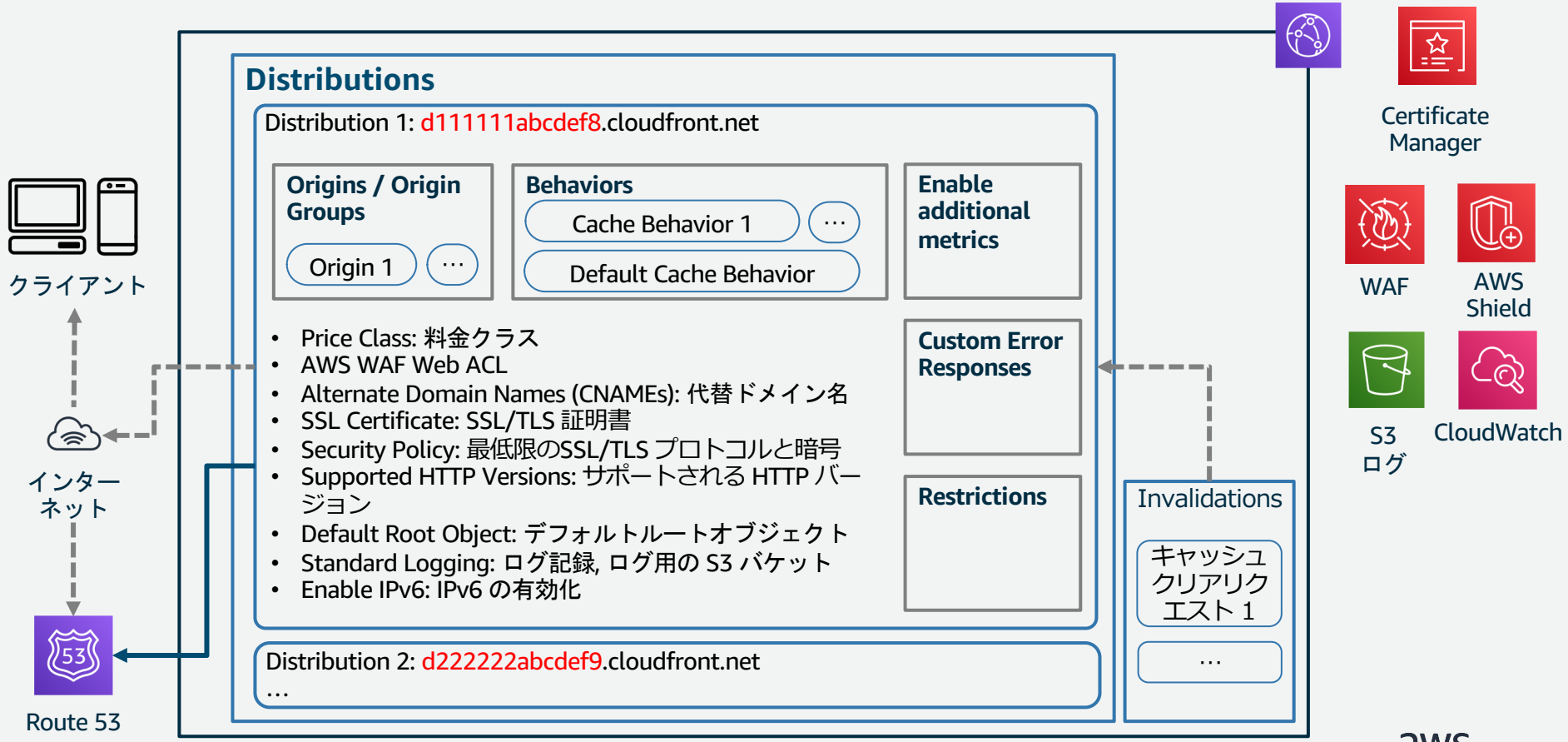
aws

CloudFront 設定

続き

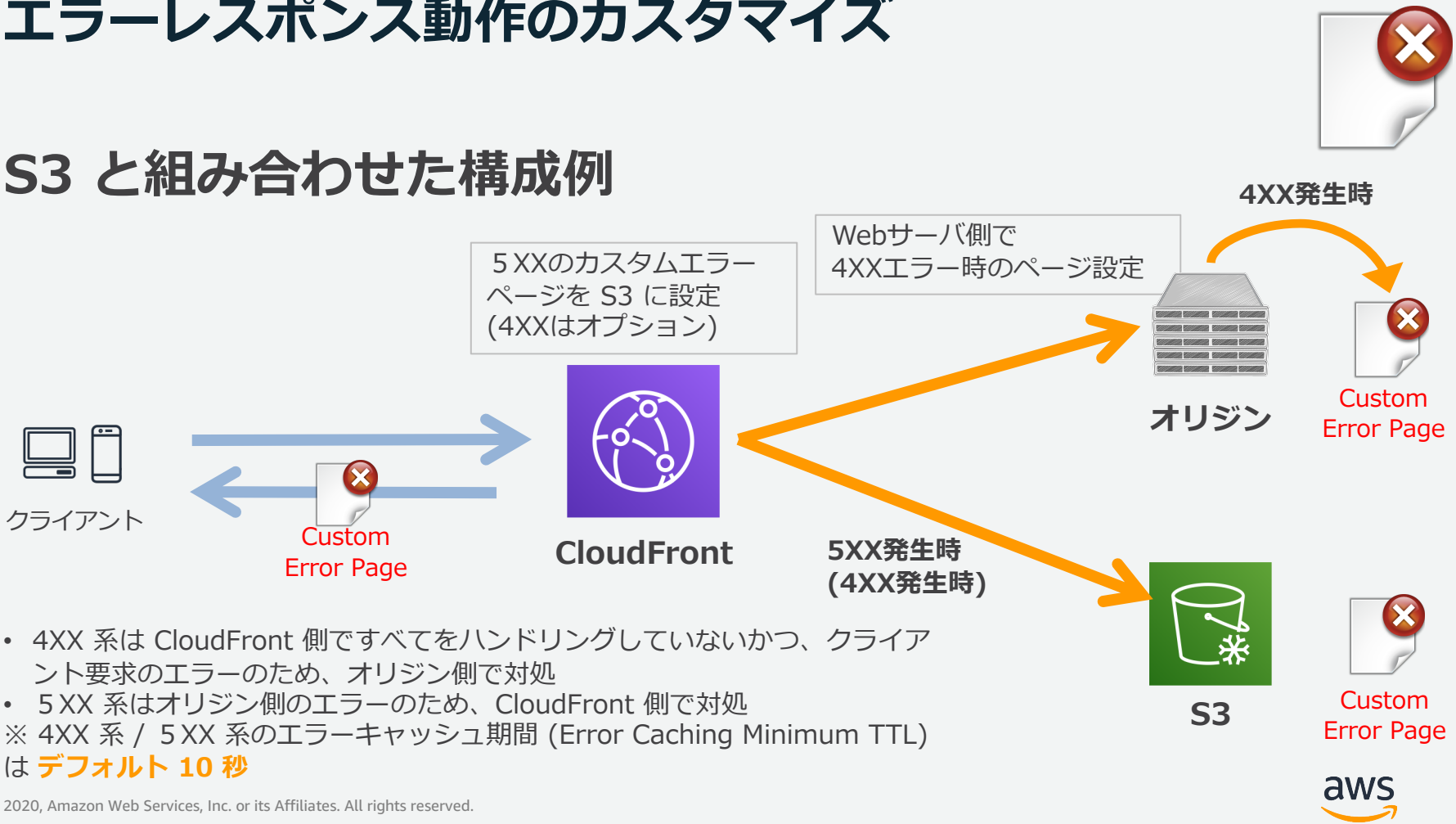
4. Distribution に関連する機能
    - Custom Error Responses: エラーレスポンス動作のカスタマイズ
    - Restrictions: 特定の国のユーザー
    - Invalidation: キャッシュファイルの無効化

Distribution 概要図



エラーレスポンス動作のカスタマイズ

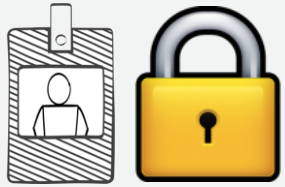
S3 と組み合わせた構成例



- 4XX 系は CloudFront 側ですべてをハンドリングしていないかつ、クライアント要求のエラーのため、オリジン側で対処
- ５XX 系はオリジン側のエラーのため、CloudFront 側で対処

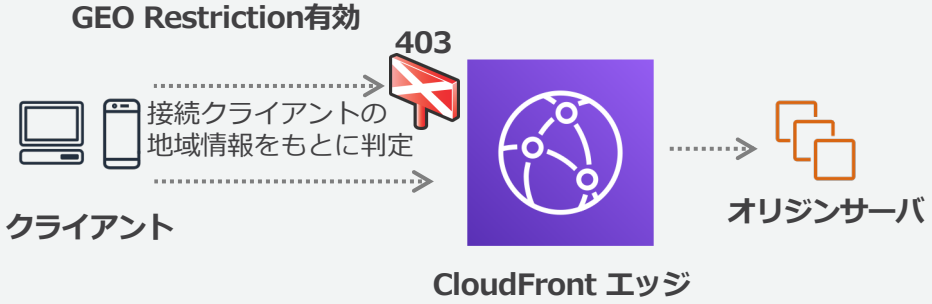
※ 4XX 系 / ５XX 系のエラーキャッシュ期間 (Error Caching Minimum TTL) は **デフォルト 10 秒**

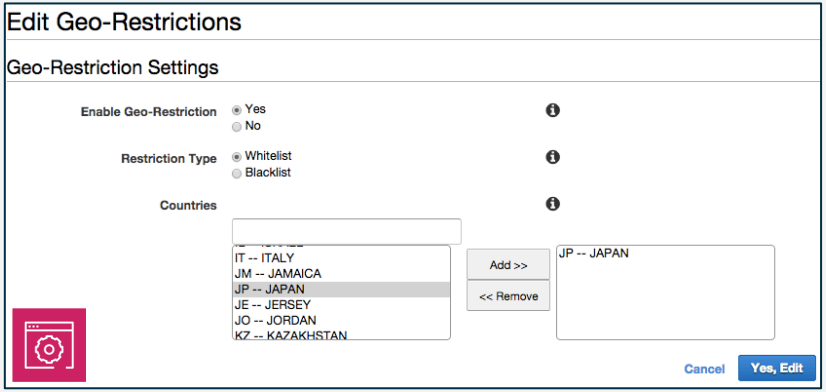
地域 (GEO) 制限



特定の国のユーザーに対するアクセス制御

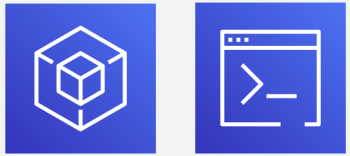
- 接続されるクライアントの地域情報を元に、エッジでアクセス判定
- 無効リストもしくは有効リストで指定可能
- Distribution 全体に対して適用される
- 制限されたアクセスには **403** を応答



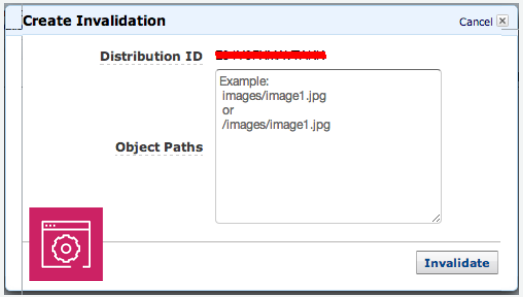


キャッシュファイルの無効化 (Invalidation)

- 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)

aws

Appendix

aws

As of 10/28/2020

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１０TB/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

⑤ CloudFront 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

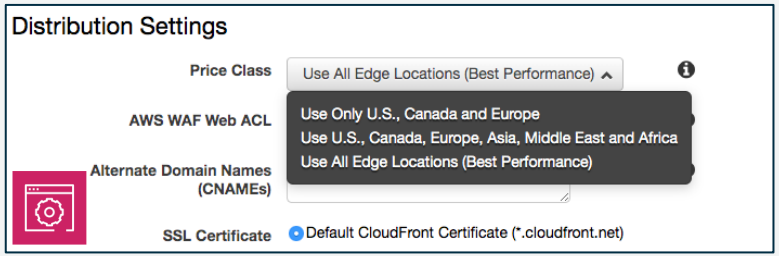
https://aws.amazon.com/jp/cloudfront/pricing/

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,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
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

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