



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]

## AWS X-Ray

サービスカットシリーズ

Solutions Architect 鈴木 太一郎  
2020/5/26

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



過去資料  
<https://amzn.to/JPArchive>



# 自己紹介

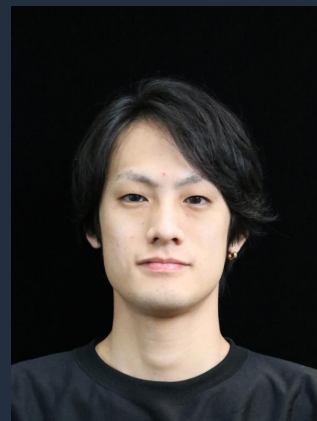
## 鈴木 太一郎

### 所属

- アマゾン ウェブ サービス ジャパン株式会社  
技術統括本部
- ソリューションアーキテクト

### 好きなサービス

- AWS X-Ray
- AWS CDK



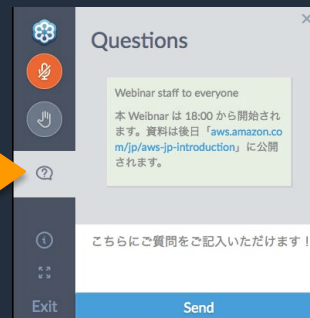
# AWS Black Belt Online Seminar とは

「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5月26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おります。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.

# アジェンダ

- AWS X-Ray とは
- AWS X-Ray の概念
- AWS X-Ray によるアプリケーション分析
- AWS X-Ray によるデータ収集の仕組み
- AWS X-Ray の利用方法
- Amazon CloudWatch との連携
- 料金・注意点
- まとめ

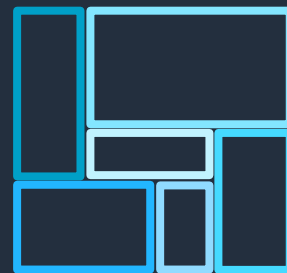
# アジェンダ

- **AWS X-Ray とは**
- AWS X-Ray の概念
- AWS X-Ray によるアプリケーション分析
- AWS X-Ray によるデータ収集の仕組み
- AWS X-Ray の利用方法
- Amazon CloudWatch との連携
- 料金・注意点
- まとめ

# モノリシック vs マイクロサービス

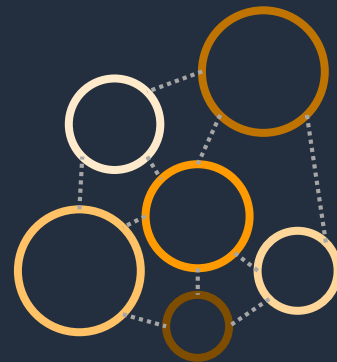
- **モノリシック アプリケーション**

- ✓ 多数の機能を1プロセスで稼働



- **マイクロサービス アプリケーション**

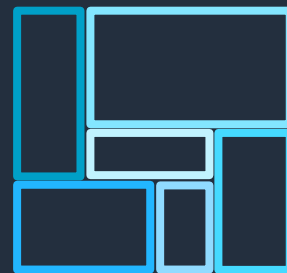
- ✓ 少数のまとまった機能を持つ複数のサービスで構成
- ✓ 各サービスが独立したプロセスとして稼働
- ✓ サーバレスやコンテナと相性が良い



# モノリシック vs マイクロサービス

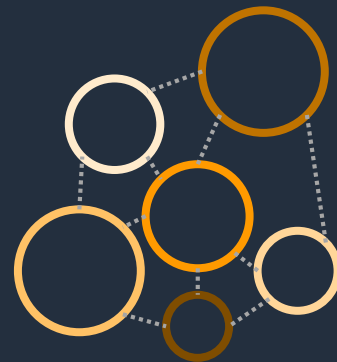
## • モノリシック アプリケーション

- ✓ 大規模アプリでは、高速なイテレーションが難しい
- ✓ 効率的なスケールが難しい
- ✓ 不具合や障害が全体に影響



## • マイクロサービス アプリケーション

- ✓ 各サービスで独立した開発・デプロイ
- ✓ 各サービスで独立したスケーリング
- ✓ 障害の直接的な影響が各サービス内





# マイクロサービスにおける障害対応・パフォーマンス分析

- **障害発生時の原因究明**
  - ログが分散・それぞれ独自のフォーマット
  - 依存しあう多数の多様なサービス群
- **パフォーマンス分析が難しい**
  - ボトルネックのサービスはどこなのか
  - 日々変化するサービス間の依存関係と全体像



# AWS X-Ray の概要

アプリケーションやその基盤となるサービスの実行状況を把握し  
パフォーマンスの問題やエラーの根本原因を特定する



## AWS X-Ray

AWS X-Ray helps you analyze and debug modern applications built using microservices and serverless architecture and quantify customer impact



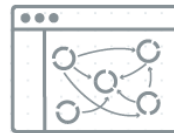
### Collect traces

Collect data about the request from each of the underlying application services it passes through



### Record traces

X-Ray combines the data gathered from each service into singular units called traces



### View service map

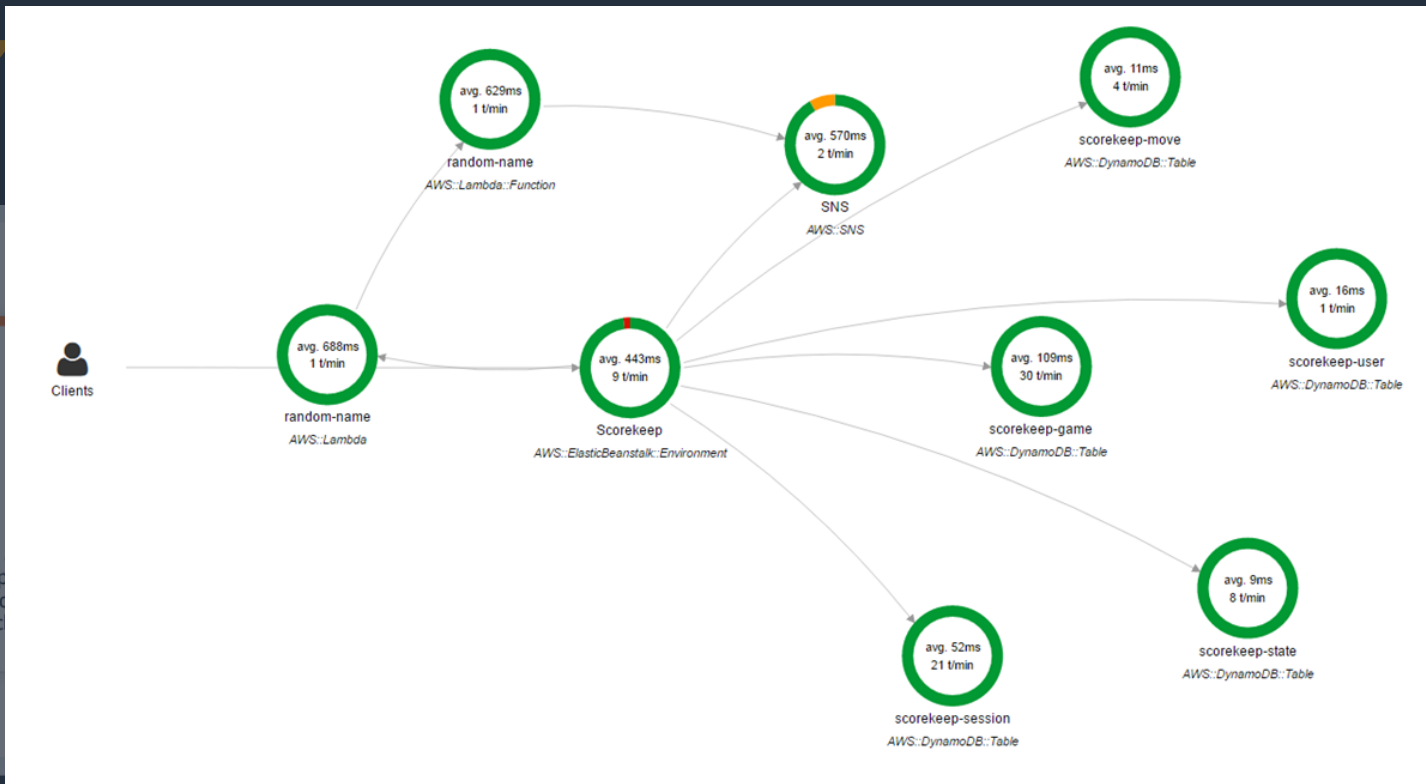
View the service map to see trace data such as latencies, HTTP statuses, and metadata for each service



### Analyze issues

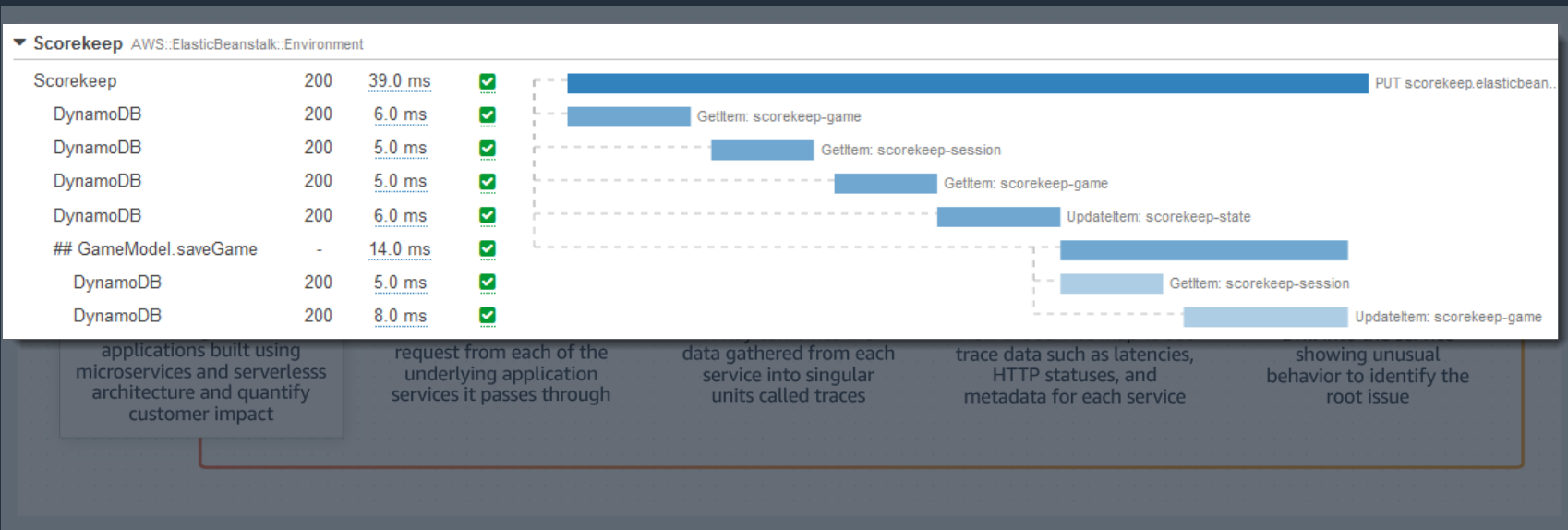
Drill into the service showing unusual behavior to identify the root issue

# AWS X-Ray の概要



# AWS X-Ray の概要

アプリケーションやその基盤となるサービスの実行状況を把握し  
パフォーマンスの問題やエラーの根本原因を特定する

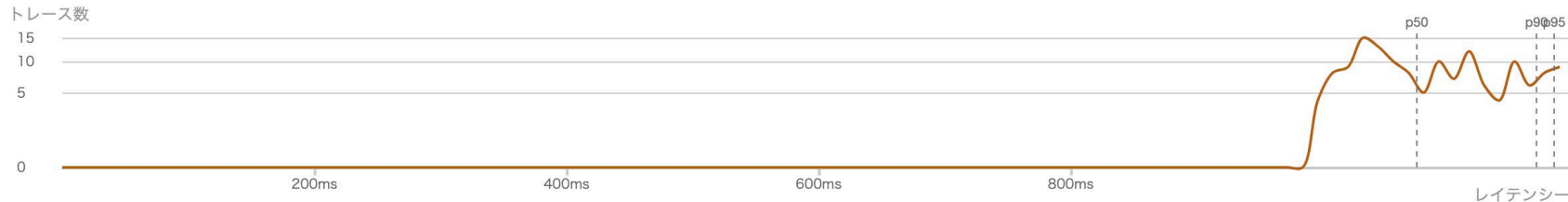


# AWS X-Ray の概要

## アプリケーションやその基盤となるサービスの実行状況を把握し

### 応答時間の分布

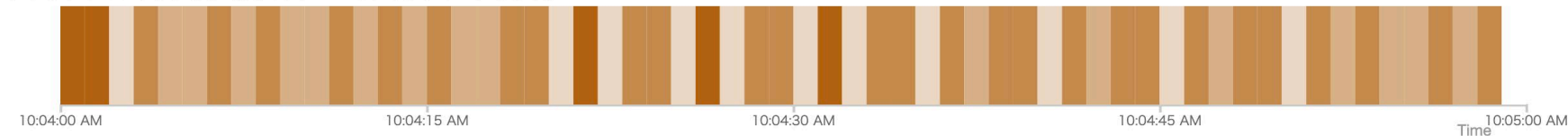
クリックしてドラッグし、応答時間でトレースをフィルタリングします。



応答時間の分布  期間分布

### 時系列アクティビティ

クリックしてドラッグし、時間でトレースをフィルタリングします。



# AWS X-Ray のメリット



## リクエスト実行状況の確認

アプリケーションを構成する個々のサービスやリソースの実行結果ステータスを集計し、アプリケーションの実行状況を End-to-End で確認可能



## アプリケーションの問題の検出

アプリケーションの実行状況についての関連する情報を収集し、問題の根本原因を調査可能



## アプリケーションのパフォーマンス向上

サービスやリソースの関係をリアルタイムで表示し、レイテンシ増加やパフォーマンス低下などのボトルネックを特定可能



## AWSとの連携

Amazon EC2, Amazon ECS, AWS Lambda, AWS Elastic Beanstalk などと連携



## さまざまなアプリケーション向けの設計

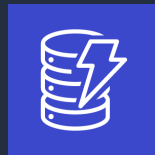
非同期のシンプルなイベント呼び出し、3層のウェブアプリケーション、数千のサービスから構成される複雑なマイクロサービスも分析可能

# アジェンダ

- AWS X-Ray とは
- **AWS X-Ray の概念**
- AWS X-Ray によるアプリケーション分析
- AWS X-Ray によるデータ収集の仕組み
- AWS X-Ray の利用方法
- Amazon CloudWatch との連携
- 料金・注意点
- まとめ

# AWS X-Ray の概念

Amazon DynamoDB



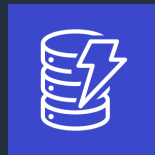
AWS Lambda

AWS Lambda が Amazon DynamoDB を3回呼び出し



# AWS X-Ray の概念

Amazon DynamoDB



AWS Lambda

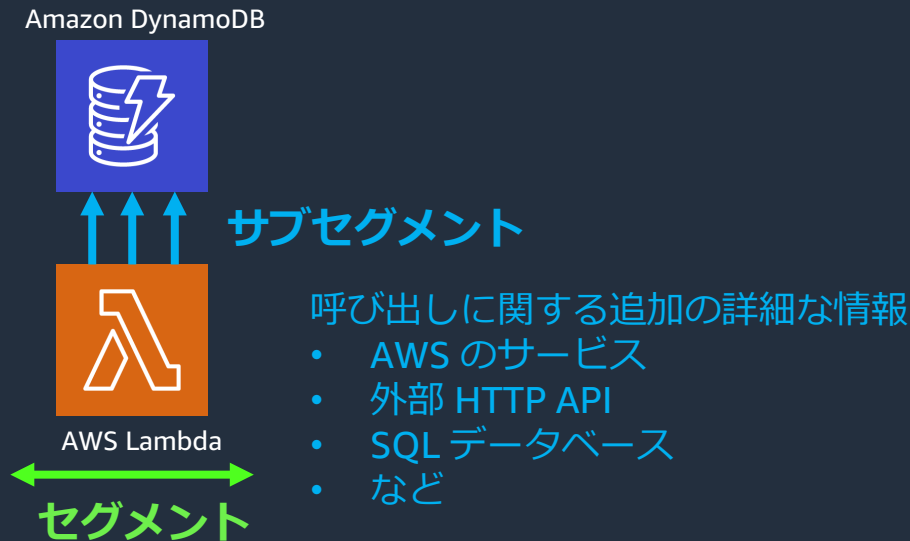


セグメント

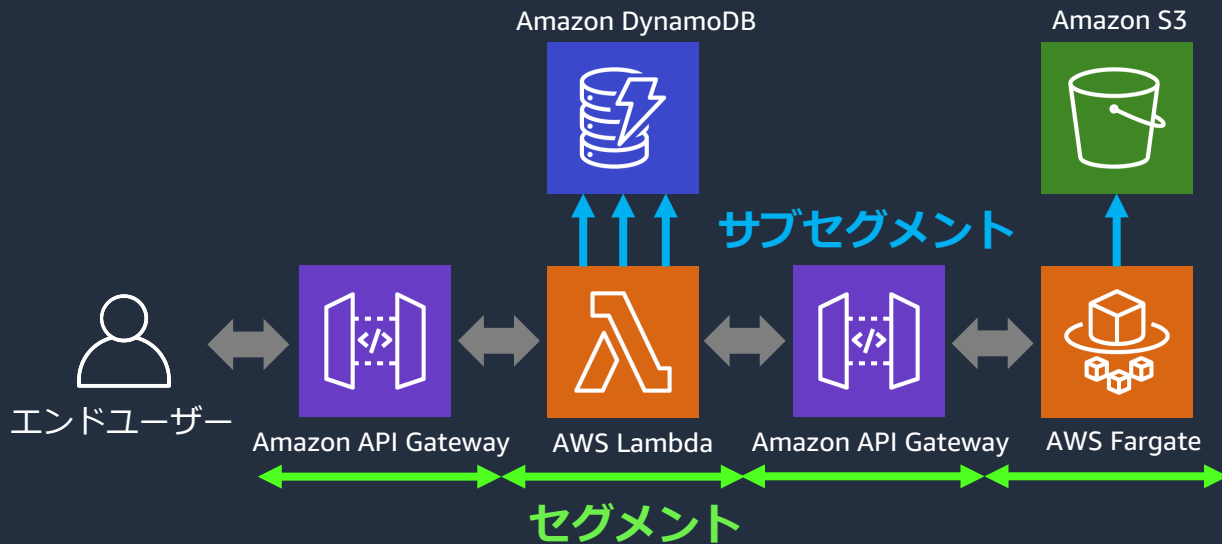
動作に関するデータ

- リソース名
- リクエストの詳細
- 行った作業の詳細

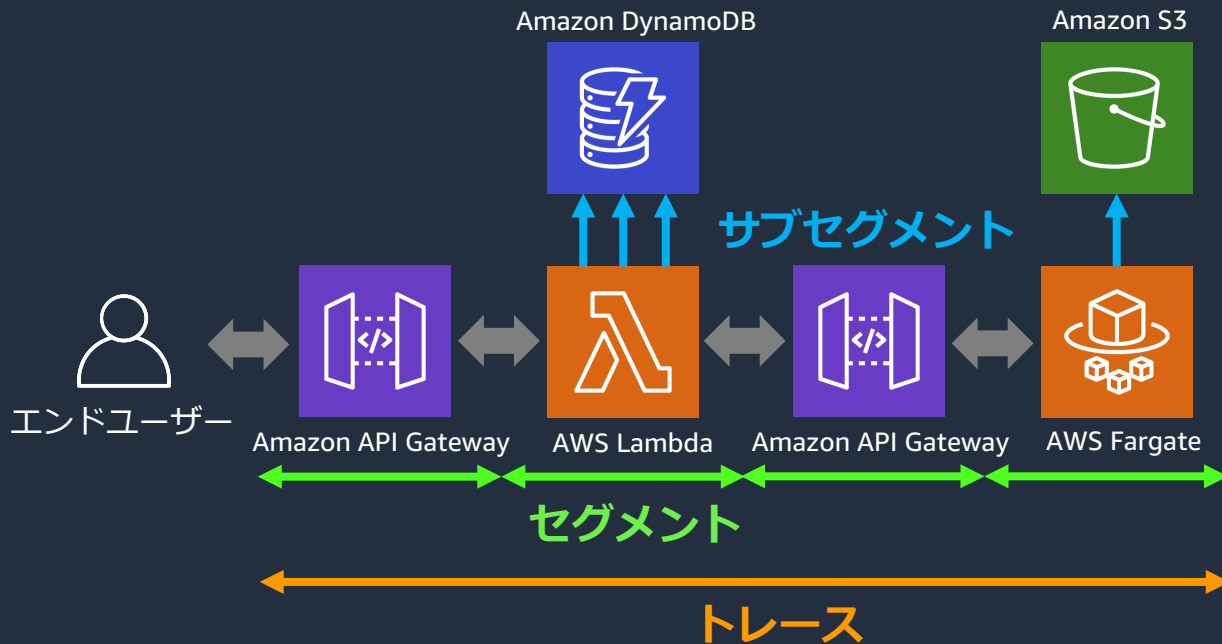
# AWS X-Ray の概念



# AWS X-Ray の概念



# AWS X-Ray の概念



1つのリクエストで生成されたセグメントの全てを収集したもの

# AWS X-Ray の概念

- どのようにトレースが含むセグメントを管理している？
- セグメントのつながり・順序はどう管理している？

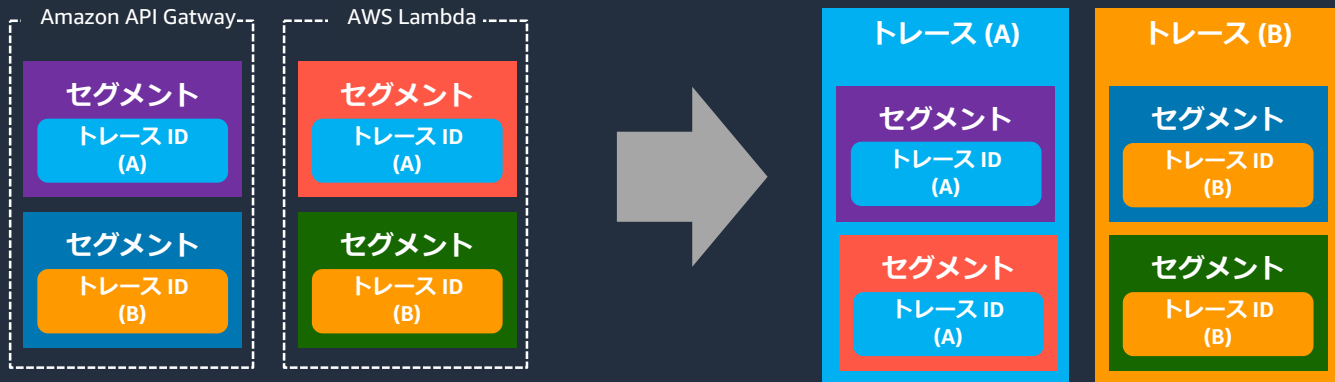


# AWS X-Ray の概念

どのようにトレースが含むセグメントを管理している？

- **トレース ID**

- リクエストを追跡するための ID
- リクエストごとに一意
- 基本的に最初にセグメントを生成するタイミングで生成

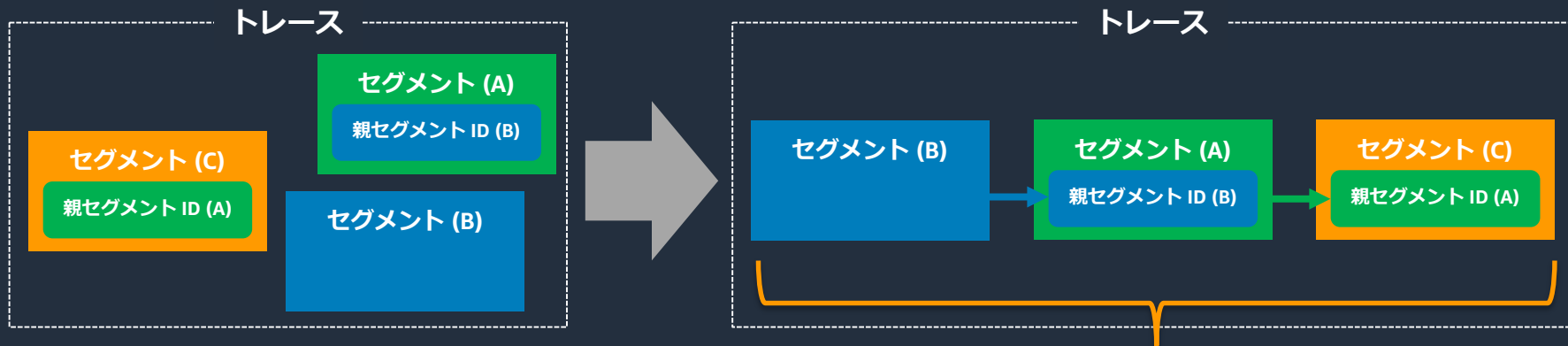


# AWS X-Ray の概念

セグメントのつながり・順序はどう管理している？

- **セグメント ID**

- トレース内のセグメントに対して一意に決まる
- 単一のサービスのリクエストを追跡するための ID
- **親セグメント ID** によってセグメントのつながり・順序が決まる



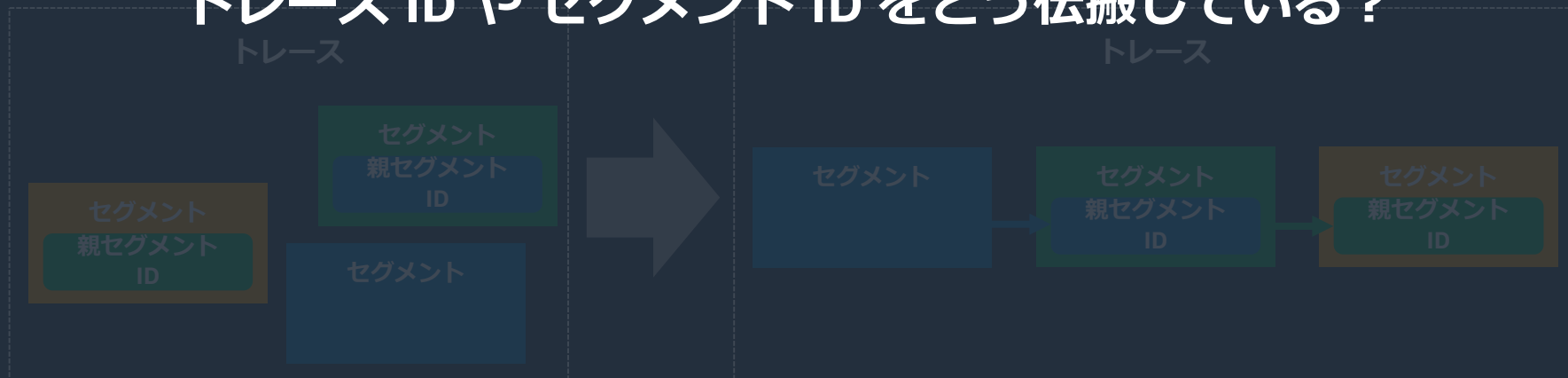
# AWS X-Ray の概念

セグメント同士のつながりや順序はどう管理している？

## • セグメント ID

- 単一のサービスのリクエストを追跡するためのID
- 親セグメント ID によってセグメントのつながりが決まる

トレース ID や セグメント ID をどう伝搬している？





# AWS X-Ray の概念

トレース ID や セグメント ID をどう伝搬している？

- **トレースヘッダー**

- トレース ID、親セグメントID、サンプリングデシジョンが定義された HTTP ヘッダー

```
X-Amzn-Trace-Id: Root=1-57...93;Parent=53...d8;Sampled=1
```

トレース ID

親セグメント ID

サンプルデシジョン サンプリング対象か否か (1=true, 0=false)

# AWS X-Ray の概念

## セグメントに情報を追加したい

検索しやすくするため

- **注釈**

- フィルタ式 (後述) で使用するためインデックス化されたキーと値のペア
- 例 `"aws:api_stage": "prod"`

デバッグや分析のため

- **メタデータ**

- 任意のタイプの値を持つことができるキーと値のペア
- フィルタ式で使用するためのインデックスは作成されない
- 例 `"game": {"id": "abcdefg", "session": "xxxyyyzzz"}`

# アジェンダ

- AWS X-Ray とは
- AWS X-Ray の概念
- **AWS X-Ray によるアプリケーション分析**
- AWS X-Ray によるデータ収集の仕組み
- AWS X-Ray の利用方法
- Amazon CloudWatch との連携
- 料金・注意点
- まとめ

# AWS X-Ray によるアプリケーション分析



End-to-End のリクエストの確認



**サービスマップ**  
トレースデータの可視化

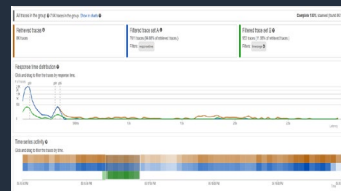
特定のサービスに関連する  
トレースリストの確認



**トレースリスト**  
トレースの詳細な情報



特定のサービスに関連する  
トレースリストを分析



**アナリティクス**  
トレース分析用ツール



分析対象のトレースリストの  
詳細を確認



# サービスマップ

## AWS X-Ray

はじめに

## Service Map

Traces

アナリティクス

☐ 設定

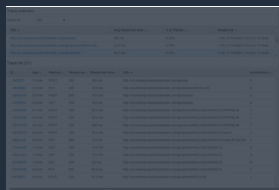
サンプリング

暗号化

- 読み取れる情報
- フィルタ式
- グループ



## サービスマップ

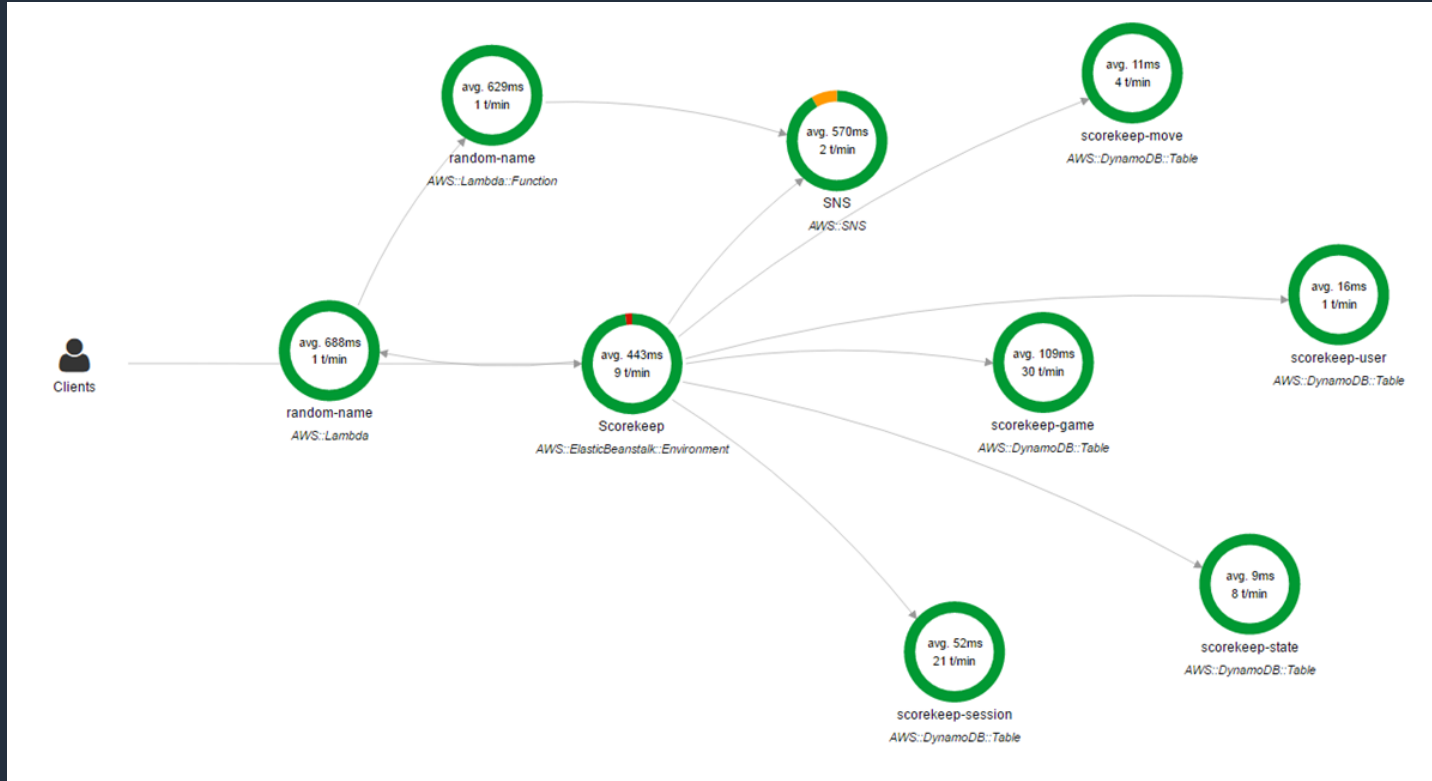


トレースリスト

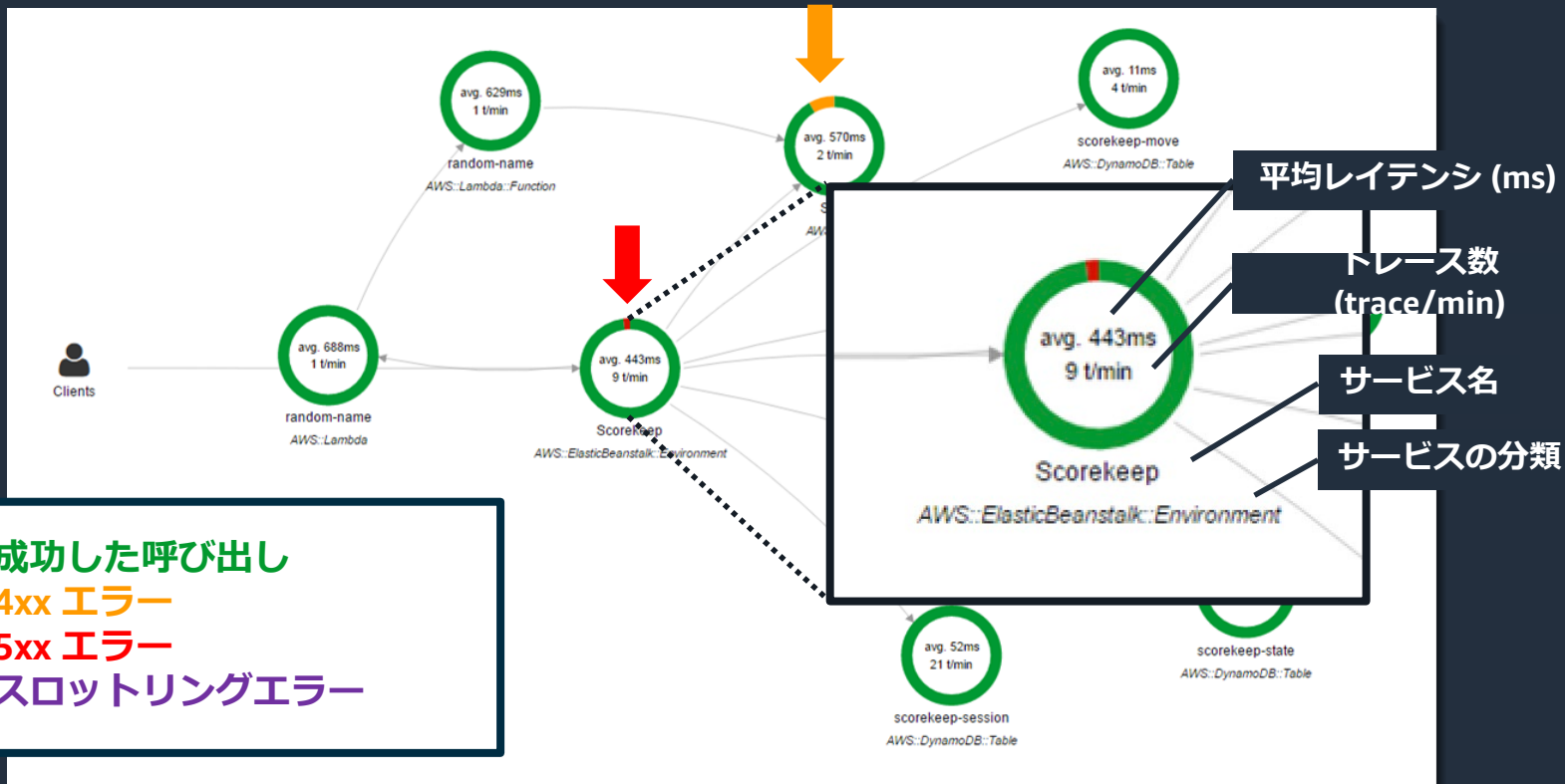


アナリティクス

# サービスマップ

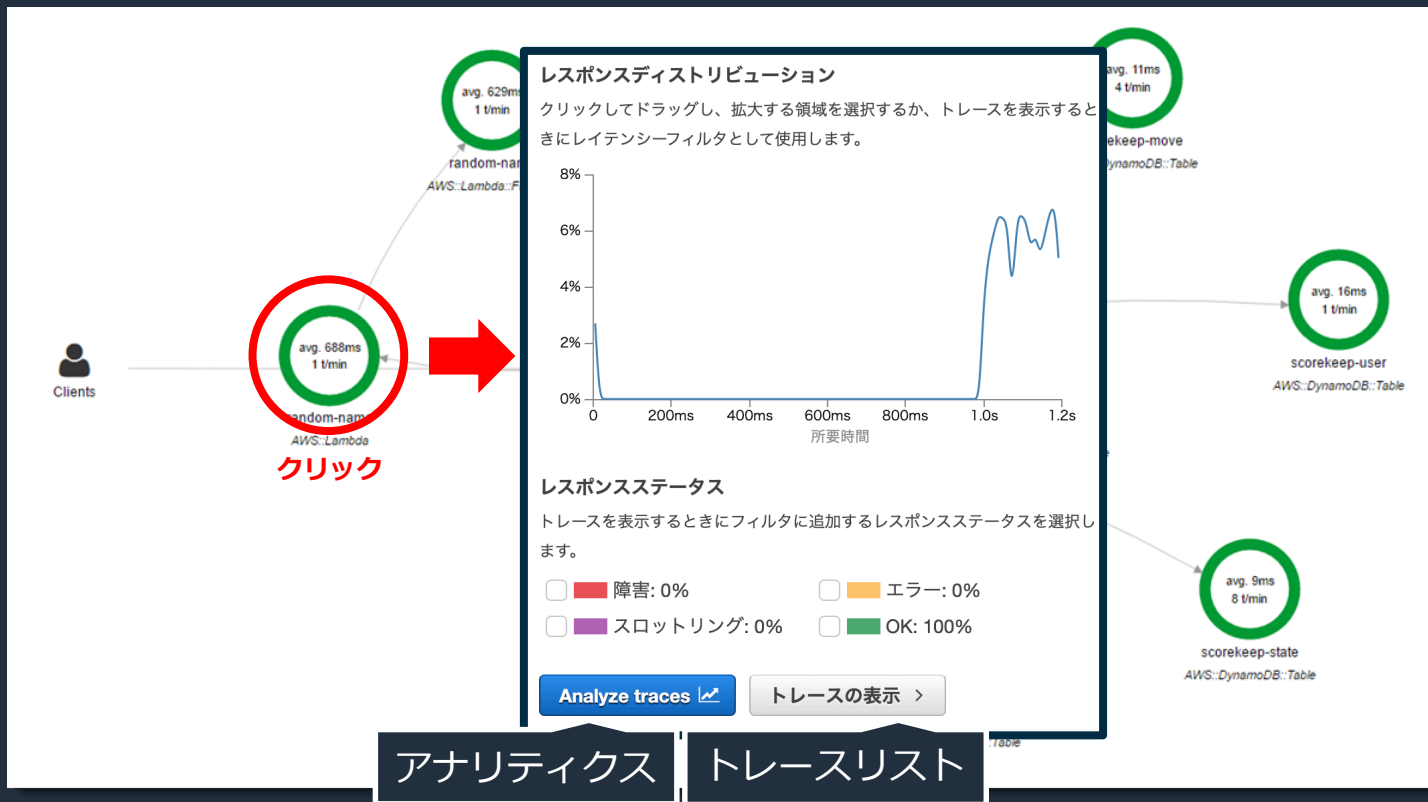


# サービスマップ



- 成功した呼び出し
- 4xx エラー
- 5xx エラー
- スロットリングエラー

# サービスマップ





# フィルタ式

デフォルト ▾

🔍 サービス名、注釈、トレースID を入力します。または、[ヘルプ] アイコンをクリックして詳細を確認します。

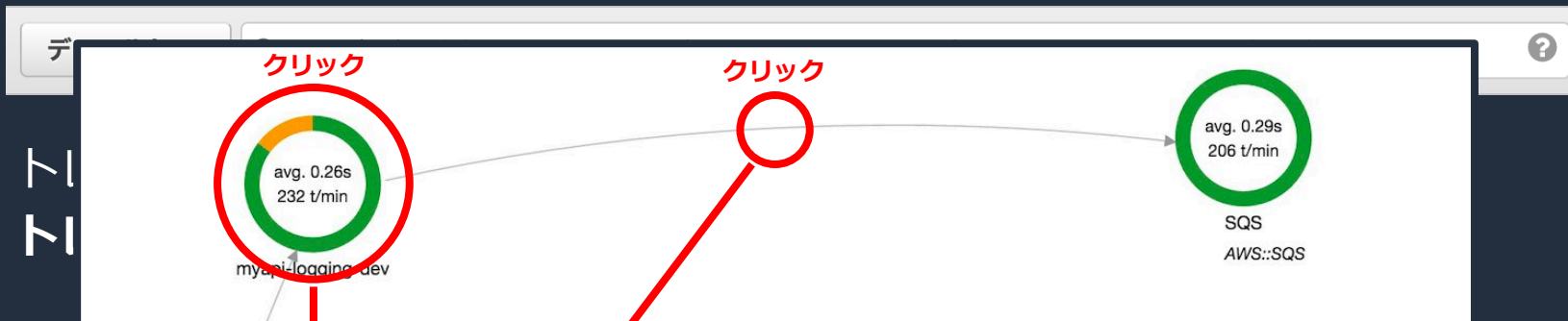


- トレースを絞り込むための機能
- **トレースリストとアナリティクス**で利用可能
  - `duration >= 5 AND duration <= 8` 総所要時間が5秒～8秒
  - `http.status != 200` レスポンスステータスが 200 OK ではない
  - `service(name) { filter }` 特定のサービスを含んだリクエスト
  - `edge(source, destination) { filter }` 特定のサービス間の接続

[https://docs.aws.amazon.com/ja\\_jp/xray/latest/devguide/xray-console-filters.html](https://docs.aws.amazon.com/ja_jp/xray/latest/devguide/xray-console-filters.html)

# フィルタ式

- 
- 



- `http.status != 200` レスポンスステータスが 200 OK ではない
- `service(name) { filter }` 特定のサービスを含んだリクエスト
- `edge(source, destination) { filter }` 特定のサービス間の接続

[https://docs.aws.amazon.com/ja\\_jp/xray/latest/devguide/xray-console-filters.html](https://docs.aws.amazon.com/ja_jp/xray/latest/devguide/xray-console-filters.html)

# グループ

デフォルト ▾

名、注釈、トレースIDを入力します。または、[ヘルプ] アイコンをクリックして詳細を確認します。



- グループ
  - フィルタ式で定義されるトレースのコレクション
  - サービスマップ、トレースリスト、アナリティクスで利用可能
  - Amazon CloudWatch にメトリクスを作成 (トレース数)

名前

high-latency

フィルタ式

```
service("api.example.com") { fault = true OR responsetime > 5 }
```

[https://docs.aws.amazon.com/ja\\_jp/xray/latest/devguide/xray-console-filters.html](https://docs.aws.amazon.com/ja_jp/xray/latest/devguide/xray-console-filters.html)

# トレースリスト

## AWS X-Ray

はじめに

Service Map

Traces

アナリティクス

設定

サンプリング

暗号化

- トレースリスト
- セグメント・サブセグメント
- カスタムサブセグメント
- セグメント詳細



サービスマップ

Method	Response	Response time	URL	Annotations
GET	200	391 ms	http://scorekeep-elasticbeanstalk.com/api/user	0
POST	200	33.0 ms	http://scorekeep-elasticbeanstalk.com/api/session/8N63LUQ6	0
POST	200	19.0 ms	http://scorekeep-elasticbeanstalk.com/api/session	0
POST	200	19.0 ms	http://scorekeep-elasticbeanstalk.com/api/session	0
POST	200	19.0 ms	http://scorekeep-elasticbeanstalk.com/api/session	0
POST	200	19.0 ms	http://scorekeep-elasticbeanstalk.com/api/move/8N63LUQ6/2N56AC7L/PPMPBLIB	1
POST	200	19.0 ms	http://scorekeep-elasticbeanstalk.com/api/move/8N63LUQ6/2N56AC7L/PPMPBLIB	1
POST	200	19.0 ms	http://scorekeep-elasticbeanstalk.com/api/move/8N63LUQ6/2N56AC7L/PPMPBLIB	1
POST	200	19.0 ms	http://scorekeep-elasticbeanstalk.com/api/game/8N63LUQ6/2N56AC7L/users	1
POST	200	19.0 ms	http://scorekeep-elasticbeanstalk.com/api/game/8N63LUQ6/2N56AC7L/rules/TicTacToe	1
POST	200	19.0 ms	http://scorekeep-elasticbeanstalk.com/api/game/8N63LUQ6/2N56AC7L	0
POST	200	19.0 ms	http://scorekeep-elasticbeanstalk.com/api/game/8N63LUQ6/2N56AC7L	0
POST	200	19.0 ms	http://scorekeep-elasticbeanstalk.com/api/game/8N63LUQ6/2N56AC7L	0
POST	200	19.0 ms	http://scorekeep-elasticbeanstalk.com/api/game/8N63LUQ6/2N56AC7L	1
POST	200	19.0 ms	http://scorekeep-elasticbeanstalk.com/api/game/8N63LUQ6/2N56AC7L	1

トレースリスト



アナリティクス

# トレースリスト

URL

トレースリスト

Trace overview

Group by:

URL	Avg response time	% of Traces	Response
http://scorekeep.elasticbeanstalk.com/api/user	391 ms	4.76%	1 OK, 0 Throttled, 0 Errors, 0 Faults
http://scorekeep.elasticbeanstalk.com/api/session/8N63LUQ6	33.0 ms	4.76%	1 OK, 0 Throttled, 0 Errors, 0 Faults
http://scorekeep.elasticbeanstalk.com/api/session	90.5 ms	9.52%	2 OK, 0 Throttled, 0 Errors, 0 Faults

Trace list (21)

ID	Age	Method	Response	Response time	URL	Annotations
...f5f2df73	5.0 min	POST	200	391 ms	http://scorekeep.elasticbeanstalk.com/api/user	0
...cfe39980	5.0 min	PUT	200	33.0 ms	http://scorekeep.elasticbeanstalk.com/api/session/8N63LUQ6	0
...dd653e4c	5.0 min	POST	200	19.0 ms	http://scorekeep.elasticbeanstalk.com/api/session	0
...4765fec8	5.0 min	GET	200	162 ms	http://scorekeep.elasticbeanstalk.com/api/session	0
...84eeef29	4.7 min	POST	200	95.0 ms	http://scorekeep.elasticbeanstalk.com/api/move/8N63LUQ6/2N56AC7L/PPMPBLJB	1
...3ab33fdb	4.8 min	POST	200	95.0 ms	http://scorekeep.elasticbeanstalk.com/api/move/8N63LUQ6/2N56AC7L/PPMPBLJB	1
...237e0705	4.8 min	POST	200	295 ms	http://scorekeep.elasticbeanstalk.com/api/move/8N63LUQ6/2N56AC7L/PPMPBLJB	1
...86782227	4.9 min	POST	200	25.0 ms	http://scorekeep.elasticbeanstalk.com/api/game/8N63LUQ6/2N56AC7L/users	1
...fd82cc32	4.9 min	PUT	200	121 ms	http://scorekeep.elasticbeanstalk.com/api/game/8N63LUQ6/2N56AC7L/rules/Tic Tac Toe	1
...7ca2e05f	1.4 min	GET	200	14.0 ms	http://scorekeep.elasticbeanstalk.com/api/game/8N63LUQ6/2N56AC7L	0
...062ccac5	1.7 min	GET	200	12.0 ms	http://scorekeep.elasticbeanstalk.com/api/game/8N63LUQ6/2N56AC7L	0
...dc0ebe3c	1.9 min	GET	200	9.0 ms	http://scorekeep.elasticbeanstalk.com/api/game/8N63LUQ6/2N56AC7L	0
...524637dc	4.9 min	PUT	200	69.0 ms	http://scorekeep.elasticbeanstalk.com/api/game/8N63LUQ6/2N56AC7L	1
...fdf5bb67	4.9 min	POST	200	81.0 ms	http://scorekeep.elasticbeanstalk.com/api/game/8N63LUQ6	1

# トレースリスト

Trace overview

Group by:

URL	Avg response time	% of Traces	Response
http://scorekeep.elasticbeanstalk.com/api/user	391 ms	4.76%	1 OK, 0 Throttled, 0 Errors, 0 Faults
http://scorekeep.elasticbeanstalk.com/api/session/8N63LUQ6	33.0 ms	4.76%	1 OK, 0 Throttled, 0 Errors, 0 Faults
http://scorekeep.elasticbeanstalk.com/api/session	90.5 ms	9.52%	2 OK, 0 Throttled, 0 Errors, 0 Faults

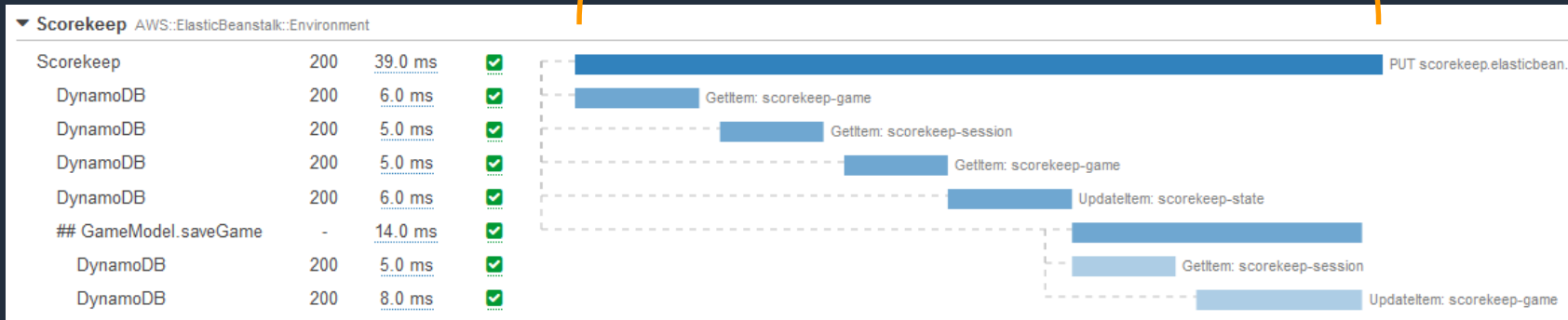
Trace list (21)

ID	Age	Method	Response	Response time	URL	Annotations
...f5f2df73	0 min	POST	200	391 ms	http://scorekeep.elasticbeanstalk.com/api/user	0
...cfe39980	5.0 min	PUT	200	33.0 ms	http://scorekeep.elasticbeanstalk.com/api/session/8N63LUQ6	0
...035564c	5.0 min	POST	200	19.0 ms	http://scorekeep.elasticbeanstalk.com/api/session	0
...4765fec8	5.0 min	GET	200	162 ms	http://scorekeep.elasticbeanstalk.com/api/session	0
...84eeef29	4.7 min	POST	200	95.0 ms	http://scorekeep.elasticbeanstalk.com/api/move/8N63LUQ6/2N56AC7L/PPMPBLJB	1
...3ab33fdb	4.8 min	POST	200	95.0 ms	http://scorekeep.elasticbeanstalk.com/api/move/8N63LUQ6/2N56AC7L/PPMPBLJB	1
...237e0705	4.8 min	POST	200	295 ms	http://scorekeep.elasticbeanstalk.com/api/move/8N63LUQ6/2N56AC7L/PPMPBLJB	1
...86782227	4.9 min	POST	200	25.0 ms	http://scorekeep.elasticbeanstalk.com/api/game/8N63LUQ6/2N56AC7L/users	1
...fd82cc32	4.9 min	PUT	200	121 ms	http://scorekeep.elasticbeanstalk.com/api/game/8N63LUQ6/2N56AC7L/rules/Tic Tac Toe	1
...7ca2e05f	1.4 min	GET	200	14.0 ms	http://scorekeep.elasticbeanstalk.com/api/game/8N63LUQ6/2N56AC7L	0
...062ccac5	1.7 min	GET	200	12.0 ms	http://scorekeep.elasticbeanstalk.com/api/game/8N63LUQ6/2N56AC7L	0
...dc0ebe3c	1.9 min	GET	200	9.0 ms	http://scorekeep.elasticbeanstalk.com/api/game/8N63LUQ6/2N56AC7L	0
...524637dc	4.9 min	PUT	200	69.0 ms	http://scorekeep.elasticbeanstalk.com/api/game/8N63LUQ6/2N56AC7L	1
...fdf5bb67	4.9 min	POST	200	81.0 ms	http://scorekeep.elasticbeanstalk.com/api/game/8N63LUQ6	1

クリック

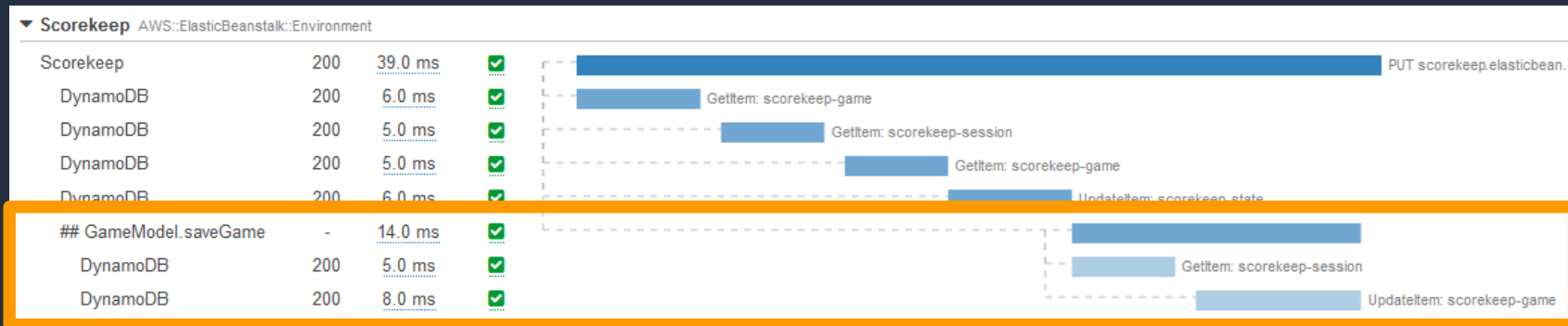
# セグメント・サブセグメント

セグメント



サブセグメント

# カスタムサブセグメント

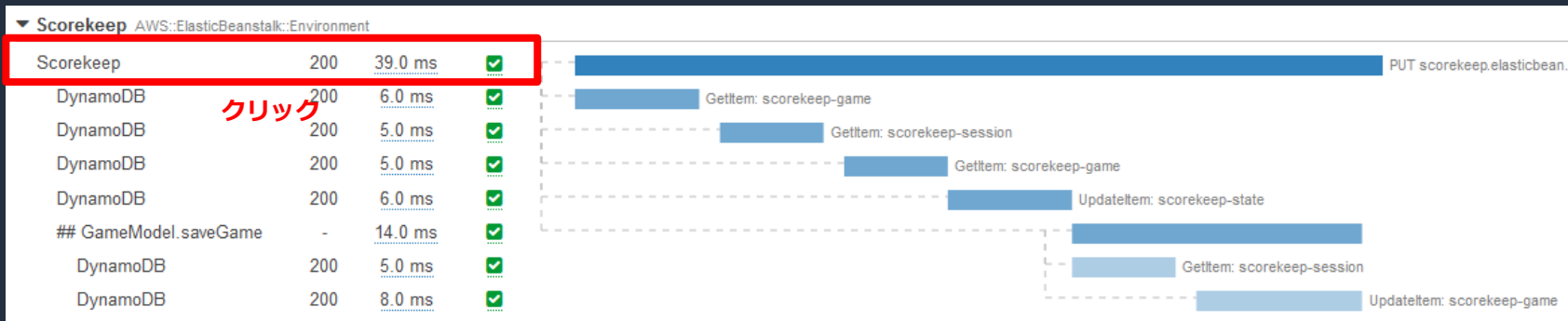


## カスタムサブセグメント

監視・計測したい機能ごとにサブセグメントとして切り出すことが可能



# セグメント詳細



# セグメント詳細

## Segment - Scorekeep

Overview Resources Annotations Metadata Exceptions

Segment ID	69e25bb5bfe4c13f
Parent ID	
Name	Scorekeep
Origin	AWS::ElasticBeanstalk::Environment

**Time**

Start time	2017-03-14 17:10:56.769 (UTC)
End time	2017-03-14 17:10:56.814 (UTC)
Duration	45.0 ms
In progress	False

**Errors & Faults**

Error	False
Fault	False

**Request & Response**

Request method	PUT
Request url	http://scorekeep-dev.pfj8pcpmww.us-west-2.elasticbeanstalk.com/api/game/1C6KM1DN/M0TTEU23/rules/TicTacToe
Request user_agent	Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Safari/537.36
Request x_forwarded_for	
Response status	200

Close

# セグメント詳細

概要 keep 注釈 例外

Overview Resources Annotations Metadata Exceptions

リソース メタデータ

Origin AWS::ElasticBeanstalk::Environment

**Time**

Start time	2017-03-14 17:10:56.769 (UTC)
End time	2017-03-14 17:10:56.814 (UTC)
Duration	45.0 ms
In progress	False

**Errors & Faults**

Error	False
Fault	False

**Request & Response**

Request method	PUT
Request url	http://scorekeep-dev.pfj8pcpmww.us-west-2.elasticbeanstalk.com/api/game/1C6KM1DN/M0TTEU23/rules/TicTacToe
Request user_agent	Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Safari/537.36
Request x_forwarded_for	
Response status	200

Close

セグメント情報

時間情報

エラー情報

リクエストレスポンス詳細

# アナリティクス

## AWS X-Ray

はじめに

Service Map

Traces

## アナリティクス

設定

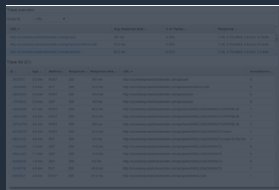
サンプリング

暗号化

- フィルタリングされたトレースセット
- 応答時間の分布
- 時系列アクティビティ
- メトリクステーブル
- ワークフローの例



## サービスマップ

A screenshot of the AWS X-Ray console showing a list of traces. The table has columns for Trace ID, Service Name, and other details. The traces are listed in a grid format, with some rows highlighted in blue.

## トレースリスト



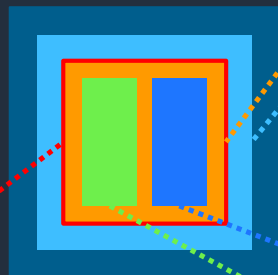
## アナリティクス

# フィルタリングされたトレースセット

グラフに対してドラッグ テーブルに対して選択

- グラフやテーブルに対して操作を行いアナリティクス内に作成
- 最大で2つ (A と B) まで作成可能
- トレースセットを活用することで素早い分析が可能

アナリティクスで可視化されるトレース

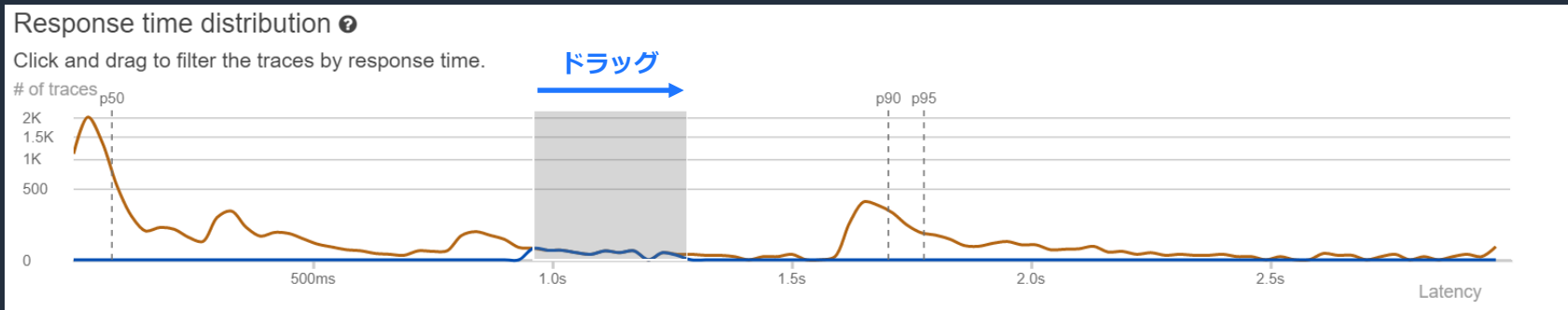


フィルタリングの結果  
グループ  
全てのトレース

フィルタリングされたトレースセット A  
フィルタリングされたトレースセット B

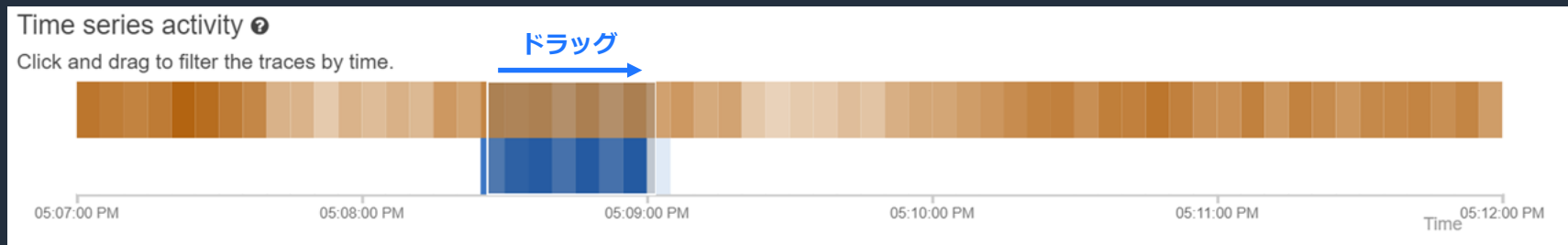
# 応答時間の分布

- 指定された応答時間におけるトレース数を示すグラフ
  - 横軸 応答時間
  - 縦軸 トレース数
- ドラッグすることで **フィルタリングされたトレースセット** を作成できる



# 時系列アクティビティ

- 指定された時間のトレース数
  - 横軸 時間
  - 濃さ トレース数 (濃い方が多い)
- ドラッグすることで **フィルタリングされたトレースセット** を作成できる



# メトリクステーブル

HTTP METHOD	COUNT	%
GET	713	98.89%
POST	5	0.69%
-	3	0.42%

HTTP STATUS CODE	COUNT	%
200	719	99.72%
500	2	0.28%

- Http Status Code や Response Time Root Cause など事前に定義された属性に基づき分類されたトレース
- テーブルの行をクリックすることでフィルタリングされたトレースセットとして設定可能



# メトリクステーブル

クリック

The screenshot shows two tables in the AWS CloudWatch Metrics console. The left table, titled 'HTTP METHOD', has a settings gear icon circled in red with the word 'クリック' (Click) above it. The right table, titled 'HTTP STATUS CODE', shows the following data:

HTTP STATUS CODE	COUNT	%
200	719	99.72%
500	2	0.28%

- テーブルの属性は変更可能
- 注釈も指定できる

# アナリティクスを用いたワークフローの例

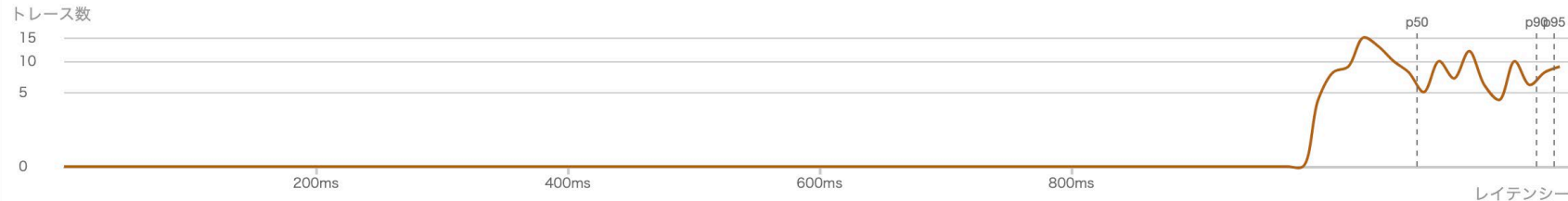
1. リリースが応答時間に影響を与えていないか確認
  - リリースした機能によってパフォーマンスが悪化した場合
  - パフォーマンス改善のリリースが効果的か調べる場合
  
2. 応答時間と時間帯に相関があるか調べる
  - サービスの負荷が高まる時間帯に応答時間が長くなっていないか調べる場合

# ワークフローの例 ①

リリースが応答時間に影響を与えていないか確認したい

## 応答時間の分布

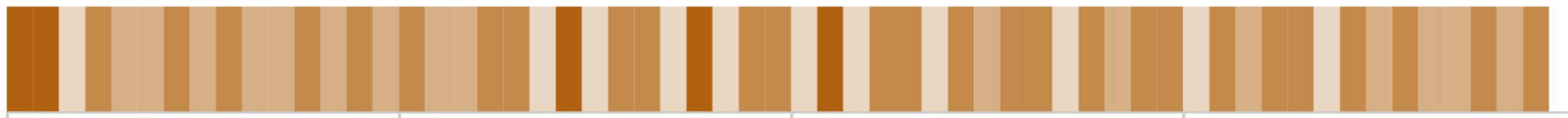
クリックしてドラッグし、応答時間でトレースをフィルタリングします。



応答時間の分布  期間分布

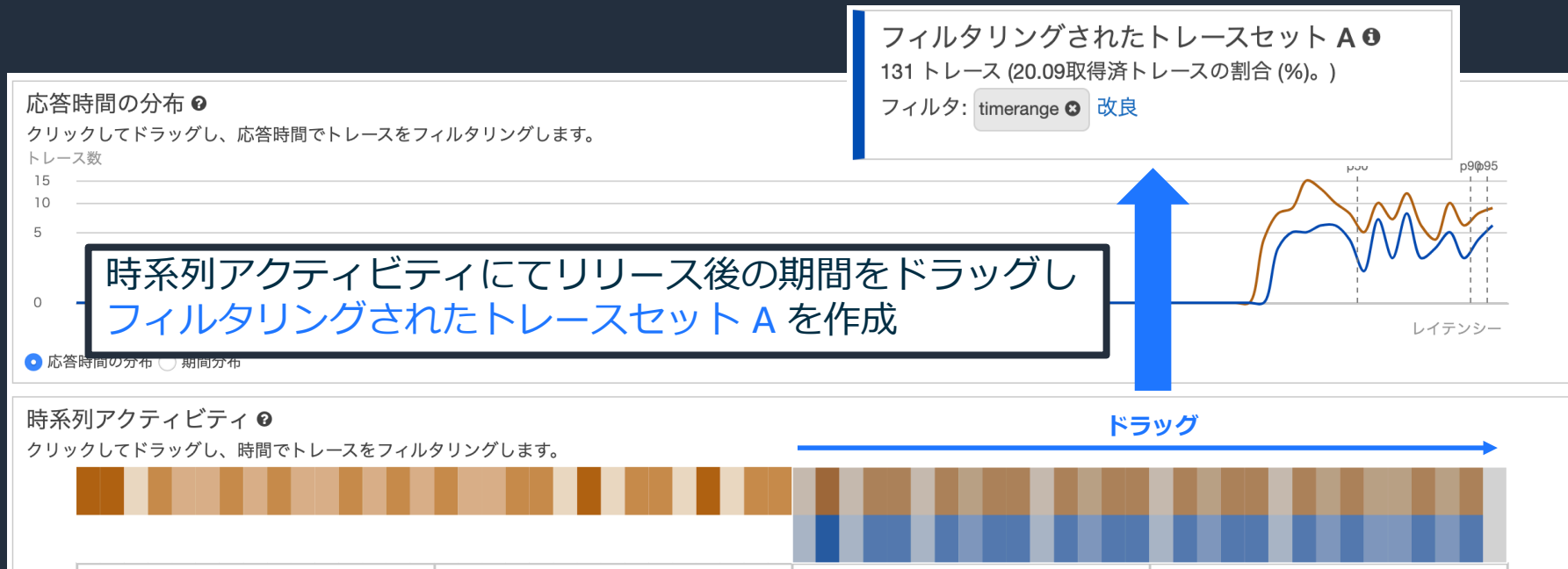
## 時系列アクティビティ

クリックしてドラッグし、時間でトレースをフィルタリングします。



# ワークフローの例 ①

リリースが応答時間に影響を与えていないか確認したい



リリースしたタイミング

# ワークフローの例 ①

リリースが応答時間に影響を与えていないか確認したい



応答時間の分布を確認し応答時間に影響がないことを確認

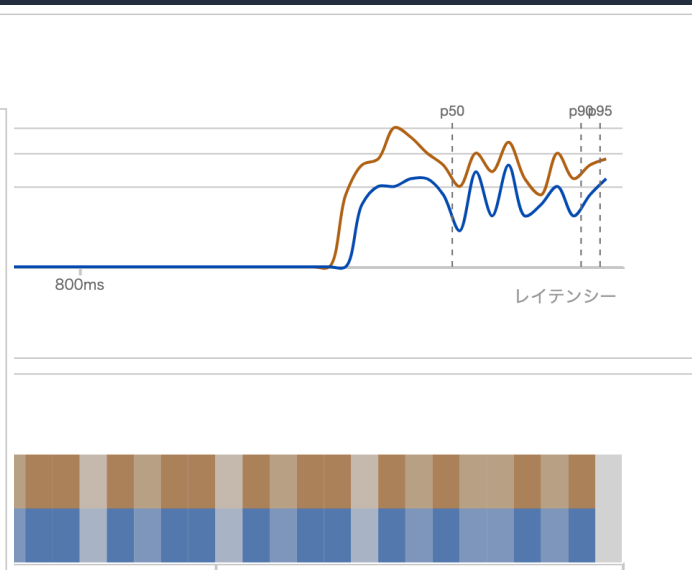
# ワークフローの例 ①

リリースが応答時間に影響を与えていないか確認したい

Http Status Code のテーブルを確認すれば  
リリースがエラーを起こしていないか確認できる

応答  
クリッ  
トシ  
15  
10  
5  
0

HTTP STATUS CODE	COUNT	%
200	130	100.00%

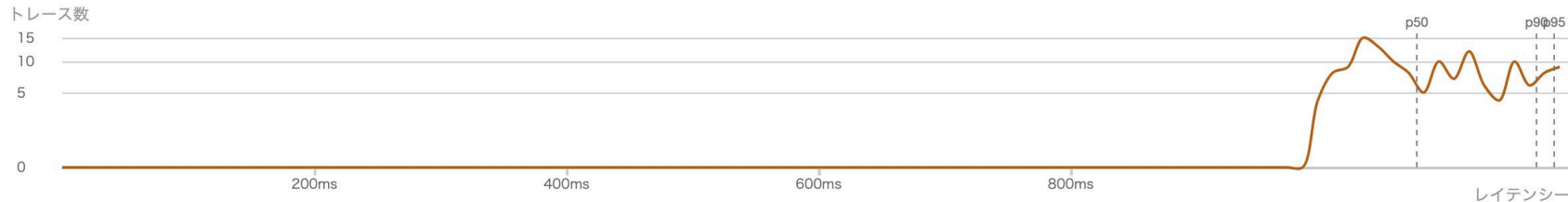


# ワークフローの例 ②

## 応答時間と時間帯に相関があるか調べる

### 応答時間の分布

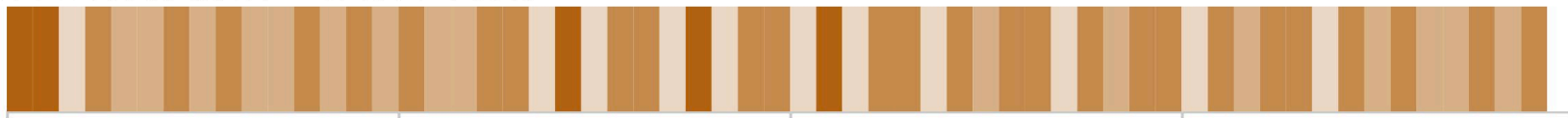
クリックしてドラッグし、応答時間でトレースをフィルタリングします。



応答時間の分布  期間分布

### 時系列アクティビティ

クリックしてドラッグし、時間でトレースをフィルタリングします。

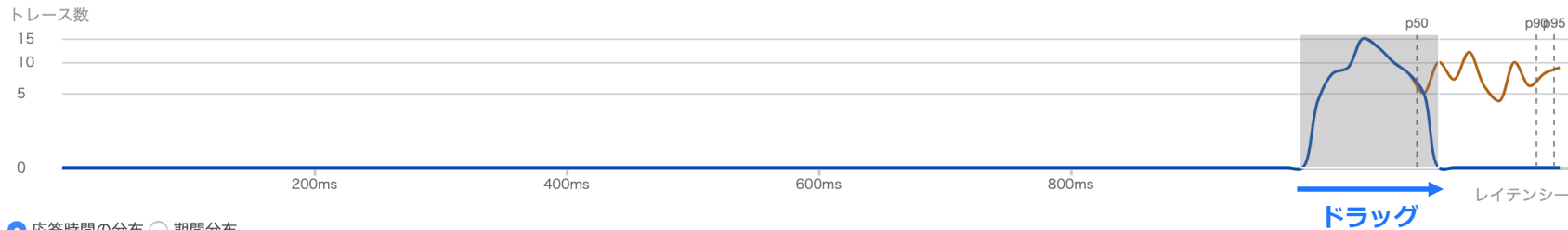


# ワークフローの例 ②

## 応答時間と時間帯に相関があるか調べる

### 応答時間の分布 ①

クリックしてドラッグし、応答時間でトレースをフィルタリングします。



### 時系列アクティビティ ①

クリックしてドラッグし、時間でトレースをフィルタリングします。



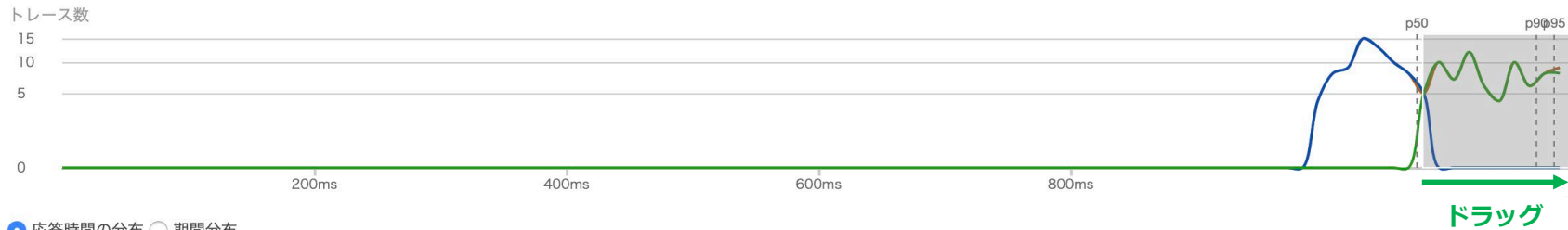


# ワークフローの例 ②

## 応答時間と時間帯に相関があるか調べる

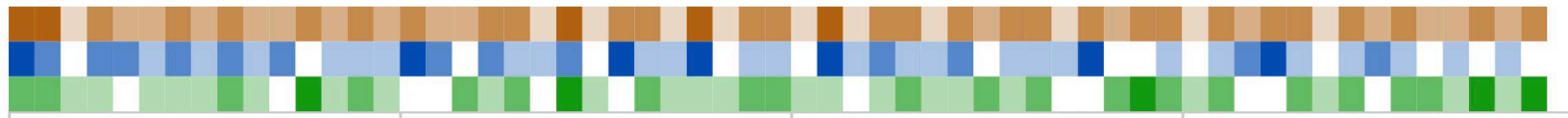
### 応答時間の分布

クリックしてドラッグし、応答時間でトレースをフィルタリングします。



### 時系列アクティビティ

クリックしてドラッグし、時間でトレースをフィルタリングします。

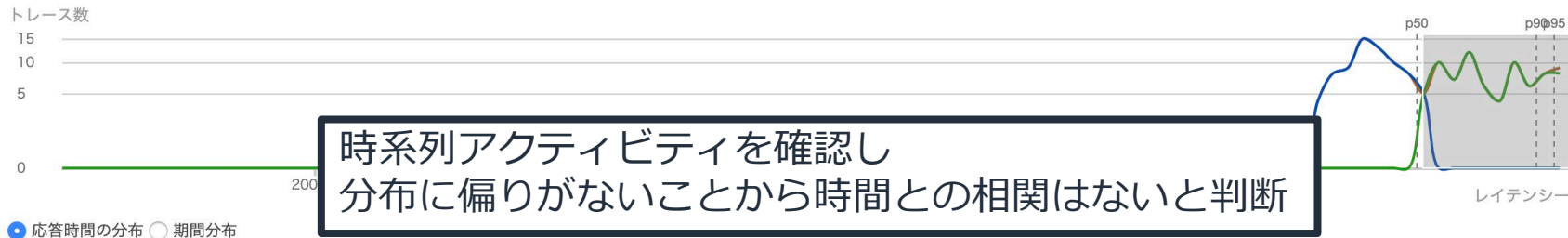


# ワークフローの例 ②

## 応答時間と時間帯に相関があるか調べる

### 応答時間の分布

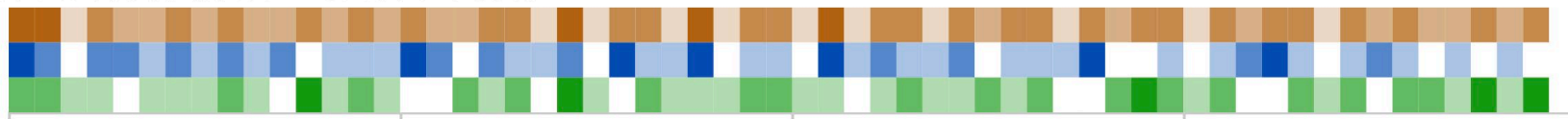
クリックしてドラッグし、応答時間でトレースをフィルタリングします。



### 時系列アクティビティ

クリックしてドラッグし、時間でトレースをフィルタリングします。

偏りが無い



# サンプリング

- サービスタイプ 一致 \*
- ホスト 一致 \*
- リソース ARN 一致 \*
- プロセス名 一致 \*
- メソッド 一致 \*

## AWS X-Ray

はじめに

Service Map

Traces

アナリティクス

☰ 設定

■ サンプリング

暗号化


- サンプリングとは 5% の固定レートに制限する
- サンプリングの作成
  - ルール名・優先度
  - リザーバサイズ・固定レート
  - 一致基準

# サンプリングとは

- AWS X-Ray では**サンプリングルールに従い記録されるリクエストが決まる**
- デフォルトのサンプリングルール
  - 1秒ごとに最初のリクエストは必ずトレースされる
  - それ以降のリクエストについては全体の5%がトレースされる
- なぜサンプリングが必要？
  - 課金額のコントロール
  - パフォーマンスへの影響をコントロール
- 設定方法
  - コンソール ← **今からご紹介**
  - ローカルルール ( SDK から設定を読み込む )

# サンプリングの作成

サンプリングルールの作成

アクション ▾ 

	優先度	ルール	トレンド 
<input type="checkbox"/>	10000	<p><a href="#">Default</a></p> <ul style="list-style-type: none"><li>サービス名 一致 *</li><li>サービスタイプ 一致 *</li><li>ホスト 一致 *</li><li>リソース ARN 一致 *</li><li>HTTP メソッド 一致 *</li><li>URL パス 一致 *</li></ul> <p>1 r/秒、その後 5% の固定レートに制限する</p>	<p>1.2 r/sec (60%)</p>

# サンプリングの作成

サンプリングルールの作成

アクション ▾

優先度	ルール	トレンド
<input type="checkbox"/> 10000	<p><u>Default</u></p> <ul style="list-style-type: none"><li>サービス名 一致 *</li><li>サービスタイプ 一致 *</li><li>ホスト 一致 *</li><li>リソース ARN 一致 *</li><li>HTTP メソッド 一致 *</li><li>URL パス 一致 *</li></ul> <p>1 r/秒、その後 5% の固定レートに制限する</p>	<p>1.2 r/sec (60%)</p>

全ての条件に一致

1 r/秒、その後 5% の固定レートに制限する

ルールが適応されたトレース数

# サンプリングの作成

クリック

サンプリングルールの作成

アクション ▾



	優先度	ルール	トレンド ⓘ
<input type="checkbox"/>	10000	<p><a href="#">Default</a></p> <ul style="list-style-type: none"><li>サービス名 一致 *</li><li>サービスタイプ 一致 *</li><li>ホスト 一致 *</li><li>リソース ARN 一致 *</li><li>HTTP メソッド 一致 *</li><li>URL パス 一致 *</li></ul> <p>1 r/秒、その後 5% の固定レートに制限する</p>	<p>1.2 r/sec (60%)</p>

# ルール名・優先度

## ルールの情報

ルールに名前を付けて優先度を設定します。ルールの作成後は、ルール名を変更できません。ルールは優先度の値が低いものから高いものへと適用されます。

名前

優先度



名前



優先度

値が低い方が優先的に適応  
デフォルトのルールは10000



# リザーバサイズ・固定レート

- リザーバサイズ 1秒間で必ず記録するトレース数
- 固定レート リザーバサイズを超えたリクエスト数のうち何%を記録するか

リザーバのサイズ	<input type="text" value="10"/>	1秒あたりのリクエスト数	
固定レート	<input type="text" value="5"/>	パーセント	

例

リザーバサイズ=50 固定レート=10%  
1秒あたり100件のリクエストがルールに一致する場合

$50$  (リザーバサイズ) +  $50 \times 0.1$  (固定レート) = **55 リクエスト**

デフォルトはリザーバサイズ= 1 固定レート=5%

# 一致基準

## 一致基準

一致させるリクエストを決定するフロントフェーシングサービスの基準を選択します。値には、複数文字一致のワイルドカード (\*) または 1 文字一致のワイルドカード (?) を指定できます。 [詳細はこちら](#)

サービス名

app1



サービス名

\*

例: MyApplication

サービスタイプ

\*

例: AWS::APIGateway::Stage

HTTP メソッド

\*

例: POST

URL パス

\*

例: /api/

リソース ARN

\*

例: arn:aws:execute-api...

ホスト

\*

例: webserver.mydomain.com

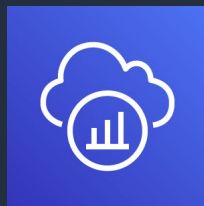
# アジェンダ

- AWS X-Ray とは
- AWS X-Ray の概念
- AWS X-Ray によるアプリケーション分析
- **AWS X-Ray によるデータ収集の仕組み**
- AWS X-Ray の利用方法
- Amazon CloudWatch との連携
- 料金・注意点
- まとめ

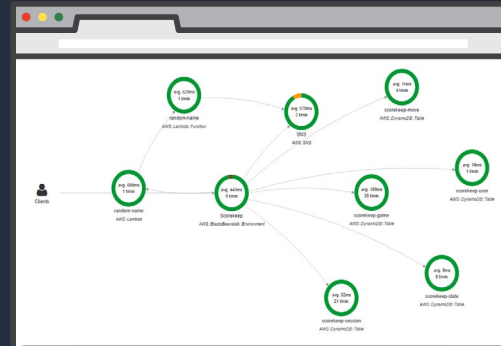
# AWS X-Ray のデータ収集の仕組み



AWS X-Ray デモン



AWS X-Ray API



AWS X-Ray コンソール



AWS X-Ray SDK

アプリケーション



AWS SDK



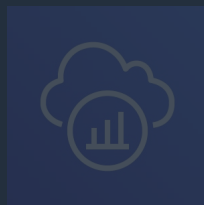
AWS CLI

スクリプトやツール

# AWS X-Ray のデータ収集の仕組み



AWS X-Ray デーモン



AWS X-Ray API



AWS X-Ray コンソール

アプリケーションで作成したセグメントデータを  
X-Ray デーモンに送信



AWS X-Ray SDK

アプリケーション



AWS SDK



AWS CLI

スクリプトやツール

# AWS X-Ray SDK



Java

- トレースデータを作成して X-Ray デーモンに送信するためのクラスとメソッドを提供



Node.js

- 受信リクエスト



Python

- AWS SDK クライアント (Amazon DynamoDB, Amazon S3, ...)



.Net

- 送信 HTTP 呼び出し



Ruby

- SQL クエリ



Go

- カスタムサブセグメント

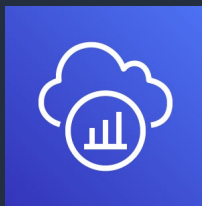
- ...



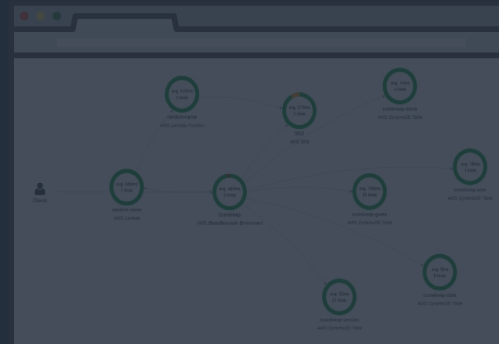
# AWS X-Ray のデータ収集の仕組み



AWS X-Ray デーモン



AWS X-Ray API



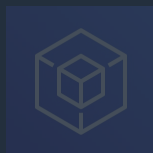
AWS X-Ray コンソール

**X-Ray デーモンはセグメントデータをバッファリングし X-Ray API に定期的に転送**

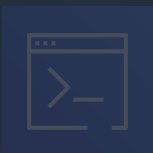


AWS X-Ray SDK

アプリケーション



AWS SDK

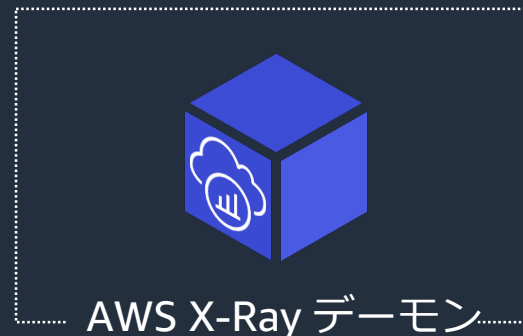


AWS CLI

スクリプトやツール

# AWS X-Ray デーモン

- 未加工のセグメントデータを収集して AWS X-Ray API に送信
- UDP ポート 2000 (デフォルト) をリッスン
- 受信したデータをバッファリングして X-Ray API に送信する
  
- 実行方法
  - Linux                    実行可能ファイル, RPMパッケージ, DEBパッケージ
  - OS X                     実行可能ファイル
  - Windows                実行可能ファイル, サービス
  
  - AWS Lambda            コンソールから設定
  - AWS Elastic Beanstalk   コンソールから設定
  - Amazon ECS             Sidecar として実行

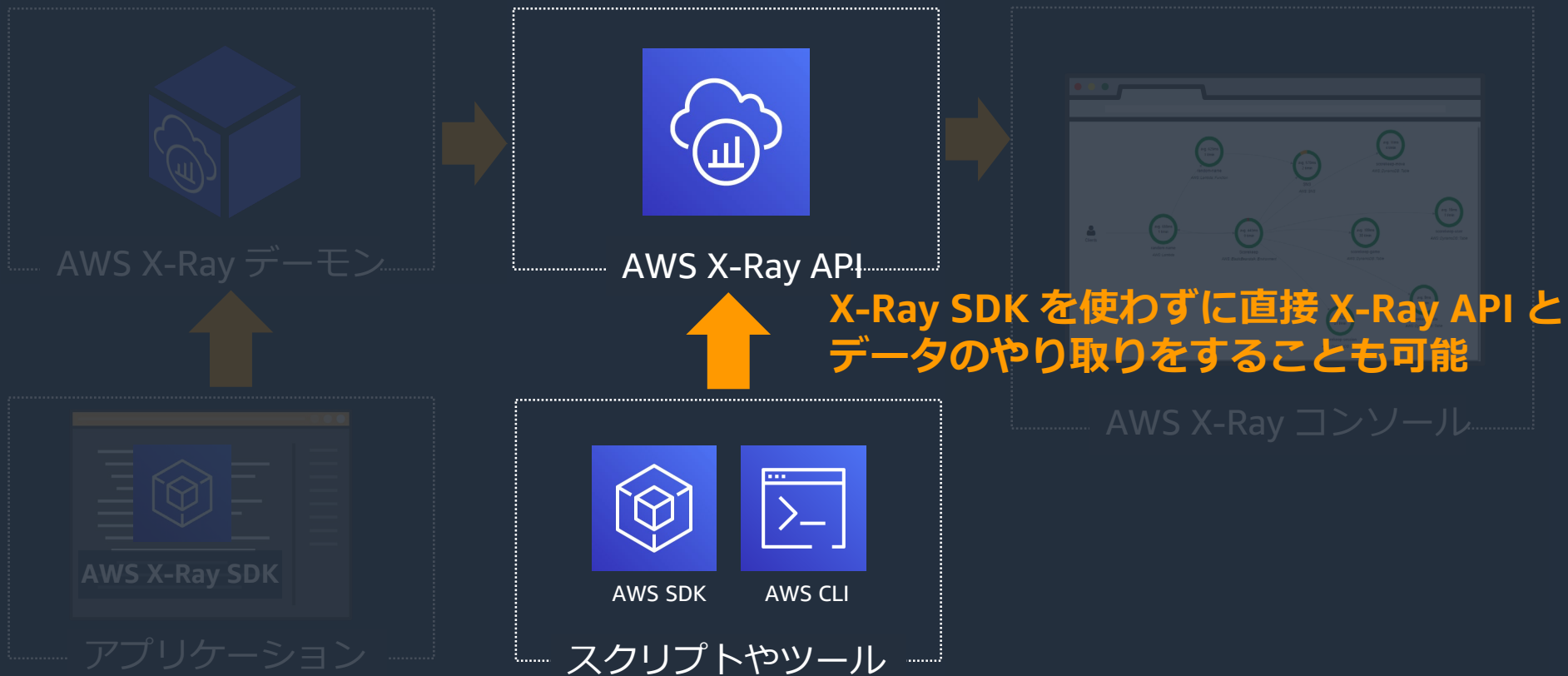




# AWS X-Ray のデータ収集の仕組み



# AWS X-Ray のデータ収集の仕組み



# AWS X-Ray API

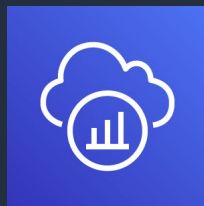
- AWS X-Ray の全ての機能にアクセス可能
  - トレースの追加・取得・更新
  - グループの作成・取得・更新・削除
  - サンプルングルールの作成・取得・更新・削除
- 利用方法
  - HTTPS 経由
  - AWS SDK 経由
  - AWS Command Line Interface 経由



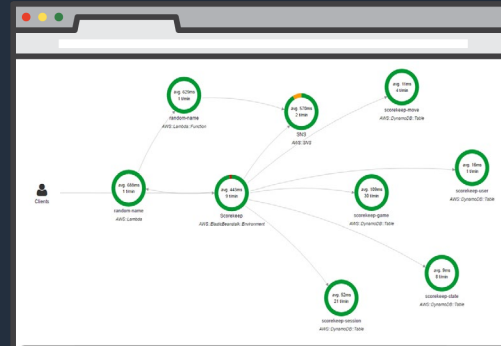
# AWS X-Ray のデータ収集の仕組み



AWS X-Ray デーモン



AWS X-Ray API



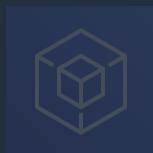
AWS X-Ray コンソール

トレースをコンソールで可視化  
分析・デバッグを行う

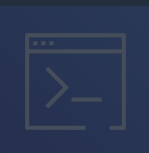


AWS X-Ray SDK

アプリケーション



AWS SDK



AWS CLI

スクリプトやツール

# アジェンダ

- AWS X-Ray とは
- AWS X-Ray の概念
- AWS X-Ray によるアプリケーション分析
- AWS X-Ray によるデータ収集の仕組み
- **AWS X-Ray の利用方法**
- Amazon CloudWatch との連携
- 料金・注意点
- まとめ

# AWS Lambda



- トレースの有効化は AWS Lambda 関数の設定から
- 多少のメモリ消費量の増加あり
- HTTP 呼び出し・注釈・メタデータの追加などには AWS X-Ray SDK が必要

**AWS X-Ray** 情報

アクティブトレースを有効にして、呼び出しのサブセットタイミングとエラー情報を記録します。

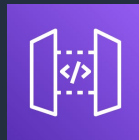
アクティブトレース

X-Ray のトレースを表示



テストイベントの選択 ▼    テスト    **保存**

# Amazon API Gateway



- ステージの設定から X-Ray トレースの有効化
- HTTP リクエストにトレースヘッダーを追加
- AWS X-Ray コンソールで定義したサンプリングルール

設定 ログ/トレース ステージ変数 SDK の生成 エクスポート デプロイ履歴 ドキュメント履歴 Canary

ステージのロギングおよびトレース設定を指定します。

CloudWatch 設定

CloudWatch ログを有効化  ⓘ

詳細 CloudWatch メトリクスを有効化  ⓘ

カスタムアクセスのログ記録

アクセスログの有効化

X-Ray トレース [詳細はこちら](#)

X-Ray トレースの有効化  ⓘ [X-Ray サンプリングルールの設定](#)

変更を保存

# AWS Elastic Beanstalk



- ソフトウェアから AWS X-Ray を有効化
- アプリケーションに AWS X-Ray SDK を組み込む作業は必要

ソフトウェア

X-Ray デモン: 無効  
ログのストリーミング: 無効  
ログのローテーション: 無効  
環境プロパティ: GRADLE\_HOME, JAVA\_HOME, M2, M2\_HOME, XRAY\_ENABLED

編集



**AWS X-Ray**

X-Ray デモン:  
(サービス料金がかかる場合があります。)

有効

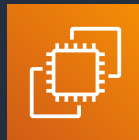
キャンセル

続行

適用



# Amazon EC2



- X-Ray デーモンが実行可能な OS
  - Amazon Linux
  - Ubuntu
  - Windows Server (2012 R2 以降)
- インスタンス起動時にユーザーデータのスクリプトを実行してデーモンを起動

```
#!/bin/bash
```

Amazon Linux の例

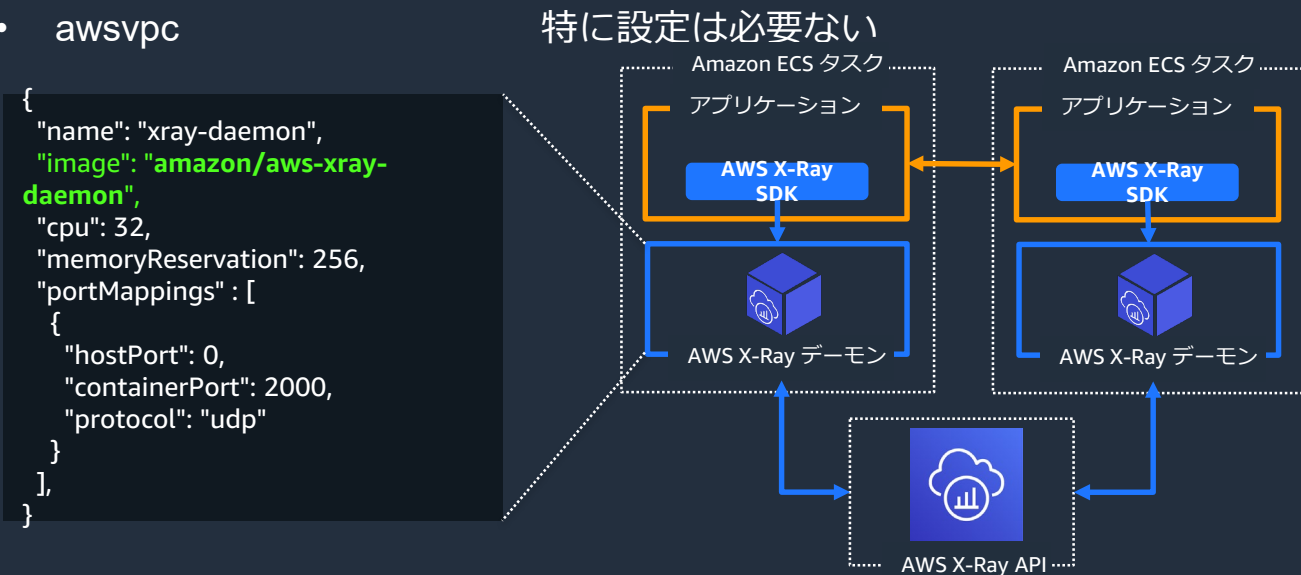
```
curl https://s3.dualstack.us-east-2.amazonaws.com/aws-xray-assets.us-east-2/xray-daemon/aws-xray-daemon-3.x.rpm -o /home/ec2-user/xray.rpm
```

```
yum install -y /home/ec2-user/xray.rpm
```

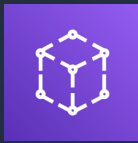
# Amazon ECS



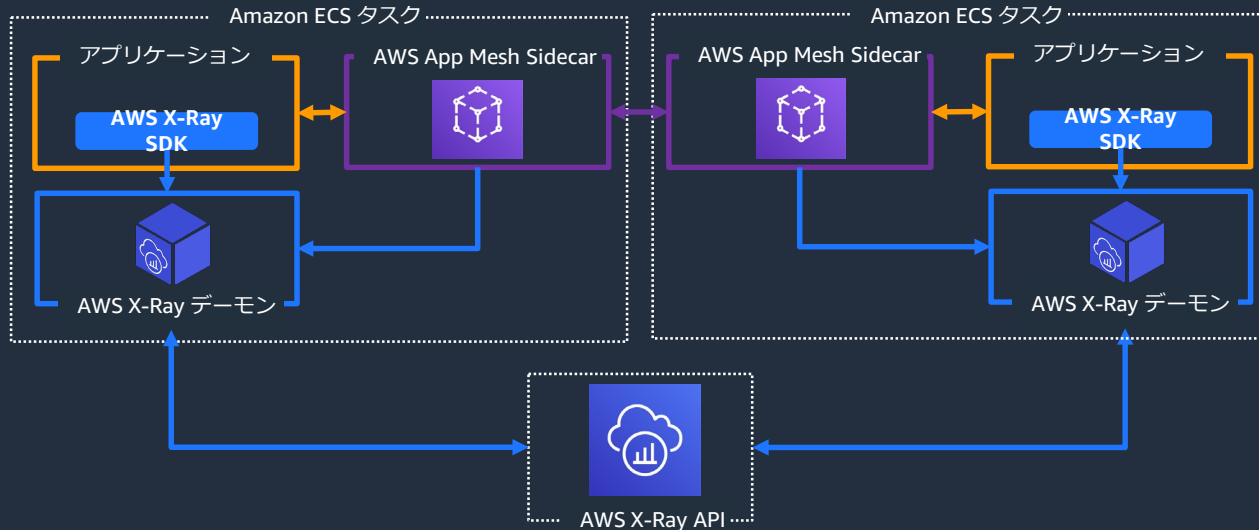
- AWS X-Ray デーモンの Docker コンテナを Sidecar として設定
- ネットワークモードによってコンテナ周りのタスク定義の設定が異なる
  - ブリッジ (デフォルト) リンクを作成し 環境変数 `AWS_XRAY_DAEMON_ADDRESS` に向き先を設定
  - awsvpc



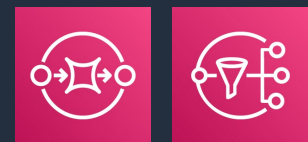
# AWS App Mesh



- AWS App Mesh の Envoy プロキシに環境変数 `ENABLE_ENVOY_XRAY_TRACING: "1"` に設定することで  
AWS X-Ray デーモン (127.0.0.1:2000) にデータを転送
- AWS AppMeshが動作している環境に応じて AWS X-Ray デーモンの設定が必要  
Amazon ECS の例



# Amazon SQS, Amazon SNS

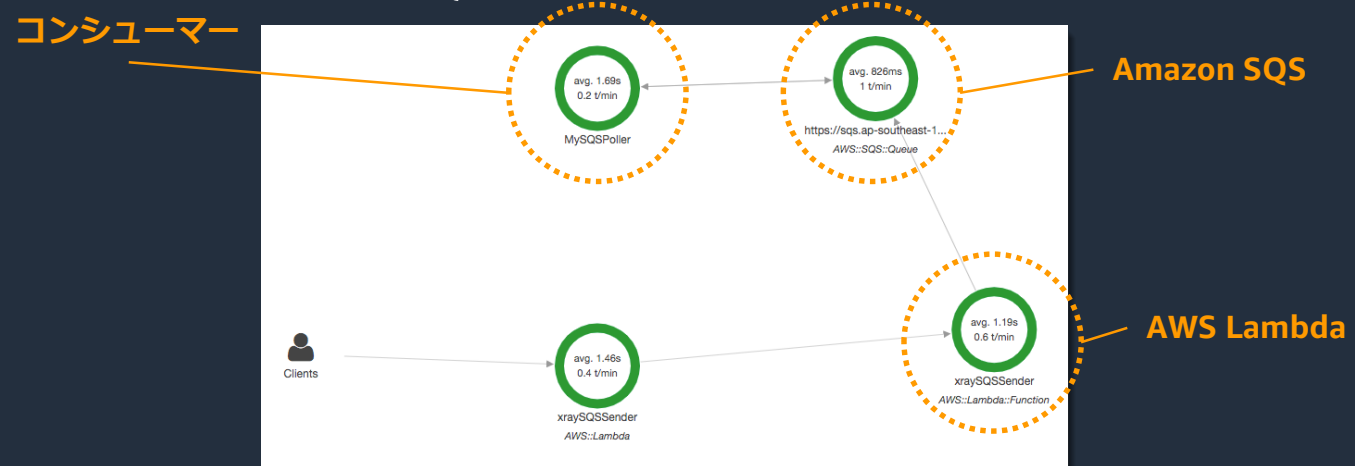


- Amazon SQS と Amazon SNS とも **パッシブ計測**
- 統合を有効化する設定は必要ない

パッシブ計測	別のサービスですでにサンプリングされているリクエストを計測	Amazon SQS, Amazon SNS
アクティブ計測	受信リクエストをサンプリングして計測	Amazon API Gateway, AWS Lambda

\*) Amazon API Gateway と AWS Lambda はパッシブ計測も可能

## Amazon SQS の例



# AWS SAM, CDK

- AWS SAM, CDK から X-Ray の有効化が可能

## AWS Lambda の例

### AWS SAM

```
Type:
AWS::Serverless::Function
Properties:
  FunctionName: MyFunction
  Handler: index.handler
  Tracing: Active
...
```

### CDK (Typescript)



```
const myFunction = new lambda.Function(this,
'myFunction', {
  functionName: 'MyFunction',
  handler: 'index.handler',
  tracing: lambda.Tracing.ACTIVE,
  ...
});
```

# AWS X-Ray SDK (Node.js)

- 受信リクエストのトレース ( express の例 )

```
const app = express();
const AWSXRay = require('aws-xray-sdk');

app.use(AWSXRay.express.openSegment('MyApp'));

app.get('/', function (req, res) {
  res.render('index');
});

app.use(AWSXRay.express.closeSegment());
```

- HTTPリクエストのトレース

```
const AWSXRay = require('aws-xray-sdk');

AWSXRay.captureHTTPGlobal(require('http'));
AWSXRay.capturePromise(); // request-promise, axios
                             利用時

// http を利用する場合
const http = require('http');
// request を利用する場合
const request = require('request');
// axios を利用する場合
const axios = require('axios');
```

# AWS X-Ray SDK (Node.js)

- AWS サービス呼び出しのトレース

```
const AWS = AWSXRay.captureAWS(require('aws-sdk'));
const s3 = new AWS.S3();

// s3 を利用した呼び出しはトレースされる
```

- カスタムサブセグメントの生成

```
const AWSXRay = require('aws-xray-sdk');
```

```
AWSXRay.captureFunc('SubSegment1', (subsegment) => {
  console.log('hello!');
});

AWSXRay.captureAsyncFunc('SubSegment2',
(subsegment) => {
  setTimeout(() => { subsegment.close() }, 1000);
});
```

# アジェンダ

- AWS X-Ray とは
- AWS X-Ray の概念
- AWS X-Ray によるアプリケーション分析
- AWS X-Ray によるデータ収集の仕組み
- AWS X-Ray の利用方法
- **Amazon CloudWatch との連携**
- 料金・注意点
- まとめ



# Amazon CloudWatch ServiceLens

- Amazon CloudWatch と AWS X-Ray を統合

- トレース ← AWS X-Ray のデータを利用
- メトリクス
- ログ
- アラート

メトリクスの詳細やノードマップ

トレースの一覧やフィルタリング機能

サービスマップ

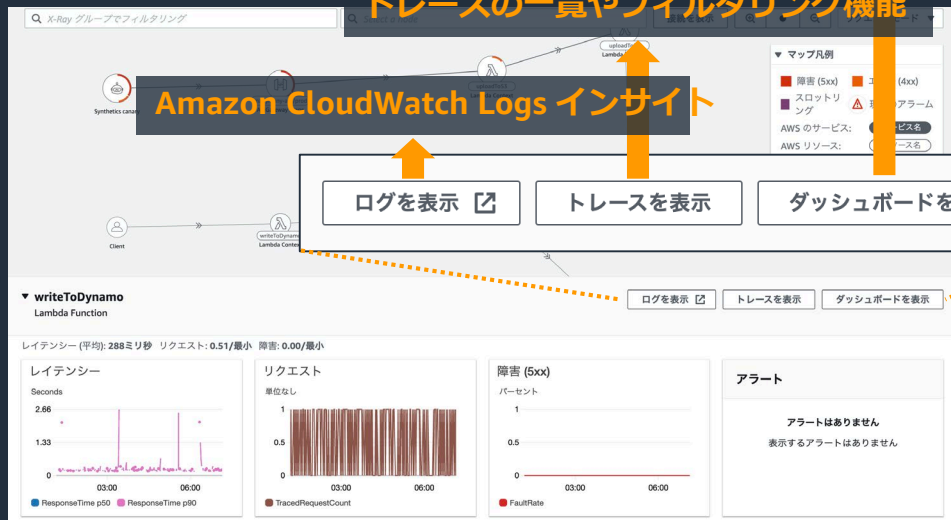
Amazon CloudWatch Logs インサイト

ログを表示

トレースを表示

ダッシュボードを表示

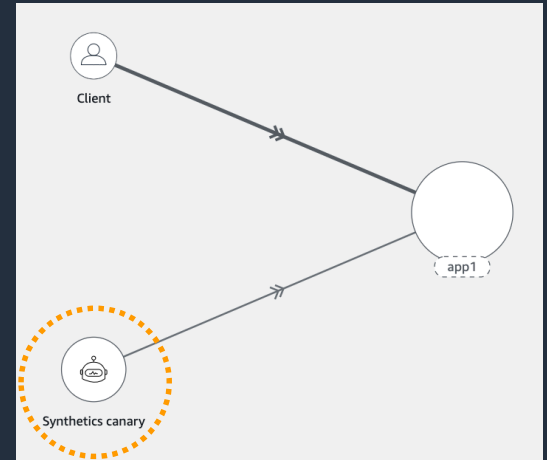
メトリクスグラフ、アラート



# Amazon CloudWatch Synthetics

- Canaryを作成してエンドポイントを監視
- Canaryとは
  - 特定のエンドポイントにアクセス
  - 定期的なスケジュールで実行
  - ロード時間データとUIのスクリーンショットを保存
  - 閾値によってアラームを作成
- Amazon CloudWatch ServiceLens、AWS X-Ray と統合

## サービスマップ



## Canary実行

# アジェンダ

- AWS X-Ray とは
- AWS X-Ray の概念
- AWS X-Ray によるアプリケーション分析
- AWS X-Ray によるデータ収集の仕組み
- AWS X-Ray の利用方法
- Amazon CloudWatch との連携
- **料金・注意点**
- まとめ

# 料金 (2020/5/26 現在)

- 記録、取得、スキャンされたトレースの数に基づいて料金が発生
- 永続的な無料利用枠
  - 毎月 トレースの記録は 10 万回まで無料
  - 毎月 トレースの取得とスキャンは合わせて 100 万回まで無料

トレースの記録	100 万件あたり 5.00USD
トレースの取得	100 万件あたり 0.50USD
トレースのスキャン	100 万件あたり 0.50USD

\* 無料利用枠を超えた分

\* リージョン アジアパシフィック (東京)

<https://aws.amazon.com/xray/pricing/>

# 注意点

- **サンプリングについて**

- デフォルトではサンプリングされる
- 100% トレースする設定も可能だが、料金とパフォーマンスへの影響を事前に確認する

- **データの完全性は保証されない**

- 監査やコンプライアンスのツールとして使用することはできない

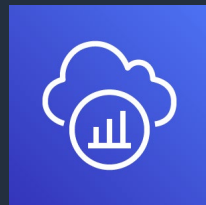
- **直近 30 日間の追跡データが保存**

- 過去 30 日間の追跡データに対してクエリを実行可能

# アジェンダ

- AWS X-Ray とは
- AWS X-Ray の概念
- AWS X-Ray によるアプリケーション分析
- AWS X-Ray によるデータ収集の仕組み
- AWS X-Ray の利用方法
- Amazon CloudWatch との連携
- 料金・注意点
- **まとめ**

# まとめ



## AWS X-Ray

### 有用なツール群

- サービスマップ    マイクロサービスアプリケーションにおいて把握が難しいサービス間の依存関係を**わかりやすく可視化**
- トレースリスト    **詳細なトレースデータ**にアクセス可能  
セグメントやサブセグメントの実行状況を可視化
- アナリティクス    インタラクティブな操作によって**素早い分析が可能**

### 利用の容易さ

- ✓ AWS X-Ray SDK や AWS X-Ray デーモンによってアプリケーションに**簡単に組み込める**
- ✓ 多くの AWS サービスと統合されていて**簡単にセットアップ可能**

# 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



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 🔍

## AWS クラウドサービス活用資料集トップ

アマゾン ウェブ サービス (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**

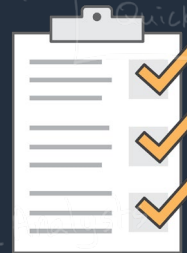
(<https://aws.amazon.com/jp/about-aws/events/>)

AWS イベント

で[検索]



AWS Well-Architected



# 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>

