

# AWS Builders Online Series

T3-3

## AWS ではじめるオブザーバビリティ - システムのどこで・何が・ なぜ起こってるのかを理解する

辻林 侑

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 西日本ソリューショングループ  
ソリューションアーキテクト



# 自己紹介

名前：辻林 侑 (Tsujobayashi Yu)

所属：アマゾン ウェブ サービス ジャパン  
技術統括本部 西日本ソリューショングループ  
ソリューションアーキテクト

経歴：国内電機メーカーでシステムエンジニア

好きなAWSサービス：Amazon CloudWatch



# 対象者、本セッションで学べること

- 対象者
  - アプリケーションの開発、運用に関わる方
  - オブザーバビリティに興味のある方
- 本セッションで学べること
  - オブザーバビリティの概要と実践方法
  - Amazon CloudWatch と AWS X-Ray の概要

# 関連セッションについて

- 3つの関連セッションで構成しています。
- 本セッションでは、オブザーバビリティの概要や基本的な考え方についてご紹介します。Amazon CloudWatch / AWS X-Ray についてご紹介する内容が含まれます。

## 本セッションで学べること

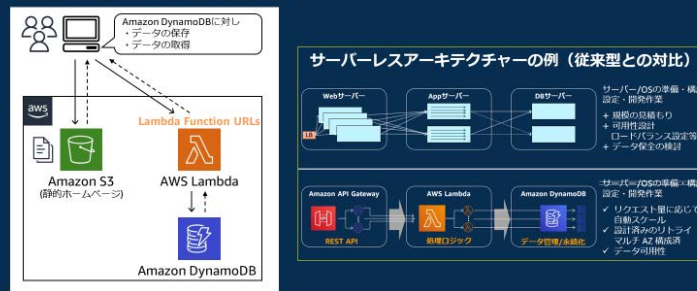
- 本セッションで学べること
  - なぜモダナイゼーションが必要か
  - モダナイゼーションを進める上でのコンテナ活用のメリット
  - AWS コンテナサービスを活用することで解決できる課題



T3-1: はじめてのコンテナワークロード - AWS でのコンテナ活用の第一歩

## 本セッションで学べること

- サーバーレスとは何か
- サーバーレスな Web アプリケーション



T3-2: はじめてのサーバーレス - AWS でのサーバーレスアプリケーション開発

## 本セッションで学べること

- オブザーバビリティとは何か、なぜ必要か
- AWS でオブザーバビリティを実現する方法



T3-3: AWS ではじめるオブザーバビリティ - システムのどこで・何が・なぜ起こってるのかを理解する

# アジェンダ

1. オブザーバビリティとは
2. オブザーバビリティを実現するには
3. AWS ではじめるオブザーバビリティ
4. まとめ



# オブザーバビリティとは



# オブザーバビリティ（可観測性）

システムのどこで・何が・なぜ起こってるのかといった動作状況を把握できている状態

→ システム運用において、判断に必要な情報がきちんと取得できている状態

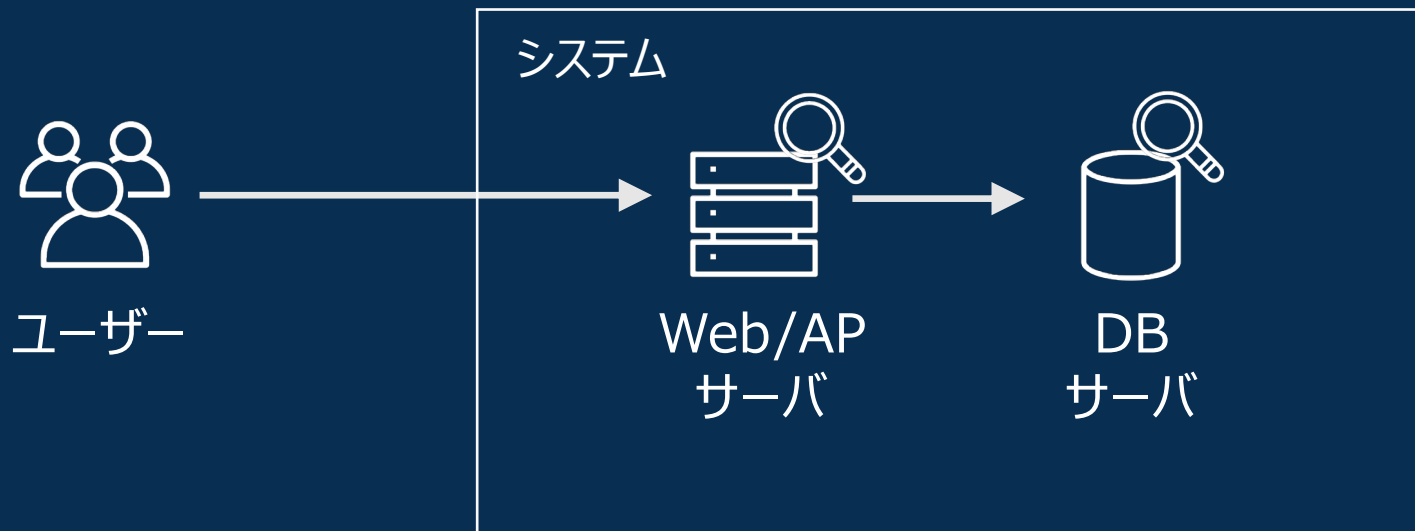


例)

- このサービスでは、メソッドの 90% が 200 ミリ秒以下で完了している
- この API は 1 秒間に 203 の HTTP リクエストを処理している
- このサービスの CPU 使用率は 85% に達している

# これまでのモニタリング（監視）

- ・ オンプレミスのシステム、モノリシックなアプリケーションなどの場合



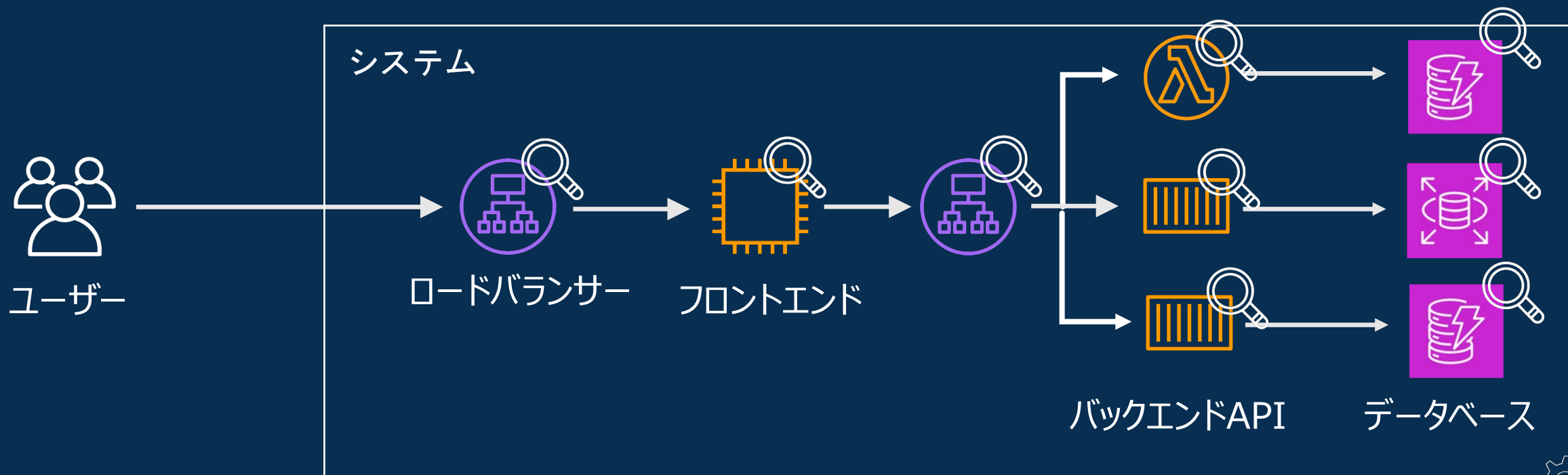
このWeb/AP サーバと DB サーバの CPU, メモリ利用率やログを監視しておけばシステムに何が起きたかわかるぞ  
→ **インフラを監視していれば、システムの状態を把握できる**





# これまでのモニタリング（監視）

- クラウドネイティブなシステム、マイクロサービスなどの場合



構成が複雑化すると監視対象も多く、コンテナやサーバレスは動的にスケールする。。これまでのやり方ではシステム全体で何が起きているか把握できない。。

→ **システム全体の状態を把握できない...**



システム  
担当者

# オブザーバビリティはなぜ必要か



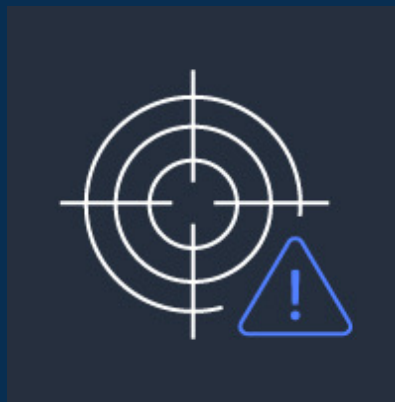
予測可能な範囲はモニタリングで対応できるが  
未知の事象や早い変化への対応にはオブザーバビリティが求められる

# オブザーバビリティで得られるメリット

- ・ システムのどこで・何が・なぜ起こってるのかといった動作状況を把握できている状態  
→ システム運用において、判断に必要な情報がきちんと取得できている状態



視認性



迅速なトラブルシューティング



顧客体験

運用からビジネス面に至るメリットを得ることができる

# オブザーバビリティを実現するには

# オブザーバビリティの実現に向けて

マイクロサービスなどの複雑なシステムを観測可能にするには？

- 観測可能なシステムは、次のイベントを発するべきである：
  - メトリクス、ログ、トレース
- 全てのコンポーネントは計測されている必要がある：
  - 可能な限り簡単に計測を
    - AWS ネイティブサービスの活用
    - マネージドオープンソースサービスの活用
- インフラストラクチャだけでなく、アプリケーション、**ユーザー体験も計測**する



Mobile



Browser



Application



Host operating system and containers



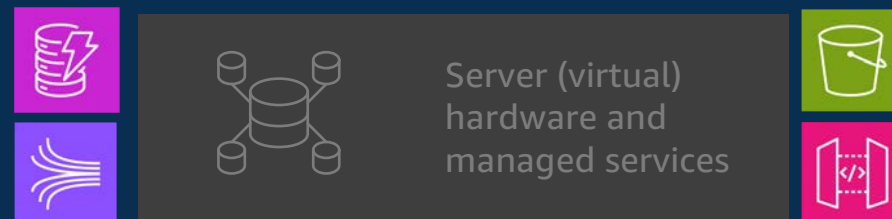
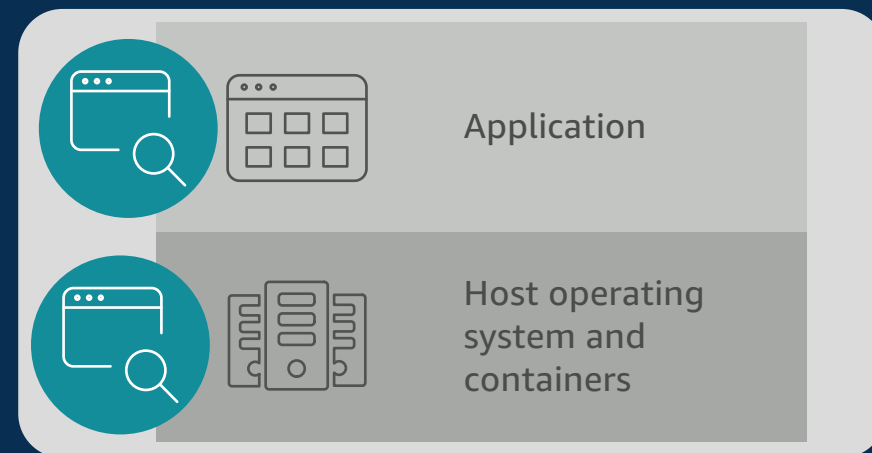
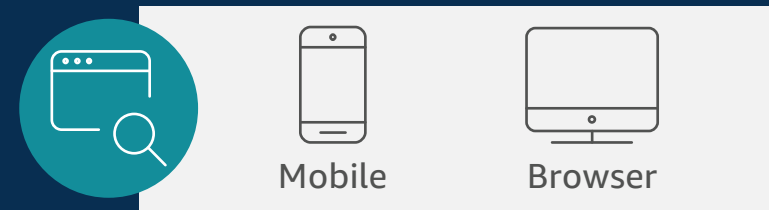
Server (virtual) hardware and managed services



# オブザーバビリティの実現に向けて

マイクロサービスなどの複雑なシステムを観測可能にするには？

- 観測可能なシステムは、次のイベントを発するべきである：
  - メトリクス、ログ、トレース
- 全てのコンポーネントは計測されている必要がある：
  - 可能な限り簡単に計測を
    - AWS ネイティブサービスの活用
    - マネージドオープンソースサービスの活用
- インフラストラクチャだけでなく、アプリケーション、ユーザー体験も計測する



# メトリクス、ログ、トレース

- ・オブザーバビリティでよく利用されるテレメトリデータ

## メトリクス



時間間隔で計測されたデータの  
数値表現

傾向の把握、予測に  
役立つ

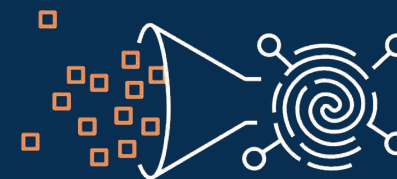
## ログ



タイムスタンプが記録された、  
時間の経過とともに起こった  
イベントの記録

予測不可能な振る舞いの発見に  
役立つ

## トレース



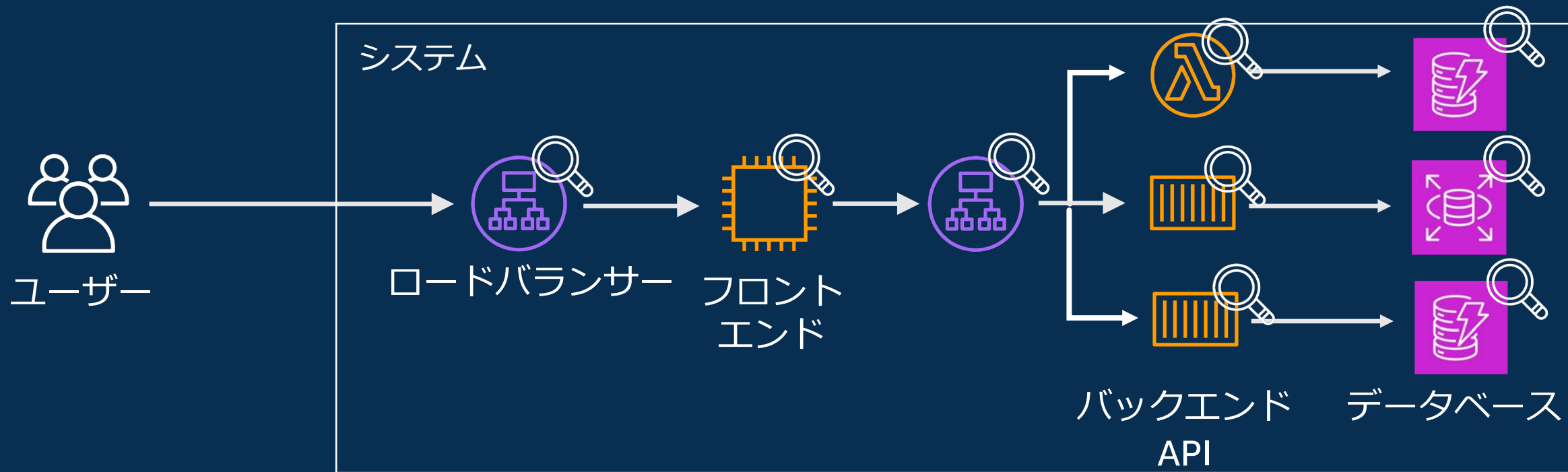
エンドツーエンドの  
リクエストフローの記録

リクエストの流れと構造の両方を  
可視化することで因果関係の追跡  
に役立つ



# トレース

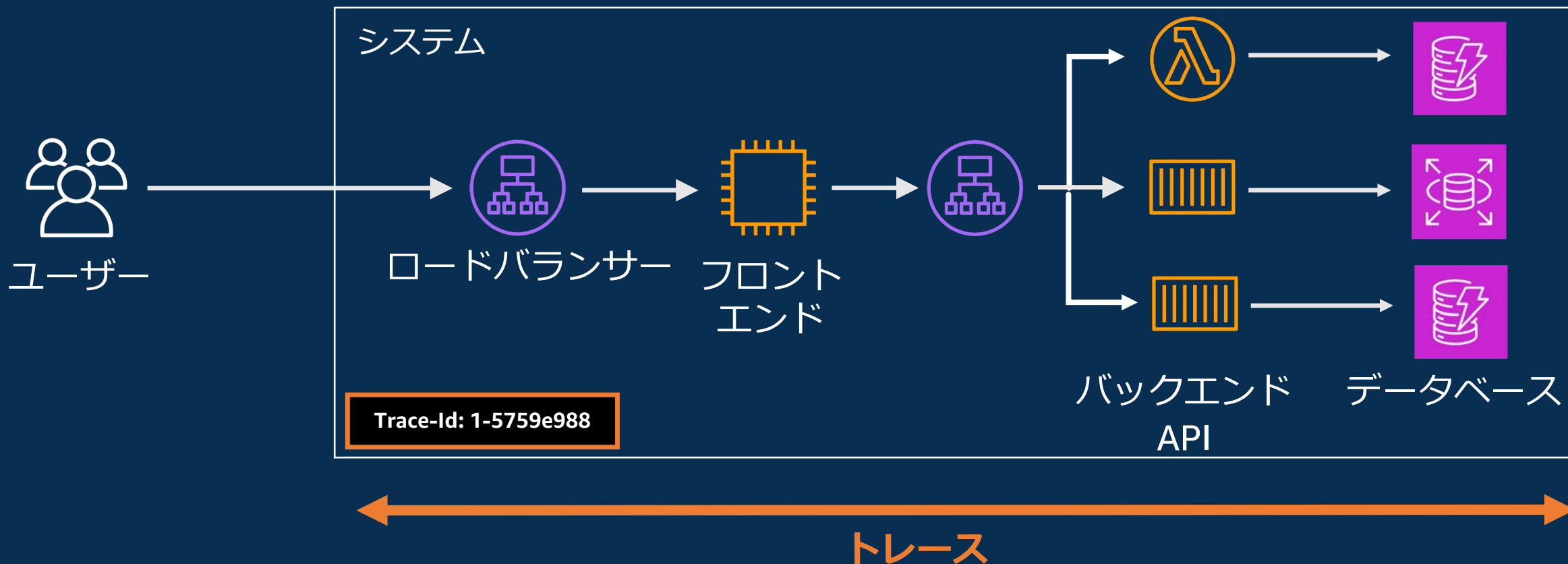
- クラウドネイティブなシステム、マイクロサービスなどの場合  
→ 多数のシステムにログインして、**各リクエストのログを見ることは、非現実的**





# トレース

- 「トレース」によりリクエストからレスポンスまでの**全体の流れを追跡、把握**
- 障害箇所の特定やパフォーマンス分析につながる情報を確認



# メトリクス、ログ、トレース

- ・オブザーバビリティでよく利用されるテレメトリデータ

## メトリクス



時間間隔で計測されたデータの  
数値表現

傾向の把握、予測に  
役立つ

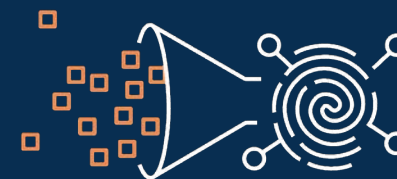
## ログ



タイムスタンプが記録された、  
時間の経過とともに起こった  
イベントの記録

予測不可能な振る舞いの発見に  
役立つ

## トレース



エンドツーエンドの  
リクエストフローの記録

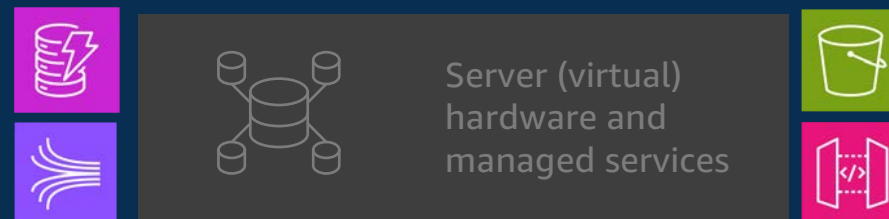
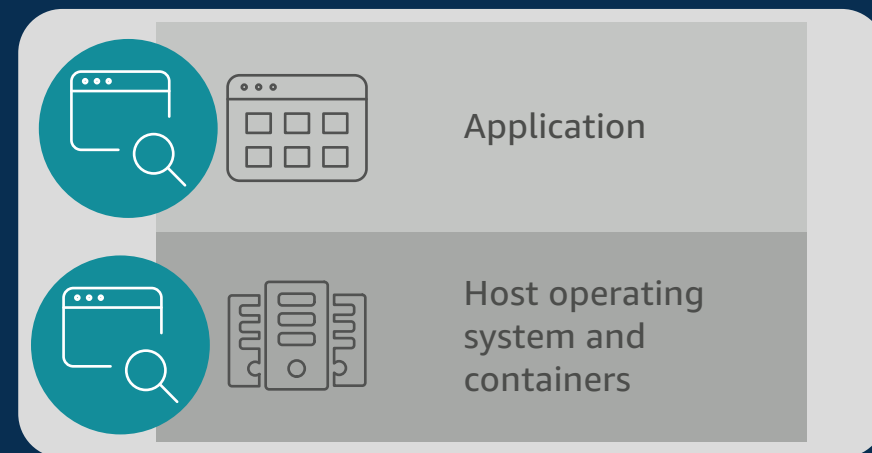
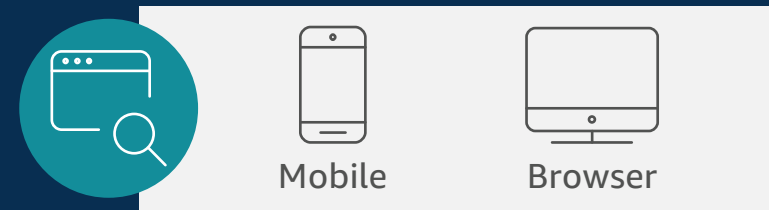
リクエストの流れと構造の両方を  
可視化することで因果関係の追跡  
に役立つ



# オブザーバビリティの実現に向けて

マイクロサービスなどの複雑なシステムを観測可能にするには？

- 観測可能なシステムは、次のイベントを発するべきである：
  - メトリクス、ログ、トレース
- 全てのコンポーネントは計測されている必要がある：
  - 可能な限り簡単に計測を
    - AWS ネイティブサービスの活用
    - マネージドオープンソースサービスの活用
- インフラストラクチャだけでなく、アプリケーション、**ユーザー体験も計測**する



# 全てのコンポーネントを計測する

計測対象

クライアント / フロント

アプリケーション

ミドルウェア

OS

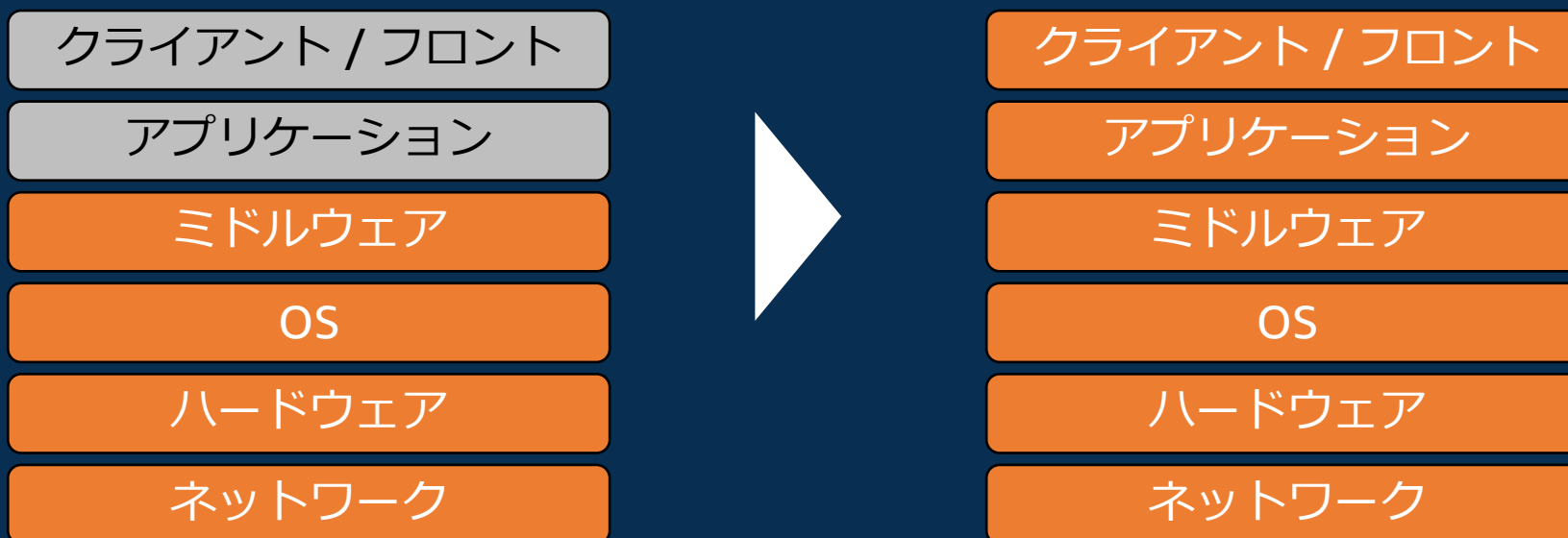
ハードウェア

ネットワーク

# 全てのコンポーネントを計測する

計測対象

- サービス状態や実際の利用状況をモニタリングする
- 障害の把握や性能改善、機能改善につながる洞察を得る



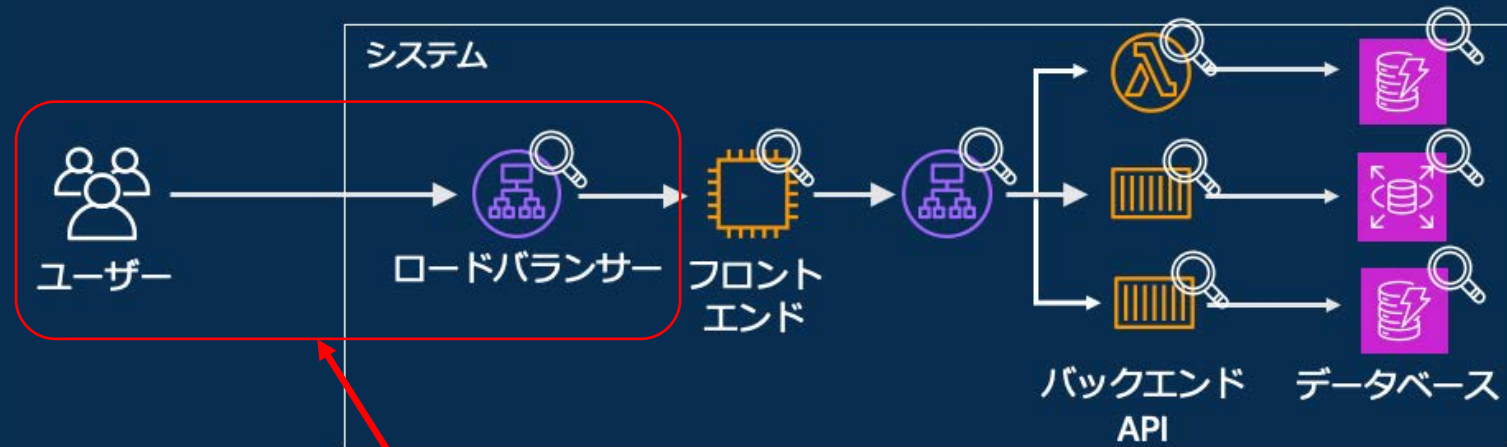
インフラに限定せずアプリケーションまで含めて  
サービスを“総合的に”評価する必要がある

# ユーザー体験を計測する

エンドユーザーにとって「良い体験」とは何かを確立することから始める

例)

- ウェブページの応答時間
- ログインページの応答速度
- 商品購入時のエラー率
- JavaScript のエラー率



CPU利用率高 = システム停止？  
システム的には問題ないはず  
だがユーザーの一部から問題が  
報告された。。なぜ？

# オブザーバビリティのサイクル



\*インストルメンテーション= メトリクス、ログ、トレースなどのデータを取得し外部に送信できるようにシステムに組み込むこと

# AWS ではじめるオブザーバビリティ



# AWS のオブザーバビリティサービス

## Observability

### AWS native services



Amazon CloudWatch ServiceLens

Container  
insights

Lambda  
insights

Contributor  
insights

Application  
insights



Synthetics



Dashboards



Alarms



RUM



Internet monitor



Metrics



Logs



AWS X-Ray

### Open-source managed services



Amazon Managed Grafana

Do it yourself (DIY) – AWS OSS Solutions



Amazon OpenSearch  
Service



Amazon Managed Service  
for Prometheus



JAEGER  
ZIPKIN  
Jaeger and  
Zipkin Tracing

Insights & ML

### Collectors and SDKs



CloudWatch  
agent



AWS X-Ray  
agent



AWS Distro for  
OpenTelemetry

## Instrumentation

# AWS のオブザーバビリティサービス

## Observability

### AWS native services



Amazon CloudWatch ServiceLens

Container  
insights

Lambda  
insights

Contributor  
insights

Application  
insights



Synthetics



Dashboards



Alarms



RUM



Internet monitor



Metrics



Logs



AWS X-Ray



CloudWatch  
agent



AWS X-Ray  
agent

Collectors and SDKs

### Open-source managed services



Amazon Managed Grafana

Do it yourself (DIY) – AWS OSS Solutions



Amazon OpenSearch  
Service



Amazon Managed Service  
for Prometheus



JAEGER  
ZIPKIN  
Jaeger and  
Zipkin Tracing



AWS Distro for  
OpenTelemetry

Insights & ML

## Instrumentation

# メトリクス、ログ、トレース

- ・オブザーバビリティでよく利用されるテレメトリデータ

## メトリクス



時間間隔で計測されたデータの  
数値表現

傾向の把握、予測に  
役立つ

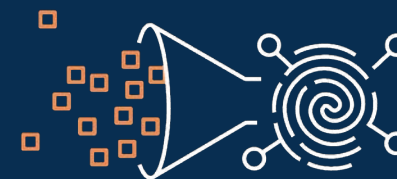
## ログ



タイムスタンプが記録された、  
時間の経過とともに起こった  
イベントの記録

予測不可能な振る舞いの発見に  
役立つ

## トレース



エンドツーエンドの  
リクエストフローの記録

リクエストの流れと構造の両方を  
可視化することで因果関係の追跡  
に役立つ



# メトリクス、ログ、トレース

- ・オブザーバビリティでよく利用されるテレメトリデータ

## メトリクス



時間間隔で計測されたデータ  
の数値表現



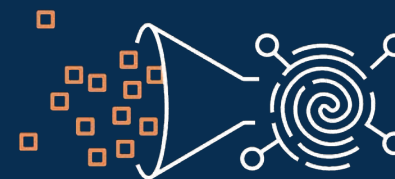
Amazon CloudWatch

## ログ



タイムスタンプが記録された、  
時間の経過とともに起こった  
イベントの記録

## トレース



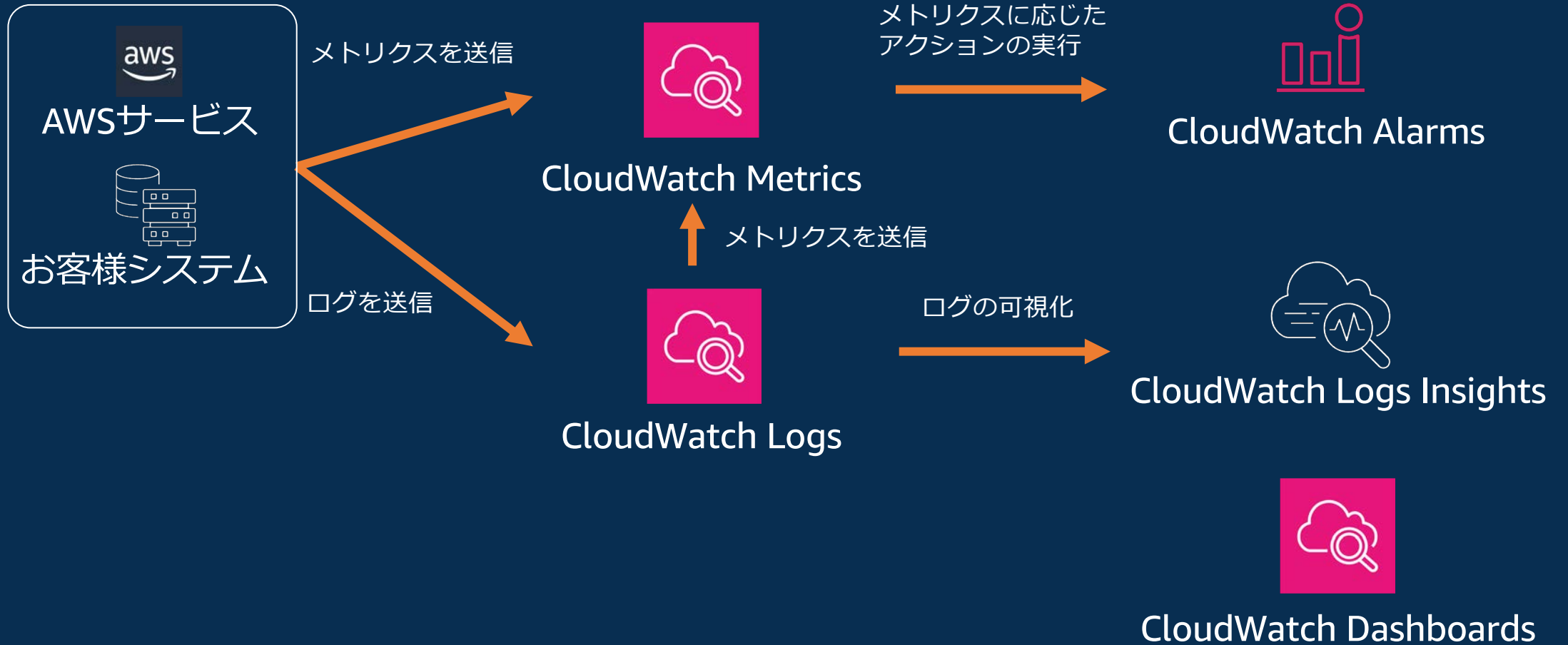
エンドツーエンドの  
リクエストフローの記録



AWS X-Ray

# Amazon CloudWatch

モニタリング・オブザーバビリティに関する様々な機能を提供



# CloudWatch Metrics

さまざまな AWS サービスからメトリクスを自動的に収集することが可能

## 標準メトリクス

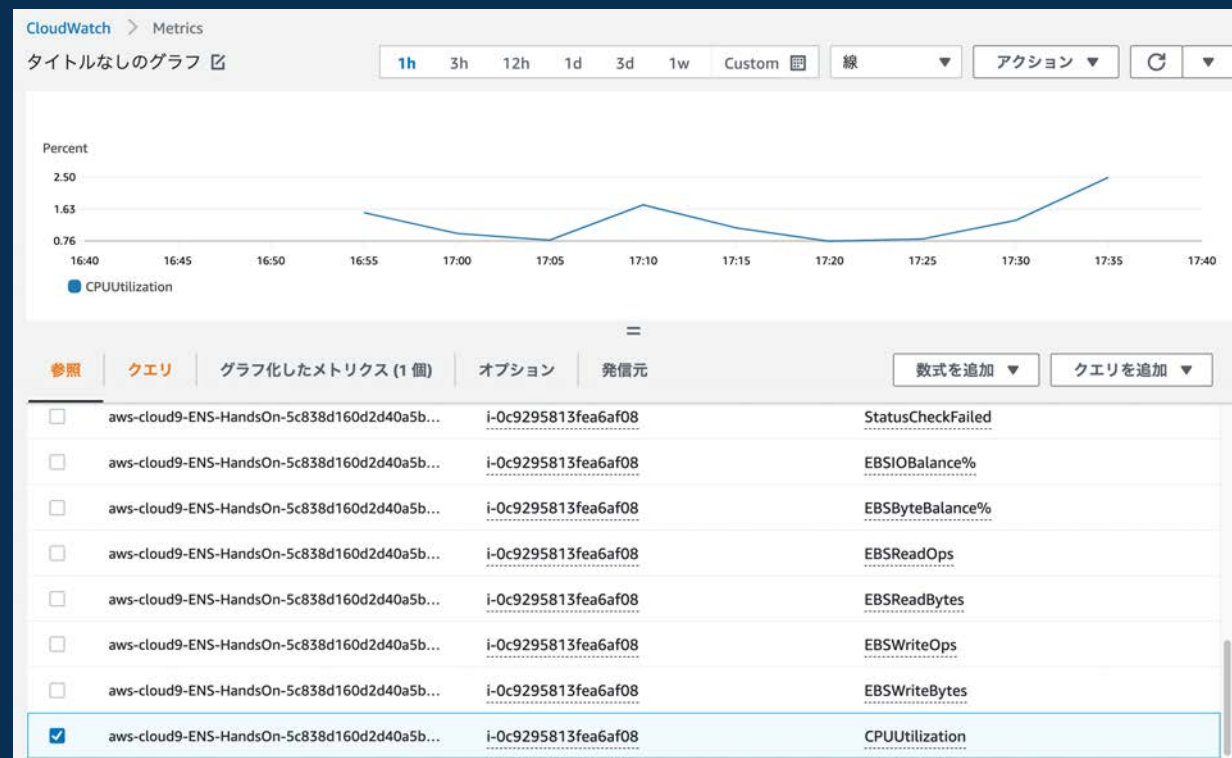
各AWSサービスにおいて、デフォルトで発行されるメトリクス例)

- Amazon EC2
  - CPU利用率
  - ネットワーク使用量
- Amazon Elastic Load Balancing
  - リクエスト数

## カスタムメトリクス

お客様自身のアプリケーションで生成されたデータをモニタリング可能

- API などからメトリクスを発行
- 標準は1分単位で最大 1 秒の解像度でメトリクスを収集可能



# CloudWatch Logs

## AWS サービスおよびアプリケーションログの監視、保存、アクセスを提供

- ログデータは耐久性が高く安価なストレージへ
- 保存ログデータの保持期間も自由に設定可能  
(1日～永久)



### メトリクスフィルタ

- ログデータから特定文字列のフィルタリング
- 一致したパターン数をメトリクスとして記録

例) ERROR ログ出現数によってメトリクスとアラームを作成、通知を行う



CloudWatch Logs



CloudWatch Metrics

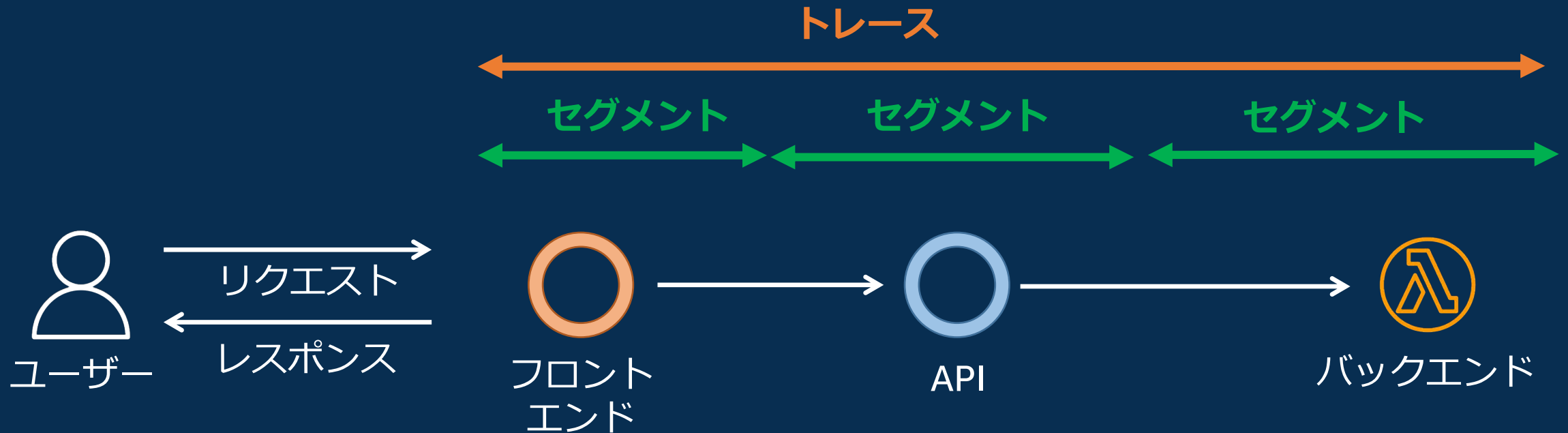


CloudWatch Alarms

# AWS X-Ray

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

- 「トレース」によりリクエストからレスポンスまでの全体の流れを追跡する
- 「セグメント」は動作に関するデータで、リソース名やリクエストの詳細が含まれる



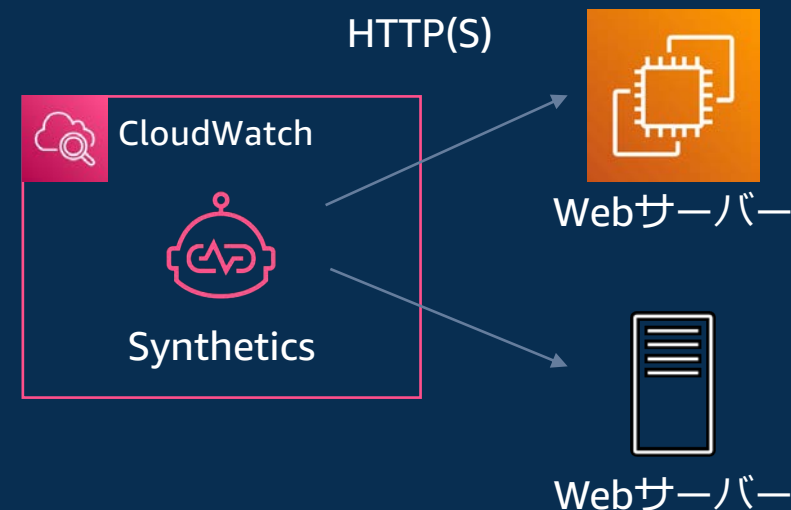


# その他 CloudWatch サービス

## アプリケーション、ユーザー体験を計測

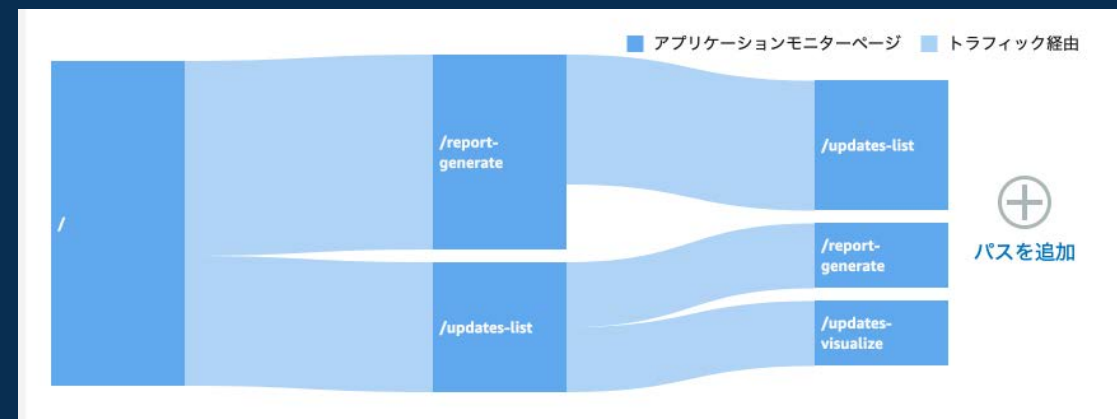
### CloudWatch Synthetics

- フルマネージドで **Synthetic Monitoring** を行うサービス
  - 一定間隔で計測プログラムがアプリケーションにアクセス、パフォーマンスを測定する手法
- 既存AWSサービスとの連携による高い柔軟性



### CloudWatch RUM (Real User Monitoring)

- ユーザーの振る舞いや体験に関わるデータを分析
  - 例) ページのロード時間、アクセス元、ユーザー遷移の流れ、エラー回数等
- CloudWatchから発行されるコードをアプリケーションに埋め込んで実装



# デモ： Amazon CloudWatch と AWS X-Ray で アプリケーションを調査してみよう

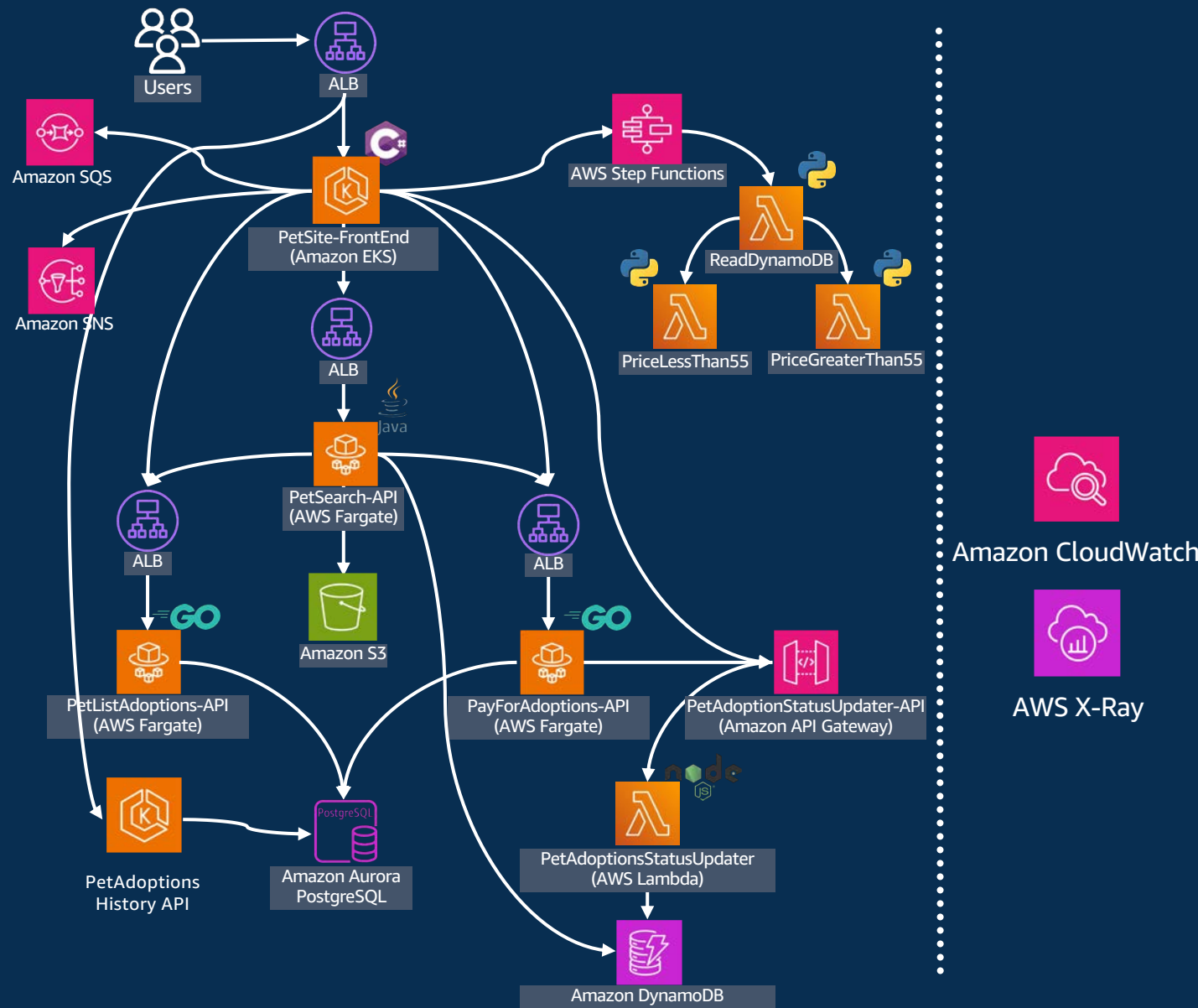
# デモの内容

- ペットの里親募集・申込サイトの運用を担当
- フロントエンド・各 API がコンテナ管理・モジュール化されたマイクロサービスアーキテクチャで構成

## 主な機能：

- ペットの検索・一覧表示機能
- 里親が既に決まったペットの一覧表示
- 里親がまだ決まってないペットの里親申請

- サービス利用中に時々エラーが発生すると報告があり、調査を開始



# 補足（料金の考え方）



## Amazon CloudWatch Metrics

- カスタムメトリクス数と CloudWatch Metrics へ API リクエスト数に応じて料金発生
  - 基本モニタリングメトリクス（デフォルトでAWS サービスから送信されるメトリクス：5分間隔）は無料
  - 10 メトリクス (カスタムメトリクスと詳細モニタリングメトリクス) まで無料
  - 100 万件の API リクエストは無料（\* 一部の API は対象外）

範囲	料金（メトリクス/月）
最初の 10,000 メトリクス	0.30 USD
次の 240,000 メトリクス	0.10 USD
次の 750,000 メトリクス	0.05 USD
1,000,000 を超えるメトリクス	0.02 USD

API種別	料金（メトリクス/月）
GetMetricData、 GetInsightRuleReport	1000 メトリクスリクエストあたり 0.01 USD
GetMetricWidgetImage	1000 メトリクスリクエストあたり 0.02 USD
GetMetricStatistics, ListMetrics, PutMetricData など	1000 リクエストあたり 0.01 USD

\* 表はカスタムメトリクス、詳細モニタリングメトリクスの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)

\* その他、詳細な料金については下記を参照

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

# 補足（料金の考え方）



## Amazon CloudWatch Logs

- データ取り込み、保存、分析されたログの量に基づいて料金が発生
  - 5 GB データ (取り込み、ストレージのアーカイブ、Logs Insights クエリによってスキャンされたデータ) は無料

種別	料金
収集（データ取り込み）	0.76 USD / GB
保存（アーカイブ）	0.033 USD / GB
分析（Logs Insight のクエリ）	スキャンしたデータ 1 GB あたり 0.0076 USD
検出およびマスク (データ保護)	スキャンしたデータ 1 GB あたり 0.12 USD
分析 (Live Tail)	0.01 USD / 分

\* 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)

\* その他 Vended Logs（AWS のサービスがお客様に代わってネイティブに発行する、AWS のサービスに関する特定のログ）に関する料金の考え方については下記を参照

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

# 補足（料金の考え方）



## AWS X-Ray

- 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
  - 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

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

\* 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/>

# まとめ

- オブザーバビリティとは
  - システムのどこで・何が・なぜ起こってるのかといった動作状況を把握できている状態
  - 複雑な構成のシステムにおいても視認性向上、迅速なトラブルシューティング、ユーザー体験向上に役立つ
- オブザーバビリティを実現するには
  - 可観測性を向上するためのデータを取得（メトリクス・ログ・トレースなど）
  - インフラだけでなく、アプリケーションの層も含めて全てのコンポーネントを計測
  - ユーザー体験を計測
- AWS でオブザーバビリティをはじめするには
  - Amazon CloudWatch、AWS X-Ray をはじめとしたマネージドサービスを活用することで、運用負荷少なくオブザーバビリティの実践が可能

# 今後に向けて

- One Observability workshop をやってみよう！
  - Amazon CloudWatch を始めとしたオブザーバビリティ関連サービスの様々な機能を体感いただけます。  
<https://catalog.workshops.aws/observability/ja-JP>
- AWS Observability Best Practices を読んでみよう！
  - オブザーバビリティや AWS の各サービスを活用したベストプラクティスについて理解を深めることができます。  
<https://aws-observability.github.io/observability-best-practices/>





# Thank you!



# AWS TRAINING & CERTIFICATION

## 600+ ある AWS Skill Builder の無料デジタルコースで学ぼう

30 以上の AWS ソリューションの中から、自分にもっとも関係のあるクラウドスキルとサービスにフォーカスし、自習用のデジタル学習プランとランプアップガイドで学ぶことができます。

## 自分に合ったスキルアップ方法で学ぼう

**EXPLORE.SKILLBUILDER.AWS »**



## あなたのクラウドスキルを AWS 認定で証明しよう

業界で認められた資格を取得して、スキルアップの一步を踏み出しましょう。AWS Certified の取得方法と、準備に役立つ AWS のリソースをご覧ください。

## **受験準備のためのリソースにアクセスしよう »**



# AWS Builders Online Series にご参加いただきありがとうございます

楽しんでいただけましたか? ぜひアンケートにご協力ください。  
本日のイベントに関するご意見/ご感想や今後のイベントについてのご希望や改善のご提案などがございましたら、ぜひお聞かせください。



[aws-apj-marketing@amazon.com](mailto:aws-apj-marketing@amazon.com)



[twitter.com/awscloud\\_jp](https://twitter.com/awscloud_jp)



[facebook.com/600986860012140](https://facebook.com/600986860012140)



<https://www.youtube.com/user/AmazonWebServicesJP>



<https://www.linkedin.com/showcase/aws-careers/>



[twitch.tv/aws](https://twitch.tv/aws)