



# Amazon CloudWatchの概要と基本

## AWS Black Belt Online Seminar

津和崎 美希

Solutions Architect  
2023/03

# AWS Black Belt Online Seminarとは

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

# 内容についての注意点

- ・ 本資料では2023年03月時点のサービス内容および価格についてご説明しています。最新の情報はAWS公式ウェブサイト(<https://aws.amazon.com/>)にてご確認ください
- ・ 資料作成には十分注意しておりますが、資料内の価格とAWS公式ウェブサイト記載の価格に相違があった場合、AWS公式ウェブサイトの価格を優先とさせていただきます
- ・ 価格は税抜表記となっています。日本居住者のお客様には別途消費税をご請求させていただきます

# 自己紹介

名前 :

津和崎 美希 (つわざき みき)

所属 :

エンタープライズ技術統括本部

経歴 :

前職ではSIerにて、オンプレミス・クラウドでの構築/運用を担当。

主な業務内容はインフラ設計/構築/試験/運用/移行/追加構築計画・実作業と幅広く担当し、  
オンコール対応や復旧のための調査・データセンターでのオペレーションも経験。

好きなAWSサービス :

Amazon CloudWatchをはじめとしたObservability関連サービス



Amazon  
CloudWatch

# 本セミナーの対象者

これからAWSで監視やObservability（可観測性）を始めたい方

Amazon CloudWatchの概要を把握したい方

※ Amazon CloudWatchの個別の機能については別途Blackbeltでご紹介予定です。

今後のBlackbeltリリースをご確認下さい。

# アジェンダ

1. モニタリング・オブザーバビリティとは？
2. AWSにおけるObservability全体像
3. Amazon CloudWatch全体像
4. 次に何をしたらよいの？

# 1. モニタリング・オブザーバビリティ とは？

# オブザーバビリティ（可観測性）とは



- システムで何がなぜ起こっているかといった動作状況を 把握できている状態のこと
  - 多くの場合、システムを計測（インストルメント）してメトリクス、ログ、またはトレースを収集することによって行われる
- システムがどのように動作しているかを理解することで、運用上の優秀性・ビジネス目標を達成することにも役立つ
  - 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

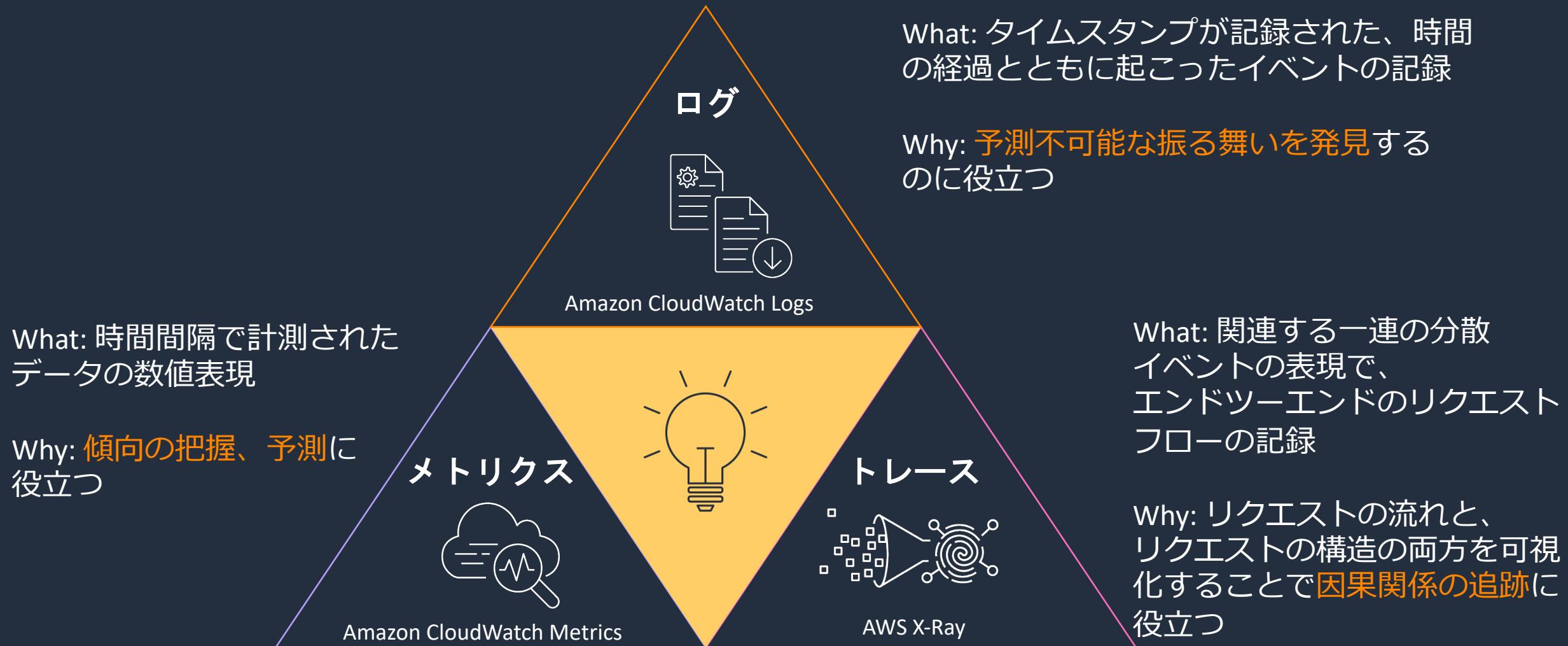
## ※モニタリング（監視）との違い

明確な定義があるわけではなく、様々な解釈がある。

監視はエラーに対応するためにシステムが機能しているかを検知・把握することを主目的としている場合が多い。

オブザーバビリティはシステムでの計測項目を継続的に追加し、新たなインサイトを得ることで、未知の障害の対処、ダウンタイムの短縮に活かし、システム障害以外のビジネス目標達成の用途にも役立てる。

# システムの状態把握に必要な3本柱

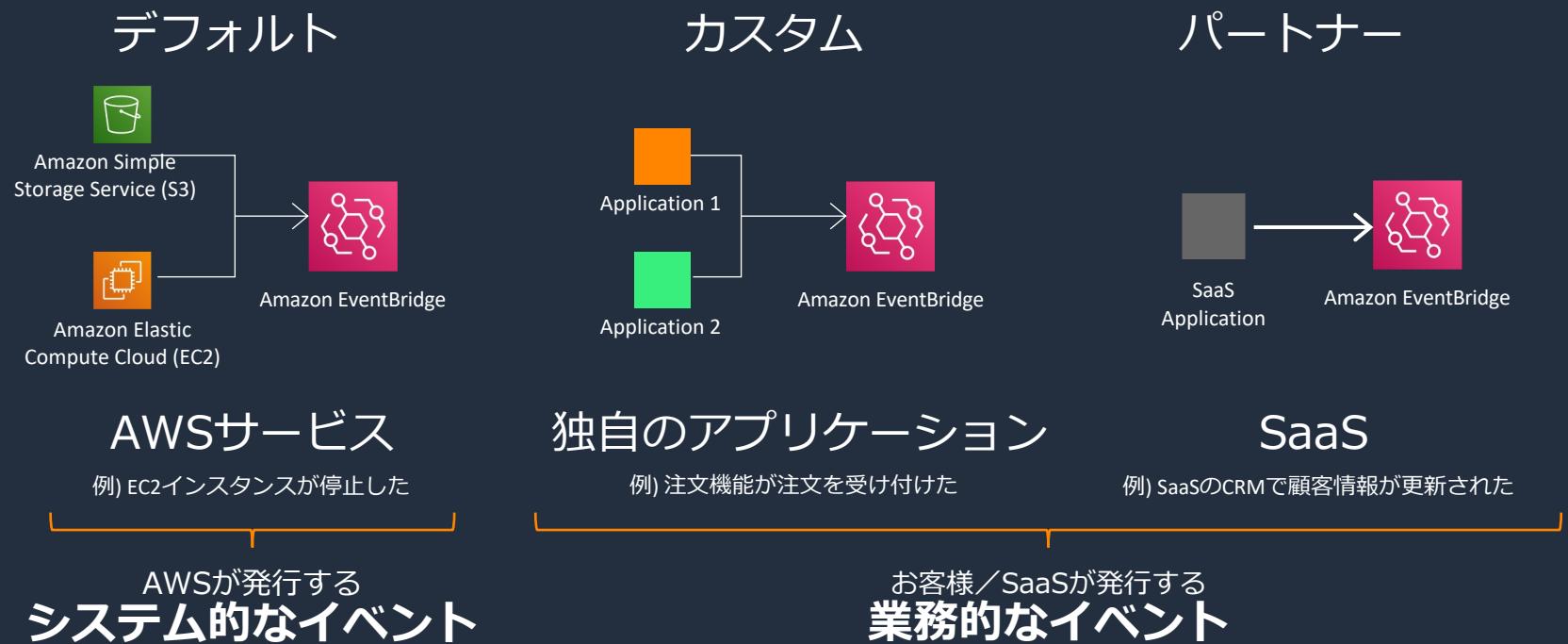


# ログ・メトリクス・トレース+“イベント”?



イベント=状態の変化。  
モニタリングと業務機能双方で活用される

モニタリングでは障害となるイベントが起きたら、リトライ・自動復旧・通知が適切に行えることが重要



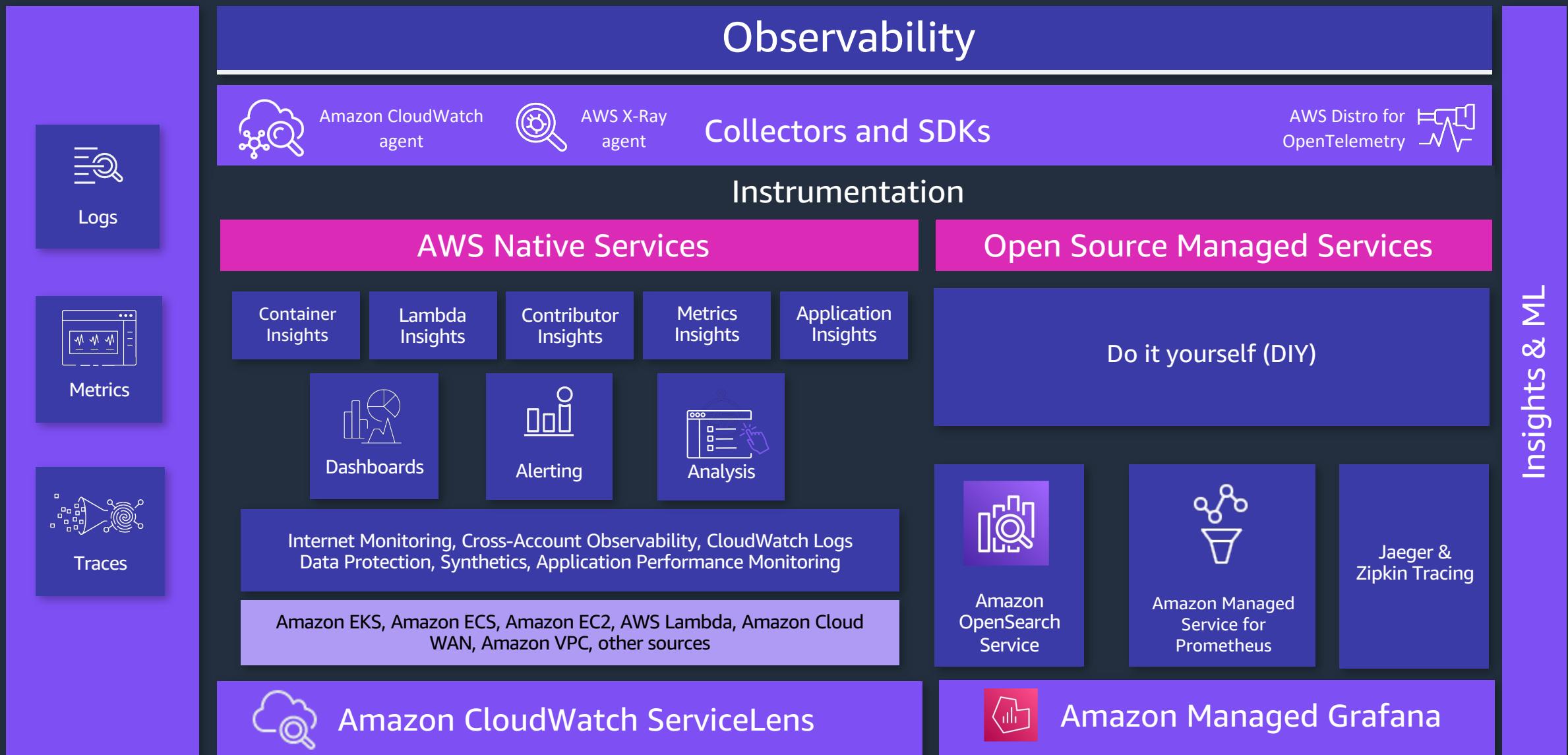
※Amazon EventBridgeはCloudWatch Eventsから派生・独立。  
イベントが発生するとターゲットにイベントを送信するが、AWSサービスだけでなく、カスタマイズができるようになっており、パートナーサービスとも連携できる。



## 2. AWSにおけるObservability全体像

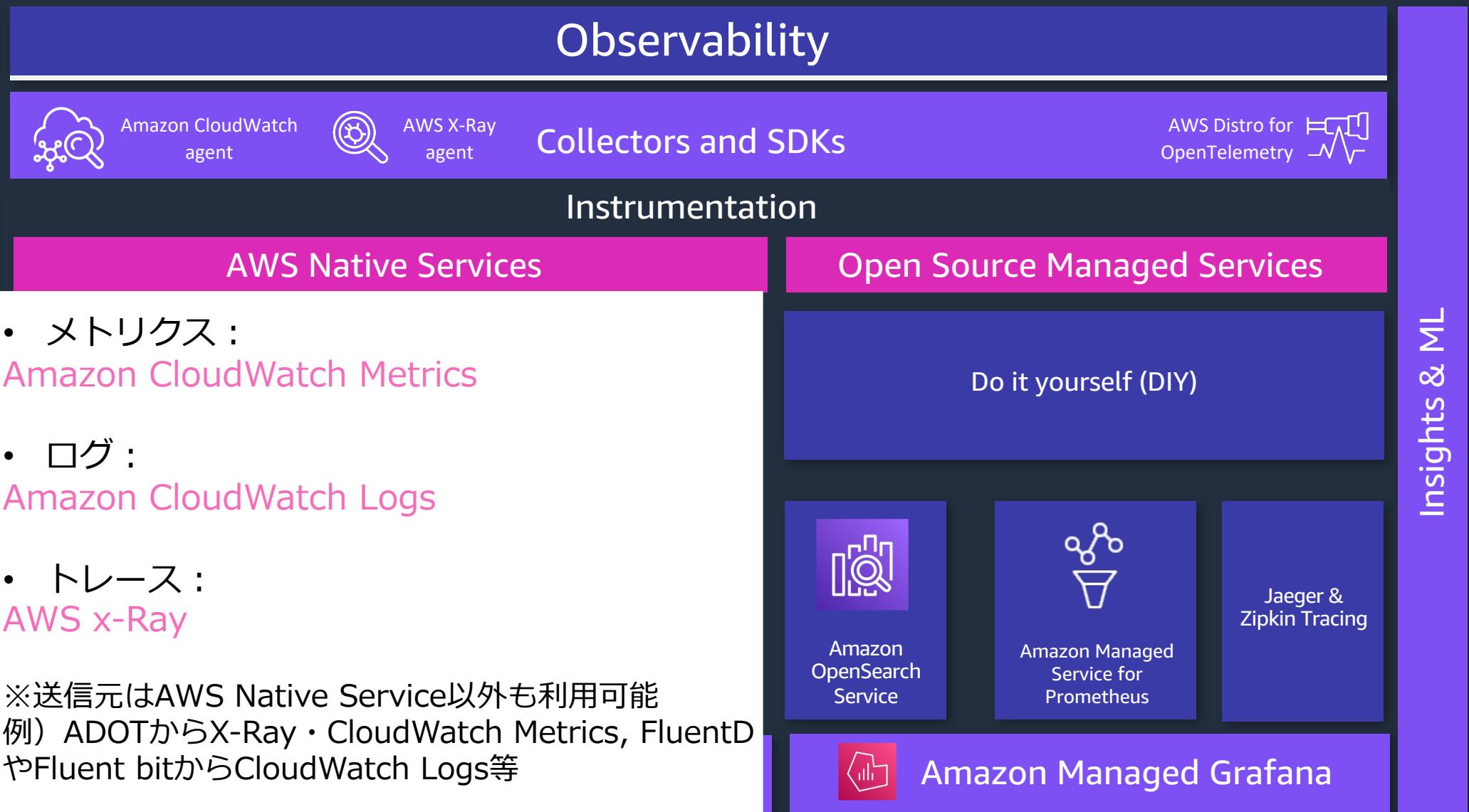
# AWS Observability

## Observability



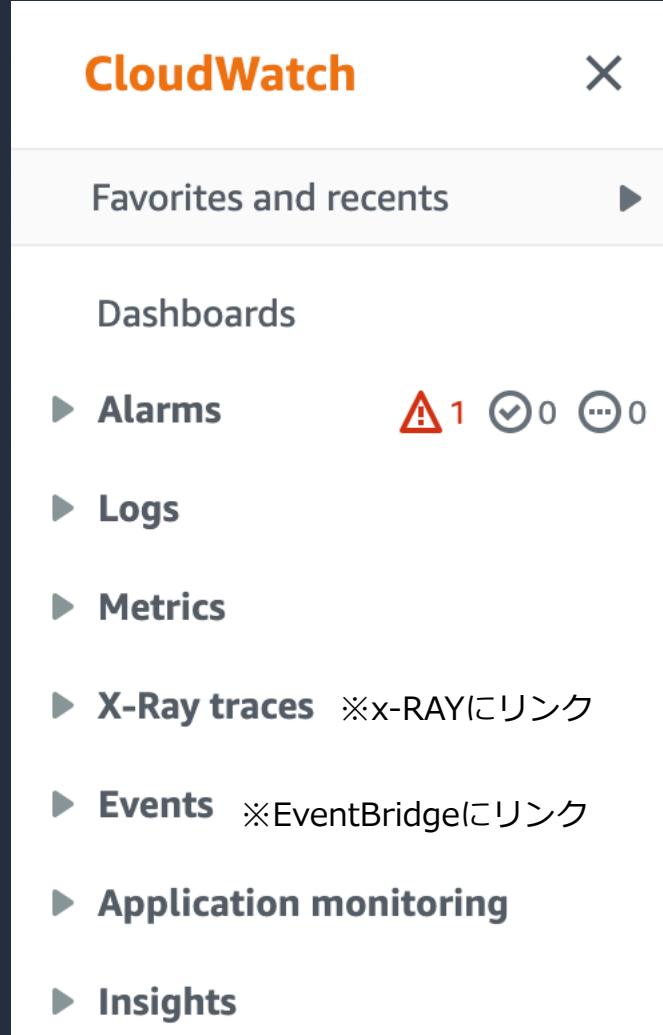
# AWS Observability

## Observability



# 3. Amazon CloudWatch全体像

# マネジメントコンソールでCloudWatchを検索



The screenshot shows the main content area of the CloudWatch Management Console. A large orange arrow points from the 'Logs' section of the sidebar towards the 'Logs' and 'Metrics' sections in the main content. Another large orange arrow points from the 'Metrics' section towards the 'Metrics' and 'Streams' sections in the main content. The main content area contains three main sections:

- Logs**: Log groups, Logs Insights
- Metrics**: All metrics, Explorer, Streams
- Application monitoring**: ServiceLens Map, Resource Health, Internet Monitor, Synthetics Canaries, Evidently, RUM

Below these sections is another section labeled 'Insights'.

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

# CloudWatchの機能一覧

## Infrastructure

AWSサービス名	概要
CloudWatch Metrics	メトリクス
CloudWatch Log	ログ
CloudWatch Alarm	アラーム
CloudWatch Dashboard	ダッシュボード
CloudWatch Metrics Explorer	メトリックス検索
CloudWatch Metrics Stream	メトリックスのリアルタイム連携
CloudWatch Events ※これはEvent Bridgeに統合されています	イベント
CloudWatch Resource Health	EC2の健全性・パフォーマンス可視化

## Application Monitoring

AWSサービス名	概要
CloudWatch Synthetics	外形監視
CloudWatch RUM	リアルユーザー モニタリング
CloudWatch Evidently	フィーチャーフラグA/Bテスト
CloudWatch Internet Monitor	インターネット監視
CloudWatch ServiceLens	トレース

コンソールからのメニューで  
19機能！

## Insights

AWSサービス名	概要
CloudWatch Contributor Insights	ログの時系列分析
CloudWatch Container Insights	コンテナ分析
CloudWatch Lambda Insights	Lambda分析
CloudWatch Application Insights	アプリケーション分析
CloudWatch Logs Insights	LOG分析
CloudWatch Metrics Insights	メトリクス分析

# Amazon CloudWatchの全体像

## Application Monitoring

- [外形監視]  
CW Synthetics
- [リアルユーザー モニタリング]  
CW RUM
- [フィーチャーフラグA/Bテスト]  
CW Evidently

- [インターネット監視]  
CW Internet Monitor
- [トレース]  
CW ServiceLens

[ダッシュボードに統合]  
CW Dashboard



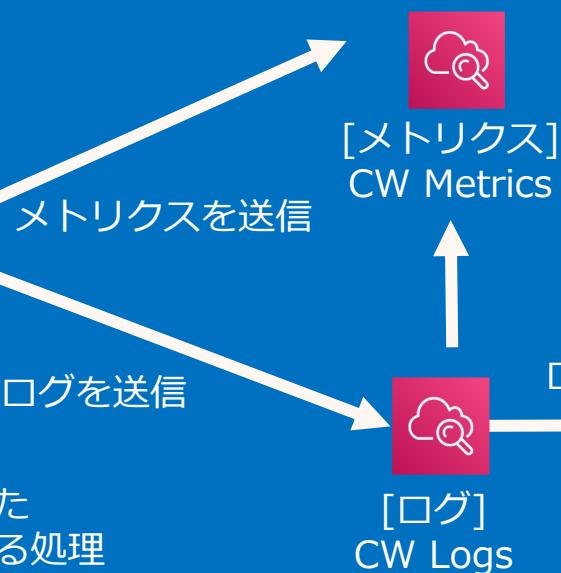
## Infrastructure



イベントに応じた  
ターゲットによる処理



[イベント]  
Amazon EventBridge/  
Amazon EventBridge Scheduler



[メトリクス]  
CW Metrics

[ログ]  
CW Logs

ログの可視化

[メトリクスストリーム]  
CW Metrics Stream

[ログ分析]  
CW Logs Insights

Amazon Kinesis  
Data Firehose

パートナー  
サービス、  
S3/Redshift



[アラーム]  
CW Alarms

メトリクスに応じた  
アクション



[リアルタイムメトリクス分析]  
CW Metrics Insight  
[タグベースの視覚化]  
CW Metrics Explorer

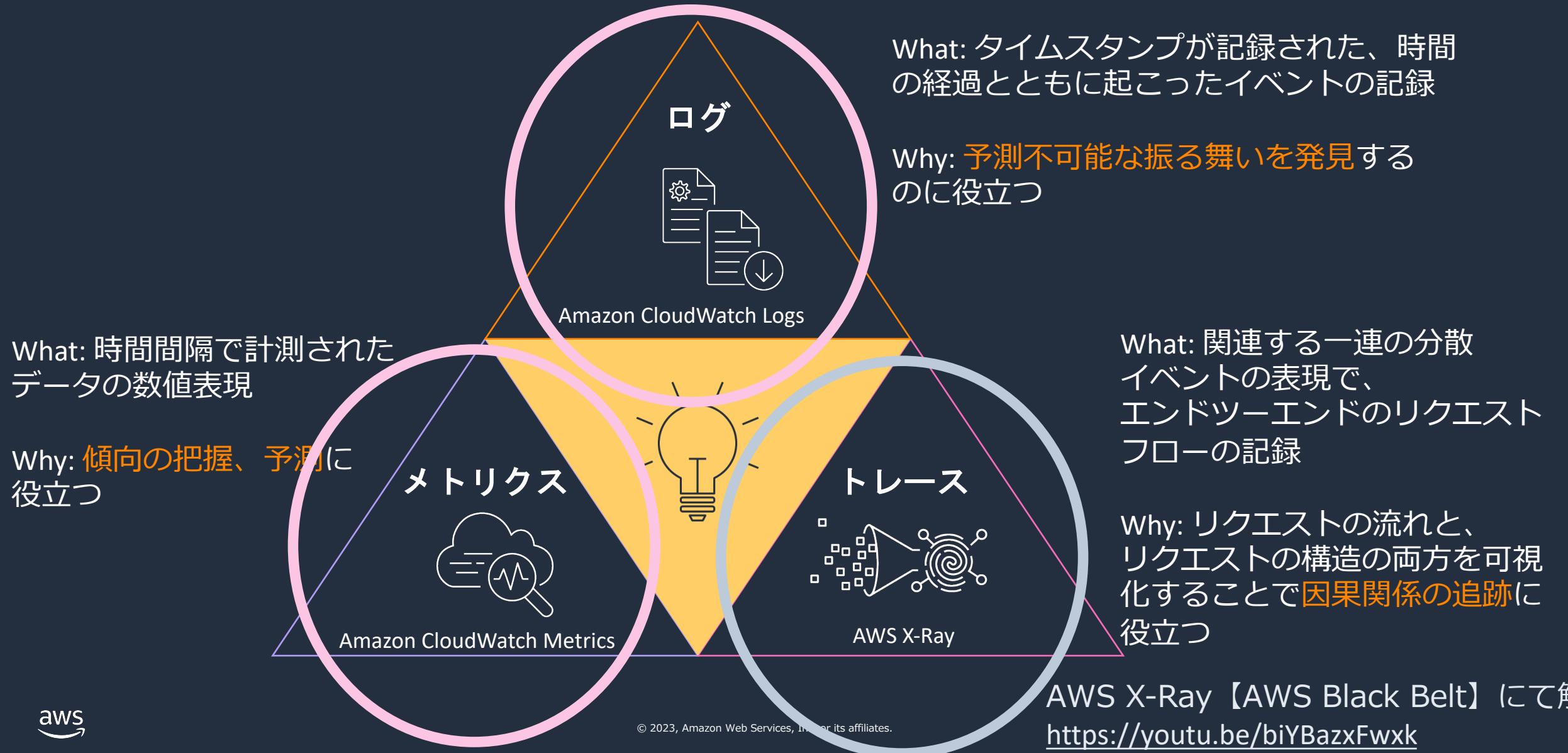
## Insights



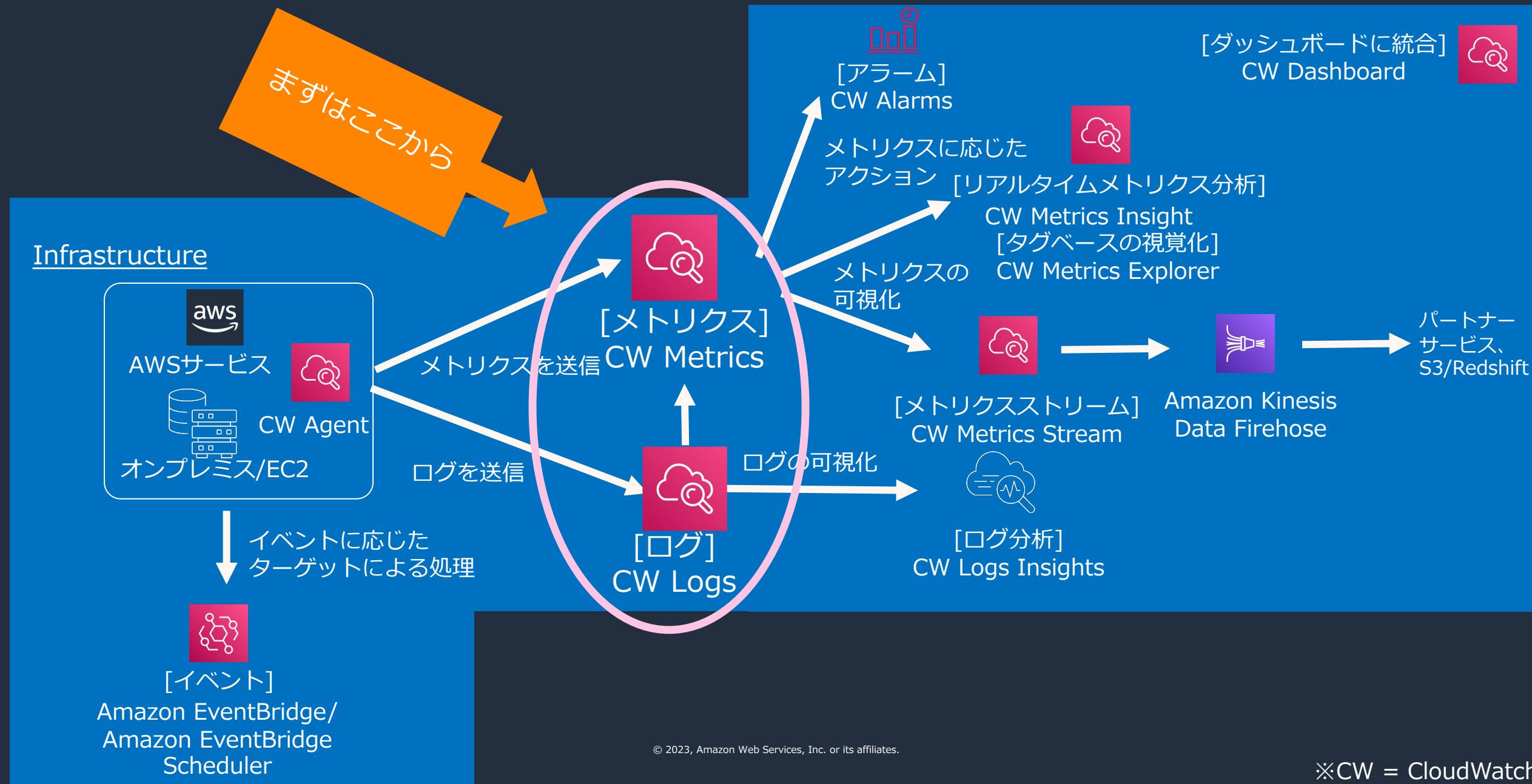
- [構造化ログによるメトリクス]  
CW Container Insights / Contributor Insights
- [Lambda拡張機能によるメトリクス]  
Lambda Insights
- [アプリケーションコンポーネントのメトリクス]  
Application Insights

※CW = CloudWatch

# 再掲：システムの状態把握に必要な3本柱



# Amazon CloudWatchの全体像

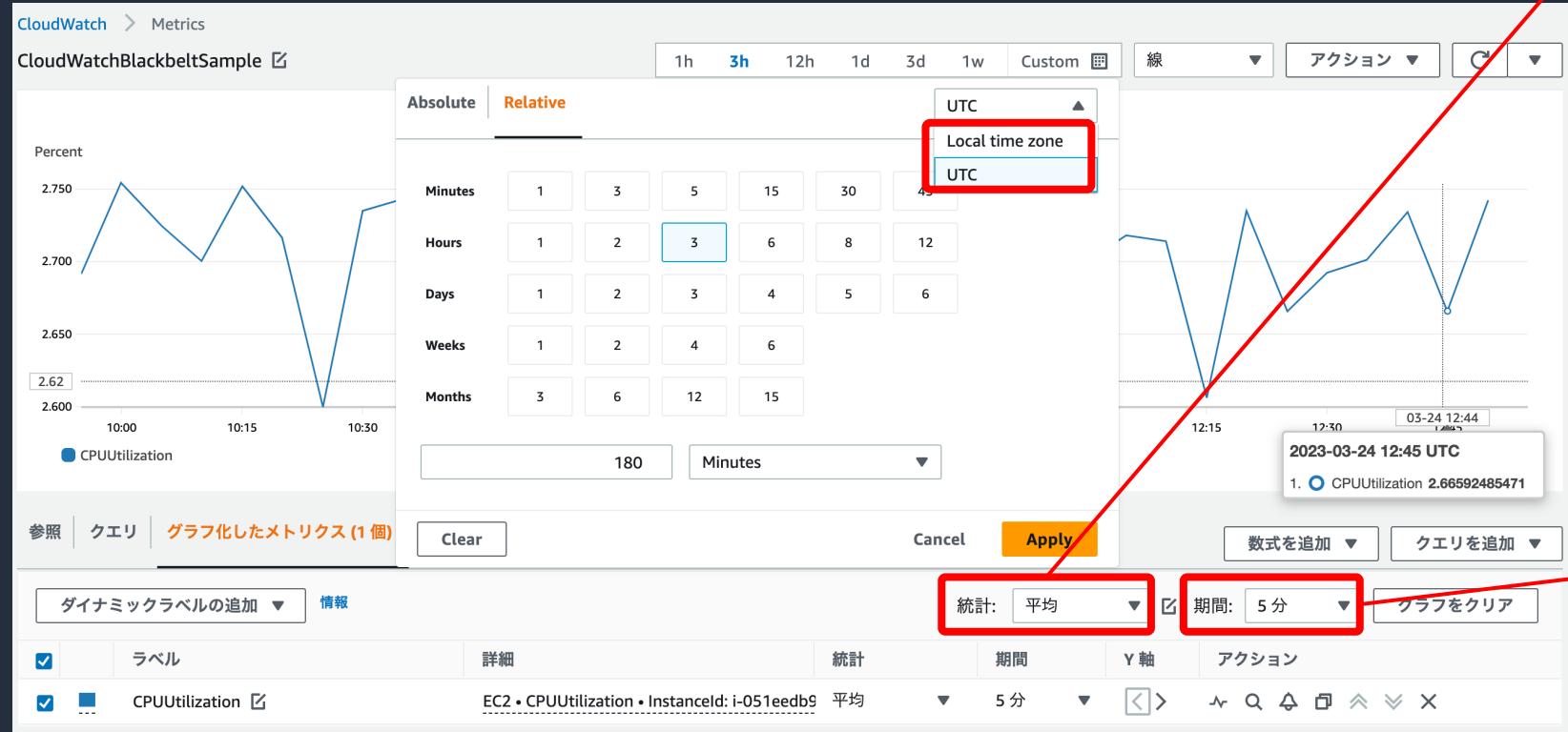


# メトリクスとは

時間間隔で計測されたデータの数値表現

期間や統計でサマリーされた情報から、傾向の把握に役立てる

## 例) EC2の標準メトリクス CPUUtilization



タイムゾーン、統計情報、統計の粒度を設定し表示



Standard
平均
最小
最大
合計
サンプル数

Expandable
PR(n:m)
TC(10%:90%)
TM(10%:90%)
TS(10%:90%)
WM(10%:90%)

※Trim値やPercentileはウェブページのレイテンシ計測など大きなサンプルサイズでネットワークの問題等で起こる外れ値を無視したい場合に活用

詳細はCloudWatch統計定義を参照

期間:
5 分
1 秒
5 秒
10 秒
30 秒
1 分
5 分
15 分
1 時間
6 時間
1 日
7 日
30 日

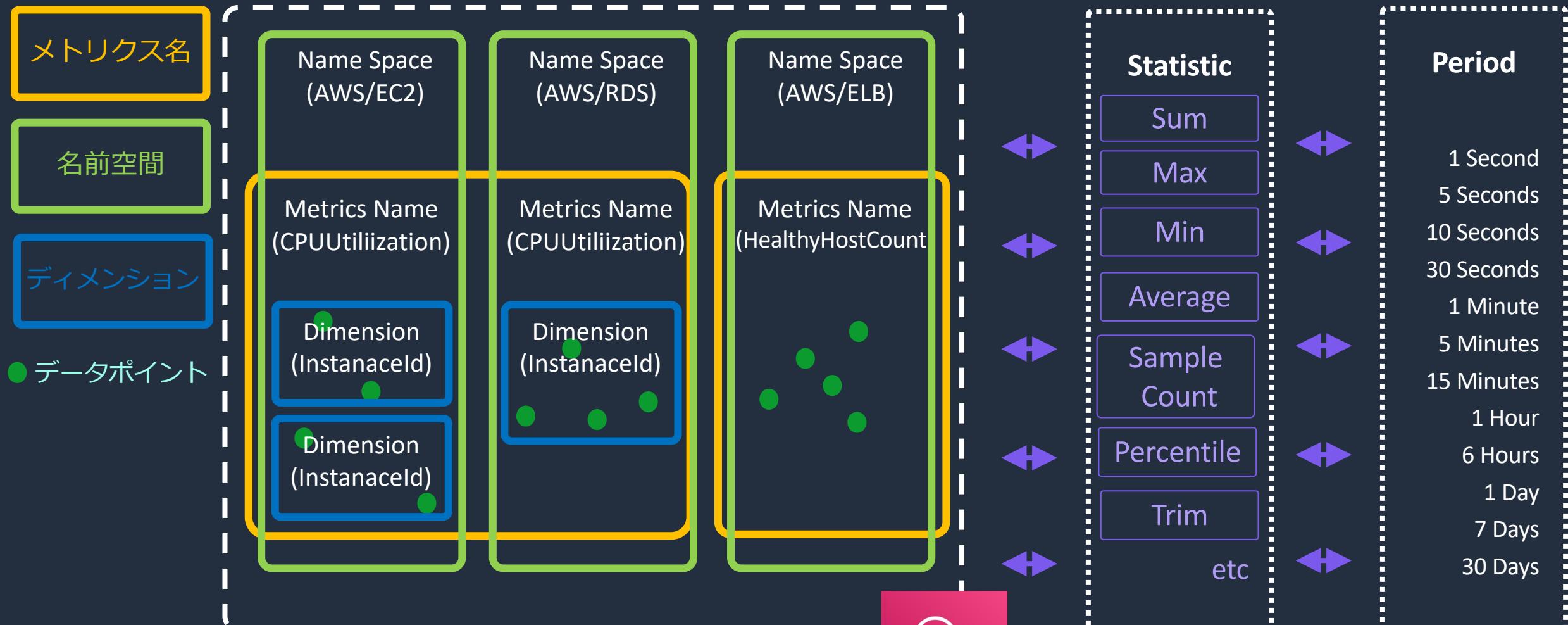
例えは・・・  
EC2のインスタンスマトリクスは標準のデフォルト5分間。(無料)  
詳細モニタリングの有効化で1分間の取得が可能。

高解像度のカスタムメトリクスを活用すれば1秒の解像度で期間を指定也可能。

※EC2でもステータスチェックメトリクスはデフォルト1分。(無料)

※高解像度のカスタムメトリクスはAPIやプラグインの設定が必要(参考Blog)

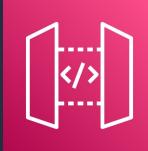
# CloudWatch Metricsの考え方



# 多くのAWSサービスでメトリクスやログを標準で発行

CloudWatch との統合により自動取得可能

## メトリクスの例



142サービスでメトリクス  
が自動取得可能  
※ 2023/03末時点

### Amazon API Gateway

- REST API: API request count, Latency, 4XXs, 5XXs, IntegrationLatency, CacheHitCount, CacheMissCount



### AWS Lambda

- Invocation count, Invocation errors, DeadLetterErrors, DestinationDeliveryFailures, Throttles など

## ログの例



30サービスでログが  
自動取得可能  
※ 2023/03末時点

### Amazon API Gateway

- REST API: **ERROR** と **INFO** の 2 レベルのログ
- HTTP API /WebSocket API: ログ変数でカスタマイズ可能



### AWS Lambda

- Lambda 関数の呼び出しの詳細、ログ
- コードから直接ログを取得: e.g. `console.log()`

※1 Amazon CloudWatch Metricsを発行するサービス  
<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/aws-services-cloudwatch-metrics.html>

※2 Lambda Insights/Container Insights/DynamoDB Contributor Insights/Amazon RDS Performance Insights等、有効化することで追加のMetricsが取得できるサービス多数

※3 Amazon CloudWatch Logs にログを発行する サービス  
[https://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/logs/aws-services-sending-logs.html](https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/logs/aws-services-sending-logs.html)

※4 AWS のサービスからのログの記録を有効にできるサービス  
[https://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/logs/AWS-logs-and-resource-policy.html](https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/logs/AWS-logs-and-resource-policy.html)

# CloudWatch Metric Insights

メトリクスのリアルタイム集計・可視化



メトリクスを効率よく分析

- 柔軟なクエリ機能により、リアルタイムでメトリクスの集計やグルーピング
- SQL ベース のクエリや選択式のビルダーで実行

例) ProductInfoService の名前空間に対して Operation ごとのエラー合計で Top5 を表示

**Metrics Insights - Query Editor**

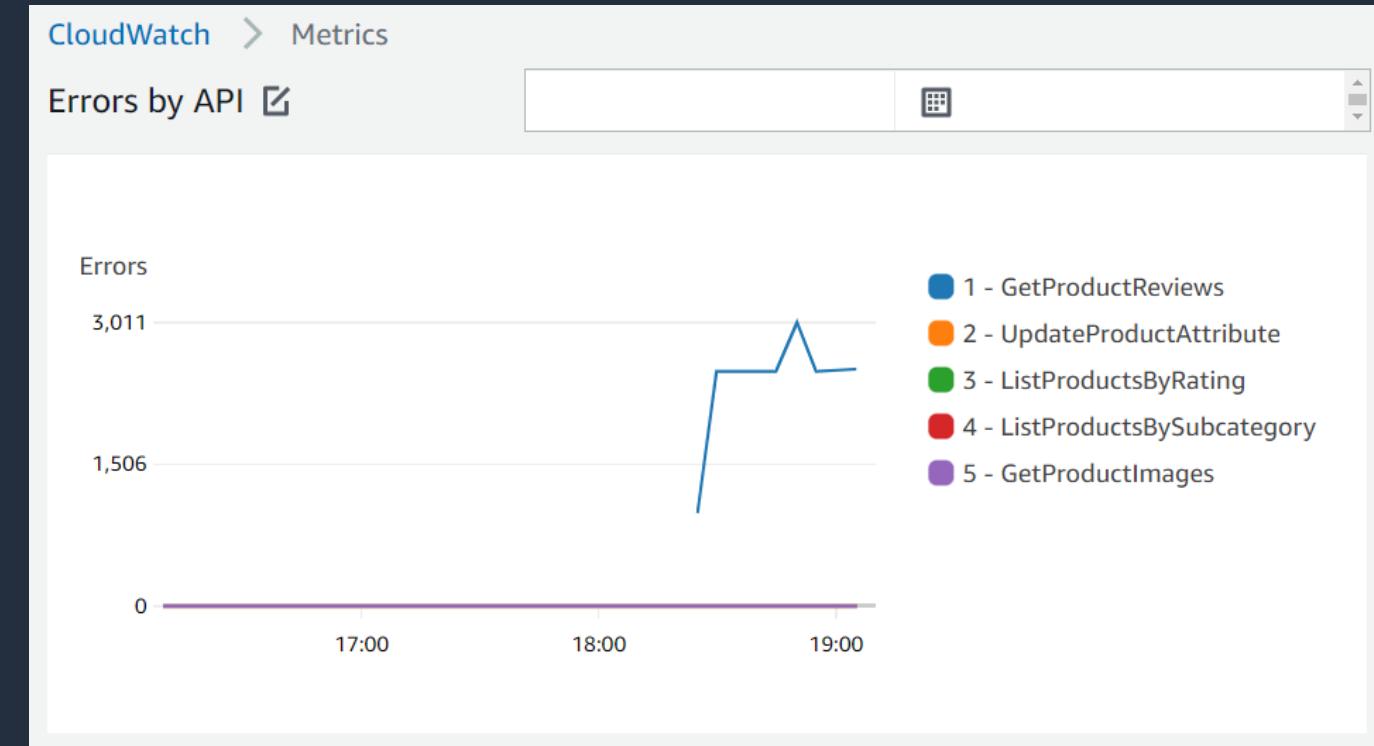
```

1 SELECT SUM(Errors)
2 FROM ProductInfoService
3 GROUP BY Operation
4 ORDER BY SUM() DESC
5 LIMIT 5
    
```

OR

**Metrics Insights - Query Builder**

名前空間	ProductInfoService	X
メトリクス名	SUM(Errors)	X
フィルター条件	Q ディメンションを選択 件 デフォルトでは、すべてのディメンションが選択されています	
グループ化条件	Operation	X
並べ替え条件	Q グループ化するディメンションを選択 件 メトリクスは常に 1 つの時系列に事前に集計されます。	
制限		5 X



### CloudWatch メトリクスに数式を使用して、新しいメトリクスを作成

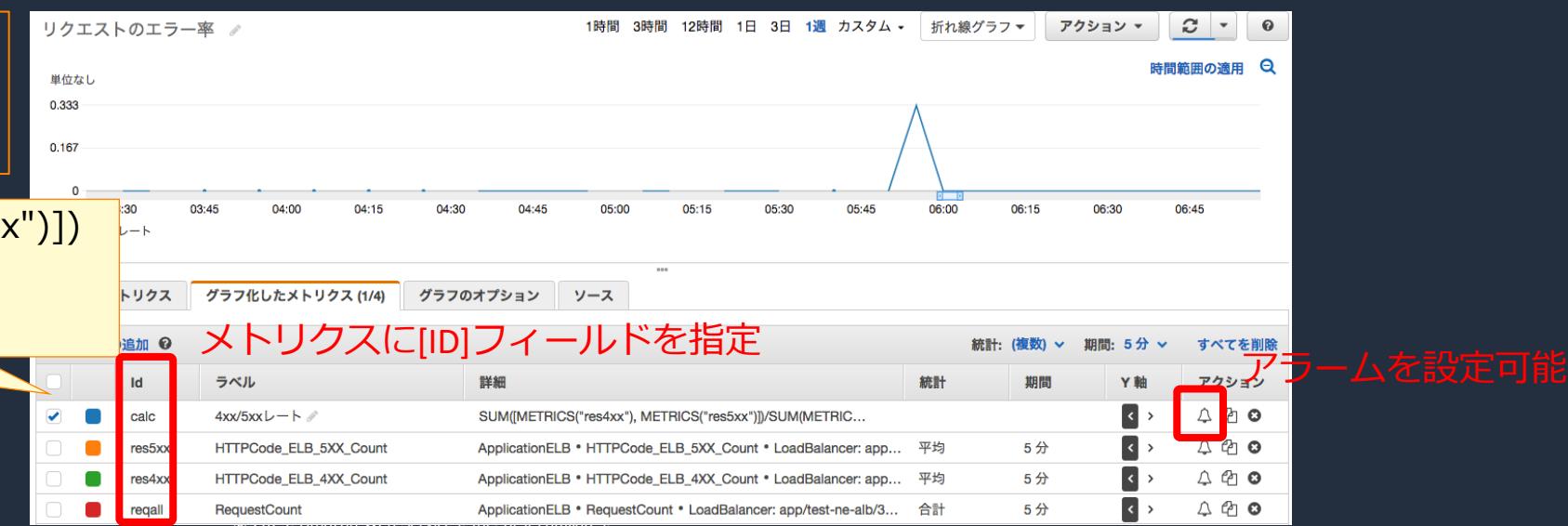
- METRICS関数、基本的な算術関数をはじめとした関数をサポート
- メトリクスに[Id]フィールドを設定し、関数で利用

コマンド	説明	例
AVG	データポイントの平均を表すスカラー	メトリクスの平均値 : AVG(METRICS())
SUM	データポイントの合計値を表すスカラー	メトリクスm1,m2の合計値 : SUM([m1,m2])
METRICS	CloudWatch メトリクスを表す	メトリクスreqall : METRICS("reqall")

etc

ユースケース  
全リクエストのうち4xx,5xx  
レスポンスの割合を表示

$\text{SUM}([\text{METRICS}("res4xx"), \text{METRICS}("res5xx")]) / \text{SUM}(\text{METRICS}("reqall"))$



# CloudWatch Alarmsでメトリクスをアクションにつなげる

アクション

## メトリクス



Amazon  
CloudWatch

## メトリクスアラームの状態

状態が変わったときに実行するアクションが指定できる

OK	ALARM	INSUFFICIENT_DATA
<b>正常値</b> ✓ 定義された閾値を下回っている	<b>異常値</b> ✓ 定義された閾値を上回っている	<b>判定不能</b> ✓ データ不足のため状態を判定できない ✓ CloudWatch特有



Amazon SNS



Amazon EC2



Amazon EC2 Auto Scaling



AWS Systems Manager OpsCenter



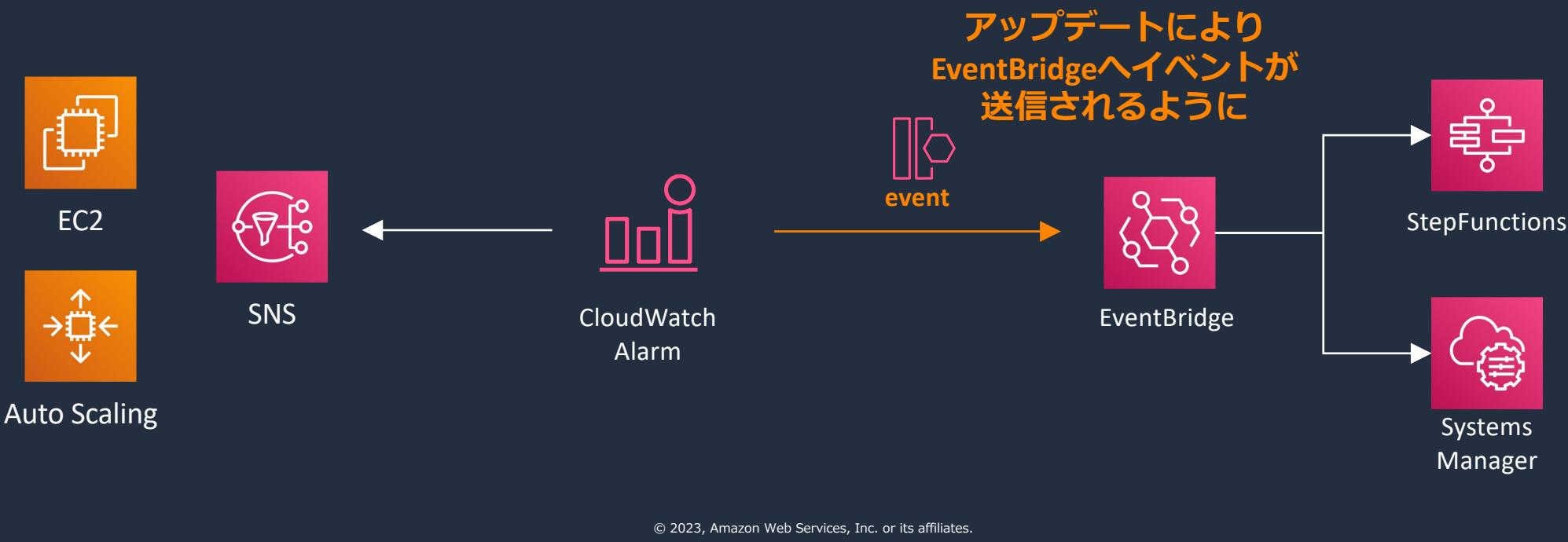
AWS Systems Manager Incident Manager

※Event Bridgeにも  
アラーム状態変化は  
配信される



# 参考：EventBridge でCloudWatch Alarmのイベントを取り扱う例

- CloudWatch Alarmのステータス変化がEventBridgeで取得可能。
- CloudWatch Alarmの機能では送信先としてEC2、SNS、AutoScaling、Systems ManagerでのOpsItem起票/Incident作成の操作のみが選択可能。
- Event Bridgeを介せばAlarmのイベントが**実装なしで様々なサービスに連携可能**



アラームを検知し、メトリクスを確認すると  
2時過ぎにインスタンスAでErrorが多かったようだ・・・



原因を特定するには？？

ログを見てみましょう

**Everything fails, all the time.**備えは忘れずに。基本的には自動復旧が大切！！

- EC2のSystem Status Checkは失敗しうる。原因を特定することは、サービスレベル向上につながらない。
- アラームのアクションやAmazon Event Bridgeの他にも、AWS Step Functions/Amazon SQS/Amazon SNSなどでリトライや障害時のデータの一貫性確保をコントロールしてもよい。

# ログとは

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  
システムで何が起こっているか、振る舞いを解析するのに役立てる

CloudWatch > ロググループ > /ecs/PayForAdoption > logs/container/eb7c8084084c4cc19b3571add02f797f

ログイベント

下のフィルターバーを使用して、ログイベント内の用語、語句、値の検索や照合ができます。 [フィルターパターンの詳細](#)

アクション ▾ メトリクスフィルターを作成

検索 イベントをフィルター クリア 1m 30m 1h 12h カスタム 表示 ▾

ログイベント一覧

タイムスタンプ	メッセージ
ロードする古いイベントがあります。 <a href="#">さらにロードします。</a>	
2023-03-28T12:43:06.562+09:00	POST /api/home/cleanupadoptions HTTP/1.1 11.0.12.188:1638 200 POST /api/home/cleanupadoptions HTTP/1.1 11.0.12.188:1638 200
2023-03-28T12:43:06.562+09:00	{"caller": "middlewares.go:84", "err": null, "method": "In CleanupAdoptions", "took": "34.438289ms", "traceId": "..."} {"caller": "repository.go:64", "repo": "sql", "sql": "\n\t\tINSERT INTO transactions (pet_id, transaction_id, ..."} {"caller": "repository.go:124", "method": "UpdateAvailability", "repo": "sql", "success": "(MISSING)", "ts": "..."} {"PetId": "001", "PetType": "puppy", "caller": "middlewares.go:57", "customer": {"ID": 1679974987142511795, "F..."} POST /api/home/completeadoption?petId=001&petType=puppy HTTP/1.1 11.0.105.43:54362 200
2023-03-28T12:43:07.142+09:00	{"caller": "repository.go:64", "repo": "sql", "sql": "\n\t\tINSERT INTO transactions (pet_id, transaction_id, ..."} {"caller": "repository.go:124", "method": "UpdateAvailability", "repo": "sql", "success": "(MISSING)", "ts": "..."} {"PetId": "001", "PetType": "puppy", "caller": "middlewares.go:57", "customer": {"ID": 1679974987142511795, "F..."} POST /api/home/completeadoption?petId=001&petType=puppy HTTP/1.1 11.0.105.43:54362 200
2023-03-28T12:43:07.303+09:00	{"caller": "repository.go:64", "repo": "sql", "sql": "\n\t\tINSERT INTO transactions (pet_id, transaction_id, ..."} {"caller": "repository.go:124", "method": "UpdateAvailability", "repo": "sql", "success": "(MISSING)", "ts": "..."} {"PetId": "020", "PetType": "kitten", "caller": "middlewares.go:57", "customer": {"ID": 1679974987362062218, "F..."} POST /api/home/completeadoption?petId=020&petType=kitten HTTP/1.1 11.0.12.188:13162 200
2023-03-28T12:43:07.362+09:00	{"caller": "repository.go:64", "repo": "sql", "sql": "\n\t\tINSERT INTO transactions (pet_id, transaction_id, ..."} {"caller": "repository.go:124", "method": "UpdateAvailability", "repo": "sql", "success": "(MISSING)", "ts": "..."} {"PetId": "020", "PetType": "kitten", "caller": "middlewares.go:57", "customer": {"ID": 1679974987362062218, "F..."} POST /api/home/completeadoption?petId=020&petType=kitten HTTP/1.1 11.0.12.188:13162 200

ログはタイムスタンプと  
メッセージからなる。

使い方 :

トレースやメトリクスを参考に見るべきログを特定し、該当の時間の該当のリソースのログを確認することで何が起こっているか把握する

# CloudWatch Logsでログを活用する流れ

## 各種ログ を収集

- エージェントやログドライバ・サイドカー等でログ収集



EC2 AWS外のServer Fargate

- 統合CloudWatchエージェント
- awslogsログドライバー
- AWS Distro for OpenTelemetry (ADOT)
- Amazon ECS Firelens  
(Fluentd や Fluent Bitイメージを活用)
- その他SaaS/OSSのAgentやCollectorなど

- Amazon CloudWatch Logs にログを発行するサービス



CloudTrail Lambda RDS VPC Flow Log など  
他多数 ※1

- 設定によりAmazon CloudWatch Logs に追加でログを発行するサービス



API Gatewayアクセスログ Step Functions実行履歴 Route 53クエリログ など  
他多数 ※2



CloudWatch Logs



通知:  
CloudWatch Alarms



データ連携:  
Amazon Kinesis Data  
Firehose



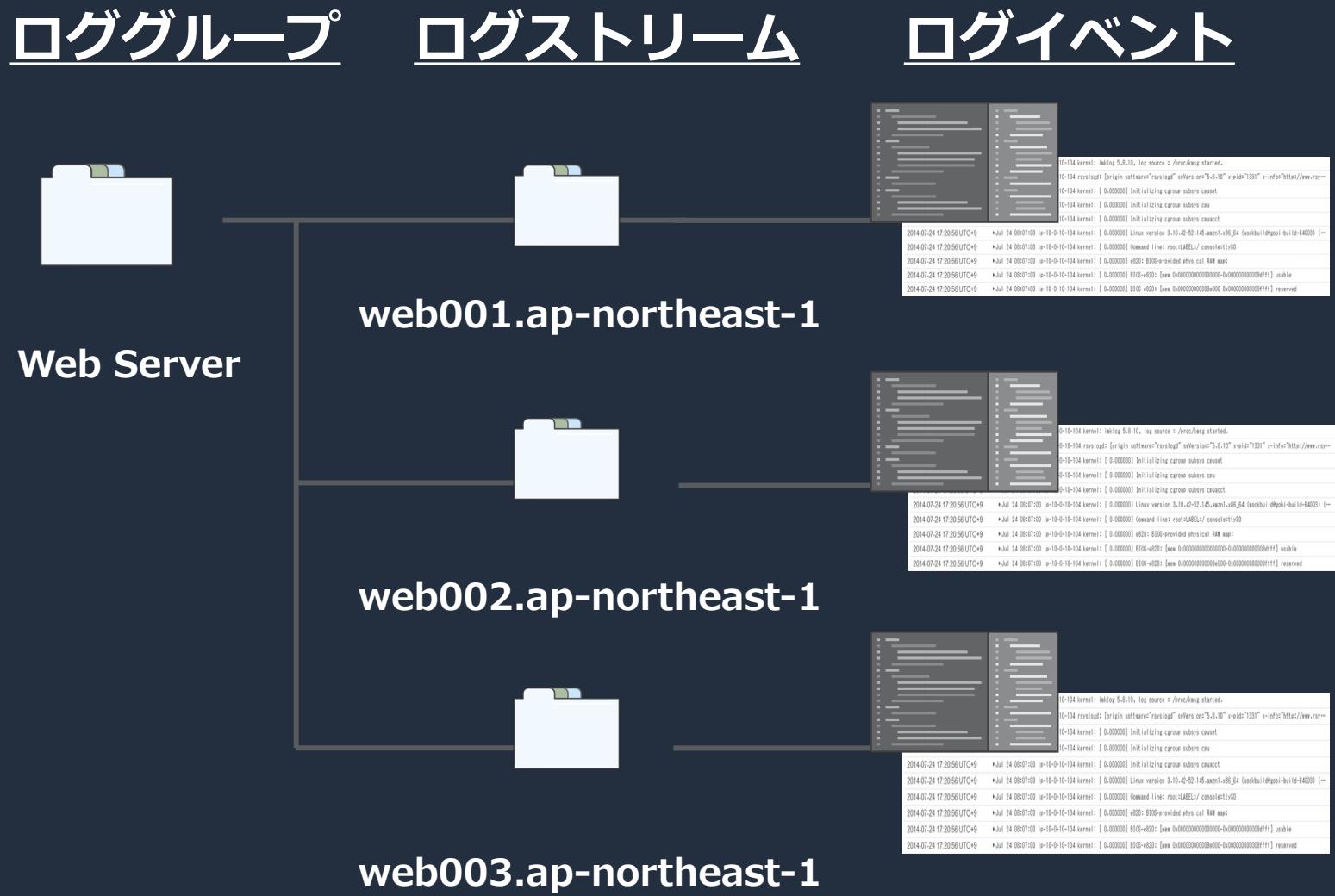
可視化:  
Amazon OpenSearch Service

※CloudFront・ALB・NLBのアクセスログなどはS3に直接出力されるなど、  
CloudWatch Logs以外に直接ログが出力されるケースもある。

※1 [https://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/logs/aws-services-sending-logs.html](https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/logs/aws-services-sending-logs.html)

※2 [https://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/logs/AWS-logs-and-resource-policy.html](https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/logs/AWS-logs-and-resource-policy.html)

# CloudWatch Logsの仕組み



# ログインイベント

- 1つのログエントリ
  - モニタリングしているリソースに  
よって記録されたアクティビティの  
レコード
  - イベント発生時のタイムスタンプお  
よび生のイベントメッセージで構成

# ログストリーム

- 同じソースを共有する一連のログイベント
  - モニタリングしているリソースのタイムスタンプ順でイベントを表す

# ロググループ

- 複数のログストリームで構成
  - 保持、監視、アクセス制御について同じ設定を共有するログストリームのグループを定義

# ロググループの設定

ログの保持期間・メトリクスフィルター・データ保護ポリシー・サブスクリプションフィルターはロググループごとに設定する。

The screenshot shows the AWS CloudWatch Log Groups interface. On the left, there's a sidebar with actions like 'Logs Insights で表示' (View in Logs Insights) and 'ロググループの検索' (Search Log Group). A red arrow points from the 'Actions' button to the '保持設定を編集' (Edit Retention Settings) option in the dropdown menu. Another red arrow points from the 'Logs Insights で表示' button to the 'Logs Insights' tab in the main content area.

**CloudWatch > ロググループ > /aws/lambda/C9-Observability-Workshop-C9DiskResizeLambda-SGRA3U9M4JRT**

**/aws/lambda/C9-Observability-Workshop-C9DiskResizeLambda**

アクション ▲ Logs Insights で表示 ロググループの検索

ロググループの削除  
保持設定を編集  
メトリクスフィルターを作成  
データ保護ポリシーを作成  
サブスクリプションフィルター ▶  
寄稿者インサイトルールを作成  
データを Amazon S3 にエクスポート  
Amazon S3 へのすべてのエクスポートを表示  
保持

メトリクスフィルター  
Amazon OpenSearch Service サブスクリプションフィルターを作成  
Kinesis サブスクリプションフィルターを作成  
Kinesis Firehose サブスクリプションフィルターを作成  
Lambda サブスクリプションフィルターを作成  
すべてのサブスクリプションフィルターを削除

データ  
データを非表示

保持期間の設定

次の期間経過後にイベントを失効:  
失効しない  
失効しない (selected)  
1 日  
3 日  
5 日  
1 週間 (7 日)  
2 週間 (14 日)  
1 か月 (30 日)  
2 months (60 日)  
3 months (90 日)  
4 months (120 日)  
5 months (150 日)  
6 months (180 日)  
12 months (365 日)  
13 months (400 日)  
18 months (545 日)  
2 年 (731 日)  
3 年 (1096 日)  
5 年 (1827 日)  
6 年 (2192 日)  
7 年 (2557 日)  
8 年 (2922 日)  
9 年 (3288 日)  
10 年 (3653 日)

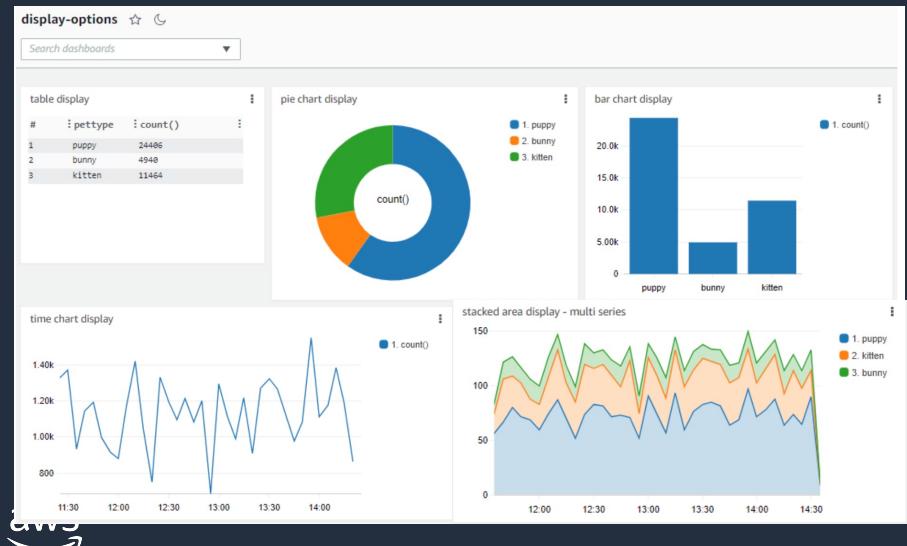
無期限に保存することも可能

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

# CloudWatch Logs Insights

## ログから実用的な洞察を導き出す

- CloudWatch Logs でログデータをインタラクティブに検索・分析
- 専用のクエリ言語
- 時系列データの可視化、個々のログイベントへのドリルダウン、クエリ結果の CloudWatch ダッシュボードへのエクスポート



ログを効率よく分析

例) 1時間あたりの例外発生件数の一覧を取得する

クエリ

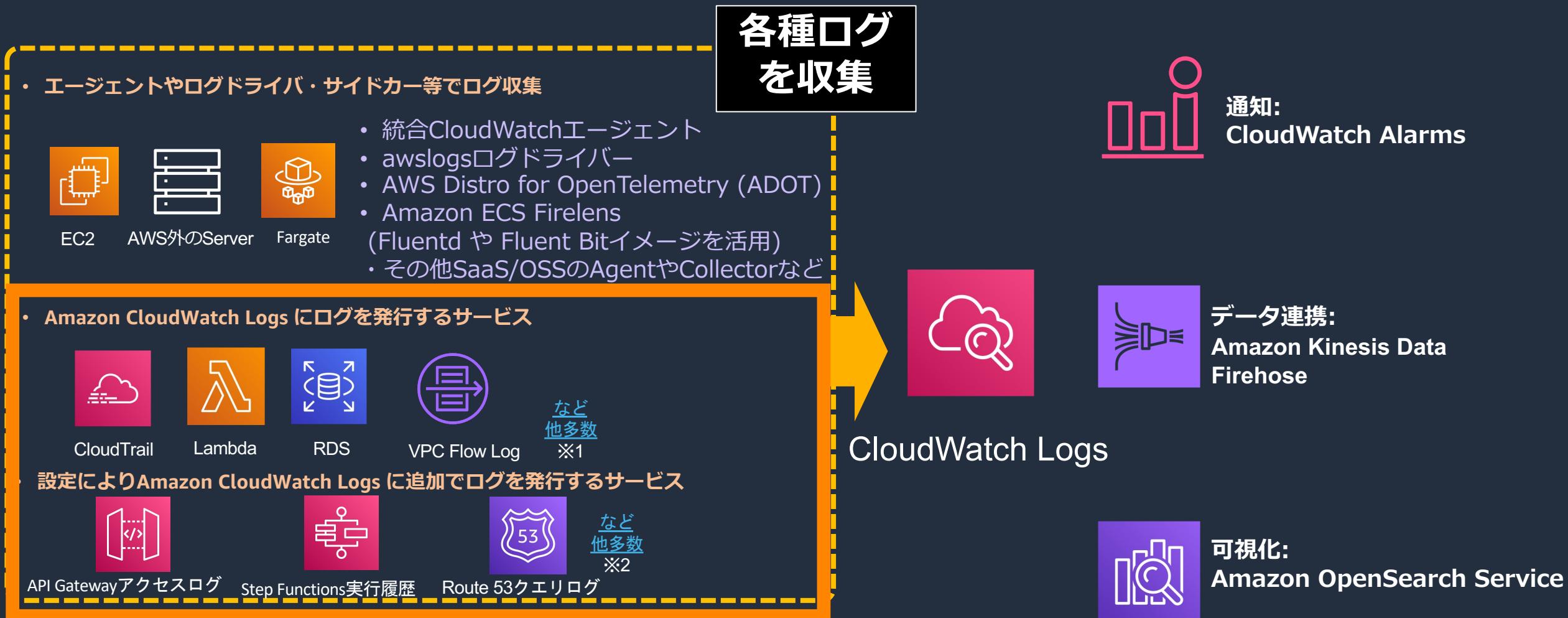
```
1 filter @message like /Exception/
2   | stats count(*) as exceptionCount by bin(1h)
3   | sort exceptionCount desc
```

クエリ結果

#	bin(1h)	exceptionCount
1	2022-07-17T20:00:00.000...	2194
2	2022-07-17T21:00:00.000...	238

CloudWatch Logs Insightsではログデータの検索・分析だけでなく、bin() 関数を使用するクエリを実行して、返された値を期間別にグループ化すると、結果を折れ線グラフ、円グラフ、棒グラフ、積み上げ面グラフとして表示可能。

# 再掲：CloudWatch Logsでログを活用する流れ



※CloudFront・ALB・NLBのアクセスログなどはS3に直接出力されるなど、CloudWatch Logs以外に直接ログが出力されるケースもある。

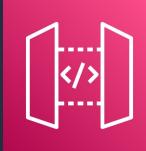
※1 [https://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/logs/aws-services-sending-logs.html](https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/logs/aws-services-sending-logs.html)

※2 [https://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/logs/AWS-logs-and-resource-policy.html](https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/logs/AWS-logs-and-resource-policy.html)

# 再掲：多くのAWSサービスでメトリクスやログを標準で発行

## CloudWatch との統合により自動取得可能

### メトリクスの例



142サービスでメトリクス  
が自動取得可能  
※ 2023/03/30時点

#### Amazon API Gateway

- REST API: API request count, Latency, 4XXs, 5XXs, IntegrationLatency, CacheHitCount, CacheMissCount



#### AWS Lambda

- Invocation count, Invocation errors, DeadLetterErrors, DestinationDeliveryFailures, Throttles など

### ログの例



30サービスでログが  
自動取得可能  
※ 2023/03/30時点

#### Amazon API Gateway

- REST API: ERROR と INFO の 2 レベルのログ
- HTTP API /WebSocket API: ログ変数でカスタマイズ可能



#### AWS Lambda

- Lambda 関数の呼び出しの詳細、ログ
- コードから直接ログを取得: e.g. console.log()

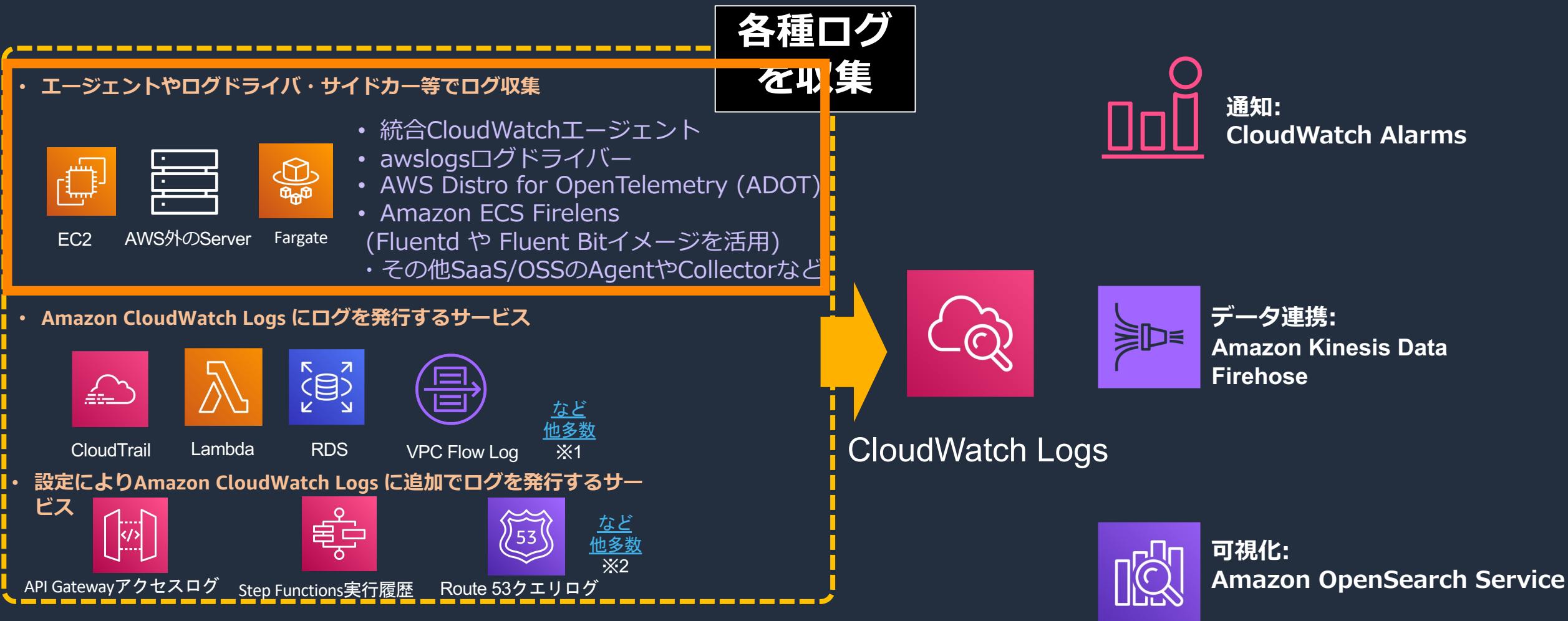
※1 Amazon CloudWatch Metricsを発行するサービス  
<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/aws-services-cloudwatch-metrics.html>

※2 Lambda Insights/Container Insights/DynamoDB Contributor Insights/Amazon RDS Performance Insights等、有効化することで追加のMetricsが取得できるサービス多数

※3 Amazon CloudWatch Logs にログを発行する サービス  
[https://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/logs/aws-services-sending-logs.html](https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/logs/aws-services-sending-logs.html)

※4 AWS のサービスからのログの記録を有効にできるサービス  
[https://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/logs/AWS-logs-and-resource-policy.html](https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/logs/AWS-logs-and-resource-policy.html)

# 再掲：CloudWatch Logsでログを活用する流れ



※CloudFront・ALB・NLBのアクセスログなどはS3に直接出力されるなど、CloudWatch Logs以外に直接ログが出力されるケースもある。

※1 [https://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/logs/aws-services-sending-logs.html](https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/logs/aws-services-sending-logs.html)

※2 [https://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/logs/AWS-logs-and-resource-policy.html](https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/logs/AWS-logs-and-resource-policy.html)

# 3本柱のAWSサービスにObservability のデータをカスタムで収集する手段

- ・収集したい情報
- ・収集元の対応
- ・送信先の対応

今必要なもの・将来どの程度  
SaaS/OSS含めた送信先の追加が  
起こりうるかがTool選びのPoint

統合CloudWatch  
エージェント

ログ



Amazon CloudWatch Logs

メトリクス



Amazon CloudWatch Metrics

コンテナの場合のAWSが用意する選択肢

- awslogsログドライバー
- Fluentd や Fluent Bitイメージを活用
  - Amazon ECS Firelens
  - AWS for Fluent Bit image
  - Fluent Bit for Amazon EKS on AWS Fargate

etc

AWS Distro for  
OpenTelemetry (ADOT) ※1

X-Ray Agent

トレース

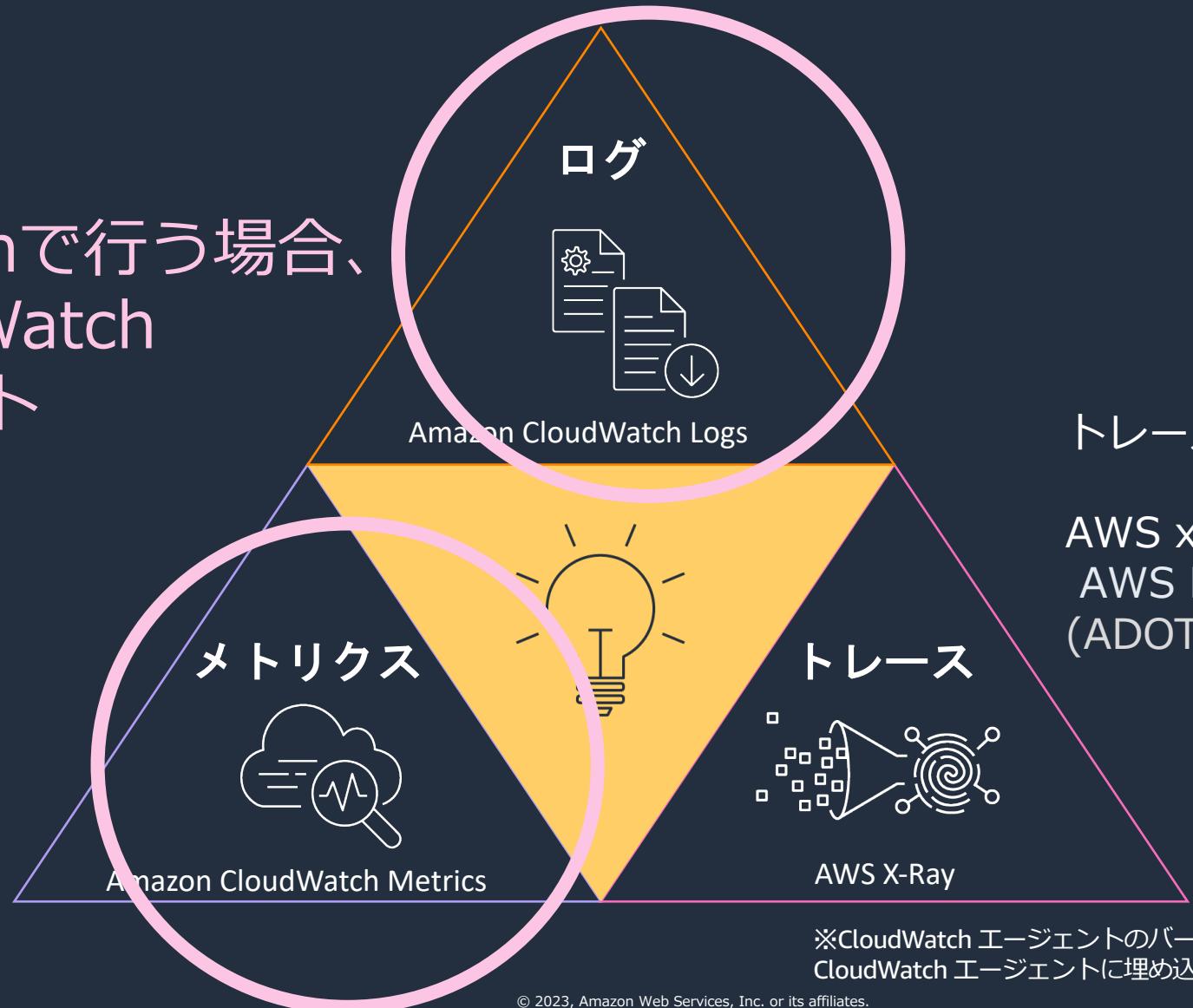


AWS X-Ray



# 3本柱のAWSサービスにObservability のデータをカスタムで収集する手段

Cloudwatchで行う場合、  
統合CloudWatch  
エージェント



SaaS

その他  
OSS

トレースはどうする？

AWS x-RAYにトレースを送るならば、  
AWS Distro for OpenTelemetry  
(ADOT)・X-Ray Agentが選択肢

※CloudWatch エージェントのバージョン 1.247356 以降、Open Telemetry Collector は  
CloudWatch エージェントに埋め込まれなくなりました。



# 統合CloudWatchエージェント

- CloudWatch Logs・Metrics双方に対応しているエージェント
- クラウドでもオンプレミスでも利用可能。
- Linux, Windowsの両方で稼働。
- 以前のCloudWatch Logs エージェントはサポートされていても非推奨※1

★インストール方法は大きく以下3パターン

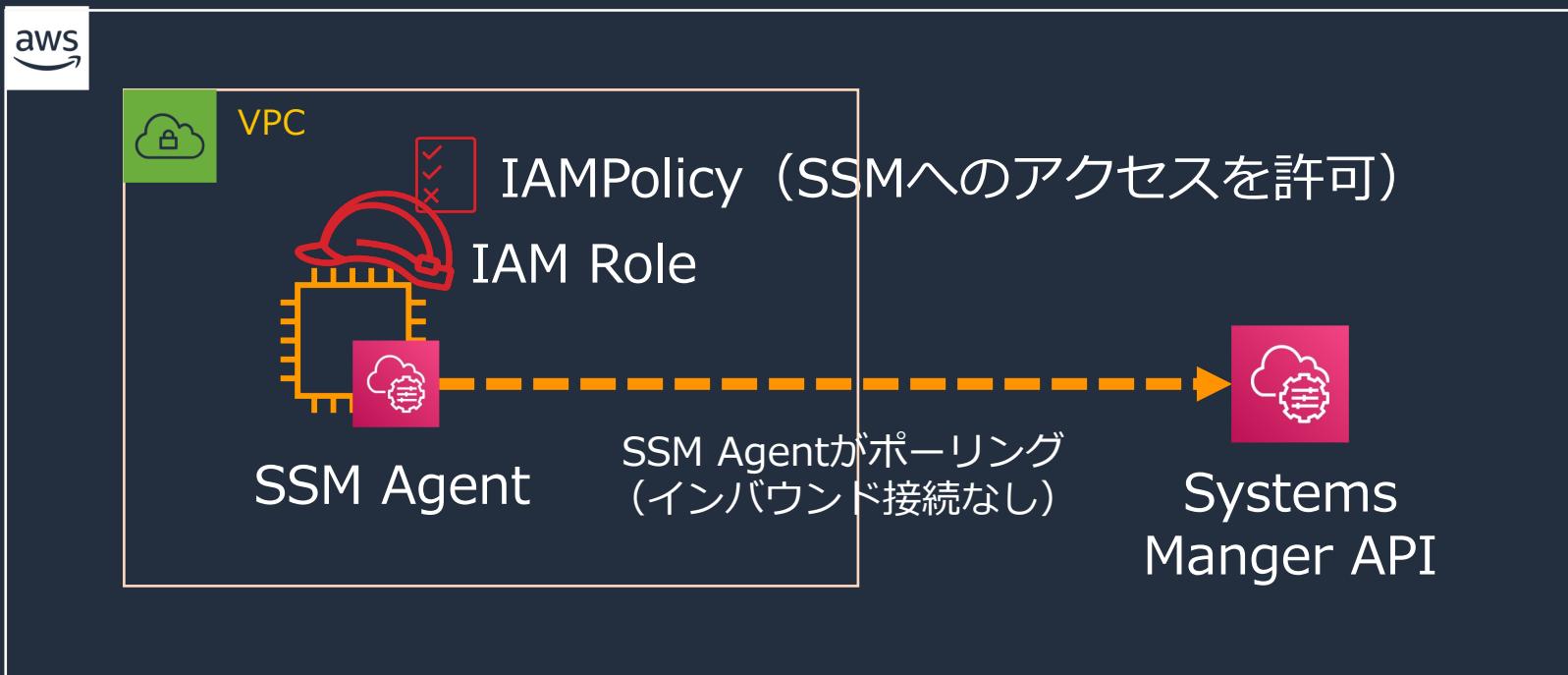
1. コマンドラインで実行（複数サーバーに行うには工夫が必要）→[コマンドラインを使用した CloudWatch エージェントのインストール](#)
2. SSMを活用（SSMなら多数のサーバーにも簡単導入）→[AWS Systems Manager を使用した CloudWatch エージェントのインストール](#)
3. IaCで（CFnで併せてAgentも導入）→[AWS CloudFormationを使用してCloudWatch エージェントのインストール](#)

※1 [https://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/logs/CWL\\_GettingStarted.html](https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/logs/CWL_GettingStarted.html)

# Systems Manager を活用した 統合CloudWatchエージェントインストール方法



# 手順1: Systems Manager Agent の導入



- AWSオフィシャルイメージには導入済みのため、この手順は不要 ※1
- 幅広い対応OS・バージョン一覧は以下URLを参照

<https://docs.aws.amazon.com/systems-manager/latest/userguide/prereqs-operating-systems.html>

–Linux (RHELなど商用含む) /macOS/Windows Serverなど多数

参考ドキュメント : SSM Agent のインストールと更新

[https://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/monitoring/download-CloudWatch-Agent-on-EC2-Instance-SSM-first.html](https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/monitoring/download-CloudWatch-Agent-on-EC2-Instance-SSM-first.html)

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

※1 2023年3月29日時点のSSM AgentがプリインストールされたAMI

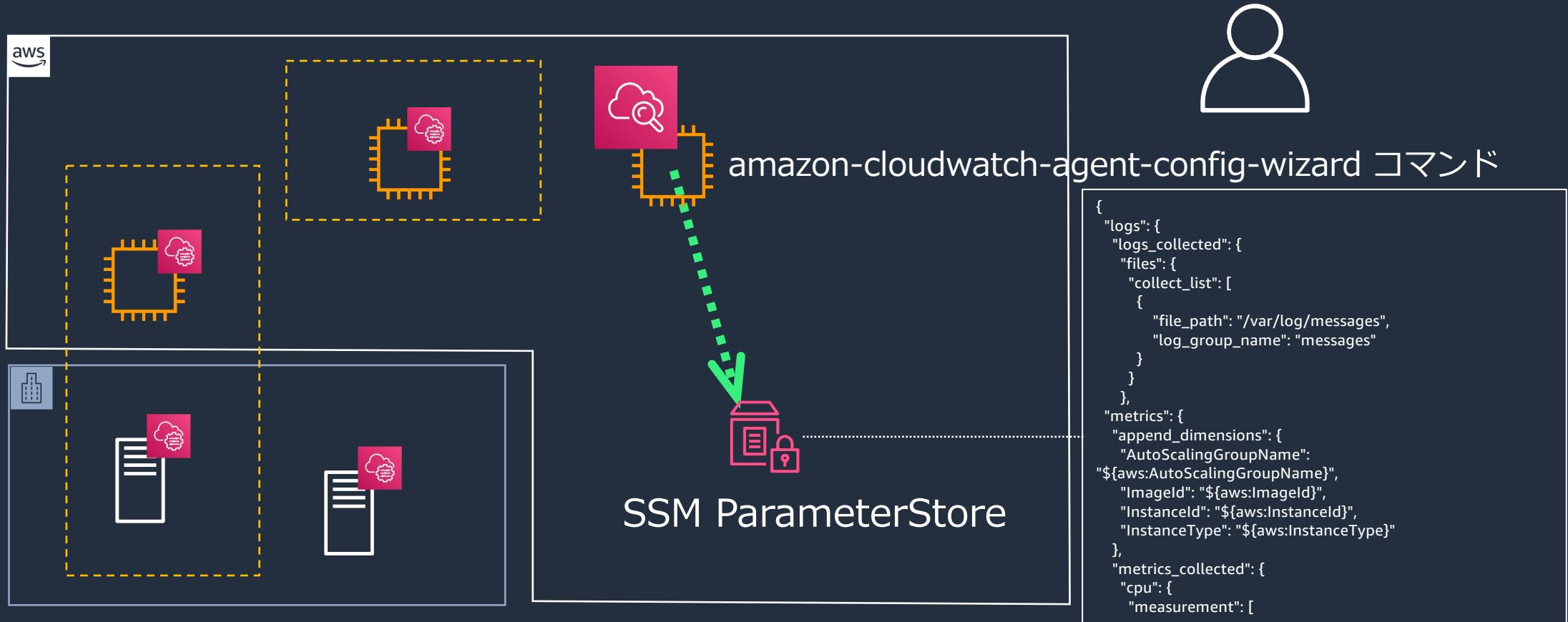
- 2017年9月以降のAmazon Linux Base AMI
- Amazon Linux 2
- Amazon Linux 2 ECSに最適化されたベース AMIs
- Amazon EKS 最適化 Amazon Linux AMIs
- macOS 10.14.x (Mojave)、10.15.x (Catalina)、11.x (Big Sur)
- SUSE Linux Enterprise Server(SLES) 12と15
- Ubuntu Server 16.04、18.04、および 20.04
- 2016年11月以降に公開されたWindows Server 2008-2012 R2 AMIs
- Windows Server 2016、2019、および 2022

[https://docs.aws.amazon.com/ja\\_jp/systems-manager/latest/userguide/ami-preinstalled-agent.html](https://docs.aws.amazon.com/ja_jp/systems-manager/latest/userguide/ami-preinstalled-agent.html)



## 手順2. 統合CloudWatch Agent の一括セットアップ (1/3)

- CloudWatch Agent の設定ファイルを生成し SSM ParameterStore へ格納
  - amazon-cloudwatch-agent-config-wizard コマンドを実行



参考ドキュメント : CloudWatch エージェント設定ファイルを作成する

[https://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/monitoring/create-cloudwatch-agent-configuration-file.html](https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/monitoring/create-cloudwatch-agent-configuration-file.html)



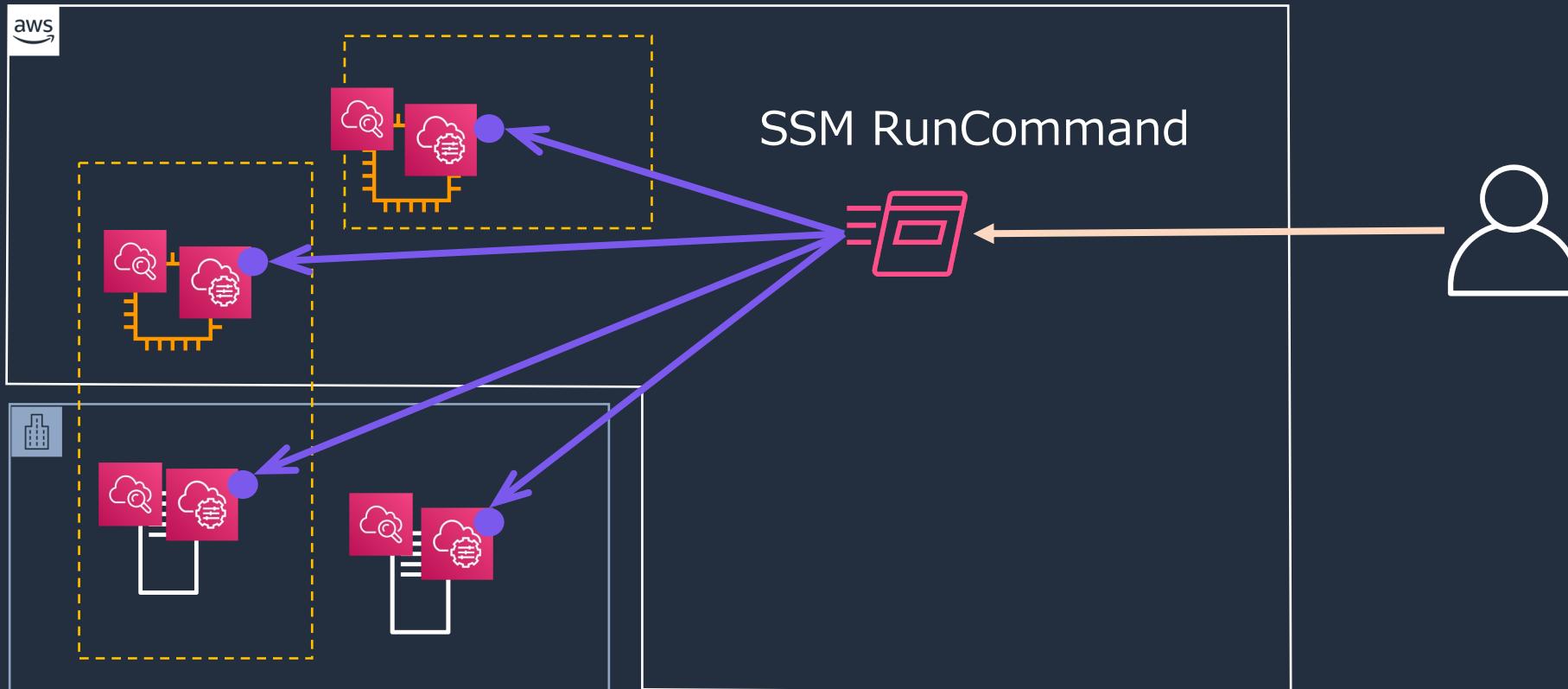
SSM Agent



CW Agent

## 手順2. 統合CloudWatch Agent の一括セットアップ (2/3)

2. RunCommand で 統合CloudWatch Agentを一括インストール
  - "AWS-ConfigureAWSPackage" ドキュメントを実行



参考ドキュメント : CloudWatch エージェントパッケージをダウンロードする

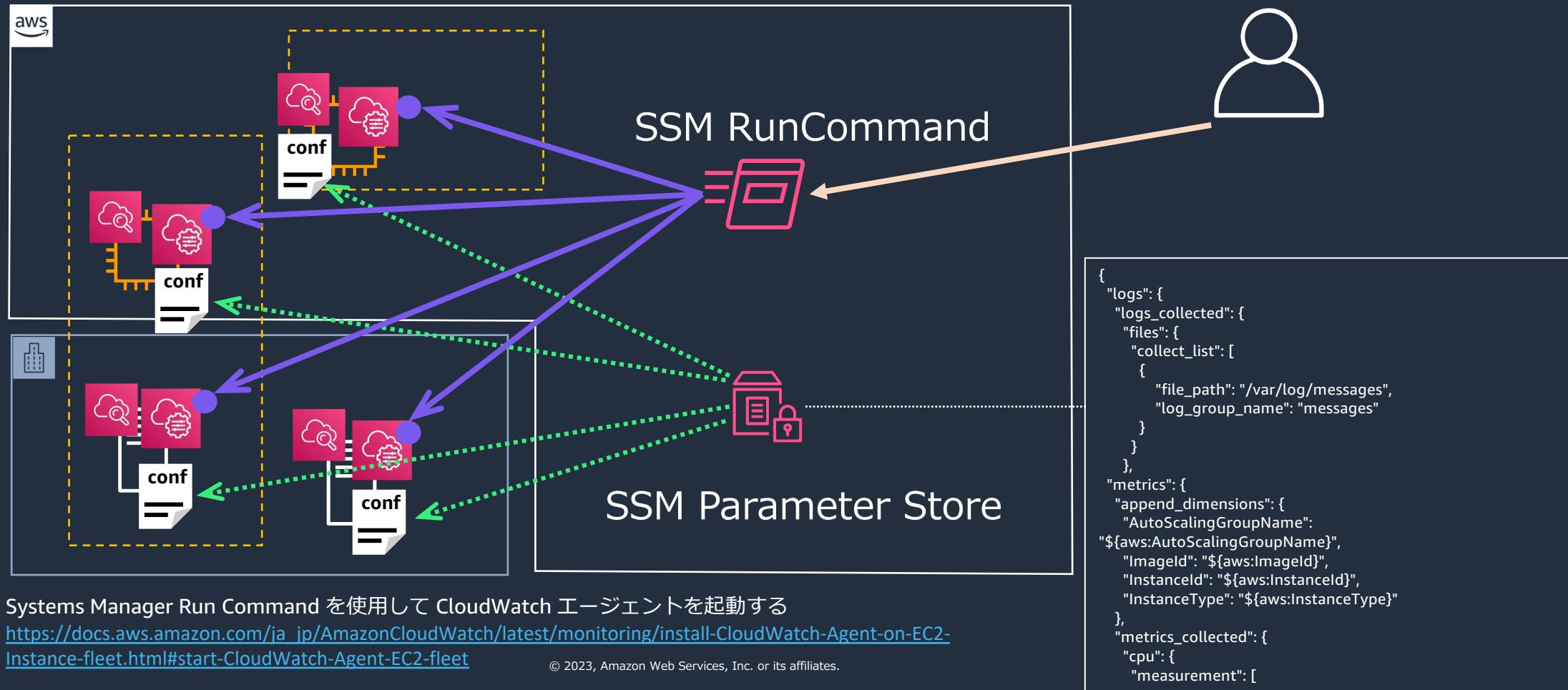
[https://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/monitoring/install-CloudWatch-Agent-on-EC2-Instance-fleet.html#download-CloudWatch-Agent-on-EC2-Instance-fleet](https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/monitoring/install-CloudWatch-Agent-on-EC2-Instance-fleet.html#download-CloudWatch-Agent-on-EC2-Instance-fleet)



## 手順3. 統合CloudWatch Agent の一括セットアップ (3/3)

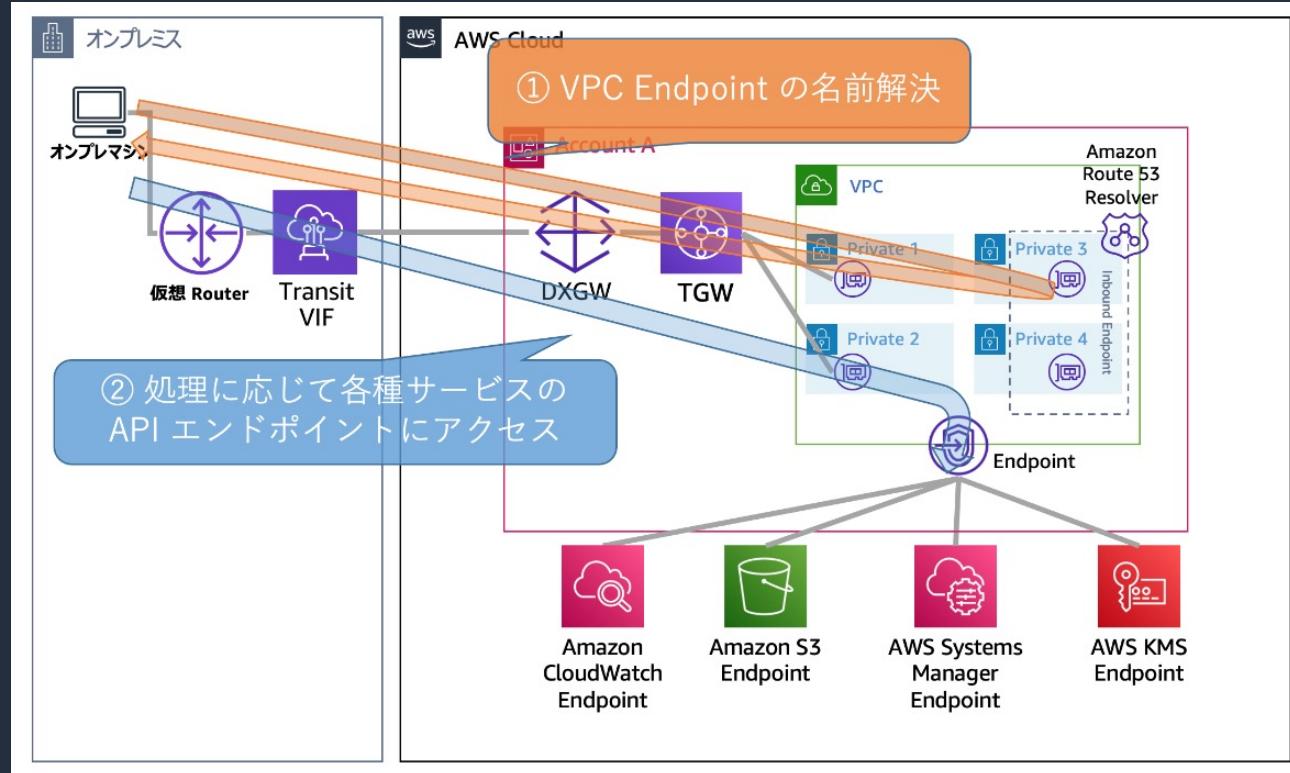
### 3. RunCommand で Parameter Store の設定を一括でロード

- "AmazonCloudWatch-ManageAgent" ドキュメントを実行



# ハイブリッド環境の運用・監視

- オンプレミスと閉域網接続の場合にもCloudWatchを活用



・オンプレミスサーバ上の AWS Systems Manager Agent (SSM Agent) と Amazon CloudWatch Agent が通信するサービスエンドポイントへの経路を VPC endpoint で構成

・オンプレミスサーバ上の SSM Agent と CloudWatch Agent が VPC endpoint の DNS 名を解決できるように Amazon Route 53 Resolver を構成

※このブログではオンプレミスと AWS を専用線で接続し、AWS Transit Gateway を経由させた構成ですが、VGW経由など他の構成でも問題ありません。

参考Blog : ハイブリッド環境の運用・監視の実現 – 閉域網で AWS Systems Manager と Amazon CloudWatch を構成する  
<https://aws.amazon.com/jp/blogs/news/operation-of-a-hybrid-environment/>

# Tips : CloudWatchのVPC Endpoint

利用するCloudWatchのサービス毎に使い分ける

AWS Service	Service name	
Amazon CloudWatch	com.amazonaws.region.monitoring	CloudWatch
	com.amazonaws.region.evidently	CloudWatch Evidentlyのサービス通信
	com.amazonaws.region.evidently-dataplane	CloudWatch Evidentlyの管理通信
	com.amazonaws.region.rum	CloudWatch RUMのサービス通信
	com.amazonaws.region.rum-dataplane	CloudWatch RUMの管理通信
	com.amazonaws.region.synthetics	CloudWatch Synthetics
Amazon EventBridge Amazon CloudWatch Events	com.amazonaws.region.events	AWS EventBridge ※CloudWatch Eventsと共通
Amazon CloudWatch Logs	com.amazonaws.region.logs	CloudWatch Logs

★SSMで統合CloudWatch Agentを使うモニタリングをVPC Endpoint経由で設定する場合、必要なVPC Endpointの例

- com.amazonaws.**region**.ssm
- com.amazonaws.**region**.ssmmessages
- com.amazonaws.**region**.ec2messages
- com.amazonaws.**region**.kms
- com.amazonaws.**region**.s3
- com.amazonaws.**region**.logs
- com.amazonaws.**region**.monitoring

参考 : AWS services that integrate with AWS PrivateLink

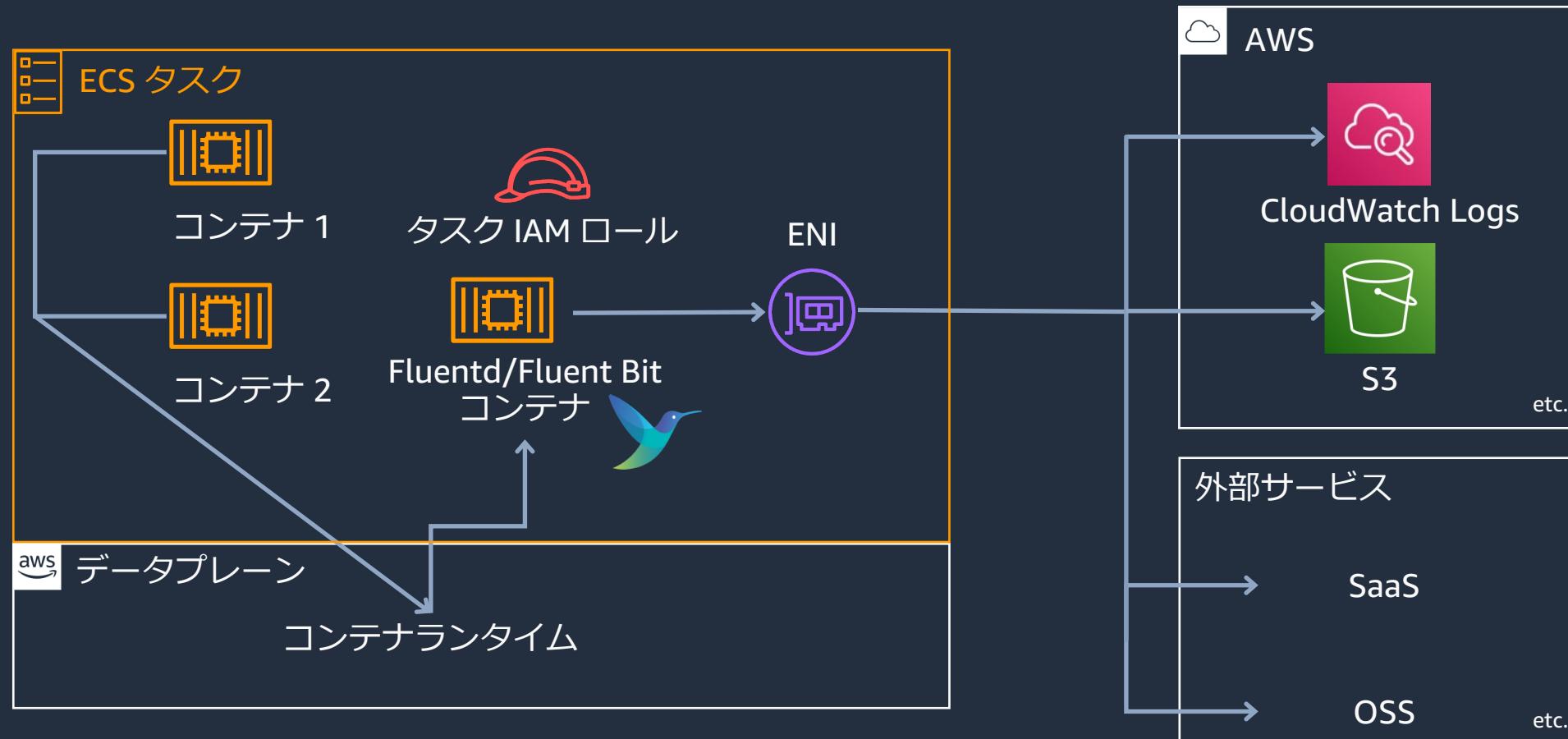
<https://docs.aws.amazon.com/vpc/latest/privatelink/aws-services-privatelink-support.html>



- コンテナ化されたアプリケーションログを **CloudWatch Logs** に送信
  - STDOUT および STDERR をログイベントとして取得
- タスク定義に logConfiguration パラメータを追加するだけで有効化が可能
  - 追加のエージェントなどのインストールは不要
  - ECS on EC2, ECS on Fargate のいずれでも利用可能



Fluentd / Fluent Bit を利用した柔軟なログルーティング



[https://docs.aws.amazon.com/AmazonECS/latest/developerguide/using\\_firelens.html](https://docs.aws.amazon.com/AmazonECS/latest/developerguide/using_firelens.html)

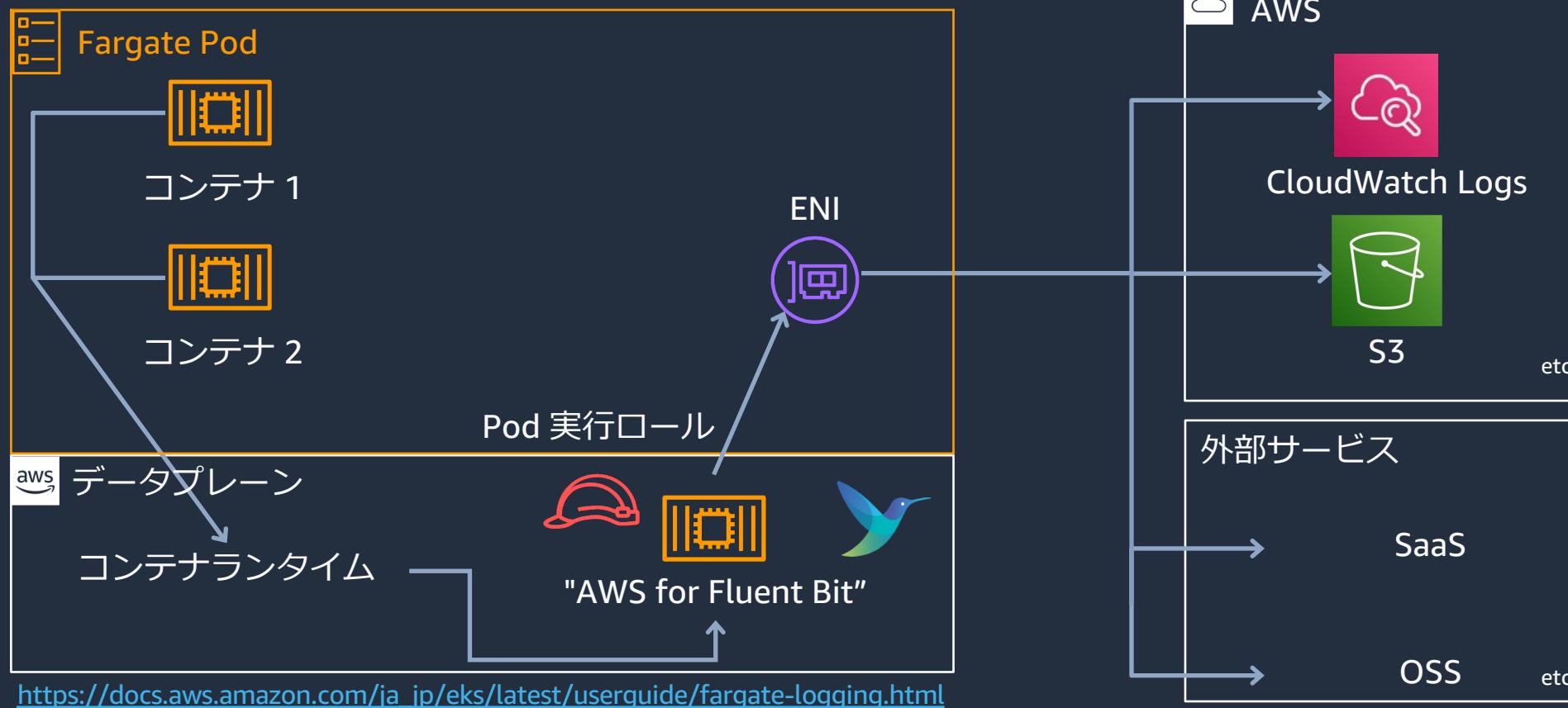
© 2021, Amazon Web Services, Inc. or its Affiliates.



# Amazon EKS on Fargate 対応 FireLens

CloudWatch Logsに  
EKS on Fargateからログを送る

Fluent Bit をベースにした組み込みのログルーター



※コントロールプレーンのログはEKSで有効化することで取得できる。

[https://docs.aws.amazon.com/ja\\_jp/prescriptive-guidance/latest/implementing-logging-monitoring-cloudwatch/kubernetes-eks-logging.htm](https://docs.aws.amazon.com/ja_jp/prescriptive-guidance/latest/implementing-logging-monitoring-cloudwatch/kubernetes-eks-logging.htm)

EKS on EC2の場合はFluentBitないしはFluentdをNodeにDaemonSetとして配置。!

[https://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/monitoring/Container-Insights-setup-logs-FluentBit.html](https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/monitoring/Container-Insights-setup-logs-FluentBit.html)

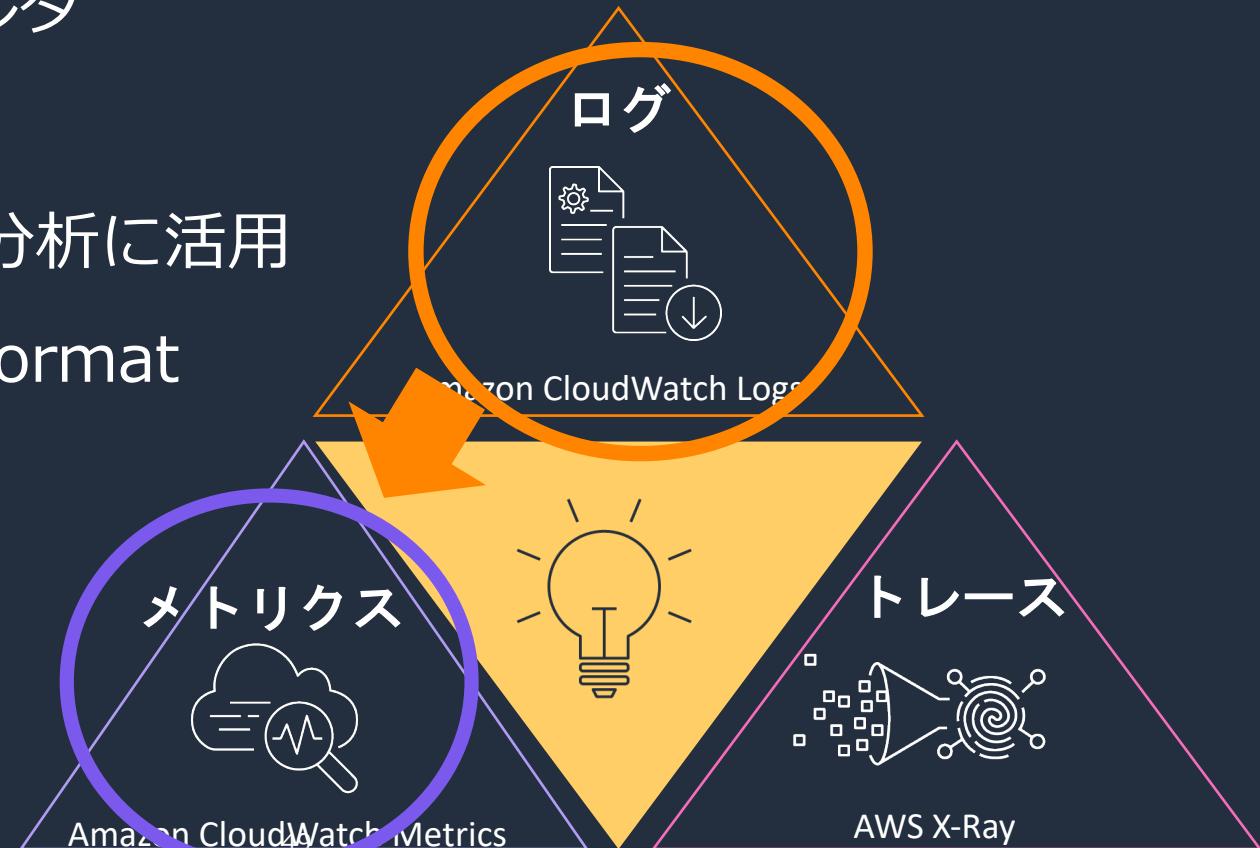
# ログからメトリクスを抽出して通知や分析に活用も

① CloudWatch Logsに集めた後、ロググループ単位で分析し、通知に活用

→ CloudWatch Logs メトリクスフィルタ

② ログ出力時にメトリクスを埋め込み、分析に活用

→ CloudWatch Embedded Metrics Format



# CloudWatch Logs メトリクスフィルタ

## CloudWatch Logsのログをフィルタしてメトリクスにパブリッシュ

- ログデータから特定の文字列のフィルタリングが可能
- フィルターとパターンの構文で指定できる範囲内で文字列をフィルターが可能
- 特定文字列のエントリ頻度等によりアラーム作成、SNS連携が可能

The screenshot shows the AWS CloudWatch Metrics Filter creation interface. On the left, there's a navigation sidebar with three steps: Step 1 (Pattern to Define), Step 2 (Metrics assigned), and Step 3 (Review and Create). The main area has two tabs: 'Pattern to Define' (selected) and 'Metrics Details'.  
**Pattern to Define Tab:**

- Filter Pattern:** A text input field containing 'ERROR'. This field is highlighted with a red box and has a red arrow pointing from the explanatory text below it to its right side.
- Matching Log Events:** A list of patterns:
  - 用語を含むログイベントの照合:
    - ERROR
    - “[INFO]”
    - “Warning:”
  - ロギングイベントと 400 レベルの HTTP 応答との照合:
    - [host, logName, user, timestamp, request, statusCode=4\*, size]
  - JSON ロギングイベントとの照合:
    - { \$.errorCode = "AccessDenied" }

**Metrics Details Tab:**

- Metrics Namespace:** A text input field containing 'Metrics namespace to input'. A red box highlights this field, and a red arrow points from the explanatory text below it to its right side.
- Metrics Name:** A text input field containing 'Metrics name to input'. A red box highlights this field, and a red arrow points from the explanatory text below it to its right side.
- Metrics Value:** A text input field containing 'Metrics value to input'. A red box highlights this field, and a red arrow points from the explanatory text below it to its right side.
- Default Value - Option:** A dropdown menu labeled 'Select a unit'.

**Explanatory Text:**

- 定義したパターンに一致した時に (When the defined pattern matches)
- メトリクスにパブリッシュされる値を設定 (ERRORが含まれた単純なパターン数を求める時はメトリクス値 1 を指定) (Set the value to be published to metrics (Specify a value of 1 when calculating the count of simple patterns containing ERROR))
- 一致したログをカウントした値をメトリクスにできる (Count the matching logs and publish them to metrics)



# CloudWatch Embedded Metrics Format (EMF)

詳細なログイベントと一緒にカスタムメトリクスを埋め込む

EMF の例

- 複雑なアプリケーションデータをログの形式で取り込み、実用的なメトリクスを生成
- CloudWatch はカスタムメトリクスをログから自動的に抽出
- クライアントライブラリを使用してEMF形式のログを生成、もしくは手動で生成した EMF 形式のログを PutLogEvents API や CloudWatch エージェントで送信
- ADOTにもCloudWatch Embedded Metric Format(EMF) Exporterがあり、対応している。

```
{  
  "_aws": {  
    "Timestamp": 1574109732004,  
    "CloudWatchMetrics": [  
      {  
        "Namespace": "lambda-function-metrics",  
        "Dimensions": [{"functionVersion"}],  
        "Metrics": [  
          {  
            "Name": "time",  
            "Unit": "Milliseconds",  
            "StorageResolution": 60  
          }  
        ]  
      },  
      "functionVersion": "$LATEST",  
      "time": 100,  
      "requestId": "989ffbf8-9ace-4817-a57c-e4dd734019ee"  
    ]  
  }  
}
```

[https://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/monitoring/CloudWatch\\_EMBEDDED\\_Metric\\_Format.html](https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/monitoring/CloudWatch_EMBEDDED_Metric_Format.html)  
<https://aws-otel.github.io/docs/getting-started/cloudwatch-metrics#cloudwatch-emf-exporter-awsemf>

# Amazon S3 へのログデータのエクスポート

コンソールかAWS CLIにより利用

コンソールの使用

- エクスポートするデータの期間
- S3バケットの指定
- バケットプレフィックスの指定

AWS CLIを使用したエクスポート

エクスポートタスクの作成

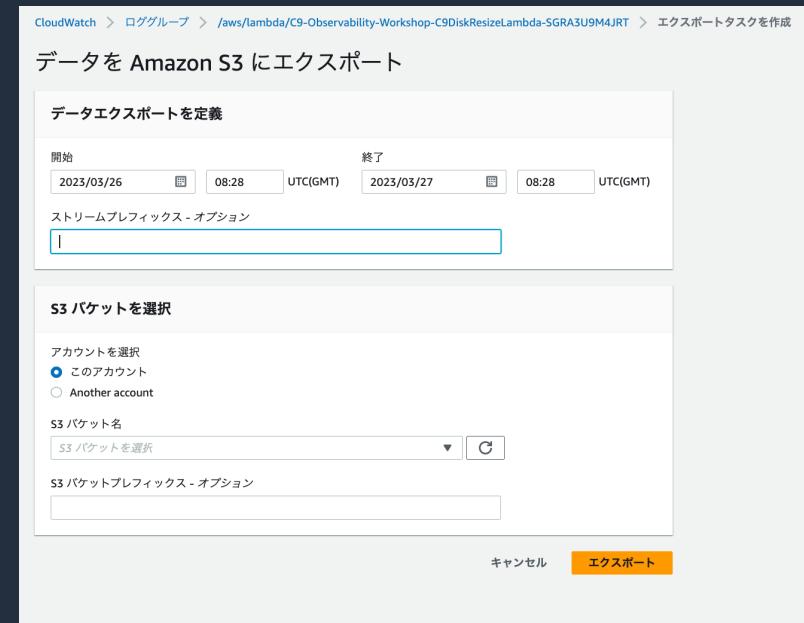
```
aws logs create-export-task --task-name "my-Log-group-09-10-2015" --log-group-name "my-Log-group" --from 1441490400000 --to 1441494000000 --destination "my-exported-Logs" --destination-prefix "export-task-output"
```

タスクの作成ステータス確認

```
aws logs describe-export-tasks --task-id "cda45419-90ea-4db5-9833-aade86253e66"
```

※オプションでストリームプレフィックスを指定すると、指定されたプレフィックスに一致するログストリームのみをエクスポートします。

ログを長期保管したい！  
他サービスで活用したい！



※ログデータは、エクスポートできるようになるまで最大 12 時間かかる場合があります

# CloudWatch Logs サブスクリプションフィルタ

CloudWatch Logsに集めたログをフィルタパターンに応じて外部にリアルタイムに転送

CloudWatch Logsに集めたログをフィルタパターンに応じてリアルタイムに  
Kinesis Data Streams/Kinesis Data Firehose/Lambdaへ転送

- 1つのロググループにつき、2つのサブスクリプションフィルタを設定可能



CloudWatch Logs



サブスクリプション  
フィルタ



AWS CLIからのみ設定可能

```
aws logs put-subscription-filter ¥
  --log-group-name "xxxxxxxx" ¥
  --filter-name "xxxxxxxx" ¥
  --filter-pattern "{xxxxxxxx = xxxxxxxx}" ¥
  --destination-arn "arn:aws:kinesis:ap-northeast-1:123456789012:stream/xxxxxxxx" ¥
  --role-arn "arn:aws:iam::123456789012:role/xxxxxxxx"
```

# ユースケース：サブスクリプションフィルタの利用例

## Lambdaを活用したOpenSearch Serviceへのストリーミング



## Kinesis Data Firehoseを活用したS3等へのデータ転送



Kinesis Data Firehoseのカスタムプレフィックスによりタイムスタンプ情報をApache Hiveフォーマットに設定可能

myPrefix/year={!timestamp:yyyy}/month={!timestamp:MM}/day={!timestamp:dd}/hour={!timestamp:HH}/

パスの例  
myPrefix/year=2018/month=07/day=06/hour=23/



[https://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/logs/CWL\\_OpenSearch\\_Stream.html](https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/logs/CWL_OpenSearch_Stream.html)

[https://docs.aws.amazon.com/ja\\_jp/AmazonCloudWatch/latest/logs/Subscriptions.html](https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/logs/Subscriptions.html)

[https://docs.aws.amazon.com/ja\\_jp/firehose/latest/dev/s3-prefixes.html](https://docs.aws.amazon.com/ja_jp/firehose/latest/dev/s3-prefixes.html)

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

# CloudWatch Metric streams

## 継続的なメトリクスのニアリアルタイムストリーム

- CloudWatch メトリクスを選択した宛先に継続的にストリーミングし、ニアリアルタイムで配信
- 送信先: AWS のサービス (Amazon Kinesis Data Firehose など) またはサードパーティのサービスプロバイダー
- ユースケース: Amazon S3等でデータレイクを構築するか、サードパーティのツールを使用してメトリクスを監視する



例: metrics を Amazon S3 bucketへ

Your metric stream MyMetrics-US-East-1A was successfully created.  
The following resources have been created: S3 bucket [ ], S3 write role [ ], Firehose [ ], Firehose write role [ ]

CloudWatch > Metric Streams

Metric streams | Monitoring

Metric streams (1) Info

Name	Status	Last Updated Time	Destination
MyMetrics-US-East-1A	Running	03/23/2021 5:48 PM	S3 [ ]

S3バケットのファイル例

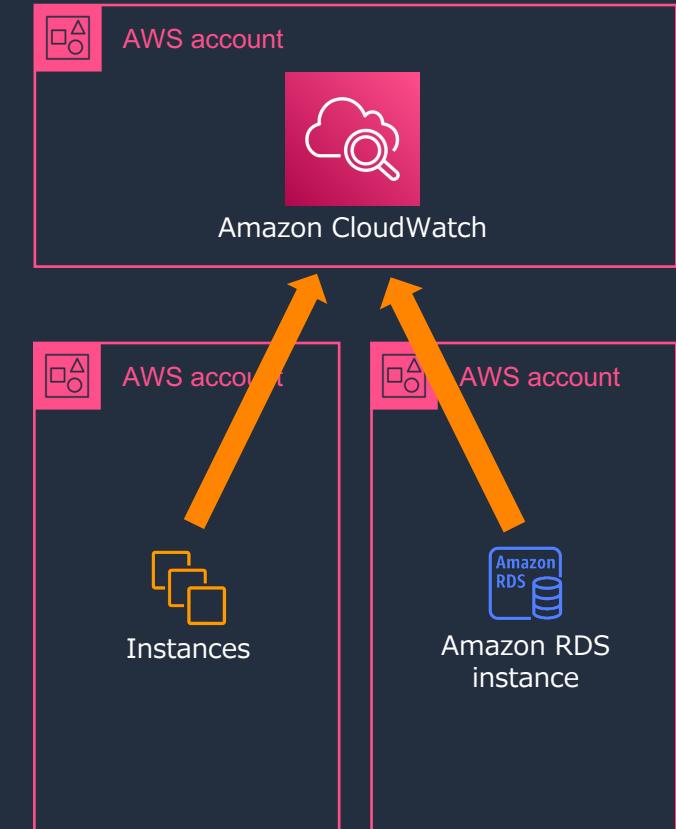
```

new 26 ms.yaml MetricStreams-MyMetrics-US-East-1A-t1YyyRge-1-2021-03-24-00-51-08-2e9e2a5f3b8a-4f5b-bbd4-2200f74492f5
1 {"metric_stream_name": "MyMetrics-US-East-1A", "account_id": "12345678901234567890123456789012", "region": "us-east-1", "dimensions": {"ClusterIdentifier": "test-cluster", "service": "WLMQueueLength", "value": {"count": 1.0, "sum": 0.0, "max": 0.0, "min": 0.0}}, "timestamp": 1616546940000, "value": {"count": 1.0, "sum": 0.0, "max": 0.0, "min": 0.0}}
2 {"metric_stream_name": "MyMetrics-US-East-1A", "account_id": "12345678901234567890123456789012", "region": "us-east-1", "dimensions": {"ClusterIdentifier": "test-cluster", "NodeID": "Compute-1", "value": {"count": 1.0, "sum": 0.0, "max": 0.0, "min": 0.0}}, "timestamp": 1616546940000, "value": {"count": 1.0, "sum": 0.0, "max": 0.0, "min": 0.0}}
3 {"metric_stream_name": "MyMetrics-US-East-1A", "account_id": "12345678901234567890123456789012", "region": "us-east-1", "dimensions": {"ClusterIdentifier": "test-cluster", "NodeID": "NetworkReceiveThroughput", "value": {"count": 1.0, "sum": 7062.616666666667, "max": 7062.616666666667, "min": 7062.616666666667}}, "timestamp": 1616546940000, "value": {"count": 1.0, "sum": 7062.616666666667, "max": 7062.616666666667, "min": 7062.616666666667}}
4 {"metric_stream_name": "MyMetrics-US-East-1A", "account_id": "12345678901234567890123456789012", "region": "us-east-1", "dimensions": {"ClusterIdentifier": "test-cluster", "NodeID": "NetworkTransmitThroughput", "value": {"count": 1.0, "sum": 2583.340277662039, "max": 2583.340277662039, "min": 2583.340277662039}}, "timestamp": 1616546940000, "value": {"count": 1.0, "sum": 2583.340277662039, "max": 2583.340277662039, "min": 2583.340277662039}}
5 {"metric_stream_name": "MyMetrics-US-East-1A", "account_id": "12345678901234567890123456789012", "region": "us-east-1", "dimensions": {"ClusterIdentifier": "prod-cluster", "NodeID": "Compute-1", "value": {"count": 1.0, "sum": 0.0, "max": 0.0, "min": 0.0}}, "timestamp": 1616546940000, "value": {"count": 1.0, "sum": 0.0, "max": 0.0, "min": 0.0}}
6 {"metric_stream_name": "MyMetrics-US-East-1A", "account_id": "12345678901234567890123456789012", "region": "us-east-1", "dimensions": {"ClusterIdentifier": "prod-cluster", "NodeID": "NetworkReceiveThroughput", "value": {"count": 1.0, "sum": 7062.616666666667, "max": 7062.616666666667, "min": 7062.616666666667}}, "timestamp": 1616546940000, "value": {"count": 1.0, "sum": 7062.616666666667, "max": 7062.616666666667, "min": 7062.616666666667}}
  
```

3rd Party Tool  
S3/Redshift

# Amazon CloudWatchでアカウント横断の監視が可能に

- Amazon CloudWatchで複数のアカウントを横断的にモニタリングできるようになった
  - メトリクス、ログ、AWS X-Rayによるトレースなどの可視化や分析をアカウント境界を気にせずに一元的に実現
  - 例えばアカウントを跨ってAWS Lambdaを介して連携するアプリケーションのエンドツーエンドでのトレースが容易に
  - 単一リージョンで展開される複数アカウント環境の監視と障害対応を支援
- メトリクスとログは追加料金なし、トレースは1つの監視アカウントでの利用は追加料金なしで提供
- 全商用リージョンで利用可能



新着 – Amazon CloudWatch のクロスアカウントオブザーバビリティ

<https://aws.amazon.com/jp/blogs/aws/new-amazon-cloudwatch-cross-account-observability/>

<https://aws.amazon.com/jp/blogs/news/new-amazon-cloudwatch-cross-account-observability/>

# 再掲：CloudWatchの機能一覧

## Infrastructure

AWSサービス名	概要
CloudWatch Metrics	メトリクス
CloudWatch Log	ログ
CloudWatch Alarm	アラーム
CloudWatch Dashboard	ダッシュボード
CloudWatch Metrics Explorer	メトリックス検索
CloudWatch Metrics Stream	メトリックスのリアルタイム連携
CloudWatch Events ※これはEvent Bridgeに統合されています	イベント
CloudWatch Resource Health	EC2の健全性・パフォーマンス可視化

## Application Monitoring

AWSサービス名	概要
CloudWatch Synthetics	外形監視
CloudWatch RUM	リアルユーザー モニタリング
CloudWatch Evidently	フィーチャーフラグA/Bテスト
CloudWatch Internet Monitor	インターネット監視
CloudWatch ServiceLens	トレース

## Insights

AWSサービス名	概要
CloudWatch Contributor Insights	ログの時系列分析
CloudWatch Container Insights	コンテナ分析
CloudWatch Lambda Insights	Lambda分析
CloudWatch Application Insights	アプリケーション分析
CloudWatch Logs Insights	LOG分析
CloudWatch Metrics Insights	メトリクス分析

グレーアウトはご紹介済み

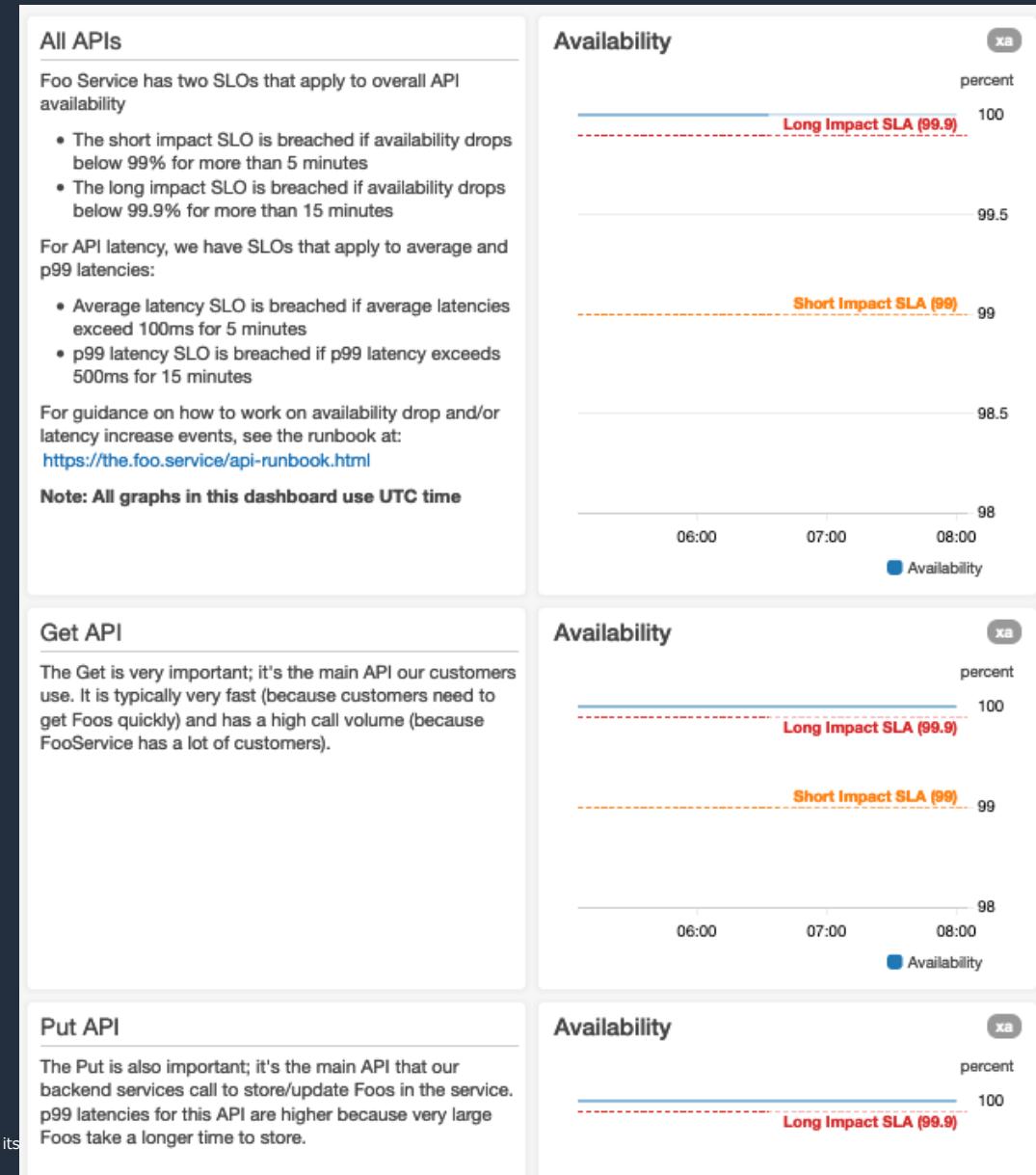
★ここからはその他のサービスを簡単にご紹介  
※個別の詳細は別途Blackbeltセッションで紹介予定

# CloudWatch Dashboard

### カスタマイズ可能なダッシュボードを作成

- 折れ線グラフ、数値、ゲージ、テキスト、アラーム、Lambdaによるカスタムウィジェットでお好みのダッシュボードを作成
  - 異なるアカウント、異なるリージョンのリソースでも、ダッシュボード化が可能
  - 自動更新間隔(10s, 1m, 2m, 5m, 15m)、時間範囲、タイムゾーンの調整が可能

## 目的別のダッシュボードを構築する



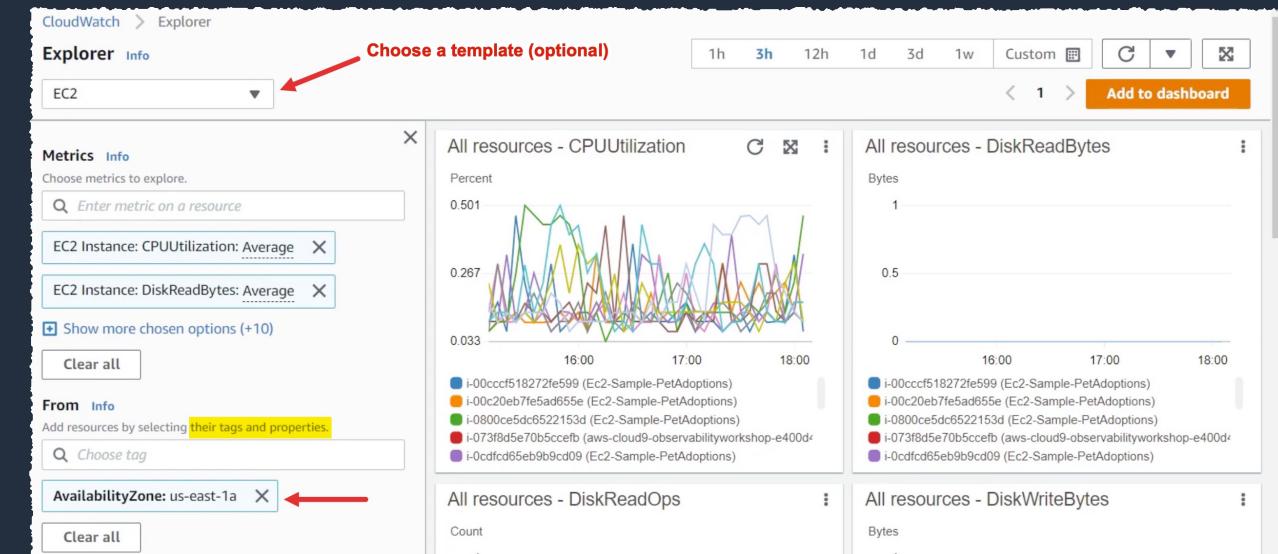
# CloudWatch Metrics explorer

## タグとプロパティでメトリクスを監視する

- タグとリソース、プロパティによるメトリクスのフィルタリング、集計、および視覚化のためのツール
- 検索条件に当てはまる場合、動的に新しいリソースのメトリクスが追加される
- 付属のテンプレートを使用すると、ワンクリックで便利にグラフを作成可能

メトリクス検索をタグで柔軟に

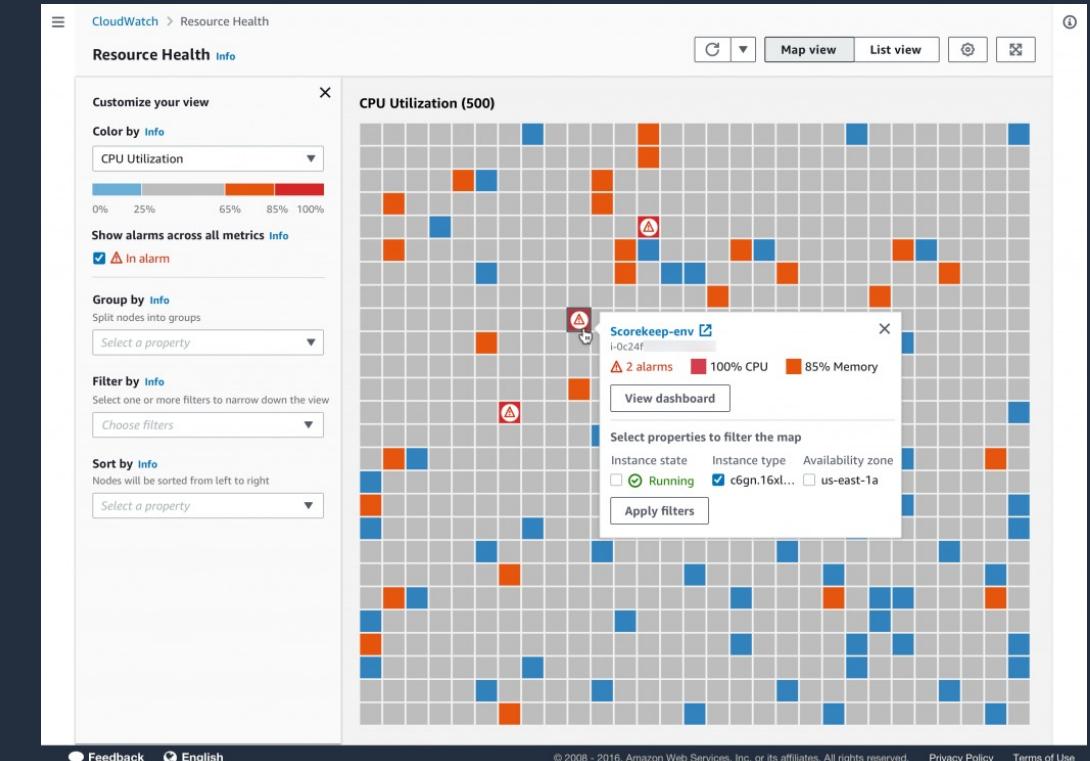
例: us-east-1a アベイラビリティーゾーンのすべてのAmazon EC2 インスタンスのメトリクスを表示します



# CloudWatch Resource Health (Amazon EC2)

Amazon EC2 ホストの正常性を自動的に検出して視覚化する

- Amazon EC2 インスタンスの正常性とパフォーマンスを自動的に検出して視覚化するフルマネージドソリューション
- AZ・VPC・AutoScalingGroup・CPUなどキャパシティの状況やタグなどの情報ごとにグループ化/フィルタリング/ソート可能



# 再掲：CloudWatchの機能一覧

## Infrastructure

AWSサービス名	概要
CloudWatch Metrics	メトリクス
CloudWatch Log	ログ
CloudWatch Alarm	アラーム
CloudWatch Dashboard	ダッシュボード
CloudWatch Metrics Explorer	メトリックス検索
CloudWatch Metrics Stream	メトリックスのリアルタイム連携
CloudWatch Events ※これはEvent Bridgeに統合されています	イベント
CloudWatch Resource Health	EC2の健全性・パフォーマンス可視化

## Application Monitoring

AWSサービス名	概要
CloudWatch Synthetics	外形監視
CloudWatch RUM	リアルユーザー モニタリング
CloudWatch Evidently	フィーチャーフラグA/Bテスト
CloudWatch Internet Monitor	インターネット監視
CloudWatch ServiceLens	トレース

## Insights

AWSサービス名	概要
CloudWatch Contributor Insights	ログの時系列分析
CloudWatch Container Insights	コンテナ分析
CloudWatch Lambda Insights	Lambda分析
CloudWatch Application Insights	アプリケーション分析
CloudWatch Logs Insights	LOG分析
CloudWatch Metrics Insights	メトリクス分析

グレーアウトはご紹介済み

★ここからはその他のサービスを簡単にご紹介  
※個別の詳細は別途Blackbeltセッションで紹介予定

# Application monitoring

## サーバーとクライアントサイドのパフォーマンスマニタリング

### Amazon CloudWatch Synthetics (canaries)



- ・アプリケーションのエンドポイントを監視
- ・Blueprintで以下のようなモニタリングが可能
  - ハートビートモニター
  - API Canary
  - リンク切れチェッカー
  - ビジュアルモニタリング
  - Canary Recorder
  - GUI ワークフロー

### Amazon CloudWatch Real-user monitoring

- ・Web アプリケーションのパフォーマンスに関するクライアント側のデータをリアルタイムで収集
- ・ユーザー数、位置情報、ブラウザなど、エンドユーザーへのさまざまな影響をモニター



### Amazon CloudWatch Evidently

- ・一般的なユーザーに向けてロールアウトする前に、新機能の実験を行う
- ・A/B テストを実施してEvidenceとデータに基づいて機能の設計を決定可能に



# CloudWatch Synthetics

Canary を使用し、ユーザートラフィックがない場合でも  
24時間365日、ユーザー体験を継続的に監視



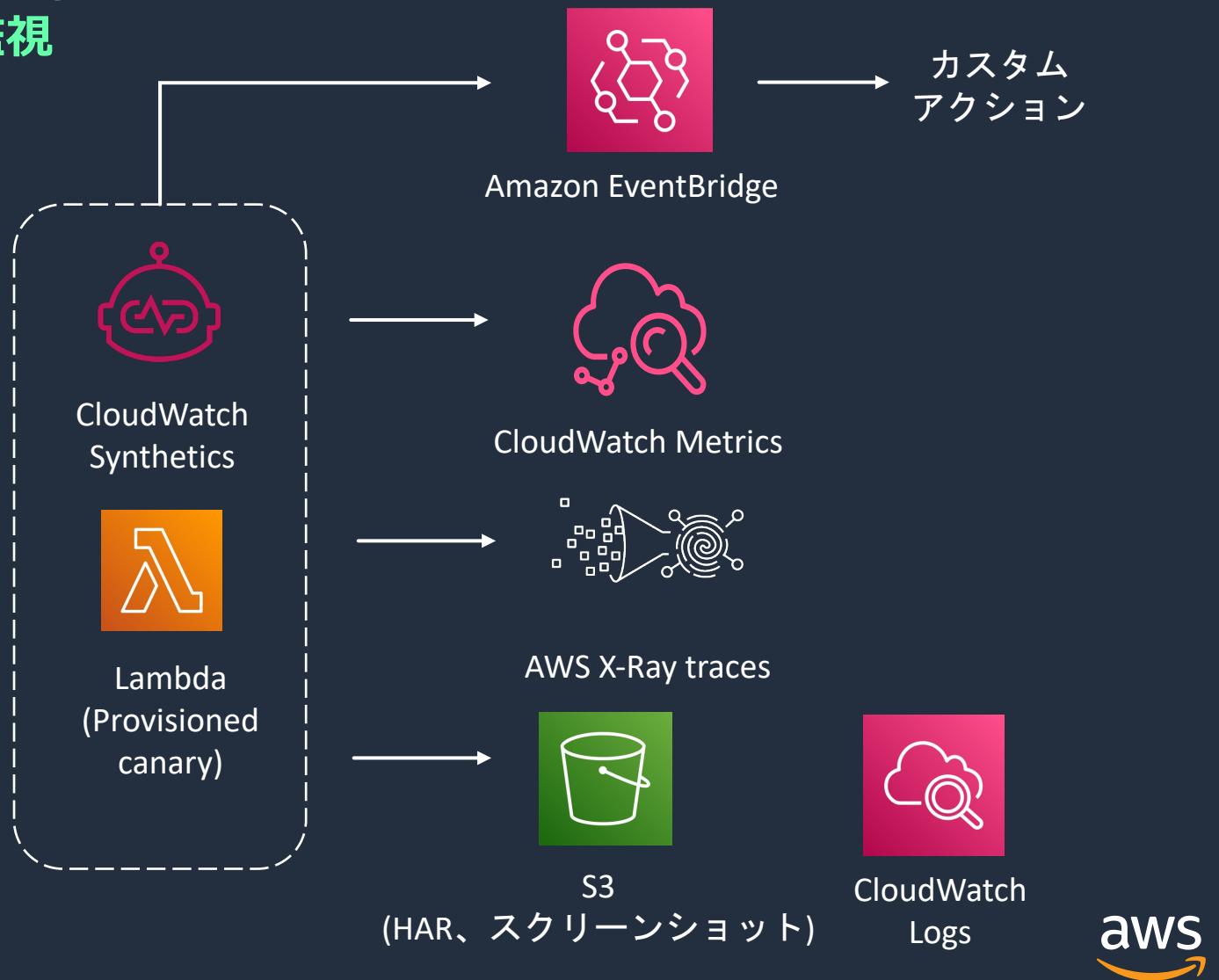
ウェブサイトおよび API  
のエンドポイント監視



クライアントサイドか  
らサーバーサイドへ



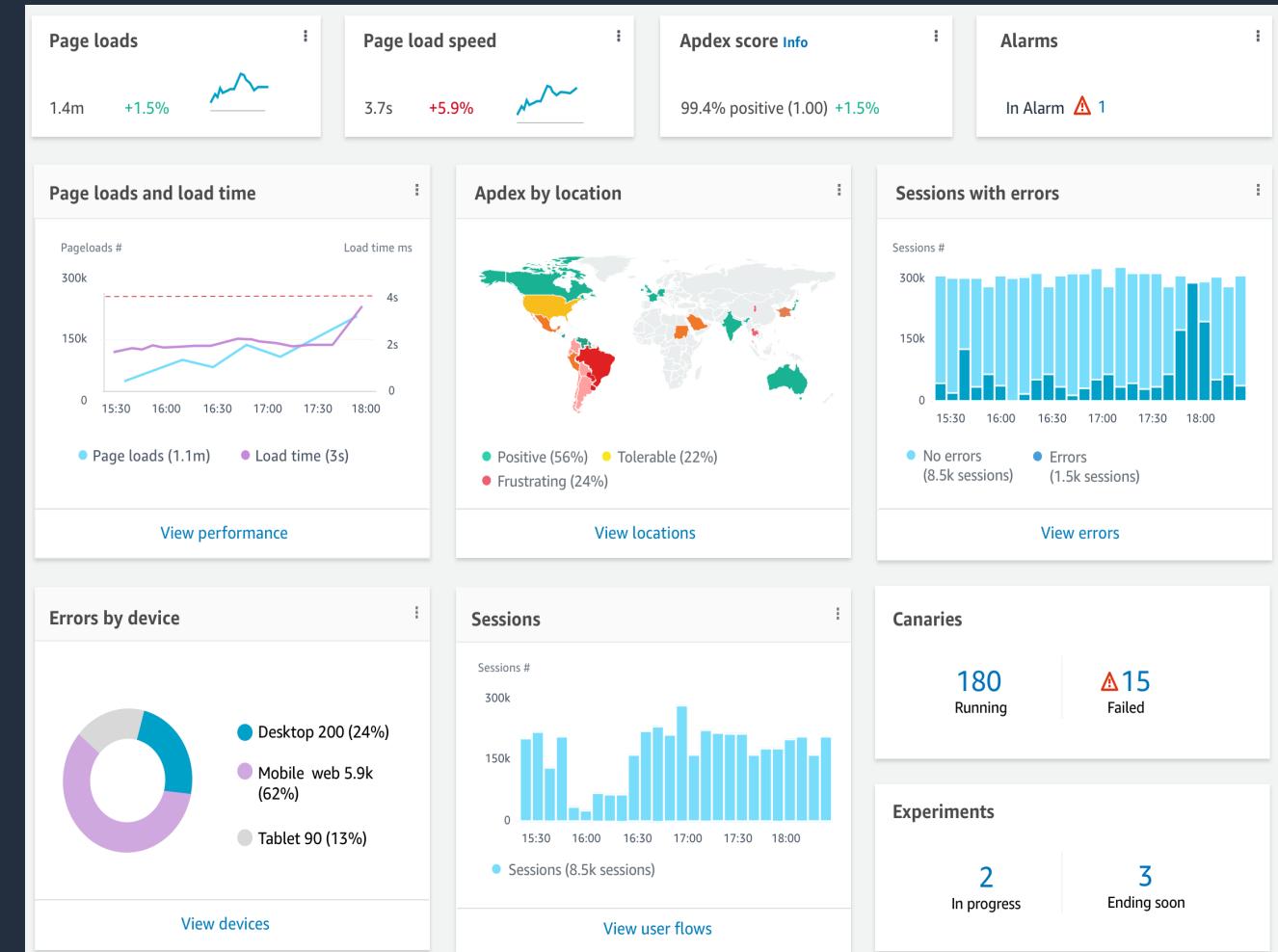
継続的なモニタリング



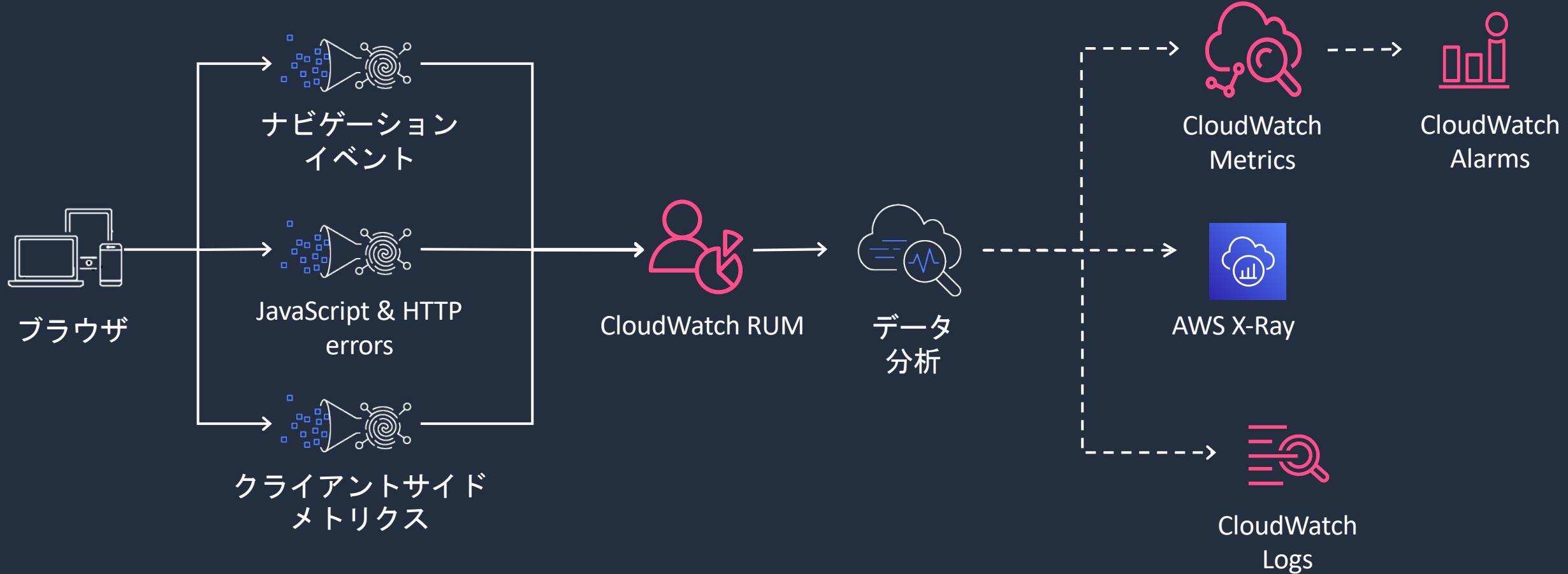
# CloudWatch RUM

アプリのパフォーマンスに関するクライアントサイドのデータをリアルタイムで取得し、顧客体験を最適化

- リアルユーザーのパフォーマンスをモニタし、ブラウザやデバイスの種類、位置、ネットワークの接続性の問題などを把握できる
  - ダッシュボードでページの読み込み順序や JavaScript / HTTP レスポンスのエラーなど、パフォーマンス問題に関する情報を可視化
  - 同じ問題の影響下にあるユーザセッション数を提示するため、改修の優先順位を付けることが容易
  - Amazon CloudWatch ServiceLens、AWS X-Ray と組み合わせることも可能



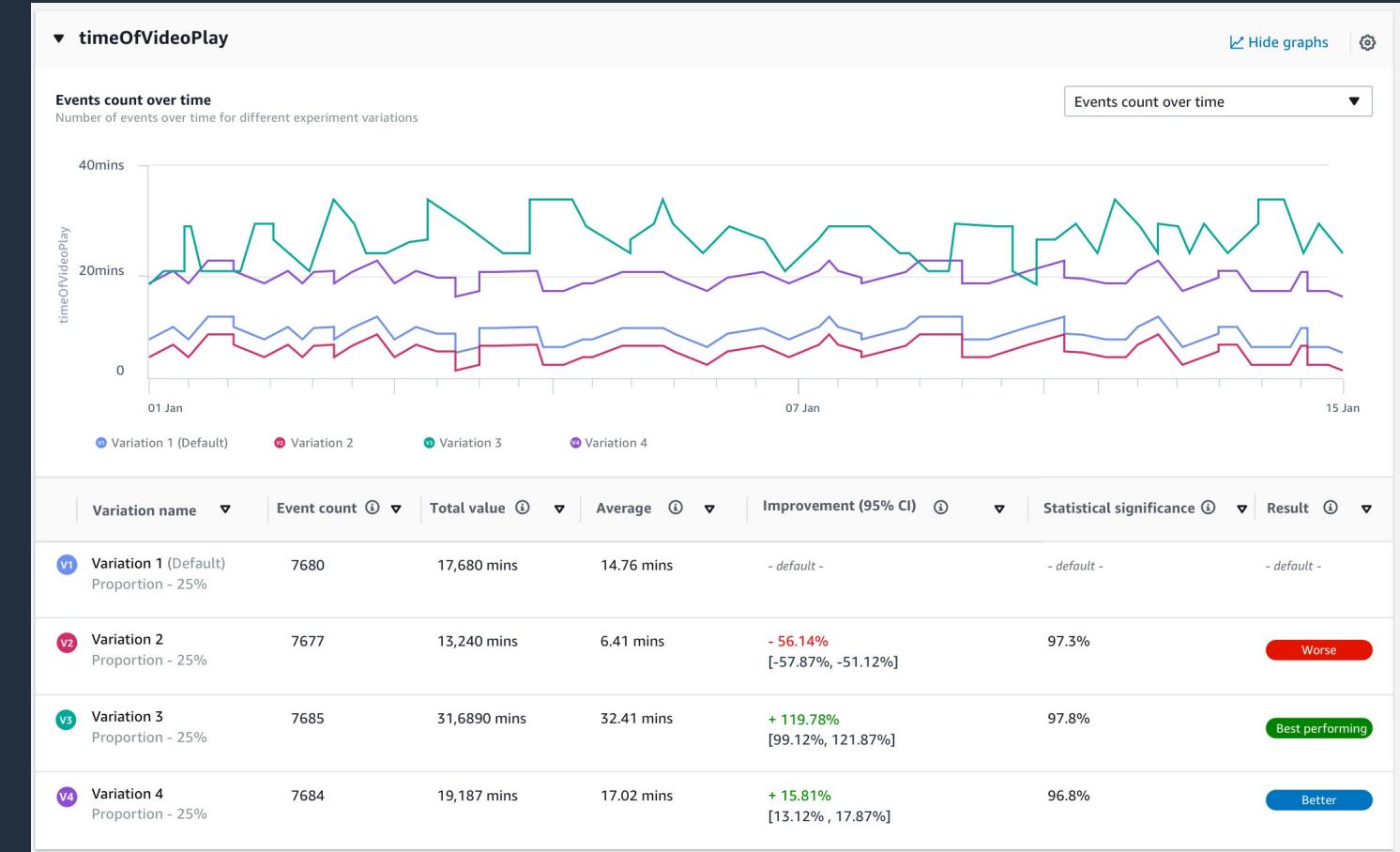
# CloudWatch RUM の動作イメージ



# CloudWatch Evidently

機能変更・リリース時に、データに基づいた良否の判断を可能に

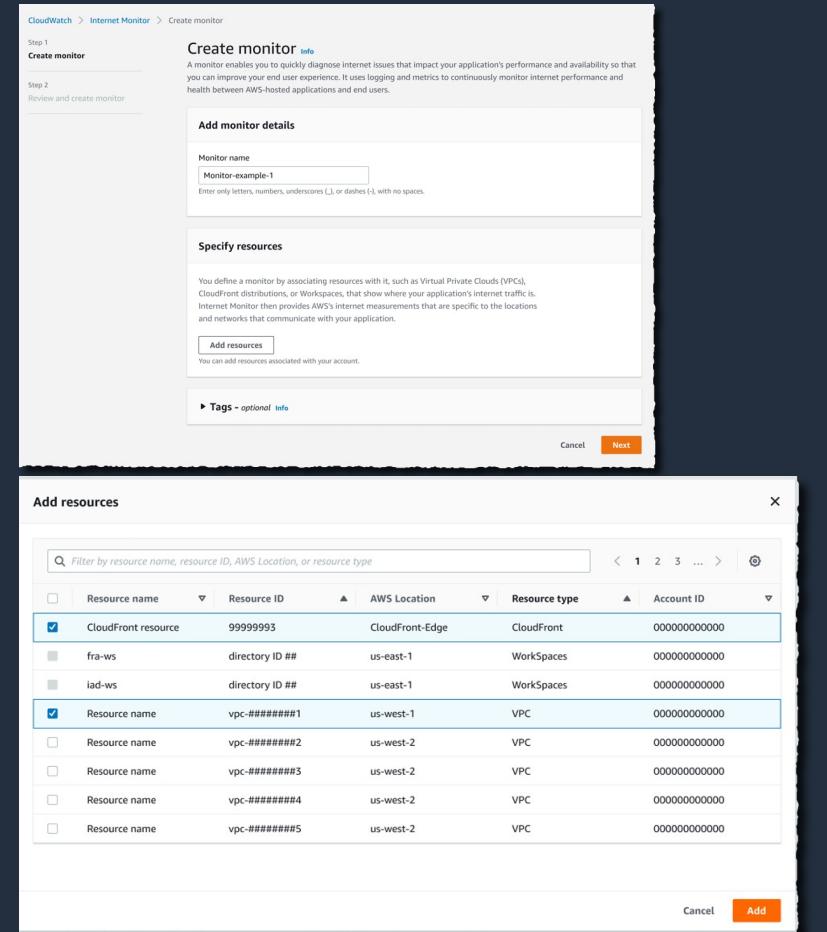
- A/Bテストやフィーチャーフラグといった手法でどちらの選択肢が望ましいか検証する際に、ユーザの挙動をモニタリングするサービス
- A/Bテスト、フィーチャーフラグで検証を行う際に、データに基づいた判断を支援。開発者が新機能を安全に評価できる
- ラフィックをコントロールするスケジュール機能や、問題発生時にロールバックするアラーム機能を備える
- Amazon CloudWatch RUMと統合されており、RUMのメトリクスを利用可能



# CloudWatch Internet Monitor

## インターネットアクセスの状況評価

- AWS 上のアプリケーションに対してインターネットからアクセスした際の可用性とパフォーマンスマトリクスを CloudWatch で可視化可能に
  - ユーザから見た場合の可用性と性能をチェックできる
  - AWS のグローバルネットワークから取得した接続データに基づきモニタリング。問題がある場合はその影響や場所、プロバイダーなどを可視化し、改善アクションを起こしやすくする
  - 例えば「概ね正常だがラスベガスからアクセスしているユーザはパフォーマンスが落ちている」といった状況が検出できる
  - VPC フローログや CloudFront ログの有効化は不要
- **2023/02/28に一般提供開始**



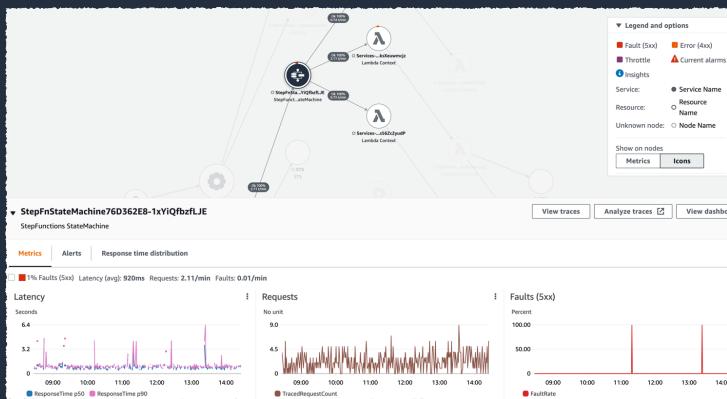
顧客が利用するアプリケーションをホストする、VPC /CloudFront ディストリビューション/WorkSpacesディレクトリを選択

# CloudWatch ServiceLens

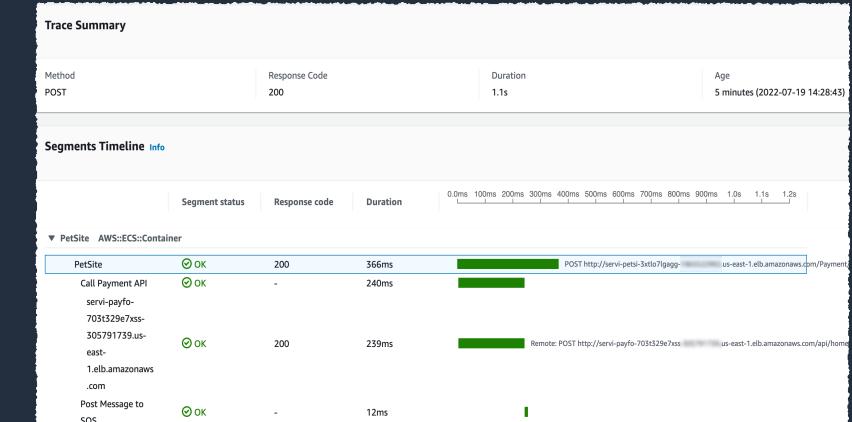
## 1箇所からサービスの健全性・可用性・パフォーマンスを確認

- Service map: サービスのコンテキストリンクを視覚化
- CloudWatch メトリクスとログ、および AWS X-Ray からのトレースを 1 カ所に結び付ける

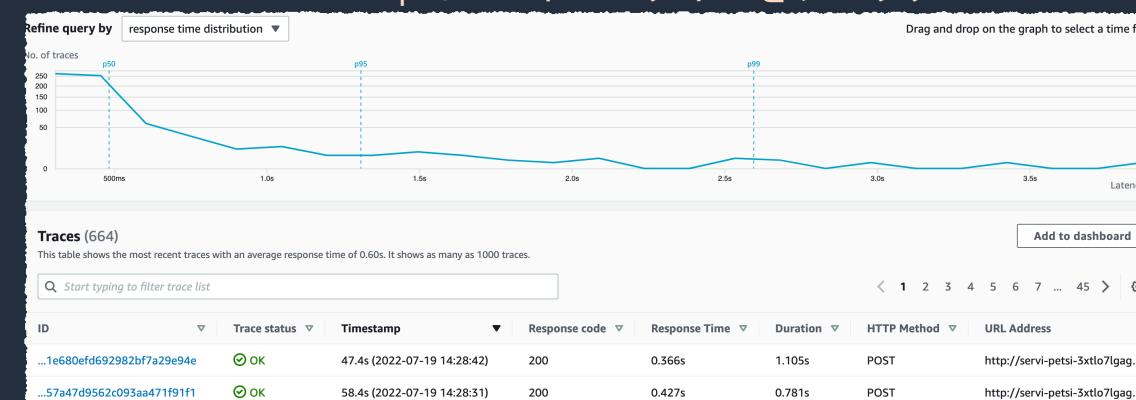
Service mapでノードとメトリクスを関連付けて確認



Service mapでトレースの詳細を見る



Service mapでレスポンスタイムをチェック



# 再掲：CloudWatchの機能一覧

## Infrastructure

AWSサービス名	概要
CloudWatch Metrics	メトリクス
CloudWatch Log	ログ
CloudWatch Alarm	アラーム
CloudWatch Dashboard	ダッシュボード
CloudWatch Metrics Explorer	メトリックス検索
CloudWatch Metrics Stream	メトリックスのリアルタイム連携
CloudWatch Events ※これはEvent Bridgeに統合されています	イベント
CloudWatch Resource Health	EC2の健全性・パフォーマンス可視化

## Application Monitoring

AWSサービス名	概要
CloudWatch Synthetics	外形監視
CloudWatch RUM	リアルユーザー モニタリング
CloudWatch Evidently	フィーチャーフラグA/Bテスト
CloudWatch Internet Monitor	インターネット監視
CloudWatch ServiceLens	トレース

## Insights

AWSサービス名	概要
CloudWatch Contributor Insights	ログの時系列分析
CloudWatch Container Insights	コンテナ分析
CloudWatch Lambda Insights	Lambda分析
CloudWatch Application Insights	アプリケーション分析
CloudWatch Logs Insights	LOG分析
CloudWatch Metrics Insights	メトリクス分析

グレーアウトはご紹介済み

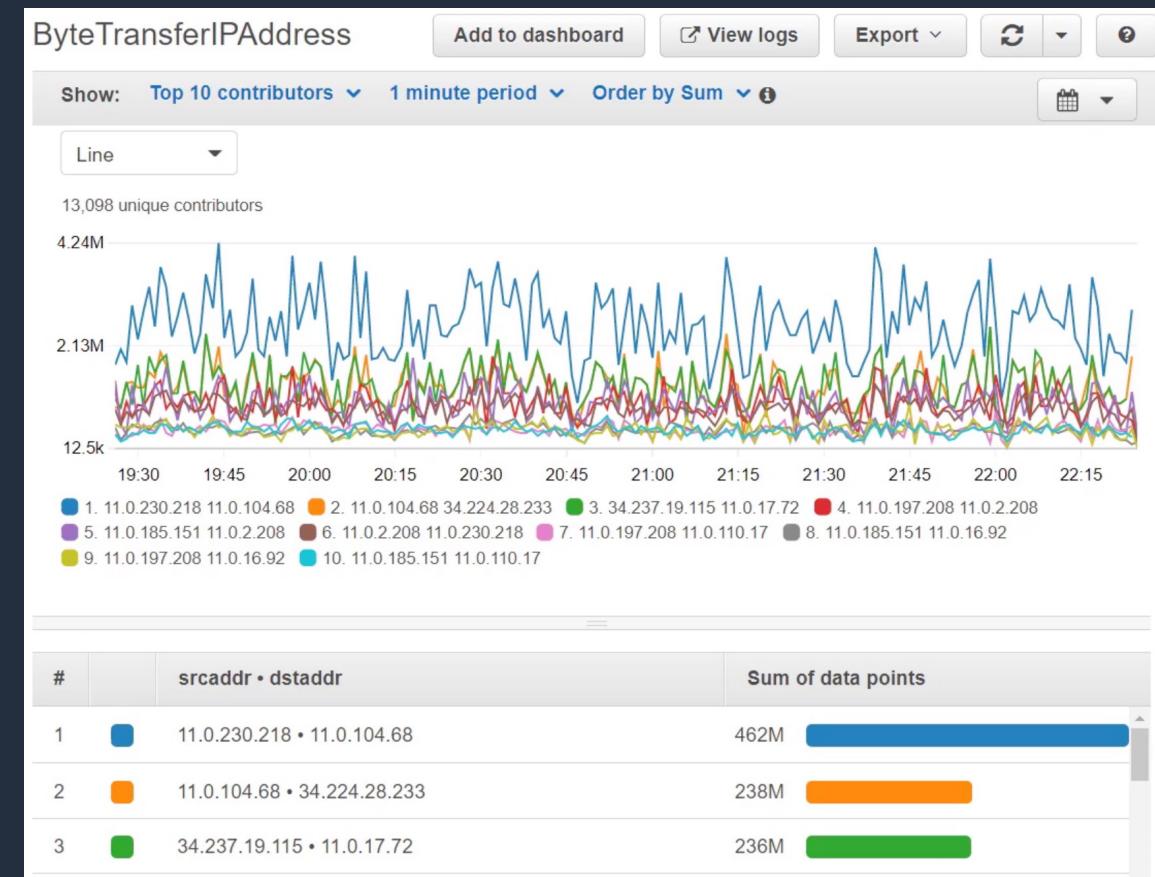
★ここからはその他のサービスを簡単にご紹介  
※個別の詳細は別途Blackbeltセッションで紹介予定

# CloudWatch Contributor Insights

## ログを解析して「上位N」のコントリビューターを見つける

- CloudWatch Logs の構造化されたログデータを解析
- コントリビューターデータを表示する時系列グラフを作成
- システムのパフォーマンスに影響を与えていている人や物を判断するのに有効
  - 例1) エラーを最も多く生成する Product Id
  - 例2) 最もネットワークが重いユーザー

例: VPCフローログで送信元および宛先 IP アドレスごとのデータ数上位3位を見つける

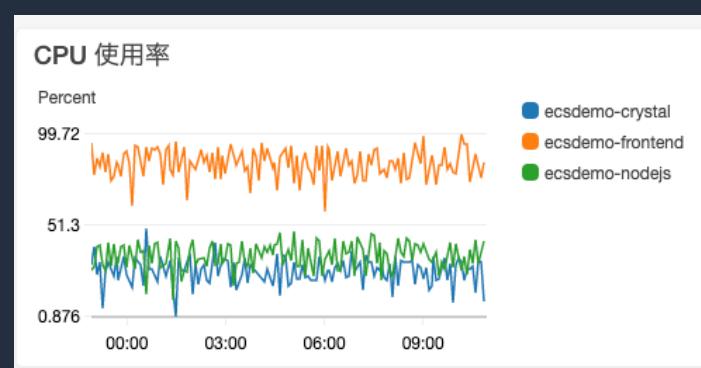


# CloudWatch Container Insights

コンテナのメトリクス

## コンテナ単位でのメトリクスを取得

- コンテナ化されたアプリケーションの メトリクスと ログ を収集、集計、要約できる CloudWatch の機能の一つ
- タスクや Pod、コンテナレベルでのメトリクスの収集が可能
- CPU やメモリ、ディスク、ネットワークなど、多数のリソースのメトリクスを自動的に収集
- 収集されたメトリクスは自動的に作成されるダッシュボードに集約
- Amazon ECS、Amazon EKS (on EC2)、および Amazon EC2 の Kubernetes プラットフォームで利用可能  
ECS サービス



A screenshot of the "Task performance" dashboard. It lists nine tasks under the search term "ecsdemo-frontend". The columns are: Task definition, サービス, 平均 CPU (%), and 平均メモリ (%). The second task in the list has a red box around it, and the entire row is also highlighted with a red box. The data for the highlighted task is as follows:

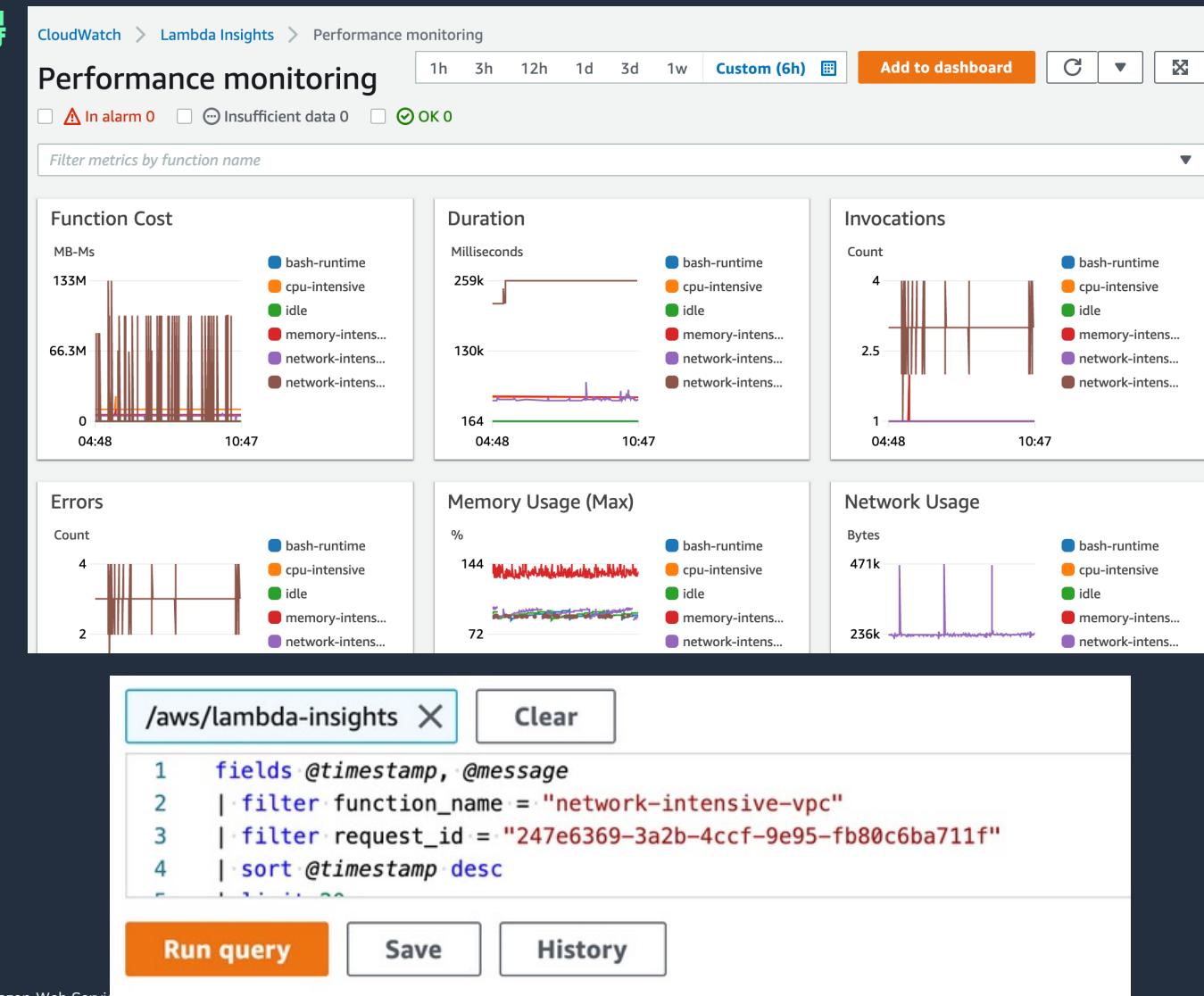
Task definition	サービス	平均 CPU (%)	平均メモリ (%)
ecsdemo-frontend	ecsdemo-frontend	1.6258	16.0156
ecsdemo-frontend	ecsdemo-frontend	99.574	96.0156
ecsdemo-frontend	ecsdemo-frontend	1.5725	16.0156

# CloudWatch Lambda Insights

## Lambda関数のデフォルトに追加でメトリクスを取得

- システムレベルのメトリクスを使用して、Lambda 関数単位でより深い洞察を得ることができる
  - CPU 時間、メモリ、ディスク、ネットワーク 使用率、コールドスタートなど
- 自動で構築されたダッシュボードを使用可能
- Lambda Insights でクエリを実行し、パフォーマンスの問題をトラブルシューティング可能
- Lambda Layer として提供される拡張機能を有効にするだけで使用可能に

## Lambdaのメトリクス



# CloudWatch Application Insights

## 最小限の労力で深いエンタープライズアプリケーションインサイトを

- エンタープライズ アプリケーションのモニタリング
- アプリケーションのリソース全体およびテクノロジースタック (Microsoft SQL Server データベース、ウェブ (IIS) サーバー、アプリケーションサーバー、OS、ロードバランサー、キューなど) で主要なメトリクス、ログ、およびアラームを簡単に設定が可能
- 例えば、CloudWatch Application Insights for .NET and SQL Server は、アプリケーションリソースをスキャンして推奨されるメトリクスとログのリスト (カスタマイズ可能) を提供するため、CloudWatch でこのメトリクスやログを簡単に監視設定できる

Application Insights for .NET and SQL Server [Edit configuration](#)

Problems detected

Severity	Problem summary	Source	Start-time	Status
Medium	EC2: High CPU A problem is going on with the resource awseb-e-jcmmrgsmpx-stack-AWSEBAutoScalingGroup-1VNFYAH14I5RS in DOT_NET_WEB_TIER tier.	CloudWatch: Application Insights Problem Id: p-2fc4d610-2ed3-4da6-8927-b07c30d08c0b <a href="#">Edit configuration</a>	Time range 1h 3h 12h 1d 3d 1w custom (20h)	Actions
Low				

Problem summary

Severity	Problem summary	Source	Start-time	Status	Resource group
Medium	EC2: High CPU	i-0bf88dbe687f32678	2019-07-24T22:29:06Z	In progress	SampleAppGroup

**Insight**

High CPU on the .NET application layer. This may result in application performance degradation due to high load on web servers or application errors. If you experience high load conditions for long periods of time, use Auto Scaling to add additional resources to process the load. To troubleshoot further, collect a full user dump with task manager on the high CPU process and collect PerfMon logs with the thread counter to identify the thread ID causing high CPU.

Help us improve our models:  This insight is useful  This insight is not useful [Submit feedback](#)

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



# 4. 次に何をしたらよいの？



# そもそも、サービスが使っているかを監視したい？

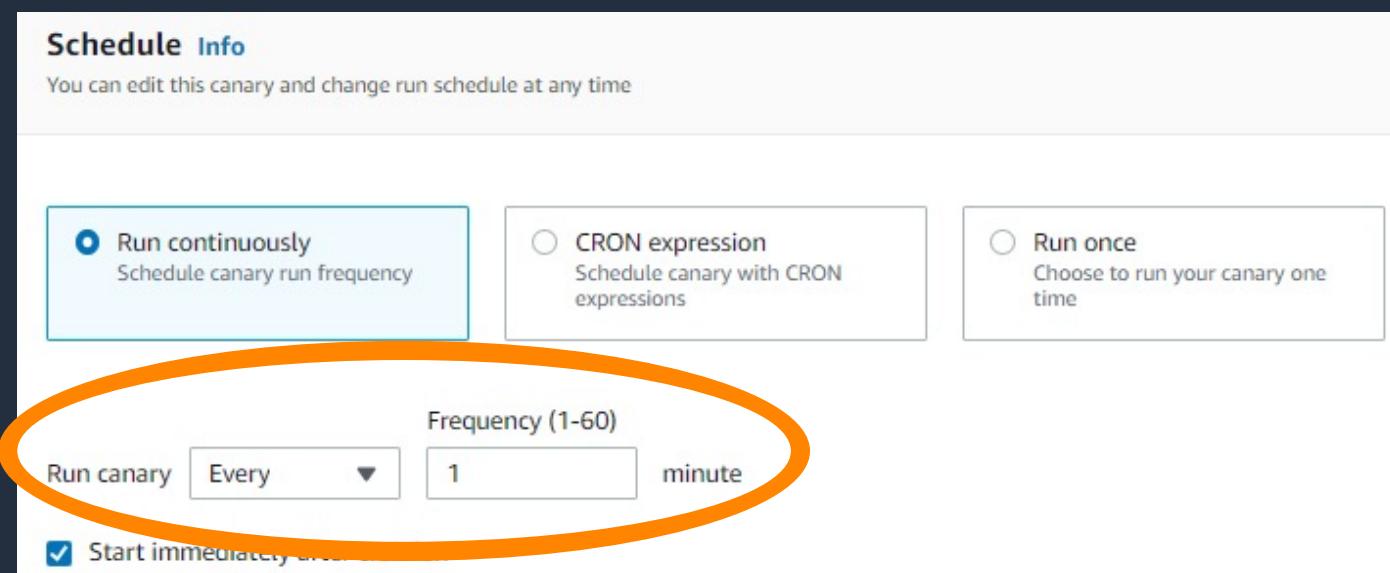
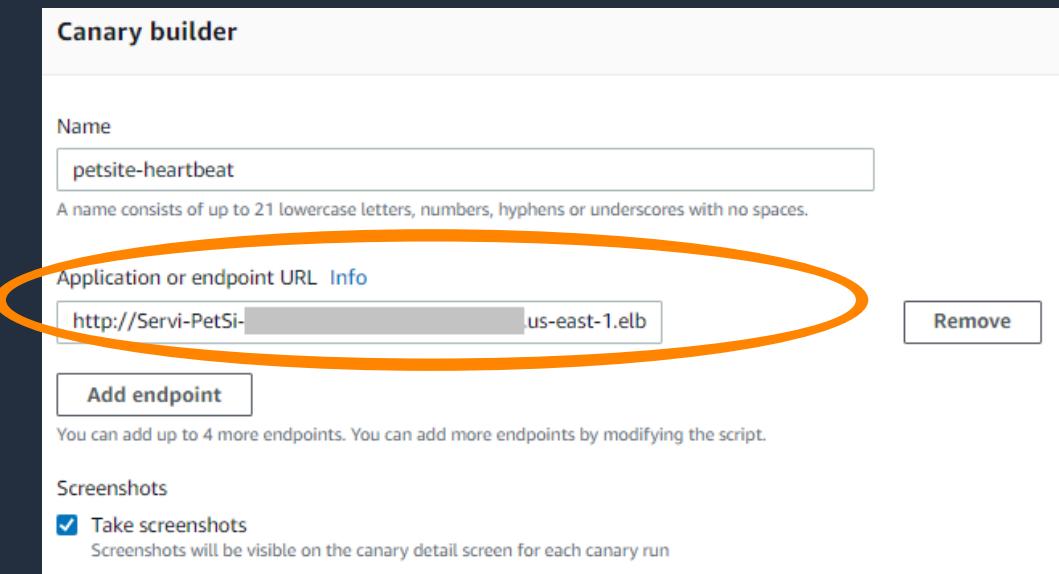
ビジネスに直結する  
メトリクスの取得と監視

→CloudWatch Syntheticsもご活用ください！

- **Canary（カナリア）を作成してサービスのエンドポイントとAPIを監視する**

## Heartbeat Canaryのblueprint例

※blueprintは、Heartbeatのような特定のユースケースのためにカスタマイズ可能なコードを提供

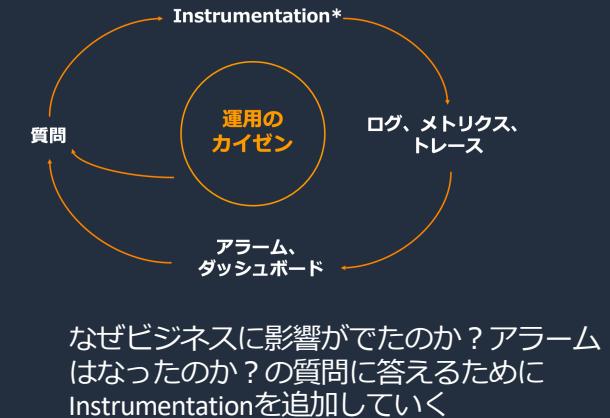
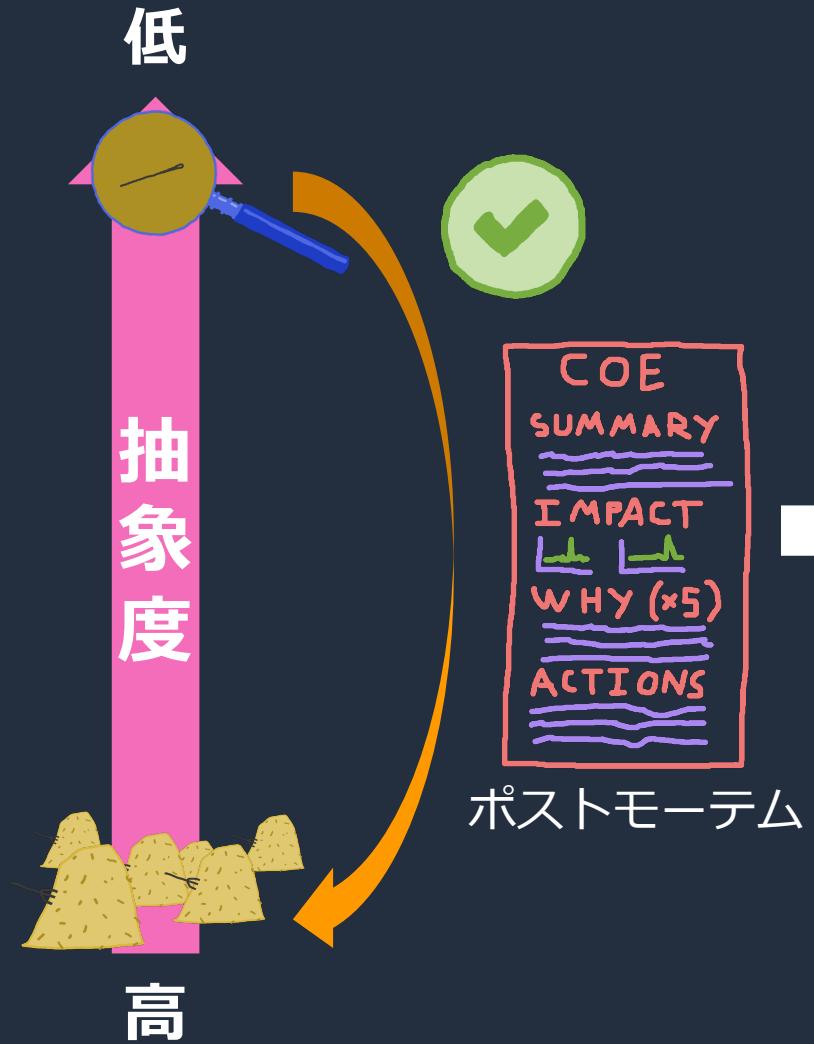
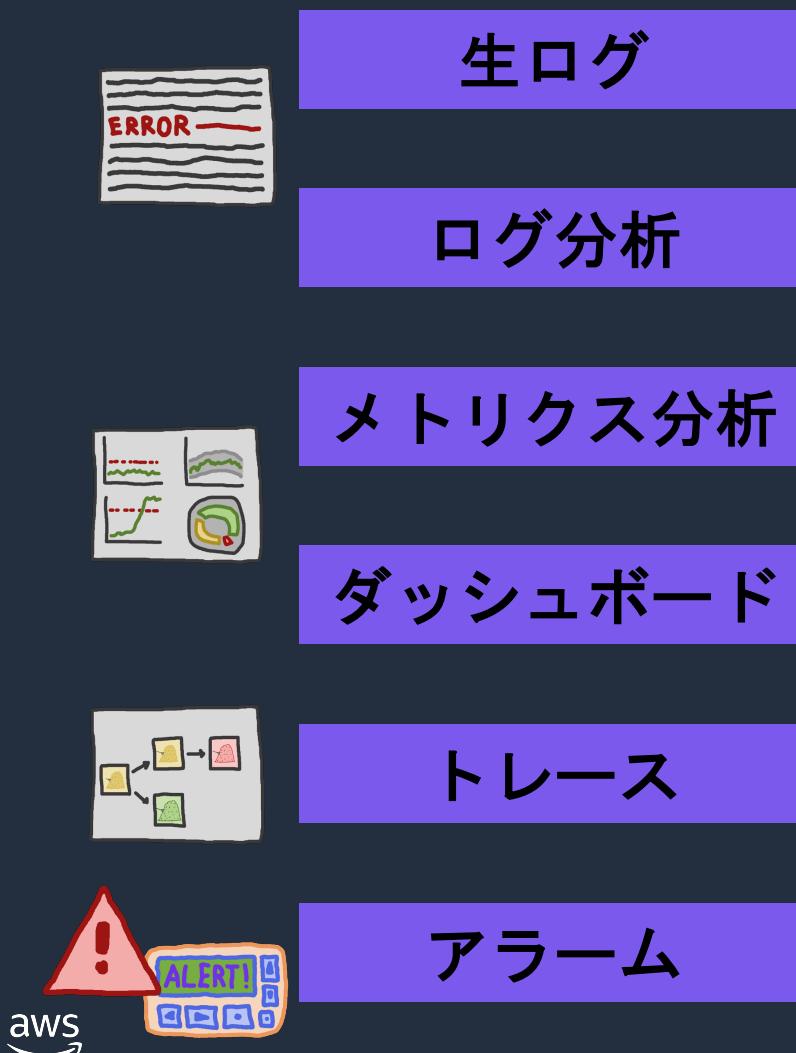


Canaryが1分毎にendpoint URLを  
Heartbeatする

個々の機能やカテゴリについては、別途Blackbeltを公開予定です。  
CloudWatch Syntheticsについても近日公開します。

# で、実際どうやって運用しているの？

## Amazon での対応例



\*Instrumentation  
ログ、メトリクス、トレースなどのデータを取得し、外部に送信できるようシステムに組み込むこと



<https://aws.amazon.com/jp/builders-library/building-dashboards-for-operational-visibility/>

最も重要なテレメトリデータはモニタリングできていますか？

アラームとダッシュボードのチェックで発生する質問に答えられる準備をしておきましょう。

Amazon CloudWatchをはじめとした  
Observability Serviceが力になります！

# 本資料に関するお問い合わせ・ご感想

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

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

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

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

具体的な案件に対する構成相談は、後述する個別相談会をご活用ください



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

# その他コンテンツのご紹介

ウェビナーなど、AWSのイベントスケジュールをご参照いただけます

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

ハンズオンコンテンツ

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

AWS 個別相談会

AWSのソリューションアーキテクトと直接会話いただけます

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



# Thank you!