

Amazon CloudWatch Application Signals ではじめるバーンレートアラーム

Yoshi Yamaguchi (@ymotongpoo)
Senior Developer Advocate
Amazon Web Services Japan, G.K.



自己紹介

山口 能迪 (やまぐち よしふみ)

アマゾンウェブサービスジャパン合同会社
シニアデベロッパーアドボケイト

専門領域

- オブザーバビリティ
- SRE全般



@ymotongpoo





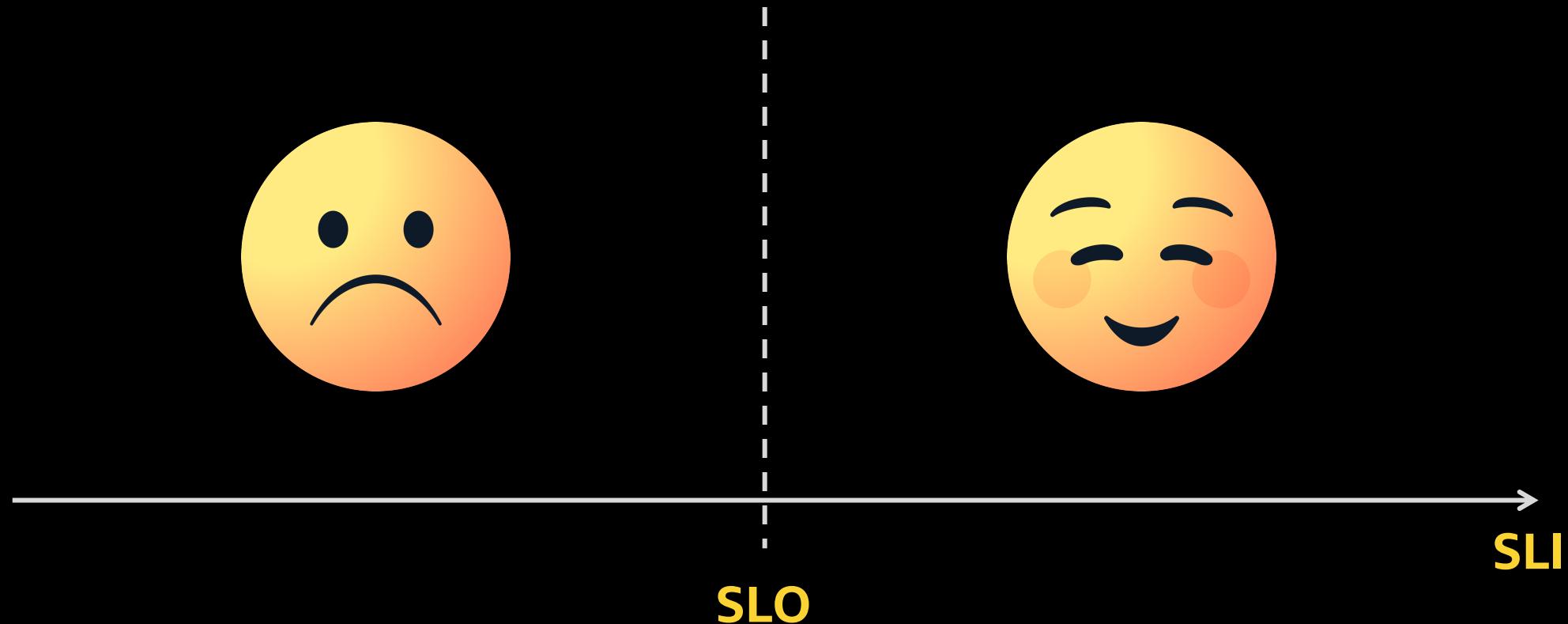
アジェンダ

- バーンレート計測の重要性
- Application Signals による SLO モニタリング

バーンレート計測の重要性

サービスレベル指標 SLI ／ サービスレベル目標 SLO

信頼性 = ユーザーがサービスに期待する性能品質 = システムメトリクスではない



サービスレベル指標 SLI / サービスレベル目標 SLO

SLI = ユーザーがサービスに期待する性能品質

$$\text{SLI} = \frac{\text{良いイベント} / \text{時間}}{\text{全イベント} / \text{時間}}$$

例1: チケットの予約にかかる時間 (レイテンシー)

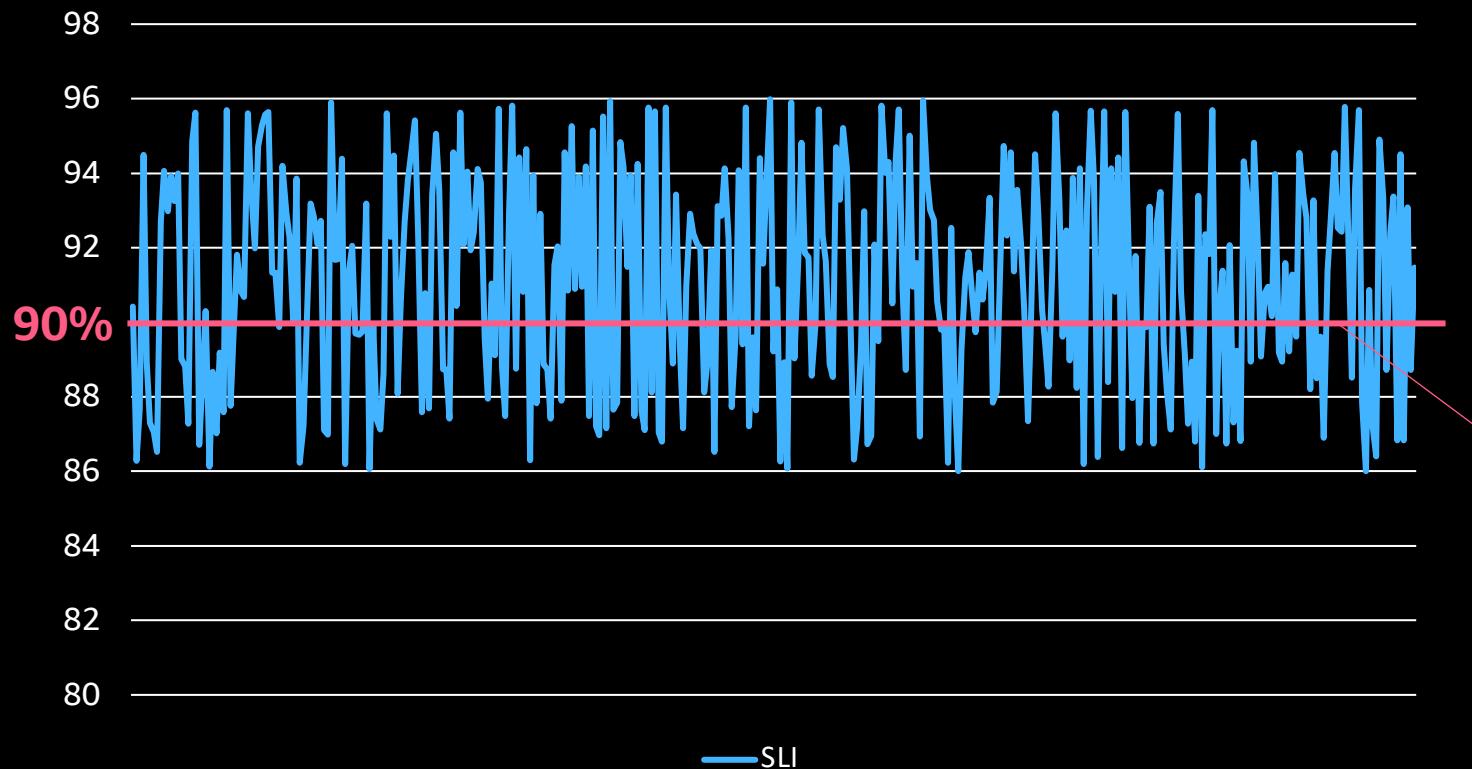
$$\frac{\text{500ms以内のレスポンスの数}}{\text{全レスポンスの数}}$$

例2: 予約ページにアクセスできる時間 (可用性)

$$\frac{\text{アクセスできてた時間}}{\text{計測中の全期間}}$$

SLOだけ見てれば安心！なのか…？

良いレイテンシーの割合



SLI: レイテンシーが1800ms以下
SLO: 28日間で90%

しきい値でアラームを設定してたら…

アラーム疲れ alarm fatigue

重要なアラームが多すぎることで
疲弊し、重要なアラームを見逃してしまった状況。

アラームはアクションが必要な状況でのみ発報すべき

[WAF: OPS08-BP04 Create actionable alerts](#)

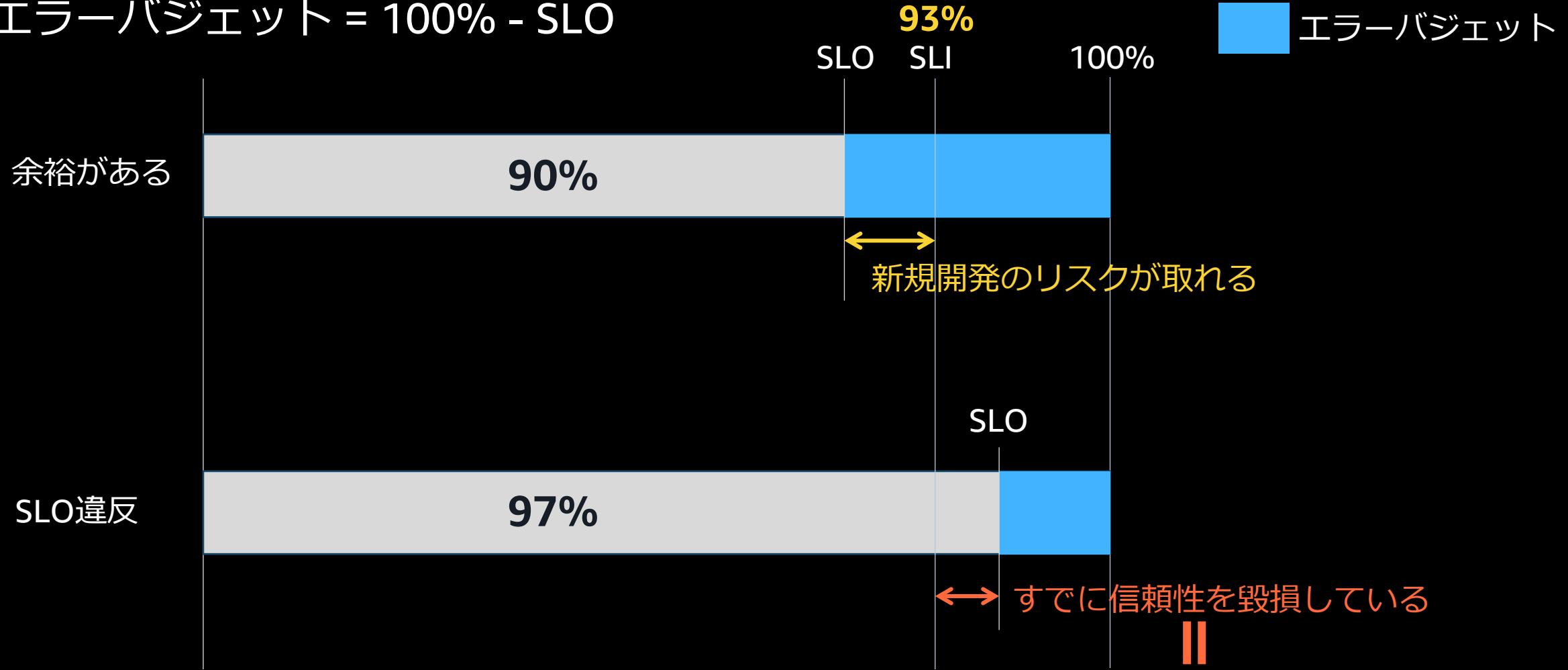
c.f.

- Amazon DevOps Guru
- Amazon CloudWatch 複合アラーム



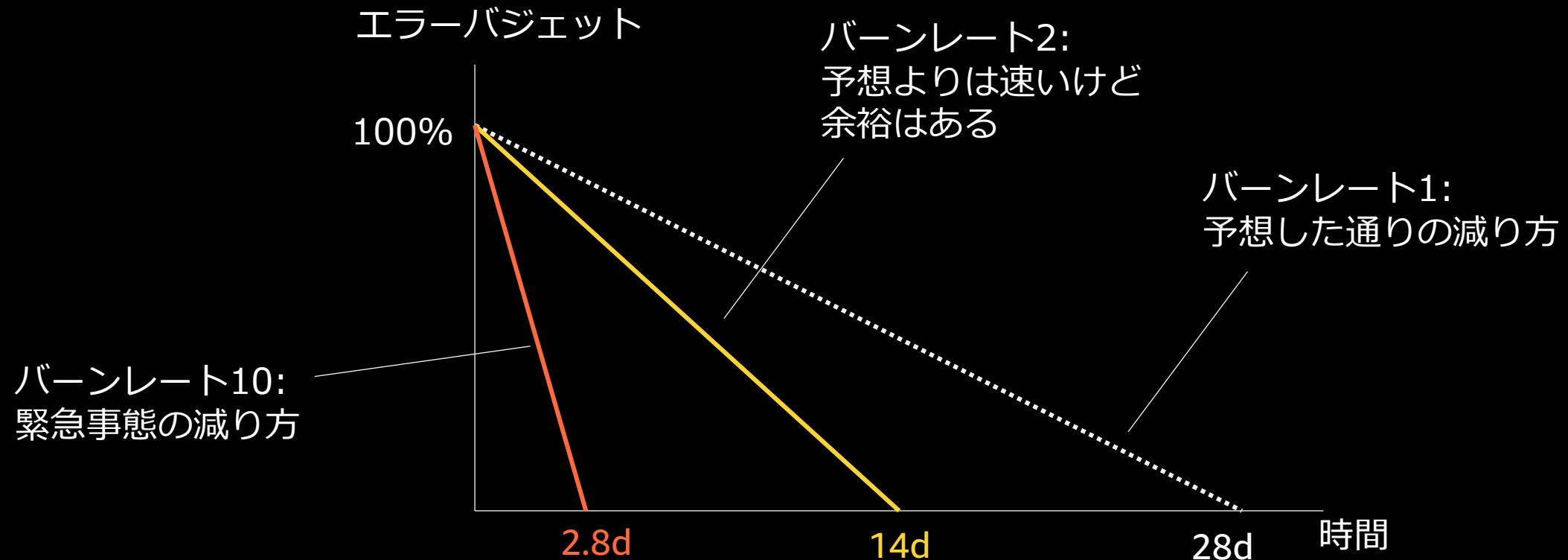
エラーバジェット

エラーバジェット = 100% - SLO



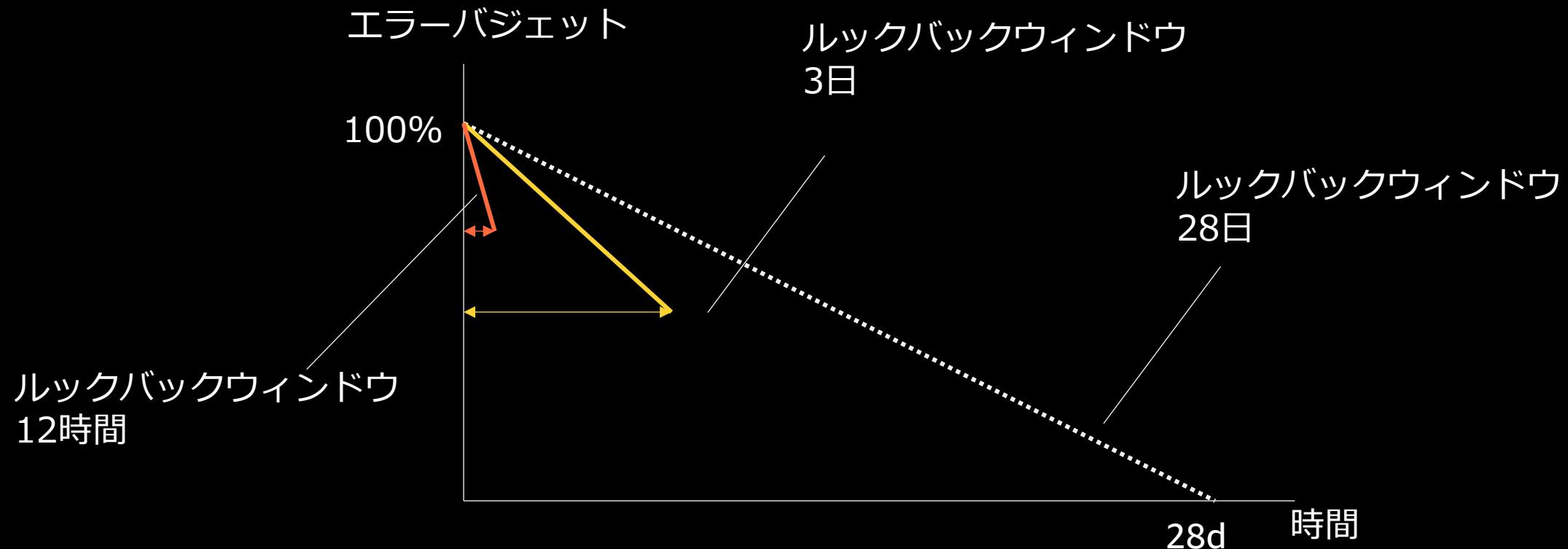
バーンレートアラーム

思ってたよりエラーバジェットが速く減ってたら知らせる

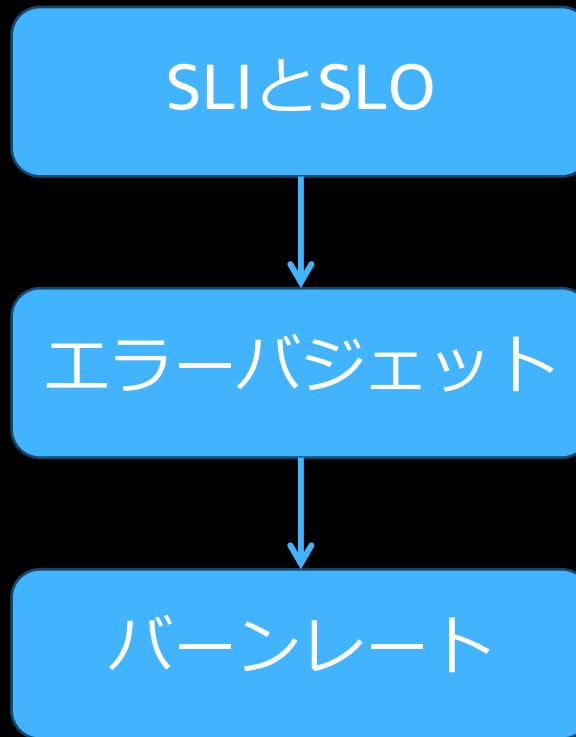


ルックバックウィンドウ

消費速度が速い場合は短い時間で気が付かないと意味がない



関係図



指標と計測期間と目標値を決めるだけ
(割り算の計算式を決めるだけ)

一定期間のデータをローリングで計算し続ける
(1次計算)

エラーバジェットの差分をローリングで計算し続ける
(2次計算)

バーンレートアラームを自分で実装するのは多少手間がかかる

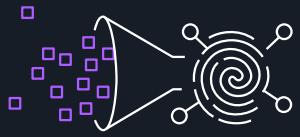
Application Signals による SLO モニタリング



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

Amazon CloudWatch Application Signals

アプリケーションの状態を把握・分析するための情報を自動で収集・可視化



テレメトリデータ
自動収集



サービスの検出/
トポロジー可視化



ダッシュボード提供
トレース・メトリクスの
確認



SLO
モニタリング



CloudWatch Synthetics
/ CloudWatch RUM
との連携

Application Signals がサポートされる環境

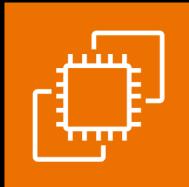
アーキテクチャ



Amazon EKS
EC2 / Fargate



Amazon ECS
EC2 / Fargate
awsvpc ネットワーク
モードのみ



Amazon EC2
コンテナ以外の
アプリケーションも
実装可能

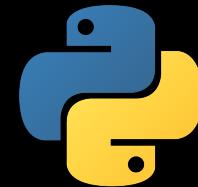


AWS Lambda

プログラム言語



Java



Python



Node.js



.NET

8, 11, 17

3.8+

14, 16, 18,
20, 22

.NET 6, 8
.NET Framework
4.6.2+

Application Signals の有効化の例

Lambda ではワンクリックで設定可能

ここをチェックするだけ →

The screenshot shows the 'Monitoring Tools' configuration page for a Lambda function named 'my-simple-sample'. The 'CloudWatch Applications Signals and AWS X-Ray' section is highlighted with a blue border. Inside this section, the 'Enable Application Signals' checkbox is checked (indicated by a blue outline). A blue arrow points from the text 'ここをチェックするだけ' (Check here) to this checkbox.

モニタリングツールを編集

Amazon CloudWatch 情報

デフォルトでは、Lambda 関数は CloudWatch Logs ストリームと標準メトリクスを生成します。

ログとメトリクス (デフォルト)

CloudWatch アプリケーションシグナルと AWS X-Ray 情報

CloudWatch アプリケーションシグナルと AWS X-Ray は、アプリケーションのパフォーマンスと状態を監視するのに役立ちます。

アプリケーションシグナル 情報

アプリケーションシグナルは、リクエスト、可用性、レイテンシー、エラー、障害などのアプリケーショントレースとメトリクスを収集するのに役立ちます。これらのトレースとメトリクスは [CloudWatch コンソール](#) で確認できます。

有効化

サポートされているランタイム: nodejs18.x, nodejs20.x, nodejs22.x, python3.10, python3.11, python3.12, python3.13, java11, java17, java21, dotnet6, dotnet8. (現在のランタイム: nodejs22.x)

Lambda サービストレース 情報

Lambda サービスの X-Ray トレースは、アプリケーション内のリクエストのパスを追跡するのに役立ちます。これらのトレースは [X-Ray コンソール](#) で確認できます。

有効化

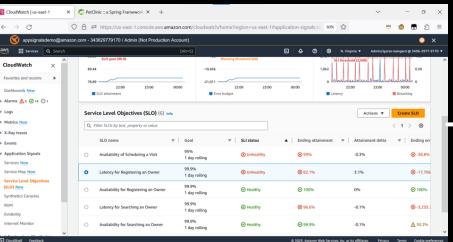
CloudWatch Lambda インサイト 情報

オンにすると、CPU 時間、メモリ、ディスク、ネットワーク使用量などのシステムレベルのメトリクスを収集できます。

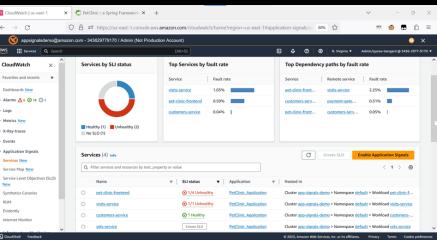
拡張モニタリング

Application Signals のダッシュボード

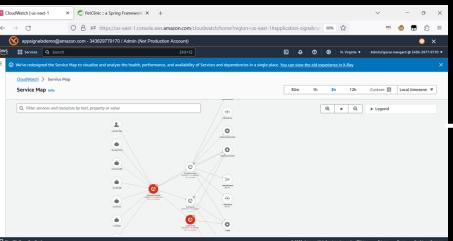
SLO
モニタリング



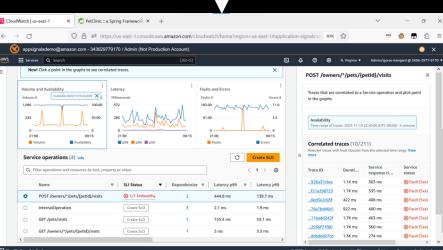
サービス
ダッシュボード



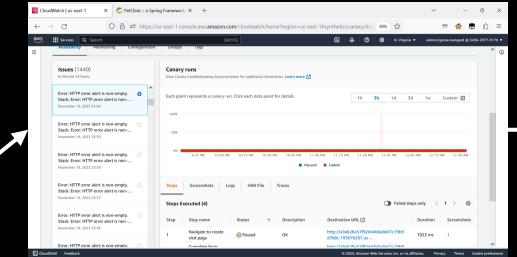
サービスマップ



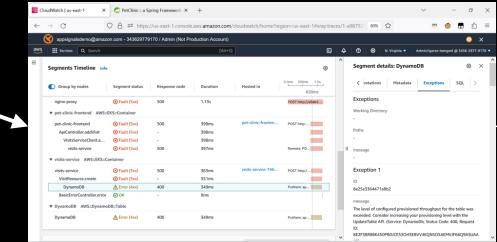
サービス



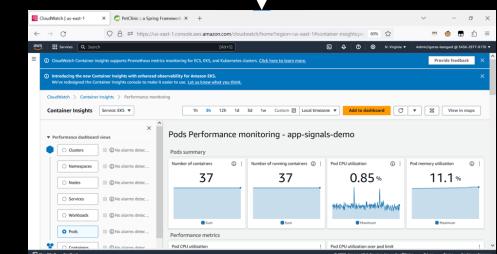
カナリア※



トレース



インサイト、ログ



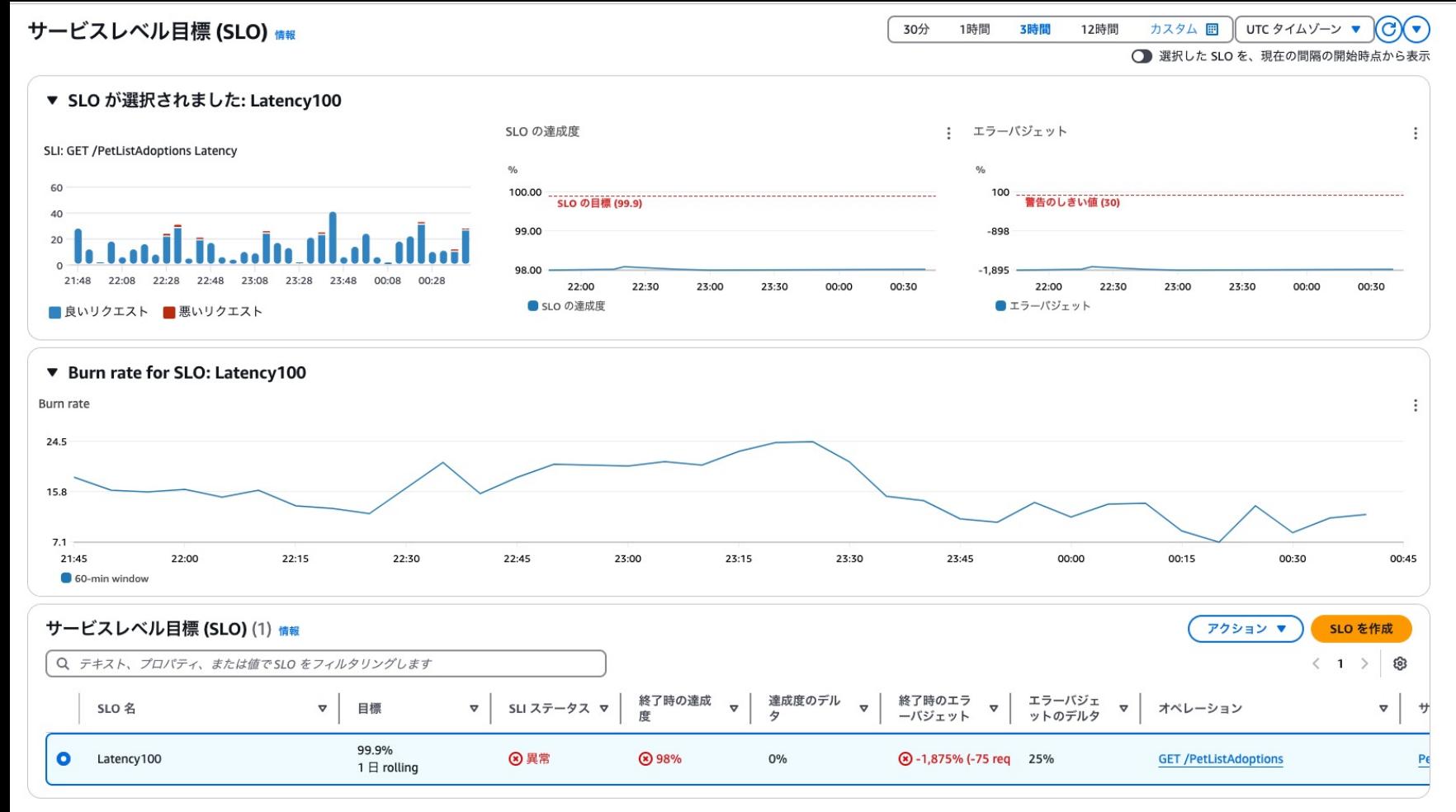
※必ず利用しなければならないわけではありません



SLOダッシュボード



SLIのSLOに対する現状を把握、バーンレートも見える



デモ: SLI→SLO→バーンレートアラーム

SLI の設定

Application Signals ではサービスの操作に関わる指標を SLI とすることを想定

サービスレベル目標 (SLO)

> 「サービスレベル目標 (SLO)」セクション



サービスレベル目標 (SLO) (0) 情報

SLO 名 目標 SLI ステータス ▲ Latest attainment 達成度のデルタ ▼ Latest error budget

SLO なし
SLO は作成されていません

SLO の詳細 [?] SLO を作成

アクション ▾ < 1 > ⚙️

サービス
> 「サービス」セクション



サービス (2) 情報

名前 SLI ステータス サービスの可用性 アプリケーション

<input checked="" type="radio"/> dotnet-cart-api	SLO を作成	36.3%	-
<input type="radio"/> dotnet-delivery-api	SLO を作成	100%	-



サービスレベル目標 (SLO)

> サービスレベル目標 (SLO) を作成

サービスレベル指標 (SLI) を設定 情報

サービスレベルインジケーター (SLI) は、サービスレベル目標 (SLO) を達成しているかどうかを判断するために使用されるメトリックです。

タイプ

サービスオペレーション
検出されたサービスから得られたサービスオペレーションメントリクエストの使用に関する信頼性の目標を設定します。

Dependency
Set a reliability target using a dependency metric from a discovered service.

CloudWatch メトリクス
CloudWatch メトリクスを使用して信頼性の目標を設定します。

サービスを選択 情報

どのサービスについてメトリクスを設定しますか?

dotnet-cart-api

オペレーションを選択 情報

どのサービスオペレーションの信頼性を測定しますか?

POST /cart

計算方法を選択 情報

カスタマーエクスペリエンスの測定にどの計算方法を使用したいですか?

リクエスト
すべてのリクエストに対して適切なリクエストの数を計算します。

期間
指定された時間間隔内の良好な期間の数を計算します。

条件を設定

許容できるエクスペリエンスであることを示すために、どのメトリクスとしきい値を使用しますか?

レイテンシー ▾ は、次以下である必要があります: 1800 ミリ秒.

POST /cart リクエストは、レイテンシーが 1800ミリ秒 に対して 次以下: である場合に、良好かつ正常であるとみなされます。

SLO の設定

フォームに沿って入力していくだけ

サービスレベル目標 (SLO) を設定 情報

サービスレベル目標 (SLO) は、SLI を使用して、顧客の期待に対するサービスの信頼性を定義します。

間隔を設定 | 情報

どのくらいの間隔で SLO を測定したいですか？

28 ローリング 日 ▾

達成目標を設定 | 情報

この期間における SLO 達成目標はどのようなものですか？

90 %.

パーセンテージは 0~100 である必要があります。

警告のしきい値を設定 | 情報

次の場合に SLO を [警告状態] としてマークします: エラーバジェット は次以下です:

30 %

パーセンテージは 0~100 である必要があります。

バーンレートアラーム

CloudWatch アラームまで設定

まずバーンレートを計算する期間を設定

バーンレートを設定 - オプション 情報

バーンレートは、エラーバージェットをどれくらいの速さで消費しているかを示します。

ルックバックウィンドウ(分単位)

2880

480

バーンレートをさらに追加

最大 10 個のバーンレートをセットアップできます



各ルックバックウィンドウでバーンレートアラームを設定

バーンレートアラームを設定 - オプション 情報

バーンレートが設定されたしきい値を超えたときに通知する CloudWatch アラームを作成します

自動的に作成するバーンレートアラームを選択します:

ルックバックウィンドウの分数分の 2880 バーンレートアラーム
Burn rate threshold
Set up threshold to alarm on the burn rate

消費予算のパーセンテージでバーンレートしきい値を設定

バーンレートしきい値を直接設定
Will fire every time burn rate is larger than
2
Value must be a number larger than 1

SNS トピック

通知を受信する SNS (Simple Notification Service) トピックを定義します。

既存の SNS トピックを選択

新しいトピックを作成

新しいトピックを作成...

トピック名は一意である必要があります。

cart-api-post-root-latency-90-slow-burn

SNS トピック名は最大 256 文字で、英数字、ハイフン (-)、アンダースコア (_) のみを使用できます。

通知を受け取る E メールアンドポイント...

メールアドレスのカンマ区切りリストを追加します。各アドレスは上記のトピックに対するサブスクリプションとして追加されます。

team-cart-api@example.com
user1@example.com, user2@example.com.

まとめ

バーンレートアラームは理由がわかりやすい！

Application Signals でバーンレートアラームを設定しましょう！



Thank you!

Any questions?

Yoshi Yamaguchi
@ymotongpoo

A LinkedIn profile card for Yoshi Yamaguchi. At the top is a circular profile picture of him wearing glasses and a dark t-shirt, set against a blue background. To the right is a large, blurred image of the Tokyo skyline at sunset, with the Tokyo Tower prominent. Below the profile picture are three circular icons: a person icon, a search icon, and a 'Follow' button. The main section contains his name, handle, title, bio, and other profile details.

Yoshi Yamaguchi
@ymotongpoo
Senior Developer Advocate at AWS | Observability, SRE/DevOps, Go | ex-Google |
opinions=mine | bsky: [@ymotongpoo.com](https://bsky:@ymotongpoo.com)
 Cloud Services & Solutions  日本 東京  ymotongpoo.com
 Joined April 2007
947 Following **13.6K Followers**