

AWS Black Belt Online Seminar

Amazon Aurora チューニングアプローチ

河合 智彦

Technical Account Manager
2025/09



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
- ・ 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=PLzWGOASvSx6FlwIC2X1nObr1KcMCBBlqY>



ご感想は X (Twitter) へ！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2025年9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。AWS のサービスは常にアップデートを続けているため、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
- ・ 技術的な内容に関しては、有料の [AWS サポート窓口](#)へお問い合わせください
- ・ 料金面でのお問い合わせに関しては、[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)

自己紹介

河合 智彦

アマゾンウェブサービスジャパン
テクニカルアカウントマネージャー

DNB/ISVのお客様を中心にご支援しています。

Amazon Aurora をより身近に感じ、気軽にご利用
ただきたいです。

好きな AWS サービス
Amazon Aurora, AWS Cost Explorer



本セミナーの対象者

- データベースのクラウド移行を検討されている方
- Amazon Aurora の利用を検討中、または今後検討をご予定の方
- Amazon Aurora をご利用中でチューニングを検討中の方

アジェンダ

1. チューニングの流れ、確認方法

- ・性能データ収集、管理に必要とされるもの
- ・Amazon RDS/Aurora で利用可能なモニタリングツール

2. CloudWatch Database Insights の紹介

- ・Performance Insights からの移行
- ・Database Insights スタンダード / アドバンスト モード

3. Cloud Watch メトリクス、Cloud Watch Logs、拡張モニタリングの紹介

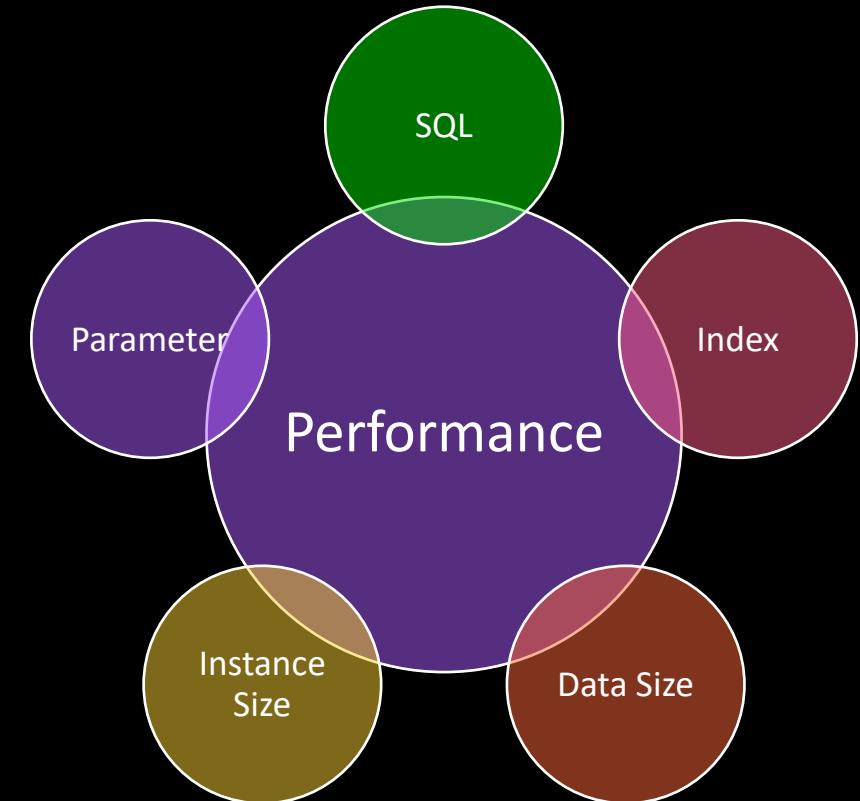
4. 待機イベントの見方

- ・データベースロード
- ・待機イベントについて
- ・データベースロードを待機イベントで分析

チューニングの流れ、確認方法

サービス最適化

- ・パフォーマンスチューニング工数の考慮
 - ・目標スケジュール、KPI、優先順位の設定
- ・個別対応・全体的な対応?
 - ・インスタンス全体 (パラメータ: メモリ、I/O等)
 - ・特定処理の最適化 (SQL最適化: SQL、インデックス等)
 - ・短期的・長期的な対応が必要か?



管理工数を抑え、継続的にデータベースを監視・管理して変化を検知
≒ 安定したサービスを提供し、常に最適なサービスを利用者に提供

一般的な性能管理のサイクルの課題

コレクション (データ収集)

アナリシス (分析)

チューニング (最適化)

- 細かい粒度のパフォーマンスデータを長期間にわたり、収集、管理していくのは運用コストが高い
 - OSのメトリクスだけではなくDB特有のメトリクスを対象データベースに負荷なく細かい粒度で取得することが困難
 - データ量が多くなる中で安定したパフォーマンス分析インフラを維持管理するコストが高い

- パフォーマンス問題のような複雑で広範な事象を特定の統計情報から分析する的是困難
- どの程度リソースが不足、余剰なのかを見積もることが困難もしくは不可能

- ボトルネックの分析が曖昧、不十分なため、SQLやDBの効果的なチューニングが困難もしくは非効率
 - どの程度チューニングされたら成功とみなすか判断できない
 - リアルタイムでDB全体のパフォーマンス測定できないため、チューニング効果の確認に時間がかかる

性能データ収集、管理に必要とされるもの

コレクション (データ収集)

アナリシス (分析)

チューニング (最適化)

- 突発的なパフォーマンス障害に対応するためには細かい粒度のデータが必要
- 過去に発生した問題を分析するためにはある程度長期間パフォーマンスデータの保存、削除するといった管理が必要
- パフォーマンス指標として一般的なOSのリソース情報やDB内部の動作を推測するためのDB統計情報に加えてDBのワークロードを端的に示す指標が必要

- シンプルなパフォーマンス指標を元に、該当時間帯でパフォーマンス問題が発生しているか否かを端的に理解できる必要がある
- パフォーマンス問題が発生している場合、問題の原因をドリルダウンして究明できる必要がある
- チューニングした後の姿を予測し、チューニングのゴールを明確にしておく必要がある

- チューニング対象におけるボトルネックを排除し、事前に設定したチューニングのゴールに向けて最適化を実施
 - 推測ではないエビデンスに基づいたチューニング
 - データ収集 → 分析の流れがシステム化されていれば、チューニング後即座に効果の測定が可能

性能データ収集、管理に必要とされるもの

コレクション

(データ収集)

アナリシス

(分析)

チューニング

(最適化)

パフォーマンスデータを平時から収集、管理し、有事には即座にそのデータを活用、分析できるインフラが必要

さらに、突発的なパフォーマンス障害を想定して**リアルタイムに近いデータ収集、分析**を可能にすることが望ましい

- チューニング対象におけるボトルネックを排除し、事前に設定したチューニングのゴールに向けて最適化を実施
 - 推測ではないエビデンスに基づいたチューニング
 - データ収集 → 分析の流れがシステム化されていれば、チューニング後即座に効果の測定が可能

Amazon RDS/Aurora で利用可能なモニタリングツール

ツール名	できること	設定方法
<u>CloudWatch</u>	Hypervisor等の各種メトリクスをデフォルト60秒間隔で取得・確認可能にする	<ul style="list-style-type: none">デフォルト有効 (無効化不可)
<u>Amazon RDS 拡張モニタリング</u>	OSレイヤーのCPU利用率やメモリなどの各種メトリクスを1、5、10、15、30、60秒間隔で取得・確認可能にする	<ul style="list-style-type: none">インスタンス作成時に有効/無効を指定 (デフォルト有効)インスタンス作成後に変更可能 (インスタンス再起動不要)
<u>CloudWatch Database Insights</u>	データベース内の性能情報を蓄積・視覚化し、15カ月過去に遡ってワークロード分析を可能にする※1	<ul style="list-style-type: none">インスタンス作成時にスタンダードかアドバンストを選択インスタンス作成後に変更可能

参照:

AWS ブログ : [Amazon RDS と Amazon Aurora のパフォーマンスとイベントの可視性を高める](#)

AWS ドキュメント : [Amazon Aurora のメトリクスのモニタリングの概要](#)

re:Post : [Amazon RDS for MySQL DB インスタンスのパフォーマンスをモニタリングする方法を教えてください。](#)

※1 性能情報の保持期間はアドバンストモードの場合であり、スタンダードモードの場合はワークロード分析においても利用できない機能があります

トップダウンからの分析



リソース監視

- Throughput
- Latency
- Utilization – CPU,IOPS,Memory
- Database availability
- Authentication failure
- Slow running queries

データベースの状況確認

- DB Load Analysis – SQL,waits,hosts,users,databases
- Database Telemetry – Metrics,Logs,Slow SQL Queries, Events, OS Processes
- Performance Analysis
- Calling Services

短いデータ収集間隔での詳細確認

- Cloud Watch Dashboard with Enhanced Monitoring
- OS process list

CloudWatch
によるリソース監視

Database Insights
による分析

拡張モニタリング
による詳細分析

CloudWatch Database Insights の紹介

Performance Insights からの移行

2025年9月時点ではデータベースのパフォーマンスチューニングとモニタリングを行う Amazon RDS Performance Insights が利用可能です。

2026 年 6 月 30 日に、RDS コンソールの Performance Insights ダッシュボード [1] とフレキシブルな保持期間とその価格設定 [2] が廃止となります。引き続き Performance Insights のメトリクスを表示するには、CloudWatch Database Insights を利用します。

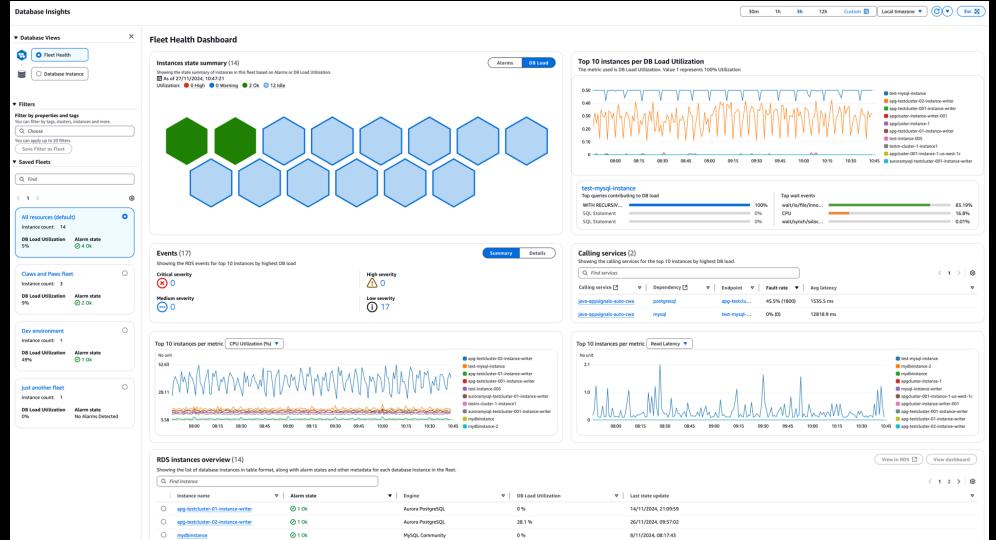
- Performance Insights (PI)ダッシュボードの廃止 ※PIのAPIは継続利用可能
- スタンダードモードとアドバンストモードがあり新規作成時に指定
- 料金設定に伴う柔軟な保持期間の廃止
 - Database Insights スタンダードモードは7日で固定 ※無料利用枠あり
 - Database Insights アドバンストモードは15か月で固定
- RDS for Oracle および RDS for SQL Server の実行計画キャプチャ機能は、Database Insights のアドバンストモードに移行
- RDS PostgreSQL、Aurora PostgreSQL、Aurora MySQL のオンデマンド分析機能は、Database Insights のアドバンストモードに移行

[1] https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/UserGuide/USER_PerfInsights.UsingDashboard.html

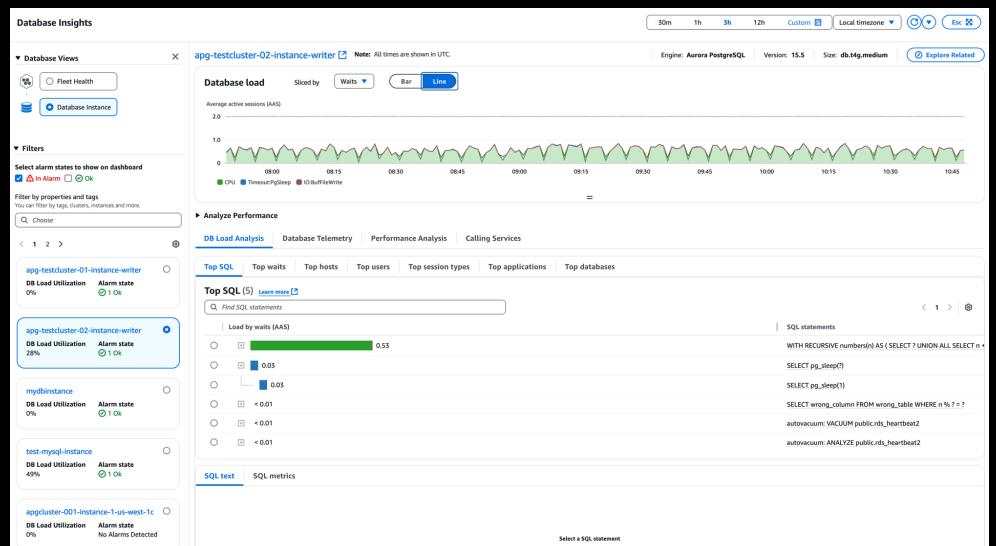
[2] <https://aws.amazon.com/rds/performance-insights/pricing/>

Database Insights について

- CloudWatchにおける統一されたビューでのすべてのデータベースステレメトリの統合
- 数百のデータベースにわたる設定可能なフリート全体のビュー
- 詳細なSQLクエリメトリクス
- 呼び出し元アプリケーションサービスと依存するデータベース間の依存関係マッピングによるアプリケーションコンテキスト
- 事前構築された健全性ダッシュボード、事前定義されたしきい値、推奨事項により簡単に着手可能
- フルマネージドのエクスペリエンス - エージェントやメンテナンスの手間なし



Fleet view



Instance view



Database Insights スタンダード / アドバンスト モード

機能	スタンダードモード (7 Days)	アドバンストモード (15 month)
DB Load の上位の寄与要因をディメンション別で分析する	サポート	サポート
最大7日間の保持期間を持つデータベースメトリクスにクエリ、グラフ、アラーム設定をする	サポート	サポート
きめ細かなアクセスコントロールポリシーを定義し、SQL テキストなどの機密性の高いディメンションへのアクセス権を制限する	サポート	サポート
実行中のプロセスごとに詳細なメトリクスを使用して、データベースで発生しているオペレーティングシステムのプロセスを分析する この機能を使用するには、[Amazon RDS 拡張モニタリング] が有効になっている必要があります。	サポート外	サポート
フリート全体のモニタリングビューを作成して保存し、何百ものデータベースの健全性を評価する	サポート外	サポート
15ヶ月の保持期間とガイド付き UX の環境で SQL ロックを分析する	サポート外	Aurora PostgreSQL でのみサポート
15ヶ月の保持期間とガイド付き UX の環境で SQL 実行プランを分析する	サポート外	Aurora PostgreSQL、RDS for Oracle、および RDS for SQL Server でのみ対応
遅い SQL クエリを分析する	サポート外	サポート
メトリクス、ログ、イベント、アプリケーションを含め、すべてのデータベーステレメトリの統合ダッシュボードを表示する	サポート外	サポート
CloudWatch で Amazon RDS イベントを表示する	サポート外	サポート
オンデマンド分析を使用し、選択した期間のデータベースのパフォーマンスを分析する	サポート外	Aurora PostgreSQL、Aurora MySQL、RDS for PostgreSQL、RDS for MySQL、および RDS for MariaDB でのみ対応

参考：[Database Insights のモード](#)

データベースビュー

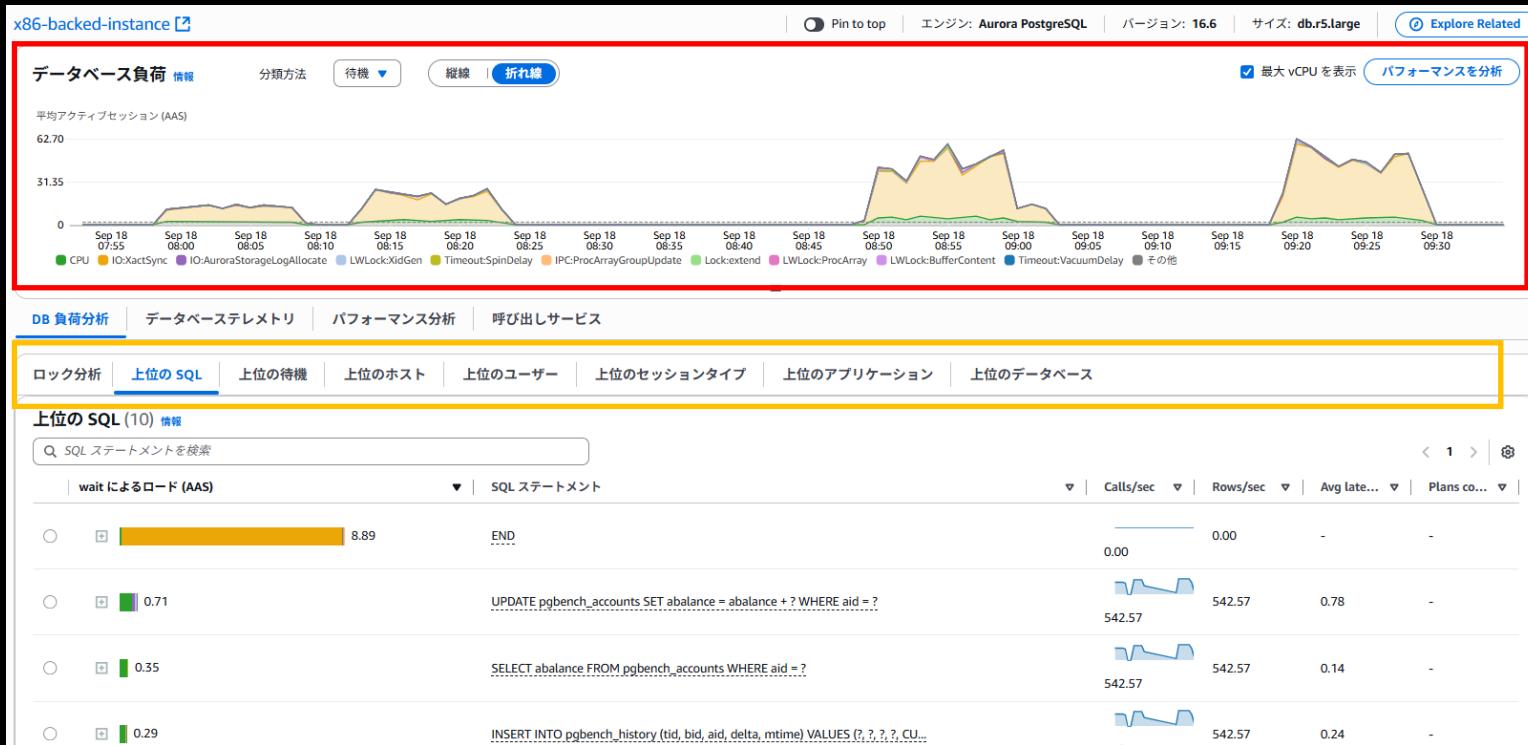
フリートの状態 (Fleet view)



- ヒートマップ (高負荷 + アラームの発生しているインスタンス)
- 高負荷な上位10インスタンス
- DB負荷に影響を与える項目の確認
 - 上位のSQLクエリ
 - 上位の待機イベント
 - 上位のアプリケーション/サービス
- 失敗した直近のデータベースイベント
- 上位10インスタンスの主要なメトリクスのウィジェット表示
- インスタンスリスト

データベースビュー

データベースインスタンス (Instance view)



「データベース負荷」

- データベース負荷指標の確認

「DB負荷分析」

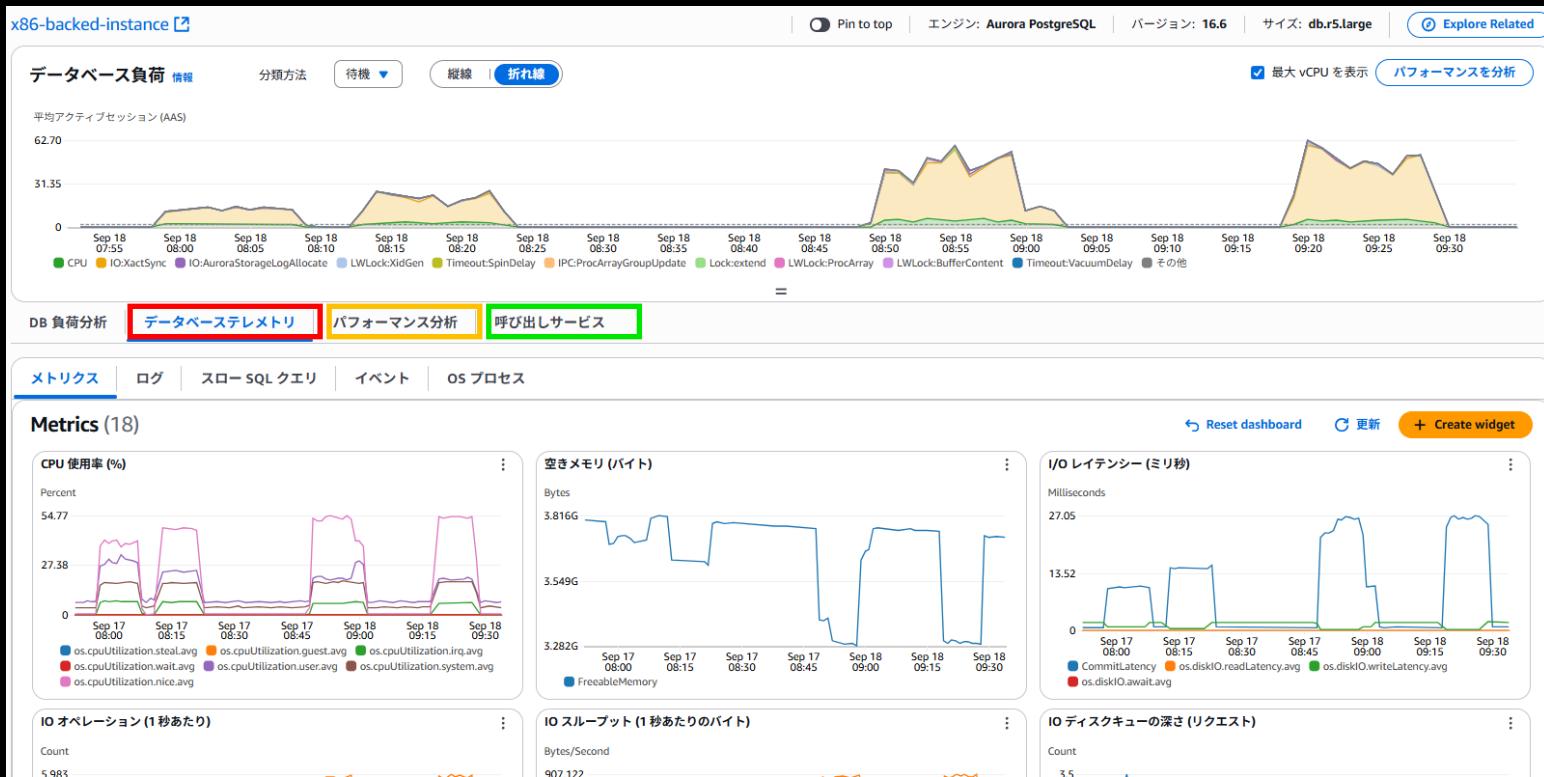
- データベース負荷に影響を与えるTop Nの確認

- 上位の SQL
- 上位の待機イベント
- 上位のホスト
- 上位のユーザ
- 上位のデータベース
- ロック分析※1
- 上位のセッションタイプ※2
- 上位のアプリケーション※2

※1 ロック分析がサポートされているDBエンジンのみ
※2 Aurora PostgreSQL のみサポート

データベースビュー

データベースインスタンス (Instance view)



「データベースステレメトリ」

- メトリクス、ログ、スローSQLクエリ、イベント、OSプロセスを単一のビューに統合

「パフォーマンス分析」

- 高度な根本原因分析

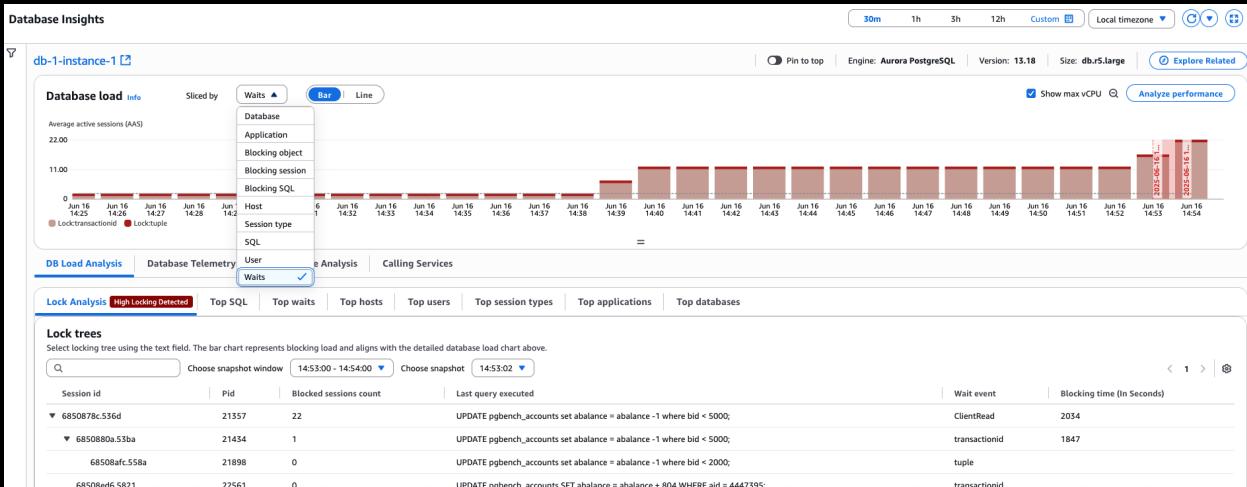
「呼び出しサービス」

- 呼び出し元サービスとオペレーションの確認

Database Insights の機能

アドバンストモードで利用可能

ロック分析



Aurora PostgreSQL のみ対応
2025/9 時点

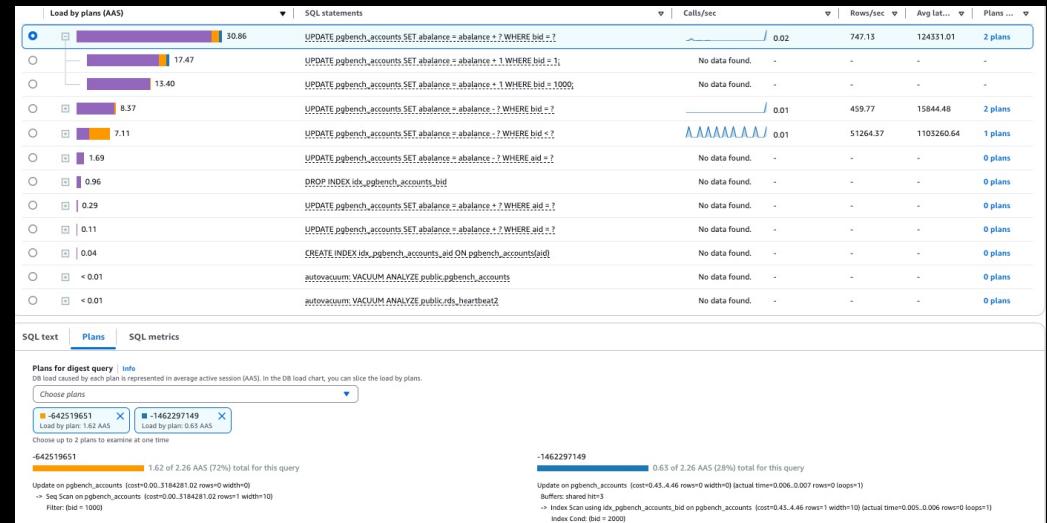
参照:

[Amazon Aurora PostgreSQL のロックツリーの分析](#)



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

実行プランの分析

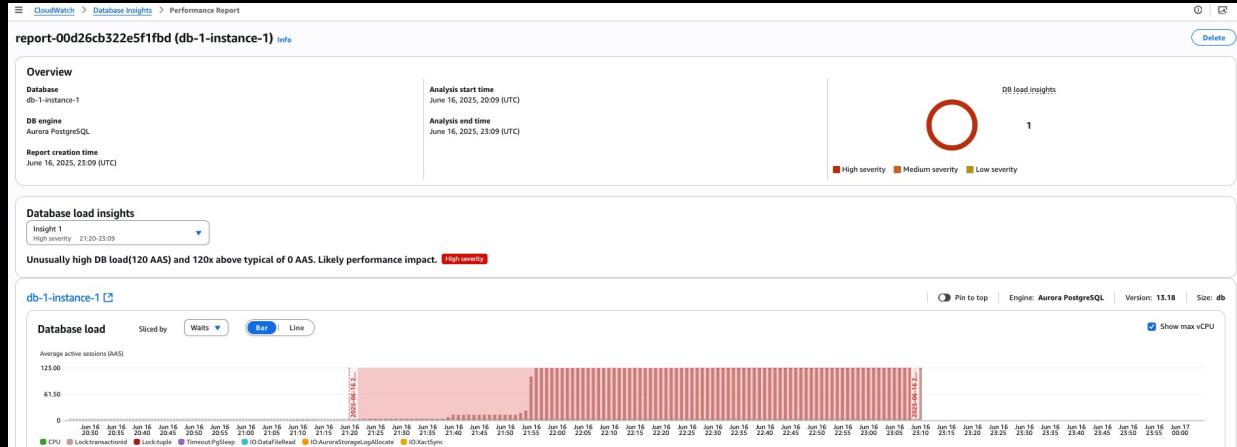


Aurora : PostgreSQL,
RDS : Microsoft SQL Server Oracleのみ対応
2025/9 時点
参照:
[実行プランの分析](#)

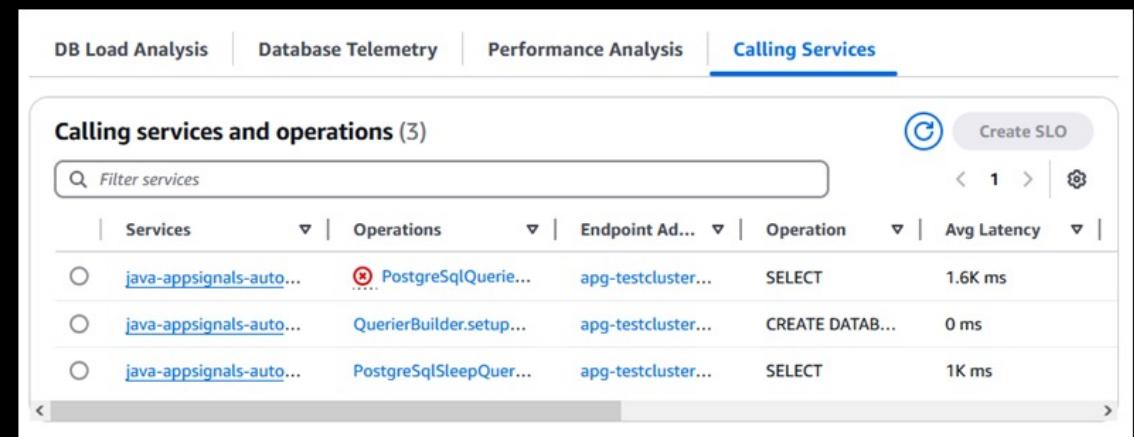
Database Insights の機能

アドバンストモードで利用可能

オンデマンド パフォーマンス分析



Application performance monitoring (APM)



参照:

[データベースのパフォーマンスをオンデマンド分析する](#)



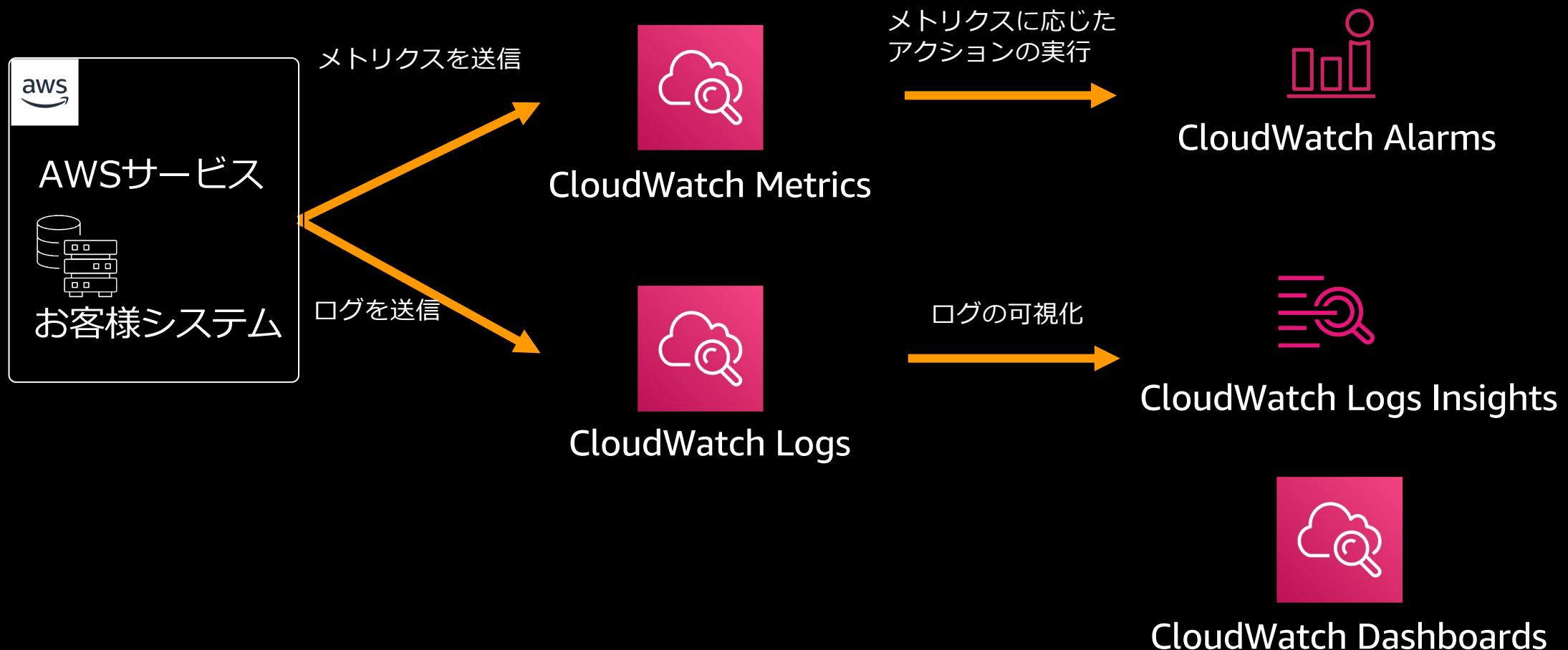
© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

参照:

[アプリケーションパフォーマンスマニタリング \(APM\)](#)

Cloud Watch メトリクス Cloud Watch Logs 拡張モニタリングの紹介

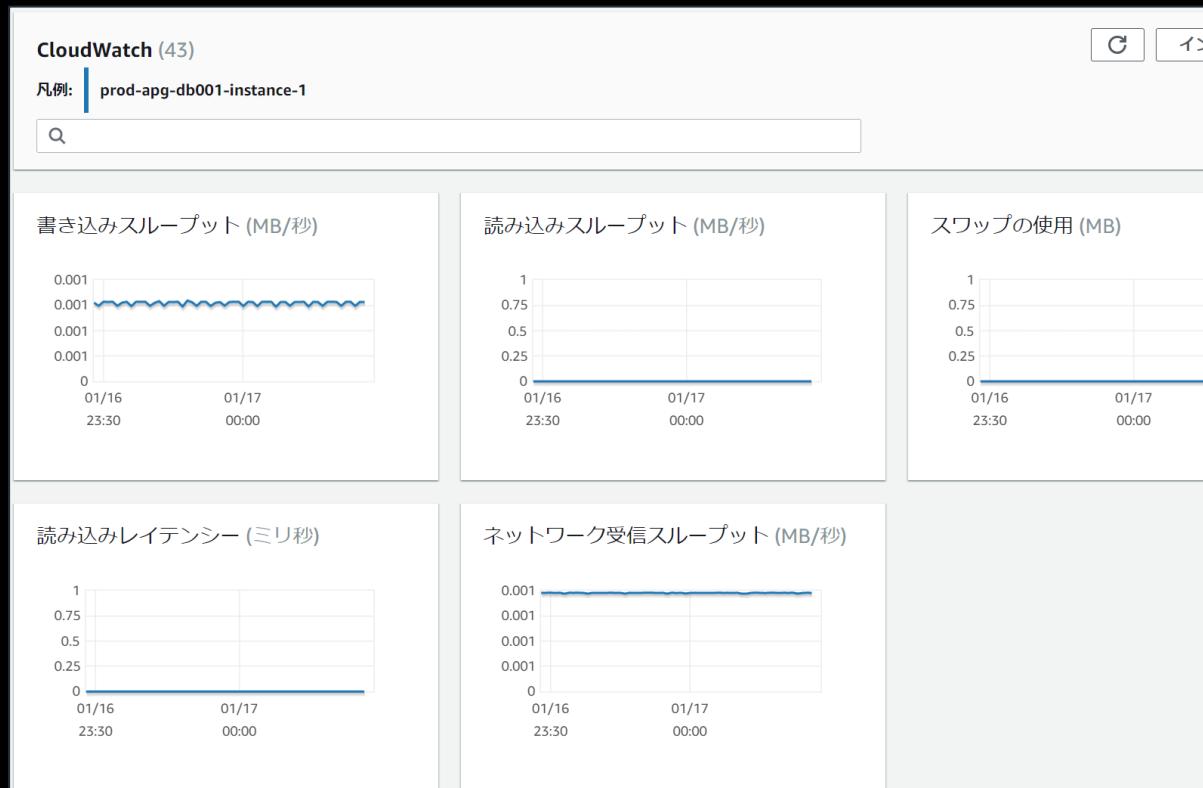
モニタリング基盤としてのAmazon CloudWatch



メトリクスとアラーム

CloudWatch Metrics

標準的なメトリクスを60秒間隔で収集



CloudWatch Alarm メトリクスの状態による通知

The configuration screen for a CloudWatch Alarm:

メトリクス

グラフ
このアラームは青線が5分内の1データポイントで上回る赤線を超える場合に、トリガーされます。

名前空間: AWS/RDS
メトリクス名: DBLoad
DBInstanceIdentifier: ams-v3-02-1-instance-1
統計: 平均値
期間: 5分

条件

しきい値の種類:
 静的
値をしきい値として使用
 異常検出
バンドをしきい値として使用

DBLoad が次の時...
アラーム条件を定義します。

より大きい
> しきい値
 以上
≥ しきい値
 以下
≤ しきい値
 より低い
< しきい値

...よりも
しきい値を定義します。
10000
数字である必要があります



CloudWatch Logs

Amazon RDS、Amazon Auroraでは「ログのエクスポート」を設定することで、CloudWatch Logsへ連携が可能

- 監査ログ/エラーログ/一般ログ/スロークエリログなど

- 監査ログ/エラーログ/一般ログ/スロークエリログなど

CloudWatch > ロググループ > /aws/rds/cluster/auroralab-mysql-cluster/slowquery > auroralab-mysql-node-1

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](#)

🔍 イベントをフィルター - Enter キーを押して検索

クリア1分30分1時間12時間カスタムローカルタイムゾーン表示

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。 さらにロードします。	
▼ 2025-09-11T07:51:42.581+09:00	# Time: 2025-09-10T22:51:42.581370Z # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2526 # Query_time: 43.580308 Lock_time: 38.766561 Rows_sent: 0 Rows_examined: 31... # Time: 2025-09-10T22:51:42.581370Z # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2526 # Query_time: 43.580308 Lock_time: 38.766561 Rows_sent: 0 Rows_examined: 3196841 SET timestamp=1757544659; UPDATE mylab.weather SET max_temp = 42 WHERE id='USC00046699'; # Time: 2025-09-10T22:51:46.938+09:00 # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2519 # Query_time: 43.018324 Lock_time: 0.000000 Rows_sent: 0 Rows_examined: 31... # Time: 2025-09-10T22:51:50.041+09:00 # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2519 # Query_time: 3.101420 Lock_time: 0.000003 Rows_sent: 1 Rows_examined: 319... # Time: 2025-09-11T07:51:51.727+09:00 # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2527 # Query_time: 43.108047 Lock_time: 0.000000 Rows_sent: 0 Rows_examined: 31... # Time: 2025-09-11T07:51:54.109+09:00 # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2519 # Query_time: 4.066375 Lock_time: 0.000002 Rows_sent: 0 Rows_examined: 319... # Time: 2025-09-11T07:51:55.705+09:00 # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2527 # Query_time: 3.957913 Lock_time: 0.000002 Rows_sent: 1 Rows_examined: 319... # Time: 2025-09-11T07:51:57.938+09:00 # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2518 # Query_time: 47.792960 Lock_time: 41.583322 Rows_sent: 0 Rows_examined: 3... # Time: 2025-09-11T07:51:59.943+09:00 # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2527 # Query_time: 4.237675 Lock_time: 0.000003 Rows_sent: 1 Rows_examined: 319... # Time: 2025-09-11T07:52:03.375+09:00 # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2518 # Query_time: 5.419015 Lock_time: 0.000002 Rows_sent: 0 Rows_examined: 319... # Time: 2025-09-11T07:52:04.256+09:00 # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2527 # Query_time: 4.310388 Lock_time: 0.000003 Rows_sent: 0 Rows_examined: 319... # Time: 2025-09-11T07:52:05.175+09:00 # User@Host: administrator[administrator] @ [172.30.0.63] Id: 2523 # Query_time: 50.618817 Lock_time: 43.383179 Rows_sent: 0 Rows_examined: 3...

ログのエクスポート

Amazon CloudWatch Logs に発行するログタイプを選択します

- 監査ログ
 - エラーログ
 - 全般ログ
 - iam-db-auth-error ログ
 - instance ログ
 - スロークエリログ

Aurora MySQL の場合の選択肢

- 任意のクエリを選択して展開すると、そのクエリに関する詳細情報を確認可能

CloudWatch Logs Insights

CloudWatch Logs の可読性を向上させるために CloudWatch Logs Insights を使用

The screenshot shows the AWS CloudWatch Logs Insights interface. At the top, there's a navigation bar with 'CloudWatch' and 'ログのインサイト'. Below it, a sub-header says 'OpenSearch による分析 - 新規'. The main area is titled 'ログのインサイト' with a '情報' tab selected. It shows a query editor with the following SQL-like code:

```
1 filter @LogStream = 'auroralab-mysql-node-1'
2 | parse @message "# Time: * User@Host: * Id: * Query_time: * Lock_time: * Rows_sent: * Rows_examined: * timestamp=;" as Time, User, Id, Query_time, Lock_time, Rows_sent, Rows_examined, Timestamp, Query
3 | display Time, Query_time, Query, Rows_sent, Rows_examined
4 | sort Query_time asc
```

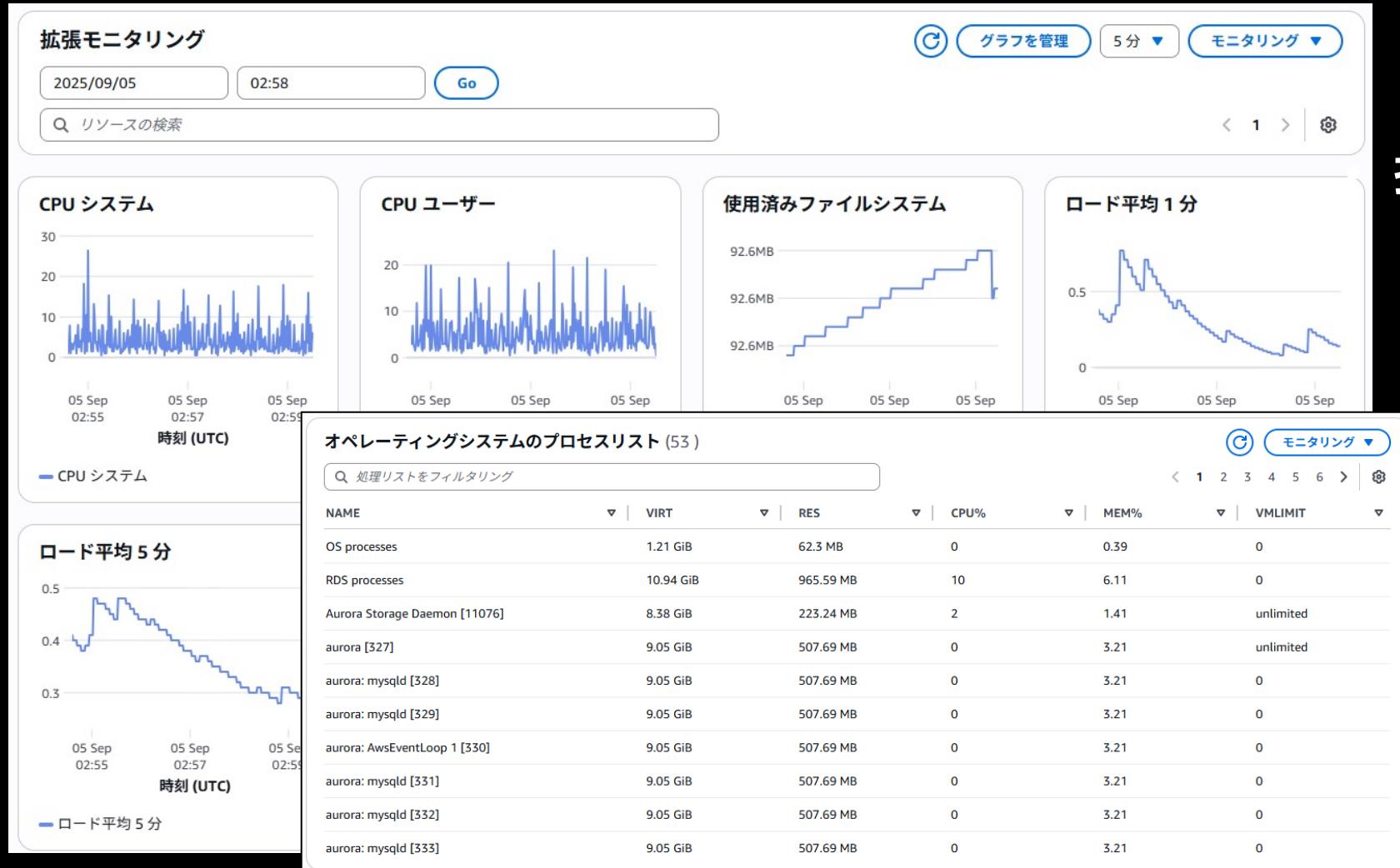
Below the editor are buttons for 'クエリの実行' (Run Query), 'キャンセル' (Cancel), '保存' (Save), and '履歴' (History). A note says 'Logs Insights QL クエリは、最大 60 分まで実行できます。' (Logs Insights QL queries can run up to 60 minutes). A message indicates 'Completed. Query executed for 1 log group.'.

The bottom section displays the results under 'Logs (647)'. It includes a histogram for 'Rows_examined' over time from 05:00 to 07:45. The results table has columns: '#', 'Time', 'query_time', 'Query', 'Rows_sent', and 'Rows_examined'. The first three rows of the table are:

#	Time	query_time	Query	Rows_sent	Rows_examined
1	2025-09-10T22:46:52.051677Z	2.725971	SELECT sql_no_cache max_temp,min_temp,station_name FROM weather WHERE max_...	1120	3197957 SET
2	2025-09-10T22:46:54.818286Z	2.736682	SELECT sql_no_cache count(id) FROM weather WHERE station_name = 'SANDBERG'...	1	3196837 SET
3	2025-09-10T22:46:37.584131Z	2.772714	SELECT sql_no_cache count(id) FROM weather WHERE station_name = 'SANDBERG'...	1	3196837 SET

- クエリを実行してスロークエリログを解析
例えば「Time」「Query_time」「Query」「Rows_sent」「Rows_examined」などの個々のフィールドを取得

Amazon RDS 拡張モニタリング



拡張モニタリング

- CloudWatch の標準メトリクスより詳細なOSのメトリクスを取得
- CPU使用率の内訳 (User/System/IO)
- ファイルシステムの使用率
- 最小で1秒ごとに各種メトリクスを取得
- プロセスごとのリソース使用量の取得など

Amazon RDS 拡張モニタリング

Db2、MariaDB、MySQL、Oracle、および PostgreSQL の主な OS メトリクス

グループ	メトリクス	コンソール名	説明
cpuUtilization	Nice	CPU Nice	最も低い優先順位で実行されているプログラムが使用中の CPU の使用率。
cpuUtilization	System	CPU システム	カーネルが使用中の CPU の使用率。
cpuUtilization	Total	CPU 合計	使用中の CPU の合計使用率。この値は nice 値を含みます。
cpuUtilization	User	CPU ユーザー	ユーザープログラムが使用中の CPU の使用率。
cpuUtilization	Wait	CPU 待機	I/O アクセスを待機中の CPU の未使用率。
swap	In	スワップイン	ディスクからスワップされたメモリの量 (キロバイト単位)。
swap	Out	スワップアウト	ディスクにスワップされたメモリの量 (キロバイト単位)。
processList	cpuUsedPc	CPU %	プロセスが使用中の CPU の使用率。
processList	memoryUsedPc	MEM%	プロセスが使用中のメモリの使用率。

参考

- 1) [Db2、MariaDB、MySQL、Oracle、および PostgreSQL の OS メトリクス](#)
- 2) [Microsoft SQL Server の OS メトリクス](#)

待機イベントの見方

データベースロード

データベースのパフォーマンスを評価するためのKPI

データベースロード → アクティブなセッション数(Active Sessions)

アクティブなセッションとは

- データベースのセッションは“アクティブ” / “アイドル”的なステータスがある
- さらに、アクティブなセッションは“CPU使用中”、“他の処理を待機中”的2つのステータス

データベースロードとは

- 1秒おきに“アクティブ”なセッションの詳細な情報をサンプリング
 - ステータス(CPU使用中、待機中(待機イベント^{(*)1})、SQL文、接続元ホスト、接続ユーザー)
- 単位時間^{(*)2}の“Average Active Sessions”とも言える

*1 各データベースの待機イベントの情報はAppendixを参照(参考となるリファレンスを掲載)

*2 データベースロードを選択する期間により平均を取得するための間隔は変動(最小で1秒間の平均値となる)

待機イベントについて(1)

待機イベントはセッションが待っているリソースを示します。

⇒ 例えば、待機イベント `io/socket/sql/client_connection` はスレッドが新しい接続を処理中であることを示します。

データベース全体の負荷状況の指標として利用できる**データベースロード(AAS)**をドリルダウンして待機イベントを分析することでパフォーマンスに影響を与える原因を知ることができます。

この待機イベントはデータベースエンジンごとに異なります。

- MariaDB および MySQL のすべての待機イベントの詳細については、MySQL ドキュメントの「[イベント待機サマリーテーブル](#)」を参照してください
- すべての PostgreSQL 待機イベントについては、PostgreSQL ドキュメントの「[PostgreSQL の待機イベント](#)」を参照してください
- すべての Oracle 待機イベントについては、Oracle ドキュメントの「[Descriptions of Wait Events](#)」を参照してください
- SQL Server のすべての待機イベントについては、SQL Server ドキュメントの「[待機の種類](#)」を参照してください

待機イベントについて (2)

Auroraの場合、MySQL と PostgreSQL に互換性があるため多くの待機イベントの意味を共有しています。しかし、一部 Aurora 独自の待機イベントもありますので、代表的な待機イベントは以下で確認できます。

- Aurora MySQL の代表的な待機イベントのリストについては、
[「Aurora MySQL の待機イベント」](#) を参照してください
- Aurora PostgreSQL の代表的な待機イベントのリストについては、
[「Amazon Aurora PostgreSQL のイベント」](#) を参照してください

Aurora MySQL の待機イベント

待機イベント名	説明
cpu	スレッドが CPU でアクティブな場合、または CPU を待っている際に発生します。
io/aurora_redo_log_flush	セッションは Aurora ストレージにデータを保持しています。通常、この待機イベントは Aurora MySQL の書き込み I/O オペレーション用です。詳細については、「 io/aurora_redo_log_flush 」を参照してください。
io/table/sql/handler	エンジンは、テーブルへのアクセスを待っています。このイベントは、データがバッファプールにキャッシュされているか、ディスク上でアクセスされているかにかかわらず、発生します。詳細については、「 io/table/sql/handler 」を参照してください。
synch/cond/innodb/row_lock_wait	複数のデータ操作言語 (DML) ステートメントが同じデータベース行に同時にアクセスしようとしています。詳細については、「 synch/cond/innodb/row_lock_wait 」を参照してください。
synch/mutex/innodb/buf_pool_mutex	スレッドがメモリ内のページにアクセスするために InnoDB バッファプールのロックを取得したときに発生します。
synch/mutex/innodb/fil_system_mutex	セッションがテーブルスペースのメモリキャッシュへのアクセスを待っているときに発生します。
synch/mutex/innodb/trx_sys_mutex	大量のトランザクションで高いデータベースアクティビティがある場合に発生します。
synch/sxlock/innodb/hash_table_locks	セッションは、バッファプール内のページを見つけることができませんでした。エンジンは、ファイルを読み取るか、バッファプールの最も長い時間使われていない (LRU) リストを変更する必要があります。バッファキャッシュのサイズを増やし、関連するクエリのアクセスパスを改善することを検討してください。

参照:

[待機イベントを使用した Aurora MySQL のチューニング](#)

Aurora PostgreSQL の待機イベント

待機イベント名	説明
CPU	スレッドが CPU でアクティブになっているか、CPU の待機中に発生します。
Client:ClientRead	Aurora PostgreSQL がクライアントからのデータ受信を待っているときに発生します。
Client:ClientWrite	Aurora PostgreSQL がクライアントへのデータ書き込みを待っているときに発生します。
IO:DataFileRead	バックエンドプロセスが必要なページをストレージから読み込む際に、ページが共有メモリで使用できないために接続が待機したときに発生します。
IO:XactSync	データベースが、Aurora ストレージサブシステムが通常のトランザクションのコミットを承認するか、準備されたトランザクションのコミットまたはロールバックの承認を待っているときに発生します。
Lock:advisory	PostgreSQL アプリケーションがロックを使用して、複数のセッションにわたるアクティビティを調整するときに発生します。
Lock:extend	バックエンドプロセスがリレーション拡張のためにロックするのを待機中に、他のプロセスが同じ目的でそのリレーションをロックしているときに発生します。
Lock:Relation	他のトランザクションによって現在ロックされているテーブルまたはビューに対するロックを取得するためにクエリが待っているときに発生します。
Lock:transactionid	トランザクションが行レベルロックを待っているときに発生します。
Lock:tuple	バックエンドプロセスがタプルのロック取得を待機中の場合に発生します。
LWLock:buffer_content	セッションがデータページのメモリ内への読み取りまたは書き込みのために待機中、そのデータページが他のセッションで書き込むためにロックされているときに発生します。

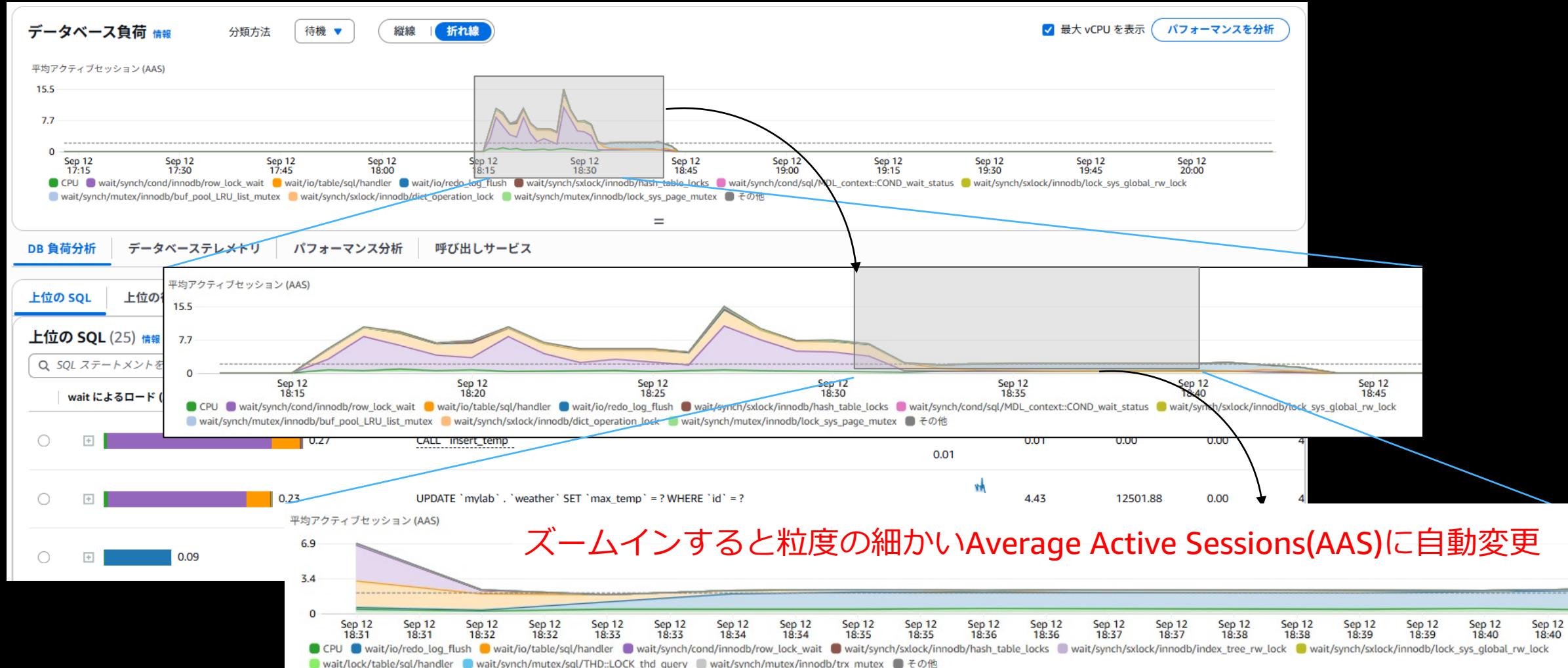
参照:

[Aurora PostgreSQL の待機イベント](#)

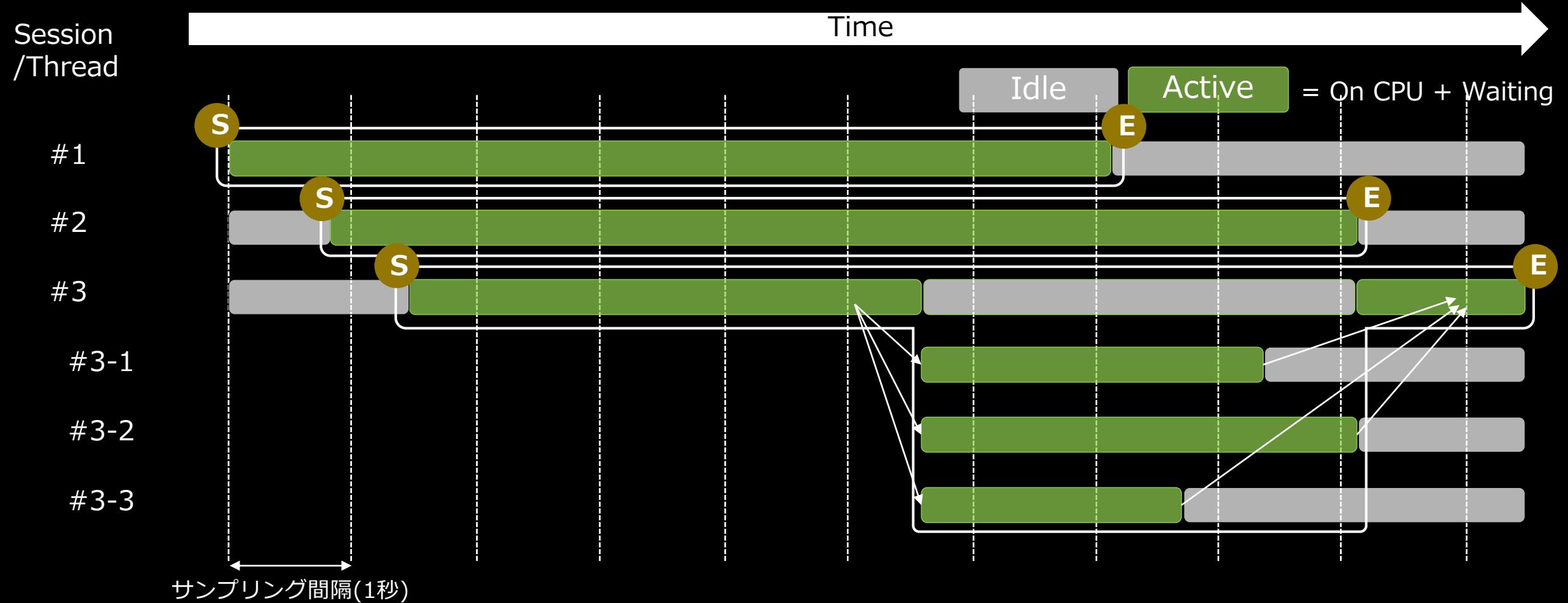
[Aurora PostgreSQL の待機イベントでのチューニング](#)

Database Insights ダッシュボードでの待機イベント確認

対象時間帯を選択して頂く事で、選択した時間帯に発生していた待機イベント詳細を確認可能

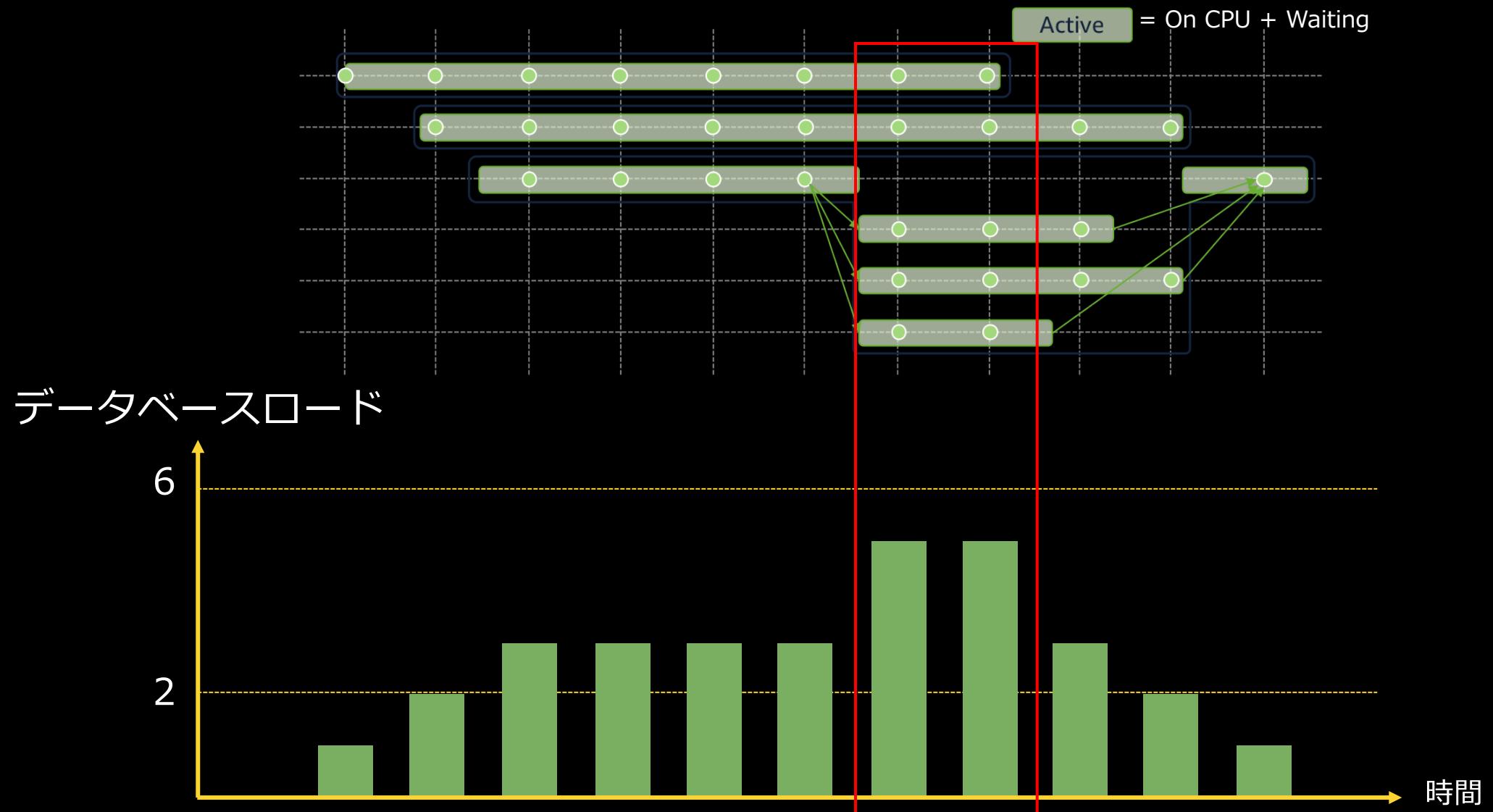


データベースロード

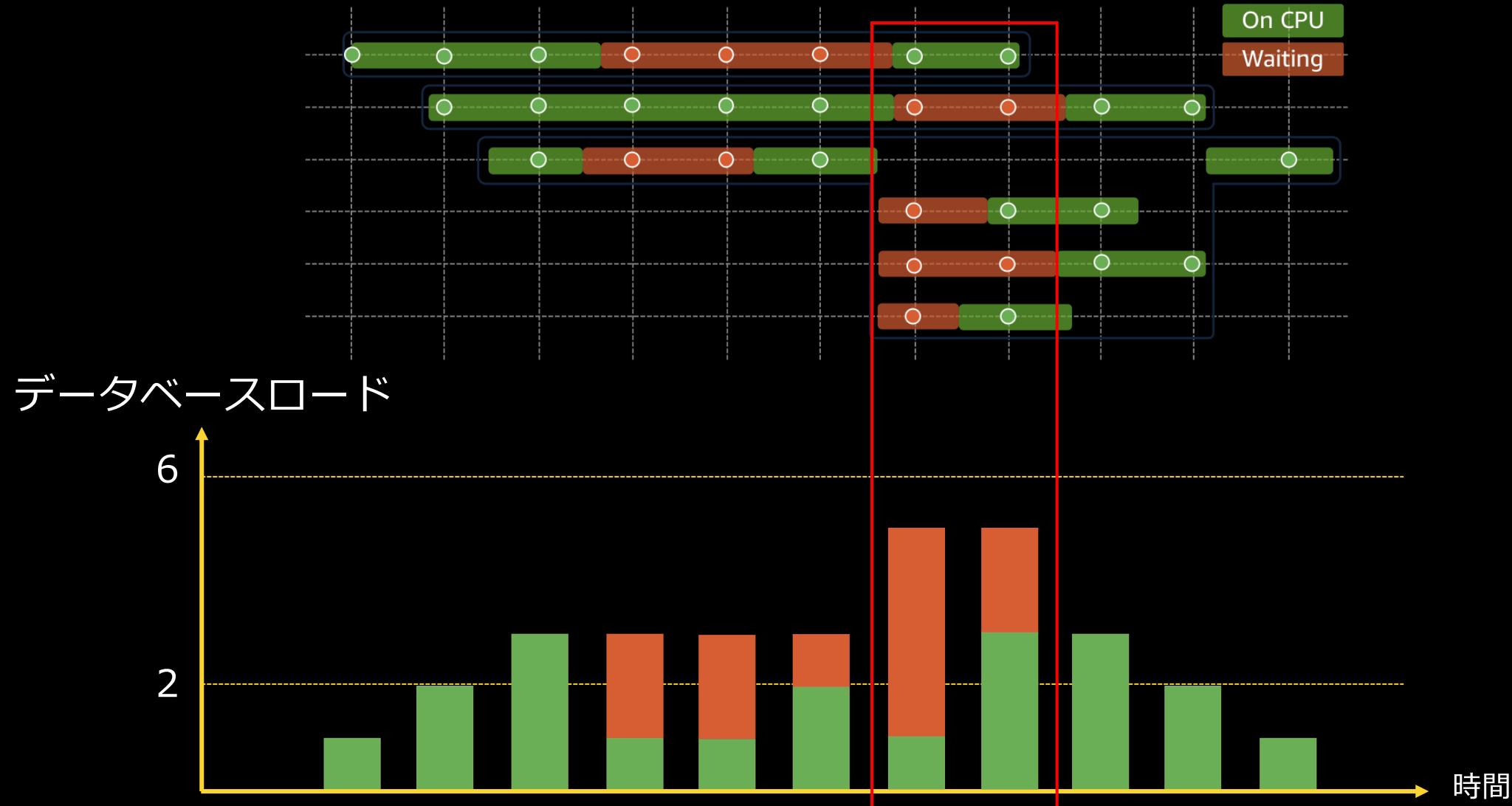


Database Insights は1秒のサンプリング間隔で**セッションの状態**を取得

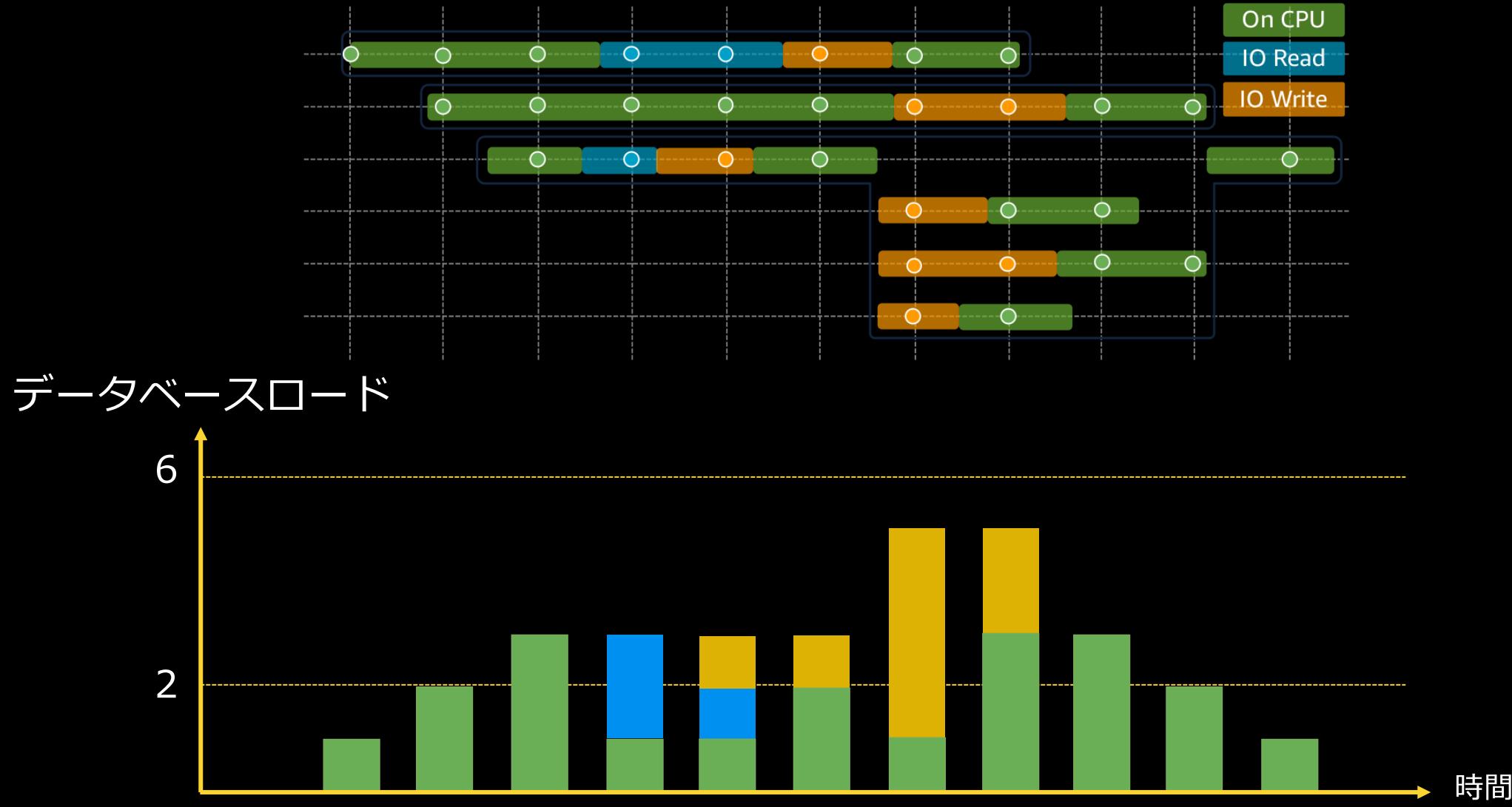
データベースロード = Average Active Sessions



アクティブセッションの状態を分析



さらに”待機イベント”で分析



データベースロードで問題の発見

データベースロード(AAS) ≈ 0

データベースロード(AAS) >= # of vCPUs

データベースロード(AAS) >> # of vCPUs

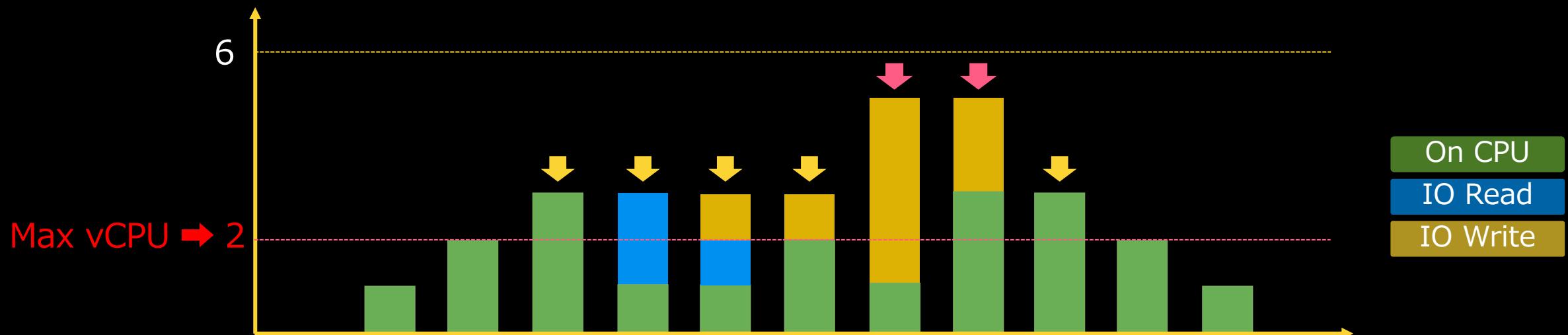
基本的にデータベースが使用されていない

パフォーマンス問題の可能性がある

パフォーマンス問題

※AAS = Average Active Sessions

→ パフォーマンス問題の有無は、データベースロードがインスタンスの最大vCPUを超えていているかどうかが基準



まとめ

- ・ チューニングのための検知・分析として活用できるツール
 - CloudWatch
 - 拡張モニタリング
 - CloudWatch Database Insights
- ・ **CloudWatch Database Insights** では
データベース全体の健全性を管理する包括的な機能を提供
 - メトリクス、ログ、スローSQLクエリ、イベント、OSプロセスを単一のビューに統合
 - アドバンストモードではオンデマンドパフォーマンス分析や実行プランの分析※1
など高度な分析も可能
- ・ データベースロード(AAS)を元に待機イベントを分析し問題を把握する

※1 サポートされているDBエンジンで利用可能 参考：[Database Insights のモード](#)

Thank you!

