Amazon RDS for MySQL と MariaDB のモニタリングおよびアラートツールとベストプラクティス

AWS 規範ガイダンス

AWS 規範ガイダンス: Amazon RDS for MySQL と MariaDB のモニタリングおよびアラートツールとベストプラクティス

Table of Contents

Amazon RDS for MySQL と MariaDB のモニタリングおよびアラートツールとベストプラクティス

Igor Obradovic、Amazon Web Services (AWS)

2025 年 3 月 (ドキュメント履歴)

データベースモニタリングとは、データベースの可用性、パフォーマンス、機能を測定、追跡、評価するプロセスを意味します。モニタリングおよびアラートソリューションを導入すると、データベースサービス、つまり関連するアプリケーションおよびワークロードのセキュリティ、高性能、レジリエンス、効率性を確保しやすくなります。では AWS、ワークロードの正常性を理解し、時間の経過とともにオペレーションからインサイトを得るために、ワークロードログ、メトリクス、イベント、トレースを収集して分析できます。

リソースをモニタリングすると、リソースが想定どおりに動作していることを確認でき、顧客に影響が及ぶ前に、問題を検出し修正することも可能です。しきい値を超える状態が生じた際にアラームを発行するには、そうしたメトリクス、ログ、イベント、トレースをモニタリングする必要があります。

このガイドでは、Amazon Relational Database Service (Amazon RDS) データベースのオブザーバビリティ、モニタリングツール、ベストプラクティスについて説明します。MySQL および MariaDB データベースに焦点を当てていますが、ほとんどの情報は、他の Amazon RDS データベースエンジンにも適用できるものです。

このガイドは、次のような皆様を対象にしています: ソリューションアーキテクト、データベースアーキテクト、DBA、シニア DevOps エンジニアのほか、 AWS クラウドで稼働するデータベースワークロードのモニタリングおよびオブザーバビリティソリューションの設計、実装、管理に携わるチームメンバー

目次

概要

モニタリングとアラートは、AWS Well-Architected フレームワークの 4 つの柱に組み込まれています。

- 運用上の優秀性の柱では、ワークロードではテレメトリとモニタリングを重視する必要があると規定されています。Amazon Relational Database Service (Amazon RDS) などの AWS サービスでは、ワークロード内部の状態 (メトリクス、ログ、イベント、トレースなど) を把握するのに必要な情報が提供されています。Amazon RDS データベースの運用では、データベースインスタンスの状態を把握するとともに、運用イベントを検出して、計画済みイベントにも計画外イベントにも対応できなければなりません。AWS に用意されているモニタリングツールを利用すると、組織およびビジネス上の成果実現が危うくなった時点や、そうなり得るタイミングを判断しやすくなります。これによって、適切なアクションを適時に実行できます。
- パフォーマンス効率の柱では、Amazon RDS DB インスタンスなどのリソースパフォーマンスのモニタリングが必要と規定されており、これを行うには、パフォーマンス関連のメトリクスをリアルタイムで収集、集約、処理しなければなりません。そうしたモニタリングにより、最適化されていない SQL クエリや不適切な設定パラメータといったパフォーマンス低下を特定して、要因を修正でき、測定値が想定範囲外になった場合に、アラームを自動生成することも可能です。通知のためだけでなく、検出したイベントに応じて自動アクションを開始するためにも、アラームを使用すると良いでしょう。モニタリングを行うと、収集したメトリクスを事前定義したしきい値と比較して評価したり、機械学習アルゴリズムを使用して異常な動作を特定したりすることも可能です。例えば、CPU 使用率の上昇傾向を検出するために、`cpuUtilization.total` メトリクスを一定期間、収集し分析できます。また、CPU 使用率がハードリミットに達する前に、その異常をプロアクティブに警告できるようにすれば、影響が顧客に及ぶ前に問題を修正できます。
- 信頼性の柱では、モニタリングとアラートは可用性の要件を満たすために重要な要素と定義されています。モニタリングソリューションでは、障害を効果的に検出できる必要があり、問題や障害を検出する主な目的は、それらについてアラートを得られるようにすることです。クラウドで回復力のあるアーキテクチャを実現するには、オブザーバビリティおよびモニタリングの継続的なプラクティスを実装することが不可欠です。また、ワークロードを改善するには、それらを測定し、状態やヘルスを把握しなければなりません。障害からの自動復旧、水平方向のスケーラビリティ、キャパシティプロビジョニングの設計原則は、正確なモニタリングおよびアラートサービスの上に成り立つものです。
- セキュリティの柱では、予期しないあるいは不要な設定変更や、予期しない動作の検出と防止が論じられています。これを実現するには、MariaDB 監査プラグインを使用して Amazon RDS for MySQL および MariaDB の DB インスタンスを設定することで、ユーザーログインや、データベースへの特定のオペレーション実行といったデータベースアクティビティを記録すると良いでしょ

う。データベースアクティビティの記録は、プラグインによってログファイルに保存でき、それらのログファイルをモニタリングツールとアラートツールに統合またはインポートすることも可能です。ログファイルは、データベース内の予期しない動作や疑わしい動作を対象に、リアルタイムで分析されます。こうした予期しない動作や疑わしい動作は、Amazon RDS DB インスタンスが侵害された可能性、つまり、起こり得るビジネスリスクの兆候を示しています。モニタリングツールでそうしたイベントを検出したら、セキュリティインシデントへの対応を開始するアラームをアクティブ化します。これにより、不審なアクティビティや悪意のあるアクティビティに対処しやすくなります。

目標とするビジネス成果

モニタリングとアラートのメカニズムにベストプラクティスを実装すると、パフォーマンス、回復力、効率、安全性に優れ、コストを最適化した、アプリケーションおよびワークロード向けインフラストラクチャを実現しやすくなります。オブザーバビリティツールを使用して、メトリクス、イベント、トレース、ログをリアルタイムで収集、保存、可視化することで、データベースのヘルスとパフォーマンスを俯瞰的に監視し分析できるため、関連 IT サービスの低下や中断を防止できます。計画外の機能低下やサービス中断が続く場合、モニタリングとアラートのツールを使用すると、タイムリーな問題検出、エスカレーション、対応、迅速な調査および解決が容易になります。クラウドデータベースワークロードの包括的なモニタリングおよびアラートソリューションは、次のようなビジネス成果実現に有用です。

- カスタマーエクスペリエンスを向上させる: 信頼性の高いサービスにより、カスタマーエクスペリエンスが向上します。多くの場合、データベースは、ウェブおよびモバイルアプリケーション、メディアストリーミング、決済サービス、business-to-business (B2B) API、統合サービスといったデジタルサービスの主要なコンポーネントとして機能しています。そのため、データベースをモニタリングしアラートを設定することで、問題を迅速に検出して効率的に調査し、可能な限り迅速な修正によってダウンタイムなどの中断を最小限に抑えられれば、顧客向けデジタルサービスの可用性、セキュリティ、パフォーマンスが向上します。
- 顧客の信頼を高める: パフォーマンスが向上し、ユーザーエクスペリエンスが円滑化されれば、顧客の信頼を得られるため、自社プラットフォームでのビジネスが拡大する可能性があります。例えば、信頼性の高いオンライン決済サービスを提供しているプロバイダーを例に取ると、顧客の信頼と口イヤルティが高まることを期待できます。これが、顧客数の増加、保持率の向上、請求可能な取引の増加、革新的なサービス提供につながり、収益も拡大するでしょう。
- 財務上の損失を回避する: データベースインフラストラクチャで予期しないダウンタイムが発生すると、その影響は、貴社のアプリケーションで顧客が実行しているビジネストランザクションに及ぶ可能性があります。場合によっては、これにより多額の経済的損失が発生します。サービスレベ

ルアグリーメント (SLA) に違反すれば、顧客の信頼を損ね、ひいては、収益を失う可能性さえあります。そうした違反は、高額な訴訟の法的根拠にもなりかねません。顧客が貴社の責任と保証契約に基づいて賠償を求める可能性があるからです。ソフトウェア会社、Atlassian Corporation の調査によると、サービスが停止した場合の平均コストは、ビジネスの種類や規模にもよりますが、1 時間あたり 140,000 USD から 540,000 USD にも上ります。長時間のサービス停止やビジネスの損失を防ぐには、安定したデータベース環境が重要なのです。

- ビジネス価値を高める: モニタリングとアラートの仕組みは、可用性、回復力、信頼性、パフォーマンス、コスト効率、安全性に優れたデジタルサービスを設計、開発、運用するのに有用ですが、これらの導入は出発点にすぎません。その後、スケーリングと拡張を徐々に行い、既存のクラウドワークロードを強化し、新しいサービスを導入する必要があります。新しいサービスは、顧客価値の向上や、自社の収益拡大につながり、ビジネス成長にフライホイール効果ももたらします。
- 開発者の生産性を向上させる: 開発者が、生産的かつ効率的でいられ、開発タスクで問題やボトルネックに遭遇しなければ、高品質の製品を迅速に提供できます。しかし、ソフトウェアエンジニアリングと IT 運用では複雑な課題が生じやすく、ワークロードやそのアーキテクチャの規模によっては、そうした複雑さが増大します。分散アプリケーション全体のパフォーマンスと一貫性を分析するには、相関メトリクスとトレースを得られる開発ツールが必要です。こうした情報があれば、欠陥のあるコードアーティファクトやインフラストラクチャコンポーネントを可能な限り迅速に特定でき、エンドユーザーへの影響も判断しやすくなります。適切なモニタリングおよびアラートのツールスイートを導入すると、コード作成とテストを迅速かつ適切に行えるでしょう。
- 運用の有効性と効率を高める: クラウドワークロードの大規模運用では、パフォーマンスがわずかに向上しただけでも、数百万ドルのコスト削減につながる可能性があります。データベースをモニタリングし、メトリクス、イベント、ログ、トレースを分析すると、将来のキャパシティニーズを把握して予測するとともに、AWS クラウド で節約したコストを活用できます。Amazon RDS ワークロードと運用上のヘルスを把握していれば、イベントへの対応、問題の修正、改善の計画が容易になります。

一般的なベストプラクティス

以下のベストプラクティスは、Amazon RDS ワークロードのヘルスを十分に把握し、必要に応じて運用イベントやモニタリングデータに適切なアクションを実行するのに役立ちます。

- Identify KPI を特定する。求めるビジネス成果を基に、主要業績評価指標 (KPI) を特定し、KPI を評価して、ワークロードの成果を判断します。例えば、コアビジネスが e コマースの場合、求めるビジネス成果の 1 つとして、顧客が e ショップを 24 時間 365 日利用できることが考えられるでしょう。このビジネス成果を得るには、e-shop アプリケーションで使用されるバックエンド Amazon RDS データベースの可用性 KPI を定義して、ベースライン KPI を週あたり 99.99% に設定します。ベースライン値と照らし合わせて実際の可用性 KPI を評価すると、目標のデータベース可用性 99.99% を満たしているかどうかを判定して、24 時間 365 日サービスを提供するというビジネス成果を得られるでしょう。
- ワークロードメトリクスを定義する。ワークロードメトリクスを定義して、Amazon RDS ワークロードの量と品質を測定します。メトリクスを評価して、ワークロードによって求める成果を得られているかどうかを判断し、ワークロードのヘルスを把握します。例えば、Amazon RDS DB インスタンスの可用性 KPI を評価するには、DB インスタンスのアップタイムやダウンタイムといったメトリクスを測定しなければなりません。その後、こうしたメトリクスを使用して、次のように可用性 KPI を計算できます。

  ```
  availability = uptime / (uptime + downtime)
  ```

  メトリクスとは、時系列のデータポイントセットであり、これには、分類と分析に役立つディメンションを含めることもできます。
- ワークロードメトリクスを収集して分析する。Amazon RDS では、設定に応じて、さまざまなメトリクスとログが生成されます。これらの中には、DB インスタンスイベント、カウンター、統計情報などを表すものもあります (例: `db.Cache.innoDB_buffer_pool_hits`)。その他のメトリクスは、オペレーティングシステムから取得します。`memory.Total` がその例であり、これによって、ホスト Amazon Elastic Compute Cloud (Amazon EC2) インスタンスのメモリ総量を測定できます。ここでは、モニタリングツールを使い、収集済みメトリクスを定期的かつプロアクティブに分析して傾向を特定し、適切な対応が必要かどうかを判断しなければなりません。
- ワークロードメトリクスのベースラインを確立する。メトリクスのベースラインを確立して、期待値を定義するとともに、正常か異常かを判断するしきい値を特定します。例えば、通常のデータベースオペレーションの場合、`ReadIOPS` のベースラインは、最大 1,000 と定義できるでしょう。これにより、定義したベースラインを比較対象にし、過剰使用率を特定できます。新規メトリ

クスによって、読み取り IOPS が一貫して 2,000 ~ 3,000 の範囲にあることがわかれば、調査、介入、改善のための対応をトリガーしてもよい偏差を特定したことになります。

- ワークロードの成果を得られない可能性がある場合にアラートを発行する。ビジネス成果実現が危ういと判断された場合に、アラートを発行します。これによって、顧客に影響が及ぶ前にプロアクティブに問題に対処したり、インシデントの影響をタイムリーに軽減したりできます。
- ワークロードアクティビティの想定パターンを特定する。メトリクスベースライン基づいて、ワークロードアクティビティのパターンを確立することで、予期しない動作を特定し、必要に応じて適切なアクションを実行します。AWS のモニタリングツールを使用すると、統計アルゴリズムと機械学習アルゴリズムを適用してメトリクスを分析し、異常を検出できます。
- ワークロードの異常が検出された場合にアラートを発行する。Amazon RDS ワークロードのオペレーションで異常が検出された場合にアラートを発行し、必要に応じて、適切なアクションを実行できるようにします。
- KPI とメトリクスを確認し修正する。Amazon RDS データベースが定義済みの要件を満たしていることを確認するとともに、ビジネス目標の達成に寄与し得る改善点を特定します。また、測定したメトリクスと評価 KPI の有効性を検証し、必要に応じて修正します。例えば、最適な同時データベース接続数に KPI を設定した後に、接続の試行と失敗に関するメトリクスと、作成され実行中のユーザースレッドをモニタリングするとしましょう。場合によっては、KPI ベースラインで定義した数よりも多くのデータベース接続があるでしょう。そうした最終結果は、現在のメトリクス分析によって検出できますが、根本原因は特定できない可能性があります。その場合は、メトリクスを修正して、その他のモニタリング指標 (テーブルロックのカウンターなど) を追加しなければなりません。こうした新規メトリクスを使用すると、データベースの接続数増加の原因が予期しないテーブルロックにあるかどうかを判断しやすくなります。

モニタリングツール

オブザーバビリティ、モニタリング、アラートの各ツールを使用して、以下を行うことをお勧めします。

- Amazon RDS 環境のパフォーマンスに関するインサイトを取得する
- 予期しない動作や疑わしい動作を検出する
- キャパシティを計画し、Amazon RDS インスタンスの割り当てについて、知識に基づく意思決定を行う
- メトリクスとログを分析し、起こり得る問題をプロアクティブに予測する
- しきい値を超える状態が生じた際にアラートを発行することで、ユーザーに影響が及ぶ前にトラブルシューティングを行い、問題を解決する

こうしたツールは、さまざまなオプションとソリューションから選択できます。例として、AWS に用意されているクラウドネイティブのオブザーバビリティおよびモニタリングのツールとサービス、無料のオープンソースソフトウェアソリューション、Amazon RDS DB インスタンスをモニタリングする商用サードパーティー製ソリューションなどが挙げられます。これらのツールの一部については、以降のセクションで説明します。

ニーズに最適なツールを判断するには、それぞれのツールが備える各種機能や総合的な性能を組織の要件と比較します。また、次の点についても、ツールを評価すると良いでしょう: デプロイの容易さ、設定と統合、ソフトウェアの更新とメンテナンス、デプロイの方法 (ハードウェアやサーバーレスなど)、ライセンス、料金、組織に固有のその他の要因

セクション

Amazon RDS が備えるツール

Amazon Relational Database Service (Amazon RDS) は、AWS クラウド で提供されるマネージド型データベースサービスです。Amazon RDS はマネージドサービスであるため、これを利用すると、データベースバックアップ、オペレーティングシステム (OS) とデータベースソフトウェアのインストール、OS とソフトウェアのパッチ適用、高可用性のセットアップ、ハードウェアライフサイクル管理、データセンターオペレーションといった、ほとんどの管理タスクから解放されます。AWS には、包括的なツールセットも用意されており、これによって、Amazon RDS DB インスタンスの包括的なオブザーバビリティソリューションを構築できます。

一部のモニタリングツールは、Amazon RDS サービスで提供されます。事前に設定されており、自動的に有効になります。Amazon RDS の新規インスタンスを起動すると、次の 2 つの自動ツールをすぐに使用できます。

- Amazon RDS インスタンスのステータス: これにより、DB インスタンスの現在のヘルスを詳しく確認できます。例えば、ステータスコードには、使用可能、停止、作成中、バックアップ中、失敗があります。インスタンスのステータスは、Amazon RDS コンソール、AWS Command Line Interface (AWS CLI)、または Amazon RDS API を使用して表示できます。詳細については、Amazon RDS ドキュメントの「Amazon RDS DB インスタンスのステータスの表示」を参照してください。
- Amazon RDS の推奨事項: DB インスタンス、リードレプリカ、DB パラメータグループ向けの推奨事項が自動的に提示されます。これらは、DB インスタンスの使用状況、パフォーマンスデータ、設定を分析すると提示されるもので、ガイダンスとして利用できます。例えば、[エンジンバージョンが古くなっています] という推奨事項が表示された場合、これは、DB インスタンスで最新バージョンのデータベースソフトウェアが稼働していないことや、最新のセキュリティ修正プログラムやその他の改善点を活用するには DB インスタンスのアップグレードが必要であることを示唆しています。詳細については、Amazon RDS ドキュメントの「Amazon RDS の推奨事項の表示」を参照してください。

CloudWatch 名前空間

Amazon RDS は、Amazon CloudWatch と統合されており、CloudWatch は、AWS で稼働するクラウドリソースとアプリケーションのモニタリングおよびアラートサービスとして提供されるものです。Amazon RDS では、DB インスタンスのオペレーション、使用率、パフォーマンス、ヘルスに関するメトリクス、ログファイル、トレース、イベントが自動収集され、長期保存、分析、アラートを目的として CloudWatch に送信されます。

Amazon RDS for MySQL と Amazon RDS for MariaDB では、デフォルトのメトリクスセットが 1 分間隔で CloudWatch に自動発行されます。これには、追加料金はかかりません。こうしたメトリクスは、メトリクスのコンテナである次の 2 つの名前空間に収集されます。

- AWS/RDS 名前空間: DB インスタンスレベルのメトリクスが含まれており、その例には、`BinLogDiskUsage` (バイナリログが占有するディスク容量)、`CPUUtilization` (CPU 使用率)、`DatabaseConnections` (DB インスタンスへのクライアントネットワーク接続の数) などがあります。
- AWS/Usage 名前空間: アカウントレベルの使用状況メトリクスが含まれており、これらを使用して、Amazon RDS のサービスクォータ内で運用しているかどうかを判断します。メトリクスの例には、`DBInstances` (AWS アカウントまたはリージョンの DB インスタンスの数)、`DBSubnetGroups` (AWS アカウントまたはリージョンの DB サブネットグループの数)、`ManualSnapshots` (AWS アカウントまたはリージョンで手動作成されたデータベーススナップショットの数) などがあります。

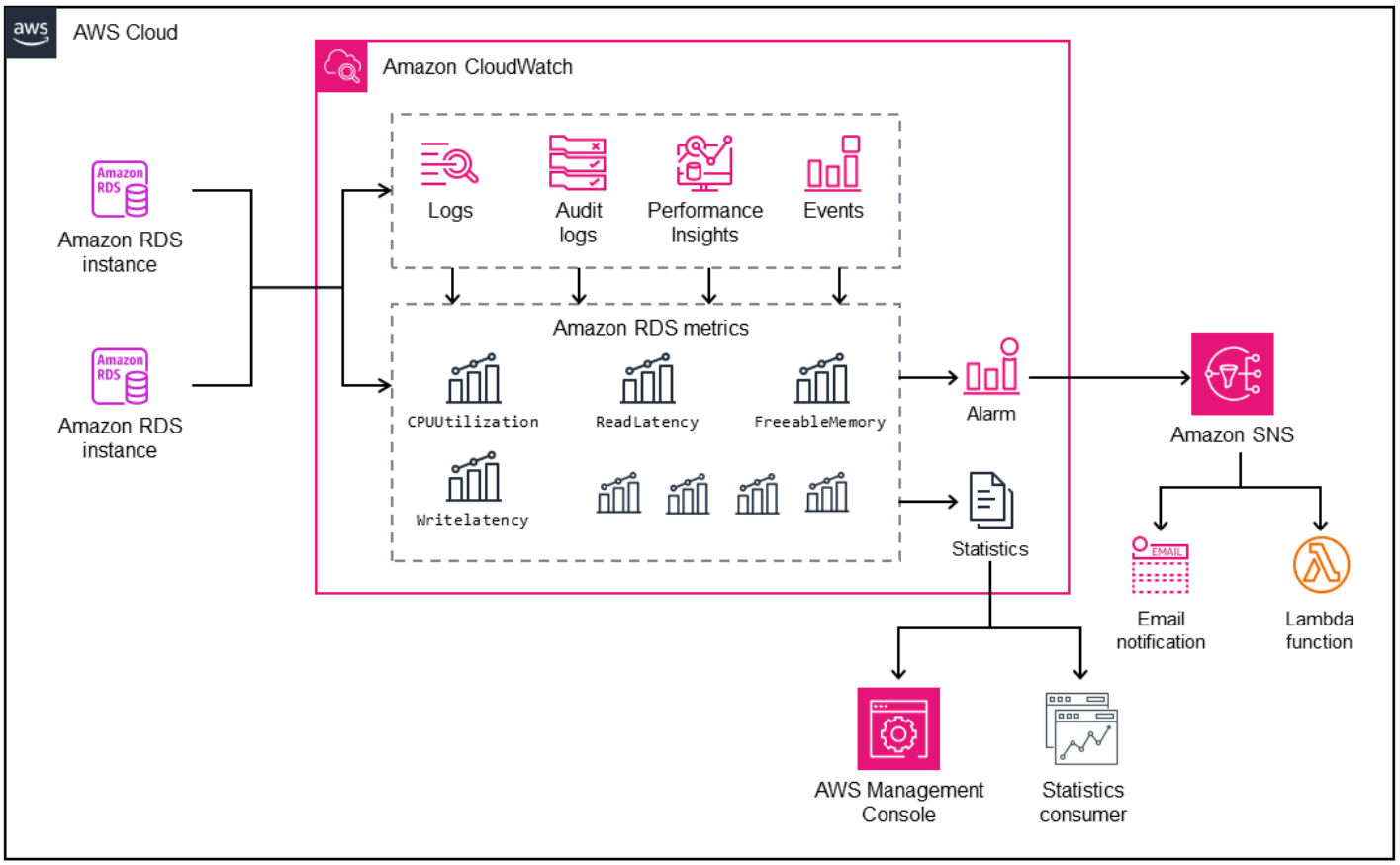
CloudWatch では、次のようにメトリクスデータを保持します。

- 3 時間: 60 秒未満間隔の高解像度カスタムメトリクスは 3 時間保持されます。3 時間後、データポイントは、1 分間のメトリクスに集約され、15 日間保持されます。
- 15 日間: 60 秒 (1 分) 間隔のデータポイントは 15 日間保持されます。15 日後、データポイントは、5 分間のメトリクスに集約され、63 日間保持されます。
- 63 日間: 300 秒 (5 分) 間隔のデータポイントは 63 日間保持されます。63 日後、データポイントは、1 時間のメトリクスに集約され、15 か月間保持されます。
- 15 か月間: 3,600 秒 (1 時間) 間隔のデータポイントは 15 か月間 (455 日間) 利用可能です。

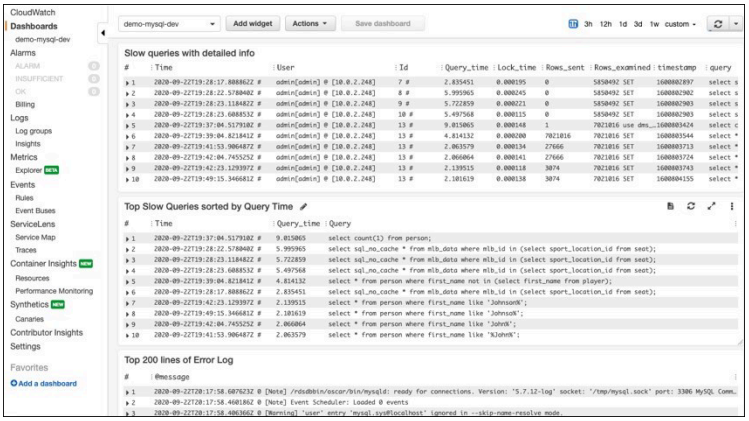
詳細については、CloudWatch ドキュメントの「メトリクス」を参照してください。

CloudWatch のアラームおよびダッシュボード

Amazon CloudWatch アラームを使用すると、特定の Amazon RDS メトリクスを一定期間モニタリングできます。例えば、`FreeStorageSpace` をモニタリングし、メトリクス値が設定したしきい値を超えた場合に 1 つ以上のアクションを実行できます。しきい値を 250 MB に設定済みで、空きストレージ容量が 200 MB (しきい値未満) の場合、アラームをアクティブ化して、Amazon RDS DB インスタンスに追加のストレージを自動プロビジョニングするアクションをトリガーさせることができます。アラームで、Amazon Simple Notification Service (Amazon SNS) を使用すると、DBA に通知用の SMS を送信することも可能です。次の図は、このプロセスを示したものです。

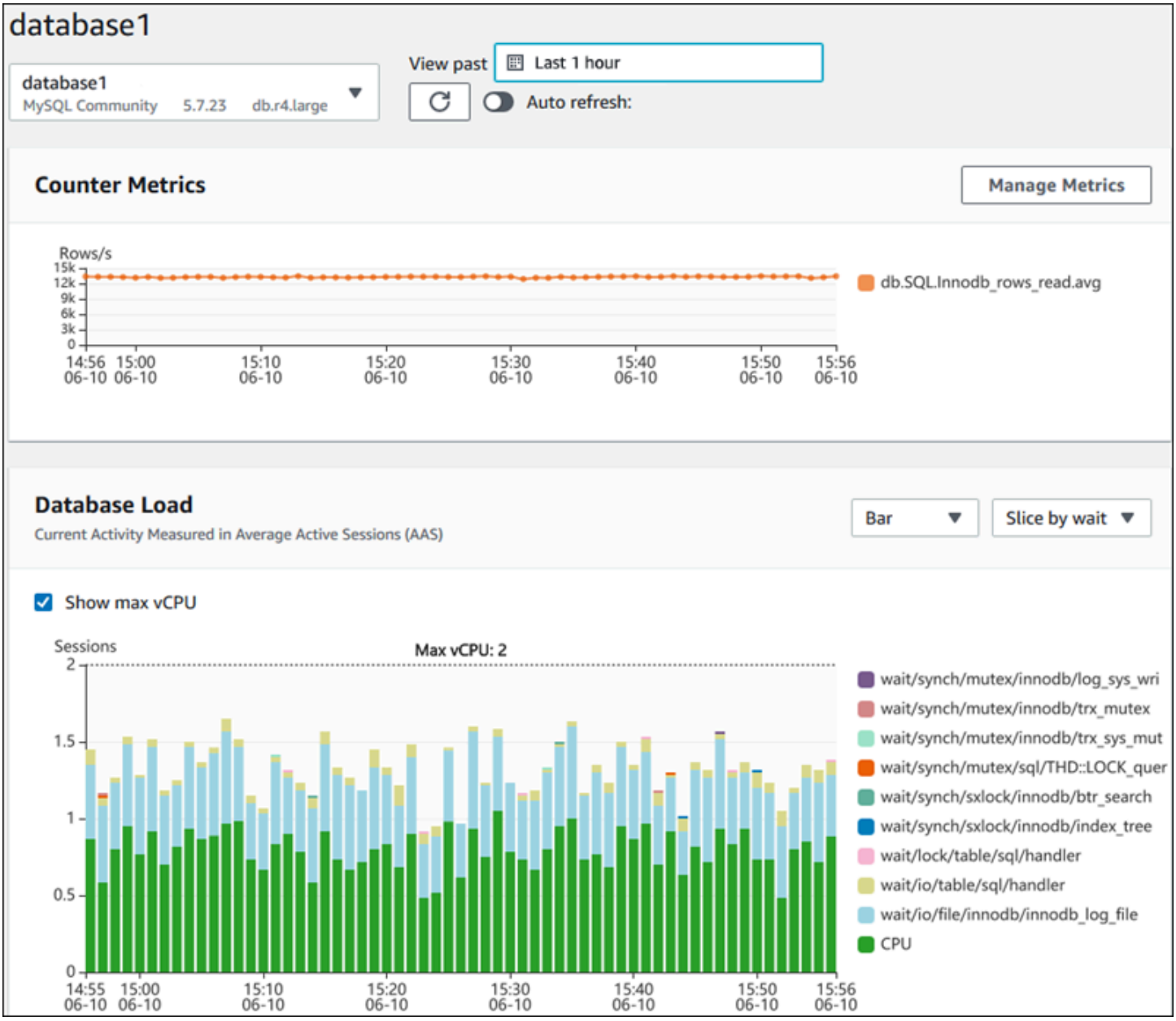


CloudWatch には、ダッシュボードもあり、これによって、メトリクスのビュー (グラフ) を作成、カスタマイズ、操作したり、カスタマイズしたビューを保存したりできます。また、CloudWatch Logs Insights を使用すると、スロークエリログとエラーログをモニタリングするダッシュボードを作成し、それらのログで特定のパターンが検出された場合にアラートを受信することも可能です。次の画像は、CloudWatch ダッシュボードの例を示しています。



Amazon RDS Performance Insights

Amazon RDS Performance Insights は、データベースのパフォーマンスチューニングおよびモニタリングツールであり、これによって、Amazon RDS のモニタリング機能を拡張できます。このツールは、データベースのパフォーマンス分析に有用です。具体的には、DB インスタンスの負荷を視覚化したり、待機、SQL ステートメント、ホスト、またはユーザーで負荷をフィルタリングしたりできます。また、複数のメトリクスが 1 つのインタラクティブグラフにまとめられるため、ロック待機、高い CPU 使用率、I/O レイテンシーなど、DB インスタンスに生じかねないボトルネックのタイプを特定し、その原因となっている SQL ステートメントを判断することも可能です。次の画像は、可視化の例を示しています。



アカウント内で稼働する Amazon RDS DB インスタンスのメトリクスを収集するには、DB インスタンスの作成プロセス中に Performance Insights を有効にする必要があります。無料利用枠には、7 日間のパフォーマンスデータ履歴と 1 か月あたり 100 万件の API リクエストが含まれていますが、必要に応じて、これより長いデータ保持期間を購入することもできます。料金情報の詳細については、「Performance Insights の料金」を参照してください。

Performance Insights を使用して DB インスタンスをモニタリングする方法については、このガイドの後半にある DB インスタンスのモニタリングセクションを参照してください。

Performance Insights により、メトリクスが CloudWatch に自動的に公開されます。さらに、Performance Insights ツールだけでなく、CloudWatch に用意されている追加機能も活用できます。Performance Insights のメトリクスの確認に、CloudWatch コンソール、AWS CLI、または CloudWatch API を使用できるのです。他のメトリクスと同様に、CloudWatch アラームを追加することも可能で、例えば、DBA への SMS 通知をトリガーしたり、DBLoad メトリクスが設定済みしきい値を超えた場合に修正アクションを実行したりできます。Performance Insights メトリクスは、既存の CloudWatch ダッシュボードに追加することもできます。

拡張モニタリング

拡張モニタリングは、Amazon RDS DB インスタンスが稼働するオペレーティングシステム (OS) のメトリクスをリアルタイムで取得するツールです。こうしたメトリクスにより、CPU、メモリ、Amazon RDS および OS プロセス、ファイルシステム、ディスク I/O データなどを対象に、最小で 1 秒の粒度を実現できます。こうしたメトリクスには、Amazon RDS console からアクセスして、それらを分析できます。Performance Insights と同様に、拡張モニタリングメトリクスが Amazon RDS から CloudWatch に配信されるため、これによって、分析用メトリクスの長期保存、メトリクスフィルターの作成、ダッシュボードでのグラフ表示、アラーム設定といった追加機能を CloudWatch で利用できます。デフォルトの場合、Amazon RDS DB インスタンスを新規作成すると、拡張モニタリングは無効になりますが、DB インスタンスの作成または変更の際に、有効にすることができます。利用料は、Amazon RDS から CloudWatch Logs に転送されたデータの量とストレージレートに基づいて課金されます。拡張モニタリングが有効になっている DB インスタンスの粒度および数に応じて、モニタリングデータの一部を CloudWatch Logs の無料利用枠に含めることができます。料金の詳細については、「Amazon CloudWatch 料金表」を参照してください。ツールの詳細については、Amazon RDS ドキュメントと拡張モニタリングに関するよくある質問を参照してください。

追加の AWS サービス

AWS では、Amazon RDS および CloudWatch との連携も可能なサポートサービスを複数提供しており、これらを使用すると、データベースのオブザーバビリティをさらに強化できます。例えば、Amazon EventBridge、Amazon CloudWatch Logs、AWS CloudTrail などのサポートサービスを利用できます。

- Amazon EventBridge は、サーバーレスイベントバスであり、これによって、アプリケーションと AWS リソース (Amazon RDS DB インスタンスなど) からイベントを受信し、それらをフィルタリング、変換、ルーティング、配信することができます。Amazon RDS イベントは、Amazon RDS 環境で生じた変更を示すものです。例えば、DB インスタンスのステータスが使用可能から停止に変わると、Amazon RDS では `RDS-EVENT-0087 / The DB instance has been stopped` というイベントが生成されます。Amazon RDS から、CloudWatch Events および EventBridge へは、イベントがほぼリアルタイムで配信されます。EventBridge と CloudWatch Events を使用すると、対象となる特定の Amazon RDS イベントにアラートを送信するルールを定義して、イベントがルールに一致した際に実行するアクションを自動化できます。イベントに対応するために、さまざまなターゲットが用意されています。例えば、修正アクションを実行する AWS Lambda 関数や、DBA や DevOps エンジニアに E メールや SMS を送信してイベントを通知する Amazon SNS トピックなどが利用可能です。
- Amazon CloudWatch Logs サービスを利用すると、Amazon RDS for MySQL、MariaDB DB インスタンス、AWS CloudTrail など、あらゆるアプリケーション、システム、AWS サービスから取得したログファイルのストレージを一元化できます。この機能を DB インスタンスで有効にすると、Amazon RDS から CloudWatch Logs に次のログが自動発行されます。
  - エラーログ
  - スロークエリログ
  - 全般ログ
  - 監査ログ

  CloudWatch Logs Insights を使用すると、ログデータへのクエリと、そうしたデータの分析を行えます。この機能には、専用のクエリ言語も用意されており、これによって、定義したパターンに一致するログイベントを検索しやすくなります。例えば、MySQL DB インスタンスのテーブル破損を追跡するには、エラーログファイルで次のパターンをモニタリングします: `"ERROR 1034 (HY000): Incorrect key file for table '*'; try to repair it OR Table * is marked as crashed"`。フィルタリングしたログデータは、CloudWatch メトリクスに変換できます。変換後、そのメトリクスを使用して、グラフや表形式データのあるダッシュボードを作成することも、定義したしきい値を超えた場合にアラームを設定することも可能です。これが特

に有用なのは、監査ログを使用する場合です。なぜなら、モニタリングを自動化して、想定外の動作や不審な動作が検出された際にアラートを送信し、是正アクションを実行できるからです。データベースログへのアクセスとそうしたログの管理には、AWS マネジメントコンソール、AWS CLI、Amazon RDS API、AWS SDK for CloudWatch Logs を使用できます。

- AWS CloudTrail では、AWS アカウント のユーザーおよび API のアクティビティを口グに記録し、継続的にモニタリングします。このツールは、Amazon RDS for MySQL または MariaDB DB インスタンスの監査、セキュリティモニタリング、運用上のトラブルシューティングに役立ちます。CloudTrail は Amazon RDS と統合されています。つまり、すべてのアクションがログに残り、Amazon RDS のユーザー、ロール、または AWS サービスによって実行されたアクションの記録を確認できます。例えば、ユーザーが Amazon RDS DB インスタンスを新規作成すると、イベントが検出され、ログには、リクエストされたアクション (`"eventName": "CreateDBInstance"`)、アクションの日時 (`"eventTime": "2022-07-30T22:14:06Z"`)、リクエストパラメータ (`"requestParameters": {"dBInstanceIdentifier": "test-instance", "engine": "mysql", "dBInstanceClass": "db.m6g.large"}`) などの情報が出力されます。CloudTrail のログイベントには、Amazon RDS コンソールからの呼び出しも、Amazon RDS API を使用するコードからの呼び出しも含まれています。

サードパーティー製モニタリングツール

一部のシナリオでは、AWS が Amazon RDS に用意したクラウドネイティブの包括的な、オブザーバビリティおよびモニタリングツールスイートに加え、他のソフトウェアベンダーのモニタリングツールを使用することもできます。こうしたシナリオでは、ハイブリッドデプロイが行われます。例えば、オンプレミスデータセンターで多数のデータベースが稼働し、AWS クラウド でもデータベースが複数稼働しているといったシナリオです。自社で既にオブザーバビリティソリューションを確立している場合は、既存のツールを引き続き使用して、それらを AWS クラウド のデプロイ環境に拡張すると良いでしょう。サードパーティー製モニタリングソリューションを設定する場合の一般的な課題は、マネージドサービスである Amazon RDS によって、保護が適用されることです。例えば、データベースホストマシンへのアクセスが拒否されるため、DB インスタンスが稼働するホストオペレーティングシステムにエージェントソフトウェアをインストールすることはできません。ただし、多くのサードパーティー製モニタリングソリューションを Amazon RDS と統合できる方法があります。CloudWatch やその他の AWS クラウド サービス上にそれらを構築するのです。例えば、Amazon RDS メトリクス、ログ、イベント、トレースをエクスポートして、サードパーティー製モニタリングツールにインポートすることで、より詳細な分析、可視化、アラート発行を行えます。これらのサードパーティー製ソリューションには、Prometheus、Grafana、Percona などがあります。

Prometheus と Grafana

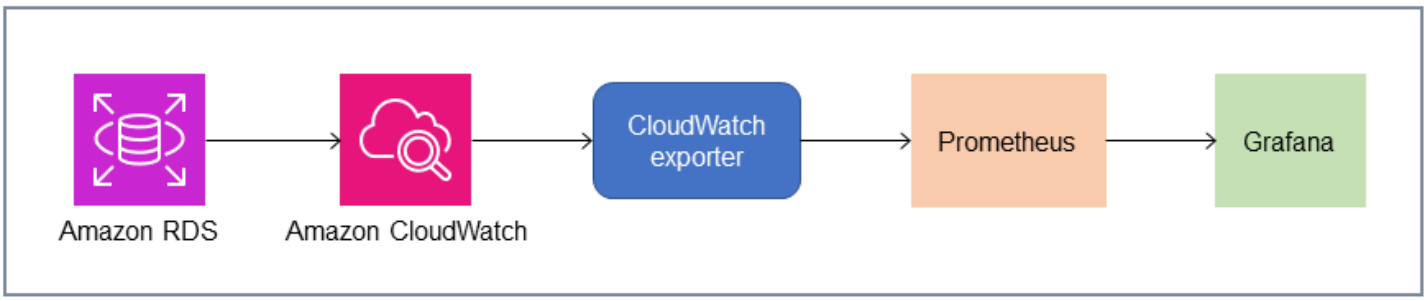
Prometheus は、設定されたターゲットから特定の間隔でメトリクスを収集するオープンソースのモニタリングソリューションであり、あらゆるアプリケーションやサービスをモニタリングできる汎用モニタリングソリューションでもあります。Amazon RDS DB インスタンスをモニタリングすると、CloudWatch では、Amazon RDS からメトリクスが収集されます。その後、YACE Exporter や CloudWatch Exporter などのオープンソースエクスポーターを使用して、メトリクスが Prometheus サーバーにエクスポートされます。

- YACE Exporter では、データエクスポートのタスクを最適化するために、CloudWatch API にリクエストを 1 回のみ送信し、複数のメトリクスを取得します。メトリクスが Prometheus サーバーに保存されると、ルール式がそこで評価され、指定した条件が観測されたときにアラートが生成されます。
- CloudWatch Exporter は Prometheus によって公式に管理されています。このエクスポータにより、CloudWatch API を介して CloudWatch メトリクスを取得します。さらに、HTTP エンドポイントへの REST API リクエストを使用して、Prometheus と互換性のある形式で、それらのメトリクスを Prometheus サーバーに保存します。

エクスポーターを選択して、デプロイモデルを設計し、エクスポーターインスタンスを設定するときには、CloudWatch と CloudWatch Logs のサービスおよび API クォータの利用を検討してください。なぜなら、CloudWatch メトリクスを Prometheus サーバーにエクスポートする機能は CloudWatch API を介して実装されるからです。例えば、CloudWatch Exporter の複数のインスタンスを単一の AWS アカウント およびリージョンにデプロイし、数百の Amazon RDS DB インスタンスをモニタリングすると、スロットリングエラー (ThrottlingException) とコード 400 エラーが発生する可能性があります。こうした制限を取り払うには、YACE エクスポーターの使用を検討すると良いでしょう。このエクスポーターは、1 回のリクエストで最大 500 個の異なるメトリクスを収集するように最適化されています。さらに、多数の Amazon RDS DB インスタンスをデプロイするには、ワークロードを単一の AWS アカウント に集中させずに複数の AWS アカウント を使用することと、各 AWS アカウント のエクスポーターインスタンス数を制限することを検討しなければなりません。

アラートは Prometheus サーバーによって生成され、アラートマネージャーによって処理されます。このツールでは、アラートの重複排除、グループ化、適切な受信者 (E メール、SMS、Slack ユーザー) へのルーティング、自動応答アクションの開始が処理されます。Grafana という別のオープンソースツールでは、こうしたメトリクスを可視化でき、高度なグラフ、ダイナミックダッシュボードといった豊富な視覚化ウィジェットに加え、アドホッククエリやダイナミックドリルダウンなどの分

析機能を利用可能です。Grafana は、ログを検索し分析する機能に加え、メトリクスとログを継続的に評価して、データがアラートルールに一致したら通知が送信されるアラート機能も備えています。



Percona

Percona Monitoring and Management (PMM) は、MySQL および MariaDB 向けに無料で提供されている、オープンソースのデータベースモニタリング、管理、オブザーバビリティソリューションであり、これによって、DB インスタンスとそのホストから数千のパフォーマンスメトリクスを収集できます。また、ダッシュボードでデータを可視化するウェブ UI や、データベースヘルス評価の自動アドバイザーといった、追加機能の利用も可能です。PMM を使用すると、Amazon RDS をモニタリングできますが、PMM クライアント (エージェント) は、ホストにアクセスできないため、Amazon RDS DB インスタンスの基盤ホストにはインストールできません。代わりに、このツールでは、Amazon RDS DB インスタンスに接続して、サーバー統計、`INFORMATION_SCHEMA`、sys スキーマ、パフォーマンススキーマをクエリし、CloudWatch API を使用してメトリクス、ログ、イベント、トレースを取得します。PMM には AWS Identity and Access Management (IAM) ユーザーアクセスキー (IAM ロール) を持たせる必要があり、これによって、モニタリングに使用可能な Amazon RDS DB インスタンスを自動検出します。PMM ツールは、データベースのモニタリング用にプロファイリングされているため、データベース固有のメトリクスを Prometheus よりも多く収集できます。PMM クエリ分析ダッシュボードを使用するには、クエリソースとしてパフォーマンススキーマを設定する必要があります。クエリ分析エージェントが Amazon RDS 用にインストールされておらず、スロークエリログを読み取れないからです。代わりに、MySQL および MariaDB の DB インスタンスから `performance_schema` を直接クエリし、メトリクスを取得します。PMM の際立った特徴の 1 つは、アラート機能であり、これによって、データベース内で検出した問題について、DBA にアラートを発行し、アドバイスを提示できます。PMM には、一般的なセキュリティ脅威、パフォーマンスの低下、データ損失、データ破損を検出できる一連のチェック機能も用意されています。

こうしたツールだけでなく、市販のオブザーバビリティおよびモニタリングソリューションにも Amazon RDS と統合できるものがあります。その例には、Datadog Database

Monitoring、Dynatrace Amazon RDS Monitoring、AppDynamics Database Monitoring などがあります。

DB インスタンスのモニタリング

DB インスタンスとは、Amazon RDS の基本的な構成要素であり、クラウド上で動作する独立したデータベース環境でもあります。MySQL および MariaDB データベースの DB インスタンスは、MySQL サーバーとも呼ばれる mysqld プログラムで、これは、SQL パーサー、クエリオプティマイザ、スレッド/接続ハンドラー、システム変数とステータス変数、1 つ以上のプラガブルストレージエンジンといった、複数のスレッドやコンポーネントで構成されます。ストレージエンジンはそれぞれ、特殊なユースケースに対応できるよう設計されています。デフォルトの推奨ストレージエンジンは InnoDB です。InnoDB は、汎用的なトランザクショナルリレーショナルデータベースエンジンであり、原子性、一貫性、独立性、耐久性 (ACID) モデルに準拠しています。また、インメモリ構造 (バッファプール、変更バッファ、アダプティブハッシュインデックス、ログバッファ) とオンディスク構造 (テーブルスペース、テーブル、インデックス、UNDO ログ、REDO ログ、二重書き込みバッファファイル) を特徴としています。データベースを ACID モデルに厳密に準拠させるために、InnoDB ストレージエンジンには、トランザクション、コミット、ロールバック、クラッシュリカバリ、行レベルのロック、マルチバージョン同時実行制御 (MVCC) といった多数の機能が実装されています。

DB インスタンスのこうしたすべての内部コンポーネントが連携して動作することで、データの可用性、整合性、セキュリティが想定どおりのパフォーマンスレベルで維持されます。ワークロードによっては、各コンポーネントと機能が、CPU、メモリ、ネットワーク、ストレージサブシステムなどのリソースに負荷をかける場合があります。特定のリソース需要が急増し、プロビジョニングした容量や、そのリソース (設定パラメータまたはソフトウェア設計によって要求される) のソフトウェア制限を超えると、DB インスタンスのパフォーマンスが低下したり、完全に利用できない、あるいは破損という状態になったりする可能性があります。したがって、これらの内部コンポーネントを測定およびモニタリングして、定義済みのベースライン値と比較し、モニタリングした値が期待値から逸脱した場合にアラートを生成することがきわめて重要です。

前述のように、さまざまなツールを使用して、MySQL インスタンスと MariaDB インスタンスをモニタリングできます。モニタリングとアラートには Amazon RDS Performance Insights と CloudWatch のツールを使用すると良いでしょう。なぜなら、これらを Amazon RDS と連携させると、高解像度のメトリクスを収集して、最新のパフォーマンス情報をほぼリアルタイムで取得し、アラームを生成できるからです。

どのモニタリングツールを選択するかに関係なく、MySQL と MariaDB DB のインスタンスでは、パフォーマンススキーマを有効にすることをお勧めします。パフォーマンススキーマは、MySQL サーバー (DB インスタンス) のオペレーションを低レベルでモニタリングするオプション機能であり、データベース全体のパフォーマンスへの影響を最小化できるように設計されています。この機能を管

理するには、performance_schema パラメータを使用します。このパラメータはオプションですが、SQL ごとの高解像度 (1 秒) メトリクス、アクティブセッションのメトリクス、待機イベント、その他の詳細な低レベルモニタリング情報を Amazon RDS Performance Insights で収集する場合は、これを使用する必要があります。

セクション

- DB インスタンスの Performance Insights メトリクス
- DB インスタンスの CloudWatch メトリクス
- Performance Insights のメトリクスを CloudWatch に発行する

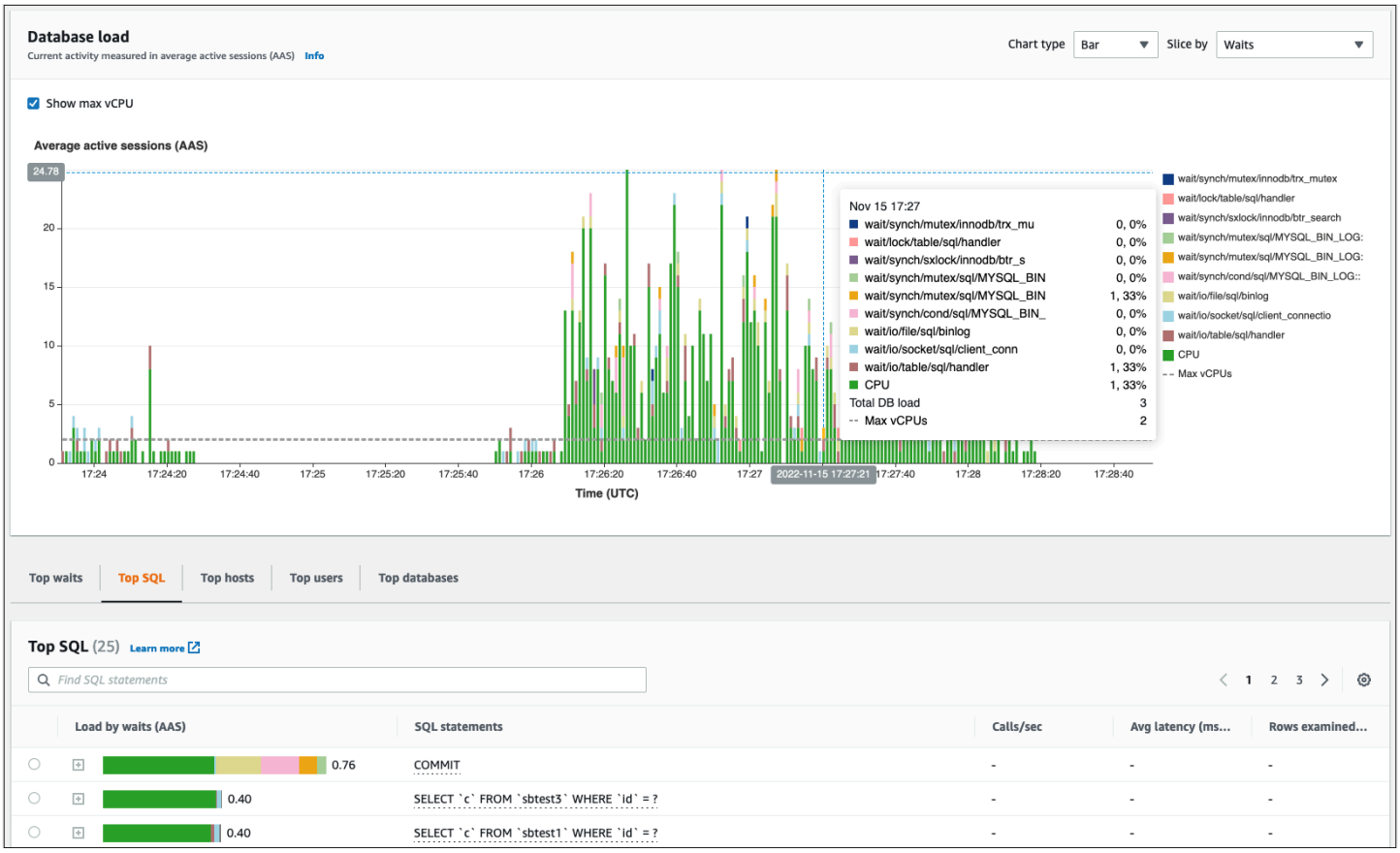
DB インスタンスの Performance Insights メトリクス

Performance Insights では、このセクションで説明するように、さまざまなタイプのメトリクスをモニタリングできます。

データベース負荷

データベースロード (DBLoad) は、データベース内のアクティビティレベルを測定する Performance Insights の主要なメトリクスであり、毎秒収集され、Amazon CloudWatch に自動的に発行されます。このメトリクスは、平均アクティブセッション (AAS) で生じている、DB インスタンスのアクティビティを表すもので、AAS とは、SQL クエリを同時実行しているセッションの数を意味します。また、DBLoad メトリクスは、その解釈に、待機、SQL、ホスト、ユーザー、データベースの 5 つのディメンションのいずれかを使用するという点が、他の時系列メトリクスとは異なっています。これらのディメンションは、DBLoad メトリクスのサブカテゴリであり、データベース負荷のさまざまな特性を表すために、カテゴリごとにスライスとして使用できます。データベース負荷の計算方法については、Amazon RDS ドキュメントの「データベース負荷」で詳しくご覧いただけます。

次のスクリーンショットは、Performance Insights ツールを示しています。



ディメンション

- 待機イベントとは、データベースセッションの処理を続行するために、リソース処理や別のオペレーションが完了するまで待機している状態を指します。`SELECT * FROM big_table` のような SQL ステートメントを実行し、そのテーブルが割り当てた InnoDB バッファプールよりもはるかに大きい場合、セッションが、`wait/io/file/innodb/innodb_data_file` 待機イベントによって待機状態になる可能性が非常に高くなります。これらのイベントはデータファイルの物理 I/O オペレーションが原因で生じます。データベースをモニタリングする場合、待機イベントは、パフォーマンスのボトルネックの可能性を示すという点で重要なディメンションとなります。このイベントは、セッション内で実行中の SQL ステートメント処理において、どのようなリソースやとオペレーションに対し、非常に長い時間待機状態になっているかを示すものです。例えば、`wait/synch/mutex/innodb/trx_sys_mutex` イベントは多数のトランザクションを持つデータベースアクティビティが多い場合に発生し、`wait/synch/mutex/innodb/buf_pool_mutex` イベントは、特定のスレッド処理において、メモリ内ページに排他的にアクセスできるよう InnoDB バッファプールがロックされている場合に発生します。MySQL と MariaDB のすべての待機イベントに関する情報については、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。計測名の解釈方法については、MySQL ドキュメントの「パフォーマンススキーマインストゥルメント命名規則」を参照してください。

- SQL は、データベースの総ロードに最も寄与している SQL ステートメントを示しています。Amazon RDS Performance Insights のデータベース負荷チャートにある上位ディメンションテーブルはインタラクティブに操作できます。[待機別の負荷 (AAS)] 列のバーをクリックすると、SQL ステートメントに関連付けられた待機イベントについて詳細なリストを取得できます。そのリストで SQL ステートメントを選択すると、関連する待機イベントが [データベース負荷] チャートに、SQL ステートメントテキストが [SQL テキスト] セクションに表示されます。SQL 統計は、[上位ディメンション] テーブルの右側に表示されます。
- ホストは、接続済みクライアントのホスト名を示しています。このディメンションにより、どのクライアントホストが非常に多くの負荷をデータベースにかけているかを特定しやすくなります。
- ユーザーを使用すると、データベースにログインしているユーザーごとに DB 負荷をグループ化できます。
- データベースを使用すると、クライアントが接続しているデータベースの名前で DB 負荷をグループ化できます。

カウンターメトリクス

カウンターメトリクスは、累積メトリクスであり、これによって、DB インスタンスの再起動時にのみ値を増加させたり、ゼロにリセットしたりできます。カウンターメトリクスの値を以前の値に減らすことはできません。これらのメトリクスは、単調に増加する 1 つのカウンターを表すものです。

- ネイティブカウンターとは、Amazon RDS ではなく、データベースエンジンによって定義されるメトリクスです。例えば、次のようになります。
  - `SQL.Innodb_rows_inserted` は、InnoDB テーブルに挿入された行数を表します。
  - `SQL.Select_scan` は、最初のテーブルのフルスキャンを完了した結合の数を表します。
  - `Cache.Innodb_buffer_pool_reads` は、InnoDB エンジンがバッファプールから読み取れず、ディスクから直接読み取る必要があった論理読み取りの数を表します。
  - `Cache.Innodb_buffer_pool_read_requests` は、論理読み取りリクエストの数を表します。

  すべてのネイティブメトリックの定義については、MySQL ドキュメントの「サーバーステータス可変」を参照してください。
- 非ネイティブカウンターは、Amazon RDS によって定義されています。これらのメトリクスを取得するには、特定のクエリを使用するか、計算に 2 つ以上のネイティブメトリクスを使用しま

す。非ネイティブカウンターメトリクスにより、レイテンシー、比率、ヒット率を表すことができます。例えば、次のようになります。

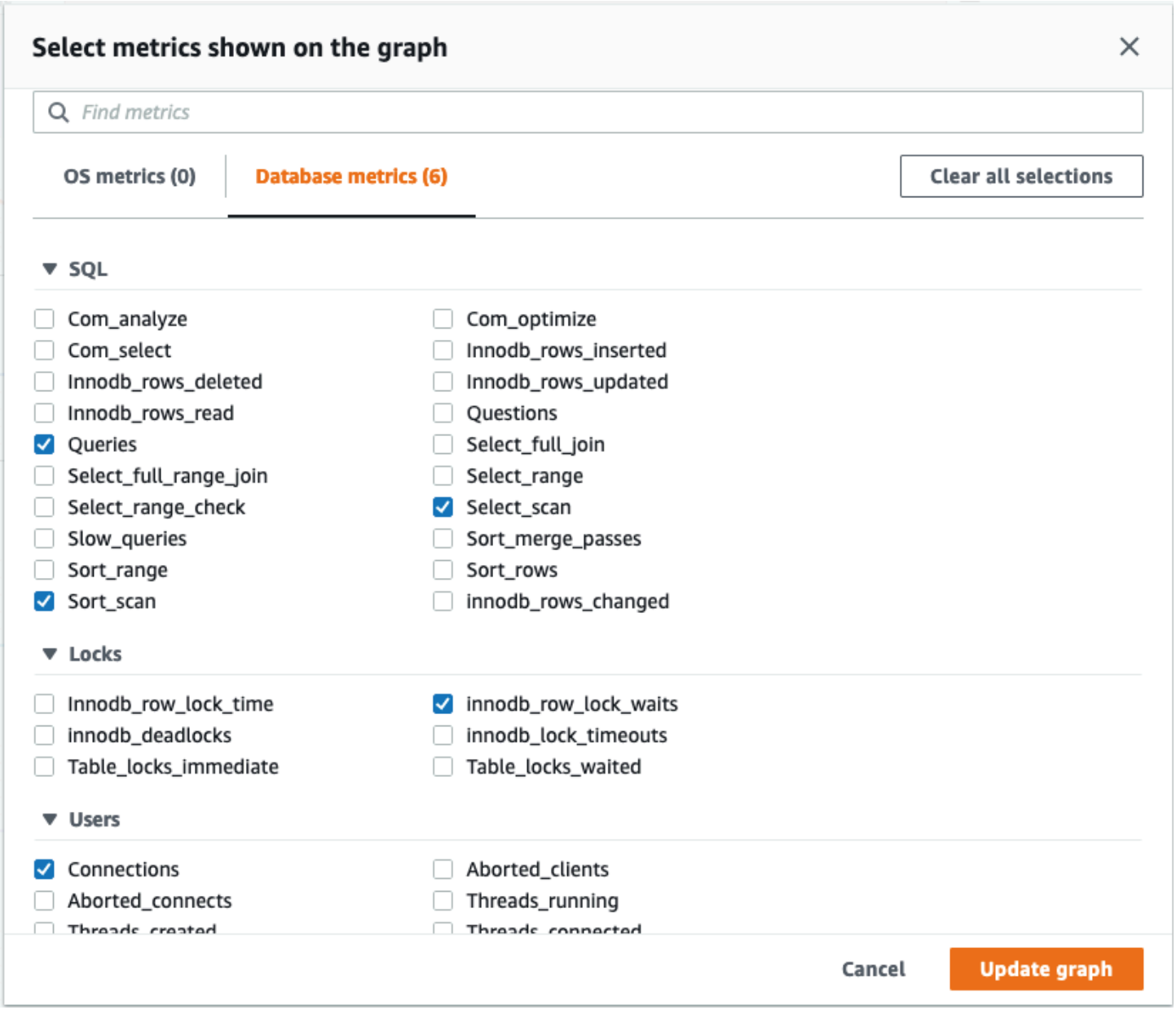
- `Cache.innoDB_buffer_pool_hits` は、InnoDB がディスクを使用せずにバッファプールから取得できる読み取りオペレーションの数を表しています。これは、ネイティブカウンターメトリクスに基づいて、次のように計算されます。

```
db.Cache.Innodb_buffer_pool_read_requests - db.Cache.Innodb_buffer_pool_reads
```

- `IO.innoDB_datafile_writes_to_disk` は、InnoDB データファイルによる、ディスクへの書き込みオペレーションの数を表しています。データファイルへのオペレーションのみをキャプチャするもので、二重書き込みや REDO ログの書き込みオペレーションはキャプチャされません。これは、次のように計算されます:

```
db.IO.Innodb_data_writes - db.IO.Innodb_log_writes - db.IO.Innodb_dblwr_writes
```

DB インスタンスメトリクスは、Performance Insights ダッシュボードで直接視覚化できます。次の図に示すように、[メトリクスを管理] を選択して、[データベースメトリクス] タブを選択し、目的のメトリクスを選択します。



[グラフを更新] ボタンを選択すると、次の図に示すように、選択したメトリクスが表示されます。

SQL 統計

Performance Insights では、クエリを実行している 1 秒ごと、および SQL コールごとに、SQL クエリに関するパフォーマンス関連メトリクスを収集します。一般的には、ステートメントおよびダイジェストレベルで SQL 統計を収集しますが、MariaDB および MySQL DB インスタンスの場合、ダイジェストレベルでのみ収集します。

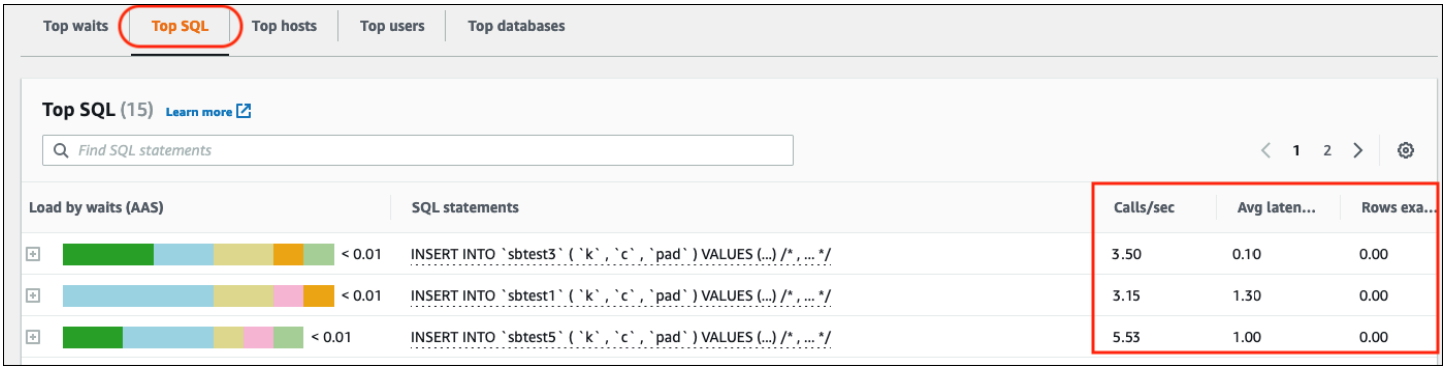
- ダイジェスト統計は、複合メトリクスであり、パターンが同じであっても最終的には異なるリテラル値を持つすべてのクエリで構成されます。ダイジェストでは、次のように、特定のリテラル値を変数に置き換えます。

```
SELECT department_id, department_name FROM departments WHERE location_id = ?
```

- ダイジェストした SQL ステートメントごとに 1 秒あたりの統計を表すメトリクスも用意されています。例えば、`sql_tokenized.stats.count_star_per_sec` は、1 秒あたりの呼び出し回数 (SQL ステートメントが 1 秒あたりに実行された回数) を表します。

- Performance Insights は、各 SQL ステートメントの呼び出しごとの統計情報を得られるメトリクスも備えています。例えば、`sql_tokenized.stats.sum_timer_wait_per_call` は、SQL ステートメント 1 回あたりの平均レイテンシーをミリ秒単位で示しています。

SQL 統計情報は、Performance Insights ダッシュボードの [上位ディメンション] テーブルにある [上位 SQL] タブで確認できます。



DB インスタンスの CloudWatch メトリクス

Amazon CloudWatch には、Amazon RDS によって自動的に公開されるメトリクスも用意されています。`AWS/RDS` 名前空間に存在するメトリクスは、インスタンスレベルのメトリクスですが、mysqld プロセスにおける厳密な意味での DB インスタンスではなく、Amazon RDS (サービス) のインスタンス (クラウドで稼働する独立したデータベース環境) を指します。したがって、こうしたデフォルトのメトリクスのほとんどは、用語の厳密な定義において、OS メトリクスのカテゴリに分類されます。その例として、`CPUUtilization`、`WriteIOPS`、`SwapUsage` などが挙げられます。ただし、MariaDB と MySQL に適用される次のような DB インスタンスメトリクスがあります。

- `BinLogDiskUsage` – バイナリログが占めるディスク容量。
- `DatabaseConnections` - DB インスタンスへのクライアントネットワーク接続の数。
- `ReplicaLag` – リードレプリカ DB インスタンスとソース DB インスタンス間で生じるタイムラグ。

Performance Insights のメトリクスを CloudWatch に発行する

Amazon RDS Performance Insights は、ほとんどの DB インスタンスのメトリクスとディメンションをモニタリングし、 AWS マネジメントコンソールの Performance Insights ダッシュボードから利用できるようにします。このダッシュボードは、データベースのトラブルシューティングと根本原因の分析に適していますが、パフォーマンス関連メトリクスのアラームを Performance Insights で作成す

ることはできません。Performance Insights メトリクスに基づいてアラームを作成する場合は、それらのメトリクスが CloudWatch に存在する必要があります。

Performance Insights により、メトリクスが CloudWatch に自動的に公開されます。Performance Insights からも同じデータを照会できますが、CloudWatch にメトリクスが存在すると、CloudWatch アラームの追加や既存の CloudWatch ダッシュボードへのメトリクスの追加が容易になります。カウンターは、オペレーティングシステムとデータベースのパフォーマンスメトリクスであり、その例には、`os.memory.free` や `db.Locks.Innodb_row_lock_time` などがあります。OS メトリクスの収集は、拡張モニタリングの設定によって異なります。拡張モニタリングがオフになっている場合、OS メトリックは 1 分ごとに収集されます。拡張モニタリングがオンになっている場合、OS メトリクスは、選択した期間を対象に収集されます。詳細については、Amazon RDS ドキュメントの「拡張モニタリングのオンとオフを切り替える」を参照してください。

Performance Insights を使用すると、DB インスタンス用に事前設定した、またはカスタマイズしたメトリクスダッシュボードを CloudWatch にエクスポートできます。メトリクスダッシュボードを新しいダッシュボードとして エクスポートするか、それらを既存の CloudWatch ダッシュボードに追加できます。Performance Insights メトリクスダッシュボードを CloudWatch ダッシュボードにエクスポートすると、システムの状態を包括的な統一ビューで確認できます。具体的には、EC2 インスタンス、Amazon Elastic File System (Amazon EFS) リソース、Elastic Load Balancing (ELB) リソースといった、システム内の各種リソースに関連付けたメトリクスの概要に加え、DB インスタンスのメトリクスも取得することが可能です。

また、CloudWatch `DB_PERF_INSIGHTS` メトリクス数学関数を使用すると、CloudWatch の Performance Insights メトリクスに基づいて、アラームとグラフをクエリおよび作成できます。Performance Insights メトリクスにアラームを作成するには、CloudWatch ドキュメントの手順に従います。例えば、DB インスタンス内のアクティブなトランザクションの合計が特定のしきい値に達したときにアラームをトリガーする場合は、そのページの手順に従って、`DB_PERF_INSIGHTS` 数式を使用し、[適用] を選択します。

```
DB_PERF_INSIGHTS('RDS', 'db-BQ2TPYY7HG2GDFC7APMB3BVB3M',
 'db.Transactions.active_transactions.avg')
```

この式の `db-BQ2TPYY7HG2GDFC7APMB3BVB3M` は DB インスタンスのリソース ID です。期間 (1 分など) と条件 (1000 より大きいなど) を指定します。アラームの作成を完了するには、アラームアクションを設定して、名前と説明を追加します。その後、アラームをプレビューし作成します。

OS のモニタリング

Amazon RDS for MySQL または MariaDB の DB インスタンスは、Linux オペレーティングシステムで稼働しており、この OS は、基盤システムリソース、つまり、CPU、メモリ、ネットワーク、ストレージを使用しています。

```
MySQL [(none)]> SHOW variables LIKE 'version%';
+-------------------------+----------------------+
| Variable_name           | Value                |
+-------------------------+----------------------+
| version                 | 8.0.28               |
| version_comment         | Source distribution  |
| version_compile_machine | aarch64              |
| version_compile_os      | Linux                |
| version_compile_zlib    | 1.2.11               |
+-------------------------+----------------------+
5 rows in set (0.00 sec)
```

データベースと基盤オペレーティングシステムの全体的なパフォーマンスは、システムリソースの使用率によって大きく異なります。例えば、CPU は、システムパフォーマンスにかかわる重要なコンポーネントと言えます。データベースソフトウェアの指示を実行するとともに、他のシステムリソースも管理しているからです。CPU の使用率が高すぎる場合 (つまり、負荷の処理に DB インスタンスにプロビジョニングした CPU 性能よりも高い性能が必要な場合)、この問題の影響は、データベースのパフォーマンスと安定性、ひいてはアプリケーションにまで及ぶでしょう。

メモリの割り当てと解放は、データベースエンジンによって動的に行われます。RAM に現在の作業を実行するのに十分なメモリがない場合、ディスクに存在するスワップメモリにメモリページが書き込まれます。ディスクの読み書きはメモリよりもはるかに遅いため、SSD NVMe 技術に基づくディスクであっても、メモリの過剰な割り当ては、パフォーマンス低下を招きます。メモリ使用率が高いと、データベースレスポンスのレイテンシーが増大します。記憶域を増やすために、ページファイルのサイズも大きくなるからです。メモリの割り当てが過剰になり、RAM とスワップメモリのスペースがともに枯渇すると、データベースサービスが利用できなくなる可能性があります。これによって、ユーザー側で、`[ERROR] mysqld: Out of memory (Needed xyz bytes)` などのエラーが発生しかねません。

MySQL および MariaDB データベース管理システムでは、ディスク上の構造を保存するディスクからなるストレージサブシステムが使用されます。構造とは、テーブル、インデックス、バイナリログ、REDO ログ、UNDO ログ、二重書き込みバッファファイルなどを指します。そのような理由か

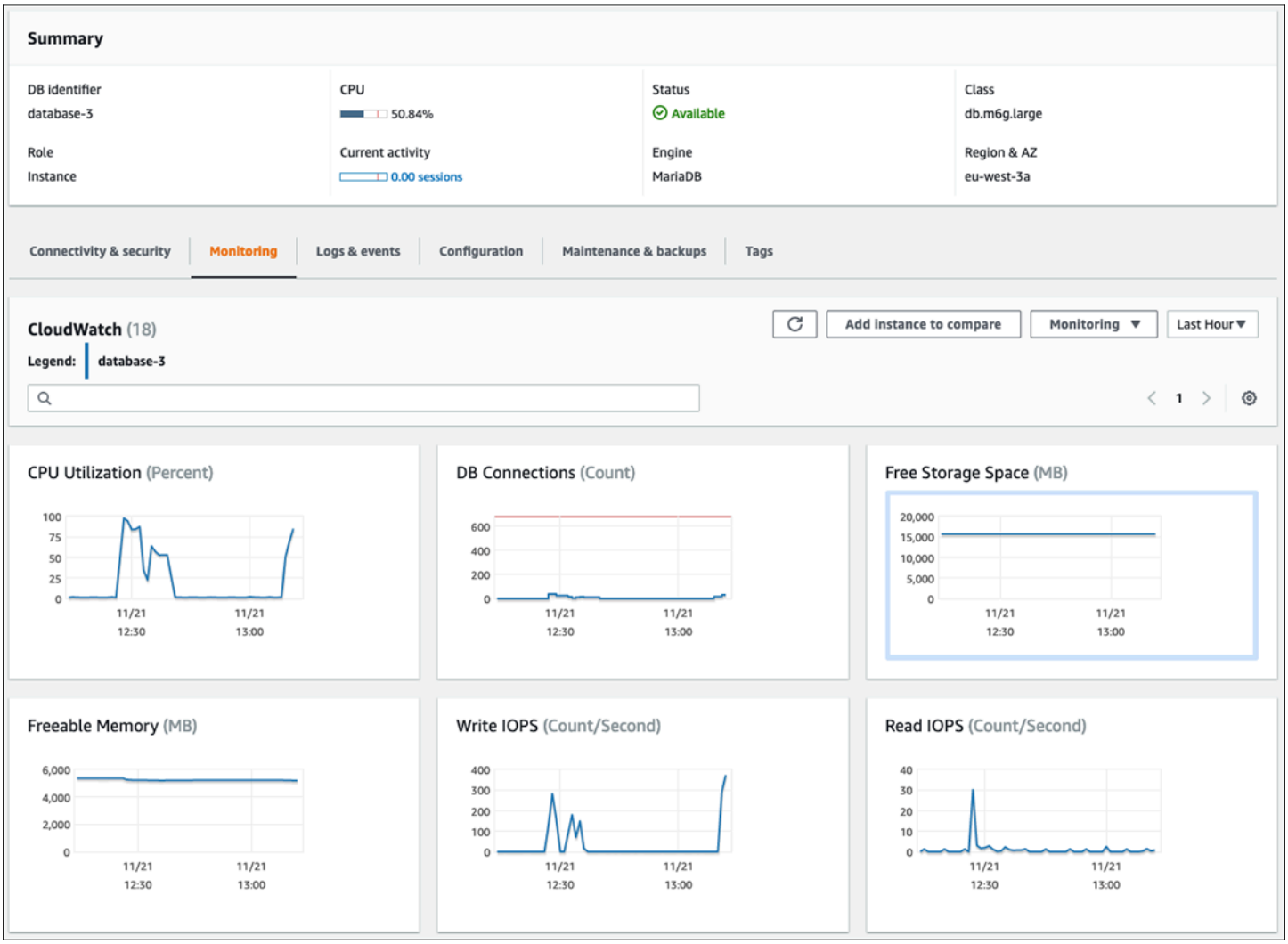
ら、データベースでは、他のタイプのソフトウェアとは対照的に、大量のディスクアクティビティの実行が必要となります。データベースオペレーションを最適化するには、ディスク I/O 使用率とディスク容量の割り当てをモニタリングし調整することが重要です。ディスクで対応可能な最大 IOPS またはスループットの上限に達すると、データベースのパフォーマンスに影響が及びかねません。例えば、インデックススキャンによるランダムアクセスが急増すると、1 秒あたりに多数の I/O オペレーションが発生し、最終的に基盤ストレージの上限に達する可能性があります。フルテーブルスキャンの場合、IOPS の上限には達しない可能性がありますが、高スループット (1 秒あたりのメガバイト数で測定) が発生することがあります。ディスク容量の割り当て状況をモニタリングし、アラートを発行することは、きわめて重要です。`OS error code 28: No space left on device` などのエラーが発生すると、データベースが利用できなくなったり、データが破損したりする恐れがあるからです。

Amazon RDS では、DB インスタンスが稼働するオペレーティングシステムのメトリクスをリアルタイムで確認でき、1 つの OS メトリクスセットが CloudWatch に自動的に発行されます。これらのメトリクスは、Amazon RDS コンソールと CloudWatch ダッシュボードに表示して分析でき、CloudWatch で選択したメトリクスにアラームを設定することも可能です。以下に例を示します。

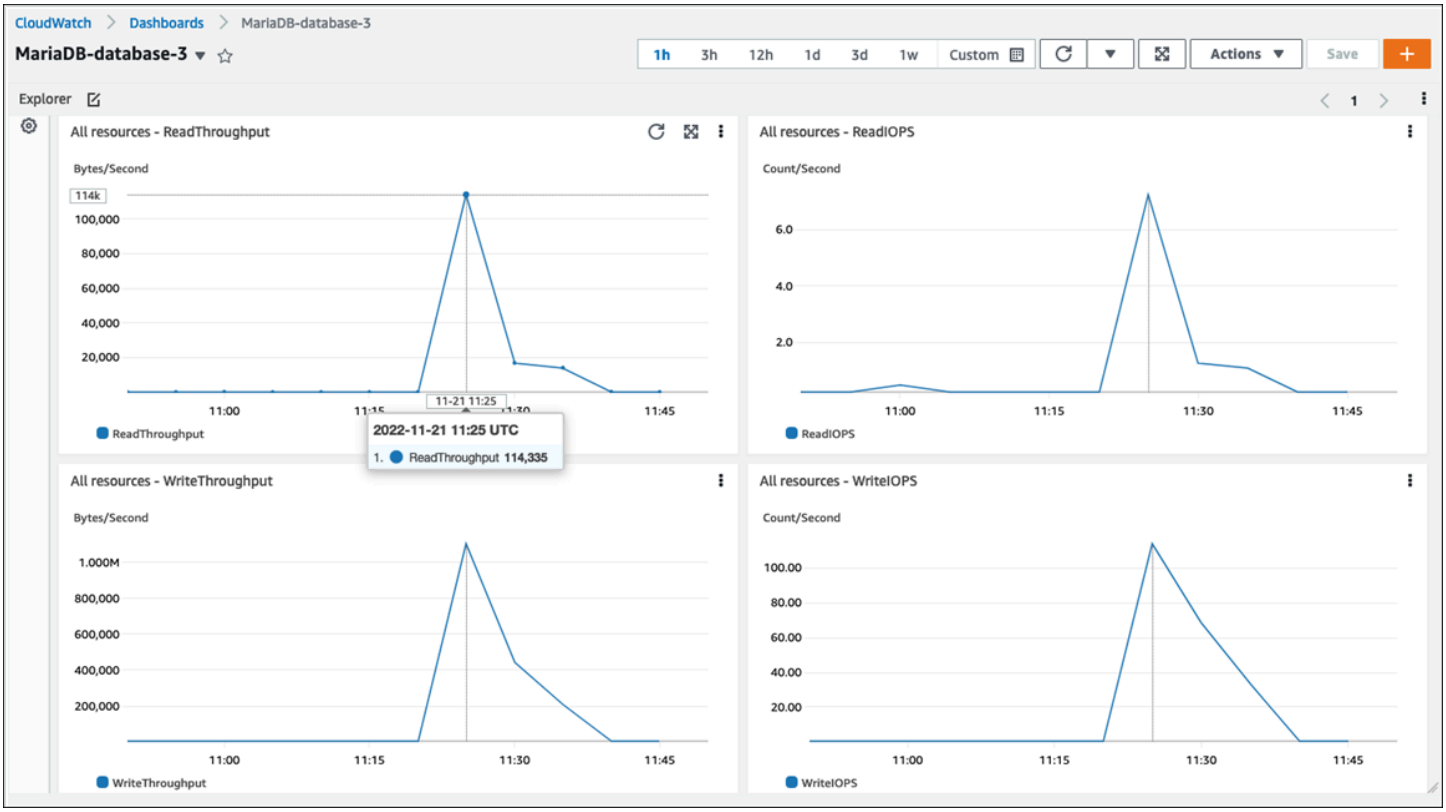
- `CPUUtilization` – CPU 使用率。
- `BinLogDiskUsage` – バイナリログが占めるディスク容量。
- `FreeableMemory` – 使用可能な RAM 容量。これは、`/proc/meminfo` の `MemAvailable` フィールドの値を示しています。
- `ReadIOPS` – 1 秒あたりのディスク読み取り I/O オペレーションの平均回数。
- `WriteThroughput` – ローカルストレージに対し 1 秒間にディスクに書き込まれる平均バイト数。
- `NetworkTransmitThroughput` – DB ノードの送信ネットワークトラフィック。データベーストラフィックと、モニタリングおよびレプリケーションに使用される Amazon RDS トラフィックの合計を示しています。

Amazon RDS から CloudWatch に発行される全メトリクスの総合リファレンスについては、Amazon RDS ドキュメントの「Amazon RDS の Amazon CloudWatch メトリクス」を参照してください。

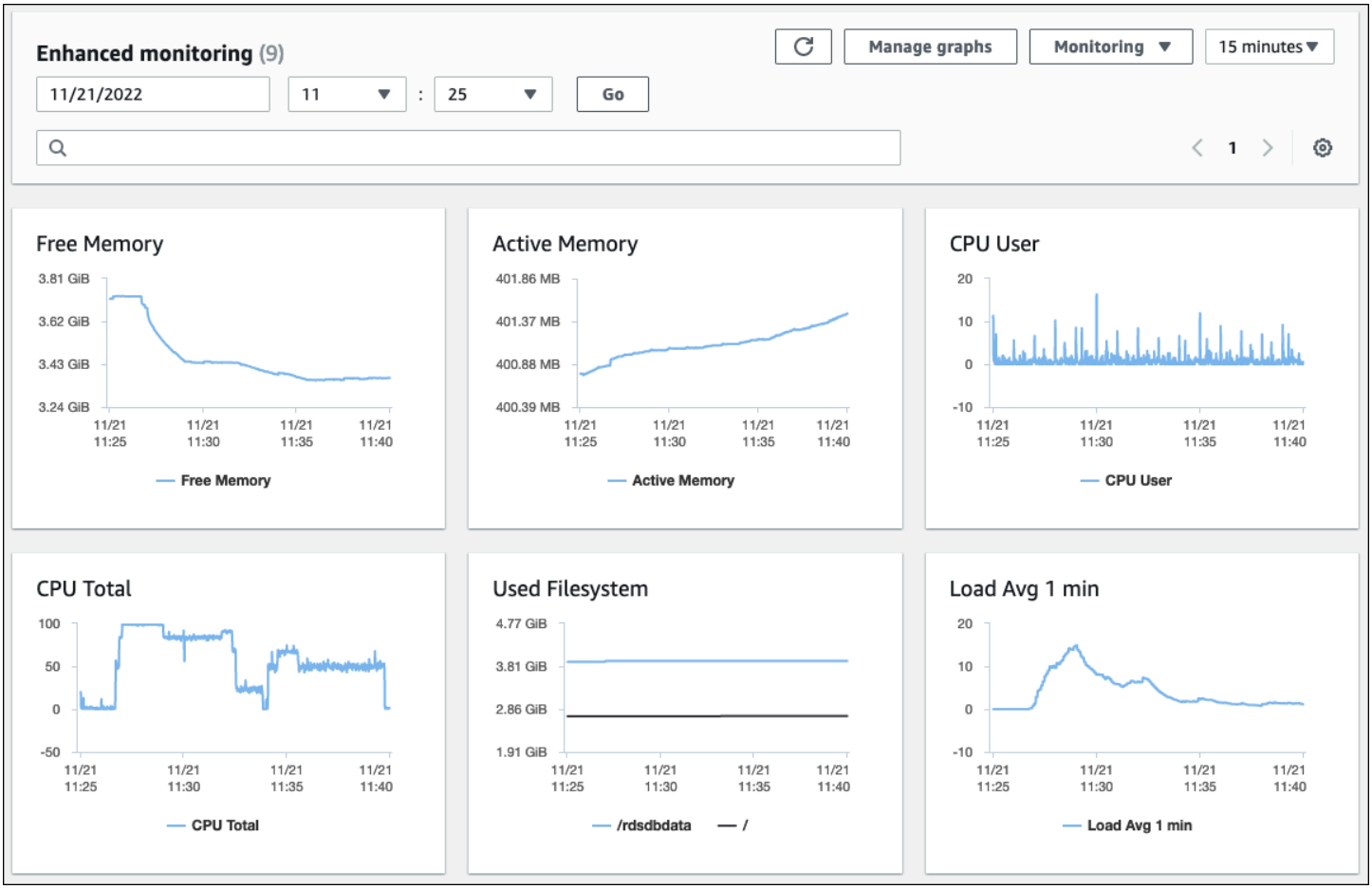
次の画像は、Amazon RDS コンソールに表示される Amazon RDS 向け CloudWatch メトリクスの例を示しています。



次の画像は、CloudWatch ダッシュボードに表示される同様のメトリクスを示しています。



他の一連の OS メトリクスは、Amazon RDS の拡張モニタリングによって収集します。このツールによって、リアルタイムのシステムメトリクスと OS プロセス情報を得ることで、Amazon RDS for MariaDB および Amazon RDS for MySQL DB インスタンスのヘルスをより詳細に可視化できます。DB インスタンスで拡張モニタリングを有効にし、目的の粒度を設定すると、オペレーティングシステムのメトリクスとプロセス情報が収集されます。こうした情報は、次の画像に示すように、Amazon RDS コンソールで表示し分析できます。



拡張モニタリングに用意されている主要なメトリクスを次に示します。

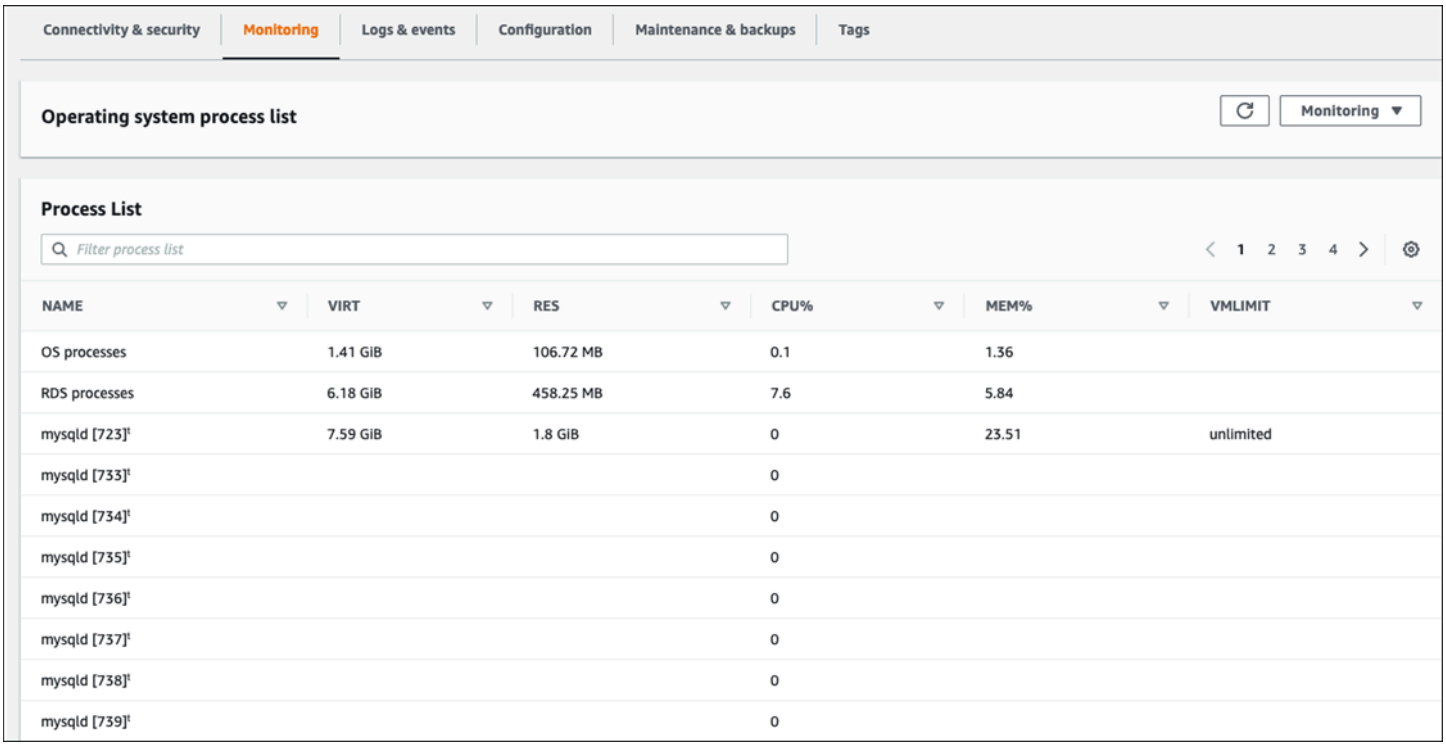
- `cpuUtilization.total` – 使用中の CPU の合計使用率。
- `cpuUtilization.user` – ユーザープログラムが使用中の CPU の使用率。
- `memory.active` – 割り当てられたメモリの量 (キロバイト単位)。
- `memory.cached` - ファイルシステムベースの I/O のキャッシュに使用されたメモリの量。
- `loadAverageMinute.one` – 過去 1 分間に CPU 時間をリクエストしたプロセスの数。

メトリクスが網羅されたリストについては、Amazon RDS ドキュメントの「拡張モニタリングの OS メトリクス」を参照してください。

Amazon RDS コンソールの OS プロセスリストには、DB インスタンスで稼働している各プロセスの詳細が表示されます。リストは 3 つのセクションに分かれています。

- OS プロセス – このセクションは、すべてのカーネルプロセスとシステムプロセスを集約した概要を示しています。一般的に、これらのプロセスによって、データベースパフォーマンスへの影響が最小化されます。
- RDS プロセス – このセクションは、Amazon RDS DB インスタンスのサポートに必要な AWS プロセスの概要を示しています。例えば、Amazon RDS 管理エージェント、モニタリングおよび診断プロセス、その他の同様のプロセスなどがこれに該当します。
- RDS 子プロセス – このセクションは、DB インスタンスをサポートする Amazon RDS プロセスの概要を示しています (この画像では `mysqld` プロセスとそのスレッド)。`mysqld` スレッドは親 `mysqld` プロセスの下にネストされて表示されます。

次の画像は、Amazon RDS コンソールの OS プロセスリストを示しています。

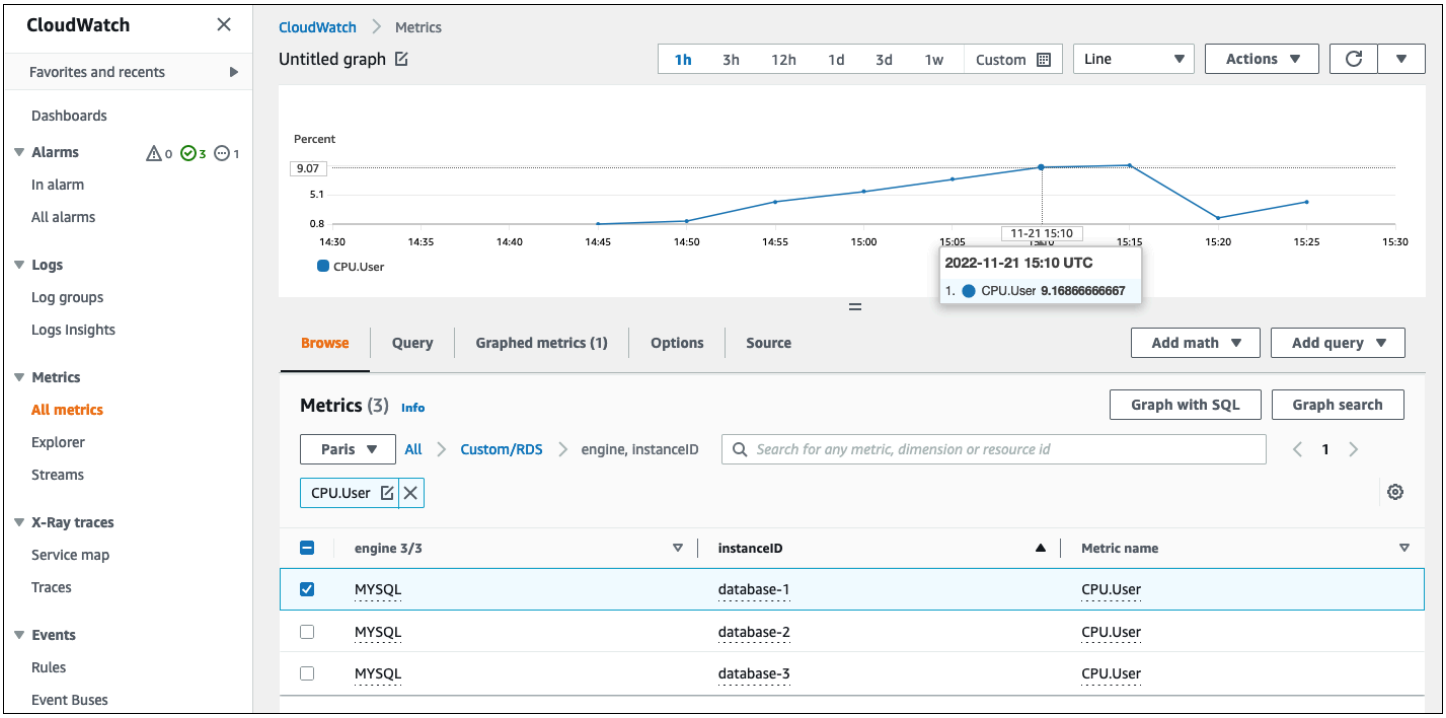


Amazon RDS では、拡張モニタリングで収集したメトリクスが、CloudWatch Logs アカウントに配信されます。Amazon RDS コンソールに表示するモニタリングデータは、CloudWatch Logs から取得します。DB インスタンスのメトリクスは、ログストリームとして CloudWatch Logs から取得することもでき、これらのメトリクスは JSON 形式で保存されます。選択したモニタリングシステムで CloudWatch Logs からの拡張モニタリング JSON 出力を使用できます。

CloudWatch ダッシュボードにグラフを表示したり、メトリックが定義済みのしきい値を超えた場合にアクションを実行するアラームを作成したりするには、CloudWatch でメトリクスフィルターを作成して、CloudWatch Logs からメトリクスを取り出す必要があります。詳細な手順について

は、AWS re:Post の記事で、拡張モニタリングの CloudWatch Logs をフィルタリングして Amazon RDS の自動カスタムメトリクスを生成する方法をご確認ください。

次の画像の例は、Custom/RDS 名前空間の CPU.User カスタムメトリクスを示しています。このカスタムメトリクスを作成するには、CloudWatch Logs にある cpuUtilization.user 拡張モニタリングメトリクスをフィルタリングします。



CloudWatch リポジトリでメトリクスが利用可能になったら、CloudWatch ダッシュボードでそれらを表示して分析したり、数学およびクエリオペレーションをさらに適用したりできます。また、アラームを設定してこの特定のメトリクスをモニタリングし、観測値が定義済みのアラーム条件と一致しない場合にアラートを発行することも可能です。

イベント、ログ、監査証跡

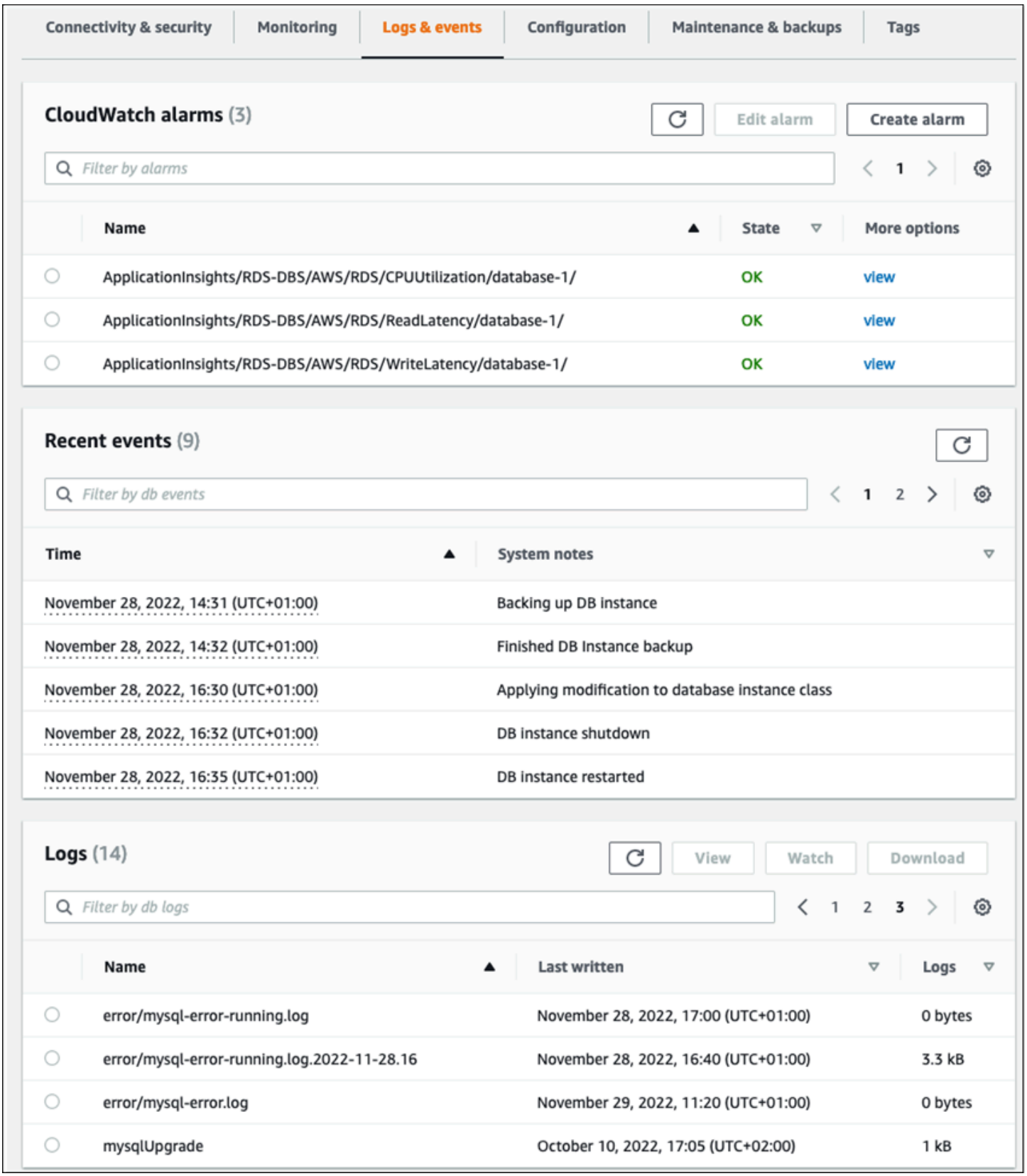
DB インスタンスメトリクスと OS メトリクスのモニタリング、傾向分析、メトリクスとベースライン値の比較、定義済みしきい値を超えた場合のアラート生成は、いずれも必要なオペレーションであり、Amazon RDS DB インスタンスの信頼性、可用性、パフォーマンス、セキュリティの確保と維持に役立つベストプラクティスでもあります。しかし、包括的なソリューションを実現するには、MySQL および MariaDB データベースのデータベースイベント、ログファイル、監査証跡もモニタリングしなければなりません。

セクション

- Amazon RDS イベント
- データベースログ
- 監査証跡

Amazon RDS イベント

Amazon RDS イベントは、Amazon RDS 環境で生じた変更を示すものです。例えば、DB インスタンスのステータスが開始中から使用可能に変わると、Amazon RDS ではイベント `RDS-EVENT-0088 The DB instance has been started` が生成されます。Amazon RDS から Amazon EventBridge へは、ほぼリアルタイムでイベントが配信されます。イベントにアクセスするには、Amazon RDS コンソール、AWS CLI コマンド describe-events、または Amazon RDS API オペレーション DescribeEvents を使用します。次の画像は、Amazon RDS コンソールに表示されるイベントとログを示しています。


Connectivity & security
Monitoring
Logs & events
Configuration
Maintenance & backups
Tags
CloudWatch alarms (3)
Edit alarm
Create alarm
Filter by alarms
1
Name
State
More options
ApplicationInsights/RDS-DBS/AWS/RDS/CPUUtilization/database-1/
OK
view
ApplicationInsights/RDS-DBS/AWS/RDS/ReadLatency/database-1/
OK
view
ApplicationInsights/RDS-DBS/AWS/RDS/WriteLatency/database-1/
OK
view
Recent events (9)
Filter by db events
1
2
Time
System notes
November 28, 2022, 14:31 (UTC+01:00)
Backing up DB instance
November 28, 2022, 14:32 (UTC+01:00)
Finished DB Instance backup
November 28, 2022, 16:30 (UTC+01:00)
Applying modification to database instance class
November 28, 2022, 16:32 (UTC+01:00)
DB instance shutdown
November 28, 2022, 16:35 (UTC+01:00)
DB instance restarted
Logs (14)
View
Watch
Download
Filter by db logs
1
2
3
Name
Last written
Logs
error/mysql-error-running.log
November 28, 2022, 17:00 (UTC+01:00)
0 bytes
error/mysql-error-running.log.2022-11-28.16
November 28, 2022, 16:40 (UTC+01:00)
3.3 kB
error/mysql-error.log
November 29, 2022, 11:20 (UTC+01:00)
0 bytes
mysqlUpgrade
October 10, 2022, 17:05 (UTC+02:00)
1 kB

Amazon RDS では、DB インスタンスイベント、DB パラメータグループイベント、DB セキュリティグループイベント、DB スナップショットイベント、RDS Proxy イベント、ブルー/グリーンデプロイイベントといった、各種イベントが発行されます。これにより、次の情報が得られます。

- ソース名とソースタイプ。例: `"SourceIdentifier": "database-1", "SourceType": "db-instance"`
- イベントの日付と時刻。例: `"Date": "2022-12-01T09:20:28.595000+00:00"`
- イベントに関連付けられたメッセージ。例: `"Message": "Finished updating DB parameter group"`
- イベントカテゴリ。例: `"EventCategories": ["configuration change"]`

網羅的なリファレンスについては、Amazon RDS ドキュメントの「Amazon RDS のイベントカテゴリとイベントメッセージ」を参照してください。

Amazon RDS イベントのモニタリングをお勧めします。なぜなら、これらのイベントは、DB インスタンスの可用性ステータス変更、設定変更、リードレプリカのステータス変更、バックアップおよびリカバリイベント、フェイルオーバーアクション、障害イベント、セキュリティグループの変更といった、さまざまな情報や通知を示すものだからです。例えば、データベースのパフォーマンスと耐久性の向上を目的にリードレプリカ DB インスタンスを設定している場合は、DB インスタンスに関連付けられたリードレプリカイベントカテゴリの Amazon RDS イベントをモニタリングすると良いでしょう。`RDS-EVENT-0057 Replication on the read replica was terminated` などのイベントは、リードレプリカがプライマリ DB インスタンスと同期されていないことを示しているからです。そのようなイベントの発生を担当チームに通知すれば、問題のタイムリーな緩和に役立ちます。Amazon EventBridge とその他の AWS のサービス (AWS Lambda、Amazon Simple Queue Service (Amazon SQS)、Amazon Simple Notification Service (Amazon SNS) など) は、データベースの可用性の問題やリソースの変更といったシステムイベントへの応答を自動化するのに有用です。

Amazon RDS コンソールでは、過去 24 時間のイベントを取得できます。AWS CLI または Amazon RDS API を使用してイベントを表示する場合は、次のように describe-events コマンドを使用して、過去 14 日間のイベントを取得できます。

```
$ aws rds describe-events --source-identifier database-1 --source-type db-instance
{
    "Events": [
        {
            "SourceIdentifier": "database-1",
            "SourceType": "db-instance",
```

```
            "Message": "CloudWatch Logs Export enabled for logs [audit, error, general,
 slowquery]",
            "EventCategories": [],
            "Date": "2022-12-01T09:20:28.595000+00:00",
            "SourceArn": "arn:aws:rds:eu-west-3:111122223333:db:database-1"
        },
        {
            "SourceIdentifier": "database-1",
            "SourceType": "db-instance",
            "Message": "Finished updating DB parameter group",
            "EventCategories": [
                "configuration change"
            ],
            "Date": "2022-12-01T09:22:40.413000+00:00",
            "SourceArn": "arn:aws:rds:eu-west-3:111122223333:db:database-1"
        }
    ]
}
```

イベントを長期的に保存 (指定した有効期限まで、あるいは永続的に保存) する場合は、CloudWatch Logs を使用して、Amazon RDS で生成されたイベントの情報をログに記録できます。このソリューションを実装するには、Amazon SNS トピックを使用して Amazon RDS イベント通知を受信し、Lambda 関数を呼び出して CloudWatch Logs にイベントをログ記録します。

1. イベントで呼び出す Lambda 関数を作成し、イベントから取得した情報を CloudWatch Logs に記録します。CloudWatch Logs は Lambda と統合されているため、`stdout` にイベント情報を書き出す print 関数を使用して、それらの情報を簡単にログに記録できます。
2. Lambda 関数へのサブスクリプションを使用して SNS トピックを作成し ([プロトコル] を Lambda に設定)、[エンドポイント] を、前のステップで作成した Lambda 関数の Amazon リソースネーム (ARN) に設定します。
3. Amazon RDS イベント通知を受信するように SNS トピックを設定します。詳細な手順については、Amazon SNS トピックで Amazon RDS 通知を受信する方法に関する AWS re:Post の記事を参照してください。
4. Amazon RDS コンソールで、イベントサブスクリプションを新規作成します。[ターゲット] を ARN に設定し、前に作成した SNS トピックを選択します。要件に応じて、[ソースタイプ] と [含めるイベントカテゴリ] を設定します。詳細については、Amazon RDS ドキュメントの「Amazon RDS イベント通知にサブスクライブする」を参照してください。

データベースログ

MySQL および MariaDB データベースでは、監査やトラブルシューティングのためにアクセスできるログが生成されます。これらのログを次にしめします。

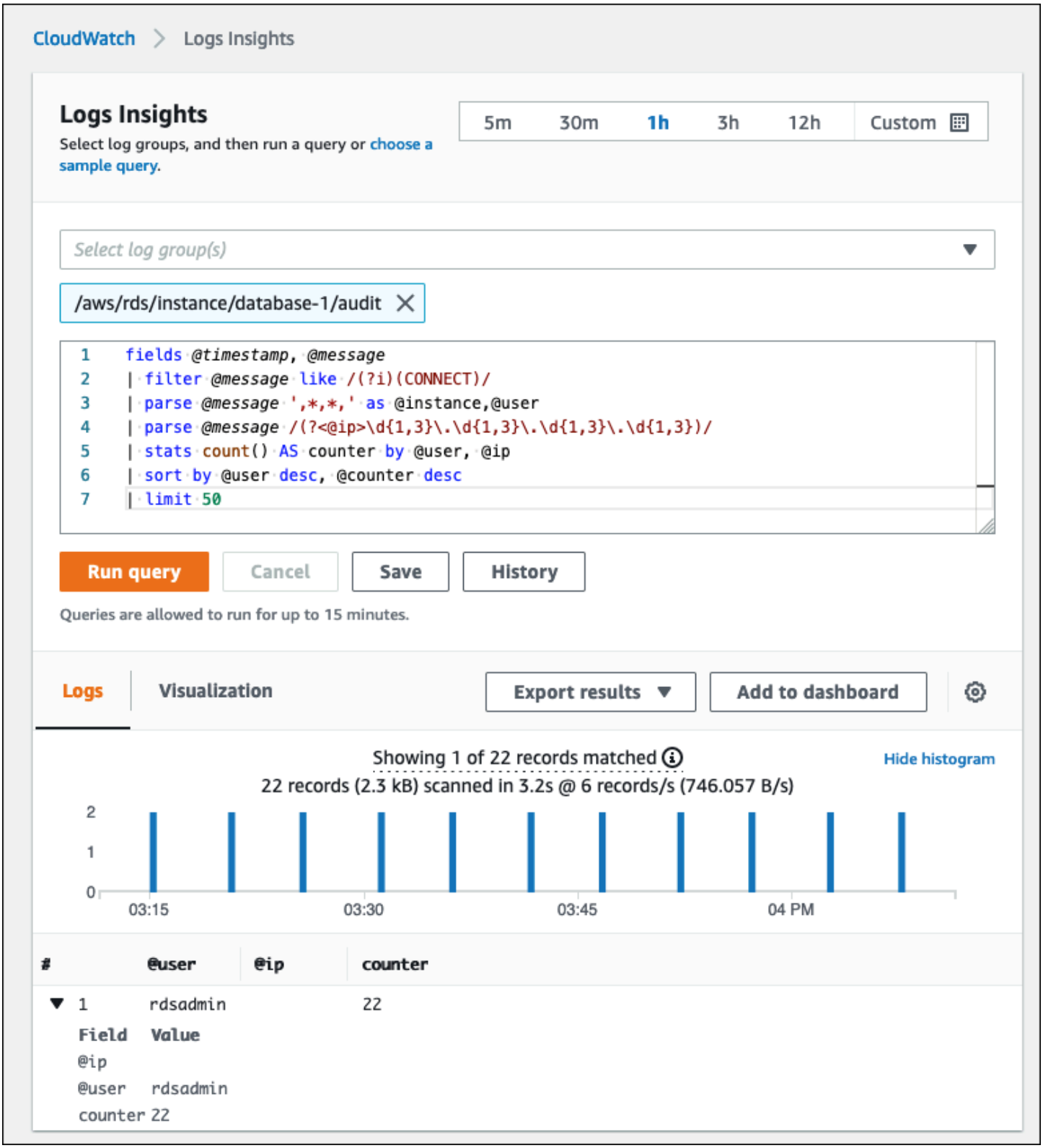
- 監査 – 監査証跡は、サーバーのアクティビティが記録されている一連のレコードです。これには、クライアントセッションごとに、サーバーに接続したユーザー (ユーザー名とホスト)、実行したクエリ、アクセスしたテーブル、変更したサーバー変数が記録されます。
- エラー – このログには、サーバー (mysqld) の起動およびシャットダウン時刻に加え、サーバーの起動、シャットダウン、稼働時に発生したエラー、警告、通知といった診断メッセージが記録されます。
- 全般 – このログには、mysqld のアクティビティ (各クライアントの接続および切断アクティビティ、クライアントから受信した SQL クエリなど) が記録されます。全般のクエリログは、エラーが疑われる場合や、クライアントが mysqld に送信した内容を正確に把握する場合に非常に有用です。
- スロークエリ – このログには、実行に長い時間がかかった SQL クエリが記録されます。

ベストプラクティスを実行するには、Amazon RDS から Amazon CloudWatch Logs にデータベースログを発行する必要があります。CloudWatch Logs を使用すると、ログデータのリアルタイム分析、耐久性に優れたストレージへのデータ保存、CloudWatch Logs エージェントによるデータ管理を行えます。データベースログは、Amazon RDS コンソールからアクセスして監視できます。CloudWatch Logs Insights を使用すると、CloudWatch Logs 内のログデータをインタラクティブに検索し、分析することも可能です。次の例は、監査ログのクエリを示しています。このクエリにより、ログに CONNECT イベントが出力されている回数、接続者、接続元クライアント (IP アドレス) を確認します。監査ログからの抜粋を次に示します。

```
20221201 14:07:05,ip-10-22-1-51,rdsadmin,localhost,821,0,CONNECT,,,0,SOCKET
20221201 14:07:05,ip-10-22-1-51,rdsadmin,localhost,821,0,DISCONNECT,,,0,SOCKET
20221201 14:12:20,ip-10-22-1-51,rdsadmin,localhost,822,0,CONNECT,,,0,SOCKET
20221201 14:12:20,ip-10-22-1-51,rdsadmin,localhost,822,0,DISCONNECT,,,0,SOCKET
20221201 14:17:35,ip-10-22-1-51,rdsadmin,localhost,823,0,CONNECT,,,0,SOCKET
20221201 14:17:35,ip-10-22-1-51,rdsadmin,localhost,823,0,DISCONNECT,,,0,SOCKET
20221201 14:22:50,ip-10-22-1-51,rdsadmin,localhost,824,0,CONNECT,,,0,SOCKET
20221201 14:22:50,ip-10-22-1-51,rdsadmin,localhost,824,0,DISCONNECT,,,0,SOCKET
```

Log Insights クエリの例は、rdsadmin が localhost から 5 分おきにデータベースに接続していることを示しており、次の図を見ると、それが合計 22 回発生していることがわかります。これらの結

果によると、このアクティビティは、モニタリングシステムといった内部の Amazon RDS プロセス自体によって生じています。



ログイベントには、MySQL および MariaDB DB インスタンス関連オペレーションで生じる警告やエラーといった、カウント対象となる重要なメッセージが頻繁に出力されます。例えば、操作が失敗した場合、次のエラーが発生し、エラーログファイルに記録される場合があります: `ERROR 1114 (HY000): The table zip_codes is full`。エラー傾向を把握するために、これらのログのモニタリングをお勧めします。Amazon RDS ログからカスタム CloudWatch メトリクスを作成するには、フィルターを使用して Amazon RDS データベースログの自動モニタリングを有効にします。次に、特定のログで特定のパターンをモニタリングし、想定動作に違反する挙動があった場合にアラームを生成するよう設定します。例えば、ロググループ `/aws/rds/instance/database-1/error` にメトリクスフィルターを作成し、これによって、エラーログをモニタリングし、`ERROR` のような特定のパターンを検索します。フィルターパターンを `ERROR` に、メトリクス値を `1` に設定します。このフィルターによって、キーワード `ERROR` が含まれるログレコードをすべて検出し「ERROR」のあるログイベントごとにカウントを 1 増やします。フィルターを作成したら、MySQL または MariaDB エラーログでエラーが検出された際にその旨を通知させるアラームを設定できます。

CloudWatch ダッシュボードの作成と CloudWatch Logs Insights の使用によるスロークエリログおよびエラーログモニタリングの詳細については、ブログ記事「Creating an Amazon CloudWatch dashboard to monitor Amazon RDS and Amazon Aurora MySQL」を参照してください。

監査証跡

監査証跡 (監査ログ) は、AWS アカウント で発生したイベントについて、セキュリティ関連の時系列レコードを得られます。Amazon RDS のイベントも対象になるため、データベースまたはクラウド環境に影響を与えた一連のアクティビティについて、ドキュメント化された証拠を収集できます。Amazon RDS for MySQL または MariaDB では、監査証跡を使用して次のオペレーションを行います。

- DB インスタンスの監査ログをモニタリングする
- AWS CloudTrail で Amazon RDS API コールをモニタリングする

Amazon RDS DB インスタンスが対象の場合、一般的に、次のような監査目的があります。

- 以下について説明責任を果たすこと。
  - パラメータまたはセキュリティ設定で行われた変更
  - データベーススキーマ、テーブル、行で実行されたアクション、あるいは、特定の内容に影響が及ぶアクション
- 侵入の検出および調査

- 疑わしいアクティビティの検出および調査
- 認可上の問題の検出。例: 正規ユーザーまたは特権ユーザーによるアクセス権の悪用を特定する

データベースの監査証跡は、次のようなよくある疑問の解消に有用です: どのユーザーがデータベース内の機密データを表示または変更したか? この事象はいつ発生したか? この特定のユーザーはどこからデータにアクセスしたか? この特権ユーザーは無制限のアクセス権を悪用したか?

MySQL と MariaDB ではともに、MariaDB 監査プラグインを使用して DB インスタンスの監査証跡機能を実装します。このプラグインにより、データベースへのユーザーログインや、データベースへのクエリ実行といった、データベースのアクティビティを記録します。データベースのアクティビティのレコードはログファイルに保存されます。監査ログにアクセスするには、DB インスタンスは `MARIADB_AUDIT_PLUGIN` オプションを指定してカスタムオプショングループを使用する必要があります。詳細については、Amazon RDS ドキュメントの「MySQL に対する MariaDB 監査プラグインのサポート」を参照してください。監査ログのレコードは、プラグインで定義されている特定の形式で保存されます。監査ログ形式の詳細については、MariaDB Server のドキュメントを参照してください。

AWS アカウントで AWS クラウド の監査証跡を得るには、AWS CloudTrail サービスを利用します。CloudTrail は、Amazon RDS の API コールをイベントとしてキャプチャします。Amazon RDS でのアクションは、すべてログに記録されます。CloudTrail では、ユーザー、ロール、その他の AWS サービスによって Amazon RDS で実行されたアクションを記録します。例えば、AWS マネジメントコンソール、AWS CLI、AWS SDK および API で実行されたアクションなどのイベントが記録対象となります。

例

一般的な監査シナリオでは、AWS CloudTrail の証跡、データベース監査ログ、Amazon RDS イベントモニタリングの結果を総合的に確認する必要があるでしょう。例えば、Amazon RDS DB インスタンス (`database-1` など) のデータベースパラメータが変更され、その変更者、変更された対象、変更されたタイミングを特定するタスクを任されるといったシナリオが考えられます。

そのタスクを実行するには、次のステップに従います。

1. データベースインスタンス `database-1` で発生した Amazon RDS イベントを一覧表示して、`configuration change` カテゴリに属するイベントのうち、`Finished updating DB parameter group` というメッセージを持つイベントがあるかどうかを確認します。

```
$ aws rds describe-events --source-identifier database-1 --source-type db-instance
```

```
{
    "Events": [
        {
            "SourceIdentifier": "database-1",
            "SourceType": "db-instance",
            "Message": "Finished updating DB parameter group",
            "EventCategories": [
                "configuration change"
            ],
            "Date": "2022-12-01T09:22:40.413000+00:00",
            "SourceArn": "arn:aws:rds:eu-west-3:111122223333:db:database-1"
        }
    ]
}
```

2. DB インスタンスで使用されている DB パラメータグループを特定します。

```
$ aws rds describe-db-instances --db-instance-identifier database-1 --query
 'DBInstances[*].[DBInstanceIdentifier,Engine,DBParameterGroups]'
[
    [
        "database-1",
        "mariadb",
        [
            {
                "DBParameterGroupName": "mariadb10-6-test",
                "ParameterApplyStatus": "pending-reboot"
            }
        ]
    ]
]
```

3. AWS CLI を使用して、CloudTrail イベントを検索します。具体的には、database-1 がデプロイされているリージョンで、ステップ 1 で検出した Amazon RDS イベントの発生時刻前後の期間を対象に EventName=ModifyDBParameterGroup のイベントを特定します。

```
$ aws cloudtrail --region eu-west-3 lookup-events --lookup-attributes
 AttributeKey=EventName,AttributeValue=ModifyDBParameterGroup --start-time
 "2022-12-01, 09:00 AM" --end-time "2022-12-01, 09:30 AM"

{
    "eventVersion": "1.08",
    "userIdentity": {
```

```
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Role1",
                "accountId": "111122223333",
                "userName": "User1"
            }
        }
    },
    "eventTime": "2022-12-01T09:18:19Z",
    "eventSource": "rds.amazonaws.com",
    "eventName": "ModifyDBParameterGroup",
    "awsRegion": "eu-west-3",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "parameters": [
            {
                "isModifiable": false,
                "applyMethod": "pending-reboot",
                "parameterName": "innodb_log_buffer_size",
                "parameterValue": "8388612"
            },
            {
                "isModifiable": false,
                "applyMethod": "pending-reboot",
                "parameterName": "innodb_write_io_threads",
                "parameterValue": "8"
            }
        ],
        "dBParameterGroupName": "mariadb10-6-test"
    },
    "responseElements": {
        "dBParameterGroupName": "mariadb10-6-test"
    },
    "requestID": "fdf19353-de72-4d3d-bf29-751f375b6378",
    "eventID": "0bba7484-0e46-4e71-93a8-bd01ca8386fe",
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
```

```
    "sessionCredentialFromConsole": "true"
}
```

この CloudTrail イベントによると、AWS アカウントが 111122223333 で、ロール Role1 を持つ User1 が、2022-12-01 at 09:18:19 h に DB インスタンス database-1 で使用されていた DB パラメータグループ mariadb10-6-test を変更していました。2 つのパラメータが変更され、値が以下に設定されています。

- innodb_log_buffer_size = 8388612
- innodb_write_io_threads = 8

CloudTrail と CloudWatch Logs が備えるその他の機能

過去 90 日間に発生した運用上およびセキュリティ上のインシデントにトラブルシューティングを行うには、CloudTrail コンソールで [イベント履歴] を表示します。その保持期間を延長して、追加のクエリ機能を利用する場合は、AWS CloudTrail Lake を使用できます。AWS CloudTrail Lake では、イベントデータを、最大 7 年間、イベントデータストアに保存できます。さらに、このサービスは、複雑な SQL クエリに対応しています。カスタマイズ可能で詳細なイベントビューを得られるという点で、[イベント履歴] での単純なキーと値のルックアップよりも優れています。

監査証跡をモニタリングして、アラームを設定し、特定のアクティビティが発生したときに通知を受け取るには、CloudTrail を設定して、その証跡レコードが CloudWatch Logs に送信されるようにする必要があります。証跡レコードが CloudWatch Logs として保存されたら、メトリクスフィルターを定義して、用語、フレーズ、または値に一致するログイベントを評価し、メトリクスフィルターにメトリクスを割り当てることができます。さらに、指定したしきい値と期間に基づいてアラームを生成する CloudWatch アラームを作成することも可能です。例えば、担当チームに通知を送信するアラームを設定すると、チームでは適切なアクションを実行できます。アラームへの対応アクションが自動的に実行されるように CloudWatch を設定することもできます。

[アラート]

アラートは、IT インフラストラクチャと IT サービスを運用する上で、セキュリティ、可用性、パフォーマンス、信頼性に関する最も重要な情報源の 1 つです。IT チームでは、アラートの通知によって、進行中のセキュリティの脅威、停止、パフォーマンスの問題、システム障害などの情報を得ています。

Information Technology Infrastructure Library (ITIL) のうち、とりわけ IT サービス管理 (ITSM) のプラクティスでは、モニタリング、イベント管理、インシデント管理のベストプラクティスで注視すべき点に自動アラートを設定します。

インシデントアラートとは、モニタリングツールでアラートを生成し、IT 環境の変更、高リスクのアクション、障害などの情報をチームと自動ツール (自動実行可能な項目の場合) に通知します。IT アラートは、重大なインシデントにつながりかねないシステム停止や変更を防御するための最前線として機能します。システムを自動的にモニタリングし、停止やリスクのある変更に関するアラートを生成することで、ダウンタイムを最小限に抑え、それに伴う高コストを削減できます。

Well-Architected AWS フレームワークでは、ベストプラクティスとして、モニタリングを使用してアラームベースの通知を生成し、プロアクティブにモニタリングおよびアラームを行うことを規定しています。CloudWatch やサードパーティー製のモニタリングサービスを利用して、メトリクスが想定範囲外となった旨が通知されるアラームを設定すると良いでしょう。

アラート管理の目的は、ログ記録、分類、アクション定義と実装、クローズ、インシデント後のレビューアクティビティなどにより、IT 関連のイベントとインシデントを処理する効率的で標準化された手順を確立することです。

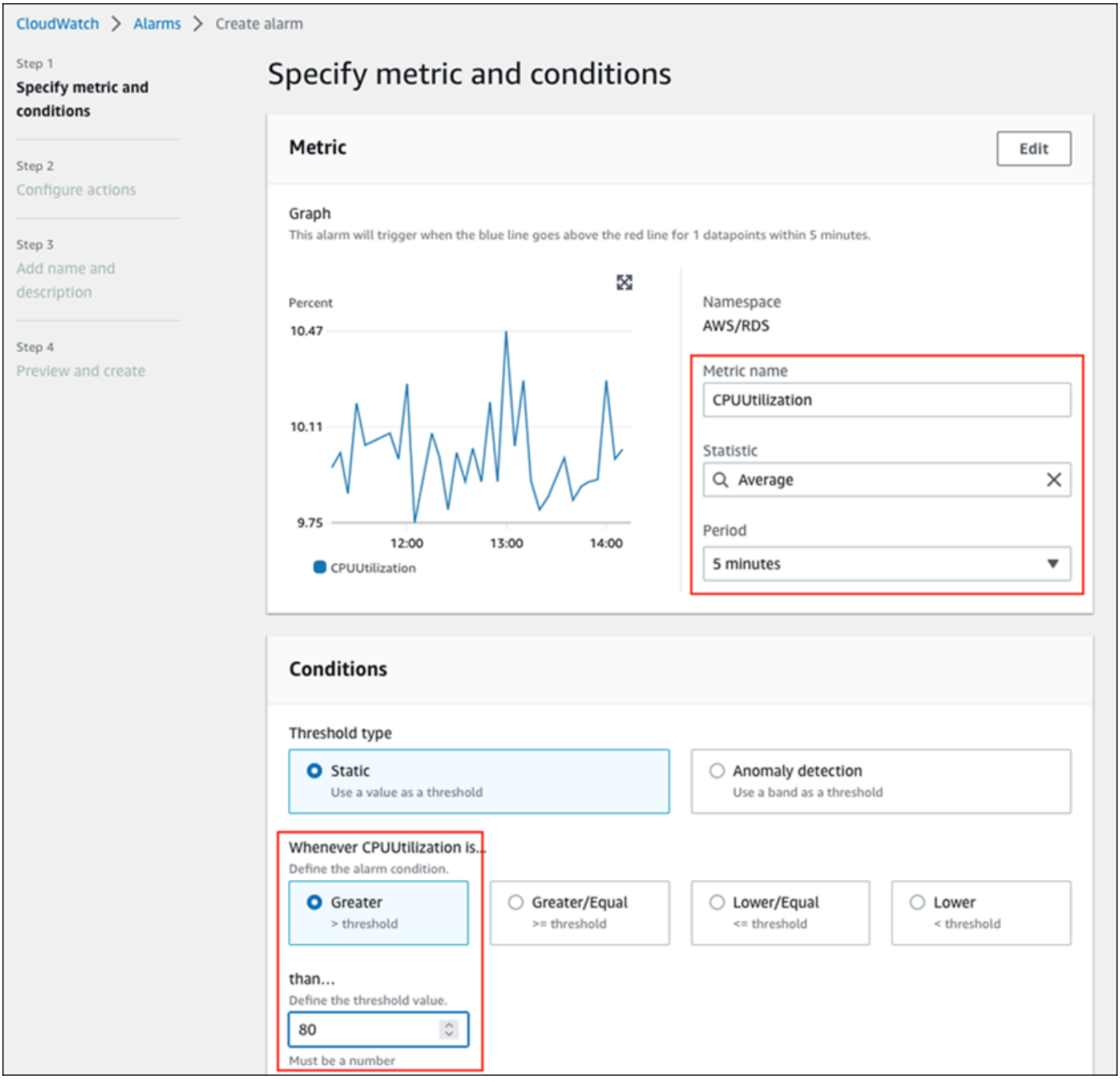
セクション

- CloudWatch アラーム
- EventBridge ルール
- アクションの指定、アラームの有効化と無効化

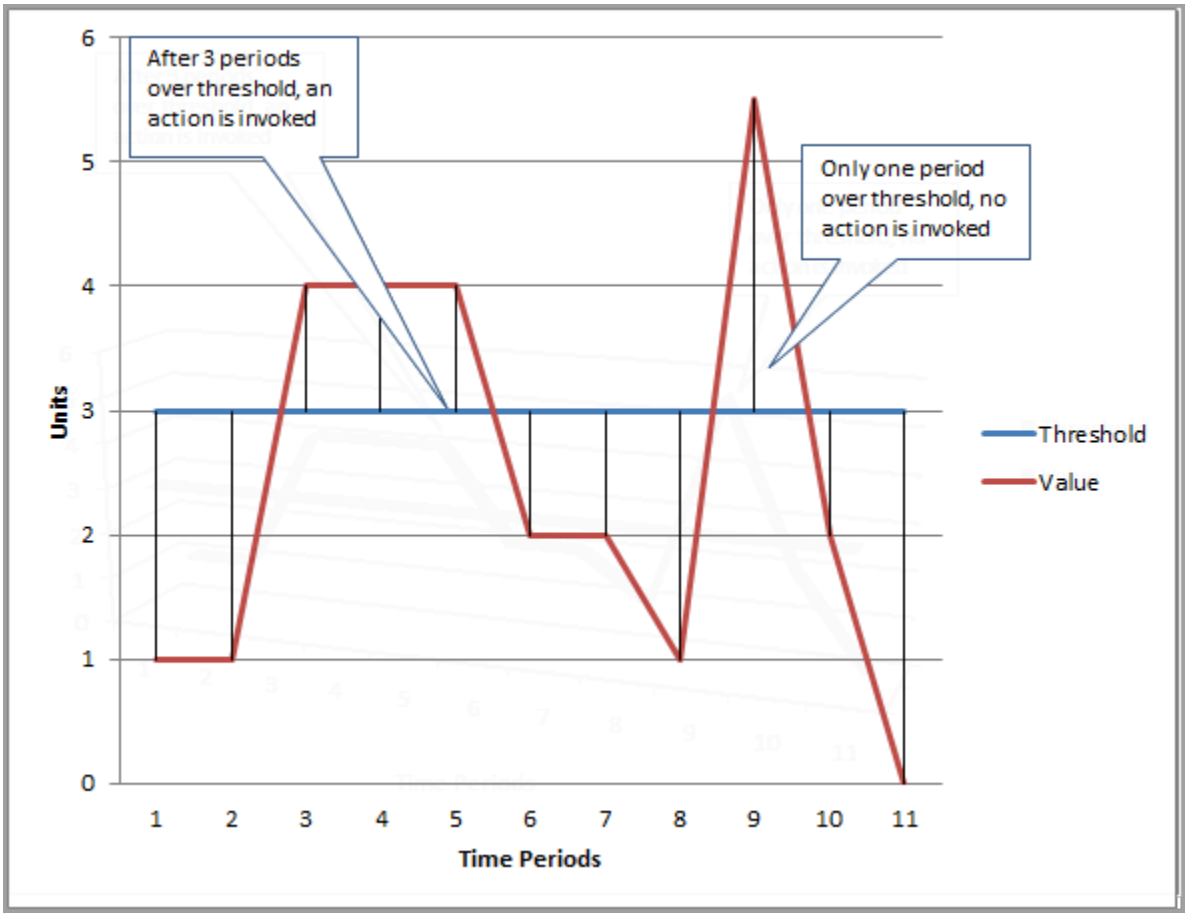
CloudWatch アラーム

Amazon RDS DB インスタンスの操作では、さまざまな種類のメトリクス、イベント、トレースをモニタリングしてアラートを生成します。MySQL および MariaDB データベースの場合、DB インスタンスメトリクス、OS メトリクス、イベント、ログ、監査証跡が重要な情報源となります。指定した期間全体で 1 つのメトリクスを監視するには、CloudWatch アラームを使用すると良いでしょう。

次の例は、すべての Amazon RDS DB インスタンスで CPUUtilization メトリクス (CPU 使用率) を監視するアラームの設定方法を示しています。DB インスタンスの CPU 使用率が 5 分間の評価時間中に 80% を超えた場合にアラームがトリガーされるように設定します。



つまり、いずれかのデータベースで 5 分以上 CPU 使用率が高い (80% を超える) 状態が続く場合、アラームは ALARM の状態になります。CPU が 80% を超える使用率まで短時間バーストし、再びしきい値を下回った場合、アラームは OK の状態が維持されます。そのロジックを以下のグラフに示します。



CloudWatch アラームでは、メトリクスアラームと複合アラームがサポートされています。

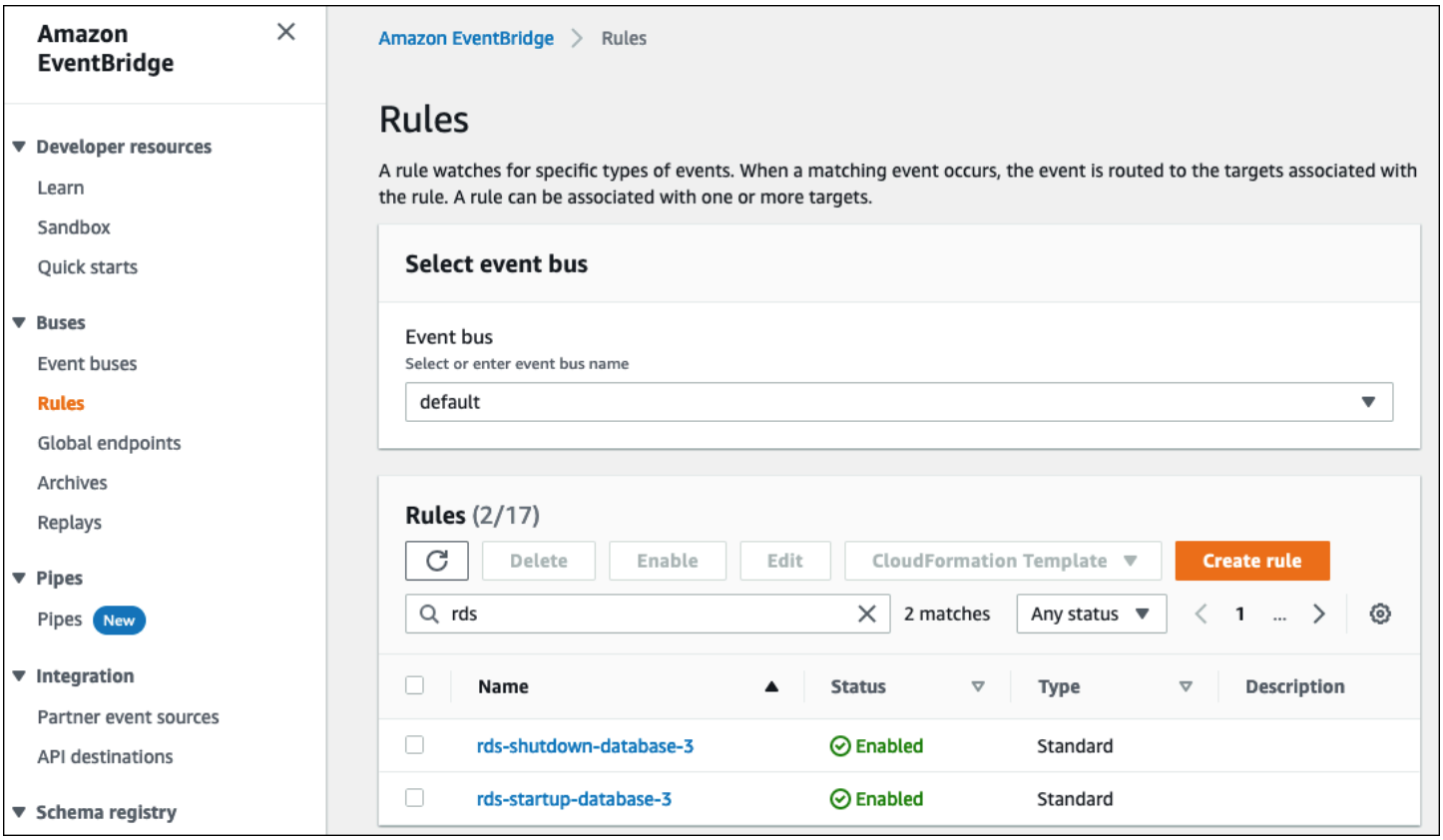
- メトリクスアラーム: 単一の CloudWatch メトリクスを監視し、メトリクスで数式を実行できます。また、Amazon SNS メッセージを送信することで、複数の期間にわたって特定のしきい値のメトリクス値に基づいて 1 つ以上のアクションを実行できます。
- 複合アラーム: 複数のアラームの状態を評価し、ルールのすべての条件が満たされた場合にのみ ALARM 状態になるルール式に従って動作します。一般的、複合アラームは、不要なアラートの数を減らすために使用します。例えば、複合アラームに、アクションを実行しないように設定した複数のメトリクスアラームが含まれているとします。この場合、複合アラームに含まれる各メトリクスアラームがすべて ALARM の状態になると、アラートが送信されます。

CloudWatch アラームで監視できるのは、CloudWatch のメトリクスのみです。エラー、スロークエリ、一般的なログに基づいてアラームを作成する場合は、それらのログから CloudWatch メトリクスを作成する必要があります。これを行うには、「OS モニタリング」セクションと「イベント」セクションで説明したように、フィルターを使用してログイベントからメトリクスを作成します。同様

に、拡張モニタリングメトリクスに基づいてアラートを生成するには、CloudWatch Logs のデータから CloudWatch でメトリクスフィルターを作成する必要があります。

EventBridge ルール

Amazon RDS のイベントは Amazon EventBridge に配信されますが、それらのイベントには、EventBridge ルールを使用して対応できます。例えば、1 つの特定の DB インスタンスが停止または起動した場合に通知し、アクションを実行する EventBridge ルールを作成できます。その例を次の図に示します。

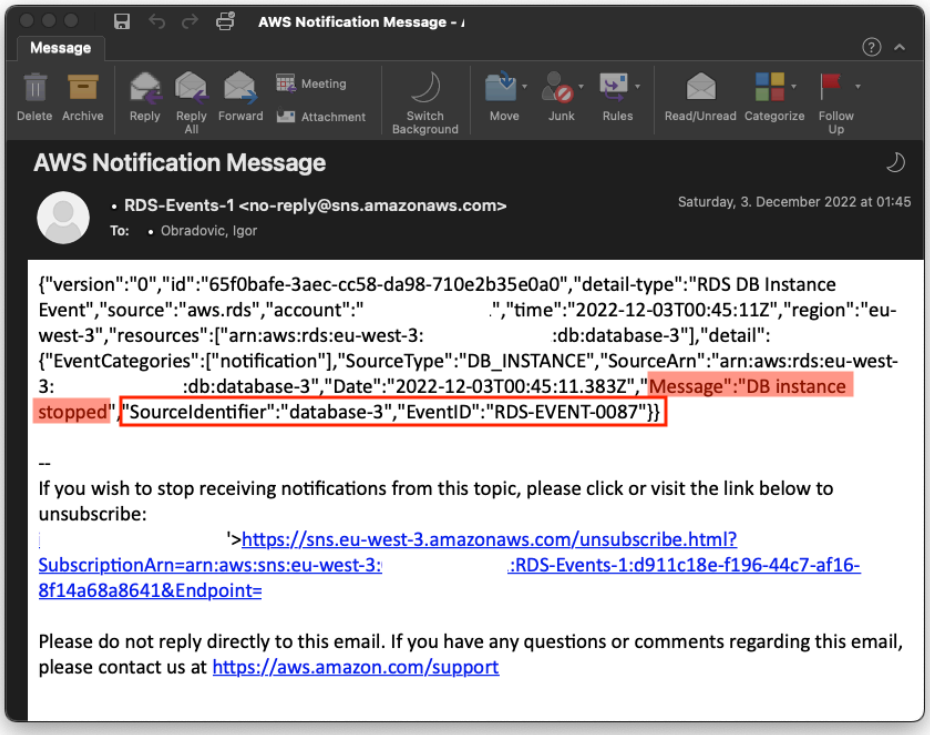


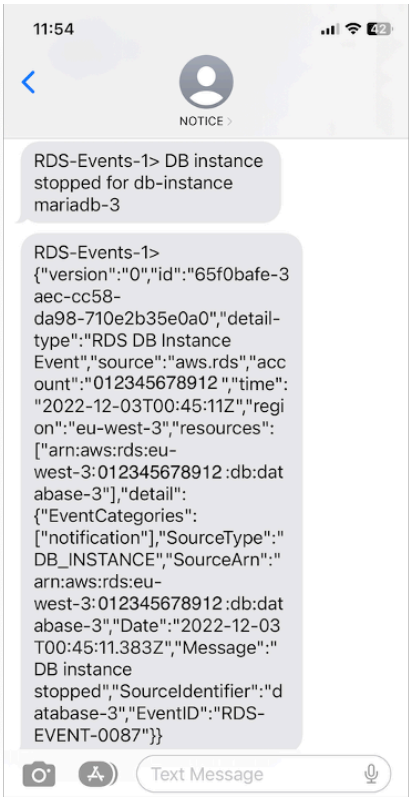
The DB instance has been stopped のイベントを検出するルールには Amazon RDS イベント ID (RDS-EVENT-0087) があるため、そのルールの Event Pattern プロパティは、次のように設定します。

```
{
  "source": ["aws.rds"],
  "detail-type": ["RDS DB Instance Event"],
  "detail": {
    "SourceArn": ["arn:aws:rds:eu-west-3:111122223333:db:database-3"],
    "EventID": ["RDS-EVENT-0087"]
```

```
  }
}
```

このルールでは DB インスタンス database-3 のみをモニタリングして、RDS-EVENT-0087 のイベントを監視します。EventBridge でイベントが検出されると、ターゲットであるリソースまたはエンドポイントにイベントが送信されます。このルールでは、Amazon RDS インスタンスがシャットダウンした場合に実行するアクションを指定できます。SNS トピック、Amazon Simple Queue Service (Amazon SQS) キュー、 AWS Lambda 関数、 AWS Systems Manager Automation、 AWS Batch ジョブ、Amazon API Gateway など、考えられる多くのターゲットにイベントを送信できます。例えば、通知 E メールと SMS を送信する SNS トピックを作成して、その SNS トピックを EventBridge ルールのターゲットとして割り当てることができます。Amazon RDS DB インスタンス database-3 が停止した場合、Amazon RDS から EventBridge にイベント RDS-EVENT-0087 が配信され、検出されます。その後、EventBridge では、ターゲット (指定した SNS トピック) が呼び出されます。SNS トピックは、E メール (次の図を参照) と SMS を送信するように設定しておきます。





アクションの指定、アラームの有効化と無効化

CloudWatch アラームを使用すると、OK、ALARM、INSUFFICIENT_DATA の各状態にアラームが変化したときに実行するアクションを指定できます。CloudWatch には、SNS トピックとの連携機能

が組み込まれており、Amazon Elastic Compute Cloud (Amazon EC2) のアクションや Amazon EC2 Auto Scaling グループのアクションといった、Amazon RDS メトリクスには適用されないアクションカテゴリが複数追加されています。EventBridge の一般的な用途は、ルールを記述して、Amazon RDS メトリクスのアラームがトリガーされたときにアクションを実行するターゲットを定義することです。CloudWatch アラームの状態が変化するたびに、CloudWatch から EventBridge にイベントが送信されます。これらのアラーム状態変更イベントを使用して、EventBridge でイベントターゲットをトリガーできます。詳細については、CloudWatch ドキュメントの「アラームイベントと EventBridge」を参照してください。

アラームの管理が必要となる場合もあります。例えば、計画した設定変更またはテスト中にアラームを自動的に無効にし、計画したアクションが終了したらアラームを再度有効にするなどです。例を挙げましょう。ダウンタイムが必要な計画済みデータベースソフトウェアアップグレードを予定しており、データベースが使用できなくなった場合に有効化されるアラームがある場合は、API アクション DisableAlarmActions と EnableAlarmActions、あるいは AWS CLIの disable-alarm-actions と enable-alarm-actions コマンドによってアラームを無効化または有効化できます。アラーム履歴は、CloudWatch コンソール、DescribeAlarmHistory API アクション、または AWS CLIの describe-alarm-history コマンドを使用して確認することもできます。CloudWatch は、アラーム履歴を 2 週間保存します。CloudWatch コンソールでは、ナビゲーションペインの [お気に入りと最近のアクセス] メニューを選択して、お気に入りのアラームまたは最近使用したしたアラームへのアクセスと、それらの設定を行えます。

次のステップとリソース

リレーショナルデータベースの AWS クラウド への移行方法については、「AWS 規範ガイダンス」ウェブサイトに記載の以下の戦略で詳しくご確認いただけます。

- リレーショナルデータベースの移行戦略

データベース移行パターンについては、「AWS 規範ガイダンス」で詳しく解説しています。ここでは、AWS クラウド で稼働する貴社固有のリレーショナルデータベースに関するステップバイステップの手順を、モニタリング、移行、データ管理の関連タスクも含めご確認いただけます。

その他のリソースについては、次の内容を参照してください。

- Amazon Relational Database Service ユーザーガイド
- Amazon CloudWatch ユーザーガイド
- Amazon RDS よくある質問
- Performance Insights のよくある質問
- Deliver Amazon RDS Performance Insights counter metrics to a third-party Application Performance Monitoring service provider using Amazon CloudWatch Metrics Stream (AWS ブログ記事)
- Creating an Amazon CloudWatch dashboard to monitor Amazon RDS and Amazon Aurora MySQL (AWS ブログ記事)
- Tuning Amazon RDS for MySQL with Performance Insights (AWS ブログ記事)

ドキュメント履歴

以下の表は、本ガイドの重要な変更点について説明したものです。今後の更新に関する通知を受け取る場合は、RSS フィード をサブスクライブできます。

変更	説明	日付
Performance Insights に関する情報を更新しました	Performance Insights メトリクスの CloudWatch への公開に関するセクションを最新の情報で更新しました。	2025 年 3 月 11 日
エクスポーターに関する情報を更新しました	エクスポーターに関する情報を更新し、エクスポーターを選択するためのガイドラインを追加しました。	2024 年 6 月 13 日
初版発行	—	2023 年 6 月 30 日

AWS 規範ガイダンスの用語集

以下は、 AWS 規範ガイダンスによって提供される戦略、ガイド、パターンで一般的に使用される用語です。エントリを提案するには、用語集の最後のフィードバックの提供リンクを使用します。

数字

7 Rs

アプリケーションをクラウドに移行するための 7 つの一般的な移行戦略。これらの戦略は、ガートナーが 2011 年に特定した 5 Rs に基づいて構築され、以下で構成されています。

- リファクタリング/アーキテクチャの再設計 — クラウドネイティブ特徴を最大限に活用して、俊敏性、パフォーマンス、スケーラビリティを向上させ、アプリケーションを移動させ、アーキテクチャを変更します。これには、通常、オペレーティングシステムとデータベースの移植が含まれます。例: オンプレミスの Oracle データベースを Amazon Aurora PostgreSQL 互換エディションに移行する。
- リプラットフォーム (リフトアンドリシェイプ) — アプリケーションをクラウドに移行し、クラウド機能を活用するための最適化レベルを導入します。例: お客様のオンプレミスの Oracle データベースを AWS クラウドの Oracle 用の Amazon Relational Database Service (Amazon RDS) に移行する。
- 再購入 (ドロップアンドショップ) — 通常、従来のライセンスから SaaS モデルに移行して、別の製品に切り替えます。例: 顧客関係管理 (CRM) システムを Salesforce.com に移行する。
- リホスト (リフトアンドシフト) — クラウド機能を活用するための変更を加えずに、アプリケーションをクラウドに移行します。例: お客様のオンプレミスの Oracle データベースを AWS クラウドの EC2 インスタンス上の Oracle に移行する。
- 再配置 (ハイパーバイザーレベルのリフトアンドシフト) – 新しいハードウェアを購入したり、アプリケーションを書き換えたり、既存の運用を変更したりすることなく、インフラストラクチャをクラウドに移行できます。オンプレミスプラットフォームから同じプラットフォームのクラウドサービスにサーバーを移行します。例： Microsoft Hyper-Vアプリケーションを に移行します AWS。
- 保持 (再アクセス) — アプリケーションをお客様のソース環境で保持します。これには、主要なリファクタリングを必要とするアプリケーションや、お客様がその作業を後日まで延期したいアプリケーション、およびそれらを移行するためのビジネス上の正当性がないため、お客様が保持するレガシーアプリケーションなどがあります。
- 廃止 — お客様のソース環境で不要になったアプリケーションを停止または削除します。

A

ABAC

「属性ベースのアクセス制御」をご覧ください。

抽象化されたサービス

「マネージドユーザー」をご覧ください。

ACID

「原子性、一貫性、分離性、耐久性 (ACID)」をご覧ください。

アクティブ/アクティブ移行

(双方向レプリケーションツールまたは二重書き込み操作を使用して) ソースデータベースとターゲットデータベースを同期させ、移行中に両方のデータベースが接続アプリケーションからのトランザクションを処理するデータベース移行方法。この方法では、1 回限りのカットオーバーの必要がなく、管理された小規模なバッチで移行できます。アクティブ/パッシブ移行よりも柔軟な方法ですが、さらに多くの作業が必要となります。

アクティブ/パッシブ移行

ソースデータベースとターゲットデータベースを同期させながら、データがターゲットデータベースにレプリケートされている間、接続しているアプリケーションからのトランザクションをソースデータベースのみで処理するデータベース移行方法。移行中、ターゲットデータベースはトランザクションを受け付けません。

集計関数

複数行に処理を行い、グループ全体を対象に単一の戻り値を計算する SQL 関数。集計関数の例としては、SUM や MAX などがあります。

AI

「人工知能」をご覧ください。

AIOps

「AI オペレーション」をご覧ください。

匿名化

データセット内の個人情報を完全に削除するプロセス。匿名化は個人のプライバシー保護に役立ちます。匿名化されたデータは、もはや個人データとは見なされません。

アンチパターン

繰り返し起こる問題に対して頻繁に用いられる解決策で、その解決策が逆効果であったり、効果がなかったり、代替案よりも効果が低かったりするもの。

アプリケーション制御

マルウェアからシステムを保護するために、承認されたアプリケーションのみを使用できるようにするセキュリティアプローチ。

アプリケーションポートフォリオ

アプリケーションの構築と維持にかかるコスト、およびそのビジネス価値を含む、組織が使用する各アプリケーションに関する詳細情報の集まり。この情報は、ポートフォリオの検出と分析プロセス の重要な要素であり、移行、モダナイズ、最適化するアプリケーションを特定し、優先順位を付けるのに役立ちます。

人工知能 (AI)

コンピューティングテクノロジーを使用し、学習、問題の解決、パターンの認識など、通常は人間に関連づけられる認知機能の実行に特化したコンピュータサイエンスの分野。詳細については、「人工知能 (AI) とは何ですか?」をご覧ください。

AI オペレーション (AIOps)

機械学習技術を使用して運用上の問題を解決し、運用上のインシデントと人の介入を減らし、サービス品質を向上させるプロセス。 AWS 移行戦略での AIOps の使用方法については、オペレーション統合ガイド を参照してください。

非対称暗号化

暗号化用のパブリックキーと復号用のプライベートキーから成る 1 組のキーを使用した、暗号化のアルゴリズム。パブリックキーは復号には使用されないため共有しても問題ありませんが、プライベートキーの利用は厳しく制限する必要があります。

原子性、一貫性、分離性、耐久性 (ACID)

エラー、停電、その他の問題が発生した場合でも、データベースのデータ有効性と運用上の信頼性を保証する一連のソフトウェアプロパティ。

属性ベースのアクセス制御 (ABAC)

部署、役職、チーム名など、ユーザーの属性に基づいてアクセス許可をきめ細かく設定する方法。詳細については、 AWS Identity and Access Management (IAM) ドキュメントの「 の ABAC AWS」を参照してください。

信頼できるデータソース

最も信頼性のある情報源とされるデータのプライマリーバージョンを保存する場所。匿名化、編集、仮名化など、データを処理または変更する目的で、信頼できるデータソースから他の場所にデータをコピーすることができます。

アベイラビリティーゾーン (AZ)

他のアベイラビリティーゾーンの障害から AWS リージョン 隔離され、同じリージョン内の他のアベイラビリティーゾーンへの低コストで低レイテンシーのネットワーク接続を提供する 内の別の場所。

AWS クラウド導入フレームワーク (AWS CAF)

組織がクラウドへの移行を成功させるための効率的で効果的な計画を立て AWS るための、 のガイドラインとベストプラクティスのフレームワークです。 AWS CAF は、ビジネス、人材、ガバナンス、プラットフォーム、セキュリティ、運用という 6 つの重点分野にガイダンスを整理しています。ビジネス、人材、ガバナンスの観点では、ビジネススキルとプロセスに重点を置き、プラットフォーム、セキュリティ、オペレーションの視点は技術的なスキルとプロセスに焦点を当てています。例えば、人材の観点では、人事 (HR)、人材派遣機能、および人材管理を扱うステークホルダーを対象としています。この観点から、 AWS CAF は、クラウド導入を成功させるための組織の準備に役立つ人材開発、トレーニング、コミュニケーションに関するガイダンスを提供します。詳細については、AWS CAF ウェブサイトと AWS CAF のホワイトペーパー を参照してください。

AWS ワークロード認定フレームワーク (AWS WQF)

データベース移行ワークロードを評価し、移行戦略を推奨し、作業見積もりを提供するツール。AWS WQF は AWS Schema Conversion Tool (AWS SCT) に含まれています。データベーススキーマとコードオブジェクト、アプリケーションコード、依存関係、およびパフォーマンス特性を分析し、評価レポートを提供します。

B

不正なボット

個人や組織に混乱や損害を与えることを目的としたボット。

BCP

「ビジネス継続性計画 (BCP)」をご覧ください。

動作グラフ

リソースの動作とインタラクションを経時的に示した、一元的なインタラクティブビュー。Amazon Detective の動作グラフを使用すると、失敗したログオンの試行、不審な API 呼び出し、その他同様のアクションを調べることができます。詳細については、Detective ドキュメントの「動作グラフのデータ」を参照してください。

ビッグエンディアンシステム

最上位バイトを最初に格納するシステム。「エンディアン性」もご覧ください。

二項分類

バイナリ結果 (2 つの可能なクラスのうちの 1 つ) を予測するプロセス。例えば、お客様の機械学習モデルで「この E メールはスパムですか、それともスパムではありませんか」などの問題を予測する必要があるかもしれません。または「この製品は書籍ですか、車ですか」などの問題を予測する必要があるかもしれません。

ブルームフィルター

要素がセットのメンバーであるかどうかをテストするために使用される、確率的でメモリ効率の高いデータ構造。

ブルー/グリーンデプロイ

それぞれが独立しているが、同一の環境を 2 つ作成するデプロイ戦略。現在のアプリケーションバージョンを 1 つの環境 (ブルー) で実行し、新しいアプリケーションバージョンを別の環境 (グリーン) で実行します。この戦略は、最小限の影響で迅速にロールバックするのに役立ちます。

ボット

インターネット経由で自動タスクを実行し、人間のアクティビティややり取りをシミュレートするソフトウェアアプリケーション。インターネット上の情報のインデックスを作成するウェブクローラーなど、一部のボットは有用または有益です。悪質なボットと呼ばれる他のボットの中には、個人や組織を混乱させたり、損害を与えたりすることを意図したものもあります。

ボットネット

マルウェアに感染しており、ボットハーダーまたはボットオペレーターと呼ばれる単一の当事者によって制御されているボットのネットワーク。ボットネットは、ボットとその影響力を拡大する仕組みとして、非常によく知られています。

ブランチ

コードリポジトリに含まれる領域。リポジトリに最初に作成するブランチは、メインブランチといいます。既存のブランチから新しいブランチを作成し、その新しいブランチで機能を開発した

り、バグを修正したりできます。機能を構築するために作成するブランチは、通常、機能ブランチと呼ばれます。機能をリリースする準備ができたら、機能ブランチをメインブランチに統合します。詳細については、「ブランチの概要」(GitHub ドキュメント) を参照してください。

ブレークグラスアクセス

例外的な状況では、承認されたプロセスを通じて、ユーザーが AWS アカウント 通常アクセス許可を持たない にすばやくアクセスできるようにします。詳細については、 AWS Well-Architected ガイダンスの「ブレークグラス手順の実装」インジケータを参照してください。

ブラウンフィールド戦略

環境の既存インフラストラクチャ。システムアーキテクチャにブラウンフィールド戦略を導入する場合、現在のシステムとインフラストラクチャの制約に基づいてアーキテクチャを設計します。既存のインフラストラクチャを拡張している場合は、ブラウンフィールド戦略とグリーンフィールド戦略を融合させることもできます。

バッファキャッシュ

アクセス頻度が最も高いデータが保存されるメモリ領域。

ビジネス能力

価値を生み出すためにビジネスが行うこと (営業、カスタマーサービス、マーケティングなど)。マイクロサービスのアーキテクチャと開発の決定は、ビジネス能力によって推進できます。詳細については、AWSでのコンテナ化されたマイクロサービスの実行ホワイトペーパーの「ビジネス機能を中心に組織化」セクションを参照してください。

ビジネス継続性計画 (BCP)

大規模移行など、中断を伴うイベントが運用に与える潜在的な影響に対処し、ビジネスを迅速に再開できるようにする計画。

C

CAF

「AWS クラウド導入フレームワーク」を参照してください

カナリアデプロイ

エンドユーザーへのバージョンリリースを、時間をかけて段階的に行うこと。確信が持てたら新規バージョンをデプロイして、現在のバージョン全体を置き換えます。

CCoE

「Cloud Center of Excellence」を参照してください。

CDC

「変更データキャプチャ」を参照してください。

変更データキャプチャ (CDC)

データソース (データベーステーブルなど) の変更を追跡し、その変更に関するメタデータを記録するプロセス。CDC は、ターゲットシステムでの変更を監査またはレプリケートして同期を維持するなど、さまざまな目的に使用できます。

カオスエンジニアリング

障害や破壊的なイベントを意図的に導入して、システムの耐障害性をテストすること。AWS Fault Injection Service (AWS FIS) を使用して、 AWS ワークロードにストレスを与え、その応答を評価する実験を実行できます。

CI/CD

「継続的インテグレーションと継続的デリバリー」を参照してください。

分類

予測を生成するのに役立つ分類プロセス。分類問題の機械学習モデルは、離散値を予測します。離散値は、常に互いに区別されます。例えば、モデルがイメージ内に車があるかどうかを評価する必要がある場合があります。

クライアント側の暗号化

ターゲットがデータ AWS のサービス を受信する前のローカルでのデータの暗号化。

Cloud Center of Excellence (CCoE)

クラウドのベストプラクティスの作成、リソースの移動、移行のタイムラインの確立、大規模変革を通じて組織をリードするなど、組織全体のクラウド導入の取り組みを推進する学際的なチーム。詳細については、 AWS クラウド エンタープライズ戦略ブログの CCoE 投稿を参照してください。

クラウドコンピューティング

リモートデータストレージと IoT デバイス管理に通常使用されるクラウドテクノロジー。クラウドコンピューティングは、一般的に、エッジコンピューティングに接続されています。

クラウド運用モデル

IT 組織において、1 つ以上のクラウド環境を構築、成熟、最適化するために使用される運用モデル。詳細については、「クラウド運用モデルの構築」を参照してください。

導入のクラウドステージ

組織が、 AWS クラウドへの移行時に通常実行する 4 つの段階。

- プロジェクト — 概念実証と学習を目的として、クラウド関連のプロジェクトをいくつか実行する
- 基礎固め — お客様のクラウドの導入を拡大するための基礎的な投資 (ランディングゾーンの作成、CCoE の定義、運用モデルの確立など)
- 移行 — 個々のアプリケーションの移行
- 再発明 — 製品とサービスの最適化、クラウドでのイノベーション

これらのステージは、 AWS クラウド エンタープライズ戦略ブログのブログ記事「クラウドファーストへのジャーニー」と「導入のステージ」で Stephen Orban によって定義されました。移行戦略との関連性については、 AWS 「移行準備ガイド」を参照してください。

CMDB

「構成管理データベース (CMDB)」を参照してください。

コードリポジトリ

ソースコードやその他の資産 (ドキュメント、サンプル、スクリプトなど) が保存され、バージョン管理プロセスを通じて更新される場所。一般的なクラウドリポジトリには、GitHub や Bitbucket Cloud があります。コードの各バージョンはブランチと呼ばれます。マイクロサービスの構造では、各リポジトリは 1 つの機能専用です。1 つの CI/CD パイプラインで複数のリポジトリを使用できます。

コールドキャッシュ

空である、または、かなり空きがある、もしくは、古いデータや無関係なデータが含まれているバッファキャッシュ。データベースインスタンスはメインメモリまたはディスクから読み取る必要があり、バッファキャッシュから読み取るよりも時間がかかるため、パフォーマンスに影響します。

コールドデータ

めったにアクセスされず、通常は過去のデータです。この種類のデータをクエリする場合、通常は低速なクエリでも問題ありません。このデータを低パフォーマンスで安価なストレージ階層またはクラスに移動すると、コストを削減することができます。

コンピュータビジョン (CV)

機械学習を使用してデジタルイメージやビデオといった、ビジュアル形式の情報を分析および抽出する AI の分野。例えば、Amazon SageMaker AI では、CV 用の画像処理アルゴリズムを利用できます。

設定ドリフト

ワークロードにおいて、設定が想定した状態から変化すること。これによって、ワークロードが非準拠になる可能性がありますが、この状態は、徐々に生じ、意図的なものではありません。

構成管理データベース (CMDB)

データベースとその IT 環境 (ハードウェアとソフトウェアの両方のコンポーネントとその設定を含む) に関する情報を保存、管理するリポジトリ。通常、CMDB のデータは、移行のポートフォリオの検出と分析の段階で使用します。

コンフォーマンスパック

コンプライアンスチェックとセキュリティチェックをカスタマイズするためにアセンブルできる AWS Config ルールと修復アクションのコレクション。YAML テンプレートを使用して、コンフォーマンスパックを AWS アカウント および リージョンの単一のエンティティとしてデプロイすることも、組織全体にデプロイすることもできます。詳細については、 AWS Config ドキュメントの「コンフォーマンスパック」を参照してください。

継続的インテグレーションと継続的デリバリー (CI/CD)

ソフトウェアリリースプロセスのソース、ビルド、テスト、ステージング、本番の各ステージを自動化するプロセス。CI/CD は一般的にパイプラインと呼ばれます。プロセスの自動化、生産性の向上、コード品質の向上、配信の加速化を可能にします。詳細については、「継続的デリバリーの利点」を参照してください。CD は継続的デプロイ (Continuous Deployment) の略語でもあります。詳細については「継続的デリバリーと継続的なデプロイ」を参照してください。

CV

「コンピュータビジョン」を参照してください。

D

保管中のデータ

ストレージ内にあるデータなど、常に自社のネットワーク内にあるデータ。

データ分類

ネットワーク内のデータを重要度と機密性に基づいて識別、分類するプロセス。データに適した保護および保持のコントロールを判断する際に役立つため、あらゆるサイバーセキュリティのリスク管理戦略において重要な要素です。データ分類は、 AWS Well-Architected フレームワークのセキュリティの柱のコンポーネントです。詳細については、「データ分類」を参照してください。

データドリフト

実稼働データと ML モデルのトレーニングに使用されたデータとの間に有意な差異が生じたり、入力データが時間の経過と共に有意に変化したりすることです。データドリフトは、ML モデル予測の全体的な品質、精度、公平性を低下させる可能性があります。

転送中のデータ

ネットワーク内 (ネットワークリソース間など) を活発に移動するデータ。

データメッシュ

非一元的で分散型のデータ所有権を持つとともに、一元的な管理およびガバナンスを行えるアーキテクチャフレームワーク。

データ最小化

厳密に必要なデータのみを収集し、処理するという原則。でデータ最小化を実践 AWS クラウドすることで、プライバシーリスク、コスト、分析のカーボンフットプリントを削減できます。

データ境界

AWS 環境内の一連の予防ガードレール。信頼された ID のみが、期待されるネットワークから信頼されたリソースにアクセスできるようにします。詳細については、「 でのデータ境界の構築 AWS」を参照してください。

データの前処理

raw データをお客様の機械学習モデルで簡単に解析できる形式に変換すること。データの前処理とは、特定の列または行を削除して、欠落している、矛盾している、または重複する値に対処することを意味します。

データ出所

データの生成、送信、保存の方法など、データのライフサイクル全体を通じてデータの出所と履歴を追跡するプロセス。

データ件名

データを収集、処理している個人。

データウェアハウス

分析などのビジネスインテリジェンスをサポートするデータ管理システム。データウェアハウスには、一般的に、大量の履歴データが含まれており、多くの場合、それらはクエリや分析に使用されます。

データベース定義言語 (DDL)

データベース内のテーブルやオブジェクトの構造を作成または変更するためのステートメントまたはコマンド。

データベース操作言語 (DML)

データベース内の情報を変更 (挿入、更新、削除) するためのステートメントまたはコマンド。

DDL

「データベース定義言語」を参照してください。

ディープアンサンブル

予測のために複数の深層学習モデルを組み合わせます。ディープアンサンブルを使用して、より正確な予測を取得したり、予測の不確実性を推定したりできます。

深層学習

人工ニューラルネットワークの複数層を使用して、入力データと対象のターゲット変数の間のマッピングを識別する機械学習サブフィールド。

多層防御

一連のセキュリティメカニズムとコントロールをコンピュータネットワーク全体に層状に重ねて、ネットワークとその内部にあるデータの機密性、整合性、可用性を保護する情報セキュリティの手法。この戦略を に採用するときは AWS、リソースの保護に役立つように、 AWS Organizations 構造の異なるレイヤーに複数のコントロールを追加します。たとえば、多層防御アプローチでは、多要素認証、ネットワークセグメンテーション、暗号化を組み合わせることができます。

委任管理者

では AWS Organizations、互換性のあるサービスが AWS メンバーアカウントを登録して組織のアカウントを管理し、そのサービスのアクセス許可を管理できます。このアカウントを、そのサービスの委任管理者と呼びます。詳細、および互換性のあるサービスの一覧は、 AWS

Organizations ドキュメントの「AWS Organizationsで使用できるサービス」を参照してください。

トラブルシューティング

アプリケーション、新機能、コードの修正をターゲットの環境で利用できるようにするプロセス。デプロイでは、コードベースに変更を施した後、アプリケーションの環境でそのコードベースを構築して実行します。

開発環境

「環境」を参照してください。

検出管理

イベントが発生したときに、検出、ログ記録、警告を行うように設計されたセキュリティコントロール。これらのコントロールは副次的な防衛手段であり、実行中の予防的コントロールをすり抜けたセキュリティイベントをユーザーに警告します。詳細については、「 AWSでのセキュリティコントロールの実装」の「検出的コントロール」を参照してください。

開発バリューストリームマッピング (DVSM)

ソフトウェア開発ライフサイクルのスピードと品質に悪影響を及ぼす制約を特定し、優先順位を付けるために使用されるプロセス。DVSM は、もともとリーンマニュファクチャリング・プラクティスのために設計されたバリューストリームマッピング・プロセスを拡張したものです。ソフトウェア開発プロセスを通じて価値を創造し、動かすために必要なステップとチームに焦点を当てています。

デジタルツイン

建物、工場、産業機器、生産ラインなど、現実世界のシステムを仮想的に表現したものです。デジタルツインは、予知保全、リモートモニタリング、生産最適化をサポートします。

ディメンションテーブル

スタースキーマにおいて、ファクトテーブルの定量データに関するデータ属性が含まれる小さいテーブル。ディメンションテーブルの属性は、通常、テキストフィールド、またはテキストのように扱える個別の数値で示されます。これらの属性は、一般的に、クエリの制約、フィルタリング、結果セットのラベル付けに使用されます。

ディザスタ

ワークロードまたはシステムが、導入されている主要な場所でのビジネス目標の達成を妨げるイベント。これらのイベントは、自然災害、技術的障害、または意図しない設定ミスやマルウェア攻撃などの人間の行動の結果である場合があります。

ディザスタリカバリ (DR)

ディザスタによるダウンタイムとデータ損失を最小限に抑えるための戦略とプロセス。詳細については、 AWS Well-Architected フレームワークの「 でのワークロードのディザスタリカバリ AWS: クラウドでのリカバリ」を参照してください。

DML

「データベース操作言語」を参照してください。

ドメイン駆動型設計

各コンポーネントが提供している変化を続けるドメイン、またはコアビジネス目標にコンポーネントを接続して、複雑なソフトウェアシステムを開発するアプローチ。この概念は、エリック・エヴァンスの著書、Domain-Driven Design: Tackling Complexity in the Heart of Software (ドメイン駆動設計:ソフトウェアの中心における複雑さへの取り組み) で紹介されています (ボストン: Addison-Wesley Professional、2003)。strangler fig パターンでドメイン駆動型設計を使用する方法の詳細については、「コンテナと Amazon API Gateway を使用して、従来の Microsoft ASP.NET (ASMX) ウェブサービスを段階的にモダナイズ」を参照してください。

DR

「ディザスタリカバリ」を参照してください。

ドリフト検出

ベースライン設定からの偏差を追跡します。たとえば、 AWS CloudFormation を使用してシステムリソースのドリフトを検出したり、 を使用して AWS Control Tower 、ガバナンス要件への準拠に影響するランディングゾーンの変更を検出したりできます。

DVSM

「開発バリューストリームマッピング」を参照してください。

E

EDA

「探索的データ分析」を参照してください。

EDI

「電子データ交換」を参照してください。

エッジコンピューティング

IoT ネットワークのエッジにあるスマートデバイスの計算能力を高めるテクノロジー。クラウドコンピューティングと比較すると、エッジコンピューティングは通信レイテンシーを短縮し、応答時間を改善できます。

電子データ交換 (EDI)

組織間で行う、ビジネスドキュメントの自動交換。詳細については、「電子データ交換とは」を参照してください。

暗号化

人間が読み取り可能なプレーンテキストデータを暗号文に変換するコンピューティング処理。

暗号化キー

暗号化アルゴリズムが生成した、ランダム化されたビットからなる暗号文字列。キーの長さは決まっておらず、各キーは予測できないように、一意になるように設計されています。

エンディアン

コンピュータメモリにバイトが格納される順序。ビッグエンディアンシステムでは、最上位バイトが最初に格納されます。リトルエンディアンシステムでは、最下位バイトが最初に格納されます。

エンドポイント

「サービスエンドポイント」を参照してください。

エンドポイントサービス

仮想プライベートクラウド (VPC) 内でホストして、他のユーザーと共有できるサービス。を使用してエンドポイントサービスを作成し AWS PrivateLink 、他の AWS アカウント または AWS Identity and Access Management (IAM) プリンシパルにアクセス許可を付与できます。これらのアカウントまたはプリンシパルは、インターフェイス VPC エンドポイントを作成することで、エンドポイントサービスにプライベートに接続できます。詳細については、Amazon Virtual Private Cloud (Amazon VPC) ドキュメントの「エンドポイントサービスを作成する」を参照してください。

エンタープライズリソースプランニング (ERP)

エンタープライズの主要なビジネスプロセス (会計、MES、プロジェクト管理など) を自動化および管理するシステム。

エンベロープ暗号化

暗号化キーを、別の暗号化キーを使用して暗号化するプロセス。詳細については、 AWS Key Management Service (AWS KMS) ドキュメントの「エンベロープ暗号化」を参照してください。

環境

実行中のアプリケーションのインスタンス。クラウドコンピューティングにおける一般的な環境の種類は以下のとおりです。

- 開発環境 — アプリケーションのメンテナンスを担当するコアチームのみが使用できる、実行中のアプリケーションのインスタンス。開発環境は、上位の環境に昇格させる変更をテストするときに使用します。このタイプの環境は、テスト環境と呼ばれることもあります。
- 下位環境 — 初期ビルドやテストに使用される環境など、アプリケーションのすべての開発環境。
- 本番環境 — エンドユーザーがアクセスできる、実行中のアプリケーションのインスタンス。CI/CD パイプラインでは、本番環境が最後のデプロイ環境になります。
- 上位環境 — コア開発チーム以外のユーザーがアクセスできるすべての環境。これには、本番環境、本番前環境、ユーザー承認テスト環境などが含まれます。

エピック

アジャイル方法論で、お客様の作業の整理と優先順位付けに役立つ機能カテゴリ。エピックでは、要件と実装タスクの概要についてハイレベルな説明を提供します。例えば、 AWS CAF セキュリティエピックには、ID とアクセスの管理、検出コントロール、インフラストラクチャセキュリティ、データ保護、インシデント対応が含まれます。 AWS 移行戦略のエピックの詳細については、プログラム実装ガイドを参照してください。

ERP

「エンタープライズリソース計画」を参照してください。

探索的データ分析 (EDA)

データセットを分析してその主な特性を理解するプロセス。お客様は、データを収集または集計してから、パターンの検出、異常の検出、および前提条件のチェックのための初期調査を実行します。EDA は、統計の概要を計算し、データの可視化を作成することによって実行されます。

F

ファクトテーブル

[スタースキーマ]の中央にあるテーブル。ビジネスオペレーションに関する定量的データが保存されます。一般的に、ファクトテーブルは、2 種類の列で構成されます。1 つは測定値が含まれる列、もう 1 つはディメンションテーブルへの外部キーが含まれる列です。

フェイルファスト

開発ライフサイクルを短縮するために、頻繁かつ段階的にテストを行う哲学であり、アジャイルアプローチでは、この考え方がきわめて重要です。

障害分離境界

では AWS クラウド、障害の影響を制限し、ワークロードの耐障害性を高めるのに役立つアベイラビリティーゾーン AWS リージョン、コントロールプレーン、データプレーンなどの境界。詳細については、「AWS 障害分離境界」を参照してください。

機能ブランチ

「ブランチ」を参照してください。

特徴量

お客様が予測に使用する入力データ。例えば、製造コンテキストでは、特徴量は製造ラインから定期的にキャプチャされるイメージの可能性もあります。

特徴量重要度

モデルの予測に対する特徴量の重要性。これは通常、Shapley Additive Deskonations (SHAP) や積分勾配など、さまざまな手法で計算できる数値スコアで表されます。詳細については、「 を使用した機械学習モデルの解釈可能性 AWS」を参照してください。

機能変換

追加のソースによるデータのエンリッチ化、値のスケーリング、単一のデータフィールドからの複数の情報セットの抽出など、機械学習プロセスのデータを最適化すること。これにより、機械学習モデルはデータの恩恵を受けることができます。例えば、「2021-05-27 00:15:37」の日付を「2021 年」、「5 月」、「木」、「15」に分解すると、学習アルゴリズムがさまざまなデータコンポーネントに関連する微妙に異なるパターンを学習するのに役立ちます。

数ショットプロンプト

LLM に、タスクと望ましい出力を示す例を少数提示した後に、類似のタスクを実行させること。この手法は、プロンプトに記述された例 (ショット) からモデルが学習する「インコンテキスト学

習」の一種です。数ショットプロンプトは、特定のフォーマット、推論、専門知識が必要なタスクに効果的です。「ゼロショットプロンプト」も参照してください。

FGAC

「きめ細かなアクセス制御」を参照してください。

きめ細かなアクセス制御 (FGAC)

複数の条件を使用してアクセス要求を許可または拒否すること。

フラッシュカット移行

変更データのキャプチャによる継続的なデータ複製を利用して、段階的なアプローチではなく、可能な限り短時間でデータを移行するデータベース移行方法。目的はダウンタイムを最小限に抑えることです。

FM

「基盤モデル」を参照してください。

基盤モデル (FM)

大規模な深層学習ニューラルネットワークであり、一般化およびラベル付けされていないデータからなる大規模データセットでトレーニングされています。FM により、言語理解、テキストおよび画像生成、自然言語での会話といった、一般的な各種タスクを実行できます。詳細については、「基盤モデルとは何ですか?」を参照してください。

G

生成 AI

AI モデルのサブセット。大量のデータでトレーニングされており、シンプルなテキストプロンプトを使用して、画像、動画、テキスト、オーディオなどの新しいコンテンツやアーティファクトを作成できます。詳細については、「生成 AI とは何ですか?」を参照してください。

ジオブロッキング

「地理的制限」を参照してください。

地理的制限 (ジオブロッキング)

特定の国のユーザーがコンテンツ配信にアクセスできないようにするための、Amazon CloudFront のオプション。アクセスを許可する国と禁止する国は、許可リストまたは禁止リスト

を使って指定します。詳細については、CloudFront ドキュメントの「コンテンツの地理的ディストリビューションの制限」を参照してください。

Gitflow ワークフロー

下位環境と上位環境が、ソースコードリポジトリでそれぞれ異なるブランチを使用する方法。Gitflow ワークフローは古いと見なされている方法であり、トランクベースのワークフローは推奨されている新しい方法です。

ゴールデンイメージ

システムまたはソフトウェアのスナップショットであり、システムまたはソフトウェアの新規インスタンスをデプロイするテンプレートとして使用されます。製造の例で言えば、ゴールデンイメージを使用すると、複数のデバイスにソフトウェアをプロビジョニングして、デバイス製造オペレーションの速度、スケーラビリティ、生産性を向上させることができます。

グリーンフィールド戦略

新しい環境に既存のインフラストラクチャが存在しないこと。システムアーキテクチャにグリーンフィールド戦略を導入する場合、既存のインフラストラクチャ (別名ブラウンフィールド) との互換性の制約を受けることなく、あらゆる新しいテクノロジーを選択できます。既存のインフラストラクチャを拡張している場合は、ブラウンフィールド戦略とグリーンフィールド戦略を融合させることもできます。

ガードレール

組織単位 (OU) 全般のリソース、ポリシー、コンプライアンスを管理するのに役立つ概略的なルール。予防ガードレールは、コンプライアンス基準に一致するようにポリシーを実施します。これらは、サービスコントロールポリシーと IAM アクセス許可の境界を使用して実装されます。検出ガードレールは、ポリシー違反やコンプライアンス上の問題を検出し、修復のためのアラートを発信します。これらは AWS Config、、 AWS Security Hub CSPM、Amazon GuardDuty、、 AWS Trusted Advisor Amazon Inspector、およびカスタム AWS Lambda チェックを使用して実装されます。

H

HA

「高可用性」を参照してください。

異種混在データベースの移行

別のデータベースエンジンを使用するターゲットデータベースへお客様の出典データベースの移行 (例えば、Oracle から Amazon Aurora)。異種間移行は通常、アーキテクチャの再設計作業の一部であり、スキーマの変換は複雑なタスクになる可能性があります。AWS は、スキーマの変換に役立つ AWS SCTを提供します。

高可用性 (HA)

課題や災害が発生した場合に、介入なしにワークロードを継続的に運用できること。HA システムは、自動的にフェイルオーバーし、一貫して高品質のパフォーマンスを提供し、パフォーマンスへの影響を最小限に抑えながらさまざまな負荷や障害を処理するように設計されています。

ヒストリアンのモダナイゼーション

製造業のニーズによりよく応えるために、オペレーションテクノロジー (OT) システムをモダナイズし、アップグレードするためのアプローチ。ヒストリアンは、工場内のさまざまなソースからデータを収集して保存するために使用されるデータベースの一種です。

ホールドアウトデータ

機械学習モデルのトレーニング用データセットから保留される、ラベル付き履歴データの一部。ホールドアウトデータを使用すると、モデル予測をホールドアウトデータと比較して、モデルのパフォーマンスを評価できます。

同種データベースの移行

お客様の出典データベースを、同じデータベースエンジンを共有するターゲットデータベース (Microsoft SQL Server から Amazon RDS for SQL Server など) に移行する。同種間移行は、通常、リホストまたはリプラットフォーム化の作業の一部です。ネイティブデータベースユーティリティを使用して、スキーマを移行できます。

ホットデータ

リアルタイムデータや最近の翻訳データなど、頻繁にアクセスされるデータ。通常、このデータには高速なクエリ応答を提供する高性能なストレージ階層またはクラスが必要です。

ホットフィックス

本番環境の重大な問題を修正するために緊急で配布されるプログラム。緊急性が高いため、通常の DevOps のリリースワークフローからは外れた形で実施されます。

ハイパーケア期間

カットオーバー直後、移行したアプリケーションを移行チームがクラウドで管理、監視して問題に対処する期間。通常、この期間は 1～4 日です。ハイパーケア期間が終了すると、アプリケーションに対する責任は一般的に移行チームからクラウドオペレーションチームに移ります。

I

IaC

「Infrastructure as Code」を参照してください。

ID ベースのポリシー

AWS クラウド 環境内のアクセス許可を定義する 1 つ以上の IAM プリンシパルにアタッチされたポリシー。

アイドル状態のアプリケーション

90 日間の平均的な CPU およびメモリ使用率が 5～20% のアプリケーション。移行プロジェクトでは、これらのアプリケーションを廃止するか、オンプレミスに保持するのが一般的です。

IIoT

「インダストリアル IoT」を参照してください。

イミュータブルインフラストラクチャ

既存インフラストラクチャの更新、パッチ適用、変更などを行わずに、本番環境ワークロードに使用する新規インフラストラクチャをデプロイするモデル。本質的に、イミュータブルインフラストラクチャは、ミュータブルインフラストラクチャよりも一貫性、信頼性、予測性に優れています。詳細については、 AWS Well-Architected フレームワークにある「イミュータブルインフラストラクチャを使用してデプロイする」のベストプラクティスを参照してください。

インバウンド (受信) VPC

AWS マルチアカウントアーキテクチャでは、アプリケーションの外部からネットワーク接続を受け入れ、検査し、ルーティングする VPC。AWS Security Reference Architecture では、アプリケーションとより広範なインターネット間の双方向のインターフェイスを保護するために、インバウンド、アウトバウンド、インスペクションの各 VPC を使用してネットワークアカウントを設定することを推奨しています。

増分移行

アプリケーションを 1 回ですべてカットオーバーするのではなく、小さい要素に分けて移行するカットオーバー戦略。例えば、最初は少数のマイクロサービスまたはユーザーのみを新しいシステムに移行する場合があります。すべてが正常に機能することを確認できたら、残りのマイクロサービスやユーザーを段階的に移行し、レガシーシステムを廃止できるようにします。この戦略により、大規模な移行に伴うリスクが軽減されます。

インダストリー 4.0

2016 年に Klaus Schwab 氏が提唱した用語で、接続、リアルタイムデータ、オートメーション、分析、AI/ML の進歩による、ビジネスプロセスのモダナイズを意味します。

インフラストラクチャ

アプリケーションの環境に含まれるすべてのリソースとアセット。

Infrastructure as Code (IaC)

アプリケーションのインフラストラクチャを一連の設定ファイルを使用してプロビジョニングし、管理するプロセス。IaC は、新しい環境を再現可能で信頼性が高く、一貫性のあるものにするため、インフラストラクチャを一元的に管理し、リソースを標準化し、スケールを迅速に行えるように設計されています。

インダストリアル IoT (IIoT)

製造、エネルギー、自動車、ヘルスケア、ライフサイエンス、農業などの産業部門におけるインターネットに接続されたセンサーやデバイスの使用。詳細については、「インダストリアル IoT (IIoT) デジタルトランスフォーメーション戦略の構築」を参照してください。

インスペクション VPC

AWS マルチアカウントアーキテクチャでは、VPC (同一または異なる 内 AWS リージョン)、インターネット、オンプレミスネットワーク間のネットワークトラフィックの検査を管理する一元化された VPCs。AWS Security Reference Architecture では、アプリケーションとより広範なインターネット間の双方向のインターフェイスを保護するために、インバウンド、アウトバウンド、インスペクションの各 VPC を使用してネットワークアカウントを設定することを推奨しています。

IoT

インターネットまたはローカル通信ネットワークを介して他のデバイスやシステムと通信する、センサーまたはプロセッサが組み込まれた接続済み物理オブジェクトのネットワーク。詳細については、「IoT とは」を参照してください。

解釈可能性

機械学習モデルの特性で、モデルの予測がその入力にどのように依存するかを人間が理解できる度合いを表します。詳細については、「 を使用した機械学習モデルの解釈可能性 AWS」を参照してください。

IoT

「IoT」を参照してください。

IT 情報ライブラリ (ITIL)

IT サービスを提供し、これらのサービスをビジネス要件に合わせるための一連のベストプラクティス。ITIL は ITSM の基盤を提供します。

IT サービス管理 (ITSM)

組織の IT サービスの設計、実装、管理、およびサポートに関連する活動。クラウドオペレーションと ITSM ツールの統合については、オペレーション統合ガイドを参照してください。

ITIL

「IT 情報ライブラリ」を参照してください。

ITSM

「IT サービス管理」を参照してください。

L

ラベルベースアクセス制御 (LBAC)

強制アクセス制御 (MAC) の実装で、ユーザーとデータ自体にそれぞれセキュリティラベル値が明示的に割り当てられます。ユーザーセキュリティラベルとデータセキュリティラベルが交差する部分によって、ユーザーに表示される行と列が決まります。

ランディングゾーン

ランディングゾーンは、スケーラブルで安全な、適切に設計されたマルチアカウント AWS 環境です。これは、組織がセキュリティおよびインフラストラクチャ環境に自信を持ってワークロードとアプリケーションを迅速に起動してデプロイできる出発点です。ランディングゾーンの詳細については、「安全でスケーラブルなマルチアカウント AWS 環境のセットアップ」を参照してください。

大規模言語モデル (LLM)

大量のデータで事前トレーニングされた深層学習 AI モデル。LLM では、質問への回答、ドキュメントの要約、他言語へのテキスト翻訳、文を完成させるなど、さまざまなタスクを実行できます。詳細については、「大規模言語モデル (LLM) とは何ですか?」を参照してください。

大規模な移行

300 台以上のサーバの移行。

LBAC

「ラベルベースアクセス制御」を参照してください。

最小特権

タスクの実行には必要最低限の権限を付与するという、セキュリティのベストプラクティス。詳細については、IAM ドキュメントの「最小特権アクセス許可を適用する」を参照してください。

リフトアンドシフト

「7 Rs」を参照してください。

リトルエンディアンシステム

最下位バイトを最初に格納するシステム。「エンディアン性」もご覧ください。

LLM

「大規模言語モデル」を参照してください。

下位環境

「環境」を参照してください。

M

機械学習 (ML)

パターン認識と学習にアルゴリズムと手法を使用する人工知能の一種。ML は、モノのインターネット (IoT) データなどの記録されたデータを分析して学習し、パターンに基づく統計モデルを生成します。詳細については、「機械学習」を参照してください。

メインブランチ

「ブランチ」を参照してください。

マルウェア

コンピュータのセキュリティやプライバシーを侵害するように設計されたソフトウェア。マルウェアは、コンピュータシステムの中断、機密情報の漏洩、不正アクセスを招く可能性があります。マルウェアの例には、ウイルス、ワーム、ランサムウェア、トロイの木馬、スパイウェア、キーロガーなどがあります。

マネージドサービス

AWS のサービス はインフラストラクチャレイヤー、オペレーティングシステム、プラットフォーム AWS を運用し、エンドポイントにアクセスしてデータを保存および取得します。マネージドサービスの例として、Amazon Simple Storage Service (Amazon S3) と Amazon DynamoDB が挙げられます。このサービスは、抽象化されたサービスとも呼ばれます。

製造実行システム (MES)

生産プロセスを追跡、モニタリング、文書化、制御するソフトウェアシステムであり、工場では、これによって、原材料から製品を完成させます。

MAP

「Migration Acceleration Program」を参照してください。

メカニズム

ツールを作成してその導入を推進し、導入結果を調べて調整を行うための包括的なプロセス。メカニズムとは、運用中にそれ自体を強化し改善するサイクルを意味します。詳細については、AWS 「 Well-Architected フレームワーク」の「メカニズムの構築」を参照してください。

メンバーアカウント

組織の一部である管理アカウント AWS アカウント 以外のすべて AWS Organizations。 アカウントが 組織のメンバーになることができるのは、一度に 1 つのみです。

MES

「製造実行システム」を参照してください。

Message Queuing Telemetry Transport (MQTT)

発行/サブスクライブのパターンに基づく、軽量のマシンツーマシン (M2M) 通信プロトコルであり、リソースに限りのある IoT デバイスに使用されます。

マイクロサービス

明確に定義された API を介して通信し、通常は小規模な自己完結型のチームが所有する、小規模で独立したサービスです。例えば、保険システムには、販売やマーケティングなどのビジネス

機能、または購買、請求、分析などのサブドメインにマッピングするマイクロサービスが含まれる場合があります。マイクロサービスの利点には、俊敏性、柔軟なスケーリング、容易なデプロイ、再利用可能なコード、回復力などがあります。詳細については、AWS「サーバーレスサービスを使用したマイクロサービスの統合」を参照してください。

マイクロサービスアーキテクチャ

各アプリケーションプロセスをマイクロサービスとして実行する独立したコンポーネントを使用してアプリケーションを構築するアプローチ。これらのマイクロサービスは、軽量 API を使用して、明確に定義されたインターフェイスを介して通信します。このアーキテクチャの各マイクロサービスは、アプリケーションの特定の機能に対する需要を満たすように更新、デプロイ、およびスケーリングできます。詳細については、「 でのマイクロサービスの実装 AWS」を参照してください。

Migration Acceleration Program (MAP)

組織がクラウドに移行するための強力な運用基盤を構築し、移行の初期コストを相殺するのに役立つコンサルティングサポート、トレーニング、サービスを提供する AWS プログラム。MAP には、組織的な方法でレガシー移行を実行するための移行方法論と、一般的な移行シナリオを自動化および高速化する一連のツールが含まれています。

大規模な移行

アプリケーションポートフォリオの大部分を次々にクラウドに移行し、各ウェーブでより多くのアプリケーションを高速に移動させるプロセス。この段階では、以前の段階から学んだベストプラクティスと教訓を使用して、移行ファクトリー チーム、ツール、プロセスのうち、オートメーションとアジャイルデリバリーによってワークロードの移行を合理化します。これは、AWS 移行戦略 の第 3 段階です。

移行ファクトリー

自動化された俊敏性のあるアプローチにより、ワークロードの移行を合理化する部門横断的なチーム。移行ファクトリーチームには、通常、運用、ビジネスアナリストおよび所有者、移行エンジニア、デベロッパー、およびスプリントで作業する DevOps プロフェッショナルが含まれます。エンタープライズアプリケーションポートフォリオの 20～50% は、ファクトリーのアプローチによって最適化できる反復パターンで構成されています。詳細については、このコンテンツセットの移行ファクトリーに関する解説とCloud Migration Factory ガイドを参照してください。

移行メタデータ

移行を完了するために必要なアプリケーションおよびサーバーに関する情報。移行パターンごとに、異なる一連の移行メタデータが必要です。移行メタデータの例としては、ターゲットサブネット、セキュリティグループ、 AWS アカウントなどがあります。

移行パターン

移行戦略、移行先、および使用する移行アプリケーションまたはサービスを詳述する、反復可能な移行タスク。例: AWS Application Migration Service を使用して Amazon EC2 への移行をリホストします。

Migration Portfolio Assessment (MPA)

オンラインツール。これによって、 AWS クラウドに移行するビジネスケースの検証に必要な情報を得られます。MPA は、詳細なポートフォリオ評価 (サーバーの適切なサイジング、価格設定、TCO 比較、移行コスト分析) および移行プラン (アプリケーションデータの分析とデータ収集、アプリケーションのグループ化、移行の優先順位付け、およびウェーブプランニング) を提供します。MPA ツール (ログインが必要) は、すべての AWS コンサルタントと APN パートナーコンサルタントが無料で利用できます。

移行準備状況評価 (MRA)

AWS CAF を使用して、組織のクラウド準備状況に関するインサイトを取得し、長所と短所を特定し、特定されたギャップを埋めるためのアクションプランを構築するプロセス。詳細については、移行準備状況ガイドを参照してください。MRA は、AWS 移行戦略の第一段階です。

移行戦略

ワークロードを AWS クラウドに移行するために使用するアプローチ。詳細については、この用語集の 7 Rs エントリと、「組織を動員して大規模な移行を加速する」を参照してください。

ML

「機械学習」を参照してください。

モダナイゼーション

古い (レガシーまたはモノリシック) アプリケーションとそのインフラストラクチャをクラウド内の俊敏で弾力性のある高可用性システムに変換して、コストを削減し、効率を高め、イノベーションを活用します。詳細については、「AWS クラウドでのアプリケーションのモダナイズ戦略」を参照してください。

モダナイゼーション準備状況評価

組織のアプリケーションのモダナイゼーションの準備状況を判断し、利点、リスク、依存関係を特定し、組織がこれらのアプリケーションの将来の状態をどの程度適切にサポートできるかを決定するのに役立つ評価。評価の結果として、ターゲットアーキテクチャのブループリント、モダナイゼーションプロセスの開発段階とマイルストーンを詳述したロードマップ、特定されたギャップに対処するためのアクションプランが得られます。詳細については、「AWS クラウドでのアプリケーションのモダナイゼーションの準備状況を評価する」を参照してください。

モノリシックアプリケーション (モノリス)

緊密に結合されたプロセスを持つ単一のサービスとして実行されるアプリケーション。モノリシックアプリケーションにはいくつかの欠点があります。1 つのアプリケーション機能エクスペリエンスの需要が急増する場合は、アーキテクチャ全体をスケーリングする必要があります。モノリシックアプリケーションの特徴を追加または改善することは、コードベースが大きくなると複雑になります。これらの問題に対処するには、マイクロサービスアーキテクチャを使用できます。詳細については、「モノリスをマイクロサービスに分解する」を参照してください。

MPA

「Migration Portfolio Assessment」を参照してください。

MQTT

「Message Queuing Telemetry Transport」を参照してください。

多クラス分類

複数のクラスの予測を生成するプロセス (2 つ以上の結果の 1 つを予測します)。例えば、機械学習モデルが、「この製品は書籍、自動車、電話のいずれですか?」 または、「このお客様にとって最も関心のある商品のカテゴリはどれですか?」と聞くかもしれません。

ミュータブルなインフラストラクチャ

本番ワークロードに使用する既存のインフラストラクチャを更新および変更するためのモデル。Well-Architected AWS フレームワークでは、一貫性、信頼性、予測可能性を向上させるために、イミュータブルインフラストラクチャの使用をベストプラクティスとして推奨しています。

O

OAC

「オリジンアクセス制御」を参照してください。

OAI

「オリジンアクセスアイデンティティ」を参照してください。

OCM

「組織変更管理」を参照してください。

オフライン移行

移行プロセス中にソースワークロードを停止させる移行方法。この方法はダウンタイムが長くなるため、通常は重要ではない小規模なワークロードに使用されます。

OI

「オペレーション統合」を参照してください。

Ola

「オペレーショナルレベルアグリーメント」を参照してください。

オンライン移行

ソースワークロードをオフラインにせずにターゲットシステムにコピーする移行方法。ワークロードに接続されているアプリケーションは、移行中も動作し続けることができます。この方法はダウンタイムがゼロから最小限で済むため、通常は重要な本番稼働環境のワークロードに使用されます。

OPC-UA

「Open Process Communications - Unified Architecture」を参照してください。

Open Process Communications - Unified Architecture (OPC-UA)

産業オートメーション用のマシンツーマシン (M2M) 通信プロトコル。OPC-UA により、相互運用の際に、データ暗号化、認証、認可の各スキームを標準化できます。

オペレーショナルレベルアグリーメント (OLA)

サービスレベルアグリーメント (SLA) をサポートするために、どの機能的 IT グループが互いに提供することを約束するかを明確にする契約。

運用準備状況レビュー (ORR)

質問と関連するベストプラクティスのチェックリスト。インシデントや起こり得る障害を理解、評価、防止したり、その範囲を縮小したりする際に役立ちます。詳細については、 AWS Well-Architected フレームワークの「Operational Readiness Reviews (ORR)」を参照してください。

運用テクノロジー (OT)

産業オペレーション、機器、インフラストラクチャを制御するために物理環境と連携させるハードウェアおよびソフトウェアシステム。製造分野では、Industry 4.0 への変革を進める上で、OT と情報技術 (IT) システムの統合に焦点が当てられています。

オペレーション統合 (OI)

クラウドでオペレーションをモダナイズするプロセスには、準備計画、オートメーション、統合が含まれます。詳細については、オペレーション統合ガイドを参照してください。

組織の証跡

組織 AWS アカウント 内のすべての のすべてのイベント AWS CloudTrail を ログに記録する によって作成された証跡 AWS Organizations。証跡は、組織に含まれている各 AWS アカウント に作成され、各アカウントのアクティビティを追跡します。詳細については、CloudTrail ドキュメントの「組織の証跡の作成」を参照してください。

組織変更管理 (OCM)

人材、文化、リーダーシップの観点から、主要な破壊的なビジネス変革を管理するためのフレームワーク。OCM は、変化の導入を加速し、移行問題に対処し、文化や組織の変化を推進することで、組織が新しいシステムと戦略の準備と移行するのを支援します。 AWS 移行戦略では、クラウド導入プロジェクトに必要な変化のスピードにより、このフレームワークは人材アクセラレーションと呼ばれます。詳細については、OCM ガイドを参照してください。

オリジンアクセス制御 (OAC)

Amazon Simple Storage Service (Amazon S3) コンテンツを保護するための、CloudFront のアクセス制限の強化オプション。OAC は AWS リージョン、すべての S3 バケット、 AWS KMS (SSE-KMS) によるサーバー側の暗号化、S3 バケットへの動的 PUT および DELETEリクエストをサポートします。

オリジンアクセスアイデンティティ (OAI)

CloudFront の、Amazon S3 コンテンツを保護するためのアクセス制限オプション。OAI を使用すると、CloudFront が、Amazon S3 に認証可能なプリンシパルを作成します。認証されたプリンシパルは、S3 バケット内のコンテンツに、特定の CloudFront ディストリビューションを介してのみアクセスできます。OAC も併せて参照してください。OAC では、より詳細な、強化されたアクセス制御が可能です。

ORR

「運用準備状況レビュー」を参照してください。

OT

「運用テクノロジー」を参照してください。

アウトバウンド (送信) VPC

AWS マルチアカウントアーキテクチャでは、アプリケーション内から開始されたネットワーク接続を処理する VPC。AWS Security Reference Architecture では、アプリケーションとより広範なインターネット間の双方向のインターフェイスを保護するために、インバウンド、アウトバウンド、インスペクションの各 VPC を使用してネットワークアカウントを設定することを推奨しています。

P

アクセス許可の境界

ユーザーまたはロールが使用できるアクセス許可の上限を設定する、IAM プリンシパルにアタッチされる IAM 管理ポリシー。詳細については、IAM ドキュメントのアクセス許可の境界を参照してください。

個人を特定できる情報 (PII)

直接閲覧した場合、または他の関連データと組み合わせた場合に、個人の身元を合理的に推測するために使用できる情報。PII の例には、氏名、住所、連絡先情報などがあります。

PII

「個人を特定できる情報」を参照してください。

プレイブック

クラウドでのコアオペレーション機能の提供など、移行に関連する作業を取り込む、事前定義された一連のステップ。プレイブックは、スクリプト、自動ランブック、またはお客様のモダナイズされた環境を運用するために必要なプロセスや手順の要約などの形式をとることができます。

PLC

「プログラマブルロジックコントローラー」を参照してください。

PLM

「製品ライフサイクル管理」を参照してください。

ポリシー

次の操作を可能にするオブジェクト: アクセス許可を定義する (ID ベースのポリシーを参照)。アクセス条件を指定する (リソースベースのポリシーを参照)。 AWS Organizations の組織における全アカウントにアクセス許可の上限を定義する (サービスコントロールポリシーを参照)。

多言語の永続性

データアクセスパターンやその他の要件に基づいて、マイクロサービスのデータストレージテクノロジーを個別に選択します。マイクロサービスが同じデータストレージテクノロジーを使用している場合、実装上の問題が発生したり、パフォーマンスが低下する可能性があります。マイクロサービスは、要件に最も適合したデータストアを使用すると、より簡単に実装でき、パフォーマンスとスケーラビリティが向上します。

ポートフォリオ評価

移行を計画するために、アプリケーションポートフォリオの検出、分析、優先順位付けを行うプロセス。詳細については、「移行の準備状況の評価」を参照してください。

述語

`true` または `false` を返すためのクエリ条件。一般的に、`WHERE` 句に記述されます。

述語プッシュダウン

データベースクエリを最適化する手法。これによって、転送前にクエリ内のデータをフィルタリングします。この手法を取ると、リレーショナルデータベースから取得し処理する必要のあるデータの量が減少するため、クエリのパフォーマンスが向上します。

予防的コントロール

イベントの発生を防ぐように設計されたセキュリティコントロール。このコントロールは、ネットワークへの不正アクセスや好ましくない変更を防ぐ最前線の防御です。詳細については、「AWSでのセキュリティコントロールの実装」の「予防的コントロール」を参照してください。

プリンシパル

アクションを実行し AWS 、リソースにアクセスできる のエンティティ。このエンティティは通常、、IAM AWS アカウントロール、または ユーザーのルートユーザーです。詳細については、IAM ドキュメントの「ロールに関する用語と概念」にあるプリンシパルを参照してください。

プライバシーバイデザイン

開発プロセス全体を通してプライバシーが考慮されているシステムエンジニアリングのアプローチ。

プライベートホストゾーン

1 つ以上の VPC 内のドメインとそのサブドメインへの DNS クエリに対し、Amazon Route 53 がどのように応答するかに関する情報を保持するコンテナ。詳細については、Route 53 ドキュメントの「プライベートホストゾーンの使用」を参照してください。

プロアクティブコントロール

非準拠リソースのデプロイ防止を目的としたセキュリティコントロール。このコントロールにより、プロビジョニング前にリソースをスキャンします。コントロールに準拠していないリソースは、プロビジョニングされません。詳細については、 AWS Control Tower ドキュメントの「コントロールリファレンスガイド」および「 セキュリティコントロールの実装」の「プロアクティブコントロール」を参照してください。 AWS

製品ライフサイクル管理 (PLM)

製品の設計、開発、発売から、成長、成熟、衰退、廃棄に至る、製品のライフサイクル全体を通してデータとプロセスを管理すること。

本番環境

「環境」を参照してください。

プログラマブルロジックコントローラー (PLC)

製造分野で使用される、信頼性と適応性に優れたコンピュータであり、これによって、マシンをモニタリングするとともに、製造プロセスを自動化します。

プロンプトチェイニング

1 つの LLM プロンプトによる出力を次のプロンプトの入力に使用して、より良いレスポンスを生成します。この手法を使用すると、複雑なタスクをサブタスクに分割したり、事前レスポンスを繰り返し改良または拡張したりできます。これによって、モデルのレスポンスの精度と関連性が向上し、粒度の高いパーソナライズされた結果を得られます。

仮名化

データセット内の個人識別子をプレースホルダー値に置き換えるプロセス。仮名化は個人のプライバシー保護に役立ちます。仮名化されたデータは、依然として個人データとみなされます。

発行/サブスクライブ (pub/sub)

マイクロサービス間の非同期通信を可能にするパターン。これにより、スケーラビリティと応答性を向上させます。例えば、マイクロサービスベースの MES の場合、マイクロサービスは、他のマイクロサービスがサブスクライブ可能なチャネルにイベントメッセージを発行できます。このシステムでは、発行サービスの変更なしに、新規マイクロサービスを追加できます。

Q

クエリプラン

手順などの一連のステップであり、SQL リレーショナルデータベースシステムのデータにアクセスするために使用されます。

クエリプランのリグレッション

データベースサービスのオプティマイザーが、データベース環境に特定の変更が加えられる前に選択されたプランよりも最適性の低いプランを選択すること。これは、統計、制限事項、環境設定、クエリパラメータのバインディングの変更、およびデータベースエンジンの更新などが原因である可能性があります。

R

RACI マトリックス

「実行責任者、説明責任者、協業先、報告先 (RACI)」を参照してください。

RAG

「検索拡張生成」を参照してください。

ランサムウェア

決済が完了するまでコンピュータシステムまたはデータへのアクセスをブロックするように設計された、悪意のあるソフトウェア。

RASCI マトリックス

「実行責任者、説明責任者、協業先、報告先 (RACI)」を参照してください。

RCAC

「行と列のアクセス制御」を参照してください。

リードレプリカ

読み取り専用に使用されるデータベースのコピー。クエリをリードレプリカにルーティングして、プライマリデータベースへの負荷を軽減できます。

リアーキテクト

「7 Rs」を参照してください。

目標復旧時点 (RPO)

最後のデータリカバリポイントからの最大許容時間です。これにより、最後の回復時点からサービスが中断されるまでの間に許容できるデータ損失の程度が決まります。

目標復旧時間 (RTO)

サービスの中断から復旧までの最大許容遅延時間。

リファクタリング

「7 Rs」を参照してください。

リージョン

地理的エリア内の AWS リソースのコレクション。各 AWS リージョン は、耐障害性、安定性、耐障害性を提供するために、他の から分離され、独立しています。詳細については、「アカウントが使用できる AWS リージョン を指定する」を参照してください。

リグレッション

数値を予測する機械学習手法。例えば、「この家はどれくらいの値段で売れるでしょうか?」という問題を解決するために、機械学習モデルは、線形回帰モデルを使用して、この家に関する既知の事実 (平方フィートなど) に基づいて家の販売価格を予測できます。

リホスト

「7 Rs」を参照してください。

リリース

デプロイプロセスで、変更を本番環境に昇格させること。

再配置

「7 Rs」を参照してください。

リプラットフォーム

「7 Rs」を参照してください。

再購入

「7 Rs」を参照してください。

回復性

中断に抵抗または中断から回復するアプリケーションの機能。 AWS クラウドでの回復力を計画する際には、一般的に、高可用性とディザスタリカバリが考慮されます。詳細については、「AWS クラウド の耐障害性」を参照してください。

リソースベースのポリシー

Amazon S3 バケット、エンドポイント、暗号化キーなどのリソースにアタッチされたポリシー。このタイプのポリシーは、アクセスが許可されているプリンシパル、サポートされているアクション、その他の満たすべき条件を指定します。

実行責任者、説明責任者、協業先、報告先 (RACI) に基づくマトリックス

移行活動とクラウド運用に関わるすべての関係者の役割と責任を定義したマトリックス。マトリックスの名前は、マトリックスで定義されている責任の種類、すなわち責任 (R) 、説明責任 (A) 、協議 (C) 、情報提供 (I) に由来します。サポート (S) タイプはオプションです。サポートが含まれる場合は RASCI マトリックスと呼ばれ、含まれない場合は RACI マトリックスと呼ばれます。

レスポンシブコントロール

有害事象やセキュリティベースラインからの逸脱について、修復を促すように設計されたセキュリティコントロール。詳細については、「 AWSでのセキュリティコントロールの実装」の「レスポンシブコントロール」を参照してください。

保持

「7 Rs」を参照してください。

廃止

「7 Rs」を参照してください。

検索拡張生成 (RAG)

生成 AI の技術。これにより、LLM では、レスポンスの生成前に、トレーニングデータソースの外部にある信頼できるデータソースが参照されます。例えば、RAG モデルによって、組織のナレッジベースまたはカスタムデータのセマンティック検索を実行できる場合があります。細については、「RAG (検索拡張生成) とは何ですか?」を参照してください。

ローテーション

定期的にシークレット情報を更新して、攻撃者が認証情報にアクセスするのをより困難にするプロセス。

行と列のアクセス制御 (RCAC)

アクセスルールが定義された、基本的で柔軟な SQL 表現の使用。RCAC は行権限と列マスクで構成されています。

RPO

「目標復旧時点」を参照してください。

RTO

「目標復旧時間」を参照してください。

ランブック

特定のタスクを実行するために必要な手動または自動化された一連の手順。これらは通常、エラー率の高い反復操作や手順を合理化するために構築されています。

S

SAML 2.0

多くの ID プロバイダー (IdP) が使用しているオープンスタンダード。この機能を使用すると、フェデレーティッドシングルサインオン (SSO) が有効になるため、ユーザーは組織内のすべてのユーザーを IAM で作成しなくても、 AWS マネジメントコンソール にログインしたり AWS 、API オペレーションを呼び出すことができます。SAML 2.0 ベースのフェデレーションの詳細については、IAM ドキュメントの「SAML 2.0 ベースのフェデレーションについて」を参照してください。

SCADA

「監視制御とデータ取得」を参照してください。

SCP

「サービスコントロールポリシー」を参照してください。

シークレット

暗号化された形式で保存する AWS Secrets Managerパスワードやユーザー認証情報などの機密情報または制限付き情報。シークレット値とそのメタデータで構成されます。シークレット値には、バイナリ、1 つの文字列、複数の文字列を指定できます。詳細については、Secrets Manager ドキュメントの「Secrets Manager シークレットの概要」を参照してください。

セキュリティバイデザイン

開発プロセス全体を通してセキュリティが考慮されているシステムエンジニアリングのアプローチ。

セキュリティコントロール

脅威アクターによるセキュリティ脆弱性の悪用を防止、検出、軽減するための、技術上または管理上のガードレール。セキュリティコントロールには、主に 4 つの種類があります。4 つとは、予防、検出、レスポンシブ、プロアクティブです。

セキュリティ強化

アタックサーフェスを狭めて攻撃への耐性を高めるプロセス。このプロセスには、不要になったリソースの削除、最小特権を付与するセキュリティのベストプラクティスの実装、設定ファイル内の不要な機能の無効化、といったアクションが含まれています。

Security Information and Event Management (SIEM) システム

セキュリティ情報管理 (SIM) とセキュリティイベント管理 (SEM) のシステムを組み合わせたツールとサービス。SIEM システムは、サーバー、ネットワーク、デバイス、その他ソースからデータを収集、モニタリング、分析して、脅威やセキュリティ違反を検出し、アラートを発信します。

セキュリティレスポンスの自動化

セキュリティイベントへの自動レスポンスまたは自動修復を目的として、事前定義およびプログラムされたアクション。これらの自動化は、セキュリティのベストプラクティスを実装するのに役立つ検出的または応答的な AWS セキュリティコントロールとして機能します。自動レスポンスアクションの例には、VPC セキュリティグループの変更、Amazon EC2 インスタンスへのパッチ適用、認証情報の更新などがあります。

サーバー側の暗号化

送信先で、それ AWS のサービス を受け取る によるデータの暗号化。

サービスコントロールポリシー (SCP)

AWS Organizationsの組織内の、すべてのアカウントのアクセス許可を一元的に管理するポリシー。SCP は、管理者がユーザーまたはロールに委任するアクションに、ガードレールを定義したり、アクションの制限を設定したりします。SCP は、許可リストまたは拒否リストとして、許可または禁止するサービスやアクションを指定する際に使用できます。詳細については、 AWS Organizations ドキュメントの「サービスコントロールポリシー」を参照してください。

サービスエンドポイント

のエントリポイントの URL AWS のサービス。ターゲットサービスにプログラムで接続するには、エンドポイントを使用します。詳細については、「AWS 全般のリファレンス」の「AWS のサービス エンドポイント」を参照してください。

サービスレベルアグリーメント (SLA)

サービスのアップタイムやパフォーマンスなど、IT チームがお客様に提供すると約束したものを明示した合意書。

サービスレベルインジケータ (SLI)

エラー率、可用性、スループットといった、サービスパフォーマンス面の指標。

サービスレベル目標 (SLO)

サービスレベルインジケータによって測定され、サービスの状態を表すターゲットメトリクス。

責任共有モデル

クラウドのセキュリティとコンプライアンス AWS について と共有する責任を説明するモデル。AWS はクラウドのセキュリティを担当しますが、 はクラウドのセキュリティを担当します。詳細については、「責任共有モデル」を参照してください。

SIEM

「Security Information and Event Management システム」を参照してください。

単一障害点 (SPOF)

特定のアプリケーションを構成する単一の重要なコンポーネントで発生し、システム稼働に支障をきたす可能性のある障害。

SLA

「サービスレベルアグリーメント」を参照してください。

SLI

「サービスレベルインジケータ」を参照してください。

SLO

「サービスレベルの目標」を参照してください。

スプリットアンドシードモデル

モダナイゼーションプロジェクトのスケーリングと加速のためのパターン。新機能と製品リリースが定義されると、コアチームは解放されて新しい製品チームを作成します。これにより、お客様の組織の能力とサービスの拡張、デベロッパーの生産性の向上、迅速なイノベーションのサポートに役立ちます。詳細については、「AWS クラウドでのアプリケーションをモダナイズするための段階的アプローチ」を参照してください。

SPOF

「単一障害点」を参照してください。

スタースキーマ

データベースの編成構造を意味し、1 つの大きいファクトテーブルにトランザクションデータまたは測定データが保存され、1 つ以上の小さいディメンションテーブルにデータ属性が保存されます。この構造は、データウェアハウスやビジネスインテリジェンスを用途とするように設計されています。

strangler fig パターン

レガシーシステムが廃止されるまで、システム機能を段階的に書き換えて置き換えることにより、モノリシックシステムをモダナイズするアプローチ。このパターンは、宿主の樹木から根を成長させ、最終的にその宿主を包み込み、宿主に取って代わるイチジクのつるを例えています。そのパターンは、モノリシックシステムを書き換えるときのリスクを管理する方法として Martin Fowler により提唱されました。このパターンの適用方法の例については、「コンテナと Amazon API Gateway を使用して、従来の Microsoft ASP.NET (ASMX) ウェブサービスを段階的にモダナイズ」を参照してください。

サブネット

VPC 内の IP アドレスの範囲。サブネットは、1 つのアベイラビリティーゾーンに存在する必要があります。

監視制御とデータ取得 (SCADA)

製造分野において、ハードウェアとソフトウェアを使用して物理アセットと本番運用をモニタリングするシステム。

対称暗号化

データの暗号化と復号に同じキーを使用する暗号化のアルゴリズム。

合成テスト

ユーザーとのやり取りをシミュレートして、起こり得る問題を検出したり、パフォーマンスをモニタリングしたりすることで、システムをテストします。Amazon CloudWatch Synthetics を使用すると、こうしたテストを作成できます。

システムプロンプト

コンテキスト、指示、ガイドラインなどを提示して、LLM に動作を指示する手法。システムプロンプトは、コンテキストを設定して、ユーザーとやり取りするルールを確立するのに有用です。

T

タグ

AWS リソースを整理するためのメタデータとして機能するキーと値のペア。タグは、 リソースの管理、識別、整理、検索、フィルタリングに役立ちます。詳細については、「AWS リソースのタグ付け」を参照してください。

ターゲット変数

監督された機械学習でお客様が予測しようとしている値。これは、結果変数のことも指します。例えば、製造設定では、ターゲット変数が製品の欠陥である可能性があります。

タスクリスト

ランブックの進行状況を追跡するために使用されるツール。タスクリストには、ランブックの概要と完了する必要のある一般的なタスクのリストが含まれています。各一般的なタスクには、推定所要時間、所有者、進捗状況が含まれています。

テスト環境

「環境」を参照してください。

トレーニング

お客様の機械学習モデルに学習するデータを提供すること。トレーニングデータには正しい答えが含まれている必要があります。学習アルゴリズムは入力データ属性をターゲット (お客様が予測したい答え) にマッピングするトレーニングデータのパターンを検出します。これらのパターンをキャプチャする機械学習モデルを出力します。そして、お客様が機械学習モデルを使用して、ターゲットがわからない新しいデータでターゲットを予測できます。

トランジットゲートウェイ

VPC とオンプレミスネットワークを相互接続するために使用できる、ネットワークの中継ハブ。詳細については、 AWS Transit Gateway ドキュメントの「トランジットゲートウェイとは」を参照してください。

トランクベースのワークフロー

デベロッパーが機能ブランチで機能をローカルにビルドしてテストし、その変更をメインブランチにマージするアプローチ。メインブランチはその後、開発環境、本番前環境、本番環境に合わせて順次構築されます。

信頼されたアクセス

ユーザーに代わって AWS Organizations およびそのアカウントで組織内でタスクを実行するために指定したサービスにアクセス許可を付与します。信頼されたサービスは、サービスにリンクされたロールを必要なときに各アカウントに作成し、ユーザーに代わって管理タスクを実行します。詳細については、 ドキュメントの「Using AWS Organizations with other AWS services AWS Organizations 」を参照してください。

チューニング

機械学習モデルの精度を向上させるために、お客様のトレーニングプロセスの側面を変更する。例えば、お客様が機械学習モデルをトレーニングするには、ラベル付けセットを生成し、ラベルを追加します。これらのステップを、異なる設定で複数回繰り返して、モデルを最適化します。

ツーピザチーム

2 枚のピザを分け合えることができるくらい小さな DevOps チーム。ツーピザチームの規模では、ソフトウェア開発におけるコラボレーションに最適な機会が確保されます。

U

不確実性

予測機械学習モデルの信頼性を損なう可能性がある、不正確、不完全、または未知の情報を指す概念。不確実性には、次の 2 つのタイプがあります。認識論的不確実性は、限られた、不完全なデータによって引き起こされ、弁論的不確実性 は、データに固有のノイズとランダム性によって引き起こされます。

未分化なタスク

ヘビーリフティングとも呼ばれ、アプリケーションの作成と運用には必要だが、エンドユーザーに直接的な価値をもたらさなかったり、競争上の優位性をもたらしたりしない作業です。未分化なタスクの例としては、調達、メンテナンス、キャパシティプランニングなどがあります。

上位環境

「環境」を参照してください。

V

バキューミング

ストレージを再利用してパフォーマンスを向上させるために、増分更新後にクリーンアップを行うデータベースのメンテナンス操作。

バージョンコントロール

リポジトリ内のソースコードへの変更など、変更を追跡するプロセスとツール。

VPC ピアリング

プライベート IP アドレスを使用してトラフィックをルーティングできる、2 つの VPC 間の接続。詳細については、Amazon VPC ドキュメントの「VPC ピア機能とは」を参照してください。

脆弱性

システムのセキュリティを脅かすソフトウェアまたはハードウェアの欠陥。

W

ウォームキャッシュ

頻繁にアクセスされる最新の関連データを含むバッファキャッシュ。データベースインスタンスはバッファキャッシュから、メインメモリまたはディスクからよりも短い時間で読み取りを行うことができます。

ウォームデータ

アクセス頻度の低いデータ。この種類のデータをクエリする場合、通常は適度に遅いクエリでも問題ありません。

ウィンドウ関数

現在のレコードに何らかの形で関連している行のグループに計算を実行する SQL 関数。ウィンドウ関数は、移動平均を計算したり、現在の行の相対位置に基づいて他の行の値にアクセスするといったタスクの処理に役立ちます。

ワークロード

ビジネス価値をもたらすリソースとコード (顧客向けアプリケーションやバックエンドプロセスなど) の総称。

ワークストリーム

特定のタスクセットを担当する移行プロジェクト内の機能グループ。各ワークストリームは独立していますが、プロジェクト内の他のワークストリームをサポートしています。たとえば、ポートフォリオワークストリームは、アプリケーションの優先順位付け、ウェーブ計画、および移行メタデータの収集を担当します。ポートフォリオワークストリームは、これらの設備を移行ワークストリームで実現し、サーバーとアプリケーションを移行します。

WORM

「Write-Once-Read-Many」を参照してください。

WQF

「AWS ワークロード資格フレームワーク」を参照してください

Write-Once-Read-Many (WORM)

データを 1 回のみ書き込むことで、データの削除や変更を防ぐストレージモデル。承認済みユーザーは、必要な回数だけデータを読み取ることができますが、変更することはできません。このデータストレージインフラストラクチャは、イミュータブルと見なされます。

Z

ゼロデイエクスプロイト

ゼロデイ脆弱性を悪用した攻撃 (一般的にマルウェアによる)。

ゼロデイ脆弱性

実稼働システムにおける未解決の欠陥または脆弱性。脅威アクターは、このような脆弱性を利用してシステムを攻撃する可能性があります。開発者は、よく攻撃の結果で脆弱性に気付きます。

ゼロショットプロンプト

LLM にタスク実行の手順は提示するが、実行のガイドとして役立つ例 (ショット) は提示しない方法。LLM は、事前トレーニング済みの知識を使用してタスクを処理する必要があります。ゼロショットプロンプトの有効性は、タスクの複雑さとプロンプトの品質によって異なります。「数ショットプロンプト」も参照してください。

ゾンビアプリケーション

平均 CPU およびメモリ使用率が 5% 未満のアプリケーション。移行プロジェクトでは、これらのアプリケーションを廃止するのが一般的です。

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。