Amazon RDS for MySQL と MariaDB のモニタリングと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 のモニタリングと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 のモニタリングと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は、データベースの可用性、パフォーマンス、および機能を測定、追跡、評価するプロセスです。モニタリングおよびアラートソリューションは、組織がデータベースサービス、ひいては関連するアプリケーションとワークロードが安全で、高性能で、回復力があり、効率的であることを確認するのに役立ちます。では AWS、ワークロードの状態を把握し、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して、リソースが期待どおりに動作していることを確認し、顧客に影響を与える前に問題を検出して修正できます。しきい値を超えたときにアラームを発生させるには、モニタリングするメトリクス、ログ、イベント、トレースを使用する必要があります。

このガイドでは、Amazon Relational Database Service (Amazon RDS) データベースのデータベースオブザーバビリティおよびモニタリングツールとベストプラクティスについて説明します。このガイドでは、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されます。

このガイドは、 で実行されてい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、管理を行う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、DBAs、上級 DevOps エンジニア、およびその他のチームメンバーを対象としています AWS クラウド。

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 つの柱に含まれています。

- 運用上の優秀性の柱では、ワークロードをテレメトリと monitoring. AWS services を含むように設計する必要があります。Amazon Relational Database Service (Amazon RDS) などのサービスは、ワークロードの内部状態 (メトリクス、ログ、イベント、トレースなど) を理解するために必要な情報を提供します。Amazon RDS データベースを運用するときは、データベースインスタンスの状態を把握し、運用イベントを検出し、計画されたイベントと計画外のイベントの両方に対応できるようにする必要があります。 AWS には、組織とビジネスの成果がいつリスクにさらされているか、またはリスクにさらされている可能性があるかを判断するのに役立つモニタリングツールが用意されているため、適切なアクションを適切なタイミングで実行できます。
- パフォーマンス効率の柱では、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、処理することで、Amazon RDS DB インスタンスなどのリソースのパフォーマンスをモニタリングすることを規定しています。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など、パフォーマンスの低下を特定し、その原因を修復できます。測定値が想定範囲外になると、アラームを自動的に生成できます。通知だけでなく、検出され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ことをお勧めします。収集したメトリクスを事前定義されたしきい値と照らし合わせて評価したり、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できます。たとえば、CPU 使用率の増加傾向を検出するには、一定期間にわたって`cpuUtilization.total`メトリクスを収集して分析できます。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アラートすることで、顧客に影響を与える前に問題を修正できます。
- 信頼性の柱は、可用性の要件を満たすためにモニタリングとアラートを重要と定義します。モニタリングソリューションは、障害を効果的に検出できる必要があります。問題や障害を検出する場合、その主な目的はそれらの問題を警告することです。クラウド内の回復力のあるアーキテクチャには、継続的なオブザーバビリティとモニタリングのプラクティスを実装することが不可欠です。ワークロードを改善するには、ワークロードを測定し、ワークロードの状態と状態を理解できる必要があります。障害からの自動復旧、水平スケーラビリティ、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によって異なります。
- セキュリティの柱では、予期しない設定変更や不要な設定変更、および予期しない動作の検出と防止について説明します。MariaDB 監査プラグインを使用して Amazon RDS for MySQL および MariaDB DB インスタンスを設定し、ユーザーログインやデータベースに対して実行される特定のオペレーションなどのデータベースアクティビティを記録できます。 MariaDB プラグインは、データベースアクティビティの記録をログファイルに保存します。ログファイルは、統合してモニタリングおよびアラートツールにインポートできます。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についてリアルタイムで分析されます。このような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ことを示している可能性があり、ビジネスに潜在的なリスクが及ぶ可能性があります。モニタリングツールがこのようなイベントを検出すると、アラームがアクティブ化されてセキュリティインシデントへの応答が開始され、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するのに役立ちます。

ターゲットを絞ったビジネス成果

モニタリングおよびアラートメカニズムのベストプラクティスを実装することで、アプリケーションとワークロードに対して、高パフォーマンス、回復力、効率、安全性、コスト最適化のインフラストラクチャを確保できます。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、視覚化するオブザーバビリティツールを使用して、データベースのヘルスとパフォーマンスの全体像を監視および分析し、関連する IT サービスの低下や中断を防ぐことができます。計画外の機能低下やサービスの中断が続く場合、モニタリングおよびアラートツールは、問題、エスカレーション、反応、迅速な調査と解決をタイムリーに検出するのに役立ちます。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、以下のビジネス成果を達成するのに役立ちます。

- カスタマーエクスペリエンスを向上させます。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。データベースは、多くの場合、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、支払い、business-to-business (B2B) APIs、統合サービスなどのデジタルサービスの主要なコンポーネントです。データベースでアラートをモニタリングして設定し、問題を迅速に検出して効率的に調査し、できるだけ早く修正してダウンタイムやその他の中断を最小限に抑えることができれば、顧客のデジタルサービスの可用性、セキュリティ、パフォーマンスを向上させることができます。
- 顧客の信頼を構築する。 パフォーマンスの向上とユーザーエクスペリエンスの円滑化により、顧客の信頼を獲得できるため、プラットフォームでのビジネスが増える可能性があります。例えば、信頼性の高いオンラインサービスを提供する支払い処理サービスプロバイダーは、顧客の信頼と忠誠心が高いことを期待できます。これにより、顧客が増え、保持率が向上し、請求可能な取引が増加し、新しい革新的なサービスによって収益が増えます。
- 財務上の損失を回避します。 データベースインフラストラクチャの予期しないダウンタイムは、アプリケーションを使用して顧客が実行するビジネストランザクションに影響を与える可能性があります。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 (SLAs) に違反すると、顧客の信頼が失われ、結果として収益が失われる可能性があります。また、顧客がお客様の責任と保証契約に基づいて補償を要求する可能性のある、高額なトライアルの法的根拠となる場合もあります。ソフトウェア会社である Atlassian Corporation の調査によ

ると、サービス停止の平均コストは、ビジネスのタイプと規模に応じて、1 時間あたり 140,000 USD ~ 540,000 USD の範囲です。安定したデータベース環境は、長時間の停止やビジネスの損失を防ぐ上で重要です。

- 値を展開します。モニタリングとアラートのメカニズムは、可用性、回復力、信頼性、パフォーマンス、コスト効率、安全性の高いデジタルサービスを設計、開発、運用するのに役立ちますが、これはほんの始まりにすぎません。組織が時間の経過とともにスケーリングと拡張を行い、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ことが必要になります。新しい サービスは、顧客にさらなる価値を提供し、ビジネスにより多くの収益をもたらし、ビジネスの成長にフライホイール効果をもたらします。
- 開発者の生産性を向上させます。 生産的で効率的で、開発タスクで問題やボトルネックに遭遇しない開発者は、高品質の製品を短時間で提供できます。ただし、ソフトウェアエンジニアリングと IT 運用には複雑な課題がよくあります。ワークロードとそのアーキテクチャの規模によっては、この複雑さが増します。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、開発者に相関メトリクスとトレースを提供できるツールが必要です。これらは、欠陥のあるコードアーティファクトとインフラストラクチャコンポーネントをできるだけ早く特定し、エンドユーザーへの影響を判断するのに役立ちます。適切なモニタリングツールとアラートツールスイートは、デベロッパーによるコード作成とテストを迅速かつ適切に行うことができます。
- 運用の有効性と効率を向上させます。 クラウドワークロードを大規模に運用する場合、パフォーマンス向上のごく一部であっても、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ことで、将来の容量ニーズを理解して予測し、 で利用できるコスト削減を活用できます AWS クラウド。Amazon RDS ワークロードと運用状態を理解することは、イベントへの対応、問題の修正、改善の計画に役立ちます。

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 ワークロードの状態を十分に把握し、運用イベントやモニタリングデータに応じて適切なアクションを実行するのに役立ちます。

- KPIs を特定します。期待されるビジネス成果に基づいて、主要業績評価指標 (KPIsを特定します。KPI KPIs。例えば、コアビジネスが e コマースの場合、希望するビジネス成果の 1 つは、顧客が e ショップを 24 時間 365 日利用できることです。このビジネス成果を達成するには、e ショッピングアプリケーションが使用するバックエンド Amazon RDS データベースの可用性 KPI を定義し、ベースライン KPI を毎週 99.99% に設定します。ベースライン値に対する実際の可用性 KPI を評価することで、希望するデータベースの可用性を 99.99% 満たしているかどうかを判断し、24 時間 365 日サービスを提供することでビジネス成果を達成できます。
- ワークロードメトリクスを定義します。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、ワークロードが望ましい成果を達成しているかどうかを判断し、ワークロードの状態を把握します。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 インスタンスの稼働時間とダウンタイムなどのメトリクスを測定する必要があります。その後、これらのメトリクスを使用して、可用性 KPI を次のように計算できます。

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

  メトリクスは、時系列のデータポイントのセットを表します。メトリクスには、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します。Amazon RDS は、設定に応じてさまざまなメトリクスとログを生成します。これらの一部は、DB インスタンスイベント、カウンター、または などの統計を表しますdb.Cache.innoDB_buffer_pool_hits。その他のメトリクスは、ホスト Amazon Elastic Compute Cloud (Amazon EC2) インスタンスの合計メモリ量memory.Totalを測定する などのオペレーティングシステムから取得されます。モニタリングツールは、収集されたメトリクスを定期的に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する必要があります。
- ワークロードメトリクスのベースラインを確立します。メトリクスのベースラインを確立して、想定値を定義し、良いしきい値と悪い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では、 のベースラインReadIOPSを最大 1,000 に定義できます。その後、このベースラインを使用して比較し、過剰使用率を特定できます。新しいメトリクスで読み取り IOPS が 2,000 ~ 3,000 の範囲にあることが一貫して示されている場合は、調査、介入、改善のための対応をトリガーする可能性のある逸脱を特定しました。

- ワークロードの成果がリスクにさらされているときにアラートします。ビジネス成果がリスクにさらされていると判断した場合は、アラートを発行します。その後、顧客に影響を与える前に問題を事前に対処するか、インシデントの影響をタイムリーに軽減できます。
- ワークロードのアクティビティの予想されるパターンを特定します。メトリクスのベースラインに基づいて、ワークロードアクティビティのパターンを確立して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で対応します。 AWS には、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、異常を検出するモニタリングツールが用意されています。
- ワークロードの異常が検出されたときに警告します。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ときは、必要に応じて適切なアクションで対応できるように警告を発します。
- KPIs。Amazon RDS データベースが定義された要件を満たしていることを確認し、ビジネス目標を達成するための潜在的な改善点を特定します。測定されたメトリクスと評価された KPIs の有効性を検証し、必要に応じて修正します。例えば、同時データベース接続の最適な数に KPI を設定し、試行された接続と失敗した接続、および作成されて実行中のユーザースレッドに関するメトリクスをモニタリングするとします。KPI ベースラインで定義されている接続よりも多くのデータベース接続がある場合があります。現在のメトリクスを分析することで、結果を検出できますが、根本原因を特定できない場合があります。その場合は、メトリクスを修正し、テーブルロックのカウンターなどの追加のモニタリング対策を含める必要があります。新しいメトリクスは、データベース接続の数の増加が予期しないテーブルロックによって引き起こされているかどうかを判断するのに役立ちます。

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、次の操作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いた意思決定を行う
- メトリクスとログを分析して潜在的な問題をプロアクティブに予測する
- ユーザーが影響を受ける前に問題をトラブルシューティングして解決するために、しきい値を超えたときにアラートを生成する

AWS提供のクラウドネイティブなオブザーバビリティとモニタリングツールとサービス、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ための商用サードパーティーソリューションなど、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。これらのツールの一部については、以下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、各ツールの機能と機能を組織の要件と比較します。また、デプロイのしやすさ、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、価格、および組織に固有のその他の要因についてツールを評価することをお勧めします。

セクション

Amazon RDS に含まれるツール

Amazon Relational Database Service (Amazon RDS) は、 のマネージドデータベースサービスです AWS クラウド。Amazon RDS はマネージドサービスであるため、データベースのバックアップ、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、データセンターのオペレーションなど、ほとんどの管理タスクから解放されます。 には、Amazon RDS DB インスタンスの完全なオブザーバビリティソリューションを構築できる包括的なツールセット AWS も用意されています。

一部のモニタリングツールは、Amazon RDS サービスに含まれ、事前設定され、自動的に有効になります。新しい Amazon RDS インスタンスを起動するとすぐに、次の 2 つの自動ツールを使用できます。

- Amazon RDS インスタンスのステータスは、DB インスタンスの現在の状態に関する詳細を提供します。たとえば、ステータスコードには、使用可能、停止、作成、バックアップ、失敗が含まれます。Amazon RDS コンソール、 AWS Command Line Interface （ AWS CLI ）、または Amazon RDS API を使用して、インスタンスのステータスを確認できます。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 レコメンデーションは、DB インスタンス、リードレプリカ、および DB パラメータグループに対する自動レコメンデーションを提供します。これらの推奨事項は、DB インスタンスの使用状況、パフォーマンスデータ、および設定を分析することで提供され、ガイダンスとして提供されます。例えば、エンジンバージョンの古い推奨事項では、DB インスタンスがデータベースソフトウェアの最新バージョンを実行していないこと、および最新のセキュリティ修正やその他の改善点を活用できるように DB インスタンスをアップグレードする必要が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 レコメンデーションの表示」を参照してください。

CloudWatch 名前空間

Amazon RDS は Amazon Amazon CloudWatch と統合されます。Amazon CloudWatch は、 で実行され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です AWS。Amazon RDS 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、イベントを自動的に収集し、長期保存、分析、アラートのために CloudWatch に送信します。

Amazon RDS for MySQL および Amazon RDS for MariaDB は、追加料金なしでデフォルトのメトリクスセットを 1 分間隔で CloudWatch に自動的に発行します。これらのメトリクスは、メトリクスのコンテナである 2 つの名前空間に収集されます。

- AWS/RDS 名前空間には、DB インスタンスレベルのメトリクスが含まれています。例としては、`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 名前空間には、Amazon RDS サービスクォータ内で運用されているかどうかを判断するために使用されるアカウントレベルの使用状況メトリクスが含まれています。例としては、 `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で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。

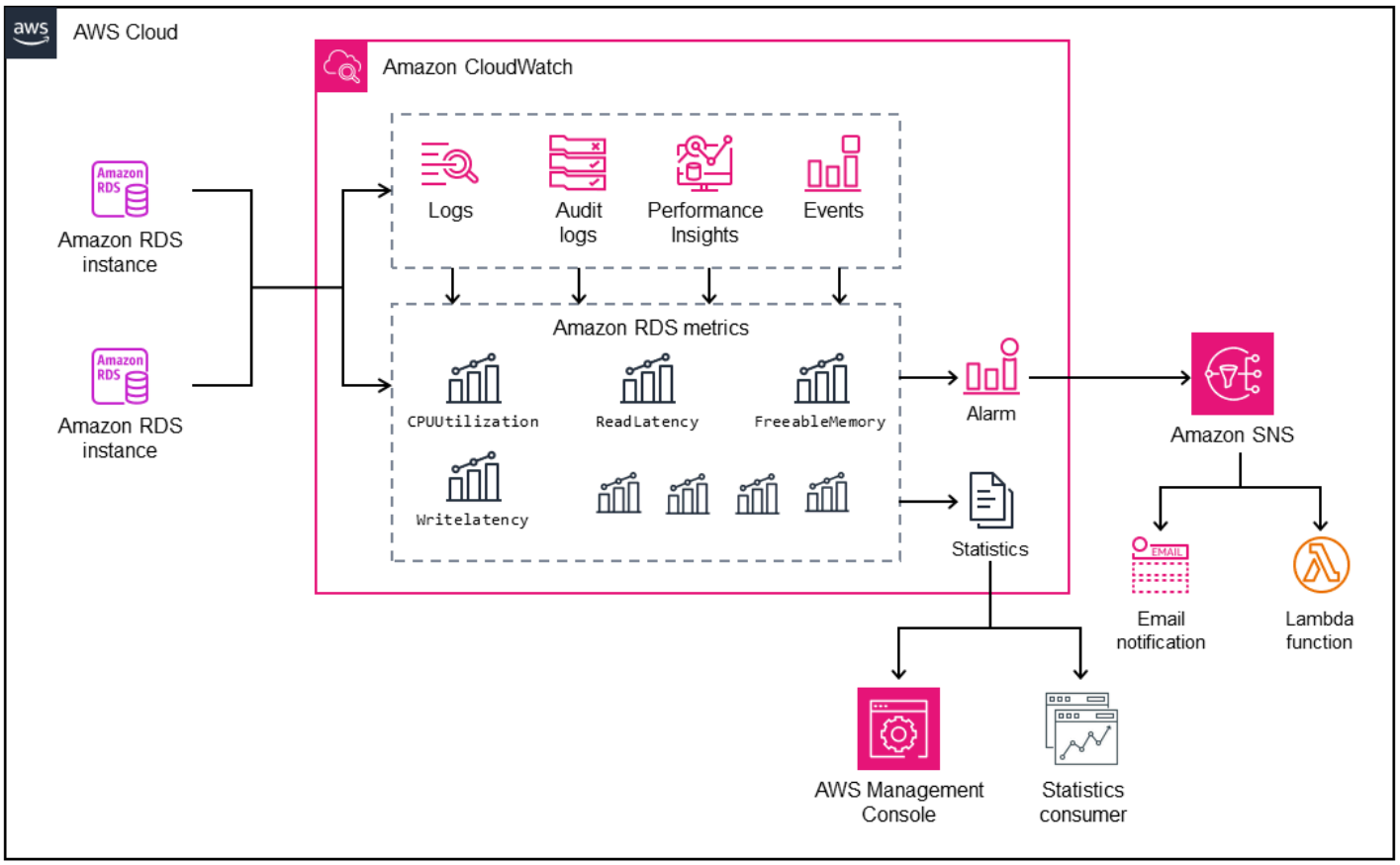
CloudWatch では、次のようにメトリクスデータを保持します。

- 3 時間: 期間が 60 秒未満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: 期間が 60 秒 (1 分) 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) 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: 期間 3,600 秒 (1 時間) のデータポイントは 15 か月 (455 日) 利用できます。

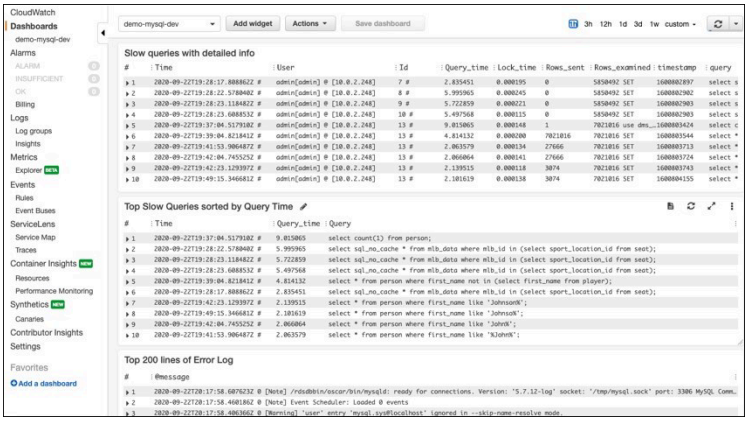
詳細については、CloudWatch ドキュメントの「メトリクス」を参照してください。

CloudWatch アラームとダッシュボード

Amazon CloudWatch アラームを使用して、特定の Amazon RDS メトリクスを一定期間監視できます。たとえば、 をモニタリングし`FreeStorageSpace`、メトリクスの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 に設定し、空きストレージ容量が 200 MB (しきい値未満) の場合、アラームはアクティブ化され、Amazon RDS DB インスタンスに追加のストレージを自動的にプロビジョニングするアクションをトリガーできます。アラームは、Amazon Simple Notification Service (Amazon SNS) を使用して DBA に通知 SMS を送信することもできます。次の図は、このプロセスを示したものです。

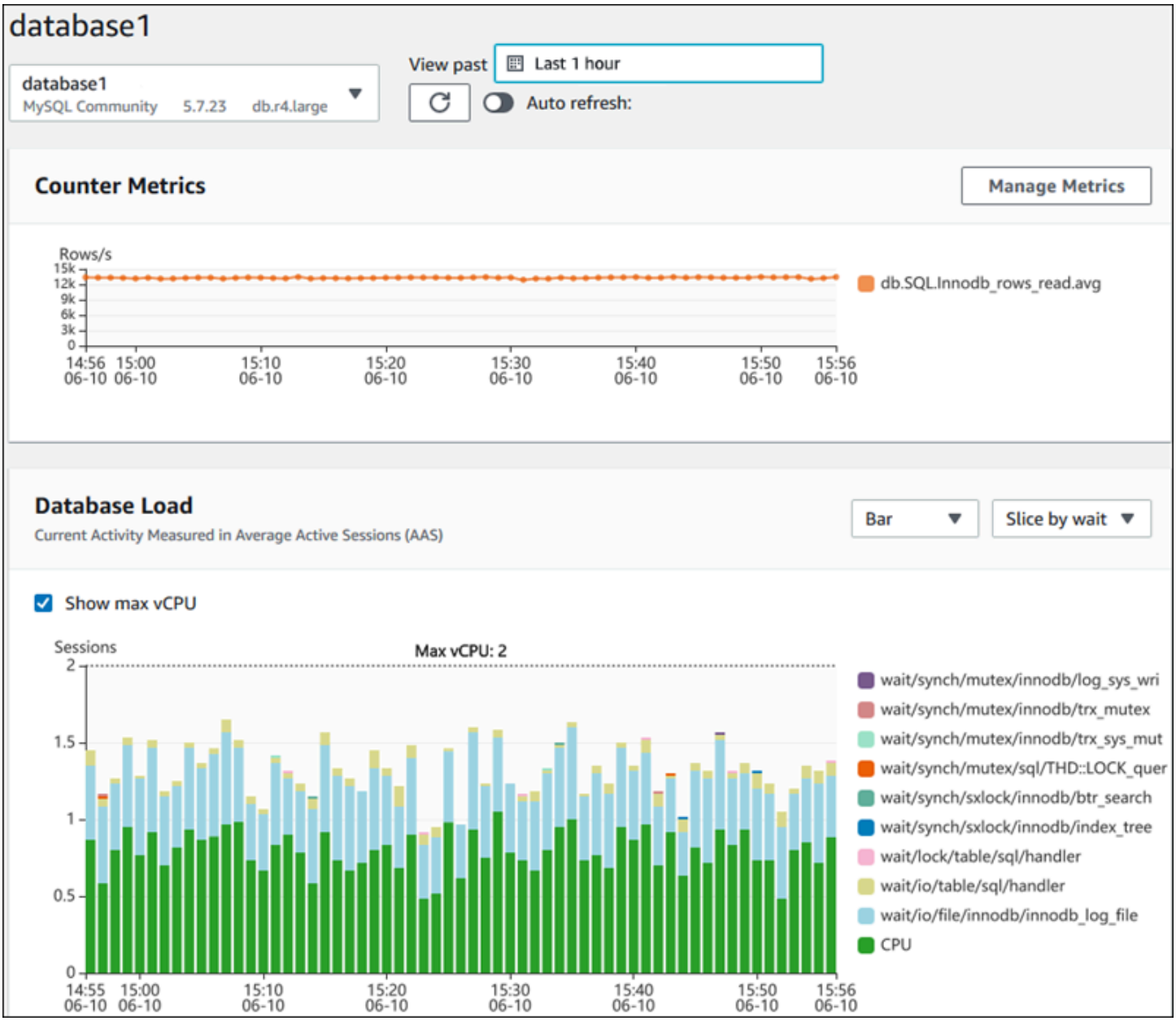


CloudWatch には、メトリクスのカスタマイズされたビュー (グラフ) を作成、カスタマイズ、操作、保存するために使用できるダッシュボードも用意されています。CloudWatch Logs Insights を使用して、スロークエリログとエラーログをモニタリングするためのダッシュボードを作成し、それらのログで特定のパターンが検出された場合にアラートを受信することもできます。次の画面は、CloudWatch ダッシュボードの例を示しています。



Amazon RDS Performance Insights

Amazon RDS Performance Insights は、Amazon RDS モニタリング機能を拡張するデータベースパフォーマンスのチューニングおよびモニタリングツールです。DB インスタンスの負荷を視覚化し、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することで、データベースのパフォーマンスを分析するのに役立ちます。このツールは、ロック待機、高い CPU 消費量、I/O レイテンシーなど、DB インスタンスが持つ可能性のあるボトルネックのタイプを特定し、ボトルネックの原因となっている SQL ステートメントを決定するのに役立つ複数のメトリクスを 1 つのインタラクティブグラフに結合します。次の画面は、視覚化の例を示しています。



アカウント内の 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ます。必要に応じて、より長い保持期間を購入できます。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 は CloudWatch にメトリクスを自動的に発行します。Performance Insights ツールの使用に加えて、CloudWatch が提供する追加機能を活用できます。Performance Insights メトリクスは、CloudWatch コンソール、 AWS CLI、または CloudWatch API を使用して調べることができます。他のメトリクスと同様に、CloudWatch アラームを追加することもできます。例えば、DBAs への SMS 通知をトリガーしたり、DBLoadメトリクスが設定した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 メトリクスを既存の 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 インスタンスが実行されてい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キャプチャするツールです。これらのメトリクスは、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 データなどに最大 1 秒の詳細度を提供します。これらのメトリクスには、Amazon RDS コンソールでアクセスして分析できます。Performance Insights と同様に、拡張モニタリングメトリクスは Amazon RDS から CloudWatch に配信されます。ここでは、分析用のメトリクスの長期保存、メトリクスフィルターの作成、CloudWatch ダッシュボードへのグラフの表示、アラームの設定などの追加機能を利用できます。デフォルトでは、新しい Amazon RDS DB インスタンスを作成すると、拡張モニタリングは無効になります。DB インスタンスを作成または変更するときに、この機能を有効にできます。料金は、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います。拡張モニタリングが有効になっている DB インスタンスの詳細度と数に応じて、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 の料金」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 AWS サービス

AWS は、Amazon RDS および CloudWatch とも統合されているいくつかのサポートサービスを提供し、データベースのオブザーバビリティをさらに強化します。これには、Amazon EventBridge、Amazon CloudWatch Logs、 が含まれます AWS CloudTrail。

- Amazon EventBridge は、Amazon RDS DB インスタンスを含むアプリケーションと AWS リソースからイベントを受信、フィルタリング、変換、ルーティング、配信できるサーバーレスイベントバスです。Amazon RDS イベントは、Amazon RDS 環境の変更を示します。たとえば、DB インスタンスのステータスが Available から Stopped に変更されると、Amazon RDS はイベント を生成しますRDS-EVENT-0087 / The DB instance has been stopped。Amazon RDS は、ほぼリアルタイムで CloudWatch Events と EventBridge にイベントを配信します。EventBridge と CloudWatch Events を使用して、対象となる特定の Amazon RDS イベントにアラートを送信し、イベントがルールに一致するときに実行するアクションを自動化するルールを定義できます。修正アクションを実行できる AWS Lambda 関数や、DBAs または DevOps エンジニアにイベントを通知する E メールまたは SMS を送信できる Amazon SNS トピックなど、イベントに応じてさまざまなターゲットを使用できます。
- Amazon CloudWatch Logs は、Amazon RDS for MySQL や MariaDB DB インスタンス、 など、すべてのアプリケーション、システム、サービスからのログファイルのストレージを一元化する AWS サービスです AWS CloudTrail。DB インスタンスでこの機能を有効にすると、Amazon RDS は CloudWatch Logs に次のログを自動的に発行します。
  - エラーログ
  - スロークエリログ
  - 全般ログ
  - [監査ログ]

  CloudWatch Logs Insights を使用して、ログデータのクエリと分析を行うことができます。この機能には、定義したパターンに一致するログイベントの検索に役立つ専用のクエリ言語が含まれています。たとえば、MySQL DB インスタンスのテーブルの破損を追跡するには、エラーログファイルをモニタリングしてパターン を確認します"ERROR 1034 (HY000): Incorrect key file for table '*'; try to repair it OR Table * is marked as crashed"。フィルタリングされたログデータは CloudWatch メトリクスに変換できます。その後、メトリクスを使用してグラフまたは表形式のデータを含むダッシュボードを作成したり、定義されたしきい値を超えた場合にアラームを設定したりできます。これは、予期しない動作や疑わしい動作が検出された場合に自動的にモニタリング、アラートを送信し、修正措置を講じることができるため、監査ログを使用する場合に特に便利です。 AWS マネジメントコンソール、、Amazon RDS API AWS CLI、

または AWS SDK for CloudWatch Logs を使用して、データベースログにアクセスして管理できます。

- AWS CloudTrail は、 のユーザーおよび API 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 AWS アカウント。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。すべてのアクションを記録でき、CloudTrail は 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提供します。たとえば、ユーザーが新しい Amazon RDS DB インスタンスを作成すると、イベントが検出され、口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"}`) などに関する情報が含まれます。CloudTrail によって口グに記録されるイベントには、Amazon RDS コンソールからの呼び出しと、Amazon RDS API を使用するコードからの呼び出しの両方が含まれます。

サードパーティーのモニタリングツール

一部のシナリオでは、Amazon RDS に AWS が提供す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ツールの完全なスイートに加えて、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。このようなシナリオには、オンプレミスデータセンターで多数のデータベースが実行されているハイブリッドデプロイと、 で別のデータベースセットが実行されている場合があります AWS クラウド。すでに企業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 AWS クラウド デプロイに拡張することをお勧めします。サードパーティーのモニタリングソリューションを設定する際の課題は、多くの場合、Amazon RDS がクラウドマネージドサービスとして課す保護にあります。たとえば、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 インスタンスを実行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、CloudWatch やその他の AWS クラウド サービス上に構築することで、多くのサードパーティーのモニタリングソリューションを Amazon RDS と統合できます。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、サードパーティーのモニタリングツールにインポートして、さらなる分析、可視化、アラートを行うことができます。これらのサードパーティーソリューションには、Prometheus、Grafana、Percona などがあります。

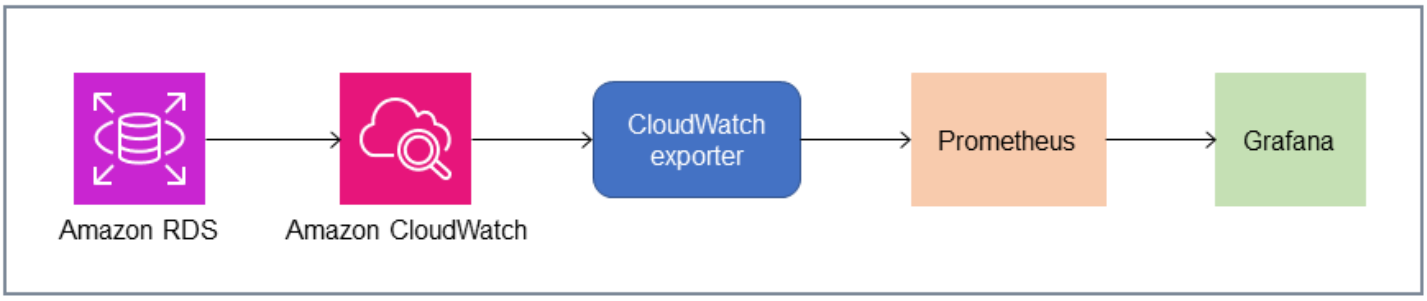
Prometheus と Grafana

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す。これは、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す。Amazon RDS DB インスタンスをモニタリングすると、CloudWatch は Amazon RDS からメトリクスを収集します。次に、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、メトリクスを Prometheus サーバーにエクスポートします。

- YACE エクスポーターは、CloudWatch API への 1 回のリクエストで複数のメトリクスを取得することで、データエクスポートタスクを最適化します。メトリクスが Prometheus サーバーに保存されると、サーバーはルール式を評価し、指定された条件が観察されたときにアラートを生成できます。
- CloudWatch Exporter は Prometheus によって公式に管理されています。CloudWatch API を介して CloudWatch メトリクスを取得し、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 Prometheus サーバーに保存します。

エクスポーターを選択し、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は、CloudWatch および CloudWatch Logs サービスと API クォータを検討してください。これは、CloudWatch メトリクスの Prometheus サーバーへのエクスポートが CloudWatch API 上に実装されるためです。例えば、CloudWatch Exporter の複数のインスタンスを 1 つの リージョン AWS アカウント と リージョンにデプロイして数百の 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。このような制限を克服するには、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た YACE エクスポーターの使用を検討してください。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 1 つの に一元化し、それぞれのエクスポーターインスタンスの数を制限するのではなく AWS アカウント、複数の AWS アカウント を使用することを検討する必要があります AWS アカウント。

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。このツールは、E メール、SMS、Slack などの正しい受信者へのアラートの重複排除、グループ化、ルーティング、または自動応答アクションの開始を処理します。Grafana と呼ばれる別のオープンソースツールでは、これらのメトリクスの視覚化が表示されます。Grafana は、高度なグラフ、動的ダッシュボード、アドホッククエリや動的ドリルダウンなどの分析機能など、豊富な視覚化ウィジェットを提供します。また、ログを検索および分析したり、メトリクスとログを継続的に評価するためのアラート機能を含めたり、データがアラートルールに一致すると通知を送信したりすることもできます。



パーコナ

Percona Monitoring and Management (PMM) は、MySQL および MariaDB 用の無料のオープンソース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す。PMM は、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します。ダッシュボード内のデータを視覚化するウェブ UI と、データベースのヘルス評価の自動アドバイザーなどの追加機能を提供します。PMM を使用して Amazon RDS をモニタリングできます。ただし、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となるホストにはインストールされません。代わりに、このツールは Amazon RDS DB インスタンスに接続し、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) が必要で、モニタリングに使用できる Amazon RDS DB インスタンスを自動的に検出します。PMM ツールはデータベースモニタリング用にプロファイリングされ、Prometheus よりも多くのデータベース固有のメトリクスを収集します。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。これは、クエリ分析エージェントが Amazon RDS 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ることができないためです。代わりに、MySQL および MariaDB DB インスタンス`performance_schema`から直接 をクエリしてメトリクスを取得します。PMM の主な機能の 1 つは、ツールがデータベース内で特定した問題について DBAs に警告し、助言する機能です。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が用意されています。

これらのツールに加えて、Amazon RDS と統合できる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が市場にいくつかあります。例としては、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 インスタンスは、Amazon RDS の基本的な構成要素です。クラウドで実行される分離されたデータベース環境です。MySQL および MariaDB データベースの場合、DB インスタンスは MySQL サーバーとも呼ばれる mysqld プログラムです。これには、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などの複数のスレッドとコンポーネントが含まれます。各ストレージエンジンは、特殊なユースケースをサポートするように設計されています。デフォルトおよび推奨されるストレージエンジンは InnoDB です。InnoDB は、アトミック性、一貫性、分離性、耐久性 (ACID) モデルに準拠したトランザクション、汎用、リレーショナルデータベースエンジンです。InnoDB は、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、元に戻すログ、REDO ログ、二重書き込みバッファファイル) を備えています。データベースが ACID モデルに厳密に準拠するように、InnoDB ストレージエンジンは、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) など、データを保護するための多数の機能を実装します。

DB インスタンスのこれらの内部コンポーネントはすべて、データの可用性、整合性、セキュリティを期待どおりの十分なパフォーマンスレベルで維持するために共同で機能します。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にリソース需要を課す場合があります。特定のリソースの需要の急増が、プロビジョニングされた容量またはリソースのソフトウェア制限 (設定パラメータまたはソフトウェア設計によって課される) 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、完全に利用できなくなったり破損し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、定義されたベースライン値と比較し、モニタリングされ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重要です。

前述のようにツールを使用して 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。これらのツールは Amazon RDS と統合され、高解像度メトリクスを収集し、最新のパフォーマンス情報をほぼリアルタイムで提示し、アラームを生成するため、モニタリングとアラートには Amazon RDS Performance Insights と CloudWatch ツールを使用することをお勧めします。

任意のモニタリングツールに関係なく、MySQL および MariaDB DB インスタンスで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ための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限に抑えるように設計されています。この機能は、

performance_schemaパラメータを使用して管理できます。このパラメータはオプションですが、Amazon RDS Performance Insights によって収集される高解像度 (1 秒) の SQL ごとのメトリクス、アクティブなセッションメトリクス、待機イベント、その他の詳細で低レベルのモニタリング情報を収集するために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 メトリクスを CloudWatch に発行する

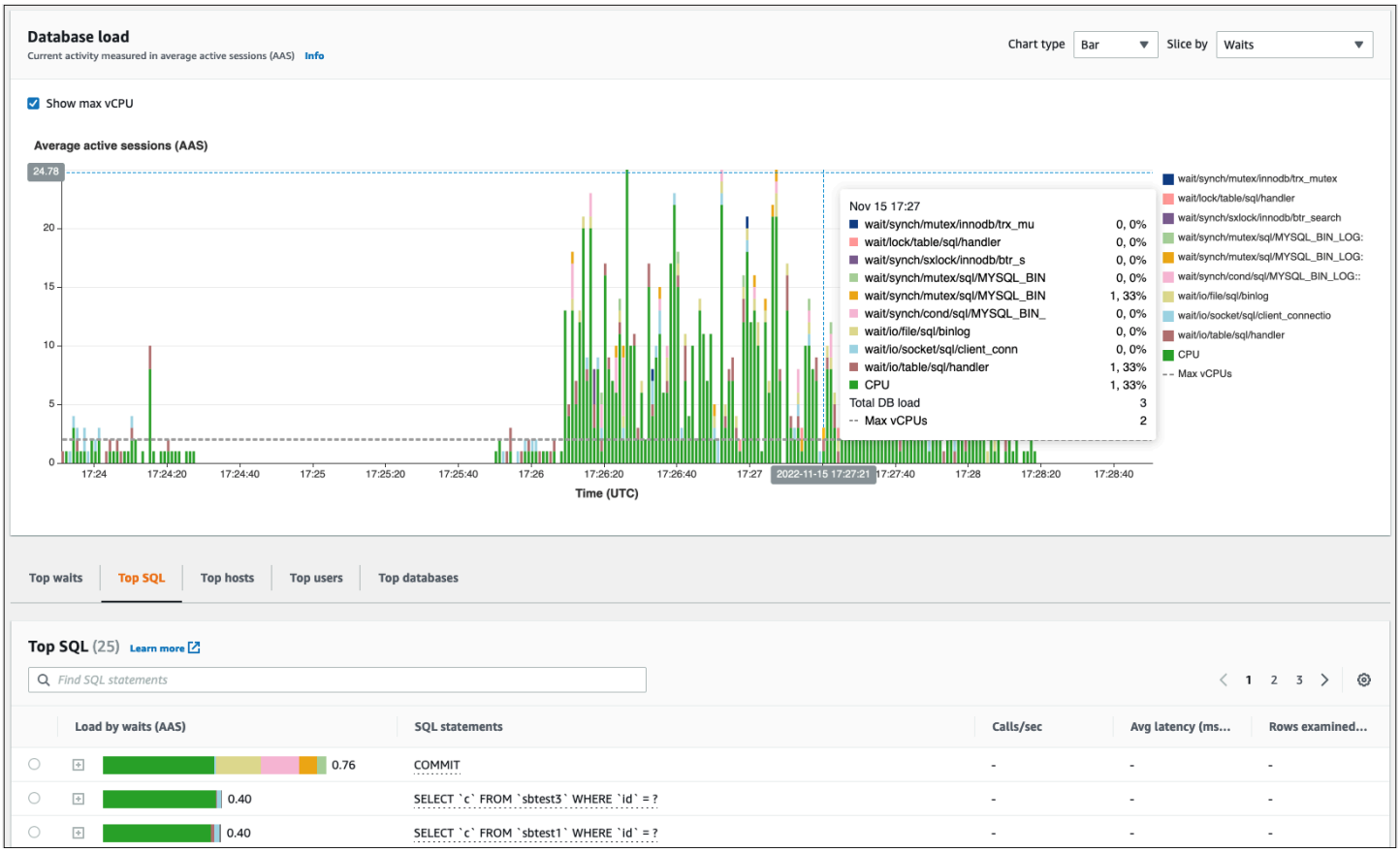
DB インスタンスの Performance Insights メトリクス

Performance Insights は、以下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します。

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す。これは 1 秒ごとに収集され、Amazon CloudWatch に自動的に発行されます。これは、SQL クエリを同時に実行しているセッション数である平均アクティブセッション (AAS) における DB インスタンスのアクティビティを表します。DBLoad メトリクスは、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して解釈できるため、他の時系列メトリクスとは異なります。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す。カテゴリごとにスライスとして使用して、データベース負荷のさまざまな特性を表すことができます。データベースロードの計算方法の詳細については、Amazon RDS ドキュメントの「データベースロード」を参照してください。

次の画面図は、Performance Insights ツールを示しています。



ディメンション

- 待機イベントは、データベースセッションが処理を続行するためにリソースまたは別のオペレーションが完了するまで待機する条件です。などの SQL ステートメントを実行しSELECT * FROM big_table、このテーブルが割り当てられた InnoDB バッファプールよりもはるかに大きい場合、セッションは、データファイルの物理 I/O オペレーションによって引き起こされるwait/io/file/innodb/innodb_data_file待機イベントを待機する可能性が高くなります。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ため、データベースモニタリングにとって重要なディメンションです。待機イベントは、セッション内で実行している SQL ステートメントが待機に最も多くかかるリソースとオペレーションを示します。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高い場合に発生し、wait/synch/mutex/innodb/buf_pool_mutexイベントはスレッドがInnoDB バッファプールのロックを取得してメモリ内のページにアクセスする場合に発生します。すべての MySQL および MariaDB 待機イベントの詳細については、MySQL ドキュメントの「待機イベントの概要テーブル」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「Performance Schema Instrument Naming Conventions」を参照してください。

- SQL は、どの SQL ステートメントがデータベースの総ロードに最も寄与しているかを示します。Amazon RDS Performance Insights のデータベースロード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です。待機によるロード (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の詳細なリストを取得できます。リストで SQL ステートメントを選択すると、Performance Insights は関連する待機イベントをデータベースロードチャートに、SQL ステートメントテキストを SQL テキストセクションに表示します。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、接続されたクライアントのホスト名を表示します。このディメンションは、ロードの大部分をデータベースに送信しているクライアントホストを特定するのに役立ちます。
- ユーザーは、データベースにログインしているユーザーごとに DB 負荷をグループ化します。
- データベースは、クライアントが接続されているデータベースの名前で DB 負荷をグループ化します。

カウンターメトリクス

カウンターメトリクスは、DB インスタンスの再起動時にのみ値を増加またはゼロにリセットできる累積メトリクスです。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、単一の単調に増加するカウンターを表します。

- ネイティブカウンターは、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。以下に例を示します。
  - `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、最初のテーブルの完全な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取得できず、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クスの定義については、MySQL ドキュメントの「サーバーステータス変数」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。ネイティブではないカウンターメトリクスは、レイテンシー、比率、ヒット率を表すことができます。以下に例を示します。

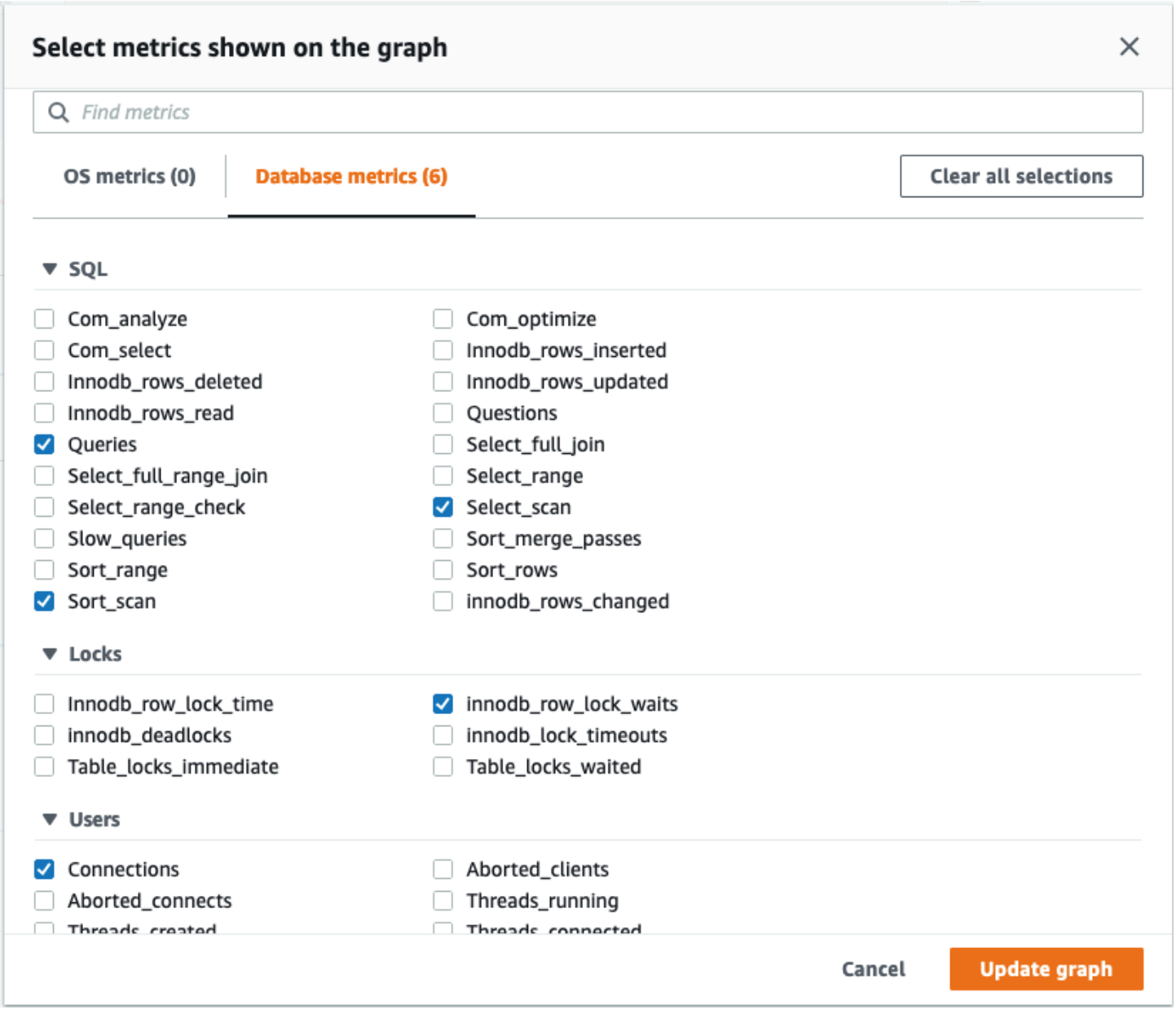
- `Cache.innoDB_buffer_pool_hits` は、InnoDB がディスクを使用せずにバッファプールから取得できる読み取りオペレーションの数を表します。これは、ネイティブカウンターメトリクスから次のように計算されます。

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

- `IO.innoDB_datafile_writes_to_disk` は、ディスクへの InnoDB データファイルの書き込みオペレーションの数を表します。データファイルに対するオペレーションのみをキャプチャします。二重書き込みややり直しの口グ記録の書き込みオペレーションはキャプチャしません。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、メトリクスの管理を選択し、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。



グラフの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 は、クエリが実行されている 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のメトリクスを収集します。一般的に、Performance Insights 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。ただし、MariaDB および MySQL DB インスタンスの場合、統計はダイジェストレベルでのみ収集されます。

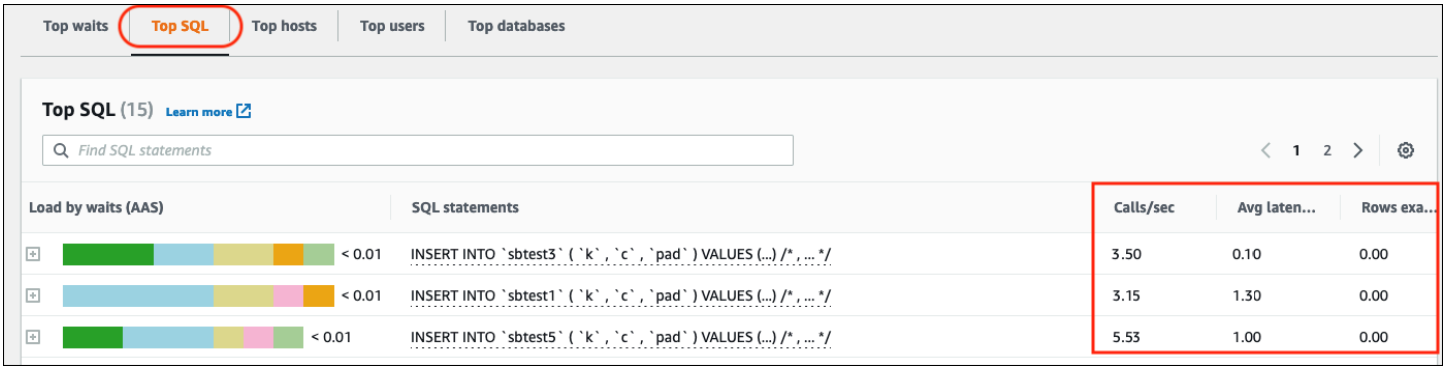
- ダイジェスト統計は、パターンは同じですが、最終的には異なるリテラル値を持つすべてのクエリの複合メトリクスです。ダイジェストは、特定のリテラル値を変数に置き換えます。次に例を示します。

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

- ダイジェストされた各 SQL ステートメントの 1 秒あたりの統計を表すメトリクスがあります。たとえば、 は 1 秒あたりの呼び出し (SQL ステートメントが 1 秒あたりに実行された回数) `sql_tokenized.stats.count_star_per_sec`を表します。

- Performance Insights には、SQL ステートメントの呼び出しごとの統計情報を提供するメトリクスも含まれています。たとえば、 は、呼び出しあたりの SQL ステートメントの平均レイテンシーをミリ秒単位でsql_tokenized.stats.sum_timer_wait_per_call表示します。

SQL 統計は、上位ディメンションテーブルの上位 SQL タブにある Performance Insights ダッシュボードで使用できます。



DB インスタンスの CloudWatch メトリクス

Amazon CloudWatch には、Amazon RDS が自動的に発行するメトリクスも含まれています。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。これは、mysqld プロセスの厳密な意味で DB インスタンスではなく、Amazon RDS (サービス) インスタンス (クラウドで実行されている分離されたデータベース環境) を参照します。したがって、これらのデフォルトのメトリクスのほとんどは、用語の厳密な定義で OS メトリクスのカテゴリに分類されます。例としてはCPUUtilization、、SwapUsage、 WriteIOPSなどがあります。ただし、MariaDB と MySQL に適用される DB インスタンスメトリクスがいくつかあります。

- BinLogDiskUsage – バイナリログが占有するディスク容量。
- 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 インスタンスがソース DB インスタンスより遅れる時間。

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

Amazon RDS Performance Insights は、DB インスタンスのメトリクスとディメンションのほとんどをモニタリングし、 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。ただし、パフォーマンス関連の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、それらのメトリクスは CloudWatch に存在する必要があります。

Performance Insights は CloudWatch にメトリクスを自動的に発行します。Performance Insights から同じデータをクエリできますが、CloudWatch にメトリクスを含めると、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、`os.memory.free`や などの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す`db.Locks.Innodb_row_lock_time`。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 メトリクスは 1 分に 1 回収集されます。拡張モニタリングが有効になっている場合、OS メトリクスは選択した期間にわたって収集されます。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の切り替え」を参照してください。

Performance Insights では、DB インスタンスの事前設定済みまたはカスタム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。 CloudWatch 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、EC2 インスタンス、Amazon Elastic File System (Amazon EFS) リソース、Elastic Load Balancing (ELB) リソースなど、システム内のさまざまなリソースに関連付けられたメトリクスの概要と DB インスタンスメトリクスを、統一された包括的なビューで表示できます。

CloudWatch `DB_PERF_INSIGHTS`メトリクス数学関数を使用して、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 メトリクスでアラームを作成するには、CloudWatch ドキュメントの指示に従います。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')
```

ここで、 `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。アラームの作成を確定するには、アラームアクションを設定し、名前と説明を追加し、アラームをプレビューして作成します。

OS モニタリング

Amazon RDS for MySQL または MariaDB の DB インスタンスは、基盤となるシステムリソースである CPU、メモリ、ネットワーク、ストレージを使用する Linux オペレーティングシステムで実行されます。

```
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となる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大きく依存します。たとえば、CPU はデータベースソフトウェアの指示を実行し、他のシステムリソースを管理するため、システムのパフォーマンスにとって重要なコンポーネントです。CPU が過剰に使用されている場合 (つまり、負荷が DB インスタンスにプロビジョニングされたよりも多くの CPU 電力を必要とする場合) 、この問題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影響します。

データベースエンジンは、メモリを動的に割り当てて解放します。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、システムはディスクにあるスワップメモリにメモリページを書き込みます。ディスクはメモリよりもはるかに低速であるため、ディスクが SSD NVMe テクノロジーに基づいていても、メモリの割り当てが多すぎるとパフォーマンスが低下します。メモリ使用率が高いと、ページファイルのサイズが大きくなり、追加のメモリがサポートされるため、データベースレスポンスのレイテンシーが増加します。メモリ割り当てが高すぎて RAM とスワップメモリスペースの両方が枯渇すると、データベースサービスが使用できなくなる可能性があり、ユーザーは などのエラーが発生する可能性があります`[ERROR] mysqld: Out of memory (Needed xyz bytes)`。

MySQL および MariaDB データベース管理システムは、テーブル、インデックス、バイナリログ、REDO ログ、元に戻すログ、二重書き込みバッファファイルなどのディスク上の構造を保存するディスクで構成されるストレージサブシステムを使用します。したがって、データベース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を実行する必要があります。デー

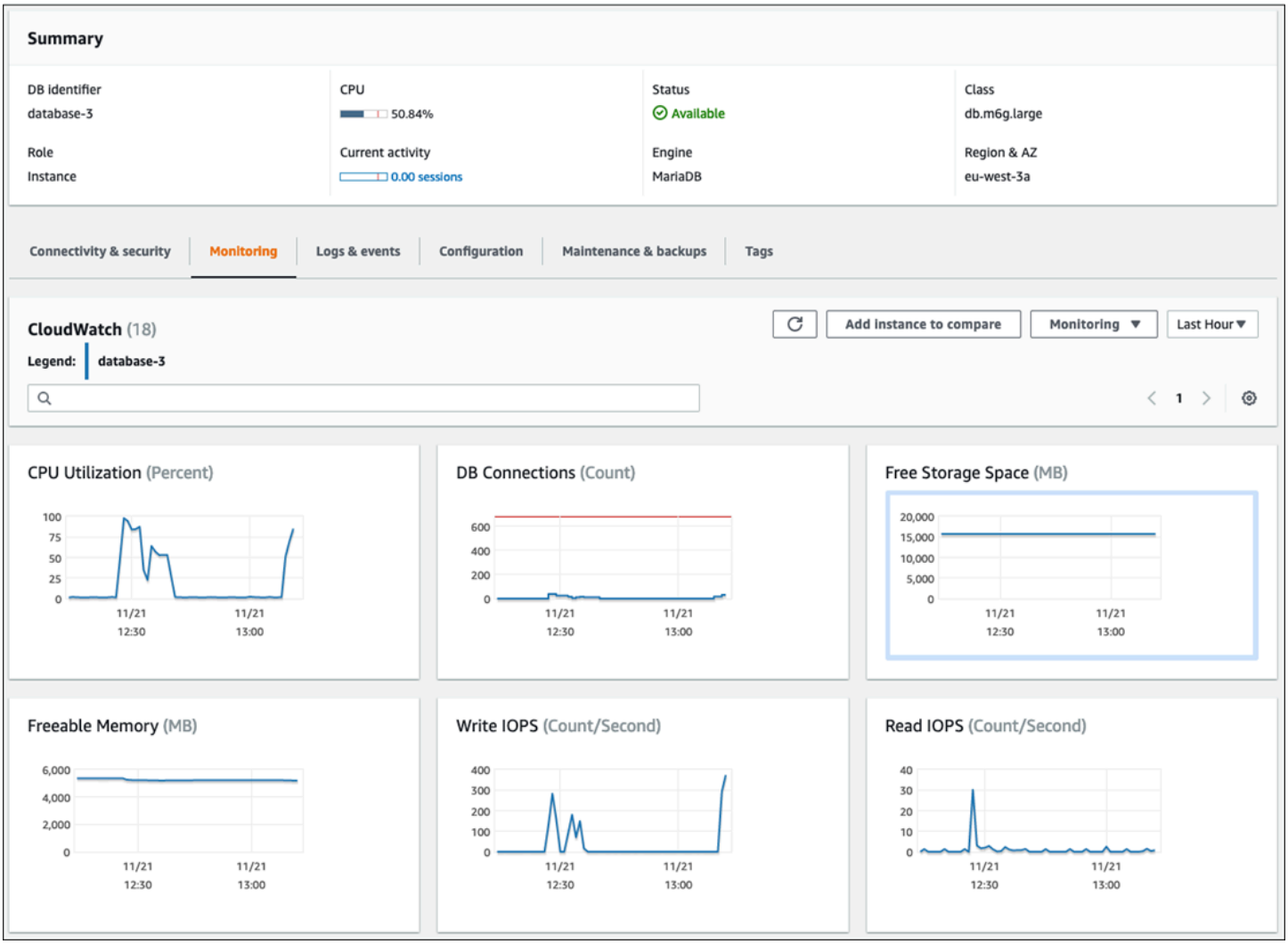
タベースの最適なオペレーションを行うには、ディスク I/O 使用率とディスク容量の割り当てをモニタリングして調整することが重要です。データベースのパフォーマンスは、データベースがディスクでサポートされる最大 IOPS またはスループットの制限に達したときに影響を受ける可能性があります。例えば、インデックススキャンによるランダムアクセスのバーストにより、1 秒あたりに多数の I/O オペレーションが発生し、最終的に基盤となるストレージの制限に達する可能性があります。フルテーブルスキャンは IOPS 制限に達しない可能性がありますが、1 秒あたりのメガバイト数で測定される高スループットが発生する可能性があります。などの エラーによりデータベースが使用できなくなったり破損したりする`OS error code 28: No space left on device`可能性があるため、ディスク容量の割り当てをモニタリングしてアラートを生成することが重要です。

Amazon RDS は、DB インスタンスが実行され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提供します。Amazon RDS は、1 つの OS メトリクスセットを CloudWatch に自動的に発行します。これらのメトリクスは、Amazon RDS コンソールと CloudWatch ダッシュボードで表示および分析でき、CloudWatch で選択したメトリクスにアラームを設定できます。以下に例を示します。

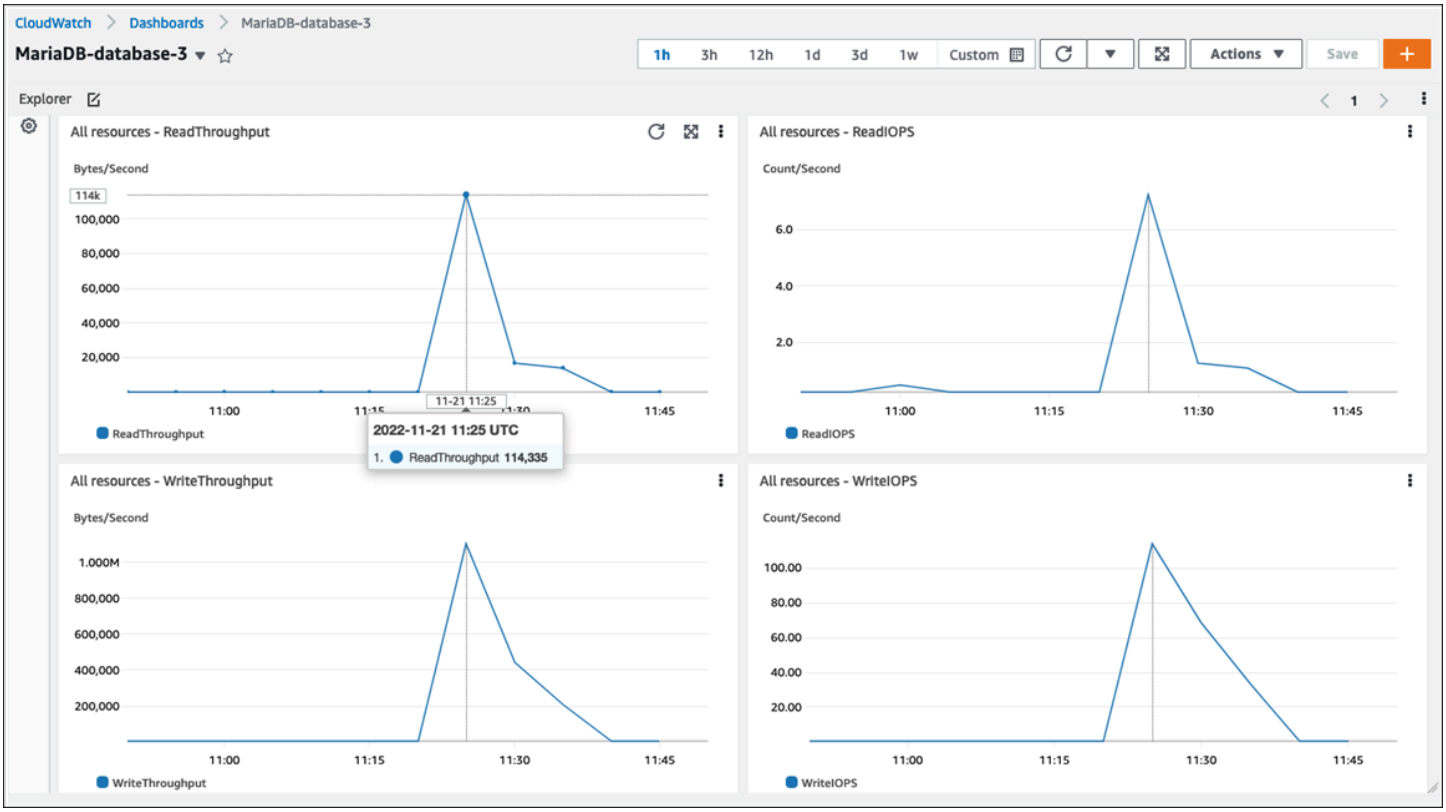
- `CPUUtilization` – CPU 使用率の割合。
- `BinLogDiskUsage` – バイナリログによって占有されているディスク容量。
- `FreeableMemory` – 使用可能なランダムアクセスメモリの量。これは、 の `MemAvailable`フィールドの値を表します`/proc/meminfo`。
- `ReadIOPS` – 1 秒あたりのディスク読み取り I/O オペレーションの平均数。
- `WriteThroughput` – ローカルストレージのディスクに書き込まれた 1 秒あたりの平均バイト数。
- `NetworkTransmitThroughput` – DB ノードの送信ネットワークトラフィック。モニタリングとレプリケーションに使用されるデータベーストラフィックと Amazon RDS トラフィックの両方を結合します。

Amazon RDS から CloudWatch に発行されたすべてのメトリクスの完全な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。

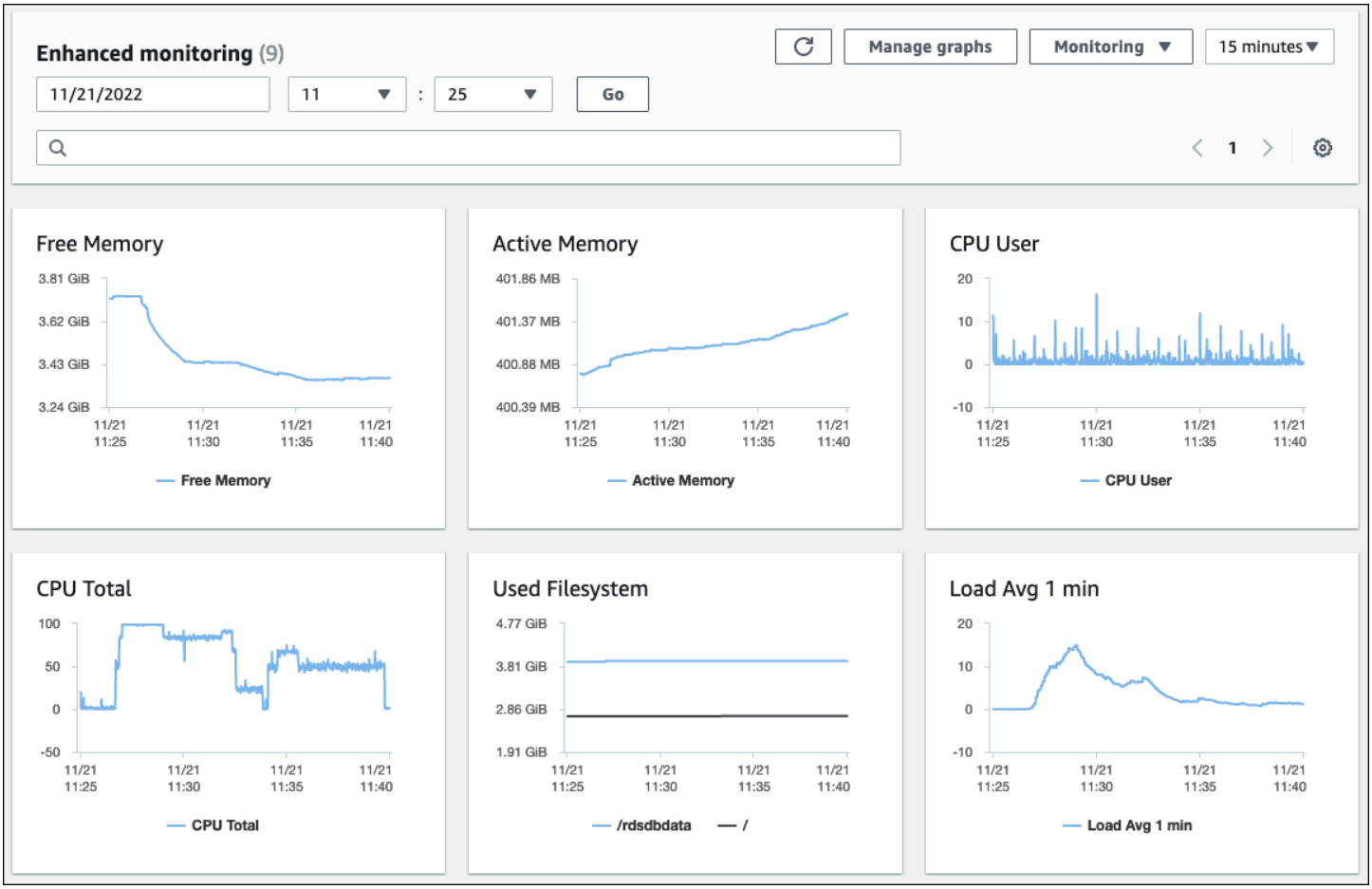
次の図は、Amazon RDS コンソールに表示される Amazon RDS の CloudWatch メトリクスの例を示しています。



次のグラフは、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されます。このツールを使用すると、リアルタイムのシステムメトリクスと OS プロセス情報を提供することで、Amazon RDS for MariaDB および Amazon RDS for MySQL DB インスタンスの状態をより詳細に把握できます。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、ツールは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を収集します。この情報は、次の画面に示すように、Amazon RDS コンソールで表示および分析できます。



拡張モニタリングが提供する主要なメトリクスには、次のようなものがあります。

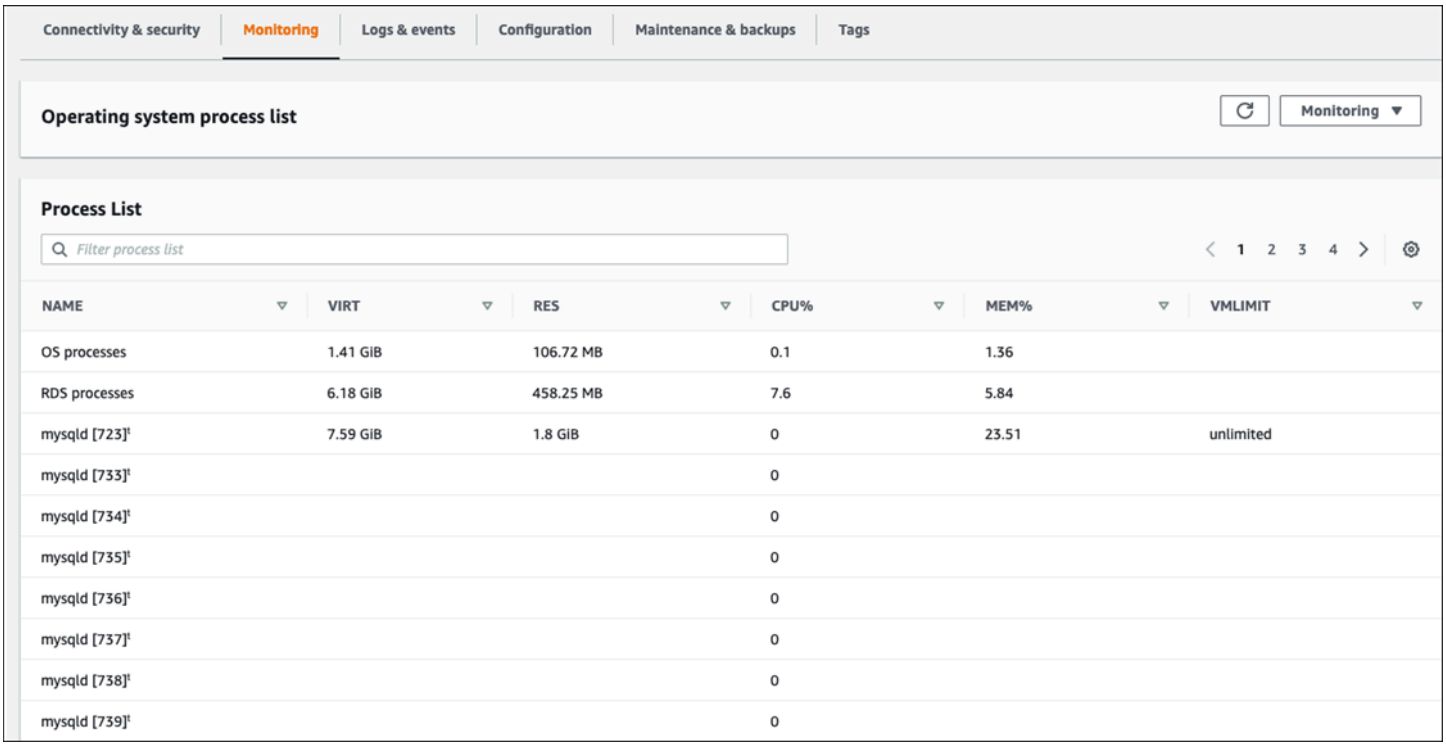
- `cpuUtilization.total` – 使用中の CPU の合計パーセンテージ。
- `cpuUtilization.user` – ユーザープログラムによって使用されている CPU の割合。
- `memory.active` – 割り当てられたメモリの量をキロバイト単位で表します。
- `memory.cached` – ファイルシステムベースの I/O のキャッシュに使用されるメモリの量。
- `loadAverageMinute.one` – 過去 1 分間に CPU 時間をリクエストしたプロセスの数。

メトリクスの完全なリストについては、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 コンソールでは、OS プロセスリストに、DB インスタンスで実行され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の集約された概要を表します。これらのプロセスは、通常、データベースのパフォーマンスへの影響は最小限です。
- RDS プロセス – このセクションでは、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ます。例えば、Amazon RDS 管理エージェント、モニタリングと診断プロセス、および同様のプロセスが含まれます。
- RDS 子プロセス – このセクションでは、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ます。この場合、`mysqld`プロセスとそのスレッドです。`mysqld` スレッドは親`mysqld`プロセスの下にネストされた状態で表示されます。

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

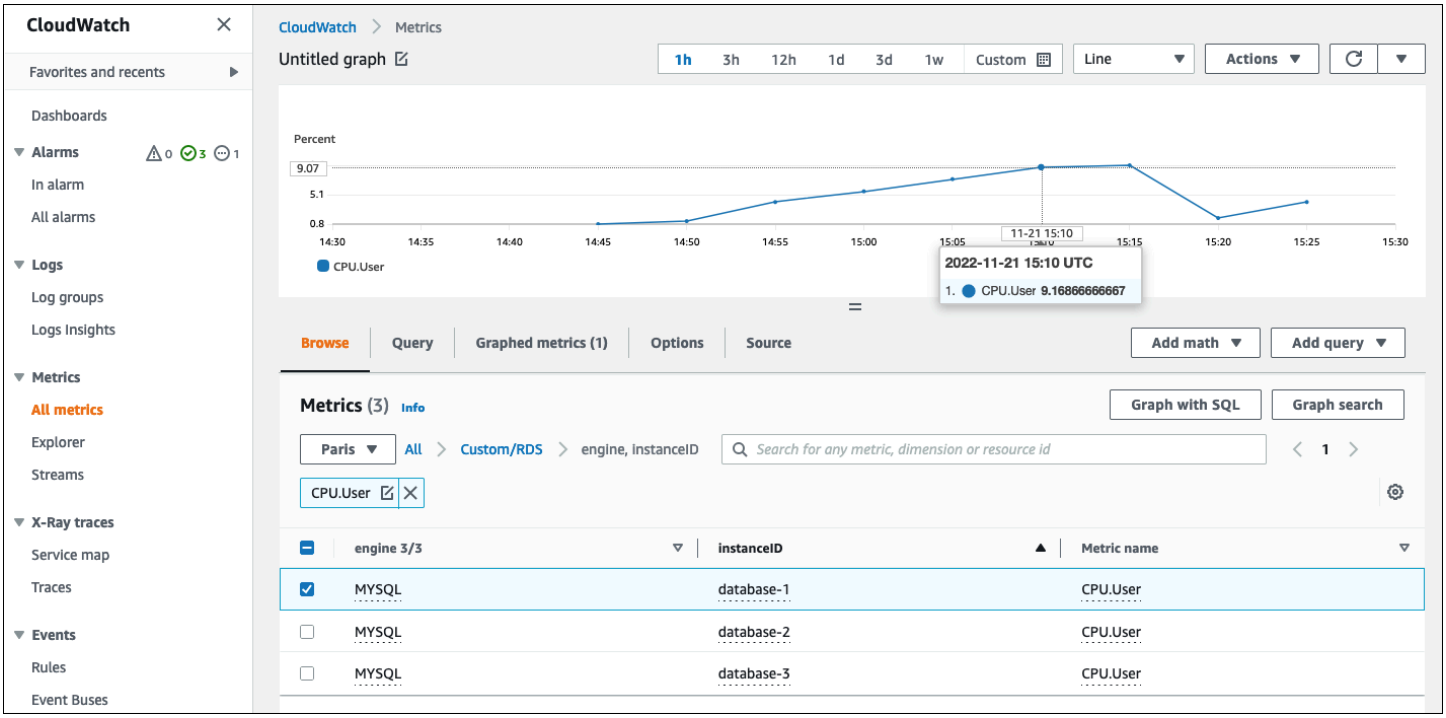


Amazon RDS は、拡張モニタリングのメトリクスを CloudWatch Logs アカウントに配信します。Amazon RDS コンソールに表示されるモニタリングデータは、CloudWatch Logs から取得されます。DB インスタンスのメトリクスを CloudWatch Logs からログストリームとして取得することもできます。 CloudWatch これらのメトリクスは JSON 形式で保存されます。選択したモニタリングシステムで CloudWatch Logs からの拡張モニタリング JSON 出力を使用できます。

CloudWatch ダッシュボードにグラフを表示し、メトリクスが定義されたしきい値を超えた場合にアクションを開始するアラームを作成するには、CloudWatch Logs から CloudWatch にメトリクスフィルターを作成する必要があります。詳細な手順については、拡張モニタリング 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に関する AWS re:Post 記事を参照してください。

次の例は、 Custom/RDS名前空間CPU.Userの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は、CloudWatch Logs からcpuUtilization.user拡張モニタリングメトリクスをフィルタリングすることによって作成されます。



CloudWatch リポジトリでメトリクスが使用可能な場合、CloudWatch ダッシュボードでメトリクスを表示および分析し、さらに数学およびクエリオペレーションを適用し、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、観測値が定義されたアラーム条件と一致しない場合にアラートを生成できます。

イベント、ログ、監査証跡

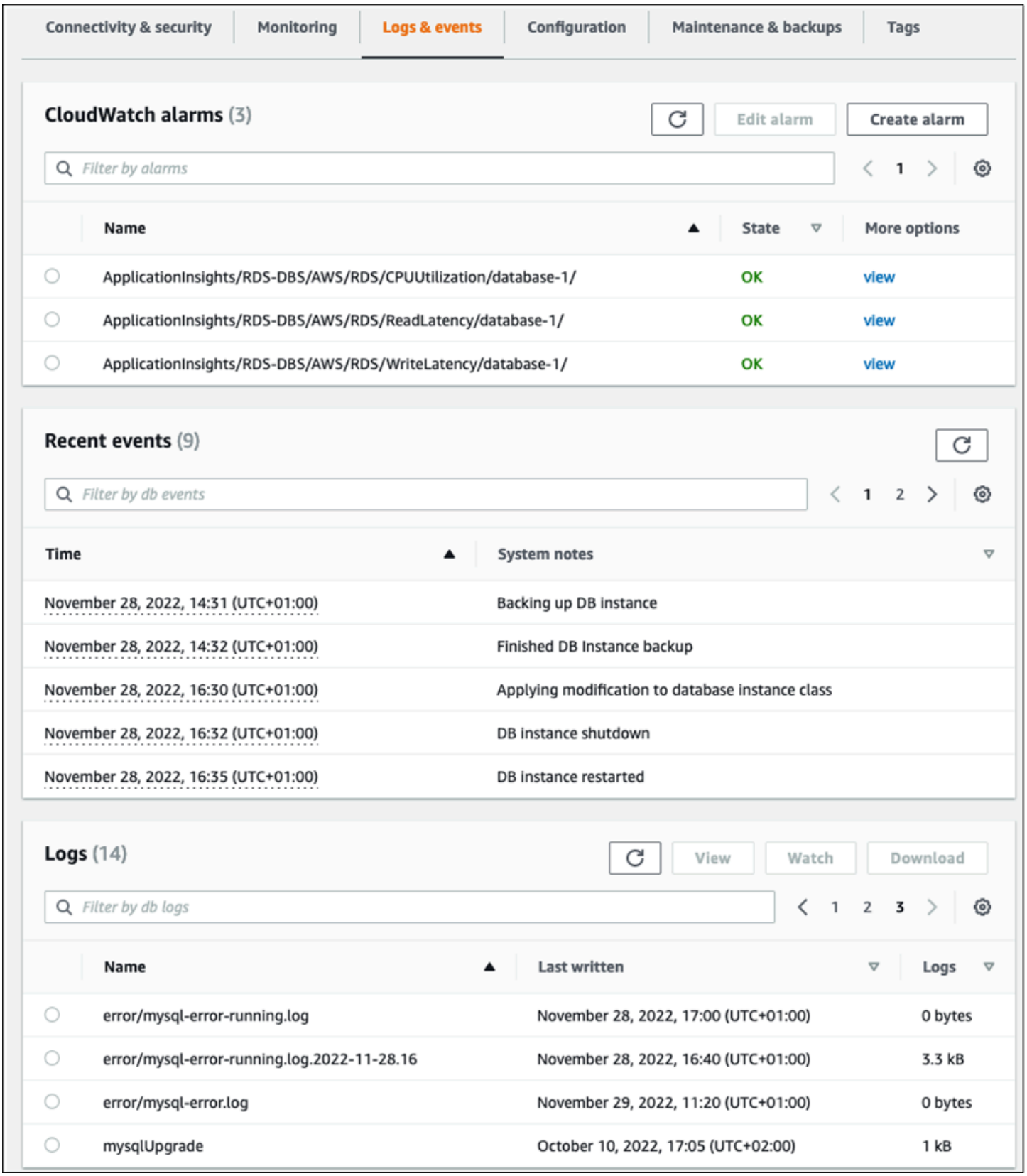
DB インスタンスメトリクスと OS メトリクスのモニタリング、傾向の分析、メトリクスとベースライン値の比較、値が定義されたしきい値を超えた場合にアラートを生成することがすべて必要であり、Amazon RDS DB インスタンスの信頼性、可用性、パフォーマンス、セキュリティを達成および維持するのに役立つベストプラクティスです。ただし、完全なソリューションでは、MySQL および MariaDB データベースのデータベースイベント、ログファイル、監査証跡もモニタリングする必要があります。

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 環境の変更を示します。例えば、DB インスタンスのステータスが開始から使用可能に変わると、Amazon RDS はイベント を生成しますRDS-EVENT-0088 The DB instance has been started。Amazon RDS は、ほぼリアルタイムで Amazon EventBridge にイベントを配信します。イベントには、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 からアクセスできます。次の画面図は、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 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など、さまざまなタイプのイベントを出力します。情報には以下が含まれます。

- 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"]`

完全なリファレンスについては、Amazon RDS ドキュメントの「Amazon RDS 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。

これらのイベントは、DB インスタンスの可用性におけるステータスの変更、設定の変更、リードレプリカのステータスの変更、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、その他多くの通知を示しているため、Amazon RDS イベントをモニタリングすることをお勧めします。例えば、データベースのパフォーマンスと耐久性を向上させるようにリードレプリカ 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ことをお勧めします。これは、 などのイベントが、リードレプリカがプライマリ DB インスタンスと同期していない`RDS-EVENT-0057 Replication on the read replica was terminated`ことを示しているためです。このようなイベントが発生したことを担当チームに通知すると、問題のタイムリーな緩和に役立ちます。Amazon EventBridge や Amazon Simple Queue Service (Amazon SQS) AWS Lambdaや Amazon Simple Notification Service (Amazon SNS) などのその他の は AWS のサービス、データベースの可用性の問題やリソースの変更などのシステムイベントへの対応を自動化するのに役立ちます。

Amazon RDS コンソールでは、過去 24 時間からイベントを取得できます。または Amazon RDS API を使用して AWS CLI 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"
        }
    ]
}
```

指定された有効期限まで、または永続的にイベントを保存する場合は、CloudWatch Logs を使用して、Amazon RDS によって生成されたイベントに関する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. イベントで呼び出される Lambda 関数を作成し、イベントから CloudWatch Logs に情報をログに記録します。CloudWatch Logs は Lambda と統合されており、 への印刷関数を使用してイベント情報をログに記録する便利な方法を提供しますstdout。
2. Lambda 関数へのサブスクリプション (プロトコルを Lambda に設定) を使用して SNS トピックを作成し、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 トピックを取得して Amazon RDS 通知を受信する方法に関する AWS re:Post 記事を参照してください。
4. Amazon RDS コンソールで、新しいイベントサブスクリプションを作成します。ターゲットを ARN に設定し、以前に作成した SNS トピックを選択します。ソースタイプとイベントカテゴリを設定して、要件に応じて含めます。詳細については、Amazon RDS ドキュメントの「Amazon RDS イベント通知のサブスクライブ」を参照してください。

データベースログ

MySQL および MariaDB データベースは、監査とトラブルシューティングのためにアクセスできるログを生成します。これらのログは次のとおりです。

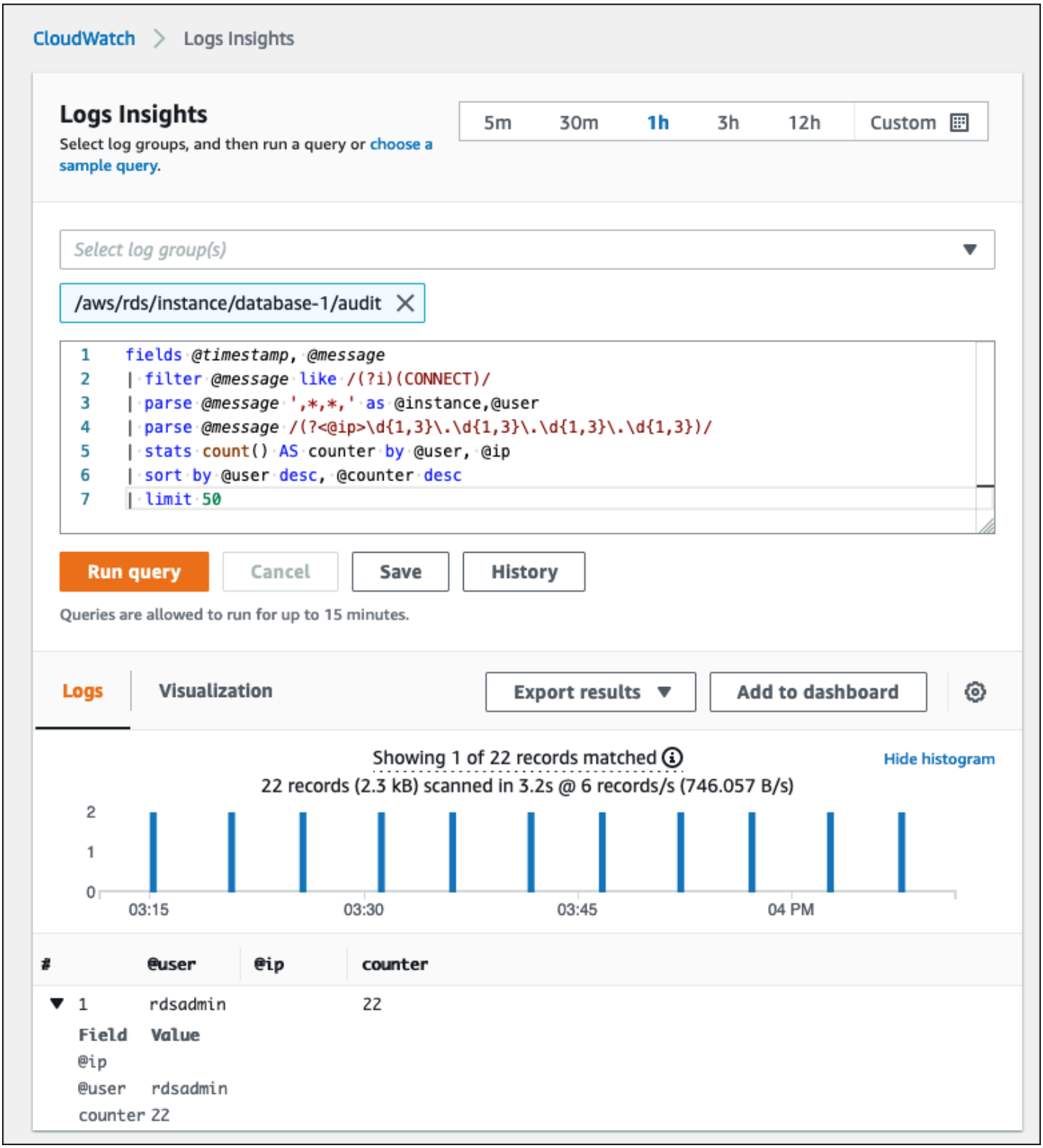
- 監査 – 監査証跡は、サーバーのアクティビティを記録する一連のレコードです。クライアントセッションごとに、サーバーに接続したユーザー (ユーザー名とホスト) 、実行されたクエリ、アクセスされたテーブル、変更されたサーバー変数を記録します。
- エラー – このログには、サーバーの (mysqld) 起動時間とシャットダウン時間、サーバーの起動時とシャットダウン時、およびサーバーの実行中に発生するエラー、警告、メモなどの診断メッセージが含まれます。
- 全般 – このログにはmysqld、各クライアントの接続および切断アクティビティ、クライアントから受信した SQL クエリなど、 のアクティビティが記録されます。一般的なクエリログは、エラーが疑われ、クライアントが に何を送信したかを正確に知りたい場合に非常に役立ちますmysqld。
- スロークエリ – このログは、実行に長い時間がかかった SQL クエリのレコードを提供します。

ベストプラクティスとして、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を実行し、耐久性の高いストレージにデータを保存し、CloudWatch Logs エージェントでデータを管理できます。Amazon RDS コンソールからデータベースログにアクセスして監視できます。CloudWatch Logs Insights を使用して、CloudWatch Logs のログデータをインタラクティブに検索および分析することもできます。次の例は、ログにCONNECTイベントが表示される回数、接続者、接続元のクライアント (IP アドレス) を確認する監査ログのクエリを示しています。監査ログからの抜粋は次のようになります。

```
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
```

次の図に示すように、Log Insights クエリの例では、 が 5 分localhostごとに からデータベースrdsadminに接続され、合計 22 回接続されていることを示しています。これらの結果は、アク

ティビティがモニタリングシステム自体などの内部 Amazon RDS プロセスから発生したことを示しています。



ログイベントには、MySQL および MariaDB DB インスタンスに関連付けられたオペレーションに関する警告やエラーなど、カウントする重要なメッセージが頻繁に含まれます。たとえば、オペレーションが失敗すると、エラーが発生し、エラーログファイルに次のように記録される場合があります。 `ERROR 1114 (HY000): The table zip_codes is full`これらのエントリをモニタリングして、エラーの傾向を把握できます。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、特定のパターンの特定のログをモニタリングし、予想される動作の違反があった場合にアラームを生成します。たとえば、エラーログを`/aws/rds/instance/database-1/error`モニタリングし、 などの特定のパターンを検索するロググループのメトリクスフィルターを作成しますERROR。フィルターパターンを に設定ERRORし、メトリクス値を に設定します1。フィルターは、キーワード を持つすべてのログレコードを検出しERROR、「ERROR」を含むログイベントごとにカウントを 1 ずつ増分します。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場合に通知するアラームを設定できます。

CloudWatch ダッシュボードを作成し、CloudWatch Logs Insights を使用してスロークエリログとエラーログをモニタリングする方法の詳細については、ブログ記事「Amazon RDS と Amazon Aurora MySQL をモニタリングするための Amazon CloudWatch ダッシュボードの作成」を参照してください。

監査証跡

監査証跡 (または監査ログ) は、 内のイベントのセキュリティ関連の時系列レコードを提供します AWS アカウント。これには、Amazon RDS のイベントが含まれます。Amazon RDS は、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のドキュメント証拠を提供します。Amazon RDS for MySQL または MariaDB では、監査証跡の使用には以下が含まれます。

- DB インスタンス監査ログのモニタリング
- での Amazon RDS API コールのモニタリング AWS CloudTrail

Amazon RDS DB インスタンスの場合、監査の目的には通常以下が含まれます。

- 以下の説明責任の有効化：
  - パラメータまたはセキュリティ設定で実行される変更
  - データベーススキーマ、テーブル、または行で実行されるアクション、または特定のコンテンツに影響するアクション
- 侵入の検出と調査

- 疑わしいアクティビティの検出と調査
- 認可の問題の検出。たとえば、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ため

データベース監査証跡は、データベース内の機密データを誰が表示または変更したかという一般的な質問に答えようとします。これが発生したのはいつですか？ 特定のユーザーがデータにアクセスした場所 特権ユーザーは無制限のアクセス権を悪用しましたか？

MySQL と MariaDB はどちらも、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。このプラグインは、データベースにログオンしているユーザーや、データベースに対して実行されているクエリなどのデータベース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 の 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 ドキュメントを参照してください。

AWS アカウントの AWS クラウド 監査証跡は、 AWS CloudTrailサービスによって提供されます。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。すべての Amazon RDS アクションがログに記録されます。CloudTrail は、ユーザー、ロール、または別の AWS サービスによって実行された Amazon RDS のアクションの記録を提供します。イベントには、 AWS マネジメントコンソールで実行されたアクション AWS CLI、 SDKs と AWS APIs。

例

一般的な監査シナリオでは、証 AWS CloudTrail 跡をデータベース監査ログと Amazon RDS イベントモニタリングと組み合わせる必要がある場合があります。例えば、Amazon RDS DB インスタンスのデータベースパラメータ (などdatabase-1) が変更され、タスクが変更を行ったユーザー、変更内容、変更日時を特定するシナリオがあるとします。

タスクを実行するには、次の手順に従います。

1. データベースインスタンスで発生した Amazon RDS イベントを一覧表示database-1し、メッセージ `configuration change`があるカテゴリにイベントがあるかどうかを確認しますFinished updating DB parameter group。

```
$ 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 インスタンスが使用している 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. を使用して AWS CLI 、 がデプロイされているリージョン、ステップ 1 で検出された Amazon RDS イベント前後の期間、 の CloudTrail イベントを検索しますEventName=ModifyDBParameterGroup。 database-1

```
$ 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"
}
```

CloudTrail イベントは、アカウント 111122223333Role1「」の AWS ロールUser1を使用してmariadb10-6-test、 の DB インスタンスで使用された DB パラメータグループ を変更したことを明らかにしdatabase-1ます2022-12-01 at 09:18:19 h。2 つのパラメータが変更され、次の値に設定されています。

- innodb_log_buffer_size = 8388612
- innodb_write_io_threads = 8

CloudTrail と CloudWatch Logs のその他の機能

CloudTrail コンソールでイベント履歴を表示することで、過去 90 日間の運用インシデントとセキュリティインシデントのトラブルシューティングを行うことができます。保持期間を延長し、追加のクエリ機能を利用するには、AWS CloudTrail Lake を使用できます。 AWS CloudTrail Lake を使用すると、イベントデータをイベントデータストアに最大 7 年間保持できます。さらに、このサービスは、イベント履歴の単純なキーと値のルックアップによって提供されるビューよりも、より深くカスタマイズ可能なイベントのビューを提供する複雑な SQL クエリをサポートします。

特定のアクティビティが発生したときに監査証跡をモニタリングし、アラームを設定し、通知を受け取るには、証跡レコードを CloudWatch Logs に送信するように CloudTrail を設定 CloudWatchする必要があります。証跡レコードが CloudWatch Logs として保存されると、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従って生成される CloudWatch アラームを作成できます。例えば、責任のあるチームに通知を送信するアラームを設定して、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 サービスのセキュリティ、可用性、パフォーマンス、信頼性に関して最も重要な情報源の 1 つです。継続的なセキュリティの脅威、停止、パフォーマンスの問題、またはシステム障害について IT チームに通知し、通知します。

情報技術インフラストラクチャライブラリ (ITIL)、特に IT サービス管理 (ITSM) のプラクティスは、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の焦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がアラートを生成して、IT 環境の変更、高リスクのアクション、障害についてチームや自動ツール (自動的に実行可能な項目の場合) に通知することです。IT アラートは、重大なインシデントにつながる可能性のあるシステム停止や変更に対する防御の最前線です。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、IT チームはダウンタイムを最小限に抑え、それに伴う高いコストを削減できます。

Well-Architected AWS フレームワークは、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 またはサードパーティーのモニタリングサービスを使用して、メトリクスが想定範囲外であることを示すアラームを設定します。

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、インシデント後レビューアクティビティを通じて IT 関連のイベントとインシデントを処理するための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

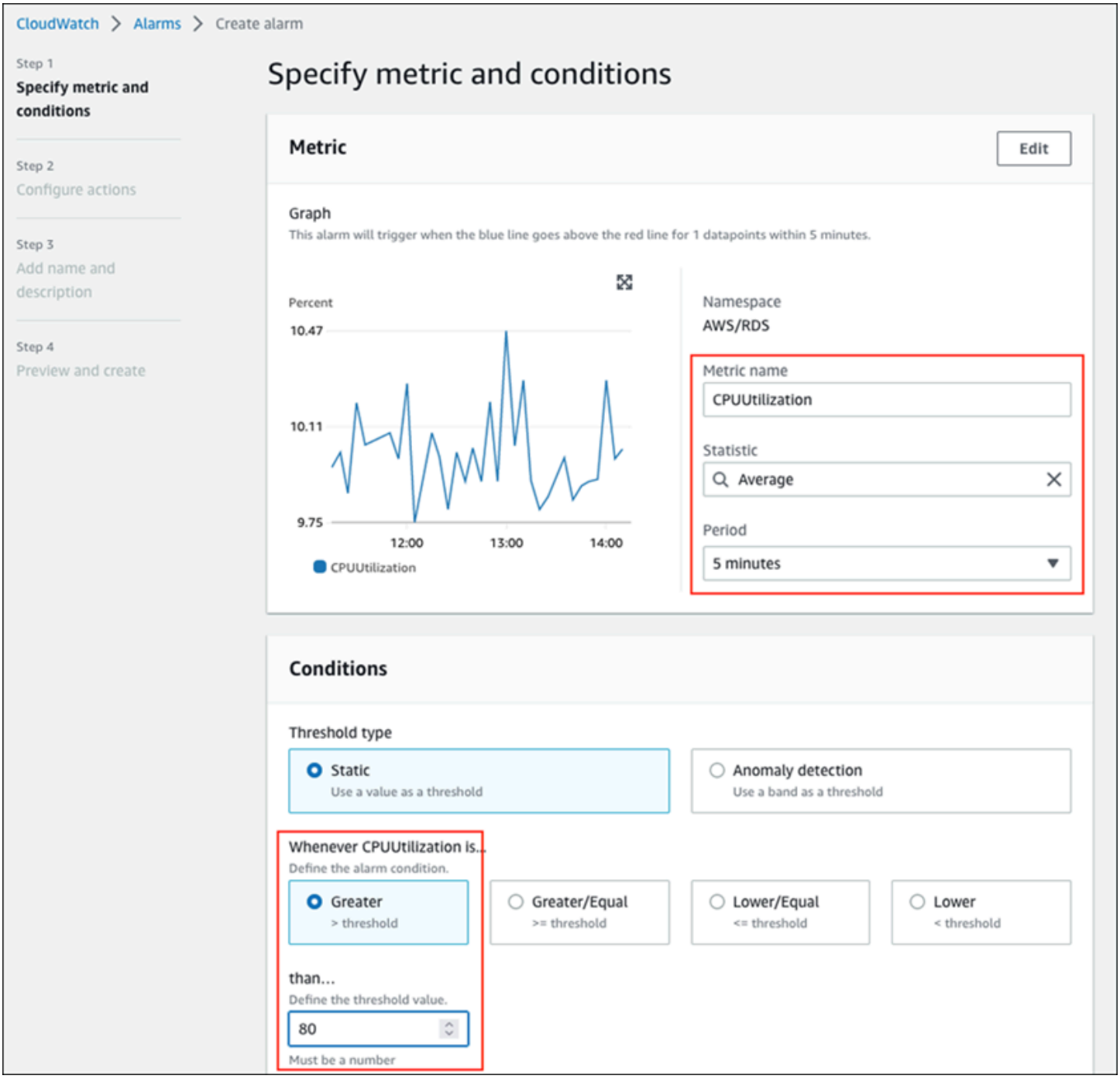
- CloudWatch アラーム
- EventBridge ルール
- アクションの指定、アラームの有効化と無効化

CloudWatch アラーム

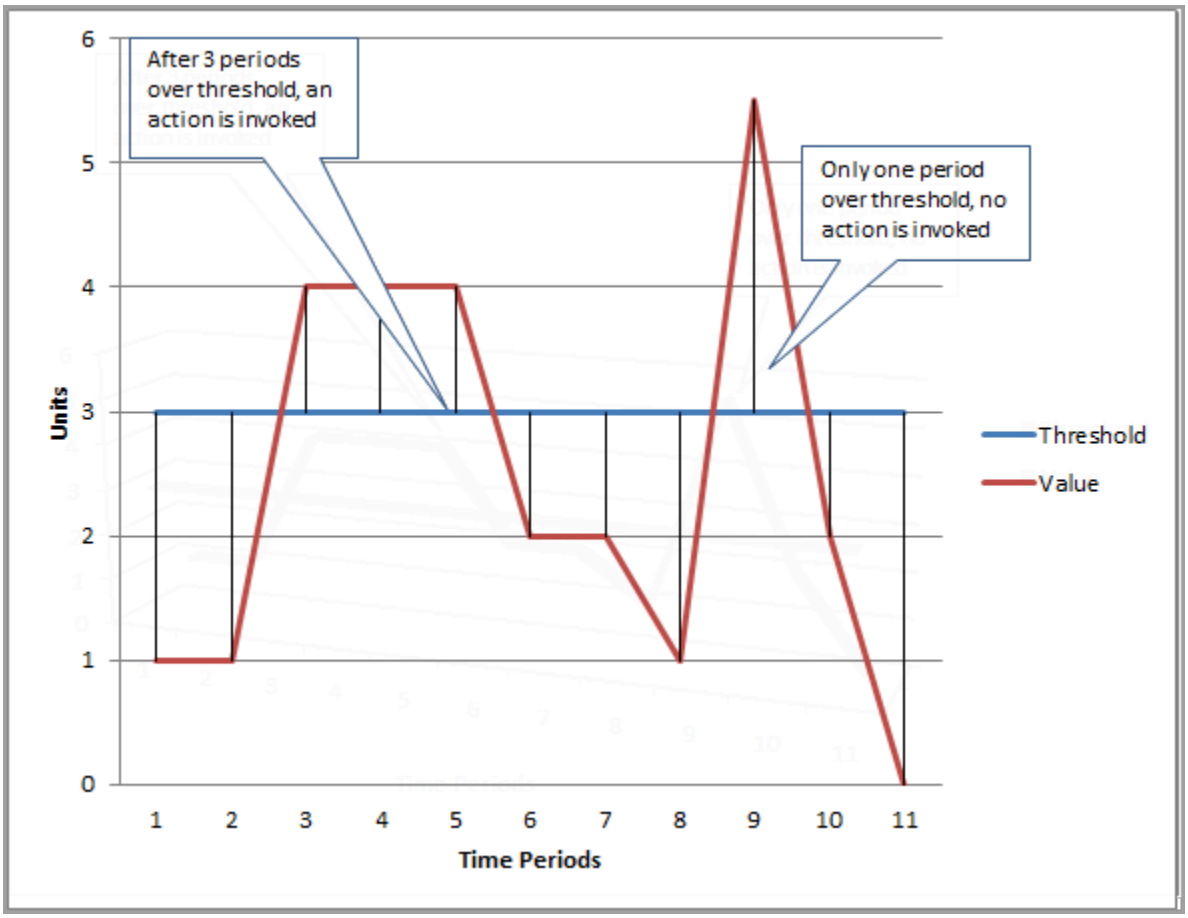
Amazon RDS DB インスタンスを操作するときは、さまざまな種類のメトリクス、イベント、トレースをモニタリングしてアラートを生成する必要があります。MySQL および MariaDB データベースの場合、重要な情報源は DB インスタンスメトリクス、OS メトリクス、イベント、ログ、監査証跡

です。CloudWatch アラームを使用して、指定した期間にわたって 1 つのメトリクスを監視することをお勧めします。

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を設定する方法を示しています。DB インスタンスの CPU 使用率が 5 分間の評価期間で 80% を超えた場合にトリガーされるようにアラームを設定します。



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) 場合、アラームはALARM状態になります。CPU が短時間で 80% を超える使用率までバーストし、再びしきい値を下回る場合、アラームは OK状態のままになります。次のグラフは、このロジックを示しています。



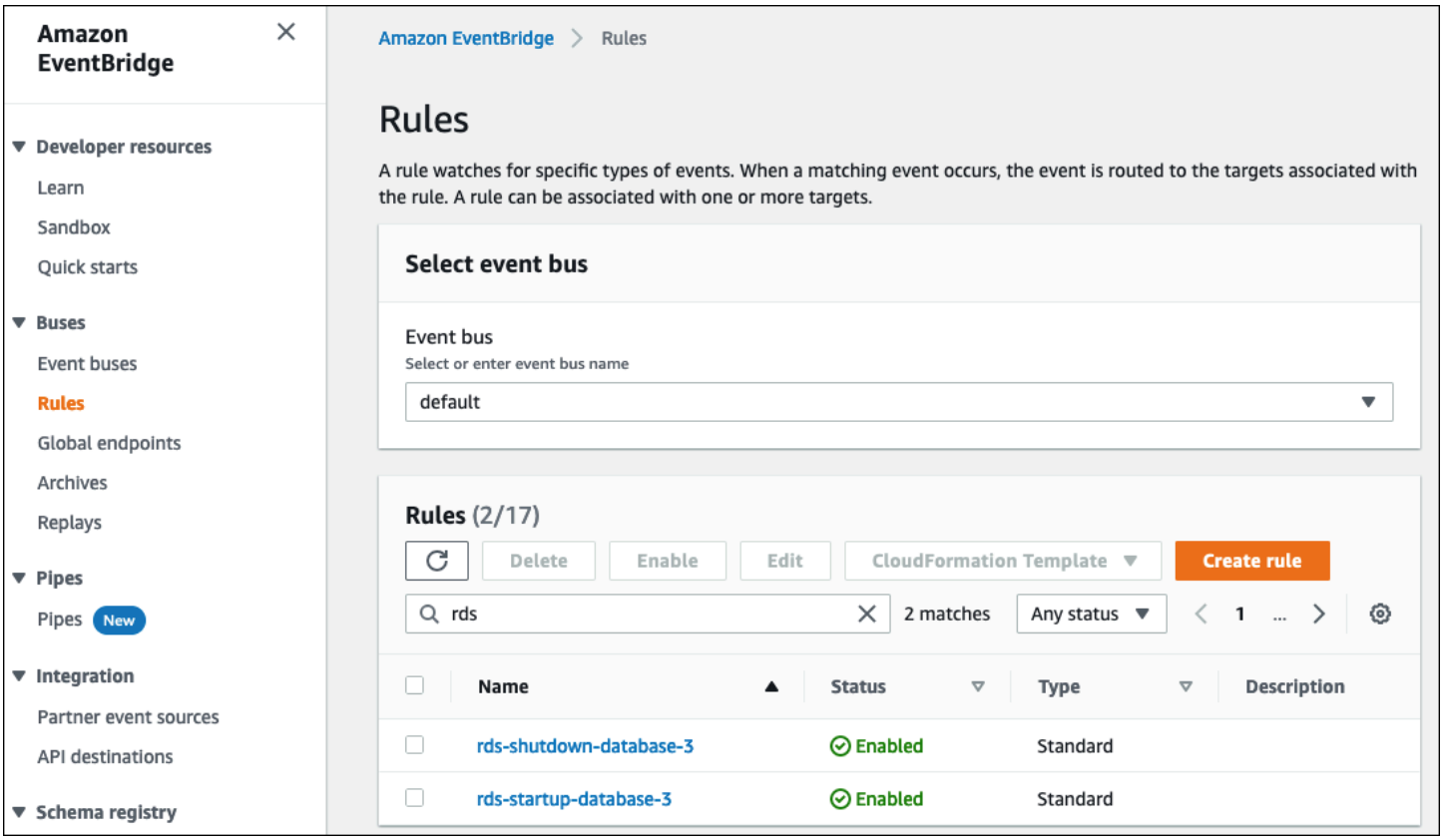
CloudWatch アラームは、メトリクスアラームと複合アラームをサポートします。

- メトリクスアラームは、単一の CloudWatch メトリクスを監視し、メトリクスに対して数式を実行できます。メトリクスアラームは Amazon SNS メッセージを送信できます。これにより、複数の期間にわたって特定のしきい値に対するメトリクスの値に基づいて 1 つ以上のアクションを実行できます。
- 複合アラームはルール式に基づいており、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状態になります。複合アラームは通常、不要なアラートの数を減らすために使用されます。例えば、アクションを実行しないように設定された複数のメトリクスアラームを含む複合アラームがあるとします。複合アラームは、複合内のすべての個々のメトリクスアラームが既に ALARM

CloudWatch アラームは CloudWatch メトリクスのみを監視できます。エラー、スロークエリ、または一般ログに基づいてアラームを作成する場合は、ログから CloudWatch メトリクスを作成する必要があります。OS モニタリングおよびイベント、ログ、監査証跡のセクションで前述したように、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することで、これを実現できます。同様に、拡張モニタリングメトリクスにアラートを送信するには、CloudWatch Logs 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 イベントは Amazon EventBridge に配信され、EventBridge ルールを使用してそれらのイベントに対応できます。例えば、次の画面に示すように、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。

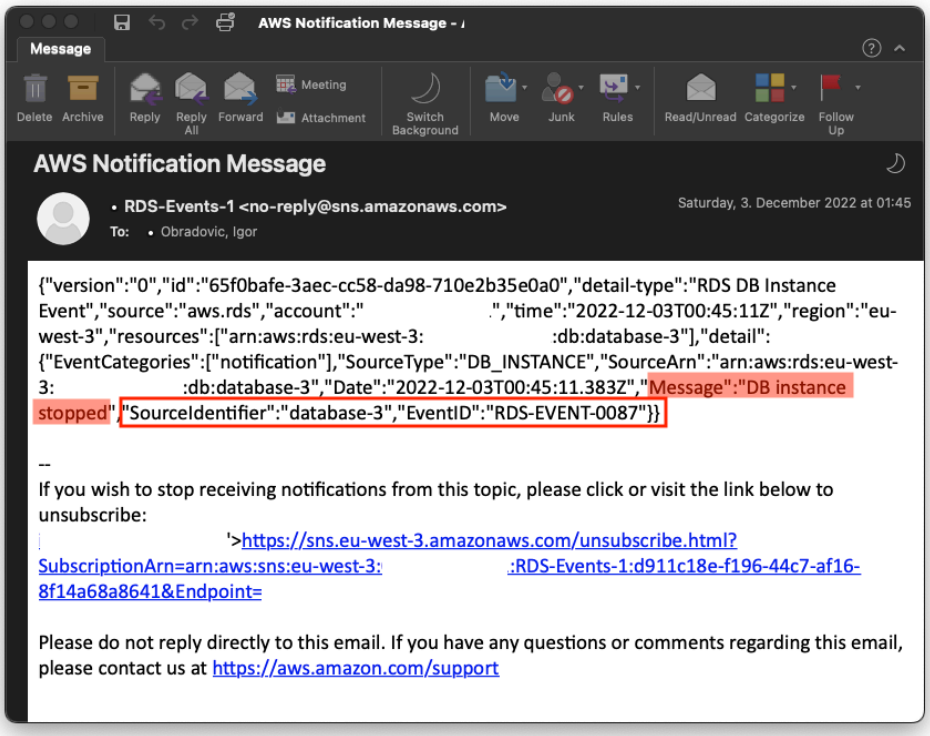


The DB instance has been stopped イベントを検出するルールには Amazon RDS イベント ID があるためRDS-EVENT-0087、ルールの Event Patternプロパティを次のように設定します。

```
{
  "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"]
  }
}
```

このルールは DB インスタンスdatabase-3のみを監視し、 RDS-EVENT-0087イベントを監視します。EventBridge はイベントを検出すると、ターゲットと呼ばれるリソースまたはエンドポイントにイベントを送信します。ここでは、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。イベントは、SNS トピック、Amazon Simple Queue Service (Amazon SQS) キュー、 AWS Lambda 関数、 AWS Systems Manager オートメーション、 AWS Batch ジョブ、Amazon API Gateway、Incident Manager の対応計画、 の一機能など、考えられるAWS Systems Manager多くのターゲットに送信できます。たとえば、通知 E メールと SMS を送信する SNS トピックを作成し、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 はイベントを EventBridge RDS-EVENT-0087に配信し、そこで検出されます。その後、EventBridge は SNS トピックであるターゲットを呼び出します。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されています。



アクションの指定、アラームの有効化と無効化

CloudWatch アラームを使用して、OK、、、ALARMおよび INSUFFICIENT_DATA状態の間でアラームが変化したときにアラームが実行するアクションを指定できます。CloudWatch には、Amazon Elastic Compute Cloud (Amazon EC2) アクションや Amazon EC2 Auto Scaling グループアクションなど、Amazon RDS メトリクスに適用されない SNS トピックおよびいくつかの追加のアクションカテゴリとの統合が組み込まれています。EventBridge は通常、Amazon RDS メトリクスに対して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ルールを記述し、ターゲットを定義するために使用されます。CloudWatch は、CloudWatch アラームの状態が変わるたびに EventBridge にイベントを送信します。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。

また、アラームを管理する必要もあります。たとえば、計画された設定変更やテスト中にアラームを自動的に無効にし、計画されたアクションが終了したときにアラームを再度有効にします。例えば、ダウンタイムを必要とする計画的なスケジュールされたデータベースソフトウェアのアップグレードがあり、データベースが使用できなくなった場合にアラームがアクティブ化される場合は、 の API アクション DisableAlarmActions と EnableAlarmActions、または disable-alarm-actions と enable-alarm-actions コマンドを使用してアラームを無効化および有効化できます AWS CLI。CloudWatch コンソールで、または の DescribeAlarmHistory API アクションまたは describe-alarm-history コマンドを使用して、アラームの履歴を表示することもできます AWS CLI。CloudWatch は、アラーム履歴を 2 週間保存します。CloudWatch コンソールで、ナビゲーションペインのお気に入りと最近のメニューを選択して、お気に入りおよび最近アクセスしたアラームを設定してアクセスできます。

次のステップとリソース

リレーショナルデータベースを に移行する方法の詳細については AWS クラウド、 AWS 規範ガイダンスウェブサイトの以下の戦略を参照してください。

- リレーショナルデータベースの移行戦略

モニタリング、移行、データ管理に関連するタスクなど、 で実行されている特定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については、データベース移行パターンを調べることができます。 step-by-step AWS クラウド

そのページのフィルターを使用して、 AWS サービス (Amazon RDS や Amazon Aurora への移行など)、ワークロード (MySQL や MariaDB データベースを含むオープンソースなど)、または計画された使用 (本番稼働やパイロット) でパターンを検索します。

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 FAQs
- Performance Insights FAQs
- Amazon CloudWatch Metrics Stream を使用して、Amazon RDS Performance Insights カウンターメトリクスをサードパーティーの Application Performance Monitoring サービスプロバイダーに配信する (AWS ブログ記事)
- Amazon RDS と Amazon Aurora MySQL をモニタリングするための Amazon CloudWatch ダッシュボードの作成 (AWS ブログ記事)
- Performance Insights を使用した Amazon RDS for MySQL のチューニング (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 メトリクスを CloudWatch に公開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 規範ガイダンスが提供する戦略、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します。
- 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: オンプレミスの Oracle データベースを の Oracle 用 Amazon Relational Database Service (Amazon RDS) に移行します AWS クラウド。
- 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: カスタマーリレーションシップ管理 (CRM) システムを Salesforce.com に移行します。
- 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: オンプレミスの Oracle データベースを の EC2 インスタンス上の Oracle に移行します AWS クラウド。
- 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。

- 使用停止 — 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コントロール」を参照してください。

抽象化されたサービス

「 マネージドサービス」を参照してください。

ACID

アトミック性、一貫性、分離性、耐久性を参照してください。

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。より柔軟ですが、アクティブ/パッシブ移行よりも多くの作業が必要です。

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の方法。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

行のグループで動作し、グループの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 などがありますMAX。

AI

「人工知能」を参照してください。

AIOps

「人工知能オペレーション」を参照してください。

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。

アプリケーションコントロール

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 の需要要素であり、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?」を参照してください。

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

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の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

個人や組織を混乱させたり、損害を与えたりすることを意図したボット。

BCP

「事業継続計画」を参照してください。

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Data in a behavior graph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。エンディアン性も参照してください。

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

2 つの異なる同一の環境を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 (青) 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 (緑) 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、ボットハーダーまたはボットオペレーターとして知られる 1 人の当事者が管理しているボットのネットワーク。ボットは、ボットとその影響をスケールするための最もよく知られているメカニズムです。

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、 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 AWS 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、ホワイトペーパー AWSでのコンテナ化されたマイクロサービスの実行 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。

Canary デプロイ

エンドユーザーへのバージョンのスローリリースと増分リリース。確信が持てば、新しいバージョンをデプロイし、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

「データキャプチャの変更」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します。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テクノロジー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 に移行するときに通常実行する 4 つのフェーズ AWS クラウド :

- 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

「設定管理データベース」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または が含まれますBitbucket Cloud。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などのビジュアル形式から情報を分析および抽出する AI の分野。例えば、Amazon SageMaker AI は CV 用の画像処理アルゴリズムを提供します。

設定ドリフト

ワークロードの場合、設定が想定状態から変化します。これにより、ワークロードが非準拠になる可能性があり、通常は段階的かつ意図的ではありません。

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

一元管理とガバナンスを備えた分散型の分散型データ所有権を提供す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。信頼できる ID のみが、予想されるネットワークから信頼できる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通常、大量の履歴データが含まれており、通常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る。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。

ディープラーニング

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。この戦略を採用するときは AWS、 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て、リソースの安全性を確保します。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。

デプロイ

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、Implementing security controls on AWSのDetective controls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

スタースキーマでは、ファクトテーブル内の量的データに関するデータ属性を含む小さなテーブル。ディメンションテーブル属性は通常、テキストフィールドまたはテキストのように動作する

離散数値です。これらの属性は、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一般的に使用されます。

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)

災害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に使用する戦略とプロセス。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

「開発値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、応答時間を短縮できます。

電子データ交換 (EDI)

組織間の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プロセス。

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

星スキーマの中央テーブル。事業運営に関する量的データを保存します。通常、ファクトテーブルには、メジャーを含む列とディメンションテーブルへの外部キーを含む列の 2 つのタイプの列が含まれます。

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で段階的なテストを使用する哲学。これはアジャイルアプローチの重要な部分です。

障害分離の境界

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

同様のタスクの実行を求める前に、タスクと必要な出力を示す少数の例を LLM に提供します。この手法は、プロンプトに埋め込まれた例 (ショット) からモデルが学習するコンテキスト内学習のアプリケーションです。少数ショットプロンプトは、特定のフォーマット、推論、またはドメインの知識を必要とする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コントロール」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

段階的なアプローチを使用する代わりに、変更データキャプチャによる継続的なデータレプリケーションを使用して、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)

一般化およびラベル付けされていないデータの大規模なデータセットでトレーニングされている大規模な深層学習ニューラルネットワーク。FMs は、言語の理解、テキストと画像の生成、自然言語の会話など、さまざまな一般的なタスクを実行できます。詳細については、「基盤モデルとは」を参照してください。

G

生成 AI

大量のデータでトレーニングされ、シンプルなテキストプロンプトを使用してイメージ、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る AI モデルのサブセット。詳細については、「生成 AI とは」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 ワークフローはレガシーと見なされ、トランクベースのワークフローはモダンで推奨されるアプローチです。

ゴールデンイメージ

そのシステムまたはソフトウェアの新しいインスタンスをデプロイするためのテンプレートとして使用されるシステムまたはソフトウェアのスナップショット。例えば、製造では、ゴールデンイメージを使用して複数のデバイスにソフトウェアをプロビジョニングし、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、、Amazon GuardDuty AWS Security Hub、、 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。

ハイアベイラビリティ (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に使用されるデータセットから保留される、ラベル付きの履歴データの一部。モデル予測をホールドアウトデータと比較することで、ホールドアウトデータを使用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

「産業用モノのインターネット」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

既存のインフラストラクチャを更新、パッチ適用、または変更する代わりに、本番環境のワークロード用に新しいインフラストラクチャをデプロイするモデル。イミュータブルインフラストラクチャは、本質的に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、予測性が高くなります。詳細については、 AWS 「 Well-Architected フレームワーク」の「イミュータブルインフラストラクチャを使用したデプロイ」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 によって導入された用語で、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ゼーションを指します。

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。

産業分野における 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、「Building an industrial Internet of Things (IIoT) digital transformation strategy」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 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

「モノのインターネット」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)

大量のデータに対して事前トレーニングされた深層学習 AI モデル。LLM は、質問への回答、ドキュメントの要約、テキストの他の言語への翻訳、文の完了など、複数のタスクを実行できます。詳細については、LLMs」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

「ラベルベースのアクセスコントロール」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。エンディアン性も参照してください。

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または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、不正アクセスにつながる可能性があります。マルウェアの例としては、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。Amazon Simple Storage Service (Amazon S3) と Amazon DynamoDB は、マネージドサービスの例です。これら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、制御するためのソフトウェアシステムで、原材料を工場の完成製品に変換します。

MAP

「移行促進プログラム」を参照してください。

メカニズム

ツールを作成し、ツールの導入を推進し、調整を行うために結果を検査する完全なプロセス。メカニズムは、動作中にそれ自体を強化して改善するサイクルです。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。

メッセージキューイングテレメトリトランスポート (MQTT)

リソースに制約のある IoT デバイス用の、パブリッシュ/サブスクライブパターンに基づく軽量 machine-to-machine (M2M) 通信プロトコル。

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)

に移行するためのビジネスケースを検証するための情報を提供するオンラインツール AWS クラウド。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 に移行するために使用するアプローチ AWS クラウド。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、「」の「アプリケーションをモダナイズするための戦略 AWS クラウド」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、『』の「アプリケーションのモダナイゼーション準備状況の評価 AWS クラウド」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

「移行ポートフォリオ評価」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。

ミュータブルインフラストラクチャ

本番ワークロードの既存のインフラストラクチャを更新および変更するモデル。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

「オリジンアクセスコントロール」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

「 オペレーションの統合」を参照してください。

OLA

「運用レベルの契約」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。

オープンプロセス通信 - 統合アーキテクチャ (OPC-UA)

産業用オートメーション用のmachine-to-machine (M2M) 通信プロトコル。OPC-UA は、データの暗号化、認証、認可スキームを備えた相互運用性標準を提供します。

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)

インシデントや潜在的な障害の理解、評価、防止、または範囲の縮小に役立つ質問とそれに関連するベストプラクティスのチェックリスト。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するハードウェアおよびソフトウェアシステム。製造では、OT と情報技術 (IT) システムの統合が、Industry 4.0 変換の主な焦点です。

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。

オリジンアクセスコントロール (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コントロールが可能です。

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

「プログラム可能な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

アクセス許可を定義 (アイデンティティベースのポリシーを参照)、アクセス条件を指定 (リソースベースのポリシーを参照)、または の組織内のすべてのアカウントに対する最大アクセス許可を定義 AWS Organizations （ サービスコントロールポリシーを参照) できるオブジェクト。

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。詳細については、マイクロサービスでのデータ永続性の有効化 を参照してください。

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、「移行準備状況ガイド」を参照してください。

述語

`true` または を返すクエリ条件。一般的に`false`は `WHERE`句にあります。

述語プッシュダウン

転送前にクエリ内のデータをフィルタリングするデータベースクエリ最適化手法。これにより、リレーショナルデータベースから取得して処理する必要があるデータの量が減少し、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、Implementing security controls on AWSのPreventative controls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内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じてプライバシーを考慮するシステムエンジニアリングアプローチ。

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を防ぐ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、プロビジョニング前にリソースをスキャンします。リソースがコントロールに準拠していない場合、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)

設計、開発、発売から成長と成熟まで、製品のデータとプロセスのライフサイクル全体にわたる管理。

本番環境

「環境」を参照してください。

プログラム可能なロジックコントローラー (PLC)

製造では、マシンをモニタリングし、製造プロセスを自動化する、信頼性の高い適応可能なコンピュータです。

プロンプトの連鎖

1 つの LLM プロンプトの出力を次のプロンプトの入力として使用して、より良いレスポンスを生成します。この手法は、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善または拡張したりするために使用されます。これにより、モデルのレスポンスの精度と関連性が向上し、より詳細でパーソナライズされた結果が得られます。

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。

パブリッシュ/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。スケーラビリティと応答性を向上させます。たとえば、マイクロサービスベースの MES では、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できるチャネルにイベントメッセージを発行できます。システムは、公開サービスを変更せずに新しいマイクロサービスを追加できます。

Q

クエリプラン

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る手順などの一連のステップ。

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

責任、説明責任、相談、通知 (RACI) を参照してください。

RAG

「取得拡張生成」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

責任、説明責任、相談、情報 (RACI) を参照してください。

RCAC

「行と列のアクセスコントロール」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。

再設計

「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 とは独立しています。詳細については、AWS リージョン「アカウントで使用できる を指定する」を参照してください。

回帰

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。

プラットフォーム変更

「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

中断に抵抗または回復するアプリケーションの機能。高可用性とディザスタリカバリは、 で回復性を計画する際の一般的な考慮事項です AWS クラウド。詳細については、AWS クラウド「レジリエンス」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。サポートを含めると、そのマトリックスは RASCI マトリックスと呼ばれ、サポートを除外すると 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、Implementing security controls on AWSのResponsive controls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。

取得拡張生成 (RAG)

LLM がレスポンスを生成する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を参照する生成 AI テクノロジー。たとえば、RAG モデルは、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する場合があります。詳細については、「RAG とは」を参照してください。

ローテーション

攻撃者が認証情報にアクセスすることをより困難にするために、シークレットを定期的に更新するプロセス。

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 Management Console 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

「監視コントロールとデータ取得」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。シークレット値は、バイナリ、単一の文字列、または複数の文字列にすることができます。詳細については、Secrets Manager ドキュメントの「Secrets Manager シークレットの内容」を参照してください。

設計によるセキュリティ

開発プロセス全体でセキュリティを考慮するシステムエンジニアリングアプローチ。

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、予防的、検出的、応答的、プロアクティブの 4 つの主なタイプがあります。

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に自動的に応答または修復するように設計された、事前定義された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。自動応答アクションの例としては、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

送信先にあるデータの、それ AWS のサービス を受け取る による暗号化。

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、スループットなど、サービスのパフォーマンス側面の測定。

サービスレベルの目標 (SLO)

サービスレベルのインジケータによって測定される、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、お客様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

セキュリティ情報とイベント管理システムを参照してください。

単一障害点 (SPOF)

システムを中断する可能性のあるアプリケーションの 1 つの重要なコンポーネントの障害。

SLA

「サービスレベルの契約」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、『』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 AWS クラウド」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

1 つの大きなファクトテーブルを使用してトランザクションデータまたは測定データを保存し、1 つ以上の小さなディメンションテーブルを使用してデータ属性を保存するデータベース組織構造。この構造は、データウェアハウスまたはビジネスインテリジェンスの目的で使用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。

監視制御とデータ収集 (SCADA)

製造では、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稼働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潜在的な問題を検出したり、パフォーマンスをモニタリングしたりする方法でシステムをテストします。Amazon CloudWatch Synthetics を使用して、これらのテストを作成できます。

システムプロンプト

LLM にコンテキスト、指示、またはガイドラインを提供して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、ユーザーとのやり取りのルールを確立するのに役立ちます。

T

tags

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「 を他の AWS のサービス AWS Organizations で使用する 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 枚のピザで養う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。詳細については、深層学習システムにおける不確実性の定量化 ガイドを参照してください。

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する行のグループに対して計算を実行する SQL 関数。ウィンドウ関数は、移動平均の計算や、現在の行の相対位置に基づく行の値へのアクセスなどの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 「ワークロード認定フレームワーク」を参照してください。

Write Once, Read Many (WORM)

データを 1 回書き込み、データの削除や変更を防ぐストレージモデル。承認されたユーザーは、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

ゼロデイ脆弱性を利用する攻撃、通常はマルウェア。

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 にタスクを実行する手順を提供しますが、タスクのガイドに役立つ例 (ショット) はありません。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。