New Relic

White Paper

クラウドを正しく使う

New Relic でインストルメントされた AWS Well-Architected フレームワーク

Amazon Web Services（AWS）でクラウドネイティブアプリケーションを開発する場合でも、既存のアプリケーションをモダン化してAWSに移行する場合でも、アプリケーションのライフサイクルにおいて多数の決定を下すことになります。使用する言語や必要なアプリケーションプログラミングインターフェイス（API）など、一部の決定は見解や経験に基づいて行われます。

しかし、直面する決定の多く、たとえばAWSインスタンスのサイズやデータストレージの種類については、データ駆動型のベストプラクティスに従う必要があります。データに導かれた、ベストプラクティスによるアプローチを採用することで、緻密に設計されたアプリケーションとAWSのアプリケーション環境を構築することができます。

AWSのWell-Architectedフレームワーク（WAF）の目的は、「顧客とパートナーが、長期間にわたり拡張できるアーキテクチャーを評価し、設計を実施するための一貫性あるアプローチ」を提供することにあります。

WAFは、セキュアで高パフォーマンス、復元力が高く、効率的なアーキテクチャーを構築してきた数千ものパートナーとの経験に基づいて構築されています。5つの柱、すなわち**オペレーショナルエクセレンス、セキュリティ、信頼性、パフォーマンスの効率性**、そして**コスト最適化**に基づき、WAFは詳細なガイダンスとベストプラクティス、さらには長期間にわたり拡張していくアーキテクチャーの評価と実施に関する基本的な質問事項を提供します。

New Relicでインストルメンテーションを行う場合、WAFの柱はデジタル企業の運営のための実行可能なオブザーバビリティ プラットフォームの一部となります。New Relic Oneは、デバイスやアプリケーション、インフラストラクチャを通じたエンドユーザー体験から、完全な可視性とコンテキスト、実行可能な洞察を提供します。これは5つの柱それぞれから十全なメリットを得るために必須であり、顧客のために尽力する企業にとって鍵となる成功要因です。

New Relicを使用すると、以下の内容を理解できます。

- 自社のアプリケーションと環境をどの程度WAFと連携できるか
- WAFとの連携を強化する機会がどこにあるか
- 既存のAWS環境におけるリスクがどこにあり、そのリスクをどのように低減できるか

このホワイトペーパーは、AWS WAFのベストプラクティスをサポートするNew Relicの使用を開始するのに役立ちます。アプリケーションやインフラストラクチャー、AWSサービスから収集されるメトリクスやイベント、ログ、トレース（M.E.L.T.）のテレメトリデータを使用した主要部分のインストルメント化について、New Relicの活用例を挙げながら、それぞれの柱を確認します。

> *「オペレーショナルエクセレンスの柱は、ビジネス価値を提供するためのシステムの運用とモニタリング、そして継続的な改善プロセスと手順に注目します。重要なトピックには、変更の管理と自動化、イベントへの対応、日常業務を正常に管理するための標準の定義が含まれます。」*
>
> **AWS Well-Architectedフレームワーク**

オペレーショナルエクセレンス

この柱には、3つのベストプラクティス領域、すなわち準備、運用、発展が含まれます。これらの領域は、チームが自社のビジネスと顧客のニーズを理解し、求められるビジネス成果の達成を測定するのに役立ちます。New Relicプラットフォームを使用することで提供されるオブザーバビリティと併せてAWSのベストプラクティスを活用する場合、以下の項目を提供することで、オペレーショナルエクセレンスを向上することができます。

- 全チームが単一のロケーションからビジネスの価値と成果を確認できる
- 変化の影響に関する測定とコミュニケーション
- 問題解決の自動化
- インシデント レビューのための貴重な履歴データ

ベストプラクティスの例：KPIダッシュボードを作成する

ビジネス価値を提供するモニタリングシステムの鍵となるのは、アプリケーションのさまざまな側面に関連する主要パフォーマンス指標 (KPI) を単一の場所で表示できる機能です。KPIには、アクティブユーザーの数、ユーザーの居住国、現在のオーダー数、平均オーダー額、リスクのあるユーザーと収益などのメトリクスが含まれます。図1のKPIダッシュボードの例は、これらのKPIを表示するチャートを示しています。

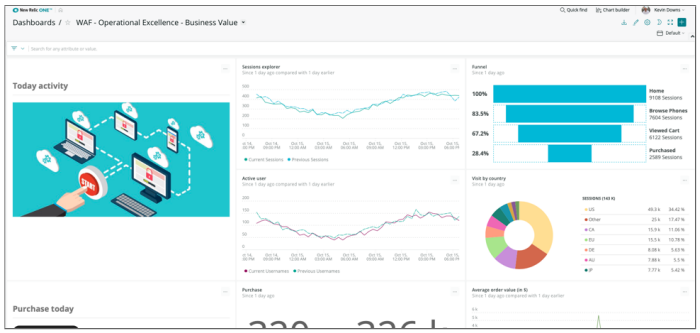


図1. ユーザー、オーダー、リスク：KPIダッシュボードに表示される重要なビジネス価値のKPI

変化を管理、自動化するため、デプロイメントマーカーを追加して測定を開発プロセスに統合します。WAFの設計原則では、小規模かつ頻繁で、容易に元へ戻せる変更を実施するよう推奨しています。インストルメンテーションを適切に作成することで、チームはそれらの変更がシステムに与える影響をすべて把握できます。各変更の開始前、実施中、完了後の具体的かつ測定可能なメトリクスを取得することで、変更を個別に最適化でき、システム内で発生するその他の作業への影響を低減できます（図2を参照）。

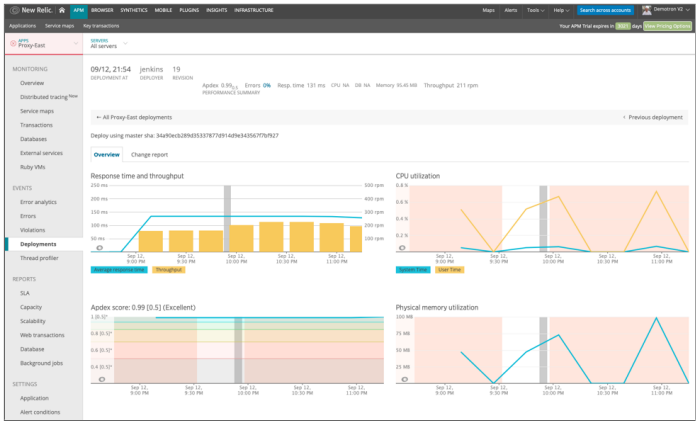


図2. Jenkinsとのインテグレーションを通じてNew Relicに送信されるデプロイメントマーカー。各変更の開始前、実施中、完了後の測定可能なメトリクスを表示

デプロイメントと、コードやインフラストラクチャの変更がエンドユーザー体験にどのような影響を与えるか追跡することも重要です。デプロイメントの追跡は、急激な、長期的な、または漸進的なアプリケーションの劣化の根本原因を判定する重要な方法です。これは、New RelicがWAFにアプリケーションと顧客体験の情報を組み込み、拡張するひとつの方法です。

ベストプラクティスの例：ソリューションを自動化する

Amazon CloudWatchのアラートを使用する場合でも、New Relicプラットフォームのイベント検索を選択する場合でも、解決を自動化することは重要です。単純で反復的なインシデントへの対応タスクを自動化することで、効率性を高め、インシデントの影響を最小限に抑えます。適切な自動化を導入することで、障害のあるアプリケーションコンポーネントを、通知の発行後ではなく、アラートの閾値に達した時点ですぐに無効化または隔離できます。

たとえば、デジタルメディア企業のアプリケーション管理チームは、コメントサービスにエラーがある場合、ウェブサイトからコメント機能を削除できることを望んでいるとします。この場合、次のことが可能です。

1. 記事のコメント投稿に関連するUIコンポーネントを有効化または無効化する機能フラグを切り替えるフロントエンドのウェブアプリケーションに、エンドポイントを追加する。
2. コメント作成サービスで保持されたエラー率で設定された閾値で、アラートポリシーを作成する。
3. POSTリクエストを送信するwebhook通知チャネルを、このエンドポイントおよび標準のチーム通知チャネルに割り当てる。

このシナリオでは、コメント作成システムのエラーによりwebhookが起動し、ウェブサイトからコメント作成UIを削除します。ユーザーは以後も、コメント作成サービスが生成するエラーを目にすることなく、サイトの中心的機能を使用できます。アプリケーションは安定しつつも劣化した状態を維持するため、チームはユーザーのアクセスを妨げるというプレッシャーを感じることなく、復旧作業に集中することができます。

ベストプラクティスの例：クローズドループのフィードバック

もうひとつのベストプラクティスは、イベント完了後にループを閉じることです。WAFの設計原則では、すべてのオペレーション上の失敗から教訓を学ぶ必要があると述べています。インシデントの解決後、主要なステークホルダーと参加者には、レビューを確立するためにインシデントの正確かつ完全なドキュメンテーションを入手することが推奨されています。

ドキュメントには、少なくとも以下の内容を記載することが推奨されます。

- 根本原因の分析
- 修正手順とその結果の年表および要約、またそれらが成功したかどうか
- ユーザー体験と財務上の損失の観点から見たビジネスへの影響の測定結果（可能な場合）
- インシデント解決のために、エンジニアリングとオペレーションに費やした時間の測定
- 再発を防ぐためのシステムまたは機能の改善に関する推奨事項
- プロセスとコミュニケーションの改善に関する推奨事項

共有ドライブフォルダやwikiなど、見やすく検索可能なリポジトリに事後検証レポート（例として図3を参照）を保存します。このプロセスでは、罰を与えたり叱責するのではなく、建設的な学習と改善に重点を置くことが不可欠です。

Post mortem	Comments
Date	March 1, 2018
Executive summary	From approximately 1:45PM until 2:30PM, users could not add items to their carts, which prevented any checkouts from occurring during the incident period.
Root cause	We determined that a change was made to the CSS rules on the product detail page that effectively disabled the Add to cart button.
Timeline	• 1:50PM: Successful checkouts < 10 for 5 minutes alert triggered; assigned to Alice. • 1:55PM: After reviewing the ecommerce team dashboard, Alice determined that the threshold was breached immediately following a deploy by Bob. She notified him of the incident. • 2:00PM: Alice and Bob began troubleshooting. Attempts at recreating the issue in production were successful. • 2:20PM: Bob determined that his change to the CSS on the product detail page disabled the Add to cart button. He deployed a hotfix. • 2:30PM: Functionality was restored and the incident was resolved.
Impact	No checkouts were completed during the duration of the incident. Our typical revenue for a Thursday during this timeframe is $30,000.
Recommendations	We have been discussing implementing New Relic Synthetics for awhile now. If we had a Synthetic check on the checkout process, this issue would have been detected immediately. We should also implement more thorough unit tests in the front-end web app.

図3. 事後検証レポートの例

「セキュリティの柱では、情報とシステムの保護に重点が置かれています。主要なトピックには、データの機密性と完全性、特権的管理者とその権限の特定と管理、システム保護、セキュリティイベントを検知するためのコントロールの確立があります。

AWS Well-Architectedフレームワーク

セキュリティ

この柱には、5つのベストプラクティス領域があります。すなわち、管理の特定とアクセス、検知コントロール、インフラストラクチャの保護、データの保護、そしてインシデント対応です。基本的に、ベストプラクティスはデータとシステムの保護のため、組織がいかにAWSのセキュリティ機能とサービスを使用するべきか（AWSマーケットプレイスを通じて使用できるサードパーティのセキュリティソリューションも同様）に注目する一方で、New Relicは以下の3つの重要なセキュリティ原則の実行をサポートします。

1. セキュリティイベントに迅速に対応する
2. セキュリティアラートをモニタリングして追跡する
3. セキュリティ上のリスクを低減する

New Relicは、あらゆるアプリケーションイベントにアラートを発し、セキュリティコンテキストのあるログエントリを転送できます。

ベストプラクティスの例：セキュリティイベントを検知する

セキュリティイベントを検知するため、New Relicは主に2つの戦略を推奨しています。

第1に、アプリケーション内にセキュリティ通知を構築し、それらの通知をログやカスタムイベントの形式でNew Relicに送信します。New Relic Logsでは、迅速で拡張可能なログ管理プラットフォームが提供され、ログデータとその他のテレメトリデータを結びつけることができます。これにより、セキュリティ通知をアプリケーションのコンテキストに応じて確認できるようになります（図4を参照）。

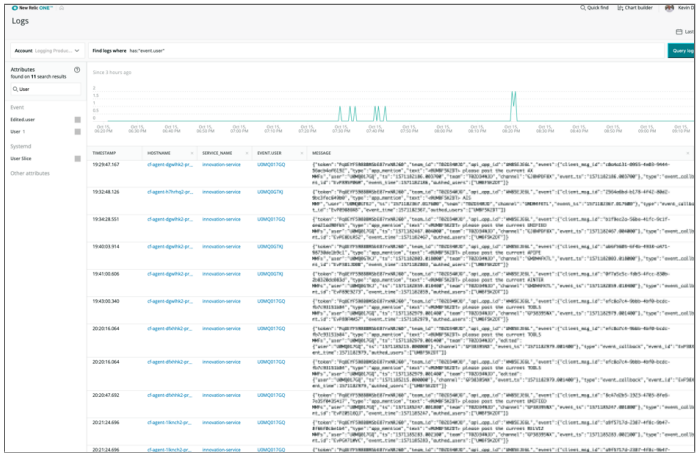


図4. New Relic Oneで表示されるログデータ（セキュリティ保護のためイメージはピクセル化されています）

第2に、AWSや貴社が使用するその他あらゆるサードパーティのセキュリティサービスとNew Relicを統合します。AWS CloudTrailとのインテグレーションにより、AWSのアカウント活動は、AWS Identity and Access Management（IAM）に関するイベントを含むその他のセキュリティ情報とのコンテキストに応じて取得、表示されます（図5を参照）。

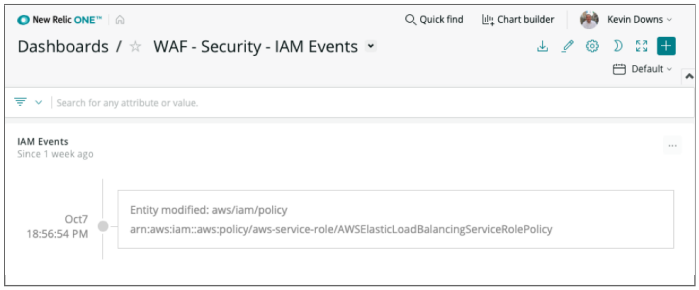


図5. New Relicダッシュボードで表示されるIAMイベント

New Relic

ベストプラクティスの例：セキュリティアラートをモニタリングする

その他の側面は、潜在的なセキュリティリスクに関するモニタリングと報告です。含めるべき重要なセキュリティインジケーターは、オペレーティングシステムとそのバージョン、また使用されているAmazon Machine Images (AMI)です。セキュリティアーキテクトは、基盤となるシステムが古くなっているかどうか、またアプリケーションに影響を与えるセキュリティアラートが発行されているか知ろうとします。

この情報を取得することで、セキュリティ問題を抱えるホストについて、セキュリティチームからガイダンスを得ることができます。更新が必要なホストを簡単に見つけるには、New Relic内のフィルタ機能を使用します（図6を参照）。ホストがAWSクラウド内にある場合、New Relicインテグレーションで使用できる追加タグ（インスタンスの種類、地域、アプリケーションやステータスなどのカスタムタグ）でフィルタをかけることができます。

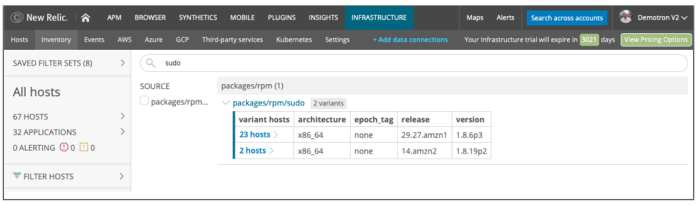


図6.New Relic インフラストラクチャは、セキュリティの脅威があるパッケージのバージョンを含む2つのホストを表示

ベストプラクティスの例：セキュリティ上のリスクを低減する

セキュリティ上のリスクを低減するひとつの方法は、AWSサービスを活用して、自社チームが自主管理する必要のあるテクノロジーの数を減らすことです。これは、新規アプリケーション向けにマネージドサービスの使用を選択するか、既存のアプリケーションをマネージドサービスにリプラットフォーミングすることで実行できます。たとえば、Amazon Auroraのようなマネージドサービスへのデータベースのプラットフォーム再構築が可能です。多くのセキュリティタスクが管理されているため、他の領域に注力できるようになります。このようにして、サービス管理の責任をAWSに移管します。

AWS環境を確保する上での自身の役割を詳しく理解するため、自身の責任範囲とAWSの責任範囲を説明するAWS責任共有モデルをレビューします。

New Relicは、AWSの管理サービスへのプラットフォーム再構築の潜在的な候補を含め、アプリケーションのすべてのコンポーネントを表示することができます（図7を参照）。

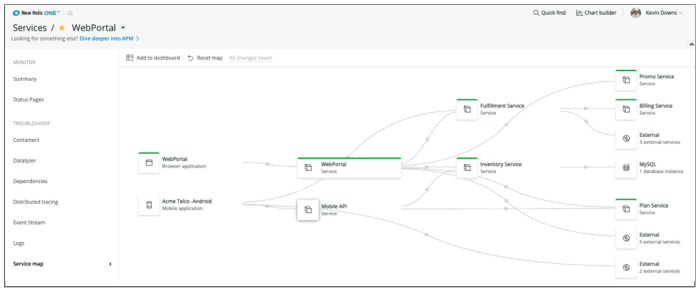


図7. アプリケーションを構成するコンポーネントを示したサービスマップ

AWSマネージドサービスを選択することで、脆弱性とパッチ管理について自ら責任を持たなければならないインフラストラクチャの数を減らすことになります。依然として、データ保護やアクセス権を持つ人員の管理のための適切なルールを提供する必要はありますが、オペレーティングシステムのパッチや更新など、低レベルのセキュリティ問題はAWSによって管理されます。

理解すべきもうひとつの重要なメトリックは、組織がマネージドサービスを使用することにより、セキュリティの範囲と労力をどの程度削減しているかということです。セキュリティの観点から、追跡、管理、およびパッチの適用が不要となったデータベースはいくつありますか？データベースの数に加えて、データベースの種類（MySQL、Oracle、Microsoft SQL Serverなど）とバージョン（1.2、2.7、5.4.2など）の数はそれ

ぞれいくつありますか？保護する必要のあるデータベースの種類とバージョンの数が増えるほど、セキュリティ上の課題はより大きくなります。

> *「信頼性の柱は、ビジネスや顧客の需要に応えるために、失敗を予防し、迅速に復旧するための能力にフォーカスします。主要なトピックには、設定、部門横断型プロジェクトの要件、復旧計画、変更の管理方法に関する基本的要素があります。*
>
> **AWS Well-Architectedフレームワーク**

信頼性

信頼性の柱には、3つの主なベストプラクティスが含まれます。それは、基盤、変更管理、障害管理です。New Relicのデータにより、安定して予測可能な、可用性の高い信頼性あるシステムを構築することができます。New Relicを使用することで、優れた顧客体験の提供という目標に向け、初期段階のエラーを早期に発見し、不具合を予防し、発生した不具合から迅速に復旧できるようになります。

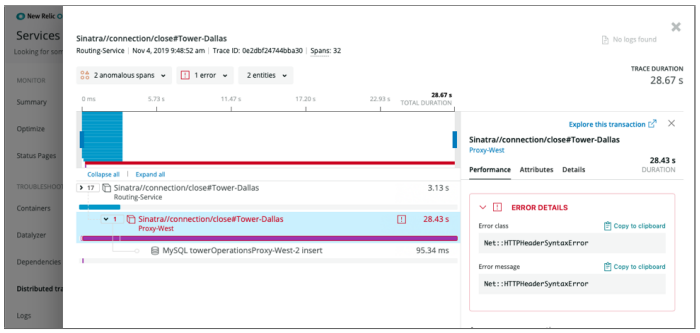
ベストプラクティスの例：エラーを早期に発見する

信頼性あるアプリケーションを構築し、稼働環境での不具合を予防するためのベストプラクティスは、継続的インテグレーションと継続的配信（CI/CD）のコンセプトを使用して可能な限り早期にエラーを発見することです。

継続的インテグレーションとは、開発者がコード互換性に関して厳密な管理を維持することを指します。具体的に言うと、開発者が変更を行うたびに、ビルドを自動化し、コードをテストするプロセスです。しかし、継続的インテグレーションは、企業が高品質なソフトウェアを迅速にユーザーの手に届けるための方法の一部でしかありません。この方向性の次のステップが、継続的配信です。

継続的配信を行う開発者は、常にデプロイ可能ですぐに稼働環境に移れるコードを開発します。継続的配信は、品質を向上させながら市場投入までの時間を加速させる、一連のソフトウェア開発の実践と方策です。

現代のアプリケーションとウェブサイトは、ますます多くの相互接続サービスを使用するようになっています。多数のサービスまたはマイクロサービスに依存したアプリケーションアーキテクチャーのことを、一般的に分散システムと呼びます。New Relicプラットフォームのディストリビューティッド（分散）トレーシングを使用することで、アプリケーションへのリクエストから生じるアクティビティの追跡が可能になります。複雑なシステム内を移動するリクエストのパスをトレースできれば、そのパスに沿ったコンポーネントのあらゆるレイテンシを発見できるようになります。コンポーネントのレイテンシを追跡することで、アプリケーションのボトルネックを見つけることができます。アプリケーション内のボトルネックが、往々にしてエラーの発生場所です。エラーの発見が早ければ、修正も早くなり、アプリケーション全体のパフォーマンスと信頼性が高まります。



ディストリビューティッド（分散）トレーシングを使用して、エラーを迅速に発見する。この場合は、HTTPHeaderSyntaxErrorが28秒の遅延の原因であることがわかる。

New Relic

ベストプラクティスの例：SLAに対する稼働時間を追跡する

今日のデジタルビジネスでは、重要なアプリケーションの稼働時間に対して厳しい要件が設けられています。最初に、サービスレベルアグリーメント（SLA）と稼働時間の要件を明確に理解しましょう。次に、信頼性に関するメトリクスを比較し、SLAに対するアプリケーションのパフォーマンスレベルを確認します。アプリケーション可用性の割合はどのくらいか？ユーザーがエラーの影響を受ける頻度は？New Relicは、稼働時間に関するKPIとSLAのモニタリングをサポートします（図8を参照）。

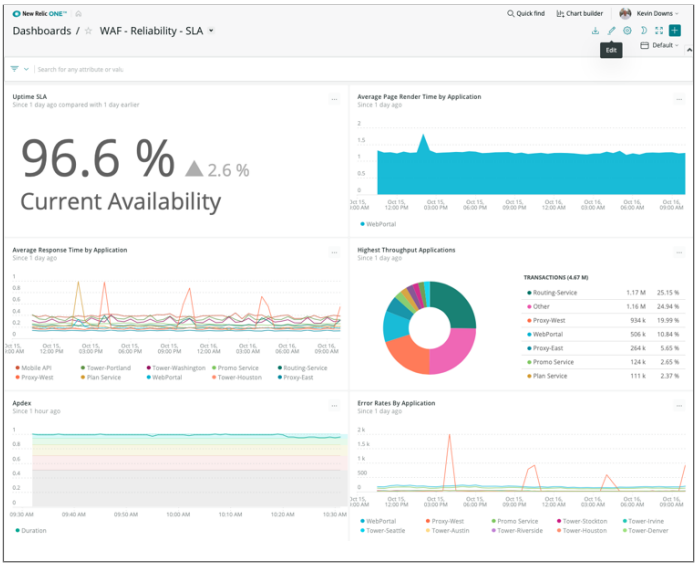


図8．SLAのモニタリングは、アプリケーションの信頼性を理解するためのベストプラクティス。

ベストプラクティスの例：スケーリング性能を理解する

アプリケーションの多くは、ほんの数か月先のことであるとしても、将来のスケーリングに向けた要件の予測と予想が困難です。アプリケーションが少数のユーザーのみのサポートを計画している場合は、アプリケーションのスケーリングについてそれほど心配することはありません。しかし、ほとんどのアプリケーションでは、増加を続ける多数のユーザーをサポートすることが予想されます。アプリケーションが確実にユーザー数の増加（減少）をサポートできるようにするため、スケーリングは自動的に実行する必要があります。自動スケーリングのルールを調整するために時間を費やすべきではありません。New Relicインテグレーションと AWS Auto Scalingを併用することで、Auto Scalingグループに関するメトリクスとインベントリデータを取得できます（図9を参照）。

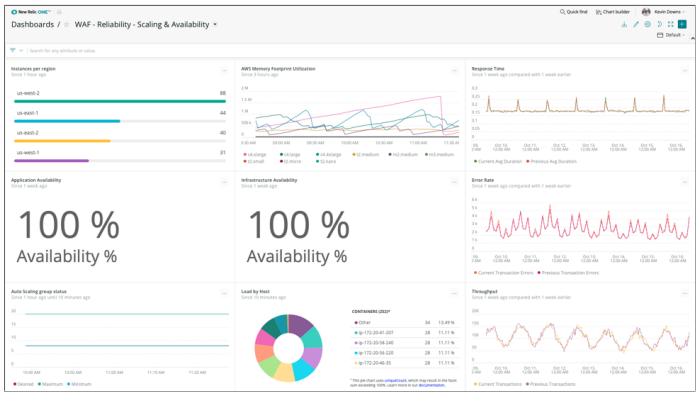


図9. アプリケーションのスケーリングおよび可用性の情報

「パフォーマンス効率性の柱は、ITの使用とコンピューティングリソースの効率性に注目します。鍵となるのは、ワークロード要件に基づいた正しいリソースの種類とサイズの選択、パフォーマンスのモニタリング、ビジネスニーズの発展に伴う効率性を維持するための情報に基づいた決定です。

AWS Well-Architectedフレームワーク

パフォーマンス効率

この柱は、コンピューティングリソースを効率的に使用してシステム要件を満たし、需要の変化やテクノロジーの進化に合わせて効率を維持することを目的としています。この柱における4つのベストプラクティス領域である、選択、レビュー、モニタリング、トレードオフは、それぞれがデータを

根拠にして、効率的なパフォーマンスを維持するための適切なインフラストラクチャーとテクノロジーの選択および発展をサポートします。

New Relicでは、AWSでのパフォーマンス効率性に関して、盲点を排除してすべてをインストルメント化できます。現状での環境のベースラインを設定してから、インフラストラクチャーの適正サイズに関する情報に基づいた決定を行います。実際のユーザーKPIを使用して、ベストパフォーマンスを実現するためのクラウドリソースを配置する地理的な場所を決定します。最後に、よりよい成果を提供するための試みとして、新しいクラウドテクノロジー（Amazon EKSやAWS Lambdaなど）をモニタリングします。

ベストプラクティスの例：情報に基づいた決定を行う

アプリケーションのリソース使用状況を最適化するため、まずNew Relicを使用してAWS環境のベースラインパフォーマンスデータを収集します。現状を理解することで、どこにどうやって改善努力を向けるべきかについて、情報に基づいた決定を行うことができます。

AWSクラウドでは、顧客がアプリケーションのホスティングにおいて効率的なパフォーマンスを達成できるように、多数の高度なテクノロジーを提供しています。主なテクノロジーとして、Amazon Elastic Compute Cloud（EC2）、FaaS（Function as a Service）のAWS Lambda、Amazon Elastic Kubernetes Service（EKS）、さらにAmazon Relational Database Service（RDS）内で多くのデータベースを利用できます。これらの、サイズが適正化され、モニタリングされているクラウドサービスは、顧客需要の変化や、技術の進歩に応じて、アプリケーションのシステム要件を満たすことができます。

ベストプラクティスの例：環境のサイズを適正化する

New Relicは、アプリケーションがインフラストラクチャのパフォーマンスにどのように影響するか理解するのに役立ち、環境のサイズを適正化するために重要なデータ駆動型のアプローチを提供します。図10の例では、c4.largeとc4.xlargeがCPUに十分に活用されていないことが明らかです。これは、下層のティアC（計算）のインスタンスを使用すべきであることを示唆しています。

しかし、これらのインスタンスのメモリ使用には、どちらも経時的に80%から20%までの幅があります。CPUとメモリの両方の情報を確認することで、これらのインスタンスをメモリに最適化されたインスタンスに置き換えることができます。このケースでは、メモリ向けにサイズを適正化することで、CPUも増加し、このアプリケーションのパフォーマンスをより効率化することができます。（備考：図10の右側にあるチャートでは、適正サイズの検討時に顧客体験がどのような影響を受けるか観察できます。）

図10.環境のサイズを適正化するには、複数のKPIを考慮する必要があります。

ベストプラクティスの例：ユーザーのロケーションを最適化する

パフォーマンスの検討におけるもうひとつの要素はユーザーの地域性です。ユーザーがいるロケーションと地域を合わせていますか？もしそうなら、各ロケーションでどのようなパフォーマンスを体験していますか？AWSでは、数分のうちに世界中のロケーションに向けて、グローバルに拡張することができます。地理的な強みを活用し、ユーザーのロケーションとビジネスの達成目標に基づいてリソースを配置することで、顧客により効率的にサービスを提供できるようになります。

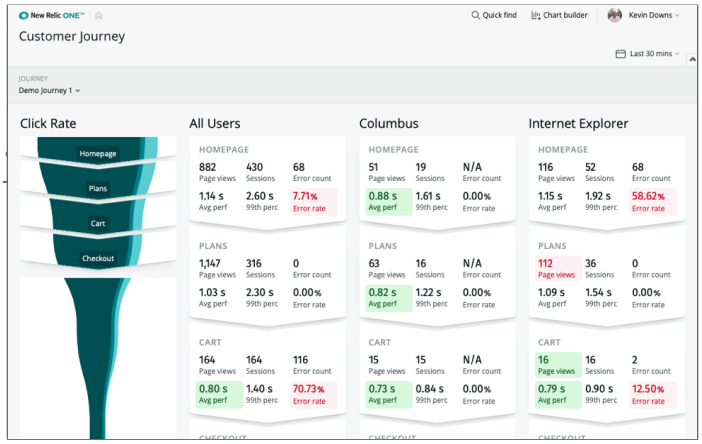


図11. ターゲットのGEOを示すパフォーマンスモニタリング

ベストプラクティスの例：新しいテクノロジーを試す

AWSのメリットのひとつは、頻繁に試みる機能があることです。全体のパフォーマンス向上を実現するために新しいテクノロジーを試してみることは、自分にも顧客にもメリットがあります。

多くの組織が使用し始めている、成長の著しい新技術領域のひとつとして、コンテナ化されたワークロードを管理するミッションクリティカルなプラットフォームであるコンテナとKubernetesがあります。New Relicは、Kubernetesクラスタの多次元的な表現であるKubernetesのクラスタエクスプローラーを提供し、これによりネームスペース、デプロイメント、ノード、ポッド、コンテナ、そしてアプリケーションの詳細を確認できます。クラスタエクスプローラーで、これらの要素がどのように関連しているか理解するためのデータとメタデータを容易に回収できるようになります。

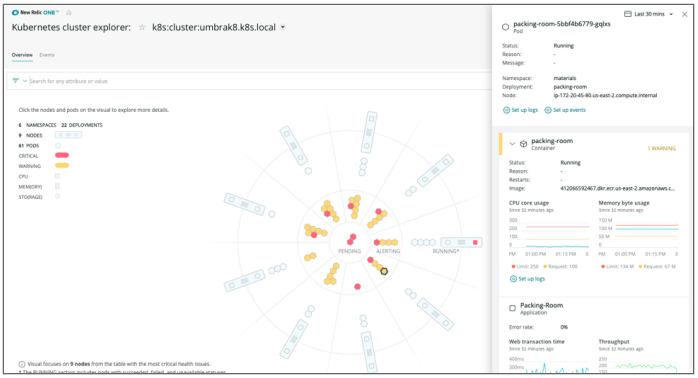


図12. 特定のポッドのKPIを表示するNew RelicのKubernetesクラスタエクスプローラー。

さらに新しい技術として、AWS LambdaなどのFaasS（Function as a Service）というコンセプトがあります。Lambdaは、追加的なリソースの割り当て不要（AWSが代行します）で、スケーリングを行う能力を提供します。これにより、急激な数値変化を経験する企業にパフォーマンス面で優位性を与えます。

New Relicは、サーバーレスAWS Lambda関数のパフォーマンスモニタリングを提供します。これは、CloudWatchデータとコードレベルのインストゥルメンテーションの両方を使用する拡張的なLambdaモニタリングであり、より緻密なモニタリング体験を提供します（図13を参照）。New RelicのLambdaモニタリングにより、以下の項目を把握できます。

- 持続時間やメモリ使用、コールドスタート、例外、トレースバックなどのパフォーマンスデータを含む、Lambda関数のすべての呼び出し。
- その他のAWSサービス（簡易通知サービス（SNS）トピックのメッセージ公開やDynamoDBテーブルに配置された項目など）の呼び出し、およびこれらのサービスの運用とターゲット
- ディストリビューティッド（分散）トレーシング経由でLambdaにつながるリクエストパスと、自社環境内においてLambdaがいかにその他の分散トレーシング範囲に影響を与えるか。
- Lambdaをトリガしたソースについての情報

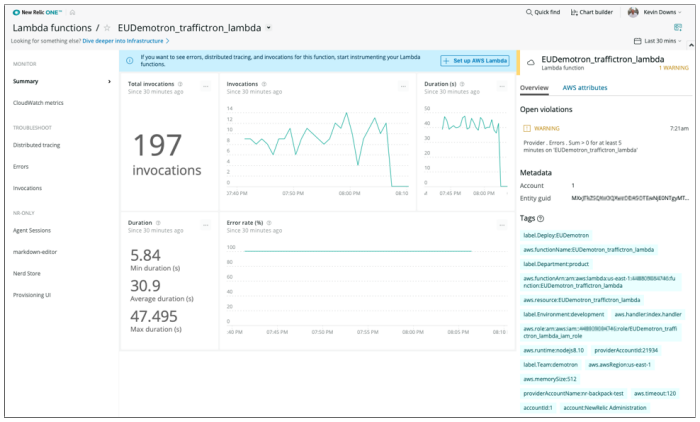


図13.AWS LambdaのNew Relicモニタリング

「コスト最適化は、不要なコストを防ぐことに注力します。主要なトピックは、資金支出先の理解と管理、リソースタイプの最適かつ正しい数の選択、経時的支出の分析、過剰支出をすることなくビジネスニーズを満たすためのスケーリングです。

AWS Well-Architectedフレームワーク

コストの最適化

New Relicは、コスト最適化の柱を新たなレベルに引き上げ、4つのベストプラクティス領域である、支出の認識、コスト効率の高いリソース、需要と供給のマッチング、経時的最適化の適用をサポートします。コスト最適化は単純な支出額ではなく、支出されたアプリケーションの影響であるため、New Relicでは、顧客別のイベントを通じて地域ごとの収益やパフォーマンス影響などの成果を追跡するため、ビジネスデータを組み込むことができます。

ベストプラクティスの例：投資を目的と合致させる

たとえば、チリでのパフォーマンスが望ましいレベルに達していないとします。しかし、現在この地域は最優先のターゲットに含まれないため、この地域への投資は合理的ではありません。現在、中国にはごく少数のエンドユーザーしかいないかもしれませんが、この国でのビジネスを成長させるため、企業は大規模投資を行いたいと考えています。そのため、中国のユーザー体験を優先させる必要があります。こうしたパフォーマンスデータと企業の目標の組み合せは、適切な投資を決定する方法に反映する必要があります。

ベストプラクティスの例：ユーザー満足度を追跡する

コストが最適化されているかどうかを知るもう一つの方法として、New Relicでアプリケーションの Apdexスコアを確認することが挙げられます。Apdex（Application Performance Index）は、ウェブアプリケーションやサービスのレスポンスタイムに関するユーザー満足度を測定するための業界標準であり、ユーザーがアプリケーションに満足しているか確認するのに役立ちます。New Relicを使用して、Apdexによる測定結果をアプリケーションで発生したコストと比較できます。比較結果は、アプリケーションが果たす役割に対して適切なレベルの投資であることを示していますか？エンドユーザー体験の目標はどうあるべきでしょうか？目標は、これらのメトリクスとコストのバランスを保ち、予算を超過することなくエンドユーザー体験に関する目的を達成することです。

表14の例では、Apdexと可用性は100%であり、顧客は非常に速いページロードタイムを経験しています。しかし、インフラストラクチャーとデータベースの使用、またキューの長さを見ると、これらの値が非常に低いことがわかります。これは、クラウドインフラストラクチャー環境が過剰供給になっているからです。これらの要素のサイズを最適化すれば、クラウド支出とエンドユーザー体験についてバランスの改善を実現できます。

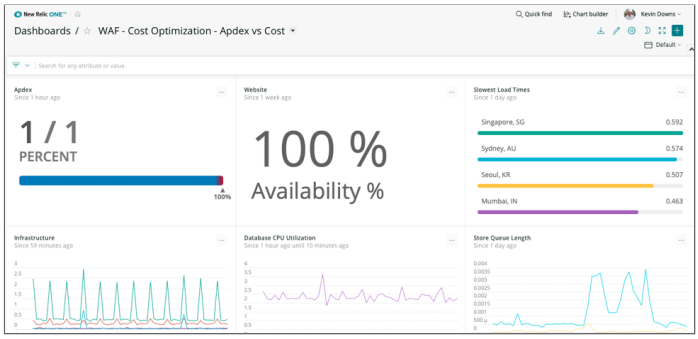


図14.Apdexとコストの比較により、望ましい支出とパフォーマンスが合致するようインフラストラクチャーを最適化できます。

また、New RelicはCloud Optimizeツールを提供しています。このツールを使用して、過剰供給または供給不足のインスタンスを迅速に特定できます（図15を参照）。

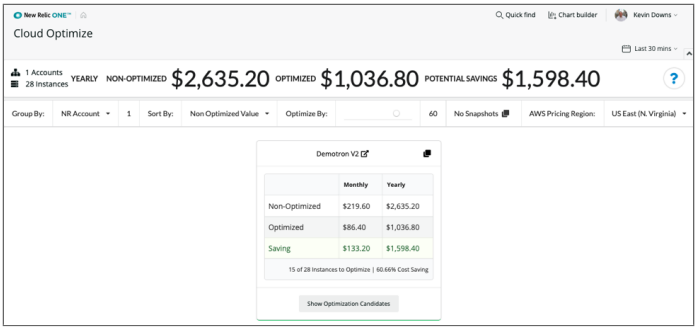


図15. 潜在的なコスト削減を示すNew RelicのCloud Optimizeツール。

結論

クラウド環境に関する決定を行う際のガイドとしてAWS Well-Architectedフレームワークを使用することで、モダン化された新規アプリケーション向けの、信頼性がありスケール化の可能な、セキュアで効率がよく、コスト効率の高い環境の設計が確保されます。その結果として構築された環境では、DevOpsチームが、組織で達成したいビジネス成果をもたらす安定した効率性の高いシステムを開発・運用できます。

New RelicでインストルメントされたWAFのベストプラクティスは、デジタル企業内で測定可能かつ観察可能なものとなり、顧客が求める高品質なアプリケーションの配布ができるようになります。

AWS環境からより多くの恩恵を得る

モダン化ジャーニーのすべてのステージにおいてオブザーバビリティを得ることで、クラウドのメリットを十分に活用できるようになります。newrelic.comをご参照ください。