

# ゲーム分析パイプライン

## AWS 実装ガイド

*Kyle Somers*

*Greg Cheng*

*Daniel Lee*

*Timur Tulyaganov*

2020 年 5 月

最終更新日: 2020 年 9 月 ([改訂](#)を参照)



Copyright (c) 2020 by Amazon.com, Inc. or its affiliates.

「ゲーム分析パイプライン」ソリューションは、MIT No Attribution (<https://spdx.org/licenses/MIT-0.html>) の条件に基づいてライセンスされます。

## 目次

<b>概要</b> .....	<b>4</b>
コスト .....	5
アーキテクチャの概要 .....	6
<b>ソリューションコンポーネント</b> .....	<b>8</b>
AMAZON KINESIS .....	8
AMAZON API GATEWAY .....	8
AWS LAMBDA 関数 .....	8
AMAZON SIMPLE STORAGE SERVICE .....	9
AWS GLUE .....	10
AMAZON DYNAMODB .....	10
AMAZON CLOUDWATCH .....	11
AMAZON SIMPLE NOTIFICATION SERVICE .....	11
AMAZON ATHENA .....	11
<b>実装に関する考慮事項</b> .....	<b>12</b>
適切なストリーミングデータの取り込み方法の選択 .....	12
ゲームクライアントとの統合 .....	12
ゲームバックエンドとの統合 .....	13
AWS SDK との統合 .....	14
AMAZON QUICKSIGHT との統合 .....	14
デプロイ可能な AWS リージョン .....	14
<b>AWS CLOUDFORMATION テンプレート</b> .....	<b>15</b>
<b>自動デプロイ</b> .....	<b>15</b>
前提条件 .....	15
デプロイの概要 .....	16
ステップ 1. スタックを起動する .....	16
ステップ 2. サンプルゲームイベントを生成する .....	19
ステップ 3. AMAZON ATHENA でサンプルクエリをテストする .....	20
ステップ 4. AMAZON ATHENA を AMAZON QUICKSIGHT に接続する .....	21

ステップ 5. AMAZON QUICKSIGHT ダッシュボードを作成する.....	23
ステップ 6. パイプラインの運用ダッシュボードにリアルタイムのメトリクスを表示して追加する.....	39
<b>セキュリティ.....</b>	<b>41</b>
認証.....	41
保管時の暗号化.....	41
<b>リソース.....</b>	<b>42</b>
<b>付録 A: ソリューションのモニタリング.....</b>	<b>42</b>
ゲーム分析パイプラインの運用ダッシュボード.....	42
アラームと通知.....	43
<b>付録 B: ソリューションのアンインストール.....</b>	<b>44</b>
AWS マネジメントコンソールを使用する.....	44
AWS CLI を使用する.....	45
AMAZON S3 バケットを削除する.....	45
AMAZON QUICKSIGHT 分析とデータセットの削除.....	46
<b>付録 C: 運用メトリクスの収集.....</b>	<b>46</b>
<b>ソースコード.....</b>	<b>48</b>
<b>ドキュメントの改訂.....</b>	<b>48</b>

## このガイドについて

この実装ガイドでは、アマゾン ウェブ サービス (AWS) クラウドに「ゲーム分析パイプライン」ソリューションをデプロイするためのアーキテクチャ上の考慮事項と設定手順について説明します。これには、セキュリティと可用性に関する AWS のベストプラクティスを使用してこのソリューションをデプロイするために必要な AWS のサービスを起動および設定する [AWS CloudFormation](#) テンプレートへのリンクが含まれています。

このガイドは、AWS クラウドにおけるアーキテクチャの設計の実務経験があるゲーム開発者、IT インフラストラクチャアーキテクト、管理者、DevOps プロフェッショナルを対象としています。

## 概要

ゲーム業界では、Games-as-a-Service 運用モデルの採用が増加しています。このモデルでは、ゲームが製品というよりもサービスのようになり、アプリ内課金、サブスクリプション、その他の手法によって繰り返し収益が生み出されることがよくあります。この移り変わりにより、プレイヤーがゲームや関連サービスの機能をどのように使用しているかをより深く理解することが重要になります。この理解により、プレイヤーのエンゲージメントを維持するために必要な変更を継続的に適用し、変更を加えることができます。

プレイヤーの使用パターンはそれぞれ大きく異なる可能性があるため、マーケットプレイスでゲームが成功するかどうかを予測することは簡単ではありません。このため、プレイヤーの数に応じて拡張できるソリューションを、コスト効率が高く管理が容易なままで、構築および維持することは困難です。[Amazon Pinpoint](#) やサードパーティ製オプションなどのターンキーユーザーエンゲージメントおよび分析ソリューションにより、ゲームからの分析のキャプチャを簡単に開始できます。しかしながら、多くのゲーム開発者やゲームメーカーは、アプリケーション全体のデータを共通の形式に一元化して、[データレイク](#)や分析アプリケーションと統合することを望んでいます。アマゾンウェブサービス (AWS) では、インフラストラクチャを管理することなく、ゲームからデータを取り込み、保存、分析するのに役立つ、包括的な[分析ソリューション](#)スイートを提供しています。

ゲーム分析パイプラインソリューションを使用すれば、ゲームやサービスから生成されたテレメトリデータを取り込み、保存、分析するためのスケーラブルな[サーバーレス](#)データパイプラインをゲームデベロッパーが起動できます。このソリューションではデータのストリーミング取り込みをサポートしているため、ユーザーは数分以内にゲームやその他のアプリケーションから洞察を得ることができます。ゲームテレメトリを取り込んで処理するための REST API と [Amazon Kinesis](#) サービスも提供しています。コスト効率の高いストレージと分析用に最適化した形式で、データを自動的に検証および変換し、[Amazon Simple Storage Service](#) (Amazon S3) に配信します。このソリューションを使用して、Amazon S3 でデータを整理および構造化し、かつデータセットのメタデータをカタログ化するように [AWS Glue](#) を設定することで、データレイクを統合します。このため、他のアプリケーションやユーザーとのデータの統合および共有が容易になります。

このソリューションでは、フルマネージド型のストリーミングデータ分析を提供しています。これにより、ゲームデベロッパーは、洞察と重要業績指標 (KPI) がすぐに必要なときに、ゲームイベントからカスタムメトリクスをリアルタイムで生成できます。ストリーミングデータ分析の結果は、[Live Ops](#) チームが使用して、リアルタイムのダッシュボードやライブゲームにアラートを表示したり、ストリーミング分析を使用して[イベント駆動型](#)のアプリケーションとワークフローを構築できるエンジニアリングチームが使用したりできます。

このソリューションは、ゲームイベントを分析とストレージ用データレイクに取り込むためのフレームワークを提供するように設計されており、基盤となるインフラストラクチャでの操作を管理する必要がなく、ソリューションの機能拡張に集中できます。

## コスト

このソリューションのデプロイの実行中に使用した AWS サービスのコストは、お客様の負担となります。公開の時点で、このソリューションを実行するコストは、ソリューションでロード、保存、処理、分析されるデータの量によって異なります。**開発者モード (Dev モード)** でソリューションをデプロイし、付属のデモスクリプトを使用してパイプラインにイベントを送信すると、**1 日あたり約 10.00 USD** の費用がかかります。

このソリューションをカスタマイズしてゲームデータセットを分析する場合は、コスト要因に取り込まれたストリーミングデータの量、分析されるデータのサイズ、ワークフローの各ステップに必要なコンピューティングリソース、およびワークフローの期間が含まれます。より正確なコスト見積もりについては、サンプルデータセットをベンチマークとして実行することをお勧めします。

料金は変更される可能性があります。詳細については、このソリューションで使用される各 AWS サービスの料金表ウェブページを参照してください。

## アーキテクチャの概要

このソリューションをデフォルトのパラメータでデプロイすると、AWS クラウドに次の環境が構築されます。

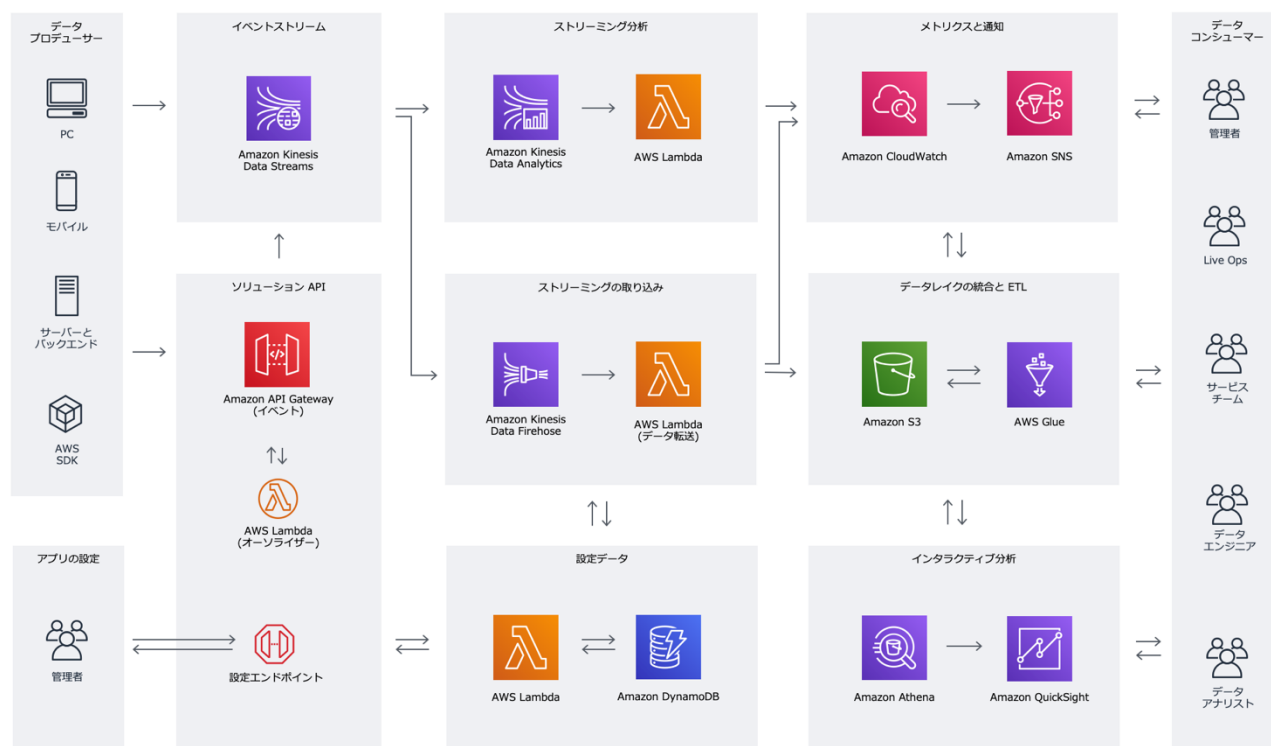


図 1: ゲーム分析パイプラインのアーキテクチャ

AWS CloudFormation テンプレートは、ゲームクライアント、ゲームサーバー、その他のアプリケーションからゲームデータを取り込むために AWS リソースをデプロイします。ストリーミングデータは、データレイクの統合とインタラクティブな分析のために Amazon S3 に取り込まれます。ストリーミング分析では、リアルタイムイベントを処理して、メトリクスを生成します。データコンシューマーは、Amazon CloudWatch のメトリクスデータと Amazon S3 内の未処理のイベントを分析します。特にこのテンプレートでは、このソリューションのために、次の基本的なビルディングブロックをデプロイします。

- ソリューション API と設定データ - [Amazon API Gateway](#)** はこのソリューションでゲームアプリケーションを登録し、ゲームのテレメトリデータを取り込んで [Amazon Kinesis Data Streams](#) にイベントを送信するための REST API エンドポイントを提供します。 [Amazon DynamoDB](#) はゲー

ムアプリケーションの設定と API キーを保存して、イベントをソリューション API に送信するときに使用します。

- **イベントストリーミング** - Amazon Kinesis Data Streams はゲームからストリーミングデータをキャプチャし、[Amazon Kinesis Data Firehose](#) と [Amazon Kinesis Data Analytics](#) によるリアルタイムのデータ処理を可能にします。
- **ストリーミング分析** - Amazon Kinesis Data Analytics は Amazon Kinesis Data Streams からのストリーミングイベントデータを分析し、[カスタムメトリクス](#)を生成します。このカスタムメトリクスの出力は、[AWS Lambda](#) を使って処理され、[Amazon CloudWatch](#) に発行されます。
- **メトリクスと通知** - Amazon CloudWatch は、AWS リソースの使用状況を監視してログに記録し、アラームを生成して、運用ダッシュボードを作成します。また、Amazon CloudWatch では、Amazon Kinesis Data Analytics が生成したカスタムメトリクス向けにメトリクスのストレージも提供します。[Amazon Simple Notification Service](#) (Amazon SNS) は、Amazon CloudWatch アラームに違反したときに、ソリューション管理者と他のデータコンシューマーに通知を配信します。
- **ストリーミングの取り込み** - Amazon Kinesis Data Firehose は Kinesis Data Streams からのデータを取り込み、データが Amazon S3 に配信される前に、サーバーレスデータ処理と変換向けのイベントのバッチで AWS Lambda を呼び出します。
- **データレイクの統合と ETL** - Amazon S3 では、未処理および処理済みのデータにストレージを提供しています。AWS Glue では、[AWS Glue データカタログ](#)に ETL 処理ワークフローとメタデータストレージを提供しています。これにより、柔軟な分析ツールと統合するためのデータレイクの基盤を提供します。
- **インタラクティブな分析** - ゲームイベントの分析を行うために、[Amazon Athena](#) のサンプルクエリがデプロイされます。[Amazon QuickSight](#) と統合する手順については、[自動デプロイ](#)セクションを参照してください。

デモスクリプトは、ゲーム内のさまざまな一般的なタイプのプレイヤーアクションを表すサンプルゲームイベントを生成します。これらのイベントはバッチ処理され、ソリューション機能をテストするために Amazon Kinesis Data Streams に自動的に送信されます。全体として、このソリューションではゲーム分析データの取り込み、分析、監視、およびレポートを可能にし、サーバーレスデータパイプラインをサポートするためのインフラストラクチャをセットアップします。

# ソリューションコンポーネント

## Amazon Kinesis

「ゲーム分析パイプライン」ソリューションでは、Amazon Kinesis を使用してゲームからのテレメトリデータを取り込み、処理します。Amazon Kinesis Data Streams は受信データを取り込み、Amazon Kinesis Data Firehose はストリーミングデータを取り込んで Amazon Simple Storage Service (Amazon S3) に配信します。[Amazon Kinesis Data Analytics for SQL Applications](#) は、ストリーミング SQL クエリを使用してリアルタイムのメトリクスを生成します。Amazon Kinesis Data Streams では、保管時の暗号化に [AWS Key Management Service](#) (AWS KMS) を使用します。

## Amazon API Gateway

このソリューションは、パイプラインにテレメトリデータを送信するためのイベントエンドポイントを提供する REST API を起動します。これにより、Amazon Kinesis と直接統合できないアプリケーションの統合オプションが提供されます。また、この API には、管理者がゲームアプリケーションをこのソリューションに登録し、開発者が REST API にイベントを送信するときに使用する API キーを生成するために使用する設定エンドポイントも用意されています。

## AWS Lambda 関数

このソリューションでは、AWS Lambda 関数を使用して、データ処理とアプリケーションバックエンドのビジネスロジックを提供します。次の AWS Lambda 関数をデプロイします。

- **ApplicationAdminServiceFunction 関数** - このソリューションの REST API 設定エンドポイントのバックエンドビジネスロジックを提供する関数。この関数は、新しく作成されたゲームアプリケーション設定を Amazon DynamoDB に保存し、開発者が REST API イベントエンドポイントと統合するための API キー認証を生成します。
- **LambdaAuthorizer 関数** - このソリューションの REST API イベントエンドポイントに送信されるイベントの認証をリクエストする関数。この関数は、ゲームの既存認証システム、および [Amazon Cognito](#) および外部の ID プロバイダーと統合するようにカスタマイズできます。

- **EventsProcessingFunction 関数** - イベントが Amazon S3 にロードされる前に Amazon Kinesis Data Firehose からの入力ゲームイベントを検証、変換、処理する関数。この関数は、入力イベントレコードのバッチで呼び出され、変換されたレコードを Amazon Kinesis Data Firehose に戻す前に検証と処理を実行してから、Amazon S3 のストレージに送信します。この関数は、追加のデータ処理を加えるために必要に応じて修正することが可能です。
- **AnalyticsProcessingFunction 関数** - Amazon Kinesis Data Analytics アプリケーションからのカスタムメトリクス出力を処理する関数。この関数は、Amazon Kinesis Data Analytics アプリケーションからの SQL 結果のバッチを処理し、カスタムメトリクスとして Amazon CloudWatch に発行します。この関数は、さまざまな出力形式を受け入れるようにカスタマイズすることも、追加の処理を行ったり、他の送信先への配信をしたりすることもできます。
- **GluePartitionCreator 関数** - 定期的なスケジュールで AWS Glue データカタログの `raw_events` AWS Glue テーブルに新しい日次パーティションを自動的に生成するユーティリティ関数。
- **SolutionHelper 関数** - AWS CloudFormation スタックのオペレーション中に使用される [AWS Lambda-backed カスタムリソース](#)。スタックのプロビジョニングとデプロイに使用します。

このソリューションによって作成された AWS Lambda 関数は、[AWS X-Ray](#) で有効化され、パフォーマンスのチューニングとトラブルシューティングのためにリクエストのトレースを行います。

## Amazon Simple Storage Service

このソリューションでは、Amazon Simple Storage Service (Amazon S3) を使用して、未処理のデータセットと処理済みデータセットにスケーラブルでコスト効率の高いストレージを提供します。Amazon S3 バケットは、[Amazon S3 Intelligent-Tiering](#) などの[オブジェクトライフサイクル管理](#)ポリシーで設定されており、データレイクなどのアクセスパターンが不明または変化するデータセットのコスト削減を実現します。

Amazon S3 の設定とソリューション内での使用方法の詳細については、[ゲーム分析パイプライン開発者ガイド](#)を参照してください。

## AWS Glue

このソリューションでは、次のメタデータストレージと ETL 処理用の AWS Glue リソースをデプロイします。

- AWS Glue データカタログの設定
- 取り込まれたデータのメタデータストアとして機能する AWS Glue データベースの作成
- ゲームイベントを処理するための AWS Glue ETL ジョブのデプロイ
- 処理結果で AWS Glue データカタログを更新するための AWS Glue クローラのデプロイ

このソリューションでは、AWS KMS を使用して AWS Glue データカタログを暗号化します。

## Amazon DynamoDB

このソリューションでは、Amazon DynamoDB テーブルを使用して、アプリケーション設定データを保存します。各 Amazon DynamoDB テーブルでは、[Amazon DynamoDB オンデマンドキャパシティ](#)を使用してプロビジョニングし、柔軟なスケーリングと予測不可能なアクセスパターンに対するコスト削減を実現します。

このソリューションでは、次の Amazon DynamoDB テーブルが作成されます。

- **Applications テーブル** - ApplicationId によって識別されるデータプロデューサーによって登録されたアプリケーションに関するデータを含むテーブル。取り込まれたイベントは、ストレージにロードされる前にこのテーブルに対してチェックされ、有効なアプリケーションに関連付けられていることが確認されます。
- **Authorizations テーブル** - ApiKeyId によって識別される API キー、および ApplicationId へのマッピングを格納するテーブル。各 API キーには、関連付けられた ApplicationId にイベントを発行する権限があります。このテーブルは、認証を管理するために ApplicationAdminServiceFunction AWS Lambda 関数によってアクセスされ、イベントエンドポイントで API 認証ルックアップを実行するために LambdaAuthorizer AWS Lambda 関数によってクエリされます。

このソリューションでは、AWS KMS を使用して Amazon DynamoDB テーブルの保管時の暗号化を行います。

## Amazon CloudWatch

このソリューションでは、Amazon CloudWatch を使用してソリューションのリソースを監視してログに記録し、Amazon Kinesis Data Analytics からリアルタイムで生成されたメトリクスのストレージを提供しています。このソリューションでは、Amazon CloudWatch アラームをデプロイして AWS リソースの使用状況を追跡し、問題が検知された場合に登録済み管理者にアラートを出します。Amazon CloudWatch にメトリクスを送信することにより、このソリューションでは、リアルタイムと AWS リソースの両方のメトリクスで単一の格納先に依存することが可能になります。

このソリューションでの Amazon CloudWatch の設定方法については、[ゲーム分析パイプライン開発者ガイド](#)を参照してください。

## Amazon Simple Notification Service

Amazon Simple Notification Service (Amazon SNS) では、リソースがしきい値を超えたときやエラーが検出されたときに、Amazon CloudWatch アラームによって生成される通知を配信します。このソリューションの管理者は、監視とトラブルシューティングの目的で通知を設定できます。

このソリューションは、AWS KMS を使用して Amazon SNS トピックの保管時の暗号化を行います。

## Amazon Athena

Amazon Athena は、Amazon S3 に保存されたゲームイベントデータにクエリとレポートを実行するために使用します。このソリューションには、事前に構築され、保存されたクエリー式も含まれており、ゲームイベントデータを詳しく調べることができます。このソリューションでは、追加のクエリを作成して保存できる [Athena ワークグループ](#)をデプロイします。

# 実装に関する考慮事項

## 適切なストリーミングデータの取り込み方法の選択

「ゲーム分析パイプライン」ソリューションには、ゲームイベントテレメトリデータをソリューションに取り込むための次のオプションが用意されています。

- **Amazon Kinesis Data Streams との直接統合** - Amazon Kinesis と統合するために AWS でサポートされている方法のいずれかを使用して Amazon Kinesis Data Streams と直接統合する場合は、このオプションを選択します。
- **ソリューション API イベントエンドポイントを使ったプロキシ統合** - ゲームイベントの取り込みにカスタム REST プロキシ統合が必要な場合は、こちらを選択します。アプリケーションは、Amazon Kinesis Data Streams にリクエストを同期的にプロキシしてクライアントに応答を返すこのイベントエンドポイントに、イベントを送信します。アプリケーションからイベントを取り込むようにソリューション API をセットアップする方法については、[ゲーム分析パイプライン開発者ガイド](#)を参照してください。

このソリューションは、Amazon Kinesis Data Streams に直接データを送信するか、Amazon Kinesis Data Streams にイベントを転送するソリューション API に HTTPS API リクエストを送信することで、ゲームイベントデータを取り込みます。REST API は、カスタム REST API プロキシ統合を必要とするアプリケーションのエントリーポイントです。Amazon Kinesis Data Streams では、ストリームにデータを直接発行するためのデータプロデューサー向けオプションが複数用意されており、大半のユースケースに適しています。詳細については、*Amazon Kinesis Data Streams* 開発者ガイドの [Amazon Kinesis Data Streams へのデータの書き込み](#)を参照してください。

## ゲームクライアントとの統合

モバイルゲーム開発者、または既存バックエンドのないゲームを開発している場合は、ソリューション API を必要とせずに、ゲームやサービスから直接 Amazon Kinesis Data Streams にイベントを発行できます。クライアントを Amazon Kinesis Data Streams に直接統合するには、[Amazon Cognito ID プール \(フェデレーティッドアイデンティティ\)](#) を設定して、クライアントが Amazon Kinesis Data

Streams などの AWS サービスと安全にやり取りすることを許可する一時的な AWS 認証情報を生成します。

Amazon Kinesis Data Streams と直接統合する場合は、[ゲーム分析パイプライン開発者ガイド](#)を参照して、Amazon Kinesis Data Streams にデータレコードを送信するためにこのソリューションに必要な形式を確認してください。

または、ソリューション API のイベントエンドポイントと統合して、カスタム REST インターフェースを使用してクライアントからバックエンド実装を抽象化するか、取り込みデータをさらにカスタマイズする必要があります。

**注意:** Amazon API Gateway を使用してデータを取り込むと、追加料金が発生します。ソリューション API を使用してデータを取り込む場合は、[Amazon API Gateway REST API](#) の料金情報を参照して、使用要件に基づいてコストを決定してください。

## ゲームバックエンドとの統合

ゲームサーバーや他のアプリケーションバックエンドなど、ゲームのバックエンドを操作する場合は、[Kinesis Agent](#)、[Amazon Kinesis Producer Library \(KPL\)](#)、AWS SDK、または他のサポートされている Kinesis の統合を使ってバックエンドから Amazon Kinesis Data Streams に直接データを送信します。このアプローチにより、ゲームクライアントや他のアプリケーションでは、既存のクライアント / サーバー接続と認証を再利用して、テレメトリイベントをバックエンドに送信できます。このバックエンドは、イベントを取り込み Amazon Kinesis Data Streams に送信するように設定できます。また、このアプローチでは、クライアント統合の変更を最小限に抑えたい場合や、高スループットのユースケースを実装したい場合などにも使用できます。

バックエンド内の複数のクライアントからイベントを収集して集約することで、全体的なバッチ処理と取り込みスループットを向上させ、Amazon Kinesis Data Streams にデータを送信する前に追加のコンテキストでデータを強化できます。これにより、コスト削減、セキュリティ強化、既存バックエンドを使ったゲームのクライアント統合をシンプルにできます。既存 Amazon Kinesis Data Streams オプションの多くは、自動再試行、エラー処理、追加の組み込み関数を提供しています。KPL と AWS SDK はカスタムデータプロデューサーの開発に一般的に使用されており、Kinesis エージェントをゲームサ

ーバーにデプロイして、ログファイルでテレメトリイベントを処理し、それらを Amazon Kinesis Data Streams に送信します。

## AWS SDK との統合

AWS SDK を使用して Amazon Kinesis Data Streams をアプリケーションに直接統合できます。AWS SDK では、[.NET](#) や [C++](#) などの一般的に使用されるさまざまな言語をサポートし、ゲームイベントのバッチを直接 Amazon Kinesis Data Streams に送信する同期および非同期統合向けの [.NET の PutRecords](#) や [PutRecordsAsync](#) などのメソッドを提供します。Amazon Kinesis Data Streams にイベントを直接送信する場合は、デプロイ前にこのソリューションがカスタマイズされていない限り、各データレコードが[ゲーム分析パイプライン開発者ガイド](#)で定義されているゲームイベントスキーマに準拠していることを想定しています。イベントはスキーマに対して検証し、このソリューションのデータ品質を強化して、Amazon Kinesis Data Firehose が Amazon S3 にロードを試みる前に不正な形式のデータを処理します。ゲーム開発者は、EventsProcessingFunction AWS Lambda 関数に付属のゲームイベントスキーマを開発中のリファレンスとして使用できます。詳細については、*Amazon Game Tech Blog* の [AWS SDK の使用開始のゲーム開発者ガイド](#)を参照してください。

## Amazon QuickSight との統合

Amazon QuickSight を使用して、ゲームのデータを視覚化します。Amazon QuickSight を Amazon Athena に接続して、ゲームデータにアクセスします。詳細なデータ探索を行うために、ダッシュボードを作成、設定、カスタマイズできます。[Amazon Athena を Amazon QuickSight に接続して、Amazon QuickSight ダッシュボードを構築する手順](#)については、[自動デプロイセクション](#)を参照してください。

## デプロイ可能な AWS リージョン

このソリューションでは、Amazon Kinesis Data Analytics、Amazon Kinesis Data Firehose、AWS Glue、Amazon Athena、および Amazon QuickSight を使用しています。これらは現在、特定の AWS リージョンでのみ利用できます。そのため、このソリューションはこれらのサービスが利用可能な AWS リージョンで起動する必要があります。リージョン別のサービスの利用状況に関する最新情報については、[AWS のリージョン表](#)を参照してください。

# AWS CloudFormation テンプレート

このソリューションでは、AWS CloudFormation を使用して、AWS クラウドへの「ゲーム分析パイプライン」ソリューションのデプロイを自動化します。このソリューションには次の AWS CloudFormation テンプレートが含まれており、デプロイ前にダウンロード可能です。

## テンプレートを表示

**game-analytics-pipeline.template:** このテンプレートを使用して、「ゲーム分析パイプライン」ソリューションおよびすべての関連コンポーネントを起動します。デフォルト設定では、Amazon API Gateway、AWS Lambda、Amazon Kinesis、Amazon Simple Notification Service (Amazon SNS)、Amazon DynamoDB、AWS Glue、Amazon Athena、Amazon CloudWatch、AWS Key Management Service (AWS KMS)、および Amazon Simple Storage Service (Amazon S3) がデプロイされますが、特定のニーズに基づいてテンプレートをカスタマイズすることもできます。

## 自動デプロイ

自動デプロイを開始する前に、このガイドで説明されているアーキテクチャ、設定、およびその他の考慮事項をよくお読みください。このセクションの手順に従って、このソリューションを設定して AWS アカウントにデプロイします。

**デプロイ時間:** 約 5 分

## 前提条件

このソリューションに含まれている Python スクリプトを実行するには、最新バージョンの [AWS コマンドラインインターフェイス](#) (AWS CLI) がインストールされている必要があります。AWS CLI がインストールされていない場合は、AWS コマンドラインインターフェイスユーザーガイドの [AWS CLI のインストール](#) を参照してください。または、AWS Cloud9 環境を使用することで、スクリプトのデプロイを簡単にできます。詳細については、AWS Cloud9 ユーザーガイドの [EC2 Environment を作成する](#) を参照してください。

## デプロイの概要

このアーキテクチャを AWS にデプロイする手順は、次のステップで構成されます。詳細な手順については、各ステップのリンクを参照してください。

### [ステップ 1. スタックを起動する](#)

- 自身の AWS アカウントで AWS CloudFormation テンプレートを起動します。
- テンプレートのパラメータを確認し、必要に応じて調整します。

### [ステップ 2. サンプルゲームイベントを生成する](#)

- サンプルイベントを発行します。

### [ステップ 3. Amazon Athena でサンプルクエリをテストする](#)

- サンプルイベントをクエリして、生成されたデータに関する正しい情報を取得します。

### [ステップ 4. Amazon Athena を Amazon QuickSight に接続する](#)

- Amazon Athena を Amazon QuickSight に接続します。
- 必要な計算フィールドを作成します。

### [ステップ 5. Amazon QuickSight ダッシュボードを作成する](#)

- サンプルイベントを視覚化して、生成されたデータに関する正しい情報を取得します。

### [ステップ 6. パイプラインの運用ダッシュボードにリアルタイムのメトリクスを表示して追加する](#)

- Amazon CloudWatch コンソールにアクセスして、Amazon Kinesis Data Analytics によって生成されたリアルタイムのメトリクスを表示して、運用ダッシュボードに追加します。

## ステップ 1. スタックを起動する

この自動化 AWS CloudFormation テンプレートは、「ゲーム分析パイプライン」ソリューションを AWS クラウドにデプロイします。

**注意:** このソリューションの実行中に使用した AWS サービスのコストは、お客様の負担となります。詳細については、[コスト](#)セクションを参照してください。詳細については、このソリューションで使用される各 AWS サービスの料金表ウェブページを参照してください。

1. AWS マネジメントコンソールにサインインし、右側のボタンを使用して `game-analytics-pipeline.template` AWS CloudFormation テンプレートを起動します。

ソリューション  
の起動

あるいは、独自にカスタマイズするために[テンプレートをダウンロード](#)することもできます。

2. テンプレートは、デフォルトで米国東部 (バージニア北部) リージョンで起動されます。別の AWS リージョンでこのソリューションを起動するには、コンソールのナビゲーションバーのリージョンセクターを使用します。

**注意:** このソリューションでは、Amazon Kinesis Data Analytics、Amazon Kinesis Data Firehose、AWS Glue、Amazon Athena、および Amazon QuickSight を使用しています。これらは現在、特定の AWS リージョンでのみ利用できます。そのため、このソリューションはこれらのサービスが利用可能な AWS リージョンで起動する必要があります。AWS リージョン別の利用状況に関する最新情報については、[AWS のリージョン表](#)をご参照ください。

3. **スタックの作成** ページで、正しいテンプレート URL が **Amazon S3 URL** テキストボックスに示されていることを確認し、**[次へ]** を選択します。
4. **スタックの詳細を指定** ページで、このソリューションのスタックに名前を割り当てます。
5. **パラメータ** で、テンプレートのパラメータを確認し、必要に応じて変更します。このソリューションでは、次のデフォルト値を使用します。

パラメータ	デフォルト	説明
<code>EnableStreamingAnalytics</code>	Yes	Amazon Kinesis Data Analytics for SQL をこのソリューションにデプロイするかどうかを決定する切り替えスイッチ。
<code>KinesisStreamShards</code>	1	Amazon Kinesis Data Streams 用にプロビジョニングするシャードの数を識別する数値。

パラメータ	デフォルト	説明
		<p><b>注意:</b> スループットに必要なシャードの決定については、<a href="#">Amazon Kinesis Data Streams 開発者ガイドの Amazon Kinesis Data Streams の用語と概念</a>を参照してください。</p>
<b>SolutionAdminEmailAddress</b>	false	このソリューションのリソースによって生成されて、Amazon CloudWatch によって配信される運用通知を受信する E メールアドレス。デフォルトの false パラメータは、Amazon SNS トピックへのサブスクリプションを無効にします。
<b>SolutionMode</b>	Dev	このソリューションのデプロイモード。サポートされている値は Dev と Prod です。
		<p><b>注意:</b> Dev モードは Amazon Kinesis Data Firehose のバッファ間隔を 1 分ごとに短縮し、テスト中の Amazon S3 へのデータ配信を高速化しますが、これによりバッチ処理の最適化が低下します。また、Dev モードはサンプル用の Athena クエリをデプロイし、テストの目的でサンプルアプリケーションと API キーを作成します。Prod モードは、15 分のバッファ間隔で Amazon Kinesis Data Firehose を設定し、サンプル用の Athena クエリまたはサンプルアプリケーションおよび API キーをデプロイしません。</p>

- [次へ] を選択します。
- スタックオプションの設定** ページで、[次へ] を選択します。
- レビュー** ページで、設定を確認して確定します。テンプレートが [AWS Identity and Access Management](#) (IAM) リソースを作成することを確認する 3 つのチェックボックスを必ずオンにします。
- [**スタックの作成**] を選択してスタックをデプロイします。

スタックのステータスは、AWS CloudFormation コンソールの**ステータス**列で表示できます。約 5 分で `CREATE_COMPLETE` のステータスが表示されます。

10. スタックがデプロイされたら、**出力**タブに移動します。**GameEventsStream** と **TestApplicationId** の値を記録します。これらの値は、次のステップで必要になります。

## ステップ 2. サンプルゲームイベントを生成する

Python のデモスクリプトを使用して、テストとデモンストレーションの目的でサンプルゲームイベントデータを生成します。

**注意:** Python スクリプトを実行するには、最新バージョンの [AWS コマンドラインインターフェイス](#) (AWS CLI) がインストールされている必要があります。AWS CLI がインストールされていない場合は、AWS コマンドラインインターフェイスユーザーガイドの [AWS CLI のインストール](#)を参照してください。または、AWS Cloud9 環境を使用することで、スクリプトのデプロイをシンプルにできます。詳細については、AWS Cloud9 ユーザーガイドの [EC2 Environment を作成する](#)を参照してください。

1. [GitHub リポジトリ](#)にアクセスし、`./source/demo/publish_data.py` から Python のデモスクリプトをダウンロードします。
2. ターミナルウィンドウで、次の Python コマンドを実行して、デモスクリプトの前提条件をインストールします。

```
python3 -m pip install --user --upgrade pip
python3 -m pip install --user virtualenv
python3 -m venv env
source env/bin/activate
pip install boto3 numpy uuid argparse
```

これらのコマンドで次のエラーが発生した場合: `python3: can't open file 'publish_data.py': [Errno 2] No such file or directory`、`publish_data.py` がルートディレクトリにあることを確認してください。

**注意:** このソリューションでは、[Boto 3](#) (AWS SDK for Python) を使用して Amazon Kinesis とやり取りします。このソリューションでは、numpy、uuid、および argparse を使用して引数を受け入れ、ランダムなサンプルゲームイベントデータを生成します。

3. デモスクリプトをインストールするには、`./source/demo/` フォルダに移動し、次の Python コマンドを実行してください。

```
python3 publish_data.py --region <aws-region> --stream-name  
<GameEventsStream> --application-id <TestApplicationId>
```

`<aws-region>` を AWS CloudFormation スタックをデプロイした [AWS リージョンコード](#) に置き換えます。`<GameEventsStream>` と `<TestApplicationId>` を、AWS CloudFormation スタックの出力タブから記録した値に置き換えます。これらのインプットは、指定されたアプリケーションに対して 100 個のランダムなゲームイベントのバッチを継続的に生成するようにスクリプトを設定し、PutRecords API を使用してイベントを Amazon Kinesis に発行します。

## ステップ 3. Amazon Athena でサンプルクエリをテストする

このソリューションでは、Athena ワークグループに保存されているサンプルの Athena クエリを提供しています。Amazon Athena でサンプルクエリを実行するには、この手順を使用してください。

1. [Amazon Athena コンソール](#) に移動します。
2. Athena ホームページから、[今すぐ始める] を選択します。
3. ページの上部にある [ワークグループ] タブを選択します。
4. `GameAnalyticsWorkgroup-<your-cloudformation-stackname>` という名前のワークグループを選択し、[ワークグループを切り替える] を選択します。
5. ワークグループの切り替えダイアログボックスで、[切り替え] を選択します。
6. [保存したクエリ] タブを選択します。
7. 既存クエリのいずれかを選択し、[クエリの実行] を選択して SQL を実行します。

- クエリをカスタマイズします。

**注意:** クエリのカスタマイズについては、Amazon Athena ユーザーガイドの [Amazon Athena を使用した SQL クエリの実行](#) を参照してください。

## ステップ 4. Amazon Athena を Amazon QuickSight に接続する

Amazon QuickSight 内でデータソースとして Amazon Athena を設定するには、この手順を使用します。

- [Amazon QuickSight コンソール](#) に移動し、ページの右上隅から **[Admin]** を選択して、**[QuickSight の管理]** を選択します。
- アカウントページで、**[セキュリティとアクセス権限]** を選択します。
- QuickSight の AWS のサービスへのアクセス** で、**[追加または削除する]** を選択します。
- [Amazon Athena]** を選択し、**[次へ]** を選択します。
- Amazon Athena** ダイアログボックスで、**[次へ]** を選択します。

**注意:** Amazon QuickSight の設定を以前に設定している場合、ダイアログボックスを表示するには、**Amazon Athena** のチェックボックスを選択解除して再度選択する必要があります。

- Amazon S3 バケットの選択** ダイアログボックスで、**QuickSight アカウントにリンクされている S3 バケット** タブが表示されていることを確認し、次のステップを実行します。
  - [AnalyticsBucket]** リソース (AWS CloudFormation で以前に作成した) を選択します。
  - 書き込みアクセス許可** 列で、**Athena Workgroup** の横にあるチェックボックスをオンにします。

**注意:** **AnalyticsBucket** リソースを識別するには、AWS CloudFormation スタックの**出力**タブを参照してください。

7. [完了]、[更新] の順に選択します。
  8. 左上隅の QuickSight ログを選択して、[Amazon QuickSight コンソール](#)に移動します。
  9. 左のナビゲーションペインで、[データセット] を選択します。
  10. [新しいデータセット] を選択します。
  11. [Athena] を選択します。
  12. 新規 Athena データソースダイアログボックスのデータソース名フィールドに、名前 (例: game-analytics-pipeline-connection) を入力します。
  13. Athena ワークグループフィールドで、GameAnalyticsWorkgroup-*<your-cloudformation-stackname>* という名前のワークグループを選択し、[接続を検証] を選択します。接続が検証されたら、[データソースを作成] を選択します。
  14. テーブルの選択ダイアログボックスのデータベースフィールドで、AWS CloudFormation によってデプロイされたデータベースを選択します。使用可能なテーブルのリストが表示されます。
- 注意:** データベースの値は、AWS CloudFormation スタックの出力タブでキー名が **GameEventsDatabase** で確認できます。
15. [raw\_events] を選択し、[選択] を選択します。
  16. データセットの作成を終了するダイアログボックスで、[クエリを直接実行] を選択し、[データの編集/プレビュー] を選択します。
  17. [計算フィールドを追加] を選択して、計算フィールドを作成します。
  18. 計算フィールドを追加ページで、名前を追加テキストボックスのプレースホルダーで、level\_id と入力します。
  19. 式フィールドに、`parseJson({event_data}, "$.level_id")` と入力し、[保存] を選択します。式フィールドとは、ページの中央にある大きな空白です。

**注意:** ステップ 17 ~ 19 を繰り返して、event\_data から抽出するデータ型ごとに計算フィールドを作成します。非構造化 event\_data の詳細については、[ゲーム分析パイプライン開発者ガイド](#)を参照してください。

20. **[計算フィールドを追加]** を選択して、計算フィールドを作成します。
21. **計算フィールドを追加** ページで、**タイトルを追加** テキストボックスのプレースホルダーで、`event_timestamp_time_format` と入力します。
22. **式** フィールドに `epochDate({event_timestamp})` と入力し、**[保存]** を選択します。式フィールドとは、ページの中央にある大きな空白です。
23. **Data Prep** ページで、**[保存]** を選択します。

## ステップ 5. Amazon QuickSight ダッシュボードを作成する

視覚化からダッシュボードを構築するには、この手順を使用します。ダッシュボードには、時間の経過に伴う一意のプレイヤーセッションを示す折れ線グラフと、レベル完了率（ゲームのステージやクエストなど「面」のクリア率のこと）とチュートリアルを進捗を示す複数の棒グラフが含まれます。

### event\_type でフィルタリングされたイベントの数

折れ線グラフを作成するには、この手順を使用します。

1. [Amazon QuickSight コンソール](#) に移動します。
2. **Amazon QuickSight** ページで、左のナビゲーションペインから **[分析]** を選択し、右上のメニューバーから **[新しい分析]** を選択します。
3. **ユーザーのデータセット** ページで、**[raw\_events]** を選択します。
4. **raw\_events** ダイアログボックスで、**[分析の作成]** を選択します。空白のビジュアルを持つ新しいシートが表示されます。

**注意:** 空白のビジュアルが表示されない場合は、メニューから **[+ 追加]** を選択し、ドロップダウンリストから **[ビジュアルを追加]** を選択します。

5. **ビジュアルタイプ** ペインで、折れ線グラフのアイコンを選択します。Amazon QuickSight でビジュアルが作成されます。

6. フィールドリストペインで、`[event_timestamp_time_format]` を選択します。

**注意:** フィールドリストペインが表示されない場合は、左のメニューオプションから **[視覚化]** を選択します。

7. フィールドウェルページで、フィールドが視覚化されていることを確認します。

8. フィールドリストペインで、`event_id` をフィールドウェルの **[値]** にドラッグします。

9. フィールドリストペインで、`event_type` をフィールドウェルの **[色]** にドラッグします。

10. フィールドウェルページで、`event_timestamp_time_format` の横にあるドロップダウン矢印を選択し、**[集計: 日]** を強調表示して、最後に **[分]** を選択します。

11. デフォルトのビジュアルラベルテキストを **イベント数 (event\_type でフィルタ)** に置き換えて、視覚化のタイトルを変更します。

12. ビジュアルで歯車アイコンを選択し、**ビジュアルのフォーマットペイン**で、次のアクションを実行します。

- **[X 軸]** を選択し、X 軸のラベルを入力します。(例: Event Timestamp)
- **[Y 軸]** を選択し、Y 軸のラベルを入力します。(例: Event Count)
- **[凡例]** を選択し、凡例のラベルを入力します。(例: Event Type)

図 2 は、折れ線グラフの視覚化の例です。

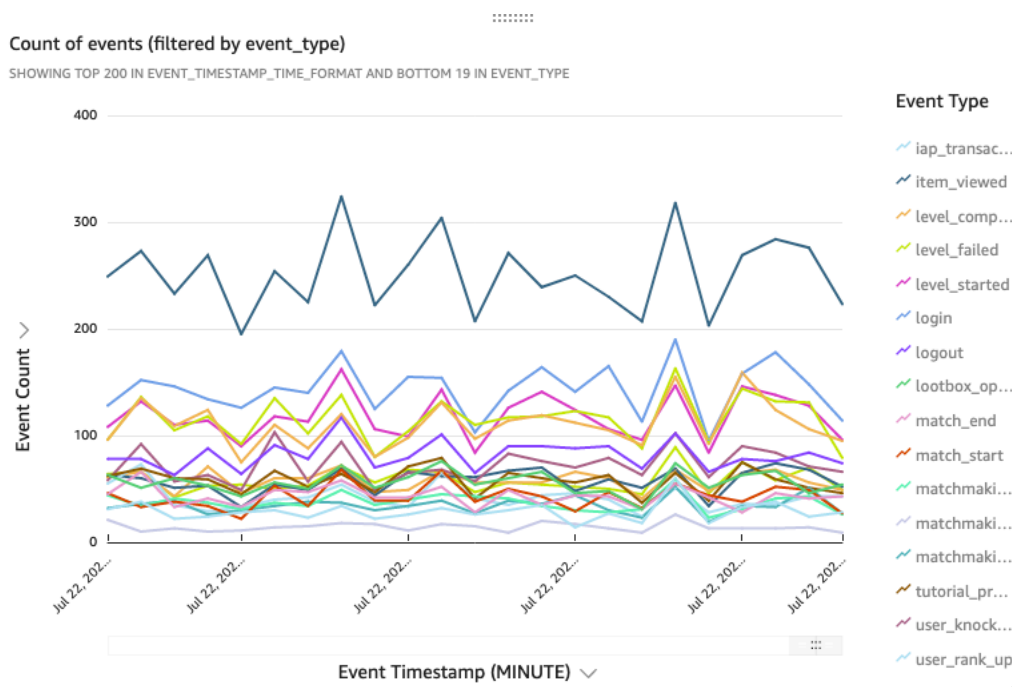


図 2: event\_type の視覚化でフィルタリングされたイベントの数の例

## アプリケーションバージョン別のチュートリアルを進捗

水平棒グラフを作成するには、この手順を実行します。

1. 同じシートで、アプリケーションバーから **[+ 追加]** を選択し、ドロップダウンメニューで **[計算フィールドを追加]** を選択します。
2. **計算フィールドを追加** ページで、**タイトルを追加** テキストボックスのプレースホルダーで、`tutorial_screen_id` と入力します。
3. **式** フィールドに、`parseJson({event_data}, "$.tutorial_screen_id")` と入力し、**[保存]** を選択します。
4. アプリケーションバーから **[+ 追加]** を選択し、ドロップダウンリストで **[ビジュアルを追加]** を選択します。
5. **フィールドリスト** ペインで、`tutorial_screen_id` を選択します。`tutorial_screen_id` のドロップダウンメニューを展開し、**a ~ z** で並べ替えます。

**注意:** フィールドリストペインが利用できない場合は、[視覚化] を選択してください。

6. フィールドリストペインで、[app\_version] をフィールドウェルの [グループ/色] にドラッグします。
7. ビジュアルタイプペインで、垂直棒グラフのアイコンを選択します。Amazon QuickSight でビジュアルが作成されます。
8. ビジュアルに null 値を表示するバーがある場合は、そのバーを選択し、[空白を除く] を選択します。これにより、視覚化から tutorial\_screen\_id フィールドのないすべてのゲームイベントを除外するフィルタが作成されます。
9. デフォルトのビジュアルラベルテキストを [Tutorial progression by application version] に置き換えて、視覚化のタイトルを変更します。ビジュアルで、歯車アイコンを選択します。ビジュアルのフォーマットペインで [X 軸] を選択し、X 軸のラベルを入力します。(例: Tutorial Screen) [Y 軸] を選択し、Y 軸のラベルを入力します。(例: Count) [凡例] を選択し、凡例のラベルを入力します。(例: App Version)

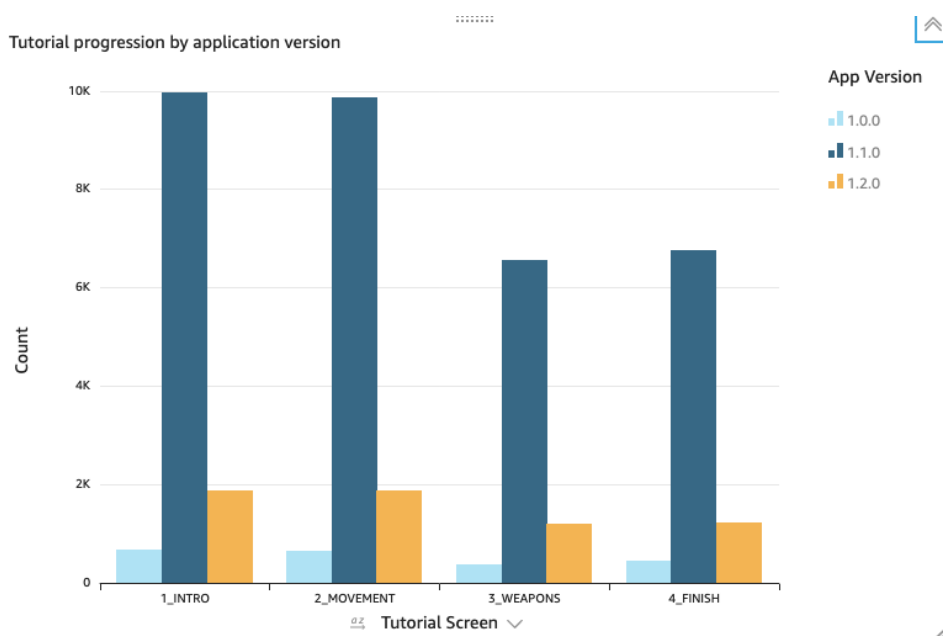


図 3: アプリケーションバージョン別の視覚化によるチュートリアルの進捗の例

## アプリケーションバージョン別のレベル完了率

カスタム SQL を使用して垂直棒グラフを作成するには、この手順を使用します。

1. [Amazon QuickSight コンソール](#)に移動します。
2. 左側のナビゲーションペインから [データセット] を選択します。
3. [新しいデータセット] を選択します。
4. [Athena] を選択します。新規 Athena データソースダイアログボックスのデータソース名フィールドに、名前を入力します。(例: `level_completion_rate by application version and day`)
5. Athena ワークグループフィールドで、`GameAnalyticsWorkgroup-<your-cloudformation-stackname>` という名前のワークグループを選択し、[接続を検証] を選択します。接続が検証されたら、[データソースを作成] を選択します。
6. テーブルの選択ダイアログボックスのデータベースフィールドで、AWS CloudFormation によってデプロイされたデータベースを選択します。使用可能なテーブルのリストが表示されます。

**注意:** データベースの値は、AWS CloudFormation スタックの出力タブでキー名が **GameEventsDatabase** で確認できます。

7. [raw\_events] を選択し、[カスタム SQL を使用] を選択します。
8. カスタム SQL クエリの入力ダイアログボックスで、SQL クエリの名前 (例: `Level completion rate by application version and day`) を入力し、次の SQL クエリをコピーして、[SQL クエリに入力します。] フィールドに入力します。

**注意:** `<GameEventsDatabase>` に置き換えるデータベースの値は、AWS CloudFormation スタックの出力タブでキー名が **GameEventsDatabase** で確認できます。

```
with t1 AS
  (SELECT json_extract_scalar(event_data,
    '$.level_id') AS level, count(json_extract_scalar(event_data,
    '$.level_id')) AS level_count, app_version, date_trunc('hour',
    from_unixtime(event_timestamp)) as event_hour_utc
  FROM "<GameEventsDatabase>". "raw_events"
  WHERE event_type='level_started'
  GROUP BY json_extract_scalar(event_data, '$.level_id'), app_version,
  date_trunc('hour', from_unixtime(event_timestamp))), t2 AS
  (SELECT json_extract_scalar(event_data,
    '$.level_id') AS level, count(json_extract_scalar(event_data,
    '$.level_id')) AS level_count, app_version, date_trunc('hour',
    from_unixtime(event_timestamp)) as event_hour_utc
  FROM "<GameEventsDatabase>". "raw_events"
  WHERE event_type='level_completed' GROUP BY json_extract_scalar(event_data,
    '$.level_id'), app_version, date_trunc('hour',
    from_unixtime(event_timestamp)))
SELECT t2.level,
  t2.app_version,
  (cast(t2.level_count AS DOUBLE) / (cast(t2.level_count AS DOUBLE) +
  cast(t1.level_count AS DOUBLE))) * 100 AS level_completion_rate,
  t2.event_hour_utc
FROM t1
JOIN t2
  ON t1.level = t2.level
ORDER BY level;
```

9. **[クエリの確認]** を選択します。
10. **データセットの作成を終了する** ダイアログボックスで、**[クエリを直接実行]** を選択します。
11. **[データの編集/プレビュー]** を選択します。
12. カスタム SQL は、結果のテーブルを生成します。メニューから **[保存]** を選択します。
13. [Amazon QuickSight コンソール](#) に移動します。
14. **Amazon QuickSight** ページで、左のナビゲーションペインから **[分析]** を選択し、**[raw\_events analysis]** を選択します。
15. 新しいデータセットを追加するには、データセットペインで、編集アイコンを選択します。
16. **[データセットを追加]** を選択します。
17. **データセットを選択して追加** ダイアログボックスで、**[Level completion rate by application version and day]** を選択し、**[選択]** を選択します。

18. 新しいデータセットを選択するには、データセットペインで、データセットのドロップダウンを展開し、[**Level completion rate by application version and day**] を選択します。
19. アプリケーションバーから [+ 追加] を選択し、ドロップダウンメニューで [**ビジュアルを追加**] を選択します。
20. フィールドリストペインで、視覚化する **level** を選択します。

**注意:** フィールドリストペインが表示されない場合は、[**視覚化**] を選択してください。

21. **ビジュアルタイプ**ペインで、垂直棒グラフのアイコンを選択します。デフォルトでは、**level** は X 軸として定義されます。Amazon QuickSight でビジュアルが作成されます。
22. フィールドウェルの [**値**] に Y 軸の **level\_completion\_rate** を追加し、**level\_completion\_rate (Sum)** の横にあるドロップダウンメニューで [**集計: 合計**] を選択して [**平均**] に変更します。**level\_completion\_rate** の同じドロップダウンメニューで、[**形式**]、[**その他のフォーマットオプション**] の順に選択します。**データをフォーマット**メニューで、[**単位**] メニューを展開し、**サフィックス**ダイアログボックスで % を追加します。メニューの右上のセクションにある [**x**] を選択して、**データをフォーマット**メニューを閉じます。
23. [**app\_version**] をフィールドウェルの [**グループ/色**] に追加します。
24. **歯車**アイコンを選択して、ビジュアルを編集します。**ビジュアルのフォーマット**メニューで、[**Y 軸**] メニューを展開し、軸のラベルを入力します。(例: Level Completion Rate) **カスタム範囲**で、**最小**ダイアログボックスに 0、**最大**ダイアログボックスに 100 と入力します。**軸ステップ**で、[**ステップのサイズ**] を選択し、ダイアログボックスに 20 と入力します。**X 軸**メニューを展開し、軸のラベルを入力します。(例: Level) **凡例**メニューを展開し、凡例のラベルを入力します。(例: App Version)
25. **フィールドウェル**ペインで、**Level** のドロップダウンメニューを展開し、**a ~ z** でソートを選択します。
26. デフォルトのビジュアルラベルテキストを [**Level completion rate by application version**] に置き換えて、視覚化のタイトルを変更します。

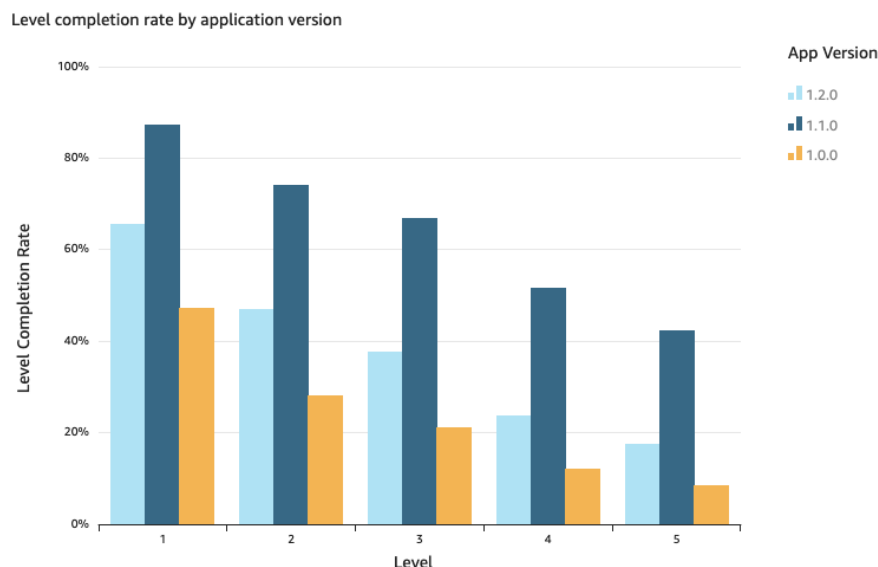


図 4: アプリケーションバージョン別の視覚化によるレベル完了率の例

## 時間の経過に伴うレベル完了率の分布

カスタム SQL を再利用して垂直棒グラフを作成するには、この手順を使用します。

1. 同じシートで、**データセット**が [Level completion rate by application version and day] に設定されていることを確認します。
2. アプリケーションメニューから [+ 追加] を選択し、ドロップダウンメニューで [ビジュアルを追加] を選択します。
3. フィールドリストペインで、[event\_hour\_utc] を選択して視覚化します。
4. **ビジュアルタイプ**で、垂直棒グラフのアイコンを選択します。デフォルトでは、event\_hour\_utc は X 軸として定義されます。Amazon QuickSight でビジュアルが作成されます。
5. フィールドウェルの [値] に Y 軸の level\_completion\_rate を追加し、level\_completion\_rate (Sum) の横にあるドロップダウンメニューで [集計: 合計] を選択して [平均] に変更します。level\_completion\_rate の同じドロップダウンメニューで、[形式]、[その他のフォーマットオプション] の順に選択します。データをフォーマットメニューで、[単位] メ

メニューを展開し、**サフィックス**ダイアログボックスで % を追加します。メニューの右上のセクションにある [x] を選択して、**データをフォーマット**メニューを閉じます。

6. フィールドウェルの [グループ/色] に [Level] を追加します。
7. **歯車**アイコンを選択してビジュアルを編集し、次のアクションを実行します。
  - **ビジュアルのフォーマット**メニューで、[Y 軸] メニューを展開し、軸のラベルを入力します。  
(例: Level Completion Rate (%))
  - **カスタム範囲**で、**最小**ダイアログボックスに 0、**最大**ダイアログボックスに 100 と入力します。
  - **軸ステップ**で、[**ステップのサイズ**] を選択し、ダイアログボックスに 20 と入力します。
  - **X 軸**メニューを展開し、軸のラベルを入力します。(例: Event Time)
  - **凡例**メニューを展開し、軸のラベルを入力します。(例: Level)

**注意:** この視覚化では、レベル完了率を最も詳細な時間の範囲で集計するカスタム SQL クエリが使用されます。そのため、このビジュアルでの「分」へのドリルダウンは、クエリを分単位でグループ化するように変更しないとサポートされません。これにより、視覚化のロード時間が遅くなる可能性があります。

8. デフォルトのビジュアルラベルテキストを [Level completion rate over time] に置き換えて、視覚化のタイトルを変更します。

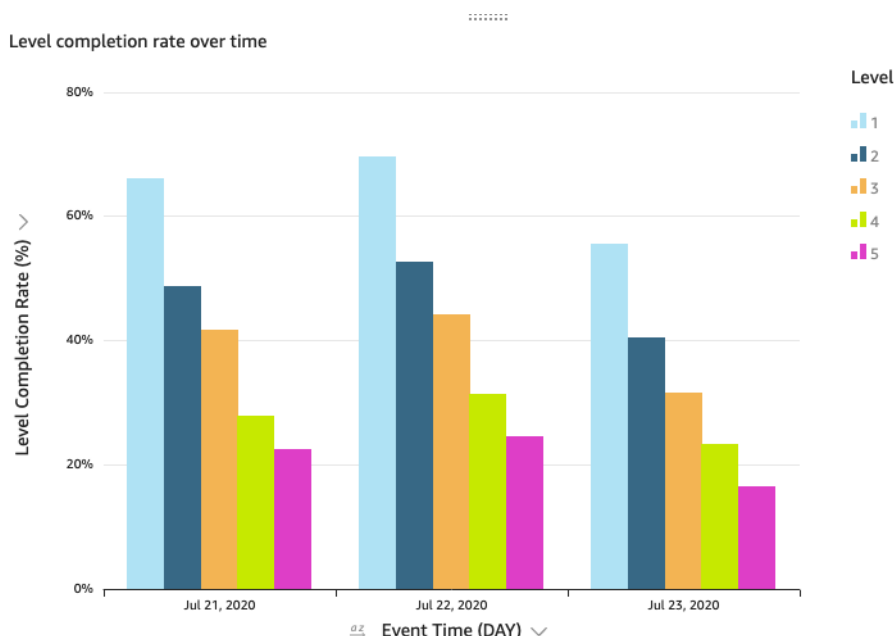


図 5: 時間の経過に伴うレベル完了率の視覚化の例

## イベントのトレンド KPI

パイプラインを通過するイベント数の傾向を示す KPI を作成するには、この手順を使用します。

1. 同じシートで、[視覚化] タブを選択し、データセットを [raw\_events] に変更します。
2. [+ 追加] を選択し、ドロップダウンリストで [ビジュアルを追加] を選択します。
3. ビジュアルタイプペインで、[Key Performance Indicator (KPI)] アイコンを選択します。
4. フィールドリストペインで、[event\_id] をフィールドウェルの [値] にドラッグします。

**注意:** フィールドリストペインが表示されていない場合は、[視覚化] を選択してください。

5. フィールドリストペインで、[event\_timestamp\_time\_format] をフィールドウェルの [トレンドグループ] にドラッグします。event\_timestamp\_time\_format の横にあるドロップダウン矢印を選択し、[集計: 日] が選択されていることを確認します。分析しようとしている傾向に基づいて集計単位を変更できます。
6. 歯車アイコンを選択してビジュアルを編集します。ビジュアルのフォーマットメニューの比較方法で、[パーセント (%) としての差異] を選択します。

7. KPI ビジュアルで省略記号アイコンを選択し、ドロップダウンメニューで **[条件付き書式設定]** を選択します。条件付き書式設定ペインで **[アイコンを追加する]** を選択し、リストの 3 番目のオプションであるアイコンセットのドロップダウンメニューで色付きの矢印を選択して、**[適用]** を選択します。
8. デフォルトのビジュアルラベルテキストを **[Total events trend]** に置き換えて、視覚化のタイトルを変更します。

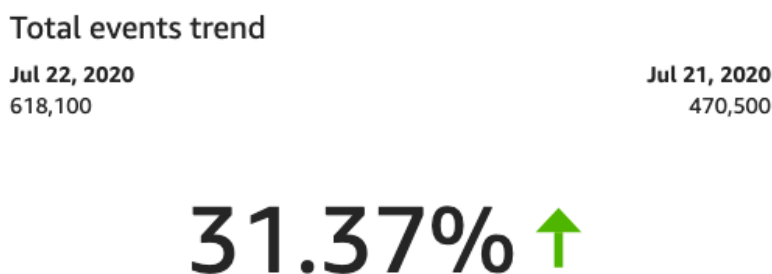


図 6: 合計イベントのトレンド KPI

## レベル完了率のトレンド KPI

全体のレベル完了率の傾向を示す KPI を作成するには、この手順を使用します。

1. 同じシートで、**[視覚化]** タブを選択し、データセットを **[Level completion rate by application version and day]** に変更します。
2. **[+ 追加]** を選択し、ドロップダウンリストで **[ビジュアルを追加]** を選択します。
3. **ビジュアルタイプ** ペインで、**[Key Performance Indicator (KPI)]** アイコンを選択します。
4. **フィールドリスト** ペインで、**[level\_completion\_rate]** をフィールドウェルの **[値]** にドラッグします。level\_completion\_rate (Sum) の横にあるドロップダウンメニューで、**[集計: 合計]** を選択し、**[平均]** に変更します。level\_completion\_rate と同様にドロップダウンメニューで、**[形式]**、**[その他のフォーマットオプション]** の順に選択します。データをフォーマットメニューで、少数位メニューを展開し、**[カスタム]** を選択して、ボックスの値を 2 に変更します。単位メニュー

を展開し、サフィックスダイアログボックスで % を追加します。**データをフォーマット**メニューを閉じます。

**注意:** フィールドリストペインが表示されていない場合は、**[視覚化]** を選択してください。

5. フィールドリストペインで、`[event_hour_utc]` をフィールドウエルの **[トレンドグループ]** にドラッグします。`event_hour_utc` の横にあるドロップダウン矢印を選択し、**[集計: 日]** が選択されていることを確認します。分析しようとしている傾向に基づいて集計単位を変更できます。

**注意:** この視覚化では、レベル完了率を最も詳細な時間の範囲で集計する、以前に作成したカスタム SQL クエリを使用します。そのため、このビジュアルでの「分」へのドリルダウンは、クエリを分単位でグループ化するように変更しないとサポートされません。これにより、視覚化のロード時間が遅くなる可能性があります。

6. **歯車**アイコンを選択してビジュアルを編集します。ビジュアルのフォーマットメニューの**比較方法**で、**[パーセント (%) としての差異]** を選択します。
7. KPI ビジュアルで省略記号アイコンを選択し、ドロップダウンメニューで **[条件付き書式設定]** を選択します。**[アイコンを追加する]** を選択し、**アイコンセット**のドロップダウンメニューで色付きの矢印を選択します。その次に **[適用]** を選択します。
8. デフォルトのビジュアルのラベルテキストを **[Level completion rate trend]** に置き換えて、視覚化のタイトルを変更します。



図 7: レベル完了率のトレンド KPI

## レベル完了率のゲージ

レベル完了率を示すゲージグラフを作成するには、この手順を使用します。

1. 同じシートで **[+ 追加]** を選択し、ドロップダウンリストで **[ビジュアルを追加]** を選択します。
2. **ビジュアルタイプ** ペインで、**[ゲージグラフ]** アイコンを選択します。
3. **フィールドリスト** ペインで、**[level\_completion\_rate]** をフィールドウェルの **[値]** にドラッグします。**level\_completion\_rate (Sum)** の横にあるドロップダウンメニューで、**[集計: 合計]** を選択し、**[平均]** に変更します。**level\_completion\_rate** と同様にドロップダウンメニューで、**[形式]**、**[その他のフォーマットオプション]** の順に選択します。**データをフォーマット** メニューで、**少数位** メニューを展開し、**[カスタム]** を選択して、ボックスの値を 2 に変更します。**単位** メニューを展開し、サフィックスダイアログボックスで **%** を追加します。**データをフォーマット** メニューを閉じます。
4. **歯車** アイコンを選択してビジュアルを編集します。**ビジュアルのフォーマット** メニューの**軸スタイル** の範囲で、**最小** ダイアログボックスを 0 に設定し、**最大** ダイアログボックスを 100 に設定します。
5. ゲージビジュアルで省略記号アイコンを選択し、ドロップダウンメニューで **[条件付き書式設定]** を選択します。**[アイコンを追加する]** を選択し、アイコンセットのドロップダウンメニューで色付きの矢印を選択して、**[適用]** を選択します。
6. デフォルトのビジュアルのラベルテキストを **[Overall level completion rate]** に置き換えて、視覚化のタイトルを変更します。

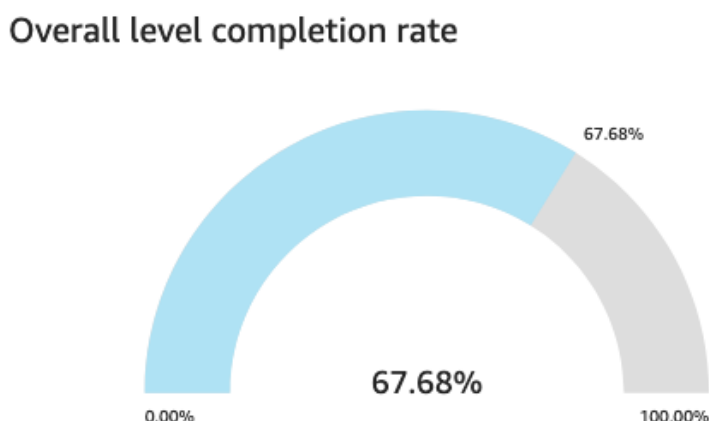


図 8: 全体的なレベル完了率

## ダークテーマ

ダッシュボードをダークテーマに変更するには、この手順を使用します。

1. 同じシートで、左側のナビゲーションバーから [テーマ] を選択します。
2. [Midnight] を選択します。

## タイトル

分析にタイトルを追加するには、この手順を使用します。

1. 同じシートで、[+ 追加] を選択し、ドロップダウンメニューで [タイトルを追加] を選択します。
2. 現在の分析ページの上部にタイトルフィールドが表示されます。そこに、分析のタイトルを入力してください。

**注意:** タイトルを削除するには、同じシートで [+ 追加] を選択し、ドロップダウンメニューで [タイトルの削除] を選択します。

## パラメータとコントロール

特定の時間範囲とアプリケーションバージョンのイベントを表示するダッシュボードにコントロールを追加するには、この手順を使用します。

1. 左側のツールバーで **[パラメータ]** を選択し、**[作成 ...]** を選択します。
2. **新しいパラメータを作成** ウィンドウの **名前** テキストボックスに `From` と入力します。**データタイプ** ドロップダウンで **[日時]** を選択し、**静的デフォルト値** ボックスで今日の日付を選択して、**[作成]** を選択します。
3. **パラメータが作成されました** ウィンドウで、**[コントロール]** を選択します。
4. **コントロールを追加** ウィンドウで、**表示名** ダイアログボックスに `From` と入力し、**[追加]** を選択します。
5. **パラメータ** ツールバーで、**[+]** を選択して別のパラメータを追加します。
6. **新しいパラメータを作成** ウィンドウの **名前** ダイアログボックスに `To` と入力します。**データタイプ** ドロップダウンで **[日時]** を選択し、**静的デフォルト値** ボックスで明日の日付を選択して、**[作成]** を選択します。
7. **パラメータが作成されました** ウィンドウで **[コントロール]** を選択します。
8. **コントロールを追加** ウィンドウで、**表示名** ダイアログボックスに `To` と入力し、**[追加]** を選択します。
9. **パラメータ** ツールバーで、**[+]** を選択して別のパラメータを追加します。
10. **新しいパラメータを作成** ウィンドウの **名前** ダイアログボックスに `AppVersion` と入力します。**データタイプ** ドロップダウンで **[文字列]** を選択し、**[動的デフォルト値を設定]** を選択します。
11. **単一の値 に動的なデフォルト値を設定する** ウィンドウで、**デフォルト値とユーザー情報を含むデータセット** のドロップダウンで **[raw\_events]** を選択し、**ユーザー名列** のドロップダウンで **[app\_version]** を選択し、デフォルト値の列で **[app\_version]** を選択して、**[適用]** を選択します。
12. **新しいパラメータを作成** ウィンドウで **[作成]** を選択します。
13. **パラメータが作成されました** ウィンドウで **[コントロール]** を選択します。
14. **コントロールを追加** ウィンドウで、**表示名** ダイアログボックスに `AppVersion` と入力し、**スタイル** のドロップダウンで **[ドロップダウン]** を選択し、**[データセットフィールドへのリンク]** ラジオ

ボタンを選択します。次に、**データセットの選択**のドロップダウンで `[raw_events]` を選択し、**Field** のドロップダウンで `[app_version]` を選択して、**[追加]** を選択します。

15. **フィルタ** ペインで、**[1 つ作成 ...]** を選択し、ドロップダウンメニューで `[event_timestamp_time_format]` を選択します。

16. 先ほど作成した新しい `[event_timestamp_time_format]` フィルタを選択します。

17. 最初のドロップダウンで、**[このビジュアルのみ]** を **[該当するすべてのビジュアル]** に変更し、**[パラメータを使用]** チェックボックスをオンにして、**開始日のパラメータ** のドロップダウンで **[From]** を選択し、**終了日のパラメータ** のドロップダウンで **[To]** を選択します。**[開始日を含める]** と **[終了日を含める]** のチェックボックスが選択されていることを確認し、**[適用]** を選択します。

18. **フィルタ** ペインで **[+]** を選択し、ドロップダウンメニューで `[app_version]` を選択します。

19. 作成した `[app_version]` フィルタを選択します。

20. 最初のドロップダウンで、**[このビジュアルのみ]** を **[該当するすべてのビジュアル]** に変更し、**フィルタタイプ** のドロップダウンで **[カスタムフィルタ]** を選択し、**[パラメータを使用]** ボックスをオンにして、次のドロップダウンで **[AppVersion]** を選択してから、**[適用]** を選択します。

21. Amazon Quicksight では、データセットごとに独自のフィルタが必要なため、**level completion by application version and day** データセットに対して `event_day` フィルタと `app_version` フィルタを作成する必要があります。これを行うには、視覚化で **[level completion rate by application version]** を選択し、**[フィルタ]** を選択して、ステップ 15 ~ 20 に従いフィルタを再作成します。時間フィルタの作成手順の変更のひとつに、フィルタする値として `[event_hour_utc]` を選択することがあります。

シートの上部にあるコントロールを調整して、目的の時間範囲の値を表示できます。また、特定のアプリケーションバージョンの値を表示することもできます。

分析の開発が完了したら、ビジュアルのサイズを変更して配置し、必要に応じてビジュアルのタイトルサイズを調整して、図 9 のようにすることができます。

次のダッシュボードの例に示すように、イベントタイプ別のイベント数、アプリケーションバージョン別のチュートリアルの新進捗、アプリケーションバージョン別のレベル完了率、時間の経過に伴うレベル完了率、および複数の KPI を含むダッシュボードを作成しました。

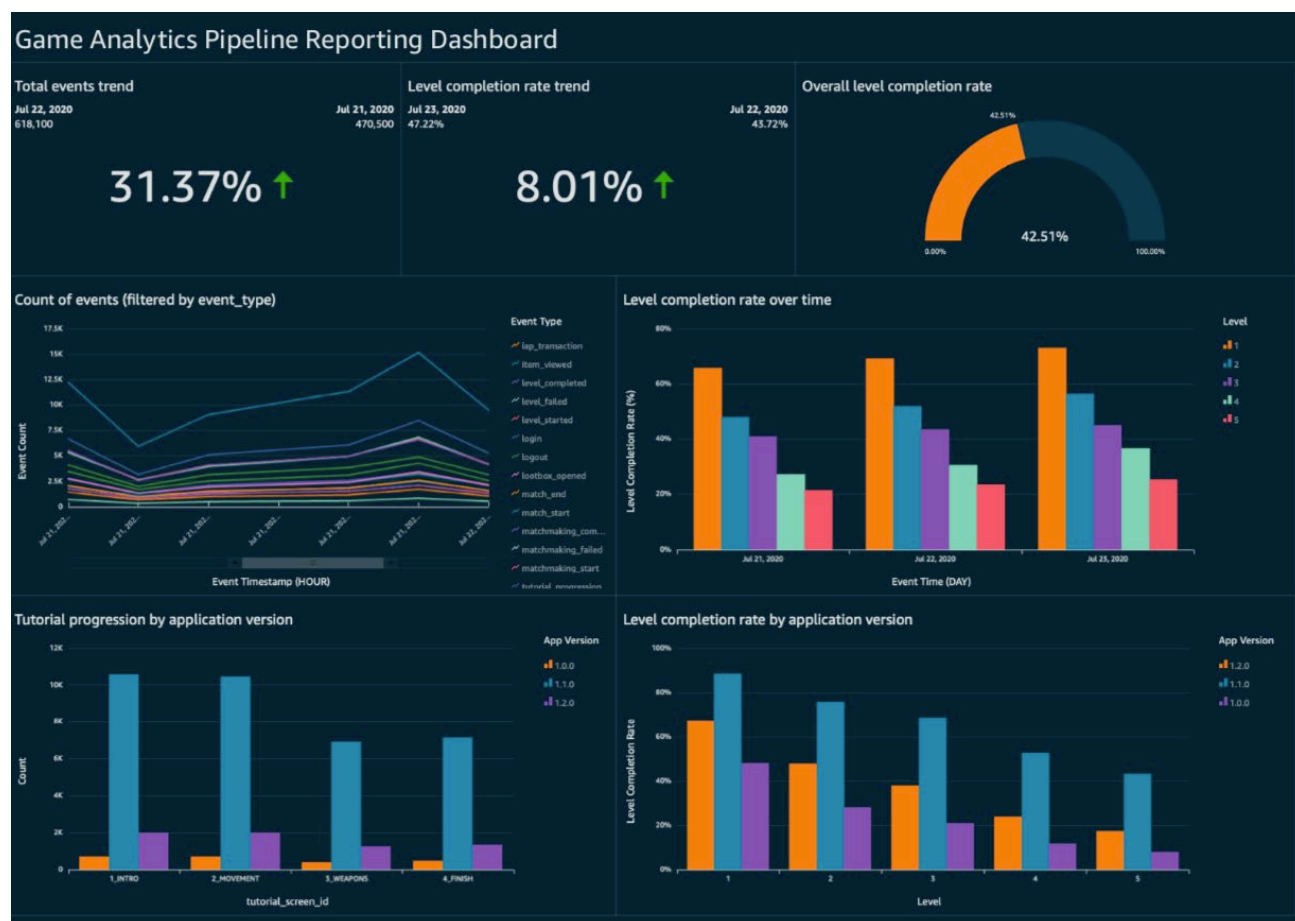


図 9: Amazon QuickSight ダッシュボードの例

Amazon QuickSight ビジュアルのカスタマイズの詳細については、[Amazon QuickSight ビジュアルの使用](#)を参照してください。

## ステップ 6. パイプラインの運用ダッシュボードにリアルタイムのメトリクスを表示して追加する

この手順を使用して、Amazon Kinesis Data Analytics によって生成され、AnalyticsProcessingFunction AWS Lambda 関数によって Amazon CloudWatch に発行されるカスタムメトリクスを表示します。カスタムメトリクスは、`<aws-cloudformation-stack name>/AWSGameAnalytics` 形式のカスタム名前空間を使用して Amazon CloudWatch に発行されます。

## Amazon CloudWatch 運用ダッシュボードにカスタムメトリクスを追加する

1. AWS CloudFormation スタックの出力タブから [RealTimeAnalyticsCloudWatch] のハイパーリンク値を取得して、Amazon CloudWatch コンソールに移動します。
2. Amazon CloudWatch コンソールで、ディメンション (例: **APPLICATION\_ID**) を選択し、表示する**メトリクス名** (例: **TotalEvents**) を選択します。Amazon CloudWatch は、コンソールにメトリクスをグラフ表示します。
3. [アクション] を選択し、[ダッシュボードに追加] を選択します。
4. **ダッシュボードに追加** ダイアログボックスで、`PipelineOpsDashboard_<your-cloudformation-stackname>` という名前のダッシュボードを選択します。[ダッシュボードに追加] を選択します。Amazon CloudWatch は、新しいウィジェットとしてダッシュボードにメトリクスを追加します。
5. Amazon CloudWatch ダッシュボードコンソールで、ダッシュボードのウィジェットを確認し、[ダッシュボードを保存] を選択します。

**注意:** この運用ダッシュボードにカスタムメトリクスを追加するには、ステップ 1 ~ 5 を繰り返してください。

このソリューションをテストして、ゲームデータを統合する準備ができれば、[ゲーム分析パイプライン開発者ガイド](#)を使用して、ゲームを新しいアプリケーションとしてソリューションに登録し、パイプラインへのイベントの送信を開始します。[ゲーム分析パイプライン開発者ガイド](#)の手順に従って、ソリューション API を使用してこのソリューションへのデータ送信をテストすることもできます。

**重要:** ソリューション機能のテストが完了したら、スクリプトを停止するか、スクリプトを引き続き使用してデプロイをさらにテストできます。料金は、スクリプトが実行され続ける限り適用されます。

# セキュリティ

AWS インフラストラクチャでシステムを構築する場合、セキュリティ上の責任はお客様と AWS の間で共有されます。この責任共有モデルにより、AWS がホストオペレーティングシステムと仮想化レイヤーからサービスが運用されている施設の物理的なセキュリティに至るまでのコンポーネントを運用、管理、および制御をするため、お客様の運用上の負担を軽減するのに役立ちます。AWS のセキュリティの詳細については、[AWS クラウドセキュリティ](#)を参照してください。

## 認証

AWS Identity and Access Management (IAM) ロールにより、AWS のサービスとユーザーに対してアクセスポリシーとアクセス許可を詳細に割り当てることができます。このソリューションでは、AWS Lambda 関数に付与する複数の IAM ロールを作成し、ソリューションで使用している他の AWS サービスにアクセスする権限を持つリソースをデプロイします。これらのロールは、AWS アカウントでゲーム分析のデータを収集、処理、保存することをサービスに許可するために必要です。ソリューション API では、[ゲーム分析パイプライン開発者ガイド](#)で説明されているように、IAM を使用して、アプリケーションと認証エンドポイントへのリクエストを認証および承認します。

## (オプション) イベントエンドポイントで IAM 認証を有効にする

ソリューション API のイベントエンドポイントは、`LambdaAuthorizer` AWS Lambda 関数で実装された API キー認証の代わりに、代替の Amazon API Gateway オートライザータイプを使用するように変更できます。例えば、ユーザー認証機能がない無料プレイゲームがあるとします。Amazon Cognito ID プール (フェデレーティッドアイデンティティ) を作成して、ゲームクライアントが認証されたユーザーと認証されていないユーザーの IAM 認証情報を生成できます。この認証情報は、テレメトリイベントを送信するために Amazon Kinesis や Amazon API Gateway などの AWS サービスと統合するために使用できます。

## 保管時の暗号化

このソリューションは、[Amazon S3 で管理された暗号キー](#) (SSE-S3) と AWS Key Management Service (AWS KMS) カスタマーマスターキー (CMK) を使用した保管時の暗号化を提供します。

## リソース

- [AWS CloudFormation](#)
- [AWS Lambda](#)
- [AWS Glue](#)
- [Amazon Kinesis](#)
- [Amazon Athena](#)
- [Amazon Simple Storage Service](#)
- [Amazon Simple Notification Service](#)
- [Amazon Cognito](#)
- [Amazon QuickSight](#)
- [Amazon DynamoDB](#)
- [Amazon API Gateway](#)
- [AWS Identity and Access Management \(IAM\)](#)
- [Amazon CloudWatch](#)
- [AWS X-Ray](#)
- [AWS Key Management Service \(KMS\)](#)

## 付録 A: ソリューションのモニタリング

「ゲーム分析パイプライン」ソリューションでは、Amazon CloudWatch メトリクスを統合して、リアルタイムのメトリクスとソリューションリソースの監視を行います。[Amazon CloudWatch Logs](#) では、ソリューションリソースからのログ記録レポートを提供しています。

### ゲーム分析パイプラインの運用ダッシュボード

このソリューションは、ストリーミングの取り込みと分析ストリーミングコンポーネントからの重要なメトリクスを集計する Amazon CloudWatch 運用ダッシュボードをデプロイします。このダッシュボードには、Amazon API Gateway、Amazon Kinesis リソース、AWS Lambda からの主要メトリクスが含まれます。

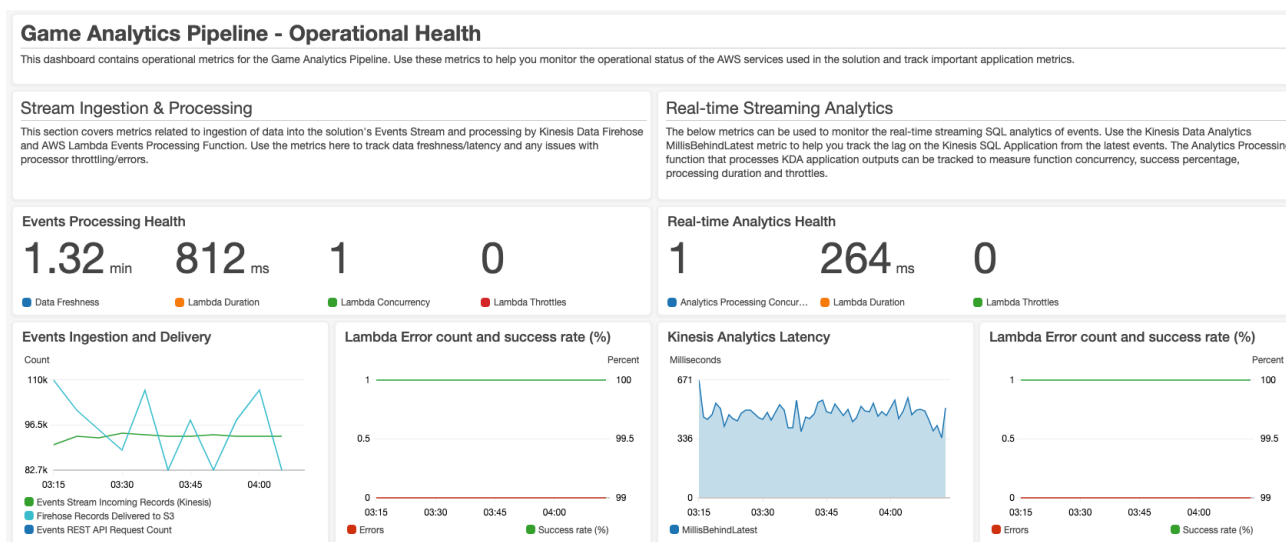


図 10: ソリューションでデプロイされた Amazon CloudWatch ダッシュボード

この運用ダッシュボードでは、イベントの取り込みを追跡し、管理者がパイプラインの正常性を監視できるようにメトリクスを処理します。このダッシュボードでは、データ取り込み率、データの最新性、および EventsProcessingFunction AWS Lambda 関数のパフォーマンスと状態を監視します。AWS CloudFormation テンプレートでストリーミング分析が有効になっている場合は、リアルタイムのストリーミング分析のメトリクスウィジェットに、AWS Lambda の処理状況と Amazon Kinesis Data Analytics の MillisBehindLatest メトリクスが事前設定されます。

## アラームと通知

このソリューションでは、特定の AWS リソースが使用率のしきい値を超えた場合、またはエラーステータスのしきい値を超えた（潜在的な運用上の問題を示す）場合にアラートを生成するいくつかの Amazon CloudWatch アラームが設定されています。

これらのアラートは、Amazon Simple Notification Service (Amazon SNS) 通知トピックに通知を送信するように設定されています。管理者は、スタックのデプロイ中に E メールアドレスを指定することで、このトピックをサブスクライブできます。次の Amazon CloudWatch アラームは、AWS CloudFormation テンプレートによって事前設定されています。

- **Amazon API Gateway REST API > 1% 4xx/5xx エラー率** - このソリューションでは、4xx または 5xx の API エラー率が 5 分にわたって 1% を超えた場合に、アラームと Amazon SNS 通知を生成するように設定されています。
- **AWS Lambda のエラーとスロットル** - このソリューションでは、各 AWS Lambda 関数のエラーとスロットルを監視し、これらの問題が発生したときにアラームを生成します。
- **Kinesis Throttling** - このソリューションは、Amazon Kinesis Data Streams `WriteProvisionedThroughputExceeded` および `ReadProvisionedThroughputExceeded` メトリクスを監視して、ストリームで読み取りまたは書き込みスロットリングが検出された場合は、アラームと Amazon SNS **通知**トピックを生成します。このアラームは、Amazon S3 に対する Amazon CloudWatch の `DataFreshness` メトリクスも追跡します。
- **Amazon DynamoDB のスロットリングとユーザー / システムエラー** - このソリューションでは、Amazon DynamoDB テーブルのエラーを監視し、`LambdaAuthorizer` AWS Lambda 関数がアクセスする `Authorizations` テーブルのスロットリングを追跡します。

## 付録 B: ソリューションのアンインストール

「ゲーム分析パイプライン」ソリューションは、AWS マネジメントコンソールまたは AWS コマンドラインインターフェイス (AWS CLI) を使用してアンインストールできます。ただし、Amazon Simple Storage Service (Amazon S3) バケットと Amazon QuickSight 分析とデータセットは手動で削除する必要があります。

**注意:** アンインストール中、AWS CloudFormation は Athena ワークグループを削除します。これにより、そのワークグループに関連付けられている保存済みの Athena クエリも削除されます。スタックを削除する前に、保持するクエリを保存します。

### AWS マネジメントコンソールを使用する

1. [AWS CloudFormation コンソール](#)にサインインします。
2. **スタック**ページで、このソリューションスタックを選択します。

### 3. [削除] を選択します。

**重要:** Amazon Simple Storage Service (Amazon S3) バケットと Amazon QuickSight 分析とデータセットは手動で削除する必要があります。

## AWS CLI を使用する

AWS CLI が自身の環境で使用できるかどうかを調べます。インストール手順については、AWS CLI ユーザーガイドの [AWS コマンドラインインターフェイスとは](#)を参照してください。AWS CLI が使用可能であることを確認したら、次のコマンドを実行します。

```
$ aws cloudformation delete-stack --stack-name <cloudformation-stack-name>
```

**重要:** Amazon Simple Storage Service (Amazon S3) バケットと Amazon QuickSight 分析とデータセットは手動で削除する必要があります。

## Amazon S3 バケットを削除する

このソリューションは、偶発的なデータ損失を防ぐために、analytics と solutionlogsbucket の Amazon S3 バケットを保持するように設定されています。このソリューションをアンインストールした後にデータを保持する必要がない場合は、これらのバケットを手動で削除する必要があります。

または、Amazon S3 バケットを自動的に削除するように AWS CloudFormation テンプレートを設定できます。スタックを削除する前に、AWS CloudFormation の [DeletionPolicy 属性](#)の削除動作を変更します。

## Amazon QuickSight 分析とデータセットの削除

この手順を使用して、Amazon QuickSight 分析とデータセットを手動で削除してください。

1. **raw\_events\_analysis** を削除します。手順については、*Amazon QuickSight* ユーザーガイドの [分析の削除](#) を参照してください。
2. **Level Completion rate** および **raw\_events** データセットを削除します。手順については、*Amazon QuickSight* ユーザーガイドの [データセットの削除](#) を参照してください。

## 付録 C: 運用メトリクスの収集

このソリューションには、匿名の運用メトリクスを AWS に送信するオプションが含まれています。当社はこのデータを使用して、お客様がこのソリューション、関連サービスおよび製品をどのように使用しているかをよりよく理解し、提供するサービスや製品の改善に役立てます。有効にすると、次の情報が収集され、AWS に送信されます。

- **Solution ID:** AWS ソリューション識別子
- **Unique ID (UUID):** このソリューションのデプロイごとにランダムに生成された一意の識別子
- **Timestamp:** データ収集タイムスタンプ
- **GameEventsStreamShardCount:** Game Events Kinesis Data Streams でプロビジョニングされたシャードの数
- **StreamingAnalyticsEnabled:** リアルタイムストリーミング分析がソリューションの一部としてデプロイされたかどうかを示すフラグ (true/false)
- **SolutionMode:** ソリューションのデプロイに使用されるパラメータ設定 (Dev、Prod)

AWS は、このアンケートを通じて収集されたデータを所有します。データ収集には、[AWS プライバシーポリシー](#) が適用されます。この機能を無効にするには、ソリューションをデプロイする前に、AWS CloudFormation テンプレートマッピングセクションを次のように変更します。

```
Mappings:
  Solution:
    Data:
      SendAnonymousData: 'Yes'
```

を次に変更します。

```
Mappings:
  Solution:
    Data:
      SendAnonymousData: 'No'
```

# ソースコード

[GitHub リポジトリ](#)にアクセスして、このソリューションのテンプレートとスクリプトをダウンロードし、カスタマイズを他のユーザーと共有できます。

## ドキュメントの改訂

日付	変更
2020 年 5 月	初回リリース
2020 年 9 月	この更新により、ソリューションが修正され、ソリューションによって取り込まれて処理された JSON イベントテレメトリからプレーヤー識別可能フィールドとクライアント識別可能フィールドの追跡が削除されます。バージョン 1.1.0 の詳細については、GitHub リポジトリの <a href="#">CHANGELOG.md</a> ファイルを参照してください。

### 注意

お客様は、この文書に記載されている情報を独自に評価する責任を負うものとします。このドキュメントは、(a) 情報提供のみを目的としており、(b) AWS の現行製品とプラクティスを表したものであり、予告なしに変更されることがあり、(c) AWS およびその関連会社、サプライヤー、またはライセンサーからの契約義務や確約を意味するものではありません。AWS の製品やサービスは、明示または暗示を問わず、いかなる保証、表明、条件を伴うことなく「現状のまま」提供されます。お客様に対する AWS の責任は、AWS 契約により規定されます。本書は、AWS とお客様の間で行われるいかなる契約の一部でもなく、そのような契約の内容を変更するものでもありません。

「ゲーム分析パイプライン」ソリューションは、MIT No Attribution <https://spdx.org/licenses/MIT-0.html> の条件に基づいてライセンスされます。

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.