

# Google Cloud の「イベントメッシュ」でゲームにおけるアナリティクスを加速する

高田 泰志

Google Cloud 技術部長

ヒューストン ベンジャミン

Google Cloud カスタマー エンジニア



# スピーカー自己紹介



高田 泰志

Google Cloud  
技術部長

日系 Sler にて、システム エンジニアとしてエンジニアのキャリアを開始する。ミッション クリティカルなシステムにアーキテクトとして参画し、フレームワーク / ライブラリの開発を行う。2017 年 8 月に Google Cloud にジョイン。ゲーミング チームのカスタマー エンジニアとして、お客様の開発を支援。2020 年 9 月より現職。



# スピーカー自己紹介



ヒューストン ベンジャミン

Google Cloud  
カスタマーエンジニア

Google Cloud ゲーミング チームのカスタマーエンジニアです。Kubernetes-native の CI / CD、GitOps、MLOps、イベントドリブン アーキテクチャにフォーカスし、ゲーム業界のお客様を支援。Google 入社以前は、機械学習の R&D やプラットフォーム エンジニアリングの分野で活躍していました。

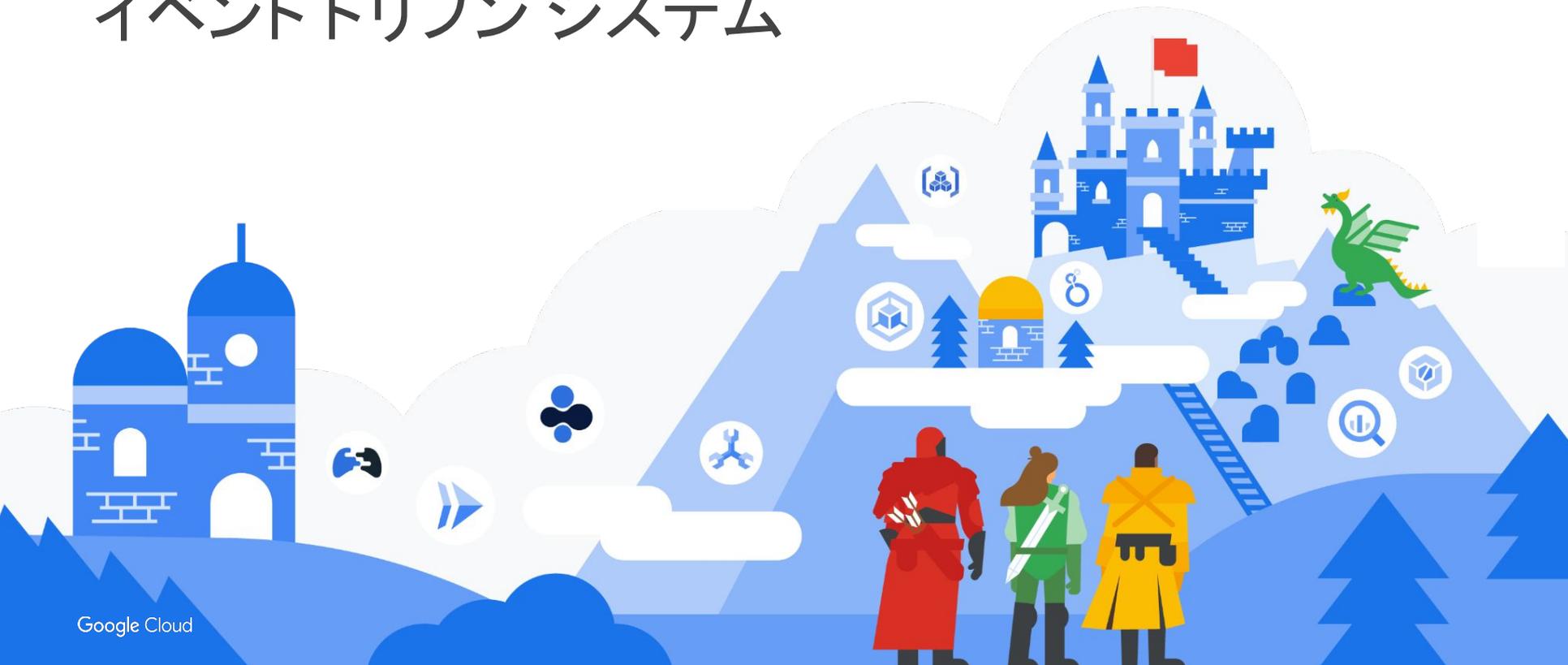


# TL;DR;

- 01 | イベントドリブン アーキテクチャにより、ゲーム開発者はサービスの依存関係を疎結合化、回復力を高め、データ共有を簡素化することができます
- 02 | Google Cloud の Eventarc は、ゲームのユースケースにおいて、イベントドリブン システム構築を根本的に簡素化する、次世代の汎用イベント管理フレームワークです
- 03 | ゲーム開発者は、Eventarc を使用することで、BigQuery や Vertex AI など Google Cloud のアナリティクスソリューションで得られたインサイトを、最小限のコード変更で簡単にゲームに直接取り込むことができます



# イベントドリブンシステム

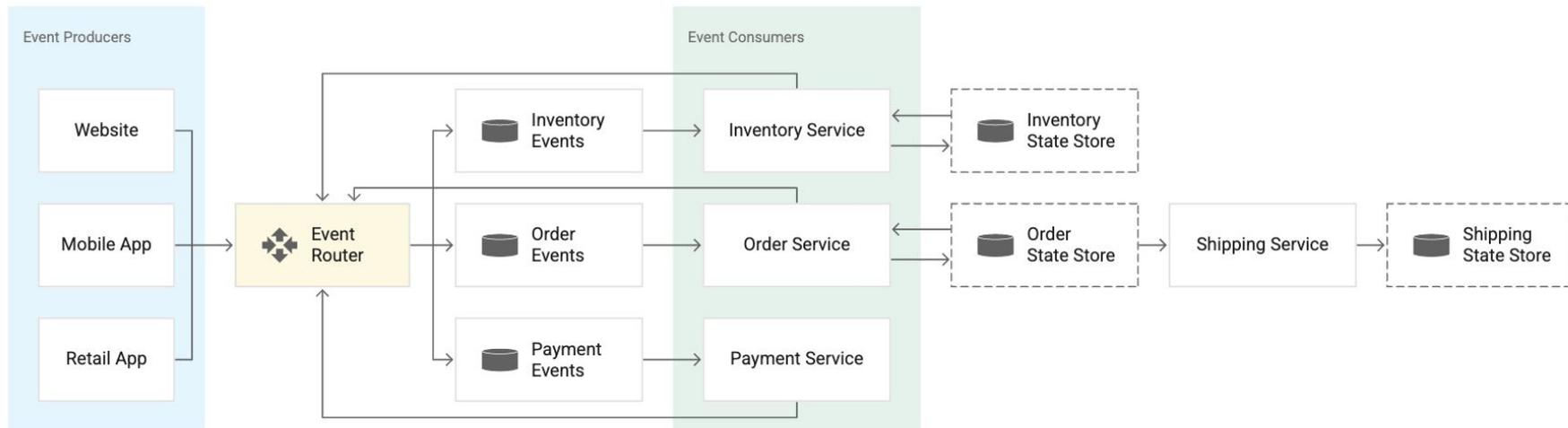


# イベントドリブンシステム

イベントドリブンシステムとは？ マイクロサービスがイベントと呼ばれる状態の変化に反応するソフトウェアのデザイン パターンに基づいたシステム

## メリット:

- ・各サービスの依存関係を切り離す
- ・システムの回復力を高める
- ・情報ソースの簡素化



# ゲームのイベントドリブン ユースケース

- オブジェクトストレージ、データベース テーブル、コンテナ、その他のリソースに保存されているゲーム ステータスの変更を監視し、**アラートを受信**します。
- 1つのゲーム イベントを複数のコンシューマにファンアウトします。イベントルーターを利用し、カスタマイズしたコードを書かなくても、ゲームイベントをすべての適切なイベント コンシューマーにプッシュすることができます。ゲームサービスは、**同じイベントを並行して処理**することができます。
- 各ゲーム開発コンポーネントの独立性を維持しながら、**共有されるサービスの相互運用性**を提供します。イベントドリブン アーキテクチャにより、チーム間の依存関係を軽減し、特定のゲームサービス イベントの所有権を容易に移管することができます。

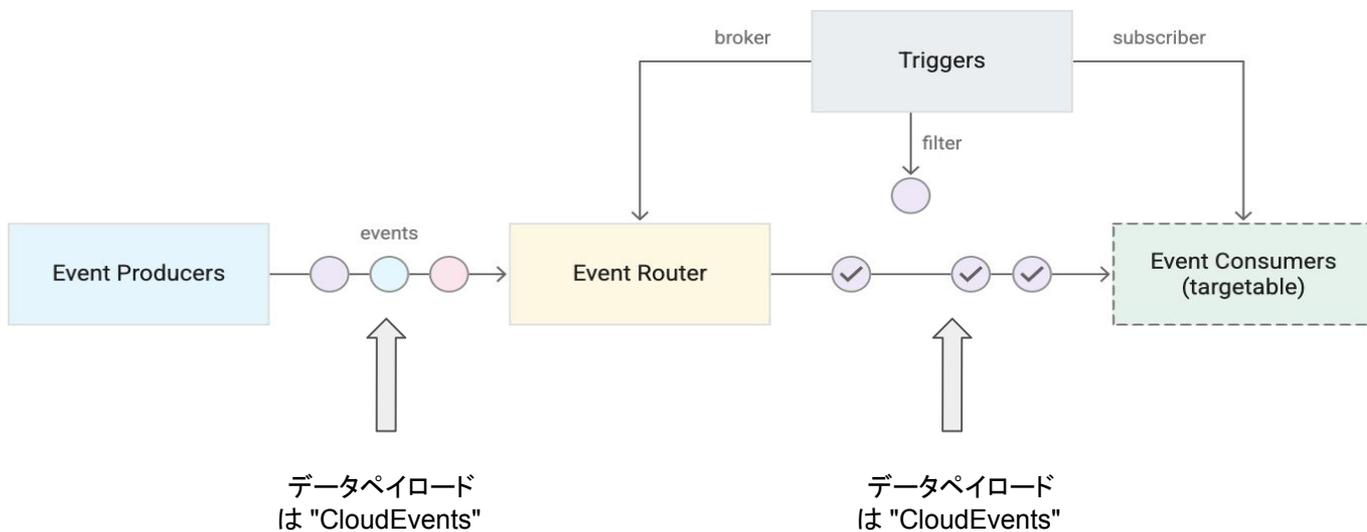


# イベントメッシュでイベント管理を簡素化

ゲームのイベントドリブンシステムをもっと簡単に構築できるようになるには？

- すべてのシステムの入出力フォーマットを標準化する
- イベントソース間のカスタム コネクタを不要にする

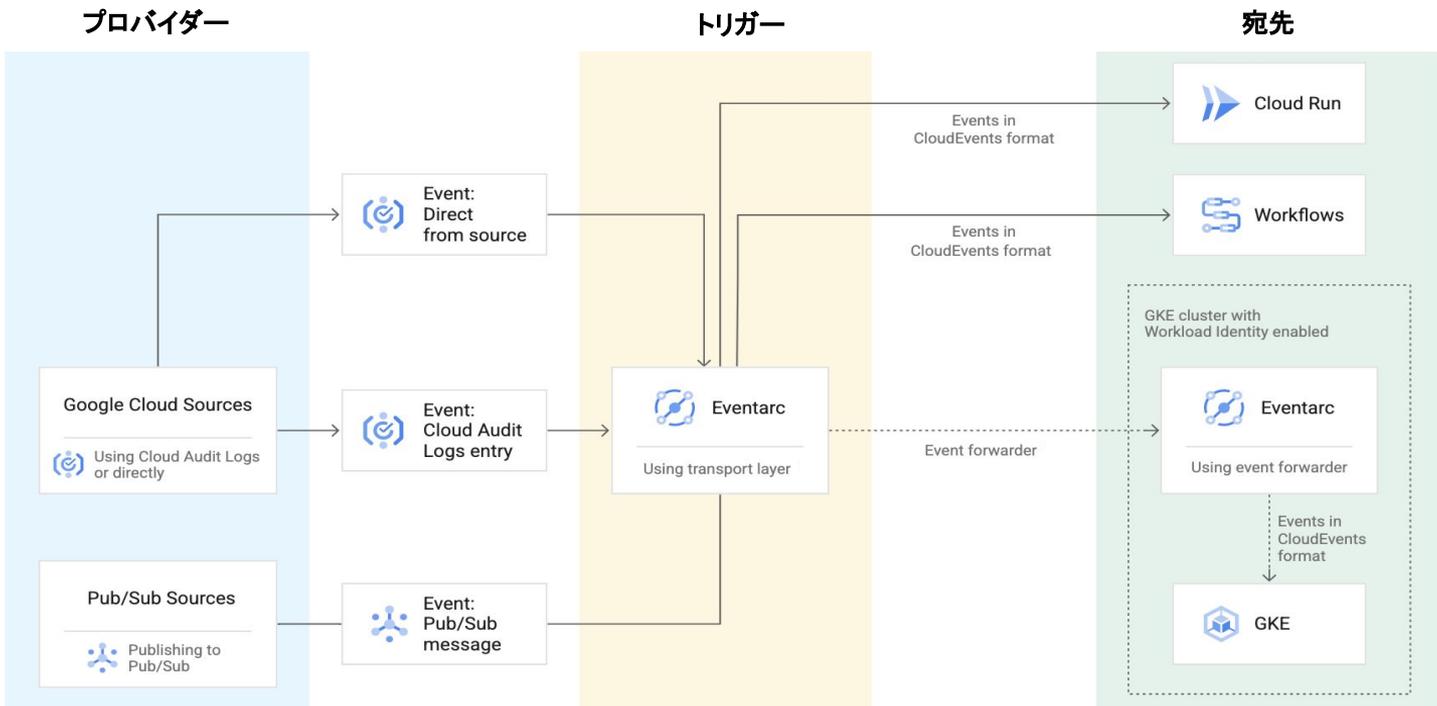
Google Cloud の Eventarc のようなイベントメッシュは、上記の 2 つの機能を提供するフレームワーク





# Eventarc の概要

**Eventarc** は、状態の変化に反応する疎結合なサービスを使って、Google Cloud のサービスや独自のアプリケーションからイベントを非同期に配信することができます。Eventarc はインフラ管理を必要としません。イベントドリブンのソリューションを構築しながら、生産性とコストを最適化することができます。



# Eventarc の概要

## イベントプロバイダー(ソース)

イベントは、イベントプロバイダ(ソース)から関心のあるイベントコンシューマに転送される。ルーティングはイベントに含まれる情報によって実行されるが、イベントは特定のルーティングデスティネーションを特定しない。

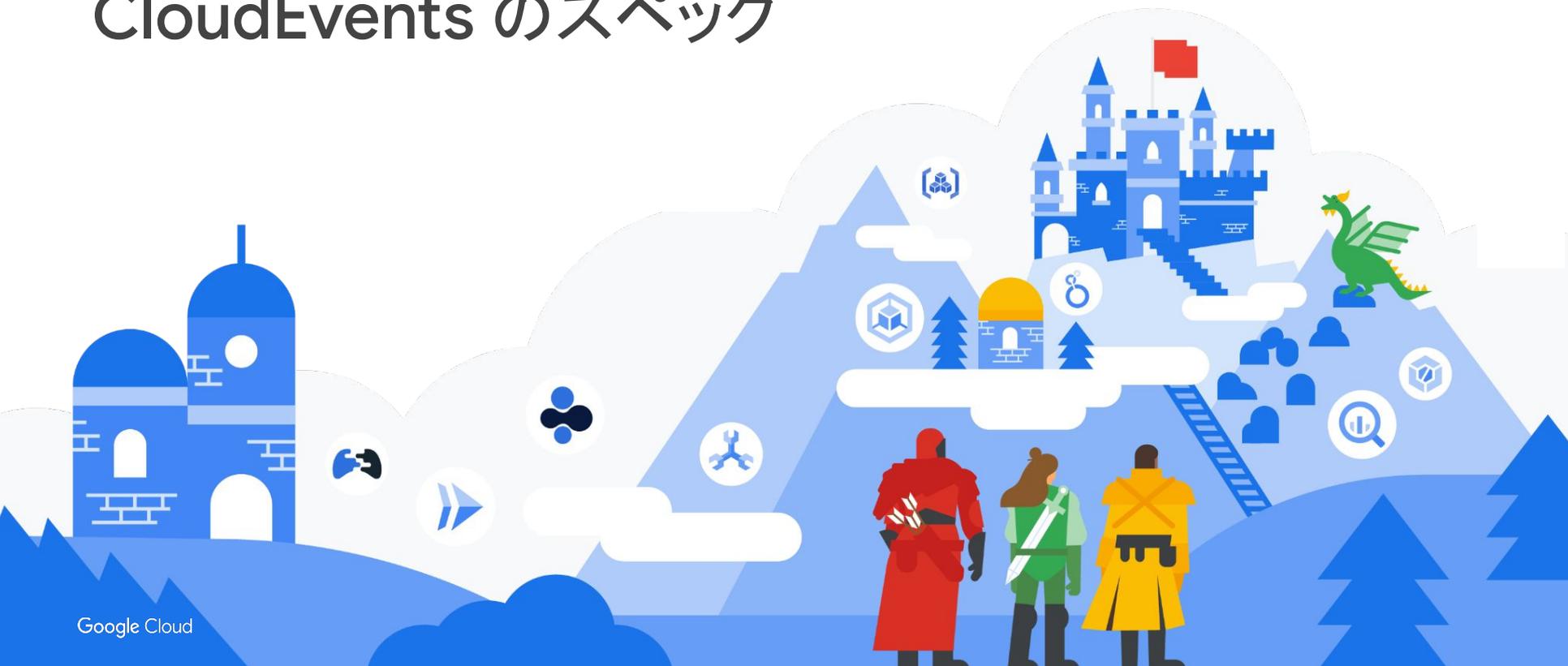
## イベントトリガー

トリガーとは、特定のイベントや一連のイベントに対する関心表明です。トリガーを作成するとき、イベントソースからターゲットデスティネーションへのルーティングを含め、特定のイベントを捕捉して対処することができるトリガーのフィルターを指定します。

## 宛先 (シンク)

イベントは、Pub/Sub のプッシュ通知によって、イベントコンシューマと呼ばれる特定の宛先にルーティングされます。

# CloudEvents のスペック





# cloudevents

---

A specification for describing event data in a common way

# CloudEvents のスペック

CloudEvents は、イベントデータを記述するための仕様であり、サービス、プラットフォーム、システム間のイベントの相互運用性を大幅に向上させるものである。

CloudEvents は以下を提供します。

- イベント処理、またはイベント記述のための統一スキーマ
- 異なる環境間でイベントデータの配信を標準化する共通のライブラリ

CloudEvents を[参照](#)

## CloudEvents データ ペイロードの例

```
{
  "specversion" : "1.0",
  "type" : "com.game-xyz.level.completed",
  "source" :
  "https://game-xyz.com/cloudevents/level",
  "subject" : "level-change",
  "id" : "A234-1234-1234",
  "time" : "2018-04-05T17:31:00Z",
  "datacontenttype" : "application/json",
  "data" : {
    "playerStatA" : "abc",
    "playerStatB" : 123,
    "playerStatC" : true
  }
}
```

# CloudEvents SDKs

CloudEvents のデータ構造を操作するため、多数の言語のライブラリが利用可能です。

- Go
- JavaScript
- Java
- C#
- Ruby
- PHP
- Python
- Rust
- PowerShell

SDK を [参照](#)

## C# CloudEvents SDK の例

```
using CloudNative.CloudEvents;
using CloudNative.CloudEvents.Http;
using CloudNative.CloudEvents.NewtonsoftJson;

// Create and marshal a CloudEvent
CloudEvent cloudEvent = new CloudEvent
{
    Id = "event-id",
    Type = "event-type",
    Source = new Uri("https://cloudevents.io/"),
    Time = DateTimeOffset.UtcNow,
    DataContentType = "text/plain",
    Data = "This is CloudEvent data"
};

CloudEventFormatter formatter = new JsonEventFormatter();
HttpRequestMessage request = new HttpRequestMessage
{
    Method = HttpMethod.Post,
    Content =
cloudEvent.ToHttpContent(ContentMode.Structured, formatter)
};
```

# 他の CloudEvents SDK の例

## PHP の例

```
use CloudEvents\V1\CloudEvent;
use CloudEvents\Http\Marshaller;
use CloudEvents\Http\Unmarshaller;

// Create and marshal a CloudEvent
$newEvent = new CloudEvent(
    '1n6bFxDMHZFCh1I4TVI9tdzphB9',
    '/examples/php-sdk',
    'com.example.type',
    ['example' => 'first-event'],
    'application/json'
);
$request =
    Marshaller::createJsonMarshaller()->marshalStructuredRequest($newEvent);

// Unmarshal a CloudEvent
$event = Unmarshaller::createJsonUnmarshaller()->unmarshal($request);
```

## Python の例

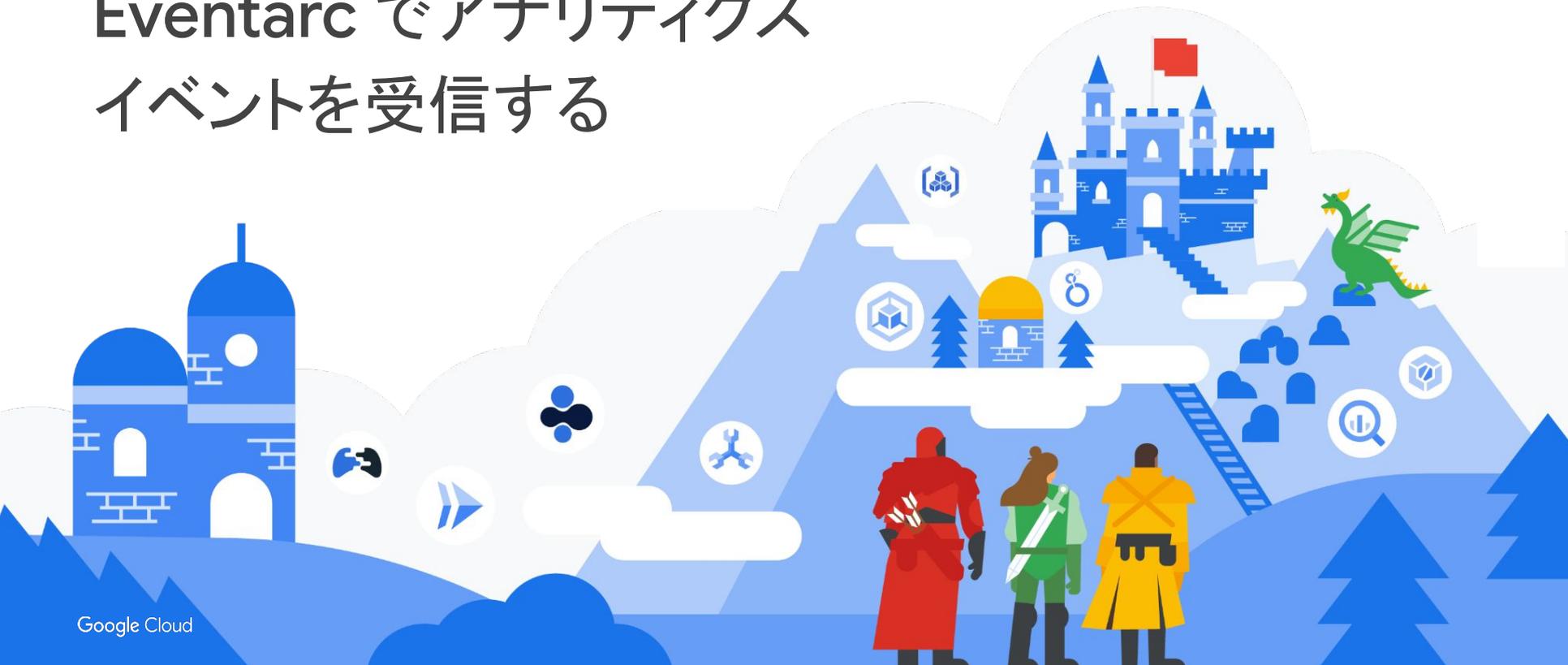
```
from cloudevents.http import CloudEvent, to_structured
import requests

# Create a CloudEvent
attributes = {
    "type": "com.example.sampletype2",
    "source": "https://example.com/event-producer",
}
data = {"message": "Hello World!"}
event = CloudEvent(attributes, data)

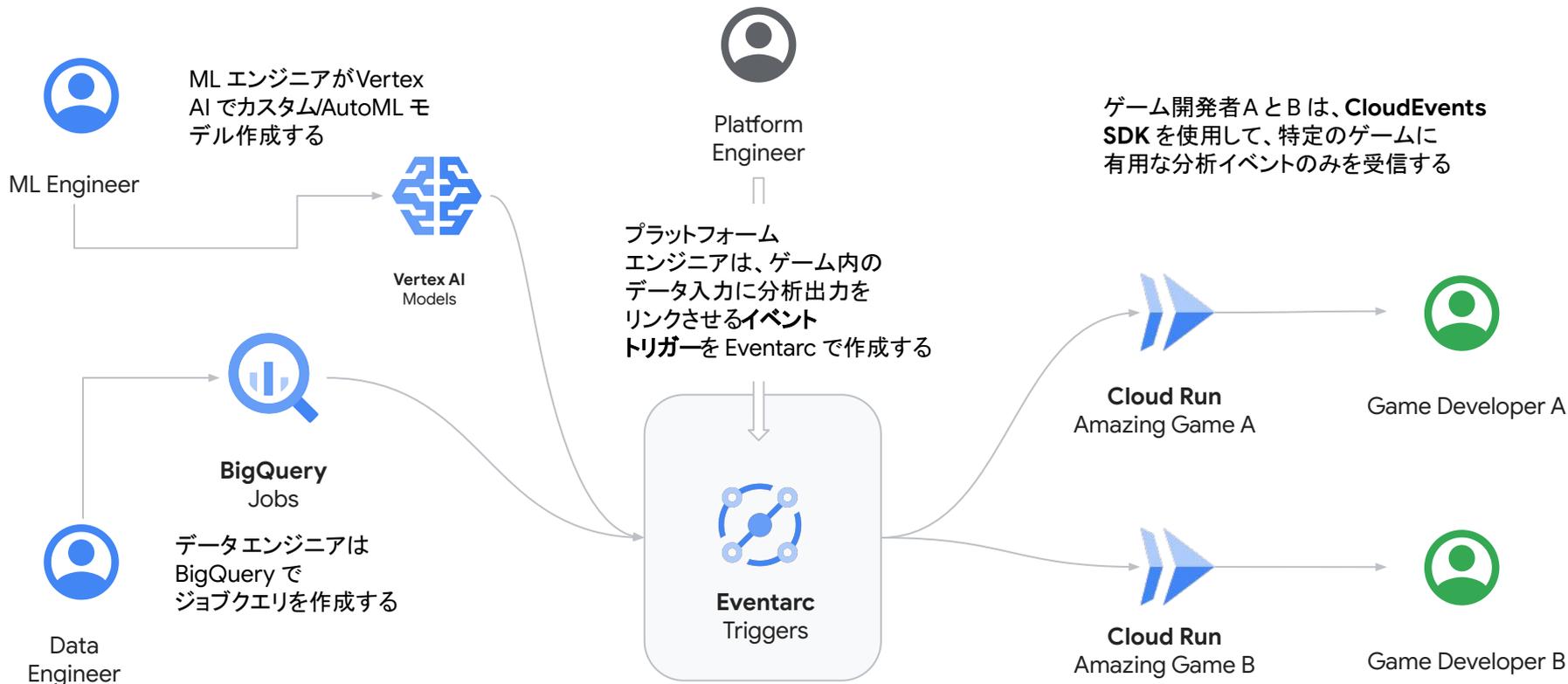
# POST CloudEvent
headers, body = to_structured(event)
requests.post("<some-url>", data=body, headers=headers)
```



# Eventarc でアナリティクス イベントを受信する



# ゲームスタジオの Eventarc 活用法



# Eventarc トリガーの管理

## Terraform Provider

```
resource "google_eventarc_trigger" "primary" {
  name = "name"
  location = "europe-west1"
  matching_criteria {
    attribute = "type"
    value =
"google.cloud.pubsub.topic.v1.messagePublished"
  }
  destination {
    cloud_run_service {
      service =
google_cloud_run_service.default.name
      region = "europe-west1"
    }
  }
  labels = {
    foo = "bar"
  }
}
```

Terraform を[参照](#)

## Kubernetes Provider

```
apiVersion: eventarc.cnrm.cloud.google.com/v1beta1
kind: EventarcTrigger
metadata:
  name: eventarc-trigger-sample
  labels:
    foo1: bar1
spec:
  location: us-central1
  destination:
    cloudRunService:
      serviceRef:
        external: eventarc-trigger-dep
      region: us-central1
  serviceAccountRef:
    name: eventarc-trigger-dep
  transport:
    pubsub:
      topicRef:
        name: eventarc-trigger-dep
  matchingCriteria:
  - attribute: "type"
    value: "google.cloud.pubsub.topic.v1.messagePublished"
  projectRef:
    external: "projects/my-project-id"
```

Kubernetes を[参照](#)

## CLI

```
gcloud eventarc triggers create my-trigger \
  --event-filters="type=google.cloud.pubsub..." \
  --destination-run-service=my-service
```

gcloud CLI を[参照](#)

Google Cloud Platform gameops-001

Eventarc Create trigger

Trigger name \*  
amazing-game-trigger-001

Event provider \*  
AI Platform (under Vertex AI)

Event \*  
google.cloud.aiplatform.v1beta1.PredictionService.Predict

SHOW DETAILS

Resource

Any resource

Specific resource

Path pattern

Full resource name  
projects/\_/buckets/eventarc-bucket/objects/game-001-ml-predictions.txt

Only the events in this project that match this filter will be included. [Learn more](#)

Region \*  
asia-northeast1 (Tokyo)

Service account  
Compute Engine default service account

Identity to be used by the created Eventarc trigger.

Event destination \*  
Cloud Run

Select a Cloud Run service \*  
amazing-game-001

Service URL path  
/ml-predictions

e.g. /route, /route/subroute?param=1

CREATE

1 Google Cloud コンソールでは、**Event Provider**と**Event Destination**のフィールドに入力することで、簡単に**Event Trigger**を定義することができます

2 イベントプロバイダーごとに数百のイベントタイプから選択可能です

Event \*

Filter prediction

- google.cloud.aiplatform.v1.PredictionService.Predict
- google.cloud.aiplatform.v1.PredictionService.RawPredict
- google.cloud.aiplatform.v1beta1.JobService.CancelBatchPredictionJob
- google.cloud.aiplatform.v1beta1.JobService.CreateBatchPredictionJob
- google.cloud.aiplatform.v1beta1.JobService.GetBatchPredictionJob
- google.cloud.aiplatform.v1beta1.JobService.ListBatchPredictionJobs
- google.cloud.aiplatform.v1beta1.PredictionService.Explain
- google.cloud.aiplatform.v1beta1.PredictionService.Predict

3 「CREATE」をクリックすると、実際にVertex AI モデル予測のイベントトリガーが作成されます。将来のモデル予測は、自動的にHTTP cloud eventとしてCloud Run ゲームに転送されます。

# Vertex AI モデル予測 トリガーの作成

Google Cloud Platform gameops-001

Eventarc Trigger details EDIT DELETE

Details

|                      |   |
|----------------------|---|
| Name                 | amazing-game-vertex-ai-trigger  |
| Region               | asia-northeast1   |
| Service account      | 517567087097-compute@developer.gserviceaccount.com  |
| Event provider       | AI Platform (under Vertex AI)   |
| Event type           | google.cloud.audit.log.v1.written   |
| Event                | google.cloud.aiplatform.v1beta1.PredictionService.  |
| Path pattern         | projects/_/buckets/eventarc-bucket/objects/game-001-ml-predictions/*                                    |
| Topic                | <a href="#">projects/gameops-001/topics/eventarc-asia-northeast1-amazing-game-vertex-ai-trigger-201</a> |
| Infrastructure       | Cloud Audit Logs  |
| Destination platform | Cloud Run   |
| Destination          | <a href="#">amazing-game-001 (asia-northeast1)</a>  |
| Error conditions     | None  |
| Last updated         | Jun 4, 2022, 1:56:25 PM   |

Google Cloud Platform gameops-001

Eventarc Create trigger

Trigger name \*  
amazing-game-bigquery-trigger

Event provider \*  
BigQuery

Event \*  
google.cloud.bigquery.v2.JobService.Query

SHOW DETAILS

Resource

Any resource

Specific resource

Path pattern

Path pattern  
projects/\_/buckets/eventarc-bucket/objects/game-001-bq-queries/\*  
e.g. /projects/\_/buckets/bucket-1/objects/\* Only resources in this project that match this path pattern filter will be included. [Learn more](#)

Region \*  
asia-northeast1 (Tokyo)

Service account  
Compute Engine default service account  
Identity to be used by the created Eventarc trigger.

Event destination \*  
Cloud Run

Select a Cloud Run service \*  
amazing-game-001

Service URL path  
/bq-queries  
e.g. /route, /route/subroute?param=1

CREATE CANCEL

1 Google Cloud コンソールでは、BigQuery ジョブのクエリ結果を返す Event Trigger を定義することも可能です

# BigQuery ジョブのクエリトリガーの作成

2 すべての Eventarc イベントは、非常によく文書化された Google Cloud サービスメソッドに対応しています

Method: jobs.query

Runs a BigQuery SQL query synchronously and returns query results if the query completes within a specified timeout.

HTTP request

POST https://bigquery.googleapis.com/bigquery/v2/projects/{projectId}/queries

The URL uses gRPC Transcoding syntax.

Path parameters

Parameters

string  
Required. Project ID of the query request.

3 イベントのレスポンスボディには、BigQuery ジョブのクエリ結果が含まれます。クラウド Audit Log のデータ入力として最初に書き込まれます。そして、Eventarc はこの Audit Log を cloud event に変換します。クエリの結果は「data」フィールドに格納されます。

|  | Cloud Audit Logs  | Cloud Storage | Pub/Sub                                 |
|--|---|---------------|---|
| Proto files  | google/events/cloud/audit/v1/events.proto   |               | google/events/cloud/audit/v1/data.proto |
| Envelope message                                   | google.events.cloud.audit.v1.AuditLogWrittenEvent   |               |   |
| Data message                                       | google.events.cloud.audit.v1.LogEntryData   |               |   |
| Event type<br>(when an audit log entry is written) | google.cloud.audit.log.v1.written   |               |   |
| Event extensions                                   | <ul style="list-style-type: none"> <li>serviceName</li> <li>methodName</li> <li>resourceName</li> </ul> |               |   |
| Example  | Pub/Sub audit log<br>BigQuery audit log   |               |   |

# BigQueryジョブの クエリトリガーの作成

### JSON representation

```
{
  "kind": string,
  "schema": {
    object (TableSchema)
  },
  "jobReference": {
    object (JobReference)
  },
  "totalRows": string,
  "pageToken": string,
  "rows": [
    {
      object
    }
  ],
  "totalBytesProcessed": string,
  "jobComplete": boolean,
  "errors": [
    {
      object (ErrorProto)
    }
  ],
  "cacheHit": boolean,
  "numDmlAffectedRows": string,
  "sessionInfo": {
    object (SessionInfo)
  },
  "dmlStats": {
    object (DmlStats)
  }
}
```

4 Eventarc が作成する cloud event 内には、BigQuery ジョブのクエリ結果が「rows」キーの中に含まれます

rows[]

object (Struct format)

An object with as many results as can be contained within the maximum permitted reply size. To get any additional rows, you can call `jobs.getQueryResults` and specify the `jobReference` returned above.

5 「rows」キーは、CloudEvents スペックの「data」部分に含まれます。

```
{
  "data": {
    "insertId": "1234567",
    ...
    "methodName": "jobservice.query",
    "resourceName": "projects/test-project/jobs/sample-job",
    "serviceData": {
      ...
      "rows": [
        {
          "playerStatA": "abc",
          "playerStatB": 123,
          "playerStatC": true,
          ...
        }
      ]
    }
  },
  "datacontenttype": "application/json; charset=utf-8",
  ...
  "specversion": "1.0",
  "type": "google.cloud.audit.log.v1.written"
}
```

# データの流れの例



CloudEvents の転送



Eventarc  
'BigQuery to Cloud Run'  
Trigger

CloudEvents の転送

Cloud Run

in-game update



Excited  
Game Players

## CloudEvents データペイロード

```
{
  "data": {
    "insertId": "1234567",
    ...
    "methodName": "jobservice.query",
    "resourceName": "projects/test-project/jobs/sample-job",
    "serviceData": {
      ...
      "rows": [{
        "playerId": "bestPlayer001",
        "gameId": "AmazingGameXYZ",
        "score": 999,
        ...
      }],
    },
    "datacontentType": "application/json; charset=utf-8",
    ...
    "specversion": "1.0",
    "type": "google.cloud.audit.log.v1.written"
  }
}
```

## C#で書かれたUnity エンジンのゲーム

```
using CloudNative.CloudEvents;
using CloudNative.CloudEvents.Http;
using CloudNative.CloudEvents.NewtonsoftJson;
using UnityEngine;
using UnityEngine.Networking;

public class GameUpdate
{
    [JsonProperty("playerId")]
    public string PlayerId { get; set; }

    [JsonProperty("gameId")]
    public string GameId { get; set; }

    [JsonProperty("score")]
    public int Score { get; set; }
}

CloudEventFormatter formatter = new JsonEventFormatter();
CloudEvent cloudEvent = await request.ToCloudEventAsync(formatter);
JsonObject dataAsJsonObject = (JsonObject) cloudEvent.Data;
GameUpdate gameInput = dataAsJsonObject.ToObject<GameUpdate>();

public class AmazingUnityGameLogic
{
    // Unity Engine game logic that uses gameInput...
    // ...
    // ...
}

// Etc etc...
```

# 次のステップへ



Eventarc

---

[Read the docs](#)



CloudEvents

---

[Read the docs](#)



GitHub のサンプル

---

[See the latest on Github](#)



# まとめ

01 | イベントドリブン アーキテクチャにより、ゲーム開発者はサービスの依存関係を疎結合化、回復力を高め、データ共有を簡素化することができます

02 | Google Cloud の Eventarc は、ゲームのユースケースにおいて、イベントドリブンシステム構築を根本的に簡素化する、次世代の汎用イベント管理フレームワークです

03 | ゲーム開発者は、Eventarc を使用することで、BigQuery や Vertex AI などの Google Cloud のアナリティクス ソリューションで得られたインサイトを、最小限のコード変更で簡単にゲームに直接取り込むことができます



# Thank you

