



# 日経の DX を支える BigQuery を中心とした分析基盤の運用

小室貴之

日本経済新聞社

# スピーカー自己紹介

- 名前: 小室 貴之 (Takayuki Komuro)
- 所属
  - 2013 - 2017: 株式会社ドワンゴ
  - 2017 - 2018: 株式会社マンバ
  - 2018-: 日本経済新聞社
- やっていること:
  - データ分析基盤上のアプリケーションと  
インフラの開発・運用

# 今日話すこと

- 日経におけるデータ分析基盤の課題
- BigQuery の活用方法
- Data Catalog によるドキュメンテーション
- Cloud Composer によるジョブ管理



# 日本経済新聞社におけるデータ分析

# 日経電子版とは

- 日経の記事を Web で配信
- 有料ユーザー 80 万人以上
- 無料ユーザー 450 万人以上



# 日経におけるデータ分析

- データ量
  - ユーザーイベントのみで1日約1.5億レコード
- 利用人数
  - 間接的な利用を含めるとほぼ全社員(約3000人)
- クエリ実行数
  - アドホック: 約2000件/日
  - バッチ: 約1万件/日

# 日経におけるデータ活用例

- 編集部向けリアルタイム ダッシュボード
- 記事推奨
- マーケティング施策分析
- リアルタイム マーケティング オートメーション
- ...etc

# 編集向けダッシュボード

## 記事詳細

記事	記事タイトル 🔒	公開日時	2021-XX-XX XX:XX	媒体	電子版
出稿部	東京証券部	記者名	日経 太郎	掲載箇所	マーケットトップ オピニオントップ
				文字数	4,171

※公開からの経過時間によって、更新頻度が異なります。?

エンゲージメントスコア?: 75.9

★★★★☆ 3.8

読者数?: xxx,535

★★★★☆ 4.4

有料会員: xxx,682(71.5%)

無料会員: xxx,634(27.4%)

アクセス数?: xxx,689

★★★★☆ 4.4







# BigQuery を採用した背景

# 課題 1: スケールの難しさ

- 日々増大していくクエリの実行やデータ量に対して  
従来利用していたデータウェアハウスでは  
スケールするのが難しかった
  - ストレージの追加が追いつかず、古いデータをDWH  
の外部に退避させていた
  - 夜間などの大量のバッチが実行される時間帯に  
一時的にコンピューティング リソースが  
足りなくなる現象が発生していた

## 課題 2 : 運用が追いつかない

- 利用者からの依頼でテーブル作成やカラム変更、DWH へのアクセス権の付与等をデータエンジニアが行っていたが、依頼が増えるに連れてその作業に割かれる時間が増えていった



# スケーラビリティの解決

# BigQuery とは

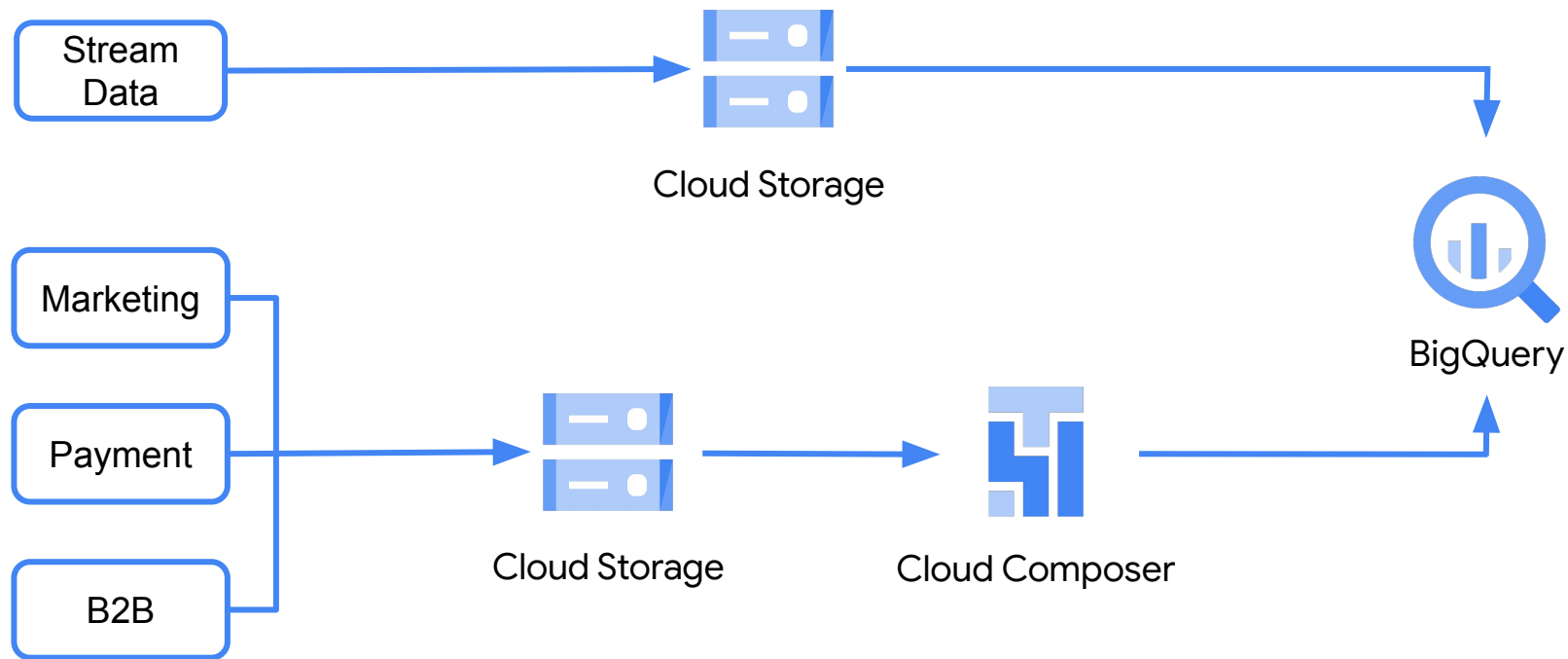


ビジネスのアジリティに対応して設計された、サーバーレスでスケーラビリティと費用対効果に優れたマルチクラウド データウェアハウスです。”

出典: <https://cloud.google.com/bigquery?hl=ja>

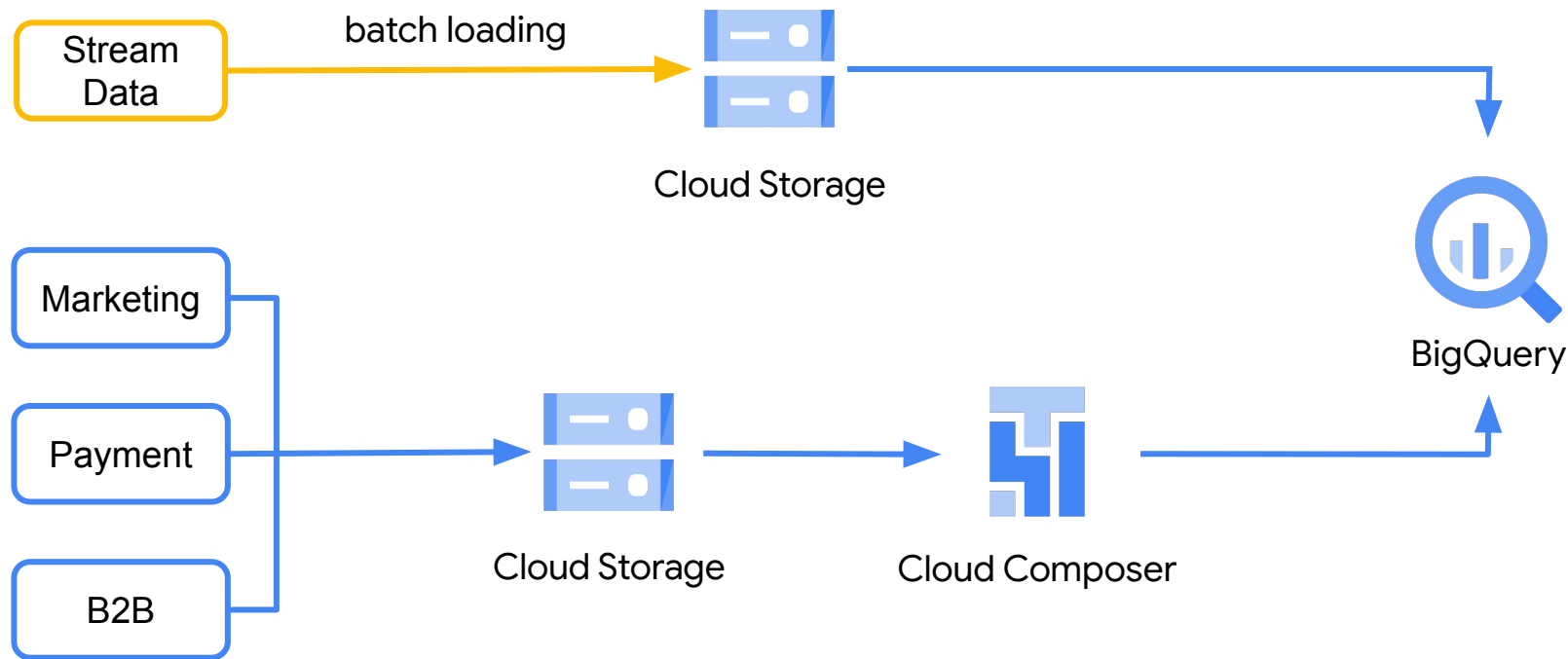


# データ分析基盤のアーキテクチャ



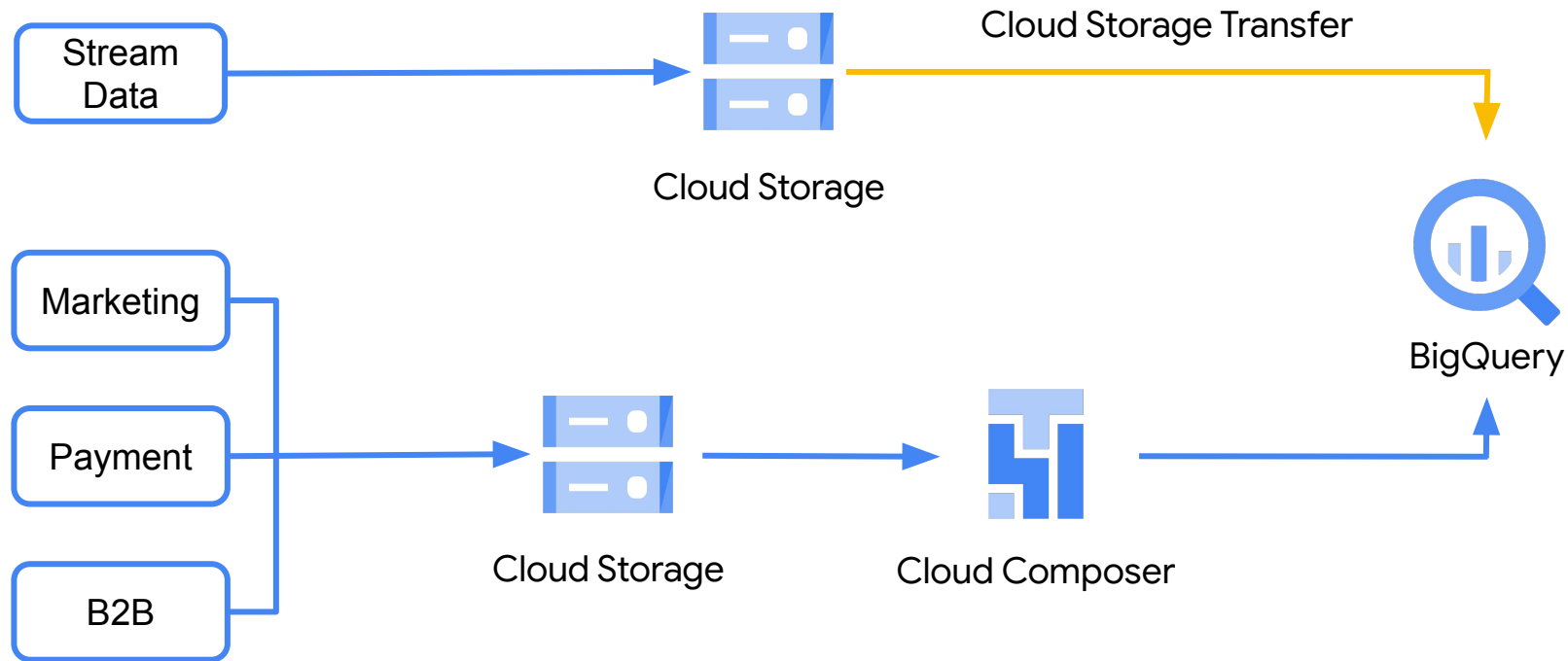
他部署からのデータ

# データ分析基盤のアーキテクチャ



他部署からのデータ

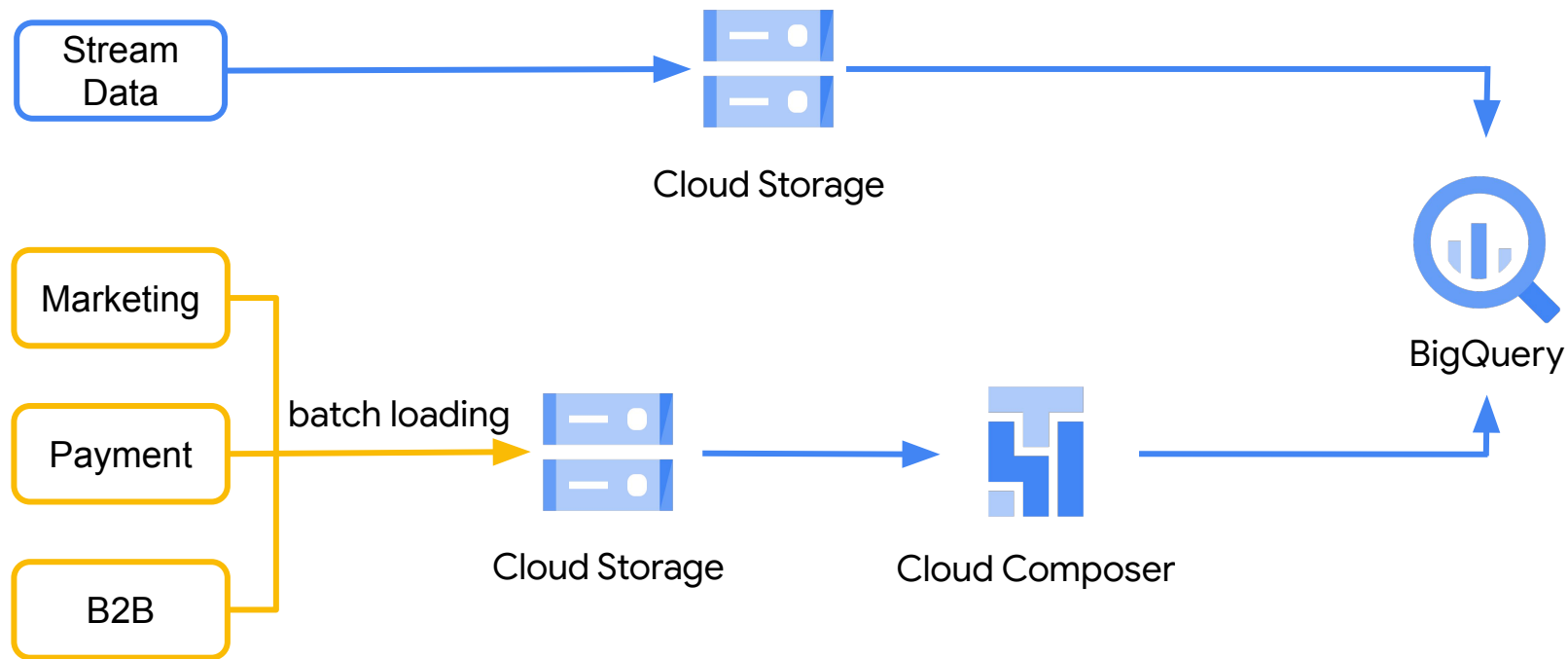
# データ分析基盤のアーキテクチャ



他部署からのデータ

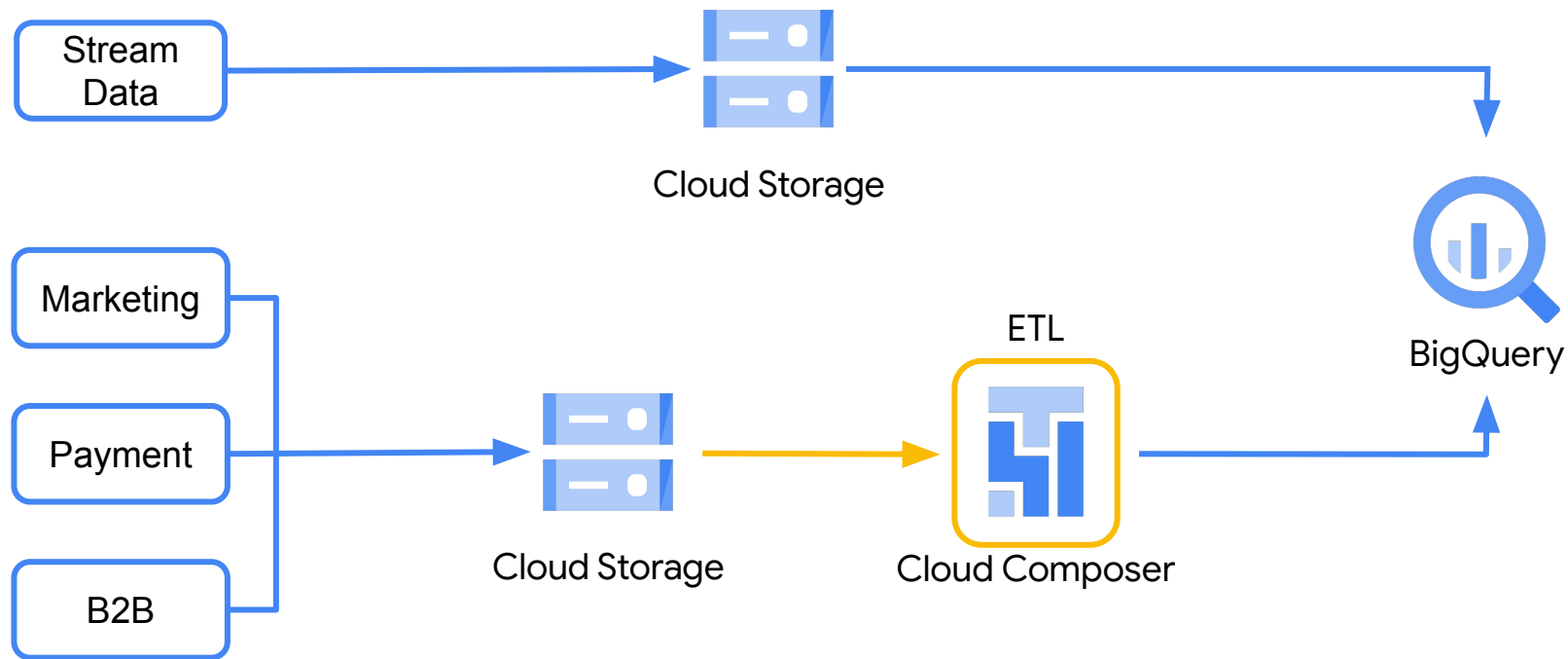


# データ分析基盤のアーキテクチャ



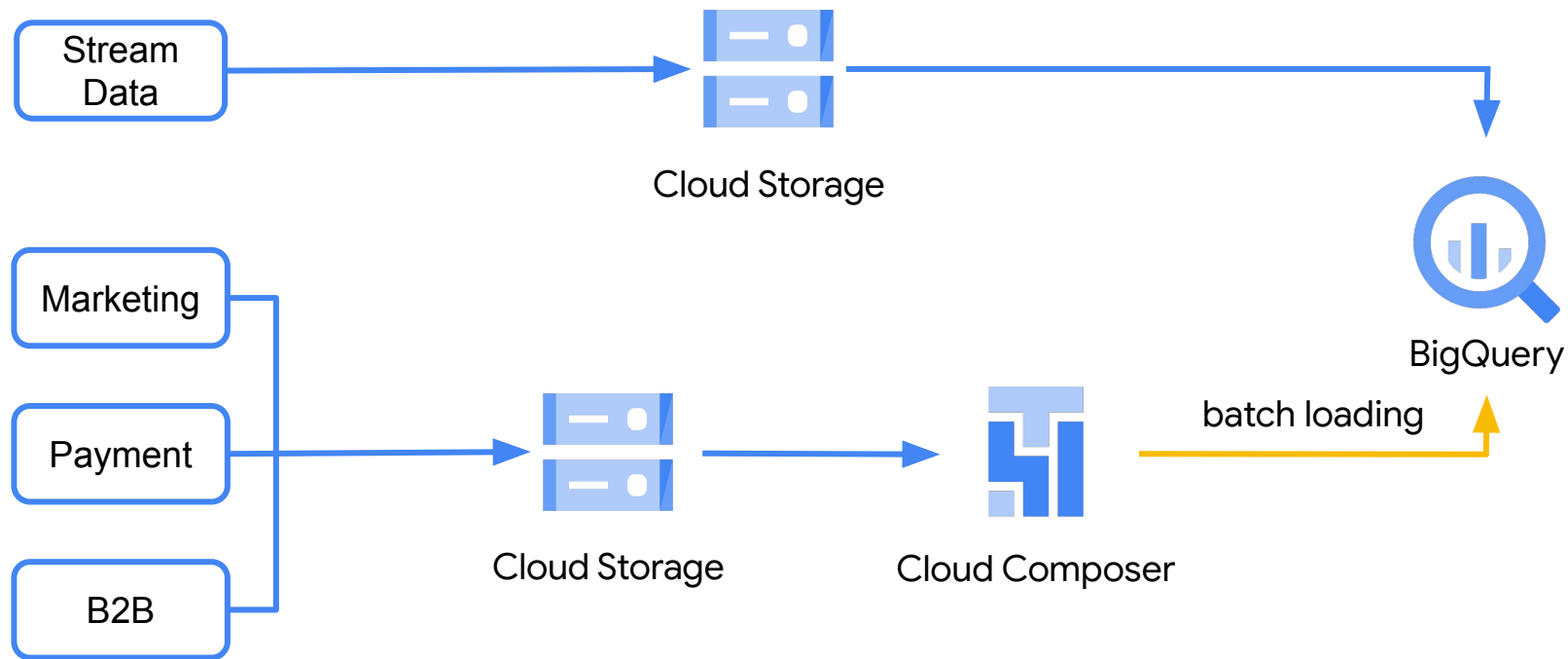
他部署からのデータ

# データ分析基盤のアーキテクチャ



他部署からのデータ

# データ分析基盤のアーキテクチャ



他部署からのデータ

# BigQuery の活用法

- 日経ではかなりのアドホック クエリが実行されるためコストの把握しやすい**定額料金モデル**で利用している
- また、夜間バッチなどの急激な利用の増加に対しては**Flex Slots** を利用している
  - 仕様としてコミット期間が 60 秒、その後はいつでもキャンセル可能
  - 秒単位の課金

# 待機ジョブをトリガーにした Flex Slots の割当

BigQuery



Cloud Composer

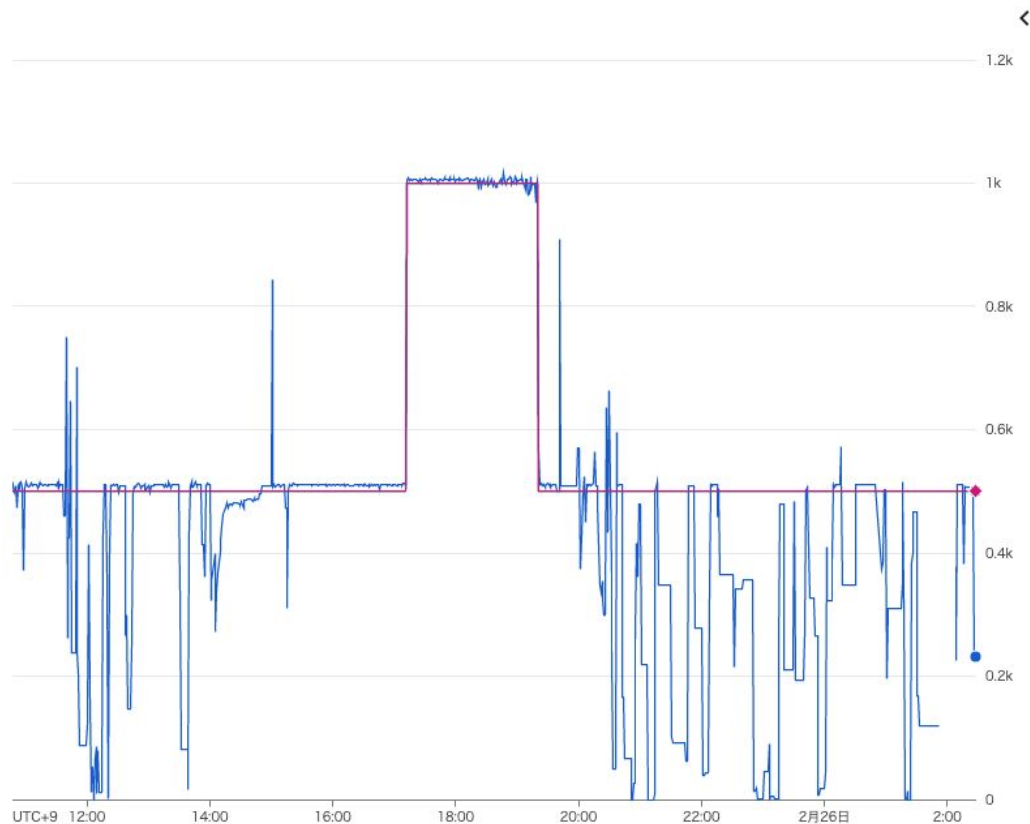


polling



Cloud Composer から BigQuery API を呼び出してジョブ一覧を取得。待機ジョブ数が一定の閾値を超えたら BigQuery Reservation の API を呼び出して Flex Slots を割り当てる。

# 実際に Flex Slots を割り当てたときの様子



# スケーラビリティの解決

- BigQuery のストレージ容量は無制限なため、データ量が増加するごとにノードの追加やデータの退避などの必要がなくなった
- Flex Slots の利用により急激な利用増加に対するスケールが容易になった



# 運用工数増大の解決



# terraform の活用

- Google Cloud 上のインフラはほぼすべて terraform を利用して構築している
- テーブル作成やカラム追加等の作業は  
テンプレート化された terraform モジュールに対してPull Request を送って貰う形で依頼する方式に変更

# テーブル作成の コードの例

terraform を書きなれて  
いない方でも Pull Request が  
出来るように抽象化。

```
variable "tables" {  
  type = map(map(string))  
  default = {  
    "example.table1" = {  
      description = "example table",  
      partition = "column1",  
      clustering = "column2,column3"  
    }  
  }  
}
```

# テーブル定義の例

テーブル定義はJSONで  
記述出来るようになって  
おり、フォーマットはBigQuery  
標準のものに  
則っている。

```
[
  {
    "name": "column1",
    "type": "DATETIME",
    "mode": "REQUIRED",
    "description": "example column 1",
  },
  {
    "name": "column2",
    "type": "STRING",
    "mode": "NULLABLE",
    "description": "example column 2",
  },
  {
    "name": "column3",
    "type": "INTEGER",
    "mode": "NULLABLE",
    "description": "example column 3",
  }
]
```

# Pull Request からテーブル作成までのフロー



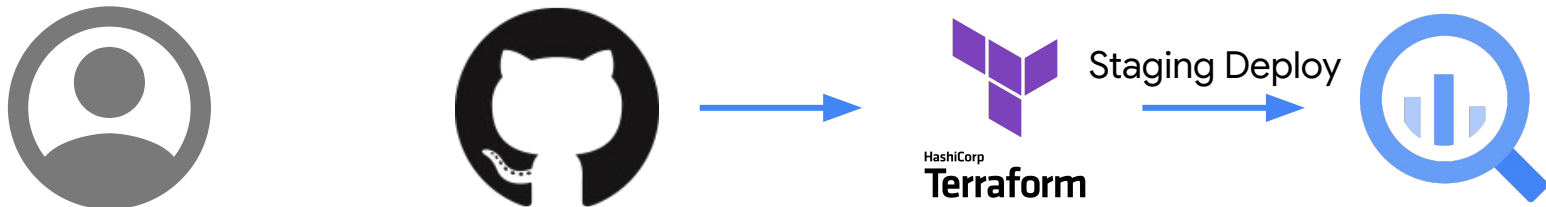
# Pull Request からテーブル作成までのフロー



Pull Request



# Pull Request からテーブル作成までのフロー



# Pull Request からテーブル作成までのフロー



Approved!



Production  
Deploy



# terraform によるアクセス制御

- BigQuery のデータセットやテーブルへのアクセス権限の付与もすべて terraform によって管理している
- テーブル作成等と同様にテンプレート化
- BigQuery の基本ロールでは必要以上の権限が付与され、不意にテーブルが変更されてしまう等の問題が発生したため現在ではカスタムロールを定義して利用者に付与している



# アクセス制御の コードの例

```
variable "bigquery_dataset_access_control" {  
  default = {  
    example_dataset = {  
      "user1@example.iam.gserviceaccount.com"  
    = "WRITER"  
    }  
  }  
}
```



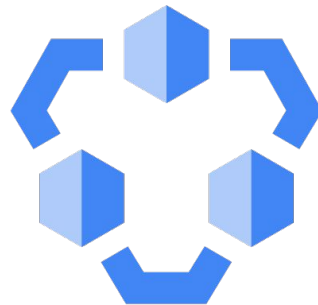
# Data Catalog による ドキュメンテーション

# Data Catalog とは



Data Catalog は、Google Cloud の  
データ分析プロダクト ファミリーに含まれる、  
フルマネージドのスケラブルな  
メタデータ管理サービスです。”

出典: <https://cloud.google.com/data-catalog/docs/concepts/overview>



# Data Catalog の活用

- 日経ではデータ分析基盤の利用者向けのドキュメント用の Web サイトを社内に公開している
- その中で Data Catalog を利用して BigQuery の テーブルやカラムの意味を検索できるページを作成

# Data Catalog によるドキュメンテーション

Atlas Document

## Data Catalog

BigQueryデータ検索

全てのリソース ▾ | 全てのキーワード ▾ | 関連度順 ▾

keyword Search

### SEARCH RESULT

#	Type	Name
1	table	<a href="#">test_ds01.t_latest</a>
2	table	<a href="#">test_ds01.test_tbl01</a>
3	table	<a href="#">test_ds01.test_tbl02</a>
4	table	<a href="#">test_ds01.test_tbl03</a>
5	table	<a href="#">test_ds01.test_tbl04</a>
6	table	<a href="#">test_ds01.test_tbl05</a>
7	dataset	<a href="#">test_ds01</a>
8	table	<a href="#">test_ds01.test_tbl06</a>
9	table	<a href="#">test_ds01.test_tbl07</a>
10	table	<a href="#">test_ds01.test_tbl08</a>

More

### DETAIL

Name	test_ds01.t_latest
Description	テストデータセット
Create time	2021/9/10 12:36:28
Update time	2021/12/7 21:17:55
Type	BIGQUERY_TABLE

### SCHEMA

[Download as JSON](#)

#	Column	Type	Mode	Description
1	col01	STRING	NULLABLE	テスト列01
2	col02	STRING	NULLABLE	テスト列02
3	col03	STRING	NULLABLE	テスト列03



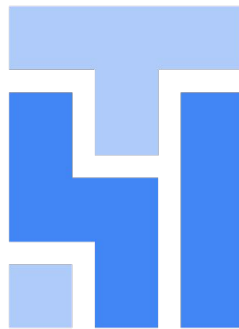
# Cloud Composer を利用した ジョブの運用

# Cloud Composer とは



Cloud Composer は、フルマネージドのワークフロー オーケストレーションサービスです。クラウドとオンプレミスデータセンターにまたがるワークフローの作成、スケジューリング、モニタリング、管理ができます。”

出典: <https://cloud.google.com/composer/docs/concepts/overview>



# Cloud Composer の活用

- DWH を BigQuery に移行する際に  
ジョブ スケジューラも Cloud Composer に移行
- 以前は様々なジョブが別々のスケジューラで  
動いていたためジョブの依存関係の把握が困難だった
- DAG をそのまま利用者に書かせるのはハードルが  
高いので頻繁に利用されるケースはテンプレート化



# ジョブのコードの例

select 結果を特定の S3  
バケットに転送するジョブ(SQL  
は別途必要)

```
variable "bigquery_select_to_s3" {  
  type = map(object(...))  
  default = {  
    "example_job" = {  
      schedule_interval = "timedelta(minutes=30)",  
      start_date        = "datetime(2022,4,19,0,0,0)",  
      retries            = 3,  
      retry_delay        = "timedelta(minutes=5)",  
      email              = "xxx@example.slack.com",  
      aws_conn_id        = "s3_example",  
      dest_s3_prefix     = "s3://example_bucket/file_{{ ts_nodash }}_",  
      replace            = "True",  
      format              = "CSV",  
      plus_extention     = "",  
      compression        = "",  
      option = {  
        "header"        = "true",  
        "field_delimiter" = ",",  
      }  
    }  
  }  
}
```

# まとめ

- 全社に DX を推進していく中での課題
  - 少人数のエンジニアでもスケール出来る運用体制の確保
  - 利用が拡大してもスケールできる DWH
- BigQuery、Cloud Composer と terraform の活用
  - 多くの人に利用してもらうためにリソースが  
作りやすくする仕組み作り

# We're Hiring!

- データエンジニアを絶賛募集中です
- 興味がある方は、
  - Twitter (@nikkeideveloper) に DM
  - 技術ブログ・採用サイト: <https://hack.nikkei.com>
- カジュアル面談大歓迎
  - <https://herp.careers/v1/nikkei>

# Thank you.

