データエンジニアリング研修
Day 1

技術部データ基盤チーム 小松ももか

2022.07.29

GMOペパボ

自己紹介

小松 ももか @UsaMomokawa

2020年新卒入社
データエンジニア
技術部 データ基盤チーム（2021/11〜）

1日目のゴール

目標

- 自サービスで ◯◯ を分析するために Bigfoot を使う方法が分かる

演習

- データを目的に沿って変換する
- データをData Studioで可視化する

2日目のゴール

目標

- データのELT(Extract, Load, Transform)する方法を学ぶ

演習

- BigQueryにデータをロードする

データエンジニアリングって
なんですか？

データエンジニアリングってなんですか？

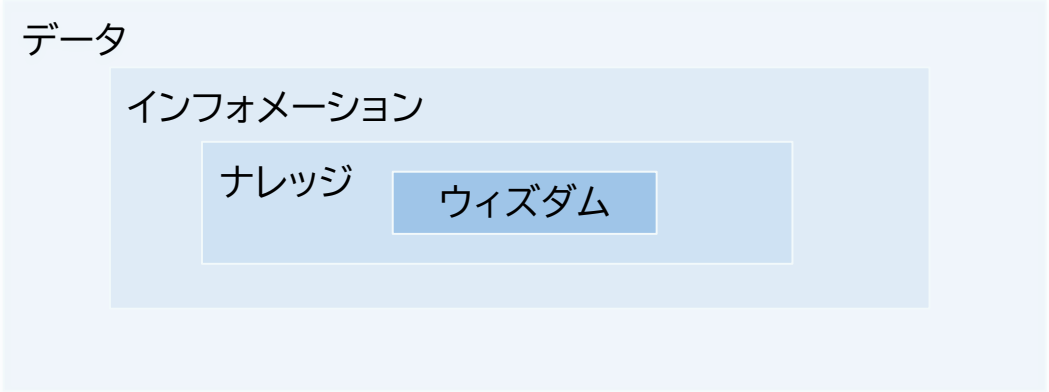
データ基盤を構築、管理する技術領域全般を
データエンジニアリングと呼びます。

データエンジニアリングを駆使し、ビジネスに価
値をもたらすのがデータエンジニアと呼ばれる
比較的新しい専門職です。

下田 倫大,寶野 雄太,饗庭 秀一郎,吉田 啓二 (2021)
「Google Cloud ではじめる実践データエンジニアリング入門」技術評論社,
Kindle版 - はじめに, 8段落目より引用

データとは

- データ ⊃ インフォメーション
  - データは「インフォメーションの原材料」
  - インフォメーションは「コンテキストを持ったデータ」



DAMA International（2018）「データマネジメント知識体系ガイド 第二版」日経BP社 P.43より引用

```
date-id-name-num0-num1-num2-num3
2021-07-15-0123456789-test0-12-34-567-890
2021-07-15-9876543210-test1-98-76-543-210
```

```
date-id-name-num0-num1-num2-num3
2021-07-15-0123456789-test0-12-34-567-890
2021-07-15-9876543210-test1-98-76-543-210
```

```
date,id,name,num0,num1,num2,num3
2021-07-15,0123456789,test0,12,34,567,890
2021-07-15,9876543210,test1,98,76,543,210
```

手段1: インフォメーションになりうるデータをつくる

- そもそもデータを集めるのはなぜか？
  - 何かを知りたいから
  - 適切な理解は適切なデータ作成から
- データを作るにはナレッジ（ドメイン知識）が必要
  - 目的に沿ったデータを作ることが大事
  - システムだけでなく**人もデータを作る**
    - スプレッドシート
    - Notion etc…

手段2: 既存のデータをインフォメーションに変換する

- 誰もが分かる表現に加工する
  - 例）0～6 で曜日を表現している
- 別のデータとつなぎ合わせる
  - 例）ユーザー情報 * 注文情報
- メタデータ（データを説明するデータ）を追加する
  - 例）5W1H

GMOペパボ

データに基づいた判断を行う段階

1	顧客接点のデジタル化
2	事業活動データの収集
3	データ蓄積・分析基盤
4	データ処理パイプライン
5	データ可視化とリテラシー
6	機械学習プロジェクト管理
7	マーケティング自動化
8	自動的な意思決定

1: 収集

データが出力され、取りまとめられている段階

2: 分析

取りまとめたデータを可視化、一元的に分析できる段階

3: 活用

データにより継続的なサービス改善を行える段階

＊1～8は、日本CTO協会が示している企業のデジタル化とソフトウェア活用のためのガイドライン「DX Criteria」のデータ駆動に関するカテゴリから引用
https://dxcriteria.cto-a.org/d44c9631ebfb4c8f8ff1ae69361dfa65

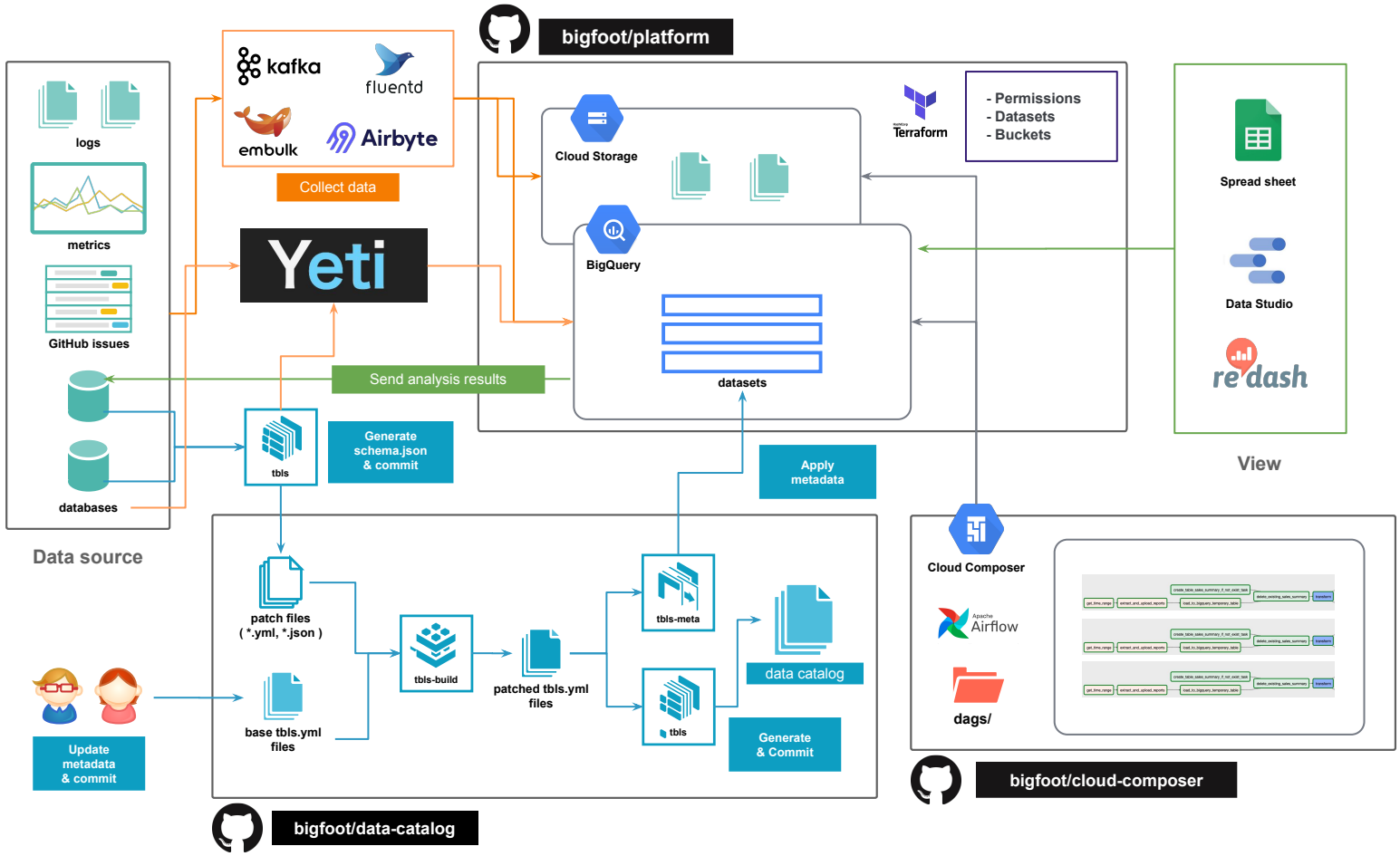
GMOペパボ

各段階で必要になる知識・技術

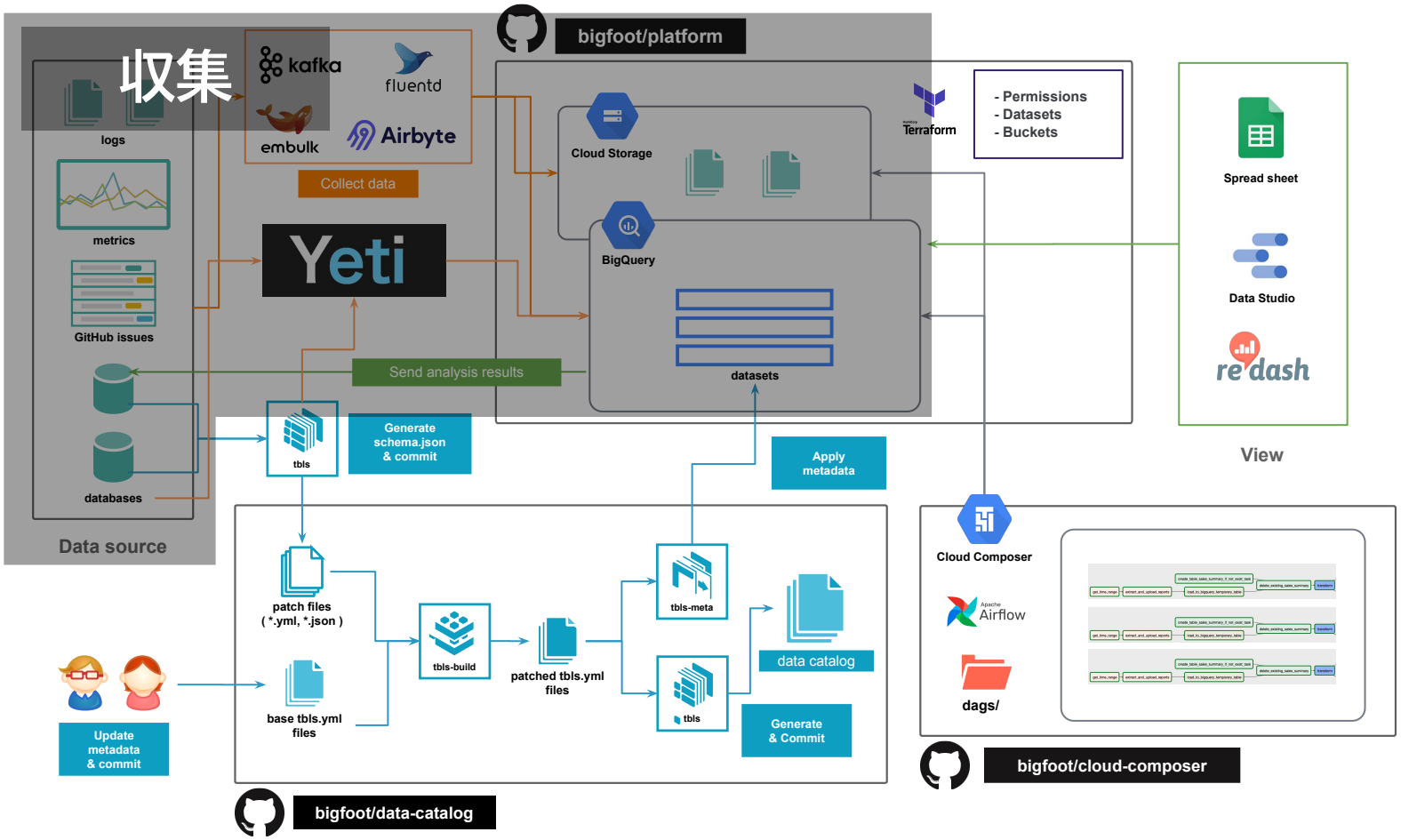
	段階	システム（実行のために仕組みとして必要なもの）	＋	リテラシ（データを解釈するために必要なもの）	＝	データ駆動
1 顧客接点のデジタル化	1: 収集 データが出力され、取りまとめられている段階	データウェアハウス Logger				
2 事業活動データの収集						
3 データ蓄積・分析基盤						
4 データ処理パイプライン	2: 分析 取りまとめたデータを可視化、一元的に分析できる段階	BI / Dashboard ワークフロー データ連携		データ集計 統計知識 事業価値の理解	→	統計的な判断
5 データ可視化とリテラシー						
6 機械学習プロジェクト管理	3: 活用 データにより継続的なサービス改善を行える段階	機械学習基盤 適応的改善機構		情報推薦 機械学習	→	サービスの動的改善 自動的な意思決定
7 マーケティング自動化						
8 自動的な意思決定						

Bigfoot 入門

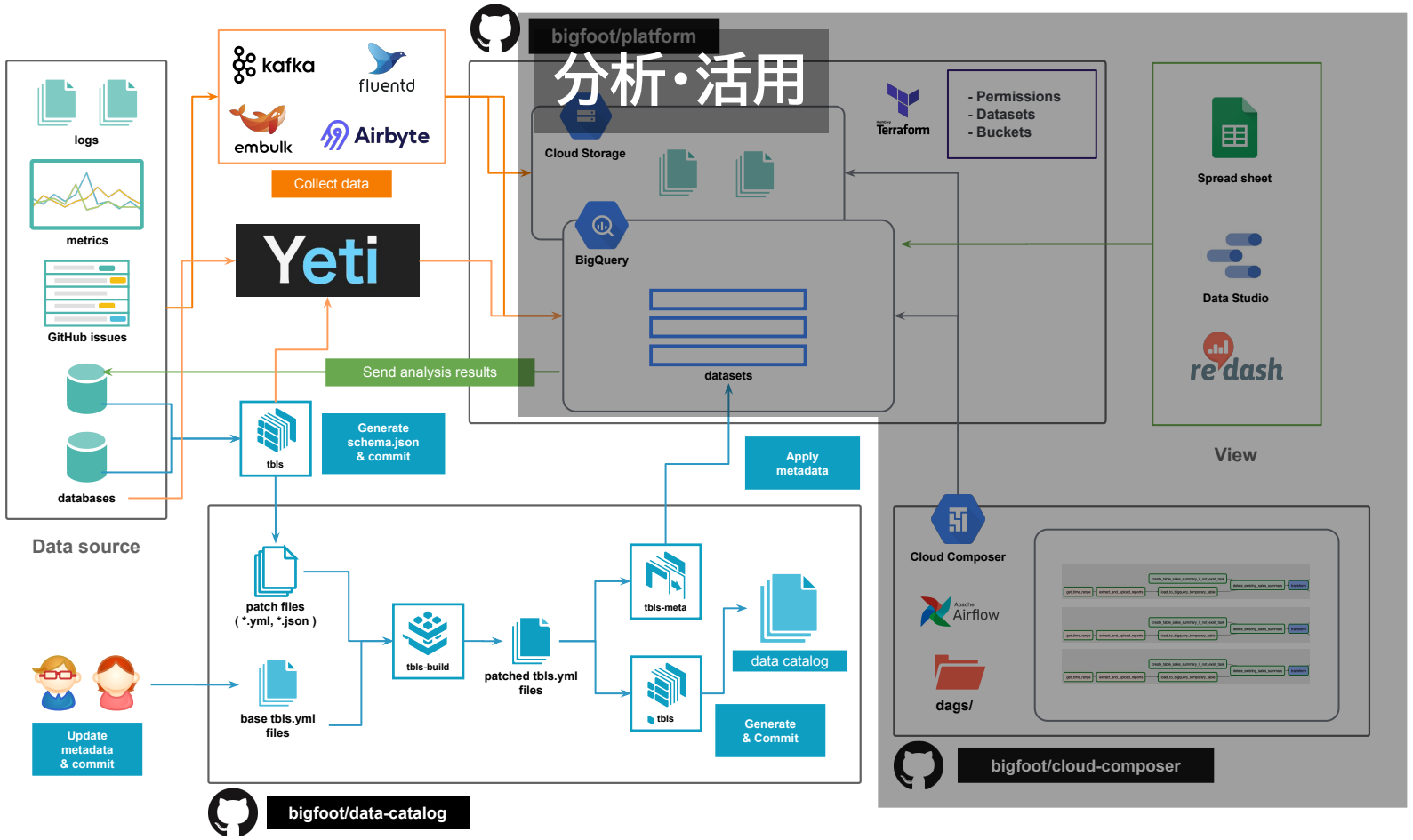
Bigfoot 概略図


bigfoot/platform
kafka
fluentd
embulk
Airbyte
Collect data
logs
metrics
GitHub issues
databases
Data source
Yeti
Cloud Storage
BigQuery
datasets
Terraform
- Permissions
- Datasets
- Buckets
Send analysis results
tbls
Generate schema.json & commit
Apply metadata
Spread sheet
Data Studio
re'dash
View
patch files (*.yml, *.json)
base tbls.yml files
Update metadata & commit
tbls-build
patched tbls.yml files
tbls-meta
tbls
data catalog
Generate & Commit
bigfoot/data-catalog
Cloud Composer
Apache Airflow
dags/
bigfoot/cloud-composer

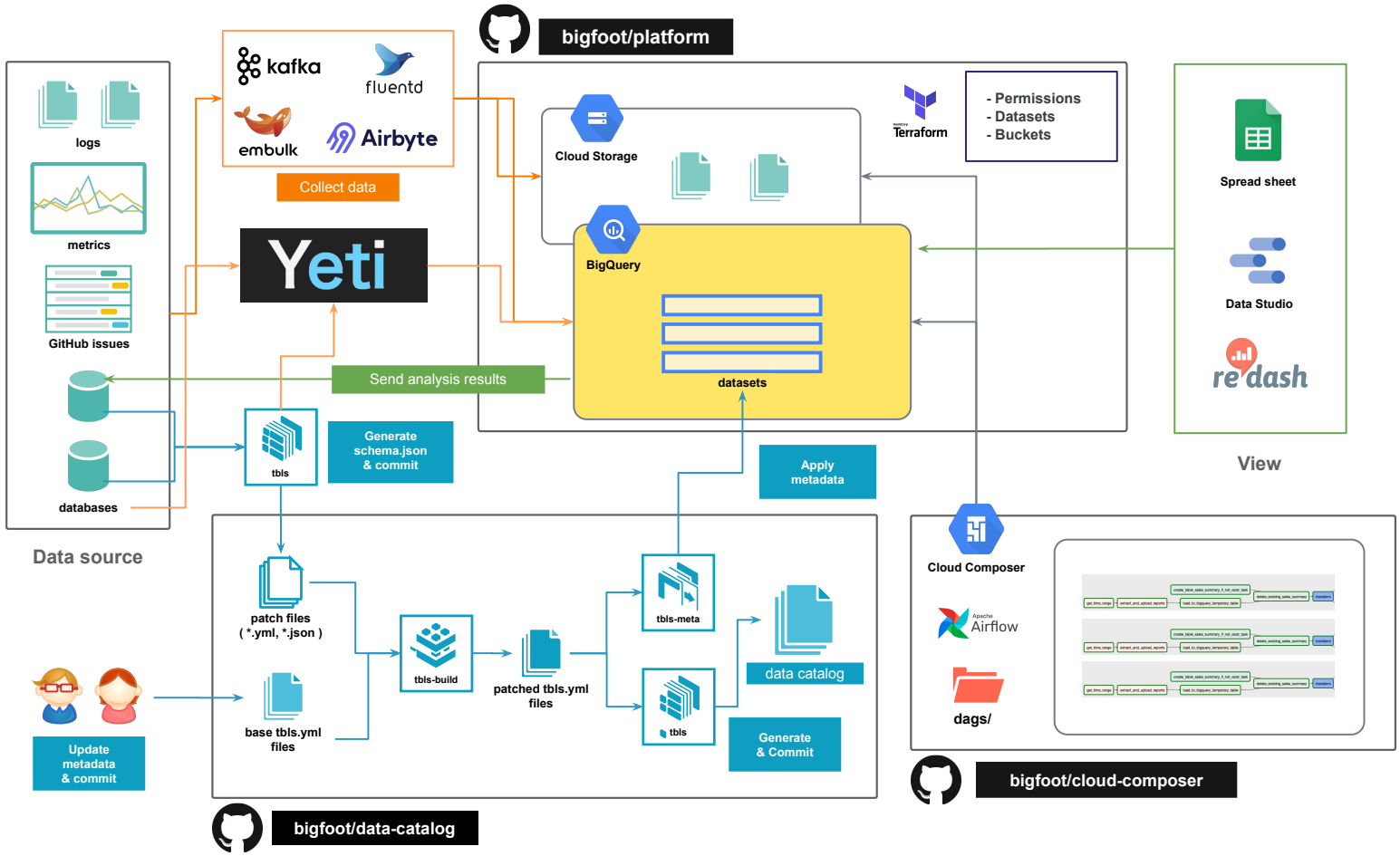
Bigfoot 概略図


収集
kafka
fluentd
embulk
Airbyte
Collect data
logs
metrics
GitHub issues
databases
Data source
Yeti
bigfoot/platform
Cloud Storage
BigQuery
datasets
Terraform
- Permissions
- Datasets
- Buckets
Spread sheet
Data Studio
re'dash
View
Send analysis results
tbls
Generate schema.json & commit
Apply metadata
patch files (*.yml, *.json)
base tbls.yml files
tbls-build
patched tbls.yml files
tbls-meta
tbls
data catalog
Generate & Commit
Update metadata & commit
bigfoot/data-catalog
Cloud Composer
Apache Airflow
dags/
bigfoot/cloud-composer

Bigfoot 概略図


bigfoot/platform
分析・活用
kafka
fluentd
embulk
Airbyte
Collect data
logs
metrics
GitHub issues
databases
Data source
Yeti
Send analysis results
tbls
Generate schema.json & commit
Cloud Storage
BigQuery
datasets
Terraform
- Permissions
- Datasets
- Buckets
Spread sheet
Data Studio
re'dash
View
Apply metadata
Cloud Composer
Apache Airflow
dags/
bigfoot/cloud-composer
patch files (*.yml, *.json)
base tbls.yml files
tbls-build
patched tbls.yml files
tbls-meta
tbls
data catalog
Generate & Commit
Update metadata & commit
bigfoot/data-catalog

Bigfoot 概略図


bigfoot/platform
kafka
fluentd
embulk
Airbyte
Collect data
logs
metrics
GitHub issues
databases
Data source
Yeti
Cloud Storage
BigQuery
datasets
Terraform
- Permissions
- Datasets
- Buckets
Spread sheet
Data Studio
re'dash
View
Send analysis results
tbls
Generate schema.json & commit
Apply metadata
patch files (*.yml, *.json)
base tbls.yml files
tbls-build
patched tbls.yml files
tbls-meta
tbls
data catalog
Generate & Commit
Update metadata & commit
bigfoot/data-catalog
Cloud Composer
Apache Airflow
dags/
bigfoot/cloud-composer

BigQuery 入門

GMOペパボ

Google Cloud BigQuery

- 大規模データ分析対応のフルマネージド型データウェアハウス
- 一般的なデータベースと同様に標準SQLが利用可能
- コンピュート、ストレージ、メモリが分離され、それぞれ独立してスケーリングできる
  - https://cloud.google.com/blog/products/bigquery/bigquery-under-the-hood
  - データの保存量が事実上無制限
  - クエリ実行速度が超速
- Bigfoot の BigQuery は定額なので、気軽にクエリを投げてください

GMOペパボ

Google BigQuery: インターフェース

- Google Data Studio
- Google Sheets
- BigQuery Web Console
- Google Apps Script
- Google Colaboratory
- bq command-line tool
- API Client Libraries

  C#, Go, Java, Node.js, PHP,

  Python, Ruby
- REST API

Google BigQuery: インターフェース

- Google Data Studio
- BigQuery Web Console
- Google Sheets
- bq command-line tool

1日目の演習課題で扱います

2日目の演習課題で扱います

データウェアハウスとリレーショナルデータベースの違い

- BigQuery などのデータウェアハウスは**列指向**が多い
  - 特定列に対する集計処理が得意
    - 例）1億行10列のテーブルで列Aの平均を算出 → 列ごとにデータを保存しているので列Aのみ走査
- MySQL, PostgreSQLなどのリレーショナルデータベースは**行指向**
  - 特定行に対する操作が得意
    - 行の特定を高速に行う仕組み（インデックス）がある
    - 構造的に全列を走査するので列方向の集計処理はリソース効率的に向いていない

BigQueryは大規模なデータを効率よく処理できる。

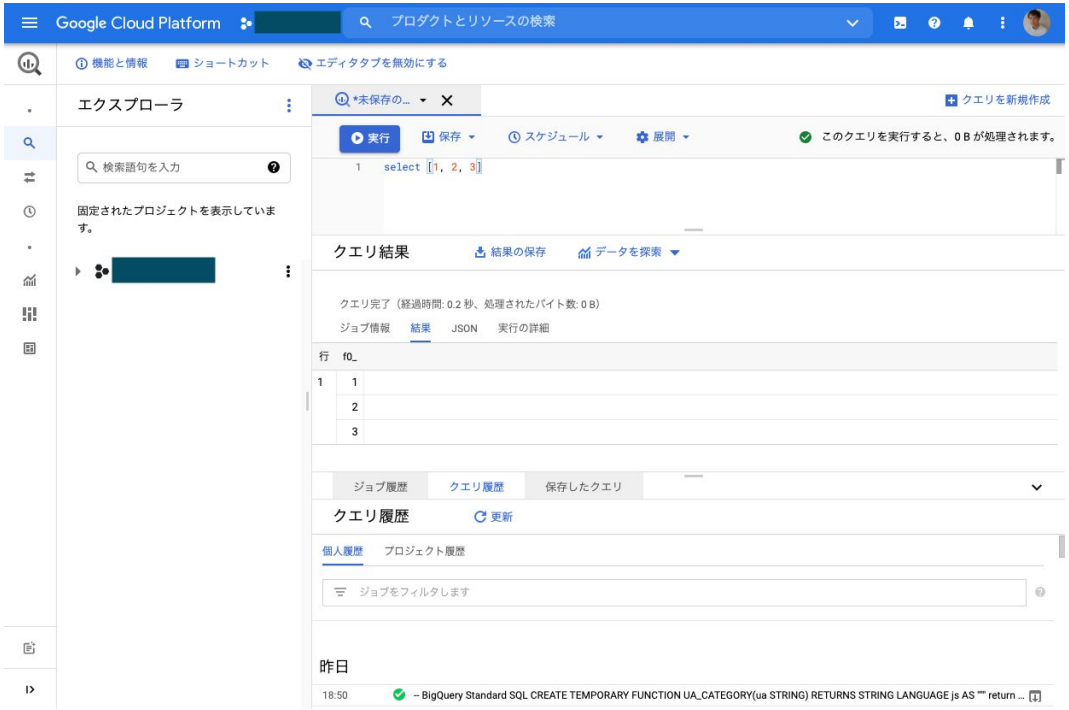
一方で、行単位の更新や削除が頻繁に行われるデータを格納するのは苦手。

BigQueryでクエリを叩いてみよう

https://console.cloud.google.com/bigquery

エクスプローラ

プロジェクト
└ データセット
　└ テーブル
　　ビュー



タブ領域

- クエリエディタ
- クエリスキャン見積
- データセット情報参照
- テーブル情報参照

クエリ結果

- クエリ実行結果の表示
- クエリ実行結果の保存
- ジョブ実行情報

プロジェクト

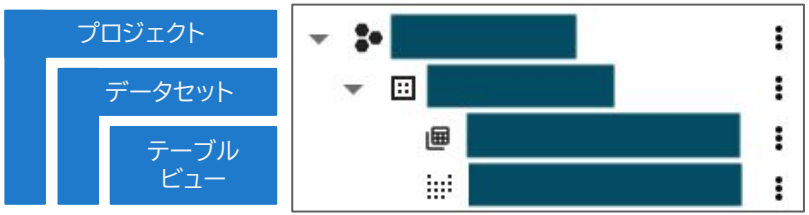
GCPのリソースやコストの管理をするためのグループ

データセット

テーブルやビューをグループ化する概念

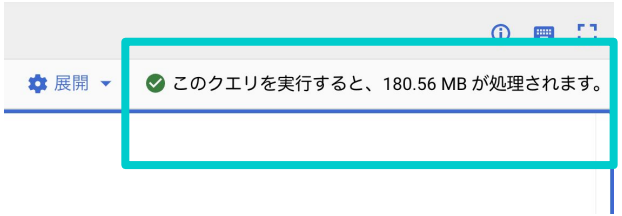
Bigfootでは、サービス単位で作成している

データセットでアクセス権限を管理

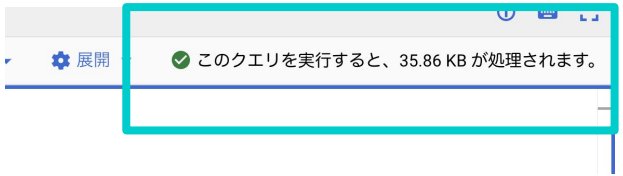


GMOペパボ

スキャン量の違いを見てみよう




```
SELECT
    *
FROM
    training.github_coverages
;
```




```
SELECT
    repo
FROM
    training.github_coverages
;
```

時間が余ったら、WHERE や LIMIT をかけてスキャン量を見てみよう

GMOペパボ

スキャン量の違いを見てみよう

- 全ての列を指定すると
  表全体をスキャン

```
SELECT
    *
FROM
    training.github_coverages
;
```

- 特定の列を指定すると
  その列のみをスキャン

```
SELECT
    repo
FROM
    training.github_coverages
;
```

時間が余ったら、WHERE や LIMIT をかけてスキャン量を見てみよう

BigQuery 演習

- (1-1) GitHub Enterprise Server (GHES)のリポジトリ単位で、
  日別のコードカバレッジを集計してください
  - コードカバレッジ は「コードカバレッジで対象となるコードの行数に対する、テストによってカバーできている行数の割合(%)」

- BigQueryのリファレンスはこちら
  - https://cloud.google.com/bigquery/docs/reference/libraries-overview

NOTE:
演習では、octocov で収集したコードメトリクスを利用しています
https://github.com/k1LoW/octocov

BigQuery 演習

- （1-2）1-1 の集計結果を新しいテーブルに書き込んでください
  - SQLを使ってテーブルを作成し書き込んでください

回答例

- （1-1）GHESのリポジトリ単位で、日別のコードカバレッジを集計してください

考え方: 同リポジトリで同日に複数レコードが登録されているので、**最新のレコードを使いたい**

```
SELECT
    owner,
    repo,
    coverage_covered,
    coverage_total,
    TIMESTAMP_TRUNC(timestamp, DAY, 'Asia/Tokyo') AS _date,
FROM
    training.github_coverages
ORDER BY
  _date DESC
;
```

GMOペパボ

1-1: 回答例1


```
SELECT
  *
FROM (
  SELECT
    owner,
    repo,
    CONCAT(owner, '/', repo) AS owner_repo,
    coverage_covered,
    coverage_total,
    timestamp,
    TIMESTAMP_TRUNC(timestamp, DAY, "Asia/Tokyo") AS _date,
    IF(coverage_total > 0, ROUND(100 * coverage_covered / coverage_total, 1), 0) AS coverage_percentage,
  FROM
    training.github_coverages
)
QUALIFY
  ROW_NUMBER() OVER (PARTITION BY owner, repo, _date ORDER BY timestamp DESC) = 1
;
```

ROW NUMBER() で絞る方法

GMOペパボ

1-1: 回答例2

経過時間	消費したスロット時間	シャッフルされたバイト数
350 ミリ秒	175 ミリ秒	393.15 KB

```
WITH coverages AS (
  SELECT
  coverage.*,
  _date,
FROM (
  SELECT
    owner,
    repo,
    TIMESTAMP_TRUNC(timestamp, DAY, 'Asia/Tokyo') AS _date,
    ARRAY_AGG(t ORDER BY t.timestamp DESC LIMIT 1)[offset(0)] AS coverage
  FROM
    training.github_coverages AS t
  GROUP BY
    owner,
    repo,
    _date
  )
)

SELECT
  CONCAT(owner, '/', repo) AS owner_repo,
  _date,
  coverage_covered,
  coverage_total,
  IF(coverage_total > 0, ROUND(100 * coverage_covered / coverage_total, 1), 0) AS coverage_percentage,
FROM
  coverages
;
```

ARRAY_AGG() で絞る方法が推奨されています
https://cloud.google.com/bigquery/docs/best-practices-performance-compute

- （1-2）1-1 の集計結果を新しいテーブルに書き込んでください

```
CREATE TABLE training_test.github_coverages AS
...
```

Data Studio 入門

Data Studio

- Google マーケティングプラットフォーム で提供されているBIサービス
- Google アナリティクスも、Google マーケティングプラットフォームのサービスのひとつ
- 日本では「データポータル」と呼ばれていますが、ここでの表記は Data Studio に統一します。

どうやってアクセスするの？

こちらへ

https://datastudio.google.com/

GMOペパボ

使い方

公式のヘルプページ の解説がわかりやすいです。

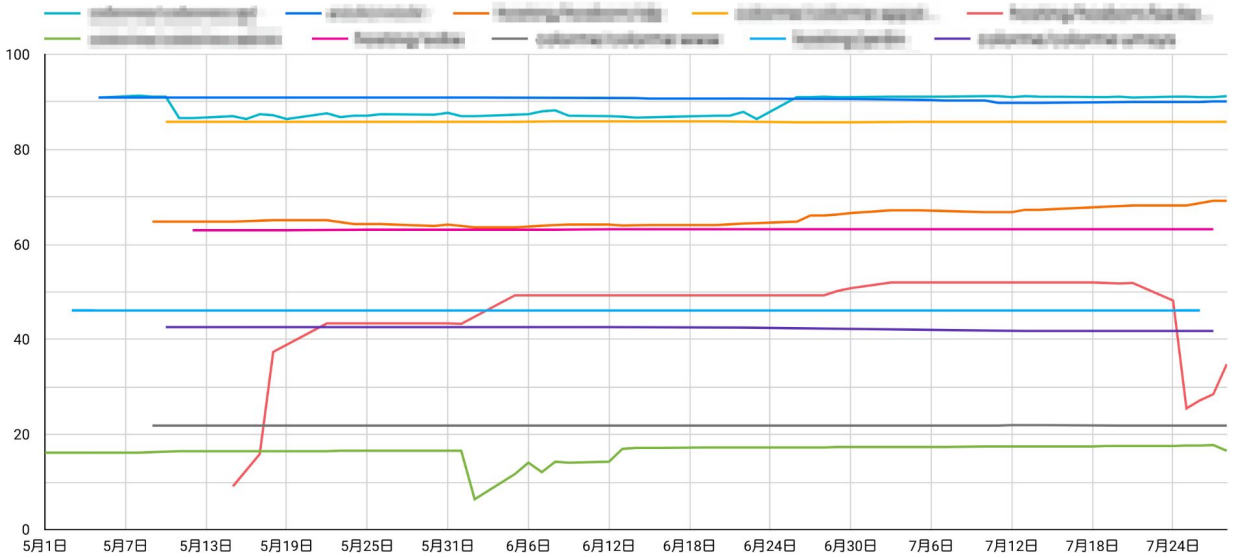
Data Studio のホーム画面に表示されているチュートリアルもおすすめです。

Data Studio 演習

- （1-3）リポジトリ単位で日別のコードカバレッジを出し、**過去3ヶ月分**のコードカバレッジの**時系列推移**を表示してみましょう
  - データソースは、1-2 の課題で作成したテーブルを指定してください
  - グラフの横軸を日付、縦軸をコードカバレッジ(%)にしてください
  - リポジトリ別に色分けして表示するには...？ => 内訳ディメンション
  - スタイル > 全般 > 線形補完 で線をなだらかにできます
  - フィルタ機能を使ってデータを絞りましょう

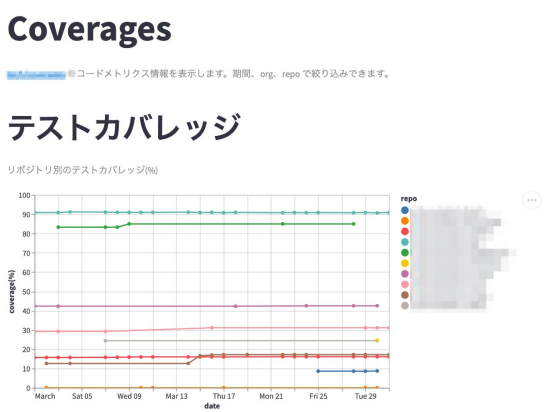
回答例

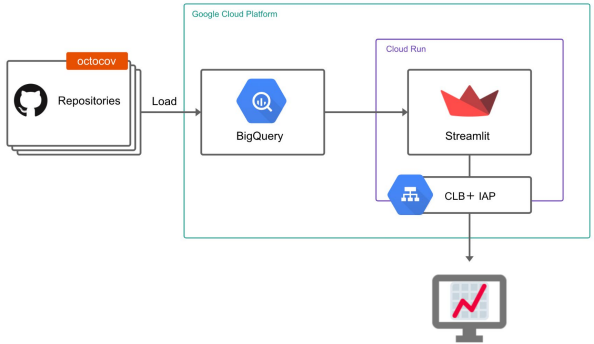
テストカバレッジ


100
80
60
40
20
0
5月1日
5月7日
5月13日
5月19日
5月25日
5月31日
6月6日
6月12日
6月18日
6月24日
6月30日
7月6日
7月12日
7月18日
7月24日

可視化の事例紹介: Verne

- Verne(ヴェルヌ)
- Dashboard as Code を実現しています





https://tech.pepabo.com/2022/04/25/code-metrics-dashboard/

参考文献

DAMA International (2018)『データマネジメント知識体系ガイド 第二版』日経BP社

ゆずたそ, はせりょ(2020)『データマネジメントが30分でわかる本』

ゆずたそ, 渡部徹太郎, 伊藤徹郎『実践的データ基盤への処方箋』技術評論社

下田 倫大, 寶野 雄太, 饗庭 秀一郎, 吉田 啓二 (2021)『Google Cloud ではじめる実践データエンジニアリング入門』技術評論社