



2024年3月5日

S3とQuickSightで作る BIシステム構築の勘所



partner
network



自己紹介

□ 隅田 徹 (Toru Sumida)

□ 株式会社スタイルズ

- ▶ AWSを主としたアプリ・インフラ混合チーム所属
- ▶ BIシステムの設計・開発
- ▶ IoTサービスの設計・開発

□ 好きなAWSサービス

- ▶ Amazon QuickSight
- ▶ Amazon Athena



本日、お話する内容

□対象

- ▶ S3とQuickSightでBIシステムの構築を検討している方
- ▶ 既にQuickSightでBIシステムを構築されている方

□ゴール

- ▶ S3とQuickSightでBIシステムを構築する際の勘所を知る

データを大量に貯めているが生かせていない
どうやって活用したらいい？



BIシステムを導入する事が解決策の一つです



BIシステムのBIとは

BIとは

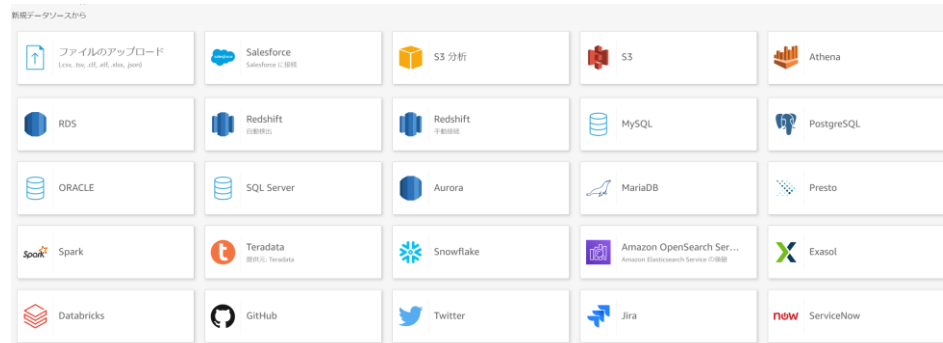
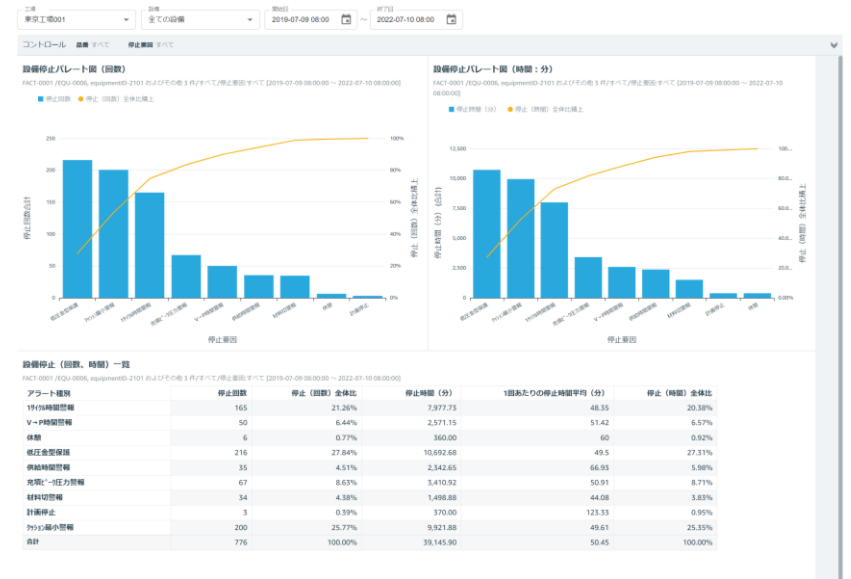
- ビジネスインテリジェンスの略
- データに基づいた意思決定プロセスを支援すること
- BIツール
 - ▶ 各種データソース（企業データ）との接続・蓄積
 - ▶ データの可視化・オンライン分析
 - ▶ ダッシュボード
 - ▶ など
- AWSのBIツール Amazon QuickSight



Amazon QuickSightの特徴

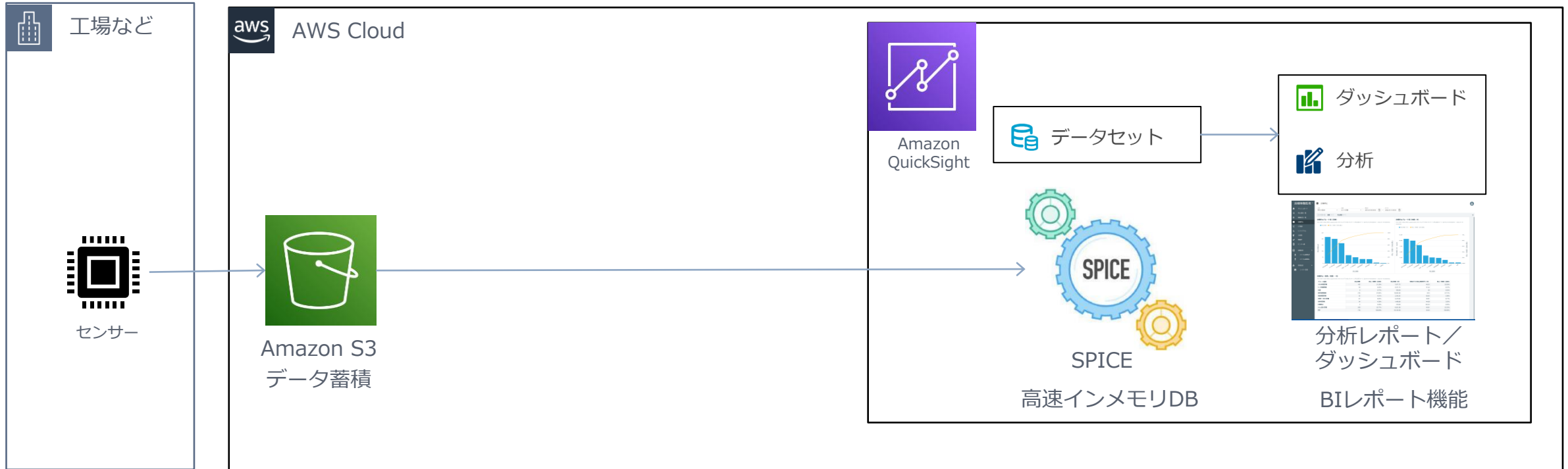


- 美しくインタラクティブなダッシュボード
- サーバレスアーキテクチャ
- 高速インメモリDB SPICEを内蔵
- 多様なデータソースへの接続
- 閲覧ユーザの利用料が従量



システム構成パターン

最も単純な構成



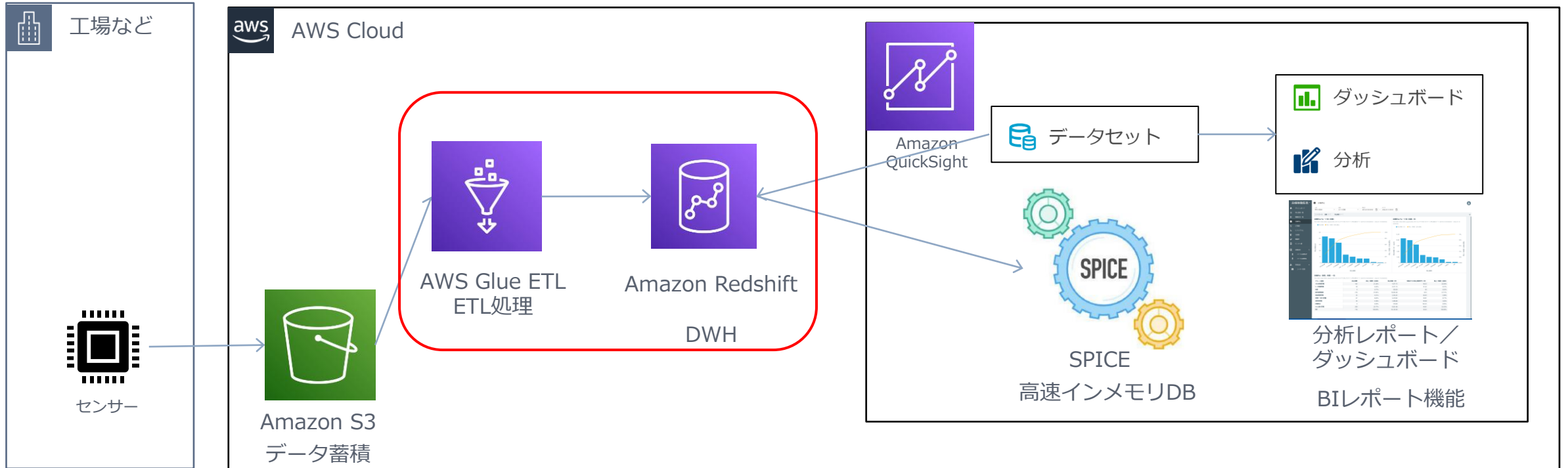
メリット

- 利用するサービスが最小限で構築が簡易

デメリット

- データ追加時は常に全件更新
- 取込時にデータ項目の取捨選択ができない

ETL・DWH構成



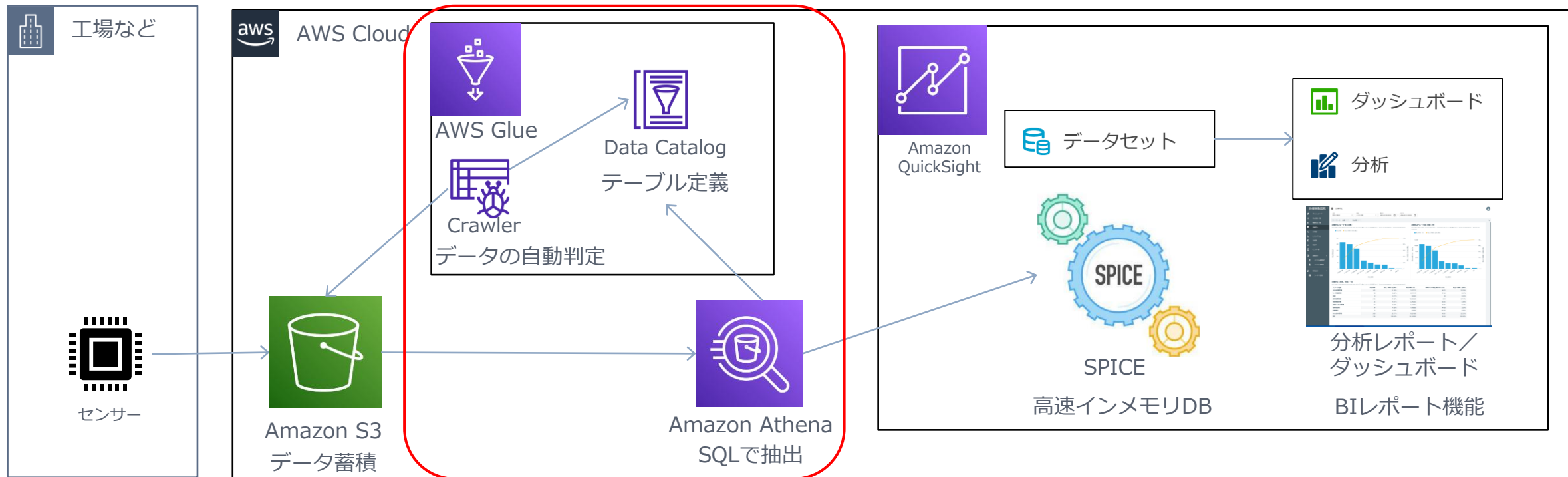
メリット

- リアルタイム分析が可能
- 複雑なデータ処理にも対応できる

デメリット

- コストが比較的高くなる
- 構築期間が比較的にかかる

バランス構成



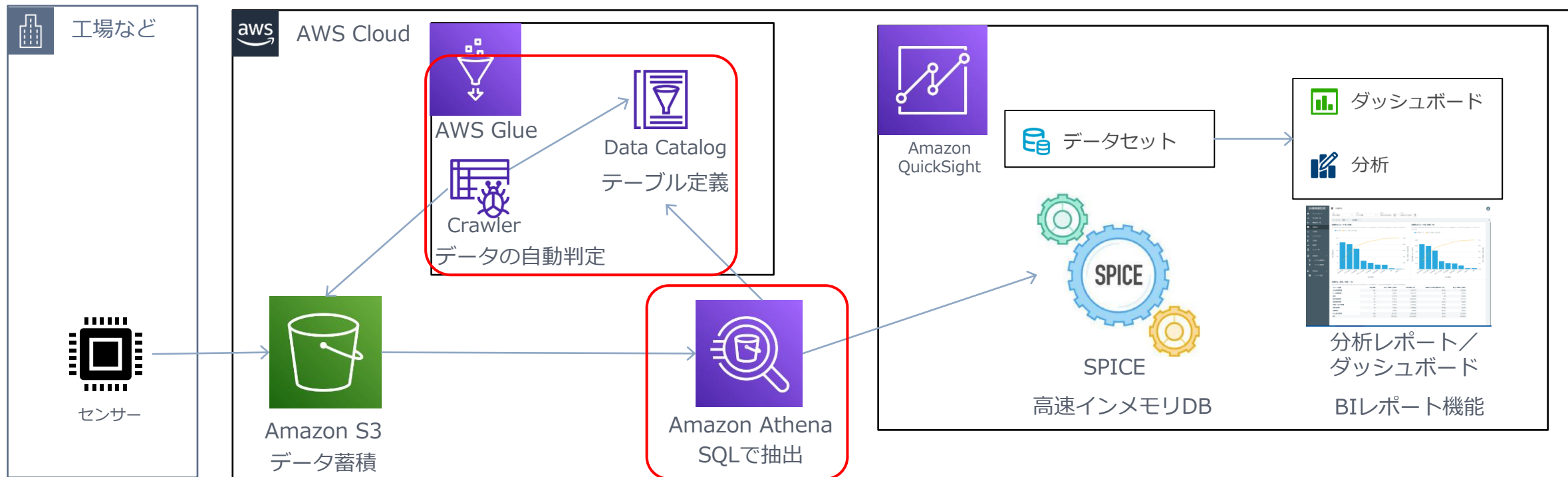
メリット

- DWH構成より手軽に構築できる
- パーティショニングで抽出コストを削減
- 複雑な処理が必要になったタイミングで Glue ETLを導入することでカバーできる

デメリット

- 直接クエリ（リアルタイム参照）には向いていない

バランス構成 Glue、Athenaの役割



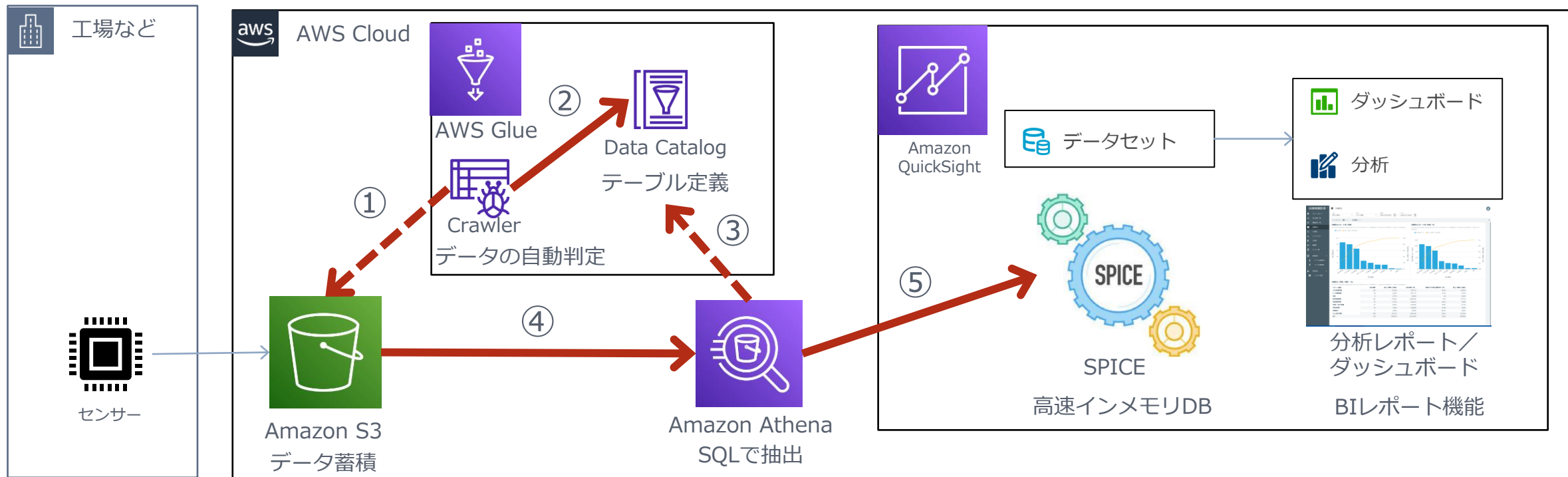
メリット

- DWH構成より手軽に構築できる
- パーティショニングで抽出コストを削減
- 複雑な処理が必要になったタイミングで Glue ETLを導入することでカバーできる

デメリット

- 直接クエリ（リアルタイム参照）には向いていない

バランス構成 データの流れ



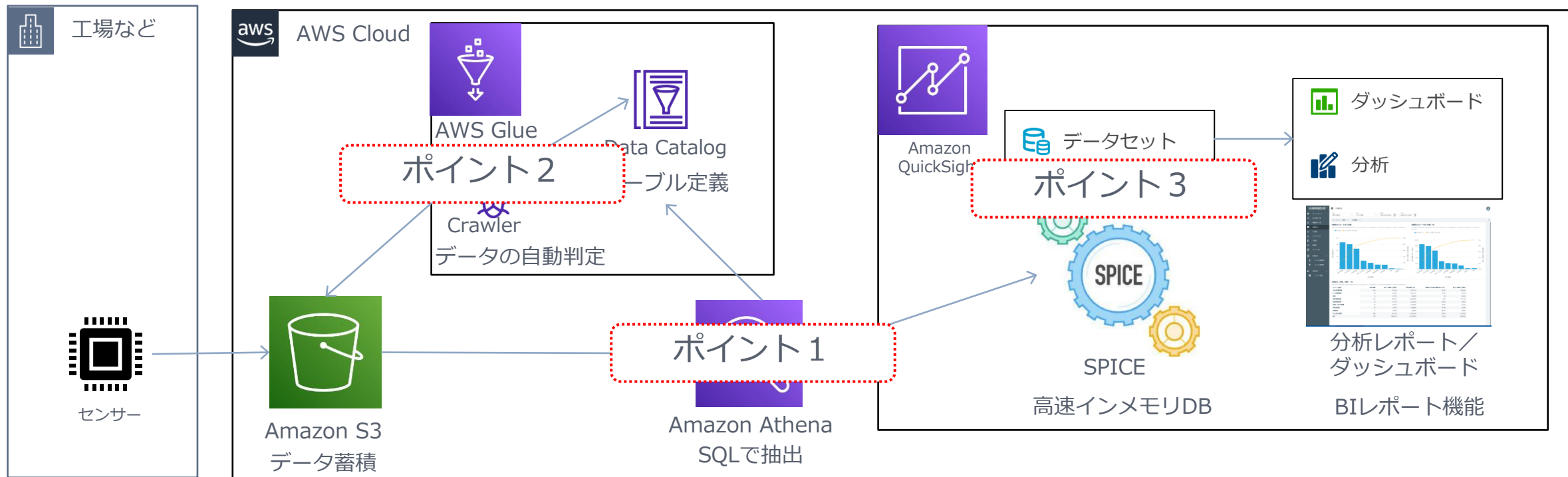
- ① S3をクロール
- ② データカタログを作成
- ③ データカタログを参照
- ④ S3データをSQLで抽出
- ⑤ QuickSightのSPICEに書込

凡例

←----- 参照

←----- 抽出・書込

バランス構成 構築時のポイント



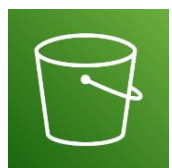
- ポイント1 Athenaのパーティショニング機能
- ポイント2 Glueクローラー利用時の注意点
- ポイント3 QuickSightデータセットの参照方式

ポイント1 Athenaパーティショニング機能

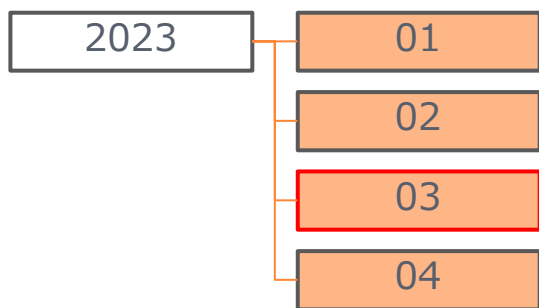
Athenaパーティショニング機能

□ 2023-03のデータを抽出する

▶ パーティショニングが有効でない場合

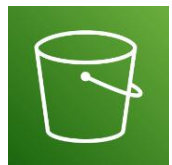


S3 Bucket



S3から全てのデータが一度抽出されてしまう

▶ パーティショニングが有効な場合



S3 Bucket



S3から2023-03のみ抽出できる

Athenaパーティショニングを利用するためのポイント

□ S3のパス設計

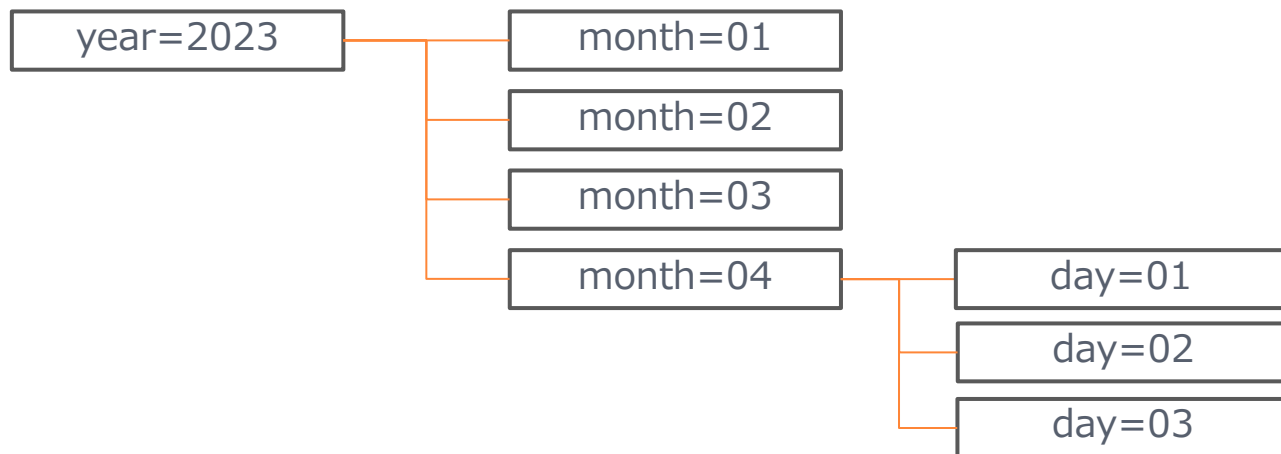
S3のパスは「key=val」（Apache Hive 形式）で設計する

Apache Hive形式のパス例：

`s3:/stylez-sensor-bucket/year=2023/month=03/day=01/`

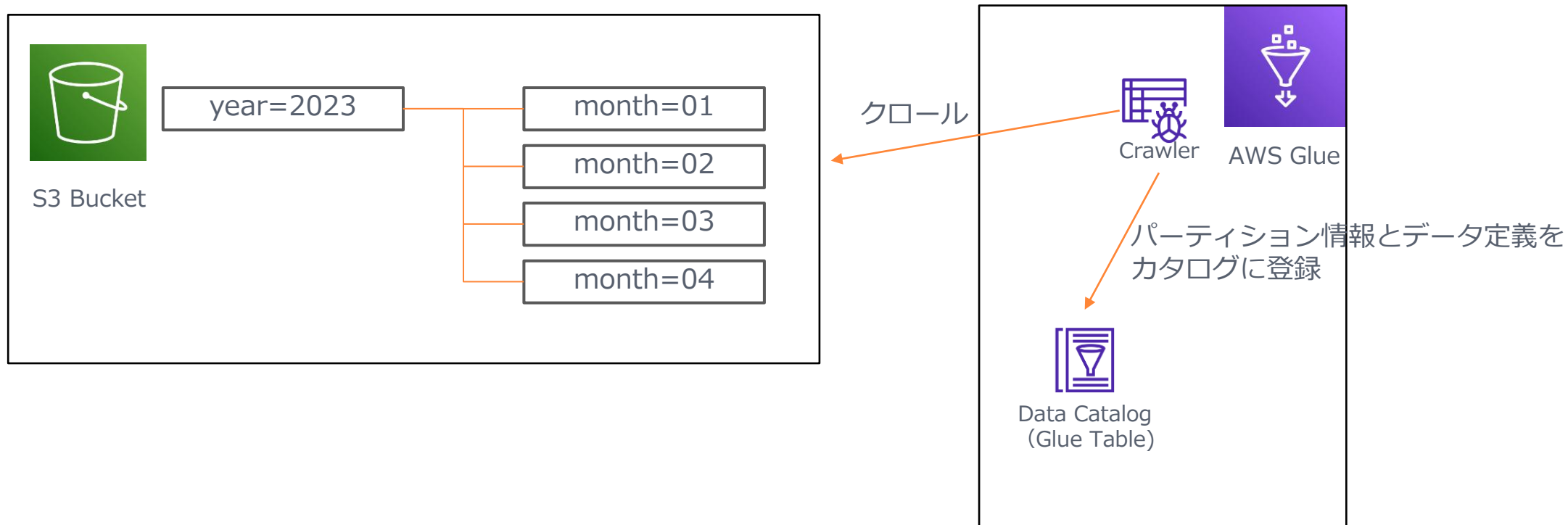


S3 Bucket



Athenaパーティショニング情報の登録・更新方法

1. Glueクローラーで定期的にクロールする



2. MSCK REPAIR TABLE コマンドを実行する

```
Athenaコンソール > MSCK REPAIR TABLE;
```

ポイント2 Glueクローラー利用時の注意点

Glue クローラー利用時の注意点



□ HIVE_PARTITION_SCHEMA_MISMATCHエラーの発生

✖ HIVE_PARTITION_SCHEMA_MISMATCH: There is a mismatch between the table and partition schemas. The types are incompatible and cannot be coerced. The column 'thresholdsensormax' in table 'sampleddb.stylez_sensor_bucket' is declared as type 'double', but partition 'year=2024/month=02/day=01' declared column 'thresholdsensormax' as type 'bigint'.

このクエリは、クエリで修飾されていない限り、「sampleddb」データベースに対して実行されました。エラーメッセージを [フォーラム](#) に投稿するか、クエリ ID: 854c0918-14ba-471f-aba8-becb72944c9a とともに [カスタマーサポート](#) にお問い合わせください。

#	Column name	Data type	Partition key
1	equipmentid	string	-
2	equipmentname	string	-
3	factoryid	string	-
4	factoryname	string	-
5	line	string	-
6	operator	string	-
7	product	string	-
8	sensorid	string	-
9	sensorname	string	-
10	sensoridnamekey	string	-
11	sensortype	bigint	-
12	keyname	string	-
13	timestamp	string	-
14	value	double	-
15	thresholdsensormax	double	-
16	thresholdsensormin	bigint	-
17	year	string	Partition (0)
18	month	string	Partition (1)
19	day	string	Partition (2)

Glue クローラー利用時の注意点

□ HIVE_PARTITION_SCHEMA_MISMATCHエラーの対処法

▶ 対処法 1

- Glueクローラーの設定で「既存パーティションのメタデータも全て更新する」を設定

- ✓ Update all new and existing partitions with metadata from the table

Partitions inherit metadata properties — such as their classification, input format, output format, SerDe information, and schema — from their parent table.

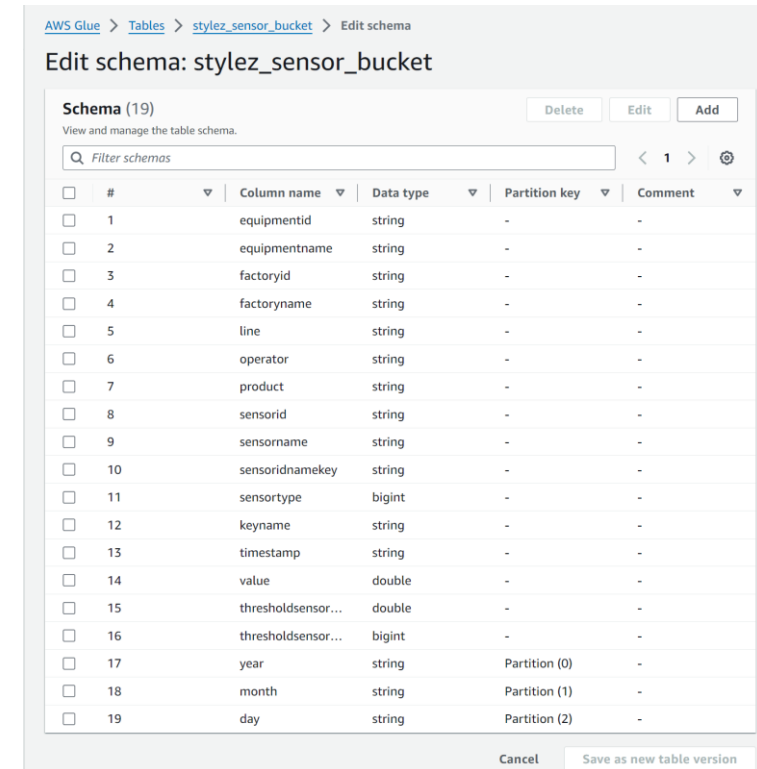
Glue クローラー利用時の注意点

□ HIVE_PARTITION_SCHEMA_MISMATCHエラーの対処法

▶ 対処法 2

- 手動でテーブルを定義する
- パーティション情報はMSCK REPAIR TABLE コマンドで更新する

```
Athenaコンソール > MSCK REPAIR TABLE;
```



The screenshot shows the AWS Glue console interface for editing the schema of a table named 'stylez_sensor_bucket'. The table has 19 columns. The first 16 columns are standard data types, and the last three (year, month, day) are partition keys.

#	Column name	Data type	Partition key	Comment
1	equipmentid	string	-	-
2	equipmentname	string	-	-
3	factoryid	string	-	-
4	factoryname	string	-	-
5	line	string	-	-
6	operator	string	-	-
7	product	string	-	-
8	sensorid	string	-	-
9	sensorname	string	-	-
10	sensoridnamekey	string	-	-
11	sensortype	bigint	-	-
12	keyname	string	-	-
13	timestamp	string	-	-
14	value	double	-	-
15	thresholdsensor...	double	-	-
16	thresholdsensor...	bigint	-	-
17	year	string	Partition (0)	-
18	month	string	Partition (1)	-
19	day	string	Partition (2)	-

ポイント3 QuickSightデータセット参照方式

QuickSightデータセットの役割

- データセットの役割は主に分析・ダッシュボードが参照するデータの定義

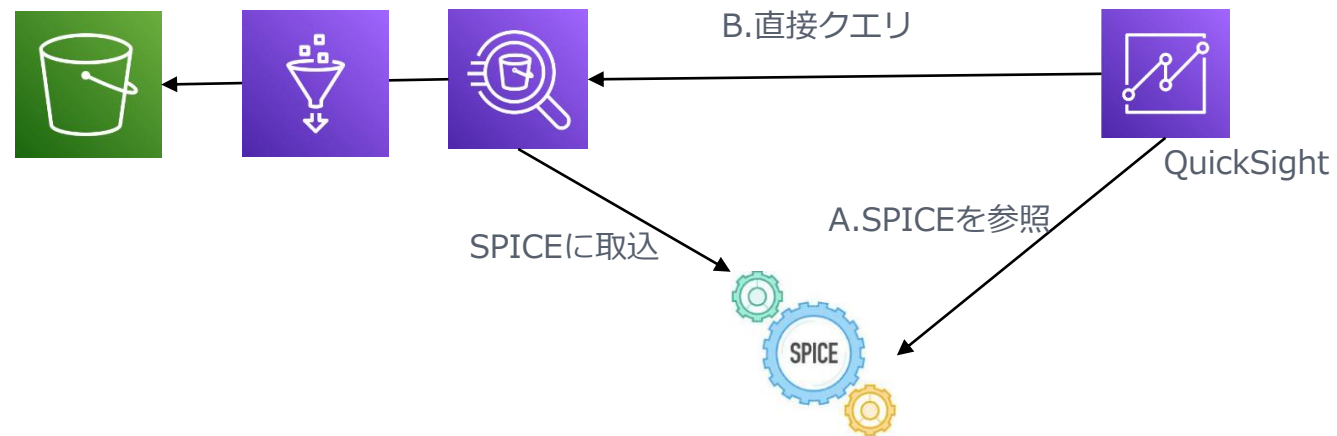


QuickSightデータセットの参照方式について

□ 2つの選択肢

A. QuickSightのインメモリDB「SPICE」に取込んで参照する

B. 直接クエリ



QuickSightデータセット参照方式選択のポイント

□ SPICEか直接クエリか

	メリット	デメリット
A.SPICE	<ul style="list-style-type: none">大量データでも処理速度が速いデータソースへの負荷が少ない (取込時のみアクセス)	<ul style="list-style-type: none">データ更新間隔が最短15分作成者ユーザ数×10G以上の容量を使う場合追加料金がかかる
B.直接クエリ	<ul style="list-style-type: none">リアルタイムなデータ参照が可能容量追加料金はなし	<ul style="list-style-type: none">データソースへの負荷が高くなり S3リクエストコストが大幅に上がる可能性がある処理速度がSPICEに比べて遅い結合データセットでは利用できない

S3 + Athenaの直接クエリで失敗した話

導入前テスト時にS3のリクエスト料金が**通常の1,000倍近く**になってしまった



なぜか

QuickSight発行クエリでパーティショニングが使えず
画面操作毎にS3の全件を抽出してしまっていた

※過去の話である事や画面が複雑だったという事もあり、現在は仕様が変わっている可能性もあります。

SPICE利用時のポイント

増分更新設定の「日付の列」

増分更新を設定 ×

Incremental Refresh では、ウィンドウサイズ設定と日付列を使用して、データセット全体のどの部分が毎回更新されるかを決定します。

日付の列
partition_timestamp

ウィンドウサイズ (数値) ウィンドウサイズ (単位)
2 日

続行

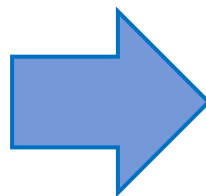
▶ パーティションキーを結合した項目を日付型にして設定

```
date_parse(concat(year,'-',month,'-',day),'%Y-%m-%d') as partition_timestamp
```

データセット型定義のポイント

- 自動で設定されたデータセット型はチェックする

データセット		
sensorID	sensorName	sensorType
<input type="checkbox"/> 文字列	<input type="checkbox"/> 文字列	<input checked="" type="checkbox"/> 整数
SEN-0020	汎用センサー (...)	30
SEN-0010	汎用センサー (...)	92
SEN-0010	汎用センサー (...)	91
SEN-0010	汎用センサー (...)	91



データセット		
sensorID	sensorName	sensorType
<input type="checkbox"/> 文字列	<input type="checkbox"/> 文字列	<input checked="" type="checkbox"/> 文字列
SEN-0020	汎用センサー (...)	30
SEN-0010	汎用センサー (...)	92
SEN-0010	汎用センサー (...)	91
SEN-0010	汎用センサー (...)	91

データセット型定義のポイント

□ データセットの型が適切でない場合

The screenshot shows a list of data set fields. The field 'sensorType' is highlighted with a red box. A context menu is open over it, listing several actions. The 'sensorType' field is marked with a '#' icon, while others are marked with a blue square icon. The context menu options are:

- ビジュアルから削除
- このフィールドにフィルターを追加
- 形式: **テキスト** >
- カウント形式: **1234.5678** >
- ディメンションに変換 (highlighted with a red box)

The list of fields is as follows:

- # sensorType
- sensorTypeNameEN
- sensorTypeNameJA
- sensorTypePulldown
- # thresholdSensorMax
- # thresholdSensorMin

まとめ

まとめ

- S3とQuickSightでBIシステムを構築する場合、Glue、Athenaを使う構成が比較的始めやすい
- Athenaパーティションを使う事でS3抽出コストを圧縮できる
- データセットの直接クエリはリアルタイム分析できる反面、リクエスト負荷が跳ね上がるといったデメリットもある
- QuickSightのデータセットの型は自動判定のままとせず適切な値を設定するとユーザの負荷を減らせる可能性がある



実績豊富なエンジニア集団の技術と開発ツールで
「開発期間/コスト削減」「品質向上」を実現

株式会社スタイルズ

<https://www.stylez.co.jp>

東京都千代田区神田小川町1-2 風雲堂ビル6階

Tel:03-5244-4111

オープンソースソフトウェア推進