



Amazon Timestream

サービス概要

アマゾン ウェブ サービス ジャパン合同会社
長久保 武
Database Solutions Architect

2023/09

内容についての注意点

- 本資料では 2023 年 08 月時点のサービス内容および価格についてご説明しています。AWS のサービスは常にアップデートを続けているため、最新の情報は AWS 公式ウェブサイト (<https://aws.amazon.com/>) にてご確認ください
- 資料作成には十分注意しておりますが、資料内の価格と AWS 公式ウェブサイト記載の価格に相違があった場合、AWS 公式ウェブサイトの価格を優先とさせていただきます
- 価格は税抜表記となっております。日本居住者のお客様には別途消費税をご請求させていただきます
- 技術的な内容に関しましては、有料の [AWS サポート窓口](#)へお問い合わせください
- 料金面でのお問い合わせに関しましては、[カスタマーサポート窓口](#)へお問い合わせください (マネジメントコンソールへのログインが必要です)

アジェンダ

1. 概要とユースケース
2. アーキテクチャー
3. データモデル
4. 価格設定

概要とユースケース

AWSのデータベースサービス

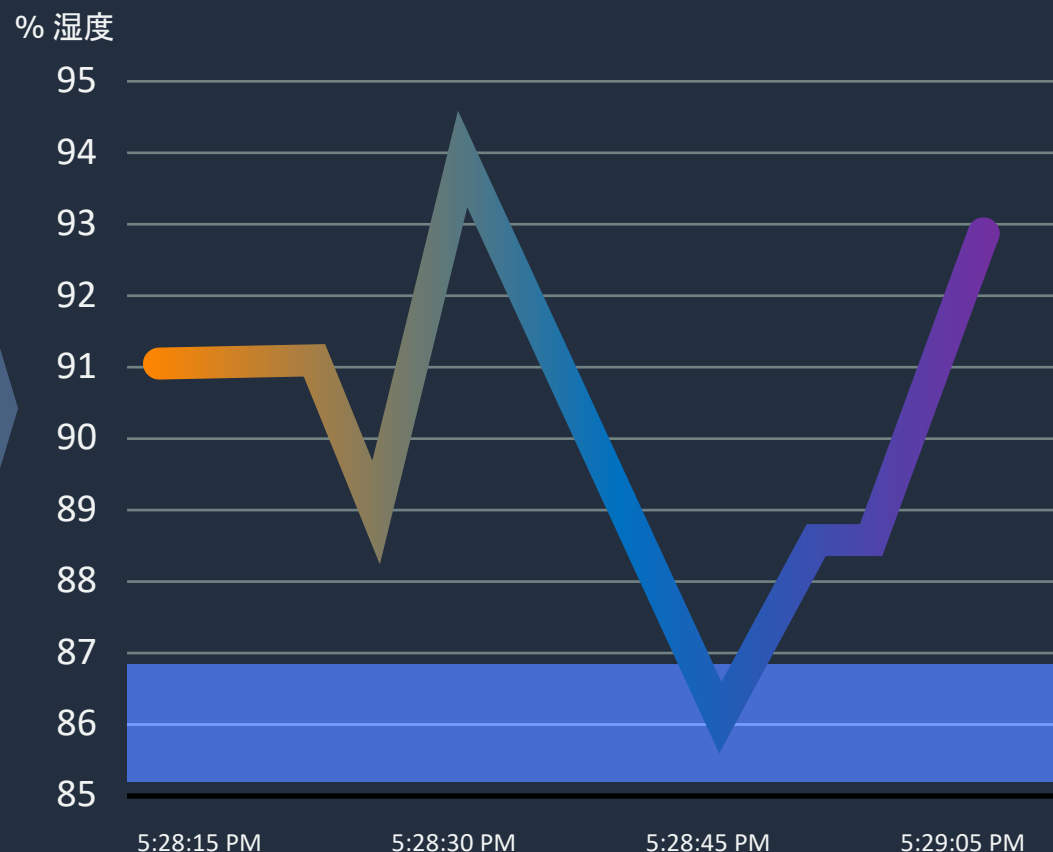
時系列データを
専門に扱うデータベース



時系列データとは

時間経過とともに変化する事象を測定するために、タイムスタンプとともに記録されたシーケンスデータ

timestamp, 測定情報, 属性, ...
2020-01-26 21:42:53, 57, Tokyo ...
2020-01-26 21:42:54, 60, Tokyo ...
2020-01-26 21:42:55, 59, Tokyo ...
...



湿度
86.0
%

時系列データの活用場所



IoT applications



センサー
ネットワーク監視



在庫計画



交通の混雑状況や
遅延のモニタリング



製造・物流の
管理



DevOps analysis



クリックストリーム
分析



株市場の予測



商品需要予測



企業データの分析



App analysis



物理システムの
モニタリング



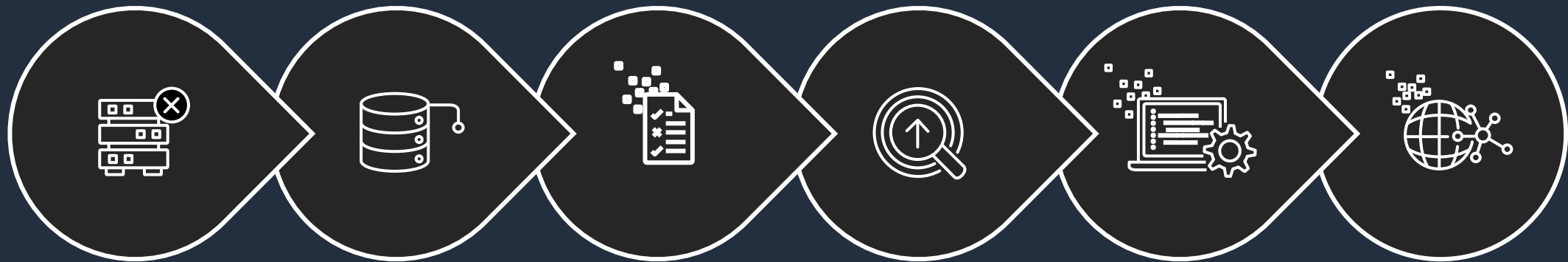
アプリケーションの
モニタリング



ソフトウェアシステムの
モニタリング

Amazon Timestream

Amazon Timestreamは、時間順に並べられたデータを収集、保存、処理するサーバーレス時系列データベース



取り込み、保存、クエリのためのサーバーレス自動スケーリング



データライフサイクル管理



事前にschemaの定義は不要



アダプティブクエリエンジンを使用して、ストレージ階層間を跨いだデータにアクセスが可能



SQLに組み込まれた時系列分析



ML、分析、データコレクターなどのインテグレーション



カスタマーユースケース

<https://aws.amazon.com/timestream/customers/>

業種・業界

- メディア、エンターテインメント
- 家電
- エネルギー
- 広告
- ヘルスケア
- ソフトウェアとサービス
- 物流、運輸業界
- 電気通信事業
- 金融サービス

ユースケース

- アプリケーションの監視
- IoTデバイスの監視
- エネルギー予測
- フリートの管理と最適化
- ネットワーク監視
- 侵入検知
- メディアインテリジェンスと分析
- 行動ログ分析やパーソナライズ



Amazon Timestream customers



Abios is a B2B provider of advanced esports data and statistics products.

"With Amazon Timestream, we are seeing the value of a database purpose-built for time series data in our application cluster, managing time series data with block storage. Timestream allows us to scale our database solution without worrying about our database solution while reducing our overhead costs. We are using Timestream's powerful query engine to push the limits of what we can achieve with our time series data."

Anton Janer, CTO – Abios Games



Autodesk is a global leader in software for architecture, engineering, construction, manufacturing, and media industries.

"At Autodesk, we make software for people who make things. This includes everything from medical devices, and consumer electronics, to the movies and video games that we all know and love. We're looking for the potential to help deliver new workflows by providing a cloud-hosted, scalable time-series database that improves product performance and reduce waste in manufacturing. The key differentiator that we're looking for is without adding a data management burden for the customers nor Autodesk."

Scott Reese, SVP of Manufacturing, Cloud, and Production Products — Autodesk.



Cake is a performance marketing software company that stores and analyzes billions of data points.

"Previously, we used a DIY time series solution that was cumbersome to manage and maintain. AWS was building a time series database service, Amazon Timestream, we signed up for it. We've worked very closely with the AWS service team, giving them feedback and data. Timestream excels in production for the size and scale of time series data we're dealing with. And it's a fully serverless database. It's the first time we've had a single solution for our time series data in partnership with AWS and excited to see what's in store for Timestream."

Tyler Agee, Principal Architect – Cake Software



Cardinal Peak reduces the risk of outsourcing an engineering project by providing end-to-end solutions, leveraging skills in hardware, electronics, embedded, cloud and end-user software.

"Our team has worked closely with AWS on the Amazon Timestream preview. Cardinal Peak is a time-series database to our solutions toolkit, enhancing our solutions where ingesting and analyzing data is required – very applicable for our large-scaled IoT solutions."

Sean O'Neil, VP of Applications Engineering – Cardinal Peak



CleanAir is a global air quality consulting and testing firm that has a long history of providing air quality solutions.

"Our growing digital engineering team is excited to use Amazon Timestream for our time series data. The speed with which we can store, read, and retrieve time series data for analytics, while maintaining high availability, CleanAir partnered with TensorIoT, an APN Advanced Consulting Partner delivering cloud-based engineering, machine learning, and artificial intelligence. Moving to a fully managed database team to focus more on creating great products for our clients and less on database management."

Dan Pearson, Business Leader – CleanAir



Dashbird is a monitoring and analytics platform for serverless applications built on top of AWS Lambda.

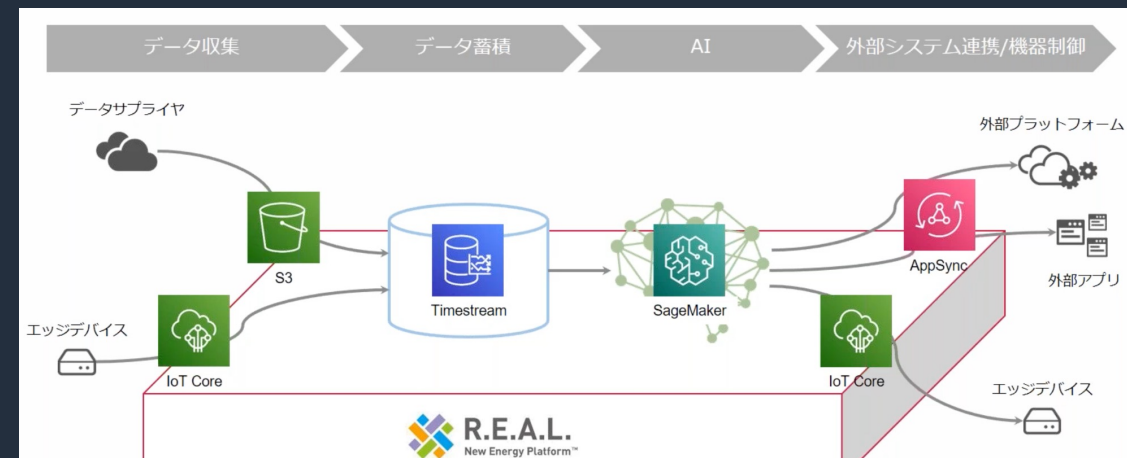
"Amazon Timestream is an amazing addition to the serverless offering. We really value the efficiency of its automated metrical data storing and scaling the load without any overhead."

お客様事例：株式会社アイ・グリッド・ラボ

AWSをフル活用し、迅速な連携を低コストかつ短期間で実現



- 仮想発電所（VPP）事業、エネルギーマネジメント事業、電力供給事業を通じた脱炭素ソリューションを統合的に展開するエネルギーサービスプロバイダー事業を展開
- 再生可能エネルギープラットフォームの構築
 - エッジデバイスや外部からのデータを収集し、Timestream に格納して一元管理、高速なアクセスを実現
 - Amazon SageMaker と組み合わせた機械学習の活用による電力需要・発電量の予測精度向上
 - サーバレスアーキテクチャを効率的に活用し、事業構想からプロトタイプ構築まで6ヶ月の短期開発を実現



お客様事例：株式会社アイ・グリッド・ラボ

AWSをフル活用し、迅速な連携を低コストかつ短期間で実現



Amazon Web Services ブログ

【寄稿】株式会社アイ・グリッド・ラボの太陽光 PPA 事業から蓄電池・EV サービスまでの幅広い GX ソリューションでの Amazon Timestream 活用 (Part1)

by Mitsuaki Tsugo | on 15 6月 2022 | in Amazon Timestream, AWS IoT Core, AWS IoT Greengrass, Customer Solutions, General, Industries, Power & Utilities | Permalink | Share

この投稿はタスデザイングループ 代表取締役 甲田 将史氏から株式会社アイ・グリッド・ソリューションズの AWS IoT Greengrass V2、Amazon Timestream、Amazon SageMaker を利用した再生可能エネルギープラットフォームの構築の取り組みについて寄稿頂いたものです。

Amazon Timestream に格納するエネルギー関連データ

【寄稿】株式会社アイ・グリッド・ラボによる AI・IoT 技術で再生可能エネルギー活用を最適化する次世代エネルギープラットフォーム①では、グリーントランスフォーメーション（GX）の取り組みとして株式会社アイ・グリッド・ソリューションズ（以下「アイ・グリッド」）が運営する R.E.A.L. New Energy Platform®（以下「プラットフォーム」）構築までの経緯とそのアーキテクチャについて紹介しました。

ここからはプラットフォームを構成する主な AWS サービスの具体的な利用方法を数回に分けてご紹介します。初回となる今回は、Amazon Timestream を取り上げます。

向けた取り組みをご紹介したいと思います。

AWS で構築したシステムを運用する際、多くの方が日々コストを気にしていると思います。本プラットフォームにおいても、Amazon Timestream のコスト増減の特性が把握しきれていなかったため注視して運用していましたが、2021 年 9 月の商用リリースからプラットフォームを利用するサービスは順調に拡大を続け、Amazon Timestream のデータ量も増

ら蓄電
amazon

Utilities |

の AWS IoT
プラットフォームの構

減

ソリューションで
構成するAWSサ
でのコスト低減に

Amazon Web Services ブログに
活用ノウハウを寄稿いただきました

[アイ・グリッド・ラボ Timestream]
で検索ください

<https://aws.amazon.com/jp/blogs/news/igrig-lab-ai-iot-nextgen-renewable-ems-platform-2-1/>

<https://aws.amazon.com/jp/blogs/news/igrig-lab-ai-iot-nextgen-renewable-ems-platform-2-2/>

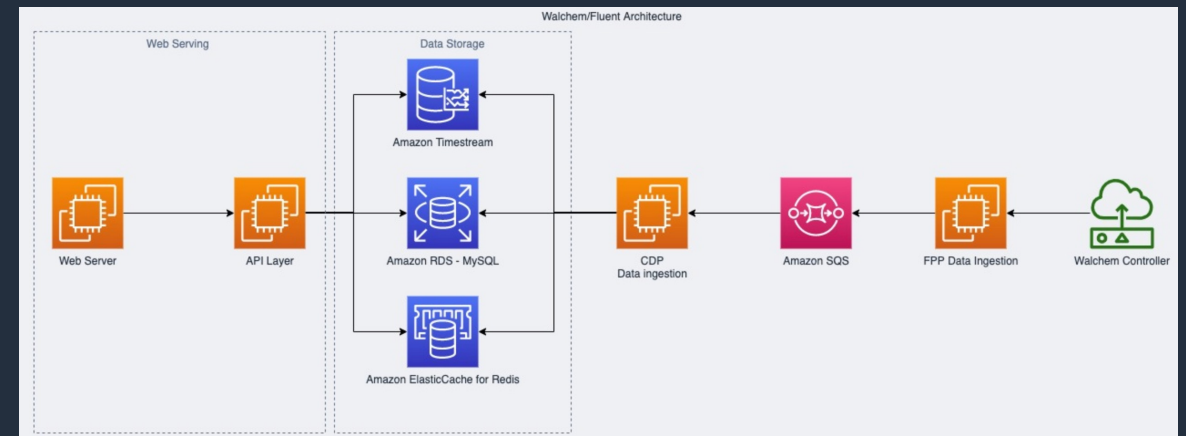


お客様事例：Walchem

オンプレミスのMySQLからTimestreamへ移行、性能向上&運用コストの低減を実現



- オンライン分析機器や電磁定量ポンプを提供するアメリカの大手製造メーカー
- 水処理施設の稼働状況をリアルタイムに監視するための基盤にTimestreamを活用し、スケーラブルなIoTプラットフォームを構築
 - MySQLにて5300台のデバイスから2200万件/日のトランザクションを受け入れ処理するには、ピーク時のスケーリングなど性能担保が困難であり、また運用コストの増大が課題
 - データモデルやSQLの変更は最小限に、Timestreamへの移行が完了。インデックスの管理なども不要になった
 - ストレージコストを50%削減、30-40秒かかっていたグラフの描画が2秒以内に

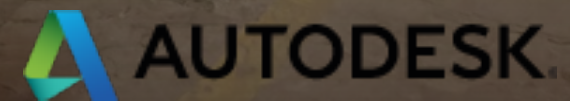


<https://aws.amazon.com/jp/blogs/database/how-bluefin-helped-build-a-scalable-iot-data-platform-for-walchem-using-amazon-timestream/>



お客様事例

- Autodeskは、建築、エンジニアリング/建設、製品設計、製造業界のお客様向けソフトウェアの世界的リーダーです。
- Amazon Timestream は、クラウドでホストされるスケーラブルな時系列データベースを提供することで、新しいワークフローの提供に役立つ能力があります。時系列データにより製品性能が向上し、製造における無駄が削減されます。Timestream が提供する主な差別化要因は、Autodeskのお客様のデータ管理の負担を軽減する価値です。



お客様事例

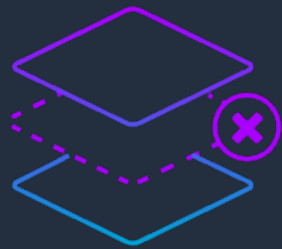
- Disney+ は、時系列データを使用して、ストーリーミングアプリケーションプラットフォームからのテレメトリを保存および分析します。
 - ユーザーエンゲージメント
 - 視聴回数
 - デバイス視聴者数
 - フレームドロップ
- 既存のソリューションに関する課題-InfluxDB on EC2
 - 運用のスケール
 - Influx は、同時クエリで大量の連続データ挿入を効果的にスケールできない。
 - 運用上のオーバーヘッド
 - メンテナンスの負担 (ホストとInflux)



お客様事例

• TimestreamによるDisneyの課題への回答

- Timestream を使用して、動画ストリーミングデータのリアルタイム分析を追跡し、ユーザーの行動やメディアパフォーマンスに関する運用上の洞察を得ることができます。このデータにより、顧客に対するプロアクティブなサービス対応が可能になり、運用コストが削減されます。



シングルプラットフォーム

履歴データを保存し、すべてのデータストレージ階層にわたってクエリを実行しながら、リアルタイムのデータ取り込みをサポートします



サーバーレス

データの取り込み、ストレージ、クエリを完全に切り離します



分析関数

平滑化、近似、補間関数により、リアルタイムにデータを活用し傾向とパターンを特定します

お客様事例

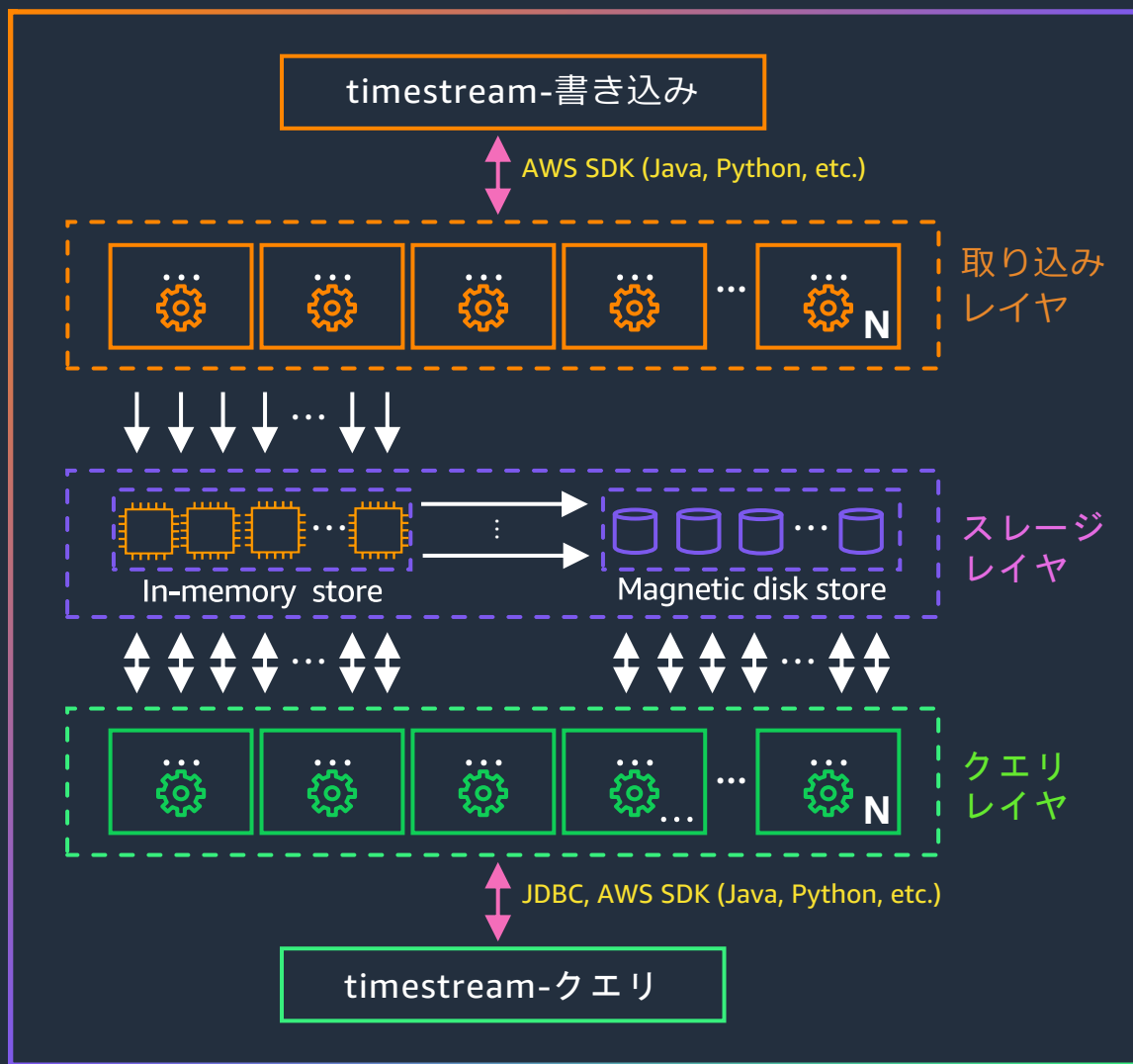
- Guardian Life Insurance Companyは世界最大の生命保険相互会社の1つであり、生命保険、障害保険、その他の福利厚生を提供しています。
- 同社のチームは、システムやアーティファクトリポジトリからメトリクスを収集して処理するアプリケーションを構築しています。彼らはこのデータを自己ホスト型の時系列データベースに保存していました。
- Amazon Timestream のサーバーレス、自動スケールリング、データライフサイクル管理機能に感銘を受け、そしてAmazon Timestream に保存された時系列データを Grafana で視覚化することに楽しみにされています。



アーキテクチャー



Amazon Timestream アーキテクチャ



デカップリング・アーキテクチャを採用

- 取り込みレイヤ、ストレージレイヤ、クエリレイヤが独立し、それぞれ負荷に応じて柔軟にリソースを割り当て
- 可用性：99.99% のSLAを提供

Amazon Timestream アーキテクチャ

高スループットでスケーラブルなデータ取り込み

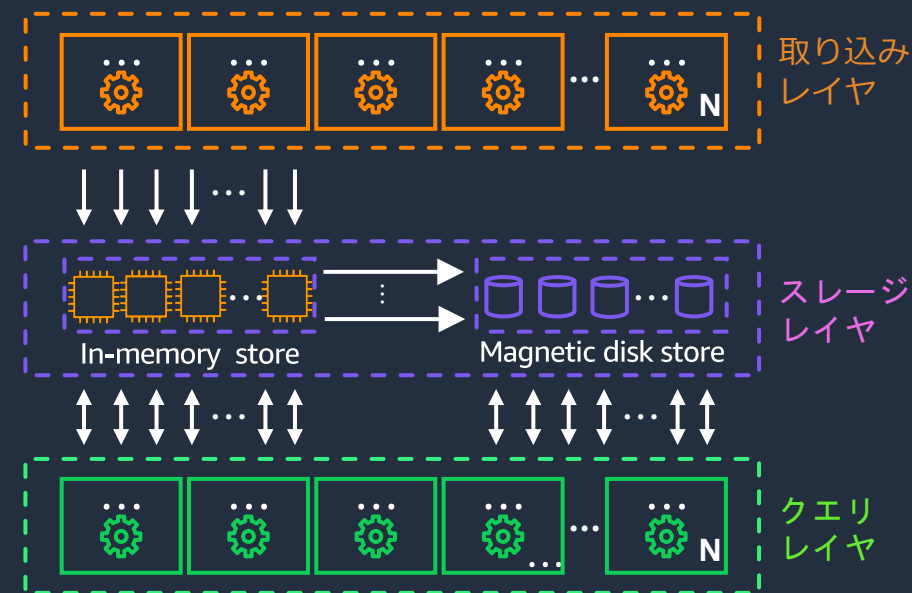
- 複数のアベイラビリティーゾーンにデータを分散配置
- データの重複は自動で除外
- 書き込み I/O のプロビジョニングや構成は不要

複数階層のストレージ

- ストレージ管理は不要、ペタバイト以上まで格納可能
- 2種類のストレージ階層
 - メモリストア
 - 高速なポイントインタイムクエリ用に設計
 - 保持期間を過ぎたデータはマグネティックストアへ自動移行
(最小保持期間：1時間、最大保持期間：1年)
 - マグネティックストア
 - 大量データを参照する分析用に設計、低コストで長期保管が可能
 - 保持期間を過ぎたデータは自動削除
(最小保持期間：1日、最大保持期間：200年)

スケーラブルな SQL クエリエンジン

- 異なるストレージ階層を跨いだ透過的なクエリ実行が可能
- インデックス等のチューニングは不要
- ANSI-2003 SQL標準に準拠。時系列分析関数がデフォルトで用意されている



データモデル

時系列に特化したデータモデルの採用

Database

Series	Timestamp	Measure Value
Dimension location: Tokyo Dimension os_version: 20.04 Dimension hostname: AAAA Measure name: cpu_utilization	2022-06-17 19:00:00	35.0
	2022-06-17 19:00:01	36.7
	2022-06-17 19:00:02	39.4
Dimension location: Tokyo Dimension os_version: 20.04 Dimension hostname: AAAA Measure name: mem_utilization	2022-06-17 19:00:00	54.0
	2022-06-17 19:00:01	56.2
	2022-06-17 19:00:02	57.8
Dimension location: Osaka Dimension hostname: BBBB Measure name: mem_utilization	2022-06-17 19:00:00	21.9
	2022-06-17 19:00:01	23.3

Table: 属性情報 (Series), 測定値情報 (Timestamp, Measure Value), Record

- ✓ タイムスタンプをKeyとした専用データモデル
- ✓ 測定情報毎に型定義
- ✓ データをinsertしたタイミングでスキーマを自動で定義
 - 追加したい属性情報・測定値は、該当データをinsertすることで自動定義される
- ✓ パーティションやindexもTimestreamが自動定義・管理

コンセプトと用語

Database

Table

Series	Timestamp	Measure Value
Dimension location: Tokyo Dimension os_version: 20.04 Dimension hostname: AAAA Measure name: cpu_utilization	2022-06-17 19:00:00	35.0
	2022-06-17 19:00:01	36.7
	2022-06-17 19:00:02	39.4
Dimension location: Tokyo Dimension os_version: 20.04 Dimension hostname: AAAA Measure name: mem_utilization	2022-06-17 19:00:00	54.0
	2022-06-17 19:00:01	56.2
	2022-06-17 19:00:02	57.8
Dimension location: Osaka Dimension hostname: BBBB Measure name: mem_utilization	2022-06-17 19:00:00	21.9
	2022-06-17 19:00:01	23.3

属性情報

測定値情報

レコード

Database

- Tableセットの最上位のコンテナ

Table

- 時系列データを保持するコンテナ

ディメンション (Dimensions)

- 測定値を識別するための属性情報。時間の経過とともに変化しない属性

メジャー (Measures)

- 測定値。名前(measure_name)と値(measure_value)から構成される。メジャーを持たないレコードは存在しない

レコード (Record)

- 単一の時系列データポイント
- 各レコードは、1つ以上のディメンション、タイムスタンプ、時間の経過とともに変化するメジャーで構成される

書き込みのデータモデル

書き込みでは、**シングルメジャーレコード**と**マルチメジャーレコード**の2つのデータモデルをサポート

シングルメジャーレコード

単一のメジャーを1つのテーブル行に格納するデータモデル

time	hostname	measure_name	measure_value::double
2022-06-17 19:00:00	host-24GJU	cpu	35.0
2022-06-17 19:00:00	host-24GJU	memory	54.9
2022-06-17 19:00:00	host-24GJU	disk_iops	38.2

マルチメジャーレコード (re:Invent2021より追加されたデータモデル)

複数のメジャーを1つのテーブル行に格納することが可能な新しいデータモデル。デバイスが複数の測定値を同時に送信した場合に、データ書き込み・保存・クエリスキャンのコスト効率に優れる

time	hostname	measure_name	cpu	memory	disk_iops
2022-06-17 19:00:00	host-24GJU	metrics	35.0	54.9	38.2

データの書き込み方法

データの挿入には AWS SDK,
Timestream API を利用

全ての AWS SDK 言語
のサポート

Java, Python, Node.js, .NET, etc.

AWS Command Line
Interface (AWS CLI)

大量データのロード
バッチロード(CSV)

アダプターとプラグイン

AWS IoT Core

Amazon Kinesis Data
Analytics for Apache Flink
connector (GitHub)

Telegraf connector
(GitHub)

クエリ方法

ANSI-2003 SQL 標準に準拠

時系列関数、補完関数、
gap-filling関数を用意

250以上のスカラー関数、集約関数
、ウィンドウ関数をカバー

AWS SDK または AWS CLI を使
用してクエリを実行

Java, Python, Node.js, .NET, etc.

JDBC Driver (read-only driver)

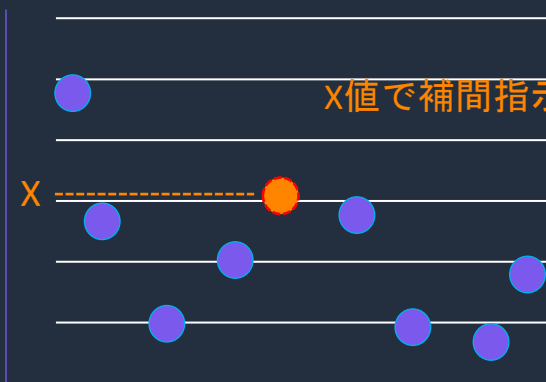
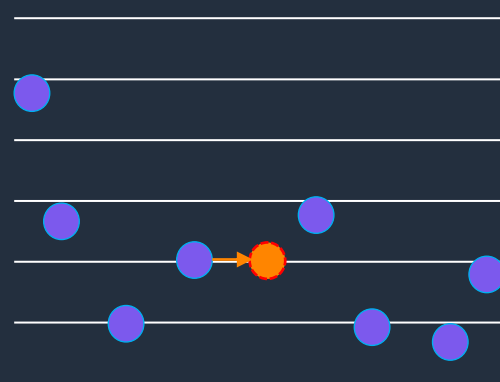
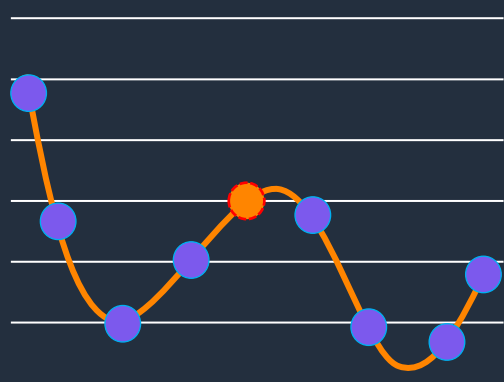
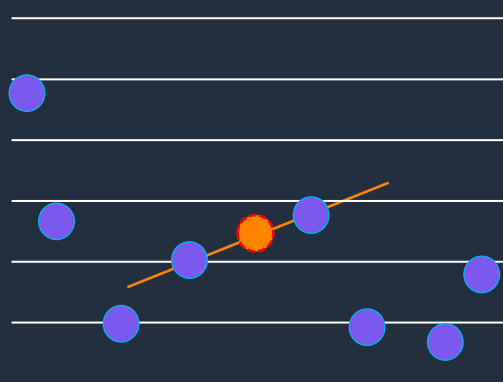
Amazon QuickSight

Grafana (Open Source Edition)

時系列機能 - 補間関数

時系列データ型を利用して欠損値の補間が可能。4種の補間手法を提供し、利用者側での複雑な計算処理が不要

● 補間数値



補間手法

線形補間

キュービク
スプライン補間
(3次スプライン補間)

Last Observation Carried
Forward補間
(LOCF補間)

固定値補間

対応関数

`interpolate_linear()`

`interpolate_spline_cubic()`

`interpolate_locf()`

`interpolate_fill()`



クエリ用のデータモデル

フラットモデルと時系列モデルの2つのデータモデルをサポート。データはフラットモデルを使用して保存され、時系列分析には時系列モデルを使用

フラットモデル(シングル/マルチ)

クエリ時のデフォルトモデル。時系列データを表形式で表す。各行は時系列内の特定の時間に対する測定値のアトミックなデータポイント。ほとんどの標準 SQL 関数が使用可能

time	vpc	instance_id	measure_name	measure_value :: double
2022-06-04 19:00:00.00000000	vpc-1a2b3c4d	i- 1234567890abc def0	cpu_utilization	35
2022-06-04 19:00:01.00000000	vpc-1a2b3c4d	i- 1234567890abc def0	cpu_utilization	38.2
2022-06-04 19:00:02.00000000	vpc-1a2b3c4d	i- 1234567890abc def0	cpu_utilization	45.3
2022-06-04 19:00:00.00000000	vpc-1a2b3c4d	i- 1234567890abc def0	memory_utilizat ion	54.9

時系列モデル(Time-Series)

時系列分析に使用されるデータモデル。特定のディメンションに対応したメジャーをタイムスタンプ付き値の配列で表す。

region	vpc	instance_id	cpu_utilization
us-east-1	vpc- 1a2b3c4d	i- 1234567890a bcdef0	[{time : 2019-12-04 19 : 00 : 00.0000000000,value : 35}, {time : 2019-12-04 19 : 00 : 01.0000000000,value : 38.2}, {time : 2019-12-04 19:00 : 02.0000000000,value : 45.3},]

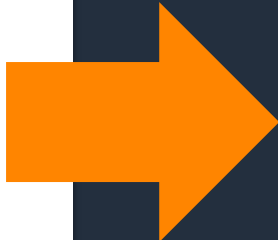


INTERPOLATE_LINEAR (timeseries , array[timestamp])

Timestream が持つ補間関数の一つ（線形補間）

第 1 引数 : 時刻と気温の時系列データ
第 2 引数 : 10 分間隔を指定

```
SELECT
device_id,
INTERPOLATE_LINEAR(
CREATE_TIME_SERIES(time, temperature),
SEQUENCE( min(time) , max(time), 10m )
) AS INTERPOLATED_TEMP
FROM Blog.SensorReading
WHERE
time BETWEEN ago(365d) AND now()
GROUP BY device_id
```



device_id	interpolated_temp
real1	[{ time: 2021-01-01 09:01:00.000000000, value: 102 }, { time: 2021-01-01 09:11:00.000000000, value: 102 }, ...]
	[

9:11:00 のデータ有無に関わらず、value が取得出来る

価格設定



Timestream の価格設定

2023/8時点

東京リージョン（リージョン毎に異なる価格設定となります）

詳細はドキュメントをご参照ください

<https://aws.amazon.com/jp/timestream/pricing/>

https://docs.aws.amazon.com/ja_jp/timestream/latest/developerguide/metering-and-pricing.html

書き込み

- 書き込みバイト数に応じて料金発生。1 回の書き込みは 1 KB
- 100 万回書き込みあたりの料金は \$0.625
- 1 KB 未満の書き込みは1 KB の単位に切り上げ

ストレージ

- インメモリストア：1 GB あたりの料金は、1 時間あたり \$0.045
- マグネティックストア：1 GB あたりの料金は、1 か月あたり \$0.0375

クエリ

- スキャンされたバイト数に対して料金発生
- クエリでスキャンされたデータ 1 GB あたりの料金は \$0.0125
- 10 MB 未満のクエリは 10 MB の単位に切り上げ

1ヶ月の無料トライアル開始

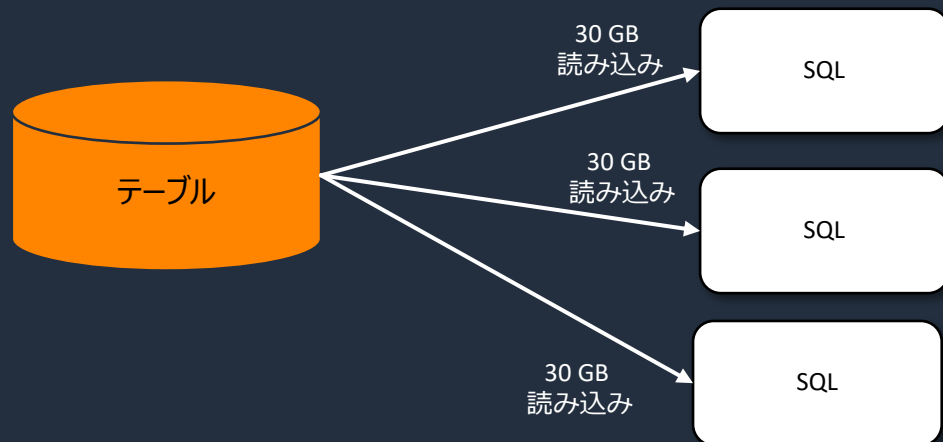
- 2023/6 より、1 ヶ月間の無料トライアルを開始しました。
- 以下の範囲で無料トライアル可能です。
 - 書き込み：50 GB
 - インメモリストア：750 GB-時間
 - マグネティックストア：100 GB-月
 - クエリ：750 GB



Scheduled Query

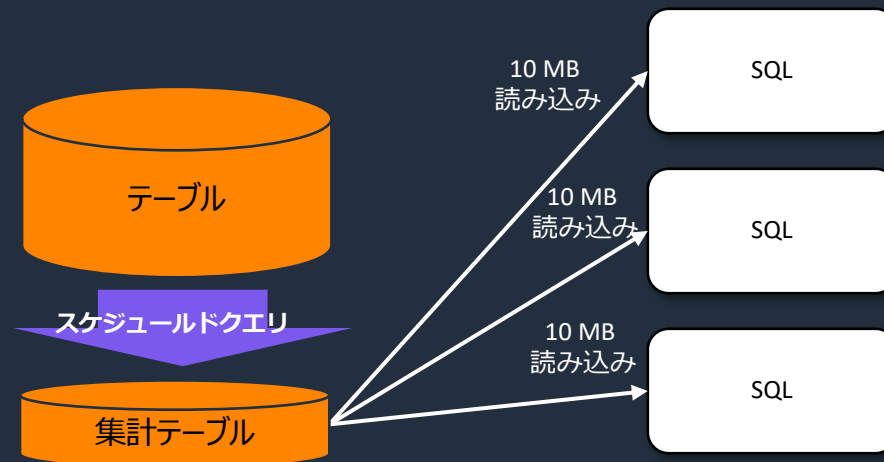
クエリをスケジュール実行し結果データを別テーブルとして保持、SQLの読み込み時間の短縮し、クエリコストを最適化

スケジュールドクエリ：無



読み込み量の多い SQL を投げるような場合、画面表示毎に同じ SQL が実行され、結果、コストがかかる場合がある。

スケジュールドクエリ：有



定期的集計テーブルに結果を作成しておき、アプリケーションからは集計テーブルを参照。読み込み量を抑えることでコスト最適化が可能。



Thank you!