なぜビットバンクは Amazon Redshiftを 選んだのか？

ビットバンク株式会社
システム部門 プラットフォーム部 谷津 香

bitbank

自己紹介

谷津香

ビットバンク株式会社
システム部門 プラットフォーム部
インフラチーム エンジニア

@mdps513

本日の内容

話すこと

・Redshiftの選定理由

・データ連携方法

話さないこと

・暗号資産取引所のデータの中身

会社紹介

会社概要

会社名	ビットバンク株式会社
URL	https://bitbank.cc/
業務内容	暗号資産（仮想通貨）関連事業、関東財務局長（暗号資産交換業者） 登録番号 第00004号
設立	2014年5月
役員	代表取締役社長 執行役員 CEO 廣末紀之、取締役 執行役員 CCO 林潔 社外取締役 都木聡、社外取締役 金山伸宏、社外取締役 駒林素行、社外取締役 大澤弘之 監査役 高木明、監査役 杉田義明、常勤監査役 高橋洋二郎
従業員数	119名 ※2022年2月16日現在　役員・業務委託含む
資本金	86億4,721万円（資本準備金含む）
住所	〒141-0031 東京都品川区西五反田7丁目20-9 KDX西五反田ビル7F

事業概要

ビットバンク株式会社は日本でTop3（預り資産）に入る暗号資産取引所を運営している暗号資産交換業者

（関東財務局長：登録番号第00004号）

主要事業

bitbank.cc
暗号資産取引

bitbank MARKETS
暗号資産マーケットの情報サイト

「bitbank.cc」のサービス概要

bitbank

- 暗号資産取引所
- 暗号資産販売所
- 貸して増やす（レンディング）

取り扱い通貨

BTC　LTC　XRP
ETH　MONA　XLM
BCC (BCH)　BAT　QTUM
OMG　XYM　LINK
MKR

ビットコインの技術で、
世界中にあらゆる価値を流通させる

OUR MISSION.

ビットバンクがなぜ**Amazon Redshift**を
選んだかの前に

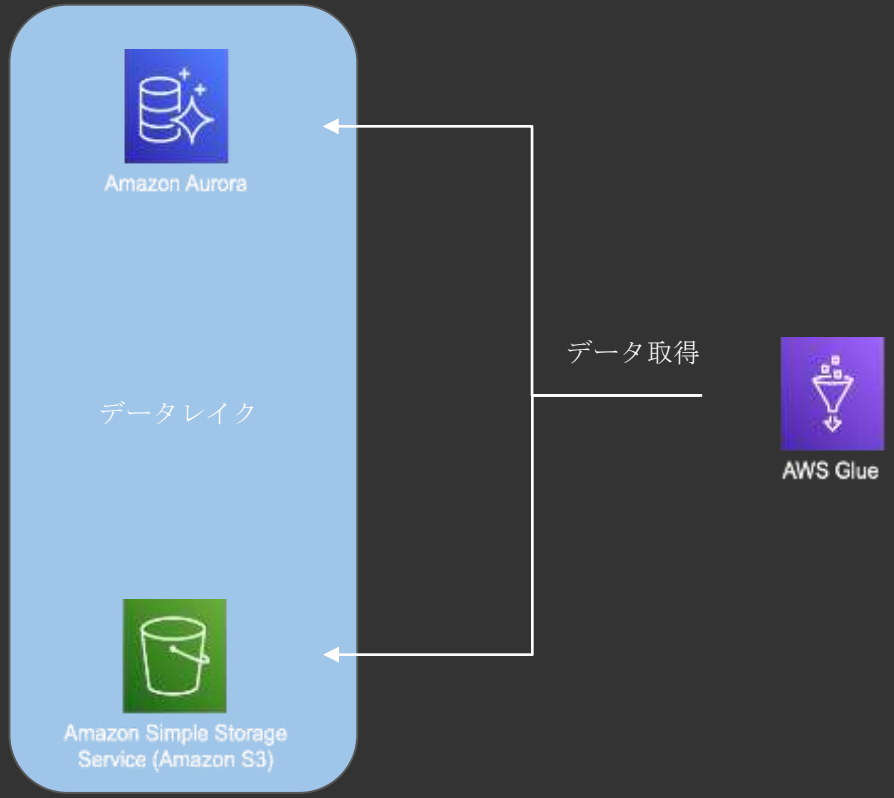
まず、ビットバンクの既存のデータ分析基盤に
ついてお話しします。

当初の分析基盤の要件(初期)

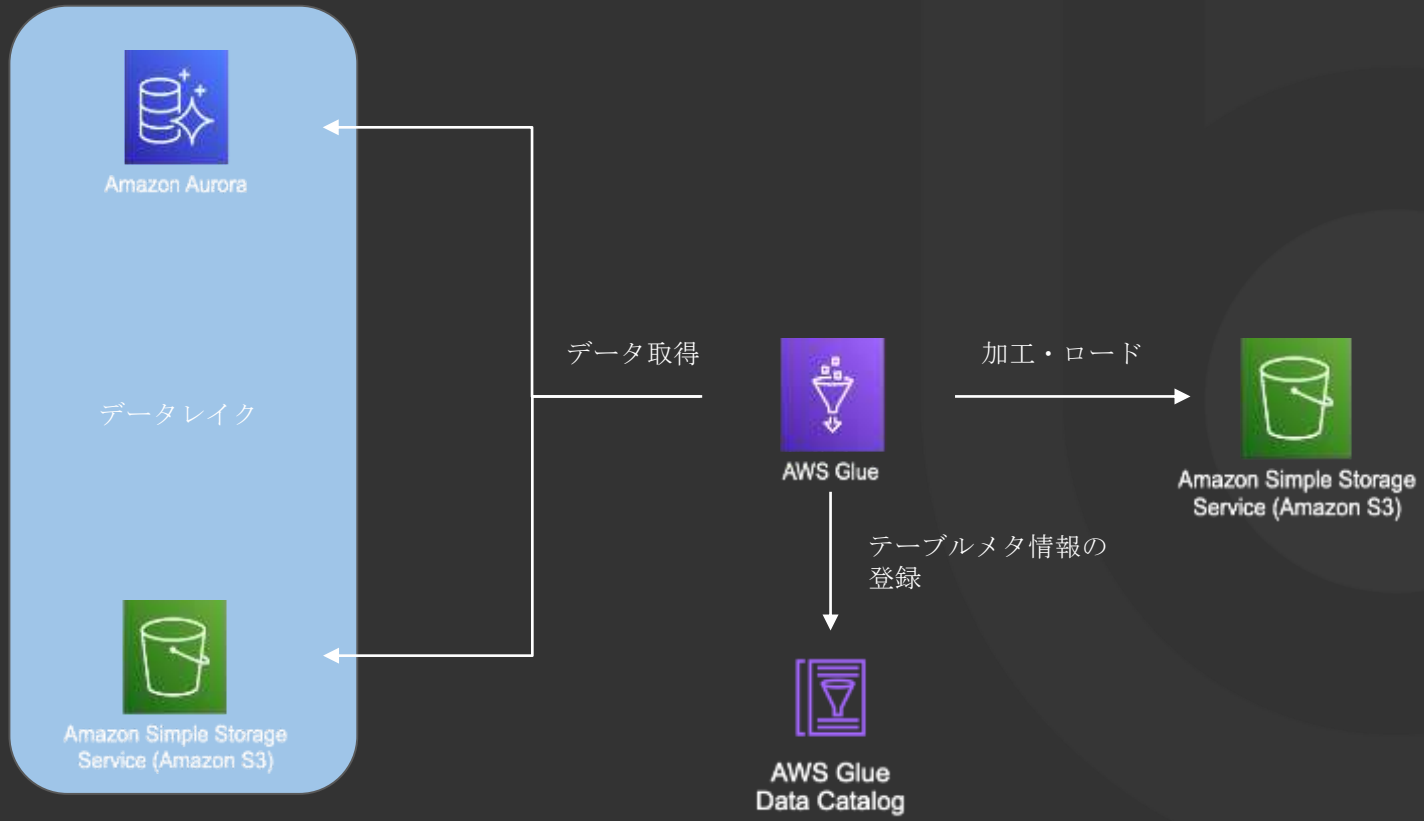
- データはAurora, S3に保存されている
- Aurora, S3から必要なデータのみを取り出し、連携する
- BIツールはRedashで利用者はエンジニアに限らずビジネス部門も利用する

アーキテクチャ図

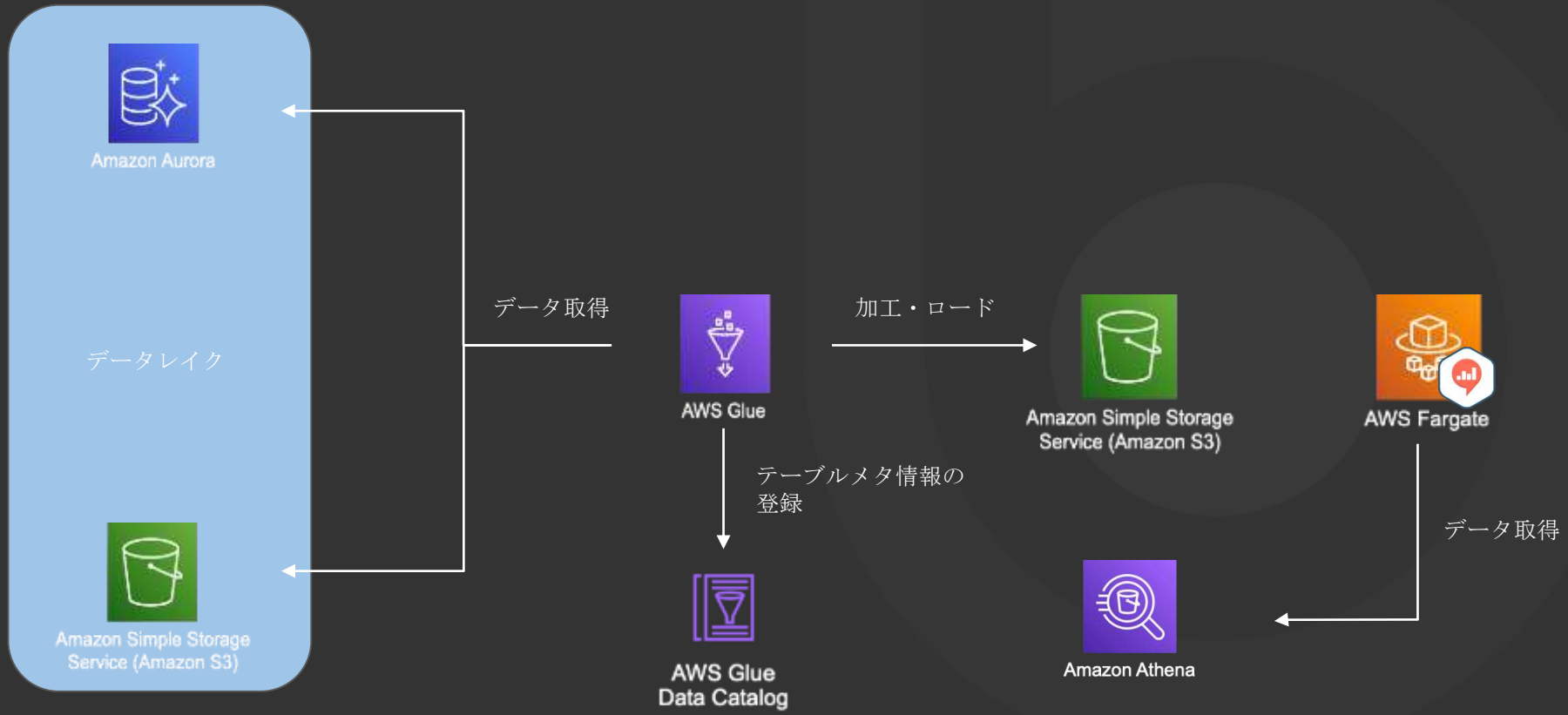
アーキテクチャ図



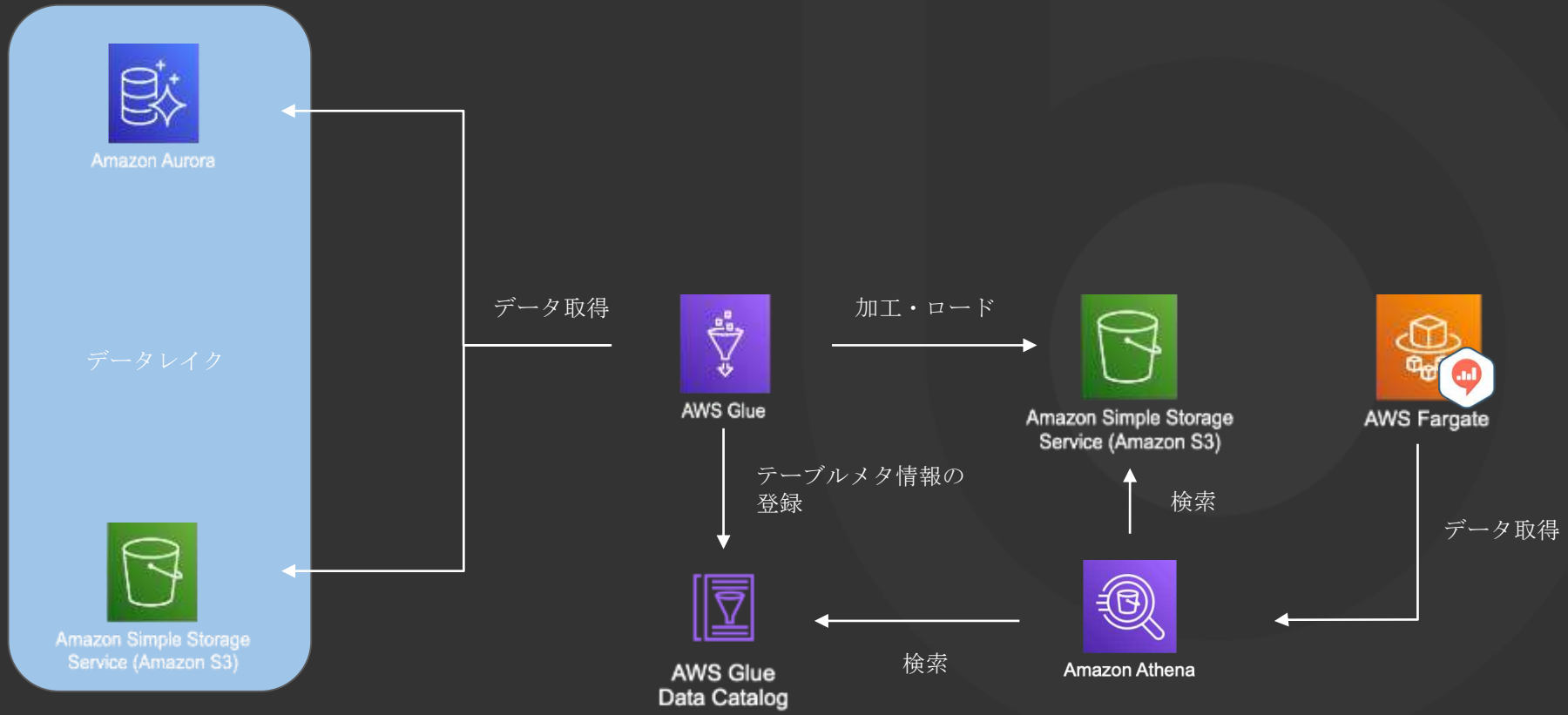
アーキテクチャ図



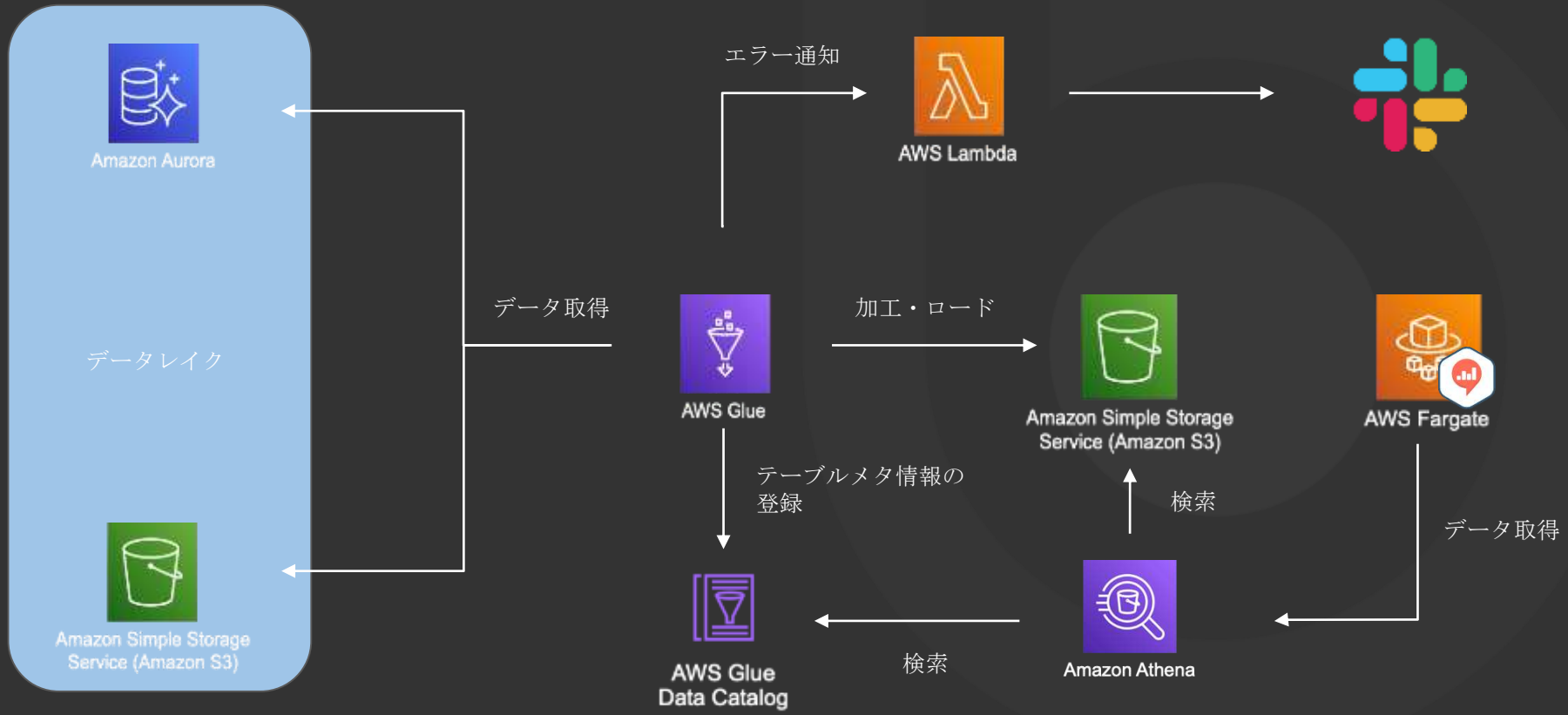
アーキテクチャ図



アーキテクチャ図



アーキテクチャ図



ユーザー側

ユースケースの例

- 事業計画の推進における意思決定
- Webマーケティング施策・効果検証
- 市場動向の分析・予測
- 会計利用の元となる帳票データ
- 取引モニタリング

などなど...

一見問題なさそうに見えるが・・・？

運用していくうちに出てきた問題

- 個人情報を含むデータを簡単に利用したいニーズへの対応ができない
  - アクセスコントロールの仕組みがない
  - セキュリティ/統制上の課題がある
  - エンジニアデータ抽出業務依頼やサポート負荷がかかる
- 基盤/構成自体の性能の限界
  - Athena 自体のクエリ実行速度が遅い
  - S3 の API制限によりAthena から大きめのクエリが実行できない、失敗する
    - 約60GBのスキャン量に対してAthenaの終了に30分ほどかかる
- 柔軟なデータソース連携に対応していない
  - Athenaからのソースは事実上S3から

これでは組織のスケールに基盤側が追いつかない・・・

ここで、改めて分析基盤の要件・用途を整理する

- データはAurora, S3に保存されている
- Auroraから必要なデータのみを取り出し、連携する
- BIツールはRedashで利用者はエンジニアに限らずビジネス部門も利用する

ここで、改めて分析基盤の要件・用途を整理する

- データはAurora, S3に保存されている
- Auroraから必要なデータのみを取り出し、連携する
- BIツールはRedashで利用者はエンジニアに限らずビジネス部門も利用する
- 高いパフォーマンス ← **New!!**
- 多様なデータソースサポートと柔軟なデータアクセス ← **New!!**
- 個人情報取扱いのためのアクセスコントロール ← **New!!**
- 監査ログなどの**IT-GC/IT-AC**含む金融セキュリティ、統制要件 ← **New!!**

ここで、改めて分析基盤の要件・用途を整理する

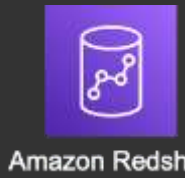
- データはAurora, S3に保存されている
- Auroraから必要なデータのみを取り出し、連携する
- BIツールはRedashで利用者はエンジニアに限らずビジネス部門も利用する
- 高いパフォーマンス ← **New!!**
- 多様なデータソースサポートと柔軟なデータアクセス ← **New!!**
- 個人情報取扱いのためのアクセスコントロール ← **New!!**
- 監査ログなどの**IT-GC/IT-AC**含む金融セキュリティ、統制要件 ← **New!!**

→より高度な機能・統制基盤が求められるフェーズに

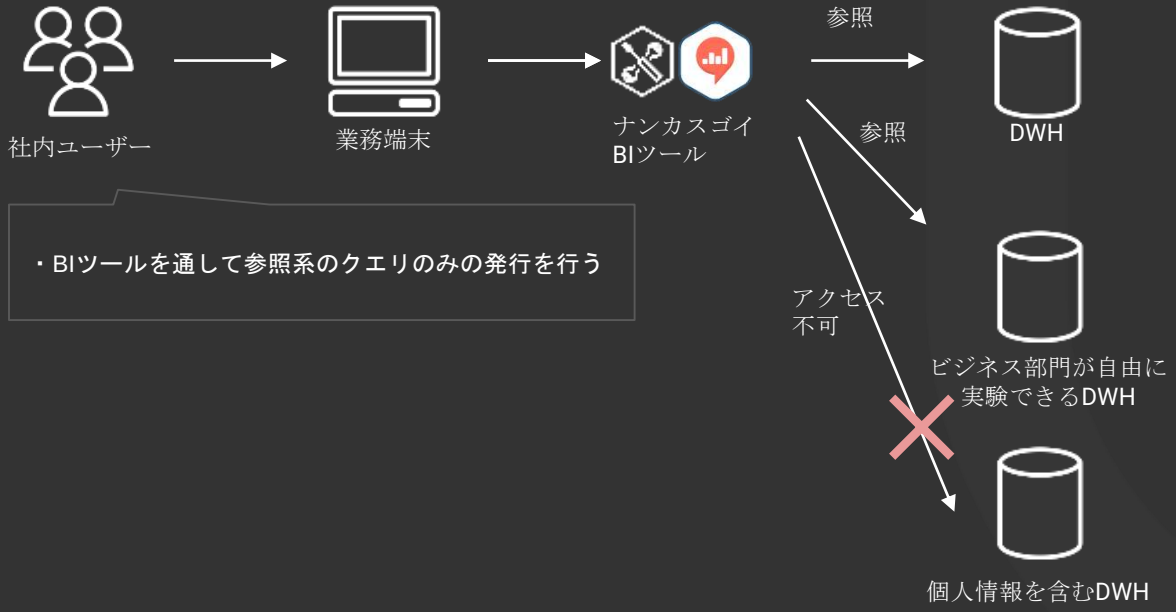
もしかして、**DWH**の出番？

Data Ware Houseを選定する

有名どころ3製品の比較を実施

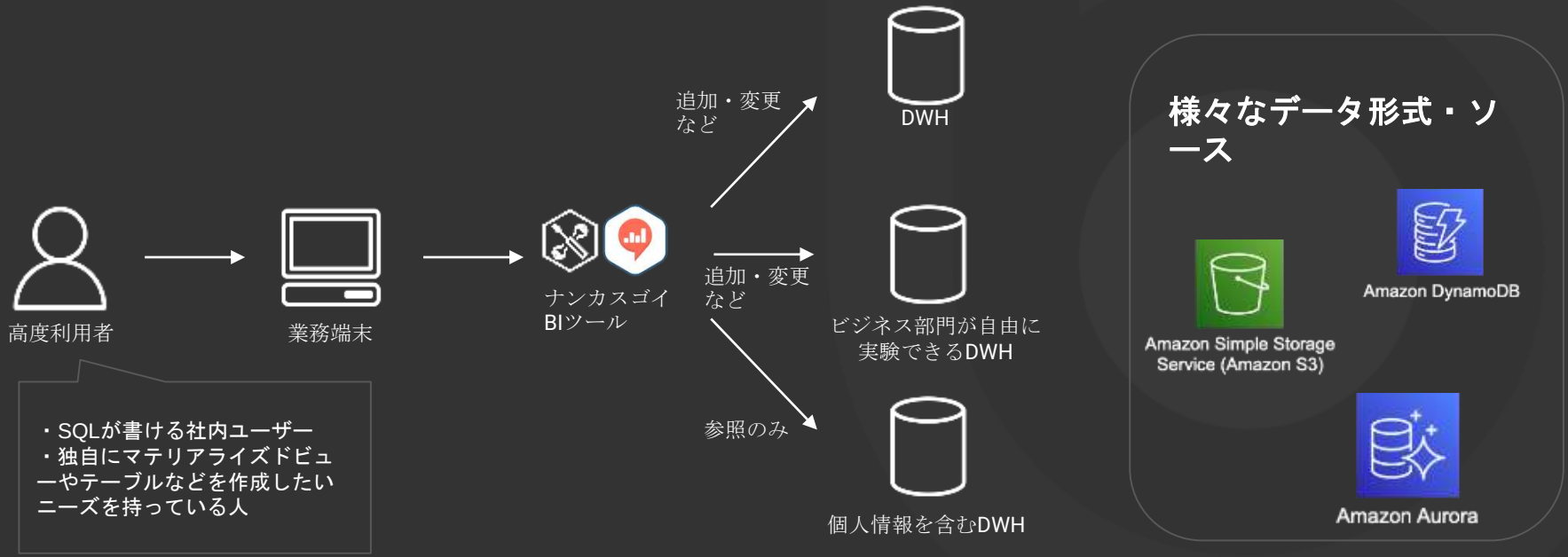
DWH	Amazon Redshift	Snowflake	Google BigQuery
			
言語	PostgreSQLベース	標準SQL	標準SQL/レガシーSQL
特徴	・AWSサービスとの相性が良い ・AQUAで検索性能のブーストができる ・時間課金型 ・RIを買うことでコストを安く抑えられる	・他サービス連携とのバランスが良い ・インスタンス管理がわんぽちで簡単 ・SaaSサービスなので他二社より管理が楽 ・クエリ課金/インスタンス課金型(クレジット形式)	・Googleの他サービス(GoogleAnalytics, Firebase)との相性がいい ・裏側で自動的にスケールするためメンテがほぼ不要 ・クエリの書き方が独特 ・クエリ課金型

利用イメージ

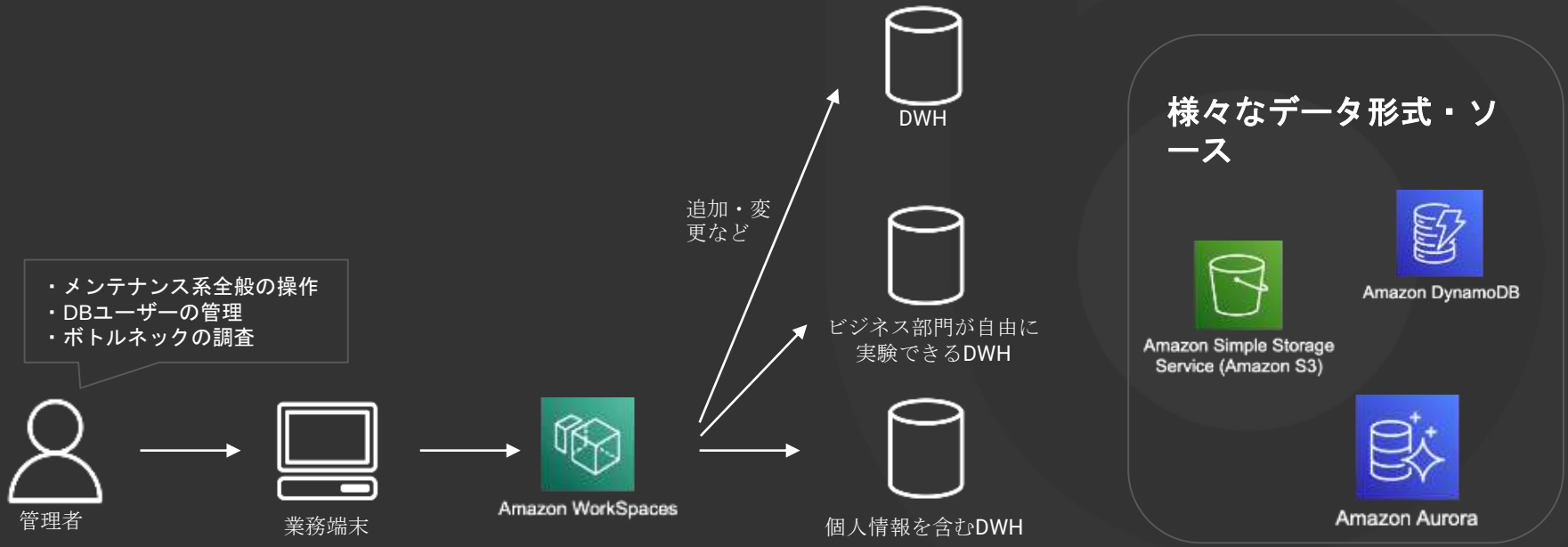


・BIツールを通して参照系のクエリのみの発行を行う

利用イメージ



利用イメージ



比較項目

- データソース
- DWH自体の機能
- 監査ログ
- BIツール
- アプリケーションからの操作
- アクセスコントロール
- セキュリティ
- レスポンス
- スケーラビリティ
- 初回のデータ取り込み
- コスト

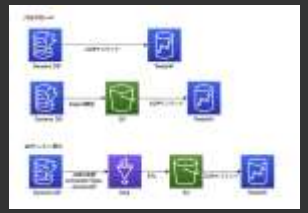
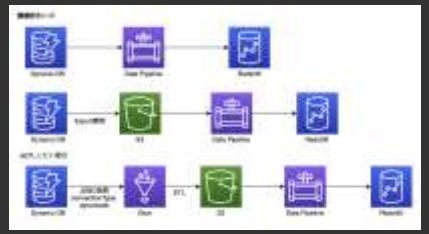
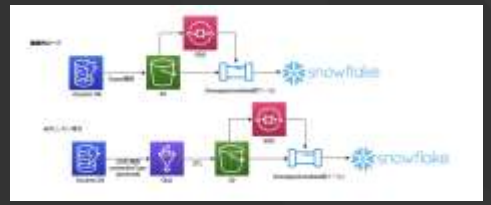
比較項目(今回取り上げる項目)

- データソース
- DWH自体の機能
- 監査ログ
- BIツール
- アプリケーションからの操作
- アクセスコントロール
- セキュリティ
- レスポンス
- スケーラビリティ
- 初回のデータ取り込み
- コスト

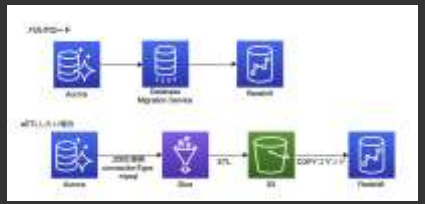
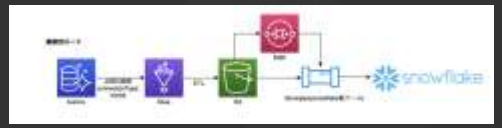
AWSの様々なサービスからの連携で比較

1. DynamoDB
2. Aurora
3. S3

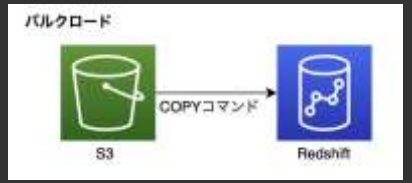
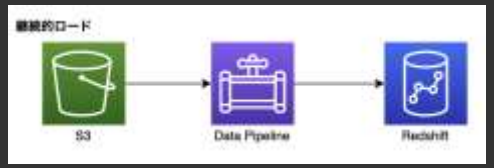
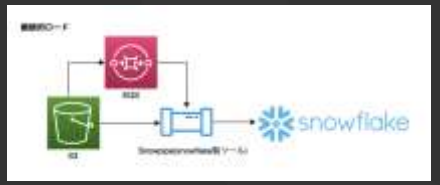
データソース(DynamoDB)

ソース	Amazon Redshift	Snowflake	Google BigQuery
説明	継続的なデータロード: Data Pipeline データ加工: Glue Federated Queryで外部テーブルとして定義する方法もある	一括で既存のデータロード: COPYコマンド 継続的なデータロード: datapipeline データ加工: Glue	S3にエクスポートして、Cloud Storage経由でBigQueryに渡す
DynamoDBからの経路	バルクロード  継続的ロード 	バルクロード  継続的ロード 	バルク・継続的ロード 
評価	○	○	△

データソース(Aurora)

ソース	Amazon Redshift	Snowflake	Google BigQuery
説明	継続的なデータロード: Data Pipeline データ加工: Glue Federated Queryで外部テーブルとして定義する方法もある	一括で既存のデータロード: COPYコマンド 継続的なデータロード: datapipeline データ加工: Glue	S3にエクスポートして、Cloud Storage経由でBigQueryに渡す
Auroraからの経路	バルクロード  継続的ロード 	バルクロード  継続的ロード 	バルク・継続的ロード 
評価	○	○	△

データソース(S3)

ソース	Amazon Redshift	Snowflake	Google BigQuery
説明	一括で既存のデータロード: COPYコマンド 継続的なデータロード: Data Pipeline	・S3からの一括ロード ・Snowpipeで継続的なデータのロード	Cloud Storage経由でBigQueryに渡す
Auroraからの経路	バルクロード  継続的ロード 	バルクロード  継続的ロード 	バルク・継続的ロード 
評価	○	○	△

データソース

- AWS以外のサービスの連携の容易さ
  - Google Analytics, Adjust, Marketoを調査

対象AWSサービス	Amazon Redshift	Snowflake	Google BigQuery
Google Analytics	AppFlowを用いてワンポチ連携してS3にロードして対応	Snowpipeを利用	ワンポチ連携 Analytics以外にもGoogle系は全部連携可能
Adjust	S3にロードする SQL COPYコマンド	デフォルトで機能の用意なし	Cloud Storageに同期可能
Marketo	AppFlowを用いてワンポチ連携してS3にロードして対応	デフォルトで機能の用意なし	他ELTサービスを使う or AppFlow→S3→Cloud Storage→BigQueryに渡す or GCP内で独自で実装する
評価	○	△	○

データソース

- 連携方法: ノーコードで連携が可能か？

Amazon Redshift	Snowflake	Google BigQuery
No 一部はAppFlowを用いてワンポチ連携してS3にロードして対応できるがそれ以外はサードパーティのETL製品を利用	No サードパーティのETL製品を利用	結論: No Googleネイティブのものはノーコードは可能だが権限制御は必要 サードパーティのETL製品を利用
△	△	△

アプリケーションからの操作

- API(またはそれに準ずるもの)が用意されているか？

	Amazon Redshift	Snowflake	Google BigQuery
用意されているか	DataAPIが用意されている	Snowflake SQL APIが用意されている	BigQuery APIが用意されている
認証方法	Secrets managerで接続 passwordを持つ必要がない	password + ネットワークポリシーによる IP制限が可能	クレデンシャルを利用する or GCP認証を利用する
評価	◎	○	◎

それぞれAPIの用意があるが、使い勝手が良いのがAmazon or Googleという結果に

アプリケーションからの操作

- アプリケーションがDWHに対してどのような方法でテーブル/レコードの操作ができるか？

アプリケーション	Amazon Redshift	Snowflake	Google BigQuery
Lambda	DataAPIを利用する	Snowflake SQL API or Lambdaに対してコネクタを含めたDocker イメージを利用する	BigQuery APIを利用する
Glue	Glueの機能として提供している (内部的にはjdbc接続でSQLを発行する)	JDBCドライバー or Sparkコネクタを利用しての接続	AWS Glueカスタムコネクタを利用 (Marketplaceにてサブスクライブ)
評価	○	○	○

アクセスコントロール

- カラムレベルに対してユーザー・グループ単位のアクセスコントロールの仕組みがあるか？

Amazon Redshift	Snowflake	Google BigQuery
SQL GRANT文で可能	GRANT文で可能 ダイナミックデータマスキングを利用することでレコードレベルでの制御が可能	ポリシーで制御を行う
○	◎	○

Snowflakeだけレコード単位でのマスキング(ダイナミックデータマスキング)がある

セキュリティ

	Amazon Redshift	Snowflake	Google BigQuery
第三者認証	取得済 SOC、PCI、FedRAMP、HIPAA など	取得済 SOC、PCI、FedRAMP、HIPAA など	取得済 SOC、PCI、FedRAMP、HIPAA など
IP制限	SecurityGroupで制限	ネットワークポリシーを利用	IAM & VPC Service Controls
持ち出し制限	Workspacesでの持ち出し制限が可能	Workspacesでの持ち出し制限が可能	WorkSpacesでの持ち出し制限が可能
評価	○	○	○

AWS⇄GCP間がインターネットに出ないようにするためには一工夫必要な点が気になった

レスポンス

現在利用しているAthenaとの比較を実施

RedshiftはRA3.xplusで行い、チューニング、AQUAなしで実施

実行クエリ	Athena(開発環境)	Amazon Redshift	Snowflake	Google BigQuery
既存の本番環境で10分程度かかるクエリ	123.82 秒 Data Scanned 444.424MB	80 ミリ秒 Data Scanned 212.67 KB	XSサイズ: 8.58 秒 Sサイズ: 5.04 秒 Mサイズ: 4.36 秒 Lサイズ: 3.11 秒 Data Scanned: 790MB	3.2 秒 Data Scanned 6.5 GB
既存の本番環境で20分程度かかるクエリ	83.18 秒 Data Scanned 1.09 GB	9.336 秒 Data Scanned 1.5 GB	XSサイズ: 7.0 秒 Sサイズ: 4.5 秒 Mサイズ: 3.44 秒 Lサイズ: 2.59 秒 Data Scanned: 1.5 GB	3.5 秒 Data Scanned 3.3 GB
既存の本番環境で30分程度かかるクエリ	189.94 秒 Data Scanned 1.52 GB	23.897 秒 Data Scanned 12.07 GB	XSサイズ: 11.06 秒 Sサイズ: 6.68 秒 Mサイズ: 5.29 秒 Lサイズ: 4.4 秒 Data Scanned: 1.0 GB	4.3 秒 Data Scanned 4.5 GB
評価	-	○	○	○

スケーラビリティ

- 負荷が増えた時にスケールアップ、スケールアウトできること
- データ量が増えた時にスケールアップ、スケールアウトができること

	Amazon Redshift	Snowflake	Google BigQuery
負荷上昇時	インスタンスサイズをあげてスケールアップ インスタンス数をあげてスケールアウト ワークロード管理 (WLM)機能 AQUAを有効化	インスタンスサイズをあげてスケールアップ インスタンス数をあげてスケールアウト	裏側でオートスケールする チューニング方法はない 頭打ちになった場合は別料金で対応
データ増加時	インスタンスサイズをあげてスケールアップ インスタンス数をあげてスケールアウト	インスタンスサイズをあげてスケールアップ インスタンス数をあげてスケールアウト	裏側でオートスケールする チューニング方法はない 頭打ちになった場合は別料金で対応
評価	◎	○	○

Redshiftではスケール戦略以外にもメンテナンス機能があるのが特徴

コスト(前提)

3製品同一条件で見積もりを行う

- 130TBのスキャン量
- 1.4TBのストレージ
- ミニマム構成で検討

コスト

コスト	Amazon Redshift	Snowflake	Google BigQuery
合計(月額)	**$1307.74~$1945.04**	**$2021.416667**	**$824.8~$13324.8**
コンピューティング	ra3.xlplus 2ノード $1868.80 RI購入時: $1231.5	クラスタ$1621.65 サーバレス$373.825	130TBスキャン $6 x 130 = $780
ストレージ 1.4TBと仮定	$36.54	$25.92	1.4TB全てActive storageと仮定 $0.023 x 1400 = $32.2
初回データ転送	r5.xlarge: マルチAZ: 7.94USD * 5hの稼働と仮定 $39.7	無料	バッチ処理の場合、S3側のみ課金 $0.086 x 1400GB = $120.4
継続データ転送	無料	無料	バッチ処理の場合、S3側のみ課金 $0.086/GB リアルタイム転送は別途課金
監査ログ	S3に出力されるため軽微	無料	$0.525 x 25 = $12.5
サポートFee	加入済みのため考えない	サービス使用料に含まれる	ベーシック $0 スタンダード $29（月額）+合計費用の 3% エンハンスト $500（月額）+合計費用の 3% プレミアム $12,500 （月額）+合計費用の 4% （1 年間の最小コミットメントあり）

Pros

- Redshift
  - 既存の環境とのフィット具合
  - AWSの新機能を取り入れやすい環境
  - RIがあり、時間単位での課金なので見積もりが容易
- Snowflake
  - 性能をワンクリックで上げられる
  - 基盤自体のメンテはほぼ不要
- BigQuery
  - 基盤自体のメンテはほぼ不要
  - Google周りの連携(Firebase, Analyticsなど)がシームレス

Cons

- Redshift
  - 基盤の運用をチームで負担することになる
- Snowflake
  - 年間契約がクレジットの概念なので足りなくなったときの追加などの処理が煩雑
  - SaaSの管理が必要
  - サポート品質への将来的な不安
- BigQuery
  - アカウント管理を一から作成にすることになるので導入までに時間がかかる
  - クラウドを跨ぐので不具合の調査が大変
  - チューニングしたくなっても金で殴るしか解決方法がない

Redshiftに決定！

決め手

1. 既存の環境とのFit

2. コスト予測性の容易さ

3. Amazon Redshift機能開発の将来性

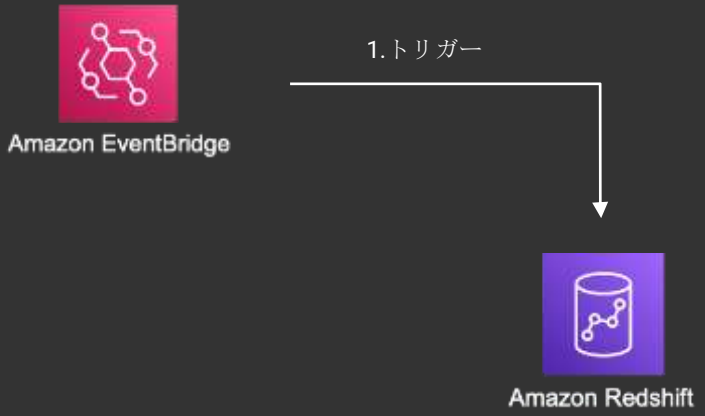
4. 安定したAWS Supportの品質

実際のアーキテクチャ

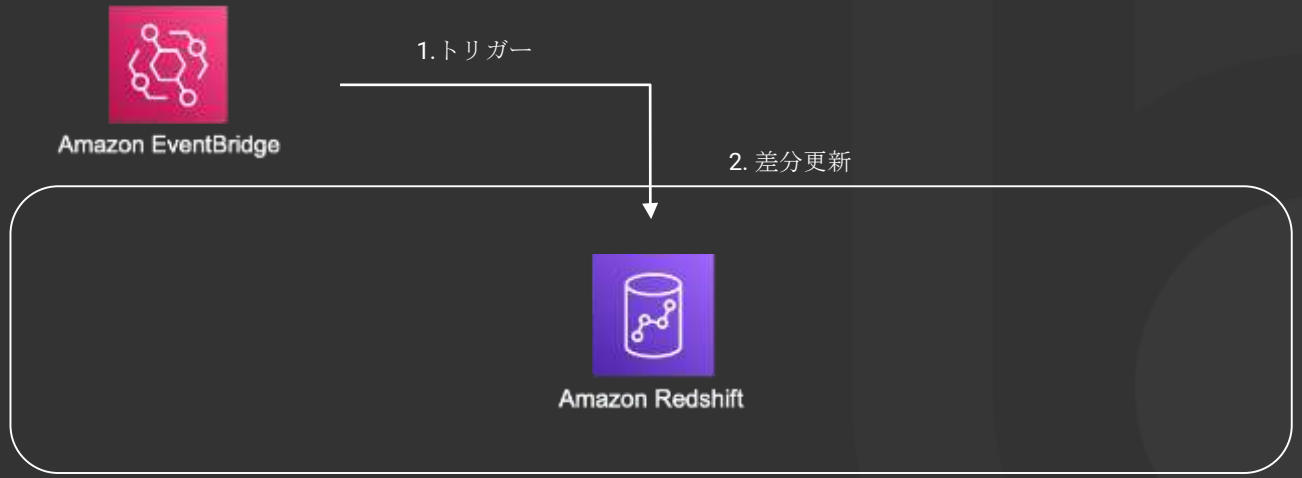
アーキテクチャ図(初回ロード)

アーキテクチャ図(初回ロード)

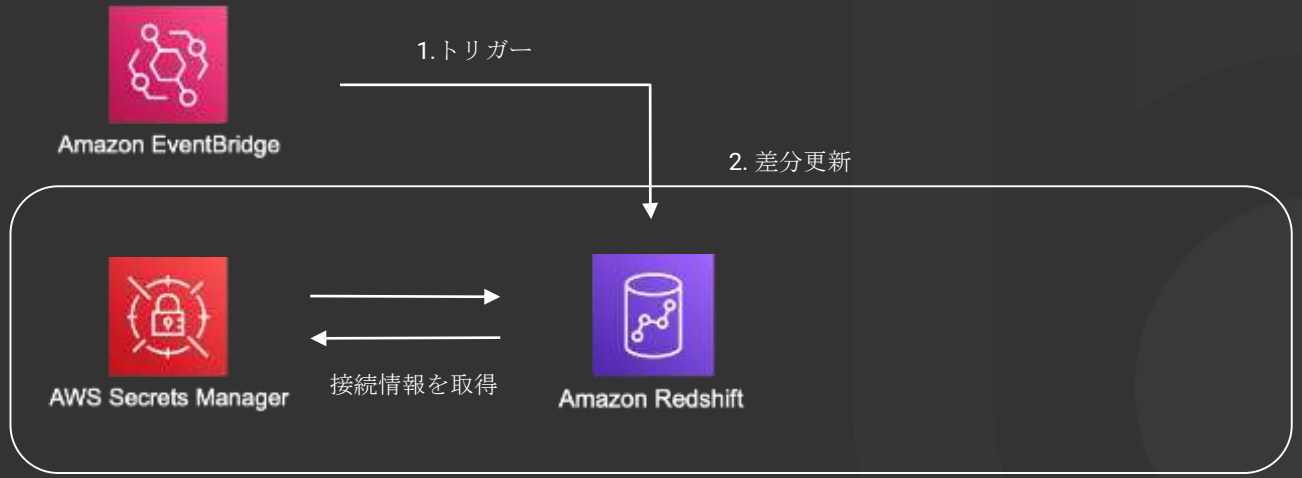
アーキテクチャ図(継続的ロード)



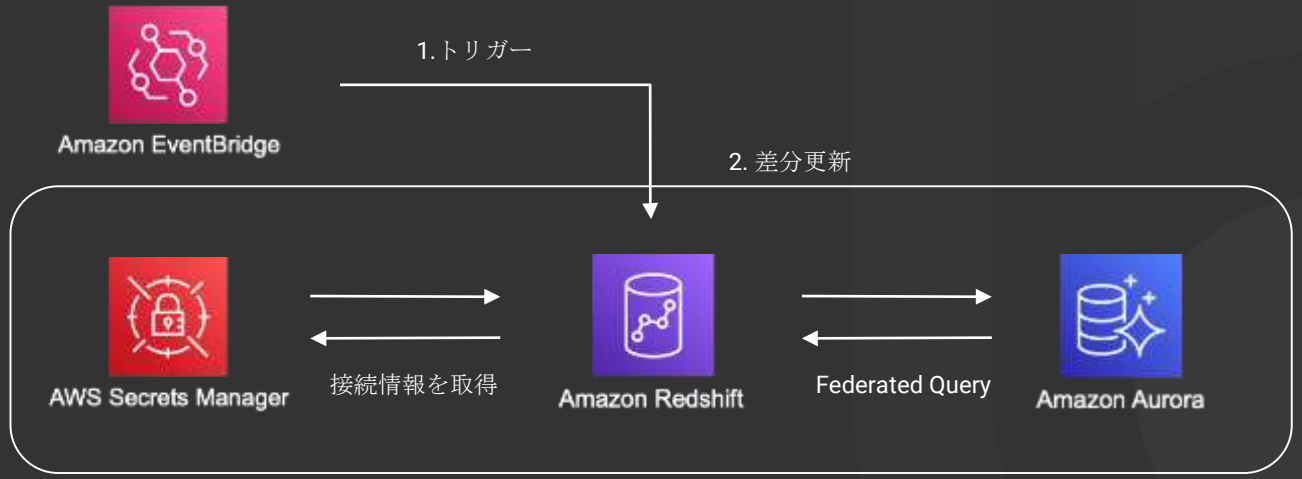
アーキテクチャ図(継続的ロード)



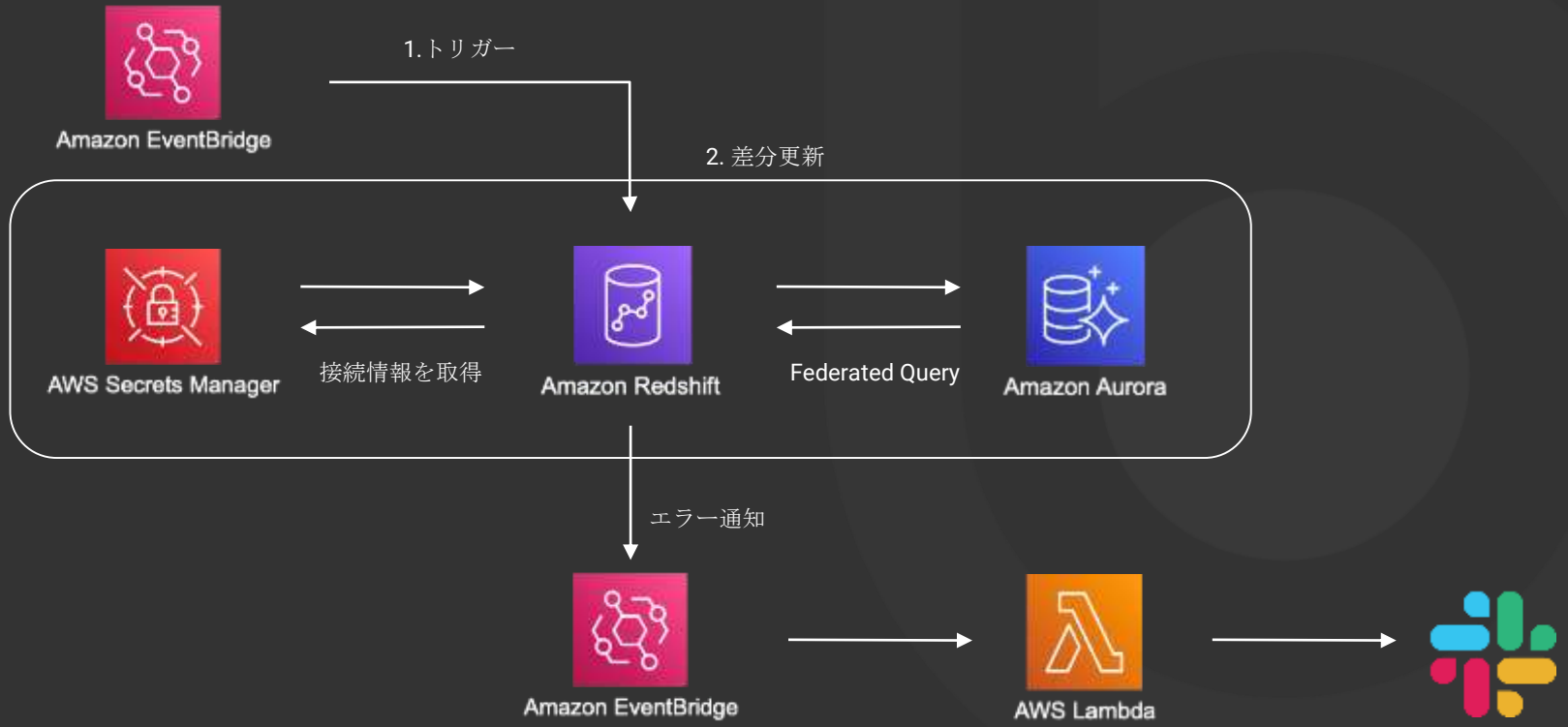
アーキテクチャ図(継続的ロード)



アーキテクチャ図(継続的ロード)



アーキテクチャ図(継続的ロード)



なんでこのアーキテクチャにしたか？

- 将来的にAuroraのデータをパージする計画がある
- 運用をできるだけ最小限にしたかった
- 本番アプリケーションに影響を出ない形にしつつ、DBには負荷をかけても良い構成としたかった

まとめ・所感

- 各社DWHでクリティカルに大きな機能の差はない
- サービスの有名度で入れる前に、各DWHの特性を踏まえながら要件・既存のアーキテクチャとの組み合わせでどの構成が合っているか？を一旦立ち帰って考える
- 料金や運用コストをどこまで許容するか？将来的にヘビーユーズされて増加していく上でチューニングの余地があるか？と照らし合わせながら考える
- 最新の機能が各社リリースされていくので、ベストプラクティスは常に時代と共に移り変わるものである

今回の要件

- データはAurora, S3に保存されている
- Auroraから必要なデータのみを取り出し、連携する
- BIツールはRedashで利用者はエンジニアに限らずビジネス部門も利用する
- 高いパフォーマンス
- 多様なデータソースサポートと柔軟なデータアクセス
- 個人情報取扱いのためのアクセスコントロール
- 監査ログなどのIT-GC/IT-AC含む金融セキュリティ、統制要件

基盤としては満たせるようになった！！

将来の展望

- データはAurora, S3に保存
- Auroraから必要なデー
- BIツールはRedashで

BIツールの更なる活用！

SageMakerとRedshiftとの統合！

通知基盤とRedshiftの統合！などなど...

ポートと柔軟なデータア

のアクセスコントロール

- 監査[illegible]とのIT-GC/IT-AC含む金融セキュリティ統制要件

基盤としては満たせるようになった！！

だけど活用においてまだまだやることがある・・・！！！


We Are Hiring!

ビットバンク 採用

検索



ビットコインの技術で
世界中にあらゆる価値を流通させる