

# LIXIL が考える、 異種間データベースマ イグレーションを成功さ せる秘訣とは

Google  
Cloud  
Next

Tokyo

Proprietary

**菅野 寛史**

**出井 彰記**

**志村 遼太**

**LIXIL**



# Index

- 01 LIXILについて
- 02 Oracle Database の移行  
複数システムにおける戦略
- 03 MS SQL Server の移行  
提案見積システムの戦略
- 04 IBM Db2 の移行  
生産管理システムの戦略

# 01. LIXIL について



# LIXIL

世界 150 カ国以上で約 53,000 人の  
従業員を擁するグローバル企業。  
毎日 10 億人以上の人びとが LIXIL の  
製品を利用しています

<https://www.lixil.com/jp/about/about.html>



# LIXIL について

- 2011 年に国内の主要な建材・設備機器メーカー、トステム、INAX、新日軽、サンウエーブ工業、東洋エクステリアの 5 社が統合して誕生
- GROHE や American Standard といった世界的ブランドを統合し、グローバルに事業を展開
- 日本のものづくりの伝統を礎に、世界をリードする技術やイノベーションで、日々の暮らしの課題を解決する高品質な製品を提供している

## LIXIL の Purpose (存在意義) を 追求し、先進的な水まわり製品と 建材製品を開発・提供



毎日 **10 億**  
人  
以上の生活を  
支える



世界 **150** カ国  
以上で  
事業を展開

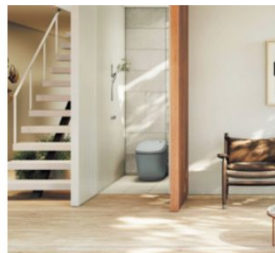


**100** 年以上の  
ブランドの歴史

# LIXILの事業

<https://www.lixil.com/jp/business/>

## ウォーターテクノロジー事業 (LWT)



売上収益



事業利益<sup>\*1</sup> 369億円  
従業員数<sup>\*2</sup> 27,151人  
研究開発費 164億円

\*1 日本基準における営業利益に相当  
\*2 非正規雇用者を除く

### 主要製品およびサービス

#### 水まわり設備

衛生機器、シャワートイレ(温水洗浄便座)、水栓金具、浴槽、ユニットバス、スマート製品、シャワー

#### タイル建材類

住宅・ビル外装タイル、内装タイル等

### 製品ブランド

・グローバルブランド

American  
Standard

GROHE

INAX

LIXIL

・スペシャリティブランド

COBRA

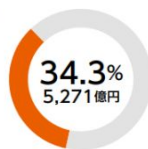
DXV

SATO

## ハウジングテクノロジー事業 (LHT)



売上収益



事業利益<sup>\*1</sup> 260億円  
従業員数<sup>\*2</sup> 16,249人  
研究開発費 63億円

\*1 日本基準における営業利益に相当  
\*2 非正規雇用者を除く

### 主要製品およびサービス

#### 金属製建材

住宅・ビル・店舗用サッシ、玄関ドア、各種シャッター、門扉、カーポート、手摺、高欄、カーテンウォール等

#### その他建材類

窓枠、造作材、サイディング、屋根材等

#### 住宅・サービス関連

工務店のフランチャイズチェーンの展開、建築請負、不動産管理、不動産事業のフランチャイズチェーンの展開支援、住宅ローン等

#### その他

太陽光発電システム、スマート製品等

### 製品ブランド

・グローバルブランド

LIXIL TOSTEM

・スペシャリティブランド

NODEA

・カテゴリブランド

AT

EXSIOR

SUPER WALL

・グループ会社ブランド

アイフルホーム

フィアスホーム

GLホーム



## リビング事業 (Living)



売上収益



事業利益<sup>\*1</sup> 72億円  
従業員数<sup>\*2</sup> 4,029人  
研究開発費 26億円

\*1 日本基準における営業利益に相当  
\*2 非正規雇用者を除く

### 主要製品およびサービス

#### 水まわり設備

システムキッチン、セクショナルキッチン、洗面化粧台、洗面カウンター等

#### インテリア建材

室内ドア・引き戸、室内用窓、床材、カウンター・階段、収納等

### 製品ブランド

・グローバルブランド

INAX

LIXIL

# LIXIL のクラウドシフト

Google Cloud を活用して多くの事業を推進してきました。そして新たなフェーズとして次の戦略に注力します

## 1. データベース運用の効率化

レガシーなデータベース運用を刷新し、Google Cloud のマネージドサービスを積極的に導入しています。迅速かつ正確なバックアップ・リストア、クラウドネイティブな監視を実現し、モダンで効率的な DB運用へと転換を図っています。

## 2. 基幹システムの戦略的クラウド移行

重要かつ規模の大きい提案見積・生産管理システムについてもクラウドシフトを進めています。複雑な既存システムをクラウドネイティブなアーキテクチャへと移行する高難易度のマイグレーションです。この実績を通じたノウハウは、全社的なクラウドシフトの横展開において重要な礎となります。

## 02. Oracle Database の移行

# プロジェクトについて



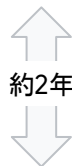
## 移行の背景

- 老朽化
  - 仮想基盤
  - サーバOS
- クラウドシフト戦略
  - OSS系の推進
  - 費用・運用の削減
  - Google Cloudとの親和性



## スケジュール

- 方針出し: 2021.1
- 体制発足: 2023.10
- DMS検証: 2024.1
- 移行開始: 2024.4
- 移行完了: 2025.8(予定)



約2年



## 移行したシステム

- 74システム (サーバ: 222)
- Version
  - Oracle Database 8i
  - Oracle Database 9i
  - Oracle Database 10g
  - Oracle Database 11g

# DBサービスの評価

Cloud SQLやAlloyDBでは...

## 運用負荷の軽減

- バックアップ、リストア、モニタリングといった運用の工数を大幅に削減できる
- DataStream を用いることで、リアルタイムでデータ分析基盤と連携ができる

## 可用性の向上

- オンプレミス環境と比較して、マルチゾーンやマルチリージョン構成を容易に採用できる
- Fault Injection を実行させ、復元するまでのプロセスをシミュレート可能

## コスト構造の最適化

- 適切なデータベースの採用によりライセンス費用の削減

# プロジェクトの推進

## 1. 各システムに対してドキュメントを公開

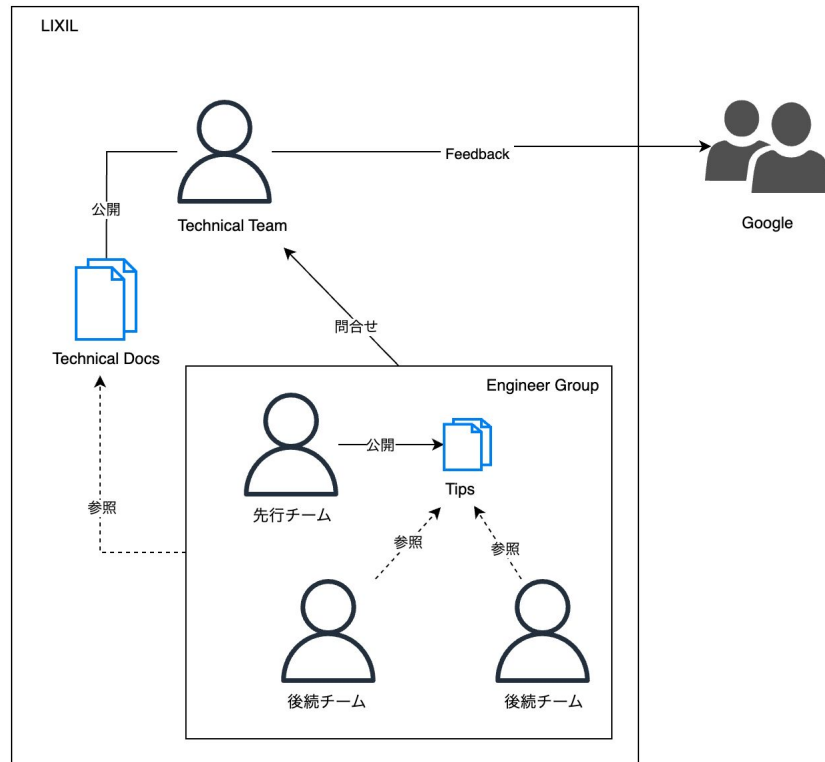
- 移行手順
- DB サービスの選定基準
- DB サービスのガイドライン
- よくある問合せ

## 2. 先行チームから後続にTipsを共有

- Data Validation Tool の検証結果
- Cloud SQL のデータフラグの利用例
- 組み込み関数の修正例

## 3. Googleに移行課題を相談

- DB エキスパートが複数名参加
- ガイドラインに記載する内容について相談
- DB サービスのアップデートをLIXILに共有



# 主なドキュメント

## 移行ツールの利用手順

- Database Migration Service
- Ora2Pg
- Data Validation Tool

## Database Migration Service (DMS)

- データ移行サービス
  - MySQL・PostgreSQL・SQL Server 同士
  - **Oracle → PostgreSQL**
  - SQL Server → PostgreSQL ...**New!!**
- 注意点
  - Oracle 11g 以降がサポート対象
  - LIXIL ではそれ以前の Ver は Ora2Pg を利用

## Database Migration Service 使用手順書

LIXIL

### 目次

- はじめに
  - 1.1 本書について
  - 1.2 目的
  - 1.3 用語の定義
- 前提
  - 2.1 Google Cloud リソース
  - 2.2 移行元 Oracle Database のバージョン
  - 2.3 ダウンタイム
  - 2.4 手順書の記載範囲
- Oracle の設定手順(LIXIL機記載予定)
- Cloud SQL の設定手順
  - 4.1 推奨される Cloud SQL の構成
  - 4.2 移行先データベース作成手順
  - 4.3 ユーザーアカウント作成手順
  - 4.4 作成したユーザーへの権限付与手順  
参考
- 接続プロファイル作成手順
  - 5.1 Oracle 用接続プロファイル作成手順
  - 5.2 Cloud SQL 用接続プロファイル作成  
参考
- コンバージョンワークスペース使用手順
  - 6.1 コンバージョンワークスペース作成
  - 6.2 オブジェクト変換手順
  - 6.3 スキーマ移行のテスト手順
  - 6.4 スキーマ移行手順  
参考
- 移行ジョブ作成手順  
参考
- 移行ジョブ実施手順
  - 8.1 移行先スキーマに対する権限付与手
  - 8.2 移行ジョブ開始手順
  - 8.3 移行確認手順(LIXIL機記載予定)
  - 8.4 移行ジョブ終了手順(LIXIL機記載予定)
  - 8.5 ユーザーアカウント削除手順  
参考

### 6.コンバージョンワークスペース使用手順

コンバージョンワークスペースは、移行元データベースのスキーマとオブジェクトを、宛先データベースと互換性のために使用します。  
本章では、このコンバージョンワークスペースの使用手順を示します。

#### 6.1 コンバージョンワークスペース作成手順

1. 「コンバージョンワークスペース」を押下します。

データベースの移行	コンバージョンワークスペース	<input checked="" type="checkbox"/> ワークスペースをセットアップ
移行ジョブ	コンバージョンワークスペースでは、移行元のデータベースのスキーマとオブジェクトを、 タは変更されません。ワークスペースを使用すると、全体的なコンバージョン構成を確認。	
接続プロファイル	▼ ファイルタ	
プライベート接続	<input type="checkbox"/> コンバージョンワークスペースへ ↑ <input type="checkbox"/> コンバージョンワークスペースID	
	<input type="checkbox"/> ca-test-workspace	ca-test-workspace

2. 「ワークスペースをセットアップ」を押下します。

データベースの移行	コンバージョンワークスペース	<input checked="" type="checkbox"/> ワークスペースをセットアップ
移行ジョブ	コンバージョンワークスペースでは、移行元のデータベースのスキーマとオブジェクトを、 タは変更されません。ワークスペースを使用すると、全体的なコンバージョン構成を確認。	
接続プロファイル	▼ ファイルタ	
プライベート接続	<input type="checkbox"/> コンバージョンワークスペースへ ↑ <input type="checkbox"/> コンバージョンワークスペースID	
	<input type="checkbox"/> ca-test-workspace	ca-test-workspace

3. 「新しいワークスペースを設定」を押下します。

コンバージョンワークスペース

新しいワークスペースをセットアップ

新しいワークスペースのご紹介

新しいコンバージョンワークスペースでは、スキーマとコードの変換、互換性でのコード変更が可能です。変更先のデータベースに作業内容を簡単に適用できます。

新しいワークスペースを設定

新しいワークスペースを使用する準備が整っていない場合は、以前のワークスペースを

# 主なドキュメント

## DB ガイドライン

- 選定基準や構築方法などを記載。
- LIXIL における推奨 / 非推奨の設定や監視メトリクスについても言及。

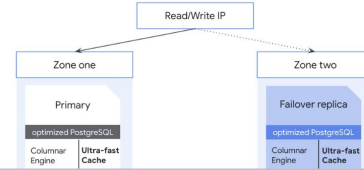
## DB ガイドラインの勘所

- 推奨・非推奨・禁止事項を明言し、構築のハードルを下げる。
- 監視に言及する。移行後に監視レベルが下がってしまうケースがよくあるが品質は下げない。
- Query Insights・リードレプリカなど慣れない機能は操作方法まで記載する。

### 1.3 DBテスト (Fault injection)

高可用性クラスターのAlloyDBはデータベース障害を自動的に検出し、60秒以内にフェイルオーバーをトリガーします。これはデータベースのサイズや負荷に依存しません。詳細は下記を参照ください。

[https://cloud.google.com/alloydb/docs/instance-primary-fallover#perform\\_a\\_manual\\_fallover](https://cloud.google.com/alloydb/docs/instance-primary-fallover#perform_a_manual_fallover)



### 3. データベースフラグの設定

フラグを構成することで、AlloyDBをカスタマイズできます。一部のフラグ更新では、影響を受けるインスタンス(プライマリ インスタンス、および場合によってはリードレプリカ)の再起動が必要になります。この場合、低ダウンタイムと強制適用という2つのメンテナンス ポリシーのいずれかを選択する必要があります。

- データベースフラグの設定の詳細については、「データベースフラグを設定する」を参照してください。  
<https://cloud.google.com/alloydb/docs/instance-configure-database-flags>

- サポートされているフラグは、「サポートされているフラグ」を参照してください。AlloyDB クラスターの作成時にフラグを設定でき、後でフラグを更新することもできます。  
<https://cloud.google.com/alloydb/docs/reference/database-flags>

それぞれのポリシーの概要です。

#### 低ダウンタイム

- デフォルトで有効なポリシー
- 実稼働環境で使用することを推奨
- プライマリ インスタンスの再起動が必要なフラグを更新すると
  - プライマリ インスタンスでは、ワークロードでダウンタイムが1秒未満になる
  - プライマリ インスタンスのフラグは、約15分後に更新を終了します
- フラグの更新にリードレプリカの再起動が必要な場合、次のような影響があります。
  - 読み取りプールのインスタンスのダウンタイムなし
  - 読み取りプールインスタンスのフラグは、約10分後に更新を終了します

#### 強制適用

- 開発環境などダウンタイムよりも生産性を重視する環境での実行が推奨
- CLI ( `gcloud betaloydbinstancesupdate` ) でのみ利用可能
- クラスターのプライマリ インスタンスで再起動が必要なフラグを更新すると、次のような影響があります。
  - プライマリ インスタンスでは約1分間のダウンタイムが発生します
  - プライマリ インスタンスのフラグは1~2分後に更新を終了します
- 読み取りプール インスタンスでの再起動が必要なフラグを更新すると、次のような影響があります。
  - 読み取りプール インスタンスには約1分間のダウンタイムが発生します。
  - 読み取りプール インスタンスのフラグは、1~2分後に更新を終了します。

#### 推奨事項

- フラグを更新する際は各ランドスケープごとに実行方法を検討する
  - 例) 本番環境では低ダウンタイムでの更新を行う
  - 例) 開発環境では強制適用による更新を行う

注意事項

高可用性の検証のため下記はインジェクションを意図的に行わないでください。  
<https://cloud.google.com/alloydb/docs/instance-fault-injection>

シングル構成のAlloyDBでは、実行を避けてください。

### 1.4 推奨の設定

#### 必須事項

- 本番環境のバックアップ

#### 推奨事項

- 本番環境の継続的パッチ適用
- 障害を想定してバックアップ
- 高可用性が要件にある場合は、高可用性クラスターを使用してください。

# 問合せの例

**Q PostgreSQL に移行したがパフォーマンスが悪い**

A 前方一致の LIKE を使用したクエリ・ロケール「ja\_JP.utf8」であれば INDEX が機能していない。Index に演算子クラス (bpchar\_pattern\_ops など) を適用してみる。もしくは、C ロケールを設定する。

**Q Ora2Pg で生成される DDL が正しくない**

A デフォルト設定は NUMBER→bigint。DEFAULT\_NUMERIC の項目を変更してください。

**Q \COPY による SJIS→UTF8 への変換でエラーになる**

A 対応していない特殊文字の可能性がある。PowerShell の Get-Content・Encoding オプションによる変換事例あり。

# 苦勞した点

- レガシーな Oracle Version の移行には Ora2Pg など代替ソリューションが必要であり、追加検証が発生した。
- クエリの変換などアプリ側の対応工数を減らすため、DMS のコンバージョン機能や社内の AI チームから提供された変換ツールを利用した。
- 可用性・セキュリティ・監視などマイグレーション後の品質を下げないように対策が必要であった。DR 戦略や Cloud Monitoring (主に Metrics) について見直しを行った。

# 成功要因

副次的にガイドラインによってBackup・セキュリティなど DB 周りのガバナンスを強化に繋がった

## 1. 技術ドキュメント・Tipsを共有

- 手順書・ガイドラインにより移行や構築のハードルを限りなく下げた
- 移行方法では DMS・Ora2Pg により広範囲の Oracle 移行に対応した
- 問合せの体制を整え、Google のサポートもありながら開発チームの課題を解決した

DMS が SQL Server → PostgreSQL にも対応したため、今後の計画にも応用

## 2. Google Cloud の移行ツールを採用

- 有料の移行ツールでは規模が大きい今回のケースでは採用が難しく、地道な移行方法ではスキルセットや工数的に展開が困難であった。
- 一方、DMS はドキュメントが豊富 & 無償であり、結果として社内で共通の方法で移行を遂行できた

# 03. MS SQL Server の移行

# LIXIL 提案見積システムについて

- 概要

LIXIL の複数商材の見積システム

- 利用ユーザー

ショールーム・営業、社外パートナー

- データボリューム (production)

合計 989 テーブル

データ総量 7TB

- リクエスト数

約 200~400 万 / 日

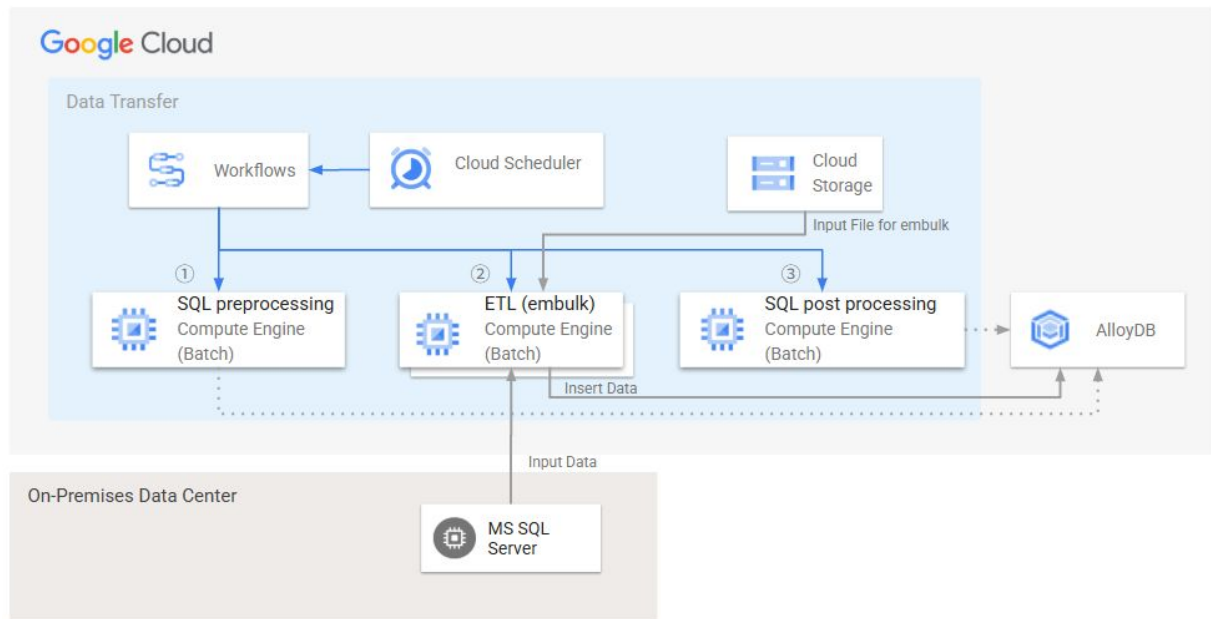
The screenshot displays the LIXIL proposal estimation system interface. At the top, there is a navigation bar with buttons for 'Home', '商品を選ぶ' (Select Product), 'プランを選ぶ' (Select Plan), '仕様を選ぶ' (Select Specification), '見積確認' (Check Estimate), and '見積完了' (Complete Estimate). A '合計金額' (Total Amount) field is visible on the right. Below the navigation bar, there is a breadcrumb trail: 'リシエル24/センターキッチン/アイランド型D97/274.8 cm (272.8 cm)'. The main area features a 3D rendering of a kitchen island with a sink and a range hood. Below the rendering are various configuration options for different components, including '基本仕様' (Basic Specification), '詳細仕様' (Detailed Specification), and 'オプション' (Options). The options are organized into a grid with categories like '色・取手' (Color/Handle), 'フックトップ' (Hook Top), 'シンク' (Sink), '水栓金具' (Faucet Hardware), '浄水器' (Water Purifier), '加熱機器' (Heating Device), '食器洗い乾燥機' (Dishwasher), 'レンジフード' (Range Hood), '収納プラン' (Storage Plan), 'キャビネット材' (Cabinet Material), 'コンロ前面ガラス' (Stove Front Glass), and 'シンク前面スクリーン' (Sink Front Screen). At the bottom, there are buttons for '初期のプランに戻す' (Return to Initial Plan), '勝手交換' (Change), 'フリー配置に進む' (Proceed to Free Configuration), and '見積確認に進む' (Proceed to Check Estimate). A note at the bottom states: '仕様を確認して、「見積確認に進む」を押してください。表示画像はイメージ画像のため、実際の商品と異なる場合があります。' (Please check the specifications and click 'Proceed to Check Estimate'. The displayed image is an image image, so it may differ from the actual product.)

# 移行前の課題について

- 老朽化
  - DB 起因の原因不明の障害が複数件発生
    - ユーザーが見積できない多大な被害を発生させてしまった
  - バージョンが古く、新技術に遅れを取る
    - 抜本的なパフォーマンス改善などユーザーに価値提供が難しい
- リソース変更
  - On-premises ではリソース変更に必要な期間が長い(申請・調整など)
    - 利用展開や障害復旧後の処理増加に対応した柔軟で安定した運用が難しい
- サービス時間
  - 365 日サービス稼働により移行に伴うサービス停止時間が限定される

# 移行アーキテクチャ

## Architecture: Database Migration



# 工夫した点・苦労した点

- 検討当時、SQLServer → PostgreSQL への DMS が未対応
  - 様々な SaaS を模索したがコストなどで断念
  - Embulk を業務利用していたため、採用
- 並列度を上げてデータ移行時間を短くする方法を検討
  - Workflows、Batch で並列実行させる仕組みを構築
  - 7TB のデータ約 1 日半で移行できるまで短縮
  - 毎日作成される 1 日分の同期データは 1 時間程度
- DBMS 違いによるクエリのパフォーマンスチューニング
  - ユースケースごとの自動テストシナリオを用意し、ベンチマークを取得
  - 影響が大きいシナリオで実行されるクエリを特定
  - 実行プランを確認しつつ、AI 活用でクエリ修正

# 工夫した点・苦労した点

- 本番切り替え時のサービス停止を極力少なく
  - ある時点でのデータを一括移行(約1日半)
  - 日次で差分データを移行(約1時間/日)
  - 本番切り替え時は当日差分データのみ移行(約1時間)



# 移行によるメリット

- DB インフラコスト **-43% 削減**
- システム分析用ダッシュボードによる**可視化**で利用状況把握が容易
  - 利用増加を検知し、柔軟なリソース変更が可能
- Query Insights のクエリのリアルタイム分析により、リリース後にも迅速に**パフォーマンス改善**が可能
- リードプールインスタンスによる**リアルタイムデータ利用**
  - リードプール参照の embulk を活用した柔軟なデータ移行基盤を構築
- 性能評価や負荷テストの**データ準備が容易**
  - 本番バックアップからリストアして、実施後はすぐに廃棄することで短期間にコストを抑えながら本番相当のデータでテストが可能

# 04. IBM Db2 の移行

# LIXIL 生産管理システムについて

- 概要

LIXIL の生産工場で利用される生産管理システム

- 利用拠点数、ユーザ数

約 30 拠点、5,400 人が利用

- データボリューム

約 500 テーブル、約 4TB のデータサイズ

- トランザクション数

1 日平均 2,000,000 アクセス

**LIXIL のものづくりを支えるクリティカルなシステムで安定した DB サービス提供が必須**

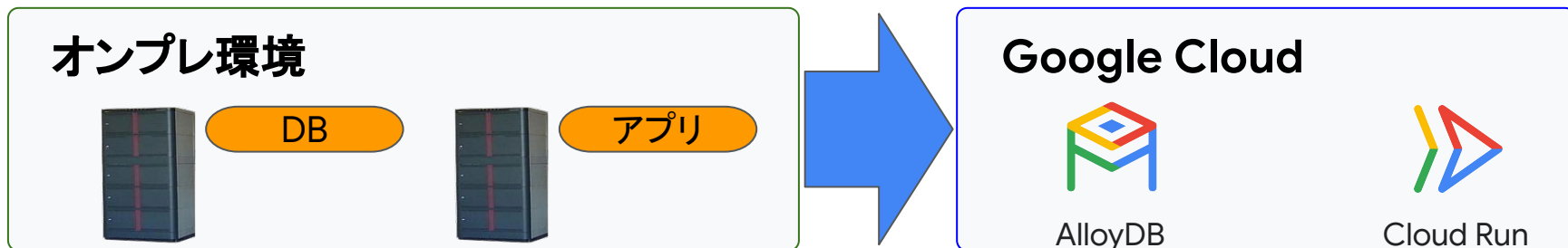
# 移行前の課題・移行方法

- サーバの老朽化(システム導入から15年以上経過)
  - DB 障害が定期発生
  - OS 制約等、最新ツールが採用できない

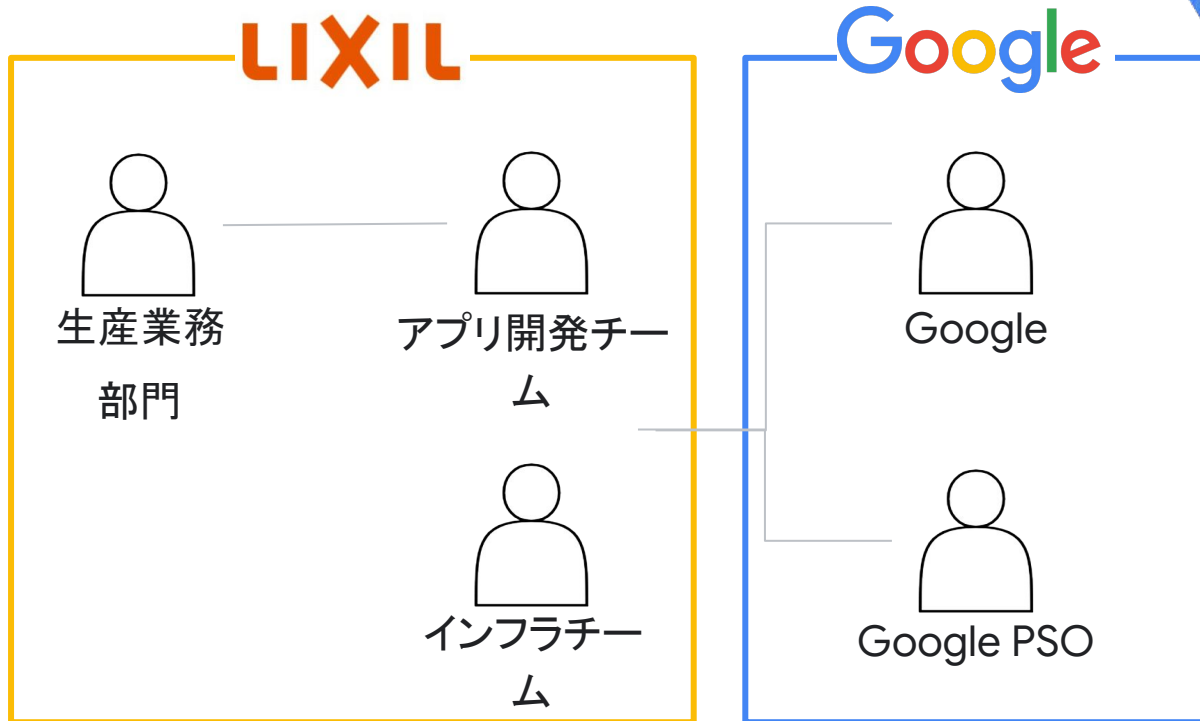
⇒DB サーバを AlloyDB、App サーバを CloudRun に刷新

- IBM Db2 有識者の減少
  - 社内に IBM Db2 利用者が少ない

⇒社内で採用事例がある PostgreSQL ベースに



# 体制



# 苦勞した点・工夫した点

- トランザクションスコープ変化  
エラーハンドリングロジックの改修 (約 300 箇所)
- 数値の丸め誤差  
主要データの新旧コンペア検証
- ロック・分離レベル  
システム処理の挙動の差異を事前検証

**事前検証で検出してリソース調達したことで納期通りリリース**

# 苦労した点・工夫した点（データ移行）

- 限られた時間での移行
  - データを早く抜き、早く入れることに注力
    - PSO の支援  
IBM Db2 ⇒ PostgreSQL への移行ツール提案  
移行方法の妥当性確認
    - 業務調整  
データサイズが大きなマスタを事前移行
    - 多重で Export - Import  
多重度をあげることによるDB リソースへの影響確認

# 移行によるメリット

- インフラコスト **-53% 削減**
- DB バックアップをとるための**サービス停止時間が 0**  
オンラインバックアップが可能になり、  
サービス停止なしでデータの静止断面が取得できるようになりました
- DB バックアップからの**復元が容易**  
性能検証用のテスト断面を一度作成して再利用することで、データの準備が不要になり、効率が大幅に向上
- レプリケーション**速度が早い**  
処理量が多く、負荷が高いタイミングでも数十分以上のラグなく同期
- BigQuery への**データ連携が容易**  
ツールの選択肢 増加(DataStream を利用)



# まとめ

- LIXIL はクラウドシフト戦略の中で、Oracle を Google Cloud (Cloud SQL・AlloyDB) へ大規模に移行しました。この移行では、Database Migration Service を活用した**共通の手順**が、**スムーズな展開の鍵**となりました。
- さらに、SQL Server や Db2 といった他のデータベースシステムについてもGoogle Cloud への移行を成功させています。特に、**データ移行時間の短縮に工夫**を凝らすことで、重要なシステムにおいてもユーザーへの影響を最小限に抑えることができました。
- 今回の Google Cloud への移行により、LIXIL は**コスト面と運用面**の両方で大きなメリットを享受しています。