

アプリケーションを内製化する際のデータベース選択指針

Google Cloud

データベース スペシャリスト

佐藤 貴彦



01

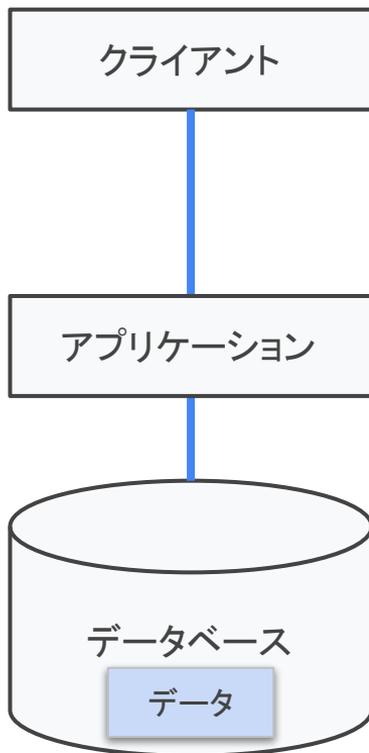
内製化で考えなければならない データベースの要素

システムの要: アプリあるところに DB あり

システムで使うデータベースは何を使えばいいんだろう？

使いたい機能を持ったデータベースを選ぶ？

機能以外ではどんな基準で考えればいいのだろう？



システムに求められる様々な非機能要件

可用性

システムの継続、耐障害性、災害対策などの要件

性能と拡張性

通常時及びピーク時のレイテンシやスループット、リソースの拡張性などの要件

運用と保守性

バックアップ、メンテナンス、障害時運用などの要件

セキュリティ

認証、認可、監査、暗号化などの要件

データベースで考える様々な非機能要件

可用性

DB の稼働率、RTO、RPO など

性能と拡張性

DB のスケールアウト、ストレージの拡張など

運用と保守性

DB の運用監視、バックアップ、メンテナンス、障害時運用など

セキュリティ

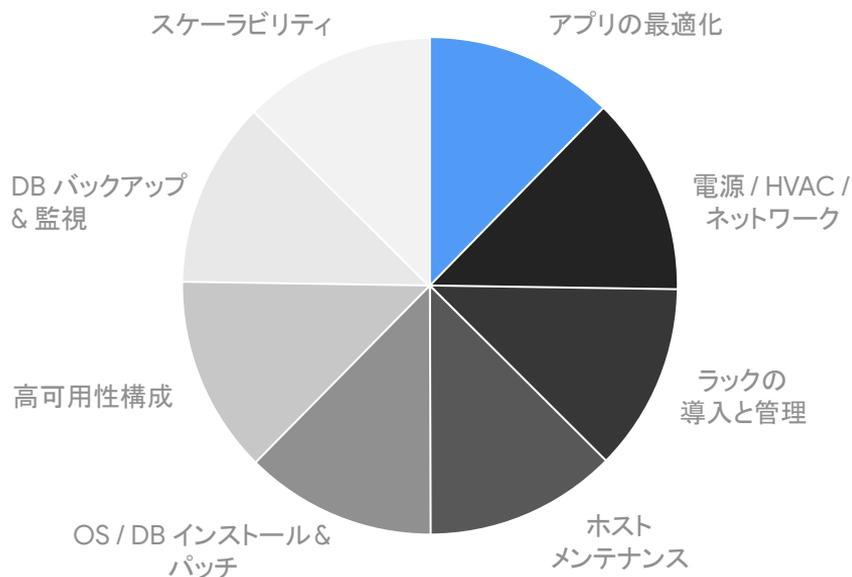
DB アクセスに対する認証、認可、監査、及びデータの暗号化など

DB では考えないといけない
ことが結構多いんです。

データベース

非機能要件を満たすために膨れる運用負担

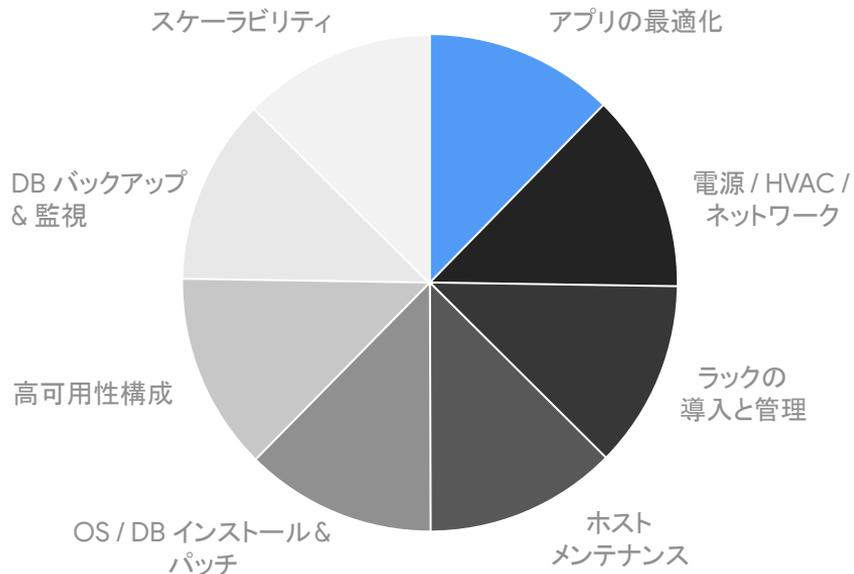
典型的なデータベース管理



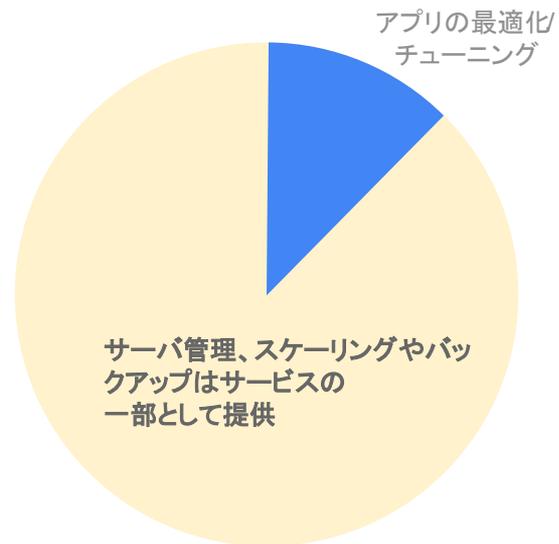
- 非機能要件を満たすには、データベースサーバーに様々なものが求められる
- オンプレミス環境では、各ハードウェアや、DB サーバー上の OS など、様々な運用負担が生じる
- 特に高可用性を実現するには、上記全てについて冗長構成を構築し適切に運用が必要

マネージド DB に全部任せてしまいましょう！

典型的なデータベース管理



マネージド DB を用いたデータベース管理



02

Google Cloud の マネージド データベース の全体解説

Google Cloud におけるマネージド データベースの選択肢

ユーザー数や経験者も非常に多い OSS の DB や商用 DB を
マネージド サービス化

クラウド環境用に設計された
Google Cloud 独自の DB のこと

OSS 及び商用のマネージド DB



Memorystore
for Redis
for Memcached

キャッシュ



Cloud SQL
for PostgreSQL
for MySQL
for SQL Server

リレーショナルデータベース (SQL データベース)



AlloyDB
Preview
for PostgreSQL
(完全互換)



Cloud Spanner
PostgreSQL
SQL 方言のみ互換



Cloud Bigtable
KVS / ワイドカラム

NoSQL データベース



Firestore
ネイティブモード
Datastore モード

ドキュメント DB



BigQuery

DWH

サーバーレス



mongoDB.



OSS パートナー
マネージド サービス

DB と一緒に使う周辺サービス

データベース移行サービス



Database Migration Service

サーバーレス CDC サービス



Datastream

アプリの裏で利用する Google Cloud のデータベース サービス

DATABASES		
	Bigtable	
	Datastore	 
	Database Migration	 
	Firestore	 
	Memorystore	 
	Spanner	
	SQL	
	AlloyDB for Postgr...	 

Google Cloud では、アプリの要件に合わせて使える
6 種類のマネージド DB を用意。


Cloud SQL
for MySQL
for PostgreSQL
for SQL Server

New!

AlloyDB
Preview
for PostgreSQL
(完全互換)


Cloud Spanner

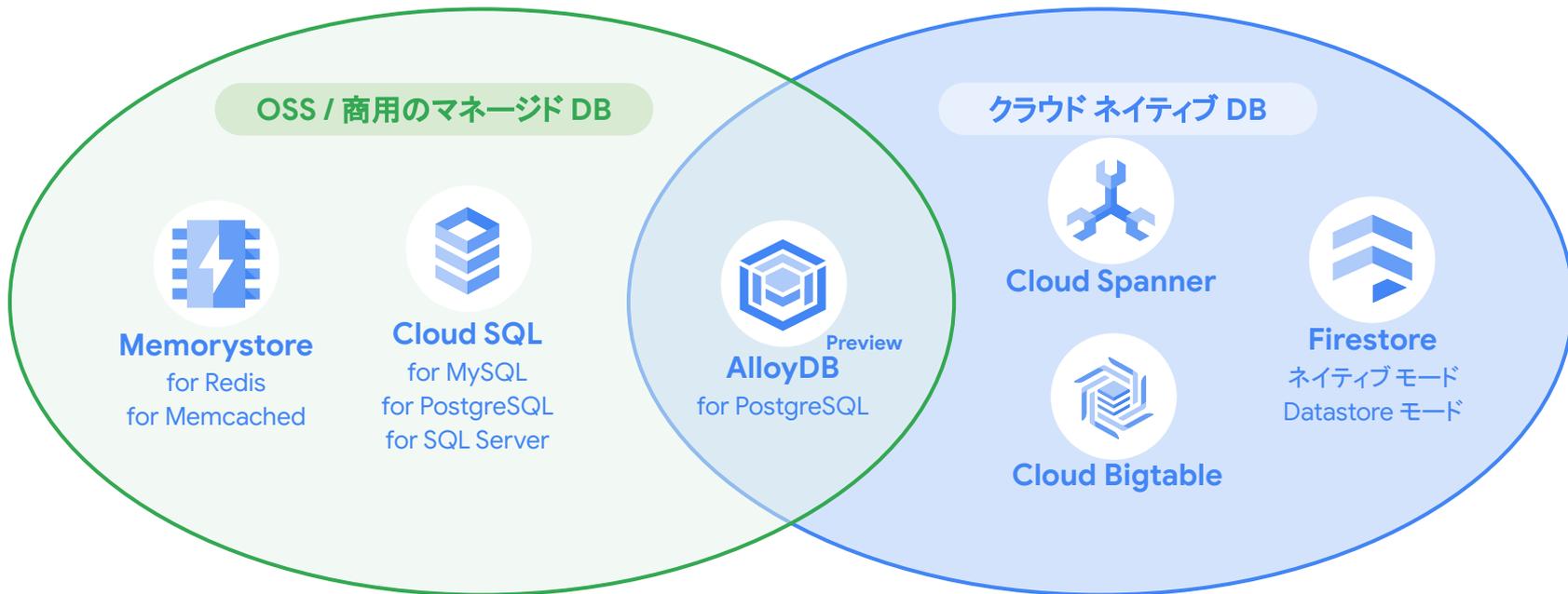
 BigQuery は
データ分析系の
サービスなので
別枠です


Memorystore
for Redis
for Memcached


Firestore
ネイティブ モード
Datastore モード


Cloud Bigtable

まずは 2 種類に大分類



OSS / 商用 DB をマネージド サービスとして提供



Memorystore
for Redis
for Memcached



Cloud SQL
for MySQL
for PostgreSQL
for SQL Server



AlloyDB
Preview
for PostgreSQL

ユーザー数や経験者も非常に多い OSS や商用 DB を、運用が簡単なマネージドサービスとして提供。



Cloud SQL for MySQL のインスタンス

マネージド サービスになっているだけで、中身は普通に OS の上で動く RDBMS が動作している。

Memorystore キャッシュ

インメモリ データストアである **Redis** と **Memcached** をマネージド サービスとして提供。

Cloud SQL リレーショナルDB

RDBMS である **MySQL**、**PostgreSQL**、**SQL Server** をマネージド サービスとして提供。リードレプリカや HA 構成の構築や自動管理、ストレージ容量の無停止自動拡張、自動バックアップなど、運用を容易にする様々な機能を提供。

Preview AlloyDB リレーショナルDB

RDBMS である **PostgreSQL** を、ストレージ層などを強化し、マネージド サービスとして提供。Cloud SQL が持つ運用容易性に加え、書き込み性能向上、低遅延のレプリケーション、可用性の向上など、様々な強化が行われている。**現在プレビュー版として提供。**

クラウド環境に合わせて設計開発されたマネージド DB



Cloud Spanner



Cloud Bigtable



Firestore
ネイティブモード
Datastore モード

クラウドネイティブ DB は、クラウド上の複数ゾーンやリージョンを活用した**高可用性**と、クラウドの柔軟なリソースを活用した**スケーラビリティ**を備えている。アップグレードやメンテナンスも完全無停止で行われる。



AlloyDB は、PostgreSQL をクラウドネイティブ DB として提供した、両方の特性を持つデータベース。

Cloud Spanner リレーショナルDB

高可用性、スケーラビリティ、運用容易性、SQL のサポートなどを備えた、フルマネージド RDBMS。

Cloud Bigtable KVS / ワイドカラム

NoSQL の一種である**ワイドカラムストア**を提供。高スループットと低レイテンシな読み書きが可能。Apache HBase との互換性もあり。

Firestore ドキュメントDB サーバーレス

(ネイティブモード / Datastore モード)

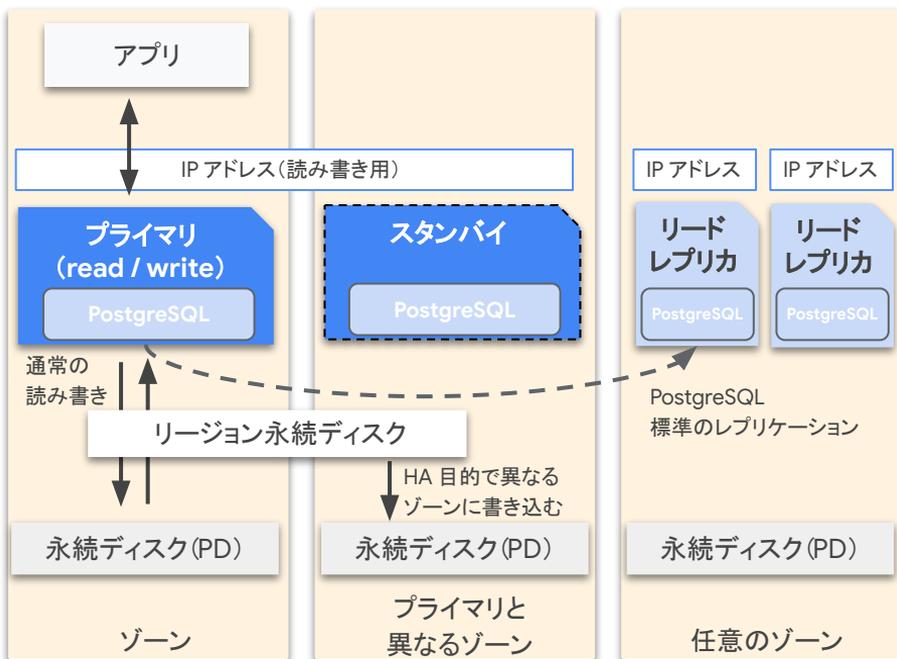
NoSQL の一種である**ドキュメントストア**をサーバーレス DB として提供。Firebase との親和性が高く、モバイルアプリから直接接続が可能であり、サーバーサイド要らずで開発を高速化。

マネージド DB である Cloud SQL のアーキテクチャ



Cloud SQL

Cloud SQL はデータベース自体は各種 OSS や商用 DB であって、それを Google Cloud のインフラに乗せたマネージド サービス。



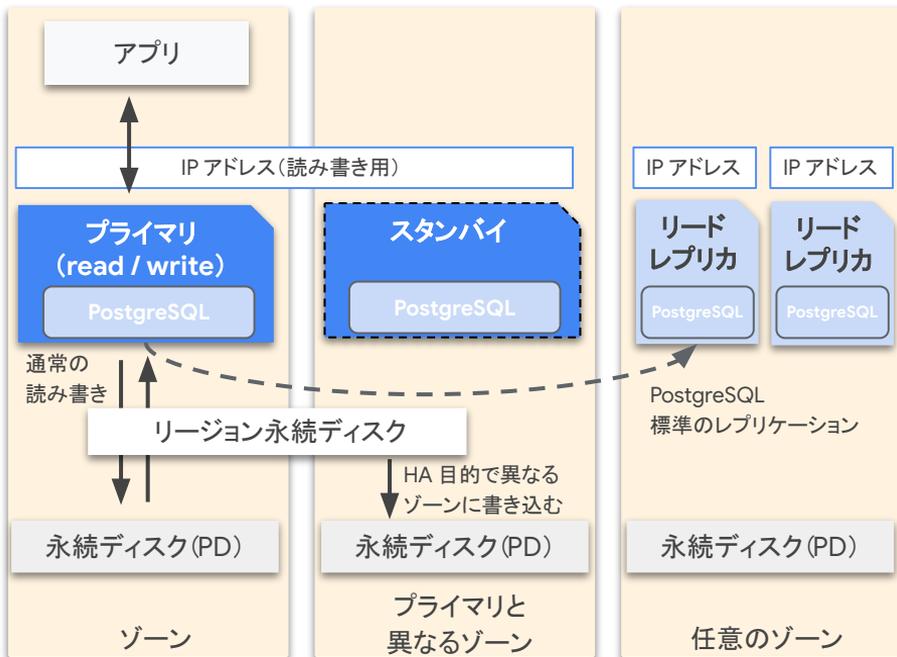
- 基本的には各種 DB が、そのまま VM 上で動いている。
- 例えばネットワークやディスク、フロントにあるロードバランサなどのインフラについては、Google Cloud が提供しているものの恩恵に預かることができる。
- 一般にデータベースは、書き込み性能がボトルネックになりがちであり、それはマネージド DB であっても変わらない。

IO 性能を向上させた AlloyDB のアーキテクチャ



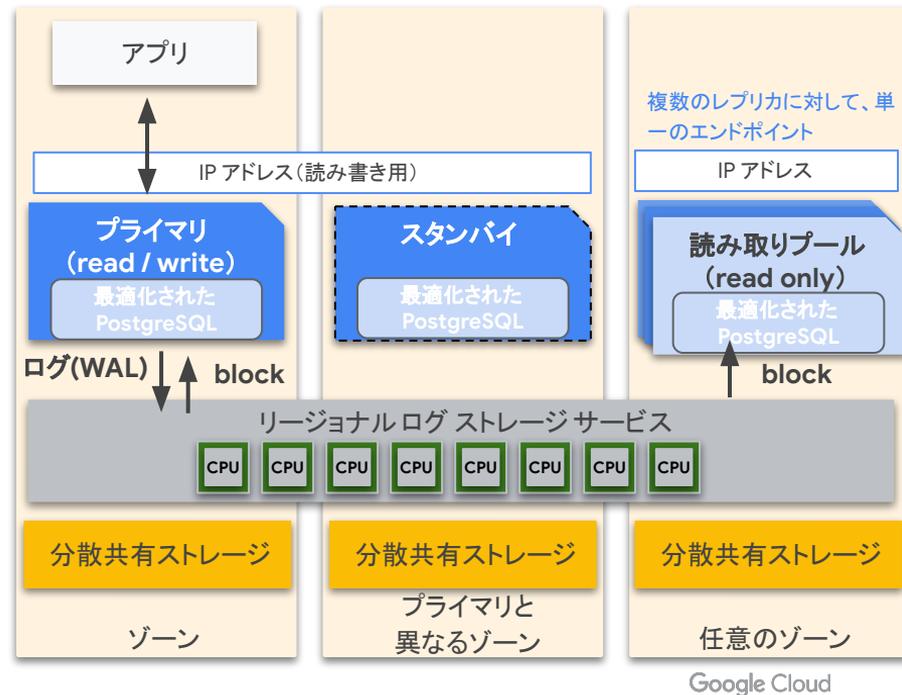
Cloud SQL

Cloud SQL for PostgreSQL とは、データベース自体は PostgreSQL であり、それを Google Cloud のインフラに乗せたマネージド サービス。



AlloyDB

AlloyDB は専用のログストレージ サービスを導入することで、書き込み性能、レプリケーションラグ、可用性を大幅に向上させている。

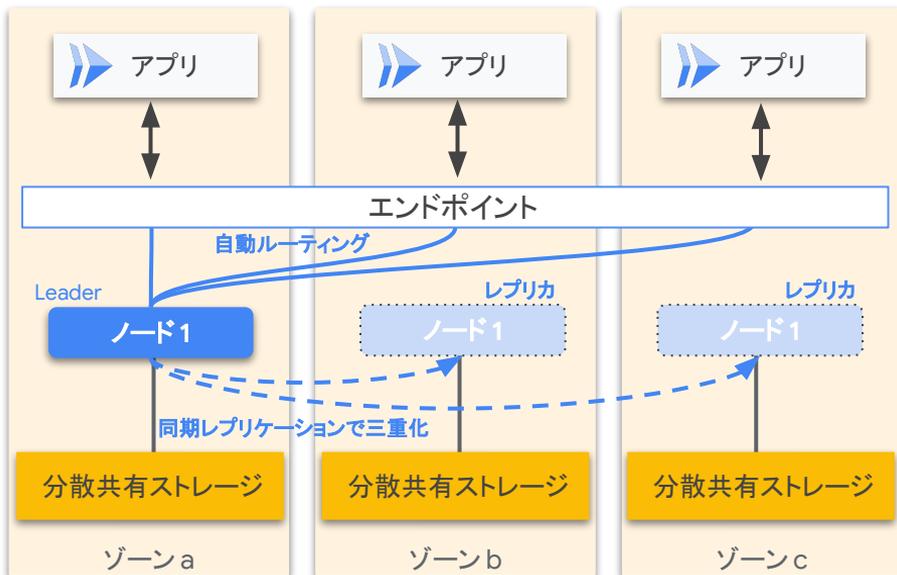


クラウドネイティブ DB の可用性 - Cloud Spanner の例



Cloud Spanner

Cloud Spanner インスタンスは、ゾーンごとに処理用のタスクが起動しており、1ノード構成であっても可用性を保つ構成になっている。1つがリーダー(Leader) となり読み書きを処理し、トランザクションのコミット時には、他のゾーンにレプリケーションを行う。レプリカは読み込み専用としても利用可能。



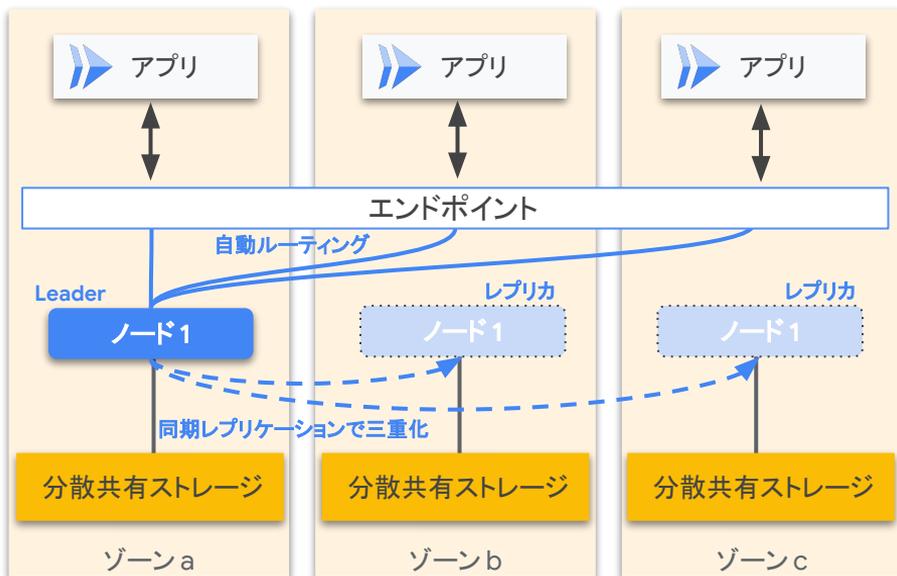
正常時

クラウドネイティブ DB の可用性 - Cloud Spanner の例



Cloud Spanner

ゾーン障害発生時は、停止時間無く(RTO=0)、データ欠損も無く(RPO=0)残りのゾーンで処理を継続可能。
このアーキテクチャの延長でマルチリージョン構成も可能で、リージョン障害発生時であっても同様に、停止時間やデータ欠損無く処理を継続可能。この仕組みを利用し、内部的なメンテナンスも無停止で可能。



正常時



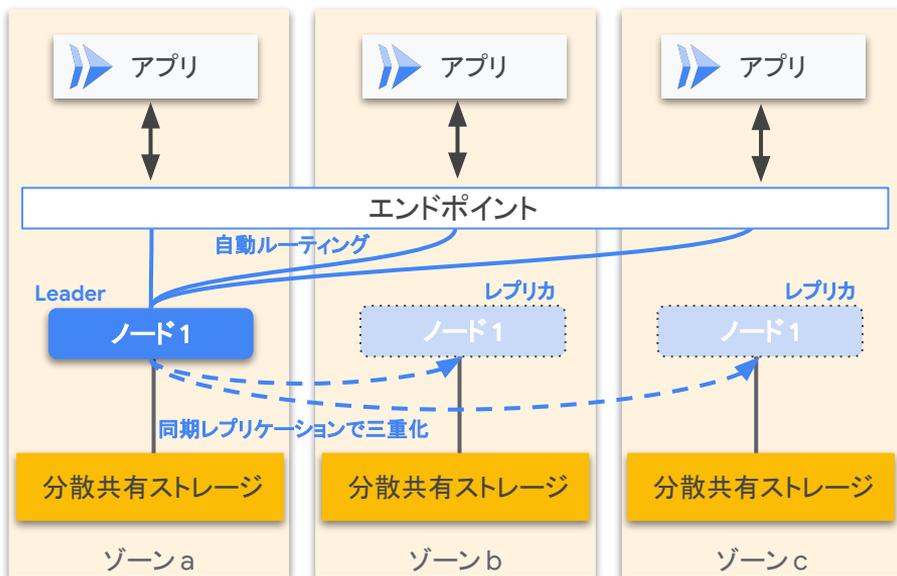
ゾーン障害発生時

クラウドネイティブ DB の拡張性 - Cloud Spanner の例

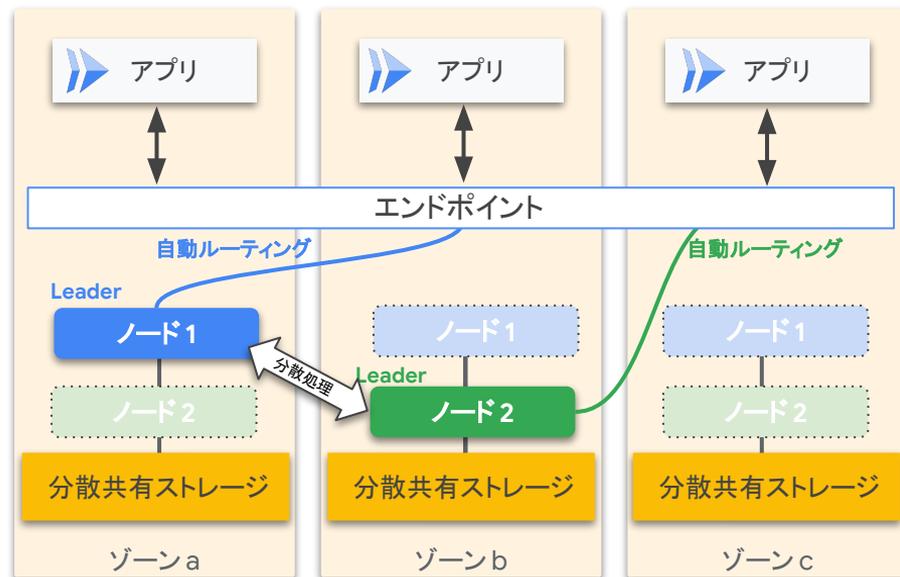


Cloud Spanner

ノード追加時も各ゾーンに追加のプロセスが起動する。データは分散共有ストレージに保存されており、自動的に分割（シャーディング）されたデータを、複数ノードがそれぞれ担当する。すべて自動ルーティングや内部的に分散処理されるため、アプリはシームレスにデータを処理することが可能。



正常時(1ノード構成)



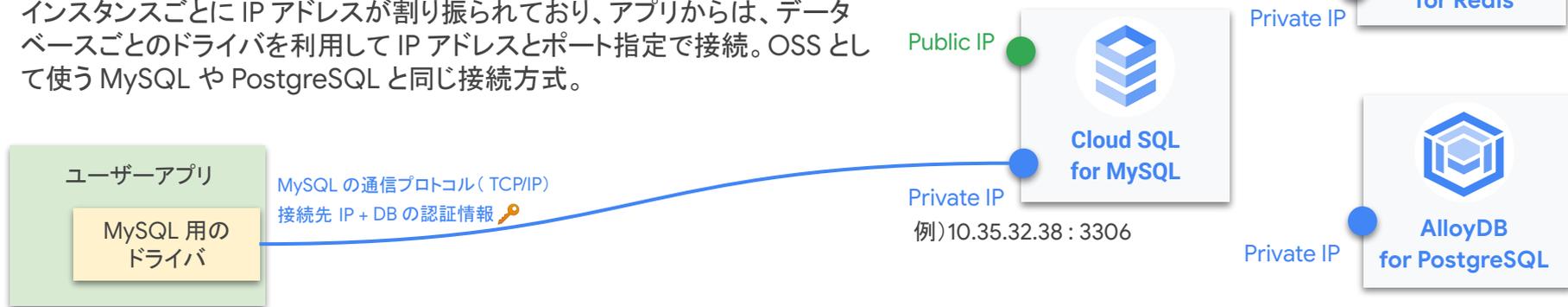
正常時(2ノード構成)

アプリとデータベースの接続方式の違い

OSS / 商用のマネージド DB

Cloud SQL / Memorystore / AlloyDB

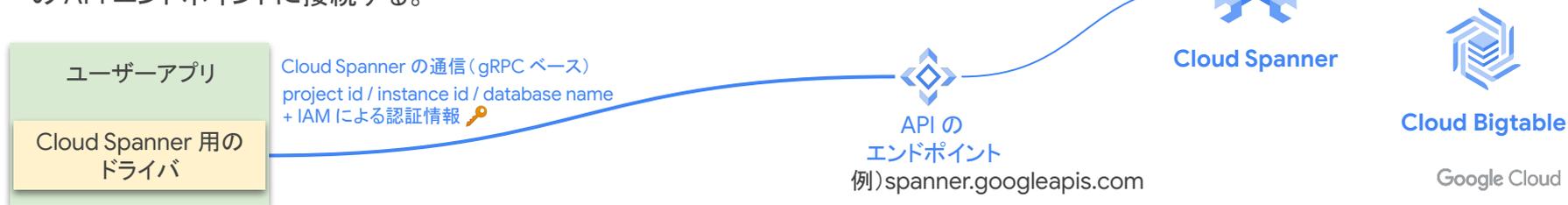
インスタンスごとに IP アドレスが割り振られており、アプリからは、データベースごとのドライバを利用して IP アドレスとポート指定で接続。OSS として使う MySQL や PostgreSQL と同じ接続方式。



クラウド ネイティブ DB

Cloud Spanner / Cloud Bigtable / Firestore

サービスごとに API エンドポイントがあり、アプリからは、Google Cloud が提供している各プロダクト用のドライバを利用して、共通の API エンドポイントに接続する。



Cloud SQL Auth Proxy とは

Auth Proxy とは、**経路の暗号化、IAM による認証を容易に実現するためのプロキシ**。Cloud SQL インスタンスの Public IP 接続時は、承認済みネットワーク(接続元として許可する IP アドレス リスト)の構成も自動的に行われる。クライアントのバイナリを必要に応じてダウンロードして起動させて利用する。Cloud Run などでは標準で組み込まれており、UNIX ソケット経由のローカル通信で、Auth Proxy を利用できる。

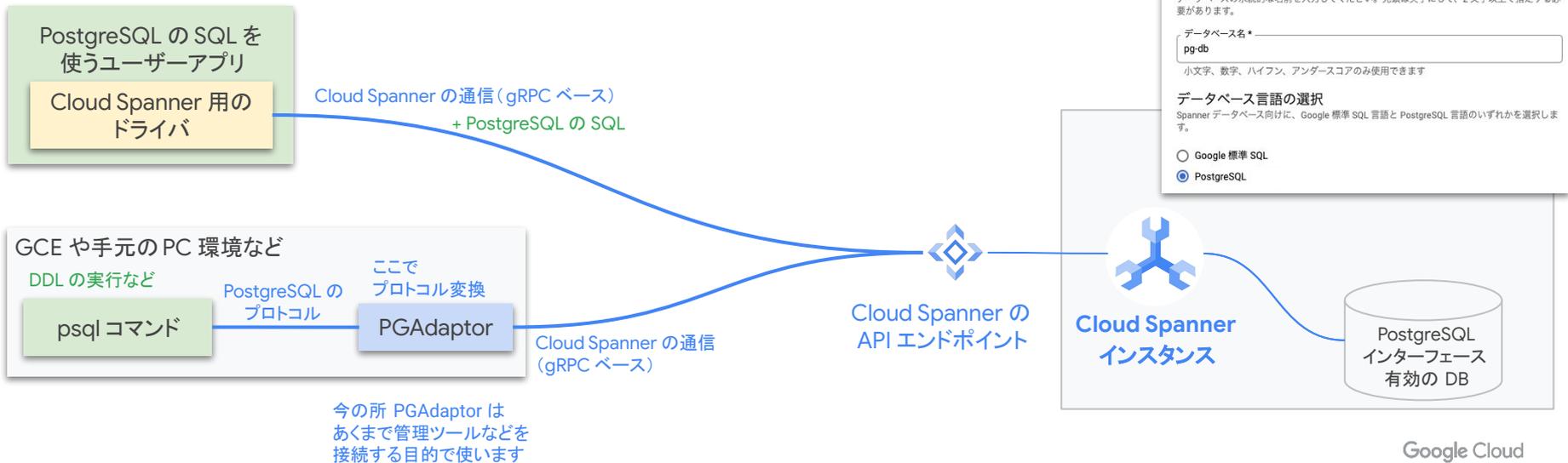
Auth Proxy 利用時の注意点として、Auth Proxy はあくまで認証(Auth)のプロキシであって、Auth Proxy とデータベースは、Public IP または Private IP でネットワーク的に直接接続できる必要がある。



Cloud Spanner の PostgreSQL インターフェース

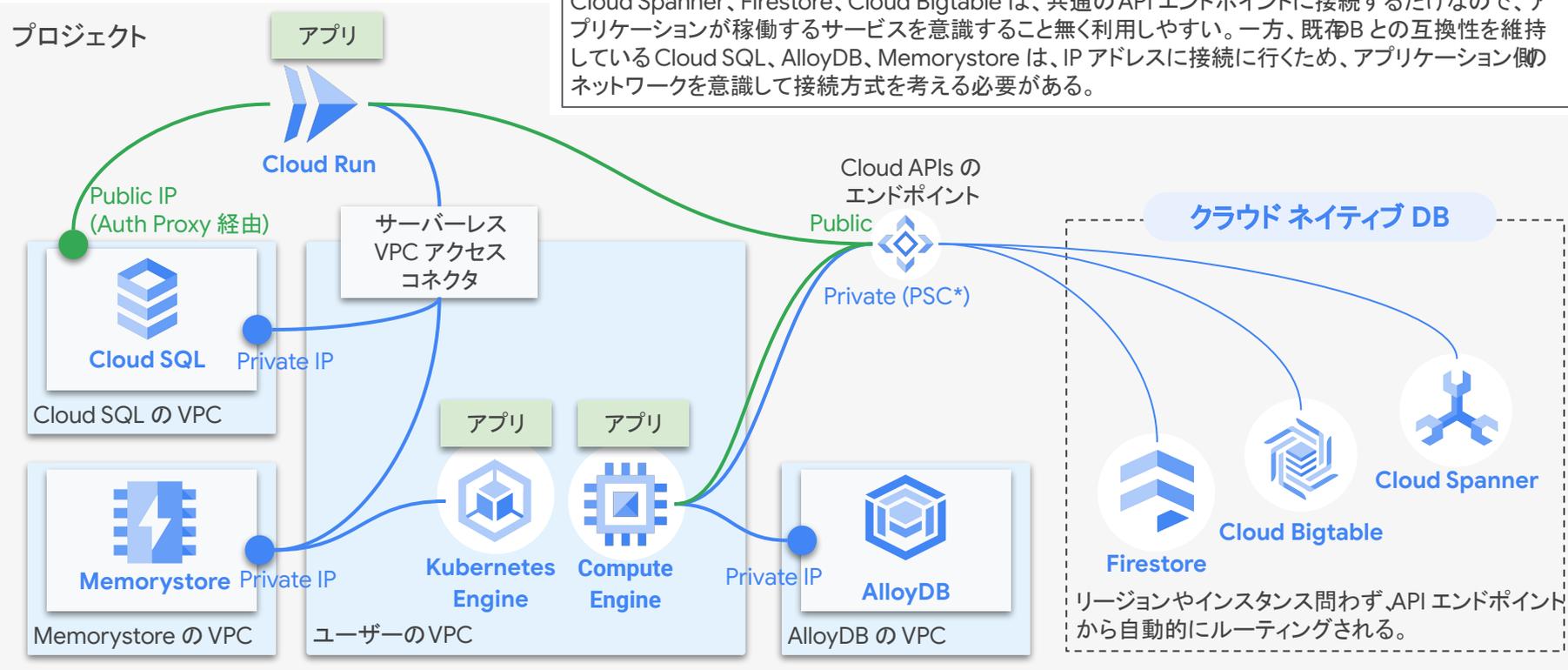
インスタンス内に、PostgreSQL と SQL の互換性をもった Cloud Spanner DB を作ることができる。これまで利用している Cloud Spanner クライアント ライブラリ経由で PostgreSQL 互換の SQL を実行可能になる。DDL を流したいなど、管理目的などで psql コマンドなどから接続したい場合は、別途 PGAdaptor という、プロトコル変換用のアダプタ経由で接続可能。

あくまで中身は Cloud Spanner なので、Cloud Spanner に存在しない機能やデータ型は利用できないので注意。設計のベストプラクティスも、PostgreSQL ではなく Cloud Spanner のものに合わせる。



アプリとデータベースの接続俯瞰図

Cloud Spanner、Firestore、Cloud Bigtable は、共通の API エンドポイントに接続するだけなので、アプリケーションが稼働するサービスを意識すること無く利用しやすい。一方、既存DBとの互換性を維持している Cloud SQL、AlloyDB、Memorystore は、IP アドレスに接続に行くため、アプリケーション側のネットワークを意識して接続方式を考える必要がある。



リージョンやインスタンス問わず API エンドポイントから自動的にルーティングされる。

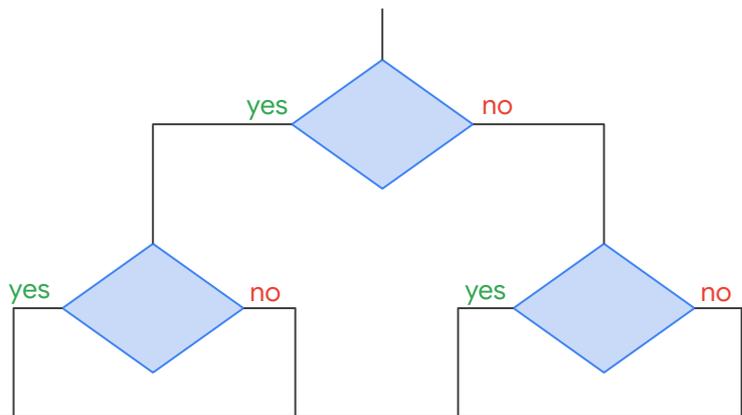
*) PSC = Private Service Connect

03

Google Cloud の マネージド データベース の選び方

内製化向き

データベース選定を Yes / No だけで決めるのは難しい



Yes / No での決定は難しい

DB の選定には様々な視点があるため、Yes / No だけで決定するのは難しい。求められる要件から総合的に考える必要がある。内製化を考えるなら、

内製化に向けたDBをまずは選ぶ

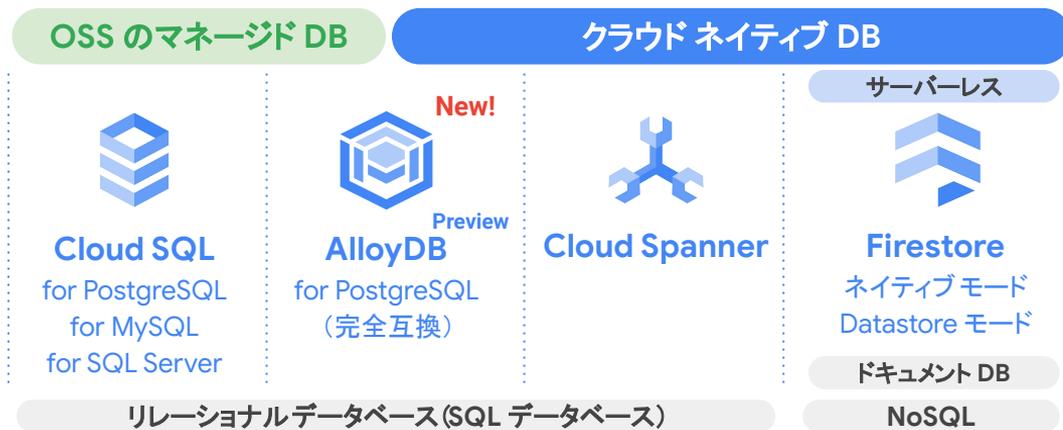
- 使い慣れたDBをマネージドで使える
- DBのことを考えなくても使える

DBの特性を正しく理解して使う

- 各種要件から特定用途にあったものを選ぶ

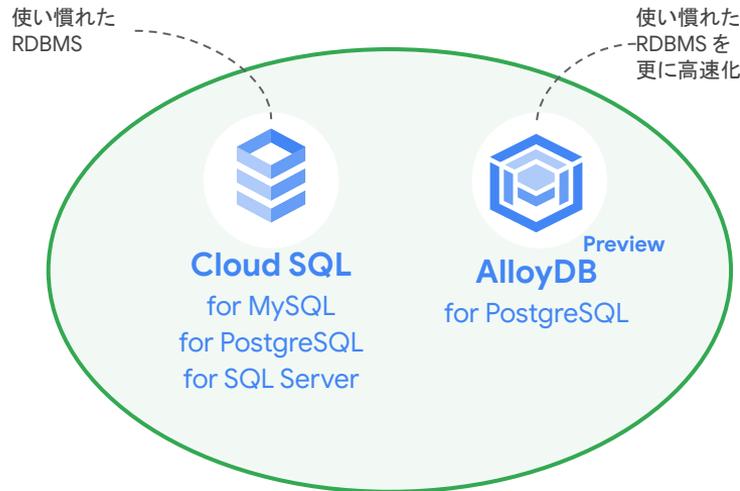


内製化に向けた DB を選ぶなら検討したいこの 4 つ



注意)
AlloyDB は 2022 年 6 月現在はまだプレビュー期間中であり、サポート対象ではないため AlloyDB GA 後の状況で選ぶ選択肢として今回は紹介。

使い慣れたデータベースをマネージド DB で使いたい場合



特に OSS DB として有名な **MySQL**、**PostgreSQL** を使いたい場合は、Cloud SQL、AlloyDB (GA 後) を選択するのが良い。

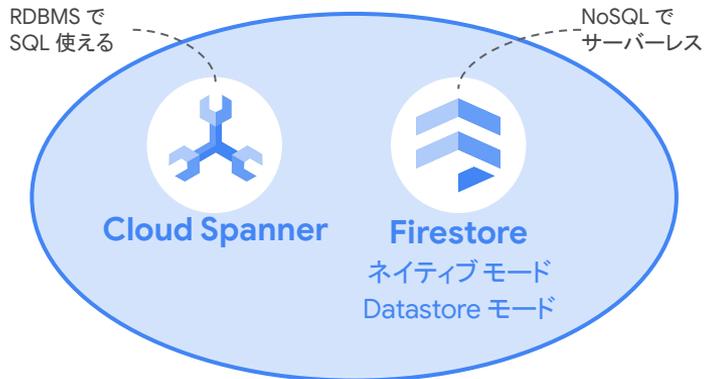
AlloyDB は大変魅力的な選択肢だが、2022 年 6 月現在はまだプレビュー期間中であり、サポート対象ではないため本番環境での利用は非推奨。選択肢となるのは GA 後。

これらを使う場合、読み込み性能はリードレプリカによって向上可能。可用性については HA 構成など適切な選択をすることで、稼働率は Cloud SQL で 99.95%、GA 後の AlloyDB で 99.99% の SLA を提供可能

OSS / 商用 のマネージド DB の利点

- 使い慣れた既存 DB 製品を利用可能
- 既存アプリケーションの移行が容易
- 運用容易なマネージド DB として活用

クラウドに特化した DB で運用をとにかく楽にしたい場合



クラウド ネイティブ DB の利点

- SLA にして99.99%～99.999% の稼働率
- 無停止でのスケールアウト
- メンテナンス ダウンタイム無し
- 設定不要のレプリケーション
- 全自動のシャーディング管理
- 全自動の障害復旧
- 全自動であり運用負担ほぼなし！

クラウド環境に特化した DB を選ぶなら、Cloud Spanner または Firestore が選択肢としてあがる。Cloud Spanner は必要な性能を数値で入力するだけ。Firestore はサーバーレス DB なので必要に応じて性能を自動でスケール可能。稼働率は Cloud Spanner、Firestore とともに、リージョン構成で 99.99%、マルチリージョン構成で 99.999% の SLA を提供可能。必要な運用は共にほぼなく、アプリ開発に集中できる。

構成を選択

ノードとデータの配置を決定します。費用、パフォーマンス、レプリケーションに影響します。マルチリージョン構成では、リーダーレプリカのデフォルトのリーダーリージョンが選択されます。リーダーリージョンは、DDL ステートメントによっていつでも変更できます。詳細

COMPARE REGION CONFIGURATIONS

- リージョン
 マルチリージョン

asia-northeast1 (東京)

コンピューティング容量の割り当て

コンピューティング容量によって、インスタンスのデータスループット、秒間クエリ数 (QPS)、ストレージの制限が決まります。1 ノードは、1,000 処理ユニットに相当します。この選択によって費用は変動します。

単位*
ノード

数量*
3

1 以上の整数を入力してください

Cloud Spanner を使うときは
インスタンスを立てるリージョンと、ノード数
(欲しい性能値)を決めるだけ

モバイル及びウェブアプリを高速に開発する為の DB なら

Firestore (ネイティブ モード)

NoSQL の一種であるドキュメント ストアを API コール課金のサーバーレス DB として提供。Firebase との親和性が高く、モバイルアプリから直接接続が可能であり、DB への意識はおろかサーバーサイド要らずで開発を高速化。



Realtime Database for Firebase



Firestore 由来の **Realtime Database**
Google Cloud の **Cloud Datastore**
(App Engine の)

この両者をベースに、Google Cloud の新しいインフラ上に構築された、新しいサーバーレス データベース。ネイティブ モードと互換性維持の Datastore モードがある。

サーバーレスなサーバーサイドと相性のよい DB なら



Firestore (Datastore モード)

主に App Engine とセット利用されてきた Cloud Datastore の後継であり、API の互換性を維持したまま、Firestore の可用性や強い整合性といったメリットを得られる。

大規模な共有バックエンドに支えられた、オートスケール可能なサーバーレス環境であり、App Engine や Cloud Run などと組み合わせることで、運用いらずのシステムを構築できる。なおこのモードでは、Firebase SDK は使えない。



⚠ このプロジェクトに対し、ここで選択したモードは変更できません

	ネイティブ モード	Datastore モード
	オフライン サポートとリアルタイム同期を使用して、Cloud Firestore のすべての機能を有効にします。 ネイティブ モードを選択	Cloud Firestore の強力なストレージレイヤ上で、Cloud Datastore のシステム動作を利用します。 DATASTORE モードを選択
API	Firestore	Datastore
スケーラビリティ	数百万のクライアント同時実行まで自動的にスケールします	毎秒数百万回の書き込みまで自動的にスケールします
App Engine のサポート	App Engine の標準 Python 2.7 および PHP 5.5 ランタイムではサポートされない	すべてのランタイム
1 秒あたりの最大書き込み数	10,000	上限なし
リアルタイム アップデート	✓	×
オフライン データの永続性をサポートするモバイルおよびウェブクライアントライブラリ	✓	×
クエリの一貫性	強力	強力
データモデル	ドキュメント / コレクション	エンティティ / 種類
ウェブ コンソール	Google Cloud Platform と Firebase の [Firestore] ページ	Google Cloud Platform の [データストア] ページ

データ活用するための分析 DB を選ぶなら



各種エコシステム

BigQuery

フルマネージドなサーバーレス DWH。無制限のストレージと Exa バイト級のクエリが可能なエンジンを提供。インデックス チューニングなども不要で、アジャイルなデータ活用が可能。

単なる DWH ではなくデータ活用基盤であり、BigQuery を中心としたエコシステムを構成。機械学習や GIS など高度なユースケースもデータ移動や追加環境不要で実現可能。



大量データを低レイテンシに処理するための DB なら

Cloud Bigtable

NoSQL の一種である**ワイド カラムストア**を提供。

シンプルに言えば、低レイテンシな読み書きを、高スループットで提供可能な、スケールするキーバリューストアのようなもの。データ分析系の製品との連携性も高い。Apache HBase との互換性もあり。

—列の値が入っていたり入っていなかったりする、スカスカのテーブルのこと

スパースなテーブルを効率よく扱うことができるため、例えば数千や数万列からなるテーブルにデータを格納して処理することが可能。

テーブルに大量の列を保持する事になりがちな、IoT センサー値の処理基盤、機械学習のデータ基板といったシステム用のデータベースに適している。

また上記に加えレイテンシ要件が非常に厳しいシステムで選択肢になりうる。

Row Key								
1234500	bob	1	100	4	1	100		
1234515	alice	2	0	124	2	0	124	
1234529	carol	1	50	33	1			
1234543	steve	2	10000	2		10000		
1234551	oscar	2	1000					
1234578	dave	3	700		3		124	3



04

まとめ

まず 4 種の DB から検討し、必要に応じて特化型の DB を利用

OSS 及び商用のマネージド DB

クラウド ネイティブ DB (可用性 99.99% 以上を実現可)



Memorystore
for Redis
for Memcached

キャッシュ



Cloud SQL
for PostgreSQL
for MySQL
for SQL Server



AlloyDB
for PostgreSQL
(GA 後に選択肢に)



Cloud Spanner
PostgreSQL
SQL 方言のみ互換

リレーショナルデータベース (SQL データベース)

内製化向き

まずはこれらの DB で要件を満たせるかを検討する。

可用性

99.95%
別途メンテナンスあり

99.99%
メンテナンス含む SLA

99.99 ~ 99.9999%
メンテナンスによる
停止時間などは一切なし

性能と拡張性

書き込み含めた自在なスケーラビリティが必要ななら Cloud Spanner。
提供されているインスタンス サイズで足りる性能でよければ、Cloud SQL または AlloyDB (GA 後) を選択するのがよい。



Cloud Bigtable

KVS / ワイドカラム

NoSQL データベース

特定用途に特化

高スループットと低レイテンシが、同時に必要な場合の選択肢。

大量の IoT センサーデータの処理や、機械学習でのレコメンデーションに利用する大量データなどを、リアルタイムに読み書きするといったユースケースなどに適している。

サーバーレス



Firestore
ネイティブモード
Datastore モード

ドキュメント DB

NoSQL データベース

内製化向き

Cloud Run + Firestore (Datastore モード) は、サーバーレスの恩恵を最大限活用できる、運用いらずの組み合わせ。

特定用途に特化

Firebase 連携するなら、Firestore ネイティブモード



BigQuery

DWH

特定用途に特化

分析 DB、分析プラットフォームとして使うなら、間違いなくこれ。

機械学習モデルの学習基盤などにも最適。

フロントにキャッシュが必要なときには検討