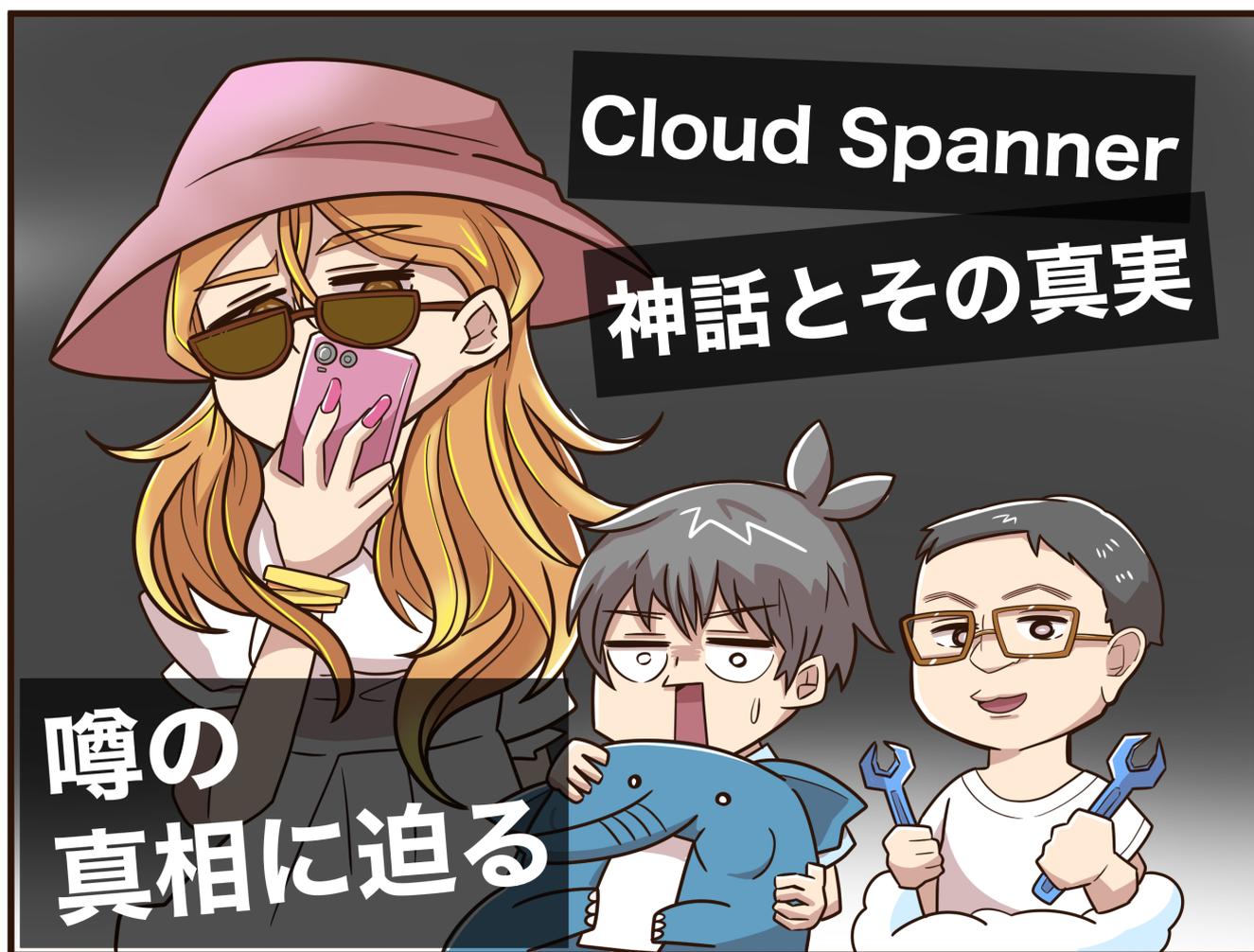


# マンガでわかる Google Cloud

Cloud Spanner 編







### 7つの神話

- ① [Cloud Spanner は大規模用途向け?](#)
- ② [Cloud Spanner は費用が高い?](#)
- ③ [Cloud Spanner はスケール、整合性、レイテンシの間でトレードオフが必要?](#)
- ④ [Cloud Spanner は独自のインターフェースしか存在しない?](#)
- ⑤ [Cloud Spanner は特定のコンソールからしか監視できない?](#)
- ⑥ [Cloud Spanner は複数リージョンを使うグローバル用途にのみ向いている?](#)
- ⑦ [Cloud Spanner でスキーマ変更すると大規模なロックがかかる?](#)

## 概要

Google Cloud にはワークロードや用途に応じたいろいろな種類のデータベースが用意されています。Cloud Spanner はクラウド向けに設計されたスケーラビリティと可用性を兼ね備えたリレーショナル データベースのサービスです。今回は、Cloud Spanner でよく質問をいただくポイントについて、7つの神話とその真相という形で、最新情報を交えながら紐といていきます。

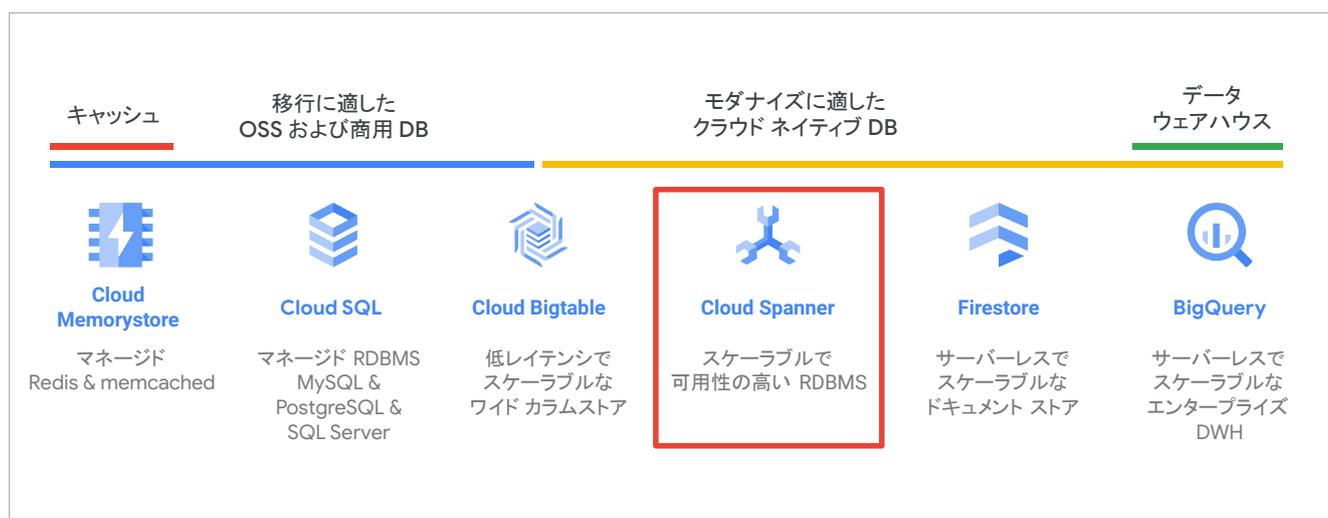
## 解説者



Google Cloud  
ソリューション & テクノロジーグループ  
データ・プラットフォーム事業開発部長 大久保 順

金融系 SI 企業でシステム エンジニアとしてキャリアをスタート。その後データベース製品のプロダクト マネージャー、クラウド プラットフォームの事業開発などを経て、2021 年より現職。Google Cloud のデータ マネジメント ソリューションを日本のお客様に広く知っていただき、ビジネスに活用いただくための活動に従事。

# Google Cloud の マネージド データベース サービス ラインナップ



本題に入る前に、Google Cloud のマネージド データベース サービスのラインナップと Cloud Spanner の立ち位置についてお話ししますね。

そうそう。Google Cloud の中でもデータベースのラインナップが多くてどれが良いかわからなかったんです。



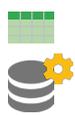
Cloud Spanner はクラウド向けに設計された、スケーラビリティと可用性を兼ね備えたリレーショナル データベースのサービスです。

Cloud SQL と BigQuery はなんとなく知ってたけど、他にもいろいろあるんですね。



Cloud Spanner の 3 つの特徴をご紹介しますね。

## クラウド ネイティブなデータベース Cloud Spanner の特徴



### 特徴 1 - 運用いらずのフルマネージドRDBMS

フルマネージド データベースで、メンテナンス含めあらゆる運用は全自動  
テーブルに対して SQL でのクエリや、ACID トランザクションをサポート



### 特徴 2 - リージョン障害にも耐えられる高可用性

ゾーン障害はもちろんリージョン障害にも耐え、最大 99.999% の可用性を提供  
メンテナンスやノード数変更時のダウンタイムなども無し



### 特徴 3 - 必要な性能を容易に得られるスケーラビリティ

必要な性能に合わせて、無停止でスケールアウトやスケールインが可能  
テーブル内のデータは自動シャーディングされるため、一切の運用負担はなし

神話 1

## Cloud Spanner は大規模用途向け？



Spanner は YouTube 、 Google ドライブ、 Gmail などに利用されています。ピーク時には 1 秒間に 20 億回以上のリクエストを処理しているんですよ。

名だたるサービスに利用されているんですね！うーん、やっぱり、うちのサービスには高性能すぎるような…。やっぱり大規模向けなんですね。



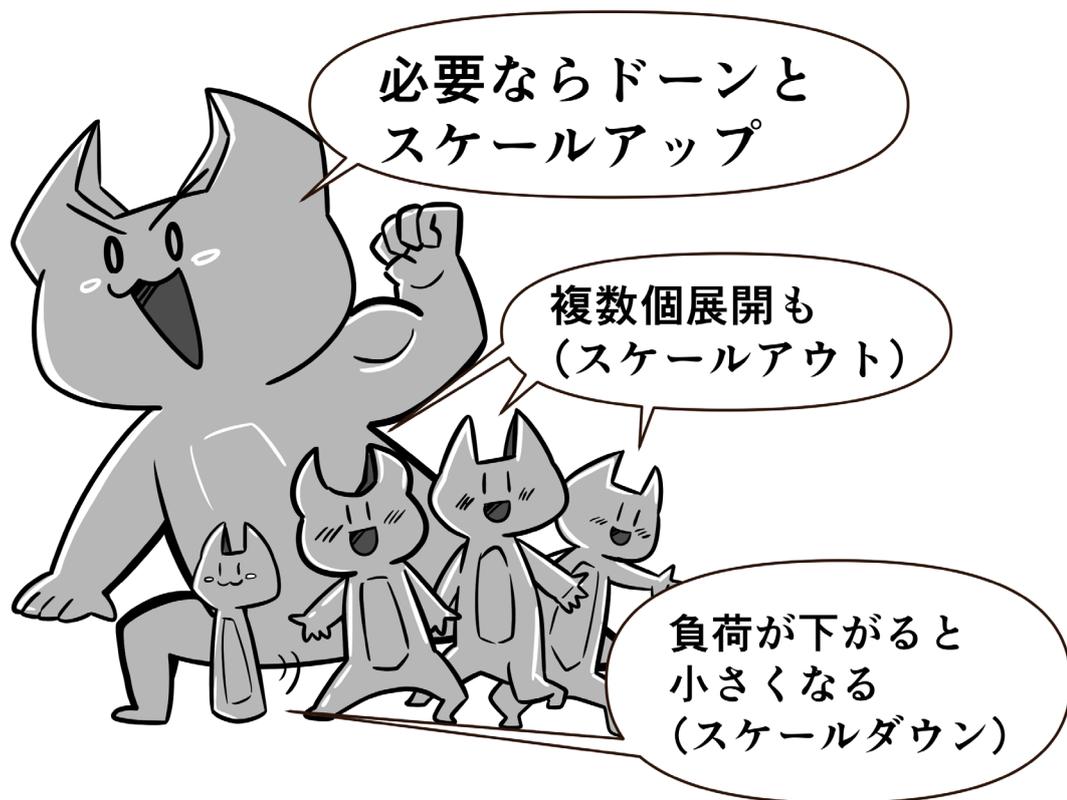
いえいえ、小規模なデータベースでも、パッチ適用やアップグレードなど運用負荷とコストの削減効果は高いですよ。多くのお客様が可用性とスケーラビリティの観点から、小規模なワークロードにも Spanner を活用されています。Google 内でも、Google Password Manager というサービスで、小規模なワークロードで使っていたりしますよ。

このようなユースケースは、ダウンタイムがユーザに与える影響が大きいため高可用性が必要で、また将来の成長に備えてスケーラビリティを必要としているためです。

クラウド化を進める多くのお客様は、ワークロードの規模に関わらず、拡張性と可用性のメリットを求めると同時に、パッチ適用やアップグレード等のメンテナンスの運用負荷とコストを軽減したいと考えていらっしゃいますね。

なるほど。拡張性と可用性のメリットがあるのはクラウド サービスのメリットですね！



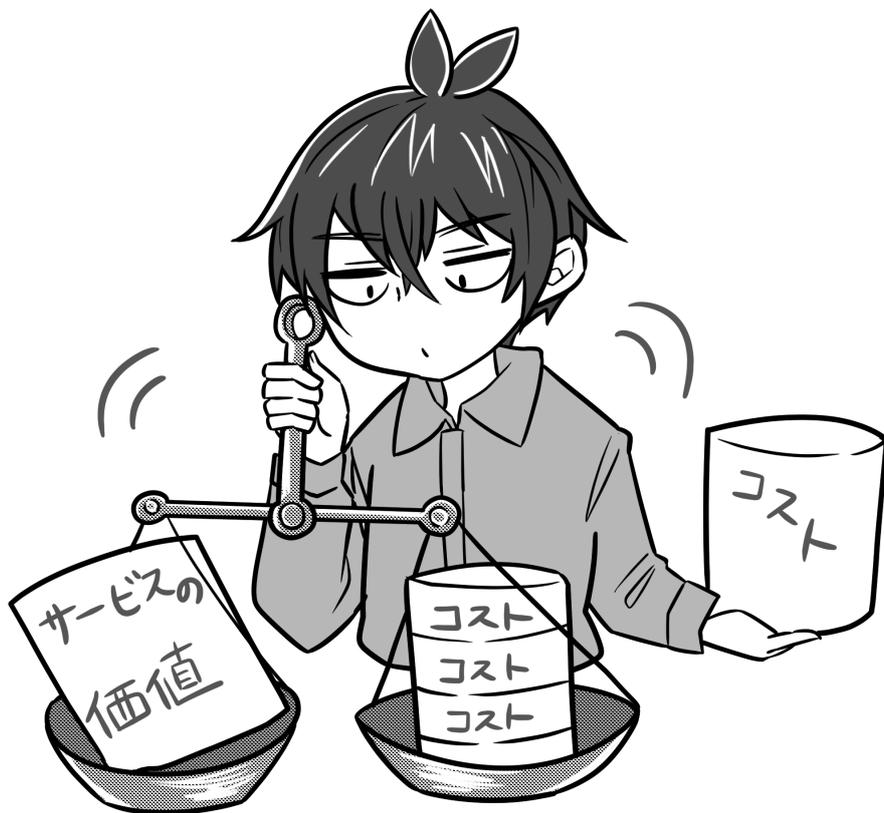


## 神話 2 Cloud Spanner は費用が高い？

小規模なワークロードでも十分メリットがあるのはわかったんですが、やっぱりこう、高い気がして…。



データベースのコストを見る場合、価格そのものよりも TCO (Total Cost of Ownership) とそのサービスが提供する価値を考えることがポイントです。従来のデータベース管理では、実際にデータセンターの準備から、ハードウェアの調達、OS や DB のインストールやパッチ適用、構成バックアップや監視、スケーラビリティの準備を自前で行う必要がありましたが、Cloud Spanner ではその全てが自動化されています。ユーザーの皆さんは、こうした時間を生産性の高い作業に振り分けることができますよ。



データベースのメンテナンスに時間をとられてコードを書く時間が足りないことが悩みだったので、コードを書くことに集中できるのは有難いです！実際に使ったらどれぐらいかかるんだろう…？（試算サイトでポチポチ）最低単位の1ノードを東京リージョンで1年動かすと…150万円かあ…。ここからさらにストレージ料金もかかるし、う、うーん、やっぱりちょっと高いです。大久保さん、もうちょっとなんとかありませんか？



Granular Instance Sizing という機能を提供しています。これで1ノードの10分の1の単位、100 PU (Processing Unit) から使えるようになりました。

この機能の導入によって、東京リージョンでは最小構成の場合、1ヶ月約1万円から使い始めることが可能となっています。さらに確定利用割引というものもあって、1年以上の継続利用を約束いただくと大幅な割引価格で利用できますよ。

おお！それなら経営層に提案しやすい気がします！



神話 3

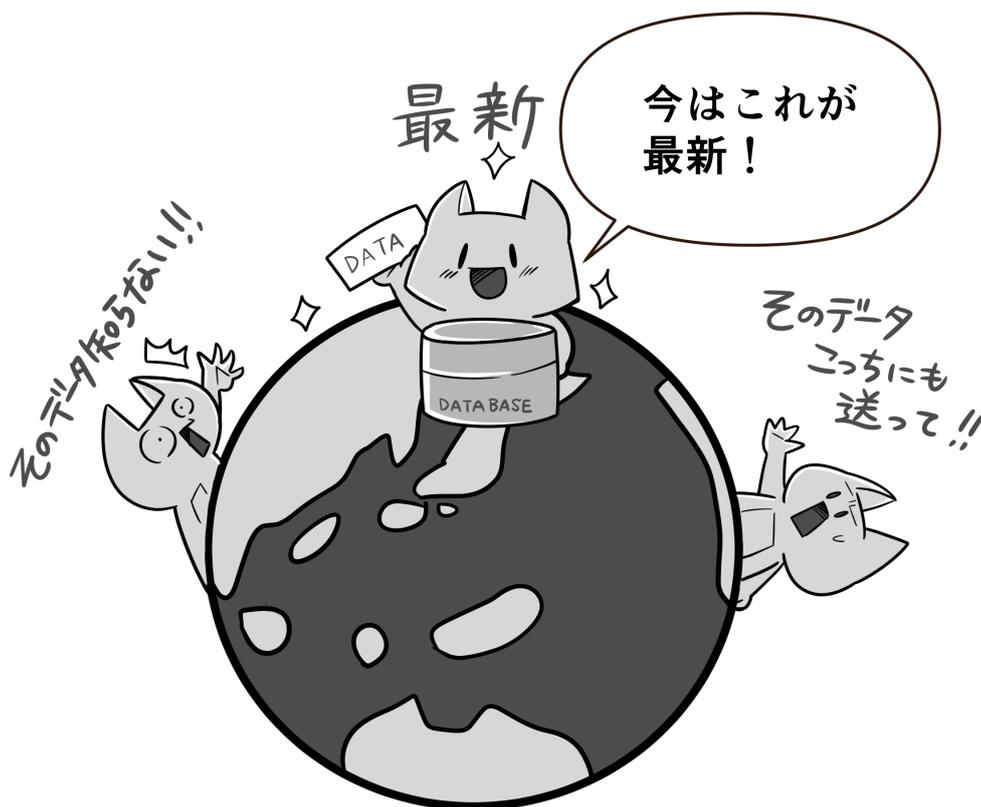
## Cloud Spanner はスケール、整合性、レイテンシの間でトレードオフが必要？



スケール、整合性、レイテンシ（※）の間でトレードオフが必要ですか？  
という質問もよくいただきますね。

※データを要求してから実際に返ってくるまでの時間のこと

それは思いました！スケールが大きくなるほどデータの整合性を保つ  
のが大変そうだし、レイテンシも落ちそうだなと。

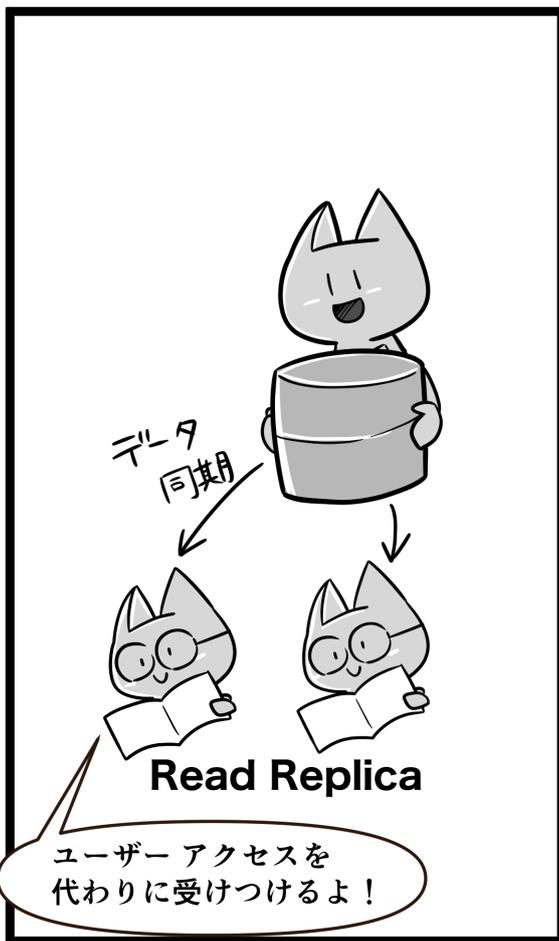


スケール、整合性、レイテンシの間でトレードオフが必要ですか？  
という質問もよくいただきますね。



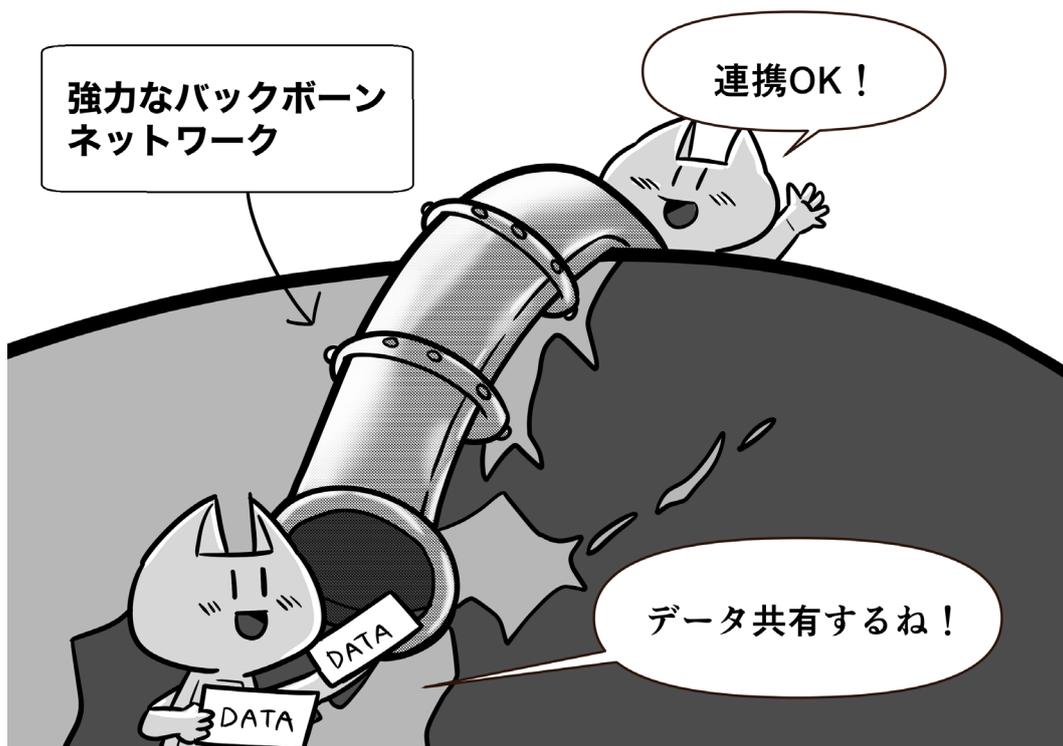
ユースケースや構成に応じて、一貫性、レイテンシ スケールのどれかを選ぶ必要がないように、Spanner を使用することができるようになっています。

強力なデータ一貫性を提供するために、Paxos ベースの同期レプリケーションの仕組みを使用しています。(データの) 書き込みは、クォーラムと呼ばれるレプリカの過半数が書き込みをコミットすることに同意したとき確定されます。





シングル リージョン構成の場合、レプリカはリージョン内に存在するため、マルチリージョン構成より書き込みが早くなります。マルチリージョン構成で書き込みが遅いのかということはありません。書き込みレイテンシが許容範囲内に収まるよう、Google のバックボーンネットワークを利用し、地理的な構成に十分注意した設計になっているので、安心してくださいね。



神話 4

## Cloud Spanner は独自のインターフェースしか存在しない？

ところで Cloud Spanner のインターフェースってどうなのでしょう？ Spanner のメリットはわかったんですけど、現行のシステムで PostgreSQL を使っていて、そこから使い勝手が大きく変わるとなると、やはり移行の荷が重いです。



ANSI 2011 準拠の Google 標準 SQL や REST、gRPC API インターフェースに加えて PostgreSQL と互換性を持つインターフェースを提供しています。

### Cloud Spanner PostgreSQL Interface Spanner PostgreSQL のインターフェース

#### 使い慣れたスキル及びツールを利用可能

これまで培ったスキルやツールを利用しながら、Spanner が持つスケーラビリティ、99.999% の可用性を享受可能

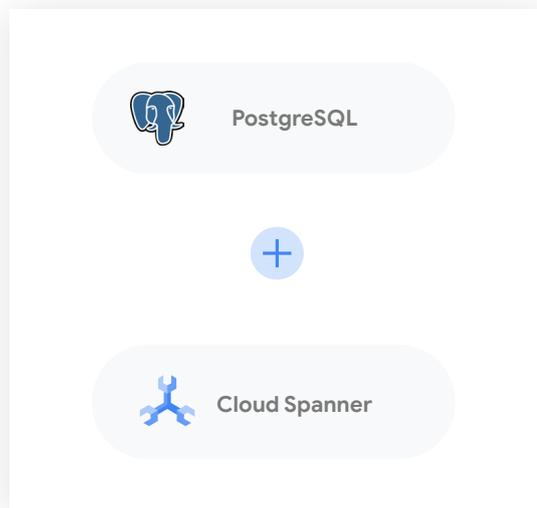
#### アプリケーションのポータビリティ向上

他の PostgreSQL でも実行可能な SQL なので、別の PostgreSQL 環境へも移植可能

#### 開発効率の向上

既存の PostgreSQL リソース及びナレッジを活用することで、学習コストを削減

詳細: <https://cloud.google.com/spanner/docs/postgresql-interface>



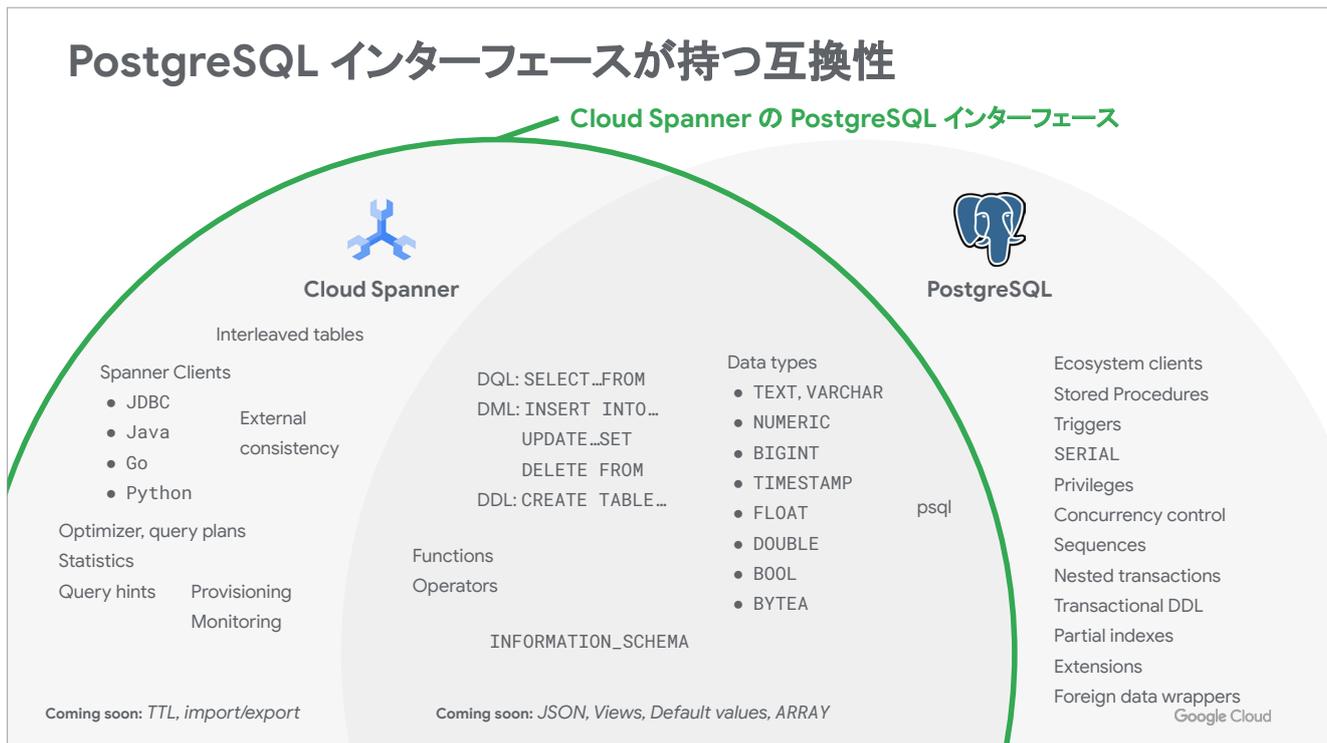
Google Cloud

PostgreSQL のインターフェースがそのまま使えるんですか？





PostgreSQL を使用されているなら、Spanner との互換性は以下の通りです。丸が重なっている部分は利用することができますよ。



PostgreSQL の使用感を保ったまま、Cloud Spanner のオートスケールやフルマネージドといった機能を享受することができるわけですか。いいなあ！



ユーザーエクスペリエンスとしては、データベースを作るときに Cloud Spanner の標準 SQL を使うか、PostgreSQL 互換の SQL を使うかを選ぶだけです。非常にシンプルで簡単に使い始めることができますよ。

## ユーザー エクスペリエンス

PostgreSQL クエリを Cloud Spanner インターフェースへ実行 + PostgreSQL プロトコル

- Cloud Spanner のデータベース作成時に SQL 方言を指定
- PostgreSQL の SQL を PostgreSQL プロトコルで実行可能
  - PostgreSQL コミュニティのツールを利用  
(現時点では **psql** をサポート)
- Cloud Spanner のエンドポイントへ直接 PostgreSQL の SQL を実行
  - オープンソースのドライバを利用  
(現時点では **JDBC, Go, Python** をサポート)
  - gcloud CLI
  - Cloud Console UI
- リソースの作成、モニタリングは既存の Cloud Spanner と同様

Name your database  
Enter a permanent name for your database of at least two characters, starting with a letter

Database name \*

Traditional\_SQL, PROD

Supported values: traditional\_sql, traditional\_sql\_uppercase, traditional\_sql\_lowercase

- Google Standard SQL
- PostgreSQL

Google Cloud



また、各種 ORM (Object Relational Mapping) マッパーへの対応が進み、アプリケーション開発者は Cloud Spanner 独自のインターフェースを意識せずにコードを書くこともできるんですよ。

Cloud Spanner ではこちらのスライドに示した各種 ORM に対応しています。データベース エンジンごとの差異を吸収し、アプリケーション開発者がバックエンドのデータベースの種類を強く意識しなくても、アプリケーション コードを記述することができます。

## Cloud Spanner 用の ORM 一覧



### Google Cloud によって OSS として提供

- Java - Hibernate
  - <https://github.com/GoogleCloudPlatform/google-cloud-spanner-hibernate>
- Python - Django ORM
  - <https://github.com/googleapis/python-spanner-django>
- Python - SQLAlchemy
  - <https://github.com/googleapis/python-spanner-sqlalchemy>
- Ruby - Active Record
  - <https://github.com/googleapis/ruby-spanner-activerecord>
- C# Entity Framework
  - <https://github.com/googleapis/dotnet-spanner-entity-framework>

### コミュニティによる提供

- PHP - Laravel
  - <https://github.com/colopl/laravel-spanner>

Google Cloud

コードの修正も思ったより少なそうで助かりますー。



### 神話 5

## Cloud Spanner は特定のコンソールからしか監視できない？

うちの会社の運用チームが監視システムとコンソールを Prometheus と Grafana で作り込んでいるので、そこに Cloud Spanner の監視と運用を追加するとなると、学習コストがかかりそうです。そうになると導入に待ったがかかるとかも…。

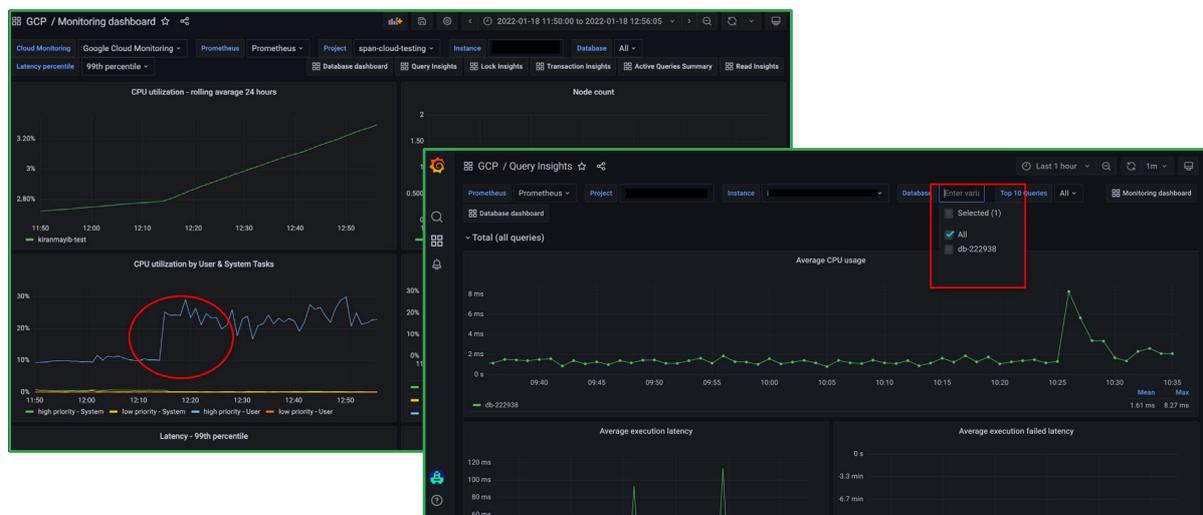




標準のコンソール以外にも、様々な監視方法に対応していますよ。Cloud Spanner のクライアント ライブラリは OpenCensus によるメトリクス収集とトレーシングをサポートしています。クライアント サイドのトレースとメトリクスにはセッションとトランザクションに関する情報が含まれているので、実際に本番環境で起きた問題をデバッグする際に役立ちます。OpenTelemetry レシーバーにも対応しているので、好みのアプリケーション監視ツールにエクスポートする方法も提供しています。例えば、Cloud Spanner のシステム テーブルからのメトリクスを Prometheus + Grafana で可視化することもできるんですよ。Grafana 向けのサンプルダッシュボードも公開されています。(※)

※サンプルダッシュボードへのリンク <https://grafana.com/orgs/architjugran>

## Prometheus + Grafana による可視化イメージ



Google Cloud



Splunk、Datadog、Dynatrace、New Relic、AppDynamics などの商用製品もソリューションとして用意しています。Google Cloud のコンソールも、提供する観測可能なエクスペリエンスを進化させ、お客様がどこにいても最高の体験を得られるように製品の改良を進めています。

それなら、うちの監視システムにも組み込めそうです！



神話 6

## Cloud Spanner は複数リージョンを使うグローバル用途にのみ向いている？

ここまで話を聞いていて、スケールが違うというか、うちのシステムには大きすぎるような気がするんですよね。サービスの特性上、グローバル展開なんかはそうそうないというか、今の段階ではありえないと思っていますし。



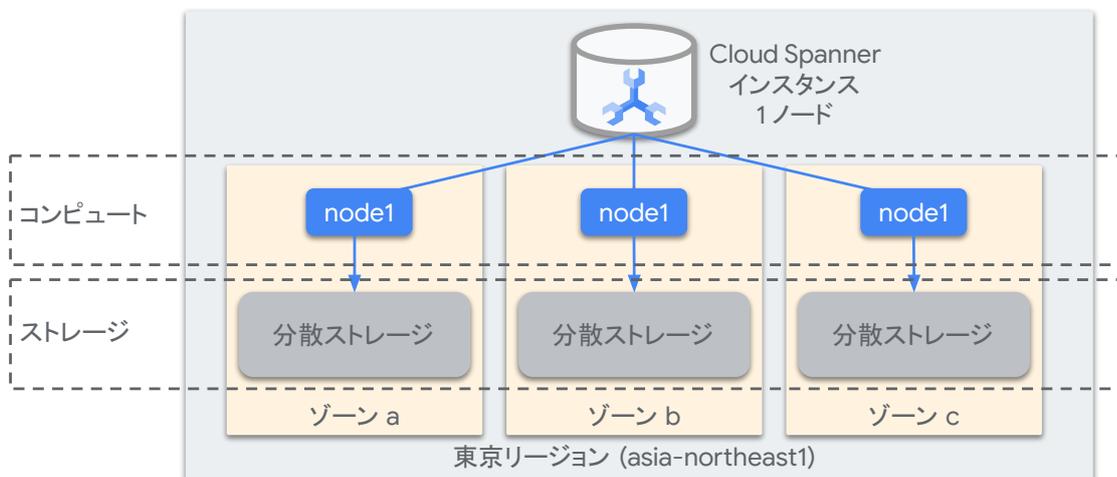
Cloud Spanner は様々なマルチリージョン構成を取ることが可能ですが、各 Google Cloud リージョン内でのシングル リージョン構成も提供していますよ。

シングル リージョン構成でも、Cloud Spanner が持つ高い可用性やスケラビリティという特徴は変わりません。マルチリージョン構成は、アプリケーションが複数の地域でワークロードを実行する場合や、ビジネスで 99.999%の可用性 SLA (Service Level Agreement、サービス提供者が顧客に対してどの程度までの品質を保証するのかを示す水準) が求められる場合、リージョン障害に備える必要がある場合のユースケースに向いています。

シングル リージョン構成時の Cloud Spanner のアーキテクチャを見ましょう。

### Cloud Spanner アーキテクチャ (シングル リージョン構成)

- 標準で冗長化されている (各ゾーンにサーバタスク、ストレージが割り当てられ三重化)
- コンピュートとストレージの分離により、障害影響の分離や効率の良いスケールアップが可能
- データは同期レプリケーションにより、強整合性を持った複製を保持





コンピュータ ノードと分散ストレージ上のデータはリージョン内の 3 つのゾーンに複製されています。データは神話 3 でも解説した同期レプリケーションにより強い整合性を持った複製を保持しています。シングルリージョン構成では 99.99%の可用性 SLA とゾーンの障害に対する保護が提供されています。

うちの場合はシングルリージョン構成でも十分そうです。シングルと聞くと不安な感じがしますが、内部でしっかり冗長化されているんですね！



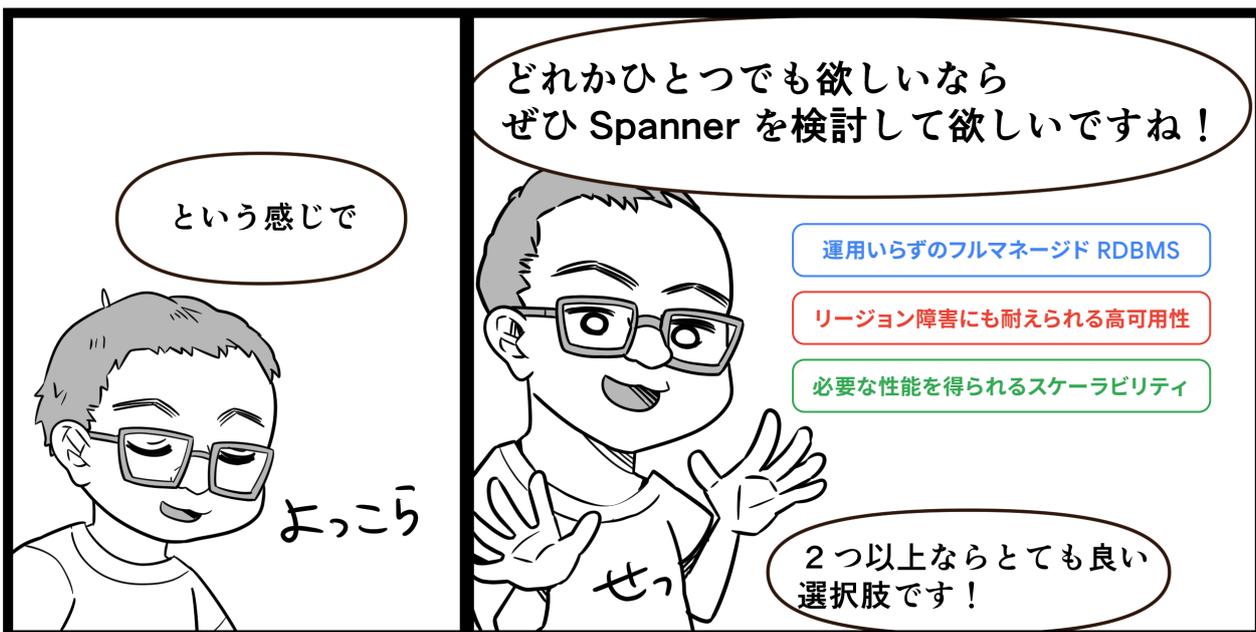
## 神話 7

### Cloud Spanner でスキーマ変更すると大規模なロックがかかる？

Cloud Spanner が強力なことは十分わかったんですが、強力であるが故にいろいろ触るのが怖いな—とも思うんですよ…。今弊社はサービスを拡充していこう！という段階でして、例えばスキーマ変更したときにロックがかかりますよね？ サービス停止につながったりしないかちょっと怖いっす。



Cloud Spanner ではテーブルレベルでロックがかかるということはないんですよ。マルチ バージョン同時実行制御アーキテクチャにより、スキーマとデータの同時実行バージョンを管理しています。スキーマ更新を行う際にテーブルの行数が何億行と入っていても、データベースへの書き込みと読み込みを中断することなく継続することができます。とはいえ、スキーマ変更は全くオンライン処理に影響がないかというそんなことはありません。例えばスキーマの変更を検知すると、スパンのクライアント ライブラリが自動的にトランザクションの再実行を行います。このようにアプリケーション開発者側で認識しておいた方が良いポイントもありますので、Cloud Spanner のドキュメントもあわせて参照いただければと思います。



## 参照リンク

1. [Cloud Spanner 製品紹介ページ](#)
2. [Cloud SQL for PostgreSQL のドキュメント](#)
3. [Cloud Spanner 神話とその真実 ~ 噂の真相にせまる ~ / 大久保 順](#)

## 漫画制作



### 夏目葉太（なつよ）

富山在住のインフラエンジニア。『インフラ女子の日常』著者。小さい頃から絵を描いたり、何かを組み立てたりすることが好き。会社勤めで IT エンジニアをしつつ、個人事業主としてイラスト・漫画作成や Live2D によるアバター作成を行う。最近では Google Cloud と Kubernetes と MongoDB を勉強中。

- [amazon 著者ページ](#)
- [Twitter](#)

## 製品、サービスに関するお問い合わせ



[goo.gl/CCZL78](https://goo.gl/CCZL78)

Google Cloud の詳細については、上記 URL もしくは QR コードからアクセスしていただくか、同ページ「お問い合わせ」よりお問い合わせください。

© Copyright 2022 Google

Google は、Google LLC の商標です。その他すべての社名および製品名は、それぞれ該当する企業の商標である可能性があります。