

PlanetScaleの無料プランがなくなるので、NeonとTiDBを試してみた



Press Space for next page →



自己紹介

- 📄 飯野陽平 ([wheatandcat](#))
- 🏢 会社: [合同会社UNICORN](#) 代表社員
- 📖 Blog: <https://www.wheatandcat.me/>
- 🛠️ 今までに作ったもの
 - [memoir](#)
 - [OOMAKA](#)
 - [MarkyLinky](#)

PlanetScaleとは

- クラウドベースのサーバーレスDB
- MySQLと互換性のあるインターフェースを持つ
- オートスケールやノンブロッキングスキーマの変更をサポート
- YoutubeやSlackなどの大量のデータを扱う企業が利用されている
- 詳細
 - [PlanetScale](#)
- 軽くコンソールを見せてDEMO
 - <https://app.planetscale.com/>

無料プランがなくなる件

- 5 GBまで無料で使用できたが、2024年4月で無料プランの提供が終了
 - <https://planetscale.com/blog/planetscale-forever>
- 最小プランでも月額\$50/月くらいはかかる
- 個人開発で使うには結構辛い

今回の開発で使えるDBサービスの条件

- 今回は個人開発で使用する前提なので以下の条件で調査
 - 個人開発で使用する範疇なら無料で使用できるDBサービス
 - Prismaを使用できる

代替えの候補

- Neon
- TiDB Serverless

Neonとは

- サーバーレスのフルマネージドPostgreSQLサービス
- コードはGitHubで公開されている
 - [neondatabase/neon](https://github.com/neondatabase/neon)
 - Rust製
- AWS AuroraやAlloyDBと似た機能を持ったアーキテクチャを採用（詳しくは以下の記事を参照）
 - <https://neon.tech/blog/architecture-decisions-in-neon>
 - ストレージとコンピューティングを分離した構成で柔軟なスケーリングが可能
- 軽くコンソールを見せてDEMO
 - <https://console.neon.tech/app/projects>

メリット・デメリット

- メリット

- 無料枠が**512MB**
- ブランチングができるので、環境毎にDBを複製ことが可能
- Webのコンソールの機能が充実していて以下の機能が使用できる
 - SQL editor
 - IP制限
 - 定期バックアップ

- デメリット

- 東京リージョンが無い
 - 日本で使用する場合はシンガポールリージョンが一番近くなる

TiDB Serverlessとは

- MySQL互換のあるNewSQLデータベースサービス
- 高い分散性能を持っており、大規模プロジェクトでも使用されている
- コードはGitHubで公開されている
 - <https://github.com/pingcap/tidb>
 - Go製
- 詳細なアーキテクチャは以下の記事を参照
 - <https://docs.pingcap.com/ja/tidb/stable/tidb-architecture#tidb-architecture>
- 軽くコンソールを見せてDEMO
 - <https://tidbcloud.com/console/clusters>

メリット・デメリット

- メリット

- 無料枠が**5000万リクエストユニット (RU) /月**
- NewSQLなのにPrismaが使用できる
- 自動でデータ分散管理されるためシャーディングが不要
- Writeのスケールも可能
- 柔軟なトランザクションの設定が用意されていて用途に合わせて変更可能
 - <https://docs.pingcap.com/ja/tidb/stable/transaction-isolation-levels>
- 運営会社のPingCAPに日本支社ができたので日本語のドキュメントが充実している
 - <https://docs.pingcap.com/ja/tidb/stable/overview>

- デメリット

- MySQLは完全互換では無い（詳細は[こちら](#)）
- 学習コストが高い
- 参考記事: [TiDBにおけるパフォーマンス検証の進め方とつまづきポイント](#)

まとめ

- 初めてNeonとTiDB Serverlessに触ったが、どちらも使いやすくて良かった
- Webコンソール機能的にはNeonの方が充実していたが、東京リージョンがないのが結構痛い
- 今回は**TiDB Serverless**を採用することにした
- 途中移行は難しそうだが、新規プロジェクトでの採用は積極的に検討した方が良さそう
- 特に**TiDB**のアーキテクチャは興味深いので、今後も追いかけていきたい

ご清聴ありがとうございました 🎉