

SupabaseとCloudflare：AIエージェントで作ったものの置き場所

いつ必要になるか、無料枠と落とし穴（一時停止）、RLS、AIとつなぐときの注意（MCP）、公開前の確認とDNSの扱い

更新日：2026-10-03（2026年10月時点の情報） | HUMAC 伊藤晴通

対象ツール：Claude Code

考え方は、どのAIエージェントにも共通です。ただし、**ファイル名・画面・設定・コマンドは、Claude Code**（Claudeのデスクトップアプリの「Code」タブ）の場合です。ほかのAIエージェントでは、読み替えが必要です。

対象の方：開発者ではない方（経営者・事務・現場の担当者）。プログラミングは不要です。**コードは書かず、AIエージェントに頼んで、自分の業務アプリをつくる**ところまでを、順に進めます（ターミナルの黒い画面は、使わない形で書いています）。

AIエージェントで「動くもの」を作ると、置き場所が要る

AIエージェントに、台帳アプリや予約の仕組みを作ってもらくと、必ず「データをどこに置くか」「どうやって公開するか」が問題になる。その定番の置き場所が、**Supabase（スーパーベース）**と**Cloudflare（クラウドフレア）**。

たとえるなら、**Supabase＝「倉庫と受付」**（データをしまい、ログインを受け付ける）、**Cloudflare＝「店舗の入口と看板」**（世界に公開し、配信する）。どちらも、無料から始められる。

1. いつ、必要になるのか

作りたいもの	必要なもの
自分のパソコンの中だけで使う集計・資料づくり	どちらも不要（E1のフォルダと、E4の保存で足りる）
ホームページ・案内ページ（見せるだけ）	Cloudflare（公開と配信）
台帳・予約・顧客管理（データを貯めて、複数人で使う）	Supabase（データ）＋Cloudflare（公開）

作りたいもの	必要なもの
ログインして使う社内アプリ	Supabase（ログインとデータ） + Cloudflare（公開）
AIの鍵（APIキー）を、安全にしまいたい	Supabase の Vault（→ E6）

2. それぞれ、何ができるのか

Supabase：データの置き場所

機能	ひとことで
データベース	表の形でデータをしまう（PostgreSQL）。台帳・顧客・売上など
ログイン（Auth）	メールアドレスなどでのログインを受け付ける
ファイル保存（Storage）	写真やPDFをしまう
Edge Functions	小さなサーバー処理（決まったときに動くプログラム）
Vault	鍵・パスワードを、暗号化してしまう（→ E6）

Cloudflare：公開と配信の置き場所

機能	ひとことで
Workers	小さなプログラムや、ページを、世界中から高速に配信する。Cloudflare が、新しいプロジェクトに最初に勧める基盤（公式）
Pages	ホームページなどの「出来上がったファイル」を公開する。引き続き使えるが、新規はWorkersから始めるのが公式の推奨
保存先（KV・D1・R2）	KV=キーと値の保存、D1=軽いSQLデータベース、R2=ファイル保存
Secrets	鍵を、使う場所に直接しまう。保存後は、値を画面で見られない（→ E6）
ドメイン・DNS	ホームページのアドレスと、メールの宛先を、決める設定
Access	自作のWebアプリに、ログインの関所を付ける仕組み（条件は公式で確認）

3. 無料枠と、落とし穴（2026年10月時点・公式）

	無料枠	落とし穴
Supabase	プロジェクト2つまで／データベース500MB／ファイル1GB／月間利用者5万人／Edge Functions 月50万回	1週間アクセスがないと、自動で一時停止する。 止まると、そのプロジェクトのデータにも、鍵にも届かない。有料（Pro）は月25ドルから
Cloudflare Workers	1日10万リクエスト／1回の処理のCPU時間10ミリ秒／1回のサブリクエスト50まで	人気が出ると、1日10万を超えて止まる。重い処理は、無料枠に収まらない
Cloudflare Pages	ビルド月500回／プロジェクト100／ファイル2万個まで（1つ25MiBまで）	新規はWorkersが推奨（公式）

Supabaseの「一時停止」に注意

無料のプロジェクトは、1週間まったく使われないと止まる。試作のうちは問題ないが、**本番で毎日使うものは、止まると業務が止まる。**長く使うものは、有料プランにするか、止まっても困らない作りにする。鍵の置き場所に使う場合は、特に注意（→ E6）。

4. 最初の一步：AIエージェントに頼む

アカウントを作り（どちらも無料）、最初は「試作用」のプロジェクトで練習する。本番のデータは、入れない。

コピーして使う

〔作りたいもの：例「お客様の予約台帳」〕を、Supabase にデータを置いて、Cloudflare で公開する形で作りたいです。

いきなり作らず、先に私に質問してください（誰が使うか、何人か、個人情報扱うか、ログインは要るか）。

そのうえで、①必要な表、②ログインの有無、③置き場所の構成を、図と短い説明で提案してください。私が確認してから、作り始めてください。

条件：Supabase の表には、必ず「行ごとのアクセス制限（RLS）」を付けてください。鍵は、チャットにもファイルにも書かないでください。

- AIには、Supabaseと「つなぐ道具」（MCP）を渡せる。ただし、後述のとおり、最初は試作用に限る。
- 作ったものは、E1のフォルダに置き、E4の方法で保存する。

- 無料枠に収まらない・特別なソフトや24時間の処理が要る場合は、**自分専用のサーバー（VPS）**を借りる選択肢がある（→ E7）。

5. 安全に使う：AIが作ったアプリで、いちばん多い事故

Supabase：「RLS」を必ず付ける

Supabaseは、データベースを、インターネット経由のAPIとして公開する仕組み。そのため、公式が次のように警告している：「**行ごとのアクセス制限（RLS）が無い表は、権限を持つ誰でも、読み書きできる**」。AIが作った表に、RLSが付いていないと、**お客様のデータが、誰にでも見える・書き換えられる状態になる**。

鍵の種類	使う場所	ルール
公開用の鍵（publishable／anon）	ブラウザ・スマホのアプリ	見えても大丈夫な、最小限の権限。RLSで守られていることが前提
秘密の鍵（secret／service_role）	サーバー側だけ	RLSを無視して、すべてを操作できる。ブラウザやスマホに、絶対に出さない（公式）

コピーして使う

この Supabase のプロジェクトを点検してください。①RLS が無効なテーブルを一覧にしてください②各テーブルに、誰が何をできるかのポリシーがあるか③秘密の鍵（service_role）が、ブラウザ側のコードや公開ファイルに入っていないか。結果を表で報告し、直す前に、私の確認を取ってください。値（鍵そのもの）は、表示しないでください。

AIとSupabaseをつなぐとき（MCP）

Supabase の公式は、AIをデータベースにつなぐ危険性を、はっきり書いている。**お客様が入力した文章の中に、悪意のある命令が紛れていると、AIがそれに従って、データを読み出したり、消したりしかねない（プロンプトインジェクション）**。公式の推奨は次のとおり。

- **本番のデータには、必要なときだけつなぐ**。プロジェクトを指定し、**読み取り専用**にする。
- **道具を呼ぶたびの承認**を、残しておく（自動承認にしない）。
- 使う機能を、必要なものだけに絞る。
- 試すときは、**本番ではなく、開発用の枝（ブランチ）**で。
- AIの出力は、実行する前に、人が確認する。

AIの画面に、鍵や個人情報が出ると、会話の記録に残る

AIに「データを見せて」と頼んで、表の中身（個人情報や鍵）が表示されると、それは**会話の記録に残り、モデルの提供会社にも送られる**。本番のデータや鍵は、AIに表示させない（赤の情報。D1）。講師は、鍵を表示させた事故の教訓から、「鍵は見ない・表示させない」をルールにしている（→ E2、E6）。

Cloudflare：公開前の確認と、DNSの扱い

- **公開する前に、中身を確認する**：ホームページなどを公開する操作は、人が押す（関所。E2）。まず、確認用のURL（プレビュー）に出して、人が見てから、本番にする。
- **鍵は、Secretsに入れる**：WorkersのSecretsは、保存すると、**画面でも、コマンドでも、値を読み出せない**（置き換えるだけ）。環境変数に、鍵を平文で置かない（公式）。
- **ドメイン・DNS・メールの設定は、慎重に**：ドメインの設定を間違えると、**ホームページだけでなく、メールが止まる**ことがある。変更の前に、いまの設定を記録し、AIに任せきりにせず、人が確認する。特に、メールの宛先（MXレコード）には触らない。
- ログインを付けたいときは、**Cloudflare Access**のような「関所」の仕組みがある。

6. どちらを使うか、迷ったら

1

Q1 データを貯める？

はい → Supabase。いいえ → Q2へ

2

Q2 公開する？

はい → Cloudflare（新規はWorkersから）

3

Q3 ログインが必要？

はい → SupabaseのAuth、またはCloudflare Access

4

Q4 個人情報を扱う？

はい → RLS必須。本番は有料プランも検討。まずD1

7. はじめての日のチェックリスト

☐ 確認すること

- ☐ 作りたいものに、データベース・公開が本当に必要か、先にAIに質問してもらった
- ☐ 試作用のプロジェクトで始め、本番データは入れていない
- ☐ すべての表に、RLSが付いている（AIに点検してもらった）
- ☐ 秘密の鍵が、ブラウザ側・公開ファイル・チャットに出ていない
- ☐ AIをSupabaseにつなぐときは、試作用・読み取り専用・承認あり

☐	確認すること
☐	公開は、プレビューで確認してから、人が押した
☐	無料枠の上限と、「一時停止」の条件を、理解した
☐	鍵の置き場所を決めた（→ E6）

関連資料

- E6 APIキー（秘密）の扱い … .env に頼らず、Vault に置く
- E1 はじめてのAIエージェント：フォルダ構成と管理
- E4 Gitの導入と設定
- D1 信号機ルール／E2 ハーネスエンジニアリング入門

出典（2026年10月確認）

- Supabase 料金（無料枠・一時停止・Pro） (<https://supabase.com/pricing>)
- Supabase Docs：Row Level Security (<https://supabase.com/docs/guides/database/postgres/row-level-security>)
- Supabase Docs：MCP（セキュリティ上の注意） (<https://supabase.com/docs/guides/getting-started/mcp>)
- Cloudflare Workers：制限（無料プラン） (<https://developers.cloudflare.com/workers/platform/limits/>)
- Cloudflare Pages：制限 (<https://developers.cloudflare.com/pages/platform/limits/>)
- Cloudflare Pages（新規はWorkersを推奨） (<https://developers.cloudflare.com/pages/>)
- Cloudflare Workers：保存先の選択肢 (<https://developers.cloudflare.com/workers/platform/storage-options/>)
- Cloudflare Workers：Secrets (<https://developers.cloudflare.com/workers/configuration/secrets/>)
- Cloudflare Access：自作アプリの認証 (<https://developers.cloudflare.com/cloudflare-one/applications/configure-apps/self-hosted-public-app/>)