
今ほど、プレビュー環境が 重要な時代はない

Vercel × Supabase による個人開発の実践から

Yoshifumi Kondo

スタートアップ Acting Manager / エンジニア

個人開発が好きで、iOSアプリ2本・Webサービス2本をリリース。

今の推しは [KORENANI](#) — AIカメラで9言語を学べる子ども向けアプリ。



ソロプレナーラボ — 個人開発を学んで実践するPodcastもやってます

X [@ysfmkndon](#)

アジェンダ

01 サービス全体の挙動は、人が見て判断するしかない

02 PRごとのプレビュー環境が人の判断を支える

03 まとめ

01

サービス全体の挙動は、
人が見て判断するしかない

コードの確認と挙動の確認は別物

コードの中身を確認する

- テスト (Unit / E2E)
- コードレビュー
- 静的解析・型チェック

→ すでにAIが一部担っている。
今後さらに自動化が進む

サービス全体の挙動を確認する

- 実際に触って体験する
- 全レイヤーが連動した動きを見る
- 「使ってみてどうか」を判断する

→ **ここは、しばらく人に残る**

AI時代、PRが並列に増え、人の確認が追いつかない

各エージェントが worktree を切って独立作業 → PR作成まで完結

Agent A

worktree A

→ PR #1

Agent B

worktree B

→ PR #2

Agent C

worktree C

→ PR #3

コードは自動で検証できる。挙動は誰が確認する？

個人開発では全体の体験を丸ごと確認したい

01

レイヤーごとに分けて確認するほど**成熟していない**

個人開発のサービスは、全体で体験を確認するフェーズがほとんど

02

DB・API・CRON・Webhookが**連動した体験**を見たい

フロントだけ動いても、全体としてどうなるかは分からない

03

portlessなど**ローカルも進化**しているが...

worktreeごとにURLが分かれても、全レイヤーの統合はローカルでは再現しきれない

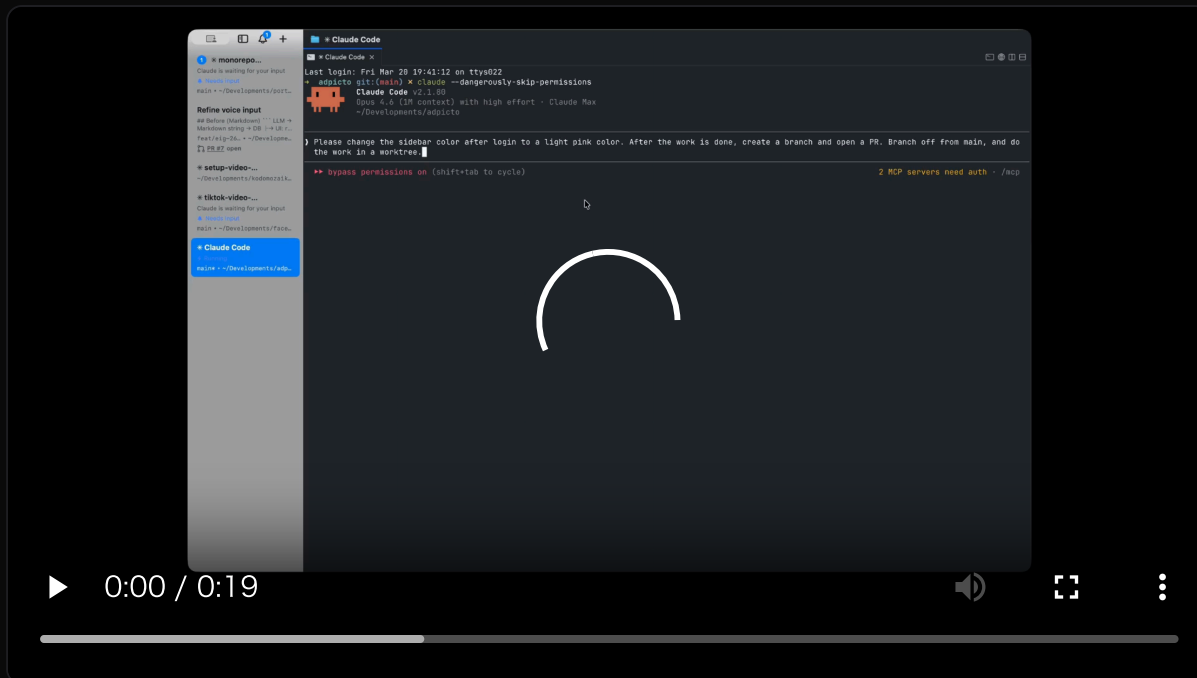
02

PRごとのプレビュー環境が
人の判断を支える

コードはPRで確認する

挙動はプレビュー環境で確認する

デモ

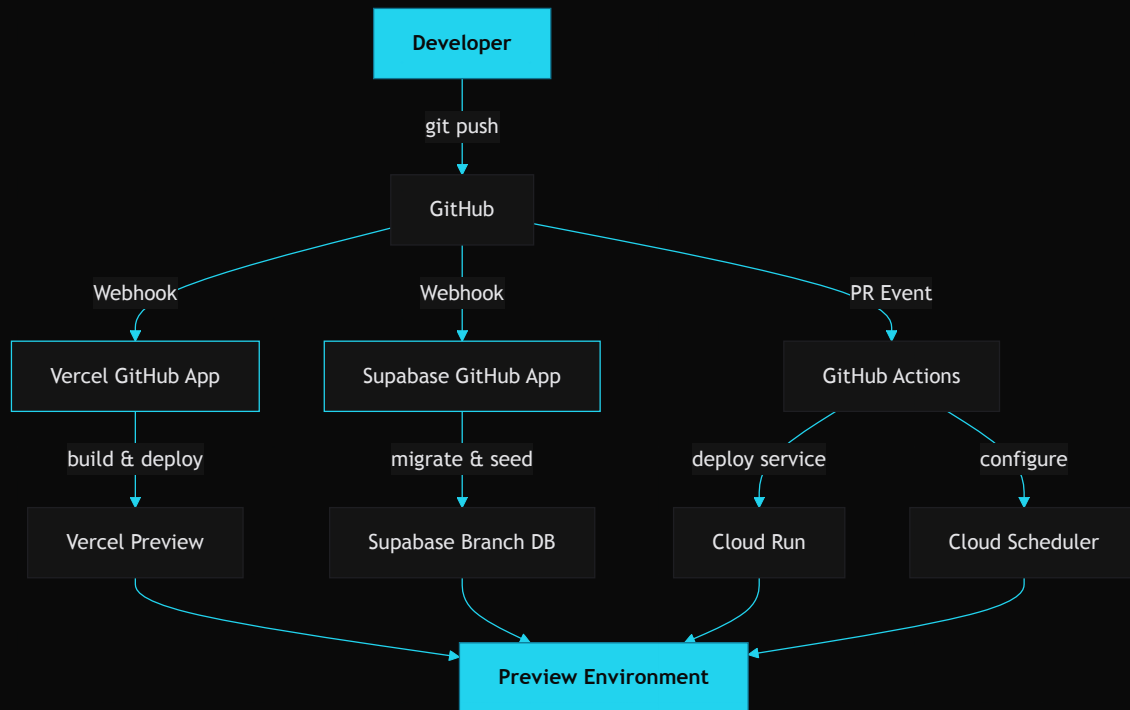


1. エージェントが実装・commit・push・PR作成まで自動実行

2. Vercel自動デプロイ + Supabase ブランチDBにseedデータ注入

3. サンプルデータが入った状態で実際の挙動を目視確認

git pushだけで全レイヤーが自動構築される



GITHUB APP

Vercel / Supabase — Webhook で自動。Actions 不要

GITHUB ACTIONS

Cloud Run / Scheduler など自前インフラのみ

PRごとに全レイヤーが独立で立ち上がる

GITHUB APP

Vercel

- Webhookでビルド自動実行
- ブランチごとの固有Preview URL
- PRマージ/クローズで自動失効

GITHUB APP

Supabase

- PRごとにPostgresインスタンス分岐
- migrations/ を自動適用 + seed.sql 実行
- PRクローズで自動削除

GITHUB ACTIONS

Cloud Run

- ブランチ名ベースのサービス起動
- オートスケーリング設定
- PRクローズ時に自動削除

GITHUB ACTIONS

その他

- Cloud Scheduler / Secrets 注入
- E2E テスト実行
- PRクローズで自動削除

すべてのリソースがPRのライフサイクルに同期 — **手動作業ゼロ**

ローカルで足りない部分をプレビュー環境が補う

ローカルでできること

- ✓ 個別コンポーネントの動作確認
- ✓ portlessでworktreeごとのURL
- ✓ 高速なイテレーション

- ✗ Webhook・CRON・外部API
- ✗ 本番同等のDB連携
- ✗ 全レイヤー統合の体験確認

プレビュー環境が補完すること

- ✓ PRごとに全レイヤーが立ち上がる
- ✓ 公開URLでWebhook・CRON実行
- ✓ ブランチDBでマイグレーション検証
- ✓ サービス全体の体験を目視確認
- ✓ 並列PRを同時に確認可能

GitHub Appを連携するだけで始められる



GitHub App を連携

Vercel と Supabase を
リポジトリに接続するだけ
GitHub Actionsは不要



自前インフラだけ Actions

Cloud Run / Scheduler など
GitHub App がない部分だけ
ワークフローを書く



PR を出すだけ

フロント・DB・バックエンドが
自動で立ち上がり
プレビューURLが届く



03

まとめ

01

動いているのを見てから マージする

テスト通過だけで判断しない
サービス全体の体験を人の目で確認してからマージする

02

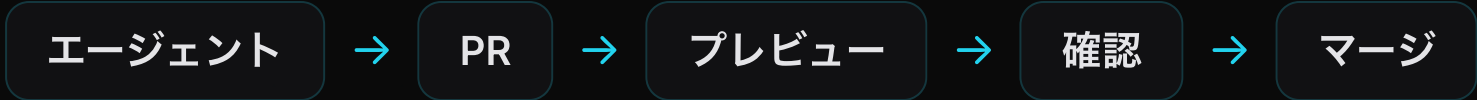
PRごとに独立した環境を作る

DB・API・フロント・スケジューラ
全レイヤーがブランチ単位で立ち上がる

03

ライフサイクルを自動化する

作成も削除もPRイベントに紐づける
手動管理は負債になる




書いたコードを安心して確認できる仕組みを持とう

ありがとうございました

Yoshifumi Kondo

✕ [@ysfmkndon](https://twitter.com/ysfmkndon)

 [Podcast on Spotify](https://open.spotify.com/show/ysfmkndon)