

株式分析プラットフォーム

— 注目銘柄に絞った分析アプローチ —

1 背景

個人開発の株価分析ツールは数多く存在するが、
コインストス（勝率50%）を超えるものは少ない。

2 制約

yFinance等の**API制限**により、
東証全銘柄の株価データを取得することが困難。

3 着眼点

全銘柄を網羅的に分析するより、
「注目が集まる銘柄」に絞った方が
効率的ではないか？

4 アプローチ

静的銘柄

- Core30 (TOPIX核心銘柄)
- 政策銘柄 (Claude Codeで選定)

動的銘柄

毎日23時に**Grok (xAI API)** で「Xで話題の上昇期待銘柄」を15-20銘柄選定

5 仮説

注目が集まる銘柄に
絞って分析することで、

**コインストス以上の
勝率・リターン**

が期待できるのではないか。

株式分析プラットフォーム

– 技術スタック & 開発スタイル –

技術スタック

Backend Python FastAPI

Frontend TypeScript Next.js Vercel

Infrastructure AWS Terraform Docker

Auth Cognito Passkey

CI/CD GitHub Actions

AI xAI Grok Claude

開発スタイル

Vibe Coding

AIと対話しながらコードを書く開発手法。
要件を伝えるだけで設計からデプロイまでAIがサポート。

ハイライト

Grok AI

毎日の銘柄選定 + マーケットサマリー自動生成

AI駆動開発

Vibe Coding + IaC (Terraform) でインフラも含めAI対応

yFinance

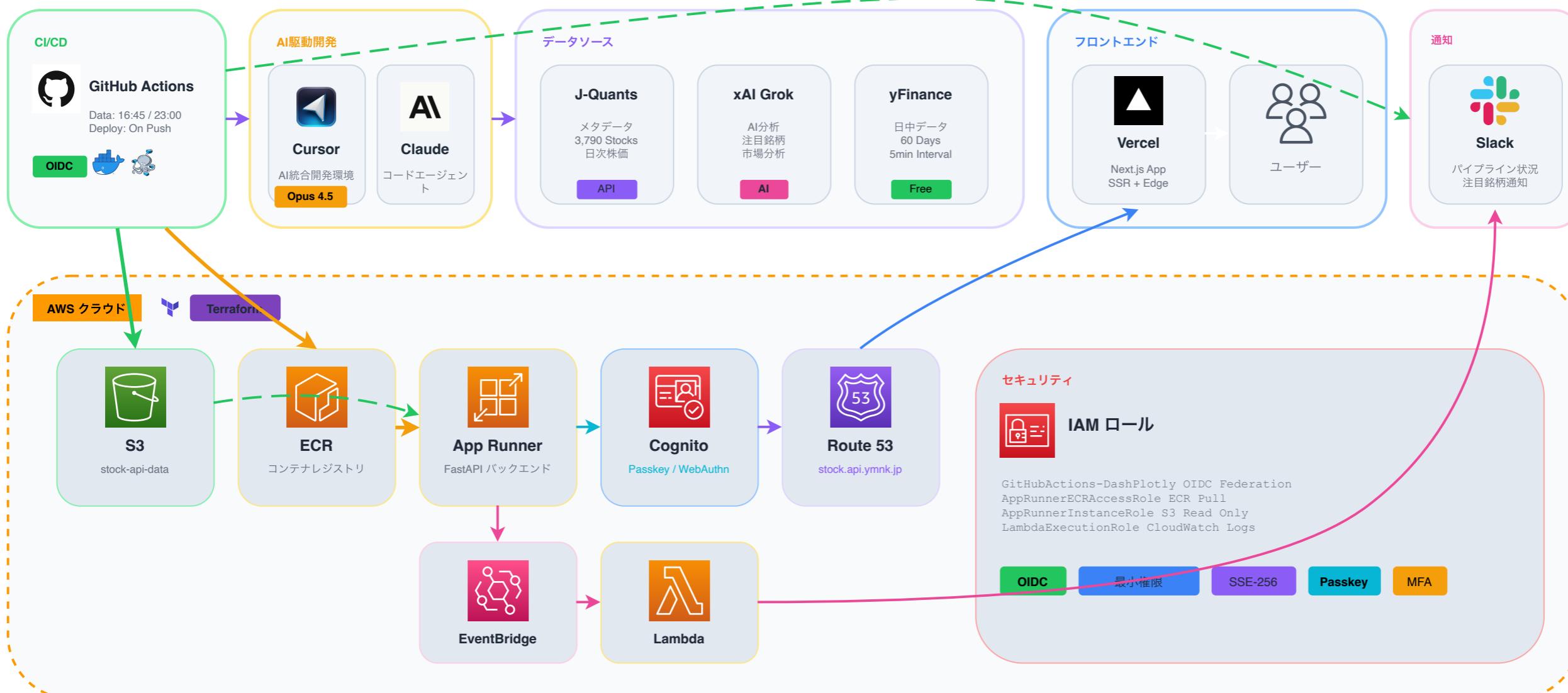
長期分析 + リアルタイムデータ (バックエンド/フロントエンド両対応)

Passkey + Cognito

ログイン管理 + パスワードレス認証で堅固なセキュリティ

株式分析プラットフォーム

フルスタック / リアルタイム分析 / 自動パイプライン / Vibe Coding



データフロー

データパイプライン

- GitHub Actions 16:45/23:00 JST 実行
- J-Quants, yFinance, Grok からデータ取得
- Parquet 形式で S3 へアップロード
- 結果を Slack へ通知

デプロイ

- main ブランチへの push でデプロイ開始
- Dockerイメージをビルド、ECRへプッシュ
- App Runner が自動デプロイ
- EventBridge → Lambda → Slack 通知

API + 認証

- ユーザーが Vercel フロントエンドにアクセス
- Cognito で Passkey/WebAuthn 認証
- フロントエンドが stock.api.ymnk.jp を呼出
- App Runner が S3 を読み取り JSON を返却

技術スタック

Backend: FastAPI (Python)
Frontend: Next.js (Vercel)
Data: Parquet on S3
Auth: Cognito + Passkey
IaC: Terraform
AI: xAI Grok, Claude Opus

開発環境

Editor: Cursor
Agent: Claude Code
Model: Opus 4.5
Style: Vibe Coding



Terraform on AWS

ap-northeast-1

Stock Analysis Platform / 22 Resources Managed

