

# 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	

## 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

## 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。

# 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

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

### デプロイ

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

### API + 認証

1. ユーザーがVercelフロントエンドにアクセス
2. Cognitoで Passkey/WebAuthn認証
3. フロントエンドがstock.api.ymnk.jpを呼出
4. 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

Stock Analysis Platform / 22 Resources Managed

ap-northeast-1

AWS CLOUD

STORAGE




**Amazon S3**


stock-api-data

Versioning · SSE-256 · Lifecycle


COMPUTE



**ECR**  
Container Registry




**App Runner**  
Auto Scaling API



**Lambda**  
Notifications


NETWORKING



**Route 53**

stock.api.ymnk.jp

AUTHENTICATION



**Cognito**


User Pool + WebAuthn

Passkey

OAuth 2.0


MFA

EVENTS




EventBridge

CI/CD



**GitHub Actions**

SECURITY



**IAM · 4 Roles**

OIDC

TERRAFORM STATE

22

Resources

4

IAM Roles

7

Services

```
aws_s3_bucket · aws_ecr_repository · aws_apprunner_service · aws_route53_zone · aws_eventbridge_rule · aws_lambda_function · aws_cognito_user_pool · aws_cognito_user_pool_client (x2) · aws_iam_role (x4) · aws_iam_policy (x3)
```

Terraform

Local State

Tokyo