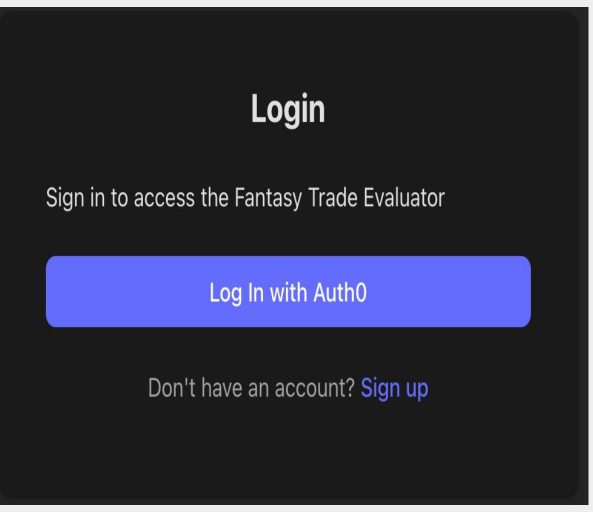
LevelField
Cloud-Native Full-Stack Deployment

Michael Davis, Stephen Kain, Vance Keesler, Matthew Sheehan

LevelField is a lightweight fantasy football trade evaluation tool designed to promote fair, data-driven trades in fantasy leagues. It combines live player data, user authentication, and automated updates to provide a trade evaluation tool.

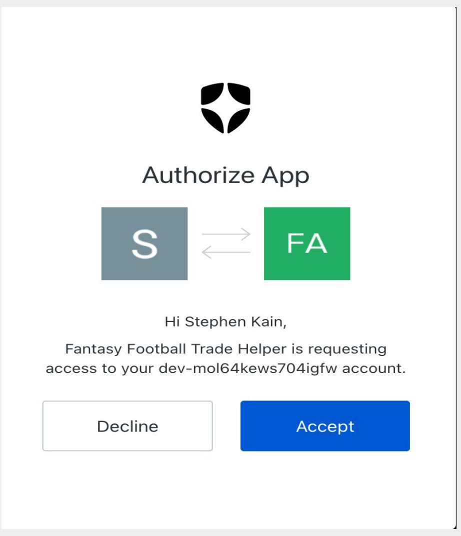


Login

Sign in to access the Fantasy Trade Evaluator

Log In with Auth0

Don't have an account? Sign up

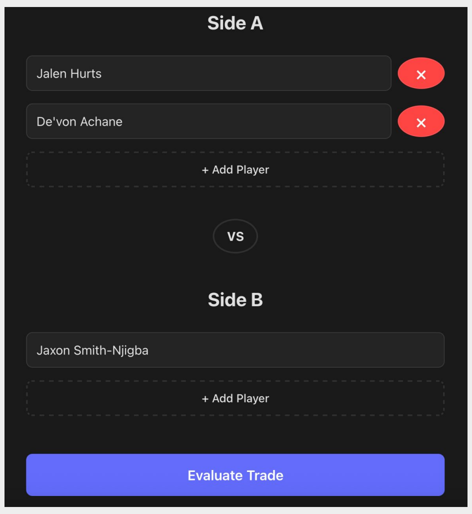


Authorize App

Hi Stephen Kain,

Fantasy Football Trade Helper is requesting access to your dev-mol64kews704igfw account.

Decline | Accept



Side A

Jalen Hurts

De'von Achane

+ Add Player

vs

Side B

Jaxon Smith-Njigba

+ Add Player

Evaluate Trade

System Architecture

Frontend (React + Vite + NGINX):
- Auth0 login + session persistence
- Stateless multi-stage Docker build
- NodePort Service (30080) serves UI
- Calls backend API via /api

Backend (Node.js + Logging Sidecar):
- Trade evaluation logic
- Provides REST API endpoints
- Auth0 token verification middleware
- Fluent Bit logging sidecar for centralized log collection
- MongoDB persistence (trade history)

Storage Layer (MongoDB + Python Script):
- Mongo deployment + ClusterIP service
- PersistentVolume + PersistentVolumeClaim for durable storage

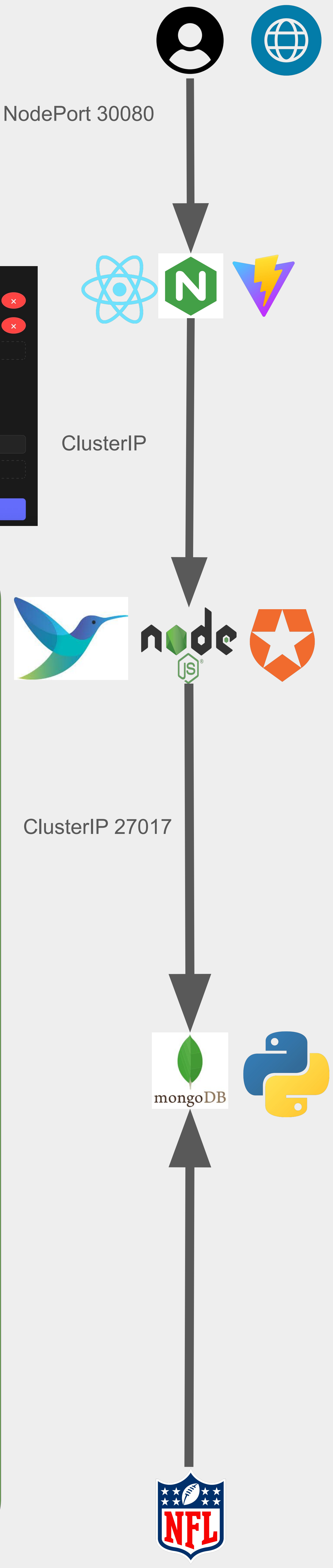
Data Ingestion Layer
- Kubernetes CronJob runs hourly
- Python container fetches NFL player data from Sleeper API
- Inserts refreshed data into MongoDB
- Uses shared MONGO_URI to access same database

Infrastructure (Kubernetes)
- Separate Deployments for frontend, backend, MongoDB, CronJob
- Internal DNS for service discovery
- Persistent storage for long-term data retention
- imagePullPolicy: Always ensures new images are pulled on pod restart
- NodePort access for external traffic

Key Features

- Auth0 secure login
- Automated NFL data ingestion via CronJob
- Persistent trade history stored in MongoDB
- Stateless and isolated service components
- Internal DNS-based communication (backend-service, mongo-service)
- Clean separation between UI, API, storage, and scheduled tasks

Kubernetes Architecture



NodePort 30080
ClusterIP
ClusterIP 27017
node JS
mongoDB
NFL