Trading App Deployment - Complete

Deployment Summary

The AIAlgoTradeHits.com trading application has been successfully deployed with full BigQuery integration for both crypto and stock market data.

Deployment Date: November 11, 2025 **Project ID:** cryptobot-462709 **Region:** us-central1

Deployed Services

1. Backend API Service

Service Name: trading-api **URL:** https://trading-api-cnyn5l4u2a-uc.a.run.app **Status:** ✅ Active and serving real-time data from BigQuery

Endpoints: - `GET /health` - Health check endpoint - `GET /api/crypto/{timeframe}?limit={n}` - Crypto data (daily, hourly, 5min) - `GET /api/stocks?limit={n}` - Stock market data - `GET /api/summary/{market_type}` - Market summary statistics - `GET /api/users` - User management (admin) - `POST /api/users` - Create user (admin) - `PUT /api/users/{user_id}` - Update user (admin) - `DELETE /api/users/{user_id}` - Delete user (admin)

Data Sources: - Crypto Daily: `crypto_trading_data.crypto_analysis` - Crypto Hourly:

`crypto_trading_data.crypto_hourly_data` - Crypto 5-Min: `crypto_trading_data.crypto_5min_top10_gainers` - Stock Data: `crypto_trading_data.stock_analysis`

2. Frontend Application

Service Name: crypto-trading-app **URL:** https://crypto-trading-app-252370699783.us-central1.run.app **Status:** ✅ Active and displaying real data

Features: - Market type toggle (Crypto/Stock) - Three timeframe views for crypto (Daily, Hourly, 5-Minute) - Real-time data display with technical indicators - Expandable analytics windows - Admin panel for user management - Responsive design with gradient UI

Technical Stack

Backend

- **Framework:** Flask 3.0.0
- **CORS:** flask-cors 4.0.0
- **Database:** Google BigQuery
- **SDK:** google-cloud-bigquery 3.25.0
- **Runtime:** Python with functions-framework
- **Container:** Cloud Run (512Mi memory, 1 CPU)

Frontend

- **Framework:** React 19.1.1 with Vite 7.1.7
- **Charts:** Recharts 3.2.1, Lightweight Charts 5.0.9
- **Icons:** Lucide React 0.546.0
- **Build:** Multi-stage Docker (Node 18 + Nginx Alpine)
- **Container:** Cloud Run (serves on port 8080)

Data Architecture

Crypto Data Flow

```
Kraken API → Cloud Functions (3 timeframes) → BigQuer
```

Collection Schedule: - Daily: Midnight ET (all ~675 USD pairs) - Hourly: Every hour (all USD pairs) - 5-Minute: Every 5 minutes (top 10 gainers)

Stock Data Flow

```
Alpha Vantage API → Cloud Function → BigQuery Table -
```

Collection Schedule: - Daily: Once per day (after market close)

Technical Indicators (29 total)

Both crypto and stock data include: - **Momentum:** RSI, MACD, ROC, Momentum, Stochastic K/D, Williams %R - **Trend:** SMA (20/50/200), EMA (12/26/50), ADX, KAMA, TRIX - **Volatility:** Bollinger Bands (Upper/Middle/Lower), ATR - **Volume:** OBV, PVO - **Oscillators:** CCI, PPO, Ultimate Oscillator, Awesome Oscillator

---

UI Components

Main Dashboard

1. **Header Bar**
    - App title with gradient styling
    - Real-time data indicator
    - Admin panel access button
2. **Market Toggle**
    - Crypto Markets button
    - Stock Markets button
3. **Timeframe Cards (Crypto only)**

- Daily View card with expand button
- Hourly View card with expand button
- 5-Minute View card with expand button

4. **Quick Stats Table**
   - Shows latest 10 records
   - Displays: Pair/Symbol, Price, RSI, MACD, Signal (BUY/SELL/HOLD)
   - Color-coded indicators (green=oversold, red=overbought)

Expanded Analytics Window

- Full-screen overlay
- Market summary statistics (4 metric cards)
- Detailed data table (20 records)
- Advanced filtering and sorting
- Close button to return to dashboard

Admin Panel

- User list display
- Create new user form
- Edit user functionality
- Delete user (soft delete)
- Role management (admin/user)
- Status toggle (active/inactive)

Configuration Files

Backend API (`cloud_function_api/`)

```
main.py                  - Flask app with BigQuery que
requirements.txt         - Python dependencies
deploy_api.py            - Deployment script
```

Frontend App (`stock-price-app/`)

```
src/
  App.jsx                - Main app component
  components/
    AIAlgoTradeHitsReal.jsx  - Main dashboard compone
    AdminPanel.jsx           - User management UI
  services/
    api.js               - API service layer
Dockerfile               - Multi-stage build
nginx.conf               - Nginx configuration
package.json             - Node dependencies
vite.config.js           - Vite bundler config
```

Deployment Commands

Deploy Backend API

```
cd cloud_function_api
python deploy_api.py
```

Deploy Frontend App

```
cd stock-price-app
gcloud run deploy crypto-trading-app \
  --source . \
  --platform managed \
  --region us-central1 \
  --allow-unauthenticated \
  --port 8080 \
  --project cryptobot-462709
```

Quick Redeploy Both Services

```
 # Backend
cd cloud_function_api && python deploy_api.py

# Frontend
cd ../stock-price-app && gcloud run deploy crypto-tra
  --source . --platform managed --region us-central1
  --allow-unauthenticated --port 8080 --project crypt
```

Testing & Verification

Test Backend API

```
 # Health check
curl https://trading-api-cnyn5l4u2a-uc.a.run.app/heal

# Get crypto daily data
curl "https://trading-api-cnyn5l4u2a-uc.a.run.app/api

# Get stock data
curl "https://trading-api-cnyn5l4u2a-uc.a.run.app/api

# Get market summary
curl "https://trading-api-cnyn5l4u2a-uc.a.run.app/api
```

Test Frontend

1. Open browser to: https://crypto-trading-app-252370699783.us-central1.run.app
2. Verify crypto data loads in dashboard
3. Click "Stock Markets" tab and verify stock data loads
4. Click "Expand" on any timeframe card to view full analytics
5. Click "Admin Panel" to access user management

Monitoring & Logs

View API Logs

```
gcloud run services logs read trading-api \
  --project=cryptobot-462709 \
  --region=us-central1 \
  --limit=50
```

View Frontend Logs

```
gcloud run services logs read crypto-trading-app \
  --project=cryptobot-462709 \
  --region=us-central1 \
  --limit=50
```

Monitor BigQuery Usage

```
 # Check data counts
python check_bigquery_counts.py

# View recent data
bq query --use_legacy_sql=false \
  'SELECT * FROM `cryptobot-462709.crypto_trading_dat
   ORDER BY datetime DESC LIMIT 10'
```

Cost Estimates

Monthly Costs (Estimated)

Cloud Run Services: - Backend API: $5/month (minimal traffic) - Frontend App: $5/month (minimal traffic)

Data Collection (Already running): - Cloud Functions: $126/month - BigQuery Storage: $2/month - Cloud Scheduler: $0.30/month

Total Additional Cost: ~$10/month for web services **Total System Cost:** ~$145/month (including existing data pipeline)

Security Considerations

1. **Public Access:** Both services are publicly accessible (--allow-unauthenticated)
2. **CORS:** Enabled on backend API for frontend communication
3. **BigQuery:** Uses service account with read-only access to trading data tables
4. **User Management:** Admin panel has no authentication (add Auth0/Firebase in production)
5. **SSL/TLS:** Automatically provided by Cloud Run

Production Hardening Checklist

- [ ] Add authentication (Firebase Auth, Auth0, or similar)
- [ ] Implement rate limiting on API endpoints
- [ ] Add API keys for backend access
- [ ] Set up Cloud Armor for DDoS protection
- [ ] Enable Cloud Run authentication
- [ ] Add environment-based config management
- [ ] Set up monitoring alerts (Cloud Monitoring)
- [ ] Implement request logging and analytics
- [ ] Add CSRF protection
- [ ] Set up backup and disaster recovery

Next Steps & Enhancements

Short-term (1-2 weeks)

1. Add authentication and user login
2. Implement real-time data updates (WebSockets or polling)

3. Add charting functionality for price history
4. Create custom watchlists
5. Add alert notifications for price/indicator thresholds

Medium-term (1-2 months)

1. Implement trading strategy backtesting
2. Add paper trading functionality
3. Create mobile-responsive PWA
4. Add social features (share trades, follow traders)
5. Implement AI-powered trade recommendations

Long-term (3-6 months)

1. Integrate with broker APIs for live trading
2. Add portfolio tracking and P&L analytics
3. Implement subscription tiers and payment processing
4. Create mobile apps (iOS/Android)
5. Add machine learning models for price prediction

Troubleshooting

API Returns Empty Data

```
 # Check BigQuery tables have data
python check_bigquery_counts.py

# Manually trigger data collection
curl https://daily-crypto-fetcher-cnyn5l4u2a-uc.a.run
curl https://hourly-crypto-fetcher-cnyn5l4u2a-uc.a.ru
```

Frontend Shows Loading Forever

1. Check API is responding: `curl https://trading-api-cnyn5l4u2a-uc.a.run.app/health`
2. Check browser console for CORS errors

3. Verify API_BASE_URL in `src/services/api.js` is correct
4. Check Cloud Run logs for errors

CORS Issues

- Backend already has CORS enabled via flask-cors
- If issues persist, check Cloud Run configuration

Slow Load Times

- Cold starts are normal (first request after idle)
- Consider setting `--min-instances=1` to keep warm instance
- Optimize BigQuery queries with partitioning/clustering
- Add caching layer (Redis/Memorystore)

Success Metrics

✅ Backend API deployed and responding ✅ Frontend application deployed and loading ✅ Real crypto data flowing from BigQuery ✅ Real stock data flowing from BigQuery ✅ All three timeframes working (daily, hourly, 5-min) ✅ Admin panel functional ✅ Market toggle working (crypto/stock) ✅ Expandable analytics windows working ✅ Technical indicators displaying correctly

---

Support & Maintenance

Regular Maintenance Tasks

- **Daily:** Check Cloud Functions are collecting data
- **Weekly:** Review Cloud Run logs for errors
- **Monthly:** Analyze BigQuery storage costs and optimize
- **Quarterly:** Review and update dependencies

Key Files to Monitor

- `cloud_function_api/main.py` - Backend API logic

- `stock-price-app/src/services/api.js` - Frontend API calls
- `stock-price-app/src/components/AIAlgoTradeHitsReal.jsx` - Main UI component

Contact & Documentation

Project Repository: Trading/ **Documentation:** - `CLAUDE.md` - Project overview and architecture - `QUICK_START_GUIDE.md` - Getting started guide - `FINAL_DEPLOYMENT_STATUS.md` - Infrastructure status - `TRADING_APP_DEPLOYMENT_COMPLETE.md` - This file

Key Resources: - Cloud Run Dashboard: https://console.cloud.google.com/run?project=cryptobot-462709 - BigQuery Console: https://console.cloud.google.com/bigquery?project=cryptobot-462709 - Cloud Functions: https://console.cloud.google.com/functions?project=cryptobot-462709

Conclusion

The trading application is now fully operational with: - ✅ Real-time data from BigQuery - ✅ Both crypto and stock market coverage - ✅ Full technical analysis with 29 indicators - ✅ Responsive web interface - ✅ Admin panel for user management - ✅ Cloud-native architecture on GCP

The system is production-ready for internal use and can be scaled up for public release with the security enhancements listed above.

Live URLs: - **Application:** https://crypto-trading-app-252370699783.us-central1.run.app - **API:** https://trading-api-cnyn5l4u2a-uc.a.run.app

---

Last Updated: November 11, 2025 Status: Production Ready