Options Scanner

Raspberry Pi Deployment Guide

Prepared: February 27, 2026 | Branch: main @ c658ef5

Context for Next Agent

The Options Scanner codebase has just completed a comprehensive audit remediation (39 findings fixed across 23 files). The code is merged to **main** on the local workspace and needs to be pushed to GitHub, then pulled onto the Raspberry Pi, where the Docker image must be rebuilt and restarted.

> **IMPORTANT:** requirements.txt changed (dependencies removed, versions pinned, re-encoded from UTF-16LE to UTF-8). A **--no-cache** Docker build is required.

Repository Information

Item	Value
GitHub Remote	https://github.com/horlash/Options.git
Current Branch	main
Latest Commit	c658ef5 — Merge fix/audit-remediation-v2
Docker Image	horlamy/options-scanner:1.0.1
Container Name	leap_scanner_prod
App Port	5000
Python Version	3.12-slim
Database	SQLite (scanner) + PostgreSQL (paper trading)

Docker Compose Files

File	Purpose	Command
docker-compose.yml	Production: scanner + SQLite	docker compose up -d
docker-compose.dev.yml	Dev: full stack + PostgreSQL on :5433	docker compose -f docker-compose.dev.yml up -d
docker-compose.paper.yml	Paper trading DB only: PostgreSQL on :5432	docker compose -f docker-compose.paper.yml up -d

Deployment Steps

Step 1: Push to GitHub (from workspace or local machine)

If the workspace still has the repo, push main to origin:

```
cd /home/user/workspace/Options
git push origin main
```

If the agent does not have push access (no SSH key or token configured), the user must push manually from a machine that does. Alternatively, create a Personal Access Token and use:

```
git remote set-url origin https://<TOKEN>@github.com/horlash/Options.git
git push origin main
```

Step 2: SSH into Raspberry Pi

The agent will need the Pi's IP address and SSH credentials. Ask the user for:

- Pi IP address or hostname (e.g., 192.168.x.x or raspberrypi.local)
- SSH username (typically **pi** or a custom user)
- SSH password or key path

```
ssh <username>@<pi-ip-address>
```

Step 3: Pull Latest Code on Pi

Navigate to the project directory on the Pi and pull:

```
cd ~/Options # or wherever the repo lives on the Pi
git pull origin main
```

If there are local changes on the Pi that conflict, stash them first:

```
git stash
git pull origin main
git stash pop
```

Step 4: Rebuild Docker Image (REQUIRED — no-cache)

Critical: requirements.txt was re-encoded and dependencies changed. A cached build will use stale layers. You MUST use --no-cache.

```
docker compose down
docker compose build --no-cache
docker compose up -d
```

Or as a single command:

```
docker compose down && docker compose build --no-cache && docker compose up -d
```

Note on ARM build time: Building on a Raspberry Pi (ARM64) can take 15-30 minutes due to compiling native Python packages (numpy, pandas, TA-Lib). This is normal.

Step 5: Verify the Container is Running

```
docker ps
docker logs leap_scanner_prod --tail 50
```

Expected output should show:

- Container **leap_scanner_prod** status: Up
- Port 5000 mapped
- No import errors or crash loops in logs

Step 6: Verify Paper Trading DB (if using PostgreSQL)

If the paper trading PostgreSQL container is also used:

```
docker compose -f docker-compose.dev.yml up -d paper_db
# or
docker compose -f docker-compose.paper.yml up -d
```

Step 7: Quick Smoke Test

From the Pi (or any machine on the same network):

```
# Health check
curl http://localhost:5000/api/health

# Scan a single ticker
curl -X POST http://localhost:5000/api/scan/ticker \
-H 'Content-Type: application/json' \
-d '{"ticker": "AAPL", "direction": "CALL"}'
```

Environment Configuration

The **.env** file on the Pi should contain all API keys and config. It is NOT in git (.gitignore excludes .env files). Verify it exists:

```
cat ~/Options/.env
```

Required environment variables:

Variable	Purpose	Required
ORATS_API_KEY	Options chain data, IV, earnings	Yes
FINNHUB_API_KEY	News sentiment, institutional data	Yes
TRADIER_API_KEY	Broker integration (sandbox)	Yes
TRADIER_USE_SANDBOX	True for paper, False for live	Yes
FMP_API_KEY	Financial Modeling Prep (sector scan)	Yes
PAPER_TRADE_DB_URL	PostgreSQL connection string	For paper trading
PERPLEXITY_API_KEY	AI analysis (sonar-pro model)	Optional
SECRET_KEY	Flask session key	Yes
FLASK_DEBUG	False for production	Yes
PORT	5000	Yes

Troubleshooting

Symptom	Likely Cause	Fix
Container exits immediately	Missing .env or bad import	docker logs leap_scanner_prod
ModuleNotFoundError	Stale Docker cache	Rebuild with --no-cache
requirements.txt encoding error	Old UTF-16LE file cached	docker compose build --no-cache
Port 5000 already in use	Old container still running	docker compose down first
PostgreSQL connection refused	Paper DB not started	Start paper DB compose file
Permission denied on .db file	Volume mount ownership	chmod 666 leap_scanner.db on host

Rollback Procedure

If something breaks, revert to the previous commit:

```
cd ~/Options
git log --oneline -5 # find the commit before the merge
git checkout e01afbb # previous main commit
docker compose down && docker compose build --no-cache && docker compose up -d
```