Spark Setup Documentation

Abraham Sharum, Isaac Tunchez, Andrew Appleyard
Class: CS 30003 - Distributed Systems

December 11, 2025

Demonstration Guide

Installation Steps

1. Install Docker and Docker Compose.
2. Clone the repository.
3. Create a root `.env` with at least:
   - `FINNHUB_API_KEY=`
   - `INFLUXDB_TOKEN=`
   - `INFLUXDB_USERNAME=`
   - `INFLUXDB_PASSWORD=`
   - `INFLUXDB_ORG=`
   - `INFLUXDB_BUCKET=`
4. Ensure ports 4000, 7077, 8001, 8080, 8081, 8082, 8086, 8800 are free.

Run the Containers (Exact Commands)

- Build and start: `docker compose up --build`
- Stop: `docker compose down`
- Rebuild a single service (example backend): `docker compose build backend`
- Spark services: master + two workers come up via compose; workers are limited to `SPARK_WORKER_CORES=2` and `SPARK_WORKER_MEMORY=2g`.

Run the CRUD Application

- Backend FastAPI: `http://localhost:4000` (Swagger UI at `/docs`).
- Frontend dashboard: `http://localhost:8800`.
- Key endpoints:
  - `GET /portfolio` — list all trades.
  - `POST /trades` — create (e.g., `{"symbol":"AAPL","shares":5,"buy_price":180}`).
  - `PUT /trades/{symbol}` — update buy price (e.g., `{"new_price":185}`).
  - `DELETE /trades/{symbol}` — delete/sell a trade; optional body `{"sale_price":190}` to compute net gain.
  - `GET /cache` — view in-memory price cache (refreshed every 5s for tracked symbols).
- Finnhub quote API: `http://localhost:8001/quote/{symbol}` (uses `FINNHUB_API_KEY`).
- Grafana (optional dashboards): `http://localhost:3000`.

Reproduce the Custom Example

1. Start services: `docker compose up --build`.
2. Create a trade:

```
curl -X POST http://localhost:4000/trades \
  -H "Content-Type: application/json" \
  -d '{"symbol":"AAPL","shares":5,"buy_price":180}'
```

3. Update it:

```
curl -X PUT http://localhost:4000/trades/AAPL \
  -H "Content-Type: application/json" \
  -d '{"new_price":185}'
```

4. Delete it:

```
curl -X DELETE http://localhost:4000/trades/AAPL
```

5. Verify via `GET /portfolio` or refresh the frontend dashboard.