



CIS 4951 Capstone II - Fall 2025

Green Vault

Operations Guide

Members:
Giorgio Abboud, B.S, Malik Dagher, B.S
Audrey Gamero, B.S, Daniella Lacotera, B.S
Rodrigo Munoz Salas, B.S



CIS 4951 Capstone II

Giorgio Abboud

Malik Dagher

Audrey Gamero

Daniella Lacotera

Rodrigo Munoz Salas

December 1, 2025

1 Environment File Maintenance

1.1 Files to Maintain

- `/.env.db` → PostgreSQL credentials
- `internal/.env.docker` → Internal API DB URL + JWT signing key
- `ui/.env` → Vite server URLs for UI
- `analyzer/.env` → Twelve Data API key

1.2 Best Practices

- Rotate keys quarterly: update any `JWT_SIGNING_KEY`, database password, or API key every 90 days.
- Never commit real secrets: only store `.env.example` (no values) in git history.
- File permissions: allow read/write only for the owner for all env files. This blocks access from other users/app on the system

2 Deploying and Rotating API Keys Safely

2.1 Twelve Data API Key (`analyzer/.env`)

- When updating the Twelve Data API key:
 - Log in to the official dashboard of Twelve Data.
 - Generate a new key, set expiration if available.
 - Update `analyzer/.env` with the new value.
 - Restart analyzer service via Docker: **`docker compose restart analyzer`**
- If key is exposed or leaked:
 - Immediately revoke it in Twelve Data dashboard.
 - Generate a replacement and update the `.env` file.

2.2 JWT Compromise (`internal/.env.docker`)

- Symptom: 401 spikes, suspicious internal API calls, or reports of stolen token
- When updating JWT Key:
 - Open `internal/.env.docker`
 - Replace `JWT_SECRET=NEW_SECURE_RANDOM`
 - Generate a new secure key with openssl: **`openssl rand -hex 32`**
 - Restart all services: **`docker compose restart`**
- If key is stolen:
 - Treat it as an active incident:
 - Rotate signing key immediately.
 - Invalidate all issued tokens if your API has a logout/denylist strategy.
 - Communicate to the team via your incident channel.

3 Services Images

3.1 Images Versions

All services employ the latest images from Chainguard for each service. This could cause some issues in the future as Chainguard provides more updates. The current setup successfully runs with no dependency issues:

- Golang 1.25.4
- Python 3.14.0
- Node 25.2.1
- Postgres 18.1

3.2 Minimal Images

- All services (Internal, Analyzer, UI, and Postgres DB) run on minimal & secure images.
- Meeting 0 Common Vulnerability Exposures (CVEs), keeping the system free of flaws and weaknesses.

4 Credentials, Backup, and Restore Postgres

4.1 Credentials

- Update all user, password, and db name in `/.env.db` if change of db is required

4.2 Backup

- Run to create a compressed export file you can restore from:
`docker compose exec insert bash -c "pg_dump $POSTGRES_DB -U $POSTGRES_USER -Fc -f /backup/green_vault_latest.dump"`

4.3 Restore

- Run to restore:
`docker compose exec insert bash -c "pg_restore -d $POSTGRES_DB -U $POSTGRES_USER /backup/green_vault_latest.dump"`

4.4 Tips

- Store backups in a mounted volume like:

```
volumes:
  backup:
    driver: local
```

5 Common On-Call & Quick Fixes

Scenario	Action
API key leaked	Revoke and regenerate key in provider dashboard, update <code>.env</code> , then restart analyzer service
Internal JWT stolen	Rotate signing key (<code>env.docker</code>) and restart stack
DB credentials need update	Update <code>env.db</code> and restart stack
Services not responding	Check logs using <code>docker compose logs -f</code> , then restart services with <code>docker compose restart</code>
Backups failing	Validate mounted <code>/backup</code> volume and container permissions

Table 1: Green Vault - Common Scenarios & Fixes