

# Ubuntu Deployment Guide – v1.x

---

*Deploying & Updating **Mr-Moneybags-v1.x** on Ubuntu 22.04 LTS*

---

## Table of Contents

---

1. Prerequisites & Initial Setup
  2. Standard Deployment Workflow (Future Enhancements)
  3. Database Migration Procedure
  4. Service Management with PM2
  5. Health Checks & Validation
  6. Roll-Back Procedures
  7. Troubleshooting Common Issues
  8. Deployment Scenarios (Examples)
  9. Best Practices & Safety Considerations
  10. Maintenance & Monitoring
-

# 1 Prerequisites & Initial Setup

Item	Recommended Version	Install Command
Ubuntu Server	22.04 LTS (64-bit)	N/A
PostgreSQL	16 +	<code>sudo apt install -y postgresql</code>
Node.js	≥ 18 LTS	<code>curl -fsSL https://deb.nodesource.com/setup_18.x   sudo -E bash -</code> <code>sudo apt install -y nodejs</code>
Git	latest in repo	<code>sudo apt install -y git</code>
PM2	5.x (global)	<code>sudo npm i -g pm2</code>
Curl & jq	for health checks	<code>sudo apt install -y curl jq</code>

## 1.1 System User & Directory Layout

```
sudo adduser --system --group fundapp
sudo mkdir -p /opt/mr-moneybags-v1.x
sudo chown -R fundapp:fundapp /opt/mr-moneybags-v1.x
```

## 1.2 Database Bootstrap (one-time)

```
#
# Two idempotent helper scripts now ship with the repository:
#   • **setup-ubuntu-database.sh** - _full fresh install_
#   • **fix-ubuntu-permissions.sh** - _repair / data-load on existing server_
#
# _____
# A)  Fresh Installation  (first time on a new server)
# _____
sudo chmod +x setup-ubuntu-database.sh
sudo ./setup-ubuntu-database.sh
# • creates *npfadmin / npfal23* role
# • (re)creates **fund_accounting_db**
# • runs **db-init.sql** → tables & constraints
# • runs **insert-complete-nacha-data.sql** → full sample data
# • grants ALL + DEFAULT privileges to npfadmin
# • loads The Principle Foundation sample data (entities, funds, vendors, NACHA)
# • writes/updates *.env*
#
# _____
# B)  Fix Existing Installation  (permission errors, missing data)
# _____
sudo chmod +x fix-ubuntu-permissions.sh
sudo ./fix-ubuntu-permissions.sh
# • repairs schema/table/sequence privileges for npfadmin
# • re-runs test-data loader (safe & idempotent)
#
# Both scripts are safe to run multiple times.  If you prefer
# manual SQL, you can still run:
#   sudo -u postgres psql -f setup-database-cross-platform.sql
```

## 2 Standard Deployment Workflow (Future Enhancements)

All commands run as *fundapp* user unless noted

```
sudo -iu fundapp
cd /opt/mr-moneybags-v1.x

# 1. Fetch changes
git fetch --all --tags

# 2. Checkout release tag or main
git checkout v1.0.0          # example
git pull --ff-only

# 3. Install exact dependencies
npm ci

# 4. Apply migrations (see §3)
# (only if schema has changed)
npm run db:schema:refresh # optional helper → re-runs db-init.sql

# 5. Restart services (see §4)
pm2 restart fund-api fund-ui
```

Expected output:

```
> git checkout v1.0.0
HEAD is now at 1ab23cd Release 1.0.0 - initial Mr-Moneybags release
> npm ci
added 321, removed 4, audited 325 packages in 7s
```

## 3 Schema & Sample-Data Refresh

---

The v1.x release uses a **single consolidated schema**.

Re-deploying or refreshing data is therefore straightforward:

```
# 1. Re-apply schema (idempotent)
psql -U npfadmin -d fund_accounting_db -f database/db-init.sql

# 2. Re-load sample NACHA / vendor data (optional, idempotent)
psql -U npfadmin -d fund_accounting_db -f database/insert-complete-nacha-data.sql
```

Both scripts are designed to be **safe to run multiple times**; they use `IF NOT EXISTS` / `ON CONFLICT DO NOTHING` to avoid duplication.

## 4 Service Management with PM2

### 4.1 First-time Startup

```
# API (uses the `start` script from package.json → `node server-modular.js`)  
pm2 start "npm start" --name fund-api  
pm2 start "npx http-server . -p 8080 --no-cache" --name fund-ui  
pm2 save  
pm2 startup systemd -u fundapp --hp /home/fundapp
```

### 4.2 Common PM2 Commands

Action	Command
List processes	<code>pm2 ls</code>
View logs	<code>pm2 logs fund-api</code>
Restart	<code>pm2 restart fund-api fund-ui</code>
Stop	<code>pm2 stop fund-api fund-ui</code>
Remove	<code>pm2 delete fund-api</code>

## 5 Health Checks & Validation

---

### 5.1 API Health Endpoint

```
curl -s http://localhost:3000/api/health | jq
```

Expected:

```
{
  "status": "OK",
  "message": "Server running",
  "connected": true
}
```

### 5.2 Frontend Check

```
curl -I http://localhost:8080 | head -n1
# HTTP/1.1 200 OK
```

### 5.3 Database Check

```
psql -U npfadmin -d fund_accounting_db -c "SELECT COUNT(*) FROM entities;"
```

---

## 6 Rollback Procedures

---

### 6.1 Git Rollback

```
git checkout v1.0.0      # known good tag  
npm ci  
pm2 restart fund-api fund-ui
```

### 6.2 Database Rollback

If migrations were applied with `node-pg-migrate`:

```
node-pg-migrate down -1
```

Or restore nightly dump:

```
pg_restore -U npfadmin -d fund_accounting_db /backups/2025-07-16.dump
```

### 6.3 Full Rollback Script

`deploy-ubuntu.sh --rollback` automatically:

- Restores previous commit
  - Restores `.env`
  - Restores `database.dump`
  - Restarts services
-



## 7 Troubleshooting Common Issues

Symptom	Likely Cause	Fix
<code>DB Offline</code> in UI	PostgreSQL stopped	<code>sudo systemctl start postgresql</code>
PM2 shows <b>errored</b>	Env vars missing	Copy <code>.env.example</code> → <code>.env</code> , then <code>pm2 restart fund-api</code>
<code>could not connect to server: Connection refused</code>	Wrong <code>PGHOST</code> /port	Verify <code>.env</code> and <code>pg_hba.conf</code>
Migrations fail on "duplicate column"	Script not idempotent	Wrap column add in <code>IF NOT EXISTS</code>

Logs to inspect:

```
tail -n 50 ~/.pm2/logs/fund-api-error.log
journalctl -u postgresql
```

## 8 Deployment Scenarios (Examples)

---

### 8.1 Deploy Main Branch Nightly

```
./deploy-ubuntu.sh --branch main
```

### 8.2 Deploy Specific Hotfix

```
./deploy-ubuntu.sh --tag v1.0.1-hotfix
```

### 8.3 Skip Dependency Install (small CSS tweak)

```
./deploy-ubuntu.sh --skip-dependencies
```

### 8.4 Force Deploy Ignoring Failed Health Check (maintenance window)

```
./deploy-ubuntu.sh --force
```

---

## 9 Best Practices & Safety Considerations

---

- **Use Tags** – tag each release ( `git tag v1.0.0` ) for reproducibility.
  - **Nightly Backups** – `pg_dump -Fc` to `/backups`.
  - **Environment Isolation** – never edit `.env` directly on server; use `scp` and reload.
  - **Least Privilege** – application uses `npfadmin`, management uses `postgres`.
  - **Idempotent Scripts** – every SQL migration should be safe to rerun.
  - **CI / CD** – automate tests and linting before merge to `main`.
  - **Monitor Logs** – integrate `pm2-logrotate` to prevent disk bloat.
-

## 10 Maintenance & Monitoring

---

### 10.1 Automated Tasks

Task	Frequency	Tool
Log rotation	daily	<code>pm2 install pm2-logrotate</code>
DB dump	nightly	<code>cron + pg_dump -Fc</code>
OS updates	weekly	<code>unattended-upgrades</code>
Security scan	monthly	<code>lynis audit system</code>

### 10.2 Service Status Dashboard

```
pm2 monit          # live CPU / memory / restart count
```

### 10.3 Upgrade Node & PM2

```
sudo npm i -g n
sudo n 18          # upgrade to latest LTS
sudo npm i -g pm2@latest
pm2 update
```

## Appendix A – Reference Commands

```
# Quick start (first server install)
git clone https://github.com/tpfbill/mr-moneybags.git /opt/mr-moneybags-v1.x
cd /opt/mr-moneybags-v1.x
cp .env.example .env
npm ci
psql -U postgres -f setup-database-cross-platform.sql
pm2 start "npm start" --name fund-api
pm2 start "npx http-server . -p 8080 --no-cache" --name fund-ui
pm2 save
```

You are now ready to **develop, deploy, and evolve Mr-Moneybags-v1.x** on Ubuntu with confidence.

Happy deploying! 🚀

Mr. MoneyBags v1.x Documentation  
© 2025 The Principle Foundation  
Last updated: August 6, 2025