Windows Hyper-V Deployment Guide

Nonprofit Fund Accounting System v8.8 Ubuntu 22.04 LTS | PostgreSQL 16 | Node.js 18

Table of Contents

---

1 Prerequisites

Item	Minimum
Windows 10/11 Pro / Enterprise	Hyper-V enabled
ISO	Ubuntu 22.04 LTS
Host hardware	4 vCPU · 8 GB RAM · 127 GB disk (dynamic)
Internet	Package installs & GitHub
GitHub repo	`https://github.com/tpfbill/nonprofit-fund-accounting`

---

2 Create the Ubuntu 22.04 VM

1. Hyper-V Manager → **Action** → **New** → **Virtual Machine**
2. Name **Nonprofit-Fund-Accounting**
3. **Generation 2**
4. Startup memory **4096 MB** (enable Dynamic Memory)
5. Network Adapter → **Default Switch** (NAT)
6. Virtual Disk → **127 GB** (dynamic VHDX)
7. Installation Media → Ubuntu 22.04 ISO
8. Finish wizard → **Settings** → **Security** → disable **Secure Boot**

---

3 Install Ubuntu with LVM

1. Start VM → **Install Ubuntu**
2. Normal installation (updates optional)
3. *Installation type* → **Guided – use entire disk and set up LVM**
4. Review summary (≈ 62 GB root, rest free in VG) → **Install**
5. Create user **admin** (sudo)

6. *Ubuntu Pro* → **Skip for now**
7. Reboot & login

---

4 Extend the Root Filesystem

```
df -h                            # current size
sudo lvextend -l +100%FREE /dev/ubuntu-vg/ubuntu-lv
sudo resize2fs /dev/ubuntu-vg/ubuntu-lv
df -h                            # root now ≈127 GB
```

---

5 Install & Configure PostgreSQL 16

```
# Add PostgreSQL repo
echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb_release
        -cs)-pgdg main" | \
  sudo tee /etc/apt/sources.list.d/pgdg.list
wget -qO- https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo
        apt-key add -
sudo apt update
sudo apt install -y postgresql-16 postgresql-client-16
        postgresql-16-pgcrypto

# Hard-set postgres password
sudo -u postgres psql -c "ALTER USER postgres WITH PASSWORD
        'npfa123';"

# Create application DB & enable pgcrypto
sudo -u postgres psql -c "CREATE DATABASE fund_accounting_db OWNER
        postgres;"
sudo -u postgres psql -d fund_accounting_db -c "CREATE EXTENSION IF
        NOT EXISTS pgcrypto;"
```

Optional remote access: set `listen_addresses='*'` in `postgresql.conf` and add `host all all 0.0.0.0/0 md5` to `pg_hba.conf`, then:

```
sudo systemctl restart postgresql@16-main
sudo systemctl enable  postgresql@16-main
```

---

6 Install Node.js 18

```
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt install -y nodejs build-essential git
node -v   # v18.x
npm  -v   # 8.x+
```

---

7 Clone & Prepare the Application (v8.8)

```
sudo mkdir -p /opt && cd /opt
sudo git clone https://github.com/tpfbill/nonprofit-fund-accounting.git
cd nonprofit-fund-accounting
sudo git checkout v8.8
sudo chown -R $USER:$USER .
npm install
```

---

8 Initialize the Database

```
sudo -u postgres psql -d fund_accounting_db \
  -f /opt/nonprofit-fund-accounting/src/db/db-init.sql
```

If you see “invalid input syntax for type uuid”, wrap the script:

```
SET session_replication_role = replica;
-- schema here
SET session_replication_role = default;
```

---

9 Run the Application

```
cd /opt/nonprofit-fund-accounting
npm start            # listens on 0.0.0.0:3000
```

Optional PM2 Service

```
sudo npm install -g pm2
pm2 start server.js --name npfa
pm2 startup systemd
```

```
sudo env PATH=$PATH:/usr/bin pm2 startup systemd -u $USER --hp $HOME
pm2 save
```

---

10 Port-Forward Hyper-V NAT → VM

Open **PowerShell as Administrator** on the host:

```
# one-time NAT (if absent)
New-NetNat -Name "HyperVNAT" -InternalIPInterfaceAddressPrefix
        "172.21.0.0/16"

# forward host 3000 → VM 3000  (replace INTERNAL_IP)
Add-NetNatStaticMapping -NatName "HyperVNAT" -Protocol TCP `
  -ExternalIPAddress 0.0.0.0 -ExternalPort 3000 `
  -InternalIPAddress 172.21.209.52 -InternalPort 3000 `
  -Name "NPFA3000"

# firewall rule
New-NetFirewallRule -DisplayName "NPFA Port 3000" `
  -Direction Inbound -Action Allow -Protocol TCP -LocalPort 3000
```

Browse to **http://localhost:3000** on the Windows host.

---

11 (Optional) External Virtual Switch

1. Hyper-V Manager → **Virtual Switch Manager** → *New External*
2. Select physical NIC, name `ExternalNet`
3. VM → Settings → Network Adapter → switch `ExternalNet`
4. VM receives LAN IP (e.g., 192.168.1.x) – NAT not required.

---

12 Backups to Windows Host (dedicated *backup* user)

12.1 Create backup User (Windows)

1. **Win** + **R** → `lusrmgr.msc`
2. *Users* → **Action** → **New User…**
   • Username **backup** • strong password • *Password never expires*
3. *(Optional)* add **backup** to **Administrators** group for full control.

12.2 Create & Share Backup Folder

```
New-Item -Path "C:\NPFA_Backups" -ItemType Directory -Force
icacls "C:\NPFA_Backups" /grant backup:(OI)(CI)F /T
New-SmbShare -Name "NPFA_Backups" -Path "C:\NPFA_Backups" -FullAccess
        "backup"
```

12.3 Mount Share in Ubuntu VM

```
sudo apt install -y cifs-utils
sudo mkdir -p /mnt/windows_backups

sudo tee /root/.smbcredentials <<EOF
username=backup
password=YOUR_BACKUP_PASSWORD
EOF
sudo chmod 600 /root/.smbcredentials

# replace WINDOWS_HOST_IP
echo "//WINDOWS_HOST_IP/NPFA_Backups /mnt/windows_backups cifs
        credentials=/root/.smbcredentials,vers=3.0,iocharset=utf8 0
        0" | sudo tee -a /etc/fstab
sudo mount -a
```

Verify with `ls -la /mnt/windows_backups`

12.4 Daily Backup Script

```
sudo tee /opt/backup-npfa.sh <<'EOF'
#!/bin/bash
TS=$(date +%Y%m%d_%H%M%S)
DIR=/mnt/windows_backups
mountpoint -q $DIR || mount $DIR || exit 1

# database dump
sudo -u postgres pg_dump fund_accounting_db > $DIR/db_${TS}.sql

# application files
tar -czf $DIR/app_${TS}.tar.gz -C /opt nonprofit-fund-accounting

# keep last 14 days
find $DIR -type f -mtime +14 -delete
EOF

sudo chmod +x /opt/backup-npfa.sh
```

```
(crontab -l 2>/dev/null; echo "0 2 * * * /opt/backup-npfa.sh") |
        crontab -
```

Backups now land in **C:_Backups** on the Windows host.

---

13 Troubleshooting

Symptom	Fix
`EADDRINUSE :3000`	Another process uses port 3000 → `sudo lsof -i :3000` then kill.
`psql: could not connect`	`sudo systemctl status postgresql@16-main`; check firewall/service.
404 in browser	Ensure `app.listen(3000,"0.0.0.0")` and NAT rule.
Disk full	`du -h /opt` & `df -h`; clean logs or extend LVM.
UUID errors during schema load	Wrap script with `SET session_replication_role` ….

---

14 Useful Commands Appendix

```
# PostgreSQL service
sudo systemctl {start|stop|restart|status} postgresql@16-main

# LVM
sudo pvs; sudo vgs; sudo lvs

# PM2
pm2 list; pm2 logs npfa; pm2 restart npfa

# Network
ip addr
curl -I http://localhost:3000
sudo netstat -tulpn | grep 3000
```

```
# Manual backup test
sudo /opt/backup-npfa.sh
ls -la /mnt/windows_backups
```