GHL Knowledge Hub Deployment Guide

React Application Deployment on VPS with Nginx Proxy Manager

Overview

This guide shows how to deploy the React-based GHL Knowledge Hub to your VPS using the same infrastructure as your n8n Hub, but with the necessary modifications for a React application.

Key Differences from Single HTML Deployment

- React apps require a **build process** to create static files
- The built files are in the `/dist` folder (not a single HTML file)
- We need to serve the entire `/dist` directory structure
- The main entry point is `/dist/index.html`

Step-by-Step Deployment Process

1. Prepare the Built Files

The React app has already been built and the production files are in the `/dist` folder:

Plain Text

```
ghl-knowledge-hub/dist/
├── index.html
├── assets/
│   ├── index-B76pUR5L.js
│   ├── index-kyXgNrOH.css
│   └── madpanda3d-logo-MyAqBYtE.png
└── favicon.ico
```

2. Create Directory on VPS

Bash

```bash
sudo mkdir -p /var/www/ghlhub
sudo chown -R www-data:www-data /var/www/ghlhub
sudo chmod -R 755 /var/www/ghlhub
```

3. Upload the Built Files

Upload the **entire contents** of the `/dist` folder to your VPS:

Option A: Using SCP (from your local machine)

Bash

```bash
# Extract the zip file first, then upload the dist folder contents
scp -r ghl-knowledge-hub/dist/* root@YOUR_VPS_IP:/var/www/ghlhub/
```

Option B: Upload zip and extract on VPS

Bash

```bash
# Upload the zip file
scp ghl-knowledge-hub-essentials.zip root@YOUR_VPS_IP:/tmp/

# On VPS: Extract and move files
cd /tmp
unzip ghl-knowledge-hub-essentials.zip
sudo cp -r ghl-knowledge-hub/dist/* /var/www/ghlhub/
sudo chown -R www-data:www-data /var/www/ghlhub
```

4. Verify File Structure

Bash

```bash
ls -la /var/www/ghlhub/
# Should show:
# index.html
# assets/ (directory)
# favicon.ico
```

5. Install http-server (if not already installed)

Bash

```bash
sudo npm install -g http-server
```

6. Create Systemd Service

Create `/etc/systemd/system/ghlhub.service`:

Plain Text

```
[Unit]
Description=GHL Knowledge Hub Static Server
After=network.target

[Service]
Type=simple
User=www-data
WorkingDirectory=/var/www/ghlhub
ExecStart=/usr/bin/http-server -p 8082 -a 0.0.0.0 --cors
Restart=always

[Install]
WantedBy=multi-user.target
```

Important Notes:

- Using port **8082** (different from n8n Hub's 8081)
- Working directory points to `/var/www/ghlhub`
- The `-a 0.0.0.0` flag is crucial for external access

7. Start and Enable Service

Bash

```bash
sudo systemctl daemon-reload
sudo systemctl enable ghlhub.service
sudo systemctl start ghlhub.service
sudo systemctl status ghlhub.service
```

8. Test Local Access

Bash

```
curl http://localhost:8082
curl http://YOUR_VPS_IP:8082
```

9. Configure DNS

Create an A record for your subdomain:

- **Subdomain**: `ghlhub.yourdomain.com` (or whatever you prefer)
- **Type**: A
- **Value**: Your VPS IP address

10. Configure Nginx Proxy Manager

In your Nginx Proxy Manager web interface:

Add New Proxy Host:

- **Domain Names**: `ghlhub.yourdomain.com`
- **Scheme**: `http`
- **Forward Hostname/IP**: `127.0.0.1` (or your VPS IP)
- **Forward Port**: `8082`
- **Enable SSL**: Yes
- **Force SSL**: Yes

Troubleshooting

Common Issues and Solutions

1. Service won't start (status=203/EXEC)

- Check if http-server path is correct: `which http-server`
- Update service file with correct path (usually `/usr/bin/http-server`)

2. Port 8082 already in use

- Check what's using the port: `sudo netstat -tlnp | grep 8082`
- Choose a different port and update the service file

3. 404 errors for assets

- Ensure the entire `/dist` folder contents were uploaded
- Check file permissions: `ls -la /var/www/ghlhub/assets/`

4. Site loads but looks broken

- Check if CSS and JS files are accessible
- Verify CORS headers with `--cors` flag in service

5. Can't access externally

- Ensure service uses `-a 0.0.0.0` flag
- Check firewall settings for port 8082
- Verify Nginx Proxy Manager configuration

Monitoring and Maintenance

Check Service Status

Bash

```
sudo systemctl status ghlhub.service
sudo journalctl -u ghlhub.service -f
```

Update the Application

To update with new versions:

1. Stop the service: `sudo systemctl stop ghlhub.service`
2. Replace files in `/var/www/ghlhub/`
3. Start the service: `sudo systemctl start ghlhub.service`

Resource Monitoring

Bash

```bash
# Check resource usage
htop
# Check network connections
sudo netstat -tlnp | grep 8082
```

Security Considerations

- The service runs as `www-data` user (non-root)
- Files have appropriate permissions (755 for directories, 644 for files)
- HTTPS is enforced through Nginx Proxy Manager
- CORS is enabled for API compatibility

This deployment method provides the same reliability and security as your n8n Hub while properly serving the React application's static assets.