

ESP32-Tunnel Relay Server Hosting Guide (VPS)

Overview

This guide explains how to host the ESP32-Tunnel relay server on a Linux VPS, connect ESP32 devices, configure systemd, and expose the service securely.

1. VPS Preparation

Install Ubuntu on a VPS. Connect using SSH and update the system: `apt update && apt upgrade -y`
Install packages: `git, python3, python3-pip, python3-venv, nginx`.

2. Download the Repository

```
cd /opt && git clone https://github.com/HamzaYslmn/esp32-tunnel.git
```

3. Create Python Environment

```
cd /opt/esp32-tunnel/python && python3 -m venv ../venv && source ../venv/bin/activate && pip install fastapi uvicorn websockets psutil
```

4. Run the Relay Server

```
uvicorn main:app --host 0.0.0.0 --port 8000
```

Verify operation using: `curl http://127.0.0.1:8000/api/status`

5. Query String Fix

Modify `api/tunnel.py` so query strings are appended to the forwarded path. This allows URLs such as: `/api/gpio?pin=2&state=1` to reach the ESP32 correctly.

6. Create a Systemd Service

```
[Unit]
Description=ESP32 Tunnel Relay Server
After=network.target
[Service]
Type=simple
User=root
WorkingDirectory=/opt/esp32-tunnel/python
Environment=PORT=8000
ExecStart=/opt/esp32-tunnel/venv/bin/uvicorn main:app --host 0.0.0.0 --port 8000
Restart=always
RestartSec=5
[Install]
WantedBy=multi-user.target
```

Enable: `systemctl daemon-reload`
enable `esp32-tunnel`
`systemctl start esp32-tunnel`

7. Firewall

Open port 8000: `ufw allow 8000/tcp`

8. ESP32 Configuration

Example: `const char *TUNNEL_SERVER = "https://your-domain.com/nexintel-esp";`

9. GPIO Control API

Example URLs: <https://your-domain.com/nexintel-esp/api/gpio?key=PASSWORD&pin;=2&state;=1>
<https://your-domain.com/nexintel-esp/api/gpio?key=PASSWORD&pin;=2&state;=0>

10. Recommended Production Setup

Use Nginx as a reverse proxy, obtain a Let's Encrypt certificate, and expose the service through HTTPS/WSS for secure operation.