

ESP32-Tunnel Professional VPS Deployment Manual

Comprehensive Guide for Self-Hosted Remote ESP32 Access

1. Introduction

This document describes the complete process of deploying, securing, and operating the ESP32-Tunnel relay server on a Linux VPS. The guide is intended for engineers, system integrators, and IoT solution providers.

2. System Architecture

The relay server acts as a bridge between internet clients and ESP32 devices. ESP32 establishes an outbound WebSocket connection to the relay. Clients access the relay through HTTPS, and requests are forwarded to the ESP32.

3. Hardware and Software Requirements

VPS: 1 vCPU, 1GB RAM minimum. Ubuntu 22.04/24.04 recommended. ESP32 with Wi-Fi connectivity. Python 3.10+.

4. Network Flow

Browser/PHP Application → VPS Relay → WebSocket Tunnel → ESP32 → Local API Endpoint.

5. Creating the VPS

Provision a VPS from Hostinger, DigitalOcean, Vultr, Linode, or another provider. Ensure SSH access is enabled.

6. Initial Server Hardening

Update packages, change default passwords, configure SSH keys, disable password logins if possible, and enable a firewall.

7. Installing Required Packages

```
apt update  
apt upgrade -y  
apt install git python3 python3-pip python3-venv nginx -y
```

8. Downloading ESP32-Tunnel

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

9. Creating Python Environment

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

10. Running the Relay Manually

```
uvicorn main:app --host 0.0.0.0 --port 8000
curl http://127.0.0.1:8000/api/status
```

11. Understanding Tunnel Operation

Each ESP32 connects to `/api/tunnel/ws?id=device-id`. The relay maintains pending request queues and forwards HTTP requests through the WebSocket channel.

12. Query String Forwarding Fix

Modify `api/tunnel.py` so the original query string is appended to the forwarded path. This is required for GPIO control and parameterized APIs.

13. Creating a Systemd Service

```
[Unit]
Description=ESP32 Tunnel Relay Server
After=network.target

[Service]
Type=simple
User=root
WorkingDirectory=/opt/esp32-tunnel/python
ExecStart=/opt/esp32-tunnel/venv/bin/uvicorn main:app --host 0.0.0.0 --port 8000
Restart=always

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

14. Service Management

```
systemctl daemon-reload
systemctl enable esp32-tunnel
systemctl start esp32-tunnel
systemctl status esp32-tunnel
```

15. Firewall Configuration

```
ufw allow 22/tcp
ufw allow 80/tcp
ufw allow 443/tcp
ufw allow 8000/tcp
```

16. Nginx Reverse Proxy

Place Nginx in front of Uvicorn to provide better performance, TLS termination, and easier certificate management.

17. SSL Certificates

Use Let's Encrypt and Certbot to generate and automatically renew certificates.

18. Domain Configuration

Create an A record pointing `tunnel.yourdomain.com` to the VPS public IP address.

19. ESP32 Firmware Configuration

```
const char *TUNNEL_SERVER =  
"https://tunnel.yourdomain.com/nexintel-esp";
```

20. Creating REST APIs on ESP32

Create routes such as `/api/data`, `/api/gpio`, `/api/gpio/read`, and custom sensor endpoints using `ESPAsyncWebServer`.

21. GPIO Control Examples

```
https://tunnel.yourdomain.com/nexintel-esp/api/gpio?key=PASSWORD&pin=2&state=1  
https://tunnel.yourdomain.com/nexintel-esp/api/gpio?key=PASSWORD&pin=2&state=0
```

22. PHP Integration

```
$url='https://tunnel.yourdomain.com/nexintel-esp/api/gpio?pin=2&state=1';  
echo file_get_contents($url);
```

23. Monitoring and Logging

Use `journalctl`, Nginx logs, and application logs to monitor device connections and troubleshoot issues.

24. Troubleshooting Guide

Connection refused: verify firewall. Tunnel not found: check ESP32 connection. Missing parameters: verify query-string forwarding.

25. Security Recommendations

Use API keys, TLS, strong passwords, rate limiting, and VPNs where practical.

26. Scaling Considerations

One VPS can support multiple ESP32 devices. Use unique tunnel IDs and monitor memory consumption.

27. Backup and Recovery

Back up configuration files, systemd service files, SSL certificates, and modified source code.

28. Example Deployment Workflow

Deploy VPS → Install software → Configure relay → Create service → Configure SSL → Connect ESP32 → Test APIs.

29. Maintenance Checklist

Update OS monthly, renew certificates, monitor logs, verify backups, and test device connectivity.

30. Conclusion

A self-hosted ESP32-Tunnel deployment provides secure remote access to ESP32 devices without requiring port forwarding or public IP addresses.