

Cloudflare Tunnel Permanent Setup (EC2 + Spring Boot)

This document contains the complete command list to create a Cloudflare Tunnel, run it permanently in the background, and safely close the terminal.

1. Connect to Your EC2 Instance

```
ssh ubuntu@YOUR_EC2_PUBLIC_IP
```

2. Install Cloudflared

Download package:

```
wget https://github.com.cloudflare/cloudflared/releases/latest/download/  
cloudflared-linux-amd64.deb
```

Install:

```
sudo dpkg -i cloudflared-linux-amd64.deb
```

Verify installation:

```
cloudflared --version
```

3. Confirm Spring Boot Backend is Running

Check port 8080:

```
sudo lsof -i :8080
```

Test locally:

```
curl http://localhost:8080
```

✓ 4. Stop Any Old Tunnel Process (Clean Start)

```
pkill cloudflared
```

✓ 5. Create a Permanent Quick Tunnel Service (No Login Required)

Create systemd service file:

```
sudo nano /etc/systemd/system/quicktunnel.service
```

Paste this configuration:

```
[Unit]
Description=Cloudflare Quick Tunnel
After=network.target

[Service]
ExecStart=/usr/bin/cloudflared tunnel --url http://localhost:8080
Restart=always
RestartSec=5
User=root

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

Save and exit:

- Press **CTRL + X**
 - Press **Y**
 - Press **ENTER**
-

6. Reload systemd

```
sudo systemctl daemon-reload
```

7. Enable Tunnel Auto-Start After Reboot

```
sudo systemctl enable quicktunnel
```

8. Start Tunnel Now (Runs in Background)

```
sudo systemctl start quicktunnel
```

9. Check Tunnel Status

```
sudo systemctl status quicktunnel
```

Expected:

```
Active: running
```

10. Get Your Public HTTPS Tunnel URL

Run:

```
journalctl -u quicktunnel | grep trycloudflare
```

Example output:

```
https://silver-bird-1234.trycloudflare.com
```

That is your backend HTTPS URL 🎉

11. Test Tunnel URL

```
curl https://silver-bird-1234.trycloudflare.com
```

12. Use Tunnel URL in Vercel

Add Environment Variable:

```
VITE_BASE_URL=https://silver-bird-1234.trycloudflare.com
```

React fetch example:

```
fetch(`${import.meta.env.VITE_BASE_URL}/api/auth/login`)
```

13. Close Terminal Safely

Tunnel runs permanently as a background service.

You can exit SSH:

```
exit
```

Or close terminal window — tunnel will continue running.

Useful Commands Later

View live tunnel logs:

```
journalctl -u quicktunnel -f
```

Stop tunnel:

```
sudo systemctl stop quicktunnel
```

Restart tunnel:

```
sudo systemctl restart quicktunnel
```

Disable auto-start:

```
sudo systemctl disable quicktunnel
```

!! Final Result

Spring Boot running on EC2 (`localhost:8080`)

Cloudflare HTTPS tunnel URL working

Tunnel stays alive after logout

React frontend on Vercel can securely call backend

If you want, next step is creating a **named tunnel with permanent fixed URL** (requires Cloudflare login).