

Minecraft Server über Cloudflare Tunnel öffentlich machen

■ Vollständige Schritt-für-Schritt-Anleitung (Template mit Platzhaltern)

Ziel: Einen bestehenden Minecraft-Server (lokal oder via Playit) über einen sicheren Cloudflare Tunnel öffentlich erreichbar machen, z. B. über mc1.<deinedomain>.com.

Voraussetzungen:

- 1 Eine Domain, die über Cloudflare verwaltet wird (z. B. <deinedomain>.com)
- 2 Einen laufenden Minecraft-Server (lokal oder in Docker)
- 3 Cloudflare Zero Trust Account (kostenlos)
- 4 Docker installiert

Schritt 1 – Cloudflare Tunnel anlegen:

1. Melde dich unter <https://dash.cloudflare.com> an.
2. Wähle deine Domain (<deinedomain>.com).
3. Öffne **Zero Trust** → **Tunnels**.
4. Falls noch keine Organisation existiert, erstelle eine (z. B. <orgname>).
5. Klicke auf „+ Create a Tunnel“ und gib einen Namen ein, z. B. MC1-Tunnel.
6. Kopiere den generierten **Token** (Format: eyJhIjoiYmVuLW1ha2VzLXR1bm5lbC0xMjMiLCJpIjoiODc2NzgifQ==).

Schritt 2 – Cloudflared Container starten:

Führe folgenden Befehl in PowerShell oder deinem Terminal aus (eine Zeile, ohne Umbrüche):

```
docker run -d --name <container_name> --network <docker_network>  
cloudflare/cloudflared:latest tunnel run --token <dein_token>
```

Beispiel:

```
docker run -d --name cloudflared_mc1 --network minecraft-net  
cloudflare/cloudflared:latest tunnel run --token  
eyJhIjoiYmVuLW1ha2VzLXR1bm5lbC0xMjMiLCJpIjoiODc2NzgifQ==
```

Schritt 3 – Route in Cloudflare einrichten:

1. Öffne deinen Tunnel im Cloudflare Dashboard.
2. Wähle **Public Hostnames** → **Add a public hostname**.
3. Trage Folgendes ein:

Feld	Wert
Subdomain	mc1
Domain	<deinedomain>.com
Type	TCP
URL / Service	<zieladresse>:<port> (z. B. pass-beach.gl.joinmc.link:25565)

■ NoTLSVerify ist bei **TCP** automatisch aktiv, also kein zusätzlicher Schalter nötig.

Schritt 4 – DNS-Konflikt beheben (falls vorhanden):

Falls beim Speichern die Fehlermeldung Error: An A, AAAA, or CNAME record with that host already exists erscheint:

1. Gehe zu **DNS** → **Records** in Cloudflare.
2. Lösche den bestehenden Eintrag (z. B. mc1 CNAME).
3. Versuche erneut, den Public Hostname hinzuzufügen.

Schritt 5 – Verbindung testen:

Prüfe in der Konsole, ob dein Tunnel aktiv ist:

```
docker logs <container_name>
```

Wenn die Ausgabe Connection established. Tunnel ID: ... enthält → Tunnel läuft!

Jetzt kannst du in Minecraft einfach mit mc1.<deinedomain>.com verbinden.

Optional – Persistente Konfiguration (empfohlen):

Um den Tunnel beim Neustart automatisch zu verbinden, kann eine config.yml verwendet werden. Lege sie z. B. unter /docker/cloudflared/config/config.yml an und mounte sie so:

```
docker run -d --name cloudflared_mcl --network minecraft-net -v  
C:\Docker\cloudflared\config:/etc/cloudflared cloudflare/cloudflared:latest  
tunnel run
```

Die Datei config.yml sollte enthalten:

```
tunnel: <tunnel-id> credentials-file: /etc/cloudflared/<tunnel-id>.json  
ingress: - hostname: mc1.<deinedomain>.com service:  
tcp://<zieladresse>:<port> - service: http_status:404
```

■ Template erstellt mit Platzhaltern für öffentliche Dokumentation (z. B. GitHub-Readme oder Tutorial-Post).

Autor: <Dein Name oder Projekt> | Lizenz: MIT / frei verwendbar