dvb-WarpPool

Dein eigener Bitcoin-Solo-Mining-Pool

Öffentlicher 0%-Solo-Pool — Betreiberhandbuch

dvb-WarpPool als offenen, non-custodial, gebührenfreien Pool betreiben

Version 1.52.0 · Ausgabe November 2026

Inhalt

1 Was ein öffentlicher 0%-Pool ist

2 Den per-worker-Modus aktivieren

3 Was Miner eintragen

4 Öffentliche Wallet-Seiten & der Eigentümer-Bereich

5 Betreiber-Privatsphäre

6 Spenden annehmen (optional)

7 Härtungsprofil für den öffentlichen Betrieb

8 TLS

9 Hinter einem Reverse Proxy

10 Deployment-Checkliste

11 Support & Links

1 · Was ein öffentlicher 0%-Pool ist

Ein öffentlicher 0%-Pool macht aus dvb-WarpPool einen **offenen, non-custodial, gebührenfreien Solo-Pool**: jeder Miner, der sich verbindet, mint auf *seine eigene* Bitcoin-Adresse — aus dem Stratum-Login — und die volle Blockbelohnung geht direkt an denjenigen, der den Block findet, on-chain. Der Pool hält nie Gelder. Es gibt nichts abzuheben und nichts, worauf man vertrauen müsste.

Technisch ist das der **per-worker-Auszahlungsmodus**. Im Standard-Modus `single` zahlt die gesamte Belohnung an eine Betreiberadresse — das gewöhnliche „Solo + Freunde"-Setup, bei dem jeder Miner für *deine* Wallet arbeitet. Im **per-worker**-Modus liefert jeder Miner seine eigene Adresse als Stratum-Benutzernamen, und die coinbase wird pro Verbindung so gebaut, dass sie genau diese Adresse bezahlt: **ein Output, die volle Subsidy, keine Gebühr.**

Die vier Eigenschaften, die ihn definieren

- **Non-custodial.** Der Pool berührt nie die Coins von irgendjemandem. Das Bitcoin-Netzwerk zahlt den Finder direkt.
- **Per-worker.** Verschiedene Miner können verschiedene Adressen nutzen; jeder behält die volle Belohnung für die Blöcke, die er findet.
- **Volle Belohnung an den Finder.** Ein einziger coinbase-Output zahlt die gesamte Subsidy plus Gebühren an die Adresse im Login des siegreichen Miners — 0% Abschöpfung.
- **Fail-closed.** Der Daemon verweigert den Start, wenn eine Gebühr oder eine Betreiberspende zusammen mit dem per-worker-Modus konfiguriert ist, sodass du nicht versehentlich einen „öffentlichen" Pool betreiben kannst, der still von der Belohnung abschöpft.

Ehrlichkeits-Leitplanke — bitte lesen. dvb-WarpPool hat **noch keinen Mainnet-Block gefunden.** Der Konsens-Pfad (per-worker-coinbase → ein echter Bitcoin-Core-Node akzeptiert den Block → die Belohnung zahlt den Finder) ist auf regtest und signet Ende-zu-Ende verifiziert, aber beim ersten Mainnet-Block heißt es weiterhin **„noch nicht"**. Bewirb den Pool für genau das, was er ist — ein frischer, offener, non-custodial Solo-Pool — und **niemals** als „battle-tested". Wenn du eine Landing-Page veröffentlichst, halte diese Leitplanke sichtbar.

Abgrenzung zum `single`-Modus. Wenn du nur deine eigenen Rigs oder eine feste Gruppe von Freunden minen willst, die alle an eine Wallet zahlen, bleib im `single`-Modus — siehe das *Benutzerhandbuch*. Dieses Betreiberhandbuch geht speziell darum, den Pool so zu öffnen, dass Fremde ihre Miner darauf richten können und jeder behält, was er findet.

2 · Den per-worker-Modus aktivieren

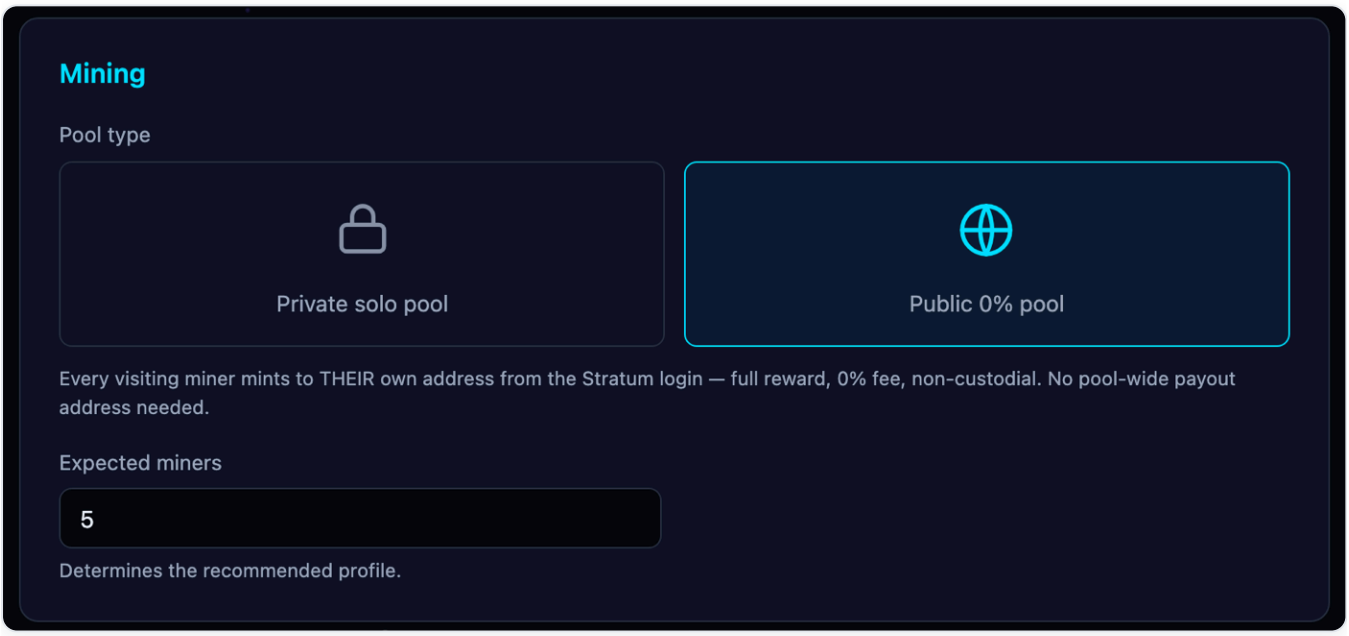
Es gibt zwei Wege zum selben Ergebnis: den setup wizard (empfohlen für eine neue Installation) oder eine Ein-Zeilen-Änderung in `config.toml`. Beide schreiben dasselbe Flag `payout_mode = "per_worker"`.

Weg A — der setup wizard

Wenn du über den setup wizard installierst, bietet seine **Mining**-Karte eine Wahl des **Pool-Typs**: **Privater Solo-Pool** versus **Öffentlicher 0%-Pool**. Die Wahl von **Öffentlicher 0%-Pool** erledigt drei Dinge für dich:

- schreibt `payout_mode = "per_worker"`;
- blendet das poolweite Feld **Payout-Adresse** aus (es wird in diesem Modus nicht genutzt — jeder Miner bringt seine eigene mit);
- setzt die **Pool-Fee** auf 0 und blendet die Gebührenfelder aus.

Kein Editieren von Hand erforderlich.



Die Mining-Karte des setup wizards: die Wahl von „Öffentlicher 0%-Pool“ stellt den Pool auf per-worker-Auszahlungen um und blendet die Felder für Auszahlungsadresse und Gebühr aus.

Weg B — `config.toml`

Wenn du die Config von Hand verwaltest, genügt ein einziges Flag als Funktionsschalter:

```
[mining]
payout_mode = "per_worker"
# payout_address is IGNORED in per_worker mode (each miner brings its own).
# pool_fee_percent / operator_donation_percent MUST be 0 — the daemon
# refuses to start otherwise (full reward goes to the finder).
```

Fail-closed bei Gebühr oder Spende

Der Daemon **verweigert beim Start den Dienst** (fail-closed), wenn ein von null verschiedenes `pool_fee_percent` oder `operator_donation_percent` zusammen mit `per_worker` konfiguriert ist. Er bootet nicht, bis Gebühr und Spende wieder bei null sind. Das ist eine bewusste Sicherheits-Invariante: ein

per-worker-Pool zahlt dem Finder die *volle* Belohnung, es gibt also keine Konfiguration, in der er still einen Anteil nehmen kann.

> **Tippfehler zahlen nie an den Falschen.** Eine ungültige oder netzfremde Login-Adresse wird bei `mining.authorize` (Stratum V1) oder beim Channel-Open (Stratum V2) **abgelehnt**. Es gibt **keinen Rückfall** auf eine Betreiberadresse — eine vertippte Adresse wird abgewiesen, niemals still umgeleitet.

3 · Was Miner eintragen

Ein Miner, der deinem öffentlichen Pool beitritt, konfiguriert sein Stratum-Login genau wie bei jedem Solo-Pool, mit einer Regel: **der Benutzername ist die eigene Bitcoin-Adresse des Miners**, optional gefolgt von einem Rig-Label nach einem Punkt.

```
Username:  bc1qyour-own-address.rig1
Password:  x   (beliebig, wird nicht genutzt)
```

- Alles **vor dem ersten Punkt** ist die Auszahlungsadresse; alles danach ist ein freies Rig-Label, das nur für Statistiken verwendet wird.
- Das Passwort kann beliebig sein (üblich `x`).

Unterstützte Adresstypen

Jede gültige Mainnet-Adresse funktioniert. Der Pool akzeptiert:

Typ	Beispiel-Präfix
P2PKH (Legacy)	`1…`
P2SH (inkl. Wrapped SegWit)	`3…`
P2WPKH (natives SegWit)	`bc1q…`
P2WSH (natives SegWit-Skript)	`bc1q…`
P2TR (Taproot)	`bc1p…`

> **Falsche Adresse = abgewiesen, nicht umgeleitet.** Ist das Login keine gültige Adresse für dieses Netzwerk, wird der Miner zum Zeitpunkt der Autorisierung abgelehnt. Es gibt keine Rückfalladresse, sodass ein Tippfehler schlicht nicht verbindet, statt die Blockbelohnung an den falschen Empfänger zu zahlen.

4 · Öffentliche Wallet-Seiten & der Eigentümer-Bereich

Jede Adresse, die auf deinem Pool mint, bekommt ihre eigene Login-freie Status-Seite — der „meine Adresse nachschlagen“-Ablauf, den ein öffentlicher Solo-Pool haben sollte. Miner setzen ihre eigene als Lesezeichen; niemand braucht ein Konto.

Die öffentliche Seite unter `/users/<address>`

Öffne `/users/<address>` und du siehst für diese Adresse:

- **Gesamt-Hashrate** in großer Schrift, summiert über jedes Rig, das unter der Adresse angemeldet ist, mit einem Fenster-Umschalter 1 m / 5 m / 1 h / 1 d / 7 d;
- ein **24-Stunden-Hashrate-Diagramm** mit den Best Shares als Punkten;
- die **höchste jemals erreichte Share-Difficulty**, **Rigs online** und die **insgesamt akzeptierten Shares**;
- eine **Tabelle pro Rig** — eine Zeile pro `address.rig`-Login, mit Online-Punkt, Hashrate, Best-Share, letztem Share und Firmware, dazu eine WarpTune-Spalte, wenn der Besitzer dieses Badge teilt;
- ein ehrliches **Block-Chancen-Panel** (siehe unten).

Kein Login, keine Betreiber-Beteiligung — jeder, der die Adresse hat, kann sie öffnen.

Das **Block-Chancen-Panel** ersetzt die alte einzelne „geschätzte Zeit bis zum Block“ — ein Mittelwert, der sich bei einem Heim-Rig als „nie“ liest und den Sinn des Solo-Minings verbirgt: Der Vorgang ist **gedächtnislos**, jede Sekunde eine neue Ziehung, bei der nichts näher rückt. Stattdessen zeigt es die echte Verteilung: die **Chance in den nächsten 24 Stunden** und **innerhalb eines Jahres** (lange Quoten als „1 zu N“, Prozentwerte abgerundet, damit sie nie überzeichnen), den **50/50-Punkt** (Median, nicht Durchschnitt) und das **Glück bisher** — erwartete vs. gefundene Blöcke mit einem ehrlichen „N % der Miner mit deiner Historie hätten ebenfalls noch keinen“. Eine Gedächtnislosigkeits-Fußnote macht den Kernpunkt explizit: Noch so langes Warten macht einen Block nicht wahrscheinlicher.



Eine öffentliche Seite pro Wallet unter /users/<address>: aggregierte hashrate, Aufschlüsselung pro Rig und Best Share — kein Login erforderlich.

ckpool-kompatibles JSON. Dieselben Zahlen werden unter `GET /api/users/<address>` im ckpool-Feldformat ausgeliefert (`hashrate1m` , `hashrate1hr` , `bestshare` , `worker[]` , …), sodass Miner-Firmware und Statistik-Werkzeuge wie `ckstats` sie direkt lesen können. Beide Routen sind einfaches HTTP unter dem Status-Port, hinter demselben Reverse Proxy wie das dashboard.

Der Eigentümer-Bereich (opt-in)

Standardmäßig ist eine Wallet-Seite öffentlich. Ein Miner, der mehr Kontrolle will, kann den **Besitz** der Adresse **beweisen** und den **Eigentümer-Bereich** freischalten — vollständig pro Miner, nichts, was der Betreiber konfiguriert:

1. Die Seite gibt eine kurze einmalige **Challenge-Nachricht** aus.
2. Der Miner signiert sie im *Sign Message*-Feld seiner eigenen Wallet — die Standard-BIP-137-„Adresse verifizieren"-Funktion in Electrum, Sparrow, Bitcoin Core und Hardware-Wallets. **Kein privater Schlüssel verlässt die Wallet, keine Gelder bewegen sich, kein Konto wird erstellt** — der Pool prüft nur die Signatur.
3. Eine gültige Signatur setzt ein kurzlebiges Owner-Cookie für diese Adresse und offenbart drei Steuerelemente:
   - **Privatsphäre** — die Adresse aus der öffentlichen Wallet-Liste ausblenden und ihre `/users/<address>`-Daten für alle außer dem verifizierten Besitzer leeren (der direkte Link funktioniert für ihn weiter).
   - **Benachrichtigungen** — ein persönliches ntfy-Topic oder einen Webhook auf die Meilensteine dieser Adresse richten: *Block gefunden*, ein neuer *persönlicher Best-Share* (über alle ihre Rigs), erstmaliger Einzug in die *All-Time-Top-100* des Pools und *Rig offline*. Jeder hat seinen eigenen Cooldown gegen Spam, und ein gefundener Block sendet eine Schlagzeile statt drei Meldungen.
   - **WarpTune-Badge** (seit v1.31.0) — einen **Push-Token** für diese Adresse erzeugen, ihn auf dem Rechner, auf dem dein dvb-WarpTune-Sidecar läuft, als die in `warptune.toml` genannte Umgebungsvariable (`pool_push_token_env`) setzen und `pool_push = true` eintragen. Das Sidecar schickt dann seinen Snapshot an diesen Pool, und die Rigs-Tabelle zeigt je Rig ein Badge: Modus, Frequenz, Simulation / live / Steuerung aus / veraltet; fahre mit der Maus darüber, um die letzte Entscheidung zu sehen. Besucher sehen das reduzierte Badge erst nach **Badge für Besucher anzeigen**. Der Token wird nur einmal angezeigt, lässt sich hier erneuern oder widerrufen, und der Pool schickt nie etwas an einen Miner zurück.

Coinbase-Text (v1.45, per-worker-Pools): Hast du deine Adresse bewiesen, kannst du bis zu 16 Zeichen (Buchstaben, Ziffern, Satzzeichen, Leerzeichen; kein `/`) festlegen, die in die Coinbase jedes Blocks kommen, den dieser Pool für deine Adresse baut — direkt hinter dem Pool-Tag, deine Spur auf der Chain, in jedem Block-Explorer sichtbar. Gilt ab dem nächsten Job; der Betreiber kann die Funktion abschalten.

Taproot funktioniert jetzt auch. Das Besitz-Signieren deckt Single-Key-Adressen **P2PKH / P2WPKH / P2SH-P2WPKH** (BIP-137) ab und **Taproot (P2TR)** per **BIP-322** — signiere mit einer Wallet, die das unterstützt, z. B. Sparrow. Der Eigentümer-Bereich braucht **keine Betreiber-Config** — er wird vollständig vom Miner gesteuert.

5 · Betreiber-Privatsphäre

Ein öffentlicher Mainnet-Pool sollte keine aufzählbare Liste aller Adressen und Rig-Setups herausgeben. dvb-WarpPool gibt dir **zwei unabhängige Schichten** der Kontrolle. Beide sind auf das offene Verhalten voreingestellt, sodass Härtung hier opt-in ist.

Schicht A — Config: Sichtbarkeit von Verzeichnis & Leaderboard

Zwei Schlüssel in `[privacy]` legen die Standard-Sichtbarkeit der gesamten UI fest. In beiden Fällen sieht der **angemeldete Betreiber weiterhin die vollständigen Daten** — diese betreffen nur anonyme Besucher, also lass die Admin-Authentifizierung aktiviert.

```
[privacy]
# Hide the browsable Workers/wallet directory from anonymous visitors. Each
# /users/<address> page still works by direct link; the logged-in operator
# still sees the full directory (so keep admin auth enabled).
wallet_directory_public = false
# Mask the Bitcoin address in the public best-shares leaderboard, server-side.
# The logged-in operator still sees full addresses.
leaderboard_addresses_public = false
```

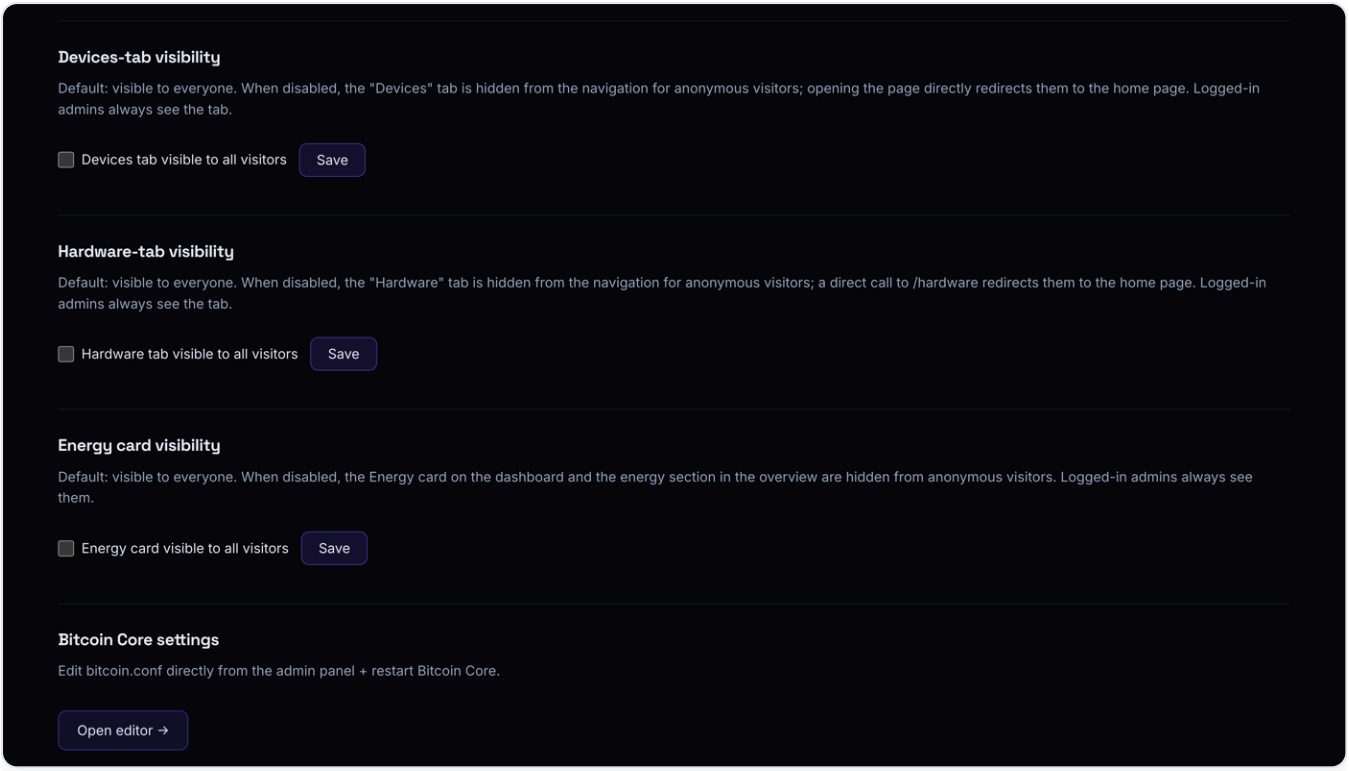
Mit dieser Haltung erreichen Miner ihre eigene Seite per Link, aber der Pool veröffentlicht keine durchsuchbare Liste aller, und Leaderboard-Adressen werden serverseitig maskiert. Es berührt weder Auszahlungen noch den Blockbau.

Schicht B — Sichtbarkeits-Schalter der Admin-UI

Unter **Admin** → **Profil** blendet eine Reihe von Sichtbarkeits-Schaltern ganze Ansichten vor anonymen Besuchern aus:

- den **Geräte**-Tab (die Route `/miners` — der interaktive Geräte-Manager mit Telemetrie pro Gerät);
- den **Hardware**-Tab (Host-Ressourcen und Bild des Größenprofils);
- die **Energie**-Karte auf dem dashboard.

> **Welchen Tab blendet der Schalter aus?** Der Sichtbarkeits-Schalter steuert den Tab **Geräte** — die Route `/miners` mit Live-Geräte-Telemetrie — **nicht** die Miner-/Worker-Auflistung. Die Namen sind leicht zu verwechseln (Worker = Wallets, Miner = Stratum-Logins, Geräte = die Hardware); der Schalter betrifft die Geräte-Manager-Ansicht.



Admin → Profil: Sichtbarkeits-Schalter blenden die Ansichten Geräte/Hardware/Energie vor anonymen Besuchern aus; Adressen im öffentlichen Leaderboard werden serverseitig maskiert.

> **Empfohlene Haltung für öffentliches Mainnet.** Setze `wallet_directory_public = false` und `leaderboard_addresses_public = false` und blende die Ansichten Geräte, Hardware und Energie vor anonymen Besuchern aus. Miner erreichen ihre eigene `/users/<address>`-Seite weiterhin per Link; der Betreiber behält das vollständige Bild, wenn er angemeldet ist.

Einen Worker entfernen

Bittet dich ein Miner, von der Bestenliste zu verschwinden, oder stehen dort alte Testnamen, öffne **Admin → Worker**. **Entfernen** vergisst einen Worker, der gerade nicht verbunden ist: seinen Eintrag, seine Bestenlisten-Shares (damit ist er von der Bestenliste weg) und sein VarDiff-Gedächtnis. Share-Historie und gefundene Blöcke bleiben. Einen verbundenen Worker kannst du nicht entfernen: Der Miner muss sich zuerst trennen, und meldet er sich später wieder an, fängt er einfach von vorn an. Ein Miner, der gerade aufgehört hat, steht als **kürzlich aktiv** da und lässt sich schon entfernen. Namen, deren Worker-Eintrag die Aufbewahrung schon gelöscht hat, die aber noch auf der Bestenliste stehen, zeigt eine zweite Tabelle **Reste ohne Worker-Eintrag**; sie lassen sich genauso entfernen. Jeder Versuch landet im Audit-Log.

6 · Spenden annehmen (optional)

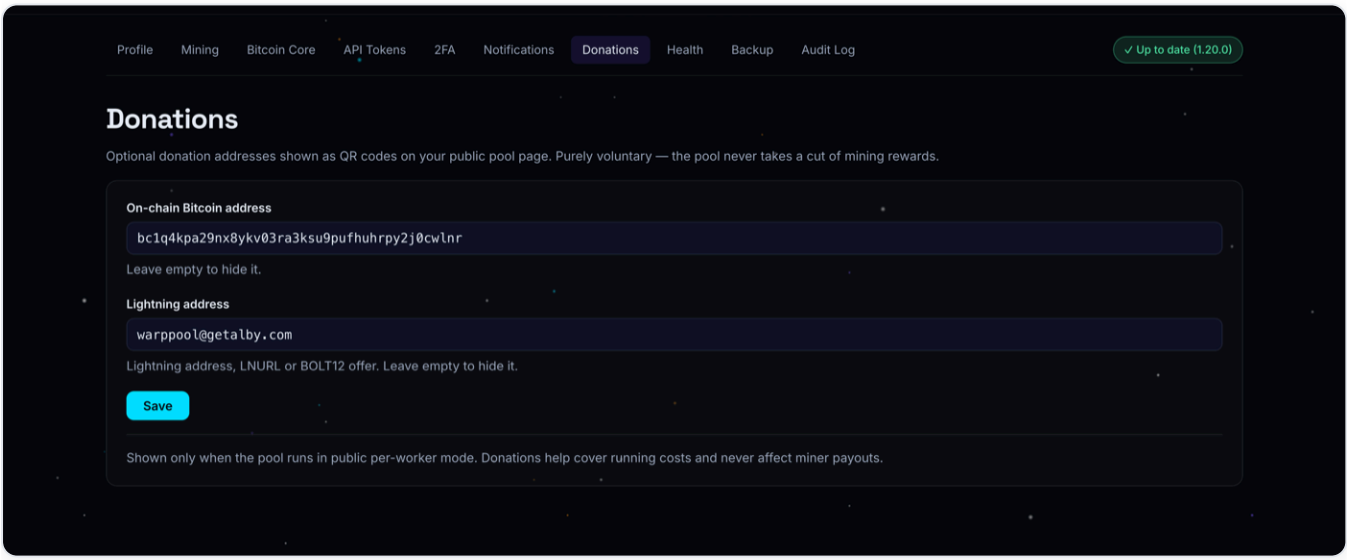
Weil ein per-worker-Pool **0%** von jeder Belohnung nimmt, ist der einzige Weg, das Hosting zu finanzieren, eine freiwillige Spende. Das dashboard hat dafür eingebaute Unterstützung, und die ganze Funktion erscheint **nur im per-worker-Modus** — ein privater Solo-Pool zeigt sie nie.

Die Adressen setzen

Setze sie unter **Admin → Spenden**. Der Navigationseintrag ist **verborgen, außer der Pool läuft im per-worker-Modus**.

Feld	Was einzutragen ist
On-Chain-Bitcoin-Adresse	Eine beliebige gültige Bitcoin-Adresse. Sie wird beim Speichern validiert und als `bitcoin:`-QR-Code gezeigt, den die Wallet des Besuchers direkt öffnen kann.
Lightning-Adresse	Eine Lightning-Adresse, ein LNURL- oder BOLT12-String, so gespeichert, wie du ihn eingibst. Als eigenes Kopierfeld und QR-Code gezeigt.

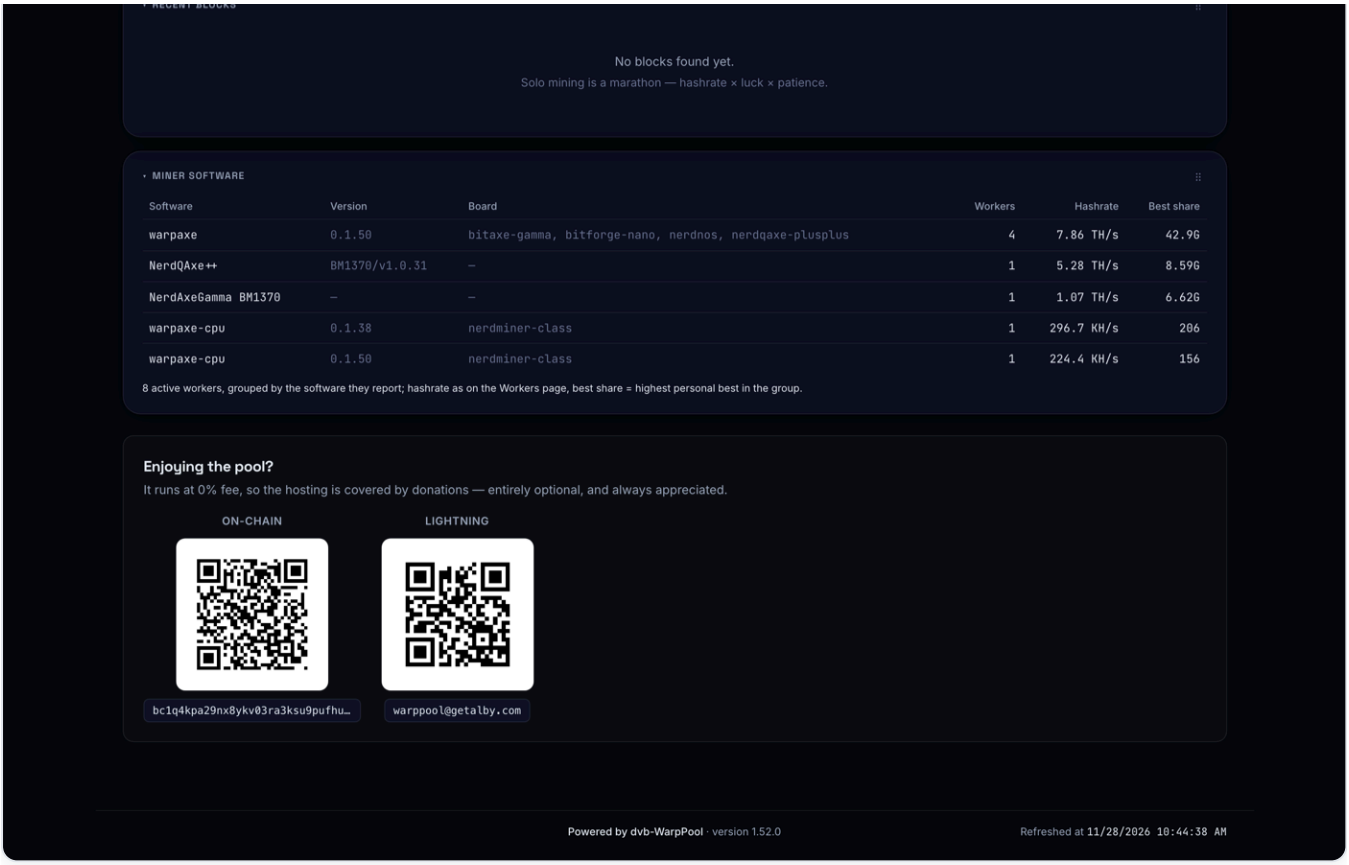
Lass eines der Felder leer, um diese Option auszublenden. Ist mindestens eines gesetzt, erscheint eine Karte **„Gefällt dir der Pool?“** mit QR-Codes und Adressen, die sich per Klick kopieren lassen, am **unteren Rand des öffentlichen dashboards**.



Admin → Spenden (nur per-worker-Pools): eine On-chain- und/oder Lightning-Trinkgeldadresse setzen; eine Spendenkarte erscheint dann auf dem öffentlichen dashboard.

Spenden berühren nie die Blockbelohnung. Diese Adressen werden als Pool-Einstellungen gespeichert (nicht in `config.toml`), sodass das Speichern sofort **ohne Neustart** wirksam wird. Sie sind reine Betreiber-Metadaten — sie werden **nie in den coinbase-Bau eingemischt**, sodass das Setzen einer Adresse die Fail-closed-Regel nicht beeinflussen kann, dass ein per-worker-Pool dem Finder die volle Belohnung zahlt. Eine „Spende“ ist hier ein Besucher, der sats an eine angezeigte Adresse sendet, nicht ein Anteil an der Blockbelohnung von irgendjemandem.


dvb-WarpPool
Dashboard
Hashrate
Overview
Blocks
Workers
Miners
RPC READY
IPC
LARGE (≤100 MINERS)
LIVE
EN
Login
Dashboard
Live status of your Bitcoin solo pool.
How to connect your miner
POOL
POOL HASHRATE LIVE
15.63 TH/s
24h avg 14.88 TH/s · 32.0 /min (2.13 /worker)
BLOCKS
0
WORKERS
16
39 miner
BEST SHARE
42.92G
bc1qtg…eqz0.bitaxe602
upd. 7/23/2026
UPTIME
1d 4h
ACTIVE MINERS
4 / 15
4 worker · 15 miner
3 TLS · 6 SV2
POOL FEE
0.00 %
POOL HASHRATE (24H)
Full view →
BEST SHARE
588M
441M
294M
147M
0
35.0 TH/s
30.0 TH/s
25.0 TH/s
20.0 TH/s
15.0 TH/s
10.0 TH/s
5.00 TH/s
0 TH/s
10:43
12:00
14:00
16:00
18:00
20:00
22:00
00:00
02:00
04:00
06:00
08:00
10:00
NETWORK
NETWORK HASHRATE
948.77 EH/s
DIFFICULTY
145.09 T
LAST DIFF CHANGE
+7.61 %
at Block #977.760
DIFF FORECAST
-4.95 %
at Block #979.776
NEXT DIFF
13d 13h
1.855 blocks remaining
BLOCK HEIGHT
977.922
BTC PRICE
$59,539
54.533 €
upd. 1m ago
CHAIN
main
Bitcoin Core 31.3.0
BLOCK
Block assembly checked by Core: OK (9m ago, height 977922)
ACTIVE WORKERS
USER
HASHRATE
DIFF
SHARES
LAST SEEN
bc1qtg…eqz0.WarpMiner 224.4 KH/s 0.009 0.36 /min 0s ago
bc1qtg…eqz0.bitaxe602 1.11 TH/s 2.84k 5.44 /min 6s ago
bc1qtg…eqz0.NerdQaxe++ 4.90 TH/s 11.63k 5.89 /min 6s ago
bc1qtg…eqz0.BitForgeNano 1.83 TH/s 6.21k 4.11 /min 18s ago
15Hifh…TMUY 5.51 TH/s 14.44k 5.33 /min 40s ago
bc1qtg…eqz0.NerdNOS 189.5 GH/s 1.04k 2.56 /min 57s ago
bc1qk0…syhp.TimsNerdAxeGamma 1.07 TH/s 5.51k 3.00 /min 1m ago
bc1q4y…zspt.LilyGO 296.7 KH/s 0.005 0.50 /min 2m ago
1Bitco…9kuE – – – 46m ago
All workers →
BEST SHARES
All Time
Per Worker
Rank Worker Difficulty When Hash
bc1qtg…eqz0.bitaxe602 42.92G 7/23/2026 199dee6cefbe…3c9ca18fcc56
bc1qtg…eqz0.NerdQaxe++ 19.53G 11/17/2026 384af3541a97…c1afd81f916c
bc1qtg…eqz0.NerdQaxe++ 11.51G 11/27/2026 5f8e5adef237…f55da7a8b992
#4 bc1qtg…eqz0.NerdOctaxe 10.71G 8/13/2026 66a824b0acd9…35508ca9498c
#5 bc1qtg…eqz0.NerdOctaxe 9.72G 8/14/2026 711d7240efea…3ae7aeb509eb
#6 15Hifh…TMUY 8.59G 9/27/2026 8002a41fdc00…ad0d47121dbb
#7 bc1qtg…eqz0.AvalonQ 8.52G 7/17/2026 8113f47a8de1…757e58a6141d
#8 bc1qtg…eqz0.NerdQaxe++ 7.05G 9/12/2026 9bf171c07dc0…5befad08c821
#9 bc1qtg…eqz0.NerdAxeGamma 6.89G 8/25/2026 9f942b44525b…25197368acfd
#10 bc1qk0…syhp.TimsNerdAxeGamma 6.62G 10/28/2026 a607b0f96e9f…ddfa06a6485f
Show all (50)



Das öffentliche dashboard, wie es ein anonymer Besucher sieht — Live-Pool- und Netzwerk-Statistiken, ein Best-Shares-Leaderboard mit maskierten Adressen (jeder Miner behält sein Rig-Label) und die optionale Spendenkarte mit QR-Codes am unteren Rand.

Wartungen an deine Miner ankündigen

Ein öffentlicher Pool hat Besucher, die du nicht über Discord erreichst. Unter **Admin** → **Banner** kannst du einen kurzen Hinweis — ein Wartungsfenster, einen geplanten Node-Resync — als laufendes Band oben auf der Pool-Seite anbringen, wo ihn jeder sieht, der bei dir mint. Du legst fest, wie lange er laufen soll (Minuten, Stunden oder Tage), und er verschwindet von allein, wenn die Zeit um ist, sodass ein vergessener Hinweis nicht hängen bleibt; ein Stoppen-Knopf beendet ihn vorzeitig. Der Text wird als reiner Text gezeigt und ist auf 200 Zeichen begrenzt. Geplante Ausfälle so anzukündigen ist auf einem Pool, auf den sich andere verlassen, schlicht guter Stil.

7 · Härtungsprofil für den öffentlichen Betrieb

Ein Pool, der dem Internet ausgesetzt ist, braucht die Missbrauchs-Kontrollen, die ein reiner Freundes-Pool überspringen kann. Jeder Schlüssel unten existiert bereits; keiner ist per-worker-spezifisch — sie gelten für jedes öffentliche Deployment.

```
[stratum]
# Bound how many simultaneous V1 connections one source IP may hold. On a
# public pool this stops a single host from hogging slots. Leave 0 only if you
# expect many miners behind one NAT.
max_connections_per_ip = 50

[ratelimit]
enabled = true
connects_per_sec = 5
connect_burst = 20
auths_per_sec = 1.0
auth_burst = 10

# Auto-ban IPs that keep tripping the connect limit (escalating). Loopback and
# private LAN are never banned.
autoban_enabled = true
autoban_strikes = 3
autoban_base_secs = 60
autoban_max_secs = 86400
autoban_window_secs = 600

# Ban a worker that floods invalid shares (share-flood defence). Off by
# default; turn on for a public pool.
reject_ban_enabled = true
reject_ban_min_samples = 100
reject_ban_ratio = 0.9

# Optional subnet access control. `ban_cidrs` denies ranges outright;
# `allow_only = true` flips to a strict whitelist (only allow_cidrs connect).
# ban_cidrs = ["203.0.113.0/24"]
# allow_only = false
# allow_cidrs = ["192.168.0.0/16"]
```

IPv6-Clients werden nach ihrem /64 gezählt (seit v1.50). Ein Internetanschluss mit IPv6 bekommt meist einen ganzen Adressblock — ein „/64“, die erste Hälfte der Adresse —, und das Gerät dahinter darf sich darin jede Adresse aussuchen. Damit dieser eine Client die Grenzen nicht durch einen Adresswechsel umgehen kann, zählt jede der Grenzen pro Client weiter oben (Verbindungs- und Login-Raten, der Auto-Ban, der Share-Flood-Ban, `max_connections_per_ip`) einen öffentlichen IPv6-Client nach seinem /64. Eine Sperre, die du von Hand unter **Admin → Profil → IP-Sperrliste** auf eine IPv6-Adresse setzt, sperrt ihr ganzes /64; die Liste zeigt und entsperrt die Adresse weiterhin so, wie du sie eingegeben hast. **Unique-Local**- (`fc00::/7`) und **Link-Local**-Adressen (`fe80::/10`) zählen als dein eigenes LAN, wie private IPv4-Adressen: pro Adresse und nie automatisch gesperrt. Für IPv4 ändert sich nichts.

Node-Ausfälle und deine Miner. Liefert Bitcoin Core 10 Minuten lang kein Block-Template, trennt der Pool jeden Miner und weist neue Verbindungen ab, bis wieder Templates fließen, damit die Miner auf ihrem

Fallback-Pool hashen statt auf einem veralteten Job (seit v1.45 standardmäßig an; `[server] node_unhealthy_disconnect` und `node_unhealthy_grace_secs`). Sag den Leuten, die auf deinem Pool minen, dass sie in ihrer Firmware einen Fallback-Pool eintragen sollen.

Invite-only-Variante

Kombiniere per-worker-Auszahlungen mit einer **Adress-Allowlist**, sodass nur bekannte Adressen minen dürfen. Ein Miner, dessen Auszahlungsadresse nicht gelistet ist, wird auf beiden Transportwegen abgelehnt (V1-authorize und Sv2-Channel-Open):

```
[mining]
payout_mode = "per_worker"
address_allowlist = ["bc1qalice...", "bc1qbob..."]
```

Eine leere `address_allowlist` (der Standard) lässt den Pool für jeden offen. Beachte, dass `allow_only` / `allow_cidrs` festlegen, *welche Netzwerke* sich verbinden dürfen, während `address_allowlist` festlegt, *welche Auszahlungsadressen* minen dürfen — die beiden sind unabhängig.

8 · TLS

Firmware, die nur öffentlichen Zertifizierungsstellen vertraut, braucht ein echtes Zertifikat. Die relevanten Schlüssel liegen in `[stratum]`:

```
[stratum]
stratum_tls_listen = ":3334"
tls_cert_path = "/etc/dvb-warppool/tls/cert.pem"
tls_key_path  = "/etc/dvb-warppool/tls/key.pem"
tls_reload_secs = 300
```

Lass die beiden Pfade **ungesetzt**, um das automatisch generierte selbstsignierte Zertifikat des Daemons zu nutzen — in Ordnung für Firmware der Klasse NerdMiner/Bitaxe, bei der du es pinnen kannst.

Mit dem selbstsignierten Zertifikat zeigt das Panel „Miner verbinden“ des Pools außerdem eine Zeile **TLS-Schlüssel-Fingerabdruck (SHA-256)**, eine Zeichenfolge wie `AB:CD:…`, die den Schlüssel des Pools kennzeichnet. Firmware, die ein selbstsigniertes Zertifikat anhand seines Fingerabdrucks pinnt, zeigt denselben Wert, bevor sie dem Pool vertraut — **WarpAxe 0.1.41 und neuer** tut das und bittet den Besitzer des Miners um Bestätigung —, sodass die Miner-Besitzer beide Werte vergleichen können. Mit einem CA-signierten Zertifikat fehlt die Zeile, weil diese Prüfung nicht nötig ist.

Erneuerungen laufen von allein. Ein echtes Zertifikat von Let's Encrypt wird alle paar Monate ersetzt, und was auch immer es ausgestellt hat — Caddy, certbot, deine eigene PKI — schreibt dabei einfach die beiden Dateien neu. Der Pool bemerkt das und liefert von sich aus das neue Zertifikat aus. Du musst ihn dafür nicht neu starten, und Miner, die gerade verbunden sind, bleiben verbunden: Sie behalten das Zertifikat, auf das sie sich bereits geeinigt haben, und nur neue Verbindungen bekommen das frische. `tls_reload_secs` legt fest, wie oft der Pool nachsieht (Standard 300 Sekunden; setze es auf `0`, um die Prüfung abzuschalten). Wird eine Erneuerung halb geschrieben erwischt, liefert der Pool weiter das alte Zertifikat aus und versucht es beim nächsten Durchlauf erneut, statt den Listener fallen zu lassen.

Stratum V2. Wenn du auch Stratum V2 betreibst, lauscht es an seiner eigenen `sv2_listen`-Adresse (zum Beispiel `:34254`). Der Sv2-Transport ist immer NOISE-verschlüsselt, braucht also kein separates TLS-Zertifikat. Miner brauchen zusätzlich den öffentlichen NOISE-Authority-Schlüssel des Pools, um sich überhaupt zu verbinden. Zwischen *Standard*- und *Extended*-Channel wählen deine Miner in ihrer eigenen Firmware; verweise sie auf *Extended*, weil sie damit die Coinbase prüfen können, die sie hashen. Hier wird beides angenommen, und auf deiner Seite musst du dafür nichts einstellen.

Ein Port für alles. Mit `single_port = true` in `[stratum]` nimmt der unverschlüsselte Port (`[server] pool_listen`, 3333) auch Stratum V1 über TLS und Stratum V2 an. Jede Verbindung wird an ihren ersten Bytes erkannt und an den passenden Server übergeben, mit denselben Limits, Sperren und Zählern wie auf dem eigenen Port des Protokolls. TLS und Sv2 brauchen weiterhin ihre eigenen Listener wie oben konfiguriert, und diese Ports laufen weiter. Hinter einem TCP-Balancer kommt der PROXY-Header zuerst und wird geprüft, bevor das Protokoll erkannt wird (Kapitel 9). Ab Werk aus.

9 · Hinter einem Reverse Proxy

Wenn die Status-UI über HTTPS durch nginx oder Caddy ausgeliefert wird, setze drei Schlüssel in `[server]`:

```
[server]
status_public_url = "https://pool.example.com"
trust_proxy_headers = true    # so per-IP limits see the real client IP
cookie_secure = true          # UI is HTTPS
```

`status_public_url` nennt die eigene öffentliche Adresse des Pools: Ein Miner, dessen Pool-Einstellung auf diesen Host zeigt, gilt als hier schürfend (Tab „Miner“, Meldung „Miner mint jetzt auf einem anderen Pool“). Setze `cookie_secure = true`, wann immer die UI über HTTPS erreicht wird.

> **Vertraue nur Proxy-Headern, die du kontrollierst.** Mit `trust_proxy_headers = true` vertraut der Daemon dem Header `X-Forwarded-For`, um die echte Client-IP zu ermitteln. Aktiviere es **nur**, wenn ein Proxy, den du kontrollierst, diesen Header tatsächlich setzt — andernfalls könnte ein Client seine **IP fälschen** und direkt an deinen Per-IP-Limits und Bans vorbeikommen.

Stratum hinter einem TCP-Balancer: das PROXY-Protokoll

Die Stratum-Ports sind rohes TCP und laufen über einen L4-Balancer (HAProxy TCP-Modus, nginx `stream`), nie über den HTTP-Proxy. Hinter so einem Balancer sähe der Pool nur dessen Adresse — ein einzelner störender Miner könnte so den ganzen Balancer sperren lassen. Seit v1.45 kann der Balancer die echte Client-Adresse vorausschicken (PROXY-Protokoll v1/v2, HAProxy `send-proxy-v2`, nginx `proxy_protocol on`); der Pool liest sie **nur** von den Adressen, die du einträgst, und die **müssen** sie dann schicken:

```
[server]
pool_listen = "127.0.0.1:13333"       # Daemon auf internen Ports, der Balancer besitzt 3333
[stratum]
stratum_tls_listen = "127.0.0.1:13334"
sv2_listen = "127.0.0.1:13336"
proxy_protocol = true
proxy_protocol_trusted = ["127.0.0.1"]   # Adresse(n) des Balancers
```

Rate-Limits, Sperren, das Per-IP-Limit und die Worker-Seite zeigen dann den Miner, nicht den Balancer. Beachte: Jede Verbindung von einer eingetragenen Adresse muss den Header tragen — ein lokaler Miner oder Sidecar, der direkt auf die Daemon-Ports geht, wird getrennt; lass ihn stattdessen auf den Balancer zeigen.

10 · Deployment-Checkliste

Gehe das durch, bevor du den Pool für die Öffentlichkeit öffnest:

- ☐ `payout_mode = "per_worker"`, mit **keiner** Pool-Gebühr oder Betreiberspende (der Daemon startet — sonst fällt er geschlossen aus).
- ☐ Bitcoin Core erreichbar; erwäge `[node] template_source = "ipc"` für geringere Latenz (braucht Core ≥ 31 mit IPC gebaut — GBT bleibt der Standard).
- ☐ `[ratelimit] enabled = true` mit `autoban_enabled` und `reject_ban_enabled` an.
- ☐ `max_connections_per_ip` gesetzt (außer deine Miner sitzen hinter einem NAT).
- ☐ TLS vor Stratum (dein eigenes Zertifikat für Firmware, die nur öffentlichen CAs vertraut).
- ☐ Reverse Proxy: `trust_proxy_headers` und `cookie_secure` korrekt gesetzt.
- ☐ Teste einen Miner mit `your-address.rig1`; bestätige, dass er unter `/users/<address>` erscheint und dass seine Shares akzeptiert werden.
- ☐ Wenn du eine Landing-Page veröffentlichst, halte die Ehrlichkeits-Leitplanken ein: kein „battle-tested“; der Mainnet-Block ist „noch nicht“.
- ☐ Optionale `[legal]`-Fußzeile (Terms-/Privacy-Links) für ein öffentliches oder kommerzielles Deployment.

11 · Support & Links

Online-Anleitung	docs.warppool.org/public-pool-setup.html
Quellcode & Issues	git.warppool.org/dvb-projekt/dvb-WarpPool
Funktionsvergleich	docs.warppool.org/comparison.html
Kontakt	dvbprojekt@gmx.de

dvb-WarpPool ist freie und open-source Software (AGPL-3.0-or-later, mit verfügbarer kommerzieller Lizenz). Dieses Handbuch beschreibt Version 1.52.0. Bildschirme können sich zwischen Releases leicht unterscheiden. Solo-Mining ist von Natur aus probabilistisch; nichts hier ist ein Versprechen einer Belohnung. Ein öffentlicher 0%-Pool ist non-custodial — der Pool hält nie die Coins von irgendjemandem, und der Finder wird direkt vom Bitcoin-Netzwerk mit der vollen Belohnung bezahlt.



Offen, non-custodial, gebührenfrei — und möge der Finder alles behalten.